



Středoškolská technika 2019

Setkání a prezentace prací středoškolských studentů na ČVUT

Návrh a výroba generátoru periodických signálů

Matěj Turinský

VOŠ.SŠ, COP

Dukelská 421, Sezimovo Ústí,

Prohlášení

Prohlašuji tímto, že jsem maturitní projekt vypracoval samostatně pod vedením pana učitele Ing. Jiřího Bumby a uvedl jsem veškerou použitou literaturu a další informační zdroje včetně internetu.

V Sezimově Ústí dne

.....

Poděkování

Maturitní práce byla zpracována jako závěrečná práce v rámci řádného ukončení studijního oboru – Počítače a robotika. Vedoucím této mé práce byl pan učitel Ing. Jiří Bumba, kterému tímto děkuji za odbornou pomoc při řešení zadání.

Také tímto děkuji ostatním pedagogům, kteří mi s prací pomáhali po stránce technické, jazykové a stylistické. Dále jsem vděčný škole a spolku KRA Písek, kteří mi poskytli možnost užívat její vybavení laboratoří k návrhu a realizaci tohoto projektu a spolužákům za pomoc při realizaci.

Anotace

Tato práce je zaměřena na praktickou realizaci funkčního generátoru. V úvodu práce jsou představeny způsoby generování periodických signálů. Blíže se práce zabývá digitálními metodami. Dále je v práci popsán hardware a software generátoru. V závěru je sestavený funkční generátor podroben testu funkčnosti.

Generátor je sestaven okolo integrovaného obvodu AD9833, jehož výstupní signál je upravován operačními zesilovači. Hlavním výstupem je sinusový, trojúhelníkový nebo obdélníkový analogový signál s nastavitelným kmitočtem, amplitudou a stejnosměrným posunem. Druhým výstupem je digitální signál pro obvody TTL nebo CMOS s nastavitelnou střídou.

Klíčová slova

Funkční generátor, generování periodických signálů, přímá digitální syntéza

Annotation

The focus of this work is on the practical realization of function generator. The basic ways of generating periodical signals at the beginning. The work closely follows digital methods. The generator's hardware and software are described next. Constructed function generator is tested in the end.

Generator is compiled around integrated circuit AD9833, whose output signal is adjusted by operation amplifiers. Main output is sinusoidal, triangular or rectangular analog signal with adjustable frequency, amplitude and DC shift. Second output is digital signal for TTL or CMOS circuits with adjustable alternate.

Keywords

Function generator, generating periodical signals, direct digital synthesis

Seznam zkratk a symbolů

CMOS – complementary metal-oxide-semiconductor; technologie využívaná pro výrobu IO

DAC – digital-analog convertor; digitálně-analogový převodník

DAS – direct analog synthesis; přímá analogová syntéza

DDS – direct digital synthesis; přímá digitální syntéza

DPS – deska plošných spojů

IO – integrovaný obvod

LCD – liquid crystal display; display na bázi tekutých krystalů

OZ – operační zesilovač

PPL – phase locked loop; fázový závěs

RISC – reduced instruction set computer; počítač s redukovanou sadou instrukcí

TTL – transistor-transistor-logic; tranzistorově-tranzistorová logika

VD – vývojový diagram

μP – jednočipový mikropočítač

| | |
|-------|--|
| Obsah | |
| 1 | Úvod.....7 |
| 2 | Generování periodických signálů8 |
| 2.1 | Druhy signálů8 |
| 2.2 | Způsoby generování periodických signálů.....9 |
| 2.2.1 | Analogové způsoby generování signálů9 |
| 2.2.2 | Digitální způsoby generování signálu.....12 |
| 3 | Analýza zadání a návrh řešení14 |
| 3.1 | Možnosti řešení funkčního generátoru.....14 |
| 3.2 | Zdůvodnění výběru daného řešení14 |
| 4 | Hardware generátoru.....15 |
| 4.1 | Integrovaný obvod AD9833.....15 |
| 4.2 | Napájecí obvody.....15 |
| 4.3 | Řídící jednotka16 |
| 4.4 | Ovládací a zobrazovací obvody17 |
| 4.4.1 | Ovládací prvky.....17 |
| 4.4.2 | Displej.....18 |
| 4.5 | Výstupní obvody19 |
| 4.5.1 | Úprava signálu pomocí OZ.....19 |
| 4.5.2 | Výstup pro TTL/CMOS.....20 |
| 4.5.3 | Koncový stupeň21 |
| 5 | Software generátoru21 |
| 5.1 | Arduino IDE.....22 |
| 5.2 | Obsluha uživatelského rozhraní23 |
| 5.3 | Komunikace mezi řídicí jednotkou a ostatními bloky24 |
| 6 | Stavba a zprovoznění generátoru25 |
| 6.1 | Stavba funkčního generátoru.....25 |
| 6.2 | Test funkčnosti26 |
| 7 | Závěr27 |
| 8 | Použitá literatura28 |
| 9 | Použité programy28 |
| 10 | Seznam obrázků.....28 |
| 11 | Seznam příloh29 |
| 12 | Seznam příloh na CD.....29 |

1 ÚVOD

Poslední dobou se stále více zajímám o elektroniku a tvorbou elektronických zapojení, zejména mě zajímá oblast audio zesilovačů. K jejich tvorbě je potřeba nejen náradí pro výrobu a součástky, ale také přístroje k měření a diagnostice. Některé věci již ve své dílničce mám, ale jiné mi stále chybí, jako například funkční generátor. A hlavně proto jsem se rozhodl pro jeho tvorbu.

Funkční generátoru je elektronické zařízení určené pro generování napět'ových signálů s požadovanými tvary a kmitočty. Hlavní užití těchto zařízení je v diagnostice, vývoji popřípadě analýze elektronických obvodů pracujících s elektrickými veličinami jako zdrojem signálu (např. právě práce se zesilovači nebo bezdrátový přenos nf i vf signálů).

