

# **Řízení zkušebního zařízení pro mechanické zatěžování materiálů pro opravu komunikací**

Control of equipment for mechanical endurance testing of materials for reparation of roadways

Bc. Tomáš Hrubý

---

Diplomová práce  
2011



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2010/2011

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš HRUBÝ**  
Osobní číslo: **A07694**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Automatické řízení a informatika**

Téma práce: **Řízení zkušebního zařízení pro mechanické zatěžování materiálů pro opravu komunikací**

Zásady pro vypracování:

1. Vytvořte řídicí aplikaci zkoušky asfaltových zálivek.
2. HW řešení aplikace vyřešte pomocí FPGA karty.
3. Realizujte v prostředí LabView 8.2.
4. Realizujte na principu generování řídicích pulsů pro krokový motor za současného snímání tlakové nebo tahové síly.
5. Uvažujte různé režimy průběhů(cykly) řízení podle síly nebo podle deformace.
6. Vytvořte uživatelské rozhraní aplikace s možností archivace dat a grafického záznamu průběhu.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

1. VLACH, Jaroslav, HAVLÍČEK, Josef, VLACH, Martin. Začínáme s LABVIEW. Praha : BEN – technická literatura, 2008. 248 s. ISBN 978-80-7300-245-9.
2. BALÁTĚ, Jaroslav. Automatické řízení. 2. aktualiz. vyd. Praha : BEN – technická literatura, 2004. 664 s. ISBN 80-7300-148-9.
3. LabVIEW :FPGA Interface User Guide. Austin: National Instruments Corporation, 2004. 5 s.
4. LabVIEW :FPGA Module User Manual. Austin: National Instruments Corporation, 2004. 62 s.
5. LabVIEW :Real-Time Module User Manual. Austin: National Instruments Corporation, 2004. 97 s.

Vedoucí teoretické části:

doc. Ing. Marek Kubalčík, Ph.D.

Ústav řízení procesů

Vedoucí praktické části:

Ing. Roman Dlabaja, Ph.D.

Datum zadání diplomové práce:

25. února 2011

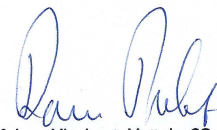
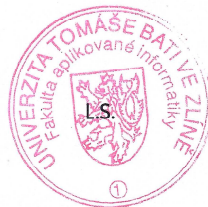
Termín odevzdání diplomové práce:

3. června 2011

Ve Zlíně dne 25. února 2011



prof. Ing. Vladimír Vašek, CSc.  
děkan



prof. Ing. Vladimír Vašek, CSc.  
ředitel ústavu

## **ABSTRAKT**

Cílem diplomové práce bylo vytvořit aplikaci ve vývojovém prostředí LabVIEW pro test asfaltových zálivek dilatačních spár používaných v dopravních stavbách při napojení mostní konstrukce s pozemní komunikací. Program měl umožňovat dvě zkoušky. V prvním testu se asfaltová zálivka namáhá při konstantní tlakové nebo tahové síle a sleduje se deformace. Ve druhém testu se po dosažení dané deformace sleduje průběh síly. Důraz je kladen také na archivaci dat k dalšímu zpracování a následné analýze. Soustava krokového motoru s řídicí jednotkou a snímačem tlaku je připojena k PC přes FPGA kartu pomocí Real Time aplikace.

Klíčová slova: LabVIEW, asfaltová zálivka, FPGA, Real Time

## **ABSTRACT**

Goal of diploma work was create application in developmental environment LabVIEW for test compound dilatation join used in the traffic constructions at inosculation bridge structure. Application has possible two tests. At first test compound subject to constant compressive or tension power and traces deformation. In second test to the achievement given to deformation traces process force. Emphasis is laying too on archiving of dates to the next processing and resulting analysis. System stepping motor with control unit plus sensor pressure is connected to PC over FPGA card by the help of Real Time application.

Keywords: LabVIEW, compound, FPGA, Real Time

Tímto bych chtěl poděkovat vedoucímu mé diplomové práce panu Doc. Ing. Markovi Kubalčíkovi Ph. D. za odborné vedení, rady, připomínky a hlavně trpělivost. To vše mi poskytl během vedení mé diplomové práce. Také bych rád poděkoval panu Mgr. Romanovi Dlabajovi Ph.D. z firmy ITC Zlín a.s. za cenné rady z praxe a pomoc při programování v LabVIEW. V neposlední řadě děkuji za trpělivost a podporu také mé ženě Pavlíně se synem Štěpánem.

*Motto*

*Zatímco ztrácíme svůj čas váháním a odkládáním, život utíká.*

*Seneca*

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>10</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 POPIS ZKUŠEBNÍ SESTAVY</b> .....	<b>12</b>
1.1 KROKOVÝ MOTOR .....	13
1.1.1 Obecný popis .....	13
1.1.2 Technické údaje.....	14
1.1.3 Charakteristiky krokového motoru.....	15
1.2 OVLADAČ KROKOVÉHO MOTORU.....	16
1.2.1 Základní vlastnosti .....	16
1.2.2 Vstupní a výstupní logické signály.....	17
1.2.3 Výkonné vstupy a výstupy.....	19
1.2.4 Hlavní nastavení podle návodu výrobce .....	19
1.2.5 LED indikace stavu .....	20
1.2.6 Schéma připojení ovladače.....	21
1.2.7 Regulovatelný zdroj MDR-60.....	22
1.3 SNÍMAČ SÍLY .....	23
1.4 REDUKCE NI SCB-68.....	24
1.5 KABEL SHC68-68 RMIO.....	26
1.6 NI PCI-7831R.....	28
1.6.1 Obecný popis.....	28
1.6.2 Vlastnosti karty .....	30
1.7 REAL-TIME PC.....	31
1.8 HOST PC .....	34
1.9 VZOREK – ASFALTOVÁ ZÁLIVKA.....	35
<b>2 ČESKÁ TECHNICKÁ NORMA PRO ASFALTOVÉ ZÁLIVKY</b> .....	<b>36</b>
2.1 OPAKOVANÉ PROTAHOVÁNÍ A STLAČOVÁNÍ .....	36
2.2 PŘERUŠENÉ PROTAŽENÍ.....	36
<b>3 PROSTŘEDÍ LABVIEW</b> .....	<b>37</b>
3.1 POJEM LABVIEW.....	37
3.2 PO SPUŠTĚNÍ LABVIEW .....	38
3.3 UŽIVATELSKÉ PROSTŘEDÍ.....	40
3.4 ČELNÍ PANEL (FRONT PANEL).....	40
3.5 BLOKOVÝ DIAGRAM (BLOCK DIAGRAM).....	41
3.6 PALETY NÁSTROJŮ .....	43
3.6.1 Paleta nástrojů (Tools) .....	43
3.6.2 Paleta ovládacích a zobrazovacích prvků (Controls) .....	44
3.6.3 Paleta Functions .....	45

3.7	CONTEXT HELP .....	46
3.8	DATOVÉ TYPY .....	47
3.9	PROGRAMOVÉ STRUKTURY .....	49
3.9.1	Cyklus While .....	49
3.9.2	Cyklus For .....	49
3.9.3	Posuvný registr .....	50
3.9.4	Větvení Case .....	50
3.9.5	Sequence .....	51
3.10	GRAFICKÉ ZOBRAZOVAČE .....	51
3.11	TOK DAT V PROGRAMU .....	52
3.12	DAQ .....	53
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>54</b>
<b>4</b>	<b>POPIS ŘÍDÍCÍ A VIZUALIZAČNÍ APLIKACE PRO ZKOUŠKY ASFALTOVÝCH ZÁLIVEK .....</b>	<b>55</b>
4.1	VOLBA ULOŽENÍ DO SOUBORU .....	55
4.2	VOLBA MANUÁLNÍHO POSUNU MOTORU .....	56
4.3	ZKOUŠKA 1 .....	56
4.4	ZKOUŠKA 2 .....	56
4.5	GRAFY .....	57
<b>5</b>	<b>NÁVRH A POPIS JEDNOTLIVÝCH ČÁSTÍ PROJEKTU V NI LABVIEW .....</b>	<b>58</b>
5.1	HOST PC (MY COMPUTER) .....	59
5.1.1	My Computer .....	59
5.1.2	HOSTCode.vi .....	59
5.1.2.1	Základní nastavení .....	60
5.1.2.2	Stavy tlačítek (událostní struktura) .....	60
5.1.2.3	Zobrazení grafů .....	64
5.1.2.4	Zápis do souboru .....	65
5.2	REAL-TIMEPC .....	65
5.2.1	RTCode.vi .....	66
5.2.1.1	Komunikace s FPGA kartou (PCI-7831R) .....	66
5.2.1.2	Stavový automat .....	68
5.2.2	FPGACode.vi .....	73
5.2.2.1	Smyčka čtení AI0 .....	74
5.2.2.2	Smyčka zápisu na DIO0 a DIO1 .....	74
5.3	SUB VI - PODPROGRAMY .....	75
5.4	SDÍLENÉ PROMĚNNÉ .....	75
<b>6</b>	<b>VÝSLEDKY Z NAMĚŘENÝCH ZKOUŠEK .....</b>	<b>77</b>

---

6.1	GRAFY ZKOUŠKY 1 .....	77
6.2	GRAFY ZKOUŠKY 2 .....	78
<b>ZÁVĚR</b>	.....	<b>79</b>
<b>ZÁVĚR V ANGLIČTINĚ</b>	.....	<b>80</b>
<b>SEZNAM POUŽITÉ LITERATURY</b>	.....	<b>81</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ, ZKRATEK A ODBORNÝCH VÝRAZŮ</b>	.....	<b>82</b>
<b>SEZNAM OBRÁZKŮ</b>	.....	<b>83</b>
<b>SEZNAM TABULEK</b>	.....	<b>86</b>
<b>SEZNAM PŘÍLOH</b>	.....	<b>87</b>

## ÚVOD

Aby si člověk mohl ověřit, že produkty lidské činnosti splňují požadované funkce, vytváří různě důmyslné systémy pro měření, testování a zkoušení těchto produktů.

V České republice byl zřízen Úřad pro technickou normalizaci, metrologii a státní zkušebnictví (ÚNMZ) na základě zákona České národní rady č. 20/1993 Sb. o zabezpečení výkonu státní správy v oblasti technické normalizace, metrologie a státního zkušebnictví. Hlavním posláním ÚNMZ je zabezpečovat úkoly vyplývající ze zákonů ČR upravujících technickou normalizaci, metrologii a státní zkušebnictví a úkoly v oblasti technických předpisů a norem. Od roku 2009 zajišťuje také tvorbu a vydávání českých technických norem, podle kterých se řídí zkušební úřady [7].

V ČR existuje mnoho zkušebních úřadů a společností, které jsou ve většině případů zaměřeny na určitý obor produktů lidské činnosti. V konkrétním případě asfaltových zálivek jsou to společnosti zaměřené na zkoušení a ověřování stavebních materiálů.

U stavebních konstrukcí je důležité přesné vymezení požadavků na výrobek nebo konstrukci v provozních podmínkách a současně určit časové vymezení, po jakou dobu je požadováno tyto podmínky plnit. Spolehlivost u stavebních konstrukcí tvoří hlavně únosnost, použitelnost a trvanlivost.

Zkoušky jsou prováděny ve zkušební laboratoři nebo dokonce i na místě stavby. V případě asfaltové zálivky jsou připraveny vzorky v laboratoři.

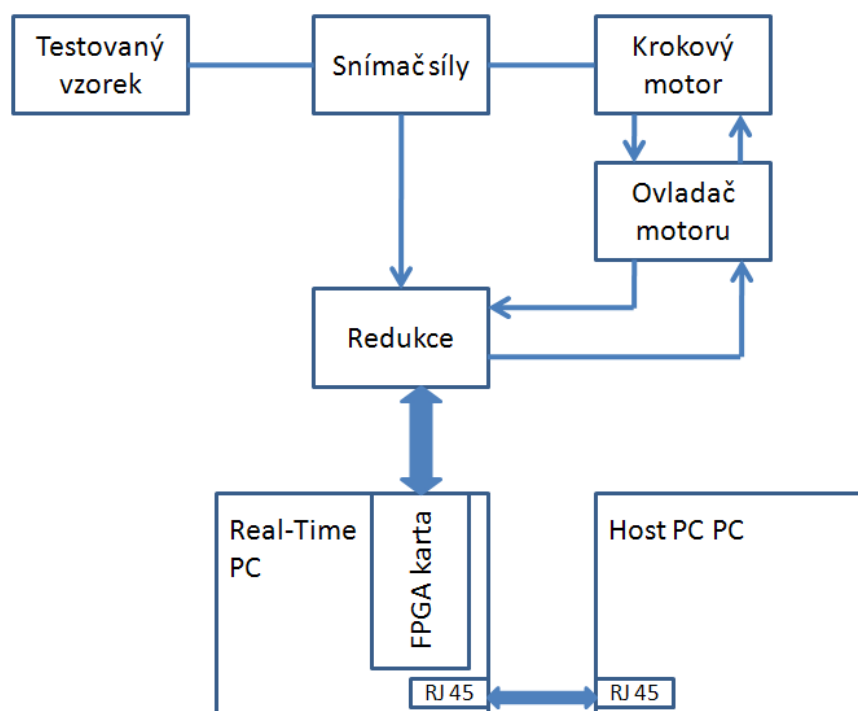
Hlavní náplní práce je vytvoření a popsání aplikace řízení zkoušky podle tlakové a tahové síly a deformace asfaltové zálivky. Popisuje vývojové prostředí LabVIEW, ve kterém je aplikace vytvořena, poté konkrétní aplikaci zkoušky asfaltové zálivky a jednotlivých částí cele zkušební sestavy.

## **I. TEORETICKÁ ČÁST**

## 1 POPIS ZKUŠEBNÍ SESTAVY

Celá zkušební sestava se nachází v laboratoři a sestává se z těchto hlavních částí (Obr. 1):

- Krokový motor ZSE MEZ Z42VV141
- Ovladač krokového motoru, řada NDC
- Redukce NI SCB-68
- Kabel NI SHC68-68 RMIO
- Snímač síly
- Host PC
- Real-Time PC (karta FPGA)
- Vzorek – asfaltová zálivka

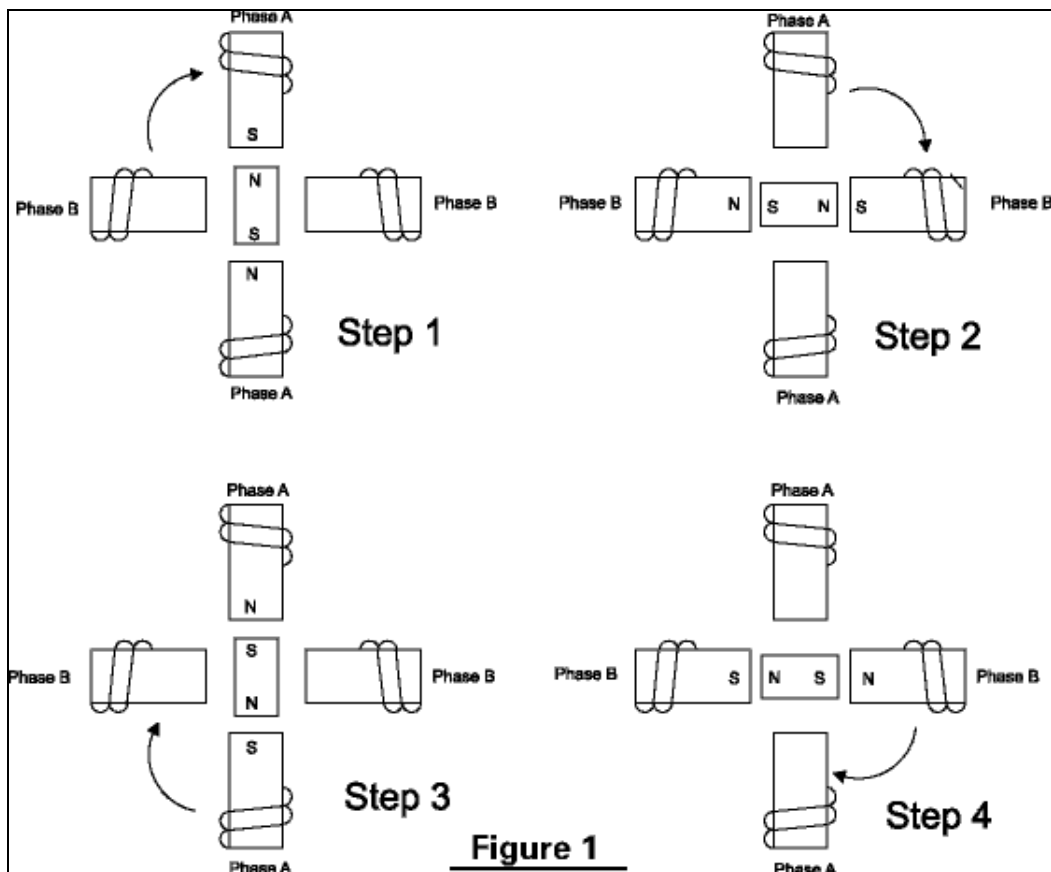


Obr. 1 Zkušební sestava

## 1.1 Krokový motor

### 1.1.1 Obecný popis

Jedná se o krokový motor s aktivním rotorem, kde rotor je zmagnetizován a natáčí se podle magnetické polariry pólů statoru (Obr. 2). Vinutí motoru je dvoufázové, čtyřpólové. V rotoru je uložen trvalý magnet a je uložen v kuličkových ložiscích. Ložiskové štíty jsou hliníkové a kostra ocelová. Provedení motoru je přírubové se dvěma volnými konci hřídele. Motor se připojuje k ovladači šesti volnými vývody. Technický život motoru je 5000 pracovních hodin [8]. Motor vyráběla na začátku 80. let firma ZSE MEZ Náchod, nyní se firma jmenuje ATAS elektromotory Náchod a.s.

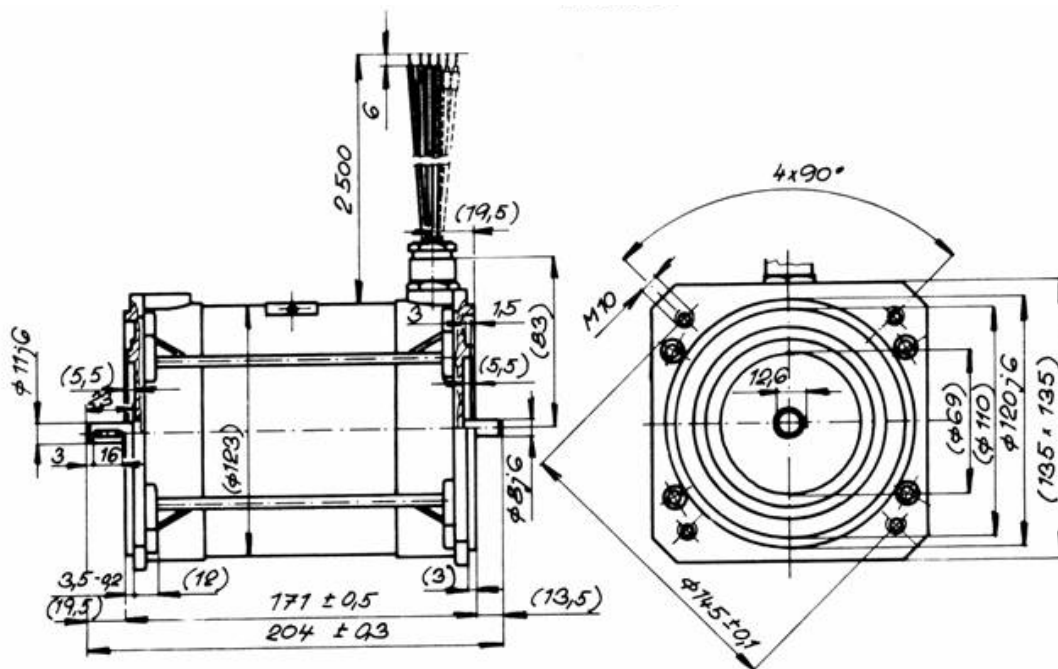


Obr. 2 Obecná funkce krokového motoru s aktivním rotorem

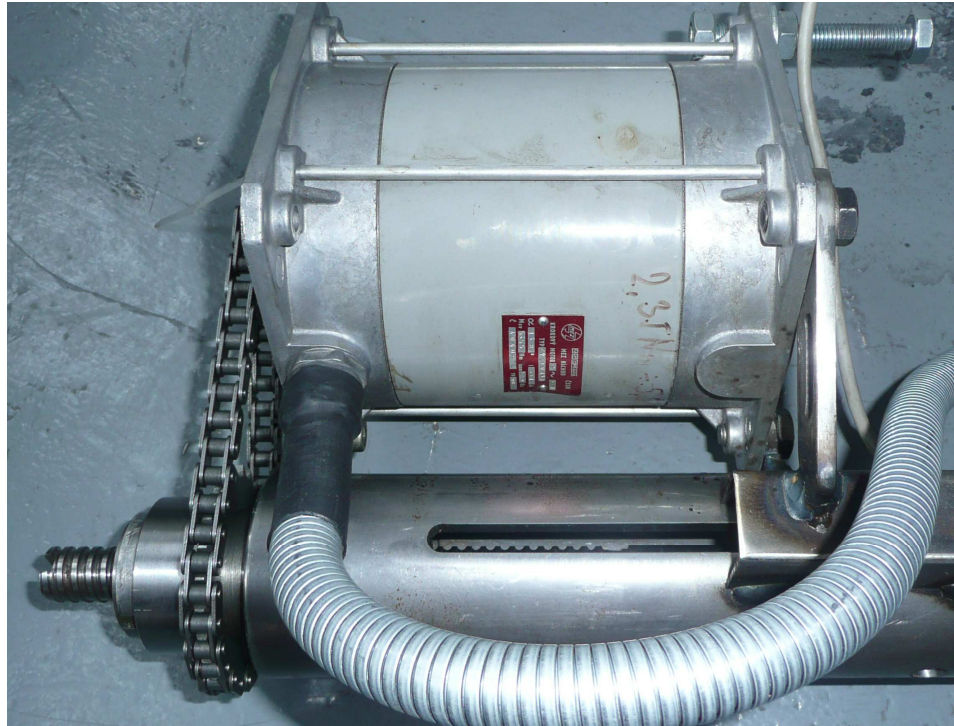
## 1.1.2 Technické údaje

Typ motoru		Z42VV141
Velikost kroku	[ ° ]	1,8
Tolerance kroku	[ ° ]	±0,1
Jmenovitý proud	[A]	7,1
Statický vazební moment	[N·m]	5,55
Nejvyšší provozní moment	[N·m]	4,0
při mezním provozním kmitočtu	[Hz]	150
Mezní provozní moment	[N·m]	0,76
při mezním provozním kmitočtu	[Hz]	5000
Nejvyšší rozběhový kmitočet	[Hz]	550
Činný odpor jedné fáze vinutí	[Ω]	0,36
Moment setrvačnosti rotoru	[kg·m <sup>2</sup> ]	11,3 · 10 <sup>-4</sup>
Hmotnost	[kg]	7,85
Izolační třída		B
Krytí		IP40

Tab. 1 Technické údaje

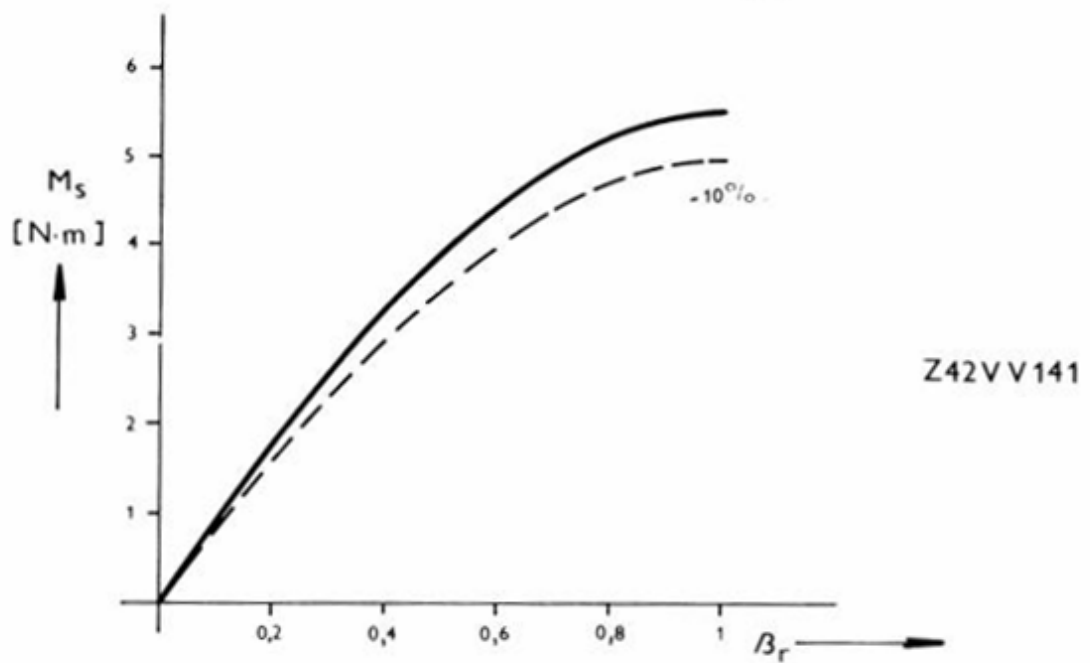


Obr. 3 Technický náčrt motoru Z42VV141

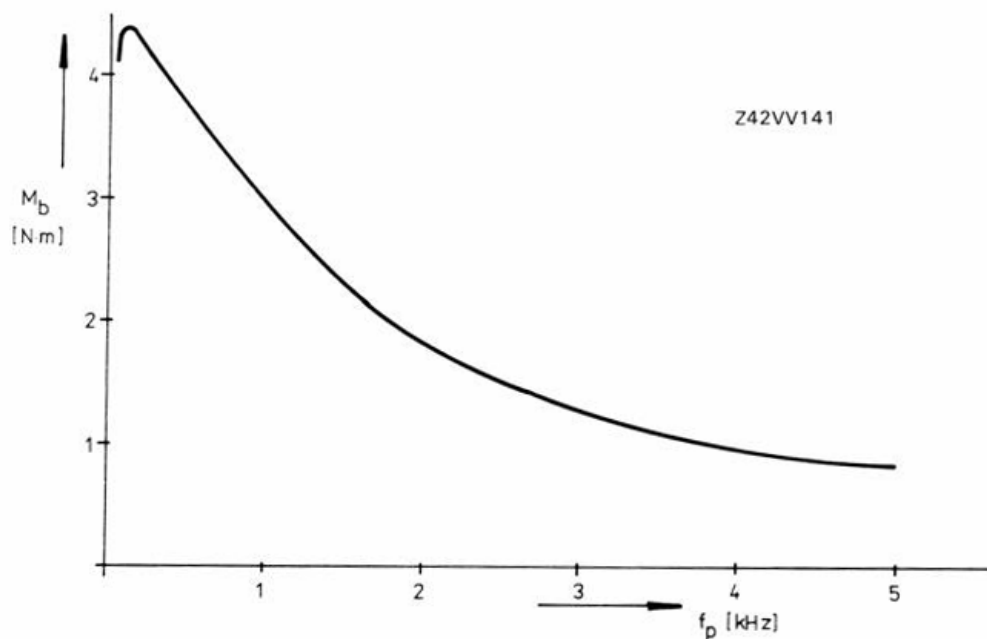


Obr. 4 Zapojení krokového motoru se šroubovicí

### 1.1.3 Charakteristiky krokového motoru



Obr. 5 Charakteristika statického momentu



Obr. 6 Provozní charakteristika

## 1.2 Ovladač krokového motoru

Ovladače řady NDC od firmy ENIKA.CZ spol. s r.o. jsou zkonstruovány pro řízení dvoufázových krokových motorů s úhlem kroku 1,8 stupně.

