

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA ELEKTROENERGETIKY A EKOLOGIE**

# **DIPLOMOVÁ PRÁCE**

**System pro kontinuální měření a analýzu dat  
z fotovoltaické elektrárny**

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**  
**Fakulta elektrotechnická**  
Akademický rok: **2017/2018**

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

Jméno a příjmení: **Bc. Martin STEINBACH**  
Osobní číslo: **E15N0117P**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Elektroenergetika**  
Název tématu: **System pro kontinuální měření a analýzu dat z fotovoltaické elektrárny**  
Zadávající katedra: **Katedra elektroenergetiky a ekologie**

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

1. Vysvětlete obecně problematiku měření fotovoltaických systémů.
2. Seznamte se s vývojovým prostředím LabVIEW.
3. V prostředí LabVIEW realizujte systém pro měření a analýzu dat z fotovoltaické elektrárny.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **40 - 60 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

**1. Přednášky KEE/SOES, KEE/POE**

Vedoucí diplomové práce: **Ing. Milan Bělík, Ph.D.**

Katedra elektroenergetiky a ekologie

Datum zadání diplomové práce: **10. října 2017**

Termín odevzdání diplomové práce: **24. května 2018**

Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan



Doc. Ing. Karel Noháč, Ph.D.  
vedoucí katedry

V Plzni dne 14. října 2016

## **Abstrakt**

Předkládaná diplomová práce je zaměřena na realizaci systému umožňujícího kontinuální měření a analýzu dat z fotovoltaické elektrárny s využitím vývojového prostředí LabVIEW. V úvodu práce jsou popsány elektrické parametry fotovoltaických článků, matematické modely využívané pro jejich simulaci a standardní testovací podmínky. Zabývá se obecnou problematikou měření fotovoltaických elektráren, přičemž popisuje současné metody používané k jejich diagnostice. Prostřední část práce se věnuje seznámení čtenáře s vývojovým prostředím LabVIEW pro snadnější pochopení závěrečného bodu práce, ve kterém je řešena jeho samotná realizace, a kde jsou popisovány části vytvořeného programu společně s jeho ovládáním.

## **Klíčová slova**

DAQ, fotovoltaický článek, LabVIEW, National Instruments, NI PCI-6221, STC

## **Abstract**

This diploma thesis is focused on realization of a system enabling continuous measurement and analysis of data from a photovoltaic power plant using the LabVIEW development environment. At the beginning of the thesis are described electrical parameters of photovoltaic cells, mathematical models used for their simulation and standard test conditions. It deals with the general issue of measuring photovoltaic power plants, describing the current methods used for their diagnosis. The middle part of the thesis is devoted to familiarizing the reader with the LabVIEW development environment for easier understanding of the final point of the thesis, in which it is solved its own realization and where parts of the program are described together with its control.

## **Keywords**

DAQ, photovoltaic cell, LabVIEW, National Instruments, NI PCI-6221, STC

## **Prohlášení**

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

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

.....

podpis

V Plzni dne 24.5.2018

Bc. Martin Steinbach

## **Poděkování**

Tímto bych rád poděkoval vedoucímu mé diplomové práce Ing. Milanu Bělíkovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce. Též bych rád poděkoval Ing. Karlu Kalistovi za ochotu a pomoc při tvorbě programu v LabVIEW.

# Obsah

<b>OBSAH</b> .....	<b>8</b>
<b>ÚVOD</b> .....	<b>10</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>11</b>
<b>1 PROBLEMATIKA MĚŘENÍ FOTOVOLTAICKÝCH SYSTÉMŮ</b> .....	<b>13</b>
1.1 ELEKTRICKÉ PARAMETRY FOTOVOLTAICKÉHO ČLÁNKU.....	14
1.1.1 Napětí naprázdno .....	14
1.1.2 Zkratový proud .....	14
1.1.3 Bod maximálního výkonu.....	15
1.1.4 Činitel zaplnění.....	15
1.1.5 Proud $I_{450}$ .....	16
1.1.6 Účinnost.....	16
1.2 MODELÝ FOTOVOLTAICKÉHO ČLÁNKU.....	19
1.2.1 Popis nejpoužívanějších modelů.....	19
1.2.2 Model FV článku s jednou diodou.....	21
1.3 STANDARDNÍ ZKUŠEBNÍ PODMÍNKY .....	23
1.4 DIAGNOSTICKÉ METODY FVE.....	23
1.4.1 Termografie .....	24
1.4.2 Měření V-A charakteristik .....	26
1.4.3 FLASH test.....	27
1.4.4 Elektroluminiscence.....	28
<b>2 VÝVOJOVÉ PROSTŘEDÍ LABVIEW</b> .....	<b>31</b>
2.1 LABVIEW – NÁSTROJ VIRTUÁLNÍ INSTRUMENTACE.....	31
2.2 ÚVODNÍ OBRAZOVKA LABVIEW .....	32
2.3 ČELNÍ PANEL (FRONT PANEL) .....	33
2.3.1 Obrazovka čelního panelu.....	33
2.3.2 Nástrojová lišta čelního panelu.....	34
2.3.3 Prvky čelního panelu .....	35
2.4 BLOKOVÝ DIAGRAM (BLOCK DIAGRAM).....	37
2.4.1 Obrazovka blokového diagramu.....	37
2.4.2 Nástrojová lišta blokového diagramu.....	38
2.4.3 Prvky blokového diagramu.....	40
2.4.4 Ikona a konektor VI.....	41
2.5 PALETY .....	42
2.5.1 Paleta Tools.....	42
2.5.2 Paleta Controls.....	43
2.5.3 Paleta Functions.....	44
2.6 DATOVÝ TOK (DATA FLOW).....	45
2.6.1 Datové typy.....	46
2.7 ZÍSKÁVÁNÍ DAT (DATA ACQUISITION) .....	48
2.7.1 Senzory .....	49
2.7.2 DAQ zařízení .....	49
2.7.3 Počítač v DAQ systému .....	50
2.8 NI PCI-6221/CB-68LP ACADEMIC STARTER KIT.....	51
2.8.1 Multifunkční karta NI PCI-6221.....	52
2.8.2 Konektorový blok CB-68LP.....	54
<b>3 REALIZACE SYSTÉMU PRO MĚŘENÍ A ANALÝZU DAT</b> .....	<b>55</b>
3.1 UŽIVATELSKÉ PROSTŘEDÍ.....	55
3.2 OVLÁDÁNÍ PROGRAMU .....	56
3.2.1 Nastavení parametrů .....	56
3.2.2 Ukládání dat .....	56



3.2.3	Zobrazení dat.....	57
3.2.4	Běh DAQ.....	58
3.3	POPIS ČÁSTÍ PROGRAMU .....	59
3.3.1	Počáteční inicializace při spuštění VI .....	59
3.3.2	Programové mazání grafů.....	60
3.3.3	DAQ multifunkční kartou.....	61
3.3.4	Interval záznamu a ukládání.....	63
3.3.5	Zobrazování a zápis dat do souboru.....	63
3.3.6	Zastavení DAQ, zobrazování a ukládání.....	66
3.3.7	Načtení a zobrazení dat ze souboru.....	67
<b>ZÁVĚR.....</b>		<b>69</b>
<b>SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ.....</b>		<b>70</b>
<b>SEZNAM PŘÍLOH.....</b>		<b>74</b>

## Úvod

Výkon fotovoltaické elektrárny ovlivňuje celá řada faktorů, zejména klimatické podmínky, které mohou být i velmi lokalizované, použitá technologie, kvalita výstavby a údržby, vyšší prašnost či jiné lokální znečištění fotovoltaických panelů. Výkon elektráren ve stejné lokalitě se tak může lišit v řádu několika procent.

Předkládaná práce je zaměřena zejména na vytvoření programu ve vývojovém prostředí LabVIEW, který by měl být schopen kontinuálně měřit a analyzovat data z fotovoltaické elektrárny.

První kapitola práce se zaměřuje na problematiku fotovoltaických systémů z hlediska měření. V rámci této kapitoly jsou nejprve zmíněny a popsány elektrické parametry fotovoltaických článků a vlivy ovlivňující jejich účinnost. Jsou zde také zmíněny nejpoužívanější matematické modely, jenž tvoří základ měřicích přístrojů používaných k diagnostice. Popisuje standardní testovací podmínky a rozebírá současné diagnostické metody fotovoltaických elektráren se snahou zohlednit moderní trendy.

Druhá kapitola tvoří teoretický základ pro pochopení praktické části práce. Obsahuje seznámení se základy vývojového prostředí LabVIEW, a také seznamuje se základy získávání dat (DAQ). Jsou zde uvedeny vlastnosti a parametry multifunkční karty NI PCI-6221, která byla poskytnuta vedoucím mé práce pro možnost realizace testovacího měření.

Třetí kapitola práce se týká praktické části, tedy samotného vývoje programu. V úvodní části do kapitoly jsou popsány vlastnosti a možnosti vytvořeného programu. Dále je popsáno uživatelské prostředí a způsob ovládání jeho jednotlivých funkcí. Rozsáhlou podkapitolou je pak podrobný popis nejdůležitějších dílčích částí, ze kterých je program složen.

## Seznam symbolů a zkratk

$A_c$ (m <sup>2</sup> )	.....	Plocha fotovoltaického článku
$AM$ (-)	.....	Koeficient atmosférické masy ( <i>Air Mass Index</i> )
$e$ (C)	.....	Elementární náboj ( $e = 1,602 \times 10^{-19}$ C)
$E$ (W/m <sup>2</sup> )	.....	Intenzita záření
$FF$ (-, %)	.....	Činitel zaplnění ( <i>Fill Factor</i> )
$I$ (A)	.....	Elektrický proud generovaný fotovoltaickým článkem
$I_0$ (A)	.....	Saturační proud
$I_{450}$ (A)	.....	Proudu při napětí 450 mV
$I_D, I_{D1}, I_{D2}$ (A)	.....	Elektrický proud protékající diodou
$I_{MPP}$ (A)	.....	Proud při max. výkonu
$I_p$ (A)	.....	Elektrický proud protékající přes paralelně zapojený rezistor
$I_{ph}$ (A)	.....	Fotoelektrický proud
$I_{SC}$ (A)	.....	Zkratový proud ( <i>Short-Circuit Current</i> )
$k$ (J·K <sup>-1</sup> )	.....	Boltzmannova konstanta ( $k = 1,38 \times 10^{-23}$ J·K <sup>-1</sup> )
$n$ (-)	.....	Činitel ideálnosti diody
$P_{MPP}$ (W)	.....	Výkon v bodě max. výkonu
$R_p$ (Ω)	.....	Paralelní odpor
$R_{ph}$ (Ω)	.....	Fiktivní odpor fotovoltaiky
$R_s$ (Ω)	.....	Sériový odpor
$T$ (K)	.....	Teplota P-N přechodu
$T_{cell}$ (K)	.....	Teplota článku
$U$ (V)	.....	Elektrické napětí na výstupních svorkách fotovoltaického článku
$U_{MPP}$ (V)	.....	Napětí při max. výkonu
$U_{OC}$ (V)	.....	Napětí naprázdno ( <i>Open-Circuit Voltage</i> )
$U_T$ (V)	.....	Teplotní napětí
$\eta$ (-, %)	.....	Účinnost

CCD.....	Technologie obrazového snímače ( <i>Charged Coupled Device</i> )
D, D <sub>1</sub> , D <sub>2</sub> .....	Dioda
DAQ .....	Pořizování a práce s daty ( <i>Data Acquisition</i> )
DSP.....	Zpracování digitálních signálů ( <i>Digital Signal Processing</i> )
ELCD.....	Test využívající elektroluminiscenci ( <i>Electroluminescence Crack Detection</i> )
FVČ .....	Fotovoltaický článek
FVE.....	Fotovoltaická elektrárna
GPIB .....	Rozhraní pro měřicí a zkušební přístroje ( <i>General Purpose Interface Bus</i> )
LabVIEW.....	Laboratorní pracoviště virtuálních přístrojů ( <i>Laboratory Virtual Instruments Engineering Workbench</i> )
MPP .....	Bod maximálního výkonu ( <i>Maximum Power Point</i> )
NI.....	Společnost National Instruments
NIR .....	Blízké infračervené záření ( <i>Near-Infrared</i> )
PCI.....	Počítačová sběrnice pro připojení periférií k základní desce ( <i>Peripheral Component Interconnect</i> )
PID.....	Fyzikální jev způsobující pokles výkonu FV panelů kvůli rozdílu potenciálů vůči zemi ( <i>Potential Induced Degradation</i> )
RTSI .....	Specializovaná vysokorychlostní digitální sběrnice ( <i>Real-Time System Integration</i> )
STC.....	Standardní zkušební podmínky ( <i>Standard Test Conditions</i> )
SubVI.....	Podprogram virtuálního přístroje ( <i>Subroutine Virtual Instrument</i> )
VI.....	Virtuální přístroj ( <i>Virtual Instrument</i> )
Wp .....	Špičkový výkon ( <i>Watt-peak</i> )

# 1 Problematika měření fotovoltaických systémů

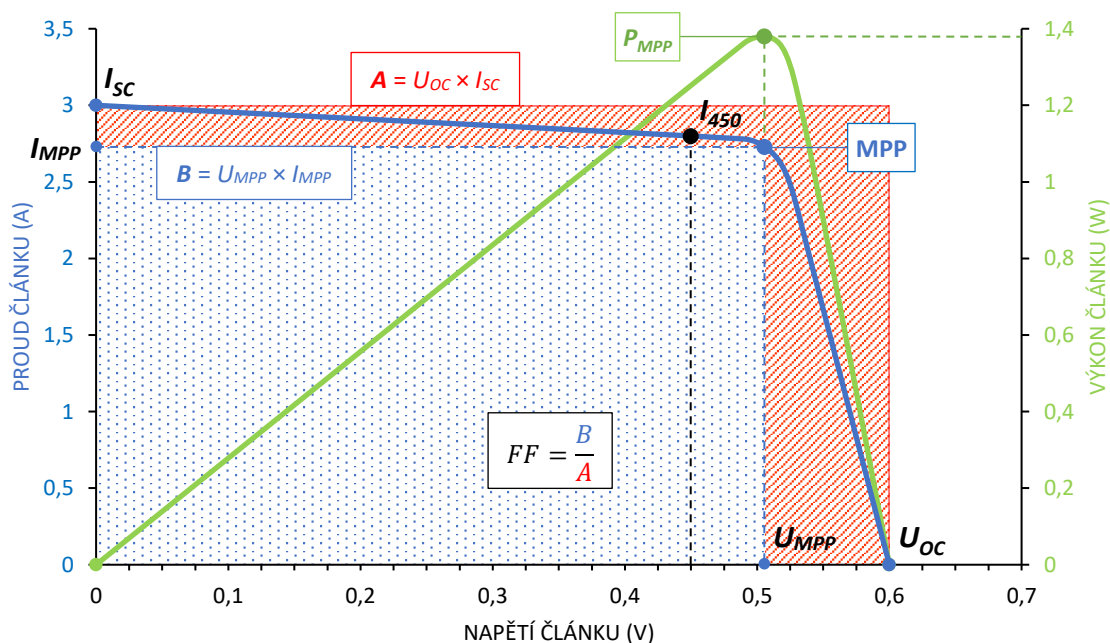
Praxe ukazuje, že systémy fotovoltaických elektráren (FVE) překvapivě celkem často vykazují různé poruchy a odchylky od běžného normálu. V případě nových FVE je většina snadno průkazných poruch zatím kryta zárukou technologie. [1]

Otázkou tedy je, jak zjistit poruchy malé a na první pohled nezřetelné, které ale přesto snižují výkon celé FVE. Zde je nutno podotknout, že žádná FVE nepodává a ani nemůže podávat 100 % svého nominálního výkonu. Důvodem pro tuto skutečnost jsou částečně poškozené fotovoltaické panely, chyby v zapojení, špatně dimenzované kabely, střídače (či také měniče nebo invertory), odrazy povrchů a znečištění panelů. Problémem je, že na výkonu celé FVE a ani u jednotlivých stringů (řetězců sériově spojených panelů) nelze poznat, že některá část zařízení je poškozena a nepracuje na 100 %. Většinou až do té doby, než zcela vypadne z provozu a výrazněji začne ovlivňovat výkon některého segmentu. [1]

Obvyklé monitorovací soustavy použité na většině FVE pracují nad úrovní jednotlivých stringů, kde je ke střídači paralelně připojeno často až 24 fotovoltaických panelů. Relativně nízká účinnost panelů a současně neustálé změny ve slunečním svitu pak komplikují možnost rozpoznat menší odchylky způsobené poruchami jednotlivých panelů. Panely většinou díky používanému ochrannému systému By-pass (překlenovacích) diod nevypadnou kompletně, ale fungují jen z části a sníží se jejich maximální výkon. Z těchto výše zmíněných důvodů je proto vhodné provádět pravidelnou diagnostiku FVE. [1]

## 1.1 Elektrické parametry fotovoltaického článku

Vlastnosti fotovoltaického článku (FVČ) a jeho chování v závislosti na okolních klimatických podmínkách je popsáno charakteristickými křivkami. Příklad těchto křivek s vyznačenými parametry je zobrazen na Obr. 1.1, kde křivka s modrou barvou představuje závislost proudu na napětí a zelená křivka pak výkonu na napětí.



Obr. 1.1: Příklad charakteristických křivek proud-napětí a výkon-napětí s vyznačenými parametry

### 1.1.1 Napětí naprázdno

Napětí naprázdno  $U_{oc}$  (*Open-Circuit Voltage*) je maximální možné napětí na výstupních svorkách FVČ, kterého je za daných podmínek dosaženo při  $I = 0$  A čili pokud na jeho svorky není připojena zátěž. Toto napětí je dáno především použitým polovodičem a výrazný vliv na velikost  $U_{oc}$  má také teplota FVČ.

### 1.1.2 Zkratový proud

Zkratový proud  $I_{sc}$  (*Short-Circuit Current*) též nazývaný jako proud nakrátko je maximální možný proud generovaný FVČ, kterého je za daných podmínek dosaženo při  $U = 0$  V čili pokud jsou jeho výstupní svorky spojeny nakrátko. Tento proud je u FVČ dán především hodnotou intenzity dopadajícího záření. Při konstantní intenzitě záření se FVČ chová jako

zdroj konstantního proudu, jehož intenzita je limitována počtem fotonů dopadajících za jednotku času. Mezi další faktory ovlivňující velikost  $I_{SC}$  patří:

- Spektrální citlivost FVČ
- Plocha FVČ
- Teplota FVČ

### 1.1.3 Bod maximálního výkonu

Bod max. výkonu **MPP** (*Maximum Power Point*) je bod na charakteristické křivce FVČ, ve kterém je článok za daných podmínek schopen dosáhnout maximálního výkonu. Poloha tohoto bodu je na charakteristické křivce FVČ určena prostřednictvím napěťové a proudové souřadnice ( $U_{MPP}$ ,  $I_{MPP}$ ).

- $U_{MPP}$  – napětí, při kterém FVČ dodává maximální výkon
- $I_{MPP}$  – proud, při kterém FVČ dodává maximální výkon
- $P_{MPP}$  – maximální elektrický výkon, kterého lze dosáhnout v MPP a který je schopen FVČ dodávat, výpočet jeho hodnoty vyjadřuje rovnice (1)

$$P_{MPP} = U_{MPP} \cdot I_{MPP} \text{ (W)} \quad (1)$$

**Pozn.:** hodnota  $P_{MPP}$  v katalogu výrobců je měřena při STC a uváděná jednotka je Watt-peak (Wp).

### 1.1.4 Činitel zaplnění

Činitel zaplnění **FF** (*Fill Factor*) je parametr udávající poměr mezi hodnotou výkonu v MPP a hodnotou teoreticky maximálního výkonu daného napětím naprázdno a zkratovým proudem. Výpočet tohoto parametru uvádí rovnice (2).

$$FF = \frac{P_{MPP}}{U_{OC} \cdot I_{SC}} = \frac{U_{MPP} \cdot I_{MPP}}{U_{OC} \cdot I_{SC}} = \eta_{el} \text{ (-, \%)} \quad (2)$$

Tento činitel dosahuje v ideálním případě hodnoty 1 (MPP je pak dán souřadnicemi  $U_{OC}$  a  $I_{SC}$ ). V reálném případě dosahuje  $FF$  vždy menší hodnoty (cca 0,5 až 0,85) a podle jeho

velikosti lze posuzovat, jak kvalitní je příslušný FVČ. Graficky se dá  $FF$  určit z poměru plochy  $B$  k ploše  $A$  (viz Obr. 1.1).