Hlavním cílem práce je tedy návrh a stavba funkčního generátoru, jeho zprovoznění a tvorba podkladů pro jeho výrobu. Dalším důvodem, proč jsem se pustil do stavby generátoru, je snaha se zdokonalit v programování, v práci s operačními zesilovači a ve vývoji elektronických zapojení a zařízení.

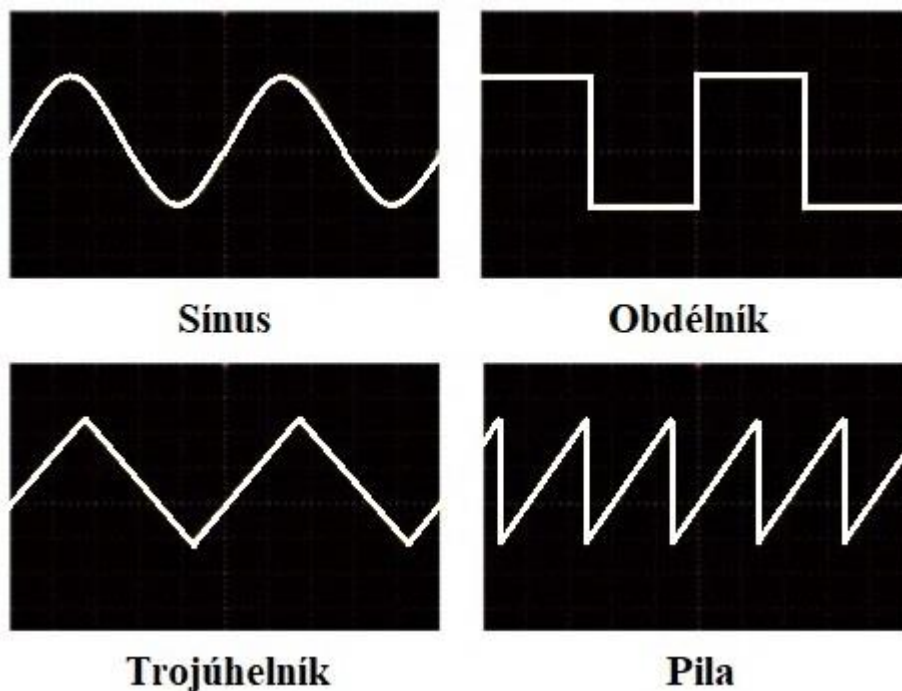
2 GENEROVÁNÍ PERIODICKÝCH SIGNÁLŮ

2.1 Druhy signálů

Základem generování periodických signálů je nutnost vědět, co to periodické signály vlastně jsou. Nejjednodušší způsob dělení signálů je dělení na stejnosměrné a střídavé signály.

Stejnoseměrné signály jsou takové, které nemění svou polaritu. Typickým zdrojem těchto signálů je stejnosměrný napájecí zdroj (napěťový či proudový). Dalšími stejnosměrnými zdroji může být například dynamo nebo pouhé usměrnění napětí rozvodné sítě.

Střídavé signály oproti stejnosměrným svou polaritu mění. Podle toho, jak svou polaritu a tvar mění, se dají dále rozdělit na signály periodické, které jsou pro nás stěžejní, náhodné a pseudonáhodné. Náhodné signály jsou takové, jejichž průběh nemůžeme nijak popsat či očekávat. Nejčastěji je známe v podobě rušení. Dále pak signály pseudonáhodné. Ty se jako náhodné pouze jeví a ve skutečnosti se jedná o signály periodické s periodou, která je buď delší, nebo srovnatelně dlouhá s dobou měření.

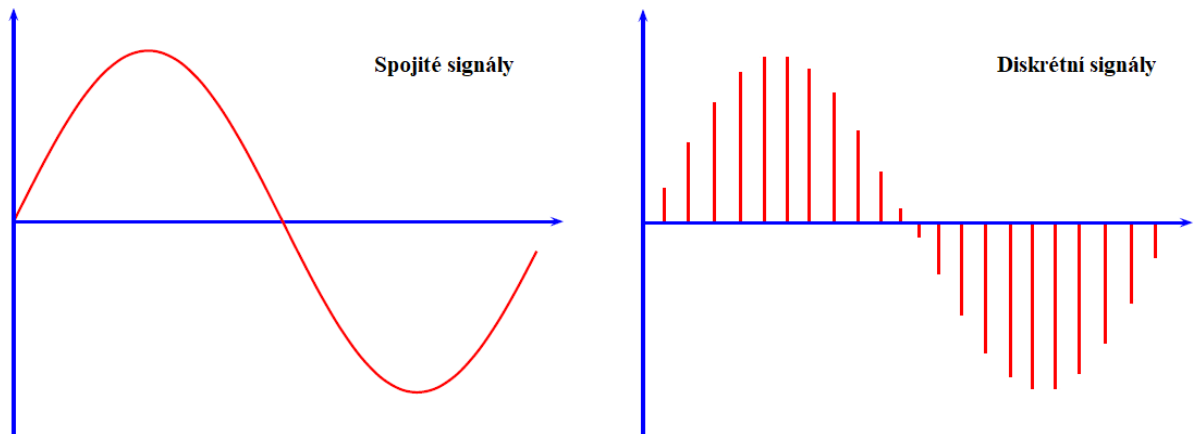


obr. 2-1: Periodické signály

Nyní se dostáváme k pro nás nejpodstatnější skupině střídavých signálů. Signály periodické jsou takové signály, které s určitou periodou opakují svůj průběh a dají se často popsat matematickou funkcí. Mezi periodické signály typicky řadíme tyto: sinusový, obdélníkový a trojúhelníkový (nebo také pilový) průběh. Ty jsou zde znázorněny na obrázku obr. 2-1: Periodické signály.

