



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# **UNIVERZÁLNÍ MĚŘICÍ ÚSTŘEDNA**

UNIVERSAL MEASUREMENT STATION

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. MAREK ŠANTAVÝ**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. VÁCLAV ŠIMEK**

BRNO 2012

## Abstrakt

Cílem této práce je vytvoření praktické aplikace pro měření vstupních signálů. Softwarový modul měřící ústředny je určen pro průmyslové panelové PC. Za hardwarovou platformu bylo zvoleno panelové PC Power Panel 520 od firmy Bernecker + Reiner doplněné o měřící kartu NI USB-6215. Vizualizace a činnost měření dat je řízena programem vytvořeným v LabVIEW 2011. Vzájemné propojení měřící a řídicí (PLC) části je řešeno rozhraním OPC, které je standardem průmyslové komunikace. Vzdálený dohled nad činností měřící ústředny je realizován zařízením Apple iPad 2 prostřednictvím síťově sdílených proměnných. Propojení technologií různých výrobců a řešení přívětivé z pohledu operátora, je bodem zájmu řešení.

## Abstract

The aim of this thesis is to create data acquisition application. The software module is designed for the industrial panel PC. Power Panel 520 from Bernecker + Reiner and the data acquisition board NI USB-6215 from National Instruments represent the fundamental hardware components for this project. Data acquisition and visualization tasks are controlled by a standalone application made in LabVIEW 2011. OPC, industrial standard in communication, is responsible for mutual interconnection between control (PLC) and data acquisition part. Network shared variables and Apple iPad 2 allow remote surveillance for operator. Finally, the creation of user-friendly interface and integration of technologies from different vendors, are main goals of this project.

## Klíčová slova

LabVIEW, panelové PC, OPC, měření dat, vizualizace, TDMS, trigger, konfigurační soubor, vzorkování signálu, iPad, Automation Studio, Data Dashboard, NI-DAQmx, shared variable

## Keywords

LabVIEW, panel PC, OPC, data acquisition, visualisation, TDMS, trigger, config file, signal conditioning, iPad, Automation Studio, Data Dashboard, NI-DAQmx, shared variable

## Citace

Marek Šantavý: Univerzální měřící ústředna, diplomová práce, Brno, FIT VUT v Brně, 2012

# Univerzální měřicí ústředna

## Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením pana Ing. Václava Šimka

.....  
Marek Šantavý  
7. května 2012

## Poděkování

Chtěl bych poděkovat Ing. Václavu Šimkovi za odborné vedení a věcné komentáře při řešení.

© Marek Šantavý, 2012.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Použité technologie a nástroje</b>	<b>4</b>
2.1	LabVIEW 2011	4
2.2	LabVIEW Shared Variable	4
2.3	NI-DAQmx	5
2.4	TDMS	6
2.4.1	Formát souboru	6
2.4.2	Výhody tohoto formátu	6
2.5	Hardware konfigurace řešení	7
2.5.1	B&R Power Panel 520	7
2.5.2	NI-6215 USB	8
2.5.3	Napájecí zdroj	10
2.5.4	Apple iPad 2	10
2.5.5	Schématické zapojení součástí	12
2.6	Softwarová konfigurace řešení	12
2.6.1	Windows 7 Embedded	12
2.6.2	ARwin	13
2.7	Aplikace Data Dashboard for LabVIEW	14
2.8	OPC	14
2.8.1	Komunikační rozhraní	14
2.8.2	Princip funkčnosti	15
2.8.3	Součást řešení	15
2.8.4	Řízení systému	16
2.9	Porovnání existujících nástrojů	16
<b>3</b>	<b>Implementace řešení</b>	<b>17</b>
3.1	Architektura vláknového zpracování	17
3.1.1	Využití architektury	17
3.1.2	Fázová inicializace, deinicializace	18
3.1.3	Zpracování chyb	18
3.1.4	Synchronizace a sdílení dat	18
3.1.5	Modularita a robustnost	19
3.1.6	Architekturní šablona modulů	19
3.2	Měřicí část	20
3.2.1	Vzorkování signálu	21
3.2.2	Synchronizovaná vícekanálová měření	21
3.3	Trigger	21

3.3.1	Řízení běhu úloh . . . . .	21
3.3.2	SW trigger . . . . .	22
3.3.3	HW trigger . . . . .	22
3.4	Konfigurace měřících úloh . . . . .	22
3.4.1	Měření napětí . . . . .	22
3.4.2	Uložení dat . . . . .	24
3.4.3	Konfigurace triggeru . . . . .	24
3.5	Stavy měřících úloh . . . . .	25
3.6	GUI . . . . .	26
3.6.1	Řídící panel . . . . .	27
3.6.2	Správa úloh . . . . .	28
3.6.3	Historie měření . . . . .	28
3.6.4	Stavový řádek . . . . .	28
3.6.5	Náhledový prostor . . . . .	29
3.6.6	Přehlednost a přívětivost ovládání . . . . .	32
3.6.7	Optimalizace pro výkon . . . . .	32
3.7	Systém . . . . .	33
3.7.1	Synchronizace úloh . . . . .	33
3.7.2	Logika řízení, zpracování událostí . . . . .	34
3.8	Obslužné části programu . . . . .	34
3.8.1	OPC . . . . .	34
3.8.2	Shared Variable . . . . .	34
3.8.3	Konfigurační soubor . . . . .	34
3.8.4	Alarm . . . . .	38
3.8.5	Logování . . . . .	38
3.9	Měřená historická data . . . . .	38
3.9.1	Načtení historických dat . . . . .	38
3.10	Vzdálený dohled . . . . .	39
3.10.1	Knihovna sdílených proměnných . . . . .	40
3.10.2	Konfigurace NI Data Dashboard . . . . .	40
3.10.3	Aplikace NI Data Dashboard za běhu . . . . .	40
3.11	Řízení systému . . . . .	41
3.11.1	Ruční řízení . . . . .	41
3.11.2	Řízení přes OPC . . . . .	41
3.11.3	Příklad šíření zpráv měřící ústřednou . . . . .	43
3.11.4	Tvorba OPC konfigurace, nahrání na server . . . . .	43
3.12	Implementace řešení v LabVIEW . . . . .	45
<b>4</b>	<b>Publikování aplikace</b>	<b>48</b>
<b>5</b>	<b>Testování systému</b>	<b>51</b>
<b>6</b>	<b>Závěr</b>	<b>56</b>
6.1	Přínos řešení . . . . .	56
<b>A</b>	<b>Zdrojové kódy vybraných modulů</b>	<b>60</b>
<b>B</b>	<b>Plakát</b>	<b>63</b>

# Kapitola 1

## Úvod

Rychlý rozvoj v oblasti řízení a automatizace je možný díky širokému užívání standardů a pokročilých hardware i software nástrojů. Pro většinu výrobních linek je typické, že jejich součástí je řídicí a měřicí jednotka. Aktuálním trendem je použití panelových PC, které umožňují spojení obou zařízení v jedno. Zároveň dodávají operátorovi, díky dotykové obrazovce, unikátní pracovně přívětivý prostředek činnosti.

Cílem této práce je vytvoření praktické aplikace pro měření vstupních signálů. Je kontrolována buď ručními pokyny operátora nebo programovými příkazy od řídicího PLC. Celý systém je lokalizován v průmyslovém panelovém PC. Pro potřeby řešení tohoto projektu je třeba analyzovat požadavky kladené na hardware i software. Za hardwarovou platformu bylo zvoleno panelové PC Power Panel 520 od firmy Bernecker + Reiner doplněné o měřicí kartu NI USB-6215. Vizualizace a činnost měření je řízena programem vytvořeným v LabVIEW 2011. Vzájemné propojení měřicí a řídicí (PLC) části je řešeno rozhraním OPC, které je standardem průmyslové komunikace. Vzdálený dohled nad činností měřicí ústředny je realizován zařízením Apple iPad 2 prostřednictvím síťově sdílených proměnných. Propojení technologií různých výrobců a řešení přívětivé z pohledu operátora, je bodem zájmu řešení.

## Kapitola 2

# Použité technologie a nástroje

### 2.1 LabVIEW 2011

Součástí portfolia produktů společnosti National Instruments (NI) je grafický návrhový systém LabVIEW. Jeho aktuální verze LabVIEW 2011 reprezentuje vyspělou platformu sledující aktuální trendy ve vývoji informačních technologií a to nejenom v oblasti měření a automatizace. Tato nejnovější verze implicitně využívá například výhod paralelních výpočtů vytvářených programů nebo podpory sady speciálních instrukcí typu SSE.

Vývoj s tímto nástrojem je zcela grafický bez nutnosti psaní zdrojových kódů. Pracovní plocha je rozdělena na návrh grafického uživatelského rozhraní a návrh samotného programu, který je ve výsledku vykonáván za tímto GUI. Návrh programu probíhá přes skládání grafických funkčních bloků s propojením jejich vstupů a výstupů. Dostupné jsou bloky například pro cykly a podmínky. Průběh činnosti programu se řídí dataflow, kdy každý blok je spuštěn teprve ve chvíli, kdy má dostupné všechny data na vstupu.

Rozsáhlá paleta hardwarových zařízení od společnosti NI rozšiřuje již tak silný nástroj, jakým je právě LabVIEW. Jsou to zařízení určená pro různé oblasti průmyslu, rychlého a přesného měření. Dalšími rozšířeními pro LabVIEW jsou toolkity a moduly. Za zmínku stojí především nástroje pro vývoj programů pro FPGA, zpracování obrazu, real-time systémy nebo pokročilý matematický aparát pro signálovou analýzu. Dostupnost těchto nástrojů předurčuje projekt pro snadnou rozšířitelnost a prostor pro budoucí nadstavby.

Při řešení projektu byly využity pouze standardně dostupné nástroje, které jsou součástí balíku vývojového prostředí LabVIEW.

### 2.2 LabVIEW Shared Variable

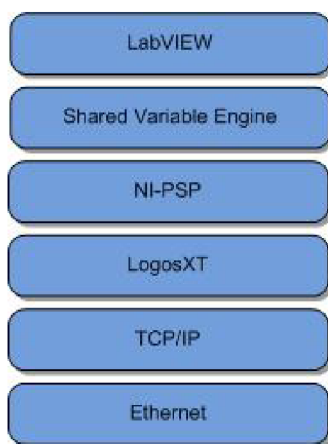
Dostupné typy proměnných využitelných v prostředí LabVIEW jsou rozděleny do přehledné tabulky 2.1.

Na základě tabulky je vyvozeno, že pro potřeby sdílení (předávání) proměnných v síti lze využít jediný typ proměnných a to konkrétně Network-published shared variable. Dále v odseku bude uvažován právě tento typ proměnných. Na obrázku 2.1 lze vidět jednotlivé vrstvy pro komunikaci Network shared variable. Pouze v krátkosti budou tyto vrstvy představeny s objasněním jejich významu.

Shared Variable Engine (dále SVE) je hostitelem proměnných na síti. Jestliže má být proměnná dostupná na síti, pak je třeba ji nejdříve zavést pomocí SVE [23]. SVE poté

Typ proměnné	Rozsah
Local variable	Jediný zdrojový soubor
Global variable Functional global variable Single-process shared variable	Více zdrojových souborů na stejném počítači
Network-published shared variable	Dostupnost v síti

Tabulka 2.1: Typy proměnných v LabVIEW



Obrázek 2.1: Vrstvy komunikace Network shared variable [23]

předává aktualizované hodnoty proměnných všem přihlášením příjemcům. Jedná se tedy o vztah více klientů, jeden server. Restartování stanice, která je hostitelem některé ze sdílených proměnných, nemá vliv na dostupnost těchto proměnných v síti. Pokud je třeba tyto proměnné odstranit, musí to být provedeno příslušným příkazem.

Vrstva NI-PSP je zodpovědná za optimalizaci a zajištění doručení proměnných, jak uvádí [20]. K tomuto účelu využívá spolehlivý protokol TCP/IP.

Vysokou propustnost u aplikací využívajících síťovou komunikaci zajišťuje vrstva LogosXT. Na základě výkonnostních testů bylo zjištěno, že pro dosažení maximální propustnosti je lepší posílat větší objemy dat v delších intervalech [23]. Malé objemy dat v krátkých intervalech zapříčiňují vícenásobnou komunikaci nadbytečných dat (např. hlavičky paketů). Důsledkem tohoto zjištění bylo vytvoření 8KB přenosového bufferu, který je komunikován každých 10ms, nebo dříve při dosažení zaplnění bufferu. Pro aplikaci s požadavkem na minimální zpoždění přenosu dat v síťové komunikaci lze programově řídit předčasný přenos dat. V tomto případě je třeba zvážit dopad na propustnost dat v síti.

## 2.3 NI-DAQmx

K ovládání celé řady měřících karet od společnosti National Instruments slouží ovladač s názvem NI-DAQmx. Tento vospělý a pokročilý ovladač následuje postupný vývoj, který se odehrál v oblasti ovladačů společnosti National Instruments.

Příčinou vzniku ovladače NI-DAQmx v pozdních 90-tých letech minulého století bylo především to, že jeho předchůdce Traditional NI-DAQ nedokázal patřičně reflektovat nejnovější trendy na poli hardwaru a softwaru [22]. Z toho důvodu jakékoliv rozšíření funkciona-



lity nebo přidání nového zařízení vedlo pouze k dalšímu zesložiténí a zanesení nových chyb do stávajícího ovladače. Byly proto vytyčeny cíle pro vytvoření nového robustního ovladače, který bude vůči všem těmto změnám odolný a navíc bude dosahovat lepších výkonnostních parametrů. Tyto parametry splňuje právě ovladač NI-DAQmx, kdy mezi hlavní cíle tohoto ovladače patří zvýšení výkonnosti operací měření dat, zvýšení kvality ovladače samotného, podpora vícevláknového měření a v neposlední řadě i usnadnění rozšiřování stávající funkcionality. Zvýšení výkonnosti bylo dosaženo sadou pokročilých API, které dovolují uživateli získání více kontroly nad časově náročnými operacemi, které jsou vykonávány v průběhu měření (těmi jsou například potvrzení, rezervování zdrojů). Zpracování výjimečných stavů jako odpojení datového kabelu, neočekávané ukončení měřicí úlohy a sada automatizovaných testů nad změnami v ovladači zajišťují vysokou kvalitu všech dodávaných ovladačů společnosti National Instruments. Pro účely tohoto projektu hraje velkou roli podpora vícevláknových měření, kdy je dovoleno různým vláknům přistupovat k ovladači v jeden čas. Díky tomu není ovladač úzkým hrdlem při vícenásobném přístupu k měřícím zařízením.

## 2.4 TDMS

Pro uchování naměřených dat na datovém úložišti existuje celá řada formátů souborů. Od těch nejjednodušších na úrovni textových ASCII souborů, až po binární formáty. Plain-textový formát lze využít zejména pro omezené množství zapisovaných dat, bez potřeby náhodného přístupu do obsahu souboru. Tyto typy souborů jsou snadno editovatelné a zobrazitelné. Požadavek na výkon při práci se soubory splňují binární formáty. Nestandardní binární formáty je třeba opatřit detailním popisem struktury dat, které obsahují [13].

### 2.4.1 Formát souboru

Formát souboru TDMS následuje před připravenou strukturu pro uchování měřených dat s možností doplnění dodatečných informací, například o podmínkách průběhu měření [13]. Základní organizační jednotkou jsou skupiny kanálů (Channel Group), které dále obsahují jednotlivé kanály (Channel). Měřená data jsou uchovávána v kanálech. Pro vložení doplňujících informací slouží vlastnosti (Properties), které lze přiřadit celému souboru, skupině kanálů nebo samotným kanálům.

### 2.4.2 Výhody tohoto formátu

Platforma LabVIEW přichází se standardizovaným binárním formátem souboru TDMS. Formát TDMS přináší benefity, které jsou dostupné buď pouze u plain-text, nebo pouze u binárních souborů. Díky binární struktuře si zachovává vysokou výkonnost čtecích/zápisových operací. Jednoduchost prohlížení souborů je zajištěna podporou nástrojů NI a programovým toolkitem pro práci se soubory ve vyvíjených aplikacích. Formát je veřejně dostupný a dobře zdokumentovaný. Nutnost využití nástrojů NI není vyžadována. Pro tento případ lze soubor TDMS konvertovat do tabulky Microsoft Excel. Tato konverze je však dostupná pouze pro čtení souborů díky rozšíření (add-onu) a nelze s jeho pomocí soubory ve formátu TDMS opět ukládat.

## 2.5 Hardware konfigurace řešení

Hardwarové sestavení projektu je tvořeno dvojicí zařízení. Prvním z nich je průmyslové panelové PC s dotykovou obrazovkou. Panelové PC zajišťuje činnost řízení PLC pro akční členy, kterými ovládá požadované procesy. Část systému, která vykonává řízení, již leží mimo rozsah řešení této práce a proto již nebude dále rozebírána. Vzhledem k potřebě zpětné vazby v podobě uchování naměřených dat běží na tomto PC i úloha shromažďování a ukládání dat.

Pro měření signálů byla zvolena USB měřicí karta NI-6215 USB, která svými vlastnostmi plně dostačuje kladeným požadavkům na výkonnost a přesnost měřených dat. Využití komunikačního rozhraní panel PC umožňuje budoucí možnost rozšíření nebo snadnou výměnu této karty z důvodu servisování.

### 2.5.1 B&R Power Panel 520

Panelové PC, které je produktem společnosti Bernecker & Rainer (B&R), reprezentuje aplikaci nejnovějších technologií výroby dotykových panelových PC. Společnost B&R se zaměřuje na odvětví průmyslové automatizace a řízení. Její zastoupení v širokém rozsahu průmyslových odvětví od textilního, potravinářského až po kovozpracující a elektrotechnický průmysl předznamenává vysokou úroveň kvality a nasaditelnosti jejich řešení. Využití Panel PC je zamýšleno jako součást rozvodné skříně, která bude přímo součástí výrobního nebo testovacího zařízení.

#### Specifikace

Cílem nasazení panelového PC je prostředí, které vyžaduje instalaci PC doplněného o zobrazovací zařízení (monitor) v omezeném prostoru a to nejčastěji v rozvodných skříních. U této řady zařízení byl kladen zvláštní důraz na jejich mechanickou odolnost proti poškození z vnějšího prostředí. Další pozitivní vlastností je i bez ventilátorová instalace, jejíž správnému chodu a údržbě musí jinak být v průmyslu věnovaná náležitá pozornost. Všechny komponenty, které vyžadují chlazení z důvodu zahřívání jsou uzpůsobeny tak, aby jejich teplo bylo rozváděno přes heat sink na zadní část tohoto zařízení.

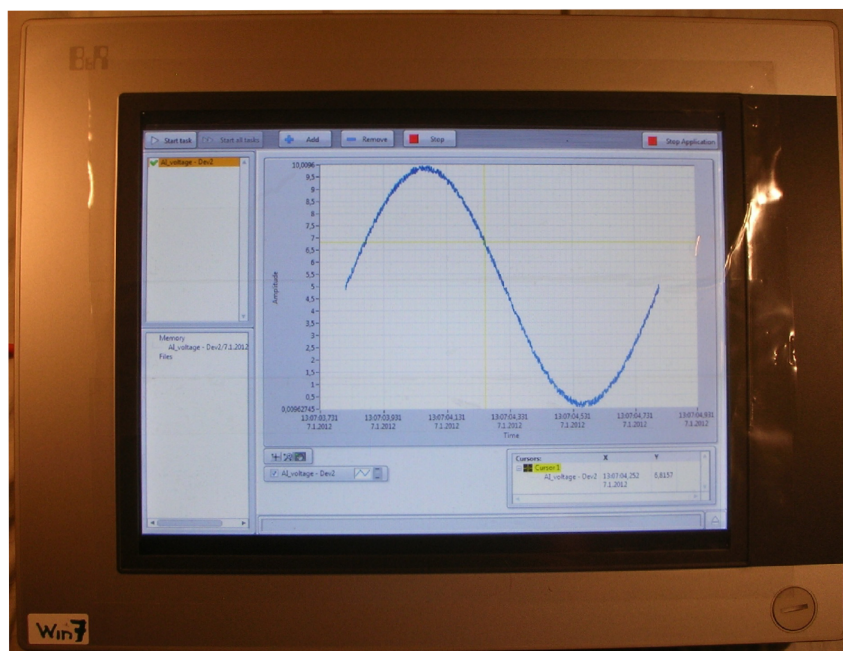
Název zařízení	Power Panel 520
Monitor	15" XGA TFT s dotykovou obrazovkou
Procesor	Intel Atom Z530, 1600 MHz, 533 MHz FSB, 512 kB L2 cache
Hlavní paměť	SO-DIMM DDR2 2048MB PC2-5300
Operační systém	Microsoft Windows Embedded Standard 7 Premium 32-bit
Datové úložiště	Compact Flash 8192MB B&R
Rozhraní	1x RS232, 3x USB 2.0, 1x Ethernet 10/100/1000
Ventilátory	Žádné
Odolnost	IP66 (přední strana), IP65 (zadní strana)
Hmotnost	5100g

Tabulka 2.2: Specifikace B&R Power Panel 520

S výhodou lze také využít splnění standardů odolnosti IP66 pro přední část panelu a IP65 pro zadní část panelu [3]. Tyto vlastnosti zajišťují odolnost zařízení proti všem vlivům

prostředí. Jednoduchá konstrukce z nerezové oceli minimalizuje hrany a mezery, kde se běžně usazuje prach a nečistoty.

Konfigurace použitého systému je přehledně uvedena v tabulce 2.2 s názornou ukázkou podoby spuštěné aplikace na obrázku 2.2.



