

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA TECHNOLOGIÍ A MĚŘENÍ**

# **DIPLOMOVÁ PRÁCE**

**Kalibrace wattmetru LMG500**

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2018/2019

**ZADÁNÍ DIPLOMOVÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan STANĚK**  
Osobní číslo: **E17N0029P**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Komerční elektrotechnika**  
Název tématu: **Kalibrace wattmetru LMG500**  
Zadávající katedra: **Katedra technologií a měření**

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

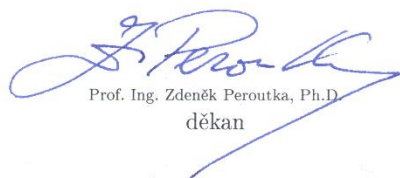
1. Seznamte se s obsluhou wattmetru LMG500 a kalibrátoru Fluke 5500A.
2. Připravte řídicí program v Labview pro kalibraci wattmetru.
3. Proveďte kontrolní kalibraci.

Rozsah grafických prací: podle doporučení vedoucího  
Rozsah kvalifikační práce: 40 - 60 stran  
Forma zpracování diplomové práce: tištěná/elektronická  
Seznam odborné literatury:

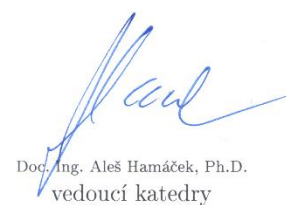
**1. Začínáme s LabVIEW**

Vedoucí diplomové práce: Ing. Aleš Voborník, Ph.D.  
Katedra technologií a měření

Datum zadání diplomové práce: 5. října 2018  
Termín odevzdání diplomové práce: 30. května 2019



Prof. Ing. Zdeněk Peroutka, Ph.D.  
děkan



Doc. Ing. Aleš Hamáček, Ph.D.  
vedoucí katedry

V Plzni dne 5. října 2018

## **Abstrakt**

Předkládaná diplomová práce je zaměřena na řešení návrhu a realizaci programu pro kalibraci wattmetru LMG500 v automatizovaném režimu pomocí programovacího jazyka LabVIEW.

## **Klíčová slova**

LabVIEW, kalibrace, wattmetr, LMG500, Fluke 5500A

## **Abstract**

The diploma thesis is focused on designing and realization of a program for calibration of wattmeter LMG500 in automated mode using programming language LabVIEW.

## **Key words**

LabVIEW, calibration, wattmeter, LMG500, Fluke 5500A

## Prohlášení

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

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

.....

podpis

V Plzni dne 24.5.2019

Bc. Jan Staněk

## **Poděkování**

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

# Obsah

<b>OBSAH</b> .....	<b>8</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>9</b>
<b>ÚVOD</b> .....	<b>10</b>
<b>1 KALIBRACE</b> .....	<b>11</b>
<b>2 PŘÍSTROJE</b> .....	<b>13</b>
2.1 KALIBRÁTOR FLUKE 5500A .....	13
2.1.1 <i>Technické parametry</i> .....	13
2.2 WATTMETR LMG500 .....	15
2.2.1 <i>Technické parametry</i> .....	15
<b>3 LABVIEW</b> .....	<b>17</b>
3.1 ČELNÍ PANEL (FRONT PANEL) .....	18
3.1.1 <i>Prvky čelního panelu</i> .....	19
3.1.2 <i>Nástrojová lišta čelního panelu</i> .....	20
3.2 BLOKOVÝ DIAGRAM (BLOCK DIAGRAM) .....	21
3.2.1 <i>Nástrojová lišta blokového diagramu</i> .....	22
3.2.2 <i>Prvky blokového diagramu a datové typy</i> .....	23
3.2.3 <i>Datový tok</i> .....	24
3.3 PALETY .....	25
3.3.1 <i>Paleta Tools</i> .....	25
3.3.1 <i>Paleta Controls</i> .....	26
3.3.2 <i>Paleta Functions</i> .....	27
3.4 PROGRAMOVÉ STRUKTURY .....	28
3.4.1 <i>Smyčka For</i> .....	28
3.4.2 <i>Smyčka While</i> .....	29
3.4.3 <i>Struktura Case</i> .....	30
3.4.4 <i>Struktura Sequence</i> .....	31
3.5 PRÁCE S DATOVÝMI SOUBORY .....	32
3.5.1 <i>Funkce pro práci se soubory</i> .....	32
3.5.2 <i>Rozdíl mezi high-level a low-level funkcemi</i> .....	33
<b>4 PROGRAM</b> .....	<b>34</b>
4.1 ČELNÍ PANEL .....	34
4.2 BLOKOVÝ DIAGRAM .....	37
4.2.1 <i>Nastavení sériových portů</i> .....	37
4.2.2 <i>Smyčka While</i> .....	37
4.2.3 <i>Spouštěcí smyčka For</i> .....	41
4.2.4 <i>Měřicí smyčka For</i> .....	43
4.2.5 <i>Podprogramy pro vyhodnocení měření a zápis dat</i> .....	46
<b>ZÁVĚR</b> .....	<b>51</b>
<b>SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ</b> .....	<b>52</b>



## Seznam symbolů a zkratek

<i>DAQ</i> .....	Data acquisition (česky pořizování dat)
<i>GPIB</i> .....	General Purpose Interface Bus (také IEEE 488)
<i>LabVIEW</i> .....	Laboratory Virtual Instruments Engineering Workbench (česky laboratorní pracoviště virtuálních přístrojů)
<i>NI</i> .....	National Instruments
<i>RS – 232</i> .....	Recommended Standard 232 (sériový port)
<i>RTD</i> .....	Resistance Temperature Detectors
<i>SubVI</i> .....	podprogram
<i>USB</i> .....	Universal Serial Bus
<i>VI</i> .....	virtuální přístroj
<i>VISA</i> .....	prostředí pro komunikaci s externími zařízeními (Virtual Instrument Software Architecture)

## Úvod

Předkládaná diplomová práce je zaměřena na využití programovacího jazyka LabVIEW při kalibraci wattmetru LMG500. Hlavním cílem je vytvoření programu pro automatizovaný běh kalibrace v programovacím jazyce LabVIEW a jeho úspěšné otestování. Wattmetr může být osazen až osmi měřícími kanály. Výstupem této práce by měl být funkční program, který je schopen automaticky vyhodnocovat přesnost měření, určovat, zda měřené kanály na wattmetru vyhovují třídě přesnosti, a vypisovat naměřené hodnoty a vypočtené chyby do Excelu.

Text práce je rozdělen do čtyř částí. První část se zabývá kalibracemi. Druhá část obsahuje technické údaje o použitých přístrojích. Třetí část shrnuje základní informace o programovacím jazyce LabVIEW. Tato kapitola představuje jak prvky LabVIEW (čelní panel, blokový diagram a pracovní palety), tak i to, jakým principem programovací jazyk pracuje, jelikož tento jazyk není textový, ale patří do skupiny grafických jazyků a je řízen datovým tokem, což znamená, že je schopen paralelních běhů programu. Čtvrtá část obsahuje vytvořený program a vysvětlení funkce programu a podprogramů.

# 1 Kalibrace

Kalibrace je činnost, která za specifikovaných podmínek stanoví vztah mezi hodnotami veličiny, které jsou naměřeny na kalibrovaném přístroji, a odpovídajícími hodnotami, které jsou produkovány etalony. Kalibrace určuje metrologické charakteristiky přístroje, systému nebo referenčního materiálu. K tomu se většinou využívá přímé porovnání s etalony nebo certifikovanými referenčními materiály. Po vyhodnocení kalibrace se vystavuje kalibrační list a kalibrované měřidlo se opatřuje štítkem. [1]

Nejistota měření je parametr vztahující se k výsledku měření, který charakterizuje rozptýlení hodnot, které je možné přiřadit k měřené veličině. Je kvantitativní mírou kvality výsledku měření, umožňující porovnat výsledky měření s jinými výsledky, referencemi, specifikacemi nebo etalony. Nejistota měření může být stanovena různými způsoby. Zdroje nejistot mohou být: [2]

- Nevhodný výběr přístroje
- Nevhodný výběr vzorků měření
- Nedokonalá definice měřené veličiny
- Zaokrouhlování
- Subjektivní vliv obsluhy
- Nepřesnost etanolů a referenčních materiálů
- Nedodržení shodných podmínek při opakovaných měřeních
- Neznámé nebo nekompenzované vlivy prostředí

Při kalibraci wattmetru je potřeba dodržovat referenční podmínky prostředí: [2]

- Teplota prostředí  $(23 \pm 1)^\circ\text{C}$  pro wattmetry třídy přesnosti 0,05 až 0,3,  $(23 \pm 2)^\circ\text{C}$  pro wattmetry třídy přesnosti 0,5 a horší
- Relativní vlhkost vzduchu (40 až 60) %

Základní chyba  $\delta_0$  se vyjadřuje v procentech měřicího rozsahu a vypočítá se ze vzorce: [2]

$$\delta_0 = \frac{A_N - A_S}{A_m} \times 100 (\%)$$

$A_N$ -naměřená hodnota na kalibrovaném wattmetru

$A_S$ -nastavená hodnota na etalonového kalibrátoru

$A_M$ -maximální hodnota měřicího rozsahu

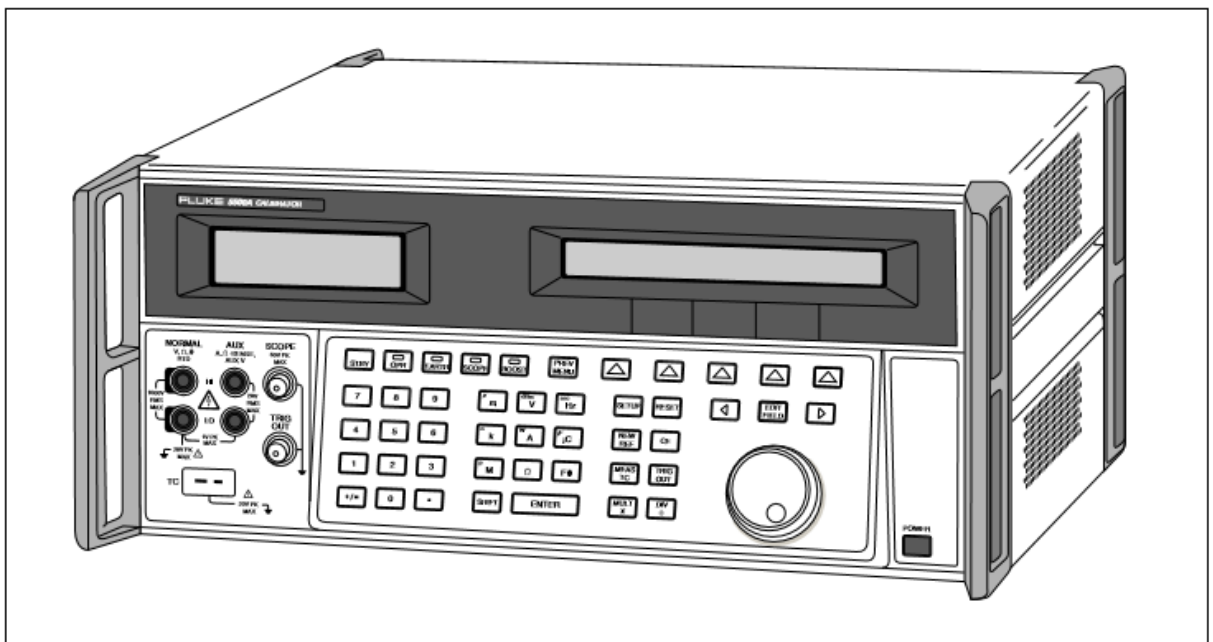
Elektronická měřidla vyšší přesnosti mohou od výrobce obsahovat údaj o době, tzv. doba náběhu, po kterou musí být přístroj zapnut, než dosáhne své plné přesnosti. Po uplynutí této doby je možné začít s měřením. [3]

Dojde-li k případu, že kalibrovaný wattmetr nevyhovuje požadavkům na některém měřicím rozsahu při zkoušce na základní chyby tak, že vyhovuje vyšší (horší) třídě přesnosti, může být do této třídy přesnosti přerazen. Poté musí být změna třídy přesnosti viditelně označena na wattmetru (na štítku přístroje). V kalibrační listě musí být tento úkon zaznamenán. V případě, kdy je zjištěná chyba větší, musí být z přístroje odstraněn kalibrační štítek. [2]

## 2 Přístroje

### 2.1 Kalibrátor Fluke 5500A

Kalibrátor Fluke 5500A je přesný přístroj, který slouží pro kalibraci široké škály elektrických měřicích přístrojů. Kalibrátor 5500A může být využit pro kalibraci přesných multimetrů, které měří střídavé nebo stejnosměrné napětí, střídavý nebo stejnosměrný proud, střídavý nebo stejnosměrný výkon, odpor, kapacitu a teplotu. Přístroj 5500A může být použit ke kalibraci analogových a digitálních osciloskopů. [4],[5]



Obr. 2.1 Kalibrátor Fluke 5500A [3]

#### 2.1.1 Technické parametry

Kalibrátor může být řízený z čelního panelu v lokálním režimu nebo vzdáleně pomocí portů RS-232 nebo IEEE-488. Pro oba typy portů jsou ve většině případů komunikační příkazy stejné. Použité komunikační příkazy pro RS-232 jsou rozebrány v kapitole 4. K dispozici jsou dva porty RS-232 na zadním panelu přístroje: SERIAL 1 FROM HOST a SERIAL 2 TO UUT. Oba porty jsou určeny pro sériovou komunikaci během kalibračních procedur. [5],[6]

Kalibrátor Fluke 5500A je plně programovatelný zdroj přesnosti pro: [5]

- DC napětí od 0 V do +1020 V
- AC napětí od 1 mV do 1020 V, s výstupem od 10 Hz do 500 kHz
- AC proud od 0,01  $\mu$ A do 11,0 A, s výstupem od 10 Hz do 10 kHz

- DC proud od 0 do +11,0 A
- hodnoty odporu od zkratu do 330 M $\Omega$
- hodnoty kapacity od 330 pF do 1100  $\mu$ F
- simulovaný výstup pro tři typy odporových teplotních detektorů (RTD)
- simulovaný výstup pro devět typů termočlánků

### DC Voltage Specifications

Range	Absolute Uncertainty, tcal $\pm 5$ °C $\pm$ (% of output + $\mu$ V)				Stability 24 hours, $\pm 1$ °C $\pm$ (ppm output + $\mu$ V)	Resolution $\mu$ V	Maximum Burden <sup>[1]</sup>
	90 days		1 year				
0 to 329.9999 mV	0.005	3	0.006	3	5 ppm + 1	0.1	50 $\Omega$
0 to 3.299999 V	0.004	5	0.005	5	4 + 3	1	10 mA
0 to 32.99999 V	0.004	50	0.005	50	4 + 30	10	10 mA
30 to 329.9999 V	0.004	500	0.0055	500	4.5 + 300	100	5 mA
100 to 1020.000 V	0.0045	1500	0.0055	1500	4.5 + 900	1000	5 mA
<b>Auxiliary Output (dual output mode only) <sup>[2]</sup></b>							
0 to 329.999 mV	0.03	350	0.04	350	30 + 100	1	5 mA
0.33 to 3.3 V	0.03	350	0.04	350	30 + 100	10	5 mA
[1] Remote sensing is not provided. Output resistance is < 5 m $\Omega$ for outputs $\geq$ 0.33 V. The AUX output has an output resistance of < 1 $\Omega$ .							
[2] Two channels of dc voltage output are provided.							