2.2 Způsoby generování periodických signálů

Způsobů, jak vytvořit periodický signál, je hned několik. Důležité pro rozdělení způsobů generování signálů je to, jak se signálem pracujeme. Je-li signál spojitý a jeho spojitost není v průběhu tvorby signálu porušena, jedná se o cestu analogové tvorby signálu. Pokud však na signál začneme nahlížet jako na nespojitý (tyto signály se také často nazývají diskrétní) a vytváříme jej pomocí napět'ových úrovní a podobně, začne jít o digitální způsob generování signálu. Oba způsoby mají své klady i zápory, se kterými je nutno počítat.

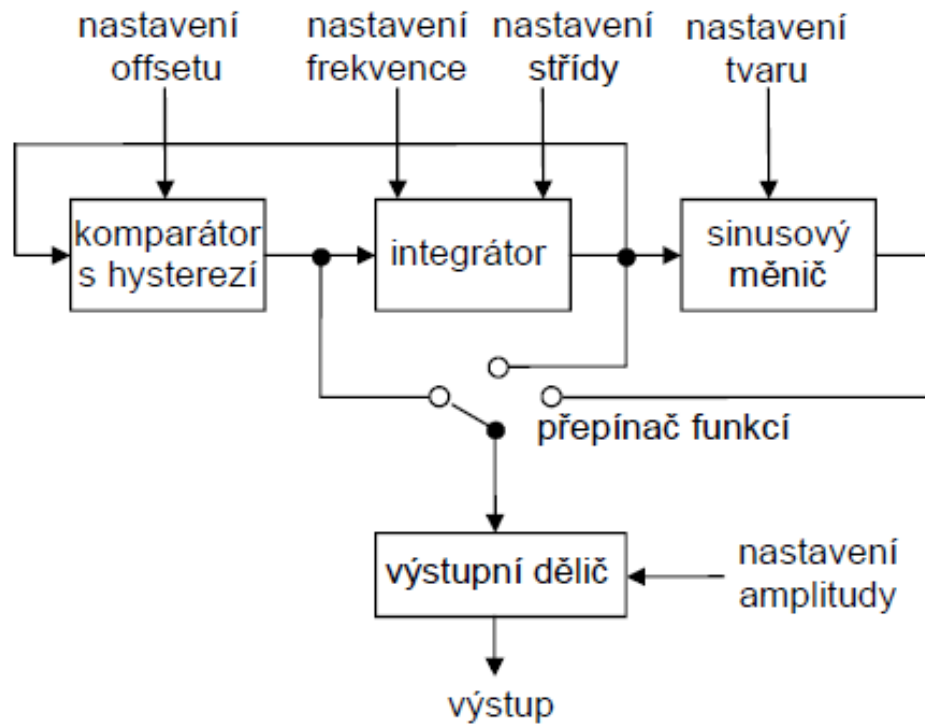


obr. 2-2: Spojité a diskrétní signály

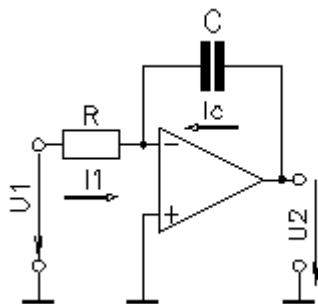
2.2.1 Analogové způsoby generování signálů

Nejjednodušším druhem funkčních generátorů jsou tzv. analogové oscilátory. Ty se skládají z LC oscilátoru a generátoru obdélníkového a pilového průběhu. Jsou vytvořeny ze základních elektronických součástek a OZ, jejichž zapojení poté ovlivňuje tvar a amplitudu. Jako LC oscilátor se používá Hartleyův nebo Colpittsův oscilátor. Pro méně kvalitní oscilátory se dá použít RC oscilátor.

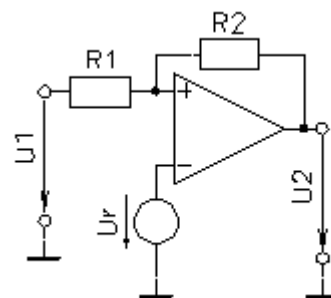
Možností zapojení pro generování obdélníkového a pilového signálu je více, ale nejčastěji se používá koncepce typu komparátor-integrátor spojené do zpětnovazební smyčky. První OZ (komparátor) je zapojen jako neinvertující zesilovač s hysterézí a druhý OZ (integrátor) jako invertující integrátor. [1] Tato zapojení jsou znázorněna na obrázcích obr. 2-3 a obr. 2-4. Obrázek obr. 2-5 pak zobrazuje blokové schéma analogového oscilátoru s tím, že se ve většině případů zapojení se vynechává nastavení/regulace střídý.



obr. 2-5: Blokové schéma analogového oscilátoru [1]



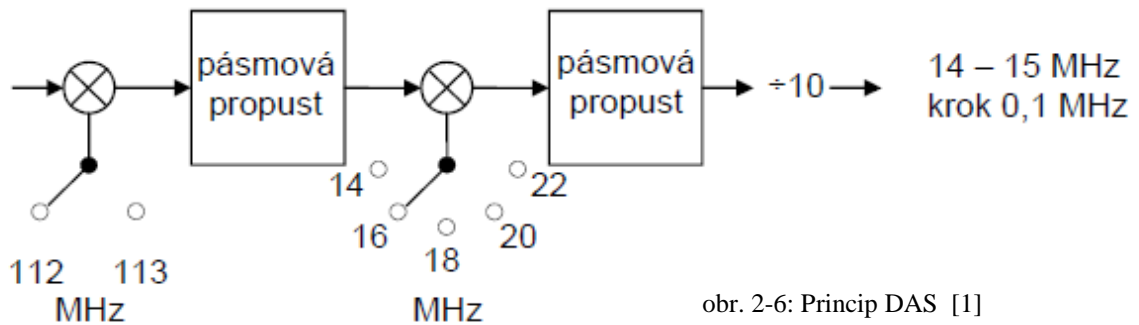
obr. 2-3: Invertující integrátor



obr. 2-4: Neinvertující komparátor s hysterézí

Výhoda těchto generátorů signálu je jednoduchost zapojení, a proto i nízké výrobní náklady. Zapojení má samozřejmě také své nevýhody, a to poměrně velká tolerance, nemožnost číslicového řízení a nestálost parametrů pasivních součástek v čase (popř. se změnou teploty).

