



ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Fakulta elektrotechnická
Katedra aplikované elektroniky a telekomunikací

BAKALÁŘSKÁ PRÁCE

Akustický generátor

Autor práce: Vojtěch Ježek
Vedoucí práce: Ing. Zuzana Petránková, Ph.D.

Plzeň 2015

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vojtěch JEŽEK**
Osobní číslo: **E12B0363P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Název tématu: **Akustický generátor**
Zadávající katedra: **Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování:

1. Navrhněte a zhotovte funkční vzorek číslicově řízeného akustického generátoru, který je určen pro experimentální laboratorní práci se sluchovými podněty.
2. Akustický generátor bude umožňovat práci ve dvou různých režimech provozu. V režimu 1 bude umožňovat nastavení počtu generovaných podnětů, frekvenci podnětů a délku jejich trvání. V tomto režimu bude ovládat samostatný výstup pro synchronizační značky umožňující start průměrování snímaného signálu. V režimu 2 bude umožňovat nastavení frekvence podnětů. V tomto režimu bude ovládán samostatným vstupem.
3. Funkční vzorek bude bateriově napájený.
4. Vytvořené technické a programové prostředky podrobně popište.

Rozsah grafických prací: **podle doporučení vedoucího**

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

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

Seznam odborné literatury:

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

Vedoucí bakalářské práce:

Ing. Zuzana Petránková, Ph.D.

Katedra aplikované elektroniky a telekomunikací

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

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

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



Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 15. října 2014

Abstrakt

Tato bakalářská práce se zabývá návrhem a zhotovením akustického generátoru s proměnnými parametry, který bude sloužit k experimentální laboratorní práci se sluchovými podněty.

Generátor umožňuje práci v různých režimech provozu. V režimu 1 je možno generovat výstupní akustický signál s proměnnými parametry jako je frekvence, délka trvání jednotlivých impulzů a jejich počet. V režimu 2 je možno signál o nastavené frekvenci vyvést na externí zesilovač a pomocí vývodů na desce plošných spojů řídit externě dobu trvání jednotlivých pulzů.

Celý systém je řízen procesorem ATmega328P, naprogramovaným pomocí jazyka WIRING. Ke generaci sinusového signálu je použito obvodu DDS. Bateriové napájení je řešeno pomocí externího připojení power-banky.

Klíčová slova

akustický generátor, tvorba sinusového signálu, přímá digitální syntéza, programovací jazyk WIRING, Arduino

Abstract

Ježek, Vojtěch. *Acoustic generator [Akustický generátor]*. Pilsen, 2015. Bachelor thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications. Supervisor: Zuzana Petránková

This thesis describes the design and fabrication of the acoustic generator with variable parameters that will be used for experimental laboratory work with auditory stimuli.

The generator can work in various modes of operation. In mode 1 it is possible to generate the output audio signal with variable parameters like frequency, duration of individual pulses and their number. In Mode 2 can set the frequency of the signal output at an external amplifier and via terminals on the PCB to manage externally duration of each pulse.

The entire system is controlled by the processor ATmega328P, programmed using the programming language WIRING. To generate the sinusoidal signal is used DDS circuit. Battery power is supplied via an external connection of power-Bank.

Keywords

acoustic generator, creation of a sine wave, direct digital synthesis, WIRING, ARDUINO

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci 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 závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 8. června 2015

Vojtěch Ježek

.....

Podpis

Obsah

Seznam obrázků	vi
Seznam tabulek	vii
Seznam symbolů a zkratk	viii
1 Úvod	1
2 Metody generování sinusového signálu	2
2.1 Zpětnovazební oscilátory	2
2.1.1 Oscilátor s Wienovým článkem	2
2.2 Fázový závěs	3
2.2.1 Fázový detektor	4
2.2.2 Napětím řízený oscilátor	4
2.3 Přímá digitální syntéza	5
2.4 Funkční generátory	6
3 Popis jednotlivých hardwarových prvků	7
3.1 Popis mikrokontroléru ATmega328P	8
3.2 Popis DDS obvodu AD9833	9
3.3 Popis analogového přepínače 74HC4066	11
3.4 NF zesilovač LM386-N	12
3.5 Vnitřní reproduktor	13
3.6 LCD display	13
3.7 Uložný box pro zařízení	14
3.8 Bateriový zdroj	14
4 Popis softwarové části zařízení	15
4.1 Programovací jazyk WIRING a prostředí programu Arduino IDE	15
4.2 Použité datové typy, funkce a knihovny	16
4.2.1 Funkce a knihovny použité v programu a jejich popis	17
4.2.1.1 Kód pro SPI komunikaci s AD9833	18
4.3 Popis program	20

5 Závěr	22
Reference, použitá literatura	23
Přílohy	24
A Deska plošných spojů a schéma navrženého akustického generátoru	24
A.1 Deska plošných spojů testovací verze	24
A.2 Schéma zapojení	26

Seznam obrázků

2.1	Schéma Wienova oscilátoru	3
2.2	Blokové schéma obvodu PLL	4
2.3	Blokové schéma obvodu DDS	5
2.4	Blokové schéma obvodu XR-2206 [3]	6
3.1	Blokové schéma navrženého obvodu	7
3.2	Blokové schéma obvodu AD9833 [5]	9
3.3	Konfigurace pinů obvodu DDS [5]	10
3.4	Schéma jednoho spínače obvodu 74HC4066 [8]	12
3.5	Základní zapojení obvodu LM386 [7]	12
3.6	Konfigurace pinů obvodu LM386 [7]	12
4.1	Arduino IDE [10]	16
4.2	Vývojový diagram hlavní části programu	20
4.3	Vývojový diagram pro režim 1	21
4.4	Vývojový diagram pro režim 2	21
A.1	DPS: Osazovací výkres vrstvy BOTTOM	24
A.2	DPS: Vrstva TOP	25
A.3	DPS: Vrstva BOT	25
A.4	Schéma zapojení akustického generátoru	26

Seznam tabulek

3.1	Popis jednotlivých pinů obvodu ATmega328P [6]	8
3.2	Popis jednotlivých pinů obvodu AD9833 [5]	10
3.3	Popis jednotlivých pinů obvodu 74HC4066 [8]	11
3.4	Popis jednotlivých pinů LCD displeje [9]	13
4.1	Datové typy a jejich rozsah	16
4.2	Přehled jednotlivých funkcí společně s počtem a typem vstupních parametrů a návratovou hodnotou [10]	17
4.3	Význam jednotlivých bitů při komunikaci s AD9833 [5]	18

Seznam symbolů a zkratek

PLL	Phase Locked Loop. Fázový závěs.
VCO	Voltage Controlled Oscillator. Napětím řízený oscilátor.
XOR	Obvod, který vykonává logickou funkci exklusivního součtu.
NCO	Numerical Controlled Oscillator. Číslicově řízený oscilátor.
DDS	Direct Digital Synthesis. Přímá digitální syntéza.
AD	Analogově digitální
DA	Digitálně analogový
ROM	Read Only Memory. Paměť pouze pro čtení.
SPI	Serial Peripheral Interface. Sériové rozhraní pro periferie.
LCD	Liquid Crystal Display. Display z tekutých krystalů.
CPU	Central Processing Unit. Centrální procesorová jednotka.
PDIP	Plastic Dual-In-line Package. Plastové klasické pouzdro pro montáž součástek.
GPR	General-Purpose register. Registry pro univerzální použití v procesorech
PWM	Pulse Width Modulation. Pulzní šířková modulace.
THD	Total Harmonic Distortion. Harmonického zkreslení.
USB	Universal Serial Bus. Univerzální sériová sběrnice.
IDE	Integrated Development Environment. Integrované vývojové prostředí.
MISO	Master-In,Slave-Out. Sériová komunikace, kdy signál jde směrem k zařízení označeném jako master.
MOSI	Master-Out,Slave-In. Sériová komunikace, kdy signál jde směrem k zařízení označeném jako slave.
SS	Slave Select. Výběr požadovaného zařízení označeného jako slave.
MSB	Most significant bit. Nejvyšší bit slova.
LSB	Least significant bit. Nejnižší bit slova.

1

Úvod

V této bakalářské práci budou postupně probrány jednotlivé kroky při návrhu a zhotovení číslicově řízeného akustického generátoru. Akustický generátor, jak již název napovídá, slouží ke generování akustických signálů o různé frekvenci. Tyto signály, nejčastěji sinusové, je možné generovat mnoha způsoby. Některé způsoby generování budou podrobně probrány v kapitole 2. V této kapitole budou také uvedeny možné způsoby propojení daných generátorů s číslicovou technikou.

