



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

LABVIEW DRIVERY PRO ODPOROVÉ DEKÁDY

LABVIEW DRIVERS FOR RESISTOR DECADE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Matěj Procházka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Marie Havlíková, Ph.D.

BRNO 2016

Bakalářská práce

bakalářský studijní obor Automatizační a měřicí technika
Ústav automatizace a měřicí techniky

Student: Matěj Procházka
Ročník: 3

ID: 125294
Akademický rok: 2015/16

NÁZEV TÉMATU:

LabVIEW drivery pro odporové dekády

POKYNY PRO VYPRACOVÁNÍ:

1. Nastudujte a popište problematiku tvorby driverů v prostředí LabVIEW s popisem jednotlivých kroků nutných při jejich tvorbě.
2. Seznamte se s funkcemi dekád řady M63X a M64X a jejich možnostmi vzdáleného ovládání.
3. Navrhněte způsob odzkoušení driveru.
4. Realizujte funkční Drivery LabVIEW pro dekády M63x a M64x.
5. Realizujte Exemplary LabVIEW pro základní funkce dekád M63x a M64x s aplikací těchto ovladačů.

DOPORUČENÁ LITERATURA:

Technická dokumentace k odporovým dekádám M63X a M64X

Termín zadání: 8.2.2016

Termín odevzdání: 23.5.2016

Vedoucí práce: Ing. Marie Havlíková, Ph.D.

Konzultant bakalářské práce: Ing. Tomáš Kapinus

doc. Ing. Václav Jirsák, CSc., předseda oborové rady

UPOZORNĚNÍ:

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

Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně / Technická 3058/10 / 616 00 / Brno

Klíčová slova:

Odporová dekáda M63x, funkce, odpor, platina, nikl, ovladač, teplota.

Klíčová slova - anglicky:

Resistance decade M63x, function, resistance, platinum, nickel, driver, temperature.

Abstrakt:

Tato bakalářská práce se zabývá popisem odporové dekády M63x a M64x od firmy Meatest. Popisují se zde funkce a možnosti vzdáleného přístupu k dekádě. Dále je zde rozebráno, jak se vytváří driver ve vývojovém prostředí LabVIEW. Ke konci je uvedeno, jak jsem prováděl testování driveru.

Abstrakt- anglicky:

This bachelor's thesis deals with description resistance decade M63x and M64x from Meatest company. I described functions and possibilities of remote access. Further are described how it's developed instrument driver in LabVIEW. At the end is cited how I conduct with testing this driver.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma LabVIEW drivery pro odporové dekády jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2016**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Marii Havlíkové, Ph.D. a odbornému konzultantovi Ing. Tomášovi Kapinusovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **20. května 2016**

.....
podpis autora

Bibliografická citace mé práce:

PROCHÁZKA, M. *LabVIEW drivery pro odporové dekády*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 43s. Vedoucí bakalářské práce Ing. Marie Havlíková, Ph.D.

Obsah

Obsah	7
1 Úvod.....	9
2 Softwarové ovladače pro měřicí přístroje	10
2.1 Obecné informace o přístrojovém ovladači	10
2.1.1 Interactive Interface	11
2.1.2 Application Programming Interface (dále API).....	11
2.1.3 Virtual Instrument Software Architecture (dále jen VISA).....	11
2.1.4 Instrument driver.....	11
2.1.5 Support VIs	11
3 Problematika tvorby driveru v LabVIEW.....	12
3.1 Vývojové prostředí LabVIEW	12
3.2 Přístrojový ovladač v LabVIEW	13
3.2.1 Získání přístrojového ovladače	14
3.2.2 Komunikace s přístrojem	14
3.2.3 Postup tvorby přístrojového ovladače v LabVIEW	15
4 Odporové dekády řady M630 a M630A	18
4.1 Obecné informace	18
4.2 Přední panel odporové dekády řady M63x.....	18
4.3 Zadní panel dekády řady M63x.....	20
4.4 Zapnutí	20
4.5 Volba funkce	20
4.5.1 Funkce RESISTANCE - ODPOR.....	21
4.5.2 Funkce PLATINUM - PLATINA.....	21
4.5.3 Funkce NICKEL - NIKL	21
4.5.4 Funkce USER.....	22
4.5.5 Funkce TIMING - SEKVENCE	22
4.6 Vysvětlení přepínacích módů v závislosti na změně odporu.	23
4.7 Vlastnosti odporových dekád	23
4.8 Komunikační rozhraní.....	23
4.8.1 Ovládání dekády pomocí RS232.....	23
4.8.2 Ovládání pomocí sběrnice GPIB.....	24
4.9 Rozdíl mezi řadou M63x a M64x	25
5 Tvorba ovladače pro dekády řady M63x a M64x	26
5.1 Tvorba vzorových příkladů použití funkcí pro ovládání dekády	32
5.1.1 Pravidla pro tvorbu příkladů použití	32
5.1.2 Funkce Odpor.....	33

5.1.3	Funkce Platina.....	34
5.1.4	Funkce Nikl.....	35
5.1.5	Funkce <i>Timing</i>	36
5.1.6	Funkce User Function	37
6	Ověření funkčnosti vytvořeného ovladače.....	38
6.1	Časová náročnost driveru	38
6.2	Virtuální instrument	38
6.3	Error Handling.....	38
6.4	Funkční testování driveru.....	39
6.4.1	Spuštění LabVIEW	39
6.4.2	Navolení potřebné hodnoty a funkce	39
6.4.3	Spuštění aplikace	39
6.4.4	Vykonání příslušné operace	39
6.4.5	Ukončení vzdáleného přístupu.....	40
7	Závěr	41
	Seznam zkratk	42
	Literatura.....	43

1 Úvod

Ovladače, které se v dnešní době více a více rozšiřují, plní jednoduchou úlohu, pomáhají nám vyvarovat se chyb v obsluze přístroje a šetří náš čas. Je mnoho druhů vývojových prostředí pro různé druhy ovladačů na různé instrumenty. Pokud se ovšem nacházíme ve výrobním, výzkumném nebo lékařském prostředí, tak nejčastější vývojové prostředí je LabVIEW. Pro tuto práci byla použita verze trial LabVIEW pro studenty. Firma Meatest si vyžádala vyhotovení driveru v prostředí LabVIEW pro přístroje, které sama vyrábí. Je to pochopitelné, prostředí LabVIEW je čím dál více využívané a velice intuitivní z pohledu uživatele i programátora. Mým úkolem bylo tento driver zhotovit. V první části bakalářské práce se věnuji obecnému popisu driverů. Další kapitola popisuje, jak se daný driver programuje, od popisu programového prostředí až k samotným blokům driveru. Poslední kapitola je věnovaná testování driveru.

2 Softwarové ovladače pro měřicí přístroje

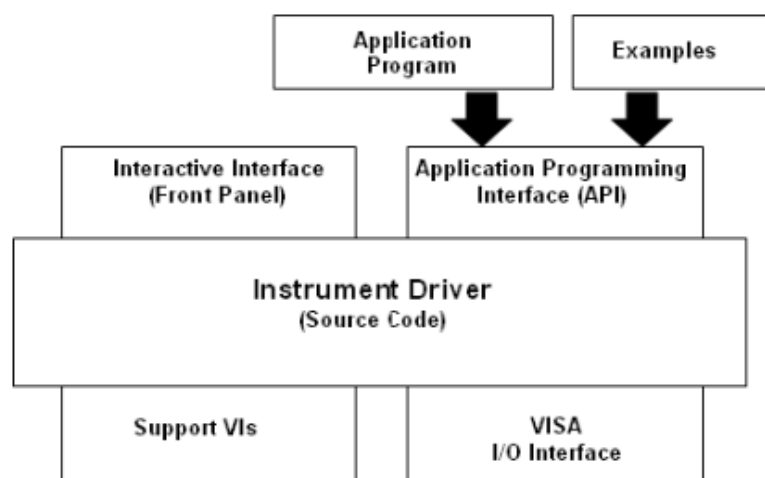
Tato kapitola popisuje Obecné informace o přístrojovém ovladači. Je zde rozebrán obecný model ovladače.

Softwarové ovladače plní důležitou činnost a to takovou, že zprostředkovávají komunikaci mezi uživatelem a přístrojem na základě naprogramovaných funkcí, které jsou poté vhodně aplikovány v příkladech použití v každém ovladači. S příchodem přístrojových ovladačů se ve všech směrech zefektivnila výroba. Oproti běžnému ovládání přístroje je přístrojový ovladač rychlejší a výskyt chybového zadání hodnoty je menší. Další jeho výhodou je rychlá a snadná možnost přidání nových uživatelem volaných funkcí a tím i nových příkladů použití.

2.1 Obecné informace o přístrojovém ovladači

Pro vytvoření správného driveru je zapotřebí podívat se na problematiku z širší perspektivy. Není cílem vytvořit jeden funkční jednoúčelový program, ale ovladač, díky kterému jsme schopni vytvořit ovládací nástroj dle našich představ. Proto musíme postupovat podle obecných pokynů pro tvorbu driveru [4].

Přístrojový ovladač (Instrument driver), dále jen ID, je soubor vyšších funkcí (vyšších proto, že se využívají napříč celým rozhraním). Dále v sobě obsahuje nižší funkce, které slouží například pro komunikaci v GPIB(General Purpose Interface Bus), sériové lince, nebo USB (Universal Serial Bus). Ukázkový model ID je zobrazen na obrázku 1.



Obrázek 1 Obecný model přístrojového ovladače [4]

2.1.1 Interactive Interface

Slouží jako podpora pro uživatele driveru, aby pochopil vnitřní logiku samotného driveru. Nachází se zde ovládací rozhraní.

2.1.2 Application Programming Interface (dále API)

API má v sobě obsaženy uživatelem volané funkce, jako je například nastavení hodnoty odporu pro odporové dekády, nebo funkce pro sepnutí výstupních svorek. Dále také obsahuje příklady použití a to jsou ukázky programů, které jsou sestaveny právě z uživatelem volaných funkcí, které programátor naprogramuje. Tyto příklady použití poté komunikují s přístrojem a představují využitelnost implementovaných funkcí v API.

2.1.3 Virtual Instrument Software Architecture (dále jen VISA)

Jedná se o knihovnu funkcí, která podporuje komunikaci na nejběžnějších sběrnících, které se využívají u měřicích přístrojů. Jedná se o GPIB, RS232, USB a další. Zprostředkovává vstupní a výstupní komunikaci.

2.1.4 Instrument driver

V bloku *Instrument driver* na obrázku 1 se nachází veškerý zdrojový kód přístrojového ovladače. Obsahuje subpalety, které podporují funkcionální přístrojového ovladače. Detailně je tento blok popsán v podkapitole 3.2 Přístrojový ovladač v LabVIEW

2.1.5 Support VIs

Přístrojové ovladače mají pevnou hierarchii se kterou jsou psány. Využívají standardních programovacích technik a proto mohou tyto ovladače přistupovat a používat funkce z jiných knihoven a to právě prostřednictvím bloku *Support VIs*. Nacházejí se zde podporovatelé funkce pro matematické operace, porovnávací funkce, operace se znakovými řetězci a jiné.

3 Problematika tvorby driveru v LabVIEW

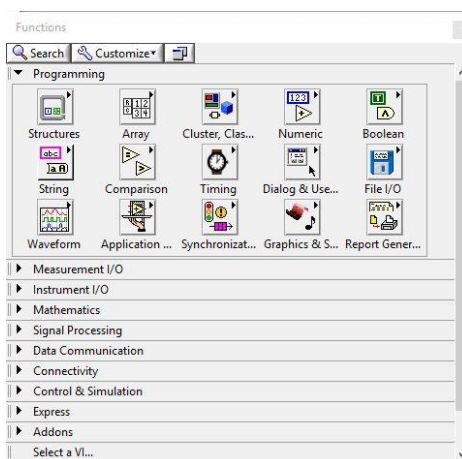
Tato kapitola je věnována vývojovému prostředí LabVIEW, jak s ním pracovat a teoreticky je zde popsáno, jak tvořit drivery pro laboratorní přístroje. První podkapitola je věnována programu LabVIEW a druhá kapitola popisuje přístrojový ovladač obecně.

