



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV ELEKTROENERGETIKY

DEPARTMENT OF ELECTRICAL POWER ENGINEERING

NÁVRH MĚŘÍCIHO PRACOVISTĚ V LABVIEW
PRO ÚČELY MĚŘENÍ SPEKTRA A
SVĚTELNÉHO TOKU

DESIGN OF MEASURING WORKSPACE IN LABVIEW FOR PURPOSE OF SPECTRUM
AND LUMINOUS FLUX MEASUREMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Sláma

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Škoda, Ph.D.

BRNO 2017

Diplomová práce

magisterský navazující studijní obor **Elektroenergetika**
Ústav elektroenergetiky

Student: Bc. Pavel Sláma

ID: 158229

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Návrh měřicího pracoviště v LabView pro účely měření spektra a světelného toku

POKYNY PRO VYPRACOVÁNÍ:

1. Důkladné seznámení s měřicími přístroji
2. Seznámení se s prostředím LabView
3. Vytvoření ovládacího softwaru pro měření spektra, světelného toku apod.

DOPORUČENÁ LITERATURA:

podle pokynů vedoucího práce

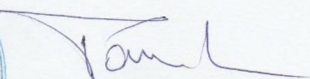
Termín zadání: 6. 2. 2017

Termín odevzdání: 22.5.2017

Vedoucí práce: Ing. Jan Škoda, Ph.D.

Konzultant:




doc. Ing. Petr Toman, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Práce se zabývá problematikou měření světelných veličin pomocí spektroradiometru a možností realizace takového měření pomocí programu LabView.

V první kapitole je rozebráno, jaké světelné veličiny lze měřit spektroradiometrem a jaký je jejich význam.

Druhá část obsahuje popis hardware vybavení, které bude použito k realizaci měření.

Třetí část popisuje samotný program LabView a také možnosti propojení s periferiemi.

Následuje část, ve které je rozebrán postup použitý při oživování použitých přístrojů.

Další část řeší, jak fungují obslužné programy stejnosměrného a střídavého zdroje.

V následující části se pak nachází popis struktury hlavního programu.

Poslední kapitola se potom zabývá popisem uživatelského prostředí.

Klíčová slova

LabView, Měření světelných veličin, Spektroradiometr, Komunikace LabView, Jeti Specboss 1211, výpočet teploty chromatičnosti, výpočet indexu podání barev

Abstract

This thesis deals with luminance parameters measurement and ways to accomplish this using LabView software.

The first part focuses on luminance parameters measurable by spectroradiometer and their meaning.

Following part introduces reader to hardware equipment that is used in the measurement.

Third part contains description of LabView software and explains what is required to make a communication between equipment and PC work.

Next part explains how the communication with peripherals was achieved.

Following up is the part where it is described how programs controlling AC and DC power supplies work.

In this part, the user interface is described.

Keywords

LabView, Luminance parameters measurement, Spectroradiometer, LabView communication, Jeti Specboss 1211, Correlated colour temperature calculation, Colour rendering index calculation

Bibliografická citace:

SLÁMA, P. *Návrh měřicího pracoviště v LabView pro účely měření spektra a světelného toku*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 76 s. Vedoucí diplomové práce Ing. Jan Škoda, Ph.D..

Prohlášení

„Prohlašuji, že svou závěrečnou práci na téma *Návrh měřicího pracoviště v LabView pro účely měření spektra a světelného toku* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **22. května 2017**

.....
podpis autora

Poděkování

Rád bych na tomto místě poděkoval svojí rodině a blízkým za podporu, kterou mě zahrnovali po celou dobu studia.

Dále bych chtěl poděkovat svému vedoucímu Ing. Janu Škodovi, Ph.D. za pedagogickou a odbornou pomoc při zpracování mé diplomové práce

V Brně dne **22. května 2017**

.....
podpis autora

Obsah

1 Úvod.....	1
1.1 Cíle práce	1
2 Měření vlastností zdrojů světla.....	2
2.1 Základní vlastnosti světla.....	2
2.1.1 Světelný tok.....	3
2.1.2 Barevné parametry	4
3 Měřicí přístroje a pomůcky.....	8
3.1 Spektroradiometr	8
3.1.1 Základní princip	8
3.2 Kulový integrátor	10
3.3 Laboratorní zdroje.....	11
4 Program LabView	12
4.1 Programování v LabVIEW.....	12
4.2 Struktury	14
4.2.1 Cyklus While	15
4.2.2 Struktura Event	15
4.2.3 Struktura Case.....	15
4.3 Komunikace HW a SW	15
4.3.1 Rozhraní GPIB	16
4.3.2 USB a LAN	16
4.4 Ovladače.....	17
5 Navazování komunikace mezi PC a přístroji.....	19
5.1 Stejnoseměrný zdroj	19
5.2 Střídavý zdroj	20
5.3 Spektroradiometr	20
6 Tvorba programu v labview	21
6.1 Program pro ovládání AC zdroje.....	22
6.2 Program pro ovládání DC zdroje.....	25
6.3 Společné náležitosti programů pro AC i DC zdroj	28
7 Hlavní program	30
7.1 Nastavení výchozích hodnot programu	30
7.2 Zapnutí programu pro ovládání AC zdroje.....	31
7.3 Zapnutí programu pro ovládání DC zdroje	32
7.4 Inicializace spektrometru a stav připojení.....	32
7.5 Načítání externích souborů.....	32
7.6 Kontrola připojení zdrojů	33
7.7 Zavření okna	33

7.8 Struktura event	34
8 Uživatelské prostředí programu.....	55
8.1 Čelní strana programu	55
8.2 Strana se změřenými daty	57
9 Závěr	61
Literatura	63
Seznam symbolů, veličin a zkratk.....	65
Seznam příloh.....	66

Seznam obrázků

Obrázek 1 Princip fotoelektrického článku [3].....	2
Obrázek 2 Zobrazení zářivého spektra [14]	3
Obrázek 3 Poměrná spektrální citlivost lidského oka [13].....	3
Obrázek 4 CIE 1931 - Průběhy kolorimetrických koeficientů [1]	5
Obrázek 5 Princip klasického spektrometru [4]	8
Obrázek 6 Spektrometr JETI Specboss 1211 [7]	9
Obrázek 7 Lambertův zářič [6]	10
Obrázek 8 Schéma uspořádání kulového integrátoru [1]	10
Obrázek 9 Zdroj APT 310 XAC [9]	11
Obrázek 10 Zdroj Elektroautomatik PSI 5210 A [12]	11
Obrázek 11 Čelní panel virtuálního měřicího přístroje.....	12
Obrázek 12 Zdrojový kód programu	13
Obrázek 13 Datové typy	14
Obrázek 14 Automatické přetypování	14
Obrázek 15 Konektor GPIB [15].....	16
Obrázek 16 Výběr VI pro ovládání přístrojů	17
Obrázek 17 Testovací program zdroje Elektroautomatik.....	19
Obrázek 18 Testovací program pro zdroj APT	20
Obrázek 19 Testovací program spektrometru.....	20
Obrázek 20 Nastavení hodnoty vstupu v jiném než momentálně aktivním VI.....	21
Obrázek 21 Nastavení hodnoty v programu EXE skrze rozhraní ActiveX	21
Obrázek 22 Náhled na program pro ovládání střídavého zdroje	23
Obrázek 23 Zapnutí zdroje	24
Obrázek 24 Vypnutí zdroje.....	24
Obrázek 25 Podmínka splněna	24
Obrázek 26 Podmínka nesplněna	24
Obrázek 27 Levá polovina programu DC zdroje	25
Obrázek 28 Podmínka True bodu 4	26
Obrázek 29 Pravá polovina programu DC zdroje	26
Obrázek 30 Podmínka False bodu 3	27
Obrázek 31 Stránka struktury case pro zapnutí zdroje	27
Obrázek 32 Stav true	28
Obrázek 33 Naznačení struktury hlavního programu	30
Obrázek 34 Otevření programu pro ovládání AC zdroje	31
Obrázek 35 Inicializace spektrometru	32
Obrázek 36 Kontrola připojení zdrojů	33
Obrázek 37 Struktura event – okno timeout.....	35

Obrázek 38	Struktura event – okno Zapnout AC	37
Obrázek 39	Struktura event – okno Vypnout AC	37
Obrázek 40	Struktura event – funkční část okna DC blok	38
Obrázek 41	Struktura event – okno Measurement	39
Obrázek 42	Nastavení integračního času	40
Obrázek 43	Volba časového měření	41
Obrázek 44	Programové přepnutí struktury event	41
Obrázek 45	Struktura event – okno save	42
Obrázek 46	Zápis do pevného sloupce	43
Obrázek 47	Struktura event – okno prumerne (nevyužitý prostor byl oříznut)	44
Obrázek 48	Struktura event – okno del_zazn (nevyužitý prostor byl oříznut)	46
Obrázek 49	Struktura event – Data k zobrazení	47
Obrázek 50	Struktura event – Seznam změřených zdrojů	48
Obrázek 51	Struktura event – vymaz_data	49
Obrázek 52	Struktura event – proved_korekci	50
Obrázek 53	Struktura event – výřez z okna save_data	52
Obrázek 54	Struktura event – výřez okna nacist data	53
Obrázek 55	Struktura event - výřez okna Open Button	54
Obrázek 56	Přední panel programu	55
Obrázek 57	Strana se změřenými daty	57
Obrázek 58	Trichromatický trojúhelník xy (obrázek na pozadí převzat a upraven z programu JETI LiVal)	58
Obrázek 59	Časová závislost zvolené veličiny	58
Obrázek 60	Nastavení parametrů pro provedení korekce	59

1 ÚVOD

Světelné zdroje představují nedílnou součást našeho každodenního života. Téměř na každém kroku narazíme na osvětlení, které dokáže člověku prodloužit a zpříjemnit dobu produktivity či zábavy. Ačkoli si to laik třeba ani neuvědomuje, celý proces osvětlování neznámá jenom návštěvu obchodu a koupi žárovky či moderního LED svítidla. Za vývojem každého takového zdroje leží řada vývojářů a vědců, kteří věnují svoje úsilí vytvoření světelného zdroje, který bude pro danou aplikaci co nejvhodnější. A protože i ten nejlepší vývojář, nebo naopak obzvláště on potřebuje ujištění, že jeho návrh odpovídá realitě, je nezbytné vždy každý světelný zdroj ozkoušet a změřit.

Pryč jsou však doby, kdy bylo třeba trávit čas nad fotometrickými lavicemi k získání relevantních výsledků. Dnes je vše řízeno digitálně a tento trend zasáhl i měřicí techniku. Máme k dispozici spínané elektronické zdroje s PFC, které se dokáží samy regulovat na potřebnou veličinu. Jsou k dispozici spektrometry, které lze schovat do kapsy, přičemž nijak výrazně neutrpěla jejich přesnost.

Byla by však škoda se zastavovat na tomto stupni, když laboratorní přístroje disponují rozhraními pro připojení k počítači, na kterém můžeme vše vidět elegantně na jednom místě.

Myšlenka propojení počítače s laboratorní technikou není úplně nová. Jenom National Instruments, jejichž produktem je LabView na tomto poli pracují přes dvacet let. Vždy však neplatilo že počítače jsou tak snadno dostupné, nebo že mají vhodná rozhraní pro spojení s přístroji. Vybavení počítače rozhraním, na kterém tato komunikace začínala, tedy GPIB dodnes stojí deseti-, nebo i statisíce. Od doby, kdy většina výrobců přešla na USB či LAN se tedy otevřely dveře pro spojení s laboratorní technikou přece jenom širší veřejnosti.

1.1 Cíle práce

Krom toho, že má práce za cíl splnění zadání, je zde několik dalších ambicí. Stávající aparatura využívající spektrometru Jeti Specboss 1211 v laboratoři světelné techniky využívá manuálního ovládání zdrojů stejnosměrného i střídavého napětí a obslužný program Jeti LiVal. Práce má za cíl vytvoření programu, ze kterého bude možné ovládat jak oba laboratorní zdroje, tak i spektrometr. Kromě samotného měření a výpočtu světelných veličin pak má program obsahovat i paměť, kde bude možné pracovat s více měřeními po sobě a funkce pro ukládání dat.

Co je však ale asi nejdůležitější, program má snížit nejistotu měření tím, že budou uživateli dostupná k zobrazení i korekční data a možnost z nich počítat průměr, což stávající program neumožňuje. Dále má program obsahovat funkci načítání dat, aby při opakovaných měřeních nebylo nutné měřit znova to, co je již změřené.

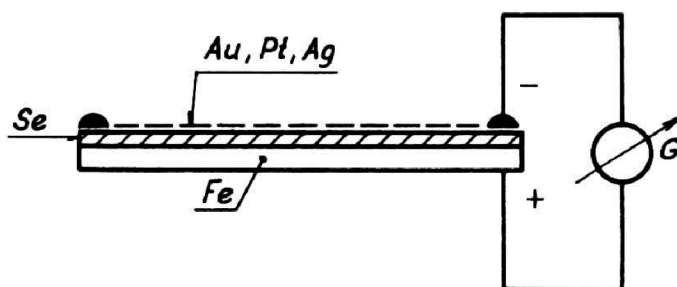
2 MĚŘENÍ VLASTNOSTÍ ZDROJŮ SVĚTLA

Světelné zdroje mají řadu parametrů, jejichž určení je nezbytné pro správný návrh osvětlení. Pro ověření, zda daný zdroj splňuje udávané parametry je nezbytné tyto parametry umět měřit.

Na základě principu měření světelných veličin rozdělujeme metody na vizuální a fyzikální.

Vizuální metody, nazývané také subjektivní, využívají při měření lidského zraku. To už z principu znamená, že se při měření takovými metodami dopouštíme chyby, která závisí na individuálních zrakových možnostech osoby, která měření provádí. Navíc jsou nezbytné rozměrné přístroje a drahé světelné normály, přičemž samotné měření bývá poměrně zdouhavé. Z těchto důvodů se již subjektivní metody pro měření zpravidla nepoužívají [1].

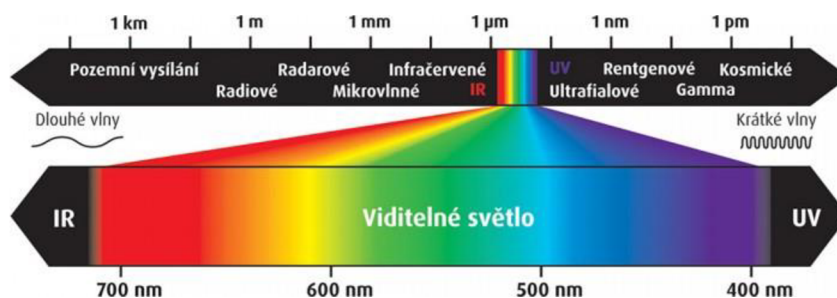
Fyzikální, popř. objektivní metody nahradily vizuální s nástupem elektroniky. Základem těchto metod jsou fotoelektrické články. To jsou součástky založené na fotoelektrickém jevu. Ten znamená, že pokud je vystavena záření vrstva polovodiče, která je nanášena na vrstvě kovu, vznikne mezi těmito dvěma vrstvami rozdíl potenciálů [3].



Obrázek 1 Princip fotoelektrického článku [3]

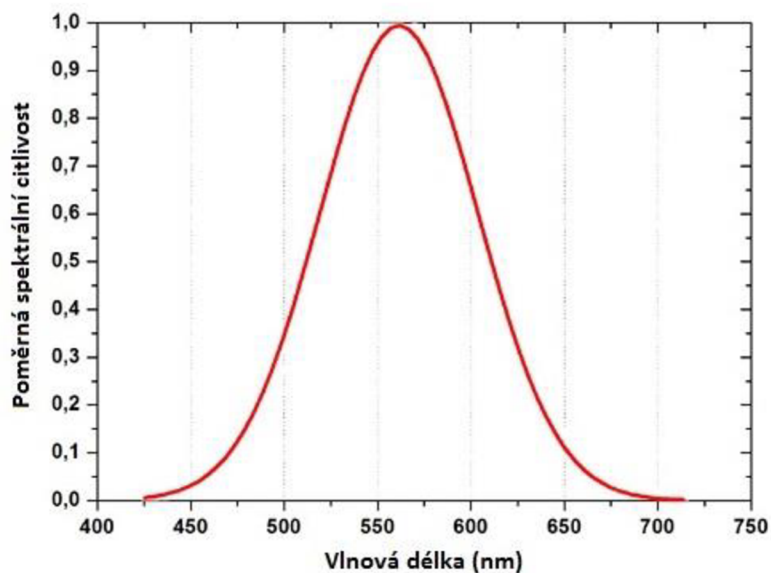
2.1 Základní vlastnosti světla

Vzato ze široka, světlo je vlastně úzká část spektra elektromagnetického záření. Je definováno jako záření, které je schopno vyvolat zrakový vjem v lidském oku. To má poté při denním vidění, za pomoci zrakových receptorů, tzv. čípků, schopnost budit vjem barvy. Spektrum viditelného záření se u lidského oka pohybuje zhruba mezi 360–830 nm. Tento rozsah je značně volný, neboť jde o subjektivní parametr. Nejčastěji, hlavně pak při výpočtech, je udáván rozsah 380-780 nm.



Obrázek 2 Zobrazení zářivého spektra [14]

Zároveň pak lidské oko nemá stejnou citlivost na všech vlnových délkách. Aby bylo možné objektivně porovnávat vlastnosti světelných zdrojů, byl komisí CIE vydán barevný prostor CIE 1931 XYZ. Ten mimo jiné obsahuje křivku tzv. $V(\lambda)$. Ta nám říká, jaká je citlivost lidského oka pro jednotlivé vlnové délky. Na základě definice kandely, jednotky SI, má maximální hodnotu 683 lm/W, která se nachází na 555 nm [1].



Obrázek 3 Poměrná spektrální citlivost lidského oka [13]

2.1.1 Světelný tok

Tato veličina je popisem zářivé energie, kterou dokáže naše oko interpretovat jako zrakový vjem. Jde tedy o světelnou analogii výkonu a u světelných zdrojů označuje celkovou světelnou energii, která ze zdroje vychází. Tato veličina respektuje spektrální citlivost lidského oka.

Značkou světelného toku je Φ a jeho jednotkou je 1 lm (Lumen) [1].
U monochromatického záření se světelný tok určí ze vztahu:

$$\Phi(\lambda) = K(\lambda) \cdot \Phi_e(\lambda) = K_m \cdot V(\lambda) \cdot \Phi_e(\lambda) \quad (2.1)$$

kde: $K(\lambda)$ je světelný účinek monochromatického záření ($\text{lm}\cdot\text{W}^{-1}$)

$\Phi_e(\lambda)$ je zářivý tok (W)

K_m je maximální hodnota veličiny $K(\lambda)$ ($\text{lm}\cdot\text{W}^{-1}$)

$V(\lambda)$ je poměrná světelná účinnost monochromatického záření (-)

Světelný tok složený z více monochromatických záření lze vypočítat:

$$\Phi = K_m \cdot \int_0^{\infty} \left(\frac{d\Phi_e(\lambda)}{d\lambda} \right) V(\lambda) \cdot d\lambda \quad (2.2)$$

K_m má pro fotopické vidění hodnotu $683 \text{ lm}\cdot\text{W}^{-1}$ [1].

2.1.2 Barevné parametry

Pojem barva lze použít nejen pro popis předmětů, ale i světla. Při popisu barvy předmětů se užívá pojmu kolorita a u barvy světla platí pojem chromatičnost. Protože se tato práce zabývá měřením světelných parametrů, budeme dále rozebírat pouze pojem chromatičnost [1].

Chromatičnost světla a tím i barevné vlastnosti světelného zdroje lze vyjádřit řadou způsobů, které mohou být uváděny zvlášť, nebo se mohou doplňovat. Takovým případem mohou být obaly běžně prodávaných světelných zdrojů, na kterých obvykle nalezneme údaj o tom, jakou má produkt teplotu chromatičnosti a index podání barev. Následují tedy způsoby, jakými lze chromatičnost popsat.

- **Trichromatické soustavy**

Z názvu vyplývá, že popisem barev jsou u nich zapotřebí tři číselné údaje. Ty vyjadřují, v jakém poměru by musely být spolu smíchány tři barevné podněty, aby vytvořily libovolnou barvu.

RGB

Jde o nejstarší popis barevného podnětu kolorimetrickou soustavou. Vychází z reálného použití tří monochromatických světelných zdrojů o červené, zelené a modré barvě k míchání výsledné barvy. Výhodou tohoto prostoru bylo, že se srovnáním měřeného světelného zdroje a směsí RGB zdrojů namíchaných na stejnou barvu okamžitě dalo určit, jaké světelné souřadnice daný zdroj má. Nevýhodou je, že kolorimetrické koeficienty tohoto prostoru mohou nabývat záporných hodnot [1].

Výpočet trichromatických souřadnic rgb ze soustavy xyz probíhá následujícím způsobem:

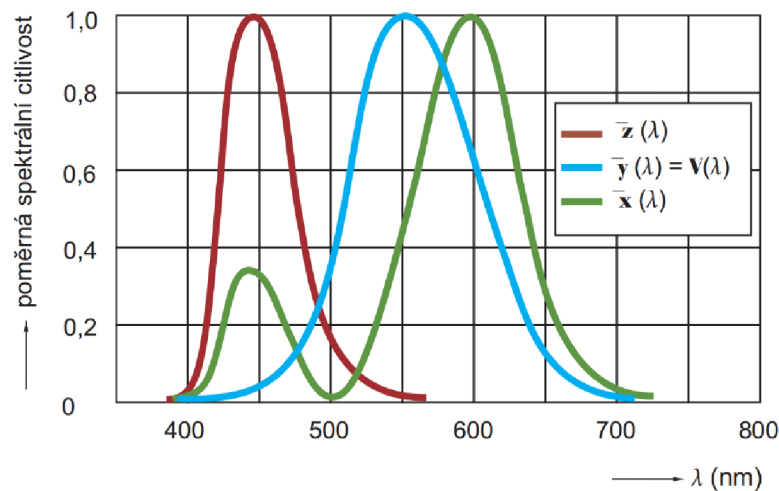
$$r = 2,5623 \cdot x - 1,1661 \cdot y - 0,3962 \cdot z$$

$$g = -1,0215 \cdot x + 1,9778 \cdot y + 0,0437 \cdot z \quad (2.3)$$

$$b = 0,0752 \cdot x - 0,2562 \cdot y + 1,1810 \cdot z$$

XYZ

Jako vylepšení RGB prostoru byl představen prostor XYZ. Ten vychází ze svého předchůdce, je však transformován tak, aby hodnoty jeho kolorimetrických koeficientů nabývaly pouze kladných hodnot [1].



