

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA TECHNOLOGIÍ A MĚŘENÍ**

# **BAKALÁŘSKÁ PRÁCE**

**Návrh monitorovacího systému palivového článku PEM  
v LabVIEW**

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2016/2017

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan STANĚK**  
Osobní číslo: **E13B0222P**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Komerční elektrotechnika**  
Název tématu: **Návrh monitorovacího systému palivového článku PEM v LabVIEW**  
Zadávací katedra: **Katedra technologií a měření**

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

1. Zpracujte rešerši o využití systému LabVIEW ve spojení s DAQ pro měření s palivovými články. Do rešerše zahrňte i předchozí diplomové práce na toto téma.
2. Analyzujte systém palivového článku StaXX a navrhňte optimální počet měřicích míst a měřených veličin.
3. Pomocí karty DAQ NI USB vytvořte vhodné zapojení, otestujte jeho funkčnost v konkrétním PC a konkrétní verzi LabVIEW a Windows. Použijte EXPRES funkce.
4. Vytvořte funkční program pro snímání vybraných veličin pomocí DAQ do PC, jejich zpracování a vizualizaci. Prověřte optimální vzorkovací periody a nastavení.
5. Kompletní systém vybavte vhodným vizuálním rozhraním pro prezentaci naměřených dat v reálném čase a vytvořte souhrnnou dokumentaci.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **30 - 40 stran**

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

Seznam odborné literatury:

**Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.**


Vedoucí bakalářské práce: **Ing. Pavel Štekl, Ph.D.**  
Katedra teoretické elektrotechniky

Datum zadání bakalářské práce: **14. října 2016**

Termín odevzdání bakalářské práce: **8. června 2017**

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



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

V Plzni dne 14. října 2016

## **Abstrakt**

Předkládaná bakalářská práce je zaměřena na řešení návrhu a realizaci monitorovacího systému pro palivový článek PEM.

## **Klíčová slova**

LabVIEW, DAQ, palivový článek, StaXX

## **Abstract**

This bachelor thesis is focused on the design and implementation of the PEM fuel cell monitoring system.

## **Key words**

LabVIEW, DAQ, fuel cell, StaXX

## **Prohlášení**

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

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

.....

podpis

V Plzni dne 5.6.2017

Jan Staněk

## **Poděkování**

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Pavlu Šteklovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

# Obsah

<b>OBSAH</b> .....	<b>8</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>9</b>
<b>ÚVOD</b> .....	<b>10</b>
<b>1 PALIVOVÝ ČLÁNEK</b> .....	<b>11</b>
1.1 HISTORIE .....	11
1.2 VÝHODY A NEVÝHODY .....	12
1.3 PRINCIP .....	13
1.4 PALIVOVÝ ČLÁNEK STAXX .....	14
<b>2 LABVIEW</b> .....	<b>16</b>
2.1 ČELNÍ PANEL .....	17
2.1.1 Vstupy a výstupy .....	17
2.1.2 Nástrojová lišta čelního panelu .....	19
2.2 BLOKOVÝ DIAGRAM .....	20
2.2.1 Datové typy .....	21
2.2.2 Nástrojová lišta blokového diagramu .....	22
2.2.3 Datový tok .....	23
2.3 DAQ .....	23
2.4 MULTIFUNKČNÍ KARTA NI USB-6008/6009 .....	24
<b>3 ANALÝZA PŮVODNÍHO STAVU</b> .....	<b>26</b>
3.1 BLOKOVÝ DIAGRAM PŮVODNÍHO PROGRAMU .....	26
3.2 ČELNÍ PANEL PŮVODNÍHO PROGRAMU .....	28
3.3 ZHODNOCENÍ .....	28
<b>4 NÁVRH A REALIZACE MONITOROVACÍHO SYSTÉMU</b> .....	<b>29</b>
4.1 VÝVOJ .....	30
4.2 POPIS NOVÉHO PROGRAMU .....	31
4.2.1 Dodatky k programu .....	33
4.3 ČELNÍ PANEL NOVÉHO PROGRAMU .....	34
<b>ZÁVĚR</b> .....	<b>35</b>
<b>SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ</b> .....	<b>36</b>
<b>PŘÍLOHY</b> .....	<b>1</b>
PŘÍLOHA LABVIEW .....	1
PŘÍLOHA STAXX .....	2



## Seznam symbolů a zkratek

<i>PEM</i> .....	Polymer Exchange Membrane
<i>LabVIEW</i> .....	Laboratory Virtual Instruments Engineering Workbench (česky laboratorní pracoviště virtuálních přístrojů)
<i>DAQ</i> .....	Data acquisition (česky pořizování dat)
<i>ppm</i> .....	Parts per million (česky počet částic na jeden milion)
<i>NI</i> .....	National Instruments
<i>subVI</i> .....	podprogram
<i>VI</i> .....	virtuální přístroj
<i>AI</i> .....	analogový vstup

## Úvod

Předkládaná bakalářská práce je zaměřena na využití programovacího jazyka LabVIEW pro monitorování palivového článku PEM. Hlavním cílem je podstatná úprava a vylepšení již stávajícího monitorování s přidáním nových měřících stanic. Starý řídicí program je nutné převést na novou verzi software a s touto inovací je nutně jej celý přepracovat. Stará verze programu bude tedy sloužit pouze pro stanovení výchozího stavu a návrh designu, funkčnost je poplatná nové verzi i funkcím.

Text práce je rozdělen do čtyř částí; první se zabývá palivovými články, druhá se zabývá vlastnostmi programového prostředí LabVIEW. Třetí část obsahuje analýzu původního programu se stanovením priorit pro tvorbu nové verze. Poslední část popisuje návrh a realizaci monitorovacího systému pro palivový článek PEM.

První část se zabývá vlastnostmi palivového článku a jeho využitím. Druhá část se zabývá programovacím jazykem LabVIEW, kde se nachází základní vlastnosti a prvky programu. Součástí jsou i popisy prostředí programu, datové typy a funkce DAQ. Třetí část se zabývá analýzou předchozí verze programu a určováním dalších měřících míst. Čtvrtá část obsahuje návrh nového programu a jeho vizuální stránky. Je zde porovnán předchozí program se stávajícím. Porovnávána je funkčnost i vzhled.

# 1 Palivový článek

## 1.1 Historie

Již v 1. polovině 19. století Sir William Grove objevil při svých experimentech s elektrodami a elektrolyty princip palivového článku s vodíkovou a kyslíkovou částí. Následovalo jej ještě několik fyziků a chemiků se svými vlastními modely podobných elektrodových systémů, nicméně až do počátku 50. let 20. století byly snahy o využití a uplatnění těchto nových typů palivových článků málo úspěšné a mnohdy neefektivní. [1]

S prvními lety do vesmíru se začalo uvažovat o alternativních zdrojích. Bylo třeba napájet palubní elektroniku a klasické baterie byly těžké a nebylo snadné je nabít. Solární články tehdy neměly ještě tak propracovaný systém přeměny záření na elektrickou energii a tehdy velmi módní, ale velmi nebezpečná atomová energie byla v prostředí kosmických letů z mnoha důvodů nepoužitelná. [1]

První plně fungující palivový článek předvedl Francis T. Bacon roku 1959. První větší využití palivového článku s protonovými membránami uskutečnila společnost NASA pro vesmírný program Gemini. Jednalo se o alkalické palivové články, které do značné míry využívaly zdrojů pro let rakety, kdy palivem byl zpravidla vodík a oxidovadlem byl kyslík. Oba plyny si vesmírná mise vždycky nesla s sebou. A naopak produktem této katalytické reakce byla čistá voda, která také okamžitě našla ve vesmíru své místo. [1]

## 1.2 Výhody a nevýhody

Palivové články jsou většinou porovnávány s motory s vnitřním spalováním a elektrotechnickými akumulátory. Jejich přednosti oproti ostatním zdrojům jsou: [1]

- šetrné vůči životnímu prostředí,
- vyšší termodynamická účinnost než tepelné motory,
- dobré dynamické charakteristiky,
- tichý chod,
- nízké opotřebení a vysoká životnost,
- nevyžadují dobíjení oproti akumulátorům,
- nejsou přítomny pohyblivé části.

Palivové články mají také značné nevýhody a hlavní nevýhodou je absence efektivní infrastruktury pro dodávky vodíku. Mezi další nevýhody patří tyto: [1]

- náchylný na změnu teplot,
- vzácná a drahá platina,
- nízký výkon,
- náchylnost na čistotu paliva,
- nutnost odstraňovat zplodiny z chemických reakcí,
- nutnost vlhčení membrán i při skladování,
- složité řídicí systémy.

### 1.3 Princip

Princip palivového článku je velmi jednoduchý. Tento jednoduchý elektrochemický zdroj slouží k přeměně chemické energie paliva a oksyličovadla na energii elektrickou. Palivový článek má katodu a anodu, které odděluje kapalný nebo tuhý (polopropustná membrána) elektrolyt. Na anodu je přiváděno palivo nejčastěji ve formě vodíku a na katodu oksyličovadlu nejčastěji ve formě kyslíku nebo mnohdy postačuje i vzduch. [2]

Přivedený vodík se na anodě rozdělí na protony a elektrony. Protony mohou přes polopropustnou membránu projít ke katodě, kde reagují s přiváděným kyslíkem (vzduchem). Elektrony putují směrem ke katodě vnějším obvodem, kde vlivem pohybu elektronů získáváme elektrickou energii. V laboratorních podmínkách jsou palivové články schopny dosáhnout až účinnosti cca 60 %, zatímco reálná účinnost se pohybuje běžně mezi hodnotami 30-50 %. [1],[2]

Palivových článků je několik typů podle typu elektrolytu:

- Polymer Electrolyte Membrane Fuel Cell (PEMFC)
- Alkaline Fuel Cell (AFC) – alkalický elektrolyt
- Phosphoric Acid Fuel Cell (PAFC) – s kyselinou fosforečnou
- Molten Carbonate Fuel Cell (MCFC) – s tavenými uhličitany
- Solid Oxide Fuel Cell (SOFC) – s tuhými oxidy [2]

Také se články dělí dle teploty:

- Nízkoteplotní 60-130 °C
- Středněteplotní 160-220 °C
- Vysokoteplotní 600-1050 °C [3]