### 1.1.5 Proud $I_{450}$

Proud označovaný jako  $I_{450}$  je hodnota proudu generovaného FVČ při napětí 450 mV na jeho výstupních svorkách. Jedná se o základní měřený parametr, který je dán mezinárodním standardem IEC 60904-1 (totožný s ČSN EN 60904-1 ed.2), sloužící pro porovnávání a třídění FVČ do výkonnostních tříd. Touto operací prochází každý článek při závěrečné fázi výroby, neboť články použité ve fotovoltaických panelech musí mít přibližně shodné vlastnosti. [3][4][5]

Dalším faktem pro měření tohoto parametru může být lepší určení (odhadnutí) polohy MPP, neboť porovnáním polohy MPP na V-A charakteristikách s průsečíky vnesených souřadnic  $U = 450$  mV a  $I = I_{450}$  (A) dojdeme k závěru, že MPP leží právě v blízkosti bodů určených těmito souřadnicemi. [6][7]

**Pozn.:** hodnota parametru  $I_{450}$  je měřena při STC (viz kapitola 1.3).

### 1.1.6 Účinnost

Účinnost  $\eta$  FVČ určuje maximální elektrický výkon, který může vyrobit určitá plocha článku. Vychází z poměru výkonu dodávaného FVČ a výkonu ozáření zdrojem. Vypočítá se z elektrického výkonu v bodě maximálního výkonu  $P_{MPP}$ , intenzity záření zdroje  $E$  ( $W/m^2$ ) a plochy  $A_c$  fotovoltaického článku dle rovnice (3). [8][9]

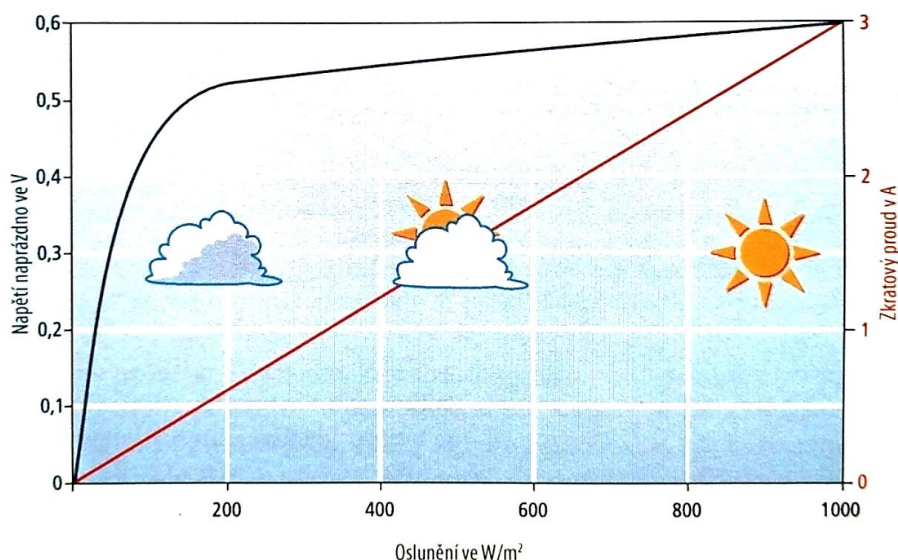
$$\eta = \frac{P_{MPP}}{A_c \cdot E} = \frac{U_{MPP} \cdot I_{MPP}}{A_c \cdot E} = \frac{FF \cdot U_{OC} \cdot I_{SC}}{A_c \cdot E} \quad (-, \%) \quad (3)$$

V datových listech výrobců je účinnost uvedena vždy při standardních zkušebních podmínkách  $\eta_n = \eta_{STC}$ , z toho vyplývá jmenovitá účinnost FVČ viz rovnice (4).

$$\eta_n = \frac{P_{MPP(STC)}}{A_c \cdot 1000 \text{ W/m}^2} \quad (-, \%) \quad (4)$$

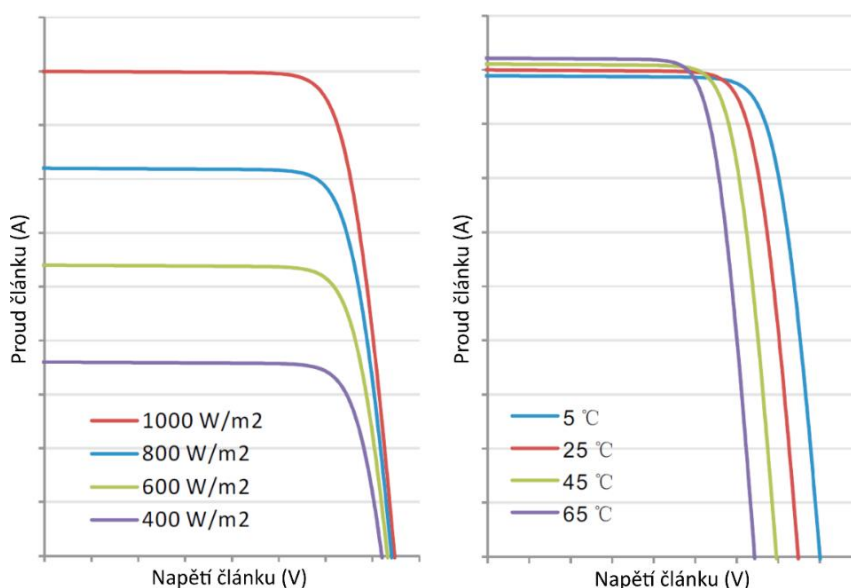


**Vliv intenzity záření** – Intenzita záření se přímo projevuje především na zkratový proud FVČ a na napětí naprázdno má pouze malý vliv (viz Obr. 1.3 vlevo). Pokud se sníží intenzita záření na polovinu, dodává články jen poloviční proud. Z toho vyplývá lineární závislost zkratového proudu na oslunění, proto na Obr. 1.2 vychází přímka. [9]



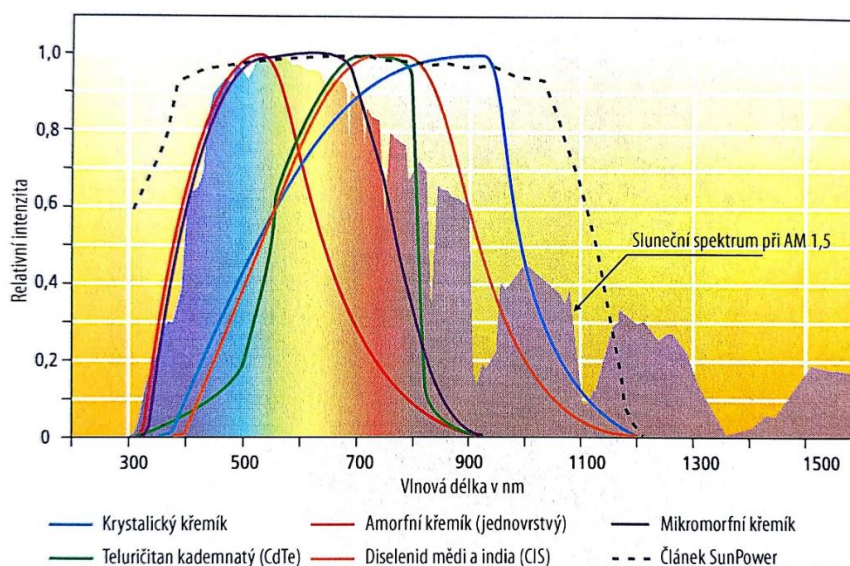
**Obr. 1.2:** Vliv oslunění na napětí naprázdno a zkratový proud [9]

**Vliv teploty** – Teplota FVČ ovlivňuje nepatrně pozitivně zkratový proud ale zejména pak negativně napětí naprázdno (viz Obr. 1.3 vpravo). Velikost napětí stoupá při nízkých teplotách, naopak při vyšších teplotách napětí klesá. V důsledku toho klesá při zahřátí FVČ účinnost a tím pádem i výkon. [8]



**Obr. 1.3:** Závislost charakteristických křivek FVČ na intenzitě záření a teplotě

**Spektrální citlivost** – Popisuje, v jakém rozsahu vlnové délky FVČ nejefektivněji pracuje a ovlivňuje tak účinnost při různých poměrech ozáření. Sluneční záření je složeno ze složek s rozdílnými vlnovými délkami. Krátké vlnové délky tvoří ultrafialové světlo, střední energeticky bohaté viditelné světlo a dlouhé vlnové délky infračervené světlo. Podle použitého materiálu a technologie mohou FVČ různá barevná spektra slunečního světla převádět na energii více či méně účinně. Obr. 1.4 ukazuje sluneční spektrum při  $AM = 1,5$  a spektrální citlivost různých materiálů FVČ. [8][9]



**Obr. 1.4:** Spektrální citlivost různých druhů FVČ [9]

**Vliv sériového a paralelního odporu** – V každém FVČ vzniká ztrátový výkon způsobený vlivem sériového odporu  $R_s$  a paralelního odporu  $R_p$ , přičemž nám znalost velikostí těchto dvou odporů poskytuje informaci o jeho kvalitě. Oba tyto odpory totiž způsobují zploštění charakteristické křivky FVČ a negativně tak ovlivňují celkovou účinnost. V ideálním případě je velikost  $R_s = 0$  a  $R_p = \infty$ , těchto hodnot však u reálného FVČ nelze dosáhnout, a proto je alespoň snaha o minimální  $R_s$  a maximální  $R_p$ . [6][9]

Prostřednictvím sériového odporu je vyjádřen úbytek napětí, ke kterému dochází při putování nosičů náboje z polovodiče k elektrickým kontaktům a jeho reálná hodnota leží v rozsahu několika ( $m\Omega$ ).

Paralelní odpor je dán defekty krystalické mřížky nebo tzv. svodovými proudy, které vznikají kolem okrajů článku ( $R_p \gg 10 \Omega$ ). Příliš nízká hodnota  $R_p$  vypovídá o vadném FVČ

a článek se chová, jako by byl zevnitř zkratován. [6][9]

## 1.2 Modely fotovoltaického článku

Pro získávání charakteristických křivek FVČ existují různé modely, které se od sebe liší jejich složitostí řešení, přesností, tak i vhodností použití pro různé typy FVČ. V nadcházejících částech této kapitoly budou zmíněny nejpoužívanější modely a jejich součásti, schémata náhradního zapojení, proudové rovnice a posouzení přesnosti. Dále bude zjednodušeně rozebrán pouze standardní model FVČ, neboť podrobný popis a výpočet jednotlivých parametrů modelů není hlavním cílem této práce.

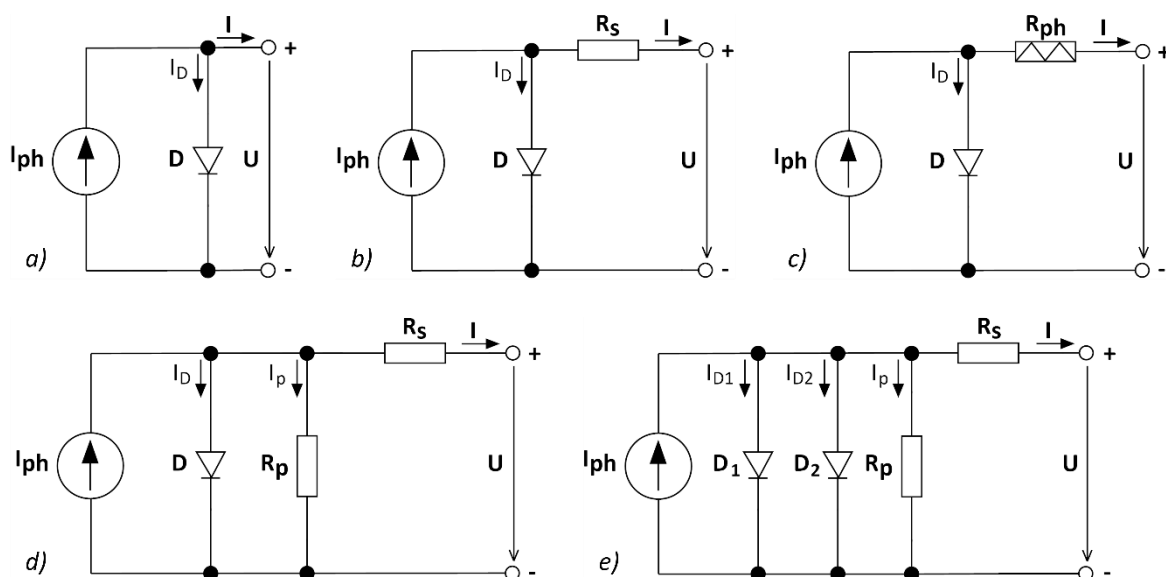
### 1.2.1 Popis nejpoužívanějších modelů

Ideální model osvětleného FVČ představuje paralelní zapojení proudového zdroje a diody, který nerespektuje vlivy sériového a paralelního odporu a jedná se tak o nejjednodušší možný model. V případě jednoduchého modelu je k ideálnímu modelu doplněn sériově zapojený rezistor. Bude-li uvažován vliv sériového a paralelního odporu, vznikne standardní (tzv. jedno-diodový) model, který je díky své jednoduchosti hojně používán a dokáže celkem přesně modelovat FVČ, což je pro běžné modelování naprosto dostačující. Je-li potřeba vyšší přesnost, používá se dvou-diodový nebo efektivní model FVČ. Dvou-diodový model vychází ze standardního modelu a je zde přidána další paralelně zapojená dioda. Efektivní model je zvláštní případ standardního modelu, který shrnuje oba jeho odpory do tzv. fiktivního odporu fotovoltaiky  $R_{ph}$ . Tento odpor fotovoltaiky může mít jak kladné, tak i záporné hodnoty.

V *Tab. 1.1* je uveden přehled nejpoužívanějších modelů pro FVČ, příslušných proudových rovnic, posouzení jejich přesnosti a *Obr. 1.5* představuje náhradní zapojení jejich schémat.

**Tab. 1.1:** Modely FVČ s rovnicemi charakteristických křivek a posouzením přesnosti

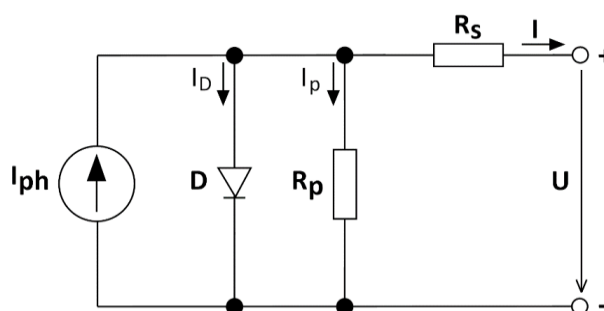
Typ modelu	Rovnice charakteristické křivky FVČ	Přesnost
ideální	$I = I_{ph} - I_0 \left( e^{\frac{U}{U_T}} - 1 \right)$	malá
jednoduchý	$I = I_{ph} - I_0 \left( e^{\frac{U+IR_s}{U_T}} - 1 \right)$	dobrá
efektivní	$I = I_{ph} - I_0 \left( e^{\frac{U+IR_{ph}}{U_T}} - 1 \right)$	velmi dobrá
jedno-diodový	$I = I_{ph} - I_0 \left( e^{\frac{U+IR_s}{U_T}} - 1 \right) - \frac{U + IR_s}{R_p}$	dobrá
dvou-diodový	$I = I_{ph} - I_{01} \left( e^{\frac{U+IR_s}{U_{T1}}} - 1 \right) - I_{02} \left( e^{\frac{U+IR_s}{U_{T2}}} - 1 \right) - \frac{U + IR_s}{R_p}$	velmi dobrá


**Obr. 1.5:** Schéma náhradního zapojení pro a) ideální, b) jednoduchý, c) efektivní, d) jedno-diodový, e) dvou-diodový model FVČ

Všechna schémata náhradního zapojení, příp. modelů FVČ mají za cíl matematicky popsat fotovoltaické charakteristické křivky s dostatečnou kvalitou. Pomáhají při teoretickém pochopení a vytváří základ pro měřicí a regulační zařízení ve fotovoltaice (např. regulátory MPP) nebo pro simulační programy. S jejich využitím se dají určit body maximálního elektrického výkonu při střídavých provozních podmínkách a optimálně tak nastavit pracovní bod FV systému. [9]

## 1.2.2 Model FV článku s jednou diodou

Standardním matematickým modelem FVČ používaného ve fotovoltaice je model využívající pouze jedné diody (jedno-diodový) a jeho schéma náhradního zapojení znázorňuje *Obr. 1.6*. Vychází z lineárního nezávislého proudového zdroje označeného v náhradním schématu jako  $I_{ph}$ , ke kterému je paralelně zapojena dioda označena jako  $D$  charakterizující vztah mezi driftovým a difúzním proudem uvnitř FVČ. Dále model zahrnuje dvojici rezistorů s označením  $R_s$  a  $R_p$  implementujících elektrický odpor, přičemž sériově zapojený rezistor  $R_s$  představuje vliv odporu materiálu a proudových sběrnic a rezistor  $R_p$  v paralelním zapojení, který je též nazýván jako svodový rezistor, představuje defekty krystalické mřížky či případně svodový proud na okrajích FVČ. [9][10]



**Obr. 1.6:** Náhradní schéma modelu FV článku s jednou diodou

V tomto modelu je zahrnuto pět neznámých parametrů se kterými lze po jejich určení nasimulovat grafické znázornění charakteristické křivky konkrétního FVČ. Hodnoty parametrů jsou variabilní, závisí na intenzitě dopadajícího záření na FVČ a jeho vlastní teplotě. Jmenovitě se jedná o:

- $I_{ph}$  (A) – fotoelektrický proud generovaný fotony
- $I_0$  (A) – zbytkový saturační (nasycený) proud diody v závěrném směru
- $R_s$  ( $\Omega$ ) – sériový odpor
- $R_p$  ( $\Omega$ ) – paralelní odpor
- $n$  (–) – činitel ideálnosti diody

**Pozn.:** činitel ideálnosti diody je někdy též nazývaný jako emisní koeficient a závisí na materiálovém složení a technologii výroby ( $1 \leq n \leq 2$ ).

Celkový výstupní proud  $I$  generovaný FVČ je vyjádřen pomocí rovnice (5) s využitím aplikace prvního Kirchhoffova zákona. Proud protékající diodou  $I_D$  je vyjádřen pomocí Shockleyho rovnice (6), ve které  $U_T$  představuje teplotní napětí viz rovnice (7).

$$I = I_{ph} - I_D - I_p \quad (\text{A}) \quad (5)$$

$$I_D = I_0 \left[ \exp\left(\frac{U}{n \cdot U_T}\right) - 1 \right] \quad (\text{A}) \quad (6)$$

$$U_T = \frac{kT}{e} \quad (\text{V}) \quad (7)$$

kde:  $k...$  Boltzmannova konstanta  $1,38 \times 10^{-23} \text{ (J} \cdot \text{K}^{-1})$

$T...$  Termodynamická teplota P-N přechodu (K)

$e...$  Elementární náboj  $1,602 \times 10^{-19} \text{ (C)}$

Rovnice (5) celkového výstupního proudu generovaného FVČ je příliš stručná, proto je pro lepší matematické vyjádření přepsána do rozvinutého tvaru charakterizujícího vztah mezi napětím a proudem FVČ viz (8).

$$I = I_{ph} - I_0 \left[ \exp\left(\frac{e(U + IR_s)}{nkT}\right) - 1 \right] - \frac{U + IR_s}{R_p} \quad (\text{A}) \quad (8)$$

Výhodou tohoto modelu je, že značnou část parametrů nutných k určení charakteristických křivek FVČ lze zjistit z datových listů výrobců. Zbývající parametry, kterými jsou  $R_s$  a  $R_p$  lze určit na základě matematiky např. s využitím Newton-Raphsonovy iterační metody založené na napětí nakrátko a zkratovém proudu nebo případně pomocí speciálních měření jako je třeba flash test (viz kapitola 1.4.3) a podobně. [10]

### 1.3 Standardní zkušební podmínky

Pro možnost porovnávání jednotlivých FVČ nebo také fotovoltaických panelů mezi sebou je důležité porovnávat jejich hodnoty za stejně definovaných podmínek. Proto byly k určení elektrických parametrů stanoveny jednotné podmínky, při nichž lze zjistit charakteristickou křivku FVČ. Tyto tzv. standardní zkušební podmínky (STC – *Standard Test Conditions*) stanovené normou IEC 60904 příp. ČSN EN 60904 se týkají:

- Kolmé intenzity záření –  $E = 1000 \text{ W/m}^2$
- Teploty článku –  $T_{cell} = 25 \text{ °C}$  s tolerancí  $\pm 2 \text{ °C}$
- Definovaného spektra světla –  $AM$  (*Air Mass*) = 1,5

Protože je intenzita slunečního záření proměnlivá v závislosti na aktuálním počasí, bylo jako referenční hodnota pro určení účinnosti stanoveno ozáření  $1000 \text{ W/m}^2$ . Jelikož je výkon FVČ či panelů závislý také na teplotě a slunečním spektru, jsou definovány i tyto dvě hodnoty.