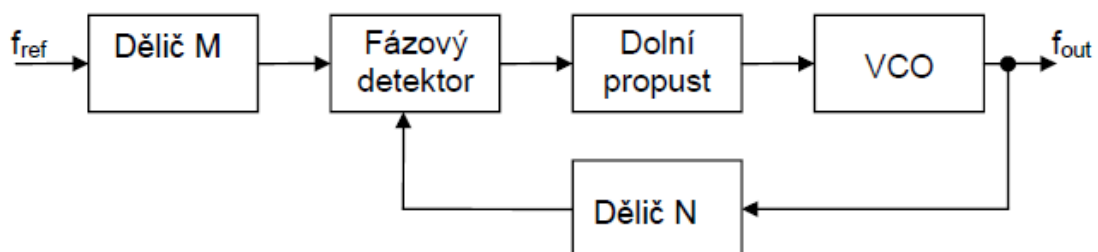
Druhým z analogových způsobů vytváření periodických signálů je DAS. Tato metoda se také nazývá filtrování či směsování signálu. Princip tohoto způsobu je založen upravování kmitočtu signálu pomocí základních matematických operací (násobení, sčítání, ... kmitočtu). Proces je nazýván přímým, protože je vynechán proces korekce chyb, takže kvalita výstupního signálu přímo souvisí s kvalitou vstupního signálu. [1]



obr. 2-6: Princip DAS [1]

Podle obrázku obr. 2-6 výše se dá říci, že přímá analogová syntéza pracuje jako přepínatelná skupina oscilátorů. Šum je minimální (většinou způsoben hlavně přenosem), proto je výsledný signál velmi kvalitní, předpokládáme-li, že vstupní signál je také kvalitní. Nevýhodou těchto generátorů jsou velké výrobní náklady (kvůli velkému počtu oscilátorů) a velký krok mezi změnami kmitočtu.

Posledním z analogových způsobů je nepřímá syntéza, také známá jako PPL. Tato metoda využívá oscilátoru řízeného napětím (VCO – Voltage Controlled Oscillator). Signál na výstupu tohoto oscilátoru je porovnáván se signálem referenčním. Odchytkami těchto signálů je tento oscilátor zpětně řízen. Zjišťování odchylek a řízení oscilátoru se děje ve fázovém detektoru. Regulace frekvence je zajištěna nastavitelným děličem, který dělí frekvenci signálu přicházejícího do fázového detektoru, čili výstupní frekvence se v daném poměru násobí. Pro lepší regulaci je možno za referenční oscilátor zařadit ještě jeden dělič kmitočtu. Výsledná frekvence výstupního signálu je pak dána vztahem: $f_{out} = f_{ref} (N/M)$ [Hz, Hz, -, -]. [1]

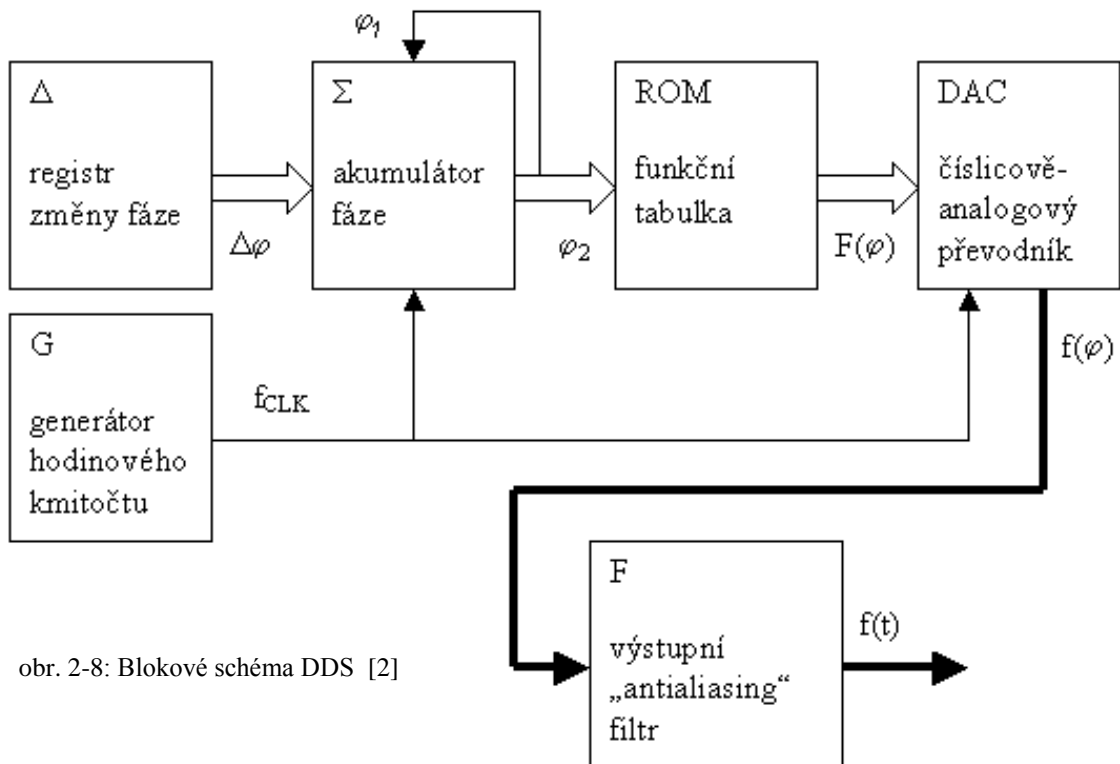


obr. 2-7: Blokové schéma PPL [1]

2.2.2 Digitální způsoby generování signálu

Jak jsem se již zmínil v úvodu kapitoly, druhým ze způsobů vytváření periodických signálů je digitální cesta. Všechny postupy tvorby periodických signálů digitální cestou vychází z DDS. V poslední době tato možnost převažuje nad všemi ostatními, přitom je princip DDS v základu dosti jednoduchý. Oproti výše uvedeným způsobům, které mají základ v oscilátoru a následném tvarování a úpravě signálu ve směšovačích a tvarovačích, je DDS výjimečná tím, že signál je zde vytvářen číslicově.