Obr. 2.2 Nejistota kalibrátoru pro rozsahy DC napětí [5]

### DC Current Specifications

Range	Absolute Uncertainty, tcal $\pm 5$ °C $\pm$ (% of output + $\mu$ A)				Resolution	Compliance Voltage	Maximum Inductive Load
	90 days		1 year				
0 to 3.29999 mA	0.010	0.05	0.013	0.05	0.01 $\mu$ A	4.5 V	1 $\mu$ H
0 to 32.9999 mA	0.008	0.25	0.01	0.25	0.1 $\mu$ A	4.5 V	200 $\mu$ H
0 to 329.999 mA	0.008	3.3	0.01	3.3	1 $\mu$ A	4.5 to 3.0 V <sup>[1]</sup>	200 $\mu$ H
0 to 2.19999 A	0.023	44	0.03	44	10 $\mu$ A	4.5 to 3.4 V <sup>[2]</sup>	200 $\mu$ H
0 to 11 A	0.038	330	0.06	330	100 $\mu$ A	4.5 to 2.5 V <sup>[3]</sup>	200 $\mu$ H
<b>5725A Amplifier</b>							
0 to 11 A	0.03	330	0.04	330	100	4 V	400 $\mu$ H
[1] The actual voltage compliance (Vc) is a function of current output (Io), and is given by the formula: Vc = -5.05*Io+4.67. The highest compliance voltage is limited to 4.5 V.							
[2] The actual voltage compliance (Vc) is a function of current output (Io), and is given by the formula: Vc = -0.588*Io+4.69. The highest compliance voltage is limited to 4.5 V.							
[3] The actual voltage compliance (Vc) is a function of current output (Io), and is given by the formula: Vc = -0.204*Io+4.75. The highest compliance voltage is limited to 4.3 V.							

Obr. 2.3 Nejistota kalibrátoru pro rozsahy DC proudu [5]

## 2.2 Wattmetr LMG500

ZES ZIMMER Electronic Systems GmbH sídlí v Oberurselu v Německu. Firma působí v oboru již přes tři desetiletí a specializuje se návrhy, vývoj, výrobu a prodej analyzátorů výkonu. Multikanálový měřič výkonu LMG500 rozšiřuje produktový rozsah multimetru ZES o měření výkonu. Využívá zkušeností a know-how z úspěšné řady ZES LMG310, LMG95 a LMG450. [7]

Vzhledem k vysoké vzorkovací frekvenci, která se používá v tomto přístroji, je možné provádět přesná měření výkonu a účinnosti v konfiguracích 1 až 8 fázových systémů s různými složkami zátěže a signálů, které obsahují frekvence v rozsahu od DC do 10MHz. Konkrétně může být přístroj využit pro měření na střídačích, spínacích zdrojích, osvětlení a spínané výkonové elektronice s krátkou dobou trvání impulsu <100ns. [8]

### 2.2.1 Technické parametry

Nejdůležitější vlastnosti modelu LMG500 jsou: [8],[9]

- Široký rozsah dynamického měření od 3 V do 1000 V / 3200 V<sub>peak</sub>, 20 mA až 32 A / 120 A<sub>peak</sub> v přímém měření pro každý vstup napětí a proudu
- Modulárnost s možností zapojení 1 až 8 kanálů pro měření výkonu
- Přesnost měření 0,015% čtení + 0,01% rozsahu při 45-65 Hz
- Až osm měřicích kanálů, z nichž každý se vzorkuje absolutně synchronně s 3MSa / s
- Zachycení přechodových a rychlých změn signálů při spouštění událostí paralelně s probíhajícími měřeními
- Proudové a napěťové cesty kanálů pro měření výkonu jsou všechny izolované proti sobě a proti zemi
- Analogové a digitální vstupy / výstupy
- Harmonické a interharmonické až 50 kHz interně a až 1 MHz s externím PC
- Kmitání, interakce mezi sítí a zátěží
- Ergonomické uživatelské rozhraní pro snadné a intuitivní použití analyzátoru výkonu
- Grafický barevný displej pro zobrazení naměřených hodnot, tvaru vlny, diagramů vypočtených hodnot, sloupcových grafů a vektorových diagramů
- Hodnocení měření v reálném čase
- Datová komunikační rozhraní s vysokou přenosovou rychlostí (RS232, USB,

## IEEE488.2, Ethernet)

Pro stavbu programu je důležité znát, jaké má wattmetr rozsahy, přesnost přístroje pro vyhodnocení měření a komunikační příkazy. Pro měření stejnosměrných napětí a proudů je dovolená odchylka do 0,08% (viz obr. 2.4).

Measuring accuracy	± (% of measuring value ± % of measuring range)									
	DC	0.05Hz...45Hz	45Hz...65Hz	65Hz...3kHz	3kHz...15kHz	15kHz...100kHz	100kHz...500kHz	500kHz...1MHz	1MHz...3MHz	3MHz...10MHz
Voltage										
$U^*$	0.02+0.06	0.02+0.03	0.01+0.02	0.02+0.03	0.03+0.06	0.1+0.2	0.5+1.0	0.5+1.0	3+3	$f/1MHz^{1.2} + f/1MHz^{1.2}$
$U_{\text{sensor}}$	0.02+0.06	0.015+0.03	0.01+0.02	0.015+0.03	0.03+0.06	0.2+0.4	0.4+0.8	0.4+0.8	$f/1MHz^{0.7} + f/1MHz^{1.5}$	$f/1MHz^{0.7} + f/1MHz^{1.5}$
Current										
$I^*$ (20mA ... 5A)	0.02+0.06	0.015+0.03	0.01+0.02	0.015+0.03	0.03+0.06	0.2+0.4	0.5+1.0	0.5+1.0	$f/1MHz^2$	-
$I^*$ (10A ... 32A)					0.1+0.2	0.3+0.6	0.5+1.0	0.5+1.0	$f/1MHz^2$	-
$I_{\text{HP}}^*$					0.03+0.06	0.2+0.4	0.4+0.8	0.4+0.8	$f/1MHz^2$	-
$I_{\text{HP}}^*$					0.03+0.06	0.2+0.4	0.4+0.8	0.4+0.8	$f/1MHz^2$	-
Power										
$U^* / I^*$ (20mA ... 5A)	0.032+0.06	0.028+0.03	0.015+0.01	0.028+0.03	0.048+0.06	0.24+0.3	0.8+1.0	0.8+1.0	$f/1MHz^{3.2} + f/1MHz^{2.5}$	-
$U^* / I^*$ (10A ... 32A)					0.104+0.13	0.32+0.4	$f/100kHz^2 + f/100kHz^2$	-	-	-
$U^* / I_{\text{HP}}^*$					0.048+0.06	0.24+0.3	0.8+1.0	0.8+1.0	$f/1MHz^{3.2} + f/1MHz^{2.5}$	-
$U^* / I_{\text{HP}}^*$					0.048+0.06	0.24+0.3	0.72+0.9	0.72+0.9	$f/1MHz^2 + f/1MHz^{2.3}$	$f/1MHz^{1.5} + f/1MHz^{1.4}$
$U_{\text{sensor}} / I^*$ (20mA ... 5A)		0.024+0.03		0.024+0.03	0.048+0.06	0.32+0.4	0.72+0.9	0.72+0.9	$f/1MHz^{1.4} + f/1MHz^{1.8}$	-
$U_{\text{sensor}} / I^*$ (10A ... 32A)					0.104+0.13	0.40+0.5	$f/100kHz^2 + f/100kHz^2$	-	-	-
$U_{\text{sensor}} / I_{\text{HP}}^*$					0.048+0.06	0.32+0.4	0.72+0.9	0.72+0.9	$f/1MHz^{1.4} + f/1MHz^2$	-
$U_{\text{sensor}} / I_{\text{HP}}^*$					0.048+0.06	0.32+0.4	0.64+0.8	0.64+0.8	$f/1MHz^{1.12} + f/1MHz^{1.5}$	$f/1MHz^{1.12} + f/1MHz^{1.5}$
Additional measurement uncertainty	in the ranges from 10A to 32A: $\pm (I_{\text{rms}})^2 \cdot 50 \mu A / A^2$									
Accuracies based on:	1. sinusoidal voltage and current 2. ambient temperature (23 ± 3) °C 3. warm up time 1h 4. definition of power range as the product of current and voltage range, $0 \leq  A  \leq 1$ ( $A$ – power factor – P/S) 5. calibration interval 12 months									
Other values	All other values are derived from the current, voltage and active power values. Accuracies for derived values depend on the functional relationship (e.g. $S = I \cdot U$ , $AS/S = \Delta I/I + \Delta U/U$ )									

Obr. 2.4 Přehled přesností wattmetru pro měřené veličiny [9]

Tab. 2.1 Rozsahy napěťových a proudových senzorů [9]

Napětí [V]	3	6	12,5	25	60	130	250	400	600	1000		
Proud [A]	0,02	0,04	0,08	0,15	0,3	0,6	1,2	2,5	5	10	20	32



### 3 LabVIEW

National Instruments (NI) je americká společnost, která byla založená v roce 1976 Jamesem Truchardem, Jeffreyem Kodoskym a Williamem Nowlinem. Při společném řešení vhodného připojení testovacího zařízení k počítači dali vzniknout myšlence „virtuální instrumentace“. Vývoj programovacího prostředí LabVIEW (z angl. Laboratory Virtual Instruments Engineering Workbench) neboli „laboratorní pracoviště virtuálních přístrojů“ byl zahájen v 80. letech a za „otce“ LabVIEW je považován Jeffrey Kodovsky. [10]

Tento program není klasickým textovým programovacím jazykem, ale G-jazykem („grafickým jazykem“), který je založen na myšlence, aby člověku, který je schopen vyjádřit své myšlenky do blokového diagramu, bylo umožněno je snadno vložit i do programu. Základem programu je možnost sestavit virtuální přístroje a upravovat si je dle potřeb pomocí různých funkcí, které jsou reprezentované ikonkami. Ikonky se propojují virtuálními vodiči a tím vzniká algoritmus programu. Společnost NI se stala průkopníkem ve virtuální instrumentaci uvedením LabVIEW, což podnítilo rozvoj a všeobecné povědomí o virtuální instrumentaci. [10]

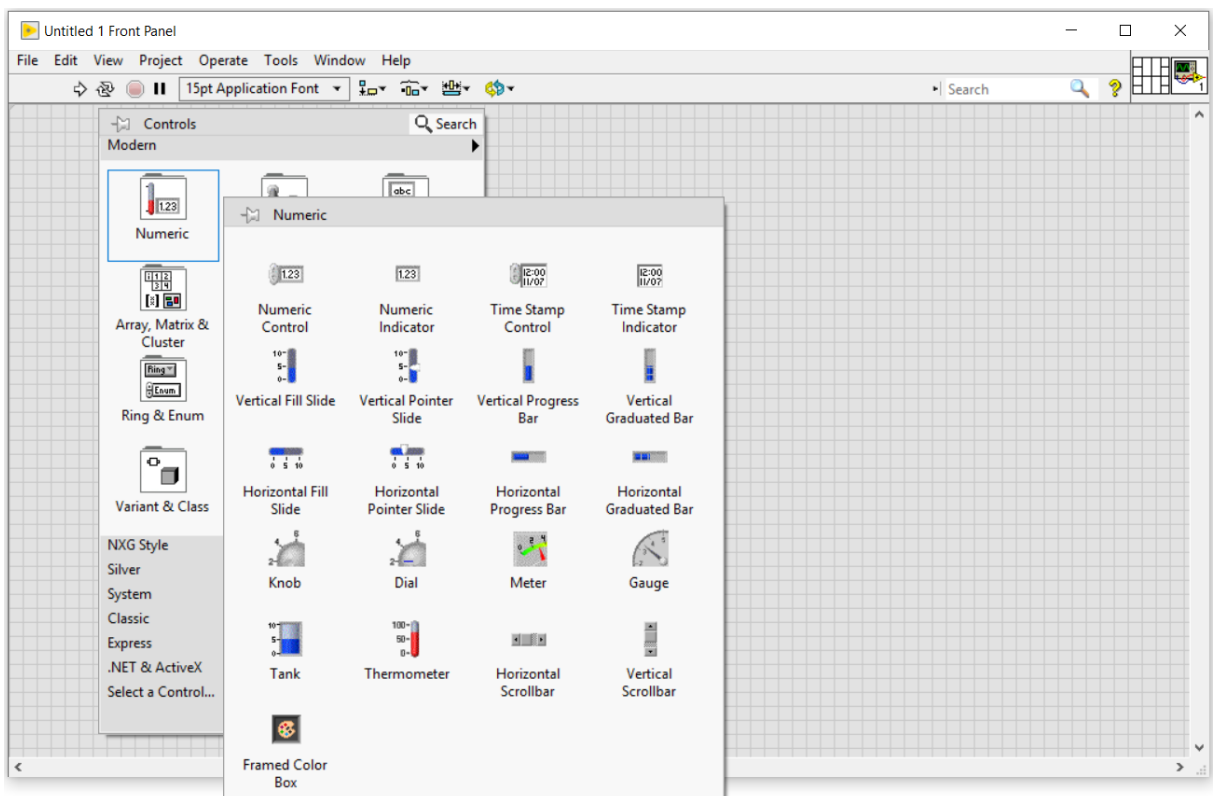
Hlavní cílem virtuální instrumentace je odstranit časově, prostorově a hlavně finančně náročné použití technických prostředků (hardware) za pomoci programových prostředků (software). Tato varianta řešení způsobuje rychlejší navrhování aplikací a jejich případnou konfiguraci, což by u skutečného přístroje ze skutečných součástí bylo složité a nákladné.