3.1 Vývojové prostředí LabVIEW

Jedná se o grafické programovací prostředí u kterého je program definován blokovým diagramem. Řádky kódu tedy nejsou psány jako při programování ve VisualStudiosu v C nebo C++. LabVIEW je vyvíjeno firmou National Instruments, aktuální podoba LabVIEW vzdáleně připomíná Simulink v Matlabu.

LabVIEW je rozděleno do dvou oken. První okno je již zmíněný blokový diagram, kde je vyobrazena logická část aplikace, druhé okno je ovládací panel aplikace, kde dochází k ovládání aplikace uživatelem.

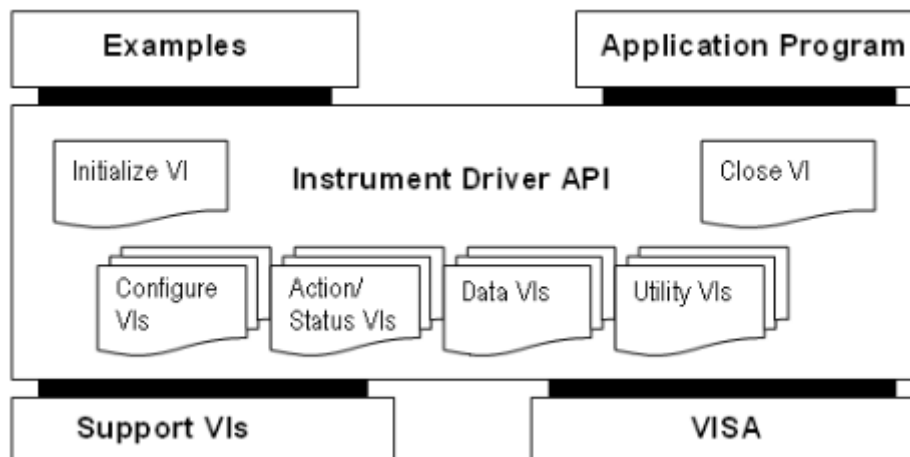
Při programování se jednotlivé funkce a programovací struktury vybírají z "FunctionsPalette", dále jen paleta (viz obrázek 2), která je dělena do jednotlivých záložek. Každá záložka na paletě skrývá specifické nástroje pomocí kterých lze vytvořit aplikaci.



Obrázek 2 Paleta

3.2 Přístrojový ovladač v LabVIEW

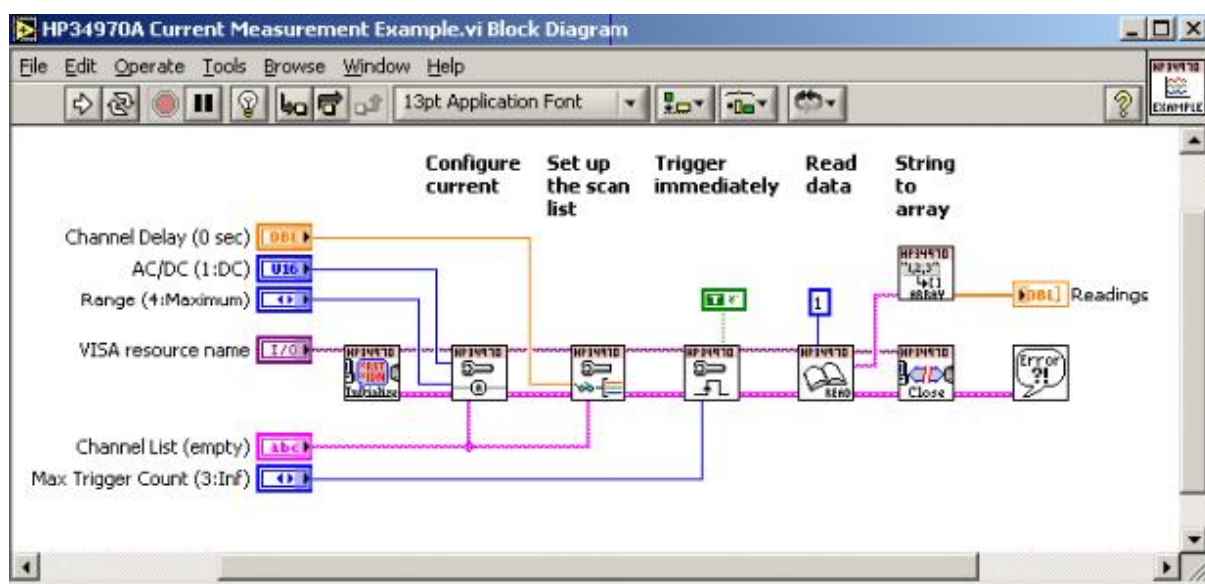
Obecně vytvořený přístrojový ovladač (Instrument Driver – ID) pro jakékoliv zařízení by měl vypadat podle modelu na obrázku 3.



Obrázek 3 API model přístrojového ovladače v LabVIEW [4]

V kořenovém adresáři je funkce *Initialize VI*, která zajišťuje nastavení komunikace, někdy je zde také řešena identifikace přístroje. Funkce *Close VI* slouží k ukončení celé komunikace. Kromě toho jsou zde čtyři subpalety: *Configure*, která by měla obsahovat nastavitelné prvky a která může obsahovat dodatečné nastavení komunikace. Další subpaletou je *Action/Status*, zde dochází ke spuštění funkcí a zahájení měření. Subpaleta *Data* zahrnuje funkce pro přeposílání dat do přístroje nebo čtení dat z přístroje. Funkce typu self-test, reset, nebo třeba kalibrace obsahuje poslední subpaleta *Utility*.

Na obrázku 4 je vidět ukázkový příklad s názvem Current Measurement od společnosti HP, jedná se o přístroj HP34970A.



Obrázek 4 Ukázka vzorového příkladu Current Measurement pro přístroj HP34970A [4]

3.2.1 Získání přístrojového ovladače

Nejčastější typ driveru je Plug and Play. Jedná se o driver, který se pomocí implementovaného vyhledávače v prostředí LabVIEW (Tools→Instrumentation→Find Instrument Drivers) nejprve vyhledá, poté jednoduše pomocí tlačítka *Install* nainstaluje a následně je driver připraven k použití.

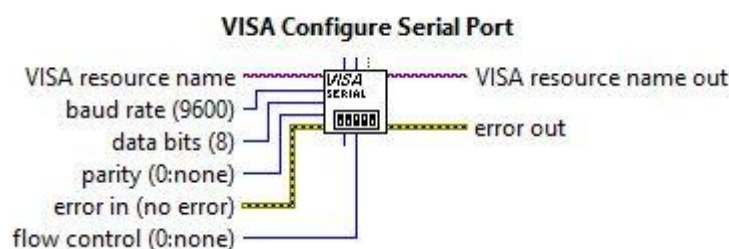
Další možností je využití samotných vzorových příkladů, které nepotřebují další funkce (subVI) ke svému chodu. Lze je vyhledat např. na stránkách czech.ni.com [4].

Dalším zdrojem přístrojových ovladačů je externí odkaz na stránkách výrobce. Přes tento odkaz si uživatel může stáhnout zip archiv driveru, který rozbalí do složky určené pro drivery v LabVIEW adresáři (Program files→National Instruments→instr.lib).

3.2.2 Komunikace s přístrojem

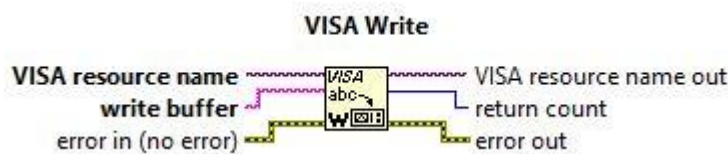
Je důležité si předem stanovit, zda komunikace bude probíhat přes sériovou linku, nebo jiné komunikační rozhraní.

Většina přístrojů má sériovou linku RS232. Nejprve proběhne konfigurace komunikačního rozhraní. Pro tento účel se používá nástroj *VISA Configure Serial Port* (viz obrázek 5), který je v záložce Instruments I/O → Serial. Pomocí tohoto nástroje můžeme nastavit rychlost komunikace, počet bitů, paritu a další.



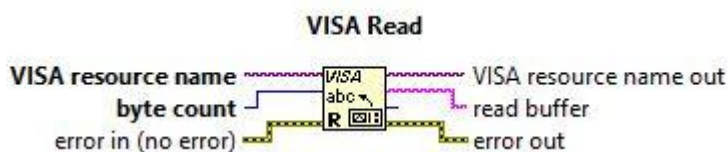
Obrázek 2 VISA Configure Serial Port

Dalším krokem je vytvoření nástroje pro odeslání příkazu do přístroje. K tomu je určen nástroj *VISA Write* (viz obrázek 6), který je ve stejné záložce jako *VISA Configure Serial Port*. Tento nástroj, který je umístěn v záložce Instruments I/O → VISA, posílá řetězec *string* do připojeného zařízení.



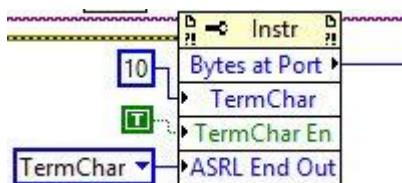
Obrázek 3 VISA Write

Potřebným nástrojem je také *VISA Read* (viz obrázek 7). Tento nástroj čte zpětnou vazbu z přístroje, pokud tedy přístroj nějakou zpětnou vazbu odesílá, jinak je nečinný. Je umístěn v záložce Instruments I/O → VISA.



Obrázek 4 VISA Read

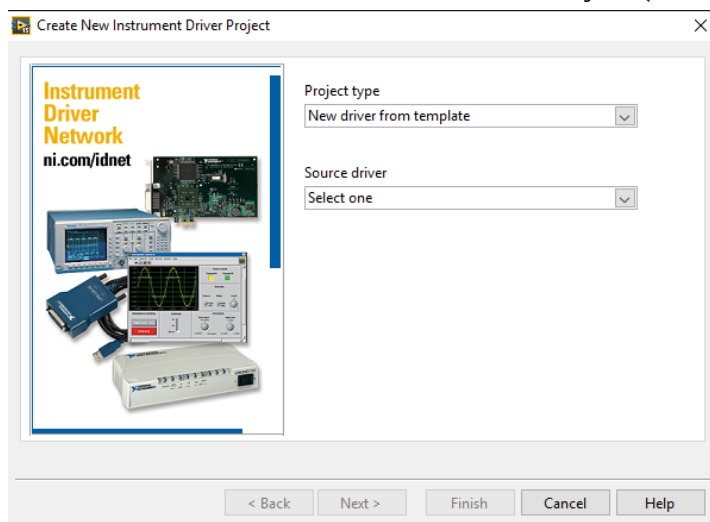
Dalším důležitým nástrojem je *Property Node* (viz obrázek 8), ve kterém lze nastavit ukončovací znak. Tento ukončovací znak je uveden vždy na konci každého odesílaného příkazu. Daný nástroj je dále důležitý pro nastavení nástroje *VISA Read*, kterému předává počet bytů, které jsou právě dostupné, jako aktuální zpětná vazba. Je umístěn v záložce Instruments I/O → VISA → Advanced.



Obrázek 5 Property Node

3.2.3 Postup tvorby přístrojového ovladače v LabVIEW

Po spuštění vývojového prostředí LabVIEW je vytvořen nový projekt pro tvorbu přístrojového ovladače. Ten lze nalézt v liště *Tools* → *Instrumentation* → *Create Instrument Driver Project*. Po potvrzení se zobrazí okno Instrument Driver Project (viz obrázek 9).