Pro snazší orientaci vkládám obrázek obr. 2-8, na kterém bude dále vysvětlen princip DDS.

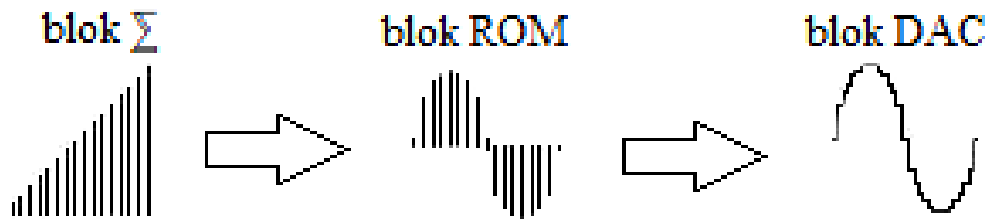


obr. 2-8: Blokové schéma DDS [2]

Základem všech číslicových zařízení je generátor hodinových impulzů. Ani DDS není výjimka a v blokovém diagramu je znázorněn blokem G (*generátor hodinového kmitočtu*) s výstupem f_{CLK} .

Prvním funkčním blokem, pro celou DDS, je blok Σ (*akumulátor fáze*). Jedná se v podstatě o čítač, do kterého se přičítá konstantní hodnota uložená v registru změny fáze (blok Δ *registr změny fáze*). Ve fázovém registru, který právě s časovačem dává dohromady akumulátor fáze, tedy lineárně roste hodnota a dá se vyjádřit funkcí, jejíž strmost je přímo závislá na hodnotě uložené v registru změny fáze. Hodnota ve fázovém registru se při přetečení vynuluje a čítač čítá opět od začátku.

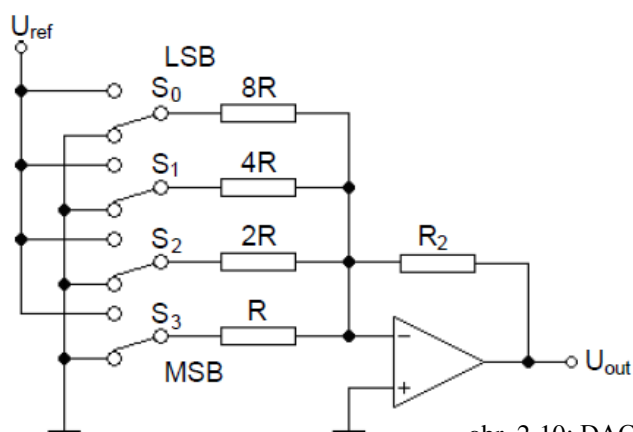
Nyní máme lineárně rostoucí průběh. Ten dále pokračuje do bloku ROM (*funkční tabulka*). Zde jsou v paměti (nejčastěji typu ROM) uloženy průběhy jednotlivých signálů. Každému bodu lineárního průběhu z bloku Σ je přidělen bod odpovídající požadovanému tvaru signálu. Jakýkoli tvar tedy vytvoříme tím, že jeho průběh popíšeme v tabulce. A to je velká výhoda oproti ostatním způsobům generování periodických signálů. Kmitočet signálu je poté volen velikostí konstanty (směrnice růstu lineárního signálu) uložené v registru změny fáze. Máme tedy přesný číslicově řízený funkční generátor. Průběhy signálu v jednotlivých blocích, které jsou popsány v tomto odstavci, jsou naznačeny na obrázku obr. 2-9.



obr. 2-9: Průběhy signálu v DDS

Posledním blokem je blok F (*výstupní „antialiasing“ filtr*). Tento blok je důležitý, ale nemá přímý dopad na funkci DDS. Jeho účelem je odfiltrovat hrany jednotlivých vzorkovacích úrovní. Jedná se o dolní propust se strmým spádem, která odfiltrovává vysoké kmitočty vytvořené právě těmito hranami. Výstupní signál je tedy potom hladší a důvěryhodněji vystihuje požadovaný tvar (například sinusový průběh).

Blok F sice signál odfiltruje od vyšších harmonických a tak signál vyhladí, ale nejprve je třeba tento signál nějak vytvořit. K převodu číslicového vyjádření signálu na signál analogový je určen blok DAC (*číslicově-analogový převodník*). Tento blok může být realizován mnoha způsoby, například pomocí odporové sítě R2R nebo pomocí zapojení s váhovými odpory v síti OZ (viz obr. 2-10).



obr. 2-10: DAC s váhovými odpory

Je tedy jasné, že v dnešní době naprosto převažují přístroje a zapojení na bázi DDS, a to nejen ve spojení s funkčními generátory, ale také jako rozmítače signálu nebo násobiče hodinových kmitočtů a další. [2]

3 ANALÝZA ZADÁNÍ A NÁVRH ŘEŠENÍ

Tato kapitola popisuje možnosti realizace funkčního generátoru a následně zdůvodní výběr řešení pro jeho tvorbu.

3.1 Možnosti řešení funkčního generátoru

Za cíl jsem si stanovil vytvořit generátor, který bude schopen konkurovat komerčně řešeným (zvoleným normálem byl funkční generátor GFG-8015G) s tím rozdílem, že moje řešení bude tzv. na míru. Tím je vyloučena možnost koupit již hotový funkční generátor.

Prvním požadavkem na generátor byl kmitočtový rozsah 1Hz až 1MHz. Dalším parametrem byly požadované tvary výstupních signálů. Zde jsem se rozhodl pro tři základní a to sinusový průběh, trojúhelníkový průběh a obdélníkový průběh. Posledním ze základních požadavků byla amplituda nezatíženého výstupního signálu $\pm 10V$. Dále se na výsledný generátor nabalovaly další funkce, jako například výstup pro TTL a CMOS obvody nebo regulace střídy.

