



Bakalářská práce

Propojení prostředí LabView a Matlab za účelem náběru biomedicínských dat

Studijní program:

B3944 Biomedicínská technika

Studijní obor:

Biomedicínská technika

Autor práce:

Marek Vaniček

Vedoucí práce:

Ing. Vojtěch Lindauer

Fakulta zdravotnických studií

Liberec 2022



Zadání bakalářské práce

Propojení prostředí LabView a Matlab za účelem náběru biomedicínských dat

<i>Jméno a příjmení:</i>	Marek Vaníček
<i>Osobní číslo:</i>	D19000024
<i>Studijní program:</i>	B3944 Biomedicínská technika
<i>Studijní obor:</i>	Biomedicínská technika
<i>Zadávací katedra:</i>	Fakulta zdravotnických studií
<i>Akademický rok:</i>	2021/2022

Zásady pro vypracování:

Cíle práce:

- skript pro ovládání programu Labview přímo z prostředí Matlab
- vložení kódu Matlab do programu LabView
- popsání bloků pro náběr dat pomocí NI USB modulu

Teoretická východiska:

Bakalářská práce vychází z potřeby propojení a snadného vytváření systému pro náběr dat v prostředí Labview a výpočetní rychlosti aritmetických operací v prostředí Matlab. Z dostupné dokumentace vyplývá, že pro tyto prostředí je implementováno rozhraní, které tuto komunikaci umožňuje. Výstupem je pak podrobný popis komunikace, komunikačních prostředků a použitých prvků pro propojení prostředí Matlab a Labview.

Výzkumné předpoklady / výzkumné otázky:

Je možné vytvořit skript pro ovládání programu v jazyce LabView přímo z prostředí Matlab?

Jaké jsou možnosti vložení kódu '.m' a funkcí prostředí Matlab do Labview bloků?

Jaké bloky jsou potřebné pro náběr biomedicínských dat pomocí USB modulu?

Metoda:

Programování

Technika práce:

Za pomoci programu v jazyce LabView se předají data programu v jazyce Matlab a zde se provede zpracování.

Místo a čas realizace výzkumu:

Fakulta zdravotnických studií Technické univerzity v Liberci, leden – duben akademického roku 2021/2022

Rozsah grafických prací:
Rozsah pracovní zprávy: 50–70
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

ČIHÁK, Radomír. 2016. Anatomie 3. Svazek III, Periferní nervový systém, kůže a kožní orgány, smyslové orgány. 3. vyd. Praha: Grada. ISBN 978-80-247-5636-3.

EHSANI, Behzad. 2016. Data acquisition using LabVIEW: transform physical phenomena into computer acceptable data using a truly object-oriented language. Birmingham: Packt Publishing. ISBN 978-1-78217-216-1.

HAMPEL, D., J. JANOVA a L. VISKOTOVA. 2018. MATLAB. Brno: Mendelova univerzita v Brně, ISBN 978-80-7509-543-5.

JENNINGS, Richard a Fabiola DE LA CUEVA. 2019. Labview graphical programming: practical applications in instrumentation and control. 5th ed. New York: McGraw-Hill Education. ISBN 978-1-260-13526-8.

LOGANR24. 2018. Simple ECG Circuit and LabVIEW Heart Rate Program: 6 Steps – Instructables. AUTODESK. Yours for the making – Instructables [online]. Mill Valley: Autodesk, [cit. 2021-11-14]. Dostupné z: <https://www.instructables.com/Simple-ECG-Circuit-and-LabVIEW-Heart-Rate-Program/>

MATHWORKS. 2021. Help Center for MATLAB, Simulink and other MathWorks Products. MATHWORKS. MathWorks – Makers of MATLAB and Simulink [online]. Natick: MathWorks, [cit. 2021-11-14]. Dostupné z: <https://uk.mathworks.com/help/>

NATIONAL INSTRUMENTS. 2020. LabVIEW NXG 5.1 manual: interfaces for MATLAB®. NI. Engineer Ambitiously – NI [online]. Austin: NI, aktualiz. 2020-22-20 [cit. 2021-11-14]. Dostupné z: <https://www.ni.com/documentation/en/labview/latest/language-integration/interfaces-for-matlab/>

NAVRÁTIL, Leoš et al. 2019. Medicínská biofyzika. 2. vyd. Praha: Grada. ISBN 978-80-271-0209-9.

PŠENČÍKOVÁ, Jana. 2021. Algoritmizace. 2. vyd. Prostějov: Computer Media. ISBN 978-80-740-2414-6.

VEDRAL, Josef. 2020. Číslicové měřicí přístroje. Praha: Česká technika – nakladatelství ČVUT. ISBN 978-80-01-06777-2.

VEDRAL, Josef a Jakub SVATOŠ. 2018. Zpracování a digitalizace analogových signálů v měřicí technice. Praha: Česká technika – nakladatelství ČVUT. ISBN 978-80-01-06424-5.

Vedoucí práce: Ing. Vojtěch Lindauer
Fakulta zdravotnických studií

Datum zadání práce: 30. listopadu 2021
Předpokládaný termín odevzdání: 29. července 2022

L.S.

prof. MUDr. Karel Cvachovec, CSc.,
MBA
děkan

V Liberci dne 31. ledna 2022

Rozhodnutí o žádosti studenta

Jiný typ žádosti, uveďte se cíl žádosti ručně

Jméno a příjmení: **Marek Vaníček**

Osobní číslo: **D19000024**

Datum podání žádosti **11.08.2022**

Rozhodnutí děkana ze dne 25.08.2022 :

VYHOVĚL

Odůvodnění

Termín pro odevzdání závěrečné bakalářské práce Propojení prostředí LabView a Matlab za účelem náběru biomedicínských dat, pod vedením Ing. Vojtěcha Lindauera, se prodlužuje do 30.6.2023.

Poučení

Rozhodnutí nemá odvolání.

Rozhodnutí rektora ze dne

NEUVEDENO

Odůvodnění:

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Poděkování

V těchto několika řádcích bych chtěl poděkovat panu Ing. Vojtěchu Lindauerovi za vedení této bakalářské práce. Především bych chtěl poděkovat za jeho cenné odborné rady, jeho trpělivost a ochotu. Rád bych také poděkoval za konzultace a pomoc při seznámení se s přístroji použitými v této práci.

Propojení prostředí LabVIEW a MATLAB za účelem náběru biomedicínských dat

Abstrakt

Softwarové propojení dvou programovacích jazyků a systémů jako jsou LabVIEW a MATLAB je velmi žádané. To je způsobeno především kvůli jejich silným stránkám. LabVIEW vyniká na poli hardwarového řešení hlavně v oblasti měřicí techniky. MATLAB na druhou stranu je výkonným prototypovacím jazykem v oblasti matematických výpočtů a zpracování dat.

Z těchto důvodů bylo vytvořeno obousměrné propojení mezi LabVIEW a prostředím MATLAB prostřednictvím bloku v LabVIEW umožňujícím přímé provádění programu napsaného v jazyce MATLAB. Ze strany MATLABu je toto propojení realizováno pomocí Data Acquisition Toolboxu. Tato práce se zaměřuje na popis způsobů, jak s těmito prvky pracovat a jak je lze implementovat v případě záznamu biomedicínských dat, jako je EKG a zvuk.

Klíčová slova: LabVIEW, MATLAB, měření, programování, záznam dat, EKG, digitální stetoskop

LabVIEW and MATLAB environment interface for biomedical data acquisition purpose

Abstract

The software connection of two programming languages and systems such as LabVIEW and MATLAB is in great demand. This is mainly due to their strengths. LabVIEW excels in the field of hardware solutions, mainly in the field of measuring technology. MATLAB, on the other hand, is a powerful prototyping language in the field of mathematical calculations and data processing.

For these reasons, a two-way connection between LabVIEW and the MATLAB environment was created through a block in LabVIEW allowing the direct execution of a program written in the MATLAB language. From the MATLAB side, this connection is implemented using the Data Acquisition Toolbox. This work focuses on describing ways to work with these elements and how they can be implemented in the case of recording biomedical data such as ECG and audio.

Keywords: LabVIEW, MATLAB, measurement, programming, data recording, EKG, digital stethoscope

Obsah

Seznam zkratek	11
1 Úvod	15
2 NI a MathWorks	16
2.1 MATLAB	16
2.1.1 Přednosti MATLABu	16
2.2 LabVIEW	17
2.2.1 Přednosti LabVIEW	17
3 Analogově/digitální převodníky	19
3.1 Kvantování analogových dat	19
3.2 Vzorkovací teorém	20
3.2.1 Volba vzorkovací frekvence	21
4 Instalace softwarových komponent	23
4.1 Instalace pro použití NI myDAQ v prostředí MATLAB	23
4.2 Instalace pro použití NI myDAQ zařízením	23
5 Měření a zpracování dat ze zařízení NI DAQ	25
5.1 Načtení dat z souboru vytvořeného pomocí LabVIEW	25
5.1.1 Textový soubor	25
5.1.2 TDMS soubor	26
5.1.3 Soubor Excelu	28
5.2 Měření prostřednictvím MATLABu	29

5.2.1	Zaznamenání dat pomocí aplikace Analog Input Recorder	29
5.2.2	Zaznamenání dat pomocí funkcí	31
5.3	Měření pomocí LabVIEW	35
5.3.1	Záznam dat pomocí šablony v LabVIEW	35
5.3.2	Založení projektu z šablony	35
5.3.3	Šablona Continuous Measurement and Logging (NI DAQmx)	38
5.3.4	Šablona Finite Measurement (NI DAQmx)	41
5.3.5	Záznam dat pomocí asistenta	46
5.3.6	Záznam pomocí jednotlivých bloků	54
5.3.7	Záznam pomocí ukázkových souborů	56
5.4	Použití funkčních bloků Formula Node a Matlab script	58
5.5	Praktická ukáзка měření biomedicínských data za pomocí LabVIEW a MATLABu	60
5.5.1	Základní informace o použitých zařízení a jejich nastavení	60
5.5.2	Naměření dat EKG	62
5.5.3	Naměření zvuku z digitálního stetoskopu	66
6	Závěr	70

Seznam zkratek

EKG Elektrokardiogram

ECG Electrocardiography(Elektrokardiografie)

DAQ Data acquisition(Sběr dat)

NI National Instruments(nyní je „NI“ název společnosti nikoliv zkratka)

GUI Graphic User Interface(Grafické uživatelské rozhraní)

Seznam obrázků

3.1	Ukázka aplikace „Analog Input Recorder“	22
5.1	Ukázka aplikace „Analog Input Recorder“	29
5.2	Ukázka aplikace „Signal Analyzer“ s výčtem základních funkcí	31
5.3	Ukázka dat naměřených z dvou kanálů pomocí MATLABu	34
5.4	Úvodní okno po spuštění programu LabVIEW	36
5.5	Založení projektu LabVIEW z šablony	37
5.6	Nastavení nového projektu	38
5.7	Okno projektu a soubor Main.vi	39
5.8	Okno pro nakonfigurování měření	40
5.9	Vizualizace pořízeného záznamu	41
5.10	Program z šablony „Finite Measurement“	42
5.11	Okno nastavení záznamu u šablony „Finite Measurement“	43
5.12	Vizualizace dat pomocí šablony „Finite Measurement“	45
5.13	Data Acquisition kategorie s DAQ asistentem	46
5.14	Okno průvodce nastavením DAQ Assistant modulu pro výběr vstupního signálu	47
5.15	Okno výběru vstupních kanálů v průvodci nastavením DAQ Assistant modulu	48
5.16	Hlavní nastavení DAQ Assistant modulu	49
5.17	Záložka pro záznam dat v DAQ Assistant	50
5.18	Karta zapojení pinů pro vybraný vstup	51
5.19	I/O file kategorie s Write To Measurement File modulem	52

5.20	Nastavení bloku „Write Meas File“	53
5.21	Blokové schéma a přední panel programu využívající blok „DAQ Assistant“	54
5.22	Schéma zapojení pomocí jednotlivých bloků LabVIEW	55
5.23	Přední panel zapojení pro zaznamenání dat pouze pomocí jednotlivých bloků LabVIEW	55
5.24	Příklad programu s kontinuálním měřením ze složky „examples“	56
5.25	Příklad programu s konečným měřením ze složky „examples“	57
5.26	Čelní panel generátoru sinusových vln	59
5.27	Blokové schéma generátor sinusových vln	60
5.28	Měřicí sestava pro záznam EKG	62
5.29	Záznam EKG zobrazený v aplikaci „Signal Designer“	64
5.30	Použití filtrů v nastavení „Preprocess“	65
5.31	Využití funkce „Panner“, přiblížení výsledného signálu	66
5.32	Měřicí sestava pro záznam srdečních ozev	66
5.33	Čelní panel programu pro záznam audio signálu z digitálního stetoskopu	67
5.34	Blokové schéma použité pro záznam audio signálu z digitálního stetoskopu	67

Seznam tabulek

5.1	Tabulka základních specifikací zařízení NI myDAQ	61
-----	--	----

1 Úvod

Tato práce popisuje oboustranné propojení zmíněných prostředí MATLAB a LabVIEW. Práce slouží zároveň jako návod pro instalaci a využití, za pomoci popsaných postupů s příloženými skripty jako příklady. Tento požadavek vzešel díky častému používání těchto dvou prostředí samostatně, kdy obě ve své oblasti přinášejí dobré výsledky. LabVIEW je vytvořeno za účelem snadného programování měření. MATLAB je koncipován pro analýzu takovýchto dat.

V druhé kapitole jsou popsány základní rozdíly mezi těmito dvěma prostředími a popisují se zde také odlišnosti v přístupu k programování jednotlivých programů.

Třetí kapitola je věnována dvou základním pojmům a to kvantování a vzorkování. Bez pochopení těchto pojmů by nebylo možné správně nastavit parametry pro záznam jakéhokoli signálu.

Po této kapitole následuje krátká kapitola věnující se instalaci softwarových komponent potřebných pro příklady v kapitole pět.

Poslední pátá kapitola se již jakožto praktická část této práce věnuje jednotlivým postupům a metodám pro samotný záznam dat. Postupně se zde popisují metody, které využívají v maximální míře prostředí MATLAB, přes metody kde je využití rovnocenné, až po metody které využívají pouze prostředí LabVIEW. Na konci této kapitoly je pak jedna sekce věnována praktickému využití v biomedicínské technice. Jako příklad bylo zvoleno měření EKG a záznam z digitálního stetoskopu tedy akustického signálu.

Práce je koncipována zároveň jako návod pro použití například při výuce budoucích biomedicínských techniků, ale lze ji použít i při jiných činnostech, kde je možné tohoto propojení využít.

2 NI a MathWorks

Tato kapitola se zabývá programovacími prostředími a jejich tvůrci. Popisují se zde výhody a nevýhody jednotlivých řešení od instalace až po programování samotné.

2.1 MATLAB

MATLAB je programovací skriptovací jazyk a programové prostředí. MATLAB vyvinula společnost MathWorks, která byla založena v roce 1984. Jedná se o americkou firmu se sídlem v Natick ve státě Massachusetts. Mezi přední produkty firmy patří nejen MATLAB ale také Simulink, který slouží pro simulaci různých systémů. Simulink je na rozdíl od MATLABu je Simulink stejně jako LabVIEW grafickým programovacím jazykem. MATLAB a Simulink jsou spolu úzce propojeni [3].

2.1.1 Přednosti MATLABu

MATLAB je programovací skriptovací jazyk s mnoha implementovanými funkcemi. MATLAB v sobě nese zkrácení názvu „matematická laboratoř“ z toho již jasně vyplývá že jeho hlavní předností jsou matematické výpočty. V MATLABu se snadno aplikují maticové výpočty. Mezi přednosti MATLABu patří také matematické modelování a pokročilé matematické výpočty. MATLAB disponuje funkcemi pro řešení parciálních diferenciálních rovnic. Dalším příkladem by pak mohla být možnost aplikovat metodu konečných prvků pro numerické řešení diferenciálních rovnic. Při zpracování signálů je pak velmi užitečná tvorba filtrů.

MATLAB je vhodný jazyk pro prototypování. Díky tomu není potřeba příliš širokých znalostí programování, které jsou ale jistě výhodou. Pokud s MATLABem

začínáme jistě nás také potěší velmi dobře zpracovaná a rozsáhlá dokumentace ve formě HTML stránek.

Mezi další výhody MATLABu jsou desítky toolboxů, které mají svá vizuální prostředí a skutečně usnadňují práci. Dobrá je i podpora takzvaných souborů „MEX“ což jsou soubory obsahující kód v jazyce C nebo C++. MATLAB má i podporu dalších jazyků jako je populární Python nebo Java, či Fortran.

MATLAB disponuje značnou všestranností v oblasti zpracování dat a proto je dostupnost funkcí pro jejich reprezentaci důležitý prvkem.

Z hlediska zaznamenávání dat prostřednictvím NI DAQ zařízení je jistě výhodná užší spolupráce s některými výrobci hardwaru používaného k záznamu dat. Konkrétně pro zařízení NI DAQ má podporu v Data Acquisition Toolboxu. Tato podpora je sice omezená, ale dostačující [1].

2.2 LabVIEW

LabVIEW je grafické vývojové prostředí vyvinuté společností National Instruments. National Instruments nyní pouze NI, je americká nadnárodní společnost se sídlem v Austinu státě Texas. Společnost NI, se zaměřuje na výrobu zařízení sloužících k automatickému testování, měření a strojovému vidění. Obvykle tedy slouží ke sběru dat a řízení přístrojů a zařízení. Organizace byla založena v roce 1976. Mezi hlavní produkty patří jak softwarové LabVIEW, tak hardwarové řešení ve formě DAQ systémů [14].

2.2.1 Přednosti LabVIEW

LabVIEW na rozdíl od MATLABu využívajícího k programování skriptovací jazyk, využívá LabVIEW k tvorbě programu jazyk grafický. LabVIEW vzniklo s myšlenkou vytvořit program z blokového schématu bez potřeby programátora, který by toto schéma převedl do programovacího jazyka. Z toho vyplývá, že znalosti programování potřebné k tvorbě jednoduchého programu v LabVIEW jsou minimální. Naproti tomu pro někoho kdo přechází z programování v skriptovací jazyce do La-

LabVIEW, můžou jeho počátky být těžké.

Největší výhodou LabVIEW v tvorbě programu, je jednoduchost sestavováním jednotlivých bloků skoro jako puzzle. Prostředí samo vyhodnocuje zda je dané spojení přípustné či ne a uživatele na to upozorní. Snadno se také vytváří nové bloky, které se následně spojují s jinými dle potřeby.

Oproti skriptovacímu jazyku se zde jednoduše vytváří také paralelní procesy. To je způsobeno tím, že výsledný program je spouštěn v podobě virtuálního přístroje. Ten je řízen takzvaným „tokem dat“. LabVIEW je zaměřeno hlavně na měření dat, ve kterém také vyniká. LabVIEW je proto užíváno pro automatizaci měření v mnoha oblastech, jelikož umožňuje generovat i výstupní signály pro řízení [15].

3 Analogově/digitální převodníky

Signály, se kterými se v technice můžeme setkat, jsou spojitého charakteru. Mezi ně lze zařadit intenzitu světla, napětí, které je proměnlivé v čase, nebo například rychlost chemických reakcí, které jsou přímo závislé na teplotě. Pro tyto účely byly vytvořeny převodníky které jsou schopny tyto analogové signály převést na jejich digitální podobu. V této podobě jsou již zpracovatelné pro počítače, bez jejichž výpočetního výkonu by ve většině případů nebylo možné tyto data zpracovat.

Digitální informace se pojí oproti signálu analogovému se dvěma hlavními neduhy. Prvním je vzorkování, tedy rozdíl v čase zaznamenání hodnoty. S tímto pojmem se pojí počet vzorků za sekundu [S/s] („samples per second“, počet vzorků za sekundu). Kvantování je pak problémem samotné naměřené hodnoty. Udává přesnost s jakou jsme schopni danou hodnotu naměřit.

