



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**NÁVRH SYSTÉMU PRO SBĚR PROVOZNÍCH DAT
OBRÁBĚCÍHO STROJE**

DESIGN OF DAQ SYSTEM FOR PRODUCTION MACHINE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Eduard Sláma

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Andrš, Ph.D.

BRNO 2018

Zadání diplomové práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Bc. Eduard Sláma**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Ondřej Andrš, Ph.D.**
Akademický rok: 2017/18

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 systému pro sběr provozních dat obráběcího stroje

Stručná charakteristika problematiky úkolu:

Cílem této práce bude navrhnout a implementovat systém pro sběr dat z obráběcího stroje. Základní částí uvažovaného systému bude řídicí jednotka, která bude měřit provozní data (teploty, vibrace, proudy, napětí a popřípadě deformace) v procesu výroby uvažovaného zařízení. Řídicí jednotku bude představovat platforma cRIO firmy National Instruments doplněná o příslušné měřicí moduly dle specifických měřených veličin. Měřená data se budou kromě jednotky ukládat i do databáze a reportovat prostřednictvím emailu.

Cíle diplomové práce:

1. Seznámit se s vývojovým prostředím LabVIEW a platformou cRIO.
2. Provést rešeršní studii daného tématu.
3. Navrhnout systém pro sběr dat.
4. Implementovat systém pro sběr dat.
5. Implementovaný systém otestovat a vyzkoušet.

Seznam doporučené literatury:

HAVLÍČEK Josef, VLACH Jaroslav, VLACH Martin, LACHOVÁ Viktorie: Začínáme s LabVIEW, BEN - technická literatura, 2008, ISBN /978-80-7300-245-9

LACKO Ľuboslav : Mistrovství v SQL Server 2012, COMPUTER PRESS, ISBN: 978-80-251-3773-4

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

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

ABSTRAKT

Cílem diplomové práce je zhotovit funkční aplikaci pro sběr dat z obráběcího stroje s využitím programu LabVIEW a platformy cRIO. Diplomová práce se v první části zabývá obecně obráběcími stroji a poté již konkrétními případy, které mají na přesnost obrábění vliv, a možnostmi jejich měření. Druhá část diplomové práce je věnována již konkrétní tvorbě programu. Navržený program využívá jak LabVIEW FPGA Modul, tak i LabVIEW Real-Time Modul.

ABSTRACT

The aim of the diploma thesis is to make functional application for design of DAQ system for productive machine by using LabVIEW and platforms cRIO. The first part is focused on productive machine. After the thesis continues with specific examples. These examples have impact on machining accuracy and their measurements. The second part of diploma thesis is focused on making measurement program. LabVIEW FPGA Module, LabVIEW Real-Time Module are used in the program.

KLÍČOVÁ SLOVA

Senzory, LabVIEW, LabVIEW FPGA Modul, LabVIEW Real-Time Modul, Chassis, C Series Module

KEYWORDS

Sensors, LabVIEW, LabVIEW FPGA Module, LabVIEW Real-Time Module, Chassis, C Series Module

BIBLIOGRAFICKÁ CITACE

SLÁMA, E. *Návrh systému pro sběr provozních dat obráběcího stroje*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 82 s. Vedoucí diplomové práce Ing. Ondřej Andrš, Ph.D.

PODĚKOVÁNÍ

Chtěl bych poděkovat svému vedoucímu Ing. Ondřeji Andršovi, Ph.D., za cenné připomínky a rady při tvorbě mé práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Ondřeje Andrše, Ph.D., a s použitím literatury uvedené v seznamu.

V Brně dne 23.5.2018

.....

Sláma Eduard

OBSAH

1	MOTIVACE	15
2	NC OBRÁBĚCÍ STROJE	16
2.1	Historie automatizovaných obráběcích strojů	16
2.2	Rozdělení automatizovaných obráběcích strojů.....	16
2.3	Zařazení NC a CNC strojů do výrobního procesu	18
3	VLIVY NA PŘESNOST OBRÁBĚCÍCH STROJŮ	20
3.1	Tepelné vlivy.....	21
3.1.1	Vnější tepelné vlivy	21
3.1.2	Vnitřní tepelné vlivy	21
3.2	Vibrace-kmitání.....	23
3.2.1	Kmitání vlastní volné.....	24
3.2.2	Kmitání buzené	24
3.2.3	Kmitání samobuzené.....	26
3.2.4	Trhavé pohyby	26
3.2.5	Kompensace vibrací.....	26
3.3	Deformace stroje	27
3.3.1	Statická tuhost	28
3.3.2	Dynamická tuhost	29
3.4	Shrnutí poznatků z možných deformací obráběcího stroje	29
4	SENZORY.....	30
4.1	Základní vlastnosti senzorů.....	30
4.2	Senzory teploty.....	31
4.2.1	Dotykové senzory teploty	31
4.2.2	Bezdotykové senzory teplot	32
4.3	Senzory síly	33
4.3.1	Odporové tenzometry.....	33
4.3.2	Piezoelektrické senzory síly	34
4.4	Senzory vibrací.....	35
4.4.1	Akcelerometry.....	35
4.4.2	Gyroskopy.....	36
4.5	Senzory napětí a proudu.....	37
4.6	Shrnutí poznatků o senzorech	38
5	MOŽNOSTI ZPRACOVÁNÍ DAT SE SENZORŮ	39
5.1	PLC	39
5.2	dSPACE	39
5.3	Platformy National Instruments	40
5.3.1	CompactRIO	41
5.3.2	LabVIEW pro Host-PC.....	42
5.3.3	LabVIEW Real-Time	43
5.3.4	Labview FPGA	44
5.3.5	Ukládání dat	45
6	DATABÁZE.....	47
7	POUŽITÉ MĚŘICÍ VYBAVENÍ	50
7.1	NI CompactRIO-9036.....	50
7.2	C Series Modul NI-9237	52
7.3	C Series Modul NI-9234	52

7.4	Backhoff EK1100.....	53
8	PROGRAM V LABVIEW.....	55
8.1	Makroarchitektura programu	56
8.2	Program běžící na Host-PC	58
8.2.1	Uživatelské rozhraní programu	58
8.2.2	Struktura programu	61
8.3	Program v LabVIEW Real-Time	63
8.3.1	Uživatelské rozhraní programu	63
8.3.2	Struktura programu	64
8.4	Program v LabVIEW FPGA	67
8.4.1	Uživatelské rozhraní programu	67
8.4.2	Struktura programu	68
8.5	Struktura zdrojového kódu (projektu)	69
9	STRUKTURA DATABÁZE.....	70
10	OVĚŘENÍ FUNKČNOSTI PROGRAMU.....	72
11	ZÁVĚR.....	76
12	SEZNAM ZDROJŮ	77
13	SEZNAM OBRÁZKŮ.....	80
14	SEZNAM PŘÍLOH	82

1 MOTIVACE

S rozvíjejícím se průmyslem, elektroniky a znalostmi řízení je stále větší důraz kladen na automatizaci podniků, čímž vzniká pojem průmysl 4.0. V průmyslu 4.0 je nutností zajistit plynulý chod přístrojů a výrobního procesu s co možná nejmenším využitím lidské práce. Aby toho bylo dosaženo, je potřeba neustále monitorovat chod strojů. Na chod strojů může mít vliv mnoho faktorů, a proto je potřeba všechny faktory monitorovat a vyhodnocovat. Čím více bude informací o strojích shromážděno, tím lze snáze predikovat budoucí chování a popřípadě se lze lépe připravit na poruchové stavy stroje. Za tímto účelem nabírá na svém významu zadání diplomové práce *Návrh systému pro sběr provozních dat obráběcího stroje*. Práce se bude zabývat sestavením takového programu, který bude schopen sbírat data ze senzorů a následně je ukládat jak na disk, tak i na databázový server. Poté se mohou takto uložená data zpracovávat a vyhodnocovat jak pro budoucí návrh nového stroje, tak jako diagnostika stávajícího stroje.

Práce se bude věnovat v první části obecné problematice CNC strojů a vlivům na přesnost obrábění, přičemž budou uvedeny i senzory, kterými se dají nepřesnosti působící na obrábění měřit. Na tuto část navazuje již samotné pojednání o programech, které jsou využívány ke zpracování signálů ze senzorů a se kterými je pracováno v rámci diplomové práce. Hlavní motivace práce, je vytvořit funkční program pro sběr dat ze senzorů, který by poté mohl být používán pro sběr dat.

2 NC OBRÁBĚCÍ STROJE

Tématem práce je také měření provozních dat na CNC (computer numeric control) stroji, tudíž bude v následujících podkapitolách rozebrána historie NC (numeric control) strojů. Historie bude rozebrána pro usnadnění vstupu do problematiky NC obráběcích strojů. Pro celkovou orientaci v problematice obráběcích strojů se tato kapitola bude zabývat také rozdělením NC obráběcích strojů a jejich zařazením do automatizovaného výrobního systému.

2.1 Historie automatizovaných obráběcích strojů

Počátky NC stroje jsou datovány do poloviny dvacátého století a obecně je NC řízení přisuzováno Parsonsovi. Jako první si všiml, že pro výpočty se používají kalkulačky s děrnými štítky a rozhodl se tuto myšlenku aplikovat na obrábění. Jeho myšlenka byla velice jednoduchá, stačilo stanovit body, ve kterých se provede řez, a následně v těchto bodech řez provést. Sestavit tabulku odpovídajících bodů, ve kterých se mají provádět řezy, nebyl problém. Takto sestavená tabulka byla předána do dílny, kde následně proces obrábění probíhal. Dělníci si tabulku převzali a postupovali podle ní tak, že jeden pracovník předčítal čísla z tabulky pro osy x a y a druhý pracovník v těchto místech prováděl řezy. Avšak při ručním ovládáním se čas nezkrátil, pouze se přesunul z jedno místa na druhé. Proto začal Parsons přemýšlet, jak tento proces více zautomatizovat. [1]

Parsonsovův nápad na automatickou frézu zprvu nezaujal, až po sléze letectví USA vyčlenilo pro Parsonse finanční prostředky, aby mohl zhotovit svoji frézku. Avšak Parsons nedokázal mít se svojí frézku požadovanou přesnost, a tak stále vznikaly jen hrubé obrobky. Hlavní problém byl v tom, že pro různé řezy byla potřeba jiná energie a nebylo možné lineárně ovládat posun. [1]

Z toho důvodů se Parsons rozhodl obrátit na Massachusetts Institute of Technology (dále jen MIT), aby mu pomohli problém s regulací řezu vyřešit. Byla uzavřena trojnásobná dohoda mezi Parsonsem, MIT a letectvím. Nicméně nakonec MIT uzavřela nový kontrakt s letectvím bez Parsonsna. Nový produkt MIT sice slavil úspěch, protože dokázal vyrábět složité řezy s vysokou přesností, ale byl však příliš složitý a také velmi nákladný.

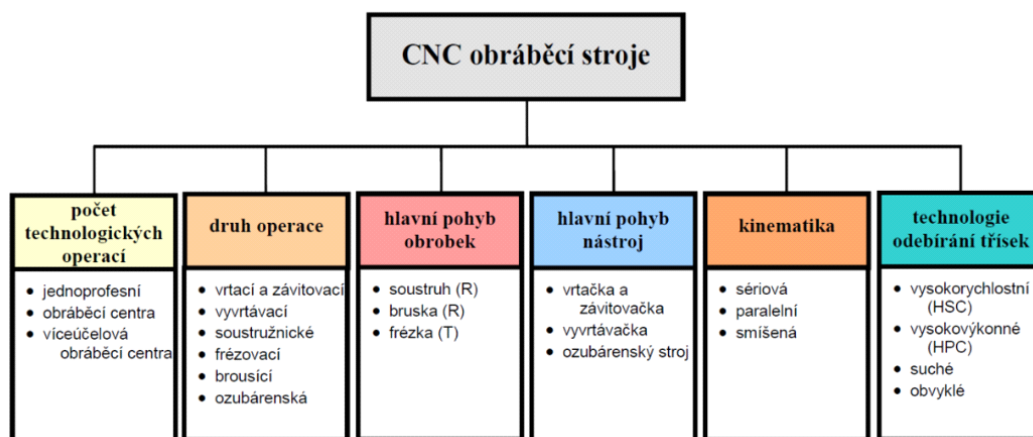
Kolem roku 1960 uvedla firma Kearney&Trecker první obráběcí centrum na trh. Následující roky se nesly ve znamení zdokonalování NC systému. „V 80. letech začaly být stroje vybavovány zásobníky nástrojů i obrobků a do konstrukce NC strojů se aplikovaly senzory pro sledování pohonů a jednotlivých mechanismů.“ [2] V tomto období došlo k prosazení frézovacích a soustružnických center do technologií třískového obrábění. S nástupem lepších a výkonnějších počítačů mohl začít vznikat složitější a propracovanější hardware i software. Přestávají se používat dírkované pásky a stroje začínají být řízeny počítačem, vznikají CNC stroje. V nynější době je snaha o co největší integraci do externí počítačové stanice, aby byla obsluha snadnější a s tím i související kontrola provozních dat obráběcích strojů. [2]

2.2 Rozdělení automatizovaných obráběcích strojů

V předchozí podkapitole bylo představeno něco málo z historie a vývoje NC obráběcích strojů a nyní se tato podkapitola bude zabývat rozdělením CNC obráběcích strojů. Toto téma je důležité k celkovému doplnění přehledu NC obráběcích strojů a k vstupu do problematiky komplexních obráběcích center, která jsou v nynější době čím dál více automatizovaná. S tím

přichází i více problémů, které je potřeba řešit. CNC obráběcí stroje lze rozdělit podle druhů operací, které vykonávají, a podle NC vývojové generace.

Jak lze vidět na obrázku 1, lze CNC obráběcí stroje dělit celkem do šesti kategorií. Můžeme si povšimnout kategorie *počet technologických operací*, v které jsou zařazeny CNC jednoprofesní obráběcí stroje, což znamená, že počítačové řízení (numeric control) se nevyužívá pouze pro důmyslná obráběcí centra, ale nachází uplatnění i v jednoprofesních obráběcích strojích. Na základě zmíněného faktu lze říci, že NC se dá využít kdekoliv, kde je na místě sériová výroba. V kategoriích mohou nastat různé kombinace jako například *CNC frézka se sériovou kinematikou pro vysokorychlostní obrábění*. [1] [3]



Obrázek 1: Rozdělení CNC obráběcích strojů z hlediska vykonávané práce [3]

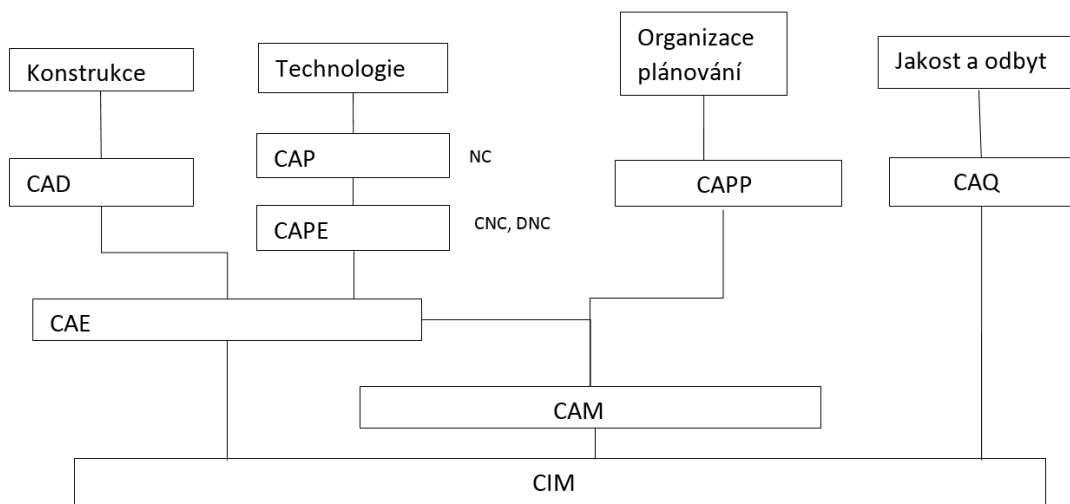
Dále lze dělit CNC obráběcí stroje do vývojových generací podle stupně vývoje NC systému, kde každá další generace má čím dál užší vazbu na osobní počítače. Nyní se dá obrazně říci, že lze obsluhovat a monitorovat chod CNC stroje z pohodlí domova, aniž by člověk musel být osobně přítomen u CNC stroje. Níže je popsáno rozdělení NC řízení do sedmi vývojových generací, podle úrovně NC řízení.

- **1. generace** – stroje odvozeny od běžných konvenčních strojů, do kterých bylo přidáno NC řízení.
- **2. generace** – stroje speciálně zkonstruované pro NC řízení, u kterých výměna opotřebovaných nástrojů probíhala ručně.
- **3. generace** – stroje uzpůsobeny pro nasazení v automatizovaných výrobních soustavách, u nichž lze měnit automaticky výrobky, avšak nikoli opotřebované nástroje.
- **4. generace** – v této generaci lze již automaticky vyměňovat jak výrobky, tak i opotřebované nástroje. Tato generace je plně automatická.
- **5. Generace** – u strojů se začínají uplatňovat mechatronické prvky, jako například kompenzace chyb nebo odměřování polohy pomocí laseru
- **6. Generace** – dochází k další modernizaci strojů předešlé generace, a to především v oblasti rychlosti obrábění a celkové diagnostiky chyb.
- **7. Generace** – nejvyšší generace strojů začíná využívat prvky umělé inteligence.

2.3 Zařazení NC a CNC strojů do výrobního procesu

Zkratky jako CNC a NC mohou být do značné míry matoucí, protože se často zkratka NC používá i tam, kde je zjevné, že stroj řídí počítač, a tudíž použití zkratky CNC by bylo dle mého názoru vhodnější, i když užití NC zkratky není zcela špatné, ale pro čtenáře zavádějící. Dá se říci, že všeobecně používané zkratky jako například NC, CNC nebo CAD se vážou s mnoha dalšími zkratkami jako například s CIM, CAM nebo CAE, které jsou jim nadřazený.

Jak jsem již psal, vývoj NC strojů jde ruku v ruce s vývojem hardwaru a softwaru. Čím lepší software bude, tím lepší je možno zrealizovat NC řízení. V nynější době jsou elektronické součástky v daleko vyšším stádiu vývoje než například před třiceti lety, a proto lze realizovat složitější, ale zároveň výkonnější hardware, díky čemuž je možno realizovat i propracovanější software. Nyní nastává otázka, proč stroj sedmé generace, který má adaptivní řízení, nepropojit i s jinými stroji a tím zautomatizovat celý výrobní proces. Bude potřeba méně lidské práce, čímž se minimalizují lidské chyby, a výkonost stoupne. Každý si jistě dokáže představit mnoho dalších výhod a nevýhod zautomatizované výroby. S touto myšlenkou vzniká nový pojem, který nabývá v poslední době na svém významu, a to je Industry 4 a s ním ve výrobě CIM (Computer Integrated Manufacturing) neboli počítačem integrovaný výrobní systém. Aby mohl vzniknout CIM, je potřeba splnit několik dílčích kroků. Tyto kroky znázorňuje obrázek 2 níže.



Obrázek 2: Návaznost systému na CIM [2]

Zkratky, které lze vidět na obrázku 2, jsou zkratkami anglických slov, a to následujících: [2]

CIM (Computer Integrated Manufacturing) – počítačem integrovaný výrobní systém

CAM (Computer Aided Manufacturing) – systém počítačové podpory výroby, který zahrnuje přímé řízení NC techniky, robotů, mezioperační dopravu materiálu, polotvarů i výrobků a nástrojů

CAE (Computer Aided Engineering) – systém počítačové podpory inženýrských činností

CAD (Computer Aided Design) – počítačová podpora systému konstruování

CAPE (Computer Aided Production Engineering) – systém pro tvorbu a údržbu informací v TPV (technologická příprava výroby), který zahrnuje plánování výroby, technologičnost

konstrukcí, tvorbu technologických postupů, NC programů a volbu nástrojového i měřicího vybavení

CAP (Computer Aided Process Planning) – systém pro zpracování NC programu stroje

CAPP (Computer Aided Programming Planning) – systém, který zahrnuje plánování výroby včetně návrhů a tvorby korekcí plánů s ohledem na dodržování smluvených termínů zakázek a požadavků na materiální i nástrojové vybavení (zajištění)

CAQ (Computer Aided Quality) – systém počítačové podpory kontrol a řízení jakosti

CA (Computer Aided) – počítačová podpora

NC (Numerical Control) – číslicové řízení operací obrábění (přímé vkládání číslicových údajů)

CNC (Computer Numerical Control) – počítačem řízený NC stroj

DNC (Direct Numerical Control) – centrálním počítačem řízená a kontrolovaná síť NC strojů

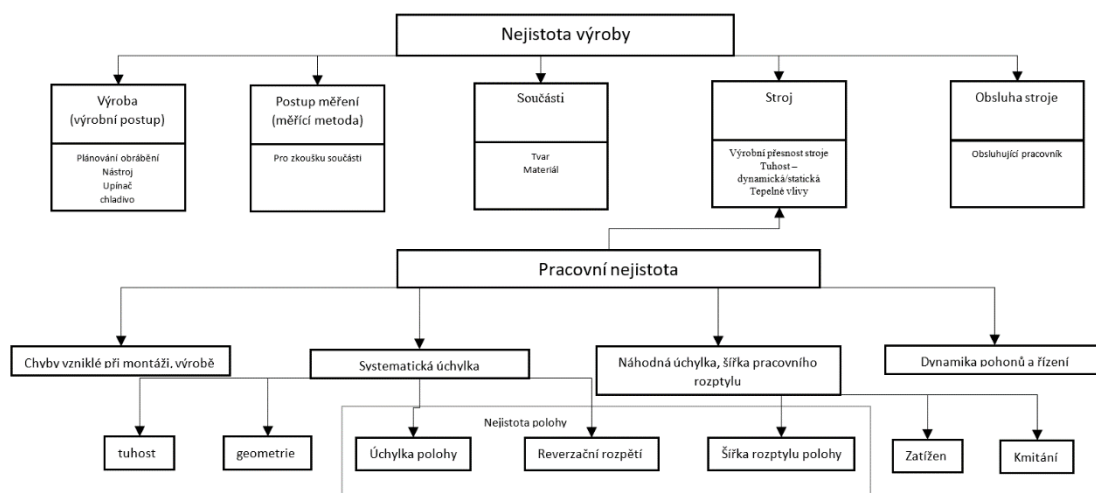
Jak již bylo zmíněno, k vytvoření integrovaného počítačem řízeného systému je potřeba splnit několik dílčích kroků. Dílčí kroky by nemohly být nikdy splněny bez přispění vývoje technologií. Názorným příkladem je systém CAD. Dříve bylo běžné, že inženýři rýsovali výkresy ručně na rýsovacím prkně, a nebylo možné tyto výkresy přenášet do elektronické podoby. S nástupem nových technologií se přešlo od rýsovacích prken k počítačům a teprve nyní se mohli začít inženýři zabývat tím, jak tyto výkresy promítnout do svých obráběcích strojů. Dále se například pracovalo na tom, jak odstranit problém s výrobky, které stojí na paletě, a není s nimi nijak zacházeno. Problémy s odbytem a jakostí bylo potřeba také vyřešit stejně tak jako i problémy s organizací a plánováním. Některé z výše zmíněných problémů upravují normy ISO 9001. Po vyřešení všech dílčích problémů začíná vznikat CIM neboli integrovaný počítačem řízený systém. [2]

3 VLIVY NA PŘESNOST OBRÁBĚCÍCH STROJŮ

Na přesnost obráběcího stroje má vliv mnoho faktorů, nelze říci, že stroj, který dokáže obrábět s určitou obráběcí přesností na jednom místě, bude i po převozu na jiné místo obrábět se stejnou přesností. Je potřeba každý stroj seřadit na místě a přizpůsobit ho okolním podmínkám. Okolní podmínky mohou mít na přesnost obráběcího stroje dalekosáhlé důsledky. Obecně lze říci, že na stroj působí tři hlavní vnější vlivy, jakými jsou:

- Teplo
- Vibrace
- Nečistoty

Řekněme ale, že stejnými vlivy může obráběcí stroj působit na okolí, a tím pádem ovlivňovat například jiné stroje nebo přímo svoji obráběcí přesnost, např. tvorbou vibrací na vřetenu a špatným odvodem tepla při obrábění. [1]



Obrázek 3: Nejistota výroby a pracovní nejistota [1]

Na obrázku 3 lze vidět, že na míru nejistoty výroby neboli na míru přesnosti, s „níž může být vyrobena součást na zadaném stroji při definovaném provozním stavu“ [1] nemají vliv pouze tři výše zmíněné faktory, ale také součásti, výroba, obsluha stroje, postup měření.

Jak si lze povšimnout, existují dva druhy nejistot, a to nejistota výroby a pracovní nejistota. Jak již bylo zmíněno výše, na nejistotu výroby kromě obráběcího stroje mají vliv i další faktory. Špatné upnutí součásti může mít stejný dopad jako špatně zhotovený pracovní postup, nehledě na to, že je potřeba mít správně kvalifikovanou obsluhu, aby bylo vše správně připraveno. Zmíněné nejistoty mají na celkovou nejistotu výroby velký, a proto nezanedbatelný vliv, avšak téma diplomové práce se týká měření a zpracování veličin z obráběcího stroje, a tudíž se budu dále zabývat pouze nejistotami výroby, které vytváří stroj, a nikoliv lidé.

Jak bylo uvedeno, na stroj mají vliv jak tepelné a dynamické jevy, tak i pracovní nejistoty, které mohou vznikat jednak při montáži, jednak při řízení pohonů. Montáž je velice důležitá, protože se lze při chybně namontovaném jednom prvku dopustit řady chybně namontovaných navazujících prvků (toleranční pole) a tím zvyšovat míru nepřesnosti stroje. Samotné provedení správné montáže nestačí, je potřeba korektně nastavit i samotné řízení stroje. Pro takové řízení se mohou využívat PID regulátory, zpětnovazební smyčky a další vyspělé metody řízení a regulace. Při chybném regulování se může stát, například při nájezdu nože na obrobek, že se

soustava rozkmitá, což je pro přesnost obrábění nežádoucí, nehledě na to, že rozkmitání stroje může mít za následek zničení obráběcího nože, anebo dokonce celého stroje. Rozkmitání stroje se snažíme zabránit. Samozřejmě, že vibrace neboli kmitání nemusí vznikat jen samotným strojem, ale vibrace mohou vznikat i v okolí stroje a jsou nežádoucí. Pro podrobnější pochopení problematiky budou v následujících podkapitolách rozebrány druhy nejistot, které na obráběcí stroj a na jeho přesnost působí, a to vibrace, tepelné vlivy, proudy, napětí, deformace. Zbylé nejistoty působící na obráběcí stroj nebudou popisovány z toho důvodu, že nejsou pro vypracování zadání diplomové práce nezbytné, a proto se s nimi nebude dále pracovat.

3.1 Tepelné vlivy

Každý materiál má svůj součinitel tepelné roztažnosti. Součinitel tepelné roztažnosti udává číslo, o kolik změní materiál svoji velikost při změně teploty o jeden stupeň celsia na součásti se jmenovitým rozměrem jeden metr. Tepelné vlivy mohou na součást působit z vnějšku nebo je může vytvářet stroj sám. Z toho důvodu lze rozdělit tepelné vlivy na vnější a vnitřní.

3.1.1 Vnější tepelné vlivy

Tepelné vlivy působící na obráběcí stroj z vnějšku jsou nejčastěji patrné v zimním období a v letních slunečných dnech. V zimním období dochází k vyhřívání místností v důsledku vytápění, a tím ke změně okolní teploty. Teplota v místnosti ve večerních hodinách bude odlišná od teploty v denních hodinách při provozu se zapnutým topením. Změna teploty může dosahovat řádu desítek stupňů. Nicméně je potřeba dbát i na to, kde bude stroj postaven, protože sluneční paprsky mají na stroj také vliv. Jestliže bude stroj stát na slunci celý den, zahřeje se a materiál se roztáhne o jeho teplotní součinitel, přičemž ve dnech, kdy bude zataženo, se materiál roztáhne o jiný teplotní součinitel, a tím pádem bude i změna geometrie stroje v obou případech odlišná, což bude mít za následek nepřesnosti při obrábění a nebude dosaženo požadované tolerance. Také je důležité se podívat na tzv. teplotní gradient, který bude jiný při jednorázovém otevření například okna (prudká změna teploty), a jiný při pomalém chladnutí místnosti. Dá říci, že teplotní gradienty, jsou změny teploty v čase.

