ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA ELEKTROTECHNICKÁ

KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

BAKALÁŘSKÁ PRÁCE

Realizace ovládací softwaru pro signálové přepínače

Pavel Štěpnička **2015**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2014/2015**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Pavel ŠTĚPNIČKA**
Osobní číslo:	**E12B0319P**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Elektronika a telekomunikace**
Název tématu:	**Realizace ovládacího softwaru pro signálové přepínače**
Zadávající katedra:	**Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování:

1. Prostudujte způsob připojení elektronických signálových přepínačů k počítači PC.
2. Navrhněte a realizujte ovládací software pro řízení základních funkcí přepínačů z počítače PC.
3. Program optimalizujte pro operační systémy MS Windows.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: **Ing. Oldřich Tureček, Ph.D.**
Katedra technologií a měření

Datum zadání bakalářské práce: **15. října 2014**

Termín odevzdání bakalářské práce: **8. června 2015**

Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan

L.S.

Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 15. října 2014

Anotace

Tato bakalářská práce se zabývá ovládáním signálových přepínačů pro měření elektroakustických měničů, pomocí softwaru z osobního počítače.

V úvodních kapitolách se práce věnuje seznámením s Audioústřednou, jejím popisem, funkcí a využitelností.

Hlavní částí práce je popis samotného programu ovládajícího Audioústřednu, jeho předností, GUI a zároveň je vysvětleno v jakém vývojovém prostředí a jazyce je napsán.

Klíčová slova

program, Audioústředna, akustika, měření, software, přepínače

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.
Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....

podpis

V Plzni dne 5.6.2015 Pavel Štěpnička

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Oldřichovi Turečkovi, Ph.D. za připomínky, metodické vedení práce a také, že si na mě vždy udělal čas. Dále bych rád poděkoval i ostatním pedagogům a pracovníkům ZČU za mnoho zkušeností a vědomostí, které jsem během svého studia získal.

Obsah

Úvod

Předkládaná práce se zaměřuje na popis a realizaci ovládacího programu pro skupinu přepínačů, dále jen Audioústředna, která slouží pro měření elektroakustických měničů (reproduktorů).

Text je rozdělen do tří částí, kde první část se zabývá Audioústřednou, jejím popisem z funkčního hlediska a také s jakými prostředky Audioústřednou komunikovat. Ve druhé části práce se věnuji vývojovému prostředí Visual Studio, programovému jazyku C# a platformě .NET, které byly použity pro vývoj programu. V poslední části je popis samotného programu, jeho grafického rozhraní (GUI) a funkce předvoleb, ale hlavně popisem vývojového diagramu.

Seznam symbolů a zkratek

IDE ..... Vývojové prostředí (*Integrated Development Environmen)*

GC ..... Garbage collection je způsob spravování paměti

COM..... Rozhraní sériového portu

GUI..... Grafické uživatelské rozhraní (*Graphical User Interface)*

.NET ..... Soubor softwarových prostředků tvořící celou platformu

C#..... Objektově orientovaný programovací jazyk

USB ..... Univerzální sériová sběrnice

RS232..... Standard sériové linky

VS..... Visual studio

Audioústředna..... Soubor přepínatelných spínačů

CLI ..... Příkazový řádek (angl. Command Line Interface)

OS ..... Operační systém

1 Audioústředna

Audioústředna slouží pro náročnější akustická měření, kde je zapotřebí proměřit více kanálů najednou, to běžné měřicí přístroje většinou nemají k dispozici. Tato Audioústředna byla navržena pro měření v bezodrazové komoře. Jedná se o ovladatelnou skupinu spínačů pomocí rozhraní, které je popsáno v kapitole *1.2.*

1.1 Struktura Audioústředny

Struktura je tvořena třemi díly:

1.1.1 Centrální jednotka

Tato jednotka slouží k napájení a řízení celé Audioústředny. Obsahuje automaticky přepínatelné rozhraní RS232 a USB 1.1, pomoci kterými se jednotka připojuje k osobnímu počítači. Napájení zajišťuje transformátor, jehož napětí je usměrněno na 24 V.