Oba tyto problémy způsobují ztrátu přesnosti a důvěryhodnosti převedeného signálu a snižuje se zde kvantita a kvalita informací. V následujících sekcích bude vysvětleno vše potřebné k pochopení problematiky převodu měřených analogových dat.

3.1 Kvantování analogových dat

Kvantování je jedním z problémů digitalizace spojitých (reálných dat) do digitální (počítačové) podoby. Počítače pracují na základě informací uložených v základních jednotkách, bitech. Dnes jsou informace nejčastěji reprezentovány jako hodnoty uložené ve více bitech seskupených po 8. Takovéto seskupení se nazývá Byte.

Podle toho, kolik bitů reprezentuje jednu informaci, lze hovořit o přesnosti informace. S tím se pojí kvantizační chyba. Předvedeme-li si to na příkladu s čísly, bude

to vypadat následovně. Berme v úvahu dvě čísla, například číslo 2 a 10. Pokud bychom vzaly tyto hodnoty, potřebujeme odlišné množství bitů potřebných k jejich uložení. Číslo 2 by se vyjádřilo jako „10“ v dvojkové soustavě. Potřebujeme tedy pouze dva bity. Naproti tomu číslo 10 se vyjádří v dvojkové soustavě jako hodnota „1010“. Jak je patrné, pro uložení této hodnoty je potřeba 4 bitů, tedy dvojnásobné množství. V minulých řádcích byl použit pojem dvojková soustava. Tato soustava je používána v „řeči“ počítačů a nabývá pouze hodnot logická 0 nebo logická 1, také označováno jako „true“ a „false“.

Abychom si blíže popsali chybu, která z tohoto příkladu vyplývá, použijeme jiný příklad. Máme 6 čísel reprezentujících naměřená napětí [0.10, 0.25, 0.43, 0.56, 0.901.5] a 3-bitový převodník s rozsahem 0 – 1V. Nyní převedeme hodnoty do bitové podoby na výstupu převodníku. Hodnoty se tvoří následovně. Máme rozsah jednoho voltu a 3 bity na rozlišení. Hodnoty jsou rozlišovány rovnoměrně, tedy pokud máme 3 bity tak je to 8 možností. Mezní hodnoty jsou tedy následovné [0, 0.1429, 0.2857, 0.4286, 0.5714, 0.7143, 0.8571, 1.0000]. Výsledné hodnoty, které dostaneme budou v desítkové soustavě [0, 1, 4, 4, 7, 8]. Jak je na první pohled patrné, chyba je zde opravdu velká, jelikož jsme náš původní signál, který měl rozlišení na jednu deseti tisícinu, tedy 10000 hodnot, nyní popsali pouze 8 různými číslicemi. Tento příklad je pouze pro znázornění. Běžnou praxí je použití 8-bitových, 16-bitových nebo i 24 či 32-bitových převodníků. Kvantizační chybu lze popsat jako model náhodného šumu, kde v důsledku nepřesnosti rozlišení je ke každému vzorku přidána malá část odpovídajícího šumu. Ten má například u 16-bitového převodníku hodnotu přibližně $1/227000$. tento model má jednu velkou výhodu a tou je, že stačí tento šum přidat k šumu, který už je přítomen [17].

3.2 Vzorkovací teorém

Řešení otázky týkající se volby vzorkovací frekvence je jednoduché. Stačí, aby bylo možné proces vzorkování zvrátit, tedy ze vzorků replikovat původní signál. Určení vzorkovací frekvence je klíčovou vlastností pro zpracování analogového signálu. Po-

kud jsou data špatně navzorkována, dochází ke zmatení dat či jejich neúplnosti. Zmatením je myšleno například snížení frekvence u sinusového signálu. Příkladem neúplnosti dat by pak byla ztráta krátkého píku, jednoduše proto, že se do něj vzorkovací frekvence kvůli své dlouhé periodě netrefí.

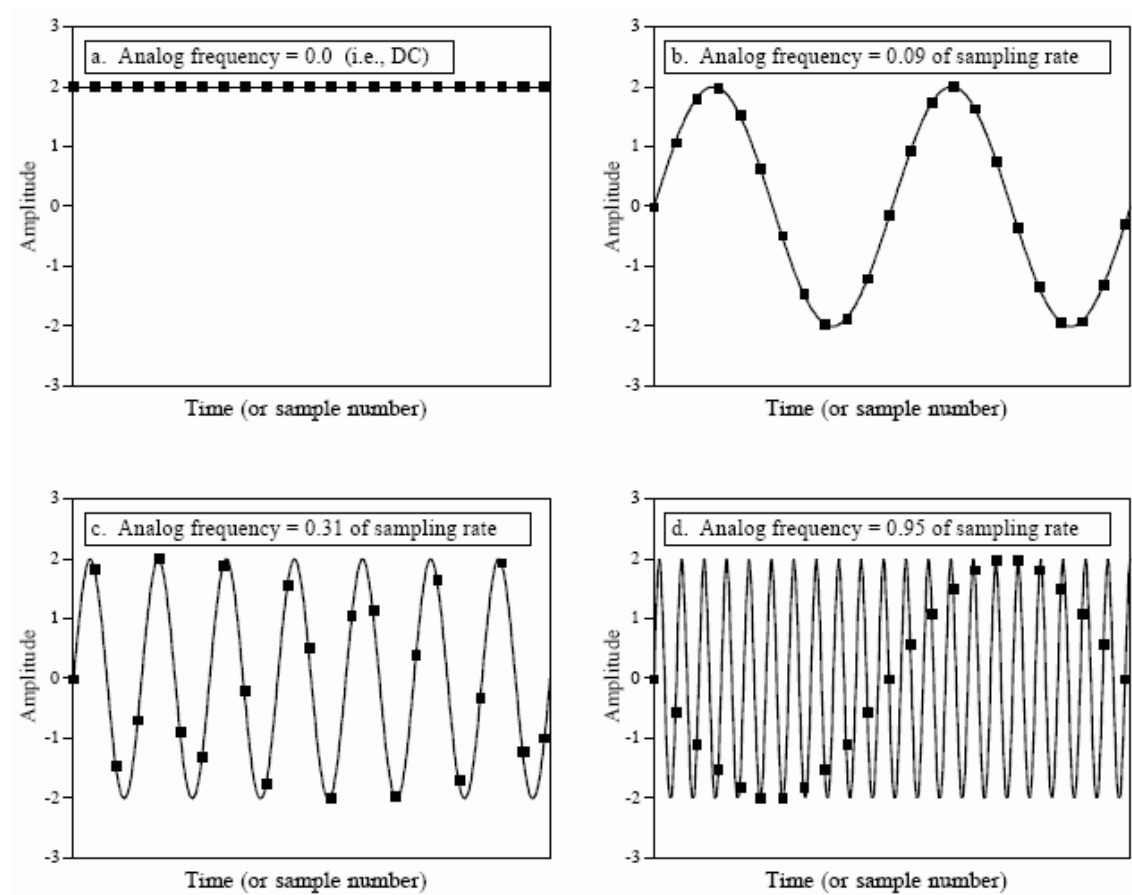
3.2.1 Volba vzorkovací frekvence

Předpokládejme sinusový signál na vstupu (vpravo nahoře 3.1), který má frekvenci 0,09 vzorkovací frekvence. Takováto hodnota by mohla nabývat například 90 cyklům za sekundu, při odpovídající rychlosti vzorkování 1000 vzorků za sekundu. Pokud tyto hodnoty přepočteme dojdeme k 11,1 cyklům na jednu periodu sinusoidy. Abychom dokázaly tento signál zpětně rekonstruovat není možné pouze naměřené hodnoty propojit lineární aproximací. Otázkou tedy je zda tyto hodnoty dostatečně reprezentují tuto křivku. Odpověď na tu to otázku je: „Ano reprezentují“. Vychází to díky tomu, že v rozumných mezích neexistuje žádná jiná sinusoida či jejich kombinace, která by spolehlivě propojila naměřené body. Tyto body tak odpovídají pouze jedné podobě analogového signálu a jsou tedy věrohodnou reprezentací původního analogového signálu.

Pokud poměr vzorkovací frekvence zvýšíme (vlevo dole 3.1) na hodnotu 0,31 vzorkovací frekvence. Dostaneme pouhých 3,2 vzorku na jednu periodu sinusového signálu. V tomto případě by se mohlo na první pohled zdát, že vzorky nemají žádný specifický trend, podle kterého by se řídili. Avšak není tomu tak. Hodnoty jsou opět dobrou reprezentací původního signálu, jelikož i zde je možné hodnoty v rozumných mezích propojit pouze jednou sinusoidou či kombinací více sinusových vln.

Graf (vpravo dole 3.1) je znázorněním změny poměru vzorkovací frekvence v mnohem vyšší hodnotě a to 0,95. Počet vzorků je tedy pouze 1,05 vzorku na jednu periodu. Položíme-li opět stejnou otázku týkající se správnosti reprezentace dat, správně dojdeme k závěru tehdy, bude-li naše odpověď nyní záporná. Data zcela očividně nerepresentují původní data. Tato vlna se v tomto případě špatně repre-

zentuje jako vlna o frekvenci 0,05. Tento jev se nazývá Aliasing. Jednoduše se dá tento princip přirovnat k falešné identitě, která má z účel zmást nepřítele. Sinusoida v digitální podobě zde přebírá frekvenci jiné sinusoidy, která se v signálu vůbec nenachází. V tomto případě se tedy jedná o nesprávný způsob vzorkování [16].



Obrázek 3.1: Ukázka aplikace „Analog Input Recorder“

vlevo nahoře - záznam DC signálu, **vpravo nahoře** a **vlevo dole** - ukázka správné volby vzorkovací frekvence, **vpravo dole** - příklad špatné volby vzorkovací frekvence

4 Instalace softwarových komponent

Prvním krokem pro zaznamenání dat pomocí LabVIEW a MATLABu je instalace potřebných softwarových komponent. Tato kapitola v krátkosti shrnuje jaké softwarové komponenty, jsou potřeba pro měření ukázaná v nadcházející kapitole.

4.1 Instalace pro použití NI myDAQ v prostředí MATLAB

Proto abychom mohli měřit na zařízení myDAQ od společnosti NI v prostředí MATLAB. Potřebujeme mít nainstalováno prostředí MATLAB [5] a to nejlépe nejnovější verzi. Dále budeme potřebovat ovladače pro DAQ software pro zařízení DAQ [11]. Vzhledem k tomu, že záznam dat se provádí skrze toolbox Data Acquisition Toolbox [6], budeme potřebovat stáhnout a nainstalovat i ten. To provedeme tak, že přejdeme na liště hlavního okna MATLABu do záložky „HOME“, klikneme na „ADD-ons“ a vyhledáme heslo „Data Acquisition Toolbox“. Tento toolbox by měl být navrchu nabídky výsledků. Toolbox nainstalujeme a softwarové prostředky máme připraveny.

4.2 Instalace pro použití NI myDAQ zařízením

Pro práci s NI myDAQ budeme potřebovat oficiální software LabVIEW, který je nabízen i v nekomerční verzi. Stažení a následná instalace je možná z oficiálních stránek společnosti NI [9]. Pro práci s myDAQ budeme potřebovat také ovladače k těmto účelům slouží balíček myDAQ Software Suite [10]. Po jehož nainstalování

je zařízení připraveno k provozu, pouze je nutný restart počítače. Od verze 5.1 je v LabVIEW přímo zabudovaná podpora pro spouštění souborů MATLABu, takže není nutná další doinstalace pro jejich používání [7].

5 Měření a zpracování dat ze zařízení NI DAQ

V této kapitole jsou popsány různé přístupy a možnosti jak pracovat s LabVIEW a MATLABem společně či separovaně.

5.1 Načtení dat z souboru vytvořeného pomocí LabVIEW

V této sekci jsou popsány způsoby jakými lze data uložená pomocí příkladů ukázaných v této kapitole nahrát do prostředí MATLAB k jeho zpracování. Jsou zde použity tři formáty. Formát „.txt“ tedy textový soubor. Excelový soubor „.xls nebo .xlsx“, tedy tabulkový soubor. A TDMS soubor což je typ souboru, který používá NI pro ukládání záznamů měření.

5.1.1 Textový soubor

Textový soubor má výhodu spustitelnosti téměř na jakémkoliv zařízení. Není však možné provádět výpočty a další zpracování, a proto je nutné importovat data do programu, který těmito vlastnostmi disponuje. Naměřená data pomocí LabVIEW uložená v „.txt“ lze importovat do MATLABu o něco složitěji nežli u Excelového souboru nebo souboru TDMS. Pro načtení dat můžou posloužit následující řádky kódu.

Prvním krokem bude otevření souboru pomocí funkce `fopen()`, která přebírá argument obsahující název souboru i s příponou „.txt“. Další funkci kterou použijeme je funkce `textscan()`. První argument představuje objekt vytvořený funkcí `fopen()`, druhý pak typ dat, která chceme načíst, v našem případě se jedná o data v podobě textu.

```
1         f = fopen("Log_data.txt");
2         data = textscan(f, '%s');
```

Následně data očistíme od hlavičky a převedeme na pole „stringů“. Na pole stringů data převádíme pro jednodušší manipulaci.

```
1         data = string(data{:});
2         data = data(11:end);
```

Nyní data poskládáme tak aby v jednom sloupci byl čas a v druhém naměřená data. Na řádku s číslem 2 se provádí sloučení obsahu prvního a druhého sloupce, tak že se mezi ně vloží znak pro mezeru. Následně je k nim řádkově přiřazen třetí sloupec.

```
1         data = reshape(data, 3, [])';
2         data = [data(:,1) + " " + data(:,2), data(:,3)];
```

Nakonec odstraníme objekt souboru, který jsme vytvořili.

```
1         fclose(f);
```

5.1.2 TDMS soubor

Základní soubor pro ukládání dat z měření vyvinutý a spravovaný společností NI. Jedná se o soubory s koncovkou „.tdms“. Tento soubor obsahuje informace v hi-

erarchickém rozložení tří úrovní. Kdy nejvyšší úrovně je objekt specifikující název souboru nebo například autora. Každý takovýto objekt pak dále obsahuje neomezené množství skupin, které mají taktéž neomezený počet kanálů.

Prvním krokem pro nahrání souboru do prostředí MATLAB je specifikovat název souboru s koncovkou „.tdms“. Následně toto jméno předáme funkci `tdmsinfo()`. Pro další práci nás bude zajímat seznam kanálů, který vyčteme z tabulky vrácené právě funkcí `tdmsinfo()`.

```
1         fileName = 'Log_data.tdms';
2         info = tdmsinfo(fileName);
3         chanelList = info.ChannelList
```

Nyní definujeme skupinu kanálů a kanál, který chceme importovat do MATLABu.

```
1         group = '_unnamedTask<1>';
2         channel = 'myDAQ1/ai0';
```

V dalším kroku pak extrahujeme potřebná data. Do proměnné „chanelProp“ uložíme veškerá data, která by se při následném zpracování můžou hodit. Proměnná „stepTime“ nám definuje krok s jakým byly data naměřena. Z této hodnoty lze odvodit vzorkovací frekvenci záznamu. Počáteční čas záznamu je pak definován v proměnné „startTime“.

```
1         chanelProp = tdmsreadprop(fileName, ...
                ChannelGroupName=group, ChannelName=channel)
2         timeStep = tdmsreadprop(fileName, ...
                ChannelGroupName=group, ChannelName=channel, ...
                PropertyNames='wf_increment')
3         startTime = tdmsreadprop(fileName, ...
                ChannelGroupName=group, ChannelName=channel, ...
                PropertyNames='wf_start_time')
```


Data, která jsme získali v předešlých řádcích nyní využijeme v posledním kroku kde získáme výsledky uložené v proměnné typu „timetable“. První sloupec obsahuje čas pořízení záznamu a v druhém sloupci je vepsána hodnota naměřené fyzikální veličiny, s názvem sloupce specifikujícím kanál, na kterém byla data pořízena.

```
1 data = tdmsread(fileName, ...
    StartTime=startTime.wf_start_time, ...
    TimeStep=seconds(timeStep.wf_increment));
```

5.1.3 Soubor Excelu

Data uložená v souboru Excelu jsou snadno čitelná a umožňují jednoduchou práci právě v tomto programu od společnosti Microsoft. Pro větší výpočty a datovou analýzu větších data však může být tento program nedostačující. Import dat do MATLABu je velmi jednoduchý.

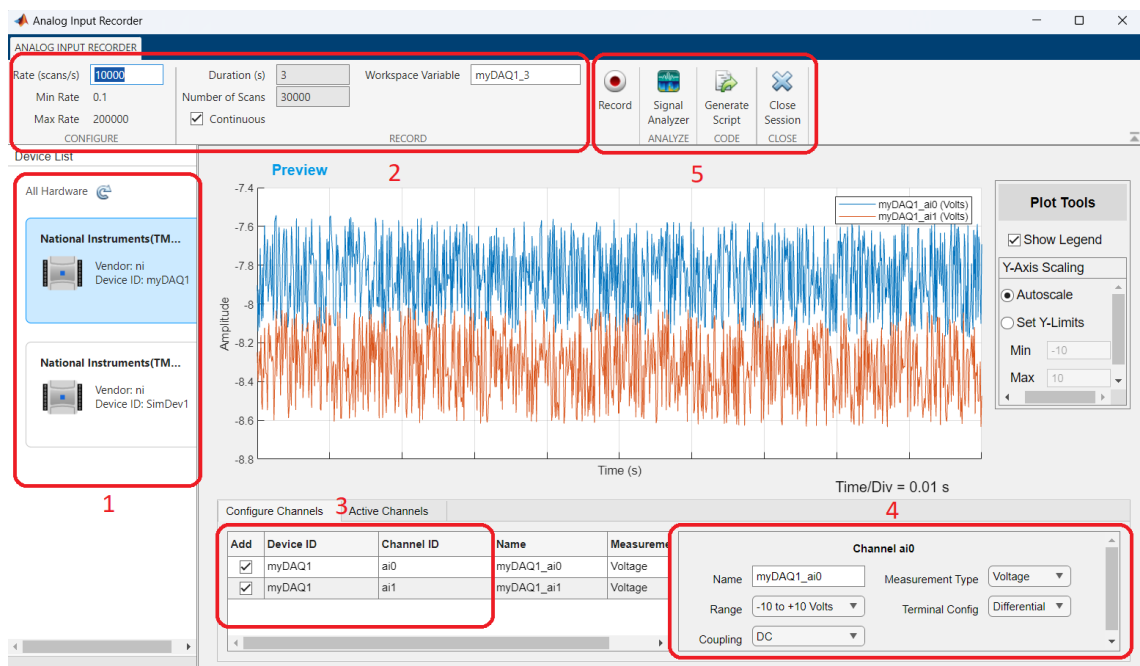
Abychom Importovali data do MATLABu můžeme využít tlačítka v záložce „Home“ nesoucí název „Import Data“. Tato cesta je vhodná pokud chceme vybrat pouze specifickou část dat. Pokud však potřebujeme nahrát tato data všechna, jednoduše použijeme pouze následující dva řádky kódu. Pokud bychom chtěli data v jiném formátu, například maticovém, místo tabulkového. Stačí nahradit funkci `readtable()` funkcí `readmatrix()` a vstup se nemění.

```
1 filename = 'test.xlsx';
2 data = readtable(filename);
```

5.2 Měření prostřednictvím MATLABu

5.2.1 Zaznamenání dat pomocí aplikace Analog Input Recorder

Nejjednodušším zaznamenáním dat ze zařízení DAQ pomocí MATLABu je využití aplikace „Analog input recorder“. Tato aplikace nám umožňuje rychle zaznamenat data díky jednoduchému nastavení a plně automatizovanému procesu propojení. Postup záznamu je následující.



