



Středoškolská technika 2016

Setkání a prezentace prací středoškolských studentů na ČVUT

JARVIS – Virtuální domácí asistent

Boris Karavasilev

Střední průmyslová škola Brno, příspěvková organizace

Purkyňova 97, Brno

Prohlášení

Prohlašuji, že svou práci na téma Jarvis – virtuální domácí asistent jsem vypracoval samostatně pod vedením Bc. Lucie Svobodů a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

V Brně dne 15.2.2016

Podpis



STŘEDNÍ PRŮMYSLOVÁ ŠKOLA BRNO,
Purkyňova, příspěvková organizace



Jihomoravský kraj



Poděkování

Děkuji svému učiteli Ing. Jaroslavu Nesvadbovi, CSc. a Bc. Lucii Svobodů za obětavou pomoc, podnětné připomínky a nekonečnou trpělivost, kterou mi během práce poskytovali.

Anotace

Cílem této práce je návrh a vývoj virtuálního asistenta poskytujícího hlasové ovládání počítače, který má možnost propojení s HW periferiemi a rozšíření jeho funkcí. Díky automatickému rozpoznávání hlasových povelů je možné velmi intuitivně komunikovat s počítačem a naučit jej řadu hlasových povelů, které zjednoduší a urychlí používání počítače či ostatních elektrických spotřebičů v běžné místnosti.

Virtuální asistent je užitečný pro usnadnění každodenního života, částečnou domácí automatizaci nebo i usnadnění obtížných úkonů zdravotně postiženým. Asistent je schopný pomocí hlasových povelů spouštět libovolné programy, otevírat nastavené webové stránky, hlasově vyhledávat na internetu nebo dokonce i prostřednictvím periférií ovládat připojené spotřebiče. Při kombinaci běžného používání počítače s hlasovými příkazy vzniká velice zábavný a efektivní způsob práce.

Klíčová slova

virtuální asistent; HW; periferie; hlasové příkazy; domácí automatizace

Annotation

The goal of this work is a plan and development of a virtual assistant capable of computer voice control, which is able to connect to HW peripherals and extend his functions. Thanks to automated voice recognition it is possible to communicate very intuitively with a computer and teach it loads of voice commands, which will make easier and faster the use of your computer or other connected devices in an ordinary room.

Virtual assistant is useful for making your everyday life easier, partial home automation or even make difficult every day activities easier for disabled people. The assistant is able thanks to voice commands to run any programs, open set webpages, voice search the internet or even thanks to HW extensions control connected devices. When you combine regular use of your PC with voice commands, arises a very fun and effective way to use your computer.

Keywords

virtual assistant; HW; peripherals; voice commands; home automation

Obsah

ÚVOD.....	9
1 V ČEM JSEM JARVISE VYVÍJEL?.....	10
1.1 POČÍTAČOVÝ PROGRAM.....	10
1.2 USB MODUL.....	10
2 NÁVRH LOGICKÉ STRUKTURY PROJEKTU.....	11
2.1 HLAVNÍ ROZDĚLENÍ.....	11
2.2 ROZDĚLENÍ PODLE FUNKCE.....	11
3 OKNA.....	12
3.1 MINIMALISTICKÝ INDIKÁTOR INTERAKCÍ.....	12
3.1.1 <i>Design</i>	13
3.1.2 <i>Zdrojový kód okna</i>	13
3.2 OKNO PRO PŘIZPŮSOBENÍ PŘÍKAZŮ.....	14
3.2.1 <i>Design</i>	14
3.2.2 <i>Zdrojový kód okna</i>	14
3.3 FORMULÁŘ ZÁKLADNÍCH INFORMACÍ.....	16
3.3.1 <i>Design</i>	16
3.3.2 <i>Zdrojový kód okna</i>	17
3.4 OKNO OBECNÉHO NASTAVENÍ.....	17
3.4.1 <i>Design</i>	17
3.4.2 <i>Zdrojový kód okna</i>	17
3.5 OKNO VIRTUÁLNÍHO DÁLKOVÉHO OVLADAČE.....	18
3.5.1 <i>Design</i>	18
3.5.2 <i>Zdrojový kód okna</i>	18
.....	18
4 TŘÍDY.....	19
4.1 TŘÍDA COMMANDPROCESSOR.....	19
4.1.1 <i>Metoda findAnswer()</i>	19
4.1.2 <i>Metoda executeCommand()</i>	19
4.2 TŘÍDA USERINFO.....	20
4.3 TŘÍDA USBCONTROLLER.....	21
5 PŘÍDAVNÝ USB MODUL.....	22
5.1 HARDWARE.....	22
5.2 SOFTWARE.....	23
6 POTENCIÁL BUDOUCÍHO ROZVOJE.....	25
6.1 ZDRAVOTNICTVÍ.....	25
6.2 VÝVOJ ELEKTRONIKY.....	26
6.3 SHROMAŽĎOVÁNÍ INFORMACÍ.....	26
ZÁVĚR.....	27
LITERATURA.....	28
SEZNAM OBRÁZKŮ.....	29
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK.....	30
SEZNAM PŘÍLOH.....	31
PŘÍLOHY.....	32
PŘÍLOHA Č. 2.....	32

Úvod

Jarvis vznikl původně jen jako sen, který jsem měl jako malý, kdy jsem se nechal inspirovat filmem Iron Man. V tomhle filmu si hlavní hrdina milionář Tony Stark vytvořil robotický oblek, díky kterému se mu podařilo uprchnout ze zajetí teroristů. On však po té svůj oblek dále vylepšoval na novější a propracovanější verze. Hlavní věc, která mne zaujala, byl Jarvis. Byl to program, který mu umožňoval hlasově ovládat nejen oblek ale i velkou část jeho domu. Byl skoro jako jiná lidská bytost schovaná ve vašem počítači schopná si s vámi o čemkoliv a kdykoliv popovídat. Od malička jsem snil o takovém vlastním Jarvisovi, ale teď jsem se konečně dočkal, protože jsem si ho vytvořil! Můj program sice ještě není tak propracovaný jako ten ve filmu ale věřím, že jednou bude.

Na začátku je popsána struktura celé práce. Nalezneme tu popis vytvoření myšlenkové mapy jakožto přehledný strukturovací nástroj pro nastínění propojení jednotlivých částí a modulů. Z této mapy vyplívá návaznost a souvislost jednotlivých částí mezi sebou. Myšlenková mapa v jejím plném rozsahu je přiložena v přílohách. Struktura práce je tvořena výsečemi z této mapy a jejich detailnějším popisem.

Celá práce je rozdělena na dvě hlavní části. První část se zabývá tvorbou programu hlasového asistenta určeného pro počítače s operačním systémem Windows, protože hlavní nástroj umožňující hlasové rozpoznávání je součástí tohoto operačního systému. Druhá část je zaměřena na proces vývoje modulu od fáze před-prototypového řešení až po hotový prototyp představující malou krabičku, kterou si USB kabelem připojíte k počítači.

Tuto práci jsem začal vypracovávat, protože jsem zatím nenarazil na takového hlasového asistenta, jakého bych si představoval. Není mi známo, že by byl zatím vytvořený takový hlasový asistent, který by poskytoval možnost jednoduchého přizpůsobení jednotlivých příkazů nebo přidání svých vlastních s možností přiřazení vykonání libovolných funkcí které nejsou omezeny jen na softwarovou úroveň, ale umožňují i ovládat cokoli co jde připojit do zásuvky. A co víc, k takovéto personalizaci asistenta není potřebná žádná znalost programování a zvládne ji opravdu každý člověk se znalostí angličtiny a se základními zkušenostmi s prací na počítači.

1 V čem jsem Jarvise vyvíjel?

Na začátek bych chtěl zmínit, že veškeré zdrojové kódy jsou k nalezení na přiloženém CD v přílohách práce. Nemá velký význam tu rozpitvávat konkrétní kusy kódu a rozebírat jak přesně jsem programy psal protože, pokud člověk není opravdu dobře seznámený s programováním a nemá s tím zkušenost tak by z této práce moc neměl. Já bych si přál, aby tato práce byla pochopitelná pro co možná největší množství lidí. Aby i lidé s technikou skoro neseznámeni díky ní zjistili, jak Jarvis funguje a jak bych ho chtěl v budoucnu rozvíjet. Proto jsem se v této práci snažil kód popisovat spíše z funkčního a logického hlediska. Ti technicky zdatnější, kteří by se zajímali o detaily, je v přílohách naleznou a mohou mne kdykoliv kontaktovat a probrat hlubší aspekty této práce.