Program je tvořen pomocí dvou sdružených oken, kterými jsou čelní panel (*Front panel*) a blokový diagram (*Block diagram*). Čelní panel slouží jako vizuální prostředí pro uživatele. Naopak v blokovém diagramu jsou propojovány jednotlivé ikony či bloky, což má za důsledek tvorbu samotného algoritmu programu. [10]

### 3.1 Čelní panel (Front panel)

Čelní panel tvoří uživatelské rozhraní zvolené aplikace a určuje její vzhled i chování. Je tvořen grafickými prvky, které slouží pro ovládání a vizualizaci průběhu aplikace. Všechny objekty lze podle potřeby měnit a editovat, programově nastavovat či přizpůsobovat velikosti okna. V čelním okně lze používat širokou škálu prvků, které jsou základní součástí programovacího prostředí, nebo je možnost vytvářet své vlastní prvky (*subVI*). Grafické prvky jsou rozděleny do dvou základních skupin: vstupy (ovladače) a výstupy (zobrazovače). Ovládací prvky (*Controls*) si můžeme představit jako vstupní zařízení, stejně jako by tomu bylo na reálném přístroji (tlačítka, přepínače atd.). Zobrazovací prvky (*Indicators*) slouží jako ukazatel výstupních dat. [10],[11]

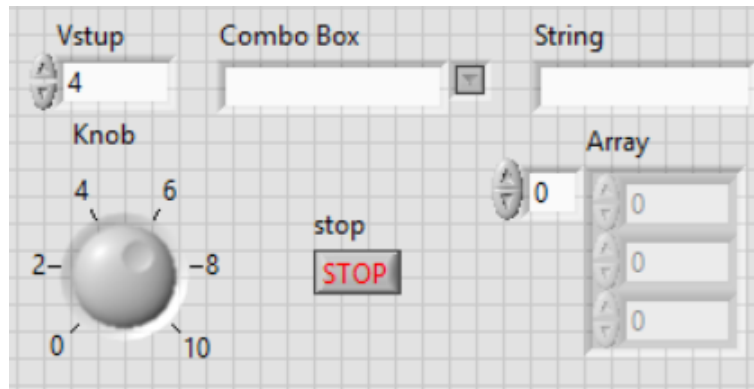
Všechny prvky mají vlastní popisek (Label), který se objeví při vložení na čelní panel, sloužící pro pojmenování objektu. Doporučuje se prvky pojmenovávat hned při vložení, jelikož se tento popisek zobrazuje i v blokovém diagramu a mohlo by dojít k záměně prvků. [10]



Obr. 3.1 Čelní panel programu s vybranou paletou knihovny Controls

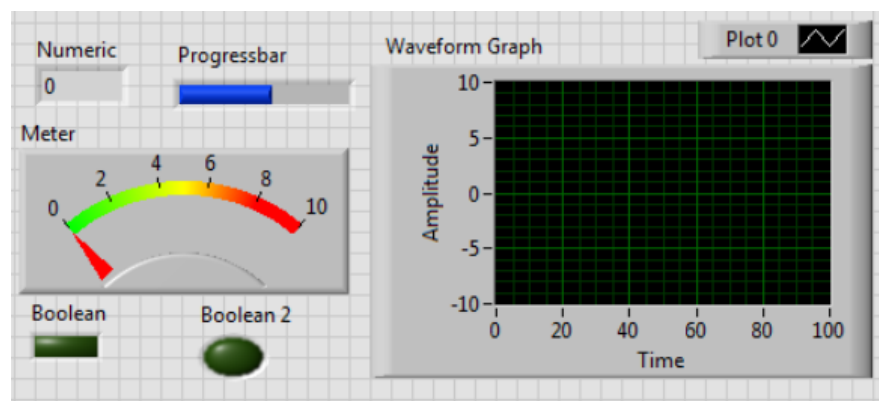
### 3.1.1 Prvky čelního panelu

Vizuální prvky čelního panelu se dělí z funkčního hlediska i principu na dvě základní skupiny, a to jsou vstupy a výstupy (**Controls** a **Indicators**). Vstupy mají různé vizuální podoby ovladačů. Patří mezi ně: tlačítka (**Button**), otočný knoflík (**Knob**), textový a číselný ovladač nebo číselné pole (**Array**) (viz obr. 3.2). [10]



Obr. 3.2 Ukázka čelního panelu s několika vstupy

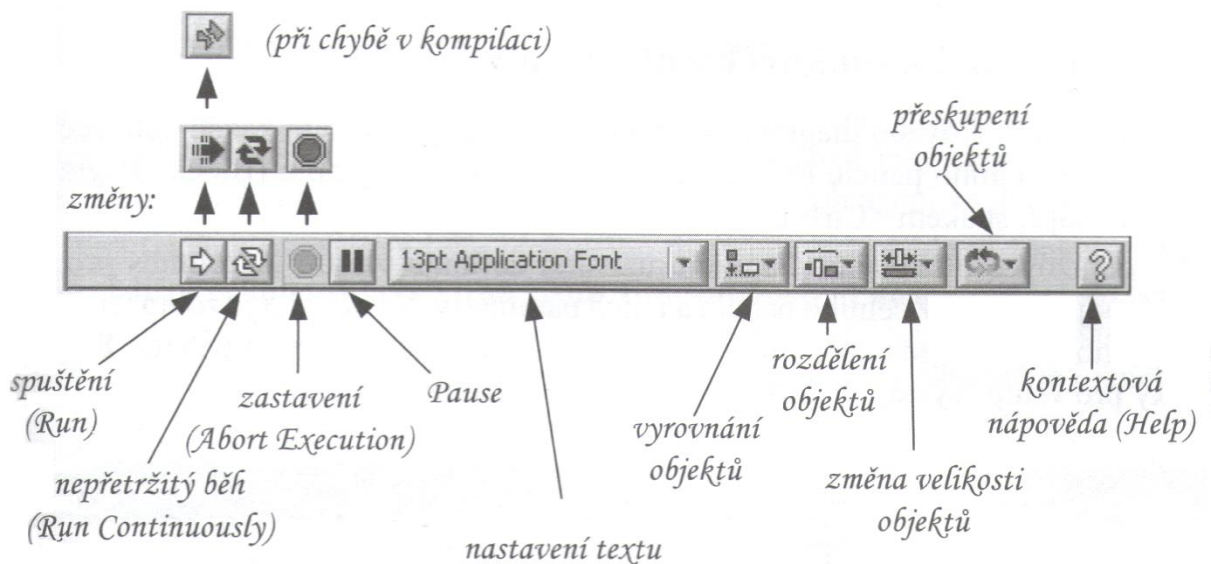
Výstupy mohou nabývat různé podoby zobrazovačů (**Indicators**). Patří mezi ně například ručičkové měřidlo (které připomíná analogový měřicí přístroj), textový nebo číselný zobrazovač, kontrolky (LED), proužkové diagramy (**Progressbar**) a různé typy grafů (viz obr. 3.3). [10]



Obr. 3.3 Ukázka čelního panelu s několika výstupy

### 3.1.2 Nástrojová lišta čelního panelu

Na nástrojové liště čelního panelu se nacházejí prvky pro ovládání a indikaci stavu programu. Všechny prvky slouží jako tlačítko a současně jako ukazatel zvoleného stavu. Tlačítkem **Run** se program spustí jen pro jeden cyklus programu. Pro opakovaný běh programu slouží tlačítko **Run Continuously**. V tomto případě pro zastavení spuštěného programu používáme **Abort Execution** nebo programově vytvořené stop tlačítko či zastavovací podmínku, která program ukončí. Pro pozastavení běhu programu se využívá **Pause**. Tlačítko **Run** také indikuje chybu při kompilaci programu tím, že přejde do stavu **Broken Run** (přerušená šipka). Program nelze spustit a vyskočí **Error list**. Zbylé funkce lišty slouží k úpravě objektů na ploše a jejich popisků. [10]



Obr. 3.4 Přehled nástrojové lišty čelního panelu[8]

## 3.2 Blokový diagram (Block diagram)

V blokovém diagramu se vytváří vlastní algoritmus programu propojováním prvků čelního panelu pomocí virtuálních vodičů (viz Obr 3.4). Program samozřejmě obsahuje pro toto okno své vlastní palety (kapitola 3.3), které umožňují vytvářet podmínky (smyčky *For*, *While*) nebo časování programu.



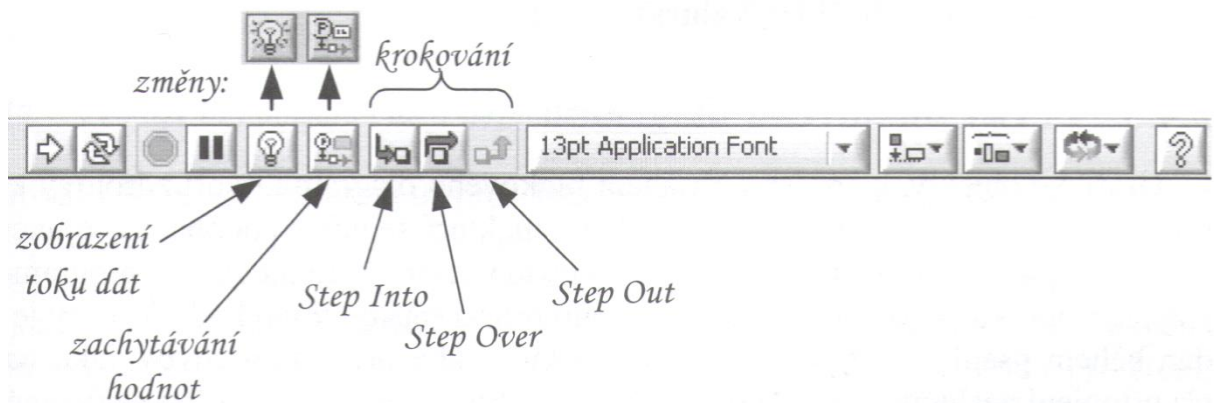
Obr. 3.5 Ukázka propojení vstupu a výstupu pomocí blokového diagramu

Blokový diagram se skládá z terminálů, SubVI, konstant, struktur a spojení, které umožňují datový tok mezi jednotlivými prvky blokového diagramu. Prvky vložené na čelním panelu jsou vždy vyobrazeny v okně blokového diagramu jako vstupy či výstupy. Vstupy a výstupy jsou obdobou parametrů a konstant v textových programovacích jazycích. [10]

Všechny prvky blokového diagramu mají podle své specifikace vyobrazeny vstupní či výstupní připojovací body (terminály). Terminály se propojují funkcí *Wiring Tool*. Terminály obsahují i informaci o datovém typu, který je třeba přivést na vstup nebo naopak jaký bude vycházet z daného pinu. Když podmínky na datový typ nejsou splněny, terminály se sice propojí, ale tento vodič bude pouze černé barvy a při pokusu o spuštění programu se zobrazí *Error list* a program nebude spuštěn. Při propojování prvků se dbá na směr toku informací, zpravidla zleva doprava, aby se jednotlivé části nacházely v místě, kde mají potom být definovány podmínky, cykly nebo popřípadě jiné algoritmické struktury. Tyto programové struktury samozřejmě svůj obraz na čelním panelu nemají, jsou skryté a prezentují se běžnému uživateli pouze jako výkonné funkce, s jejich pomocí ovládá vykonávání programu. [10]

### 3.2.1 Nástrojová lišta blokového diagramu

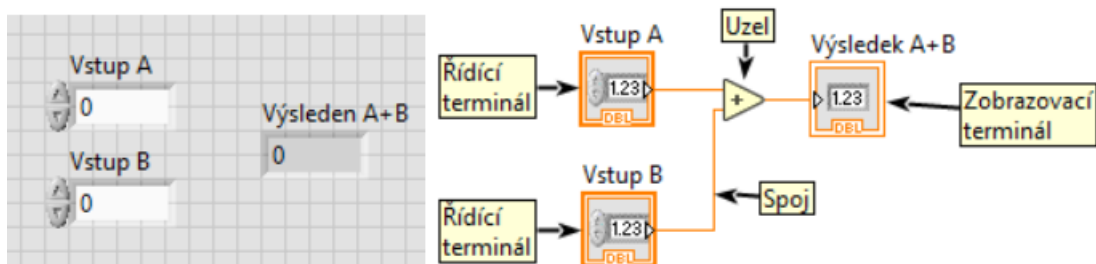
Nástrojová lišta blokového diagramu ve svém základu obsahuje stejná tlačítka jako jsou na nástrojové liště čelního panelu (**Run**, **Run Continuously**, **Pause**, **Abort Execution** a tlačítka pro úpravu objektů a popisků). Jsou zde ale navíc tlačítka určená k ladění programu. Tato tlačítka se využívají pro pomoc při ladění programu, kdy je například sledován datový tok z prvků. Tlačítka **Step** se používají ke krokování programu z uzlu do uzlu, kde uzel je chápán jako funkce, struktura či SubVI. Funkce **Breakpoint**, která slouží k zachytávání dat na určitém spoji či uzlu a k pozastavení programu. Posledním novým symbolem je **Highlight Execution** (symbol žárovky). Při použití této funkce se zobrazí datový tok na jednotlivých vodičích a běh programu se zpomalí, což umožňuje sledovat přehledně celý program a hledat případné chyby v kódu. Po ukončení odlaďování programu je důležité symbol žárovky opět vypnout, protože v momentu, kdy program tvoří několik SubVI a v jednom podprogramu by zůstal aktivován, průběh programu bude zpomalen, což je nežádoucí. [10]



Obr. 3.6 Ukázka propojení vstupu a výstupu pomocí blokového diagramu[8]

### 3.2.2 Prvky blokového diagramu a datové typy

Vizuální prvky, jež obsahuje blokový diagram, vytvářejí zdrojový kód programu. Blokový diagram, který se podobá vývojovému diagramu, v textovém jazyce odpovídá řádkům textu. Blokový diagram tvoří vzájemně propojené prvky, které vykonávají určitou funkci v programu. Na obr. 3.6 je uveden příklad VI s vyobrazením tří typů součástí, ze kterých je složen blokový diagram: uzel, terminál a spoj. [10]







Obr. 3.7 Příklad VI popisující uzly, terminály a spoje

- **Uzel (Node)** – Je prvkem pro vykonávání programu. Lze si ho představit jako příkaz, funkci nebo podprogram v textovém programovacím jazyce. LabVIEW rozlišuje tři druhy uzlů: funkce, SubVI a struktury. Funkce provádějí různé základní operace (práce se soubory, sčítání apod.) a jsou základními prvky blokového diagramu. SubVI jsou vlastně VI, které jsou použité jen jako součást (podprogram) jiného VI. Struktury mají za úkol řízení průběhu programu (smyčka For, While apod.).
- **Terminál (Terminal)** – Je branou, přes kterou procházejí data mezi čelním panelem a blokovým diagramem. Terminál by v textovém programovacím jazyce reprezentoval proměnnou nebo konstantu. Terminály se dělí na zdrojové (Control), což jsou vždy začátky datových cest, a na koncové (Indicator), což jsou konce datových cest, kde se například data zobrazí v číselném indikátoru, grafu či se zapíše do souboru.
- **Spoj (Wire)** – Tvoří cestu pro datový tok mezi terminálem a uzlem a v běžném programovacím jazyce odpovídá proměnným. Blokový diagram může obsahovat různé typy prvků, proto i spoje mohou být různých typů v závislosti na datovém typu a mohou se lišit barvou a tloušťkou.

