

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# 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

## NÁSTROJ PRO VIZUALIZACI A POSLECH VIBRAČNÍCH SIGNÁLŮ

TOOL FOR VISUALIZATION AND LISTENING OF VIBRATION SIGNALS

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jan Slavov

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Čala, Ph.D.

BRNO 2021

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Jan Slavov

**ID:** 211174

**Ročník:** 3

**Akademický rok:** 2020/21

**NÁZEV TÉMATU:**

## Nástroj pro vizualizaci a poslech vibračních signálů

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem bakalářské práce je vytvoření nástroje pro grafickou vizualizaci a možnost poslechu vibračních signálů. Aplikace bude umožňovat oběma uvedenými způsoby signály také porovnat. Aplikace bude vytvořena v prostředí LabVIEW a bude využívat standardní zvukový výstup v počítači, nadto může být realizováno využití externího zařízení – např. myDAQ. Zadání lze shrnout do následujících bodů:

1. Seznamte se s možnostmi grafického a zvukového vyjádření signálů v LabVIEW a osvojte si jejich použití.
2. Proveďte průzkum nástrojů, které slouží k práci se signály či se zvukem. Ukažte jejich typické funkce a inspirujte se jimi.
3. Vytvořte nástroj, který dokáže načíst jeden nebo více signálů z různých zdrojů (zejména z různých typů souborů, ale také z HW pomocí DAQmx ovladače), a dále je předzpracovat, zobrazit a přehrát.
4. Umožněte přidání nových funkcí pro zpracování v budoucnu.
5. Nástroj popište, otestujte a vyjádřete se k jeho silným a slabým místům.

**DOPORUČENÁ LITERATURA:**

[1] BLUME, Peter A. The LabVIEW Style Book. Crawfordsville (Indiana): Pearson Education, 2007. ISBN 978-0131458352.

**Termín zadání:** 8.2.2021

**Termín odevzdání:** 24.5.2021

**Vedoucí práce:** Ing. Martin Čala, Ph.D.

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

**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.

## **Abstrakt**

Tato bakalářská práce se zabývá vývojem nástroje pro vizualizaci a poslech vibračních signálů. Cílem práce je zjistit možnosti vývojového prostředí LabVIEW v oblasti, ke které je aplikace určena, dále vyhodnotit podobné programy nabízené na trhu a vytvořit aplikaci, která umožní uživateli základní zpracování signálu, při kterém uživatel přímo uvidí reakci signálu na změnu nastavení. Aplikace dále umožní práci se zařízením myDAQ a poslech signálů pomocí reproduktorů, při kterém se mu zobrazí aktuální pozice přehrávání.

## **Klíčová slova**

Aplikace v LabVIEW, analýza signálů, vibrační signály, vizualizace signálů, poslech signálů, zpracování signálů, práce se zařízením myDAQ

## **Abstract**

This bachelor thesis deals with the development of a tool for visualizing and listening to vibration signals. The aim of the thesis is to investigate the capabilities of the LabVIEW development environment in the area for which the application is intended, to evaluate similar programs offered on the market, and to create an application that allows the user to perform basic signal processing where the user can directly see the signal response to a change in settings. The application will also allow the user to work with the myDAQ device and listen to the signals through the speakers, displaying the current playback position.

## **Keywords**

LabVIEW application, signal analysis, vibration signals, signal visualization, signal listening, signal processing, working with myDAQ

## **Bibliografická citace**

SLAVOV, Jan. *Nástroj pro vizualizaci a poslech vibračních signálů* [online]. Brno, 2021 [cit. 2021-05-24]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/134770>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Martin Čala

## Prohlášení autora o původnosti díla

<b>Jméno a příjmení studenta:</b>	Jan Slavov
<b>VUT ID studenta:</b>	211174
<b>Typ práce:</b>	Bakalářská práce
<b>Akademický rok:</b>	2020/21
<b>Téma závěrečné práce:</b>	Nástroj pro vizualizaci a poslech vibračních signálů

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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: 24. května 2021

-----  
podpis autora

## **Poděkování**

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Martinovi Čalovi Ph.D. za cenné rady a připomínky a za vstřícnost při konzultacích a vypracování Bakalářské práce

V Brně dne: 24. prosince 2020

-----  
podpis autora

# Obsah

1.	Úvod.....	10
2.	Možnosti grafické a zvukové reprezentace signálů v LabVIEW.....	11
2.1	Grafická reprezentace.....	11
2.1.1	Waveform graph .....	11
2.1.2	Waveform chart .....	11
2.1.3	XY graph.....	12
2.1.4	Intensity graph a Intensity Chart.....	12
2.1.5	3D graph .....	13
2.1.6	Polar Plot.....	14
2.1.7	Mixed signal graph .....	14
2.1.8	Zhodnocení grafů.....	14
2.2	Zvuková reprezentace .....	15
2.2.1	Files.....	15
2.2.2	Output .....	16
2.2.3	Input.....	17
2.2.4	Zhodnocení funkcí pro práci se zvukem v LabVIEW .....	17
3.	Průzkum nástrojů pro práci se signály či se zvukem .....	19
3.1	DADiSP.....	19
3.2	DIAdem.....	19
3.3	Audacity .....	20
3.4	Zhodnocení programů .....	21
4.	Hlavní část aplikace .....	22
4.1	Struktura aplikace.....	22
4.2	Přední panel aplikace .....	23
5.	Načtení dat .....	25
5.1	Načtení signálů ze souboru .....	25
5.2	Načtení dat ze souboru INI .....	25
5.3	Načtení dat z myDAQ.....	26
6.	Ukládání dat .....	27
6.1	Uložení signálů do souboru.....	27
6.2	Ukládání dat do souboru INI.....	27



7.	Přehrávání signálů.....	28
7.1	Přehrávání pomocí reproduktoru.....	28
7.2	Přehrávání pomocí zařízení DAQ .....	29
8.	Funkce pro práci s projektem .....	30
9.	Funkce pro práci se seznamy .....	31
9.1	Tvorba seznamu .....	31
9.2	Výběr prvků ze seznamu.....	31
10.	Základní operace se signály .....	32
10.1	Matematické operace.....	32
10.2	Škálování lineární funkcí.....	32
10.3	Výběr podmnožiny signálu.....	33
10.4	Sloučení průběhů.....	33
10.5	Výběr každého N-tého vzorku signálu .....	33
10.6	Filtrace pomocí FIR i IIR filtrů .....	34
10.7	Výpočet FFT.....	34
10.8	Časové zarovnání signálů.....	34
11.	Testování a dokumentace práce .....	36
12.	Závěr .....	37

## Seznam obrázků

Obr. 3-1 Ukázka menu programu DADiSP .....	19
Obr. 3-2 Ukázka menu programu DIAdem .....	20
Obr. 3-3 Ukázka menu programu Audacity.....	21
Obr. 4-1 Blokový diagram aplikace.....	22

# 1. ÚVOD

Vibrace jsou velmi podobné zvuku, a proto s nimi můžu pracovat stejně. Vibrace lze rozdělit na spektrum složené z harmonických signálů stejně jako zvuk. Tento signál můžu zobrazit do grafů nebo jej můžu převést na zvuk a poslechnout si jej. Pro zpracování těchto signálů existuje množství programů, avšak většina je zpoplatněná a má větší či menší nedostatky.

Cílem práce je vytvořit aplikaci, která umožní grafickou vizualizaci a poslech vibračních signálů. Aplikace dále musí umět načtení signálu z různých zdrojů, zpracovat a analyzovat naměřená data. K vývoji této aplikace bude použito prostředí LabVIEW.

Vývojové prostředí LabVIEW od společnosti National Instrument se používá pro grafické programování v jazyce „G“. Tento nástroj je vhodný pro vývoj automatických testovacích systémů, systémů pro ověření návrhů. Dále se používá na měření a analýzu dat a také řízení a monitorování průmyslových zařízení a procesů.

V teoretické části práce se nejdříve zaměřím na možnosti grafické vizualizace a poslechu signálů v LabVIEW. Nejdříve si proberu jednotlivé typy grafů, které se v LabVIEW používají, jejich možnosti a vlastnosti a jaké formáty dat daný graf přijímá na vstupu. Dále se podívám na funkce v paletě sound, které slouží k práci se zvukem, a jejich využití. V posledním bodě teoretické části se podívám na již vytvořené aplikace, které se používají k podobným účelům. Ty mi budou sloužit jako inspirace pro tvorbu mého programu a zhodnotím jejich vlastnosti.

V praktické části se zaměřím na samotnou tvorbu aplikace, její strukturu a celou funkci. Blíže popíši proces přehrávání signálů pomocí reproduktorů, dále proces komunikace se zařízením myDAQ a proces načítání dat z s do něj. Dále se budu zabývat jednotlivými funkcemi na předzpracování signálů a jejich implementací. Do práce jsem také zakomponoval kapitolu zabývající se strukturou samotného projektu a funkcemi pracujících s tímto projektem. Mezi kapitolami se také nachází část která se zabývá tvorbou seznamů z projektu a výběrem z těchto seznamů. Na konci práce ještě popíši proces testování a dokumentace.

## 2. MOŽNOSTI GRAFICKÉ A ZVUKOVÉ REPREZENTACE SIGNÁLŮ V LABVIEW

### 2.1 Grafická reprezentace

V LabVIEW pro grafické zobrazování signálu lze používat celou paletu grafů, kde každý graf se používá pro různý formát dat a umožňuje různé zobrazovací či ovládací funkce.

#### 2.1.1 Waveform graph

Prvním typem grafu, který se používá v LabVIEW, je waveform graph. Tento typ umožňuje zobrazení jednoho, či více rovnoměrně vzorkovaných průběhů. Zobrazované průběhy musí mít tvar  $y = f(x)$ , z toho vyplývá, že každé  $x$  může mít pouze jednu hodnotu  $y$ .

Podporované vstupní datové formáty:

- 1D pole – graf interpretuje hodnotu prvků v poli jako ypsilonové hodnoty bodů v grafu a hodnota  $x$  pro první prvek pole je nulová a pro každá následující prvek roste o jedna.
- 2D pole – umožňuje zobrazit více průběhů. V základním nastavení každý řádek představuje jeden průběh. Pokud chceme, aby graf zobrazoval průběhy ze sloupců 2D pole, může se v nastavení grafu použít funkci transpose. Ostatní vlastnosti pro zobrazení průběhu zůstávají stejné jako u 1D pole.
- Cluster obsahující pole,  $x_0$  a  $\Delta x$  – přidává možnost nastavení počáteční hodnoty  $x$  a růstu  $x$  pro každý prvek. První prvek clusteru představuje počáteční hodnotu  $x$ , v druhém prvku nastavíme hodnotu delta  $x$  a posledním prvkem je 1D nebo 2D pole podle toho, kolik průběhů chceme zobrazit.
- Pole clusterů obsahující prvky  $x_0$ ,  $\Delta x$  a 1D pole – umožňuje zobrazit více grafů o různých délkách.
- Waveform data type – mnoho VI používaných k analýze a tvorbě průběhů, používá právě tento typ. Tento typ uchovává začátek signálu,  $\Delta t$  a 1D pole.
- Dynamický datový typ – přidává další informace o grafu např. popisky.

#### 2.1.2 Waveform chart

Tento typ grafu je velmi podobný jako waveform graph, také umí zobrazovat jeden nebo více průběhů a zobrazované průběhy musí mít tvar  $y = f(x)$ . Rozdíl je v tom, že tenhle graf umí vykreslovat průběhy postupně bod po bodu nebo více bodech a také nerovnoměrně vzorkované průběhy. Tyto body si chart ukládá jako historii dat do bufferu. Počet těchto bodů je v základu 1024, ale lze jej upravit pravím kliknutím na chart a vybráním Chart History Length.