## 1.4 Palivový článek StaXX

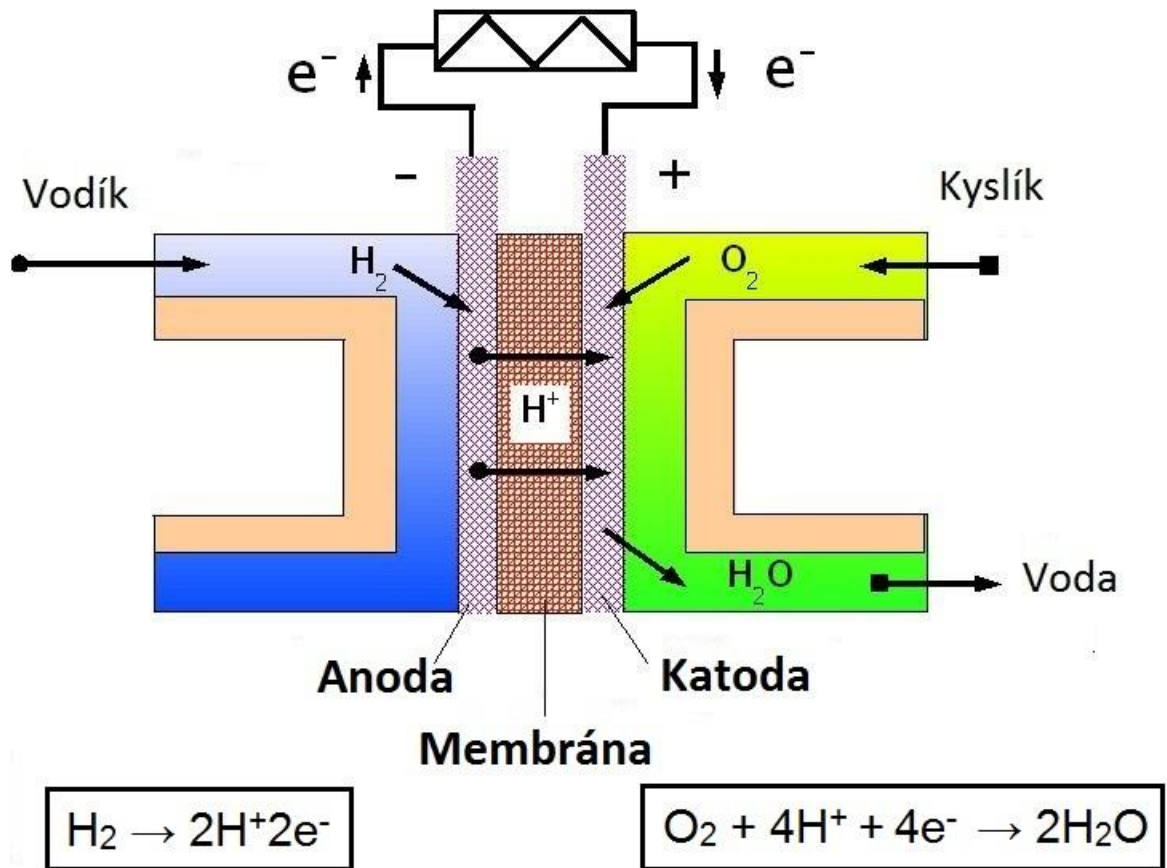
StaXX je druh vodíkového palivového článku typu PEMFC (Proton Exchange Membrane Fuel Cell). Jak je již z názvu znatelné, jedná se o membránový druh palivového článku s protonovou vodivostí. Článek vytvořila německá firma H-TEC. [2]

Tenká membrána v článku slouží jako tuhý kationový elektrolyt, tudíž propouští kladně nabitě ionty. Elektrody se zde nalézají blízko slabé vrstvy uhlíku obsahující malé množství platiny, která zde plní funkci katalyzátoru a usnadňuje rozklad plynného vodíku. Celý princip článku je založen na oxidaci paliva, kterým je plynný vodík. [2]

Na anodě a katodě dochází k dvěma odlišným elektrochemickým reakcím. Na anodu je přiváděn vodík a dochází zde ke štěpení atomu vodíku na proton a elektron. Protony projdou skrz membránu, zatímco elektrony membránou neprojdou a procházejí externím okruhem. Vodíkové protony, elektrony a přiváděný kyslík se slučují na katodě a tím vzniká voda. Přívod kyslíku je většinou ve formě vzduchu, ale je zde možnost i přívodu čistého kyslíku. [2]

U článku je důležité dostatečně hydratovat membránu, pokud by se tak nedělo, článek by ztrácel na funkčnosti. Operační teplota je limitována použitým polymerem. Pracovní teplota se udává do 120 °C, ale záleží na druhu použitého polymeru. Velmi důležité je také to, aby odpařovaná voda nepřevyšovala vodu, která se při procesu vyrobí. [2]

Článek je velmi náchylný na čistotu použitého vodíku. Vodík používaný pro tento článek se proto získává elektrolýzou destilované vody. Pro platinu (sloužící jako katalyzátor) je nežádoucí síra, její sloučeniny a olovo. Koncentrace těchto nečistot v palivu nesmí být vyšší než 5 ppm. Tento fakt do značné míry vylučuje například vodík připravený klasickou cestou ze zemního plynu reformováním.[2]



Obr. 1.1 Zjednodušený model článku typu PEMFC [7]

Elektrody jsou na tomto článku stálé a neopotřebovávají se jako u běžných alkalických nebo zinkových monočlánků. Ubývá pouze okysličovadlo a palivo. Elektrody musí mít dobré mechanické vlastnosti, mít dobrou vodivost, chemickou stabilitu a musí mít dobré katalytické vlastnosti. Pokud je v článku přítomný jak vodík (palivo), tak kyslík (okysličovadlo), pak probíhá neustálá výroba stejnosměrného elektrického proudu. Článek je velmi flexibilní a je možné ho použít i jako zdroj pro mobilní zařízení. Články se dají také spojovat a vytváří se z nich palivové celky. Spojování se provádí skládáním jednotlivých článků za sebe. [2]

Pospojováním jednotlivých článků do systému sendviče vznikne takzvaný stack, což je průmyslová forma palivového článku. Výkon článku je definován jako celková plocha elektrody, v tomto případě tedy plocha základní polymerní membrány. Stohováním jednotlivých článků za sebe do struktury sendviče tak vzniká článek s velkou plochou článku, aniž by podstatně narůstaly jeho geometrické rozměry. Takový článek pak může mít i relativně značný výkon udávaný v kW. [2]

## 2 LabVIEW

Americká společnost National Instruments (NI) v 80. letech přišla s novým programovacím jazykem LabVIEW (Laboratory Virtual Instruments Engineering Workbench čili „laboratorní pracoviště virtuálních přístrojů“). Společnost National Instruments je průkopníkem v odvětví virtuální instrumentace, která v novém tisíciletí zažívá obrovský rozvoj v oblasti vědy, výzkumu, školství a průmyslu. [4]

Tento program, a dalo by se spíše říci systém, není již klasický textový programovací jazyk, ale patří do skupiny takzvaných „grafických jazyků“. Tyto nové výrazové prostředky jsou založeny na myšlence, že člověk, který je schopen vyjádřit své myšlenky do blokového diagramu, umí totéž definovat s pomocí grafického vyjádření. Základem programu je také možnost sestavit si vlastní virtuální přístroje a upravovat si je dle potřeb, aniž by se musely vyrábět ze skutečných součástek. Samozřejmě k většině virtuálních přístrojů jsou třeba nějaké multifunkční karty, které jsou potřebné pro komunikaci mezi počítačem a měřeným objektem. Na výrobu těchto HW zařízení se společnost NI také specializuje. [4]

Program komunikuje s uživatelem a tvůrcem pomocí dvou základních oken. Tato okna se nazývají čelní panel (*Front panel*) a blokový diagram (*Block diagram*). Čelní panel tvoří vizuální a uživatelské rozhraní, kde může uživatel ovládat program pomocí ovládacích prvků a zobrazují se zde výstupní data. Blokový diagram umožňuje propojování vstupů a výstupů programátorem a tvorbu vlastního algoritmického řešení včetně třeba přenosu signálu, definice typů signálů, časování a mnoho dalších funkcí. Do této části běžný uživatel zpravidla nezasahuje, ale vstupuje sem pouze programátor. Ideou společnosti NI je ovšem to, aby uživatel i programátor byl jednou osobou. [4]



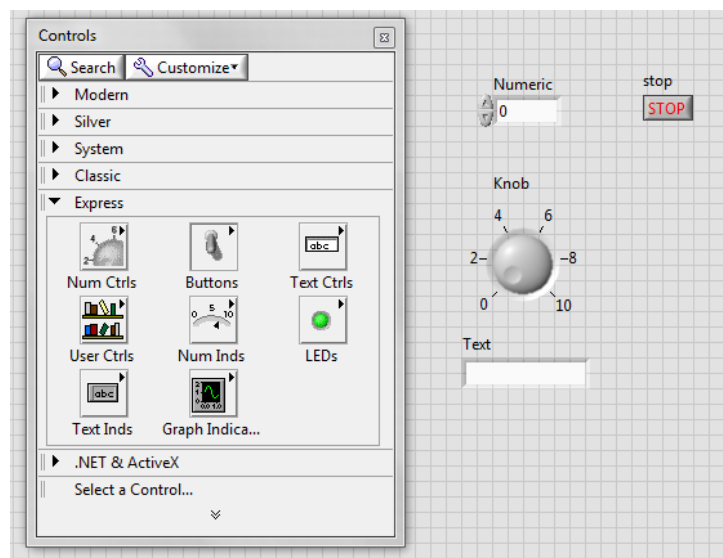
## 2.1 Čelní panel

Čelní panel tvoří uživatelsky orientované rozhraní aplikace LabVIEW. Je tvořen grafickými prvky, které slouží pro ovládání a vizualizaci průběhu aplikace. Jsou zde projekce vstupních a výstupních zařízení v grafické podobě, nastavují se zde důležité parametry pro funkčnost programu a jeho ovládání. Na čelním panelu se vytváří vizualizační vrstva aplikace tak, jak ji uvidí koncový uživatel. Ovládací prvky (**Controls**) si můžeme představit jako vstupní aktivní zařízení, stejně jako by tomu bylo na reálném přístroji (tlačítka, přepínače, atd.). Zobrazovací prvky (**Indicators**) ukazují výstupní data a mají tvary displejů a ručkových zařízení. [4]

Ovládacím i zobrazovacím prvkům se může určit přesná poloha, nadefinovat název, jejich vzhled a chování. V programu lze vytvářet vlastní uživatelsky orientované prvky zvané SubVI nebo používat předdefinované prvky z mnoha knihoven. [4],[5]