Vytvořený funkční vzorek bude moci pracovat v různých režimech práce. V prvním režimu bude možné přes interní zesilovač a reproduktor generovat akustický signál o různé frekvenci, době trvání a o různém počtu generovaných impulzů. Ve druhém režimu se bude moci připojit externí zesilovač, a tím si bude moci uživatel řídit kvalitu reprodukce sám a nebude muset spoléhat na interní reproduktor. V tomto režimu bude možné připojit externí generátor, který bude určovat délku jednotlivých impulzů. Pro tyto účely je zde kapitola 3, ve které budou popsány jednotlivé stavební prvky zařízení a také zde bude vysvětlena jejich funkce. Jelikož je nutné, aby výsledné zařízení bylo schopné pracovat s bateriovým napájením, tak další důležitou informací o těchto obvodech bude také jejich spotřeba.

V poslední části bakalářské práce bude probrána softwarová část zařízení. Vybraný procesor od společnosti atmel je programovatelný v jazyce WIRING pomocí vývojového prostředí Arduino IDE. V této části budou podrobně probrány výhody a nevýhody tohoto jazyka a také zde bude popsáno již zmíněné vývojové prostředí. Dále zde bude popsán software, který bude komunikovat s vybraným typem zařízení, které generuje sinusový signál. Poslední část této kapitoly bude věnována již vytvořenému softwaru a jeho popisu.

2

Metody generování sinusového signálu

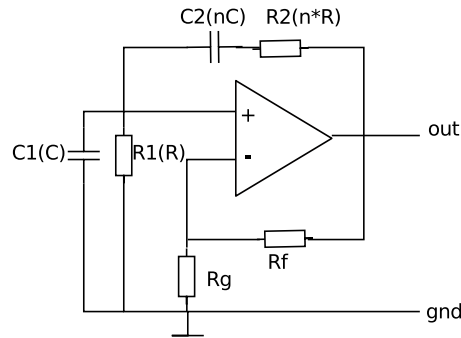
Vytvořený akustický generátor má za úkol generovat výstupní akustický signál s proměnnými parametry. V režimu 1 jsou těmito parametry frekvence, délka trvání a počet jednotlivých impulzů. V režimu 2 jsou těmito parametry frekvence a délka jednotlivých impulzů, která se bude řídit externím generátorem. V této kapitole se budu zabývat metodami vytváření sinusových průběhů a změnou jejich frekvence.

2.1 Zpětnovazební oscilátory

Při přivedení kladné zpětné vazby do obvodu zesilovače se může stát, že daný obvod přejde do nestabilního stavu a změní se v oscilátor. K tomuto jevu dojde při splnění dvou podmínek. První podmínkou, tzv. amplitudovou, je, aby se zesílení zesilovače se zavedenou kladnou zpětnou vazbou blížilo k nekonečnu. Druhou podmínkou, tzv. fázovou, je, aby fázový úhel mezi úhlem signálu ze zesilovače a úhlem signálu způsobeným zpětnovazebním obvodem byl 0, resp. $2 \cdot k \cdot \pi$, kde $k = \dots, -2, -1, 0, 1, 2, \dots$. Tyto podmínky musí platit pro jednu frekvenci. Při splnění těchto podmínek pro více frekvencí by docházelo ke zdeformování výsledného signálu vlivem těchto frekvencí. Pro frekvence řádů jednotek Hz do několika kHz se nejčastěji používají zpětnovazební RC oscilátory. Pro vysoké frekvence se používají LC oscilátory.

2.1.1 Oscilátor s Wienovým článkem

Wienův oscilátor můžeme zařadit mezi zpětnovazební RC oscilátory v můstkovém zapojení. Základem obvodu jsou dvě větve záporné a kladné zpětné vazby. Tento obvod se chová jako pásmová propust se středním rezonančním kmitočtem ω_0 , na kterém má nejmenší útlum a nulový fázový posun [1]. Schéma základního zapojení Wienova článku je na obr. 2.1.



Obr. 2.1: Schéma Wienova oscilátoru

Pokud zvolíme $n=1$, tzn. hodnoty R_1 a R_2 (resp. C_1 a C_2) jsou stejné, tak pro komplexní napěťový přenos Wienova článku platí:

$$F(j \cdot \omega) = \frac{U_2(j \cdot \omega)}{U_1(j \cdot \omega)} = \frac{1}{3 + j \cdot \omega \cdot R \cdot C + \frac{1}{j \cdot \omega \cdot R \cdot C}} \quad (2.1)$$

Z tohoto přenosu plyne, pokud mají být splněny obě podmínky vzniku oscilací, že imaginární část přenosové funkce je rovna 0. Tato situace nastává při platnosti podmínky:

$$j \cdot \omega \cdot R \cdot C = -\frac{1}{j \cdot \omega \cdot R \cdot C} \quad (2.2)$$

Po úpravě vzorce dostáváme vzorec pro frekvenci oscilací Wienova oscilátoru, a to:

$$f_{OSC} = \frac{1}{2 \cdot \pi \cdot R \cdot C} \quad (2.3)$$

Pro napěťový přenos platí při použití f_{OSC} a tím nulové imaginární složky:

$$F(j \cdot \omega) = \frac{1}{3} \quad (2.4)$$

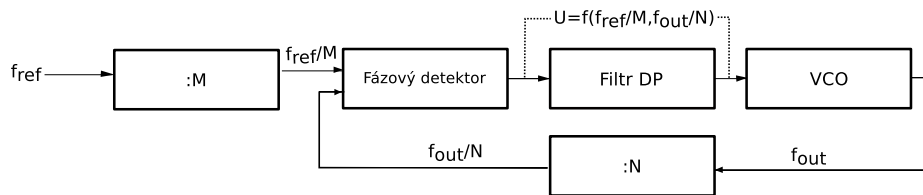
Pro zajištění netlumených oscilací je však potřeba napěťového přenosu rovné 1. Proto se k oscilátoru přidává operační zesilovač se zesílením rovným 3. Toto zesílení se nastavuje pomocí externích rezistorů v obrázku 2.1 označených jako R_f a R_g . Při použití Wienova článku nastane ten problém, že výběr daných součástek platí jen pro určitou frekvenci. Jediným způsobem pro digitální přeladění oscilátoru by bylo použití digitálních potenciometrů, a tím řídit jak zesílení pomocí rezistorů R_g a R_f , tak i výstupní frekvenci oscilátoru pomocí změny R [4].

2.2 Fázový záměr

Fázový záměr (angl. Phase-Locked Loop, dále jen PLL) je obvod, který se hojně využívá v přenosových systémech, v rádiové a televizní technice či v regulační technice. Lze ho však využít i pro frekvenční syntézu.

Fázový závěs obsahuje fázový detektor. Tento detektor porovnává vstupní a výstupní frekvenci a z rozdílů fází jednotlivých kmitočtů vytváří napětí. Toto napětí je přivedeno na dolnofrekvenční propust, která vytvoří střední hodnotu a vyrovná případná zvlnění napětí. Toto napětí je přivedeno na vstup VCO (VCO, Voltage Controlled Oscillator). VCO vytvoří kmitočet, který je úměrný přivedenému napětí. Tato situace se opakuje až do vyrovnání frekvencí jednotlivých kmitočtů za předpokladu, že daný PLL je již zavěšený (tj. frekvence se nachází v pásmu udržení). Blokové schéma PLL je na obrázku 2.2.

Při zavedení programovatelné kmitočtové děličky do vstupního signálu a do zpětné vazby fázového závěsu můžeme regulovat frekvenci na výstupu dle vztahu 2.5.



Obr. 2.2: Blokové schéma obvodu PLL

$$f_{out} = f_{ref} \cdot \frac{N}{M}, \quad (2.5)$$

kde M je dělicí poměr na děličce referenčního kmitočtu, N je dělicí poměr na zpětné vazbě PLL a f_{ref} je referenční kmitočet.

Problém v případě PLL je však ten, že většina integrovaných obvodů, které jsou dnes na trhu, vytváří obdélníkový výstupní signál, a ne sinusový průběh. Řešení tohoto problému by spočívalo v tom, že by se daný obdélníkový signál převedl pomocí integračního členu na signál trojúhelníkový¹ a následně by se vyfiltrovaly vyšší harmonické složky tohoto signálu a převedl by se daný signál na harmonický.

2.2.1 Fázový detektor

Fázový detektor lze realizovat mnoha způsoby. Jedním z nejjednodušších je logický obvod XOR, ten na svých vstupech porovnává fázový rozdíl přivedených napětí a na svém výstupu generuje vysokou úroveň při rozdílných hodnotách a nízkou úroveň při stejných hodnotách napětí.

Podmínkou funkce takového obvodu je velká strmost hran vstupních signálů a střída signálu by měla být rovna jedné.