Podporované vstupní datové formáty:

- Skalární hodnota – připiše se na konec průběhu.
- 1D pole – přidává více hodnot najednou.

- Cluster s X prvky skalárních hodnot nebo 1D polí – zobrazuje více průběhů a nastaví pevný počet zobrazených průběhů.
- 2D pole – umožňuje programově měnit počet zobrazovaných průběhů. V základu každý sloupec znamená jeden průběh, ale v nastavení grafu pomocí funkce transpose změním zobrazování průběhů z řádků místo sloupců.
- Waveform data type – vytváří pomocí funkce build waveform. Obsahuje počáteční hodnotu x a hodnotu delta x, které lze využít pro nerovnoměrné vzorkování.

Lze zde také nastavit jeden ze tří typů obnovování grafu. Prvním nastavením je strip. Při tomto nastavení a po dojetí průběhu na konec začne chart posouvat předchozí hodnoty k počátku a nové vkládá na konec průběhu. Při nastavení scope se po dojetí průběhu na konec grafu smaže celý zobrazený průběh, a začne se vykreslovat nový na začátku grafu. Nastavení sweep vypadá podobně pouze s tím rozdílem, že po dojetí průběhu na konec grafu se průběh nemaže, ale postupně přepisuje od začátku grafu.

### 2.1.3 XY graph

XY graph je kartézský souřadnicový systém, který umožňuje zobrazit funkce s více hodnotami y pro jedno x, jako jsou kruhové tvary. Zkratka XY graph umožňuje zobrazit libovolnou sadu bodů.

Podporované vstupní datové formáty pro zobrazení jednoho průběhu:

- Cluster se dvěma 1D poli – první 1D pole představuje souřadnice X a druhé souřadnice Y.
- Pole clusterů se dvěma skalárními čísly – první číslo představuje souřadnici X a druhé souřadnici Y, přičemž každý cluster znázorňuje jeden bod.
- Pole komplexních čísel – reálná část znázorňuje souřadnici X a imaginární Y

Podporované vstupní datové formáty pro zobrazení více průběhů:

- Pole clusterů se dvěma 1D poli – pole v clusteru udávají souřadnice X a Y a každý cluster představuje jeden průběh.
- Pole clusterů s polem komplexních čísel – opět každý cluster představuje jeden průběh. Reálná část nám udává čísla souřadnici X a imaginární část souřadnici Y.
- Pole clusterů obsahující pole s clusterem skalárních čísel – skalární čísla představují souřadnice X a Y. Cluster se dvěma čísly představuje bod a druhý cluster obsahující pole představuje průběh.

### 2.1.4 Intensity graph a Intensity Chart

Intenzitní grafy nám umožňují zobrazit 3D data do 2D prostoty pomocí barevných bloků umístěných do kartézského systému. Na vstup připojují 2D pole, kde hodnota prvku odpovídá hodnotě z a tedy i barvě zobrazené v grafu a indexy prvku odpovídají souřadnicím v grafu, přičemž graf začíná indexovat od levého dolního rohu. Barevný blok má jednotkovou plochu, která je vymezena body definovanými indexy pole. Řádky pole představují jednotlivé sloupce grafu, pokud chci, aby řádky pole představovaly řádky grafu musím v nastavení grafu zvolit funkci Transpose Array.

Jediný rozdíl mezi typy intensity graph a intensity chart je ten, že pokud dám na vstup intensity grafu nová data, nahradí tyto data všechna předchozí, kdežto intensity chart vždy posouvá počátek kartézského systému napravo od posledního zobrazeného datového bloku. Nový datový blok se poté zobrazí napravo od předešlého. Po dojetí na konec okna se graf updatuje dle zvoleného režimu podobně jako waveform chart. Intensity chart si také ukládá historii dat do vyrovnávací paměti s velikostí 128 bodů, kterou lze navýšit pomocí nastavení Chart History Length.

V grafech intenzity může zobrazit až 256 diskrétních barev. Tato škála je v základním nastavení počítána od bílé přes modrou po černou, avšak tyto barvy lze vyměnit za jiné, či přidat další. Pokud mi tato barevná škála nestačí, je možné si do LabVIEW nainstalovat knihovnu sound and vibration, která obsahuje prvek SV intesity graph. Tento graph se chová velmi podobně jako intesity graph, ale v základním nastavení zobrazuje barvy v rainbow rozložení, čímž se stává přehlednější.

### 2.1.5 3D graph

3D grafy slouží k vizualizaci trojrozměrných dat. V LabVIEW existuje velké množství různých 3D grafů, a proto jsou uloženy v samostatné paletě 3D graph.

Mezi 3D grafy patří:

- Scatter – zobrazuje data jako jednotlivé barevné body, jejichž barvu určuje dle souřadnice z.
- Stem – zobrazuje data jako barvené impulzy, u kterých barva závisí na velikosti pulzu.
- Comet – vytváří jednobarevnou křivku, která postupně prochází všemi zadanými body.
- Surface – propojuje všechny data, ale místo křivky vytváří plochu, která je barevně rozdělena na jednotlivé roviny.
- Contour – zobrazuje plochu protínající všechny body, ale navíc na této ploše vytváří barevné vrstevnice, kde každá vrstevnice má svou barvu podle výšky, ve které se nachází.
- Mesh – buduje z bodů síťový povrch s mezerami, kde každý bod je propojený sítí s okolními body. Celá síť je spektrálně barevná, dle úrovně z.
- Quiver – vytváří povrch spojující všechny body, kde na každé rovině povrchu je vyznačen normálový vektor. Celý povrch nabývá spektrálního zbarvení dle hodnoty z v daném místě.
- Ribbon – ztvárňuje graf paralelních barevných čar rovnoběžných s osou x, jejichž barva je určena dle hodnoty y.
- Bar – vytváří graf vertikálních sloupců, přičemž jednotlivé sloupce mají barvu dle hodnoty y.
- Pie – výsečový graf v 3D prostoru, kde každá výseč má svou barvu.
- Waterfall – kaskádový graf, který vykresluje křivky paralelně k ose x a pod nimi barevnou oblast. Barva oblasti je určena dle nejvyšší hodnoty křivky.
- 3D surface graph – vykresluje barevný povrch v 3D prostoru podobně jako surface graf, ale narozdíl od něj mění barvu plynule dle hodnoty z.
- 3D parametric graph – vytváří plochu zadáním parametrů do parametrických rovnic, čímž dovede vytvářet geometrická tělesa.

- 3D Line Graph – vytváří křivku stejně jako comet, pouze s tím rozdílem, že tato křivka mění barvu v závislosti na z. 3D Line Graph se používá k vizualizaci dráhy pohybujícího se objektu.

### 2.1.6 Polar Plot

Polární graf se používá pro zobrazení hodnot v závislosti na úhlu. Pro zobrazení polárního grafu existují v LabVIEW dva způsoby. Prvním způsobem je použití prvku polar plot na front panelu a přivést na něj vstup ve formě 2D pole clusterů se třemi 1D poli, prvním pro velikost druhým pro fázi a třetím pro čas/RPM. Druhým je použití VI Polar Plot with Point Options, na které se musí přivést 1D pole s clusterem dvou skalárních hodnot, první pro velikost a druhou pro fázi. VI poté z přijatých dat vytvoří obrázek polárního grafu, který se dá zobrazit.

### 2.1.7 Mixed signal graph

Mixed signal graph dovoluje zobrazit analogové a digitální data v jednom prvku. Tento graf přijímá všechny datové typy Waveform grafů, XY grafů a Digital waveform grafů.

Graf lze rozdělit na více oblastí vykreslení grafu, každá oblast může zobrazit pouze digitální nebo analogové grafy, nikoli obojí. Tyto oblasti si vytváří automaticky dle potřeby, tak aby vyhovovali analogovým nebo digitálním datům. Každá oblast vykreslení si vytvoří vlastní měřítko y, ale všechny oblasti sdílejí společnou stupnici x, tím umožňuje srovnání více signálů digitálních a analogových dat.

Pro zobrazení průběhů jednoho typu stačí použít libovolný datový typ, který umožňuje vykreslit průběh do Waveform grafů, XY grafů nebo Digital waveform grafů, pro vykreslení více různých průběhů se spojí datové typy do clusteru, které se přivedou na vstup grafu.

Legenda grafu se skládá ze stromů průběhů, který se nachází nalevo od oblastí grafu. Každá oblast grafu je ve stromu označena jako group s číslem odpovídající pořadí, ve kterém byla umístěna do grafu. K přesunutí průběhů z jedné oblasti do jiné se může použít legenda grafu .

### 2.1.8 Zhodnocení grafů

V LabVIEW existují ještě další grafy, které jsou pro tuto práci nedůležité. A těmi jsou Digital waveform graph a 2D graph. Digital waveform graph je nedůležitý, protože slouží ke zobrazení digitálních dat, která v této práci nepoužiju. 2D graph zde uvádím, protože všechny 2D grafy jsou také XY grafy a lze je tedy zobrazit na XY graph.

V našem programu budu pracovat pouze se signály, které mají již danou velikost a není potřeba je vykreslovat postupně, proto nevyužiji žádných grafů typu chart.

3D grafy jsou v LabVIEW složité, často velmi nepřehledné, a náročnější na výkon počítače. Z těchto důvodů je nejhodnější pro zobrazení trojrozměrných dat použít intenzitní grafy, zejména SV intenzity graf z knihovny Sound and Vibration, který má přehlednější barevnou škálu.

Průběh zvukových signálů má tvar  $y = f(x)$ , a proto nejčastějším grafem pro zobrazení bude waveform graph, který je určen právě pro tento typ signálů. Na tento typ grafu přivede data ve formátu waveform, který je jednoduchý na použití a umožňuje přenášet

velké množství informací o signálech. Pro ostatní 2D data bude použit XY graph, avšak zatím se nepředpokládá jeho použití, v takovém případě by též stalo zvážit využití mixed grafu, který umožňuje použít XY a Waveform graf současně.

## 2.2 Zvuková reprezentace

Pro práci se zvukem se v LabVIEW používá paleta sound. Ta je rozdělena na sekce files, output, a input.

### 2.2.1 Files

Sekci files se nachází mnoho VI, které slouží k práci se soubory typu WAV, pokud chci pracovat s jinými soubory je nutné používat data pluginy. První VI je Sound file open, toto VI slouží k otevření souboru WAV. Lze zde nastavit, zda chci soubor otevřít pro čtení či pro zápis. K tomu to VI je nutné přivést na vstup cestu k souboru, který chci otevřít, dále můžu přivést error cluster, pro přenos chyby mezi VI. Při nastavení práv VI pro zápis je možné přivést na vstup i cluster definující formát souboru. Zde je uvedena vzorkovací frekvence, počet kanálů a velikost vzorku bitech, pokud nebudeme formát zadávat, VI si automaticky nastaví vzorkovací frekvenci na 22050 S /s, počet kanálů na 2 a velikost vzorku na 16 b. Funkce vrací odkaz na soubor a error cluster informující o chybě ve VI.

