



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ALGORITMUS PRO ROZEZNÁVÁNÍ ŘEČOVÝCH PŘÍKAZŮ NA MIKROKONTROLÉRU

SIMPLE SPEECH RECOGNITION ALGORITHM FOR MICROCONTROLLER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Lukáš Vedra

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Zbyněk Fedra, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Lukáš Vedra
Ročník: 3

ID: 106866
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Algoritmus pro rozeznání řečových příkazů na mikrokontroleru

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti rozeznávání několika jednoduchých příkazů pomocí mikrokontroleru AVR. Definujte potřebné parametry vzorkování signálu, délky zpracovávané sekvence a počtu rozeznatelných příkazů s ohledem na možnosti cílové platformy. Otestujte základní algoritmus na získání otisku příkazu a jeho úspěšnost.

Pokuste se optimalizovat parametry algoritmu rozpoznávání příkazů a realizujte odpovídající HW část. Vytvořte program, který umožní naučit dané zařízení několik příkazů a ty pak správně rozpoznávat. Po domluvě s vedoucím realizujte konkrétní aplikaci (např. hlasové ovládání stmívače světla).

DOPORUČENÁ LITERATURA:

[1] HARISON, A., SHAH CH. Voice Recognition Robotic Car [online]. Cornell University ECE 4760 Designing with Microcontrollers Final Projects, 2006 - [cit. 20.5.2010]. Dostupné na [www: http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/avh8_css34/avh8_css34/index.html](http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/avh8_css34/avh8_css34/index.html)

[2] MANN, B. C pro mikrokontroléry. BEN - technická literatura, Praha 2003

Termín zadání: 7.2.2011

Termín odevzdání: 27.5.2011

Vedoucí práce: Ing. Zbyněk Fedra, Ph.D.

prof. Dr. Ing. Zbyněk Raida

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 následku 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 problematikou rozeznávání řečových příkazů na mikrokontroléru firmy Atmel rodiny AVR. Práce popisuje způsoby rozeznávání, vliv volby vzorkovací frekvence, délky použité sekvence a počet možných příkazů s ohledem na výpočetní rychlost a paměťovou kapacitu mikrokontroléru. Dále je testována úspěšnost získávání příkazů. Rovněž rozebírá možnosti porovnávání uloženého otisku příkazu ve slovníku resp. v databázi mikrokontroléru a právě vysloveného slova. Práce se také zabývá mechanickou konstrukcí zařízení.

KLÍČOVÁ SLOVA

Lineární filtrace, vzorkovací frekvence, porovnávací funkce, mikrokontrolér

ABSTRACT

The bachelor thesis deals with speech recognition commands on the microcontroller Atmel AVR family. The thesis describes methods of recognition, influence of a sampling frequency selection, lengths of a sequence used, and a number of possible orders with respect to computing speed and memory capacity of the microcontroller. Furthermore, successful acquisition of orders is tested. It also discusses comparison options of an imprint saved in a dictionary command, or more precisely in a database of the microcontroller and a word which has just been uttered. The work also deals with a mechanical system design of the device.

KEYWORDS

Linear filtration, sample rate, comparison functions, microcontroller

VEDRA, L. Algoritmus pro rozeznání řečových příkazů na mikrokontroléru. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2011. 31 s., 6 s. příloh. Bakalářská práce. Vedoucí práce: Ing. Zbyněk Fedra, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Algoritmus pro rozeznání řečových příkazů na mikrokontroléru 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 bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, 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. Zbyňkovi Fedrovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	viii
Seznam tabulek	ix
Úvod do problematiky	1
1 Základní způsoby rozeznávání řeči	3
1.1 Rychlá Fourierova transformace	3
1.2 Lineární filtrace	5
1.2.1 Filtr typu FIR	6
1.2.2 Filtr typu IIR	7
2 Parametry algoritmu pro rozpoznání příkazů	8
2.1 Parametry vzorkování signálu	8
2.2 Délka zpracovávané sekvence	9
2.3 Počet rozeznatelných příkazů	10
3 Algoritmus na získání otisku příkazu	11
3.1 Nastavení přerušení časovače/čítače a A/D převodu	12
3.2 Měření hluku v místnosti	12
3.3 Funkce filtrů	13
3.3.1 Volba filtrů	13
3.3.2 Získání parametrů filtrů	15
3.4 Vytvoření otisku	16
3.4.1 Vytvoření otisku v PC	16
3.4.2 Vytvoření otisku v algoritmu	17
3.5 Porovnávací funkce	17
3.5.1 Korelace	18
3.5.2 Euklidovská vzdálenost	18
3.5.3 Manhattanská vzdálenost	18
3.6 Vložení nahrávací funkce	19
3.7 Ovládací funkce	19

4	Optimalizace algoritmu	20
4.1	Počet filtrů.....	20
4.2	Velikost okna a délka sekvence	21
4.3	Úspěšnost získávání otisku příkazu	22
4.3.1	Počet a výběr slov	22
4.3.2	Vliv vzdálenosti od mikrofonu	23
5	Realizace zařízení	25
5.1	Mikrofon	25
5.2	Zesilovač.....	26
5.3	Mikrokontrolér.....	27
5.3.1	Požadavky algoritmu na mikrokontrolér	28
5.3.2	Použitý mikrokontrolér	28
5.4	Napájení zařízení	29
5.4.1	Spínaný zdroj	29
5.4.2	Výběr stabilizátoru.....	29
5.5	Mechanické provedení	30
6	Závěr	31
	Literatura	32
	Seznam příloh	34

SEZNAM OBRÁZKŮ

Obr. 1 Rozdělení signálu na úseky (okna).....	3
Obr. 2 Převod z časové oblasti do kmitočtové.....	4
Obr. 3 Využití filtru pro odstranění rušivého signálu	5
Obr. 4 Tři nejpoužívanější aproximace filtrů (převzato z [3]).....	6
Obr. 5 Porovnání slova "rozsvítit" a "rozsvit"	9
Obr. 6 Blokové schéma algoritmu pro rozeznání příkazu	11
Obr. 7 Přechodové charakteristiky filtrů.....	13
Obr. 8 Kaskádní realizace IIR filtrů (převzato z [4]).....	14
Obr. 9 Realizace filtru v algoritmu	15
Obr. 10 Vytvoření otisku v algoritmu.....	17
Obr. 11 Grafická závislost počtu slov na počtech filtrů při stejné velikosti otisku	20
Obr. 12 Rozdělení délky sekvence na jednotlivá okna	21
Obr. 13 Průměrná úspěšnost rozpoznávání příkazů.....	24
Obr. 14 Blokové schéma zařízení	25
Obr. 15 Směrové charakteristiky mikrofónů	26
Obr. 16 Obvodové schéma zesilovacího bloku	26
Obr. 17 Kmitočtová charakteristika dvou operačních zesilovačů se stejným zesílením (převzato z [14]).....	27
Obr. 18 Obvodové zapojení pomocného zdroje (převzato z [15])	29
Obr. 19 Zapojení stabilizátoru (převzato z [17])	30
Obr. 20 Elektroinstalační krabice průměru 68 mm (použito z [17])	30

SEZNAM TABULEK

Tab. 1 Přehled počtu strojových cyklů v závislosti na řádu filtru a použitého programovacího jazyka (převzato z [8])	9
Tab. 2 Přehled velikosti proměnných (převzato z [9])	10
Tab. 3 Počet strojových cyklů funkce filtry.....	21
Tab. 4 Úspěšnost rozpoznávání osmi příkazů při použití 5 filtrů	22
Tab. 5 Úspěšnost rozpoznávání čtyř příkazů při použití 8 filtrů	23
Tab. 6 Úspěšnost rozpoznávání příkazů v závislosti na vzdálenost od mikrofonu bez zdroje hluku	23
Tab. 7 Úspěšnost rozpoznávání příkazů v závislosti na vzdálenost od mikrofonu se zdrojem hluku	24
Tab. 8 Srovnání mikrokontrolérů ATmega.....	28

ÚVOD DO PROBLEMATIKY

Řeč člověk reprodukuje již odedávna a je základním prvkem komunikace, která mimo jiné slouží k získání informací. Již na začátku 20. století byla snaha vytvořit algoritmus pro rozeznávání řeči. Později byly algoritmy vytvořeny a jsou dále vyvíjeny. S tímto poznatkem bylo napsáno mnoho knih a natočeno spousta filmů typu science fiction. Dnes se tato science fiction stává skutečností. Funkce rozpoznávání řeči je obsažena téměř v každém mobilním telefonu, různých automatech, osobních počítačích apod. Je otázka času, kdy pomocí hlasu budeme ovládat důležité stroje jako například automobily a vznikne vyspělá umělá inteligence.

Zařízení řízených pomocí hlasu je celá řada, jak již bylo uvedeno. Existuje mnoho verzí algoritmů, které jsou různé svou obtížností a výslednou úspěšností. Algoritmus použitý v této práci je primitivní a stavěn pro 8bitové procesory firmy Atmel rodiny AVR. Tato platforma je omezena pamětí a rychlostí zpracování instrukcí. Výsledný algoritmus musí z těchto poznatků vycházet a vytvořit kompromis mezi těmito vlastnostmi procesoru. Tato práce vychází z projektu [1] vzniklého v roce 2006 v americké univerzitě Cornell University.

Práce je rozdělena do pěti kapitol, které vycházejí ze zadání. Kapitola 1 popisuje teorii způsobu rozeznávání řeči. Jsou zde porovnávány dvě metody, a to rychlá Fourierova transformace a lineární filtrace. V kapitole 2 jsou rozebrány základní parametry pro rozpoznání příkazů, jako je vzorkovací frekvence, délka zpracovávané sekvence a počet rozeznatelných příkazů. V kapitole 3 je podrobně popsána funkce algoritmu na získání a rozeznání příkazu. Obsahuje nastavení jednotlivých registrů, volbu vhodného typu filtru, samotné vytvoření otisku, použité porovnávací funkce, vložení nahrávací funkce a výstupní řízení výkonového prvku. Kapitola 4 popisuje optimalizaci algoritmu, tzn. volbu počtu filtrů, velikost okna, délky sekvence, vzdálenost od zařízení a jejich vliv na úspěšnost vyhodnocení příkazu. V kapitole 5 je uvedena realizace konkrétního zařízení. Obsahuje popis a volbu použitých součástek a následně mechanický způsob provedení.

Algoritmus nevyužívá vytvořených modelů řečových signálů, které by byly pro danou platformu nepoužitelné. Systém zpracovává analogový signál z vhodně zvoleného typu mikrofону. Ten však nemá dostatečnou úroveň, a proto je amplitudově i frekvenčně upraven v zesilovacím obvodu. Takto upravený signál je přiváděn do mikrokontroléru a analogově-digitálním převodníkem převeden na diskretní signál. Měřením hluku v okolí zjistí systém prahovou hodnotu pro detekci slova. Při vhodné úrovni začne systém nahrávat a příkaz analyzovat. Příchozí příkaz je rozdělen na určitý počet oken. Velikost okna stanoví celkový počet vzorků daný délkou sekvence a vzorkovací frekvencí. Každé okno je filtrováno použitou metodou, tedy lineární filtrací, která obsahuje řadu matematických operací danou volbou filtru. Pro každé okno je výstupem z jednoho filtru jedna hodnota, která vznikne odezvou filtru. Jestliže je použito osmi filtrů a šestnácti oken, je získáno 128 hodnot. Tyto odezvy filtrů tvoří takzvaný otisk příkazu. V tomto otisku není uchovávána informace o vzorkování, barvě a tónu hlasu. Otisk tvoří pouze kopii informace o slově danou aktuální amplitudou příchozího signálu. Takto vytvořený otisk je porovnáván s otiskem vzniklým dříve v PC pomocí matematicko-technického softwaru nebo s otiskem vytvořeného nahrávací

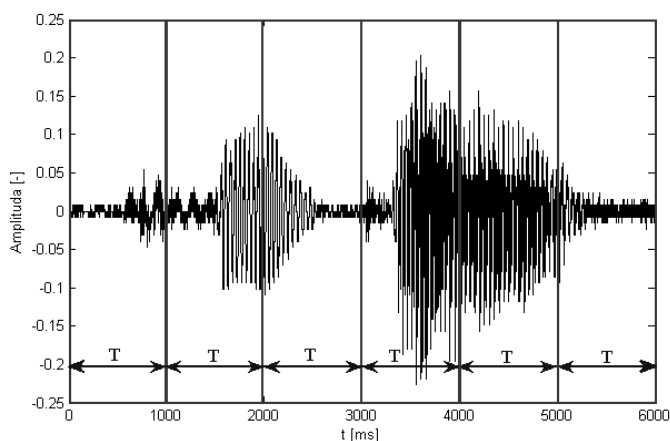
funkcí umístěného přímo v mikrokontroléru. Oba otisky, jak už vytvořený v PC nebo v mikrokontroléru, jsou uloženy v datovém paměťovém prostoru. V datové paměti mikrokontroléru je tedy vytvořena databáze příkazů, která je nazývána slovníkem. Tento slovník obsahuje určitý počet slov, který je dán kompromisem použitého typu mikrokontroléru, počtem filtrů a jejich úspěšnosti. V případě shody porovnání otisku uloženého v mikrokontroléru a otisku právě vysloveného slova je možno ovládat periferie mikrokontroléru resp. výstupní část zařízení.