1.1 Počítačový program

Program hlasového asistenta jsem programoval v jazyku C# ve vývojovém prostředí Visual Studio 2015 jako aplikaci typu WPF (Windows Presentation Foundation). U tohoto typu aplikace se design píše ve značkovacím jazyku XML, který je velice podobný svou syntaxí jazyku pro tvorbu webových stránek HTML. Co se týče funkčnosti tak C# umožňuje využívat Windowsovský modul pro rozpoznávání i generování řeči. Tohle je hlavní důvod proč jsem si C# vybral, ale také protože je to obecně velice silný a rozmanitý nástroj. Jsem na tento jazyk zvyklý, jelikož jsem v něm už vyvíjel řadu jiných aplikací.

1.2 USB modul

Řídící část modulu tvoří mikropočítač Arduino. Ten se programuje v jazyku C++ přizpůsobeném konkrétně pro Arduino. Pracoval jsem ve vývojovém prostředí, které je zdarma poskytované ke stažení přímo od výrobců na oficiálních stránkách Arduina. Více podrobnějších informací naleznete v kapitole věnované výhradně přídatnému USB modulu.

2 Návrh logické struktury projektu



Obrázek 1 - logo programu XMind[1]

Na úplném začátku vývoje Jarvise jsem se rozhodl, že bude nejlepší si tento projekt dobře promyslet a naplánovat. Pro tyto účely jsem si vybral nástroj XMind umožňující tvorbu myšlenkových map. Myšlenkovou mapu jsem si vybral pro zachycení struktury mého projektu, protože je přehledná a na první pohled je z ní zřejmé, co s čím souvisí.

2.1 Hlavní rozdělení

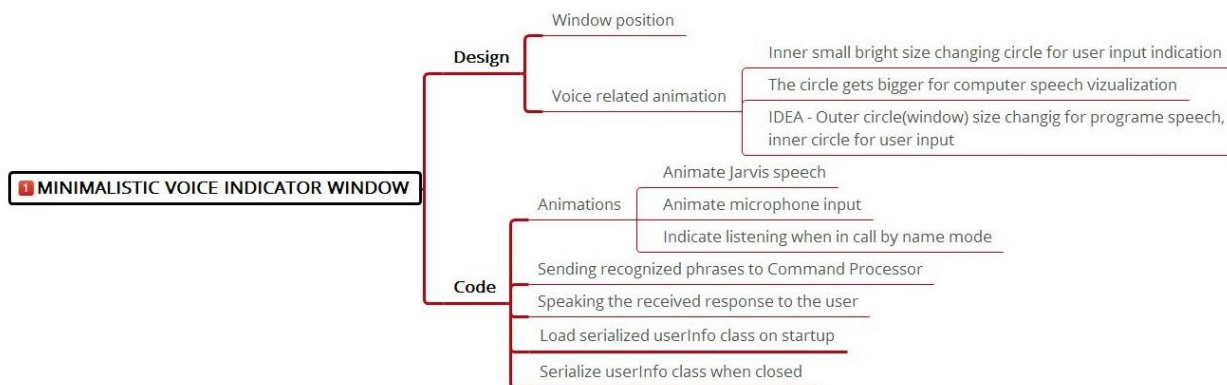
Můj projekt se dělí na dvě hlavní části. První a významnější část je počítačový program komunikující s uživatelem vykonávající jeho příkazy. Pak na část vedlejší a to je přídatný USB modul umožňující rozšíření funkcí Jarvise na bezdrátové ovládání spotřebičů. Jarvis běží samostatně a nezávisle na tomto modulu ve vašem PC. Přidáním tohoto modulu pouze obohatíte Jarvisovi možnosti. Přídatný USB modul s počítačem komunikuje pomocí sériové komunikace. Modul od počítače přijímá příkazy a informuje ho o detekci pohybu.

2.2 Rozdělení podle funkce

Tyto dvě skupiny jsou dále rozděleny podle jejich funkcí. Hlavní rozdělení programu je na dialogová okna a třídy. U jednotlivých oken je popsáno, co by měla obsahovat za komponenty a jaké by měli vykonávat funkce. U tříd jsem sepsal jaké metody a proměnné by měly obsahovat. Jednotlivé funkční bloky jsou ve struktuře seřazeny sestupně podle důležitosti a je u nich číslo priority, se kterou je třeba je realizovat. Začínal jsem vyvíjet od položek s nejvyšší prioritou, neboť ty jsou esenciální součástí celého systému a postupně jsem se propracovával k realizaci dalších částí programu, které jej obohatili o další velmi užitečné funkce. Díky pečlivému rozvržení práce se mi podařilo dospět k uspokojivým výsledkům a dokázat vyvinout tento projekt do úplně jiného rozměru než kdybych tuto přípravu zanedbal.

3 Okna

3.1 Minimalistický indikátor interakcí



Obrázek 2 - část myšlenkové mapy pro indikátor interakcí

Pro vytvoření lepšího pouta mezi uživatelem a Jarvisem jsem vytvořil hlasový indikátor vašeho i Jarvisova hlasu. Tento vizualizační prvek je v podobě malého huňatého kolečka vyskytujícího se v pravém dolním či horním rohu obrazovky. Samotnými indikátory jsou dvě vnitřní pulzující kolečka, každé pro jeden hlas a ty se navzájem prolínají. Takováto decentní, ale výstižná vizualizace přispívá k vyvolání pocitu jistoty a porozumnění u uživatele.

V případě, kdy by tento vizualizační prvek uživateli jakýmkoliv způsobem vadil nebo uživateli nevyhovoval, tak Jarvisovi nečiní sebemenší problém se na povel zneviditelnit nebo se jen přesunout do jiné části obrazovky. Uživatel může svou nespokojenost dát najevo prostým vyslovením intuitivního příkazu „go up“, „go down“, „hide“ nebo „show“.

Veškerá komunikace s Jarvisem probíhá v angličtině, ale v budoucnu mám v plánu Jarvis rozšířit o další jazyky. Prozatímní pouto k anglickému jazyku plyne z Jarvisovi podstaty. Jarvis je založený na modulech pracujících s řečí zabudovaných v OS Windows. V budoucnu chci Jarvis kompletně přepracovat a bude naprosto nezávislý na uživatelově PC a bude dodáván s miniaturním počítačem, na kterém bude schopný běžet 24h denně.

3.1.1 Design

O designu jsem se už trochu zmínil, ale hlavní myšlenkou byla miniaturní vizualizace, která by uživateli nijak nepřekážela v práci a zároveň v něm vyvolávala pocit, že mu Jarvis neustále naslouchá.



Obrázek 3 – indikátor[2]

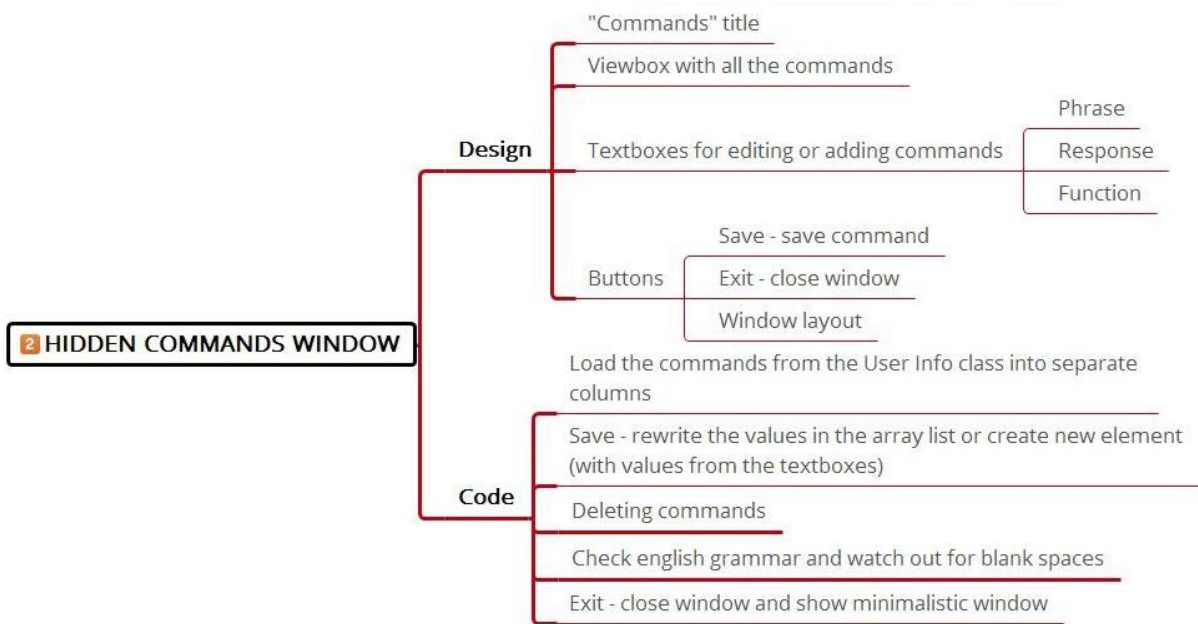
3.1.2 Zdrojový kód okna

Zdrojový kód obsluhující funkčnost, který se skrývá za tímto velice nestandardním „oknem“ aplikace se stará o několik hlavních věcí. Při spuštění programu se prověří, zda byla už někdy spuštěná a jestli si ho uživatel přizpůsobil. Pokud je vyhodnoceno, že se nejedná o první spuštění programu, tak se načte serializovaná třída UserInfo a inicializuje se aplikace s uživatelskými údaji a přizpůsobenou sadou příkazů. Naopak při vypnutí programu, se třída UserInfo obsahující veškerá data která je potřeba si pamatovat, uloží serializací do binárního souboru.