Dalším VI je Sound file info. Toto VI slouží k zjištění informací o WAV souboru. Zde je nutné přivést na vstup buďto odkaz na soubor vytvořený z VI Sound file open nebo přímo cestu k danému souboru. Funkce pak vrací odkaz nebo cestu dle vloženého vstupu a informace o formátu souboru a počtu vzorků na jednom kanále. Funkce umožňuje stejně jako Sound file open přivést na vstup error cluster pro přenos chyby a na výstup vrací error cluster s informací o chybě ve VI.

Pokud chci soubor přečíst, použiji funkci Sound file read. U tohoto VI je nutné přivést na vstup odkaz na soubor. Dále je možné určit počet vzorků, které se mají načíst z každého kanálu souboru. Další volitelné vstupy jsou offset polohy a poziční režim, tyto vstupy společně určují místo odkud se data začínají číst. Poziční režim lze nastavit buď na absolute, pro spuštění funkce na začátku souboru nebo relative pro spuštění funkce v bodě umístění značky. Základní nastavení pozičního režimu je relative. Offset polohy pak určí posun od místa určeného režimem pozice. Udává se v počtu vzorků, o který se chci posunout. Základní nastavení offsetu je 0. Tato funkce má opět na vstupu a na výstupu error cluster pro přenášení chybové hlášky. Dalšími výstupy jsou odkaz na soubor, offset, který označuje nové umístění značky vzhledem k začátku souboru, informace, zda byl soubor přečten do konce, a data v podobně pole průběhů.

Pro zápis do souboru se používá VI Sound file write. K tomu je třeba přivést na vstup funkce odkaz na soubor. Dalšími vstupy jsou data ve formátu pole křivek a error cluster. Na výstupu se objeví odkaz na soubor a error cluster informací o chybě.

Po otevření souboru je nutné jej zavřít, k tomu slouží funkce Sound file close. Zde se na vstup přivede pouze reference na soubor a error cluster a na výstupu je také error cluster.

Předposlední funkcí v sekci files je Sound file read simple, ta automaticky otevře soubor, vyčte data a zavře soubor. Vstupy i výstupy jsou velmi podobné jako u Sound file read kromě toho, že místo odkazu na soubor používá cestu k souboru a také tato funkce nemá výstup s informací, zda byl přečten konec souboru.



Posledním VI v této sekci je Sound file write simple, to automaticky otevře soubor, zapíše do něj data a zavře soubor. Vstupy i výstupy jsou podobné funkci Sound file write, jen používá cestu k souboru místo odkazu na něj a má navíc vstup určující velikost jednoho vzorku.

## 2.2.2 Output

Další sekci je output, tato sekce se zabývá konfigurací a ovládáním výstupních zvukových zařízení. Nachází se zde Express VI Play waveform, které umí přehrát data ze zvukového výstupního zařízení pomocí konečného vzorkování. Toto expresní VI automaticky nakonfiguruje úlohu výstupního zařízení a po dokončení ji ukončí. K tomu používá dialogové okno, kde se nachází seznam připojených zařízení. Po vybrání zařízení se zobrazí jeho možnosti, zde se dá najít minimální a maximální vzorkovací frekvenci, kterou zařízení dovede přehrát, dále kvalitu vzorku v bitech a počet kanálů. Na VI se pak připojují data, které chceme přehrát, číslo zařízení, na kterém se budou přehrávat data, a error cluster na vstupu a na výstupu pro přenos chyby.

Další možností pro okamžité přehrání je VI Play sound file, který umožní otevřít soubor typu WAV a rovnou jej přehrát. K tomu se musí na vstup přivést cesta k souboru, který má být přehrán. Dalším vstupem je ID zařízení, které chci použít. Vstup timeout určuje čas v sekundách, po který VI čeká na dokončení operace. Výchozí hodnota je nastavena na 0. Pokud toto nemá nastavený časový limit na 0 a tento čas vyprší, vrátí chybu. Při nastavení na nulu se VI ukončí. Zatímco zvuk pokračuje v přehrávání. Pokud je tento vstup roven -1 VI čeká, dokud se nedokončí přehrávání. Na výstup VI vrací ID úlohy a také má na vstupu a výstupu error cluster.

Funkci čekání na konec přehrávání umožňuje též VI Sound output wait. Na toto VI se připojuje ID úlohy, error cluster a timeout. Timeout určuje čas, po který VI čeká na dokončení zvukové operace. Pokud tento čas vyprší, vrací chybu. Výchozí hodnota je 10 a pokud nastavíte časový limit na -1, čeká neomezeně dlouho.

Pokud nechci nahrát data ze souboru a nevyhovuje mi přehrávání před Express VI, musím nejdřív nakonfigurovat výstupní zařízení. K tomu použiji VI Sound output configure. Na toto VI se může přivést error cluster, ID zařízení, ke kterému chci přistupovat, počet vzorků na kanál, zvukový formát, který udává rychlost přehrávání, počet kanálů a počet bitů na vzorek, a sample mode, který určuje, zda chci zapisovat pouze jednou (možnost Finite Samples) nebo průběžně (Continuous Samples). Na výstupy nám VI přivede opět error cluster a ID úlohy.

Po nakonfigurování výstupu můžu zapsat data. K tomu slouží VI Sound output write. Toto VI zapíše data na výstupní zařízení a také rovnou spustí přehrávání. Zde přivádím na vstup ID úlohy, data, která chci přehrát, timeout, který funguje stejně jako u VI Play sound file, pouze jeho výchozí hodnota je 10. Dále má VI vstupní a výstupní error cluster. Na výstup také přivádí ID úlohy.

Dalším VI je Sound output stop, to slouží k zastavení přehrávání. Jeho opakem je VI Sound output start, které spouští přehrávání po jeho zastavení. Obě tyto funkce mají na vstupu i výstupu pouze ID úlohy a error cluster.

Pro ukončení úlohy je potřeba provést funkci Sound output clear, tím se zastaví přehrávání zvuku, vymažou se data z paměti a ukončí se úloha. K tomu se musí přivést na vstup ID úlohy a na výstupu už jej nevrací. Toto VI má také na vstupu i výstupu error cluster.

Předposlední funkcí v této sekci je Sound output set volume, tato funkce slouží k nastavení úrovně hlasitosti přehrávání výstupního zařízení. Pro nastavení hodnoty se přivede na vstup ID úlohy a hodnotu v rozsahu 0 až 100, na kterou se hlasitost nastaví. Dalším vstupem je error cluster a na výstup nám funkce vrátí ID úlohy a error cluster.

Posledním VI v sekci output je Sound output info, která po přijetí ID úlohy, které vrátí na výstup ID úlohy, předá na výstup informaci, zda se právě něco přehrává a jaká je aktuálně nastavená hlasitost. Dále VI umožňuje připojení error clusteru na vstup a zjištění chyby na výstupu.

### 2.2.3 Input

Poslední sekci je input, tato sekce slouží ke konfiguraci a ovládání vstupních zvukových zařízení. Tato sekce má méně funkcí než Output, ale také má jedno express VI Acquire sound. Toto expresní VI automaticky nakonfiguruje vstupní úlohu, získá data ze zařízení a po dokončení úlohu vymaže. V dialogovém okně tohoto express VI se může nastavit většina vstupů. Těmi jsou číslo zařízení, které chci použít, počet kanálů, rozlišení určující kvalitu vzorků, dobu trvání v sekundách, po kterou chci získávat zvuk, vzorkovací frekvenci. Po zadání těchto vstupů si můžu zobrazit graf s náhledem, jak bude signál vypadat. Všechny vstupy v dialogovém okně je možné přivést na vstup a řídit programově, navíc je zde vstup pro přívod předchozí chyby. Výstupem funkce jsou data načtené ze zařízení a informace o chybovém stavu funkce.

Pokud nechci použít express VI musím nakonfigurovat vstupní zařízení pomocí VI Sound input configure. Na toto VI můžu přivést error cluster, ID zařízení, ke kterému chci přistupovat, počet vzorků na kanál, zvukový formát, který udává rychlost přehrávání, počet kanálů a počet bitů na vzorek, a sample mode, který určuje, zda chci číst data ze zařízení pouze jednou (možnost Finite Samples) nebo průběžně (Continuous Samples). Na výstupy nám VI přivede error cluster a ID úlohy.

Po nakonfigurování zařízení můžu začít číst data, k tomu slouží VI Sound input read. Vstupními parametry jsou ID úlohy, počet vzorků na kanál, tím se určí, kolik vzorků se má z kanálů ze všech kanálů načíst. Dalším vstupem je Timeout, který určuje dobu, po kterou VI čeká na dokončení nahrávání a po jeho vypršení vrátí chybu na výstupu. V základním nastavení má hodnotu 10 a pokud nastavíme hodnotu -1, tak VI čeká, dokud se operace nedokončí. Dalšími výstupy jsou data, získaná ze zařízení, a ID úlohy.

Dalším VI je Sound input stop, které používám, pokud chci zastavit nahrávání. Opačným VI je Sound input start, které spouští nahrávání po jeho zastavení. Obě tyto funkce mají na vstupu i výstupu pouze ID úlohy a error cluster.

Pro ukončení úlohy je potřeba provést funkci Sound input clear, tím zastavím nahrávání zvuku, vymažou se data z vyrovnávací paměti a ukončí se úloha. K tomu se musí přivést na vstup ID úlohy a na výstupu už se nevrací. Toto VI má také na vstupu i výstupu error cluster.

### 2.2.4 Zhodnocení funkcí pro práci se zvukem v LabVIEW

V závěrečné aplikaci nebudu pro nahrávání dat využívat mikrofon, ale použiji zařízení myDAQ. Z toho důvodu je pro mě sekce input nevhodná a je nutné využít jiné knihovny. V našem případě se jedná o knihovnu DAQmx.

Další způsob získávání dat bude načtení ze souboru. K tomu, aby výsledná aplikace byla více univerzální je nutné umožnit práci s různými datovými formáty souborů, proto

sekce files, která umí pracovat jen se soubory typu WAV, není vhodná pro použití a místo ní budu pracovat se soubory přes data pluginy a následně načtená data převádět na formát, který používá typ WAV.

Ze sekce output nevyužiji Express VI, protože by mne znemožnilo dále ovládat přehrávání. VI Play sound file, také není vhodné, protože stejně jako sekce files pracuje jen se soubory typu WAV. Z toho vyplývá, že nejvhodnějším řešením pro moji aplikaci je použít VI pro konfiguraci zařízení, poté přes VI Sound output write, zapsat data do zařízení, následně pomocí VI Sound output start a VI Sound output stop ovládat přehrávání souboru, a nakonec použít VI Sound output clean pro ukončení přehrávání a celé úlohy. K tomu je vhodné použít VI pro ovládání hlasitosti a případně i VI Sound output info pro zjištění informace, zda se právě data přehrávají.

## 3. PRŮZKUM NÁSTROJŮ PRO PRÁCI SE SIGNÁLY ČI SE ZVUKEM

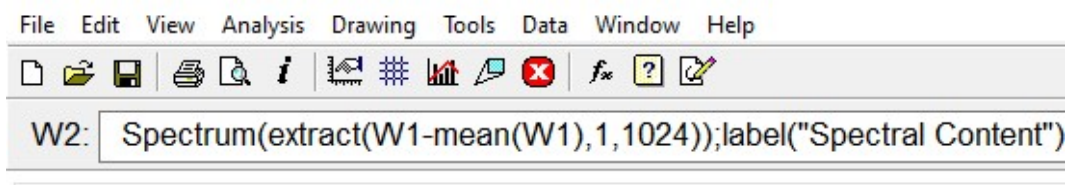
Existuje množství různých nástrojů pro práci se signály či se zvukem. V této práci se podívám na tři zástupce z různých kategorií, u kterých si rozeberu jejich základní funkce a některé funkce následně aplikuji do výsledné aplikace.