Obrázek 4 CIE 1931 - Průběhy kolorimetrických koeficientů [1]

Výpočet trichromatických složek XYZ a trichromatických souřadnic xyz lze provést následujícím způsobem:

$$\begin{aligned}
 X &= 683 \cdot \int_0^{\infty} \varphi_{e\lambda}(\lambda) \cdot \bar{x}(\lambda) d\lambda & x &= \frac{X}{X + Y + Z} \\
 Y &= 683 \cdot \int_0^{\infty} \varphi_{e\lambda}(\lambda) \cdot \bar{y}(\lambda) d\lambda & y &= \frac{Y}{X + Y + Z} \\
 Z &= 683 \cdot \int_0^{\infty} \varphi_{e\lambda}(\lambda) \cdot \bar{z}(\lambda) d\lambda & z &= \frac{Z}{X + Y + Z}
 \end{aligned} \tag{2.4}$$

kde: $\varphi_{e\lambda}(\lambda)$ je spektrální hustota zářivého toku ($W \cdot m^{-2}$)
 $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ jsou kolorimetrické koeficienty (-)
 λ je vlnová délka (nm)

- **Teplota chromatičnosti**

Dalším způsobem, kterým lze popsat barevné vlastnosti světla je teplota chromatičnosti. Jedná se o teplotu tepelného zářiče, při které je jeho chromatičnost stejná jako chromatičnost zkoumaného zdroje. Při překročení hranice 5000 K se místo teploty chromatičnosti používá náhradní teplota chromatičnosti. Ta se porovnává se smluvním světlem D, namísto tepelného zářiče. Značí se T_c , nebo v anglicky mluvících zemích CCT její jednotkou je K (Kelvin) [1].

Výpočet teploty chromatičnosti lze provést více způsoby. Pokud se však chceme dopátrat přesného výsledku, nelze použít aproximační metody. Mezi přesnými numerickými metodami je asi nejpoužívanější Robertsonova metoda. Ta využívá toho, že máme k dispozici tabulku s trichromatickými souřadnicemi pro jednotlivé průřezky izoterm na Planckově křivce, přičemž pro každou izotermu je dána i směrnice. Na základě našich vypočítaných souřadnic najdeme v tabulce ty, mezi nimiž naše souřadnice leží a lineární interpolací potom dopočítáme naši výslednou teplotu chromatičnosti [15].

- **Index podání barev**

Při návrhu osvětlovacích soustav je nezbytné pamatovat na to, aby barvy pod osvětlením působily přirozeně. U teplotních zářičů, tj. žárovek nešlo o příliš složitou problematiku, s nástupem moderních zdrojů jako výbojek či LED však začalo docházet k problémům. Mají totiž od žárovky značně odlišné spektrum.

Index podání barev vyjadřuje, do jaké míry se vjem barvy objektů osvětlených daným zdrojem shoduje s objekty osvětlenými referenčním zdrojem.

Značkou této veličiny je R_a a nabývá hodnoty 0-100. Pokud je teplota chromatičnosti zdroje do 5000 K, odpovídá R_a 100 černému zářiči, při překročení 5000 K odpovídá smluvnímu světlu D. Aby vyšlo R_a 0, musí být použit zdroj monochromatického světla.

V obytných prostorách by mělo platit $R_a > 80$ (ČSN EN 12464-1) [1].

Výpočet indexu podání barev začíná tím, že najdeme trichromatické souřadnice zkoumaného zdroje. Dalším předpokladem je zjištění teploty chromatičnosti, např. dle postupu uvedeného výše. Podle velikosti teploty chromatičnosti zvolíme jako referenční zdroj absolutně černé těleso, nebo referenční zářič typu D. Do 5000 K počítáme s absolutně černým tělesem, při vyšších teplotách pak se zářičem D. Je potřeba vypočítat spektrální průběh příslušného referenčního zdroje. Tím potom ozáříme 15 vzorků barev. Ozáření lze provést matematicky tak, že roznásobíme intenzitu testovacích vzorků a referenčního zářiče pro všechny vlnové délky. To bychom mohli zapsat následujícím způsobem:

$$I_{výsl}(\lambda) = I_{ref}(\lambda) \cdot I_{test}(\lambda) \quad (2.5)$$

Obdobně provedeme ozáření i u zkoumaného zdroje.

Následně spočítáme souřadnice všech výsledných spektrálních průběhů v prostoru CIE 1964.

U testovaného zdroje provedeme Von Kriesovu transformaci [16], která nám pomůže přiblížit se vnímání barev lidským okem.

Výsledky převedeme do soustavy CIE $U^*W^*V^*$ a podle následujícího vzorce spočítáme vzdálenost souřadnic zkoumaného zdroje a referenčního zdroje pro jednotlivé vzorky:

$$\Delta E_i = \sqrt{(\Delta U^*)^2 + (\Delta V^*)^2 + (\Delta W^*)^2} \quad (2.6)$$

Speciální index podání barev R_i se potom spočítá pomocí vzorce:

$$R_i = 100 - 4,6 \cdot \Delta E_i \quad (2.7)$$

Index podání barev potom jednoduše spočítáme jako aritmetický průměr speciálních indexů podání barev pro prvních 8 vzorků [18].

3 MĚŘÍCÍ PŘÍSTROJE A POMŮCKY

Jak již bylo zmíněno výše, v dnešní době se již ve světelně technických měřeních nevyužívá vizuální fotometrie, ale pouze fotometrie fyzikální. Úkolem práce jsou měření světelného spektra a dalších veličin. Aby se měření dalo provést jedním přístrojem, je nezbytné použití spektrometru, který dokáže velmi přesně změřit spektrum, přičemž ostatní potřebné veličiny je možné zjistit výpočtem.

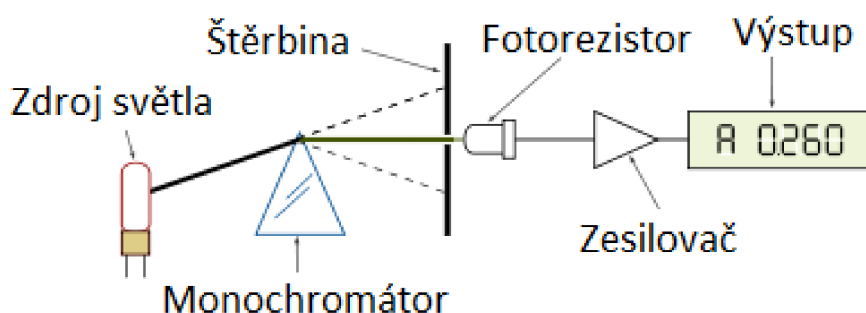
3.1 Spektrometr

Je to zařízení určené k měření spektrálního rozložení zářivého výkonu. Na základě měřeného spektra je možné dále vypočítat radiometrické, fotometrické, popřípadě kolorimetrické veličiny. Z přístrojů používaných k určení barevných parametrů světla jsou tím nejpřesnějšími.

Je ještě záhodno zmínit, že spektrometr je zařízení pro měření záření o vlnové délce mimo oblast viditelného světla (i když ji může také obsahovat). Jejich podskupinou jsou tedy spektrofotometry, které slouží k měření v oblasti viditelného světla, popř. s určitým přesahem do infračervené (VIS-IR) či ultrafialové oblasti (UV-VIS) [1].

3.1.1 Základní princip

Klasické spektrometry se skládají ze vstupní štěrbiny, otočného rozptylového hranolu, popř. mřížky a výstupní štěrbiny, po které následuje jeden detektor. To vlastně znamená, že ze světla jsou postupně separovány jednotlivé vlnové délky a vzniklé monochromatické záření je přivedeno na detektor [2].



Obrázek 5 Princip klasického spektrometru [4]

Výhodou tohoto principu je vysoká citlivost a přesnost. Nevýhod je však hned několik. Tato zařízení jsou poměrně rozměrná, měření trvá dlouho a zařízení obsahuje pohyblivé části, čímž je náchylnější k poškození.

Proto je klasická konstrukce v dnešní době nahrazována spektrometrií se sensorovým polem. To znamená, že nemusí být měřena jedna vlnová délka po druhé, ale jsou měřeny všechny zároveň. Toto řešení je časově úspornější, neobsahuje mechanicky pohyblivé části a umožňuje konstrukci kompaktních zařízení.

Také se již dnes nepoužívají optické hranoly, ale rozptylové mřížky. Důvodem je nižší cena a zároveň lepší rozptylové vlastnosti [2].

Spektrometrie se sensorovým polem, jak název napovídá, obsahuje větší množství sensorických prvků seřazených za sebou. Jejich množství udává, jaké bude mít přístroj rozlišení. Jeho vlastnosti jsou potom dány tím, jaká technologie je použita. Existují tři možnosti [2]:

- CCD pole – vysoká citlivost, nízký dynamický rozsah, vhodné pro měření nižších světelných úrovní
- Pole fotodiód – rychlé měření, široký rozsah vlnových délek
- Pole CMOS sensorů – široký dynamický rozsah, univerzální použití

Spektrometr JETI Specboss 1211

Jde o zařízení schopné měřit zářivé spektrum mezi 350 – 1000 nm. To odpovídá viditelnému světlu a části infračerveného záření. Rozlišení zařízení ve spektrální oblasti je 1 nm. Napájení je realizováno USB portem, který zároveň slouží ke komunikaci s počítačem.

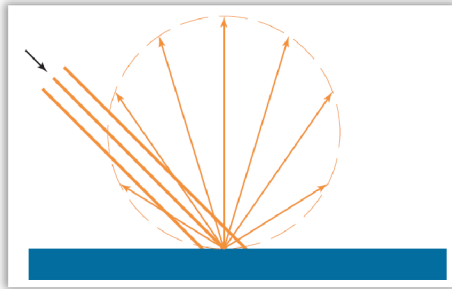
Rozložení světla na jednotlivé vlnové délky je realizováno disperzní mřížkou a detekce je prováděna polem CCD sensorů o rozlišení 2048 pixelů [7].



Obrázek 6 Spektrometr JETI Specboss 1211 [7]

3.2 Kulový integrátor

Světelný tok je veličinou celkové světelné energie vyzářené zdrojem. Abychom tuto veličinu nemuseli složitě počítat z křivek svítivosti, používá se k jejímu měření právě kulového integrátoru. Může jít o kouli různých rozměrů od několika centimetrů po několik metrů, princip je ovšem vždy stejný. Vnitřek koule musí být natřen speciálním nátěrem s maximální možnou hodnotou indexu odrazivosti a zároveň se musí chovat jako ideální Lambertův zářič [1].

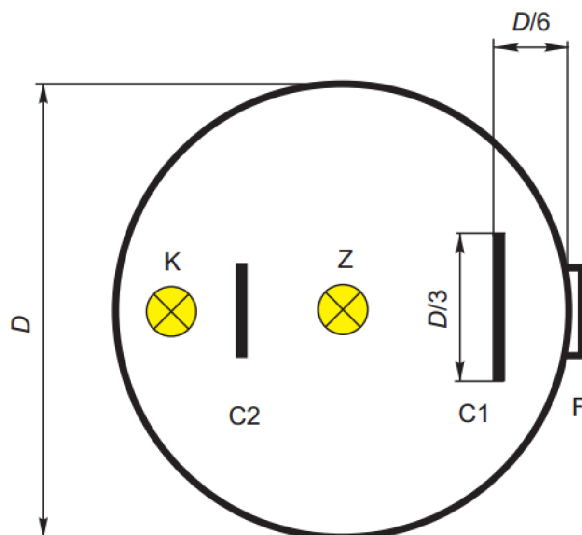


Obrázek 7 Lambertův zářič [6]

Lambertovým zářičem se nazývá takový povrch, na kterém dochází k difuznímu odrazu. To znamená, že paprsek světla se po dopadu na povrch rozdělí ve všech směrech stejně neboli že tato plocha bude mít ve všech směrech stejný jas. Princip Lambertova zářiče je dobře patrný na obrázku 7 [6].

Kromě výše zmíněných světelných vlastností může být potřeba aby kulový integrátor splňoval některé další. Musí být dostatečně mechanicky odolný, neboť se do integrátoru může prášit, takže může být nutné ho vyčistit. Dále musí být teplotně stálý, protože některé velmi výkonné světelné zdroje dosti hřejí. Nutnost těchto dvou parametrů už však závisí na zvolené aplikaci [11].

Pro povrch kulových integrátorů určené pro měření v oblasti viditelného světla se používá Síran barnatý (BaSO_4), Oxid hořečnatý (MgO) nebo ještě například Polytetrafluorethylen neboli PTFE [10].



Z – měřený světelný zdroj

K – korekční světelný zdroj

C1, C2 – clonky

D – průměr kulového integrátoru

F – fotočlánek

Obrázek 8 Schéma uspořádání kulového integrátoru [1]

Principem kulového integrátoru je, že světlo se musí několikanásobně odrazit před tím, než vstoupí do fotočlánku. Aby to bylo dodrženo, je nezbytné umístění clonky před fotočlánek.

Může se stát, že měřený zdroj má větší plochy, konstrukční nebo jiné, které světlo pohlcují. Pro takové případy je kulový integrátor vybaven korekčním zdrojem za druhou clonkou, který je schopen tento vliv kompenzovat. Zároveň je díky tomuto zdroji možné měřit účinnost svítidel [1].

3.3 Laboratorní zdroje

Pro měření v kulovém integrátoru se používají dva druhy zdrojů. Stejnosemřný a střídavý. Střídavý zdroj se používá pro napájení měřeného světelného zdroje, k napájení korekční žárovky se používá zdroj stejnosměrný.

Zdroj APT 310 XAC

Výstupní parametry pro AC režim:

Napětový rozsah: 5–300 V
Maximální výkon: 1 kVA
Výstupní frekvence: 40-1000 Hz



Obrázek 9 Zdroj APT 310 XAC [9]

Jde o modulární střídavý laboratorní zdroj připravený pro externí ovládání. V základní konfiguraci nabízí rozhraní USB, lze ale doobjednat i rozhraní LAN, či GPIB. Výhodou zdroje je, že nabízí měření výstupu s dobrou přesností, a proto není pro obvyklé použití nutné připojovat dodatečné měřicí přístroje. Zařízení lze použít i v režimu stejnosměrného zdroje včetně regulace na konstantní velikost proudu [9].

Zdroj Elektro automatik PSI 5210 A

Výstupní parametry:

Napětový rozsah 0-200 V
Proudový rozsah 0-10 A
Maximální výstupní výkon 640 W



Obrázek 10 Zdroj Elektroautomatik PSI 5210 A [12]

Jedná se o programovatelný stejnosměrný laboratorní zdroj s možností dálkového ovládání přes rozhraní USB nebo Ethernet. Kromě ovladačů pro LabVIEW je dostupný i hotový software pro ovládání přes PC [12].

4 PROGRAM LABVIEW

Název tohoto programu vychází z anglické zkratky Laboratory Virtual Instrument Engineering Workbench. Program vznikl v roce 1983 ve firmě National Instruments za účelem propojení měřících přístrojů s PC. Tvůrci vycházeli z myšlenky, že technik uvyklý na používání blokových diagramů bude schopen podobným procesem vytvořit počítačový program.

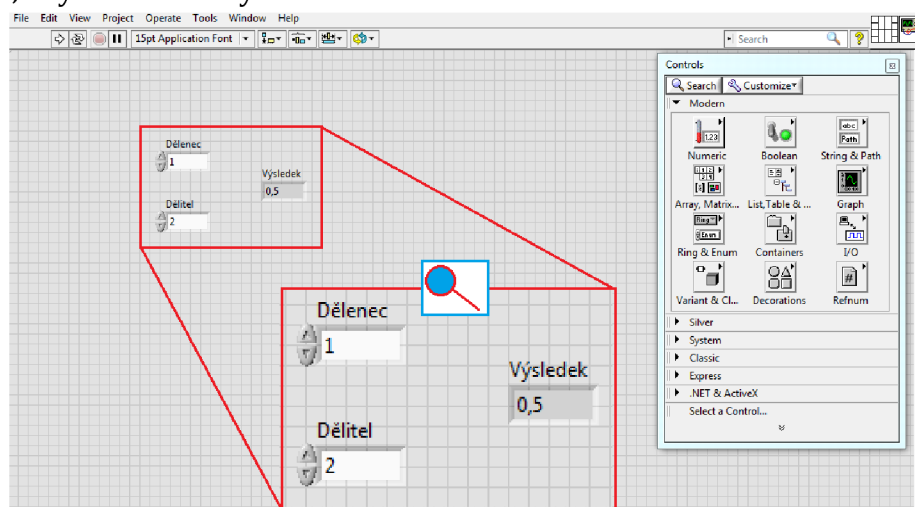
LabVIEW je tedy grafický programovací jazyk, využívající místo textových příkazů, vykonávajících se řádek po řádku, bloky funkcí propojených datovým tokem. To znamená, že každý blok vždy čeká na všechny vstupní hodnoty a teprve po jejich obdržení vykoná svou funkci. Tyto bloky se nazývají VI, což znamená virtual instrument, neboli virtuální přístroj. Mohou zastávat jednoduché matematické či logické funkce, nebo může jít o celé komplexní programy umožňující různé analýzy, zápis do souboru a podobně. Program je v prostředí LabVIEW kompilován a lze z něj vytvořit EXE soubor spustitelný na libovolném počítači [8].

4.1 Programování v LabVIEW

Program se skládá ze dvou obrazovek. První z nich je čelní panel, kde se nachází ovládací a indikační prvky. Díky nim můžeme programu zadávat informace, ovládat ho, či odtud odečítat hodnoty. Mohou to být různá tlačítka, indikátory, přepínače, posuvníky, grafy, ale i textová a číselná pole.

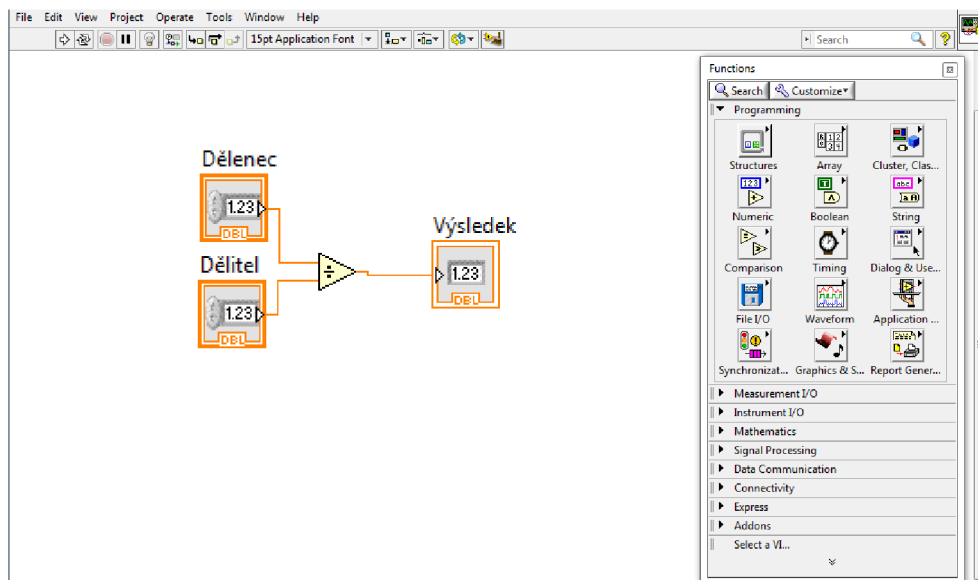
Čelní panel je vlastně obrazovkou virtuálního měřícího přístroje. Na obrázku 11 vidíme programové rozhraní. Skládá se z plochy, na kterou se umísťují prvky vybrané z nabídky po pravé straně, popřípadě lze vyvolat obdobnou nabídku stisknutím pravého tlačítka myši.

Program se spouští pomocí vrchní nástrojové lišty s ikonami, kde lze zvolit i různé režimy chodu programu včetně krokování. Program lze také pozastavit či nastavit, aby běžel ve smyčce.



Obrázek 11 Čelní panel virtuálního měřícího přístroje

Druhá obrazovka pak obsahuje to, co se odehrává na pozadí. Jde vlastně o zdrojový kód samotného programu, kam odkazuje každý prvek umístěný na čelním panelu. Na příkladu z obrázku 12 jsou vstupní bloky připojeny na blok ze záložky *Numeric* s operací dělení. Výstup z tohoto bloku je pak předán zpět na čelní panel.



Obrázek 12 Zdrojový kód programu

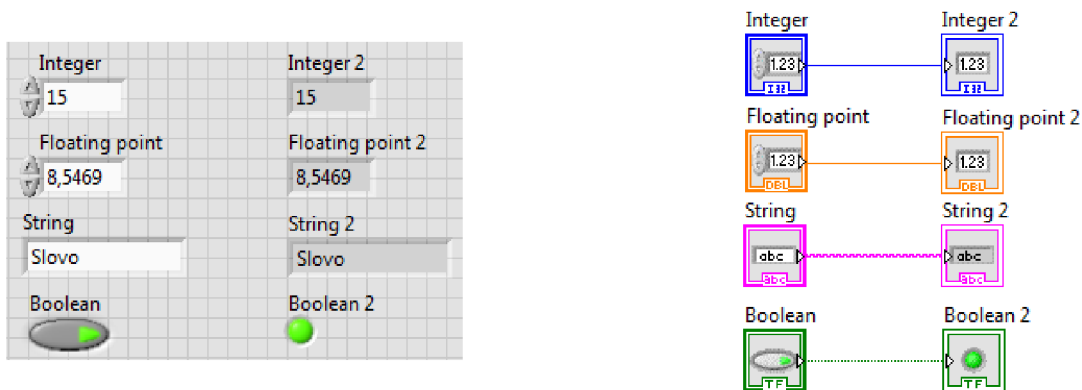
Ve zdrojovém kódu se nachází dva hlavní druhy bloků. Tím prvním jsou funkční bloky. Mají černé ohraničení a vykonávají nějakou činnost na základě podnětu od vstupního datového toku. Tato činnost je vykonána, jakmile je na všechny nezbytné vstupy dodán signál. V příkladu je zástupcem funkčních bloků ten trojúhelníkový se znaménkem dělení.

Druhým druhem bloku je datový blok. Ten provádí vstup nebo výstup dat. Podle toho, o jaká data se jedná má potom barevné ohraničení. V programu na obrázku 12 je jeho reprezentací *Dělenec*, *Dělitel* a *Výsledek*. V tomto případě jsou všechny vstupy a výstupy vyvedeny na čelní panel, ale být tomu tak vždy nemusí.

Stejně jako mají datové bloky různé barevné ohraničení, má různé barvy i datový tok, kterým jsou bloky propojované. Propojení může mít také různou tloušťku či typ čáry.

Velkou výhodou LabVIEW je skutečnost, že často dokáže provést automaticky přetypování, pokud je například u vstupu vyžadován jeden typ datové proměnné, ale přivedeme tam jiný, podobného typu. Například pokud je na vstupu vyžadována proměnná typu *Floating point* ale přivedený datový tok *Integer*. Přetypování pak neprovede v nesmyslných případech, jako by byla například konverze *String* na *Integer*.

Jak vypadají základní datové typy v LabVIEW je uvedeno na obrázku 13:



Obrázek 13 Datové typy

Jak probíhá výše zmíněné přetypování jde pak vidět na obrázku 14, kde jde *Integer* a *Floating point* normálně sečíst, ale při sčítání *String* a *Floating point* je zobrazena chyba a program není možné kompilovat.



Obrázek 14 Automatické přetypování

V případě, že pracujeme s větším množstvím proměnných, lze je sloučit do tzv. Clusteru. To znamená, že proměnné sdílejí datový tok a v případě, že některou chceme číst nebo ji přepsat, vložíme do datového toku clusteru „odbočku“ ve formě příslušného prvku. Tím může být *Bundle* a *Unbundle*, čili *Sdružit* a *Oddělit*.

4.2 Struktury

Velice důležitou součástí programu Labview jsou tzv. struktury. Ty nám umožňují řídit různým způsobem běh programu. Nahrazují tak některé funkce dostupné při klasickém textovém programování, jako jsou cykly *for*, *case*, *while* a další.