V tomto okně také běží nepřetržitě rozpoznávání řeči. V momentě kdy program rozpozná jakoukoliv známou fázi či větu, tak si okamžitě zavolá příslušnou metodu ze třídy CommandProcessor pro její vyhodnocení. Tam se provede zpracování příkazu a hlavnímu oknu se už jen vrátí odpověď pro uživatele.[3]

Poslední z významných funkcí je také obstarávání všech animací okna, což zahrnuje pulsuující kroužek znázorňující vstup hlasu z mikrofonu, kroužek měnící velikost pro indikaci Jarvisovi zpětné vazby a také animace přesunů po obrazovce.[4]

3.2 Okno pro přizpůsobení příkazů



Obrázek 4 - část myšlenkové mapy okna pro přizpůsobení příkazů

Toto okno se zobrazí na displej po vyslovení příkazu „commands“. Je to jedno z nejdůležitějších oken této aplikace, protože umožňuje uživateli upravovat známé příkazy nebo vytvářet nové. Aplikace je navržena tak aby si ji mohl uživatel do velké míry modifikovat a přizpůsobit k obrazu svému bez jakékoliv znalosti programování. Možnost jednoduše si modifikovat tento systém vidím jako obrovskou výhodu na poli široké veřejnosti, protože to znamená, že by si tuto aplikaci mohl nastavit snad opravdu každý. Od mladého teenagera nepřilíš zasněveného do sféry IT, člověka středního věku až po babičku které by s konfigurací systému pomohl vnuček.

3.2.1 Design

Design je navrhnutý tak aby byl intuitivní a jednoduchý. Ovládání spočívá v jednoduchém označování příkazů v tabulce. Obsah označeného políčka se načte do textových polí, kde pak uživatel může změnit znění příkazu, odpověď na něj nebo jméno funkce která se má po jeho vyslovení vykonat. Pro přidání nového příkazu uživatel klikne do posledního prázdného políčka a vyplní jeho obsah. Po stisknutí tlačítka „Save“ se vytvořený příkaz přidá do tabulky příkazů a uloží se do paměti.[5]

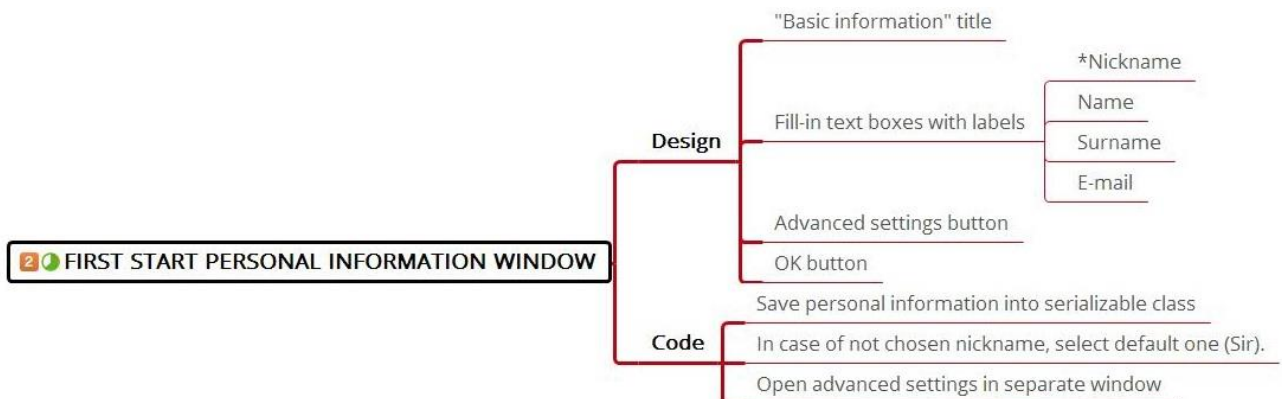
3.2.2 Zdrojový kód okna

Po spuštění dialogového okna se načtou všechny příkazy z třídy UserInfo. Při načítání jsou tři části příkazu jednotlivě přidán do oddělených sloupečků v tabulce. První část příkazu je fráze, na kterou Jarvis uslyší. Druhá část příkazu je odpověď, kterou Jarvis vysloví a poslední část je volitelná funkce k vykonání. Funkce mají běžné anglické názvy.[6]



Obrázek 5 - vzhled okna pro přizpůsobení příkazů

3.3 Formulář základních informací



Obrázek 6 - část myšlenkové mapy pro formulář základních informací

Tento formulář se zobrazí při prvním spuštění aplikace. Zde si uživatel může nastavit jak ho má Jarvis oslovovat a také zde zadá svůj e-mail, jméno, příjmení a přezdívkou. Má na výběr ze tří možností oslovení. První možnost je že ho Jarvis bude oslovovat „Sir“ tedy česky pane. Druhá možnost je „Madam“ tedy paní a jako poslední možnost vás bude oslovovat přezdívkou, jakou si sami zadáte.

Všechny tyto informace jsou nakonec uloženy do třídy UserInfo a tudíž nebudou zapomenuty po ukončení programu. Pokud by uživatel chtěl změnit některé z těchto informací tak stačí vyslovit příkaz „Personal information“ a tento formulář se mu znovu objeví na obrazovce.

Snažil jsem se program maximálně personalizovat. Tohle do velké míry přispívá k boření bariéry mezi člověkem a počítačem. Díky skutečnosti, že se oba oslovují jménem, program působí mnohem příjemněji a uživatel časem začne pomocníka vnímat jako opravdového kamaráda v PC.

Není povinné vyplňovat žádnou z těchto informací. V případě že nespecifikujete, ani to jak by vás měl Jarvis oslovovat, tak se oslovení nastaví na defaultní hodnotu „Sir“.

3.3.1 Design

Tohle okno je navrženo jako běžný vyplňovací formulář. První pole slouží k zadání přezdívkou, při pozdějším výběru oslovení přezdívkou se čerpá hodnota odsud. Další pole pro vyplnění jsou jméno, příjmení a email. Tyto hodnoty se prozatím k ničemu nevyužívají, ale v pozdějších rozšířeních programu pomohou k personalizaci programu.

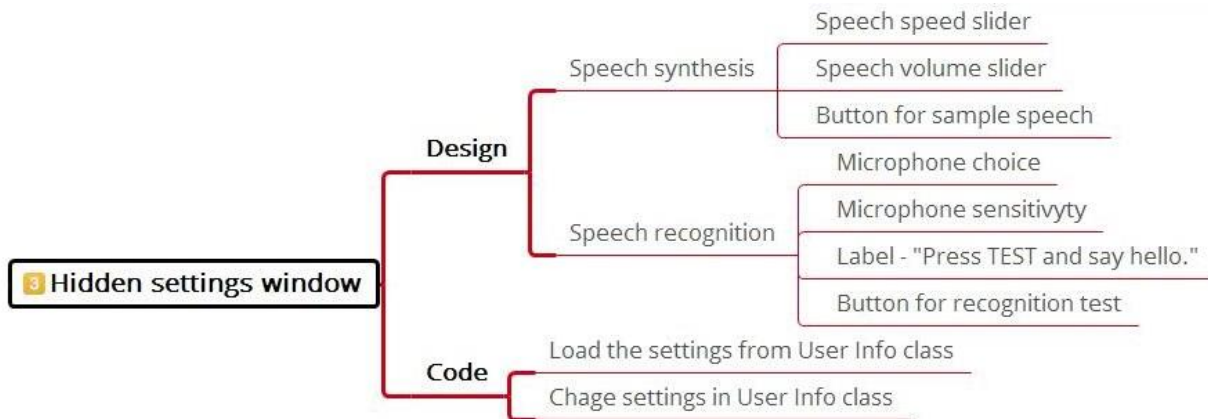
The screenshot shows a yellow dialog box titled "Personal information". It contains four text input fields labeled "Nickname:", "Name:", "Surname:", and "E-mail:". Below these fields is a section titled "How should I call you ?" with three radio button options: "Sir", "Madam", and "Nickname". At the bottom of the dialog is an "OK" button.