1.1.2 Výkonový multiplexer

Výkonový multiplexer slouží k přepínání výkonových signálů do 250V / 10 A střídavých a 300V / 10 A stejnosměrných např. přepínání reproduktorů. Multiplexer obsahuje 2x4 kanály s vývody na zdířky. Dále obsahuje mikropočítač, který řídí jednotlivá relé a příjem příkazů z centrální jednotky.

1.1.3 Signálový miltiplexer

Signálový multiplexer je určený k přepínání slabých signálů do 60V. Multiplexer obsahuje 2x4 kanály s vývody na BNC konektory. Dále obsahuje mikropočítač, který řídí jednotlivá relé a příjem příkazů z centrální jednotky.

1.2 Komunikace s Audioústřednou

V této kapitole se práce zabývá komunikací, komunikačním rozhraním a jejich nastavením a také samotnými příkazy.

1.2.1 Popis dostupných komunikačních rozhraní

Audioústředna disponuje dvěma vstupními rozhraní, přičemž mezi nimi je automaticky řešené přepínání a to podle toho, které z nich je momentálně aktivní. Jedná se o rozhraní RS232 a USB 1.1, kde primárně *Centrální jednotka* přijímá řídicí signály z RS232 . Tomu tak je, dokud rozhraní USB 1.1 není připojeno k PC, jinak *Centrální jednotka* přijímá signály z USB. Na rozhraní USB je v *Centrální jednotce* převodník UART, takže oboje rozhraní pro PC se chovají jako COM port. Řídící signály se poté v Centrální jednotce převádějí do rozhraní RS485. Přes rozhraní RS485 následně *Centrální jednotka* komunikuje s ostatními periferiemi.

1.2.2 Použité komunikační rozhraní a jeho nastavení

Ať použijeme rozhraní USB 1.1 nebo RS232, stále se *Centrální jednotka* pro PC chová jako COM port. Proto je třeba předem nadefinovat komunikační parametry. Komunikační rychlost je nastavena na 115,2 kBd s délkou slova 8 bitů bez parity a jedním stop bitem. Signál DTR není třeba definovat, protože USB rozhraní tento signál standardně podporuje, ale i tak je v programu nastavený jako aktivní, aby bylo možné případně využít rozhraní RS232.

1.3 Komunikační příkazy Audioústředny

Příkazy vyslané z PC pomocí RS232 nebo USB 1.1, zpracuje mikropočítač v *Centrální jednotce*, dále je přeposílá do jednotlivých multiplexorů a následně je vyhodnocuje. Při správném provedení příkazu vrátí mikropočítač 1, když nastala chyba v příkazu (prvních 5 znaků se shoduje) vrátí 8, pokud se shodují pouze tři první znaky, tak vrátí 9. Pokud se vyslaný příkaz neshoduje ani ve třech prvních znacích, zpráva se ignoruje.

1.3.1 Výkonový multiplexor

PM1R	Test přítomnosti, pokud je připojen, vrací 0
PM1MAx	Vypne všechna relé multiplexeru A a po prodlevě 16 ms sepne relé x = 1-4
PM1MBx	Vypne všechna relé multiplexeru B a po prodlevě 16 ms sepne relé x = 1-4
PM1OAx	Vypne relé x = 1-4 multiplexeru A, pro x = A vypne všechna relé multiplexeru A
PM1OBx	Vypne relé x = 1-4 multiplexeru B, pro x = A vypne všechna relé multiplexeru B
PM1CAx	Sepne relé x = 1-4 multiplexeru A
PM1CBx	Sepne relé x = 1-4 multiplexeru B

Tab. 1 Seznam příkazů pro výkonový multiplexor

1.3.2 Signálový multiplexor

SM1R	Test přítomnosti, pokud je připojen, vrací 0
SM1MAx	Vypne všechna relé multiplexeru A a po prodlevě 16 ms sepne relé x = 1-4
SM1MBx	Vypne všechna relé multiplexeru B a po prodlevě 16 ms sepne relé x = 1-4
SM1OAx	Vypne relé x = 1-4 multiplexeru A, pro x = A vypne všechna relé multiplexeru A
SM1OBx	Vypne relé x = 1-4 multiplexeru B, pro x = A vypne všechna relé multiplexeru B
SM1CAx	Sepne relé x = 1-4 multiplexeru A
SM1CBx	Sepne relé x = 1-4 multiplexeru B

