

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY A ELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

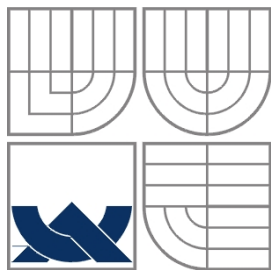
DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC ENGINEERING

AUTOMATIZOVANÉ MĚŘENÍ AUTOMOBILOVÝCH ALTERNÁTORŮ POMOCÍ PROGRAMU LABVIEW

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

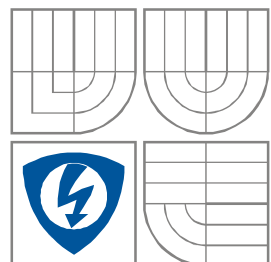
AUTOR PRÁCE
AUTHOR

Bc. Daniel Kufa



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**

**ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY
A ELEKTRONIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC
ENGINEERING

AUTOMATIZOVANÉ MĚŘENÍ AUTOMOBILOVÝCH ALTERNÁTORŮ POMOCÍ PROGRAMU LABVIEW

MEASUREMENT OF AUTOMOTIVE ALTERNATORS USING LABVIEW

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

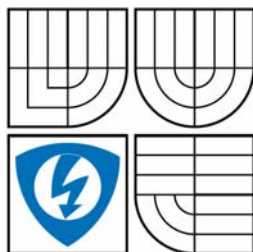
Bc. DANIEL KUFA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ VÍTEK, Ph.D.

BRNO 2009



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky a
komunikačních technologií**

Ústav výkonové elektrotechniky a elektroniky

Diplomová práce

magisterský navazující studijní obor
Silnoproudá elektrotechnika a výkonová elektronika

Student: Bc. Daniel Kufa

ID: 83560

Ročník: 2

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Automatizované měření automobilových alternátorů pomocí programu LabVIEW

POKYNY PRO VYPRACOVÁNÍ:

1. Popište konstrukci a princip funkce drápkového alternátoru a analyzujte možnosti automatizovaného měření pomocí programu LabView.
2. Navrhněte a realizujte přípravek pro měření.
3. Vytvořte komplexní měřicí systém pro analýzu drápkových alternátorů.
4. Zpracujte výsledky, vytvořte manuál k měřicímu systému a navrhněte další postup.

DOPORUČENÁ LITERATURA:

Dle doporučení vedoucího.

Termín zadání: 1.10.2008

Termín odevzdání: 22.5.2009

Vedoucí práce: Ing. Ondřej Vítek, Ph.D.

doc. Ing. Čestmír Ondrůšek, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Anotace

Práce seznamuje čtenáře s vývojovým prostředím LabVIEW a s možností tvorby měřicích virtuálních přístrojů. Konkrétní realizace měřicího virtuálního přístroje v rámci této práce je aplikována na měření základních charakteristik drápkového alternátoru. Je zde stručně popsána konstrukce a princip funkce zmíněného alternátoru. Vytvořený virtuální měřicí přístroj je v práci podrobně popsán. Kompletní blokové schéma, které je v prostředí LabVIEW zdrojovým kódem přístroje se nachází pouze v elektronické verzi. V tištěné verzi je k dispozici čelní panel přístroje a téměř všechny struktury z blokového schématu.

Klíčová slova

Virtuální měřicí přístroj, LabVIEW, drápkový alternátor

Abstract

This work presents development system LabVIEW and the possibility to create virtual measuring devices. Particular realization of virtual measuring device within the frame of this work is applied on measurement of basic characteristics of claw-pole alternator. The structure and the principle of function of this alternator is described in brief. Created virtual measurement device is described in detail. The entire block scheme, that is the source code for LabVIEW development system, is attached only in electronic form. The printed version contains front panel of the device and almost all the structures of block scheme.

Keywords

Virtual measurement instrument, LabVIEW, claw-pole alternator

Bibliografická citace

KUFA, D. Automatizované měření automobilových alternátorů pomocí programu LabVIEW. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 67 s. Vedoucí diplomové práce Ing. Ondřej Vítek, Ph.D.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Automatizované měření automobilových alternátorů pomocí programu LabVIEW jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

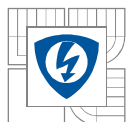
Podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Ondřeji Vítkovi, Ph.D. za konzultace, odbornou pomoc a užitečné rady, rovněž děkuji Ing. Rostislavu Huzlíkovi za připomínky a rady.

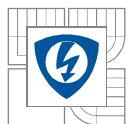
V Brně dne

Podpis autora



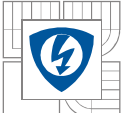
OBSAH

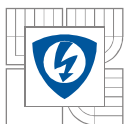
1 ÚVOD	12
2 ALTERNÁTOR S DRÁPKOVÝM ROTOREM.....	13
2.1 PRINCIP FUNKCE	13
2.2 KONSTRUKCE	13
3 LABVIEW.....	15
3.1 PROBLEMATIKA FLEXIBILITY MĚŘICÍHO PŘÍSTROJE	15
3.2 LABVIEW A VIRTUÁLNÍ MĚŘICÍ PŘÍSTROJ.....	16
4 PODROBNÝ POPIS VYTVOŘENÉ APLIKACE A JEJÍ TVORBY.....	20
4.1 VZHLED A FUNKCE VIRTUÁLNÍHO PŘÍSTROJE	20
4.2 NASTAVENÍ DAQ ASSISTANTA.....	25
4.3 JEDNOTLIVÉ ČÁSTI A STRUKTURY ZDROJOVÉHO KÓDU	28
5 PŘÍSTROJE SPOLUPRACUJÍCÍ S VYTVOŘENOU APLIKACÍ	55
5.1 MĚŘICÍ KARTA NI USB 6251	55
5.2 ELEKTRONICKÁ ZÁTĚŽ	55
5.3 ZDROJ BUDÍCÍHO PROUDU AGILENT E3634A	56
5.4 PŘEVODNÍK GPIB-USB.....	57
6 MĚŘICÍ PŘÍPRAVEK A SAMOTNÉ MĚŘENÍ	59
6.1 NÁVRH A REALIZACE PŘÍPRAVKU.....	59
6.2 MĚŘENÍ VYTVOŘENÝM VIRTUÁLNÍM PŘÍSTROJEM	60
7 ZÁVĚR.....	66
LITERATURA	67



SEZNAM OBRÁZKŮ

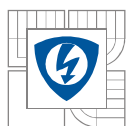
<i>Obrázek 2.1 Hlavní části alternátoru [2].....</i>	<i>13</i>
<i>Obrázek 2.2 Základní části alternátoru [2]</i>	<i>14</i>
<i>Obrázek 3.1 Ukázka čelního panelu virtuálního přístroje</i>	<i>17</i>
<i>Obrázek 3.2 Ukázka paletového menu prvků čelního panelu</i>	<i>17</i>
<i>Obrázek 3.3 Ukázka paletového menu funkcí a struktur blokového diagramu.....</i>	<i>18</i>
<i>Obrázek 3.4 Příklad blokového schémat virtuálního přístroje</i>	<i>19</i>
<i>Obrázek 4.1 Čelní panel virtuálního přístroje</i>	<i>20</i>
<i>Obrázek 4.2 Panel nastavení.....</i>	<i>21</i>
<i>Obrázek 4.3 Pokročilé nastavení 1.....</i>	<i>21</i>
<i>Obrázek 4.4 Pokročilé nastavení 2.....</i>	<i>22</i>
<i>Obrázek 4.5 Nastavení zdroje</i>	<i>23</i>
<i>Obrázek 4.6 Displej aktuálního měření.....</i>	<i>25</i>
<i>Obrázek 4.7 Úvodní okno DAQ Assistanta</i>	<i>26</i>
<i>Obrázek 4.8 Nastavení DAQ Assistantu.....</i>	<i>28</i>
<i>Obrázek 4.9 Různé bloky.....</i>	<i>29</i>
<i>Obrázek 4.10 Inicializační Case struktura.....</i>	<i>30</i>
<i>Obrázek 4.11 První algoritmus</i>	<i>32</i>
<i>Obrázek 4.12 Druhý algoritmus.....</i>	<i>35</i>
<i>Obrázek 4.13 Třetí algoritmus 1</i>	<i>36</i>
<i>Obrázek 4.14 Zátěž True</i>	<i>37</i>
<i>Obrázek 4.15 Zátěž False.....</i>	<i>37</i>
<i>Obrázek 4.16 Třetí algoritmus 2</i>	<i>38</i>
<i>Obrázek 4.17 Struktura Nastav proud.....</i>	<i>40</i>
<i>Obrázek 4.18 Ukončovací struktura.....</i>	<i>41</i>
<i>Obrázek 4.19 Podmínka 1</i>	<i>42</i>
<i>Obrázek 4.20 Podmínka 2</i>	<i>42</i>
<i>Obrázek 4.21 Podmínka 3</i>	<i>43</i>
<i>Obrázek 4.22 Příklad zprávy.....</i>	<i>43</i>
<i>Obrázek 4.23 Struktura Konec měření.....</i>	<i>43</i>
<i>Obrázek 4.24 Struktura Ukaž.....</i>	<i>44</i>
<i>Obrázek 4.25 Blokovací Case struktura.....</i>	<i>45</i>
<i>Obrázek 4.26 Ochranná Case struktura - True.....</i>	<i>46</i>

	ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY A ELEKTRONIKY Fakulta elektrotechniky a komunikačních technologií Vysoké učení technické v Brně	10
Obrázek 4.27	Ochranná Case struktura – False.....	47
Obrázek 4.28	Nastavovací Case struktura	48
Obrázek 4.29	Měřicí Case struktura	48
Obrázek 4.30	Vypínací a Zapínací Case struktury.....	49
Obrázek 4.31	Zjišťovací Case struktura.....	49
Obrázek 4.32	Ověřit stav výstupu?.....	50
Obrázek 4.33	Resetovací Case struktura	50
Obrázek 4.34	Zatěžovací Case struktura	51
Obrázek 4.35	DAQ Asisstant Zátěže	51
Obrázek 4.36	Status Case struktura	52
Obrázek 4.37	Přenastavovací case struktura.....	53
Obrázek 4.38	Blikací Case struktura	53
Obrázek 4.39	Varovná struktura	54
Obrázek 4.40	Mód struktura	54
Obrázek 4.41	Ukládací Case struktura	54
Obrázek 5.1	Měřicí karta [10]	55
Obrázek 5.2	Elektronická zátěž [12]	56
Obrázek 5.3	Zdroj Agilent E3634A [11]	57
Obrázek 5.4	GPIB-USB převodník [13].....	58
Obrázek 6.1	Schéma zapojení vstupů a výstupu.....	59
Obrázek 6.2	Návrh desky plošného spoje.....	59
Obrázek 6.3	Osazení DPS součástkami	60
Obrázek 6.4	Realizace DPS.....	60
Obrázek 6.5	Zjednodušené schéma zapojení.....	61
Obrázek 6.6	Charakteristika naprázdno 1	63
Obrázek 6.7	Charakteristika naprázdno 2	63
Obrázek 6.8	Zatěžovací charakteristiky	64
Obrázek 6.9	Graf z přístroje 1.....	64
Obrázek 6.10	Graf z přístroje 2.....	65



SEZNAM TABULEK

<i>Tabulka 4.1 Inteligentní systém blokace prvků čelního panelu přístroje.....</i>	<i>25</i>
<i>Tabulka 4.2 Konfigurace analogových vstupů [10]</i>	<i>27</i>
<i>Tabulka 4.3 Informační zprávy a jím příslušné priority</i>	<i>52</i>
<i>Tabulka 6.1 Závislost $U_{výst}$ na I_b 1.....</i>	<i>61</i>
<i>Tabulka 6.2 Závislost $U_{výst}$ na I_b 2.....</i>	<i>61</i>
<i>Tabulka 6.3 Závislost $U_{výst}$ na n</i>	<i>62</i>
<i>Tabulka 6.4 Závislost I, M na n 1.....</i>	<i>62</i>
<i>Tabulka 6.5 Závislost I, M na n 2.....</i>	<i>62</i>



1 ÚVOD

V elektrotechnice a nejen v tomto odvětví je potřeba měřit, analyzovat a vyhodnocovat různé veličiny. K tomuto účelu slouží mnoho různorodých měřicích přístrojů lišící se zejména v účelu použití. Výrobci měřicích přístrojů je celá řada, proto i kvalita měřicích přístrojů bývá odlišná. Výrobci měřicích přístrojů jsou, stejně jako jakýkoliv jiní výrobci, součástí tržního hospodářství. Jejich cílem tedy je ovládnout co největší část trhu. To je cíl každého výrobce, který chce v široké škále konkurence na trhu úspěšně obstát. Snaží se tedy své výrobky přizpůsobovat tak, aby co nejvíc vyhovovaly poptávce potencionálního uživatele, a tím uspokojili jeho potřebu. V trendu a vývoji měřicích přístrojů platí, že čím je měřicí přístroj flexibilnější, tím lépe dokáže uspokojit potřeby zákazníka. Avšak moderní zákazník je stále náročnější a mnohdy se nespokojí ani s velmi kvalitním a vícefunkčním multimetrem.

To byl jeden z důvodů, proč se začaly vyvíjet vývojová prostředí pro oblast měření. Další z důvodů byla snaha o vyloučení lidského činitele při přebírání naměřených dat (obvykle opis z displeje nebo ukazatele měřicího přístroje), a tím celý proces měření zefektivnit a omezit zdroje chyb. Jedním z takovýchto vývojových prostředí je LabVIEW. Patří mezi rozšířenější prostředí jak na vysokých školách (i středních průmyslových), tak ve firmách. Je to vývojové prostředí pro průměrného uživatele přívětivé, jelikož je orientováno graficky a při tom je na úrovni jazyka C, který je však orientován textově. Díky tomuto prostředí má jeho uživatel téměř neomezené možnosti při tvorbě měřicích analyzujících simulačních či vyhodnocovacích aplikací. Po vytvoření této aplikace je možno ji kdykoliv modifikovat nebo jen upravit vlastnosti stávající nebo při podstatnější změně vytvořit zcela jinou aplikaci. Právě proto LabVIEW z hlediska flexibility splňuje nároky uživatele takřka dokonale. Uživatel má možnost vytvořit si virtuální měřicí přístroj přesně na míru, dle svých představ a požadavků. Při tvorbě virtuálního přístroje už záleží pouze na jeho schopnostech, kreativitě a myšlení.

Pro automatizované měření je správně vytvořený virtuální přístroj rovněž vhodný, což je jeho další nesporná výhoda. I proto je využito vytvořeného virtuálního přístroje v rámci této diplomové práce k měření charakteristik drápkového alternátoru. Drápkový alternátor je v automobilové elektrotechnice, zejména u osobních automobilů, téměř výhradně používán jako zdroj energie pro jedoucí vozidlo. Jeho funkce je dobíjet akumulátor a dodávat energii do elektrické sítě automobilu. Nepotřebuje téměř žádnou údržbu. V dnešní době už jsou alternátory velice kvalitní, takže i jejich poruchovost klesá a životnost se prodlužuje. Téměř každý by čekal, že náchylnější k poruše je mechanická část. Není tomu tak, nejčastěji se kazí usměrňovače.

Vzhledem k výše zmíněným důvodům je zvoleno prostředí LabVIEW k vytvoření virtuálního měřicího přístroje v rámci tohoto semestrálního projektu. Toto prostředí je v některých aspektech velmi intuitivní, v některých nikoliv a člověku se usnadňuje práce až když má alespoň mírné programátorské uvažování a přemýšlení. Vytvořený virtuální měřicí přístroj je vhodný pro měření základních charakteristik alternátoru. Což jsou charakteristiky naprázdno a zatěžovací. Při měření virtuální přístroj ukládá data na pevný disk počítače, takže s nimi jde následně libovolně manipulovat dle aktuální potřeby. Velice snadno a jednoduše se naměřená data uložená do souboru *.lvm dají importovat kupříkladu do programu Microsoft Excel. Odpadá tak zbytečná práce, která doprovází klasické měření, kdy se hodnoty zapisují na papír, a teprve pak se přepisují do tabulkového procesoru ke zpracování.

2 ALTERNÁTOR S DRÁPKOVÝM ROTOREM

V automobilech a u větších motorových vozidel, kde je potřebný větší výkon alternátoru, se používají alternátory, kde je rotor buzen stejnosměrným proudem, a u nichž je nezbytná regulace napětí. Jako zdroj energie je proto nejčastěji používán synchronní alternátor s drápkovým rotorem.

2.1 Princip funkce

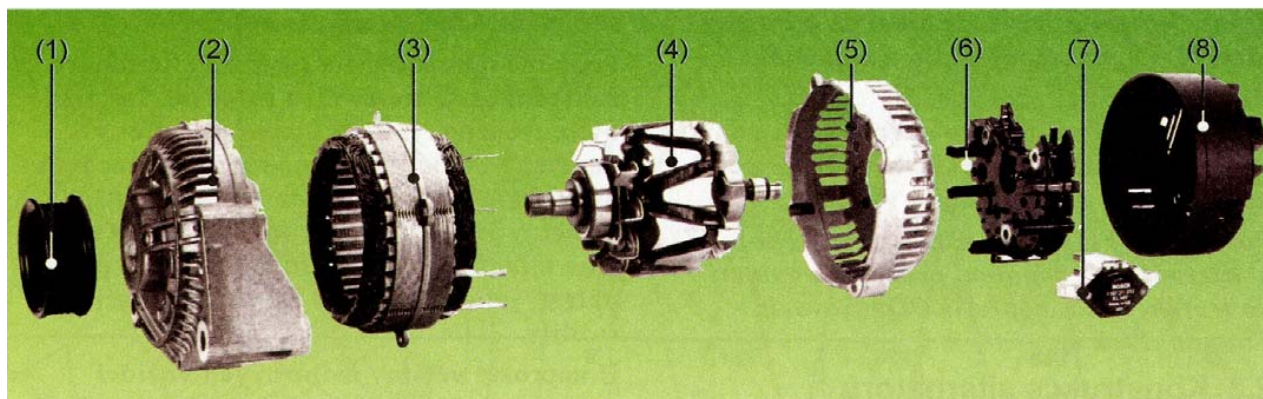
Účelem a smyslem každého alternátoru je převést mechanickou energii na energii elektrickou. Alternátor je točivý elektrický stroj a při své funkci využívá točivé magnetické pole. Na rotoru je umístěno budící vinutí, kterým protéká stejnosměrný proud. Lze si tedy představit, že na rotoru je permanentní magnet. Na statoru je pracovní vinutí. Točící se magnet (rotor) a tedy i točící se magnetické pole působí na vodiče statoru. A stejně tak, jako se ve vodiči pohybujícím se v magnetickém poli indukuje napětí se ve stojícím vodiči protínaném pohyblivým (točivým) magnetickým polem indukuje rovněž. Tímto způsobem alternátor vyrábí střídavý elektrický proud, který je usměrněn v šestipulsním usměrňovači.

2.2 Konstrukce

Jeho výhodou je malý rozměr a nižší poruchovost vzhledem k jiným točivým zdrojům energie. Liší se oproti klasickému synchronnímu stroji s vyniklými póly konstrukcí rotoru. Na rotoru se nachází místo klasických vyniklých pólů póly drápkové – odtud název alternátor s drápkovým rotorem. Díky této konstrukci rotoru je pak možné vytvořit větší počet pólů při menší délce stroje. Synchronní alternátory s drápkovým rotorem bývají obvykle navíc doplněny na víku stroje neřízeným šestipulsním usměrňovačem, tak aby na výstupu bylo přímo jen stejnosměrné zvlněné napětí.

Obrázek 2.2 velmi názorně zobrazuje nejzákladnější části alternátoru navíc je v něm zahrnuto i zapojení vinutí a usměrňovače. Způsob jakým na sebe jednotlivé díly alternátoru navazují je patrný z Obrázek 2.1. Jelikož alternátor neslouží pouze jako zdroj elektrické energie, ale i k nabíjení akumulátoru, je potřeba usměrnit střídavý proud v něm vyrobený. Nedílnou součástí alternátoru je tedy i usměrňovač. Regulátor je rovněž jeho součástí.

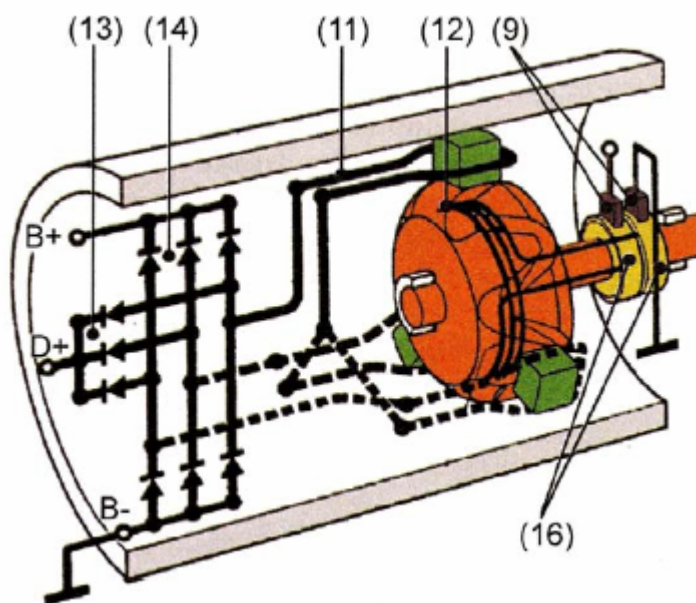
Základní části u alternátoru tvoří stator (3) a rotor (4) – podobně jako u dynam. Avšak v tomto případě, jak již bylo řečeno, je budící vinutí umístěno na rotoru a na statoru je pak vinutí



Obrázek 2.1 Hlavní části alternátoru [2]

- (1) - drážková řemenice
- (3) - stator
- (5) - zadní víko(diodové)
- (7) - držák kartáčů s regulátorem

- (2) - přední víko (břemenové)
- (4) - rotor
- (6) - usměrňovač
- (8) - krycí víko

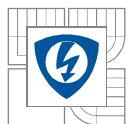


- (9) - kartáče
- (13) - pomocný usměrňovač
- (11) - pracovní vinutí
- (14) - hlavní usměrňovač
- (12) - budicí vinutí
- (16) - sběrací kroužky

Obrázek 2.2 Základní části alternátoru [2]

pracovní. Určité rozdíly v konstrukci jednotlivých druhů alternátorů jsou dány oblastí jejich použití, ale elementární konstrukční principy zůstávají stejné u všech.

V drtivé většině motorových vozidel se v současné době používají právě alternátory s drápkovým rotorem. Hlavní rozdíly v konstrukci jednotlivých typů jsou v umístění ventilátoru (uvnitř nebo vně), v počtu oběžných kol ventilátoru (jedno nebo dvě), ve velikosti sběracích kroužků a v umístění usměrňovače (uvnitř nebo vně předního břemenového víka).



3 LABVIEW

3.1 Problematika flexibility měřicího přístroje

Starší klasické analogové měřicí přístroje, které obvykle výhradně plnily jedinou funkci (např. voltmetr určen pro měření napětí) a jejich flexibilita byla tedy mizivá, brzy nevyhovovaly nárokům uživatelů, kteří byli nuceni kupovat další a další přístroje, pokud chtěli realizovat složitější měřicí aplikaci. Čili potřeby uživatele jediný přístroj, schopný měřit pouze jedinou veličinu, málokdy uspokojil. Nejen proto vznikly měřicí přístroje vybaveny komunikačním rozhraním poskytující přímou komunikaci přístroje s počítačem.

Propojení počítače s měřicím přístrojem umožňuje doplnit funkce definované jeho výrobcem funkcemi, které lze naprogramovat do řídicího počítače. Funkce měřicího přístroje lze v tomto případě obvykle omezit pouze na sejmutí měřených dat a jejich přenos přes rozhraní do počítače, kde proběhne jejich následné zpracování, při kterém lze softwarově realizovat i funkce, které do měřicího přístroje nebyly implementovány. Takto lze např. z digitálního osciloskopu jednoduchým způsobem udělat frekvenční analyzátor, který je obvykle o řád dražší (osciloskop pouze sejme měřený průběh a pošle jej přes sběrnici do počítače, kde lze podrobit tato data frekvenční analýze např. algoritmem rychlé Fourierovy transformace FFT).

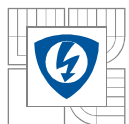
Nahrazení koncového uživatele počítačem přináší dvě zásadní výhody:

- možnost automatizace celého měření a tím jeho zefektivnění z hlediska časové náročnosti celého procesu. Počítač zvládá mnohem rychlejší toky dat, což ve svém důsledku zkracuje jak dobu konfigurace měřicího přístroje, tak dobu vyčítání naměřených hodnot.
- vyloučením lidského faktoru z měřicího řetězce lze eliminovat další zdroj chyb, které zatěžují výsledek měření, a to chyby osobní.

Při analýze propojení měřicího přístroje přes komunikační rozhraní na počítač přišli výrobci měřicí techniky na to, že se z měřicího přístroje někdy využije pouze vstupních obvodů a paměti dat, což vedlo k dalšímu kroku vývoje této techniky směrem k tzv. virtuálním měřicím přístrojům.

Podstatou virtuálního přístroje je doplnění otevřené architektury personálního počítače tím, co mu chybí, aby mohl plnit úlohu měřicího přístroje. V oblasti hardwaru je to zásuvná multifunkční karta vybavena příslušným konektorem. V oblasti softwaru je to vhodný program pro počítač, který realizuje všechny funkce měřicího přístroje. Tyto karty mají však omezení hlavně v parametru dosažitelné vzorkovací frekvence a současnosti snímání z více kanálů. Běžné měřicí karty např. pro ISA sběrnici dosahují maximální vzorkovací frekvence řádu stovek tisíc až miliónů vzorků za sekundu (100 kS/s – 1 MS/s) pro signál, který není periodický. Špičkové osciloskopy dnes mají tento parametr až o čtyři řády lepší (10 GS/s).

Filosofie virtuálních měřicích přístrojů je velmi pokroková, neboť umožňuje, při zachování výkonnostních parametrů klasické měřicí techniky, vytvářet přístroje, jejichž funkce přesně odpovídají požadavkům uživatele, neboť jsou tyto funkce realizovány softwarově. Navíc tato koncepce umožňuje doplňovat další funkce podle narůstajících a měnících se potřeb koncového uživatele. Cena takto vytvářených měřicích přístrojů bývá nižší, než je cena klasické analogové měřicí techniky. Navíc je možno kdykoliv změnou programu vytvořit jiný měřicí přístroj nebo upravit vlastnosti stávajícího. Při spojení měřicího přístroje a počítače přes rozhraní nebo při vytváření virtuálních měřicích přístrojů hraje vedle hardwarových prostředků v podobě počítače a



přídavných karet čím dál významnější roli software - stává se klíčovým komponentem měřicího systému.

3.2 LabVIEW a virtuální měřicí přístroj

Vývojové prostředí LabVIEW stojí na jednoduché myšlence. Vychází se zde z poznatku, že tím, kdo ví, co měřit, jak analyzovat a jak prezentovat data, je technik, který nemusí být sám zkušeným programátorem. Své představy tedy předává programátorovi obvykle v podobě blokového schématu. Programátor toto schéma potom převádí do syntaxe zvoleného programovacího jazyka, což je činnost poměrně zdoluhavá a náročná na přesnost a nepřináší již do procesu měření obvykle žádné další nové informace. Cílem vývojového prostředí LabVIEW je to, aby blokové schéma bylo koncovým tvarem aplikace, který se již dále nebude převádět do textové podoby. LabVIEW (Laboratory Virtual Instruments Engineering Workbench) je obecným vývojovým prostředím s bohatými knihovnami pro vytváření aplikací zaměřených do oblasti měření ve všech fázích tohoto procesu - tj. sběru, analýzy a prezentace naměřených dat. Poskytuje uživateli plnohodnotný programovací jazyk se všemi odpovídajícími datovými a programovými strukturami v **grafické podobě** - tzv. G jazyk (Graphical language). Virtuální přístroj jako základní jednotka aplikace vytvořené v tomto vývojovém prostředí obsahuje:

- interaktivní grafické rozhraní (Graphical User Interface - GUI) ke koncovému uživateli - tzv. **čelní panel** (Front Panel), který simuluje čelní panel fyzického přístroje. Obsahuje prvky pro ovládání a indikaci (knoflíky, tlačítka, LED indikátory, grafy ...). Tento čelní panel ovládá uživatel myší nebo z klávesnice.
- činnost virtuálního přístroje je dána jeho **blokovým schématem** (Block Diagram). Toto blokové schéma je vytvořeno ikonami reprezentujícími v koncových blocích ovládací a indikační prvky čelního panelu a ve svých uzlových blocích jsou to bloky zpracovávající procházející data. Tento blokový diagram je zdrojovou podobou každé aplikace. [4]

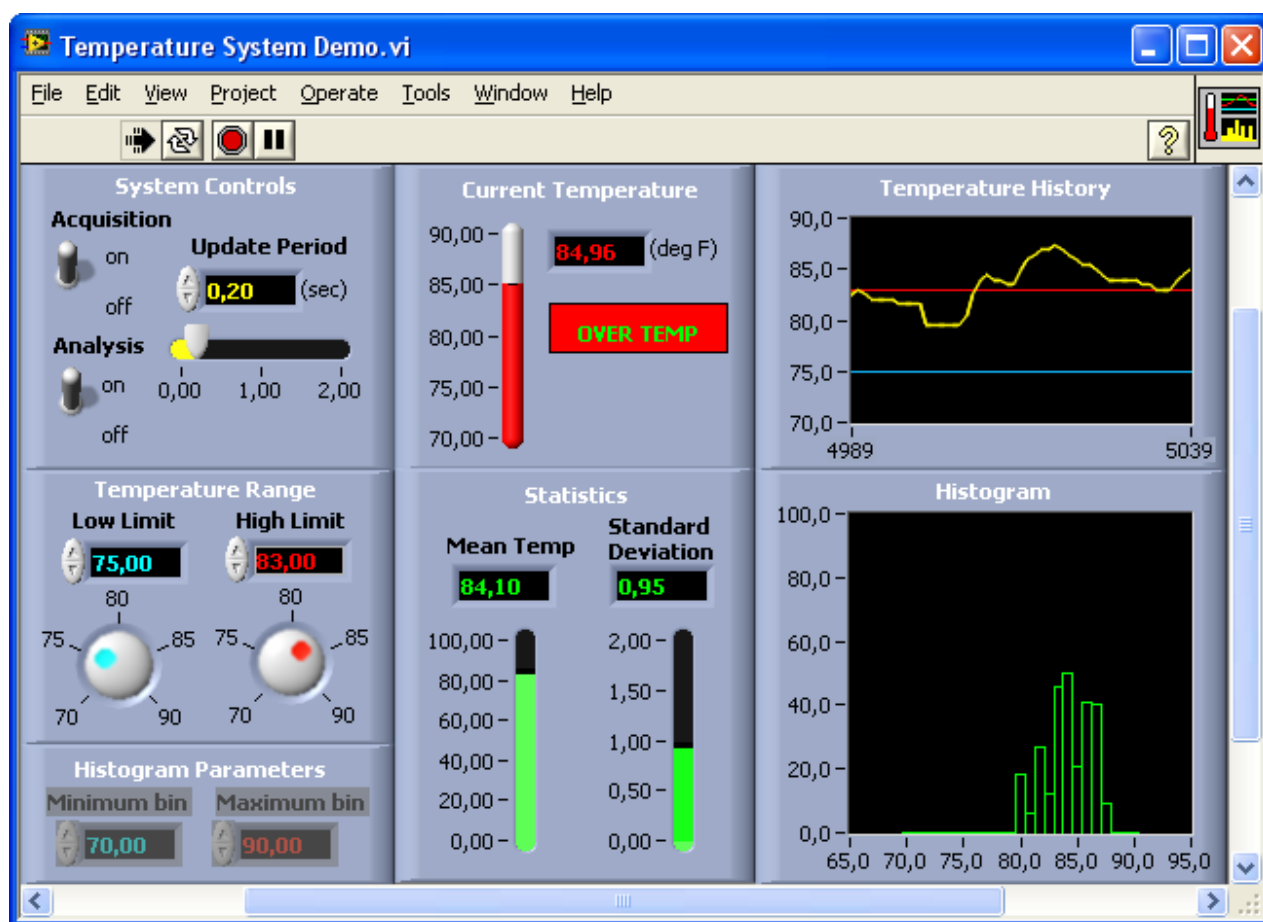
3.2.1 Čelní panel virtuálního přístroje

Grafické rozhraní k uživateli plní tutéž úlohu jako čelní panel fyzického přístroje. Pro vytváření čelního panelu je k dispozici samostatné okno. Příklad toho, jak může vypadat čelní panel virtuálního přístroje ukazuje Obrázek 3.1.