4.2.1 Cyklus *While*

Jde v podstatě o nejdůležitější a dost možná i nejpoužívanější strukturu v Labview. Bez ní by totiž náš program proběhl jednou a potom se sám ukončil. Tato struktura tedy udržuje chod programu ve smyčce. Tu můžeme ukončit až posláním signálu na příslušný terminál, např. formou tlačítka ukončit.

4.2.2 Struktura *Event*

Jako další si popíšeme strukturu *event*. Tu je totiž vhodné kombinovat se strukturou *while* pro vytvoření uživatelského rozhraní. Struktura *event* nám umožňuje vytvářet několik oken, do kterých se umisťují programové bloky, přičemž jednotlivá okna jsou vyvolána, pokud se odehraje akce jim přiřazená, tedy např. stisk nějakého tlačítka. Výhodou této struktury je to, že umožňuje při nepotřebě vykonávání činnosti na pozadí zůstat úplně netečná, čímž minimálně zatěžuje procesor počítače. Pro potřeby vykonávání procesů na pozadí pak také obsahuje okno *timeout*, u kterého si můžeme nastavit, po jakém čase nečinnosti se má vykonat.

K této struktuře je třeba připojit ještě jednu poznámku, a to, že nepodporuje chod tlačítka s automatickým vymáčknutím a aby tlačítko nezůstalo v zapnutém stavu je třeba ho pokaždé vypnout. To se provede jednoduše tak že na místní proměnnou daného tlačítka pošleme konstantu *false*.

4.2.3 Struktura *Case*

Podobně jako struktura *event*, i *case* má za úkol vykonat nějakou činnost na základě jiného vstupního impulsu. není však většinou vhodná pro rozhodování o běhu celého programu, neboť má proti struktuře *event* omezené možnosti, ať už co se týče vstupů, tak i řízení běhu programu.

4.3 Komunikace HW a SW

Základním kamenem komunikace LabVIEW s externími přístroji je rozhraní VISA, neboli Virtual Instrument Software Architecture. To bylo vytvořeno uskupením řady výrobců přístrojů, jako National Instruments, Tektronix, Racal a dalšími. Cílem bylo zajištění vzájemné kompatibility různých zařízení napříč výrobci. Tento standard tedy není omezen pouze pro použití v LabVIEW, ale lze ho implementovat i například v programovacích jazycích C, C++ či Visual Basic. VISA slouží výrobcům přístrojů k programování ovladačů pro svá zařízení ve standardním prostředí, které používá stejné operace a datové typy nezávisle na použitém fyzickém rozhraní s přístrojem.

At' už tedy jde o propojení rozhraním USB, LAN či GPIB, propojení s programem LabVIEW zajišťuje VISA. Propojení přístroje s počítačem lze zkontrolovat v nástroji NI-MAX, neboli Measurement and Automation Explorer. Dokáže totiž zobrazit všechna zařízení s připojením na VISA. Těmto zařízením je také přiřazena jistá adresa, kterou lze zde taktéž zobrazit [6].

4.3.1 Rozhraní GPIB

Vytvoření tohoto rozhraní v podstatě znamenalo impulz k vytvoření programu LabVIEW, jelikož představilo standard pro komunikaci mezi měřicími a zkušebními přístroji a počítačem.

Název tohoto rozhraní je zkratkou pocházející z názvu General Purpose Interface BUS. Dalšími názvy jsou také IEEE 488 nebo HP-IB. Rozhraní vyvinula v roce 1972 firma Hewlett-Packard a postupně se stalo nejpoužívanějším pro propojení měřících přístrojů a PC.



Obrázek 15 Konektor GPIB [15]

Ačkoli je toto rozhraní v dnešní době stále nejpoužívanější, bylo již technicky překonáno a většina moderních zařízení používá jiný druh připojení. Dnešní počítače zpravidla tuto sběrnici neobsahují, čímž přibývá nutnost použití převodníku, a také je rozhraní v porovnání s moderními alternativami pomalé se svojí rychlostí 1 MB/s (pozn. existuje modernizovaná HS488 verze s maximální rychlostí 8 MB/s). [7]

4.3.2 USB a LAN

V dnešní době jsou do měřících přístrojů zpravidla nasazována tato rozhraní. Krom toho, že jsou výrazně rychlejší než GPIB, je jejich hlavní výhodou přítomnost na drtivé většině počítačů. To dovoluje používat k měření běžné PC místo drahých specializovaných zařízení.

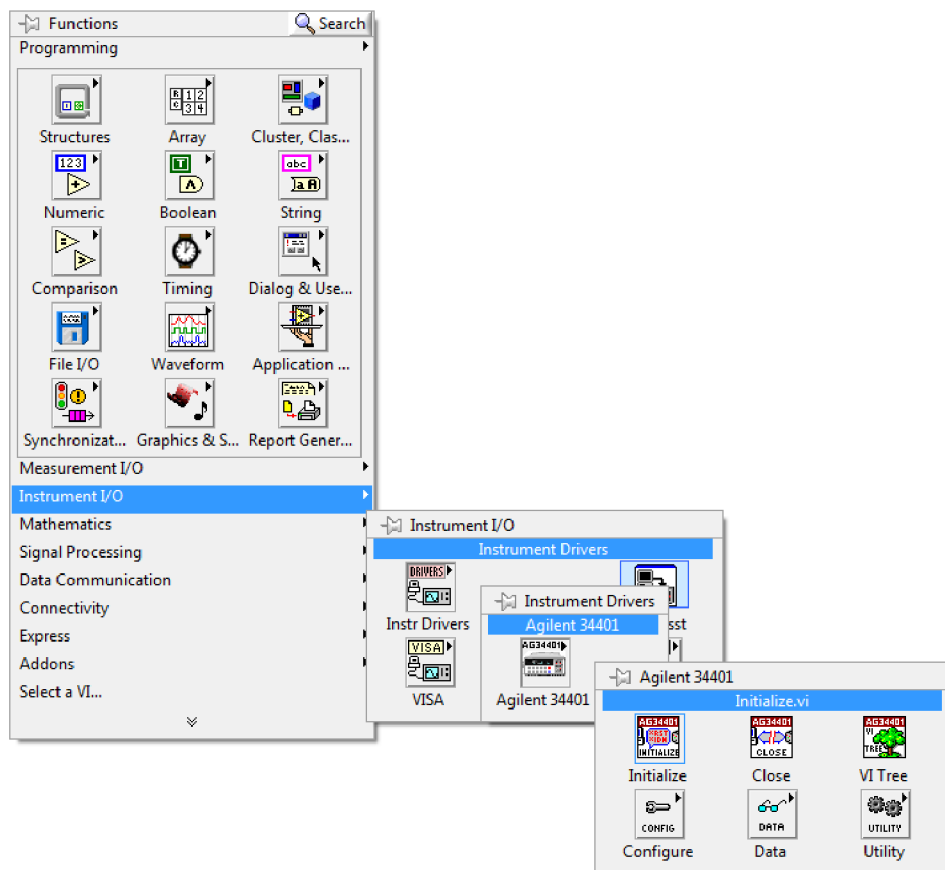
4.4 Ovladače

Pokud chceme připojit k počítači s programem LabView nějaký přístroj, je zpravidla nutné nejdříve nainstalovat příslušné ovladače. Ty bývají buď dodávané přímo s přístrojem na CD, popř. flash disku, nebo je lze stáhnout ze stránek výrobce. Řada výrobců také umísťuje ovladače přímo na stránky National Instruments, kde je přístupná jejich databáze i s vyhledáváním, ovšem pouze pro registrované uživatele.

Ať už získáme ovladače odkudkoli, jejich provedení je vždy stejné. Mají formu objektů VI, tedy tradičních programů prostředí LabVIEW. Jejich instalace probíhá vložením těchto programů do příslušného adresáře v průzkumníku souborů. Při standardní instalaci je jeho umístění následující:

`C:\Program Files\National Instruments\LabVIEW\instr.lib`

Když jsou ovladače vloženy sem, lze už pak k nim přistupovat jako ke každému jinému bloku a přidat je do programu. To může vypadat jako na obrázku 16:



Obrázek 16 Výběr VI pro ovládání přístrojů

Obvyklá struktura základního programu pro ovládání přístrojů je následující:

- Inicializační VI – jeho úkolem je projít rozhraní počítače a najít připojené přístroje, jejichž základní údaje dále pošle přes sběrnici VISA
- Čtecí/zapisovací VI – pomocí nich provádíme nastavení a čtení z připojených přístrojů, například nastavení napětí na laboratorním zdroji
- Ukončení přenosu – tento blok má za úkol ukončit spojení mezi PC a přístrojem

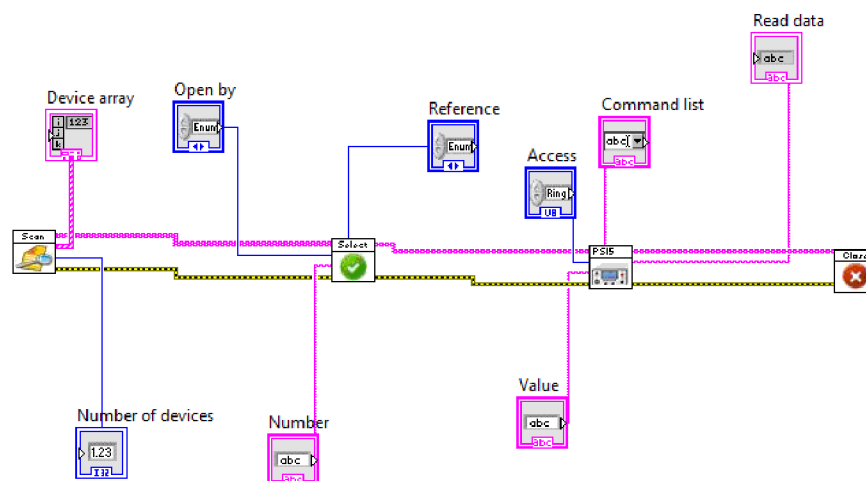
5 NAVAZOVÁNÍ KOMUNIKACE MEZI PC A PŘÍSTROJI

Z dodávaných médií, popř. z internetových stránek výrobce byly staženy ovladače a nainstalovány dle výše uvedeného postupu.

Dalším krokem bylo pro jednotlivé přístroje vytvořit programy pro komunikaci. V tomto kroku nebyl kladen důraz ani tolik na to, aby programy měly nějaké funkce, ale šlo spíše o to, aby byl počítač schopný identifikovat přístroje a vyměnit si s nimi základní data. Tento postup byl zvolen proto, aby v případě problémů či úplného selhání základní komunikace byl čas problémy řešit.

5.1 Stejnoseměrný zdroj

Ovladače pro tento přístroj jsou dostupné na webu výrobce. Obsahují spíše menší množství bloků, veškeré čtení, popř. zápis je prováděno pomocí jednoho bloku, kterému určíme, zda má číst, nebo zapisovat, zadáme mu, jakého parametru se operace týká a jestliže jde o zápis tak zadáme i příslušnou zapisovanou hodnotu.



Obrázek 17 Testovací program zdroje Elektroautomatik

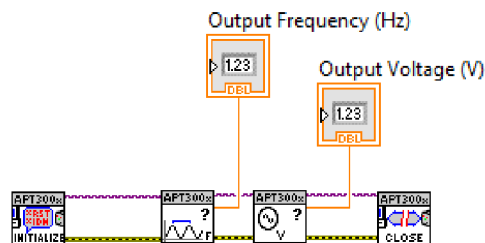
U zdroje Elektroautomatik PSI 5210A byl vytvořen program na obrázku 17. Prvním funkčním blokem je zde *Scan*, který slouží k prohledání vstupů počítače. Další blok *Select* slouží k výběru zařízení, které chceme ovládat. Za ním se nachází hlavní blok. Tomu pomocí vstupu *Access* určujeme, jestli má číst či zapisovat, pomocí

Command list vybíráme činnost a vstupem *Value* zadáváme hodnoty. Následně je komunikace uzavřena blokem *Close*.

5.2 Střídavý zdroj

Pro zdroj ATP 310 XAC jsou ovladače dostupné na internetových stránkách výrobce.

Ovladače obsahují značně větší množství funkčních bloků než zdroj Elektroautomatik. Každý parametr má zde vlastní blok pro čtení/zápis. Jinak je struktura programu podobná.

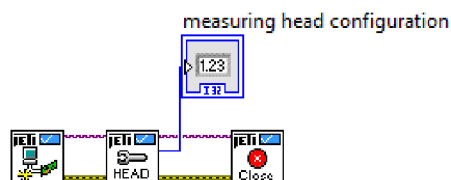


Obrázek 18 Testovací program pro zdroj APT

Komunikace je zahájena pomocí bloku *Initialize*. Ten slouží k ohledání vstupů počítače a zároveň i k výběru zařízení. Následují funkční bloky. Zde byly jako příklad použity bloky, které mají za cíl zjistit výstupní hodnotu frekvence a napětí. Program je zakončený blokem *Close* pro ukončení komunikace.

5.3 Spektrometr

Ovladače k tomuto přístroji byly výrobcem dodány na CD. Obdobně jako u zdroje APT je zde větší množství funkčních bloků. Je zde klasicky inicializační blok pro začátek komunikace a blok *Close* pro její ukončení. Jako příklad je zde uveden program pro čtení konfigurace měřící hlavy.

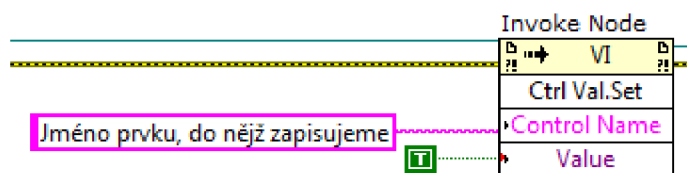


Obrázek 19 Testovací program spektrometru

Uvedené programy jsou, jak již bylo řečeno velmi jednoduché, neboť je jejich cílem pouze ověřit funkčnost komunikace mezi počítačem a přístroji. Nemají za úkol samotnou realizaci nějakého měření.

6 TVORBA PROGRAMU V LABVIEW

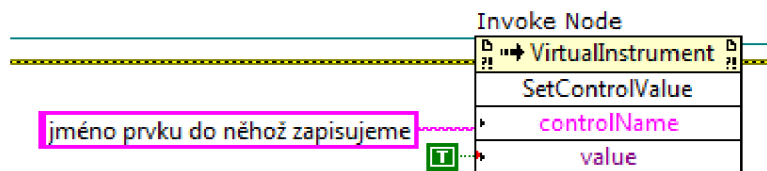
S vedoucím práce bylo domluveno, že program bude vytvořen ve formátu *.exe*, tzn. klasický program v prostředí Windows. To znamená vytvořit si v Labview projekt, přičemž soubory *.vi*, jenž jsou jeho součástí, bude potom obsahovat i výsledný *.exe* soubor. Zde ovšem nastal problém, jelikož při umístění bloků ovladačů zdroje APT a Elektroautomatik do jednoho *.vi* či do jednoho projektu docházelo k odpojení jednoho či druhého zdroje a chybě komunikace. Experimentálně bylo zjištěno, že jediný způsob, jak provozovat oba zdroje současně bez úprav ovladačů je jejich umístění do samostatných *.vi*. To ovšem také znamenalo nezbytnost komunikace mezi dílčími programy. Labview umožňuje vzájemnou komunikaci mezi běžícími *.vi* tak, že si načte referenci *.vi*, na který mu udáme adresu. Potom lze buďto číst, nebo zapisovat do proměnných.



Obrázek 20 Nastavení hodnoty vstupu v jiném než momentálně aktivním VI

Aby bylo možné program provozovat i na počítači, který neobsahuje licenci Labview, bylo rozhodnuto, že řešení pomocí *.vi* nestačí a je třeba vytvoření samostatných programů *.exe*, které spolu budou schopné komunikovat. Odesílání či přijímání dat lze provést řadou způsobů. Existují zde sdílené proměnné či komunikace přes síťové protokoly. Pro svou rychlost odezvy bylo však zvoleno rozhraní ActiveX. To je produktem Microsoftu a slouží přímo pro sdílení dat mezi programy. Využívají ho například kancelářské programy Microsoft Office pro sdílení dat s programy třetích stran.

Využití ActiveX bylo také zvoleno proto, že funguje na podobném principu jako výše zmíněná výměna dat mezi *.vi*. Programy *.exe* vytvořené v Labview mají totiž specifikum, že de facto obsahují *.vi*, ze kterého byly vytvořeny.



Obrázek 21 Nastavení hodnoty v programu EXE skrze rozhraní ActiveX

Rozhraní ActiveX má vždy dvě části. Jedná se o serverovou část, tedy zdroj dat a klientskou část, která data přijímá, popřípadě zapisuje.

6.1 Program pro ovládání AC zdroje

Jak již bylo zmíněno výše, z důvodů omezení ovladačů dodaných výrobcem byla zvolena koncepce třech samostatných programů. Jako první zde bude popsán program pro ovládání AC zdroje. V hierarchii ActiveX tento program pracuje jako server. Je tedy sám odpovědný za ovládání periferie, tzn. AC zdroje a klientský program už do něj jen zapisuje požadované hodnoty, nebo je případně čte.

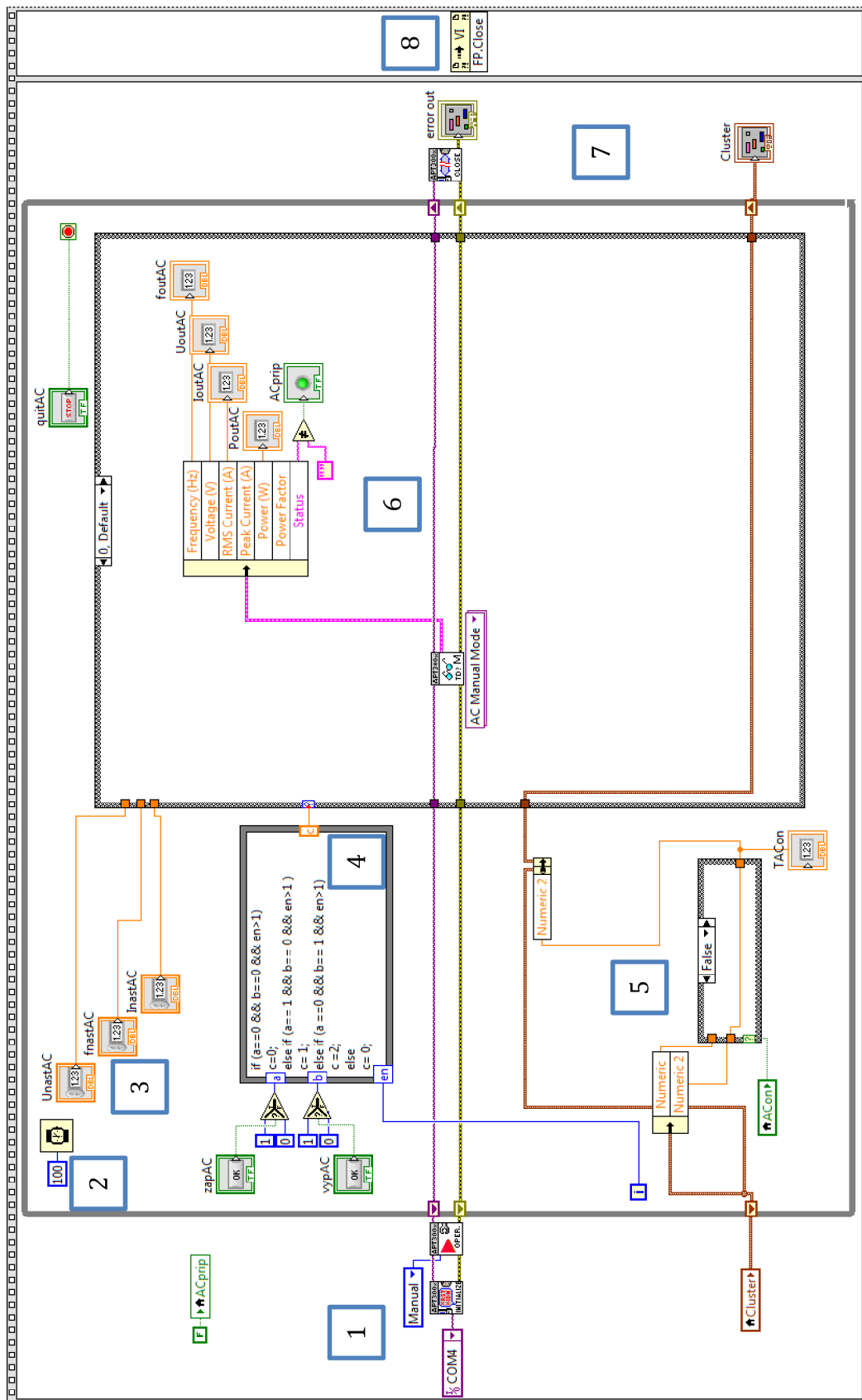
Tento program má poněkud netypickou základní strukturu. Obvykle je řešením takového programu použití struktury *event*, která se stará o vyhodnocování událostí. Protože však tento program není určen pro přímý kontakt s uživatelem, ale je serverem ActiveX, bylo přikročeno ke struktuře *case*. U struktury *event* je totiž problém se spouštěním jejich jednotlivých událostí na základě programového impulzu místo impulzu uživatelského. Ačkoli i tento problém je možné řešit, v tomto případě by o bylo zbytečně komplikované. Tím, že uživatel nepřijde s tímto programem do přímého kontaktu navíc odpadá nutnost řešit blokaci vstupů, aby nenastala neošetřená událost.

Na obrázku 22 vidíme, jak vypadá celý program. Celý se nachází v sekvenční struktuře, která zajišťuje, že bude dodržena zvolená posloupnost úkolů. Zde je zvolena, protože na konci se nachází blok *FP.Close*, sloužící k zavření okna s programem po jeho ukončení. Jde tedy o to, aby toto nebylo vykonáno příliš brzy.

Zde jsou v bodech popsány jednotlivé části programu na obrázku 22:

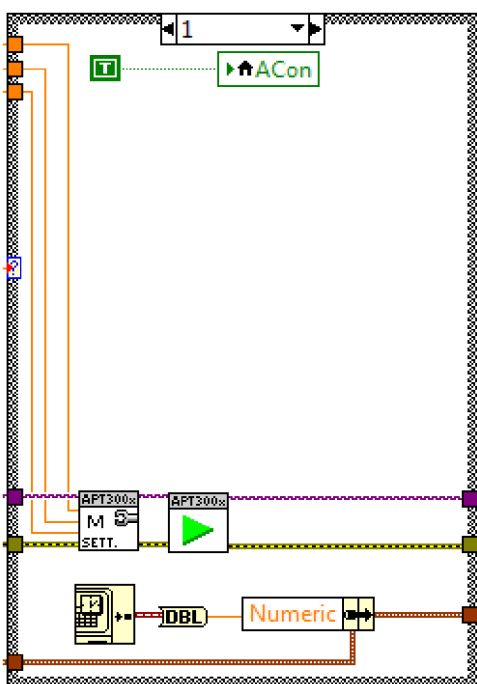
1. Tato část programu slouží pro inicializaci zdroje. Dodáme sem informaci, na jakém portu počítače je zdroj připojen (zde port COM4) a dále nastavíme operační režim na manuální. Zdroj nabízí ještě jeden pokročilejší režim, ten je však pro naše účely celkem zbytečný.
2. Aby nebyl program zbytečně náročný na systémové prostředky, nachází se zde zpoždovač cyklu.
3. Toto jsou vstupní okénka pro zadání kýžených hodnot zdroje.
4. Tato struktura má za cíl vybrat, jaká akce bude následovat. Pokud uživatel zadá akci, že chce zapnout zdroj, přepne se struktura *case* na příslušnou další stránku, pokud je zdroj již zapnutý a uživatel ho chce vypnout, je postup obdobný. Bez uživatelské interakce bude probíhat čtení dat. Struktura, která zde řeší výběr je v podstatě nadstavbou LabView umožňující textové programování, jehož syntaxe je velice podobná programovacímu jazyku C.
5. Tato část se stará o odečítání času, jenž uběhl od zapnutí zdroje. Popis funkce bude následovat níže.
6. Tato část má za úkol číst a zobrazit dostupné parametry výstupu zdroje.
7. Zde se nachází pouze blok pro uzavření komunikace se zdrojem a výstup clusteru, který je součástí podprogramu pro výpočet času.
8. Tento blok již slouží pouze k zavření čelního panelu programu.