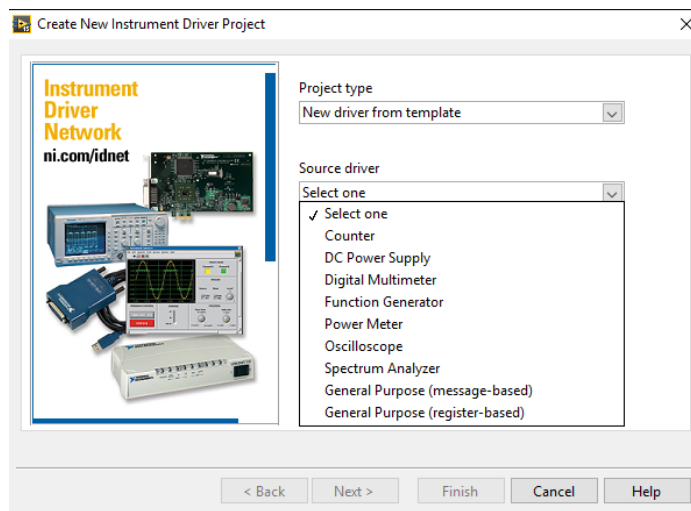


Obrázek 9 Okno - New Instrument Driver Project

Zde je na výběr, zda vytvořit nový ovladač za pomoci šablony, která je dostupná od firmy National Instruments, nebo vytvořit nový ovladač kopií jiného, již existujícího ovladače. První varianta má určité výhody. Při tvorbě prvního ovladače je lepší začínat

s přednastavenou paletou funkcí a tím je zajištěno, že nebudou přenášeny případné chyby z již existujícího ovladače.

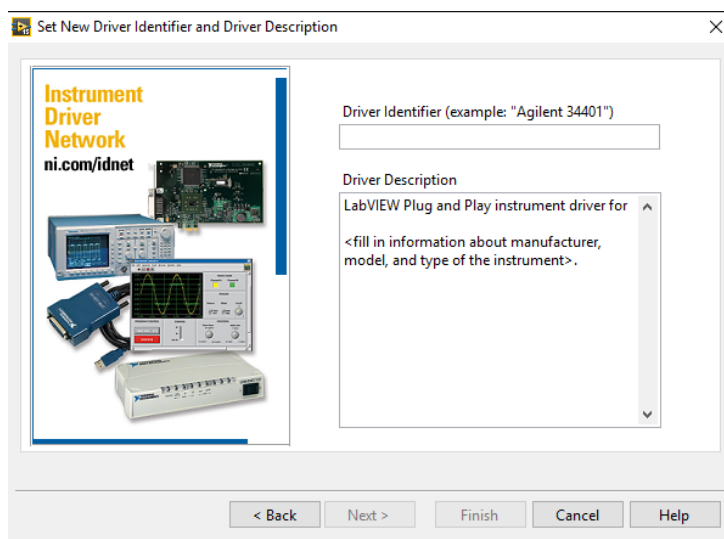
Dalším krokem je výběr šablony, podle které lze daný ovladač vytvářet (viz obrázek 10).



Obrázek 60 Vybrání šablony

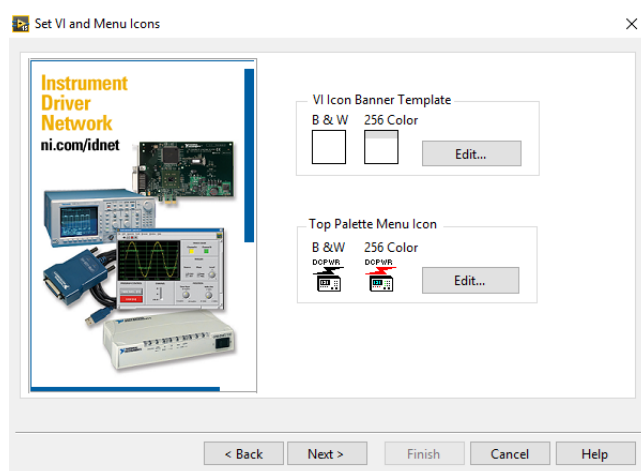
Zde je k dispozici devět variant předpřipravených šablon. Pokud je vyvíjen driver pro čítač, tak je vhodné zvolit šablonu Counter. V případě, že by byla zvolena jiná šablona, která neodpovídá skutečnému přístroji, tak se vytvoří funkce, které nebudou využity. Šablona je vytvořena tak, aby usnadňovala další kroky. To znamená, že pokud je vybrána šablona *Oscilloscope*, tak LabVIEW vytvoří přístrojový ovladač s odpovídajícími příklady pro nastavování různých signálů atd.

Po vybrání příslušné šablony se zobrazí okno, kde je nutné vyplnit jméno budoucího ovladače a přidat krátký popis (viz obrázek 11).



Obrázek 117 Obrazovka pro zadání jména a popisku ovladače

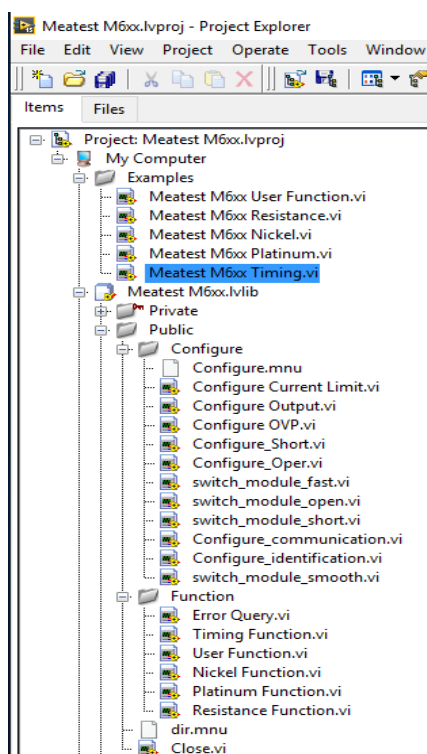
Po vyplnění se zobrazí další okno (viz obrázek 12), kde je možné změnit zobrazovací ikonku ovladače.



Obrázek 12 Obrazovka pro nastavení ikony

Po potvrzení se dále LabVIEW dotáže, kam bude budoucí driver uložen a po stisku tlačítka *Finish* se automaticky vytvoří prototyp driveru.

Typická struktura přístrojového ovladače vytvořeného v prostředí LabVIEW je na zobrazena obrázku 13.



Obrázek 13 8Projekt přístrojového ovladače v prostředí LabVIEW

Tímto je založení projektu přístrojového ovladače v LabVIEW ukončeno a nyní je nutné vytvořit konkrétní funkce (VI) pro ovládání přístroje a vzorové příklady pro jejich použití.

4 Odporové dekády řady M630 a M630A

V této kapitole jsou popsány dva druhy odporových dekád, pro které jsou realizovány přístrojové ovladače. Nejprve jsou uvedeny obecné informace, poté podrobně popsán přední i zadní panel přístroje a jeho vybavenost. Dále je popsáno chování dekády při samotném spuštění a volba funkcí. Závěr kapitoly je věnován komunikačnímu rozhraní a vzdálené komunikaci.

4.1 Obecné informace

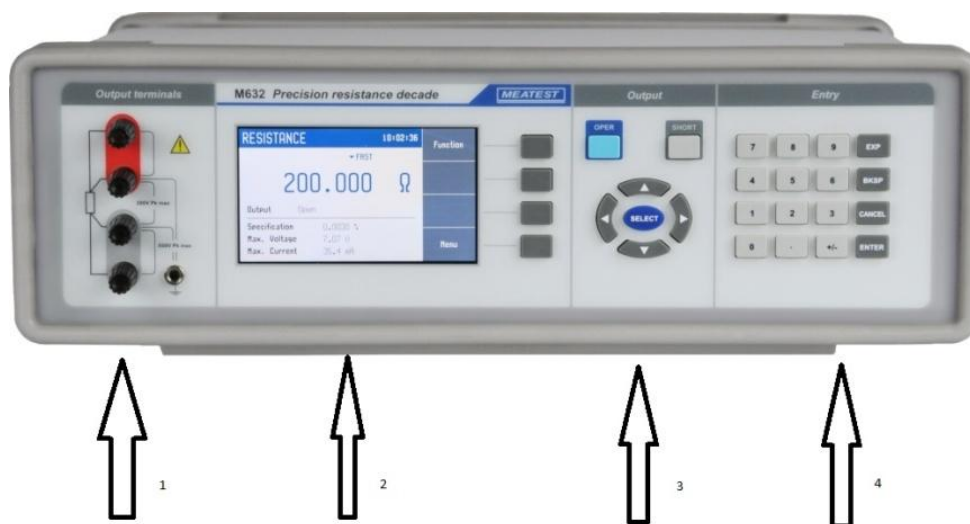
Odporová dekáda M630 a její rozšířená verze M630A jsou vhodné pro simulaci různých druhů odporových čidel (síla, dráha, teplota atd.). Tyto dekády umožňují eliminovat nestabilitu nulové hodnoty odporu R_0 .

Během výroby byly použity součástky, jako jsou například: relé s nízkým termoelektrickým napětím nebo stabilní foliové rezistory s nízkým teplotním koeficientem. Zobrazení hodnot je zprostředkováno pomocí TFT (Thin Film Transistor) displeje s vysokým rozlišením. M630 obsahuje rekalibrační proceduru, která umožňuje opravit jakoukoliv odchylku vnitřních etalonů, bez nutnosti mechanického zásahu. [1]

4.2 Přední panel odporové dekády řady M63x

V této kapitole je popsáno uspořádání předního panelu pro dekádu M630. Na obrázku 14 je přední panel imaginárně rozdělen do 4 sekcí, pod označením 1, 2, 3 a 4. Jejich význam je popsán dále v textu pod obrázkem.

Za zvýrazněným (tučným) textem dále v textu, je uvedena číslice, která odpovídá označení u dané sekce.



Obrázek 14 Přední panel dekády M630 [2]

Výstupní svorky - 1

Měřicí obvod lze připojit pomocí dvou, tří nebo čtyř vodičů. Jsou zde barevně odděleny dva póly, červený a šedý. Oba póly jsou plovoucí až do 500 V špičkového napětí proti zemnicí zdířce a skříni přístroje.

Zemnicí zdířka - 1

Zdířka je připojena ke kovovému tělu přístroje. Pokud není připojený měřicí přístroj uzemněn, je vhodné šedý výstupní terminál připojit právě k zemnicí zdířce.

Obrazovka - 2

Na obrazovce se mohou vyskytnout tři barvy písma - červená, černá a modrá. Červeně jsou označeny naměřené hodnoty. Modře jsou reprezentovány hodnoty, které lze měnit přímo na obrazovce nebo z hlavní nabídky. Černě jsou zobrazeny hodnoty, které nelze měnit.

Obrazovka je rozčleněna do několika oblastí. Horní modrá lišta informuje o jménu funkce a aktuálním čase. Pod touto lištou se nachází hlavní oblast, ve které je hlavní údaj, který je svou velikostí odlišen od vedlejších údajů. Vedlejší údaje jsou:

- Teplotní standard (PT385, PT3916,...)
- Přepínací mód (FAST, VIA OPEN,...)
- Odpor R0(pro funkce Platinum a Nickel)

Output (informuje o tom jaký odpor je připojen na výstupní svorky)

Další oblastí na obrazovce je specifikační oblast, která se nachází pod hlavní oblastí. Jsou zde uvedeny specifikace vztažené k aktuální hodnotě výstupu.

Poslední oblast je lišta čtyř štítků, které představují systémová tlačítka, jejichž funkce a názvy se liší od zvolené funkce.