### 1.4 Diagnostické metody FVE

Fotovoltaické panely mohou obsahovat mnoho závad, které snižují jejich nominální výkon a je třeba je analyzovat. Diagnostika slouží k nalezení skrytých chyb, které mají za následek snížení výkonu FVE. Typickým případem je porucha jednotlivého panelu, díky kterému je omezena výroba celého stringu (řetězce sériově propojených panelů). Při pravidelné diagnostice je možné optimalizovat chod FVE a zajistit tak její delší životnost i výtěžnost. V oblasti diagnostiky fotovoltaických panelů existují v zásadě čtyři základní metody, které lze rozdělit do dvou skupin: [13][14][15]

- Terénní testy – Provádí se a probíhají přímo v areálu FVE, za spolupráce odpovědného technika. Při těchto testech tedy není nutné demontovat a převážet panely na specializovaná pracoviště. Konektory a spojovací krabice panelů ale musí být přístupné. Tyto testy jsou běžně podkladem pro většinu reklamací. [1]
- Laboratorní testy – Provádějí se ve specializovaných pracovištích. V případě těchto testů je vždy nutné panely odpojit ze stringů a převézt je. Testy v laboratoři odhalí naprosto detailně všechny závady, ale jsou poměrně drahé. Pro některá reklamační řízení jsou ovšem nezbytné. [1]

### 1.4.1 Termografie

Pojem termografie označuje zobrazovací metodu, která umožňuje analyzovat a graficky znázornit rozložení teploty na povrchu sledovaného objektu. Pro tyto účely se používá zařízení tzv. termokamera (infračervená kamera). Měření termokamerou je založeno na bezkontaktním měření tepelné emisivity povrchů těles v infračerveném spektru elektromagnetického záření, které je schopno diagnostikovat skutečný aktuální tepelně-technický stav jakéhokoliv objektu, a to i včetně slabých anomálií. [15][16]

Měření termokamerou je nejrychlejším základním způsobem diagnostiky velkého množství fotovoltaických panelů a dalších komponent FVE při provozních podmínkách, a to bez omezení a zásahů do chodu celé elektrárny. Je současně metodou nejlevnější a bývá nejčastěji podkladem pro různá následná reklamační řízení. Cílem je na základě anomálních teplot detekovat body s vyšší teplotou, tzv. *hotspots* a odhalit tak poškozené fotovoltaické panely a komponenty, které způsobují často značné poklesy výkonu FVE. Dokáže rychle, jednoduše a spolehlivě odhalit v terénu jinak v podstatě nezjistitelné závady a další problémy FVE mezi které patří: [1][16][17]

- Nečistoty a jiné zastínění
- Nevyrábějící stringy
- Nezapojené a zkratované panely
- PID (*Potential Induced Degradation*)
- Poruchy střídačů a kabeláže
- Vadné By-pass diody
- Vadné články panelů
- Vysoké přechodové odpory

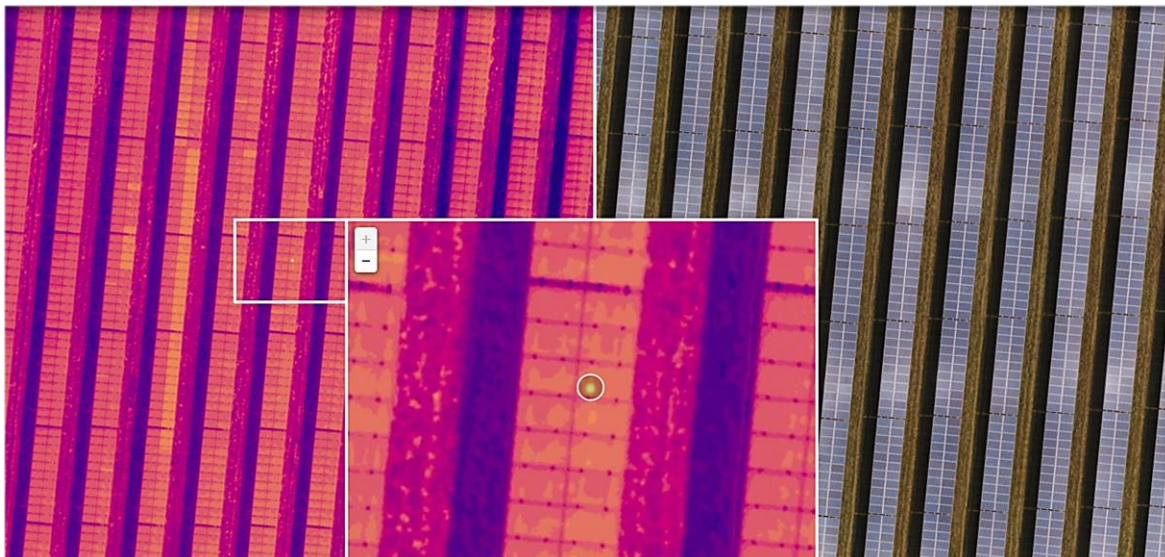
Při skrytých závadách vadných článků, znečištění nebo dlouhodobém zastínění a dalších problémech dochází k přehřívání jednotlivých článků panelů či jiných komponent. Měření termokamerou umožní včas a snadno detekovat odchylky od normálních teplot, a předejít dále možným poruchám zařízení, ztrátám a následně komplikacím v dodávkách a výrobě elektrické energie. [1]

Pro diagnostiku rozsáhlých elektráren lze využít letecké termografie prostřednictvím



bezpilotního letounu (dronu), který FVE prohlédne z ptačí perspektivy a poskytne tak dokonalý přehled o stavu všech panelů. Dron může nést společně s termokamerou také klasický RGB snímač. Tato kombinace pak umožňuje komplexní inspekci jak v infračerveném spektru pro nalezení typických závad, které se vyznačují jako místa se zvýšenou teplotou (*hot spoty*), tak i kontrolu ve viditelném spektru, kde lze odhalit především mechanické poškození panelů. [18][21]

Příkladem výstupu inspekce provedené přední českou společností JamCopters je kromě detailního revizního protokolu také ortofotomapa celé FVE s přesně vyznačenými závadami. Pro každý panel si lze pak intuitivně prohlédnout data z termokamery i RGB snímače včetně porovnání těchto dvou vrstev (viz *Obr. 1.7*). Vrstva snímků ve viditelném spektru může dosahovat rozlišení až 1 cm/pixel a volitelně lze vytvořit také 3D model celé FVE. Výsledky inspekce jsou k dispozici v online rozhraní odkudkoliv na PC, tabletu nebo telefonu, to vše bez nutnosti stahovat kompletní balík dat. Internetový prohlížeč zobrazuje soupis všech závad a zobrazuje jejich umístění na mapě FVE včetně vlastní pozice zařízení. To z něj dělá pomocníka pro rychlé nalezení vadného panelu v celém areálu FVE. [21]



**Obr. 1.7:** Snímek části FVE pořízený dronem vybaveného RGB snímačem a termokamerou s označením nalezené závady

Využití dronů představuje inovativní metodu. Drony lze libovolně nastavit v závislosti na potřebách diagnostického testu a díky tomu získat co možná nejrychlejší a nejpreciznější obraz závad FVE. Spojením termokamery a bezpilotního dronu je tedy možno zkrátit kontrolu na několik desítek minut a provozovatel FVE získává ucelený termogram (snímek pořízený

termokamerou) celé elektrárny ve formě mapy s přesnými souřadnicemi závad a volitelným 3D modelem. Pro možnost kontroly mechanického poškození panelů je součástí i detailní mapa ve viditelném spektru s vysokým rozlišením. [21][22]

#### 1.4.2 Měření V-A charakteristik

Jedná se o základní, přesto efektivní metodu pro vyhodnocení V-A charakteristik na úrovni celých stringů a jednotlivých fotovoltaických panelů. Měření je možné provádět přímo na konstrukci v areálu FVE a není tak potřeba panel demontovat. Jedinou nutností je pouze přístup ke konektorům panelů. [15][23]

Samotné měření V-A charakteristik se provádí pomocí analyzátoru fotovoltaických panelů často s automatickým skenem. Současně jsou také přesně měřeny standardní parametry, intenzita osvětlení pomocí referenčního FVČ, teplota panelu a okolního prostředí. Na *Obr. 1.8* je příklad schématu zapojení takového analyzátoru společně s měřením standardních parametrů. Výstupem z analyzátoru je V-A charakteristika a naměřené hodnoty typických parametrů při provozních podmínkách (OPC – *Operating Conditions*), které charakterizují stav jednotlivých panelů či celých stringů. Získaná data je možno obvykle normalizovat (extrapolovat na hodnoty při STC) a ihned porovnat s nominálními údaji deklarovanými výrobcem jednotlivých panelů. Velice rychle se tak dá určit, zda string nebo panel odpovídá charakteristikám deklarovaných výrobcem. Při diagnostice analyzátozem je standardně měřeno a vypočítáváno: [1][23]

- Napětí naprázdno
- Zkratový proud
- Maximální výkon
- Napětí a proud při max. výkonu
- Činitel zaplnění
- Intenzita záření
- Teplota panelu a okolního prostředí



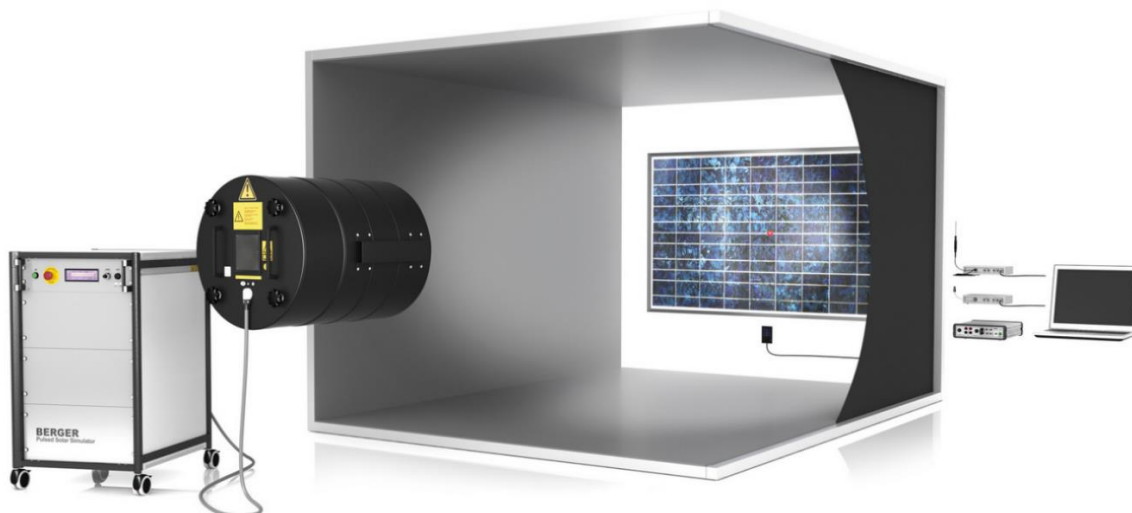
**Obr. 1.8:** Vlevo – Schéma zapojení analyzátoru HT I-V400W pro měření charakteristik fotovoltaických panelů, vpravo – detail referenčního článku

Měření V-A charakteristik v areálu FVE dokáže odhalit odchylky v charakteristikách panelů bez časově a finančně náročných testů v laboratoři. Pomáhá doplnit měření při podezření na aktivovanou By-pass diodu (vadný string v panelu). [24]

### 1.4.3 FLASH test

Diagnostická metoda označovaná jako FLASH test je základní metodou pro posouzení elektrického výkonu fotovoltaického panelu a jeho případných odchylek. Tento test změří V-A charakteristiku zkoušeného fotovoltaického panelu na kalibrovaném měřicím přístroji tzv. FLASH testeru (viz Obr. 1.9). Ten v laboratorních podmínkách simuluje ideální světelné i tepelné podmínky při kterých jsou panely testovány dle STC (viz kapitola 1.3). Následně jsou skutečné naměřené hodnoty porovnány s parametry, které udává výrobce. Pomocí tohoto testu lze určit všechny důležité parametry fotovoltaického panelu: [1][15][24]

- $U_{OC}$  – napětí naprázdno
- $I_{SC}$  – zkratový proud
- $P_{MPP}$  – maximální výkon
- $U_{MPP}, I_{MPP}$  – napětí a proud při max. výkonu
- $R_s, R_p$  – sériový a paralelní odpor
- $\eta$  – účinnost panelu
- $FF$  – činitel zaplnění



**Obr. 1.9:** Přehled a uspořádání prvků solárního simulátoru pro FLASH test

FLASH test je v podstatě jedinou metodou, která umí posoudit kompletní elektrickou charakteristiku fotovoltaického panelu, tzn. jeho výkon s možnými odchylkami. Zároveň slouží jako jediné směrodatné kritérium pro reklamace elektrických vlastností panelu. [13][15][24]

#### 1.4.4 Elektroluminiscence

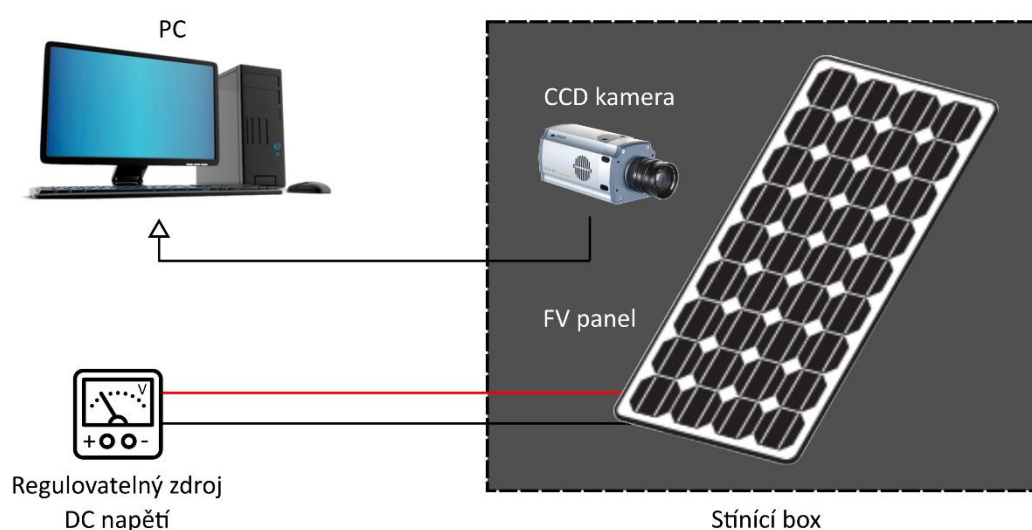
Elektroluminiscence je fyzikální děj, kdy při průchodu elektrického proudu o určité minimální velikosti polovodičovým přechodem dochází k emisi světelného záření z FVČ. Vlnová délka tohoto záření je už mimo viditelné spektrum a musí být tedy zaznamenáváno speciálně upravenou vědeckou CCD (*Charged Coupled Device*) kamerou pracující v NIR (*Near-Infrared*) spektru. [15][24][25]

Konkrétně se zkouška využívající skutečnosti elektroluminiscence označuje jako ELCD (*Electroluminescence Crack Detection*). Je používána za účelem odhalování materiálových a procesních defektů uvnitř FVČ, které nemusí být jinak vidět. Detekuje rekombinační záření, přičemž zviditelňuje rozložení proudu v daném panelu (jeho článcích) a zvýrazňuje tak nehomogenity a případné lokální poruchy. Typy defektů, které lze pomocí ELCD zkoušky odhalit:[1][15][25]

- Praskliny a mikropraskliny v článku
- Přerušenu metalizaci na jednotlivých článcích

- Elektricky neaktivní části panelu (části článků nebo celého stringu)
- Ostatní nehomogenity

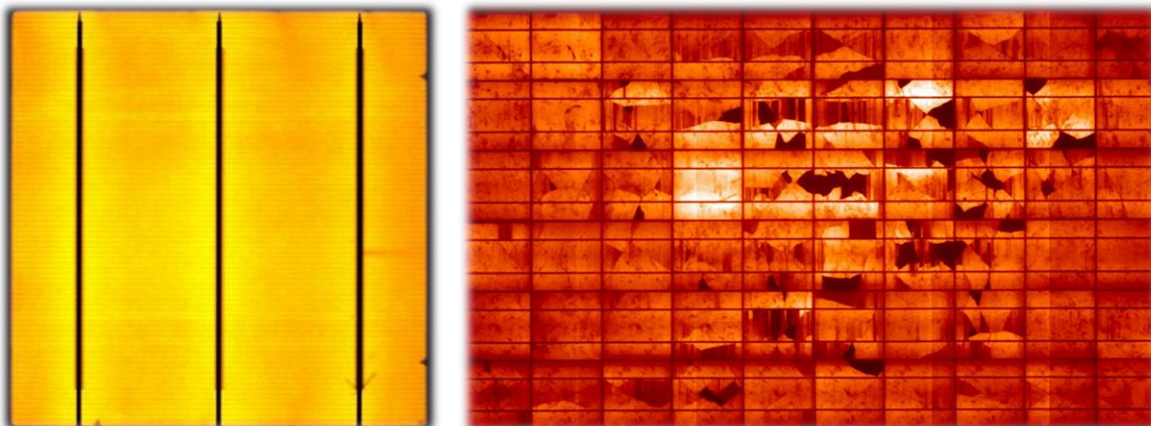
Při použití ELCD zkoušky (viz Obr. 1.10) je fotovoltaický panel ve tmě napájen z pomocného regulovatelného zdroje stejnosměrného napětí v propustném směru při asi 50 % jeho špičkového výkonu. Panel následně vydává záření, které snímá CCD kamera umístěná uvnitř stínícího boxu spolu s panelem. Díky různým světelným filtrům, které jsou vestavěny v použité CCD kameře, je možné pozorovat a analyzovat odlišné druhy defektů na připojeném počítači. [26]



**Obr. 1.10:** Schéma zapojení pro ELCD zkoušku

Detekce defektů a rozložení proudu ve fotovoltaickém panelu z pořízeného snímku spočívá v rozlišení tmavých a světlých míst. Ideální FVČ by měl vykazovat konstantní proudovou hustotu v celé ploše přechodu, a tím i jeho vyzářené záření by mělo mít na celé ploše stejnou intenzitu. Místa s nižší či žádnou intenzitou záření indikují poruchu ve struktuře materiálu nebo celý neaktivní článek případně celý string. Naopak jasně světlejší místa znázorňují oblasti s vyšší proudovou hustotou. [25]

Na snímku získaném ELCD zkouškou (viz Obr. 1.11 vlevo) je plně funkční monokrystalický FVČ bez závad, který má homogenní rozložení světlé barvy. Druhý snímek zobrazuje poškozený fotovoltaický panel, kde jsou zřetelná tmavá a světlá místa (viz Obr. 1.11 vpravo).



**Obr. 1.11:** Elektroluminiscenční snímky: Vlevo – nepoškozeného FVČ, vpravo – poškozeného fotovoltaického panelu [27][28]

Jedná se o nejvíce detailní metodu diagnostiky fotovoltaických panelů, která odhalí skryté vady nezjistitelné ostatními metodami (termografie, měření V-A charakteristik, flash test). Již jsou dostupné systémy schopné provádět diagnostiku metodou elektroluminiscence i v terénních podmínkách a provést tak měření přímo v areálu FVE. Nicméně není tento způsob diagnostiky hojně rozšířen. U nás v České republice tento způsob dokonce nikdo nenabízí, a tak lze považovat tuto metodu stále čistě za laboratorní. [1][13]

## 2 Vývojové prostředí LabVIEW

Za start vývoje programovacího prostředí LabVIEW (z angl. *Laboratory Virtual Instruments Engineering Workbench*) neboli „laboratorní pracoviště virtuálních přístrojů“ lze považovat 80. léta (1983) uvedením desky pro rozhraní GPIB (*General Purpose Interface Bus*) z produkce americké společnosti NI (*National Instruments*). Tato firma byla založena v roce 1976 Jamesem Truchardem, Jeffreyem Kodoskym a Williamem Nowlinem. V důsledku jejich společného výzkumu dali tito tři kolegové vzniknout myšlence „virtuální instrumentace“. [29]

Výchozím předpokladem byla představa, aby technik, který je schopen zapsat své poznatky a požadavky do blokového diagramu, mohl intuitivně zapsat podobně i program. Tento předpoklad tak dal vzniku vývojovému prostředí, které umožňuje tvořit programy v grafické podobě, což je intuitivnější a samozřejmě také rychlejší než klasické textové programování. Program obsahuje různé funkce, reprezentované ikonami, které lze vzájemně propojovat virtuálními vodiči a výsledky pak následně zobrazovat např. do grafů. [29]

Firma NI je průkopníkem a největším výrobcem v oblasti virtuální instrumentace, technické disciplíny, která zažívá veliký rozvoj v oblasti vývoje, výzkumu, školství a průmyslu.

### 2.1 LabVIEW – nástroj virtuální instrumentace

Prostředí LabVIEW, někdy též nazývané jako G-jazyk („grafický jazyk“), je vhodné pro celou řadu systémů různých složitostí. Nejen k programování systémů pro měření a analýzu signálů, řízení a vizualizaci technologických procesů, ale také k programování složitých systémů jako je třeba robot. [29]

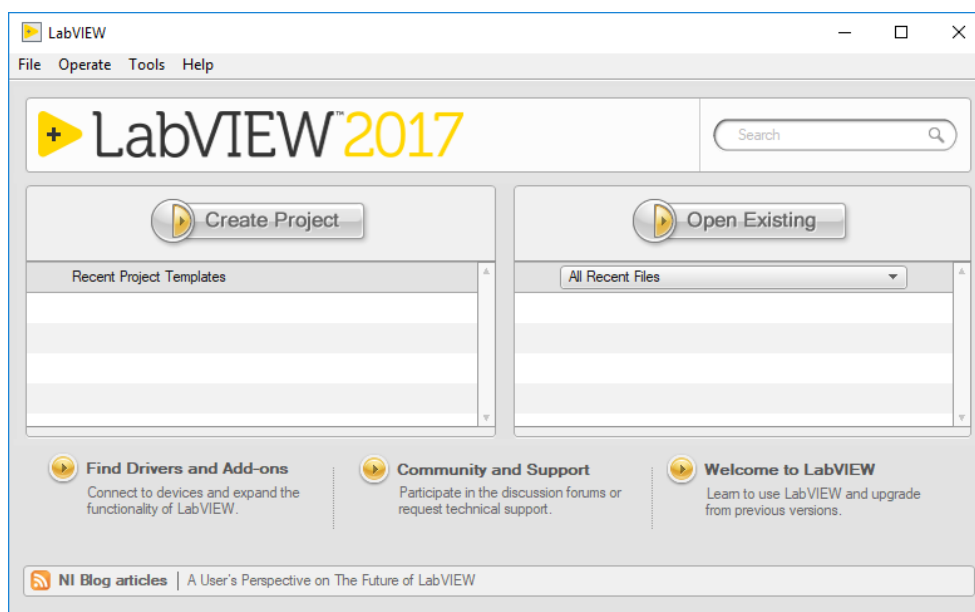
Z důvodu ušetření prostoru, financí a mnohdy i časové náročnosti využití je hlavním cílem virtuální instrumentace dočasně nebo i trvale nahradit **hardware** (technické prostředky), řešením virtuálním (zdánlivým) za pomoci **software** (programové prostředky) a zejména pak grafickými a vizuálními prostředky a poskytnout tak uživateli maximální názornost. [29]

Navrhování nových aplikací je díky tomuto řešení rychlé a lze snadno provádět změny v konfiguraci daného virtuálního přístroje, což je u realizace skutečnými nástroji složených z fyzických součástí často velmi finančně nákladné nebo zcela nemožné. [29]

Do souborů, resp. programů vytvořených v prostředí LabVIEW je pojem virtuální instrumentace také zanesen a promítnut příponou **.vi** u vytvořených VI (virtuálních instrumentů).

## 2.2 Úvodní obrazovka LabVIEW

Ihned po spuštění programu LabVIEW, za předpokladu aktivované licence, je jako první uživateli zobrazena úvodní obrazovka LabVIEW 2017 (dle příslušné nainstalované verze LabVIEW) s navigačními dialogovými okny pro práci se soubory a pro hledání dalších informací a zdrojů, zde na horní liště také najdeme „klasické“ základní menu, náhled na tuto obrazovku je na *Obr. 2.1*.



**Obr. 2.1:** Úvodní obrazovka LabVIEW 2017

Programování a tvorbu nového VI lze zahájit například volbou **Blank VI** v základním menu provedením kroků **File»New VI**, tímto způsobem budou dále v této práci uváděny veškeré kroky popisující postup provedených úkonů. Po provedení příslušných kroků je uživateli zobrazen prázdný čelní panel (*Front Panel*) a také prázdný blokový diagram (*Block Diagram*) s předdefinovaným prozatímním názvem *Untitled 1*. Tato dvě asociovaná okna



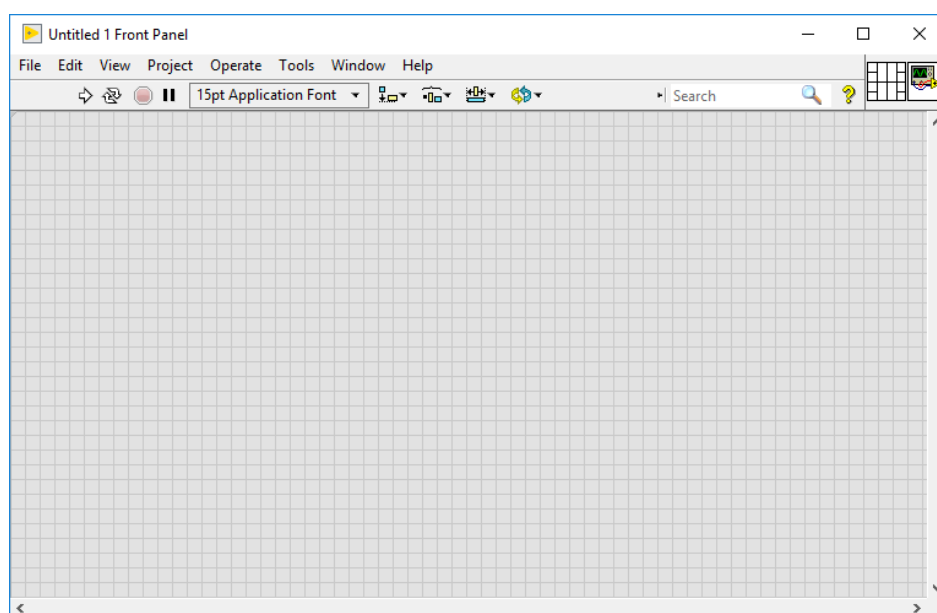
tvoří základní uživatelské rozhraní každého nového VI, prostřednictvím kterých LabVIEW komunikuje s uživatelem.