Těmto požadavkům by vyhovovaly všechny druhy funkčních generátorů, ať už analogové či digitální. Proto jsem se musel rozhodnout mezi níže uvedenými možnostmi.

První možností je vytvořit celý generátor. Toto řešení by bylo zapojeno buď jako analogový oscilátor, nebo by se jednalo o zapojení s PPL. Obě řešení jsou realizovatelná, ale problém by nastal při dosažení parametrů komerčních zapojení. O možnosti vytvářet zapojení s přímou analogovou syntézou zde vůbec neuvažuji kvůli vysokým nákladům na realizaci.

Druhá možnost je pokusit se o vytvoření číslicového funkčního generátoru. Ten by pracoval jako číslicový syntetizér s vnějším či vnitřním oscilátorem. Toto zapojení by již snadno dosáhlo na parametry komerčně dostupných funkčních generátorů. Jeho druhou velkou výhodou je snadná a přesná nastavitelnost požadovaných hodnot, a dokonce možnost programování vlastních průběhů signálu.

Třetí možnost je pracovat s již vyrobeným IO, který je schopen generovat periodické signály. IO by samozřejmě nepracoval sám o sobě, ale vyžadoval by ke své funkci další obvodové části. Ty by pak ovlivňovaly parametry funkčního generátoru, a proto by se stále jednalo o řešení „na míru“, jak jsem si v úvodu jasně stanovil.

3.2 Zdůvodnění výběru daného řešení

Pro finální realizaci jsem si vybral třetí možnost. Důvod byl jasný. Analogové funkční generátory, jak jsem při vyhledávání informací zjistil, se již používají méně často, protože jsou výrobně složitější. Proto jsem je z výběru vyřadil. Druhá možnost byla vytvoření vlastního číslicového funkčního generátoru. To mi ovšem nedovolil čas. Proto jsem se rozhodl použít již existující IO pro generování periodických signálů a nadále s ním pracovat.

4 HARDWARE GENERÁTORU

V této kapitole popisují hlavní části hardwaru mého řešení funkčního generátoru, pro které jsem se rozhodl poté, co byly stanoveny veškeré požadované parametry i způsob generování signálu. Blokové schéma zapojení je obsaženo v příloze A.

Zapojení lze rozdělit do čtyř funkčních částí. Napájení všech bloků zajišťuje napájecí zdroj. Hlavní částí generátoru je blok řídicích obvodů, který obsahuje vlastní destičku generátoru a řídicí mikropočítač. Signál je dále zpracováván v analogovém zesilovači, na který navazují koncové obvody. Dále jsou na blokovém schéma ovládací a zobrazovací obvody.

4.1 Integrovaný obvod AD9833

Jak už jsem se zmínil v kapitole 3.2, rozhodl jsem se pro použití IO, který bude generovat periodický signál. Na trhu je větší množství IO splňujících tento požadavek. Já jsem se rozhodl pro AD9833. Ten byl totiž velmi často a kladně citovaný na fórech zabývajících se tvorbou hardwaru (*např.: tvorba.hw.cz*) a navíc existuje i ve formě tzv. shieldu. Ten můžete vidět na obrázku obr. 4-1.

V příloze 7 přikládám oscilogramy signálu přímo generovaného pomocí AD9833 v porovnání se signálem generovaným pomocí GFG-8015G (normál pro tvorbu funkčního generátoru).

Technické parametry AD9833 (viz data sheet produktu): [3]



obr. 4-1: AD9833 shield

- Kmitočtový rozsah: 0 Hz až 12,5 MHz
- Generované průběhy: sinus, obdélník, trojúhelník
- Způsoby ovládání: digitální - kmitočtu a fáze
- Způsob komunikace: tří-vodičová SPI komunikace
- Napájecí napětí: +2,3 až +5,5 V
- Počet kvantizačních stupňů: 28 bitové rozlišení v rozsahu 0,1 Hz až 25 MHz generátor hodinových kmitočtů

4.2 Napájecí obvody

Stejně tak jako je více způsobů pro tvorbu periodických signálů, je i více způsobů realizace napájecích zdrojů. Kvůli možnému rušení jsem vypustil možnost užití spínaného zdroje. Jako druhou možnost jsem zvažoval užití lineárních stabilizátorů 78xx a 79xx. Ty jsem nakonec využil pro své zapojení.

V celém zapojení budou užity tři napájecí napětí. Pro napájení shieldu s AD9833, řídicí jednotku a další vedlejší obvodové částí jsem zvolil napětí +5V. Pro napájení operačních zesilovačů potom symetrické napětí $\pm 12V$.