### 1.2.1 Základní vlastnosti

Typ	NDC 06.V
$V_{DC \text{ nom}}$	24÷75 V
$I_{NP \text{ min}}$	1,9 A
$I_{NP \text{ max}}$	6 A
rozměry	93,5 x 101 x 25 mm
Pracovní teplota	Od +5 do +45 °C

Tab. 2 Základní vlastnosti

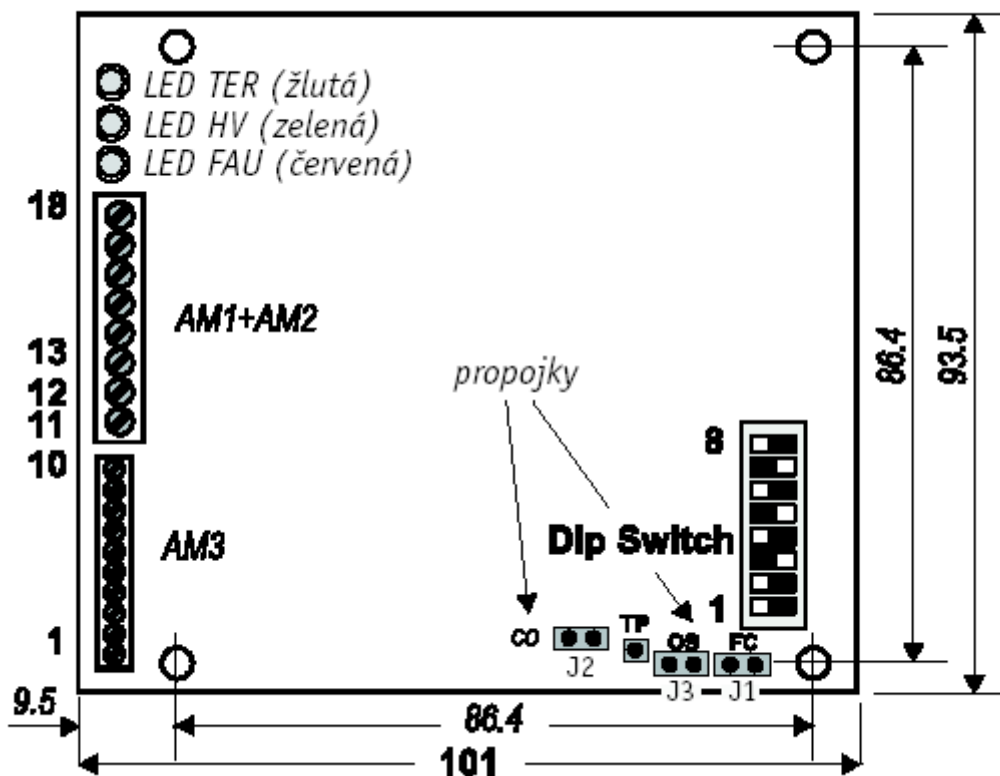
$V_{DC \text{ nom}}$  Nominální hodnota napájecího napětí DC(rozsah) při které může ovladač pracovat bez stabilizovaného napájení

$I_{NP}$  Nominální proud fáze (špičková hodnota), který teče v každém vinutí motoru, měřená při otáčení motoru malou rychlostí. Tento proud se vybírá ze čtyř hodnot pomocí přepínače DIP SWITCH.

$I_{NP \text{ min, max}}$  Minimální a maximální hodnota nominálního proudu fáze.

### 1.2.2 Vstupní a výstupní logické signály

Vstupní a výstupní logické signály jsou opticky izolované vzhledem k vnitřnímu napájení. Pro všechny vstupy a výstupy se dodržuje plusová a minusová svorka. Rozmístění všech vstupů a výstupů jsou zobrazeny na obrázku (Obr. 7). Pro logické vstupy a výstupy je určena 10 pólová šroubová svorkovnice s roztečí 2,54 mm s označením AM3.



Obr. 7 Rozměry a rozmístění součástek (svorkovnice, LED diody, přepínače a propojky) na desce plošných spojů

Číslo svorky (polarita)	Význam signálu
2(-); 3(+)	<p><b>Vypnutí proudu</b></p> <p>Když je signál ON, je ovladač aktivní. Pokud je signál OFF, je ovladač neaktivní a točivý moment krokového motoru je nulový. Ovšem s propojkou CO (Obr. 5) je ovladač vždy aktivní.</p>
4(-); 5(+)	<p><b>Směr</b></p> <p>Když je signál ON, bude otáčení motoru opačné, než bylo se vstupem OFF. Tento signál bude platný nastane-li 100 <math>\mu</math>s před signálem Krok a zůstane v tomto stavu ještě min. 100 <math>\mu</math>s po posledním signálu Krok.</p>
6(-); 7(+)	<p><b>Krok</b></p> <p>Krok nastane při přechodu signálu ON-OFF. Pracovní cyklus 50%. Max frekvence 60 kHz s obdelníkovým průběhem ze zdroje o napětí od 3,5 V do 13 V. Perioda musí být delší, než 8 <math>\mu</math>s.</p>
8(-); 9(+)	<p><b>Chyba</b></p> <p>Při sepnutém výstupu pracuje ovladač normálně, při rozepnutém je nefunkční.</p>
1; 10	<p><b>Vnitřní zem</b></p> <p>Použití pro stínění kabelů logických signálů. Jsou spojeny s jedním pólem napájecího napětí a jsou galvanicky izolované od log. vstupů a výstupů</p>

Tab. 3 Zapojení vstupních a výstupních logických signálů, svorkovnice AM3

### 1.2.3 Výkonné vstupy a výstupy

Výkonné vstupy a výstupy jsou na ovladači krokového motoru uvedeny jako svorkovnice AM1 a AM2. Jedná se o šroubové svorkovnice s roztečí 3,8 mm s 8 póly (Obr. 7). V tabulce (Tab. 4) je uvedeno zapojení výkonných vstupů a výstupů [9].

Číslo svorky	Svorkovnice	Popis vstupů a výstupů
11	AM2	Kladný pól napájecího zdroje (+VDC nom)
12	AM2	Záporný pól napájecího zdroje (-VDC nom)
13	AM1	Ground; spojit s ochranným vodičem (PE)
14	AM1	Vinutí motoru B-
15	AM1	Vinutí motoru B
16	AM1	Vinutí motoru A-
17	AM1	Vinutí motoru A
18	AM1	Stínění kabelu k motoru

Tab. 4 Zapojení výkonných vstupů a výstupů

### 1.2.4 Hlavní nastavení podle návodu výrobce

Nastavení se provádí podle návodu (Tab. 5) přepínači a propojkami (Obr. 7). Hlavním důvodem nastavení je správný chod krokového motoru podle jeho typu a pracovních podmínek. V tabulce je nastavení pro typ krokového motoru ZSE MEZ Z42VV141 podbarveno žlutou barvou.

Jmenovitý proud INP (dip 1, 2, 3)			
dip 1	dip 2	dip3	NDC 06.A
ON	ON	ON	1,9 A
ON	ON	OFF	2,3 A
ON	OFF	ON	2,7 A
ON	OFF	OFF	3,1 A
OFF	ON	ON	3,6 A
OFF	ON	OFF	4,3 A
OFF	OFF	ON	5,1 A
OFF	OFF	OFF	6,0 A
Propojka		Význam propojení	
FC (J1)	ON	automatická redukce proudu vyřazena	
FC (J1)	OFF	automatická redukce proudu aktivní	
CO (J2)	ON	ovladač aktivní	
CO (J2)	OFF	ovladač blokován	
OS (J3)	ON	funkce RUN, používá se vnitřní oscilátor	
OS (J3)	OFF	vnější řízení funkce Krok	
Tlumení (dip 4)			
ON	aktivní		
OFF	vyřazeno		
Druh činnosti - kroků na otáčku (dip 6, 7, 8)			
dip7	dip 8	dip 6 ON	dip 6 OFF
ON	ON	4000	3200
ON	OFF	2000	1600
OFF	ON	1000	800
OFF	OFF	500	400

Tab. 5 Možnosti nastavení přepínačů a propojek

Funkce tlumení zajišťuje snížení hluku při otáčení rotoru krokového motoru. Tento hluk způsobují vibrace typické pro krokové motory. Jak je vidět z tabulky, vibrace se snižují zvýšením počtu kroků na otáčku. Samozřejmě se musí brát v úvahu tzv. max. rozlišení motoru, tedy počet kroků/otáčku.

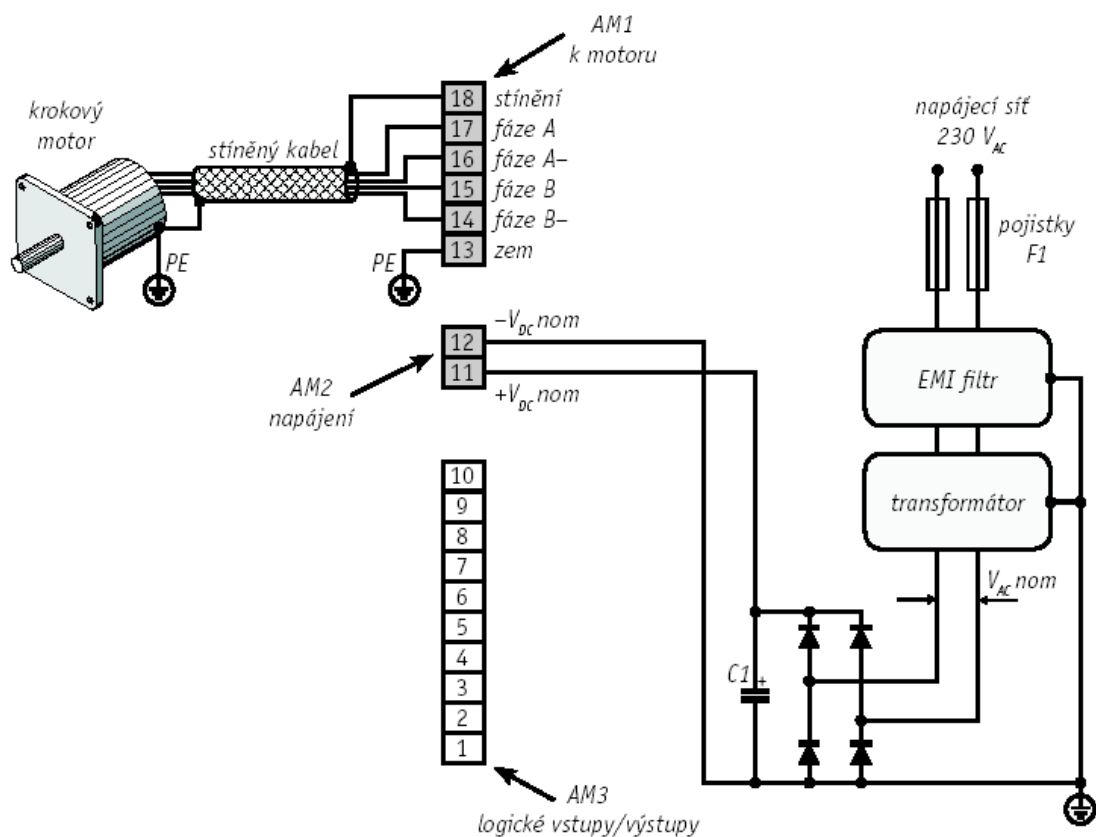
### 1.2.5 LED indikace stavu

Na ovladači se nachází mimo jiné LED diody ve třech barvách. Tyto LED indikují stav ovladače (Tab. 6).

Barva LED	Stav	Význam
zelená	ON	Hodnota napájení je v pracovním rozsahu
	OFF	Napájení je mimo pracovní rozsah nebo chybí
červená	ON	Ovladač není funkční kvůli některé z chyb:
		a) Napětí max. nebo min.
		b) Zkrat na výstupu ovladače
OFF	Ovladač je aktivní	
žlutá	ON	Ovladač není funkční kvůli teplotní ochraně
	OFF	Ovladač je aktivní

Tab. 6 LED indikace stavu

### 1.2.6 Schéma připojení ovladače



Obr. 8 Schéma připojení ovladače

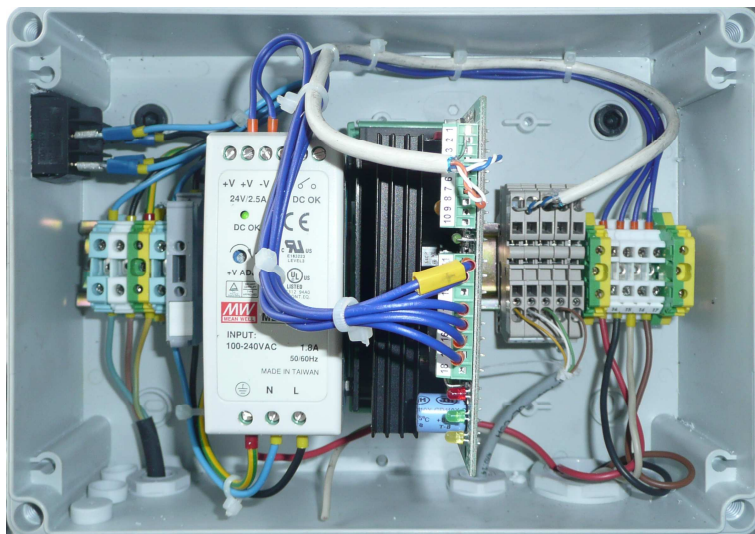
### 1.2.7 Regulovatelný zdroj MDR-60

Podle Obr. 8 je zapojen do svorkovnice AM2 regulovatelný zdroj pro průmyslovou automatiku MDR-60 od firmy Mean Well enterprise co. ltd s těmito parametry [10]:

- uzavřená forma
- montáž na DIN lištu
- šroubové přípoje s ochranou před dotykem
- rozsah vstupního napětí od 85...264 V AC, 47...63 Hz a 120...370 V DC
- ochrana proti přetížení ohraničením proudu, auto recovery
- ochrana před zkratem
- 100% Burn-In-Test při plném zatížení
- zabudovaný odrušovací filtr, nízké zbytkové vlnění
- single output
- 1A vstupní jmenovitý proud při 230V
- 60A náběhový proud pro studený start s 230V
- rozsah provozní teploty: -20...+70°C
- ochrana proti přetížení: 105...160%
- ochrana proti přepjetí: 115...135%
- izolační napětí od vstupu do výstupu 3000 V AC
- izolační odpor od vstupu do výstupu 100 MΩ
- 50ms překlenutí výpadku sítě při 230V
- izolační třída II
- LED označení pro Power On



Obr. 9 Regulovatelný zdroj MDR-60

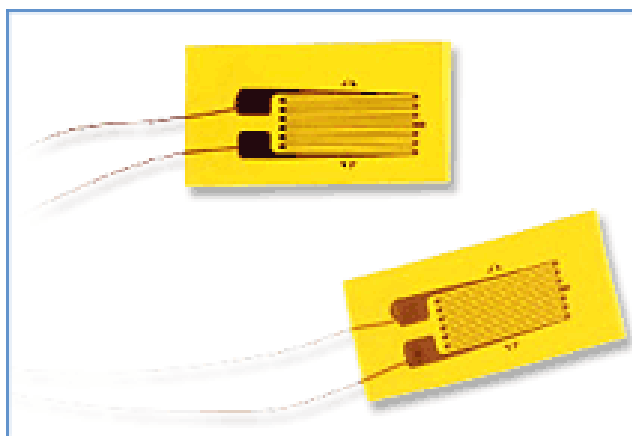


Obr. 10 Ovladač krokového motoru v zapojení s regulovatelným zdrojem

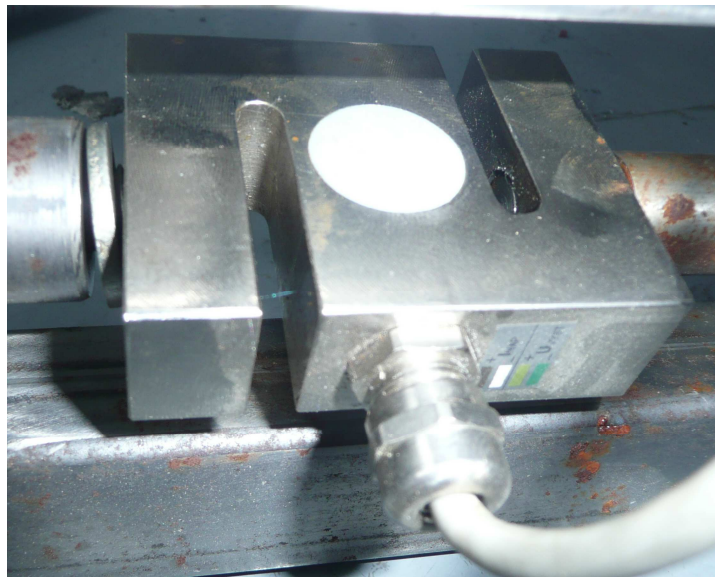
### 1.3 Snímač síly

Ve zkušební sestavě byl použit snímač síly v tahu i tlaku S-článek se čtyřmi tenzometry zapojenými do Wheatstoneova můstku a s otvory se závitem na obou stranách. Přesnost snímače je 0,03% z rozsahu měření.

Rozsah měření je  $\pm 5000$  N. Výstup:  $3 \text{ mV/V} \pm 0,0075 \text{ mV/V}$ . Výstupní odpor:  $350 \Omega \pm 5 \Omega$  [11].



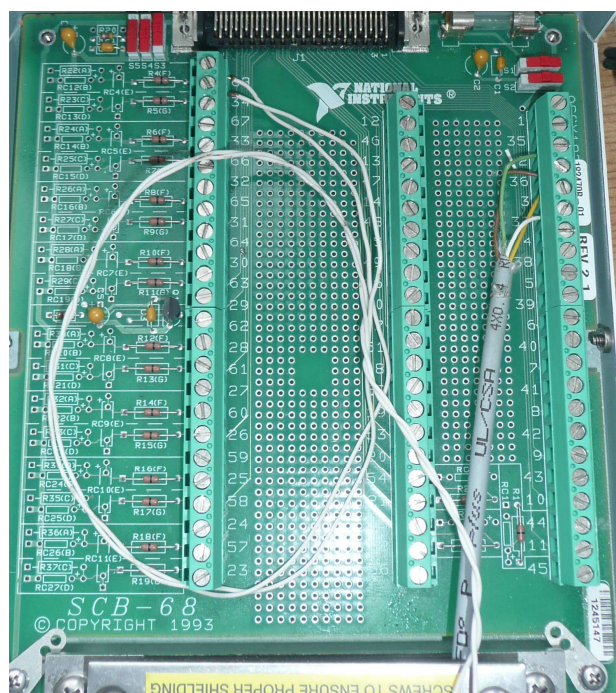
Obr. 11 Typické fóliové nalepovací tenzometry



Obr. 12 Snímač síly připevněný mezi krokovým motorem a měřeným vzorkem

#### 1.4 Redukce NI SCB-68

Redukce SCB-68 je vlastně propojovací svorkovnice se 68 šroubovými svorkami pro snadné propojení mezi 68-PIN DAQ zařízením sběru dat (FPGA karta PCI-7831R) a samotnými snímači s řízeným zařízením, v našem případě tenzometrem a krokovým motorem. Zapojení do redukce SCB-68 je vidět na Obr. 13 a bylo provedeno podle uživatelského manuálu [6]. Tištěný spoj se šroubovými svorkami je usazen v masivním plechovém boxu s možností uzavření víka (Obr. 14).

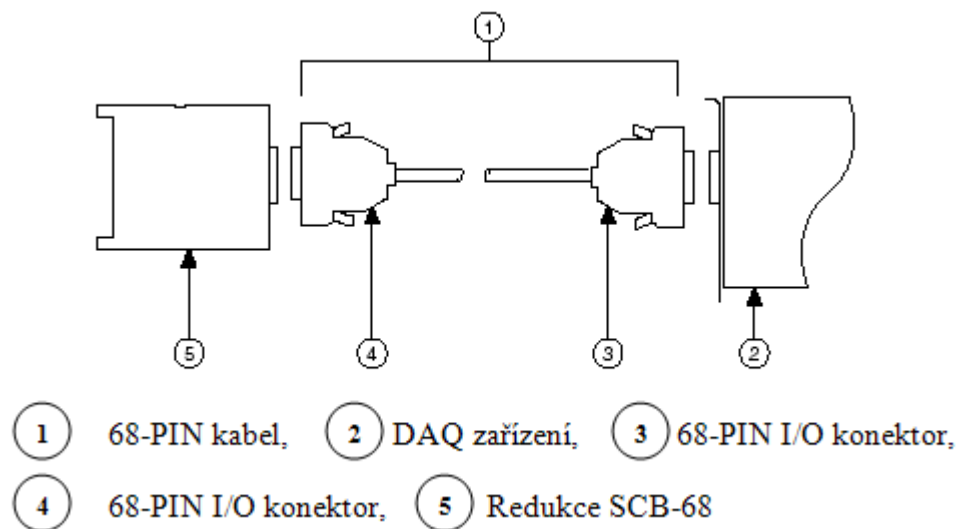


Obr. 13 Zapojení tenzometru a krokového motoru do redukce



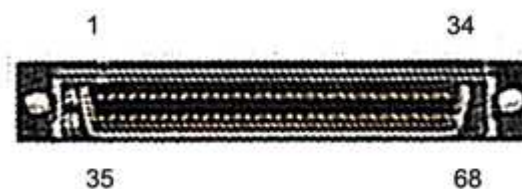
Obr. 14 Propojovací redukce SCB-68

## 1.5 Kabel SHC68-68 RMIO



Obr. 15 Schéma použití konektoru 68-PIN I/O

Obrázek (Obr. 15) ukazuje použití 68-PIN kabelu pro propojení mezi Redukcí SCB-68 a DAQ 68-PIN zařízením sběru dat (NI PCI-7831R).

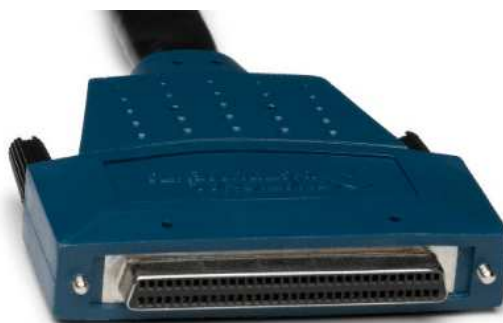


Obr. 16 Číslování pinů konektoru SCB-68

Výkonný stíněný kabel je určený ke konkrétním kartám sběru dat ze série R. Uvnitř vedení jsou jednotlivě stíněny analogové výstupy, analogové vstupy a digitální I/O.



Obr. 17 Kabel SHC68-68 RMIO



Obr. 18 Konektor pro zapojení do propojovací  
redukce SCB-68



Obr. 19 Konektor pro zapojení do FPGA karty  
PCI-7831R

## 1.6 NI PCI-7831R

### 1.6.1 Obecný popis

NI PCI-7831R (Obr. 20) je multifunkční karta určená k získávání (DAQ – Data Acquisition) a odesílání dat s programovatelným FPGA (Field Programmable Gate Array) polem [6]. Umisťuje se do slotu PCI v Real-Time PC.

FPGA pole pracuje na základě naprogramovaného pole logických bloků (Obr. 21). K naprogramování se nejvíce používá programovací jazyk C, ale také určité modifikace, jako je grafické vývojové prostředí LabVIEW, kde se programuje v grafickém rozhraní pomocí umisťování ovládacích prvků, indikátorů, logických funkcí a dalších bloků do různých podmínkových smyček.



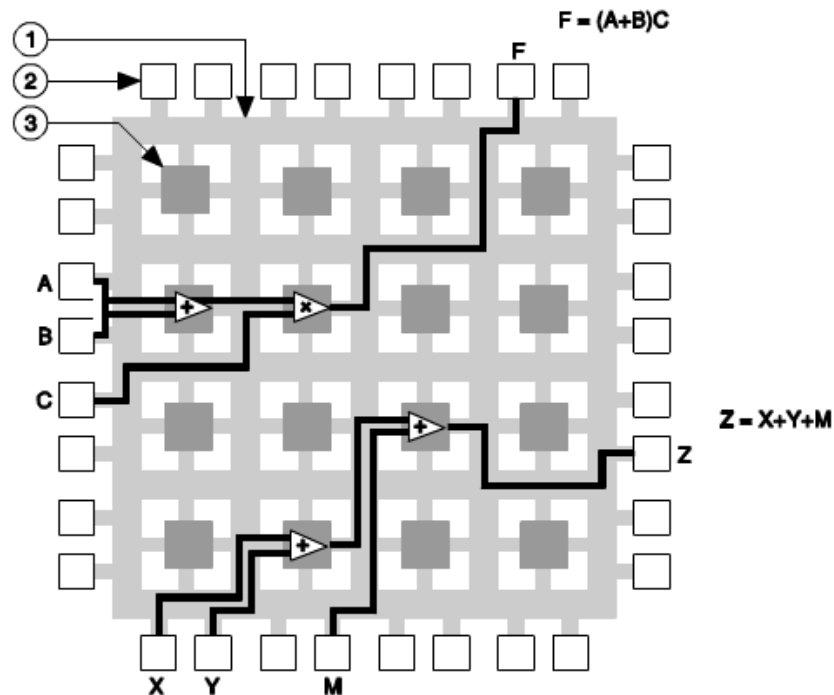
Obr. 20 FPGA karta NI PCI-7831R

Základní struktura FPGA je tvořena polem programovatelných logických bloků (na Obr. 21 č. 3). Ty se většinou dělí ještě na menší logické buňky. Logické buňky obsahují strukturu pro tvoření kombinačních funkcí a klopné obvody. Logický blok dovoluje vytvořit jen poměrně jednoduché funkce a k vytvoření složitějších je třeba tyto bloky propojit. K tomu je určena propojovací struktura (na Obr. 21 č. 1). Celé pole logických bloků s propojovací strukturou je obklopeno vstupně-výstupními bloky (na Obr. 21 č. 2), které jsou připojeny

k vývodům FPGA a jejich úkolem je propojení vnějších signálů se signály v poli logických bloků [12].

Najít optimální propojení logických bloků a maximální využití minimálního počtu logických buněk v FPGA je natolik složité, že se tomu používají různé vývojové programy, aplikace a hlavně prostředí jako je např. LabVIEW.

Kromě logických buněk, které jsou základními prvky, jsou v FPGA vytvořeny další struktury pro použití např. většího počtu hodinových signálů a další přídavné prvky. Aby tyto a mnohem větší množství nových vlastností obvodů FPGA mohl člověk využít, musí mít dostatek znalostí o nich. Je však jen těžko představitelné, že by někdo tyto znalosti obsáhl. K tomu je určeno mnoho návrhových systémů.



Obr. 21 Struktura FPGA

### 1.6.2 Vlastnosti karty

Karta PCI-7831R má tyto vlastnosti a obsahuje [6]:

- 8 analogových vstupů vzorkovacích frekvencí až 200 kHz, 16 bitové rozlišení  $\pm 10$  V
- 8 analogových výstupů vzorkovacích frekvencí až 1 MHz, 16 bitové rozlišení  $\pm 10$  V
- 96 možných digitálních vývodů nastavených jako vstupy, výstupy, čítače nebo jiné logiky
- Čip Virtex II 1M – FPGA (Obr. 22) programovatelný v prostředí LabVIEW



Obr. 22 Čip FPGA

- 3 vysokorychlostní komunikační DMA kanály
- Určená pro PCI sběrnici
- Operační systém FPGA, Real-Time, Windows
- Produktová řada R Serie
- 80 kB paměti

- Watchdog Timer
- Hodnota max. vstupu je 0 V, 5 V
- Hodnota max. výstupu je 0 V, 3,3 V
- Počet čítačů: 96
- Minimální šířka pulzu: 12,5 ns
- Logické úrovně: TTL, 2,5 V, 3,3 V, LVTTTL

Karta je osazena v tzv. Real-Time PC a komunikuje přes výše uvedený kabel SHC68-68 RMIO a propojovací redukci (svorkovnicí) SCB-68 s vnějšími signály od tenzometru a krokového motoru.