1 ZÁKLADNÍ ZPŮSOBY ROZEZNÁVÁNÍ ŘEČI

V možnosti rozeznávat řeč je zahrnuto mnoho oborů, které se touto problematikou zabývají. Patří sem například i lingvistika, analýza signálů, neuronové sítě apod. S ohledem na cílovou platformu, tedy 8 bitový mikrokontrolér AVR, můžeme možnosti rozpoznávání řeči rozdělit na dvě nejpoužívanější:

- A) Rychlá Fourierova transformace
- B) Lineární filtrace

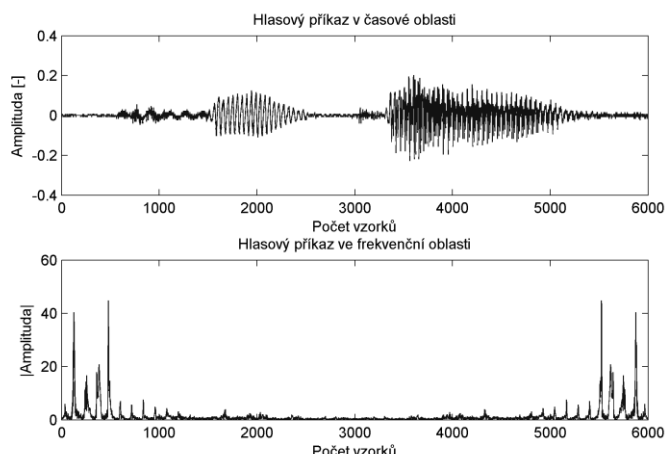
Jednotlivé metody nelze zpracovávat spojitě, proto se signál rozdělí na několik úseků, tzv. okna (frame). Velikost všech oken se volí stejná o rozměru T . Rozdělení signálu na úseky je zobrazeno na obr. 1.



Obr. 1 Rozdělení signálu na úseky (okna)

1.1 Rychlá Fourierova transformace

Při využití Fourierovi transformace (dále jen FT) se jednotlivá okna převedou z časové oblasti do frekvenční, tzn. není zpracovávána výchylka signálu v čase, ale její harmonické složky. Každá harmonická složka je definována svojí frekvencí a amplitudou. Srovnání vstupního signálu a následný převod do kmitočtové oblasti je uveden na obr. 2.



Obr. 2 Převod z časové oblasti do kmitočtové

V případě použití FT by při porovnávání signálu s uloženým otiskem bylo použito pouze pár vzorků harmonických složek, v nichž je uložena informace o slově. Nejvhodnější způsob porovnávání vzorků pro tuto metodu je metoda Euklidovy vzdálenosti. Porovnávací funkce jsou popsány v kapitole 3.5.

Pro převod z časové do kmitočtové oblasti se používá diskrétní Fourierova transformace (dále jen DFT), která zajišťuje kmitočtovou analýzu. Pro výpočet DFT se využívá následujícího vztahu [2]:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W^{nk} \quad k = 0, 1, \dots, N-1, \quad (1)$$

kde N značí délku sekvence, $x(n)$ hodnoty vstupní sekvence a W^{nk} komplexní exponenciálu, která je dána vztahem [2]:

$$W^{nk} = e^{-j2\pi / N}, \quad (2)$$

kde N opět značí délku sekvence.

Samotný výpočet je velmi složitý, proto v roce 1965 byla objevena optimalizace algoritmu pro DFT. Algoritmus zvaný rychlá Fourierova transformace (dále jen FFT). Existuje několik verzí algoritmu. Nejpoužívanější jsou dvě skupiny algoritmů.

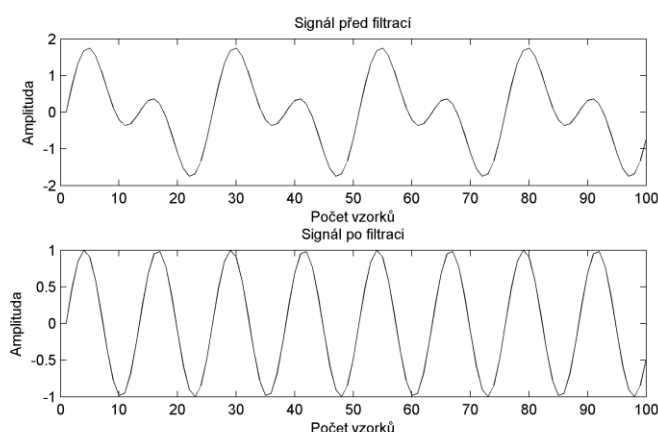
1. DIT (Decimation In Time) – vstupní sekvence hodnot je rozdělena na dvě poloviny a to na členy se sudou a lichou částí. Další posloupnost se dělí rovněž na sudou a lichou část.
2. DIF (Decimation In Frequency) – vstupní sekvence hodnot je rozdělena na dvě poloviny, ale bez rozčlenění na sudé a liché členy. Vzniklá posloupnost se opět dělí na poloviny atd.

Tato možnost je pro zadaný projekt nepoužitelná, protože vstupní sekvence má 2000 hodnot a pro zadanou platformu, by byl výpočet zdlouhavý. Tato metoda se

využívá u signálových procesorů a je přesnější než pomocí banky filtrů.

1.2 Lineární filtrace

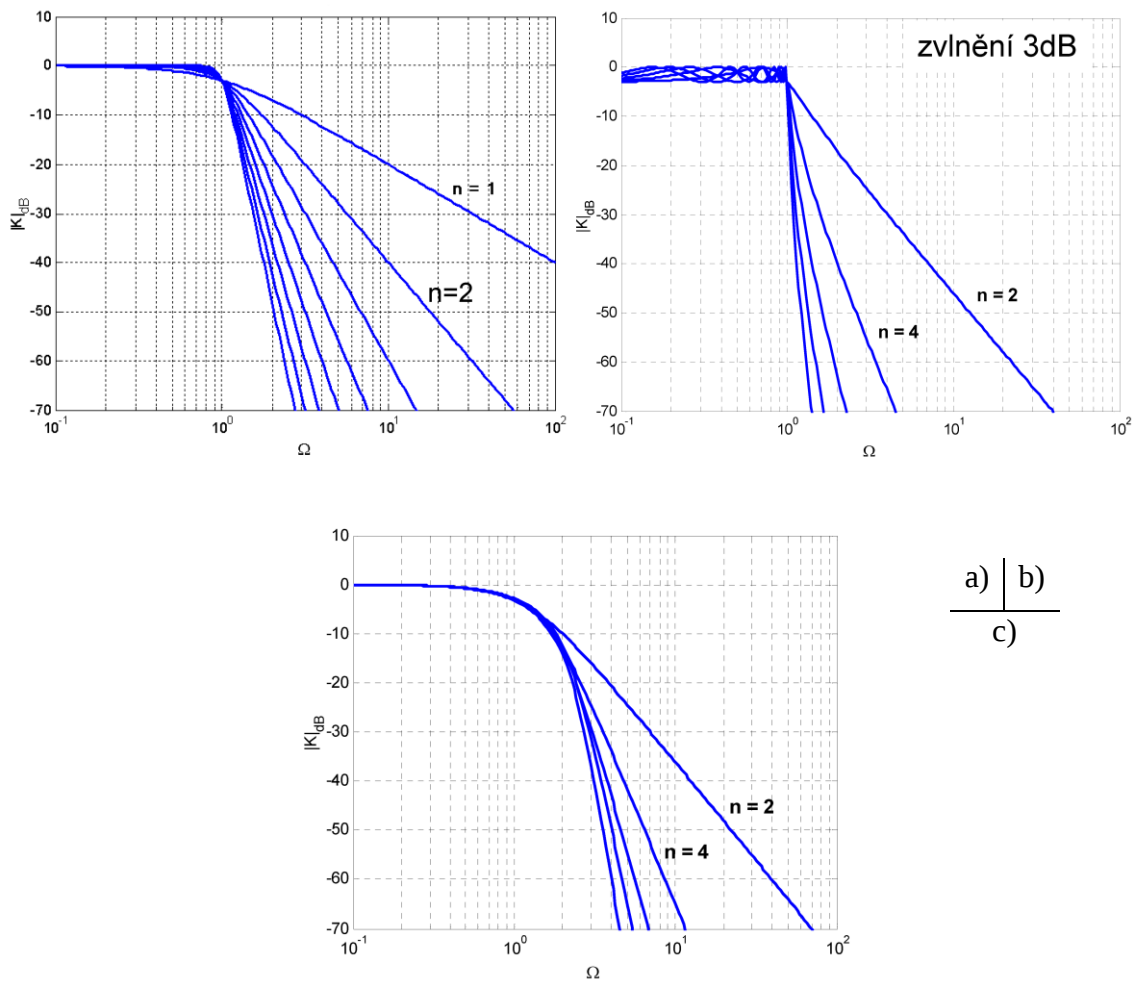
Předchozí metoda využívá frekvenční oblasti k získání otisku, oproti tomu lineární filtrace zpracovává vzorky v časové oblasti. Opět je nutno odstranit vysoké frekvence ze vstupního signálu, které nepřináší informace o vysloveném slově, ale zvyšují chybu v úspěšnosti porovnávání a získání příkazu. Část filtrace je provedena analogově přímo na obvodu zesilovače a část je provedena číslicově na mikroprocesoru pomocí digitálních filtrů.



Obr. 3 Využití filtru pro odstranění rušivého signálu

Digitální lineární filtrace má obrovské využití ve zpracování signálu. Jejich účel je především potlačení některých frekvenčních složek a zdůraznění jiných. Používají se pro tvarování signálu, ovlivnění spektra signálu, výběr rušivého signálu nebo pro tento projekt využívaný výběr užitečného signálu. Na obr. 3 je zobrazeno využití filtru pro odstranění rušivého signálu. Typy jednotlivých frekvenčních charakteristik jsou stejné jako u analogové filtrace, tzn. dolní propust, horní propust, pásmová propust a pásmová zadrž. Rovněž typ aproximace je totožný s analogovým. Podle toho jaké vlastnosti od filtrů požadujeme, takovou volíme aproximaci. Níže jsou uvedeny tři nejpoužívanější aproximace a jejich vlastnosti:

1. Butterworthova aproximace (viz obr. 1a) – maximálně plochá charakteristika, malá strmost přechodu mezi pásmy
2. Čebyševova aproximace (viz obr. 1b) – zvlnění v propustné části, velká strmost přechodu mezi pásmy, zákmity na přechodové odezvě
3. Besselova aproximace (viz obr. 1c) – konstantní skupinové zpoždění, zachování tvaru signálu



Obr. 4 Tři nejpoužívanější aproximace filtrů (převzato z [3])

a) Butterworthova aproximace b) Čebyševova aproximace c) Besselova aproximace

Digitální filtry dále rozdělujeme do dvou skupin podle typu algoritmu:

1. FIR (Finite Impulse Response)
2. IIR (Infinite Impulse Response)

1.2.1 Filtr typu FIR

Jak už z názvu vyplývá, jedná se o filtr s konečnou impulsní charakteristikou. Filtr nemá zpětnovazební smyčku, tedy nerekurzivní struktura realizace, a z tohoto důvodu je vždy stabilní. Výstupem je konečná diskrétní konvoluce vstupního signálu s impulsní charakteristikou [4]:

$$y_n = \sum_{k=0}^{N-1} x_{n-k} h_k, \quad (3)$$

kde x_{n-k} značí hodnoty vstupní sekvence, h_k hodnoty impulsní charakteristiky filtru.

Nejdůležitější výhody a nevýhody FIR filtrů

- jsou jednodušší na návrh,
- jsou vždy stabilní,
- jsou náročnější na počet operací jednoho vzorku výstupního signálu.

1.2.2 Filtr typu IIR

Filtr s nekonečnou impulsní charakteristikou, který obsahuje zpětnovazební smyčku, tedy rekurzivní struktura realizace. Přenos je dán polynomem a při špatně zvolených nulách a pólech může nastat situace, kdy systém bude nestabilní. Filtr je dán diferenční rovnicí [4]:

$$y_n = \sum_{i=0}^r L_i x_{n-i} - \sum_{i=0}^m K_i y_{n-i}, \quad (4)$$

kde L_i , K_i jsou realizační konstanty, x_{n-i} hodnoty vstupní sekvence, y_{n-i} hodnoty zpětnovazební sekvence.

Nejdůležitější výhody a nevýhody IIR filtrů

- jsou složitější na návrh,
- mohou být nestabilní, a proto je nutno ověřovat stabilitu,
- jsou méně náročné na počet operací jednoho vzorku výstupního signálu,
- vyšší řády se realizují kaskádní nebo paralelním spojením bloků nižší složitosti.

2 PARAMETRY ALGORITMU PRO ROZPOZNÁNÍ PŘÍKAZŮ

Aby algoritmus byl plně funkční, nemůže být signál znehodnocen snímáním a nesmí nést příliš mnoho nadbytečných informací. Tyto podmínky definují základní parametry pro snímání signálu.

2.1 Parametry vzorkování signálu

Zvukový signál má frekvenční pásmo dané vlastnostmi lidského ucha. Tento rozsah nabývá hodnot od 16 Hz do 16 kHz a je velmi individuální. Pro program je zbytečné využívat celé pásmo, ale jen řečový signál, který je od 300 Hz do 3,4 kHz. Řečový signál bývá v telefonní technice snímán do 4 kHz, přibližně 600 Hz je použito jako rezerva. Shannonův teorém (nebo též Nyquistův teorém, Kotělnikovův teorém apod.) říká, že vzorkovaná frekvence musí být minimálně rovna nebo větší než frekvence vzorkovaného signálu. Vzorkovací teorém je uveden v následujícím vztahu [5]:

$$f_v \geq 2 f_m, \quad (5)$$

kde f_v značí vzorkovací frekvenci a f_m je horní mezní kmitočet vzorkovaného signálu.

Pokud má tedy vzorkovaný signál horní mezní frekvenci 4 kHz, měla by být jeho vzorkovaná frekvence minimálně 8 kHz, tzn. každých 125 μ s je získán nový vzorek. Pokud není systém schopen toto vzorkování zajistit, může dojít ke ztrátě nebo znehodnocení informace. Tomuto problému lze zamezit použitím antialiasingového filtru před vstup A/D převodníku. Tento filtr je dolní propust s vylepšenými vlastnostmi, jako je například strmější útlum, který však způsobuje větší fázový posun.