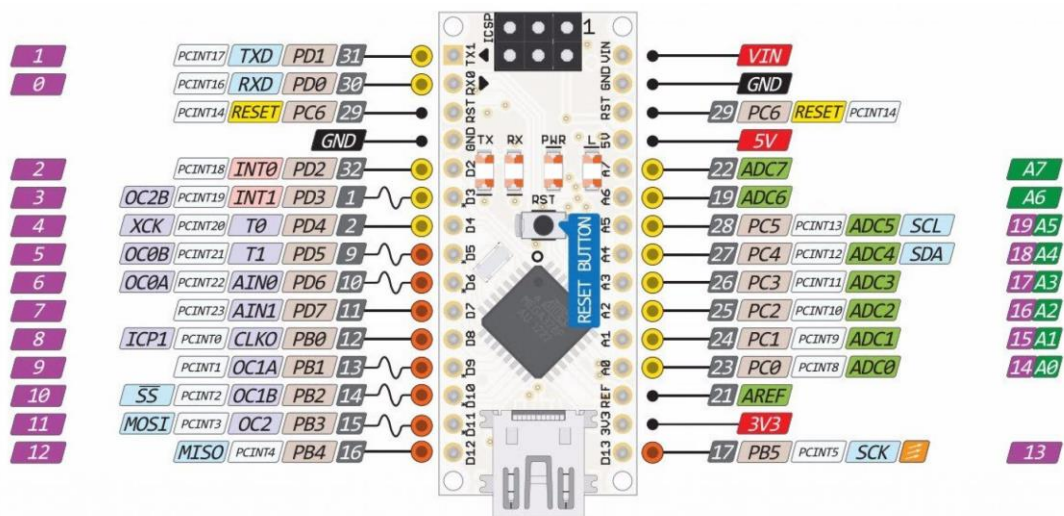
4.3 Řídicí jednotka

Jako řídicí jednotku jsem zvolil vývojovou desku Arduino. Základem této desky je mikrokontroler Atmega328P. Použitý μP je osmibitový mikrokontroler založený na architektuře RISC v pouzdře TQFP s 32 vývody. Výkon procesoru je 20 MIPS při kmitočtu 20 MHz, který je zároveň pro tento mikrokontroler maximální. Procesor disponuje 32 kB paměti flash, 1 kB paměti EEPROM a 2 kB paměti SRAM. Dále pak mikrokontroler obsahuje některé periferní obvody, jakými jsou například 10 bitový A/D převodník, programovatelný watchdog timer, vnější i vnitřní zdroj přerušení a další. [4] Více informací o mikrokontroleru Atmega328P lze najít v jeho datasheetu.

Z rodiny vývojových desek Arduino jsem zvolil typ NANO. Ten mi vyhovoval kvůli menším rozměrům, než má deska Arduino UNO, při zachování stejného počtu digitálních a analogových vstupů/výstupů. Dále pak deska oproti například typu MICRO umožňuje programování bez programátoru, pouze po propojení s PC pomocí USB / micro USB.

Dalším důvodem pro výběr právě desek Arduino, a ne práce přímo s mikrokontrolerem, ať už PIC nebo Atmel, byla snazší a uživatelsky příjemnější práce. Také jsem si práci s těmito zařízeními chtěl osvojit, protože jinde jsem neměl tu možnost.

Na obrázku níže (obr. 4-2) můžete vidět zapojení vývodů desky Arduino NANO. Jak je vidět, většina vývodů má více funkcí. To je dáno větším množstvím funkcí na vývodech mikrokontroleru, které jsou propojeny s výstupními vývody. Konkrétní funkce závisí na konfiguraci μP .



obr. 4-2: Arduino NANO – pinout [4]

4.4 Ovládací a zobrazovací obvody

Nároky na tyto části zapojení nejsou nijak velké. Proto jsem se při rozhodování jaké použít ovládací a zobrazovací prvky spokojil s těmi jednoduššími. Pro zobrazování jsem použil LCD zobrazovač. Jako ovládací prvky jsem zvolil rotační enkodery a tandemové potenciometry.

4.4.1 Ovládací prvky

Teď blíže k ovládacím prvkům. Popisovat, jak funguje potenciometr, je asi poněkud zbytečné, proto zde jen krátce. Pomocí potenciometrů v zapojení (viz příloha A) ovládám tři veličiny: amplitudu a posun analogového výstupního signálu a střidu signálu na digitálním výstupu pro TTL/CMOS obvody. Na jeden krajní vývod přední sekce tandemového potenciometru je přivedeno +5V a na druhý GND. Napětí na jezdcí přední sekce potenciometru je úměrné úhlu pootočení potenciometru a je vyhodnocováno řídicí jednotkou. Tento údaj slouží k zobrazení dané veličiny. Zadní sekce potenciometru přímo ovlivňuje zpracovávaný signál.

Druhým ovládacím prvkem jsou v mém zapojení rotační enkodery. Ten lze vidět na obrázku obr. 4-3. Jedná se o rotační ovládací prvek podobný potenciometru. Oproti potenciometru však enkodér nepodává informaci o konkrétní hodnotě veličiny. Princip činnosti je naznačen na obrázku obr. 4-3. Při rotaci jedním směrem přejede jezdec kontakty tak, že spojí nejdříve společný vývod COM s vývodem A a až poté s vývodem B. Při rotaci opačným směrem se kontakty propojí v pořadí B a A. Takto je tedy možné vyhodnotit jakým směrem se enkodér otočil. V mém případě při rotaci jedním směrem hodnoty inkrementují a při rotaci opačným směrem dekrementují.