Zde tedy následuje obrázek celého programu:

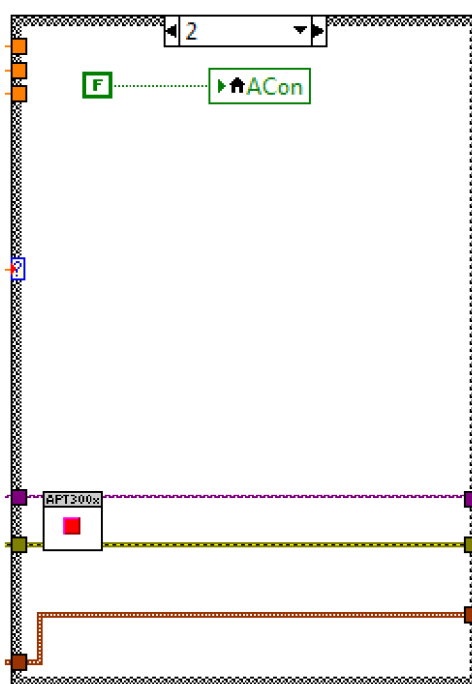


Obrázek 22 Náhled na program pro ovládání střídavého zdroje

Následují zbylé stránky struktury *case*:



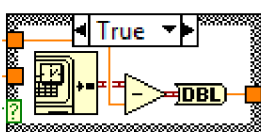
Obrázek 23 Zapnutí zdroje



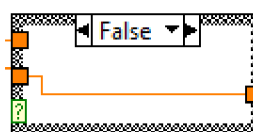
Obrázek 24 Vypnutí zdroje

Na obrázku 23 vidíme dva bloky k ovládání zdroje. Ten první přijímá hodnoty napětí, frekvence a proudu, které byly výše označeny bodem 3 a následující blok zapne výstup zdroje.

Níže se pak nachází část podprogramu pro odečítání doby spuštění zdroje. Tato část zapíše do první proměnné clusteru přesný čas v době zapnutí zdroje. Zbytek podprogramu se nachází, jak bylo výše nastíněno v blízkosti bodu 5. Jde o další strukturu *case*.



Obrázek 25 Podmínka splněna



Obrázek 26 Podmínka nesplněna

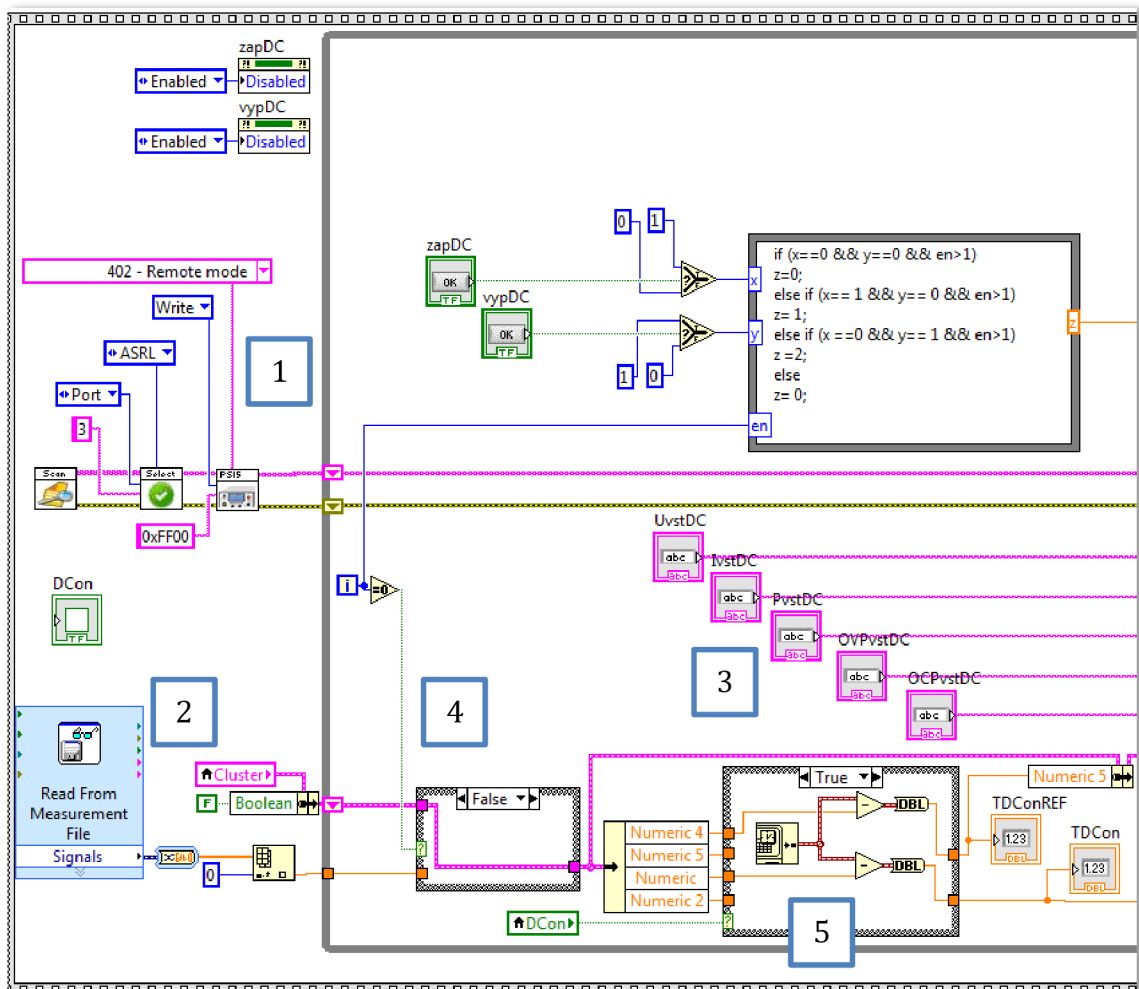
Pokud je podmínka splněna, tedy zdroj je v zapnutém stavu, tak neustále odečítáme od aktuálního času čas od zapnutí zdroje a výsledek se ukládá do druhé proměnné clusteru. V momentě kdy se zdroj vypne, načte se z clusteru obsah druhé proměnné a tatáž hodnota se neustále posílá zpět tamtéž a zároveň do indikátoru. V momentě kdy zdroj opět zapneme, vše pak začíná nanovo, tedy od nuly.

Obrázek 24 pak vypovídá již sám za sebe, toto okno obsahuje pouze blok pro vypnutí zdroje.

6.2 Program pro ovládání DC zdroje

Tento program je již příliš velký na zobrazení v jednom obrázku. Čtenář snad promine, že byl rozdělen na dva. Polehčující okolností může být možná fakt, že většina prvků tohoto programu se shoduje s programem AC zdroje. Odlišnosti můžeme najít především v ovládacích blocích zdroje, kterých je pro dosažení stejného účelu více.

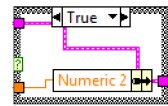
Dalším podstatným rozdílem je podprogram pro počítání času, od kterého je požadována funkce navíc. Tou je počítání celkového času, po který byl DC zdroj zapnut. K němu je totiž trvale připojena korekční žárovka, která má přesně udáno, po jakou dobu provozu jsou její parametry garantované. Aby tento celkový čas byl uložen a následně dobře dostupný, ukládá se do zvláštního textového souboru. Pokud by byl tedy zdroj provozován mimo tento program, není problém textový soubor otevřít a hodnotu připsat.



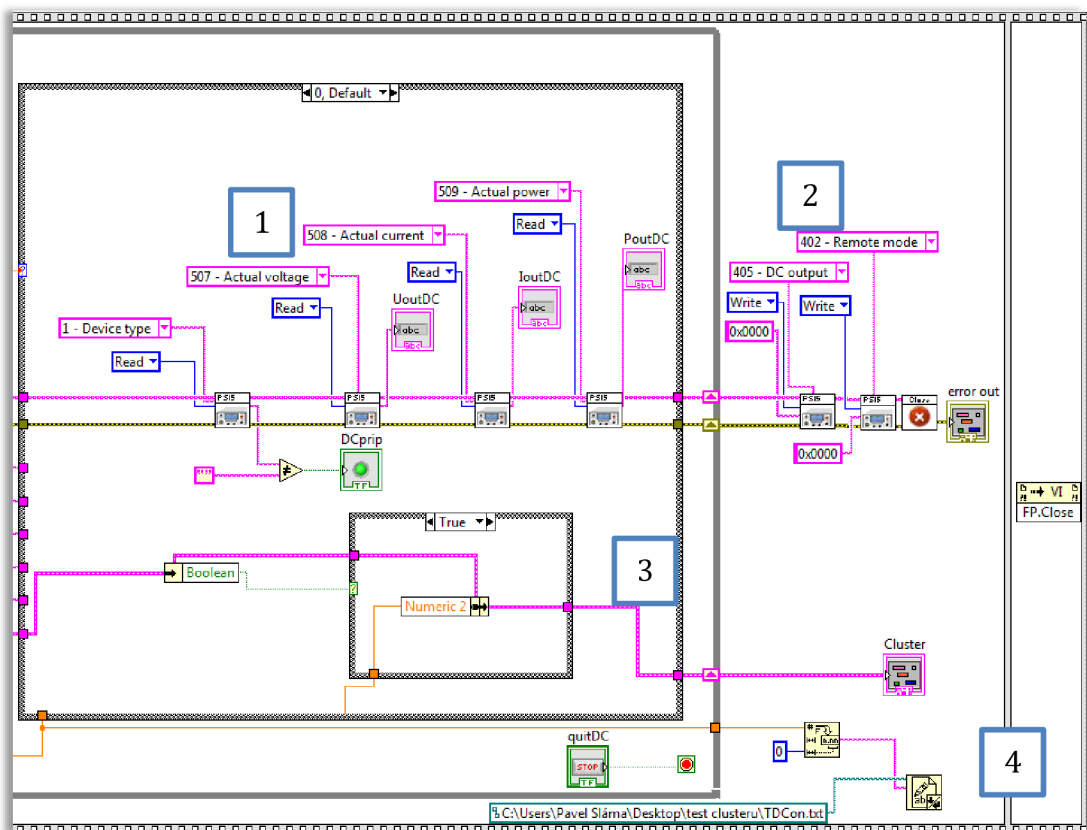
Obrázek 27 Levá polovina programu DC zdroje

Na obrázku 27 je tedy první polovina programu pro ovládání DC zdroje. Analogicky jako při popisu AC zdroje si zde program popíšeme po bodech. Protože se některé z nich shodují, budou z popisu vynechány.

1. Prvním bodem je inicializace zdroje. Na začátku je blok, který prohledá porty počítače a další blok provádí výběr portu. Následuje blok, pomocí kterého dojde k přepnutí zdroje do dálkově ovládaného režimu.
2. Tento blok slouží pro otevírání externích souborů. V tomto případě jde o textový soubor ukrývající dosavadní celkovou dobu zapnutí korekčního zdroje. Informace z textového souboru je z formátu *string* následně převedena na číselný formát.
3. Rozdílem proti AC zdroji je zde větší množství parametrů, které lze nastavit. Zdroj dokáže regulovat výstup na proud i na výkon, a proto můžeme zadat oba parametry. Navíc je zde OVP a OCP, neboli hraniční hodnota napětí a proudu, při jejíž překročení zdroj vypne.
4. Tato case struktura pouze kontroluje, zda jde o nultou iteraci programu. Jestliže ano, načte data ze souboru a uloží je do clusteru. Viz obrázek 28. Při dalších iteracích jí datový tok už jen projde, jak je vidět v samotném obrázku 27.
5. Zde probíhá výpočet času zapnutí. Algoritmus bude rozveden níže.



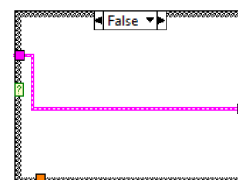
Obrázek 28
Podmínka *True*
bodu 4



Obrázek 29 Pravá polovina programu DC zdroje

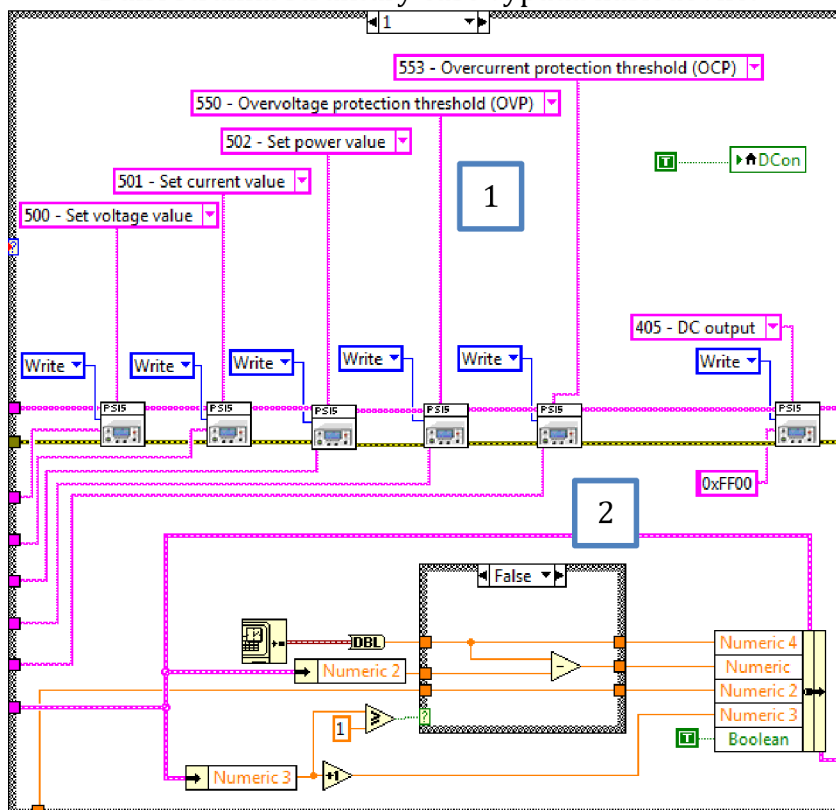
Na obrázku 29 je druhá polovina programu DC zdroje.

1. Největší část zde zabírá struktura *case*. V tomto případě je její velikost dána především bloky pro zápis a čtení do zdroje. Také obsahuje část podprogramu pro měření času. Ten bude popsán níže.
2. Zde se uzavírá komunikace se zdrojem. Pokud je zapnutý, je provedeno vypnutí, a je přepnut zpět na lokální ovládání.
3. V tomto bodě se opět kontroluje, jestli byl už zdroj zapnut. Pokud ne, datový tok pouze projde naskrz. Po jeho zapnutí je však už vždy zapsána nová hodnota spuštění zdroje. Musí to tak být, protože nechceme přepsat hodnotu načtenou ze souboru. V momentě spuštění zdroje totiž dojde k jejímu odečtení od doby posledního spuštění zdroje a proměnná může být dále přepsána. To bude dále popsáno u obrázku 31.
4. Po opuštění smyčky programu zde dojde k uložení doby spuštění zdroje do textového souboru.



Obrázek 30 Podmínka False bodu 3

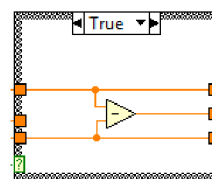
Další stránka struktury *case* vypadá následovně:



Obrázek 31 Stránka struktury *case* pro zapnutí zdroje

1. Nachází se zde bloky, které jsou přepnuty do režimu zápisu dat. Nejdříve se zapíše všechny parametry a potom dojde k zapnutí zdroje.
2. V tomto bodě je popsáno, jakým způsobem funguje algoritmus pro měření času. Zde je zobrazena podmínka *false*, která proběhne jen při prvním zapnutí zdroje. Do proměnné *Numeric 4* ukládáme aktuální čas. Bude použit pro výpočet doby od spuštění zdroje. Do proměnné *Numeric* uložíme hodnotu celkového spuštění zdroje načtenou ze souboru odečtenou od aktuálního času, čímž získáme základ pro další počítání celkové doby spuštění. Položka *Numeric 2* pak bude sloužit k udržení celkové doby spuštění zdroje po jeho vypnutí. Proměnná *Numeric 3* je pouze pomocná a ukládá se do ní počet spuštění zdroje. Proměnná *Boolean* je pak na tom stejně a slouží k identifikaci, jestli byl zdroj už spuštěn.

V případě že je podmínka ve stavu *True*, tzn. zdroj byl spuštěn více než jednou, je situace podobná, pouze do proměnné *Numeric* je ukládán rozdíl aktuálního času a času jenž proběhl od začátku.



Obrázek 32 Stav *true*

Na poslední stránce struktury case probíhá pouze vypnutí zdroje, a proto není nezbytné ji zde více rozebírat.

6.3 Společné náležitosti programů pro AC i DC zdroj

Zde provedeme krátký rozbor náležitostí nezbytných, aby program fungoval i na dalších počítačích, např. při obměně výbavy laboratoře.

- První bod je poměrně nasnadě, je totiž potřeba změnit cestu k textovému souboru sloužícímu pro ukládání celkové doby spuštění zdroje v programu DC zdroje. Ta se nalézá na obrázku 29 v bodě 4.
- Dalším bodem je vytvoření nového *.exe* souboru pro oba programy. Zde je totiž potřeba myslet na to, že chceme programy zprovoznit pro rozhraní *ActiveX*. Při vytváření *.exe* souborů je potřeba v nabídce *Advanced* zvolit *Enable ActiveX server* neboli v nabídce *Pokročilé* zvolit *Povolit server ActiveX*. V obecných informacích pak už jenom zvolíme cíl uložení a můžeme soubor vytvořit.

- Následně je třeba zanést náš ActiveX server do registrů Windows. Ačkoli by toto při vytváření *.exe* mělo probíhat automaticky, neukázalo se to být pravdou. Jistější je proto následující postup.

V první řadě si najdeme náš soubor *.exe* a spustíme ho s administrátorskými právy (pravé tlačítko myši – *Spustit jako správce*). Program poté můžeme ukončit.

Dále zvolíme *Start* – vyhledáme program *Spustit* a otevřeme ho. Do okénka zadáme následující příkaz:

```
<plná cesta k .exe souboru> \RegServer
```

Příklad:

```
D:\ProgDip\DC\Application.exe \RegServer
```

Toto provedeme pro oba programy [19].

- Provedeme poslední krok, který slouží ke skrytí běžících programů pro ovládání obou zdrojů. Tím je přidání řádku do konfiguračního souboru, který se nalézá ve stejné složce, ve které jsme vytvořili *.exe* soubor. Má stejné jméno jako náš program, pouze s příponou *.ini*.

Na konec tohoto souboru přidáme řádek:

```
HideRootWindow=True
```

- Posledním krokem je potom provedení úpravy ve hlavním programu, ze kterého budeme výše popsané programy ovládat. Tento krok bude popsán níže při popisu samotného programu.

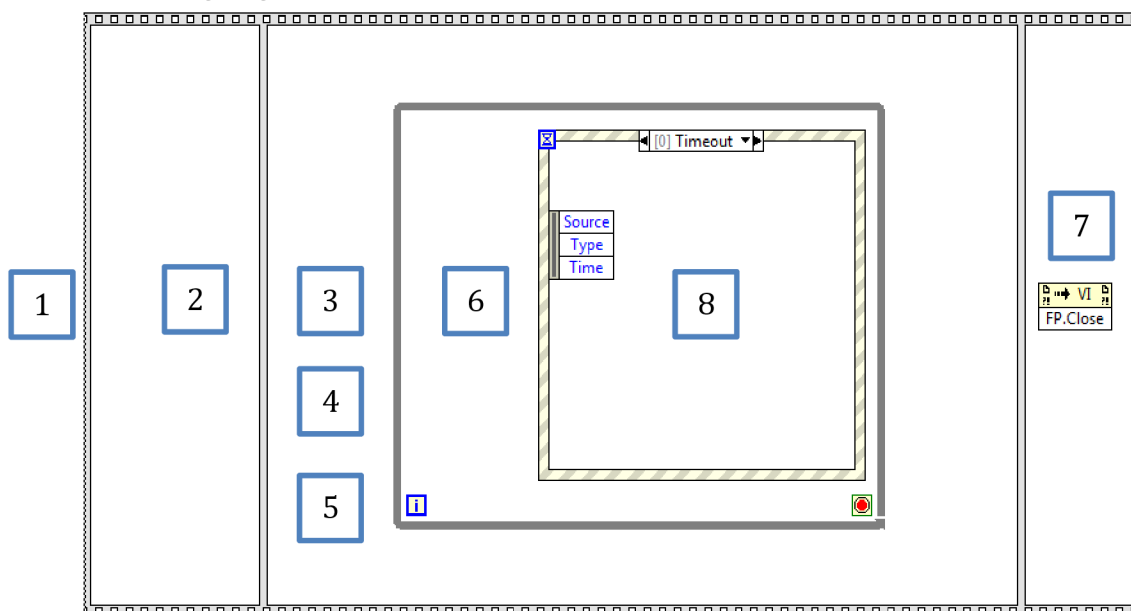
7 HLAVNÍ PROGRAM

Nyní se konečně dostáváme k hlavní části diplomové práce, a to programu pro měření spektra. Již v úvodu je však třeba předznamenat, že zde bude popis dosti obtížný, neboť se jedná o poměrně rozsáhlý program. Jelikož by se celý nevešel na stránku tak, aby byla alespoň trochu vidět struktura, alespoň si ji naznačíme a následně se zaměříme na jednotlivé části. Zájemci o zhlédnutí celého programu nezbude než se podívat do příloh, za což mu náleží omluva.

Jednotlivé body budou tentokrát objemnější, a proto jim budou věnovány podkapitoly.

Pozn.: Na některých obrázcích bude proti originálnímu programu lehce změněno rozložení prvků. Funkčnost bude zachována, jde pouze o to, aby se sem vše vešlo.

Pozn.2.: Tento program v sobě má již komentáře. Jsou psány úsečně aby nezabíraly zbytečně mnoho místa.