Tab.2 Seznam příkazů pro signálový multiplexor

2 Softwarové prostředky

2.1 Vývojové prostředí

Vývojové prostředí (IDE) je program umožňující vyvíjet aplikace většinou v jednom programovacím jazyce, jsou ovšem i IDE, ve kterých lze vyvíjet v několika programovacích jazycích. Jedná se hlavně o větší vývojové prostředí jako Visual Studio, Netbeans nebo Eclipse. Každé solidní vývojové prostředí obsahuje nástroje pro snadnější programování. O jaké nástroje IDE obsahuje, jsou popsány níže.

2.1.1 Grafický editor

Grafický editor slouží k vytvoření uživatelského rozhraní (GUI). Přes toto rozhraní uživatel ovládá, pomocí interaktivních prvků, vývoj celého programu. Ovládacími prvky jsou například tlačítka, formuláře, ikony, a tak dále.

2.1.2 Editor zdrojového kódu

Jedná se o textový editor pro psaní kódu v daném jazyce. Často barevně vyznačuje specifické části kódu, jako jsou proměnné, podmínky, funkce atd. Editory většinou podporují automatické formátování textu, pokud je syntakticky správně zapsán. Další funkcí editoru je automatické dokončování příkazů, nejčastěji editor nabídne nejpravděpodobnější možnost a programátor tak nemusí vypisovat celé příkazy, ale stačí jen jejich část. Barevné zvýraznění a přehledné formátování velice zprůhlední zdrojový kód a snadněji se v něm ostatní programátoři orientují.

2.1.3 Debugger

Debugger je software, který se používá pro hledání programátorských chyb v kódu programu. Při ladění jsou zobrazeny jednotlivé řádky kódu a je možné takzvaně krokovat. Díky tomu je ihned možné odhalit případnou chybu, které způsobuje různé nepřípustné stavy programu, jako je například dělení nulou.

Debugger obsahuje takzvané breakpointy, to jsou místa, kde se program během ladění zastaví, takže nemusíme krokovat každý řádek programu, ale pouze kritické části kódu. Dále během ladění jsou zobrazeny proměnné a jejich obsah.

2.1.4 Překladač

Překladač též nazýváno kompilátor, slouží pro překlad algoritmů zapsaných ve vyšších programovacích jazycích do nižších jazyků nebo do strojového kódu. Jedná se o nezbytný nástroj pro programování, který musí každé IDE obsahovat.

Většinou překladače jsou rozděleny na dvě části. První část překládá zdrojový kód do mezikódu, někdy nazývaného taky jako bytekód. Ve druhé části se tento mezikód překládá do strojového kódu cílené architektury.

Jednou ze zajímavých částí překladače, je část, která se zabývá optimalizací kódu. Jedná se o fázi překladu, kdy jsou vynechávány části kódu, ke kterým nikdy nemůže dojít, odstraňování bloků textu označené jako komentáře nebo dosazení vypočítaných konstant.