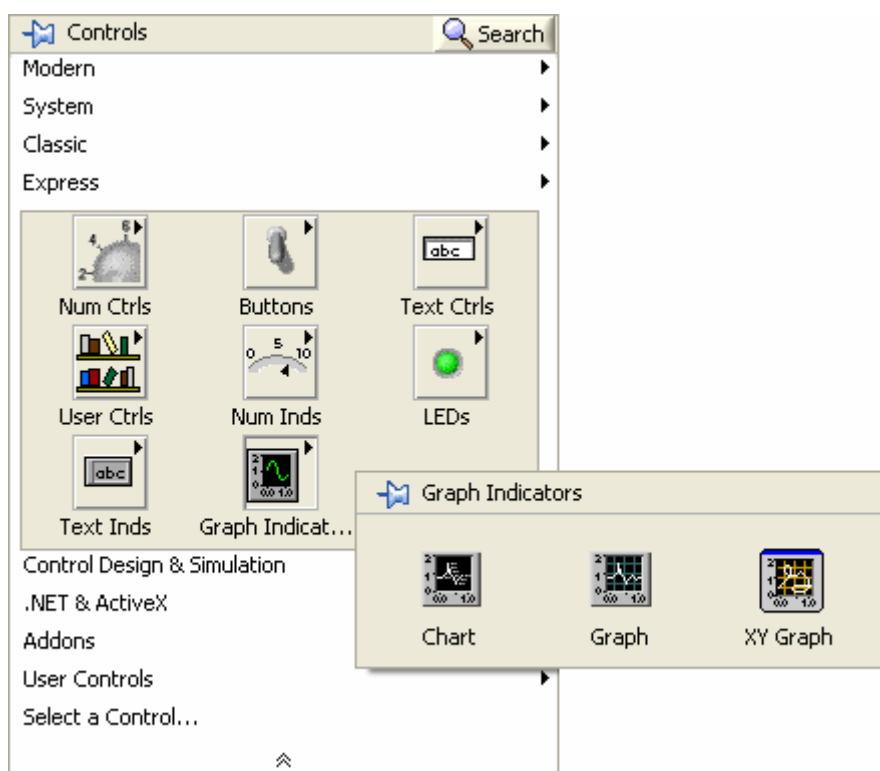
Na čelním panelu se nacházejí dva typy prvků:

- **ovládací (controls)** - simulují vstupní zařízení (v blokovém schématu jsou reprezentovány vstupními bloky) a slouží pro ovládání virtuálního přístroje - zprostředkují tedy vstup informace od uživatele do aplikace, v blokovém diagramu jsou to bloky, z nichž signál vystupuje do algoritmu.
- **indikační (indicators)** - simulují výstupní zařízení (v blokovém schématu jsou reprezentovány výstupními bloky) a slouží k indikaci stavu virtuálního přístroje a výstupu výsledků - zprostředkují tedy předávání informací směrem od aplikace k uživateli, v blokovém schématu jsou to bloky, v nichž signálové cesty končí.

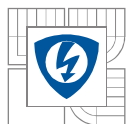
Oba typy prvků lze vybírat pomocí paletové nabídky (Obrázek 3.2), kterou lze zpřístupnit pomocí roletové nabídky **View/Show Controls Palette** nebo kliknutím na pravé tlačítko myši kdekoli v okně čelního panelu. [4]



Obrázek 3.1 Ukázka čelního panelu virtuálního přístroje



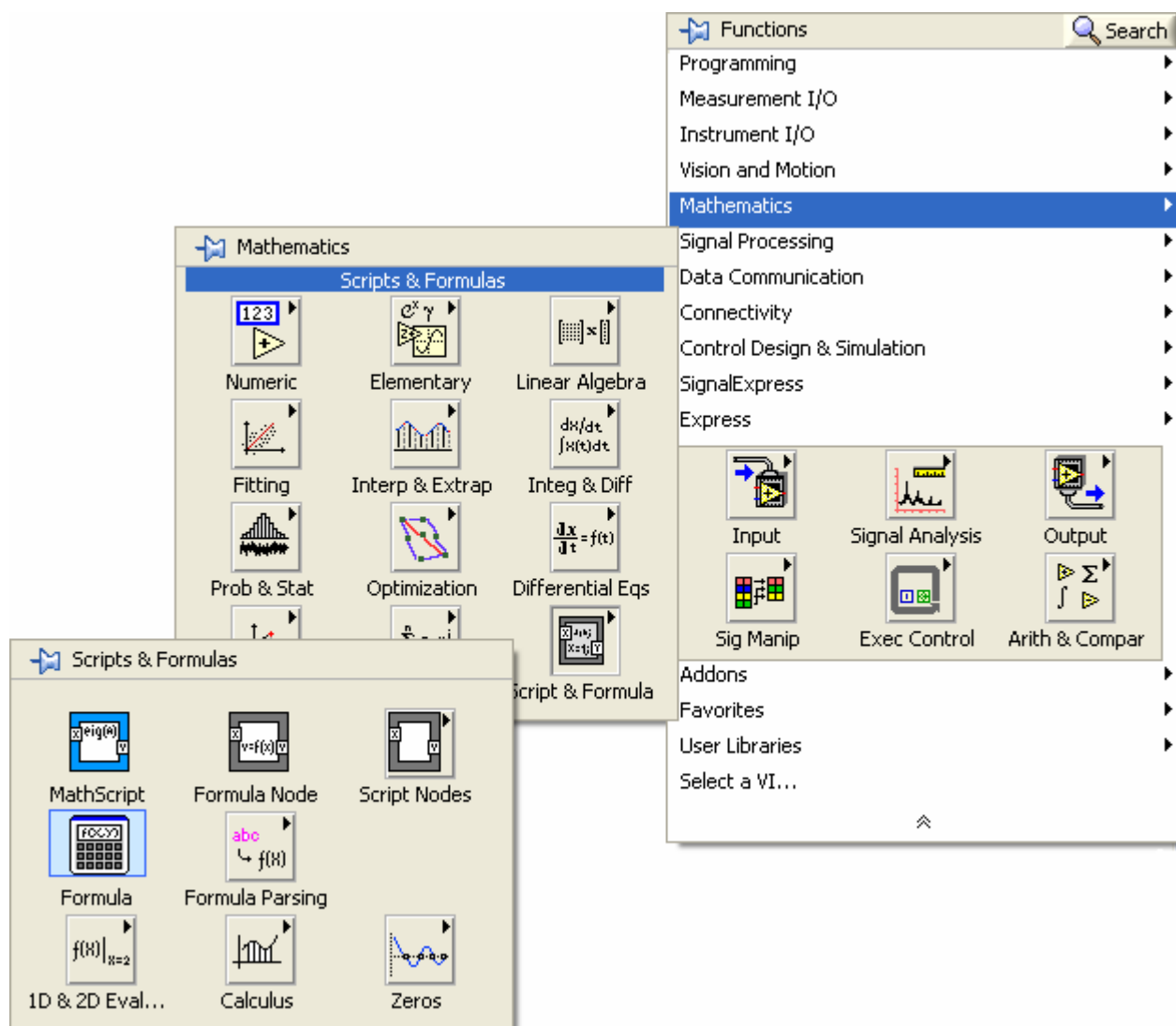
Obrázek 3.2 Ukázka paletového menu prvků čelního panelu



Po výběru prvku v paletové nabídce přidržetím levého tlačítka myši lze vytáhnout obrys vybraného prvku na čelní panel a umístit jej. Po umístění prvku v okně čelního panelu lze u něj měnit velikost, umístění, barvu atd. výběrem vhodného nástroje. Tyto akce lze nad vybraným prvkem čelního panelu provádět i dodatečně. Každý prvek čelního panelu má navíc roletové menu vyvolatelné při poloze kurzoru myši na prvku pravým tlačítkem myši a v tomto menu lze měnit další vlastnosti tohoto prvku.

3.2.2 Blokové schéma

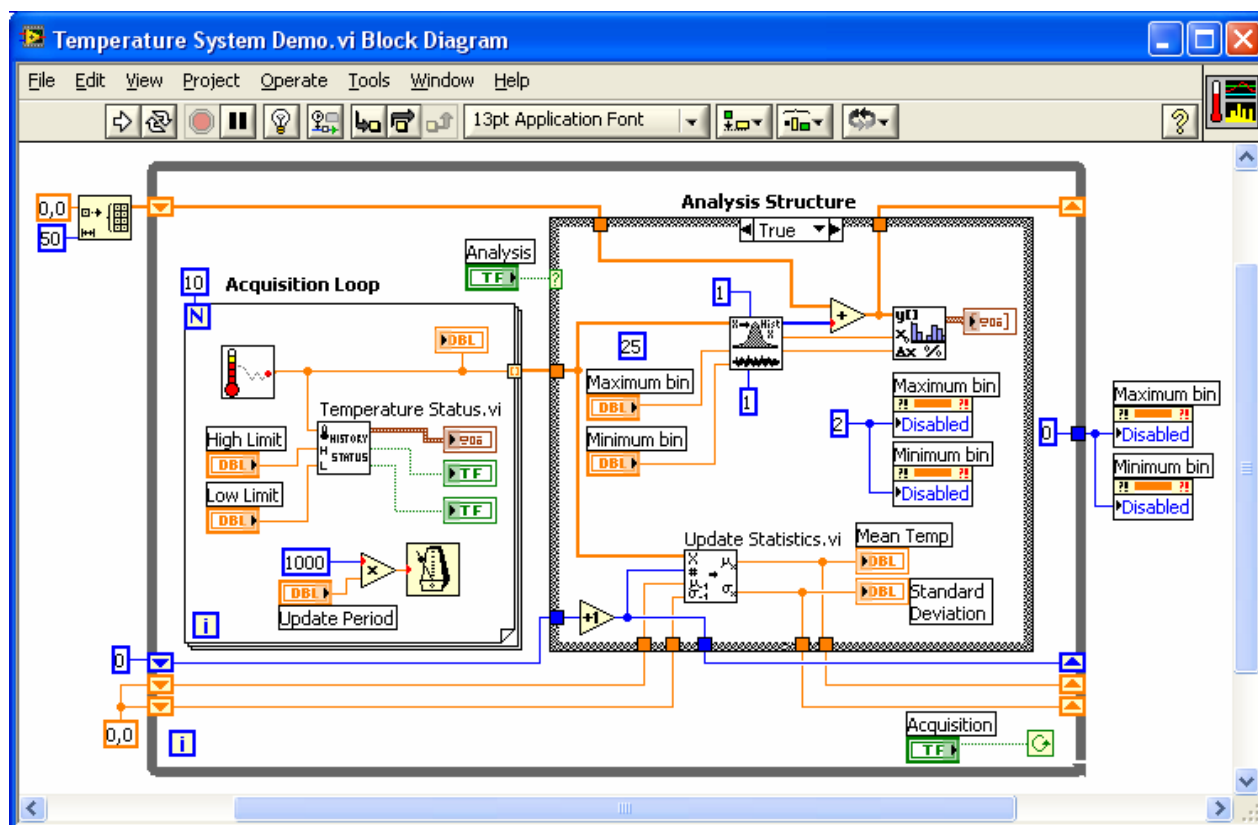
Blokové schéma je grafickým vyjádřením zdrojového kódu virtuálního přístroje. Konstruuje se propojováním jednotlivých bloků signálovými cestami. Bloky jsou tvořeny jednak **koncovými bloky** (zdrojovými a cílovými) (source and sink terminals), které jsou na panel blokového schématu umístěny automaticky při tvorbě čelního panelu a jednak **uzlovými bloky** (nodes), které reprezentují bloky zpracování signálu, které lze vybírat z paletového menu **Functions** reprezentujícího vlastně knihovnu funkcí v okně blokového diagramu. Paletové menu s funkcemi (Obrázek 3.3) lze vyvolat obdobným způsobem jako paletové menu prvků čelního panelu - buď přes menu **View/Show Functions Palette** nebo kliknutím na pravé tlačítko myši v okně blokového diagramu. [4]



Obrázek 3.3 Ukázka paletového menu funkcí a struktur blokového diagramu

Při přepínání mezi oknem čelního panelu a oknem blokového diagramu, jsou-li trvale zobrazeny paletové nabídky **Controls** nebo **Functions**, dochází automaticky k jejich přepínání, podle toho, které okno je právě aktuální. K rychlému přepínání mezi oknem čelního panelu a oknem blokového diagramu je dobré si zapamatovat horké klávesy **Ctrl E**. Je to i nejrychlejší způsob v okamžiku, kdy má uživatel otevřeno větší počet oken k nalezení odpovídajícího protějšku k oknu aktuálnímu. Další užitečnou klávesovou zkratkou je kombinace **Ctrl B**, která najednou odstraní všechny přerušené signálové cesty, uživatel tak není nucen zdlouhavě mazat každou zvlášť. Platí, že uživatel může mít k danému virtuálnímu přístroji otevřena obě okna (čelního panelu i blokového diagramu), nebo jen okno čelního panelu. Nelze mít otevřeno pouze okno blokového diagramu. Při zavření okna blokového diagramu se zavírá pouze toto okno, při zavření okna čelního panelu se zavírají automaticky okna obě. Uzlové bloky jsou v blokovém diagramu ekvivalentní příkazům, operátorům, funkcím a podprogramům klasických programovacích jazyků. Bloky se propojují signálovými cestami (wires) a kromě tohoto propojení existují ještě programové struktury (structures), nahrazující v grafické podobě standardní konstrukce užívané v programovacích jazycích (podmíněný příkaz, přepínač, cyklus...).

LabVIEW má i propojení na externí bloky textově orientovaného kódu a na textově orientované výrazy. Signálovou cestou se propojují zdrojové a cílové koncové bloky. Nelze propojit vzájemně dva zdrojové koncové bloky, ale lze spojit jeden zdrojový koncový blok k několika cílovým. Tvar a barva čáry reprezentující definovanou signálovou cestu rozlišuje typ proměnné, která prochází daným místem. Následující Obrázek 3.4 ukazuje příklad blokového schématu virtuálního přístroje, jehož čelní panel byl uveden na Obrázku 3.1. Tento virtuální přístroj je vybrán z bohaté sbírky příkladů, která je součástí standardní instalace LabVIEW. [4]



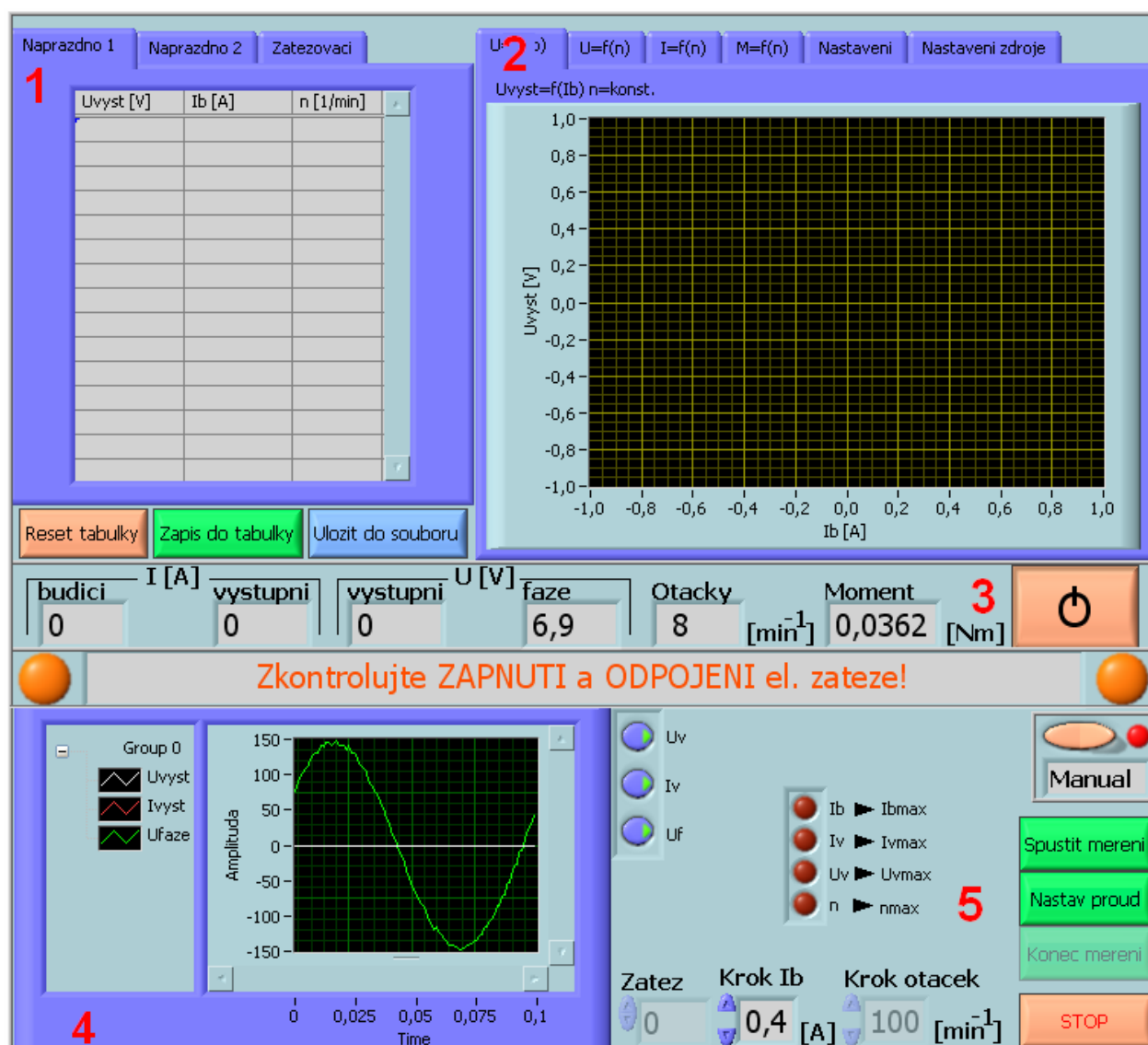
Obrázek 3.4 Příklad blokového schématu virtuálního přístroje

4 PODROBNÝ POPIS VYTVOŘENÉ APLIKACE A JEJÍ TVORBY

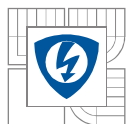
4.1 Vzhled a funkce virtuálního přístroje

V této kapitole přistupujeme k virtuálnímu přístroji jako k reálnému přístroji, který fyzicky existuje, ale nevíme co je uvnitř. Pomiňme tedy co se skrývá v blokovém schématu přístroje, což je zajímavé spíše jen z hlediska programátorského. Navíc o tom pojednává celá kapitola 4.3. Popíšeme si funkci jednotlivých řídicích prvků, tlačítek, grafů apod.

Čelní panel přístroje je vidět na Obrázku 4.1. Na první pohled mu dominují tři výstupní vizualizační prvky – označeny červenými čísly 1, 2, 4. Hlavní funkce prvku 1 je možnost volby charakteristiky, kterou chceme proměřit. Výběrem konkrétní záložky (*Naprazdno 1*, *Naprazdno 2*, *Zatezovací*) volíme i danou charakteristiku. Každá záložka prvku 1 obsahuje jednu tabulku. V té se v průběhu měření zobrazují přehledně naměřené hodnoty. V režimu *Manual*, umožňuje přístroj zapisovat do tabulky tlačítkem *Zapis do tabulky* umístěným hned pod tabulkou,



Obrázek 4.1 Čelní panel virtuálního přístroje



stejně jako tlačítko *Reset tabulky*, které maže aktuálně zobrazenou tabulku a graf této tabulce náležící. Tlačítko *Uložit do souboru* umožňuje data zobrazena v tabulce uložit do souboru s příponou *lvm* (*LabView Measurement*). Prvek 2 je na pohled podobný prvku 1. Obsahuje čtyři záložky s grafy a další dvě záložky pro nastavení. Do grafů se rovnou při probíhajícím měření vykreslují jednotlivé charakteristiky. Uživatel tak má hned dvojí zpracování měřených hodnot na očích – jak tabulku, tak graf. Záložka *Nastavení* skrývá různorodá nastavení pro chod přístroje, které si nyní popíšeme. Při pohledu na Obrázek 4.2 vidíme, že tato záložka obsahuje i pokročilou část nastavení. Jednoduché nastavení obsahuje čtyři ovládací prvky pro ochranné hodnoty a tři ovládací prvky hodnot konců měření, jejichž význam si hned vysvětlíme. Po překročení ochranných hodnot daných veličin přístroj reaguje stáhnutím budicího proudu na nulu, případně stáhnutím zátěže, probíhalo-li by zrovna měření zatěžovací charakteristiky. Poté uživatele informuje, která veličina byla překročena. Sledovanými veličinami jsou výstupní napětí, výstupní proud, budicí proud a otáčky alternátoru. Hodnoty pro konce měření určují v jakém rozsahu se mají jednotlivé charakteristiky měřit. Například pro charakteristiku $U=f(I_b)$ je stěžejní hodnota proudu I_b . Pakliže při měření dosáhne nastavené hodnoty I_{b_do} nebo vznikne požadavek na nastavení ještě většího budicího proudu než tato hodnota, měření se ukončí. Obdobně pro charakteristiku zatěžovací, dosáhne-li při měření výstupní proud hodnoty větší než I_{vyst_do} , měření se považuje za dokončené a dojde k jeho ukončení.

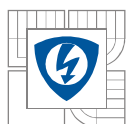


Obrázek 4.2 Panel nastavení

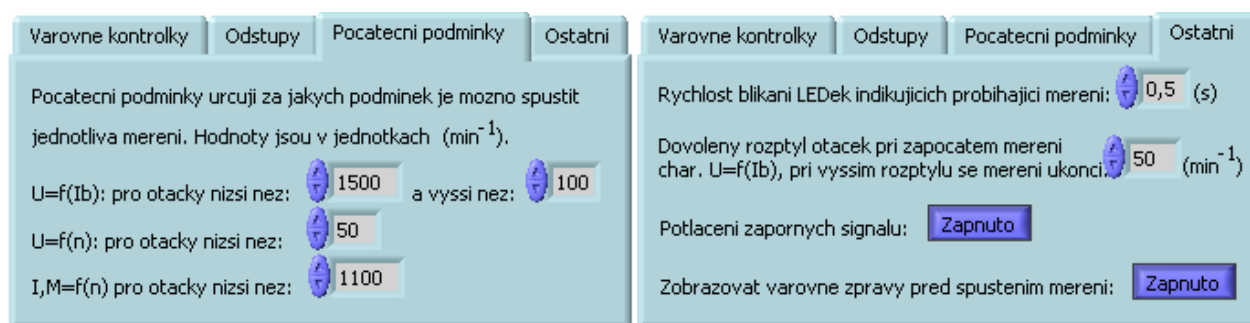
Nyní si popíšeme význam ovládacích prvků na jednotlivých záložkách pokročilého nastavení. Začneme záložkou *Varovné kontroly* viditelnou jak na Obrázku 4.2 tak na Obrázku 4.3. Na předním panelu přístroje na Obrázku 4.1 v oblasti označené číslem 5 je vidět skupina čtyř diod – varovných kontrolky. Ovládací prvky v záložce *Varovné kontroly* slouží právě k tomu, aby určily za jakých okolností se mají varovné kontroly rozsvítit. K jejich rozsvícení dojde



Obrázek 4.3 Pokročilé nastavení 1

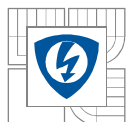


tehdy, jestliže se měřená veličina, náležící dané kontrolce, přiblíží na nastavený rozdíl k hodnotě ochranné. Např. je-li ochranná hodnota pro otáčky alternátoru nastavená na 6000 min^{-1} a rozdíl je nastaven na hodnotu 300 min^{-1} , kontrolka začne svítit, když měřená hodnota otáček alternátoru překročí hodnotu 5700 min^{-1} . Záložka *Odstupy* rovněž viditelná na Obrázku 4.3 definuje velikosti odstupů hodnot konců měření od ochranných hodnot. Důvod je prostý. Aby nedošlo k rovnosti těchto hodnot přístroj vždy při přenastavení libovolné hodnoty ochranné udržuje tento minimální nastavený odstup od příslušné hodnoty konce měření. Opět vysvětlíme na příkladu: Je-li ochranná hodnota výstupního napětí U_{max} nastavena na hodnotu např. $19,5 \text{ V}$ a odstup na hodnotu 1 V , přístroj nedovolí uživateli nastavit konečnou hodnotu měření $U_{\text{vyst_do}}$ vyšší než $18,5 \text{ V}$. Následující záložka *Pocatecni podminky* určuje za jakých podmínek lze spustit jednotlivé měření charakteristik. Toto opatření například zabraňuje nelogickému spuštění měření charakteristiky $U=f(n)$ při vysokých otáčkách alternátoru. Obvykle se tato charakteristika začíná měřit při nulových nebo velmi nízkých otáčkách. Více viz. 4.3.1.3. Poslední záložkou z pokročilého menu nastavení je záložka *Ostatni*. Z Obrázku 4.4 je patrné co v ní lze nastavit. Za vysvětlenou stojí druhý řádek. Jak víme charakteristika $U=f(I_b)$ se měří při konstantních otáčkách alternátoru. Proto je krajně nevhodné manipulovat s točivou rychlostí alternátoru v době započatého měření neboť by se tím průběh měření znehodnotil. Pakliže se tak stane a hodnota otáček se při započatém měření výrazně změní (více než se v tomto menu dovolený rozptyl nastaví) přístroj situaci vyhodnotí tak, že ukončí měření a informuje o tom uživatele. Funkce potlačení záporných signálů se vztahuje na měřené veličiny, u nichž očekáváme kladné hodnoty. Opět příklad: Při měření charakteristiky $U=f(I_b)$ očekáváme pro nulový budící proud nulové výstupní napětí. Ovšem vlivem různého rušení, může napětí nepatrně přebíhat do záporných hodnot. Což je nežádoucí třeba pro vykreslování charakteristiky do grafu. Když by se pro nulový budící proud zapsala záporná hodnota výstupního napětí, byl by tento graf zkreslen. Proto je vhodné použít funkci potlačení záporných signálů. Tím je popis jak záložky *Ostatni* tak záložky *Nastaveni* vyčerpán.

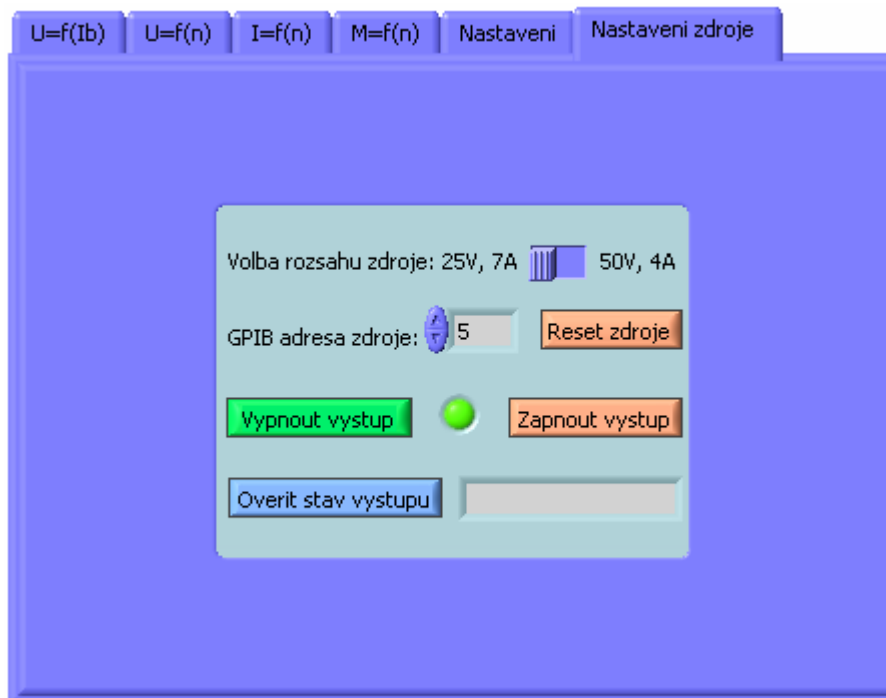


Obrázek 4.4 Pokročilé nastavení 2

Přejdeme na další záložku prvku 2 z Obrázku 4.1. Tou je záložka *Nastaveni zdroje*. Je zde vidět možnost nastavení rozsahu zdroje budícího proudu, nastavení GPIB adresy pro komunikaci se zdrojem, možnost vypnout nebo zapnout výstup zdroje či si jen ověřit jeho stav a také možnost zdroj resetovat. Pro správnou funkci přístroje je velice důležité patřičně nastavit GPIB adresu zdroje, ta je ve výchozím stavu (před zapnutím a po vypnutí přístroje) nastavena na 5. Přístroj totiž zdroj řídí právě přes GPIB rozhraní. Dioda, rovněž slouží k indikaci stavu výstupu zdroje. Svítí-li zeleně výstup je odpojen, svítí-li červeně výstup je připojen. Po stisku tlačítka *Overit stav vystupu* se tato informace zobrazí v písemné podobě na displeji nacházejícím se hned vedle tohoto tlačítka. Po stisku tlačítka *Vypnout vystup* přístroj odpojí výstup zdroje a přestane měřit hodnotu budícího proudu. Po stisku tlačítka *Zapnout vystup* vykoná přesně opačné kroky. Po stisku tlačítka *Reset zdroje* se zdroj resetuje, rovněž dojde k odpojení jeho výstupu a přestane



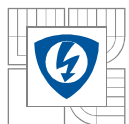
měřit hodnotu budícího proudu. Po spuštění měření libovolné charakteristiky si přístroj nastavení výstupu zdroje řídí sám a menu pro nastavení zdroje je v průběhu měření zablokováno více viz kap. 4.3.4 a 5.3.



Obrázek 4.5 Nastavení zdroje

Oblast přístroje označená na Obrázku 4.1 číslem 3 obsahuje šest číslicových ukazatelů hodnot jednotlivých měřených veličin, tlačítko pro vypnutí/zapnutí přístroje a informační panel, táhnoucí se téměř přes celou šířku přístroje, doplněn po stranách dvěma diodami. Číslicové ukazatele budícího proudu, výstupního proudu a výstupního napětí ukazují stejnosměrnou hodnotu měřené veličiny v ampérech resp. ve voltech. Ukazatel hodnoty napětí fáze ukazuje hodnotu efektivní, také ve voltech. Zbylé dva ukazatele otáček a momentu ukazují měřenou veličinu v příslušných jednotkách tj. min^{-1} resp. Nm. Informační displej uživateli zobrazuje pokyny nebo zprávy o stavu měření. Systém zpráv je doplněn o zprávy typu „OK“. Tzn., že ty se zobrazí ve zvláštním samostatném okně a je potřeba je odkliknout tlačítkem OK, čímž uživatel potvrzuje, že je bere na vědomí. Příklad takové zprávy ukazuje Obrázek 4.22. Diody po stranách informačního panelu začnou po spuštění měření blikat střídavě barvami žlutá-oranžová. To je pro uživatele varovný vizualizační prvek, aby byl více ostražitější. Navíc se o spuštěném měření zobrazuje po celou dobu měření zpráva na informačním panelu. Tlačítko vypnout/zapnout vypíná a zapíná přístroj. Ten v době vypnutí nemá žádnou funkci.