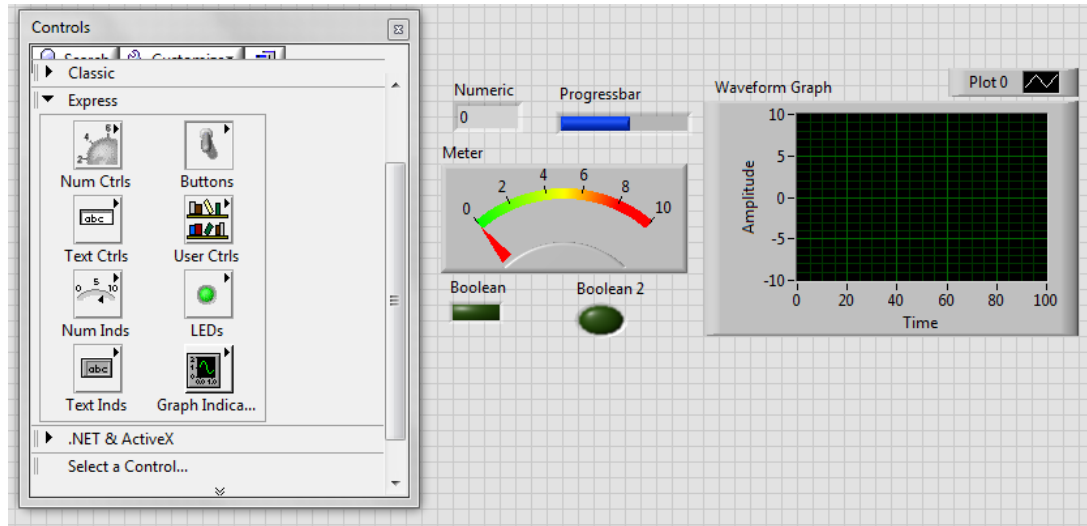
### 2.1.1 Vstupy a výstupy

Vizuální prvky se rozdělují z funkčního hlediska i principu na dvě základní skupiny, a to jsou vstupy a výstupy (input a output). Mezi vstupní ovladače (**Controls**) patří například tlačítko (**Button**), otočný knoflík (**Knob**), textový nebo číselný ovladač (Viz obr. 1.1). [4]



Obr. 2.1 Ukázka čelního panelu s několika vstupy

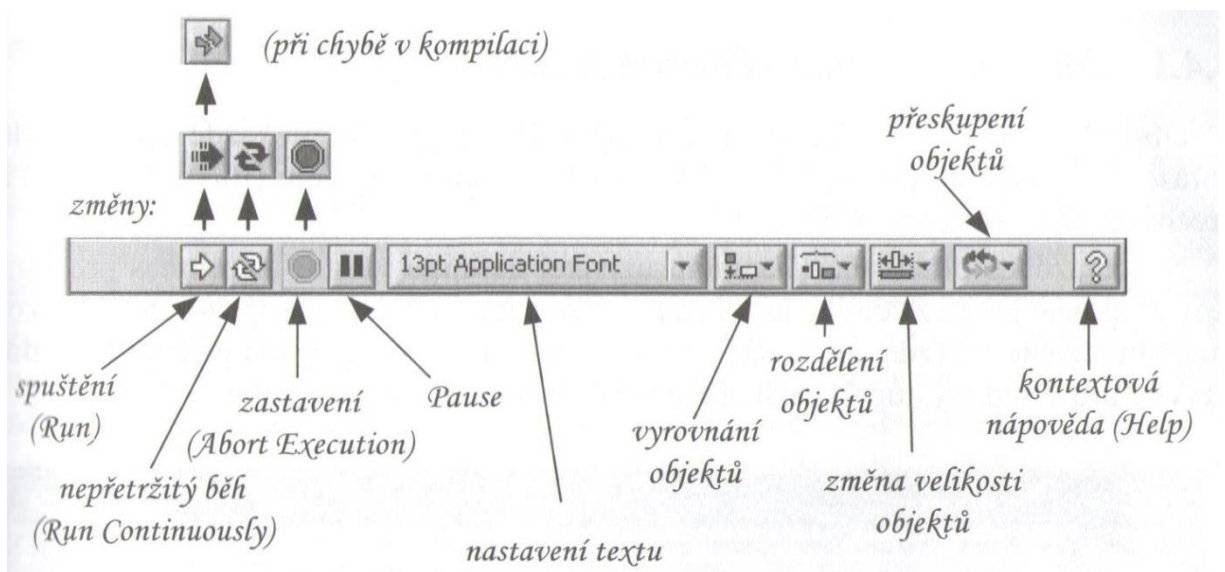
Mezi výstupní zobrazovače (*Indicators*) patří například grafický model ručičkového měřidla (který připomíná skutečný analogový měřicí přístroj), textový nebo číselný zobrazovač, kontrolky (*LED*), proužkové diagramy (*Progressbar*) a různé typy grafů (Viz obr. 1.2). [4]



Obr. 2.2 Ukázka čelního panelu s několika výstupy

## 2.1.2 Nástrojová lišta čelního panelu

Na nástrojové liště čelního panelu se nacházejí prvky pro ovládání a indikaci stavu programu, jeho stavů nebo jeho částí. Všechny prvky jsou využity jako tlačítka nebo zobrazovače stavu programu. Tlačítkem **Run** (šipka doprava) lze spustit program a tím proběhne jeden cyklus programu. Toto tlačítko detekuje i případnou chybu při kompilaci programu tím, že se vykreslí v režimu „přerušené šipky“, program nelze spustit a vyskočí **Error list** s dokumentací chyby a návodem na její odstranění. Pokud chce uživatel opakovaný běh programu, použije druhé tlačítko v pořadí zleva (**Run Countiuously**). Tento režim se také používá pro odladování programů před definicí cyklů. Pro celkové zastavení spuštěného programu používáme **Abort Execution** (třetí tlačítko zleva) a pro časově omezené pozastavení běhu programu využíváme **Pause**. Zbylé funkce lišty slouží k úpravě objektů na ploše a jejich popisků. Jsou určeny pro dokončení vizuálního stylu úpravy aplikačního listu s tím, že je zde možno prvky vycentrovat, srovnat pod sebe nebo k okraji a definovat mezery mezi prvky obecně. [4]



Obr. 2.3 Přehled nástrojové lišty čelního panelu [4]

## 2.2 Blokový diagram

V okně s názvem blokový diagram se sestavuje vlastní grafický algoritmus programu, jinak řečeno propojení prvků z čelního panelu a jejich parametry. V blokovém diagramu můžeme libovolně propojovat vstupy a výstupy, které mají vlastní ikonku na čelním panelu. [4]



Obr. 2.4 Ukázka propojení vstupu a výstupu pomocí blokového diagramu

Blokový diagram obsahuje terminály, subVI, funkce, konstanty, struktury a spojení, které přesouvají data mezi jednotlivými prvky blokového diagramu. Prvky čelního panelu jsou automaticky zobrazeny jako vstupy nebo výstupy v blokovém diagramu. Vstupy a výstupy jsou obdobou parametrů a konstant v textových programovacích jazycích. [5]

Každý prvek blokového diagramu má podle své funkce nastaveny vstupní či výstupní připojovací body (terminály). Tyto body lze propojit datovým spojem s dalšími prvky pomocí propojovacího nástroje (**Wiring Tool**). Nezapojený bod má barvu bílou, již zapojený bod je označen šedě, černě je označen bod korespondující s vybraným prvkem. Prvky propojujeme samozřejmě podle logického směru toku informace, tedy nejčastěji zleva doprava a dbáme, aby se jednotlivé části nacházely v místě, kde mají potom být definovány podmínky, cykly nebo popřípadě jiné algoritmické struktury. Tyto programové struktury samozřejmě svůj obraz na čelním panelu nemají, jsou skryté a prezentují se běžnému uživateli pouze jako výkonné funkce, s jejich pomocí ovládá vykonávání programu. [4]

### 2.2.1 Datové typy

Vstupy a výstupy objektů jsou vzájemně propojeny spoji různých datových typů. Objekty různých vzájemně neslučitelných datových typů nelze vzájemně propojovat. Například LED dioda může mít za vstup pouze datový typ boolean (binární datový typ). Otočný prvek (**Knob**) může reprezentovat celočíselný integer nebo číselný s pohyblivou čárkou (floating point). V LabVIEW se také dají redefinovat a propojovat prvky integer s floating point. [4]

Problematika propojování objektů s ohledem na datové typy je mnohem složitější, systém LabVIEW ale do určité míry uživatele vede a mnoho problémů, například s datovými konverzemi, dokáže při návrhu systém úspěšně zvládnout sám pomocí přizpůsobení jednotlivých bloků jeden druhému. Samozřejmě pouze do té míry, že i systém LabVIEW respektuje datové typy z hlediska paměťové náročnosti a vnitřní datové konverze beze ztráty přesnosti. [4]

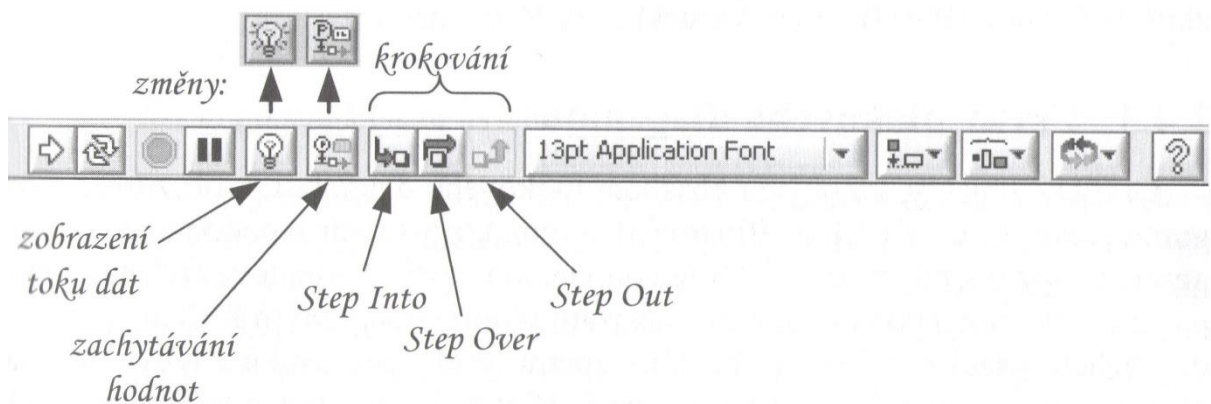
Tab. 2.1 Základní datové typy v LabVIEW [4]

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

## 2.2.2 Nástrojová lišta blokového diagramu

Většina tlačítek nástrojové lišty blokového diagramu je shodná s tlačítky nástrojové lišty čelního panelu (**Run**, **Run Continuously**, **Pause**, **Abort Execution** a tlačítka pro úpravu objektů a popisek). Nová tlačítka slouží k ladění programu. Při odladování programu můžeme použít možnosti krokování. Dále je zde funkce **Breakpoint**, která slouží k zachytávání dat na určitém spoji či uzlu a k zastavení programu. Posledním novým symbolem je **Highlight Execution** (symbol žárovky). Tato funkce nám pomáhá zobrazit datový tok při běhu programu v blokovém diagramu. Pokud je žárovka rozsvícena (zapnuta), celý program se zpomalí vinou náročné animace a na všech vláknech programu lze sledovat tok dat. Tento režim je velmi vhodný pro případy, kdy sledujeme například biorizaci vykonávání instrukcí a potřebujeme přesně predikovat posloupnost vykonávaných sekvencí a výsledný datový typ. [4]