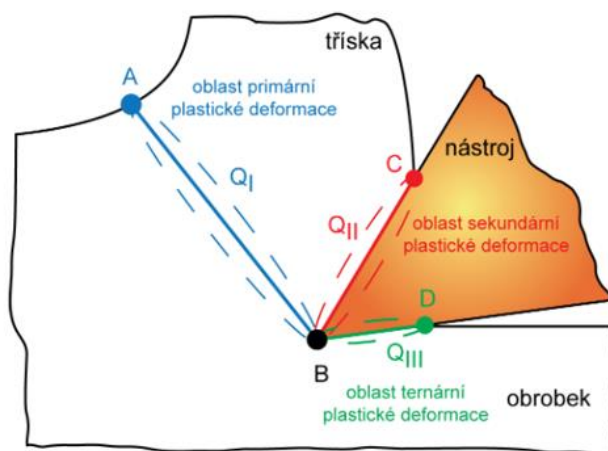
Pro materiály jsou teplotní gradienty důležité, protože každý materiál reaguje jinak na změnu teploty v čase. Podstatné je si uvědomit, že existují i polohové gradienty, které udávají informaci o teplotě v závislosti na nadmořské výšce, přičemž v nižších polohách bude teplota nižší než ve vyšších polohách. [4]

Problém s vnějším působením teploty může být vyřešen několika způsoby. Základní způsob je se snažit udržet konstantní teplotu v místnosti, což není vždy zcela možné, proto je aspoň potřeba umístit stroj tam, kde svítivost na stroj bude konstantní, ať už od žárovky, nebo od slunce. Dalším způsobem, ale nákladnějším, při němž se dá dosáhnout větší přesnosti, je odměřování teploty na různých místech u stroje, následné dopočítávání součinitele tepelné roztažnosti, zanesení chyby způsobené tepelnou roztažností stroje do systému a podle ní následná úprava řezných a jiných podmínek. Ale toto řešení není zcela triviální záležitostí, protože je potřeba brát v úvahu i teplo, které stroj sám svým provozem vyrábí, ale o tom v další podkapitole.

3.1.2 Vnitřní tepelné vlivy

Je zřejmé, že na obráběcí stroj má vliv i vnitřní teplo, tj. teplo, které si stroj sám vytváří. Vnitřní teplo může mít několik zdrojů, jednak ve zdroji, který napájí samotný obráběcí stroj, jednak např. v ložiscích a motorech vlivem tření, přičemž nejvíce tepla vzniká při samotném řezném

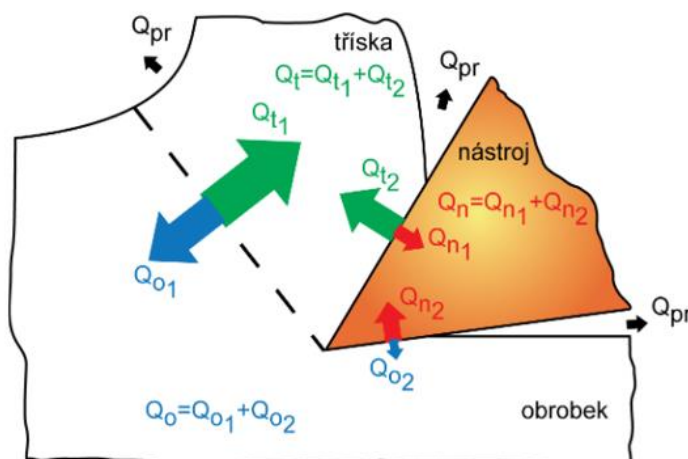
procesu. Vznikají jak plastické, tak i elastické deformace v důsledku tření mezi hřbetem nástroje a přechodovou plochou obrobku. [5]



Obrázek 4: Oblasti vzniku tepla při třískovém obrábění [5]

Na obrázku 4 si lze povšimnout různě velkých ploch, mezi kterými dochází k výměně tepla. Je patrné, že k přenosu největšího objemu tepla bude docházet mezi body A a B, tj. mezi obrobkem a třískou, zatímco nejmenší výměna tepla bude mezi body B a D, tj. mezi nástrojem a obrobkem. Dále si lze povšimnout plochy mezi nástrojem a třískou (body B a C). Zde se snažíme, aby plocha byla co nejmenší, a tím pádem nám přecházel co nejmenší objem tepla do nástroje a z nástroje do vřetena atd. Dochází k celkovému ovlivnění stroje teplem. Takto vzniklé teplo má vliv mimo jiné nejen na životnost nástroje a jeho opotřebení, ale také i na vlastnosti obrobku. Negativní vliv tepla vzniklého při třískovém obrábění lze minimalizovat pomocí vhodně zvoleného úhlu hřbetu nástroje a dále také například vhodně zvolených rezných podmínek.

Cílem je, aby co nejvíce vzniklého tepla bylo odvedeno do třísky, a z ní následně do okolního prostředí. Tím se zamezí negativním vlivům, které má teplo vzniklé při obrábění jak na obrobek i nástroj, tak na celkové mechanické vlastnosti stroje.



Obrázek 5: Oblasti vzniku tepla při třískovém obrábění [5]

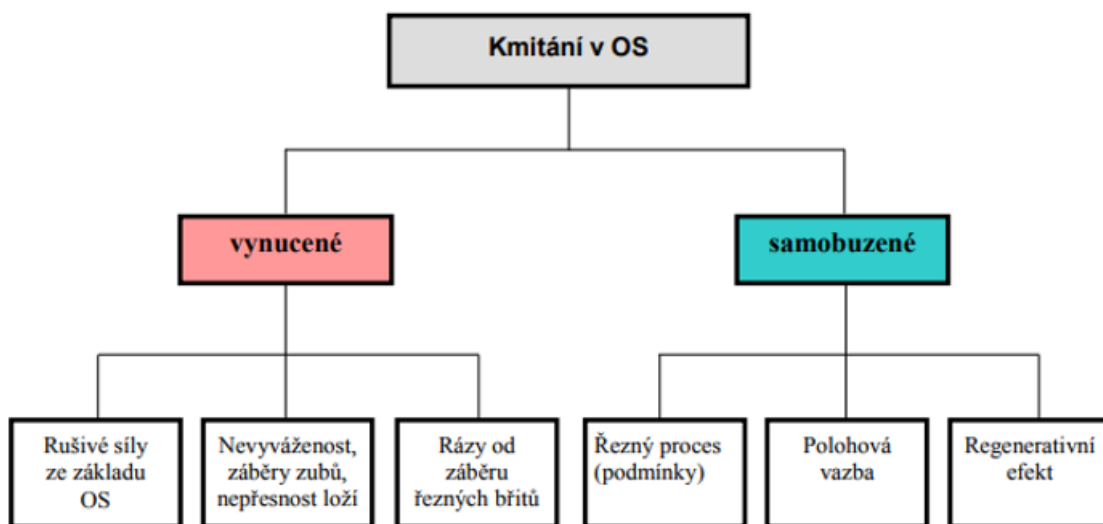
Obrázek 5 doplňuje představu o přechodu tepla při obrábění. Je důležité si uvědomit, že vzniklé teplo se musí rovnat teplu odvedenému, což plyne ze zákona zachování energie.

Samozřejmě teplo produkované samotným strojem, které vzniká z více zdrojů, jak při obrábění, tak i z elektrického napájení stroje, nebo dále například z mechanických vlastností stroje, má vliv na okolní teplotu prostředí a je potřeba se zamyslet nad tím, zda je výhodné dát dva stroje vedle sebe, anebo nikoliv, s vědomím tepelných jevů, které mohou ovlivňovat stroje.

Ovlivnit velikost vnitřního tepla a udržet ho konstantní je mnohem obtížnější úloha než u vnějšího působení tepla, avšak lze se aspoň přiblížit k udržení konstantního tepla. Hodně tepla, které lze ovlivnit, vzniká třením jak už v ložiscích, tak v motorech a všech na sebe dosedajících ploch. Z toho důvodu je potřeba mít dobré chlazení stroje a mazání. Chlazením stroje se odvádí teplo do okolí a tím dochází k zahřívání místnosti, kde je stroj umístěn, přičemž právě toto teplo může působit na okolní stroje. Jak již bylo zmíněno výše, hlavní část tepla, která je vytvářena strojem, vzniká při samotném řezném procesu. Toto vzniklé teplo lze rovněž snížit chlazením a mazáním, přičemž je snaha, aby co nejvíce tepla bylo odvedeno třískou a neměnil se tak tvar obráběcího nože. Teplo lze opět měřit na různých místech stroje a dopočítávat teplotní roztažnost, která má vliv na obrábění, a podle toho řídit obráběcí proces.

3.2 Vibrace-kmitání

Jak již bylo zmíněno, kmitání neboli vibrace jsou další z nežádoucích jevů, které působí na obráběcí stroj. Vibrace mohou opět vznikat jak vně stroje, tak i v okolí stroje, a mají nežádoucí dopad nejen na přesnost obráběcího stroje, ale také na samotnou životnost stroje. Nežádoucí vibrace zvyšují namáhání součástí až na mez pevnosti materiálů, a to může vést k fatálním důsledkům pro strojové součásti. Jak je již patrné z obrázku 6 v obráběcím stroji mohou vznikat vibrace vynucené nebo samobuzené, přičemž každý druh vibrací má jiný původ vzniku.



Obrázek 6: Příčiny kmitání v obráběcích strojích [6]

„Níc méně popsat kmitání v obráběcích strojích je velmi složité, neboť se jedná o soustavu vzájemně se ovlivňujících hmotných a pružných těles, která jsou spolu navzájem spojena.“ [8]

Přesto lze kmitání rozdělit na základní čtyři druhy, a to na:

- Kmitání vlastní (volné)
- Kmitání buzené (vynucené)
- Kmitání samobuzené
- Trhavé pohyby (Slip Stick)

Každý druh kmitání má jiný druh původu a také má odlišné dopady. Avšak nejdříve je potřeba si představit rovnici kmitání (1),

$$m\ddot{x} + b\dot{x} + kx = F(t) \quad (1)$$

kde:

m ...hmotnost

b ...tlumení

k ...tuhost pružiny

$\ddot{x}$...zrychlení

$\dot{x}$...rychlost

x ...poloha

$F(t)$...budicí síla

3.2.1 Kmitání vlastní volné

Kmitání vlastní volné může být jak tlumené, tak i netlumené, přičemž zanedbáváme budicí sílu $F(t)$ tak, že:

$$F(t) = 0 \quad (2)$$

Po modifikaci rovnice (1) s podmínkou z rovnice (2) dostáváme rovnici pro kmitání vlastní volné tlumené:

$$m\ddot{x} + b\dot{x} + kx = 0 \quad (3)$$

Za podmínek z rovnice (3) a jestliže:

$$b = 0 \quad (4)$$

lze získat rovnici pro kmitání vlastní volné netlumené kmitání:

$$m\ddot{x} + kx = 0 \quad (5)$$

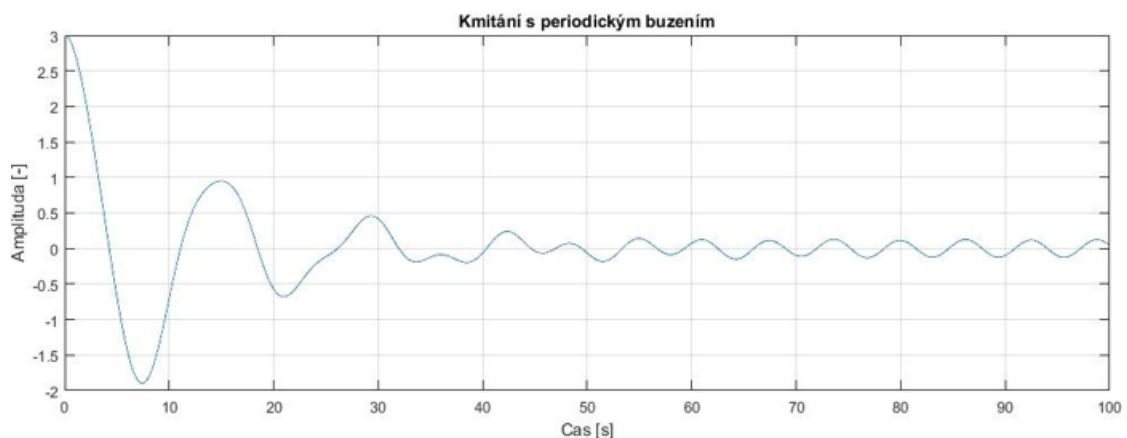
3.2.2 Kmitání buzené

Zde již nelze zanedbat budicí sílu. Charakteristické pro buzené kmitání je, že se shoduje budicí frekvence s frekvencí kmitání. Jak se lze dočíst ve zdroji číslo [7], příčinou vzniku těchto vibrací mohou být vlastnosti stroje nebo vlastní řezný proces, přičemž u vlastností stroje se jedná zejména o:

- nevyváženost rotujících součástí,
- setrvačné síly prvků, které konají přímočarý nebo kruhový vratný pohyb,
- periodické síly dané principem pohonů,
- nesymetričnost rotačních částí (hřídele s drážkou), kdy se projeví proměnlivá hodnota tuhosti a vznikne periodická deformace během otáčení,

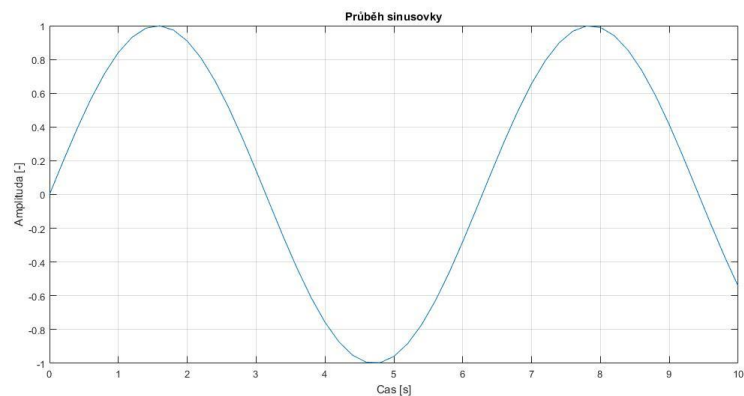
a u řezného procesu o:

- změnu průřezu třísky,
- proměnlivý řezný odpor,
- vynucené kmitání vlivem periodického házení obrobku nebo brusného kotouče při broušení,
- kmity vyvolané vlastním řezáním, které mají značnou amplitudu a projevují se hlavně při hrubování a v menší míře při dokončování.



Obrázek 7: Kmitání s periodickým buzením

Na obrázku 7 lze vidět průběh kmitání buzeného tlumeného, přičemž soustava byla buzena periodickou sinusovou křivkou. Jak si lze povšimnout, frekvence sinusové křivky a kmitání znázorněného na obrázku 8 se shodují. Tento fakt potvrzuje slova, která byla napsána v úvodu podkapitoly Kmitání buzené.



Obrázek 8: Průběh sinusové křivky

U vynuceného kmitání neboli buzeného kmitání lze zabránit anebo předcházet vzniku nežádoucího kmitání dodržením několika zásad, jako jsou například: [6]

- použití torzního tlumiče nebo frézovacích hlav s nestejnou roztečí nožů při frézování,
- zmenšení hloubky třísky, čímž se sníží amplitudy vynuceného kmitání, avšak za cenu sníženého výkonu,
- zvýšení nebo snížení otáček hlavního vřetena, díky čemuž se lze vyhnout rezonanci a pracovat v oblasti, ve které účinek kmitání není tak výrazný,
- uložení obráběcího stroje na zvláštní základ, čímž se zabrání přenosu chvění z okolí na stroj,
- dokonalého dynamického vyvážení rotujících součástí (hřídele, ozubená kola, spojky, nástroje, rotory elektromotorů apod.) na speciálních strojích pro vyvažování.

3.2.3 Kmitání samobuzené

Ke vzniku kmitání samobuzeného stačí první impulz, ten může například vznikat při nájezdu nože na obrobek, čímž dojde k vychýlení ze stabilní polohy a těleso začne kmitat. Dále již kmitající těleso nepotřebuje impulzy ze vnějšku, nýbrž energie, kterou těleso ztrácí tlumením, bude nahrazována energií přiváděnou do kmitající soustavy. Důležité je si uvědomit, že frekvence kmitání může být blízká nějaké vlastní frekvenci tělesa ze soustavy. Vlastní frekvence jsou pro tělesa zvláště nebezpečné. [7]

Kmitání samobuzené v obráběcích strojích je často doprovázeno hlukem a vyskytuje se jen v určitém rozmezí řezných podmínek, avšak není závislé na otáčkách stroje. [7]

Existují dvě nejznámější teorie zabývající se vznikem kmitání samobuzeného během řezného procesu:

- Reprodukční princip
- Princip polohové vazby

Reprodukční princip je založen na myšlence, že obráběná plocha je zvlněná, přičemž se mění průřez třísky, a to zapříčiňuje měnící se řezné síly, které v konečném důsledku vyvolají samobuzené kmitání. [8]

Princip polohové vazby je založen na předpokladu, že řezné síly nesouhlasí se směrem maximální vazby. [6]

3.2.4 Trhavé pohyby

Trhavé pohyby neboli Slip Sticks vznikají v „*důsledku nelineárního průběhu závislosti mezi rychlostí posunované části po vedení a třecí silou ve vedení*“. [8] Trhavým pohybům se lze vyhnout velice jednoduše, a to buď dobrým mazáním, anebo pomocí využití kluzných materiálů, a částečně také zvýšením tuhosti konstrukce posuvného pohybu.

3.2.5 Kompenzace vibrací

Snaha je co možná nejvíce kompenzovat vzniklé vibrace a pokud možno jim i předcházet. Existují dva druhy přístupu ke kompenzaci vibrací, a to pasivní kompenzace vibrací a aktivní kompenzace vibrací.

Pasivní kompenzace vibrací

Podstatou téhle metody je přenést nežádoucí vibrace, které působí na primární soustavu na sekundární soustavu, přičemž nám pomáhá tzv. hltič, který je připojen k primární soustavě pomocí prvku tuhosti a prvku tlumení. Nutností u tohoto typu řešení je správné nastavení

parametrů hltiče (tuhost, tlumení, připojené členy). Při správném nastavení hltiče dochází k nahrazení původního rezonančního vrcholu dvojicí nových vrcholů, které nejsou již pro soustavu nebezpečné. [9]

Dalšími možnými způsoby pasivní kompenzace jsou: [9]

- vhodná volba nástroje a jeho geometrie,
- použití nerovnoměrné rozteče zubů a nestejného stoupání,
- použití různých druhů materiálu, z kterých je zhotoven rám obráběcího stroje.

Aktivní kompenzace vibrací

Do systému jsou přidávány takové prvky, které umožňují aktivní tlumení. Jsou to různé druhy tlumičů, jako například tlumiče aplikované v pohybových osách co nejbližše potencionálním zdrojům kmitání. Přidáním aktivního prvku do soustavy a jeho správným řízením docílíme toho, že hltič bude schopen pracovat v širším frekvenčním rozsahu. Aktivní prvek tedy do systému vnáší energii a může využívat jako zdroj vnější energie hydraulické kapaliny, tlakové plyny, elektrické energie a jiné zdroje. [9] [6]

Jak je uvedeno v práci Jana Dvořáka [9], nevýhodou je nebezpečí zesílení vysokých frekvencí systému vlivem zpětné vazby a obecně nebezpečí destabilizace systému.

3.3 Deformace stroje

Další z nežádoucích vlivů, které působí na obráběcí stroj a potažmo na obrobek, jsou různé deformační síly neboli deformace způsobené například řeznou silou. Míra odolnosti tělesa vůči deformacím je daná tzv. tuhostí. Tuhost udává míru změny délky pružného tělesa vzhledem k velikosti působící vnější síly při pružné deformaci:

$$k = \frac{\text{síla působící na těleso}}{\text{přetvoření}} = \frac{F}{\Delta l} = \frac{F}{\delta} \quad (6)$$

kde:

k...tuhost

F...působící síla

$\Delta l = \delta$...přetvoření/deformace

Vzorec (6) je platný pro konstantní zatížení, z čehož vyplývá, že na těleso musí působit statická síla, aby vzorec platil. Avšak z předešlé kapitoly je zřejmé, že mohou při obrábění vznikat vibrace, a proto je potřeba počítat i tzv. dynamickou tuhost, jejíž vzorec při kmitajícím zatížení je následující:

$$k = \frac{\text{amplituda zatížení}}{\text{amplituda deformace}} \quad (7)$$

Je tedy zřejmé, že obráběcí stroj mohou ovlivňovat dva druhy tuhostí, dynamická tuhost a statická tuhost, přičemž je potřeba si uvědomit, že součásti se mohou deformovat jednotlivě, jako například ložiska, šrouby atd., anebo celkově jako soustavy. [10] [11]

Obecně lze říci, že dynamické zatěžování součástky má daleko horší dopad na životnost součástky než statické zatěžování součástky a je snaha nebo přímo nutnost, aby po celou dobu zatěžování součástky a její následném odlehčení platil Hookův zákon, tedy aby se těleso po

celou dobu zatěžování nedostalo za mez úměrnosti pružné deformace (materiál, z kterého je těleso vyrobeno, je schopno se vrátit do původního stavu).

0A – pružná deformace, platí Hookův zákon

σ_u – **mez úměrnosti** |

AB – dopružování

σ_d – **mez pružnosti**

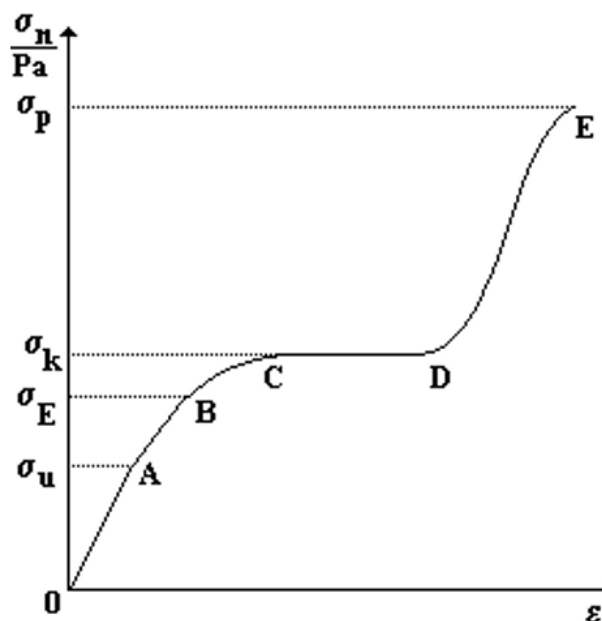
BE – plastická deformace

σ_K – **mez kluzu**

CD – tečení materiálu

DE – zpevnění materiálu

σ_p – **mez pevnosti**



Obrázek 9: Hookův zákon [12]

Nyní budou rozebrány jednotlivé druhy zatížení, avšak s větším důrazem na statickou tuhost, protože dynamická tuhost se z velké části týká vibrací a kmitání, které bylo rozebráno v podkapitole výše.

3.3.1 Statická tuhost

Aby se dalo uvažovat o statické tuhosti, musí na těleso nebo na soustavu těles působit konstantní síla. Ukazatel statické odolnosti materiálu udává, jaká bude deformace tělesa při určité statické síle neboli jak je schopno těleso odolávat konstantnímu zatížení. Lze rozlišovat čtyři druhy zatížení, které mohou vznikat rozličnými způsoby, jakými jsou vlivy silových složek hmotnosti a síly vznikající v technologickém procesu. Těmito zatíženími jsou myšleny:

- Tah
- Tlak
- Krut
- Ohyb

Jak uvádí Borský [13] deformace smykové jsou zpravidla zanedbatelné a rovněž deformace v tahu a tlaku jsou s ohledem na tvar a rozměry součásti i charakter vnějšího zatížení nepatrné, a proto se tuhost v tlaku a tahu až na některé zvláštní případy nevyšetřují.

Jak již bylo uvedeno, lze určovat tuhost pro dílčí součástky zvlášť, pod čím si lze například představit vřeteno bez vlivu deformace ložisek, skříně vřeteníku a spojení vřeteníku s ložem nebo jako celkovou soustavu. Tuhost soustavy je dána součtem tuhostí jednotlivých součástí. Je důležité si uvědomit, co člověk chce měřit. Jestliže bude měřeno již zmíněné vřeteno jako soustava, nelze zjistit tuhost samotného vřeteníku a tím pádem nelze optimalizovat tvar, rozměr a materiál vřeteníku, avšak lze tímto měřením zjistit, jaká je celková tuhost soustavy při daném pracovním a hmotnostním zatížení. [13]

Dále lze měřit absolutní a relativní tuhosti. Absolutní tuhost je vztažena k základně stroje a lze se o ní bavit, když řešíme odolnost stroje vůči vnějšímu zatížení. Avšak relativní tuhost je také velice důležitá, zejména se s ní lze setkat mezi obrobkem a nástrojem. [13]

V některých případech je výhodné zavést poddajnost, která říká, jakou deformaci vyvolá jednotkové zatížení. Jde o obrácenou hodnotu tuhosti.

3.3.2 Dynamická tuhost

O dynamické tuhosti se bavíme tehdy, působí-li na těleso časově proměnná síla, tzv. dynamické zatížení. Dynamická zatížení jsou při provozu stroje daleko běžnější než statická zatížení, a to z toho důvodu, že ve stroji vznikají vibrace, jejichž příčinou může být nespočet faktorů. Jako jedna z mnoha příčin vzniklých vibrací může být nájezd nástroje na obrobek. Více o vibracích a kmitáních pojednává kapitola 3.2, kde jsou rovněž uvedeny možné způsoby kompenzace vibrací.

3.4 Shrnutí poznatků z možných deformací obráběcího stroje

Z výše zmíněných důvodů vzniká snaha o zabránění působení negativních vlivů na obráběcí stroj. Ačkoli nelze zabránit všem negativním vlivům, dá se jim aspoň předcházet. Některým negativním vlivům se dá předcházet lépe, některým hůře. Jak již bylo zmiňováno mnohokrát, existují vnější a vnitřní původci negativních vlivů. Vnější původci se ve většině případů dá předcházet daleko lépe, přičemž i zákazník sám dokáže těmto vlivům předcházet tím, že bude dodržovat několik zásad a bude schopen analyzovat prostor, do kterého chce zasadit koupený obráběcí stroj. S vnitřními původci je to komplikovanější a samotný zákazník s těmito vlivy sám moc nezmuže. Odstraňovat tyto vlivy a předcházet jim je věcí dodavatele, jako například správně zvolená tuhost rámu, supportu, určení správných ložisek a mnoho dalšího.

V současnosti je kladen stále větší důraz na proměřování jednotlivých veličin, které působí na stroje, a to z jednoho prozaického důvodu. Koupě a zabudování senzorů do obráběcího stroje je dražší než koupě stroje bez senzorů, ale z dlouhodobějšího hlediska ušetří stroj se senzory mnoho financí. Nyní si představme situaci, kde stroj bez senzorů, který vyrábí denně tisíce kusů, najednou přestane obrábět na požadovanou přesnost a začnou vznikat zmetky. Podle toho, jak je obsluha stroje svědomitá v proměřování, tak si všimne, že stroj již nepracuje, jak má, až za nějakou kratší či delší dobu. Obvyklý postup je ten, že je k danému stroji zavolán technik. Jelikož ale neví, co se strojem vlastně je a kde může být příčina jeho nefunkčnosti, začne proměřovat jednotlivé fyzikální veličiny, které by na způsobenou chybu mohly mít vliv. Avšak než najde zdroj chyby, zabere mu to mnoho času. Jelikož se stroj rozbil v jednom okamžiku, tak práce, kterou stroj vykonával, nemohla být dostatečně rychle přesunuta jinam a každým okamžikem, kdy je stroj mimo provoz, uniká možný zisk.