Přejdeme nyní k oblasti označené číslem 4. Tento prvek plní funkci maximálně jednoduchého vícekanálového osciloskopu, s minimální možností cokoliv nastavit. Zobrazuje časový průběh okamžitých hodnot měřených veličin. Tlačítka umístěná po pravé straně tohoto prvku, umožňují navolit, který průběh měřené veličiny chceme zrovna sledovat. Můžeme sledovat průběhy výstupního napětí (tlačítko s popiskem U_v), výstupního proudu (tlačítko s popiskem I_v) a fázového napětí (U_f). Tyto průběhy lze sledovat všechny najednou, jelikož bylo řečeno, že „osciloskop“ je vícekanálový. Jednotlivé průběhy jsou pak odlišeny barevně tak, jak ukazuje legenda na levé straně tohoto „osciloskopu“. Měřítka os obrazovky „osciloskopu“ se automaticky přizpůsobuje měřené veličině, která vykazuje největší amplitudu. Není tedy vhodné

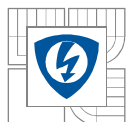


sledovat současně signál s amplitudou 1V se signálem, který má amplitudu o několik řádů vyšší, např. 100V.

Poslední nepopsanou oblastí z Obrázku 4.1 je oblast 5. Obsahuje varovné kontrolky, o kterých už zmínka padla. Dále jsou zde vidět řídicí prvky s názvy *Zatez*, *Krok Ib* a *Krok otacek*, přepínač pro volbu režimu přístroje a tlačítka *Spustit mereni*, *Nastav proud*, *Konec mereni* a *STOP*. Řídicí prvky *Krok Ib* a *Krok otacek* určují s jakým krokem budícího proudu resp. otáček se mají zapisovat data do tabulky a vykreslovat do grafu při automatickém měření patřičné charakteristiky. Pro charakteristiku $U=f(I_b)$ je podstatný prvek *Krok Ib*, pro charakteristiky $U=f(n)$ a $I, M=f(n)$ je to prvek *Krok otacek*. Přístroj se totiž při zápisu dat do tabulky, v automatickém režimu měření, orientuje podle aktuálních hodnot měřených veličin a dle nastavení kroku podstatných řídicích prvků pro dané charakteristiky. Například pro automatické měření charakteristiky $U=f(n)$ a zvoleném kroku otáček 100min^{-1} za předpokladu, že první zápis dat se provede pro nulovou hodnotu otáček, jsou data do následujících řádků tabulky zapsány vždy když hodnota otáček alternátoru dosáhne přibližně hodnoty 100, 200, 300,...tedy jsou zapisovány s přibližným krokem každých 100 otáček. Důvod zápisu dat s přibližným, a nikoliv přesným, krokem vyplývá z určitých technických omezení nastavování hodnot otáček, které jsou vždy nastavovány manuálně. Více viz. kap. 4.3.1.1 – druhý algoritmus. Obdobně funguje zápis dat i při měření charakteristik $I, M=f(n)$, okamžik zápisu dat se opět orientuje dle aktuální hodnoty otáček alternátoru.

Tlačítko *Spustit mereni* spouští automatizované měření zvolené charakteristiky, je-li přepínač nastaven na režim *Auto*. Navíc vymaže obsah aktuální tabulky a grafu. Tlačítko *Nastav proud* spouští měření pro režim *Manual*. Při měření charakteristik $U=f(n)$ nebo $I, M=f(n)$ se po stisku tlačítka *Spustit mereni* v případě *Auto* režimu nebo tlačítka *Nastav proud* v případě režimu *Manual*, nastaví budící proud zdroje na hodnotu určenou ovládacím prvkem *Ib_do*. Pak už jen přístroj čeká na příhodné okamžiky zápisu dat do tabulky a grafu v závislosti na vhodné změně otáček alternátoru. Při automatickém měření charakteristik $I, M=f(n)$ navíc vyregulovává hodnotu výstupního napětí na cca 13,5V. Více viz. kap. 4.3.1.1 - třetí algoritmus. Při manuálním měření těchto charakteristik, je regulace přenechána na uživateli. K čemuž slouží řídicí prvek *Zatez*. Ten určuje velikost řídicího napětí, kterým se externě ovládá elektronická zátěž. Více viz. kap. 5.2. Při spuštění automatického měření charakteristiky $U=f(I_b)$ přístroj postupně nastavuje hodnotu budícího proudu dle zvoleného kroku, zapisuje data do tabulky a vykresluje graf, dokud nedojde k ukončení měření. Při měření této charakteristiky v režimu *Manual*, plní tlačítko *Nastav proud* inteligentní funkci nastavování budícího proudu, dle zvoleného kroku. Po každém stisku tohoto tlačítka přístroj navýší aktuální hodnotu budícího proudu na výstupu zdroje o hodnotu určenou ovládacím prvkem *Krok Ib*. Uživatel tak například může změřit začátek charakteristiky s hrubým krokem a její konec s krokem jemným. Pakliže by případné nastavení budícího proudu mělo přesáhnout hodnotu nastavenou ovládacím prvkem *Ib_do*, přístroj místo toho nastaví proud nulový a měření ukončí. Při prvním stisku tlačítka *Nastav proud* (tzn. při spuštění měření v režimu *Manual*), v případě měření této charakteristiky, se vždy nastaví hodnota budícího proudu rovněž na nulu. V režimu *Manual* data zapisuje uživatel sám tlačítkem *Zapis do tabulky*. Případně je smazává tlačítkem *Reset tabulky* (jak graf tak tabulku).

Tlačítko *Konec mereni* bezpečně ukončuje jakékoliv započaté měření v obou režimech přístroje. Jak je vidět na Obrázku 4.1, má toto tlačítko „potlačenou“ barvu, to znamená, že je zablokováno a nelze jej stisknout. V odblokovaném stavu má stejnou barvu jako tlačítko *Nastav proud*. Tlačítko se odblokuje jen když je spuštěno měření, protože v opačném případě jeho funkce nemá význam. Stejně tak se zablokuje případně odblokuje ostatní řídicí a jiné prvky podle



toho, která záložka (charakteristika) je právě zvolena a jaký režim přístroje je nastaven. Například po přepnutí na automatický režim se zablokuje tlačítka *Reset tabulky*, *Zapis do tabulky* a *Nastav proud* vždy, bez ohledu na to jaká záložka je zvolena, nebo jestli probíhá měření či nikoliv, jelikož v režimu *Auto* je jejich funkce nepotřebná. Ostatní řídicí prvky jako *Zatez*, *Krok Ib*, *Krok otacek* aj. se blokují/odblokuje v závislosti na zvolené záložce. Inteligentní systém blokování tlačítek, ovládacích a jiných prvků přehledně ukazuje následující tabulka:

Tabulka 4.1 Inteligentní systém blokace prvků čelního panelu přístroje

Zvolená záložka:	Naprazdno 1		Naprazdno 2		Zatezovací	
Režim:	Auto	Manual	Auto	Manual	Auto	Manual
<i>Reset tabulky</i>	x		x		x	
<i>Zapis do tabulky</i>	x		x		x	
<i>Uložit do souboru</i>	m		m		m	
"Auto/Manual"	m	m	m	m	m	m
<i>Spustit mereni</i>	m		m		m	
<i>Nastav proud</i>	x		x	m	x	m
<i>Konec mereni</i>	x, o	x, o	x, o	x, o	x, o	x, o
<i>Zatez</i>	x	x	x	x	x	
<i>Krok Ib</i>	m		x	x	x	x
<i>Krok otacek</i>	x	x	m	x	m	x
záložky	i	i	i	i	i	i
DAM**	v	v	v	v	v	v
<i>Nastavení zdroje</i>	m	m	m	m	m	m

* Tento prvek je přepínač plnicí funkci přepínání přístroje na režim Auto/Manual.

** Zkratka pro displej aktuálního měření.

x - Prvek je blokován už v době zapnutí přístroje (aniž by se spustilo měření).

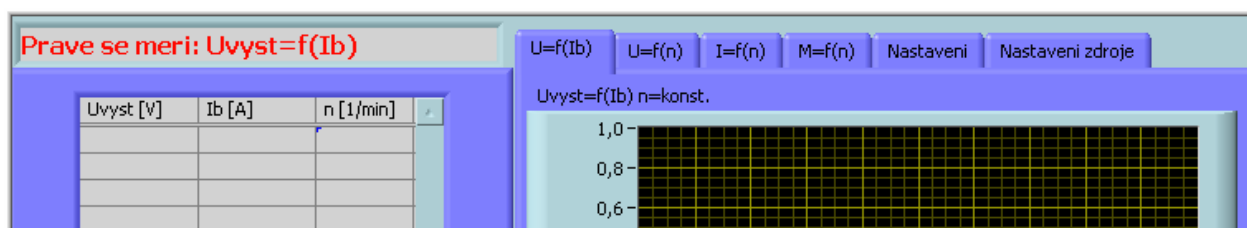
m - Prvek je zablokován až po spuštění měření, po skončení je odblokován.

o - Prvek je po spuštění měření odblokován a po jeho ukončení zablokován.

i - Prvek je po spuštění měření skryt - invisible a po skončení odkryt zpět.

v - Prvek je po spuštění měření odkryt - visible a po skončení skryt zpět.

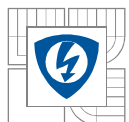
Inteligentní systém blokace tlačítek přispívá k tomu, aby byl přístroj co nejvíc odolný proti chybnému lidskému faktoru. Více viz. kap. 4.3.2. Displej aktuálního měření se ukáže vždy jen po spuštění měření a pak zase zmizí. Protože na Obrázku 4.1 není vidět, ukazuje nám ho následující obrázek:



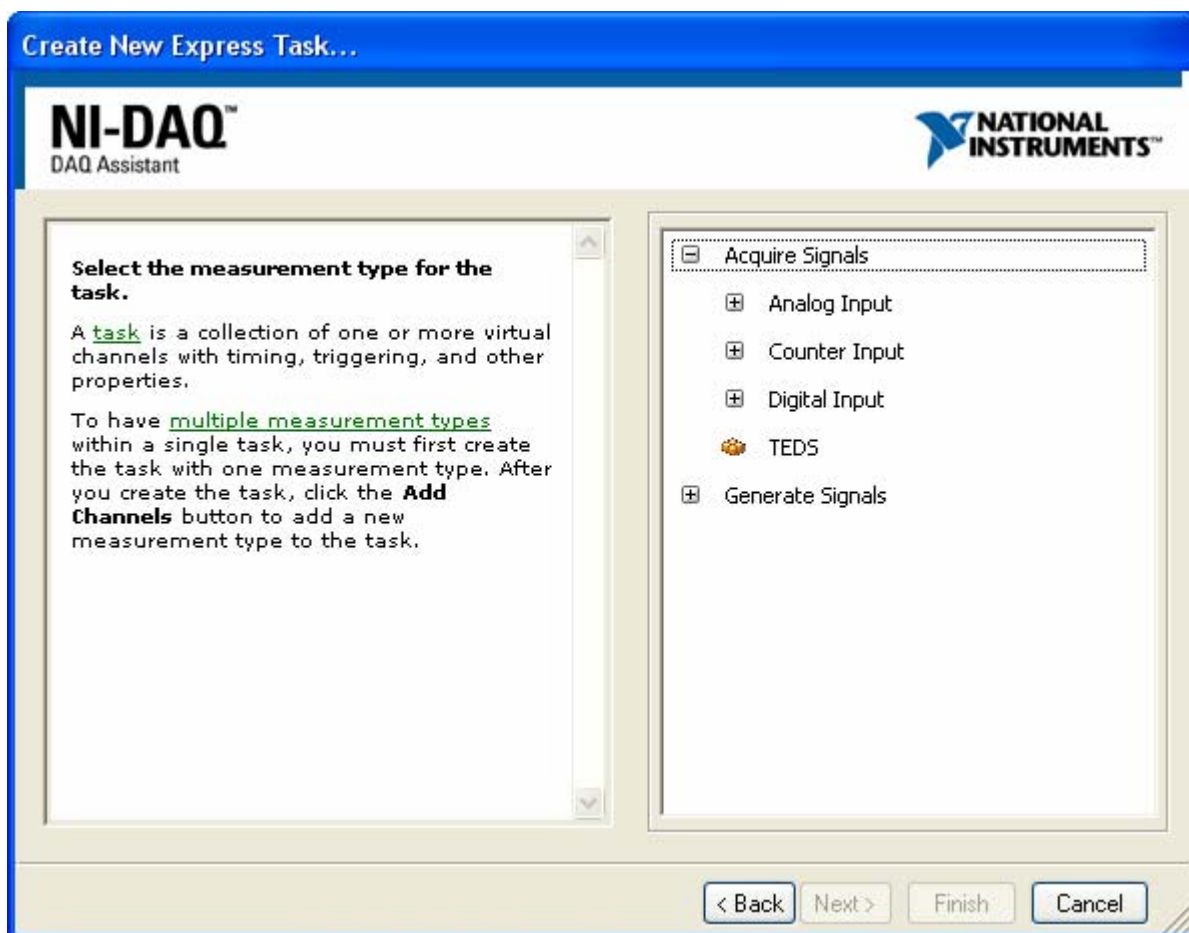
Obrázek 4.6 Displej aktuálního měření

4.2 Nastavení DAQ Assistanta

Počínaje touto kapitolou se již budeme zabývat co se vlastně za čelním panelem virtuálního měřicího přístroje skrývá, jak je vše řešeno a co je k tomu zapotřebí. K samotnému měření pomocí LabVIEW je potřeba mít měřicí kartu a desktop nebo laptop s příslušným softwarem. Software zahrnuje LabVIEW (v našem případě verze 8.6) a drivers potřebné ke komunikaci

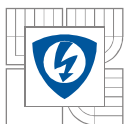


LabVIEW s měřicí kartou. V LabVIEW se pak projeví nainstalované drivery mimo jiné tím, že se v paletovém menu funkcí objeví ikonka DAQ Assistant v sekci Input. Po vložení DAQ Asistanta do okna blokového schématu vytvářené aplikace se objeví okno s nabídkou nastavení typu měření pro danou úlohu (viz.Obrázek 4.7). Volí se zde, které konkrétní typy vstupů případně výstupů (analogové, digitální) měřicí karty se budou využívat. Pro naše potřeby plně postačila volba analogových vstupů a jednoho analogového výstupu.



Obrázek 4.7 Úvodní okno DAQ Asistanta

Po zvolení požadovaných vstupů a výstupu se objeví další okno (viz.Obrázek 4.8), které je již přímo nastavením DAQ Asistanta. Nezobrazí se pouze v případě, že k počítači není připojena měřicí karta (nebo je třeba vypnutá). Místo toho se ukáže okno s hlášením, že nebylo nalezeno podporované zařízení. Pro naše účely je využita měřicí karta připojující se k PC přes USB vstup, konkrétně typ NI USB-6251 viz. kap. 5.1. V okně DAQ Asistanta v záložce *Express Task* lze nastavit, mimo jiné, jakým způsobem se má měřit napětí na vstupech karty a vzorkovací frekvenci, případně lze dodatečně přidat další vstupní kanál. Napětí je možno měřit několika způsoby tak, jak ukazuje Tabulka 4.2. V našem vytvořeném přístroji je zvoleno diferenční měření jednotlivých veličin, protože tento způsob měření je odolnější vůči rušení než klasické měření proti zemi. Nastavení vzorkovací frekvence je obzvlášť důležité, měří-li se např. příliš rychle se měnící aperiodické signály nebo periodické signály vysokých frekvencí. Při volbě vzorkovací frekvence se očekává, že uživatel měřicí karty ví, co by chtěl pomocí ní měřit. Tudíž by měl způsobit tomu i vhodně nastavit tuto frekvenci. Ze Shannon-Kotělnikovova teorému, který říká, že vzorkovací frekvence musí být minimálně 2x vyššího kmitočtu než kmitočet nejvyšší

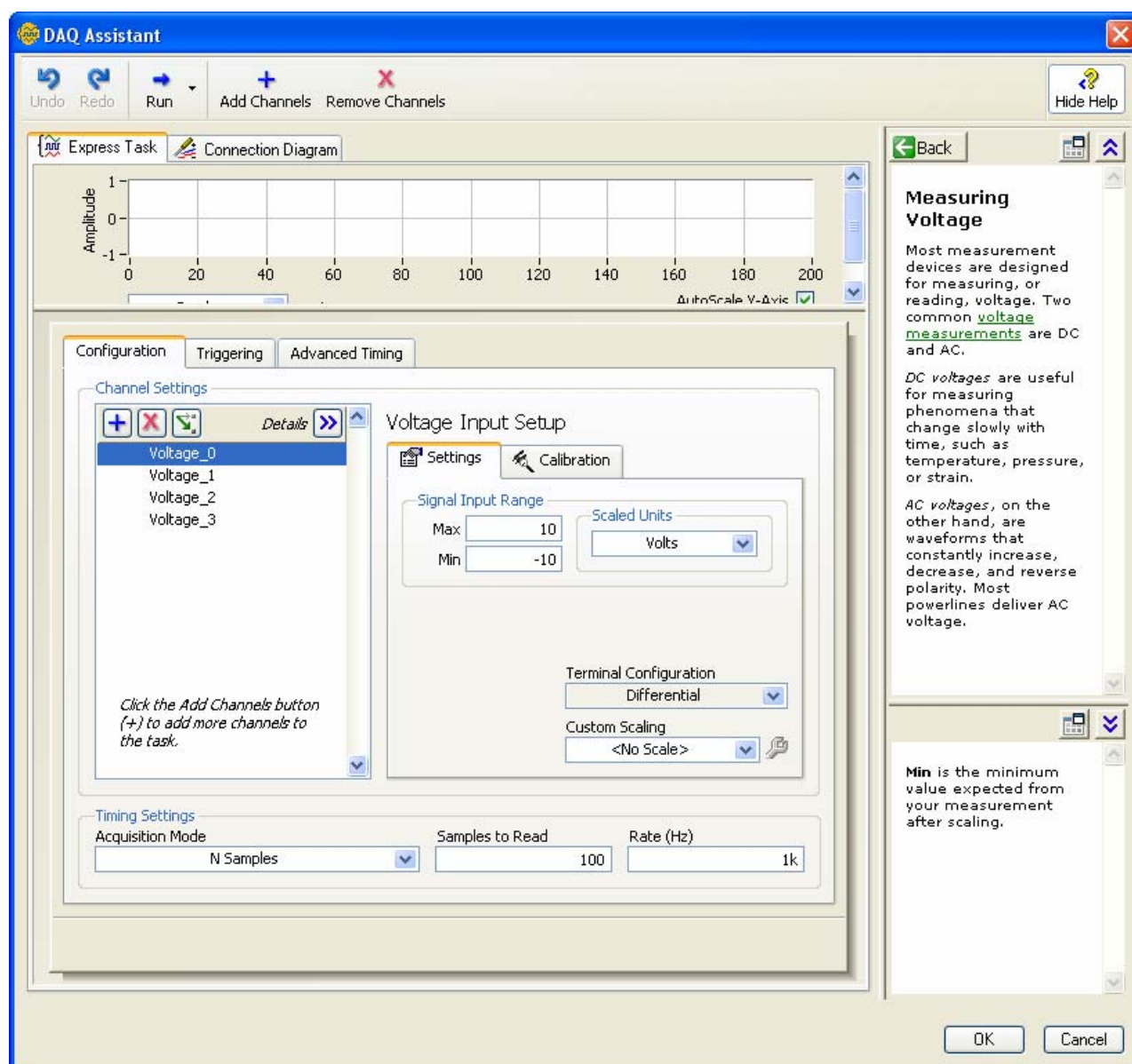


Tabulka 4.2 Konfigurace analogových vstupů [10]

AI Ground-Reference Setting*	Floating Signal Sources (Not Connected to Building Ground)	Ground-Referenced Signal Sources†
Differential		
Non-Referenced Single-Ended (NRSE)		
Referenced Single-Ended (RSE)		
* Refer to the Analog Input Ground-Reference Settings section for descriptions of the RSE, NRSE, and DIFF modes and software considerations. † Refer to the Connecting Ground-Referenced Signal Sources section for more information.		

harmonické složky vzorkovaného signálu, jasně vyplývá jak by se měla vzorkovací frekvence nastavit. Pro naše potřeby stačí nastavit malou vzorkovací frekvenci, jelikož všechny měřené veličiny až na jednu výjimku jsou stejnosměrné. Tedy je ponecháno výchozího nastavení DAQ Asistenta. Po nastavení všeho potřebného je možno tlačítkem *Run* zběžně ověřit, zda-li jsou na zvolených vstupech očekávaná napětí. Jejich průběh lze sledovat v horní části okna nastavení DAQ Asistenta. Pokud by se zobrazovaly např. nějaké šумы, příčin může být spousta, ale je dobré ověřit, jsou-li vstupy správně zapojeny. K tomuto účelu slouží záložka *Connection Diagram*, která jasně a názorně ukazuje jak mají být vstupy správně zapojeny při daném nastavení. Důležité je si rovněž uvědomit, že napěťový rozsah vstupů (a rovněž i výstupů) měřicí karty je pouze $\pm 10\text{V}$, takže signály s vyšším potenciálem napětí je nutno na vstupy karty připojit přes vhodný napěťový dělič.

Po odklepnutí tlačítka *OK* je DAQ Assistant připraven k použití a stává se součástí blokového schématu vznikajícího virtuálního měřicího přístroje. Nyní přejdeme k popisu jednotlivých struktur blokového schématu vytvořené aplikace.

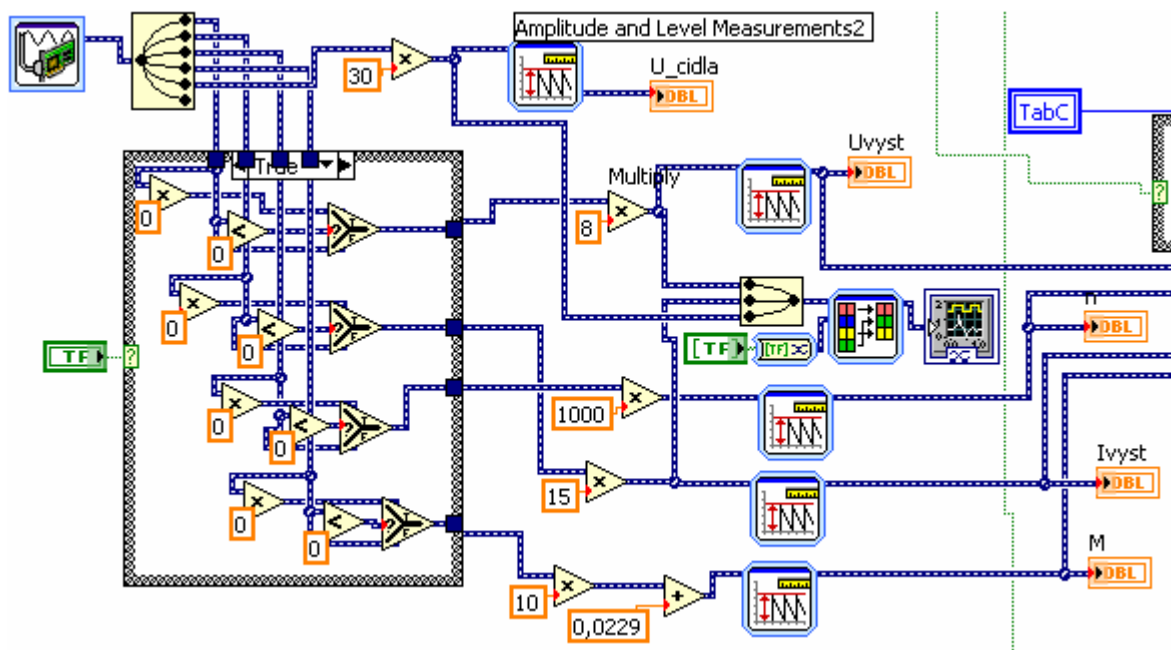


Obrázek 4.8 Nastavení DAQ Assistantu

4.3 Jednotlivé části a struktury zdrojového kódu

V každém blokovém schématu, jenž představuje zdrojový kód automatizované aplikace se musí nacházet různé bloky, spoje, konstanty a jiné objekty potřebné k jejímu vytvoření. Jedním z nepostradatelných prvků je tzv. *While Loop* struktura. V tomto grafickém prostředí symbolizuje podmínku, opakovaně se vykonávej dokud nebude zmáčknuto tlačítko *STOP*. Tato smyčka zajistí, že po spuštění aplikace běží stále a neprovede se pouze jedenkrát, protože to by bylo bezúčelné. Tlačítko *STOP* je s touto smyčkou pevně spjato a bez jeho přítomnosti nelze ani smyčku použít. LabVIEW by okamžitě indikovalo chybu. Tlačítko *STOP* slouží k regulárnímu zastavení běhu aplikace a tedy k přerušení její činnosti. Jedinou další regulární možností jak přerušit běh aplikace, je zmáčknout ikonku *Abort Execution* v horní liště programu LabVIEW. Také se může stát, že aplikaci přeruší chybové hlášení, čemuž se samozřejmě už při návrhu snažíme předcházet, čili se této možnosti co nejlépe vyhýbat. Chybové hlášení se může objevit například, když zapomeneme zapnout měřicí kartu. Vznikne tím problém komunikace bloku

DAQ Assistant s měřicí kartou. Tento blok v podstatě obstarává rozhraní mezi měřicí kartou a vytvořenou aplikací. Čili vhodně zpracovaná data získané z měřicí karty posílá na svůj výstup, kde je zpracovávají další bloky. Na výstupu bloku DAQ Assistant jsou v podstatě jen navzorkované údaje o signálech přivedených na jednotlivé kanály měřicí karty. Z DAQ Assistantu je vyvedeno spojení s blokem *Split Signals*, který rozdělí data vyvedena z DAQ Assistantu na jednotlivé kanály, aby se tyto mohly dále zpracovávat samostatně a dle potřeby. Každý kanál je přiveden do bloku *Amplitude and Level Measurements*, což je blok měřící hodnotu napětí. Je to jakýsi virtuální voltmetr, jehož výstupem je hodnota napětí fyzicky přítomná **na svorkách jednotlivých kanálů měřicí karty**. Tento blok dokáže měřit i RMS hodnoty a jiné. Jeho nastavení tedy závisí na typu signálu, který s ním chceme měřit. Ovšem měřicí rozsah na vstupech karty, jak již bylo řečeno, je omezen hodnotami ± 10 Voltů. Takže jsou-li signály přivedeny na vstupy karty přes dělič nebo jiným způsobem upraveny, je potřeba toto mít na paměti a vhodně to ve vytvářené aplikaci zohlednit tak, aby ukazatelé správně zobrazovali hodnotu, která by se dala opravdu fyzicky naměřit například na výstupních svorkách alternátoru. K těmto účelům slouží bloky *Multiply* (případně *Formula*). Ty přepočítají hodnotu napětí na jednotlivých kanálech měřicí karty na hodnotu odpovídající měřené veličině. Pro jednotlivé měřené veličiny jsou přepočítávací konstanty následující: Pro výstupní napětí 8, jelikož napětí přiváděné do měřicí karty vede z monitorovacího výstupu napětí z elektronické zátěže, který skutečné napětí právě 8x dělí. Pro moment je konstanta 10 (plus nastavení offsetu 0,0229Nm), pro otáčky 1000, jelikož zvolený rozsah otáčkoměru je 100mV-6000mV~100rpm-6000rpm. Pro výstupní proud je konstanta 15, neboť ten je rovněž měřen z elektronické zátěže tentokrát z monitorovacího výstupu proudu. Pro napětí fáze je konstanta 30. Budící proud je měřen přímo ze zdroje budícího proudu, který s vytvořenou aplikací komunikuje přes GPIB rozhraní více viz. kap. 4.3.4 a 5.3. Za *Multiply* bloky následují číselné ukazatele (*Numeric Indicator*), což jsou koncové bloky, které se umísťují na čelním panelu (viz. Obrázek 4.1 oblast 3) a plní funkci digitálních displejů. Bloky *DAQ Assistant*, *Split Signals*, *Multiply*, *Amplitude and Level Measurements*, *Numeric Indicator* a jiné jsou vidět na Obrázku 4.9.

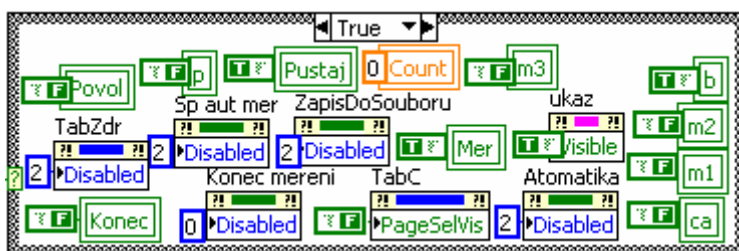


Obrázek 4.9 Různé bloky

4.3.1 Funkce hlavních algoritmů a struktur pro automatické a manuální měření

Na Obrázku 4.10 je *Case* struktura, která se vykoná vždy, když dojde ke spuštění automatizovaného měření bez ohledu na to, kterou charakteristiku právě chceme měřit. Tzn. že tato *Case* struktura je společná pro všechny tři automatizované algoritmy. Spustí se po stisku tlačítka *Spustit mereni*, jsou-li splněny počáteční podmínky pro měření (viz. kap. 4.3.1.3). V popisu jednotlivých algoritmů už tuto strukturu nebudeme zmiňovat, ale její úloha je důležitá. Je to totiž hlavní inicializační struktura. Jak se dozvíme později každý algoritmus zároveň ještě spustí další vedlejší inicializační strukturu, která je „šitá na míru“ jednotlivým algoritmům. Nyní popisovaná struktura provede ty inicializační kroky, jenž jsou algoritmům společné. Což obnáší následující:

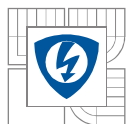
V první řadě se pomocí bloků *Property Node* změní určité vlastnosti tlačítek a jiných prvků na čelním panelu měřicího přístroje. Konkrétně pro prvky *TabZdr*, *Sp aut mer*, *ZapisDoSouboru*, *Automatika* se nastaví vlastnost *Disabled* na *Disabled & Grayed* a pro *Konec mereni* se nastaví naopak na *Enabled*. Tomu odpovídají konstanty přivedené do jednotlivých *Property Node*ů (2=*Disabled & Grayed*, 0=*Enabled*). Těmto prvkům (krom *TabZdr*, což je prvek typu *Tab Control*) náleží na čelním panelu tlačítka *Spustit mereni*, *Ulozit do souboru*, přepínač „beze jména“ a *Konec mereni* všechny viditelné na Obrázku 4.1. Zbylé dva *Property Node* bloky *TabC* a *ukaz* nastavují vlastnost *Visible* pro displej aktuálního měření (viz. Obrázek 4.6) a pro záložky prvku *Tab Control* viz. rovněž Obrázek 4.1. Zde *True*=*Visible*, *False*=*Invisible*. Úkolem všech zmíněných *Property Node*ů je zajištění odolnosti virtuálního přístroje proti lidským chybám, více viz. kap. 4.3.2. Nastavení ostatních proměnných a příznaků je více méně nutné pro korektní spuštění automatizovaného měření. Například příznak *Pustaj* indikuje spuštěné automatizované měření, proměnná *Mer* spouští *Měřicí* strukturu (viz. kap. 4.3.4), příznaky *b*, *ca*, *m1*, *m2*, *m3* hrají roli při měření manuálním (viz. kap. 4.3.1.2), proto se při spuštění automatizovaného měření nastavují na hodnoty, které jsou výchozími pro měření manuální.



