



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

SOFTWAREVÝ INTERFACE PRO PRÁCI S DATABÁZOVÝMI SOUBORY

SOFTWARE INTERFACE FOR DATABASE FILES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Adam Novák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Uher

BRNO 2019



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Adam Novák

ID: 174366

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Softwarový interface pro práci s databázovými soubory

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je kompletní realizace obousměrného softwarového propojení pro práci s databázovými soubory založenými na formátu MS Access 2.0 (*.mdb). Uvedené databázové soubory mají známou strukturu a představují soubory s kompletními daty o testu na vibračním testovacím zařízení Zkušební laboratoře CVVOZE. Úkolem interface bude zajištění aplikačního přístupu k datům zkoušky, které jsou v této databázi uloženy. Realizace kódu je předpokládána ve vývojovém prostředí MS Visual C#/C++ a očekává se taktéž odevzdání příslušné dokumentace.

Předpokládaná osnova BP:

1. Zpracujte průzkum možností přístupu k obsahu databázových souborů založených na formátu MS Access 2.0 (*.mdb).
2. Dokumentujte databázové soubory, které obsahují kompletní data o testech na vibračním testovacím zařízení firmy RMS umístěném ve ZL CVVOZE v rozsahu zkoušek zadaných vedoucím práce.
3. Navrhněte strukturu vlastního obousměrného softwarového rozhraní pro práci s výše uvedenými databázovými soubory.
4. Navržený softwarový interface z bodu 3 zadání realizujte a otestujte jeho správnou funkcionalitu.

DOPORUČENÁ LITERATURA:

1. Bai, Y.: Practical Database Programming With Visual C#.NET. ISBN: 978-0-470-46727-5. 936 s. 2010, Wiley-IEEE Press.
2. další literatura dle pokynů vedoucího

Termín zadání: 4.2.2019

Termín odevzdání: 20.5.2019

Vedoucí práce: Ing. Miroslav Uher

Konzultant:

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

UPOZORNĚNÍ:

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

Abstrakt

Bakalářská práce se zabývá způsobem jak přistupovat k databázovým souborům s příponou (*.shk), (*.sin) a (*.rau) postavené na základu Microsoft Acces 2.0. Dokumentací těchto souborů a vytvoření softwarového interfacu. Interface vytvoří dané typy databázových souborů. Nahraje do databázových souborů nastavení pro měření na zařízením od firmy RMS. Po měření vytáhne z databáze data, která se můžou pak dále vyhodnocovat.

Klíčová slova

ODBC, C++, SQL, MDB, SW interface, Microsoft Acces 2.0, Microsoft Visual 2015+

Abstract

The bachelor thesis deals with the way how to access database files with the extension (*.shk), (*.sin) and (*.rau) based on Microsoft Acces 2.0. Document these files and create a software interface. The interface creates the given types of database files. Uploads measurement settings to RMS device measurement files. After the measurement, it extracts data from the database, which can then be further evaluated

Keywords

ODBC, C++, SQL, MDB, SW interface, Microsoft Acces 2.0, Microsoft Visual 2015+

Bibliografická citace:

NOVÁK, Adam. *Softwarový interface pro práci s databázovými soubory*. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/119271>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miroslav Uher.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma „Softwarový interface pro práci s databázovými soubory“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této diplomové bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne:

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Miroslavu Uhrovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne:

.....
podpis autora

Obsah

1.	Úvod.....	12
2.	Použité vývojové nástroje	13
2.1	Volba jazyka.....	13
2.2	Databáze	13
2.3	ODBC.....	13
3.	Práce s databází MDB v c++.....	14
3.1	Vytvoření databázového souboru MDB.....	14
3.2	Připojení k databázovému souboru	15
3.3	Vytvoření tabulky a zápis dat.....	15
3.4	Čtení dat ze souboru.....	16
3.5	Přepisování dat v souboru	16
4.	sinusové vibrace	17
4.1	Základní funkce pro sinusové vibrace.....	17
4.1.1	Vytvoření databázového souboru pro sinusové vibrace	17
4.1.2	Nahrání dat do databázového souboru pro sinusové vibrace.....	17
4.1.3	Čtení dat z databázového souboru pro sinusové vibrace	18
4.2	Tabulka TargetCurve pro sinusové vibrace	19
4.3	Tabulka SystemDataText pro sinusové vibrace	21
4.4	Tabulka SystemDataFloat pro sinusové vibrace	21
4.4.1	Karta Windows Settings	22
4.4.2	Karta Input Channels pro sinusové vibrace	23
4.4.3	Karta Notching pro sinusové vibrace.....	26
4.4.4	Karta Control Parameters pro sinusové vibrace	27
4.5	Tabulka Logbook pro sinusové vibrace	29
4.6	Tabulka GraphStyle pro sinusové vibrace	29
4.7	Tabulka Drive pro sinusové vibrace	31
4.8	Tabulka DiagrammStyleText pro sinusové vibrace.....	31
4.9	Tabulka DiagrammStyleFloat pro sinusové vibrace	32
5.	Náhodné Vibrace.....	33
5.1	Základní funkce pro náhodné vibrace	33
5.1.1	Vytvoření databázové souboru pro náhodné vibrace.....	33

5.1.2	Přepis dat v databázovém souboru pro náhodné vibrace	33
5.1.3	Čtení dat z databázového souboru pro náhodné vibrace.....	33
5.2	Tabulka TargetCurve pro náhodné vibrace.....	35
5.3	Tabulka SystemDataText pro náhodné vibrace	36
5.4	Tabulka SystemDataFloat pro náhodné vibrace	36
5.4.1	Karta Input Channels pro náhodné vibrace.....	37
5.4.2	Karta Set Level Table pro náhodné vibrace.....	39
5.4.3	Karta Control Parameters pro náhodné vibrace	41
5.5	Tabulka Logbook, Drive, DiagrammStyleText a DiagrammStyleFloat pro náhodné vibrace	43
5.6	Tabulka GraphStyle pro náhodné vibrace.....	43
6.	Rázy.....	44
6.1	Základní funkce pro rázy	44
6.1.1	Vytvoření databázového souboru pro rázy	44
6.1.2	Přepis dat do databázového souboru pro rázy	44
6.1.3	Čtení dat z databázového souboru pro rázy	44
6.2	Tabulky UserToleranceband a UserCurveType pro rázy.....	46
6.3	Tabulka User_Pre_PostPulses pro rázy	46
6.4	Tabulka TargetCurve pro rázy	47
6.5	Tabulka SystemDataText pro rázy	47
6.6	Tabulka SystemDataFloat pro rázy	48
6.6.1	Karta Input Channels pro rázy	48
6.6.2	Karta Control Parameters pro rázy	51
6.6.3	Karta Reference Curve pro rázy	52
6.7	Tabulky Logbook, Drive, DiagrammStyleText, DiagrammStyleFloat a GraphStyle pro rázy	53
	Závěr	54

Seznam obrázků

Obrázek 3.1 Okno Properties projektu ve VS 2015+	14
Obrázek 3.2 Funkce na vytvoření databázového souboru	15
Obrázek 3.3 Připojení k databázovému souboru	15
Obrázek 3.4 Funkce vytvoření tabulky a vkládání dat	15
Obrázek 3.5 Funkce čtení dat z databázového souboru.....	16
Obrázek 3.6 Funkce na přepisování dat v databázovém souboru.....	16
Obrázek 4.1 Screen karty Reference Curve pro sinusové vibrace.....	20
Obrázek 4.2 Screen karty Window Settings -> Output	22
Obrázek 4.3 Screen karty Input Channels pro sinusové vibrace	23
Obrázek 4.4 Screen podkarty Notching pro sinusové vibrace.....	26
Obrázek 4.5 Screen karty Control Parameters pro sinusové vibrace.....	27
Obrázek 4.6 Screen karty Graph Window pro sinusové vibrace	30
Obrázek 4.7 Screen karty Graph Window pro sinusové vibrace	32
Obrázek 5.1 Screen karty Random Reference Spectrum pro náhodné vibrace	35
Obrázek 5.2 Screen karty Input Channels pro náhodné vibrace.....	37
Obrázek 5.3 Screen karty Set Level Table pro náhodné vibrace.....	41
Obrázek 5.4 Screen karty Control Parameters pro náhodné vibrace	42
Obrázek 6.1 Screen karty Reference Curve pro rázy.....	46
Obrázek 6.2 Screen karty Input Channels pro rázy	48
Obrázek 6.3 Screen karty Control Parameters pro rázy.....	51

Seznam tabulek

Tabulka 4.1 Způsob ukládání řádků pro proměnnou z funkce na čtení řádku.....	18
Tabulka 4.2 Seznam proměnných struktury NastavenaData pro sinusové vibrace ..	19
Tabulka 4.3 Popis tabulky TargetCurve pro sinusové vibrace	20
Tabulka 4.4 Vzorce pro přepočítání v tabulce TargetCurve pro sinusové vibrace.....	21
Tabulka 4.5 Popis dat z karty Input Channels -> Info Text pro sinusové vibrace....	21
Tabulka 4.6 Popis dat z karty Window Settings.....	22
Tabulka 4.7 Přehled veličiny Type of Input pro sinusové vibrace	23
Tabulka 4.8 Přehled veličiny Sensitivity pro sinusové vibrace	24
Tabulka 4.9 Přehled veličiny Units pro sinusové vibrace	24
Tabulka 4.10 Přehled veličiny Limits pro Type of Input = Measure a Control pro sinusové vibrace.....	24
Tabulka 4.11 Přehled veličiny Limits pro Type of Input = Abort pro sinusové vibrace.....	25
Tabulka 4.12 Přehled metod pro kartu Input Channels pro sinusové vibrace	25
Tabulka 4.13 Seznam veličin pro Notching pouze Channel 8.....	26
Tabulka 4.14 Metody pro Notching pouze pro Channel 8	27
Tabulka 4.15 Přehled veličin z karty Control Parameters a seznam metod pro sinusové vibrace.....	28
Tabulka 4.16 Možných kombinací a hodnoty pro řádek 212	28
Tabulka 4.17 Možných kombinací a hodnoty pro řádek 211	29
Tabulka 4.18 Ukázka ukládání dat Graph Window a Graph Print	30
Tabulka 4.19 Pořadí v jakém se ukládají hodnoty do databázového souboru.....	31
Tabulka 4.20 Kompletní ukázka tabulky DiagrammStyleText z DB.....	31
Tabulka 5.1 Seznam proměnných struktury NastavenaData pro náhodné vibrace ..	34
Tabulka 5.2 Popis dat z tabulky TargetCurve pro náhodné vibrace.....	35
Tabulka 5.3 Vzorce pro přepočítání v tabulce TargetCurve pro náhodné vibrace	36
Tabulka 5.4 Přehled veličiny Info Text a její metody pro náhodné vibrace.....	36
Tabulka 5.5 Přehled veličiny Type of Input pro náhodné vibrace.....	37
Tabulka 5.6 Přehled veličiny Sensitivity pro náhodné vibrace.....	38
Tabulka 5.7 Přehled veličiny Limits pro Type of Input = Measure a Control u náhodných vibrací.....	38

Tabulka 5.8 Přehled veličiny Limits pro Type of Input = Abort u náhodných vibrací	39
Tabulka 5.9 Přehled metod pro kartu Input Channels u náhodných vibrací.....	39
Tabulka 5.10 Přehled veličin z karty Set Level Table a jejich metody pro náhodné víbrace.....	40
Tabulka 5.11 Přehled veličin z karty Control Parameters pro náhodné vibrace.....	42
Tabulka 5.12 Seznam hodnot pro Curve Save na indexu 147	43
Tabulka 6.1 Seznam proměnných struktury NastavenaData pro rázy	45
Tabulka 6.2 Popis dat Pre/-Post Shock z karty Reference Curve pro rázy.....	47
Tabulka 6.3 Přehled veličiny Info Text a její metody pro rázy	47
Tabulka 6.4 Přehled veličiny Type of Input pro rázy	49
Tabulka 6.5 Přehled veličiny Sensitivity pro rázy	49
Tabulka 6.6 Přehled veličiny Limits pro Type of Input = Measure a Control pro rázy	49
Tabulka 6.7 Přehled veličiny Limits pro Type of Input = Abort pro rázy.....	50
Tabulka 6.8 Přehled metod pro kartu Input Channels pro rázy	50
Tabulka 6.9 Přehled veličin z karty Control Parameters pro rázy	52
Tabulka 6.10 Seznam hodnot pro zatržítka Curve save z karty Control Parameters pro rázy	52
Tabulka 6.11 Přehled veličin pro kartu Reference Curve pro rázy.....	53
Tabulka 6.12 Seznam hodnot pro veličinu Standard a Curve-Type pro rázy.....	53

1. ÚVOD

V této bakalářské práci se chci zabývat analýzou databázových souborů, které jsou postavené na formátu MS Access a používají se pro nastavení měřicího vibračního testovacího zařízení od firmy RMS ve zkušební laboratoři CVVOZE. Budu zkoumat metody, jakým způsobem se dá k těmto souborům přistupovat pomocí programu napsaném v jazyce C++ a posléze procovat s daty z databázových souborů.

Před započítím této práce jsem musel nastudovat programovací jazyk SQL [2] a najít způsob, jak propojit jazyk SQL s jazykem C++.

Dalším úkolem je dokumentovat 3 databázové soubory pro tři druhy vibrací, protože dodaná dokumentace databázových souborů k zařízení od firmy RMS je naprosto minimální. Tato dokumentace se skládá jen z vyscreenovaných tabulek bez popisu. Pro rázy, sinusové a náhodné vibrace. Z této dokumentace bude vycházet softwarový interface, který se chystám zrealizovat.

Softwarový interface by měl umět vytvořit databázový soubor pro daný typ vibrací. Po vytvoření takového souboru, by se měly dát přehrát data pro požadované měření. Po doměření na vibračním stolku by se opět otevřel databázový soubor a vyčetl by naměřená a nastavená data pro další zpracování. Toto zpracování dat nebo grafické rozhraní už tento softwarový interface nebude řešit.

Softwarový interface by měl zrychlit a zefektivnit práci na tomto vibračním zařízením tím, že se bude moci nastavit databázový soubor ze vzdáleného pracoviště.

2. POUŽITÉ VÝVOJOVÉ NÁSTROJE

2.1 Volba jazyka

Interface pro práci s databázovými soubory typu (*.mdb) jsem měl možnost psát v C++ nebo C#. Zvolil jsem C++ a to z důvodů, že jsem měl v C++ základní znalosti.

Výhoda jazyka C++ je, že vytváří přímo spustitelný kód a uživatel k tomu nepotřebuje žádné enginy. C# je přívětivější k uživateli a jednodušeji se s ním operuje s pamětí.

2.2 Databáze

Databáze jsou pro dnešní civilizaci skoro až nezbytným nástrojem ve všech oborech. Používají se k evidenci občanů, užívání a předepisování jejich léku, jako je vládní databáze "eRecept", nebo sítí mobilních telefonů. Pomocí automatizace a moderní techniky jsou sbírána data ze všech možných vědních oborů a ukládána do databází.

Databáze je systém dat s pevnou strukturou záznamu, která v dnešní době nahrazují papírové kartotéky. Dnešní databáze jsou ale mnohem složitější, dokážou různá data mezi sebou provazovat a spojovat, filtrovat, třídít a rychle zobrazovat. Záznamy mají k sobě přiřazený unikátní klíč, podle kterého se řadí a rozlišují. Data, která jsou ukládána do databází, jsou vyzorované nebo změřené vlastnosti zkoumaného objektu. Když tyto data správně spojíme, získáme informaci o zkoumaném objektu a tyto informace můžeme pak použít například k regulaci daného objektu. Přípona MDB, pro kterou budu vytvářet interface, vznikla z názvu Microsoft Access Database a vytváří se pomocí programu Microsoft Access.[10]

2.3 ODBC

Je celá řada datových a databázových souborů. Každý druh takového souboru většinou používá pro ukládání dat svůj vlastní formát a metodu přístupu. Aby programátoři aplikací nemuseli čelit různorodosti takových zdrojů dat, byl vyvinut standardní nástroj ODBC, který přistupuje k různým datovým souborům pomocí standardizované platformy. ODBC je zkratka Open Database Connectivity. Tento nástroj jsem si vybral kvůli jeho kompatibilitě. Ostatní nástroje pro práci s (*.mdb) ve většině případů stejně vychází z tohoto standardu.[8]

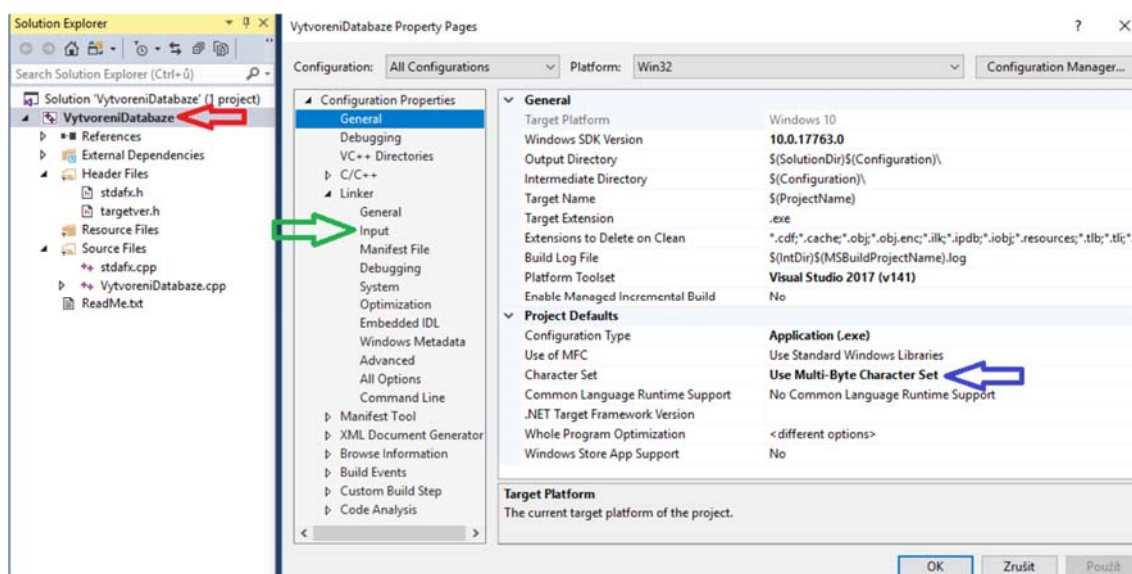
3. PRÁCE S DATABÁZÍ MDB V C++

V následující kapitole se budu zabývat základními úkony pro práci s databází v C++ pomocí ODBC v prostředí VisualStudio 2015+.