Z datasheetu procesoru [6] a z projektu „Hlasem rozpoznávaný zabezpečovací systém“ [7] lze odvodit vztah:

$$\text{počet cyklů} = \frac{1}{f_v} f_{osc}, \quad (6)$$

kde f_v značí vzorkovací frekvenci, f_{osc} taktovací frekvenci procesoru a výsledná hodnota udává počet strojových cyklů, které je možné použít během příchozího nového vzorku. Jestliže je dosazeno za vzorkovací frekvenci 8 kHz a za taktovací frekvenci nejvyšší možná, která je podporována procesorem 16 MHz, lze vypočítat ze vztahu (6) hodnotu 2000 cyklů. Tento vzorkovací kmitočet je možno použít pro přibližně 4 filtry podle tab. 1 (převzato z [8]). Pro přesnější výsledek je nutno použít filtrů více. Jejich počet strojových cyklů převyšuje hodnotu 2000. Z tohoto důvodu je použita vzorkovací frekvence 4 kHz a před vstupem A/D převodníku umístěn antialiasingový filtr s mezním kmitočtem 2821,9 Hz. Tato vzorkovací frekvence může být použita, protože systém nezkoumá řečový signál jako takový, ale získává z něho pouze jeho otisky pomocí

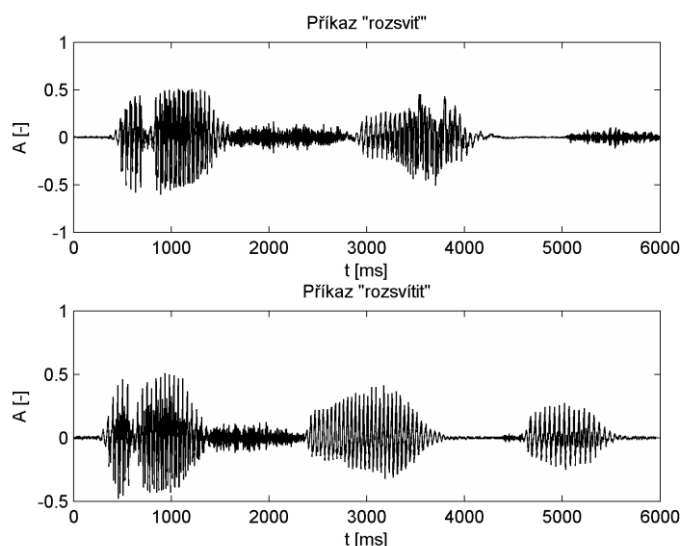
digitální filtrace.

Filtr	Řád/Jazyk	2	3	4
FIR	GCC	425	498	578
	GCC a JSA	236	255	274
	JSA	216	239	258
IIR	GCC	714	891	1041
	GCC a JSA	427	469	494
	JSA	357	397	433
IIR kanoická	GCC	512	677	842
	GCC a JSA	296	332	368
	JSA	225	260	295

Tab. 1 Přehled počtu strojových cyklů v závislosti na řádu filtru a použitého programovacího jazyka (převzato z [8])

2.2 Délka zpracovávané sekvence

Zadaný procesor není schopen porovnávat signál delší jak půl vteřiny a to kvůli náročnosti na rychlost výpočtu a na paměťovou kapacitu dat. Tento problém však není kritický, protože kořen slova je obsažen na začátku. Z tohoto důvodu je možno částečně tolerovat i rozdíly ve skloňování a časování slov (např. rozsvít – rozsvítit). Tato výhoda je zobrazena na obr. 5.



Obr. 5 Porovnání slova "rozsvítit" a "rozsvít"

Velikost zpracovávané sekvence se v programu volí pomocí proměnné `segcnt`. Hodnota může nabývat velikosti 1 až 32, kde 32 odpovídá vteřině. Tato velikost je dána použitím vzorkovací frekvence 4 kHz, tedy 4000 vzorků za sekundu, a použitého okna o velikosti 125 vzorků. Z této závislosti je dána hodnota 32 (4000/125).

Aby nahrávání slova začalo vždy ve správný okamžik, mikroprocesor porovnává klouzavý průměr hluku v okolí a pokud dojde signál požadované úrovně, začne nahrávat.

2.3 Počet rozeznatelných příkazů

Počet rozeznatelných příkazů je omezen paměťovou kapacitou procesoru, mnohem více než výpočetní rychlost porovnávání. Každé slovo (příkaz) je uloženo ve slovníku `dic [počet slov][počet hodnot pro jedno slovo]`. Počet hodnot pro jedno slovo je dán počtem filtrů násobeno délkou sekvence, kde hodnota 32 odpovídá jedné vteřině. Jestliže je použito 8 filtrů a nahráváno je půl vteřiny, výsledná velikost jednoho slova je $8 \cdot 16 = 128$. Výsledné hodnoty jsou dány odezvou filtrů, tedy desetinná čísla. Podle tab. 2 (převzato z [9]) by při použití proměnných typu float byla velikost jednoho slova $128 \cdot 4 = 512$ bajtů.

Typ	Rozsah hodnot	Počet bajtů
char	-128 až 127	1
int	-32768 až 32767	2
float	-3,4E-38 až 3,4E38	4
double	-1,7E-308 až 1,7E308	8

Tab. 2 Přehled velikosti proměnných (převzato z [9])

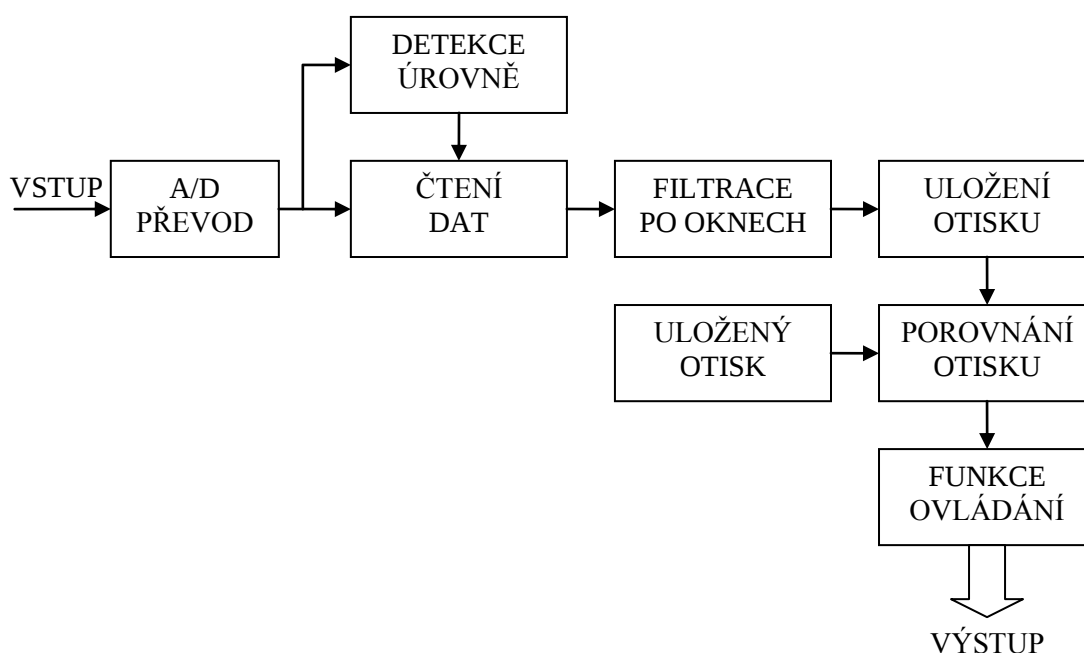
Toto řešení není možné již z důvodu velkého obsazení paměti a početní náročnosti s použitím proměnné typu float. Z těchto důvodů jsou použity proměnné typu integer, které ušetří polovinu paměťové kapacity a např. násobení trvá pouhé dva strojové cykly. Pro přepočítání z desetinného čísla na celé je hodnota vynásobena 2^N , kde N je rovno 8 (osmi bitové slovo), tedy hodnotou 256.

Pro tuto práci byl použit mikrokontrolér ATmega32, který obsahuje 2 kB SRAM. Program beze slov v datové paměti má velikost 788 bajtů (38,5%). Jedno slovo v datové paměti zabírá 256 bajtů, tzn. že do paměťové kapacity procesoru, s použitím osmi filtrů, je možno uložit 4,92 slov.

V kapitole 4.3.1 je uvedena závislost úspěšnosti rozpoznávání příkazů na jejich počtu a optimalizace velikosti otisků.

3 ALGORITMUS NA ZÍSKÁNÍ OTISKU PŘÍKAZU

Celý algoritmus je psán v jazyce C v softwaru firmy Atmel AVR studio a přeložen vestavěným překladačem GCC. Algoritmus využívá interních prvků mikrokontroléru jako je například A/D převodník, generátor PWM apod. Blokové schéma algoritmu je naznačeno na obr. 6.



Obr. 6 Blokové schéma algoritmu pro rozeznání příkazu

Na vstup je přiváděn zesílený a frekvenčně omezený signál z mikrofону v rozmezí 0 až 5V. Analogově-digitální 10bitový převodník převede spojitý signál se vzorkovací frekvencí 4 kHz na číslicový, odpovídající hodnotám 0 až 1023. Následuje funkce, která je spuštěna jen jednou, měření hluku v místnosti, kde je získána úroveň prahu hluku. Po detekci užitečného signálu, který je dán hodnotou větší jak práh hluku, je spuštěna filtrace. Digitální filtrace je realizována po oknech o velikosti 125 vzorků, 8 IIR filtrů a délky sekvenec 0,5 vteřiny. Jakmile je zpracováno 2000 vzorků, filtrace je ukončena a je získán otisk o rozměru 128 hodnot. Nově vytvořený otisk je porovnáván s otiskem uloženým ve slovníku pomocí porovnávací funkce Manhattanské vzdálenosti. Po ověření shody, která je dána celým číslem odpovídajícím příkazu, je spuštěna funkce ovládání, která řídí zařízení např. pomocí PWM. Výstupní signál může ovládat výkonový prvek, který zajišťuje funkce například typu zapnout, vypnout, změna výkonu apod.

3.1 Nastavení přerušení časovače/čítače a A/D převodu

Vstupní signál je snímán se vzorkovací frekvencí 4 kHz, a proto je každých 250 μ s získán nový vzorek. Aby snímání bylo uskutečněno v této časové prodlevě, je v programu nastaveno přerušení pomocí časovače/čítače0. Pro přerušení je nastaven v registru TCCR0 mód Output Compare Match. V tomto módu jsou porovnávány registry TCNT0 a OCR0. Registr TCNT0 inkrementuje o 1 a v případě shody s OCR0 je vynulován, čítání začíná opět od nuly. Registr OCR0 je nastaven na hodnotu 249 podle vztahu (7) převzatého z [6].

$$f_{ocn} = \frac{f_{CLK\ I/O}}{2 \cdot N \cdot (1 + OCRn)}, \quad (7)$$

kde f_{ocn} je frekvence přerušení (tzn. $1/250 \cdot 10^{-6}$), $f_{CLK\ I/O}$ značí frekvenci procesoru (použitý kmitočet je 16MHz), N udává velikost předděličky (8), $OCRn$ je registr pro porovnání nabývajících hodnot 0 až 255. Toto přerušení je povoleno lokálně pomocí registru TIMSK.

V přerušení časovače/čítače0 je zajištěna obsluha A/D převodníku, tzn. pomocí registru ADC je načten nový vzorek do paměti. Dále je zajištěn nový A/D převod nastavením logické jedničky ADSC v registru ADCSRA. Parametry A/D převodu jsou nastaveny v registru ADCSR, kde je lokálně povolen převod, a v registru ADMUX je řečeno, na jaký pin převodníku je signál přiveden. Pro převodník je stanovena předdělička 128, která odpovídá frekvenci 125 kHz. Pro vzorkovací frekvenci 4 kHz je interní převodník vhodný a dostatečně rychlý.

Jako výstupní signál je použito PWM (pulzně šířková modulace). Pro generování modulace je použit časovač/čítač2, který je 8bitový. Pro generování je opět použit mód Compare Match v registru TCCR2, kde se porovnávají registry TCNT2 a OCR2. Hodnota registru OCR2 se volí podle potřeby velikosti střidy.

3.2 Měření hluku v místnosti

Algoritmus pro měření hluku zajišťuje, aby rozpoznávání slova bylo spuštěno na začátku jeho vyslovení. Používá k tomu funkci, která zpracovává dlouhodobý průměr měřeného signálu. Jakmile je detekována náběžná hrana, je spuštěno nahrávání a následná analýza dat.

V programu jsou pro toto měření použity tři proměnné `noise1`, `noise2` a `noise3`. Do každé proměnné je načítána 256 krát hodnota z A/D převodníku. Následně je vybrána proměnná s nejvyšší hodnotou a uložena do proměnné `threshold`. Pro detekci právě vysloveného slova je třeba hodnotu uloženou v `threshold` vynásobit 2 až 4 krát, aby byl rozpoznán začátek slova. Z experimentálních poznatků je tato hodnota násobena 4 krát.

Hodnota v `threshold` se nemusí dělit 256, protože je porovnávána s hodnotou z A/D převodu násobenou rovněž 256. Obě hodnoty jsou proměnné typu `integer`, které tím pádem ušetří paměťové místo a porovnání je rychlejší.

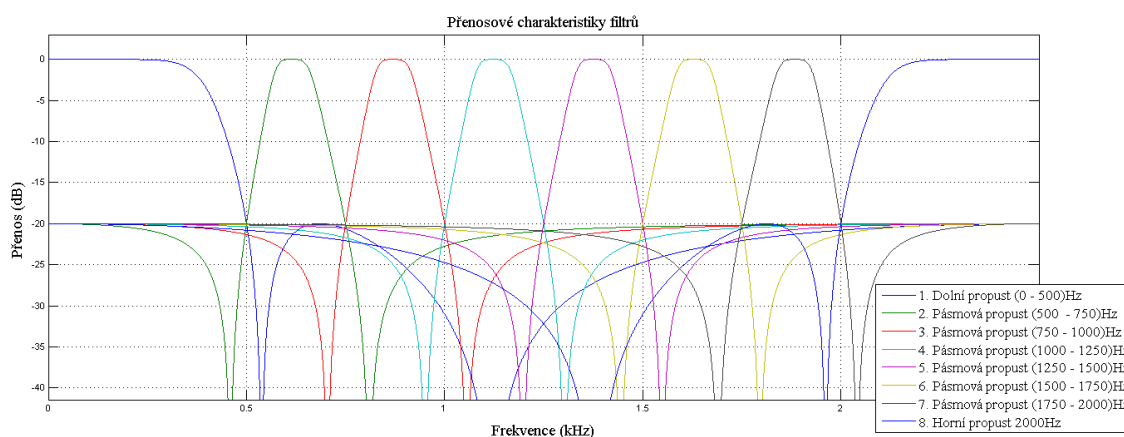
Průměrná velikost prahu hluku byla změřena 43 tedy 0,21V. Tímto hodnota ADC, která má velikost 1024 (16bitů), ztrácí 6bitů, které reprezentují velikost hluku.