Stejně jako v klasických programovacích jazycích i zde máme možnost vkládat do funkčních bloků takzvané sondy (probes), které vždy otevírají malé okénko s přehlednými informacemi o hodnotě na tom kterém místě označeném touto sondou. Tento režim je zdatným pomocníkem pro odladování a testování jednotlivých bloků a funkcí programu. [4]



Obr. 2.5 Ukázka propojení vstupu a výstupu pomocí blokového diagramu [4]

### 2.2.3 Datový tok

V LabVIEW se program vykonává datovým tokem, nikoliv lineárním vykonáváním řádků kódů, ale může být i lineárním, jak je běžné u textových programovacích jazyků. Datový tok určuje směr provádění programu. V blokovém diagramu se provede zpracování informace na jednotlivém subVI tehdy, pokud jsou na všech jeho vstupech přivedena data. Pokud ano, proces se provede a subVI pošle dále zpracovaná data. V případě, že se vyskytne situace, kdy nějaký uzel tvořený například součtovou funkcí nebude mít na obou vstupech zároveň platná data, tak program čeká do okamžiku, než tato data dorazí a teprve poté se součet vykoná a data z tohoto uzlu se posouvají dále. [4]

## 2.3 DAQ

Práce s daty získanými při měření a generování fyzikálních signálů systémem LabVIEW ve spojení s HW bloky National Instruments se nazývá pořizování dat neboli Data AcQuisition (DAQ). DAQ systémy určené pro pořizování dat umožnily zvýšení rychlosti měření a zpracování dat. Systémy pro získávání dat obvykle převádějí analogové průběhy na digitální hodnoty pro zpracování. [4],[6]

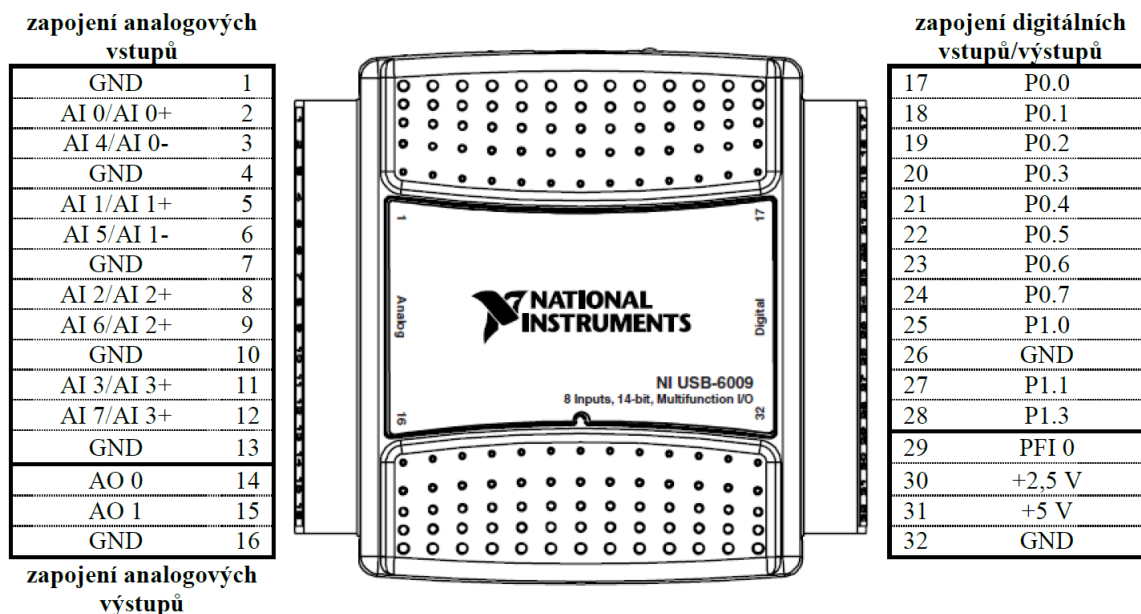
Pro získávání dat jsou obvykle použity softwarové programy vytvořené pomocí různých obecně používaných programovacích jazyků, jako je Assembly, BASIC, C, C ++, C #, Fortran, Java, LabVIEW, Lisp, Pascal atd. [6]

K dispozici jsou také softwarové balíky s otevřeným zdrojovým kódem, které poskytují veškeré potřebné nástroje pro získávání dat z různých hardwarových zařízení. Tyto nástroje pocházejí z vědecké komunity, kde komplexní experiment vyžaduje rychlý, flexibilní a přizpůsobitelný software. Tyto balíčky jsou obvykle přizpůsobeny pro získávání dat, lze je snadno upravit a používají se v mnoha fyzikálních experimentech po celém světě. [6]

## 2.4 Multifunkční karta NI USB-6008/6009

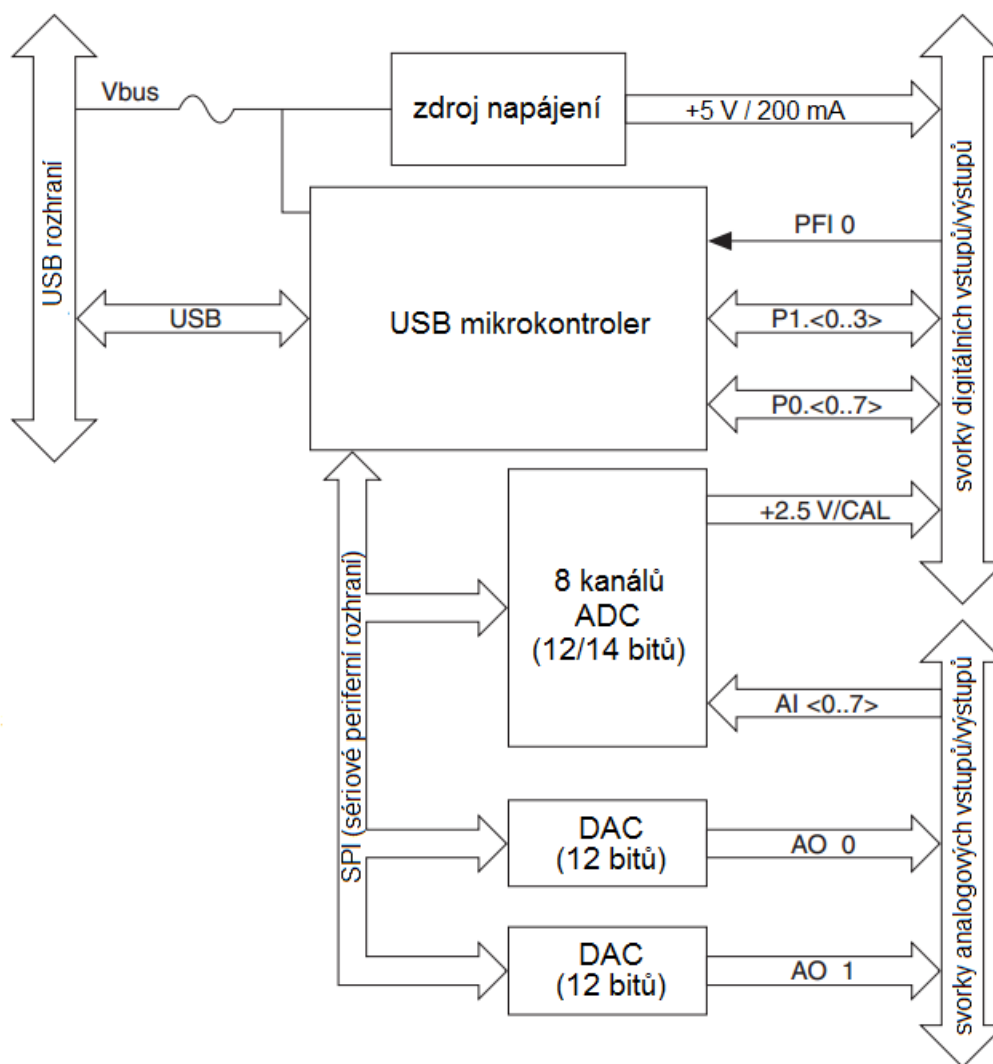
Multifunkční karta je profesionální zařízení DAQ od firmy NI pro využití v prostředí LabVIEW, která se připojuje k počítači pomocí USB. Systém pro sběr dat (DAQ) se používá k přenosu upraveného elektrického signálu do počítače za účelem softwarové analýzy a zaznamenávání dat. [4]

Blokové schéma multifunkční karty NI USB-6008/6009 je znázorněno na *obr. 1.6*. Karta obsahuje osm analogových vstupů (AI 0 až AI 7), dva analogové výstupy (AO), dvanáct digitálních vstupních/výstupních kanálů (P0.0 až P1.3) a jedním 32bitovým čítačem. Analogové vstupy AI můžeme využít jako osm nesymetrických kanálů, nebo jako čtyři diferenciální kanály. [4]



Obr. 2.6 Rozmístění signálů na konektorech multifunkční karty NI USB-6008/6009 [4]





Obr. 2.7 Blokové schéma multifunkční karty NI USB-6008/6009 [4]

S kartou NI USB-6008/6009 je dodáván kabel USB (typu A-B) o délce 1 m, dvě zásuvné svorkovnice se sadou štítků, návod v anglickém jazyce a dvě CD s programovým vybavením NI-DAQmx. [4]