### 3.1 DADiSP

Prvním programem, na který se podívám, je DADiSP. Pro provoz tohoto programu je nutné koupit licenci s předplacením na měsíc, rok nebo na pořád. DADiSP se přímo používá pro analýzu vibrací, umožňuje zobrazení signálu v různých podobách, jako jsou tabulky, spojnicové grafy, XY graf, 3D povrchové grafy, histogramy, grafy intenzity a mnoho dalších grafů.

DADiSP v základu pracuje se soubory DAT, lze ale doinstalovat moduly pro práci s typy COMTRADE, ASCII, WDQ, WDC, WDH, MAT, WVF, WDF, WAV, DXD, D7D, D7Z, DXZ, ISF, PNRF, NRF, LRF, TDM, TDMS a UFF.

Ovládání programu je velmi jednoduché, Pohyb v grafu se provádí stisknutím levého tlačítka myši a jejím posunem ve směru, kam chci graf posunout. Přibližování a oddalování se provádí pomocí kolečka myši. V menu se nachází základní tlačítka pro vytvoření prázdné plochy, uložení, načtení a tisk plochy, dále tlačítka pro náhled tisku, zobrazení názvů os a grafů, mřížky grafu, a tlačítka pro změnu typu grafu.



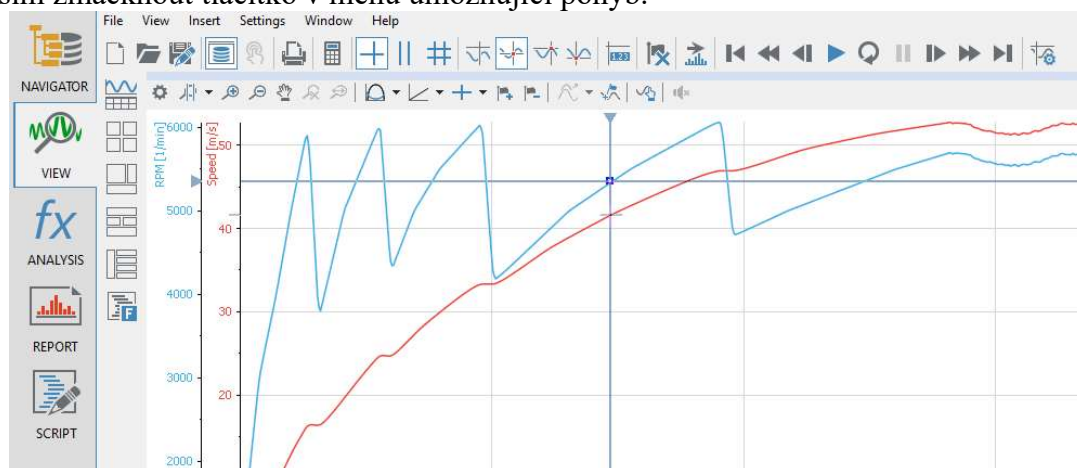
Obr. 3-1 Ukázka menu programu DADiSP

Celá práce se signálem se provádí zápisem funkcí obdobně jako např. v programu Excel. DADiSP umí provádět základní operace se signály jako jsou sčítání, odčítání, zobrazení signálu x-krát za sebou, nalezení minima, maxima a vrcholů. Dále tento program má funkce pro nastavení vzorkování a prahu i funkce pro zobrazení spektra signálu a fourierovi transformace. Tento program také umí vytvářet spojnicu trendu pro zadaný signál.

### 3.2 DIAdem

DIAdem od společnosti National instruments je sice placený, ale zároveň je to všestranný software pro práci a analýzu s naměřenými daty. Tento program lze rozdělit na pět základních oken. První okno se nazývá Navigator a slouží k načítání a výběru dat. DIAdem umí pracovat s těmito typy souborů TDM, TDMS, ATRF, ATF, WAV, WMA, MP3, AIF, AU, AFC, DAT, DOCX, JPG, PDF, PNG, PPTX, TIF, XLSX, ODS, XLS, XLMS, GPX, LVM, NMEA, CSV, UFF, UF, UNV, BUNV.

Druhé okno se jmenuje View a slouží k zobrazení vybraných dat. Pro zobrazení signálu využívá DIADem stejných typů grafů jako DADiSP. Také ovládací menu je velmi podobné jako u DADiSP, ovšem najdou se zde i rozdíly. Tím hlavním je, že tento program umí přehrávat zobrazený signál, s tím souvisí i nové funkce jako je výběr začátku a konec spouštěného úseku, a ovládací tlačítka v menu pro toto přehrávání např. start, stop, opakování atd. Také se v grafu ukazuje kurzor, který udává polohu přehrávání. Zde také nastává rozdíl v ovládacím menu, kdy stisknutí tlačítka s pohybem myši neudává posun grafu, jak tomu bylo u DADiSP, ale posun kurzoru. Pro posun grafu musím zmáčknout tlačítko v menu umožňující pohyb.



**Obr. 3-2 Ukázka menu programu DIADem**

Třetím oknem je Analysis, zde se nachází veškeré funkce pro práci s daty, které umožňuje tento program. Funkce jsou velmi podobné, jako u DADiSP, hlavní rozdíl je v tom, že v DIADemu se funkce zadávají s palety funkcí, a ne do řádku.

Předposledním oknem je Report, ten slouží k vytváření zprávy z měření a finálnímu zobrazení získaných dat.

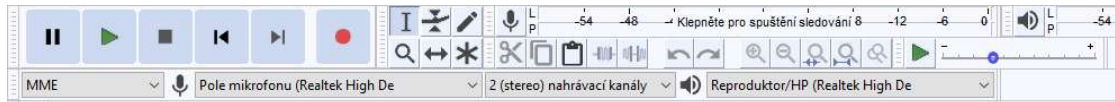
Posledním oknem je Script, tímto oknem DIADem umožňuje zpracování velkého množství souborů do stejné finální podoby pomocí kódu.

### 3.3 Audacity

Posledním vybraným zástupcem je Audacity, tento program slouží k editaci zvuku. Patří mezi nejpoužívanější programy pro editaci zvuku na trhu a to především, protože je ke stažení zdarma jako open source program a zároveň umožňuje široké množství funkcí pro editaci zvuku. Do této práce jsem si vybral právě tento program, kvůli části projektu pracujícího se zvukem. Umožňuje pracovat se soubory typu MP3, MP2, WAV, AIFF, FLAC, AUP, OGG.

Tento program umí zobrazit a přehrát několik zvukových stop najednou, jeho základní ovládací je jednoduché. Kolečkem myši lze přibližovat zvukovou stopu. V menu se nachází mnoho tlačítek. Pro tuto práci jsou nejdůležitější tlačítka pro ovládací, přehrávání a také tlačítko pro nahrávání zvuku. Dále se zde nachází ovládací hlasitosti nahrávání a přehrávání a umožňuje výběr vstupního a výstupního zařízení. Přes tyto tlačítka je umožněn přístup k nástrojům Audacity, mezi které patří nástroj pro výběr části zvukové stopy, dále nástroj pro přiblížení stopy a nástroj pro posouvání v stopě. Dalšími zajímavými nástroji jsou nástroj pro kreslení, který umožní měnit velikost jednotlivých vzorků stopy a nástroj obalové křivky, který umožňuje měnit hodnoty amplitudy vzorků

v poměru ke křivce. Posledním nástrojem je více-účelový nástroj, ten automaticky vybírá, který z předešlých nástrojů chce použít a podle něj se chová. Audacity také dovoluje číselné nastavení začátku a konce přehrávaného úseku a zobrazuje aktuální časovou pozici.



**Obr. 3-3 Ukázka menu programu Audacity**

### 3.4 Zhodnocení programů

V moji práci se budu nejvíce inspirovat programem DIAdem, který se nejvíce podobá požadavkům na tuto práci. V tomto programu se zobrazuje přehled aktuálně otevřených souborů dat, s kterými můžu pracovat, a to bych také chtěl do své práce zahrnout. Dále by bylo vhodné se inspirovat vytvořením palet funkcí jako je tomu v sekci analysis. Mezi tyto funkce budou především patřit matematické operace se signály a škálování lineární funkcí. Dále vybrání podmnožiny ze signálu, sloučení signálů a změna vzorkování signálu. Program by měl také umožňovat používání filtrů a Furierovy transformace.

Ovládání tohoto programu je velmi příjemné, a proto jej chci také využít. Především možnosti posunu grafu a posunu začátku přehrávání pomocí myši, dále se zde budou nacházet velmi podobná funkční tlačítka, která jsou též dosti podobná těm v DADiSP. Avšak na rozdíl od DADiSP musím přidat tlačítka pro přehrávání, a proto chci umožnit uživateli nastavit začátek a konec přehrávání.

Z audacity by bylo vhodné využít funkce, které umožňují upravovat velikost jednotlivých vzorků. Dále by bylo pro uživatele přívětivé umožnit zobrazovat průběh přehrávání a nastavovat začátek a konec přehrávání číselně. Poslední důležitým bodem je přidat k tlačítkům pro přehrávání také tlačítko pro ovládání nahrávání, které v DIAdemu není.

Mezi nejpoužívanější typy souborů patří TDMS, MAT, CSV, WAV a DAT a ty by též měl podporovat můj program.

## 4. HLAVNÍ ČÁST APLIKACE

Celá aplikace je vyvíjena ve vývojovém prostředí LabVIEW 2019 doplněném o následující knihovny:

Digital Filter Design Toolkit - pro tvorbu, analýzu a práci s FIR a IIR filtry.

Sound and Vibration Toolkit - poskytuje funkce pro audio testy, akustická měření a aplikace pro měření vibrací.

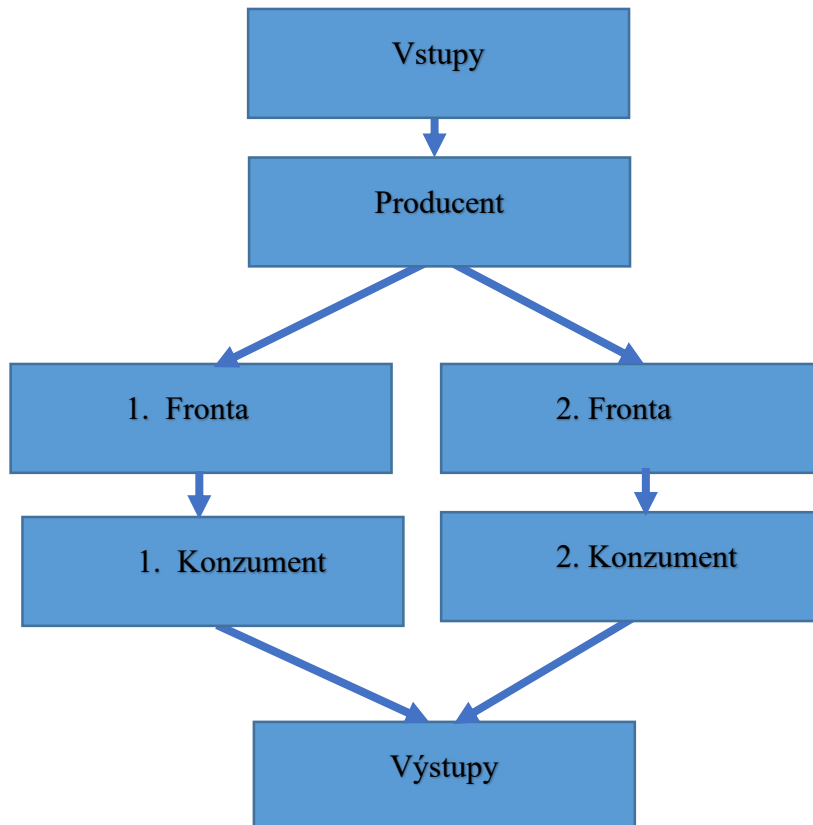
DAQmx – ovladač určený pro komunikaci se systémem typu DAQ určeným pro sběr dat a úpravu signálů.