## 2.3 Čelní panel (Front Panel)

Čelní panel tvoří uživatelské rozhraní vytvořeného VI a určuje jeho vzhled a chování. Přes objekty čelního panelu (ovládací a indikační prvky) může uživatel řídit běh aplikace, zadávat vstupní parametry a získávat informace o výstupních zpracovaných datech. Popis ovládacích a indikačních prvků bude dále popsán v kapitole 2.3.3. Veškeré objekty je možné měnit a upravovat, programově nastavovat, automaticky přizpůsobovat velikost okna atd.

### 2.3.1 Obrazovka čelního panelu

Na Obr. 2.2 je vidět náhled na doposud prázdnou obrazovku čelního panelu při spuštění nového VI s výchozím názvem Untitled 1.

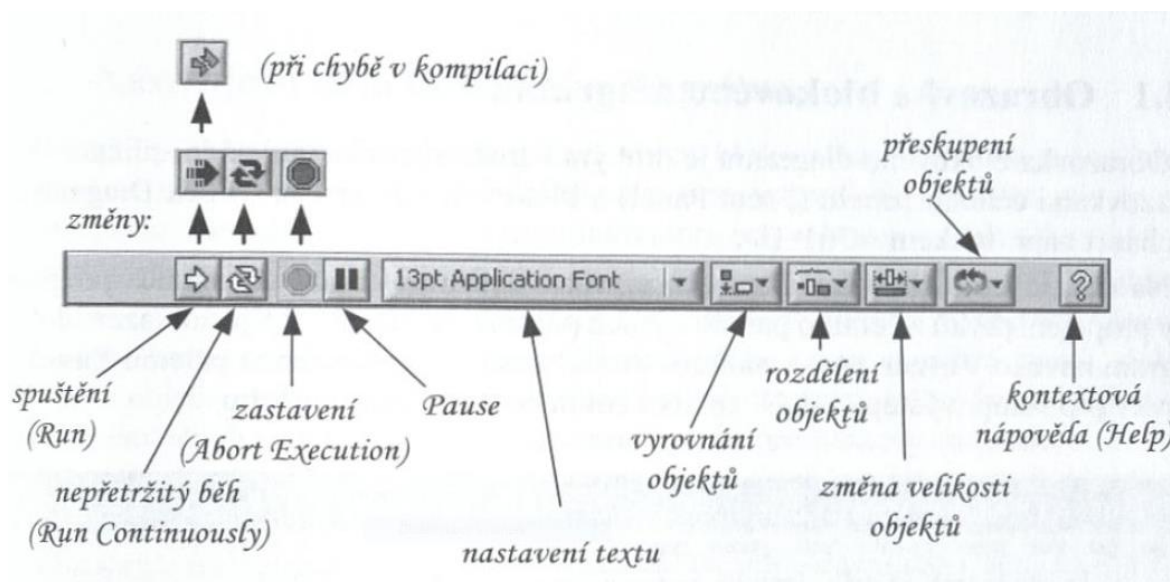


**Obr. 2.2:** Obrazovka čelního panelu

V tomto okně čelního panelu uživatel vytváří vnější vzhled aplikace, umísťuje zde ovládací prvky, definuje jejich polohu, chování a vzhled. Na této ploše lze pracovat nejen s vlastními vytvořenými prvky, ale také s poměrně rozsáhlou škálou prvků, které jsou součástí knihoven samotného vývojového prostředí LabVIEW, tyto konkrétní prvky pro čelní panel si programátor vybírá z nabídky palety **Controls** popsané v kapitole 2.5.2.

### 2.3.2 Nástrojová lišta čelního panelu

Na nástrojové liště čelního panelu se nacházejí hlavní prvky pro ovládání a indikaci jeho stavu viz *Obr. 2.3*. Veškeré prvky na nástrojové liště čelního panelu mají funkci jak tlačítka, tak současně i indikátoru pro zobrazení jeho stavu.



**Obr. 2.3:** Nástrojová lišta čelního panelu

Spuštění jednoho cyklu programu VI lze zahájit z nástrojové lišty stiskem tlačítka vpravo směřující šipky (**Run**), nepřetržitý běh programu má uživatel možnost spustit stiskem tlačítka v podobě dvou šipek ve smyčce (**Run Continuously**). Po přechodu do uživatelem zvoleného stavu se grafická prezentace příslušného tlačítka změní a je indikován stav běhu programu.

Průběh činnosti programu lze pozastavit stiskem tlačítka (**Pause**), v tomto okamžiku tlačítko změní barvu na dva červené proužky a program je pozastaven, opakovaným stiskem tohoto tlačítka lze běh programu znovu obnovit (**Continue**).

Je-li program spuštěn, je zvýrazněno také tlačítko (**Abort Execution**). Stiskem tohoto tlačítka dojde k zastavení běhu programu a indikační prvky nástrojové lišty jsou nastaveny do výchozího stavu.

Pokud je v programu vyskytnuta syntaktická chyba (chyba v kompilaci), změní se ikona tlačítka (**Run**) do stavu přerušené šipky (**Broken Run**) a spuštění programu s chybou není

uživateli umožněno. Při stisknutí tohoto tlačítka je zobrazena obrazovka **Error list** s popisem chyby a návodem na její odstranění a po stisku tlačítka **Show Error** je následně zvýrazněna a zobrazena chyba v blokovém diagramu.

Uprostřed lišty se nachází tlačítka pro nastavení velikosti a tvaru zobrazovaného textu v popisu jednotlivých prvků na čelním panelu. Stiskem velké plochy tohoto tlačítka lze volit velikost a typ fontu písma, stiskem malé šipky vpravo přejdeme do roletového menu, v němž lze volit další funkce pro změnu parametrů textu.

Další čtyři tlačítka umožňují práci s objekty umístěnými na čelním panelu a jsou určena pro dokončení vizuálního stylu úpravy finálního VI. Je zde možnost prvky vycentrovat, srovnat pod sebe nebo ke zvolenému okraji a definovat mezery mezi prvky obecně.

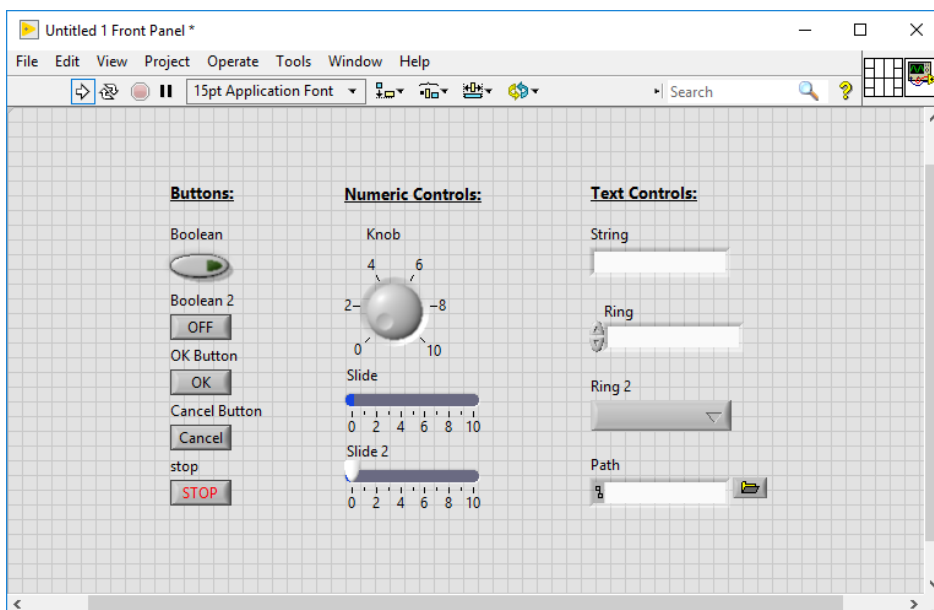
Zcela vpravo se nachází tlačítka pro aktivaci kontextové nápovědy (**Help**), to uživateli umožní zobrazit okno **Context Help** s informacemi o objektu LabVIEW, se kterým zrovna pracuje (na který právě ukazuje šipka kurzoru).

### 2.3.3 Prvky čelního panelu

Čelní panel je tvořen grafickými prvky, které jsou určeny pro řízení a ovládání VI, a prvky pro zobrazování. Rozdělení vizuálních prvků je z funkčního hlediska děleno na dvě základní skupiny:

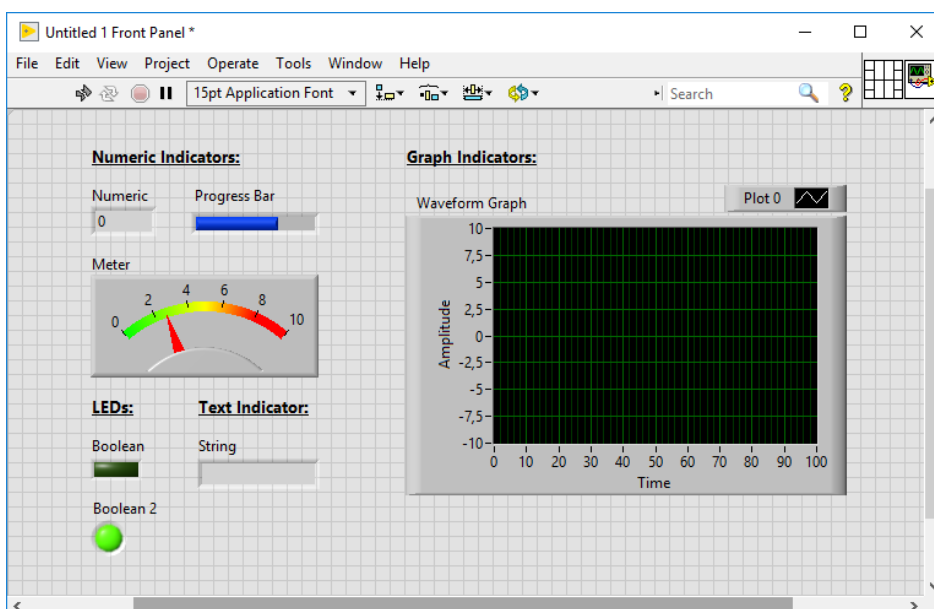
- Vstupy (**Controls**)
- Výstupy (**Indicators**)

Ovládací prvky (**Controls**) představují vstupní zařízení, podobně jako na skutečném přístroji nalezneme různé knoflíky, přepínače či tlačítka a jiná zařízení umožňující uživateli zadávat vstupní data a informace do programu. Vstupy v LabVIEW mohou mít tedy různé podoby ovladačů, jako je např. tlačítka (**Button**), otočný knoflík (**Knob**), posuvný ovladač (**Slide**), textový ovladač pro vkládání textů (**Text Control**), číselný ovladač pro vkládání číselných hodnot (**Numeric Control**) a další. Ukázka těchto zmíněných ovládacích prvků, které byly vybrány ze subpalety **Express** umístěných na čelním panelu je zobrazena na *Obr. 2.4*.



Obr. 2.4: Ukázka čelního panelu se vstupy ze subpalety Express

Výstupy mohou mít naopak různé podoby zobrazovačů (*Indicators*), jako je např. grafický model ručkového měřidla (*Meter*), textový či číselný zobrazovač (*Text Indicator*, *Numeric Indicator*), kontrolky (*LEDs*), proužkový diagram (*Progress Bar*), různé typy grafů (*Graph Indicators*) a další. Ukázka těchto zmíněných zobrazovacích prvků, které byly vybrány ze subpalety **Express** umístěných na čelním panelu je zobrazena na Obr. 2.5.



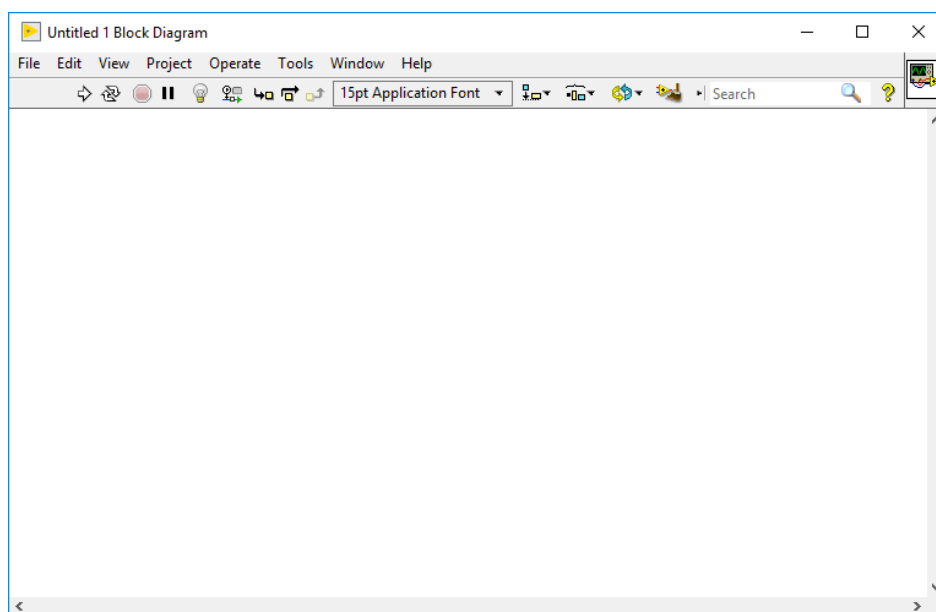
Obr. 2.5: Ukázka čelního panelu s výstupy ze subpalety Express

## 2.4 Blokový diagram (Block Diagram)

Blokový diagram se skládá ze tří typů součástí: uzel, terminál a spoj. Jednotlivé typy součástí budou popsány v kapitole 2.4.3. Společně tyto součásti představují obsah blokového diagramu, kde jsou vzájemně mezi sebou propojeny v podobě grafických objektů vykonávajících určitou funkci, které tvoří zdrojový kód VI. Blokový diagram, který se může podobat vývojovému diagramu programu, odpovídá rovněž řádkům textu v textově orientovaných programovacích jazycích. Ve skutečnosti je blokový diagram přímo spustitelným kódem, který je překládán již během psaní s možností okamžité zpětné vazby při vzniku chyby, jako např. při připojení neslučitelných datových typů.

### 2.4.1 Obrazovka blokového diagramu

Pro každou aplikaci v LabVIEW je druhým sdruženým oknem obrazovka blokového diagramu. Náhled na doposud prázdnou obrazovku blokového diagramu při spuštění nového VI (zde opět s výchozím názvem Untitled 1) je vidět na *Obr. 2.6*.



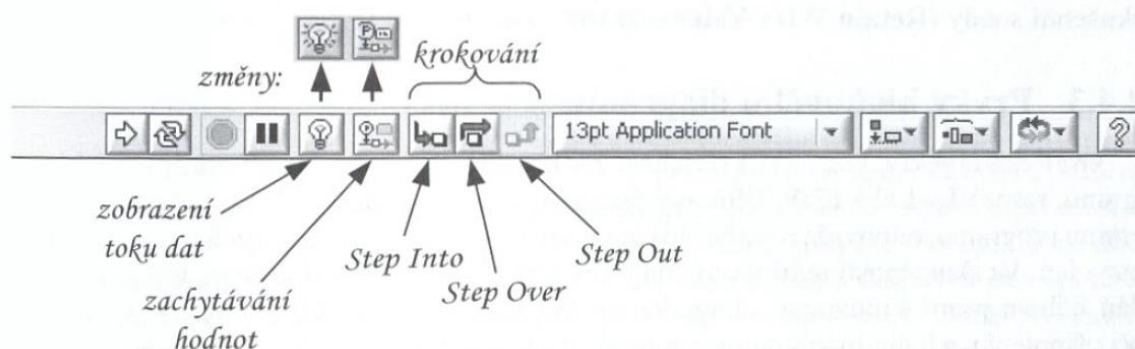
**Obr. 2.6:** Obrazovka blokového diagramu

V tomto okně blokového diagramu uživatel definuje vlastní grafický algoritmus programu, tudíž propojení prvků situovaných na čelním panelu a jejich parametry. Vkládání dalších prvků na obrazovku blokového diagramu potřebných pro algoritmus si programátor vybírá z palety knihovních prvků **Functions** popsané v kapitole 2.5.3, které jsou svázány s prvky

čelního panelu, s konstantami zadanými z knihovny **Functions** nebo s globální či lokální proměnnou v módu pro čtení nebo zápis.

### 2.4.2 Nástrojová lišta blokového diagramu

Na nástrojové liště blokového diagramu (viz *Obr. 2.7*) se nacházejí hlavní prvky pro ovládání a indikaci stavu blokového diagramu a obdobně jako u nástrojové lišty čelního panelu zde mají všechny prvky funkci tlačítka a zároveň indikátoru pro zobrazení aktuálního zvoleného stavu.



**Obr. 2.7:** Nástrojová lišta blokového diagramu

Můžeme zde vidět patřičnou podobu s nástrojovou lištou čelního panelu v rámci shodnosti některých tlačítek, kde tato tlačítka nástrojové lišty blokového diagramu mají shodnou funkci s tlačítky nástrojové lišty čelního panelu. Jedná se o tlačítka (**Run, Run Continuously, Pause, Abort Execution**), dále pak tlačítko pro nastavení velikosti zobrazovaného textu v popisu prvků, tlačítek pro práci s objekty a tlačítko pro spuštění kontextové nápovědy (**Help**). Chování a jejich grafická změna při provedené akci je také shodná se změnami provedenými na nástrojové liště čelního panelu. Rozdíl je v účinnosti právě pro blokový diagram.

V nástrojové liště blokového diagramu přibylo ke shodným tlačítkům ještě dalších pět tlačítek, jejichž funkce a význam je popsán na *Obr. 2.7*.

Tato přidaná tlačítka na nástrojové liště blokového diagramu jsou určena pro pomoc programátorovi při ladění programu (tzv. *debugging*). Při odladování programů v LabVIEW lze využít možnosti krokování, resp. uzel po uzlu (*node to node*). Jako uzel je chápán

výkonný prvek, což může být funkce, struktura (smyčka apod.) nebo SubVI. Při využití možnosti krokování pro odladování lze s výhodou použít tři dalších tlačítek:

- Step Into
- Step Over
- Step Out

Vykonání jednoho uzlu se spustí stiskem tlačítka (**Step Into**), tento uzel je současně zvýrazněn. Spuštění celého uzlu (struktury nebo SubVI) jako jednoho kroku v další hlavní funkci lze provést stiskem tlačítka (**Step Over**) a stiskem tlačítka (**Step Out**) se spustí vykonání uzlu jako jeden krok, ale ve volajícím VI nebo struktuře (pokud existuje, jinak je tlačítko neaktivní).

Dalším nástrojem pro odladování programu je tzv. vložení bodu přerušení (**Break Point**), kde je do toku dat v blokovém diagramu vložen bod, po jehož dosažení se aplikace v tomto bodě pozastaví a dále je možnost ji krokovat.

Tlačítko s grafickým symbolem žárovky (**Highlight Execution**) umožňuje po jeho stisku programátorovi zobrazit při běhu programu jeho datový tok (*Data Flow* viz kapitola 2.6) v blokovém diagramu. Znázornění aktivovaného zobrazení datového toku je reprezentováno grafickou ikonkou v podobě rozsvícené žárovky, naopak v neaktivním stavu má žárovka pohaslou podobu. Aktivací se běh programu sice zpomalí vinou náročné animace, avšak programátor může sledovat tok dat a jejich okamžitou hodnotu. Ta část, která je právě prováděna je zvýrazněna, okamžité hodnoty jsou znázorněny na odpovídajících výstupech a jednotlivé datové cesty signálu jsou znázorněny pohybujícími se kuličkami po virtuálních vodičích v barvě odpovídající příslušnému datovému typu. Tento režim je vhodný pro případy, kdy potřebujeme přesně predikovat posloupnost vykonávaných instrukcí a výsledný datový typ.

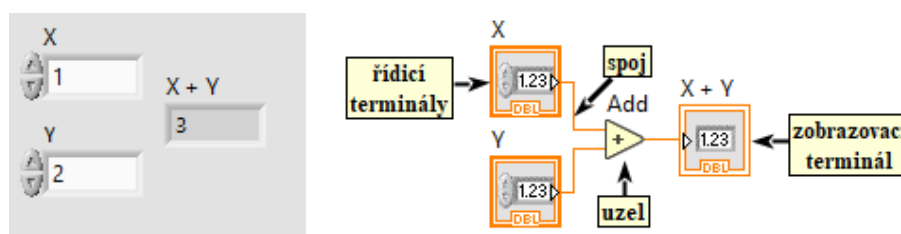
Tlačítko s grafickým symbolem zkušební sondy (**Retain Wire Values**) umožňuje po jeho stisku programátorovi vložení zkušební sondy (*Probe*) do cesty toku dat a zobrazit tak okamžitou hodnotu procházející konkrétním datovým vodičem. Znázornění aktivované zkušební sondy je reprezentováno zvětšením její ikony a její aktivace nezdržuje běh vykonávání programu.

### 2.4.3 Prvky blokového diagramu

Jak již bylo zmíněno v úvodu kapitoly 2.4, prvky blokového diagramu představují tři základní součásti, ze kterých se blokový diagram skládá:

- Uzel
- Terminál
- Spoj

Ukázka jednotlivých tří typů součástí blokového diagramu na příkladu jednoduchého programu VI, který realizuje sčítání dvou čísel hodnot X a Y je zobrazen na Obr. 2.8.