Obrázek 5.1: Ukázka aplikace „Analog Input Recorder“

1 - Výběr vstupního zařízení, 2 - nastavení základních vlastností měření, 3 - výběr měřených kanálů, 4 - nastavení vlastností jednotlivých kanálů, 5 - tlačítko spuštění a pomocné aplikace

Na obrázku 5.1, kde je zachyceno hlavní okno programu, jsou očíslované sekce jednotlivých částí nastavení. Nejprve začneme výběrem zařízení na kterém chceme data měřit a to v oblasti 1. V našem případě máme vybrané zařízení s označením myDAQ1. V 2. kroku nastavujeme počet záznamů za sekundu. Pod tímto polem je také napsána minimální a maximální hodnota, kterou je dané zařízení schopno

přijmout. V našem případě zvolíme vzorkovací frekvenci s hodnotou 10kHz. Vedle nastavení vzorkovací frekvence si můžeme nastavit časové vlastnosti záznamu a to buď v sekundách a celkový počet záznamů se dopočítá, nebo naopak. Máme zde také možnost spustit záznam jako kontinuální a přerušit ho ručně. Více vpravo je pak nastavení názvu proměnné, do které se naměřená data uloží. Tuto proměnnou následně najdeme v „MATLAB workspace“. Na místě označeném číslem 3 si pak vybereme, které vstupní kanály chceme pro záznam použít. Pod číslem 4 pak nastavujeme vlastnosti pro jednotlivé kanály. Můžeme zde nastavit jméno, měřenou veličinu a typ vstupu. Nalezneme zde také nastavení pro záznam stejnosměrného nebo střídavého proudu. Poslední možností nastavení vstupu je rozsah, v jakém se budou hodnoty nacházet, zde se tak nastavuje kvantovací krok. V poslední části označené číslem 5 máme vlevo tlačítko, které spustí námi nastavený záznam. Třetí tlačítko nám umožňuje vygenerovat skript kopírující naše nastavení. To se může hodit pokud vytváříme vlastní aplikaci a měření dat je její součástí. Tlačítko úplně vpravo ukončí celý běh program. Druhé tlačítko je odkaz na aplikaci Signal Analyzer viz 5.2, který může být velmi dobrým nástrojem pro základní zpracování naměřených dat. Umožňuje nám signál měřit v různém rozsahu. Poskytuje frekvenční a frekvenčně časovou analýzu. Užitečnou funkcí je pak rozhodně statistická analýza, která je užitečná například při záznamu zvuku. Při kliknutí na tlačítko „Input Recorder“ v aplikaci se nám data automaticky nahrají do aplikace „Signal Analyzer“, avšak tuto aplikaci můžeme spustit i samostatně z nabídky aplikací v MATLABu. Pokud tak učiníme, data, která chceme analyzovat pak jednoduše přesuneme z nabídky vpravo dole, kde jsou vybrána analyzovatelná data z „MATLAB Workspace“. Uprostřed aplikace máme okno grafu a vlevo od něj pak jeho nastavení. V tomto grafu se nám zobrazují námi naměřená data a mi je tak můžeme ihned vizuálně zkontrolovat. Samozřejmostí je nastavení os grafu či možnost ukázat legendu.



Obrázek 5.2: Ukázka aplikace „Signal Analyzer“ s výčtem základních funkcí

5.2.2 Zaznamenání dat pomocí funkcí

Pro zaznamenání dat pomocí funkcí začneme vyčištěním pracovního prostoru, příkazového pole a zavřením všeho co je otevřené.

```

1           clc;
2           clear;
3           close all;

```

Pokračujeme detekcí dostupných zařízení NI. To provedeme následujícím příkazem.

```

1           daqlist

```

Funkce daqlist nebo daqlist() se volá bez argumentu. Její návratová hodnota je tabulka vypisující všechny hardwarové komponenty vhodné pro použití s Data Acquisition Toolbox. Pokud není očekávané zařízení v seznamu je možné, že je Data

Acquisition Toolbox nepodporuje nebo nejsou nainstalovány potřebné ovladače. Bližší informace jsou dostupné v dokumentaci. (příkaz - doc daqlist)

Abychom vyfiltrovali pouze zařízení firmy NI, zadáme následující příkaz.

```
1 list = daqlist('ni') % nacte zarizeni NI
2 deviceID = list.DeviceID % vyjme ID zarizeni
```

Při vložení argumentu typu char 'ni' jsou vráceny pouze zařízení typu NI. Tento argument může nabývat i jiných hodnot dle hledaného typu zařízení, možnosti jsou dostupné v dokumentaci. Následující kód předpokládá pouze jedno připojené zařízení NI. V opačném případě by bylo nutné specifikovat řádek vybraného zařízení pomocí závorkové syntaxe "deviceID = list.DeviceID(x)". x = řádek vybraného zařízení. Bližší informace jsou dostupné v dokumentaci: příkaz - doc daqlist

Po zadání následujícího kódu je možné zjistit informace ohledně vstupů a výstupů zařízení dostupných pro Data Acquisition toolbox. Jednička zde specifikuje řádek tabulky, na kterém se zařízení nachází. "DeviceInfo" je název sloupce tabulky.

```
1 deviceInfo = list{1, 'DeviceInfo'} % vypise ...
   informace o zarizeni
```

Nyní vytvoříme daq objekt se specifikací pro zařízení NI. Funkce daq s argumentem typu char 'ni' vytvoří Data Acquisition objekt pro práci se zařízením NI. Bližší informace jsou dostupné v dokumentaci: příkaz - doc daq

```
1 daqDevice = daq('ni') % vytvori daq objekt
```

Když máme vytvořený DaqAcquisition objekt můžeme přidat kanál vstupu. Pro přidání využijeme funkci addinput, která má čtyři argumenty. Prvním je daqDevice, což je objekt daq vytvořený pomocí funkce daq. Druhým je deviceID, jenž obsahuje

ID zařízení zjištěné z tabulky vytvořené pomocí funkce `daqlist`. Třetím argumentem je `channel`, což je název kanálu/ů. Posledním čtvrtým argumentem je `measureType`, což je veličina, ve které chceme měřit. Bližší informace jsou dostupné v dokumentaci: příkaz - doc `daq.addinput`.

```
1         channel = 'ai0'
2         measureType = 'Voltage'
3         addinput(daqDevice, deviceID, channel, ...
                 measureType) % přida vstup mereni
```

Teď se již dostáváme k načtení dat ze zařízení NI. Data jsou ze zařízení načítána pomocí funkce `read`. Ta přebírá dva argumenty, zařízení `daq`, a čas po který měření probíhá. Čas je udáván v sekundách. Návrátovou hodnotou této funkce je tabulka datového typu `timetable`(specifický datový typ v prostředí MATLAB, více v dokumentaci ("doc `timetable`")). U záznamu je možné kromě délky záznamu v čase měnit i frekvenci za sekundu s jakou jsou data snímána. Tato hodnota se mění změnou proměnné "Rate"u objektu `daq`.

```
1         daqDevice.Rate = 20000
2         time = seconds(2)
3         data = read(daqDevice, time)
```

První sloupec výstupních data ve formátu `timetable` je čas. Každý další sloupec jsou pak data naměřená v jednotlivých kanálech viz 5.3. Bližší informace jsou dostupné v dokumentaci: příkaz - doc `daq.read`

	Time	1 myDAQ1_ai1	2 myDAQ1_ai0	3	4	5	6	7	8	9	10	11
1	0 sec	-0.0599	0.8832									
2	0.0039062 sec	-0.0367	1.0309									
3	0.0078125 sec	0.0104	0.8344									
4	0.011719 sec	-0.0176	1.0846									
5	0.015625 sec	0.0392	0.9955									
6	0.019531 sec	-0.0385	1.0636									
7	0.023438 sec	0.0259	0.8570									
8	0.027344 sec	-0.0227	0.5585									
9	0.03125 sec	-0.0481	0.7688									
10	0.035156 sec	-0.0413	0.6503									
11	0.039062 sec	0.0598	0.8188									
12	0.042969 sec	-0.0444	0.5548									
13	0.046875 sec	-0.0575	0.7791									
14	0.050781 sec	0.0054	0.8957									
15	0.054688 sec	0.0139	0.9180									
16	0.058594 sec	-0.0383	0.9522									
17	0.0625 sec	-0.0040	0.7663									
18	0.066406 sec	-0.0511	0.9201									
19	0.070312 sec	0.0361	1.0376									
20	0.074219 sec	0.0047	0.7379									

Obrázek 5.3: Ukázka dat naměřených z dvou kanálů pomocí MATLABu

Spuštění programu LabVIEW pomocí MATLABu

Jednou z možností jak automatizovat proces měření pomocí MATLABu je spuštění jednotlivých procesů měření, respektive programů LabVIEW, měřících data na přístroji, pomocí propojení ActiveX. Následujících několik řádků ukazuje použití tohoto propojení pro spuštění generátor, který je vytvořen v sekci 5.4.

Na prvním řádku se vytvoří do proměnné „server“ ActiveX server specifikovaný pro LabVIEW aplikace. Proměnná „vipath“ specifikuje úplnou cestu k souboru „.vi“, ve kterém je uložen náš program. Lze použít i relativní cestu. Pomocí funkce invoke() vytvoříme spojení, které nám umožňuje komunikovat s programem.

```

1         server = actxserver('LabVIEW.Application');
2         vipath = 'C:\Users\marku\OneDrive - ...
              TUL\Plocha\Math block\Untitled.vi';
3         vi = invoke(server, 'GetVIREference', vipath);

```

Nyní spustíme pomocí příkazu „!start“ náš program. Můžeme použít jak relativní adresu tak úplnou záleží v jakém umístění se naše soubory nachází. Teď je program spuštěn. Pro zahájení měření spustíme proces pomocí tečkové notace příkaz „Run“

na objektu „vi“. Funkce `pause()`, která pozdrží činnost na specifikovanou dobu, je zde kvůli času měření, který je potřeba pro provedení záznamu. Nyní se program v LabVIEW dokončil a mi můžeme program zavřít, to lze provést pomocí příkazu „CloseFrontPanel“ obdobně jako u příkazu „Run“.

```
1      !start Untitled.vi
2      vi.Run;
3      pause(10);
4      vi.CloseFrontPanel;
```

5.3 Měření pomocí LabVIEW

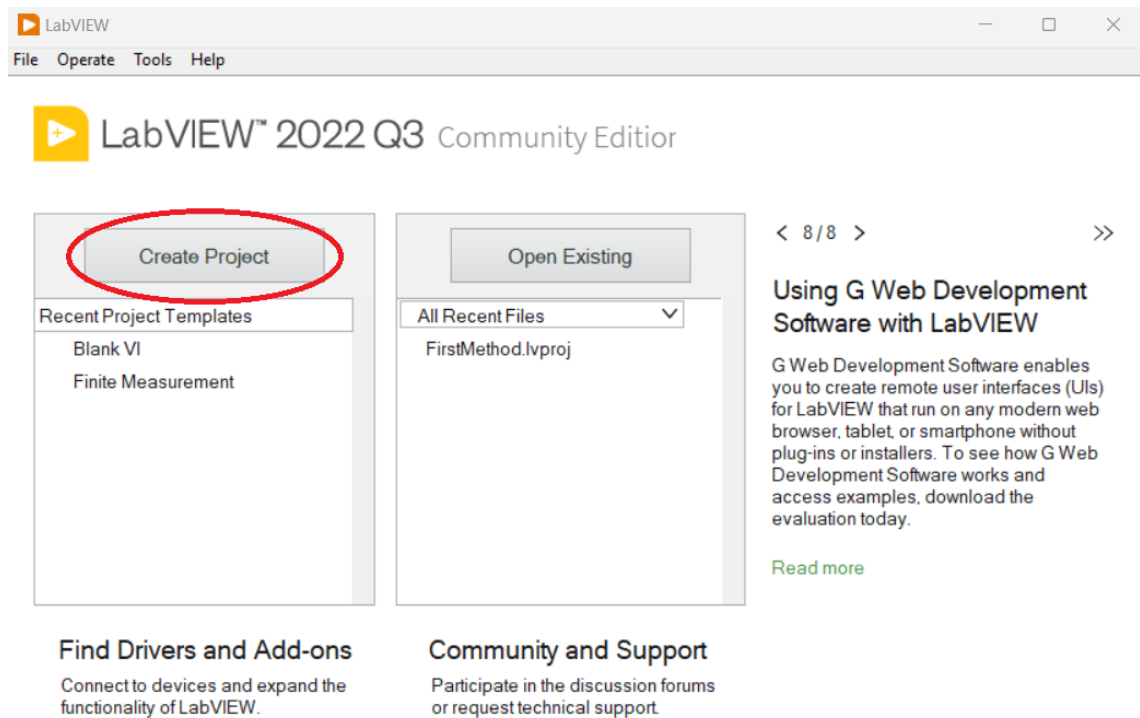
Při použití jazyka LabVIEW pro záznam dat je možné využít předem připravených šablon, nebo vytvořit vlastní kód. Pro následující ukázky využijeme zařízení NI myDAQ.

5.3.1 Záznam dat pomocí šablony v LabVIEW

Nejednodušším výběrem postupu pro záznam dat je volba šablony. Tyto předem připravené programy obsahují veškeré potřebné nastavení a náležitosti pro uskutečnění záznamu.

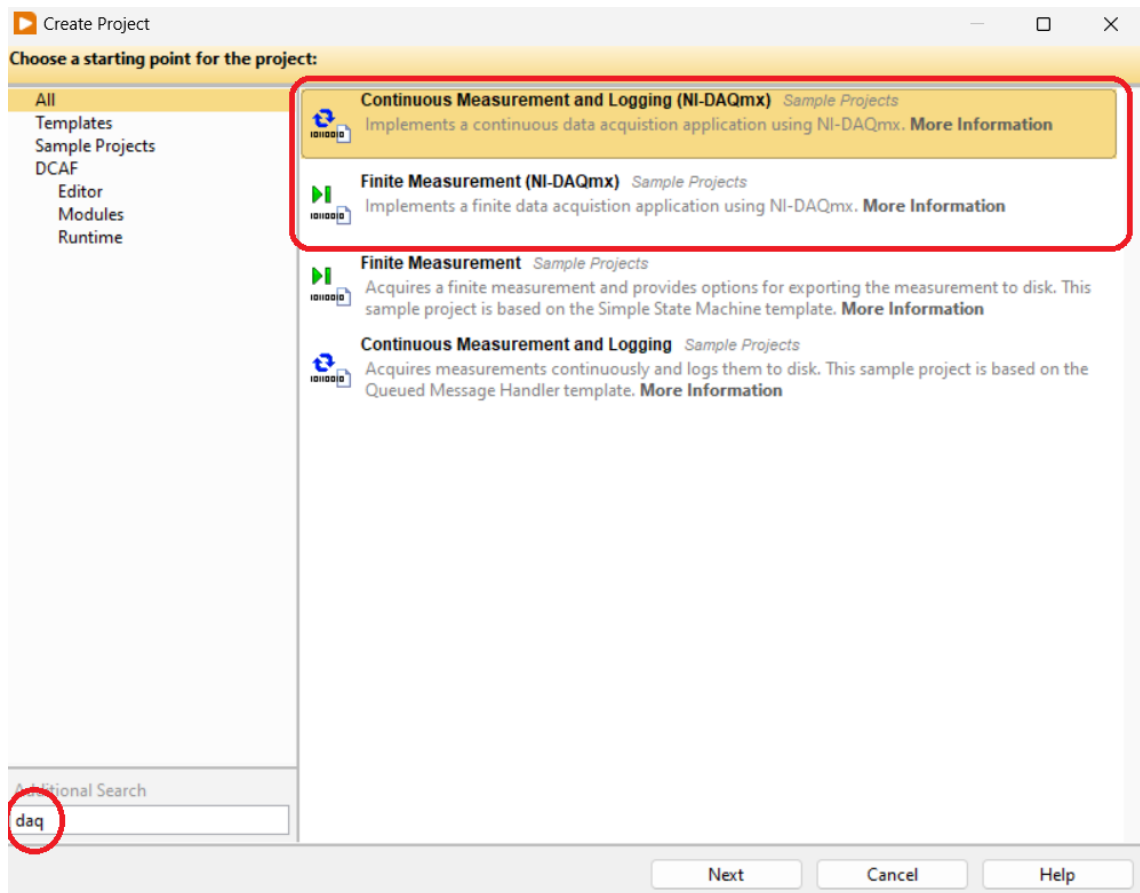
5.3.2 Založení projektu z šablony

Založení projektu ze šablony je rychlé a jednoduché. Pro založení nového projektu otevřeme program LabVIEW. Následně klikneme na tlačítko „Create Project“, které se nachází v červeně vyznačené oblasti na obrázku 5.4.



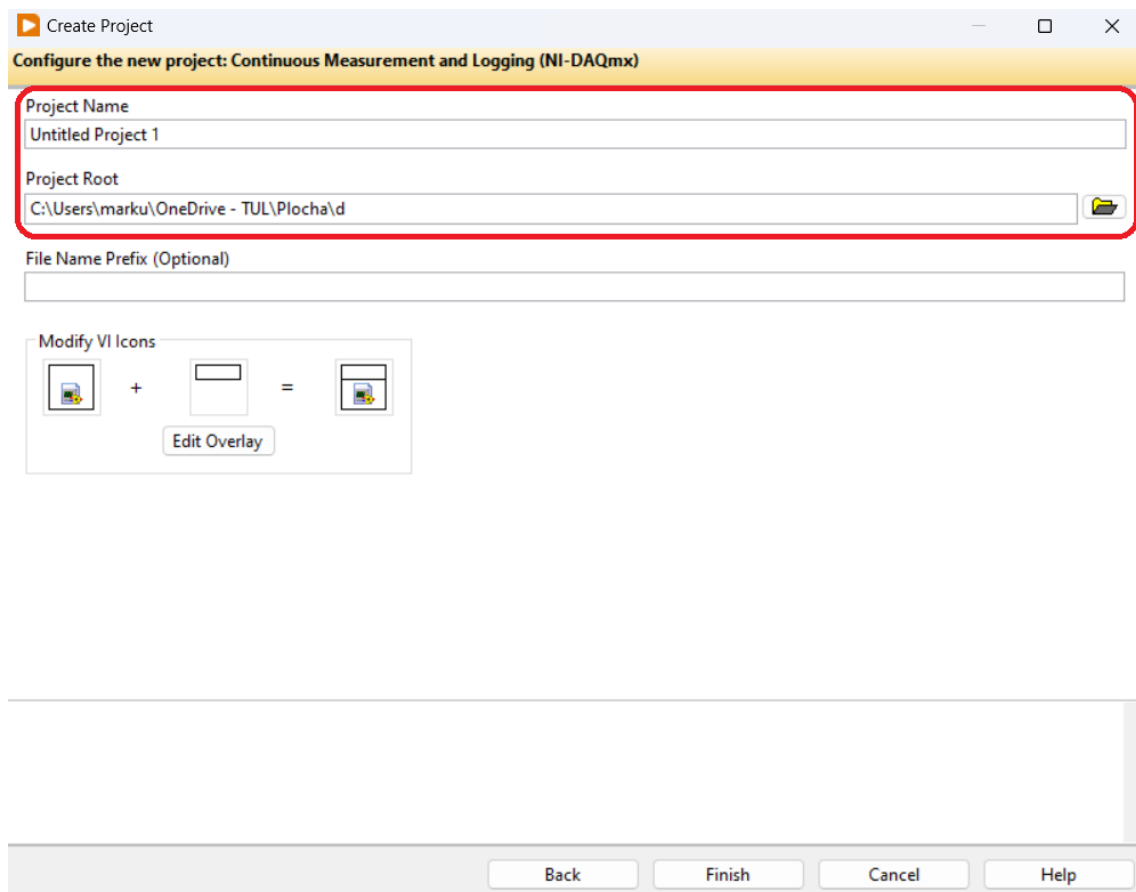
Obrázek 5.4: Úvodní okno po spuštění programu LabVIEW

Po kliknutí se otevře okno pro nastavení projektu. Pro zjednodušení hledání zadáme do vyhledávacího pole v levém dolním rohu okna výraz „daq“. Z vyhledaných výsledků nás budou zajímat dva konkrétní, jsou to „Continuous Measurement and Logging (NI-DAQmx)“ a „Finite Measurement (NI-DAQmx)“. Jak takové okno po vyhledání klíčového výrazu vypadá je zachyceno na obrázku 5.5. V obou případech se po dvojkliku nebo výběru a následném kliknutí na tlačítko „next“ objeví nové okno.