Advance Signal Processing Toolkit – knihovna určená pro zpracovávání signálů, jejich spektra atd.

openG – umožňující pokročilou práci se soubory, dále obsahuje funkce rozšiřující možnosti práce se základními prvky LabVIEW, jako jsou pole, řetězce, varianty a zpravování chyb.

### 4.1 Struktura aplikace

Aplikace má strukturu producent-konzument, kde jedno vlákno bude generovat požadavky a další vlákna tyto požadavky zpracovávají. V mé aplikaci je použit jeden producent a dva konzumenti. První konzument je určen k práci s uživatelským rozhraním a volání funkcí pro předzpracování signálu. Druhý konzument slouží pro přehrávání signálu pomocí reproduktorů nebo systému DAQ, dále vykovává komunikaci s těmito zařízeními a během přehrávání posouvá ukazatel aktuální pozice přehrávání.



Obr. 4-1 Blokový diagram aplikace

Producent předává data jednotlivým konzumentům pomocí fronty, kdy každý konzument má vlastní frontu. Konzumenti předávají producentovi pouze informaci a ukončení jejich vlákna pomocí eventu a po obdržení obou eventů se producent také ukončí. Tím docílím, že producent může vytvářet požadavky nezávisle na rychlosti konzumenta a konzumenti budou postupně všechny požadavky zpracovávat bez ztráty dat i při zahlcení požadavky.

Dále si každý konzument ukládá svá data, potřebná pro jeho funkci, k tomu jsou vytvořeny typové definice pro každého konzumenta. První konzument obsahuje informace o posledním vybraném políčku v seznamu filtrů, operací a signálů, ID posledního vybraného signálu, filtru a operace. Dále si ukládá tři dvojrozměrné pole reprezentující jednotlivé seznamy, a nakonec úložiště o třech polích reprezentující samotný projekt, zde jsou všechny informace o všech vytvořených operacích, signálech a filtrech.

Mezi uložené informace o filtru patří číslo reprezentující ID filtru, dále řetězec znaků jako jeho název, poté datový typ filtru, ve kterém jsou uložena data charakterizující filtr: Poté je zde další řetězec, pro uložení popisu filtru, dále číslo ID operace, ukazující na operaci, ze které vznikl a datum jeho vzniku. Přičemž jméno, ID filtru a ID operace je vždy unikátní pro každá zápis.

V informacích o operaci ukládám, stejně jako u filtru číslo reprezentující ID, řetězec jako jméno operace, dále nastavení operace v podobě element variant, který se převede na základě typu operace na správný formát. Proto dále ukládám enum ukazující typ operace, a nakonec se ukládá i její popis v podobě řetězce. U každého záznamu operace je unikátní její ID, název a variant jejího nastavení.

Do informací o signálu patří ID signálu, jeho jméno, poté samotná data signálu v podobě prvku waveform, popisek, ID operace, ze které byl signál vytvořen a pole ID signálů, které bylo potřeba pro jeho vytvoření, a nakonec čas, kdy byl signál stvořen. U záznamu signálu je unikátní pouze ID signálu a jeho jméno.

Druhý konzument si ukládá data o nastavení používaného výstupního zařízení. Pro systém DAQ jsou to čísla určující maximum a minimum rozsahu napětí, odkaz na fyzický kanál zařízení, nastavení výstupního terminálu, odkaz na časový zdroj a vzorkovací frekvenci zařízení. Pro reproduktory ukládá ID zařízení, dále celkový počet vzorků a informace o zvukovém formátu, kde se nachází vzorkovací frekvence, počet kanálů a počet bitů na jeden vzorek.

Mezi další data druhého konzumenta patří odkaz na úlohu systému DAQ a odkaz na úlohu reproduktoru. Pro určení aktuální pozice během přehrávání se u reproduktorů ukládá čas pomocný čas a u systému DAQ aktuální pozice posledního zapsaného prvku na DAQ a volné místo v zařízení. Konzument si dále ukládá upravená data, které se posílají do zařízení a pomocnou binární proměnou určující chod přehrávání.

## 4.2 Přední panel aplikace

Na předním panelu aplikace se nachází control tab se čtyřmi stránkami. První stránka slouží pro vizualizaci signálů, jejich přehrávání a volání funkcí na předzpracování. V horní části se nachází tlačítka na načítání, ukládání a vytváření projektu, tlačítka na výběr signálu a tlačítka pro přehrávání signálu. Vedle těchto tlačítek jsou prvky určené pro nastavování přibližování v grafu, jako je citlivost na přibližování pomocí kolečka, výběr osy, na které se má přibližovat a číslo určující rozsah zobrazení signálu před a za kurzorem během přehrávání. Pod těmito prvky se nachází samotný graf, do kterého lze



vložit dva signály jeden pro levý kanál přehrávání a druhý pro pravý. Průběhy jsou v legendě označeni R-jméno signálu pro pravý kanál a L-jméno kanálu pro levý. Vedle grafu se nachází control tab určující typ zařízení, na kterém se mají signály přehrát a obsahující nastavení pro toto zařízení. Pod tímto nastavením se nachází tlačítka volající příslušné funkce ke zpracování signálu.

Na druhé stránce je seznam všech vytvořených signálů, který lze řadit pomocí prvků nad tímto seznamem určující sloupec a typ třízení. Dále se zde nachází tlačítka pro ukládání a načítání signálů, tlačítka pro změnu názvu a popisu signálu a řetězce obsahující název a popis signálu.

Třetí stránka je určena pro seznam vytvořených operací a čtvrtá stránka pro seznam filtrů. Obě stránky jsou vzhledově shodné se stránkou pro zobrazení seznamu signálů.

## 5. NAČTENÍ DAT

Aplikace umožňuje načtení signálů ze souborů nebo z externího zařízení myDAQ. Dále umožňuje načtení filtrů, operací a celého projektu se souboru INI.

### 5.1 Načtení signálů ze souboru

Pro načtení signálů ze souboru, byla použita knihovna Storage/DataPlugin. Díky této knihovně aplikace umožňuje otevřít všechny datové typy, pro které má uživatel nainstalované data pluginy. Ty jsou dostupné na stránkách National Instruments.

Spuštěním této funkce dojde k její inicializaci a k otevření nového okna, ve kterém jsou tři funkční tlačítka. Tlačítkem Open se spustí proces načtení dat. Tento proces se skládá z několika funkcí. Nejdříve se otevře okno průzkumníka souborů, ve kterém si uživatel vybere soubor, který chce otevřít. Poté dojde ke kontrole, zda soubor existuje nebo nedošlo k zavření tlačítkem cancel, pokud tato událost nastala funkce dále nic nevykonává a vrací chybové hlášení. V opačném případě se spustí funkce na otevření souboru, které sama identifikuje podle koncovky souboru, jaký data plugin je třeba použít. Další funkcí lze vyčíst ze souboru reference na jednotlivé kanály, které zpracuje další funkce. Ta ze souboru vyčte informace o každém kanálu, kterými jsou jméno, popis a samotná data tvaru pole waveformů. Tyto data se zobrazí v okně funkce a dojde k zavření souboru.

Po načtení může uživatel provádět úpravy jmen a popisů průběhů, případně načíst nový soubor, kterým přepíše předchozí načtená data. Případně je možné si přiblížit průběhy v grafu pomocí kolečka myši.

Své úpravy uživatel potvrdí tlačítkem OK, kdy dojde k uložení všech signálů do úložiště hlavní aplikace a dojde k ukončení funkce. Pokud dojde během ukládání k chybovému hlášení funkce nadále pokračuje ve svém běhu a je třeba danou chybu vyřešit. Chybou může být například shoda ukládaného jména signálu s jiným signálem v úložišti. Pokud chce uživatel pouze aplikaci ukončit stačí stisknout tlačítko cancel, kdy nebudou provedeny žádné změny v projektu a funkce se ukončí.

### 5.2 Načtení dat ze souboru INI

Operace, filtry a projekt mají velmi podobnou strukturu funkce, kdy při volání se otevře okno průzkumníka souborů, ve kterém si uživatel vybere soubor, který se má otevřít. Poté dojde ke kontrole, zda soubor existuje nebo nedošlo k zavření tlačítka cancel, pokud tato událost nastala funkce dále nic nevykonává a vrací chybové hlášení. V opačném případě funkce otevře soubor na dané adrese a pomocí funkce z openG knihovny dojde k načtení dat ze souboru INI. Tato funkce vrací data ve formě variantu, který je třeba převést na správný datový typ, který je návratovou hodnotou funkce. Po načtení dat dojde k zavření souboru a přetypování variantu a ukončení funkce.

V případě načítání operací nebo filtrů jsem za tuto funkci přidal ještě proces přidání záznamu do projektu, do kterého vstupují načtená data a pro jméno využívá prvek na předním panelu aplikace.

Při načtení projektu se data přímo načtou a dojde k odebrání výběru prvků v seznamech z předchozího projektu a vymazání dat v grafu.

## 5.3 Načtení dat z myDAQ

Načítání dat ze zařízení myDAQ je založeno na funkcích z knihovny DAQmx. Pomocí této knihovny lze ovládat zařízení DAQ přímo z aplikace.

Po spuštění funkce se objeví dialogové okno a dojde inicializaci počátečních hodnot. V okně se nachází prvky, které slouží k nastavení komunikace se zařízením DAQ. S tisknutím tlačítka play se spustí proces nahrávání a deaktivují se ostatní tlačítka kromě tlačítka stop. Tento proces začíná nakonfigurováním zařízení a vytvořením virtuálního komunikačního kanálu pomocí funkce Create Channel a touto funkcí následuje nastavení časového zdroje pro komunikaci, kde se nastavuje množství uložených dat v zařízení dat na jednu desetinu vzorkovací frekvence. Poté se ještě nastaví režim spouštění a dojde ke spuštění měření.

Během měření dochází každých 50 ms k výčtu dat pomocí funkce read, aby nedošlo k přetečení nastavené paměti. Data se postupně ukládají do pole průběhů, kdy dochází ke sloučení právě načtených dat s předchozími a zároveň se průběžně aktualizuje zobrazení v grafu.

Stisknutím tlačítka stop dojde k zastavení nahrávání, k aktivaci tlačítek a ukončí se úloha se zařízením DAQ. Opětovným spuštěním nahrávání dojde k přepsání předchozích hodnot.

V této funkci lze navíc nahrát předchozí nastavení, pomocí tlačítka use setting. Po jeho stisknutí se objeví okno se seznam operací daného typu, které jsou vloženy do projektu. Zde si uživatel může vybrat jednu operaci po vybrání se vrátí index, kterým další funkce najde operaci v projektu a vrací její záznam. Ze záznamu se poté vybere nastavení operace, které je z variantu převedeno na správný formát a dojde k přepsání prvků na panelu funkce.

Stisknutím tlačítka „Ok“ potvrdíte nahraná data a jejich názvy, dojde k uložení nastavení operace a signálu do úložiště, dialogové okno se ukončí a předá úložiště hlavnímu programu. Při stisknutí tlačítka „Cancel“ se okno ukončí a do hlavního programu se předá původní úložiště.