Obr. 2.8: Ukázka součástí blokového diagramu na VI pro sčítání

**Uzel (Node)** – Prvek pro vykonání programu. U textově orientovaných programovacích jazyků odpovídá příkazu, funkci nebo podprogramu. V případě LabVIEW rozlišujeme tři typy uzlů:

- Funkce – umožňují provádět různé základní operace (sčítání, práce se soubory...)
- SubVI – jsou VI pro pozdější použití v jiném VI
- Struktura – slouží pro řízení průběhu vykonávání programu

**Terminál (Terminal)** – Brána, přes kterou mohou procházet data mezi čelním panelem a blokovým diagramem, resp. mezi uzlem a blokovým diagramem. U textově orientovaných programovacích jazyků odpovídá konstantě nebo proměnné. Terminály se dále rozdělují na zdrojové (*Source*), případně ovládací (*Control*), což jsou počátky datových cest spojené s ovládacími prvky, konstantami, globální či lokální proměnnou v módu pro čtení, a koncové (*Destination*), případně zobrazovací (*Indicator*), což jsou konce datových cest spojené se zobrazovacími prvky, globální či lokální proměnnou v módu pro zápis.



**Spoj (Wire)** – Vytváří cestu pro tok dat mezi terminálem a uzlem a v běžném programovacím jazyku odpovídá proměnným. Z důvodu různých typů objektů, ze kterých se může blokový diagram skládat, mohou být různých typů i spoje v závislosti na datovém typu. Odlišení typu spoje je provedeno barvou a typem čáry a závisí na přenášeném datovém typu. Základní barevné rozlišení spojů a typů čar je uvedeno v *Tab. 2.1* resp. v *Tab. 2.2*.

*Tab. 2.1: Základní datové typy spojů a barva jejich čar*

Datový typ spoje	Barva čáry
celočíslný ( <i>Integer</i> )	modrá
číslný s pohyblivou čárkou ( <i>Floating Point</i> )	oranžová
binární ( <i>Boolean</i> )	zelená
znakový řetězec ( <i>String</i> )	růžová

*Tab. 2.2: Typy čar v závislosti na přenášeném datovém typu*

Přenášený datový typ	Typ čáry
skalár	—
1D pole	—
2D pole	==
cluster	----

**Pozn.:** typ čáry pro pole s více rozměry, než 2D se jeví jako tytéž čáry s rozšiřující se mezerou mezi nimi.

#### 2.4.4 Ikona a konektor VI

V pravém horním rohu každého čelního panelu a blokového diagramu VI se nachází ikona VI (viz *Obr. 2.2* resp. *Obr. 2.6*), v případě čelního panelu se zde vedle ikony nachází ještě konektor VI.

**Ikona** – Grafický symbol reprezentující daný VI při další práci a tento VI lze pak použít jako SubVI v jiné aplikaci a tato ikona jej bude zastupovat.

**Konektor** – Množina terminálů sloužících pro připojení SubVI v jiném VI, kterým lze ikonu opatřit, tedy definuje přípojná místa pro přivedení dat do dané ikony a odvedení dat z ní.

## 2.5 Palety

Palety jsou grafické panely obsahující různé nástroje a objekty pro vytváření a VI, programů a aplikací v LabVIEW. Tyto palety lze po pracovní ploše obrazovky posouvat tak, aby byly po ruce a zároveň nepřekážely při práci. V LabVIEW je možno rozlišit tři druhy palet:

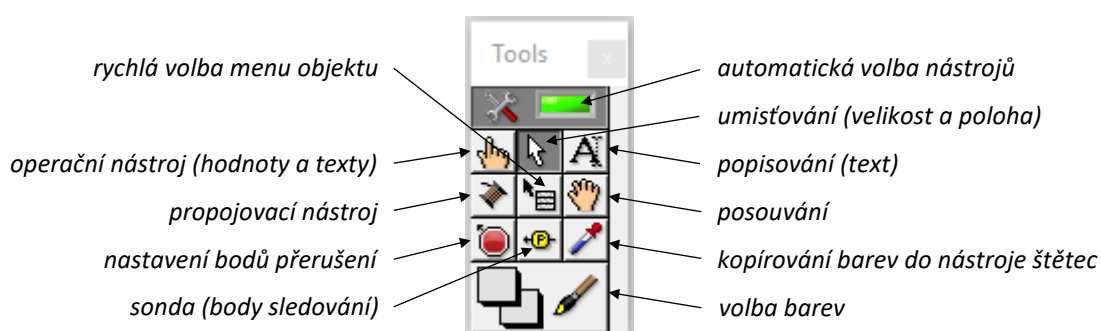
- Paleta nástrojů (**Tools**)
- Paleta ovládacích a zobrazovacích prvků (**Controls**) pro čelní panel
- Paleta funkcí (**Functions**) pro blokový diagram

**Pozn.:** nabídka prvků palet **Controls** a **Functions** se liší podle verze a nainstalovaných knihoven.

### 2.5.1 Paleta Tools

Vývojové prostředí LabVIEW má vytvořen zvláštní režim práce s kurzorem myši pro práci při sestavování nového VI. S využitím jeho pomoci si lze práci relativně dobře usnadnit, leč za předpokladu, že se s tímto režimem naučíme efektivně pracovat.

Paletu nástrojů **Tools** lze vyvolat např. tímto postupem v základním menu otevřeného VI **View»Tools Palette** a následně je uživateli na ploše čelního panelu eventuálně blokového diagramu zobrazen panel nástrojů (viz *Obr. 2.9*).



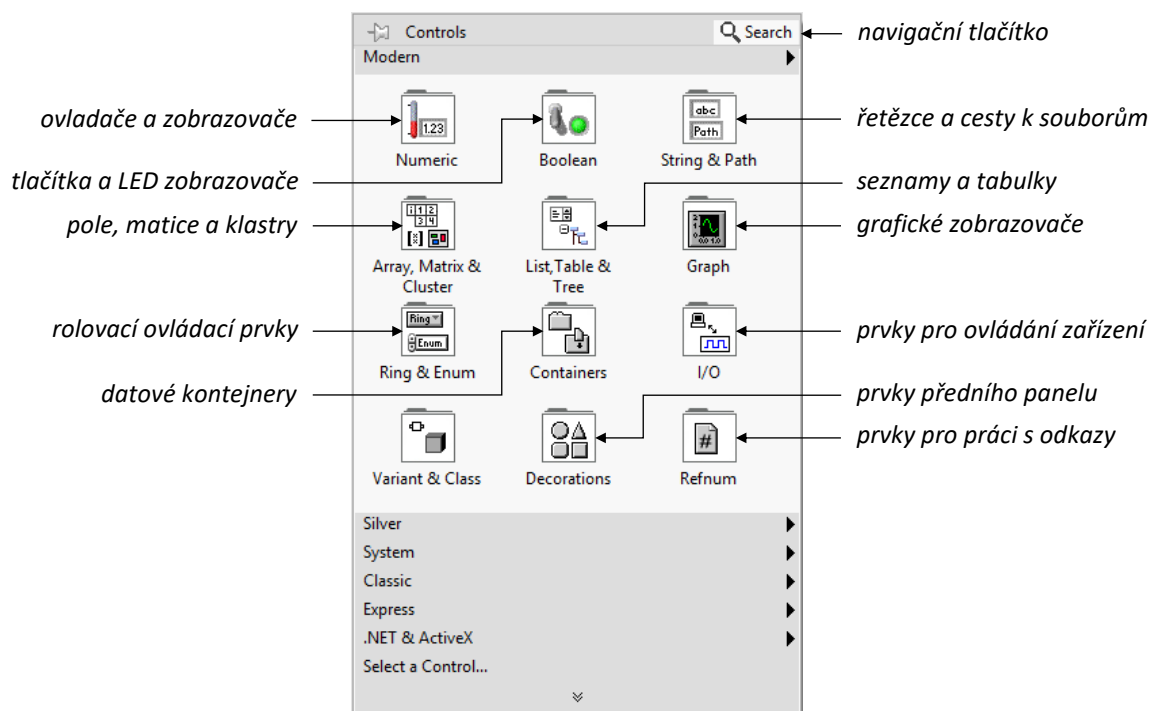
**Obr. 2.9:** Paleta Tools s popisem funkcí tlačítek

Pokud je zvolen režim automatické volby nástrojů tzn. grafický indikátor stavu na tlačítku svítí zeleně, LabVIEW automaticky nabízí odpovídající nástroj při umístění kurzoru nad daným objektem. Pokud je z nějakého důvodu výhodnější automatickou volbu nástrojů zakázat, provedeme to kliknutím na tlačítko či výběrem konkrétního nástroje z nabídky

palety a grafický indikátor stavu zhasne.

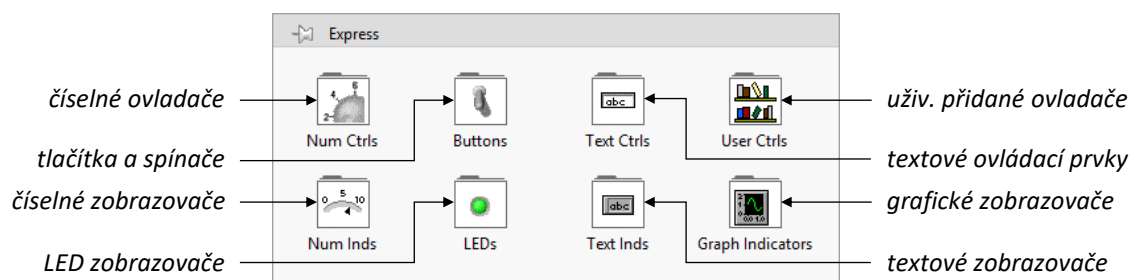
## 2.5.2 Paleta Controls

Paleta **Controls** (viz Obr. 2.10) je složena z ikon reprezentujících další subpalety s objekty pro vytváření čelního panelu a lze ji vyvolat např. stiskem pravého tlačítka myši na čelním panelu či podobně jako paletu nástrojů **View»Controls Palette**.



**Obr. 2.10:** Paleta Controls s popisem ikon

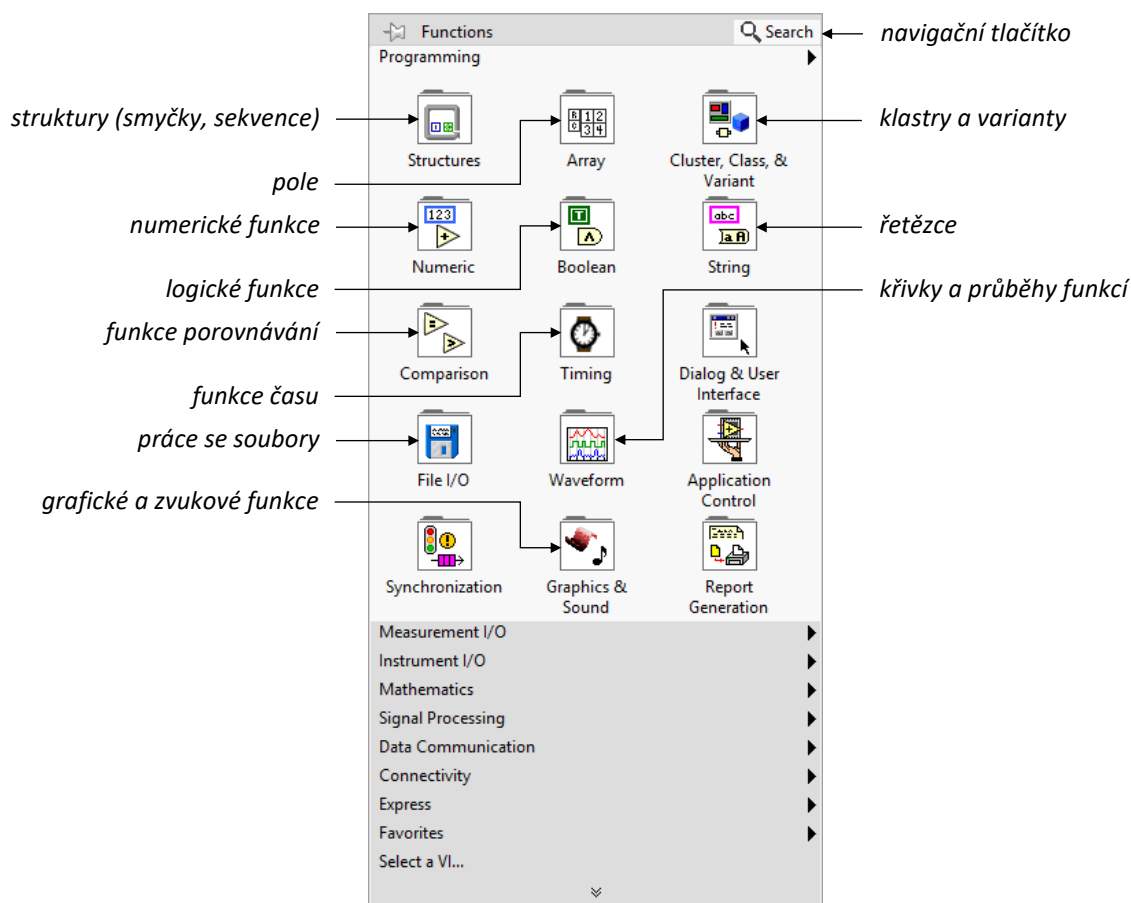
Jedná se o různá tlačítka, knoflíky, přepínače, prvky pro zadávání vstupních hodnot, indikátory, zobrazovače, měřiče a další řídicí (vstupní), resp. indikační (výstupní) prvky. Jelikož je konkrétních prvků poměrně velké množství, jsou v paletě knihovnických prvků členěny na další subpanely pro snadnější orientaci a výběr konkrétního prvku. Při umístění kurzoru nad ikonu subpanelu se tento subpanel otevře, a v něm potom lze vybrat konkrétní prvek přidržením levého tlačítka myši a přetáhnutím na plochu čelního panelu. Na Obr. 2.11 je uveden jako příklad popis prvků subpalety **Express**, ta obsahuje výběr často používaných prvků pro tvorbu běžných typů čelního panelu (např. měřicí přístroje...).



Obr. 2.11: Subpaleta Express

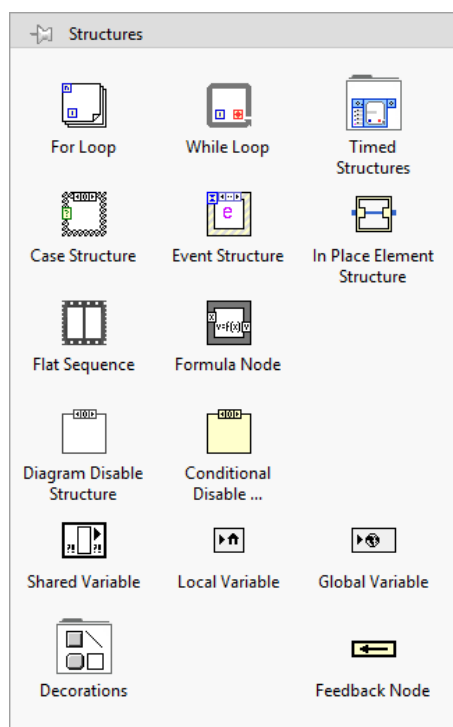
### 2.5.3 Paleta Functions

Paleta **Functions** viz Obr. 2.12 se (obdobně jako paleta **Controls**) skládá z ikon, které zastupují další subpalety s objekty pro vytváření blokového diagramu a lze ji vyvolat např. stiskem pravého tlačítka myši v okně blokového diagramu či přes základní menu **View»Functions Palette**.



Obr. 2.12: Paleta Functions s popisem ikon

Jedná se o různé funkce a všemožné „životně důležité prvky“ pro VI. Práce s paletou **Functions** je obdobná práci s paletou **Controls** popsanou výše. Na *Obr. 2.13* je pro ukázkou subpaleta **Structures**, která obsahuje některé důležité prvky programových struktur (smyčky For a While, funkce, struktury Case a další) pro tvorbu blokového diagramu VI.



**Obr. 2.13:** Subpaleta Structures

## 2.6 Datový tok (Data Flow)

Vykonávání programu v LabVIEW je řízeno datovým tokem a nikoli lineárním vykonáváním řádků kódu, jak je obvyklé u textových programovacích jazyků (tzv. řízený tok, angl. *Control Flow*), kde jsou singulární části programu řazeny sekvenčně. Datový tok tak jednoznačně určuje směr provádění programu. Zpracování informace v jednotlivých uzlech blokového diagramu se provede tehdy, existují-li na všech jeho vstupech nezbytné informace k jeho provedení. Po ukončení zpracování má uzel na výstupu definované informace, které se pošlou dále.

V textových programovacích jazycích se pro práci s konkrétními hodnotami používají programové proměnné (*Variables*), které je před prvním použitím, tzn. obvykle na začátku programu, nutné deklarovat, tedy definovat jejich název a vlastnosti. Určitou obdobou

v prostředí LabVIEW je datový tok reprezentovaný datovým spojem, kterým je přenos dat realizován. Funkce a VI automaticky alokují potřebnou paměť pro data a nejsou-li již data používána, přiřazená paměť je uvolněna. Jsou-li přidána nová data do pole nebo do řetězce, automaticky se přiřazená paměť zvětší.

### 2.6.1 Datové typy

Různé objekty, jejich vstupy či výstupy, jsou vzájemně propojeny spoji různých datových typů viz *Tab. 2.1*. Objekty různých, vzájemně neslučitelných, datových typů nelze vzájemně propojovat. Např. tlačítko je v blokovém diagramu označeno zeleným rámečkem, což odpovídá dle *Tab. 2.1* binárnímu datovému typu (*Boolean*) a může být tedy spojeno pouze spojem zelené barvy s dalším vstupem prvku s rámečkem zelené barvy. Knoflík představující otočný prvek s možností plynulého nastavování vstupních hodnot je zase označen rámečkem oranžovým, což značí číselný datový typ s pohyblivou čárkou (*Floating Point*) a může být propojen spojem oranžové barvy se vstupem prvku označeným rámečkem oranžové barvy. Naopak zeleně označené tlačítko nelze připojit na vstup oranžové barvy.

**Pozn.:** spoje jsou označeny stejnou barvou, jakou jsou označeny připojovací terminály.

Datový typ **Numeric** je číselný datový typ a v zásadě se dělí na typy:

- Uchovávající pouze celočíselné hodnoty (*Integer*)
- Pro reálná čísla (*Fixed-Point, Floating-Point*)

V aplikacích je vhodné diferencovat konkrétní typ a nastavit jejich formát předem tam, kde předvídáme, s jakými daty bude program pracovat a tím snížit dobu výpočtů a nadměrnou alokaci paměti.

Datový typ **Boolean** je výsledkem všech logických výrazů a může nabývat pouze dvou hodnot:

- Pravda (*True*)
- Nepravda (*False*)

Datový typ **Waveform** je speciální formát (tzv. klast – *Cluster*) pro data z časové oblasti, který obsahuje tyto položky:

- **t0** – start time ( $t_0$ ) – čas začátku (první bod) průběhu
- **dt** – delta t – časový interval mezi jednotlivými po sobě jdoucími body průběhu
- **Y** – waveform data – 1D pole čísel

Datový typ **Variant** nepředstavuje konkrétní datový typ, může obsahovat kterýkoli jiný datový typ. Obsahuje atributy, jako je např. název a jednotka měřicího kanálu.

Datový typ **String** je posloupnost znaků, které mohou být:

- Zobrazitelné (textové znaky)
- Nezobrazitelné (řídící znaky)

Datový typ **Array** je množina prvků stejného datového typu v uspořádání, které určuje dimenzi pole. Pozici prvku v poli určuje v každém rozměru jeho index, což je celé číslo v rozsahu 0 až  $n - 1$ , kde  $n$  je počet prvků v daném rozměru pole.

Datový typ **Cluster** je datová struktura, která může obsahovat několik různých datových typů společně a může být buď:

- Vstupní (obsahuje ovladače – *Controls*)
- Výstupní (obsahuje zobrazovače – *Indicators*)

Klaster se hned automaticky přizpůsobí prvnímu vloženému typu (ovladači nebo zobrazovači) a další přidané prvky automaticky změní na stejný typ. Seskupením logicky příslušných prvků se vytvoří klast a ten lze spojit jediným datovým spojem.