Obrázek 7 - vzhled okna formuláře základních informací

3.3.2 Zdrojový kód okna

Zdrojový kód triviální a pouze vyplněné hodnoty uloží do třídy UserInfo. Veškeré zdrojové kódy naleznete v přílohách.

3.4 Okno obecného nastavení



Obrázek 8 - část myšlenkové mapy pro okno obecného nastavení

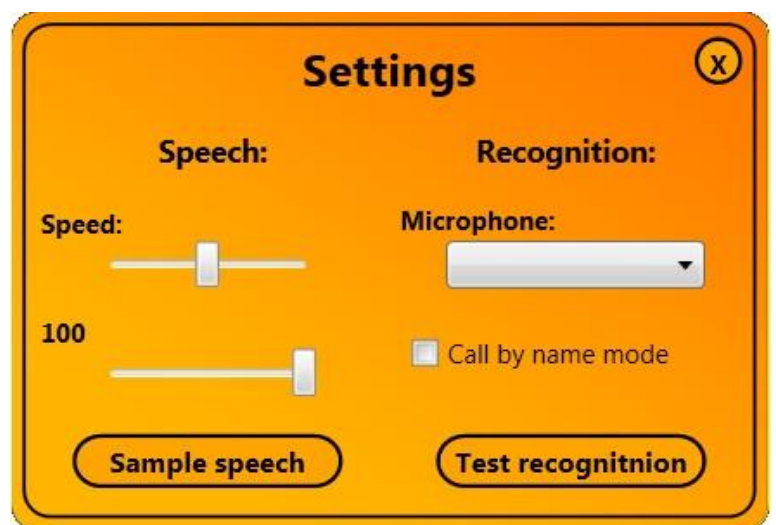
Zde se dají nastavovat parametry generování a rozpoznávání řeči. Tohle okno dává uživateli další možnost jak si Jarvis přizpůsobit. Kladu velký důraz na vytvoření pouta mezi programem a uživatelem, abych toho mohl dosáhnout tak je třeba, aby byl program opravdu vysoce přizpůsobitelný.

3.4.1 Design

Design je tu navržený tak aby vizuálně odděloval nastavení pro generování a rozpoznávání řeči. Co se týče generování řeči neboli Jarvisova hlasu, tak jsou pro něho vyhrazeny dva posuvníky. První pro nastavení rychlosti a druhý pro nastavení hlasitosti řeči. V rámci nastavení rozpoznávání řeči je tu kombinovaný seznam pro výběr mikrofону a přepínač pro zapnutí funkce „Call by name“. Tato funkce přepne Jarvis do módu, ve kterém ignoruje veškeré příkazy, dokud ho nevoláte jménem. Pro obě části je implementováno testovací tlačítko, díky kterému je možné vyzkoušet, jak program reaguje na provedené změny.

3.4.2 Zdrojový kód okna

Po spuštění okna se načtou dosavadně nakonfigurovaná nastavení, která jsou jako všechny důležité informace uložena ve třídě UserInfo. Při vypínání tohoto okna se veškeré provedené změny uloží do třídy UserInfo, která je nadále uchovává do další změny.



Obrázek 9 - vzhled okna obecného nastavení

3.5 Okno virtuálního dálkového ovladače



Obrázek 10 - část myšlenkové mapy pro okno virtuálního ovladače

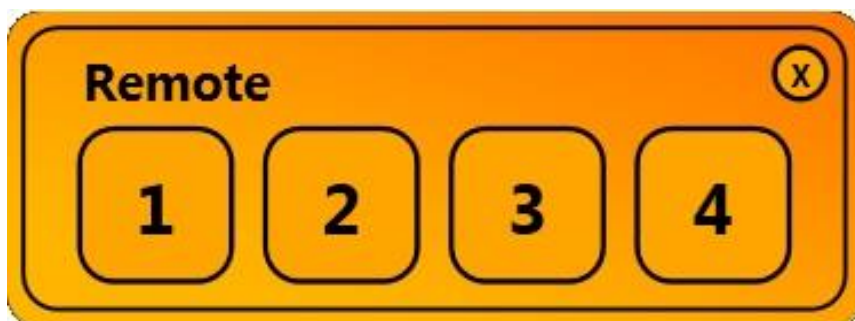
Pokud by uživatel preferoval manuální ovládání spotřebičů, ať už z důvodů že zrovna nechce nebo nemůže používat hlasové ovládání, tak je to díky tomuto oknu možné pomocí klikání na tlačítka z pohodlí vašeho počítače můžete ovládat všechny připojené periferie. Už nemusíte vstávat ze židle aby, jste vypnuly světlo. Už vám jen stačí stisknout tlačítko na obrazovce

3.5.1 Design

Design tohoto okna je velice jednoduchý. Okno obsahuje pouze čtyři on/off tlačítka.

3.5.2 Zdrojový kód okna

Po zmáčknutí jakéhokoliv z tlačítek se odešle signál přes třídu USBcontroller. USB modul, který tento signál obdrží, už obstará zapnutí či vypnutí požadovaného spotřebiče. Více podrobnějších informací o komunikaci s USB modulem naleznete v sekci „Třídy“ pod názvem „Třída USBcontroller“.



Obrázek 11 - vzhled okna dálkového ovladače

4 Třídy

4.1 Třída CommandProcessor



Obrázek 12 - část myšlenkové mapy pro třídu CommandProcessor

Tato třída slouží výhradně ke zpracování rozpoznávaného příkazu. Obsahuje metody pro vyhodnocení správné odpovědi a případné vykonání přiřazené funkce.

Samotné rozpoznávání i generování řeči je ohromně složitý proces, který bych nebyl schopen sám vymyslet, ani vytvořit. Je stále vyvíjen a zdokonalován různými týmy vědců na planetě. Proto jsem si pro tento úkol vybral hotové řešení. Nejvhodnější z dostupných možností se mi zdálo rozpoznávání hlasu zakomponované v OS Windows, kterého jde využívat v programovacím jazyku C#. Tohle bylo hlavní kritérium, proč jsem si vůbec tento jazyk zvolil.

4.1.1 Metoda findAnswer()

Tato metoda je volána při rozpoznání řeči. Porovná rozpoznávaný výraz vůči seznamu známých frází a vrátí na něho adekvátní odpověď. Pokud je u této fráze uloženo i jméno určité funkce tak se předá metodě executeCommand() a ta už si s vykonáním této funkce poradí.

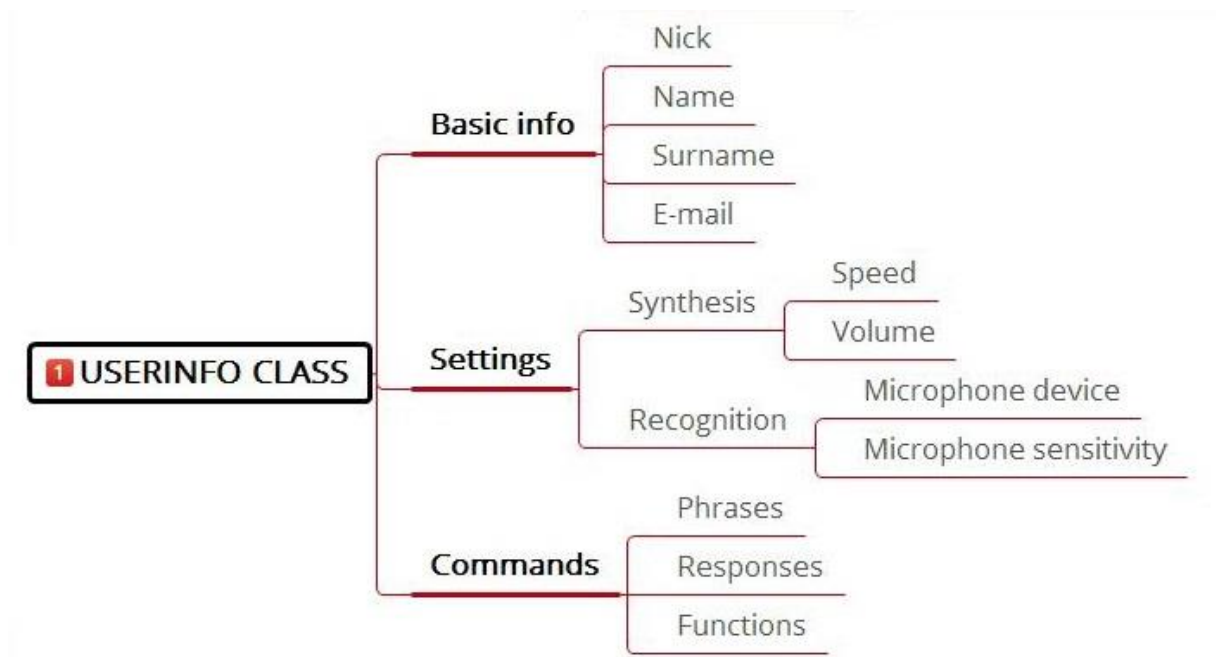
4.1.2 Metoda executeCommand()