## 1.7 Real-Time PC

Požadavky LabVIEW na PC, co by cíl real-time aplikace a umístění FPGA karty jsou následující, avšak National Instruments jako výrobce systému LabVIEW nedoporučuje ze dvou důvodů používat PC jako real-time PC. První je ten, že jeden výrobce např. procesoru (CPU) nezaručuje stejné vlastnosti všech vyrobených CPU z jedné série a série další. Ve druhém případě se dlouho hledá nejlépe vyhovující HW vybavení.

Jedním z nejdůležitějších požadavků je, aby šlo vývojové prostředí LabVIEW na PC nainstalovat a aby bezvadně fungovalo. Dál pak musí na PC pracovat samozřejmě i LabVIEW Real-Time MODULE a Real-Time operační systém.

Minimální požadavky na HW jsou pro použití verze LabVIEW 8.2 tyto [6]:

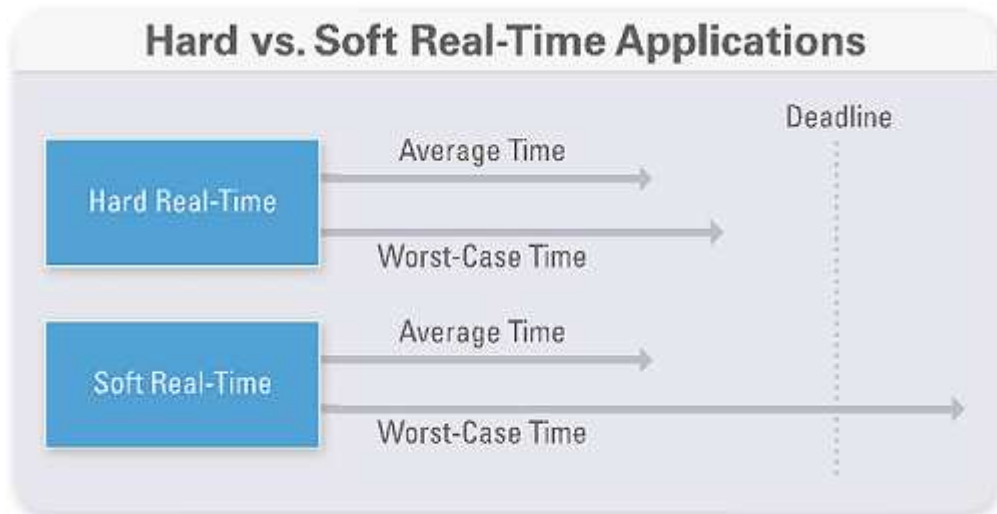
- Procesor
  - procesory Intel® Celeron®
  - Intel® Pentium® M
  - AMD Athlon™ 64
  - AMD Athlon™ XP
- Síťový chipset
  - Intel® 82550

- Intel® 82540
- Pevný disk
  - Sběrnice Parallel ATA (IDE)
- Paměť RAM
  - Minimální velikost 2 GB
- Základní deska
  - Napětí 5 V a současně 3,3 V
- Formát souborového systému
  - FAT32

Pak je také možné si z internetových stránek společnosti National Instruments [6] uložit aplikaci k rozpoznání vhodnosti HW vybavení PC pro Real-Time PC. Počítač v laboratoři všechny tyto požadavky splňuje a je na něm nainstalováno potřebné programové vybavení.

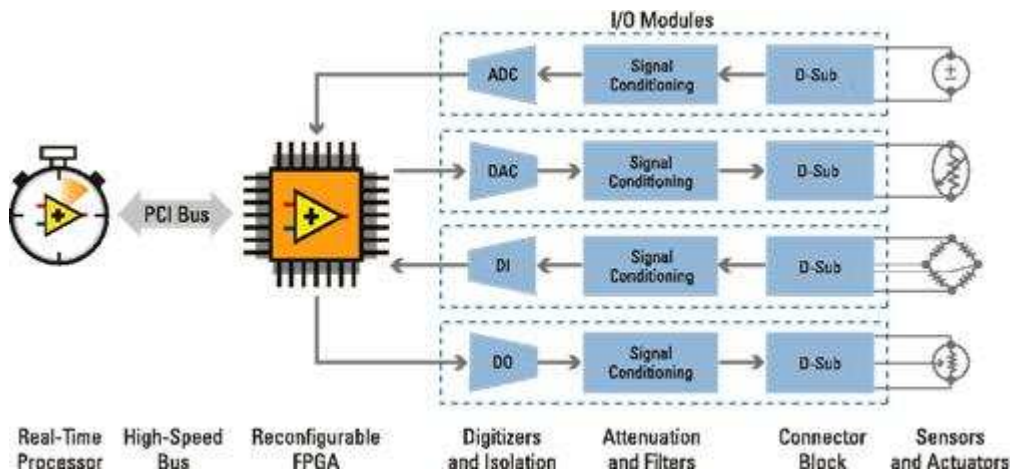
Real-Time systém zpracování dat vykonává programy se specifickými požadavky na časování, které jsou v určitých průmyslových a vědeckých aplikacích potřebné. Jediná nezbytná komponenta pro práci v systému v reálném čase je výše zmíněný Real-Time operační systém.

Pro mnoho vývojářů je neakceptovatelné pracovat ve standardním operačním systému, protože se zde nachází hodně rušivých vlivů, které by mohly brzdit výkon a neumožňovaly by chod programu bez přerušení. Jsou to např. různé aktualizace, spuštěný antivirový program a systémové procesy spuštěné na pozadí. Všechny tyto a více různých spouštějících se procesů a aplikací běží ve standardním operačním systému jako by najednou (multitasking). Ovšem v Real-Time operačních systémech běží většinou jednoduchý program s velmi přesným časováním. Real-Time aplikace se dělí na tzv. Hard Real-Time a Soft Real-time. Hard Real-Time aplikace je taková, ve které je zapotřebí absolutně zamezit selhání řízeného systému, to aby nedošlo k vážným a často drahým poškozením (Obr. 23). Hard Real-Time aplikace se používají v průmyslu velmi často. Naopak Soft Real-Time aplikace se používají např. v optických a video systémech, kde žádné poškození nehrozí [6].



Obr. 23 Rozdíl mezi Hard a Soft Real-Time aplikacemi

Ne všechny systémy jsou předurčené pracovat v reálném čase, ovšem pokud bychom potřebovali zamezit velkému nebezpečí poškození zařízení nebo dokonce ohrožení lidského zdraví, pak potřebujeme Real-Time systémy s vysokou spolehlivostí a precizním časováním. Nesmíme zapomenout na Watchdog funkci hlídání zastavení běhu programu.



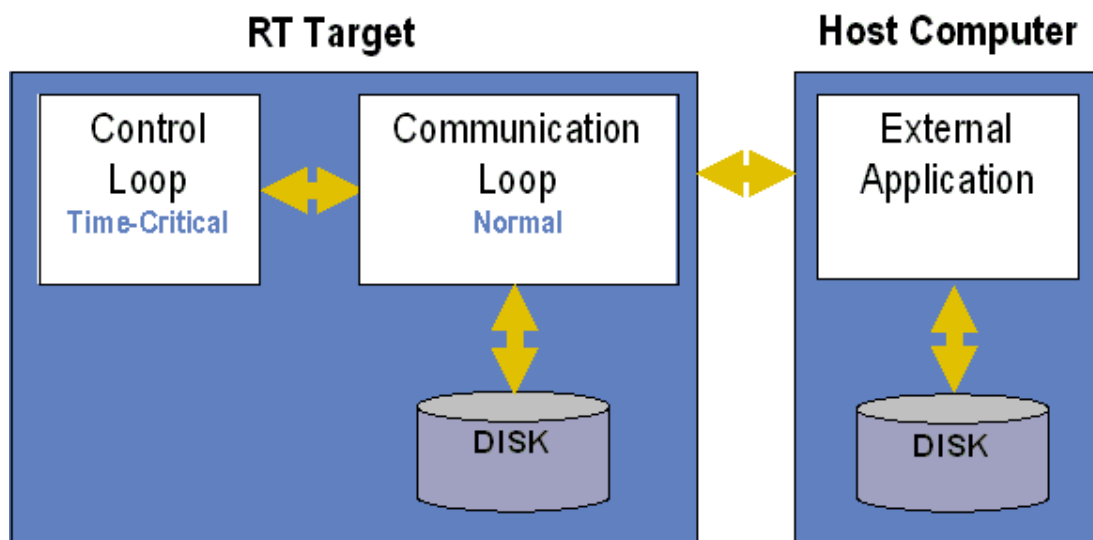
Obr. 24 Real-Time PC – FPGA – senzory

Na Obr. 24 [6] je zřejmé schéma toku signálů a informací v Real-Time PC z procesoru přes sběrnici do karty s FPGA ven k senzorům a akčním členům.

## 1.8 Host PC

Použití Real-Time PC je z důvodu jednak operačního systému reálného času a Real-Time časové smyčky a výše popsaných výhod použití. Kdežto na druhém počítači s názvem Host PC běží uživatelské rozhraní pro konfiguraci parametrů měření a řízení, běží na něm průběžné ukládání měřených dat do paměti a jejich průběžné zobrazování (úspora paměti Real-Time PC). A jednorázové uložení naměřených dat do souboru.

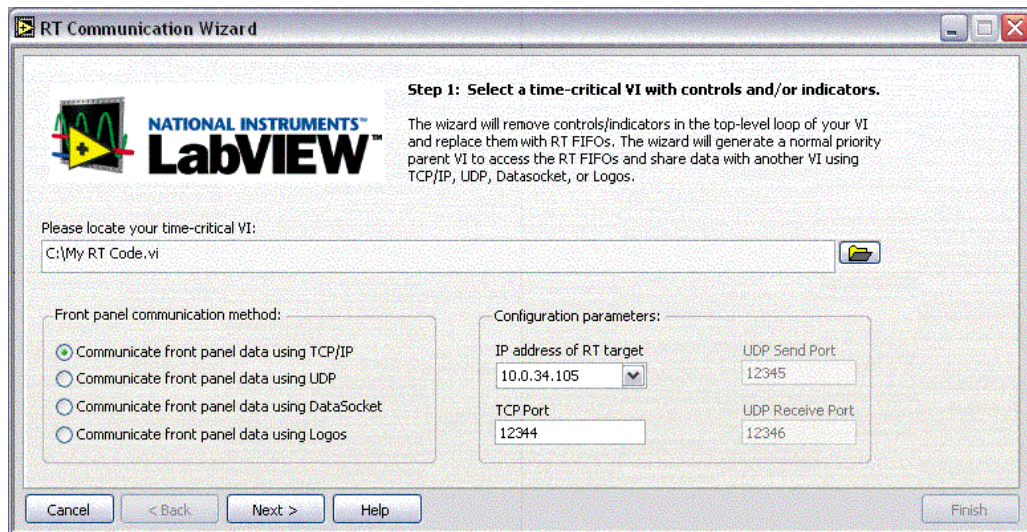
Toto rozdělení na dva PC propojené Ethernet kabelem nepřetěžuje Real-Time PC a umožňuje tak zaznamenávat velmi dlouhá, někdy i nepřetržitá měření.



Obr. 25 Komunikace mezi Host PC a Real-Time PC

Na Real-Time PC běží paralelně dvě smyčky, Control Loop a Communication Loop. Control Loop je nejdůležitější smyčka, proto také Time-Critical. Tato smyčka obsahuje řídicí kód a je nastavena na nejvyšší prioritu. Communication Loop obsahuje kód prostřednictvím kterého dochází k přenosu informací mezi Real-Time PC a Host PC [6]. Priorita je nastavena tak aby přenos časově neomezil běžící řídicí smyčku.

Host PC a Real-time PC spolu komunikují pomocí různých komunikačních protokolů, které se dají také nastavit pomocí průvodce nastavení komunikace RT Communication Wizard (Obr. 26).



Obr. 26 RT Communication Wizard

## 1.9 Vzorek – asfaltová zálivka

Asfaltové zálivky se používají v dopravních stavbách a aplikují se do dilatačních spár silničních a mostních konstrukcí. Jejich vlastnosti by měly být takové, aby se při rozšiřování a zmenšování spáry nenarušily a nezačaly se v trhlinách usazovat jiné látky a nečistoty, hlavně voda. Testovaný vzorek je aplikován mezi dva betonové kvádry (Obr. 27) a podle ČSN musí mít danou délku, šířku a výšku. Vše popsáno v následující kapitole.



Obr. 27 Vzorek asfaltové zálivky

## 2 ČESKÁ TECHNICKÁ NORMA PRO ASFALTOVÉ ZÁLIVKY

Všeobecně pro zálivky používané za horka existují české technické normy ČSN EN 13880-1 až ČSN EN 13880-13, dělí se tedy na 13 částí [7]. V této práci je vyvinuta aplikace pro zkoušení asfaltových zálivek podle ČSN EN 13880-10 (Zálivky za horka - Část 10: Zkušební metoda pro stanovení adheze a koheze po opakovaném kontinuálním protahování a stlačování) vydanou 1.11.2004 a podle ČSN EN 13880-13 (Zálivky za horka - Část 13: Zkušební metoda pro stanovení koheze a adheze přerušovaným protažením).

Jedná se o dvě zkoušky adhezních a kohezních vlastností v mém případě zkušebních vzorků (Obr. 27), asfaltových zálivek. Podstatou zkoušek je zjistit zda asfaltová zálivka dostatečně přilne k betonu. Pro obě zkoušky musí mít zkušební vzorek délku ( $50\pm 0,5$ )mm a šířku ( $24\pm 0,5$ )mm vzniklou ve spáře mezi dvěma betonovými bloky o délce ( $125\pm 1$ )mm, šířce ( $50\pm 0,5$ )mm, výšce ( $50\pm 0,5$ )mm a vlhkosti ( $5\pm 0,5$ )%.

### 2.1 Opakované protahování a stlačování

První zkouška odpovídá postupu popsaném v ČSN EN 13880-10. Se zkušebními vzorky proběhnou tři cykly. Každý cyklus probíhá tak, že se spára roztáhne o 18mm rychlostí ( $6\pm 0,25$ )mm/h a okamžitě se stlačí stejnou rychlostí na původní rozměry. Zaznamenává se síla. Zjištění a výpočty napětí v tahu a kontrola vzorků po skončení zkoušky není předmětem této práce.

### 2.2 Přerušené protažení

ČSN EN 13880-13 popisuje postup druhé zkoušky. Po uchycení vzorku se spára roztahuje po krocích po ( $0,1\pm 0,01$ )mm každých ( $360\pm 20$ )s až do roztažení o 5mm, kdy se roztažení zastaví a po dobu, kdy pokles síly vždy po 30 minutách bude menší než 5% se zaznamenává síla.

### 3 PROSTŘEDÍ LABVIEW

Tato kapitola je o stručném popisu a seznámení se s grafickým programovacím prostředím LabVIEW, které bylo použito pro sběr dat a řízení krokového motoru v této práci.

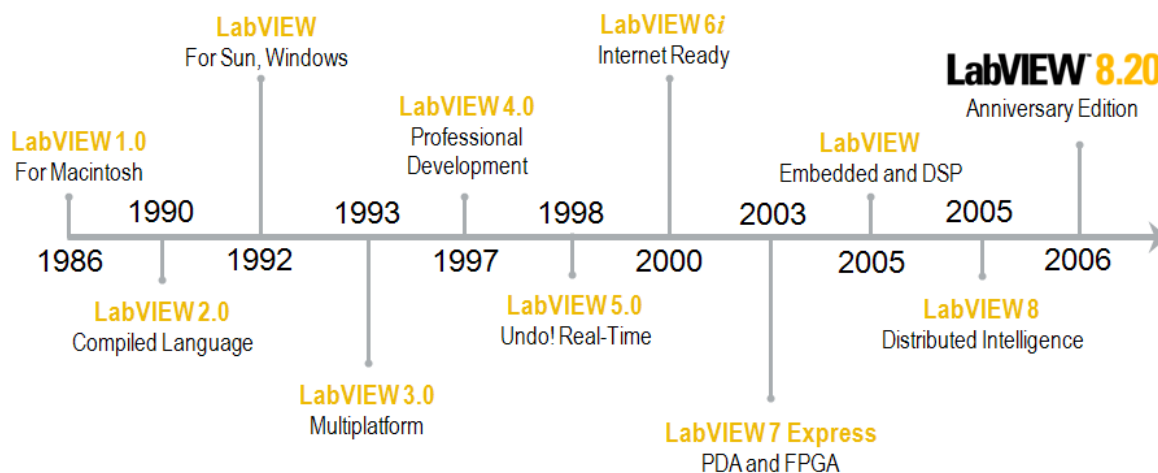
#### 3.1 Pojem Labview

LabVIEW (z angl. Laboratory Virtual Instruments Engineering Workbench) je grafický programovací jazyk od společnosti National Instruments a je na světě od roku 1986, kdy byla uvedena první verze LabVIEW 1.0 pro počítače Macintosh (Obr. 28) na kterém bylo uvedeno první uživatelské grafické rozhraní (GUI) [6].



Obr. 28 LabVIEW 1.0 a počítač  
Macintosh

Po dvaceti letech se vývoj dostal k verzi LabVIEW 8.2 (Obr. 29), ve které je aplikace naprogramována. V současné době se vývojáři firmy NI zaměřují především na vývoj LabVIEW při využití vícejádrových procesorů (LabVIEW 2010).



Obr. 29 Časová osa vývoje LabVIEW

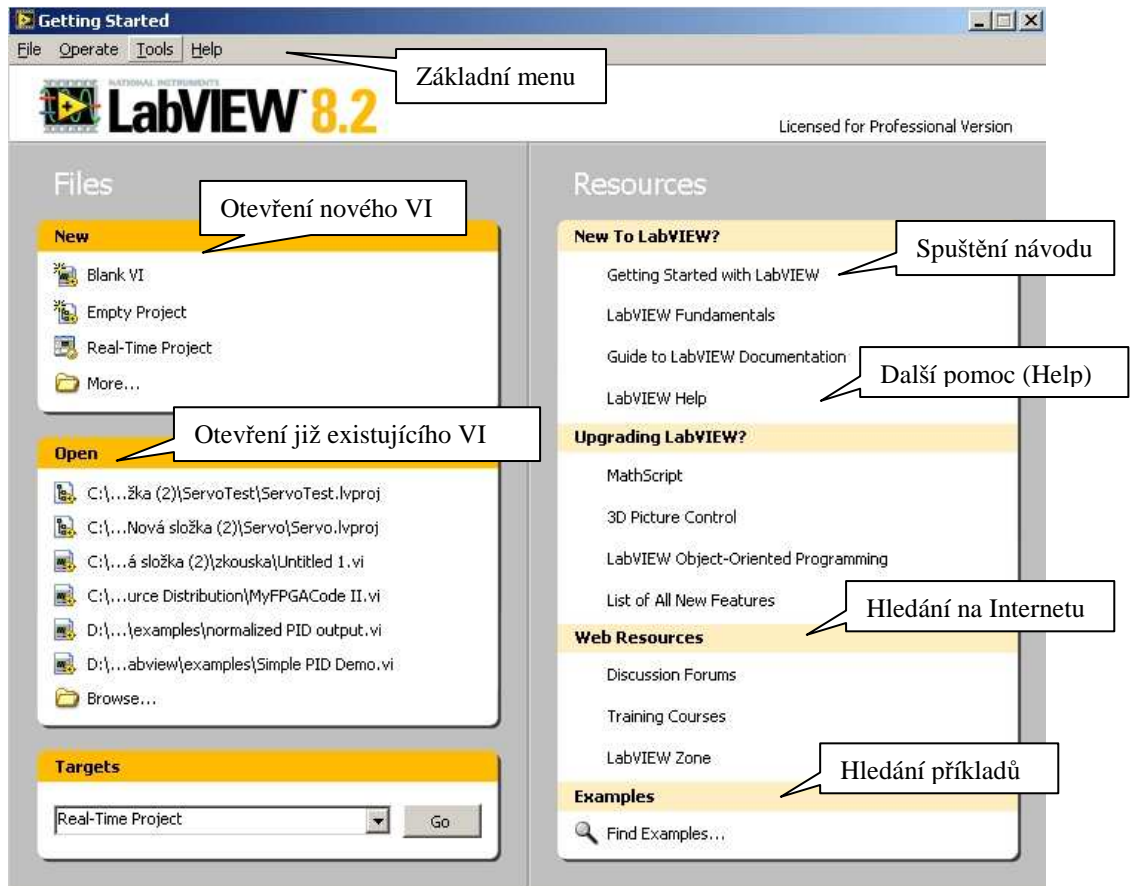
LabVIEW je vlastně v překladu laboratorní pracoviště virtuálních přístrojů, je vhodné pro měření a analýzu signálů, řízení a vizualizaci technologických procesů. Dalo by se říct, že prostředí LabVIEW nemá omezení své použitelnosti [1].

Hlavním důvodem vytvoření virtuální instrumentace bylo a je nahradit, mnohdy rozměrově velké a finančně nákladné technické prostředky pro měření a řízení virtuálními zařízeními. Toto řešení umožňuje rychlé navrhování nových aplikací a hlavně možnosti provádění změn v nastavení a vizualizaci. Tyto možnosti rychlé změny jsou u technických prostředků za pomoci reálných součástí často z finančního hlediska nemožné.

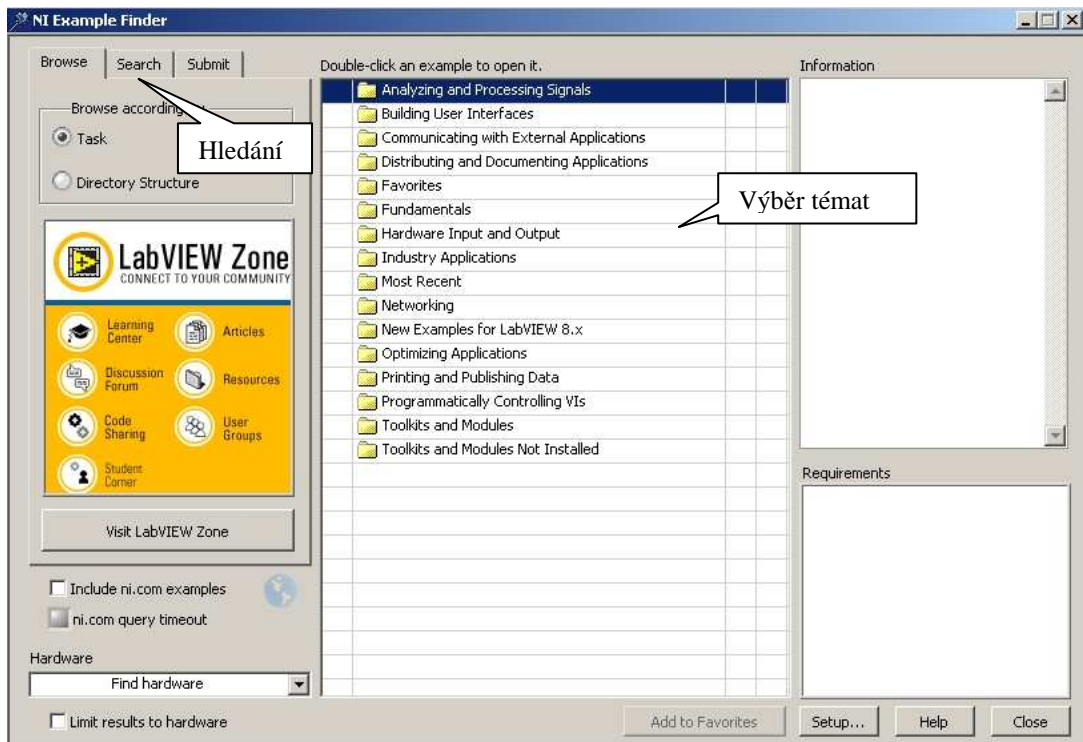
V LabVIEW se pojem virtuální instrumentace objevuje také jako koncovka souborů (programů) \*.vi (VI aplikace).

### 3.2 Po spuštění LabVIEW

Vždy po spuštění LabVIEW se zobrazí úvodní obrazovka Getting Started (Obr. 30) s ikonami pro práci se soubory a pro hledání dalších informací pro vytváření VI. Po kliknutí na Blank VI se objeví už tzv. čelní panel a blokový diagram. Po kliknutí na hledání příkladů se otevře okno NI Example finder, kde může programátor nalézt mnoho potřebných informací a podobných řešení problémů, kterými se zabývá. Tomuto informačnímu kanálu je v LabVIEW věnována velmi významná pozornost. Dá se to poznat i intuitivním roztříděním témat a velkému množství příkladů.



Obr. 30 Úvodní obrazovka Getting Started



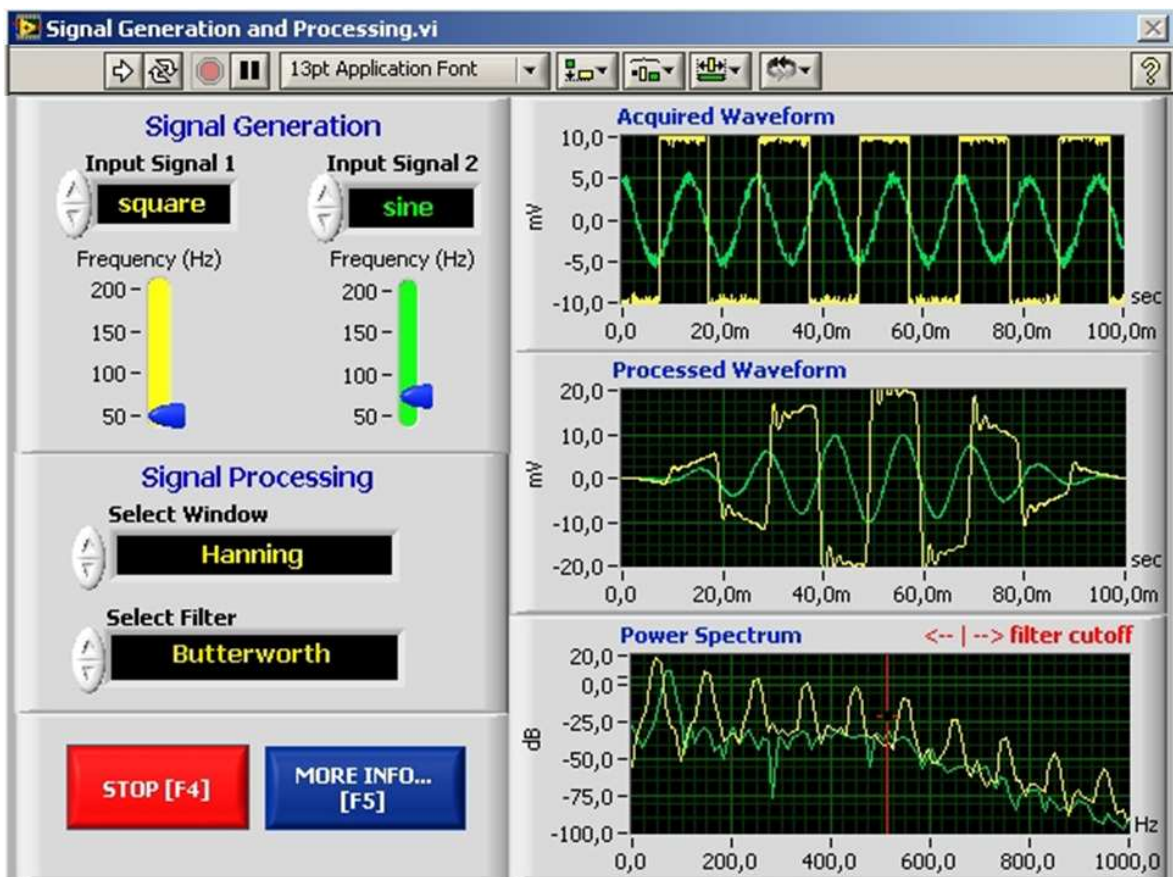
Obr. 31 Obrazovka hledání příkladů NI Example Finder

### 3.3 Uživatelské prostředí

V programu LabVIEW se po otevření nového nebo už existujícího VI otevrou vždy dvě sdružené okna. Čelní panel (Front Panel) a blokový diagram (Block Diagram). Čelní panel má většinou podobu panelu měřicího přístroje a blokový diagram určuje posloupnost signálu procházejícího bloky uvnitř přístroje.