2.2.2 Napětím řízený oscilátor

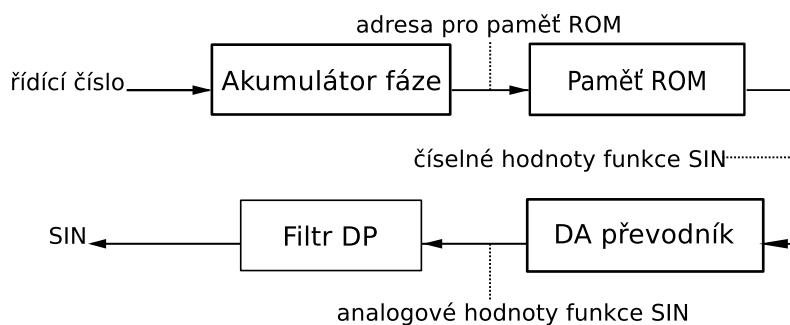
Napětím řízené oscilátory lze rozdělit do dvou základních skupin. První z nich jsou oscilátory analogové, zpravidla konstruované jako relaxační nebo harmonické oscilátory. Druhou

¹Toto řešení vede k snadnějšímu návrhu a realizaci filtrů

skupinu tvoří digitální, někdy též zvané číslicově řízené oscilátory. Relaxační generátory jsou tvořené z přepínaného zdroje proudu, který nabíjí a vybíjí kondenzátory, a tím určují výstupní frekvenci, která se pohybuje v řádu několika desítek kHz s chybou několika promile. Výstupní signál těchto obvodů je obdélníkový, a ne harmonický. Naproti tomu se vyrábějí oscilátory harmonické, kde se pro konstrukci využívá obvodu LC s přeladitelným varikapem, u kterého kapacita přechodu diody klesá se vzrůstajícím závěrným napětím, a tím je určena výstupní frekvence harmonického signálu. Kapacita varikapu se mění pouze v malém rozsahu (jednotky až desítky pF) a je vhodná pouze pro kmitočty řádu desítek MHz [2].

2.3 Přímá digitální syntéza

Přímá digitální syntéza (angl. Direct Digital Synthesis, dále jen DDS) je metoda, která vytváří analogový signál pomocí čistě digitálních prvků a DA převodníku. Tato metoda je vhodná pro generování sinusových průběhů. DDS však není omezen pouze na sinusové signály, ale lze s ním generovat i jiné signály jako jsou obdélníkové, pilové či trojúhelníkové.



Obr. 2.3: Blokové schéma obvodu DDS

Uvedené blokové schéma (obr. 2.3) znázorňuje principiální funkci obvodu DDS. Základem obvodu je akumulátor fáze, na který je přivedeno řídicí číslo. Toto číslo určuje, o kolik se posune čítač při jednom hodinovém impulzu referenčního hodinového signálu, který generuje adresu pro paměť ROM. Po přetečení čítače se vrací do původního stavu a čítání pokračuje dál do dalšího přetečení. Takto generovaná adresa je přiváděna na paměť ROM, která obsahuje hodnoty jednotlivých amplitud sinusového signálu. Tyto hodnoty jsou pomocí DA převodníku převedeny na již analogový signál.

Velikost výstupní frekvence signálu závisí na velikosti paměti ROM a na referenčním kmitočtu dle vztahu 2.6.

$$f_{out} = f_{ref} \cdot \frac{N}{2^n}, \quad (2.6)$$

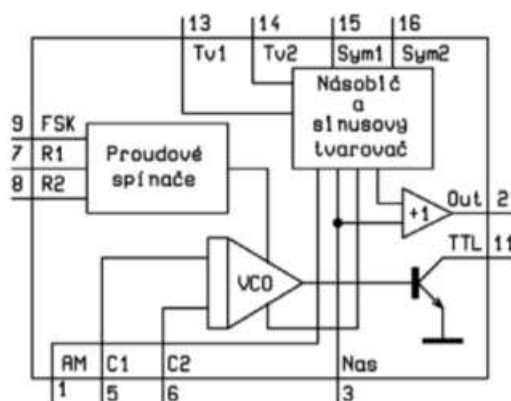
kde f_{ref} je referenční kmitočet hodinového signálu, N je řídicí číslo a n je počet bitů daného akumulátoru fáze, resp. počet řádků s příslušnými hodnotami amplitudy v paměti ROM.

Výhoda DDS je v jednoduchém řízení pomocí číslicových obvodů, velkém rozsahu a přesnosti generovaných kmitočtů. Možností vytvoření DDS je hned několik. Jedním z nich je zkonstruovat DDS pomocí mikroprocesoru doplněného o DA převodník. V takovém případě slouží mikroprocesor jako generátor hodnot amplitudy pro DA převodník. Tyto hodnoty amplitudy jsou uloženy v paměti dat a jsou vybírány pomocí programu v mikroprocesoru. Podobnou možností jak vytvořit obvod DDS je pomocí programovatelného logického pole. Základem takového obvodu je čítač, který čítá se zvoleným krokem, a tím nastavuje adresu pro paměť ROM, ze které se vyčítají hodnoty jednotlivých vzorků do DA převodníku. V dnešní době lze použít obvody DDS v integrované verzi. Tyto obvody mají všechny potřebné bloky (akumulátor fáze, paměť hodnot i DA převodník) integrované na jednom čipu. Výhodou tohoto řešení je, že k běhu je potřeba minimální množství externích součástek, malá spotřeba a malé rozměry čipu. Tyto obvody jsou většinou programovány pomocí rychlého sériového rozhraní SPI.

2.4 Funkční generátory

Funkční generátory jsou obvody, které dokáží s minimem externích součástek generovat různé typy průběhů. Pro vysvětlení funkce těchto obvodů bude popsán obvod s označením XR-2206. Tento obvod obsahuje čtyři funkční bloky (viz obrázek 2.4). První blok VCO je řízen pomocí externího kondenzátoru (C) a vstupních proudů. Tento blok určuje výstupní frekvenci signálu. Další blok jsou proudové spínače, které dodávají proud do VCO a jsou řízeny externími odpory (R1 a R2). Tyto odpory jsou přepínané pomocí vstupu FSK. Třetí blok je násobič a sinusový tvarovač. Poslední částí jsou výstupní obvody [3]. Pro výstupní frekvenci obvodu platí následující vztah.

$$f_{out} = \frac{1}{(R1 + R2) \cdot C} \quad (2.7)$$



Obr. 2.4: Blokové schéma obvodu XR-2206 [3]

a silnými je označen akustický signál. Pro zobrazování hodnot a režimu je použit LCD display.

3.1 Popis mikrokontroléru ATMega328P

Jádrem celého akustického generátoru je mikrokontrolér od společnosti atmel, a to konkrétně typ s názvem ATMega328P. Tento použitý procesor, dodávaný v 28 pinovém provedení PDIP, disponuje 32 kB programovatelné flash paměti, 1 kByte EEPROM paměti, 2 kB RAM paměti a 32 GPR registrů. Mezi periferní obvody integrované v samotném integrovaném obvodu patří tři (dva 8-bitové a jeden 16-bitový) časovače/čítače se samostatnými před-děličkami a komparátory, 6 PWM kanálů, 6 kanálů 10-bitového AD převodníku, port pro sériovou komunikaci přes SPI, programovatelný watchdog časovač atd. Mezi další vlastnosti tohoto obvodu patří až 5 možných spánkových režimů, které umožňují snížit spotřebu. Celý obvod může být napájen v širokém rozsahu hodnot napětí, konkrétně od 1,8 V do 5,5 V, a je tedy tím ideálním kandidátem pro použití s bateriovým napájením 5 V. Pro konstrukci zařízení jsem vybral mikrokontrolér s již nahaným bootloaderem od společnosti ARDUINO, který umožňuje programování v upraveném jazyku C/C++ s názvem Wiring. Tento jazyk společně s programovacím prostředím jsem podrobně popsal v kapitole 4.1. V tabulce 3.1 jsou vypsány jednotlivé piny a jejich význam.