Obrázek 2.2: B&R Power Panel 520

## 2.5.2 NI-6215 USB



Obrázek 2.3: NI USB-6215 [24]

Pro potřeby řešení výsledného systému byla zvolena měřicí karta, která splňuje dané požadavky o dostatečném počtu analogových a digitálních vstupů. Zohledněno bylo i komunika-

Název zařízení	NI USB-6215	
Operační systém	Linux, Mac OS, Windows	
Analogové vstupy		
	Jednoduché	16
	Diferenciální	8
	Rozlišení	16 bitů
	Vzorkování	250 kS/s
Maximální rozsah měřeného napětí	-10 V, 10 V	
Minimální rozsah měřeného napětí	-200 mV, 200 mV	
Paměť měřicí karty	4095 vzorků	
Digitální V/V		
	Obousměrné kanály	0
	Vstupní kanály	4
	Logické úrovně	TTL
Trigování a synchronizace		
	Triggování signálů	Digitální

Tabulka 2.3: Vybrané parametry měřicí karty NI USB-6215

ční rozhraní USB, jako univerzální a dostatečně dimenzované pro potřeby měření. Výhoda tohoto výběru spočívá především v připravených ovladačích, které výrobce zařízení dodává zdarma. Ovladače tohoto zařízení lze jednoduše ovládat díky těsné integraci v návrhovém prostředí LabVIEW, kterou umožňuje driver NI-DAQmx.

Pro získání představy o vlastnostech měřicí karty poslouží krátký přehled jejich hlavních parametrů [10] v přehledové tabulce 2.3. Zařízení je chráněno proti mechanickému poškození svým odolným plastovým obalem, viditelným na obrázku 2.3.

Maximální dosažitelná vzorkovací frekvence 250 kS/s je sdílená mezi použitými kanály v měřicí úloze. Pokud je třeba vzorkovat 5 kanálů maximální vzorkovací frekvencí, pak lze dosáhnout 50 kS/s/ch (vzorků za sekundu na kanál). Význam této vlastnosti je především ve snížení pořizovacích nákladů na zařízení. Součásti zodpovědné za úpravu signálu a jeho analogově/digitální převod jsou všemi kanály sdílené [10]. Při vzorkování signálů na více kanálech dochází k multiplexování při přístupu k tomuto sdílenému zdroji.

S rozvojem externích sběrnice (USB, Ethernet a dalších) se zvýšil požadavek na dostupnost měřících zařízení pro tyto sběrnice. Jejich výhodou byla snadná manipulace a rychlé připojení k PC. Oproti interním sběrnícím (např. PCI) nemají tyto externí sběrnice specifikacemi garantovanou přenosovou rychlost a zpoždění [19]. K tomu, aby byla zajištěna nízká latence a dostatečná propustnost několika datových streamů, byla vytvořena patentovaná technologie NI Signal Streaming [19]. U konkrétního modelu karty NI USB-6215 tato technologie spolupracuje s USB-STC2. NI USB-STC2 obsahuje customizovaný kontrolér, který dovoluje čtyřem vysokorychlostním DMA kanálům přímý přístup k příslušným USB-endpoint. Tímto způsobem, společně s kontinuálním vysíláním požadavků na data (z hostitelské strany), jsou zajištěny rychlé přesuny dat. Ke snížení latence byl vytvořen protokol výměny zpráv, který je kontrolér zařízení schopen rychle a efektivně zpracovávat. Všechna tato vylepšení dělají ze sběrnice USB spolehlivou, rychlou a průmyslově využitelnou sběrnici [19].

### 2.5.3 Napájecí zdroj

K napájení dotykového panelu byl zvolen průmyslový napájecí zdroj od společnosti Mean Well. Konkrétně se jedná o typové značení MDR-20-24, které je vyobrazeno na obrázku 2.4. Tento zdroj splňuje požadavky kladené na vlastnosti napájecího zdroje. Je vybaven



Obrázek 2.4: Spínaný zdroj Mean Well MDR-20-24 [21]

ochranou proti přepětí a přetížení. Další výhodou je snadná instalace na DIN lištu, čehož lze úspěšně využít v rozvodných skříních. Krátký přehled specifikací napájecího zdroje podle [9] je možné nalézt v tabulce 2.4.

Název zařízení	Mean Well MDR-20-24
Výstupní napětí	24 V
Rozsah výstupního proudu	0 ... 1 A
Výstupní výkon	24 W
Tolerance výstupního napětí	1%
Zvlnění a šum (špička špička)	150 mV
Účinnost	84%
Ztrátový proud, max.	1 mA
Nastavení výstupního napětí	$\pm 10\%$

Tabulka 2.4: Specifikace Mean Well MDR-20-24

### 2.5.4 Apple iPad 2

Soudobý trend vzdáleného dohledu byl zohledněn i při výběru zařízení, které by tyto požadavky splňovalo. Tímto zařízením je právě Apple iPad 2, který představuje v současné době nejlepší kombinaci hardwarových vlastností s propracovaným ekosystémem dostupných aplikací. Na trhu je k dispozici celá řada modelů, které se vzájemně liší dostupnou uživatelskou pamětí a podporou bezdrátového připojení. Podobu zvoleného modelu (s Wi-Fi + 3G a 32GB uživatelské paměti) lze vidět na obrázku 2.5. Dostupnost připojení Wi-Fi nebo

3G dělá ze zařízení iPad univerzální mobilní nástroj s vynikajícím zajištěním konektivity. Specifikace tohoto modelu jsou shrnuty v tabulce 2.5.

Název zařízení	Apple iPad 2 (model Wi-Fi + 3G)
Výška x Šířka x Hloubka	241,2 mm x 185,7 mm x 8,8 mm
Hmotnost	613 g
Uživatelské paměť	32 GB
Bezdrátové připojení	Wi-Fi (802.11a/b/g/n)
	Technologie Bluetooth 2.1 + EDR
Mobilní připojení	UMTS/HSDPA/HSUPA (850, 900, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz)
Displej	Lesklý širokoúhlý displej s podsvícením LED, ovládáním Multi-Touch, technologií IPS a úhlopříčkou 9,7 palce
- Rozlišení	1024 x 768 pixelů
Baterie	Vestavěná 25Wh Li-Pol
- Výdrž	Až 10 hodin surfování na webu v síti Wi-Fi

Tabulka 2.5: Specifikace Apple iPad 2, model Wifi + 3G

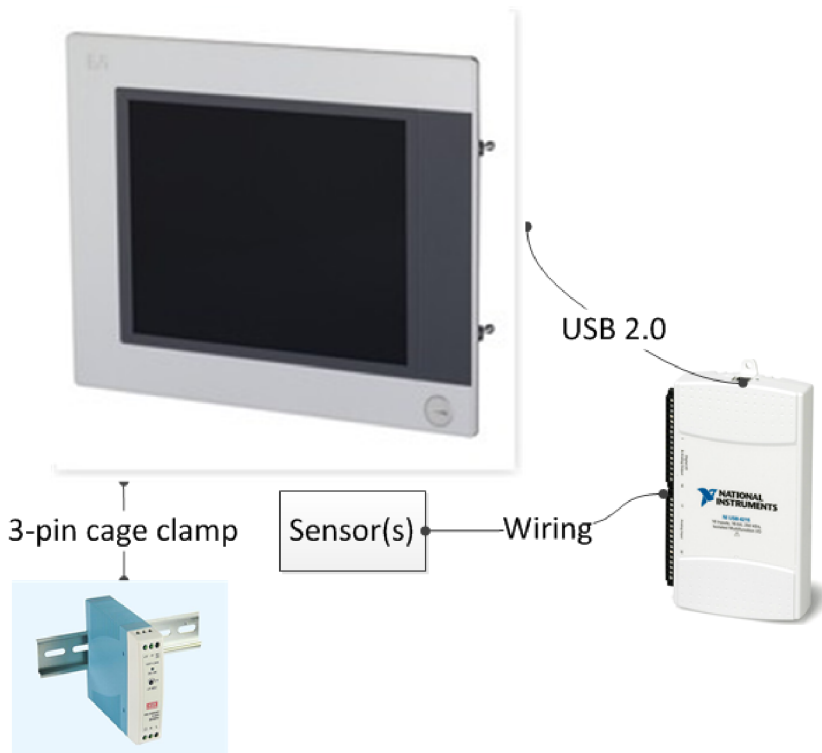


Obrázek 2.5: Apple iPad 2, model Wifi + 3G [25]

Vzhledem k faktu, že zařízení iPad dokáže pracovat pouze za využití bezdrátové komunikace, pak je třeba zajistit připojení dotykového průmyslového panelu do sítě LAN nebo WAN. Díky tomu pak tablet iPad dokáže s dotykovým panelem komunikovat. Z bezpečnostních důvodů se jako lepší varianta jeví síť LAN, která je chráněna proti útokům z vnější sítě. Řešení síťové infrastruktury a její konfigurace však již přesahuje rozsah této diplomové práce.

## 2.5.5 Schématické zapojení součástí

Na obrázku 2.6 je schématicky znázorněno výsledné hardwarové sestavení systému do použitelné podoby. V tomto sestavení je systém schopen provádět všechny své úkony tak, jak je popsáno dále v této technické zprávě.



Obrázek 2.6: Schématické vyobrazení systému ([6], [21] a [24])

## 2.6 Softwarová konfigurace řešení

### 2.6.1 Windows 7 Embedded

Použitý operační systém dotykového panelu následuje poslední trend v oblasti průmyslové automatizace. Tím je myšleno nasazení operačního systému Windows® Embedded Standard 7. Tento systém je nástupcem dříve široce používaného Windows® XP Embedded. Podle [5] obsahuje systém Windows® 7 Embedded kromě vlastností obsažených v systému Windows® 7 Professional i další nadstandardní vlastnosti. Mezi nimi je například Registry Filter nebo Enhanced Write Filter.

Systém Windows® Embedded Standard 7 je dodáván v základní a Premium verzi. Premium verze se liší v podpoře vícejazyčného prostředí a nadstandardních vlastností, jako například Applocker. Nástroj Applocker umožňuje monitorování a zabránění spuštění nevyžádaných aplikací a tak chrání dotykový panel před nepovolenými zásahy, jak uvádí [5]. Charakter řešení projektu však nevyžaduje takovou míru zabezpečení a proto nebyla Premium verze uvažována.

Samozřejmostí je dostupnost 32-bitové a 64-bitové verze. Vzhledem k použitému procesoru v tomto dotykovém panelu je nainstalována 32-bitová verze. Význam 64-bitové verze

se projeví zejména při nasazení aplikací požadujících vysoký výkon.

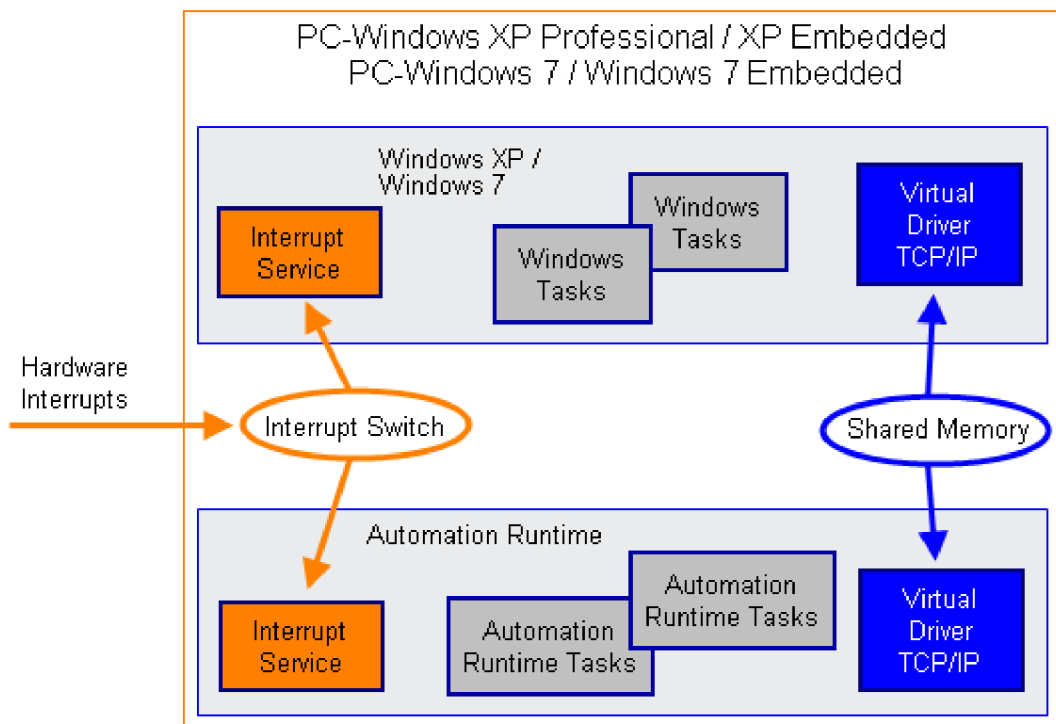
## 2.6.2 ARwin

ARwin je systém, který dovoluje současný běh běžného operačního systému (Windows 7) a real-time systému na jediném panelovém PC [2]. Tého vlastnosti je dosaženo vynucením si běhu real-time systému nad operačním systémem s tím, že operační systém běží jako real-time úloha minimální priority. Díky dnešní výkonové úrovni PC systému lze zajistit zcela plynulý běh bez vzniku úzkého výkonového hrdla.

Po načtení operačního systému Windows je automaticky spuštěn loader ARwin. Pouze jeden systém může v jednu chvíli pracovat s dostupnými hardwarovými prostředky PC. Tohoto chování systému lze dosáhnout především díky upravené verzi operačního systému, která dovoluje správnou spolupráci s real-time systémem. Real-time systém a operační systém Windows mají k dispozici oddělené paměťové prostory. Část z tohoto prostoru slouží jako sdílená paměť pro zajištění komunikace přes TCP/IP rozhraní.

### Subsystém přerušení

Pro potřeby vzájemného oddělení jsou přerušení rozdělena na úrovně [2]. Při vzniku přerušení je nejdříve o tomto přerušení informován real-time systém (vyobrazeno na obrázku 2.7), který vykoná všechny své potřebné úkony. Jakmile byly všechny nutné kroky (úlohy požadující zpracování) splněny a není třeba zpracovat žádná další data, pak je přepnut kontext řízení systému Windows.



Obrázek 2.7: ARwin s podsystémem přerušení [2]



## Rozdělení řešení Windows/real-time systém

Celá část implementovaného řešení běží pouze pod operačním systémem Windows. Časově kritické řízení systému lze vykonávat v rámci real-time. Pro potřeby měření v rozsahu projektu není dosahováno limitace výkonnosti systému Windows. Řešení tedy běží jako samostatný proces, aktivovaný po spuštění systému a to po celou dobu jeho běhu.

## 2.7 Aplikace Data Dashboard for LabVIEW

Operační systém iOS je společným operačním systémem pro zařízení od firmy Apple. Mezi těmito zařízeními jsou iPad, iPhone a iPod. Vývojáři tak mohou s výhodou využít jednotnou platformu pro různá zařízení. Aktuální sestavení operačního systému uvažovaného v tomto řešení je iOS verze 5.1.

Společnost National Instruments vydala 9.12.2011 bezplatnou aplikaci s názvem Data Dashboard for LabVIEW, která je dostupná v App Store. Ikona aplikace je vyobrazena na obrázku 2.8. Tato aplikace umožňuje vzdálené sledování Network-Published Shared Vari-



Obrázek 2.8: Ikona aplikace Data Dashboard for LabVIEW [18]

ables (síťově sdílených proměnných). Díky tomu lze sledovat prostřednictvím aktualizace hodnot sdílených proměnných, na zařízeních od firmy Apple, činnost aplikací vytvořených v LabVIEW. V současné době je umožněna pouze jednostranná komunikace, ale dá se předpokládat, že další aktualizace a vývoj aplikace změní ve prospěch i tuto vlastnost. Aplikaci lze s úspěchem provozovat i na mobilním telefonu iPhone. Avšak díky malé zobrazovací ploše není její používání natolik uživatelsky přívětivé.

## 2.8 OPC

Za vznikem OPC (ve zkratce OLE for Process Control) stojí přední výrobci hardwaru v oblasti řízení a automatizace za spolupráce společnosti Microsoft a neziskové organizace OPC Foundation, jak uvádí [26]. Právě díky komerční síle společnosti Microsoft bylo umožněno masové rozšíření protokolu a jeho rychlý nástup jako průmyslový standard, v současnosti využíváný po celém světě.

### 2.8.1 Komunikační rozhraní

OPC reprezentuje rozhraní vzájemného propojení senzorů a aktuátorů, ale i databázového úložiště se systémem podnikového řízení. Cílem je umožnění vzájemné komunikace OPC kompatibilních zařízení a odstranění nutnosti vytváření hardwarově specifických ovladačů, které jsou v důsledku změny hardware často nekompatibilní [17]. Dalším cílem je zajistit bezkonfliktní přístup k zařízením, která mohou být dodávána různými výrobci, což zajišťuje unifikovaný způsob komunikace.

## 2.8.2 Princip funkčnosti

Základem standardu OPC jsou technologie společnosti Microsoft, jmenovitě se jedná o COM a DCOM. Jelikož procesy operačního systému jsou vzájemně odděleny a chráněny proti neoprávněnému přístupu, nelze jednoduše mezi nimi komunikovat a předávat si data. K tomuto účelu slouží technologie COM, která umožňuje vzdálené spouštění procedur a tedy i předávání dat. Rozšíření DCOM spočívá ve využití této komunikace mimo lokální PC a to v LAN nebo WAN.

V návaznosti na nabývající význam alternativních operačních systémů lze OPC využít i u nich. Je však potřeba vyřešit absenci vrstvy DCOM, která bývá nejčastěji nahrazována komerčními dedikovanými produkty.

Architekturou, kterou OPC [17] pro komunikace využívá je klient/server. Základními komunikovanými typy OPC jsou vlastnosti, metody (vykonání specifického úkonu), události a alarmy (informují o výskytu změny ve sledovaném procesu). Význam událostí a alarmů spočívá v možnosti rozhlášení změny sledované hodnoty určitým klientům a odstranění nutnosti neustálého vyčítání dat klienty (polling).

OPC DA (Data access) strukturuje data umístěná na serveru do skupin (Group), které uchovávají informace o sobě samých a položkách které obsahují. Dále uchovávají četnost, s jakou jsou aktualizovány položky. Součástí popisu skupiny je i mrtvá zóna pro identifikaci procenta počtu změn hodnot pro aktualizaci hodnoty položky. Jestliže počet změn hodnoty leží pod tímto prahem, pak hodnota položky není aktualizována a nezpůsobí zbytečné zatížení komunikačních linek. Na serveru jsou umístěny položky, které nesou jedinečný identifikátor, časové razítko aktualizace hodnoty, hodnotu samotnou a kvalitu položky (good, bad, uncertain) pro identifikaci stavu komunikační linky nebo jiných chyb.

## 2.8.3 Součást řešení

Architektura řešení založená na dvou oddělených částech pro procesní řízení PLC a měřicí část nutně vede k potřebě vzájemné spolupráce těchto celků. Řídicí PLC část obsahuje hlavní rozhodovací mechanismus a měření je tedy pouze sekundární podřízenou součástí celého systému. Pro předávání příkazů, reportování stavu a výsledků bylo třeba zvolit ověřený spolehlivý systém. Tímto je právě OPC, které zajišťuje dostupnou rychlou výměnu dat a samotné vlastnosti protokolu snižují dobu nasazení produktu s tímto rozhraním na trh.

Hlavními výhodami jsou dostupná a prověřená komunikační rozhraní, která následují cíle řešení v podobě modulárního a snadno rozšířitelného systému.

Společnost Bernecker + Reiner nabízí k použití dva typy různých OPC serverů [2]. První z nich je OPC server běžící na platformě Windows 32, který je založený na PVI a umožňuje simultánní komunikaci několika cílových zařízení. Druhým typem OPC serveru je takový, který běží na systému Automation Runtime (od verze AR 3.0 a výše). Tento typ OPC serveru simultánní komunikaci neumožňuje. OPC server pro Automation Runtime splňuje specifikace OPC Common 1.1 (výpis a připojení k OPC serverům), OPC Data Access (DA) 3.00 (přenosy real-time hodnot proměnných přes OPC) a OPC A/E (Alarm Events) 1.10 (specifikace přenosů alarmů a událostí). Kromě těchto specifikací servery nepodporují žádnou z dalších OPC specifikací. V řešení této práce je využit OPC server založený na systému Windows 32.

## 2.8.4 Řízení systému

Měřicí část řešení je ovládána příkazy skrze OPC proměnné, které jsou v pravidelných intervalech vyčítány pro kontrolu jejich změny. Možnost využití alarmů se v tomto řešení ukázala jako neopodstatněná, ale do budoucna není kategoricky vyloučena. Více o řízení prostřednictvím příkazů předávaných přes OPC bude napsáno v samostatné sekci této technické zprávy.

## 2.9 Porovnání existujících nástrojů

Alternativou k řešení Univerzální měřicí ústředny jsou dva nástroje, shodou okolností rovněž od společnosti National Instruments. Těmito nástroji jsou LabVIEW SignalExpress (dále SignalExpress) a NI DIAdem (dále DIAdem). První z jmenovaných (SignalExpress) je součástí balíku nástrojů vyšších verzí grafického návrhového prostředí LabVIEW. SignalExpress je nástrojem, který minimalizuje dobu potřebnou pro zahájení rychlého měření s ukládáním, vizualizací a analýzou dat. Nástroj je určen především pro snadné prvotní testování a ověřování funkčnosti měření. Druhým nástrojem je DIAdem, který nachází své uplatnění při data-miningu, vytváření reportů a protokolů měření. Oba tyto nástroje umožňují spouštění kódu vytvořeného v LabVIEW.

Nevýhodou obou těchto nástrojů je nedostupnost zdrojových kódů, kdy programátor nemůže pozměnit funkčnost těchto nástrojů. Druhou velmi významnou nevýhodou je nutnost licencování při každém jejich nasazení. DIAdem je licencován samostatně a SignalExpress je licencován v rámci balíku s LabVIEW. Tím pádem zůstávají vysoké pořizovací náklady i při opakovaném nasazení řešení. Vzhledem k povaze těchto nástrojů je těsná integrace s PLC systémem (prostřednictvím OPC) velmi limitovaná až nemožná.