### 3.4 Čelní panel (Front Panel)

Je tvořen ovládacími grafickými prvky, které jsou určeny hlavně k řízení VI (Obr. 32). Stejně jako na fyzickém měřicím přístroji jsou na čelním panelu virtuálního přístroje ovládací prvky, které představují vstupní zařízení. Jsou to různá tlačítka, přepínače a hodnoty nastavující knoflíky. Výstupy představují různé podoby zobrazovačů, kontrolky a různé podoby grafů.



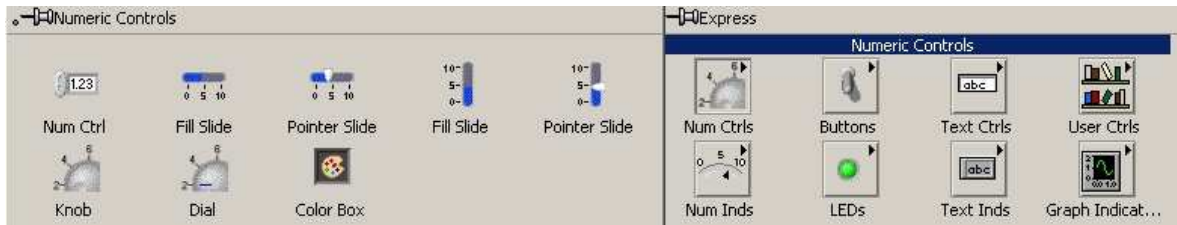
Obr. 32 Čelní panel

Nástrojová lišta je hlavním panelem pro ovládání a indikaci stavu čelního panelu (Obr. 33)



Obr. 33 Nástrojová lišta čelního panelu

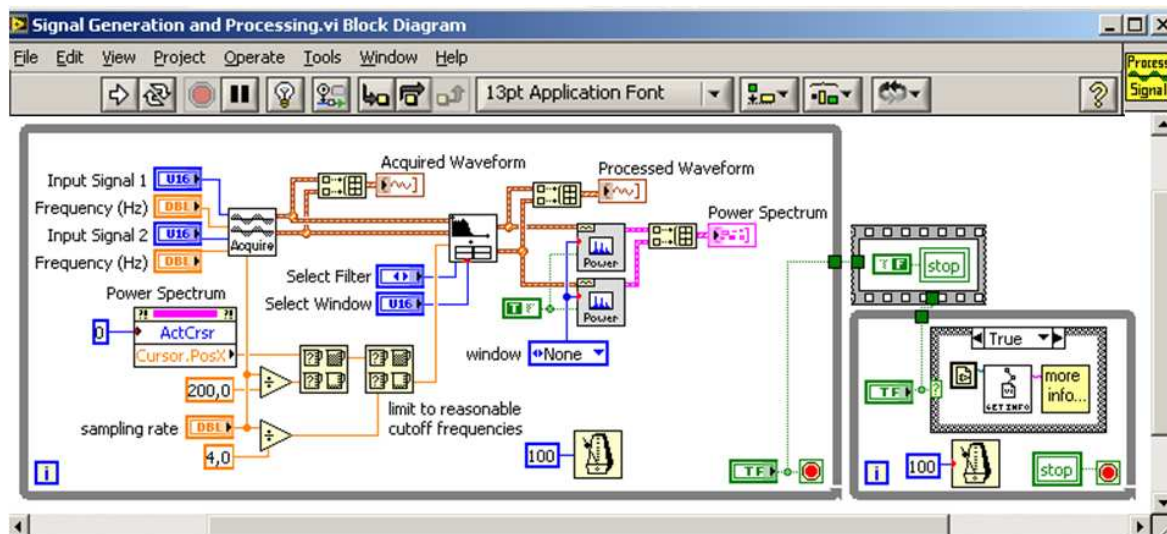
V okně čelního panelu se tedy vytváří vzhled virtuálního přístroje, všechny vstupní a výstupní prvky je možné získat nabídky palety Controls, kterou vyvoláme pravým kliknutím myši. Na Obr. 34 je rozbalená paleta Controls s podpaletou Numeric Controls.



Obr. 34 Otevřená knihovna Controls

### 3.5 Blokový diagram (Block Diagram)

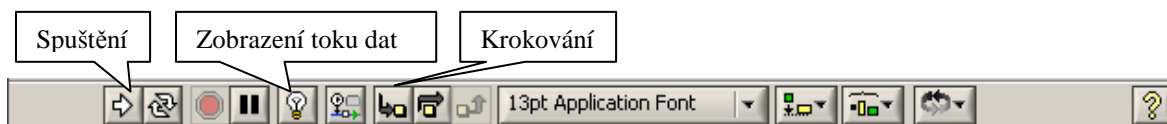
Na obrazovce blokového diagramu (Obr. 35) programátor vytváří vlastní algoritmus pomocí propojení předem nadefinovaných prvků z čelního panelu.



Obr. 35 Blokový diagram

Blokový diagram se vytváří z prvků palety Functions, kde se nacházejí struktury (smyčky), numerické funkce, práce se soubory, pole, logické funkce, funkce času a další.

Nástrojová lišta v blokovém diagramu je podobná liště čelního panelu, liší se ve funkcích pro spuštění programu a jeho testování (odladování) po krocích (Obr. 36).

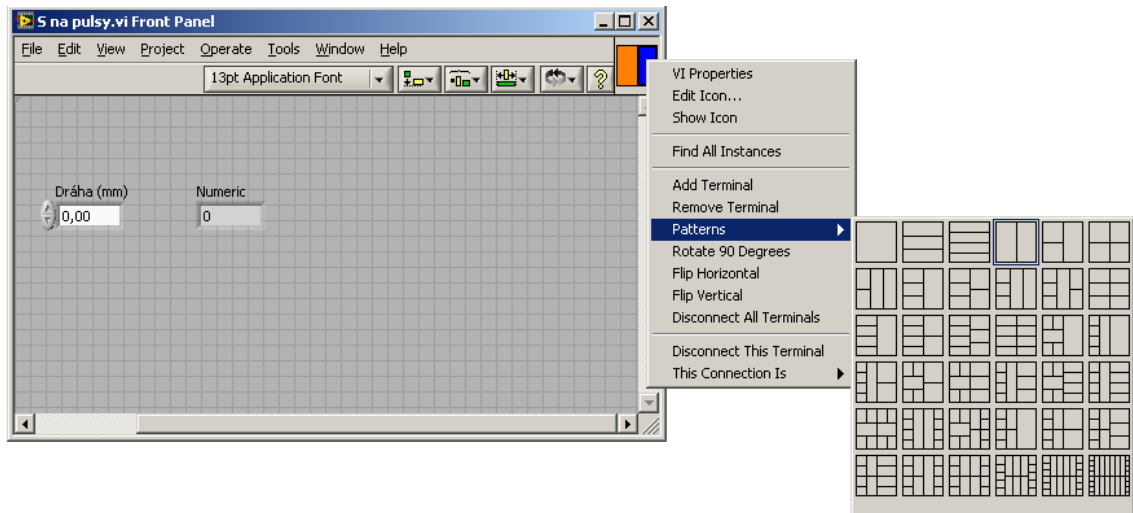


Obr 36 Nástrojová lišta blokového diagramu

Prvky blokového diagramu jsou řídicí terminály, spoje, uzly a zobrazovací terminály. Tyto prvky pospojované spoji tvoří přímo spustitelný kód, který je překládán během tvorby diagramu a po spuštění nám dává okamžitou zpětnou vazbu při vzniku chyby.

Uzel je prvek pro vykonání určité operace programu. Bývá to funkce, podprogram nebo příkaz. Např. uzel typu funkce umožní sčítání, násobení, porovnávání, práci se soubory a další. Uzel typu struktura se používají jako smyčky typu For, While a další. Terminály jsou vstupní a výstupní brány, přes které prochází data. Jsou to indikátory, grafy, přepínače, konstanty, ovládací prvky a další. Spojе jsou vlastně cesty, kterými prochází data a jsou současně propojovacími kanály mezi vstupními terminály, uzly a výstupními terminály. Každý spoj má svoji barvu, která odpovídá datovému typu proměnným (modrá - celočíselný, oranžová - číselný s pohyblivou čárkou, zelená - binární, fialová - znakový).

V pravém horním rohu blokového diagramu a čelního panelu se nachází ikona, která je současně konektorem. V nastavení konektoru se čtvereček rozdělí na takový počet vstupních a výstupních terminálů, kolik jich je v blokovém diagramu (Obr. 37) a se kterými chceme dále pracovat. Pracovat se dál dá s tímto VI po pospojování terminálů s konektorem jako se Sub VI, což znamená, že se Sub VI použije do dalšího VI jako podprogram (funkce).



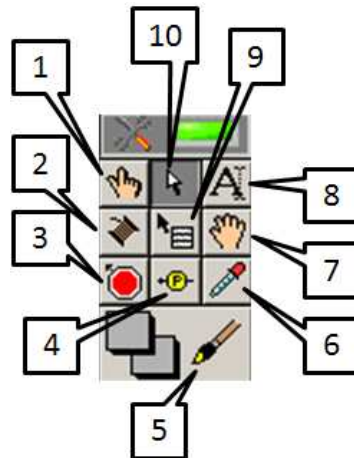
Obr. 37 Rozdělení konektoru na počet terminálů

## 3.6 Palety nástrojů

Při navrhování VI potřebujeme, jak už je výše napsáno, různé funkce a terminály, které stručně popíšu v této kapitole. Základní jsou paleta nástrojů, paleta ovládacích a zobrazovacích prvků pro čelní panel a paleta funkcí pro blokový diagram.

### 3.6.1 Paleta nástrojů (Tools)

Na ploše čelního panelu vyvoláme paletu nástrojů (Obr. 38) pomocí <shift> + pravé tlačítko myši. Paleta nástrojů slouží k usnadnění režimu práce s myší a jednoduššímu přístupu do menu.

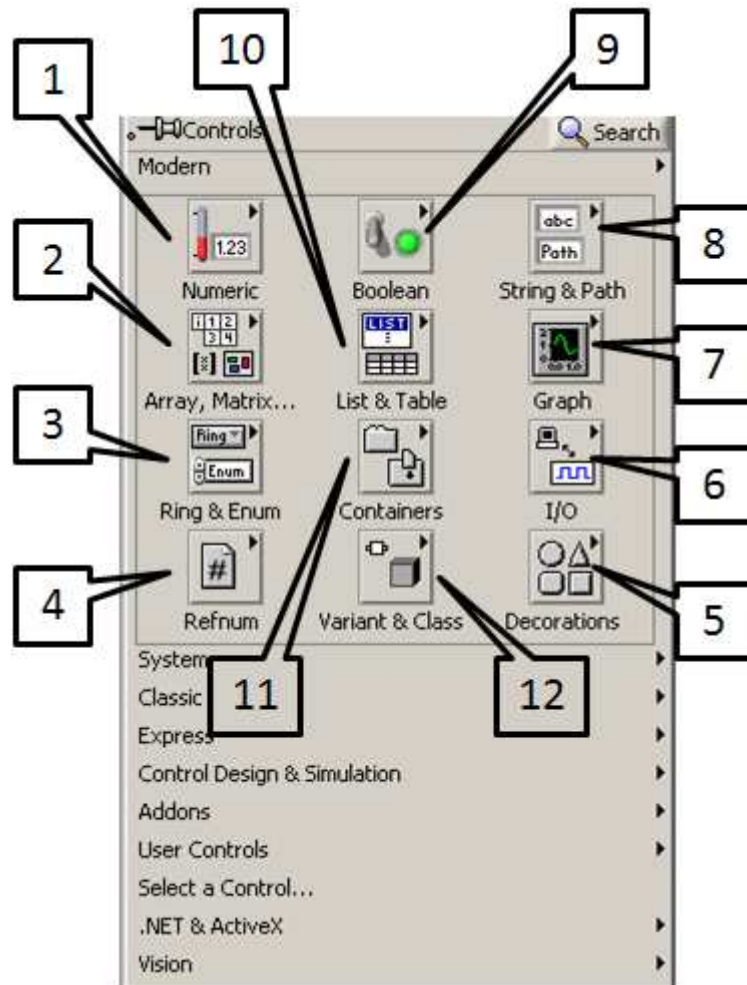


Obr. 38 Paleta nástrojů

Popis k obrázku Obr. 38 je následovný: 1 – operační nástroj pro změnu hodnot a textů, 2 – propojovací nástroj (spoje), 3 – vkládání přerušení, 4 – vložení sondy na datový vodič, 5 – barvy prvků, 6 – kopírování barvy, 7 – posouvání okna, 8 – textové popisování, 9 – nastavení rychlého menu, 10 – výběr, poloha, velikost.

### 3.6.2 Paleta ovládacích a zobrazovacích prvků (Controls)

Paleta Controls je spojena pouze s čelním panelem, takže v blokovém diagramu ji nevyvoláme. Vyvoláme ji pravým kliknutím myši na plochu čelního panelu. Na paletě Controls je mnoho dalších podpalet, které slouží k vytvoření grafické podoby čelního panelu. Obsahuje samozřejmě spoustu tlačítek, knoflíků, přepínačů, prvků pro zadávání vstupních hodnot, měřiče a v neposlední řadě také zobrazovače výstupních hodnot datového toku.

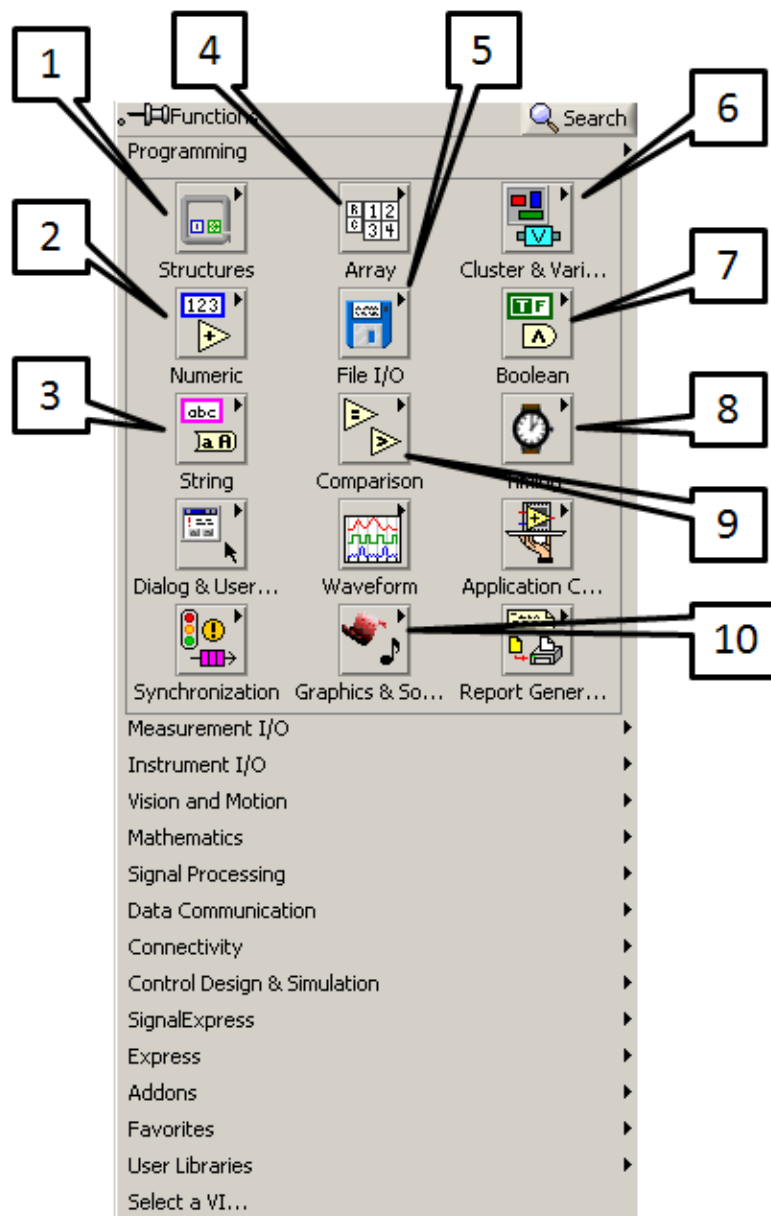


Obr. 39 Paleta Controls

Paleta controls obsahuje tyto podpalety (Obr. 39): 1 – ovladače a zobrazovače, 2 – pole, matice a klastry, 3 – rolovací ovládací prvky, 4 – prvky pro práci s odkazy, 5 – prvky různých geometrických tvarů, 6 - pro ovládání zařízení, 7 – grafické zobrazovače, 8 – řetězce a cesty k souborům, 9 – tlačítka a LED zobrazovače, 10 – seznamy a tabulky, 11 – datové kontejnery.

### 3.6.3 Paleta Functions

Práce s paletou Functions je obdobná jako v předchozím případě. Paletu je možné vyvolat kliknutím pravého tlačítka myši pouze z blokového diagramu.



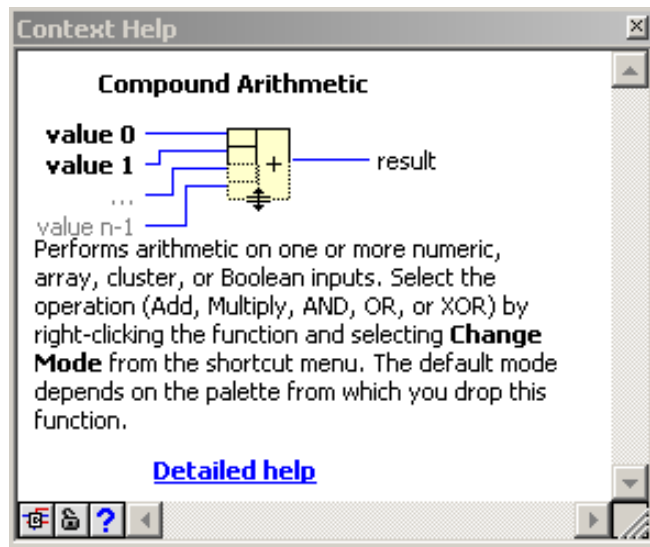
Obr. 40 Paleta Functions

Pomocí této palety je možno vytvořit tzv. zdrojový kód grafického programování. Paleta Functions obsahuje tyto nejdůležitější podpalety (Obr. 40): 1 – smyčky, 2 – numerické funkce, 3 – řetězce, 4 – pole, 5 – práce se soubory, 6 – klastry, 7 – logické funkce, 8 – funkce času, 9 – funkce porovnávání, 10 – grafické a zvukové funkce.

### 3.7 Context Help

Nesmím zapomenout na velmi vděčného pomocníka a tím je Context HELP (Obr. 41). Jedná se o okno, které v blokovém diagramu vyvoláme kliknutím myši na symbol žlutého otazníku vpravo nahoře (Obr. 35). Context Help nám po najetí myši na jakýkoli prvek

v blokovém diagramu zobrazí pomoc při jeho nastavení a popis funkce. V LabVIEW je tato možnost velmi dobře zpracovaná a proto se také hodně využívá.



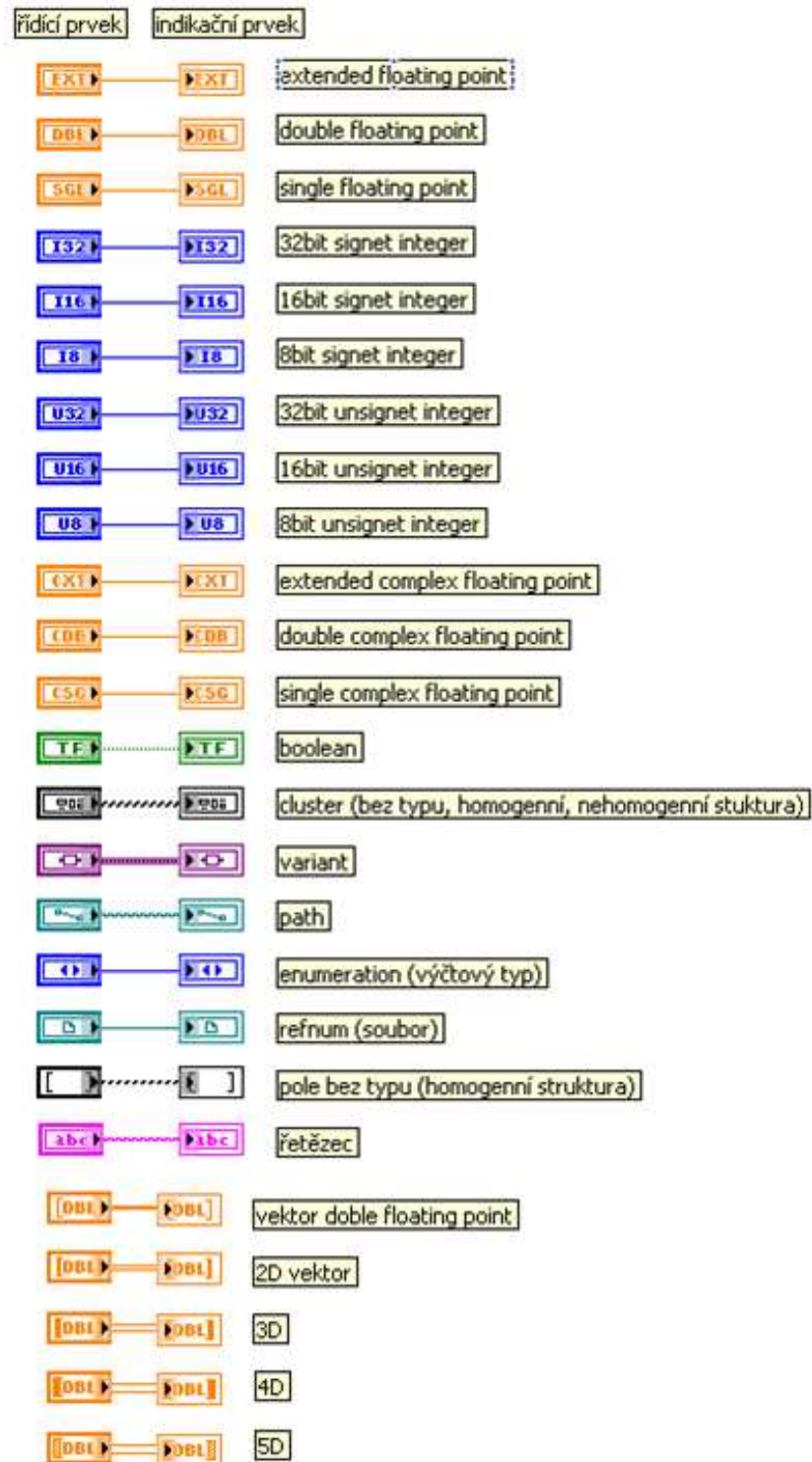
Obr. 41 Context Help

### 3.8 Datové typy

Na blokovém diagramu jsou prvky jako funkce a terminály propojené spoji, tyto spoje jsou různých datových typů, stejných jako jsou vstupy a výstupy z funkcí a terminálů. Takže je zřejmé, že nelze spojovat terminály dvou různých datových typů jedním datovým spojem (Obr. 42). Na obrázku (Obr. 43) jsou uvedeny všechny datové typy, které LabVIEW 8.2 umožňuje používat.



Obr. 42 Nemožný spoj



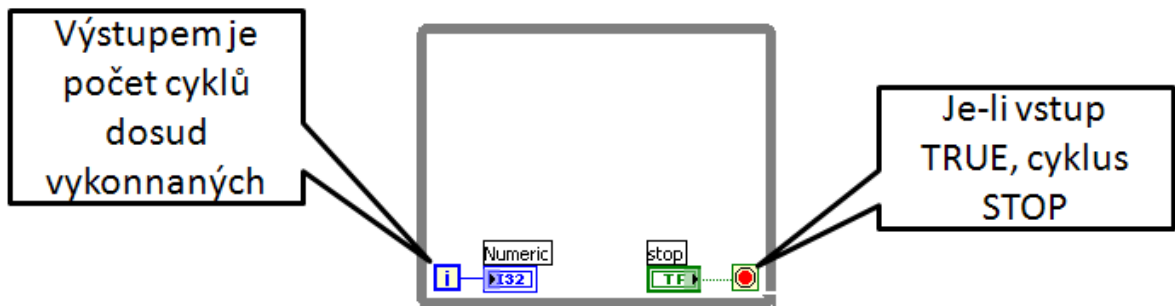
Obr. 43 Datové typy

### 3.9 Programové struktury

Jsou to struktury pro řízení průběhu části programu, používají se velmi často a jsou podobné cyklům v textových programovacích jazycích. Do blokového diagramu se vkládají z vyvolané palety Functions (Obr. 40), podpalety Structures.

#### 3.9.1 Cyklus While

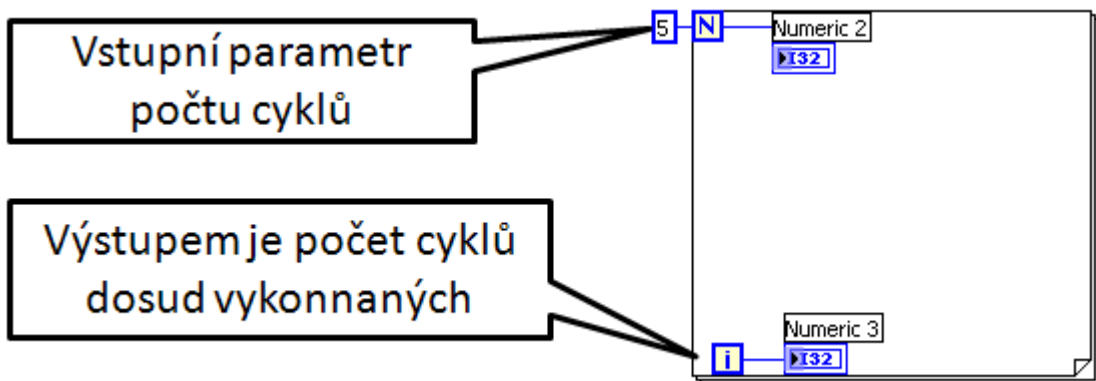
Používá se k opakování funkcí uvnitř cyklu po dobu platnosti.



Obr. 44 Cyklus While

#### 3.9.2 Cyklus For

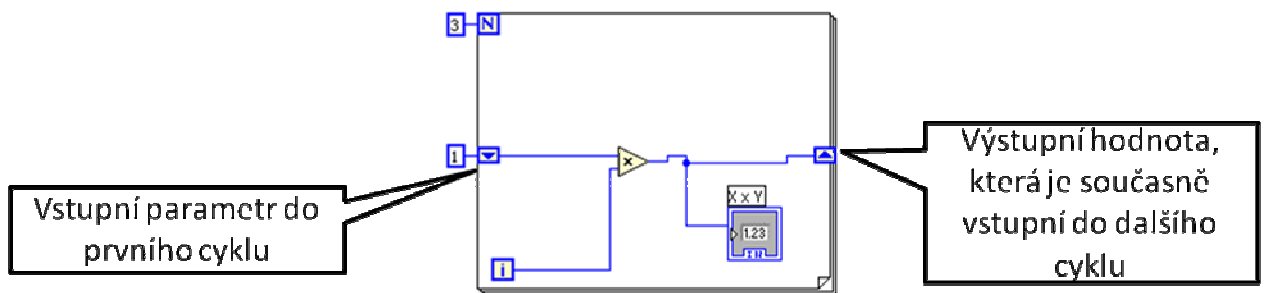
Také se používá k opakování funkcí uvnitř umístěných. Počet opakování je jasně daný nebo proměnný podle vstupního parametru.