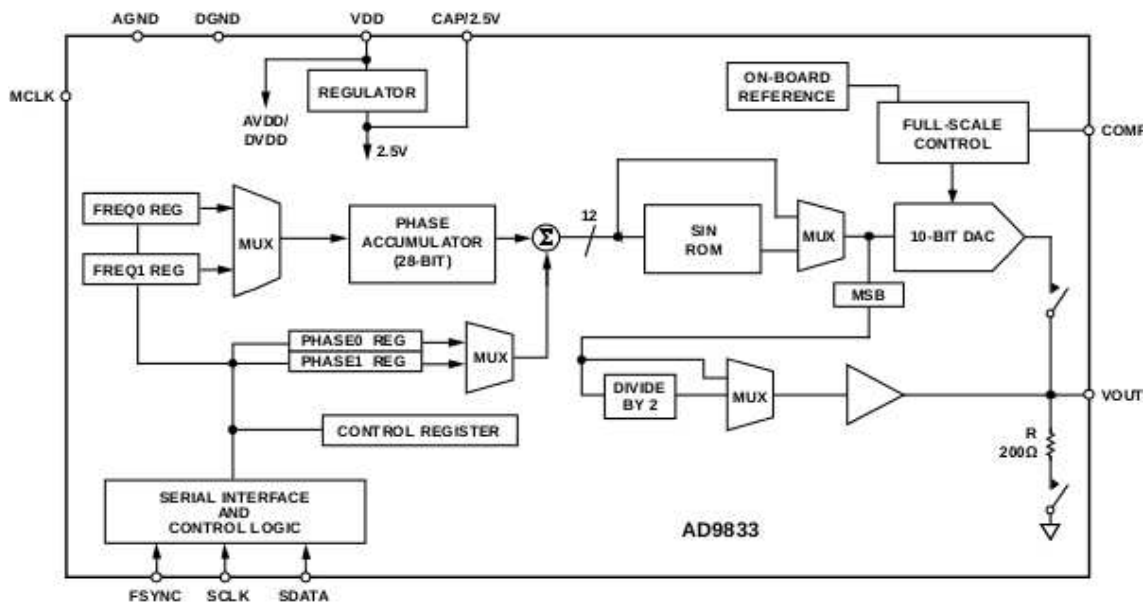
Č. pinu	Název	Popis	Č. pinu	Název	Popis
1	RST	reset	28	A5	analogový vstup 5
2	Rx	Synchronní odesílání	27	A4	analogový vstup 4
3	Tx	Synchronní příjem	26	A3	analogový vstup 3
4	D2	digitální vstup/výtup 2	25	A2	analogový vstup 2
5	D3	digitální vstup/výtup 3	24	A1	analogový vstup 1
6	D4	digitální vstup/výtup 4	23	A0	analogový vstup 0
7	VCC	napájení	22	GND	zem
8	GND	zem	21	AREF	analogové referenční napětí
9	XT1	vstup vývodu krystalu 1	20	AVCC	napájení AD převodníku
10	XT2	vstup vývodu krystalu 2	19	D13	digitální vstup/výtup 13
11	D5	digitální vstup/výtup 5	18	D12	digitální vstup/výtup 12
12	D6	digitální vstup/výtup 6	17	D11	digitální vstup/výtup 11
13	D7	digitální vstup/výtup 7	16	D10	digitální vstup/výtup 10
14	D8	digitální vstup/výtup 8	15	D9	digitální vstup/výtup 9

Tab. 3.1: Popis jednotlivých pinů obvodu ATMega328P [6]

Piny s názvem Rx a Tx lze také použít jako digitální vstupy/výstupy. To samé platí pro všechny analogové vstupy. Piny D10, D11, D12 a D13 slouží jako piny pro SPI komunikaci s perifériemi označenými jako slave. Na vývody XT1 a XT2 je paralelně připojen 16 MHz externí krystal, který udává takt pro vnitřní obvody mikrokontroléru. Celý procesor je na desce plošných spojů připojen přes 28 pinový sokl. Toto umožňuje vyjmutí procesoru pro účely jeho programování mimo desku.

3.2 Popis DDS obvodu AD9833

Jako DDS obvod byl vybrán obvod od firmy Analog Devices, a to konkrétně AD9833. Jedná se o nízkopříkonový DDS obvod se spotřebou 12,5 mW, který dokáže generovat sinusový, trojúhelníkový i obdélníkový signál v rozsahu 0 až 12,5 MHz při použití referenčního oscilátoru na frekvenci 25 MHz, a to v rozlišení 28-bitového DA převodníku. Napájecí napětí tohoto obvodu je od 2,3 V až 5,5 V a amplituda výstupního napětí (peak to peak) je 0,6 V. Pro referenční oscilátor nebyl použit externí oscilátor, ale byl použit jeden výstup z procesoru, který generuje za pomoci čítače výstupní signál 1 MHz. Tato úprava pouze snížila maximální možnou výstupní frekvenci na 500 kHz, která je však dostatečná. Funkční blokový diagram je na 3.2.



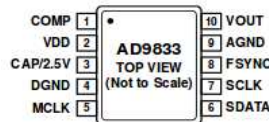
Obr. 3.2: Blokové schéma obvodu AD9833 [5]

Základem obvodu DDS je numericky řízený oscilátor(NCO), frekvenční a fázový modulátor, paměť ROM se sinovými vzorky, DA převodník a regulátor. Hlavní komponentou NCO je 28-bitový fázový akumulátor. Spojitý sinusový signál má fázový rozsah od $0-2\pi$, mimo tento rozsah se signál periodicky opakuje. To lze využít pro konstrukci NCO, kde hodnota 0 (označme $\Delta\varphi$) bude odpovídat hodnotě 0 rad a 2^{28} hodnotě 2π rad. Z tohoto předpokladu lze snadno dopočítat vzorec pro frekvenci, a to:

$$f_{out} = \frac{\Delta\varphi \cdot f_{MCLK}}{2^{28}}, \quad (3.1)$$

kde $0 < \Delta\varphi < 2\pi$, f_{MCLK} je referenční kmitočet externího oscilátoru.

Vstup $\Delta\varphi$ může být řízen ze dvou registrů, a to registr `FREQ0` nebo `FREQ1` 3.2. Mezi těmito registry lze přepínat pomocí bitu `FSELECT`. Tento signál by však nebyl sinusový, proto se za výstup NCO musí zkonvertovat z fázových hodnot pomocí paměti ROM na sinusový signál. Toto je docíleno tak, že vstupní informace o fázi se používá jako adresa paměti ROM, ve které jsou uloženy hodnoty amplitud sinusového signálu příslušející jednotlivým fázím. Tato číselná hodnota se přivede na vstup 10-bitového DA převodníku.



Obr. 3.3: Konfigurace pinů obvodu DDS [5]

Č. pinu	Název	Popis
1	COMP	Tento pin slouží k oddělení předpětí na DAC
2	VDD	Napájení obvodu
3	CAP/2,5V	DDS operuje na 2,5V, tento pin se připojuje pokud VDD přesáhne 2,7V
4	DGND	Digitální zem
5	MCLK	Referenční hodiny DDS(25MHz)
6	SDATA(MOSI)	Datový vstup
7	SCLK(SCLK)	Synchronizační hodiny
8	FSYNC(SS)	Tento pin slouží k výběru obvodu
9	AGND	Analogová zem
10	VOUT	Výstup obvodu

Tab. 3.2: Popis jednotlivých pinů obvodu AD9833 [5]

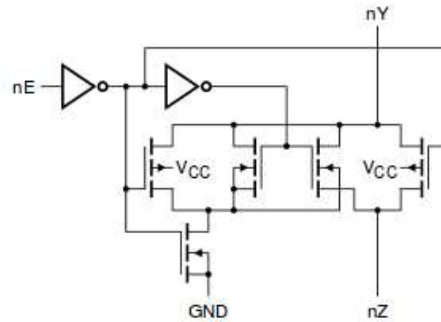
3.3 Popis analogového přepínače 74HC4066

Při vyvedení signálu z DDS přímo přes NF zesilovač by byl signál spojitý a nebylo by možné určit dobu trvání jednotlivých intervalů. Z tohoto důvodu je mezi obvod DDS a NF zesilovač zařazen integrovaný obvod 74HC4066. V integrovaném obvodu se 14 vývody jsou ukryty 4 analogové obousměrné spínače, které vykazují malý vnitřní odpor (50 Ω při VCC=4,5V) a malý činitel THD. Napájení tohoto obvodu může být od 2 V do 10 V a vstupní signál se může pohybovat v rozmezí od napětí země až do napájecího napětí obvodu. Popis jednotlivých pinů je v tabulce 3.3. Rozmístění pinů obvodu je shodné s touto tabulkou - Y značí vstupní pin, Z značí pin výstupní a E značí pin pro přepínání jednotlivých přepínačů. Jelikož je přepínač obousměrný, tzn. není rozdíl, jestli signál prochází ze vstupu na výstup nebo naopak, lze zaměnit Y za Z nebo Z za Y.

Číslo pinu	označení	Číslo pinu	označení
1	1Y	14	VCC
2	1Z	13	1E
3	2Z	12	4E
4	2Y	11	4Y
5	2E	10	4Z
6	3E	9	3Z
7	GND	8	3Y

Tab. 3.3: Popis jednotlivých pinů obvodu 74HC4066 [8]

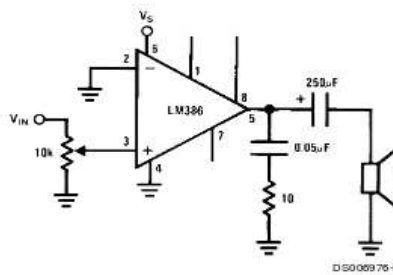
Schéma jednoho přepínače obvodu je na obrázku 3.4. Tento obvod funguje tak, že přivedením napětí na vývod obvodu (označený jako nE, kde n zastupuje číslo přepínače) zpřístupní cestu signálu. Ve stavu L se daný spínač rozezne a sinusový signál neprochází, ve stavu H přivedený signál prochází k výstupnímu pinu. Pro účely přepínání mezi interním a externím výstupem byly použity dva přepínače. Na oba dva je přiveden výstupní signál z DDS. Jeden z přepínačů vede k internímu zesilovači a dále k reproduktoru, druhý vede k výstupu na cinch ven z boxu. Pro časování sinusových impulzů jsou použity dva výstupy z mikrokontroléru, které je možné přes pinheader odstavit a použít externí zdroj klíčování sinusového signálu.