Když je tato metoda zavolána s označením požadované funkce, tak je tohle označení dále rozčleněno. Pokud je označení více než jednoslovné tak si ho po slovech rozdělí. První slovo je název příkazu a zbytek jsou atributy tohoto příkazu. Například pokud tuto metodu zavoláme s příkazem „open calculator“ tak ho zpracuje následovně



Obrázek 13 - schéma formátu psaní funkcí

4.2 Třída UserInfo



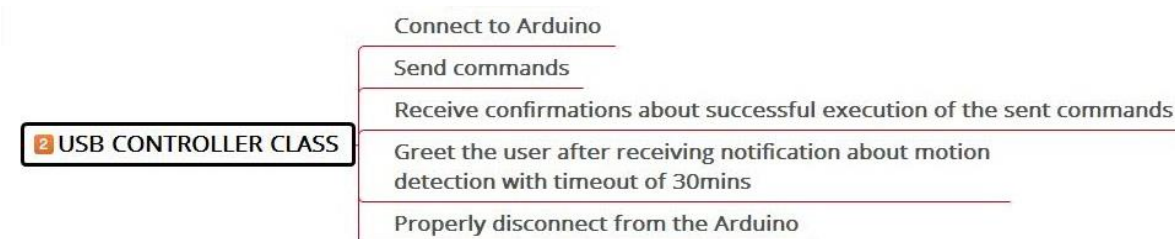
Obrázek 14 - část myšlenkové mapy pro třídu UserInfo

Tato třída obsahuje veškeré informace o uživateli a také jsou v ní uloženy všechny příkazy s příslušnou odpovědí na ně a také případným jménem funkce, kterou má Jarvis při daném příkazu vykonat.

Tato třída je serializovatelná což znamená, že je možné ji uložit například v binární podobě. Serializace představuje obrovskou výhodu v tom, že po ukončení programu si můžeme aktuální informace o uživateli a případně nově naučené příkazy uložit a program si je při dalším spuštění bude znovu pamatovat.

Tato třída hraje důležitou roli v chodu celé aplikace, bez ní by si uživatel sice mohl příkazy upravovat a přidávat, ale po vypnutí programu by veškeré změny byly nenávratně ztraceny a musel by je po restartu programu znovu provést. Kvůli tomu si dovoluji tvrdit, že bez této třídy by byla veškerá přizpůsobitelnost programu skoro zbytečná a jenom by uživatele obtěžovala muset po každém spuštění celý systém překonfigurovat.[7]

4.3 Třída USBcontroller

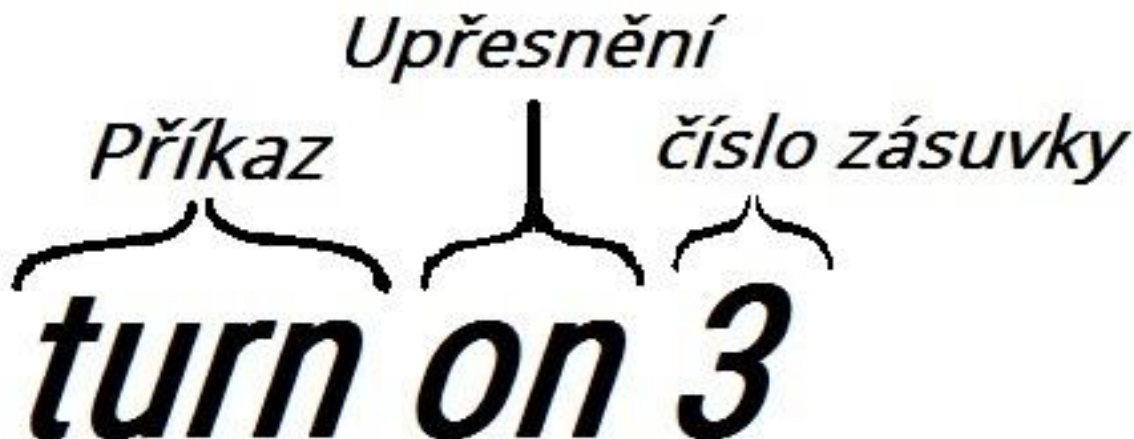


Obrázek 15 - část myšlenkové mapy pro třídu USBcontroller

Pomocí této třídy probíhá veškerá komunikace s USB modulem. Na začátku při inicializaci aplikace proběhne připojení k modulu s přenosovou rychlostí 9600 bit/s. Tato rychlost je nastavena i na samotném modulu.

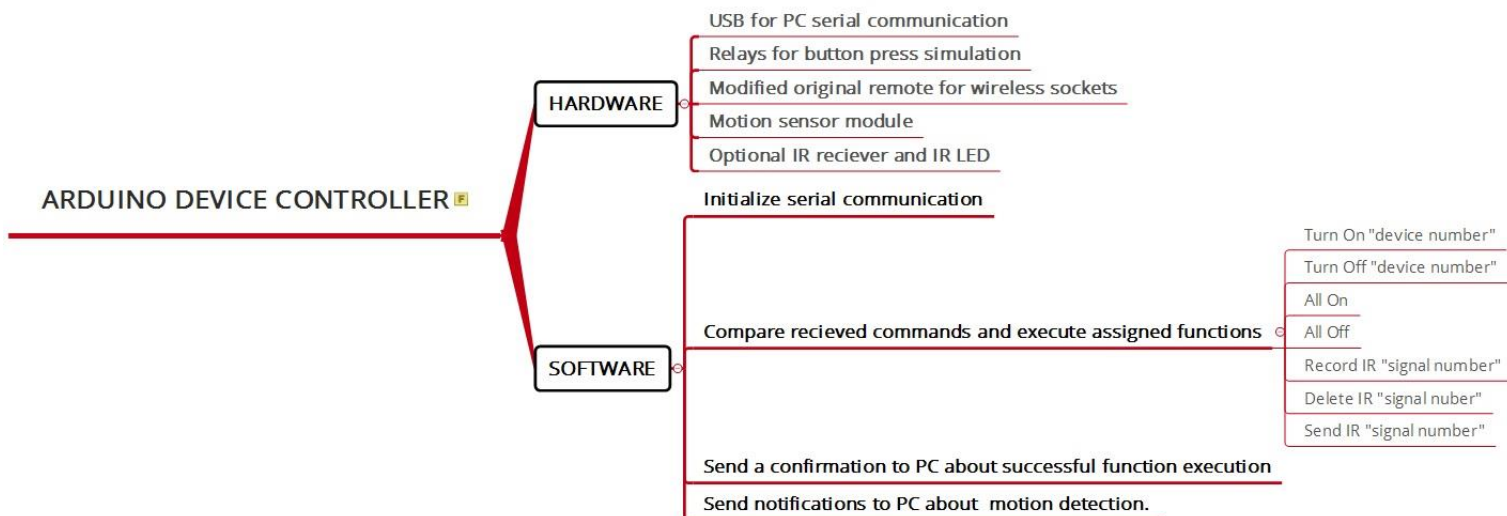
Když uživatel vysloví příkaz pro zapnutí či vypnutí některé z bezdrátově ovládaných zásuvek tak se odešle po sériové lince řetězec. První slovo tohoto řetězce je „turn“. Druhé slovo v odeslaném řetězci blíže upřesňuje, zda danou zásuvku vypnout či zapnout „on/off“. Na posledním místě je číslo zásuvky, kterou adresujeme. Tyto tři části jsou odděleny mezerou a Arduino v modulu si je dokáže separovat.