Tab. 3.1 Základní datové typy a jejich barva v LabVIEW

Datový typ spoje	Barva čáry
celočíselný (Integer)	Modrá
číselný s pohyblivou čárkou (Floating point)	Oranžová
binární (Boolean)	Zelená
znakový řetězec (String)	Fialová

Tab. 3.2 Tloušťka spoje v závislosti na datovém typu

Datový typ	Typ čáry
skalár	
1D pole	
2D pole	
Klastr ( <i>cluster</i> )	

Spoje nabývají stejné barvy, jakou jsou označeny připojovací terminály. Při propojování prvků (jejich vstupů a výstupů) nelze vzájemně propojovat prvky různých vzájemně neslučitelných datových typů. Je-li například v blokovém diagramu tlačítko ohraničeno zeleným rámečkem (datový typ boolean) může být spojeno s dalším vstupem prvku se zeleným rámečkem. Výjimku tvoří propojení prvku oranžové barvy (reálné číslo) s barvou modrou (celá čísla), zde nedojde k chybové hlášce, ale dojde k nepřesnostem kvůli zaokrouhlování a ztrátě dat. Když se propojí dva vzájemně neslučitelné prvky, je to na první pohled hned znatelné, jelikož vodič zčerná a při pokusu spustit program vyskočí chybová hláška a upozorní na nepřesně spojený vodič.

### 3.2.3 Datový tok

V tomto ohledu se LabVIEW zásadně liší od textových programovacích jazyků, jelikož běh programu se vykonává postupně datovým tokem. Datový tok určuje směr vykonávání programu, proto se může, v případě nesprávného propojení programu v blokovém diagramu, docílit nechtěného paralelního běhu, který může ovlivnit výsledek celého běhu programu. Zpracování informace na jednotlivém prvku se provede pouze v případě, že všechny vstupy již obsahují přivedená data. Pokud ano, vykoná se zpracování signálů a výstup pošle data na vstup dalšího uzlu. Pokud ne, tento uzel čeká, než se vykonají další části programu, které mají za úkol dostat signál na vstupy toho prvku.



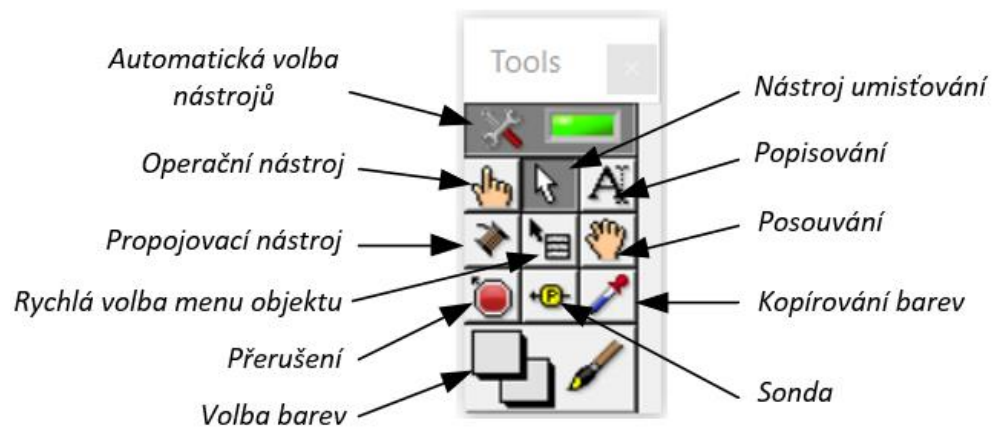
### 3.3 Palety

Palety jsou grafické panely, které obsahují různorodé nástroje a objekty pro vytváření programu. Rozlišují se tři druhy palet v LabVIEW:

- paleta nástrojů (**Tools**)
- paleta ovládacích a indikačních prvků (**Controls**) pro čelní panel
- paleta funkcí (**Functions**) pro blokový diagram

#### 3.3.1 Paleta Tools

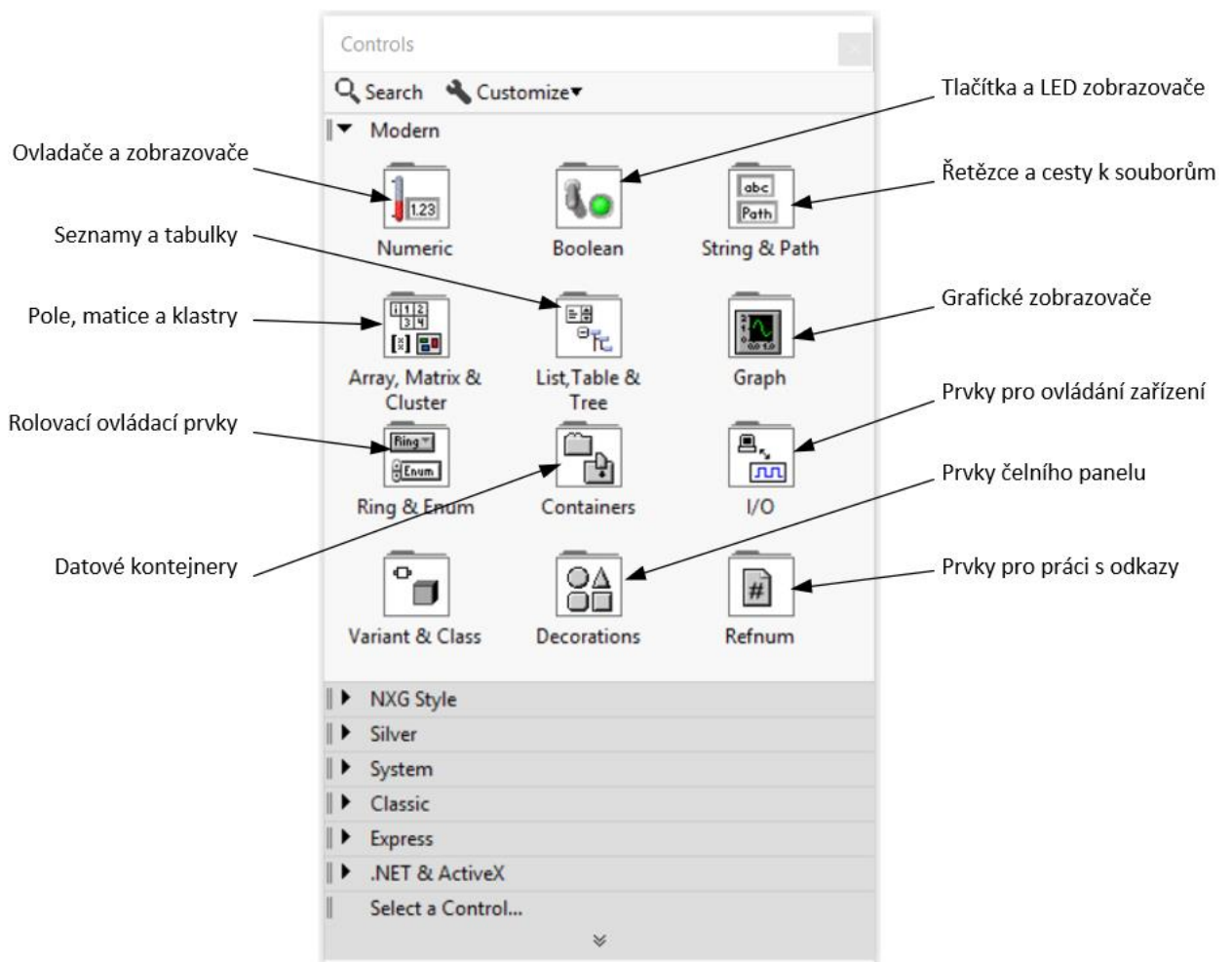
Tato paleta umožňuje základní instrukce, například posouvání oken, propojování prvků, kopírování barev. Nejdůležitější je funkce automatické volby, jelikož nabízí všechny funkce palety automaticky, například při najetí kurzorem myši na terminál se rovnou zobrazí propojovací nástroj pro vkládání vodičů. [10]



Obr. 3.8 Paleta Tools

### 3.3.1 Paleta Controls

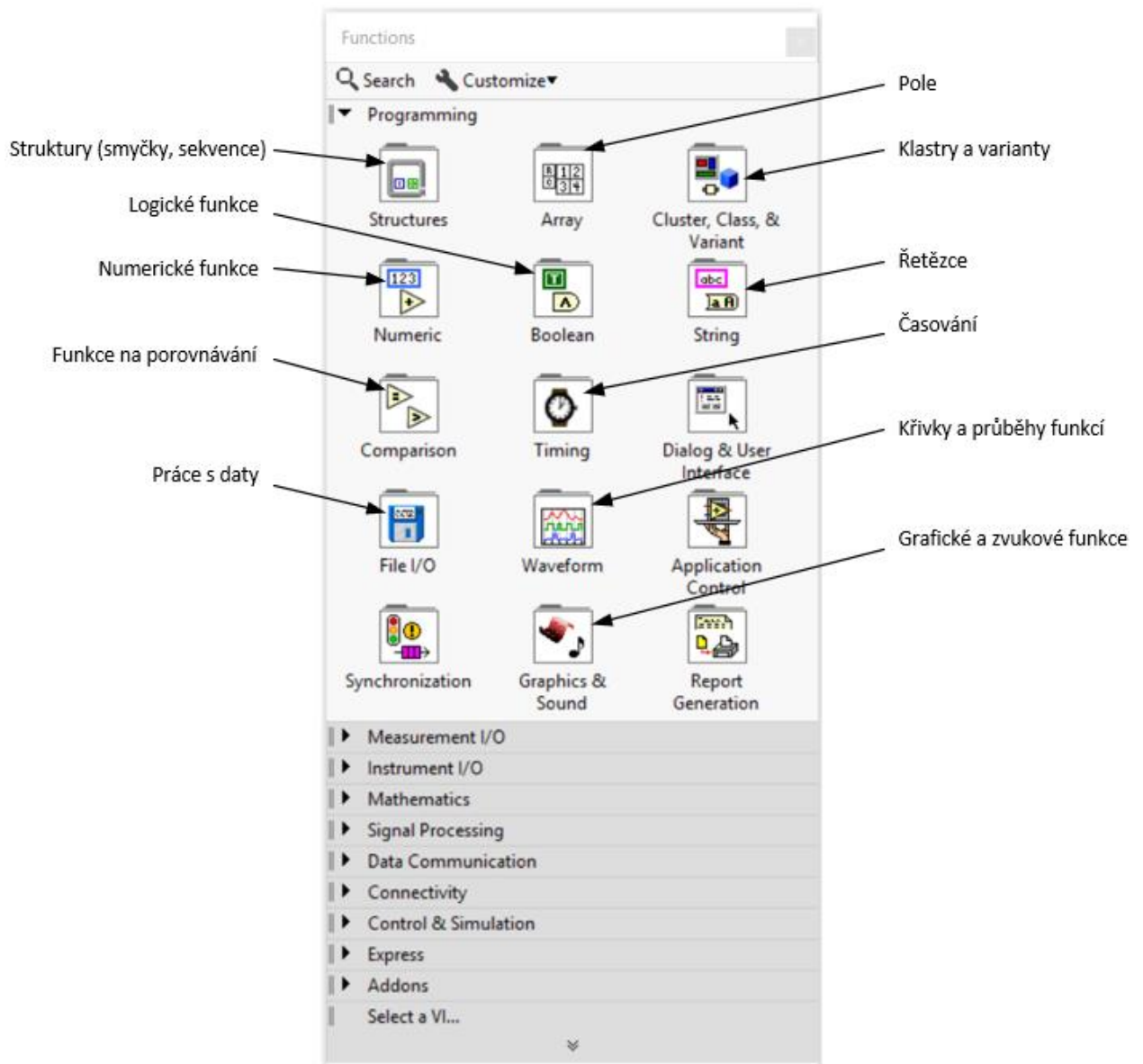
Paleta Controls je složena ze subpalet s prvky pro vytváření čelního panelu. Jedná se o různá tlačítka, knoflíky, přepínače, zobrazovače, měřiče a další řídicí a indikační prvky LabVIEW. Program obsahuje velké množství konkrétních prvků, proto bylo třeba je rozčlenit podle funkcí do menších knihoven (subpalet). Prvek se dohledá v knihovně a pouhým přetažením z knihovny umístí na čelní panel, kde se dle potřeby nakonfiguruje. Nabídka prvků se liší vlivem verze a nainstalovaných knihoven. [10]



Obr. 3.9 Paleta Controls

### 3.3.2 Paleta Functions

Paleta Functions je složena (podobně jako paleta Controls) ze subpalet s prvky pro vytváření blokového diagramu. Jedná se o knihovny se strukturami, poli, klastry, řetězci a s mnoha dalšími knihovnami. Tyto funkce a prvky jsou důležité pro stavbu VI. Prvek se dohledá v knihovně a pouhým přetažením z knihovny umístí do blokového diagramu, kde se dle potřeby nakonfiguruje. Nabídka prvků se liší vlivem verze a nainstalovaných knihoven. [10]