3.3 Funkce filtrů

Jedna z nejdůležitějších funkcí algoritmu je získání otisku. Pro získání otisku je možno použít dva způsoby a to buď rychlou Fourierovu transformaci, nebo bankou filtrů. Pro 8bitový procesor byl vybrán způsob banky filtrů.

3.3.1 Volba filtrů

Zesílený vstupní signál je frekvenčně omezený charakteristikou mikrofону od 30 Hz do 16 kHz. Z tohoto pásma je nutno využívat pouze informaci o slově, proto je signál pomocí dolní propusti omezen na 2821,9 Hz. Tato dolní propust je realizována pomocí analogového filtru před vstupem A/D převodníku. Využívaný signál v algoritmu má frekvenční pásmo od 30 Hz do 2821,9 Hz. Pro získání otisku je nutno toto pásmo rozdělit a digitálně filtrovat. Frekvenční rozdělení na subpásma je zobrazeno na obr. 7.



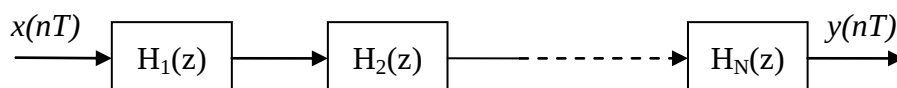
Obr. 7 Přechodové charakteristiky filtrů

Je možno volit pouze omezený počet filtrů. Tento počet je omezen paměťovou kapacitou procesoru, kde se ukládají systémové koeficienty, vstupní a výstupní zpožděné prvky a výstupní odezvy filtrů, a dále rychlostí výpočtu, která je omezena maximálním počtem 4000 strojových cyklů. Kompenzací těchto vlastností byl zjištěn ideální počet filtrů. Proto je pro tuto práci použito následujících osm filtrů:

1. Dolní propust s mezním kmitočtem 500 Hz
2. Pásmová propust s propustným pásmem 500 až 750 Hz
3. Pásmová propust s propustným pásmem 750 až 1000 Hz
4. Pásmová propust s propustným pásmem 1000 až 1250 Hz
5. Pásmová propust s propustným pásmem 1250 až 1500 Hz
6. Pásmová propust s propustným pásmem 1500 až 1750 Hz
7. Pásmová propust s propustným pásmem 1750 až 2000 Hz
8. Horní propust s mezním kmitočtem 2000 Hz

Dále je nutno vybrat typ algoritmu filtrů. Pro dolní a horní propust je možno filtr realizovat jako FIR, kdy návrh filtru je podstatně jednodušší a filtr je vždy stabilní. Naopak u pásmových propustí je lépe volit filtry typu IIR, kdy výpočet trvá méně než u filtrů typu FIR. Po několika pokusech implementovat různé typy filtrů, bylo nakonec všech osm filtrů realizováno typem algoritmu IIR. Jestliže jsou všechny filtry stejného typu, lze je napsat do jedné funkce.

Dalším parametrem je volba aproximace filtrů. Typ volby aproximace určuje také zkreslení v propustné i nepropustné části. Pro velkou strmou přechodu mezi pásmy byla vybrána Čebyševova aproximace typu II. Tato aproximace má v nepropustné části zvlnění, které nemá na tento algoritmus výrazný vliv. Strmějšího přechodu je možno dosáhnout zvolením vyššího řádu filtru. Volba vyššího řádu filtru znamená složitější realizaci, tzn. více příkazů resp. více strojových cyklů. Z důvodu vyšší strmosti a menšího zvlnění byly použity filtry řádu 4. Složitá realizace jde obejít výhodou IIR filtrů. Tyto filtry je možno zapojovat kaskádně a tím dosáhnout vyšších řádů pomocí nižších. Kaskádní realizace IIR filtrů je zobrazena na obr. 8 (převzatého z [4]).



Obr. 8 Kaskádní realizace IIR filtrů (převzato z [4])

Této výhody je využito v této práci, konkrétně sériové zapojení dvou filtrů 2. řádu. Použito je kanonické formy (Direct Form II Transposed). Diferenční rovnice použité formy je [1]:

$$y_1(n) = b_{11}x_1 + b_{12}x_1(n-1) + b_{13}x_1(n-2) - a_{11}y_1(n-1) - a_{12}y_1(n-2), \quad (8)$$

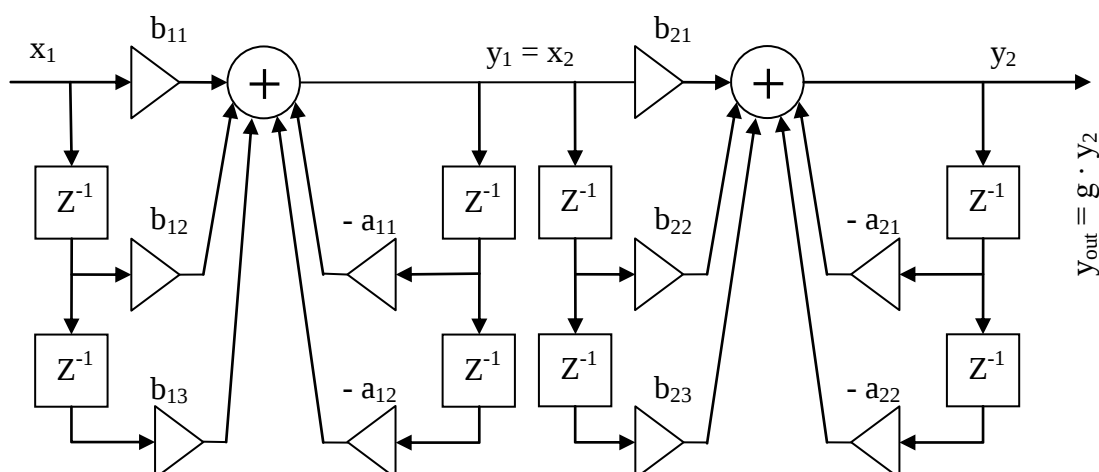
kde b_{11} , b_{12} , b_{13} , a_{11} , a_{12} popisují vlastnosti filtru, x_1 značí vstupní sekvenci hodnot. Pro druhý filtr zapojený kaskádně, platí stejná diferenční rovnice jen s jinými parametry filtru, kde x_2 je rovno y_1 [1]:

$$y_2(n) = b_{21}x_2 + b_{22}x_2(n-1) + b_{23}x_2(n-2) - a_{21}y_2(n-1) - a_{22}y_2(n-2), \quad (9)$$

výsledek diferenčních rovnic y_{out} je na závěr zesílen hodnotou g [1]:

$$y_{out} = gy_2, \quad (10)$$

Na obr. 9 je zobrazena konkrétní realizace filtru v algoritmu, kterou popisují diferenční rovnice (8), (9), (10).



Obr. 9 Realizace filtru v algoritmu

3.3.2 Získání parametrů filtrů

Pro realizaci IIR filtrů je použito kanonické formy, jejíž realizace a diferenční rovnice jsou na předchozí straně. Pro výpočet systémových koeficientů (a_{11} až a_{n2} a b_{11} až b_{n3}) je použito počítačového programu Matlab. Matlab je interaktivní systém pro vědecké a technické výpočty založený na maticovém počtu.

Matlab obsahuje nástroj TDATool, kterým lze navrhnout a analyzovat filtry. Výpočet lze provést buď uvedeným nástrojem, nebo obsahuje funkce pro výpočet filtrů. Pro výpočet systémových koeficientů byly použity následující funkce:

Pro dolní propust:

$$[B, A] = \text{cheby2}(N, R, Wst, 'low')$$

Pro horní propust:

$$[B, A] = \text{cheby2}(N, R, Wst, 'high')$$

Pro pásmovou propust:

$$[B, A] = \text{cheby2}(N, R, [Freq1, Freq2])$$

kde *cheby2* značí použití Čebyševovy aproximace typu II, N řád filtru, R velikost zvlnění v nepropustném pásmu v decibelech a Wst , $Freq1$, $Freq2$ značí mezní frekvence filtru.

Pro získání parametrů filtru 4. řádu, který vznikne zapojením kaskádně dvou filtrů, je použita funkce:

$$[SOS, G] = \text{tf2sos}(B, A, 'up', 'inf')$$

kde B , A jsou systémové koeficienty, 'up' značí vlastnosti rozložení pólů a 'inf' určuje měřítko zesílení. Výstupními hodnotami jsou zesílení G a nové systémové koeficienty uložené v následující matici:

$$SOS = \begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \dots & & & & & \\ b_{0n} & b_{1n} & b_{2n} & 1 & a_{1n} & a_{2n} \end{bmatrix}$$

Bližší informace o funkcích jsou k dispozici v nápovědě programu.

3.4 Vytvoření otisku

Vytvoření otisku lze rozdělit do dvou skupin, a to na otisk uložený v paměti mikrokontroléru, který vznikl pomocí softwaru v PC, a na otisk, který vzniká téměř v reálném čase vyslovení slova.

3.4.1 Vytvoření otisku v PC

Tento otisk je vytvářen v počítači pomocí programů pro nahrávání z mikrofону a Matlabu. Pro nahrání slova byl použit program „Záznam zvuku“ implementovaný ve Windows. Pro správnou funkčnost je třeba nastavit tyto parametry: 8bitovou hloubku, nejnižší možnou vzorkovací frekvenci 8kHz a mono nahrávku. Vyslovený příkaz by měl být dlouhý půl vteřiny a uložený ve formátu .wav. Vytvořený soubor je dále otevřen v programu Matlab pomocí funkce:

$$[Y] = \text{wavread}('soubor', N)$$

kde N je počet prvních vzorků, které mají být uloženy, 'soubor' značí cestu a název souboru a Y je výstupní vektor, který obsahuje hodnoty o velikosti -1 až 1. Pro kompenzaci zesílení a offsetu mikrofónu následuje tento postup:

1. Vynásobení vektoru hodnotou 2,5.
2. Přičtení hodnoty 2,5 k vektoru (tímto má signál velikost 0 až 5V).
3. Pro získání 10bitových hodnot je nutno použít vztah [6]:

$$ADC = \frac{V_{in}}{V_{ref}}(N-1), \quad (11)$$

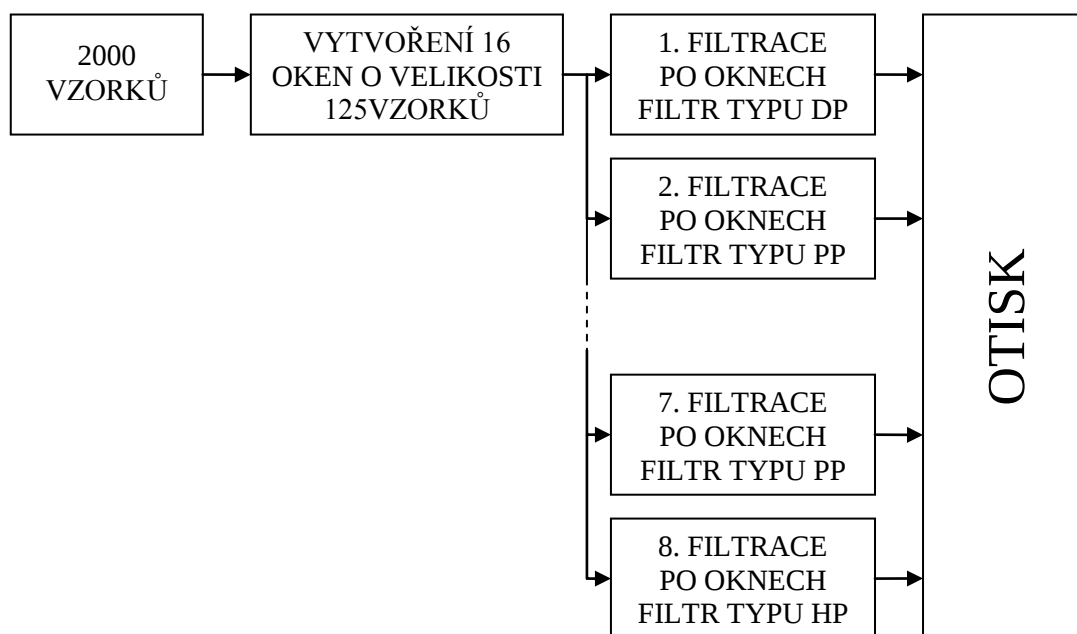
kde V_{in} je vstupní vektor, V_{ref} odpovídá hodnotě 5V a N hodnotě 1024 (10bitový A/D převodník).

4. Odečtení hodnoty 511 tzn. offsetu zesilovače.

Po tomto postupu je získáno 2000 10bitových hodnot, na které je třeba použít osm filtrů. 2000 vzorků se rozdělí do 16 oken po 125 vzorcích. Pro získání odezvy je použito diferenčních rovnic (8), (9), (10). Po odezvě filtrů jsou hodnoty nahrány do paměti dat do proměnné DIC o velikosti 128 vzorků.

3.4.2 Vytvoření otisku v algoritmu

Tento otisk vzniká při vyslovení příkazu v mikrokontroléru. Upravený signál z mikrofonu je zpracováván v reálném čase se vzorkovací frekvencí 4 kHz. Vzorky jsou tedy získávány použitím přerušení každých 250 μ s. Nový vzorek je porovnán s prahem hluku, tzn. detekci nástupní hrany vysloveného příkazu. Pokud je slovo analyzováno je zahájeno rozpoznávání. Od příchozího vzorku je odečtena hodnota 511, která odpovídá offsetu 2,5V vzniklého v obvodu zesilovače. Po této úpravě je spuštěna filtrace, která je prováděna po oknech velikosti 125 vzorků. Těchto oken je použito 16, což odpovídá 2000 vzorkům. Odezvy jednotlivých oken jsou průběžně ukládány v proměnné `word[počet oken][počet filtrů]`. Po dokončení filtrace je získaný otisk uložen do proměnné `fingerprint[počet oken · počet filtrů]`. Takto získaný otisk je připraven k porovnávání s otiskem vytvořeným v PC. Celý proces vytvoření otisku je zobrazen na obr. 10.