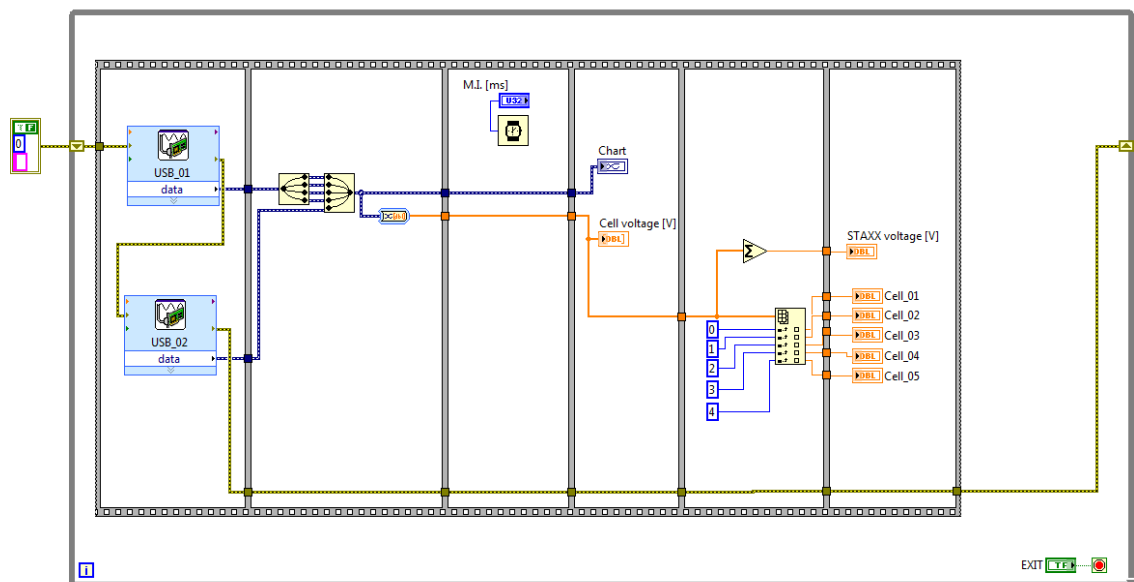
Na vstup AI můžeme přivést analogový signál v diferenciálním zapojení (kladný pól na vstup AI+ a záporný na vstup AI-) nebo v nesymetrickém zapojení (kladný pól na vstup AI a záporný na zemnicí svorku GND). U diferenciálního zapojení lze na vstupy AI přivést napětí v rozsahu maximálně  $\pm 20$  V a v nesymetrickém zapojení maximálně  $\pm 10$  V. Větší úroveň napětí je omezena. [4]

## 3 Analýza původního stavu

### 3.1 Blokový diagram původního programu

První stará, nicméně stále funkční verze programu, byla sestavena z komponentů systému LabVIEW verze 8.5 v roce 2007 a byla určena pro tehdy nový operační systém Windows 7. Tato verze byla velmi jednoduchá, ale rámcově splňovala požadavky obsluhy. Při diskuzi o tvorbě nového systému bylo rozhodnuto, že nový systém bude vycházet z této dosavadní verze, ale bude samozřejmě naprogramován v novém systému LabVIEW verze 2013, která je kompatibilní s Windows včetně aktuální verze 10. Starý systém bude v souvislosti s instalací nového operačního systému nepoužitelný, protože s ním není kompatibilní a nebude již dále podporován. Otázka tvorby nového programu byla tedy do značné míry podmíněna i přechodem na nový operační systém.

Nepřetržitý běh programu zajišťuje smyčka **While**. Smyčka se neustále opakuje a čeká na stisk tlačítka EXIT, které ukončí smyčku a tím i samotný program. Pro čtení a zpracování dat se využívá struktura **Flat sequence**, která umožňuje postupné čtení a zpracování okno po okně.



Obr. 3.1 Blokové schéma původního programu

V prvním okně vlastního sekvencéru se načtou data z multifunkčních karet USB 6008. Používají se zde dvě karty, jelikož na jednu kartu se vejdou pouze 4 měřící kanály a zde je třeba pěti měřících míst. Na kartách jsou využity analogové vstupy v diferenciálním zapojení. Na vstup je tedy možné přivést napětí  $\pm 20$  V. V původním návrhu nebylo počítáno s jinými typy

signálů ani pro snímání ani pro monitorování.

V druhém okně sekvencéru dochází ke sloučení jednotlivých datových toků z použitých kanálů. Tím vznikne jeden vícekanálový datový tok, se kterým se pak lépe pracuje jako s prvky jednorozměrného pole. První datový tok obsahuje hodnoty palivových článků 1-4. Druhý datový tok obsahuje měření pouze 5. palivového článku. Sloučením vznikne řada od jedné do pěti, což usnadňuje další zpracování dat. Po sloučení zde dochází k převodu datového toku na numerický řetězec (1D pole).

V třetím okně je využita funkce **Wait**, která celý proces pozastaví na stanovený čas v milisekundách. Čas si zde může uživatel libovolně volit. Tím se definuje vzorkovací frekvence, původně to bylo nutné z hlediska méně výkonného HW pro plynulost vykreslování a sběr dat. V základním nastavení je 100 ms, což znamená, že v ideálním případě získáváme každou sekundu 10 hodnot z každého měřicího kanálu.

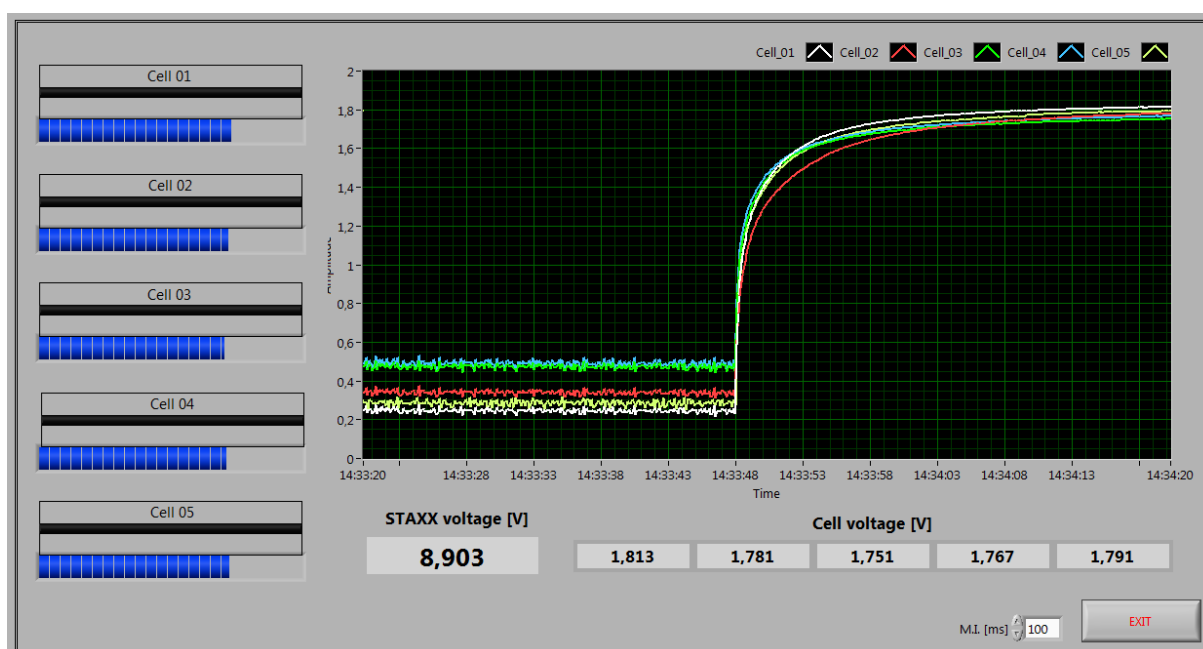
Čtvrté okno zajišťuje zápis dat do grafu a zobrazení dat v průběžném grafu (**Chart**) na čelním panelu. Díky sloučení obou datových toků karet v jeden a převedením na numerický řetězec se zapíše všechna měřicí místa najednou do Cell voltage (**Array** je 1D pole a obsahuje 5 numerických indikátorů), které slouží jako zobrazovače velikosti napětí.

V pátém okně **Flat sequence** dochází k sumaci všech napětí na člancích pomocí další struktury **Add array elements**. Je zde též rozloženo 1D pole na jednotlivé kanály pomocí **Index array**.

V posledním okně se provádí zápis součtů napětí, které byly provedeny v přechodím okně, a zobrazení jednotlivých velikostí napětí do grafu typu bar (**Horizontal graduated bar**), které jsou nastaveny pro napětí od 0 V do 2 V, jelikož se napětí na palivových člancích pohybuje v tomto rozsahu.

### 3.2 Čelní panel původního programu

Při pohledu na původní čelní panel jasně vyniká grafický zobrazovač (*Waveform Chart*). Grafický zobrazovač je nastaven na rozsah 0-2 V a časová základna činí 60 s. Pod grafem se nachází indikátor celkového napětí (StaXX voltage) a 1D pole zobrazující napětí jednotlivých palivových článků (zleva od prvního až po pátý). Úplně dole se nachází volitel časového zpoždění (*Wait*) a tlačítko EXIT, které program ukončuje. Nalevo od grafu se nachází menší grafy typu horizontální bar, které poměrově ukazují napětí jednotlivých článků.



Obr. 3.2 Čelní panel původního programu

### 3.3 Zhodnocení

Původní monitoring obsahuje 5 měřících míst, která snímají napětí na jednotlivých článcích sestavy. Pro optimalizaci monitorování by bylo dobré měřit vstupní napětí na celém komplexu elektrolyzéry. Vstupní napětí elektrolyzéry je důležité vzhledem k produkci vodíku pro všechny sériově zapojené články. Dalším měřícím místem by mohl být vstupní proud ze StaXX. Proud by bylo možné počítat z výstupního napětí, jelikož zátěž sestavy známe.

Vzhled programu by mohl být upraven paletou SILVER místo stávající CLASSIC pro lepší vizuální dojem a přehlednost. Objekty pro jednotlivé články (hodnota, barva, pořadí a bar) by bylo dobré upravit tak, aby vše bylo pospolu a přehledně uspořádané, tedy aby bylo na první pohled jasné, co k čemu patří. Program by byl doplněn o nově měřené hodnoty.

## 4 Návrh a realizace monitorovacího systému

Návrh programu pro monitoring palivového článku StaXX vychází z blokového diagramu původního programu s přidáním měření napájecího napětí elektrolyzéry a výpočtem výstupního proudu ze StaXX. Výpočet proudu je realizován z výstupního napětí StaXX a zátěže.