Ze zmíněného příkladu je zřejmé, že kdyby stroj měl zabudované senzory, dalo by se daleko dříve zjistit, že stroj bude muset být odstaven a tím pádem by se dala naplánovat jeho odstávka, přičemž by i technik, který by poté na místo přišel, dopředu věděl, kde by se chyba stroje mohla nalézat a ušetřilo by se mnoho finančních prostředků i času.

Stroje se senzory tvoří nedílnou součástí automatizovaných podniků z již zmíněných důvodů. Avšak je potřeba pro správné měření vybrat správné senzory, které nebudou zbytečně drahé, ale budou splňovat požadované parametry pro přesnost měření a budou zasazeny na místo, které je pro ně určené. Další kapitola se bude zabývat přehledem senzorů, které snímají fyzikální veličiny, které jsou v této práci rozebírány.

4 SENZORY

Senzory jsou důležitou součástí pro zjišťování diagnostiky. Bez nich nelze přesně určit, v jakém stavu se zrovna systém nachází, a předvídat budoucí chování systému. V podkapitolách budou zmíněny senzory na měření čtyř fyzikálních veličin, ale samozřejmě je celá další řada fyzikálních veličin, které lze v systémech měřit.

4.1 Základní vlastnosti senzorů

Senzor je část měřicího řetězce, která je v přímém styku s měřeným prostředím, přičemž převádí fyzikální veličinu na hodnotu, kterou jsme schopni zpracovat. Citlivá část senzoru se nazývá čidlo. Na obrázku 10 lze vidět schéma senzoru.

- Čidlo
 - Citlivá část
- Zpracování signálu
 - Zesílení
 - Filtrace
- Digitalizace
- Digitální zpracování signálu
 - Filtrace
 - Statistika
 - Přepočet na fyzikální veličinu
- Výstupní rozhraní
 - Analogové
 - Digitální
 - kombinované



Obrázek 10: Schéma senzory [14]

Senzory mohou mít mechanický nebo elektrický výstup. Mechanický výstup senzorů není příliš vhodný, protože se musí převádět na elektrický, tudíž jsou v současnosti upřednostňovány senzory s elektrickým výstupem. Senzory mají mnoho charakteristik, např. statická převodní charakteristika, která definuje matematický vztah mezi výstupní veličinou senzoru a měřenou veličinou v ustáleném stavu, dále citlivost senzoru, práh citlivosti anebo dynamický rozsah, který udává interval hodnot snímané fyzikální veličiny od prahu citlivosti do maximální hodnoty měřené veličiny. Důležitými charakteristikami jsou bez pochyby i rozlišitelnost a reprodukovatelnost. Rozlišitelnost je nejmenší změna snímané veličiny odpovídající absolutní nebo relativní chybě senzoru. Reprodukovatelnost je dána odchylkou naměřených hodnot při krátkodobém časovém sledu neměnné vstupní veličiny a neměnných rušivých vlivů okolí. [15]

Další vlastností senzorů jsou chyby v měření, které jsou způsobeny posunem jmenovité lineární charakteristiky, což má vliv především na posun nulové nebo počáteční hodnoty výstupní veličiny. Pod těmito chybami si lze například představit kvantovací chybu, nevhodné rozlišení A/D převodníku, hysterezi, která způsobuje necitlivost měření v určitém rozsahu. [15]

Na možné chyby v měření je potřeba pamatovat při nastavování a vybírání senzoru. Je potřeba si na ně dávat pozor, protože je nelze opravit softwarově. Správná instalace senzorů je

nejdůležitější předpoklad pro korektní měření. Vše, co bylo zmíněno výše, platí i pro senzory, které jsou rozebrány v podkapitolách dále. [15]

4.2 Senzory teploty

Převod tepla na elektrický signál lze provést mnoha způsoby, přičemž se využívá parazitních jevů, které nejsou v běžných aplikacích žádoucí. Senzory teplot se dělí do dvou hlavních kategorií, dotykové senzory teploty (odporové, termoelektrické, speciální atd.) a bezdotykové senzory teploty (pyrometry, infratermočlánky atd.). Nyní budou rozebrány obě kategorie snímání teplot. [16]

4.2.1 Dotykové senzory teploty

Jak již je z názvu patrné, měření tepla probíhá přestupem z jednoho tělesa (senzoru) na druhé, přičemž jsou obě tělesa v přímém kontaktu spolu. Výhoda využití dotykových senzorů teploty je hlavně v tom, že se zamezuje vlivu rušení signálu z okolního prostředí a že lze využít senzor i uvnitř těles. Dotykovým snímáním teplot se dosahuje přesnějších měření, ale pouze pro malý rozsah měřených teplot, přičemž odezva senzoru je delší než u bezdotykových senzorů. Další nevýhoda dotykových senzorů pramení z potřeby důkladné a precizní instalace senzorů, při které mohou vznikat chyby, jako například špatné ošetření měřeného povrchu. [17] [16]

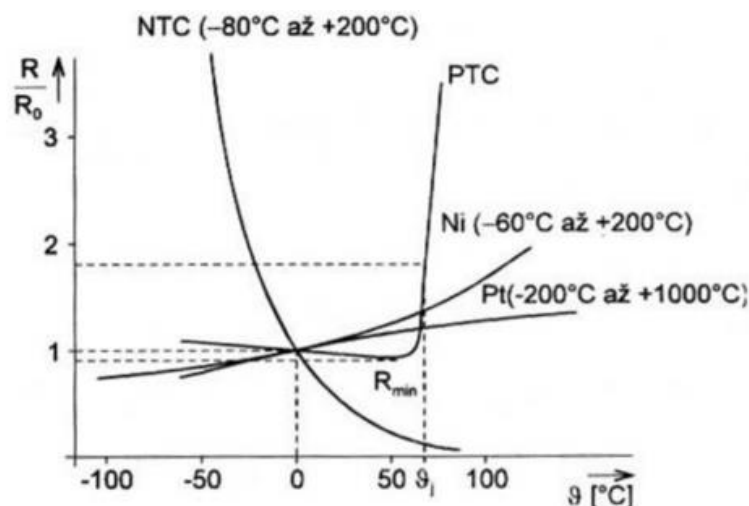
Základní dělení dotykových senzorů teplot je určeno fyzikálními principy, na kterých senzory pracují. Existují odporové senzory, které využívají závislosti odporu kovu na teplotě, nebo termoelektrické senzory teplot, které využívají Seebeckův jev, a mnoho dalších dotykových senzorů, avšak odporové a termoelektrické senzory jsou nejpoužívanější, z tohoto důvodu budou rozebrány pouze tyto dva druhy. [17]

Odporové senzory teplot

Jak již bylo psáno, odporové senzory teplot fungují na proměnlivém odporu při různých teplotách. Výhody odporových vodičů jsou zejména v dobré linearitě, vyšším rozsahu měřených teplot, přičemž nevýhody jsou v malém teplotním součiniteli (malá citlivost) a v pomalé časové odezvě na změnu teploty. Hlavní dva zástupci pro odporové kovové senzory teplot jsou platinové odporové senzory teplot, které mají rozsah měřených teplot od $-200\text{ }^{\circ}\text{C}$ do $850\text{ }^{\circ}\text{C}$ a využívají se například pro měření teploty páry nebo jako sondy pro měření teploty pečícího se pečiva v pekárnách, a niklové odporové senzory teplot, které mají menší rozsah měřených teplot než platinové senzory, a to v rozmezí $-200\text{ }^{\circ}\text{C}$ až $200\text{ }^{\circ}\text{C}$, ale mají výhodu v nižší ceně, vyšší citlivosti a rychlé časové odezvě. Dále také existují měděné odporové senzory teplot a senzory teplot ze speciálních slitin. [17]

Nelze však vynechat odporové polovodičové senzory, jež se vyrábí podobně jako integrované obvody, přičemž výhoda s tím spjatá je v jejich jednoduché výrobě, protože senzor a elektronika jsou v jednom pouzdru a jsou vyráběny jako Smart senzory (chytré senzory, které dokáží predikovat chyby, diagnostikovat vlastní činnost atd.). Na druhou stranu mají malý teplotní rozsah, mezi $-200\text{ }^{\circ}\text{C}$ až $150\text{ }^{\circ}\text{C}$, a velmi nelineární charakteristiky. Dělí se na pozistory (PTC), u nichž s rostoucí teplotou odpor roste, které se využívají jako dvoustavové senzory kvůli rychlému nárůstu teploty v závislosti na odporu, a negistory (NTC), u nichž s rostoucí teplotou odpor klesá, které se kvůli malému rozsahu teplot využívají například pro snímání venkovních teplot. [17]

Závislosti odporu na teplotě u výše zmíněných odporových senzorů teplot lze vidět na obrázku 11, kde R_0 je odpor daného senzory při $0\text{ }^{\circ}\text{C}$ a R je momentální odpor.



Obrázek 11: Charakteristika odporových senzorů [17]

Termoelektrické senzory

Termoelektrické senzory jsou založeny na tzv. Seebeckovu jevu, tj. na jevu přeměny tepelné energie na elektrickou spojením dvou vodičů z různých materiálů v jednom bodě, přičemž na volných koncích vzniká elektrické napětí. Jde tedy o aktivní senzory, které samy generují napětí. Je nutná kompenzace studeného konce. Výhody termoelektrických senzorů spočívají v lineární charakteristice a velkém teplotním rozsahu. Mezi nevýhody se řadí malá citlivost a nutnost správného zapojení pro eliminaci okolní teploty. Za výhodu se dá označit to, že lze měřit dvě různá místa mezi sebou, tedy rozdíl teplot mezi materiálem A a materiálem B, jako například materiál a vzduch. Jde tedy o relativní měření. [17]

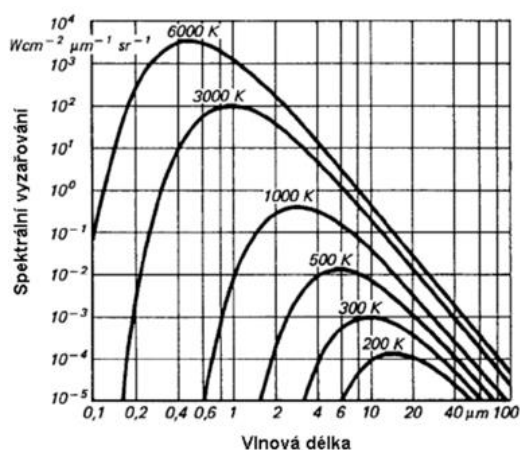
4.2.2 Bezdotykové senzory teplot

Měřené těleso a snímač nejsou v přímém kontaktu spolu, přičemž bezdotykové senzory využívají pro stanovení teploty objektu fyzikálního principu, při kterém je zachycováno infračervené záření, které je vyzařováno měřeným objektem. Toto záření je poté převedeno na elektricky měřitelnou veličinu, která je následně zpracována obdobným způsobem jako u dotykových senzorů. Bezdotykové senzory mají výhodu v tom, že jimi lze bezpečně měřit vysoké teploty, pohyblivé objekty nebo i více objektů najednou jedním senzorem a provádět rychlé měření. Nevýhody jsou zejména v tom, že objekt musí být pro senzor viditelný. Na přesnost měření má velký vliv okolí (např. nečistoty, prostupnost prostředí). Bezdotykové senzory se dělí do tří hlavních kategorií, jímž jsou tepelné senzory infračerveného záření, pyroelektrické senzory, které fungují na změně spontánní polarizace při změně teploty, přičemž na povrchu materiálu, kam dopadá infračervené záření, vzniká elektrický náboj, který se dále měří, a kvantové senzory infračerveného záření, které využívají princip přímé interakce dopadajících fotonů na strukturu senzoru. [18]

Tepelné senzory infračerveného záření

Tepelné senzory infračerveného záření neboli termoelektrické senzory záření, fungují na takovém principu, kdy „při absorpci fotonů dochází k oteplení citlivé části senzoru a pohlcená energie se vyhodnocuje nepřímo přes dotykové senzory teploty“. [15] Z uvedené citace je zřejmé, že bezdotykové tepelné infračervené senzory se chovají obdobně jako dotykové senzory

s tím rozdílem, že teplo je pohlcováno světlocitlivou plochou, která se poté zahřívá. Mezi zástupce se řadí například bolometry, u kterých se mění elektrický odpor s teplotou. Absorpční vrstva je ohřívána dopadajícím světlem a následně ohřívá vodivý povrch nosníku, kde je na spodní straně napařena odporová dráha. Měřitelné teploty jsou v rozmezí -200 °C až 1500 °C. Obecně jsou senzory závislé na vlnových délkách infračerveného záření. Na obrázku 12 je závislost vlnových délek na spektrálním vyzařování. Tepelné senzory infračerveného záření se využívají zejména pro měření změny teploty tělesa nebo při detekci osob v zabezpečovacích systémech. [18]



Obrázek 12: Závislost vlnových délek na spektrálním vyzařování [18]

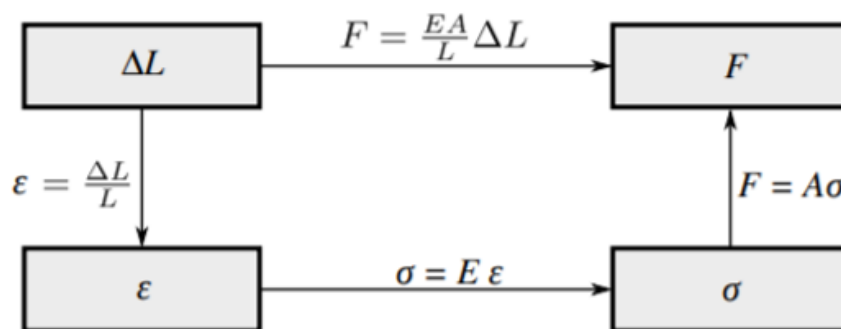
4.3 Senzory síly

Existují odporové senzory síly neboli tenzometry, kde je síla vyhodnocována na základě výstupního napětí a odporů vodičů, dále také piezoelektrické, magnetoelastické a kapacitní senzory, u nichž je síla vyhodnocována na rozdíl od tenzometrů přímo z deformace. Lze říci, že princip měření síly je založen na deformaci materiálů anebo samotných senzorů, z nichž je následně dopočítána síla pomocí fyzikálních vztahů. Jelikož každý druh senzorů vyhodnocuje působící sílu trochu odlišným způsobem, budou nyní u vybraných druhů senzorů představeny charakteristiky a základní principy, na kterých senzory pracují.

4.3.1 Odporové tenzometry

U odporových tenzometrů se využívá jevu, který již objevil Wheatstone v roce 1843. Tento jev se zakládá na tom, že změna odporu je závislá na deformaci neboli na změně parametru délky vodiče ΔL . Avšak zajímavější je údaj o relativní změně odporu $\Delta R/R$, kde je R odpor tenzometru v nenamáhaném stavu. Platí-li rovnice (8), kde ε je deformace a K je součinitel deformační citlivosti, lze pomocí vztahů, které jsou zobrazeny na obrázku 13 dopočítat sílu, která na materiál působí. Písmeno A , které lze na obrázku 13 vidět, znázorňuje plochu. Důležité je si ale uvědomit, že vztahy platí pouze v lineární oblasti Hookova zákona a v této oblasti pracují i tenzometry. [19] [20]

$$\frac{\Delta R}{R} = K \cdot \frac{\Delta L}{L} = K \cdot \varepsilon \quad (8)$$



Obrázek 13: Základní veličiny a rovnice pro rovnoměrně tažený prut [21]

Odporové tenzometry se nejčastěji zapojují do plného mostu (Wheatstonova mostu). Toto zapojení poskytuje výhodu v kompenzaci nežádoucích tepelných vlivů, chyb přívodů, anuluje činitele nelinearity a má čtyřnásobnou citlivost. Avšak změna odporu je často velmi malá, v řádech μm , proto se někdy do zapojení přidává zesilovač, který výstupní napětí zesílí.

Odporové tenzometry se skládají z kovových, polovodičových materiálů. Kovové tenzometry mají K (součinitel deformační citlivosti) konstantní, což znamená lineární charakteristiku. Typy mohou být drátkové (volné/lepené), foliové (vyráběny fotolitograficky) a vrstevné (vakuově napařované/naprašované). Polovodičové tenzometry nemají lineární charakteristiku, existují ve variantě P na Si, které mají kladnou charakteristiku a z toho důvodu jsou vhodné pro tah, a ve variantě N na Si, které mají zápornou charakteristiku a jsou vhodné pro tlak.

Tenzometry se nehodí pro dynamicky namáhané systémy a využívají se zejména tam, kde na těleso působí konstantní síla (váha).

4.3.2 Piezoelektrické senzory síly

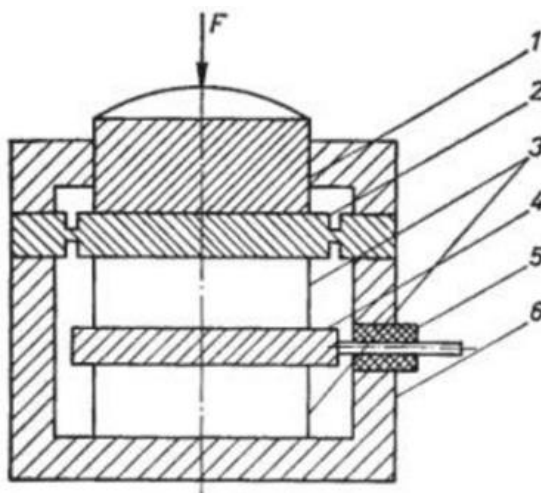
Využívá se tzv. piezoelektrického jevu, kdy při působení mechanických deformací dochází u některých druhů krystalů ke vzniku elektrického náboje podle rovnice (9), kde k_p je piezoelektrická konstanta.

$$Q = k_p \cdot F \quad (9)$$

Protože destička, na kterou působí síla a která má piezoelektrické vlastnosti, vytváří kondenzátor o kapacitě C , není již problém dopočítat vzniklé napětí podle rovnice (10).

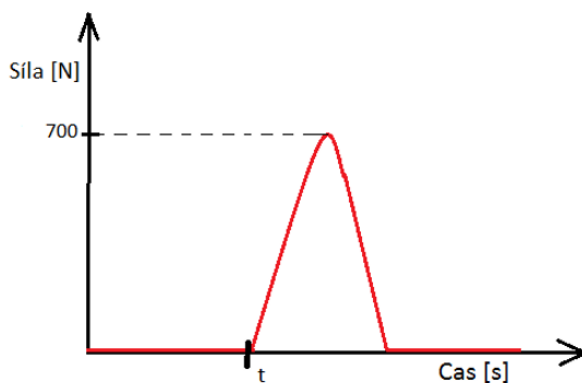
$$U = \frac{Q}{C} = k_u \cdot F \quad (10)$$

Samotný princip piezoelektrického senzory je pěkně vysvětlen ve zdroji [20] a nyní si ho dovolím doslovně citovat, přičemž se citace týká obrázku 14. „Piezoelektrický snímač obsahuje dva piezoelektrické krystaly, 3 orientované tak, aby se náboje sčítaly při působení síly na část 1 a membránu 2. Výstupní signál senzory je vyváděn z elektrody 4 izolační průchodkou 5 tak, aby se náboje sčítaly při působení síly na přítlačný článek 1 prostřednictvím membrány 2. Celý systém je uzavřen v pouzdru 6.“



Obrázek 14: Princip piezoelektrického senzory [20]

Piezoelektrické senzory se využívají zejména pro dynamická zatížení. Jejich hlavní výhoda pramení z toho, že lépe odolávají skokovému zatížení nad mezi kluzu než tenzometrické senzory. Lze jimi v různých konstrukčních obměnách měřit tlak i tah, ale důležité si je uvědomit, že zaznamenávají pouze přechodové stavy. Obrázek 15 vyobrazuje možný průběh změřené síly při konstantním zatížení 700 [N] od času t .



Obrázek 15: Vyobrazení průběhu zatížení zachycené piezometrem

4.4 Senzory vibrací

Snímat vibrace lze pomocí akcelerometru, tj. pomocí snímání zrychlení, ale také pomocí gyroskopu. Tato podkapitola bude zaměřena na jednotlivé druhy snímání vibrací ať už pomocí akcelerometru, nebo gyroskopu.

4.4.1 Akcelerometry

Všechny akcelerometry musí být provozovány v pásmu mimo svou vlastní frekvenci. Jestliže akcelerometr dosáhne své vlastní frekvence, může to vést k jeho zničení a měřené výsledky jsou chybné. Lze měřit zrychlení v jedné, dvou nebo třech osách, přičemž výsledky nemusí být pouze v ms^{-2} , ale také v tíhovém zrychlení g . Jak již bylo zmiňováno, akcelerometry lze využít nejen pro měření vibrací, ale také pro alarmy vozidel, joysticky, herní ovladače, mobilní telefony a pro mnoho dalších aplikací.

Základní princip akcelerometrů spočívá v přeměně zrychlení na měřitelný elektrický signál, který je poté převáděn na jednotky, které jsou ve výše zmíněném odstavci. V zásadě existují hlavní kategorie senzorů zrychlení, kterými jsou [22]:

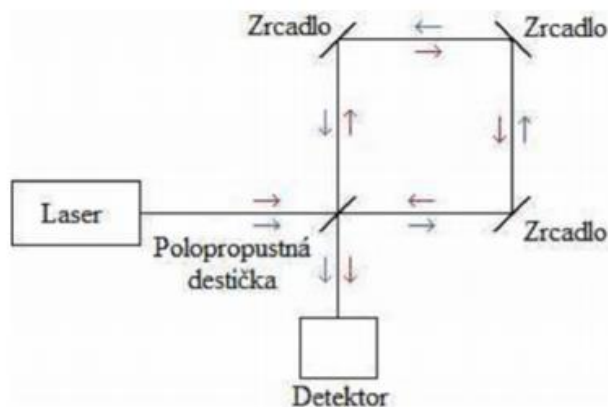
- **Piezoelektrické akcelerometry**
Je využit piezoelektrický krystal, přičemž je vytvořen náboj úměrný působící síle, která poté při zrychlení působí na každý objekt.
- **Piezorezistivní akcelerometry**
Využívá mikro křemíkovou struktura fungující na principu změny odporu při zrychlení.
- **Kapacitní akcelerometry**
Princip podobný jako u piezorezistivních akcelerometrů s tím rozdílem, že při změně zrychlení se mění kapacita.
- **Tepelné akcelerometry**
„Využívají rozdílného přenosu tepla v plynech na základě pohybujícího se topného tělíska.“ [19]
- **MEMS senzory** (Micro Electro Mechanical Systems)
Měří s přesností do 5 % a výstup mohou mít buď analogový, nebo digitální přes protokoly SPI, I2C.

4.4.2 Gyroskopy

Gyroskopy jsou senzory úhlové rychlosti anebo zrychlení. Fungují na třech základních principech:

- Mechanické gyroskopy
- Optické gyroskopy
- MEMS gyroskopy

Optické gyroskopy neboli interferometry jsou velmi přesné přístroje, jejichž princip je založen na interferenci světla. Jednou z možností je využití Sagnacova jevu, který je vyobrazen na obrázku 16. Světelný paprsek dopadá na polopropustnou destičku, kde se dělí na dva paprsky. Jeden paprsek putuje skrz soustavu zrcadel po směru hodinových ručiček, druhý proti směru hodinových ručiček. Nakonec se oba paprsky setkají a dopadají na detektor. Jestliže těleso bude v klidu, paprsky budou spolu interferovat se stejnou fází. Pakliže se těleso začne otáčet, paprsky budou interferovat s rozdílnou fází. Paprsek, který jde proti směru otáčení tělesa, bude zpožděn. Z těchto skutečností lze vypočítat fázový posun anebo zrychlení. [23]



Obrázek 16: Sagnacův interferometr [23]

MEMS gyroskopy dosahují pouze krátkodobé přesnosti, přičemž jejich princip spočívá ve vyhodnocování Coriolisova zrychlení. Princip mechanických gyroskopů spočívá v prstencích a rotujícím středu. Mechanické gyroskopy se využívají zejména pro určování azimutu, navádění a řízení raket nebo GPS systémy.

4.5 Senzory napětí a proudu

Měřit napětí anebo proud je důležité k celkové diagnostice systému. Lze měřit napětí a proud vstupující do systému, kde nám mohou změřené veličiny pomoci detekovat vysoké nebo naopak nízké hodnoty napájení. Vysoké hodnoty napájení mohou mít pro stroj destruktivní vliv, a proto je lepší stroj odpojit dříve, než bude zničen.

Dále lze měřit napětí a deformace vně stroje. Tento druh informací může mít dvě základní vypovídající hodnoty. První vypovídající hodnota souvisí s měřením napájení, protože lze vně stroje sledovat hladiny proudu a napětí, přičemž je možné opět diagnostikovat, zda se hladiny drží v mezích, které jsou pro dané elektrické zařízení vhodné.

Druhá vypovídající hodnota se váže s poruchami systému. Jestliže systém přestane pracovat a všechny senzory jsou neporušené a hlásí správné hodnoty pro danou aplikaci, například že je vhodná teplota stroje, a proudový senzor hlásí, že jím neprochází proud, lze rychle stanovit, v jaké části stroje se závada nachází, a tím i diagnostikovat v jaké kabeláži bude nejspíše problém.

Kromě diagnostiky je využíváno měření napětí i v samotných senzorech a v regulačních obvodech.

Napětí a proud lze měřit pomocí voltmetru, popřípadě pomocí ampérmetru. Princip měření obou veličin je podobný, avšak připojení zařízení do elektrického obvodu je jiné. Voltmetr se musí připojovat paralelně, zatímco ampérmetr sériově.

V nynější době existují analogové a stále využívanější číslicové měřicí přístroje. Analogové měřicí přístroje nejčastěji existují v konfiguraci elektromagnetické, elektrodynamické anebo magnetoelektrické. Všechny zmíněné analogové typy mají jako základ cívku, která je umístěna v magnetickém poli a prochází jí proud. Proud procházející cívkou vytváří magnetické pole, které může například zmagnetovat pohyblivé segmenty spojené s analogovým ukazatelem hodnot u elektromagnetického principu. Ukazatel poté přímo ukazuje na stupnici číslo, kterému odpovídá naměřený proud, napětí. Číselné měřicí zařízení, nejčastěji multimetry, pracují na podobném principu jako analogové měřicí zařízení, ale mají integrované vnitřní obvody s analogově číslicovým převodníkem a jsou přesnější než analogové měřicí přístroje. [24]

Je ale důležité si uvědomit, že výše zmíněné ampérmetry a voltmetry jsou v aplikaci kontinuálního měření nepoužitelné. Pro účely dlouhodobějšího měření jsou důležité integrované senzory, které budou měřit napětí, popřípadě proud nepřetržitě v delším časovém úseku, aniž by s nimi bylo nějak manipulováno.

Integrované senzory také pracují na přeměně elektrického proudu na napětí a existují tři základní převody proudu na napětí: [25]

- Bočníky – měření úbytku na snímači
- Proudové transformátory
- Senzory s Hallovoú sondou – měření proudem generovaného magnetického pole Hallovoú senzorem

4.6 Shrnutí poznatků o senzorech

Připojení senzorů ke stroji je nedílnou součástí automatizovaného procesu výroby. Dokáží ušetřit čas i peníze, avšak je důležité vybrat ten správný senzor. Jak se lze dočíst v kapitole 4, existuje mnoho principů, podle kterých senzory pracují, přičemž každý z principů má jinou časovou odezvu a charakteristiku, nehledě na mechanické, elektrické a tepelné vlastnosti senzorů. Většinou však podle katalogů a povědomosti o měřené veličině lze nalézt ten správný senzor pro měření dané veličiny.