Obr. 10 Vytvoření otisku v algoritmu

3.5 Porovnávací funkce

Porovnávací funkce v algoritmu zajišťuje rozpoznání příkazu. Pro rozeznání příkazu je porovnávána proměnná `fingerprint[počet oken · počet filtrů]` s každým slovem uloženým v proměnné `dic[počet slov][počet hodnot pro jedno slovo]`. Výstupem je celé číslo, které odpovídá určitému příkazu. S již rozeznáním příkazem je možno pomocí funkce `control` řídit vstupně-výstupní jednotky

mikrokontroléru.

Porovnávací funkce jde realizovat použitím různých typů metrik, konvoluce apod. Pro tuto práci byly testovány tři funkce, jejichž popis je uveden níže.

3.5.1 Korelace

Korelační koeficient uvádí míru vztahu mezi dvěma signály. Pro výpočet se používá Pearsonova korelačního koeficientu (r), který nabývá hodnot v intervalu $[-1, 1]$, jenž je zobrazen v následujícím vztahu [10]:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (12)$$

kde x_i, y_i jsou vstupní hodnoty obou otisků a $\bar{x}, \bar{y}$ jsou výběrové průměry. Výsledný koeficient je dán absolutní hodnotou. Jestliže je koeficient roven jedné jde o dva naprosto stejné signály.

Tato metoda byla použita s Manhattanskou vzdáleností pro zlepšení výsledku rozeznávání. Z důvodu úspory paměťové kapacity je použita pouze metoda Manhattanské vzdálenosti.

3.5.2 Euklidovská vzdálenost

Nejpoužívanější metrikou je Euklidovská vzdálenost. Je obdobná Pythagorově větě. Velikost vektorů otisků musí být stejná. Pro výpočet je využito vztahu [11]:

$$D = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}, \quad (13)$$

kde x_i značí hodnoty prvního otisku a y_i hodnoty druhého otisku.

Této metody nebylo využito z důvodů použití složitosti mocninného výpočtu. Výsledná hodnota není velikostně omezena.

3.5.3 Manhattanská vzdálenost

Též nazývaná newyorská metrika. Původní systém byl založen na výpočet nejkratší vzdálenosti křižovatek Manhattanských ulic. Metoda využívá vztahu [11]:

$$D = \sum_{i=1}^n |y_i - x_i|, \quad (14)$$

kde x_i značí hodnoty prvního otisku a y_i hodnoty druhého otisku.

Manhattanská metrika je využita v algoritmu. Tato metoda není velikostně omezena. Její výhody jsou jednoduchost a rychlost. Výsledkem je nejmenší hodnota. Aby rozeznáný příkaz byl vyhodnocen pouze ze slovníku otisků příkazů, je nutno zadat meze. Tyto meze byly experimentálně získány a odpovídají hodnotě 10.

3.6 Vložení nahrávací funkce

Algoritmus je schopen se naučit několik příkazů, které bude následně bezchybně rozpoznávat. Implementování nahrávací funkce bylo řešeno vnějším přerušením pomocí tlačítka. Zařízení reaguje na náběžnou hranu stisknutí tlačítka. V tomto přerušení je volána funkce `record()`.

Funkce `record` obsahuje podobné příkazy, které jsou použity při vytváření otisku pro rozpoznávání. Na začátku i konci funkce jsou nulovány výstupní hodnoty a zpožděné členy filtrů, tzn. pole `yy[8]` a `IO[8][8]`. Nové vzorky signálu jsou získány z A/D převodníku pomocí přerušení časovače/čítače0 a porovnávány s prahovou hodnotou uloženou v proměnné `threshold` reprezentující začátek slova. Dále následuje digitální filtrace po 16 oknech velikosti 125 vzorků. Jednotlivé odezvy oken jsou ukládány v dvourozměrném poli `word`. Po ukončení filtrace je vytvořený otisk uložen přímo do slovníku, který je reprezentován proměnnou `dic`. Po návratu z funkce může být nově vzniklý otisk ihned rozpoznán.

3.7 Ovládací funkce

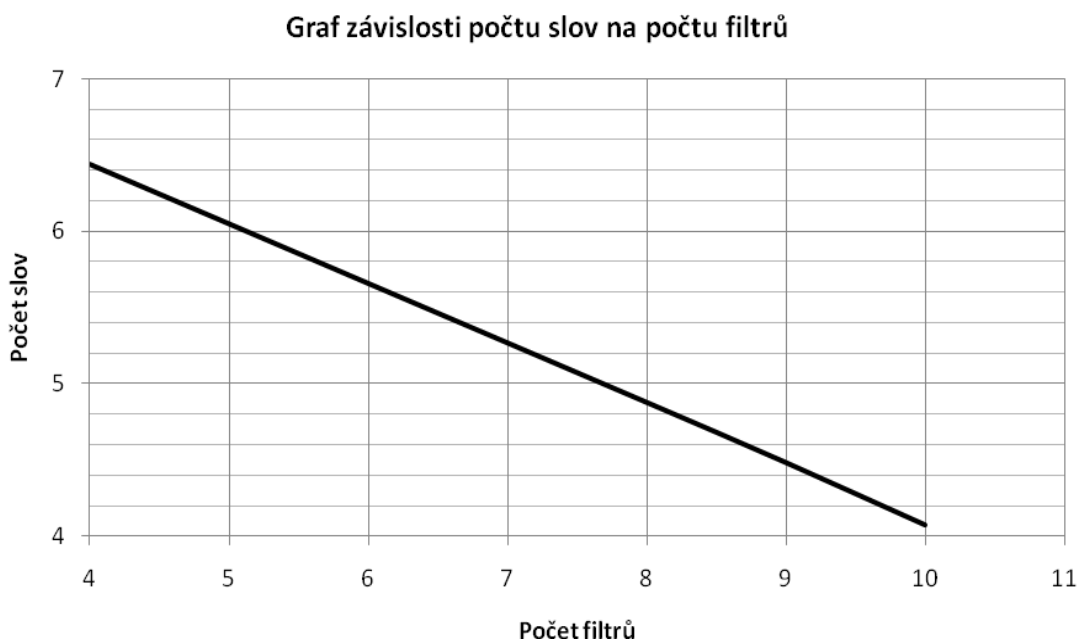
Ovládací funkce jsou funkce, které po rozpoznání příkazu, ovládají resp. řídí výstupy, které jsou od daného zařízení požadovány. Například od hlasového ovládání stmívače světla jsou očekávány následující příkazy: rozsvítit, zhasnout, zesílit případně snížit intenzitu světla. Tyto příkazy jsou realizovány pomocí pulzně šířkové modulaci, které řídí výkonový prvek např. tyristor, triak nebo tranzistor. V této práci je vyveden pin generátoru PWM pro připojení výkonového prvku, případně LED diody, která je použita pro testování.

4 OPTIMALIZACE ALGORITMU

Po vytvoření základního algoritmu pro rozeznávání řečových příkazů se program testoval. Testy probíhaly na desce nepájivého pole s mikrokontrolérem ATmega32. Algoritmus vznikl volbou parametrů a jejich následné optimalizace. Tato optimalizace je kompromisem pro použitou platformu a nejsou pevně dány. Mnoho hodnot bylo získáno experimentálně. Předchozí text popisuje algoritmus po optimalizaci, tzn. určitý počet filtrů, velikost sekvence, okna, vzorkovací frekvence apod.

4.1 Počet filtrů

Optimalizace počtu filtrů spočívá v kompromisu použití kapacity paměti a rychlosti výpočtu. Paměťová kapacita procesoru je dána typem mikrokontroléru a velikosti její datové paměti SRAM, která je nejvíce zatěžována. Výběr typu filtrů je naznačen v kapitole 3.3.1. Použitím formy Direct Form II Transposed se dvojnásobně zmenší nároky na paměťový prostor mikrokontroléru. Díky této formě může být použito více slov (příkazů). Grafická závislost ukazující poměr mezi počtem slov a počtem filtrů je zobrazena na obr. 11.



Obr. 11 Grafická závislost počtu slov na počtech filtrů při stejné velikosti otisku

Počet filtrů rovněž ovlivňuje velikost otisku, která je dána počtem použitých oken a právě počtem filtrů. Při volbě krajního počtu mezi bylo zvoleno 6 až 10 filtrů, které udávají velikost otisku 96 a 160. Po testování těchto velikostí bylo zvoleno 8 filtrů, které odpovídají otisku o rozměru 128.

Díky zvolené vzorkovací frekvenci 4 kHz a výpočetní frekvenci procesoru 16MHz je filtrace omezena podle vztahu (6) na 4000 strojových cyklů. V tab. 3 je rozepsán počet strojových cyklů každého filtru.

Počet filtrů	Typ filtru	Počet cyklů
1	Dolní propust	380
6	Pásmová propust	2280
1	Horní propust	380
Celkový počet strojových cyklů		3040

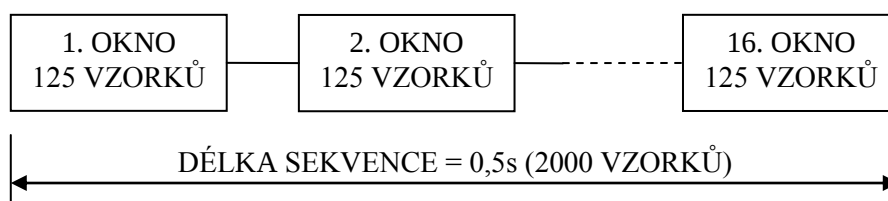
Tab. 3 Počet strojových cyklů funkce filtry

Výsledná hodnota celkového počtu strojových cyklů je menší než 4000 a velikost otisku příkazu je optimální. Z toho důvodu je pro práci použito osm filtrů.

4.2 Velikost okna a délka sekvence

Podstatnou součástí algoritmu je volba velikosti okna a délka zpracovávané sekvence. Tyto dva parametry spolu úzce souvisí. Nejprve je nutno stanovit délku sekvence. Tato délka je podle kapitoly 2.2 stanovena na půl sekundy. Podle vzorkovací frekvence 4000 vzorků za sekundu je pro jeden příkaz zpracováváno 2000 vzorků.

Velikost jednotlivých oken je přímo závislá na jejich počtu. Tento počet vznikne podílem celkového počtu vzorků. Tato hodnota musí být celé číslo, tzn. beze zbytku dělitelné. Hodnoty, které mohou být použity, jsou tedy 10, 16, 20. Počet použitých oken udává velikost otisku příkazu. Při použití hodnoty 10 je tato velikost vektoru malá, naopak při 20 je velká. Optimální hodnota počtu oken je tedy 16. Po podílu těchto hodnot je velikost okna 125 vzorků. Rozdělení délky sekvence na jednotlivá okna je zobrazeno na obr. 12.



Obr. 12 Rozdělení délky sekvence na jednotlivá okna

Po stanovení počtu filtrů a počtu použitých oken je výsledná velikost vektoru otisku rovna 128.

4.3 Úspěšnost získávání otisku příkazu

Výsledkem celé práce je správné rozpoznání řečových příkazů. Získání úspěšného rozpoznání je možno rozdělit na hardwarovou (kapitola 5) a softwarovou část. Do hardwarové části patří výběr mikrofону a zesilovacího bloku, tzn. vznik a úprava signálu do požadované podoby. Špatná volba součástek může způsobit zkreslení příp. ořez signálu. Do softwarové části patří popisovaný algoritmus, tzn. správná volba vzorkovacího kmitočtu, metody rozpoznávání tedy volbu filtrů, porovnávací metody apod. Významnou roli hraje počet a výběr použitých slov, vzdálenost od mikrofónu, vnější zdroje hluku, vada řeči (není v textu uvažována).

4.3.1 Počet a výběr slov

Vytvořený algoritmu je vložen do zařízení, které řídí jiné zařízení, případně spouští resp. ovládá jiné funkce. Pro každé zařízení je nutno použít odlišný počet příkazů. Těchto příkazů nemůže být neomezený počet, je proto vhodné používat hlasových příkazů jen pro nejpoužívanější funkce. Počet slov je opět dán paměťovou kapacitou procesoru. Podle grafické závislosti na obr. 11 závisí na počtu filtrů. Pro použitých 8 filtrů je maximální počet slov 4. V datovém paměťovém prostoru je jedno slovo reprezentováno velikostí 256 bajtů.

Pro správné rozpoznávání příkazů není vhodné využívat maximálního počtu slov, kterých je možno do paměťového prostoru vložit. Čím více slov (příkazů) je použito, tím klesá úspěšnost rozpoznávání. Tabulky 4 a 5 tuto závislost zobrazují. Každý příkaz byl analyzován 20 krát a to po každé se stejnými podmínkami. Vyslovený příkaz byl nahrán a uložen v PC. Před zařízením s vestavěným algoritmem byl umístěn reproduktor ve vzdálenosti 20cm a příkazy byly spouštěny se stejnou úrovní. Před každým měřením bylo zařízení restartováno. Bylo použito dvojce měření. První bylo realizováno v uzavřené místnosti, tzn. bez vnějšího zdroje hluku. Druhé bylo realizováno při pootevřeném okně a při spuštěném televizoru, tzn. s využitím vnějšího zdroje hluku. V tab. 5 byl při měření do mikrokontroléru nahrán maximální počet slov, resp. otisků. Chybovost je znatelná, ale při volbě vhodných slov je možnost chybovost regulovat. Při měření jejíž výsledek je uveden v tab. 4, byly využity čtyři příkazy a jejich úspěšnost je téměř stoprocentní. I v případě použití vnějšího zdroje hluku je úspěšnost dostačující, kde v nejhorším případě je z 20 příkazů rozpoznáno 17.

Příkaz/ Úspěšnost [%]	Bez zdroje hluku	Se zdrojem hluku
Zhasni	80	70
Rozsviť	90	85
Zapnout	85	80
Vypnout	100	90
Světlo	95	85
Zesílit	50	25
Zeslabit	55	30

Tab. 4 Úspěšnost rozpoznávání osmi příkazů při použití 5 filtrů

Příkaz/ Úspěšnost [%]	Bez zdroje hluku	Se zdrojem hluku
Vypnout	100	95
Zapnout	100	95
Více	95	85
Méně	100	90

Tab. 5 Úspěšnost rozpoznávání čtyř příkazů při použití 8 filtrů

Důležitou volbou je i výběr slov. Příkazy obsahující kořen slova na začátku mohou pak splývat v jeden. Tato chyba nastává i při nevhodně vybraných slovech, kde je použita i stejná předpona, viz slova v tab. 4 zesílit a zeslabit. Nízká úspěšnost byla dána podobností slov i vysoký počet použitých příkazů.