Displejové klávesy - 2

Jejich funkce je vždy označena na displeji. Pokud u dané klávesy není zobrazena žádná funkce, pak je neaktivní.

Tlačítko OPER - 3

Tato klávesa umožňuje připojení nebo odpojení hodnoty na výstupní svorky. Zda jsou svorky aktivní, je signalizováno pomocí LED diody v tlačítku.

Tlačítko SHORT - 3

Stisknutím tohoto tlačítka dojde ke zkratu namísto připojení nastaveného odporu. I zkrat musí být aktivován pomocí tlačítka OPER.

Klávesy nahoru a dolů slouží ke krokování číslice. Kdežto klávesy doprava a doleva většinou slouží ke krokování nabídky a přesunům mezi úrovněmi, někdy jimi lze i nastavovat číselnou hodnotu. Prostřední klávesou pak lze vybrat hodnotu, kterou chceme nastavit [1].

Číselná klávesnice - 4

Klávesnice slouží pro zadávání číselných hodnot na displeji. Klávesou CANCEL se ruší zadaná hodnota a klávesou ENTER se zadaná hodnota potvrzuje.

4.3 Zadní panel dekády řady M63x



Obrázek 95 Zadní panel dekády M630 [2]

Z obrázku 15 je patrné, že se na zadním panelu nachází síťová přívodka s pojistkou a vypínačem. Dále je zde vidět komunikační rozhraní RS232, které je na všech dekádách M63x a volitelně je zde i LAN, USB a IEEE488 ve verzi M630A [1].

4.4 Zapnutí

Po zapnutí síťového vypínače dojde k rozsvícení displeje na čelním panelu. Poté dekáda provádí inicializaci vnitřních obvodů, která orientačně trvá 5s.

Po úspěšné inicializaci se dekáda automaticky nastaví do referenčního stavu, který je udán výrobcem a má tyto hodnoty:

Funkce	Resistance
Hodnota	1000.00 Ω
Výstupní svorky	OPEN

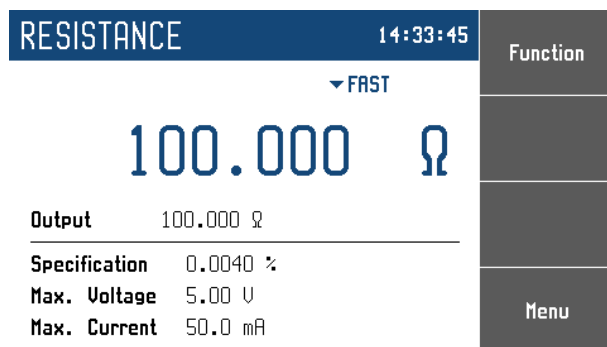
Dekáda je funkční po zapnutí a nastavení referenčního stavu, nicméně specifikovaných parametrů je dosaženo až po zahřátí přístroje po dobu 10 minut [1].

4.5 Volba funkce

Displejovou klávesou "Function" lze měnit funkce. Samotný výběr provedeme pomocí navigačních kláves nahoru a dolů, nebo displejovými klávesami a potvrzením pomocí tlačítka SELECT nebo stiskem klávesy OK.

4.5.1 Funkce RESISTANCE - ODPOR

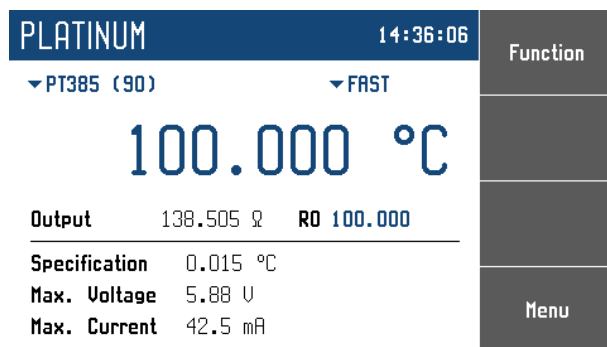
Ve funkci Odpor (viz obrázek 16) lze nastavit libovolnou hodnotu odporu v závislosti na typu dekády. Pro M630 je rozmezí odporu 16 Ω až 400 k Ω a pro verzi M630A je rozmezí odporu 1 Ω až 1,2 M Ω . Dále v této funkci lze nastavit přepínací módy: FAST, SMOOTH, VIA OPEN, VIA SHORT [1].



Obrázek 106 Obrazovka přístroje při zvolené funkci Odpor [2]

4.5.2 Funkce PLATINUM - PLATINA

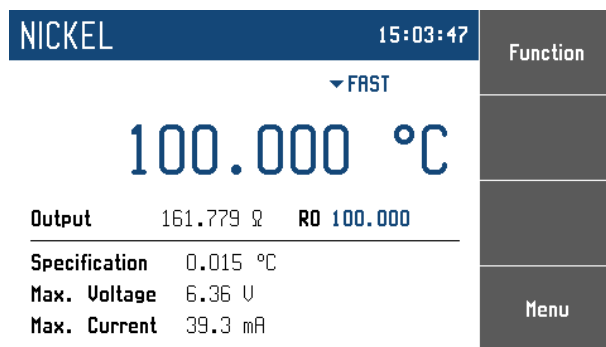
Tato funkce (viz obrázek 17) simuluje chování platinového teploměru zadáním přímé teploty. Lze zde nastavit teplotu v teplotním rozsahu od -200 °C až +850 °C. Hodnota odporu R0 se liší od verze dekády. Pro M630 je to 100 Ω až 1 k Ω pro M630A od 10 Ω do 20 k Ω . Pak lze volit teploměry: PT385(68), PT385(90), PT3916, PT3926, PT User. Nakonec lze volit i přepínací módy: FAST, SMOOTH, VIA OPEN, VIA SHORT [1].



Obrázek 117 Obrazovka přístroje při zvolené funkci Platina [2]

4.5.3 Funkce NICKEL - NIKL

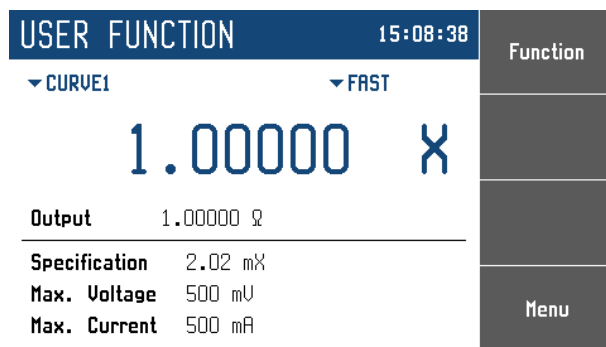
Dekáda v této funkci (viz obrázek 18) simuluje chování niklového teploměru. Lze nastavovat teplotu v teplotním rozsahu -60 °C až +300 °C, dále hodnotu odporu R0 viz Platina a přepínací módy: FAST, SMOOTH, VIA OPEN, VIA SHORT [1].



Obrázek 128 Obrazovka přístroje při zvolené funkci Nikl [2]

4.5.4 Funkce USER

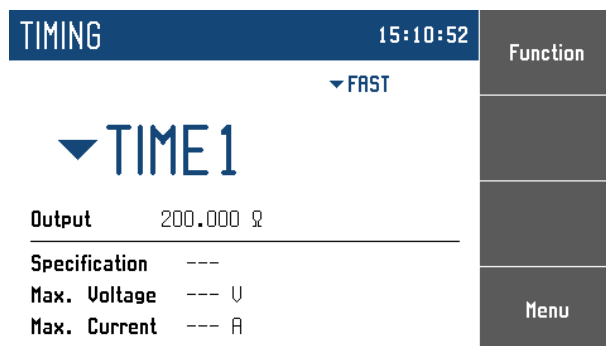
Dekáda simuluje chování konverzní křivky definované uživatelem (viz obrázek 19). Je možno definovat více křivek. Mezi zadanými body jsou hodnoty lineárně interpolovány. Lze zde nastavovat jednotku (definováno uživatelem), funkční průběh (definováno uživatelem) a přepínací módy: FAST, SMOOTH, VIA OPEN, VIA SHORT [1].



Obrázek 139 14Obrazovka přístroje při zvolené funkci User [2]

4.5.5 Funkce TIMING - SEKVENCE

Dekáda simuluje chování proměnného odporu podle časové tabulky (viz obrázek 20), která je definovaná uživatelem. Lze definovat více sekvencí. Je možno nastavit sekvenční tabulku a přepínací módy.



Obrázek 20 15Obrazovka přístroje při zvolené funkci Sekvence [2]

4.6 Vysvětlení přepínacích módů v závislosti na změně odporu.

- T je zde konstanta, která udává čas mezi přepnutím z odporu R1 na R2
- FAST: Nejrychlejší přepínací mód. T je udáváno jako 400 μ s. Odpor během přechodu není definován.
- SMOOTH: Jedná se o přepínání v "hladkém" průběhu. Změna odporu je minimální. T je stanoveno na 1 ms. Během času T se odpor nedostane z intervalu R1,R2.
- VIA OPEN: Dvoukrokový přepínací mód. R1 se přepne na OPEN a potom na R2. T je udáváno jako 1 ms.
- VIA SHORT: R1 se přepne na SHORT a potom na R2. T je udáváno jako 1 ms.

4.7 Vlastnosti odporových dekád

Krátké shrnutí vlastností odporových dekád udávané výrobcem je v následující tabulce 1.

Tabulka 1 Parametry odporových dekád řady M6xx

Typ/model	M641	M642	M631	M632
Rozsah odporu	10 Ω -300k Ω	0,1 Ω -20M Ω	16 Ω -400k Ω	1 Ω -1,2M Ω
Přesnost odporu	0,02%	0,02%	0,003%	0,003%
Max. napětí	200V	200V	200V	200V
Simulate RTD	Pt, Ni	Pt, Ni	Pt, Ni	Pt,Ni
Přesnost simulace	0,1°C	0,1°C	0,01°C	0,01°C
Zatížitelnost	5W	5W	-	-

4.8 Komunikační rozhraní

Dekáda může komunikovat v základní konfiguraci pomocí rozhraní RS232 a v případě rozšířené verze pak i přes LAN, USB, a GPIB. Komunikovat lze výhradně vždy jen pomocí jednoho druhu rozhraní. Pro aktivaci komunikace je nutno nastavit v systémovém menu příslušné rozhraní. Všechna rozhraní sdílejí stejnou sadu příkazů. Zde je uvedena výjimky příkazů, které lze použít pouze pro rozhraní RS232, LAN a USB [1].

- SYSTem:LOCal nastaví dekádu do režimu ručního ovládání
- SYSTem:REMOte nastaví dekádu do vzdáleného přístupu
- SYSTem:RWLock nastaví dekádu do režimu vzdáleného přístupu a deaktivuje klávesnici na čelním panelu

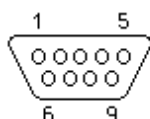
4.8.1 Ovládání dekády pomocí RS232

Pro ovládání odporové dekády je zapotřebí samotná dekáda, počítač se sběrnici RS232 (lze využít i USB redukci) a devíti-vývodový D-SUB male/female RS232 kabel.

Jak je uvedeno výše, je zapotřebí před samotnou komunikací v nastavení zvolit typ komunikace v tomto případě na RS232. Volitelný parametr je zde pouze rychlost komunikace Boudrate 1200,2400,4800,6900,19200,38400,57600 a 115200

Zbylé parametry jsou pevně zadány a nelze je měnit.

- Počet datových bitů 8
- Počet stop bitů 1
- Parita žádná
- Handshake(Xon/Xoff) Vypnuto



Obrázek 21 Konektor D-SUB s devíti vývody pro komunikaci pomocí RS232 [2]

Popis vývodů konektoru je v následující tabulce 2.