Obrázek 5.5: Založení projektu LabVIEW z šablony

V tomto okně, viz 5.6, se nachází nastavení základních parametrů nového projektu. My věnujeme pozornost pouze kolonce „Project Name“ a „Project Root“. Do kolonky „Project Name“ zadáme název našeho projektu. Pro vyplnění „Project Root“ použijeme tlačítko složky na pravé straně a za pomoci tohoto nástroje vybereme náš zvolený adresář projektu. Adresář samozřejmě můžeme vyplnit i ručně. Po vyplnění příslušných polí, klikneme na tlačítko „Finish“ na dolní liště.

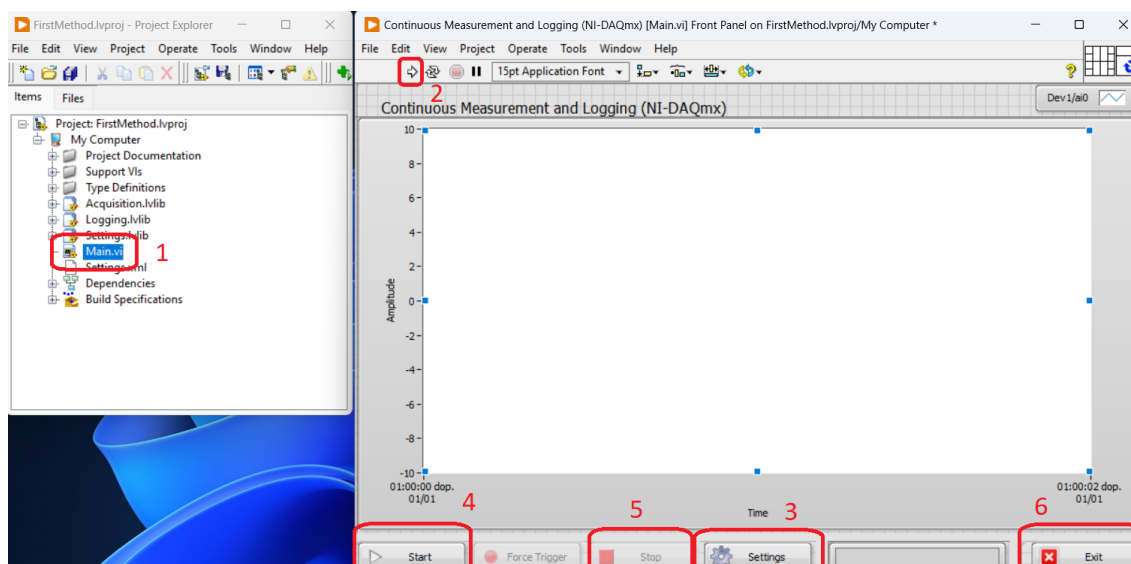


Obrázek 5.6: Nastavení nového projektu

5.3.3 Šablona Continuous Measurement and Logging (NI DAQmx)

Po založení projektu podle postupu v sekci 5.3.2 z šablony „Continuous Measurement and Logging (NI DAQmx)“ jsme vytvořili program pro záznam analogového signálu z NI DAQ zařízení. Pro zaznamenání dat pak postupujeme následovně.

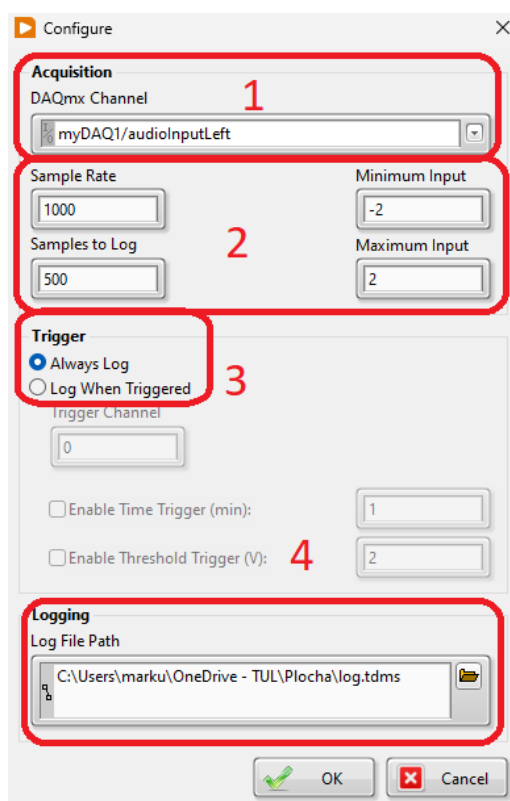
Otevřeme námi vytvořený projekt. V projektu vyhledáme soubor s názvem „Main.vi“ viz 5.7 číslo 1, a otevřeme ho. Na displeji máme nyní dvě okna viz 5.7. Pro zapnutí programu klikneme na tlačítko šipky označená na obrázku 5.7 číslem 2. Po zapnutí provedeme nastavení parametrů pro záznam dat. Klikneme na tlačítko „Settings“ označené číslem 3 na obrázku 5.7.



Obrázek 5.7: Okno projektu a soubor Main.vi

1 - hlavní soubor projektu (Main.vi), 2 - tlačítko „run“ pro spuštění souboru, 3 - nabídka nastavení, 4 - tlačítko „Start“ pro zahájení měření, 5 - tlačítko „Stop“ pro ukončení měření, 6 - tlačítko „Exit“ pro ukončení celého programu

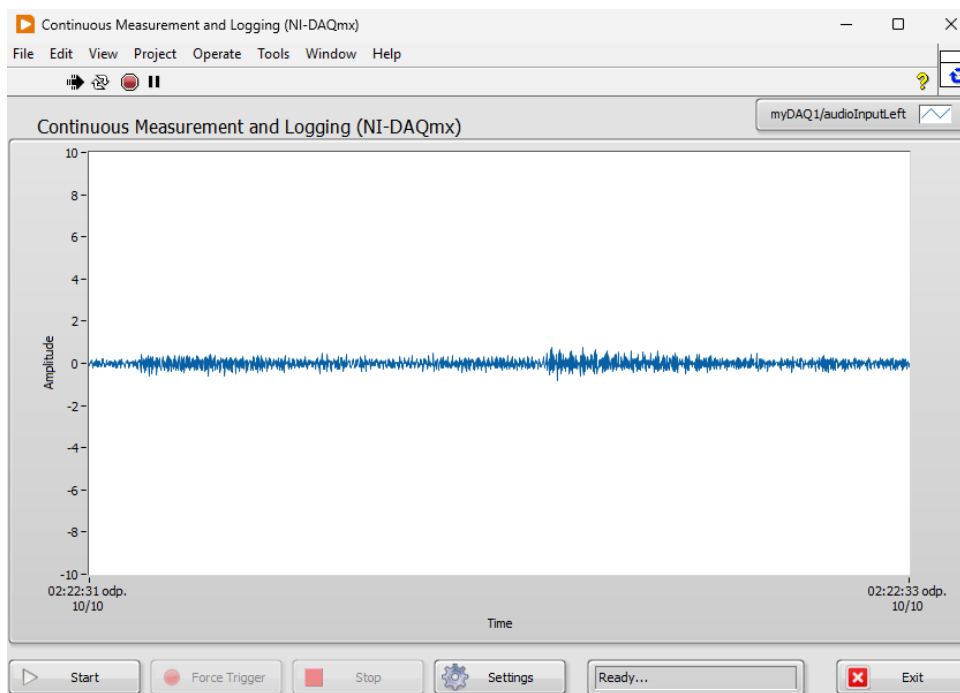
Nyní máme otevřené okno konfigurace 5.8. Toto okno lze rozdělit do 4 částí. Tyto části označují na obrázku 5.8 čísla 1-4. 1. částí je nastavení vstupního kanálu z zařízení. V našem případě se jedná o levý audio vstup. Možné je samozřejmě vybrat jakýkoliv port dostupný na zařízení. Avšak u některých kanálů je potřeba následně adekvátně nastavit i jiné parametry. V části označené číslem 2 nastavíme na levé straně vzorkovací frekvenci a počet vzorků ukládaných dávkově do „logovacího“ souboru. V pravé části pak nastavíme horní a dolní rozsah vstupních hodnot. V našem případě zadáme rozsah od -2 do 2 voltů, který je nutný pro audio vstup. V části 3 zvolíme možnost „Always Log“. Nyní se bude záznam ukládat každých 500 vzorků. Pokud bychom nechtěli ukládat takto kontinuálně za sebou, nýbrž bychom chtěli začít zaznamenávat například až poté, kdy se na některém ze vstupů objeví startovní signál, zvolíme druhou možnost a nastavíme si námi požadované vstupní hodnoty. V poslední části označené 4 volíme cestu k „logovacímu“ souboru. Opět je zde možné napsat cestu ručně nebo využít nástroje po kliknutí na tlačítko s obrázkem složky. Nyní máme vše nastavené, stačí tedy kliknout na tlačítko „Ok“.



Obrázek 5.8: Okno pro nakonfigurování měření

1 - volba záznamového zařízení a kanálu, 2 - nastavení základních parametrů, 3 - nastavení „loggingu“, 4 - volba umístění „logovacího“ souboru

Teď si můžeme spustit záznam kliknutím na tlačítko „Start“ pro spuštění záznamu. Pro dokončení záznamu použijeme tlačítko „Stop“. Jejich poloha je znázorněna na obrázku 5.7 pod čísly 4 a 5.

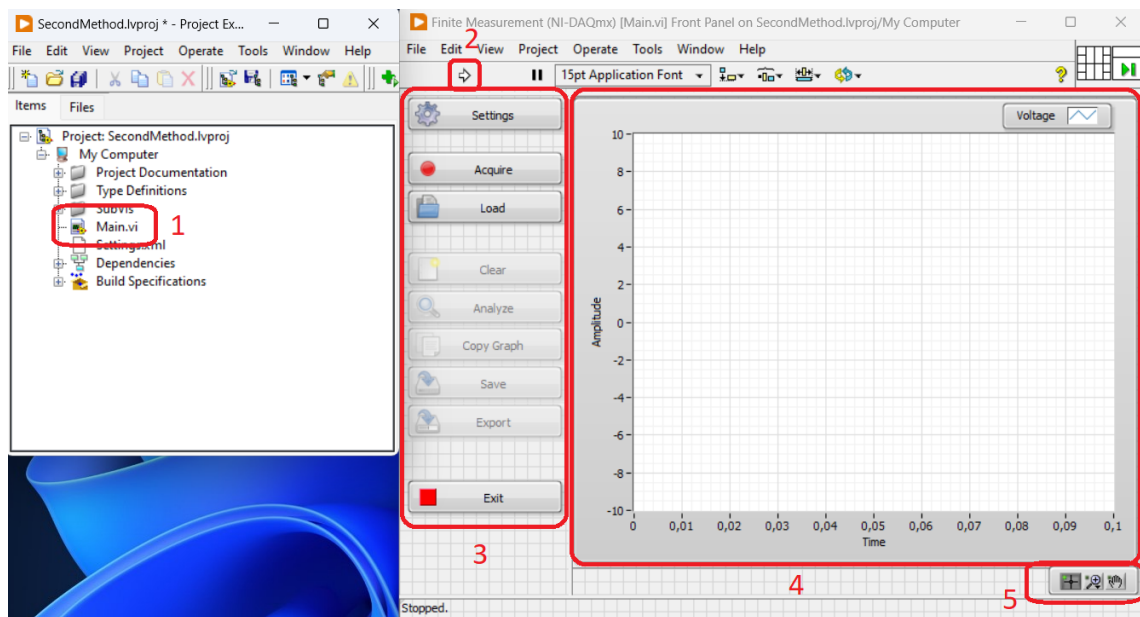


Obrázek 5.9: Vizualizace pořízeného záznamu

Při zaznamenávání dat jsou vzorky vizualizovány do spojitého grafu, viz obrázek 5.9. Zaznamenávání je možné sledovat téměř v reálném čase. Poslední tlačítko pod číslem 6 na obrázku 5.7 je ukončení celého programu.

5.3.4 Šablona Finite Measurement (NI DAQmx)

Po založení projektu podle postupu v sekci 5.3.2 z šablony „Finite Measurement (NI DAQmx)“ jsme vytvořili program pro záznam analogového signálu z NI DAQ zařízení. Tento program je již komplexnějším řešením. Pro zaznamenání dat postupujeme následovně.

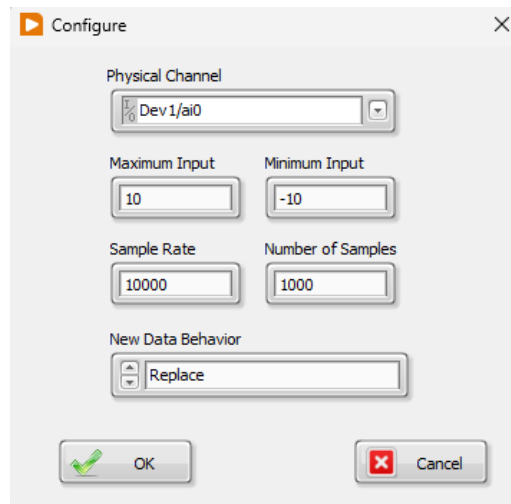


Obrázek 5.10: Program z šablony „Finite Measurement“

1 - hlavní soubor projektu (Main.vi), 2 - tlačítko „run“ pro spuštění souboru, 3 - oblast tlačítek specifikovaných zde 5.3.4, 4 - výstupní graf, 5 - panel s funkcemi pro manipulaci s grafem

Otevřeme námi vytvořený projekt. V projektu vyhledáme soubor s názvem „Main.vi“, viz 5.10 číslo 1, a otevřeme ho.

Nyní máme otevřené okno programu, viz 5.10. Pro jeho zapnutí klikneme na šipku označenou číslem 2. Na obrázku je také vyznačena část 3 kde jsou jednotlivá funkční tlačítka, která umožňují následující úlohy.



Obrázek 5.11: Okno nastavení záznamu u šablony „Finite Measurement“

Settings (Nastavení) Tlačítko „Settings“ otevře okno konfigurace viz 5.11. Zde můžeme postupně, od shora, nastavit Vstupní zařízení a jeho příslušný kanál. Následně pak maximální a minimální hodnotu vstupu, v tomto případě ve voltech. Níže je nastavitelná vzorkovací frekvence a nalevo od ní se nachází celkový počet vzorků. Poslední částí nastavení je způsob, jakým se budou data ukládat. Máme zde na výběr ze tří možností. První možností je „Replace“. Tímto nastavením nová data přepíší ta stará, která jsou tímto odstraněna. Pokud bychom chtěli novým měřením navázat na to předešlé, zvolíme možnost „Append“. Tím zajistíme že se nová data zapíše přímo za ta předešlá. Naskytne-li se nám potřeba měřit data, která je vhodné porovnat, můžeme zvolit třetí možnost, „Overlay“ jenž data zapíše samostatně, ale minulá data zachová. Nová data jsou pak znázorněna odlišnou barvou.

Acquire (Záznam) Toto tlačítko spouští záznam z vybraného zdroje. Nastavení tohoto zdroje a dalších vlastností, je možné v okně, po stisku tlačítka „Settings“.

Load (Načtení) Pokud již máme data zaznamenaná a pouze chceme provést jejich vizuální kontrolu či jiné úpravy v tomto programu. Můžeme využít této

připravené funkce. Po stisku tlačítka jsme vyzváni k výběru souboru z našeho disku. Ten vybereme a následně potvrdíme tlačítkem „Ok“.

Clear (Vyčištění) Pokud máme v programu nahraná data, tímto tlačítkem je můžeme odstranit.

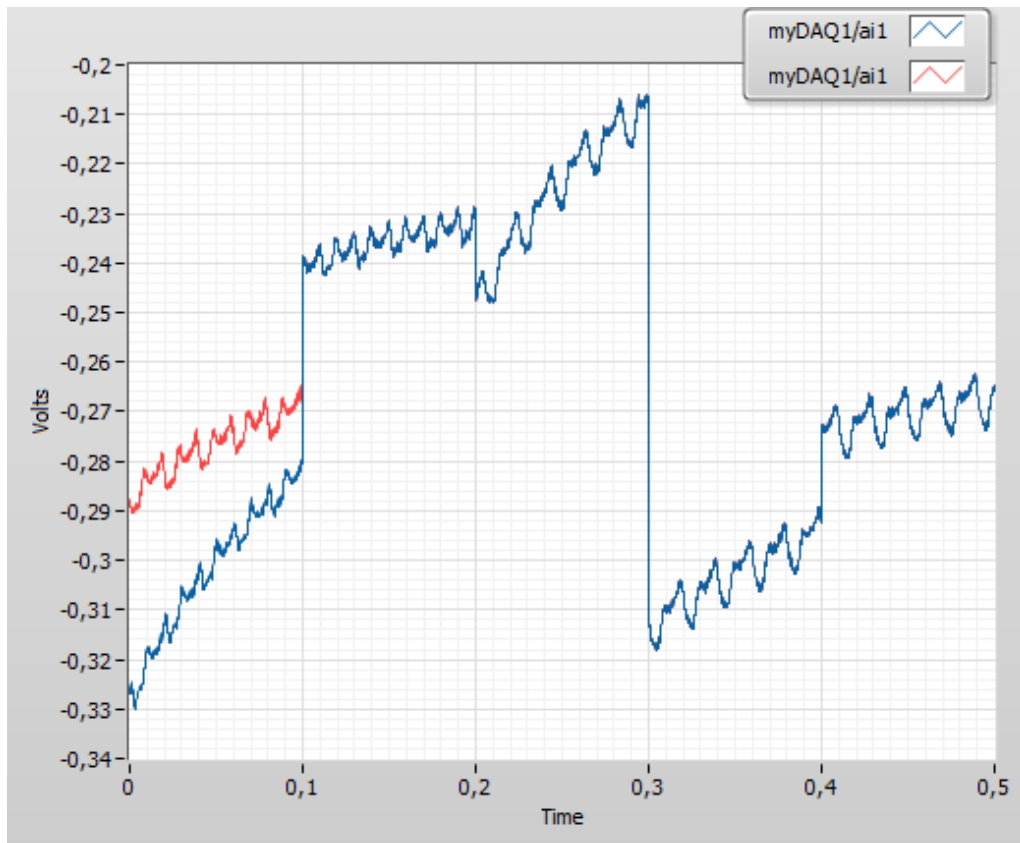
Analyze (Analýza) Tlačítko analýzy spočítá a zobrazí průměrnou hodnotu našeho signálu.

Copy Graph (Kopírovat graf) Tlačítko „Copy Graph“ uloží aktuální stav námi vytvořeného grafu do schránky počítače.

Save (Uložit) Naměřený signál je samozřejmě možné také uložit. Uložení provedeme pomocí tohoto tlačítka. Po jeho stisku nás program vyzve k výběru cesty a názvu souboru, do kterého chceme záznam uložit. Měření se nám uloží jako soubor „.tdms“ s časovými údaji o záznamu v názvu.

Export (Exportovat) Pokud bychom chtěli využít program, který neumožňuje načtení souboru „.tdms“, dobře nám poslouží funkce tohoto tlačítka. Díky ní můžeme náš soubor uložit jako textový soubor s příponou „.txt“.

Exit (Ukončit) Po kliknutí na toto tlačítko se provede ukončení celého programu a dojde ke ztrátě naměřených dat, která nebyla uložena.