2.2 Jazyk C#

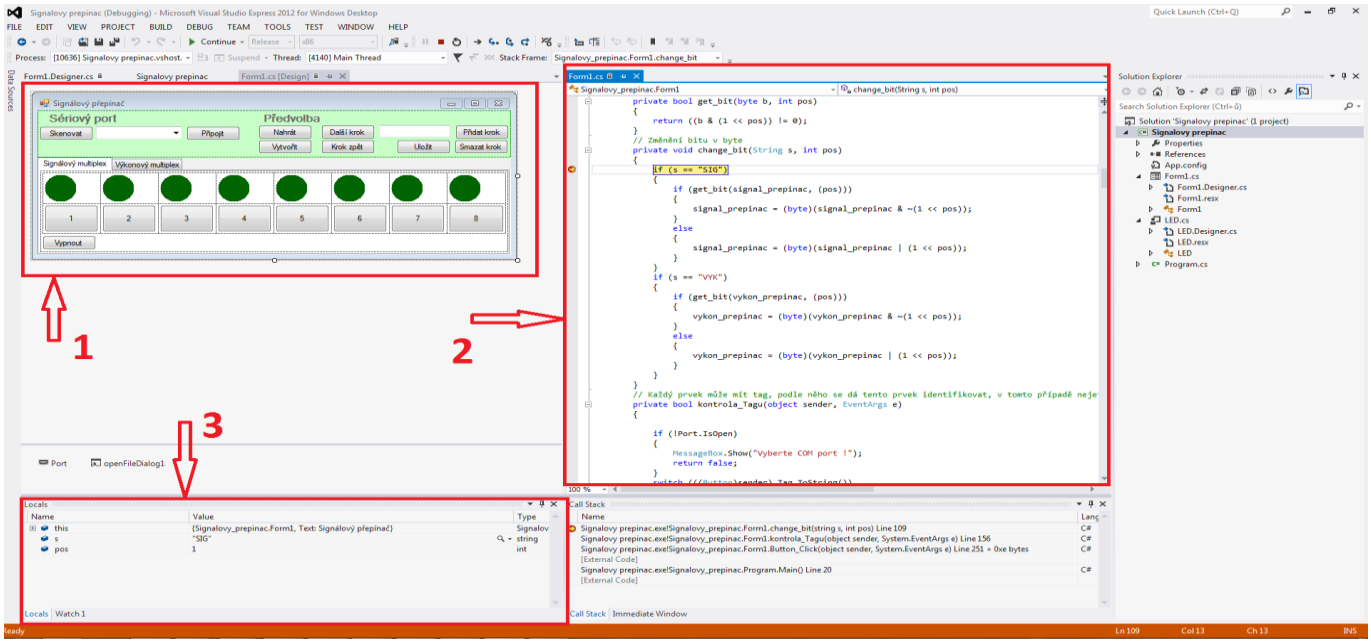
Jedná se o objektový programovací jazyk postavený na jazycích C++ a Javě, ale stále čerpá syntaxi z jazyka C. C# byl vyvinut firmou Microsoft a proto se hodí se zejména pro formulářové aplikace v operačních systémech Windows, to byl důvod, proč jsem si ho zvolil. C# lze využít i na jiné aplikace, například k databázovým programům, webové aplikace atd.

Dalším z důvodů, proč jsem si vybral právě C# je přítomnost takzvaného garbage collector (GC), což jednoduše řečeno je automatická správa paměti, kdy programátor se nemusí starat o alokování místa v RAM paměti a zároveň i jeho uvolňování. Veškerá správa paměti se tak děje automaticky, nicméně GC pro svoji činnost potřebuje procesorový čas, což by mohlo při některých aplikací vadit, ale naštěstí u tohoto programu nehrozí, protože není zde až tolik proměnných, které by GC musel kontrolovat.

2.3 Visual Studio

Pro naprogramování jsem si vybral Visual Studio (VS), protože práce v něm mi vyhovuje z několika důvodů. Hlavním z důvodů je přehledný grafický editor, který dovoluje mnoho možností jak program graficky přizpůsobit a dokonce si vytvořit vlastní komponenty nebo upravit ty již implementované.

Další výhodou VS je, automatická kontrola kódu a případná nabídka navrhovaných oprav. Tato funkce urychluje celkovou práci a spolu s režimem debuggeru to činí vhodné prostředí pro programování z důvodů efektivní tvorby kódu.



Obr. 1 Vývojové prostředí Visual Studio

Na *Obr. 1* jsou vyznačené části Visual Studia, kde se nachází jednotlivé nástroje popisované v kapitole *2.1 Vývojové prostředí.*

Část označena číslem 1 je *2.1.1 Grafický editor*. IDE se momentálně nachází v režimu debug, takže nejsou vidět všechny panely, které k němu patři. Nicméně je vidět, že úprava GUI výsledného programu je jednoduchá. Jednotlivé prvky lze přemisťovat pouhým přetažením myší, úprava velikosti prvku je taktéž intuitivní.