4.3.2 Vliv vzdálenosti od mikrofonu

Dalším kritériem pro úspěšné rozpoznání příkazů je vliv vzdálenosti od mikrofonu. Tento vliv závisí na použité směrové charakteristice mikrofonu. V zařízení je použit elektrodynamický mikrofon kulové charakteristiky, tzn. signál je přijímán ze všech stran se stejnou intenzitou.

Pro měření bylo do slovníku uloženo 4 slov (příkazů) a bylo prováděno 20 krát s každým slovem. Příkaz byl nahrán a uložen do PC. Zařízení s algoritmem bylo umístěno na podlahu a pomocí metru byl do vzdálenosti 20cm umístěn reproduktor. Vzdálenost reproduktoru od zařízení byla měřena až do 2m. V tabulce 6 a 7 je zobrazena závislost vlivu vzdálenosti mikrofonu. Rovněž byly použity pro každé měření stejné podmínky.

V tab. 6 bylo měření prováděno v uzavřené místnosti bez zdroje vnějšího hluku. Pro vzdálenost 20cm je úspěšnost stoprocentní. Opačného výsledku bylo dosaženo při měření, jejíž hodnoty jsou uvedeny v tab. 7. Zde bylo použito vnějšího zdroje hluku pootevřením okna a zapnutého televizoru. Vnější zdroj hluku je znatelný. Z měření vyplývá, že s rostoucí vzdáleností úspěšnost rozeznávání klesá. Při podmínkách bez vnějšího zdroje však není kritická. Tuto vlastnost lze využít jako výhodu, kdy využíváme v různých místnostech stejné zařízení. Při vyslovení slova v jedné místnosti nebude ve vedlejší místnosti příkaz rozpoznán.

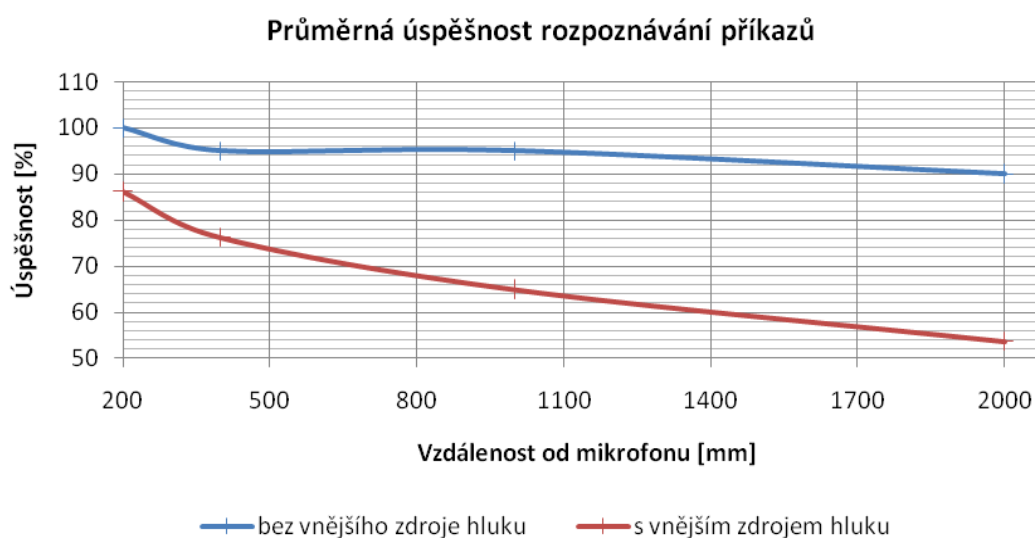
Příkaz/ Vzdálenost [mm]	200	400	1000	2000
Světlo	100	90	95	90
Tma	100	100	100	90
Méně	100	95	90	90
Více	100	95	95	90
Průměrná hodnota	100	95	95	90

Tab. 6 Úspěšnost rozpoznávání příkazů v závislosti na vzdálenosti od mikrofonu bez zdroje hluku

Příkaz/ Vzdálenost [mm]	200	400	1000	2000
Světlo	85	80	65	50
Tma	95	80	70	60
Méně	80	70	65	55
Více	85	75	60	50
Průměrná hodnota	86,25	76,25	65	53,75

Tab. 7 Úspěšnost rozpoznávání příkazů v závislosti na vzdálenosti od mikrofonu se zdrojem hluku

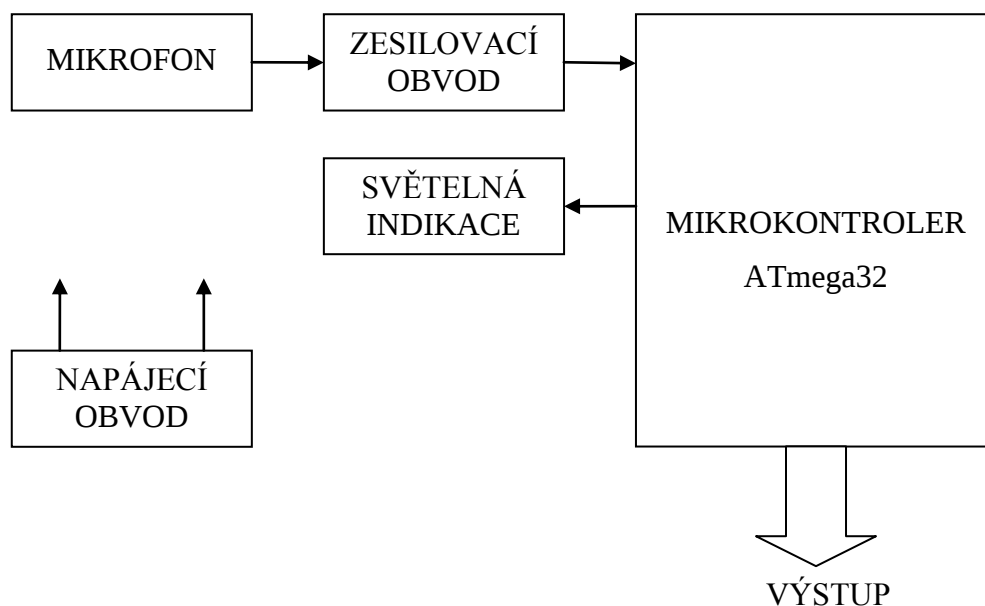
Na obr. 13 je zobrazena grafická závislost úspěšnosti rozpoznávání příkazu na vzdálenosti od mikrofonu. Při použití vnějšího zdroje hluku, jako je například zapnutý televizor či otevřené okno, klesá úspěšnost rozeznávání téměř na polovinu.



Obr. 13 Průměrná úspěšnost rozpoznávání příkazů

5 REALIZACE ZAŘÍZENÍ

Vytvořený algoritmus byl testován na desce nepájivého pole. Ovšem požadavek je aplikovat funkční algoritmus pro nějaké zařízení. Pro realizaci byla vybrána aplikace hlasového ovládání stmívače světla. Obvodové schéma je uvedeno v příloze A1 a blokové schéma celého zařízení je zobrazeno na obr. 14.



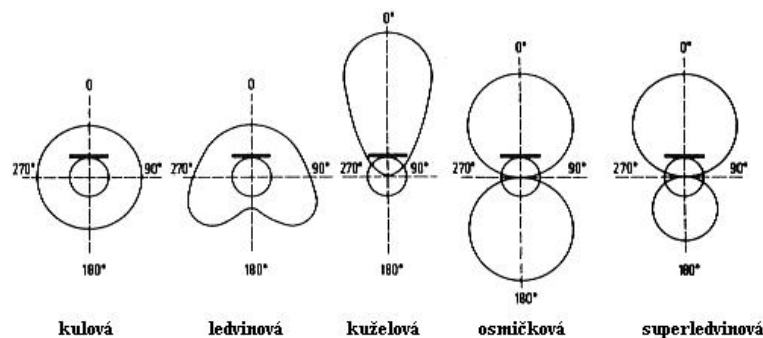
Obr. 14 Blokové schéma zařízení

Vstupní obvody jsou tvořeny bloky mikrofon a zesilovací obvod. V zesilovacím obvodu je použito tří stupňového zesilovače a dolní propusti. Jejich výstup je přiveden na A/D převodník mikrokontroléru. V mikrokontroléru po vyhodnocení příkazu je výsledek zobrazen na světelné indikaci, která je tvořena LED diodami, a také je řízen výstup. Výstup tvoří vyvedený pin generátoru PWM, na který může být připojen výkonový prvek pro řízení silnoproudé části např. výkonové řízení žárovky, nebo LED dioda pro ukázkou řízení slaboproudé části. Veškeré bloky jsou napájeny napájecím obvodem. V tomto obvodu je spínaný zdroj, který převádí střídavé napětí o velikosti 230V na stejnosměrné o velikosti 8,3V, a pevný stabilizátor na 5V. Protože se jedná o ovládání světla je výrobek zabudován do elektroinstalační krabice.

5.1 Mikrofon

Pro převod akustického signálu na elektrický je použito mikrofonu. Existuje mnoho druhů elektromechanického měniče. Pro tuto práci je zvolen elektrodynamický mikrofon. Změnou akustického tlaku je pohybováno cívkou, která je umístěna v blízkosti permanentního magnetu. Podle zákona o elektromagnetické indukci je indukováno napětí.

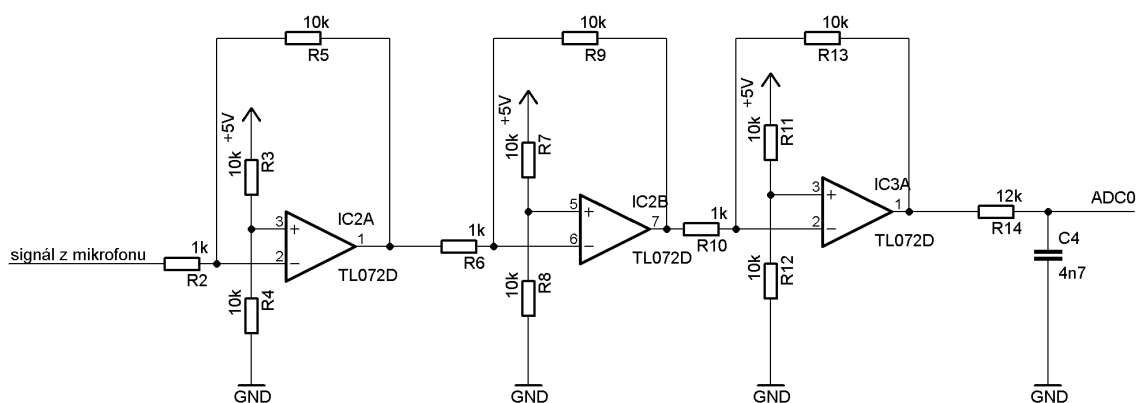
Použitý typ mikrofону je nejvíce používaný, má robustní konstrukci, malý šum a malé zkreslení. Citlivost bývá uváděna 1,5 mV/Pa, frekvenční rozsah od 30 Hz do 15 kHz. Směrovou charakteristiku má kulovou, která je pro použitou aplikaci nejvíce vhodná (snímá hlas ve všech směrech). Pro jinou aplikaci je možno využít jiných charakteristik, které jsou zobrazeny na obr. 15 (použito z [12]), např. kuželová charakteristika umožňuje přijímat signál pouze od zdroje umístěný před mikrofón.



Obr. 15 Směrové charakteristiky mikrofónů

5.2 Zesilovač

Pro úpravu signálu z mikrofónu je použito zesilovacího bloku. Výstupní hodnota z mikrofónu má velikost několika jednotek milivoltů. A/D převodník mikrokontroléru je schopen převádět signál o rozsahu 0 až 5V. Z tohoto důvodu bylo použito tři stupňového zesilovače vycházející z [13] s celkovým zesílením 1000. Jednotlivé stupně jsou tvořeny zesilovači s invertujícími zapojeními. Za posledním stupněm zesilovače je zapojen filtr typu dolní propust, který složí k ořezu vysokých frekvencí neobsahujících informaci o slově, i jako antialiasingový filtr. Dolní propust má mezní frekvenci 2821,9 Hz a pomocí Thomsonova vztahu byly vypočítány hodnoty součástek. Obvodové zapojení zesilovacího bloku je zobrazeno na obr. 16.



Obr. 16 Obvodové schéma zesilovacího bloku

Zesílení obvodu bylo stanoveno na 1000 z důvodu velikosti vstupního signálu okolo 2,5 mV. Toto zesílení bylo ověřeno na osciloskopu kvůli využití celého rozsahu. Výsledné zesílení je dáno výpočtem:

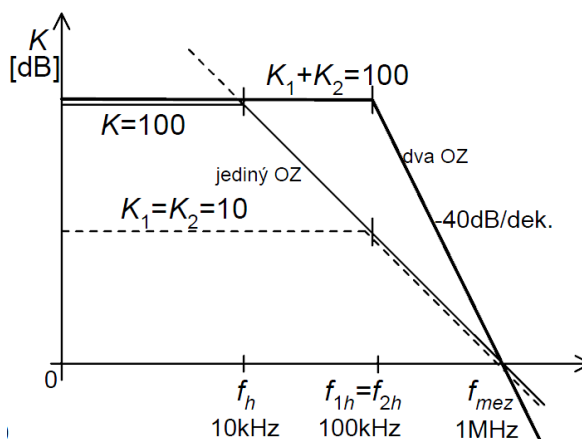
$$K = \frac{R_5}{R_2} \cdot \frac{R_9}{R_6} \cdot \frac{R_{13}}{R_{10}} = \frac{10k}{1k} \cdot \frac{10k}{1k} \cdot \frac{10k}{1k} = 1000.$$

Neinvertující vstup zesilovače bývá uzemněn. Zde je pomocí děliče napětí, tvořený dvěma rezistory o stejné velikosti, přivedeno napětí 2,5V. Tímto se plovoucí zem přenesne na hodnotu 2,5V.