Po vytvoření konzolového projektu popíšu, jak nastavit VS, aby knihovna ODBC fungovala správně. Klikneme **pravým tlačítkem** na **název projektu**, jak je znázorněno červenou šipkou na obrázku 3.1 a vybereme možnost **Properties**. V okně, které se nám otevřelo vybereme **Configuration Properties** -> **General** a v oddílu **project Defaults** nastavíme řádek **Character Set** na hodnotu: „**Use Multi-Byte Character Set**“. Opět se můžeme podívat na obrázek 3.1, kde to značí modrá šipka.

Další krok je nahrání knihovny. Přejdeme do karty **Linker**, která je na obrázku 3.1 u zelené šipky. Rozbalíme si **Input** a na řádku **Additional Dependencies** vpravo rozbalíme menu a dáme **Edit**. Otevře se nám okno a vložíme do horní buňky text: „**legacy_stdio_definitions.lib**“, stiskneme OK a znovu OK v okně pro Properties, které se nám tím zavře.

Pro práci s databází typu MDB nalinkujeme knihovny: `<sql.h>`, `<sqltypes.h>`, `<sqlext.h>`, `<odbcinst.h>` a `<windows.h>`.



Obrázek 3.1 Okno Properties projektu ve VS 2015+

3.1 Vytvoření databázového souboru MDB

Pro vytvoření databázového souboru použijeme funkci `SQLConfigDataSource`, kterou můžeme vidět na obrázku 3.2. Na 16. řádce je možné nastavit jaký typ souboru budeme vytvářet, jestli starší verzi typu (*.mdb) nebo novější typ (*.accdb). Na 17. řádce se zadává za rovnítko cesta a název souboru. Ostatní parametry ponecháme nezměněné.

```

13     SQLConfigDataSource(
14         0,
15         ODBC_ADD_DSN,
16         "Microsoft Access Driver (*.mdb)",
17         "CREATE_DB=C:\\A\\Testdata.mdb"
18     );

```

Obrázek 3.2 Funkce na vytvoření databázového souboru

3.2 Připojení k databázovému souboru

Připojení k databázovému souboru je znázorněno na obrázku 3.3. Na začátku kódu vidíme tvorbu proměnných, které potřebujeme pro připojení k databázi. Nejpodstatnější pro nás je typ **SQLHSTMT** na řádce 24, který představuje takový kurzor nebo „ukazatel“, kde se v databázi nacházíme a tedy kde se budou provádět námi zadané příkazy. Dále je pro nás podstatný řádek 34 a 35, kde se nachází funkce **SQLDriverConnect** [6], která obsahuje typ databázového souboru a cestu s názvem. Tyto informace se zadávají pomocí řetězce. Mezi řádky 37 a 39 píšeme funkce, které chceme s danou databází provádět a od řádku 40 jsou příkazy, kterými se od databáze odpojíme.

```

22     SQLHENV henv;
23     SQLHDBC hdbc;
24     SQLHSTMT hstmt = 0;
25     SQLCHAR OutConnStr[255];
26     SQLSMALLINT OutConnStrLen;
27     HWND desktopHandle = GetDesktopWindow();
28
29     SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv);
30     SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0);
31     SQLAllocHandle(SQL_HANDLE_DBC, henv, &hdbc);
32     SQLSetConnectAttr(hdbc, SQL_LOGIN_TIMEOUT, (SQLPOINTER)5, 0);
33
34     SQLDriverConnect(hdbc, desktopHandle, (SQLCHAR*)"Driver={Microsoft Access Driver (*.mdb)}; DBQ=C:\\A\\Testdata.mdb",
35         SQL_NTS, OutConnStr, 255, &OutConnStrLen, SQL_DRIVER_NOPROMPT);
36     SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt);
37     /*****
38
39     /*****
40     SQLFreeHandle(SQL_HANDLE_STMT, hstmt);
41     SQLDisconnect(hdbc);
42     SQLFreeHandle(SQL_HANDLE_DBC, hdbc);
43     SQLFreeHandle(SQL_HANDLE_ENV, henv);

```

Obrázek 3.3 Připojení k databázovému souboru

3.3 Vytvoření tabulky a zápis dat

Funkce **SQLExecDirect** [5] se používá k provádění SQL příkazů. Jako je vytvoření, kopírování, mazání tabulek, práce s relacemi a další příkazy SQL. Na obrázku 3.4 na řádce 55 můžeme vidět, jakým způsobem se vytváří tabulka se dvěma sloupci typu **float** a **int**. Na řádce 56 je pak vložen jeden řádek s hodnotami. Do prvního sloupce se uloží 500 a do druhého se uloží hodnota 1000. *hstmt* je kurzor, který ukazuje na otevřenou databázi.

```

55     SQLExecDirect(hstmt, (SQLCHAR*)"CREATE TABLE Tabulka_1 (Sloupec_1 REAL, Sloupec_2 INTEGER)", SQL_NTS);
56     SQLExecDirect(hstmt, (SQLCHAR*)"INSERT INTO Tabulka_1 VALUES(500, 1000)", SQL_NTS);

```

Obrázek 3.4 Funkce vytvoření tabulky a vkládání dat

3.4 Čtení dat ze souboru

Pro čtení použijeme funkci **SQLGetData** [4], ale ještě předtím musíme vybrat odkud budeme data číst. Opět použijeme **SQLExecDirect**, kde si pomocí SQL příkazů **SELECT * FROM** určíme z jaké tabulky budeme číst a uložíme to do kurzoru *hstmt*. Kdybych chtěl příkazem vybrat i konkrétní sloupečky napsal bych místo hvězdičky názvy těchto sloupečků a odděloval bych je čárkou. Daný příkazový SQL řetězec můžeme ještě rozšířit o slovo **WHERE**, které slouží k vyhledávání konkrétních dat. Příklad na obrázku 3.5 na řádce 50.

Využijeme funkci **SQLFetch**, kterou budeme načítat řádek pořádku a parametr bude mít kurzor, který v sobě bude mít uloženo odkud se mají řádky načítat. Nakonec využijeme funkci **SQLGetData**, která má parametry kurzoru. Další parametr je z kolikátého sloupečku chceme hodnotu přečíst, typ čtené hodnoty, místo kam se uloží, kolik hodnot chceme přečíst. U číselných hodnot necháme parametr na 1. V posledním parametru necháme hodnotu *NULL*. Příklad na obrázku 3.5 mezi řádky 44 a 49.

```
43  
44     SQLExecDirect(hstmt, (SQLCHAR*)"SELECT * FROM NaseTabulka", SQL_NTS);  
45     int i = 0;  
46     while(SQLFetch(hstmt) == SQL_SUCCESS) {  
47         SQLGetData(hstmt, 1, SQL_C_DOUBLE, &Vek, 1, NULL);  
48         SQLGetData(hstmt, 3, SQL_C_UBIGINT, &Telefon, 1, NULL);  
49     }  
50     SQLExecDirect(hstmt, (SQLCHAR*)"SELECT Telefon, Adresa FROM NaseTabulka WHERE NaseTabulka.[Vek] = 18", SQL_NTS);  
51
```

Obrázek 3.5 Funkce čtení dat z databázového souboru

3.5 Přepisování dat v souboru

I tady použijeme funkci **SQLExecDirect**, kde použijeme příkaz **UPDATE**, který určuje v jaké tabulce se budeme pohybovat. Příkazem **SET** zadáme, jaká data chceme změnit a přiřadíme jim určitou hodnotu. Příkazem **WHERE** řekneme na jakých řádcích chceme data změnit.

```
39  
40     SQLExecDirect(hstmt, (SQLCHAR*)"UPDATE Tabulka Set Slupec = 5 WHERE Tabulka.[Index] = 2", SQL_NTS);  
41
```

Obrázek 3.6 Funkce na přepisování dat v databázovém souboru

4. SINUSOVÉ VIBRACE

Zde popíši nastavení a data v databázovém souboru pro sinusové vibrace pro testovací zařízení zkušební laboratoře CVVOZE. Dále vysvětlím funkce a metody, které slouží pro vytvoření, přepis a čtení dat v databázovém souboru (*.sin).

Databázový soubor pro sinusové vibrace s příponou (*.sin) se skládá z 8 tabulek pro nastavení vibračního zařízení a 1 až 99 tabulek s naměřenými daty. Tyto tabulky se značí Data_1 až Data_99. Tabulky nastavení vibračního zařízení jsou TargetCurve, SystemDataText, SystemDataFloat, Logbook, GraphStyle, Drive, DiagrammStyleText a DiagrammStyleFloat.

4.1 Základní funkce pro sinusové vibrace

Pro práci s databázovým souborem (*.sin) potřebujeme hlavičkové soubory *SinHeader.h*, *SinFunkce.cpp* a *SinZaklad.sin*, které nalezneme v přílohách. Soubor *SinZaklad.sin* je přednastavený databázový soubor, ve kterém přepisují data. Odpadá nám tedy zakládání tabulek, sloupečků a plnění databázové souboru základními daty, které jsem naprogramoval pro sinusové vibrace. Toto náhradní řešení jsem musel použít, protože ač softwarový interface dokáže samozřejmě nový soubor databáze vytvořit, bohužel tento soubor nedokáže v originálním SW pracovat. Nejpravděpodobnější příčinou tohoto problému bude skutečnost, že knihovna ODBC nedokáže vytvářet úplně identické databáze jako tomu bylo v MS Access 2.0 z roku 1994 a řídicí software vibračního systému bohužel reaguje na tuto změnu formátu chybovou funkčností.

4.1.1 Vytvoření databázového souboru pro sinusové vibrace

Nejprve si vytvoříme databázový soubor pomocí funkce *VytvoreniSin(umisteni, zakladni_soubor)*, která má dva parametry. Oba dva jsou typu `char`. Jeden s cestou a názvem nového databázové souboru, například: "C:\\Slozka_1\\Podslozka_2\\navez_souboru_1". Cesta i s názvem by měla mít maximálně 256 znaků, jelikož 4 znaky jsou vyhrazeny na koncovku souboru. Zadávaný `char` by měl mít 252 znaků. A to z důvodu dlouhých cest na serveru. Operační systém Windows má problém pracovat s takovým souborem. A druhý `char` s cestou k souboru *SinZaklad.sin*.

4.1.2 Nahrání dat do databázového souboru pro sinusové vibrace

Pro nastavení měřícího zařízení slouží funkce *PrepisSin(umisteni, tridaSinus)*.

Parametr *umisteni* je typu `char`, který obsahuje cestu k souboru, například: "C:\\Slozka_1\\Podslozka_2\\navez_souboru_1".

Druhý parametr třída typu `Sinus`, která slouží jako pomocný nástroj pro kontrolu nastavovaných dat do databázového souboru pro zkušební laboratoř. Třída obsahuje metody, pomocí kterých data zapisujeme do třídy a tyto data prochází kontrolou jestli nepřekračují interval hodnot, se kterými je přístroj schopný pracovat. Poté co si pomocí metod nastavíme všechny hodnoty měření, které jsme potřebovali, zavoláme funkci *PrepisSin* s oběma parametry.

4.1.3 Čtení dat z databázového souboru pro sinusové vibrace

Pro čtení dat jsou určeny tři funkce. První funkce vytáhne z databázového souboru celé jedno měření, tedy `Data_*`, kde `*` představuje číslo tabulky. Tato funkce je `CteniTabulkySin(umisteni, tabulka, data)`.

Parametr `umisteni` je typu `char`, který obsahuje cestu k souboru, například: `"C:\\Slozka_1\\Podslozka_2\\nazev_souboru_1"`. Parametr `tabulka` je typu `int` a určuje nám kolikátou tabulku má vytáhnout z databázového souboru. Počet naměřených tabulek můžeme zjistit pomocí třetí funkce. Poslední parametr je struktura typu `MereniData`, kterou si nejprve vytvoříme a potom předáme do funkce jako parametr. Do této struktury se nám nahraje celé jedno měření. Tato struktura obsahuje 11 polí `floatů`, které představují vždy jeden sloupeček dat z jedné tabulky měření. Názvy polí ve struktuře odpovídají názvům sloupečků v databázovém souboru. Index (řádek) v tabulce vždy odpovídá indexu v poli, takže v polích na pozici 0 se nahraje hodnota „0“ pro lepší orientaci. Názvy sloupečků můžeme vidět v tabulce 4.1.

Druhá funkce načte vždy jeden řádek o určitému indexu ze všech naměřených tabulek. Funkce se nazývá `CteniRadkuSin(umisteni, index, pole)`. Parametr `umisteni` je opět typu `char`, který je ve stejném formátu jako u předchozí funkce. Parametr `index` je typu `int` a určuje z kolikátého řádku chceme z tabulek číst. Index může být v rozsahu 1 až 2004. To odpovídá počtu řádků v tabulce `Data_*`. Poslední parametr funkce je `floatové` pole polí, kde první index představuje z kolikáté tabulky data jsou a druhý index představuje název sloupečku. Názorná ukázka v tabulce 4.1. Pole, které budeme funkci předávat **musí** mít velikost **100*11**. Pro lepší orientaci v indexech, aby číslo měřené tabulky opravdu odpovídalo indexu prvního z polí, tak první pole má v sobě nahrané nuly. Tedy `Pole[0][x]`, kde $x \in \langle 0;10 \rangle$, se vyplní samé nuly.

Tabulka 4.1 Způsob ukládání řádků pro proměnnou z funkce na čtení řádku

Indexace - Data_1	Název sloupečku	Indexace - Data_2	Název sloupečku
Pole [1][0]	Drive	Pole [2][0]	Drive
Pole [1][1]	Channel_1	Pole [2][1]	Channel_1
Pole [1][2]	Channel_2	Pole [2][2]	Channel_2
Pole [1][3]	Channel_3	Pole [2][3]	Channel_3
Pole [1][4]	Channel_4	Pole [2][4]	Channel_4
Pole [1][5]	Channel_5	Pole [2][5]	Channel_5
Pole [1][6]	Channel_6	Pole [2][6]	Channel_6
Pole [1][7]	Channel_7	Pole [2][7]	Channel_7
Pole [1][8]	Channel_8	Pole [2][8]	Channel_8
Pole [1][9]	Control	Pole [2][9]	Control
Pole [1][10]	X_Data	Pole [2][10]	X_Data

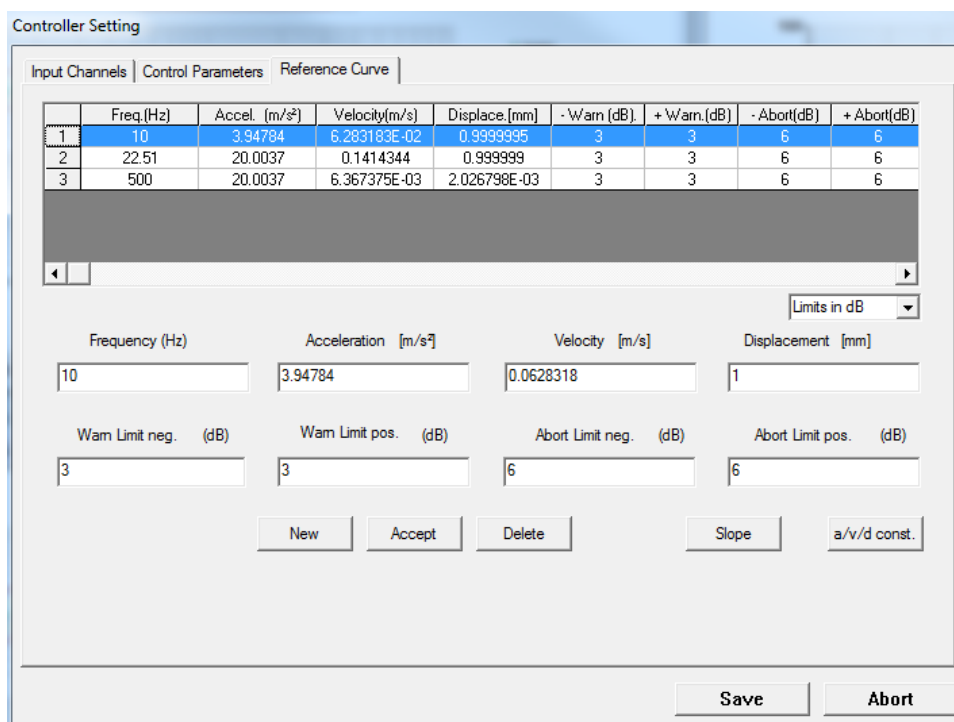
Třetí funkce načte z databázového souboru nastavení pro zkušební laboratoř, které je důležité pro sestavení protokolu. Tato funkce je `CteniNastaveniSin(umisteni, NastavenaData)`. Má dva parametry, `umisteni` typu `char`, který obsahuje cestu k souboru jako v předchozích funkčních a druhý parametr typu struktura `NastavenaData`, do kterých se uloží hodnoty z databázového souboru. Struktura `NastavenaData` obsahuje proměnné popsané v tabulce 4.2.