Obr. 3.4: Schéma jednoho spínače obvodu 74HC4066 [8]

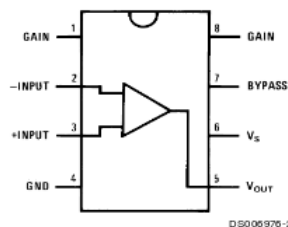
3.4 NF zesilovač LM386-N

Tento nízkofrekvenční zesilovač v pouzdru DIP 8 se díky svému napájecímu napětí (které je v rozsahu od 4 V do 12 V) a díky malé spotřebě (325 mW při napájení 6 V) hodí k bateriovému napájení. Jeho další výhodou je malý činitel THD (0,2% při 6 V) a minimum potřebných externích součástek. Základní zapojení zesilovače je na obrázku 3.6 [7].



Obr. 3.5: Základní zapojení obvodu LM386 [7]

V tomto zapojení jsou piny GAIN nepropojeny a zisk zesilovače je rovný 20. Toto zesílení je však příliš vysoké a zesilovač by se dostával do saturace, a proto je před vstup zesilovače (na obrázku 3.5 označeným jako +INPUT) předřazen potenciometr, kterým se řídí amplituda signálu tak, aby výstupní zesílený signál nebyl zkreslený.



Obr. 3.6: Konfigurace pinů obvodu LM386 [7]

3.5 Vnitřní reproduktor

Pro reprodukci akustického signálu bez použití vnějšího zesilovače a reproduktoru byl vybrán reproduktor Kingstate KAAG15008C. Tento reproduktor má akustickou šířku pásma dle dokumentace [11] od 0 Hz do 20 kHz. Výkon reproduktoru je 300 mW a jeho impedance je 8Ω .

3.6 LCD display

Pro zobrazení hodnot a nastavení procesoru byl použit LCD display od firmy TinSharp, a to typ LTM1602C. Tento display byl vybrán kvůli tomu, že obsahuje řadič typu HD44780, který dokáže přímo přes vytvořené knihovny v arduinu komunikovat s mikroprocesorem. Tento display dokáže zobrazit až 80 znaků v matici 16x2. Napájecí napětí displaye je 5 V. V následující tabulce 3.4 jsou vyjmenovány jednotlivé piny a popis jejich funkce.

Č. pinu	Název	Popis
1	GND	zem
2	VCC	napájení
3	V_0	kontrast
4	RS	řídící signál
5	R/W	čtení/zápis
6	E	
7	D0	datový vstup 0
8	D1	datový vstup 1
9	D2	datový vstup 2
10	D3	datový vstup 3
11	D4	datový vstup 4
12	D5	datový vstup 5
13	D6	datový vstup 6
14	D7	datový vstup 7
15	A	anoda
16	K	katoda

Tab. 3.4: Popis jednotlivých pinů LCD displaye [9]

Jelikož bude display pracovat ve 4bitovém módu, tak se pro zapojení využije pouze 4 datových vodičů LCD displaye, a to D4–D7. Pro ovládání podsvícení displaye byl anodový vodič připojen přes tlačítko k VCC. Toto zapojení umožňuje zapnutí podsvícení displaye pouze při nastavování, a tím šetří baterii.

3.7 Uložný box pro zařízení

Pro uložení zařízení byla vybrána krabička HAMMOND 1455J1201. Tato krabička má rozměry 120 mm x 78 mm x 27 mm a je vhodná na desky s šířkou 75 mm. Krabička má odnímatelný přední i zadní panel a dokonce i horní díl, do kterého byly vyvrtány a vyřezány otvory pro display a pro destičku, na které jsou umístěny hlavní ovládací prvky celého zařízení, vstupní pin pro synchronizační značky signálu a výstupní konektor pro výstup signálu na externí akustické zařízení. V předním dílu byl vyvrtán otvor pro výstup kabelu USB, přes který se dané zařízení napájí.

3.8 Bateriový zdroj

K napájení celého zařízení je využito power-banky. Tyto baterie mají bateriový zdroj (většinou se jedná o typ lithium-polymerový zdroj) a všechny potřebné obvody (např. stabilizátory napětí) integrované v sobě a lze je připojit přes vyvedený USB kabel. Výstupní proudy se u jednotlivých zařízení liší, ale obecně se pohybují na hodnotě 1 A. Některá dražší zařízení mají výstup i na 2 A. Kapacita integrované baterie se u levnějších zařízeních pohybuje kolem 2000 mAh, u dražších zařízení může být kapacita i přes 10000 mAh. Díky vyvedenému kabelu USB lze zařízení napájet i z běžného USB portu v počítači.

4

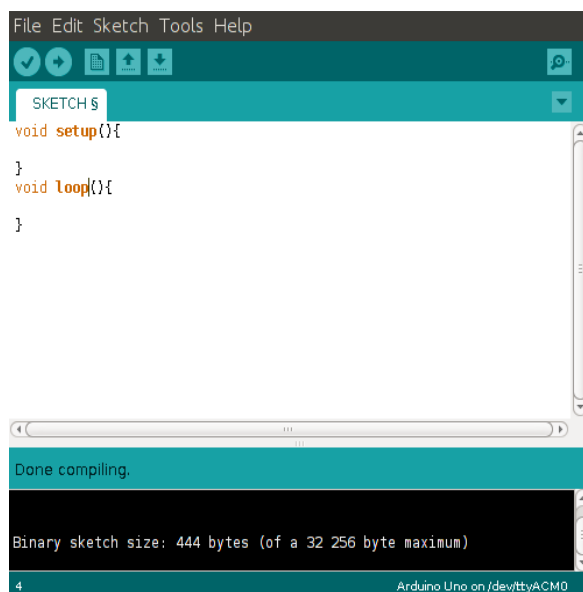
Popis softwarové části zařízení

V této kapitole budou podrobně probrány funkce a knihovny použité v programu, dále pak je zde vysvětlena struktura jazyku WIRING.

4.1 Programovací jazyk WIRING a prostředí programu Arduino IDE

Pro programování mikroprocesoru byl vybrán jazyk WIRING. Tento programovací jazyk vychází z jazyka C/C++ a slouží k programování mikrokontrolérů za použití předem definovaných knihoven, které obsahují mnoho funkcí a maker. Tato struktura umožňuje programovat mikrokontroléry pouze se základní znalostí principu a syntaxe jazyka C/C++. K programování slouží open-source platforma nazvaná Arduino IDE a primárně slouží k programování procesorů od firmy ATMEL. Program Arduino IDE je psaný v programovacím jazyku Java a je možný instalovat na všechny dnes používané operační systémy, ať už jde o Windows, Linux či iOS.

Mezi ovládacími prvky programu, které lze vidět na obrázku 4.1, je tlačítko pro kontrolu kódu, které označí chybnou syntaxi popř. špatné pojmenování proměnných či funkcí, dále pak tlačítko pro nahrávání programu do mikroprocesoru, které zároveň před nahráváním zkompileje daný kód. Pro kompilaci zdrojového kódu používá Arduino IDE dvou kompilérů, a to avr-gcc a avr-dude. Další tři tlačítka slouží k vytvoření nového souboru, k otevření minulého projektu a uložení stávajícího aktivního projektu. Posledním tlačítkem umístěným vpravo se vyvolává konzole sloužící ke čtení dat ze sériové linky. V horní listě záložek můžeme nalézt v záložce file a podsložce examples příklady použití arduina včetně zdrojových kódů s komentáři, které obsahují jak popis zapojení obvodů, tak užitečné informace o použitých funkcích. Další složka obsahuje názvy základních knihoven, které se po kliknutí přidají do výsledného souboru. Další knihovny, např. pro řízení vnitřních časovačů, lze snadno přidávat a rozšířit tak funkčnost zařízení. Velikou nevýhodou tohoto prostředí je nemožnost takzvaného debugging režimu pro testování softwaru mimo mikrokontrolér. Struktura programu je velice podobná struktuře souborů psaných v C/C++.



Obr. 4.1: Arduino IDE [10]

Na první místo se umísťují definice a makra pro další použití, také se zde definují globální proměnné. Místo funkce `main()` jsou v hlavním programu využívány dvě funkce `setup()` a `loop()`. Funkce `setup()` umožňuje nastavení počátečních podmínek mikrokontroléru, jako jsou nastavení vstupně/výstupních bran, nastavení periférií, definování bitové rychlosti sériového přenosu atd., a tato funkce se v programu provede pouze jednou. Naproti tomu funkce `loop()` se opakuje periodicky a je ukončena pouze vnějším vypnutím mikrokontroléru. Mimo tyto funkce lze vkládat vlastní funkce, které není nutné opatřit funkčními prototypy, neboť kompilátor toto provede za nás. Detailní rozbor celého programu je v kapitole 4.3.