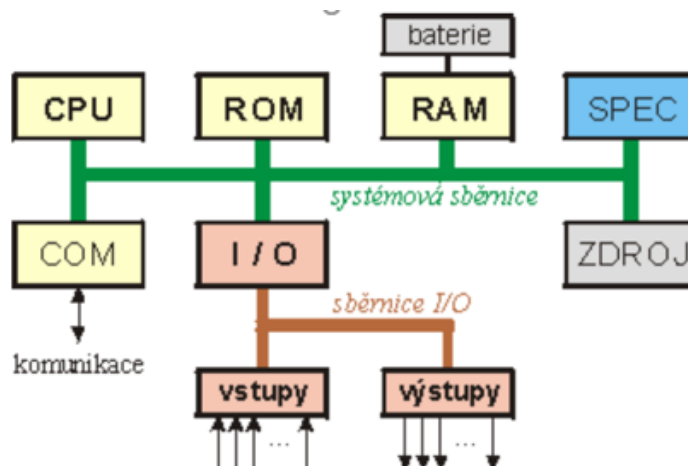
5 MOŽNOSTI ZPRACOVÁNÍ DAT SE SENZORŮ

Data real-time lze zpracovávat na různých hardwarech, a to zejména na PLC, dSPACE a platformách od National Instruments.

5.1 PLC

Programovatelné logické automaty (PLC) jsou hojně využívány v průmyslu pro řídicí aplikace, které se mohou od sebe lišit svoji technickou úrovní, která stoupá s množstvím investovaných financí. [26]

Obrázek 17 znázorňuje možné schéma PLC.



Obrázek 17: Bloková struktura programovatelného automatu [26]

Mezi hlavní bloky, které lze vidět na obrázku 17, se řadí: [26]

- I/O – Jednotka řízení vstupů a výstupů
- CPU – Centrální jednotka, která zpracovává informace
- RAM – Místo pro uložení řídicího programu, vstupních a výstupních hodnot a pomocných proměnných
- ROM – Systémové programy důležité pro činnost automatu

Program pro PLC je složen z jednotlivých podmínek neboli logických výrazů a ty poté z instrukcí. Avšak automaty různých výrobců používají poněkud odlišné programovací příkazy, přičemž struktura základních příkazů je zachována. Rozdíly nejsou nikterak velké, avšak pro programátora je značně nepříjemné, když přechází z jednoho PLC na druhý, který vyrábí jiná firma a má odlišnosti v instrukcích. Nicméně je nutné podotknout, že mnozí výrobci zároveň s PLC dodávají software, který má grafické rozhraní a umožňuje programovat pomocí symbolů logických operací, popřípadě propojováním funkčních bloků spojnicemi, které představují signály. [26]

5.2 dSPACE

Jedná se o zařízení od firmy MathWorks využívající jako software Matlab/Simulink/Stateflow a jejich podpůrné toolkity. Programování probíhá pomocí funkčních bloků, které jsou spojeny spojnicemi v podobě signálů.

Velká výhoda ve využívání dSPACE je právě jeho software, jelikož MathWorks je velká firma, která každého půl roku vydává aktualizace Matlabu, čímž je zaručena kvalita softwaru. Avšak nutno podotknout, že software nepatří zrovna mezi nejlevnější.

Další výhodou skýtá provázání s C code generator Simulink Coder™, což umožňuje kód, který byl vytvořen pomocí funkčních bloků, zkompileovat do jazyka C a poté jej nahrát například do procesoru.

5.3 Platformy National Instruments

Mezi hlavní platformy od National Instruments se řadí PXI, CompactRIO (využíván v diplomové práci), cDAQ a sbRIO.

Základní rozdíly mezi jednotlivými platformami od firmy National Instruments (dále jen NI) lze vidět na obrázku 18.

Vlastnosti	PXI	cRIO	cDAQ	sbRIO
I/O	1000s	10s-100s	10s-100s	10s
Velikost	Největší	Střední	Střední	Nejnižší
Cena	\$\$\$\$	\$\$\$	\$\$	\$
Výkon	Nejvyšší	Střední	Střední	Nejnižší
Vývojářské úsilí	++	+++	+	++++

Obrázek 18: Rozdíly mezi systémy National Instruments [27]

Jak lze vyčíst z obrázku 18, PXI umožňuje nejvyšší výkon, přičemž není zapotřebí velké vývojářské úsilí, což se samozřejmě odráží na konečné ceně. Velkou nevýhodou PXI jsou jeho větší rozměry.

Pro rychlejší běh programu a využití hardwaru v náročnějších podmínkách, jako například ve výrobní hale, kde musí běžet program nepřetržitě více hodin nebo dnů, je optimálnější využít CompactRIO. Avšak není úplně triviální věc napsat korektní program pro cRIO. CompactRIO má tu velkou výhodu, ale zároveň pro někoho nevýhodu, že je potřeba napsat tři odlišné programy, a to ve vrstvě FPGA, Real-Time (RT) a nakonec program pro Host-PC. Jednotlivé programy budou rozebrány dále v diplomové práci. Nevýhoda cRIO spočívá v tom, že pro jeho maximální rychlost je potřeba napsat programy pro tyto tři vrstvy, avšak je to i obrovská výhoda, protože právě díky vhodně napsaným programům v FPGA a RT lze dosáhnout vyšších výkonů než u jiných hardwarů od NI.

CompactDAQ je velice podobný cRIO, avšak vývoj aplikace je daleko jednodušší, není potřeba psát tři vrstvy aplikace, protože využívá DAQmx namísto Labview FPGA a RT. Dá se říci, že stačí napojit cDAQ do počítače a DAQmx už sám vede uživatele. Pro čtení dat, které není časově náročné, je naprosto postačující cDAQ. Avšak jestliže je potřeba řídit časově kritické aplikace, je lepší využít cRIO.

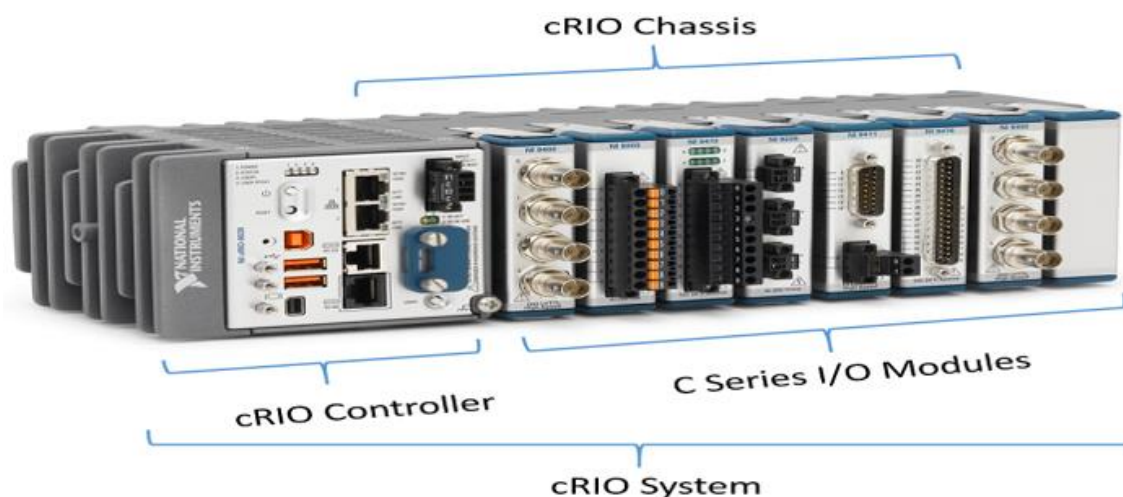
Single Board Rio je podobné cRIO, ale s jedním podstatným rozdílem, a to tím, že nemají chassis a I/O moduly, tudíž nejsou příliš vhodné pro testovací aplikace, ale spíše pro embedded aplikace.

Výběr správného hardwaru od NI vždy záleží na konkrétní aplikaci a objemu financí, které mají být investovány. Obecně lze říci, že zakoupením cRIO lze zabít všechny mouchy jednou ranou, protože je nejvíce univerzální, ale pro jednoduché aplikace až zbytečně složitý.

5.3.1 CompactRIO

Jak již bylo psáno, cRIO se hodí pro aplikace, které jsou časově kritické a mají běžet delší dobu ve zhoršených podmínkách. Navíc má cRIO výhodu ve své nepříliš velké velikosti. CompactRIO existuje v několika variantách, které se liší svojí velikostí, počtem zásuvných modulů i různými možnostmi komunikace s Host-PC. Obecně má cRIO dvoujádrový procesor a čtyři ledky, které lze zapínat a vypínat například pro ověření komunikace mezi cRIO a Host-PC.

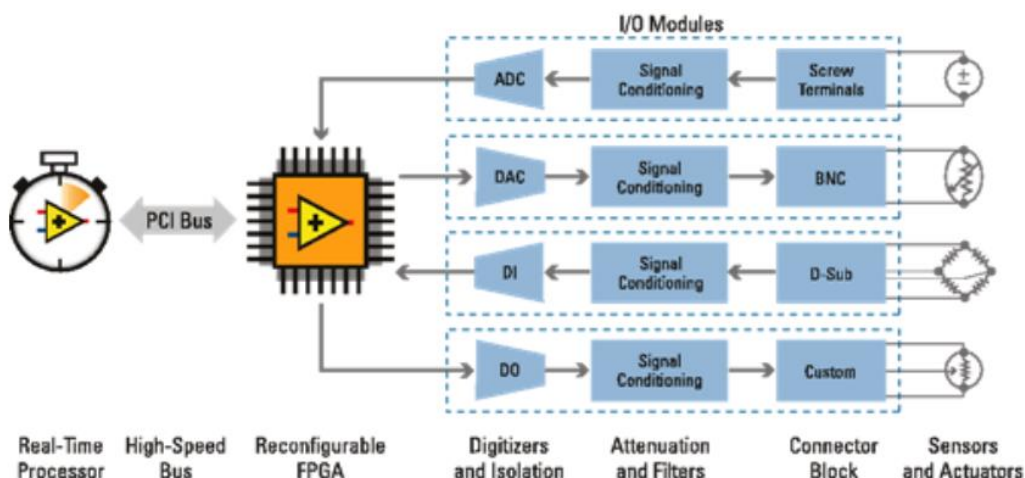
Na obrázku 19 lze vidět jeden z možných druhů cRIO.



Obrázek 19: CompactRIO [28]

CompactRIO systém se skládá z Chassis (FPGA), do kterých se vkládají zásuvné moduly C Series se vstupy a výstupy (analogové, digitální, pohybové, synchronizační), a cRIO Controlleru, na kterém běží RT procesor, se vstupy a výstupy, které se liší pro konkrétní verze cRIO. Na obrázku 19 lze vidět vstup pro ethernet, USB, sériovou komunikaci, SD kartu atd. Celá řada zásuvných modulů a různých vstupů a výstupů z cRIO Controlleru činí z cRIO systém mocný nástroj pro zpracování dat. Nutno také podotknout, že cRIO má i svůj vlastní úložný disk, kam lze naměřená data ukládat.

Vnitřní uspořádání cRIO lze vidět na obrázku 20.



Obrázek 20: Architektura cRIO [29]

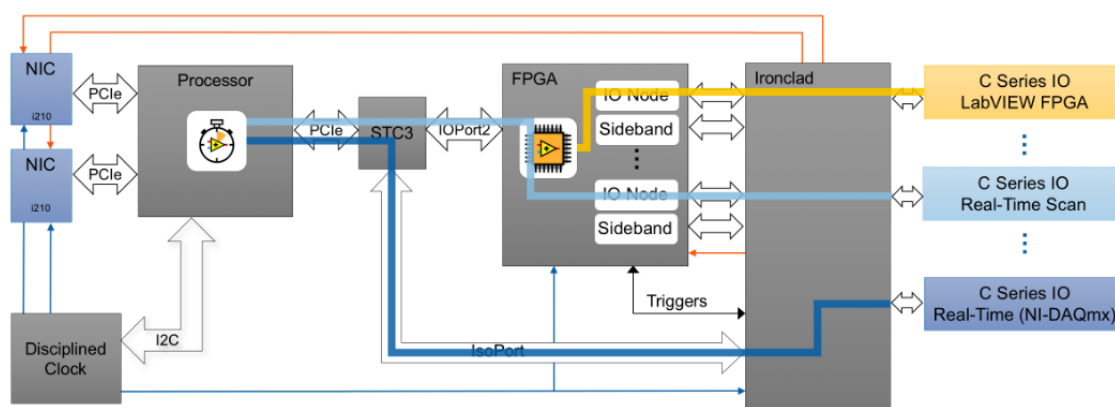
Jak si lze povšimnout, na pravé straně obrázku 20 se nacházejí senzory s aktuátory, které vysílají nebo přijímají signál z cRIO. Prostřední modře zbarvená část znázorňuje C Series

Moduly, které signál zpracují a předají ho rekonfigurovatelnému FPGA (programovatelnému hradlovému poli), které obsahuje různě propojitelné programovatelné logické obvody na rozdíl od ASIC (zákaznický integrovaných obvodů). Programování hradel se provádí například v jazyce VHDL, avšak tento jazyk je poměrně složitý a není jednoduše uchopitelný. Naštěstí NI vyvinul modul Labview FPGA, který značně usnadňuje programování hradlových polí.

Poslední část cRIO je Real-Time processor, který umožňuje zpracovávat data v reálném čase, a s možností si nastavit logické obvody podle vlastních představ se stává cRIO velice výkonným nástrojem.

CompactRIO umožňuje přístup k datům pomocí třech programovacích módů. Tyto módy lze vidět na obrázku 21. Pro uživatele je nejpohlednější NI-DAQmx mód, avšak není úplně optimální a je spíše doplňkem cRIO. Daleko zajímavější mód je Scan Engine. LabVIEW si samo dokáže přeskládat hradlové pole a umožní vyčíst data ze senzorů v Real-Time procesoru. Pro pomalejší aplikace je toto řešení neoptimálnější, protože není potřeba se zabývat hradlovými poli.

Aby byl využit plný potenciál, který skýtá cRIO, je potřeba využít modul FPGA. Při využití modulu FPGA si programátor sám nastavuje hradlová pole a umožňuje aplikaci rychlejší běh, přičemž nespolehá na automatické nastavení hradel, jak tomu bylo v módu Scan Engine.



Obrázek 21: Programovatelné moduly cRIO [29]

V následujících podkapitolách bude ukázána struktura výše zmíněná v rámci softwaru LabVIEW, který podporuje cRIO. Aby mohla být vytvořena taková struktura, jaká bude znázorněna, je potřeba mít stažené a nainstalované správné drivery a také moduly LabVIEW, kterými jsou:

- LabVIEW,
- LabVIEW Real-Time,
- LabVIEW FPGA,
- NI-RIO Driver.

5.3.2 LabVIEW pro Host-PC

Jak již bylo mnohokrát zmíněno, cRIO využívá jako svůj operační program LabVIEW. Prostředí LabVIEW se dá přirovnat k prostředí Matlab-Simulink. Program není vytvářen textově, nýbrž pomocí bloků, které se navzájem spojují, jak si lze povšimnout na obrázku 22.

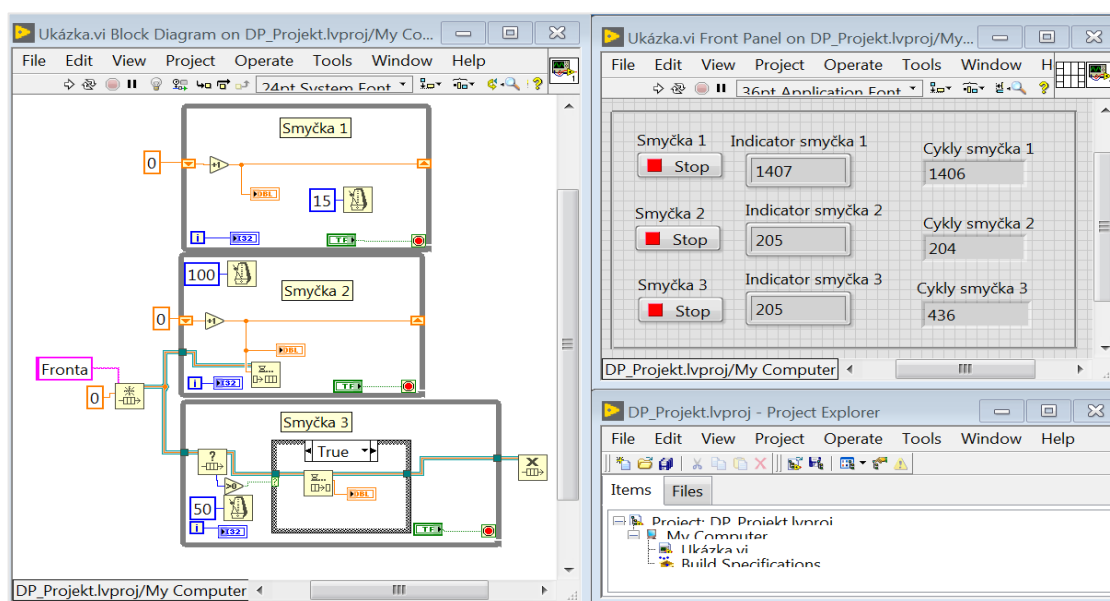
Program LabVIEW je tvořen dvěma okny, a to oknem Block Diagram, kam se píše vlastní kód, a oknem Front Panel, který slouží uživateli k ovládání a čtení údajů z Block Diagramu. Poslední

okno, které lze spatřit na obrázku 22, je okno pro projekt, do něhož se vkládají jednotlivá Virtual Instruments (VI). VI uložená v projektu nakonec tvoří strukturu aplikace.

Velikou výhodou LabVIEW je to, že se mohou jednotlivé While Loop smyčky vykonávat asynchronně. Důkaz lze spatřit na obrázku 22.

Na obrázku 22 v okně nalevo si lze povšimnout dvou na sobě nezávislých smyček, a to smyčky 1 a smyčky 2. Každá smyčka v jednom cyklu přičte jedničku k předešlé hodnotě, avšak smyčky běží asynchronně. Smyčka 1 se vykoná každých patnáct milisekund a smyčka 2 každých sto milisekund. Na obrázku 22 v okně napravo nahoře (Front Panel) lze vidět tlačítko „Stop“, které zastaví vykonávání smyčky, ke které náleží, poté „indicator“, který ukazuje aktuální hodnotu ve smyčce, a nakonec počet vykonaných cyklů smyčky. Z údajů na Front Panelu lze tedy najít opravdu důkaz, že smyčky běží asynchronně.

Jestliže je potřeba předávat data z jedné smyčky do druhé, využívají se tzv. fronty, do kterých se zařadí prvek z jedné smyčky, v našem případě ze smyčky 2, a opět jej vyřadí z fronty v jiné smyčce, v našem případě ve smyčce 3. Jak si lze opět povšimnout na Front Panelu na ukazateli počtu cyklů, které smyčky vykonávají, běží opět asynchronně. Díky frontě jsou data předávána ze smyčky 2 do smyčky 3 v pořádku, protože aktuální hodnoty ve smyčce 2 a 3 se shodují, i když smyčka 3 vykonala dvakrát více cyklů než smyčka 2.



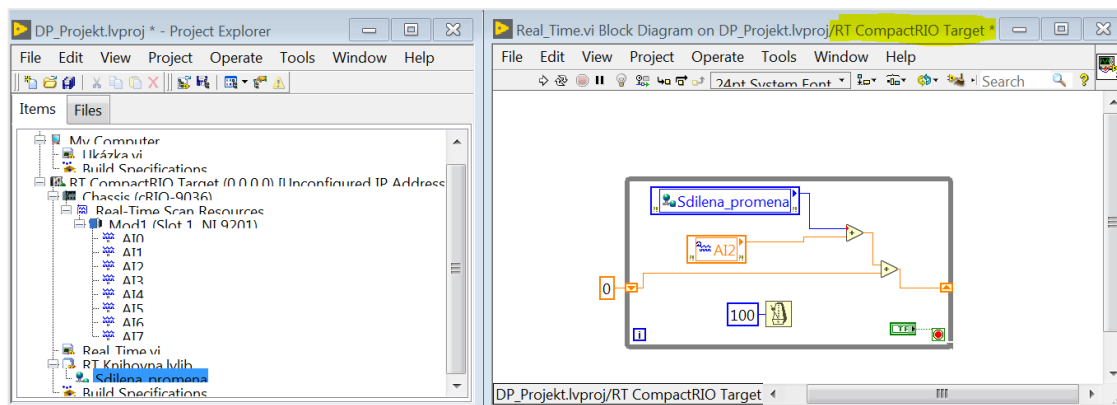
Obrázek 22: LabVIEW pro Host-PC

Program Labview, který byl vytvořen, je zařazen v projektu a na obrázku 22 zobrazen dole napravo.

5.3.3 LabVIEW Real-Time

Aby bylo možné se připojit k LabVIEW Real-Time, je potřeba nejdříve do projektu přidat cRIO. Přidání cRIO do projektu je velice jednoduché, stačí kliknout pravým tlačítkem na název projektu, poté na New → Targets and Devices... → New target or device → Real-Time CompactRIO a na konec vybrat z nabídky správný druh cRIO a stisknout tlačítko „ok“. cRIO se přidá do projektu, přičemž je potřeba v nastavení cRIO zadat jeho IP adresu. Po přihlášení k cRIO si lze vybrat jeho nastavení a zda má pracovat v módu Scan Engine. Po kliknutí pravým tlačítkem na cRIO lze vytvořit VI v Real-Time systému. Zda je VI správně vytvořeno

v Real-Time systému lze poznat podle jeho koncovky, kde bude napsáno RT CompactRIO Target, jak je zvýrazněno na obrázku 23.



Obrázek 23: LabVIEW Real-Time

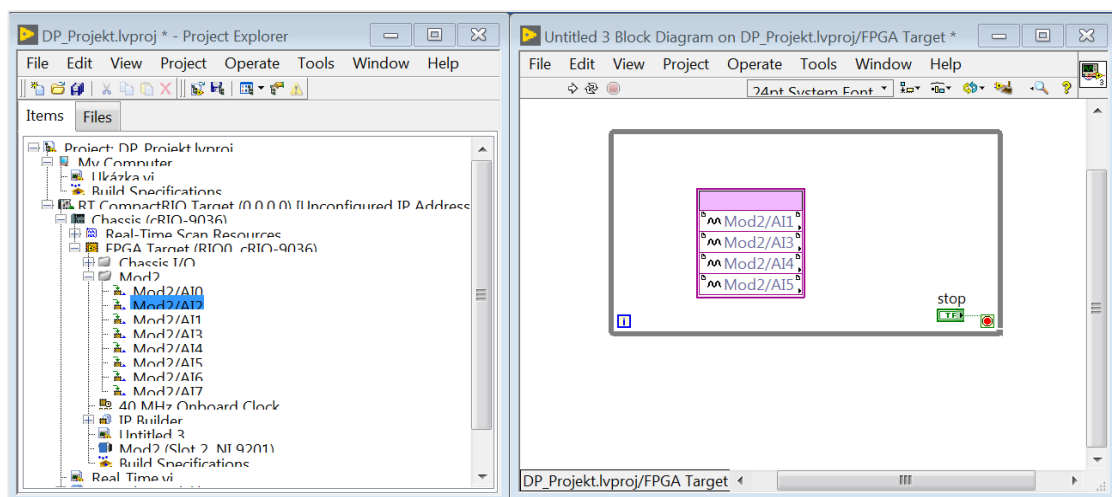
Obrázek 23 znázorňuje projekt vytvořený v LabVIEW Real-Time, kde, jak již bylo popsáno, je cRIO přidáno do projektu a vytvořeno VI Real_Time.vi. Lze si všimnout, že pod ikonou RT CompactRIO Target je vytvořena další ikona Chassis (cRIO – 9036), která přímo odkazuje na druh cRIO a obsahuje prvek Real-Time Scan Resources, pod který se přidávají jednotlivé C Series moduly. Nyní je přidán modul Mod1 (Slot 1 9201). Slot 1 značí místo umístění v rámci cRIO a číslo 9201 název C series. C series se do této struktury umisťují v případě, že je využíván mód Scan Engine.

Jestliže je využíván mód Scan Engine, je čtení dat ze senzorů velice jednoduché, stačí pouze vstup v daném C Series přetáhnout z projektu do Block Diagramu. Na obrázku 23 je takto přetažen AI2 (oranžový obdélník). Jestliže je vyžadována komunikace aplikace vytvořené v Real-Time procesoru s aplikací v Host-PC, využívají se tzv. Shared Variables (sdílené proměnné). Jejich vytvoření je velice jednoduché, stačí opět pravým tlačítkem myši kliknout na ikonu RT CompactRIO a vybrat v nabídce New → Library. Poté se ve stromu CompactRIO objeví knihovna, v případě obrázku 23 knihovna RT-Knihovna, na kterou je třeba znovu kliknout pravým tlačítkem myši a v nabídce vybrat New → Variable a následně proměnou nastavit na Network-Published a přetáhnout do vytvořeného VI v LabVIEW Real-Time, popřípadě do Host-PC (na obrázku modrý obdélník „Sdilená_proměnná“).

Možnosti aplikace v LabVIEW Real-Time se příliš neliší od aplikace v Host-PC, pouze některé toolkity nejsou podporovány, jako například Database Connectivity Toolkit. Avšak většina funkcí je na rozdíl od LabVIEW FPGA zachována.

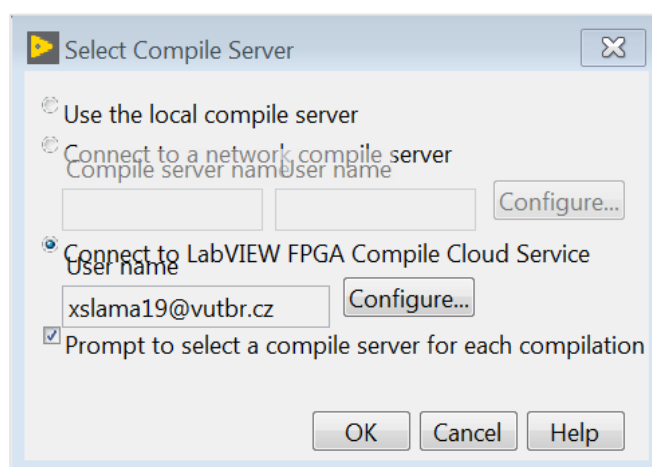
5.3.4 Labview FPGA

K přístupu do LabVIEW FPGA je zapotřebí najet myši na Chassis, stisknout pravé tlačítko a v nabídce vybrat New → FPGA Target. Do stromu projektu přibude FPGA Target. Moduly C Series je opět potřeba přidat: pravým tlačítkem myši kliknout na FPGA Target → New → C Series Modules. Jednotlivé vstupy a výstupy se dají opět jednoduše přetáhnout do VI, které je vytvořené v rámci FPGA Target. Správné vytvoření VI v LabVIEW FPGA lze zkontrolovat znovu podle názvu souboru v Block Diagramu nebo v názvu na Front Panelu. Vytvořené VI a projekt v rámci LabVIEW FPGA je znázorněn na obrázku 24.



Obrázek 24: LabVIEW FPGA

LabVIEW FPGA má několik specifík oproti předešlým módům. Jsou značně omezeny možnosti programování, protože je na výběr méně bloků a toolkitů. Další značnou změnou je provedení kompilace po prvním spuštění programu. Objeví se kompilační okno, jak je znázorněno na obrázku 25. V kompilačním okně je na výběr několik možností, jak lze napsaný kód v LabVIEW FPGA kompilovat. První možností je kompilace na lokálním počítači, nebo se lze připojit na síťový server, anebo poslední možností, a vcelku zajímavou, je možnost využít servery NI a provést kompilaci kódu na nich, přičemž National Instruments nabízí různé testovací verze.



Obrázek 25: LabVIEW FPGA – Kompilační okno

5.3.5 Ukládání dat

Jestliže data nechceme pouze zobrazovat, ale chceme je i ukládat pro pozdější použití, nastává otázka, jak data ukládat. CompactRIO umožňuje ukládat data na svůj lokální disk. Přístup k tomuto disku je velice jednoduchý a jeho velikost se liší v závislosti na druhu cRIO. Další možností je ukládání dat na USB flash disk, který lze ke cRIO připojit, a následně z uložených dat vybírat ta, která jsou pro uživatele zajímavá, a zobrazovat je například v grafu.