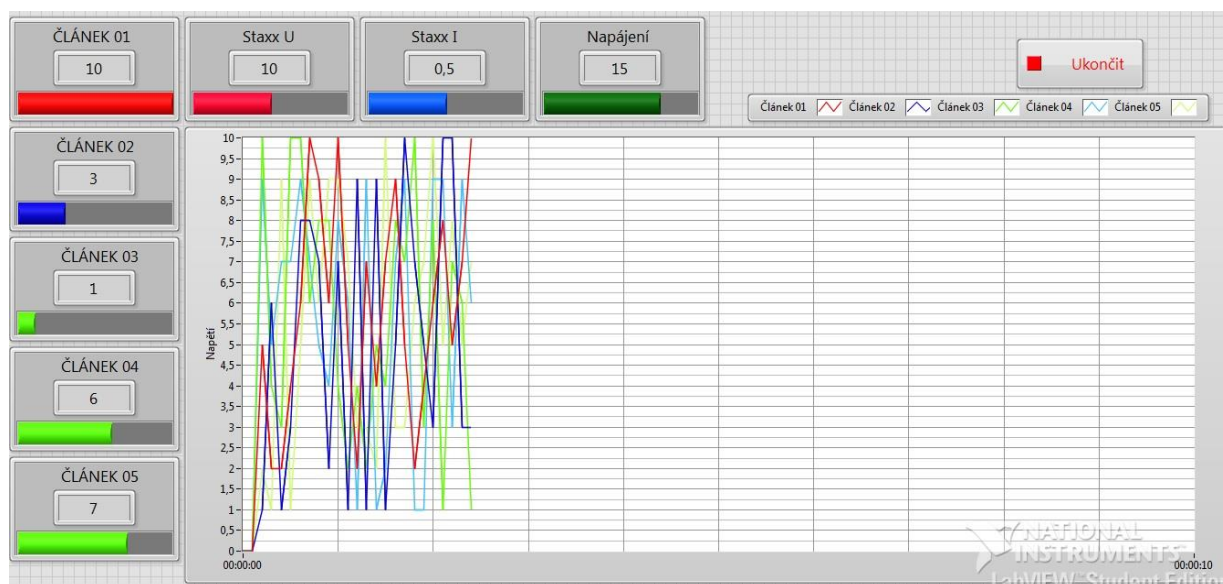
Na čelním panelu došlo k razantním změnám vzhledu s cílem dosáhnout větší přehlednosti. Indikátory velikosti napětí a bary jsou pro přehlednost a snazší čtení umístěny, co nejbližší k sobě a seřazeny podle pořadí tak, aby byla zajištěná dobrá přehlednost i pro letmý pohled obsluhy.

Hlavní vizuální rozdíl na čelním panelu by měl být ve využití komplexní návrhové palety SILVER místo původní palety CLASSIC. Temné pozadí původního provedení, které evokovalo obrazovku osciloskopu, by mělo nahradit v nové verzi bílé pozadí, evokující grafický záznam analogového stylu s různobarevnými zapisovači. Navíc je tato změna přijatelná i z důvodu orientace displeje řídicího NBT vzhledem k oknům laboratoře a jejich jasů.

Zde je třeba poukázat na fakt, že v době vzniku původního programu ještě nebyl systém LabVIEW vybaven všemi návrhovými paletami, které kromě jiných barevných schémat disponují i jinými předdefinovanými schématy měřicích přístrojů a ovládacích prvků.

## 4.1 Vývoj

Prvním krokem při vývoji programu bylo vyřešení jeho vizuálního rozhraní. Díky této změně mělo být dosaženo lepší přehlednosti a přesnějšího určení barů a numerických ukazatelů velikosti napětí pro jednotlivé palivové články, aby bylo jasné vidět, který bar a indikátor patří ke kterému článku (na rozdíl od původní verze programu). Jeden z prvních prototypů je ukázán na *obr. 4.1*. Na tomto prototypu je již vidět patrná změna, ale stále zde nebyla ideální přehlednost hlavně pro rychlé porovnání pomocí barů, které byly příliš daleko od sebe a ve výsledku byly „zbytečné“, jelikož se pro účely porovnávání příliš nehodily. Později došlo ke změně, jejímž výsledkem je konečná verze programu, kde jsou všechny bary zobrazeny přehledně – hned pod sebou.



Obr. 4.1 Prototyp čelního panelu pro nový program

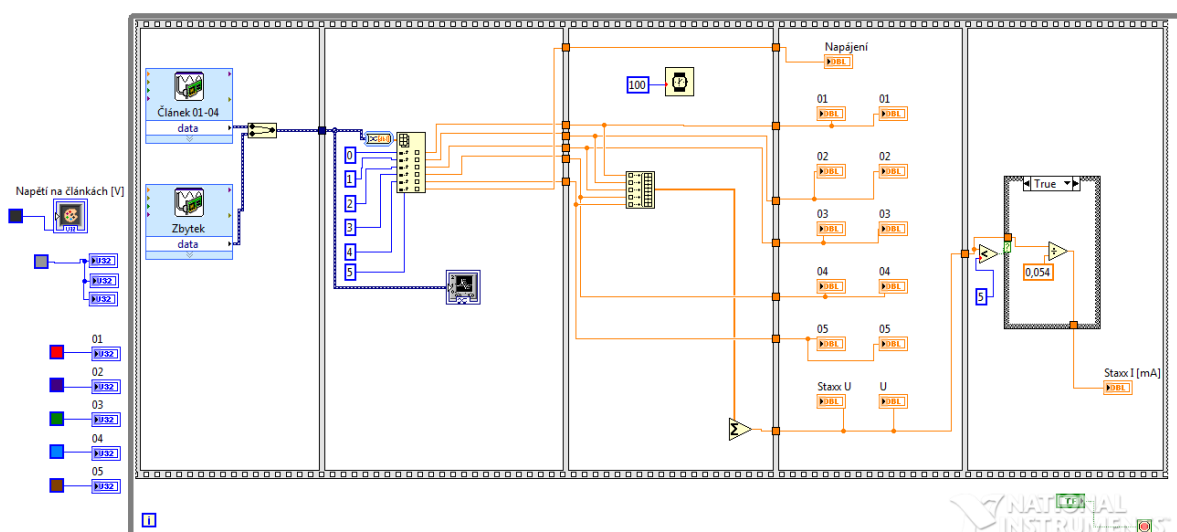
Všechny návrhy byly průběžně konzultovány s vedoucím práce a design čelního panelu i funkčnost algoritmu byla neustále upřesňována. Hlavní zásadou bylo porovnávání obou verzí z hlediska přehlednosti a rozmístění indikačních a ovládacích prvků, aby se změnou nepřišel i pokles efektivity využití v souvislosti s hledáním a nalézáním stejných funkcí, ovšem na jiných místech jako je tomu například u kancelářských systémů.

Pro prvotní testování byl vytvořen generátor čísel, díky čemuž nebylo třeba využívat DAQ Assistanty a reálný systém sběru dat. Po dokončení návrhu čelního panelu bylo třeba vytvořit algoritmus programu. Nejprve se začínalo měřením jednotlivých palivových článků, poté se přidala sumace výstupního napětí StaXX. Následně byl program vyzkoušen pomocí

DAQ Assistantů na palivovém článku, kde program uspěl i přes prvotní menší komplikace, jež způsobilo nevhodné nastavení multifunkčních karet. Po úspěšném otestování se mohl vývoj posunout dále – k návrhu výpočtu výstupního proudu ze StaXX (postup je popsán v kapitole 4.2.1). Na závěr bylo přidáno měření vstupního napětí pro elektrolyzér, kde bylo použito diferenciální zapojení analogového vstupu, které umožňuje přivést měřené napětí v rozsahu  $\pm 20$  V, což je pro měření plně dostačující.

## 4.2 Popis nového programu

V této části je rozebrán krok po kroku celý blokový diagram programu. Tento algoritmus využívá smyčku **While**, která zajišťuje nepřetržitý běh programu. Smyčka **While** je ukončena stiskem tlačítka Ukončit. Při zapnutí programu se zároveň spustí i barevné konstanty pro struktury **Color box**. I pro tento případ je použita struktura **Flat sequence**, která je ideální pro čtení a zpracování dat. Modul struktury **Flat sequence** umožňuje vykonání programu v pořadí zleva doprava a zápis je přehledný.



Obr. 4.2 Blokové schéma nového programu

V prvním okně **Flat sequence** se načítají data z multifunkčního modulu USB 6008. Pro snímání dat se používají dvě multifunkční karty. Na první multifunkční kartě (Článek 01-04) se načítají data z 1. - 4. palivového článku. Na druhé kartě (Zbytek) jsou odebírána data z 5. palivového článku a nově i vstupní napájecí napětí pro elektrolyzér. Všechny měřící kanály využívají diferenciálního zapojení. Za DAQ Assistanty jsou datové toky spojeny v jeden.

V druhém okně je tento spojený tok využit pro zápis dat přímo do struktury grafu **Chart**. Sloučený tok je zde automaticky LabVIEW konvertován z dynamického datového typu (ten v původním systému programu nebyl vůbec takto implementován) na 1D numerické pole, které následně vstupuje do modulu **Index array**, kde je poté rozloženo na jednotlivé elementy pole. Tento rozkol umožní v budoucnu přidávat případně další kanály na snímání nebo provádění dalších funkcí s jednotlivými elementy pole, které se takto pro programátora jeví jako vektor.

V třetím okně jsou, již zmíněné, elementy opět složeny pomocí modulu funkce **Build array**. Zde jsou využity pouze hodnoty od palivových článků pro sestavení 1D pole obsahujícího všech 5 prvků. Vytvořené 1D pole následně sečte sumační (součtový) člen, čímž vytvoří celkové napětí článku StaXX. Udává se zde i vzorkovací frekvence programu, která má hodnotu 100 ms. V původním programu bylo možno časové zpoždění uživatelsky nastavovat. Při experimentálních testech jiných hodnot zpoždění se došlo k závěru, že 100 ms je ideální hodnota – proto je v této nové verzi konstantou. Nový systém již počítá i s výkonovou rezervou připravovaného nového počítače, který je pro potřeby nového systému připraven.

Čtvrté okno obsahuje indikátor pro napájení elektrolyzéry, zobrazovače a grafy typu bar pro palivové články a celkové napětí StaXX. Opět jsou využity některé funkce z nové palety grafických návrhů, které v původním systému nebyly.

Poslední (páté) okno struktury **Flat sequence** obsahuje strukturu **Case**, která určuje, kdy se má provádět výpočet výstupního proudu StaXX a samotný výpočet pak následně provede. Před vlastním blokem **Case** je umístěno komparační subVI (**Less?**), do kterého přichází informace o velikosti napětí. Tento prvek porovnává přiváděné celkové napětí a rozhoduje, zda je napětí vyšší či menší než 5 V. Pokud je napětí větší než 5 V, prvek vyšle v binárním kódu nulu a struktura **Case** provede program pro nastavení **False** a výsledný proud bude nulový a je zapsán o indikátoru StaXX I. Pokud je napětí menší než 5 V, bude vykonáno nastavení pro **True**. To má za následek vypočítání výstupního proudu, který se pohybuje řádově v mA, a následnému zapsání do ukazatele StaXX I.