Na základě těchto zjištění padla volba na vytvoření vlastního řešení Univerzální měřicí ústředny. Plný přístup ke zdrojovému kódu umožní přizpůsobování nástroje nejnovějším trendům a rychlé odstranění potenciálních chyb. Tvorba spustitelných binárních souborů .exe minimalizuje pořizovací náklady při opakovaném nasazení řešení a tím maximalizuje vyhotoviteli řešení potenciální zisk. Všechna tato fakta vedla k úsilí o vytvoření implementace řešení, které je popsáno v následující kapitole.

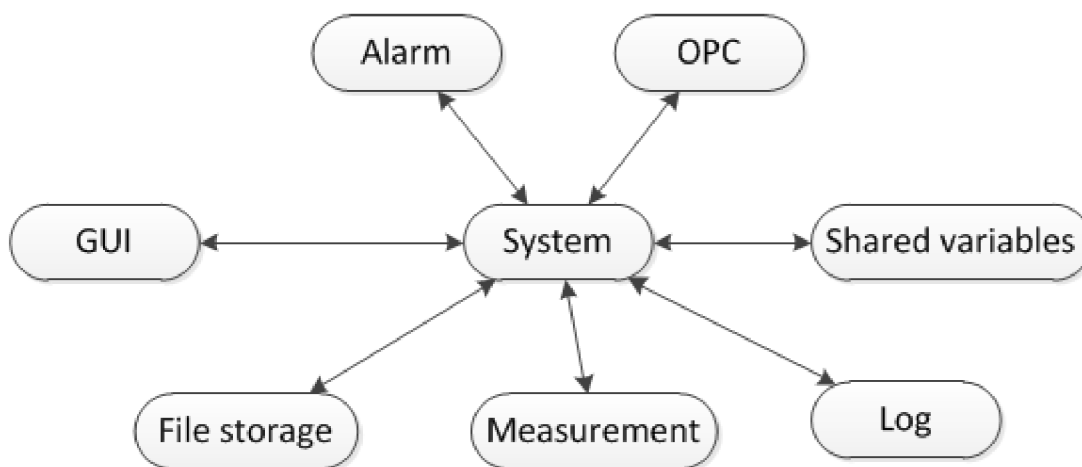
## Kapitola 3

# Implementace řešení

Zaměření této kapitoly bude pojednávat především o samotné programové implementaci řešení a jeho architekturní výstavbě.

### 3.1 Architektura vláknového zpracování

Cílem bylo vytvoření robustní a modulární architektury, která zajišťuje snadnou rozšířitelnost a upravitelnost pro přizpůsobení konkrétnímu nasazení v praxi. Náhled na stavbu této architektury umožňuje schématický obrázek 3.1. Při návrhu této architektury byla následována doporučení a ověřené postupy, které jsou součástí materiálů pokročilé tvorby systému v prostředí LabVIEW [14].



Obrázek 3.1: Architektura navrženého řešení

#### 3.1.1 Využití architektury

Současný trend navyšování výpočetní výkonnosti CPU nespočívá v navyšování kmitočtu procesorových jader CPU, s jakým jsou prováděny výpočty. Je to určeno několika faktory, zejména potom kubickým nárůstem příkonu takového systému [7]. Již tak vysoký ztrátový tepelný výkon vede ke komplikacím při řešení a realizaci chlazení. Nejenom z těchto důvodů je využito vlastnosti kompilátoru grafického návrhového systému LabVIEW, který implicitně sestavuje program pro běh ve vícevláknovém režimu. Paralelně vykonávané smyčky

jsou tedy prostředkem snadné a efektivní tvorby aplikací, které čerpají benefity paralelních architektur.

### 3.1.2 Fázová inicializace, deinicializace

Vzhledem k využití konfiguračních souborů je při spuštění Univerzální měřicí ústředny nutno provést kompletní inicializaci celého řešení. Konfigurovány jsou OPC proměnné, síťově sdílené proměnné, měřicí úlohy a další. Byl vytvořen princip fázové inicializace s dílčími kroky. Každá část měřicí stanice, která je právě konfigurovaná, okupuje vlastní fázi. Tyto fáze jsou navíc rozděleny do dílčích kroků. Příkladem je konfigurace měřících úloh. Nejdříve je zjištěn počet měřících úloh ke konfiguraci, poté jsou identifikovány typy měřících úloh a nakonec jsou vyčteny z konfiguračního souboru pouze ty parametry, které jsou adekvátní.

Při ukončování běhu Univerzální měřicí ústředny musí být zajištěna konzistence. Tím je míněno ukončení běhu měřících úloh, správné vystavení hodnot OPC proměnných, uvolnění alokovaných systémových prostředků a další. Správné vystavení hodnot OPC proměnných je nutností pro správnou koexistenci s řídicí PLC částí. Signalizace, že univerzální měřicí ústředna není připravena k provedení měření je zajištěna vystavením OPC proměnných do předem určené hodnoty.

### 3.1.3 Zpracování chyb

V průběhu činnosti modulů dochází ke vzniku chybových stavů. Tyto chyby jsou zapříčiněny chováním uživatele nebo selháním některého z funkčních bloků modulu. Výskyt chyby způsobené chováním uživatele je obecně častější. Mezi tento typ chyby jsou zahrnuty nekonzistence konfiguračního souboru, špatná posloupnost provádění úkonů. K tomuto účelu má *Systém* vyhrazen speciální stav pro zpracování očekávaných chybových stavů. Očekávané chybové stavy jsou například v místě spouštění měřicí úlohy, kdy je nutno počítat s eventualitou náhlého odpojení měřicí karty. Tento typ chyby je zpracován podle příslušnosti k modulu a zdroji chyby. Druhým typem chyb jsou tak zvané neočekávané chybové stavy, které vznikají například při zápisu zprávy do vstupní fronty *Systému*. Takové selhání je považováno za obecné selhání a po jeho signalizaci uživateli nejsou podniknuty další kroky. Jedná se o výjimečný stav vyžadující zásah operátora (případně vývojáře), protože některé ze základních součástí řešení selhávají.

### 3.1.4 Synchronizace a sdílení dat

Ruku v ruce s paralelním zpracováním algoritmů je spojena i jejich synchronizace, vzájemná komunikace dat a příkazů. Při výběru mezi základními prostředky pro synchronizaci bylo rozhodováno mezi využitím semaforů s přístupem ke sdílené paměti, notifikací nebo front. Alokace sdílené paměti a její správa za využití semaforů by mohla být vhodně využita pro malý počet vláken, kde přístupy do kritické sekce jsou v mezích. Vzhledem k modularitě, a nutnosti přípravy řešení pro rozšiřitelnost, však tento způsob nelze efektivně využít. Vlastností notifikací v prostředí LabVIEW je uchování pouze posledního příchozího oznámení, které je přepsáno v případě dostatečně rychlého nepřechtení si aktuální hodnoty. Notifikace jsou charakterizovány neblokujícím zápisem do fronty FIFO s hloubkou 1.

Všechny tyto poznatky vedly k rozvaze nad tím, že právě synchronizace skrze fronty zajistí potřebné požadavky kladené na řešení. Každý modul aplikace disponuje vlastní dedikovanou frontou předem daného datového typu (pro statickou alokaci paměťového pro-

storu). Do této fronty mohou vkládat data libovolné další moduly, což je omezeno logikou činnosti programu, kdy komunikace probíhá skrze jedno hlavní řídicí vlákno. Přímý přenos mezi vlákny (obdoba DMA z architektury počítačů) byl použit pouze ve zcela výjimečných a oprávněných případech. Nutnost nasazení přímé komunikace získává význam pouze v případě, kdy by intenzivní komunikace mezi dvojicí byla potenciálním omezením pro vlákna ostatní.

Ve zprávách jsou přenášena různá data a příkazy. Z tohoto důvodu je formát zpráv tvořen identifikací typu zprávy a datovým kontejnerem typu *variant*. Datový typ *variant* je obecným datovým typem, který může obsahovat libovolný i odvozený datový typ (např. řetězec, číslo, strukturu a další). Při zpracování příchozí zprávy jsou data ze zprávy nejdříve dekodována a to na základě typu příchozí zprávy.

V modulech jsou uplatněny dva způsoby využití front pro příjem příkazů a dat. Prvním z nich je blokující čtení, které bude popsáno v dalších odstavcích. Druhým způsobem je neblokující kontrola stavu zaplnění fronty. Při této kontrole je zjištěn celkový počet zpráv, které čekají na zpracování modulem. S výhodou je tak zajištěna dostatečná odezva modulu při výpočetně náročném úkonu. Popis využití tohoto způsobu práce s frontou je přítomen v odstavci 3.2.

### 3.1.5 Modularita a robustnost

Výsledné řešení je vzhledem ke své povaze, zamýšleného běhu na popředí cílového systému, rozšířeno o grafické uživatelské rozhraní. Aplikace se proto automaticky spouští ihned po startu systému. Ovládání aplikace je realizováno příkazy přes OPC server, dostupný konfigurační soubor nebo operátorem. Běh je tedy kontinuální bez potřeby jeho ukončení nebo restartování. Pro tento účel je třeba zajistit, aby žádné z možných chybových hlášení nevedlo k ukončení programu nebo k jeho nepředvídatelnému chování. Vznik chybových stavů musí být komunikován s řídicí částí přes OPC, aby bylo zajištěno rozšíření informace o vzniku chyby s cílem její nápravy. V případě dostupného plánu pro nápravu chybového stavu a obnovení správné funkčnosti musí být tento plán realizován s šířením zprávy o této aktivitě.

### 3.1.6 Architekturní šablona modulů

Každý z modulů následuje společnou architekturu výstavby modulů. Drobné odlišnosti jsou uplatněny u *Systémového* a *Měřícího* modulu. Důvodem je specializovaný princip fungování. Obecná architektura modulů sestává ze stavového automatu, kolekce lokálních proměnných, reference na fronty zpráv a počátečního stavu automatu. Lokální proměnné slouží k uchování konfigurací a referencí. Výhoda uchování konfigurací v patřičném modulu spočívá v usnadnění pozdější komunikace a snižuje nutné množství předávaných dat. K identifikaci konfigurace pak postačí pouze jednoduchý číselný identifikátor. Při zahájení měřící úlohy nebo připojení se k Shared Variable Engine vzniká reference k tomuto sezení. Přes referenci lze jednoduše k objektu přistupovat a zůstává zachován kontext provádění operací nad touto referencí. Stavový automat je rozdělen do dvou úrovní, kdy v nejvyšší úrovni jsou přítomny stavy pro *Idle*, *Zpracování požadavku (Process command)* a *Ukončení vlákna (Stop thread)*.

Ve stavu *Idle* modul nevykonává žádnou činnost a přechází do fáze „spánku“ v podobě blokujícího čtení z fronty. Dokud ve vstupní frontě modulu nejsou dostupná žádná data (příkaz nebo jiná data), tak modul zůstává v pasivním čekání. Výhodou této vlastnosti je minimalizace výpočetních nároků procesoru, což je obzvlášť výhodné při použití větší sady

modulů. Jakmile přijdou do vstupní fronty modulu data, tak přechází stavový automat do stavu *Zpracování požadavku*.

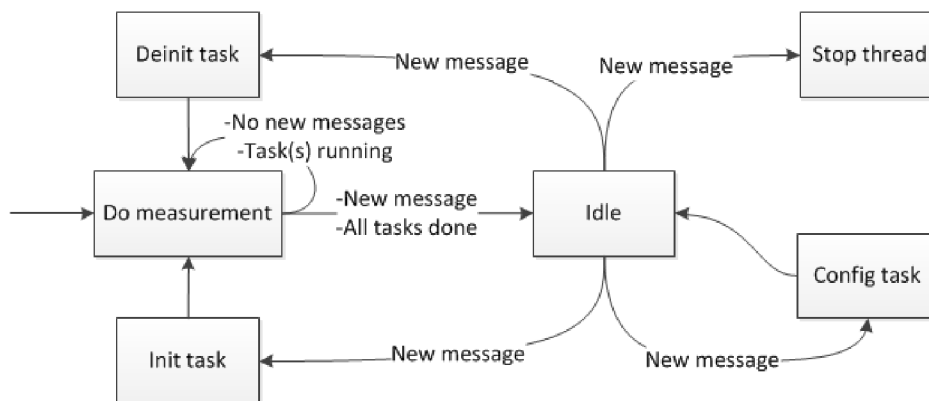
Při *Zpracování požadavku* dochází podle typu přijaté zprávy k vykonání posloupnosti úkonů. Po dokončení této posloupnosti úkonů a při nedostupnosti dalších dat ke zpracování, přechází stavový automat opět do stavu *Idle*. Některé moduly mají, za účelem pokročilejšího zpracování chyb, vyhrazený speciální stav. K tomuto stavu dojde zpracování modulu při vzniku chyby, která způsobí zaslání zprávy modulu sobě samému. Díky tomu lze z jediného místa obsloužit veškeré možné chybové stavy.

*Ukončení vlákna* obsahuje poslední úkony, které jsou modulem vykonány před jeho samotným ukončením. Všechny moduly jsou konstruovány tak, aby byla minimalizována jejich rozhodovací logika. Moduly vykonávají příkazy a hlásí stav provádění. Hlavní modul *Systém* řídí logiku a návaznost provádění úkonů. Z tohoto důvodu nejsou při *Ukončení vlákna* uvolňovány příkazy alokované zdroje. Tato činnost již musí být dokončena podle příkazů *Systému* před dokončením činnosti modulu. Je to z toho důvodu, aby na chybové stavy při uvolňování zdrojů mohl *Systém* patřičně reagovat.

## 3.2 Měřicí část

Hlavní součást řešení, která reprezentuje část programu určenou pro měření dat. Všechny další moduly zajišťují podpůrnou funkcionalitu právě tomuto modulu.

Specifikem *Měřicího modulu* je neblokující kontrola stavu zaplnění vstupní fronty. Měřicí úlohy mohou svou dobou trvání způsobit významné snížení odezvy modulu. Z toho důvodu je třeba reagovat na nově přichozí příkazy i za běhu měřicí úlohy. Pravidelné neblokující kontroly stavu zaplnění fronty zajišťují nízkou dobu odezvy. Jestliže ve vstupní frontě jsou data ke zpracování, pak jsou tato data zpracována po skončení aktuální iterace měření. Zobrazení stavového automatu *Měřicího modulu* lze vidět na obrázku 3.2. Se schopností



Obrázek 3.2: Stavový automat měřicího modulu

zpracovávat příkazy v průběhu činnosti měřicí úlohy souvisí její rozdělení do jednotlivých iterací. Každá z iterací sestává z dotazu na dokončení měřicí úlohy a vyčtení dostupných vzorků dat. Běžící úlohy prochází iteracemi měření pouze po dobu dostupnosti nových vzorků dat. V průběhu měření vede příchod nové zprávy k jejímu rychlému zpracování a dalšímu pokračování v měření. Jakmile jsou všechny vzorky vyčteny, pak je měřicí úloha dokončena. Výjimku tvoří měřicí úlohy kontinuálního běhu, jejichž ukončení je řízeno specifickým příkazem zastavení úlohy. Blokující čtení ze vstupní fronty je uplatněno ve chvíli,

kdy neběží žádná z měřících úloh.

### 3.2.1 Vzorkování signálu

Ke zpracování měřených signálů v PC je třeba, aby tyto signály byly digitalizovány, protože počítače jsou zařízení (diskrétní) pracující pouze s digitálními signály [11]. K tomuto účelu jsou měřené signály konvertovány prostřednictvím analogově-digitálních (A/D) převodníků. Ze série naměřených hodnot signálu není možno říci, s jakou vzorkovací frekvencí byly tyto hodnoty naměřeny. Pokud je vzorkovací frekvence příliš nízká, dochází k jevu nazývaném aliasing (podvzorkování signálu). Podle znění Nyquistova teorému musí být vzorkovací frekvence nastavena na hodnotu minimálně dvojnásobku maximální měřené frekvence podle vztahu 3.1.

$$f_{Nyquist} = \frac{1}{2}v \quad (3.1)$$

Nyquistova frekvence má hodnotu poloviny vzorkovací frekvence [12]. Signály s frekvencí vyšší než Nyquistova frekvence způsobí alias a objeví se ve frekvenční oblasti do hodnoty Nyquistovy frekvence. Zvýšením vzorkovací frekvence nad dvojnásobek maximální frekvence signálu je dosaženo lepšího zachování tvaru původního signálu.

### 3.2.2 Synchronizovaná vícekanálová měření

Častým cílem měření signálů ve složitějších soustavách je zjištění vzájemné vazby mezi signály. Toho lze dosáhnout synchronizovaným souběžným měřením signálů. Při pokusu o synchronizaci měření skrze současné spuštění dvou měřících úloh nelze nikdy dosáhnout tak přesného výsledku, jak při využití vlastnosti měřicí karty NI USB-6215. Tato karta dovoluje souběžné měření vícero vstupních kanálů v jednu chvíli. Interně je toho docíleno dvojicí hodinových signálů Sample clock a Convert clock. Jelikož je na měřícím zařízení NI USB-6215 k dispozici pouze jeden zesilovač signálu a jeden A/D převodník, je nutno tomu přizpůsobit fungování měřicí karty. Vzorkování signálů je řízeno Sample clock, který udává čas zahájení měření vzorku na všech kanálech. Řízení činnosti A/D převodníku a zesilovače signálu je zajištěno zmíněným Convert clock. Díky němu je zajištěno správné vzorkování signálů v rámci periody Sample clock.

## 3.3 Trigger

Vytvoření a příprava měřící úlohy není časově kritický úkon, který by vyžadoval striktní plnění v zadaný okamžik. Situace se však obrací v případě zahájení samotného měření. Čas, kdy je zahájeno vyčítání dat z měřicí karty a jejich ukládání do paměti PC, může být udán buď ručním zadáním příkazu nebo odpálením triggeru (firing the trigger).

### 3.3.1 Řízení běhu úloh

Pro řešení tohoto měřícího systému je třeba uvažovat tak zvaný triggering pro zahájení úloh, ale nikoliv již pro jejich ukončení. Dokončení měřící úlohy je dosaženo naměřením předem zadaného počtu prvků, které jsou navíc znamením úspěšného dokončení běhu úlohy.



### 3.3.2 SW trigger

Nejjednodušším způsobem spouštění měřících úloh je softwarový trigger, který vzhledem ke své povaze nutnosti softwarového zpracování nemůže dosahovat nejvyšší přesnosti. Součástí řešení je jediný typ softwarového triggeru, který je spouštěn v čase vyvolání. Jedná se tedy o spuštění úlohy přes příkaz pro zahájení měření.

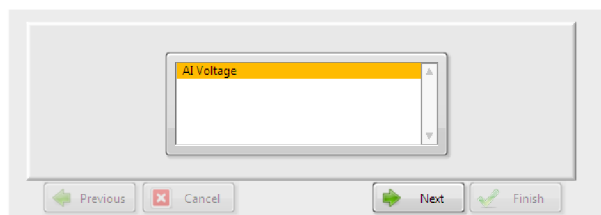
### 3.3.3 HW trigger

Nejpřesnější a nejrychlejší způsob spuštění úlohy je využití hardware triggeru. Obecný princip spočívá ve sledování hodnoty specifického vstupního signálu a na základě úrovně nebo datového vzoru, který se na signálu vyskytne, dojde ke spuštění měřící úlohy. Podpora triggování úloh se může vzájemně mezi jednotlivými zařízeními lišit. Pro řešení byla využita měřící karta NI USB-6215 a následující popis bude hardwarově závislý. Tato karta podporuje pouze digitální trigger, kdy jsou sledovány úrovně signálu, nebo datový vzor úrovní signálu.

Základní rozdělení hardwarových start a referenčních triggerů je na digitální a analogové trigger. Start trigger zahajuje měřící úlohu v čase výskytu. Referenční trigger měřená data ukládá do cyklického bufferu, který je postupně přepisován. Ve chvíli výskytu signálu sledované úrovně, je měřící úloha dokončena získáním zbylých hodnot. Konfigurace referenčního triggeru je určena před-měřenými (pre-trigger) a měřenými (post-trigger) hodnotami [12].

## 3.4 Konfigurace měřících úloh

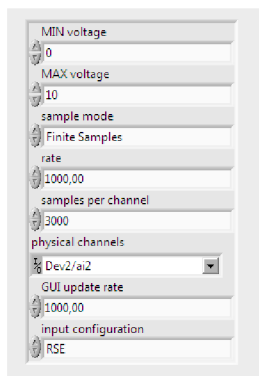
Při tvorbě měřící úlohy je potřeba nakonfigurovat parametry úlohy pro její správný běh. Úspěšné vytvoření konfigurace úlohy předchází jejímu spuštění a měření dat. Příkaz pro vytvoření nové prázdné měřící úlohy lze vyvolat stiskem tlačítka Add v hlavní nabídce řídicích tlačítek. Ihned se otevře jednoduchý průvodce vytvořením nové úlohy jako na obrázku 3.3. Nejdříve uživatel zvolí typ měřící úlohy. Dostupný typ měřících úloh bude představen v následujících odstavcích.



Obrázek 3.3: Průvodce vytvořením nové úlohy

### 3.4.1 Měření napětí

Nastavení parametrů (obrázek 3.4) úlohy měření napětí sestává z určení rozpětí měřeného napětí *MIN* a *MAX voltage*. Metoda vzorkování (*sample mode*) je implicitně nastavena na konečný počet měřených vzorků (*Finite Samples*). Dalšími možnými nastaveními jsou neomezené měření (*Continuous Samples*) a hardwarově řízené měření (*Hardware Time Single*



Obrázek 3.4: Konfigurace úlohy měření napětí

*Point*). Frekvence měření vzorků (*rate*) je udána v jednotkách Hertz. Počet vzorků na kanál (*samples per channel*) udává celkovou dobu běhu měřicí úlohy podle vztahu 3.2.

$$t = \frac{\text{samples\_per\_channel}}{\text{rate}} \quad (3.2)$$

Při neomezeném měření je počet měřených vzorků určen vzorkovací frekvencí a dobou měření, po kterou nechal operátor měření provádět.