Tabulka 4.2 Seznam proměnných struktury NastavenaData pro sinusové vibrace

Názvy proměnných struktury NastavenaData	Název karty -> Název veličiny
float TypeOfInputCh8	Input Channels->Type of Input Ch 8.
float TypeOfInputCh7	Input Channels->Type of Input Ch 7.
float TypeOfInputCh6	Input Channels->Type of Input Ch 6.
float TypeOfInputCh5	Input Channels->Type of Input Ch 5.
float TypeOfInputCh4	Input Channels->Type of Input Ch 4.
float TypeOfInputCh3	Input Channels->Type of Input Ch 3.
float TypeOfInputCh2	Input Channels->Type of Input Ch 2.
float TypeOfInputCh1	Input Channels->Type of Input Ch 1.
char InfoTextCh8[256]	Input Channels->Info Text Ch 8.
char InfoTextCh7[256]	Input Channels->Info Text Ch 7.
char InfoTextCh6[256]	Input Channels->Info Text Ch 6.
char InfoTextCh5[256]	Input Channels->Info Text Ch 5.
char InfoTextCh4[256]	Input Channels->Info Text Ch 4.
char InfoTextCh3[256]	Input Channels->Info Text Ch 3.
char InfoTextCh2[256]	Input Channels->Info Text Ch 2.
char InfoTextCh1[256]	Input Channels->Info Text Ch 1.
float SweepType	Control Parameters -> Width
float SweepVelocity	Control Parameters -> Shock Inverted
float TestTime	Control Parameters -> Curve Type
float StartFrequency	Control Parameters -> Amplitude
float ConnctionWithSlipTable	Control Parameters -> Connection with Slip Table
float MassTestFixture	Control Parameters ->Mass, Test Fixture
float MassTestItem	Control Parameters ->Mass, Test Item
float MultiChannelControl	Control Parameters ->Shock-Distance
float Amplitude	Control Parameters ->Shock Number
float Frequency[101]	Reference Curve -> Frequency
float Acceleration[101]	Reference Curve -> Acceleration
float PocetTabulek	Počet naměřených tabulek Data *

4.2 Tabulka TargetCurve pro sinusové vibrace

V tabulce TargetCurve je 7 sloupců se 100 řádky. Názvy sloupců od levého okraje tabulky jsou **WarnLower_Y**, **WarnUpper_Y**, **AbortLower_Y**, **AbortUpper_Y**, **Y_Data**, **X_Data** a **Index**. Při tvorbě databázového souboru všechny hodnoty, kromě Indexu, nastavíme na 0. Každý řádek představuje nastavení jednoho měřicího cyklu. Hodnoty v původním softwaru se do této tabulky ukládaly z karty **Reference Curve** na obrázku 4.1.



Obrázek 4.1 Screen karty Reference Curve pro sinusové vibrace

Tabulka 4.3 Popis tabulky TargetCurve pro sinusové vibrace

Název kolonky v SW (Reference Curve)	Název v sloupci v DB (TargetCurve)	Interval hodnot / jednotky
Warm Limit neg.	WarnLower_Y	<-40 ; 40> [dB]
Warm Limit pos.	WarnUpper_Y	<-40 ; 40> [dB]
Abort Limit neg.	AbortLower_Y	<-40 ; 40> [dB]
Abort Limit pos.	AbortUpper_Y	<-40 ; 40> [dB]
Acceleration	Y_Data	<0.01 ; 10 000> [m/s ²]
Frequency	X_Data	<0.01 ; 10 000>[Hz]
Řádek	Index	<1 ; 100> - celá čísla

Do sloupečků **WarnLower_Y**, **WarnUpper_Y**, **AbortLower_Y**, **AbortUpper_Y** se neukládají přímo hodnoty, které zadáváme v SW, ale jejich přepočítání dle vzorců ukázaných v tabulce 4.4. Softwarový interface tento přepočítání provádí, takže hodnoty zadáváme v intervalech, které jsou uvedeny v tabulce 4.3.

Metoda pro zapisování je `SetTargetCurve(int Index, float Acceleration, float Frequency, float WarnLimitneg, float WarnLimitpos, float AboutLimitneg, float AboutLimitpos)`. Název parametrů vychází z názvu proměnných, které zadáváme v originálním SW. Přehled můžeme vidět v tabulce 4.3. Metoda kontroluje, zda byl dobře zadán index, aby se nepřeskakovaly řádky, jestli nebyla zadána dvakrát stejná hodnota frekvence a u všech parametrů kontroluje interval hodnot. Když nalezne nesrovnalost, tak generuje výjimku s hlášením, co bylo zadáno špatně.

Tabulka 4.4 Vzorce pro přepočítání v tabulce TargetCurve pro sinusové vibrace

Název v sloupci v DB (TargetCurve)	Vzorec přepočtu
WarnLower_Y	$= \text{Acceleration} * 10^{\frac{-\text{Warm Limit neg.}}{20}}$
WarnUpper_Y	$= \text{Acceleration} * 10^{\frac{\text{Warm Limit pos.}}{20}}$
AbortLower_Y	$= \text{Acceleration} * 10^{\frac{-\text{Abort Limit neg.}}{20}}$
AbortUpper_Y	$= \text{Acceleration} * 10^{\frac{\text{Abort Limit pos.}}{20}}$

4.3 Tabulka SystemDataText pro sinusové vibrace

V tabulce jsou dva sloupce **Index** a **SystemDataText**. Index má číslování od 1 do 22, takže v tabulce je 22 řádků.

V první řádce je datum a čas založení souboru, čas je ve formátu 22.02.18 14:40:36. Druhý řádek obsahuje text zadaný v okně **EnterText**. Po té následují názvy pohledů grafů. Každý ze 4 grafů má 3 pohledy. Od indexu 15 jsou ukládány názvy měřících channelů. Seznam těchto channelů i s metodami, které zapíše názvy do třídy **Sinus**, můžeme vidět v tabulce 4.5. Maximální délka názvu channelů musí být 255 znaků.

Tabulka 4.5 Popis dat z karty Input Channels -> Info Text pro sinusové vibrace

Index	Název kolonky v SW	Metoda
15	Info Text Ch.8	SetInfoTextCh8(char InfoTextCh8)
16	Info Text Ch.7	SetInfoTextCh7(char InfoTextCh7)
17	Info Text Ch.6	SetInfoTextCh6(char InfoTextCh6)
18	Info Text Ch.5	SetInfoTextCh5(char InfoTextCh5)
19	Info Text Ch.4	SetInfoTextCh4(char InfoTextCh4)
20	Info Text Ch.3	SetInfoTextCh3(char InfoTextCh3)
21	Info Text Ch.2	SetInfoTextCh2(char InfoTextCh2)
22	Info Text Ch.1	SetInfoTextCh1(char InfoTextCh1)

Název měřícího Channelu 1-8 se zadává v kartě Input Channels, kterou můžeme vidět na obrázku 4.3. Názvy channelů jsou jediná data z této tabulky, se kterými daný softwarový interface pracuje.

4.4 Tabulka SystemDataFloat pro sinusové vibrace

Tabulka SystemDataFloat obsahuje většinu dat sloužících k nastavení měřící vibrační stanice CVVOZE. Obsahuje 2 sloupce **SystemDataFloat** a **Index**. Tato tabulka má 253 řádků.

4.4.1 Karta Windows Settings

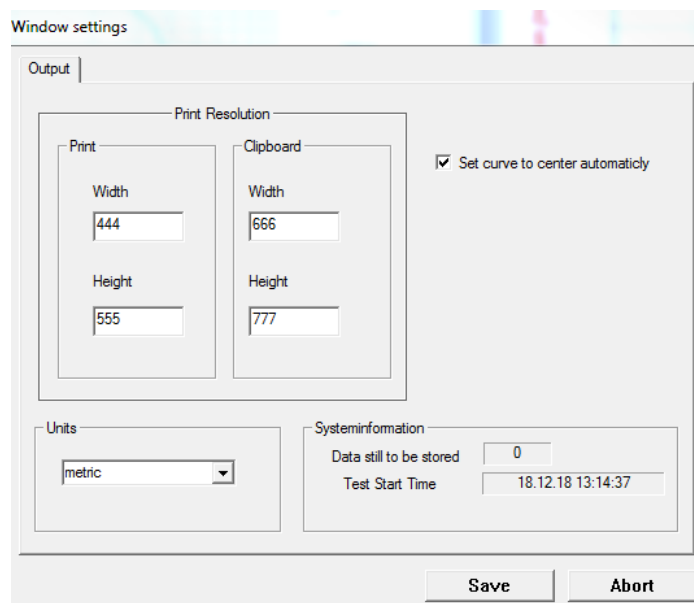
První část tabulky SystemDataFloat je znázorněna v tabulce 4.6, která vychází z karty Window setting, kterou můžeme vidět na obrázku 4.2.

Tabulka 4.6 Popis dat z karty Window Settings

Index	Popis / Název kolonky v SW	Interval hodnot
2	Počet naměřených tabulek	
4	Počet zadaných měření v kartě Reference Curve	
5	Window Settings-> Units	1 or 0
6	Window Settings-> Print-> Width	< 1 ; 4000 >
7	Window Settings-> Print->Height	< 1 ; 4000 >
8	Window Settings->Clipboard->Width	< 1 ; 4000 >
9	Window Settings->Clipboard->Height	< 1 ; 4000 >
16	Window Settings-> Set Curve to center automaticly	1 or 0

Na indexu 2 je uložen počet tabulek s naměřenými daty, která se doplňují automaticky. Na indexu 4 je počet zadaných měření v kartě Reference Curve, která se opět přepisují automaticky. Jednotky **Units** se vybírají mezi metrickými a americkými, kde **metrické = 0** a **americké = 1**, jsou na 5. řádku. **Set curve to center automaticly** je na řádku 16. Když je políčko **zaškrtnuté** ukládá = **1** a pro **nezaškrtnuté** = **0**.

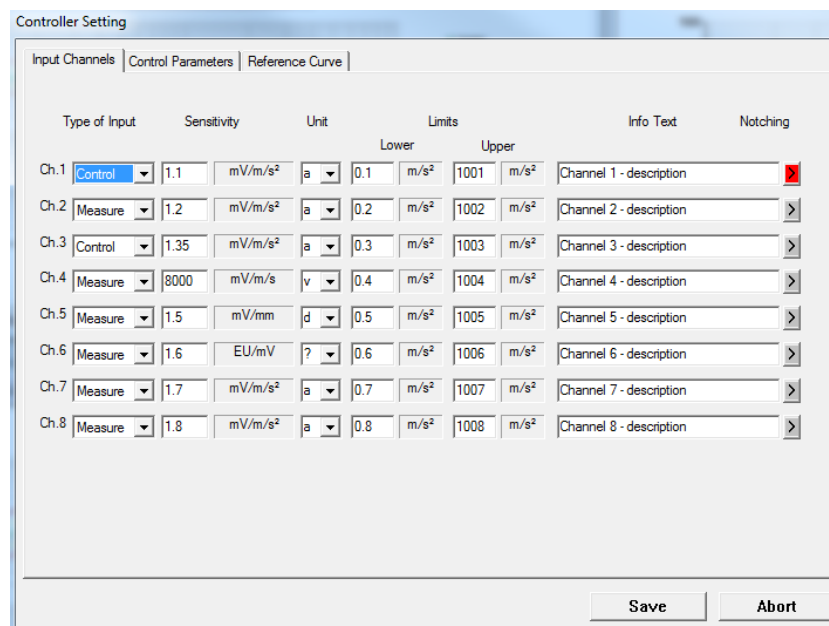
Metoda pro zapsání těchto hodnot do třídy **Sinus** je **SetWindowSettings(float aPrintWidth, float aPrintHeight, float ClipboardWidth, float aClipboardHeight, float aSetCurveToCenterAutomaticly, float aWSUnits)**. Názvy parametrů odpovídají kolonkám v kartě. Kontroluje se interval zadávaných hodnot, pokud dojde k překročení, tak generuje výjimku s chybovým hlášením.



Obrázek 4.2 Screen karty Window Settings -> Output

4.4.2 Karta Input Channels pro sinusové vibrace

Dále popíši, kam se ukládají data z karty Input Channel, která je na obrázku 4.3.



Obrázek 4.3 Screen karty Input Channels pro sinusové vibrace

Veličina Type of Input je **výběrová veličina**, která může nabývat 4 hodnoty. Pro možnost **off** = 0, **abrot** = 1, **control** = 2 a **measure** = 3. V tabulce 4.7 je znázorněno do jakých řádků se Type of Input ukládá.

Tabulka 4.7 Přehled veličiny Type of Input pro sinusové vibrace

Index	Název kolonky v SW	Interval hodnot
21	Ch.8->Type of Input	< 0 or 1 or 2 or 3 >
22	Ch.7->Type of Input	< 0 or 1 or 2 or 3 >
23	Ch.6->Type of Input	< 0 or 1 or 2 or 3 >
24	Ch.5->Type of Input	< 0 or 1 or 2 or 3 >
25	Ch.4->Type of Input	< 0 or 1 or 2 or 3 >
26	Ch.3->Type of Input	< 0 or 1 or 2 or 3 >
27	Ch.2->Type of Input	< 0 or 1 or 2 or 3 >
28	Ch.1->Type of Input	< 0 or 1 or 2 or 3 >

Kolonka **Sensitivity** (citlivost) je popsána v tabulce 4.8.

Unit (jednotky) pro citlivost. Je to výběrová položka, kde **a**(zrychlení [mV/m/s²]) = 0, **v**(rychlost [mV/m/s]) = 1, **d**(výchylka [mV/mm]) = 2, **?**[EU/mV] = 3.

Dalším parametrem, který se zadává je **Limits**, kde zadáváme **Lower** (dolní) a **Upper** (horní) meze dané limity. Pro hodnoty **Type of Input** = **Measure** nebo **Control** platí tabulka 4.9. Když hodnota **Type of Input** = **Abort**, platí tabulka 4.11. Do databázové souboru u **Type of Input** = **Abort** se ukládají hodnoty pro **Limits** vynásobené 1000. Tento přepočít dělá SW interface sám.

Tabulka 4.8 Přehled veličiny Sensitivity pro sinusové vibrace

Index	Název kolonky v SW	Interval hodnot
29	Ch.8->Sensitivity	<0.001 ; 10000>
30	Ch.7->Sensitivity	<0.001 ; 10000>
31	Ch.6->Sensitivity	<0.001 ; 10000>
32	Ch.5->Sensitivity	<0.001 ; 10000>
33	Ch.4->Sensitivity	<0.001 ; 10000>
34	Ch.3->Sensitivity	<0.001 ; 10000>
35	Ch.2->Sensitivity	<0.001 ; 10000>
36	Ch.1->Sensitivity	<0.001 ; 10000>

Tabulka 4.9 Přehled veličiny Units pro sinusové vibrace

Index	Název kolonky v SW	Interval hodnot
37	Ch.8->Unit	< 0 or 1 or 2 or 3 >
38	Ch.7->Unit	< 0 or 1 or 2 or 3 >
39	Ch.6->Unit	< 0 or 1 or 2 or 3 >
40	Ch.5->Unit	< 0 or 1 or 2 or 3 >
41	Ch.4->Unit	< 0 or 1 or 2 or 3 >
42	Ch.3->Unit	< 0 or 1 or 2 or 3 >
43	Ch.2->Unit	< 0 or 1 or 2 or 3 >
44	Ch.1->Unit	< 0 or 1 or 2 or 3 >

Tabulka 4.10 Přehled veličiny Limits pro Type of Input = Measure a Control pro sinusové vibrace

Index	Název kolonky v SW	Interval hodnot
45	Ch.8->Limits->Upper	<0.01 ; 10000>
46	Ch.7->Limits->Upper	<0.01 ; 10000>
47	Ch.6->Limits->Upper	<0.01 ; 10000>
48	Ch.5->Limits->Upper	<0.01 ; 10000>
49	Ch.4->Limits->Upper	<0.01 ; 10000>
50	Ch.3->Limits->Upper	<0.01 ; 10000>
51	Ch.2->Limits->Upper	<0.01 ; 10000>
52	Ch.1->Limits->Upper	<0.01 ; 10000>
53	Ch.8->Limits->Lower	<0.01 ; 10000>
54	Ch.7->Limits->Lower	<0.01 ; 10000>
55	Ch.6->Limits->Lower	<0.01 ; 10000>
56	Ch.5->Limits->Lower	<0.01 ; 10000>
57	Ch.4->Limits->Lower	<0.01 ; 10000>
58	Ch.3->Limits->Lower	<0.01 ; 10000>
59	Ch.2->Limits->Lower	<0.01 ; 10000>
60	Ch.1->Limits->Lower	<0.01 ; 10000>

Tabulka 4.11 Přehled veličiny Limits pro Type of Input = Abort pro sinusové vibrace

Index	Název kolonky v SW	Interval hodnot
61	Ch.8->Limits->Upper	<0.001 ; 10>
62	Ch.7->Limits->Upper	<0.001 ; 10>
63	Ch.6->Limits->Upper	<0.001 ; 10>
64	Ch.5->Limits->Upper	<0.001 ; 10>
65	Ch.4->Limits->Upper	<0.001 ; 10>
66	Ch.3->Limits->Upper	<0.001 ; 10>
67	Ch.2->Limits->Upper	<0.001 ; 10>
68	Ch.1->Limits->Upper	<0.001 ; 10>
69	Ch.8->Limits->Lower	<0.001 ; 10>
70	Ch.7->Limits->Lower	<0.001 ; 10>
71	Ch.6->Limits->Lower	<0.001 ; 10>
72	Ch.5->Limits->Lower	<0.001 ; 10>
73	Ch.4->Limits->Lower	<0.001 ; 10>
74	Ch.3->Limits->Lower	<0.001 ; 10>
75	Ch.2->Limits->Lower	<0.001 ; 10>
76	Ch.1->Limits->Lower	<0.001 ; 10>