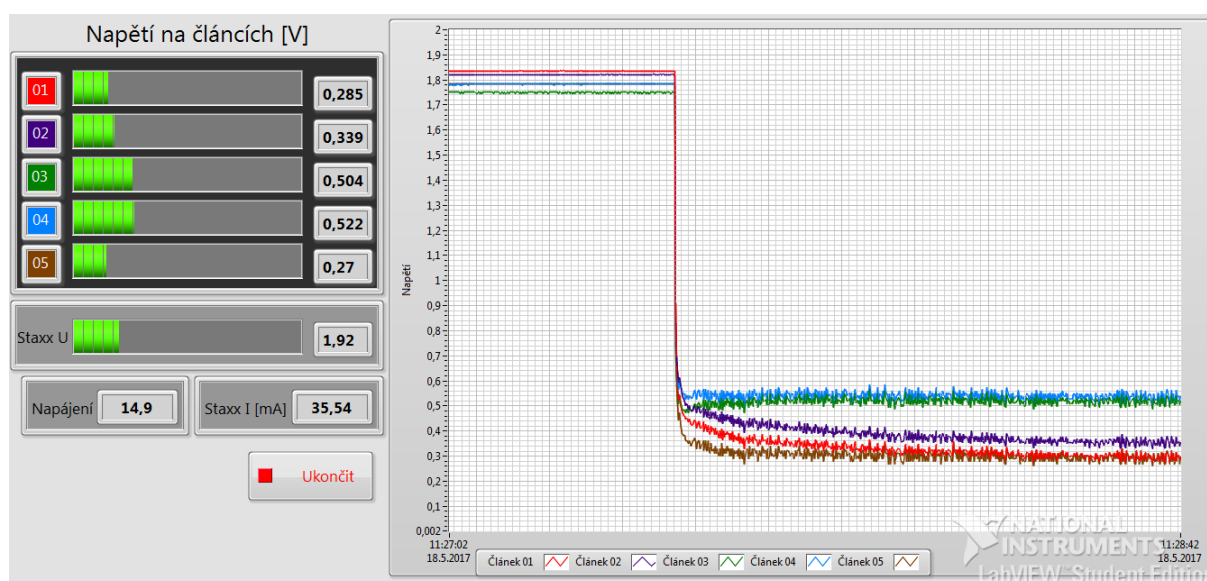


#### 4.2.1 Dodatky k programu

- **Určení velikosti zátěže**
  - Zjištění velikosti zátěže bylo provedeno experimentálně změřením výstupního proudu pomocí multimetru. Zjištěné hodnoty byly využity pro výpočet velikosti zátěže pomocí aplikace Ohmova zákona.
- **Struktura Case a výpočet proudu**
  - Struktura *Case* slouží jako virtuální spínač, který určuje, kdy je zátěž připojena a kdy ne. Výpočet proudu je prováděn tehdy, když napětí klesne pod 5 V, jinak by program nadále počítal proud, i když zátěž není připojena, což by samozřejmě byl nesmysl. Program vychází z informace, že při zapojení zátěže celkové výstupní napětí StaXX klesne z 9 V na 2 V během několika milisekund. Pro výukové a experimentální měření, které je na člancích prováděno, je tento způsob měření proudu dostačující. Výpočet výstupního proudu se vypočítává z Ohmova zákona s použitím výstupního napětí StaXX a velikosti zátěže ( $R=54 \Omega$ ). Výsledný proud se pohybuje řádově v desítkách mA. Při testování byl vypočítávaný výstupní proud zároveň měřen pomocí připojeného multimetru a tím proběhlo ověření správné funkce programu.
- **Color boxy**
  - V programu mají dva významy. První určuje barvu pozadí pro pole s indikátory a druhý určuje barvu pro jednotlivé články. Stejnou, jakou mají v grafu.

### 4.3 Čelní panel nového programu

Čelní panel ve finální verzi byl realizován paletou SILVER s cílem dosažení lepšího vizuálního dojmu. Vlevo nahoře se nachází *Color box*, který obsahuje všechny informace o stavu jednotlivých palivových článků. Každý řádek má svůj *Color box* s číselným označením palivového článku a příslušnou barvou, která je shodná s barvou křivky v modulu *Chart*. Dále řádek obsahuje *Horizontal graduated bar*, který je nastaven na hodnoty napětí od 0 V do 2 V a ukazuje procentuálně napětí na článku. Poslední na řádce je numerický indikátor, který zobrazuje velikost výstupního napětí jednotlivého palivového článku.



Obr. 4.3 Čelní panel nového programu při měření

Pod ukazateli jednotlivých článků se nachází indikátory pro výstupní napětí StaXX, výstupní proud, vstupní napětí do elektrolyzéry a tlačítko pro ukončení programu (Ukončit). Celou pravou stranu obrazovky zaujímá grafický zobrazovač (*Chart*). Modul zobrazovače je nastaven na rozsah 0-2 V a časová základna činí 100 s.

## Závěr

Pro zpracování této bakalářské práce bylo nutné získat vědomosti o palivovém článku a naučit se základy programovacího jazyka LabVIEW. V rámci bakalářské práce byl vypracován program v jazyce LabVIEW verze 2013, který monitoruje palivový článek StaXX. Vychází z původního programu pro monitoring palivového článku, který byl vytvořen v již zastaralé verzi LabVIEW 8.5.

Čelní panel nového programu byl celkově pozměněn. Nejvýraznější změna se týká zobrazení ukazatelů jednotlivých palivových článků (numerické indikátory velikosti napětí a bary), které jsou dány blíže k sobě pro přehlednější čtení dat z obrazovky. Program je nově doplněn o měření vstupního napětí pro elektrolyzér a výpočet výstupního napětí ze StaXX. Zde byl menší problém s určením toho, kdy je zátěž připojena a kdy ne. Tento problém byl vyřešen navržením virtuálního spínače pro rozeznání připojené zátěže, který využívá strukturu Case a porovnává výstupní napětí StaXX. Při měřeních bylo zjištěno, že výstupní napětí při připojené zátěži má hodnotu okolo 2 V a naprázdno okolo 9 V. Pokud tedy klesne napětí pod 5 V, což nastane během ms po připojení zátěže, program začne počítat výstupní proud. Také proběhlo mnoho testování za účelem dosažení správného barevného rozlišení a objektového rozložení obrazovky čelního programu.

Při tvorbě grafické stránky nového panelu bylo nutno přihlížet ke specifikacím a zvyklostem s předchozí verzí, která sloužila u řízení článku několik let a inovace by tak měla na tento trend navazovat. Strukturálně je ale systém přepracován. Nicméně s tím se běžný uživatel při práci s palivovým článkem neseťká.

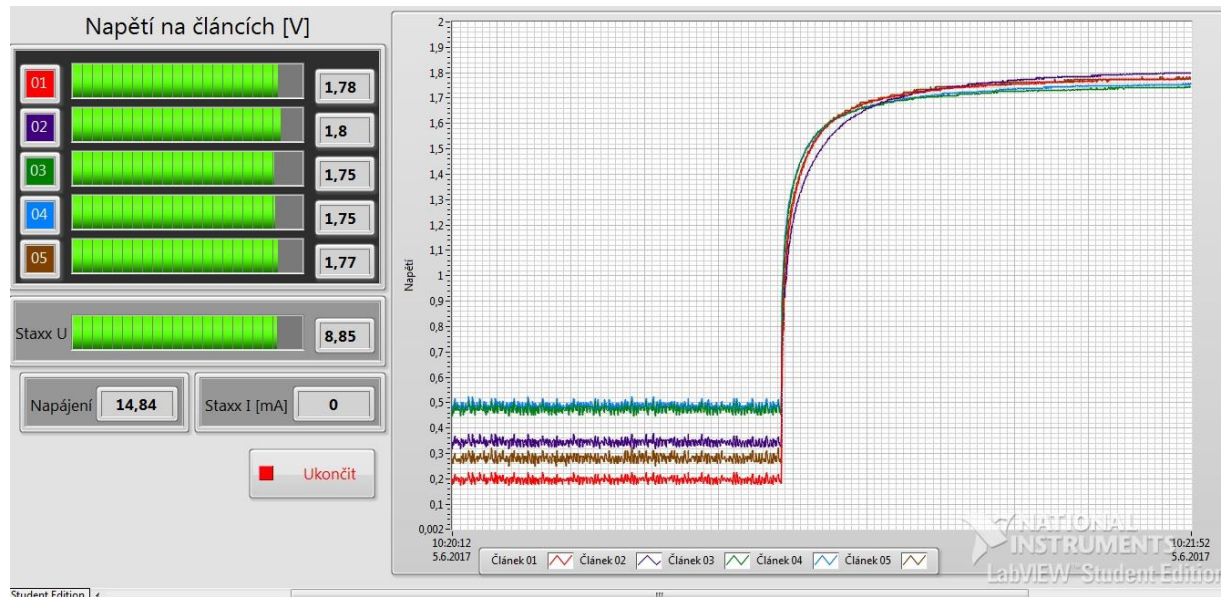
Hlavními přínosy práce bylo vytvoření novější verze programu s dalšími měřícími stanicemi a úprava designu čelního panelu. V této práci bylo možno vycházet z původní verze programu, byla ale nutná funkční úprava a designová vylepšení stávajícího algoritmu programu spojená s přechodem na novou verzi prostředí. Vzhledem k absenci dokumentace a materiálů od původního programu při tvorbě bylo třeba postupovat jako při tvorbě zcela nové verze. V současné době je tedy nová verze plně funkční a je funkčně kompatibilní s verzí starou. Cíle práce bylo dosaženo.