Obr. 3.10 Paleta Functions

### 3.4 Programové struktury

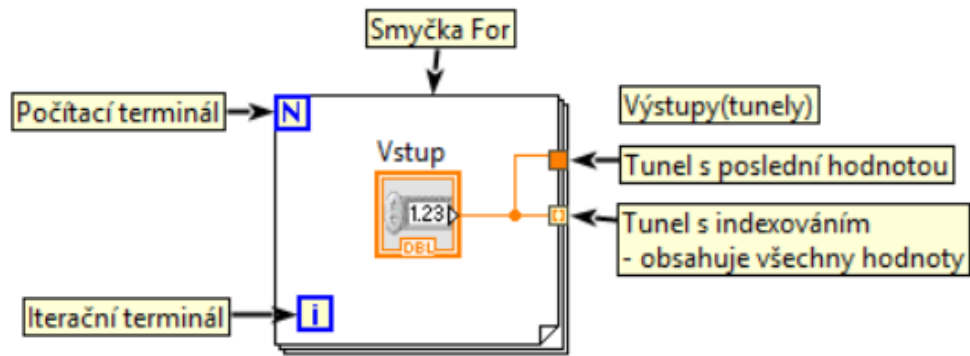
Programové struktury jsou programové prvky LabVIEW určené pro řízení průběhu výpočtu nebo měření a lze je srovnat s cykly pro větvení v textových programovacích jazycích. Vstupy a výstupy těchto struktur mají podobu tunelů, které se tvoří na hraně dané struktury. Tyto tunely mohou po ukončení cyklů obsahovat například pouze poslední hodnotu na výstupu nebo se data v tunelu mohou indexovat a vznikne tím 1D pole. V této části se seznámíme s nejčastěji využívanými strukturami: [10]

- Smyčka **For (For Loop)** – vytvoření programového cyklu s předem známým počtem opakování
- Smyčka **While (While Loop)** – vytvoření programového cyklu s opakováním v závislosti na splnění určité podmínky
- Struktura **Case** – podmíněný příkaz nebo přepínač pro větvení vykonávání algoritmu
- **Flat Sequence** – jedna z možností zajištění správného sledu prováděných funkcí (využití když není zajištěna sekvenčnost pomocí datového toku)
- **Formula Node** – možnost efektivního způsobu zadávání matematických vzorců

#### 3.4.1 Smyčka For

Tato smyčka se využívá při stavbě algoritmu, kdy je předem znám počet iterací smyčky. **For** se vykoná N-krát podle přivedené hodnoty na vstupní počítací terminál. U této struktury je nutné inicializovat, jak budou jednotlivé výstupy nastaveny na hraně struktury, kde vzniká výstupní tunel, jelikož **For** cyklus primárně tvoří 1D pole a pomocí smyčky **For** vnořené do další **For** smyčky vznikne na výstupu 2D pole. Smyčka **For** má dva terminály: [10]

- Počítací terminál slouží jako vstup pro počet cyklů smyčky, hodnotu z terminálu lze využít i uvnitř smyčky. Terminál pracuje pouze s celými čísly, v případě přivedení reálného čísla je číslo zaokrouhleno a převedeno na celé číslo (formát I32).
- Iterační terminál udává počet aktuálně vykonaných cyklů ( $i = 0$  až  $N - 1$ ). Hodnoty z tohoto terminálu lze využít uvnitř smyčky pro výpočty nebo jen jako ukazatel běhu programu.

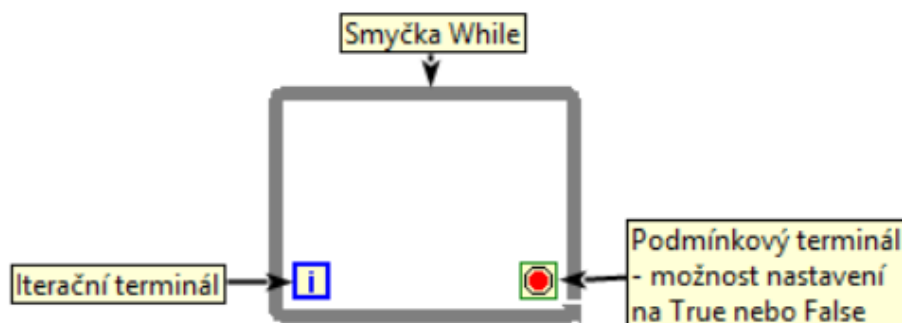


Obr. 3.11 Smyčka For

### 3.4.2 Smyčka While

Struktura **While Loop** se užívá pro vytvoření opakovaného běhu algoritmu do momentu, kdy je struktura ukončena ukončovací podmínkou. Smyčku tvoří dva terminály: [10]

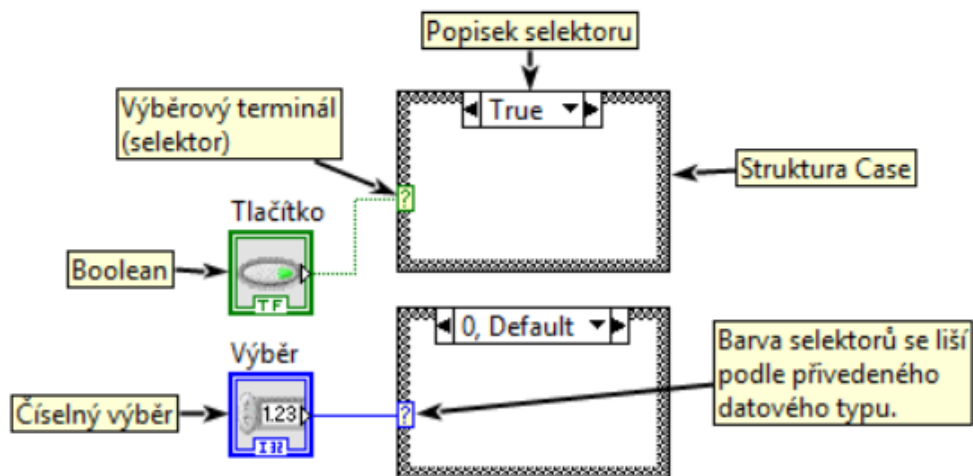
- Iterační terminál, který udává počet aktuálně vykonaných cyklů ( $i = 0$  až  $N - 1$ ). Hodnoty z tohoto terminálu lze využít uvnitř smyčky pro výpočty nebo jen jako ukazatel běhu programu.
- Podmínkový terminál, který je nejdůležitějším prvkem smyčky, slouží k ukončení smyčky v případě splnění dané podmínky. Terminál má datový formát **Boolean** a může být nastaven tak, že strukturu ukončí v případě **True** nebo **False**. Nejčastěji se k podmínkovému terminálu připojuje STOP tlačítko a terminál se nastaví na ukončovací podmínku **True**. Důležité je vložit STOP tlačítko (ukončovací podmínky) dovnitř struktury, jelikož při umístění STOP tlačítka vně struktury by se případné změny hodnoty nedostaly dovnitř struktury, protože LabVIEW pracuje na základě datového toku, což znamená, že načte hodnoty ze vstupů při spuštění struktury, ale během jejího běhu už na ni nemá nic jiného vliv.



Obr. 3.12 Smyčka While

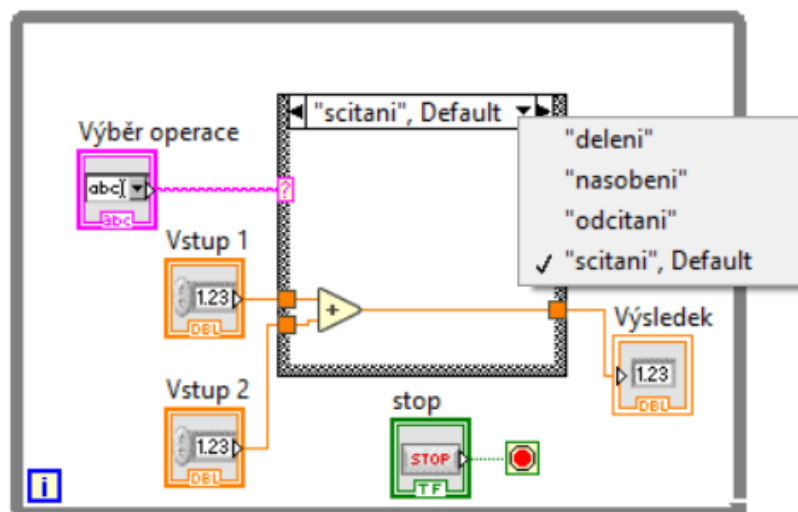
### 3.4.3 Struktura Case

Tato struktura se používá v případech, kdy je třeba pracovat s větvením programu podle podmínek. V textových programovacích jazycích by se to dalo prezentovat jako If, Case, Switch, kde se používá podmíněný příkaz. Výběr rámce se provádí signálem přivedeným na vstupní výběrový terminál (selektor). Struktura se může ovládat datovými typy Boolean nebo číselnými či textovými typy. Při použití datového typu Boolean si struktura vybírá pouze ze dvou možností (True a False). U dalších datových typů jsou rámce označeny čísly, respektive znaky. [10]



Obr. 3.13 Struktura Case

Tyto struktury lze využívat jako spouštěcí podmínky částí programu, výběr postupu zpracování dat nebo co má program měřit (napětí nebo proud). Nejjednodušší ukázkou je vytvoření programu na sčítání, odčítání, násobení a dělení, kde se pomocí Combo Boxu volí numerická operace.



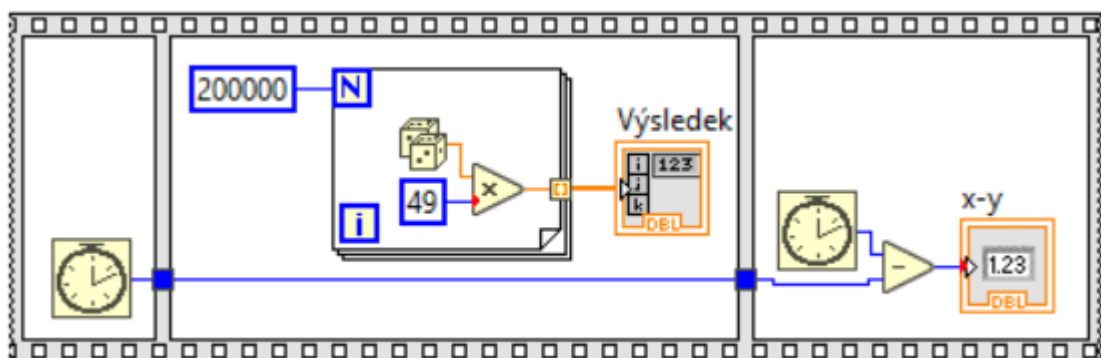
Obr. 3.14 Ukázka použití struktury Case

### 3.4.4 Struktura Sequence

Tato struktura umožňuje sekvenční vykonávání jednotlivých částí programu (rámečků). Vzhledem struktura připomíná filmový pásek, program se vykonává rámeček (snímek) po rámečku v pořadí zleva doprava. Po vykonání posledního snímku se zpřístupní výstupní data. LabVIEW rozlišuje dva druhy struktur *Sequence*: [10]

- **Flat Sequence** – Tato struktura zobrazuje všechny snímky najednou, proto se také nazývá plošná (základní). Tento typ sice nešetří místo v blokovém diagramu, ale je přehledný (při menším počtu snímků).
- **Stacked Sequence** – Tato struktura v blokovém diagramu zobrazuje vždy jen jeden snímek (podobnost se strukturou *Case*). Snímky se vykonávají podle pořadového čísla. Struktura je úsporná na prostor v blokovém diagramu, ale z hlediska přehlednosti je na tom hůře než *Flat Sequence*. Při editaci snímků je třeba listovat, což zabere čas.

LabVIEW je založeno na principu datového toku a v případech, kdy úplně nelze zajistit správný běh programu (možnost výskytu paralelního běhu) přichází na řadu právě tato struktura, která umožňuje práci s daty krok po kroku. Této možnosti lze využít při časování sledu instrukcí pro přístroje, měření dat nebo měření doby vykonávání instrukcí. Možností, kde lze tyto struktury využít, je opravdu mnoho. Na obr. 3.14 je ukázka využití struktury Flat, kde se měří jak dlouho LabVIEW trvá vygenerovat 200000 náhodných čísel. V prvním snímku se spustí časovač, ve druhém snímku se generují a zapisují hodnoty a v posledním se vypočte doba potřebná na vygenerování náhodných čísel.



Obr. 3.15 Ukázka použití struktury Flat

## 3.5 Práce s datovými soubory

Při práci s datovými soubory program může využívat paletu *File I/O*, která umožňuje zapisování dat do souborů a také jejich čtení ze souborů. Pro práci lze použít např. funkce:

- Otevření, vytvoření a zavření souborů
- Čtení nebo zápis dat
- Práce se soubory pro práci s tabulkami (spreadsheet)
- Přesouvání a přejmenování souborů a adresářů

### 3.5.1 Funkce pro práci se soubory

Základními funkcemi (low-level) umožňujícími práci se soubory jsou funkce pro vytvoření, otevření a zavření souboru, umožňující ovládání zápisu a čtení dat, kopírování nebo mazání souborů, práci s adresáři apod. Například funkce *Open\Create\Replace File* slouží k nadefinování cesty pro načtení souboru (případně vytvoření nového souboru), do kterého se budou data zapisovat nebo číst. Dále máme k dispozici komplexnější funkce (high-level), které paleta *File I/O* nabízí, u kterých už není třeba definovat otevírání, čtení, zápis a uzavření souboru, jelikož tyto instrukce jsou obsaženy v jednom bloku a jsou nakonfigurovány při tvorbě bloku. [10]

High-level funkce:

- Write To Speadsheet File.vi
- Read From Speadsheet File.vi
- Write To Measurement File.vi
- Read From Measurement File.vi





## 4 Program

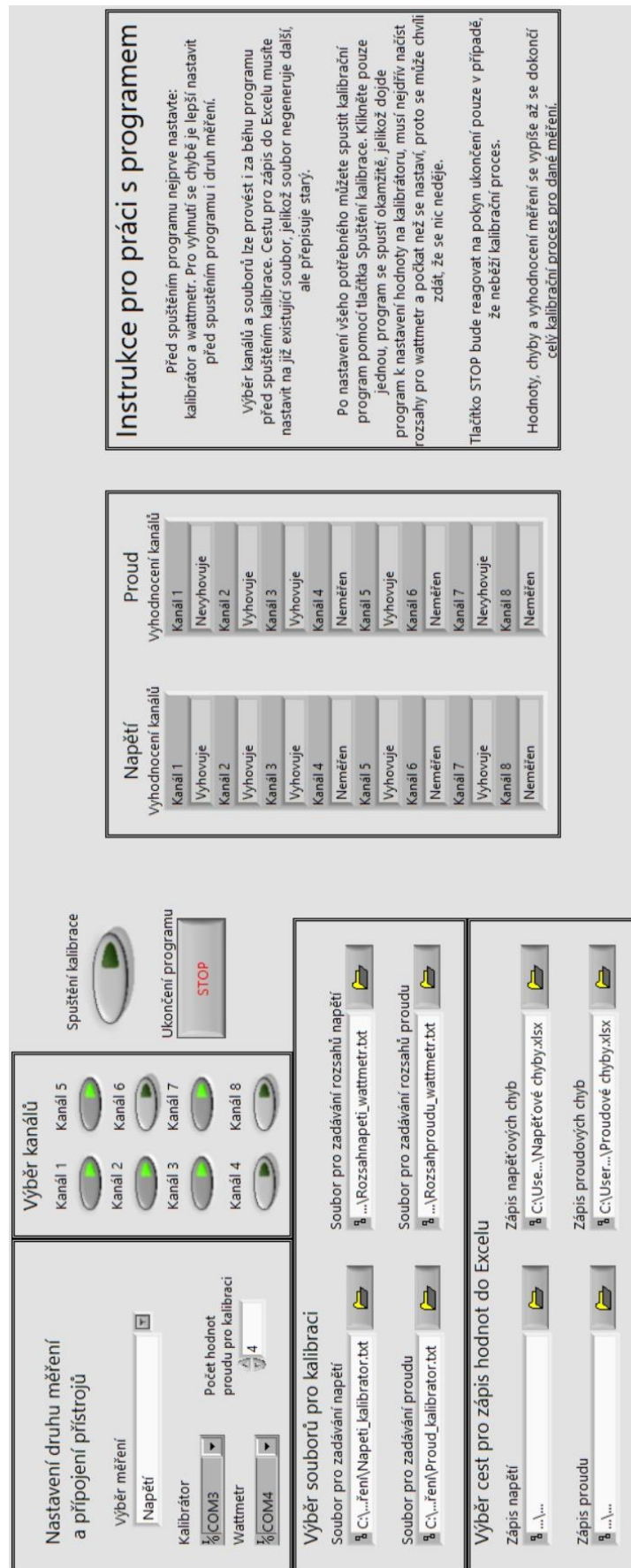
V této kapitole jsou rozebrány do detailu jednotlivé prvky programu. Program slouží ke zjištění přesnosti wattmetru LMG500. Měření je stejnosměrný proud a stejnosměrné napětí. Wattmetr je fyzicky osazen senzory na pozici 1-3 a 5-7, přičemž 6. kanál je mimo provoz. Program je vybaven možností volby jednotlivých kanálů, přičemž program je schopen měřit i kanály 4 a 8, kdyby byly na přístroj přidány.