Tabulka 4.12 Přehled metod pro kartu Input Channels pro sinusové vibrace

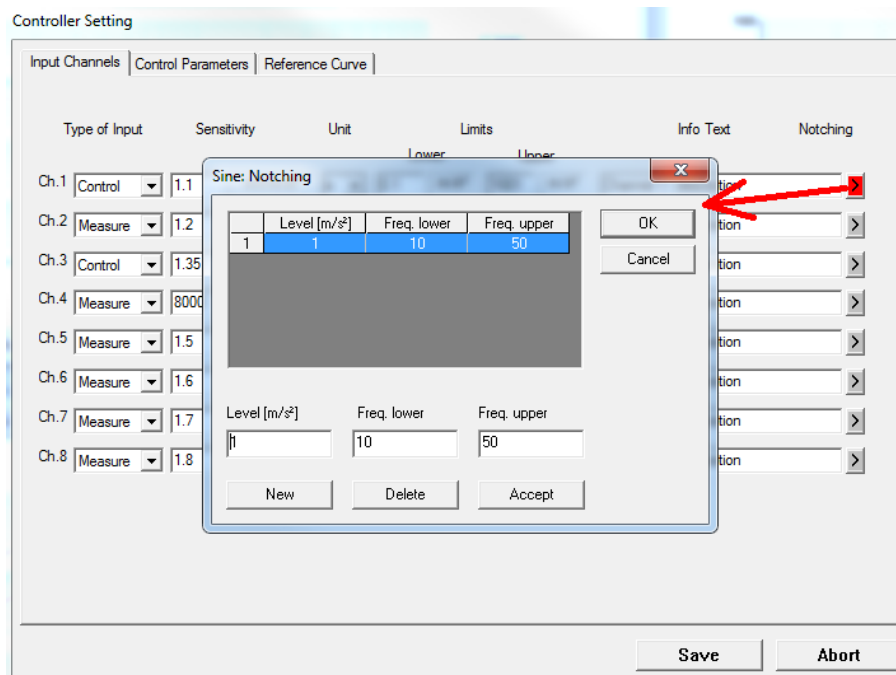
Channel	Název metody
Channel 1	SetInputChannel18(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 2	SetInputChannel17(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 3	SetInputChannel16(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 4	SetInputChannel15(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 5	SetInputChannel14(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 6	SetInputChannel13(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 7	SetInputChannel12(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)
Channel 8	SetInputChannel11(float TypeOfInput, float Sensitivity, float Unit, float LimitsUpper, float LimitLower)

V tabulce 4.12 je přehled metod, které se používají pro zapsání dat z karty Input Channels do třídy **Sinus**, která se pak může pomocí funkce uvedené v kapitole 4.1.2 přepsat v databázovém souboru.

Kolonka **Info Text**, kterou vidíme na obrázku 4.3 je popsána v kapitole 4.3.

4.4.3 Karta Notching pro sinusové vibrace

Od řádku 77 do řádku 196 jsou hodnoty zadávané v okně Notching, které můžeme vidět na obrázku 4.4. Zadávají se 3 hodnoty pro jeden řádek a maximálně 5 řádků pro každý Channel. Hodnoty, které se zadávají jsou **Level** [m/s^2], dolní a horní mez frekvence **Freq.lower** a **Freq.upper** [Hz].



Obrázek 4.4 Screen podkarty Notching pro sinusové vibrace

Tabulka 4.13 Seznam veličin pro Notching pouze Channel 8

Index	Název veličiny v SW	Interval hodnot
77	Ch.8->Notching->Level (1. řádek)	<0.001 ; 9999;9 >
78	Ch.8->Notching->Level (2. řádek)	<0.001 ; 9999;9 >
79	Ch.8->Notching->Level (3. řádek)	<0.001 ; 9999;9 >
80	Ch.8->Notching->Level (4. řádek)	<0.001 ; 9999;9 >
81	Ch.8->Notching->Level (5. řádek)	<0.001 ; 9999;9 >
82	Ch.8->Notching->Freq.Lower (1. řádek)	<50 ; 800 >
83	Ch.8->Notching->Freq.Lower (2. řádek)	<50 ; 800 >
84	Ch.8->Notching->Freq.Lower (3. řádek)	<50 ; 800 >
85	Ch.8->Notching->Freq.Lower (4. řádek)	<50 ; 800 >
86	Ch.8->Notching->Freq.Lower (5. řádek)	<50 ; 800 >
87	Ch.8->Notching->Freq.Upper (1. řádek)	<50 ; 800 >
88	Ch.8->Notching->Freq.Upper (2. řádek)	<50 ; 800 >
89	Ch.8->Notching->Freq.Upper (3. řádek)	<50 ; 800 >
90	Ch.8->Notching->Freq.Upper (4. řádek)	<50 ; 800 >
91	Ch.8->Notching->Freq.Upper (5. řádek)	<50 ; 800 >

Tabulka 4.14 Metody pro Notching pouze pro Channel 8

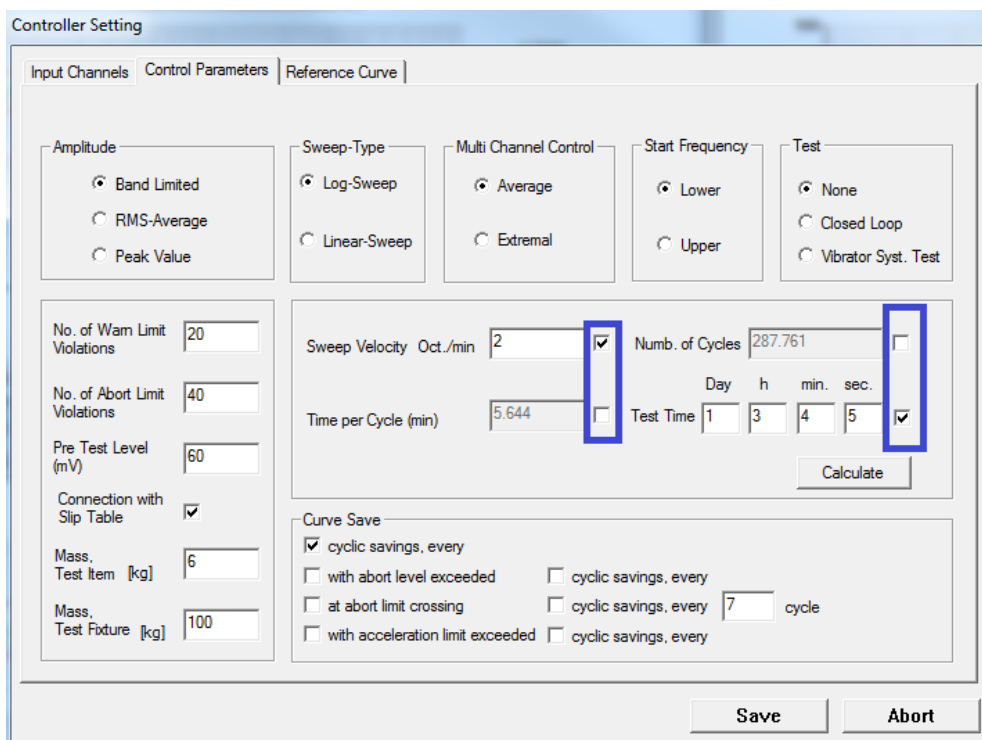
Channel	Metoda
Channel 8 - Řádek 1	SetInputChannel18Notching1(float Level, float FreqLower, float FreqUpper)
Channel 8 - Řádek 2	SetInputChannel18Notching2(float Level, float FreqLower, float FreqUpper)
Channel 8 - Řádek 3	SetInputChannel18Notching3(float Level, float FreqLower, float FreqUpper)
Channel 8 - Řádek 4	SetInputChannel18Notching4(float Level, float FreqLower, float FreqUpper)
Channel 8 - Řádek 5	SetInputChannel18Notching5(float Level, float FreqLower, float FreqUpper)

V tabulce 4.13 je popsáno na jaký řádek se ukládají hodnoty z okna Notching pro Channel 8. Hodnoty pro ostatní Channels se postupně ukládají pod tyto hodnoty. Následně se ukládá 15 hodnot pro Channel 7 a potom pro Channel 6 atd.

V tabulce 4.14 jsou napsány metody pro zápis hodnoty z karty Notching pro Channel 8. Každý channel má opět 5 metod pro zápis, které se jmenují stejně, pouze se mění v názvu metody číslo channelu.

4.4.4 Karta Control Parameters pro sinusové vibrace

V této kapitole se budu věnovat tomu, kam se ukládají hodnoty z karty Control Parameters do databázového souboru a jaké metody jsou k tomu určené ve vytvořeném interfacu. Karta Control Parameters je na obrázku 4.5.



Obrázek 4.5 Screen karty Control Parameters pro sinusové vibrace

Tabulka 4.15 Přehled veličin z karty Control Parameters a seznam metod pro sinusové vibrace

Index	Název veličiny v SW	Interval hodnot	Metoda zápisu
200	Počet zadaných měření v kartě Reference Curve		Automaticky vyplněné
201	Sweep Velocity [Oct./min]	< 0.01 ; 50.000 >	SetSweepVelocity()
202	Sweep - type	< 0 or 1 >	SetSweepType()
203	Test Time [s]	< 1 ; 4 147 200 >	SetTestTime()
204	Multi Channel Control	< 0 or 1 >	SetMultiChannelControl()
205	Amplitude	< 0 or 1 or 2 >	SetAmplitude()
206	Start Frequency	< 0 or 1 >	SetStartFrequency()
207	Pre Test Level [mV]	< 1 ; 20000 >	SetPreTestLevel()
208	Test	< 0 or 1 or 2 >	
209	Mass, Test Item [kg]	< 0 ; 300 >	SetMassTestItem()
210	Cycle	< 1 ; 1000000 >	SetCycle()
211	Kombinace zatržitek Curve Save	Viz tabulka 4.17	SetCurveSave()
212	Kombinace zatržitek v modrých obdelnících na obrázku 4.5	Viz tabulka 4.16	SetSTNT()
215	No. of Warm Limit Violations	< 1 ; 100 >	SetNoOfWarmLimitViolations()
216	No. of Abort Limit Violations	< 1 ; 100 >	SetNoOfAboarLimitViolations()
217	Mass, Test Fixture [kg]	< 0 ; 300 >	SetMassTestFixture()
218	Connection with Slip Table	< 0 or 1 >	SetConnectionWithSlipTable()

Tabulka 4.16 Možných kombinací a hodnoty pro řádek 212

Hodnota	Název zatržítka
0	žádné zatržítko není zatržené
101	Time Per Cycle + Test Time
110	Number per Cycle + Time per Cycle
1010	Sweep Velocity + Number of Cycle
1001	Sweep Velocity + Test Time

Na řádku 200 se ukládá stejně jako na řádek 4 počet zadaných řádků měření v kartě Reference Curve.

- Sweep Type 0 = Log-Sweep 1 = Lineat-Sweep.
- Multi Channel Control 0 = Avarage 1 = External
- Amplitude 0 = Band Limited 1 = RMS-Avarage 2 = Peak Value
- Start Frequency 0 = Lower 1 = Upper
- Connection with Slip Table 0 = false 1 = true

Na řádku 212 se ukládá hodnota, která představuje kombinaci zatržitek, která jsou na obrázku 4.5 v modrých rámečcích. Hodnoty, které se do dané buňky ukládají jsou v tabulce 4.16.

Řádek 211 nese hodnotu, která představuje kombinaci zatržitek **Curve Save** v dolní části obrázku 4.5. Možné kombinace a její hodnoty vidíme v tabulce 4.17.

Veličina **Cycle** zapisuje do databázového souboru dvojnásobnou hodnotu než byla zadaná do kolonky. Tento přepočít provádí SW interface.

Každá metoda, uvedená v tabulce 4.15, má jeden parametr a ten je typu **float**. Tuto hodnotu ukládá do třídy, která se pak může přepsat funkcí do databázového souboru.

Tabulka 4.17 Možných kombinací a hodnoty pro řádek 211

Hodnota	Název zatržítka
1010	Cycling saving
1010.001	With abort level exceeded + Cycling saving
1010.002	With acceleration limit exceeded + Cycling saving
1010.003	With abort level exceeded + With acceleration limit exceeded + Cycling saving
1010.004	At about limit crossing + Cycling saving
1010.006	With abort level exceeded + At about limit crossing + Cycling saving
1010.007	With acceleration limit exceeded + At about limit crossing + Cycling saving
1010.008	With abort level exceeded + With acceleration limit exceeded + At about limit crossing + Cycling saving

4.5 Tabulka Logbook pro sinusové vibrace

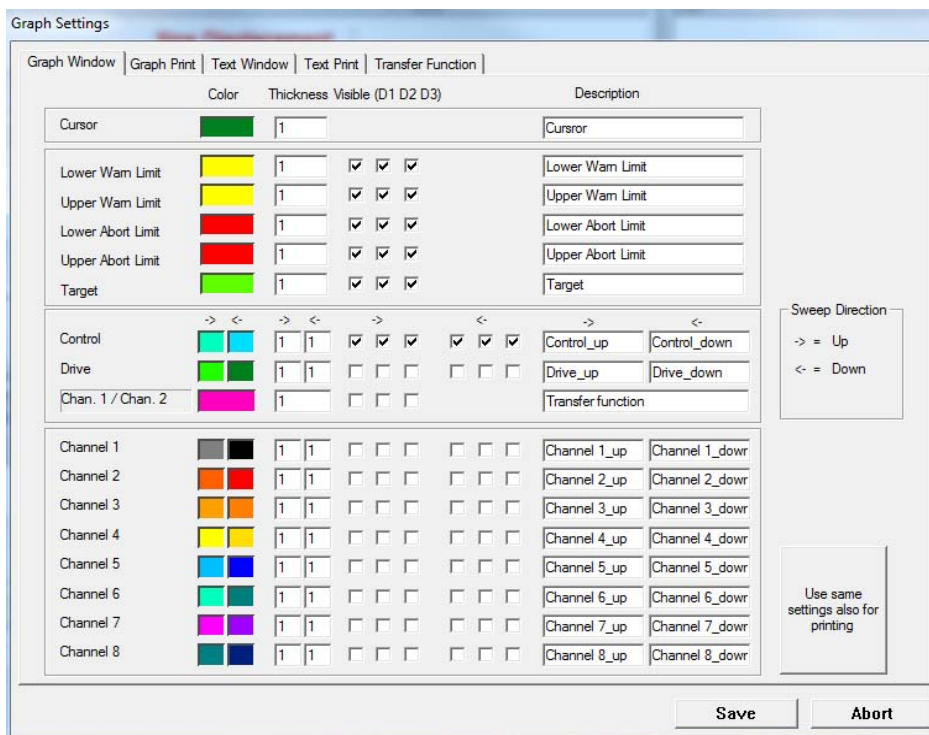
V tabulce Logbook jsou zaznamenány data a časy vzniku souboru s nastavením, spuštěním testování, ukončením testování a další důležité časové údaje. Obsahuje 4 sloupce s názvy **Data**, **Info**, **Level** a **Index**. Má 51 prázdných řádků, vyjma prvního, kde je zaznamenám ve sloupci **Info** vznik souboru datového souboru ve formátu: „Start DD.MM.RR HH:MM:SS“, kde DD znamená kolikátého dne, MM je měsíc a RR jsou poslední dvě číslice z roku. Následuje mezera s časem ve formátu hodiny:minuty:sekundy. Ve sloupci **Data** jsou na prvním řádku „--“ a do sloupce **Level** je vepsaná 0.

4.6 Tabulka GraphStyle pro sinusové vibrace

Tabulka GraphStyle nese informaci o barvách křivek a os zobrazovaných grafů v originálním softwaru pro ovládání zkušební laboratoře CVVOZE. Obsahuje 9 sloupečků a to **Farbe**, **Dicke**, **D1**, **D2**, **D3**, **LegendeText**, **LegendeFarbe**, **LegendeDicke** a **Index**. Má **216 řádků**, ale data uložená v této tabulce pro tvorbu softwarového interfacu nejsou důležitá. Na obrázku Graph Settings je vidět okno z originálního softwaru ze záložky Graph Window. Záložka Graph Print má úplně stejné rozložení.

V tabulce 4.18 vidíme ukázkou hodnot zapsaných v databázovém souboru v tabulce GraphStyle.

Na obrázku 4.6 vidíme kolonku **Color**, kde si zaškrtneme barvu. Hodnota této barvy se ukládá do sloupečku **Farbe** v databázovém souboru. Šířku čáry **Thickness** zapisujeme do sloupečku **Dicke**.



Obrázek 4.6 Screen karty Graph Window pro sinusové vibrace

Tabulka 4.18 Ukázka ukládání dat Graph Window a Graph Print

SW	Color	Thickness	Visible			Description			
			D1	D2	D3				
DB	Farbe	Dicke	D1	D2	D3	LegendeText	LegendeFarbe	LegendeDicke	Index
	98	2	-1	-1	-1	Cursor	0	0	1
	40	1	-1	-1	-1	Lower Warn Limit	0	0	2

V jakém pohledu (D1, D2 a D3) bude křivka viditelná, vybíráme v kolonce **Visible** a ukládáme do sloupečku **D1**, **D2** a **D3**. Pro viditelnou zapisujeme -1 a pro neviditelnou 0.

Kolonku **Description** ukládáme do sloupečku **LegendeText**, kde se udává název dané křivky. Maximální délka názvu může být 255 znaků.