Je potřeba si ale uvědomit, že data ukládaná z dlouhodobějšího měření nebo z měření na vysoké frekvenci mohou mít milióny řádků. Takto velká data již nelze zcela triviálně zpracovat, a právě proto byly vytvořeny databázové systémy, v kterých lze velmi rychle a elegantně vybrat konkrétní data, která uživatele zajímají.

Jelikož jsou databázové systémy v dnešní době využívány prakticky všude, kde se s objemnějšími daty pracuje, bude v další kapitole nastíněna problematika databázových systémů.

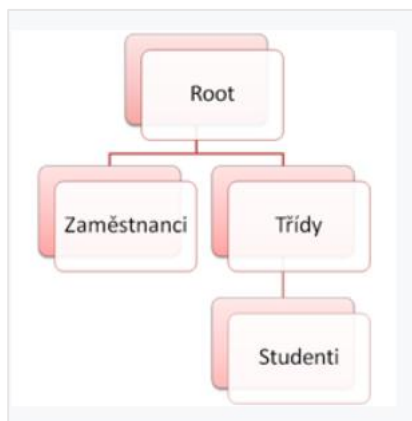
6 DATABÁZE

Databázový systém je systém, který slouží k uchování a organizaci dat podle určených specifikací. Existuje mnoho druhů databází, jako jsou například [30]:

- hierarchická databáze,
- síťové databáze,
- ploché databáze,
- relační databáze,
- objektové databáze.

Ploché databáze neboli flat databáze jsou základním typem databází. Jsou charakteristické tím, že neobsahují vazby na jiné záznamy. Ploché databáze si lze představit jako například CSV soubor, kde jsou na jednotlivých řádcích uchovány záznamy, které jsou pomocí oddělovačů rozřazeny do sloupců. Tento typ databází lze využít pro základní ukládání záznamů. [30]

Pokročilejšími databázemi a zároveň předchůdci nynějšího systému databází jsou databáze hierarchické a síťové. Možná struktura hierarchické databáze je zobrazena na obrázku 26.



Obrázek 26: Hierarchická databáze [30]

U plochých databází může docházet k zaznamenávání redundantních dat, tj. opakujících se záznamů, čemuž se hierarchické a síťové databáze snaží vyhnout. Dochází k tomu, že jsou definovány rodičovské tabulky, a poté musí každý záznam v nižší hierarchické tabulce obsahovat odkaz na jeden ze záznamů ve vyšší hierarchické tabulce. Tím dochází k vytvoření hierarchického systému databáze. Síťové databáze jsou velice podobné jako hierarchické, avšak s tím rozdílem, že lze ke každé nižší hierarchické tabulce definovat více rodičovských tabulek. [30]

Nejpoužívanějším databázovým systémem jsou databáze relační, které uchovávají informace v tabulkách organizovaných do řádků a sloupců. Důležité je, aby každá tabulka měla tzv. primární klíč, což je unikátní číslo přiřazené právě zapsané informaci do tabulky, aby nedocházelo k duplicitám. [31]

Čísla přiřazená primárními klíči k informacím v tabulce nemusí jít zcela popořadě, i když jsou přiřazována postupně. Stačí například vymazat informaci číslo dva, čímž dojde k její ztrátě, což ale nevadí, protože je stále zachována jedinečnost informace. [31]

S daty takto uloženými lze pohodlně manipulovat, a to nejčastěji pomocí databázového jazyka SQL (Structured Query Language – Strukturovaný dotazovací jazyk). Pomocí jazyka SQL se dotazuje na určitý druh dat v databázi, přičemž databáze vrací požadovaná data. [31]

Aby bylo možné využívat databáze, je potřeba mít nainstalovaný databázový server. Databázových serverů existuje celá řada například [31]:

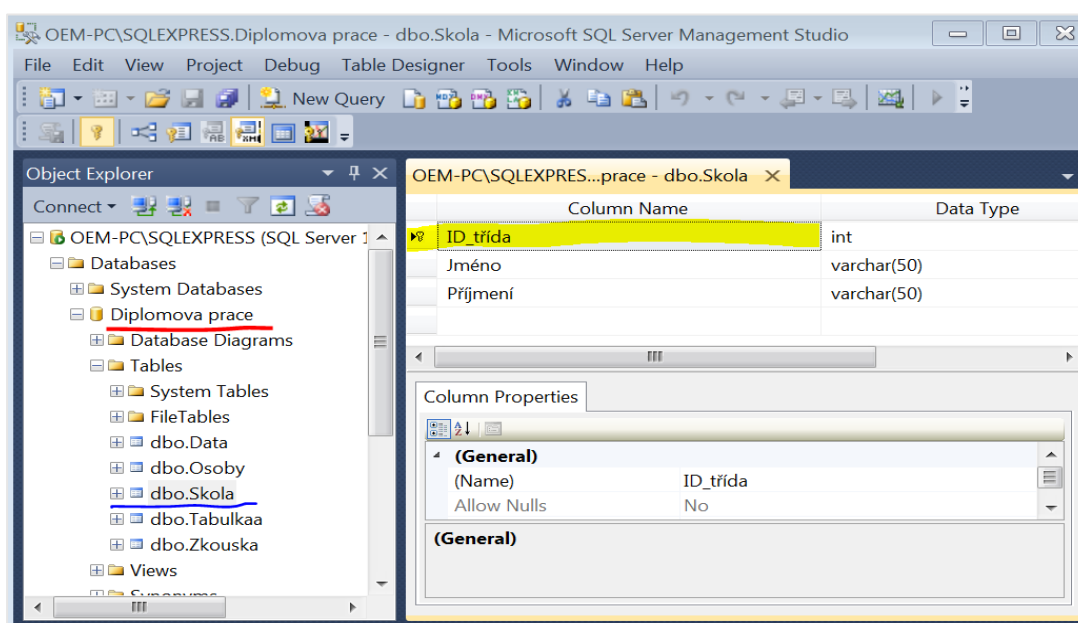
- MySQL,
- MS SQL,
- PostgreSQL,
- Oracle,
- Firebird.

Databázové servery se od sebe liší ve svém výkonu, podporovaném operačním systému, programovacím jazyce a v mnoha dalších ohledech. Dva nejznámější databázové servery MySQL a MS SQL se například liší v licenci, kde MySQL je volně dostupný, zatímco, aby bylo dosaženo plné podpory, u MS SQL je potřeba si licenci zaplatit (existují i verze zdarma, ale s omezeními), dále se také v podporovaném operačním systému, kde MySQL podporuje FreeBSD, Linux, OS X, Solaris a Windows, zatímco MS SQ podporuje pouze Linux a Windows. Dále lze najít rozdíl v podporovaných programovacích jazycích.

Důležité je si uvědomit, že ačkoliv MySQL a MS SQL mají v názvu zkratku SQL a tváří se, že využívají zcela identický dotazovací jazyky, jsou v něm nepatrné rozdíly, na které je potřeba myslet, jestliže se přechází z jednoho databázového serveru na druhý.

Aby bylo možné přistoupit k MS SQL Serveru, je potřeba mít nainstalovaný Microsoft SQL server Management Studio. Jak SQL server, tak i Management Studio lze bezplatně stáhnout z internetu v jednom balíku, přičemž jsou zde určité omezení, například v kapacitním omezení na SQL serveru, což však není pro tvorbu menšího databázového systému problém.

Vytvářet tabulky lze pomocí ručního zápisu velmi jednoduše. Stačí v Databases kliknout pravým tlačítkem myši a vybrat New Database, čímž se v stromu vytvoří nová databáze, kde obdobným způsobem lze vytvořit tabulku. Jak lze vidět na obrázku 27, vytvořili jsme tabulku Skola (modře podtrženo) v databázi Diplomova práce (červeně podtrženo) a nastavili jsme ID_třída primární klíč (žlutě zvýrazněno). ID_třída, Jméno, Příjmení znázorňují jednotlivé sloupce v tabulce a, jak si lze povšimnout, lze jim přiřazovat datové typy, proto je potřeba si dopředu rozmyslet, jaká data se budou do jednotlivých sloupců ukládat.



Obrázek 27: SQL Databáze

Vytvářet tabulky lze i pomocí SQL jazyka. SQL kód pro vytvoření tabulky, která byla výše popsána, by vypadal následovně:

```
CREATE TABLE [Skola] (  
  [ID_třída] INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
  [Jméno] VARCHAR(50) NOT NULL,  
  [Příjemní] VARCHAR(50) NOT NULL  
)
```

Takto vytvořené tabulky lze různě spojovat a rozdělovat. Nutno však podotknout, že dotazovací jazyk SQL nikdy není sám, vždy je jako podpůrný prostředek jinému jazyku. Pomocí SQL příkazů se určitý program dotáže na data obsažená v databázi, a právě SQL příkaz pošle data z databáze příslušnému programu, který je poté může zpracovávat.

7 POUŽITÉ MĚŘICÍ VYBAVENÍ

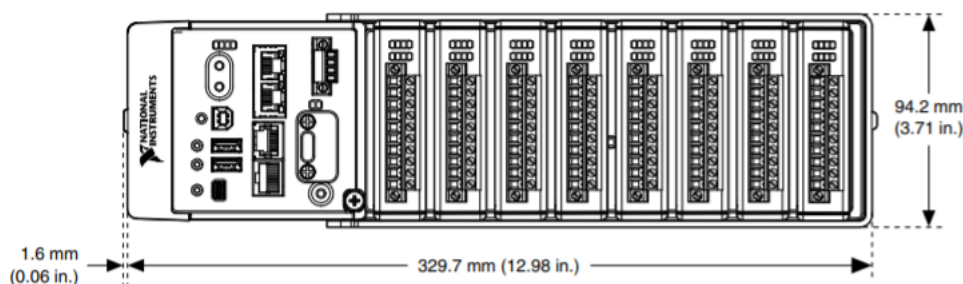
Cílem diplomové práce je navrhnout a implementovat systém pro sběr dat z obráběcího stroje, kde řídicí jednotku bude představovat platforma cRIO s příslušnými moduly. CompactRIO bylo upřednostněno před PLC, o kterém bylo psáno výše a které by na tento typ úlohy také postačovalo, hlavně z důvodu snazšího uživatelského rozhraní použitého softwaru.

CompactRIO, které je využíváno v diplomové práci, nese označení NI cRIO – 9036, které se řadí k těm dražším a výkonnějším. C Series Moduly, které jsou použity v diplomové práci, nesou označení NI 9237 a NI 9234. Díky těmto modulům lze číst data ze senzorů akcelerometru a tenzometru. Dále je také použit tzv. EtherCAT, který je připojen k cRIO pomocí sériové linky a díky kterému lze číst hodnoty ze senzoru teploty.

V následujících podkapitolách budou jednotlivé prvky, které jsou použity k čtení hodnot ze senzorů, podrobněji rozebrány.

7.1 NI CompactRIO-9036

Na obrázku 28 lze vidět pohled shora na cRIO-9036, které měří 329,7 mm na délku, čím se řadí mezi větší cRIO. Jeho Chassis je uzpůsobeno pro osm C Series Modulů, přičemž se používají pouze dva, jak již bylo dříve psáno v diplomové práci.

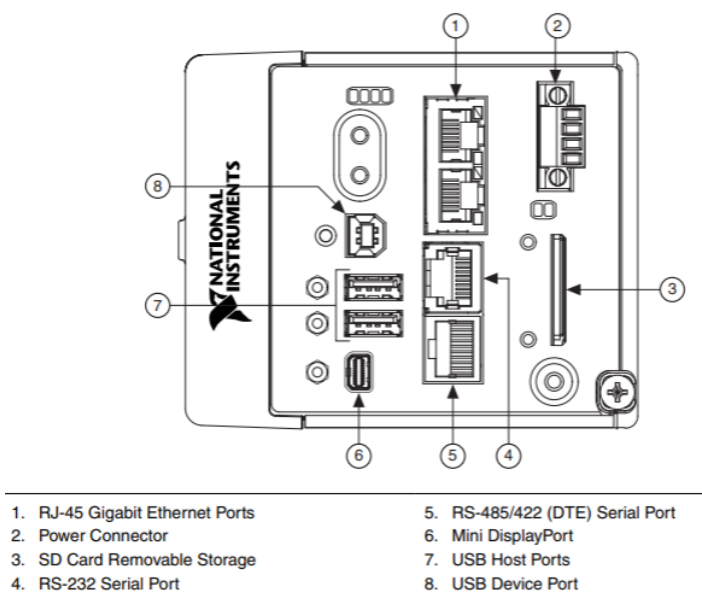


Obrázek 28: NI cRIO-9036 [32]

CompactRIO-9036 má dvoujádrový procesor s frekvencí 1,33 GHz a lze jej připojit k Host-PC pomocí USB zařízení nebo pomocí RJ-45 Gigabit Ethernetu, pro který má cRIO-9036 připraveno dva porty.

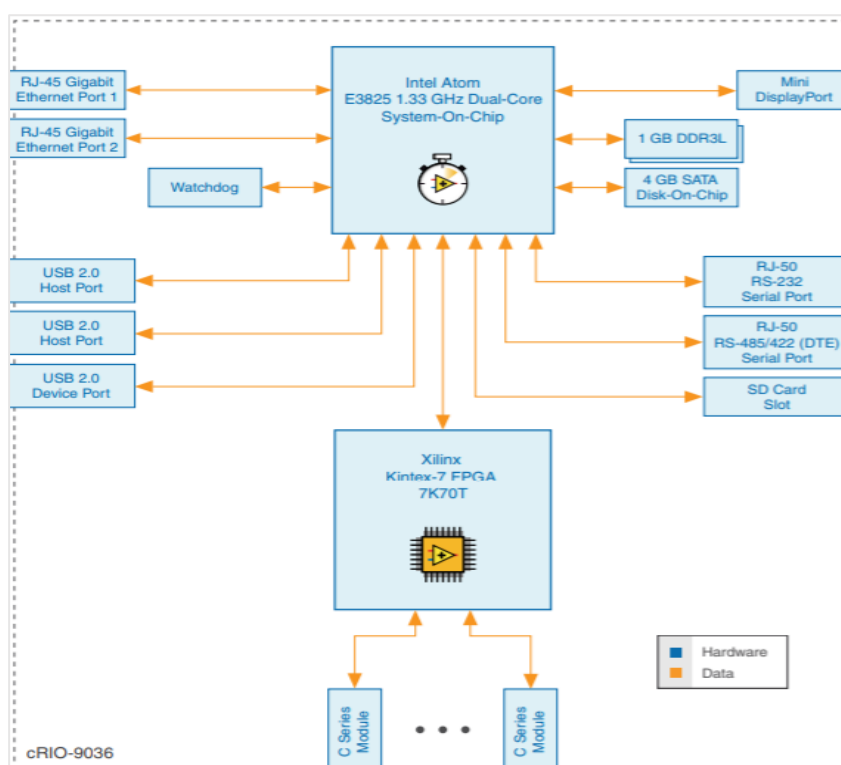
Přehled jednotlivých vstupů a výstupů, které nabízí cRIO-9036, lze vidět na obrázku 29. V rámci diplomové práce je aktivně využíván port pro Ethernet, díky kterému je Host-PC schopno navázat kontakt s cRIO odkudkoliv, jestliže není zablokovaná vnitřní ethernetová síť, na které se cRIO připojuje. Dále je také využíván sériový port RS-232, který spojuje cRIO s EtherCAT a který bude blíže popsán v následujících podkapitolách.

Lze si všimnout, že cRIO-9036 umožňuje nepřeberné množství vstupů a výstupů, jak přes rozhraní controlleru, tak přes Chassis, a to konkrétně přes C Series Moduly.



Obrázek 29: Controller cRIO-9036 [32]

Obrázek 30 pomůže dotvořit vizuální představu o architektuře zmiňovaného cRIO. Na obrázku 30 lze vidět relaci mezi Real-Time procesorem a FPGA a následnými vstupy a výstupy z platformy CompactRIO-9036.



Obrázek 30: Schéma cRIO-9036 [32]

Po představení použitého cRIO lze nyní přejít k jednotlivým komponentům, které jsou využity pro měření dat a jejich přenášení ze senzorů do cRIO-9036.

7.2 C Series Modul NI-9237

Modul NI-9237 je čtyřkanálový, používaný u senzorů, které jsou napájené a měřené na základě čtyřmůstkového zapojení. Mezi takové senzory se řadí i senzory pro měření deformací neboli tenzometry. Velkou výhodou tohoto modulu je, že umožňuje offset hodnot pro kalibraci, tudíž se výborně hodí na měření přetvoření a můstkové měření. [33]

Na obrázku 31 lze vidět základní specifikaci NI 9237.

C SERIES SIMULTANEOUS BRIDGE MODULE COMPARISON					
Model	Channels	Sample Rate	Resolution	Connectivity	Simultaneous Supported Bridges
NI 9218	2	51.2 kS/s/ch	24 bits	LEMO, 9-pin DSUB	Quarter, Half, Full
NI 9219	4	100 S/s/ch	24 bits	Spring Terminal	Quarter, Half, Full
NI 9235	8	10 kS/s/ch	24 bits	Spring Terminal	120 Ω Quarter Bridge
NI 9236	8	10 kS/s/ch	24 bits	Spring Terminal	350 Ω Quarter Bridge
NI 9237	4	50 kS/s/ch	24 bits	RJ-50, DSUB	Quarter, Half, Full

Obrázek 31: NI-9237 [33]

Podporované můstkové zapojení není jen plné, ale také poloviční nebo čtvrtinové, avšak nejpresnější hodnoty z měření lze získat po využití plného mostu, kterým lze kompenzovat nežádoucí vlivy na měření. [33]

Dále je potřebné si uvědomit, že po připojení senzoru k modulu dochází k nepřesnostem díky odporu vedení v kabelech. Datasheet od NI uvádí dvě možnosti, jak tyto chyby eliminovat, jsou jimi [33]:

- Remote sensing,
- Shunt calibration.

Jak si lze povšimnout na obrázku 31, jsou podporovány dva druhy připojení senzoru k modulu, a to pomocí RJ-50 nebo pomocí DSUB. V diplomové práci je využíváno připojení pomocí RJ-50, které je levnější variantou. [33]

7.3 C Series Modul NI-9234

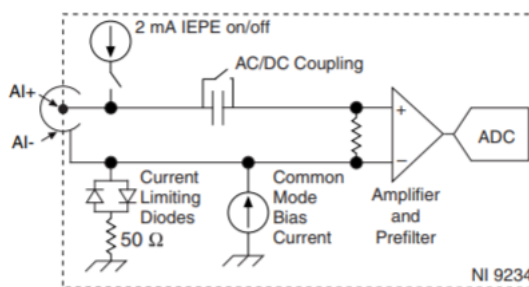
Modul NI-9234 se značně liší od Modulu 9237 už jen koncovkami k připojení senzorů. Modul NI-9234 je využíván pro velmi přesné měření z IEPE (Integrated Electronics Piezo-Electric) senzorů. Princip těchto senzorů je vysvětlen v podkapitole 4.4. [34]

NI-9234 má široké využití v aplikacích zejména při monitorování průmyslových strojů, jejich vibrací a hlučnosti. Na obrázku 32 lze vidět základní údaje k modulu NI-9234. [34]

C SERIES ANALOG MODULE COMPARISON							
Product Name	Signal Ranges	Channels	Sample Rate	Input Configurations	Noise at Maximum Sample Rate	Connectivity	Isolation Continuous
NI 9218	± 5 V	2	51.2 kS/s/ch	IEPE with AC Coupling	50 μ Vrms	9-Position DSUB, LEMO	60 VDC Ch-Ch
NI 9230	± 30 V	3	12.8 kS/s/ch	IEPE with AC Coupling, AC Coupling, DC Coupling	106 μ Vrms	Screw Terminal	60 VDC Ch-Earth
NI 9232	± 30 V	3	102.4 kS/s/ch	IEPE with AC Coupling, AC Coupling, DC Coupling	251 μ Vrms	Screw Terminal	60 VDC Ch-Earth
NI 9234	± 5 V	4	51.2 kS/s/ch	IEPE with AC Coupling, AC Coupling, DC Coupling	50 μ Vrms	BNC	None
NI 9251	± 4.24 Vpk	2	102.4 kS/s/ch	AC Coupling, DC Coupling	8.8 μ Vrms	mini XLR	None

Obrázek 32: NI-9234 [34]

Jak lze zjistit z datasheetu, analogové vstupy jsou přes odpor padesát ohmů uzemněny k chassis. Aby se minimalizoval šum, je potřeba, aby chassis bylo také uzemněné. Schéma analogového vstupu lze vidět na obrázku 33. Poté je analogový signál převáděn na digitální pomocí ADC převodníku. [34]

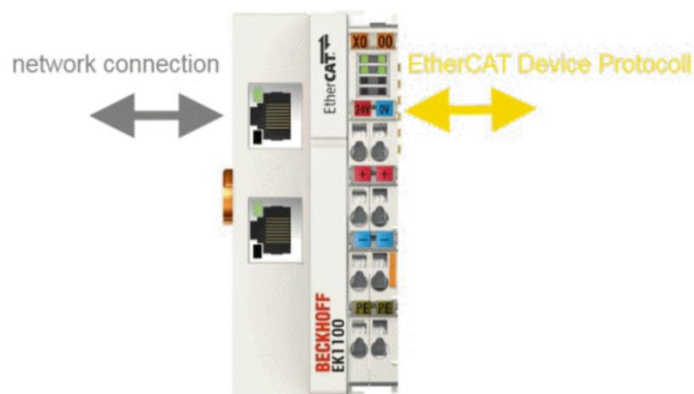


Obrázek 33: Analogový vstup [34]

Dále si lze všimnout, že k modulu NI-9234 mohou být připojeny čtyři senzory přes BNC port. Modul využívá analogovou a digitální filtraci signálu mající anti-aliasing filter, tudíž by nemělo docházet k podvzorkování dat a ztrátě informací. [34]

7.4 Backhoff EK1100

Backhoff EK1100 je využíván jako mezipřevodník mezi senzorem tepla a cRIO. Na obrázku 34 lze vidět zařízení od Backhoff. Lze si povšimnout, že levá strana je přizpůsobena síťovému připojení, zatímco pravá strana pro slave zařízení. Samotné zařízení Backhoff EK1100 je master a převádí procházející telegram z Ethernetu 100BASE-TX do E-bus signálové reprezentace, přičemž je připojen do sítě, tj. do vyššího ethernetového rozhraní. Jedna stanice může obsahovat různý počet EtherCAT terminálů a koncový bus terminál. [35]



Obrázek 34: Backhoff EK1100 [35]

EtherCAT je hojně využíván pro svoji rychlost. Jeho výhoda pramení převážně z konstrukce protokolu, který dokáže během jednoho cyklu přes slave zařízení sbírat a zároveň předávat data. Tímto se značně zrychluje komunikace oproti jiným protokolům. [35]

V dalších kapitolách bude přiblížen systém sběru dat, bude popsána aplikace vytvořená v LabVIEW a dále také databázový systém.

8 PROGRAM V LABVIEW

Jak již bylo zmíněno, operačním softwarem pro cRIO je LabVIEW, tudíž i celý měřicí řetězec byl vytvářen v LabVIEW. Jednotlivými moduly, které jsou potřebné k práci s cRIO, jsou:

- LabVIEW,
- LabVIEW Real-Time,
- LabVIEW FPGA.

Při tvorbě programu byly potřebné výše zmíněné moduly, avšak k základnímu sadě LabVIEW bylo dále nutné stáhnout Database Connectivity Toolkit, aby bylo možné pracovat s databázemi.

LabVIEW nabízí zkušební verzi na měsíc, nebo půlroční pro studenty, nejedná se však o zcela plné verze, proto bylo zapotřebí při tvorbě programu použít školní licenci, aby byl zpřístupněn všechny obsah LabVIEW, který je k tvorbě takového programu potřebný.

V první fázi tvorby programu bylo potřeba doinstalovat všechny drivery tak, aby byly kompatibilní s verzí používaného LabVIEW. Drivery se instalují v NI MAX (Measurement & Automation Explorer), což je jeden z mnoha podpůrných programů, který se nainstaluje společně s LabVIEW. Po otevření NI MAX stačí pouze najít dané cRIO a poté doinstalovat, popřípadě odinstalovat, různá rozšíření LabVIEW.

Další podpůrný program od NI (National Instruments), který lze nainstalovat zároveň s LabVIEW, je NI Update Service, v němž lze nalézt přehled jednotlivých nainstalovaných verzí programů od NI, jako například LabVIEW a různé toolkity.

Po správném nainstalování všech potřebných verzí a doplňků lze začít řešit tvorbu programu. V následujících podkapitolách budou rozebrány jednotlivé prvky programu, přičemž prvně bude zmíněna zjednodušená architektura programu neboli makroarchitektura kvůli snazšímu vstupu do problematiky navržené aplikace.

Nutno podotknout, že aplikace se skládá ze tří hlavních VI a mnoha menších SubVI. Hlavními VI pro jednotlivé módy jsou:

- Host-PC – Ovládání.vi
- LabVIEW Real-Time – RT_Hlavni.vi
- LabVIEW FPGA – Hlavní_FPGA

Přičemž přes VI Ovládání.vi se řídí celá aplikace. V odevzdaném souboru lze kromě zmíněných tří hlavních VI pro každý mód a stromu projektu se SubVI nalézt také složku s vytvořenou aplikací pro Host-PC. Tudíž jestliže program poběží v cRIO není již potřeba ve stromu projektu otevírat Ovládání.vi, ale nýbrž stačí pouze spustit aplikaci v dané složce. Aby *executables* program fungoval, je potřeba mít nainstalovaný run-time engine.

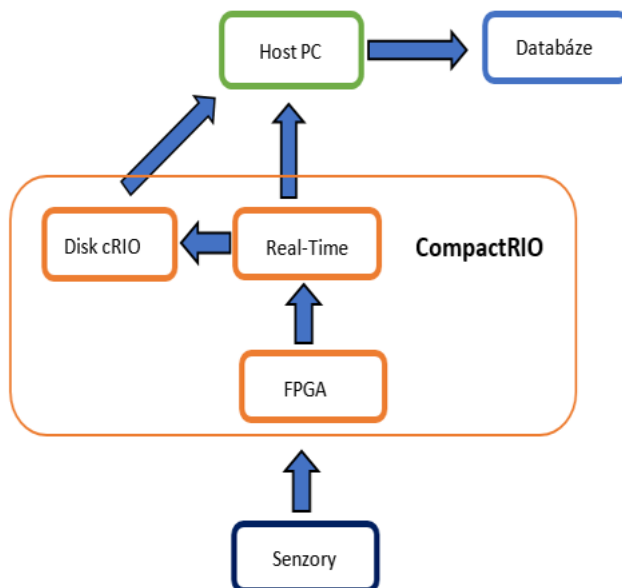
Program byl vytvořen v operačním systému Windows 7 se specifickým nastavením, pro tenhle operační systém. Při spouštění programu na jiném operačním systému s odlišným nastavením může dojít k odlišnému zobrazení aplikace Ovládání.exe nebo Ovládání.vi. Důležité je si také uvědomit, že antivirové programy mohou EXE soubory blokovat a mazat.

Nyní již k samotnému schématu programu, který je vytvářen ve verzi LabVIEW 2017.

8.1 Makroarchitektura programu

Tato podkapitola se bude zabývat celkovým přehledem programu, zejména tokem dat mezi jednotlivými vrstvami a komunikací. V pozdějších podkapitolách budou rozebrány jednotlivé bloky programu.

Architektura programu byla navržena tak, jak ji lze vidět na obrázku 35.