Část označena číslem 2 je *2.1.2 Editor zdrojového kódu.* Jsou vidět barevně odlišené texty komentářů, podmínek, konstant a zároveň je text zformátovaný. Dále je vidět část debuggeru, na které řádku je program pozastavený (žlutě vyznačeno).

Část označena číslem 3 se týká pouze debuggeru, jedná se o zobrazení aktuálních viditelných proměnných a jejich obsahu.

Celý vzhled VS lze samozřejmě přizpůsobit dle potřeb. Všechny okna lze přesouvat, zavírat či úplně vyjmout z hlavního okna a přesunout například na druhý pracovní monitor.

2.4 Platforma .NET

.NET je prostředí, určené převážně pro operační systémy Windows, které obsahuje rozsáhlé knihovny potřebné pro běh aplikací. Právě na systémy Windows je nejvíce rozšířena platforma Microsoft .NET Framework, která je v některých verzích Windows už implementována v základním balíčků, tudíž jí není třeba instalovat.

V tabulce *Tab. 3* je přehledáno znázorněna kompatibilita .NET Frameworku s operačními systémy Windows. Na to je důležité myslet při finální kompilaci programu, aby program byl spustitelný na, pokud co nejvíce OS, proto jsem zvolil verzi 4.0, která je funkční na Windows XP a i vyšší.

.NET verze	1.0	1.1	2.0	3.0	3.5	4.0	4.5
Windows 2000	lze doinstalovat	lze doinstalovat	lze doinstalovat	nelze	nelze	nelze	nelze
Windows XP	lze doinstalovat	lze doinstalovat	lze doinstalovat	lze doinstalovat	lze doinstalovat	lze doinstalovat	nelze
Windows Vista	částečná kompatibilita	částečná kompatibilita	součást systému	součást systému	lze doinstalovat	lze doinstalovat	lze doinstalovat
Windows 7	částečná kompatibilita	částečná kompatibilita	součást systému	součást systému	součást systému	lze doinstalovat	lze doinstalovat
Windows 8	nelze	lze doinstalovat	lze doinstalovat	lze doinstalovat	lze doinstalovat	kompatibilní	součást systému

Tab. 3 Kompatibilita Microsoft .NET Framework s operačními systémy Windows

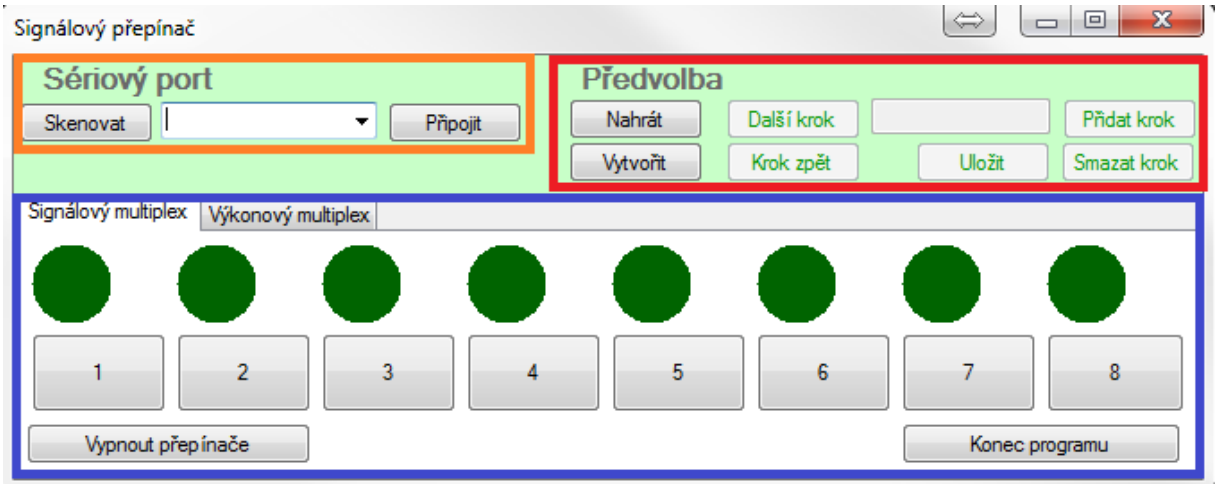
3 Realizace programu

3.1 Rozvržení GUI

GUI neboli grafické rozhraní je v dnešní době nedílnou součástí programů. Slouží k ovládání grafických interaktivních prvků, kterými je pak následně ovládán vývoj programu.