Tabulka 2 Popis vývodů konektoru pro sériové rozhraní RS232

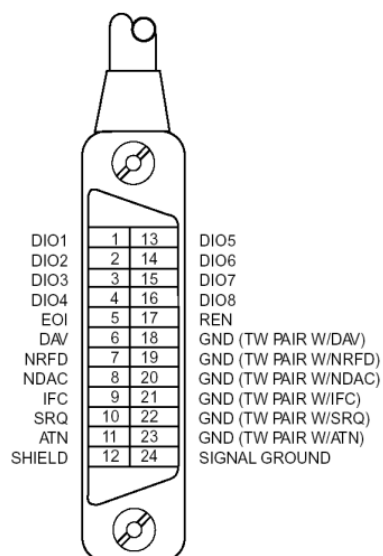
Vývod	Název	Směr	Popis
2	TXD	Výstup	Vysílač
3	RXD	Vstup	Přijímač
5	GND	-	Uzemnění

4.8.2 Ovládání pomocí sběrnice GPIB

Potřebné vybavení pro komunikaci přes GPIB: Samotná dekáda (s rozšířením GPIB), počítač se sběrnici GPIB a kabel GPIB.

Stejně jako u komunikace pomocí sběrnice RS232 je zapotřebí nastavit v systému typ komunikace na GPIB. Jediný parametr, který lze nastavit, je adresa zařízení.

GPIB Address 1 až 31



Obrázek 22 16Rozložení vývodů konektoru rozhraní GPIB [1]

Dekáda rozeznává následující příkazy:

- DCL DeviceClear nastavuje přístroj do základního stavu
- SDC SelectedDeviceClear nastavuje přístroj do základního stavu
- GTL Go To Local ukončuje režim vzdáleného přístupu, přístroj nelze z klávesnice ovládat

Je tedy patrné, že komunikace bude probíhat pomocí klíčových slov parametrů, které budou odesílány do odporové dekády, a bude záležet na zpětné vazbě od odporové dekády, jaká další klíčová slova budou odeslána na zařízení. Pro firmu Meatest je vhodnější komunikace přes RS232.

Ukázka příkazů:

```
:CALibration
:RESistance
:AMPLitude(?) <DNPD>
:SElect(?) <DNPD>
:SECure
:PASSword(?) <DNPD>
:EXIT
:DISPlay
:ANNotation
:CLOCK
:DATE
:FORMat(?) {MDYS|MDYA|DMYS|DMYO|DMYA|YMDS|YMDO}
[:STATe](?) {ON|OFF|1|0}
:BRIGhtness(?) <DNPD>
:LANGuage(?) {ENGLish}
:OUTPut
```

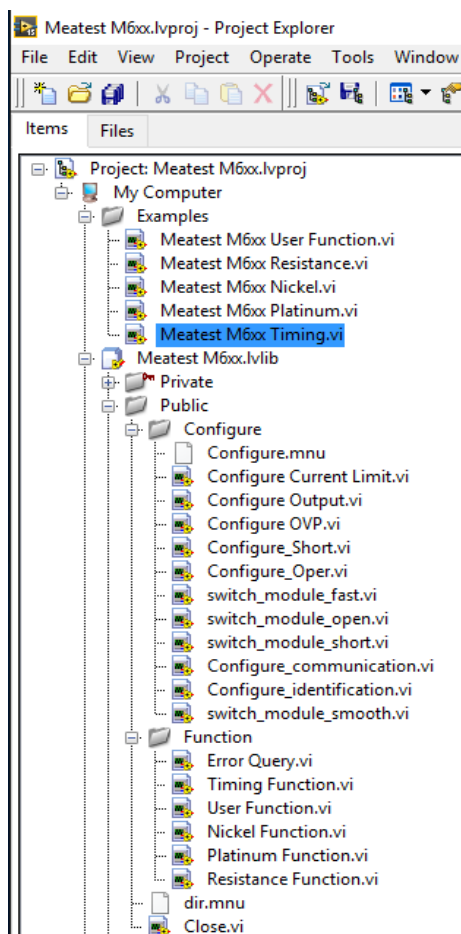
4.9 Rozdíl mezi řadou M63x aM64x

Zásadní rozdíl mezi řadou M63x aM64x je v jejich předpokládaném použití. Řada M63x by se dle výrobce měla využívat pro výzkum, je to především dáno její přesností nastavené hodnoty, která je u typu M632 0,003%. U řady M64x se spíše očekává, že bude využita pro laboratorní účely. Pro typ M642, což je odporová dekáda stejných základních parametrů jako typ M632, je udávaná přesnost 0,02%.

5 Tvorba ovladače pro dekády řady M63x a M64x

Projekt pro odporové dekády byl pojmenován Meatest M6xx, jméno bylo vhodně zvoleno tak, aby bylo jasné, že daný driver obsahuje všechny varianty odporových dekád řady M63x a M64x.

Struktura přístrojového ovladače pro odporové dekády M6xx vytvořeného v prostředí LabVIEW je na následujícím obrázku (viz obrázek 31).

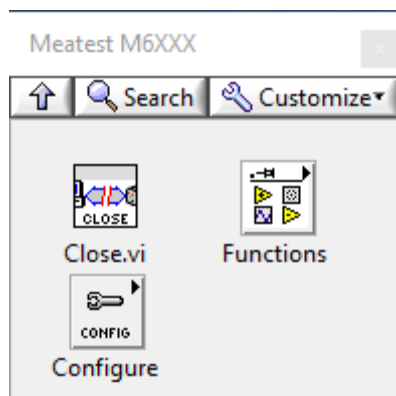


Obrázek 17 Struktura driveru v prostředí LabVIEW

Pro nový ovladač by v záložce *Examples*, *Public* byly ukázkové příklady vytvořeny firmou NI. Je nutné je přetvořit nebo nejlépe smazat, protože s tímto zařízením nemají nic společného, to platí o všech jiných VI, které se v projektu nacházejí. Po vyčištění je nutné vytvořit vlastní ovládací nástroje. Jak je vidět na obrázku 31, tak ve složce *Configure* se nacházejí nástroje pro úpravu přepínacích módů, nastavení výstupních svorek, nastavení komunikace a identifikace. Složka *Function* obsahuje jednotlivé funkce odporových dekád. Samotné bloky ve složce *Public* nejsou funkční, je zapotřebí je vhodně pospojovat a k tomu slouží vzorové příklady.

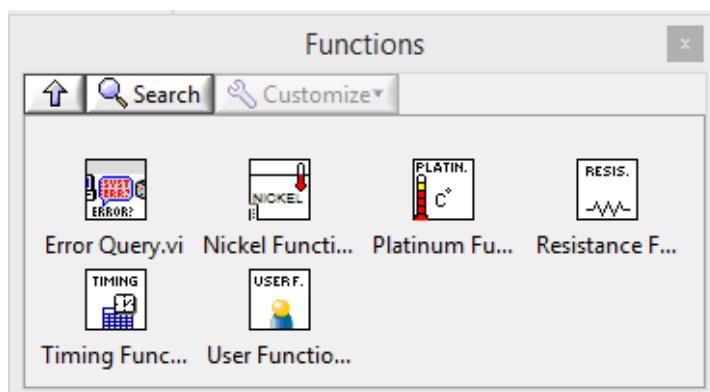
Ve složce *Examples* se nacházejí funkční ukázky pro jednotlivé funkce odporových dekád. Jedná se o volbu odpor (funkce RESISTANCE), volbu nikl (funkce NICKEL), volbu platina (funkce PLATINUM), volbu časování (funkce TIMING) a volbu funkce USER FUNCTION.

Daný ovladač obsahuje vlastní paletu (viz obrázek 32), která obsahuje nástroje pro ovládání instrumentu. Paleta lze různě editovat, může se zde přidat subpaleta, v tomto případě má název *Functions*, lze nastavit její synchronizaci se složkou. Po nastavení synchronizace se složkou dojde k zviditelnění všech VI, které se nacházejí v dané složce.



Obrázek 18 Paleta Meatest driveru M6xxx

Pod subpaletou *Functions*, která je vidět na obrázku 32, se skrývá 5 nástrojů pro ovládání příslušných funkcí přístroje (viz obrázek 33).

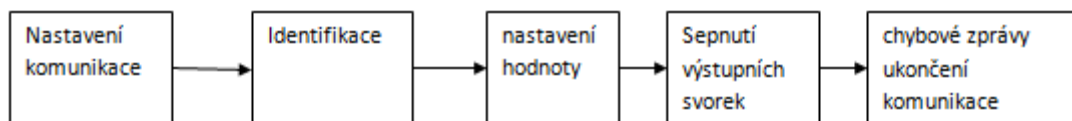


Obrázek 33 Subpaleta Functions

Subpaleta *Configure* obsahuje nástroje pro nastavení výstupních svorek a přepínacích módů.

Nejdůležitější na celém driveru jsou příklady použití, protože ukazují, jak použít jednotlivá vytvořená VI. Požadavkem firmy Meatest bylo vytvoření pěti příkladů pro ukázku funkčnosti jednotlivých funkcí odporových dekád.

Obecný blokový diagram, který popisuje strukturu vzorových příkladů pro odporovou dekádu, je na obrázku 23.



Obrázek 2319 Bloková struktura jednotlivých funkcí vytvořeného ovladače

Jednotlivé části diagramu jsou vysvětleny níže v textu a jsou zvýrazněny pro jejich snadné vyhledání. Podle pokynů firmy Meatest, by měl budoucí driver fungovat tak, že dojde k nastavení navolených funkcí a hodnot a nakonec k zpřístupnění lokálního ovládání.

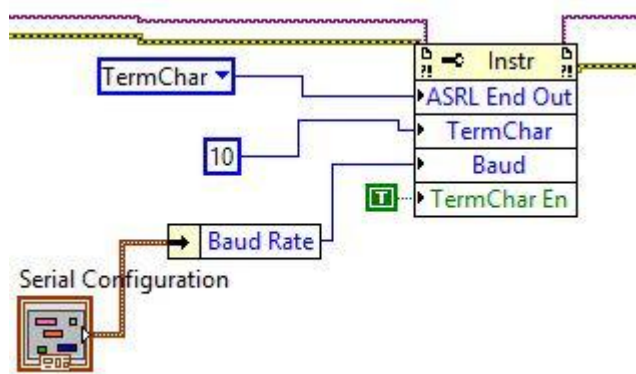
Ukázka použitých příkazů k ovládání přístroje:

- `SYSTem:REMOut<LF>` Povolení vzdáleného přístupu
- `*IDN?<LF>` Vyžádání identifikace Odpověď:
MEATEST,M630 atd.
- `SYSTem:LOCal<LF>` Povolení lokálního přístupu
- `PLAT <číslo>` Volba funkce platina <číslo> udává hodnotu teploty
- `PLAT:ZRES <číslo>` Volba odporu R0 pro platina <číslo> udává hodnotu odporu
- `NICK <číslo>` Volba funkce nikl <číslo> udává hodnotu teploty
- `NICK:ZRES <číslo>` Volba odporu R0 pro nikl <číslo> udává hodnotu odporu

První funkcí je Nastavení komunikace, zde dochází k nastavení volitelných parametrů pro komunikaci driveru s dekádou. Pro sériovou komunikaci je možné nastavit pouze rychlost přenosu. Pro tvorbu driveru byla rychlost nastavena na 9600 Bd. Testovací dekáda M630 podporuje rychlost komunikace od 1200 Bd do 115200 Bd.

Vstupem této funkce je VISA reference, chybová struktura a číselná hodnota *boud rate*. Výstupem je VISA reference a chybová struktura.