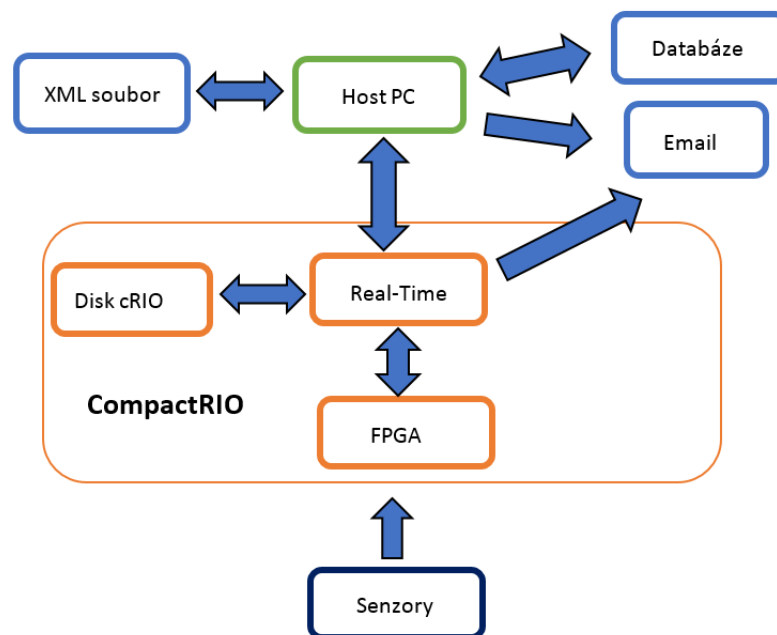
Obrázek 35: Tok naměřených hodnot

Nejdříve jsou hodnoty měřených veličin nasnímány pomocí senzorů, dané hodnoty jsou poté čteny v FPGA a následně jsou ukládány do FIFO paměti, která je poté přečtena v Real-Time. V této fázi dochází ke zpracování dat. Data jsou navzorkována a ke každé hodnotě jsou přidány informace o času, vzorkovací frekvenci a frekvenci FPGA.

Takto obohacená data jsou následně ukládána na disk cRIO a posílána na Host-PC pomocí sdílených proměnných, jak již bylo zmiňováno v podkapitole 5.3.3. Jak lze vidět z obrázku 35 data mohou být odeslána do Host-PC z Real-Time přímo, anebo z disku cRIO, kde byla dříve data uložena. O tom, jak budou data posílána, rozhoduje komunikace mezi jednotlivými vrstvami

Posledním krokem zpracování dat, je odeslání dat z Host-PC na databázový server (MS SQL server). Zde jsou data zaznamenána a připravena pro další zpracování. Konfigurace posílání dat z Host-PC na databázový server je řešena z důvodu, že LabVIEW Real-Time neumožňuje používat *LabVIEW Database Connectivity Toolkit*, díky čemuž jsou data posílána na databázový server.

Aby bylo možné data organizovat a řídit jejich tok mezi jednotlivými vrstvami, je zapotřebí vytvořit komunikaci mezi vrstvami. Principiálně to vypadá tak, že jedna vrstva vydá požadavek jiné vrstvě, že chce přijímat data, a tázaná vrstva data odešle. Na obrázku 36 je znázorněna komunikace mezi jednotlivými vrstvami.



Obrázek 36: Komunikace mezi vrstvami

FPGA vysílá požadavek senzorům, že chce přijímat data, následuje komunikace mezi FPGA a Real-Time. Real-Time posílá FPGA zprávu, že může začít zapisovat data do FIFO paměti, přičemž Real-Time začne data z FIFO paměti číst. Po ukončení programu v Real-Time je vyslán požadavek, aby se data přestala zapisovat do FIFO paměti.

Většina funkcí v programu běžících v Real-Time je podřízena ovládacímu programu z Host-PC, avšak některé funkce si ovládá sám, např. řídí FPGA nebo hlídá stav paměti cRIO. O stavu volné paměti v cRIO informuje jak Host-PC, tak i uživatele prostřednictvím e-mailu.

Nyní se dostáváme k řídicímu programu běžícímu na Host-PC. Jak bude ukázáno dále, uživatel má mnoho možností, které může provést v Host-PC v rámci ovládání programu.

Přes program běžící na Host-PC lze komunikovat s Real-Time, přičemž mu lze zadávat nejen požadavky na čtení dat, ale mimo jiné i dotazy na aktuální stav paměti cRIO, požadavek na zapisování měřených hodnot na disk cRIO nebo real-time odesílání informace o chybách a aktuálním stavu cRIO na Host-PC.

Program na Host-PC nekomunikuje pouze s cRIO, ale také s vnějším světem. Zapisuje a čte údaje z XML souboru, v kterém jsou uloženy informace o nastavení e-mailu. Takto vyčtené informace dále odesílá do cRIO a po zadání příkazu dokáže odeslat zkušební e-mail, aby zjistil, zda byly informace zadány správně.

V poslední řadě se přes Host-PC ukládají data na databázový server. Jestliže mají být data zapisována v reálném čase na databázový server, je komunikace jen jednosměrná. Program z Host-PC vyšle požadavek cRIO, že chce zapisovat data na databázový server, data jsou poté posílána z cRIO do Host-PC a následně na databázový server. Další možný požadavek je, aby data uložená na disku cRIO byla zapsána na databázový server, tudíž musí být komunikace obousměrná, aby nedocházelo k zapsání duplicitních dat. Pomocí SQL jazyka je databázovému serveru odeslán dotaz týkající se toho, která data byla zapsána na server naposledy. Přijaté informace o poslední zapsané hodnotě v dané tabulce, do které bude probíhat zápis nových

hodnot, se porovná s uloženými daty v cRIO, a poté jsou zbývající data dopsána na databázový server.

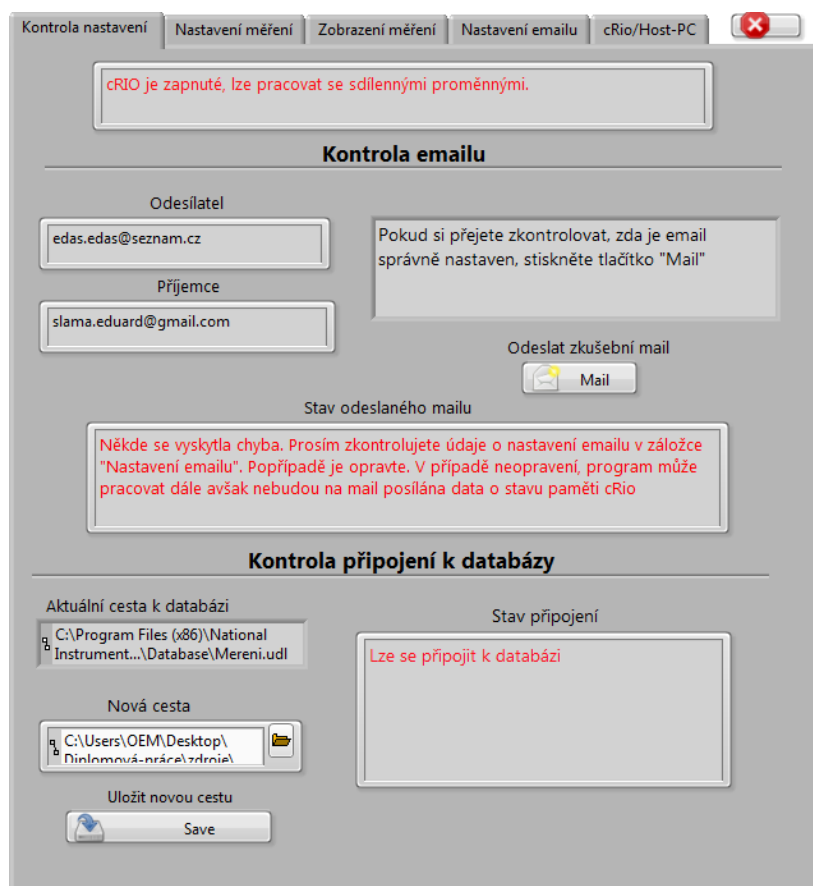
Nyní budou detailněji rozebrány jednotlivé struktury v programech a komunikace mezi programy. Celý program lze poté nalézt v příloženém souboru diplomové práce.

8.2 Program běžící na Host-PC

Program běžící na Host-PC (Ovládání.vi) je řídicí program celého systému. Lze v něm nastavovat a kontrolovat např. frekvenci vzorkování, stav paměti disku cRIO nebo aktuálně měřená data.

8.2.1 Uživatelské rozhraní programu

Uživatelské rozhraní je členěno na záložky, jak lze vidět na obrázku 37. První záložka se týká obecného nastavení. Jde hlavně o kontrolu LabVIEW, zda je schopno se připojit ke cRIO a také zda je schopné se připojit k databázovému serveru a odesílat e-maily. Jestliže se nebude schopno připojit k databázovému serveru, uživatel může změnit cestu k UDL souboru. Zasílání e-mailů lze otestovat odesláním zkušební e-mailu. Jestliže se někde vyskytne chyba, je možno zkontrolovat a upravit informace v záložce *Nastavení emailu*. Jestliže XML soubor s informacemi k e-mailu, je vyplněn správně, tak nejčastější problém je s blokováním odesílání e-mailu z privátní sítě.



Obrázek 37: Kontrola nastavení Host-PC programu

Další záložkou v řadě je záložka *Nastavení měření* (viz. obrázek 38). Zde si uživatel může nastavit vzorkovací frekvence, kde je ošetřena možnost vzorkování s nulovými anebo zápornými hertzi. Uživatel bude vyzván k opravě zadaných hodnot vzorkování. Další možností v záložce je správa měřených dat, přičemž uživatel má na výběr ze čtyř módu:

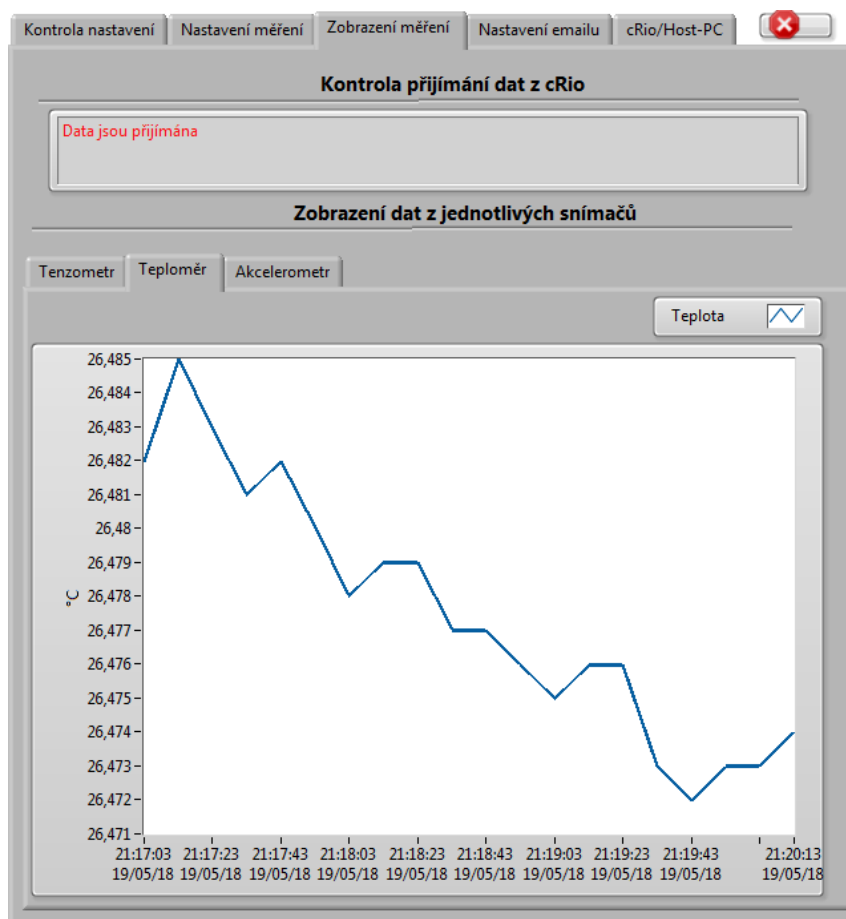
- Pouze sledování měřených dat
- Zapsání dat uložených na disku cRIO do databáze
- Zapsání dat uložených na disku cRIO do databáze a následné zapisování aktuálně měřených hodnot do databáze
- Zápis aktuálně měřených hodnot do databáze

Uživateli se zobrazuje aktuální stav, v kterém právě jeho vybraný mód je, a pohodlně si může zastavit měření tlačítkem *Zastavit měření*.

Obrázek 38: Nastavení měření Host-PC programu

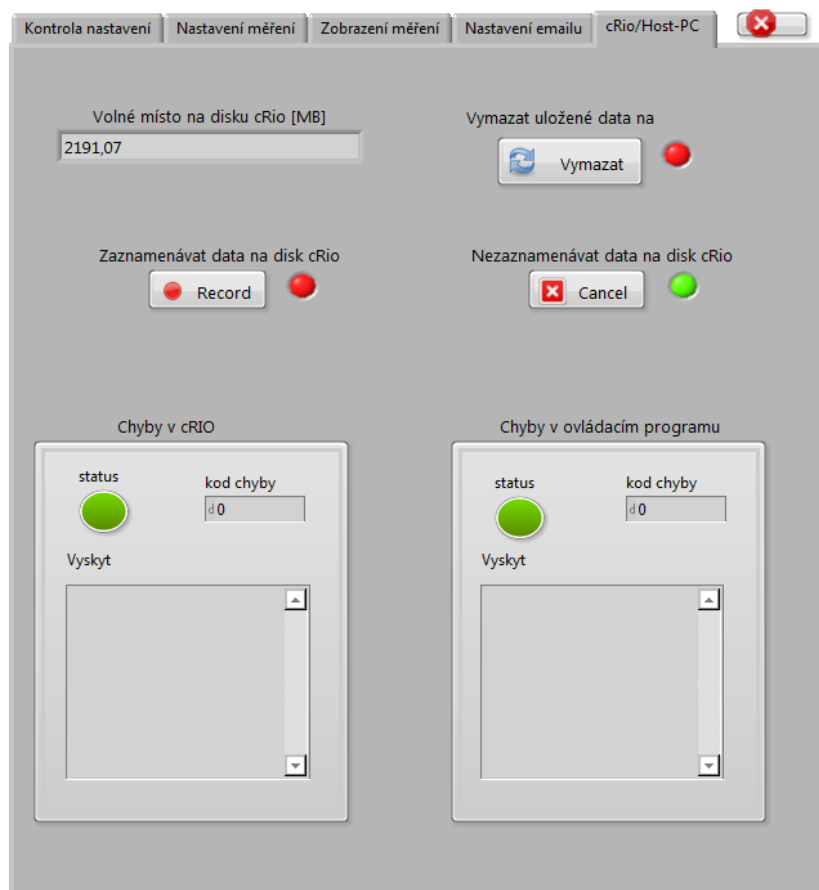
Jedna z posledních záložek, kterou je vhodné zmínit, je záložka *Zobrazení měření*. Zde se uživateli zobrazují aktuálně měřená data po vybrání měřicího módu ze záložky *Nastavení měření*.

Zobrazovaná data jsou rozdělena do tří záložek podle druhu měření (viz. obrázek 39)



Obrázek 39: Zobrazení měření Host-PC programu

V poslední záložce *cRIO/Host-PC* má uživatel možnost ukládat data na disk cRIO anebo je z něj mazat, přičemž se mu zobrazuje i hodnota aktuálně volné paměti disku cRIO. Navíc je do téhle záložky přidána informace o chybových stavech v programu běžícím v cRIO (RT-Hlavní.vi) a programu běžícím na Host-PC (Ovládání.vi). Tahle záložka je vyobrazena na obrázku 40.

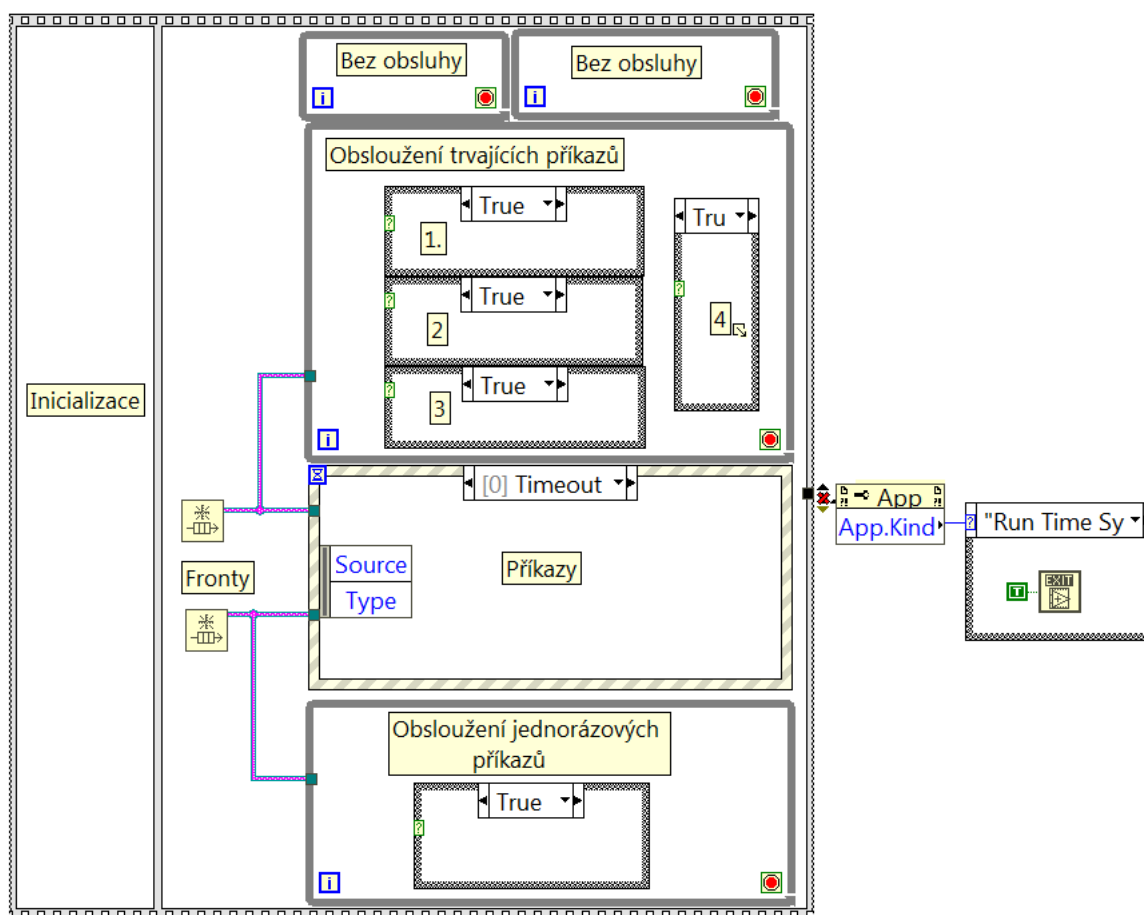


Obrázek 40: Záložka cRIO/Host-PC

Záložka *Nastavení emailu* slouží k úpravě XML souboru a nastavení nových údajů e-mailu, jako například nastavení odesílatele a příjemce, nebo také portu či hesla.

8.2.2 Struktura programu

Na obrázku 41 lze vidět strukturu programu, jehož uživatelské rozhraní bylo popsáno výše. V první fázi dochází k inicializaci proměnných. Inicializace se provádí z důvodu nastavení všech proměnných do defaultního stavu pro případ, že by bylo do některé proměnné zapsáno něco, co je při spuštění programu nežádoucí. Po inicializaci následuje spuštění vlastního programu, který je složen ze čtyř While Loops a jedné Event Structure. Event Structure je pro program nepostradatelná, protože obsluhuje veškeré příkazy, které přijdou z uživatelského prostředí, tzv. Front panelu. Jestliže je program ukončen v Run Time Systému, dojde díky Case Structure úplně napravo k uzavření uživatelského okna.



Obrázek 41: Struktura programu Ovládání.vi

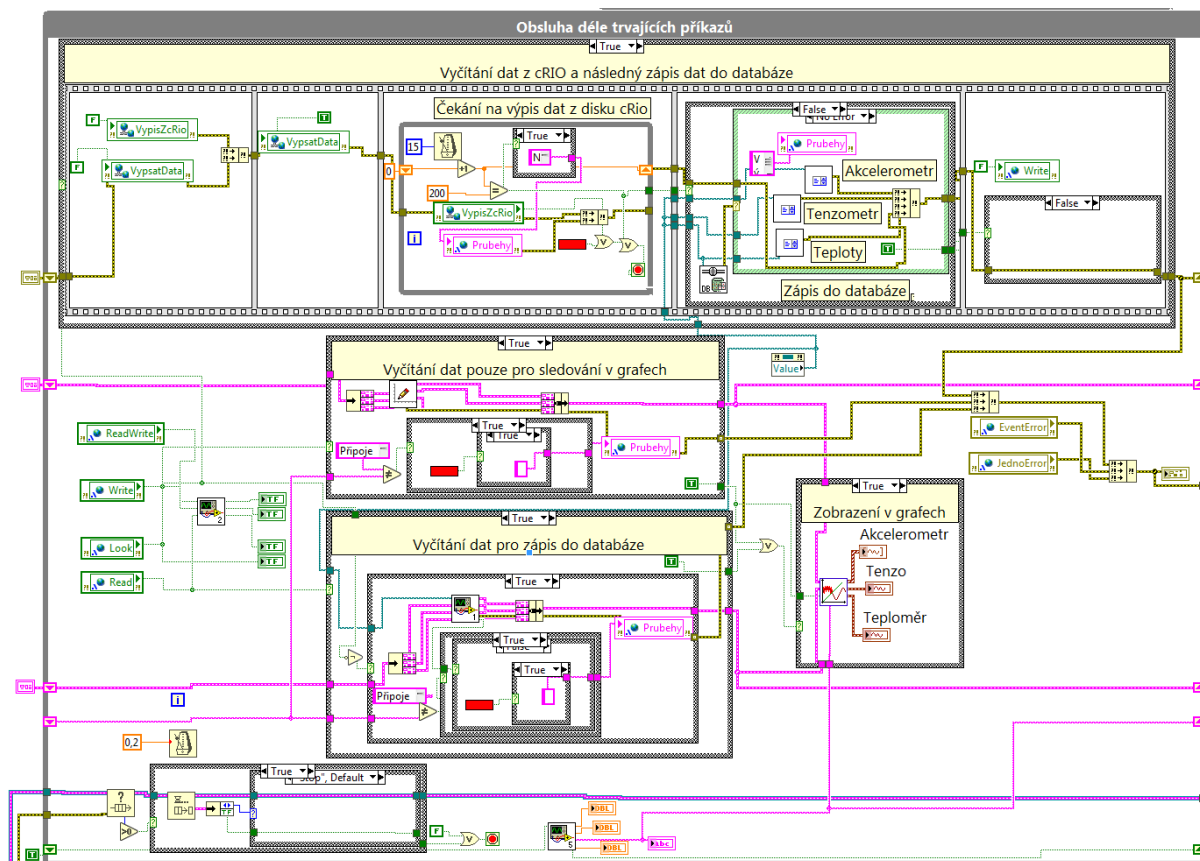
Příkaz, který je vyslán z Front panelu, je vyhodnocen v Event Structure a poté se buď vykoná While Loop, která slouží k obsluze déle trvajících příkazů anebo While Loop, která slouží k obsluze jednorázových příkazů. Parametry příkazu jsou z Event Structure do While Loop předávány pomocí front nebo pomocí lokálních proměnných, tzv. Single process.

Ve smyčce pro obslužení jednorázových příkazů se provádí akce typu *přečti, zkontroluj, nastav*.

Ve smyčce pro obslužení trvajících příkazů se provádí úkony spjaté s měřením dat a jejich zobrazováním v grafu anebo zápisem do databáze. Na obrázku 41 jsou ve While Loop tři Case Structures, čemuž v LabVIEW odpovídá *if* podmínka v jiných programovacích jazycích, kde se vždy vykonává jen jedna z nich, anebo žádná. Case Structures jsou na obrázku označeny čísly 1 až 4.

Case Structure 1, 2 a 3 slouží k čtení dat ze senzorů, ale liší se na základě požadavků, které lze vidět na obrázku 38. V jednotlivých Case Structures jsou poté SubVI, které zpracovávají data a posílají je do Case Structure 4, kde jsou poté zobrazena v grafech. Nutno však podotknout, že v grafech jsou zobrazovány pouze data, která jsou aktuálně měřená. Při zápisu dat z disku cRIO do databáze nedochází k zobrazování hodnot v grafech. Ukázku z programu, o kterém pojednával tento odstavec, lze vidět na obrázku 42. Na tomto obrázku lze vidět, že While Loop pro obsluhu déle trvajících příkazů, obsahuje o jeden Case Structures více, avšak tento Case Structure má pouze zabezpečit ukončení While Loop po příkazu z Event Structure. Dále si lze všimnout i dvou SubVI, které leží mimo Case Structures, které byly zmiňovány. SubVI nejvíce nalevo zabezpečuje správné rozsvícení kontrolky v záložce *Zobrazení měření* a druhé SubVI

ukazuje aktuální požadované frekvence vzorkování a zda je Ovládání.vi připojeno k programu běžícím na cRIO a nebo k samotnému cRIO.



Obrázek 42: Ukázka While Loop

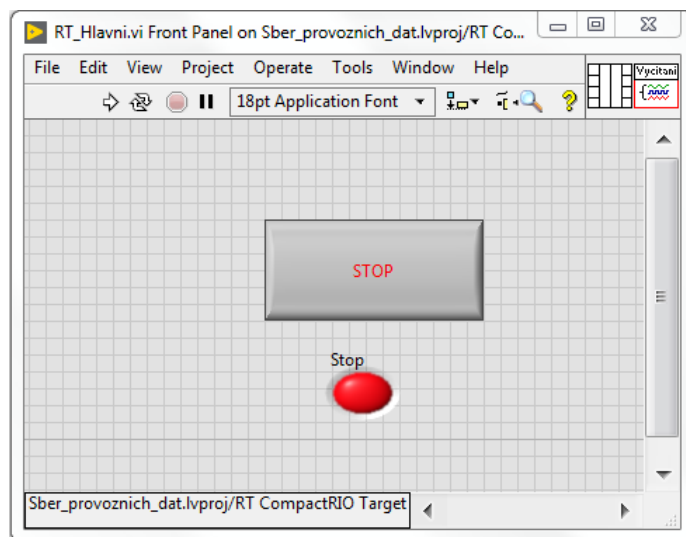
Nakonec jsou na obrázku 41 vidět dvě zbývající While Loops, které nejsou nijak ovládané a běží vůči sobě asynchronně. Tyto smyčky nic neobsluhují, pouze mají zobrazovací charakter.

8.3 Program v LabVIEW Real-Time

Vytvořený program v módu LabVIEW Real-Time je spojka mezi LabVIEW FPGA a programem v Host-PC. Primární úkol programu je číst data z LabVIEW FPGA, poté je navzorkovat a následně ukládat na disk cRIO anebo odesílat do programu běžícího na Host-PC. Mimo jiné program hlídá například stav disku cRIO. Více o programu v následujících podkapitolách.

8.3.1 Uživatelské rozhraní programu

Uživatelské rozhraní je velmi jednoduché. Předpokládá se, že program v LabVIEW Real-Time uživatel nebude obsluhovat, tudíž je pouze možnost ukončit program za běhu tlačítkem *STOP*, jak lze vidět na obrázku 43.