Přístroje jsou s počítačem propojeny pomocí sériových portů. Program ovládá přístroje pomocí komunikačních příkazů (textové řetězce). Tyto řetězce obsahují požadovanou instrukci a požadovanou veličinu pro nastavení výstupu kalibrátoru a rozsahu wattmetru, kde požadovanou hodnotu program načítá z textových souborů, které si uživatel může upravit i na jiné hodnoty, ale pro kalibraci je to nepravděpodobné, jelikož všechny požadované rozsahy jsou v textovém souboru obsaženy. Jeden řádek v textovém souboru obsahuje vždy pouze jednu hodnotu veličiny či rozsahu (místo desetinných čárek se píše tečky). Pro kompletní kalibraci jsou potřeba čtyři textové soubory pro: rozsah napětí, rozsah proudu, hodnotu napětí a hodnotu proudu.

Měření je poměrně časově náročné kvůli čekání na ustálení hodnot (20 sekund) a nastavení rozsahů (14 sekund). Jeden měřicí cyklus (hodnota) trvá cca 35 sekund. Po vykonání měření program automaticky vypočte chyby a vyhodnotí jednotlivé kanály. Toto vyhodnocení se zobrazí v čelním panelu programu. Naměřené hodnoty a vypočtené chyby se zobrazí na čelním panelu a zároveň jsou uloženy do Excelu, ale pouze v případě, že byly nastaveny cesty souborů pro zápis. Program pouze přepisuje hodnoty v nastaveném Excelu a nevytváří nový soubor.

### 4.1 Čelní panel

Čelní panel je tvořen prvky pro nastavení měření a indikátory výsledků. V pravé horní části panelu jsou instrukce pro práci s programem. Levá horní část čelního panelu obsahuje řídicí prvky programu. Před spuštěním programu je nejprve důležité vybrat správný COM port přístroje, jelikož program okamžitě po zapnutí nastavuje výchozí pozici pro kalibrátor i wattmetr a mohlo by dojít k pádu či nepřesnostem měření. Poté se vybírá druh měření – jestli napětí či proud (v případě proudu se zadává i počet hodnot pro kalibraci).



Obr. 4.1 Vzhled části čelního panelu při měření



## 4.2 Blokový diagram

Blokový diagram je srdcem celého programu. Pro vysvětlení je diagram rozdělen do následujících částí: nastavení sériových portů, smyčka While, spouštěcí smyčka For, měřicí smyčka For a vyhodnocovací podprogramy.

### 4.2.1 Nastavení sériových portů

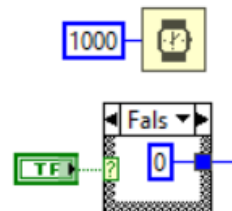
Při spuštění programu před načtením smyčky While dojde k nastavení sériových portů pro komunikaci s wattmetrem a kalibrátorem. Pro komunikační rozhraní je nastavena defaultně přenosová rychlost na 9600 Bd/s, což vyhovuje pro kalibrátor a pro wattmetr je nastaveno 115200 Bd/s.

### 4.2.2 Smyčka While

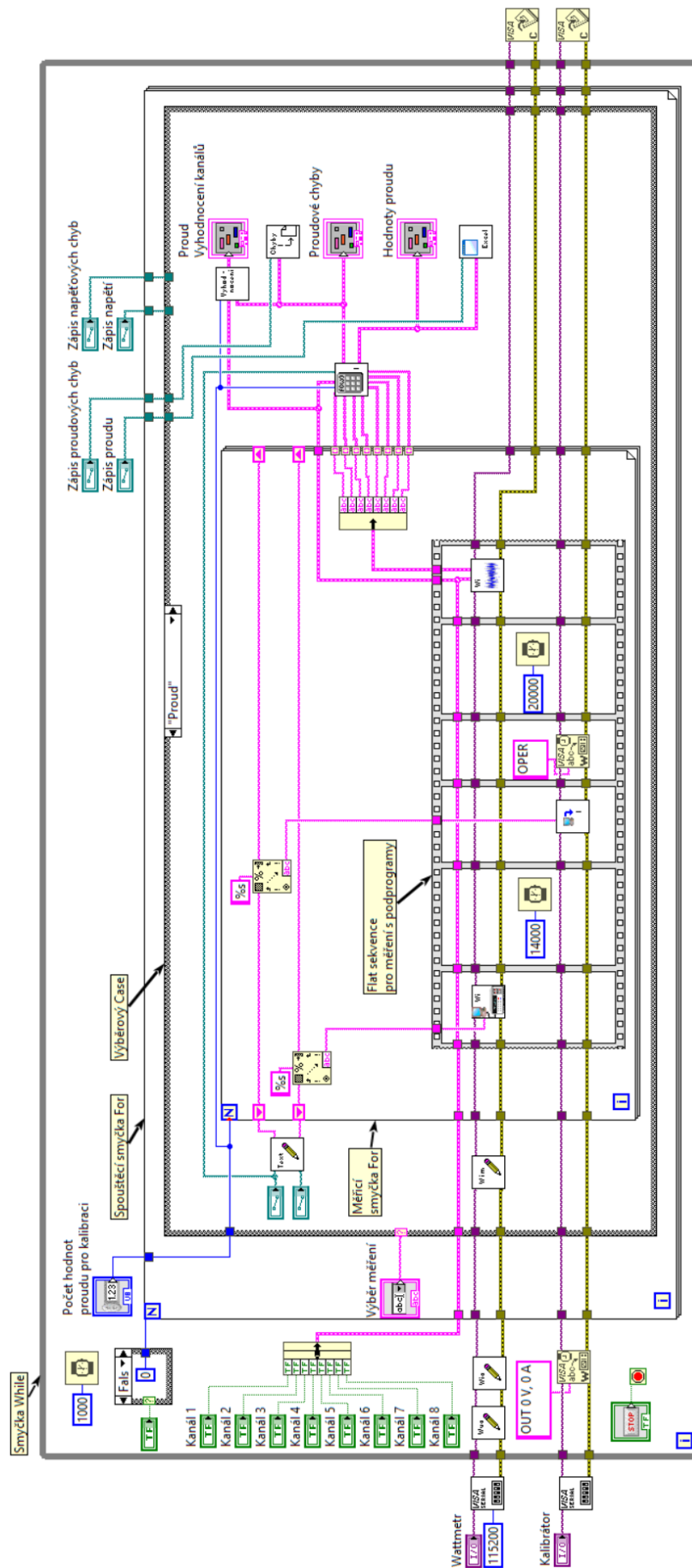
Smyčka While slouží k nastavení přístrojů do výchozího nastavení a jako vyčkávací smyčka před spuštěním kalibrace. Čas obnovy smyčky je nastaven na jednu sekundu. Ukončovací podmínka (Stop tlačítko) program ukončí pouze pokud se program nachází ve vyčkávací pozici po doměření.

- **Spouštěcí podmínka a spouštěcí smyčka For**

- Spouštěcí podmínka je řešena pomocí struktury Case, kde rozhodovací podmínka má podobu Boolean (tlačítko). Při stavu False je nastavena na výstupu „0“ a pro program v tu chvíli smyčka For „neexistuje“ (vykoná se 0x). Při stavu True bude na výstupním tunelu struktury připojeném ke smyčce For nastavena „1“, což znamená, že spouštěcí smyčka For proběhne pouze jednou a po dokončení se opět vrátí program do smyčky While a bude čekat na další pokyny. Smyčka obsahuje strukturu Case pro výběr druhu měření a jeho vyhodnocení.



Obr. 4.4 Spouštěcí podmínka a čas obnovy smyčky While



Obr. 4.5 Blokový diagram programu

- **Výběr kanálů**

- Měřený wattmetr je aktuálně osazen 6-ti kanály, z nichž jeden je bohužel mimo provoz. Wattmetr však může mít až 8 měřících kanálů. Pro tuto variabilitu přístroje byl přidán výběr kanálů, které budou měřeny. Informace o nastavení kanálů jsou sbaleny do klastru a použity při načítání dat z wattmetru a při výpočetní části.

- **Nastavení kalibrátoru do výchozí pozice**

- Při spuštění programu vždy dojde k nastavení výstupních svorek kalibrátoru na nulové hodnoty napětí a proudu. K tomuto kroku dojde i po doměření, aby byl přístroj uveden do původního stavu a nezůstalo na výstupu po doměření nastaveno nějaké napětí či proud.



Obr. 4.6 Nastavení kalibrátoru o výchozí pozice

- **Nastavení wattmetru na automatický rozsah**

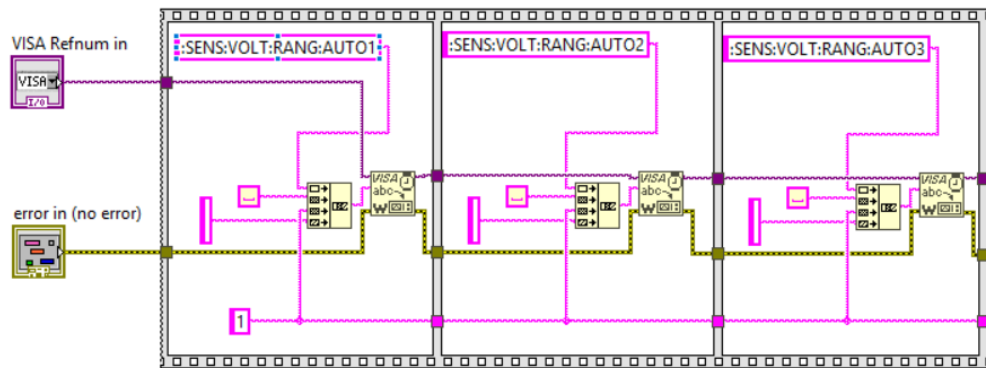
- Defaultně se při spuštění programu nastaví pro wattmetr na všech rozsazích automatická změna rozsahu. Ke změně na manuální nastavení rozsahu dochází až při měření podle výběru druhu měření. Pro tuto část kódu byly vytvořeny SubVI, jelikož nastavení všech osmi kanálů je značně prostorově neúsporné. Pokud na konci komunikačního příkazu je zvolena hodnota „1“, je wattmetr nastaven na automatický rozsah. V opačném případě je přepnut na manuální rozsah. Příkaz je sestaven pomocí *Concatenate Strings Function* a následně je odeslán skrze *VISA Write Function*. V instrukci *AUTOn* se za *n* volí číslo kanálu. Komunikační příkazy pro nastavování rozsahu auto/manuál:

- :SENS:VOLT:RANG:AUTOn 1 – automatický napěťový rozsah
- :SENS:CURR:RANG:AUTOn 1 – automatický proudový rozsah
- :SENS:VOLT:RANG:AUTOn 0 – manuální napěťový rozsah
- :SENS:CURR:RANG:AUTOn 0 – manuální proudový rozsah

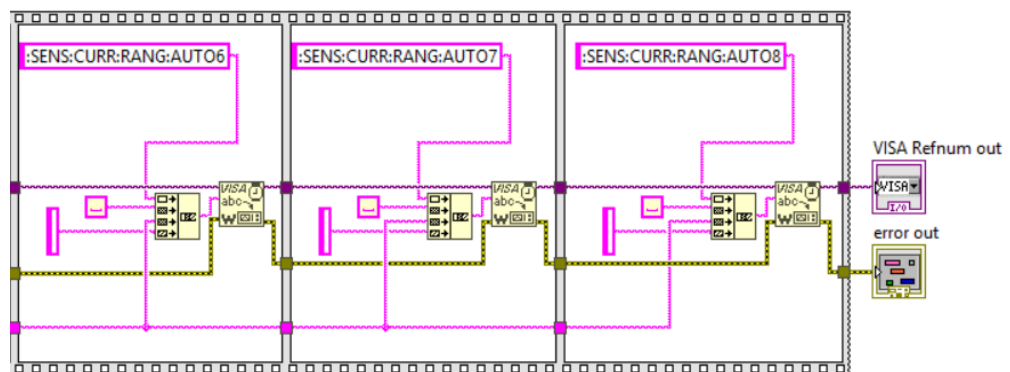


Obr. 4.7 Ikony SubVI pro automatický rozsah napětí (*Wua*) a proudu (*Wia*)





Obr. 4.8 Část SubVI pro nastavení napěťových rozsahů wattmetru na automatické



Obr. 4.9 Část SubVI pro nastavení proudových rozsahů wattmetru na automatické

- **Nastavení počtu hodnot proudu**

- Měření proudu může být variabilní na počet měřených rozsahů, proto byla přidána možnost nastavení počtu měřicích cyklů proudu. Měření napětí tento prvek neovlivňuje a napětí je vždy měřeno pro 20 hodnot napětí neboli 10 rozsahů.