Zatím Arduino tyto příkazy jen dokáže přijmout a vykonat. V budoucnu je prostor komunikaci rozšířit o zpětnou vazbu. Po každém úspěšném přijetí a vykonání příkazu by se poslalo počítači potvrzení. Kdyby nepřišlo do stanovené časové hranice potvrzení, tak by se přenos opakoval.[8]



Obrázek 16 - schéma formátu posílaných příkazů USB modulu

5 Příkladný USB modul



Obrázek 17 - část myšlenkové mapy pro USB modul

Abych rozšířil Jarvisovi možnosti jak vám zjednodušit každodenní život, tak jsem vyvinul přídatný USB modul. Po připojení tohoto modulu k vašemu počítači můžete najednou začít Jarvisovi dávat příkazy, které budou manipulovat reálnými věcmi v domácnosti.

Součástí tohoto modulu jsou bezdrátově ovládané zásuvky. Díky nim můžete ovládat jakýkoliv spotřebič, který do nich zapojíte. Abych uvedl příklad rozmanitosti jeho využití tak můžete k zásuvkám připojit např. stolní lampičku, velkou lampu stojící uprostřed obývacího pokoje, televizi, elektrický přímotop a mnoho dalšího.

5.1 Hardware

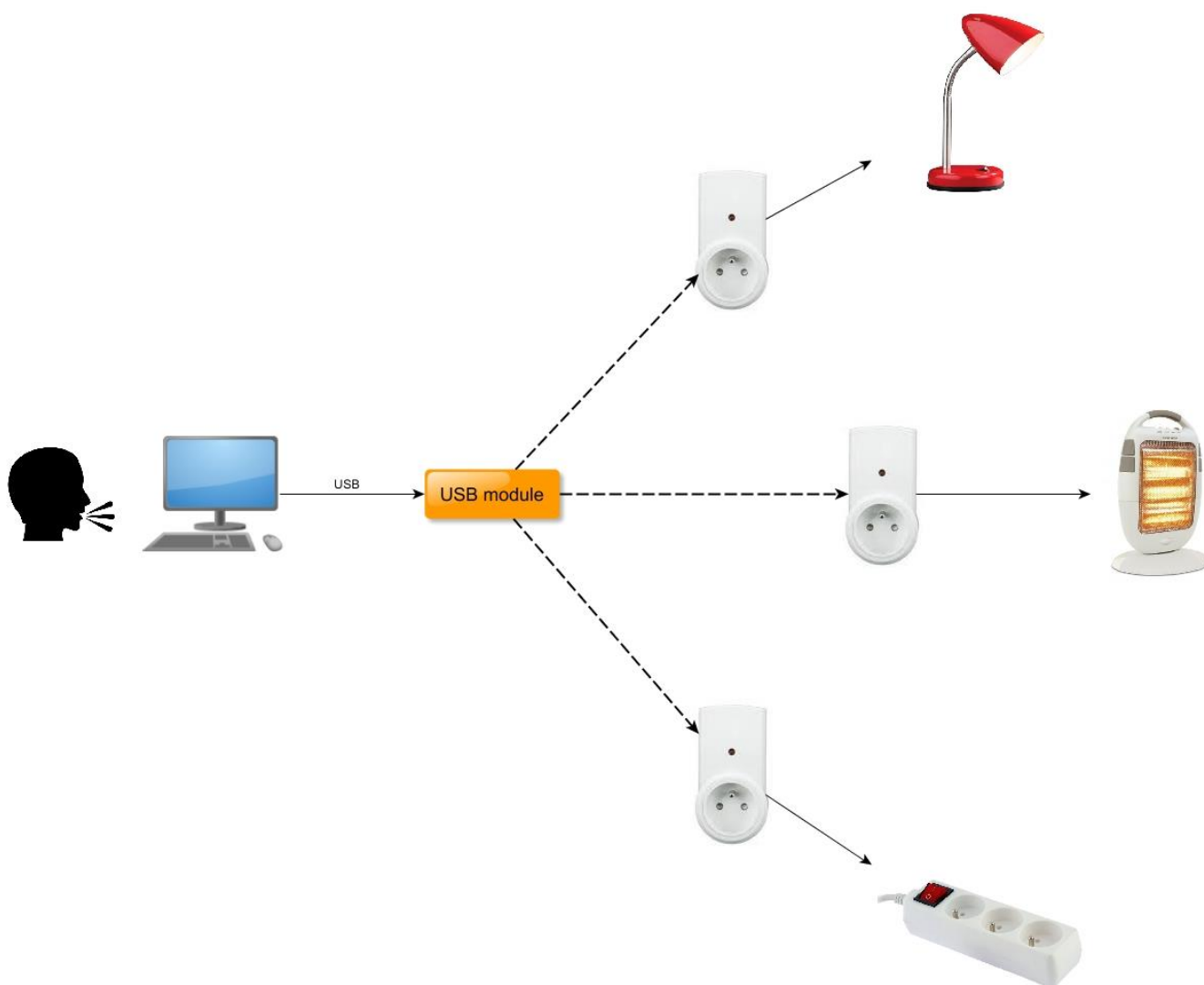
Centrálním ovládacím prvkem je mikropočítač Arduino. Ten je propojený USB kabelem s počítačem a sériově spolu komunikují. Na obou stranách musí být nastavené stejné parametry pro komunikaci. Tyto parametry jsou: přenosová rychlost 9600 bit/s, délka datového paketu je 8 bitů a používá se jeden stop bit. USB taky poskytuje celému modulu napájení, protože modul nemá velký odběr.

Z druhé strany je Arduino napojeno na relé přes které jsou simulovány stisky tlačítek na ovladači ze zakoupené a modifikované sestavy. Tímto jsem navázal na další prvek a to je set bezdrátově ovládaných zásuvek pomocí ovladače s tlačítky. Na ovladači jsou vždy dvě tlačítka pro každou zásuvku, jedno na vypnutí a druhé na zapnutí zásuvky. Zásuvky se dají rozmístit do vzdálenosti, dostačující pro pokrytí i nadstandardně rozměrných místností (až 35 m).

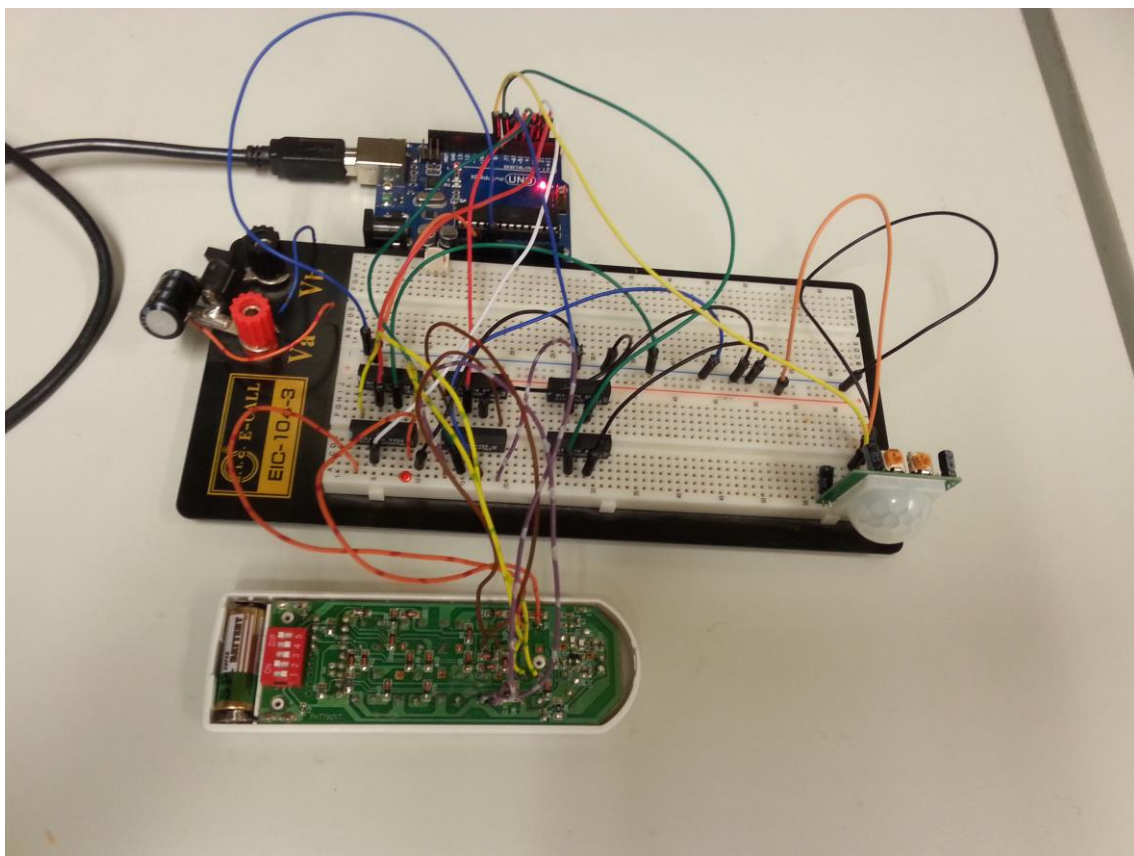
Abych mohl na ovladači simulovat stisknutí tlačítek tak jsem ho musel modifikovat. Zprvu jsem celý ovladač musel rozšroubovat a následně jsem paralelně k tlačítkům napájel relé. Jako poslední hardwarový prvek modul obsahuje infračervené čidlo na rozpoznání pohybu. Ve chvíli kdy je před čidlem rozpoznán pohyb tak je počítači odeslán signál. Po přijetí tohoto signálu Jarvis uživatele pozdraví, pokud nebyl v posledních 10 min rozpoznán pohyb tzn. uživatel přišel k PC.