Obrázek 4.10 Inicializační *Case* struktura

4.3.1.1 Algoritmy pro automatizované měření

Algoritmus pro měření první charakteristiky naprázdno tj. $U=f(I_b)$ pro $n=\text{konst.}$ vychází z požadavku na toto měření - zapsat hodnoty měřených veličin do tabulky a vykreslit do grafu požadovanou charakteristiku v závislosti na zvoleném kroku budícího proudu. Vše co je v tomto algoritmu integrováno je vidět na Obrázku 4.11 v *Case* struktuře nazvané *Naprazdno 1*. Algoritmus je kompletně aktivní pouze v případě kdy jsou po zmáčknutí tlačítka *Spustit mereni* (umístěném na čelním panelu přístroje viz. Obrázek 4.1) splněny počáteční podmínky (viz. kap. 4.3.1.3) pro dané měření. V případě kdy je zvoleno manuální měření je aktivní pouze část, která zapisuje hodnoty měřených veličin. Zápisem hodnot měřených veličin se rozumí jak zápis do



tabulky, tak zápis do grafu, tak uložení měřených hodnot do datového pole (proměnná *Array*). Je-li algoritmus aktivní v plné míře, čili po spuštění automatického měření pracuje následovně:

Nejprve zobrazí varovnou zprávu (je-li tato možnost nastavena - viz. Obrázek 4.4), poté nastaví budící proud na nulu, zablokuje ovládací ukazatel kroku proudu I_b , nastaví zobrazení příslušného grafu (v případě, že by z předchozích měření bylo nastaveno zobrazení jiného, aby tento nebyl pro uživatele matoucí), a vynuluje datové pole. Navíc se do pomocné proměnné *otOK* uloží aktuální hodnota otáček, význam této proměnné bude vysvětlen později. Tímto je algoritmus inicializován. „Otěže do rukou“ přebírá *Flat Sequence* struktura skládající se ze tří „filmových polí“. První pole vyhodnocuje jak povolení k zápisu měřených hodnot a povolení inkrementování proměnné *Ib_poz*, tak podmínky pro ukončení probíhajícího měření. Druhé pole v případě povolení ukládá měřené hodnoty do tabulky a do datového pole a vykresluje graf. Třetí pole inkrementuje hodnotu požadovaného budícího proudu (*Ib_poz*) v závislosti na zvoleném kroku a posílá příkaz k nastavení budícího proudu (přes proměnnou *Nastav* více viz.kap. 4.3.4 – *Nastavovací* struktura), nepřesahuje-li požadovaná hodnota budícího proudu hodnotu nastavenou pro konec měření (proměnná *Ib_do*). Nyní se vraťme k prvnímu poli a podmínkám pro povolení zápisu měřených hodnot. Povolení ve formě proměnné *Povol* nabývá hodnotu *True*, jsou-li podmínky splněny nebo *False* pakliže splněny nejsou. Pro jeho aktivaci (nabytí hodnoty *True*) musí být splněny čtyři nezbytné podmínky:

- Požadovaný (*Ib_poz*) a nastavený (*Ib_ind*) budící proud v absolutním rozdílu nesmí být větší než 0,01, zároveň
- požadovaná hodnota proudu I_b musí být menší nebo rovna hodnotě pro konec měření, zároveň
- proměnná *Pustaj* musí mít hodnotu *True*, a zároveň
- proměnná *Nastav* hodnotu *False*.

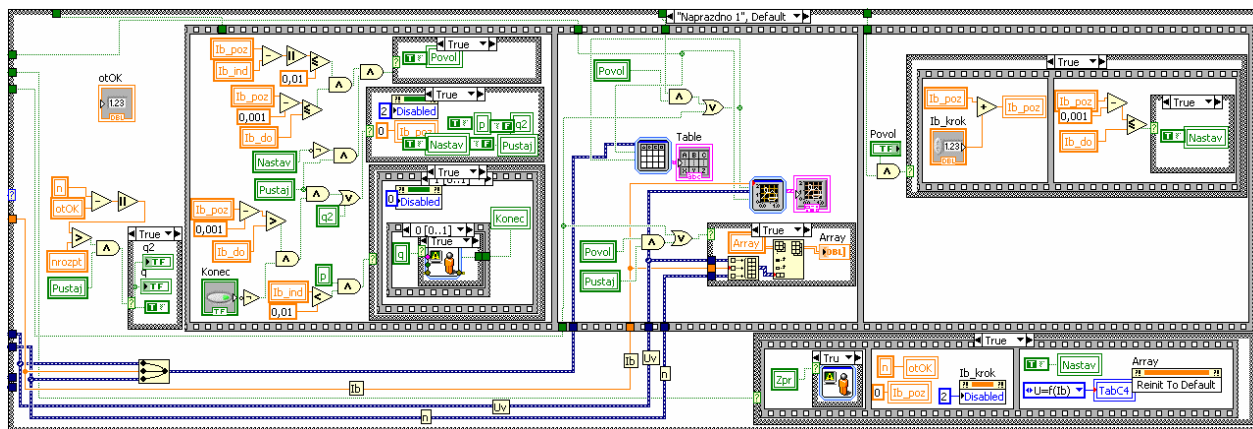
Tím, že proměnná *Nastav* nabývá hodnotu *False* je potvrzeno, že proběhla sekvence příkazů pro nastavení zdroje (více viz. kap. 4.3.4) potažmo pro nastavení budícího proudu. Pakliže se provede tato sekvence není zaručeno, že požadovaný proud na výstupu zdroje se tam objeví ihned po jejím dokončení, proto je přítomna i podmínka a). Ta zaručuje, povolení zápisu, opravdu až tehdy, je-li na výstupu zdroje naměřen požadovaný budící proud. Tato podmínka je nezbytně nutná, jelikož zajišťuje jakousi „synchronizaci“ rychlosti vykonávaného algoritmu s rychlostí reakce zdroje na podněty ke změně výstupního proudu. Kdyby zde tato podmínka nebyla a ostatní by byly splněny algoritmus by neřízeně zapisoval měřené hodnoty bez ohledu na to, jestli zdroj stíhá nastavovat požadovaný výstup nebo ne. Takže by se snadno mohlo stát, že pro jednu hodnotu budícího proudu by bylo zapsáno několik řádků dat namísto jednoho. Podmínka c) v podobě proměnné *Pustaj* je příznakem regulérního spuštění automatizovaného měření, právě tento příznak určuje, je-li aktivní celá *Case* struktura *Naprazdno 1* či nikoliv. Jinými slovy algoritmu říká, probíhá-li automatické měření či nikoliv. Podmínka b) je opakem jedné z podmínek pro ukončení měření. Ty jsou následující:

- požadovaná hodnota proudu I_b musí být větší než hodnota pro konec měření (opak podmínky b)), zároveň
- příznak *Pustaj* musí mít hodnotu *True* (aby se měření „neukončovalo“ když ani nezačalo)

anebo

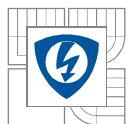
- příznak *q2* má hodnotu *True*.

Když algoritmus vyhodnotí podmínku e) nebo g) jako splněnou (samozřejmě za předpokladu splnění podmínky f)) vykonají se objektové příkazy umístěné v prostřední *Case* struktuře v prvním „filmovém poli“. Mimo jiné se budící proud vynuluje, příznak *Pustaj* a *q2* se nastaví na *False* a proměnná *p* naopak na *True*. Proměnná *p* společně s podmínkou $I_{b_ind} < 0,01$ má význam při povolení vykonání třetí *Case* struktury. Tato proměnná v podstatě slouží jako štafetový kolík, jenž předává štafetu dál, další *Case* struktuře. Propojení a souvislost prostřední a poslední *Case* struktury při ukončování měření je tedy patrná, důvod tohoto dvoufázového ukončení hned vysvětlíme. Aby byl uživatel nepřehlédnutelně informován, že se měření dokončilo, je použit objekt *Display Message to User*. Tento objekt však způsobuje, v okamžiku kdy je aktivován, dočasné „zamrznutí“ celého programu ve formě čekání na odkliknutí tlačítka *OK* po té, co na obrazovce zobrazí zprávu pro uživatele. Při aktivování tohoto objektu společně s ostatními příkazy a objekty pro ukončení měření v jedné *Case* struktuře by tak mohlo dojít k situaci, kdy se nejprve zobrazí zpráva pro uživatele a teprve po odkliknutí tlačítka *OK* by se vykonal zbytek příkazů. Takže v době kdy by zpráva informovala uživatele o ukončeném měření, by tomu tak ve skutečnosti zcela nebylo. Proto je nejprve v prostřední *Case* struktuře vykonána nutná sada příkazů pro ukončení měření. Což je v první řadě požadavek stáhnutí budícího proudu zdroje na nulu. Teprve až je na výstupu zdroje opravdu fyzicky nulový proud (ověřováno již zmiňovanou podmínkou $I_{b_ind} < 0,01$) vykoná se zbylá sada příkazů, pro ukončení měření, nacházející se ve třetí *Case* struktuře. A to tak, že jeden z posledních příkazů bude právě diskutované zobrazování zprávy pro uživatele. Poslední příkaz je vynulování proměnné *q*, která se nesmí vynulovat před zobrazením zprávy, protože určuje jaká zpráva bude zobrazena (měření ukončeno regulérně nebo přenastavením hodnoty otáček nad povolenou hodnotu v průběhu započatého měření). Zobrazení



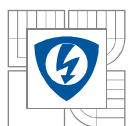
Obrázek 4.11 První algoritmus

zprávy až po vykonání výše zmíněné zbylé sady příkazů je zajištěno *Stacked Sequence* strukturou, což je obdoba *Flat Sequence* struktury. Rozdíl je pouze ve zobrazení jednotlivých „filmových polí“, kdy u první struktury je vidět vždy jen jedno (zvolené) pole, kdežto u druhé všechny. Nyní objasníme funkci proměnných *otOK*, *q* a *q2*. V hlavní *Case* struktuře *Naprazdno 1*, popisovaného algoritmu se vlevo, mimo *Flat Sequence* strukturu, nachází malá *Case* struktura se dvěma podmínkami. Dohromady toto uskupení tvoří „hlídače“ otáček. Je hlídáno, jestli se hodnota otáček alternátoru *n* neodchyluje od hodnoty otáček nastavených při započetí měření (uložených do proměnné *otOK*) víc než je dovolený rozptyl otáček (lze nastavit viz. Obrázek 4.4, představován proměnnou *nrozpt*). Charakteristika $U=f(I_b)$ se vždy měří pro konstantní otáčky. Jelikož se ovšem otáčky nastavují manuálně, takže to program nemůže ovlivnit, může dojít k přenastavení hodnoty otáček alternátoru i v průběhu měření. Ať už z důvodu neznalosti,



úmyslně nebo neúmyslně. V každém případě „hlídač“ otáček, když už to nemůže ovlivnit, může to aspoň vyhodnotit. A vyhodnocuje to tak, že skrze proměnnou $q2$ pošle příkaz k ukončení měření (nastaví jí na hodnotu *True*), jelikož by výraznou změnou otáček byla vypovídající hodnota měření znehodnocena. Proměnné $q2$ i q se nastaví na hodnotu *True*. Tím je splněna podmínka g) pro ukončení měření, takže se začne ukončovat. Proměnná q nastavená do *True* určuje zobrazení zprávy o ukončeném měření z důvodu nepovolené změny otáček. V opačném případě se při ukončení měření zobrazí zpráva „*Měření hotovo! Budící proud dosahl konečné hodnoty, dle potřeby přenastavte otáčky!*“. Jednoduše řečeno algoritmus pracuje tak, že nastaví budící proud, zapíše hodnoty měřených veličin náležící tomuto proudu a nastaví následující hodnotu budícího proudu, opět zapíše data atd. Následující hodnotu proudu I_b společně určují proměnná Ib_poz a ovládací ukazatel kroku proudu I_b .

Algoritmus pro měření druhé charakteristiky naprázdno tj. $U=f(n)$ pro $I_b=\text{konst.}$ vychází rovněž z požadavku na toto měření - zapsat hodnoty měřených veličin do tabulky a vykreslit do grafu požadovanou charakteristiku v závislosti na zvoleném kroku otáček. Celý algoritmus je vidět na Obrázku 4.12 v podobě *Case* struktury nazvané *Naprazdno 2*. Algoritmus je kompletně aktivní pouze v případě kdy jsou po zmáčknutí tlačítka *Spustit měření* splněny počáteční podmínky (viz. kap. 4.3.1.3) pro dané měření. V případě kdy je zvoleno manuální měření je aktivní pouze část, která zapisuje hodnoty měřených veličin. Zápisem hodnot měřených veličin se rozumí jak zápis do tabulky, tak zápis do grafu, tak uložení měřených hodnot do datového pole (proměnná *Array2*). Po spuštění automatického měření pracuje algoritmus následovně: Nejprve zobrazí varovná zprávu (je-li tato možnost nastavena - viz. Obrázek 4.4), poté nastaví budící proud na maximální budící hodnotu (určenou proměnnou Ib_do , lze ji nastavit dle potřeby viz. Obrázek 4.2) zablokuje ovládací ukazatel kroku otáček (viz. Obrázek 4.1), nastaví zobrazení příslušného grafu (v případě, že by z předchozích měření bylo nastaveno zobrazení jiného, aby tento nebyl pro uživatele matoucí), nastaví proměnnou *Page 02* na hodnotu *True* a vynuluje datové pole. Navíc se vynuluje pomocná proměnná *PomocnaOt*. Algoritmus je tímto inicializován a začne se sekvenčně provádět *Flat Sequence* struktura. První pole vyhodnocuje ukončovací podmínky, popíšeme později, a horní *Case* struktura hlídá, jestli proud na výstupu zdroje už dosáhl požadované hodnoty. Proměnná *Page 02* povolující tuto *Case* strukturu, je totiž, jak již bylo řečeno, nastavena hned po spuštění měření na hodnotu *True*. Jakmile je požadovaný proud na výstupu zdroje, je vyslán *True* signál pro povolení *Case* struktury nacházející se ve druhém poli. V ní se provede uložení aktuální hodnoty otáček do proměnné *PomocnaOt*, jejíž význam bude vysvětlen, a zároveň se tato struktura nepřímou samou zablokuje, nastavím proměnné *Page 02* na hodnotu *False*. Tím se zablokuje i *Case* struktura v prvním poli, čili tím na jejím výstupu v podobě signálové čáry typu boolean bude trvale hodnota *False*, která blokuje zmíněnou *Case* strukturu ve druhém poli. Funkce těchto dvou popisovaných *Case* struktur je v podstatě jakési doinicializování spuštěného měření. Pakliže se měří charakteristika $U=f(n)$ víme, že při nulových otáčkách bude logicky i nulové výstupní napětí alternátoru. Není proto úplně nutné spouštět toto měření když se poháněcí motor, čili i alternátor netočí. Nabízí se tudíž možnost spustit měření při roztočeném alternátoru (více viz. kap. 4.3.1.3). A právě proto, že tato možnost existuje, je nutné provést popisovanou doinicializaci, která zjistí jakou rychlostí se alternátor točí v okamžiku spuštění měření a uloží ji do proměnné *PomocnaOt*. Tato doinicializace se provede pouze jednou, což zajišťuje proměnná *Page 02*, která je po doinicializování nastavena na *False*. Od hodnoty proměnné *PomocnaOt* se následně odvíjí zápis jednotlivých řádků dat, protože, jak vyjde najevo později, právě ona udává kdy se mohou data zapsat. Ve třetím poli *Flat Sequence* struktury je systém povolování zápisu, obdobný systému z prvního algoritmu, skládá se ze čtyř podmínek:



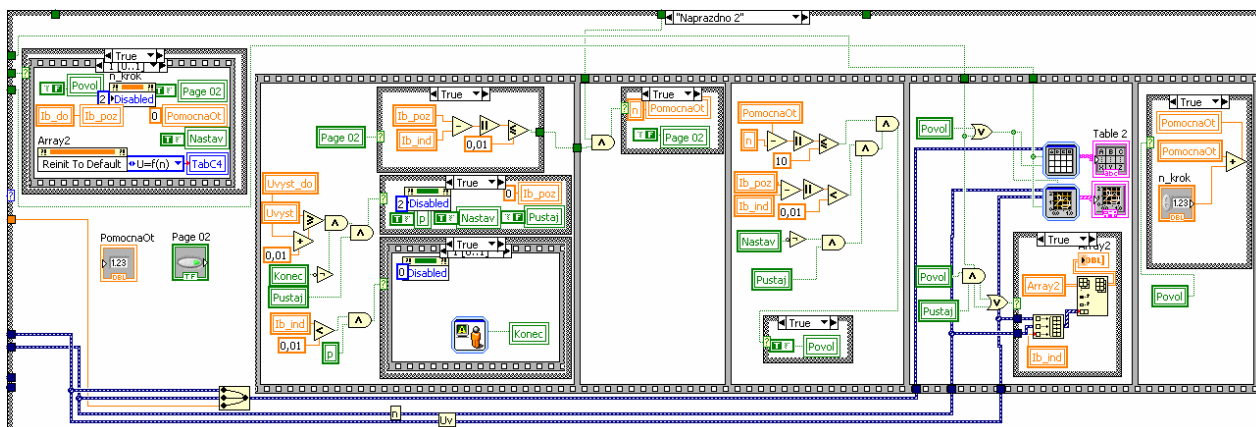
- h) Požadovaný (*Ib_poz*) a nastavený (*Ib_ind*) budící proud v absolutním rozdílu nesmí být větší než 0,01, zároveň
- i) aktuální hodnota otáček *n* se nesmí lišit od hodnoty uložené v proměnné *PomocnaOt* o víc než ± 10 , zároveň
- j) proměnná *Pustaj* musí mít hodnotu *True*, a zároveň
- k) proměnná *Nastav* hodnotu *False*.

Význam podmínek h), j), k) je prakticky stejný jako význam podmínek a), c), d) pro první algoritmus. Jen s tím rozdílem, že tentokrát pro zápis dat není směrodatná podmínka h) ale i). Jakmile je splněna podmínka zápisu, opět formou proměnné *Povol* nastavené na hodnotu *True*, je povoleno i vykonání všech objektových příkazů nacházejících se ve čtvrtém a pátém poli. To, že proměnná *Povol* figuruje v prvním algoritmu (a rovněž i ve třetím, jak se dozvíme později) ničemu nevádí, jelikož se s ní pracuje pouze v rámci *Case* struktury právě zvoleného algoritmu (*Naprazdno 1* nebo *Naprazdno 2* nebo *Zatezovací*). A jak už z podstaty *Case* struktury plyne tyto algoritmy nikdy nemůžou pracovat najednou, vždycky se provádí jen jeden z nich. Takže, co se týče proměnné *Povol*, si algoritmy „nelezou do zelí“. Čtvrté pole zajišťuje zápis naměřených dat, princip je zcela stejný jako u druhého pole v prvním algoritmu. Stěžejní je pole páté, které v případě povolení navýší hodnotu proměnné *PomocnaOt* o hodnotu, která je nastavena ovládacím ukazatelem kroku otáček (prvek *Numeric Control* s názvem *n_krok*). Tímto inkrementováním společně s podmínkou i) ve třetím poli, je algoritmu přesně řečeno kdy má zapisovat data. Podmínka i) vykazuje určité meze kdy ještě může povolit zápis. Tyto meze (hodnota otáček alternátoru *n* se neliší víc než o 10 na obě strany vůči hodnotě v proměnné *PomocnaOt*) jsou nutné z prostého důvodu. Kdyby místo zmíněné podmínky byla nastavena rovnost $n = \text{PomocnaOt}$ prakticky nikdy by nemohlo dojít k zápisu dat. Jen s těžší, spíše velkou náhodou, by se povedlo tuto rovnost přesně splnit. Tušíme správně, že je to dáno ručním nastavováním otáček. Navíc z ovládacího pultu, který nevykazuje zrovna 2x velkou citlivost na podněty ke změně otáček, jelikož jeho reakce na tyto podněty jsou velmi opožděné. Regulace rychlosti otáček pomocí tohoto pultu je značně omezená co se týče malých změn rychlosti otáček. Prakticky se nedá změnit ustálenou hodnotu otáček o hodnotu menší než cca 30 min^{-1} . To znamená, že jestliže je hodnota ustálených otáček např. 597 min^{-1} , po podnětu na zvýšení otáček se změněná hodnota ustálí, v nejlepším případě, na hodnotě okolo 627 min^{-1} . Naštěstí se tato změna provádí spojitě. Ovšem otáčkoměr není schopný zaznamenat tuto změnu spojitě pakliže proběhne příliš rychle. Tím se rozumí, že na jeho výstupu se objevují skokové změny (právě proto by podmínka $n = \text{PomocnaOt}$ nemohla tak snadno platit). Není-li mezi skokovými změnami příliš velký odstup (tzn. řádově jednotky), tak algoritmus funguje spolehlivě díky povolenému rozmezí hodnot otáček, při kterých je možno zápis dat povolit. Na druhou stranu provedla by-li se změna otáček opravdu velmi rychle (skoková změna hodnoty na výstupu otáčkoměru by byla řádově desítky až stovky) je proti tomu algoritmus bezmocný a nefungoval by správně. Čekal by totiž až se na výstupu otáčkoměru objeví hodnota např. v rozmezí 754 ± 10 , aby mohl zapsat data a nastavit následující hodnotu otáček, při kterých má dojít k zápisu (čili 854 ± 10 při zvoleném kroku otáček 100). Ale za předpokladu, že předchozí hodnota pro zápis se pohybovala v rozmezí 654 ± 10 a náhlé skokové změny otáček na hodnotu 765 a vyšší by čekal marně. Hodnota 765 je první z řady, na kterou by už algoritmus nereagoval. Číslo 765 není náhodné, vychází z předchozí hodnoty otáček kdy došlo k zápisu dat tj. 654 ± 10 a je dáno krokem (bylo řečeno, že uvažujeme krok 100) a horním rozmezím hodnot otáček pro povolení zápisu dat (+10). Takže $654 + 100 + 10 + 1 = 765$. Plus 1 určuje první hodnotu, na kterou už algoritmus nezareaguje. Tento číselný příklad byl pro snazší vysvětlení uvažován s celými čísly. Ve skutečnosti otáčkoměr měří

s přesností na desetinná místa, ale to je zanedbatelné, když si uvědomíme, že hodnoty otáček alternátoru se při měření pohybují řádově ve stovkách až tisících. Toto měření s přesností na desetinná místa je jen dalším utvrzujícím důvodem proč bylo nutné zavést okrajové hodnoty (meze) pro povolení zápisu dat. Dochází-li tedy k „přeběhu“ otáček řešením by mohlo být nastavení většího rozmezí otáček pro povolení zápisu dat. To ale není příliš chytré neboť by se tím smazával rozdíl mezi jednotlivými kroky zápisu, obzvláště při nastavení malého kroku. Z důvodů těžkopádné regulace otáček a existence mezí pro zápis je minimální možná hodnota kroku nastavitelná na 80. Vraťme se zpět k situaci kdy očekávané otáčky momentu zápisu dat „přeběhnou“ na vyšší hodnotu než je horní mez. Jedna z možností jak situaci vyřešit je otáčky snížit tak, aby k zápisu dat došlo a poté pokračovat dále v měření (nastavováním vyšších otáček). Tato varianta není zcela ideální neboť charakteristika $U=f(n)$ by se měla měřit za plynulého zvyšování otáček. Elegantní způsob jak se „přeběhnout“ otáček vyhnout je jediný. Regulovat otáčky s citem a s ohledem na nastavenou hodnotu kroku ovládacího ukazatele kroku otáček. Podmínky ukončení měření jsou dvě:

- l) Výstupní napětí (reprezentováno proměnnou U_{vyst}) dosáhne hodnoty stejné nebo vyšší než je nastavená hodnota napětí pro konec měření (reprezentováno proměnnou U_{vyst_do}) zvýšená o 0,01, a zároveň
- m) proměnná $Pustaj$ nabývá hodnoty *True*.

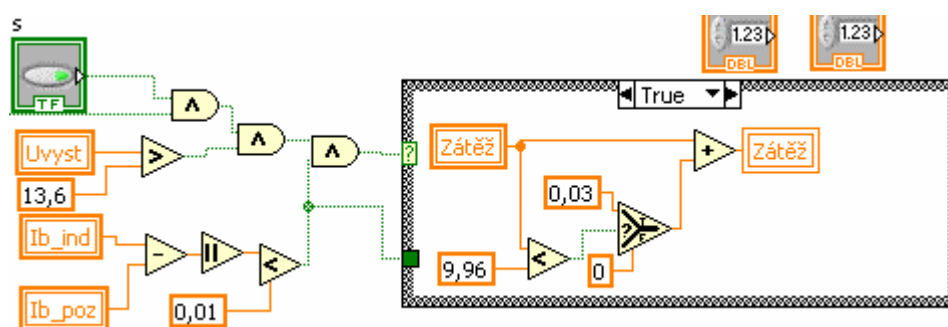
Jsou-li podmínky splněny, měření se opět regulérně dvoufázově ukončí téměř shodným způsobem jako ukončení měření v prvním algoritmu. Opět zde „předává štafetu“ proměnná p , jejíž použití i v prvním algoritmu vůbec nevadí ze stejného důvodu jako nevadí použití proměnné $Povol$. Důvod dvoufázového ukončení byl vysvětlen v popisu funkce prvního algoritmu. Shrnutí a potvrzení je chod druhého algoritmu následující: Nastavit budící proud na maximální budící hodnotu a poté jen čekat na správné okamžiky zápisu dat, které jsou dány aktuální hodnotou proměnné $PomocnaOt$. Ta je průběžně inkrementována v závislosti na nastavení ovládacího ukazatele kroku otáček a zvyšování otáček alternátoru uživatelem.



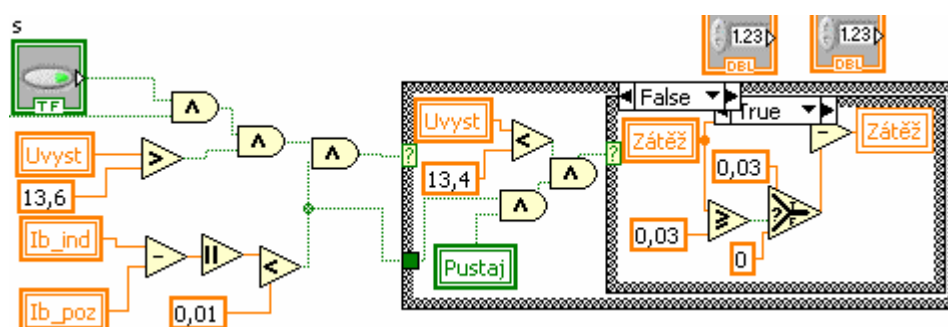
Obrázek 4.12 Druhý algoritmus

Třetí algoritmus je nejkomplikovanější ze všech, protože jeho úkolem je navíc oproti předchozím i regulovat elektronickou zátěž. Vychází totiž z požadavku na měření zatěžovacích charakteristik $I, M=f(n)$. Ten zní zapsat hodnoty měřených veličin do tabulky a vykreslit do grafů požadované charakteristiky v závislosti na zvoleném kroku otáček a udržovat za pomoci elektronické zátěže výstupní napětí na hodnotě 13,5V (s určitou tolerancí). Jelikož je algoritmus rozsáhlejší, je zobrazen na dvou obrázcích - Obrázek 4.13 a Obrázek 4.16. Celý je umístěn v *Case* struktuře s názvem *Zatezovaci*. Pro přehlednější popis jej budeme vnímat jako dvě části.

jsou inicializační a provedou se vždy při spuštění měření. První z nich (nalevo) plní obdobnou funkci jako inicializační *Case* struktury v předchozích dvou algoritmech. Ovšem obsahuje pár objektových příkazů navíc. Důležité je zejména nastavení proměnné *ZatNul* na hodnotu *True*, čímž se zavolá vykonání *Case* struktury umístěné mimo tento algoritmus, její význam a funkce je popsán v kap. 4.3.5. Nastavení proměnné *s* do hodnoty *True*, je jednou z podmínek pro povolení třetí (dolní) *Case* struktury, která reguluje hodnotu proměnné *Zátěž*, bude popsáno. Proměnná *Z0* nastavená rovněž do *True* slouží jako „štafetový kolík“ pro druhou *Case* strukturu (horní napravo) a zároveň i jako její blokátor, když se vykoná. Je zřetelné, že inicializace proběhne ve dvou fázích. Druhá fáze se může spustit až tehdy, má-li proměnná *ZatNul* hodnotu *False*. Důvod vyplývá z funkce *Case* struktury, která se, nastavením této proměnné do hodnoty *True*, volá a po svém dokončení vrací hodnotu proměnné *ZatNul* zpět na *False* (více viz. kap. 4.3.5). Dokonání inicializace spočívá v nastavení budicího proudu na maximální budicí hodnotu, obdobně jako ve druhém algoritmu. Nyní přejdeme k popisu dolní *Case* struktury regulující hodnotu proměnné *Zátěž*. Regulace pracuje pouze v případě, kdy je algoritmus regulérně spuštěn a inicializován, což zaručují mimo jiné i podmínky sdružené logickými ANDy přivedené na vstup *Case Selectoru* popisované *Case* struktury. Její funkce je vcelku jednoduchá, při výstupním napětí vyšším než 13,6V začne inkrementovat hodnotu proměnné *Zátěž* o hodnotu 0,03 dokud výstupní napětí neklesne pod hodnotu 13,6V (Obrázek 4.14). Naopak, klesne-li napětí pod hodnotu 13,4V tak hodnotu proměnné *Zátěž* dekrementuje (rovněž o hodnotu 0,03), dokud výstupní napětí nestoupne nad hodnotu 13,4V (patrně z Obrázku 4.15). Z toho jasně vyplývá, že popisovaná *Case* struktura vyregulovává hodnotu výstupního napětí tak, aby ho udržovala mezi hodnotami 13,4V a 13,6V. Hodnota v proměnné *Zátěž* totiž určuje velikost napětí na analogovém výstupu měřicí karty, které je přivedeno na vstup elektronické zátěže pro externí řízení, které ji ovládá viz. kap. 4.3.5 a 5.2. Hodnota proměnné *Zátěž* se při automatickém měření pohybuje v rozmezí 0 až 9,99 s krokem 0,03. Meze 13,4V a 13,6V, o kterých již byla řeč, tedy znamenají, že výstupní napětí rozhodně není udržováno přesně na hodnotě 13,5V. Toto



Obrázek 4.14 Zátěž True



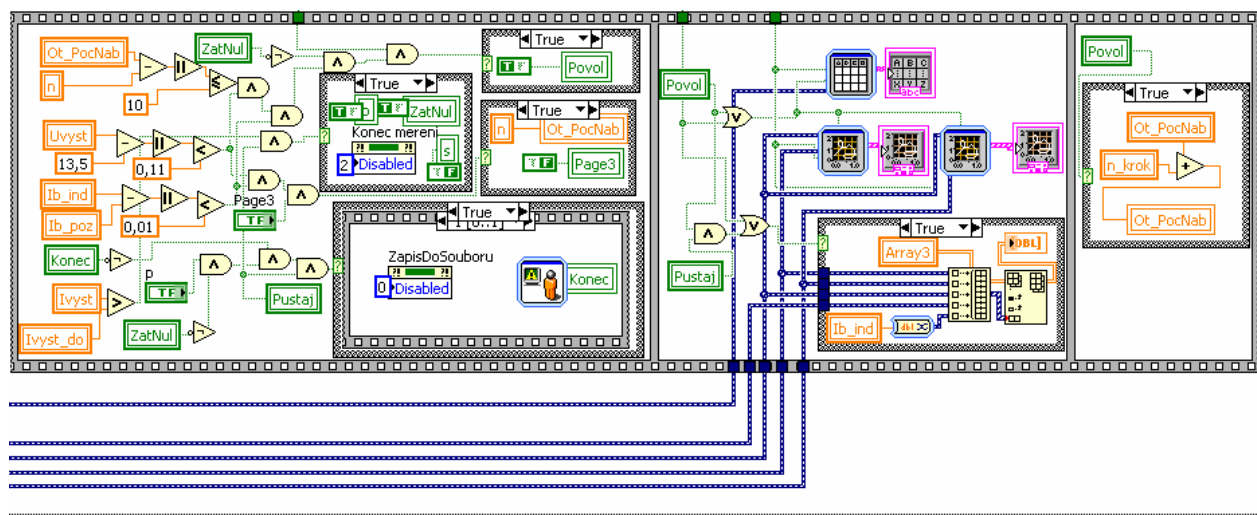
Obrázek 4.15 Zátěž False

řešení, ač zanáší do měření zatěžovací charakteristiky určitou chybu, má svůj důvod. Tím je rychlost regulace výstupního napětí v závislosti na velikosti kroku inkrementování/dekrementování proměnné *Zátěž*. Čím přesněji chceme vyregulovat výstupní napětí, tím musí být velikost kroku změny hodnoty proměnné *Zátěž* menší. Tím více je potřeba provést inkrementací/dekrementací při vyregulovávání hodnoty výstupního napětí, tím je regulace pomalejší. Což se projeví zejména, je-li tato regulace integrována v jedné *While* smyčce, jako součást rozsáhlého programu, tak jako v našem případě. Samotné vykonání jednoho cyklu programu totiž zabere nějaký čas. Pakliže je regulace v tomto programu integrována její chod se vykoná vždy jednou v rámci jednoho cyklu celého programu. Takže složitost a velikost programu má na její rychlost podstatný vliv. Tak jak je regulace řešena s ohledem na rozsáhlost celého programu a výpočetním výkonem notebooku, na kterém byla testována, vykazuje čas mezi vykonáním dvou inkrementací/dekrementací zhruba 0,11s, což je přijatelné. Příliš pomalá regulace hodnoty výstupního napětí je nežádoucí, důvody uvedeme při popisu funkce *Flat Sequence* struktury (Obrázek 4.16). Některé napadne, že by se rychlost regulace hodnoty výstupního napětí dala zvýšit jejím umístěním do samostatné paralelní *While* smyčky. Ano je to tak. Ovšem vzhledem k nestandardnosti tohoto řešení (dvě paralelní *While* smyčky neúměrně zatěžují hardware a sama o sobě je tato vize z programátorského hlediska stěží „stravitelná“) to není výhodou. Efektivita je nízká neboť po otestování regulace hodnoty výstupního napětí, řešené tímto způsobem, se její rychlost zvýšila pouze cca 5x (doba mezi jednotlivými inkrementacemi/dekrementacemi byla cca 5x kratší). Očekávané zrychlení tohoto řešení bylo řádově v násobcích desítek. Pak by se dalo uvažovat i o výrazném snížení kroku inkrementace/dekrementace, čímž by došlo i ke znatelnému zpřesnění regulace hodnoty výstupního napětí. Ale jednoduchou úvahou zjistíme, že už při 3x nižším kroku než, který je použit nyní tj. $0,03:3=0,01$ by se ve výsledku rychlost inkrementování resp. dekrementování oproti nynějšímu řešení zvýšila pouze 1,667x. Tím, že se 3x sníží krok se 3x sníží i rychlost. Původní zrychlení testované regulace umístěné v samostatné *While* smyčce bylo pětinasobné, po

trojnásobném snížení rychlosti regulace se tedy výsledná hodnota rychlosti chodu regulace oproti nynějšímu řešení projeví jako $5:3=1,667$ násobek. Což je, při umístění regulace v paralelní *While* smyčce, opravdu neefektivní a výrazně za očekáváním zaostávající výsledek tohoto způsobu řešení, takže jsme od něj upustili.