Formát zápisu názvů kanálů je popsán v tabulce 3.1.

Kanály	Formát zápisu	Příklad
Jeden kanál	identifikace zařízení/identifikace kanálu	Dev2/ai3
Souvislá skupina kanálů	identifikace zařízení/počáteční identifikace kanálu:koncový index kanálu	Dev2/ai3:9
Nesouvislá skupina kanálů	identifikace zařízení/identifikace kanálu, identifikace zařízení/identifikace kanálu...	Dev2/ai3, Dev2/ai9

Tabulka 3.1: Formát zápisu fyzických kanálů

Fyzické kanály (*physical channels*) reprezentují měřené vstupní kanály. Může zde být doplněn pouze jediný samostatný kanál nebo celá skupina kanálů, vždy však v rámci jednoho fyzického měřícího zařízení.

Aktualizace uživatelského rozhraní (*GUI update rate*) souvisí s četností přenosu dat z měřicí úlohy do GUI a tím i rychlosti aktualizace náhledového grafu v průběhu měření.

Poslední parametr konfigurace zapojení vstupních kanálů (*input configuration*) vyjadřuje princip zapojení signálů do NI-PGIA [10]. NI-PGIA je diferenciální zesilovač signálu. Na základě zapojení jeho dvojice vstupů je rozlišována trojice konfigurací. Mezi základními možnostmi jsou RSE (Referenced Single-Ended Mode), kdy je signál měřen vůči AI GND kanálu. NRSE (Non-Referenced Single-Ended Mode) měří signál vůči AI SENSE kanálu. Nakonec volba *Differential* měří potenciál mezi dvojicí kanálů. Pro správné zapojení volby *Differential* je třeba vypočíst identifikační číslo druhého kanálu, které je získáno přičtením čísla 8 k identifikaci kanálu. V tabulce 3.2 jsou přehledně znázorněna pravidla pro zapojení vstupních kanálů. Významnou výhodou diferenciálního zapojení vstupních kanálů je fakt, že je měřen rozdíl potenciálů vzájemně mezi dvojicí vodičů plus (+) a minus (-). Tímto

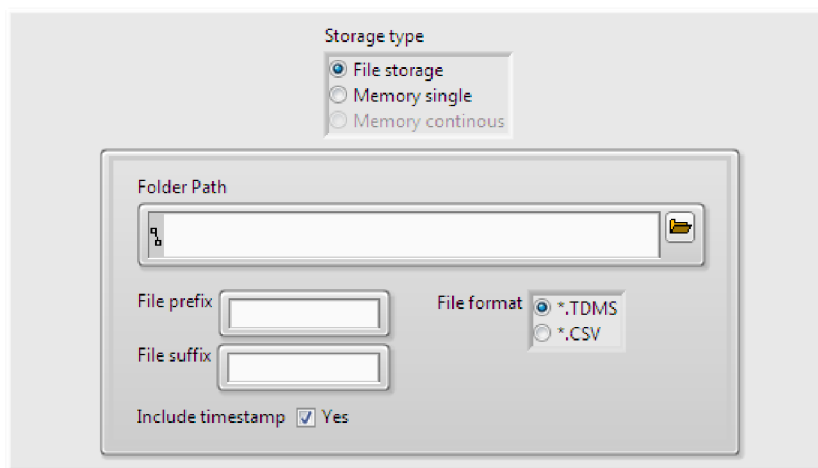
způsobem je odstraněno působení souhlasného napětí (common-mode voltages), které může být na vodičích přítomno například z důvodu rušení z vnějšího prostředí.

Konfigurace zapojení země	Pozitivní vstup NI-PGIA	Negativní vstup NI-PGIA
RSE	AI <0..31>	AI GND
NRSE	AI <0..31>	AI SENSE
DIFF	AI <0..7> AI <16..23>	AI <8..15> AI <24..31>

Tabulka 3.2: Zapojení signálů do NI-PGIA [10]

### 3.4.2 Uložení dat

Cílem pro uložení dat, viditelný na obrázku 3.5, může být operační paměť počítače (*Memory single*), kdy jsou data uchována pouze po celou dobu běhu programu. Alternativně lze data uložit do zadaného datového souboru. V tomto případě jsou data uchována jak v operační paměti počítače, pro rychlý přístup při procházení náhledů, tak i v souboru. Jelikož se předpokládá, že do souborů budou ukládána data z velkého počtu běhů měřících úloh, tak je specifikace názvu souboru parametrická. Všechny soubory, které vzniknou v rámci měřící úlohy, jsou ukládány do společné složky. Název souboru sestává ze tří volitelných částí. První z nich je prefix souboru. Druhou částí je časové razítko, jehož formát je následující *RRRRMMDDhhmmss*. Zápis 11.4.2012 a 12:54:11 by měl podobu *20120411125411*. Při vytváření názvu souboru je použit čas zahájení měření. Poslední a třetí částí názvu souboru je sufix (přípona). Podporovanými formáty pro uchování dat jsou již představený formát souboru TDMS a textový formát souboru CSV.



Obrázek 3.5: Konfigurace cíle měřených dat

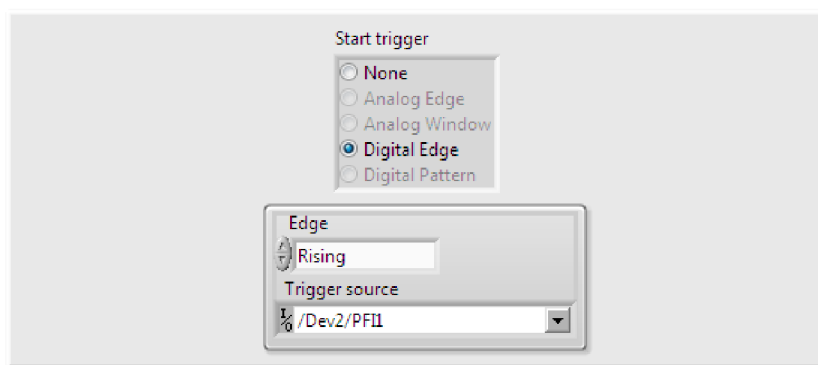
### 3.4.3 Konfigurace triggeru

Spuštění úlohy a její zahájení jsou vzájemně dva odlišné procesy. Za běžných okolností probíhají oba procesy v jeden čas, kdy je vzájemně nelze rozlišit. Situace se mění ve chvíli,

kdy je potřeba velmi přesně zahájit proces měření. Ke spuštění úlohy zpravidla dochází s předstihem a samotné zahájení je řízeno tak zvaným triggerem. Ke konfiguraci triggeru lze zvolit ze dvou možností (obrázek 3.6).

Start trigger *None* reprezentuje využití SW triggeru, kdy je úloha zahájena specifickým pokynem.









Hardwarový trigger *Digital Edge* je tvořen hranou (náběžnou nebo doběžnou), která je detekována na zadaném vstupním kanále. Konkrétní použitá měřicí karta využívá TTL logiku, kdy úrovně signálu jsou rozmístěny na intervalu 0V...5V.



Obrázek 3.6: Konfigurace triggeru pro zahájení měření

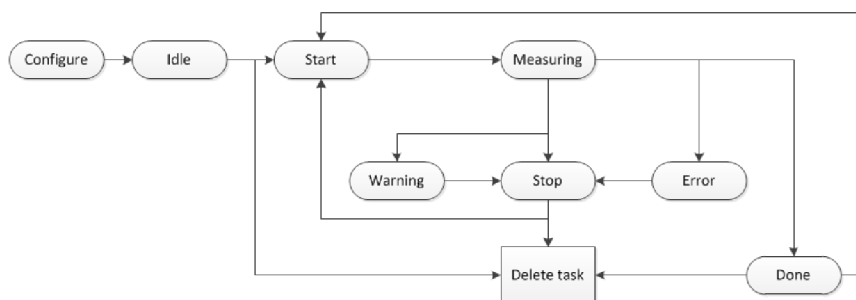
### 3.5 Stavby měřících úloh

Měřicí úloha od svého vzniku, konfigurace až po spuštění nebo zastavení uživatelem prochází stavy konečného automatu. Tyto stavy a jejich změny je možno sledovat ve *Správce úloh* grafického uživatelského rozhraní. Významy a přiřazené piktogramy jednotlivých stavů jsou představeny v tabulce 3.3.

Stav	Piktogram	Popis
Configure		Nová úloha byla vytvořena a právě probíhá její konfigurace.
Done		Nastalo úspěšné dokončení úlohy a naměření všech požadovaných vzorků. Tohoto stavu nemůže měřicí úloha dosáhnout, pokud je její běh nastaven jako kontinuální bez specifikace požadovaného počtu vzorků.
Error		Došlo k chybě v měřicí úloze, kterou nebylo možno napravit. Zásah operátora je vyžadován.
Idle		Úloha byla úspěšně vytvořena a čeká na své první spuštění.
Measuring		Probíhá měření. Úloze v tomto stavu není možné měnit konfiguraci ani ji odstranit.
Start		Zahájení provedení měření.
Stop		Vynucené zastavení měřicí úlohy. Toto je jediný způsob, jak se dá ukončit provádění měřicí úlohy s neomezeným počtem požadovaných vzorků (dobou měření).
Warning		Varovné hlášení informující o vzniklých potížích, které byly napraveny nebo nemají zásadní vliv na průběh měření.

Tabulka 3.3: Výčet stavů měřících úloh

Vzájemné závislosti mezi stavy úloh lze nejnázorněji vyjádřit prostřednictvím jednoduché posloupnosti propojených stavů. Na obrázku 3.7 jsou stavy reprezentovány oválnými tvary a děj odstranění úlohy je znázorněn obdélníkem. Odstranění úlohy může proběhnout pouze ve chvíli dokončené konfigurace úlohy, úspěšného doběhu úlohy nebo po vynuceném ukončení měřicí úlohy. Pro případ, že se úloha nachází v chybovém stavu, tak v něm setrvává do doby nápravy. Operátor musí zastavením úlohy potvrdit, že bere na vědomí zobrazené chybové hlášení a teprve poté může s úlohou dále pracovat. Může úlohu opětovně spustit, nebo ji odstranit.

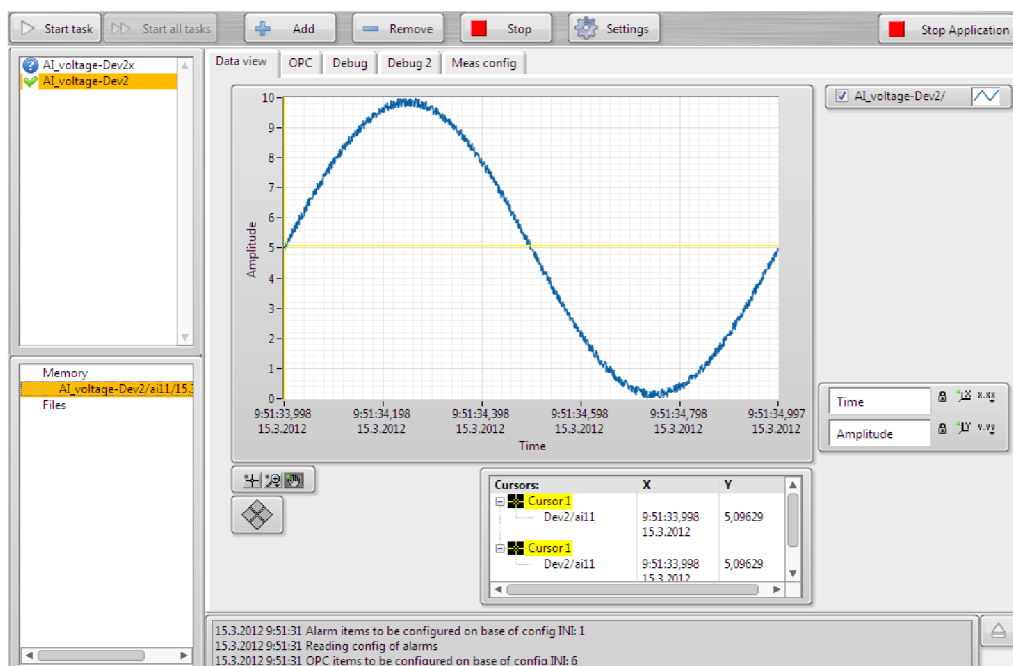


Obrázek 3.7: Přejchody mezi stavy úloh

## 3.6 GUI

Význam grafického uživatelského rozhraní narůstá v oblasti průmyslového využití. Přehledně zpracovaný systém hlášení dokáže zabránit vzniku škod v procesu výroby nebo nedodržení správných procesních postupů. Systémy s dotykovou obrazovkou, kdy uživatel po většinu

času používání programu nemá k dispozici poziční zařízení (např. myš), kladou zvýšený důraz na ergonomii celého systému. Tyto základní předpoklady byly brány v potaz při návrhu grafického uživatelského rozhraní systému. Na obrázku 3.8 je vyobrazen náhled zá-



Obrázek 3.8: Náhled GUI

kladní části GUI. Nyní bude tento náhled rozdělen do jeho jednotlivých částí a ty budou postupně představeny i s uvedením událostí, které nad nimi může uživatel provést.

### 3.6.1 Řídící panel

Správa měřících úloh se zajištěním jejich vytváření a odstraňování (obrázek 3.9). V neposlední řadě panel umožňuje spouštění jedné nebo všech měřících úloh. Ukončení úloh (pokud jsou spuštěny) slouží k zastavení měřících úloh s nekonečnou dobou běhu nebo předčasnému ukončení měřících úloh se specifikovanou dobou měření. Zastavení aplikace zajistí správnou dealokaci zdrojů a řádné ukončení aplikace. Úlohy ze seznamu vytvořených měřících úloh lze také odstraňovat. V tomto případě, ale nedochází k odstranění naměřených dat ze seznamu historicky provedených měření. Jestliže konfigurace úlohy nebyla provedena správně, a nebo se operátor chce přesvědčit o správnosti konfigurace, pak lze využít tlačítko *Settings*. Toto tlačítko zahájí proces konfigurace vybrané úlohy (ze seznamu vytvořených měřících úloh) a pokračuje od konfigurace zvoleného typu úlohy. V každém okně průvodce vytvořením měřící úlohy jsou již předvyplněna data podle již uložené konfigurace.



Obrázek 3.9: Řídící panel

### 3.6.2 Správa úloh

Seznam všech vytvořených úloh (obrázek 3.10) s indikací jejich aktuálního stavu. Změny tohoto seznamu nastávají s vytvářením nebo odstraňováním měřících úloh. Dále pak s aktivitou jednotlivých úloh. Operace nad měřícími úlohami jsou podmíněny řídicí logikou, která již byla popsána v této technické zprávě dříve.



Obrázek 3.10: Správce úloh

### 3.6.3 Historie měření

Soupis historických měření dat je na obrázku 3.11. Data jsou zde uspořádána podle cílové lokace naměřených dat. Data mohou být ukládána pouze do operační paměti nebo i do datového úložiště. Z popisu položky seznamu historie měření lze vyčíst všechny potřebné informace pro správnou identifikaci běhu měření. Formát položky je následující:

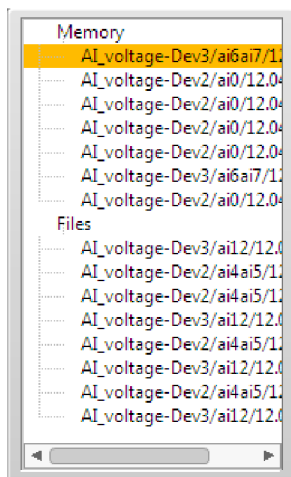
*typ měření-identifikace měřící karty/měřící kanál/datum a čas zahájení měření*

Ze seznamu měřených průběhů na obrázku 3.11 lze poznat, že měření byla prováděna na měřících kartách identifikovaných jako *Dev2* a *Dev3*. Měřenými kanály pro zařízení *Dev2* byly *ai0*, *ai4* a *ai5*. Pro zařízení *Dev3* byly použity měřící kanály *ai6*, *ai7* a *ai12*. Další informace o době zahájení měření jsou dostupné po posunu posuvníkem.

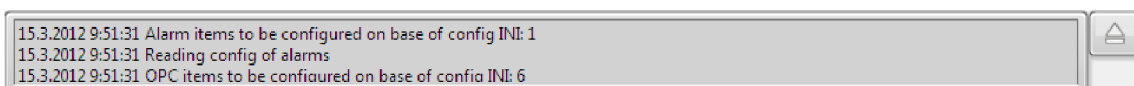
Jednoduchým poklepnutím na položku tohoto seznamu se v grafu, v náhledové části, zobrazí naměřená data. Pokud běží některá z měřících úloh, pak jsou její průběžná měřená data zobrazována v grafu. Historická data lze tedy zobrazovat pouze ve chvíli, kdy není aktivní žádná měřící úloha.

### 3.6.4 Stavový řádek

Ve stavovém řádku (obrázek 3.12) jsou vypisována veškerá hlášení programu, která jsou určena pro zobrazení v GUI. Počet hlášení se v závislosti na běžících měřících úlohách může zvyšovat. V některých případech při periodických dějích by mohlo dojít k záplavě chybových hlášení. Dobrým příkladem je periodické vyčítání OPC proměnných. Při intervalu 0,5s by pak operátor neměl možnost zachytit jakákoliv jiná hlášení v záplavě chybových hlášení způsobeným vyčítáním z OPC. Proto je omezen maximální počet opakování stejného hlášení na dvě zprávy. Při opakovaném hlášení je operátor upozorněn, že další hlášení stejného typu budou blokována. Pro potřeby čtení delších hlášení lze stavový řádek, prostřednictvím expanzního tlačítka, jednoduše expandovat do víceřádkového zobrazení.



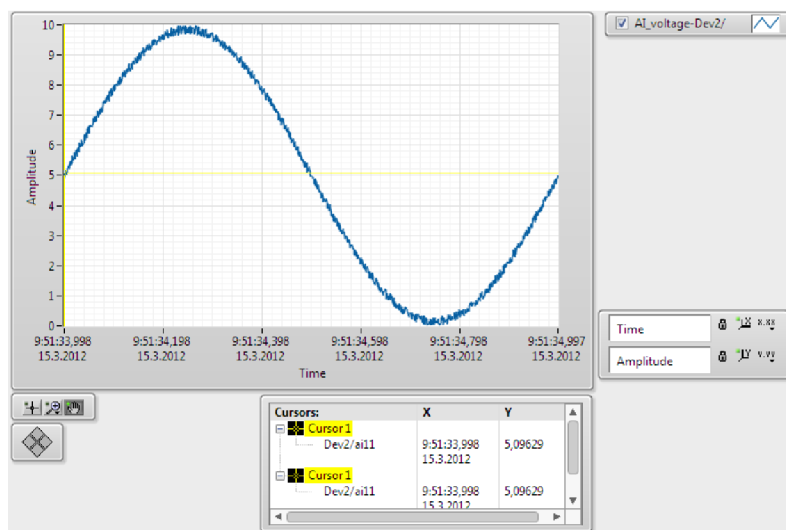
Obrázek 3.11: Historie měření



Obrázek 3.12: Stavový řádek

### 3.6.5 Náhledový prostor

Hlavní část zobrazovaného okna, která slouží pro interakci s uživatelem (obrázek 3.13). V této části okna mohou být zobrazeny průběhy aktuálně měřených signálů, historicky naměřených dat. Při vytváření nebo úpravě konfigurace měřící úlohy jsou zde zobrazovány konfigurované parametry, které může operátor měnit. Posloupnost konfiguračních obrazovek při vytváření měřící úlohy byla popsána již v předchozím úseku.



Obrázek 3.13: Náhledový prostor

Cílové zařízení disponuje rozlišením 1024x768, které dovoluje zobrazení dostatečného množství obrazových bodů pro potřeby řešení. Je však navíc nutné zvážit omezení v podobě



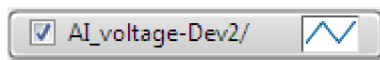
dotykové obrazovky. Tyto podmínky jsou vzájemně protichůdné, když dotyková obrazovka vyžaduje maximální možnou plochu ovládacích prvků pro ergonomii práce, ale omezené rozlišení obrazovky vytváří pevně dané meze.

Nyní budou detailněji popsány jednotlivé ovládací a zobrazovací prvky náhledového okna s představením jejich funkcionality.

### Přehled zobrazených průběhů

K identifikaci zobrazených průběhů měřených dat slouží chlívček (na obrázku 3.14), který je přehledně odlišuje. Nejenom, že jsou zde umístěny všechny v grafu zobrazené kanály, ale je možno jejich zobrazení řídit. Zatrhávací políčko určuje, jestli je měřených průběh zobrazen, nebo ne. Pro zvýšení kontrastu nebo pro lepší odlišení průběhů lze měnit jejich barvu a styl. Po klepnutí nad vlnovkou průběhu se rozbalí kontextová nabídka.

Z kontextové nabídky lze zvolit způsob zobrazení průběhu v bodech, spojitě čáře, vyplněné ploše, čarách. Měnitelnými vlastnostmi zobrazených čar jsou tloušťka, barva a styl. Dále je přístupná změna tvaru vykresleného bodu v grafu a způsob interpolace mezi body. V neposlední řadě velmi významnou volbou je možnost uložení zvoleného průběhu a to buď ve formátu dokumentu Excel, systémové schránky nebo do programu DIAdem.