Tabulka GraphStyle má 216 řádků, ty jsou rozvrženy do 8 stejných bloků. Tyto bloky znázorňují grafická nastavení pro 4 grafy v původním softwaru a 4 nastavení pro tisk grafů. V tabulce je prvně uloženo nastavení pro graf č.1 softwarové zobrazení a pak následuje nastavení tisku pro graf č.1. Dále se ukládá graf č.2, č.3 a č.4. Pro každý blok má vyhrazeno 27 řádků. Jak jsou za sebou můžeme vidět v tabulce 4. Na obrázku 4.6 je vidět že, položky Control, Drive a Channel 1-8 mají možnost zapisovat hodnoty pro Up (->) a Down (<-).

Tabulka 4.19 Pořadí v jakém se ukládají hodnoty do databázového souboru

Index	Název kolonky v SW	Index	Název kolonky v SW
1	Cursor	15	Drive Up
2	Lower Warn Limit	16	Control Up
3	Upper Warn Limit	17	Channel 8 Down
4	Lower Abort Limit	18	Channel 7 Down
5	Upper Abort Limit	19	Channel 6 Down
6	Target	20	Channel 5 Down
7	Channel 8 Up	21	Channel 4 Down
8	Channel 7 Up	22	Channel 3 Down
9	Channel 6 Up	23	Channel 2 Down
10	Channel 5 Up	24	Channel 1 Down
11	Channel 4 Up	25	Drive Down
12	Channel 3 Up	26	Control Down
13	Channel 2 Up	27	Transfer function
14	Channel 1 Up		

4.7 Tabulka Drive pro sinusové vibrace

Do tabulky Drive se neukládá žádné nastavení ani naměřená data. Jediné co obsahuje jsou dva prázdné sloupce s názvy **Drive** a **Index**.

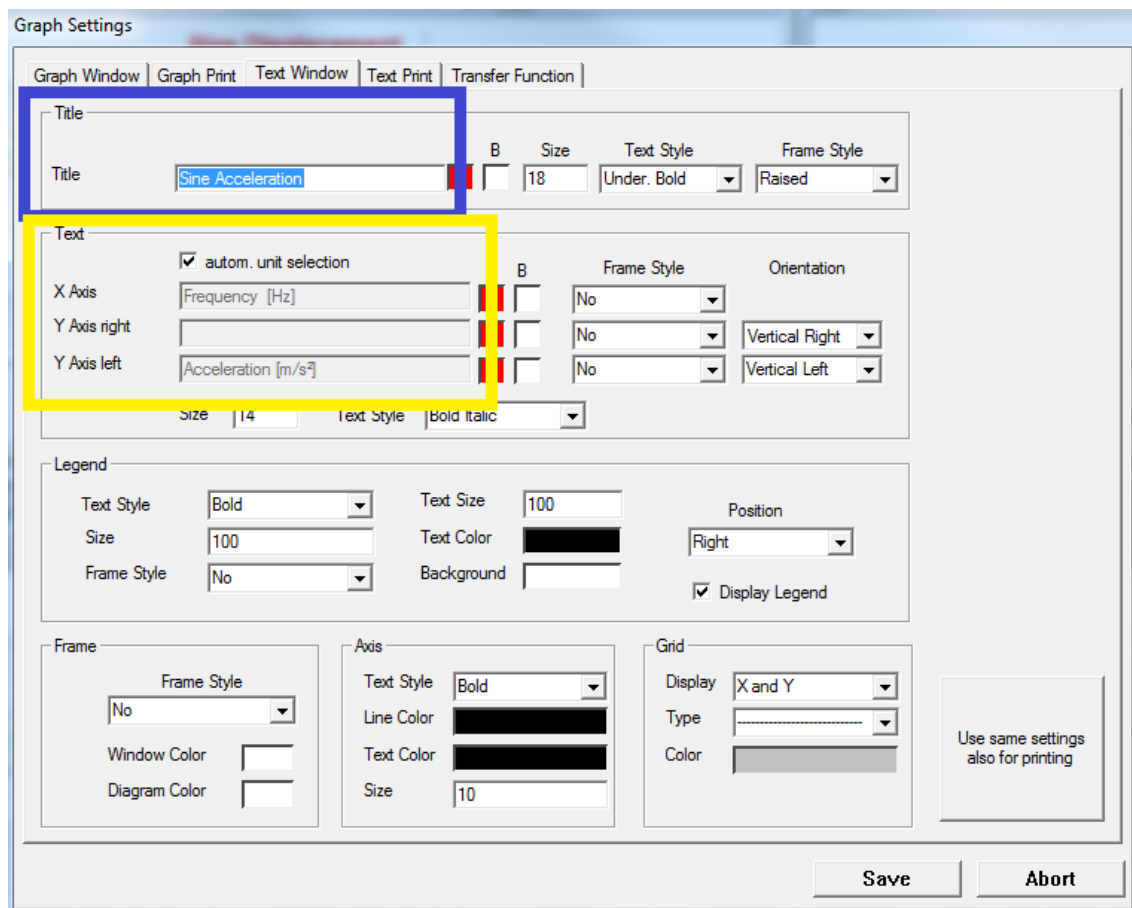
4.8 Tabulka DiagrammStyleText pro sinusové vibrace

Tato tabulka není pro tvorbu interface až tak důležitá. Data z ní nevyužijeme a slouží jen pro oficiální softwaru. Jsou v ní uvedené názvy a osy 4 grafů, které se zobrazují ve zmíněném softwaru. Má 5 sloupců v pořadí DiagrammStyleText4, DiagrammStyleText3, DiagrammStyleText1, DiagrammStyleText2 a Index. Po 8 řádcích, z toho 3. a 7. řádek je vyplněný jen ve sloupci Index, jinak jsou prázdné. V každém sloupečku jsou názvy pro jeden graf.

Tabulka 4.20 Kompletní ukázka tabulky DiagrammStyleText z DB

DiagrammStyleText4	DiagrammStyleText3	DiagrammStyleText1	DiagrammStyleText2	Index
Sine User	Sine Velocity	Sine Displacement	Sine Acceleration	1
Frequency [Hz]	Frequency [Hz]	Frequency [Hz]	Frequency [Hz]	2
				3
Control Output [mV]	Velocity [m/s]	Displacement [mm]	Acceleration [m/s,]	4
Sine User	Sine Velocity	Sine Displacement	Sine Acceleration	5
Frequency [Hz]	Frequency [Hz]	Frequency [Hz]	Frequency [Hz]	6
				7
Control Output [mV]	Velocity [m/s]	Displacement [mm]	Acceleration [m/s,]	8

Na obrázku 4.7 je otevřena karta **Graph Window** v modrém obdélníku v kolonce Title je název grafu, který je uložený ve sloupečku DiagrammStyleText2 v prvním řádku. Druhý, třetí a čtvrtý řádek obsahuje informaci o názvech os, které můžeme vidět ve žlutém obdélníku. Na zbylých čtyřech řádcích jsou zapsány stejné informace, ale z karty Text Print, určené pro tisk grafů.



Obrázek 4.7 Screen karty Graph Window pro sinusové vibrace

4.9 Tabulka DiagrammStyleFloat pro sinusové vibrace

Ani tuto tabulku pro tvorbu zadané interface nepotřebujeme. Obsahuje nastavení pro grafy v oficiálním softwaru. Opět obsahuje 5 sloupců: **DiagrammStyleFloat4**, **DiagrammStyleFloat3**, **DiagrammStyleFloat2**, **DiagrammStyleFloat1** a **Index**. Každý sloupeček je zase určen pro jeden graf. Data jsou uložena pomocí čísel typu float. Sloupečky opět obsahují data z karet Text Print a Text Window.

5. NÁHODNÉ VIBRACE

Zde popíšeme nastavení a data v databázovém souboru pro náhodné vibrace. Popis náhodných vibrací se v jistých částech bude shodovat se sinusovými vibracemi, takže se na ně budu odkazovat. Dále vysvětlím funkce a metody, které slouží pro vytvoření, přepis a čtení data v databázovém souboru s koncovkou (*.rau).

Databázový soubor pro náhodné vibrace s příponou (*.rau) se skládá z 8 tabulek pro nastavení vibračního zařízení a 1 až 99 tabulek s naměřenými daty. Tyto tabulky se značí Data_1 až Data_99. Tabulky nastavení vibračního zařízení jsou TargetCurve, SystemDataText, SystemDataFloat, Logbook, GraphStyle, Drive, DiagrammStyleText a DiagrammStyleFloat.

5.1 Základní funkce pro náhodné vibrace

Pro práci s databázovým souborem (*.rau) potřebujeme hlavičkové soubory *RauHeader.h*, *RauFunkce.cpp* a *RauZaklad.rau*, které nalezneme v přílohách. Soubor *RauZaklad.rau* stejně jako *RauZaklad.ran* je přednastavený databázový soubor. Důvod je vysvětlen v kapitole 4.1.

5.1.1 Vytvoření databázové souboru pro náhodné vibrace

Nejprve si vytvoříme kopii databázového souboru pomocí funkce *VytvoreniRau* (*umistení*, *zakladni_soubor*), která má dva parametry. Oba dva jsou typu `char`. Jeden s cestou a názvem nového databázového souboru, například: „C:\\Slozka_1\\Podszlozka_2\\nazev_souboru_1“. Cesta i s názvem by měla mít maximálně 256 znaků, jelikož 4 znaky jsou vyhrazeny na koncovku souboru. Zadávaný `char` by měl mít 252 znaků, a to z důvodu dlouhých cest na serveru. Operační systém Windows má problém pracovat s takovým souborem. A druhý `char` s cestou k souboru *RauZaklad.rau*, například: C:\\Slozka_1\\Podszlozka_2\\.

5.1.2 Přepis dat v databázovém souboru pro náhodné vibrace

Pro nastavení měřicího zařízení slouží funkce *PrepisRau*(*umistení*, *tridaRandom*).

Parametr *umistení* je typu `char`, který obsahuje cestu k souboru, například: „C:\\Slozka_1\\Podszlozka_2\\nazev_souboru_1“.

Druhý parametr *tridaRandom* typu `Random`, která slouží jako pomocný nástroj pro kontrolu nastavovaných dat do databázového souboru pro zkušební laboratoř. Třída obsahuje metody, pomocí kterých data zapisujeme do třídy a tyto data prochází kontrolou, jestli nepřekračují interval hodnot, se kterými je přístroj schopný pracovat. Poté co si pomocí metod nastavíme všechny hodnoty měření, které jsme potřebovali, zavoláme funkci *PrepisRau* s oběma parametry.

5.1.3 Čtení dat z databázového souboru pro náhodné vibrace

Pro čtení dat jsou určeny tři funkce. První funkce vytáhne z databázového souboru celé jedno měření, tedy *Data_**, kde ***, představuje číslo tabulky. Tato funkce je *CteniTabulkyRau*(*umistení*, *tabulka*, *data*).

Parametr *umisteni* je typu **char**, který obsahuje cestu k souboru, například: „C:\\Složka_1\\Podložka_2\\nazev_souboru_1“. Parametr *tabulka* je typu **int** a určuje nám kolikátou tabulku má vytáhnout z databázového souboru. Počet naměřených tabulek můžeme zjistit pomocí třetí funkce, která zde bude popsána. Poslední parametr je struktura typu **MereniData**, kterou si nejprve vytvoříme a potom předáme do funkce jako parametr. Bližší popis opět najdeme v podkapitole 4.1.3.

Druhá funkce načte vždy jeden řádek o určitému indexu ze všech naměřených tabulek. Funkce se nazývá *CteniRadkuRau(umisteni, index, pole)*. Parametr *umisteni* je opět typu **char**, který je ve stejném formátu jako u předchozí funkce. Parametr *index* je typu **int** a určuje z kolikátého řádku chceme z tabulek číst. Index může být v rozsahu 1 až 2004. Poslední parametr funkce je **floatové** pole polí. Více informací v podkapitole 4.1.3.

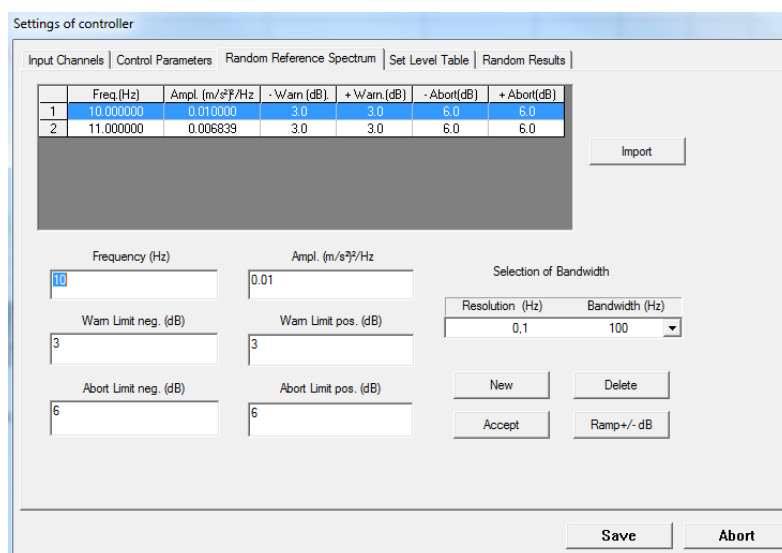
Třetí funkce načte z databázové souboru nastavení pro zkušební laboratoř, které je důležité pro sestavení protokolu. Tato funkce je *CteniNastaveniRau(umisteni, NastavenaData)*. Má dva parametry, *umisteni* typu **char**, který obsahuje cestu k souboru jako v předchozích funkcích a druhý parametr typu **NastavenaData**, do kterých se uloží hodnoty z databázového souboru. Struktura **NastavenaData** obsahuje proměnné popsané v tabulce 5.1.

Tabulka 5.1 Seznam proměnných struktury *NastavenaData* pro náhodné vibrace

Názvy proměnných struktury <i>NastavenaData</i>	Název karty -> Název veličiny
float TypeOfInputCh8	Input Channels->Type of Input Ch 8.
float TypeOfInputCh7	Input Channels->Type of Input Ch 7.
float TypeOfInputCh6	Input Channels->Type of Input Ch 6.
float TypeOfInputCh5	Input Channels->Type of Input Ch 5.
float TypeOfInputCh4	Input Channels->Type of Input Ch 4.
float TypeOfInputCh3	Input Channels->Type of Input Ch 3.
float TypeOfInputCh2	Input Channels->Type of Input Ch 2.
float TypeOfInputCh1	Input Channels->Type of Input Ch 1.
Char InfoTextCh8[256]	Input Channels->Info Text Ch 8.
Char InfoTextCh7[256]	Input Channels->Info Text Ch 7.
Char InfoTextCh6[256]	Input Channels->Info Text Ch 6.
Char InfoTextCh5[256]	Input Channels->Info Text Ch 5.
Char InfoTextCh4[256]	Input Channels->Info Text Ch 4.
Char InfoTextCh3[256]	Input Channels->Info Text Ch 3.
Char InfoTextCh2[256]	Input Channels->Info Text Ch 2.
Char InfoTextCh1[256]	Input Channels->Info Text Ch 1.
float Clipping	Control Parameters -> Width
float ConnctionWithSlipTable	Control Parameters -> Connection with Slip Table
float MassTestFixture	Control Parameters ->Mass, Test Fixture
float MassTestItem	Control Parameters ->Mass, Test Item
float MultiChannelControl	Control Parameters ->Shock-Distance
float Frequency[101]	Reference Curve -> Frequency
float Acceleration[101]	Reference Curve -> Acceleration
float PocetTabulek	Počet naměřených tabulek Data *

5.2 Tabulka TargetCurve pro náhodné vibrace

V tabulce TargetCurve je 7 sloupců se 100 řádky. Názvy sloupců od levého okraje tabulky jsou **WarnLower_Y**, **WarnUpper_Y**, **AbortLower_Y**, **AbortUpper_Y**, **Y_Data**, **X_Data** a **Index**. Tato tabulka je při tvorbě prázdná a hodnoty se zadávají postupně, nepřepisují se. Každý řádek představuje nastavení jednoho měřicího bodu. Hodnoty v původním softwaru se do této tabulky ukládaly z karty **Random Reference Spectrum**, kterou vidíme na obrázek 5.1.



Obrázek 5.1 Screen karty Random Reference Spectrum pro náhodné vibrace

Do sloupečků **WarnLower_Y**, **WarnUpper_Y**, **AbortLower_Y**, **AbortUpper_Y** se neukládají přímo hodnoty, které zadáváme v SW, ale jejich přepočítání dle vzorců ukázaných v tabulce 5.3. Tento SW interface tento přepočítání provádí, takže hodnoty zadáváme v intervalech, které jsou uvedeny v tabulce 5.2. Tabulky sice odkazují na sinusové vibrace, ale zde nedochází ke změnám, proto jsem v popisu použil tyto tabulky.

Tabulka 5.2 Popis dat z tabulky TargetCurve pro náhodné vibrace

Název kolonky v SW (Reference Curve)	Název v sloupci v DB (TargetCurve)	Interval hodnot / jednotky
Warm Limit neg.	WarnLower_Y	<-40 ; 40> [dB]
Warm Limit pos.	WarnUpper_Y	<-40 ; 40> [dB]
Abort Limit neg.	AbortLower_Y	<-40 ; 40> [dB]
Abort Limit pos.	AbortUpper_Y	<-40 ; 40> [dB]
Amplitude	Y_Data	<0.01 ; 10 000> [(m/s ²) ² /Hz]
Frequency	X_Data	<0.01 ; 10 000> [Hz]
Řádek	Index	<1 ; 100> - celá čísla

Metoda pro zapisování je `SetTargetCurve(int Index, float Amplitude, float Frequency, float WarnLimitneg, float WarnLimitpos, float AboutLimitneg, float AboutLimitpos)`. Název parametrů vychází z názvu proměnných, které zadáváme v originálním SW. Přehled můžeme vidět v tabulce 5.2. Metoda kontroluje, zda byl dobře

zadán index, aby se nepřeskakovaly řádky, frekvenci jestli nebyla zadaná dvakrát stejná hodnota a u všech parametrů kontroluje interval hodnot. : V případě, že nalezne nesrovnalost, generuje výjimku s hlášením co bylo zadáno špatně.