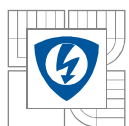
Funkce *Flat Sequence* struktury: Skládá se ze tří polí, z nichž první vyhodnocuje povolování k zápisu dat, který se provádí ve druhém poli. Rovněž povoluje inkrementování proměnné *Ot_PocNab*, které se děje v poli třetím. Také se zde vyhodnocuje případné ukončení měření, k tomu se dostaneme později. Nyní si představme podmínky, které je nutno splnit, aby mohlo dojít k zápisu dat:

- n) Požadovaný (*Ib_poz*) a nastavený (*Ib_ind*) budící proud v absolutním rozdílu nesmí být větší než 0,01, zároveň
- o) hodnota proměnné *ZatNul* musí být *False* (plyne z kap. 4.3.5), zároveň
- p) absolutní rozdílnost hodnoty výstupního napětí (reprezentován proměnnou *Uvyst*) s hodnotou 13,5 musí být menší než 0,11, zároveň
- q) hodnota otáček alternátoru *n* se nesmí lišit s hodnotou proměnné *Ot_PocNab* o víc jak 10.



Obrázek 4.16 Třetí algoritmus 2

Podmínka n) zde slouží spíše jako ověřovací, zda-li budící zdroj dodává požadovaný budící proud, ale nemá tak stěžejní význam jako stejná podmínka a) v prvním algoritmu. Podmínka o) je-li splněna, signalizuje, že inicializační proces po spuštění algoritmu se již vykonal (viz. horní dvě *Case* struktury vedle *Flat Sequenced* struktury). Jednou ze stěžejních podmínek je podmínka p) ta zaručuje, že se zápis hodnot rozhodně neprovede dřív, než se výstupní napětí vyreguluje na hodnotu mezi 13,4V a 13,6V, což je určeno *Case* strukturou pro regulaci hodnoty proměnné *Zátěž* (viz. Obrázek 4.14 a Obrázek 4.15.). Takže algoritmus pakliže chce zapsat data, čeká nejen na správnou hodnotu otáček alternátoru, což říká podmínka q), ale navíc čeká až při těchto otáčkách dojde k vyregulování výstupního napětí. Proto je nutné, aby regulace byla přiměřeně rychlá, jelikož při změně otáček, jejichž hodnota je pro případ tohoto algoritmu směrodatná pro zápis dat, dochází i ke změně výstupního napětí. Nutnost použití podmínky q) je zcela analogická jako nutnost použití podmínky i) ve druhém algoritmu. Rozdíl je pouze v použití jiné proměnné – *Ot_PocNab*. Funkce podmínky q) však zůstává rovněž analogická jako funkce podmínky i). Funkce proměnné *Povol* jako příznaku povolení/zakázání zápisu je naprosto shodná s funkcí stejné proměnné v předchozích dvou algoritmech. Neobjasněna zůstává funkce proměnné *Ot_PocNab*. Do ní se uloží hodnota otáček alternátoru *n*,



v době kdy napětí na výstupu (proměnná U_{vyst}) dosáhne hodnoty mezi 13,4V a 13,6V a za předpokladu indikování požadované hodnoty budícího proudu na výstupu zdroje. Čili v ideálním případě se tak stane při výstupním napětí o hodnotě 13,5V. Což je napětí, při kterém otáčky alternátoru dosahují hodnoty otáček počátku nabíjení. Poté se запиše první řádek měřených dat (protože podmínka q) bude nyní splněna). A následně, ve třetím poli *Flat Sequence* struktury, se hodnota proměnné Ot_PocNab navýší o hodnotu určenou ovládacím ukazatelem kroku otáček. Tím je určen následující moment zápisu dalšího řádku měřených dat. Myšlenka, důvod a princip tohoto inkrementování proměnné Ot_PocNab je naprosto stejný jako inkrementování proměnné $PomocnaOt$ ve druhém algoritmu. Uložení hodnoty otáček alternátoru n do proměnné Ot_PocNab se provede v okamžiku, kdy mají otáčky alternátoru velmi blízkou hodnotu otáček počátku nabíjení. Toto se provede jen jednou, což zajišťuje proměnná $Page3$, která je po spuštění měření nastavena na hodnotu *True* a po provedení prostřední *Case* struktury, ke které je vázaná, se nastaví trvale na hodnotu *False* (dokud není opětovně spuštěno měření) a tím zablokuje její opakované provedení. Teď si popíšeme podmínky ukončení měření:

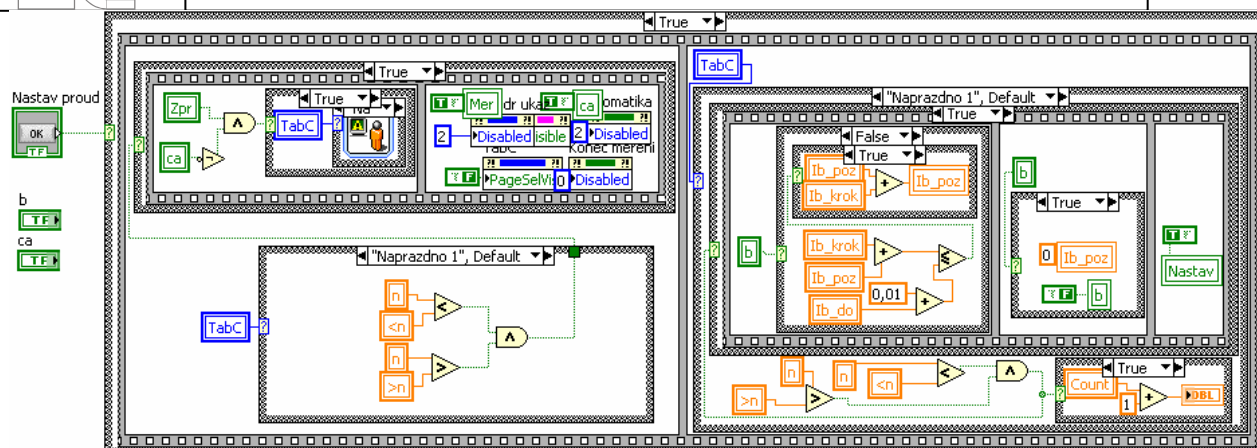
- r) Jednoznačně je to okamžik, kdy hodnota výstupního proudu I_{vyst} překročí hodnotu proměnné I_{vyst_do} (lze nastavit dle požadavků, viz. Obrázek 4.2), a zároveň
- s) proměnná $Pustaj$ má hodnotu *True*.

Podmínka r) vyplývá z obecného postupu pro měření zatěžovací charakteristiky (viz. kap. 6.2) a logicky zapadá do funkční struktury třetího algoritmu. Po splnění podmínek pro ukončení měření se opět algoritmus ukončuje dvoufázově. Nezbytnost dvoufázového ukončení je stejný jako u algoritmu jedna a dva. Při ukončení měření se vykoná levá horní *Case* struktura nacházející se v prvním poli *Flat Sequence* struktury, která nastavením proměnné $ZatNul$ na hodnotu *True* volá *Case* strukturu k této proměnné náležící (viz. kap. 4.3.5) (už o ní byla v předchozím textu řeč). Proměnná p zde opět plní funkci „štafetového kolíku“. Po provedení dolní *Case* struktury, jíž je „štafetový kolík“ předán, je měření regulérně ukončeno a uživatel je o tom informován zobrazením zprávy.

Ve výsledku algoritmus pracuje tak, že čeká na vhodné momenty, kdy může zapsat měřená data. Ty jsou určeny aktuální hodnotu proměnné Ot_PocNab a podmínkou výstupního napětí pohybujícího se v rozmezí 13,4-13,6V. Hodnota v proměnné Ot_PocNab je průběžně navyšována v závislosti na nastavení ovládacího ukazatele kroku otáček a zvyšování otáček alternátoru uživatelem.

4.3.1.2 Struktury pro manuální měření

Výše popsané algoritmy plně pracují, když je nastaveno a spuštěno automatické měření charakteristik. Ovšem virtuální přístroj umožňuje i manuální režim měření. Manuální režim je ošetřen proti případným chybám lidského faktoru (viz. kap. 4.3.2) a tudíž se k němu váží určité programové *Case* struktury, které zabezpečují, v součinnosti s jinými, bezproblémový chod manuálního režimu měření. Nejpodstatnější je *Case* struktura *Nastav proud*, která se provede vždy po zmáčknutí tlačítka *Nastav proud*. Její funkce je nejen, jak už název napovídá, nastavit proud podle zvolené měřené charakteristiky, ale také zablokovat/odblokovat a skrýt/odkrýt určité prvky na čelním panelu, obdobně jak tomu je při spouštění algoritmů automatického měření. Pochopitelně se toto provede jen v případě, jsou-li splněny počáteční podmínky viz. kap. 4.3.1.3). Pakliže nejsou splněny počáteční podmínky, je o tom uživatel informován zprávou vypsanou na displej. Blíže si popíšeme funkci této struktury pro případ *Naprazdno 1* (viz. Obrázek 4.17).

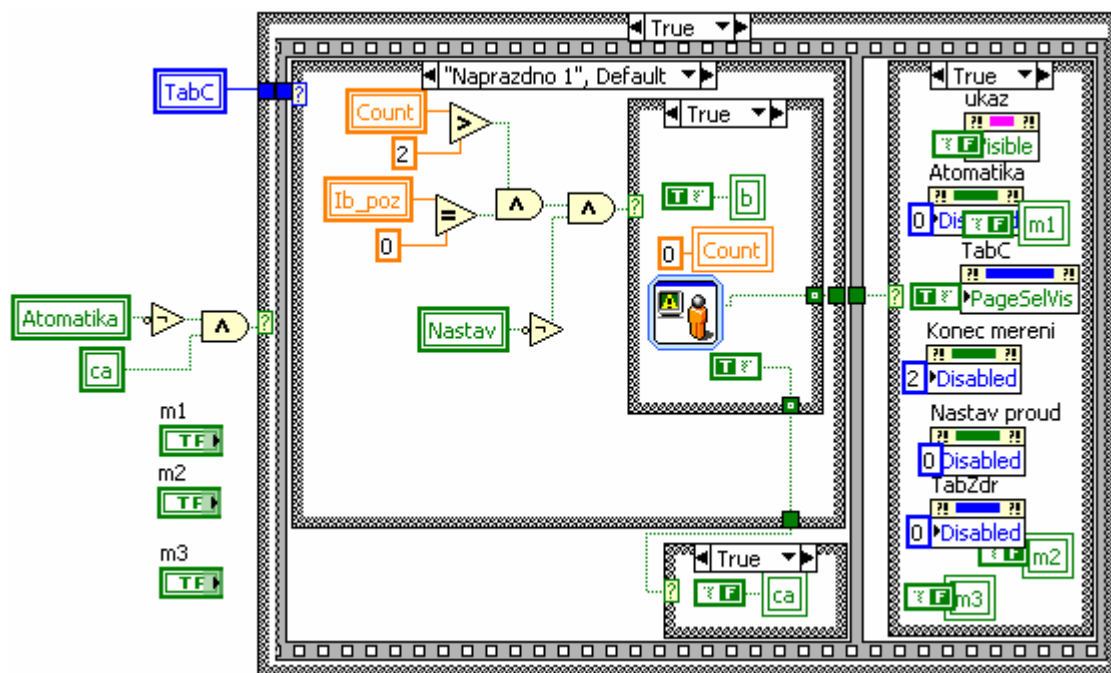


Obrázek 4.17 Struktura *Nastav proud*

Hlavní (ta největší) *Flat Sequence* struktura je tvořena dvěma kroky (poli). První je inicializační a provede se vždy jedenkrát při spuštění měření libovolné charakteristiky. Zobrazí uživateli upozorňovací zprávu a změní vlastnosti některých prvků na čelním panelu přístroje podle Tabulky 4.1. Navíc se nastaví příznak *ca*, který signalizuje, že je spuštěno měření v režimu manuálním. Druhý krok již funguje podle zvolené měřené charakteristiky. V tomto případě (*Naprazdno 1*) obsahuje podmíněnou *Flat Sequence* strukturu, skládající se ze tří kroků, z nichž první krok se po prvním stisku tlačítka *Nastav proud* nevykoná, ale následující druhý již ano. Třetí se vykoná vždy. Tím, že se vykoná druhý krok, který se pak sám zakáže, se zaručí, že se vždy po prvním zmáčknutí (tj. po zahájení měření) tlačítka *Nastav proud* budící proud nastaví na nulu. Při opakovaném stisku tlačítka již pracuje jen krok první a třetí, dokud nedojde k ukončení měření. První krok inkrementuje proměnnou *Ib_poz* o velikost určenou prvkem *Krok Ib* na čelním panelu přístroje (ve zdrojovém kódu pod názvem *Ib_krok*). Následující třetí krok, požadavek tohoto proudu na výstupu zdroje předá přes proměnnou *Nastav Nastavovací* strukturu (viz. kap. 4.3.4) a zdroj na svém výstupu nastaví požadovanou hodnotu budícího proudu. Pakliže součet aktuální hodnoty proměnné *Ib_poz* a *Ib_krok* je vyšší než hodnota *Ib_do* (která určuje konec měření), čili vznikne požadavek na příliš vysoký budící proud, nastaví se na výstupu zdroje automaticky nulový budící proud a uživateli je oznámeno, že měření je dokončeno. V podmíněné *Flat Sequence* struktuře je krom struktury *Naprazdno 1* ještě další malá *Case* struktura. Ta se provede po každém stisku tlačítka *Nastav proud* a počítá kolikrát bylo toto tlačítko stisknuto, což, jak bude vysvětleno později, má svůj význam při ukončení manuálního měření ve struktuře *Ukončovací* pro případ *Naprazdno 1*.

Struktura *Ukončovací* má význam pouze pro manuální režim měření a to v době kdy je spuštěno. Příznakem *ca* nastaveným na hodnotu *True* je totiž tato struktura neomylně aktivována. Je v ní integrováno, podobně jako ve struktuře *Nastav proud*, ukončování měření pro všechny tři charakteristiky. Zde si popíšeme jen případ *Naprazdno 1* (viz. Obrázek 4.18). Struktura v prvním kroku čeká na ukončovací podmínky, což jsou v tomto případě hodnota proměnné *Ib_poz*=0 a zároveň hodnota proměnné *Count*>2 a zároveň hodnota proměnné *Nastav*=*False*. Doplňující podmínka *Count*>2 je nutná neboť podmínka *Ib_poz*=0 je hned po prvním spuštění struktury *Nastav proud* (bylo řečeno, že vždy nastaví budící proud zdroje na nulu, čili i proměnná *Ib_poz* musí být nulová). Tím, že se několikrát stiskne tlačítko *Nastav proud* dojde ke změně hodnoty proměnné *Ib_poz* na nenulovou a je tedy jasné, že až bude opět nulová, (což bude a plyne to z popisu struktury *Nastav proud*), může se měření ukončit (podmínka *Count*>2 bude rovněž splněna). Po provedení prvního kroku *Ukončovací* struktury se proměnné *Count*, *b* a *ca* nastaví na výchozí hodnotu a uživateli se zobrazí zpráva, že bylo měření dokončeno. Tím, že ji odklikne

se povolí druhý krok, jehož úkolem je navrátit vlastnosti prvků na čelním panelu na stejné, které měly před spuštěním měření. Navíc se i příznaky $m1$, $m2$, $m3$ nastaví na své výchozí hodnoty. Příznaky $m2$ a $m3$ hrají roli při ukončování manuálního měření pro případy *Naprazzdno 2* a *Zatezovací*. Je to nutné pro dvoufázové ukončení založené na stejné myšlence jako ukončování algoritmů pro automatické měření. Navíc s příznakem $m1$ hrají roli pro zobrazování informačního textu na informačním displeji na předním panelu přístroje viz. kap. 4.3.6 - *Status* struktura. Z výše uvedeného popisu vyplývá že struktury *Nastav proud* a *Ukončovací* spolu spolupracují a mohou fungovat pouze v režimu nastavení přístroje na *Manual*.



Obrázek 4.18 Ukončovací struktura

4.3.1.3 Struktury vyhodnocující počáteční podmínky

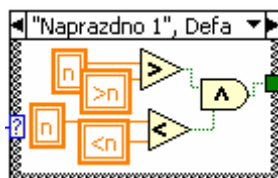
Z pohledu zdravého rozumu a logického uvažování při ručním měření charakteristik alternátoru, každého věci znalého intuitivně napadne za jakých podmínek je nesmyslné tu či onu charakteristiku začít měřit. Např. charakteristiku naprázdno $U=f(I_b)$ je nelogické začít měřit např. od hodnoty budícího proudu 2A nebo pro maximální otáčky alternátoru apod. Pro charakteristiku $U=f(n)$ je zase nesmysl začít měřit od vysokých otáček např. 1000 min^{-1} nebo pouze s malým budícím proudem. Obdobně to platí pro charakteristiky zatěžovací. Podobných a jiných chyb při začátku měření se může dopustit každý kdo problematice dostatečně nerozumí. Proto jsou do programu implementovány podmínky pro spouštění měření tak, aby k těmto chybám nemohlo dojít. Jednak je to z důvodu bezpečnostních a také z důvodu eliminování chyb lidského faktoru.

Při hlubším zamyšlení a analýze problému dojdeme k závěru, že jediná veličina, která postačí k určení, zda-li jednotlivá měření mohou začít či nikoliv jsou otáčky. Samozřejmě za předpokladu, že otáčkoměr bude správně fungovat. Jeho případná porucha není v programu žádným způsobem podchycena. Počáteční podmínky pro spuštění jednotlivých měření jsou následující:

Pro charakteristiku $U=f(I_b)$ (první algoritmus):

Hodnota otáček alternátoru n musí být vyšší než hodnota určená proměnnou $>n$ a nižší než hodnota určená proměnnou $<n$ (viz. Obrázek 4.19). Hodnoty obou proměnných lze nastavit

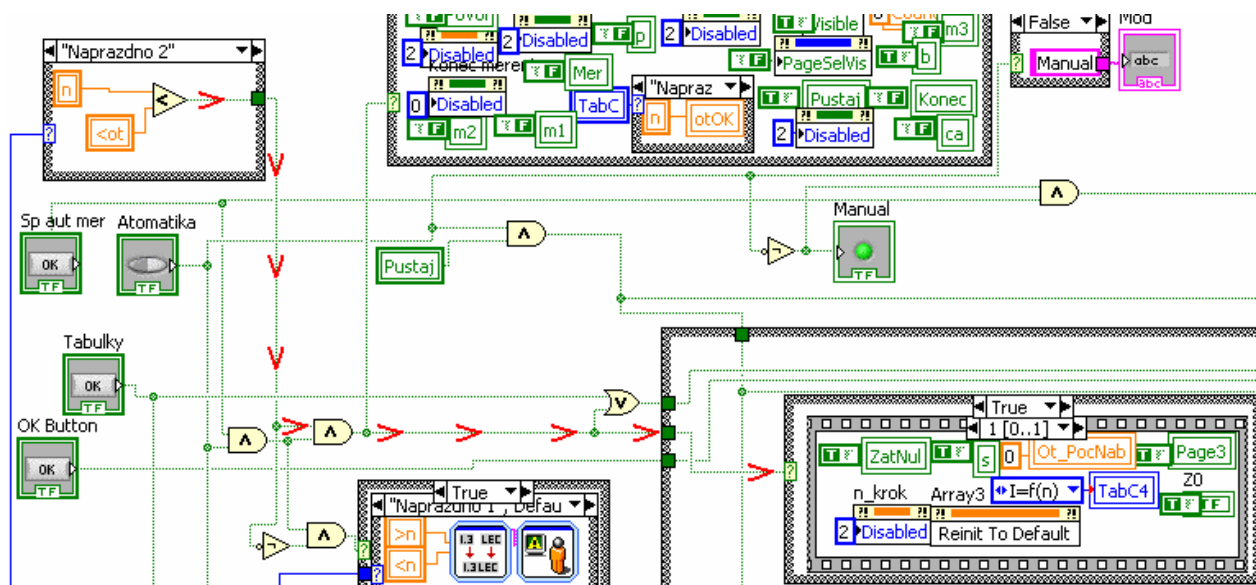
(viz. Obrázek 4.4). Je totiž nesmysl měřit tuto charakteristiku naprázdno pro příliš nízké (řádově desítky za sekundu) nebo dokonce nulové otáčky, ale zároveň není ani chytré ji zkoušet měřit pro otáčky příliš vysoké. Neboť při postupném zvyšování hodnoty budícího proudu, se zvyšuje i výstupní napětí a při příliš vysoké hodnotě otáček a budícím proudu na maximum by mohlo dosáhnout hodnoty, na kterou není alternátor navržen.



Obrázek 4.19 Podmínka 1

Pro charakteristiku $U=f(n)$ (druhý algoritmus):

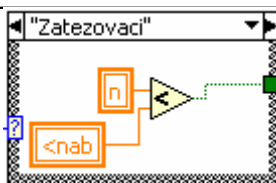
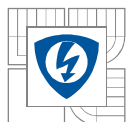
Hodnota otáček alternátoru n musí být nižší než hodnota proměnné $<ot$ viz. Obrázek 4.20. Na obrázku je vyznačeno kudy je podmínkový signál přiveden k inicializační Case struktuře druhého algoritmu. Její hodnota lze rovněž nastavit viz. Obrázek 4.4. Je nesmyslné začít měřit tuto charakteristiku naprázdno pro vysoké otáčky i přesto, že víme, že její očekávaný průběh je přímka. Dalším důvodem je, že po spuštění měření této charakteristiky se ihned nastavuje budící proud na maximum a opět by hrozilo, že výstupní napětí dosáhne nebezpečné hodnoty.



Obrázek 4.20 Podmínka 2

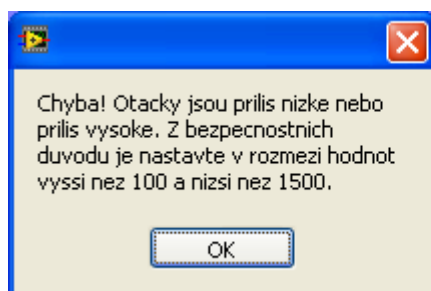
Pro charakteristiky $I, M=f(n)$ (třetí algoritmus):

Hodnota otáček alternátoru n musí být nižší než hodnota proměnné $<nab$ viz. Obrázek 4.21. Tato hodnota je samozřejmě také nastavitelná. Pro měření této charakteristiky je nelogické začít měřit pro otáčky vyšší než jsou otáčky počátku nabíjení.



Obrázek 4.21 Podmínka 3

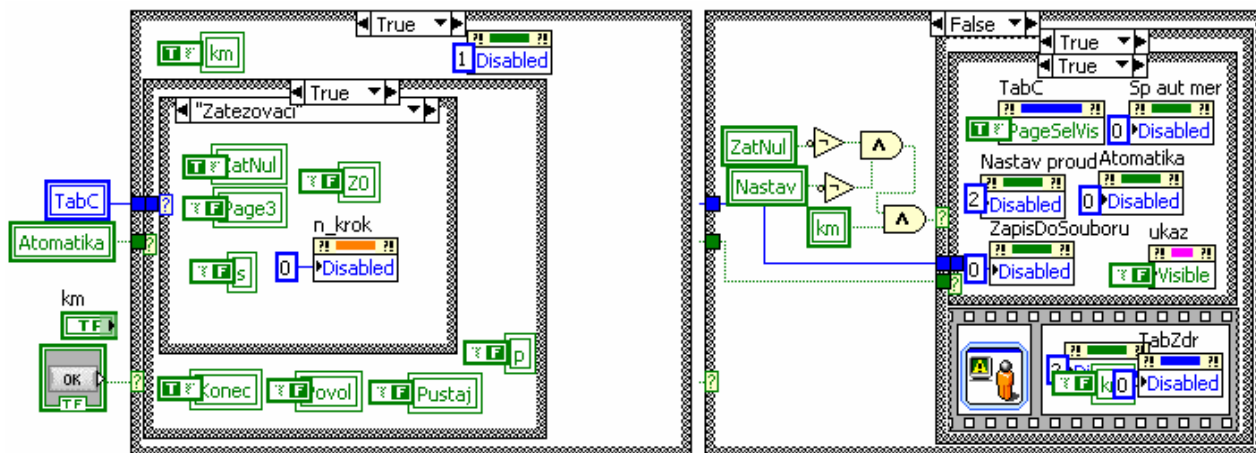
Tyto počáteční podmínky se uplatňují jak v manuálním režimu měření tak v režimu automatickém. Pakliže nejsou počáteční podmínky splněny, po stisku tlačítka *Spustit mereni* se objeví zpráva, která toto uživateli oznámí. Na Obrázku 4.22 je její příkladné znění společně s *Case* strukturou, která se o zobrazování těchto zpráv stará. Z pohledu programátorského je nutné pro spuštění měření navíc splnění dalších dvou podmínek. V režimu manuálním je to nastavení právě na tento režim a stisknutí tlačítka *Nastav proud* nacházejícím se na čelním panelu virtuálního přístroje. V režimu automatickém je to naopak nastavení na režim *Auto* a stisknutí tlačítka *Spustit mereni*.



Obrázek 4.22 Příklad zprávy

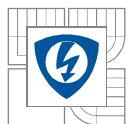
4.3.1.4 Tlačítko *Konec mereni*

Po stisku tlačítka *Konec mereni* (viz. Obrázek 4.1) se provede struktura *Konec měření* má na starosti ukončování měření na žádost uživatele, kdykoliv v průběhu započatého měření. Tzn.,



Obrázek 4.23 Struktura *Konec měření*

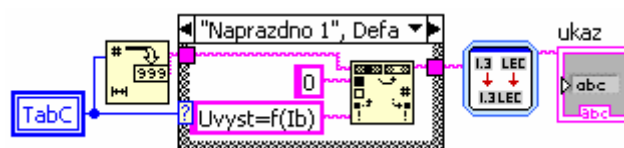
že se provede zmáčkne-li uživatel tlačítko *Konec mereni* na předním panelu virtuálního přístroje. Tato struktura vychází z toho, že může být spuštěna opravdu jen tehdy, když je započato měření (tlačítko *Konec mereni* se odblokuje). Funguje tak jako všechny struktury, které dokáží ukončit měření (např. *Ochranná*, *Ukončovací*), dvoufázově. Provede sled ukončovacích příkazů podle



toho jaká je nastavena záložka a režim přístroje. Jejím prioritním cílem je bezpečně ukončit měření (první fáze) a následně vhodně změnit vlastností prvků na čelním panelu měřicího přístroje jako jejich zablokování/odblokování, skrytí/odkrytí a nakonec uživateli vypíše zprávu, že měření se ukončilo na jeho žádost (druhá fáze). Na Obrázku 4.23 je první a druhý krok struktury *Konec měření* pro případ *Zatezovací*. První krok je vykonán ve *True* případě pro tuto strukturu a druhý pro případ *False*. Druhý krok se vykoná vždy bez ohledu na zvolenou charakteristiku měření, při které ke stisku tlačítka *Konec mereni* došlo. V tomto kroku se rozlišuje pouze, zda-li je režim přístroje automatický nebo manuální.

4.3.1.5 Displej aktuálního měření – struktura *Ukaž*

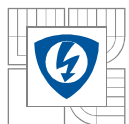
Jednou ze struktur, která má svůj nepostradatelný význam je struktura *Ukaž*, je vidět na Obrázek 4.24. Ta se stará o zobrazování obsahu displeje aktuálního měření (v programu je to objekt *String Indicator* a má jméno *ukaz*). Struktura je významná proto, že vznikla jako jeden z prvků, který spolupracuje při odolnosti aplikace vůči vnějším lidským faktorům. Jejím jediným výstupem na čelním panelu přístroje je právě displej aktuálního měření. Ovšem ten se zobrazuje pouze je-li spuštěno měření, což je důležité. Neboť po spuštění měření se hlavnímu prvku *Tab Control* (se záložkami *Naprazdno 1*, *Naprazdno 2* a *Zatezovací*), nacházejícímu se na čelním panelu virtuálního přístroje, záložky skryjí a zdánlivě se tak ztrácí přehled o tom, která charakteristika se právě měří. Ovšem od toho je zmiňovaný displej, který se v době schování záložek odkryje a zobrazuje, která charakteristika se měří. Podstatnou informaci je k tomu potřeba získat z proměnné *TabC*, která náleží prvku *Tab Control* (označeném na Obrázku 4.1 číslem 1) a její hodnota se pohybuje v rozmezí od 0 do $u-1$, kde u je počet záložek prvku *Tab Control*. V našem případě jsou tedy tři, takže proměnná *TabC* v závislosti na zvolené záložce ukazuje hodnotu nula, jedna nebo dva. Tato hodnota by pro uživatele neměla žádnou vypovídající hodnotu. Proto je nejprve pomocí prvku *Number To Decimal String* převedena na datový typ *string* a následně prvkem *Search and Replace String* nahrazena textem, který už pro uživatele vypovídající schopnost bude mít. Aby tento text nebyl sám o sobě příliš strohý je prvkem *Build text* doplněn o pár slov. Výsledný text pro záložku *Naprazdno 1* tedy bude: „Prave se meri Uvyst= $f(I_b)$ “ (viz. Obrázek 4.6).