4.2 Použité datové typy, funkce a knihovny

Název	Počet bitů	Rozsah	Rozsah unsigned varianta
char	8	-128 - 127	0 - 255
boolean	8	true/false	-
integer	16	-32768 - 32767	0 - 65535
long	32	-2147483648 - 2147483647	0 - 4294967295
float	32	$-3,4 \cdot 10^{38}$ - $3,4 \cdot 10^{38}$	-
double	32(64)	$-1,7 \cdot 10^{308}$ - $1,7 \cdot 10^{308}$	-

Tab. 4.1: Datové typy a jejich rozsah

Mezi použitými datovými typy jsou první čtyři z tabulky č.4.1. Poslední dva typy jsou pro desetinná čísla, která nebyla použita. Typ double je u Arduino desek většinou 32bitový a je shodný s datovým typem float, jenom u některých vyšších typů je dostupná 64bitová varianta.

4.2.1 Funkce a knihovny použité v programu a jejich popis

Název funkce	Počet vstupních parametrů(datové typy)	návratová hodnota
pinMode	2 (int,int)	-
digitalRead	1 (int)	int
digitalWrite	2 (int,int/boolean)	-
delay	1 (int/long)	-
LCD.begin	2 (int,int)	-
LCD.setCursor	2 (int,int)	-
LCD.print	1 (*Char)	-
LCD.clear	0	-
SPI.begin	0	-
SPI.transfer	1	-

Tab. 4.2: Přehled jednotlivých funkcí společně s počtem a typem vstupních parametrů a návratovou hodnotou [10]

U funkce pinMode slouží první parametr k nastavení pinu, jestli se má chovat jako vstup nebo výstup. Tato funkce se nachází nejčastěji ve funkci setup(), ale je možné ji použít i ve funkci loop(), pokud například potřebujeme při běhu programu změnit funkci pinu. Funkce digitalRead čte vstupní digitální data ze vstupního pinu a dává na svém výstupu odpovídající návratovou hodnotu. Podobná funkce této je analogRead, která čte analogová data z AD převodníku. Funkce digitalWrite zapisuje na definovaném pinu logickou hodnotu 1 nebo 0. Jejimi vstupními parametry jsou číslo pinu a hodnota, která má být zapsána na daném výstupním pinu. Tato hodnota je možná zapsat pomocí čísla 0 a 1 nebo pomocí boolovské proměnně true/false. Další funkcí je funkce delay. Tato časovací funkce dokáže pozastavit program na pevně daný čas určený jejím parametrem, který uvádí, na kolik milisekund se má daný program zastavit. Toto je vhodné použít při čekání na nějakou událost, u které je důležité přesné načasování. Problémem této funkce je to, že při jejím běhu se celý program zastaví a není možné například používat externí nastavení proměnných pomocí tlačítka. Funkce začínající LCD slouží k nastavování parametrů LCD displaye a dokáže na něj vypisovat data. První funkce v tabulce 4.2 LCD.begin slouží k inicializování LCD displaye. Její vstupní parametry jsou počet sloupců a počet řádků. Druhá funkce slouží k nastavení kurzoru a má stejné parametry jako předešlá. Třetí funkce vypisuje text v podobě Stringu (pole charů) na display. Poslední funkcí ovládající LCD je

LCD.clear. Tato funkce maže obsah celého displaye. Všechny tyto LCD funkce jsou uloženy v knihovně LiquidCrystal.h a je možné s nimi ovládat displaye s řadičem HD44780. Pro inicializaci a nastavení, které piny mají být užity, je použita třída LiquidCrystal nazevLCD(RS,E,D4,D5,D6,D7), kde jednotlivé parametry třídy odpovídají připojení pinů na LCD displayi. Pro SPI komunikaci s AD9833 je použita knihovna "SPI.h". Tato knihovna má několik užitečných funkcí jako například begin(). Tato funkce inicializuje přenos přes SPI a nemá žádnou návratovou hodnotu ani parametry. Nativně je u arduina použit pin D13 pro synchronizační hodiny, D12 pro MISO, D11 pro MOSI a D10 pro SS. Další funkcí je transfer(). Tato funkce má vstupní parametr typu integer a tento parametr posílá do zařízení. Ukázka kódu pro nastavení AD9833 je v kapitole 4.2.1.1.

4.2.1.1 Kód pro SPI komunikaci s AD9833

V této kapitole je popsána funkce programu, která komunikuje s AD9833, a význam jednotlivých bitů při přenosu viz tabulka 4.3.

Označení bitu	Funkce
D15	Kontrolní bit 1
D14	Kontrolní bit 2
D13	Výběrový bit délky slova frekvence.
D12	Výběrový bit pro výběr MSB nebo LSB
D11	Výběrový bit frekvenčního registru
D10	Výběrový bit fázového registru
D9	Rezervovaný bit. Vždy 0
D8	Reset bit
D7	Výběr spánkového režimu 1
D6	Výběr spánkového režimu 2
D5	Výběr obdélníkového výstupu 1
D4	Rezervovaný bit. Vždy 0
D3	Výběr obdélníkového výstupu 2
D2	Rezervovaný bit. Vždy 0
D1	Výběr obdélníkového výstupu 3
D0	Rezervovaný bit. Vždy 0

Tab. 4.3: Význam jednotlivých bitů při komunikaci s AD9833 [5]

Pokud je kontrolní bit $D15=0$ a $D14=0$, znamená to, že přijde sekvence 14 nastavovacích bitů. V tomto režimu se může nastavovat pomocí bitu D13 délka frekvenčního slova. Při $D13=0$ je použita pouze polovina frekvenčního registru. V tomto režimu je také dostupný bit D12, který vybírá, jestli má být použito vyšších 14 bitů nebo nižších 14 bitů. Při $D13=1$ je zvolen celý rozsah 28 bitového frekvenčního registru a bit D12 je ignorován. D11 souvisí s výběrem frekvenčního registru. Pokud je $D11=0$, potom je zvolen

registr `FREQ0`, jestliže je `D11=1`, tak je vybrán registr `FREQ1`. Bit `D10` slouží k výběru fázového registru a řídí se stejným pravidlem jako bit `D11`. Bit `D8` slouží k restartování obvodu `DDS`. Tento bit nemaže data z registru, pouze dává informaci, že bude načteno nové frekvenční a fázové slovo. Pokud je `D15=0` a `D14=1`, je obvod připraven na přijetí 14 bitů do registru `FREQ0`, které určují frekvenci. Tato operace se musí, při situaci kdy byl bit `D13=1`, opakovat dvakrát pro přijetí jak `MSB`, tak `LSB`. Podobná situace nastává při odeslání `D15=1` a `D14=0`, ale v tomto případě jsou bity uloženy do registru `FREQ1`. Poslední možnost je při `D15=1` a `D14=1`, kdy se nastavuje fázový registr. Následující dvě funkce ukazují zápis do `AD9833`.

```
int FSYNC=10; //nastavení SS na výstup D10
1. void SpiKomunikace(int data)
2. {
3.   digitalWrite(FSYNC,0);           // nastav výstup D10
4.   SPI.transfer(highByte(data));    // zapiš první byte dat
5.   SPI.transfer(lowByte(data));    // zapiš druhý byte dat
6.   digitalWrite(FSYNC,1);         // ukonči přenos dat
7. }
```

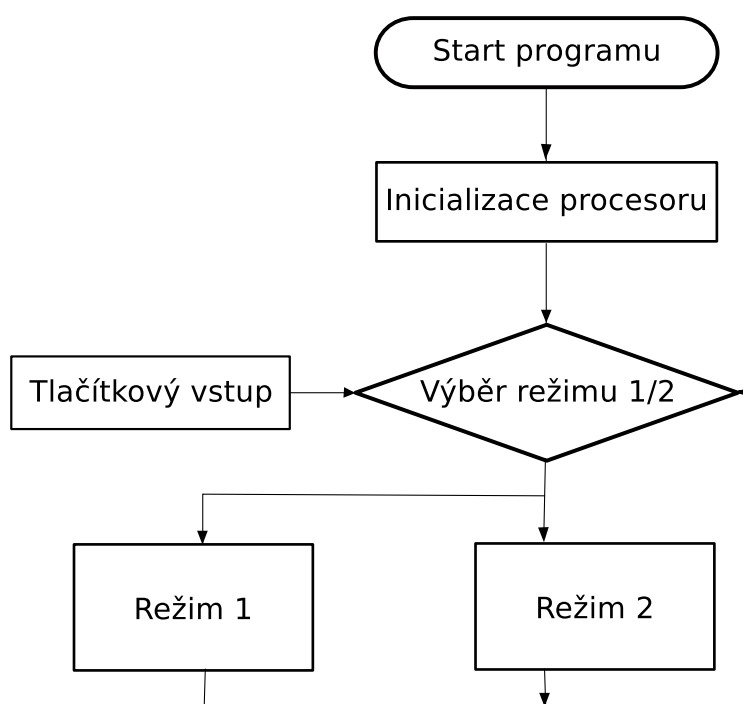
Tato funkce nastaví na výstupu `D10` logickou hodnotu 0. Tím dá nejevo obvodu `AD9833`, že bude vysílat data. Poté se pošlou data po jednom bytu a přenos se ukončí při nastavení `D10` do logické 0.

```
1. void zapisData(long data)
2. {
3.   SpiKomunikace(0x2100); // 0x2100 = 0b 0010 0001 0000 0000
4.   SpiKomunikace(LSB(data));
5.   SpiKomunikace(MSB(data));
6.   SpiKomunikace(0xC000); // 0xC000 = 0b 1100 0000 0000 0000
7.   SpiKomunikace(0x2000); // 0x2000 = 0b 0010 0000 0000 0000
8. }
```