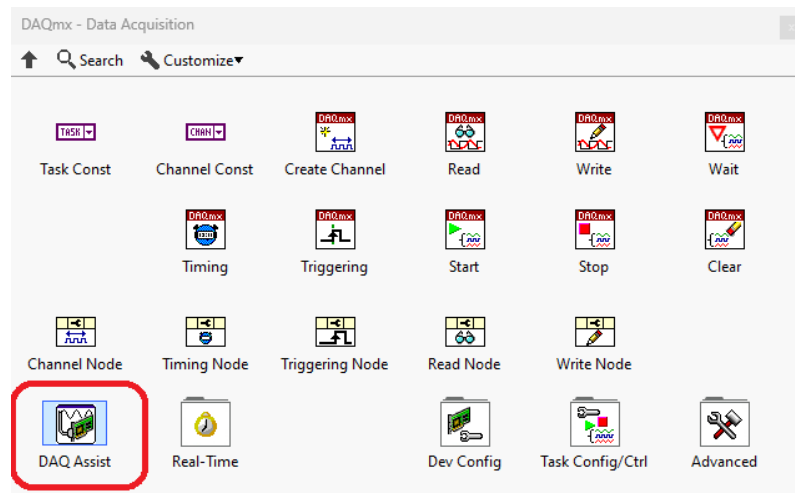
Obrázek 5.12: Vizualizace dat pomocí šablony „Finite Measurement“

Tato šablona má velkou výhodu v podobě možností nastavení grafu a legendy při vizualizaci dat, viz 5.12. Při kliknutí pravým tlačítkem myši na graf se nám ukáže nabídka možností. Mezi ně patří například možnost kopírovat nebo exportovat graf, vypnutí či zapnutí automatického zvětšování a zmenšování os, zobrazení jednotek, přidání anotace, či které prvky grafu se mají zobrazovat. Podobně je možno nastavit i legendu.

Pod číslem 5 na obrázku 5.10 pak máme velice užitečnou nabídku s nástroji pro manipulaci s grafem. Graf je možno libovolně posouvat, zvětšovat, nebo ho zmenšovat. K přiblížení a oddálení pak máme na výběr z několika způsobů, jako je výběr rozsahu pouze vertikálně či horizontálně, nebo výběrem specifické oblasti. Samozřejmostí je i navrácení do původního nastavení, pokud se nám to nepovede.

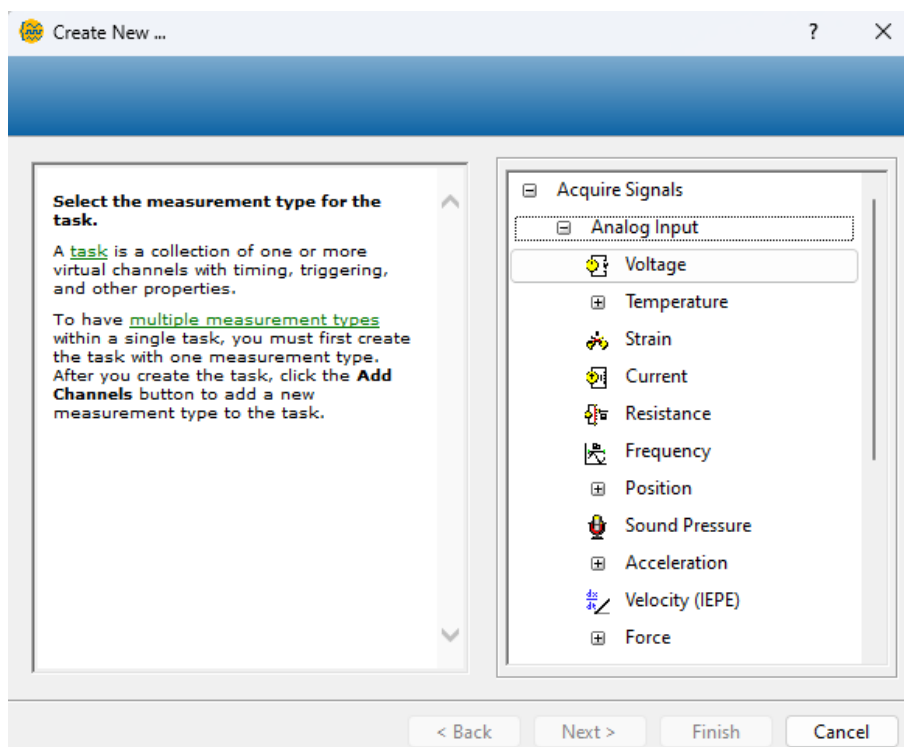
5.3.5 Záznam dat pomocí asistenta

Záznam dat pomocí asistenta je nejvhodnější volbou, pokud chceme jednoduše vytvořit záznamový program ze zařízení NI DAQ. Nastavení je komplexní a velmi přehledné. Pro zaznamenání dat a jejich uložení potřebujeme pouze dva bloky.



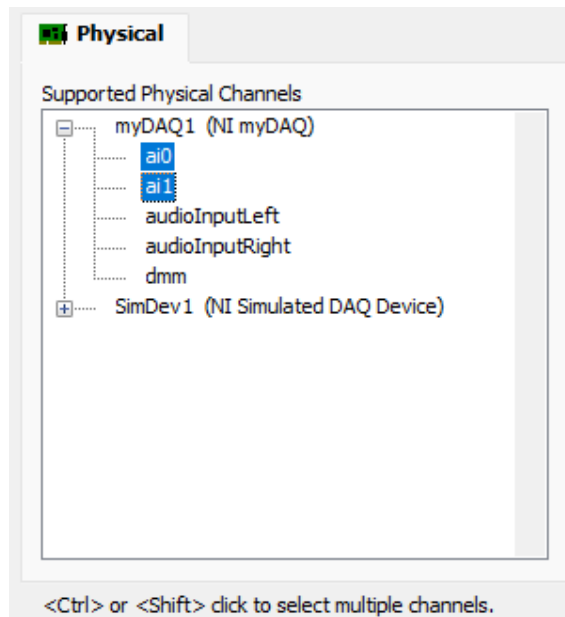
Obrázek 5.13: Data Acquisition kategorie s DAQ asistentem

Prvním blokem je blok DAQ Assistant, viz červeně označený blok na obrázku 5.13. Po přetažení bloku do blokového diagramu se automaticky spustí průvodce nastavením tohoto bloku, viz obrázek 5.14.



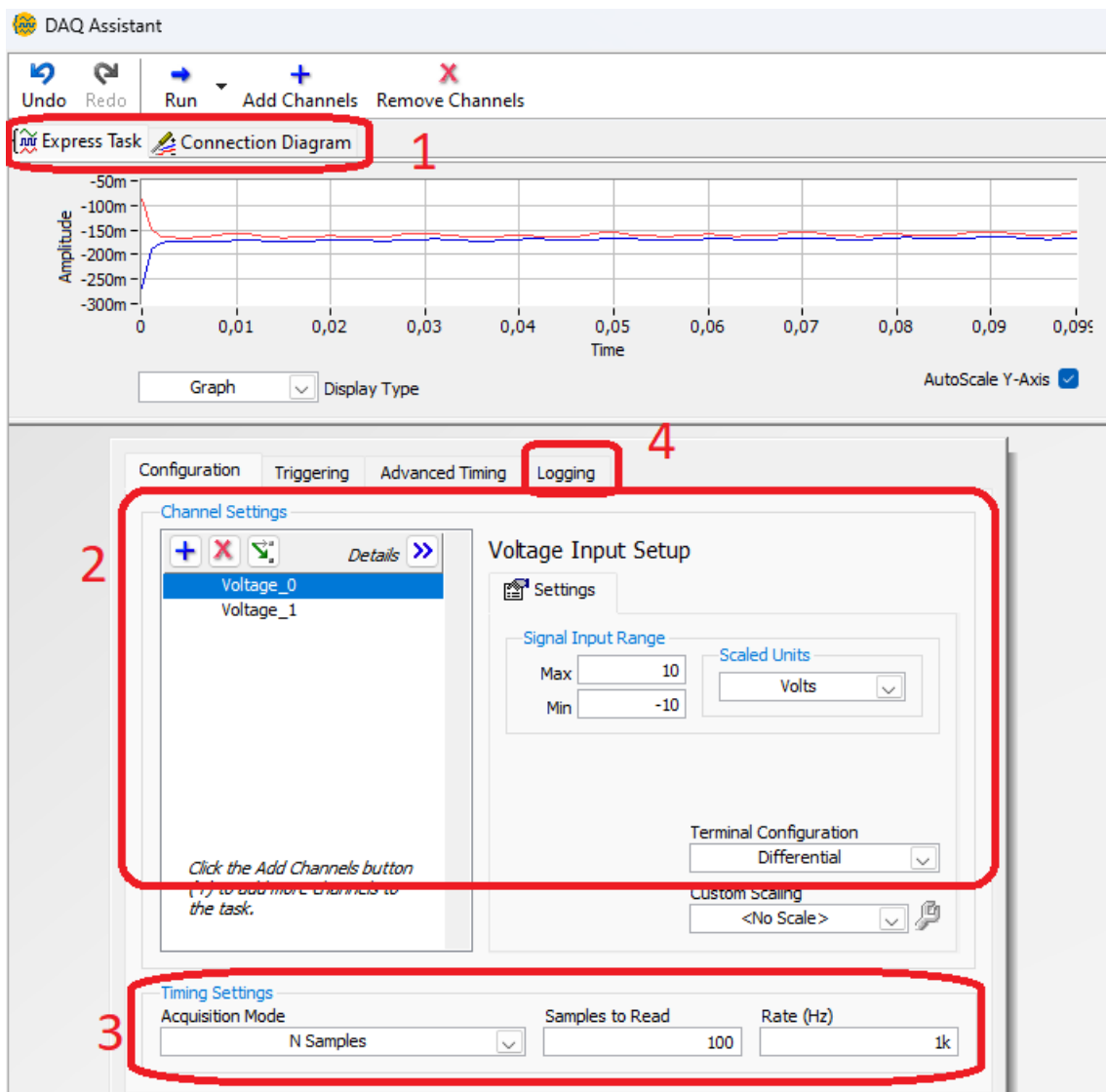
Obrázek 5.14: Okno průvodce nastavením DAQ Assistant modulu pro výběr vstupního signálu

Zde v našem případě vybereme kliknutím volbu „Voltage“ pro nastavení vstupního signálu jako elektrické napětí. Následně se nám zobrazí okno výběru vstupního či vstupních kanálů. Toto okno je zachyceno na obrázku 5.15.



Obrázek 5.15: Okno výběru vstupních kanálů v průvodci nastavením DAQ Assistant modulu

V tomto případě jsme vybrali dva vstupní kanály a to vstupní kanál „ai0“ a vstupní kanál „ai1“, viz 5.15. Výběr více kanálů provedeme podržením tlačítka „shift“, a nebo tlačítka „ctrl“ při výběru. Nakonec naši volbu potvrdíme tlačítkem „Finish“. Vzápětí se objeví okno hlavního nastavení pro záznam. Vzhled tohoto okna je zachycen na obrázku 5.16.

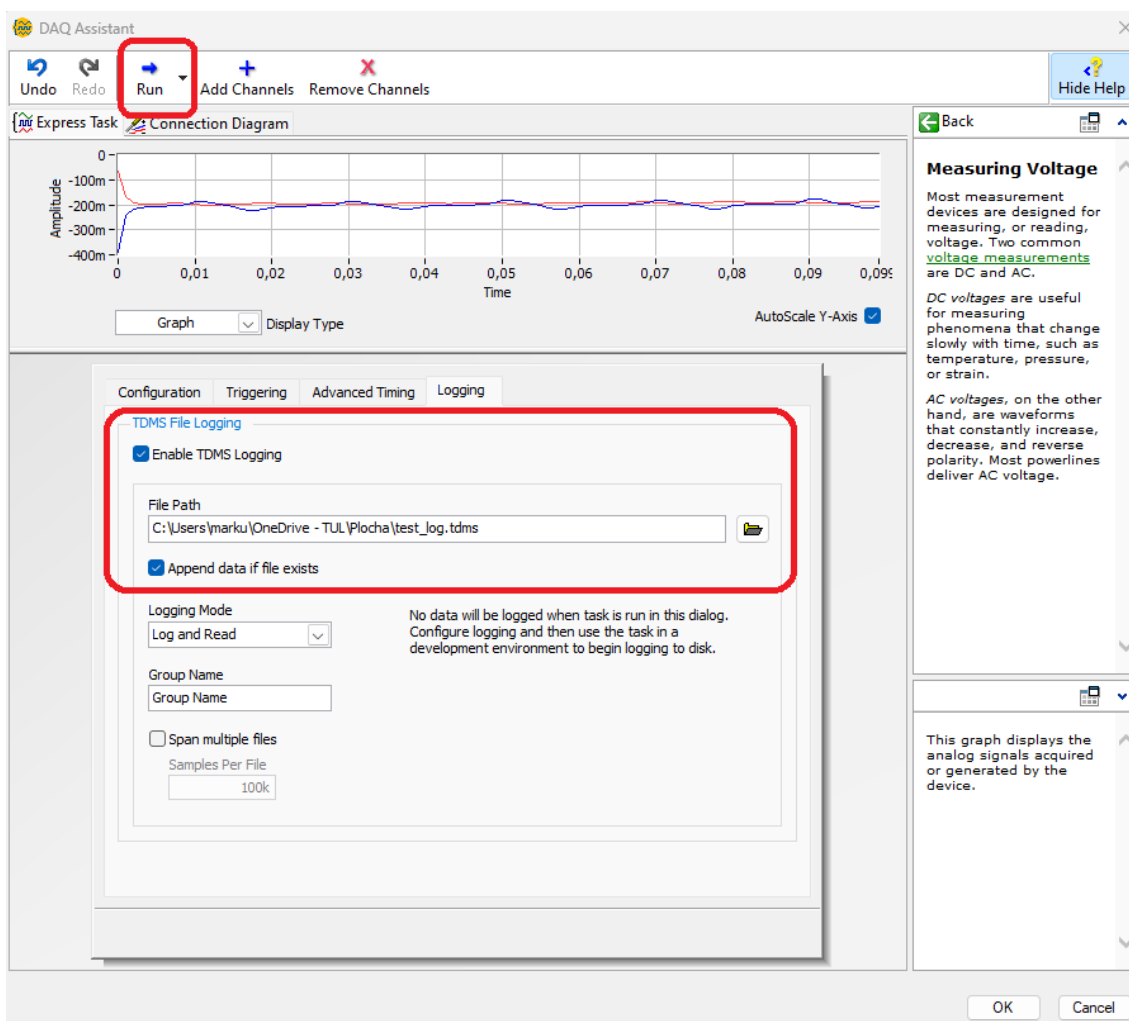


Obrázek 5.16: Hlavní nastavení DAQ Assistant modulu

1 - karta „Connection Diagram“, 2 - nastavení jednotlivých vstupních kanálů, 3 - nastavení časování, 4 - nastavení „loggingu“

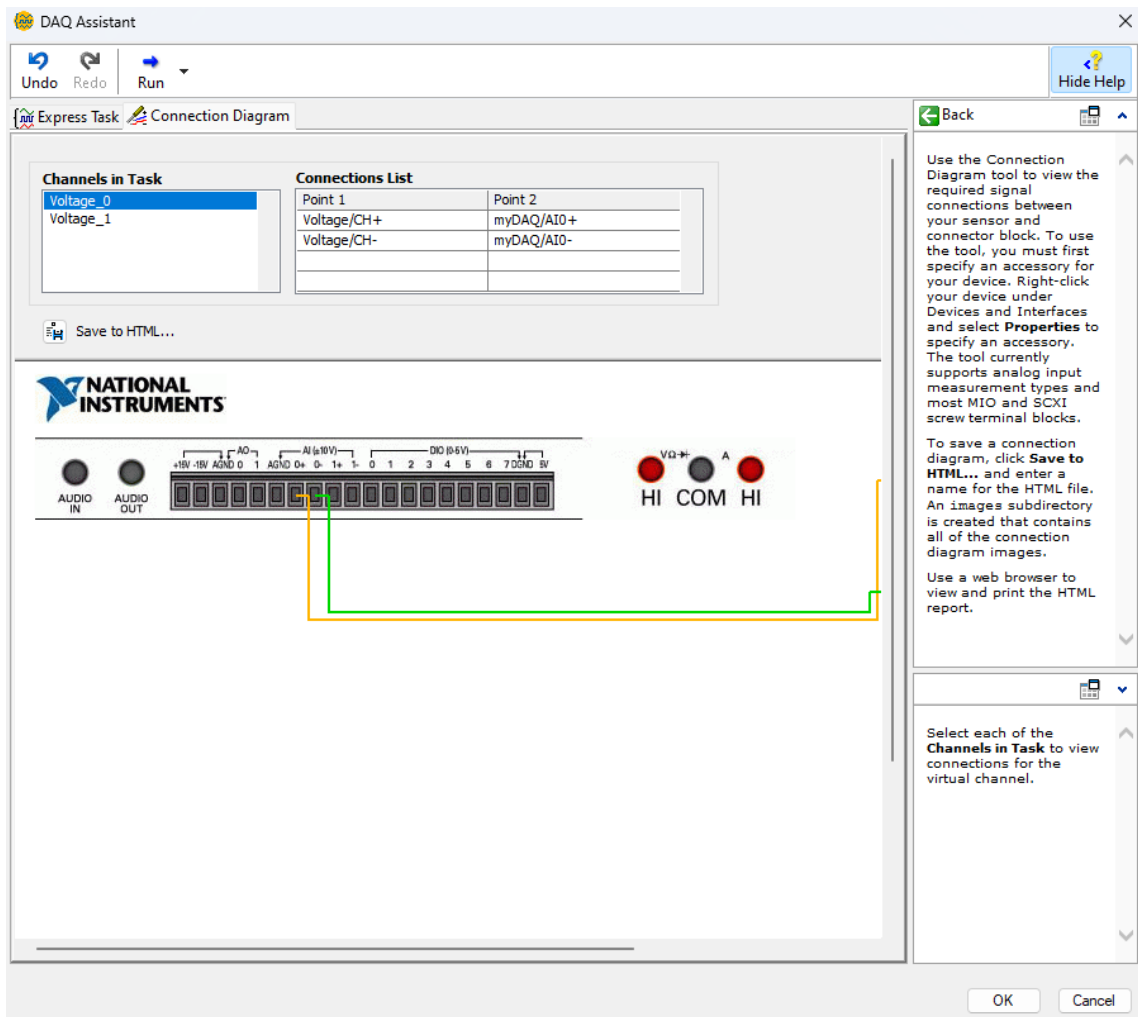
V hlavním okně nastavení DAQ Assistant začneme nastavením vlastností měření jednotlivých kanálů. Ta je v červeně označené oblasti pod číslem 2 na obrázku 5.16. V levé části můžeme přidávat a odebírat měřené vstupní kanály a po jejich výběru lze měnit velikost měřené vstupní veličiny. V pravé dolní části, pokud to zařízení umožňuje, je možné zvolit typ vstupního zapojení. V našem případě se jedná o diferenciální zapojení. V další vyznačené oblasti označené nyní číslem 3, nastavíme typ záznamu. Zde zvolíme „NI Samples“. Více vpravo pak nastavujeme

počet zaznamenaných vzorků a vzorkovací frekvenci.