Obrázek 4.24 Struktura *Ukaž*

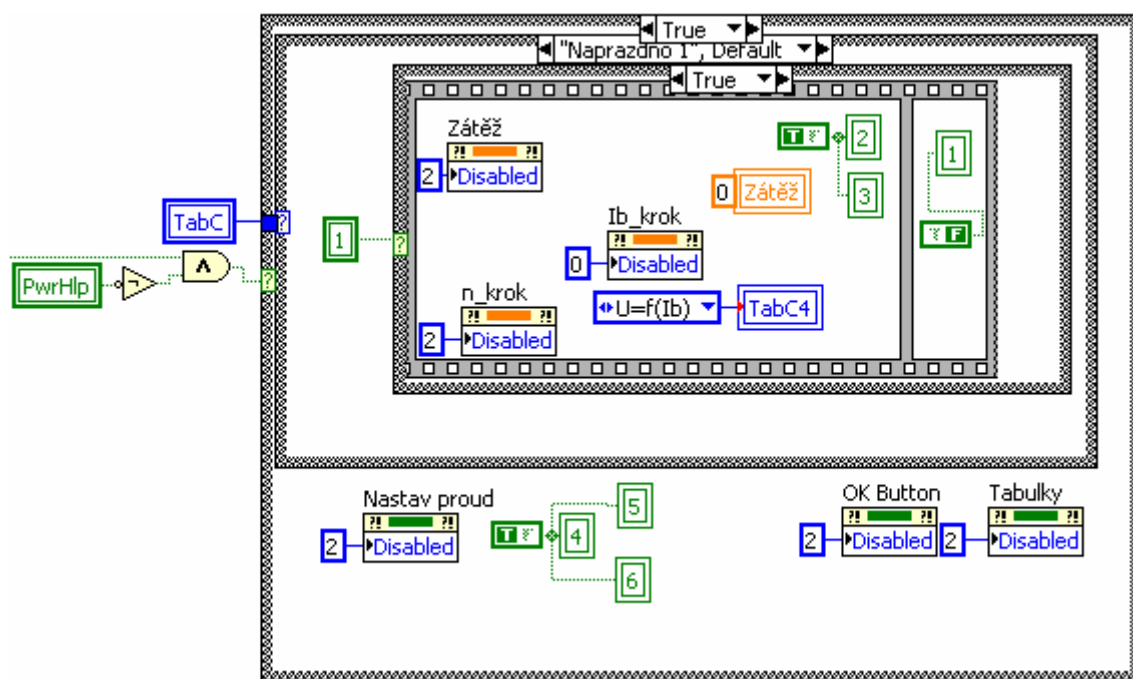
4.3.2 Odolnost proti lidským chybám

Jedním z cílů při tvorbě virtuálního měřicího přístroje bylo realizovat ho tak, aby byl v co největší míře odolný vůči vnějším chybám lidského faktoru. Což je řešeno způsobem blokace ovládacích prvků čelního panelu jako tlačítek, *Numeric Controlů*, záložek apod. Konkrétně se manipuluje s tlačítky *Reset tabulky*, *Zapis do tabulky*, *Ulozit do souboru*, *Spustit mereni*, *Nastav proud*, *Konec mereni* a „bezejmenným“ přepínačem pro přepínání režimu *Auto/Manual*. Dále s *Numeric Controly Zatez*, *Krok Ib* a *Krok otacek* a v neposlední řadě se záložkami prvku *Tab Control*, ve kterém jsou umístěny tabulky, s displejem aktuálního měření a se záložkou *Nastaveni zdroje* (viz. Obrázek 4.5). Většinu vyjmenovaných prvků je vidět na Obrázku 4.1. Systém blokace jednotlivých prvků je závislý na určeném režimu měření (*Auto/Manual*) a také na tom, jestli je měření spuštěno či nikoliv. Pakliže je spuštěno, je rozhodující, i která charakteristika se



právě měří, čili který příslušný algoritmus zrovna „běží“. Systém manipulace s prvky přehledně zobrazuje Tabulka 4.1

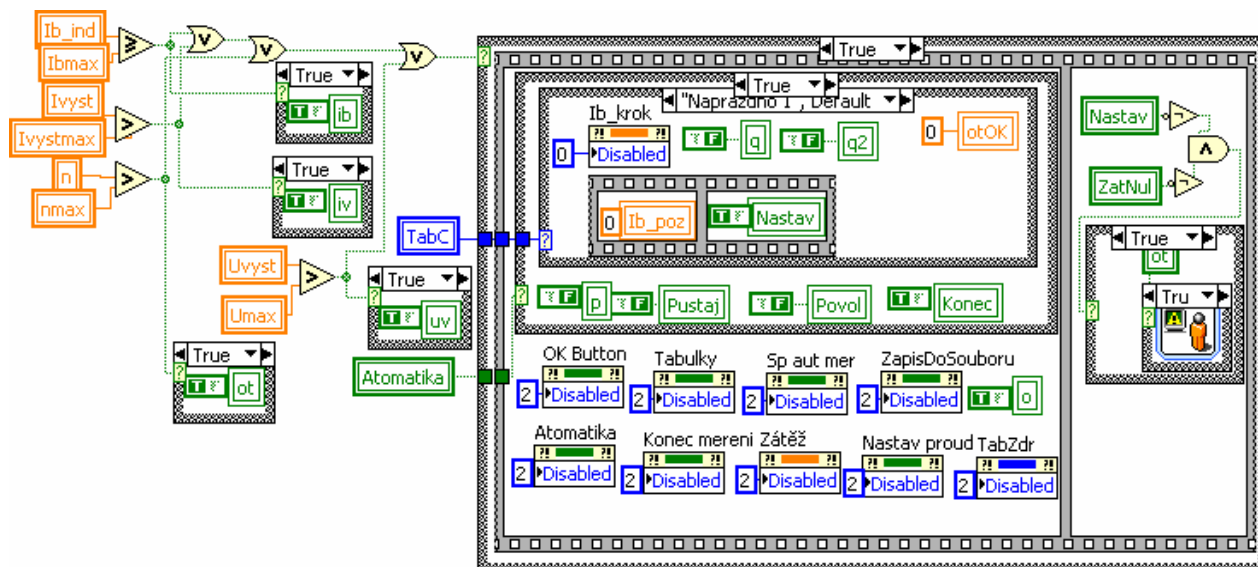
Blokovací Case struktura, která má na starosti mimo jiné blokování prvků, které mají v tabulce znak x, je na Obrázku 4.25. Je to jen její část pro příkladnou ukázkou, jelikož v sobě ukrývá celkem 6 různých kombinací jak prvky blokovat (v závislosti na nastaveném režimu, které jsou dva – Auto/Manual a také v závislosti na zvolené záložce, ty jsou tři. $3 \times 2 = 6$). Obrázek si pro lepší pochopení popíšeme. Vnitřní *Flat Sequence* struktura se vykoná pouze jednou, což zaručuje její nadřazená *Case* struktura v součinnosti s proměnnou *I*. O jednu úroveň výše je další nadřazená *Case* struktura, tou se prvkem *TabC* určuje, která záložka je právě zvolena. Úrovně nejvyšší *Case* struktura určuje, *je-li* zvolen režim *Auto* či *Manual*. Na obrázku je tedy zvolen režim *Auto* a záložka *Naprazdno 1*. *Property Node* prvky náležící tlačítkům a *Numeric Controlům* na předním panelu přístroje, jsou nastaveny přesně tak, jak udává Tabulka 4.1. *Node Ok Button* náleží tlačítku *Zapis do tabulky* a *Node Tabulky* náleží tlačítku *Reset tabulky*. Ostatní jsou poměrně jasné. Pozornému čtenáři neunikl fakt, že na obrázku není *Property Node* pro tlačítko *Konec mereni*. To je totiž zablokováno v *Case* struktuře pro vypnutí virtuálního přístroje, neboť je blokováno vždy, bez ohledu na zvolenou záložku či režim přístroje, což je vidět i v Tabulce 4.1. Vedlejší funkce *Blokovací* struktury je nastavit záložku prvku *Tab Control* s názvem *TabC4* na záložku odpovídající právě zvolené záložce *Tab Controlu TabC*. To proto, aby se na čelním panelu přístroje zobrazil patřičný graf, který náleží záložce určující právě zvolenou charakteristiku. *Blokovací* struktura sama o sobě ještě odolnost proti lidskému faktoru nezaručí. Ta funguje až v součinnosti s jednotlivými algoritmy a strukturami pro dané charakteristiky. Ty po spuštění dodatečně zablokuji/odblokuji či skryjí/odkryjí prvky dle potřeby a po svém „dokončení“ resp. dokončení měření zase vše vrátí do původního stavu tak, jak to nastavuje *Blokovací* struktura. Odolnost vůči lidským chybám rovněž zvyšuje existence počátečních podmínek pro spuštění jednotlivých měření viz. kap. 4.3.1.3.



Obrázek 4.25 Blokovací Case struktura

4.3.3 Systém ochrany

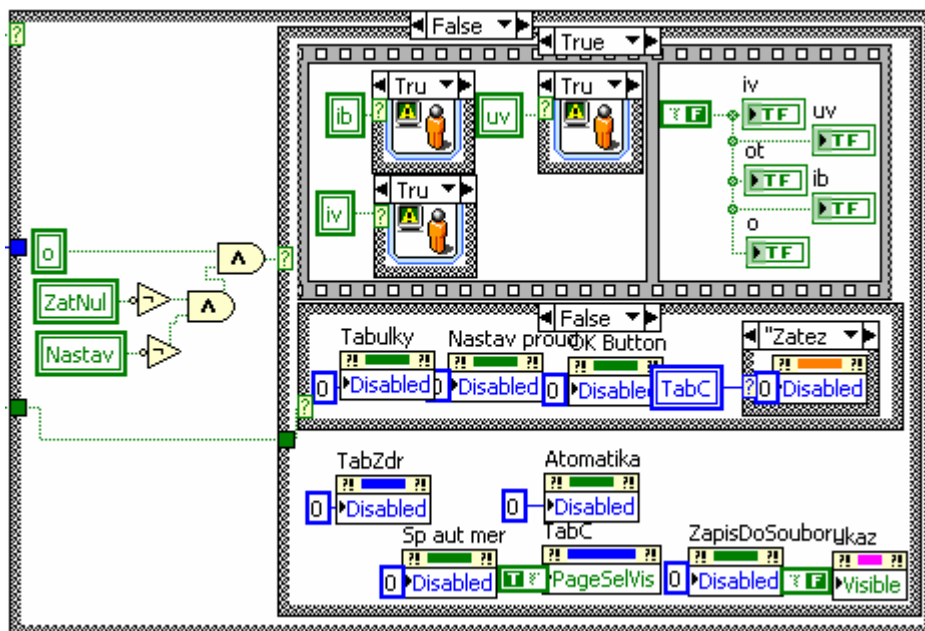
Systém ochrany reaguje na situace kdy některá ze čtyř sledovaných veličin – budící proud, výstupní napětí, výstupní proud a otáčky překročí ochrannou hranici danou nastavením v záložce *Nastavení* viz. Obrázek 4.2. *Ochranná Case* struktura pracuje dvoufázově. První fáze se vykoná pro její *True* případ a druhá pro její *False* případ. Důvod je prostý, po zareagování ochrany je vhodné vypsát na displej zprávu, která uživatele informuje o tom, která z veličin překročila dovolenou hodnotu. Čili je vhodné použít objekt *Display Message to User*. Ten má však nežádoucí účinky, o kterých je pojednáno v popisu prvního algoritmu. Z toho vyplývá nutnost dvoufázové funkce *Ochranné* struktury, aby se zmíněný objekt použil až ve druhé fázi. V první fázi se, v prvním poli *Flat Sequence* struktury, zablokuje pomocí objektů *Property Node* všechna tlačítka na čelním panelu přístroje a podle právě nastavené záložky (algoritmu) a režimu měření se spustí sada příkazů obdobně jako při bezpečném ukončení. Např. pro algoritmus *Naprazdno 1* viz. Obrázek 4.26 se vyšle požadavek stáhnutí budícího proudu na nulu (proměnná $Ib_poz=0$ v součinnosti s proměnnou $Nastav=True$). Dále se nastaví příznaky používané u daného algoritmu do výchozího stavu. Navíc se vždy nastaví příznak ochrany *o* do



Obrázek 4.26 Ochranná Case struktura - True

stavu *True*, jeho funkce se projeví ve druhém kroku *Ochranné* struktury. Ve druhém poli *Flat Sequence* struktury vidíme objekt *Display Message to User*. Je to matoucí neboť bylo řečeno, že tento objekt se použije až ve druhém kroku *Ochranné* struktury. Ne však v případě kdy je třeba uživatele varovat před nebezpečně vysokými otáčkami alternátoru. Není třeba čekat na druhý krok *Ochranné* struktury, důvod vyjde najevo z dalšího textu. Jelikož otáčky nastavuje uživatel ručně a program to nemůže ovlivnit, je vhodné uživatele varovat okamžitě po vykonání nezbytně nutných a možných kroků k zamezení případného poškození alternátoru (stáhnout budící proud na nulu, případně stáhnout zátěž). Překročí-li otáčky alternátoru ochrannou hodnotu, uživateli je zobrazena zpráva *Okamzite SNIZTE otacky!!! Je prekrocena jejich povolena hodnota!* Nic víc totiž program dělat nemůže. Ve druhém kroku ochranné struktury (viz. Obrázek 4.27) už jde jen o to zobrazit uživateli, která z veličin překročila ochrannou hodnotu (kromě otáček) a vrátit prvky na čelním panelu do výchozího stavu, tak jak jsou blokovány/odblokovány nebo skryty/odkryty před spuštěním měření. Příznak *o* zajistí provedení *False* kroku *Ochranné* struktury pouze jednou, a příznaky *ib*, *uv*, *iv* a *ot* určují, která ze zpráv má být uživateli zobrazena a po použití jsou všechny vráceny do výchozího stavu (*False*). Kde se provede nastavení příznaků na

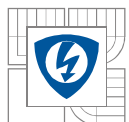
případnou hodnotu *True* je rovněž patrné z Obrázku 4.26. Nutno podotknout, že softwarová ochrana se nemůže rovnat z pohledu spolehlivosti ochranně hardwarové. Ovšem měření charakteristik alternátoru není činnost nějak zvláště a výrazně kritická právě z hlediska ochrany, takže popsaná softwarová ochrana je postačující a je vytvořena mimo jiné i z důvodů nepodcenění tohoto problému.



Obrázek 4.27 Ochranná Case struktura – False

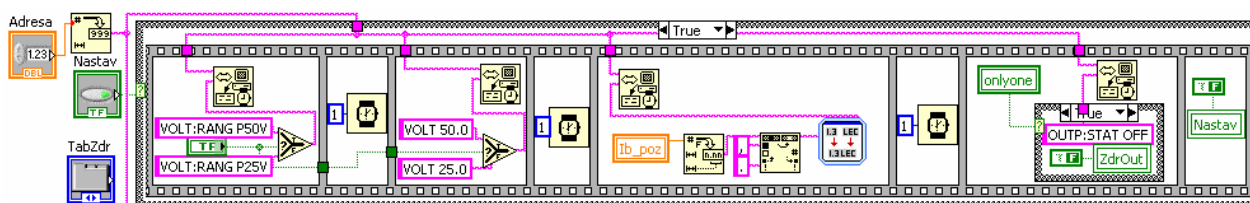
4.3.4 Struktury komunikující se zdrojem přes rozhraní GPIB

Nejdůležitější, a v programu nejpoužívanější je *Case* struktura volaná pomocí proměnné *Nastav*. Jejím úkolem je nastavit výstupní proud zdroje na požadovanou hodnotu, která je vždy určena hodnotou uloženou v proměnné *Ib_poz*. Tato proměnná je vždy, podle potřeby jednotlivých algoritmů (viz. kap. 4.3.1) a v závislosti na nastavení měření, nastavována na žádanou hodnotu. Popišme si *Nastavovací Case* strukturu viz. Obrázek 4.28. Obsahuje *Flat Sequence* strukturu o velikosti osmi polí. Tyto pole určují jednoznačné pořadí vykonání jednotlivých příkazů tak, jak mají po sobě následovat. Jako první se do zdroje pomocí objektu *GPIB Write* pošle příkaz VOLT:RANG P50V (resp. VOLT:RANG P25V, podle nastavení zdroje viz. Obrázek 4.5), který nastaví napěťový, a jemu odpovídající proudový, rozsah zdroje, následuje prvek *Wait*, který zabezpečuje prodlevu jedné milisekundy mezi posláním dalšího příkazu do zdroje. Dalším příkazem posílaným do zdroje je VOLT 50.0 resp. VOLT 25.0, tím se nastaví napěťový limit zdroje. Předposledním ze sady příkazů je nastavení požadovaného proudu. Zdroj začne pracovat v režimu CC (Constant Current) více viz. kap. 5.3. Příkaz pro nastavení proudu není z *Flat Sequence* struktury zřetelně patrný jako ostatní. Může vypadat např. takto: CURR 3.0. Problémem je že hodnota proudu je určena proměnnou *Ib_poz* a ta se dle potřeb mění, čili se tento příkaz nedá napsat jednoduše a „natvrdo“ jako ostatní. Navíc se zde naráží na další problém formátu zápisu proměnné s desetinnou čárkou, kdežto příkaz pro nastavení proudu zdroje musí být s desetinnou tečkou. Krom toho je proměnná číslo, kdežto příkaz pro zdroj je text (deklarován jako *string*). Proto nejprve proměnná vstupuje do bloku *Number To Fractional String*, kde je převedena na text. Následuje blok *Search and Replace String*, kde se desetinná čárka nahradí desetinnou tečkou a nakonec vstupuje takto upravená proměnná do bloku *Build Text*. Ten obsahuje text v podobě CURR %Ibpoz%. Řetězec znaků %Ibpoz% je vždy nahrazen aktuálně požadovanou hodnotou budícího proudu v proměnné *Ib_poz*, která do tohoto bloku vstupuje.



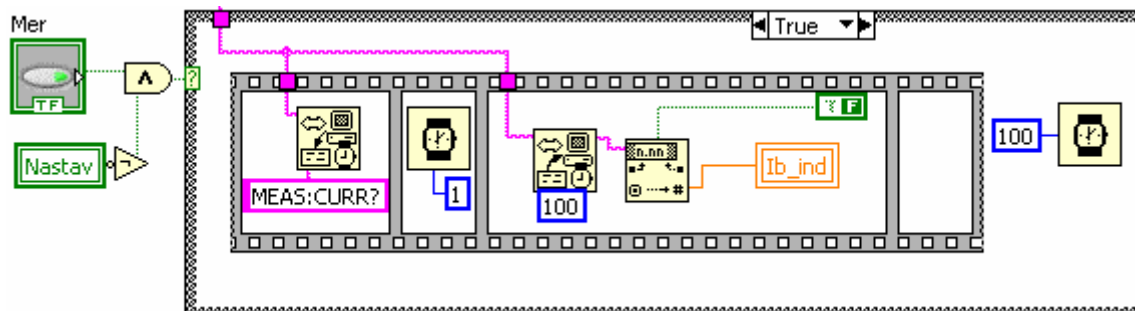
V tomto případě tedy číslem překonvertovaným na typ string. Výstup bloku *Build Text* již předává bloku *GPIO Write* pro něj smysluplný příkaz např. `CURR 0.8`. Posledním příkazem, který je zdroji poslán je `OUTP:STAT OFF` případně `OUTP:STAT ON`, což určuje proměnná *onlyone*. Tento příkaz zapne/vypne výstup zdroje. Proměnná *onlyone* indikuje, je-li virtuální měřicí přístroj vypnutý nebo zapnutý, a v závislosti na tom se výstup zdroje vypíná/zapíná a také rozsvěcuje LED dioda nacházející se v menu pro nastavení zdroje viz. Obrázek 4.5. V posledním kroku se *Nastavovací* struktura zakáže a je připravená na další volání v případě nutnosti jejího použití v programu.

Do všech bloků, které jsou součástí všech popsanych struktur v této kapitole, *GPIO Write* a *GPIO Read* na jejich vstupy s názvem *address string* vedou signály s informací o GPIO adrese zdroje. Tyto signály jsou zobrazeny v podobě růžových čar. GPIO adresu je možno zadat v menu nastavení zdroje, rovněž viz. Obrázek 4.5. Ovládací ukazatel adresy (objekt *Numeric Control* s názvem *Adresa*) je však deklarován jako číslo proto je jeho výstup nutné překonvertovat na datový typ *string* pomocí bloku *Number To Decimal String*.



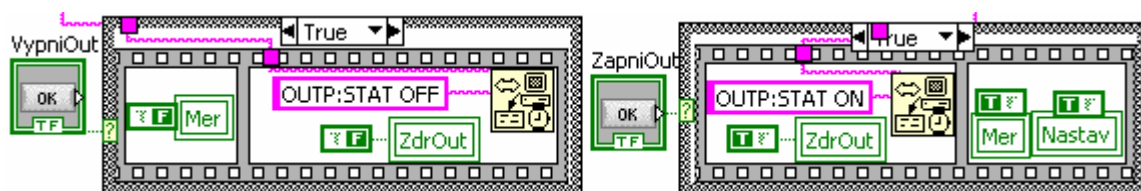
Obrázek 4.28 Nastavovací Case struktura

Další *Case* struktura, která se zdrojem komunikuje je *Měřicí* viz. Obrázek 4.29. Dá se říct, že je aktivní, když je proměnná *Mer* nastavena na hodnotu *True*. Doplnující podmínka, aby tato struktura mohla „běžet“ je nastavení proměnné *Nastav* na hodnotu *False*. To proto, aby nedošlo k souběžnému běhu struktury *Nastavovací* a *Měřicí*. Zdroj by tak mohl dostat příkazy typu *read/write*, které by mohli způsobit určitou kolizi v rámci komunikace aplikace se zdrojem, čemuž je tímto opatřením zabráněno. Měřicí struktura se vykonává každých 100ms a aktualizuje hodnotu výstupního proudu v podobě ukládání do proměnné *Ib_ind*, která je na čelním panelu virtuálního přístroje zobrazována na číselném displeji (*Numeric Indicator*). Struktura nejprve vyšle do zdroje příkaz typu *write then read* `MEAS:CURR?` pomocí bloku *GPIO Write*. Poté blokem *GPIO Read* načte hodnotu, kterou zdroj vrací, což je aktuální hodnota budícího proudu na výstupu zdroje. Je však vracena v podobě datového typu *string*, takže pro uložení do proměnné *Ib_ind* je ji ještě nutno převést na číslo, což zajistí blok *Fract/Exp String To Numer*. Nutno podotknout, že *Měřicí* struktura tak jak je navržena, by bez předchozího spuštění struktury *Nastavovací* sama o sobě nefungovala, proto je zapotřebí *Nastavovací* strukturu alespoň jednou spustit. To je v programu ošetřeno a zajištěno.



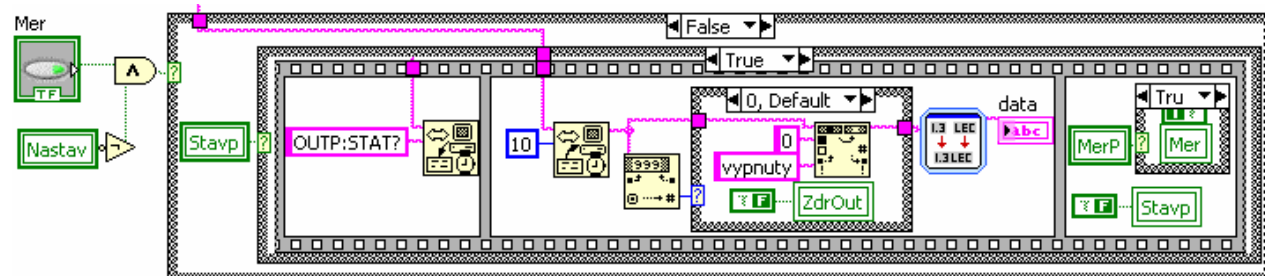
Obrázek 4.29 Měřicí Case struktura

Case struktury, které uživateli umožňují vypnout nebo zapnout výstup zdroje jsou *Vypínací* a *Zapínací* viz. Obrázek 4.30. Spouští se po uvolnění tlačítka *Vypnout vystup* resp. *Zapnout vystup* neboť ty jsou nastaveny jako klasické „klikací“ tlačítka. Jejich mechanická funkce je tedy nastavena na *Latch When Released* a fungují např. jako levé tlačítko na standardní PC myši. V programu mají tyto tlačítka název *VypniOut* a *ZapniOut*. *Vypínací Case* struktura nejprve vypne strukturu *Měřicí*, tím že do proměnné *Mer* zapíše hodnotu *False* (je zbytečné, aby měřila, když je výstup zdroje vypnutý) a poté přes blok *GPIO Write* vyšle do zdroje příkaz *OUTP:STAT OFF*. Navíc nastaví hodnotu *False* do proměnné *ZdrOut*, což je již zmiňovaná LED dioda. *Zapínací* struktura funguje přesně naopak. LED diodu nastaví na *True* a do zdroje pošle příkaz *OUTP:STAT ON*, a poté spustí *Měřicí* strukturu, čili budící proud se začne měřit a zobrazovat na čelním panelu. K čemuž je nutné spustit i provedení struktury *Nastavovací*, neboť například po resetu zdroje *Resetovací* strukturou (bude popsána) je třeba zdroj opětovně nastavit, jinak by *Měřicí* struktura sama o sobě správně nefungovala.



Obrázek 4.30 Vypínací a Zapínací Case struktury

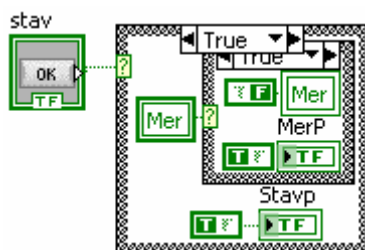
Zajímavou Case strukturou je struktura zjišťující stav výstupu zdroje (viz. Obrázek 4.31). Tzn. že se uživateli po jejím zavolání (odkliknutím tlačítka *Overit stav vystupu* v menu nastavení zdroje (viz. Obrázek 4.5) vypíše stav výstupu zdroje do informačního displeje zdroje. Tato struktura posílá do zdroje příkaz typu *write then read* (stejně jako struktura *Měřicí*). A protože



Obrázek 4.31 Zjišťovací Case struktura

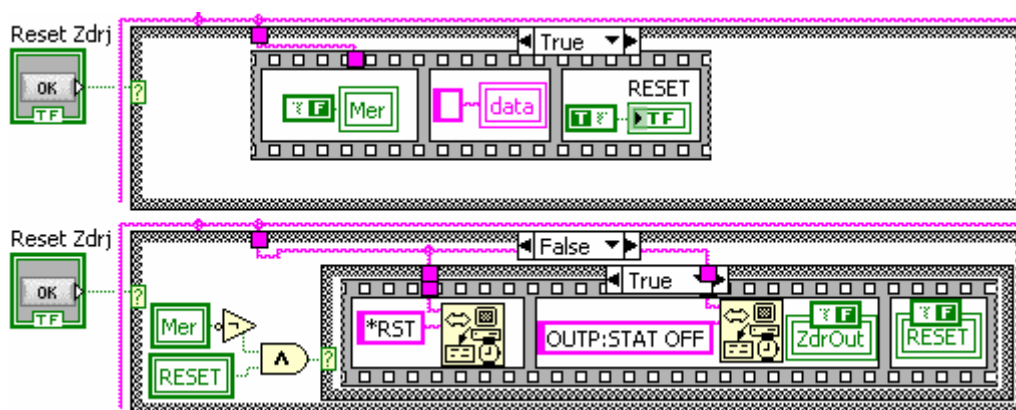
Měřicí struktura běží prakticky pořád, pokud uživatel nevypne výstup zdroje, bylo nutné *Zjišťovací* strukturu ošetřit tak, aby se nespouštěla zároveň se strukturou *Měřicí*, protože by mohlo dojít ke kolizi a zdroj by hlásil chybu. Ke spuštění *Zjišťovací* struktury dojde nepřímo po odkliknutí tlačítka *Overit stav vystupu* (v programu pod názvem *stav* viz. Obrázek 4.32). Tímto odkliknutím se totiž vypne struktura *Měřicí* a do pomocné proměnné *MerP* se uloží stav proměnné *Mer* před odkliknutím tlačítka „stav“. To proto, aby mohlo dojít k opětovnému zapnutí *Měřicí* struktury po dokončení struktury *Zjišťovací*. Proměnná *Stavp* se nastaví na hodnotu *True*. Souvislosti pochopíme po vysvětlení struktury *Zjišťovací*: Ta je z důvodu, které také vyjdou najevo, umístěna ve „False Caseu“ struktury *Měřicí*. Bylo řečeno, že struktura *Zjišťovací* a *Měřicí* se nesmí spouštět zároveň. Tím, že se *Měřicí* struktura vypne, čili přejde ze svého „True Caseu“ na „False Case“ je umístění *Zjišťovací* struktury do tohoto „False Caseu“ jistotou, že tyto struktury nebudou pracovat zároveň. I proto je potřeba *Zjišťovací* strukturu spouštět nepřímo, způsobem řečeným výše. Proměnná *Stavp* zajistí vykonání této struktury pouze 1x po stisku tlačítka „stav“. V posledním poli této struktury se totiž nastaví do hodnoty *False*. *Zjišťovací*

struktura v prvním kroku vyšle do zdroje příkaz OUTP:STAT?. V kroku následujícím je použit objekt *GPIB Read*, protože jak již bylo řečeno příkaz je typu *write then read*. Zdroj vrací hodnotu 0 v případě vypnutého výstupu a hodnotu 1 v případě výstupu zapnutého. Tato informace by pro uživatele neměla příliš velkou vypovídací schopnost. Proto je použito následujících pár bloků (*Decimal String To Numer*, *Search and Replace String* a *Build Text*), které tuto informaci převedou na text *vystup je vypnutý* resp. *vystup je zapnutý* a zobrazí ji na informačním displeji zdroje (viz. Obrázek 4.5). Navíc podle toho rozsvítí patřičnou barvou LED diodu s názvem *ZdrOut*. V posledním kroku, krom zablokování chodu struktury *Zatěžovací*, dojde k vyhodnocení, zda-li byla *Měřící* struktura před spuštěním struktury *Zjišťovací* spuštěna či nikoliv a podle toho vrátí hodnotu *Mer* do patřičného stavu.



Obrázek 4.32 Ověřit stav výstupu?

Poslední *Case* strukturou, kterou je možno vysílat příkazy do budícího zdroje, je *Resetovací* (viz. Obrázek 4.33). Ta funguje ve dvou krocích. Prvním krokem je zde myšleno vykonání „*True Caseu*“ a druhým „*False Caseu*“. V prvním kroku se vyřadí z chodu struktura *Měřící*, obsah informačního displeje zdroje se smaže a nastaví se příznak *RESET* na hodnotu *True*. Ten zajistí, aby se druhý krok provedl jenom jednou. V něm se vyšle do zdroje příkaz **RST* a následně příkaz *OUTP:STAT OFF*. Nastaví se *ZdrOut* na *False*, což je už několikrát zmíněná LED dioda, a nakonec se nastaví příznak *RESET* na hodnotu *False*, tím se *Resetovací* struktura připraví na další použití.