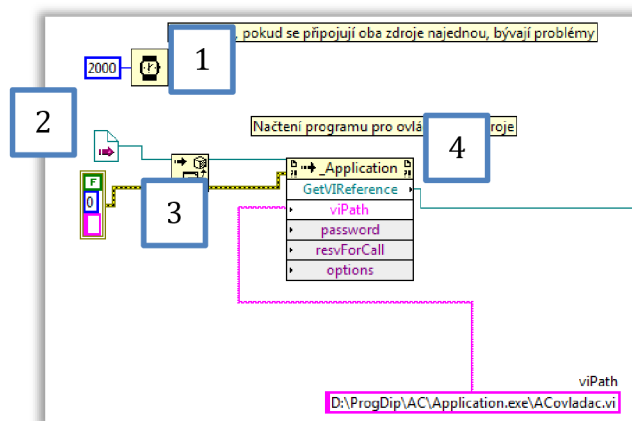
Obrázek 33 Naznačení struktury hlavního programu

7.1 Nastavení výchozích hodnot programu

V tomto bodě, tedy před samotnou strukturou programu se nachází jednak prvky, se kterými se nadále nepracuje přímo ale např. pomocí struktury *event*, pak také bloky, které přivádí jiné prvky do výchozího stavu či jiné inicializační prvky, které mají provést svoji funkci na úplném začátku programu.

7.2 Zapnutí programu pro ovládání AC zdroje

Zde se nachází bloky, které slouží k zavolání programu AC zdroje, jeho otevření a přeposlání reference na tento program dále. Pokud se totiž připojíme na server ActiveX, nepracujeme pak dále v programu s obyčejným datovým tokem ale právě s referencí na tento server.



Obrázek 34 Otevření programu pro ovládání AC zdroje

1. Jak je asi už patrné z komentáře, nachází se zde časové zpoždění. Když se připojovaly oba zdroje najednou, docházelo obvykle u jednoho z nich k přerušení spojení.
2. Tento blok slouží pro načtení reference `.exe` souboru. Pokud by bylo nezbytné program přesouvat na jiné PC, je zde třeba provést změnu. Ta spočívá v tom, že nejdříve provedeme úkony z kapitoly 6.3 a následně na tento blok klikneme pravým tlačítkem myši a zvolíme z nabídky *Select ActiveX Class* a následně *Browse*. V této nabídce vybereme soubor `.tlb` příslušející k danému programu, který chceme připojit. Tento soubor najdeme v adresáři, do kterého jsme vytvořili `.exe` soubor. Následně zaškrtneme políčko *Show Creatable Objects Only* a vybereme položku `_Application`.
3. Další blok nás pak referencí odkáže na příslušný objekt ActiveX. Tento blok je při každé změně provedené v předchozích krocích nutno smazat a vybrat ho z nabídky a umístit znovu. Je to proto, že se pokaždé tento blok přiřadí k referenci a při její změně nefunguje.
4. Tento blok nám umožní načíst soubor `.vi`, jenž je obsažen v našem souboru `.exe`. Jak to funguje je popsáno na začátku kapitoly 6. Dále už nám jako výstup pokračuje reference přímo na `.vi`.

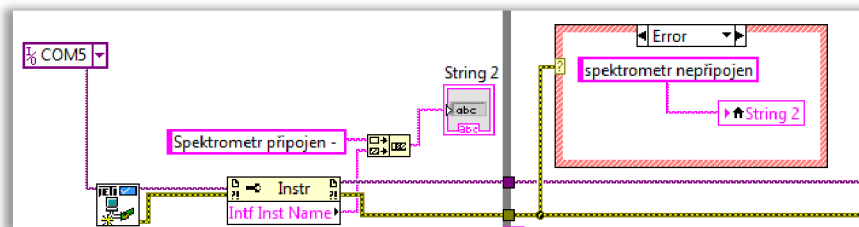
7.3 Zapnutí programu pro ovládání DC zdroje

Tento program je totožný s tím pro AC zdroj. Pouze se zde nenachází časové zpoždění a samozřejmě i adresáře jsou rozdílné.

7.4 Inicializace spektrometru a stav připojení

Inicializace je poměrně zřejmá, pouze vybereme port, na jaký jsme připojeni. Dalším krokem je čtení stavu připojení. Pokud je přístroj připojen, načte se ze sběrnice port připojení a zobrazíme si ho s hláškou v záhlaví programu.

Již ve smyčce programu se pak nachází kontrola, jestli je spektrometr připojen. Jestliže není, změní se hláška v záhlaví.



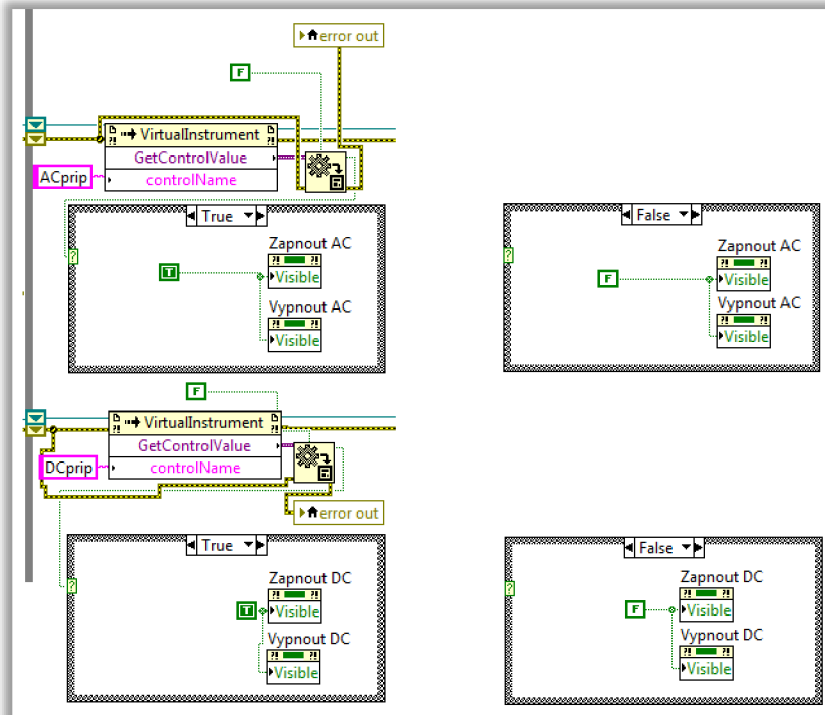
Obrázek 35 Inicializace spektrometru

7.5 Načítání externích souborů

Zde se nachází bloky, které provádí načtení tabulek s hodnotami nezbytnými pro výpočty světelných veličin. Načítají se zde hodnoty trichromatických činitelů, tabulka s izotermami pro výpočet teploty chromatičnosti, tabulka pro výpočet zářiče typu D a Spektrální hodnoty testovacích vzorků TSC1-15 pro výpočet indexu podání barev. Načtené hodnoty potom ukládáme do clusterů, kromě dat TSC1-15, které ponecháme ve formě dynamických dat, neboť je tato tabulka dosti velká pro pohodlnou manipulaci v rámci clusteru.

7.6 Kontrola připojení zdrojů

V tomto místě se nachází pro každý zdroj struktura, která čte stavovou proměnnou poslanou z příslušného programu. Tato proměnná nám říká, zda je zdroj skutečně připojen nebo ne. Na základě toho se zobrazí nebo zmizí tlačítka pro zapnutí a vypnutí zdrojů.



Obrázek 36 Kontrola připojení zdrojů

Pro oba zdroje je program podobný. Čteme proměnnou *ACprip*, popř. *DCprip* a na základě toho, jestli je podmínka splněna nebo ne, se změní příznak, nebo tzv. *Property* vybraných prvků. Konkrétně zde měníme příznak viditelnosti ovládacích tlačítek zdrojů.

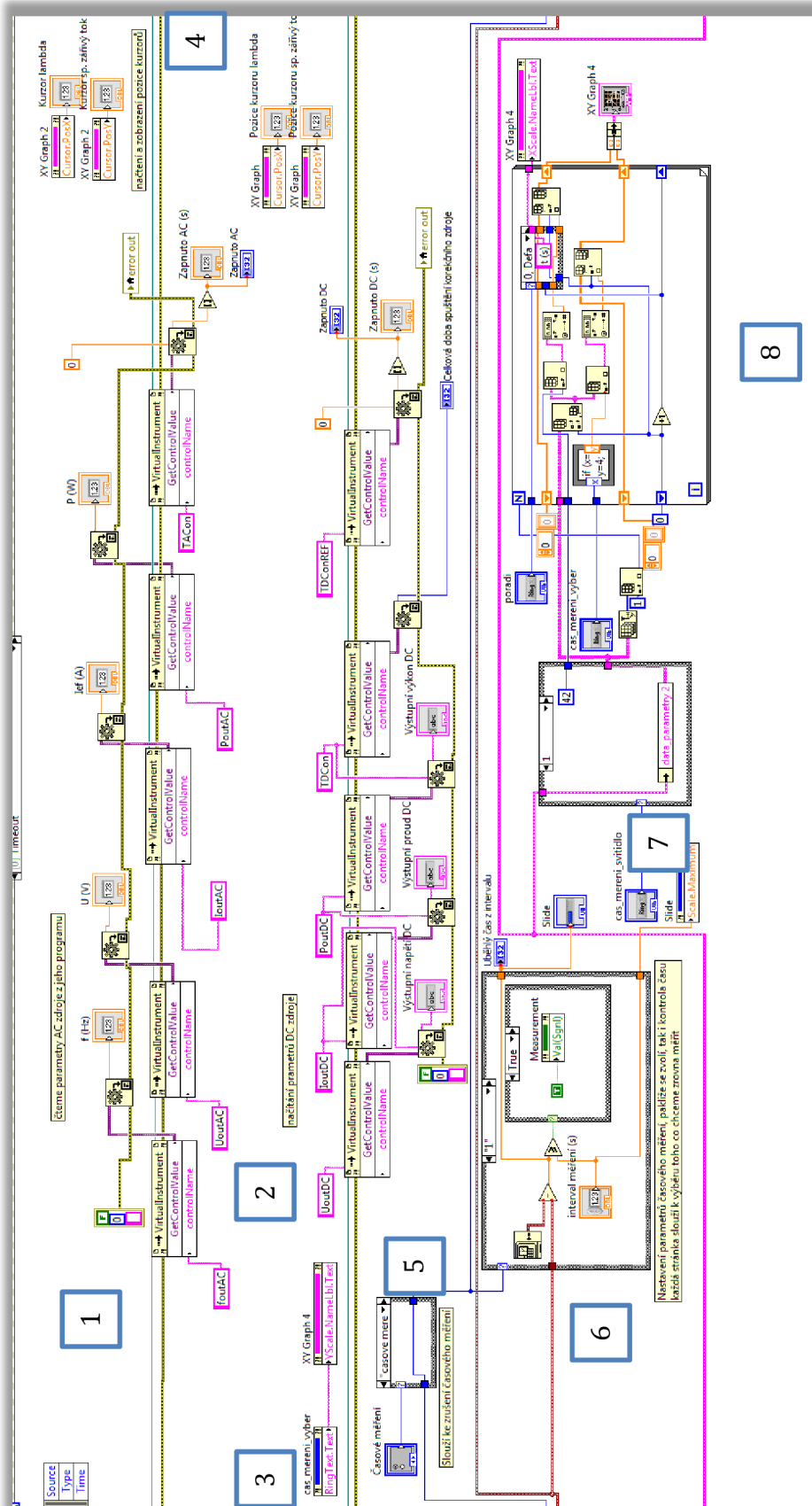
7.7 Zavření okna

Tato část struktury má svoje vlastní okénko pouze proto, aby se zavření okna s programem odehrálo až po ukončení všech ostatních aktivit v programu.

7.8 Struktura event

Tato část programu je přirozeně nejrozsáhlejší. Obsahuje většinu prvků, které vykonávají nějakou činnost. Protože program obsahuje řadu prvků, které slouží ke čtení údajů, byla zvolena struktura obsahující i okno *timeout*. To se vykoná, jestliže nedojde k interakci s programem v určitém čase. Zde byl tento čas zvolen na 100 ms, jelikož zde dochází převážně ke čtení výstupních údajů zdrojů, což zde nejsou veličiny, které by se měnily rychle v čase.

Na další straně následuje nultá stránka struktury *event*, a to *timeout*.

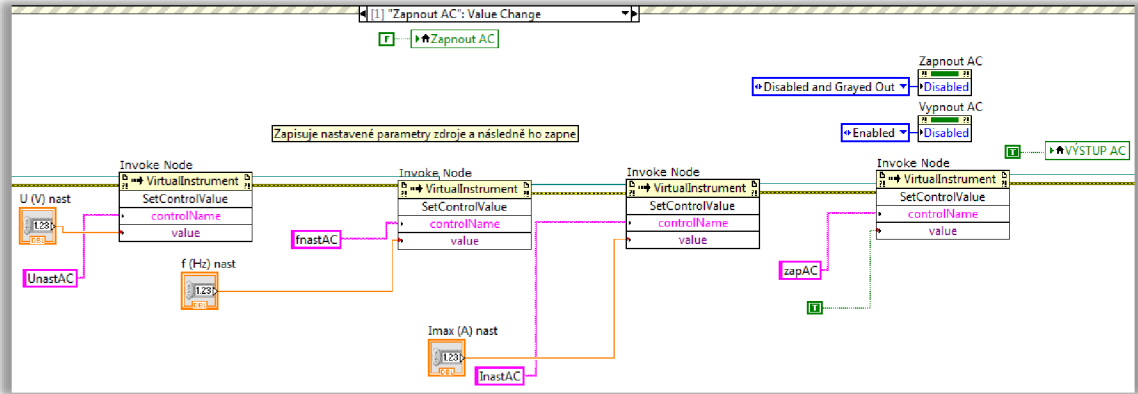


Obrázek 37 Struktura event – okno timeout

Provedeme si opět rozbor obrázku po bodech.

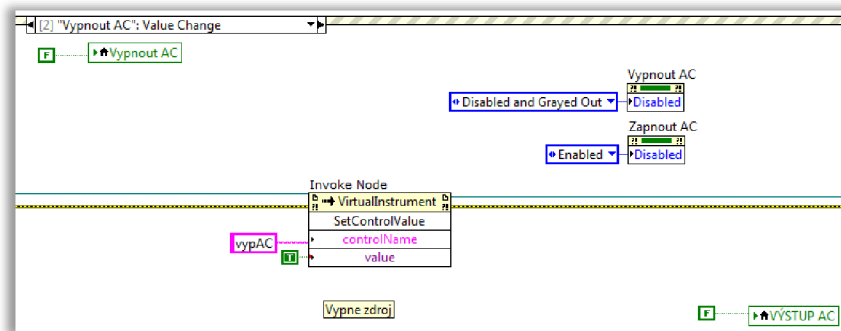
1. Zde se nachází bloky, které cyklicky čtou informace dodávané AC zdrojem. Jedná se o výstupní napětí, proud, frekvenci, výkon a dobu uplynulou od zapnutí zdroje. Tyto hodnoty jsou pak zobrazeny na čelním panelu a při měření spektrometrem se ukládají do paměti.
2. Tento bod je obdobný jako první, pouze se zde čtou hodnoty DC zdroje. jedná se o výstupní napětí, proud, výkon a časové hodnoty – celkový čas zapnutí zdroje a čas od posledního zapnutí zdroje.
3. Tento blok nastavuje, jaká veličina se zobrazí na ose X grafu časového průběhu na základě toho, jakou pozorovanou veličinu si uživatel zvolí.
4. Tento bod slouží jako ukazatel pozice kurzoru v obou spektrálních grafech obsažených v programu. Načítají příslušný parametr grafu a hodnoty jsou odeslány do příslušných okének na čelním panelu.
5. Pomocí této struktury lze ukončit probíhající časové měření.
6. Tato část slouží pro zjišťování, zda už uběhl nastavený interval při časovém měření. Od aktuálního času se odečte čas od posledního měření a je porovnáváno, zda je tento rozdíl větší než nastavená hodnota. Pakliže je, je nastaven příznak bloku *Value (Signalling)*, který přepne na námi chtěnou stránku struktury *event*. Časové měření je možno provádět pro všechny druhy měření v programu přítomné.
7. Tato struktura slouží společně s tou následující k zobrazování časového měření v grafu. Tato část má na starost výběr vstupních dat na základě toho, co si uživatel zvolí z nabídky. Na základě toho pošle do další struktury tabulku se změřenými parametry, ze kterých si uživatel vybere, který chce zobrazit. Také odešle, kolikátý v pořadí této tabulky je časový parametr, který bude vyneseno na ose X. Jsou zde také odesílány na čelní panel údaje o tom, jaká část intervalu již uběhla.
8. Tato struktura, navazující na předešlou slouží k separaci vybraných veličin z tabulky s parametry, aby mohly být zobrazeny v grafu. Jde o strukturu *for*, ve které se prochází tabulka po sloupcích a z každého jsou pak vybrány dvě veličiny. Čas nebo pořadí pro osu X a výběr z nabídky pro osu Y. Místo v tabulce je dáno textovou strukturou, která obsahuje krátký program pro přiřazení polohy v tabulce pro jednotlivé vstupy. Uživatelský vstup nabývá hodnot 0-3 a tato struktura jim jen jednoduše přiřazuje umístění v tabulce. Výsledek je potom odeslán do grafu.

Další v řadě jsou okna pro ovládání zdrojů. Jde o okna vypnutí a zapnutí. Protože jde o velmi podobný princip u AC i DC zdroje, bude zde pro příklad rozebrán pouze AC zdroj.



Obrázek 38 Struktura event – okno Zapnout AC

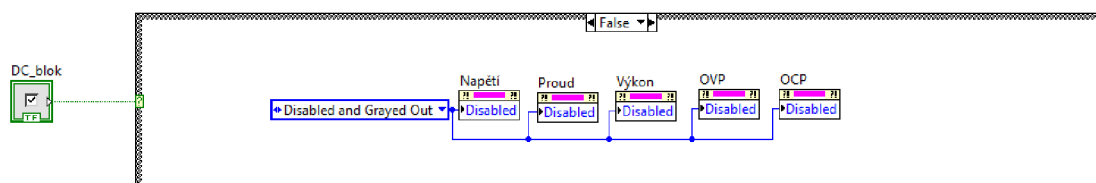
Na obrázku 36 probíhá zápis nastavených hodnot do zdroje. Poslední blok v řadě zdroj zapne. Při zapnutí zdroje dojde k „zatemnění“ zapínacího tlačítka, aby nešlo znovu zmáčknout, dokud se zdroj nevypne, a naopak vypínací tlačítko je zpřístupněno. Do indikátoru výstupu je zapsána hodnota *true*. Ten se tedy rozsvítí.



Obrázek 39 Struktura event – okno Vypnout AC

Obdobně probíhá vypnutí zdroje. Je do něj odeslán příkaz k vypnutí, je zatemněno vypínací tlačítko a zapínací tlačítko je zpřístupněno. Do indikátoru výstupu zdroje je odeslána proměnná *false*.

Jak již bylo řečeno, zapnutí a vypnutí DC zdroje probíhá obdobně a nebude proto rozebráno. Další na řadě je blokáce změny vstupních hodnot DC zdroje. Na ten je stále zapojen tentýž korekční světelný zdroj, a proto jsou tyto hodnoty chráněny proti náhodnému přepsání. Uživateli je však umožněno zaškrtnout si možnost změny. A to je právě obstaráváno tímto oknem z obrázku 40.

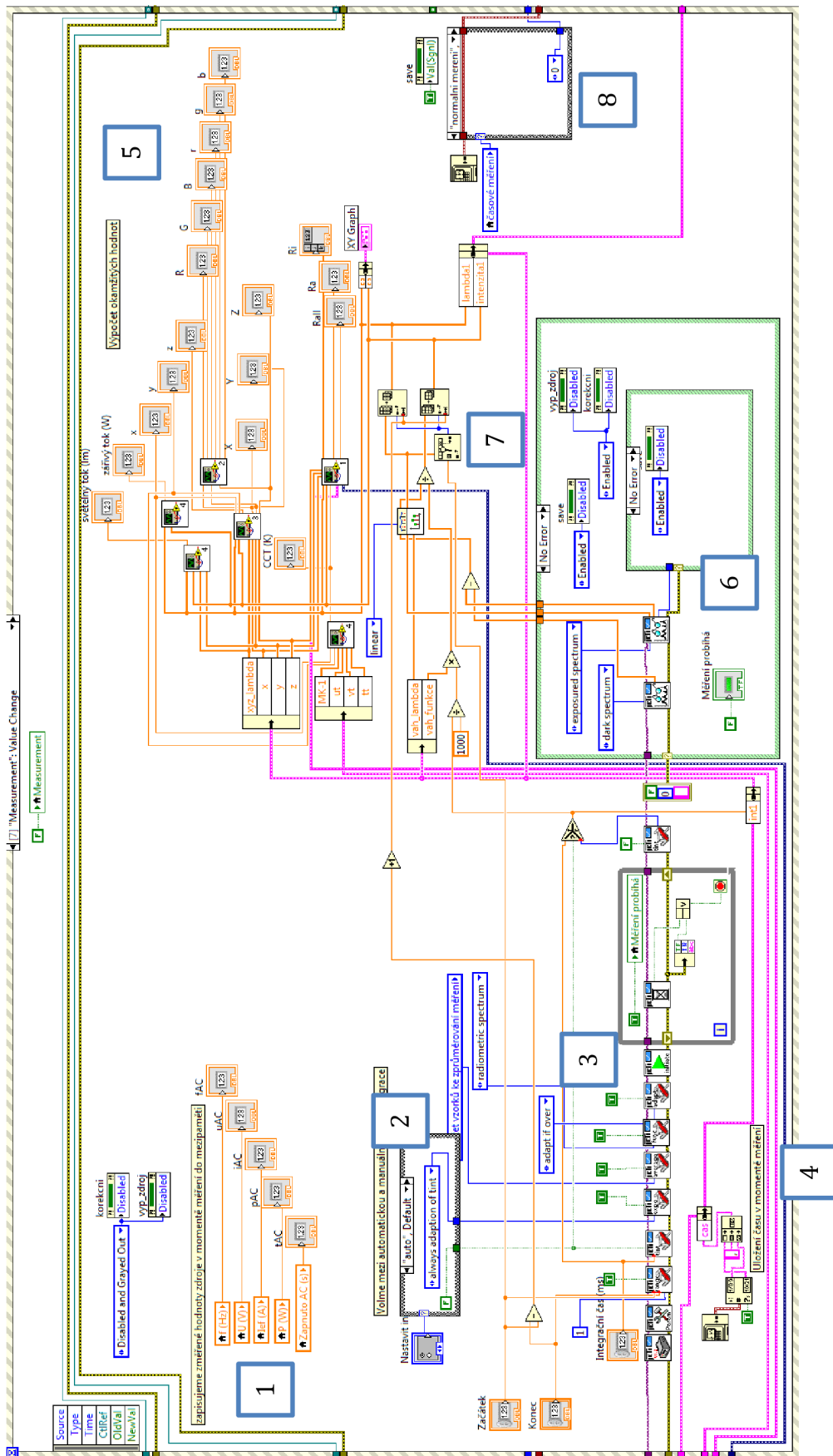


Obrázek 40 Struktura *event* – funkční část okna *DC blok*

Princip je zde v zásadě jednoduchý. Na základě stavu vstupu je změněn stav okének hodnot výstupu DC zdroje. Buďto jsou normálně přístupné, nebo jsou zatemněné. Výchozí stav je zatemněný.