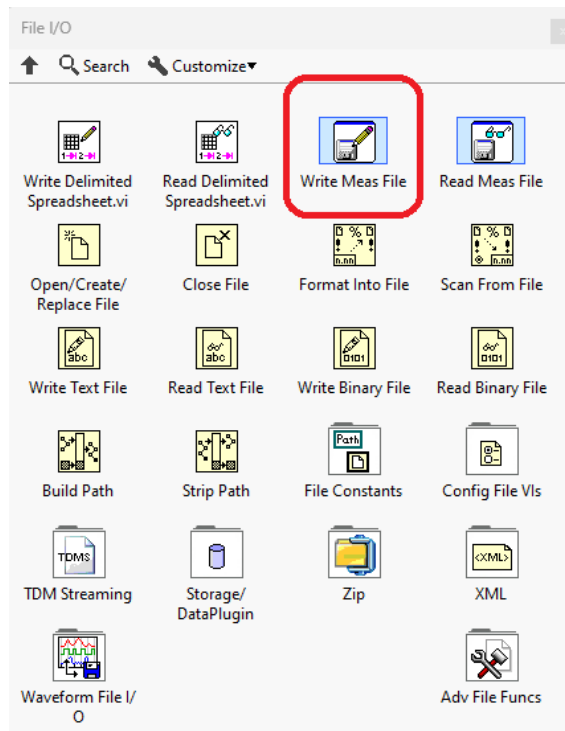
Obrázek 5.17: Záložka pro záznam dat v DAQ Assistant

V posledním kroku můžeme nastavit „Logging (uložení záznamu)“. Toto nastavení provedeme v záložce vyznačené červenou oblastí a číslem 4 na obrázku 5.16. Vzhled této záložky je zachycen na obrázku 5.17. Uprostřed tohoto obrázku se nachází červeně vyznačená oblast. Zde zaškrtneme políčko „Enable TDMS Logging“, následně níže nastavíme cílový adresář a název souboru. Ještě níže pak zaškrtneme políčko „Append data if file exist“. To nám zařídí, že při opakovaném záznamu se soubor nepřepíše, ale data se zaznamenají do jednoho souboru. V horní části je vyznačeno tlačítko „run“. To spustí námi nastavený program a můžeme tak jednoduše vyzkoušet, zda vše správně funguje.



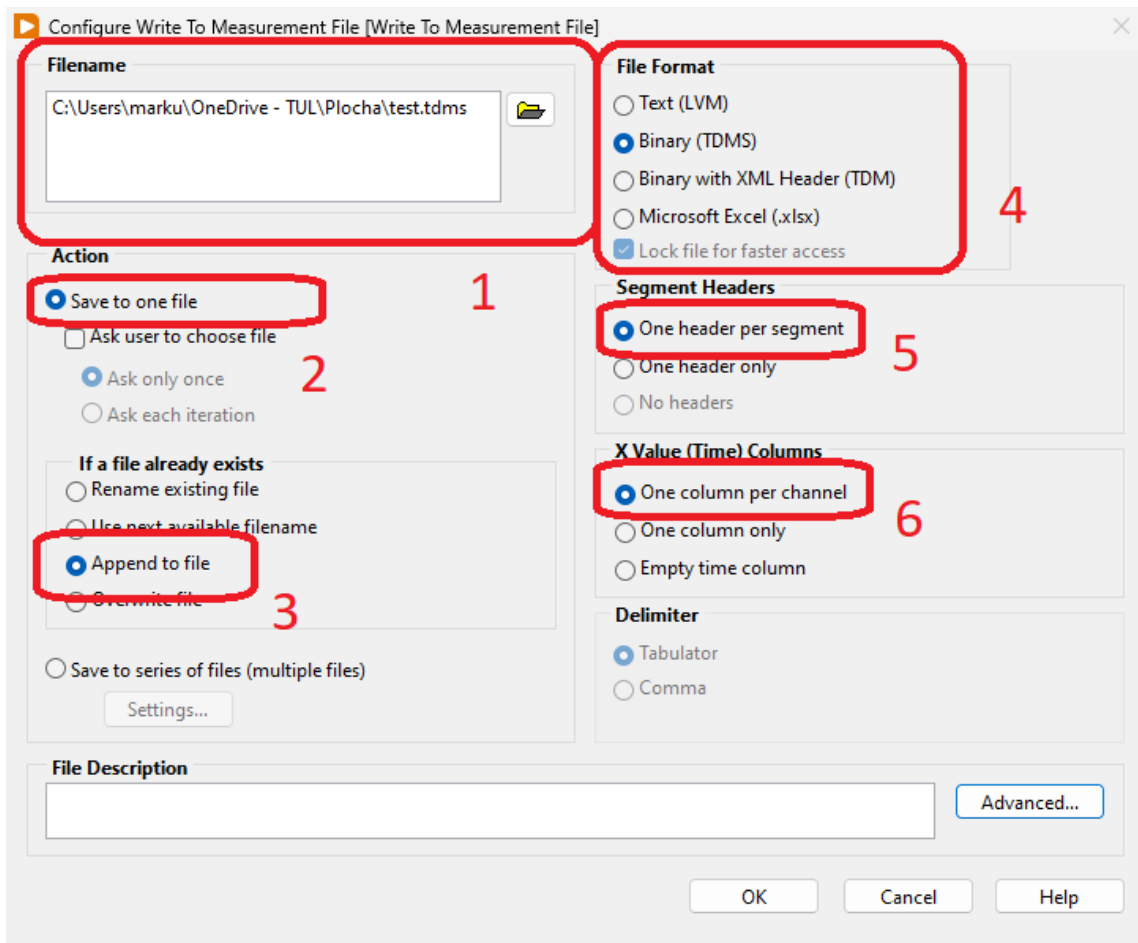
Obrázek 5.18: Karta zapojení pinů pro vybraný vstup

Před zahájením měření může být dobré zkontrolovat, že máme vodiče připojené ke správným pinům. K tomu nám dobře poslouží karta „Connection Diagram“ vyznačená na obrázku 5.16 červeně s popisným číslem 1. Na obrázku 5.18 je pak zobrazeno zapojení pro kanál „ai0“ na zařízení NI myDAQ. Nakonec potvrdíme tlačítkem „Ok“ a systém nám vygeneruje blok DAQ Assistant dle našich nastavených parametrů.



Obrázek 5.19: I/O file kategorie s Write To Measurement File modulem

V dalším kroku nastavíme ukládání naměřených dat a jejich zobrazení. Do blokového diagramu vložíme nový blok z kategorie „File I/O“ viz 5.19. Blok nese název „Write Meas File“. Po vložení se nám otevře okno konfigurace bloku. Jednotlivé kroky jsou označeny číslem a rámečkem na obrázku 5.20. V 1. kroku vybereme cílový adresář a název souboru, do kterého se budou ukládat naměřená data. V 2. kroku zaškrtneme políčko „Save to one file“, to nám zajistí, že data budou ukládána do jednoho souboru a nikoliv separovaně, například každých 1000 vzorků. Dále v kroku 3 zaškrtneme „Append to file“. Tím nastavíme ukládání dat za sebe do jednoho souboru. Ve 4. kroku přichází výběr typu souboru, v jakém se budou data ukládat. My zvolíme „Binary (TDMS)“, který se shoduje se soubory vytvářenými i v jiných postupech. Výhodný, ale může být i soubor „.xls (soubor Excelu)“. Nastavení hlavičky v 5. kroku nastavíme jako „One header per segment“. V posledním 6. kroku zaškrtneme „One column per channel“. Data z jednotlivých kanálů se tak budou ukládat do sloupců vedle sebe. Po dokončení všech kroků potvrdíme tlačítkem „OK“.

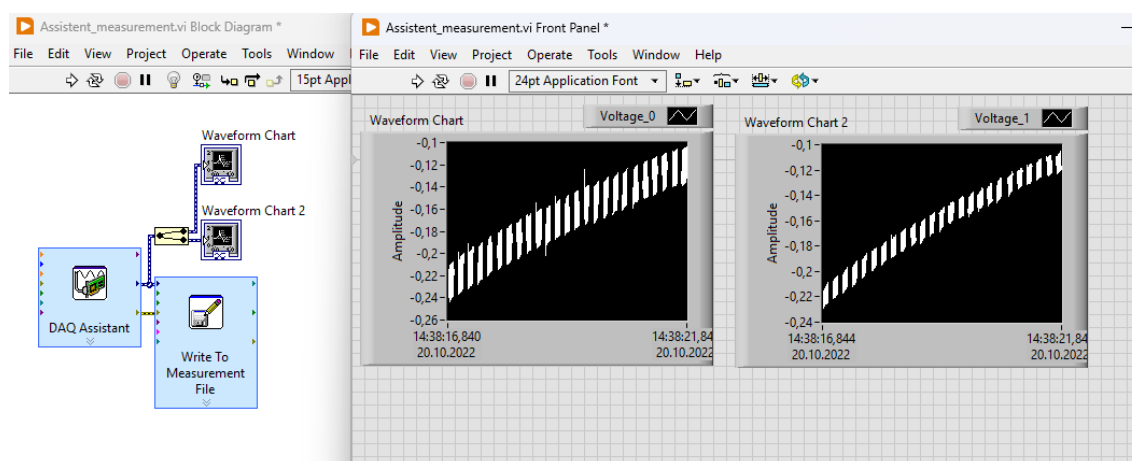


Obrázek 5.20: Nastavení bloku „Write Meas File“

1 - cílová adresa souboru pro „logging“, 2 - nastavení metody ukládání dat „loggingu“, 3 - nastavení způsobu ukládání „loggingu“, 4 - nastavení typu ukládaného souboru, 5 - nastavení hlavičky záznamu, 6 - nastavení ukládání jednotlivých kanálů

Jakmile máme bloky nastavené přejdeme k jejich propojení. Výstup „error out“ na „DAQ Assistant“ propojíme se vstupem „error in“ bloku „Write To Measurement File“. Výstup „data“ na „DAQ Assistant“ propojíme se vstupem „Signals“ bloku „Write To Measurement File“. Pro zobrazení dat, separátně pro každý kanál, přidáme dva bloky „Waveform Chart“ a jeden „Split Signal“ blok. Do „Split Signal“ bloku navedeme výstup „data“ bloku „DAQ Assistant“. Z „Split Signal“ bloku následně navedeme signál do jednotlivých bloků „Waveform Chart“. Celé zapojení je vytvořeno na obrázku 5.21. Pro spuštění programu klikneme na šipku na horním

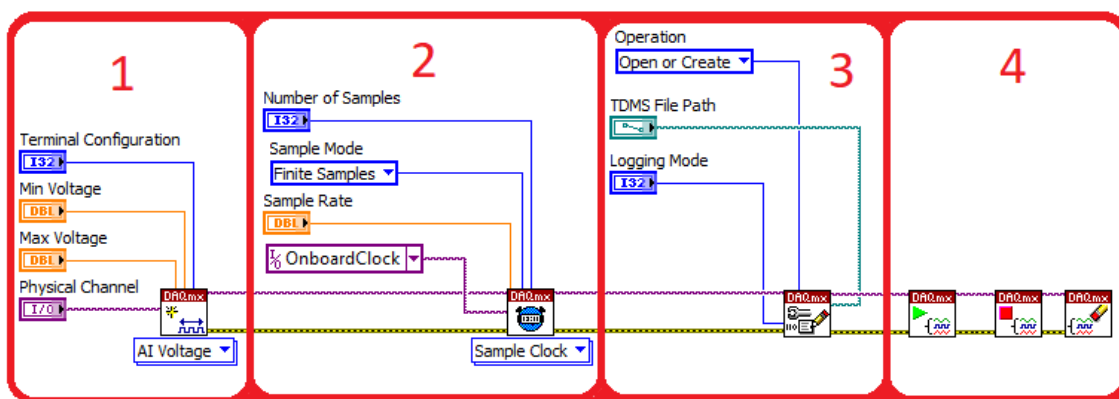
panelu.



Obrázek 5.21: Blokové schéma a přední panel programu využívající blok „DAQ Assistant“

5.3.6 Záznam pomocí jednotlivých bloků

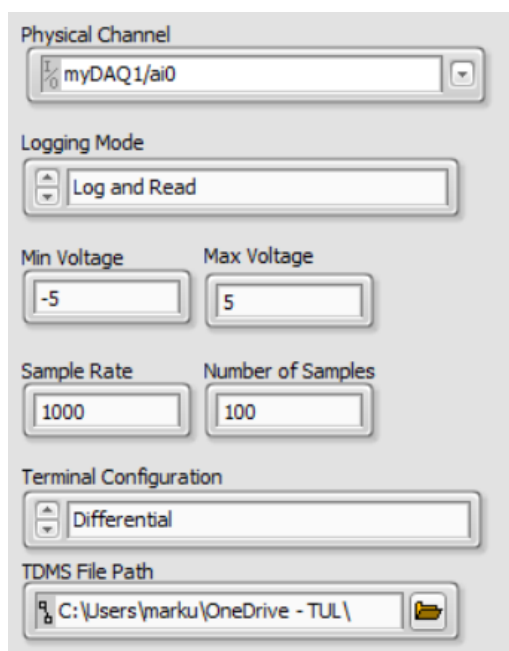
V minulé sekci jsme si ukázaly jak se dají zaznamenat data pomocí blok DAQ Assistant. V této sekci si ukážeme jak se dá tento blok rozdělit a naprogramovat postupně pomocí jednotlivých bloků LabVIEW. Blokové schéma takového programu vypadá následovně 5.22. Pod číslem 1 se nachází nastavení specifického úkol, který se bude měřit. Nastavit tu lze typ terminálu, minimální a maximální hodnotu napětí a z jakého vstupu se budou data zaznamenávat. V části 2. se nastavuje časování. Pro tuto úlohu lze nastavit počet zachycených vzorků i počet vzorků za sekundu tedy vzorkovací frekvenci. Je tu také možnost využít kontinuální měření bez specifikace požadovaného počtu naměřených hodnot. Lze také nastavit zdroj časovače. Ve 3. části se nastavuje soubor a způsob jakým se do souboru budou ukládat. Poslední 4. část pak nejprve spustí úlohu a po jejím dokončení vypne a vyčistí paměť úloh.



Obrázek 5.22: Schéma zapojení pomocí jednotlivých bloků LabVIEW

- 1 - vytvoření úlohy pro záznam, 2 - nastavení časování záznamu, 3 - nastavení ukládání dat, 4 - zapnutí, vypnutí a vyčištění

Součástí je samozřejmě automaticky vytvořený přední panel 5.23, na kterém lze interaktivně nastavovat jednotlivé vstupy.

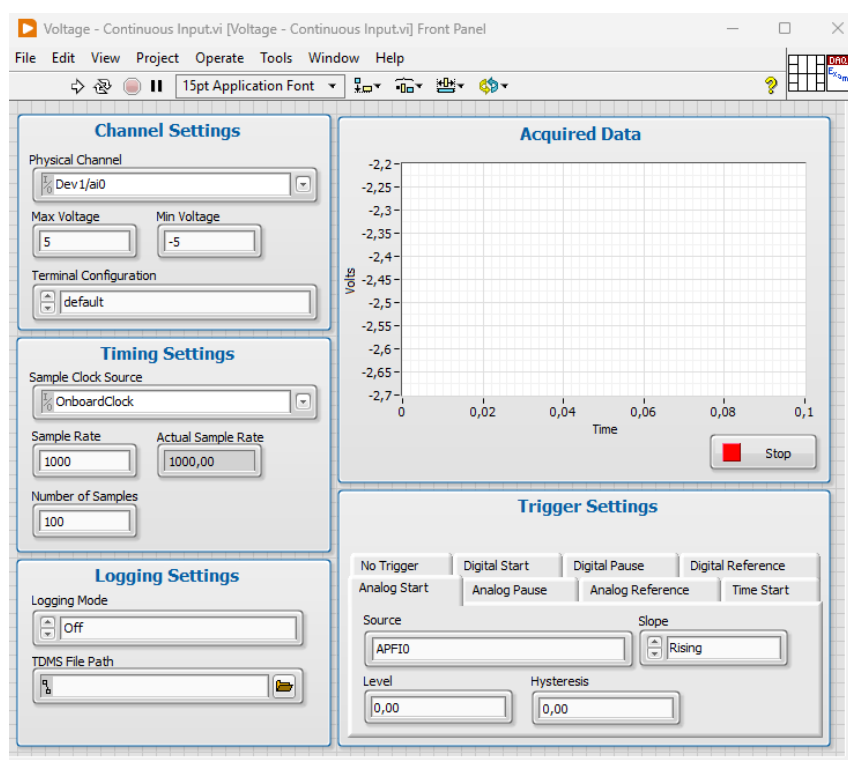


Obrázek 5.23: Přední panel zapojení pro zaznamenání dat pouze pomocí jednotlivých bloků LabVIEW

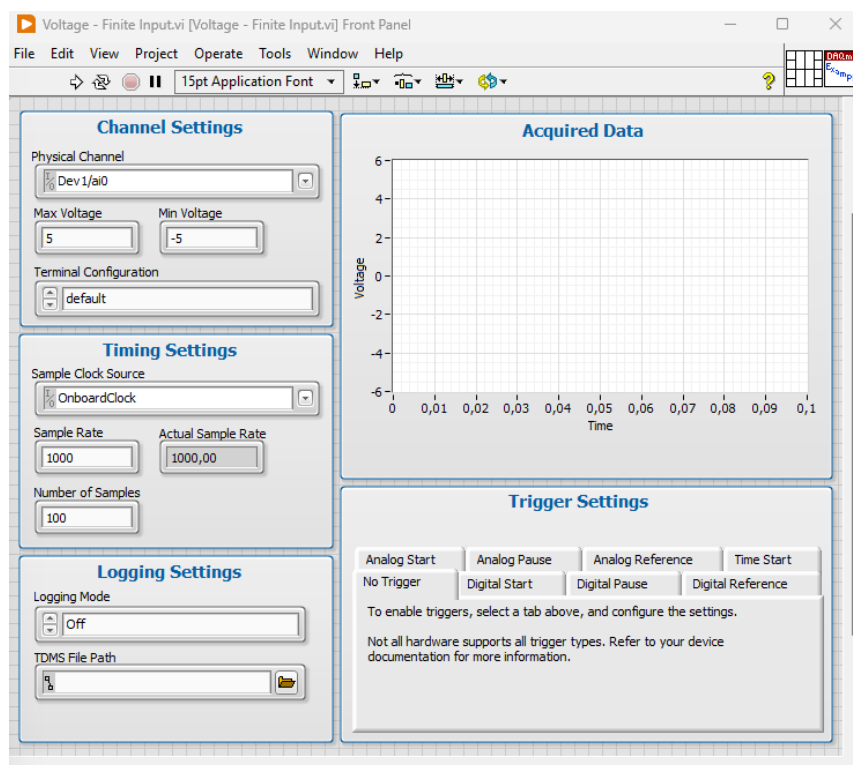
5.3.7 Záznam pomocí ukázkových souborů

Další možností je záznam pomocí souboru sloužících jako příklad pro pochopení principu fungování záznamu dat. Pokud máme nainstalované vše potřebné pro práci s NI DAQ zařízením najdeme tyto příklady ve složce „examples“, která je umístěna v adresáři `\NI\LV Addons\nidaqmx\1\examples`. Jedná se o relativní adresu a závisí tak na umístění instalace. Pro záznam napětí můžeme použít dva téměř totožné příklady. Jedná se o „Voltage - Continuous Input.vi“ a „Voltage - Finite Input“. Na stejném místě nalezneme další příklady měření nebo například filtrování různých fyzikálních veličin měřených pomocí zařízení NI.

Tyto dva příklady měření napětí jsou téměř totožné. První příklad viz 5.24, je kontinuální měření. Na rozdíl od druhého příkladu viz 5.25, má u grafu tlačítko stop sloužící k zastavení měření. Jelikož druhým příkladem je záznam jednoho konečného měření, logicky není tlačítko potřeba.