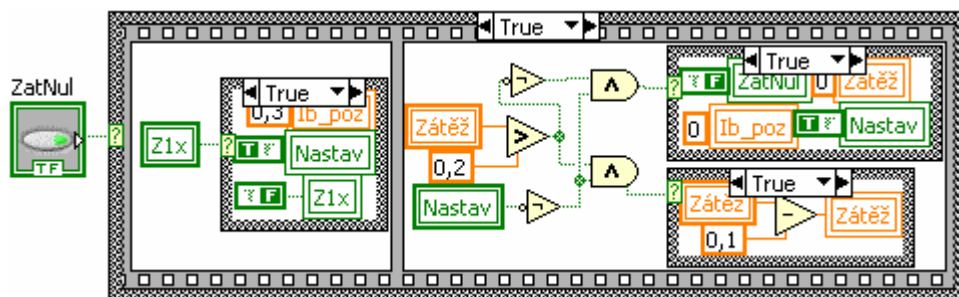
Obrázek 4.33 Resetovací Case struktura

4.3.5 Zatěžovací struktura

Zatěžovací struktura viz. Obrázek 4.34 se mimo jiné využívá pro bezpečné ukončení třetího algoritmu, ať už z důvodu ukončení automatickým nebo uživatelem nebo ochranou. Jejím úkolem je plynule snížit hodnotu proměnné *Zátěž* na nulu, tak aby nedošlo k prudké skokové změně zmíněné proměnné např. z osmi na nulu. Tím by totiž došlo i k prudké skokové změně řídicího napětí elektronické zátěže, jelikož proměnná *Zátěž* určuje jeho hodnotu. Přesněji je to tak, že hodnota proměnné *Zátěž* určuje offset simulovaného stejnosměrného napětí, které je pomocí DAQ Assistant viz. Obrázek 4.35 převedeno na výstup měřící karty a z toho je následně

přivedeno na externí řízení elektronické zátěže. Skoková změna řídicího napětí rozhodně není přípustná, v době měření zatěžovací charakteristiky, protéká-li elektronickou zátěží proud řádově v desítkách ampér. Činnost *Zatěžovací Case* struktury vychází z myšlenky jak bezpečně ukončit měření zatěžovací charakteristiky. Což lze shrnout do dvou kroků – odbudit alternátor a plynule snížit řídicí napětí elektronické zátěže. Jakmile je *Zatěžovací* struktura aktivována (nastavením hodnoty proměnné *ZatNul* na *True*) začne se provádět vnitřní *Flat Sequence* struktura, která v prvním kroku (poli) sníží budící proud na hodnotu 0,3A a nastavením proměnné *Z1x*, která je tu vyložena pro tento účel, na hodnotu *False*, sama sobě zakáže provádění prvního pole v dalším cyklu. První krok stačí totiž provést jen jednou – odbuzení. Z jistých technických důvodů, plynoucích z funkce elektronické zátěže, není budící proud v prvním kroku zcela vynulován. Když bychom totiž odbudili úplně na nulu výstupní napětí alternátoru by rovněž kleslo na nulu. Ovšem pokles výstupního napětí na nulu by způsobil, že by elektronická zátěž přestala reagovat na externí řídicí napěťový signál, což je dáno jejím nastavením (viz. kap.5.2). Tento jev je samozřejmě nežádoucí. Proto odbuzujeme na nenulovou hodnotu budícího proudu.

Druhý krok je podmíněn nastavením proměnné *Nastav* na hodnotu *False*. To zaručí provádění druhého kroku opravdu až po té co proběhne celá sekvence *Nastavovací* struktury (viz. kap.4.3.4). V tomto kroku se proměnná *Zátěž* (a tedy i řídicí napětí) plynule snižuje o hodnotu 0,1 je-li její hodnota větší než 0,2. Dosáhne-li její hodnota meze 0,2 nebo menší, tak se rovnou vynuluje. Rovněž se vyše příkaz k nastavení budícího proudu na nulu a *Zatěžovací* struktura se zablokuje nastavením proměnné *ZatNul* na hodnotu *False*. Ve „*False* cyklu“ *Zatěžovací* struktury se nastavuje proměnná *Z1x* na hodnotu *True*, aby byla *Zatěžovací* struktura připravena k dalšímu použití.



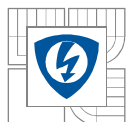
Obrázek 4.34 *Zatěžovací Case* struktura



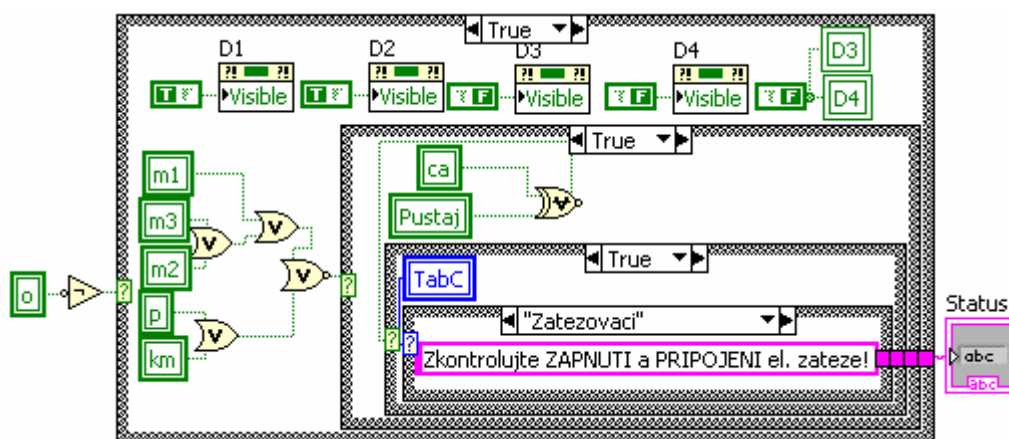
Obrázek 4.35 *DAQ Assistant Zátěž*

4.3.6 Ostatní programové struktury

Status struktura zobrazuje na informačním displeji na čelním panelu virtuálního měřicího přístroje (viz. Obrázek 4.1 oblast 3) zprávy, které uživatele informují o průběhu měření případně ho upozorňují co by měl udělat. Systém zobrazování zpráv je prioritní. Tzn. že zpráva, která se týká situace s vyšší prioritou má při zobrazení přednost před zprávou, která se týká situace s prioritou nižší. Přehledně je to shrnuto v Tabulce 4.3. Například, je-li pouze zapnutý virtuální přístroj, aniž by bylo spuštěno měření zobrazí se zpráva náležící aktuálně zvolené záložce bez ohledu na režim přístroje. Spustí-li se měření, což je indikováno příznakem *ca* pro měření manuální a příznakem *Pustaj* pro měření automatické, zobrazí se zpráva: „!! PRAVE PROBIHA MERENI !!“ bez ohledu na to, jaká je zvolená záložka. Ukončování měření, z hlediska



zobrazování zpráv, má prioritu ještě vyšší než spuštění měření. Je vlastně součástí probíhajícího měření, avšak chceme-li toto vzájemně odlišit je nutné to provést právě pomocí určitých priorit. Že se zrovna ukončuje měření indikuje příznak *p* pro automatizované algoritmy a příznaky *m1*, *m2* a *m3* při režimu manuálním. Nejvyšší prioritu má zpráva, která se týká příznaku *o*, jenž signalizuje zareagování Ochranné struktury (viz kap.4.3.3). Na Obrázku 4.36 je provedení *Status* struktury v její grafické zdrojové podobě v programu. Vedlejší funkce Ochranné struktury je „změnit barvu“ blikajících diod na čelním panelu přístroje z původní oranžové/žluté na červenou/tmavě červenou. Po odeznění příznaku *o* (jeho nastavení na hodnotu *False*) se barva diod opět vrátí zpět. Tato barevná změna je řešena za použití dvou diod vzájemně umístěných v zákrytu. Jedny blikají střídavě žlutá/oranžová (*D1*, *D2*) druhé (*D3*, *D4*) červená/tmavě červená. V okamžiku zareagování ochrany se vlastnost *Visible* původních „žlutých“ diod změní na *False* a naopak pro diody „červené“, které byly do té doby skryty se změní na *True*.



Obrázek 4.36 Status Case struktura

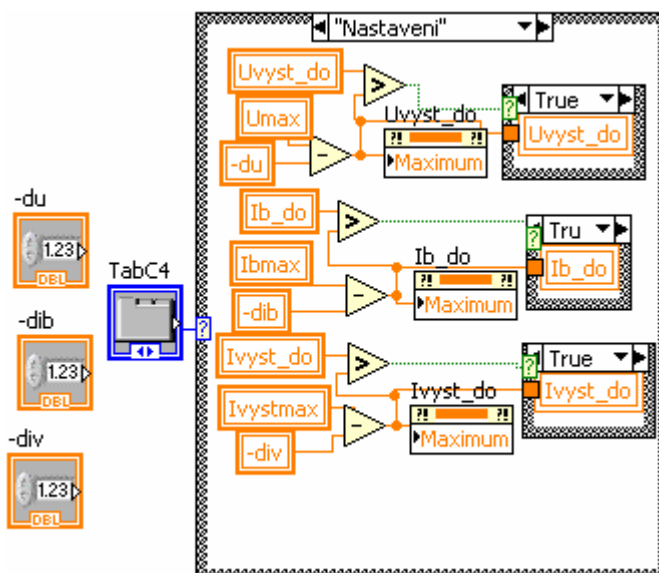
Tabulka 4.3 Informační zprávy a jím příslušné priority

Priorita:	0			1	2		3
Aktivní záložka nebo příznak:	Naprazdno 1	Naprazdno 2	Zatezovací	ca, Pustaj	m1, m2, m3, p	km	o
Zkontrolujte ZAPNUTI a ODPOJENI el. zateze!	x	x					
Zkontrolujte ZAPNUTI a PRIPOJENI el. zateze!			x				
!! PRAVE PROBIHA MERENI !!				x			
Mereni se bezpecne ukoncuje.					x		
Mereni se ukoncuje na zadost uzivatele.						x	
Nektera z velicin prekrocila povolenou hodnotu!!							x

0-nejnižší priorita, 3-nejvyšší priorita

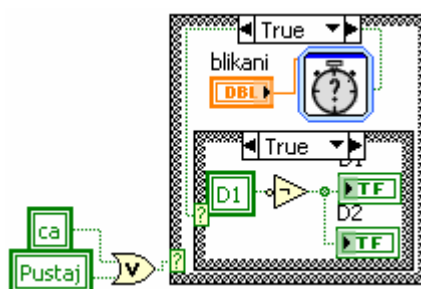
Přenastavovací Case struktura, slouží k přenastavení vlastností prvků *Numeric Control* (*Uvyst_do*, *Ib_do*, *Ivyst_do*) pomocí *Property Node*ů. Konkrétně je to vlastnost *Data Entry Limits*-> *Maximum*. Tyto prvky se nachází v záložce *Nastavení* na čelním panelu virtuálního přístroje viz. Obrázek 4.2. Účelem *Přenastavovací* struktury je zabránit tomu, aby se mohly hodnoty pro konce měření nastavit vyšší než ochranné hodnoty. Maximální nastavitelné hodnoty pro konce měření jsou vždy v závislosti na zvoleném odstupu (viz. Obrázek 4.3) o tento odstup sníženy vůči hodnotě ochranné. Například, je-li hodnota pro konec měření pro výstupní napětí

nastavena na 18V a odstup 1V, dojde-li ke snížení ochranné hodnoty výstupního napětí např. na hodnotu 17V, tak *Přenastavovací* struktura sníží hodnotu pro konec měření tohoto napětí na 16V. Odstupy mají v programu jména *-du*, *-dib*, *-div*. Proměnné s „indexem“ *do* jsou proměnné pro konce měření a proměnné s „indexem“ *max* jsou ochranné. Funkce struktury je jasná z Obrázku 4.37.



Obrázek 4.37 Přenastavovací case struktura

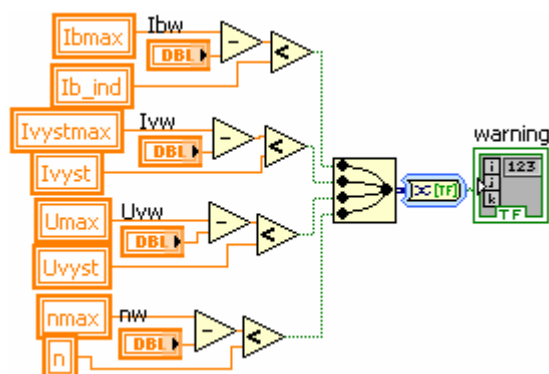
Blikací Case struktura se stará v době spuštěného měření o blikání diod signalizujících spuštěné měření. Jak je toto řešeno je patrné z Obrázku 4.38. Objekt *blikani* určuje rychlost blikání diod. Objekt *Elapsed Time* pak každých několik desetin až jednotek vteřin (v závislosti na nastavení objektu *blikani*) vyšle *Enable* signál struktury *Case*, která pomocí prvku *Not* změní aktuální stav diody (*True*=svítí-žlutě, *False*=nesvítí-oranžově) na opačný.



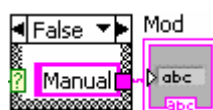
Obrázek 4.38 Blikací Case struktura

Varovná struktura se stará jen o rozsvěcování a zhasínání varovných diod, které upozorňují na to, že ta či ona veličina se blíží k hodnotě ochranné. Kontrolky začnou svítit, když se jednotlivé měřené veličiny přiblíží na nastavený rozdíl k hodnotám ochranným (maximálním dovoleným). Tyto rozdíly lze nastavit v menu pokročilého nastavení v záložce *Varovne kontrolky* (viz. Obrázek 4.3) V programu viz. Obrázek 4.39 jsou to prvky se jmény *Ibw*, *Ivw*, *Uvw* a *nw*.

Mini strukturou je *Mód* struktura (viz. Obrázek 4.40), která pouze ukazuje na čelním panelu jaký je právě zvolený režim měření virtuálního přístroje. Tzn. zobrazuje text *Auto* nebo *Manual*, v závislosti na nastavení přepínače, který je na čelním panelu beze jména, ale v programu má název *Automatika*. Výstup této struktury je logicky displej (prvek *String Indicator*) a slouží v programu jako elegantní řešení pro zobrazování stavu zmíněného přepínače.

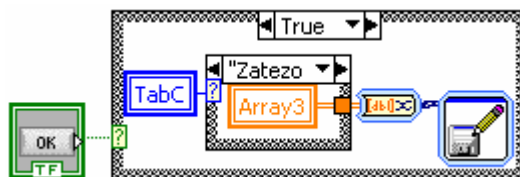


Obrázek 4.39 Varovná struktura



Obrázek 4.40 Mód struktura

Struktura *Ukládací* viz. Obrázek 4.41, je velmi důležitá z hlediska importu dat - uložení naměřených dat do souboru. Aktivuje se stiskem tlačítka *Uložit do souboru*, nacházejícím se na čelním panelu virtuálního přístroje a její funkce je velice prostá. Dle momentálně zvolené záložky prvku *Tab Control* se jménem *TabC* uloží naměřená data, která se ukládají v průběhu měření do prvku *Array*, do souboru, k čemuž slouží prvek *Write To Measurement File*. Nutný je ještě mezikrok převodu dat z 2D pole na dynamická data, což zajišťuje prvek *Convert to Dynamic Data*. Konverze je typu *2D array of scalars – columns are channels*. Uživatel si už jen vybere kam a pod jakým názvem chce data uložit. Vytvořený soubor je typu **.lvm (LabView Measurement)*.

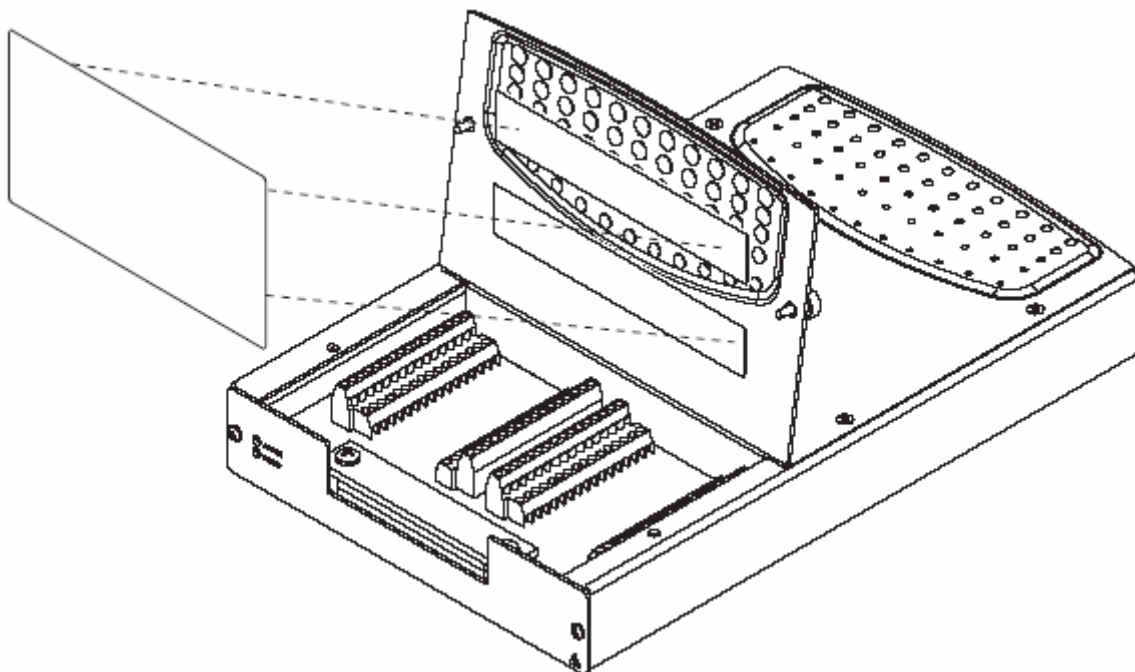


Obrázek 4.41 Ukládací Case struktura

5 PŘÍSTROJE SPOLUPRACUJÍCÍ S VYTVOŘENOU APLIKACÍ

5.1 Měřicí karta NI USB 6251

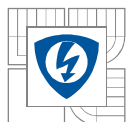
Měřicí karta je nejdůležitější hardwarový prvek, bez kterého by se reálné měření pomocí programu LabVIEW nedalo uskutečnit. Tato karta hardwarově zpracovává analogové měřené veličiny (signály přiváděné na její vstupní kanály). Zpracování obnáší například vzorkování signálu, převod A/D převodníky aj. Karta obsahuje i 2 analogové výstupy, z nichž jeden je použit k řízení elektronické zátěže. Celkem obsahuje 16 analogových vstupů a 24 digitálních vstupů/výstupů. Software s kartou pak komunikuje pomocí tzv. DAQ Assistant (viz. kap. 4.2). Jednotlivé vstupy a výstupy měřicí karty jsou realizovány jako svorkovnice (viz. Obrázek 5.1). K počítači se měřicí karta připojuje přes port USB, napájení má samostatné.



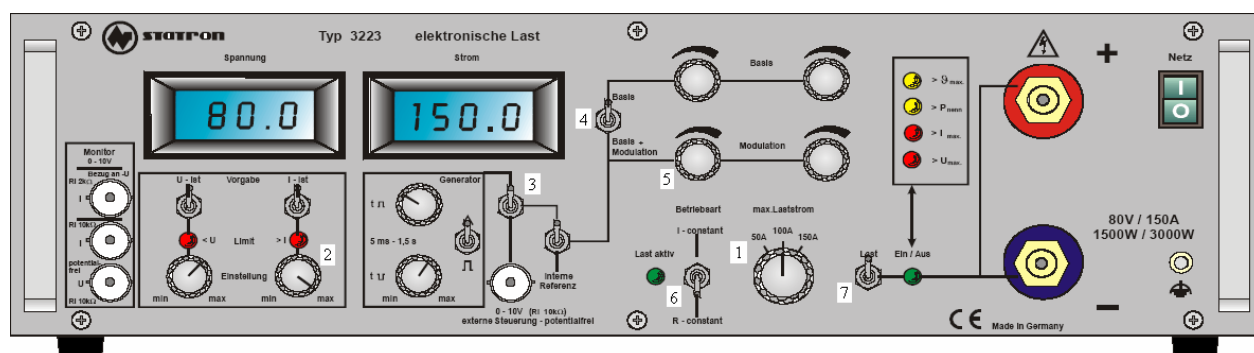
Obrázek 5.1 Měřicí karta [10]

5.2 Elektronická zátěž

Aby elektronická zátěž správně fungovala v době jejího externího ovládání přes vytvořenou aplikaci v LabVIEW je potřeba nastavit určité parametry. Za prvé je to proudové omezení. Pro naše měření postačilo nastavit proudové omezení na 50A případně 100A na hrubém rozsahu viz. Obrázek 5.2 (1) a konkrétní hodnotu na jemném rozsahu (2). Dále je nutné přepnout na externí řízení zátěže, což se děje přepnutím přepínače (3) do polohy dolů. Další přepínač (4) je nutné rovněž přepnout do polohy dolů. Tím se umožní nastavit maximální hodnotu proudu (5), která pro externí řízení představuje 100%. Tím se rozumí, že externím řízením se určuje procentuální část z maximální možné (tedy nastavené) hodnoty proudu, která má téct zátěž. Kde 0V odpovídá 0% a 10V odpovídá 100%. Externí řízení představuje tedy jen externí stejnosměrný napěťový signál. Maximální hodnotu proudu je pro správný chod externího řízení nutné nastavit o něco menší než je proudové omezení. Při externím řízení zátěže pomocí vytvořené aplikace, je pro ni zpětnou vazbou hodnota výstupního napětí alternátoru, které udržuje na hodnotě v rozmezí 13,4V



až 13,6 (viz. kap. 4.3.1.1-třetí algoritmus). Což při zvyšujících se otáčkách činí zvyšováním výstupního proudu a tím napětí snižuje na požadovaných 13,5V. Pakliže by došlo k překročení proudového omezení, elektronická zátěž by přestala reagovat na externí řízení. Mohlo by se tedy stát, že aplikace by se stále pokoušela zvedat hodnotu proudu, jelikož by dostávala informaci o výstupním napětí vyšším než 13,5V, ale elektronická zátěž by na tyto podněty už nereagovala. Což je stav, který není programově nijak ošetřen, ale dá se mu jednoduše předejít správným nastavením elektronické zátěže, která se tak či tak nastavit ručně prostě musí. Proto je třeba nastavení věnovat zvýšenou pozornost. Přepínač (6) se nastaví také do polohy dolů, tzn. do režimu provozu konstantní odpor. Poté stačí už jen připojit zátěž spínačem (7).



Obrázek 5.2 Elektronická zátěž [12][11]

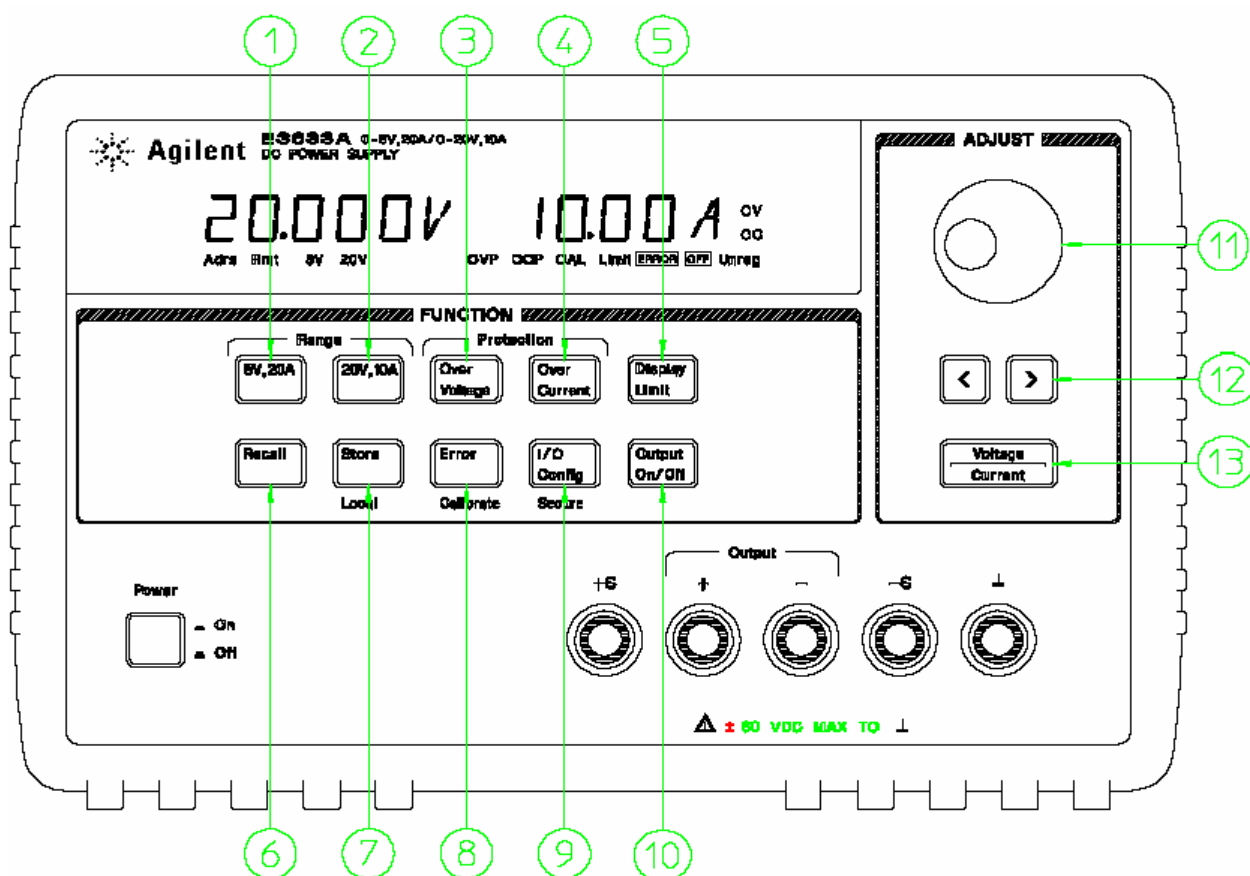
5.3 Zdroj budícího proudu Agilent E3634A

Tento zdroj (viz. Obrázek 5.3) lze ovládat ručně nebo dálkově, díky integrovanému GPIB rozhraní, což zdroji umožňuje komunikovat s okolním světem a jako takový ho lze použít právě pro naši aplikaci. Zdroj lze provozovat ve dvou režimech: Režim konstantního napětí CV (Constant Voltage) a režim konstantního proudu CC (Constant Current). Z hlediska vytvořené aplikace je zajímavý režim CC. Ruční nastavení proudového režimu je následující: Tlačítkem 1 nebo 2 zvolíme požadovaný rozsah zdroje, pak tlačítkem 5 nastavíme potřebný napěťový limit za použití tlačítek 12 a 11, poté tlačítkem 13 přepneme na nastavení proudu a zvolíme požadovanou proudovou hodnotu, opět za použití tlačítek 12, 11 a v posledním kroku zmáčkne tlačítko 10, čímž zapneme výstup zdroje. Za předpokladu, že máme ke zdroji připojen vhodný proměnný odpor, jím začne protékat požadovaný proud. Jakmile však velikost napětí adekvátní požadovanému proudu (závislé na velikosti proměnného odporu zátěže) přesáhne limitně nastavenou hodnotu napětí, zdroj přepne na režim CV a nebude dále proud zvyšovat. Pakliže proměnný odpor nastavíme zpět tak, aby napětí pokleslo pod limitně nastavenou hranici zdroj přejde zpět na režim CC. Všechno nastavení popsané výše lze provádět samozřejmě i pomocí SCPI (Standard Commands for Programmable Instruments) příkazů přes GPIB rozhraní. To je pro vytvořenou aplikaci, z hlediska řízení zdroje, stěžejní. Použité příkazy jsou dvojího typu: *Write*, dá se říct, že říkají udělej! a *write and read*, tyto „položí otázku a čekají na odpověď“. Následující příkazy se nacházejí v programu:

Příkazy typu *write*:

VOLT:RANG P50V(VOLT:RANG P25V) – příkaz nastaví rozsah zdroje na 50V, 4A resp. 25V, 7A.

VOLT 50.0 / VOLT 25.0 - příkaz nastaví napěťový limit na 50V resp. 25V (podle zvoleného rozsahu zdroje, což je v programu ošetřeno), čili nastaví napěťový limit na maximální možný, vzhledem k nastavenému rozsahu zdroje.



Obrázek 5.3 Zdroj Agilent E3634A [11]

CURR 3.0 - příkaz, který nastavuje požadovanou hodnotu proudu na výstupu zdroje.

OUTP:STAT OFF / OUTP:STAT ON – příkazy vypínající / zapínající výstup zdroje.

*RST – tento příkaz resetuje zdroj, v rámci resetu se vypne i výstup zdroje.

Příkazy typu *write then read*:

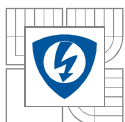
MEAS:CURR? – příkaz se ptá na hodnotu proudu na výstupu zdroje, zdroj odpovídá číselnou hodnotou proudu v datovém formátu *string* např. 3.0

OUTP:STAT? – příkaz, který se ptá na stav výstupu zdroje, zdroj vrácí hodnotu 0=OFF, 1=ON

Pro potřeby vytvořené aplikace nám toto postačí vědět, více informací a detailů o zdroji viz.[11].

5.4 Převodník GPIB-USB

Použití převodníku viz. Obrázek 5.4 je nutné neboť běžné PC ani notebooky nebývají vybaveny rozhraním GPIB, naopak bez USB portu už si v dnešní době tyto počítače ani nelze představit. Použití převodníku v zapojení aplikace je vidět z Obrázku 6.5. Je pro komunikaci zdroje budícího proudu a aplikace nezbytný. Konkrétně je použit převodník firmy National Instruments typ NI GPIB-USB-HS (Hi-Speed). Jedním z doporučených softwarů pro použití tohoto převodníku je přímo LabVIEW, takže volba použít tento převodník je vcelku přirozená. V závislosti na systémové konfiguraci počítače, schopnosti přístroje, ke kterému se převodník připojuje a na verzi USB portu, může přenosová rychlost převodníku dosáhnout až 7,2 MB/s. Více informací viz. [13].

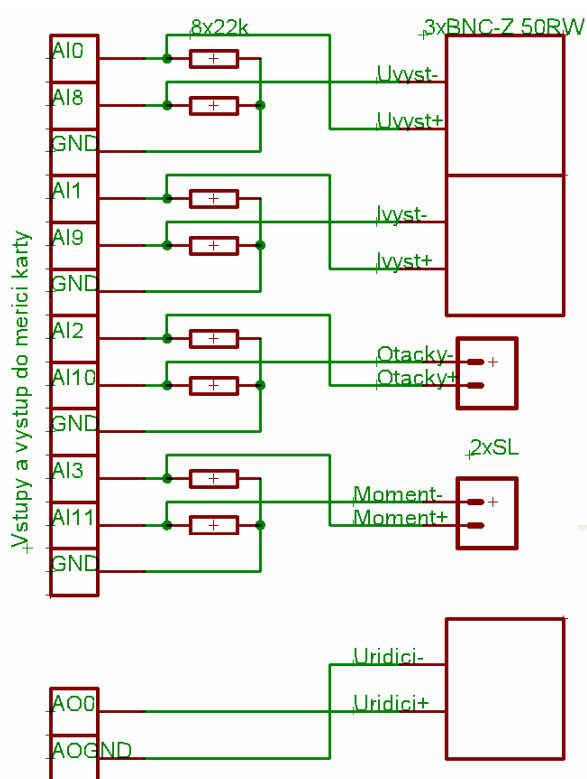


Obrázek 5.4 GPIB-USB převodník [13]

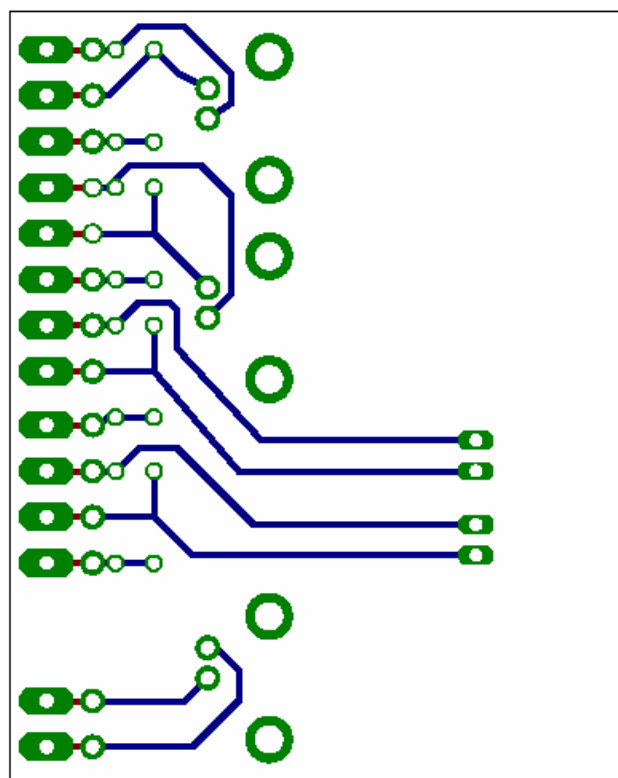
6 MĚŘÍCÍ PŘÍPRAVEK A SAMOTNÉ MĚŘENÍ

6.1 Návrh a realizace přípravku

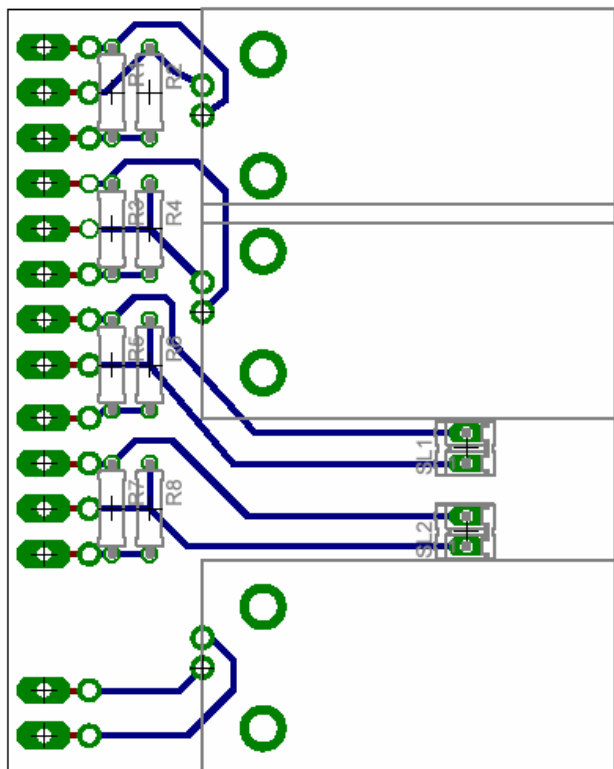
Tento návrh slouží k zamezení zbytečných chyb, které mohou vznikat při zapojování vstupních a výstupních signálů do svorkovnice měřicí karty. Podobný přípravek je dobrým způsobem jak se zmíněným chybám vyhnout. Měřicí karta se totiž využívá pro různé úlohy a měření, takže přepojovat potřebné vstupy a výstupy pokaždé podle toho, jak je navržena konkrétní aplikace pro dané měření v programu LabView, může přinést nadbytečné komplikace. Při existenci obdobného přípravku, který je navržen a realizován tak, aby byl plně asociován s navrženým softwarem, pak už jen jednoduše postačí, aby uživatel zasunul přípravek do svorkovnice měřicí karty a na jeho jasné označené vstupy a výstupy připojí potřebné signály. Uživatel se tak nemusí obávat, že například signál pro budící proud omylem připojí na vstup jiného kanálu než kam patří apod. Přesně takové chyby občas způsobí lidský faktor, a pakliže je možnost se jim vyvarovat, podobné řešení se samo nabízí.