Následuje okno pro ukončení celého programu s číslem ve struktuře 6. Ten má na starost nejen zavření okna, ale i ukončení programů pro ovládání zdrojů, které běží na pozadí. To proběhne tak, že je odeslán signál pro ukončení těchto programů a je z paměti vymazána reference, která na ně odkazuje. Také je zde ukončena komunikace se spektrometrem a následně už je jen vydán příkaz pro ukončení smyčky. Obrázek této části se zde nenachází, neboť je pro svou jednoduchost a přímočarost zbytečný.

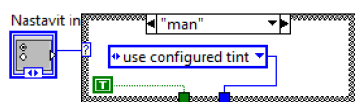
Opět následuje objemnější část programu. jde o okno měření. Podobná okna jsou v celém programu 3, protože měříme buďto samotný měřený zdroj, nebo provádíme korekční měření. Ty se sestávají ze spektra při vypnutém zdroji a zapnuté korekční žárovce a také měření spektra při vyjmutém zdroji a zapnuté korekční žárovce. Tyto struktury jsou si velice podobné, a tak bude opět pro demonstraci proveden popis pouze jedné z nich



Obrázek 41 Struktura event – okno Measurement

Předchozí obrázek bude opět po bodech rozebrán:

1. Tento bod ukládá do dočasných proměnných stav výstupních veličin zdroje. Ty budou v později načteny a uloženy do tabulky.
2. Následuje bod, který souvisí s nastavením parametrů spektrometru. Pomocí této struktury se volí, zda chceme nastavit dobu integrace ručně, nebo zda si ji má přístroj určit automaticky.



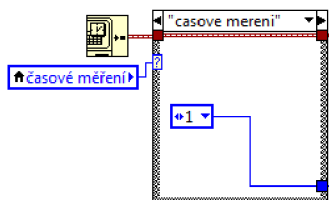
Obrázek 42 Nastavení
integračního času

Na obrázku 41 je zvolena možnost auto, takže dojde s každým měřením k automatickému určení vhodného integračního času. Na obrázku 42 je volba ručního zadávání. To probíhá samostatným okénkem.

3. Tento bod navazuje na předchozí, neboť bloky, které slouží k volbě integračního času se nacházejí zde. Stejně tak se zde nastavují všechny ostatní parametry. Uživatel tak má možnost si zvolit, z kolika hodnot chce zprůměrovat výsledné spektrum, je zde nastaveno, jak se má přístroj chovat při přesycení či jakou funkci chceme vlastně měřit.
4. Zde dochází k zápisu času, ve kterém k měření došlo. Je zde ukládán jak čas, tak i datum.
5. Další bod vypadá na první pohled poměrně chaoticky, slouží však pouze k výpočtu okamžitých radiometrických, fotometrických a kolorimetrických hodnot. Obsahuje tedy bloky, které byly vytvořeny uživatelem a nacházejí se v nich algoritmy pro výpočty požadovaných hodnot.
6. Následuje důležitý bod úzce spjatý i s bodem 7. Jde zde o to, že si načteme změřené „surové“ spektrum (neboli také „ozářené“ spektrum, z originálu *exposed*) a odečteme od něj temné spektrum, čímž vzniknou data, která je ještě třeba podělit kalibračními daty. Tyto hodnoty máme dostupné najednou, protože při standartním procesu měření vždy proběhne 1 rychlé měření, které určí integrační čas a poté dojde k měření „ozářeného“ spektra a následné měření temného spektra, u obou se stejným integračním časem. My zde však nepoužíváme celkové kalibrované spektrum, které je schopen spočítat samotný přístroj, protože výrobce nedal programátorům možnost, který kalibrační soubor si chce zvolit. Proto zde právě provádíme celý výpočet manuálně.
7. Zde tedy máme dostupné výstupy potřebných spekter a provedeme výpočet. Nejdříve odečteme temné spektrum od „ozářeného spektra. Výsledek posléze interpolujeme podle kalibrační funkce, která odpovídá odrazovým schopnostem kulového integrátoru. Můžeme si všimnout, že kalibrační funkci ještě násobíme integračním časem (čas se ještě dělí tisícem, jelikož ho převádíme z milisekund na sekundy). Nyní je kalibrační funkce i náš

dosavadní výsledek na stejných vlnových délkách a my proto můžeme kalibrační funkcí dosavadní výsledek podělit. Dále už si jenom zvolíme rozsah se kterým chceme pracovat, o čemž rozhoduje uživatel pomocí vstupu na čelním panelu.

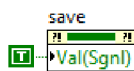
8. Následuje bod, který rozhoduje o tom, jestli se budu měřit hodnoty za sebou v časových intervalech, nebo zda bude měření manuální. Na obrázku 41 je



Obrázek 43 Volba časového měření

okno struktury pro normální měření. Neproběhne tedy žádná akce. Na obrázku 43 je potom okno pro měření v čase. jedinou změnou je přepis parametru na hodnotu jedna, odpovídající časovému měření měřeného zdroje. Tato hodnota je následně přečtena a vyhodnocena v okně *timeout*.

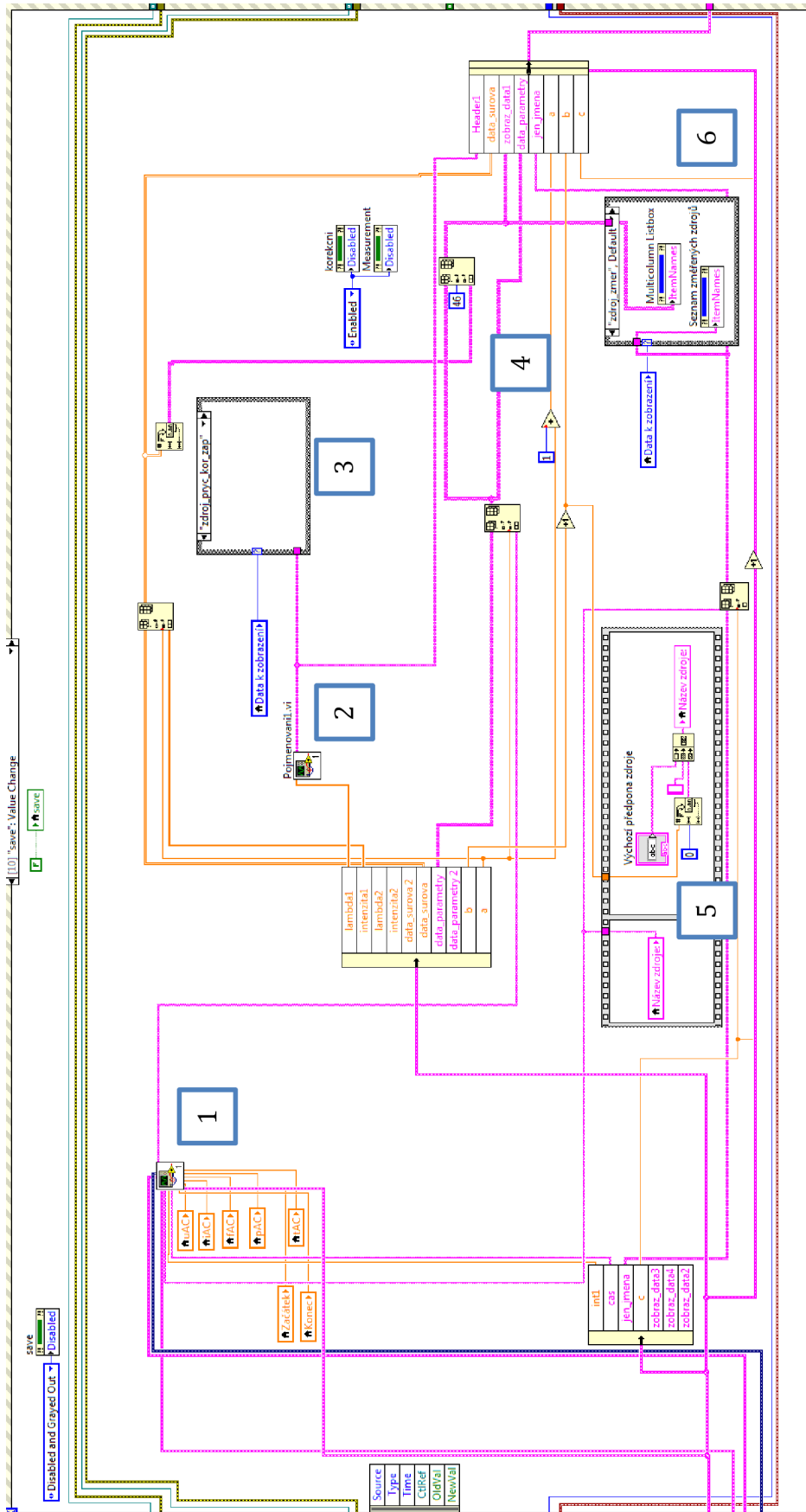
Pozorný čtenář si mohl všimnout, že nad strukturou bodu 8 se nachází ještě blok pro přepnutí stránky struktury *event* na stránku *save*.



Obrázek 44 Programové přepnutí struktury event

To znamená, že po proběhnutí všeho na stránce měření dojde rovnou ke spuštění stránky *save* neboli ukládání do paměti. Tento proces neprobíhá celý na jedné straně, protože by šlo o zhoršení přehlednosti programu.

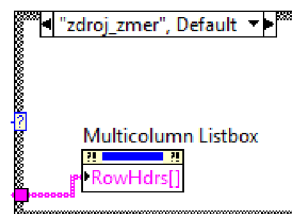
Na tomto místě tedy vynecháme zbylé stránky pro měření, které vypadají obdobně jako ta předchozí a pustíme se do popisu okna *save*.



Obrázek 45 Struktura event – okno save

Popis další stránky struktury *event* – okna *save*:

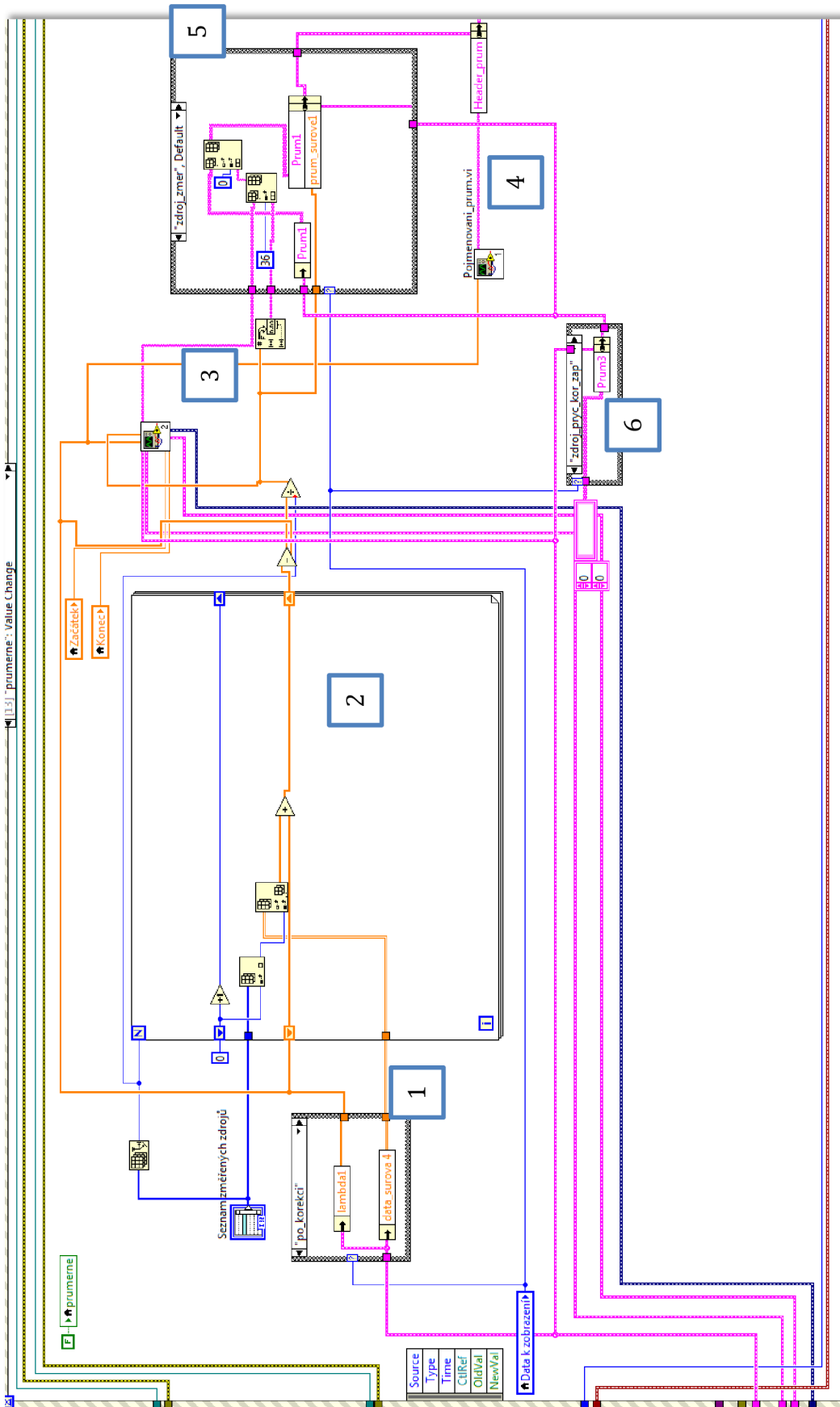
1. Prvním bodem je podprogram *Sdruzeni1* vytvořený za účelem ukládání parametrů do paměti. Má mnoho vstupů, neboť vždy vypočte parametry nově přidávaného zdroje, převede je na formát *string* a poskládá z nich tabulku. Ta bude dále připsána do širší tabulky *zobraz_data* a následně zobrazena v záložce na čelním panelu.
2. Tento bod úzce souvisí s prvním, neboť jde vlastně o „protikus“ k vypočteným parametrům. Jde o jejich názvy, které jsou pak zobrazeny v příslušném sloupci v tabulce výsledků.
3. Zde jde právě o zobrazení výsledků v pevně umístěném sloupci. Struktura kontroluje, která výsledková tabulka je vybrána k zobrazení. Pakliže je vybrána tabulka měřeného zdroje, hodnoty jsou zapsány. To lze vidět na obrázku 46. V ostatních případech struktura nevykoná nic, jak je vidět na obrázku 45.



Obrázek 46 Zápis do pevného sloupce

4. Zde dochází k připočtení jednotky k záznamovým ukazatelům. Ty nám říkají, kolikátý zdroj v pořadí měříme. Také se zde nacházejí 2 bloky funkce *Insert into array*, neboli vložit do tabulky. První z nich připsá poslední záznam parametrů do příslušné tabulky. Druhý z nich tuto tabulku vezme a přidá za ni měřené spektrum a přidá ji do tabulky k zobrazení.
5. Tato struktura má za úkol prvně načíst název zdroje a uložit ho do tabulky v bodu 1 výše a dále ho uložit do tabulky se samotnými názvy. Jakmile se toto vše vykoná, struktura se posune do druhého okénka, kde dojde k automatickému pojmenování zdroje na vyšší číslo. To je zde, pokud se uživateli nechce vyplňovat názvy, může je nechat vyplňovat automaticky. Může si také zvolit předponu, jakou mají zdroje sdílet. Výchozí nastavení je *Zdroj 1*, *Zdroj 2*, atd.
6. Zde, obdobně jako v bodě 3 dochází k rozhodnutí, zda je aktuálně vybrána k zobrazení tabulka, do které zapisujeme. Pokud je vybrána, odehraje se děj z obrázku 45. Pokud není vybrána, je okno této struktury prázdné a nestane se nic.

Dále následuje okno struktury *event*, které počítá ze sloupců v tabulce průměrné hodnoty. To je vhodné použít pro tabulky s korekčními hodnotami, jestliže měříme pouze jeden zdroj ale vícekrát. I pokud měříme více zdrojů, je dobré použít tuto funkci alespoň pro tabulku, kde je korekční zdroj zapnutý a měřený zdroj vyjmutý. Funkci však lze použít pro všechny tři tabulky a uživatel už si pak jen zvolí, z jakých hodnot chce korekci počítat.

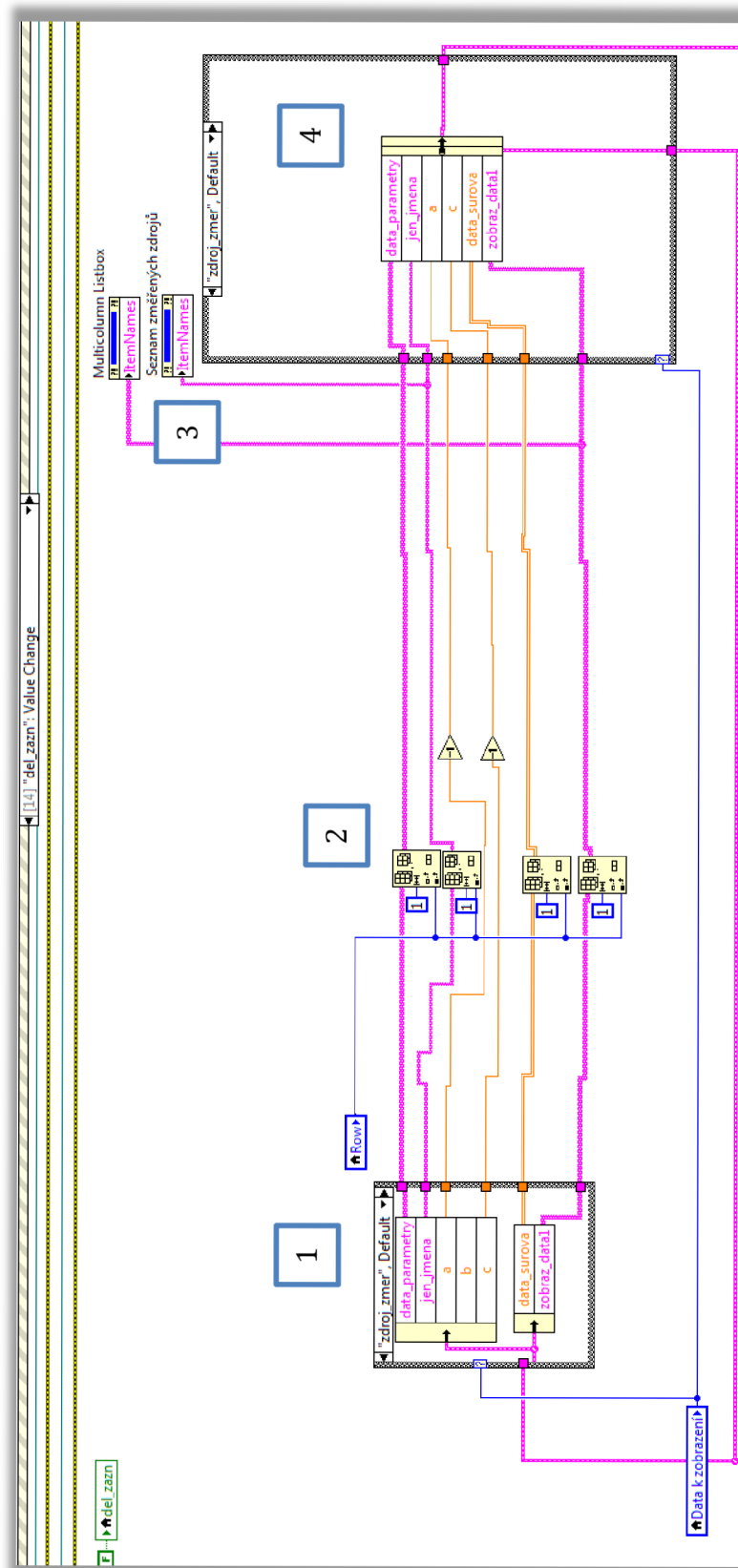


Obrázek 47 Struktura event – okno *prumerne* (nevyužitý prostor byl oříznut)

Následuje již standardně popis obrázku, tentokrát s číslem 47:

1. Na základě toho, jakou tabulku má uživatel zobrazenou je proveden výběr dat, která budou průměrována.
2. Zde se ve struktuře *for* nachází algoritmus provádějící výpočet aritmetického průměru standardně sečtením všech hodnot a následně podílem jejich počtem.
3. Jde o upravenou verzi již výše zmíněného podprogramu *Sdruzeni1*. Tento se jmenuje *Sdruzeni_prum* a složí tedy k výpočtu parametrů a jejich ukládání do tabulky.
4. Tento bod navazuje na předchozí. Jde o podprogram *Pojmenovani_prum*, který vytvoří tabulku s názvy parametrů pro zobrazení v názvovém sloupci výsledkové tabulky.
5. V tomto bodě probíhá ukládání do příslušných tabulek, opět na základě toho, kterou má uživatel zrovna zobrazenou.
6. Tento bod využívá opět stejného principu výběru, se kterými hodnotami má pracovat. Jeho funkcí je nulování tabulky s průměrnými hodnotami, než se do nich bude opět zapisovat.

Další okno struktury *event* slouží k mazání jednotlivých záznamů, se kterými uživatel není spokojen, případně je změřil omylem a podobně. Smaže vždy záznam, na který uživatel dvakrát poklepal myší. Ukazatel na pořadí záznamu, na který bylo poklepano se nachází v okně, které bude probráno až dále, a to je *Seznam změřených zdrojů*.

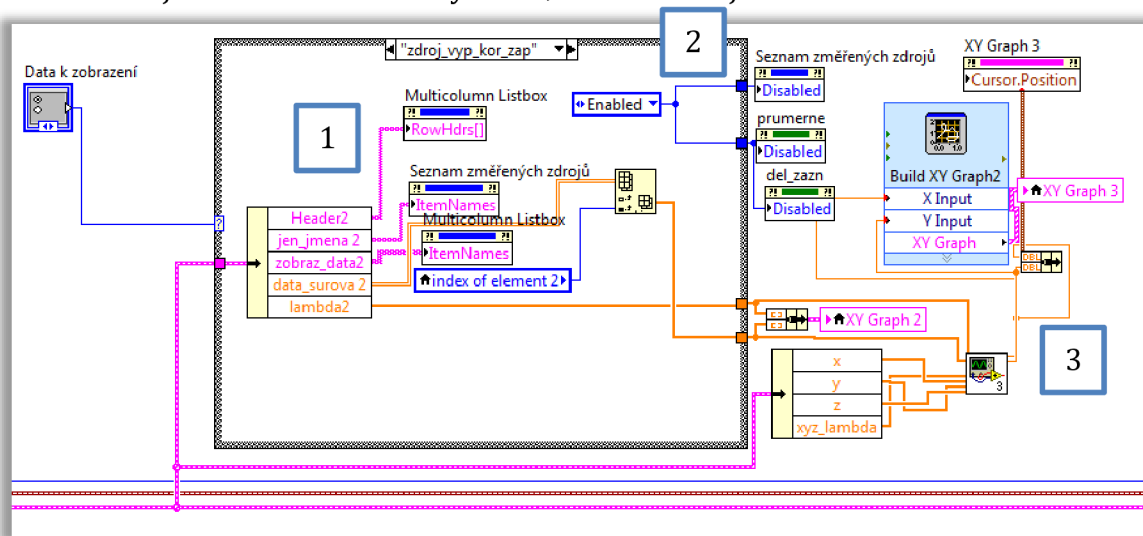


Obrázek 48 Struktura event – okno *del_zazn* (nevyužitý prostor byl oříznut)

Zde je popis obrázku 48:

1. Na základě toho, která tabulka je zobrazena si načteme data, ze kterých chceme mazat záznam.
2. Zde je provedeno samotné mazání. Index záznamu, který má být mazán získáme v jedné z následujících stránek struktury *event*, která je na obrázku 50. Když ho známe, můžeme potom z příslušných tabulek smazat vždy jeden sloupec. Ukazatele a, c, které nám říkají, kolik je již změřeno zdrojů jsou potom poníženy o jeden záznam.
3. Zde dochází k aktualizaci momentálně zvolené tabulky.
4. Podle toho, která tabulka je zobrazena jsou pak přepsány hodnoty v paměti.