Tabulka 5.3 Vzorce pro přepočítání v tabulce TargetCurve pro náhodné vibrace

Název v sloupci v DB (TargetCurve)	Vzorec přepočtu
WarnLower_Y	$= \text{Acceleration} * 10^{\frac{-\text{Warm Limit neg.}}{10}}$
WarnUpper_Y	$= \text{Acceleration} * 10^{\frac{\text{Warm Limit pos.}}{10}}$
AbortLower_Y	$= \text{Acceleration} * 10^{\frac{-\text{Abort Limit neg.}}{10}}$
AbortUpper_Y	$= \text{Acceleration} * 10^{\frac{\text{Abort Limit pos.}}{10}}$

5.3 Tabulka SystemDataText pro náhodné vibrace

V tabulce jsou dva sloupce **Index** a **SystemDataText**. Index má číslování od 1 do 16, takže v tabulce je 16 řádků. Více informací o této kapitole je v podkapitole 4.3. Nás zajímají jen názvy channelů z kolony Info Text, které zadáváme v kartě Input Channels.

Tuto kartu vidíme na obrázku 5.2. Na jaké řádky se dané názvy channelů ukládají a jaké metody používáme k uložení názvů do třídy **Random** vidíme v tabulce 5.4.

Tabulka 5.4 Přehled veličiny Info Text a její metody pro náhodné vibrace

Index	Název kolony v SW	Název metody
9	Info Text Ch.8	SetInfoTextCh8(char InfoTextCh8)
10	Info Text Ch.7	SetInfoTextCh7(char InfoTextCh7)
11	Info Text Ch.6	SetInfoTextCh6(char InfoTextCh6)
12	Info Text Ch.5	SetInfoTextCh5(char InfoTextCh5)
13	Info Text Ch.4	SetInfoTextCh4(char InfoTextCh4)
14	Info Text Ch.3	SetInfoTextCh3(char InfoTextCh3)
15	Info Text Ch.2	SetInfoTextCh2(char InfoTextCh2)
16	Info Text Ch.1	SetInfoTextCh1(char InfoTextCh1)

5.4 Tabulka SystemDataFloat pro náhodné vibrace

I tato tabulka stejně jako u sinusových vibrací obsahuje většinu nastavitelných dat, která mají vliv na průběh měření. Obsahuje 2 sloupce **SystemDataFloat** a **Index**. Tabulka má 149 řádků.

První část odpovídá doslova kapitole 4.4.1, proto se o ní tady nebudu rozepisovat. I jméno metody a její parametry odpovídají, ale je součástí třídy **Random**.

5.4.1 Karta Input Channels pro náhodné vibrace

Karta **Input Channels** se dost podobá kartě ze sinusových vibrací, ale má jisté změny. Například tu chybí kolonky pro zadávání jednotek **Units**. Tím se mění pořadí zadávaných prvků do databáze.

Obrázek 5.2 Screen karty Input Channels pro náhodné vibrace

Veličina **Type of Input** je opět **výběrová veličina**, která může nabývat 4 hodnoty. Pro možnost **off** = 0, **abrot** = 1, **control** = 2 a **measure** = 3. V tabulce 5.5 je znázorněno do jakých řádků se **Type of Input** ukládá.

Tabulka 5.5 Přehled veličiny **Type of Input** pro náhodné vibrace

Index	Název kolonky v SW	Interval hodnot
21	Ch.8->Type of Input	< 0 or 1 or 2 or 3 >
22	Ch.7->Type of Input	< 0 or 1 or 2 or 3 >
23	Ch.6->Type of Input	< 0 or 1 or 2 or 3 >
24	Ch.5->Type of Input	< 0 or 1 or 2 or 3 >
25	Ch.4->Type of Input	< 0 or 1 or 2 or 3 >
26	Ch.3->Type of Input	< 0 or 1 or 2 or 3 >
27	Ch.2->Type of Input	< 0 or 1 or 2 or 3 >
28	Ch.1->Type of Input	< 0 or 1 or 2 or 3 >

Kolonka **Sensitivity** (citlivost) je popsána v tabulce 5.6.

Další parametr, který se zadává je **Limits**, kde zadáváme **Lower** (dolní) a **Upper** (horní) meze dané limity. Pro hodnoty **Type of Input** = **Measure** a **Control** platí tabulka 5.7. Když se **Type of Input** = **Abort**, platí tabulka 5.8, do databázové souboru u

Type of Input = Abort se ukládají hodnoty pro **Limits** vynásobené 1000, než které jsem zadali. Tento přepočítá softwarový interface sám.

Tabulka 5.6 Přehled veličiny Sensitivity pro náhodné vibrace

Index	Název kolonky v SW	Interval hodnot
29	Ch.8->Sensitivity	<0.001 ; 10000>
30	Ch.7->Sensitivity	<0.001 ; 10000>
31	Ch.6->Sensitivity	<0.001 ; 10000>
32	Ch.5->Sensitivity	<0.001 ; 10000>
33	Ch.4->Sensitivity	<0.001 ; 10000>
34	Ch.3->Sensitivity	<0.001 ; 10000>
35	Ch.2->Sensitivity	<0.001 ; 10000>
36	Ch.1->Sensitivity	<0.001 ; 10000>

Tabulka 5.7 Přehled veličiny Limits pro Type of Input = Measure a Control u náhodných vibrací

Index	Popis / Název kolonky v SW	Interval hodnot
37	Ch.8->Limits->Upper	<0.01 ; 10000>
38	Ch.7->Limits->Upper	<0.01 ; 10000>
39	Ch.6->Limits->Upper	<0.01 ; 10000>
40	Ch.5->Limits->Upper	<0.01 ; 10000>
41	Ch.4->Limits->Upper	<0.01 ; 10000>
42	Ch.3->Limits->Upper	<0.01 ; 10000>
43	Ch.2->Limits->Upper	<0.01 ; 10000>
44	Ch.1->Limits->Upper	<0.01 ; 10000>
53	Ch.8->Limits->Lower	<0.01 ; 10000>
54	Ch.7->Limits->Lower	<0.01 ; 10000>
55	Ch.6->Limits->Lower	<0.01 ; 10000>
56	Ch.5->Limits->Lower	<0.01 ; 10000>
57	Ch.4->Limits->Lower	<0.01 ; 10000>
58	Ch.3->Limits->Lower	<0.01 ; 10000>
59	Ch.2->Limits->Lower	<0.01 ; 10000>
60	Ch.1->Limits->Lower	<0.01 ; 10000>

V tabulce 5.8 je přehled metod, které se používají pro zapsání dat z karty Input Channels do třídy **Random**, která se pak může zapsat do databázového souboru pomocí funkce uvedené v kapitole 5.1.3.

Kolonka **Info Text**, která je poslední kolonkou, kterou vidíme na obrázku 5.2 je popsána v podkapitole 5.3.

Tabulka 5.8 Přehled veličiny Limits pro Type of Input = Abort u náhodných vibrací

Index	Název kolonky v SW	Interval hodnot
69	Ch.8->Limits->Upper	<0.001 ; 10>
70	Ch.7->Limits->Upper	<0.001 ; 10>
71	Ch.6->Limits->Upper	<0.001 ; 10>
72	Ch.5->Limits->Upper	<0.001 ; 10>
73	Ch.4->Limits->Upper	<0.001 ; 10>
74	Ch.3->Limits->Upper	<0.001 ; 10>
75	Ch.2->Limits->Upper	<0.001 ; 10>
76	Ch.1->Limits->Upper	<0.001 ; 10>
77	Ch.8->Limits->Lower	<0.001 ; 10>
78	Ch.7->Limits->Lower	<0.001 ; 10>
79	Ch.6->Limits->Lower	<0.001 ; 10>
80	Ch.5->Limits->Lower	<0.001 ; 10>
81	Ch.4->Limits->Lower	<0.001 ; 10>
82	Ch.3->Limits->Lower	<0.001 ; 10>
83	Ch.2->Limits->Lower	<0.001 ; 10>
84	Ch.1->Limits->Lower	<0.001 ; 10>

Tabulka 5.9 Přehled metod pro kartu Input Channels u náhodných vibrací

Channel	Název metody
Channel 1	SetInputChannel18(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 2	SetInputChannel17(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 3	SetInputChannel16(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 4	SetInputChannel15(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 5	SetInputChannel14(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 6	SetInputChannel13(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 7	SetInputChannel12(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 8	SetInputChannel11(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)

5.4.2 Karta Set Level Table pro náhodné vibrace

Od řádku 93 do řádku 132 jsou uložena data z karty Set Level Table. V tabulce 5.10 je znázorněno, kam se jaká hodnota ukládá a jaká je metoda pro zapsání do třídy [Random](#).

V této kartě se dají zadávat vždy 2 hodnoty, které jsou spolu svázané **Time** a **Level**. Do sloupečku Time se ukládají na index, který je uveden ve sloupečku nalevo od

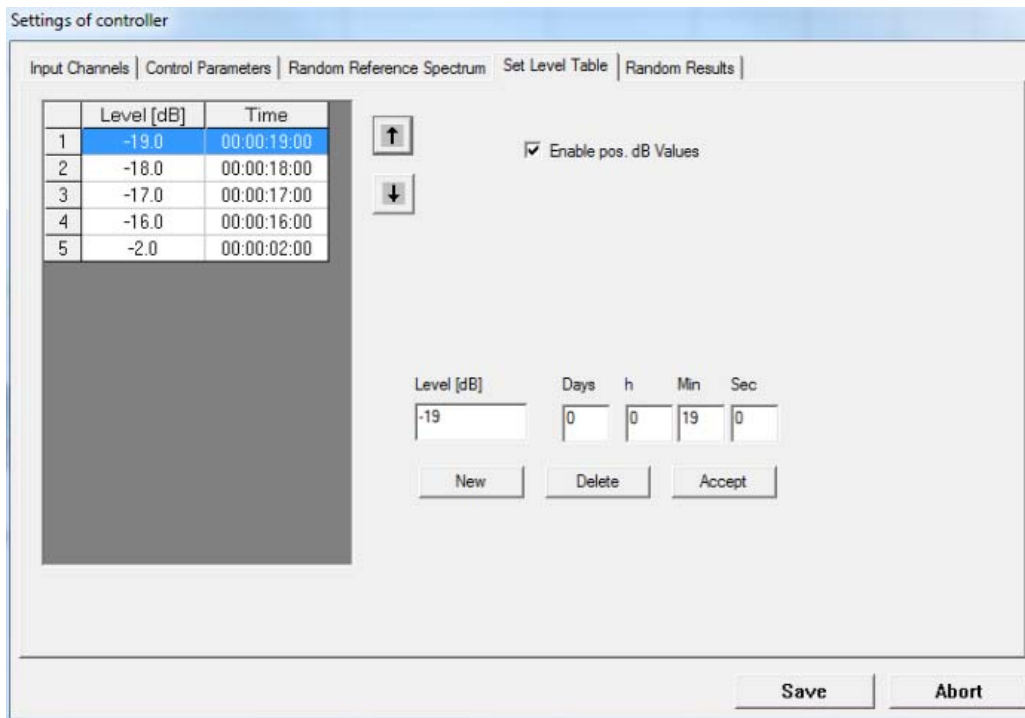
sloupečku Time. U sloupečku Level to platí také, že se ukládá na index, který je nalevo od sloupečku Level. Jak jsem psal vždy se zadávají tyto hodnoty společně, tedy Time1 a Level1 se zadávají dohromady a v naprogramovaném interface nám k tomu slouží jedna metoda. Názvy metod vidíme v posledním sloupečku v tabulce 5.8. Metody mají 2 **floatové** parametry. Parametr Time má interval hodnot $< 1 ; 3\ 628\ 800 >$ v sekundách , tedy maximálně 42 dnů. Level může nabývat hodnot $< -20 ; 0 >$ v deciBelech.

Kdybychom chtěli hodnoty přepisovat, musíme začít zapisovat znovu hodnoty od prvního řádku.

Poslední věc na této kartě je zatržítka **Enable pos. dB Values**. Ukládá se na pozici 144. Metody pro zapsání je SetEneblePosDBValues (**float** EnablePosDBValues). Nabývá hodnotu **0 = false** a **1 = true**.

Tabulka 5.10 Přehled veličin z karty Set Level Table a jejich metody pro náhodné víbrace

Index	Time [s]	Index	Level [dB]	Název metody
93	Time 1	113	Level 1	SetSetLevelTableLine1(float Time, float Level);
94	Time 2	114	Level 2	SetSetLevelTableLine2(float Time, float Level);
95	Time 3	115	Level 3	SetSetLevelTableLine3(float Time, float Level);
96	Time 4	116	Level 4	SetSetLevelTableLine4(float Time, float Level);
97	Time 5	117	Level 5	SetSetLevelTableLine5(float Time, float Level);
98	Time 6	118	Level 6	SetSetLevelTableLine6(float Time, float Level);
99	Time 7	119	Level 7	SetSetLevelTableLine7(float Time, float Level);
100	Time 8	120	Level 8	SetSetLevelTableLine8(float Time, float Level);
101	Time 9	121	Level 9	SetSetLevelTableLine9(float Time, float Level);
102	Time 10	122	Level 10	SetSetLevelTableLine10(float Time, float Level);
103	Time 11	123	Level 11	SetSetLevelTableLine11(float Time, float Level);
104	Time 12	124	Level 12	SetSetLevelTableLine12(float Time, float Level);
105	Time 13	125	Level 13	SetSetLevelTableLine13(float Time, float Level);
106	Time 14	126	Level 14	SetSetLevelTableLine14(float Time, float Level);
107	Time 15	127	Level 15	SetSetLevelTableLine15(float Time, float Level);
108	Time 16	128	Level 16	SetSetLevelTableLine16(float Time, float Level);
109	Time 17	129	Level 17	SetSetLevelTableLine17(float Time, float Level);
110	Time 18	130	Level 18	SetSetLevelTableLine18(float Time, float Level);
111	Time 19	131	Level 19	SetSetLevelTableLine19(float Time, float Level);
112	Time 20	132	Level 20	SetSetLevelTableLine20(float Time, float Level);



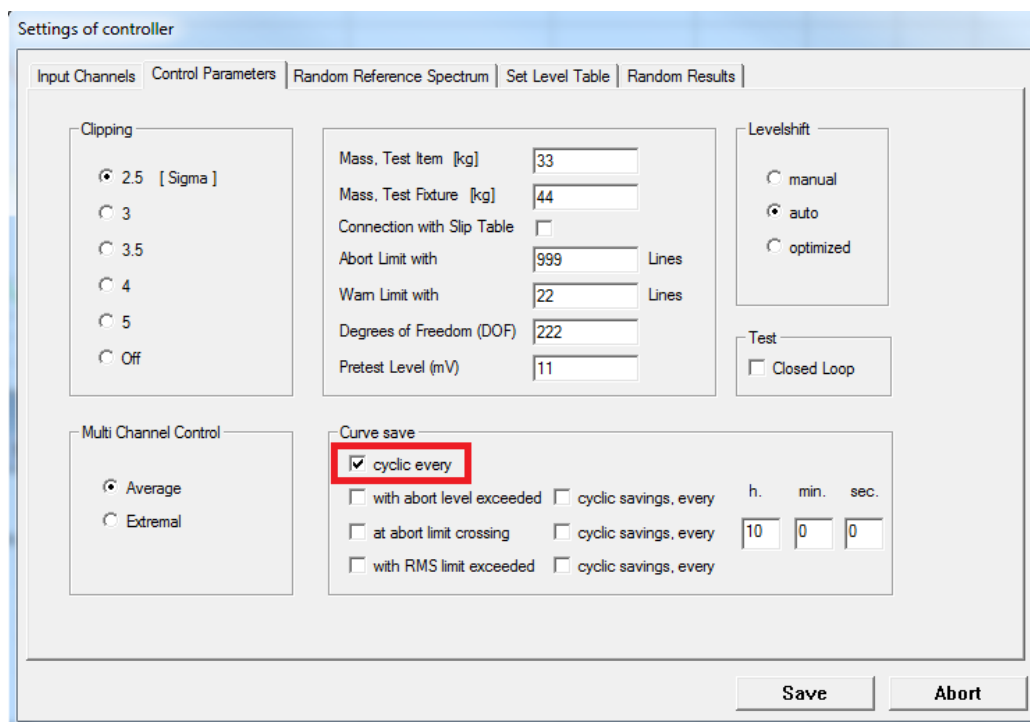
Obrázek 5.3 Screen karty Set Level Table pro náhodné vibrace

5.4.3 Karta Control Parameters pro náhodné vibrace

Hodnota pro výběrovou veličinu **Clipping** ukládá hodnoty, které vidíme v tabulce 5.11 nebo na obrázku 5.4. Pro možnost **Off** se do řádku k indexu 138 uloží hodnota 6. Pro ostatní hodnoty se ukládá vybraná hodnota **Sigmy**.

Multi Channel Control ukládá 0 = **Avarage** a 1 = **External**. Zatřítko **Connection with Slip Table** je opět 0 = **false** a 1 = **true**. **Levelshift** nabývá hodnoty 0 = **manual**, 1 = **auto** a 2 = **optimizid**. Zatřítko **test** nejde v originálním SW zatrnout a uložit, tak ani naprogramovaný SW interface s touto proměnou nepracuje. Kombinace zatřítetek **Curve save** opět nabývá různé hodnoty, které jsou uvedeny v tabulce 5.12.