Tří stupňového zesilovače je použito z důvodu velkého zesílení. Při použití jednoho stupně s operačním zesilovačem s mezním kmitočtem $f_{mez} = 50 \text{ kHz}$ a zesílením $K = 1000$ je horní mezní frekvence snížena podle vztahu převzatého z [14]:

$$f_h = \frac{f_{mez}}{K}, \quad (15)$$

na $f_h = 50 \text{ Hz}$. Po použití tří stupňů je do jmenovatele připsána třetí odmocnina a horní mezní frekvence se posune na hodnotu $f_h = 5000 \text{ Hz}$. Tato závislost popisující vliv velikosti zesílení na horním mezním kmitočtu je zobrazena na kmitočtové charakteristice na obr. 17.



Obr. 17 Kmitočtová charakteristika dvou operačních zesilovačů se stejným zesílením (převzato z [14])

5.3 Mikrokontrolér

Nejdůležitějším obvodovým prvkem je mikrokontrolér, který je jednočipový. Na jednom čipu je integrováno výpočetní jádro, paměťový prostor a periferní obvody. Výpočetní jádro obsahuje aritmeticko-logickou jednotku, která je vysoce výkonná a dokáže v krátkém čase (tzn. během několika cyklů) vykonat obtížné matematické operace. Do periferních obvodů patří A/D převodníky, časovače, čítače, komparátory, generátory PWM, vstupně výstupní porty, komunikační rozhraní a mnoho dalších. Díky nim je možno realizovat zařízení za použití velmi malého počtu elektrických součástek. S menším počtem použitých součástek vzrůstá spolehlivost celého zařízení. Mikrokontrolér je napájen malým napětím, kterým je možno řídit silnoproudé obvody, a také má nízký odběr proudu, tzn. malá spotřebu.

5.3.1 Požadavky algoritmu na mikrokontrolér

Tato práce se zabývá mikrokontroléry od firmy Atmel a je omezena pouze na 8bitové procesory. Požadavky algoritmu na mikrokontrolér jsou následující:

- Jeden kanál analogově-digitálního převodníku, který převádí zesílený signál z mikrofonu. Jeho přesnost je dána bitovou hloubkou převodu. Pro zařízení je dostačující 8bitový převodník, ale pro vyšší úspěšnost rozeznávání je nutno použít vyšší přesnosti, a to 10bitovou hloubku převodu.
- Velká datová paměťová kapacita, která je dána velikostí proměnných. Velký paměťový prostor zabírají systémové koeficienty, vstupní a výstupní zpožděné koeficienty, výstupní odezvy filtrů a slovník příkazů.
- Velká výpočetní rychlost, která je potřebná především pro výpočet odezvy filtrů. Pro splnění vzorkovací frekvence při použití funkce se 4000 strojovými cykly musí být výpočetní frekvence minimálně 16 MHz.
- Generátor PWM signálu, který umožňuje řídit výstup zařízení.

5.3.2 Použitý mikrokontrolér

S ohledem na požadavky algoritmu byl mikrokontrolér vybírán z rodiny AVR MEGA. Tyto mikrokontroléry se liší počtem periférií, velikostí paměťového prostoru, možnostmi připojení externích oscilátorů pro zvýšení výpočetního výkonu apod. Pro algoritmus je největším problémem paměťová kapacita, proto v tab. 8 jsou srovnány jednotlivé mikrokontroléry právě podle velikosti paměťového prostoru.

Mikrokontrolér	Velikost paměti programu	Velikost paměti dat
ATmega 8	8 kB	1 kB
ATmega 16	16 kB	1 kB
ATmega 32	32 kB	2 kB
ATmega 128	128 kB	4 kB

Tab. 8 Srovnání mikrokontrolérů ATmega

Pro konkrétní aplikaci byl vybrán mikrokontrolér ATmega 32. Použití čtyř příkazů a osmi filtrů nepřevyšuje velikost paměti dat 2 kB. Tento mikrokontrolér splňuje veškerá kritéria pro daný algoritmus.

Obvodové zapojení jednotlivých vývodů mikrokontroléru je zobrazeno v příloze A1. Pro uvedenou výpočetní rychlost je použito externího krystalu 16 MHz umístěného mezi vývody XTAL1, XTAL2 a použitými kondenzátory o velikosti 22 pF podle [6]. Pro výstupní generování PWM signálu je použito pinu OC2.

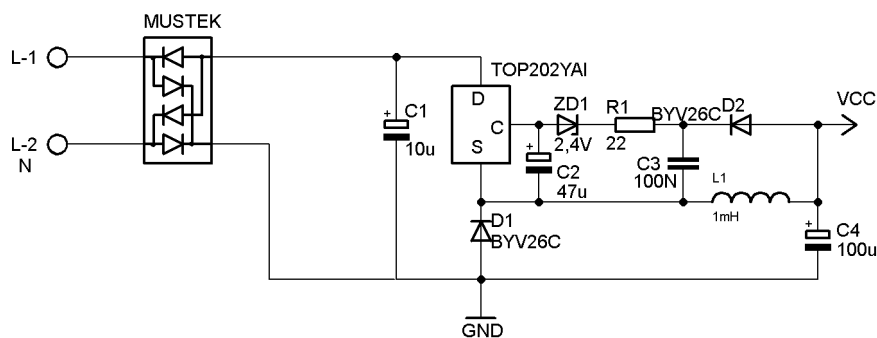
Pro programování mikrokontroléru je použito metody ISP (In System Programming). ISP využívá sériové periferní rozhraní SPI (*Serial Peripheral Interface*). Piny použité pro programování jsou MOSI, MISO, SCK a RESET. Tyto piny jsou vyvedeny pro opětovné programování mikrokontroléru.

5.4 Napájení zařízení

Každé elektrické zařízení potřebuje být napájeno, a to potřebnou úrovní napětí. Vzhledem k použité aplikaci, která je umístěna v elektroinstalační krabici, je dobré využít přivedeného střídavého napětí 230V. Proto je použito spínaného zdroje, který převádí síťové střídavé napětí na stejnosměrné bez použití transformátoru.

5.4.1 Spínaný zdroj

Díky konstrukčním podmínkám bylo požadováno, aby zdroj neměl v obvodu transformátor, který by se z důvodu velkého rozměru do krabice nevešel. Ideálním řešením je použití obvodu s TOPSwitchem. Toto zapojení není galvanicky oddělené od sítě, ale pro danou aplikaci oddělení není potřeba. Výstupní napětí je stanoveno na 8,3V, maximální proudový odběr je 200 mA.

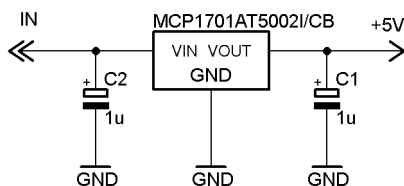


Obr. 18 Obvodové zapojení pomocného zdroje (převzato z [15])

Síťové napětí je usměrněno pomocí můstku a vyhlazeno kondenzátorem C_1 . Po sepnutí obvodu TOPSwitch vzrůstá lineárně proud na tlumivce L_1 . Po vypnutí se naakumulovaná energie uzavře v obvodu s diodou D_1 . Při dosáhnutí úrovně napětí odpovídající zenerové diodě ZD_1 je otevřen vývod Source TOPSwitchu a na cívce se znovu akumuluje energie. Výstupní napětí je dáno velikostí úbytků na diodách D_1 , D_2 , ZD_1 a napětí na rezistoru R_1 . Použité obvodové schéma je převzato z [15] a upraveno pro konkrétní zapojení.

5.4.2 Výběr stabilizátoru

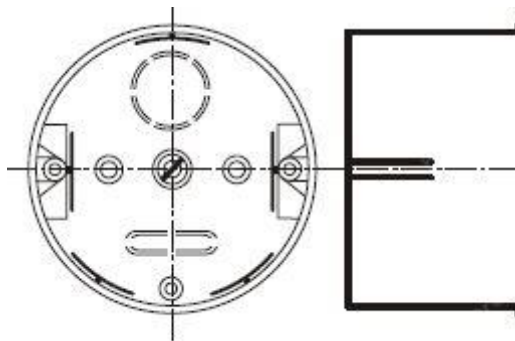
Výstupní napětí na spínaném zdroji je 8,3V. Mikrokontrolér je možné napájet napětím o velikosti od 4,5 do 5,5V. Typickým napětím podle datasheetu procesoru je 5V, které bylo vybráno. Zvolené napětí je použito i pro operační zesilovače. Pro přesný A/D převod je nutné, aby napětí bylo konstantní. Z tohoto důvodu bylo využito pevného stabilizátoru napětí MCP1701AT-5002I/CB. Tento stabilizátor vyniká velmi malými rozměry (pouzdro SOT23-3), nízkým úbytkem a velmi malou vlastní spotřebou. Maximální vstupní napětí je 10V. Obr. 19 (převzatého z [16]) zobrazuje obvodové zapojení dané datasheetem součástky.



Obr. 19 Zapojení stabilizátoru (převzato z [17])

5.5 Mechanické provedení

Požadavek na aplikaci je umístění zařízení do elektroinstalační krabice (viz obr. 20 převzatý z [17]) o průměru 68 mm. Zařízení je použito místo přístroje spínače a slouží k hlasovému ovládání světel.



Obr. 20 Elektroinstalační krabice průměru 68 mm (použito z [17])

Pro zařízení jsou vyrobeny dvě desky plošných spojů. Jedna obsahuje obvod spínaného zdroje a druhá samotný stmívač. Obě desky jsou vytvořeny ve tvaru kruhu o průměrech 50 mm (spínaný zdroj) a 66 mm (stmívač světel). Deska menšího průměru je umístěna ve spodní části krabice, deska většího průměru je umístěna výše a upevněna částí pro vruty. Fotografie mechanického provedení jsou na přiloženém CD.

6 ZÁVĚR

Bakalářská práce se zabývá algoritmem pro rozpoznávání řečových příkazů na mikrokontroléru. Rozebírá způsoby rozeznávání řeči. Nejvhodnějším způsobem je použití lineární filtrace, která je pro zadanou platformu dostatečně rychlá a tím dokáže zpracovávat více vzorků resp. delší zpracovávanou sekvenci příkazu. Nevýhodou tohoto způsobu je velká náročnost na paměťovou kapacitu mikrokontroléru. Tato nevýhoda ovšem není kritická, volbou vhodného mikrokontroléru ji lze eliminovat.

Byly definovány potřebné parametry s ohledem na cílovou platformu. Vzorkovací frekvence byla zvolena 4 kHz z důvodu potřeby použití 4000 strojových cyklů pro filtraci. Délka zpracovávané sekvence je nastavena na půl sekundy, a to díky malému výpočetnímu výkonu. Počet příkazů je dán kompromisem mezi délkou zpracovávané sekvence a počtem použitých filtrů. Výsledkem kompromisu je velikost otisku, jejíž hodnota je 128. Není vhodné volit maximální počet příkazů, protože s rostoucím počtem klesá jejich úspěšnost rozeznávání. Ideální počet byl stanoven na čtyři. Dalšími vlivy pro úspěšnost rozeznávání je vzdálenost od mikrofonu a vnějšího zdroje hluku. Při vzdalování se od mikrofonu je úspěšnost rozpoznávání nižší. Tato vzdálenost může být při vytváření otisku v PC nastavena, díky vysoké citlivosti mikrofonu. Maximální vzdálenost je však 3m. Při použití vnějšího zdroje hluku, jako je otevřené okno případně zapnutý televizor, úspěšnost rovněž klesá. S použitím menšího počtu příkazů však není vnější zdroj znatelný.

Algoritmus byl aplikován do zařízení umístěného místo přístroje spínače do elektroinstalační krabice. Zařízení je schopno ovládat základní funkce pro řízení světla.

Tato práce byla přínosem pro osvojení programovací jazyka C v mikrokontrolerech a základních znalostí se zpracováním nízkofrekvenčního signálu.

LITERATURA

- [1] Harison, Andre, Shah, Chirag. Voice Recognition Robotic Car [online] Cornell University. [cit. 21. května 2011]. Dostupný z [www: <http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/avh8_css34/avh8_css34/index.html>](http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/avh8_css34/avh8_css34/index.html)
- [2] CHASSAING, R. Digital Signal Processing and Applications with the C6713 and C6416 DSK. 1st ed. New Jersey (USA): John Wiley & Sons, 2005.
- [3] Kolka, Z., Počítačové řešení elektronických obvodů, přednáška číslo 6 [online]. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2010. 33s. [cit. 21. května 2011]. Dostupný z [www: <https://krel.feec.vutbr.cz/VYUKA/B_EST/prezencni/BREO/prednasky/P06/predn-6-filtry.pdf>](https://krel.feec.vutbr.cz/VYUKA/B_EST/prezencni/BREO/prednasky/P06/predn-6-filtry.pdf)
- [4] JAN, Jiří. Číslíková filtrace, analýza a restaurace signálů. 2. upravené a rozšířené vydání, brož. Brno: Vysoké učení technické v Brně, VUTIM 2002. 429 stran. ISBN 80-214-2911-9.
- [5] Šebesta, V., Smékal, Z., Signály a soustavy [online]. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2004. 167s. [cit. 21. května 2011]. Dostupný z [www: <https://www.vutbr.cz/www_base/priloha.php?dpid=21228>](https://www.vutbr.cz/www_base/priloha.php?dpid=21228)
- [6] ATmega32 [online]. ATMEL datasheet [cit. 21. května 2011]. Dostupný z [www: <http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf>](http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf)
- [7] Xiaowen, Lu, Shihjia, Lee. Voice recognition security systém [online] Cornell University. [cit. 21. května 2011]. Dostupný z [www: <http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/XL76_SL362/XL76%20SL362/index.html>](http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/XL76_SL362/XL76%20SL362/index.html)
- [8] ZÁPLATA, F., Digitální filtrace na 8-bitových procesorech: bakalářská práce. Brno: FEKT VUT v Brně, 2009. 35 s., 3 příl.
- [9] MANN, Burkhard. C pro mikrokontroléry. [s.l.] : [s.n.], 2003. 280 s. ISBN 80-7300-077-6.
- [10] Pavlík, T. Biostatistika, přednáška číslo 11 [online]. Brno: Masarykova universita, 2011-43s. [cit. 21. května 2011]. Dostupný z [www: <http://www.iba.muni.cz/esf/res/file/bimat-prednasky/biostatistika-pro-matematickou-biologii/BpMB-11.pdf>](http://www.iba.muni.cz/esf/res/file/bimat-prednasky/biostatistika-pro-matematickou-biologii/BpMB-11.pdf)
- [11] Jarkovský, J., Littnerová, S., Vícerozměrné statistické metody, přednáška číslo 3 [online]. Brno: Masarykova universita, 2011- 51s. [cit. 21. května 2011]. Dostupný z [www: <http://www.iba.muni.cz/esf/res/file/bimat-prednasky/vicerozmerne-statisticke-metody/VSM-03.pdf>](http://www.iba.muni.cz/esf/res/file/bimat-prednasky/vicerozmerne-statisticke-metody/VSM-03.pdf)
- [12] Kratochvíl, T., Nízkofrekvenční a audio elektronika, přednáška číslo 2 [online]. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2011. 51s. [cit. 21. května 2011]. Dostupný z [www: <https://krel.feec.vutbr.cz/VYUKA/B_EST/prezencni/BNFE/Prednasky/bnfe_prednaska_2.pdf>](https://krel.feec.vutbr.cz/VYUKA/B_EST/prezencni/BNFE/Prednasky/bnfe_prednaska_2.pdf)