Předchůdcem GUI bylo ovládání na principu vypisování příkazů do přízové řádky (CLI). Takovou příkazovou řádku najdeme ještě dnes i v systémech Windows či jiných jednoduchých programech.

Na Obr.2 je zobrazeno GUI programu pro obsluhu Audioústředny s barevně vyznačenými oblastmi. V oranžové části jsou ovládací prvky pro připojení Audioústředny pomocí sériového portu, který je si možno vybrat z důvodu, že ovládací PC může mít k dispozici několik COM portů. V modře vyznačené části je část programu pro přímé ovládání Audioústředny a tlačítko pro vypnutí programu. Dominantními prvky v této části jsou tlačítka pro přímé ovládání relé a zelené indikátory stavu, zda je relé sepnuté či vypnuté. V horní části modře označené oblasti jsou vidět záložky, pro přepínání signálového nebo výkonového multiplexeru. Zároveň tato část slouží pro nadefinování nebo upravení předvoleb. Poslední část GUI vyznačená červeně se týká předvoleb. Více o předvolbách v následující kapitole *3.2 Předvolby.*



Obr. 2 GUI programu pro ovládání Audioústředny

3.2 Předvolby

Hlavní předností tohoto programu je možnost přednastavit si jednotlivé stavy multiplexorů do jednotlivých kroků, které jdou chronologicky po sobě. Tato možnost se zejména hodí tam, kde se provádí opakované měření. Tato možnost urychluje a usnadňuje měření prováděna s Audioústřednou.

Samotné předvolby se dají v průběhu měření kdykoliv upravit smazáním, přidáním, či upravením jednotlivých kroků, to navyšuje přizpůsobivost programu. Přizpůsobivost je velice důležitá z hlediska použitelnosti Audioústředny pro měřící účely.

Na *Obr. 2* je zvýrazněna část grafického rozhraní programu, kde se obsluhují předvolby. Z obrázku je vidět, že většina ovládacích prvků programu se věnuje právě k ovládání předvoleb a to k již zmíněné praktičnosti. Pojmenování jednotlivých tlačítek vystihuje danou funkci, kterou vykonává, proto vynechám jejich podrobný popis.

3.3 Popis vývojového diagramu

V *příloze B - první část vývojového* diagramu je popsán princip programu. První blok A značí začátek programu. V bloku B se inicializují všechny proměnné, které se v průběhu programu používají. Celý program je ovládán pomocí tlačítek, naslouchání a kontrolování jaké tlačítko bylo stisknuto je symbolizováno blokem C. Bloky D, F a H testují, jaké tlačítko bylo stisknuto.

Blok D se týká tlačítek pro změnu kroku v předvolbách. Blok F kontroluje tlačítka pro přímé ovládání jednotlivých relé v Audioústředně. Bloky E a G souvisejí s komunikací s Audioústřednou a kontrolují další nezbytné podmínky, pokud nejsou tyto podmínky splněny, blok J oznámí chybu. Bloky H testuje stisknuté tlačítko konec, pokud tato událost se vyhodnotí jako pozitivní, blok K ukončuje sériové spojení a pokud byla v předvolbách nějaká neuložená změna, zeptá se uživatele, zda tyto změny chce uložit. Následně se aplikace ukončí.

Dále jsou tu bloky L a M, které spojují diagramy, které se nachází v *příloze B* a C. Diagramy jsou rozděleny z důvodu přehlednosti a oddělení nezávislých funkcí programu.