\*

---

\* Kapitola 2 (Vývojové prostředí LabVIEW) byla do této části zpracována na základě informací převzatých ze zdrojů [29][30]

## 2.7 Získávání dat (Data Acquisition)

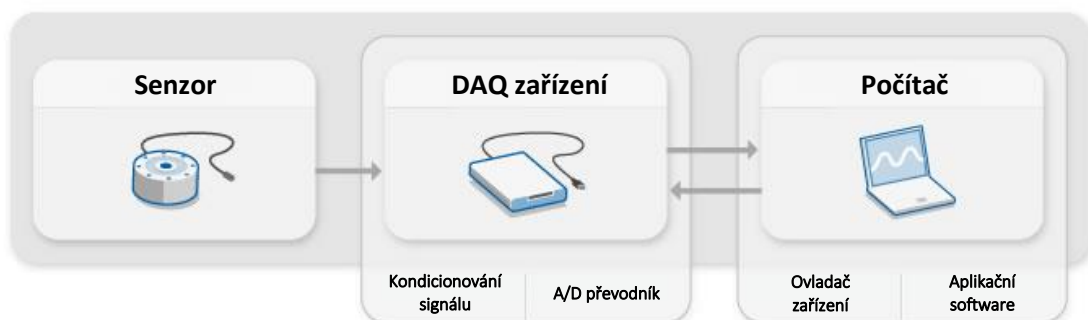
Práce s daty získanými při měření a generování fyzikálních signálů za pomoci počítače se nazývá pořizování dat (DAQ – *Data Acquisition*). Systémy určené pro pořizování dat, tzv. DAQ systémy, na základě počítače umožnily zvýšení rychlosti měření a zpracování dat. Obvodové řešení DAQ systémů umožnilo rovněž postupně nahradit analogový způsob zpracování signálů počítačovými systémy za pomoci vhodného programového vybavení (DSP – *Digital Signal Processing*). Tím vzniká tzv. virtuální přístrojová technika. Její hlavní charakteristikou je kombinace technického a programového vybavení se standardizovanými průmyslovými počítačovými technologiemi umožňující vytváření uživatelsky definovaných řešení. Moderní DAQ zařízení jsou většinou v provedení zásuvných karet (např. měřicí karty) pro vložení do systémového konektoru počítače nebo v provedení externího zařízení vybaveného rozhraním pro komunikaci s počítačem. [29]

Oblast DAQ patří mezi nejsilnější nástroje prostředí LabVIEW, umožňuje sofistikovaně analyzovat, zpracovávat a zobrazovat signál a získaná data. Každý signál z měření lze zpracovávat a vyhodnocovat z hlediska několika informací:

- Stavů – např. analogový signál, číslicový signál
- Rychlosti – např. průběh vzestupné či sestupné hrany
- Úrovně – např. hodnota v určitém místě průběhu
- Tvaru – např. průběh hrany
- Frekvenčního obsahu – např. frekvenční analýza

Systém LabVIEW počítá s tím, že fyzikální signály se přivádějí do počítače nebo z něj vedou prostřednictvím určitého konkrétního zařízení (např. měřicí karty) s využitím potřebného programového vybavení – ovladače (*driver*), které současně provede konfiguraci zařízení a jeho datových kanálů. Neustále vyvíjený a postupně rozšiřovaný ovladač pro DAQ zařízení firmy National Instruments se nazývá NI-DAQmx. [29]





Obr. 2.14: Části DAQ systému

## 2.7.1 Senzory

Měření fyzikálního jevu, jako je např. teplota, intenzita světelného záření nebo síla působící na objekt, začíná senzorem. Senzor, nazývaný též převodník, přeměňuje fyzikální jev na měřitelný elektrický signál. V závislosti na typu použitého snímače může být jeho elektrickým výstupem napětí, proud, odpor nebo jiný elektrický atribut, který se v průběhu času mění. Některé senzory mohou vyžadovat dodatečné komponenty a obvody, aby správně vytvořily signál, který může být přesně a bezpečně čten DAQ zařízením. [31]

Tab. 2.3: Běžné senzory pro měření fyzikálních jevů [31]

Senzor	Fyzikální jev
termočlánek, RTD, termistor	teplota
foto senzor	světlo
mikrofon	zvuk
tenzometr, piezoelektrický snímač	síla a tlak
potenciometr, LVDT, optický snímač	poloha a posunutí
akcelerometr	zrychlení
pH elektroda	pH

## 2.7.2 DAQ zařízení

Hardware DAQ funguje jako rozhraní mezi počítačem a signály z vnějšího světa. Především funguje jako zařízení, které digitalizuje příchozí analogové signály tak, aby je mohl počítač interpretovat. Tři klíčové součásti DAQ zařízení, které se používají k měření signálu, jsou obvody kondicionování signálu, analogově-digitální převodník (ADC – *Analog-to-Digital*

*Converter*) a počítačová sběrnice. Mnoho DAQ zařízení obsahuje další funkce pro automatizaci měřicích systémů a procesů. Třeba, digitálně-analogové převodníky (DACs – *Digital-to-Analog Converters*) produkují výstupní analogové signály, digitální I/O linky pro vstupní a výstupní digitální signály, a čítače/časovače počítají a generují digitální impulsy. [31]

**Kondicionování signálu** – Signály ze senzorů nebo z okolního světa mohou obsahovat šum nebo být příliš nebezpečné pro přímé měření. Obvody kondicionování signálu manipulují signál do formy, která je vhodná pro vstup signálu do ADC. Tento obvod může zahrnovat zesílení, útlum, filtrování a izolaci. Některá DAQ zařízení obsahují zabudované kondicionování signálu navržené pro měření specifických typů senzorů. [31]

**Analogově-digitální převodník (ADC)** – Analogové signály ze senzorů musí být převedeny na digitální, než jsou manipulovány digitálním zařízením jako je počítač. ADC je čip, který poskytuje digitální reprezentaci analogového signálu v okamžiku v čase. V praxi se analogové signály nepřetržitě mění v průběhu času a ADC provádí periodické "vzorky" signálu s předem definovanou rychlostí. Tyto vzorky jsou přeneseny do počítače přes počítačovou sběrnici, kde je původní signál rekonstruován ze vzorků v softwaru. [31]

**Počítačová sběrnice** – DAQ zařízení se připojují k počítači prostřednictvím slotu či portu. Počítačová sběrnice slouží jako komunikační rozhraní mezi DAQ zařízením a počítačem pro předávání instrukcí a naměřených dat. DAQ zařízení jsou nabízena pro nejběžnější počítačové sběrnice včetně USB, PCI, PCI Express a Ethernetu. Novější DAQ zařízení mají k dispozici 802.11 Wi-Fi pro bezdrátovou komunikaci. Existuje mnoho typů sběrnic a každá z nich nabízí různé výhody pro různé typy aplikací. [31]

### 2.7.3 Počítač v DAQ systému

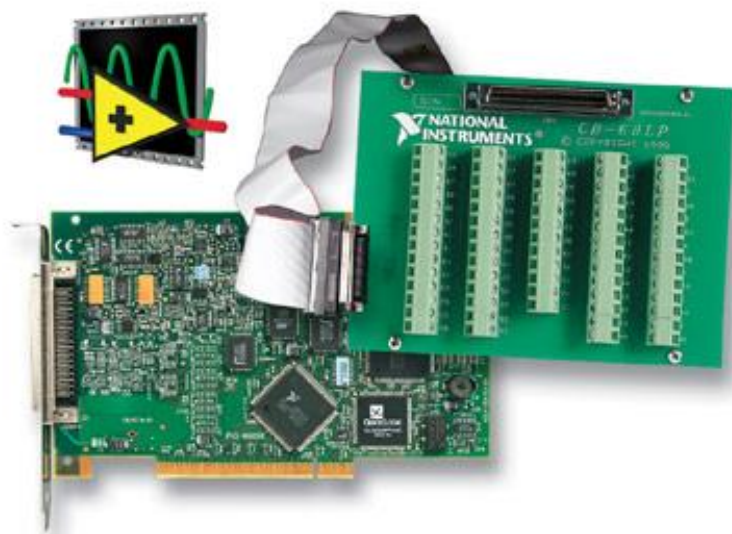
Počítač s programovatelným softwarem řídí operace DAQ zařízení a používá se pro zpracování, vizualizaci a ukládání měřených dat. V různých typech aplikací se používají různé typy počítačů. Např. v laboratoři může být použit stolní počítač pro jeho výpočetní výkon, v terénu může být používán notebook pro jeho přenositelnost, nebo může být použit průmyslový počítač ve výrobním závodě pro jeho odolnost. [31]

**Ovladač zařízení** – Programové rozhraní k danému technickému zařízení s přihlédnutím ke specifickým vlastnostem konkrétní platformy počítače. Poskytuje aplikačnímu softwaru schopnost interagovat s DAQ zařízením. Zjednodušuje komunikaci s DAQ zařízením abstrahováním hardwarových příkazů nízké úrovně a programováním na úrovni registrů. Typicky, ovladač DAQ zařízení vystavuje aplikační programovací rozhraní, které se používá v rámci programovacího prostředí pro vytváření aplikačního softwaru. [31]

**Aplikační software** – Usnadňuje interakci mezi počítačem a uživatelem pro získávání, analyzování a prezentování naměřených dat. Jedná se buď o předem připravenou aplikaci s předdefinovanými funkcemi, nebo programovací prostředí pro vytváření aplikací s vlastní funkcí. Vlastní aplikace jsou často používány k automatizaci více funkcí DAQ zařízení, provádění algoritmů zpracování signálu a zobrazení vlastních uživatelských rozhraní. [31]

## 2.8 NI PCI-6221/CB-68LP Academic Starter Kit

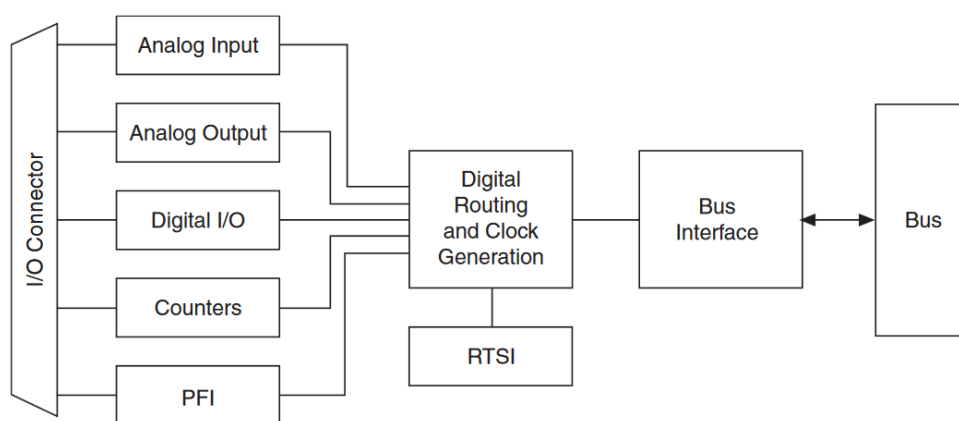
Akademická startovací sada (*Academic Starter Kit*) PCI-6221/CB-68LP od společnosti National Instruments využívá technologii řady M pro poskytnutí vysokého výkonu, spolehlivého pořizování dat a schopnosti splnit široký rozsah aplikačních požadavků. Sada NI (viz *Obr. 2.15*) je složena ze softwaru LabVIEW, multifunkční karty pro získávání dat PCI-6221, datového kabelu a konektorového bloku CB-68LP.



**Obr. 2.15:** Multifunkční karta, kabel a konektorový blok [32]

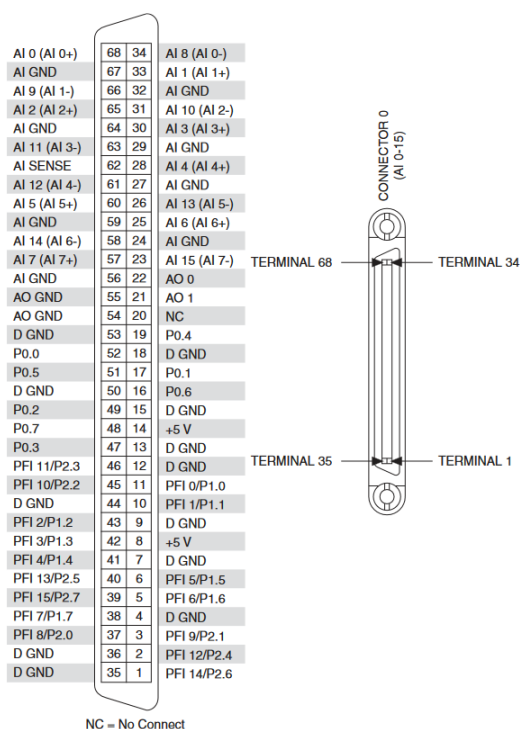
### 2.8.1 Multifunkční karta NI PCI-6221

Blokové schéma pro všechna zařízení z řady M, do které spadá i multifunkční karta NI PCI-6221 je znázorněno na *Obr. 2.16*. Karta je vybavena 16 analogovými vstupy (*Analog Input* – AI 0 až AI 15), 2 analogovými výstupy (*Analog Output* – AO 0 a AO 1), celkem 24 obousměrnými číslicovými I/O/PFI (vstupy/výstupy/programovatelnými funkčními rozhraními) rozdělenými na tři porty (P0.0 až P0.7), (PFI 0/P1.0 až PFI 7/P1.7) a (PFI 8/P2.0 až PFI 15/P2.7), dvěma víceúčelovými 32-bitovými čítači/časovači, jedním frekvenčním generátorem (libovolné PFI nebo RTSI – *Real-Time System Integration*), jednou fázově uzamčenou smyčkou pro generování vnitřní časové základny a možností využít externích digitálních spouštěčů (libovolné PFI nebo RTSI).



**Obr. 2.16:** Blokové schéma karty PCI-6221 [33]

I/O analogové nebo digitální signály se přivádějí/odvádějí na/z jednotlivé/vých piny/ů konektoru karty NI PCI-6221 (viz *Obr. 2.17*), která je umístěna v počítačové sběrnici PCI na základní desce uvnitř počítačové skříně, skrz datový kabel připojený ke konektorovému bloku CB-68LP.



Obr. 2.17: Rozmístění signálů na pinech konektoru karty NI PCI-6221 [33]

Analogové vstupní signály lze zapojit buď jako 16 nesymetrických kanálů (RSE – *Referenced Singl-Ended*) kde je připojen kladný pól např. na vstup AI 0 a záporný na zem označenou GND, nebo jako 8 symetrických kanálů (DIFF – *Differential*) kde je připojen kladný pól např. na vstup AI 0+ a záporný na vstup AI 0-.

Získávací režim karty (*Acquisition Mode*) lze nastavit buď na získání jednoho vzorku (*1 sample*) na základě požadavku (*On Demand*) či na základě hardwarových hodin (*HW Timed*), konečného počtu vzorků (*N Samples*) nebo získávání vzorků nepřetržitě (*Continuous Samples*).

**Pozn.:** více informací lze získat z manuálů dostupných z referencí [33],[34].

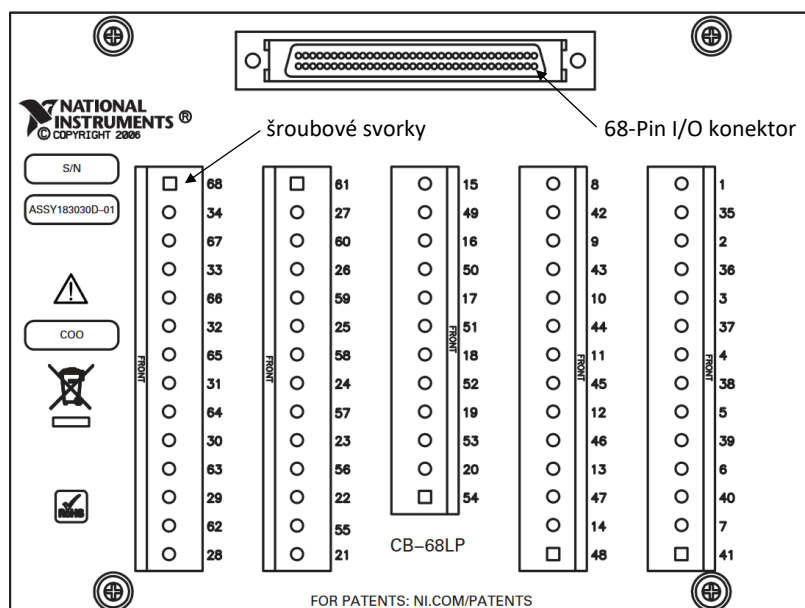
Typ zapojení jednotlivých kanálů a režim karty se nastavuje v LabVIEW při vytváření úkolu (*Task*). Nejjednodušším způsobem vytvoření požadovaného úkolu (např. měření napětí) je použití tzv. expresního VI s názvem DAQ Assistant. Po umístění expresního VI do blokového diagramu se automaticky zobrazí konfigurační okno, kde se vybere získávání nebo generování signálu, jeho typ a úkol. Následuje okno s výběrem fyzického kanálu, na který je nebo bude připojen signál. V posledním okně se zobrazí obrazovka pro konfiguraci úlohy, kde jsou následující záložky s nastavovanými parametry:

- Configuration
  - Channel Setting – správa a přidávání virtuálních kanálů
  - Time Setting – výběr režimu pro získávání či generování dat
  - Signal I/O Setup – rozsah signálu, typ zapojení
- Triggering – možnost nastavit způsob spouštění pomocí digitálního vstupu
- Advanced Timing – nastavení způsobu vzorkování a dobu ukončení měření
- Logging – možnost nastavit ukládání do TDMS (*Technical Data Management Streaming*) souboru

### 2.8.2 Konektorový blok CB-68LP

Konektorový blok s označením CB-68LP tvoří důležitou součást dodávaného příslušenství v rámci akademické startovací sady, díky jeho pomoci, je snadné přivádět nebo odvádět měřené či generované signály s využitím šroubových svorek do kterých lze vodiče jednoduše přišroubovat. Diagram konektorového bloku s rozmístěním jeho hlavních součástí, které tvoří jeden 68 pinový I/O konektor a pět svorkovnic s příslušným číselným označením jednotlivých šroubových svorek je zobrazen na *Obr. 2.18*.

**Pozn.:** více informací lze získat z manuálu, který je dostupný z reference [35].



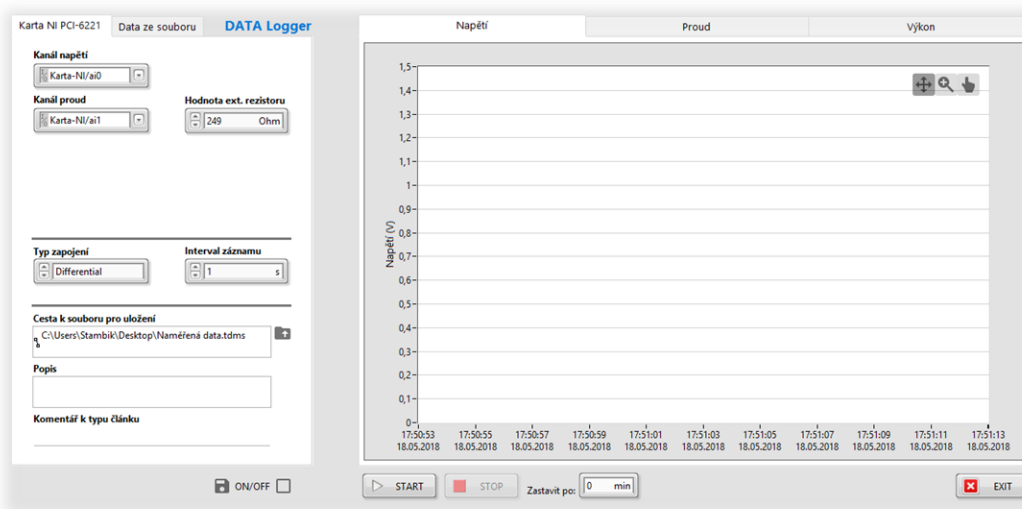
**Obr. 2.18:** Konektorový blok CB-68LP

### 3 Realizace systému pro měření a analýzu dat

K realizaci systému pro měření a analýzu dat bylo využito vývojové prostředí LabVIEW 2017. Program je pojmenován jako „DATA Logger v1.0“ a primárně je určen k měření multifunkční kartou NI PCI-6221, která byla popsána v kapitole 2.8.1. Vytvořený program je schopen kontinuálně měřit elektrické veličiny FVČ (napětí, proud a výkon) s možností definice časového intervalu a doby měření. Měřená data lze ukládat do souboru ve formátu TDMS, která mohou být zpětně zobrazena a analyzována.

#### 3.1 Uživatelské prostředí

Náhled do grafického uživatelského rozhraní (GUI – *Graphical User Interface*) virtuálního zařízení pro měření a analýzu dat vytvořeného pomocí vývojového prostředí LabVIEW představuje Obr. 3.1. Prostředí charakterizuje rozdělení na dvě hlavní části, přičemž první umožňuje volbou záložky přepínat mezi získáváním dat s využitím multifunkční karty NI PCI-6221 nebo zobrazením naměřených dat ze souboru. Druhá část prostředí zabírá největší plochu z GUI a je věnována grafickému vykreslování intervalově získávaných hodnot či zobrazení již naměřených dat. Přepínání mezi jednotlivými okny s průběhy elektrických veličin (napětí, proudu a výkonu) je řešeno opět záložkami. Ve spodní části jsou umístěny grafické ovládací prvky umožňující řízení činnosti programu.



Obr. 3.1: Uživatelské prostředí DATA Loggeru v1.0

Záložka s názvem „Karta NI PCI-6221“ je pro zpřehlednění rozdělena dekorativními prvky na tři sekce. První sekce obsahuje volbu fyzických kanálů a případnou možnost nastavení doplňujících parametrů. Je koncipována tak, aby se daly v případě upgradu programu snadno přidávat další kanály. Druhá sekce slouží pro nastavení typu zapojení a intervalu záznamu získávaných dat. Poslední sekce je věnována možnosti nastavení ukládání do souboru.

## 3.2 Ovládání programu

Po spuštění VI s názvem „DataLogger v1.0.vi“ lze zvolit záložku, ve které je k dispozici nastavení multifunkční karty NI PCI-6221 a ukládání do souboru, nebo záložku s volbou cesty k souboru s již uloženými daty. Pro každou záložku jsou ve spodní části obrazovky zobrazeny jiné prvky pro ovládání, kromě tlačítka *Exit*.

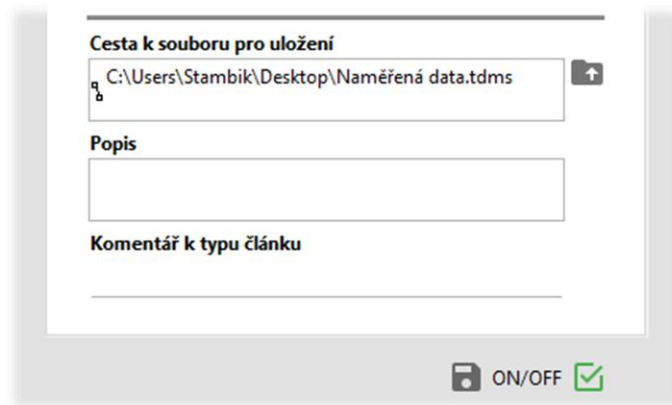
### 3.2.1 Nastavení parametrů

Před spuštěním DAQ je nutné nastavit potřebné vstupní parametry. Mezi tyto parametry patří především nastavení fyzických kanálů pro měření napětí, proudu a způsobu jejich zapojení. Pro kanál proudu je navíc nutné zadat hodnotu externího rezistoru, neboť multifunkční karta, která měří úbytek napětí na tomto rezistoru a následně ho převádí na velikost proudu nemá tento rezistor zabudovaný. Volba typu zapojení je společná pro oba dva měřené kanály a je zde na výběr buď DIFF nebo RSE způsob zapojení. Dodatečně lze nastavit interval záznamu, kde je minimální a zároveň výchozí hodnota jedna sekunda. Nastavení ukládání do souboru se věnuje samostatná podkapitola 3.2.2. Také je možnost nastavit dobu po kterou mají být data získávána. Jako výchozí hodnota je nastavena nula, tzn. kontinuální získávání dat a tuto hodnotu lze za běhu měnit.

### 3.2.2 Ukládání dat

Před spuštěním běhu získávání dat či kdykoliv během průběhu lze zapnout nebo vypnout ukládání dat do souboru. Zapnutí, resp. vypnutí ukládání se provádí zaškrtnutím políčka vedle ikony v podobě diskety s popisem „ON/OFF“. Signalizace zapnutého stavu je provedena zaškrtnutím a změnou barvy na zelenou (viz Obr. 3.2).