5.2 Software

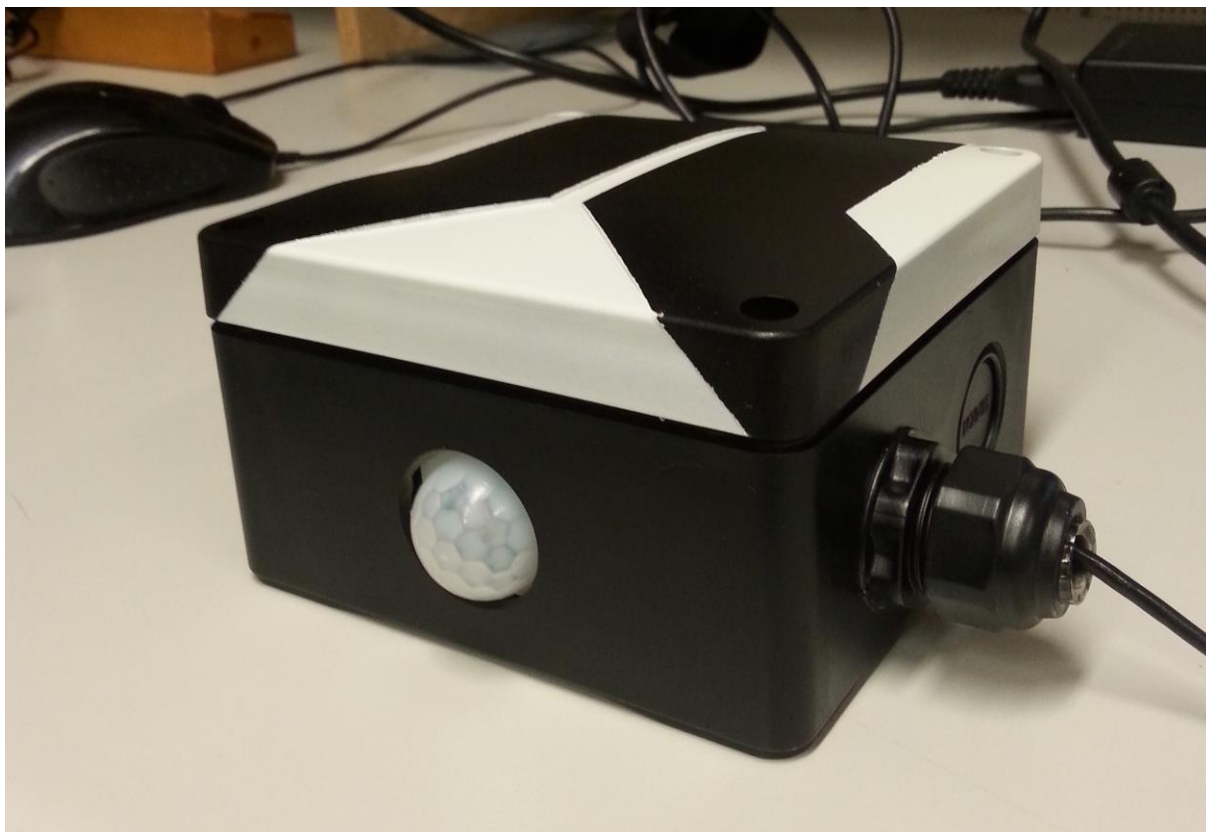
Po spuštění programu v počítači se jako první věc vykoná inicializace sériové komunikace. Poté už jen program očekává příkazy od počítače. Po přijetí příkazu se rozdělí podle jeho jednotlivých obsahových částí. Z nich už vyplývá rozhodování o vykonání správných funkcí, které od něho uživatel očekává. Příkazy se dělí na tři hlavní části. Adresovatelné, hromadné a předpřipravené příkazy na rozšíření o čtení a reprodukování infračervených signálů jako jsou například vysílány běžnými ovladači TV. Adresovatelné příkazy slouží k ovládání jednotlivých zásuvek. Hromadné jsou pro vypnutí či zapnutí všech zásuvek najednou a v poslední řadě zde popíšu svůj plán jak dokončit sekci pro práci s IR signály. Eventuální schopnost modulu nahrávat si IR signály a později je reprodukovat, by mohla umožnit naučit Jarvis ovládat např. vaši televizi či Hi-Fi sestavu. Je to opravdu krásná představa probuzení Jarvisem přáním pěkného dne, spuštěním noční lampičky a naladěním zpráv na televizi.



Obrázek 18 - schématický obrázek celého svstému



Obrázek 19 - prototypovací fáze USB modulu



Obrázek 20 - výsledný USB modul

6 Potenciál budoucího rozvoje

V budoucnu se nehodám ani zdaleka omezovat jen na „obyčejný“ set bezdrátových zásuvek a spouštění pár programů pomocí řeči. Je tu tolik možností jak Jarvise přenést do naprosto jiných rozměrů, proto jsem tomu věnoval celou kapitolu. Napadá mě obrovská škála možných rozšíření a vylepšení, které by Jarvisovi zprostředkovali možnost využití v nespočtu oborů a odvětví. Nemá smysl a ani není možné tu popsat všechna z nich, proto jsem tu vytvořil souhrn nejužitečnější a nejvýznamnějších možných expanzí seskupených podle oblasti využití.

6.1 Zdravotnictví

Způsob jak přenést Jarvise o trošku víc mezi nás do fyzického světa a zároveň tím pomoci velkému množství lidí, kteří si pomoc zaslouží nejvíce z nás a to jsou zdravotně znevýhodnění lidé. Obzvláště ti kteří za svůj osud nemohou, tak těm opravdu stojí za to pomoci. Jedna z možných variant pro expanzi v této oblasti, která by mohla nastartovat revoluci v lékařské přístrojové technice napomáhající seniorům i lidem se zdravotním postižením, by byl modul robotického manipulátoru.

Manipulátor třeba v podobě robotické ruky schopné o mnoho usnadnit vykonávání každodenních činností, které mnozí z nás považují za samozřejmé. Představte si situaci, kdy jste postižený člověk a kvůli každé malichernosti, kterou chcete vykonat, musíte volat asistentku, aby vám pomohla. Za prvé by vám mohlo přijít nepříjemné či nevhodné požádat někoho „otravovat“ i když to má asistentka v popisu práce a takováto věc by se neměla stát, ale třeba z důvodu nízkého sebevědomí zapříčiněného zdravotním postižením, by mohl takto člověk uvažovat. Za druhé je opravdu nepříjemné mít pocit závislosti na ostatních. Vůči robotické ruce by člověka nenapadli takovéto myšlenky. Díky Jarvisově provedení a přizpůsobivosti, by ho člověk vnímal nejen jako dobrého a ochotného kamaráda, ale dokonce by se s ním mohl ztotožnit a přijmout ho jako součást sebe.

Stejně jako když mozek pošle zdravé ruce signál a ona se podle něj pohne tak by tento pocit mohli lidé využívající Jarvise okusit. Stalo by se tak zaměněním signálu z mozku za hlasový příkaz k pohybu rukou. Díky úžasné schopnosti adaptace lidského těla vůči novým věcem, by si po určité době uživatelé opravdu zvykli na takovýto způsob pohybu a dodalo by jim to tu potřebnou jiskru do života.