V *příloze C* je pokračování předchozího diagramu. Je v něm vidět několik nezávislých funkcí. První funkce je reprezentována blokem U, jedná se o uložení předvolby jako textový soubor. V této funkce je ještě kontrola názvu předvolby (nesmí být prázdné políčko určené pro název), v případě nesplnění této podmínky program upozorní na chybu. Další funkcí je načtení předvolby, která je reprezentována blokem W. Tato funkce načte z textového souboru všechny kroky do vnitřní paměti programu. Nezbytnou funkcí pro celý program je připojení PC k Audioústředně přes virtuální COM port, která je reprezentována blokem X. Následuje poslední funkce pro správu předvoleb. Jedná se o tlačítka přidat nebo odebrat krok z načtené nebo vytvořené předvolby. Po provedených změnách v předvolbách se změny samy neukládají.

Všechny funkce zmíněné k *příloze C* se spouštějí stisknutím tlačítka a o jejich spuštění rozhodují bloky P, Q, R a S.

Závěr

Cílem této bakalářské práce bylo vytvořit program pro ovládání systém přepínačů pomocí osobního počítače, optimalizováno pro operační systémy Windows. Tyto podmínky byly splněny, program je odladěný a optimalizovaný pro operační systémy XP a vyšší.

V první části práce se věnuji samotným popisem systému přepínačů, též zvaný jako Audioústředna. Popisuji funkci těchto přepínačů, možností jejich využití, ale hlavně možnosti komunikaci s tímto systémem.

Ve druhé části práce se zaobírám použitými softwarovými prostředky pro tvorbu programu, především popisem vývojového prostředí Visual Studio od firmy Microsoft. Dále je v této části vysvětlen problém s platformou .NET v souvislosti s kompatibilitou programu na různých verzích operačních systémech Windows.

Třetí část práce se věnuje realizaci samotného programu. Převážně se jedná o popis vývojového diagramu a vysvětlení jeho funkce. Jedna kapitola se věnuje předvolbám, zdůvodnění proč vůbec tato funkce byla zakomponovaná do programu.

Pro tento program jsem navrhnul vývojový diagram, který se nachází v příloze B a C. Myslím si, že takto navržený vývojový diagram je správnou volbou z důvodu, že zbytečně nezatěžuje počítač na, kterým běží, protože hlavní vlákno čeká, dokud se nestiskne tlačítko. Tato metoda je mnohem efektivnější než neustálá kontrola změny stavu tlačítek.

Seznam literatury a informačních zdrojů

[1] VOBORNÍK, Aleš. Dokumentace k signálovým přepínačům : *Audioústředna*

[2] Wikipedie: Grafické uživatelské rozhraní (online). Dostupné z http://cs.wikipedia.org/wiki/Grafick%C3%A9_u%C5%BEivatelsk%C3%A9_rozhran%C3%AD

[3] Wikipedie: Příkazový řádek (online). Dostupné z http://cs.wikipedia.org/wiki/P%C5%99%C3%ADkazov%C3%BD_%C5%99%C3%A1dek

[4] Wikipedie: Vývojové prostředí (online). Dostupné z http://cs.wikipedia.org/wiki/V%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD

[5] SHARP, John. *Microsoft Visual C# 2010 krok za krokem* vyd. Brno: Computer Press, 2012