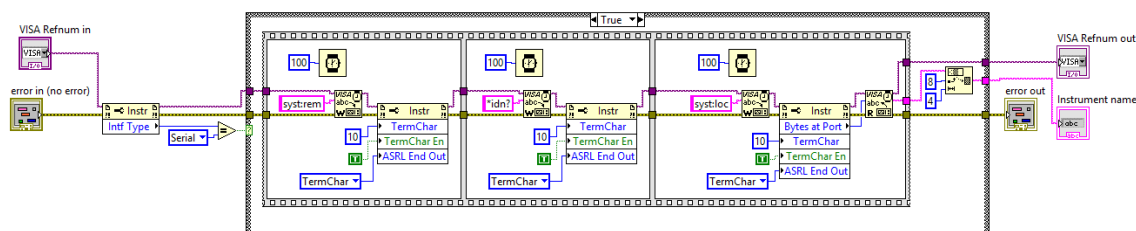
Nastavení komunikace (viz obrázek 24) v LabVIEW vypadá následovně. Místo funkce *VISA ConfigureSerial Port* byl použit *Property Node*, kde ze záložky *Serial Settings* byla vybrána rychlost komunikace.



Obrázek 2420 Nastavení komunikace

Pomocí *Property Node* nástroje je zde nastavena rychlost *baud rate*, která musí být shodná s nastavenou rychlostí na přístroji. Dále je zde nastaven ukončovací znak, který pro komunikaci po sériové lince musí být takto explicitně zadán.

Následující funkcí je Identifikace. V této funkci se provádí klasické rozpoznání přístroje standardizovaným příkazem *IDN?* Blokové schéma vypadá následovně (viz obrázek 25).

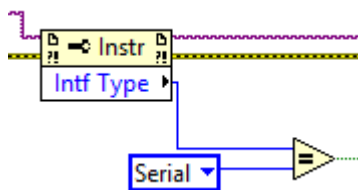


Obrázek 25 Funkce IDENTIFIKACE

Funkce *VISA Refnums In* nese informaci o typu komunikace, v tomto případě sériová linka. Konstanta označená zelenou barvou je typu *BOOL* a je nastavena na hodnotu *true*. Celý cyklus je uzavřen do smyčky *case*, která v sobě obsahuje strukturu *flat sequence*. V prvním okně *flat* struktury je posílán na dekádu příkaz pro vzdálený přístup: *SYST:REM*. Mezi každým příkazem je vloženo zpoždění, důvodem je plynulost komunikace. Dále je příkazem *IDN?* zjišťováno, jaký přístroj je připojen. Dále je uzavírán vzdálený přístup a je čtena zpětná vazba. Celý tento cyklus se provede pouze jednou.

Pokud by komunikace byla řešena pomocí GPIB, lze postupovat velice podobně jako u sériové linky. S rozdílem, že konkrétně pro tyto odporové dekády není potřeba uvádět příkazy pro povolení vzdáleného a lokálního přístupu. Nejprve je zapotřebí rozlišit, zda komunikace probíhá pomocí RS232 nebo GPIB k tomu slouží opět funkce *Property Node*, ve které je nastavena *Intf Type* (viz obrázek 33), jedná se o číselnou proměnnou, pro kterou platí, že pokud je komunikace vedená přes RS232, má hodnotu 4. Proto je využito jednoduché

porovnačí funkce a je tak zjištěno, zda komunikace je realizována po sériové lince nebo po jiném rozhraní.

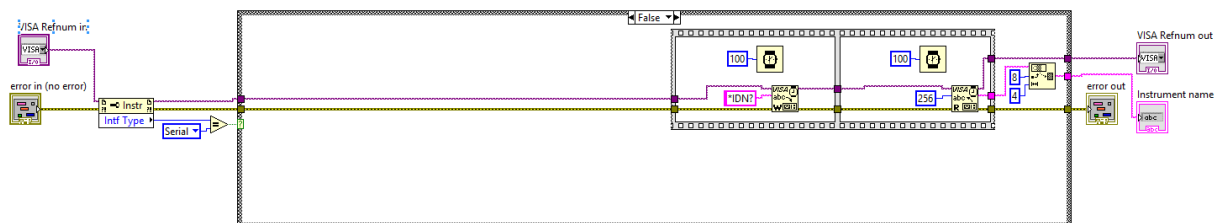


Obrázek 26 Property Node s rozpoznáváním komunikace

Logická hodnota, která je výstupem z porovnávací funkce, je převedena na podmínkový vstup struktury *case*.

Vstupem pro funkci Identifikace je VISA reference a chybová struktura. Výstupem je VISA reference, chybová struktura a *string* řetězec, nesoucí název přístroje.

Pro GPIB komunikaci vypadá blokový diagram funkce Identifikace podobně (viz obrázek 27).

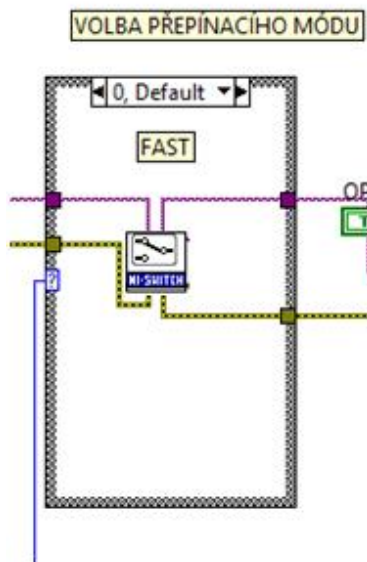


Obrázek 27 GPIB varianta funkce Identifikace

Rozdílem je, že pro GPIB komunikaci nepoužíváme příkazy pro vzdálené ovládání, jako jsou *SYST:REM* a *SYST:LOC*.

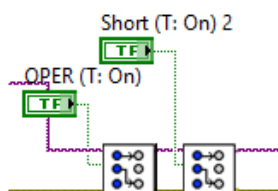
Další funkcí je Nastavení hodnoty (viz obrázek 28). Pro jednotlivé následně vytvořené příklady je tato funkce pochopitelně jiná. Ve funkci *Odpor* je nastaveno jediná hodnota, a to hodnota odporu. Ve funkci *Nikl* je nastavena hodnota teploty a hodnota odporu při teplotě 0 °C. U funkce *Platina* lze stejně jako u funkce *Nikl* nastavit hodnotu teploty a hodnotu odporu při 0 °C. U funkce *Timing* je odeslána číselná hodnota reprezentující index tabulky. Ve funkci *User Function* je odeslána hlavní hodnota a číselný index pro výběr aktivní konverzní křivky.

Po nastavení příslušných hodnot dojde, pokud je to vyžadováno, k přepnutí přepínacího módu (viz obrázek 28). Čtyř-úrovňový case, v každé úrovni se nachází jiná funkce pro daný přepínací mód. Vstupem a výstupem jsou VISA reference a chybová vlákna.



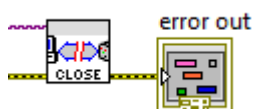
Obrázek 28 Volba přepínacího módu

Další funkcí v blokovém diagramu je Sepnutí výstupních svorek (viz obrázek 29). Dochází zde k připojení výstupních svorek popřípadě jejich zkratování. To je prováděno dvěma funkcemi, které jsou ovládány pomocí *BOOL* logiky. Pokud je přepínač, který se nachází na ovládacím rozhraní na pozici *off (false)* pro výstupní svorky, tak se výstupní svorky nepřipojí. To samé platí pro tlačítko *SHORT*.



Obrázek 29 Funkce OPER a SHORT

Nakonec může dojít k uzavření komunikace. K tomu je využívána funkce *Close* (viz obrázek 30), která je umístěna na paletě *Instrument Driver*. Společně s ukončením komunikace dochází k výpisu všech nashromážděných chybových výpisů, které byly zachyceny.



Obrázek 30 Ukončení komunikace – funkce Close

5.1 Tvorba vzorových příkladů použití funkcí pro ovládání dekády

V této kapitole je popsáno pět příkladů použití každé programovatelné funkce. Pro snadnější použití těchto příkladů jsou přednastavené některé volitelné možnosti (například u funkce Odpor hodnota odporu). Uživatel si tak může snadno a rychle odzkoušet funkčnost daného příkladu použití.

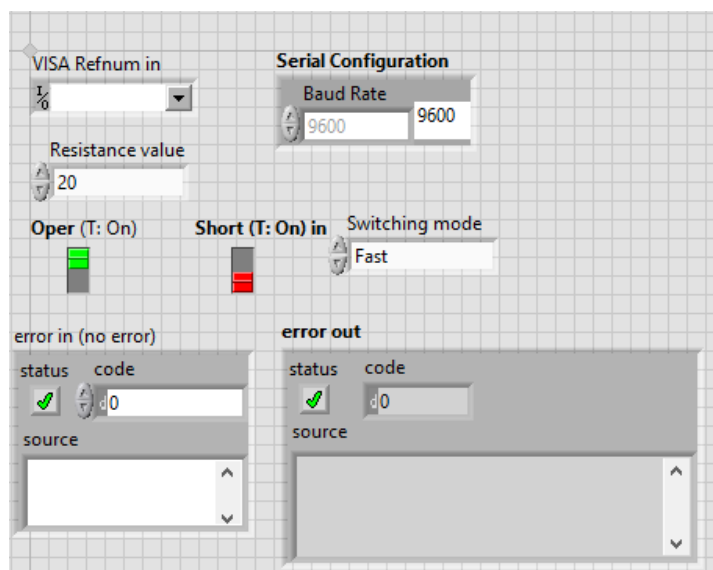
5.1.1 Pravidla pro tvorbu příkladů použití

Při tvorbě vzorových příkladů bylo dbáno na následující pravidla:

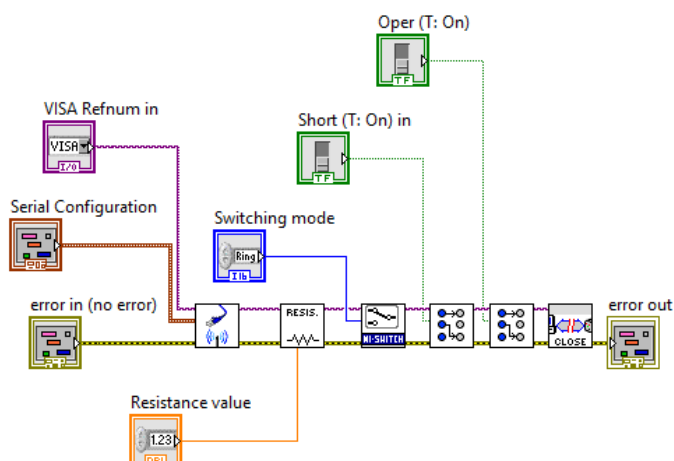
- Pokud byla vyžadována číselná hodnota, byl použit blok *numeric konstant*. Pokud to bylo možné, ve vlastnostech tohoto bloku byly nastaveny limitní hodnoty pro dolní a horní mez. Jedná se o ochranu zadávané hodnoty. Uživatel tak nemůže zadat zápornou hodnotu odporu a podobně. Obdobně to platí i pro *string* řetězec.
- Pokud bylo nutné přepínat mezi vícero možnostmi, byl využit blok *numeric ring*. Jedná se o funkci, ve které je možno nastavit libovolné množství přepínatelných výsledků. Jednotlivé výsledky mohou mít volitelné jméno a volitelný číselný index. *Numeric ring* je využit například ve funkci Odpor jako *Switching mode*, kde v uzavřeném kruhu přepínáme mezi možnostmi (FAST, SMOOTH, SHORT a OPEN)
- Pro zjednodušení rozhodovací logiky byly použity funkce, které již v sobě obsahují rozhodovací člen *if*, namísto použití nejprve samostatného členu *if* a posléze odeslání výsledku *true* nebo *false*.
- Dalším zpřehledněním zdrojového kódu bylo vytvoření podfunkcí (subVI's), namísto celistvého programu v jednom VI souboru. Vytvořily se tak uživatelem volitelné funkce jako je například Identifikace, Platinum function a další.

5.1.2 Funkce Odpor

Ovládací panel vypadá následovně (viz obrázek 34).