- **Výběr souborů pro zápis výsledků měření**

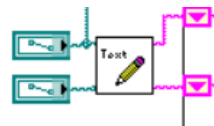
- Program v této části načítá cesty pro zápis dat z měření do Excelu. V případě, že cesty nejsou nastaveny, program měření vykoná, ale data z toho měření se nemají kam zapsat a pouze se zobrazí na čelním panelu.



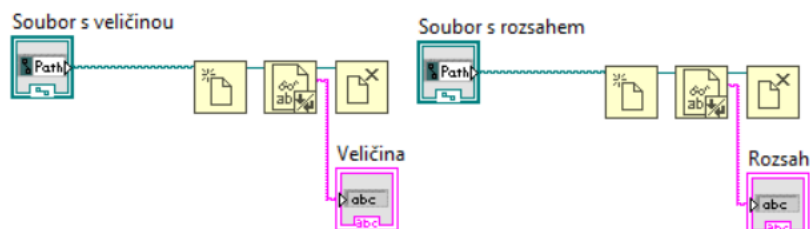
### 4.2.3 Spouštěcí smyčka For

Spouštěcí smyčka For vykoná vždy jen jeden cyklus, což odpovídá celému měřicímu cyklu. Smyčka obsahuje výběr měření, samotné měření a vyhodnocení. Po dokončení smyčky se program opět vrací do smyčky While.

- **Výběr měření a výběrový Case**
  - Struktura Case v sobě má nadefinované dva typy měření (Napětí x Proud), které se od sebe liší pouze v nastavení počtu měřících cyklů pro měřící For a komunikačními příkazy. Pro napětí je počet iterací pevně nastaven a pro proud je tento počet variabilní. Výběr konkrétního měření probíhá pomocí **Combo Boxu**, který je umístěn na čelním panelu. Combo Box musí obsahovat stejné výběry jako má výběrový Case možnosti volby.
- **Načtení souborů s daty pro veličiny a rozsahy**
  - Pro nastavení požadovaného rozsahu a veličiny na přístrojích se vytvářejí komunikační příkazy, které se skládají z instrukce a hodnoty. Hodnoty se načítají z textových souborů pomocí SubVI. Na vstup tohoto SubVI přivedeme adresu souboru. SubVI soubor otevře, načte hodnoty do textového 1D pole a soubor uzavře. Textové pole je vyvedeno na vstup měřícího For cyklu.



Obr. 4.10 Ikona SubVI pro načtení hodnot ze souboru



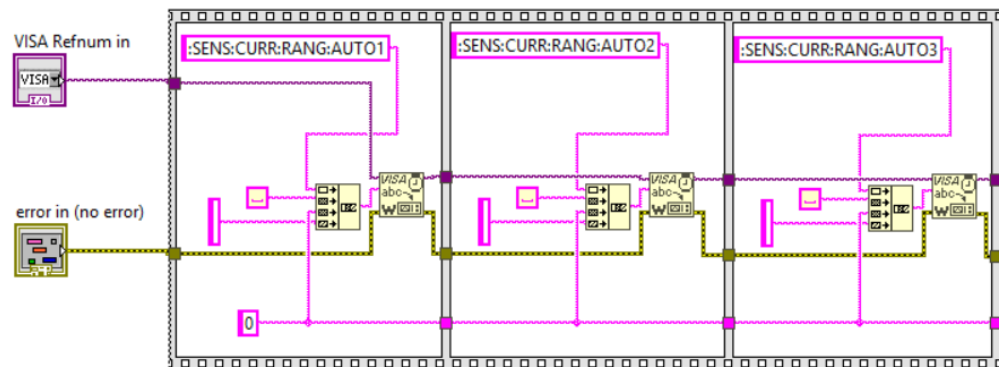
Obr. 4.11 Algoritmus SubVI pro načtení hodnot ze souboru

- **Nastavení wattmetru na manuál**

- SubVI nastaví rozsah wattmetru na manuální podle daného výběru. SubVI se liší od SubVI pro automatický rozsah pouze hodnotou na konci příkazu.



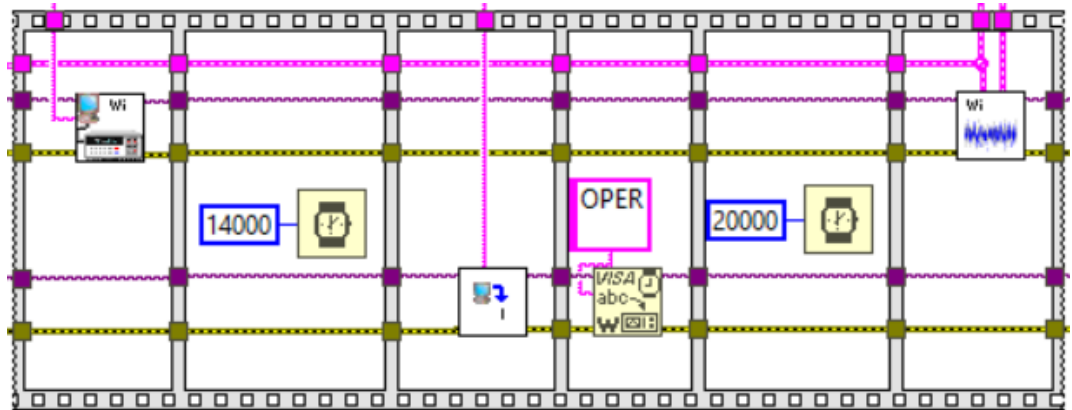
Obr. 4.12 Ikony SubVI pro manuální rozsah napětí (Wum) a proudu (Wim)



Obr. 4.13 Část SubVI pro nastavení napěťových rozsahů wattmetru na manuální

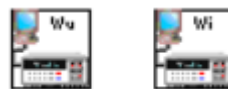
#### 4.2.4 Měřicí smyčka For

- **Shift registr**
  - *Shift registr* spolu se *Scan From String Function* slouží k rozložení 1D textového pole na jednotlivé prvky pole. *Shift registr* se každým cyklem posouvá o jednu pozici v 1D poli a *Scan From String Function* tento prvek načte a převede na požadovaný datový typ.
  
- **Flat sekvence**
  - Tato struktura slouží k určení přesného sledu měření, kde se jednotlivé rámce vykonávají postupně zleva doprava. Oba typy Flat sekvence jsou totožné, mají pouze jiná SubVI pro danou veličinu.

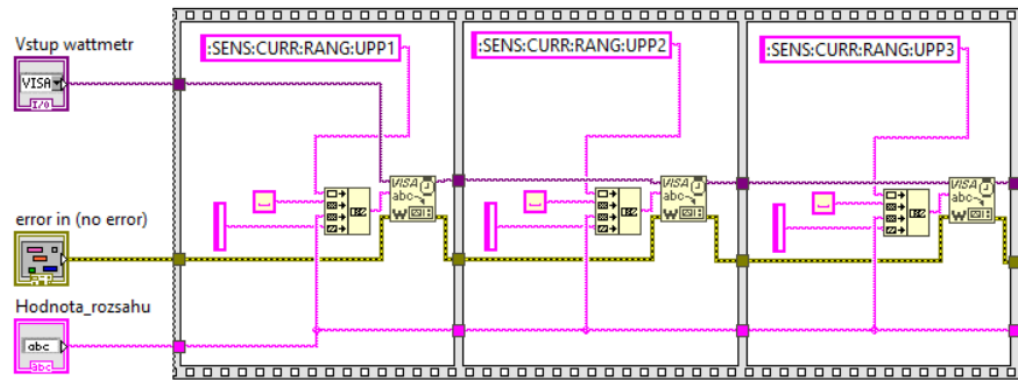


Obr. 4.14 Měřicí Flat struktura

- **První rámec** – Obsahuje SubVI pro nastavení měřicího rozsahu na wattmetru. Komunikační příkazy pro wattmetr jsou `:SENS:VOLT:RANG:UPPn x` pro napěťový rozsah a `:SENS:CURR:RANG:UPPn x` pro proudový rozsah. Příkaz je sestaven pomocí *Concatenate Strings Function* a následně je odeslán skrze *VISA Write Function*. Pro stavbu se využívají právě hodnoty z textového souboru, které byly upraveny *Shift registrem* a vkládají se v příkazu do pozice *x*.

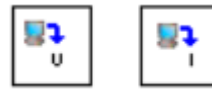


Obr. 4.15 Ikony SubVI pro zadávání hodnoty rozsahu napětí (Wu) a proudu (Wi)

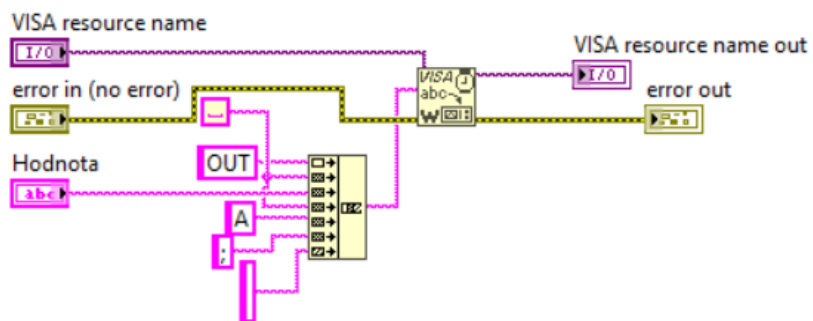


Obr. 4.16 Část SubVI pro nastavení konkrétní hodnoty proudového rozsahu wattmetru

- **Druhý rámeček** – Obsahuje funkci *Wait (ms) Function*, která má za úkol pozastavit program na 14 sekund, než se nastaví rozsahy na wattmetru.
- **Třetí rámeček** – V tomto rámečku dochází ke stavbě příkazů pro kalibrátor (stejně jako tomu bylo u wattmetru) a poté dochází ke komunikaci s kalibrátorem. Příkaz pro nastavení napětí je *OUT x V* a pro proud je *OUT x A*. Na pozici *x* přivádíme hodnotu veličiny z textového souboru.

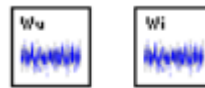


Obr. 4.17 Ikony SubVI pro nastavení napětí (U) a proudu (I) na kalibrátoru

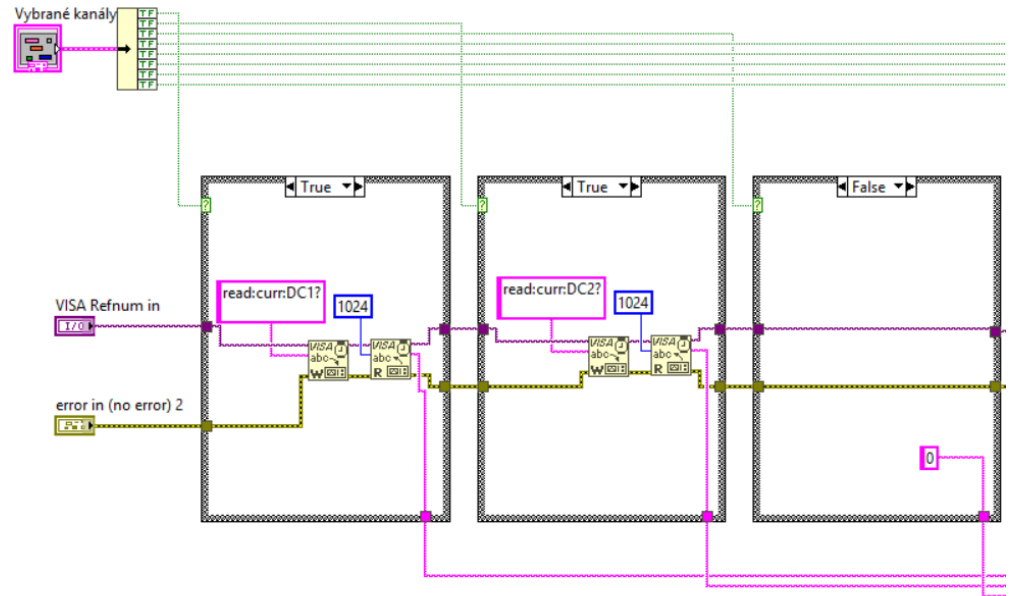


Obr. 4.18 SubVI stavby příkazu a odeslání kalibrátoru

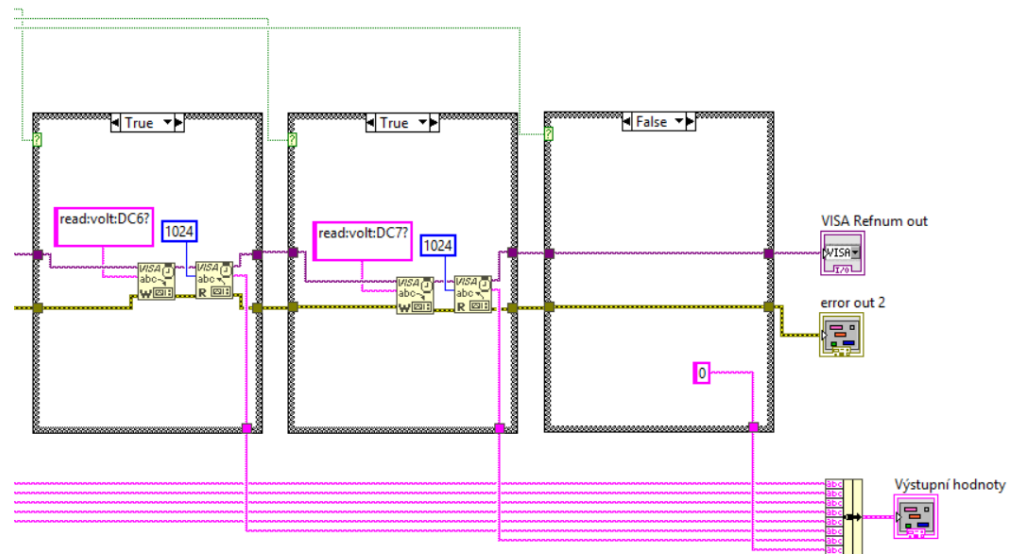
- **Čtvrtý rámeček** – Aktivuje výstupní svorky kalibrátoru instrukcí *OPER*.
- **Pátý rámeček** – Obsahuje funkci *Wait (ms) Function*, která má za úkol pozastavit program na 20 sekund, než se ustálí měřené hodnoty na wattmetru.
- **Šestý rámeček** – V posledním rámečku je umístěné SubVI pro načítání dat z wattmetru. SubVI obsahuje struktury Case určené pro načítání dat pouze z kanálů, které si uživatel vybral. V případě neměřeného kanálu vychází ze struktury Case defaultní nulová hodnota. Instrukce pro načítání dat jsou *read:volt:DCn?* pro napětí a *read:curr:DCn?* pro proud.