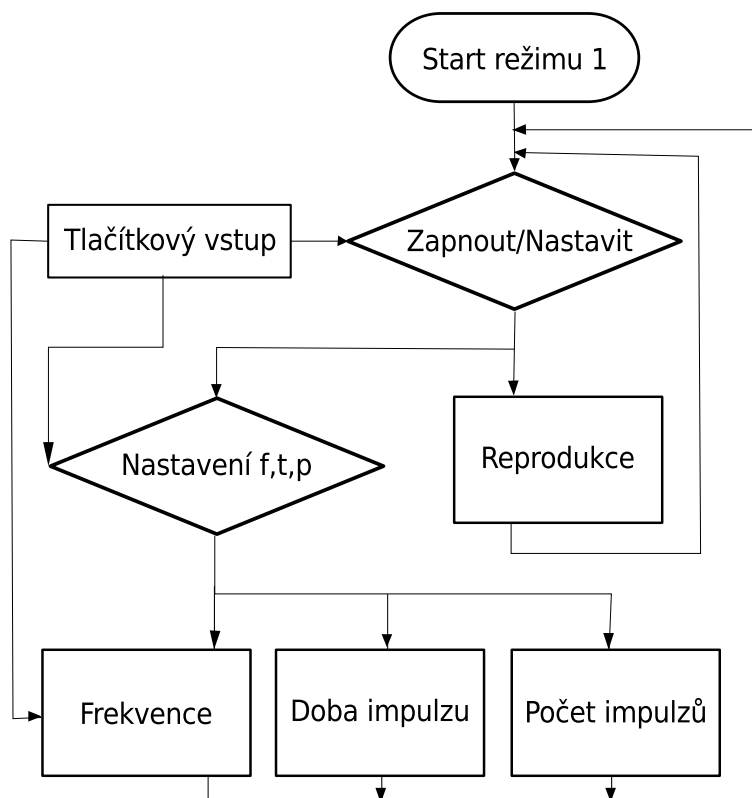
V první kroku zapsaném na řádce číslo 3 je nutné restartovat `AD9833`. K tomu slouží nastavení `D8` na hodnotu 1. Nastavením `D13=1` docílíme toho, že obvod čeká na příjem jak `MSB` tak `LSB`. V druhém kroku jsou poslána data pro registr. První část obsahuje funkci `LSB`, která vrací hodnotu nižších bitů dat, včetně úpravy prvních dvou bitů `D15` a `D14` na hodnoty 0 a 1. Druhá funkce `MSB` vrací vyšší bity dat, včetně již zmíněné úpravy. Ve třetím kroku na řádce 6 je poslán údaj o fázovém posunu, který je v tomto případě 0. Posledním krokem je poslání bitové sekvence, která spustí obvod.

4.3 Popis program

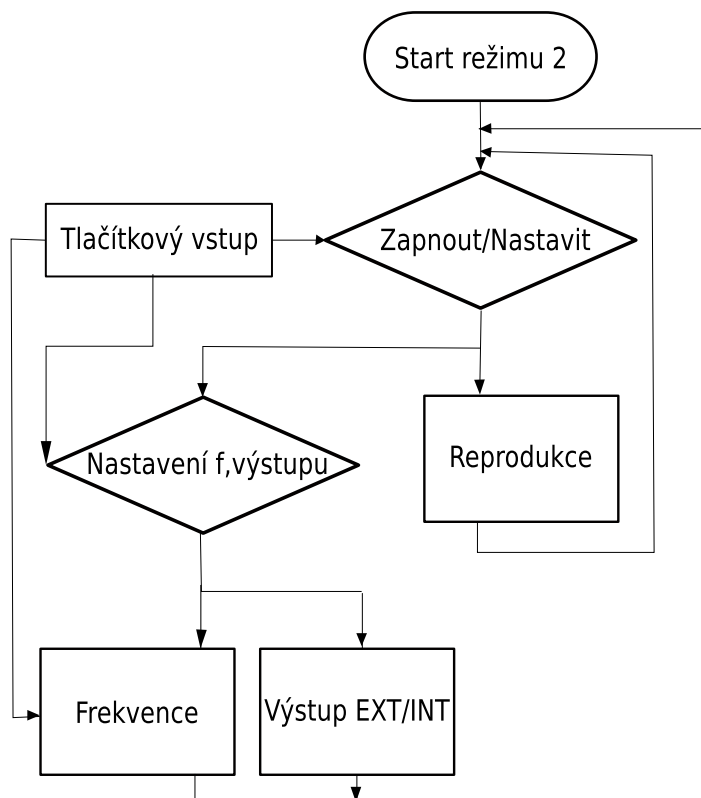
Jak již bylo řečeno v kapitole 4.1, program se skládá z několika částí. První část obsahuje definice konstant, makra a globální proměnné. Druhá část inicializuje procesor do počátečního stavu po restartu. Další částí programu jsou funkce `loop()` a vlastní funkce. Vývojový diagram na obrázku 4.2 ukazuje hlavní princip programu. Po zapnutí napájení se procesor inicializuje pomocí funkce `setup()`. Následně přeskočí do funkce `loop()`, kde se pomocí tlačítka vybírá režim generátoru. Tato smyčka skončí pouze v případě vnějšího zásahu (např. vypnutím napájení). V režimu 1 je možné pomocí tlačítkových vstupů měnit parametry výstupního signálu nebo zapnout jeho reprodukci. Při nastavování parametrů je možné měnit všechny parametry výstupního signálu. Při každém průchodu smyčkou je kontrolováno, jestli nedošlo ke změně režimu (viz obrázek 4.3). Režim 2 je podobný režimu 1. Rozdílem je, že je možné nastavit pouze frekvenci a výstupní zařízení (viz obrázek 4.4).



Obr. 4.2: Vývojový diagram hlavní části programu



Obr. 4.3: Vývojový diagram pro režim 1



Obr. 4.4: Vývojový diagram pro režim 2

5

Závěr

Cílem této bakalářské práce bylo vytvořit číslicově řízený akustický generátor s proměnnými parametry a s možností různých režimů práce. Mezi proměnné parametry zařízení patří frekvence, délka trvání jednotlivých impulzů a jejich počet. V jednotlivých kapitolách této bakalářské práce je popsán návrh a realizace takového generátoru.

Kapitola 2 je spíše teoretická a je zde vysvětleno, jaké způsoby generace sinusového signálu s proměnnou frekvencí jsou na dnešním trhu, včetně jejich úskalí a možnosti propojení s číslicovou technikou od nejméně vhodných, jako je Wienův oscilátor, až po metody vhodné, jako je fázový závěs a obvody přímé digitální syntézy.

V následující kapitole 3 jsou vyjmenovány a popsány jednotlivé obvody tvořící výsledné zařízení. U většiny obvodů jsou také zapsány důležité informace o spotřebě obvodů a jejich možném propojení. Nejdůležitějšími částmi výsledného generátoru jsou bezpochyby mikroprocesor a obvod generující sinusový signál. Pro tuto generaci byl vybrán obvod DDS, a to kvůli jeho snadnému propojení s číslicovými systémy, nízké spotřebě (která je z důvodu bateriového napájení důležitá) a také kvůli jeho vysoké přesnosti a stabilitě generovaného kmitočtu, která se i přes využití pomalých referenčních hodin (1 MHz) pohybuje v řádu jednotek až desetin Hz. Napájení generátoru nebylo vytvořeno pomocí klasických baterií, ale bylo použito tzv. power-banky. Tento zdroj energie byl vybrán kvůli tomu, že má již veškeré důležité komponenty (např. stabilizátory, které jsou důležité pro většinu číslicových obvodů) integrované v sobě a není nutné je přidávat.

V poslední kapitole 4 jsou vysvětleny hlavní konstrukce programu, použité funkce a knihovny a také je zde popis programovacího jazyku WIRING.