## 6. UKLÁDÁNÍ DAT

Aplikace umožňuje ukládání signálu do libovolného typu souboru, přičemž doporučený je typ TDM, kdy dochází k ukládání doplňujících informací k jednotlivým kanálům jako je jméno a popis, kdy u ostatních typů není zaručeno jejich uložení. Dále aplikace umožňuje uložit operace, filtry a celý projekt do souboru INI.

### 6.1 Uložení signálů do souboru

Ukládání signálů do souboru je opačnou funkcí pro načítání a z toho důvodu funkce využívá stejnou knihovnu Storage/DataPlugin, která umožňuje otevřít všechny datové typy, pro které má uživatel nainstalované data pluginy.

Spuštěním této funkce dojde k její inicializaci a k otevření nového okna, ve kterém jsou tři funkční tlačítka. Tlačítkem select se otevře okno, ve kterém se zobrazí seznam všech signálů. V tomto seznamu si uživatel navolí libovolný počet signálů a svůj výběr potvrdí tlačítkem OK. Funkce vrátí pole indexů vybraných signálů, pomocí kterých další funkce najde záznamy v projektu, ze kterých si načítá jména, popisy a průběh signálu, které zobrazí do grafu v okně pro uložení signálu. V tomto grafu si uživatel může průběhy přiblížit pomocí kolečka myši a pokud není s výběrem signálů spokojen může jej změnit opětovným stisknutím tlačítka select.

Stisknutím tlačítka Save se spustí proces ukládání. Nejdříve se otevře okno průzkumníka souborů, ve kterém si uživatel vybere soubor, do kterého chce data uložit. Poté dojde ke kontrole, zda nedošlo k zavření tlačítkem cancel, kdy vrátí chybové hlášení. V opačném případě se spustí funkce na otevření souboru, která sama identifikuje podle koncovky souboru, jaký data plugin je třeba použít. Další funkcí nahrajeme postupně jednotlivé signály jako kanály do souboru. Zároveň se ukládá i název a popis kanálu, avšak ne každý typ souboru umožňuje ukládání názvů a popisků. Některé typy souborů ukládají pouze průběhy. Z toho důvodu doporučuji použít soubory typu TDM, u kterých je jisté uložení všech informací. Stisknutím tlačítka OK se pouze ukončí funkce.

### 6.2 Ukládání dat do souboru INI

Tato funkce je opět velmi podobná pro načtení operace, filtru a projektu. Při volání funkce pro operace a filtry si funkce načte záznam vybraný ze seznamu na předním panelu a pro funkci na uložení projektu dojde k načtení úložiště. Dále otevře okno průzkumníka souborů, ve kterém si uživatel vybere soubor, do kterého se má prvek uložit. Poté dojde ke kontrole, zda nedošlo k zavření tlačítkem cancel, pokud tato událost nastala, funkce dále nic nevykonává a vrací chybové hlášení. V opačném případě se funkce otevře nebo vytvoří soubor na dané adrese a pomocí funkce z openG knihovny dojde k uložení daného datového formátu do souboru INI a nakonec dojde k zavření souboru a ukončení funkce.

# 7. PŘEHRÁVÁNÍ SIGNÁLŮ

Aplikace umožňuje přehrávat data pomocí reproduktorů nebo zařízení DAQ ve formátu mono a stereo.

## 7.1 Přehrávání pomocí reproduktoru

Pro přehrávání pomocí reproduktorů je použita knihovna sound/output, přehrávání probíhá jednorázovým načtením dat do zařízení a jeho následným ovládním. Při inicializaci aplikace se nastaví tlačítko play na aktivní a ostatní zešednou a jsou neaktivní.

Po stisknutí tlačítka play se tlačítko deaktivuje a aktivují se tlačítka stop, pause a replay, dále se deaktivuje control tab pro nastavování zařízení. Poté aplikace zjistí, zda již je vytvořena úloha přehrávání, pokud není, spustí se funkce ke konfiguraci zařízení. Pro tuto funkci se vyčtou data z grafu na předním panelu, dále zjistí, které signály jsou zadány, pokud není žádný dojde k chybovému hlášení. Pokud jsou zapojeny dva signály, musí mít stejnou hodnotu dt, jinak dojde k chybovému hlášení. V případě shodného dt se vezme dt jedno ze signálů. Pokud je připojen pouze jeden signál vezme se dt zadaného signálu. Z hodnoty dt se zjistí vzorkovací frekvence signálu, dále se zjistí největší společný dělitel pro vzorkovací frekvenci signálu a hodnotu vzorkovací frekvence zařízení zadanou uživatelem. Tímto dělitelem se podělí obě hodnoty a sestaví se faktor převzorkování.

Tento faktor se přivede na funkci Rational resample, která převzorkuje a dle zadaného filtru vyfiltruje zadané signály. Z těchto signálů se zjistí největší absolutní hodnota vzorku, kterou se podělí signály, čímž dojde k převedení signálu na rozsah od -1 do 1, který je typický pro wav soubory, na které je knihovna vytvořena. Dále se zjistí velikost největšího signálu a pomocí těchto hodnot dojde k nakonfigurování zařízení a zápisu upraveného signálu do zařízení. Po této funkci dojde ke zjištění času spuštění, který se uloží, dále se uloží data použitá k nakonfigurování zařízení a upravené průběhy signálů.

Pokud dojde ke stisknutí tlačítka play a již bylo zařízení nakonfigurováno dojde pouze ke spuštění přehrávání po pauze a zjištění času spuštění.

Stisknutím tlačítka pauzy dojde k jeho deaktivaci a aktivaci tlačítka play, dále se zjistí, zda došlo k dokončení přehrávání a zastaví se přehrávání. Pokud před pauzou bylo přehrávání aktivní dojde ke zjištění času po pauze a úpravě kurzoru v grafu. V opačném případě se již nic nevykonává.

Stisknutím tlačítka stop se úloha ukončí, dojde k vynulování odkazu na ni, kurzor v grafu se vrátí na počátek, hodnota minima v grafu se nastaví též na počátek, nastaví se tlačítko play na aktivní a ostatní zešednou a jsou neaktivní.

Tlačítkem replay dojde k aktivaci tlačítka pause a deaktivaci play. Dále se ukončí úloha reproduktoru, nakonfiguruje se nový pomocí uložených dat a nahrají se do zařízení uložené upravené signály, čímž dojde k přehrávání od počátku. Z toho důvodu se umístí kurzor do počátku a zjistí se čas spuštění.

Během přehrávání dochází k posunu kurzoru v grafu. Tento posun je zajištěn pomocí rozdílu času mezi jednotlivými posuny, protože knihovna neumožňuje zjistit stav přehrávání. Bohužel nelze zajistit sepnutí času ve správný moment a z toho důvodu dochází při přehrávání k mírnému zpoždění kurzoru, kdy pro třiceti sekundový signál šlo o zpoždění 0,03 s a pro signál dlouhý 236 s je zpoždění 0,1 s.

## 7.2 Přehrávání pomocí zařízení DAQ

Pro přehrávání pomocí zařízení DAQ je použita knihovna DAQmx, přehrávání probíhá postupným nahráváním dat do zařízení, kvůli zajištění nízkých nároků na paměť zařízení. Při inicializaci aplikace se nastaví tlačítko play na aktivní a ostatní zešednou a jsou neaktivní.

Po stisknutí tlačítka play se tlačítko deaktivuje a aktivují se tlačítka stop, pause a replay, dále se deaktivuje control tab pro nastavování zařízení. Poté aplikace zjistí pozici aktuálního zapisovaného vzorku, pomocí které lze zjistit v jakém stavu se přehrávání nachází. Pokud je aktuální pozice nula, dochází k novému přehrávání a je třeba signál upravit pro zařízení podobně jako u reproduktorů. Jediným rozdílem je, že po úpravě signálu na rozsah od -1 do 1 dojde navíc k posunu o polovinu rozdíl minima a maxima rozsahu zařízení a vynásobením o polovinu součtu minima a maxima. Tím zajistíme velikost signálu přesně na rozsah zařízení.

Pokud jsou připojeny dva signály, musí mít stejnou délku, jinak dojde k chybovému hlášení. V opačném případě dojde k nastavení komunikačního kanálu, časového zdroje komunikace, u kterého se nastaví paměť na 0,3 násobek vzorkovací frekvence. Dále se vyberou první vzorky dle velikosti paměti a zjistí se počet nastavených kanálů. Pokud jsou dva kanály, musí být připojeny dva signály. Pokud je nastaven jeden kanál, nejdříve se pokusí nahrát levý signál, pokud je levý signál prázdný nahraje se pravý signál. Po této funkci dojde ke zjištění aktuálního zapsaného prvku, který se uloží, dále se uloží data použitá k nakonfigurování zařízení a upravené průběhy signálů.

Pokud dojde ke stisknutí tlačítka play a hodnota aktuálního zapisovaného vzorku je větší než nula, znamená to, že je signál v průběhu přehrávání a došlo k pauze. Z uložených dat se nakonfiguruje komunikační kanál a dojde k nahrávání uložených signálů od aktuálního vzorku.

Stisknutím tlačítka pauzy dojde k jeho deaktivaci a aktivaci tlačítka play, dále kolik vzorků zbylo v zařízení, která se odečte od aktuálního zapisovaného prvku pro zjištění nové pozice tohoto prvku. Úloha se ukončí a dojde k posunutí kurzoru o počet vyčtených dat.

Stisknutím tlačítka stop se úloha ukončí a nastaví se aktuální pozice na nulu. Pozice kurzoru v grafu se vrátí do počátku, hodnota minima v grafu se nastaví též na počátek, nastaví se tlačítko play na aktivní a ostatní zešednou a jsou neaktivní.

Tlačítkem replay dojde k aktivaci tlačítka pause a deaktivaci play. Dále se ukončí úloha zařízení, nakonfiguruje se nový úloha pomocí uložených dat a nahrají se do zařízení uložené upravené signály, čímž dojde k přehrávání od počátku. Z toho důvodu se umístí kurzor do počátku a zjistí se část spuštění.

Během přehrávání dochází k posunu kurzoru v grafu a postupnému načítání dat. Dle zjištěného volného místa se vyberou části signálu, které se nahrají do zařízení. Posun je zajištěn pomocí rozdílu volného místa v zařízení od rozdílu volného místa předchozího průběhu s počtem nahraných dat. Tento způsob posunu vykazuje konstatní zpoždění pro jakoukoliv délku signálu o 0,1 s.

## 8. FUNKCE PRO PRÁCI S PROJEKTEM

V aplikaci je použito úložiště, ve kterém se nacházejí pole pro záznamy signálů, operací a filtrů. Operacemi pro práci s tímto projektem jsou přidání a odebrání záznamů jednotlivých prvků, dále změna jména a popisu jednotlivých prvků a nalezení záznamu podle indexu.

Pro přidání záznamu daného prvku je třeba, aby měl prvek unikátní název. Z toho důvodu funkce prochází pole daných prvků a kontroluje, zda jméno, které má nést nový prvek není shodné s již uloženými prvky. Pokud je shodné funkce vrací chybu, pokud není shodné, funkce zjistí ID posledního prvku a uloží záznam s ID o jedna vyšším.