Obrázek 5.24: Příklad programu s kontinuálním měřením ze složky „examples“



Obrázek 5.25: Příklad programu s konečným měřením ze složky „examples“

Jak měření funguje

Obdobně jako u předem připravených šablon je zde možnost nastavení vstupního kanálu i maximální a minimální hodnoty napětí vstupu. Toto nastavení se nachází v sekci „Channel Settings“. Samozřejmostí je nastavení vzorkovací frekvence a počtu vzorků. To se nachází v nastavení označeném jako „Timing Settings“. Na rozdíl od šablon zde přibyla možnost časování na zařízení, která se nastavuje v nabídce „Sample Clock Source“. Nechybí zde ani „Logging Settings“, umožňující uložení naměřených dat do souboru „.tdms“. Graf naměřených dat je opět interaktivní. Tyto příklady jsou nejlepší volbou, pokud chcete měřit jeden vstup a to ať metodou kontinuálního měření, tak měření konečného. Další výhodou je jednoduchost, a tedy možnost snáze přidat nějaký proces do průběhu měření. Tímto programy z šablon nedisponují.

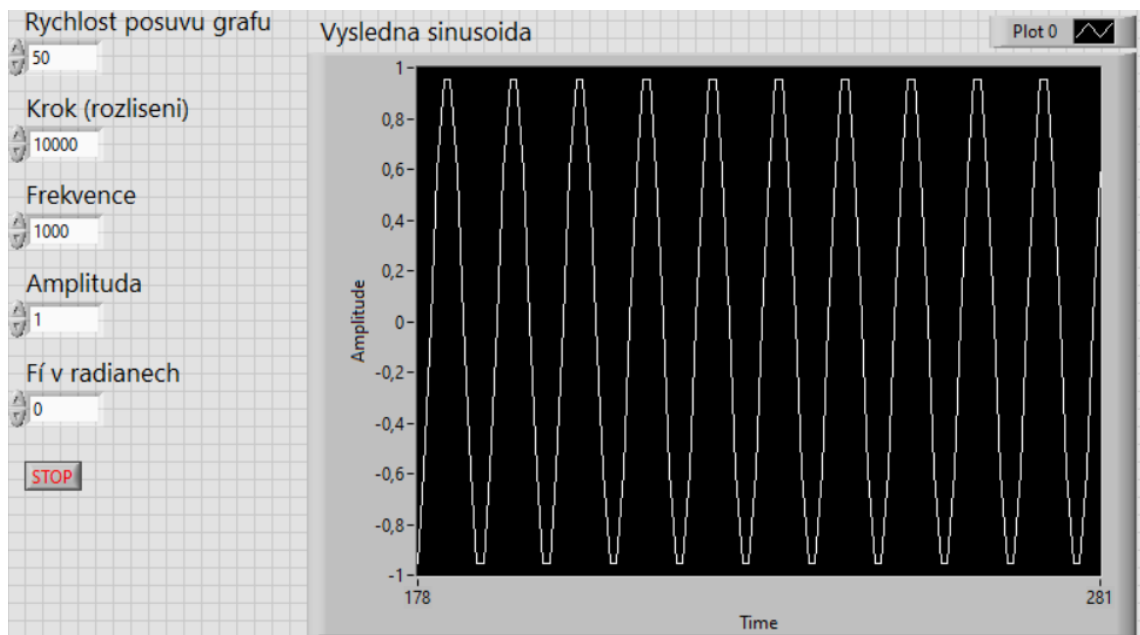
5.4 Použití funkčních bloků Formula Node a Matlab script

Složitější matematické výpočty by v blokovém programování LabVIEW mohly dosahovat značných rozměrů a složitosti. Tento problém řeší možnost využití bloku Formula Node.

Jedná se o blok, na kterém si vytváříme vstupy a výstupy jakožto proměnné. Přidáme je kliknutím na rám bloku a výběrem „add input“ nebo „add output“ je přidáme. Názvy těchto vstupů a výstupů odpovídají názvu proměnných uvnitř bloku. Do bloku můžeme psát rovnice s použitím obvyklé matematické syntaxe. Podrobnější popis bloku a jeho vlastností lze nalézt v dokumentaci [13]. V bloku nelze tvořit proměnné, ale je možno využít i řadu matematických funkcí typu sin, cos, abs a podobně.

V případě, že máme nějakou složitější matematickou formuli, například nějaký filtr. Můžeme využít i blok MATLAB script. Tento blok nám umožňuje volat funkce nebo skripty vytvořené pro jazyk MATLAB. Aby bylo možné tyto skripty a funkce volat je nutné přidat cestu k souboru do zdrojových složek MATLABu. Po spuštění souboru LabVIEW, který obsahuje tento blok se spustí i příkazové okno MATLABu. Pokud toto okno zavře program nebude funkční. Po vytvoření programu může být nutné LabVIEW restartovat to je způsobeno propojením ActiveX.

Pro demonstraci těchto bloků jsme vytvořily jednoduchý generátor sinusových vln. Programy v LabVIEW se skládají ze dvou částí a to čelního panelu a blokového schéma. Z čelního panelu 5.26 lze vyčíst, že generátor má 6 různých vstupů a jeden výstup. Výstupem je graf, který zobrazuje výslednou sinusoidu. Mezi vstupy patří rychlost posuvu grafu na ose x, krok který udává vzorkovací frekvenci, frekvence výsledné sinusoidy, amplituda sinusového signálu a fázový posuv ϕ , který se udává v radiánech. Šestá vstupem je tlačítko „STOP“, které ukončí program.



Obrázek 5.26: Čelní panel generátoru sinusových vln

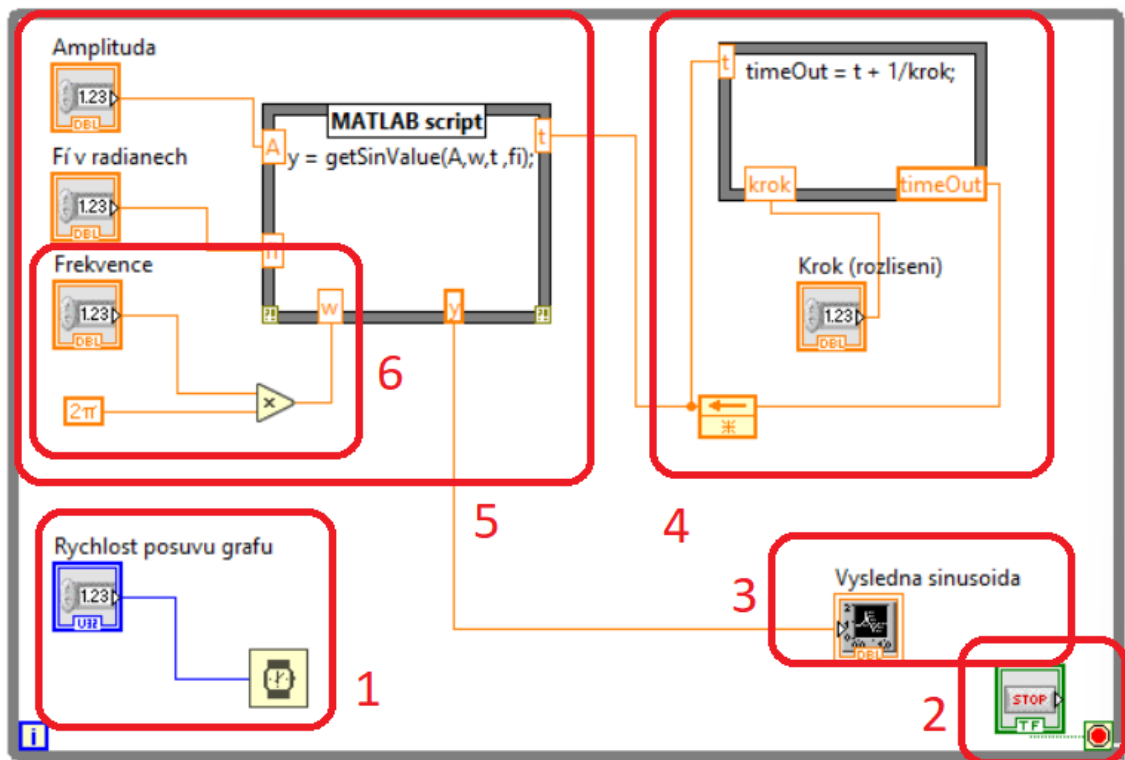
Čelní panel funguje jako GUI, to jak program pracuje se vstupy a výstupy se tvoří v blokovém schéma. Pro náš generátor pak vypadá takto 5.27.

Celý program tvoří jedna smyčka while. Ukončení této smyčky pak umožňuje blok „STOP“ označený číslem 2. Výstup této smyčky je blok „Waveform Chart“ označený číslem 3, který se aktualizuje v každém cyklu smyčky. V každé smyčce je pak nastavené zdržení pomocí bloků v části 1.

Nyní přejdeme k samotnému výpočtu hodnoty y pro aktuální x na grafu. V části 4 je využit blok Formula Node [13]. Ten využívá vstupu „ t “ a „krok“, jeho výstupem je pak „timeOut“. Proměnná krok udává velikost posuvu na ose x , takže zároveň představuje vzorkovací frekvenci. Výsledná hodnota „timeOut“ je definována uvnitř bloku, kde se sčítá aktuální hodnota na ose x a převrácená hodnota kroku (čím vyšší hodnota tím menší posun).

V části 5 se provádí již vlastní hodnoty sinusoidy. Pro její výpočet je použita vlastní funkce v jazyce MATLAB. Ta obsahuje základní předpis pro sinusoidu $y(t) = A * \sin(2 * \pi * f * t + \varphi)$ kde $2 * \pi * f = \omega$ a vrací její hodnotu. Pro její výpočet je potřeba obdobně jako u bloku Formula Node přidat vstupní a výstupní

proměnné se shodnými názvy jako uvnitř bloku. Proměnná „w“ zde představuje ω a je pro ukázkou vypočítána standardním způsobem používaným v LabVIEW a to pomocí aritmetického bloku a konstanty 2π pro upřesnění je tento výpočet označen číslem 6.



Obrázek 5.27: Blokové schéma generátor sinusových vln

1 - nastavení časování posuvu, 2 - nastavení vyskočení z cyklu while, 3 - blok pro uživatelský výstup do grafu, 4 - krokování sinusoidy, 5 - výpočet vlastních hodnot sinusoidy, 6 - definice frekvence pomocí LabVIEW bloků

5.5 Praktická ukáзка měření biomedicínských data za pomocí LabVIEW a MATLABu

5.5.1 Základní informace o použitých zařízeních a jejich nastavení

NI myDAQ Zařízení NI myDAQ je řešeno jako zařízení pro edukační účely. Konektivita je tedy za tímto účelem také koncipována. Zařízení umožňuje audio vstup

i výstup v podobě konektoru 3,5mm stereo jacku. Analogový vstup i výstup, konkrétně 2 diferenciální. Digitální vstupy i výstupy, specificky 8 pinů. Na zařízení je i konektivita pro časovač či počítání cyklů. Na přední části jsou umístěny konektory pro digitální multimetr. Multimetr bohužel nemá přímou podporu knihoven MATLABu a přístup je tedy možný buď skrze externí knihovnu, kterou lze nalézt na portálu MathWorks v kategorii „File Exchange“ [8] nebo pomocí LabVIEW. S počítačem může modul komunikovat pomocí standardu USB-A. V tabulce 5.1 jsou vybrány nejdůležitější informace potřebné k ukázkovým měřením.

Tabulka 5.1: Tabulka základních specifikací zařízení NI myDAQ

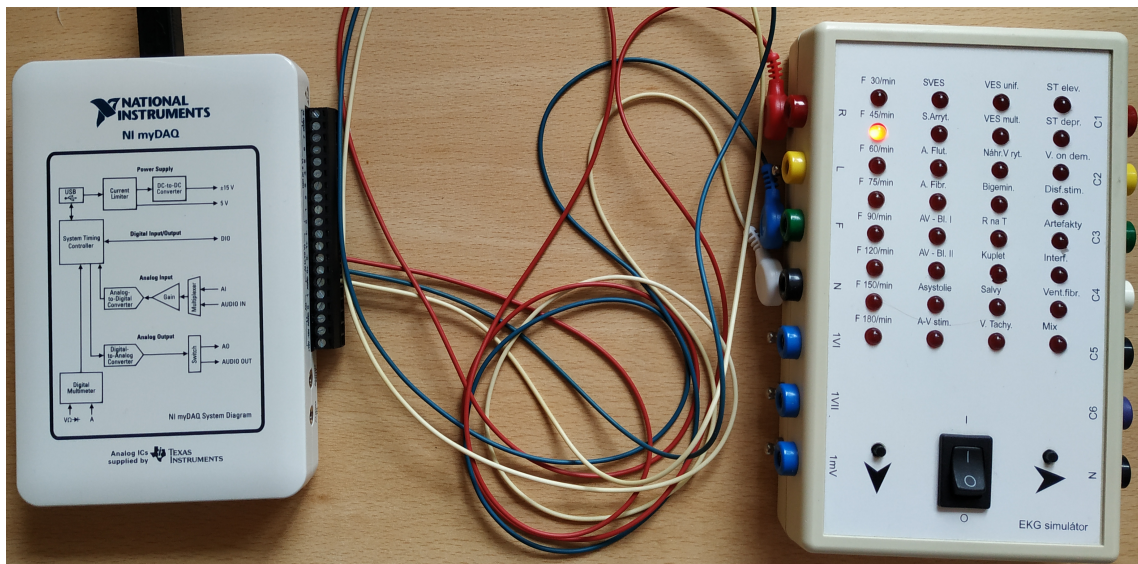
Analogové vstupy	
Počet kanálů	2 diferenciální a 1 stereo zvukový vstup
ADC rozlišení	16 bitů
Maximální počet vzorků za sekundu	200 kS/s
Měřitelný napěťový rozsah	
Analogový vstup	+/-10 V nebo +/-2 V DC-couplet
Audio vstup	+/-2V AC-coupled
Měřitelný frekvenční rozsah	
Analogový vstup	DC až 400 kHz
Audio vstup	1.5 až 400 kHz
Konektory	
Analogové vstupy	svorkovnice
Audio vstupy	3.5 mm stereo jack

Digitální stetoskop Použitý digitální stetoskop je zařízení od firmy ThinkLabs One. Zařízení je jednoduché na ovládání a nastavení. Tento přístroj disponuje nastavením výstupní hlasitosti signálu. Zesílený audio signál lze zachytit na výstupu, kterým je 3,5mm stereo jack. Konektor slouží zároveň i k nabíjení zařízení. Z toho je patrné, že zařízení je napájeno baterií, tudíž nepotřebuje být při vyšetření zapojeno do sítě. Zařízení neprovádí pouze reprezentaci akustického vlnění v podobě napě-

řového výstupu, ale umožňuje také základní filtraci. Pro účely měření bylo zařízení nastaveno na rozsah 60 – 500Hz. Jsou tedy potlačeny nízké a vysoké frekvence. Toto nastavení je doporučeno také v manuálu, který je přiložen k zařízení, ale je také dostupný online. Výstupní hlasitost byla dána na maximální hodnotu z rozsahu 1 – 10 [2].

5.5.2 Naměření dat EKG

Pro zaznamenání signálu EKG využijeme dvě zařízení 5.28. Měřící zařízení NI myDAQ 5.5.1 a simulátor EKG signálu, který nám bude konstantně generovat signál podobný lidskému srdci. Využití EKG simulátoru je výhodné pro demonstraci, jelikož se zde eliminuje rušení vzniklé pohybem člověka, tedy jeho svaly.



Obrázek 5.28: Měřící sestava pro záznam EKG

Pro záznam EKG signálu použijeme programovací jazyk a prostředí MATLAB. Zde použijeme následující kód.

Nejprve vyčistíme pracovní prostor MATLABu příkazem „clear“.

```
1 clear;
```

Abychom mohly vytvořit záznam, potřebujeme vytvořit objekt „daq“, který je z „toolboxu Data Acquisition Toolbox“.

```
1 d = daq("ni");
```

Dále přidáme námi vybraný vstupní kanál, definujeme záznamové zařízení a veličinu, kterou budeme zaznamenávat.

```
1 ch1 = addinput(d, "myDAQ1", "ai0", "Voltage");
```

Nyní můžeme nastavit parametry měření. Jelikož měříme nižší hodnoty napětí, zvolíme rozsah měřeného napětí na ± 2 Volty. Druhou možností by byla volba rozsahu ± 10 Voltů. Zde by, ale docházelo ke zkreslení signálů kvůli velkému kvantizačnímu kroku. Vzorkovací frekvenci zvolíme vzhledem k typu zaznamenávaného signálu na hodnotu 256Hz.

```
1 ch1.Range = [-2 2];  
2 d.Rate = 256;
```

Když máme nastaveny všechny potřebné parametry. Provedeme tedy záznam ze zařízení. V následujícím příkazu nastavíme čas a počkáme na provedení záznamu.

```
1 myDAQ1_2 = read(d, seconds(5));
```

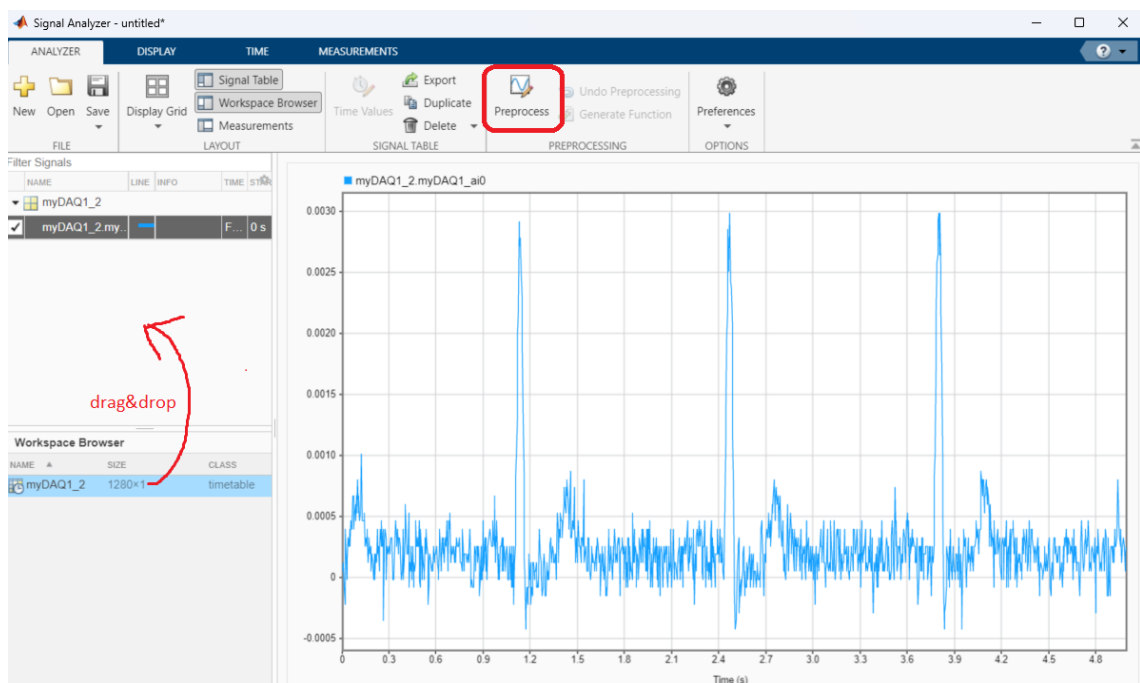
Výsledná data můžeme jednoduše zobrazit pomocí následujícího kódu.

```
1 plot(myDAQ1_2.Time, myDAQ1_2.Variables);  
2 xlabel("Time (s)");  
3 ylabel("Amplitude");  
4 legend(myDAQ1_2.Properties.VariableNames, ...  
        "Interpreter", "none");
```

Pro extrahování dat z objektu do proměnné „a“ pak použijeme tento kód.

```
1         a = myDAQ1_2.Variables;  
2         clear d ch1
```

Jak vidíme ze záznamu, viz 5.29 jsou jasně vidět R a T vlny. Ostatní části jsou zašuměné. Zde přichází velká výhoda MATLABu a tou jsou knihovny pro zpracování signálu. Otevřeme tedy aplikaci „Signal Analyzer“. Nahrajeme záznam a otevřeme nabídku „Preprocess“. Vše je také možné dělat i pomocí funkcí.