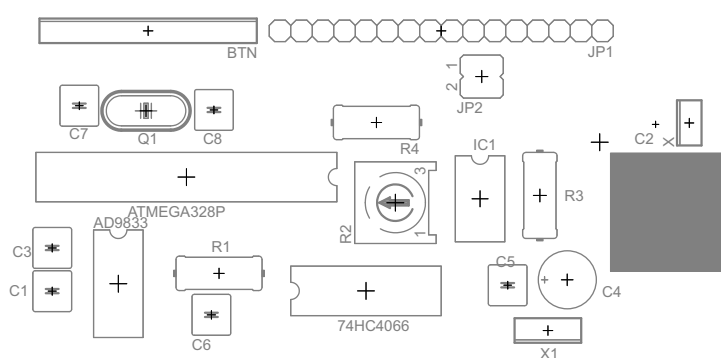
Literatura

- [1] Doleček, Jaroslav *Moderní učebnice elektroniky. 6. díl, Kmitočtové filtry, generátory signálu a převodníky dat. 1. vyd* Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-240-4.
- [2] Pinker, J., Koucký, V. *Analogové elektronické systémy. 1. a 2. část. 4. vyd.* Plzeň: Západočeská univerzita, 2010. 2 sv. ISBN 978-80-7043-917-3.
- [3] ŠKUTKOVÁ, H. *Akustický generátor pro buzení evokovaných potenciálů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 76 s. Vedoucí diplomové práce doc. Ing. Milan Chmelař, CSc.
- [4] *Andrea Ronešová - Úvod* [online]. 2013 [cit. 2015-05-25]. Dostupné z: http://home.zcu.cz/ronesova/index.php?menuitem=aes_mwg
- [5] *Analog Devices — Mixed-signal and digital signal processing ICs* [online]. 2012 [cit. 2015-05-25]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9833.pdf>
- [6] *Atmel Corporation - Microcontrollers, 32-bit, and touch solutions* [online]. 2012 [cit. 2015-05-25]. Dostupné z: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
- [7] *Embedded Processing, Semiconductor Company, Texas Instruments - TI.com* [online]. 2000 [cit. 2015-05-25]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm386.pdf>
- [8] *Home :: NXP Semiconductors* [online]. 2013 [cit. 2015-05-25]. Dostupné z: http://www.nxp.com/documents/data_sheet/74HC_HCT4066.pdf
- [9] *Adafruit Industries, Unique & fun DIY electronics and kits* [online]. 2009 [cit. 2015-05-25]. Dostupné z: <https://www.adafruit.com/datasheets/TC1602A-01T.pdf>
- [10] *Arduino* [online]. 2015 [cit. 2015-05-25]. Dostupné z: <http://www.arduino.cc/>
- [11] *Farnell element14 Česká republika - Distributor elektronických součástek* [online]. 2015 [cit. 2015-05-25]. Dostupné z: <http://www.farnell.com/datasheets/1653465.pdf>

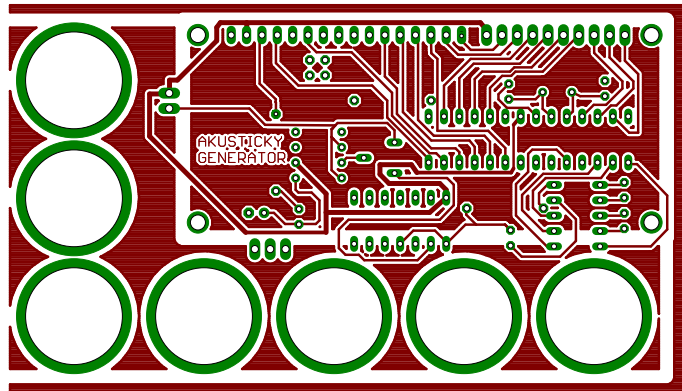
Příloha A

Deska plošných spojů a schéma navrženého akustického generátoru

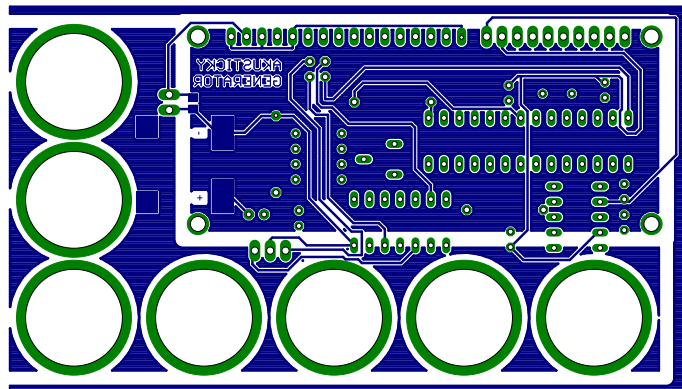
A.1 Deska plošných spojů testovací verze



Obr. A.1: DPS: Osazovací výkres vrstvy BOTTOM

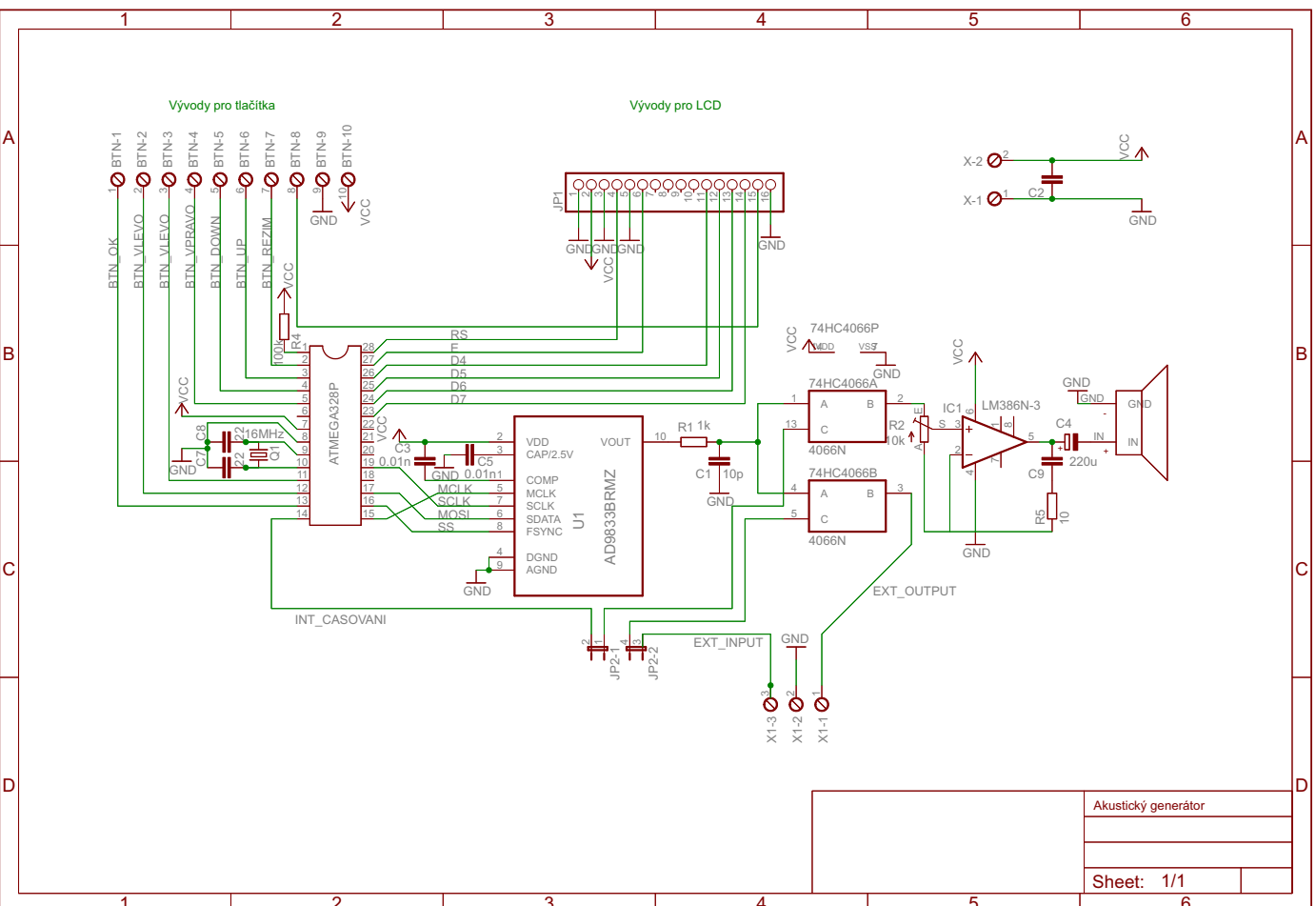


Obr. A.2: DPS: Vrstva TOP



Obr. A.3: DPS: Vrstva BOT

A.2 Schéma zapojení



Obr. A.4: Schéma zapojení akustického generátoru