Určitě bych chtěl při dalším rozvoji tohoto projektu navštívit nějakou instituci napomáhající zdravotně znevýhodněným lidem a více proniknout do jejich života a uvědomit si jakým problémům musí každodenně čelit. Taková návštěva by pomohla při rozhodování jakým směrem se konkrétně dát a jak bych mohl nejvíce těmto lidem pomoci a přinést i trošku radosti do života.

6.2 Vývoj elektroniky

Dalším oborem, pro který lze Jarvise přizpůsobit je vývoj elektroniky. Elektronik při své běžné činnosti osazování desek plošných spojů s plnými rukami náradí a součástek by se nemusel rozptylovat neustálým přerušováním práce, aby si mohl jít na počítač vyhledat na internetu data-sheet dané součástky. To by totiž zvládal Jarvis kompletně za něho. Technickému pracovníkovi by stačilo pouze nadiktovat jméno součástky a Jarvis by prohledal známé spolehlivé servery a po nalezení požadovaného dokumentu jej zobrazil na displeji. Také by bylo možné nadefinovat Jarvisovi klávesové zkratky, které by uměl „mačkat“ za vás. Tato možnost by udělala práci v právě využívaných programech mnohem efektivnější a plynulejší.

6.3 Shromažďování informací

Také jsem zvažoval vyvinutí funkcí pro vyhledávání na internetu a shromažďování požadovaných informací o zadaném subjektu.

Opět uvedu příklad. V průběhu výběrového řízení pro přijetí nového zaměstnance, je potřeba o tomto člověku zjistit co nejvíce informací z důvodu zvýšení šancí pro výběr nejvhodnější osoby. Jarvis by byl schopný pouhým zadáním celého jména požadované osoby stáhnout obsah ze známých webových stránek a serverů, odkud lze využít nechtěného úniku informací o subjektu. Většina dnešních lidí je zaregistrována aspoň na jedné stránce odkud je možné o vás něco zjistit. Například sociální síť facebook, kde mnoho z nás nemá své profily uzamčené a informace z nich jsou veřejně přístupné. Ze zdrojového kódu takovýchto stránek by Jarvis byl schopný například pomocí regulárních výrazů vyfiltrovat a nashromáždit zajímavé informace. Následně by ze všech nashromážděných informací z nejrůznějších webových stránek kde je o vás zmínka vybral ty nejdůležitější a nejdůvěryhodnější a automaticky o vás sepsal zprávu.

Jelikož je více lidí se stejným jménem, tak by vám Jarvis tyto informace roztřídil, podle toho odkud je čerpal. Rozdělení by bylo na profily jednotlivých lidí s přiloženou fotografií, pokud byla na internetu nalezena a stručným výpisem základních informací (jméno, příjmení, datum narození, bydliště, atd.). Takto připravené informace by se zobrazily jako seznam profilů nalezených osob.

Na základě toho, byste byli schopni posoudit, která ze všech nalezených osob je ta kterou hledáte. Vám už by jen stačilo si podle fotografie či pravděpodobných očekávaných informací vybrat správný profil a přečíst si o něm celý detailnější soupis. V téhle oblasti je podle mě skrytý velký potenciál a určitě by byl o tak mocný nástroj velký zájem.

Závěr

Hlavním cílem této práce bylo vytvořit virtuálního domácího asistenta, který by svým hlasovým ovládním usnadňoval práci na počítači a nejen to ale dokonce aby vám i umožňoval propojení s reálným světem v podobě ovládním domácích spotřebičů.

Před tím než jsem začal na Jarvisovi pracovat tak jsem si celý projekt důkladně rozvrhl, abych nezapomněl na žádnou důležitou část a postupoval při práci systematicky. Jako první jsem pak vytvořil program hlasového asistenta a za druhé jsem vyvinul USB modul. Díky pečlivému plánování se mi podařilo dosáhnout většiny očekávaných výsledků.

Díky zhotovenému USB modulu umožňujícímu toto propojení je zde obrovská příležitost rozšíření do budoucna. Otevírá se možnost vytvořit například manipulátor napomáhající zdravotně postiženým, ovládním polohovatelného lůžka a podobně.

Mám pro Jarvise do budoucna velké vize, které bych rád realizoval. Mezi ně patří vytvoření robotického pomocníka pro zdravotně postižené, vyvinutí softwaru pro hledání potřebné dokumentace k součástkám pro elektrotechniky a třeba i rozšíření umožňující shromáždění informací z celého internetu o určité osobě. Tato práce mi přinesla mnoho znalostí a zkušeností, které se mi budou určitě hodit při budoucí práci v oboru IT.

Literatura

- [1] Logo Xmind. *XMind* [online]. Neznámé: XMind Ltd., Neznámý [cit. 2016-02-13]. Dostupné z: <http://www.xmind.net/>
- [2] Pozadí indikátoru. *Vector Stash* [online]. Neznámé: Neznámé, 2014 [cit. 2016-02-13]. Dostupné z: <http://www.vectorstash.com/content/rough-circle>
- [3] Rozpoznávání hlasu. *MSDN Microsoft* [online]. Neznámé: Microsoft, Neznámý [cit. 2016-02-13]. Dostupné z: [https://msdn.microsoft.com/en-us/library/office/hh361683\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/hh361683(v=office.14).aspx)
- [4] Animace v C# WPF. *C# Corner* [online]. Neznámé: Shubham Srivastava, 2012 [cit. 2016-02-13]. Dostupné z: <http://www.c-sharpcorner.com/UploadFile/0524d6/animation-in-wpf-application/>
- [5] Vytvoření bezrámečkového okna v C# WPF. *Paul Sheriff's Blog for the Real World* [online]. Neznámé: Neznámé, 2009 [cit. 2016-02-13]. Dostupné z: <http://weblogs.asp.net/psheff/creating-border-less-windows-in-wpf>
- [6] Práce s komponentou listView. *StackOverflow* [online]. Neznámé: Neznámé, 2014 [cit. 2016-02-13]. Dostupné z: <http://stackoverflow.com/questions/15865829/add-items-to-columns-in-a-wpf-listview>
- [7] Serializace v C#. *MSDN Microsoft* [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-13]. Dostupné z: [https://msdn.microsoft.com/cs-cz/library/system.runtime.serialization.formatters.binary.binaryformatter\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/system.runtime.serialization.formatters.binary.binaryformatter(v=vs.110).aspx)
- [8] Sériová komunikace v C# WPF. *Paul Sheriff's Blog for the Real World* [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-13]. Dostupné z: <http://csharp.simpleserial.com/>

Seznam obrázků

Obrázek 1 - logo programu XMind[1].....	11
Obrázek 2 - část myšlenkové mapy pro indikátor interakcí	12
Obrázek 3 - indikátor[2].....	13
Obrázek 4 - část myšlenkové mapy okna pro přizpůsobení příkazů.....	14
Obrázek 5 - vzhled okna pro přizpůsobení příkazů	15
Obrázek 6 - část myšlenkové mapy pro formulář základních informací	16
Obrázek 7 - vzhled okna formuláře základních informací.....	16
Obrázek 8 - část myšlenkové mapy pro okno obecného nastavení	17
Obrázek 9 - vzhled okna obecného nastavení.....	17
Obrázek 10 - část myšlenkové mapy pro okno virtuálního ovladače.....	18
Obrázek 11 - vzhled okna dálkového ovladače	18
Obrázek 12 - část myšlenkové mapy pro třídu CommandProcessor	19
Obrázek 13 - schéma formátu psaní funkcí.....	19
Obrázek 14 - část myšlenkové mapy pro třídu UserInfo	20
Obrázek 15 - část myšlenkové mapy pro třídu USBcontroller	21
Obrázek 16 - schéma formátu posílaných příkazů USB modulu	21
Obrázek 17 - část myšlenkové mapy pro USB modul	22
Obrázek 18 - schématický obrázek celého systému.....	23
Obrázek 19 - prototypovací fáze USB modulu	24
Obrázek 20 - výsledný USB modul.....	24

Seznam symbolů, veličin a zkratek

HW	hardware
USB	universal serial bus
Arduino	malý jednodeskový počítač
C#	programovací jazyk
C++	multiparadigmatický programovací jazyk
WPF	windows presentation foundation
XML	rozšiřitelný značkovací jazyk
HTML	značkovací jazyk pro tvorbu webových stránek
PC	personal computer
OS	operační systém
IT	informační technologie
TV	television
IR	infra red

Seznam příloh

1. Celá myšlenková mapa projektu vytisknutá na papír formátu A3
2. Dodatečné fotografie
3. CD se zdrojovými kódy
 - Zdrojový kód počítačového programu
 - Zdrojový kód programu v Arduinu

Přílohy

Příloha č. 2