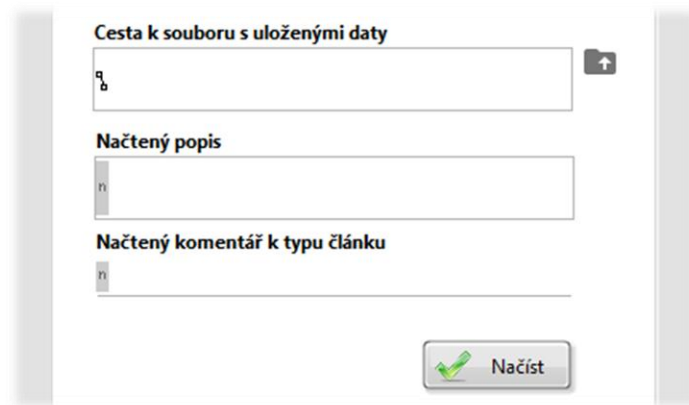
**Obr. 3.2:** Nastavení ukládání do souboru

Ještě před samotným začátkem DAQ je nutné vybrat cestu k souboru pro zápis těchto dat. Ve výchozím stavu je cesta nastavena na uživatelskou plochu, kde se automaticky při zapnutém ukládání a stisku tlačítka **Start** nebo zapnutí ukládání v průběhu vytvoří soubor s názvem „Naměřená data.tdms“. Lze vytvořit i vlastní soubor a to tak, že přes ikonku v podobě složky s bílou šipkou se zvolí cesta do složky, kde se má soubor uložit a následně přepsat výchozí název souboru. Možností je také výchozí soubor pouze přejmenovat a data se pak budou ukládat opět na uživatelskou plochu pod jiným názvem, nebo cestu k souboru napsat ručně. Ve všech případech je ale důležité zachovat příponu .tdms na konci pojmenovaného souboru.

K uloženému souboru je možnost vyplnit popis a také je vhodné zapsat komentář k typu článku. Při ukládání více prováděných měření do jednoho souboru komentář totiž zastupuje druhou úroveň hierarchie TDMS souboru (*Channel Group*). Podle vyplněného komentáře jsou tedy data zapsána ke stejné skupině nebo se vytvoří nová.

### 3.2.3 Zobrazení dat

Volba a zobrazení uložených dat, která je k vidění na *Obr. 3.3* má téměř identickou grafickou podobu s ukládáním do souboru. Jako první je na hlavní obrazovce programu nutné překliknout na záložku s názvem „Data ze souboru“ a následně zde určit cestu k uloženému souboru ve formátu TDMS. Možnosti pro určení cesty jsou obdobné jako při ukládání do souboru čili přes složku s bílou šipkou nebo ručním zápisem. Kliknutím na tlačítko **Načíst** se provede zobrazení těchto dat. Zároveň je načten popis a komentář k typu článku, pokud jsou tyto informace k dispozici.



Obr. 3.3: Volba a zobrazení uložených dat

Pokud je ve zvoleném souboru uloženo více prováděných měření, zobrazí se dialogové okno s rozbalovacím menu a požadavkem na výběr skupiny pro zobrazení. V případě nevyplnění cesty k souboru je uživateli zobrazeno dialogové okno s informací, že nebyl vybrán vstupní soubor. U zobrazených dat v grafu je ve spodní části obrazovky tlačítko **Smazat**, které umožňuje rychlejší a současně smazání vykreslených průběhů ve všech třech záložkách (napětí, proud a výkon).

### 3.2.4 Běh DAQ

Spuštění DAQ se provádí stiskem tlačítka **Start** a jeho aktivita je reprezentována změnou na zelenou barvu. Současně je povoleno používání tlačítka **Stop**, které bylo do této doby zakázáno a zešedlé. Navíc je deaktivována levá část prostředí jako ochrana před změnou nastavených parametrů za běhu DAQ. V této fázi běží DAQ a příp. ukládání do souboru s nastaveným intervalem záznamu. Program nyní dovoluje nastavovat pouze tyto parametry:

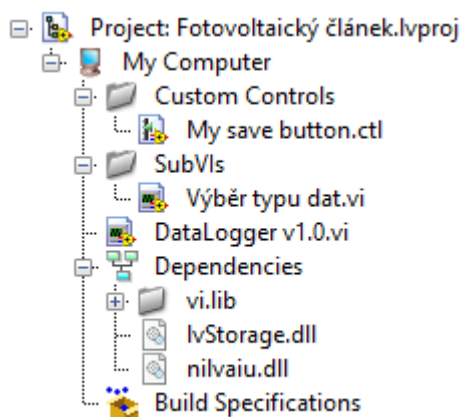
- Zapnutí/vypnutí ukládání do souboru
- Změnu doby zastavení
- Přepínání záložek s průběhy elektrických veličin

DAQ se zastaví buď automaticky po uplynutí nastavené doby zastavení nebo ihned po stisku tlačítka **Stop**. Po zastavení DAQ je zobrazeno dialogové okno s možností vyvolat *TDMS File Viewer*. Tlačítka jsou vrácena do původního stavu a aktivuje se levá část prostředí pro možnost změny nastavovaných parametrů. Funkce tlačítka **Exit** je nastavena na ukončení běhu VI, kterou lze při případném vytváření spustitelného souboru s příponou .exe změnit

na ukončení celého programu.

### 3.3 Popis částí programu

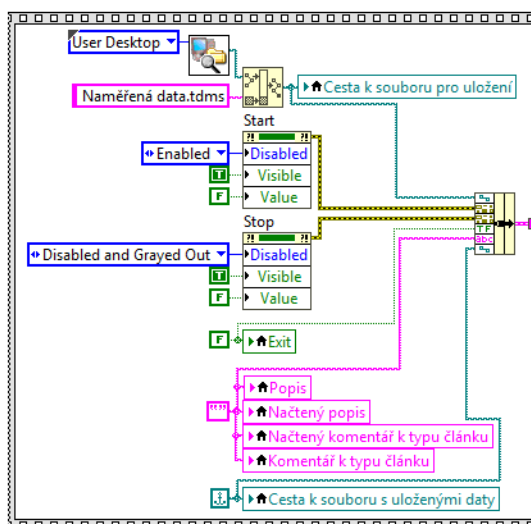
Obsahem této kapitoly bude popis blokových diagramů nejdůležitějších částí, které byly použity pro vytvoření VI ve vývojovém prostředí LabVIEW 2017. Celý systém pro měření a analýzu dat se nachází v *Project Explorer* s názvem „Fotovoltaický článek.lvproj“ (viz *Obr. 3.4*), který sjednocuje všechny důležité součásti potřebné pro správný běh vytvořeného programu s názvem „DataLogger v1.0.vi“. Pro přehlednost jsou vytvořeny virtuální složky s názvy „*Custom Controls*“ a „*SubVIs*“ obsahující vlastní vytvořené ovládací prvky a podprogramy. Důvodem pro volbu projektu namísto samostatného VI bylo to, aby byla možnost pozdějšího přidávání dalších součástí a bylo s nimi snadnější pracovat.



**Obr. 3.4:** *Project Explorer*

#### 3.3.1 Počáteční inicializace při spuštění VI

Jako první je vždy při novém spuštění VI provedena počáteční inicializace vstupních proměnných. Zajišťuje návrat všech hodnot a akcí provedených ve spuštěném programu na původní hodnotu a stav. Na *Obr. 3.5* je možno vidět konkrétní prvky umístěné v programové struktuře *Flat Sequence*, které procházejí touto počáteční inicializací. Struktura je použita právě k zajištění prvotního průběhu před ostatními programovými strukturami. Výstupy jsou sbaleny funkcí *Bundle* do jednoho výstupního *Clusteru* a poslány k dalšímu zpracování.



Obr. 3.5: Počáteční inicializace při spuštění VI

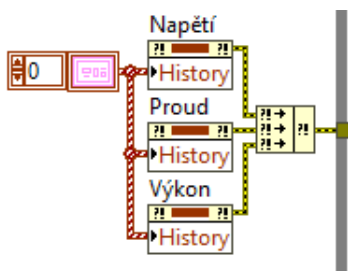
Uvnitř struktury je nastavena výchozí cesta na uživatelskou plochu, kde se při ukládání vytvoří soubor s názvem „Naměřená data.tdms“. Tlačítkům *Start* a *Stop* jsou přes *Property Node* (uzel pro programové řízení vlastností) nastaveny tyto vlastnosti:

- zda jsou povolené/zakázané/zakázané a zešedlé,
- nebo viditelné/neviditelné,
- a jestli mají nabývat hodnoty *True* či *False*.

Dále jsou ve struktuře umístěny lokální proměnné pro tlačítko *Exit*, textová pole a cestu k souboru. Lokálním proměnným pro textová pole se předá konstanta v podobě prázdného stringu (s nulovou délkou) a pro cestu k souboru je nastavena konstanta s prázdnou cestou.

### 3.3.2 Programové mazání grafů

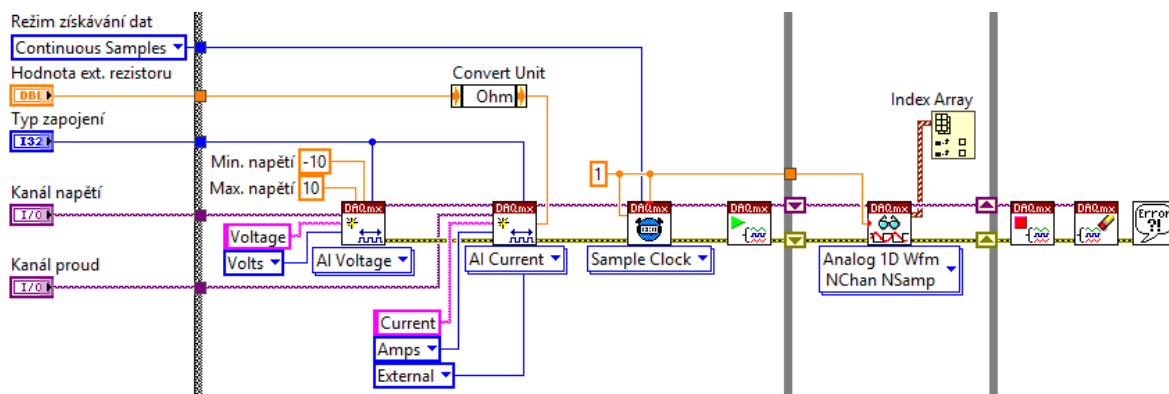
Programové mazání grafu zajišťuje *Property Node* s odkazem na zobrazovaný průběh, který je nastaven na vlastnost *History Data*. Jelikož se vykreslují tři průběhy, jsou i zde použity tři uzly vlastností, každý s odkazem na konkrétní průběh. Do každého uzlu vstupuje konstanta v podobě *Clusteru*, který obsahuje nastavované atributy specifické pro datový typ *waveform* (viz kap. 2.6.1). Z jednotlivých uzlů jsou vyvedeny výstupní chyby a sloučeny funkcí *Merge Errors*. Výstup této funkce je připojen na následující smyčku *While*. Tím je zajištěno, že se při stisku tlačítka *Start* automaticky smažou předešlé průběhy ještě předtím, než se začnou vykreslovat nové.



Obr. 3.6: Programové mazání historie grafů

### 3.3.3 DAQ multifunkční kartou

Získávání dat s využitím multifunkční karty na Obr. 3.7 je řešeno pomocí *Low Level VIs* z knihovny DAQmx – Data Acquisition. Ty mají na rozdíl od *Express VIs* např. v podobě DAQ asistentu výhodu v tom, že umožňují změnu nastavení jejich parametrů i za běhu VI z čelního panelu a také nabízejí většinou více možností uživatelského nastavení.



Obr. 3.7: Získávání dat multifunkční kartou

Po stisku tlačítka *Start* je přivedena hodnota *True* na připojený *Case Selector* a nastavené parametry vstupují do struktury *Case*, kde jsou předány *Low Level VIs*. Jako první je vytvořen virtuální kanál pro měření napětí pomocí polymorfního *DAQmx Create Virtual Channel.vi* nastaveného na *AI Voltage*. Zároveň VI vytvoří nový vstupní *Task*, což je sbírka jednoho nebo více virtuálních kanálů s časováním, spouštěním a dalšími nastavenými vlastnostmi. Konceptně, *Task* představuje měření nebo generaci, která má být provedena. Vytvořenému virtuálnímu kanálu je přiřazen fyzický kanál, jeho pojmenování, jednotka a způsob zapojení. Je nastaven také rozsah vstupního napětí na hodnotu  $\pm 10$  V, což umožňuje měřit do max. 20 V při DIFF způsobu zapojení.

Jako druhý je vytvořen virtuální kanál pro měření proudu nastavený na *AI Current*, který má

přiřazen fyzický kanál, pojmenování, jednotku a způsob zapojení společný s předchozím virtuálním kanálem pro měření napětí. Dodatečnými nastavovanými parametry jsou typ umístění rezistoru a jeho velikost, která se zadává přes *Control* na čelním panelu. Před vstupem do virtuálního kanálu musí být tato velikost převedena funkcí *Convert Unit* z fyzického čísla (číslo s jednotkou) na čisté číslo (číslo bez jednotky).

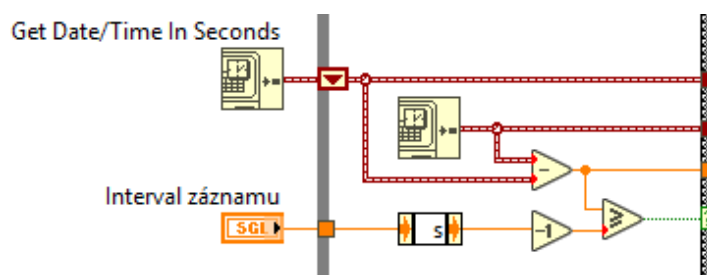
Třetím v pořadí je polymorfní *DAQmx Timing.vi*, který konfiguruje počet vzorků k získání nebo generování a v případě potřeby vytváří vyrovnávací paměť. Je nastaven na instanci *Sample Clock*, pro kterou jsou nastaveny parametry režimu získávání dat, vzorkovací rychlost a počet vzorků na kanál. Režim získávání dat je nastaven na kontinuální získávání vzorků a na vstupní terminály pro vzorkovací rychlost a počet vzorků na kanál je připojena konstanta jedna. Připojením konstanty s touto hodnotou je zajištěno, že je získán jeden vzorek s periodou jedné sekundy.

Dalším v pořadí je *DAQmx Start Task.vi*, který převádí příchozí *Task* do spuštěného stavu, tzn. aby zahájil měření nebo generaci. Spuštěný *Task* vstupuje do smyčky *While*, kde jsou jeho vzorky čteny polymorfním *DAQmx Read.vi* nastaveného na instanci *Analog 1D Wfm NChan NSamp*, tudíž čte jeden nebo více průběhů z úkolu, který obsahuje jeden nebo více analogových vstupních kanálů. Jediným nastavovaným parametrem je zde počet vzorků na kanál, přičemž je jeho hodnota nastavena konstantou obsahující jedničku. Výstupem tohoto VI je 1D pole datového typu *Waveform*, které je přivedeno na funkci *Index Array*. Funkce rozdělí přivedené 1D pole na dvě samostatná subpole pro napětí a proud obsahující pouze jeden element, neboť pořizujeme pouze jeden vzorek. Zpracovaná data jsou dále předána struktuře *Case* zajišťující zobrazení a zápis dat do souboru.

Za smyčkou *While* se nachází *DAQmx Stop Task.vi*, který zastaví *Task* a vrátí ho do stavu, ve kterém byl před spuštěním. Na konci celého řetězce se nachází *DAQmx Clear Task.vi*, který *Task* vymaže.

### 3.3.4 Interval záznamu a ukládání

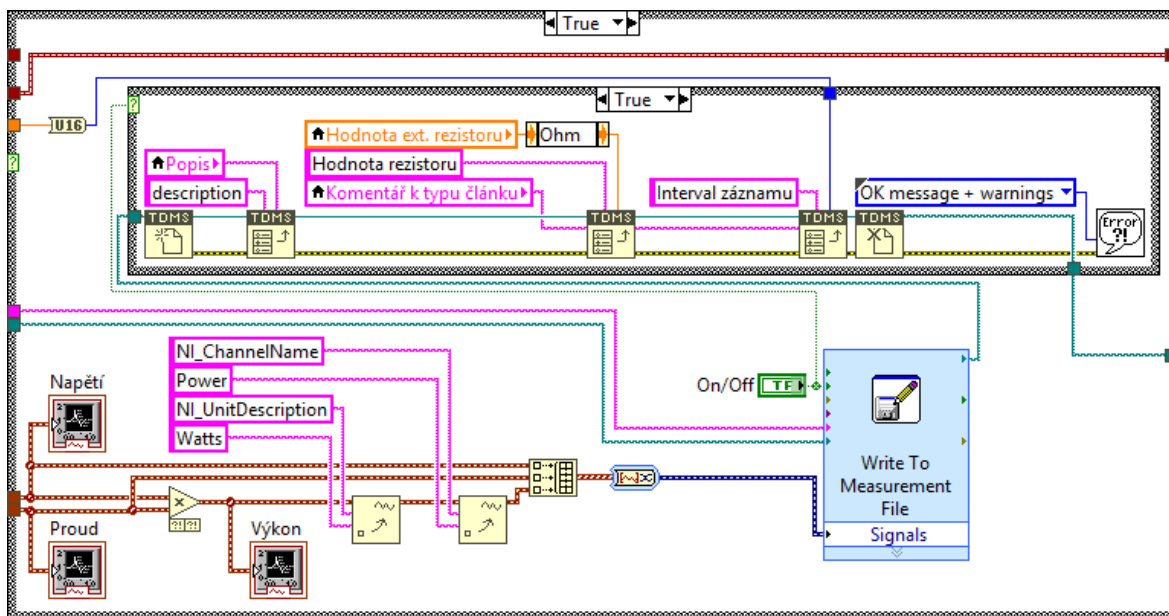
Základem pro algoritmus časování smyčky *While* (viz Obr. 3.8), která zajišťuje vykreslování a příp. ukládání do souboru jsou použity dvě funkce *Get Date/Time In Seconds*. Funkce vrátí časové razítko aktuálního času vypočítané jako počet sekund, které uplynuly od 12:00 hodin, pátku, 1. ledna 1904 světového času. Při nulté iteraci (prvním cyklu) smyčky *While* vstupuje získaný aktuální čas v sekundách do této smyčky, kde je následně proveden rozdíl hodnoty aktuálního času uvnitř smyčky a tohoto vstupního času. Následuje porovnání výstupní hodnoty rozdílu a zadaného intervalu záznamu, který musí být nejprve převeden funkcí *Convert Unit*. Porovnání je provedeno funkcí *Greater Or Equal?* a výsledek je v podobě datového typu *Boolean* (*True* nebo *False*) přiveden na *Case Selector* struktury *Case*, která řídí zobrazování a ukládání dat. Po skončení iterace je pomocí *Shift Register* (posuvného registru) výstupní hodnota časového razítka přenesena na vstup smyčky *While* a celý proces se opakuje.



Obr. 3.8: Algoritmus pro časování smyčky *While*

### 3.3.5 Zobrazování a zápis dat do souboru

Základem pro část blokového diagramu řídicího zobrazování a zápis dat do souboru jsou dvě programové struktury typu *Case* (viz Obr. 3.9). Vstupní data pro vnější strukturu čekají na příchozí hodnotu *True* přivedenou na *Case Selector* z algoritmu zajišťujícího časování smyčky *While*, ve které jsou tyto struktury umístěné.



Obr. 3.9: Blokový diagram pro zobrazování a zápis dat do souboru

Příchozí elementy pro napětí a proud v podobě datového typu *Waveform* jsou mezi sebou pro získání hodnoty výkonu vynásobeny. Pro zobrazování aktuálních hodnot napětí, proudu a výkonu jsou použity grafické zobrazovače *Waveform Charts*, nastavené na výchozí aktualizací režim *Strip Chart*, ve kterém jsou zobrazovaná běžící data plynule posouvána zleva doprava napříč grafem se starými daty vlevo a novými daty vpravo. Tyto grafické zobrazovače jsou speciálním typem číselného indikátoru, který zobrazuje jeden nebo více průběhů dat typicky získaných při konstantní rychlosti. Uchovává si historii dat nebo vyrovnávací paměť z předchozích aktualizací.

Všechny tři datové typy *Waveform* jsou poté sloučeny funkcí *Build Array* do 1D pole s tím, že jsou ještě před sloučením datovému typu *Waveform* obsahujícímu informace o průběhu výkonu přidány některé doplňující atributy. To je provedeno funkcí *Set Waveform Attribute* a je zde nastaveno jméno kanálu na *Power* atributem *NI\_ChannelName* a jednotka na *Watts* atributem *NI\_UnitDescription*. Pole je dále převedeno na dynamický datový typ s využitím expresního VI *Convert to Dynamic Data* s nastavením vstupního datového typu na *1D Array Of Waveforms* a tento převod je důležitý pro možnost použití s následujícím expresním VI.

Pro zápis dat do souboru je použito expresní VI s názvem *Write To Measurement File*, ve kterém jsou nastaveny některé základní parametry, např. formát souboru. Na vstupní terminály je přiveden komentář, cesta k souboru pro uložení, ukládané signály a hodnota ovladače (*True* nebo *False*) umístěného na čelním panelu, kterým se zapíná/vypíná ukládání



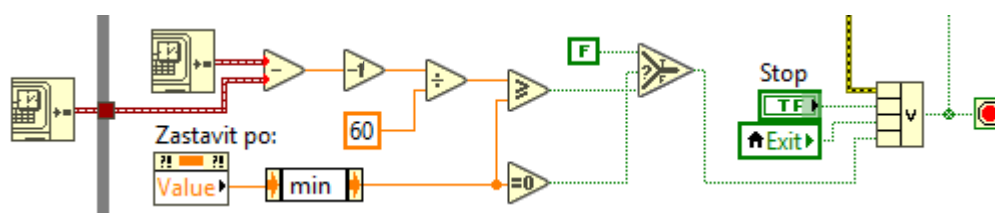
čili činnost tohoto VI. Příchozí hodnota tohoto ovladače rozhoduje také o stavu vnitřní struktury *Case*, neboť je přivedena na její *Case Selector*. Výstupem z expresního VI je soubor ve formátu TDMS se zapsanými daty dle nastavených parametrů a je přiveden na vstup vnitřní struktury *Case* pro další zpracování.