Enabla pos. dB Values je popsáno v předchozí podkapitole 5.4.2 a veličina **Selection of Bandwidth** se zadává v kartě **Random Reference Spectrum**. Ale do tabulky se ukládají mezi ostatní hodnoty z karty **Control Parameters**, tak jsem je do tabulek popsal zde. Screen této karty můžeme vidět na obrázku 5.4. Každá z metod má jen jeden parametr a to veličinu, kterou chceme zapsat do třídy **Random**.



Obrázek 5.4 Screen karty Control Parameters pro náhodné vibrace

Tabulka 5.11 Přehled veličin z karty Control Parameters pro náhodné vibrace

Index	Nazev promené v SW	Interval hodnot	Metody
133	Warm limit with Lines	< 1 ; 1000 >	SetWarmLimitWithLines(float p);
134	Abort Limit with Lines	< 1 ; 1000 >	SetAbortLimitWithLines(float p)
135	Selection of Bandwidth z karty Random Reference Spectrum	< 2000 or 3000 or 5000 >	SetSelectionofBandwidth(float p)
136	Multi Channel Control	< 0 or 1 >	SetMultiChannelControl(float p)
137	Počet zadaných řádků v Set Level Table	Automaticky	-
138	Clipping	< 2.5 or 3 or 3.5 or 4 or 5 or 6 >	SetClipping(float p)
139	Degrees of Freedom (DOF)	< 10 ; 1000 >	SetDefreesofFreedom(float p)
142	Levelshift	< 0 or 1 or 2 >	SetLevelShift(float p)
143	Mass, Test Item	< 0 ; 300 >	SetMassTestItem(float p)
144	Enebla pos. dB Values z karty Set Level Table	< 0 or 1 >	SetEneblePosDBValues(float p)
145	Pretest Level	< 1 ; 300 >	SetPretestLevel(float p)
146	Curve Save Time	< 0 ; 86 400 >	SetTimeMeasure(float p)
147	Curve Save zatržítka	Viz tabulka 5.10	SetCurveSave(float p)
148	Mass, Test Fixture	< 0 ; 300 >	SetMassTestFixture(float p)
149	Connection with Slip Table	< 0 or 1 >	SetConnctionWithSlipTable(float p)

Tabulka 5.12 Seznam hodnot pro Curve Save na indexu 147

Hodnota s Cyclic saving	Hodnota bez Cyclic saving	Název zatržítka
	1.401298e-45	Cyclic every
2.662467e-44	4.203895e-45	Cyclic every + With abort level exceeded
1.022948e-43	1.261169e-44	Cyclic every + At about limit crossing
5.184804e-44	7.006492e-45	Cyclic every With RMS limit exceeded
1.275182e-43	1.541428e-44	Cyclic every +With abort level exceeded + At about limit crossing
7.707142e-44	9.809089e-45	Cyclic every + With abort level exceeded + With RMS limit exceeded
1.527415e-43	1.821688e-44	Cyclic every + At about limit crossing + With RMS limit exceeded
1.779649e-43	2.101948e-44	Cyclic every + With abort level exceeded + At about limit crossing + With RMS limit exceeded

5.5 Tabulka Logbook, Drive, DiagrammStyleText a DiagrammStyleFloat pro náhodné vibrace

Názvy tabulek z názvu podkapitoly zde nebudu už tolik podrobně popisovat, protože jsem je popsal v kapitole pro sinusové vibrace. Tyto tabulky se buď neliší v ukládaných datech vůbec nebo jen minimálně.

Tabulka **Logbook** je popsána v podkapitole 4.5 a společně s tabulkou **Drive**, která je popsána v podkapitole 4.7 se neliší vůbec.

DiagrammStyleText se liší pouze v tom, že náhodné vibrace se v originálním SW vyhodnocují jen na dvou grafech, takže tabulka má jen 3 sloupce a to **DiagrammStyleText1**, **DiagrammStyleText2** a **Index**. Podrobně je popsána v podkapitole 4.8.

DiagrammStyleFloat je opět to samé jako **DiagrammStyleText**. Jelikož obsahuje pouze dva grafy, ubydou jí opět dva sloupce. Podrobně je popsán v kapitole 4.9.

Ani jedna z těchto tabulek není potřeba ke tvorbě Softwarového interfacu.

5.6 Tabulka GraphStyle pro náhodné vibrace

Tabulka **GraphStyle** nese informaci o barvách křivek a os zobrazovaných grafů v originálním softwaru pro ovládání zkušební laboratoře CVVOZE. Stejný druh informací, kterou jsem popisoval v kapitole 4.6. obsahuje 9 sloupečků a to **Farbe**, **Dicke**, **D1**, **D2**, **D3**, **LegendeText**, **LegendeFarbe**, **LegendeDicke** a **Index**. Má 69 řádků, ale data uložená v této tabulce pro tvorbu softwarového interfacu nejsou důležitá. V počtu řádků je ale první rozdíl, způsobený tím, že náhodné vibrace mají v originální SW k zobrazení výsledku jen dva grafy. Řádek 35 je řádkem oddělovacím a nenese žádnou informaci, pouze odděluje data z grafu jedna od grafu dva. Zase jsou rozdělené informace do čtyř bloků, každý blok představuje informace o jednom z grafu pro tisk nebo pro zobrazení v originálním SW.

6. RÁZY

Tuto kapitolu věnuji nastavení a datům v databázovém souboru pro rázy pro testovací zařízení zkušební laboratoře CVVOZE. Vysvětlím zde funkce a metody, které slouží pro vytvoření, přepis a čtení dat v databázovém souboru (*.shk).

6.1 Základní funkce pro rázy

Pro práci s databázovým souborem (*.shk) potřebujeme hlavičkový soubor *ShockHeader.h*, *ShockFunkce.cpp* a *ShockZaklad.shk*, které nalezneme v přílohách.

6.1.1 Vytvoření databázového souboru pro rázy

Nejprve si vytvoříme databázový soubor pomocí funkce *VytvoreniShock(umistení, zakladni_soubor)*, který má dva parametry. Oba dva jsou typu `char`. Jeden s cestou a názvem nového databázového souboru, například: `"C:\\Slozka_1\\Podslozka_2\\nazev_souboru_1"`. Cesta i s názvem by měla mít maximálně 256 znaků. Jelikož 4 znaky jsou vyhrazeny na koncovku souboru, tak zadávaný `char` by měl mít 252 znaků. Druhý `char` je s cestou k souboru *ShockZaklad.shk*.

6.1.2 Přepis dat do databázového souboru pro rázy

Vytvoříme si proměnnou typu `Shock`. Do která následně pomocí Set metod, které jsou popsány níže v této kapitole nahrajeme data s nastavením. Data projdou kontrolou zda odpovídají požadovanému rozsahu. Pokud by rozsah překračovaly, program generuje výjimku s chybovým hlášením, ve kterém budou zobrazena chybně zadaná data.

Po nahrání dat do třídy `Shock`, zavoláme funkci *PrepisShock(umistení, tridaShock)*, která tyto data nahraje do databázového souboru.

Parametr *umistení* je typu `char`, který obsahuje cestu k souboru, například: `"C:\\Slozka_1\\Podslozka_2\\nazev_souboru_1"`.

Druhým parametrem je třída typu `Shock`.

6.1.3 Čtení dat z databázového souboru pro rázy

Pro čtení dat jsou určeny tři funkce.

První funkce vytáhne z databázového souboru celou jednu tabulku, tedy `Data_*`, kde `*`, představuje číslo tabulky. Tato funkce je *CteniTabulkyShock(umistení, tabulka, data)*. Parametr *umistení* je typu `char`, který obsahuje cestu k souboru, například: `"C:\\Slozka_1\\Podslozka_2\\nazev_souboru_1"`. Parametr *tabulka* je typu `int` a určuje nám kolikátou tabulku má vytáhnout z databázového souboru. Poslední parametr je struktura typu `MereniData`, kterou si nejprve vytvoříme a potom předáme do funkce jako parametr. Do této struktury se nám nahraje celé jedno měření. Tato struktura obsahuje 11 polí `floatů`, které představují vždy jeden sloupeček dat z jedné tabulky měření. Názvy polí ve struktuře odpovídají názvům sloupečků v databázovém souboru. Index (řádek) v tabulce vždy odpovídá indexu v poli, takže v polích na pozici 0 se nahraje hodnota „0“ pro lepší orientaci. Názvy sloupečků můžeme vidět v tabulce 4.1. **Měřená data** v tabulkách `Data_*` se ukládají až od indexu 4 do indexu 1027. Počet naměřených tabulek `Data_*` nám přečte třetí popisovaná funkce v této podkapitole.

Druhá funkce načte vždy jeden řádek o určitému indexu ze všech naměřených tabulek. Funkce se nazývá *CteniRadkuShock(umisteni, index, pole)*. Parametr *umisteni* je opět typu **char**, který je ve stejném formátu jako u předchozí funkce. Parametr *index* je typu **int** a určuje kolikátý parametr chce z tabulek vyčíst. Poslední parametr funkce je **floatové** pole polí, kde první index představuje z kolikáté tabulky data jsou a druhý index představuje název sloupečku. Názorná ukázka v tabulce 4.1. Pole, kterému budeme funkci předávat má mít velikost 100*11. Pro lepší orientaci v indexech, aby číslo měřené tabulky opravdu odpovídalo indexu prvního z polí, tak první pole má v sobě nahané nuly.

Třetí funkce načte z databázové souboru nastavení pro zkušební laboratoř, které je důležité pro sestavení protokolu. Tato funkce je *CteniNastaveniShock(umisteni, NastavenaData)*. Má dva parametry, **umisteni** typu **char**, který obsahuje cestu k souboru jako v předchozích funkcích a druhý parametr typu **NastavenaData**, do kterých se uloží hodnoty z databázového souboru. Structura **NastavenaData** obsahuje proměnné popsané v tabulce 6.1.

Tabulka 6.1 Seznam proměnných struktury **NastavenaData pro rázy**

Názvy proměnných struktury NastavenaData	Název karty -> Název veličiny
float TypeOfInputCh8	Input Channels->Type of Input Ch 8.
float TypeOfInputCh7	Input Channels->Type of Input Ch 7.
float TypeOfInputCh6	Input Channels->Type of Input Ch 6.
float TypeOfInputCh5	Input Channels->Type of Input Ch 5.
float TypeOfInputCh4	Input Channels->Type of Input Ch 4.
float TypeOfInputCh3	Input Channels->Type of Input Ch 3.
float TypeOfInputCh2	Input Channels->Type of Input Ch 2.
float TypeOfInputCh1	Input Channels->Type of Input Ch 1.
char InfoTextCh8[256]	Input Channels->Info Text Ch 8.
char InfoTextCh7[256]	Input Channels->Info Text Ch 7.
char InfoTextCh6[256]	Input Channels->Info Text Ch 6.
char InfoTextCh5[256]	Input Channels->Info Text Ch 5.
char InfoTextCh4[256]	Input Channels->Info Text Ch 4.
char InfoTextCh3[256]	Input Channels->Info Text Ch 3.
char InfoTextCh2[256]	Input Channels->Info Text Ch 2.
char InfoTextCh1[256]	Input Channels->Info Text Ch 1.
float CurveWidth	Reference Curve -> Width
float CurveShockInverted	Reference Curve -> Shock Inverted
float CurveType	Reference Curve -> Curve Type
float CurveAmplitude	Reference Curve -> Amplitude
float ConnctionWithSlipTable	Control Parameters -> Connection with Slip Table
float MassTestFixture	Control Parameters ->Mass, Test Fixture
float MassTestItem	Control Parameters ->Mass, Test Item
float ShockDistance	Control Parameters ->Shock-Distance
float ShockNumberTotal	Control Parameters ->Shock Number
float PreShockLevel	Control Parameters -> Pre-Shock Level
float PocetTabulek	Počet naměřených tabulek Data *

6.2 Tabulky UserToleranceband a UserCurveType pro rázy

Data v těchto dvou tabulkách nejdu z originálního SW měnit. Popíši zde jen názvy sloupečků a kolik mají řádků.

Tabulka **UserToleranceband** má sloupečky **CurveNo**, **CurveType**, **StandardNo**, **Standard**, **Time**, **AbortPos**, **AbortNeg**, a **Index**. Tato tabulka má 6 řádků.

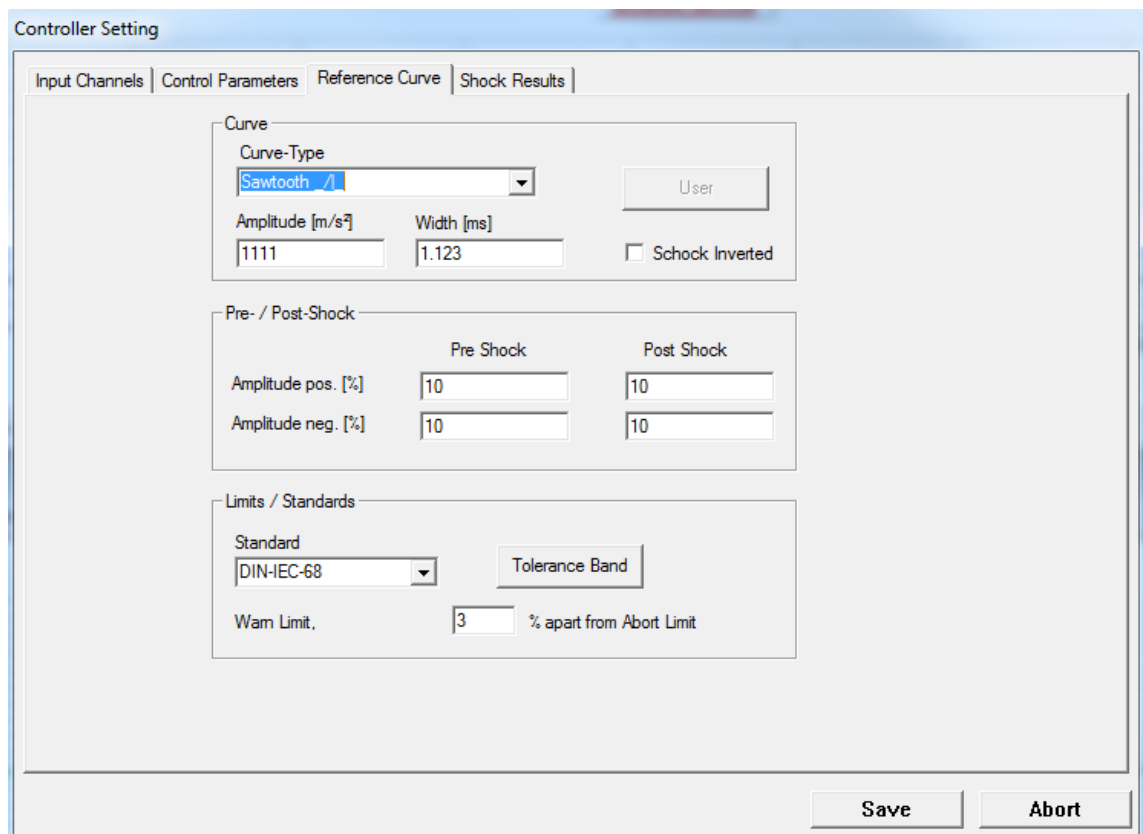
Tabulka **UserCurveType** obsahuje **UserTime**, **UserAmplitude** a **Index** s 13 řádky.

6.3 Tabulka User_Pre_PostPulses pro rázy

Tato tabulka nese informace o hodnotách z oddílu Pre-/Post Shock z karty Reference Curve, kterou vidíme na obrázku 6.1. Má 9 sloupečků a 16 řádků.

První čtyři sloupečky (**CurveNo**, **CurveType**, **StandartNo** a **Standart**) určují kombinaci na jaký řádek se hodnoty z **Pre-/Post-Shock** uloží. CurveType a Standard mají uložené slovní názvy typu a standartu. Ve sloupečcích **CurveNo** a **StandartNo** je uloženo číslo, které je k nim přiřazeno. Tyto hodnoty se nedají změnit.

Další čtyři sloupečky (**PrePositive**, **PreNegative**, **PostPositive** a **PostNegative**) uchovávají informaci z bloku Pre-/Post-Shock, kterou můžeme vidět na obrázku 6.1. Do jakého sloupečku se hodnota ukládá vidíme v tabulce 6.2. Poslední sloupeček je index, který není nutno popisovat. Metoda pro zapsání dat do třídy **Shock** se nazývá **SetPrePostShock(float PreShockPos, float PreShockNeg, float PostShockPos, float PostShockNeg)**. Názvy parametrů odpovídají veličinám na obrázku 6.1.



Obrázek 6.1 Screen karty Reference Curve pro rázy

Tabulka 6.2 Popis dat Pre/-Post Shock z karty Reference Curve pro rázy

Název sloupečku v DB	Název veličiny z originálního SW	Interval hodnot
CurveNo	Číslo přiřazené k Curve Type	Není možno přepsat
CurveType	Curve Type	Není možno přepsat
StandardNo	Číslo přiřazené Standardu	Není možno přepsat
Standard	Standard	Není možno přepsat
PrePositive	Amplitude pos. [%] Pre Shock	<1 ; 100>
PreNegative	Amplitude neg. [%] Pre Shock	<1 ; 100>
PostPositive	Amplitude pos. [%] Post Shock	<1 ; 100>
PostNegative	Amplitude neg. [%] Post Shock	<1 ; 100>
Index		Není možno přepsat

6.4 Tabulka TargetCurve pro rázy

Do tabulky TargetCurve pro rázy se nedají z originálního SW zapsat žádná data. Obsahuje, jako všechny předchozí tabulky TargetCurve, 7 sloupečků s názvy **WarnLower_Y**, **WarnUpper_Y**, **AbortLower_Y**, **AbortUpper_Y**, **Y_Data**, **X_Data** a **Index**. Má 100 řádků, ale data rozdílná od nuly obsahují jen první 4 řádky. Tato tabulka pro rázy při tvorbě SW interfacu není podstatná.