Obr. 45 Cyklus For

### 3.9.3 Posuvný registr

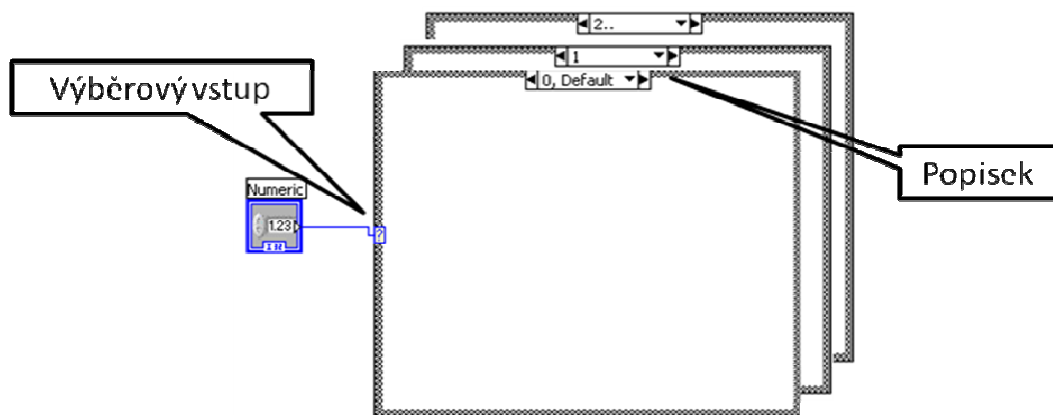
Aplikuje se na cyklus For (Obr. 46) nebo na cyklus While. Pravým kliknutím myši na boky obrysu cyklu For nebo While vybereme Add Shift Register. Po stranách se přidají dvě šipky (vstupní hodnota a výstupní hodnota). Cyklus For nyní pracuje stejně jako bez posuvného registru, pouze si navíc přenáší výstupní hodnotu do dalšího cyklu jako vstupní hodnotu.



Obr. 46 Cyklus For s posuvným registrem

### 3.9.4 Větvení Case

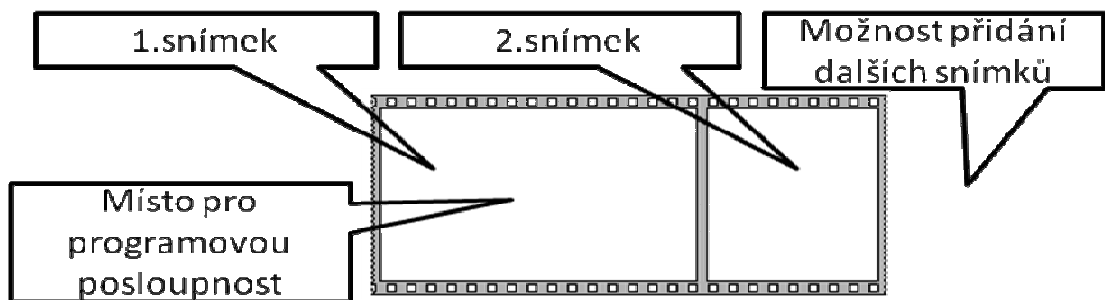
Používá se při možnosti více řešení při různých vstupních podmínkách. Na vstup je přiveden tzv. přepínač, podle kterého se provede to okno, které má v popisku shodnou hodnotu.



Obr. 47 Struktura Case

### 3.9.5 Sequence

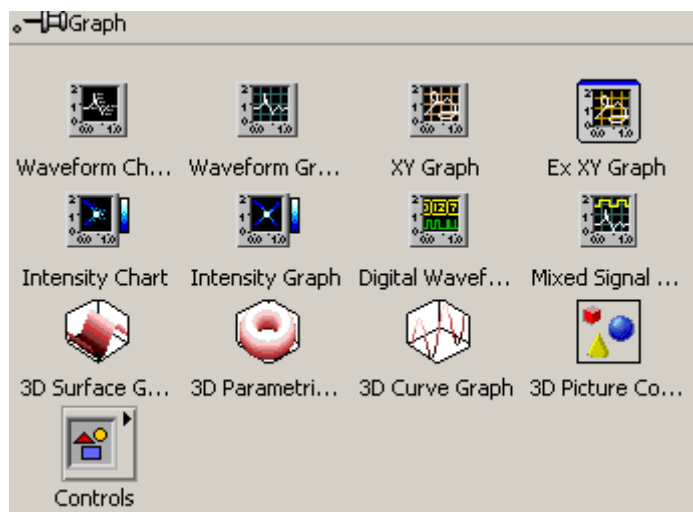
Umožňuje postupné spouštění sekvencí programu rámeček po rámečku postupně zleva doprava, podobně jako snímky filmu. Používá se hlavně v případech, kdy nejsou objekty spojeny tzv. drátkem, ale chceme je spouštět v určité posloupnosti.



Obr. 48 Sequence

### 3.10 Grafické zobrazovače

Základní jsou tři typy: Waveform Chart, Waveform Graph a XY Graph. Všechny nalezneme na paletě Graph (Obr. 49).

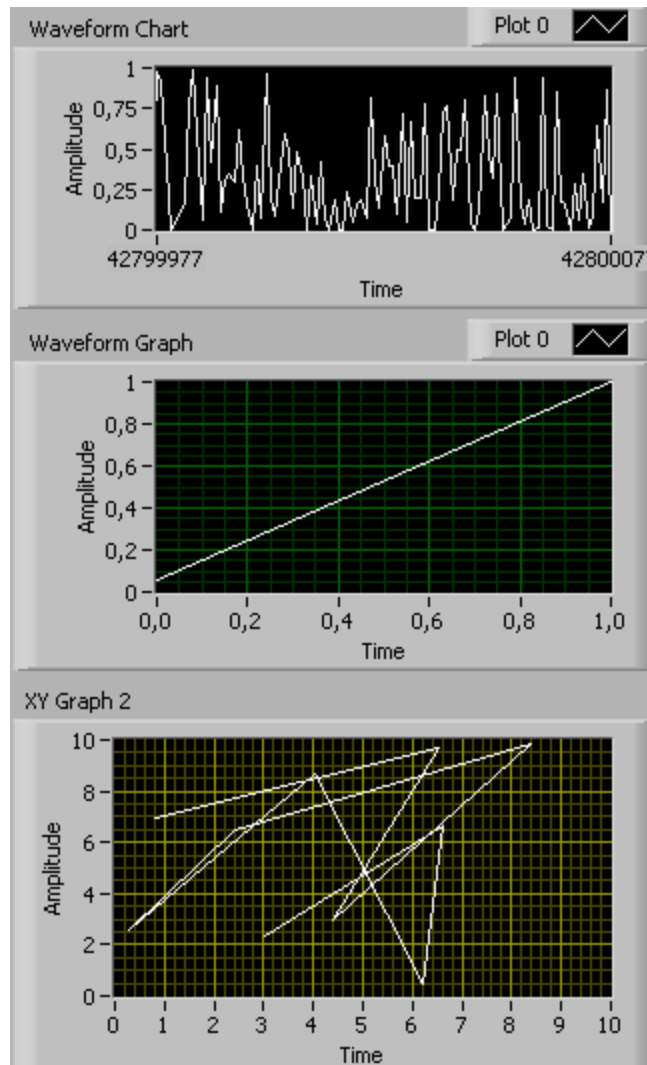


Obr. 49 Paleta Graph

Zobrazovač WaveformChart je průběžným zapisovačem se třemi druhy zobrazování hodnot dat. Waveform Graph je uzpůsoben pro zobrazování datových polí. Například pro

zobrazení vzorkování z DAQ zařízení. XY Graph je schopný zobrazovat dvě veličiny ve vzájemné závislosti nebo je určen pro data přístupná v nepravidelných intervalech.

Z těchto zobrazovačů lze grafické průběhy exportovat a ukládat do různých formátů (.bmp, .eps, .emf).



Obr. 50 Typy zobrazovačů

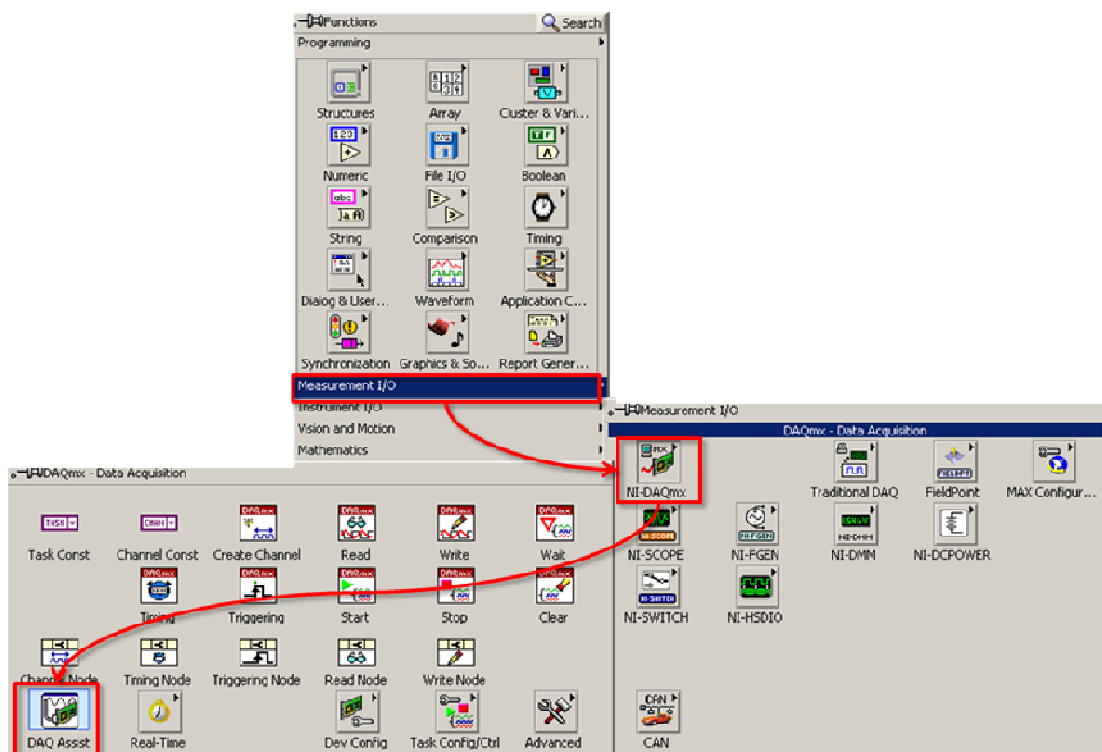
### 3.11 Tok dat v programu

Datový tok je v LabVIEW přesun dat pomocí spojů od vstupních terminálů přes funkce až k výstupním terminálům. Jakmile je splněna podmínka všech nezbytných vstupů různých prvků programu, okamžitě jsou na výstupu výsledná data. Takto postupuje datový tok až ke konci programu. Není to tedy stejné, jako když se v textových programovacích jazycích postupuje řádek po řádku.

### 3.12 DAQ

Význam DAQ je získávání dat z venkovního prostředí. Pomocí snímačů fyzikálních veličin se do DAQ zařízení dostává signál, který je dál zpracováván. DAQ zařízení jsou různé zásuvné karty do PC, zásuvné moduly do průmyslových terminálů a další, na bázi několika rozhraní (ISA, PCI, PCMCIA, PXI, ...).

Pro konfiguraci DAQ zařízení se v OS Windows používá průvodce se zkratkou MAX (Measurement & Automation Explorer). Veškeré informace o zařízení, které získá z ovladače, zapíše do registrů a poté je již zařízení rozpoznáno v LabVIEW pod určitým jménem. Nyní můžeme nakonfigurovat zařízení podle požadavků úlohy. V LabVIEW vyvoláme obrazovku konfigurace analogových vstupů a výstupů a digitálních vstupů a výstupů DAQ Assist a poté už můžeme vytvořit program pro snímání, zobrazování a řízení.

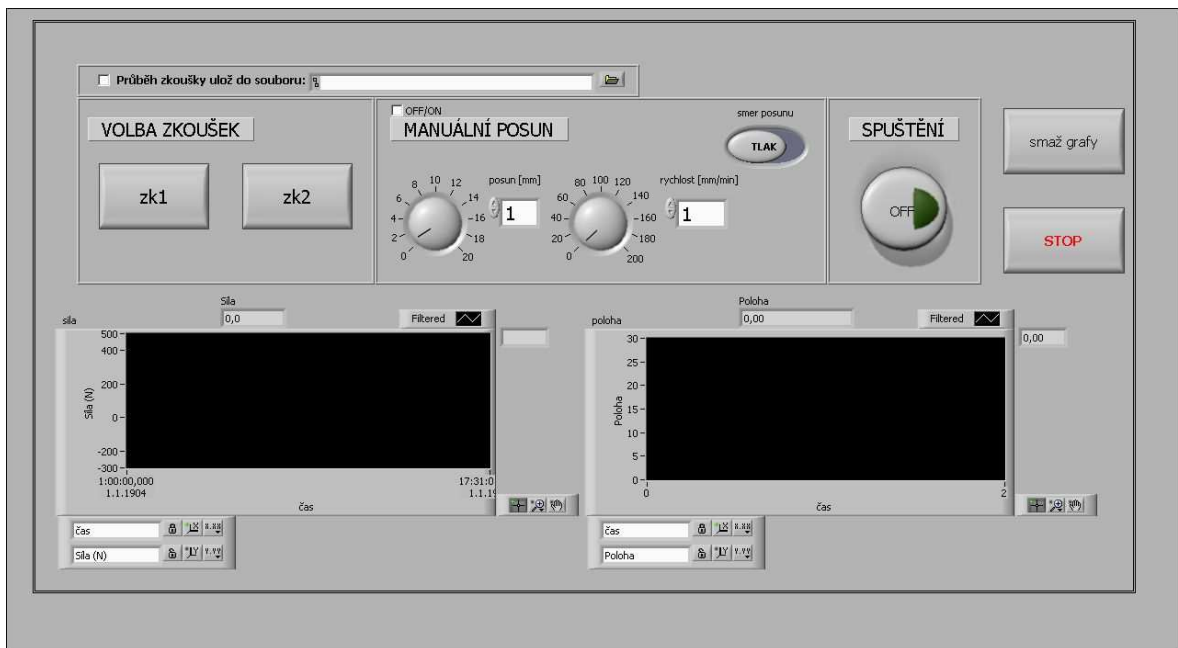


Obr. 51 DAQ Assist

## **II. PRAKTICKÁ ČÁST**

## 4 POPIS ŘÍDÍCÍ A VIZUALIZAČNÍ APLIKACE PRO ZKOUŠKY ASFALTOVÝCH ZÁLIVEK

V odstavci 1.8 je uvedeno, že na HOST PC běží uživatelské rozhraní pro konfiguraci parametrů měření a řízení. Na obr. 52 je zobrazen celý tento ovládací panel pro obě zkoušky asfaltových zálivek, manuální režim posunu motoru a možnosti uložení požadovaných hodnot do souboru.



Obr. 52 Uživatelské rozhraní

### 4.1 Volba uložení do souboru

V horní části uživatelského rozhraní je možnost uložení průběžných dat do souboru. V LabVIEW je možné měřené data ukládat do souboru ve formátech ze tří základních způsobů. Textový datový tok, binární datový tok a datalog (složený z řetězců, čísel a klastrů). Pro možnost nejjednoduššího otevření a možnosti práce s grafy jsem vybral textový soubor. Checkbox vlevo od nápisu uložení do souboru způsobuje aktivaci a deaktivaci systému ukládání do souboru. V případě zaškrtnutí checkboxu můžeme kliknout na obálku a vytvořit nové jméno souboru, v případě kdy se nezadá jméno souboru, ale checkbox je zaškrtnut, je v aplikaci nastaven základní název FILE.txt a data se uloží do složky, kde je uložena aplikace. V případě nezaškrtnutí checkboxu zředne text s rámečkem a obálkou. Do souboru se v tomto případě nezapisuje.

## 4.2 Volba manuálního posunu motoru

Nad textem „ MANUÁLNÍ POSUN “ je opět další checkbox, který má za úkol aktivaci a deaktivaci možností v tomto rámečku nabídnutých. Jsou to nastavení délky posunu, rychlosti a směru motoru.

Možnost manuálního posunu se využívá k uchycení zkušebních vzorků a najetí na přesnou polohu, na tzv. nulovou. Každý vzorek je v toleranci jinak široký, proto nemůžeme nastavit uchycení vždy do stejné nulové polohy.

Spuštění manuálního režimu probíhá tak, že nejprve aktivujeme zadávání hodnot zaškrtnutím checkboxu nad textem „MANUÁLNÍ POSUN „. Pak nastavíme posun o kolik mm chceme aby se uchopovací čelisti rozevřeli nebo uzavřeli. Dále pak rychlost v mm/min a nakonec směr posunu. Jsou dvě možnosti TAH (čelisti se posunují od sebe) nebo TLAK (čelisti se zavírají do sebe). Samozřejmě je vše jistiště nastavením maximální možné hodnoty přípustné síly, aby nedošlo k poškození částí testovacího zařízení. Po nastavení se může spustit manuální posun tlačítkem v rámečku „ SPUŠTĚNÍ “. Má dvě polohy, OFF a ON. Doporučuji nastavit rychlost na minimální hodnoty a v případě požadavku na předčasné ukončení stačí opět zmačknout tlačítko spuštění. Posun se zastaví. Pokud se nechá spuštěný manuální posun, motor se zastaví po dosažení nastavené vzdálenosti a celý manuální posun se odškrtně v checkboxu a zšedne. V případě nenadálé události je zde ještě tlačítko STOP, které zastaví veškerou aktivitu testovacího zařízení.

## 4.3 Zkouška 1

Po usazení zkušebního vzorku a najetí čelistí do nulové polohy se může spustit první zkouška. Tato zkouška se nastaví stlačením přepínače „zk1“ a spuštění probíhá stejně jako v případě manuálního režimu tlačítkem „ SPUŠTĚNÍ “. Stejně tak jako v předchozím případě se dá zkouška předběžně ukončit opětovným zmačknutím tlačítka spuštění do polohy OFF. Tlačítko STOP zastaví okamžitě veškerou činnost testovacího zařízení.

## 4.4 Zkouška 2

Druhá zkouška se spouští po nastavení stlačením přepínače „zk2“ spouštěcím tlačítkem do polohy ON. Není možné stlačit dvě tlačítka současně, vždy může být stlačeno pouze jedno z předchozích tří možností (checkbox manuální posun, tlačítko zk1 nebo tlačítko zk2).

## 4.5 Grafy

V uživatelském panelu jsou ještě dva důležité objekty a těmi jsou grafy a možnost jejich smazání. Tlačítko „smaž grafy“ se používá v případě požadavku na vyčištění dvou zobrazovačů grafů. Při manuálním režimu se totiž také zobrazuje, jaká je v reálném čase síla, která působí na čelisti a poloha v závislosti na čase. Takže graficky můžeme pozorovat, jak rychle se nám zavírá nebo otevírá uchycení pro zkušební vzorek. Samozřejmě na začátku obou zkoušek se grafický výstup automaticky smaže.

S panely grafů se dá dále pracovat, je zde několik ikon, které nám slouží pro lepší analýzu grafického průběhu. Například zvětšení některého z míst, tzv. lupou. A možnost posunutí grafu ikonou ruky.

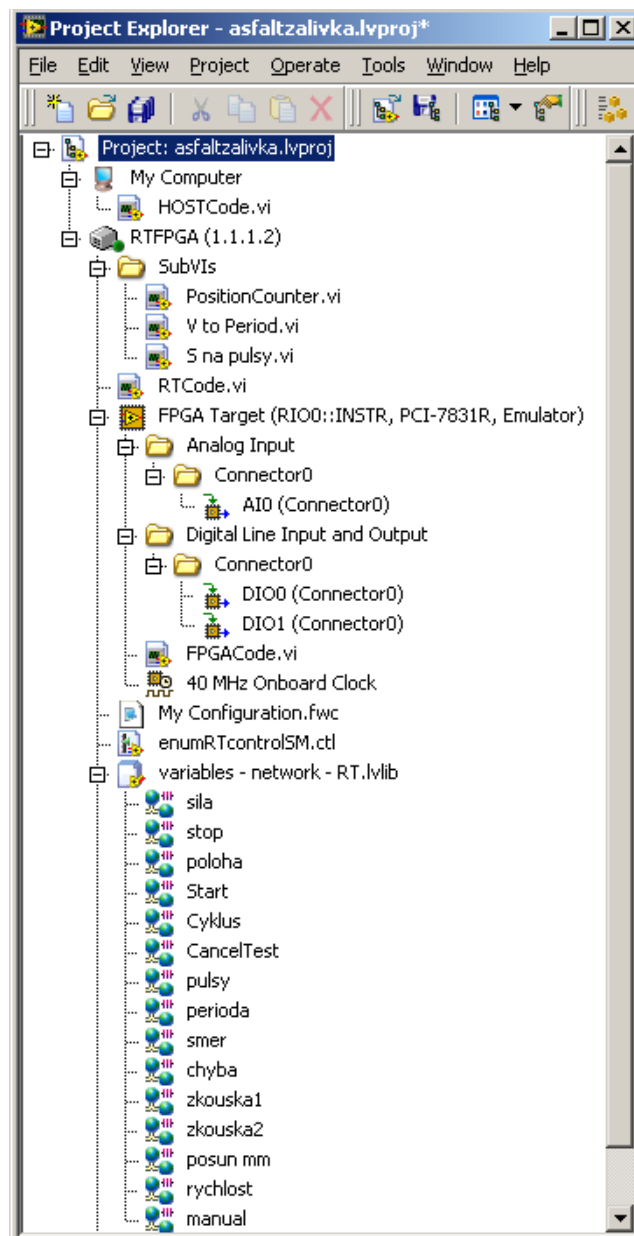
Nesmíme zapomenout na možnost vytvoření grafu průběhů zkoušek na základě uložených dat do textového souboru v některém z tabulkových editorů.

## 5 NÁVRH A POPIS JEDNOTLIVÝCH ČÁSTÍ PROJEKTU V NI LabVIEW

V NI LabVIEW se real-time projekt skládá ze tří nejdůležitějších souborů VI. Na obrázku Obr. 53 jsou pod názvem HOSTCode.vi, RTCode.vi a FPGACode.vi.

Dále se projekt dělí na dvě části, My Computer a RTFPGA. Jak už jsem výše psal, jsou to dva spolu komunikující PC, na kterých běží různé aplikace, jedná se o Host PC a Real-Time PC.

Vše, co se týká projektu, popíšu v této části práce.



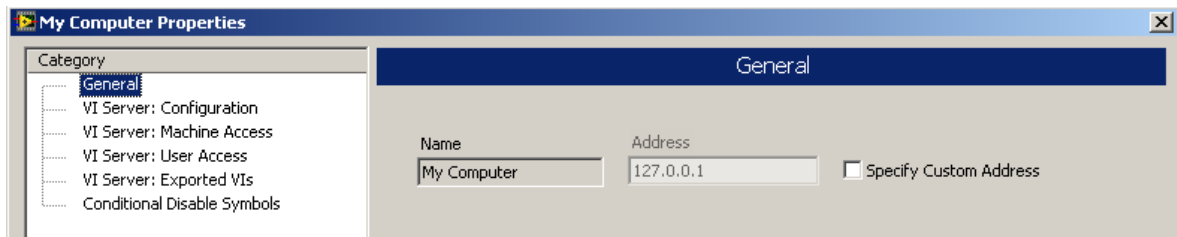
Obr. 53 Výpis částí projektu

## 5.1 Host PC (My Computer)

Na Host PC běží uživatelské rozhraní. Je to jeden ze tří nedůležitějších souborů – HOSTCode.vi. Protože musí být správně nastavena komunikace mezi Host PC a Real-Time PC, je třeba provést nastavení.

### 5.1.1 My Computer

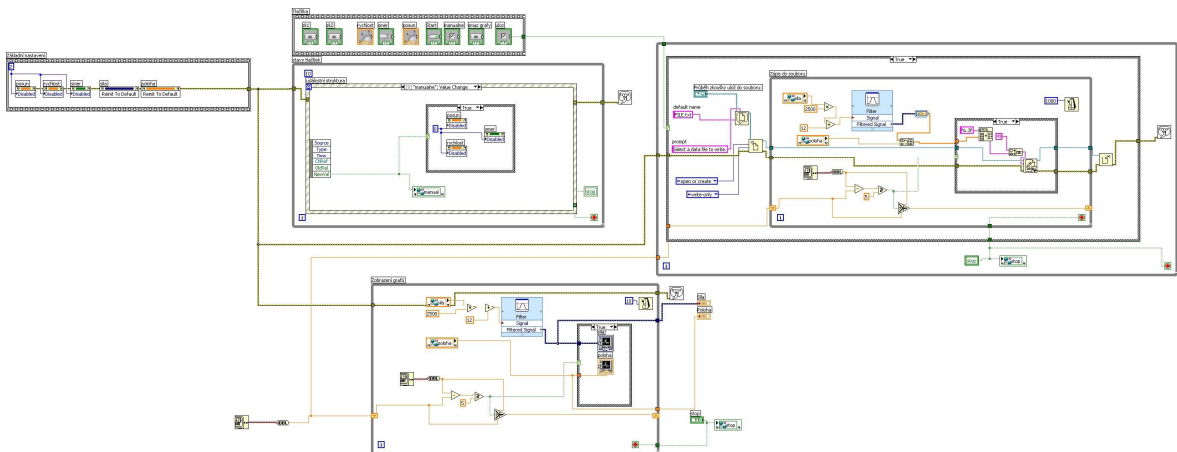
Nastavení se provede v menu výběrem položky Properties, kde se zadá IP adresa (Obr. 54).



Obr. 54 Nastavení My Computer

### 5.1.2 HOSTCode.vi

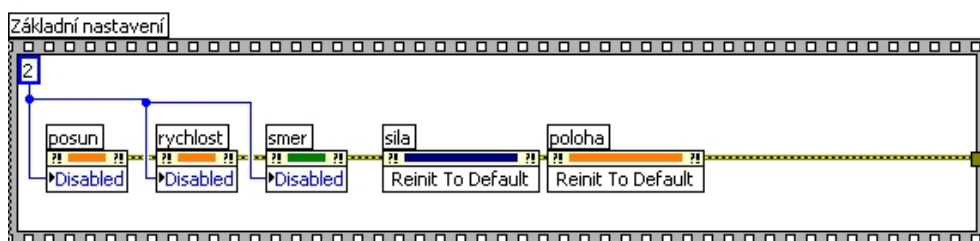
Virtuální nástroj HOSTCode.vi se skládá z části čelního panelu (Obr. 52) a blokového diagramu (Obr. 55), ve kterém se aplikace vyvíjí. Rád bych popsal a vysvětlil logiku jednotlivých částí. Jsou čtyři a to tyto: základní nastavení, tlačítka a stavy tlačítek (událostní struktura), zobrazení grafů a zápis do souboru.