[6] Wikipedie: .NET (online). Dostupné z http://cs.wikipedia.org/wiki/.NET

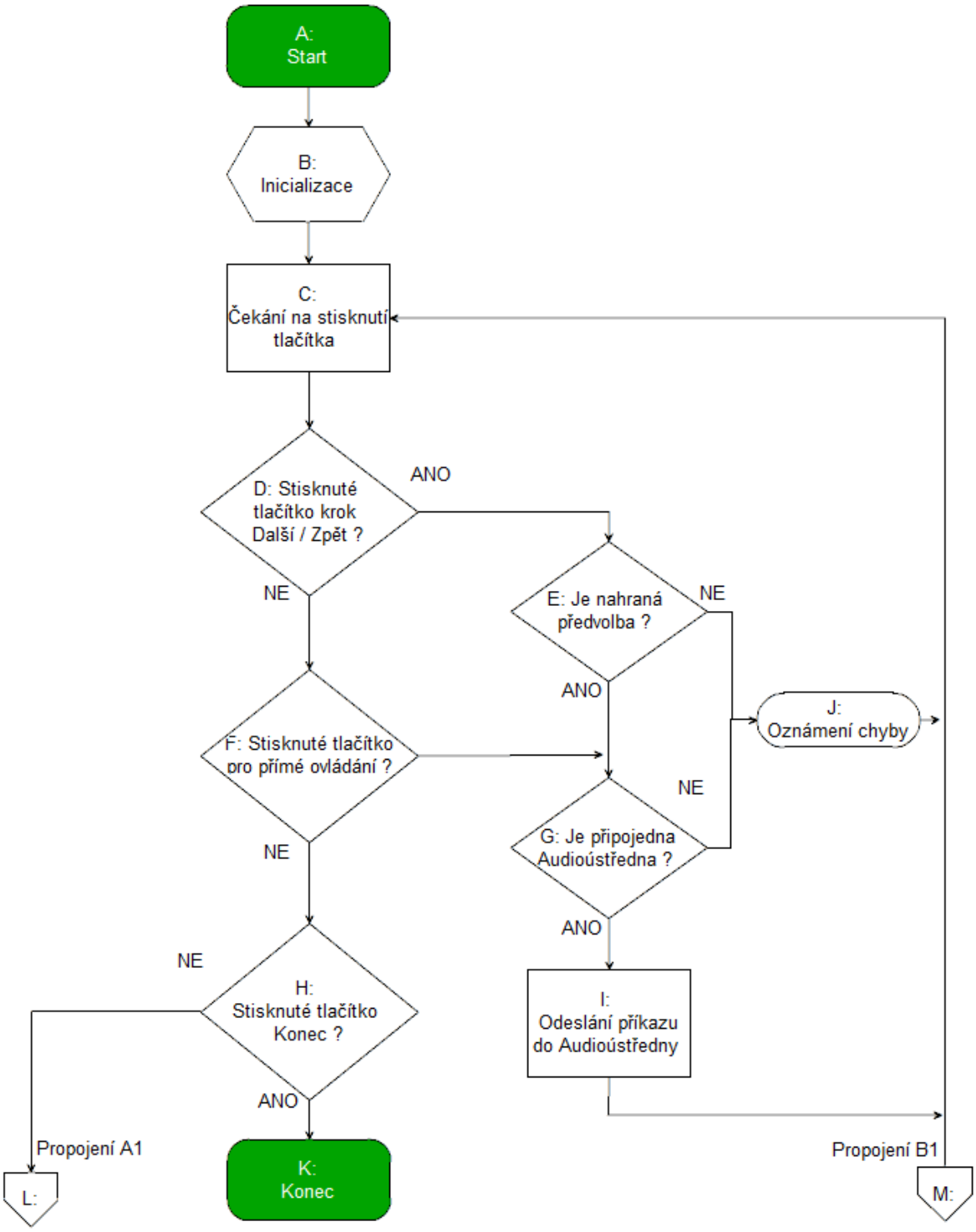
[7] Wikipedie: C Sharp (online). Dostupné z http://cs.wikipedia.org/wiki/C_Sharp

Přílohy

Příloha A – ukázka dvou funkcí ze zdrojového kódu, konkrétně funkce pro bitové operace

```
// Přečtení bitu z byte b na pozici pos (nultý bit je jedna)
private bool get_bit(byte b, int pos)
{
    return ((b & (1 << pos)) != 0);
}
// Změnění bitu v byte
private void change_bit(String s, int pos)
{
    if (s == "SIG")
    {
        if (get_bit(signal_prepinac, (pos)))
        {
            signal_prepinac = (byte)(signal_prepinac & ~(1 << pos));
        }
        else
        {
            signal_prepinac = (byte)(signal_prepinac | (1 << pos));
        }
    }
    if (s == "VYK")
    {
        if (get_bit(vykon_prepinac, (pos)))
        {
            vykon_prepinac = (byte)(vykon_prepinac & ~(1 << pos));
        }
        else
        {
            vykon_prepinac = (byte)(vykon_prepinac | (1 << pos));
        }
    }
}
```

Příloha B – první část vývojového diagramu



Příloha C – druhá část vývojového diagramu