Pro operaci je zde navíc podmínka unikátnosti nastavení operace kdy, pokud zjistí shodu nastavení s jinou operací, nedojde k uložení záznamu a funkce vrátí varování. Pro filtr je nutná podmínka, že nesmí dojít ke shodě ID operace, ze které je filtr vytvořen. Opět pokud ke shodě dojde filtr se neuloží a vrací varování.

Postup odebrání záznamu je shodný pro všechny prvky, funkce hledá záznam v poli podle ID. Pokud jej najde dojde k jeho odstranění z pole, pokud jej nenajde vrací chybu po neúspěšném nalezení

Funkce změna názvu a popisu si jsou podobné, obě hledají záznam v poli dle ID a pokud jej najdou, změní hodnotu příslušného atributu, pokud jej nenajdou, vrací chybové hlášení

Funkce hledání záznamu podle ID pouze hledá shodu ID záznamu se zadaným ID, pokud jej najde, vrací celý záznam, pokud nenajde, vrací chybové hlášení.

## 9. FUNKCE PRO PRÁCI SE SEZNAMY

V aplikaci vytvářím seznamy v podobě tabulky, která slouží pro výběr jednoho či více prvků a zároveň umožňuje jejich třídění dle jednoho vybraného sloupce.

### 9.1 Tvorba seznamu

Zavoláním funkce se vytvoří prázdné dvourozměrné pole, do kterého se na první řádek vloží hlavička seznamu. Pole má počet řádků o jedna větší, než je počet záznamů daného prvku a počet sloupců dle prvku, pro který se seznam tvoří. Seznam signálu má šest sloupců, kdy v prvním sloupci je ID záznamu, v druhém čas záznamu, poté název signálu, následuje název operace, ze které je pořízen, typ této operace, a nakonec popis záznamu signálu. Seznam operací má pouze čtyři sloupce kdy, v prvním sloupci je ID, následuje název operace, typ této operace, a nakonec popis záznamu operace. Seznam filtrů má šest sloupců, kdy v prvním sloupci je ID záznamu, v druhém čas záznamu, poté název filtru, název operace, ze které je pořízen, typ této operace, a nakonec popis záznamu signálu.

Poté funkce prochází jednotlivé záznamy v projektu a vkládá data do dvourozměrného pole, které je návratovou hodnotou funkce. Návratovou hodnotou je i vytvořená konstanta, která značí šířku jednotlivých sloupců tabulky.

### 9.2 Výběr prvků ze seznamu

Při zavolání této funkce dojde k nastavení šířek jednotlivých sloupců dle typu tabulky a k zvýraznění prvního řádku, kde se nachází hlavička. Podle názvů v hlavičce se nastaví hodnoty prvku pro třídění.

Ve funkci pro výběr jednoho prvku se po kliknutí do tabulky zjistí pozice vybraného políčka, pokud není z prvního řádku a je v rozměrech pole, dojde k zvýraznění daného řádku a uložení indexu, v opačné případě se nic neděje. Pokud opět klikneme na stejný řádek k dojde k odebrání zvýraznění a uložení index se zmaže. Pokud klikneme na jiný řádek dojde k odebrání zvýraznění, zvýraznění nového řádku a uložení nového indexu.

Ve funkci pro výběr více prvků se po kliknutí do tabulky zjistí pozice vybraného políčka, pokud není z prvního řádku a je v rozměrech pole, dojde k zvýraznění daného řádku a uložení indexu do pole, v opačné případě se nic neděje. Pokud opět klikneme na již zvýrazněný řádek k dojde k odebrání zvýraznění a uložení index se odebere z pole. Pokud klikneme na jiný řádek dojde k zvýraznění dalšího řádku a uložení nového indexu na konec pole. Indexy v poli jsou řazeny dle jejich vkládání.

Třídění seznamu je možné pomocí výběru sloupce a typu třídění. Při změně některé z hodnot dojde k provedení třídění. Vezme se původní netříděné pole, oddělí se hlavička od záznamů a do hlavičky se vloží symbol dle třídění. V případě vzestupného třídění dojde k třídění 2D pole dle vybraného sloupce pomocí funkce v LabVIEW, v případě sestupného třídění dojde navíc k převrácení pole pomocí funkce z openG knihovny.

Po třídění se odeberou předchozí zvýraznění a na základě nových pozic uložených ID se zjistí nové zvýraznění.



## 10. ZÁKLADNÍ OPERACE SE SIGNÁLY

Aplikace umožňuje základní operace se signály, mezi které patří matematické operace, škálování lineární funkcí, výběr podmnožiny z jednoho nebo více signálů, sloučení v zadaném pořadí, výběr každého N-tého vzorku signálu, filtrace pomocí FIR i IIR filtrů, výpočet FFT, časové zarovnání signálů. Všechny tyto operace vytváří dialogové okno se svým panelem, ve kterém je možné s danou operací pracovat.

### 10.1 Matematické operace

V okně této funkce se nachází seznam vstupních signálů, do kterého může uživatel vkládat libovolné množství signálů vytvořených v projektu stisknutím tlačítka select, kdy se otevře nové okno se seznamem signálů, ze kterého může uživatel vybírat ty, které chce použít. Kliknutím do seznamu vstupních signálů vybere prvek, se kterým může tlačítka pohybovat nahoru a dolů, případně jej smazat, pokud dojde ke smazání zůstává vybraný následující prvek, funkce si ukládá pomocná data jako je pole indexů vstupních signálů a pole průběhů těchto signálů, které se mění zároveň se seznamem vstupních signálů.

Pokud se prohodí signál umístěný v prvním řádku, dojde k přepočtu signálu, protože všechny funkce se odvíjí od prvního signálu. Přepočet také nastává přidáním či odebráním signálů neb změnou typu operace.

Aplikace umí tyto matematické operace:

- Sčítání – Přijímá pole s prvky Waveform. Funkce rozdělí pole na první průběh a pole s ostatními prvky. Nově vytvořené pole se poté prochází po prvcích pomocí for cyklu a tyto prvky jsou přičítány k prvnímu průběhu. Pokud nebyli průběhy stejné velikosti, funkce vrací chybu.
- Odčítání – Funguje podobně jako sčítání, jen od prvního průběhu odčítá zbylé signály.
- Násobení – Funguje podobně jako sčítání, jen vynásobí první průběh ostatními.
- Dělení – Funguje podobně jako sčítání, jen vydělí první průběh ostatními. Tato funkce navíc vrací chybu při dělení nulou.

Stisknutím tlačítka ok se uloží operace a výsledný signál do projektu, a pokud nenastane chyba dojde k ukončení funkce. Funkci lze také ukončit tlačítkem cancel, kdy dojde k návratu nezměněného projektu.

### 10.2 Škálování lineární funkcí

Do této funkce se vkládají průběhy pomocí tlačítka select a následným výběrem ze seznamu signálů. Funkce si opět vytvoří pomocná pole indexů vstupních signálů a průběhů. Opětovným stisknutím tlačítka select se přepíše vstupní signály. Po vybrání signálů nebo změně hodnot A nebo B funkce projde pole po prvcích pomocí for cyklu a každý průběh vynásobí hodnotou A, přičte k němu hodnotu B a výsledek zobrazí do grafu v dialogovém okně. Uživatel nastavení potvrdí stisknutím tlačítka ok, funkce uloží výsledné signály včetně operace do projektu a v případě, že nenastala chyba funkci ukončí. Stisknutím tlačítka cancel se funkce také ukončí, ale předá původní projekt.

### 10.3 Výběr podmnožiny signálu

Do této funkce stejně jako u předchozí se vkládají průběhy pomocí tlačítka select a následným výběrem ze seznamu signálů. Funkce si vytvoří pomocná pole indexů vstupních signálů a průběhů a opětovným stisknutím tlačítka select se přepíše vstupní signály.

Funkce umožňuje dva typy vývěru, jeden dle času, kdy uživatel zadá čas začátku a čas konce výběru a druhý pomocí vzorků, kdy uživatel zadá index prvního vzorku nového signálu a počet vzorků který se má vybrat.

Změnou hodnot nebo vstupních signálů se vykoná operace, která pomocné pole průběhů prochází po prvcích pomocí for cyklu a z každého průběhu vybere podmnožinu dle nastavení.

V případě výběru pomocí vzorků má každá podmnožina určené  $t_0$  vynásobením indexu prvního prvku podmnožiny hodnotou  $dt$  původního průběhu a přičtením k hodnoty  $t_0$  původního průběhu. Hodnota  $dt$  zůstává stejná.

V případě výběru pomocí času má každá podmnožina určené  $t_0$  přímo hodnotou počátku výběru, avšak výběr podmnožiny je složitější, protože musí dojít k přepočtu času na vzorky průběhu.

Uživatel nastavení potvrdí stisknutím tlačítka ok, funkce uloží výsledné signály včetně operace do projektu a v případě, že nenastala chyba funkci ukončí. Stisknutím tlačítka cancel se funkce také ukončí, ale předá původní projekt.

### 10.4 Sloučení průběhů

V okně této funkce se nachází seznam vstupních signálů, do které může uživatel vkládat libovolné množství signálů vytvořených v projektu stisknutím tlačítka select, kdy se otevře nové okno se seznamem signálů, ze kterého může uživatel vybírat ty, které chce použít. Kliknutím do seznamu vstupních signálů vybere prvek, se kterým může tlačítka pohybovat nahoru a dolů, případně jej smazat, pokud dojde ke smazání zůstává vybraný následující prvek, funkce si ukládá pomocná data jako je pole indexů vstupních signálů a pole průběhů těchto signálů, které se mění zároveň se seznamem vstupních signálů.

Pokud se prohodí signály, dojde k přepočtu výsledného signálu. Přepočet také nastává přidáním či odebráním signálů.

Funkce si pomocné pole průběhů rozdělí na první průběh a pole s ostatními prvky. Poté funkce projde po prvcích nově vytvořené pole, sloučí prvky do jednoho průběhu a zkontroluje, zda mají oba průběhy stejné  $dt$ . Pokud se  $dt$  liší proces se ukončí, vrátí chybu a prázdný průběh.

Uživatel nastavení potvrdí stisknutím tlačítka ok, funkce uloží výsledné signály včetně operace do projektu a v případě, že nenastala chyba funkci ukončí. Stisknutím tlačítka cancel se funkce také ukončí, ale předá původní projekt.

### 10.5 Výběr každého N-tého vzorku signálu

Do této funkce se opět vkládají průběhy pomocí tlačítka select a následným výběrem ze seznamu signálů. Funkce si vytvoří pomocná pole indexů vstupních signálů a průběhů a opětovným stisknutím tlačítka select se přepíše vstupní signály. Změnou signálů nebo hodnoty N se zavolá funkce vykonávající výběr vzorků.

Funkce přijímá pole průběhů, které projde pomocí for cyklu a pro každý průběh, určí velikost výsledného průběhu N-tých prvků dle čísla zadaného čísla N. Z této velikosti vytvoří průběh s nulovými hodnotami. Toto pole se předpřipraví, aby se nemuselo pole neustále přealokovávat s každým nalezeným N-tým prvkem a tím se také zvýší rychlost provedení této funkce. Poté pomocí for cyklu, který má daný počet interakcí velikostí nového pole, získám hodnotu každého N-tého prvku ze vstupního průběhu a touto hodnotou přepíše nulovou hodnotu nového průběhu. U nového průběhu se mění hodnota dt vynásobením hodnoty dt původního průběhu číslem n.

## 10.6 Filtrace pomocí FIR i IIR filtrů

Do této funkce stejně jako u předchozí se vkládají průběhy pomocí tlačítka select a následným výběrem ze seznamu signálů. Funkce si vytvoří pomocná pole indexů vstupních signálů a průběhů a opětovným stisknutím tlačítka select se přepíše vstupní signály.