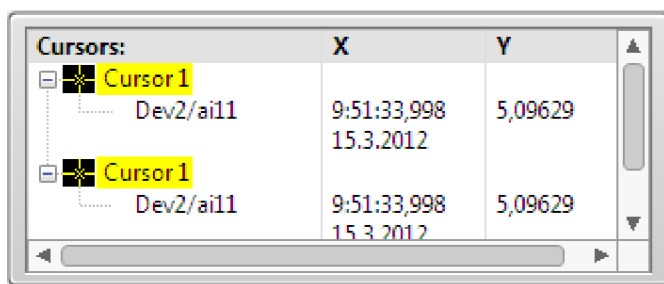


Obrázek 3.14: Přehled naměřených úloh

### Kurzory grafu

Významným pomocníkem pro analýzy nad měřenými daty jsou kurzory, které zastávají funkci ukazovátko. Kurzor v jakékoliv pozici zobrazuje svoje aktuální souřadnice (obrázek 3.15). Počet kurzorů v grafu je zcela dynamický a záleží pouze na uživateli, kolik si jich v grafu navolí. Změna vlastností kurzorů je vyvolána pravým tlačítkem kurzoru nad příslušným kurzorem. Mezi základními vlastnostmi kurzoru grafu jsou šířka a styl zobrazované čáry. Dalšími vlastnostmi pak jsou barva kurzoru grafu a tvar ukazovátko.

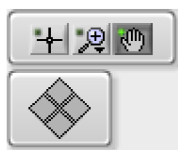
Pokud se kurzor grafu nachází mimo zobrazovanou část grafu, pak lze s výhodou využít příkaz pro vycentrování pozice kurzoru grafu. Výhodnou vlastností u kurzoru grafu typu *Single-plot* je svázání s jedním zobrazeným průběhem. Při tomto svázání pak pohyby kurzoru grafu kopírují měřený průběh a lze tak přesně procházet měřený průběh bez nutnosti ručního centrování.



Obrázek 3.15: Kurzory grafu

## Sada ovladačů grafu

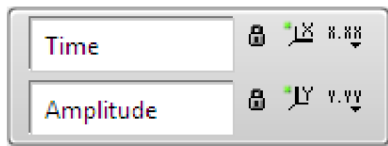
K ovládní způsobu zobrazení dat v grafu slouží paletka nástrojů z obrázku 3.16. Tyto nástroje budou popsány ve směru zleva doprava, od horu dolů. Prvním nástrojem je posun aktuálně zvoleného kurzoru po grafu. Po zvolení tohoto nástroje lze kurzorem grafu pohybovat buď tažením za jednu z jeho os, které tvoří pomyslný kříž. Dalším způsobem změny pozice v grafu je tažením samotného středu ukazovátko grafu. Druhým nástrojem je lupa (selektivní výběr). Kromě základních možností přiblížení a oddálení zobrazeného grafu lze selektivně vybírat i menší podčást grafu. Tento selektivní výběr je dvojího typu. Jedním typem se výběr omezuje na úsek na ose. Druhým typem je výběr volný zcela v ruce operátora. Pro jednoduché zrušení provedených výběrů lze jednoduše vrátit celý měřený průběh do zobrazované plochy grafu. Třetím nástrojem je ruka, s jejíž pomocí lze volně posouvat zobrazeným průběhem všemi směry. Čtyřsměrný kříž je navigační nástroj pro kurzory grafu a jeho ovládním je pohybováno aktuálně zvoleným kurzorem grafu.



Obrázek 3.16: Ovladače grafu

## Měřítka grafu

Pojmenování os a jejich konfigurace je řízena v boxu na obrázku 3.17. Při každé ose jí její pojmenování jednoznačně identifikuje. Tlačítko zámku přepíná automatické škálování osy. Škálování osy zajišťuje, že celý interval hodnot v příslušné ose bude v grafu zobrazen. Operátor tak získává přehled o celém měřeném intervalu. Jestliže je provedeno přiblížení nad podčást grafu, pak druhé tlačítko provede vynucené škálování osy a zobrazení celého intervalu měřených dat v grafu. V osách lze zobrazit kromě času i amplitudy měřených hodnot. Z tohoto důvodu lze posledním tlačítkem měnit formát zobrazených dat. K dispozici je např. oktálové, desítkové a hexadecimální zobrazení číselných údajů. Pro zobrazení časových údajů lze vybrat absolutní nebo relativní časování. Při zobrazení desetinných čísel lze volitelně měnit přesnost na počet desetinných míst. Lineární a logaritmické mapování pak představuje příjemný bonus v rozšíření dostupné funkcionality.

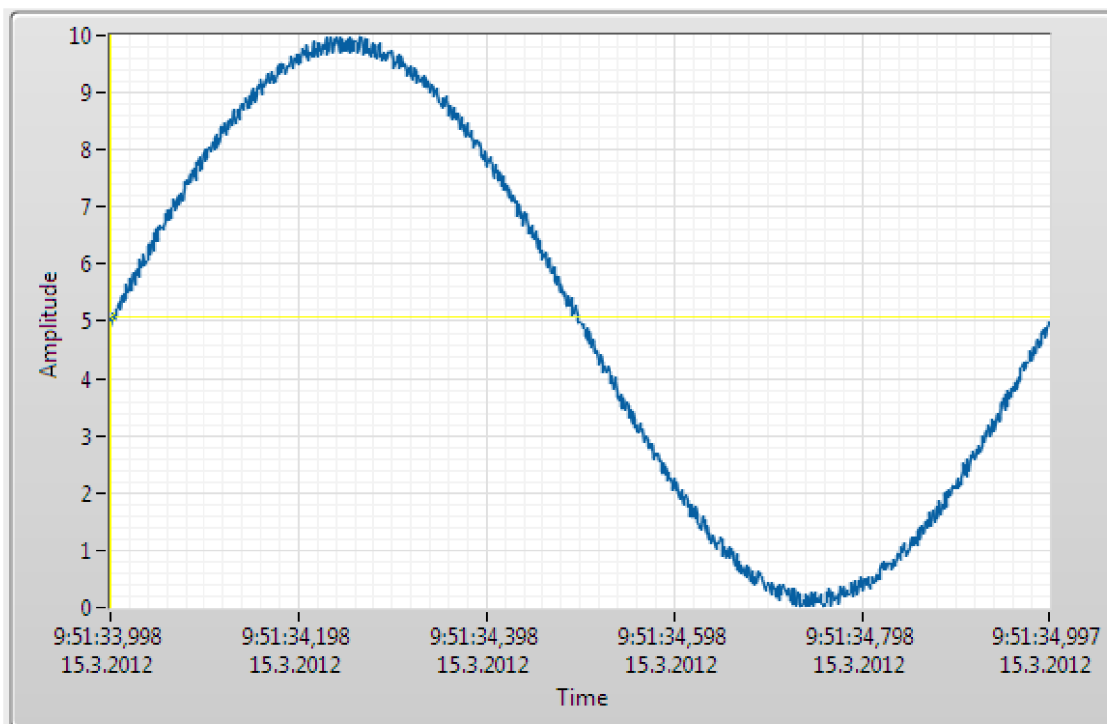


Obrázek 3.17: Konfigurace os grafu

## Graf

Nejvýznamnější částí náhledového zobrazení měřených průběhů je jejich grafické zobrazení na obrázku 3.18. Možnosti konfigurace zobrazování dat již byly popsány v předchozích kapitolách. Nyní již bude pouze zmíněna vlastnost exportu zobrazených dat. Kromě exportu

do systémové schránky, DIAdemu a Excelu lze navíc data exportovat do obrázkového formátu souboru. K dispozici jsou formáty souboru .bmp, .eps a .emf. Kontextová nabídka pro export zobrazených dat je přístupná po stisku pravého tlačítka kurzoru nad oblastí grafu.



Obrázek 3.18: Grafické zobrazení měřených průběhů

### 3.6.6 Přehlednost a přívětivost ovládání

Ke zkrácení času zaučení operátora a zajištění dobrého předpokladu pro přijetí při nasazení řešení byl kladen důraz na přehlednost systému [8]. V uživatelském rozhraní systému převládají prvky náhledové nad řídicími, kdy dominantou zobrazení je okno, kde jsou zobrazovány grafy měřených nebo vypočtených průběhů. Veškeré ovládací prvky jsou dostatečně rozměrově dimenzovány, aby vyhověly prstům lidské ruky.

Část funkcionality je skryta do událostí provedených nad komponentami uživatelského rozhraní. Otevření náhledu měřených dat se provádí poklepnutím na položku v seznamu historie měření (nebo výpočtu).

### 3.6.7 Optimalizace pro výkon

Vícevláknová architektura je využita i pro oddělení grafického uživatelského rozhraní od zbytku systému. Pro potřeby zajištění maximální možné rychlosti odezvy GUI jsou veškeré prováděné výpočty nad daty odsunuty do řídicího bloku (*Systemu*). Jakákoliv událost, kterou uživatel vyvolá, je okamžitě odeslána. Tím veškerá činnost pro grafické rozhraní skončila. Pokud ovšem *System* nezašle požadavek na provedení změny v GUI. Tento způsob dovoluje zajištění maximální možné odezvy a odstínění doby zpracování příkazu. Zobrazovaná data a změny prováděné na obrazovce jsou vždy řízeny řídicím blokem (*Systemem*).

## 3.7 Systém

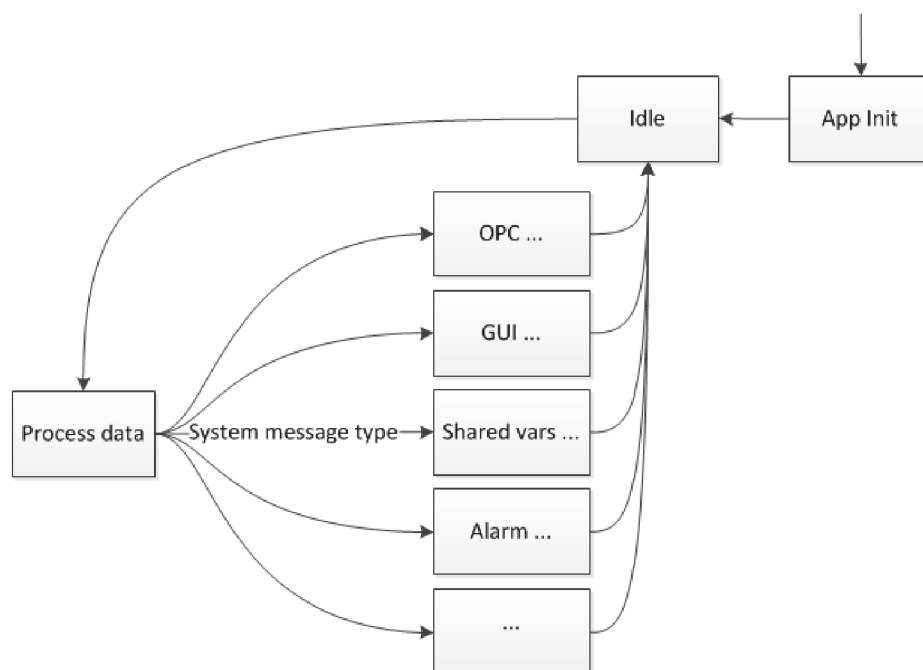
Architektura celého řešení je postavena na principu modularity a vzájemného oddělení jednotlivých výpočetních celků (vláken), které společně vytvářejí jeden celek.

### 3.7.1 Synchronizace úloh

K řízení řešení slouží centrální vlákno zvané *Systém*, které je středobodem pro řízení systému. Veškeré příkazy jsou zasílány právě tímto vláknem a odezva nebo hlášení o stavu vláken jsou opět zasílány tomuto hlavnímu vláknem. Cílem této architektury byla možnost plného oddělení jednotlivých výpočetních vláken a jejich izolace od zbytku řešení. *Systém* se účastní na rozhodovací logice, což zvyšuje jeho výpočetní náročnost, při řešení hlášení chybových stavů.

Po spuštění programu *Systém* rozlašuje inicializační příkazy pro přípravu GUI a počáteční konfiguraci vláken, které to vyžadují. Díky tomu lze jednoduše měnit chování programu změnou kódu soustředěnou v jediném místě, bez potřeby znalosti výstavby a fungování dalších modulů.

Při ukončování programu jsou všechna vlákna informována správnou posloupností příkazů vedoucích k uzavření otevřených datových souborů, ukončení probíhajících měření a dalších nezbytně nutných úkonů. Posledním ukončovaným vláknem aplikace je vždy *Systém*. Stejně jako platí pro ostatní vlákna i *Systém* přijímá příkazy přes vstupní frontu zpráv.



Obrázek 3.19: Stavový automat Systémového modulu

Pokud nejsou aktivně vykonávány úkony *Systému* a nepřichází do vstupní fronty žádná data, pak je *Systém* blokován do doby příchodu nové zprávy do fronty zpráv. Strukturu stavového automatu *Systému* lze vidět na obrázku 3.19. Tento princip chování, využitý i pro ostatní vlákna, dovoluje i při vysokém počtu modulů (vláken) v systému zachovat minimální výpočetní náročnost při neaktivitě.

### 3.7.2 Logika řízení, zpracování událostí

Některé kombinace příkazů zadávané uživatelem v GUI nemohou být provedeny z důvodu porušení logiky programu. Za tímto účelem je veškerá logika GUI soustředěna právě v *Systému*, který se stará o zachování konzistence provádění měřících úloh a celého programu. Dobrým příkladem je pokus o spuštění úlohy, která je právě konfigurována a tedy kompletně nevytvořena. Její spuštění by mohlo vést k běhu s nekompletní konfigurací a v krajním případě až k poškození měřícího zařízení. K zabránění tomuto stavu slouží právě *Systém*.

## 3.8 Obslužné části programu

Tyto části slouží jako doplněk k hlavním modulům řešení, ale jejich příspěvek má nemalý dopad na úroveň komfortní a profesionální aplikace.

### 3.8.1 OPC

Po spuštění programu jsou z konfiguračního souboru načtena nastavení pro všechny OPC proměnné, které budou v průběhu činnosti programu využity. Načítány jsou konfigurace pro proměnné určené ke čtení i zápisu. Jakmile jsou nastavení úspěšně načtena, tak dochází k jejich uplatnění a vytvoření spojení s OPC serverem. Každá proměnná je specifikována URL, typem proměnné a timeouty pro čtení/zápis a připojení k serveru. Jakmile jsou relace spojení se serverem vytvořeny, tak lze proměnné dále používat. Při každé čtecí/zápisové operaci je využito znalosti datového typu proměnné. Čtení hodnoty z proměnné je omezeno maximálním časem (timeoutem), který může tato operace vyžadovat. Ve výchozím nastavení je vyčkáváno po celý čas timeout pro změnu hodnoty oproti předešlé vyčítané hodnotě. Tímto způsobem lze zajistit maximální reaktivnost na změnu hodnoty. Nevýhodou je, že čekání na vyšší počet proměnných může způsobit postupné zaplňování fronty příkazů k provedení. Z toho důvodu nebylo nastavení využito a při požadavku na vyčtení hodnoty je okamžitě vrácena právě dostupná hodnota.

### 3.8.2 Shared Variable

Modul pro práci se sdílenými proměnnými je velmi podobný dříve představenému modulu pro práci s OPC proměnnými. Základním rozdílem je především práce se sdílenými proměnnými obecně. Využití tohoto modulu pro práci pouze se síťově sdílenými proměnnými tedy pokrývá pouze podmnožinu nabízené funkčnosti. Proměnné pro tento modul jsou konfigurovány v konfiguračním souboru .ini a formát konfigurace bude představen v dalším odstavci.

### 3.8.3 Konfigurační soubor

Pro uchování nastavení programu i při restartu panelového PC je nutno tato data umístit na datové úložiště. Ve snaze zjednodušit vzájemnou komunikaci s řídicí PLC částí je počáteční konfigurace programu umístěna do konfiguračního souboru. Tento soubor obsahuje základní nastavení měřících úloh. Jeho vyčítání probíhá okamžitě při spuštění programu ještě v čase, kdy operátor nemůže do programu nijak zasahovat. Vytváří tak jakousi sadu předepsaných úloh, které budou v měřící aplikaci po jejím startu vždy obsaženy. Díky tomu je při zadání příkazu o vykonání měřící úlohy s daným identifikátorem úlohy vždy vykonána dopředu

známá měřicí úloha. Obsah konfiguračního souboru lze rozdělit do několika částí, které se zaměřují na specifické nastavení.

Obecná nastavení sekce *header* udávají počty měřících úloh, alarmů a OPC proměnných, které budou na základě konfiguračního souboru nastaveny. Doplňující informací je pak jméno operátora, které umožňuje vytvořit a vzájemně odlišit sady konfigurací pro různé osoby nebo případy užití. Poslední položkou této sekce je verze programu, kdy dodržení správné interpretace konfigurací je zajištěno pouze pro odpovídající si verze programu.

Pro konfiguraci měřících úloh slouží vzájemně oddělené sekce, kdy každá z těchto sekcí představuje konfiguraci jedné měřicí úlohy. Názvy hodnot některých vlastností jsou více-slovné a hrozí tak výskyt chybného hláskování. Z toho důvodu jsou hodnoty těchto vlastností vyjádřeny číselně. Název těchto vlastností a jejich hodnoty jsou uvedeny v tabulce 3.4. Názvy ostatních vlastností jsou samopopisné a proto jejich výčet nebude v této technické dokumentaci uveden.

Parametr	Hodnoty
task_type	AI Voltage ... 3
sample_mode	Finite samples ... 10178
	Hardware timed single point ... 12522
	Continous samples ... 10123
start_trigger	None ... 0
	Digital Edge ... 3
start_trigger_edge	Rising ... 10280
	Falling ... 10171
reference_trigger	None ... 0
input_config	default ... -1
	RSE ... 10083
	NRSE ... 10078
	Differential ... 10106
	Pseudodifferential ... 12529

Tabulka 3.4: Parametry úlohy měření

OPC proměnné jsou definovány v sekci speciálně určené pro tyto položky. Vlastnosti, které jsou určeny číselnou hodnotou jsou v tabulce 3.5.

Parametr	Hodnoty
connection_mode	Read ... 0
	Write ... 1
	ReadWrite ... 2
	BufferedRead ... 3
	BufferedReadWrite ... 4
data_type	DBL ... 0
	Variant ... 1
	U32 ... 2
	I32 ... 3
	STR ... 4

Tabulka 3.5: Parametry OPC proměnných

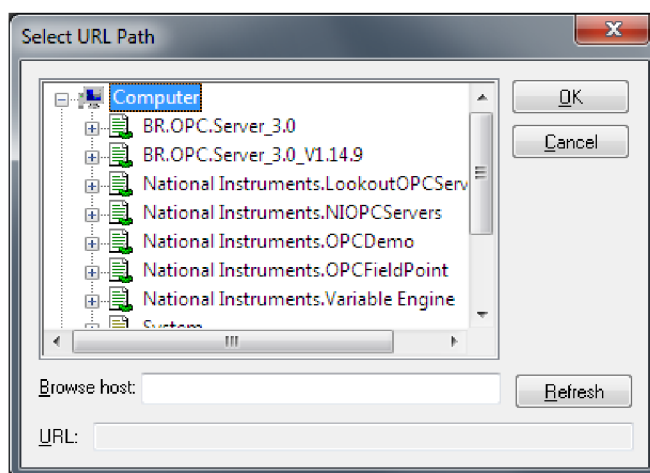
URL adresa OPC proměnné následuje standardizovaný formát. Tento formát jednoznačně udává použitý protokol pro dosažení OPC proměnné, lokalizaci proměnné a její jednoznačnou identifikaci v rámci cíle. Formát URL adresy proměnné je následující:

*opc://adresa-serveru/identifikace serveru/identifikace OPC proměnné*

Příkladem je proměnná s URL adresou:

*opc://localhost/NI.NIOPCServers/Channel1.Device1.CmdMeasTaskUsintOdBaR1*

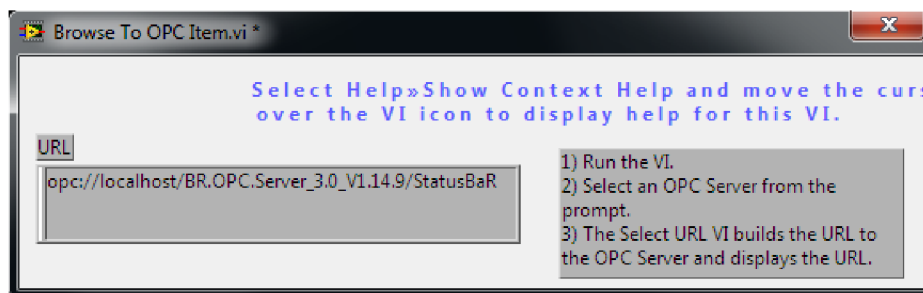
Nejjednodušší cestou k prozkoumání všech dostupných OPC proměnných na lokálním nebo vzdáleném počítači je jednoduchá aplikace, která je součástí balíku příkladů dodávaných ke grafickému návrhovému prostředí LabVIEW. Tato aplikace s názvem *Browse To OPC Item* po svém spuštění vyvolá dialogové okno z obrázku 3.20. Jakmile uživatel provede



Obrázek 3.20: Procházení dostupnými OPC proměnnými

výběr, který potvrdí stiskem tlačítka OK, tak se mu v okně aplikace zobrazí kompletní URL OPC proměnné. Výsledek činnosti podpůrné mini aplikace lze vidět na obrázku 3.21. Toto URL lze přímo použít k identifikaci OPC proměnné v konfiguračním souboru.

Sekce alarmů je tvořena pouze jednou vlastností, kterou je třeba představit. Výčet možných hodnot je v tabulce 3.6.