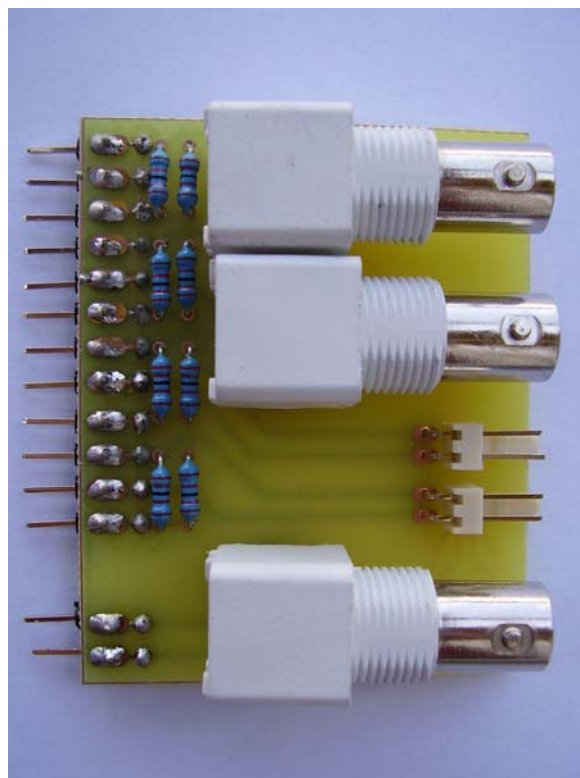
Obrázek 6.1 Schéma zapojení vstupů a výstupů



Obrázek 6.2 Návrh desky plošného spoje



Obrázek 6.3 Osazení DPS součástkami



Obrázek 6.4 Realizace DPS

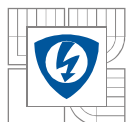
6.2 Měření vytvořeným virtuálním přístrojem

Základní charakteristiky, které se pomocí vytvořené aplikace dají měřit jsou dvě charakteristiky naprázdno $U_{\text{výst}}=f(I_b)$, $U_{\text{výst}}=f(n)$ a dvě charakteristiky zatěžovací $I_{\text{výst}}, M=f(n)$.

Postup měření

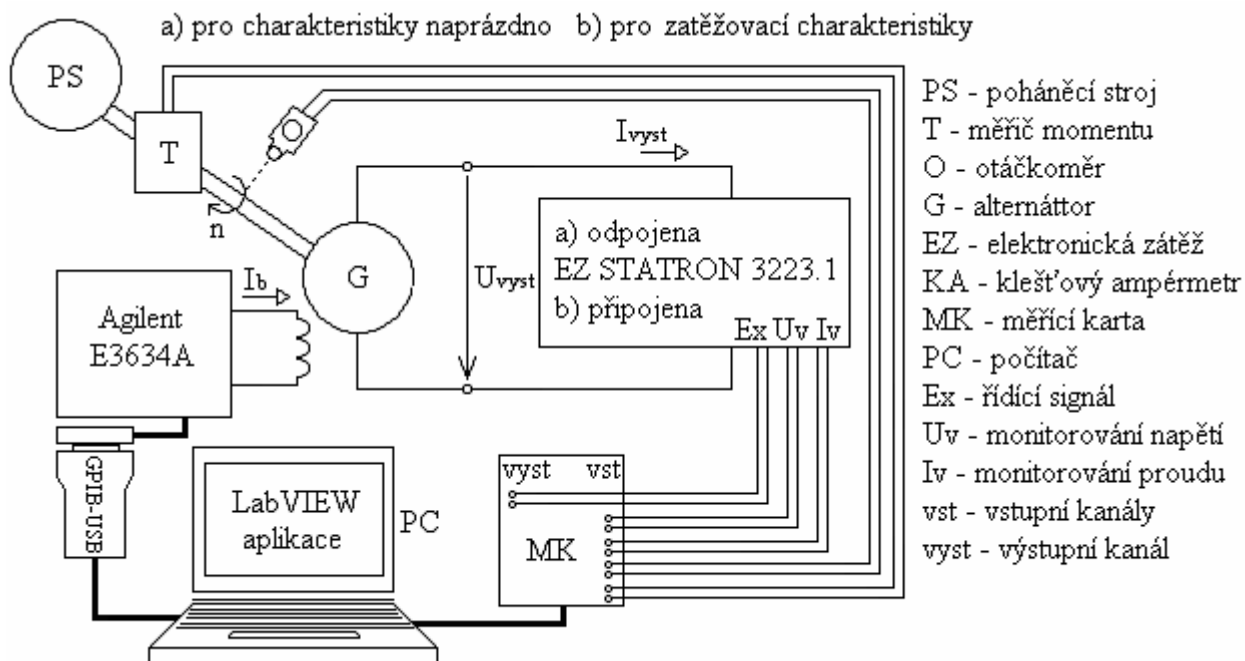
1. Pro konstantní otáčky se postupně nastavuje budící proud se zvoleným krokem a odečítá se výstupní napětí alternátoru příslušné danému budícímu proudu.
2. Při konstantním budícím proudu nastaveném na maximum tj. 3A se postupně zvyšují otáčky a odečítá se příslušné výstupní napětí. Tato charakteristika slouží k odečtení otáček počátku nabíjení, což jsou otáčky, při kterých napětí na výstupu dosáhne hodnoty 13,5V.
3. Zatěžovací charakteristiky se měří tak, že při maximálním budícím proudu (3A) se otáčky alternátoru nastaví na otáčky počátku nabíjení, $U_{\text{výst}}$ by mělo být v tomto okamžiku 13,5V. Pak se dále otáčky po krocích zvyšují a při každém zvýšení se doreguluje hodnota výstupního napětí pomocí proměnné zátěže tak, aby byla stále konstantní tj. 13,5V. Tím se mění výstupní proud. V každém kroku se odečte hodnota výstupního napětí a proudu, momentu a otáček.

Pozn.: Otáčky se nastavovaly ručně, budící proud automaticky a výstupní proud se reguloval rovněž automaticky. Naměřené veličiny se zapisovaly pomocí aplikace vytvořené v LabVIEW. Zapisují se do tabulky, která je vidět na čelním panelu vytvořeného virtuálního měřicího (viz.



Obrázek 4.1) přístroje, případně je možno data na přání uživatele uložit do souboru na pevný disk počítače. Zároveň se na čelním panelu vykresluje i graf. (příklad viz. Obrázek 6.9 a Obrázek 6.10).

Schéma zapojení:



Obrázek 6.5 Zjednodušené schéma zapojení

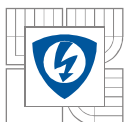
Tabulky naměřených hodnot:

Tabulka 6.1 Závislost U_{vyst} na I_b 1

n=280 min ⁻¹	
U_{vyst} [V]	I_b [A]
0,00	0,00
0,11	0,30
0,76	0,60
1,42	0,90
1,87	1,20
2,19	1,50
2,32	1,80
2,42	2,10
2,51	2,40
2,56	2,70
2,60	3,00

Tabulka 6.2 Závislost U_{vyst} na I_b 2

n=1090 min ⁻¹	
U_{vyst} [V]	I_b [A]
0,00	0,00
1,00	0,20
3,14	0,40
5,08	0,60
6,85	0,80
8,34	1,00
9,53	1,20
10,32	1,40
10,90	1,60
11,27	1,80
11,54	2,00
11,75	2,20
11,93	2,40
12,09	2,60
12,21	2,80
12,31	3,00



Tabulka 6.3 Závislost $U_{výst}$ na n

$I_b = \max. = 3A$	
$U_{výst} [V]$	$n [min^{-1}]$
0,00	0,00
0,44	126,13
1,78	227,55
3,13	331,18
4,30	431,60
5,48	531,24
6,61	634,59
7,94	739,44
9,12	833,69
10,17	926,13
11,56	1025,83
12,99	1130,02
13,92	1223,44
15,51	1322,64
16,31	1425,04
17,74	1537,42
18,67	1634,99

Tabulka 6.4 Závislost I , M na n 1

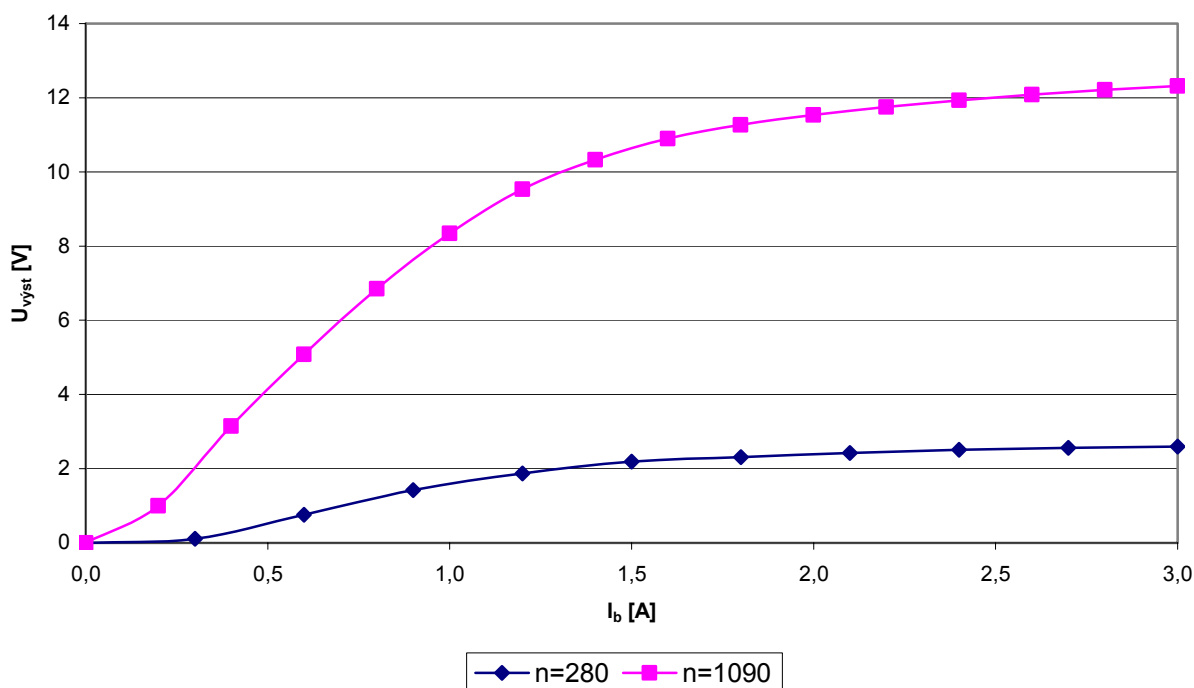
$U_{výst} = 13,5V; I_b = \max. = 3A$		
$I_{výst} [A]$	$M [Nm]$	$n [min^{-1}]$
-0,07	1,25	1156,93
2,95	0,85	1250,84
7,68	0,75	1349,94
11,98	1,32	1452,33
15,95	0,50	1551,10
19,91	0,93	1652,29
22,84	1,27	1751,12
25,76	1,20	1850,66
28,29	1,20	1950,59
30,22	1,08	2054,81
32,28	1,24	2151,09
33,94	1,41	2249,64

Tabulka 6.5 Závislost I , M na n 2

$U_{výst} = 13,5V; I_b = \max. = 3A$		
$I_{výst} [A]$	$M [Nm]$	$n [min^{-1}]$
-0,11	0,31	1153,97
-0,08	0,31	1208,25
2,11	0,86	1271,69
5,51	1,47	1327,93
8,13	1,69	1387,52
10,29	1,69	1449,89
13,27	2,20	1505,40
15,40	2,62	1565,34
18,03	2,58	1630,16
20,93	2,64	1689,34
22,41	2,85	1747,50
24,37	2,97	1811,52

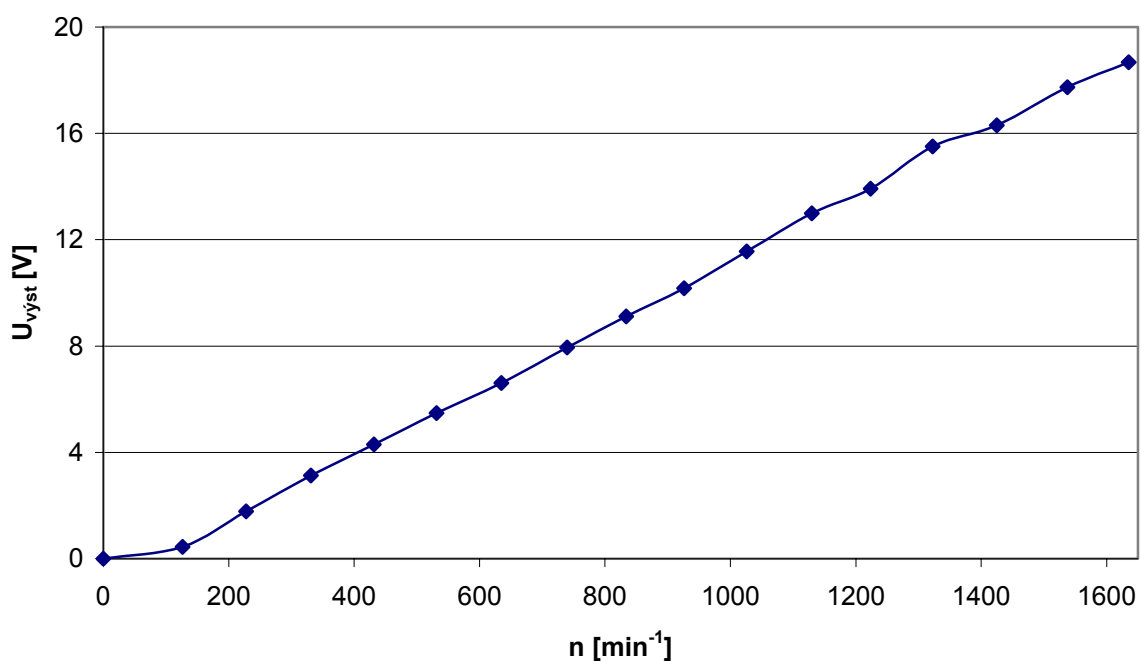
Grafy:

$$U_{\text{výst}} = f(I_b), n = \text{konst.}$$

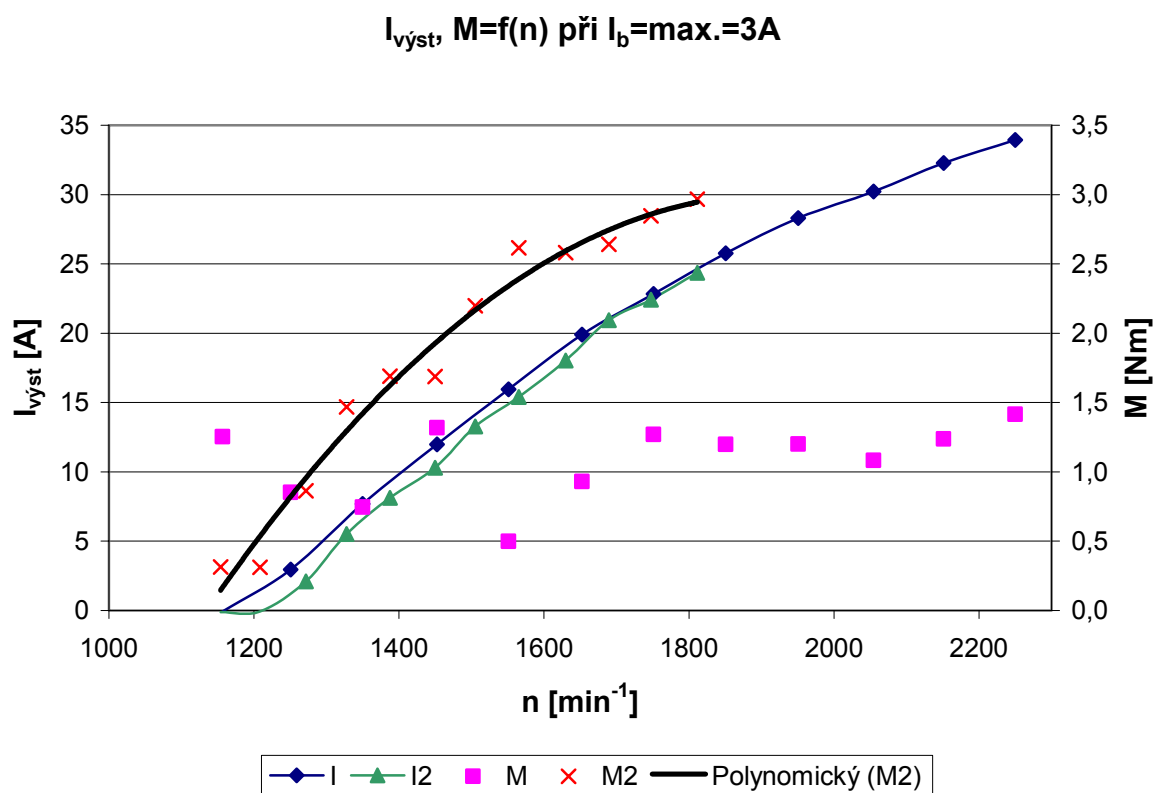
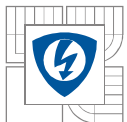


Obrázek 6.6 Charakteristika naprázdno 1

$$U_{\text{výst}} = f(n) \text{ při } I_b = \text{max.} = 3\text{A}$$

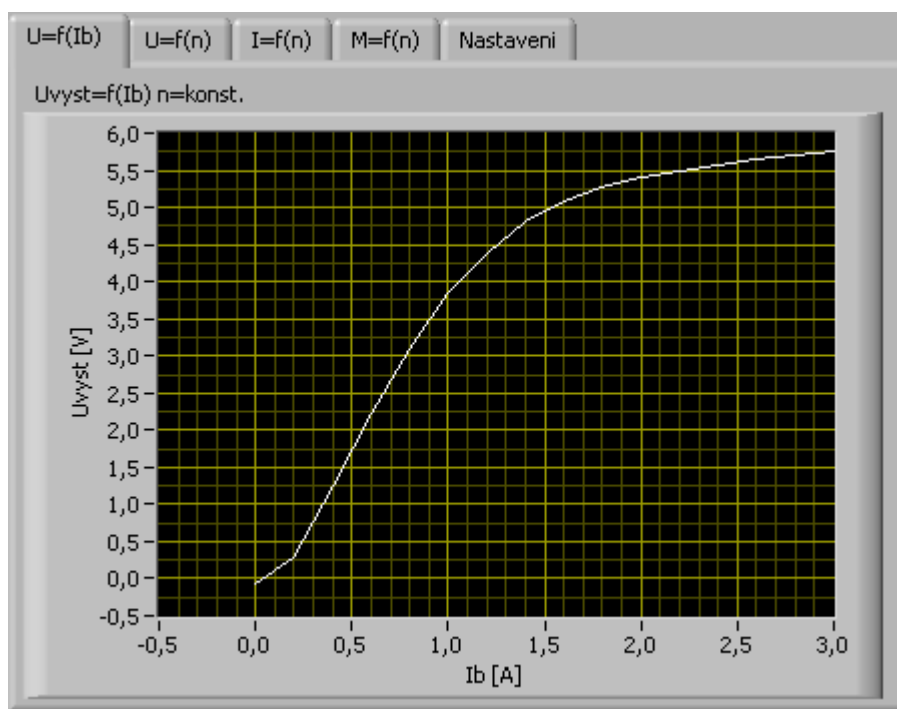


Obrázek 6.7 Charakteristika naprázdno 2

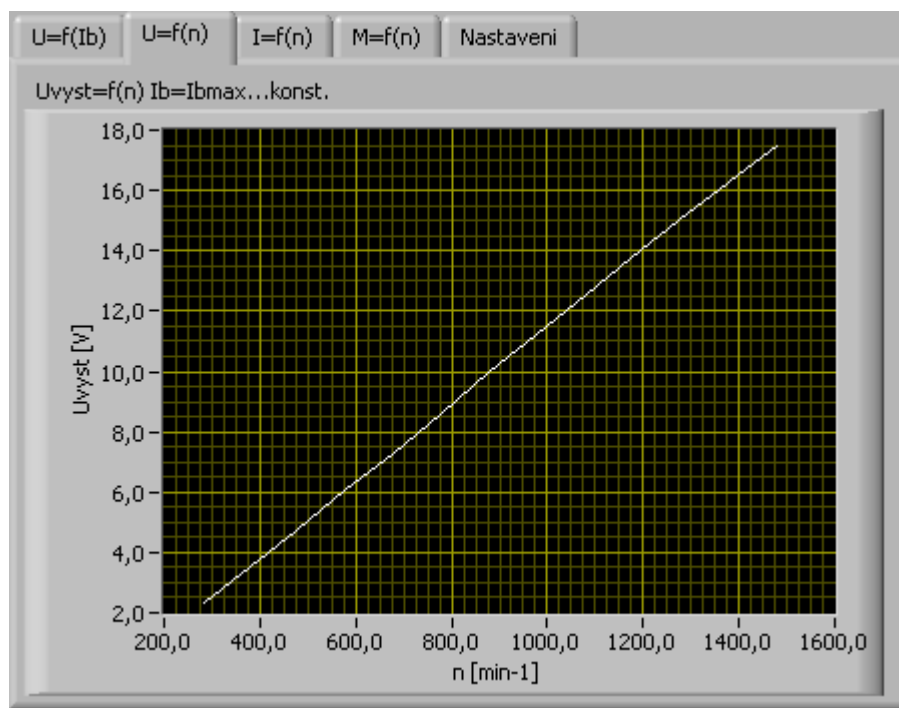


Obrázek 6.8 Zatěžovací charakteristiky

Ukázka grafů z virtuálního přístroje:



Obrázek 6.9 Graf z přístroje 1

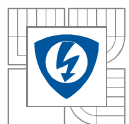


Obrázek 6.10 Graf z přístroje 2

Pozn.: Barevně se ukázkové grafy oproti čelnímu panelu virtuálního přístroje liší neboť byly pořízeny v době kdy virtuální přístroj nebyl ještě plně dokončen, navíc vzhled a design přístroje byla až poslední věc, která se na něm tvořila.

Popis grafů:

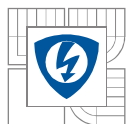
Při měření charakteristiky naprázdno (Obrázek 6.6) se projevují ztráty naprázdno, to jsou ztráty vířivými proudy a hysterezní ztráty. Křivka závislosti $U_{\text{vyst}}=f(I_b)$ připomíná svým tvarem část běžné magnetizační křivky. Indukované napětí je dáno změnou magnetického toku v čase. Magnetický tok je v lineární části charakteristiky přímoúměrný budicímu proudu. V koleně magnetizační křivky se jeho růst zpomalí, což se projeví i na indukovaném napětí. Tím, že se mění budicí proud se mění i hysterezní ztráty a ztráty vířivými proudy. Pro závislost $U_{\text{vyst}}=f(n)$ (Obrázek 6.7) naopak platí, že budicí proud je konstantní a maximální, tím se zajistí nasycení magnetického obvodu alternátoru. Napětí se se zvyšujícími se otáčkami přímo úměrně rovněž zvyšuje a závislost je lineární. Pomine-li se to, že indukované napětí je střídavé a musí se usměrňovat, lze říct, že výstupní napětí je přímo úměrné napětí indukovanému, neboť to je podle rovnice $U_i = c\Phi\omega = c\Phi \cdot \frac{2\pi n}{60}$ přímo úměrné otáčkám. Poslední graf je naměřen pouze pro počáteční část charakteristik $I_{\text{vyst}}, M=f(n)$, to proto, že na proměření celé charakteristiky nebyly v laboratoři příslušné podmínky, např. pro maximální výstupní proud cca 42A je potřeba alternátor otáčet rychlostí okolo 6000min^{-1} . Při takovýchto rychlostech se znatelně projevuje nevycentrování osy hřídele alternátoru vůči ose hřídele poháněcího stroje a mohlo by dojít k jeho mechanickému poškození, což jsme neriskovali. Průběh momentu M , který je evidentně nesmyslný je v grafu úmyslně, pro ukázkou jak mohou vypadat naměřené hodnoty, když se nepoužije měřicí přípravek a selže lidský faktor, jeden z kontaktů signálu pro moment byl totiž špatně připojen. Avšak tato charakteristika byla naměřena spíše jen pro ukázkou, že přístroj měří dobře, než pro vypovídací schopnost dané závislosti, jelikož není změřena zcela dokonce.



7 ZÁVĚR

Cílem diplomové práce bylo vytvořit měřicí aplikaci resp. virtuální měřicí přístroj, který dokáže automaticky naměřit charakteristiky drápkového alternátoru, včetně automatického zápisu hodnot do tabulek, jejich případné uložení do souboru a vykreslování grafu dané charakteristiky. Ke správné funkčnosti automatizovaných pochodů je nejdůležitější, kromě vymyšlení samotného algoritmu, jeho realizace pomocí vhodných objektů nabízejících se ve vývojovém prostředí LabView. Měření charakteristik je do určité míry pouze poloautomatické, jelikož ke kompletnímu zautomatizování chybí možnost nastavování rychlosti otáček poháněcího stroje. Avšak plně automatická je regulace výstupního napětí (resp. proudu) při měření zatěžovacích charakteristik. Nastavování velikosti budícího proudu je rovněž plně automatické – pro měření všech charakteristik. Vytvořená aplikace zohledňuje manuální nastavování otáček potažmo tedy i eventualitu vzniku lidské chyby nejen při této činnosti. Tzn., že je proti nepříznivému lidskému faktoru do určité míry velmi odolná. Pochopitelně, pakliže uživatel např. fyziky vypojí ze zásuvky budící zdroj nebo elektronickou zátěž, je proti tomu aplikace bezbranná. Vytvořená aplikace umožňuje i ruční chod měření, kdy si uživatel může nastavovat potřebné veličiny téměř dle libosti. I tento režim je samozřejmě ošetřen proti lidským chybám, či případné neznalosti problému. V obou režimech funguje inteligentní systém blokování ovládacích prvků tak, aby byly k dispozici vždy jen ty, které jsou k danému měření konkrétní charakteristiky nejnutnější. Při spuštění měření je uživatel vizuálně varován, jak textem v informačním panelu, tak blikajícími diodami, aby zvýšil svou obezřetnost a pozornost. Informační panel a systém zpráv na displej (tzv. „OK“ zprávy) uživateli usnadňují používání přístroje, které je samo o sobě navrženo tak, aby bylo velmi intuitivní. Zároveň uživatel vidí přehledně všechny měřené hodnoty na číselných displejích čelního panelu virtuálního měřicího přístroje i hodnoty zapsané v tabulce a vykreslený graf. V případě automatického režimu měření je jediným úkolem uživatele nastavit požadované parametry před spuštěním měření konkrétní charakteristiky, popřípadě měnit hodnotu otáček. Dále je v přístroji integrován systém ochrany, který při překročení některé ze sledovaných veličin (výstupní napětí, výstupní proud, otáčky) zareaguje stáhnutím budícího proudu na nulu, případně stáhne na nulu i elektronickou zátěž, probíhalo-li by zrovna měření zatěžovací charakteristiky. Hodnoty naměřené v laboratoři, vytvořeným virtuálním přístrojem, jsou patrné z tabulek a grafů v kapitole 6.2. Obdržené průběhy grafů jsou zde přiloženy pro názornou ukázkou, že virtuální přístroj opravdu měří tak jak má a to co má. To jsme si ověřili již v průběhu samotného měření, kdy se hodnoty zobrazované na panelu virtuálního přístroje shodovaly s hodnotami skutečných fyzicky přítomných měřicích přístrojů.

Další vylepšení aplikace se přímo nabízí. Zautomatizovat i řízení rychlosti otáček poháněcího stroje. Také se může doplnit o další vstupy, které budou měřit např. proudy ve všech fázích alternátoru či na diodách usměrňovače. Případně se může aplikace doplnit o výpočetní bloky, které budou z naměřených veličin počítat výkon alternátoru, jeho účinnost a vše opět rovnou vykreslovat do grafů.



LITERATURA

- [1] Alternátory Startéry a náhradní díly alternátorů a startérů. [online]. 2008. Dostupné z: http://www.mjauto.cz/alternator_starter.htm.
- [2] Alternátory s budícím vinutím. [online]. 2008. Dostupné z: <http://vyuka.fel.zcu.cz/kae/NSA/Texty/Alternatory.pdf>.
- [3] HÁJEK, V. Automobilová elektrotechnika a elektronika. 2006. Slajdy k předmětu Automobilová elektrotechnika.
- [4] ŽÍDEK, J. Grafické programování ve vývojovém prostředí LabVIEW. VŠB-TU Ostrava. Říjen 2002
- [5] LabVIEW Measurement Manual. 2003.
- [6] LabVIEW User Manual. 2003.
- [7] LabVIEW™ Basics I Course Manual. 2000.
- [8] LabVIEW™ Basics II Course Manual. 2000.
- [9] LabVIEW Help. 2009.
- [10] National Instruments - Test and Measurement [online]. [cit. 2009-05-17]. Dostupný z WWW: <http://www.ni.com/pdf/manuals/371022k.pdf>.
- [11] Agilent technologies [online]. 2003-2009 [cit. 2009-05-15]. Dostupný z WWW: http://www.home.agilent.com/agilent/redirector.jsp?action=obs&nid=-35690.0.00&lc=eng&cc=US&ckey=1000000669-1%3Aepsg%3Aman&pubno=E3634-90001<ype=&ctype=AGILENT_EDITORIAL.
- [12] BD SENSORS s.r.o. [online]. [cit. 2009-05-16]. Dostupný z WWW: <http://www.bdsensors.sk/Katalog%20Statron/skupina-J.pdf>.
- [13] National Instruments - Test and Measurement [online]. [cit. 2009-05-17]. Dostupný z WWW: <http://www.ni.com/pdf/products/us/20055194101101dlr.pdf>.