Obr. 4.19 Ikony SubVI pro načtení hodnot napětí ( $W_u$ ) a proudu ( $W_i$ ) z wattmetru



Obr. 4.20 Část SubVI pro načtení hodnot proudu z wattmetru



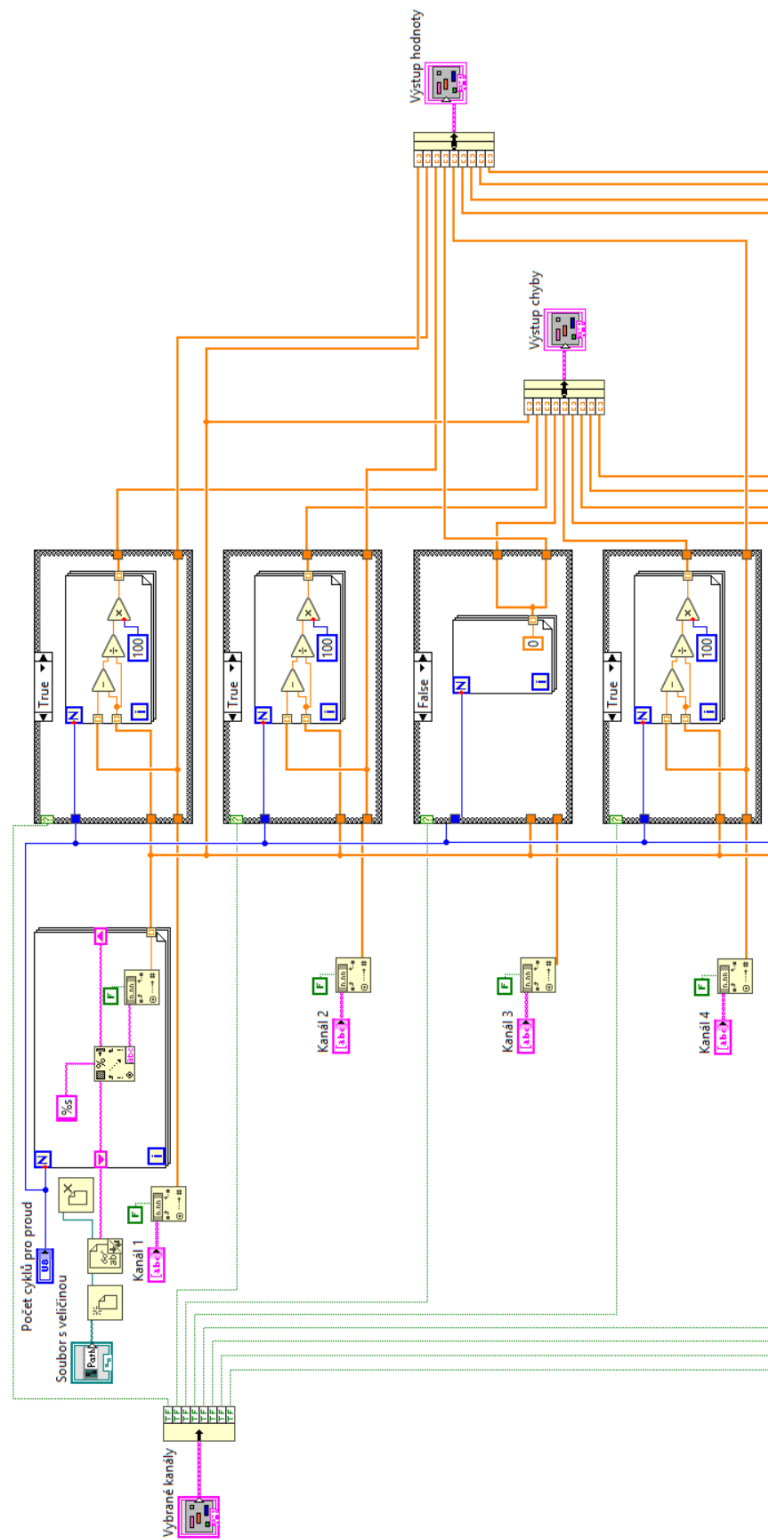
Obr. 4.21 Část SubVI pro načtení hodnot napětí z wattmetru

#### 4.2.5 Podprogramy pro vyhodnocení měření a zápis dat

- **Zpracování dat a výpočet chyb**
  - Tento podprogram nejprve převádí pomocí For cyklu hodnoty z textového souboru, které jsou potřeba pro výpočet. Pro nastavení počtu cyklů pro smyčky For se používá *Numeric Control*, ale pouze pro proud. V případě výpočtu chyby napětí je pro smyčky For konstantně nastaveno 20 cyklů. Dále se převádí skrze *Fract/Exp String To Number Function* přivedené 1D textové pole daného kanálu na 1D numerické pole. Lze vybrat, zda se čísla převedou na exponenciální tvar či na decimální tvar.
  - Struktura Case obsahuje výpočet chyby měření a průchod naměřených hodnot. Využitý vzorec pro výpočet naleznete v první kapitole Kalibrace. Pokud před začátkem měření byl vybrán daný kanál pro měření, vykoná se možnost struktury TRUE a vypočte se chyba měření. V případě možnosti FALSE budou na výstupu chyby i hodnoty nulové. Jednotlivé kanály jsou poté chronologicky srovnány do funkce *Bundle Function* a využity pro zápis do souborů a do tabulek na čelním panelu.



Obr. 4.22 Ikony SubVI pro vyhodnocení měření a zpracování dat

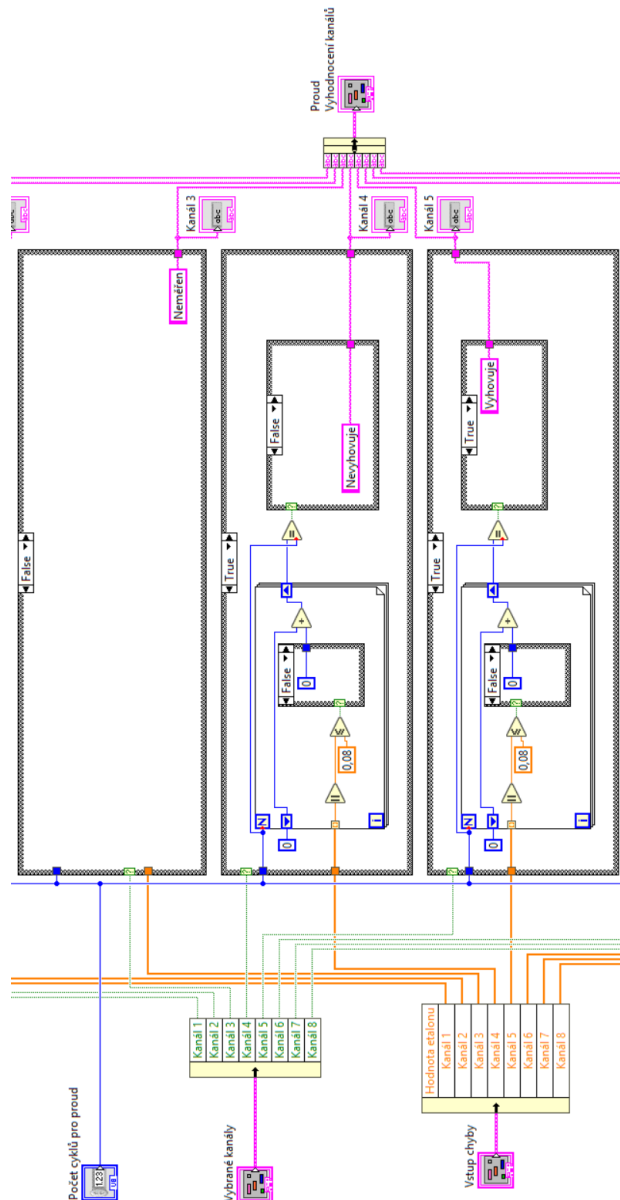


Obr. 4.23 Část SubVI pro zpracování dat a výpočet proudové chyby

- **Vyhodnocení kanálů**

- Podprogram k vyhodnocení používá vypočtené chyby a vybrané kanály k měření. Pro vyhodnocení je využita struktura Case, která má nastaveny dvě možnosti TRUE a FALSE podle toho, zda je kanál vybrán k měření. Při možnosti FALSE se na zobrazovači na čelním panelu zobrazí, že kanál nebyl měřen. Pro volbu TRUE se odchylka převede do absolutní hodnoty a porovná se podle přesnosti kalibrovaného wattmetru. Toto porovnávání se vykoná podle počtu měřených hodnot. Ve smyčce je čítač, který při každé iteraci se přičte „1“ či „0“, podle toho, zda vypočtená chyba je menší než 0,08%. Pokud se počet iterací rovná hodnotě na čítači, vnitřní struktura Case na výstupu pro zobrazovač na čelním panelu nastaví, že daný kanál vyhovuje. V opačném případě, že nevyhovuje.





Obr. 4.24 Část SubVI pro vyhodnocení proudových kanálů

- **Zápis hodnot**

- Podprogram nejprve rozbálí klastř a jednotlivá 1D pole jsou převedena na signály, které jsou následně sloučeny v jeden a zapsány do Excelu pomocí funkce *Write To Measurement File Express VI*.



## Závěr

V rámci diplomové práce byl vypracován program v programovacím jazyce LabVIEW, který je schopen po připojení přístrojů skrze čelní panel a nastavení cest pro soubory vstupních a výstupních dat samostatně změřit libovolné vstupní kanály wattmetru LMG500 pro stejnosměrné napětí a proud. Naměřená data je program schopen zpracovat a vyhodnotit, zda jednotlivé kanály vyhovují pro danou přesnost. Naměřená data a vypočtené chyby lze ukládat do Excelu. Uživatel do samotného algoritmu programu nezasahuje, ale může si v rámci textových souborů nastavit měřené rozsahy a hodnoty veličin.

Nejprve proběhlo seznámení se s programovacím prostředím, poté i seznámení se samotnými přístroji. Stavba kalibračního programu začala vytvořením podprogramu pro komunikaci s kalibrátorem. Podprogram již využíval pro tvorbu instrukcí data z textového souboru. Tento princip byl použit i pro stavbu podprogramu pro komunikaci s wattmetrem LMG500. Tyto podprogramy byly použity k stavbě Alfa verze měřicího programu a úspěšně otestovány. Program ještě neměl ukládání dat, manuální nastavení rozsahů wattmetru a vyhodnocení.

Po ověření komunikace s přístroji a otestování funkčnosti Alfa verze pro automatické měření přišel na řadu vývoj početních a ukládacích podprogramů. Podprogramy vypočítávaly chyby měření a ukládaly naměřená a vypočtená data do Excelu, ale pro všechny kanály. Tímto vznikla Beta verze programu. Beta verze stále neměla automatické vyhodnocení jednotlivých kanálů a manuální nastavení rozsahů.

V poslední verzi byla přidána možnost volit si měřené kanály. Na základě toho byl celý kalibrační program přestavěn a bylo přidáno manuální nastavení rozsahů wattmetru. Na závěr byl přidán podprogram pro automatické vyhodnocování jednotlivých kanálů, zda vyhovují třídě přesnosti. Při odladování finální verze programu byly vytvořeny instrukce pro uživatele. Při postupu podle instrukcí by se uživatel měl vyhnout všem problémům, které se při odladování vyskytly. Na závěr práce byly provedeny kontrolní kalibrace wattmetru finální verzi programu, během kterých se neobjevily žádné problémy s programem.

## Seznam literatury a informačních zdrojů

- [1] Kalibrace - Základní pojmy. Český metrologický institut [online]. [cit. 2019-04-26]. Dostupné z: <https://www.cmi.cz/node/537>
- [2] VILHELM, Ing. Vladimír. Kalibrační postup: Analogový střídavý wattmetr. Český metrologická společnost [PDF]. 2013, 10/2013 [cit. 2019-04-26]. Dostupné z: [http://spolky.csvts.cz/cms/sites/default/files/kp\\_4121113\\_analogovy\\_stridavy\\_watmetr.pdf](http://spolky.csvts.cz/cms/sites/default/files/kp_4121113_analogovy_stridavy_watmetr.pdf)
- [3] METODIKA PROVOZNIHO MĚŘENÍ NAPĚTÍ V PRŮMYSLOVÝCH APLIKACÍCH [PDF]. Praha: Česká metrologická společnost, 2017 [cit. 2019-05-09].
- [4] 5500A Multi-Product Calibrator [online]. [cit. 2019-04-22]. Dostupné z: [https://us.flukecal.com/products/obsolete-products/5500a-multi-product-calibrator?quicktabs\\_product\\_details=2](https://us.flukecal.com/products/obsolete-products/5500a-multi-product-calibrator?quicktabs_product_details=2)
- [5] 5500A Multi-Product Calibrator Operator Manual. Fluke [PDF]. USA: Fluke Corporation, 7/2006 [cit. 2019-04-22]. Dostupné z: [https://us.flukecal.com/products/obsolete-products/5500a-multi-product-calibrator?quicktabs\\_product\\_details=4](https://us.flukecal.com/products/obsolete-products/5500a-multi-product-calibrator?quicktabs_product_details=4)
- [6] Fluke [online]. [cit. 2019-04-22]. Dostupné z: <https://www.fluke.com/cs-cz>
- [7] ZES ZIMMER [online]. [cit. 2019-04-22]. Dostupné z: <https://www.zes.com/en>
- [8] ZES ZIMMER LMG500 [online]. [cit. 2019-04-22]. Dostupné z: <https://www.zes.com/en/Products/Discontinued-Products/Energy-and-Power-Meters/LMG500>
- [9] Multi Channel Power Meter LMG500: User manual [PDF]. ZES ZIMMER Electronic Systems, 2011, 2011/03/09 [cit. 2019-04-22]. Dostupné z: <https://www.zes.com/en/Products/Discontinued-Products/Energy-and-Power-Meters/LMG500>
- [10] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. Začínáme s LabVIEW. 1. vyd. Ilustrace Viktorie Vlachová. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [11] National Instruments Corporation, Manuál ke kurzu LabVIEW Core 1. Srpen 2011.
- [12] National Instruments Support [online]. [cit. 2019-04-22]. Dostupné z: <http://www.ni.com/cs-cz/support.html>
- [13] STANĚK, Jan. Návrh monitorovacího systému palivového článku PEM v LabVIEW. Plzeň, 2017. Bakalářská práce. ZČU. Vedoucí práce Ing. Pavel Štekl, Ph.D.