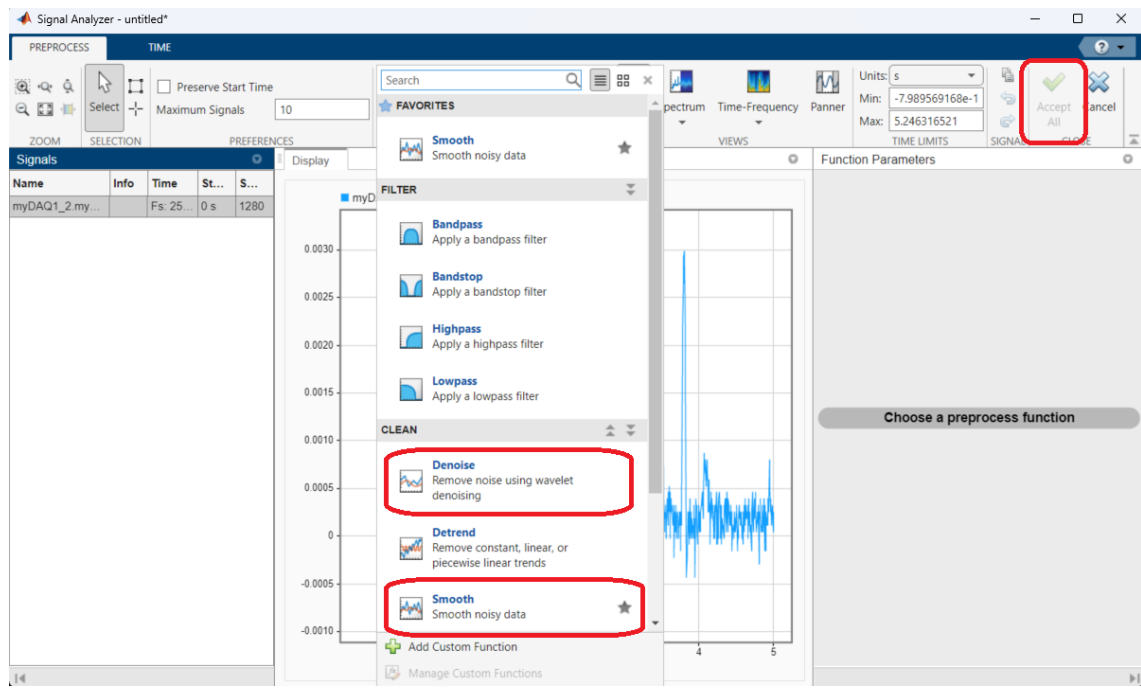
Obrázek 5.29: Záznam EKG zobrazený v aplikaci „Signal Designer“

V nabídce „Preprocess“ zvolíme postupně nejprve filtr „Denoise“, aplikujeme filtr na data a následně totéž provedeme i s filtrem „Smooth“. V obou případech ponecháme původní nastavení těchto filtrů. První filtr odstraní šum ze signálu, druhý následně signál vyhladí.

Pro používání filtrů je samozřejmostí znalost toho jak fungují a na jaká data je vhodné je aplikovat. Pro naši demonstraci to však není podstatné. Jednoduchým

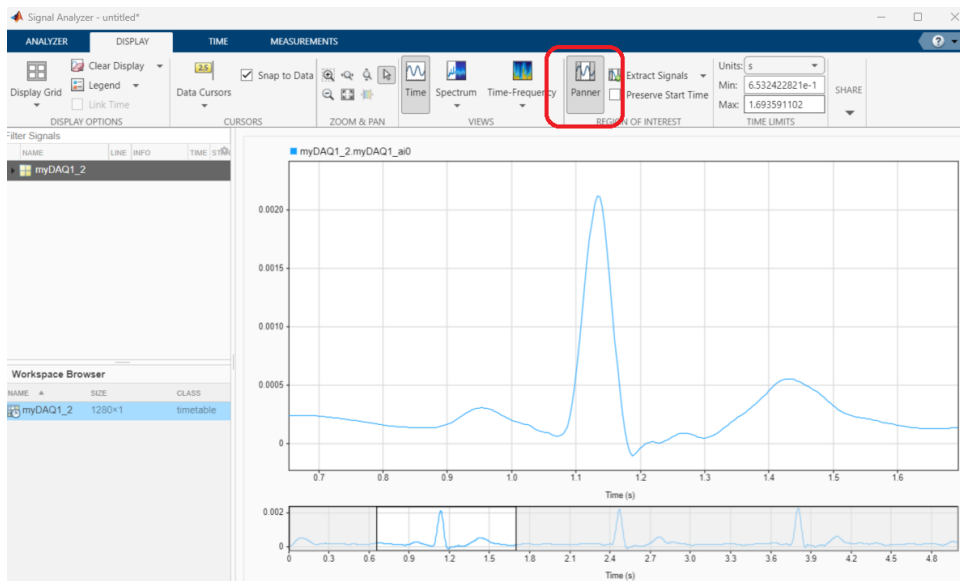
popisem je filtr „Smooth“ plovoucí průměrovací okno a filtr „Denoise“ využívá Bayesův filtr.

Po aplikaci těchto filtrů klikneme na tlačítko „Accept all“ a vrátíme se tak do předešlé části.



Obrázek 5.30: Použití filtrů v nastavení „Preprocess“

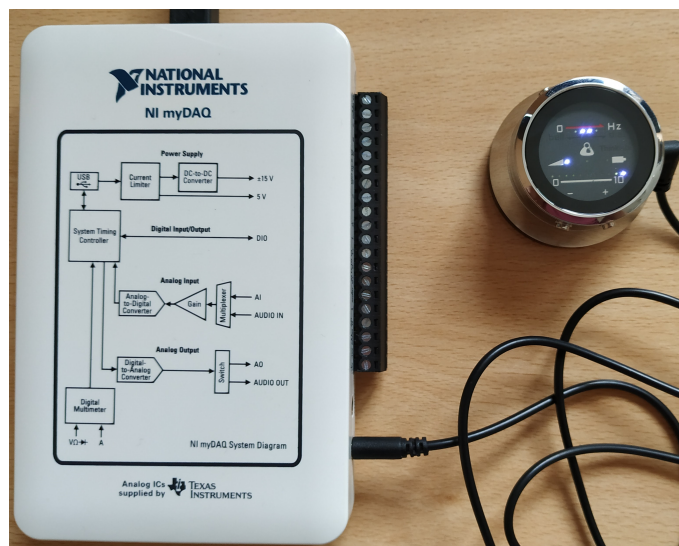
Pro zhodnocení výsledku je užitečná funkce „Panner“, která nám umožňuje přiblížit specifickou část signálu. Toho využijeme a přiblížíme si první R vlnu našeho signálu. Nyní jsou bez problému čitelné všechny podstatné části EKG signálu, viz 5.31.



Obrázek 5.31: Využití funkce „Panner“, přiblížení výsledného signálu

5.5.3 Naměření zvuku z digitálního stetoskopu

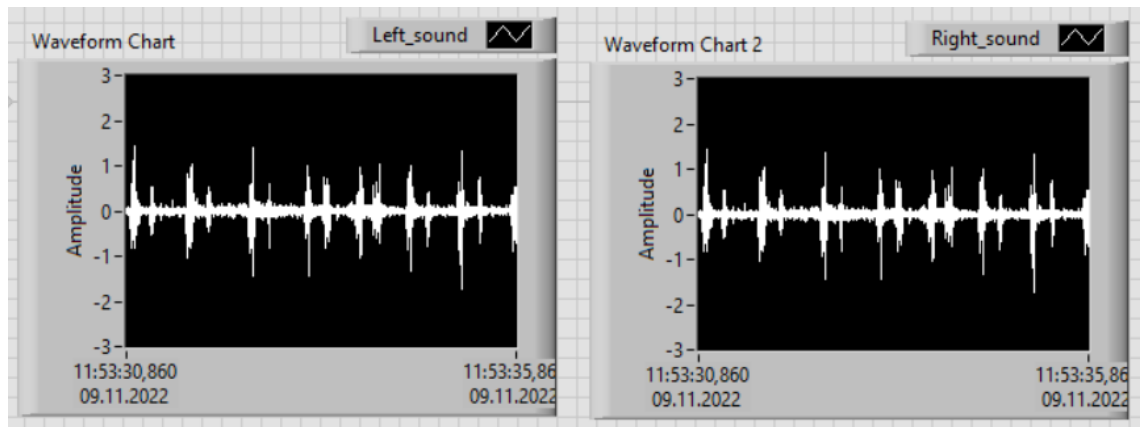
Pro záznam srdečních ozvů potřebujeme zařízení myDAQ a digitální stetoskop. Sestavu je možné vidět na obrázku 5.32 a detaily o zařízení pak nalezneme v sekci 5.5.1. Pro záznam použijeme metodu v sekci 5.3.5.



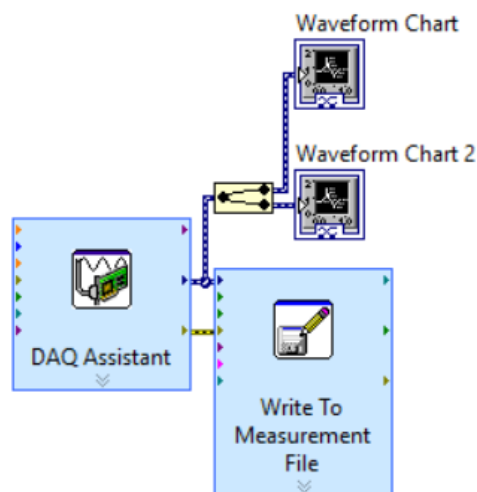
Obrázek 5.32: Měřící sestava pro záznam srdečních ozvů

Při nastavování vzorkovací frekvence zvolíme hodnotu 1000Hz. Výstup z měření

uložíme do proměnné „test.tdms“. Výsledek našeho měření vypadá na čelním panelu takto 5.33 a blokové schéma takto 5.34.



Obrázek 5.33: Čelní panel programu pro záznam audio signálu z digitálního stetoskopu



Obrázek 5.34: Blokové schéma použité pro záznam audio signálu z digitálního stetoskopu

Když máme data zaznamenána můžeme využít funkcí v MATLABu a dále signál zpracovat. Nejprve bude nutné extrahovat data z „tdms“ souboru, který jsme si vytvořili. To provedeme pomocí postupu v sekci 5.1.2. Takto extrahovaná data si připravíme do proměnné „data“.

S takto připravenými daty můžeme zacházet dle potřeby. V následujících řádcích si, ukážeme některé užitečné funkce pro zpracování audio signálu.

První užitečnou funkcí je funkce `sound()`. Tato funkce umožňuje přehrát audi-záznam na připojených reproduktorech. První argument funkce jsou zaznamenaná audio data. Druhý argument je vzorkovací frekvence, v jaké byl záznam pořízen.

```
1          sound(data{1,3}.Right_Sound,1000);
```

Další funkcí je funkce `audiowrite()`, která uloží záznam do běžného audio souboru. Funkce přebírá tři argumenty. První je název souboru, i s koncovkou, do kterého budeme ukládat nahraný zvuk. Třetím je vzorkovací frekvence záznamu.

```
1          audiowrite("zaznam.wav", data{1,3}.Right_Sound, 1000)
```

Zde si ukážeme další výhodu MATLABu a to prototypování. Jako ukázkou využijeme jednoduchý algoritmus, který nám spočítá srdeční frekvenci.

V prvním kroku si uložíme do proměnné „d“ pouze data, která budeme potřebovat, což jsou data z pravého kanálu. Můžeme použít i levý, na výsledku se nic nemění.

```
1          d = data{1,3}.Right_sound;
```

Jako první krok si zobrazíme originální data pomocí funkce `plot()` na prvním řádku. Dále použijeme funkci `abs()`. Ta provede absolutní hodnotu nad našimi daty. Získáme tak více bodů reprezentujících píky. Bylo by možné funkci nepoužít a data ponechat, ale následný algoritmus by se musel taktéž přizpůsobit a byl by o něco složitější. Na třetím řádku použijeme na data průměrovací plovoucí okno. Použité okno zahrnuje 25 prvků, což je vzhledem k datům poměrně hodně. Čtvrtý řádek očistí data, respektive vynuluje od hodnot nižších nežli 0,3. Tato hodnota byla zvolena na základě vizuálního srovnání píků vůči ostatním hodnotám.

```

1         plot(d)
2         d = abs(d);
3         d = movmean(d, 25);
4         d(d<0.3) = 0;

```

Nakonec aplikujeme tento algoritmus, který spočítá všechny úseky, jejichž data se pohybují nad hladinou 0,1. To je patrné z grafu, který se vytvoří podle předpisu na prvním řádku.

```

1         plot(d)
2         peakCounter = 0;
3         isPeriod = false;
4         for c = 1:size(d,1)
5             if d(c) > 0.1 && ~isPeriod
6                 isPeriod = true;
7                 peakCounter = peakCounter+1;
8             end
9
10            if d(c) < 0.1 && isPeriod
11                isPeriod = false;
12            end
13        end

```

Po aplikaci algoritmu můžeme spočítat srdeční frekvenci podle jednoduchého vzorce. A to tak, že počet píků vydělíme rozsahem záznamu v sekundách (1000 zde představuje vzorkovací frekvenci) a nakonec vynásobením 60 pro přepočítání na počet úderů za minutu.

```

1         heardBeat = (peakCounter / (size(d,1) / 1000))*60;

```

6 Závěr

V této práci jsme si dali za cíl vytvořit a popsat propojení mezi LabVIEW a MATLABem, ale i mezi MATLABem a LabVIEW, tedy obě varianty. Dalším cílem pak bylo popsání bloků LabVIEW pro záznam dat pomocí NI USB modulu. Ten byl využit pro všechny ukázky.

K práci s NI USB modulem v LabVIEW jsme využili toolboxu „Data Acquisition Toolbox“, který nám umožňuje pracovat s modulem pouze s nainstalovanými ovladači pro daný modul. Odpadá tedy nutnost vlastnit zároveň LabVIEW. Druhou možností je spouštění skriptů pomocí propojení umožněné frameworkem ActiveX, který zároveň poskytuje základní ovládání.

Pro splnění požadavku použít kód MATLABu při běhu LabVIEW nám pak slouží blok „Matlab Script“, který umožňuje volat funkce a skripty přímo za běhu programu LabVIEW. To však nese potřebu mít nainstalovány obě prostředí.

Z NI modulů je možné získávat data i standardním programováním v LabVIEW. Vybrané způsoby jak tyto bloky a programy použít jsou v práci podrobně popsány.

Práce je uzpůsobena k použití i jako návod na propojení oněch zmíněných prostředí.

Literatura

- [1] KOVÁŘ, B. et al. 2007. *Jemný úvod do MATLABu a Simulinku*. [Praha]: FD ČVUT. FD ČVUT. Dostupné také z: https://www.fd.cvut.cz/department/k611/pedagog/K611THO_soubory/Uvod_do_MATLABu.pdf
- [2] THINKLABS ONE. 2022. *User Manual TLL102-31E*. [Denver]: Thinklabs ONE. Thinklabs ONE. Dostupné také z: https://www.thinklabs.com/_files/ugd/51509e_126cda33a7324007b84f157e12d30367.pdf
- [3] MATHWORKS. 2022. About MathWorks. MATHWORKS. *Mathworks* [online]. [Natick]: MathWorks, 2022 [cit. 2022-12-1]. Dostupné z: https://uk.mathworks.com/company.html?s_tid=hp_ff_a_company
- [4] MATHWORKS. 2022. Help Center: MATLAB. MATHWORKS. *Mathworks* [online]. [Natick]: MathWorks, 2022 [cit. 2022-12-1]. Dostupné z: <https://uk.mathworks.com/help/matlab/>
- [5] MATHWORKS. 2022. Products: MATLAB. MATHWORKS. *Mathworks* [online]. [Natick]: MathWorks, 2022 [cit. 2022-12-1]. Dostupné z: <https://uk.mathworks.com/products/matlab.html>
- [6] MATHWORKS. 2022. Products: Data Acquisition Toolbox. MATHWORKS. *Mathworks* [online]. [Natick]: MathWorks, 2022 [cit. 2022-12-1]. Dostupné z: <https://uk.mathworks.com/products/data-acquisition.html>
- [7] MATHWORKS. 2022. Third-Party Products & Services: Data Acquisition Toolbox. MATHWORKS. *Mathworks* [online]. [Natick]: MathWorks, 2022 [cit.

- 2022-12-1]. Dostupné z: https://uk.mathworks.com/products/connections/product_detail/labview.html
- [8] MATHWORKS. 2022. File Exchange: Read DMM on a National Instruments DAQ device in MATLAB. MATHWORKS. *Mathworks* [online]. [Natick]: MathWorks, 2022 [cit. 2022-12-1]. Dostupné z: <https://uk.mathworks.com/matlabcentral/fileexchange/40900-read-dmm-on-a-national-instruments-daq-device-in-matlab>
- [9] NI. 2022. Shop: LabVIEW. NI. *NI* [online]. [Austin]: NI, [cit. 2022-12-1]. Dostupné z: <https://www.ni.com/cs-cz/shop/labview.html>
- [10] NI. 2022. Software products: MyDAQ Software Suite. NI. *NI* [online]. [Austin]: NI, [cit. 2022-12-1]. Dostupné z: <https://www.ni.com/cs-cz/support/downloads/software-products/download.mydaq-software-suite.html#305519>
- [11] NI. 2022. Drivers: NI-DAQmx. NI. *NI* [online]. [Austin]: NI, [cit. 2022-12-1]. Dostupné z: <https://www.ni.com/cs-cz/support/downloads/drivers/download.ni-daqmx.html#464560>
- [12] NI. 2022. LabVIEW: LabVIEW Documentation. NI. *NI* [online]. [Austin]: NI, [cit. 2022-12-1]. Dostupné z: https://www.ni.com/docs/en-US/bundle/labview/page/lvhelp/labview_help.html
- [13] NI. 2022. LabVIEW: Formula Node. NI. *NI* [online]. [Austin]: NI, [cit. 2022-12-1]. Dostupné z: https://www.ni.com/docs/en-US/bundle/labview/page/glang/formula_node.html
- [14] WIKIPEDIA. 2022. National Instruments. WIKIPEDIA. *Wikipedia* [online]. [San Francisco]: Wikipedia, aktualiz. 2022-09-15 [cit. 2022-8-23]. Dostupné z: https://en.wikipedia.org/wiki/National_Instruments
- [15] ŽÍDEK, Jan. 2002. *GRAFICKÉ PROGRAMOVÁNÍ VE VÝVOJOVÉM PROSTŘEDÍ LabVIEW*. [Ostrava]: VŠB-TU Os-

trava. Docplayer. Dostupné také z: <https://docplayer.cz/10343372-Graficke-programovani-ve-vyvojevem-prostredi-LabVIEW.html>

- [16] SMITH, Steven. 1998. Chapter 3: ADC and DAC: The Sampling Theorem. STEVEN W. SMITH. *The Scientist and Engineer's Guide to Digital Signal Processing* [online]. [San Diego]: California Technical Publishing, [cit. 2022-12-1]. Dostupné z: <http://www.dspguide.com/ch3/2.htm>
- [17] SMITH, Steven. 1998. Chapter 3: ADC and DAC: Quantization . STEVEN W. SMITH. *The Scientist and Engineer's Guide to Digital Signal Processing* [online]. [San Diego]: California Technical Publishing, [cit. 2022-12-1]. Dostupné z: <http://www.dspguide.com/ch3/1.htm>
- [18] SMITH, Steven. 1998. [online obrázek]. In: SMITH, Steven. 1998. Chapter 3: ADC and DAC: The Sampling Theorem. STEVEN W. SMITH. *The Scientist and Engineer's Guide to Digital Signal Processing* [online]. [San Diego]: California Technical Publishing, [cit. 2022-12-1]. Dostupné z: <http://www.dspguide.com/ch3/2.htm>