Obr. 55 Blokový diagram

### 5.1.2.1 Základní nastavení

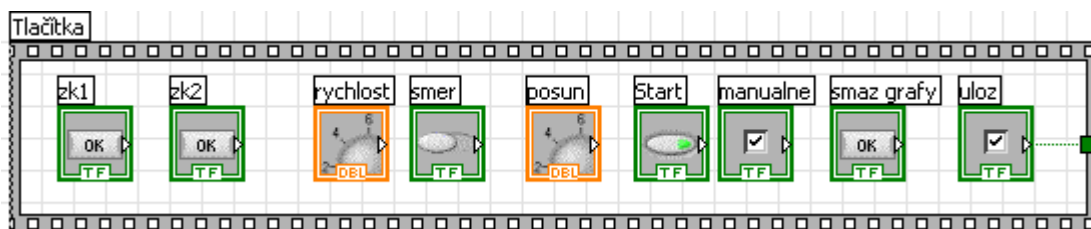
Před zahájením chodu aplikace je třeba nastavit základní hodnoty několika proměnných, nastavení vzhledu otočných knoflíků pro nastavení rychlosti a posunu v manuálním režimu a tlačítka pro změnu směru posunu motoru. Tyto mám zadané jako disable s volbou konstanty 2, což znamená zašednutí ovládacích prvků a zakázání jejich momentálního používání. Všechny objekty jsou uzavřené v sekvenční struktuře z jednoho rámce a jsou postupně propojeny drátky pro zachování toku informací a jejich posloupnosti. Žlutý čárkovaný drátek si sebou nese informace o chybových stavech, které jsou na konci posílané do možného vyvolání chybového hlášení. Dále pak prochází datový tok do další části, stavů tlačítek (událostní struktury).



Obr. 56 Základní nastavení

### 5.1.2.2 Stavů tlačítek (událostní struktura)

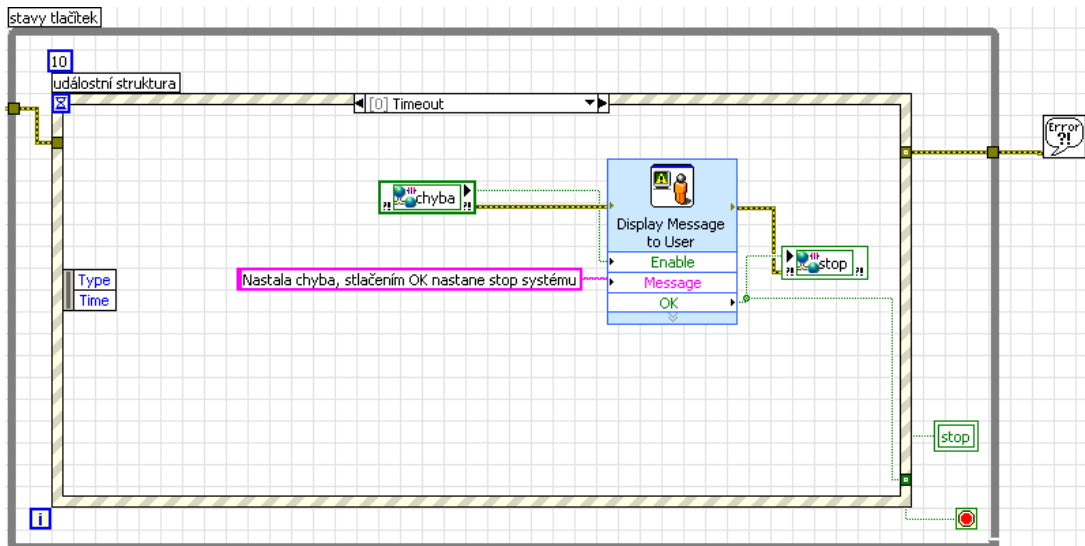
Tato část se dělí na smyčku stavů tlačítek a samotných tlačítek. Jsou to všechny tlačítka (Obr. 57), které jsou v uživatelském rozhraní a mají dané funkční události, které mohou nastat.



Obr. 57 Tlačítka

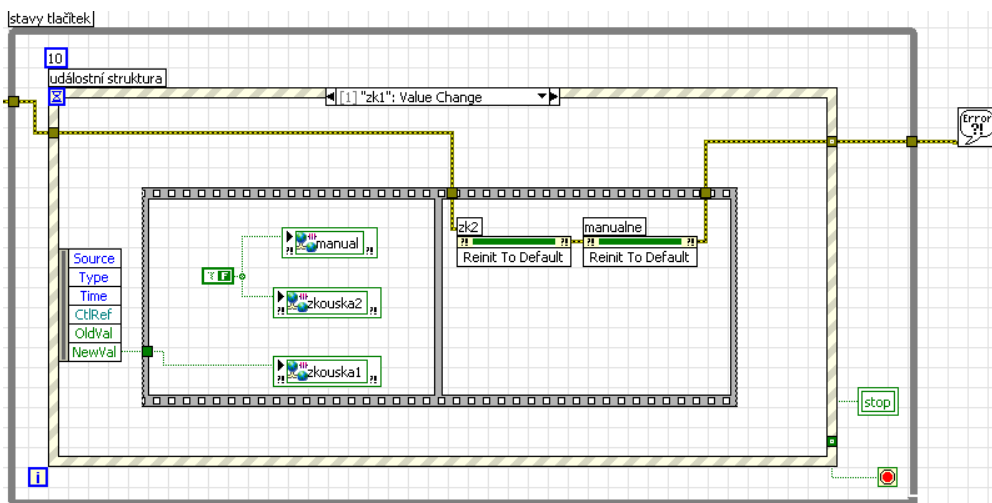
Pod sekvenční strukturou o jednom rámci všech tlačítek, checkboxů a otočných knoflíků je smyčka událostní struktury Event Structure, která je rozdělená do devíti oken. Každé okno nám zajišťuje jednu událost, která nastane změnou hodnoty zmáčknutím jednotlivého tlačítka. Je zde ještě jedno okno, které zjišťuje chybový stav (Obr. 58). Zde se v případě

chyby, která se čte ze sdílené síťové proměnné, objeví hlášení „Nastala chyba, stlačením OK nastane STOP systému“. Sdílené síťové proměnné se používají mezi Host PC a Real-Time PC, které mezi sebou komunikují na základě VI souborů HOSTCode.vi a RTCCode.vi, jejich zdrojových kódů a síťového nastavení.



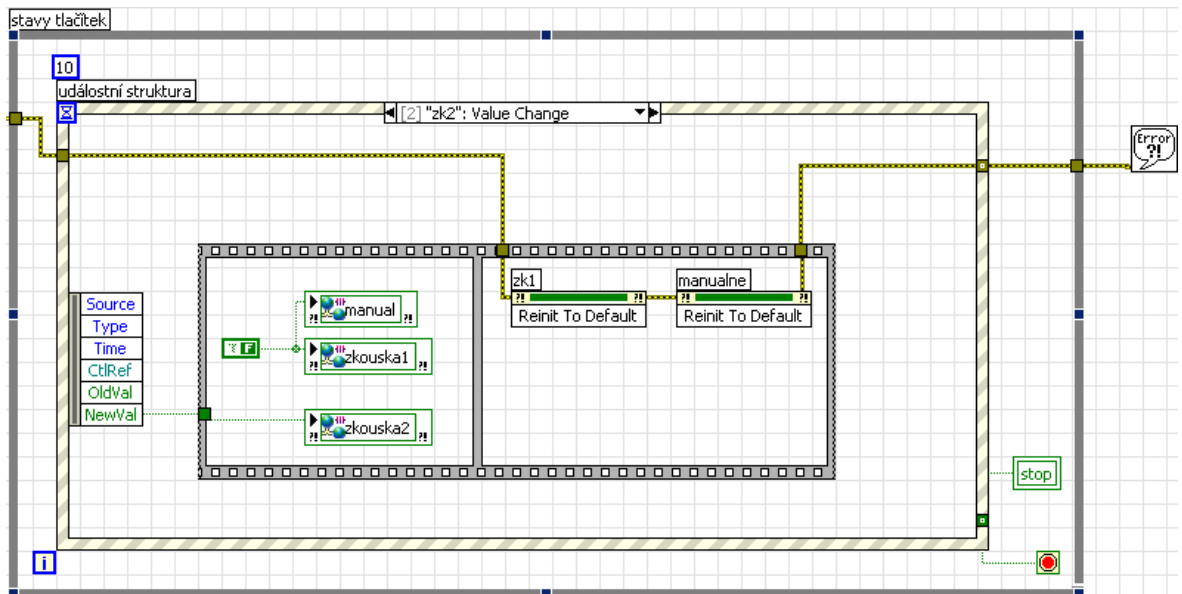
Obr. 58 Událost Timeout

První tlačítko a jeho událost (Obr. 59) je „zk1“. Při stlačení tlačítka do stavu s logickou 1 se do sdílených síťových proměnných „zkouska2“ a „manual“ uloží logické stavy 0 a do „zkouska1“ se uloží nový stav logické 1. Z těchto proměnných se poté na Real-Time PC čtou tyto stavy. V druhém rámci sekvenční struktury se pro jistotu ještě přenastaví vizuálně a stavově na Host PC tlačítka pro výběr manuálního režimu a zkoušku1 do základního nastavení, tedy logická 0.



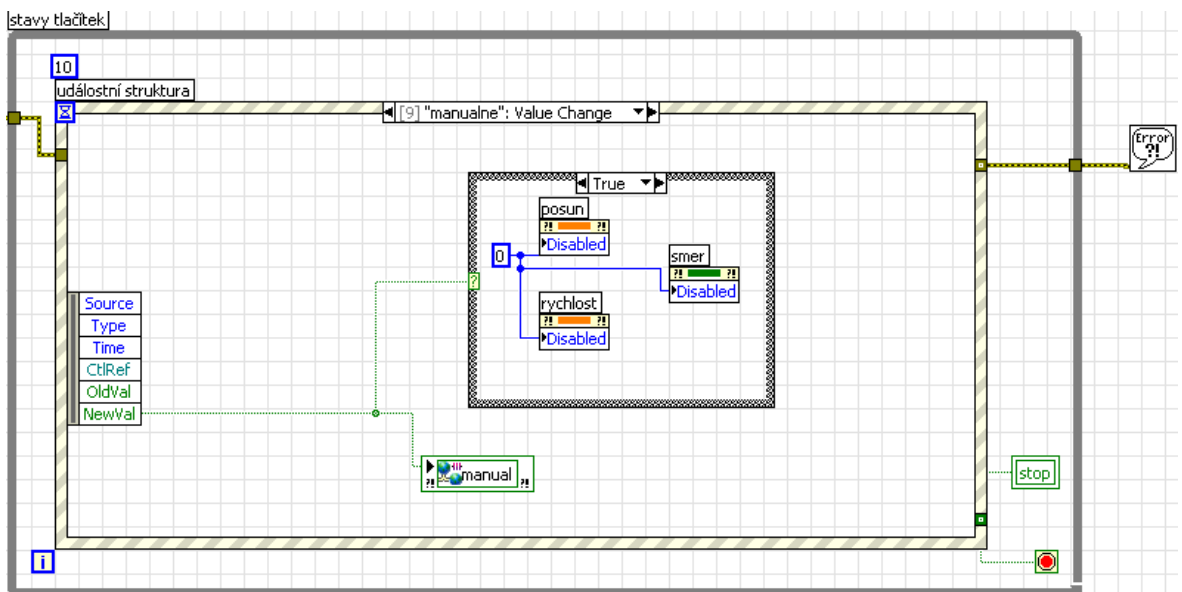
Obr. 59 Událost tlačítka „zk1“

Tlačítko „zk2“ má stejnou logiku události (Obr. 60), která nastane, jako u tlačítka „zk1“. Opět se řeší ostatní tlačítka pro zkoušku a manuální režim.



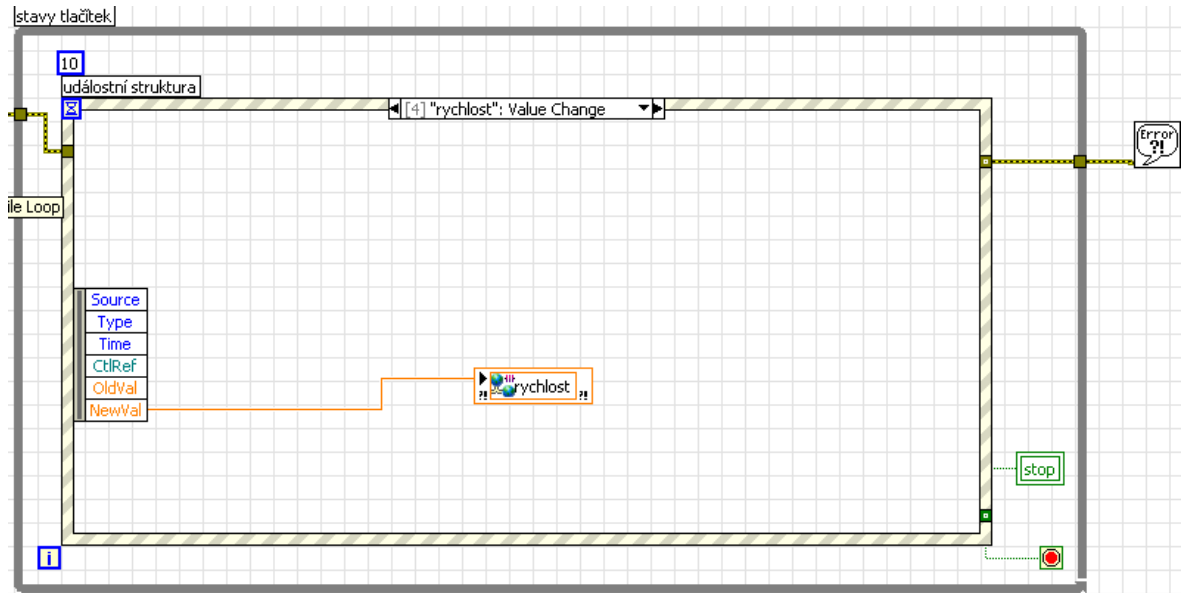
Obr. 60 Událost tlačítka „zk2“

Třetí tlačítko, neboli checkbox „manualne“ hlídá stav manuálního posunu motoru. Po události, která nastane při zaškrtnutí (Obr. 61) checkboxu „manualne“, se uloží do sdílené síťové proměnné logická 1 a současně se povolí použití všech tří možností nastavení pro posun, rychlost a směr. V opačném případě se možnost přístupu do okna manuálního posunu zakáže.



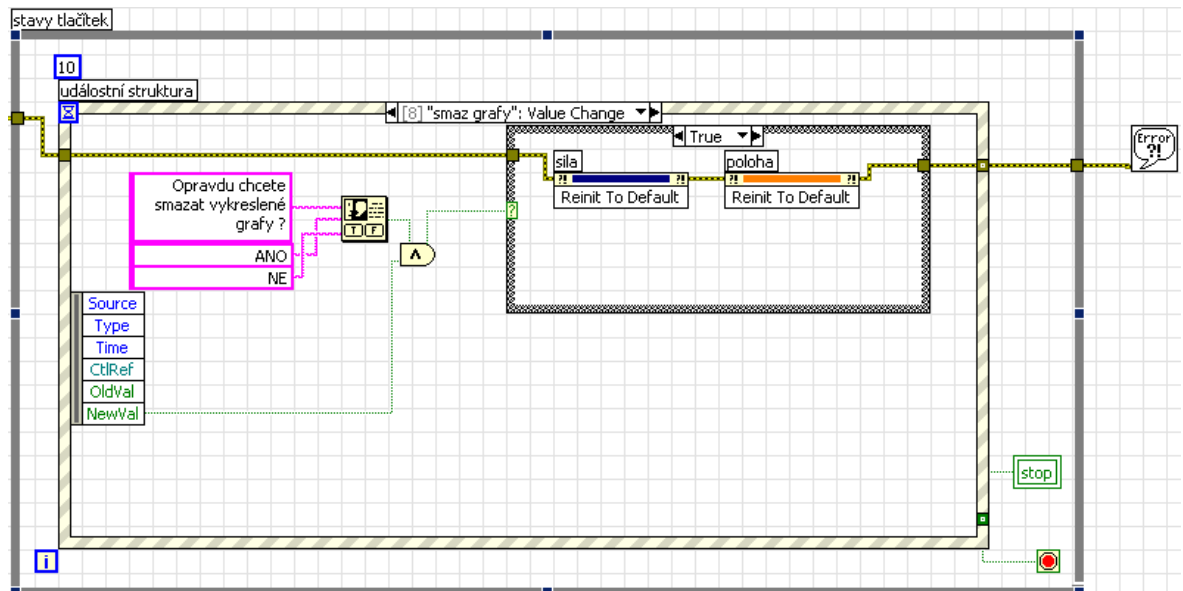
Obr. 61 Událost manuálního režimu

Otočné knoflíky „rychlost“, „smer“ a přepínač „posun“ mají každý zvlášť svou reakci na vzniklou událost (Obr. 62). Nastavením na požadovanou hodnotu se pouze tato hodnota přepíše do každé z připadající sdílené síťové proměnné.



Obr. 62 Událost tlačítka „rychlost“

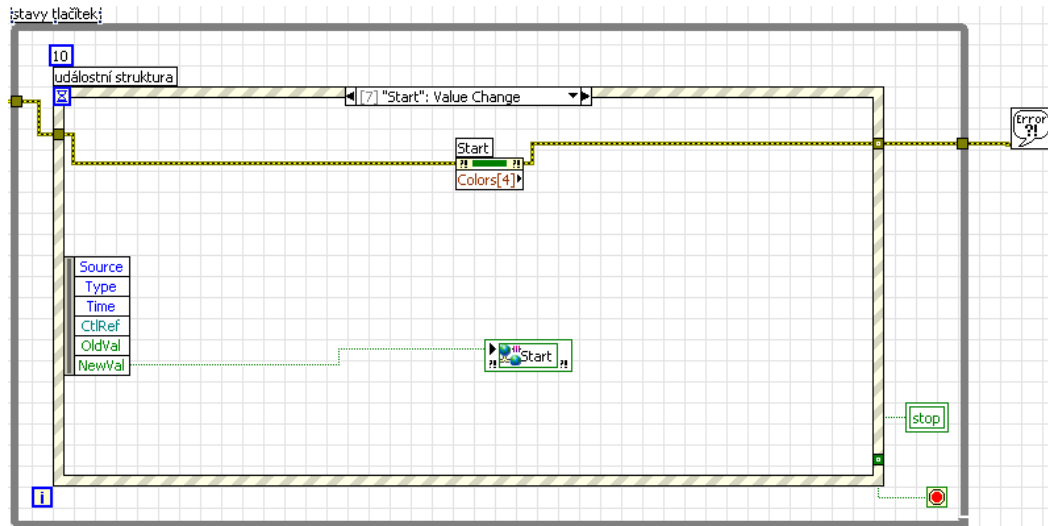
Událost na stav tlačítka „smaž grafy“ reaguje otevřením dialogového okna s požadavkem potvrdit smazání grafů v obou ze zobrazovačů (Obr. 63).



Obr. 63 Událost tlačítka „smaž grafy“

Další z tlačítek je v okně spuštění. Na vzniklou událost se reaguje uložením logického stavu tlačítka do sdílené síťové proměnné „start“ a současně se nastavuje jeho barevné

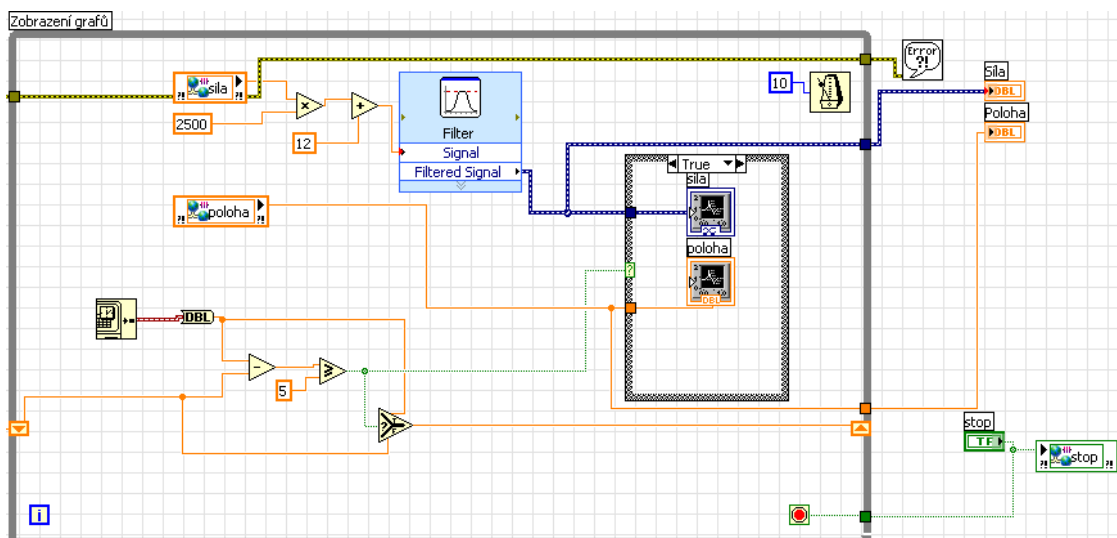
popředí a pozadí (Obr. 64). Poslední tlačítko „STOP“ ukládá logický stav do sdílené síťové proměnné „stop“ a současně do lokální proměnné „stop“, která má za úkol zastavit chod RTControl.vi.



Obr. 64 Událost tlačítka „start“

### 5.1.2.3 Zobrazení grafů

Část zobrazení grafů je uzavřena ve smyčce obnovující se každých 10ms. Z Real-Time PC se čte přes sdílenou síťovou proměnnou „síla“ hodnota síly z FPGA karty a posílá se přímo přes filtr do zobrazovače grafu. Tento ji zobrazuje, pokud je rozdíl mezi předchozí hodnotou a současnou hodnotou systémového času ve smyčce 5s. Stejně tak se zobrazuje poloha motoru, ale ta se počítá od nulové polohy samostatným SubVI (PositionCounter.vi).



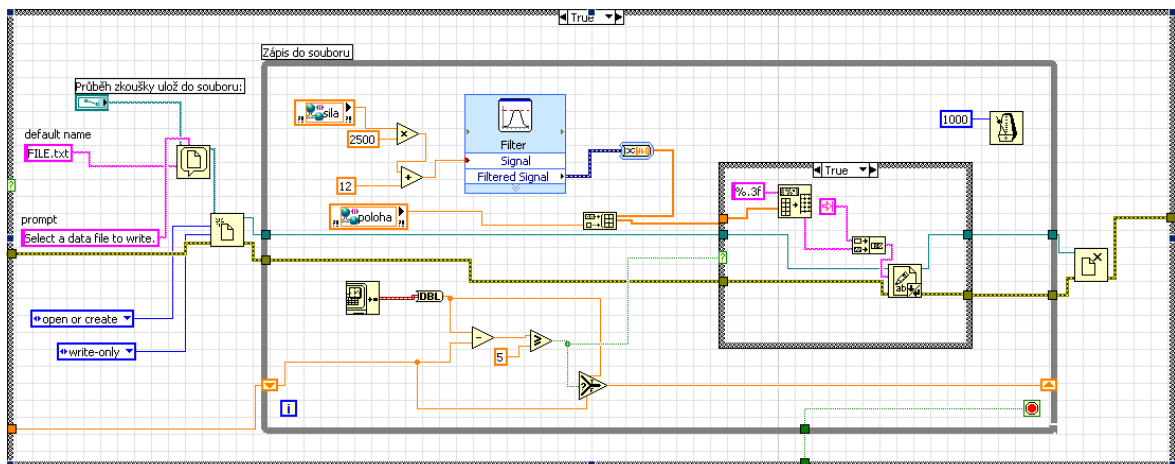
Obr. 65 Zobrazení grafů

### 5.1.2.4 Zápís do souboru

Poslední ze čtyř částí HOSTControl.vi je zápis do souboru. Celý zápis do souboru je uzavřen do struktury Case, která má dva stavy (Obr. 66). O nich rozhoduje stav checkboxu „ulož“.

První stav je logická 0, kde se nic neděje, pouze datový tok proběhne na konec této struktury a vrací se zpět na začátek všech čtyř částí HOSTCode.vi.

Druhý stav je logická 1. Zde se nejprve inicializuje soubor. Vznikne jeho jméno, popř. umístění na disku, typ a zda se bude zapisovat nebo číst. Dále tok dat postupuje do smyčky, která je časovaná 10ms. Tady se čte poloha a síla z Real-Time PC a karty FPGA a prochází přes funkci, která vytvoří 2D pole hodnot. Opět se na základě rozdílu systémového času předchozí a současné smyčky povoluje zápis těchto hodnot do pole, které se mění na textový řetězec s formátováním a přidáním mezi hodnotu síly a polohy tabulátor. Tento textový řetězec se pak ukládá na jednotlivý řádek do textového souboru. Po opuštění smyčky se tok dat dostane do funkce uzavření souboru a končí opuštěním struktury Case pro povolování zápisu dat do souboru.



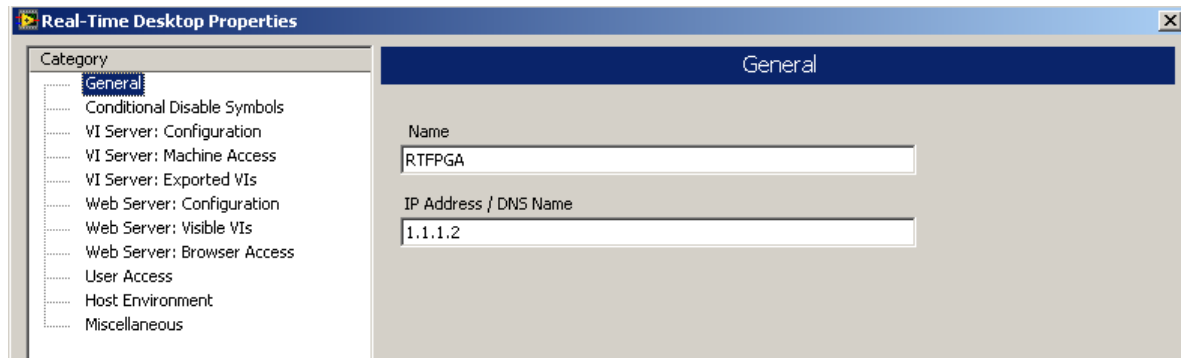
Obr. 66 Zápis do souboru

## 5.2 Real-TimePC

Real-Time PC je v projektu (Obr. 53) nazváno RTFPGA z toho důvodu, že obsahuje VI pro komunikaci mezi Host PC, Real-Time PC a FPGA kartou, jedná se o RTCode.vi. Také

je zde poslední VI a to FPGACode.vi, které se stará o čtení a zápis analogových vstupů-výstupů a digitálních vstupů-výstupů samotné FPGA karty.

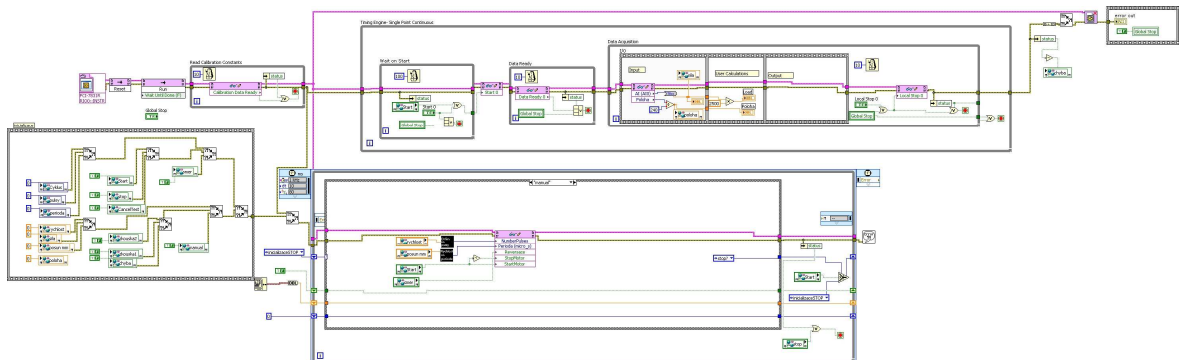
Zde se jako u Host PC provádí nastavení. Nejdůležitější je opět IP adresa (Obr. 67).