Na panelu dialogového okna se nachází dále dvě ovládací tabule, jedna pro nastavení ovládání, druhá pro zobrazení průběhů filtru, jejichž stránky se mění dle příslušných enumů.

Pokud se změní kterýkoliv z ovládacích prvků pro nastavení filtru nebo dojde k výběru jiného filtru, funkce automaticky tento filtr ze zadaného nastavení vytvoří. Pokud je použit enum pro výběr zobrazení grafů znázorňujících vlastnosti filtru např. impulsní nebo přenosovou charakteristiku, funkce automaticky vykreslí tento graf. Stisknutím tlačítka Use se použije vytvořený filtr na pomocné pole průběhů a výsledek zobrazí do grafu. Pokud je uživatel s tímto výsledkem spokojený stiskne pro tlačítko Ok, tím se program ukončí dojde k uložení operace, filtru a signálů do projektu a pokud nenastala při ukládání chyba tak i ukončí funkci. Stisknutím tlačítka cancel se funkce ukončí a vrátí původní projekt.

## 10.7 Výpočet FFT

V této funkci opět dochází k výběru pomocí tlačítka select, které otevírá okno se seznamem signálů, ze kterého může uživatel vybírat a po výběru se opět vygenerují pomocná pole. Dále lze v dialogovém okně nastavit typ okna a způsob průměrování, které se má použít pro výpočet FFT, a také nastavení výsledného zobrazení. Pro nastavení zobrazení jsou zde tři ovládací prvky, jeden určuje, zda má být amplituda výsledné FFT udávána v decibelech nebo ne, další, zda chci mít fázi danou ve stupních nebo v radiánech, a poslední, zda má být fáze rozvinutá nebo ne.

Pro výpočet je použita funkce FFT Mag Phase z palety Waveforms Measurements a výsledná amplitudová a fázová spektra jsou zobrazena a převedena do průběhů typu Waveform a zobrazena do grafů.

## 10.8 Časové zarovnání signálů

Ve funkci se opět volí vstupní signály tlačítkem select a výběrem ze seznamu. Poté je dle počtu průběhů inicializováno pole z ovládacími prvky typu bool. Pomocí těchto ovládacích prvků lze vybírat, kterým průběhem chce uživatel manipulovat. Posuv v časové oblasti je ovládán kolečkem myši, kdy tato akce je přijímána Event strukturou a v té se změní hodnota t0 aktivních průběhů o hodnotu parametru Set delay zadanou na

panelu dialogového okna. Po stisknutí tlačítka Ok se funkce ořeže signály tak, aby byly zarovnány jejich počátky a konce a nastaví jejich  $t_0$  na hodnotu 0. Tyto průběhy a nastavení operace se uloží do projektu a pokud nenastala chyba dojde u ukončení funkce Stisknutím tlačítka Cancel, funkce vrátí původní projekt.

# 11. TESTOVÁNÍ A DOKUMENTACE PRÁCE

V průběhu tvorby práce došlo k postupnému testování jednotlivých prvků aplikace na vytvořených pomocných testovacích VI, kde došlo k postupnému odladování těchto funkcí do správné formy. Nejvíce času jsem strávil při odladování přehrávání signálu pomocí zařízení myDAQ, kdy při zastavení úlohy nedocházelo k resetování zapisované pozice a zařízení po opětovném spuštění, čekalo, než dojde zařízení na tuto pozici. Poté přehrávání dále pokračovalo bez problémů. Kvůli této chybě muselo dojít ke změně, kdy při zastavení úlohy dochází k jejímu ukončení místo pozastavení. A při spuštění se musí úloha opět nakonfigurovat, místo pouhého spuštění. Tento problém by též mohlo řešit vytvoření druhého kanálu z čítače zařízení, který sloužil jako časový zdroj samotného přehrávání. Zde by docházelo k zastavování a spouštění časového zdroje, zdá se tento způsob jako lepším řešením, avšak narazil jsem na problém, kdy zařízení myDAQ neumí připojit svůj čítat jako časový zdroj.

Při testování funkcí pro sloučení signálu, byli testovány 3 verze první využívala dynamického pole, které se v průběhu slučování postupně zvětšovalo. Druhá verze si nejdříve určila velikost pole ze součtu velikostí vstupních průběhů, z této velikosti poté vytvořila pole s hodnotou prvků 0 a poté následně pouze přepisovala tyto prvky. Jako poslední verzi jsem vyzkoušel již připravenou funkci v LabVIEW. Z výsledků testování rychlosti provedení sloučení tisíce vzorků, jsem došel k závěru, že verze jedna je nejrychlejší a zvládá sloučit tisíc signálů za jednu tisícinu sekundy, třikrát pomalejší je verze tři, která tuto operaci zvládla za tři tisíciny sekundy a nejpomalejší funkcí byla verze dva, která tuto operaci zvládla za tři desetiny sekundy, díky těmto výsledkům byli ve finální aplikaci využita právě verze jedna.

Podobné testování probíhalo u tvorby výběru každého N prvku, kdy první verze využívala dynamické pole, do kterého se postupně přidávali prvky, a druhá verze si nejdříve vypočítala velikost výsledného signálu, který si inicializovala na hodnoty 0 a poté tyto hodnoty přepisovala vybranými vzorky. Zde se ukázal opačný výsledek, kdy verze jedna s dynamickým polem byla pomalejší než verze dva. Pro výběr každého desátého vzorku z tisíce signálu o délce tisíc vzorků, byla doba běhu verze jedna rovna čtyřem setinám sekundy a pro verzi dva necelá jedna setina.

Dále bylo provedeno testování celkové aplikace, avšak na detailní testování, kdy by došlo k detekci chyb při nepředvídatelném chování uživatele, již nebylo dostatek času, proto je doporučeno tuto testování ještě provést.

Dále byli v průběhu implementace vytvářeny ikony jednotlivých funkcí, které by měli být schopny popsat účel dané funkce. Také došlo k postupné tvorbě popisů vysvětlující funkci daného kódu a doplnění popisu k jednotlivých funkcí, bohužel místy je použita diakritika, a i při nainstalování českého jazyka do LabVIEW se může vyskytnout problém se správnou interpretací textu.

## 12. ZÁVĚR

V teoretickém rozboru jsem se zabýval možnostmi prostředí LabVIEW v oblasti vizualizace a přehrávání dat. V závěru této kapitoly jsem došel k tomu, že pro naši aplikaci bude nejhodnější využívat grafů typu Waveform graf, ve výjimečných případech XY graf a pro zobrazení trojrozměrných signálů použiji intenzitní graf z knihovny Sound and Vibration. Pro trojrozměrná data bych mohl použít také 3D grafy, avšak ty jsou v LabVIEW nepřehledná a špatně se s nimi pracuje.

Dále jsem zjistil, že funkce z palety Sound jsou ve většině případů také nevhodné, využiji je pouze pro práci s reproduktorem. Pro načítání souborů jsou funkce nevhodné, protože je lze použít pouze na soubory ve formátu WAV, a proto místo nich použiji funkce pracující s data pluginy, čímž zajistím možnost, načítat data z formátů, pro které jsou vytvořeny data pluginy a mám je nainstalované v počítači. Funkce pracující s mikrofonom také nevyžiji, protože data budou nahrávána pomocí zařízení myDAQ. Pro ovládání zařízení myDAQ použiji funkce z knihovny DAQmx.

V teoretickém rozboru jsem rovněž provedl průzkum programů, poskytujících podobné funkce. Zde jsem vybral tři zástupce, přičemž nejpodobnějším naší aplikaci je program DIAdem, kterým se budu nejvíce inspirovat.

Dle tohoto vzoru chci do práce zakomponovat přehled aktuálně otevřených souborů dat, s kterými lze pracovat. Dále bych se chtěl inspirovat v rozdělení operací se signály do palet. Rovněž se chci inspirovat v ovládání tohoto programu, především možnosti posunu grafu a posunu začátku přehrávání pomocí myši, dále se zde budou nacházet velmi podobná funkční tlačítka. Funkční tlačítka jsou rovněž podobná těm v programu DADiSP, avšak proti němu jsou zde navíc tlačítka pro přehrávání.

Vzhledem k požadavku na přehrávání signálu, chci též uživateli umožnit nastavit začátek a konec přehrávání. Tyto možnosti přehrávání jsou nejlépe řešeny v programu Audacity a proto se zde budu inspirovat právě ním. Dle Audacity přidám tlačítka pro ovládání přehrávání a nahrávání průběhu, rovněž budu zobrazovat průběh přehrávání. Dále by bylo vhodné z Audacity využít funkce, které umožňují upravovat velikost jednotlivých vzorků.

U průzkumu jsem zjistil, že nejpoužívanější typy souborů jsou TDMS, MAT, CSV, WAV a DAT a ty by měl podporovat náš program. Toho docílím použitím funkcí pro práci s data pluginy.

Aplikace umožňuje základní operace se signály, mezi které patří matematické operace, škálování lineární funkcí, výběr podmnožiny z jednoho nebo více signálů, sloučení v zadaném pořadí, výběr každého N-tého vzorku signálu, filtrace pomocí FIR i IIR filtrů, výpočet FFT, časové zarovnání signálů. Všechny tyto operace vytváří dialogové okno se svým panelem, ve kterém je možné s danou operací pracovat. Dále umožňuje načítání dat z a do zařízení myDAQ, přehrávání pomocí signálu pomocí reproduktorů, kdy při přehrávání dochází ke zobrazování aktuální pozice.

Během tvorby aplikace docházelo k testování jednotlivých funkcí i samotné aplikace a také k dokumentaci jednotlivých částí kódu.

## Literatura

- [1] Graph and Chart Controls and Indicators, 2018. *NATIONAL INSTRUMENTS* [online]. NATIONAL INSTRUMENTS [cit. 2020-12-29]. Dostupné z: [https://zone.ni.com/reference/en-XX/help/371361R-01/lvconcepts/types\\_of\\_graphs\\_and\\_charts/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvconcepts/types_of_graphs_and_charts/)
- [2] Sound VIs, 2018. *NATIONAL INSTRUMENTS* [online]. NATIONAL INSTRUMENTS [cit. 2020-12-29]. Dostupné z: [https://zone.ni.com/reference/en-XX/help/371361R-01/lvconcepts/types\\_of\\_graphs\\_and\\_charts/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvconcepts/types_of_graphs_and_charts/)
- [3] DIAdem, 2020. *NATIONAL INSTRUMENTS* [online]. NATIONAL INSTRUMENTS [cit. 2020-12-29]. Dostupné z: <https://www.ni.com/cs-cz/shop/software/products/diadem.html>
- [4] DataPlugins, 2020. *NATIONAL INSTRUMENTS* [online]. NATIONAL INSTRUMENTS [cit. 2020-12-29]. Dostupné z: <https://search.ni.com/nisearch/app/main/p/ap/tech/lang/cs/pg/1/sn/ssnav:dpl/>
- [5] *Audacity* [online], 2020. Audacity [cit. 2020-12-29]. Dostupné z: <https://www.audacityteam.org/>
- [6] *DADiSP the Ultimate engineer Spreadsheet* [online], 2020. DSP Development Corporation [cit. 2020-12-29]. Dostupné z: <https://www.dadisp.com/>

# Seznam příloh

Příloha A - LabVIEW projekt s aplikací a jejími funkcemi - Zdrojový kód je uložen na příloženém CD.