Obrázek 21 Ovládací panel funkce Odpor

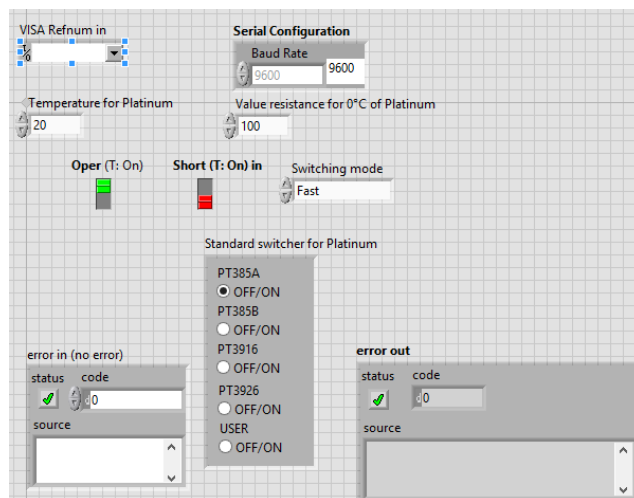


Obrázek 22 Blokový diagram funkce Odpor

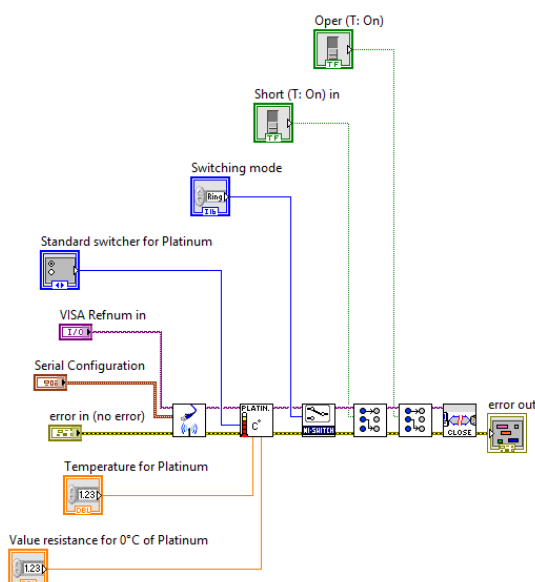
Z obrázku 35 je vidět logická posloupnost příkazů. Nejprve dojde k nastavení komunikace poté, k odeslání zadané hodnoty odporu uživatelem, následně, pokud je *bool* operand Oper v logické jedničce, dojde k sepnutí výstupních obvodů na dekádě. V poslední řadě dojde k přečtení chybových zpráv a uzavření komunikace.

5.1.3 Funkce Platina

Pro funkci Platina vypadá ovládací panel následovně (viz obrázek 36).



Obrázek 36 Ovládací panel funkce Platina

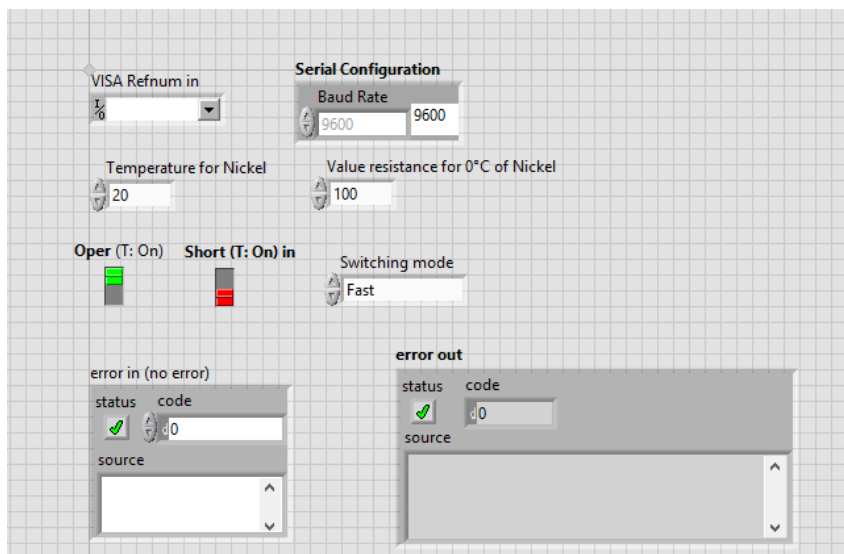


Obrázek 23 Blokový diagram funkce Platina

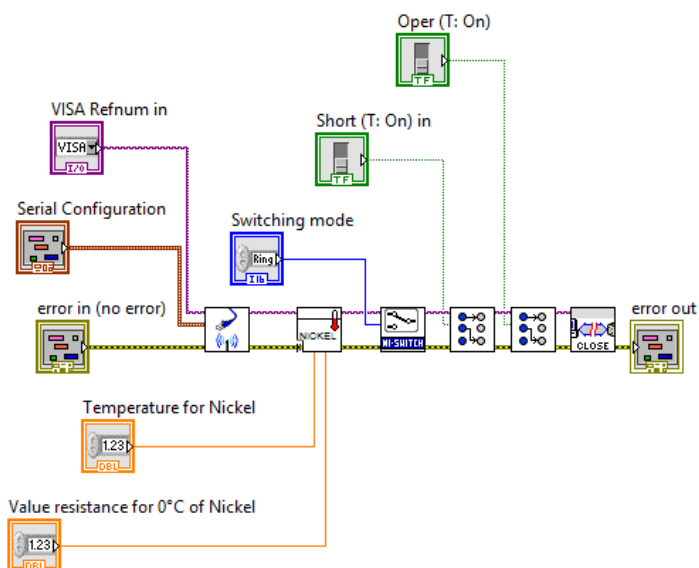
Z blokového diagramu (viz obrázek 37) je patrné, že jako první se provede opět nastavení komunikace, vstupem dalšího bloku je dvakrát hodnota *double* (*Temperature for Platinum*, *Value resistance for 0°C of Platinum*) a jednou hodnota *unsigned long* (*Standard switcher for Platinum*). Nejdříve dojde k nastavení hodnoty teploty, poté k nastavení hodnoty odporu R0 (odpor při 0 °C) a nakonec dojde k výběru jednoho ze 4 standardů pro platinové teploměry. Poté na základě hodnoty *bool* pro *Oper* a *Short* dojde buď k jejich zapnutí nebo vypnutí. Na konci dojde k vypsání chybových zpráv a ukončení komunikace.

5.1.4 Funkce Nikl

Pro funkci Nikl vypadá ovládací panel následovně (viz obrázek 38).



Obrázek 24 Ovládací panel funkce Nikl

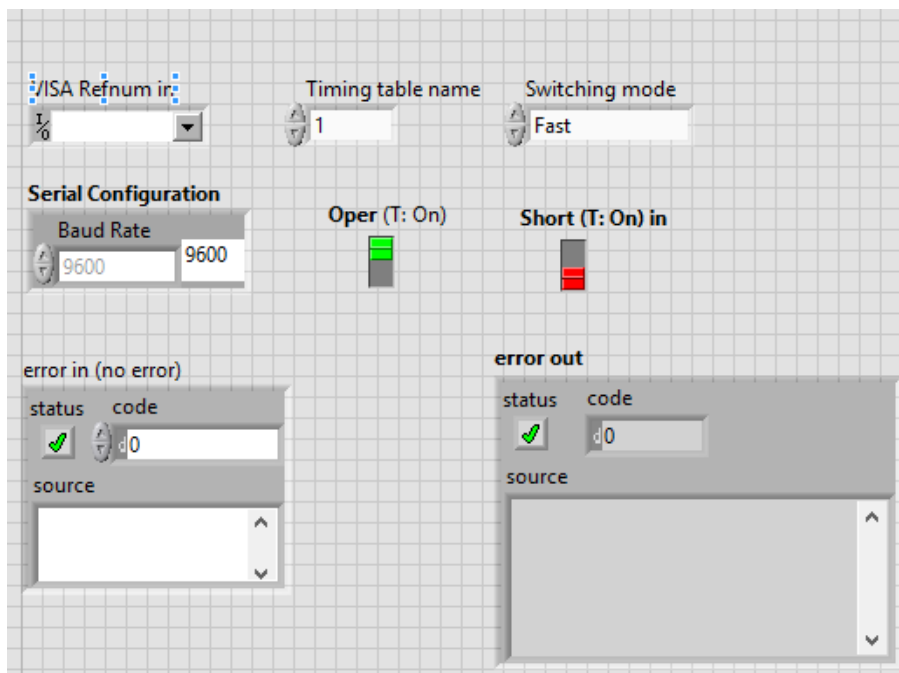


Obrázek 25 Blokový diagram funkce Nikl

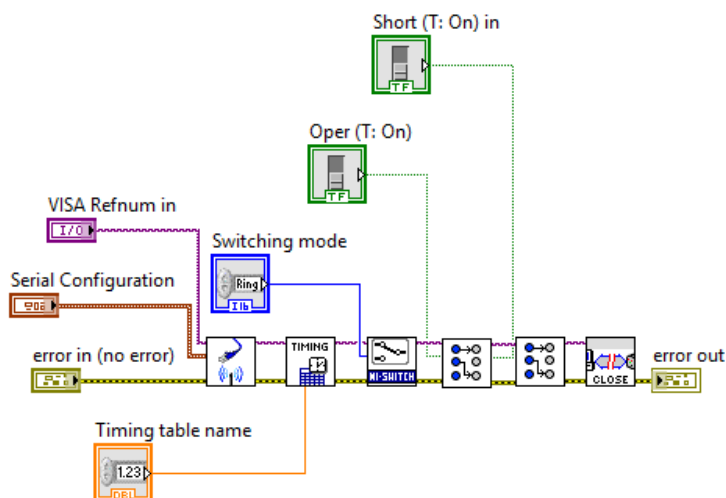
Stejně jako v předchozích případech je na obrázku 39 první funkcí blok pro nastavení komunikace, druhým krokem je nastavení hodnoty teploty pro Nikl a dále nastavení hodnoty odporu při 0 °C. Třetím a čtvrtým krokem je nastavení výstupu a nakonec dojde k přečtení chyb a ukončení komunikace.

5.1.5 Funkce *Timing*

Ovládací panel této funkce je následující (viz obrázek 40).