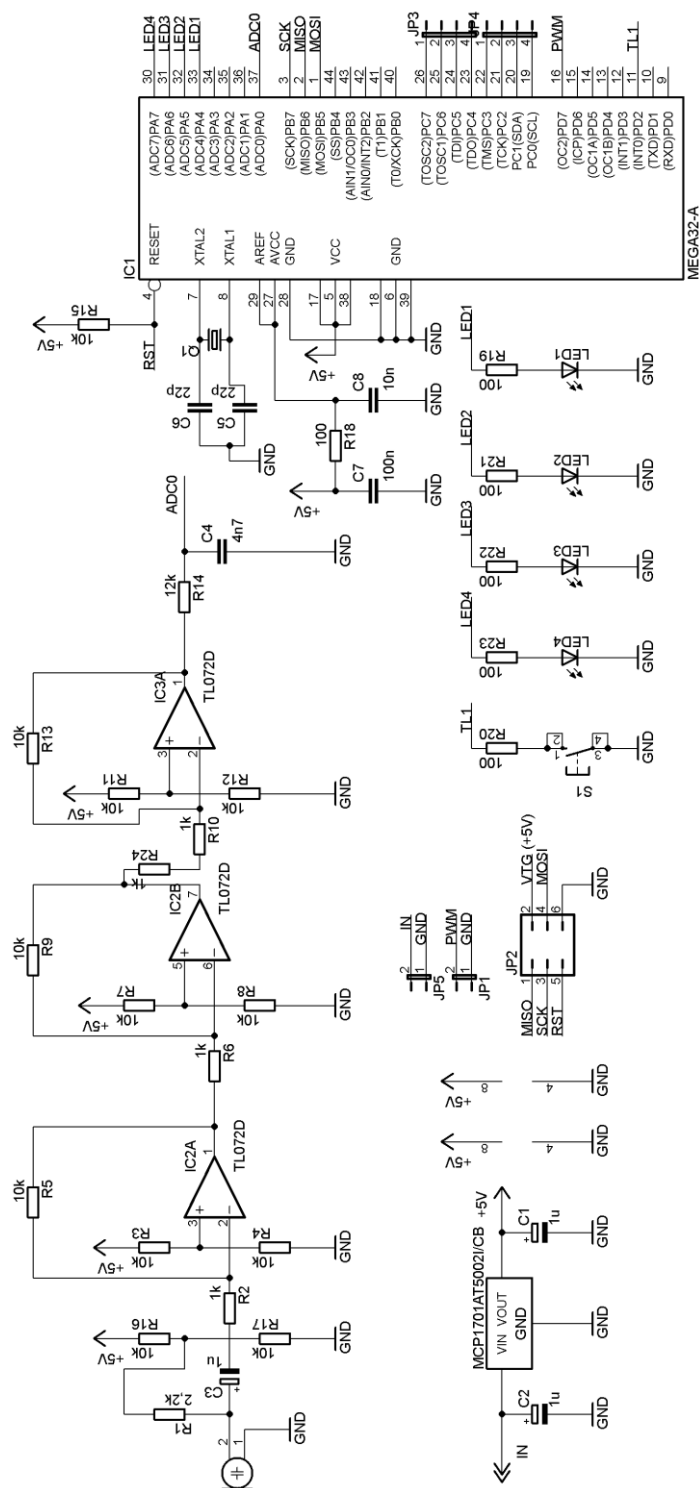
- [13] Kallukalam, I., Agrawal, U. ECE 476 Final Project Digital Voice Recorder [online] Cornell University 2005. [cit. 21. května 2011]. Dostupný z www: <http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2005/ik48_ua22/mysite/index.htm>
- [14] Vorel, P., Průmyslová elektronika [online]. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2007. 151s. [cit. 21. května 2011]. Dostupný z www: <https://www.vutbr.cz/www_base/priloha.php?dpid=21762>
- [15] KREJČERÍK, A. Spínané napájecí zdroje s obvody TOPSwitch, 1.vyd. Praha: BEN, 2002, 400s. ISBN 80-7300-031-8
- [16] MCP1701A [online]. Microchip datasheet [cit. 21. května 2011]. Dostupný z www: <<http://www.gme.cz/dokumentace/dokumenty/934/934-073/dsh.934-073.1.pdf>>
- [17] B&B elektro [online] Elektroinstalační krabice [cit. 21. května 2011]. Dostupné z www: <http://www.bbelektro.cz/fotocache/mid/polyprofil_UK-1-68_SZH.jpg>

SEZNAM PŘÍLOH

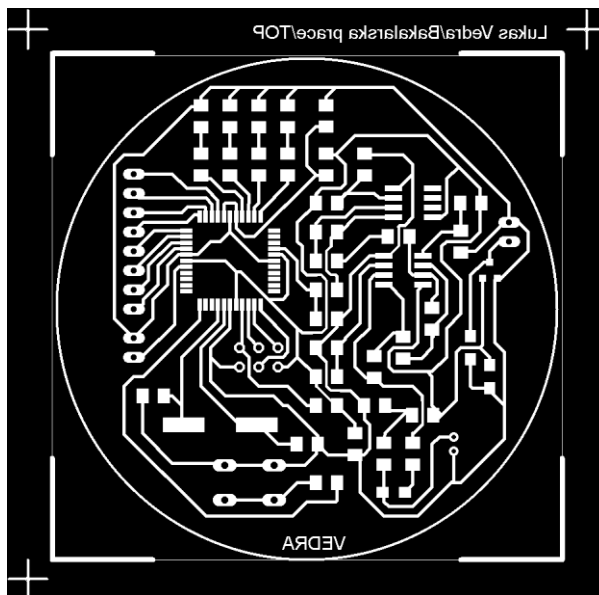
A	Návrh zařízení	35
A.1	Obvodové zapojení	35
A.2	Deska plošného spoje 1	36
A.3	Deska plošného spoje 2	36
A.4	Osazovací plán plošného spoje 1	37
A.5	Osazovací plán plošného spoje 2	37
B	Seznam součástek	38
C	Obsah CD	40

A NÁVRH ZAŘÍZENÍ

A.1 Obvodové zapojení

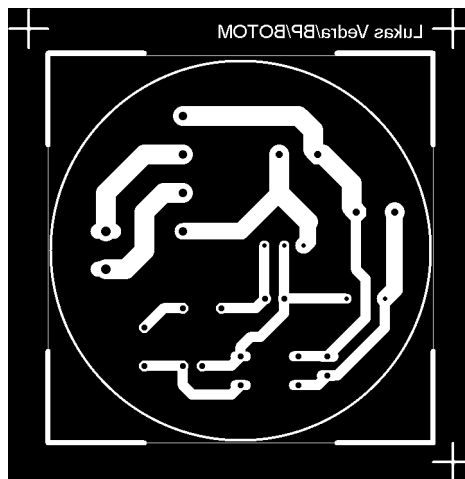


A.2 Deska plošného spoje 1



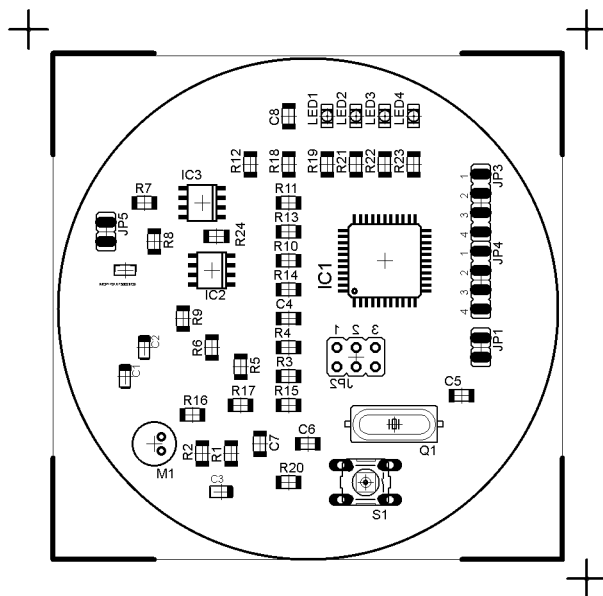
Rozměr desky 67,31 x 67,31 [mm], měřítko M1:1

A.3 Deska plošného spoje 2



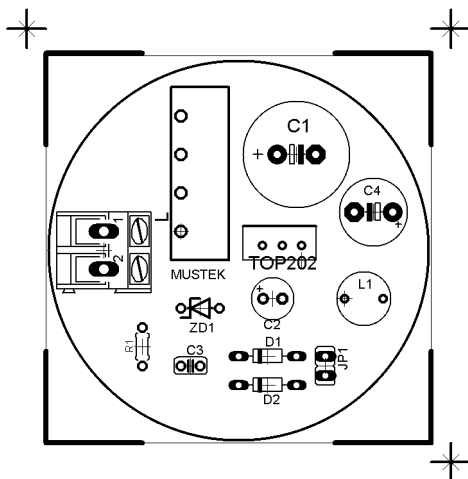
Rozměr desky 50,80 x 50,80 [mm], měřítko M1:1

A.4 Osazovací plán plošného spoje 1



Rozměr desky 67,31 x 67,31 [mm], měřítko M1:1

A.5 Osazovací plán plošného spoje 2



Rozměr desky 50,80 x 50,80 [mm], měřítko M1:1

B SEZNAM SOUČÁSTEK

Označení	Hodnota	Pouzdro	Popis
C1	1u	SMC_A	Tantalový kondenzátor
C2	1u	SMC_A	Tantalový kondenzátor
C3	1u	SMC_A	Tantalový kondenzátor
C4	4n7	C1206	Keramický kondenzátor
C5	22p	C1206	Keramický kondenzátor
C6	22p	C1206	Keramický kondenzátor
C7	100n	C1206	Tantalový kondenzátor
C8	10n	C1206	Keramický kondenzátor
IC1	MEGA32-A	TQFP44	Mikrokontrolér
IC2	TL072D	SO08	Operační zesilovač
IC3	TL072D	SO08	Operační zesilovač
JP1	1 x 2	JP1	Jumper
JP2	2 x 3	JP3Q	Jumper
JP3	1 x 4	JP4	Jumper
JP4	1 x 4	JP4	Jumper
JP5	1 x 2	JP1	Jumper
36892	White	1206	LED dioda
37257	White	1206	LED dioda
37622	White	1206	LED dioda
37987	White	1206	LED dioda
M1		WM-63PR	Elektrodynamický mikrofon
MCP	5V	SOT23-3	Stabilizátor MCP1701AT
Q1	16 Mhz	HC49UP	Krystal
R1	2,2k	R1206	Drátový rezistor
R2	1k	R1206	Drátový rezistor
R3	10k	R1206	Drátový rezistor
R4	10k	R1206	Drátový rezistor
R5	10k	R1206	Drátový rezistor
R6	1k	R1206	Drátový rezistor
R7	10k	R1206	Drátový rezistor
R8	10k	R1206	Drátový rezistor
R9	10k	R1206	Drátový rezistor
R10	1k	R1206	Drátový rezistor
R11	10k	R1206	Drátový rezistor
R12	10k	R1206	Drátový rezistor
R13	10k	R1206	Drátový rezistor
R14	12k	R1206	Drátový rezistor
R15	10k	R1206	Drátový rezistor

R16	10k	R1206	Drátový rezistor
R17	10k	R1206	Drátový rezistor
R18	100	R1206	Drátový rezistor
R19	100	R1206	Drátový rezistor
R20	100	R1206	Drátový rezistor
R21	100	R1206	Drátový rezistor
R22	100	R1206	Drátový rezistor
R23	100	R1206	Drátový rezistor
R24	1k	R1206	Drátový rezistor
S1		B3F-10XX	Tlačítko

Seznam součástek pomocného zdroje:

Označení	Hodnota	Pouzdro	Popis
C1	10u/400V	E5-13	Elektrolytický kondenzátor
C2	47u/50V	E2,5-6	Elektrolytický kondenzátor
C3	100n/35V	C025-02	Tantalový kondenzátor
C4	100u/50V	E5-8,5	Elektrolytický kondenzátor
D1	BYV26C	SOD57	Univerzální dioda
D2	BYV26C	SOD57	Univerzální dioda
JP1		JP1	Jumper
L	N	AK300/2	Konektor napájení
L1	1mH	L2	Tlumivka
MUSTEK	B250C8000	MUSTEK	Usměrňovací můstek
R1	22	0204/5	Drátový rezistor
TOP1	TOP202	TO220	TOP202YAI
ZD1	2,4V	DO35	Zenerova dioda BZX83V002.4

C OBSAH CD

- Datasheety použitých součástek (.pdf)
- Text bakalářské práce (.pdf)
- Zdrojový kód (.c a .h)
- Podklady na výrobu DPS (.sch a .brd)
- Fotografie (.jpg)