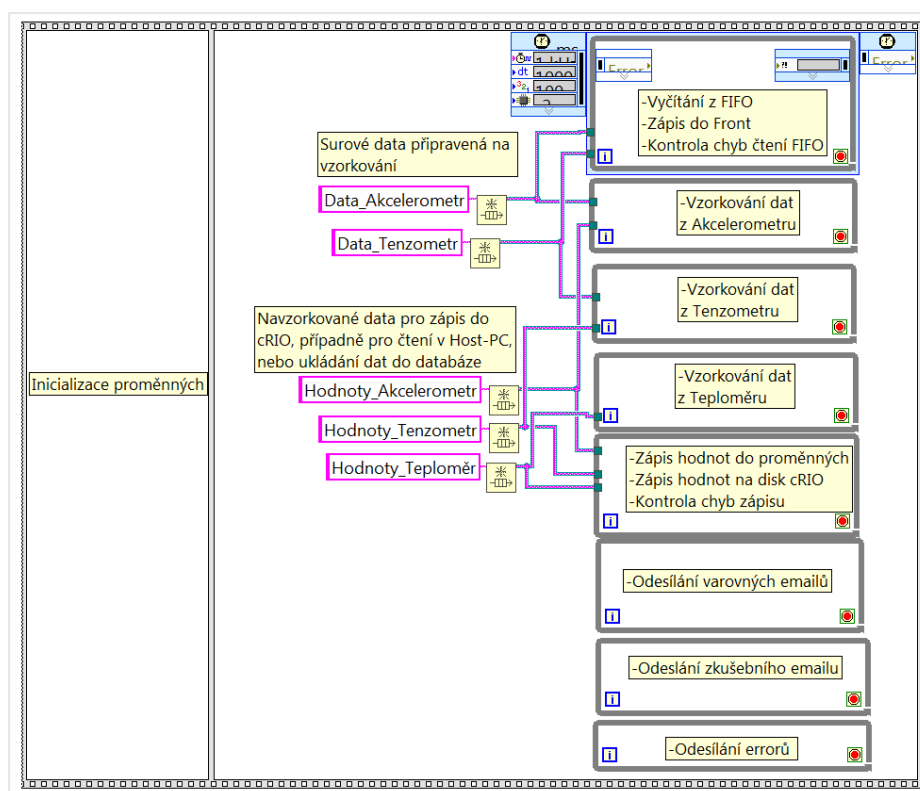


Obrázek 43: Uživatelské rozhraní Ovládání.vi

8.3.2 Struktura programu

Struktura programu (RT_Hlavní.vi) je značně složitější než struktura programu v Host-PC, i když uživatel nemá možnost ovládat program. Jeho složitost je dána do značné míry tím, že provádí většinu příkazů, které přicházejí z programu běžícího na Host-PC (Ovládání.vi).

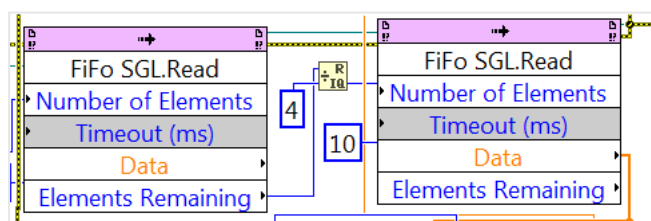
Na obrázku 44 je znázorněna struktura programu RT_Hlavní.vi.



Obrázek 44: Struktura programu RT_Hlavní.vi

Celý program je situován do Flat Sequence, čímž dojde k zajištění vykonávání příkazů postupně. Tato struktura se hodí pro inicializaci počátečních proměnných. Poté dojde k samotnému vykonávání programu.

Ve smyčce Timed Loop, která se nachází úplně nahoře v obrázku 43, dochází k čtení dat z FIFO paměti. Data do FIFO paměti jsou vkládána v programu běžícím v LabVIEW FPGA. Vyčtená data jsou poté vložena do dvou front, pro tenzometr a akcelerometr. Struktura zobrazená na obrázku 45 zajišťuje vyčtení celé FIFO paměti, čímž zabráňuje jejímu přetékání.

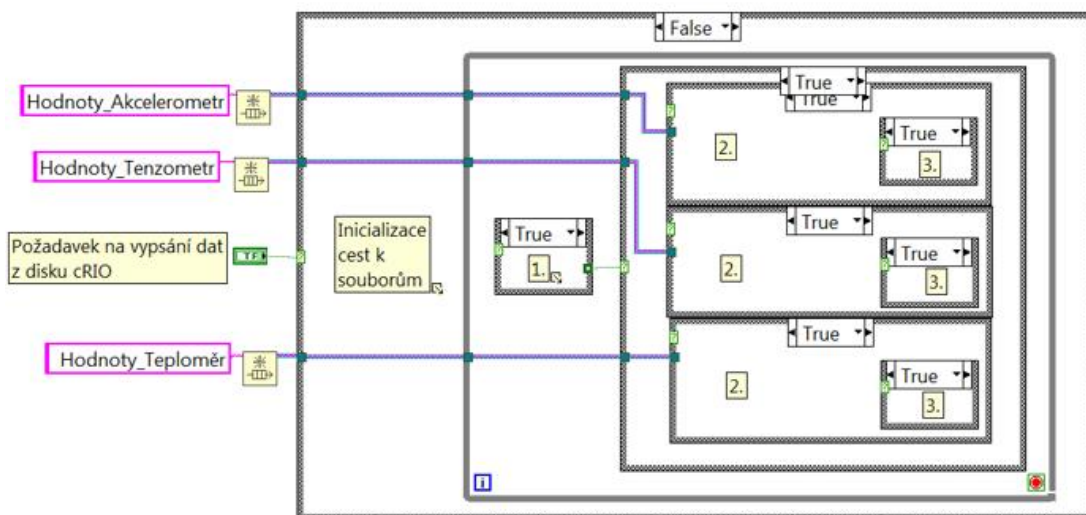


Obrázek 45: Vyčtení FIFO paměti

Dále je v této smyčce zbráněno předání dat z FIFO paměti do front, jestliže je na straně LabVIEW FPGA chyba.

Data z výše zmíněné smyčky putují pomocí front do dvou While Loops, které slouží k požadovanému navzorkování dat. Informace o vzorkovacích frekvencích přicházejí z Ovládání.vi. Data jsou navzorkována a jednotlivé hodnoty jsou odeslány pomocí další fronty do smyčky, která zpracovává data. Mezitím dochází i k vzorkování dat přicházejících z teploměru. Tato data jsou zpracovávána jinak než data z akcelerometru a tenzometru. Hlavní rozdíl je, že data z akcelerometru z tenzometru přichází z LabVIEW FPGA do LabVIEW Real-Time, zatímco data z teploměru jsou čtena pomocí Scan Enginu, přímo v LabVIEW Real-Time prostředí. Data z teploměru jsou čtena a po každé desáté přečtené hodnotě se ze všech devíti předcházejících udělá průměr a poté je tato hodnota poslána opět pomocí fronty do smyčky, která se stará o zpracování navzorkovaných dat.

Na obrázku 44 je zobrazena smyčka, která se stará o navzorkovaná data. Je v pořadí čtvrtá zespoda a obsahuje SubVI, jehož strukturu lze vidět na obrázku 46.



Obrázek 46: Správa na vzorkovaných dat (DataReadWrite2cRio (SubVI).vi) False

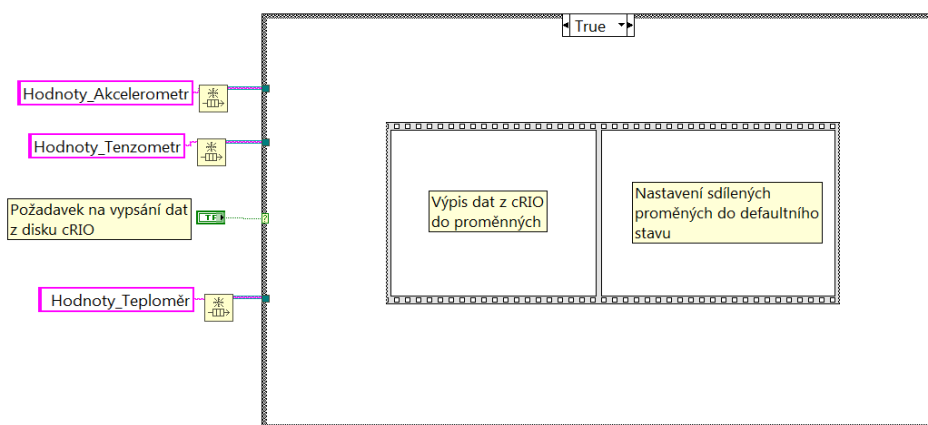
SubVI obsahuje hlavní Case Structure, která je přepínána na hodnotu True, nebo False v závislosti na požadavku z programu v Host-PC, zda mají být data vypásána z disku cRIO, nebo mají být data posílána real-time.

Jestliže nepřijde požadavek z Host-PC na zapsání uložených dat z cRIO do databáze, je While Loop defaultně nastavená na hodnotě False. Poté následuje inicializace cest k místům ukládání dat a smyčka While Loop, která probíhá kontinuálně, dokud nepřijde příkaz na vypsání dat z disku cRIO je zastavena a Case Structure se přepne na hodnotu True. Nyní následují menší Case Structures, které jsou obsaženy ve While Loop a jsou očíslovány.

Case Structure 1 slouží k rozhodování, zda mají být data vymazána z disku cRIO, anebo nikoliv. Jestliže mají být vymazána, je struktura přepnuta do stavu False a po vymazání dat se struktura opět přepíná do stavu True.

Case Structure 2 přijímá data a zapisuje je do sdílené proměnné, pokud fronta obsahuje data. Jestliže fronta neobsahuje data, je hodnota False a čeká se na data. Měřená data lze poté zobrazit v Ovládání.vi.

Case Structure 3 slouží k zapisování dat na disk cRIO. Zapisování dat se provádí pouze v tom případě, že je vydán příkaz pro zapisování dat z programu Ovládání.vi.



Obrázek 47: Správa na vzorkovaných dat (DataReadWrite2cRio (SubVI).vi) True

Na obrázku 47 lze vidět stejnou strukturu jako na obrázku 46, avšak pro stav True.

V Case Structure je pouze obsažena Flat Sequence, kde se prvně vyčtou data z disku cRIO, a zapíší se do sdílené proměnné. Poté se jiná sdílená proměnná nastaví do svého defaultního stavu False, tato sdílená proměnná se nazývá *VypsataData* a přepíná Case Structure na stavy True, a nebo False. Sdílená proměnná *VypisZcRIO* poté dá vědět programu Ovládání.vi, že data byla vypsána do sdílené proměnné a že se mohou se z ní začít zapisovat do databáze. Data ve sdílené proměnné jsou v podobě vektoru o X prvcích.

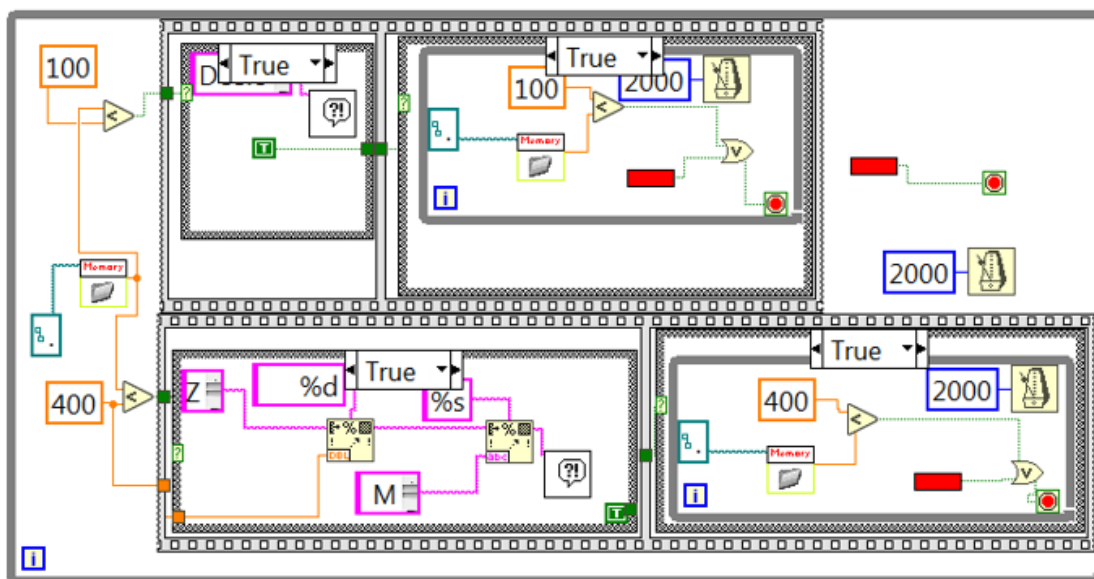
Obrázek 44 dále zobrazuje While Loop (třetí zespodu), která slouží k odesílání varovných e-mailů o stavu paměti cRIO. Tuto smyčku dále zobrazuje obrázek 48.

Jsou odeslány dva druhy e-mailů. První při zbývajícím volné paměti 400 MB varuje uživatele před docházejícím volným místem na disku cRIO a druhý e-mail při volné paměti 100 MB informuje o vymazání dat z paměti na disku cRIO.

V první fázi SubVI je zkontrolována paměť cRIO. Jestliže je vše v pořádku, program přejde do Flat Sequence a následující Case Structure budou nastaveny na hodnotu False. Tato akce se opakuje stále dokola.

Jestliže dojde k potřebě odeslat e-mail z důvodu nedostatku paměti na disku cRIO, Case Structure změní svůj stav z False na True a dojde k odeslání e-mailu o stavu paměti. Poté

následuje další Case Structure, která čeká, dokud nebude více místa na disku cRIO, aby nedocházelo k opětovným posíláním e-mailů se stejnou informací. Jestliže dojde ke zvětšení paměti nad požadované minimum, je ukončena While Loop v Case Structure a následuje opět celý proces hlídání stavu místa v paměti cRIO znovu.



Obrázek 48: Odesílání varovných e-mailů

Předposlední While Loop (viz. obrázek 44) slouží k odeslání zkušební e-mailu na určenou adresu. Odeslání e-mailu řídí program Ovládání.vi a slouží pro kontrolu správného nastavení e-mailu.

Poslední While Loop je jen pro kontrolu chyb v RT_Hlavni.vi a informace o chybách jsou odesílány pomocí sdílené proměnné do Ovládání.vi.

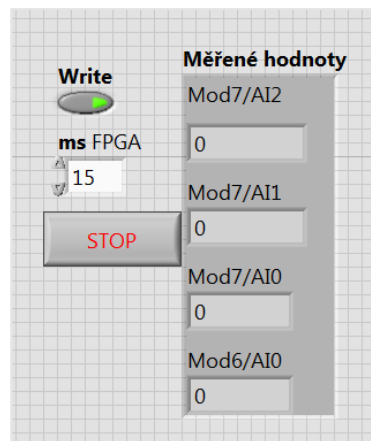
8.4 Program v LabVIEW FPGA

Program v LabVIEW FPGA se jmenuje Hlavní_FPGA.vi. Tento program je nejníže v měřicí struktuře a zajišťuje sběr dat ze senzorů a následné předávání dat do programu běžícím v LabVIEW Real-Time.

8.4.1 Uživatelské rozhraní programu

Uživatelské rozhraní je zde trochu obsáhlejší než u programu RT_Hlavní.vi. I zde je však počítáno s tím, že uživatel do programu bude zasahovat jen minimálně, jelikož po každé změně v programu je zapotřebí provést kompilaci, která zabírá řádově desítky minut na rozdíl od programu běžícím v LabVIEW Real-Time, který není potřeba kompilovat a případné přidání indicatoru pro sledování dat zde trvá řádově sekundy.

Na obrázku 49 je vyobrazeno uživatelské rozhraní programu Hlavní_FPGA.vi.



Obrázek 49: Uživatelské rozhraní Hlavní_FPGA.vi

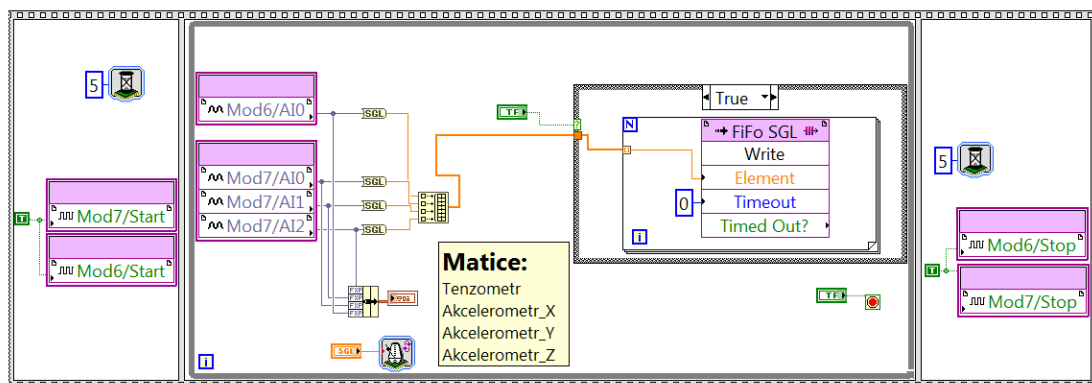
Na obrázku 49 lze vidět booleovský přepínač s názvem *Write*. Ten obsluhuje FIFO paměť, do které se mohou, ale nemusí, zapisovat hodnoty. Pod ním se nachází kontrolor *ms FPGA*, který řídí rychlost čtení dat ze senzorů v milisekundách. Jak lze na obrázku vidět, každých patnáct milisekund jsou přečtena data a zapsána do FIFO paměti. Pod ním následuje tlačítko *STOP*, které zastaví běžící program, a napravo Cluster *Měřené hodnoty*, kde má uživatel možnost si prohlížet aktuálně měřená data, která jsou čtena ze senzorů.

8.4.2 Struktura programu

Struktura programu (viz. obrázek 50) se skládá z Flat Sequence, kde nejprve dochází k vykonání příkazu pro C Series módy, aby začaly přijímat data ze senzorů. Ve struktuře FPGA jsou přijímána data z akcelerometru a tenzometru.

Následuje While Loop, kde dochází k přečtení hodnot ze senzorů, jejich následné zařazení do matice a jejich přetypování, které si lze dovolit udělat v FPGA v jednoduchých programech. Data v matici poté směřují do Case Struktury, která je zpracuje v případě, že přijde požadavek na zapisování dat do FIFO paměti. Jestliže požadavek přijde (hodnota True), data jsou z For Loop zapisována do paměti FIFO, která je poté přečtena v LabVIEW Real-Time programu RT_Hlavní.vi.

V poslední fázi Flat Structure dochází k předání příkazu C Series modulům, aby přestaly přijímat data ze senzorů.

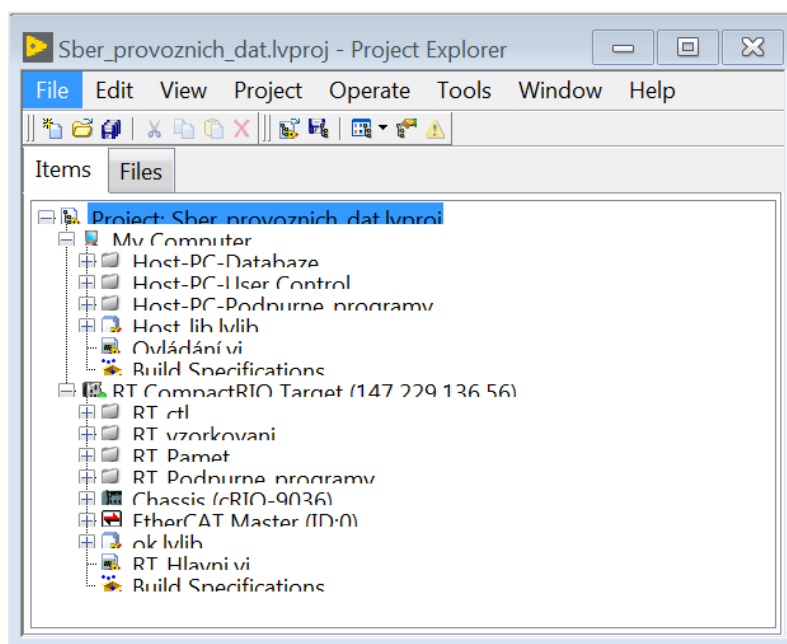


Obrázek 50: Struktura programu Hlavní_FPGA.vi

8.5 Struktura zdrojového kódu (projektu)

Projekt v LabVIEW slouží k přehlednější formě VI a SubVI, které jsou v dané aplikaci použity.

V projektu Sber_provoznich_dat.lvproj (viz. obrázek 51) jsou sdružena všechna VI, která byla v diplomové práci použita, a jsou členěna do složek podle jejich významu v projektu. Celkem se jedná o tři složky pro Host-PC a čtyři složky pro LabVIEW Real-Time. Vždy jedno VI zůstává nezařazeno. Toto VI představuje hlavní program v daném módu. Pro Host-PC jde o *Ovládání.vi*, pro LabVIEW Real-Time o *RT_Hlavní.vi* a pro LabVIEW FPGA o *Hlavní_FPGA.vi*. Přičemž po spuštění VI *RT_Hlavní.vi* lze celé měření kontrolovat pouze z VI *Ovládání.vi*, které běží na Host-PC.



Obrázek 51: Projekt: Sber_provoznich_dat.lvproj

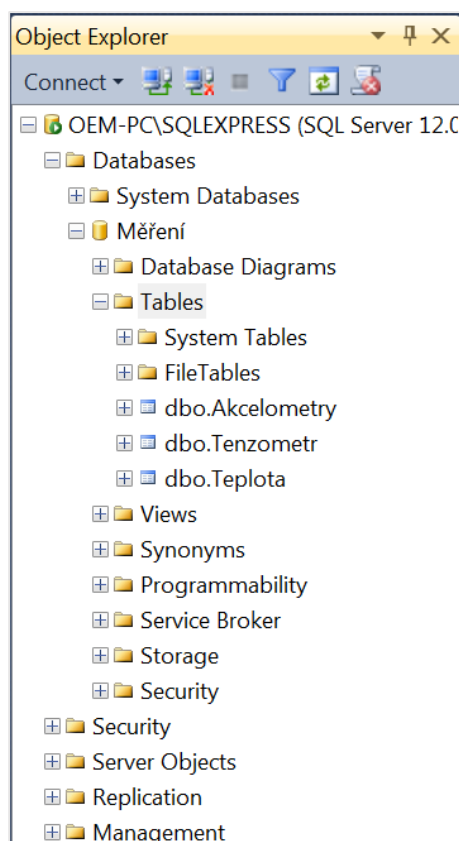
9 STRUKTURA DATABÁZE

Aby byla struktura vytvořeného programu kompletní, je potřeba zmínit vytvořený databázový systém, do kterého je možné data zapisovat, a tak je i uchovávat.

Byl zvolen databázový server MS SQL, ačkoli pro konkrétní zadání by se mohly použít i jiné databázové servery.

Přístup k databázovému serveru MS SQL zajišťuje Microsoft SQL server Managment Studio, jak již bylo psáno v kapitole 6.

V databázovém stromu byla vytvořena databáze s názvem Měření (viz. obrázek 52), která skýtá tabulky Akcelometry, Tenzometr a Teplota.



Obrázek 52: Databázový strom

Tabulky pro Akcelometr a Tenzometr, kde jsou data čtena v FPGA, mají strukturu odlišnou od tabulky pro teploty. Tabulka pro Tenzometr obsahuje celkem šest sloupců, které jsou v náhledu struktury tabulky členěny na řádky obrázek 53.

	Column Name	Data Type	Allow Nulls
▼	ID_Tenzometr	int	☐
	Data	float	☒
	Datum	varchar(200)	☒
	Čas	varchar(200)	☒
	Hz_FPGA	float	☒
	Hz_Požadované	float	☒
			☐

Obrázek 53: Tabulka pro data z tenzometru

V prvním sloupci obrázku 53 *Column Name*, jsou uvedeny názvy sloupců tabulky Tenzometr a ve druhém sloupci *Data Type* jsou jim přiřazeny datové typy. První sloupec *ID_Tenzometr* má přiřazen primární klíč, aby nedocházelo k duplicitě záznamů. Jelikož hodnoty mohou být pouze celá kladná čísla, je tomuto sloupci přiřazen datový typ *integer (int)*.

Druhý sloupec *Data* obsahuje vlastní měřená data. Protože hodnoty mohou nabývat desetinných čísel, je tomuto sloupci přiřazen datový typ *float*.

Pro sloupce *Datum* a *Čas* má databázový systém vlastní označení, a to *time (7)*, *datetime*, avšak byl zvolen datový typ *varchar* (Variable Character Field) s kterým se lépe pracuje.

Poslední dva sloupce tabulky Tenzometr, *Hz_FPGA* a *Hz_Požadované* přináší informaci o frekvenci, s jakou byla data čtena ze senzorů a informaci o požadované frekvenci vzorkování. Uživatel si díky těmto informacím může ověřit, zda nepožaduje vyšší frekvenci vzorkování, než kterou má FPGA. Tabulka pro akcelerometr vypadá obdobně, pouze má tři sloupce s naměřenými daty pro osy x, y a z.

Jelikož se čtení teplot liší od čtení hodnot z akcelerometru a tenzometru, je tabulka jinak situována (viz. obrázek 54).

	Column Name	Data Type	Allow Nulls
🔑	ID_teplosta	int	☐
	Data	float	☒
	Datum	varchar(200)	☒
	Čas	varchar(200)	☒
	S_Požadované	float	☒
	Hz_Požadované	float	☒
			☐

Obrázek 54: Tabulka pro data z teploměru

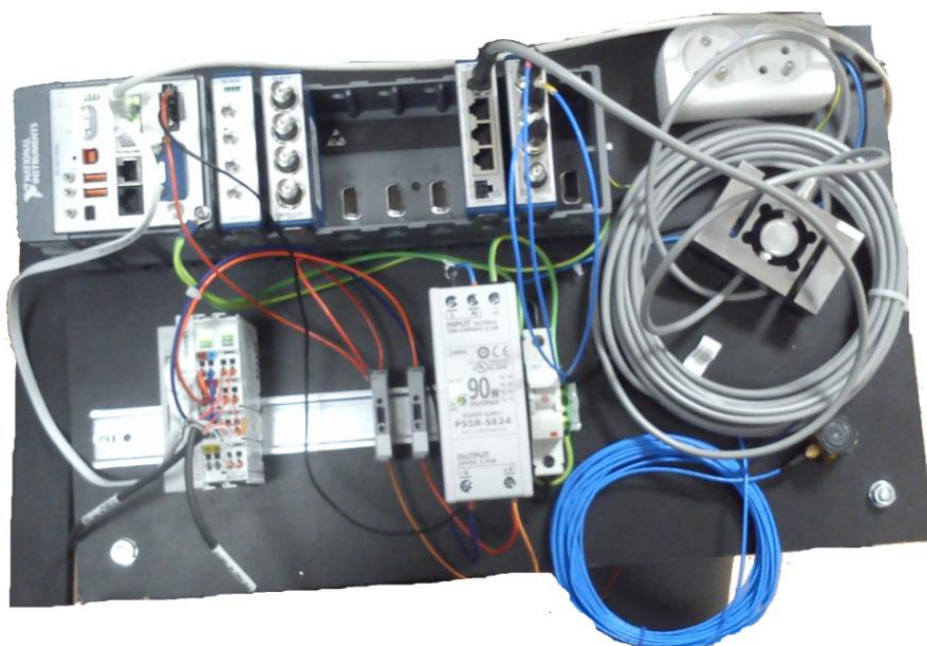
Tabulka se liší zejména v údaji *S_Požadované*, jelikož vzorkování se nastavuje v sekundách, nikoliv v hertzích. Vzorkování je požadováno v sekundách zejména z toho důvodu, že teplota stroje se nebude měřit v rámci milisekund, nýbrž sekund, kde je dále z deseti vzorků spočítán průměr a ten teprve zapsán do tabulky. Hodnota sloupce *Hz_Požadované* je pouze orientační, aby bylo zřejmé, kolika hertzům odpovídají nastavené sekundy. Jinak zbylé sloupce tabulky Teplota jsou stejné jako u předešlé tabulky.

V příloženém souboru lze nalézt složku s Excelem, kde bude nastavení tabulek exportováno z databázového serveru do Excelu.

10 OVĚŘENÍ FUNKČNOSTI PROGRAMU

Po vytvoření programu bylo zapotřebí ověřit jeho funkčnost v reálném prostředí. Měřicí soustava byla sestavena z částí, které jsou popisovány v kapitole *Použité měřící vybavení*. Po spuštění cRIO a programu *RT_Hlavní.vi* začne soustava měřit, avšak měřená data nejsou nikam odesílána, proto aby bylo možno data ukládat a provádět základní diagnostiku systému, je zapotřebí spustit na Host-PC program *Ovládání.vi* nebo EXE aplikaci *Ovladani.exe*, čímž dojde k připojení k programu běžícím na cRIO a poté lze začít s daty manipulovat.