6.5 Tabulka SystemDataText pro rázy

V tabulce jsou dva sloupečky **Index** a **SystemDataText**. Index má číslování od 1 do 22, to znamená, že v tabulce je 22 řádků. Více informací o této kapitole je v podkapitole 4.3. Nás zajímají jen názvy channelů z kolonky Info Text, které zadáváme v kartě Input Channels. Tuto kartu vidíme na obrázku 6.2. Na jaké řádky se dané názvy channelů ukládají a jaké metody používáme k uložení názvů do třídy **Shock** vidíme v tabulce 6.3.

Tabulka 6.3 Přehled veličiny Info Text a její metody pro rázy

Index	Název kolonky v SW	Metoda
15	Info Text Ch.8	SetInfoTextCh8(char InfoTextCh8)
16	Info Text Ch.7	SetInfoTextCh7(char InfoTextCh7)
17	Info Text Ch.6	SetInfoTextCh6(char InfoTextCh6)
18	Info Text Ch.5	SetInfoTextCh5(char InfoTextCh5)
19	Info Text Ch.4	SetInfoTextCh4(char InfoTextCh4)
20	Info Text Ch.3	SetInfoTextCh3(char InfoTextCh3)
21	Info Text Ch.2	SetInfoTextCh2(char InfoTextCh2)
22	Info Text Ch.1	SetInfoTextCh1(char InfoTextCh1)

6.6 Tabulka SystemDataFloat pro rázy

Tabulka SystemDataFloat obsahuje většinu dat sloužících k nastavení měřicí vibrační stanice CVVOZE. Obsahuje 2 sloupce **SystemDataFloat** a **Index**. Tato tabulka má 1140 řádků. Nastavovaná data se ukládají jen do řádku 116. Zbylé nastavení se nedá z originálního softwaru změnit, proto se jím nebudeme zabývat. První část tabulky je opět stejná jako v předchozích dvou případech a informace o této části najdeme v podkapitole 4.4.1

6.6.1 Karta Input Channels pro rázy

Způsob uložení dat do databázového souboru pro kartu Input Channels pro rázy se, liší od předchozích dvou způsobů. Data se ukládají v blocích channelů a ne jako v předchozích případech, kdy se ukládaly pohromadě vždy jedna veličina pro všechny channely. Kartu Input channels pro rázy můžeme vidět na obrázku 6.2.

Veličina Type of Input je **výběrová veličina**, která může nabývat 4 hodnoty. Pro možnost **off** = 0, **abort** = 1, **control** = 2 a **measure** = 3. V tabulce 6.4 je znázorněno do jakých řádků se Type of Input ukládá. Další změna je to, že pro **channely 5, 6, 7, 8** se **nedá** zadávat možnost pro **Type of Input = Control**.

Channel	Type of Input	Sensitivity mV/m/s ²	Limits		Info Text
			Lower	Upper	
Ch.1	Off				
Ch.2	Abort		3 V	4 V	Channel X - 4321/979268 Y axis -
Ch.3	Control	111	5 m/s ²	6 m/s ²	
Ch.4	Off				
Ch.5	Measure	333	444 m/s ²	555 m/s ²	
Ch.6	Off				
Ch.7	Abort		8 V	9 V	
Ch.8	Measure	10.98	0.9999 m/s ²	999.99 m/s ²	

Obrázek 6.2 Screen karty Input Channels pro rázy

Tabulka 6.4 Přehled veličiny Type of Input pro rázy

Index	Název kolonky v SW	Interval hodnot
22	Ch.8->Type of Input	< 0 or 1 or 3 >
31	Ch.7->Type of Input	< 0 or 1 or 3 >
40	Ch.6->Type of Input	< 0 or 1 or 3 >
49	Ch.5->Type of Input	< 0 or 1 or 3 >
58	Ch.4->Type of Input	< 0 or 1 or 2 or 3 >
67	Ch.3->Type of Input	< 0 or 1 or 2 or 3 >
76	Ch.2->Type of Input	< 0 or 1 or 2 or 3 >
85	Ch.1->Type of Input	< 0 or 1 or 2 or 3 >

Tabulka 6.5 Přehled veličiny Sensitivity pro rázy

Index	Název kolonky v SW	Interval hodnot
23	Ch.8->Sensitivity	<0.001 ; 10000>
32	Ch.7->Sensitivity	<0.001 ; 10000>
41	Ch.6->Sensitivity	<0.001 ; 10000>
50	Ch.5->Sensitivity	<0.001 ; 10000>
59	Ch.4->Sensitivity	<0.001 ; 10000>
68	Ch.3->Sensitivity	<0.001 ; 10000>
77	Ch.2->Sensitivity	<0.001 ; 10000>
86	Ch.1->Sensitivity	<0.001 ; 10000>

Tabulka 6.6 Přehled veličiny Limits pro Type of Input = Measure a Control pro rázy

Index	Popis / Název kolonky v SW	Interval hodnot
26	Ch.8->Limits->Upper	<0.01 ; 10000>
35	Ch.7->Limits->Upper	<0.01 ; 10000>
44	Ch.6->Limits->Upper	<0.01 ; 10000>
53	Ch.5->Limits->Upper	<0.01 ; 10000>
62	Ch.4->Limits->Upper	<0.01 ; 10000>
71	Ch.3->Limits->Upper	<0.01 ; 10000>
80	Ch.2->Limits->Upper	<0.01 ; 10000>
89	Ch.1->Limits->Upper	<0.01 ; 10000>
27	Ch.8->Limits->Lower	<0.01 ; 10000>
36	Ch.7->Limits->Lower	<0.01 ; 10000>
45	Ch.6->Limits->Lower	<0.01 ; 10000>
54	Ch.5->Limits->Lower	<0.01 ; 10000>
63	Ch.4->Limits->Lower	<0.01 ; 10000>
72	Ch.3->Limits->Lower	<0.01 ; 10000>
81	Ch.2->Limits->Lower	<0.01 ; 10000>
90	Ch.1->Limits->Lower	<0.01 ; 10000>

Kolona **Sensitivity** (citlivost) je popsána v tabulce 6.5.

Další parametry, které se zadávají je **Limits**, kde zadáváme **Lower** (dolní) a **Upper** (horní) mezi dané limity. Pro hodnoty **Type of Input = Measure a Control** platí tabulka 6.6. Když se **Type of Input = Abort**, platí tabulka 6.7. Do databázové souboru u **Type of Input = Abort** se ukládají hodnoty pro **Limits** vynásobené 1000. Tento přepočítá SW interface automaticky.

Tabulka 6.7 Přehled veličiny Limits pro Type of Input = Abort pro rázy

Index	Název kolony v SW	Interval hodnot
69	Ch.8->Limits->Upper	<0.01 ; 10>
70	Ch.7->Limits->Upper	<0.01 ; 10>
71	Ch.6->Limits->Upper	<0.01 ; 10>
72	Ch.5->Limits->Upper	<0.01 ; 10>
73	Ch.4->Limits->Upper	<0.01 ; 10>
74	Ch.3->Limits->Upper	<0.01 ; 10>
75	Ch.2->Limits->Upper	<0.01 ; 10>
76	Ch.1->Limits->Upper	<0.01 ; 10>
77	Ch.8->Limits->Lower	<0.01 ; 10>
78	Ch.7->Limits->Lower	<0.01 ; 10>
79	Ch.6->Limits->Lower	<0.01 ; 10>
80	Ch.5->Limits->Lower	<0.01 ; 10>
81	Ch.4->Limits->Lower	<0.01 ; 10>
82	Ch.3->Limits->Lower	<0.01 ; 10>
83	Ch.2->Limits->Lower	<0.01 ; 10>
84	Ch.1->Limits->Lower	<0.01 ; 10>

Tabulka 6.8 Přehled metod pro kartu Input Channels pro rázy

Channel	Název metody
Channel 1	SetInputChannel18(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 2	SetInputChannel17(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 3	SetInputChannel16(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 4	SetInputChannel15(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 5	SetInputChannel14(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 6	SetInputChannel13(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 7	SetInputChannel12(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)
Channel 8	SetInputChannel11(float TypeOfInput, float Sensitivity, float LimitsUpper, float LimitLower)

V tabulce 6.8 je přehled metod, které se používají pro zapsání dat z karty Input Channels do třídy, která se pak může pomocí funkce uvedené v kapitole 6.1.2 zapsat do databázového souboru.

Kolona **Info Text**, která je poslední kolonkou, kterou vidíme na obrázku 6.2 je popsána v kapitole 6.3.

6.6.2 Karta Control Parameters pro rázy

V této kapitole popíšeme veličiny z karty Control Parameters, kterou vidíme na obrázku 6.3. Seznam parametru s intervaly hodnot, na jaký řádek se ukládají a metody pomocí kterých se zadávají do třídy **Shock** je v tabulce 6.9. Každá metoda má vždy jen jeden parametr a to **floatové** číslo, které se následně uloží do třídy.

Obrázek 6.3 Screen karty Control Parameters pro rázy

Proměnná **Connection with slip Table** nabývá **0** = false a **1** = true.

Pre-Shock Level jak je vidět na obrázku 6.3, ukládá jednu ze 4 možností. **0** = 0.75dB; **1** = 1.5dB ; **2** = 3dB a **3** = 6dB.

Tabulka 6.9 Přehled veličin z karty Control Parameters pro rázy

Index	Název veličiny	Interval hodnot	Metody
100	Pre-Test Level(mV)	< 1 ; 20000 >	SetPretestLevel(float p)
101	No. of Abort Limit violations	< 0 ; 1000 >	SetNoOfAbortLimitViolations(float p)
102	No. of Warn Limit violations	< 0 ; 1000 >	SetNoOfWarnLimitViolations(float p)
109	Shock-Distance	< 0.3 ; 300 >	SetShockDistance(float p)
110	Shock Number (Total)		SetShockNumberTotal(float p)
111	Mass.Test Item [kg]	< 0 ; 300 >	SetMassTestItem(float p)
112	Shocks		SetShocks(float p)
113	Curve Save	Viz tabulka 6.10	SetCurveSave(float p)
114	Pre-Shock Level	< 0 or 1 or 2 or 3>	SetPreShockLevel(float p)
115	Mass.Test Fixture [kg]	< 0 ; 300 >	SetMassTestFixture(float p)
116	Connection with slip Table	< 0 or 1 >	SetConnctionWithSlipTable(float p)

Tabulka 6.10 Seznam hodnot pro zatržitka Curve save z karty Control Parameters pro rázy

Hodnota	Kombinace zatržitek
-431599600	Cycling saving
-431600100	Cycling saving +With abort level exceeded + With acceleration limit exceeded + At abort limit crossing
-431599700	Cycling saving + With abort level exceeded
-431599900	Cycling saving + With acceleration limit exceeded
-431599800	Cycling saving + At abort limit crossing
-431600000	Cycling saving + At abort limit crossing + With abort level exceeded
-431600200	Cycling saving + At abort limit crossing + At abort limit crossing
-431600000	Cycling saving + With acceleration limit exceeded + With abort level exceeded

6.6.3 Karta Reference Curve pro rázy

Tuto kartu můžeme vidět na obrázku 6.1 v podkapitole 6.3, kde jsem popisoval část zadávaných veličin z této karty do tabulky User_Pre_PostPulses. Hodnoty a její metody jsou v tabulce 6.11. Metody opět mají jeden parametr typu **float**, který představuje hodnotu, kterou chceme uložit do třídy **Shock**.

Tabulka 6.11 Přehled veličin pro kartu Reference Curve pro rázy

Index	Název proměnné	Interval hodnot	Metoda
103	Curve - Type	Viz. tabulka 6.12	SetCurveType (float p)
104	Standard	Viz. tabulka 6.12	SetCurveStandard(float p)
105	Amplitude [m/s ²]	< 1 ; 9999 >	SetCurveAmplitude (float p)
106	Width [ms]	< 0.49 ; 100 >	SetCurveWidth (float p)
107	Warm Limit. [%]	< 0 ; 100 >	SetWarnLimit (float p)
108	Shock Inverted	< 0 ; 1 >	SetCurveShockInverted (float p)

Tabulka 6.12 Seznam hodnot pro veličinu Standard a Curve-Type pro rázy

Standart	Hodnota	Curve-Type	Hodnota
MIL-STD-810D	0	Hlaf Sine	0
DIN-IEC-68	1	Sawtooth _/ _	1
VG-95-332	2	Sawtooth _ _	2
Free	3	Triangle	3
		Trapezoid	4
		Free defined	5

6.7 Tabulky Logbook, Drive, DiagrammStyleText, DiagrammStyleFloat a GraphStyle pro rázy

Tyto tabulky se ani v databázovém souboru pro rázy se nezměnily oproti souborům pro sinusové a náhodné vibrace. Tím se tedy odkáží na předchozí kapitoly.

Tabulka Logbook je popsána v podkapitole 4.5, tabulka Drive se nachází v podkapitole 4.7. DiagrammStyleText je v podkapitole 4.8, ve které jsou pouze změněná uložená data, která nesou názvy grafů a os pro rázy. Tabulka DiagrammStyleFloat je popsána v podkapitole 4.9. a tabulka GraphStyle je psána v podkapitole 4.6.

Žádná data z těchto tabulek, ale pro svůj SW interface nepoužívám a jsou tedy pro tuto bakalářskou práci nepodstatné.

ZÁVĚR

Cílem bakalářské práce bylo analyzovat a popsat databázové soubory pro vibrační zařízení od firmy RMS a vytvořit k těmto databázovým souborům obousměrný interface.

Nejprve jsem se musel naučit zacházet s knihovnou ODBC v jazyku C++ a správně nastavit programovací prostředí. Toto nastavení a práci s ODBC knihovnou popisují ve druhé kapitole.

V dalším kroku jsem musel postupně zadokumentovat tři databázové soubory pro rázy, sinusové a náhodné vibrace a v práci je přehledně popsat. Tato práce pak může sloužit, jako dodatková dokumentace k vibračnímu zařízení. Po zdokumentování těchto souborů přišla na řadu tvorba samotného softwarového interfacu. Vytvořený interface je jen konzolového ražení a grafické rozhraní nebylo součástí této bakalářské práce.

Pro každá druh vibrací jsem vytvořil hlavičkový a cplusplusový soubor s 5 funkcemi, které slouží pro vytvoření, přepis a čtení dat z databázového souboru. Při programovém vytváření databázového souboru na základě MS Accessu 2.0 se mi, ale nepodařilo vytvořit soubor, který by šel nahrát do vibračního stolku. Nejpravděpodobnější příčinou tohoto problému bude skutečnost, že knihovna ODBC nedokáže vytvářet úplně identické databáze jako tomu bylo v MS Access 2.0 z roku 1994 a řídicí software vibračního systému bohužel reaguje na tuto změnu formátu chybovou funkčností. Tento problém jsem vyřešil tím, že vytvořím kopii z funkčního databázového souboru. Následně pracuji s touto kopií. Kód s naprogramovanou tvorbou databázového souboru je zakomentovaný v příloženém souboru FunkceSin.cpp. S ostatními funkcemi už nebyly problém.

Softwarový interface byl otestován na vibračním zařízení a s variantou, kde se vytváří kopie databázového souboru funguje bezproblému.

Literatura

- [1] Naučte se C++ za 21 dní. Praha: Computer Press, 2002. ISBN 80-7226-774-4
- [2] SQL Podrobný průvodce uživatele. Praha: Granda Publishing, 2018. ISBN 978-80-271-0774-2.
- [3] Cplusplus [online]. [cit. 2019-04-18]. Dostupné z: <http://www.cplusplus.com/>
- [4] SQLGetData. Dokupedia Microsoftu [online]. Microsoft [cit. 2019-04-28]. Dostupné z: <https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/sqlgetdata-function?view=sql-server-2017>
- [5] SQLExecDirect. Dokupedia Microsoftu [online]. Microsoft [cit. 2019-04-29]. Dostupné z: <https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/sqlexecdirect-function?view=sql-server-2017>
- [6] SQLDriverConnect. Dokupedia Microsoftu [online]. Microsoft [cit. 2019-04-29]. Dostupné z: <https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/sqldriverconnect-function?view=sql-server-2017>
- [7] [online]. [cit. 2019-05-18]. Dostupné z: <https://www.w3schools.com/sql/default.asp>
- [8] BUREŠ, Petr. Datové zdroje ODBC - význam a použití. *NaPočítači.cz* [online]. Praha: Verlag Dashöfer, 2010 [cit. 2018-11-12]. Dostupné z: <https://www.napocitaci.cz/33/datove-zdroje-odbc-vyznam-a-pouziti-uniqueidmRRWSbk196EZY9EEhbhYd57NyUZJbB3cDa7vsDwSru0/>
- [9] *Microsoft* [online]. 1. Microsoft, 2017 [cit. 2018-11-10]. Dostupné z: <https://visualstudio.microsoft.com/cs/>
- [10] *Databázové systémy* [online]. 1. Ostrava: Microsoft, 2013, s. 4-8 [cit. 2018-11-12]. ISBN 978-80-248-3054-4. Dostupné z: http://projekty.fs.vsb.cz/463/edubase/VY_01_044/Datab%C3%A1zov%C3%A9%20syst%C3%A9my.pdf

Seznam příloh

Příloha 1 - SinFunkce.cpp je uložen na přiloženém CD

Příloha 2 – SinHeader.h je uložen na přiloženém CD

Příloha 3 - SinZaklad.sin je uložen na přiloženém CD

Příloha 4 - RauFunkce.cpp je uložen na přiloženém CD

Příloha 5 – RauHeader.h je uložen na přiloženém CD

Příloha 6 - RauZaklad.rau je uložen na přiloženém CD

Příloha 1 - ShockFunkce.cpp je uložen na přiloženém CD

Příloha 2 – ShockHeader.h je uložen na přiloženém CD

Příloha 3 - ShockZaklad.sin je uložen na přiloženém CD