Obrázek 3.21: Výsledné URL OPC proměnné

Parametr	Hodnoty
alarm_type	0 ... Periodic
	1 ... Single

Tabulka 3.6: Parametry alarmů

Konfigurace síťově sdílených proměnných (Network shared variable) je řešena samostatnou sekcí konfiguračního souboru. Názvy parametrů s výčtem konkrétních hodnot, které mohou nabývat jsou uvedeny v tabulce 3.7. Podobně jako u konfigurace OPC proměnných následují položky sdílených proměnných standardizovaný formát URL. Obecný syntax URL je:

*[Variable Engine]://[Host Name]/[Container Name]/[Variable Name]*

Network-Published Shared Variable používají PSP Variable Engine (*ni.var.psp*). *Host Name* reprezentuje cílové zařízení jako identifikaci cíle, DNS jméno nebo IP adresu. *Container Name* slouží k rozlišení knihoven proměnných. Poslední *Variable Name* specifikuje samotné jméno proměnné, které musí být v rámci kontejneru unikátní [15]. Příklad URL jedné z použitých síťových proměnných je:

*ni.var.psp://localhost/shared\_vars\_library/status*

Parametr	Hodnoty
read_access	0 ... Allowed (potenciální přístup pro čtení)
	1 ... Required (vyžaduje přístup pro čtení)
	2 ... None (nevyžaduje přístup pro čtení)
write_access	Nabývá stejných hodnot jako parametr read_access pro operaci zápisu
data_type	Využívá stejných hodnot konstant jako u konfigurace OPC proměnných

Tabulka 3.7: Parametry Network Shared Variable



### 3.8.4 Alarm

Prostředí průmyslové automatizace a řízení procesů vyžaduje již ze své podstaty kontrolu nad pravidelně se opakujícími ději. Existuje řada aplikací, kde je třeba časování provádění aktivit. Mezi nimi jsou například provádění periodického vyčítání řídicích proměnných nebo periodické spouštění měřicí úlohy. Mezi aperiodickými ději jsou opožděný zápis na disk nebo odezva na řídicí proměnnou se zpožděním.

#### Periodické a aperiodické děje

Všechny zde zmíněné příklady lze jednoduše vyřešit pomocí časovače, který však představuje nutnou výpočetní režii a degradaci výkonnosti celého systému. Motivací pro zlepšení výkonnosti bylo vytvoření samostatného vlákna, které všechna časování systému zaštiťuje.

Fungování tohoto modulu s výhodou využívá vlastnosti funkčního bloku pro vyčítání dat ze vstupní fronty modulu. Touto vlastností je čekání, s nastavením hodnoty timeout, pro příchod nových dat. V průběhu časování některého z dějů není nutné s krátkou periodou kontrolovat jejich stav. Z toho důvodu je výchozí hodnota pro kontrolu stavu všech časovačů zvoleno  $1000ms$ . Ze všech čekajících alarmů (společně s konstantou  $1000ms$ ) je vybrána nejnižší hodnota doby časování, která je využita jako timeout funkčního bloku pro kontrolu nově přichozících dat. Tímto způsobem je plně zajištěno, že nebudou ztracena žádná přichozící data pro tento modul. Zároveň zůstává zachována nízká odezva modulu *Alarm* a zaslání informace o doběhnutí alarmu *Systému* proběhne s dostatečnou přesností.

Ve výsledku je informováno centrální vlákno (*Systém*) o vzniku události doběhu časovače. Úkolem *Systému* je správná reakce na vznik této události. Například odesláním zprávy nebo na základě identifikace časovače jeho přeposlání vláknu, které si o něj zažádalo.

### 3.8.5 Logování

Zaměření systému pro kontinuální běh na průmyslových PC vede k potřebě centralizovaného schraňování hlášení o chybách nebo varovných stavech. Jelikož nelze zajistit, že v celém průběhu běhu programu bude zpracování hlášení systému obsluhováno operátorem, pak je třeba tato hlášení zpracovávat centralizovaně a automatizovaně.

K tomuto slouží vlákno pro logování činnosti a hlášení systému. Hlavním přínosem tohoto modulu je odlehčení všech dalších modulů od nutnosti řešení práce s vypisováním textů na obrazovku.

## 3.9 Měření historická data

### 3.9.1 Načtení historických dat

Průběh měření se podle provedené konfigurace ukládá pouze do operační paměti nebo i do datového souboru uloženého v datovém úložišti. Při spouštění a úspěšném dokončení měření vzniká historie měřených dat, odkud lze jednoduše a rychle přistoupit k prohlížení těchto dat. Činnost programu má být kontinuální a tedy v historii měřené průběhy budou v tomto soupisu obsaženy.

#### Využití CSV

Formát souboru CSV je jednoduchý textový formát, který je využíván především z důvodu snadného náhledu na uložená data. Nevýhodou použití tohoto formátu souboru je, že ne-

nese žádné doplňující informace k obsaženým měřeným datům. Není zde identifikace názvu měřicího zařízení, ani identifikace měřících kanálů. Na uživateli tedy zůstává, aby zajistil správnou interpretaci obsahu tohoto souboru. Data obsažená v souboru CSV následují formát:

*DD.MM.RRRR hh:mm:ss,sssss;1.kanál,2.kanál;...*

Pro názornost je uveden příklad obsahu takového souboru. První řádek obsahuje:

*12.4.2012 12:30:16,188955;4,918973E+0;5,444807E+0*

Měřicí úloha byla zahájena 12.4.2012 v čase 12:30:16 a prvním vzorkem kanálu č.1 je hodnota 4,918973E+0 a kanálu č.2 je 5,444807E+0.

### Využití TDMS

Datové soubory mohou následovat formát TDMS, který byl popsán v dřívější kapitole této technické zprávy. Hlavní příčinou pro využití tohoto formátu jsou již zmíněné výhody a navíc předpřipravená struktura, která jedinečně reflektuje charakter ukládaných dat.

Každý běh měření je doplněn o vlastnosti (*properties*) samotného průběhu měření. Výčet těchto vlastností je shrnut v tabulce 3.8.

Parametr	Popis
Autor	Osoba zodpovědná za provedené měření. V případě ručního ovládání za použití GUI se jedná o operátora (fyzickou osobu). Řízené spuštění úlohy, prokomunikované skrze OPC, je identifikováno speciálním řetězcem.
Zahájení	Datum a čas zahájení měření. Tyto údaje jsou dostupnou součástí naměřených dat. Dostupnost tohoto údaje již v rámci parametrů odstraňuje nutnost přistupovat do vzorků měřených dat.
Verze programu	Jelikož datové soubory s obsahem měřených dat mohou být v průběhu času vytvořeny několika různými verzemi programu, pak je potřeba toto razítko o verzi datového souboru uchovat. Účel pro uchování této informace je zpětná podpora datových souborů vytvořených staršími verzemi programu.
Typ měřicí úlohy	Udává specifické informace o měřené veličině.
Jednotka osy X	Upřesnění jednotky na ose X.
Jednotka osy Y	Upřesnění jednotky na ose Y.

Tabulka 3.8: Parametry měřeného průběhu

### 3.10 Vzdálený dohled

Pro využití vzdáleného dohledu nad měřicí aplikací je využit tablet Apple iPad 2 ve spojení se síťově sdílenými proměnnými (Network shared variables). Tyto proměnné jsou hostovány na dotykovém panelovém PC pomocí SVE (Shared Variable Engine). Nejdříve je tedy třeba na cílové stanici vytvořit požadované proměnné. Poté nastavit aplikaci Data Dashboard s konfigurací připojení k hostitelské stanici.

### 3.10.1 Knihovna sdílených proměnných

Součástí projektu řešení je knihovna sdílených proměnných, která obsahuje požadované proměnné pro sledování průběhu činnosti programu. Těmito sdílenými proměnnými jsou *status* (textová proměnná), která v textové podobě informuje o stavu měřících úloh. Informováno je pouze o úlohách, které jsou aktuálně zobrazeny v seznamu měřících úloh. Navíc jsou zobrazovány pouze úlohy, u kterých již byla úspěšně dokončena jejich konfigurace. Dalšími sdílenými proměnnými jsou proměnné *measurement\_1* až *measurement\_3* (proměnné datového typu *double*). Každá proměnná náleží k jedné příslušné měřící úloze. Uchovávají aktuální naměřenou hodnotu z běhu měřící úlohy. Vzhledem k limitaci aplikace Data Dashboard je z každé měřící úlohy uchovávána pouze hodnota prvního kanálu měřící úlohy. Aplikace totiž nedovoluje zobrazení číselných polí nebo jiných složitějších datových typů.

Při konfiguraci síťově sdílené proměnné v prostředí LabVIEW, při jejím vytváření, jsou použita výchozí nastavení. Výjimkou je vypnutí použití bufferování. Jeho význam vyvstává při kolísání rychlosti čtení/zápisu čtecí/zapisovací stranou. Přítomnost bufferu pak zajistí, že žádná data nebudou zahozena bez povšimnutí protistrany. Požadavky řešení neuvážují nutnost zaznamenání každé změny a navíc jsou uspořeny systémové prostředky pro uchování tohoto bufferu, jak uvádí [23].

K tomu, aby byly síťově sdílené proměnné dostupné i při zapnutém firewallu systému Windows, je třeba změnit jeho konfiguraci. Podle [16] musí být povolena příchozí spojení pro sadu programů. Těmito programy jsou:

- C:\Windows\system32\lkads.exe
- C:\Windows\system32\lktsrv.exe
- C:\Program Files\National Instruments\Shared\Tagger\tagsrv.exe

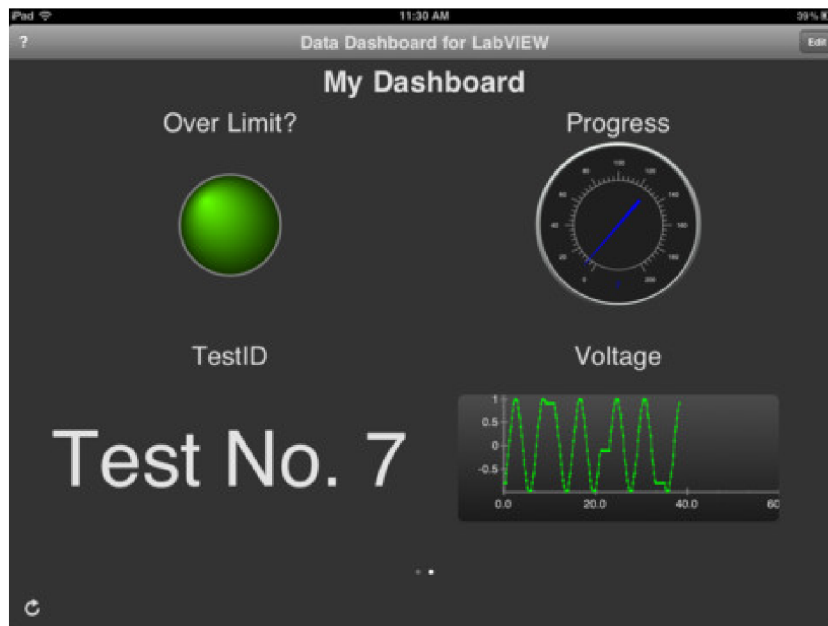
Jestliže tato sada programů má povolena příchozí spojení, tak se lze ze zařízení Apple iPad 2 bez problémů k dotykovému panelovému PC připojit.

### 3.10.2 Konfigurace NI Data Dashboard

Nejdříve je nutno zvolit šablonu rozložení pro sledované síťové proměnné. V závislosti na počtu proměnných, které mají být sledovány. K dispozici jsou šablony pro jednu, dvě, čtyři a šest proměnných. Po výběru šablony lze okna sledování populovat konkrétními proměnnými. Stiskem křížku s popiskem Add, v každém chlívěčku, vyskočí nabídka s lokalizací cílového zdroje dat. Na výběr jsou k dispozici senzory zařízení (akcelerometr, kompas) nebo vzdálené síťově sdílené proměnné (Network shared variable) a proměnné přístupné přes webové služby (web services). Číselné proměnné lze zobrazit v grafu, řetězcové hodnotě nebo v ručičkovém ukazateli. Booleovské proměnné lze zobrazit v řetězci, LED indikátoru nebo dokonce i v grafu. Po sestavení kolekce proměnných lze stiskem tlačítka Run, v horní části obrazovky, zahájit komunikaci. Při úspěšném připojení na cílovou stanici může obrazovka sledování vypadat jako na obrázku 3.22.

### 3.10.3 Aplikace NI Data Dashboard za běhu

Rychlost vyčítání síťově sdílených proměnných nelze v aplikaci NI Data Dashboard ovlivnit. Z tohoto poznatku plyne, že při vyčítání měřených průběhů nemusí a nejsou v grafu zastoupeny všechny hodnoty. Avšak význam použití této aplikace nespočívá v přenosu měřených dat do zařízení Apple iPad, ale o přibližné zachycení aktuálních měřených hodnot. Dalším



Obrázek 3.22: Příklad využití aplikace NI Data Dashboard [18]

významem je jednoduchá kontrola dostupnosti proměnných, která zaručuje správný běh průmyslového dotykového panelu a aplikace Univerzální měřicí ústředny.

## 3.11 Řízení systému

Ovládání řešení může díky pokročilosti architektury probíhat celkem třemi základními způsoby. Prvním z nich je využití grafického uživatelského rozhraní, které umožňuje plné využití všech dostupných vlastností. Druhým způsobem je ovládání přes příkazy zadávané změnou hodnot proměnných, které jsou umístěny na definovaném OPC serveru. Třetí a poslední možností řízení programu je kombinace obou variant využívající grafické uživatelské rozhraní a změnu hodnot OPC položek. GUI bylo již dopodrobna popsáno v samostatné kapitole a proto mu již nebude věnována další pozornost.

### 3.11.1 Ruční řízení

Ručním řízením je míněno řízení prováděné operátorem na dotykovém panelu, kde běží aplikace Univerzální měřicí ústředny. Operátor skrze GUI má plnou kontrolu nad všemi měřicími úlohami. Navíc může provádět některé úkony, které nejsou dostupné k provedení přes OPC příkazy. Výhodou ručního řízení je především komplexní přehled o chování a činnosti měřicích úloh. Po dokončení měření lze navíc využít silný aparát pro analýzu těchto dat v podobě kurzorů a přiblížení si významných oblastí grafu.

### 3.11.2 Řízení přes OPC

Proces řízení vytvořeného řešení následuje jednoduchý protokol vzájemné výměny příkazů a potvrzování jejich provedení s případným hlášením chyb. Seznam řídicích příkazů ze

Datový typ	Popis
CmdMeasTaskUsintFromBaR[1..10]	
unsigned integer 32-bitový	Pole hodnot, kdy každý jeden prvek pole náleží jedné měřicí úloze. Hodnoty prvků pole se poté starají o zahájení, zastavení nebo i zjištění stavu provádění úlohy. Pro zajištění jednoznačné identifikace manipulace s úlohou slouží konfigurační soubor. Pořadí úloh v konfiguračním souboru odpovídá pořadí indexů pole. Vystavení příkazu trvá vždy alespoň do doby příchodu odezvy s potvrzením. Tím je zajištěno, že všechny příkazy budou vykonány nebo vzaty na vědomí. 0 ... prázdný příkaz pro shoení aktuálního příkazu 1 ... start měření 2 ... zastavení měření

Tabulka 3.9: Vstupní řídicí příkazy OPC

strany PLC je možno nalézt v tabulce 3.9. Hodnoty těchto proměnných jsou v definovaných intervalech vyčítány z OPC serveru. Po každém úspěšném vyčtení hodnoty je hodnota porovnána s hodnotou předcházející. Jestliže nastala změna, pak následuje provedení akce podle aktualizované hodnoty proměnné. Výčet těchto příkazů představuje minimální množinu potřebnou pro vzájemnou interakci.

Datový typ	Popis
StatusMeasTaskUsintFromNi[1..10]	
unsigned integer 32-bitový	Aktuální stav měřicí úlohy určené indexem pole 0 ... úloha se nachází v chybovém stavu, doplňující informace je obsažena v StatusMeasErrorUsintFromNi 1 ... úloha připravena pro zahájení měření 2 ... měření právě probíhá
StatusMeasErrorUsintFromNi[1..10]	
signed integer 32-bitový	Chybové hlášení, které je platné ve chvíli, kdy se příslušná úloha nachází ve stavu 0
ButtonFromNiToBaR[1..10]	
unsigned integer 32-bitový	Šíření informace o stisku řídicích ovládacích prvků v GUI

Tabulka 3.10: Výstupní řídicí příkazy OPC

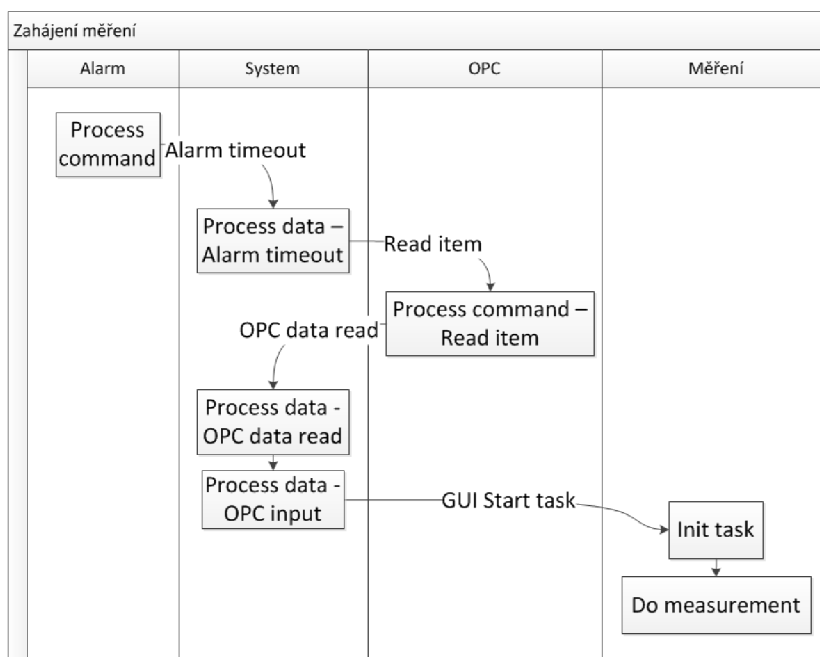
V tabulce 3.10 lze nalézt výpis výstupních stavů. Tyto stavy jsou aplikací hlášeny prostřednictvím změny proměnné OPC.

Řešení uvažuje využití pouze základních vlastností protokolu OPC. Tedy bez využití jeho pokročilých vlastností. Zjištění požadavku na provedení příkazu je odhaleno periodickým vyčítáním sady OPC proměnných. Moment k vykonání příkazu je stanoven na chvíli, kdy se objeví tak zvaná náběžná hrana. Tato náběžná hrana je určena změnou hodnoty OPC proměnné oproti poslední známé hodnotě. Nadřícená PLC část v žádném případě nevyvolá posloupnost dvou nebo více příkazů, které by nebyly postupně potvrzovány v jejich

provedení. Naproti tomu při zápisu výsledků provedení příkazu nebo chybového hlášení je vždy postupováno okamžitě bez prodlení. A to tak, aby případné prodlevy v provádění sekvence příkazů byly pokud možno minimální.

### 3.11.3 Příklad šíření zpráv měřicí ústřednou

Představené sestavení modulů vytváří netriviální ekosystém komunikací a paralelních dějů. Pochopení vzájemného provázání je nutným požadavkem v případě potřeby rozšíření nebo změny funkčnosti Univerzální měřicí ústředny. Za tímto účelem vzniklo grafické vyobrazení stavů (stavových automatů) a typů přenášených zpráv (na obrázku 3.23). Třeba vzít na vědomí, že toto vyobrazení je pouze horizontálním (vybrané moduly) a vertikálním (časovým) úsekem z celého systému. Komunikace se účastní čtveřice modulů (*Alarm*, *System*,

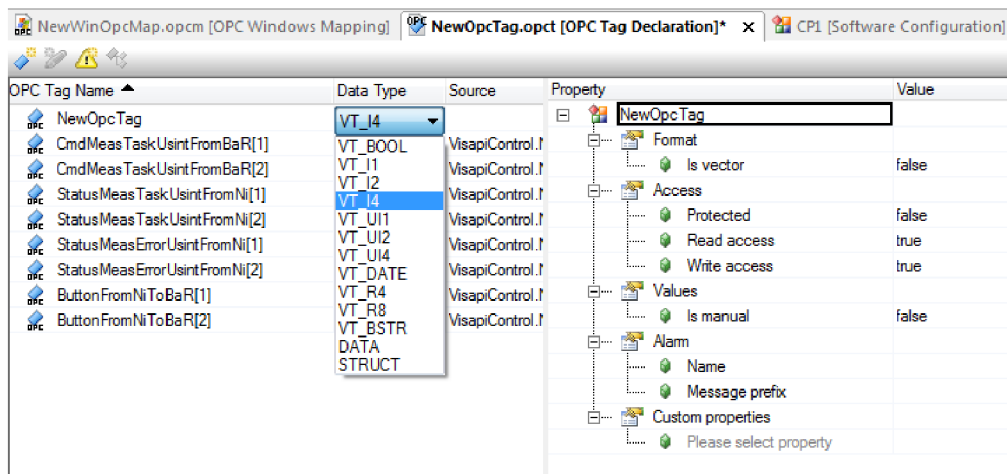


Obrázek 3.23: PLC zahájení měřicí úlohy

*OPC a Měření*). Celou sekvenci zahajuje vypršení (timeout) periodického časovače. Informace o timeoutu s identifikací časovače je vyhodnocena *Systémem* jako pokyn k dotázaní se na změnu všech OPC řídicích proměnných. *Systém* zjistí, že došlo ke změně některé z řídicích proměnných. Řídící proměnná OPC příkazuje zahájení konkrétní měřicí úlohy. *Systém* příkazuje *Měřicímu* modulu, aby zahájil provádění specifické měřicí úlohy. *Měřicí* modul vyhledá uloženou konfiguraci podle zadaného identifikátoru (*task ID*). Podle nalezené konfigurace je inicializována úloha měření. Následuje zahájení provedení měřicí úlohy. V průběhu činnosti měřicí úlohy jsou předávána měřená data, která jsou vizualizována v GUI.