Obr. 67 Nastavení Real-Time PC

### 5.2.1 RTCode.vi

Ve VI RTCode.vi (Obr. 68) je celá logika programu a skládá se ze dvou větších celků. První se stará o oživení a komunikaci s FPGA kartou a ve druhém je samotný stavový automat prováděných zkoušek a manuálního režimu.

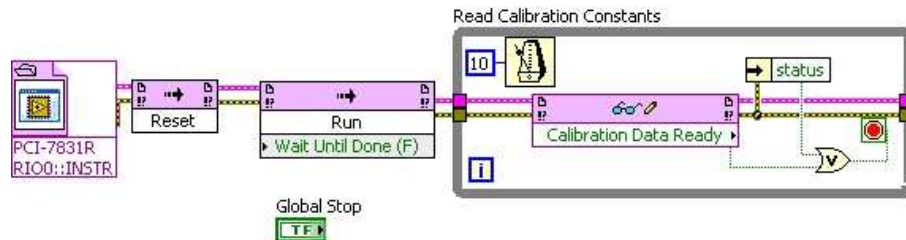


Obr. 68 RTCode.vi

#### 5.2.1.1 Komunikace s FPGA kartou (PCI-7831R)

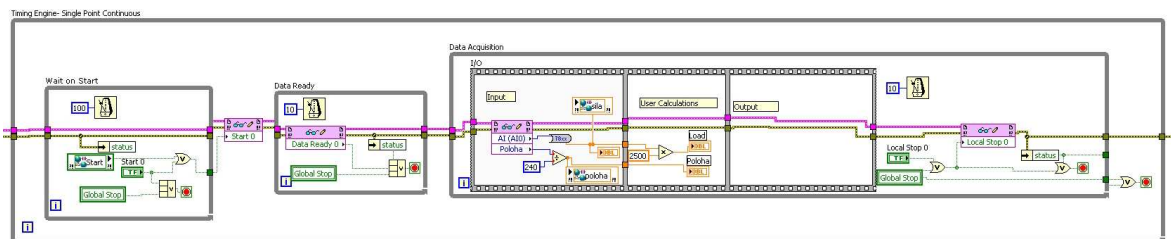
Nejprve se musí nastavit všechny sdílené síťové proměnné na nulové hodnoty. Toto je zabezpečeno v sekvenci o jednom rámci nazvané inicializace. Aby mohl HOSTCode.vi komunikovat s FPGACode.vi musí se v první části (vlevo nahoře Obr. 68) jako první detekovat FPGA karta a její typ pod názvem, v tomto případě se jedná o typ karty PCI-

7831R série R. Poté se FPGA karta resetuje a spustí (Obr. 69). Pak se nechá karta načíst všechny potřebné konstanty a proměnné potřebné ke komunikaci. Dokud nebude připravena, nepustí datový tok dál.



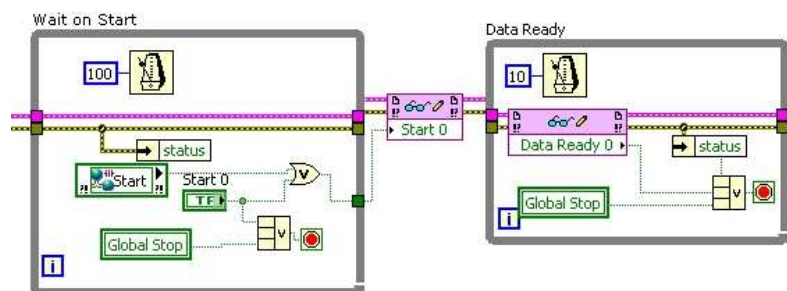
Obr. 69 Detekce karty a nastavení

Následuje smyčka Timing Engine - Single Point Continuous, která má za úkol jednotlivě zapisovat nebo číst hodnoty z FPGA karty za pomoci FPGACode.vi. Tato smyčka (Obr. 70) je standardně vygenerovaná za pomoci nástroje pro komunikaci s FPGA kartami různých typů.



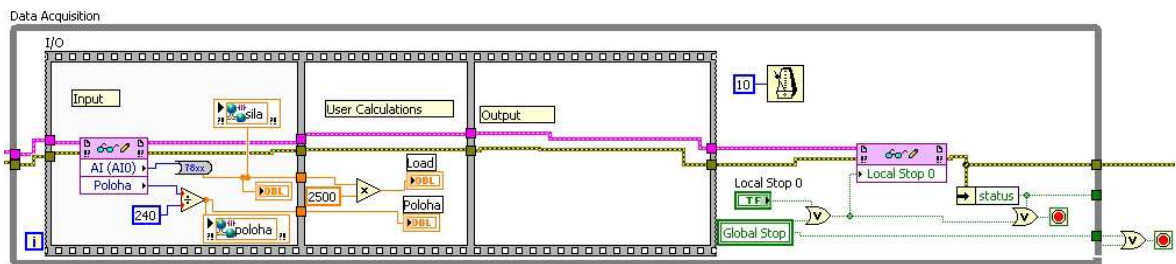
Obr. 70 Smyčka Timing Engine - Single Point Continuous

Uvnitř (Obr. 71) se kontroluje status chyby, stlačení tlačítka spuštění zkoušky nebo manuálního posunu a připravenost karty pro čtení a zápis. Samozřejmě se neustále testuje STOP tlačítko.



Obr. 71 Čekání na start a povolení číst data

A konečně poslední nejdůležitější část pro získávání dat z FPGA karty (Obr. 72). Zde se čte poloha, jak jsem už uváděl výše, nechte se ze žádných vstupů, ale ze Sub VI, které je samostatné a je vloženo do FPGACode.vi. Také se zde čte hodnota síly z analogového vstupu AI0 FPGA karty. Hodnota síly v newtonech se získá převodem na potřebný formát číselné hodnoty a vynásobením číselnou konstantou.

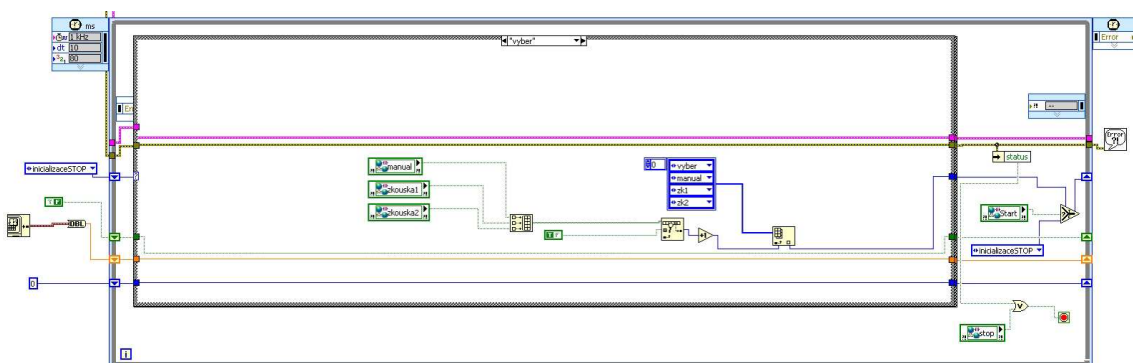


Obr. 72 Získávání dat

Pro výstupy, zápisy na kartu FPGA slouží další rámec sekvenční struktury, ale v tomto případě je zvolen stavový automat, který popíšu na následujících stránkách.

### 5.2.1.2 Stavový automat

Pro programování v NI LabVIEW je typické, že pokud nejsou objekty spojeny drátky, pak se posloupnost prováděných operací odvíjí z levého horního rohu VI do pravého dolního. Takže pak není zcela jasné, která funkce se provede dříve a která později. V tomto případě se používá smyčka Timed loop, tedy časová smyčka, ve které se nastaví počet opakování do taktu hodin a priorita. Stavový automat má prioritu nižší, nežli Smyčka Timing Engine - Single Point Continuous.

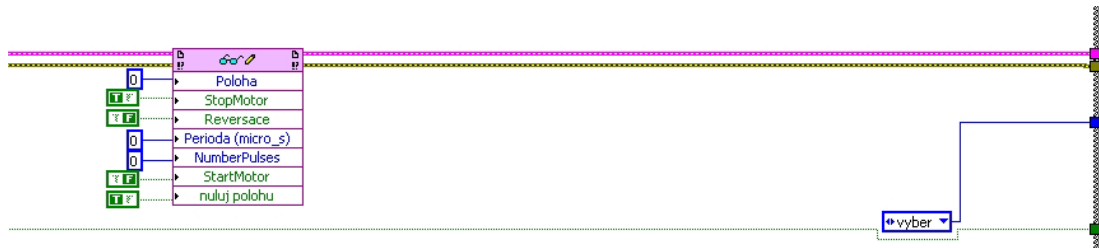


Obr. 73 Stavový automat

Stavový automat funguje na principu neustálého překlápění struktury case řídicím signálem, který je vždy na konci prováděného rámce poslán posuvným registrem na

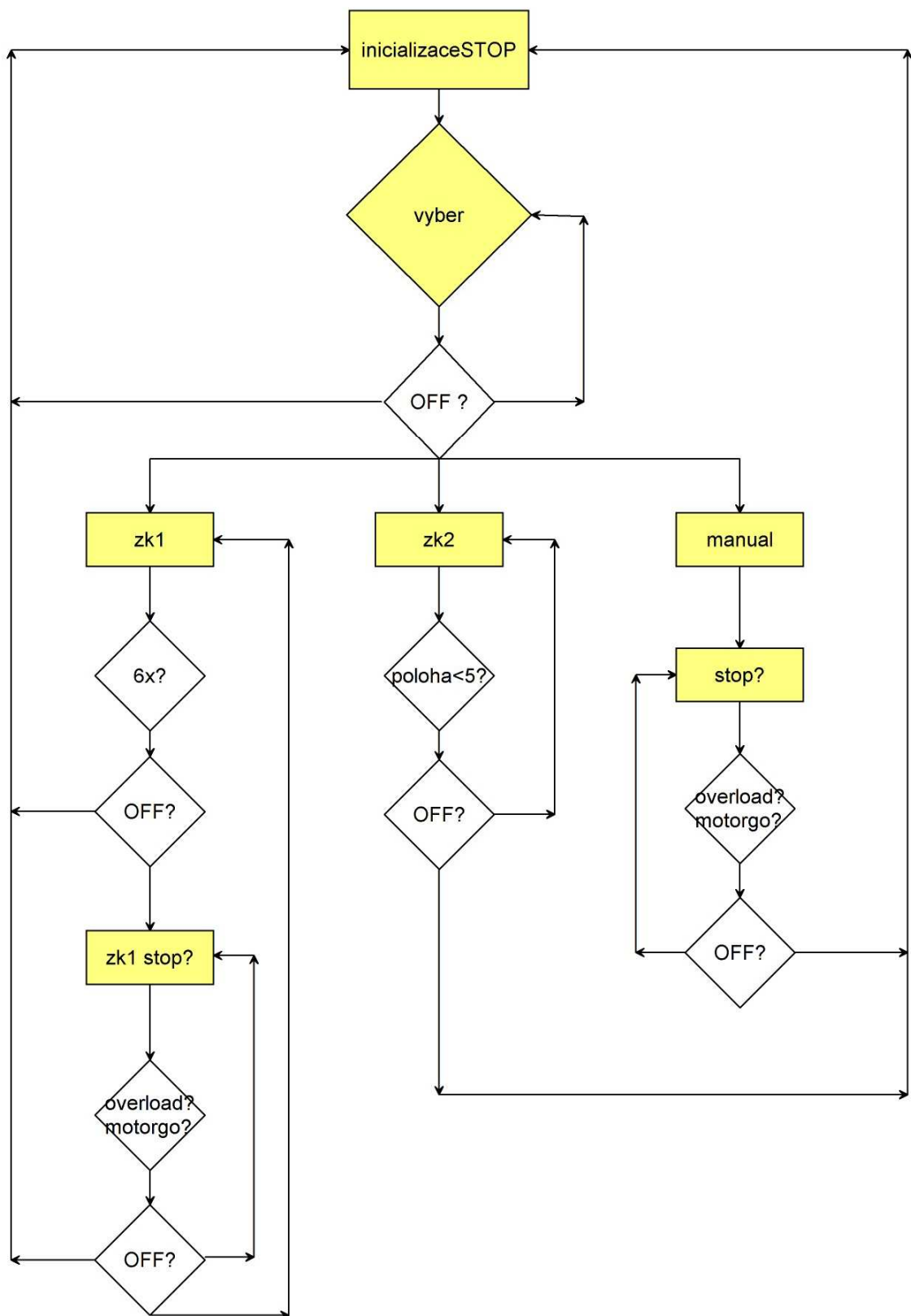
začátek do vstupního výběrového terminálu struktury case. Samozřejmě se musí někde začít a tím je první řídicí signál výčtového typu „inicializaceSTOP“.

Obsluha rámce „inicializace stop“ je nastavení základních proměnných používaných v FPGACode.vi (Obr. 74). Na konci rámce vracím zpět na začátek příkaz „vyber“ (Obr. 76).



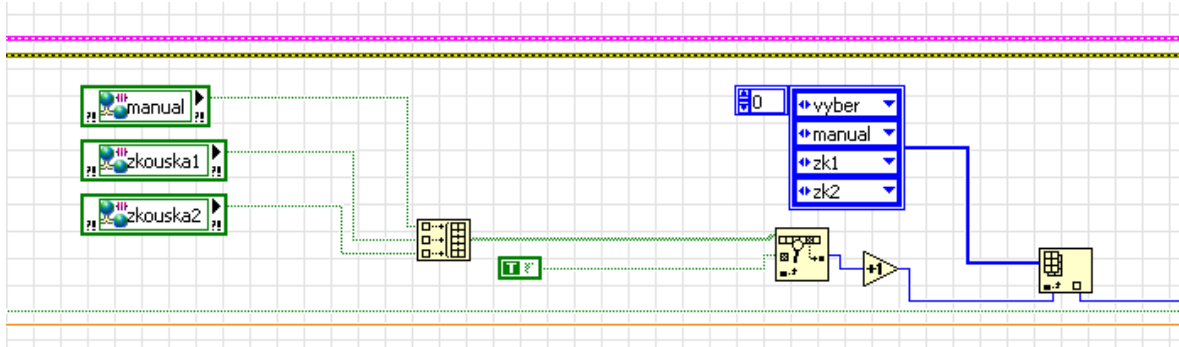
Obr. 74 inicializaceSTOP

Pro upřesnění a lepší názornost je na dalším obrázku vytvořen vývojový diagram (Obr. 75) celého stavového automatu. Žlutě jsou zvýrazněny řídicí signály výčtového typu. Podmínky „OFF?“ znamenají kontrolu přepnutého spínače spuštění zkoušek nebo manuálního posunu.



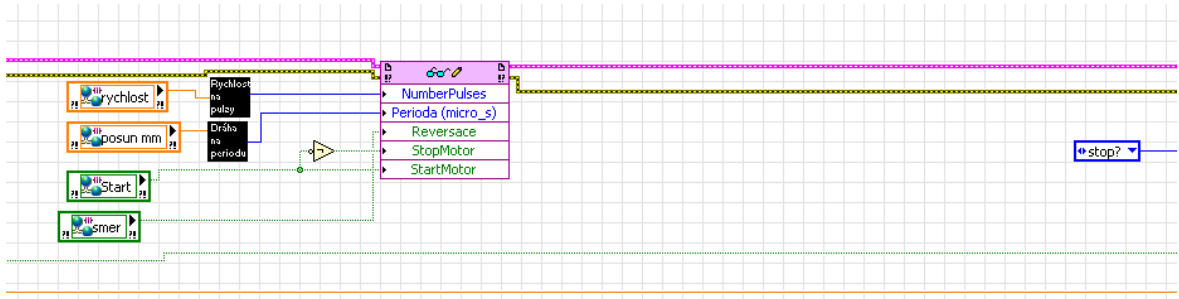
Obr. 75 Vývojový diagram stavového automatu

V rámci „vyber“ (Obr. 76) se rozhoduje která ze zkoušek nebo manuálního posunu se vykoná. Logika výběru je taková, že se všechny tři stavy vloží do pole, převede se na 1D pole a poté se vybere z pole výčtových typů příslušný řídicí signál.



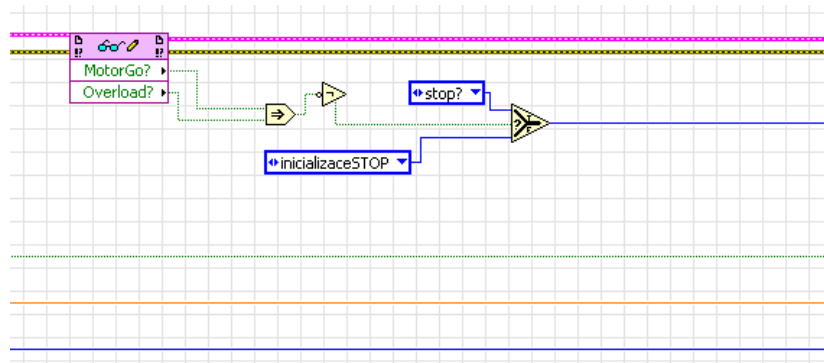
Obr. 76 Rámec „vyber“

Vybere-li se manuální posun (Obr. 77), přečtou se sdílené síťové proměnné rychlosti, posunu, směru a přepínače spuštění z HOSTCode.vi a zapíšu se do proměnných v FPGACode.vi. Po přečtení rychlosti a posunu se ještě vyvolá podprogram (Sub VI), který převede rychlost na periodu a délku posunu na počet pulzů. Je to třeba pro odeslání správných hodnot z FPGA karty řídicí jednotce krokového motoru.



Obr. 77 Rámec „manual“

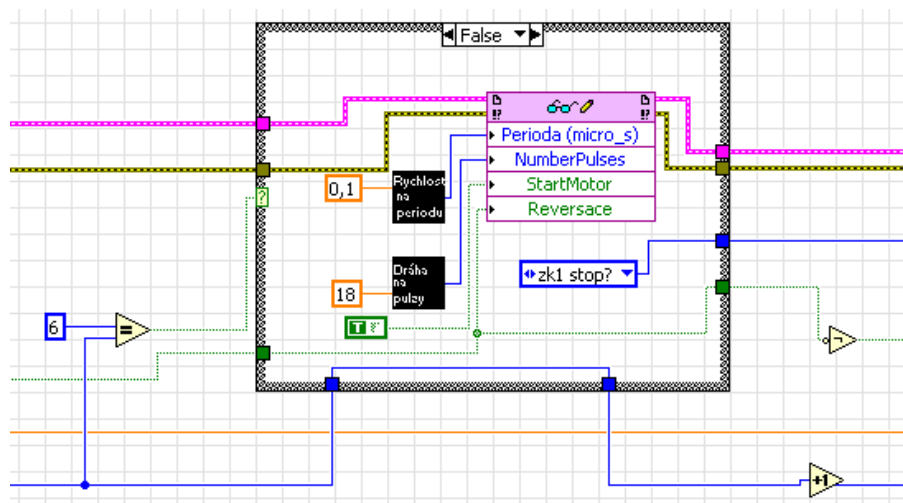
Poté se spouští jednoduchý rámec „stop?“ (Obr. 78). Tento rámec sleduje stav motoru a hodnotu maximální přípustné síly. Posílá zpět do řídicího signálu sám sebe, dokud není motor zastaven nebo síla větší, než přípustná.



Obr. 78 Rámec „stop?“

Na konci manuálního posunu se opět provádí rámec „inicializaceSTOP“.

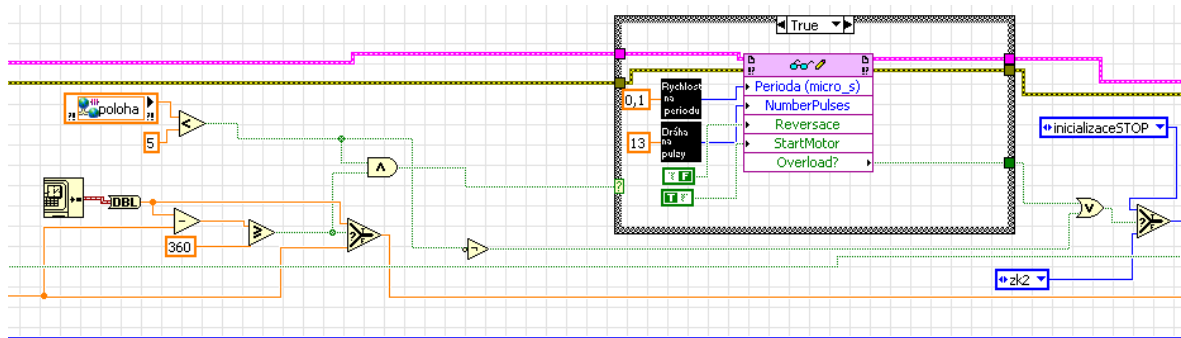
První zkouška má svůj rámec „zk1“ (Obr. 79). Logika je následující. Směr motoru je nastaven na TLAK a do periody impulsů a jejich počtu se v FPGACode.vi zapíšou konstantní hodnoty rychlosti a posunu. Motor se rozběhne a přechází se do rámce „zk1 stop?“, který má za úkol hlídat chod motoru a hodnotu maximální přípustné síly. V případě, že se motor zastaví, vrací se zpět do rámce „zk1“, změní se směr chodu motoru na TAH a opět se nastaví konstanty pro rychlost a posun a znovu do rámce „zk1 stop?“. To vše se opakuje 3 krát, poté se vrací řídicí signál „inicializaceSTOP“.



Obr. 79 Rámec „zk1“

Posledním z rámců struktury case stavového automatu je „zk2“ (Obr. 80) pro vykonání druhé zkoušky. Na začátku se čte sdílená síťová proměnná „poloha“ a zjišťuje se, zda se neztvětšila víc než o 5 mm. Současně se zjišťuje stav rozdílu systémového času před začátkem zkoušky a v současnosti. Nesmí být větší než 6 minut. Dokud jsou splněny tyto podmínky, posílají se do proměnných v FPGACode.vi konstanty pro směr, posun a rychlost

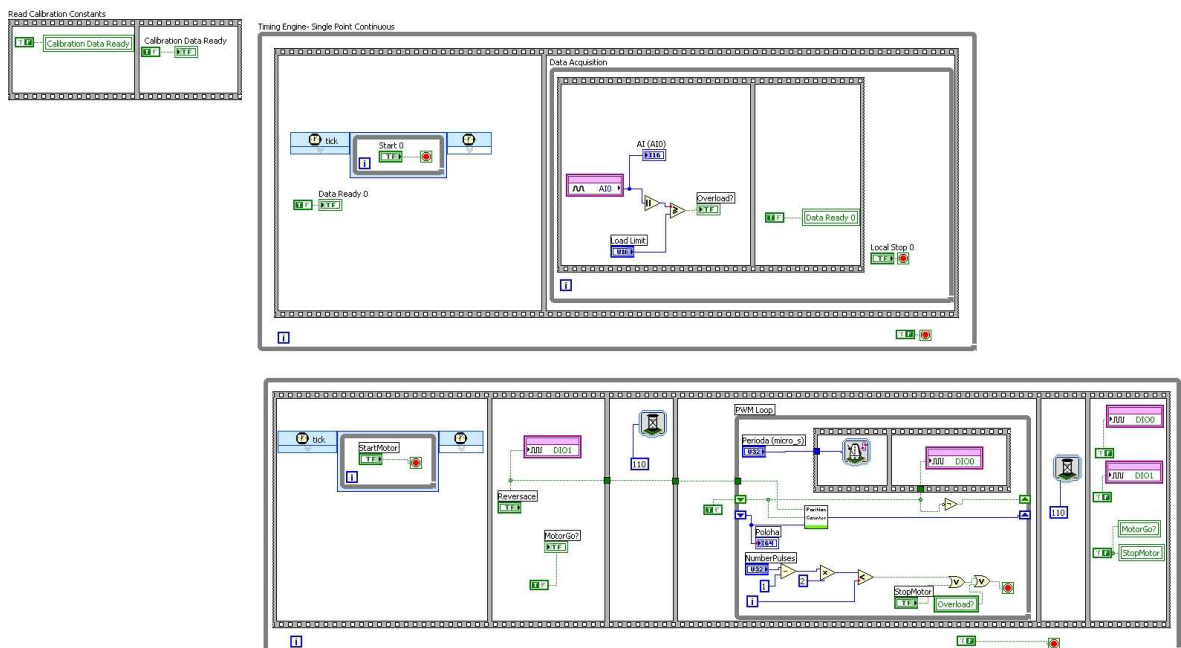
chodu motoru. V opačném případě se vrací stavový automat do stavu „inicializaceSTOP“ a následně do „vyber“.



Obr. 80 Rámec „zk2“

### 5.2.2 FPGACode.vi

Na kartě FPGA jsou v projektu (Obr. 53) uvolněny dva digitální vstupy-výstupy (DIO0 a DIO1) a jeden analogový vstup (AI0). Pomocí VI FPGACode.vi se posílají relevantní data na výstupy a přijímají se hodnoty a logické stavy ze vstupů FPGA karty.

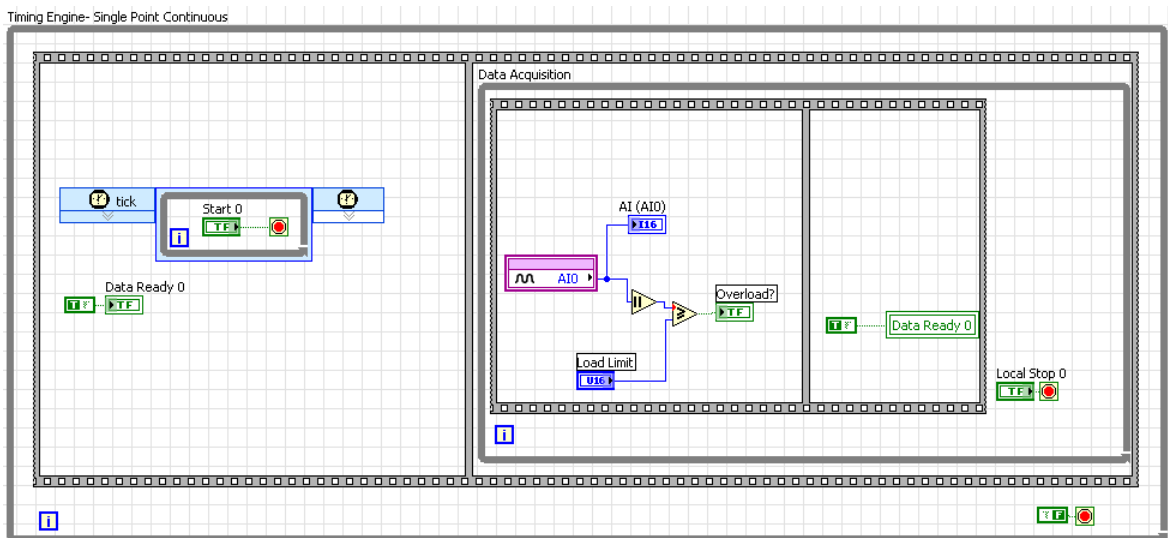


Obr. 81 FPGACode.vi

VI (Obr. 81) se dělí na dvě smyčky, první se stará o čtení z analogového vstupu AI0 a ta druhá o zápis do digitálních výstupů DIO0 a DIO1.