Jak již bylo zmíněno dříve v kapitole 3.3.3, expresní VI neumožňují specifickější nastavení parametrů. Proto je použita vnitřní struktura *Case*, ve které jsou umístěny funkce zajišťující nastavení požadovaných vlastností výstupnímu TDMS souboru z expresního VI. Struktura je nastavena na hodnotu *True* pouze tehdy, je-li povoleno ukládání, neboť se až zároveň při povoleném ukládání vytvoří nový soubor. Tento soubor je potřebný pro první funkci ve vnitřní struktuře *Case* s názvem *TDMS Open* a pokud by nebyl k dispozici, byla by vrácena chyba programu.

Pokud je tedy zapnuto ukládání do souboru, je struktura *Case* nastavena na *True* a funkcí *TDMS Open* je otevřen .tdms soubor pro čtení nebo zápis. Další v pořadí je funkce *TDMS Set Properties*, která zapisuje vlastnosti do definované úrovně TDMS souboru. Pokud není úroveň definována, jsou vlastnosti zapsány do nejvyšší úrovně *File*. První použitá funkce tak zapisuje do úrovně *File* ke jménu vlastnosti *Description* hodnotu, která je přivedena formou lokální proměnné se jménem „Popis“. Další dvě stejné funkce jsou nastaveny na úroveň *Group* TDMS souboru. Jméno skupiny je určeno lokální proměnnou se jménem „Komentář k typu článku“. První funkce ke jménu vlastnosti s názvem „Hodnota rezistoru“ zapíše hodnotu externího rezistoru příchozí přes lokální proměnnou, která je nejprve převedena funkcí *Convert Unit*. Druhá funkce ke jménu vlastnosti s názvem „Interval záznamu“ zapíše hodnotu, která přijde od algoritmu pro časování smyčky *While*. Tato příchozí hodnota je nejprve převedena funkcí *To Unsigned Word Integer* na celé číslo bez znaménka. Poslední v řetězci je funkce *TDMS Close*, která zavře .tdms soubor otevřený na začátku řetězce funkcí *TDMS Open*.

### 3.3.6 Zastavení DAQ, zobrazování a ukládání

Diagram zastavení smyčky *While* (viz Obr. 3.10) využívá opět dvě funkce *Get Date/Time In Seconds*, které byly popsány již v kapitole 3.3.4. Pro převod hodnoty jejich rozdílu ze sekund na minuty je použita funkce pro dělení s připojenou číselnou konstantou. Následně je hodnota porovnána funkcí *Greater Or Equal?* a výstup této operace v podobě *True* nebo *False* je přiveden na polymorfni funkci *Select*.



Obr. 3.10: Diagram pro zastavení smyčky *While*

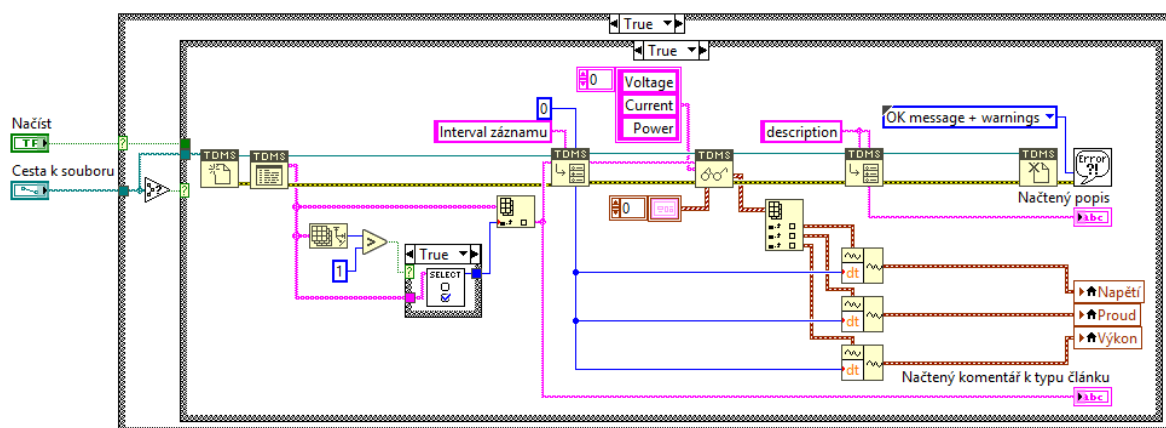
Funkce *Select* je řízena datovým typem *Boolean* a vrací hodnotu připojenou na vstup s označením *t* nebo *f* v závislosti na hodnotě vstupu *s*. O tom, zda bude vstup *s* nabývat hodnoty *True* či *False* rozhoduje příchozí hodnota nastavená na čelním panelu odpovídající době zastavení, která musí být napřed převedena pomocí funkce *Convert Unit*. Výchozí nastavenou hodnotou je zde nula, která zajišťuje kontinuální běh smyčky i při nenastavené době zastavení, neboť je funkcí *Equal To 0?* předána funkci *Select* na vstup *s* hodnota *True*, a tudíž je výstupem z této funkce hodnota v podobě konstanty *False*. Pokud je na čelním panelu zadáno jiné číslo než nula, funkce *Equal To 0?* pošle funkci *Select* na vstup *s* hodnotu *False* a výstupem je pak příchozí hodnota z předcházejícího porovnání.

Samotný výstup z funkce *Select* je přiveden na polymorfni funkci *Compound Arithmetic* nastavenou na mód provádějící logickou operaci OR, do které vstupuje mj. příchozí chyba, hodnota tlačítka *Stop* a lokální proměnná tlačítka *Exit*. Jakákoliv hodnota *True* vstupující do funkce *Compound Arithmetic* tak způsobí zastavení smyčky *While*.

Pro možnost změny doby zastavení za běhu programu je použit *Property Node* s nastavením na vlastnost *Value* a odkazem na *Control* umístěného na čelním panelu. Každou iteraci smyčky *While* je tak tato hodnota uvnitř smyčky aktualizována.

### 3.3.7 Načtení a zobrazení dat ze souboru

Blokový diagram pro načtení dat ze souboru (viz Obr. 3.11) využívá dvě programové struktury *Case*, přičemž na *Case Selector* vnější struktury je přiveden datový typ *Boolean*, který zahajuje celý proces v případě příchozí hodnoty *True*. Vnitřní struktura vykoná zobrazení dat za předpokladu, že funkce *Is Path and Not Empty?* vrátí hodnotu *True*. Tato funkce vrací hodnotu *True*, pokud je cesta k souboru jakákoliv jiná hodnota než *<Not A Path>* nebo prázdná cesta.



Obr. 3.11: Blokový diagram pro načtení a zobrazení dat ze souboru

První v pořadí je funkce *TDMS Open* otevírající *.tdms* soubor a hned za ní následuje funkce *TDMS List Contents*, která poskytuje seznam názvů skupin a kanálů obsažených ve vstupním souboru. Z této funkce jsou vzaty pouze jména skupin v podobě 1D pole datového typu *string* a je předáno funkcím *Index Array*, *Array Size* a na vstup do struktury *Case*. Nejprve je funkcí *Array Size* vráceno celé číslo, které je rovno počtu skupin obsažených v souboru. Toto číslo je následně porovnáno funkcí *Greater?* s číselnou konstantou jedna a výsledek v podobě *True* nebo *False* je předán na *Case Selector* struktury *Case*. V případě, že je obsažena pouze jedna skupina, porovnání vrátí hodnotu *False* a výstupem ze struktury *Case* je pak konstanta v podobě číselné hodnoty nula. Pokud je výsledkem porovnání hodnota *True*, tzn. je obsažena více než jedna skupina, je zavoláno *SubVI* s názvem „SELECT“. Toto jednoduché *SubVI* zobrazí uživateli dialogové okno s výběrem skupiny pro vykreslení, jehož výstupem je index odpovídající zvolené skupině. Funkce *Index Array* vybere ze vstupního 1D pole skupinu v závislosti na přivedeném indexu ze struktury *Case* a následně ji pošle funkci *TDMS Get Properties* a její jméno zobrazí na čelním panelu.

Funkce *TDMS Get Properties* získá z úrovně *Group* TDMS souboru hodnotu, přiřazenou ke jménu vlastnosti „Interval záznamu“ a pošle ji funkcím *Build Waveform*. Následující funkcí je *TDMS Read*, která má pomocí vstupu specifikovaný datový typ. Tato funkce čte data ze zvolené skupiny z úrovně *Channel* TDMS souboru. Jména kanálů, které mají být čteny jsou přivedeny v podobě 1D pole typu *String*. Výstupní data jsou předána funkci *Index Array*, která je rozdělí na tři subpole a poté vstupují do funkcí *Build Waveform*. Funkce modifikují vstupující subpole, konkrétně jejich složku *dt* na hodnotu získanou funkcí *TDMS Get Properties*, která specifikuje časový interval v sekundách mezi datovými body. Po úpravě jsou data předána k zobrazení pomocí lokálních proměnných, přiřazených grafickým zobrazovačům pro konkrétní průběh.

Za funkcí *TDMS Read* se nachází ještě jedna funkce *TDMS Get Properties*, která má za úkol z úrovně *File* TDMS souboru získat hodnotu, přiřazenou ke jménu vlastnosti „Description“ a poslat ji zobrazovači na čelním panelu. Jako poslední v řetězci je funkce *TDMS Close*, která zavře .tdms soubor otevřený na začátku řetězce funkcí *TDMS Open*.

## Závěr

V rámci této diplomové práce jsem vytvořil systém pro kontinuální měření a analýzu dat z fotovoltaické elektrárny s využitím vývojového prostředí LabVIEW, pro které mi byla poskytnuta licence. Před samotnou realizací jsem se musel nejprve naučit základy toho programovacího jazyka.

Program jsem vyvíjel na svém přenosném počítači s nainstalovanou verzí LabVIEW 2017 a ovladačem NI-DAQmx 17.1.1, který zajišťuje komunikaci s multifunkční kartou. Karta byla při vývoji virtuálně nasimulována, a to mi umožnilo vytvořit program i bez fyzického připojení karty k PC. K realizaci jsem přistupoval s tím, že je počítáno s následným upgradem, který by mohl zahrnovat např. přidání kanálů pro měření intenzity ozáření a teploty. Bylo by také možné zahrnout měření V-A charakteristik, což jsem chtěl na začátku vývoje realizovat, ale nebyla k dispozici elektronicky řízená odporová dekáda, a tudíž jsem musel od tohoto nápadu upustit.

Vytvořený program jsem pojmenoval jako „DATA Logger v1.0“ a pro grafickou podobu čelního panelu jsem použil jednoduchý a moderní design, který využívá prvky ze subpalety *Silver* a některé další prvky z doinstalovaného zásuvného modulu. Program má intuitivní ovládání a umožňuje uživateli měřit elektrické veličiny FVČ v podobě napětí, proudu a výkonu. Data je možno ukládat a zpětně zobrazovat. Uživatel má také možnost definovat interval záznamu a dobu po kterou má být měření prováděno.

Plánovaný test pro ověření funkčnosti programu v reálných podmínkách jsem nakonec nebyl schopen realizovat, neboť byl problém se zpětnou kompatibilitou. Poskytnutá karta NI PCI-6221, kterou jsem popsal v kapitole 2.8.1 se totiž připojuje do slotu PCI stolního počítače a měla být k dispozici na školním PC. Vlivem stáří školního počítače, na kterém byl nainstalován zastaralý systém Windows 2000 se tak nepodařilo zajistit průřez softwarové kompatibility mezi verzemi LabVIEW a ovladači karty NI-DAQ.

## Seznam literatury a informačních zdrojů

- [1] MĚŘENÍ FOTOVOLTAIKY. *Tzb-energ: sdružení techniků a inženýrů ve stavebnictví a energetice* [online]. [cit. 2018-05-01]. Dostupné z: <http://www.tzb-energ.cz/mereni-fotovoltaiky.html>
- [2] CHMEL, Ladislav a Ivo HAMERNÍK. *Fotovoltaika, fototermika: fyzikální podstata fotovoltaické přeměny energie* [online]. Verze 29. 2014, aktualizace 13.10.2017 [cit. 2018-04-22]. Dostupné z: <https://publi.cz/books/91/02.html>
- [3] KUBÍČKOVÁ, Kristýna. Výzkumná zpráva o řešení dílčího tématu projektu VC ZČU od MŠMT: materiály a komponenty pro ochranu životního prostředí: proměření defektů různými dostupnými metodami. *Ftp server Solartec* [online]. Solartec, 2006 [cit. 2018-04-19]. Dostupné z: <ftp://ftp.solartec.cz/download/ZCU>
- [4] HUSÁK, Marek. *Studium polovodičových struktur nedestructivními metodami* [online]. Brno, 2009 [cit. 2018-04-19]. Dostupné z: [https://is.muni.cz/th/q1ez4/dp\\_text.pdf](https://is.muni.cz/th/q1ez4/dp_text.pdf). Diplomová práce. Masarykova univerzita, Pedagogická fakulta, Katedra fyziky.
- [5] Výroba fotovoltaických článků a panelů. *Solární energie* [online]. Kusala, 2006 [cit. 2018-04-19]. Dostupné z: <https://www.cez.cz/edee/content/microsites/solarni/k41.htm>
- [6] BANNERT, Petr. *PRAKTIKA z FOTOVOLTAIKY* [online]. Varnsdorf, 2008 [cit. 2018-04-22]. Dostupné z: [https://is.muni.cz/el/1441/jaro2011/FY2BP\\_EkO/um/praktika\\_z\\_fotovoltaiky.pdf](https://is.muni.cz/el/1441/jaro2011/FY2BP_EkO/um/praktika_z_fotovoltaiky.pdf)
- [7] BĚLÍK, Milan. *Přednášky: solární elektroenergetické systémy*. Plzeň: ZČU FEL, [2017].
- [8] HASELHUHN, Ralf. *Fotovoltaika: budovy jako zdroj proudu*. Ostrava: HEL, 2011. ISBN 978-80-86167-33-6.
- [9] HASELHUHN, Ralf a Petr MAULE. *Fotovoltaické systémy: energetická příručka: pro elektrikáře, techniky, instalatéry, projektanty, architekty, inženýry, energetiky, manažery, stavitele, studenty, učitele, ostatní odborné a profesní soukromé nebo veřejné instituce a zájemce o fotovoltaický obor a energetickou nezávislost*. Přeložila Anna ROHÁČOVÁ, přeložil Pavel ROHÁČ, přeložila Anna ŽENÍŠKOVÁ,

- přeložila Eva HOŘEJŠÍ, přeložil Jana KLAMO, přeložila Jitka HICKOVÁ, přeložila Veronika MARTINOVSKÁ, přeložil Tomáš BAROCH. Plzeň: Česká fotovoltaická asociace, 2017. ISBN 978-80-906281-5-1.
- [10] ŠVACH, Tomáš. *Simulace autonomních a hybridních FV systémů* [online]. Praha, 2016 [cit. 2018-04-20]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/65215/F3-DP-2016-Svach-Tomas-svachtomDIP%2Bzadani.pdf?sequence=-1&isAllowed=y>. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra měření.
- [11] Shockleyho rovnice diody. *KD – elektro* [online]. 2010 [cit. 2018-04-17]. Dostupné z: <https://kd-elektro.webnode.cz/news/shockleyho-rovnice-diody/>
- [12] Základní elektronické prvky a jejich modely. *HIPPO* [online]. Hospodka, 2009 [cit. 2018-04-15]. Dostupné z: <http://hippo.feld.cvut.cz/vyuka/elo/pr2/modely-bl.pdf>
- [13] Diagnostika fotovoltaických panelů. *Aiolos s.r.o: Vaše energie, Vaše peníze* [online]. [cit. 2018-05-01]. Dostupné z: <http://www.vaseenergie.cz/termovizni-mereni/termovizni-diagnostika-fotovoltaiky/>
- [14] Diagnostika a optimalizace FVE. *Energy ForEver* [online]. [cit. 2018-05-06]. Dostupné z: <http://www.energyforever.cz/cz/sluzby/poradenstvi/optimalizace-fve/>
- [15] *SOLARTEC: PV test* [online]. c2015 [cit. 2018-05-01]. Dostupné z: <http://www.pvtest.cz/cs/index>
- [16] MĚŘENÍ TERMOKAMEROU. *Tzb-energ: sdružení techniků a inženýrů ve stavebnictví a energetice* [online]. [cit. 2018-05-06]. Dostupné z: <http://www.tzb-energ.cz/mereni-termokamerou.html>
- [17] Termovizní měření FV elektráren. *Sunfin PRAHA: Výstavba, provoz a SERVIS solárních elektráren (FVE) - EPC Elektroslužby - Elektroenergetika* [online]. [cit. 2018-05-01]. Dostupné z: <http://www.sunfin.cz/cs/page/termovizni-mereni-fv-elektren/>
- [18] *Eicero: INTELLIGENT POWER* [online]. [cit. 2018-05-01]. Dostupné z: <https://eicero.cz/cs/>
- [19] Průmyslové inspekce: revize fotovoltaických panelů. *FlyMedia* [online]. [cit. 2018-05-01]. Dostupné z: <http://flymedia.cz/portfolio-item/prumyslove-inspekce/>

- [20] Letecká termovize. *WINGS: letecké práce* [online]. [cit. 2018-05-01]. Dostupné z: <https://www.letecke-prace.eu/letecka-termovize>
- [21] Kontrola FVE. *JamCopters* [online]. [cit. 2018-05-01]. Dostupné z: <https://jamcopters.cz/industry/kontrola-fotovoltaickych-elektren>
- [22] Využití dronů pro diagnostiku fotovoltaických elektráren. *Solární Novinky cz* [online]. 2015 [cit. 2018-05-07]. Dostupné z: <http://www.solarninovinky.cz/?nove-produkty/2015042601/vyuziti-dronu-pro-diagnostiku-fotovoltaickych-elektren>
- [23] Fotovoltaické elektrárny: diagnostika fotovoltaické elektrárny. *Envi Energy Czech* [online]. [cit. 2018-05-01]. Dostupné z: [http://www.envienergyczech.cz/diagnostika\\_fve.php#](http://www.envienergyczech.cz/diagnostika_fve.php#)
- [24] PV test – diagnostická laboratoř. *SOLARTEC* [online]. [cit. 2018-05-01]. Dostupné z: <http://www.solartec.eu/rozsahle-instalace/provozovani-fve-elektren/pv-test-diagnosticka-laborator>
- [25] Diagnostické metody fotovoltaických článků. *ASB-portal.cz: odborný portál pro profesionály v oblasti stavebnictví* [online]. 2010 [cit. 2018-05-06]. Dostupné z: <https://www.asb-portal.cz/tzb/fotovoltaika/diagnosticke-metody-fotovoltaickych-clanku>
- [26] Zkušenost soudního znalce z posuzování FVE v roce 2011. *ELEKTRO: odborný časopis pro elektrotechniku* [online]. 2012, **22**(5), 62-64 [cit. 2018-05-06]. ISSN 1210-0889. Dostupné z: [http://www.odbornecasopisy.cz//flipviewer/Elektro/2012/05/Elektro\\_05\\_2012\\_output/web/Elektro\\_05\\_2012\\_opf\\_files/WebSearch/page0001.html](http://www.odbornecasopisy.cz//flipviewer/Elektro/2012/05/Elektro_05_2012_output/web/Elektro_05_2012_opf_files/WebSearch/page0001.html)
- [27] Electroluminescence image of a representative mono-crystalline cell. *ResearchGate* [online]. [cit. 2018-05-03]. Dostupné z: [https://www.researchgate.net/figure/Electroluminescence-image-of-a-representative-mono-crystalline-cell-in-a-test-specimen\\_fig6\\_260054342](https://www.researchgate.net/figure/Electroluminescence-image-of-a-representative-mono-crystalline-cell-in-a-test-specimen_fig6_260054342)
- [28] Electroluminescence (EL) method. *Greateyes: The Berlin Company for Scientific Cameras and Inspection Systems* [online]. [cit. 2018-05-03]. Dostupné z: [https://www.greateyes.de/projects/greateyes/static/custom/image/EL\\_Bil\\_LS%20Pr of.jpg](https://www.greateyes.de/projects/greateyes/static/custom/image/EL_Bil_LS%20Pr of.jpg)



- [29] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Ilustrovala Viktorie VLACHOVÁ. Praha: BEN-technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [30] ROBERT H. KING. *Introduction to data acquisition with LabVIEW*. 2nd ed, internat. ed. New York, NY: McGraw-Hill, 2013. ISBN 9780071326353.
- [31] National Instruments. *What Is Data Acquisition?* [online]. 2011 [cit. 2018-03-26]. Dostupné z: <http://www.ni.com/data-acquisition/what-is/>
- [32] National Instruments. *NI PCI-6221/CB-68LP Academic Starter Kit* [online]. 2005 [cit. 2018-03-25]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/14504>
- [33] National Instruments. *NI Product Manuals: M Series User Manual* [online]. 2016 [cit. 2018-03-27]. Dostupné z: <http://www.ni.com/pdf/manuals/3710221.pdf>
- [34] National Instruments. *NI Product Manuals: NI 6221 Device Specifications* [online]. © 2015-2016 [cit. 2018-03-25]. Dostupné z: <http://www.ni.com/pdf/manuals/375303c.pdf>
- [35] National Instruments. *NI Product Manuals: NI CB-68LP/CB-68LPR User Guide* [online]. 2013 [cit. 2018-04-04]. Dostupné z: <http://www.ni.com/pdf/manuals/373853a.pdf>

## **Seznam příloh**

**Příloha A** – Front panel DataLoggeru v1.0

**Příloha B** – Blokový diagram DataLoggeru v1.0

CD s vytvořeným programem

### Front panel DataLoggeru v1.0

**Karta NI PCI-6221**    Data ze souboru    **DATA Logger**

**Kanál napětí**  
Karta-NI/ai0

**Kanál proud**    **Hodnota ext. rezistoru**  
Karta-NI/ai1    249 Ohm

**Typ zapojení**    **Interval záznamu**  
Differential    1 s

**Cesta k souboru pro uložení**  
C:\Users\Stambik\Desktop\Naměřená data.tdms

**Popis**

**Komentář k typu článku**

ON/OFF

**Napětí**    Proud    Výkon

Napětí (V)

17:50:53 17:50:55 17:50:57 17:50:59 17:51:01 17:51:03 17:51:05 17:51:07 17:51:09 17:51:11 17:51:13  
18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018 18.05.2018

START STOP Zastavit po: 0 min EXIT

**Blokový diagram DataLoggeru v1.0**