## Seznam literatury a informačních zdrojů

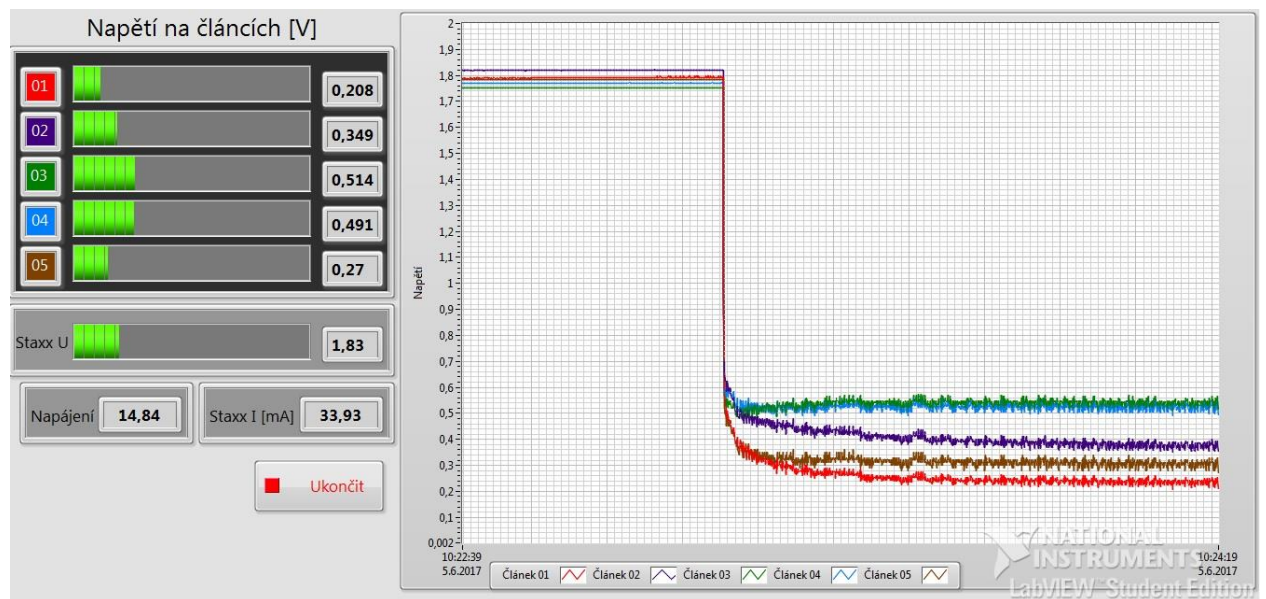
- [1] HORÁK, Bohumil, Jiří KOZIOREK, Miroslav KOPŘIVA, Martin PAPOUŠEK a Zdeněk SLANINA. *Studie pohonu mobilního prostředku s palivovým článkem* [online]. 2005, 203 [cit. 2017-05-23]. Dostupné z: <http://www.mpo-efekt.cz/dokument/5064.pdf>
- [2] DLOUHÝ, Petr a Luděk JANÍK. *Palivové články* [online]. 2007 [cit. 2017-05-24]. Dostupné z: <http://www.hytep.cz/cs/clanky/kategorie-clanku/clanky/587-palivove-clanky>
- [3] FRYDA, D. *Simulace a výpočty energetické bilance kyslíkovodíkového palivového článku pomocí programu LabView*. Bakalářská práce. Brno: Ústav elektroenergetiky FEKT VUT v Brně, 2011, 62 stran. Vedoucí bakalářské práce Ing. Jan Šlezinger.
- [4] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. 1. vyd. Ilustrace Viktorie Vlachová. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [5] National Instruments Corporation, *Manuál ke kurzu LabVIEW Core 1*. Srpen 2011.
- [6] Data acquisition. *Wikipedia, the free encyclopedia* [online]. [cit. 2017-05-24]. Dostupné z: [https://en.wikipedia.org/wiki/Data\\_acquisition](https://en.wikipedia.org/wiki/Data_acquisition)
- [7] BARÁK, Adam a Jiří VÁVRA. *Model v LabVIEW stanovuje spotřebu hybridního vozidla s palivovým článkem* [online]. 2010 [cit. 2017-05-23]. Dostupné z: [http://automa.cz/cz/casopis-clanky/model-v-labview-stanovuje-spotrebu-hybridniho-vozidla-s-palivovym-clankem-2010\\_06\\_41326\\_6284/](http://automa.cz/cz/casopis-clanky/model-v-labview-stanovuje-spotrebu-hybridniho-vozidla-s-palivovym-clankem-2010_06_41326_6284/)

# Přílohy

## Příloha LabVIEW



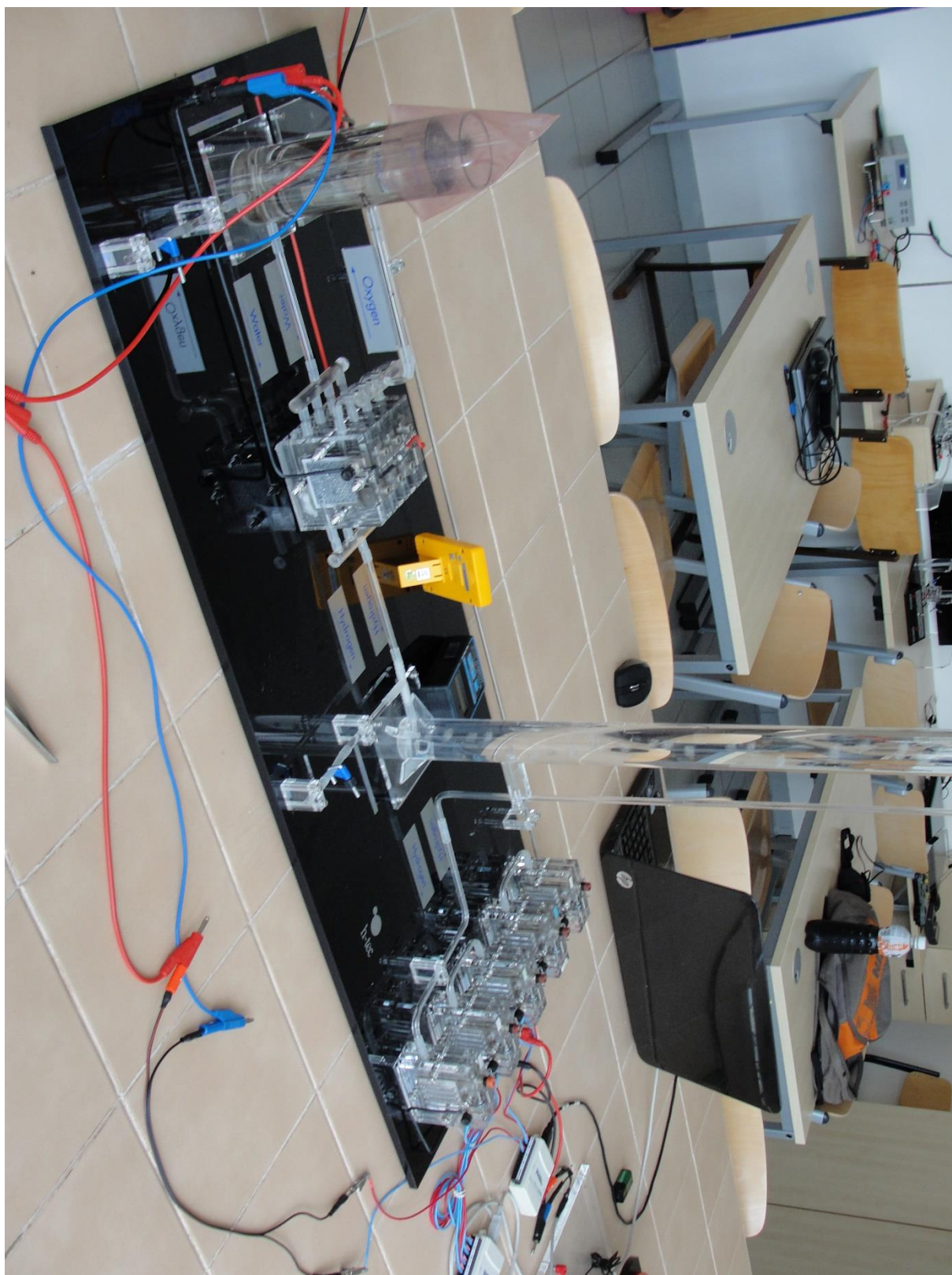
Obr. 1 Čelní panel při odpojení zátěže



Obr. 2 Čelní panel při zapojení zátěže



## Příloha StaXX

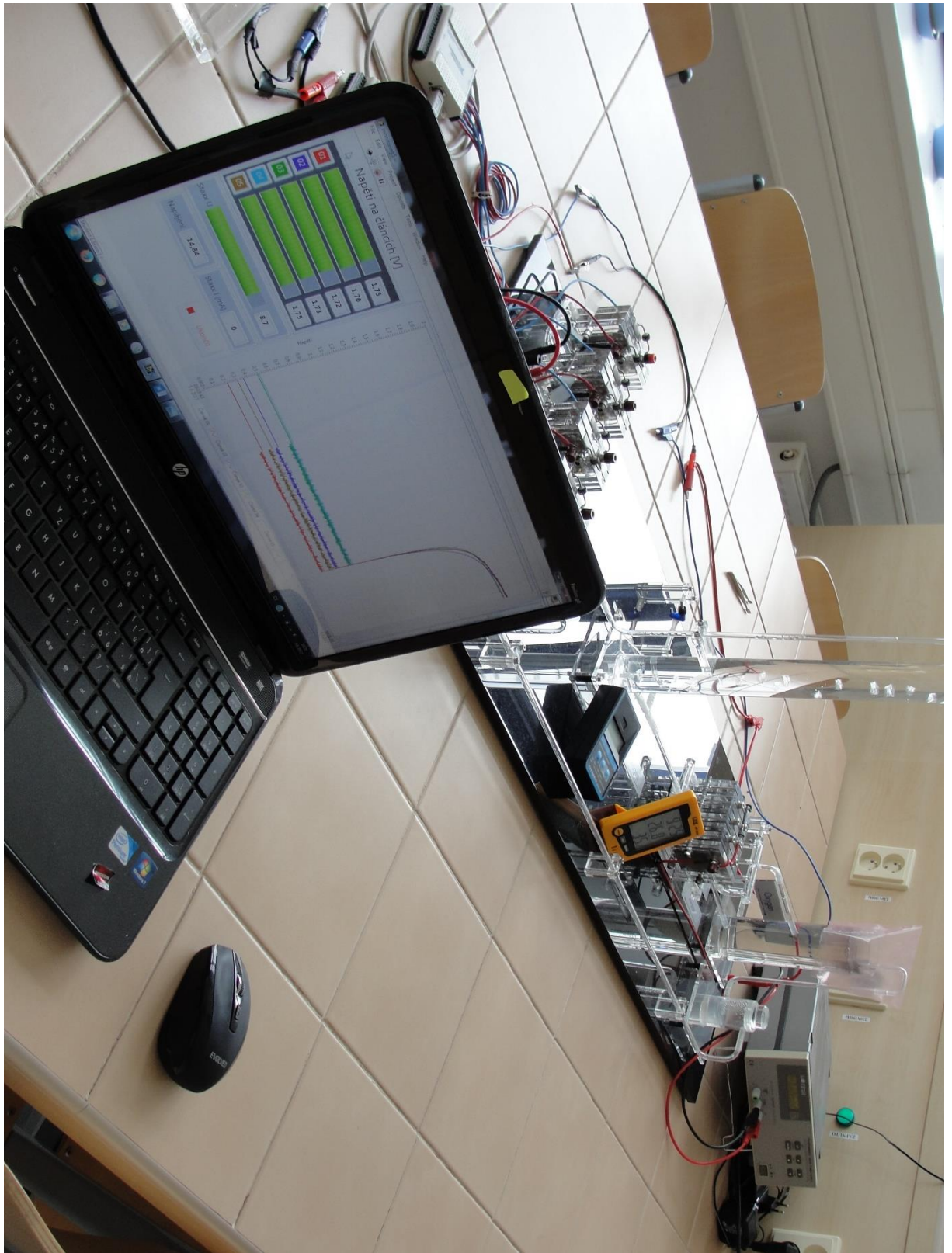


Obr. 3 Pohled na palivový článek StaXX



Obr. 4 Pohled na palivový článek StaXX se zátěží





Obr. 5 Pohled na palivový článek StaXX při měření