### 5.2.2.1 Smyčka čtení AIO

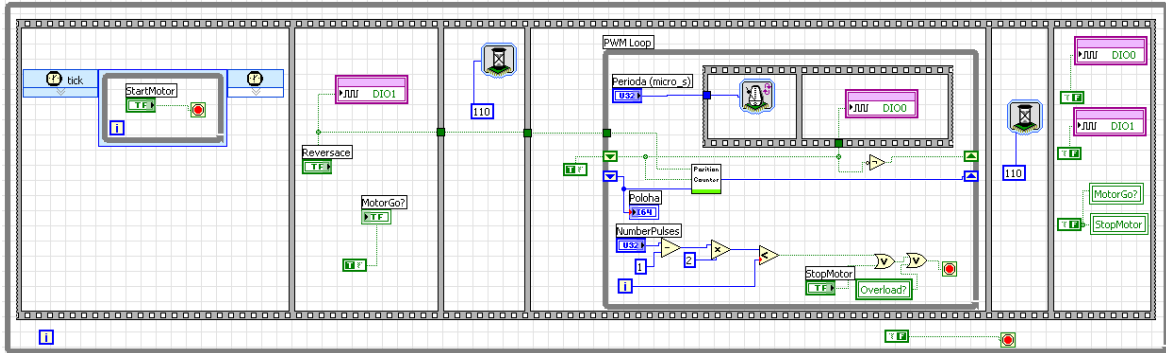
V prvním rámci sekvenční struktury se čeká na spuštění získávání dat z analogového vstupu. Spuštění je povoleno v RTCCode.vi, kde se do proměnné „Start 0“ posílá logická 1. Ve druhém rámci je smyčka, ve které běží logika získávání dat, dokud nebude opuštěna vypnutím běhu programu. Do proměnné „AI(AIO)“ se přečte hodnota z analogového vstupu AIO a její absolutní hodnota se porovnává s maximální přípustnou silou, která je nastavena v proměnné „Load Limit“.



Obr. 82 Smyčka čtení z AIO

### 5.2.2.2 Smyčka zápisu na DIO0 a DIO1

Začíná se čekáním na povolení chodu motoru (Obr. 83), ve druhém rámci se posílá logická hodnota do DIO0 pro ovladač krokového motoru, aby věděl, kterým směrem se bude motor otáčet. V dalším rámci se čeká 100ms, než se informace odešle. Pak nastává posílání logických stavů do DIO1 pro rozběh krokového motoru. V této smyčce se posílá počet pulzů hlídanych Sub VI PositionCounter.vi, které také sleduje stav změny chodu motoru. To vše při časování nastaveném dobou periody. Pak se opět čeká 100ms pro přenos informací. Tato doba je doporučená [6]. Nakonec se tzv. vynulují potřebné proměnné a DIO0 s DIO1. Všechny hodnoty proměnných použité v tomto VI se přebírají z VI RTCCode.vi.



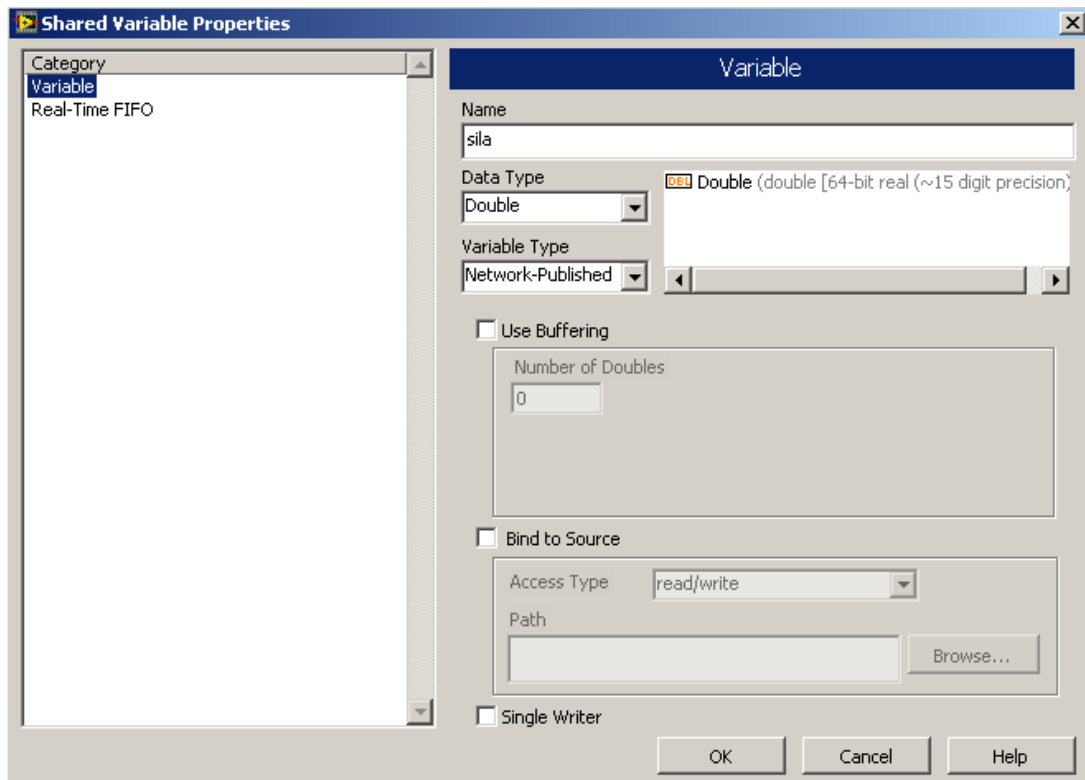
Obr. 83 Smyčka zápisu na DIO0 a DIO1

### 5.3 Sub VI - podprogramy

Pro lepší přehlednost v programování jsou zde použity také Sub VI (Obr. 53). Těchto podprogramů nebo funkcí je zde celkem 3. Jsou to převody hodnot z rychlostí v mm/min na dobu v ms periody posílaných pulzů, převod hodnot posunu v mm na počet pulzů a čítač polohy, který posílá polohu v mm na výstup, když na vstup přivádíme stávající polohu, hodnotu pulzu a směr pohybu motoru. Mohlo by jich být víc, ale zatím je přehlednost v samostatných VI dobrá.

### 5.4 Sdílené proměnné

Nevysvětleny ještě zůstávají sdílené síťové proměnné, které zde byly použity. Jak ukazuje okno projektu (Obr. 53), je jich celkem patnáct a byly použity pro přenos hodnot mezi VI Host PC a VI Real-Time PC. Po přidání nové proměnné se zadá její jméno, datový typ a zda chceme aby byla síťová (Obr. 84). Pak už stačí zadat požadavek na FIFO a uložit. S

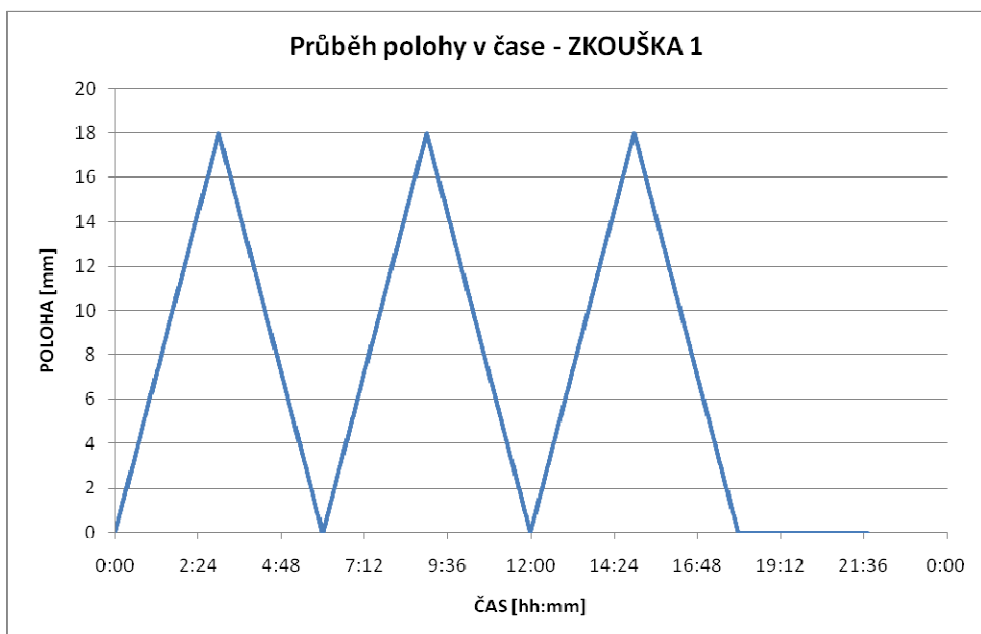
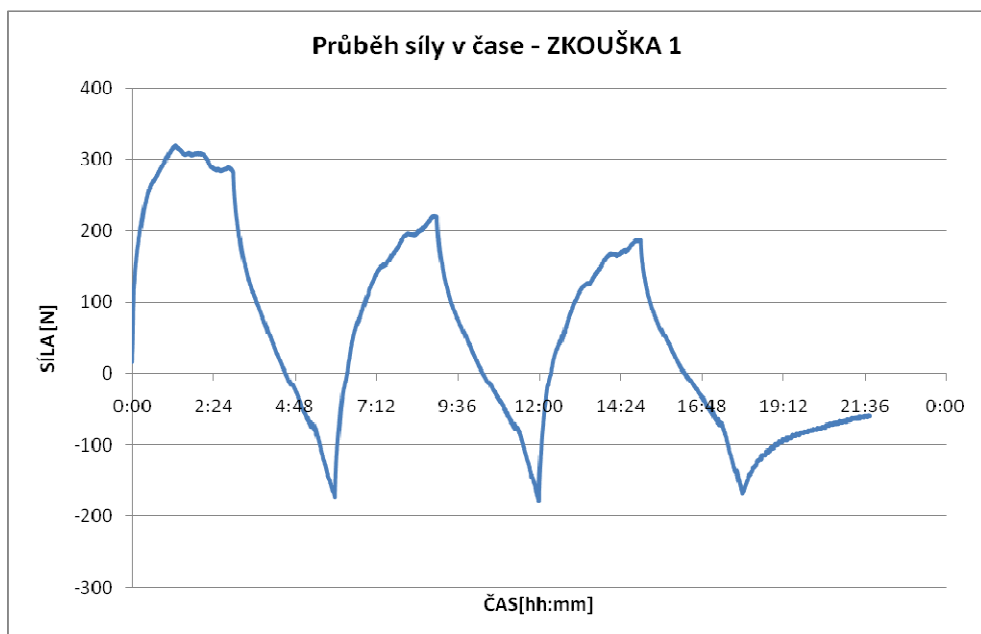


Obr. 84 Nastavení sdílených proměnných

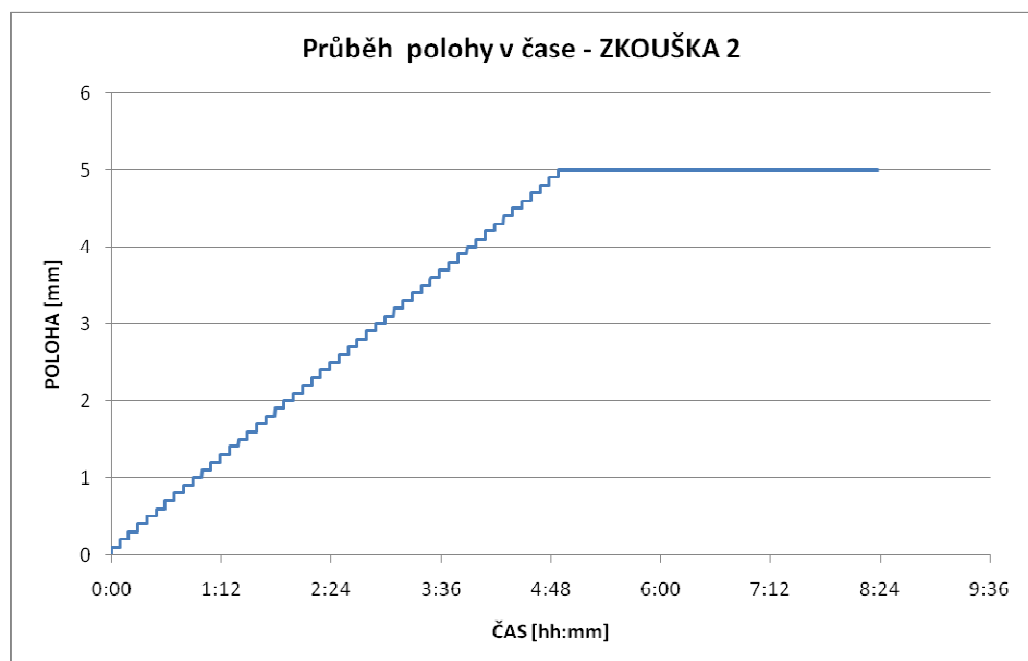
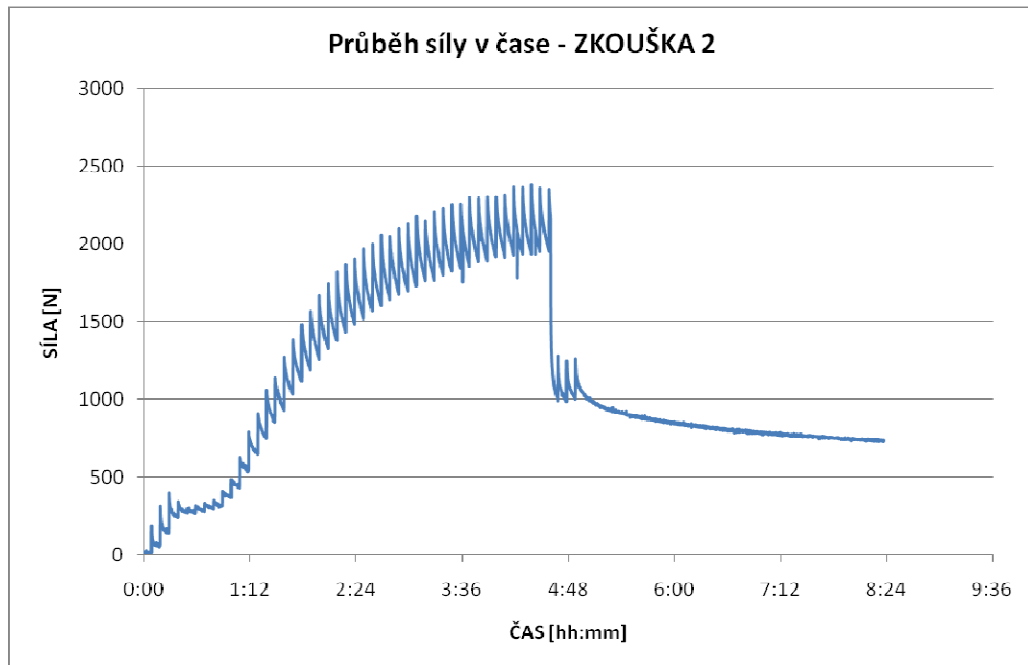
## 6 VÝSLEDKY Z NAMĚŘENÝCH ZKOUŠEK

Obě zkoušky proběhly několikrát a z jedné z nich bych rád zde vložil grafické výstupy. Tyto jsem získal z textových souborů, do kterých se mi ukládaly průběhy zkoušek. Veškeré údaje jsem převedl do tabulkového editoru, ze kterého jsou dole uvedené grafy. Grafy jsou samy o sobě vypovídající, takže bez komentáře.

### 6.1 Grafy zkoušky 1



## 6.2 Grafy zkoušky 2



## ZÁVĚR

Téma práce řízení zkušebního zařízení jsem si vybral pro lákavou představu poznání laboratorních zkoušek na lepených materiálech. V minulosti jsem vypracoval bakalářskou práci na téma lepených spojů, kde jsem se z větší části věnoval teorii lepení.

Současně mě lákalo naučit se něco nového, tím bylo programování v NI LabVIEW. Byla to pro mě úplně nová zkušenost, nikdy před tím jsem se s takovým programováním nesešel. Musel jsem začít od nejjednodušších aplikací. Trvalo dlouho, než jsem se prokousal mnoha anglicky psanými texty. Bezesporu Internet je fenomén, který mi pomohl ze všeho nejvíc, samozřejmě www stránky společnosti NI jsou napěchované příklady a také videosekvencemi jak se programuje v LabVIEW. Bohužel není moc knih v českém jazyce a pro programování Real-Time aplikací v LabVIEW žádná.

Je to opravdu úžasný nadčasový nástroj a jsem rád, že jsem tomu v určitých oblastech porozuměl. Nejvíce je to vidět na výsledné aplikaci řízení zkoušek pro asfaltové zálivky. Aplikace pro zkoušky pracuje výborně. Bylo nutné se podřídít českou technickou normou, která vyžadovala přesné nastavení vstupních hodnot aplikace. Výsledky zkoušek nejlépe zobrazují grafy průběhů polohy a síly v kapitole 6.

Programování v NI LabVIEW bych se chtěl věnovat i nadále. Je možné zakoupit moduly k PC s propojením přes USB, které mají několik analogových i digitálních vstupů a výstupů. Nicméně je třeba říct, že veškeré zařízení z dílny NI (SW i HW), které se v laboratoři používá je finančně velmi náročné.

## ZÁVĚR V ANGLIČTINĚ

My thesis focuses on the control of testing equipment as I take particular interest in the topic of laboratory testing of glued materials. My bachelor thesis was mainly aimed at providing the theoretical background regarding gluing of glued joints.

At the same time, I was interested in learning new skills in LabVIEW programming. This was an entirely new experience as I had never the opportunity to use such a programming tool. I had to start with the least complex applications and read through numerous texts in English. Without doubt, the Internet is a phenomenon that helped me most of all, the NI web pages provide many examples and video sequences describing LabVIEW programming. Unfortunately, not many books are available in the Czech language, and none of them focuses on the programming of Real-Time applications in LabVIEW.

LabVIEW is a brilliant state-of-the-art tool and I am glad I have learnt how to use it in certain areas. It is namely apparent in the resulting application of testing control for asphalt fillers. As regards these tests, the application is highly efficient. It was necessary to conform to a Czech technical standard that required precise setting of input application values. The test results are best illustrated in the graphs showing the position and force curves provided in Chapter 6.

I would like to further improve my skills in LabVIEW programming in the future. Modules that can be connected to a PC via USB slots and that feature a number of analogue and digital inputs and outputs are also available for purchase. However, it must be noted that all the equipment manufactured by NI (both SW and HW) to be used in the laboratory is very costly.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Vlach, J., Havlíček, J., Vlach, M.: Začínáme s LabVIEW, BEN – technická literatura, Praha 2008
- [2] Balátě, J.: Automatické řízení, 2. Aktualizované vydání, BEN – technická literatura, Praha 2004
- [3] LabVIEW : FPGA Interface User Guide, National Instruments Corporation, Austin 2004
- [4] LabVIEW : FPGA Module User Manual, National Instruments Corporation, Austin 2004
- [5] LabVIEW : Real-Time Module User Manual, National Instruments Corporation, Austin 2004
- [6] National Instruments [online]. [cit. 2010-05-15]. Dostupný z WWW: <<http://www.ni.com>>
- [7] Úřad pro technickou normalizaci, metrologii a státní zkušebnictví [online]. [cit. 2010-05-10]. Dostupný z WWW: <<http://www.unmz.cz>>
- [8] Firma ATAS elektromotory Náchod a. s. [online]. [cit. 2010-04-25]. Dostupný z WWW: <<http://www.atas.cz>> <[trojovsky@atas.cz](mailto:trojovsky@atas.cz)>
- [9] Firma ENIKA.CZ spol. s r. o. [online]. [cit. 2010-03-04]. Dostupný z WWW: <<http://www.enika.cz>> <[r.smidrkal@enika.cz](mailto:r.smidrkal@enika.cz)>
- [10] Firma NWK TECHNOLOGY spol. s r. o. [online]. [cit. 2010-05-05]. Dostupný z WWW: <<http://www.nwk-tech.cz>>
- [11] Firma OMEGA ENGINEERING INC. [online]. [cit. 2010-05-10]. Dostupný z WWW: <<http://www.omega.com>>
- [12] Firma HW server spol. s r. o. [online]. [cit. 2010-05-20]. Dostupný z WWW: <<http://www.hw.cz>>

## SEZNAM POUŽITÝCH SYMBOLŮ, ZKRATEK A ODBORNÝCH VÝRAZŮ

FPGA	Field programmable gate arrays, programovatelné hradlové pole.
HW	Hardware, technické vybavení.
LabVIEW	Programovací prostředí grafického jazyka.
PC	Personal computer, osobní počítač.
Real Time	Reálný čas.
ÚNMZ	Úřad pro technickou normalizaci, metrologii a státní zkušebnictví.
NI	National Instruments
LED	Light-emiting diode, světlo vyzařující dioda.
DIN	Německá národní norma.
$V_{DC\ nom}$	Nominální hodnota napájecího napětí.
$I_{NP\ min}$	Minimální hodnota nominálního proudu fáze.
$I_{NP\ max}$	Maximální hodnota nominálního proudu fáze.
DIP SWITCH	Sada manuálních elektropřepínačů.
DC	Stejnoseměrný proud.
AM3, AM2, AM1	Šroubové svorkovnice.
DAQ	Data acquisition, získávání dat.
PCI	Typ počítačové sběrnice.
Host	Síťově připojený hostitelský PC.
Ethernet	Síťová technologie.
Loop	Smyčka.
VI	Virtual Instrument, virtuální přístroj.
ISA	Typ počítačové sběrnice.
PXI	Modulární platforma, datová sběrnice.
MAX	Nástroj pro konfiguraci zařízení DAQ.

**SEZNAM OBRÁZKŮ**

Obr. 1 Zkušební sestava.....	12
Obr. 2 Obecná funkce krokového motoru s aktivním rotorem.....	13
Obr. 3 Technický náčrt motoru Z42VV141 .....	14
Obr. 4 Zapojení krokového motoru se šroubovicí .....	15
Obr. 5 Charakteristika statického momentu.....	15
Obr. 6 Provozní charakteristika .....	16
Obr. 7 Rozměry a rozmístění součástek (svorkovnice, LED diody, přepínače a propojky) na desce plošných spojů.....	17
Obr. 8 Schéma připojení ovladače .....	21
Obr. 9 Regulovatelný zdroj MDR-60 .....	22
Obr. 10 Ovladač krokového motoru v zapojení s regulovatelným zdrojem.....	23
Obr. 11 Typické fóliové nalepovací tenzometry.....	23
Obr. 12 Snímač síly připevněný mezi krokovým motorem a měřeným vzorkem .....	24
Obr. 13 Zapojení tenzometru a krokového motoru do redukce .....	25
Obr. 14 Propojovací redukce SCB-68 .....	25
Obr. 15 Schéma použití konektoru 68-PIN I/O .....	26
Obr. 16 Číslování pinů konektoru SCB-68.....	26
Obr. 17 Kabel SHC68-68 RMIO .....	27
Obr. 18 Konektor pro zapojení do propojovací redukce SCB-68.....	27
Obr. 19 Konektor pro zapojení do FPGA karty PCI-7831R.....	27
Obr. 20 FPGA karta NI PCI-7831R.....	28
Obr. 21 Struktura FPGA .....	29
Obr. 22 Čip FPGA .....	30
Obr. 23 Rozdíl mezi Hard a Soft Real-Time aplikacemi.....	33
Obr. 24 Real-Time PC – FPGA – senzory.....	33
Obr. 25 Komunikace mezi Host PC a Real-Time PC.....	34
Obr. 26 RT Communication Wizard.....	35
Obr. 27 Vzorek asfaltové zálivky .....	35
Obr. 28 LabVIEW 1.0 a počítač Macintosh .....	37
Obr. 29 Časová osa vývoje LabVIEW .....	38
Obr. 30 Úvodní obrazovka Getting Started .....	39

Obr. 31	Obrazovka hledání příkladů NI Example Finder .....	39
Obr. 32	Čelní panel .....	40
Obr. 34	Otevřená knihovna Controls .....	41
Obr. 33	Nástrojová lišta čelního panelu.....	41
Obr. 35	Blokový diagram.....	42
Obr. 36	Nástrojová lišta blokového diagramu .....	42
Obr. 37	Rozdělení konektoru na počet terminálů .....	43
Obr. 38	Paleta nástrojů.....	44
Obr. 39	Paleta Controls.....	45
Obr. 40	Paleta Functions.....	46
Obr. 41	Context Help.....	47
Obr. 42	Nemožný spoj .....	47
Obr. 43	Datové typy .....	48
Obr. 44	Cyklus While .....	49
Obr. 45	Cyklus For .....	49
Obr. 46	Cyklus For s posuvným registrem .....	50
Obr. 47	Struktura Case.....	50
Obr. 48	Sequence .....	51
Obr. 49	Paleta Graph.....	51
Obr. 50	Typy zobrazovačů.....	52
Obr. 51	DAQ Assist.....	53
Obr. 52	Uživatelské rozhraní .....	55
Obr. 53	Výpis částí projektu .....	58
Obr. 54	Nastavení My Computer.....	59
Obr. 55	Blokový diagram.....	59
Obr. 56	Základní nastavení .....	60
Obr. 57	Tlačítka .....	60
Obr. 58	Událost Timeout .....	61
Obr. 59	Událost tlačítka „zk1“ .....	61
Obr. 60	Událost tlačítka „zk2“ .....	62
Obr. 61	Událost manuálního režimu .....	62
Obr. 62	Událost tlačítka „rychlost“ .....	63
Obr. 63	Událost tlačítka „smaž grafy“ .....	63

Obr. 64 Událost tlačítka „start“ .....	64
Obr. 65 Zobrazení grafů.....	64
Obr. 66 Zápis do souboru.....	65
Obr. 67 Nastavení Real-Time PC .....	66
Obr. 68 RTCCode.vi.....	66
Obr. 69 Detekce karty a nastavení .....	67
Obr. 70 Smyčka Timing Engine - Single Point Continuous .....	67
Obr. 71 Čekání na start a povolení číst data .....	67
Obr. 72 Získávání dat.....	68
Obr. 73 Stavový automat .....	68
Obr. 74 inicializaceSTOP .....	69
Obr. 75 Vývojový diagram stavového automatu .....	70
Obr. 76 Rámec „vyber“ .....	71
Obr. 77 Rámec „manual“ .....	71
Obr. 78 Rámec „stop?“ .....	72
Obr. 79 Rámec „zk1“ .....	72
Obr. 80 Rámec „zk2“ .....	73
Obr. 81 FPGACode.vi .....	73
Obr. 82 Smyčka čtení z AI0.....	74
Obr. 83 Smyčka zápisu na DIO0 a DIO1 .....	75
Obr. 84 Nastavení sdílených proměnných .....	76

**SEZNAM TABULEK**

Tab. 1 Technické údaje .....	14
Tab. 2 Základní vlastnosti.....	16
Tab. 3 Zapojení vstupních a výstupních logických signálů, svorkovnice AM3 .....	18
Tab. 4 Zapojení výkonných vstupů a výstupů.....	19
Tab. 5 Možnosti nastavení přepínačů a propojek .....	20
Tab. 6 LED indikace stavu.....	21

## SEZNAM PŘÍLOH

- P I CD-ROM s vytvořeným řídicím a vizualizačním programem ve formě pro spuštění v LabVIEW (celý projekt).

**PŘÍLOHA P I: CD-ROM S VYTVOŘENÝM ŘÍDÍCÍM A VIZUALIZAČNÍM PROGRAMEM VE FORMĚ PRO SPUŠTĚNÍ V LABVIEW (CELÝ PROJEKT).**

**Struktura disku:**

Složka **PROGRAM**: kompletní řídicí a vizualizační program ve formě projektu spustitelného v LabVIEW 8.2.

Složka **DP**: obsahuje textovou část práce ve formátu PDF.