### 3.11.4 Tvorba OPC konfigurace, nahrání na server

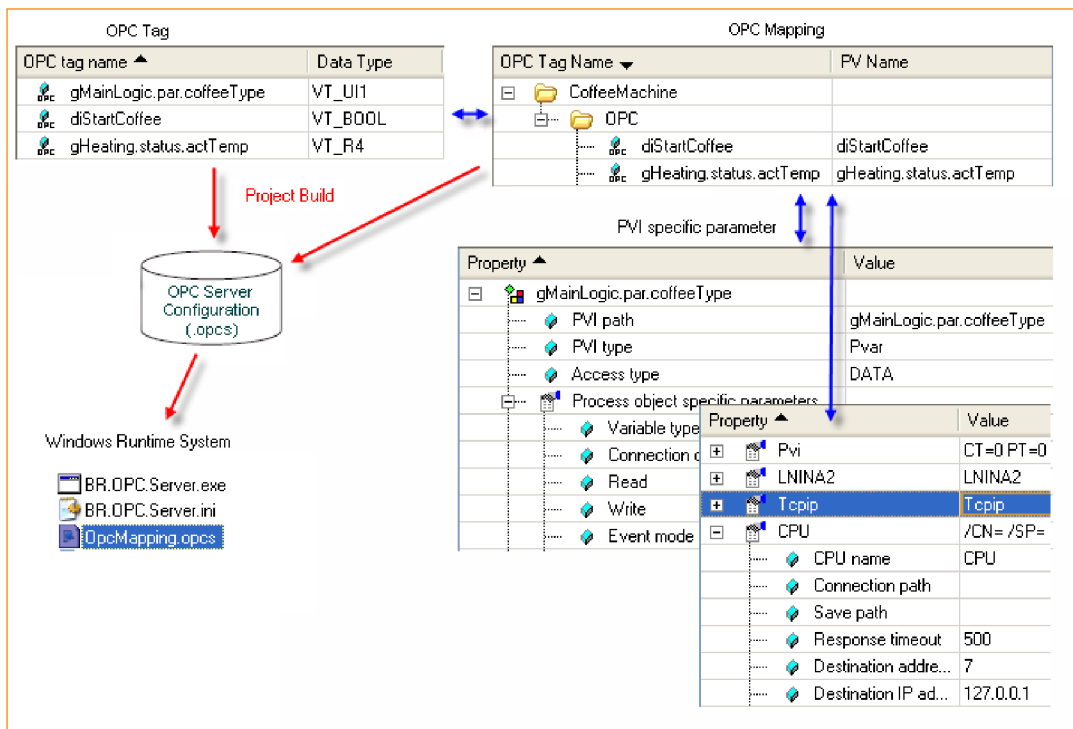
Komunikace nadřazené PLC části a podřazené měřicí ústředny je realizována prostřednictvím předem dané množiny OPC proměnných.



Obrázek 3.24: Vytvoření nového OPC tagu

Nyní bude představen způsob vytváření konfigurace OPC proměnných pro OPC server. K vytvoření konfigurace OPC proměnných byl použit nástroj Automation Studio V 3.0.90.19 SP01. Proces vytvoření konfigurace lze rozdělit do několika fází, které jsou popsány v [1]. V rámci vytvořeného projektu je třeba nejdříve v Logical view (sdružuje všechny použité elementy v projektu) definovat množinu proměnných. Tato množina je sestavována v OPC Tag Declaration editoru. Na obrázku 3.24 lze vidět proces vytváření nové proměnné. Zde lze kromě určení identifikace proměnné (OPC tagu), definovat datový typ, přístupová práva k proměnné a název alarmu vzniklého při změně hodnoty proměnné. OPC tagy musí být vzájemně unikátní a výsledkem jejich definice je soubor s příponou .opct.

Prostředí automatizace často vyžaduje komplexní řešení spolupráce několika samostatných systémů. K usnadnění jejich návrhu a vývoje lze s výhodou využít zde představených principů. Soubor s definicemi OPC tagů (.opct) ke svému použití vyžaduje mapování na specifický cíl. Mapování je definováno v editoru OPC Windows Mapping. V tomto editoru jsou definovány vlastnosti OPC komunikace a mapování OPC tagů na PVI objekty.



Obrázek 3.25: Sestavení konfigurace OPC serveru [2]

K sestavení OPC konfigurace je třeba mít k dispozici definice OPC tagů (.opct) a definice OPC mapování (.opcm). Z těchto dvou souborů je pak Automation Studio Builder schopen vygenerovat konfigurační soubor OPC serveru (.opcs). Formátem tohoto souboru je XML, který je tím pádem i ručně editovatelný. Celý postup sestavení je vyobrazen na obrázku 3.25.

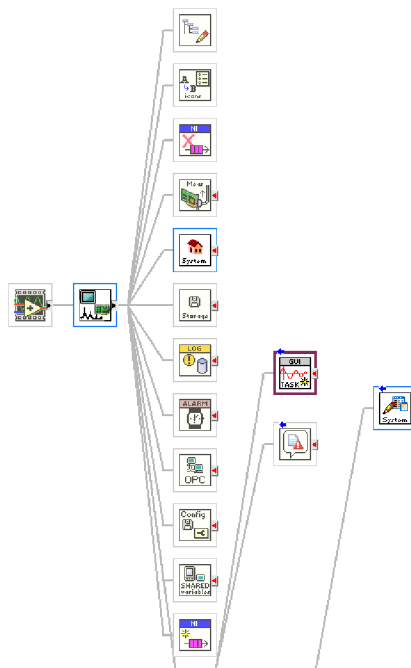
Po úspěšném sestavení konfiguračního souboru OPC serveru již zbývá pouze distribuce této konfigurace. Konfigurační soubor je třeba nahrát na PC, kde běží OPC server. Aby OPC server bral konfiguraci proměnných v úvahu, je třeba pozměnit konfigurační soubor OPC serveru a zařadit tak vytvořenou konfiguraci OPC proměnných k aktivně použitým.

### 3.12 Implementace řešení v LabVIEW

Na obrázku 3.26 lze pozorovat hierarchickou strukturu závislostí mezi zdrojovými soubory grafického návrhového prostředí LabVIEW. Na nejvyšší úrovni (umístěno zcela vlevo) stojí hlavní instance aplikace, která je závislá na hlavním zdrojovém souboru řešení. Od tohoto hlavního zdrojového souboru se dále dělí všechny závislé moduly, které jsou součástí řešení. Z tohoto obrázku je dobře patrné, že navržená modulární architektura byla úspěšně implementována. Celé řešení tak splňuje požadavky na jednoduchou rozšiřitelnost nebo záměnu modulů. Samozřejmostí je snadná údržba celého systému. Jedinou nutností při změně současného sestavení je přizpůsobení hlavního modulu *System*, který zajišťuje komunikaci se všemi moduly.

Zdrojový kód vytvořený v grafickém návrhovém prostředí LabVIEW je tvořen grafickými funkčními bloky, které jsou drátově propojeny. Kód tedy není tvořen souvislým textem, ale 2D prostorovým uspořádáním bloků. Ke zvýšení čitelnosti kódu slouží plovoucí komentáře, které provázejí náročněji pochopitelné části. Dalším vylepšením k čitelnosti

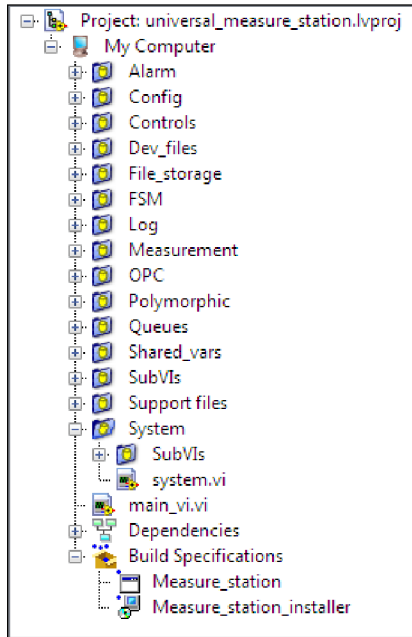




Obrázek 3.26: Hierarchie hlavních zdrojových souborů řešení

kódu jsou piktogramy, které znázorňují funkční bloky. Implementované funkční bloky proto nesou názorné piktogramy, které sledují významové souvislosti napříč systémem. Při editaci kódu v LabVIEW lze při přejetí myši nad funkčními bloky vidět tool-tip nápovědu s jednoduchým shrnutím funkcionality toho kterého funkčního bloku. Příklady částí zdrojových kódů je možno nalézt v příloze A.

Šíření dat mezi funkčními bloky v LabVIEW je založeno na dataflow modelu. Funkční blok je vykonán, jestliže má k dispozici data na všech svých vstupech. Po vykonání funkčního bloku jsou jeho výstupy šířeny na vstupy funkčních bloků, se kterými jsou spojeny. Zachování osvědčených postupů znamená, že funkční bloky jsou rozmístěny v ploše tak, aby dataflow výměna dat probíhala ve směru zleva-doprava. Snaha o minimalizaci ohybů drátových spojení mezi funkčními bloky zpřehledňuje kód a zlepšuje jeho estetický dojem. Všechny tyto techniky napomáhají vytváření samodokumentujícího kódu.



Obrázek 3.27: Hierarchie projektu univerzální měřící ústředny

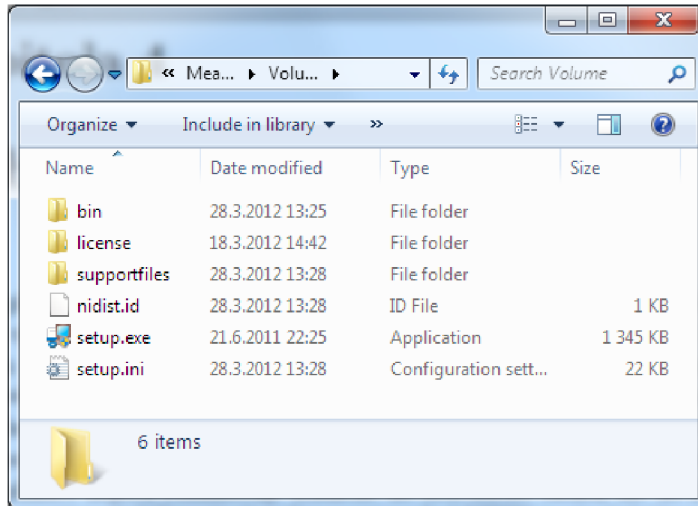
Při vytváření rozsáhlého modulárního systému je třeba sledovat promyšlenou strukturu modulů. Na obrázku 3.27 je náhled struktury projektu, kde jsou jednotlivé moduly rozmístěny do rozbalitelných složek. Tyto složky jsou typu *auto-populated*, což znamená, že jsou svázány se skutečnou složkou na diskovém úložišti. Obsah složky v projektu je tedy shodný s obsahem složky na disku. Rozšiřování obsahu složek automaticky aktualizuje soubory obsažené v projektu Univerzální měřící ústředny. Společně pro všechny moduly slouží složka *Controls*, která obsahuje všechny uživatelské datové typy použité v řešení.

## Kapitola 4

# Publikování aplikace

Vývoj aplikace měřicí ústředny probíhal odděleně od průmyslového panelového PC. Možnost spojení vývojového systému s cílovým je dostupná, ale vyžaduje větší kapacitu diskového prostoru pro uchování instalačních souborů vývojového prostředí. K zamezení nutnosti použití větší paměťové karty a také zajištění pohodlí procesu vývoje, byl projekt vyvíjen na samostatném stolním PC. Výhodou této volby je využití monitoru s dostatečným množstvím obrazových bodů. Tento požadavek je zvláště významný při grafických návrhových systémech, kdy funkční bloky aplikace jsou rozmístěny do 2D prostoru.

Vlastností grafického návrhového systému LabVIEW je, že dokáže z projektu vygenerovat instalační balík, který může být dále bez omezení distribuován. Výsledkem vytváření instalačního balíku jsou soubory a složky, které utváří instalační balík. Obsah vygenerované složky instalačního balíku je znázorněn na obrázku 4.1. Pokud nemají být součástí



Obrázek 4.1: Soubory instalačního balíku

instalačního balíku pouze zdrojové kódy programu, pak musí být nejdříve zkompileována aplikace do binárního spustitelného souboru. Před spuštěním kompilace je nutno provést základní nastavení mezi které patří například název výsledného .exe souboru. Dále je třeba nastavit zdrojové soubory, které budou při sestavení využity. Jeden zdrojový soubor je zvolen jako startovací (spuštěn při startu aplikace) a ostatní zdrojové soubory jsou přidány pro splnění závislostí. Z dalších nastaveních procesu kompilace je zajímavá podpora SSE2

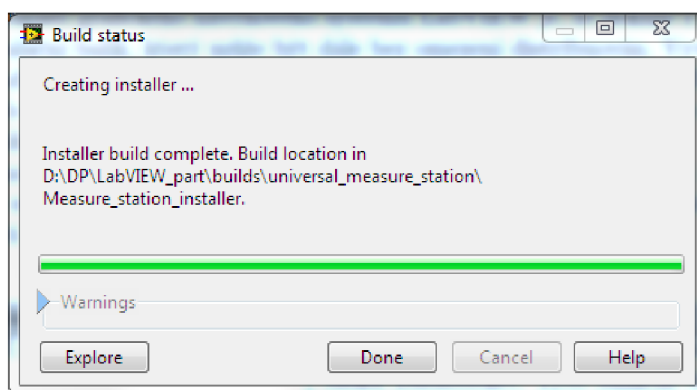
instrukcí, které jsou zahrnuty do použitého instrukčního mixu pro dosažení optimalizovaného běhu aplikace. Pro cílová zařízení, která tyto instrukce nepodporují, lze toto nastavení vypnout. Ve výsledné aplikaci jsou použity síťově sdílené proměnné (Network shared variable). Dostupnost těchto proměnných slouží nejenom ke sledování stavu činnosti řešení, ale i ke kontrole běhu Univerzální měřicí ústředny. Z toho důvodu bylo zvoleno nastavení, které při spuštění aplikace zavede síťově sdílené proměnné a při ukončení aplikace tyto proměnné zase z Shared Variable Engine (SVE) odstraní. Tím je zajištěna přítomnost sdílených proměnných jenom a pouze za doby běhu aplikace. Všechna ostatní nastavení jsou ponechána ve výchozích hodnotách.

Nyní bude text zaměřen na proces vytváření instalačního balíku. Vývojář si může zvolit celou řadu parametrů, kterými se bude instalační balík vyznačovat. Těmi jsou například zápisy do registru, cíle umístění zástupců po instalaci programu a nebo požadovanou verzi operačního systému. Všechny tyto parametry byly ponechány ve výchozích hodnotách, kdy jediné změny byly provedeny v dodatečných instalačních balíčcích, názvu projektu a názvu složky s projektem.

Jelikož se předpokládá, že v panelovém PC není předinstalována žádná aplikace ani ovladač od společnosti National Instruments, tak musí být do instalačního balíku potřebné závislosti přidány. Tyto závislosti jsou prezentovány v tabulce 4.1.

Název dodatečného balíčku	Popis
NI DataSocket 4.9	Komponenty, které umožňují využití funkčních bloků DataSocket. Tyto funkční bloky jsou využity pro komunikaci s OPC serverem.
NI LabVIEW Run-Time Engine 2011 se všemi částmi	Knihovny potřebné pro spuštění aplikací vytvořených v LabVIEW 2011
NI-DAQmx Core Runtime 9.3.5	NI-DAQmx driver pro komunikaci a řízení měření s podporovanými měřicími kartami.
NI Variable Engine 2.5.0	Podpora čtení a zápisu dat do sdílených proměnných (shared variables).

Tabulka 4.1: Soupis dodatečných instalačních balíčků



Obrázek 4.2: Úspěšné vytvoření instalačního balíku

Úspěšné zkompileování projektu a vytvoření instalačního balíku je signalizováno dialogovým oknem viditelným na obrázku 4.2.

Výhodou sestavení instalačního balíku je jeho snadná přenositelnost na libovolný počítač platformy Windows. Aplikaci lze použít nejenom na průmyslovém panelovém PC, ale i na běžných stolních počítačích. Tím se stává aplikace obecně široce využitelnou.

## Kapitola 5

# Testování systému

Ve chvíli, kdy je kompletní řešení úspěšně vytvořeno, přichází na řadu testování implementace. Při testování bylo použito hardwarové a softwarové sestavení, jak bylo uvedeno v předcházejících kapitolách. Cílem fáze testování je ověření správnosti návrhu, provedení implementace a funkčního propojení v jeden celek.

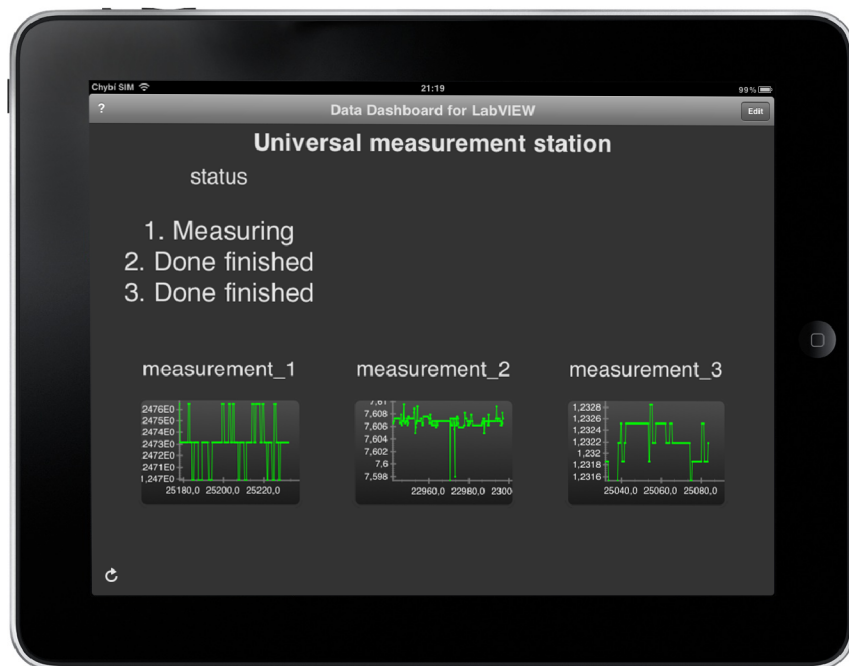
K testování byly využity celkem dva případy užití řešení. Prvním z nich je měření trojice baterií typu AAA, AA a bateriové sady jmenovitého napětí 7.2V. Ke konfiguraci měřicí ústředny je využito konfigurační souboru pouze pro nastavení síťově sdílených proměnných. Operátor provedl ruční konfiguraci celkem tří měřících úloh podle předpisu z tabulky 5.1.

Měřicí úloha	1	2	3
Typ měřicí úlohy	AI Voltage		
Rozsah měřeného napětí	0 ... 2 V	0 ... 2 V	0 ... 8 V
Vzorkovací frekvence	10 Hz		
Režim běhu úlohy	Continous sampling		
Měřicí kanály	Dev1/ai0, Dev1/AI GND	Dev1/ai1, Dev1/AI GND	Dev1/ai2, Dev1/AI GND
Konfigurace měřících kanálů	RSE		
Trigger	Ne		
Uložení měřených dat	Pouze operační paměť		

Tabulka 5.1: Konfigurace měření baterií

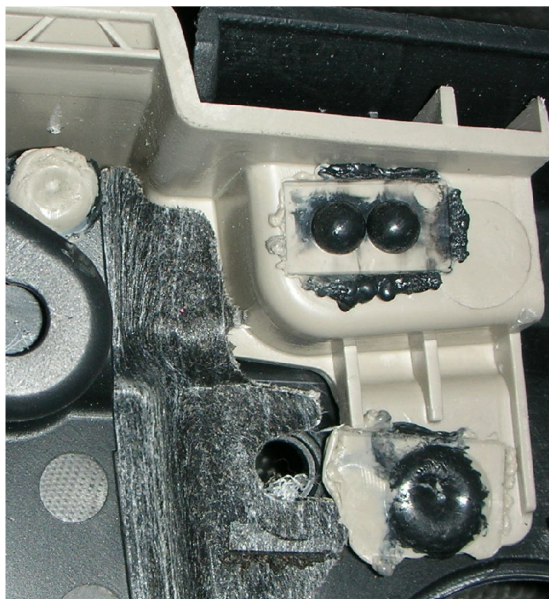
Pro měřicí úlohy je použito výlučně ruční ovládání operátorem přes dotykovou obrazovku průmyslového panelového PC. Operátor vybírá příslušnou měřicí úlohu klepnutím prstu na seznam úloh. Jakmile je objekt měření připraven k provedení úlohy, tak operátor spouští provedení úlohy adekvátním tlačítkem grafického uživatelského rozhraní. Díky aplikaci NI Data Dashboard může být činnost operátora sledována vzdáleně. Lze tak ověřit běh aplikace Univerzální měřicí ústředny i průběh měření. Na obrázku 5.1 je vidět obrazovka, která je zobrazena pracovníkovi se vzdáleným přístupem. Dává mu na vědomí, že aplikace měřicí ústředny běží a právě probíhá běh úlohy č.1.

Dalším příkladem nasazení řešení je měření sonotrod a konkrétně stroje pro ultrazvukové svařování plastů. V automobilovém průmyslu se používají ve velké míře plastové díly.



Obrázek 5.1: Náhled aplikace NI Data Dashboard

Interiér automobilu je vytvořen z velké části z plastů. Ke spojování plastů se s výhodou používají ultrazvukové generátory a sonotrody. Sonotroda je přitlačena pneumatickým válcem (s přednastaveným tlakem a tím určitou silou) na dotek s platovým dílem. Poté je sonotroda rozvibrována frekvencí asi  $30\text{kHz}$  a lokálně začíná tavit plastový nálitku ve tvaru trubičky s průměrem  $6\text{mm}$  nebo  $8\text{mm}$  a výškou  $12\text{mm}$ . Z plastového nálitku se přibližně do  $2\text{s}$  (sekund) vytvoří hlavička plastového nýtu o průměru asi  $10\text{mm}$  a výšce  $2\text{mm}$  (podle tvaru sonotrody). Sonotroda je tlačena do sváru stále stejnou silou. Technologie spojení plastů ultrazvukem vyžaduje kontrolu dráhy sonotrody. Je potřeba znát rychlost tavení plastu pod sonotrodou a včas ukončit činnost ultrazvukového generátoru. Téměř okamžitě po vypnutí ultrazvukového generátoru se zastaví i další tavení plastu. Výška hlavy nýtu musí odpovídat technologickému předpisu daného výrobku. Stroje pro ultrazvukové svařování plastů vyžadují měření průběhu dosažené dráhy sonotrody v čase. Výsledek procesu svařování sonotrodami zachycuje obrázek 5.2.