Obrázek 55 zobrazuje soustavu, na které bylo prováděno měření. Senzory jsou v nezatíženém stavu, což není zcela ideální pro ověření funkčnosti, nicméně to nevádí, protože se senzorů přicházejí data i tak, i když v případě akcelerometru a tenzometru se jedná o šum, ale teploměr ukazuje reálné teplotní hodnoty, které byly měřeny v místnosti.



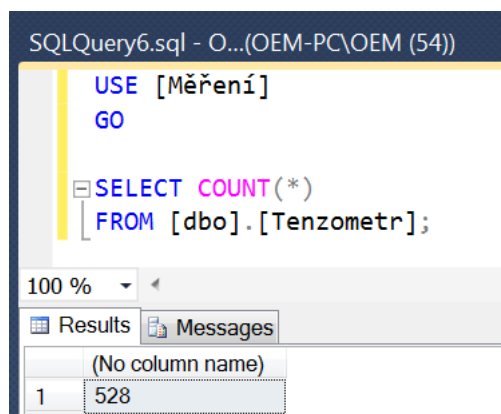
Obrázek 55: Měřicí soustava

Po spuštění všech programů, byly data pro akcelerometr a tenzometr zapsány na disk cRIO s frekvencí 1000 Hz, avšak program v LabVIEW FPGA běží na frekvenci 66,666 Hz (může běžet i rychleji, ale pro lepší představu je nastaveno takto), což odpovídá patnácti milisekundám, tudíž každých patnáct milisekund byla hodnota ze senzoru uložena do cRIO po uplynutí osmi sekund bylo měření zastaveno a data byla uložena do databáze. Zda data byla správně vyčítána a zapisována lze snadno ověřit ze známých časů.

Jestliže vyčítání z FIFO paměti v *RT_Hlavní.vi* probíhá každou sekundu a data jsou vyčítána ze senzorů, každých patnáct milisekund, tak vektor, který je vyčten v *RT_Hlavní.vi* by měl obsahovat přesně 66.666 vzorků, avšak reálně kolísá mezi 66 a 67 vzorky z důvodu nemožnosti vyčíst 0.666 vzorku.

Jestliže vzorkovací frekvence je požadována 1000 Hz a program v LabVIEW FPGA vyčítá data s frekvencí 66.66 Hz, tak musí být každý prvek, který je změřen zapsán na disk cRIO, což tedy odpovídá 66 až 67 prvkům zapsaným na disk každou sekundu. Nyní jestliže je vynásoben počet prvků krát počet sekund, po kterých bylo prováděno měření, tak výsledné číslo, odpovídá počtu prvků, které jsou zapsány v databázi. Tudíž 66 vzorků krát čas měření v sekundách, což bylo

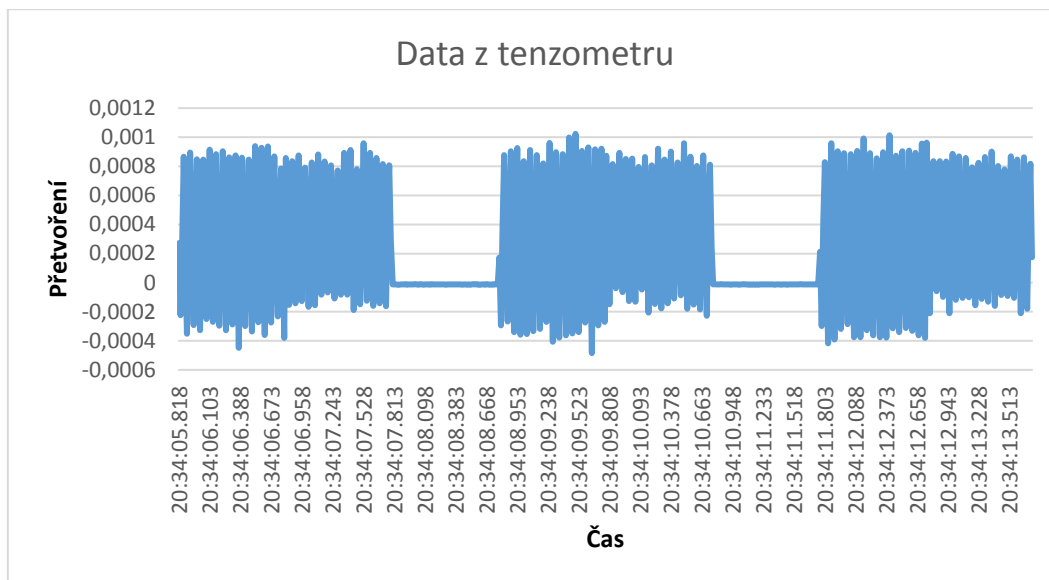
v tomto případě osm sekund, dává číslo 528. A nyní lze již snadno ověřit, zda je v databázi skutečně, zapsáno 528 vzorků pomocí SQL dotazu, který vrací hodnotu 528 (viz. obrázek 56).



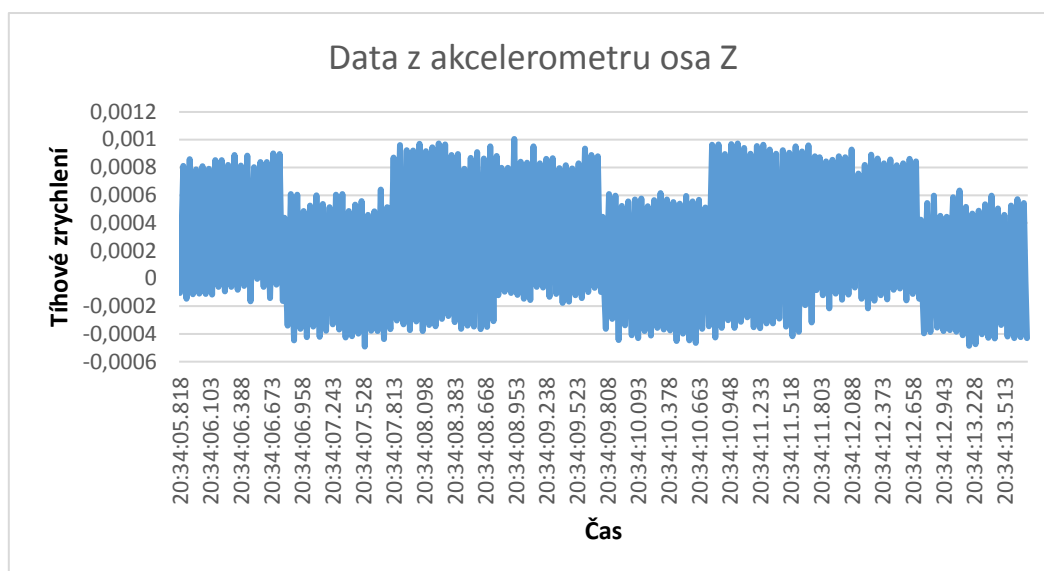
Obrázek 56: Počet řádků v tabulce

Tento výsledek dostáváme z tabulky pro tenzometr, avšak tabulka pro akcelerometr musí mít stejný počet prvků, protože data z tenzometru a akcelerometru jsou vyčítána se stejnou frekvencí. Požadovaná frekvence vzorkování, byla u obou případů nastavena stejně. Nyní pomocí SQL dotazu zmíněného na obrázku 56 se dá opět ověřit, že i tabulka pro akcelerometr má 528 řádků.

Výše popsaným příkladem bylo dosaženo ověření, že všechna data jsou zapsána na disk cRIO a následně do databáze, ale tím nelze ještě zcela říct, že program zpracovává data správně, protože je možné, že se někde zacyklil a vyčítá ty samé data stále znovu. Aby se ověřilo, že data nejsou duplicitní, tak tabulky s daty akcelerometru a tenzometru byly exportovány do Excelu, kde byl následně z hodnot vytvořen graf pro hodnoty z tenzometru a pro hodnoty z akcelerometru pro osy X Y a Z, přičemž, jestliže jedna osa nemá duplicitní data, tak je nemají ani ostatní, proto zde je zobrazen pouze jeden graf pro osu Z akcelerometru.



Obrázek 57: Data z tenzometru

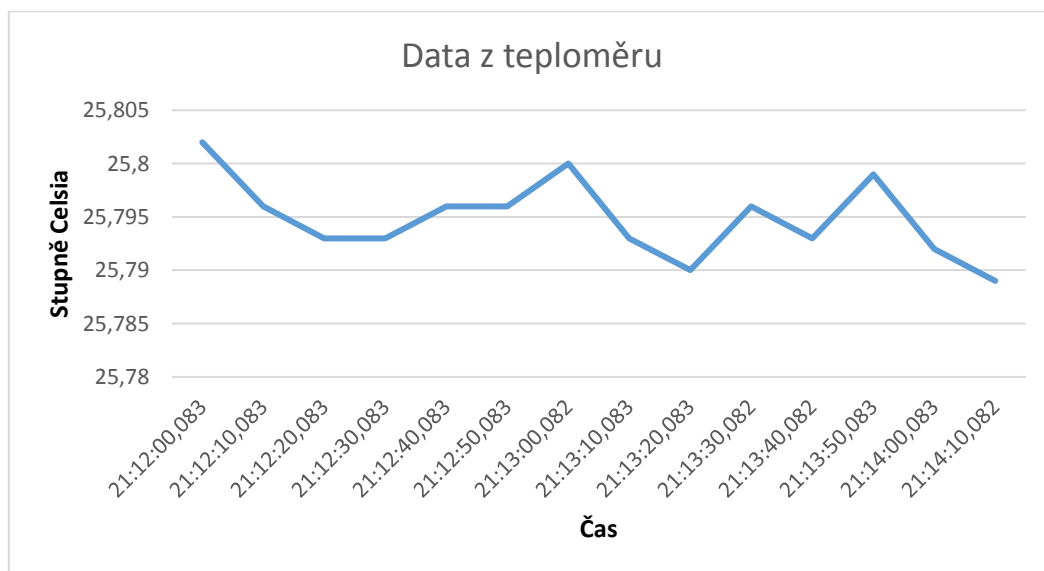


Obrázek 58: Data z akcelerometru

Jelikož data byla měřena v nezatíženém stavu, tak se budou velice podobat, ale i přesto lze z grafu vyčíst, že změřená data nejsou vyčítány duplicitně.

Jediný senzor, který byl zatížen způsobem, s kterým je počítáno i v reálné aplikaci byl teploměr, který v průběhu deseti sekund sbírá údaje o teplotě a následně je průměruje. Opět byla tato data zapsána na databázový server.

Jestliže měření trvalo 140 sekund, tak by v databázi mělo být opět zapsáno čtrnáct prvků. To lze ověřit obdobným způsobem jako na obrázku 56. Aby se ověřilo, že nejsou data duplicitní, tak byly hodnoty exportovány do Excelu a byl zhotoven graf.



Obrázek 59: Data z teploměru

Teploty v místnosti kolísají mezi 25,805 °C a 25,78 °C, avšak nejsou opět vyčítaná duplicitně. Z výše zmíněných ověření, lze konstatovat, že program, vyčítá data ze senzorů, dále je zapisuje na disk cRIO a následně do databáze správně.

Druhou možností, kterou program nabízí je zapisování dat do databáze v reálném čase. Zde již není vše optimální jak u zápisu dat na disk cRIO, protože posíláním jednotlivých dat přes veřejnou síť a následné jejich zapisování na databázový server není zcela možné pro vyšší frekvence vzorkování. Některá data, se během zápisu do databáze ztrácejí, proto není vhodné při vyšších frekvencích zapisovat přímo data na databázový server.

11 ZÁVĚR

Cílem diplomové práce bylo navrhnout systém pro sběr provozních dat obráběcího stroje, proto se práce v rešeršní části zabývala především historií obráběcích strojů a jejich rozdělením, aby byl vstup do problematiky snazší. Poté navazovala kapitola, která přibližovala základní možné měřitelné fyzikální veličiny a jejich vlivy na obráběcí stroj. Dále byly zmiňovány senzory, kterými se dají jednotlivé veličiny měřit. Senzory pro měření proudu a napětí, které nejsou v této diplomové práci řešeny, byly doplněny v rámci všeobecného přehledu. Poslední dvě kapitoly rešeršní části jsou věnovány teoretické přípravě na praktickou část diplomové práce. Jednak jsou zde uvedeny možnosti zpracování dat ze senzorů, jednak teoretické pojednání o databázových systémech.

Praktická část je již zaměřena na samotné provedení systému pro sběr provozních dat, kde jsou nejprve popsány jednotlivé prvky, které byly před zhotovením měřicího systému zapotřebí. Další část pojednává o samotném programu, kde je snaha co nejvíce přiblížit architekturu systému čtenáři, a nakonec je provedena analýza funkčnosti programu.

Zadání diplomové práce, které se týkalo zhotovení systému pro sběr dat s následným ověřením funkčnosti se povedlo splnit, avšak jak bylo popisováno v kapitole *Ověření funkčnosti programu*, nebyla měřicí soustava s programem testována na obráběcím stroji. Nicméně jak již bylo psáno, tak to nevádí, protože byla testována v nezátíženém stavu. Zde byla provedena analýza hodnot z měření a na základě téhle analýzy bylo konstatováno, že program pro mód zápisu dat na disk cRIO a jejich následné vyčtení na databázový server zvládá zápis bez ztráty dat. Ale bylo také konstatováno, že zápis aktuálně měřených hodnot do databáze, již není stoprocentní při vyšších frekvencích vzorkování a dochází ke ztrátě dat. Ke ztrátě dat dochází zejména ze dvou důvodů. Tím prvním důvodem je posílání dat přes veřejnou síť, která nemusí mít takovou odezvu, jaká je požadována. Druhým důvodem je samotný zápis dat na databázový server, počítač nemusí být natolik výkonný, aby byl schopen stíhat zapisovat data. Tyhle problémy lze řešit buď přímým napojením cRIO na Host-PC, přes ethernetový kabel, přičemž je potřeba se ujistit, že počítač je schopen zpracovat data a zapisovat je do databáze v požadovaném čase. Dále lze také zapisovat měřená data do TDSM souboru a po nějakém čase nebo po určitém počtu vzorků, data vyčíst z TDSM souboru a vypsát je na databázový server. Nejednalo by se sice o zápis aktuálně měřených dat, ale bylo zajištěno vypsání všech dat v průběhu měření na databázový server, tím by se data neztrácela.

Uživatelské rozhraní, pro ovládání programu v *Ovládání.vi* nebo *Ovládání.exe*, podle toho, zda je program spouštěn přes strom projektu anebo přes EXE aplikaci, bylo doplněno o více možností ovládání pro uživatele, než bylo zadáno v zadání diplomové práce. Zejména o různé kontrol prvky a nastavení, avšak i zde je prostor pro rozšíření, jak stávající aplikace, tak vytvoření nové aplikace, která by dokázala zpracovávat data z databázového serveru. Například data z databázového serveru zobrazovat v grafech anebo pokročileji data zpracovávat a poté je zobrazovat v grafech. Také by se mohla přidat funkce pro mazání dat na databázovém serveru, ale je otázkou, zda by měl uživatel mít možnost provádět jakékoliv úpravy na databázovém serveru.

I když jsou možnosti, jak program zlepšit, tak stávající program zabezpečuje vyčítání dat ze senzorů a jejich ukládání na databázový server bezpečně, a proto je vhodný ke zaznamenávání dat ze senzorů, které mohou následně sloužit k diagnostice měřeného systému.

12 SEZNAM ZDROJŮ

- [1] MAREK, Jiří. *Konstrukce CNC obráběcích strojů III*. Praha: MM publishing, s.r.o., 2014. MM speciál. ISBN 978-80-260-6780-1.
- [2] *Historie CNC strojů* [online]. b.r. [cit. 2017-12-19]. Dostupné z: <https://coptkm.cz/portal/reposit.php?action=0&id=22926&instance=2>
- [3] MAREK, Tomáš. *Možnosti využití vysokorychlostního obrábění při výrobě dílů pro stavbu obráběcích strojů*. Ústav techniky a automobilové dopravy (AF), 2012. Bakalářská práce. Mendelova univerzita v Brně. Vedoucí práce Doc. Ing. Vlastimil Chrást, CSc.
- [4] Řešení tepelných deformací obráběcích strojů. *Mmspektrum* [online]. [cit. 2017-12-19]. Dostupné z: <https://www.mmspektrum.com/clanek/reseni-tepelnych-deformaci-obrabecich-stroju.html>
- [5] HORT, Martin. *Technologické, ekonomické a ekologické aspekty obrábění s vysoce výkonným chlazením* [online]. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2009 [cit. 2017-12-19]. Dostupné z: <http://hdl.handle.net/11012/14261>. Diplomová práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav strojírenské technologie. Vedoucí práce Miroslav Piška.
- [6] *Systémové pojetí obráběcího stroje– kmitání* [online]. 2011 [cit. 2017-12-19]. Dostupné z: <http://www.cnckonstrukce.cz>
- [7] Všeobecně o vibracích v obráběcích strojích (6). *T-support* [online]. Praha, 2017 [cit. 2017-12-19]. Dostupné z: <https://www.t-support.cz/kat/vseobecne-o-vibracich-v-obrabecich-strojich-6>
- [8] HUSAR, Tomáš. *Měření vibrací na obráběcích strojích* [online]. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2015 [cit. 2017-12-19]. Dostupné z: <http://hdl.handle.net/11012/40110>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav automobilního a dopravního inženýrství. Vedoucí práce Jan Vetiška.
- [9] DVOŘÁK, Jan. *Vibrace při frézování kovů* [online]. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2016 [cit. 2017-12-19]. Dostupné z: <http://hdl.handle.net/11012/60094>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav strojírenské technologie. Vedoucí práce Zdeněk Fiala.
- [10] MÍČKA, Martin. *Deskripce rámců soustružnických strojů* [online]. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2008 [cit. 2017-12-20]. Dostupné z: <http://hdl.handle.net/11012/4714>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Petr Blecha.

- [11] MICHALÍČEK, Michal. *PREDIKCE PRACOVNÍ PŘESNOSTI CNC OBRÁBĚCÍCH STROJŮ* [online]. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2013 [cit. 2017-12-20]. Dostupné z: <http://hdl.handle.net/11012/28000>. Disertační práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Jiří Marek.
- [12] BOUCHALOVÁ, Monika. *Struktura a vlastnosti pevných látek* [online]. In: . Havířov, b.r. [cit. 2017-12-20]. Dostupné z: <http://slideplayer.cz/slide/2341000>
- [13] BORSKÝ, Václav. *Základy stavby obráběcích strojů*. 2. přeprac. vyd. Brno: Vysoké učení technické, 1991.
- [14] HOUŠKA, Pavel. *Senzorika a prvky umělé inteligence: Základní vlastnosti snímačů Soubor*. In: *Vysoké učení technické v Brně* [online]. Brno, b.r. [cit. 2018-02-22]. Dostupné z: https://moodle.vutbr.cz/pluginfile.php/359563/mod_resource/content/1/pred-01.pdf
- [15] HOUŠKA, Pavel. *Senzorika a prvky umělé inteligence: Základní vlastnosti snímačů Soubor* [online]. Brno, b.r. [cit. 2018-02-22]. Dostupné z: https://moodle.vutbr.cz/pluginfile.php/359563/mod_resource/content/1/pred-01.pdf
- [16] HOUŠKA, Pavel. *Senzorika a prvky umělé inteligence: Snímače teploty* [online]. Brno, b.r. [cit. 2018-02-22]. Dostupné z: https://moodle.vutbr.cz/pluginfile.php/359565/mod_resource/content/1/pred-03.pdf
- [17] Přehled principů el. měření teploty - 1. díl. *Automatizace.hw.cz* [online]. HW server s.r.o., 2014 [cit. 2017-12-26]. Dostupné z: <https://automatizace.hw.cz/prehled-principu-el-mereni-teploty-1-dil>
- [18] Přehled principů el. měření teploty - 2. díl - bezdotykové. *Automatizace.hw.cz* [online]. HW server s.r.o., 2014 [cit. 2017-12-26]. Dostupné z: <https://automatizace.hw.cz/rehled-principu-el-mereni-teploty-2-dil-bezdotykove>
- [19] Odporové tenzometry - princip, provedení, použití, historie. *Automatizace.hw.cz* [online]. HW server s.r.o, 2014 [cit. 2017-12-27]. Dostupné z: <https://automatizace.hw.cz/foliove-odporove-tenzometry-princip-provedeni-pouziti-historie.html>
- [20] Tlak tlakoměry elektrické. *E-automatizace* [online]. b.r. [cit. 2017-12-27]. Dostupné z: http://www.e-automatizace.cz/ebooks/mmv/tlak/tlak_tlakomery_elektricke.htm
- [21] JIRÁSEK, Milan, Vít ŠMILAUER a Jan ZEMAN. *Pružnost, pevnost, plasticita*. Praha, 2017. Pracovní verze výukového skriptu. České vysoké učení technické.
- [22] VACEK, Vladimír. *Přehled senzorů pro měření vibrací*. Plzeň, 2012. Bakalářská práce. Západočeská univerzita. Vedoucí práce Ing. Zuzana Kabešová.
- [23] SODOMKA, Pavel. *Měření úhlové rychlosti optovláknovým interferometrem*. Brno, 2012. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Zdeněk Havránek, Ph.D.

- [24] Měřidla a měření. *Elektřina* [online]. CEZ, b.r. [cit. 2018-01-11]. Dostupné z: <https://www.cez.cz/edee/content/microsites/elektrina/2-4.htm>
- [25] Integrované senzory proudu a problematika použití. *Automatizace.hw* [online]. HW server s.r.o., 2014 [cit. 2018-01-11]. Dostupné z: <https://automatizace.hw.cz/integrované-senzory-proudu-a-problematika-pouziti-1cast>
- [26] *Programovatelné logické automaty* [online]. [cit. 2018-05-03]. Dostupné z: <http://uprt.vscht.cz/kminekm/mrt/f5/f5k53-plc.htm>
- [27] Which NI Platform is Most Appropriate for My Automated Test System Needs?. *Viewpoint System* [online]. Viewpoint Systems, ©2018 [cit. 2018-04-20]. Dostupné z: <https://www.viewpointusa.com/test-measurement/which-ni-platform-is-most-appropriate-for-my-test-needs/>
- [28] In: *National Instruments* [online]. ©2018 [cit. 2018-04-21]. Dostupné z: <http://www.ni.com/tutorial/54281/en/>
- [29] *National Instruments* [online]. National Instruments, ©2018 [cit. 2018-04-21]. Dostupné z: <http://www.ni.com/tutorial/11197/en/>
- [30] HERCEG, Tomáš a Tomáš JECHA. *DOTNETPORTAL: lehký úvod – teorie databází* [online]. ©2018 [cit. 2018-05-05]. Dostupné z: <https://www.dotnetportal.cz/clanek/60/Lehky-uvod-teorie-databazi>
- [31] HERCEG, Tomáš. Úvod do jazyka sql. *DotNETportal* [online]. 2018 [cit. 2018-04-22]. Dostupné z: <https://www.dotnetportal.cz/clanek/50/Uvod-do-jazyka-SQL>
- [32] *NI cRIO-9036: USER MANUAL* [online]. National Instruments, 2017 [cit. 2018-04-23].
- [33] *NI 9237: Datasheet* [online]. National Instruments, 2015 [cit. 2018-05-06]. Dostupné z: http://www.ni.com/pdf/manuals/374186a_02.pdf
- [34] *NI 9234: Datasheet* [online]. National Instruments, 2015 [cit. 2018-05-06]. Dostupné z: http://www.ni.com/pdf/manuals/374238a_02.pdf
- [35] *EK110x, EK15xx: Datasheet* [online]. BECKHOFF, 2017 [cit. 2018-05-06]. Dostupné z: https://download.beckhoff.com/download/document/io/ethercat-terminals/ek110x_ek15xxen.pdf

13 SEZNAM OBRÁZKŮ

Obrázek 1: Rozdělení CNC obráběcích strojů z hlediska vykonávané práce [3].....	17
Obrázek 2: Návaznost systému na CIM [2]	18
Obrázek 3: Nejistota výroby a pracovní nejistota [1].....	20
Obrázek 4: Oblasti vzniku tepla při třískovém obrábění [5]	22
Obrázek 5: Oblasti vzniku tepla při třískovém obrábění [5]	22
Obrázek 6: Příčiny kmitání v obráběcích strojích [6]	23
Obrázek 7: Kmitání s periodickým buzením.....	25
Obrázek 8: Průběh sinusové křivky.....	25
Obrázek 9: Hookův zákon [12]	28
Obrázek 10: Schéma senzory [14].....	30
Obrázek 11: Charakteristika odporových senzorů [17].....	32
Obrázek 12: Závislost vlnových délek na spektrálním vyzařování [18]	33
Obrázek 13: Základní veličiny a rovnice pro rovnoměrně tažený prut [21]	34
Obrázek 14: Princip piezoelektrického senzory [20]	35
Obrázek 15: Vyobrazení průběhu zatížení zachycené piezometrem.....	35
Obrázek 16: Sagnacův interferometr [23]	36
Obrázek 17: Bloková struktura programovatelného automatu [26]	39
Obrázek 18: Rozdíly mezi systémy National Instruments [27].....	40
Obrázek 19: CompactRIO [28]	41
Obrázek 20: Architektura cRIO [29].....	41
Obrázek 21: Programovatelné moduly cRIO [29].....	42
Obrázek 22: LabVIEW pro Host-PC.....	43
Obrázek 23: LabVIEW Real-Time.....	44
Obrázek 24: LabVIEW FPGA.....	45
Obrázek 25: LabVIEW FPGA – Kompilační okno.....	45
Obrázek 26: Hierarchická databáze [30]	47
Obrázek 27: SQL Databáze	48
Obrázek 28: NI cRIO-9036 [32].....	50
Obrázek 29: Controller cRIO-9036 [32]	51
Obrázek 30: Schéma cRIO-9036 [32]	51
Obrázek 31: NI-9237 [33]	52
Obrázek 32: NI-9234 [34]	53
Obrázek 33: Analogový vstup [34]	53
Obrázek 34: Backhoff EK1100 [35].....	54
Obrázek 35: Tok naměřených hodnot	56
Obrázek 36: Komunikace mezi vrstvami	57
Obrázek 37: Kontrola nastavení Host-PC programu.....	58
Obrázek 38: Nastavení měření Host-PC programu	59
Obrázek 39: Zobrazení měření Host-PC programu.....	60
Obrázek 40: Záložka cRIO/Host-PC	61
Obrázek 41: Struktura programu Ovládání.vi	62
Obrázek 42: Ukázka While Loop	63

Obrázek 43: Uživatelské rozhraní Ovládání.vi	64
Obrázek 44: Struktura programu RT_Hlavní.vi	64
Obrázek 45: Vyčtení FIFO paměti	65
Obrázek 46: Správa na vzorkovaných dat (DataReadWrite2cRio (SubVI).vi) False ..	65
Obrázek 47: Správa na vzorkovaných dat (DataReadWrite2cRio (SubVI).vi) True	66
Obrázek 48: Odesílání varovných e-mailů	67
Obrázek 49: Uživatelské rozhraní Hlavní_FPGA.vi	68
Obrázek 50: Struktura programu Hlavní_FPGA.vi	68
Obrázek 51: Projekt: Sber_provozních_dat.lvproj	69
Obrázek 52: Databázový strom	70
Obrázek 53: Tabulka pro data z tenzometru	70
Obrázek 54: Tabulka pro data z teploměru	71
Obrázek 55: Měřicí soustava	72
Obrázek 56: Počet řádků v tabulce	73
Obrázek 57: Data z tenzometru	73
Obrázek 58: Data z akcelerometru	74
Obrázek 59: Data z teploměru	74

14 SEZNAM PŘÍLOH

Příl. 1. Databázová_struktura	Datový nosič
Příl. 2. LabVIEW_Program.....	Datový nosič