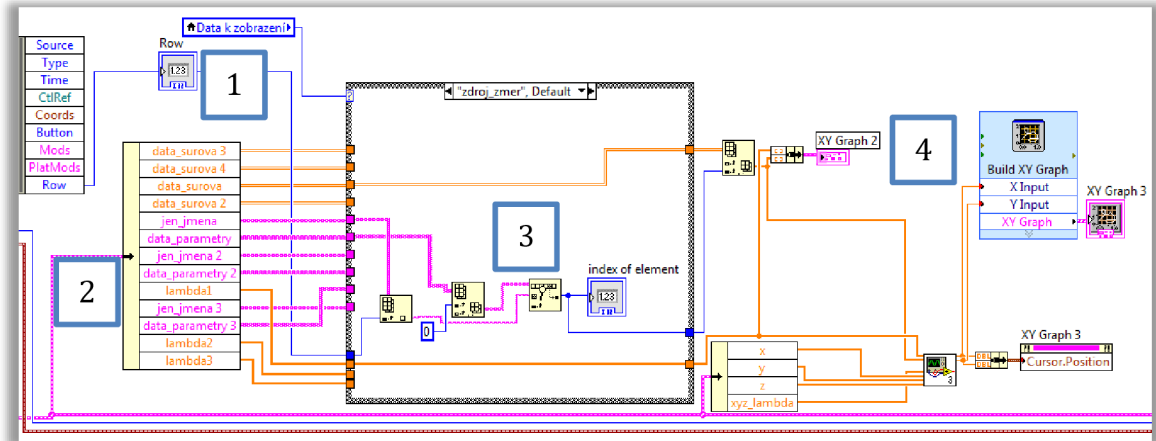
Dále následuje 15. strana struktury *event*, a to okno se jménem *Data k zobrazení*.



Obrázek 49 Struktura *event* – *Data k zobrazení*

1. Tento bod je jenom o výběru, která data se mají zobrazit. Rozhoduje o tom uživatel a na základě volby jsou pak vybrané příslušné tabulky.
2. Na základě toho, kterou tabulku si uživatel vybral k zobrazení má k dispozici i operační možnosti. Pokud si totiž vybral například jednu z tabulek s průměry, bude tam vždy jen jeden záznam a nemá cenu mu nabízet tabulku se seznamem změřených zdrojů. Dále samozřejmě nemá cenu počítat z jednoho záznamu aritmetický průměr. Je také zakázáno záznam mazat. To proto, že když chce uživatel spočítat nový průměr, tak se ten starý jednoduše přepíše.
3. Hodnoty jsou následně zobrazeny v příslušných grafech. Jsou spočítány hodnoty xy a jsou zobrazeny v kolorimetrickém trojúhelníku. Spektrum je do grafu vyneseno přímo na výstupu ze struktury *case*.

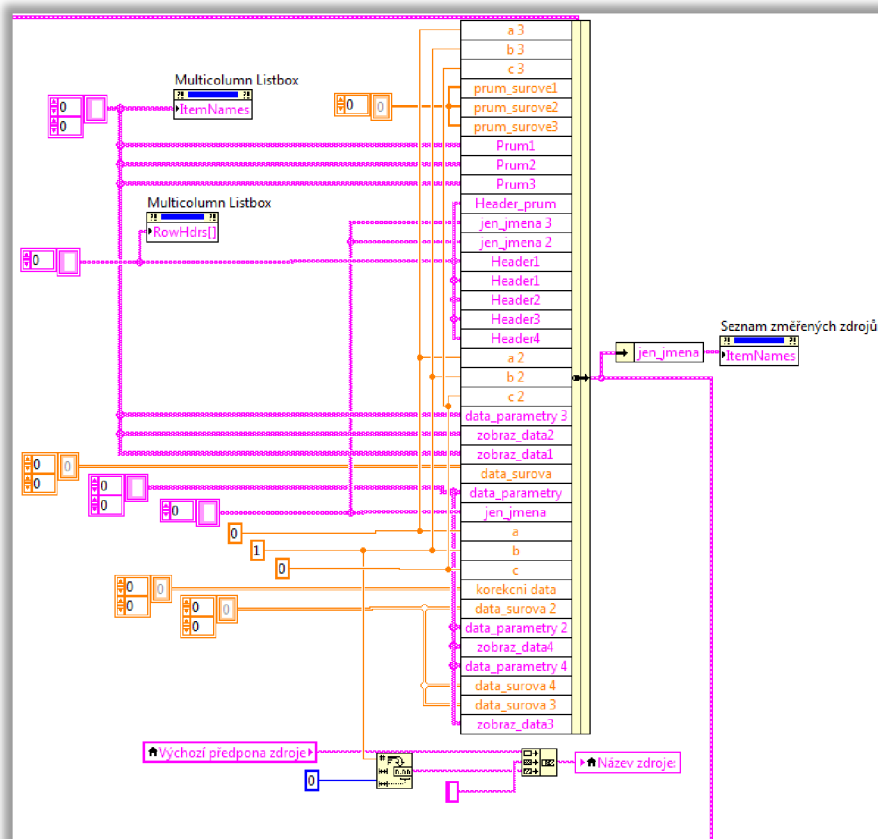
Další strana struktury *event* má číslo 16 a jmenuje se *Seznam změřených zdrojů*. Provádí akci při dvojkliku myši na změřený zdroj. Jednak nám udá jeho index a také vybírá hodnoty, které se zobrazí v grafech.



Obrázek 50 Struktura *event* – *Seznam změřených zdrojů*

1. Tato proměnná je využívána při mazání jednotlivých záznamů. Při dvojkliku na vybraný zdroj se zde uloží jeho index v tabulce.
2. Zde se nachází rozklad datového clusteru na dílčí tabulky. Z těch jsou v dalším bodě vybírány ty, které mají být zobrazeny
3. Zde jsou tedy na základě toho, jakou tabulku má uživatel zobrazenou vybírána data, která mají být zobrazena v grafech.
4. Tento bod obstarává zobrazení proměnných do grafů.

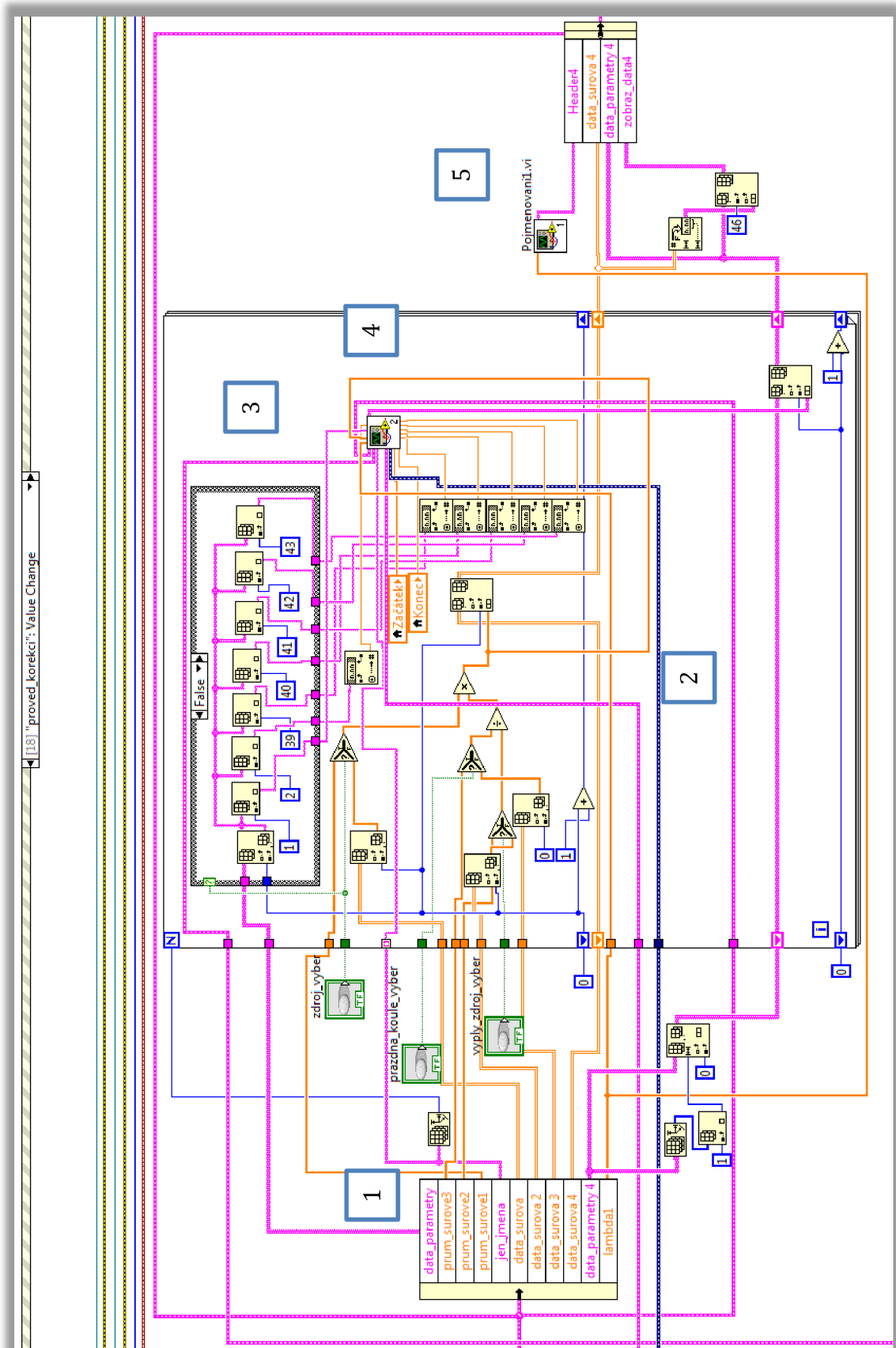
Další na řadě je strana struktury *event vymaz_data*. Ta celkem logicky slouží k vymazání všech dosud uložených dat a je nutné ji provést vždy před započítím nového měření, kdy chceme např. změnit rozsah vlnových délek.



Obrázek 51 Struktura event – vymaz_data

V této struktuře pouze dochází k přepsu tabulek a proměnných prázdnými hodnotami. Prázdné hodnoty jsou také odeslány na výstupy tabulek.

Další na řadě je velmi důležité okno *proved_korekci*, které jak název napoví vezme změřené hodnoty a na základě uživatelsky navolených parametrů vypočítá zdroj či zdroje již korigované na neaktivní plochy.



Obrázek 52 Struktura event – proved_korekci

Rozbor obrázku 52:

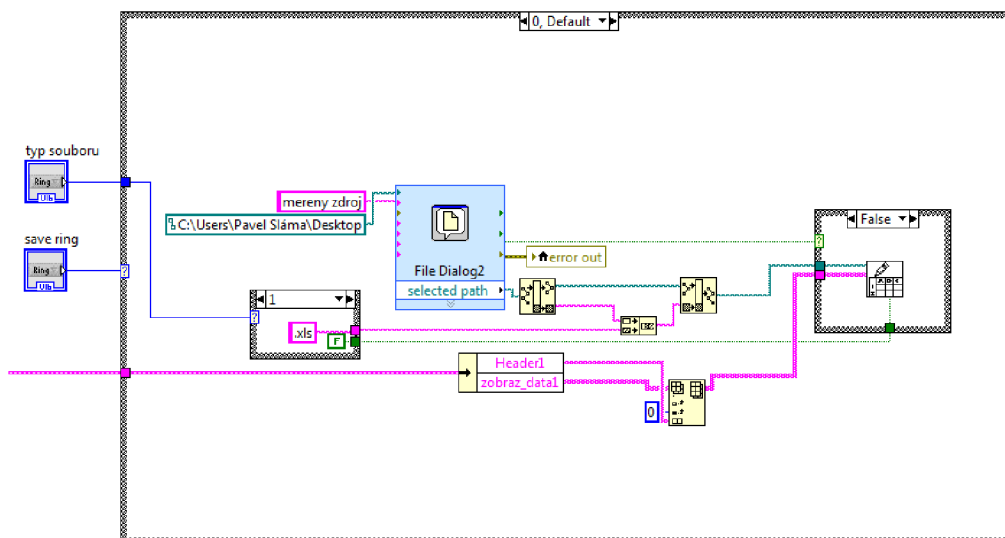
1. Zde se nachází rozdělení clusteru na jednotlivé tabulky. Podle toho, kolik zdrojů je změřeno bude tolikrát provedena následující *for* struktura.
2. V této struktuře *for* jsou vybírány vstupy na základě toho, jaké parametry korekce si uživatel zvolí. je vždy možno si vybrat, zda chceme počítat s průměrnými hodnotami, nebo ne. Pokud se nepočítá s průměrnými hodnotami, nachází se zde několik bloků *Index array*, které slouží k výběru příslušných sloupců z jednotlivých tabulek. Následně proběhne výpočet korekce dle následujícího vzorce:

$$L(\lambda)_{po\ korekci} = L(\lambda)_{měřený\ zdroj} \cdot \frac{L(\lambda)_{zdroj\ vyjmutý,korekční\ zapnutý}}{L(\lambda)_{zdroj\ vypnutý,korekční\ zapnutý}}$$

kde $L(\lambda)$ je spektrální zářivý tok (W/nm)

3. Tato struktura *case* má za úkol na základě toho, jestli máme u měřeného zdroje zvoleno počítat korekci pro každou hodnotu nebo pro průměr, načíst parametry z tabulky měřených zdrojů, které je třeba zkopírovat, nebo je v případě průměru vyplnit prázdnými hodnotami
4. Zde se nachází varianta sdružovacího programu *sdruzeni3*. Ten nám vypočítá hodnoty parametrů pro tento zdroj dostupných a uloží je do tabulky.
5. Na tomto místě probíhá uložení názvů parametrů do tabulky a k naplnění dalších tabulek příslušících ke zdroji po provedení korekce.

Následuje strana struktury *event* pro ukládání dat do souboru.



Obrázek 53 Struktura *event* – výřez z okna *save_data*

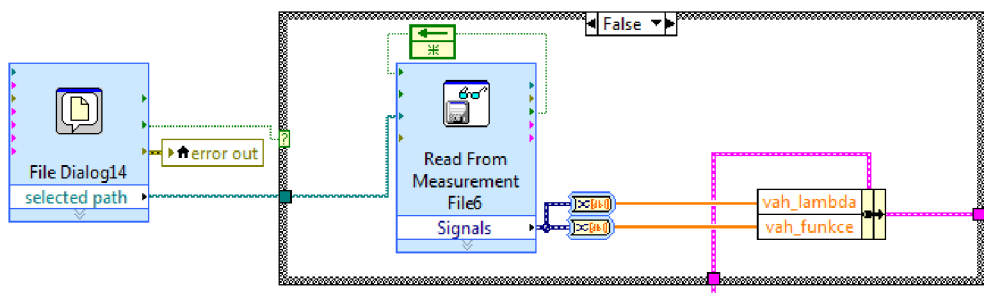
Zde se nachází struktura *case* se sedmi obrazovkami. Každá z nich připadá pro jednu z tabulek, které lze uložit do souboru. Na obrázku 53 je ukládání tabulky s měřeným zdrojem/zdroji. Pokud máme vybranou tabulku, vybereme si typ souboru. Máme na výběr soubor *.txt*, nebo *.xls*. U souboru *.xls* je třeba pamatovat na omezení množství dat, se kterými umí pracovat. Pro větší objemy dat je proto vhodnější použití formátu *.txt*.

Dialog pro výběr uložení má nastavenou jako výchozí cestu plochu počítače a jako výchozí název *měřený zdroj*. Nastavení přípony je obstaráváno strukturou *case* níže. Tato přípona se vloží do textového řetězce názvu souboru poté, co potvrdíme místo uložení. Pokud bychom místo tlačítka uložení zmáčknuli v dialogovém okně tlačítko zrušit, následuje struktura *case*, která nám podle toho zpřístupní blok zápisu do souboru, nebo ho naopak nezpřístupní.

Po bodech si opět provedeme rozbor obrázku:

1. Zde se nachází dialogové okno pro výběr souboru. Už bylo rozebráno výše, že je potřeba obstarat chování programu v případě že je výběr souboru zrušen. Na obrázku je vnitřní podmínka *case* ve stavu *false*, tedy výběr nebyl zrušen. Kdyby byl zrušen, tak je podmínka *case* ve stavu *true* a signál jí jenom projde bez jakýchkoli úprav.
2. Zde je samotný algoritmus pro načítání ze souboru. Množství bloků, které jsou zde k vidění je způsobeno tím, že potřebujeme tabulku „rozsekat“ na jednotlivé kusy. Například potřebujeme načíst samotné názvy zdrojů, pouze jejich parametry či proměnné, které nám udávají počet zdrojů. Z těch poté také zjistíme, jak se bude jmenovat další zdroj, který bychom chtěli po načtení dále měřit (za předpokladu, že si ho nepojmenujeme sami).
3. Tato část má za úkol zobrazit načtená data v tabulce, jestliže máme příslušnou tabulku zrovna vybránu k zobrazení.

Nyní je na řadě poslední obrazovka struktury *event Open Button*. Slouží k načtení převodní funkce příslušenství připojeného ke spektroradiometru. V práci se počítá s připojením ke kulovému integrátoru, jehož převodní funkce je zde také jako výchozí.



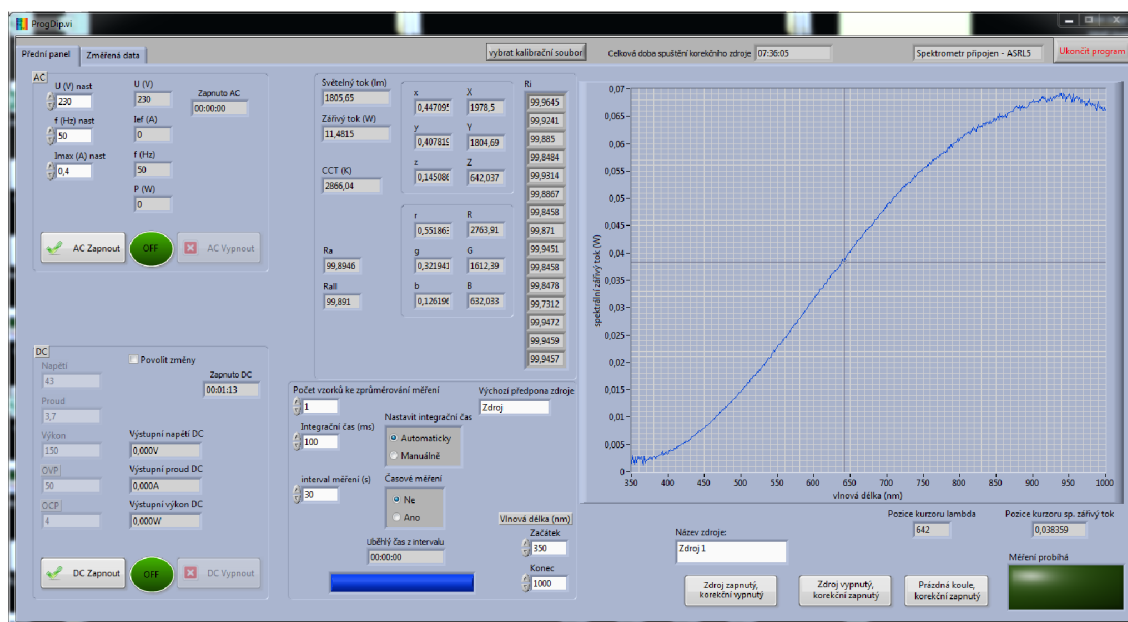
Obrázek 55 Struktura *event* - výřez okna *Open Button*

Blok s dialogovým oknem s připojenou strukturou *case* již čtenář vidí poněkoli káté, takže ji nemá cenu znovu rozebírat. Načítání souboru probíhá samostatným blokem a data jsou nahrána na příslušné místo v clusteru.

8 UŽIVATELSKÉ PROSTŘEDÍ PROGRAMU

Uživatelské prostředí programu je tvořeno dvěma hlavními záložkami. Ta první tvoří čelní panel virtuálního měřicího přístroje. Druhá je složena z tabulek a grafů se změřenými hodnotami.

8.1 Čelní strana programu



Obrázek 56 Přední panel programu

Na čelním panelu se po levé straně nachází okénka s ovládacími a indikačními prvky zdrojů. Když program zapneme, že jsou zdroje připojené poznáme podle toho, jestli se objevila ovládací tlačítka.

První vlevo nahoře je ovládání střídavého zdroje. Máme možnost si nastavit napětí, frekvenci a maximální výstupní hodnotu proudu. Z indikátorů je k dispozici Napětí, proud, výstupní frekvence, výkon a doba, které uběhla od zapnutí zdroje. Posledním indikátorem je indikátor zapnutí zdroje. Nachází se mezi tlačítky pro vypnutí a zapnutí zdroje. Při zapnutí se rozsvítí a popisek se změní z *OFF* na *ON*.

Podobné je okénko pro ovládání stejnosměrného zdroje, nacházející se vlevo dole. Hodnoty, které si můžeme nastavit jsou napětí, proud a výkon, na které je zdroj schopen regulovat. Dále si můžeme nastavit maximální hodnoty napětí a proudu, po jejichž překročení zdroj vypne výstup. Z obrázku je vidět, že hodnoty jsou zašedlé a tedy nedostupné. Od toho je vedle zaškrtačací políčko *Povolit změny*, které umožní zadat hodnoty. Ve výchozím stavu to tak není, protože je zde připojen stále stejný světelný zdroj, a to korekční žárovka. Dále jsou zde indikační kolonky napětí,

proudu, výkonu a také doba od zapnutí zdroje. Následujícím indikátorem je zde obdobně jako u AC zdroje indikátor zapnutí. Poslední indikátor je umístěn ve vrchní liště programu a jde o indikátor celkového času, po který byl korekční zdroj zapnut.

Následuje popis „světelné“ části předního panelu. Jako první po zapnutí programu je dobré sledovat okénko v pravém horním rohu. To nám řekne, jestli byl spekroradiometr zdárně připojen. Pakliže ne, program musíme ukončit a spustit znovu. To ostatně platí i pro připojení stejnosměrného a střídavého zdroje.

Když máme vše připojené, nastavíme parametry, které musejí být vybrány před začátkem měření a dále se do nich nezasahuje. Sem patří výběr kalibračního souboru, který se nachází ve vrchní liště programu. Ve výchozím stavu je soubor patřící ke kulovému integrátoru. Pro ten byl program přímo navržen a jiné soubory v něm nebyly testovány. Následuje předpona zdroje. Pokud se rozhodneme, že nechceme pojmenovávat jednotlivá měření zdroje, program generuje název automaticky, a to formou nějakého názvu a pořadového čísla. Ve výchozím stavu je tento název *Zdroj 1*, *Zdroj 2*, atd. Chce-li ale uživatel, může se rozhodnout tuto předponu změnit. Další parametr, který je třeba nastavit před započítáním měření je rozsah vlnových délek, pro který chceme měřit. Na obrázku je vidět maximální rozsah daný především rozsahem kalibračního souboru, který činí 350 - 1000 nm.