Obrázek 5.2: Výsledek procesu svařování sonotrodami

Podle průběhu dráhy v čase lze usuzovat i na kvalitu provedeného sváru. Postupem času dochází k opotřebení sonotrod. Při řešení technologických problémů je nutné porovnání průběhu dráhy v čase u téhož svaru nyní a např. před měsícem. V tomto případě lze s výhodou využít aplikaci Univerzální měřicí ústředny. Ta nabízí dostatečnou frekvenci měření a komfortní archivaci naměřených průběhů. Ovládání měřicí ústředny je intuitivní a po krátkém zaškolení dokáže výrobní operátor získávat cenné informace z výrobního procesu.

Měřicí úloha je definována v konfiguračním souboru podle předpisu z tabulky 5.2.

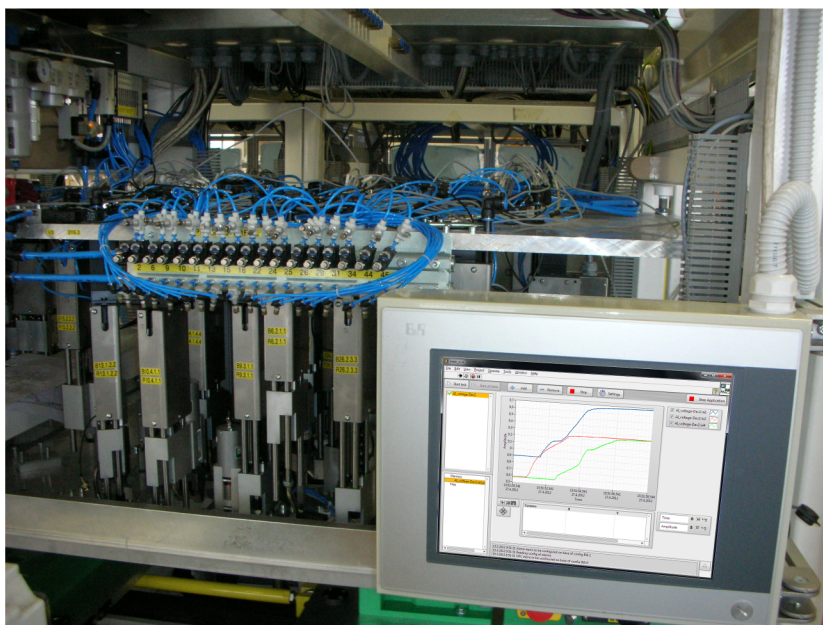
Typ měřicí úlohy	AI Voltage
Rozsah měřeného napětí	0 ... 10 V
Vzorkovací frekvence	1 kHz
Režim běhu úlohy	Finite samples
Počet vzorků úlohy	8k
Měřicí kanály	Dev2/ai1:2, Dev2/ai4, Dev2/AI GND
Konfigurace měřících kanálů	RSE
Trigger	Ne
Uložení měřených dat	Datový soubor
Typ datového souboru	.TDMS

Tabulka 5.2: Konfigurace měření polohy sonotrod

Po startu aplikace Univerzální měřicí ústředny a po načtení tohoto konfiguračního souboru, je měřicí úloha připravena ke spuštění. Její řízení je primárně v kompetenci řídicího PLC. Jelikož je třeba synchronizovat postupy výrobního procesu a archivace dat, proto je využito

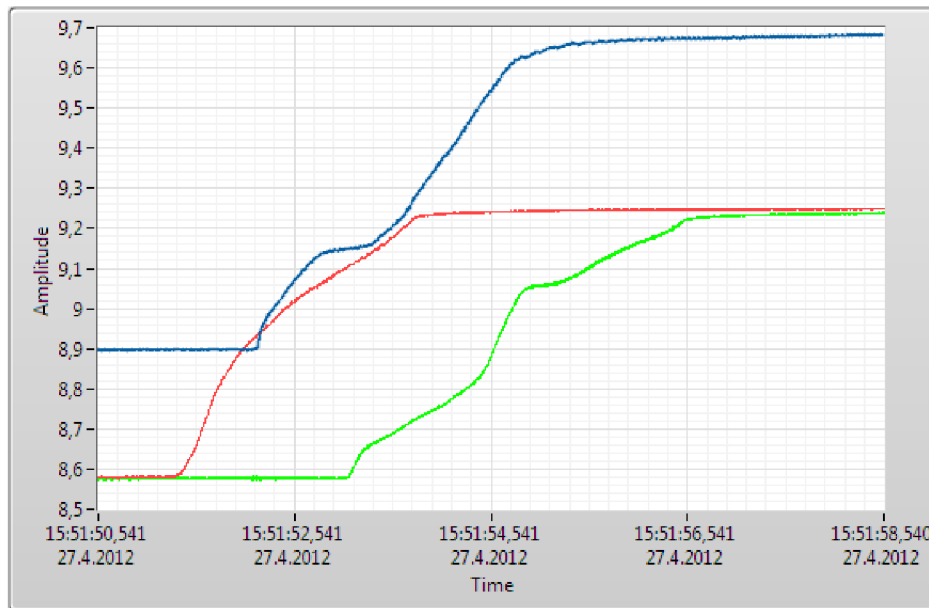


řídících OPC proměnných pro zahájení činnosti měření. Celkově jsou sledovány průběhy změny polohy tří sonotrod, které jsou zapojeny do měřicí karty NI USB-6215. Zvolená vzorkovací frekvence  $1\text{kHz}$  se jeví jako dostatečná pro přesné zachycení průběhu signálů. Fyzické zapojení měřících kanálů je takové, že při svislém pohybu sonotrod dolů narůstá hodnota napětí. Každá ze sonotrod má rozdílný rozsah možného posunu. Proto mohou být hodnoty počátečních a koncových napětí vzájemně mezi sonotrodami různé.



Obrázek 5.3: Nasazení Univerzální měřicí ústředny

Měření operátor sleduje v grafu na dotykovém panelu (obrázek 5.3) a kontroluje tak průběh měřených signálů. Automatizovaný výrobní proces vyžaduje archivaci a plnou automatizaci svých součástí. Zahájení, běh a ukončení měření je plně v kompetenci PLC. Příkazy jsou vydávány přes OPC proměnné, které slouží i ke zpětné vazbě o stavu provádění úloh a případných chybových hlášeních. Pro případ nutnosti dohledání měřeného průběhu u kusu, který je identifikován časem měření, lze nahlédnout do datového souboru obsahujícího měřený průběh.



Obrázek 5.4: Grafické zobrazení měření polohy sonotrod

Po dokončení měření si operátor zobrazuje celý průběh měření (obrázek 5.4), na jehož základě vyvodí důsledky dalšího postupu. V tuto chvíli jsou již všechna data bezpečně uložena do datového úložiště.

# Kapitola 6

## Závěr

Cílem této práce bylo vytvoření kompletního softwarového řešení pro provádění měřících úloh, uchování dat a jejich analýzu. Řízení činnosti programu lze provádět ručním zadáváním příkazu nebo prostřednictvím nastavení hodnot řídicích proměnných komunikovaných přes OPC. Optimalizace výsledného řešení pro průmyslové panelové PC s dotykovou obrazovkou byla nedílnou součástí řešení a představuje prostředek prvotního kontaktu s výslednou implementací. Dosažené cíle shrnuje vytvořený plakát, který je součástí přílohy B.

Ve fázi testování bylo ověřeno, že výsledek implementace splňuje kladená očekávání a svojí funkcí naplňuje vytyčené cíle. Rozšíření o dostupnost vzdáleného dohledu (díky zařízení Apple iPad 2) je příhodnou přidanou hodnotou nad rámec zadání.

Dalších rozšíření této práce se nabízí celá řada. Mezi nimi je širší podpora typů měření a analytické nástroje pro analýzu a zpracování měřených signálů. Díky modulární architektuře mohou být moduly do řešení jednoduše přidávány a vzájemně využívat svou specifickou funkcí.

### 6.1 Přínos řešení

Řešení Univerzální měřící ústředny vytvořené v prostředí LabVIEW od firmy National Instruments přináší následující benefity v oblasti tvorby software pro výrobní a měřící stroje.

Vývojové prostředí Automation Studio firmy Bernecker + Reiner pro softwarová PLC poskytuje vysokou efektivitu práce programátorů na rozsáhlém projektu [4]. Modulární koncept vývojového prostředí umožňuje paralelní týmový přístup ke zpracování rozsáhlého projektu. Přínosem je přehlednost kódu a zkrácení doby nutné k vývoji a ladění. V poslední době vzniká stále častěji potřeba zvládat měření a společně i řízení složitých technologických procesů. Měření fyzikálních veličin se stává stále více součástí řídicích systémů výrobních strojů. Tady už nestačí softwarová PLC. Musí nastoupit měřící ústředna vybavená hardware pro sběr velkého množství dat s vysokou vzorkovací frekvencí a dostupným kvalitním matematickým aparátem. Oblast měření je výsadou produktů firmy National Instruments. Z toho vzniká požadavek na měřící ústřednu s využitím výhod společných datových bází mezi řídicím a měřícím systémem. Obě úlohy pracují nad společnými datovými strukturami. Speciální měřící hardware firmy National Instruments nabízí velký rozsah měřících frekvencí s požadovanou přesností a příznivou cenou. Měřící úloha klade vysoké nároky na kvalifikaci specialistů pro měření a znamená i vysoké náklady na tvorbu aplikace. Je

výhodné věnovat podíl práce těchto specialistů pouze na měřicí části projektu. Touto je sběr dat, aplikace analytických matematických metod a zobrazení průběhu měření. Ostatní úlohy přebírá softwarové PLC.

PLC má vysoce efektivní nástroje pro vytvoření a diagnostiku motion úloh (tj. softwarové CNC systémy se servopohony, krokovými motory apod) [4]. PLC dnes běžně komunikují s SQL servery pro archivaci a validaci procesních dat a jsou vybaveny vyspělou souborovou správou. Diagnostika událostí, safety technologie, vysoká variabilita a možnost provázání různých komunikačních protokolů v rámci PLC umožní týmu programátorů efektivní dělbu práce při zpracování rozsáhlých projektů.

Přes rozdílný charakter úloh představuje společná datová základna a výhodné přerozdělení měřicí a ostatní softwarové práce kvalitativní posun a představuje následující výhody:

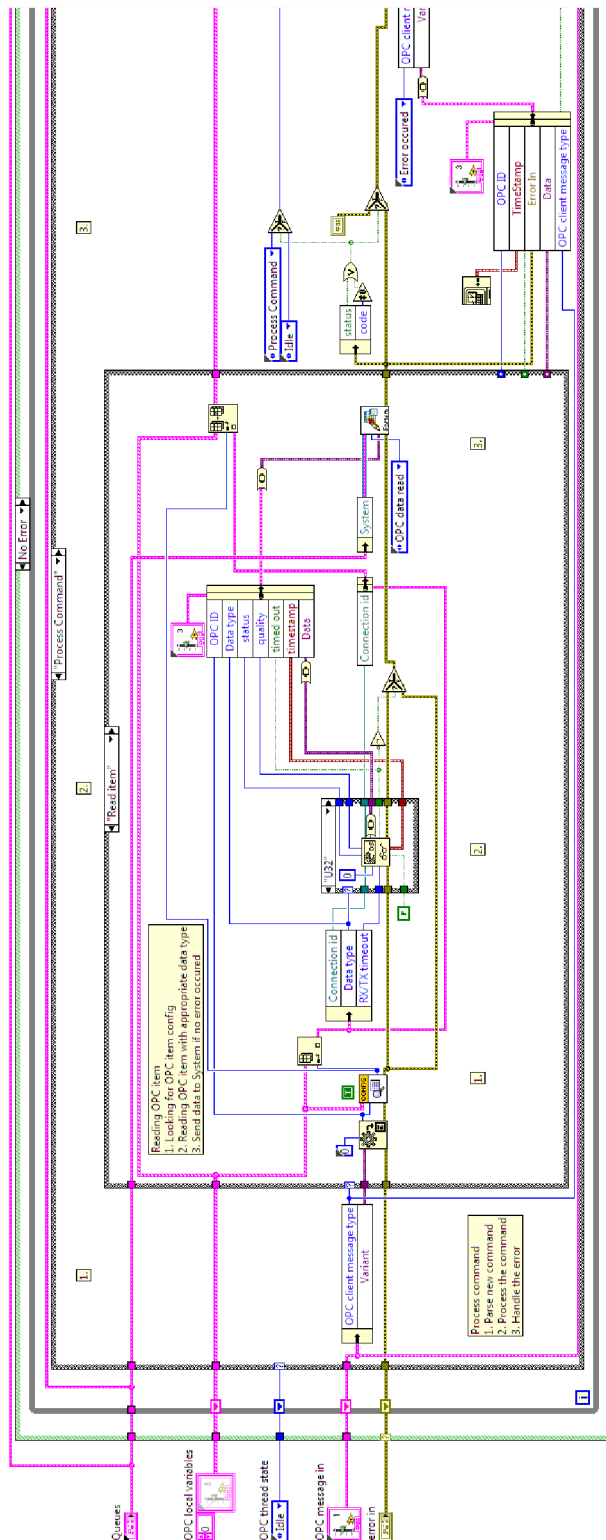
- zkrácení doby vývoje software a zkrácení doby ladění na rozsáhlých projektech
- snazší údržba software, protože změny prováděné PLC specialisty jsou snazší a více transparentní
- opakované využívání datového interface mezi měřicí úlohou a řídicí částí snižuje objem softwarové práce na projektu

# Literatura

- [1] Bernecker + Rainer Industrie-Elektronik Ges.m.b.H: *PVI OPC, TM730 (Training module)*. Leden 2009.
- [2] Bernecker + Rainer Industrie-Elektronik Ges.m.b.H: *Automation Studio 3.0.90.18 Help*. 2011.
- [3] Bernecker + Rainer Industrie-Elektronik Ges.m.b.H: *Innovations 2012*. 2011.
- [4] Bernecker + Rainer Industrie-Elektronik Ges.m.b.H: *Innovations 2012 - Controls, PCs, Panels, Motion Control, Drives, Safety Technology, Software*. Listopad 2011.
- [5] Bernecker + Rainer Industrie-Elektronik Ges.m.b.H: *Power Panel 500 User's Manual*. Květen 2011.
- [6] Bernecker + Rainer Industrie-Elektronik Ges.m.b.H: Power Panel 500, Model Number: 5PP520.1505-00 [online].  
[http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xsl/products\\_164267\\_ENG\\_HTML.htm](http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xsl/products_164267_ENG_HTML.htm), [cit. 2012-04-20].
- [7] Dvořák, V.; Drábek, V.: *Studijní opora k předmětu Architektura procesorů*. FIT VUT v Brně, 2006-01.
- [8] Hogg, S.: Creating Quality UIs with NI LabVIEW [online prezentace].  
<https://decibel.ni.com/content/docs/DOC-10961>, 2010-04-29 [cit. 2012-01-08].
- [9] Meanwell: *20W Single Output Industrial DIN Rail Power Supply MDR-20*. Červenec 2007.
- [10] National Instruments Corporation: *DAQ M Series - NI USB-621x User Manual*. Duben 2009.
- [11] National Instruments Corporation: *Data Acquisition and Signal Conditioning - Course Manual*. Únor 2010.
- [12] National Instruments Corporation: *LabVIEW<sup>TM</sup> Core 1 - Course Manual*. Srpen 2010.
- [13] National Instruments Corporation: *LabVIEW<sup>TM</sup> Core 2 - Course Manual*. Srpen 2010.
- [14] National Instruments Corporation: *LabVIEW<sup>TM</sup> Core 3 - Course Manual*. Duben 2010.
- [15] National Instruments Corporation: *LabVIEW<sup>TM</sup> Help*. Červen 2011.

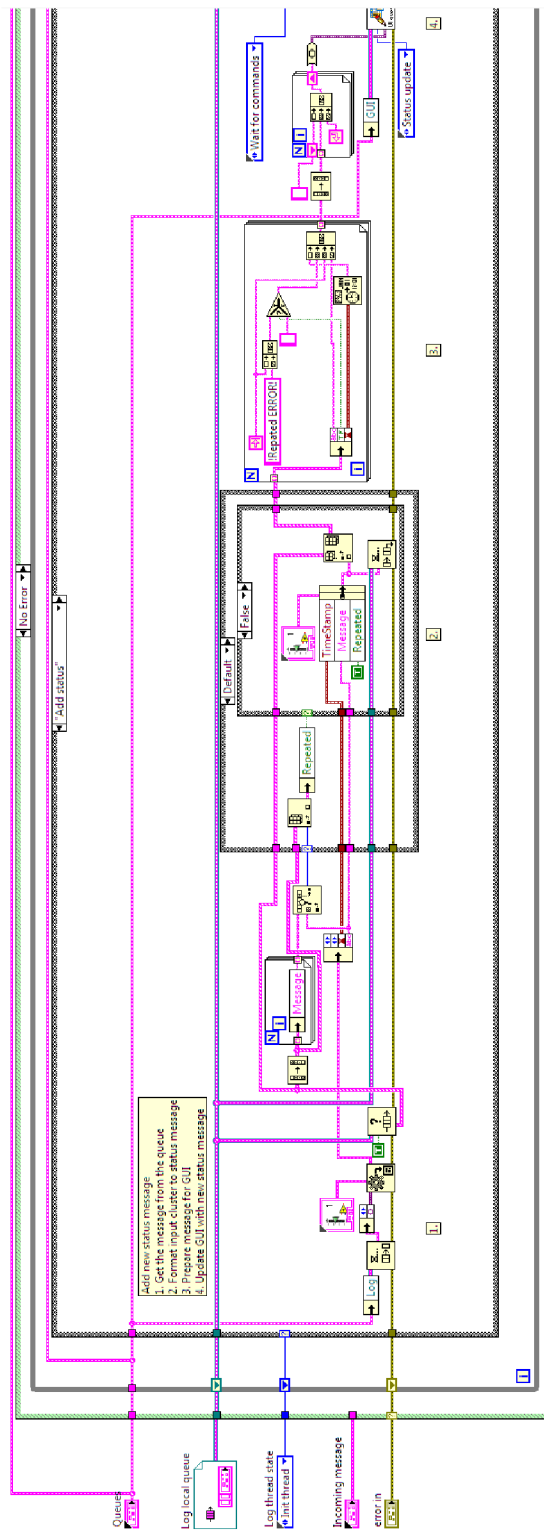
- [16] National Instruments Corporation: Configuring LabVIEW, LabVIEW DSC and Lookout to Work With the Windows Firewall [online].  
<http://digital.ni.com/public.nsf/allkb/0D7B86F4B4D19A5E86256F9A006EECB1>, 2005-01-31 [cit. 2012-04-06].
- [17] National Instruments Corporation: What Is OPC? [online].  
<http://zone.ni.com/devzone/cda/tut/p/id/7451>, 2008-12-08 [cit. 2012-01-06].
- [18] National Instruments Corporation: *Data Dashboard for LabVIEW* [online]. National Instruments, 2009-12-09 [cit. 2012-04-20].
- [19] National Instruments Corporation: NI Signal Streaming: Sustaining High-Speed Data Streams on External PC Buses [online].  
<http://zone.ni.com/devzone/cda/tut/p/id/4636>, 2010-07-30 [cit. 2012-04-07].
- [20] National Instruments Corporation: Using the Right Networking Protocol [online].  
<http://zone.ni.com/devzone/cda/tut/p/id/12079>, 2010-09-02 [cit. 2012-04-06].
- [21] National Instruments Corporation: *MDR-20-24* [online]. ATYSCO ONLINE STORE, 2010-09-28 [cit. 2012-04-20].
- [22] National Instruments Corporation: Answers to Frequently Asked Questions about NI-DAQmx and Traditional NI-DAQ (Legacy) [online].  
<http://zone.ni.com/devzone/cda/tut/p/id/3021>, 2012-01-04 [cit. 2012-04-06].
- [23] National Instruments Corporation: Using the LabVIEW Shared Variable [online].  
<http://zone.ni.com/devzone/cda/tut/p/id/4679>, 2012-03-04 [cit. 2012-04-05].
- [24] National Instruments Corporation: NI USB-6215 [online].  
<http://sine.ni.com/nips/cds/view/p/lang/en/nid/203483>, [cit. 2012-04-20].
- [25] PIXmania.com: APPLE iPad 2 [online].  
<http://www.pixmania.hu/hu/hu/10161386/art/apple/ipad-2-wifi-3g-64-gb-fekete.html>, [cit. 2012-04-20].
- [26] Stianko, M.: OPC v průmyslové komunikaci [online]. *Automa*, 2004 [cit. 2012-01-06].





Obrázek A.2: Modul OPC





Obrázek A.3: Modul Log

# Příloha B

## Plakát

**Univerzální měřicí ústředna**  
**Autor: Bc. Marek Šantavý**  
**Vedoucí: Ing. Václav Šimek**  
Vysoké Učení Technické v Brně  
Fakulta Informačních Technologíí  
Ústav počítačových systémů  
Akademický rok 2011/2012

Pokročilé měření napětí  
- 250 kS/s, +/- 10V  
- 16bit rozlišení  
Start trigger  
- HW i SW

Konfigurační soubor

Uchování měření do souborů  
- TDMS, CSV

**LabVIEW** Zdrojové kódy vytvořeny v LabVIEW

Modulární architektura

Vícevláknové zpracování

Ruční řízení  
- optimalizované GUI pro dotykový panel

Automatizované řízení  
- komunikace s PLC díky OPC proměnným

Vzdálený dohled  
- sledování průběhu měření

The poster includes several diagrams: a grid of measurement points, a block diagram of the system architecture with components like Alarm, OPC, Shared variables, Log, Measurement, and file storage, and a screenshot of the LabVIEW software interface showing a waveform plot.

Obrázek B.1: Plakát