Obrázek 26 Ovládací panel funkce *Timing*

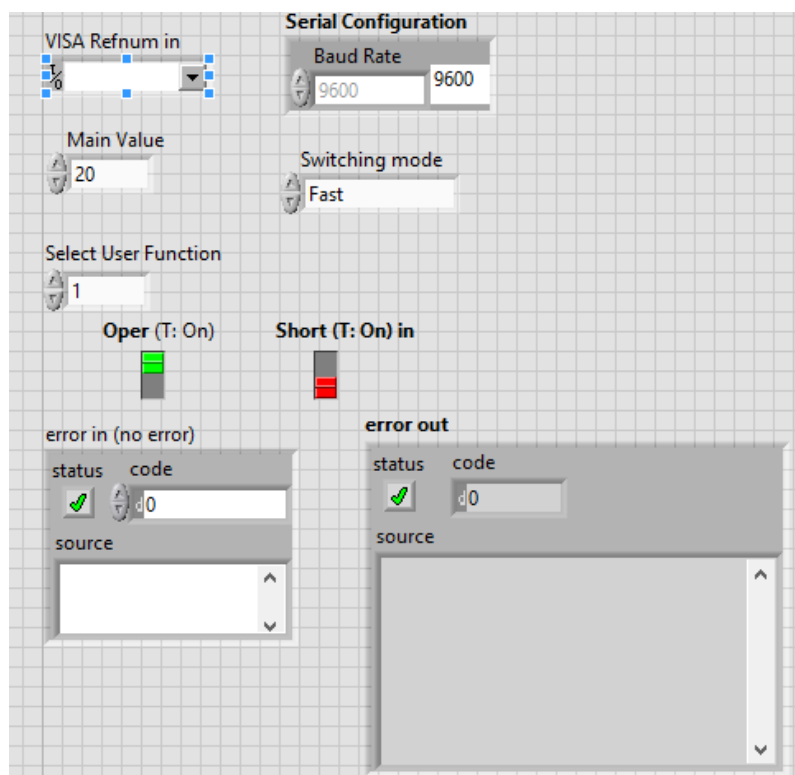


Obrázek 41 Blokový diagram funkce *Timing*

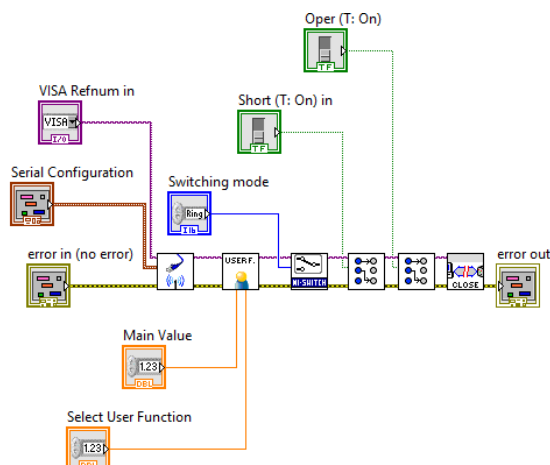
Ve funkci *Timing* (viz obrázek 41) je vidět, že nejprve opět proběhne nastavení komunikace, poté následuje funkce, kde dochází k odeslání číselného indexu, který reprezentuje pořadí časové tabulky v dekádě. Poté dochází k přepnutí přepínacího módu na základě výběru uživatelem a nakonec dojde k přečtení chybových zpráv a ukončení komunikace.

5.1.6 Funkce User Function

Posledním příkladem je *User Function*. Jeho ovládací panel vypadá takto (viz obrázek 42).



Obrázek 42 Ovládací panel User Function



Obrázek 27 Blokový diagram funkce User Function

Na obrázku 43 je vidět, že nejprve dochází k nastavení komunikace, poté k zadání hlavní hodnoty a vybrání příslušné aktivní konverzní křivky. Poté, na základě uživatelem zvolené možnosti, dojde k nastavení přepínacího módu, výpisu chybových zpráv a ukončením komunikace.

6 Ověření funkčnosti vytvořeného ovladače

Kapitola popisuje testování časové náročnosti ovladače. Dále je zde uvedena možnost využití virtuálního instrumentu. Jako poslední je zde rozebráno testování funkčnosti ovladače.

6.1 Časová náročnost driveru

Cílem testování časové náročnosti ovladače bylo zjistit, jak rychle dokáže reagovat odporová dekáda na přicházející příkazy.

Jednou z dobrých vlastností jednotlivých vzorových ukázek je rychlost obsluhy. Pro tento driver bylo nejdříve zjištěno, jak zadavatel, firma Meatest, provádí testování rychlosti v jiných komunikačních programech, doba prostojů mezi instrukcemi je volena okolo doby 100 ms. Pro tuto práci bylo však také testováno, jak rychle dokáže dekáda reagovat a jestli se její reakce liší od jiného typu dekád. Bylo zjištěno, že volba řady dekády M64x nebo M63x nehraje roli. Dekáda byla testována se zpožděním od 10 ms do 100 ms, po krocích 10 ms 50 ms a 100 ms. Během těchto testovacích intervalů dekáda vykonávala obslužný program bez chyb. U testování zpoždění 1 ms již dekáda nedokáže rychle reagovat a v průběhu provádění obslužného cyklu se zastaví.

6.2 Virtuální instrument

Pokud by došlo k situaci, kdy by byl testovaný přístroj nedostupný, je stále možné pokračovat ve vývoji ovladače nahrazením chybějícího přístroje virtuálním přístrojem. Idea je taková, že celý blok funkce, ve které je funkce pro čtení odpovědi z přístroje, by byl vložen do struktury *case* a rozhodovací logikou by byl přepínač na ovládacím panelu. Poloha přepnutí by rozhodovala o tom, zda bude spuštěn virtuální přístroj či ne. Pokud by byl virtuální přístroj v logické jedničce, tak by funkce v struktuře *case* vypisovala text, který by měl vypsát skutečný přístroj. Program by se tak neukončil a pokračoval by dál na další funkci.

V mém případě je toto řešení nepoužitelné, protože ve všech funkcích většinou nastavuji hodnoty příslušných parametrů dané funkce odporové dekády. To bohužel simulovat nejde.

6.3 Error Handling

Jedná se o jeden z nejdůležitějších aspektů přístrojového ovladače. Error Handling neboli podržení chybových zpráv. Podržení chybových zpráv je důležitým prvkem, protože od začátku jakéhokoliv příkladu použití nebo uživatelem volané funkce (viz obrázek 25 Identifikace po sériové lince RS232) se může stát, že přístroj zahlásí chybu, která je nesena na chybovém vlákne spolu s každou jinou chybou, která může nastat během prováděných instrukcí přístrojového ovladače. Nashromážděné chyby jsou poté vypsány v bloku *Close*, kde je již přednastavená funkce *Error Query*, ta vypisuje chybové hlášení (chybu po chybě na každý řádek jednu). Počet chyb je omezen zásobníkem chyb na každém instrumentu.

Během testování bylo zjištěno, že pro funkci *Error Query* není zapotřebí přístroj uvádět do stavu vzdáleného přístupu *remote* pro komunikaci po sériové lince.

6.4 Funkční testování driveru

V této kapitole je popsáno, jak bylo postupováno při testování driveru pro odporovou dekádu M63x a M64x.

Chování aplikace:

- spuštění LabVIEW a vybrání příslušného driveru (otevření hlavního panelu)
- navolení potřebné hodnoty a funkce
- spuštění aplikace
- vykonají příslušné operace
- ukončení vzdáleného přístupu (přepnutí na lokální přístup)

6.4.1 Spuštění LabVIEW

Tento krok je prováděn uživatelem na začátku celé procedury. V průmyslovém prostředí (z pohledu firmy, která si zakoupí odporovou dekádu řady M63x nebo M64x) si uživatel dohledá v hlavní záložce Instruments ->Find New Instruments firmu Meatest, následně si vybere příslušný driver a provede instalaci. Poté je již možné driver otevřít. Po spuštění se automaticky otevře přední panel vybraného vzorového příkladu.

6.4.2 Navolení potřebné hodnoty a funkce

Provede rovněž uživatel na základě viditelných nástrojů (vertikální posunovač, před vyplněná zadávací pole pro hodnoty odporů a teplot, blok pro výběr teplotního standardu) na hlavním panelu. Testování tohoto kroku bylo prováděno na pracovišti firmy Meatest. Během testování se přišlo na to, že po spuštění příslušného driveru je vhodné, aby všechny hodnoty a funkce byly již přednastavené na „tovární“ hodnoty.

6.4.3 Spuštění aplikace

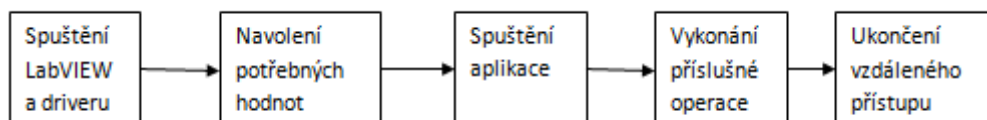
Buď provede uživatel, nebo obslužný program. Uživatel stiskne tlačítko run ve vývojovém prostředí LabVIEW, konkrétně v okně ovládacího panelu programu.

6.4.4 Vykonání příslušné operace

Již provádí driver automaticky na základě odesílaných příkazů. Během testování tohoto kroku bylo zapotřebí zvolit vhodné zpoždění mezi vykonáním daných funkcí. Důvod je takový, že dekáda nedokáže nastavovat funkce a hodnoty a přijímat data z obslužného programu v minimálním čase.

6.4.5 Ukončení vzdáleného přístupu

Provádí driver automaticky odesláním příslušného příkazu. Grafické znázornění příslušných operací (viz obrázek 44).



Obrázek 44 Vývojový diagram testování přístrojového ovladače

Testování bylo zpočátku prováděno pomocí zpětné vazby RS232 kabelu (Loopback smyčka), který byl zapůjčen od firmy Meatest. Jednoduchým odesláním znaku a následným přijetím bylo otestováno správné nastavení sériového komunikačního rozhraní. Po zprovoznění komunikace po sériové lince byla testována komunikace na dekádě. Zde bylo nutné nastavit správný čas pro dobu čekání před čtením zpětné vazby.

Po testování první funkce Identifikace bylo na řadě testování volby funkce pro daný typ dekády. Nejprve byl použit jednoduchý více polohový přepínač, který ukazoval na jednu funkci, která měla své identifikační údaje, např. pro odpor byl identifikační údaj RES, který se shodoval s příkazem pro funkci odpor RES z technické dokumentace přístroje. To se již odehrávalo v prostorách firmy a program byl testován přímo na dekádě. Zprvu byla vize firmy Meatest v tvorbě jednoho VI souboru pro každou dekádu, který bude schopen ovládat všechny funkce na dekádě. Během testování a vývoje těchto VI souborů bylo zjištěno, že VI soubor je na první pohled moc členitý, proto se přešlo na vývoj jednotlivých malých a jednoduchých příkladů použití (examplů), které obsluhují jednu funkci na libovolné dekádě.

7 Závěr

Mým cílem bylo nastudovat funkce pro programovatelné dekády řady M63x a M64x a zjistit možnosti jejich vzdálené komunikace. Ve své práci jsem se věnoval pouze komunikaci prostřednictvím RS232 a standardu GPIB, tyto druhy rozhraní jsou nejběžnější a proto klíčové pro komunikaci. Dále jsem měl vysvětlit tvorbu driverů pro tyto instrumenty a navrhnout testování vytvořeného driveru.

Popis, jak se ovladač tvoří, jsem vyčetl na czech.ni.com, kde jsem také nastudoval obecný popis a hierarchii ovladačů. Vytvořil jsem funkční driver s jednotlivými funkcemi, které jsou využity v příkladech použití (Exampley) pro daný driver. Příklady použití jsem naprogramoval pro jednotlivé funkce dekád. Dále jsem testoval časovou náročnost dekád, tedy rychlostí s jakou lze odporové dekády obsluhovat. Výsledky testování prokázaly, že jsem zhotovil funkční driver. Naprogramovaný driver pro firmu Meatest je vystaven na stránkách firmy NI.

Seznam zkratek

GPIB	General Purpose Interface Bus
USB	Universal Serial Bus
ID	Instrument Driver
VISA	Virtual Instrument Software Architecture
API	Application Programming Interface
NI	National Instruments
TFT	Thin Film Transistor

Literatura

[1]MEATEST: Programovatelná odporová dekáda. *M602 manual - m630n*. 2013, (1).[cit. 2016-03-01] Dostupné z: <http://www.meatest.cz/produkty-m630-programovatelná-odporová-dekáda-detail-214>

[2]MEATEST: Programovatelná odporová dekáda *M642 manual - m642*. 2013 (1). [cit. 2016-03-01] Dostupné z: <http://www.meatest.cz/files/download/man/m642m.pdf>.

[3]MEATEST: Programovatelná odporová dekáda. *M632 manual - m632n*. 2013, (1).[cit. 2016-03-01] Dostupné z:<http://www.meatest.cz/files/download/man/m632n.pdf>

[4]INSTRUMENT DRIVER:Www.czech.ni.coms. Instrumetn Driver [online]. 2015 [cit. 2016-04-01]. Dostupné z: <http://www.czech.ni.com/>

[5]VXI Plug & Play [online]. [cit. 2016-04-19]. Dostupné z: http://www.ivifoundation.org/VXIPlug_Play/