Dále jsme připraveni provést samotné měření. Program je stavěn na to, že umožňuje uživateli provádět korekci zdroje na neaktivní části. K tomu je však kromě změřením samotného měřeného zdroje také změřením korekce. Ta se sestává z měření, kdy je zdroj ponechán v integrátoru, ale je vypnut a korekční zdroj je zapnut. Další korekční měření je, kdy je zdroj z integrátoru vyjmut a je změřeno spektrum zapnutého korekčního zdroje. Tyto hodnoty je možné měřit v různém pořadí, popřípadě je možné si hodnoty nahrát ze souboru, o tom ale až později. Tlačítka se nacházejí pod grafem spektra. Vedle nich se nachází indikátor. Pokud spekroradiometr zrovna měří, tento indikátor se rozsvítí. Po jeho zhasnutí se ovšem výsledky nezobrazí okamžitě, neboť jejich zpracování je náročné a trvá několik sekund.

Pro každé měření si můžeme nastavit další parametry. Je to integrační čas, který si musíme přepnout z nabídky *Automaticky* na *Manuálně* a zadat hodnotu v milisekundách a také počet vzorků k zprůměrování měření, který využívá vestavěného algoritmu v přístroji a z nastaveného množství nám vypočítá výsledek jednoho měření.

Dále je možnost měřit automaticky s nastaveným časovým intervalem. To probíhá tak, že si zadáme délku intervalu v sekundách a v poli *Časové měření* zatrhneme položku *Ano*. Pak jen zmáčkneme tlačítko měření, které chceme momentálně provádět a další měření už bude provádět program automaticky. Nachází se zde ukazatel průběhu, který nám ukazuje, jaká část intervalu již uběhla a

políčko, které nám ukazuje uběhlý čas. Pokud chceme časové měření ukončit, tak v poli *Časové měření* přepneme opět na položku *Ne*, čímž je měření ukončeno. Pro zbylé druhy měření můžeme pokračovat obdobným způsobem.

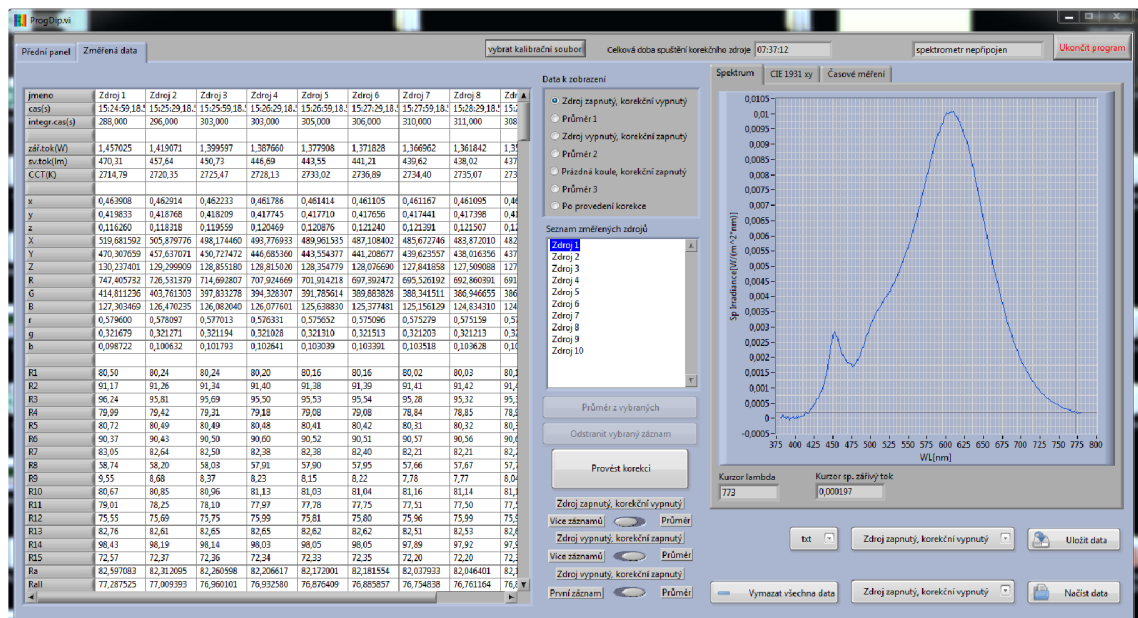
Zbývá nám popsat poslední část čelního panelu. Ta se nachází nahoře uprostřed a po každém měření nám ukazuje vypočítané parametry:

1. Světelný tok (lm)
2. Zářivý tok (W)
3. CCT – náhradní teplota chromatičnosti (K)
4. R_a – index podání barev
5. R_{all} – průměr všech patnácti speciálních indexů podání barev R_i
6. R_i – tabulka speciálních indexů podání barev
7. Trichromatické složky XYZ a souřadnice xyz
8. Trichromatické složky RGB a souřadnice rgb

Pod grafem spektrálního průběhu se ještě nachází okénka, které indikují polohu kurzoru v grafu. Ten můžeme posouvat tak, že ho uchopíme buď za jednu z os pro pohyb do stran, nebo za střed pro pohyb všemi směry.

8.2 Strana se změřenými daty

Následuje druhá strana programu. Zde můžeme pracovat s již změřenými daty.

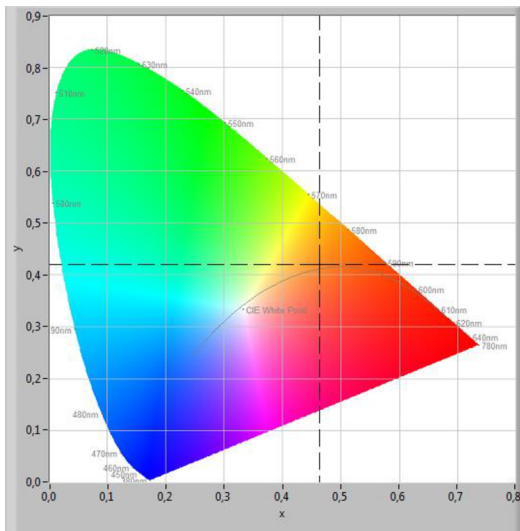


Obrázek 57 Strana se změřenými daty

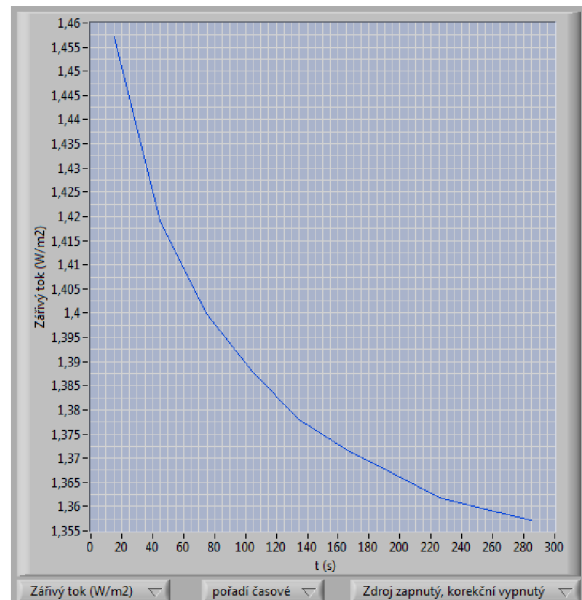
Při pohledu na graf je vidět, že měřený zdroj je teple bílá LED. Kromě grafu to poznáme i z tabulky, kde se teplota chromatičnosti pohybuje kolem 2730 K. Ano,

pohybuje se, a to proto, že měření započalo hned po zapnutí zdroje, takže se ještě neustálil. To doloží pohled na časový průběh, který bude následovat níže.

Podíváme se na další grafické zobrazení dat, a to trichromatický trojúhelník. K tomu se dostaneme přepnutím záložky přímo nad grafem a ukázkou vidíme na obrázku 58.



Obrázek 58 Trichromatický trojúhelník xy
(obrázek na pozadí převzat a upraven
z programu JETI LiVal)



Obrázek 59 Časová závislost zvolené veličiny

Na obrázku 59 následuje grafické zobrazení veličin v závislosti na čase, popř. na pořadí. To proto, kdyby se stalo, že by uživatel chtěl měřit zdroj napájený z jiného zdroje než toho propojeného s programem, neměl by zde onu časovou závislost, protože je použit čas od zapnutí zdroje. Uživatel má tedy na výběr. Vybere si, jakou veličinu chce zobrazit, vybere si, na čem má být tato veličina závislá a následně pro kterou tabulku si chce veličinu zobrazit.

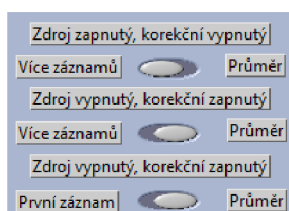
Tím se dostáváme k tabulkám. V panelu je tabulka pouze jedna, můžeme si do ní však zobrazit tabulek několik. O tom, jaká data budeme v tabulce vidět slouží okénko Data k zobrazení. Můžeme si vybrat:

1. Zdroj zapnutý, korekční vypnutý – To jsou data, která měříme, ovšem ještě před korekcí.
2. Zdroj vypnutý, korekční zapnutý – Tabulka nutná pro výpočet korekce, data, jenž obsahuje vyplývají z názvu.
3. Prázdná koule, korekční zapnutý – Jde tedy o druhou část hodnot nezbytných pro výpočet korekce.
4. Průměr 1,2,3 - Tyto tabulky slouží k zobrazení průměrných hodnot z tabulek výše. Každá z nich má jednu tabulku. My si pak můžeme vybrat, jakým způsobem budeme korekci počítat. O tom ovšem o něco níže.

Následuje okénko *Seznam změřených zdrojů*. To nám ukazuje názvy zdrojů, které už byly změřeny. Když si měníme zobrazenou tabulku, změní se i názvy v tomto okénku odpovídajícím způsobem. Právě pomocí tohoto okénka provádíme práci s tabulkami. Při dvojitým poklepu na jednu hodnotu dojde k zobrazení hodnoty v grafu. Dvojitý poklep též slouží k výběru hodnoty, pokud bychom ji chtěli smazat. To se provádí tlačítkem *Odstranit vybraný záznam*. Další činnost, kterou můžeme provádět v okně *Seznam změřených zdrojů* je označování hodnot, z kterých chceme spočítat průměr. Jak je vidno na obrázku 59 s časovým průběhem, parametry zdrojů se s časem mohou poměrně hodně měnit. Jakmile máme tedy změřeny dostatečné množství hodnot, můžeme si pro výpočet vybrat ty, při kterých je zdroj již ustálen a vyhneme se tak chybě. Výběr probíhá tak, že si v okně *Seznam změřených zdrojů* buď pouze klikáme na jednotlivé zdroje za držení klávesy *Ctrl*, nebo si označíme první záznam ze kterého chceme počítat a za držení klávesy *Shift* si označíme ten poslední a automaticky jsou označeny i všechny mezi. Následně zmáčkneme tlačítko *Průměr z vybraných* a v příslušné tabulce *Průměr 1/2/3* už si můžeme zobrazit vypočtené hodnoty. V tabulkách s průměry je okno *Seznam změřených zdrojů*, tlačítko *Průměr z vybraných* a tlačítko *Odstranit označený záznam* neaktivní, neboť pro ně zde není užitku.

Uživatel má dále možnost *Vymazat všechna data*, která odstraní obsah všech tabulek a v podstatě by měla být provedena pokaždé, kdy uživatel provede uložení změřených dat neboli má zdroj změřený a chce začít měřit nějaký další.

Následuje tlačítko *Provést korekci* a přepínače pod ním. Nejdříve k přepínačům. Ty slouží k nastavení toho, z jakých dat budou korigované hodnoty vypočteny. Výchozí stav je viditelný na obrázku 60. Je to konfigurace, za jaké je výpočet prováděn nejčastěji. To znamená, že chceme mít ve výsledku všechny hodnoty změřeného zdroje, ale korekční hodnoty chceme mít průměrné a vybrané po ustálení korekčního zdroje.



Obrázek 60 Nastavení parametrů pro provedení korekce

Tato nabídka je zde pro zvláštní případy. Pokud bychom třeba chtěli mít pouze jeden průměrný výsledek za všechny, přepneme první přepínač do polohy *Průměr*. Ještě méně častý případ by byl, kdybychom měřili více různých zdrojů,

přičemž každý bychom měřili pouze jednou. Když zajistíme, aby jednomu záznamu s měřeným zdrojem odpovídal záznam s tímto zdrojem vypnutým a korekčním zapnutým, lze tento výpočet provést. Poslední případ je opět neobvyklý, protože by znamenal, že se spokojíme s jedním záznamem při vypnutém měřeném zdroji a korekčním zdroji zapnutém. V podstatě by se dalo říct, že tato alternativní nastavení nebudou pravděpodobně využívána, ale uživatel i tak dostává možnost volby.

Následuje tedy tlačítko *Provést korekci*. Po jeho stisknutí jsou dopočítány výsledky do poslední tabulky Po provedení korekce.

Další možností, kterou uživatel má je uložení hodnot. To probíhá pro každou tabulku zvlášť a je prostřednictvím dialogového okna. Typ souboru, jaký chce uživatel uložit nevybírání v dialogovém okně, ale v nabídce vedle samotného tlačítka *Uložit data*. Má na výběr mezi *.txt* a *.xls*. Formát souboru MS Excel je vybrán proto, že LabView umožňuje převod textu přímo do něj. Například u formátu *.csv* toto bohužel není možné.

Poslední nabídkou je načtení souborů. To umožňuje uživateli načíst si soubory, které už byly tímto programem vytvořeny, a to ve formátu *.txt*. To může chtít provést buď z toho důvodu, že chce pokračovat v započaté práci, nebo proto, že chce znova měřit zdroj, pro který již má korekci. Potom je tu ještě jedna možnost, neboť měření při prázdném kulovém integrátoru a zapnutém korekčním zdroji vyjde vždy stejně. Při provedení tohoto kroku si tedy uživatel usnadní práci.

Je třeba ještě dodat, že funkce načítání souborů se musí provádět vždy před započítím měření. Nelze také provést více nahrávání do jedné tabulky za sebou.

9 ZÁVĚR

Tato práce měla dle zadání za cíl tvorbu programu v LabView pro měření spektra a výpočet světelných parametrů. Zadání je v tomhle ohledu ovšem zavádějící, neboť cíle byly poněkud komplexnější. Program měl umožnit komplexní měření v kulovém integrátoru, což obnáší i ovládání stejnosměrného a střídavého zdroje, které napájí světelné zdroje do integrátoru umístěné a také korekční zdroj, který je v integrátoru umístěn trvale. Program má tedy za cíl zajistit uživateli lepší komfort sdružením ovládacích prvků na jednom místě, a to na obrazovce počítače. Dalším cílem, který měl program splnit je poskytnout vyšší transparentnost měřených dat. Do teď probíhalo měření se spektrometrem v programu JETI LiVal, což je program přibalený výrobcem tohoto zařízení. Sice nabízí uživateli možnost měřit v kulovém integrátoru a provádět korekci na neaktivní části svítidla, tato korekční data si však již uživatel nikde nezobrazí a nemá jistotu, že je korekční zdroj v době měření v ustáleném stavu. V navrhovaném programu tedy měla být možnost si všechna data zobrazit a vybrat si, ze kterých hodnot chce korekci počítat.

Takto tedy víceméně znělo zadání po domluvě s vedoucím práce. V semestrálním projektu byl vytvořen teoretický úvod do problematiky, bylo nastudováno, jakým způsobem probíhá výpočet požadovaných parametrů, a hlavně, bylo zajištěno, aby všechny přístroje komunikovaly s počítačem. Zároveň také probíhalo studium programu LabView samotného, neboť první kontakt autora s tímto programem byl navázán právě při řešení této práce.

Diplomová práce pak měla za cíl tento program naprogramovat. Už při samotném ovládání napájecích zdrojů však došlo problémům, neboť při umístění ovládacích bloků do jednoho programu docházelo při spuštění vždy k odpojení jednoho ze zdrojů. To bylo nakonec vyřešeno tak, že každý ze zdrojů má svůj vlastní ovládací program a s hlavním programem tyto programy komunikují přes vrstvu Windows – ActiveX. Ovládací programy zdrojů pak byly „zneviditelněny“, aby veškeré ovládání probíhalo v jednom okně.

Další výraznou překážkou bylo, že programové bloky dodané výrobcem spektrometru neumožňují změnu používaného kalibračního souboru v paměti přístroje. Tyto soubory musí být v paměti pro každou používanou periférii, která nějakým způsobem ovlivní spektrum. Může jít o kosinový nástavec nebo kulový integrátor, což je náš případ. Naštěstí tento kalibrační soubor byl dostupný ve formě textového souboru, takže problém byl vyřešen načtením tohoto souboru a manuálním přepočtem z hodnot A/D převodníku přístroje. Toto má však svoje negativum, neboť to způsobí prodloužení měření o jistý „postprocessing“ signálu, tedy manuální výpočet, který jinak probíhá přímo v přístroji.

Po vyřešení těchto a několika dalších, menších problémů byl tedy program vytvořen a zprovozněn. Má formu .exe souboru a umožňuje měření v kulovém integrátoru včetně provedení korekce.

V programu proběhla řada pokusných měření za účelem zjištění správnosti výsledků, ať už spektrálních dat či vypočtených parametrů. Ověřování správnosti výpočtu změřených dat probíhalo ve volně šiřitelném programu ColorCalculator od společnosti OSRAM. Program totiž umožňuje nahrát již změřené spektrum a vypočítá z něj světelné parametry. Výsledky byly porovnány s výsledky programu vytvořeného v této práci a všechny se shodovaly, ovšem s tím, že program ColorCalculator více zaokrouhloval.

Správnost změřeného spektra byla zjišťována porovnáváním vypočtených parametrů s parametry předpokládanými či udávanými výrobcem. Srovnávání spektra změřeného programem vytvořeným v této práci a spektra změřeného v programu LiVal není zcela vypovídající, neboť měření není možné provést oběma programy zároveň a zajistit tak relevantnost výsledků, neboť při měření s odstupem dojde vlivem zahřívání k změně parametrů zdroje. Roli zde hraje také šum či nejistota měření.

Protože se do budoucna počítá s rozšířením programu o další funkce, byly ze strany vedoucího kladeny požadavky na podrobnější popis/rozbor struktury programu. Tomu jsou proto věnovány kapitoly 6 a 7, které jsou zde tedy z důvodu snadnější orientace pro případné další programátory.

Literatura

- [1] HABEL, J. A KOL., *Světelná technika a osvětlování*, FCC Public, Praha 2013, 624 s, ISBN 987-80-86534-21-3
- [2] JETI Technische Instrumente GmbH., *Basics of Spectral Measurement*, Jena, 2005, 53 s
- [2.1] HABEL, J., ŽÁK, P., Elektrické světlo 1, Kurz ČVUT FEL, 2015, Dostupné z <https://www.powerwiki.cz/wiki/A5M15ES1>
- [3] WIKIPEDIA, *Spectrophotometry*, Wikimedia Foundation, 2016, Dostupné z <https://en.wikipedia.org/wiki/Spectrophotometry>
- [4] NATIONAL INSTRUMENTS, NI-VISA Overview, ni.com, 2009, Dostupné z <http://www.ni.com/tutorial/3702/en/>
- [5] NOVÁK, T., *Projekce a konstrukce vyhrazených technických zařízení elektro II*, Ostrava, 2014, 232 s, ISBN 978-80-248-3536-5
- [6] JETI Technische Instrumente GmbH, Short operating instructions, Jena, 2015, 69 s
- [7] VLACH, J., HAVLÍČEK, J., VLACH, M., *Začínáme s LabVIEW*, BEN-Technická literatura, Praha 2008, 86 s, ISBN 978-80-7300-245-9
- [8] APT, 300XAC Series operation manual, Illinois, 2016, 151 s, Dostupné z <http://www.aptsources.com/products/300xac/310xac.aspx#>
- [9] WIKIPEDIA, *Integrating spheres*, Wikimedia Foundation, 2016, Dostupné z https://en.wikipedia.org/wiki/Integrating_sphere
- [10] MCKEE, G., *Integrating Spheres: Collecting and Uniformly Distributing Light*, Labshire inc., 2015, Dostupné z <http://www.photonics.com/EDU/Handbook.aspx?AID=25122>
- [11] ELEKTRO-AUTOMATIK, PSI5000 datasheet, Elektroautomatik, 2016, Dostupné z http://shop.elektroautomatik.de/shop/dos2/Web/Datenblaetter/datasheet_psi5000.pdf
- [12] SIEGMANN, A. E., *Lasers*, University science books California, 1986, 1283 s, ISBN 0-935702-11-3
- [13] DUŠEK, J., Zelené a UV lasery, MEGABLOG, 2012, Dostupné z <http://www.mega-blog.cz/lasery/zelene-a-uv-lasery/>
- [14] NATIONAL INSTRUMENTS, GPIB Interface, Direct Industry, 2016, Dostupné z <http://www.directindustry.com/prod/national-instruments/product-5074-38868.html>
- [15] ROBERTSON A. R., *Computation of Correlated Color Temperature and Distribution temperature*, JOSA, 1968, DOI 10.1364/JOSA.58.001528
- [16] ZAIDI H., ZAIDI Q., Colour constancy in context: Roles for local adaptation and levels of reference, *Journal of Vision*, 2004, Dostupné z <http://jov.arvojournals.org/article.aspx?doi=10.1167/4.9.3>

- [17] NATIONAL INSTRUMENTS, Internetové komunitní fórum o Labview, dostupné z <http://forums.ni.com/t5/Discussion-Forums/ct-p/discussion-forums>
- [18] JK CONSULTING, Color Rendering Index (CRI), dostupné z <http://www.kruschwitz.com/cri.htm>
- [19] NATIONAL INSTRUMENTS, *How Do I Register ActiveX Controls, ActiveX Servers, and Type Libraries?*, 2016, dostupné z <http://digital.ni.com/public.nsf/allkb/4F811A9B23F1D46E862566F700615B7A>

Seznam symbolů, veličin a zkratek

λ	Vlnová délka záření	(nm)
PFC	Power Factor Correction – úprava průběhu odebíraného proudu na průběh harmonický	
LED	Light Emitting Diode – světlo emitující dioda	
CIE	Commission Internationale de l'Éclairage – mezinárodní komise pro osvětlování	
VIS-IR	Visible-Infrared – Viditelné a infračervené záření	
UV-VIS	Ultraviolet-Visible – Ultrafialové a viditelné záření	
CCD	Charge Coupled Device – zařízení s vázanými náboji	
CMOS	Complimentary Metal Oxide Semiconductor – doplňující se kov-oxid-polovodič	
VI	Virtual Instrument – virtuální přístroj	
Integer	Datový typ o 32 bitech	
<i>Floating point</i>	Datový typ o 64 bitech s přesností max. 15 desetinných míst	
String	Datový typ textového řetězce	
True	Stav booleovského datového typu odovídající logické 1	
False	Stav booleovského datového typu odovídající logické 0	
AC	Alternating current – střídavý proud	
DC	Direct current – stejnosměrný proud	

Seznam příloh

Příloha 1. CD