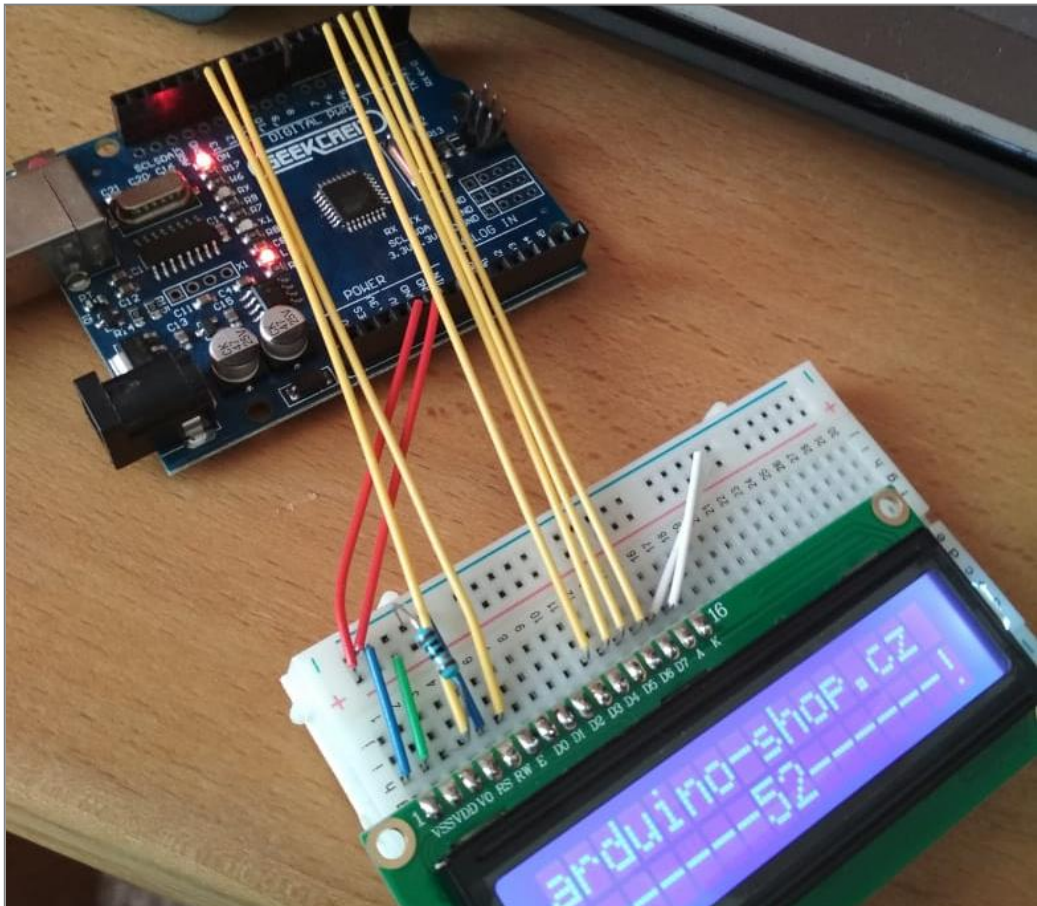


obr. 4-3: Rotační enkodér

4.4.2 Displej

Jak už bylo řečeno výše, pro zobrazování jednotlivých hodnot výstupního signálu jsem zvolil LCD. Použitý alfanumerický displej má rozlišení 16 znaků ve 2 řádcích. Pro lepší viditelnost při horších světelných podmínkách může být použito LED podsvícení. Komunikace s řídicí jednotkou probíhá po čtyřech datových (D4-D7) a dvou řídicích (RS a E) vodičích. Ostatní vodiče, které lze vidět na testovacím zapojení (obr. 4-4), jsou určené pro napájení displeje a podsvícení. Rezistor slouží k nastavení kontrastu znaků.

Ve finální aplikaci se na displeji bude zobrazovat aktuální kmitočet, střída, amplituda a stejnosměrný posun signálu. Při změně kmitočtu je nastavované řádkové místo zvýrazněno blikáním. Informace o zvoleném průběhu signálu (sinusový / obdélníkový / trojúhelníkový) se již na displej nevešla. Proto jsem využil volné vývody arduina, ke kterým jsem připojil indikační LED diody.



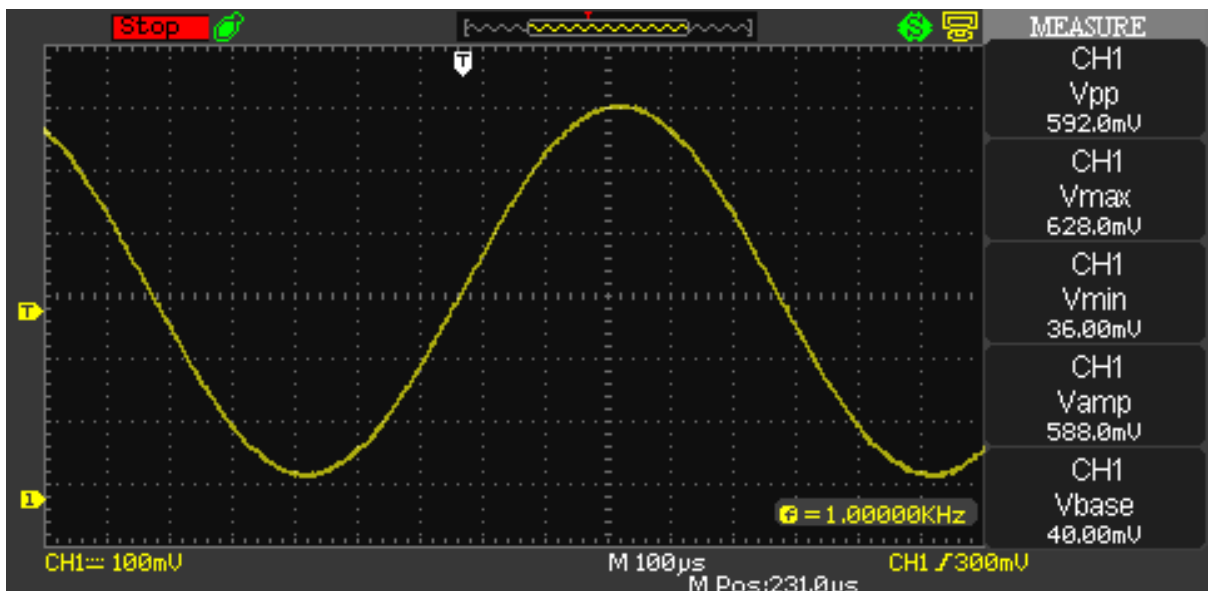
obr. 4-4: Testovací zapojení LCD

4.5 Výstupní obvody

Jelikož výstup z AD9833 nesplňuje požadavky, které jsem si pro výstupní signál stanovil, bylo nutné signál upravit a zesílit tak, aby těmto požadavkům vyhovoval. Na výstupu IO byl signál s konstantní amplitudou a ss posunem, který z něj udělal stejnosměrný signál proměnný.

4.5.1 Úprava signálu pomocí OZ

Nejprve bylo potřeba ze signálu odstranit ss složku a udělat z něj střídavý symetrický signál. Na obrázku obr. 4-5 je vidět, že časový průběh nabývá pouze kladných hodnot s minimem na desítkách mV a maximem o několik desítek mV překračujícím 0,6 V. U obdélníkového signálu má výstupní signál jiné napěťové úrovně. Logické nule odpovídá hodnota napětí přibližně 0 V. Logické jedničky pak hodnota necelých +5 V.



obr. 4-5: Oscilogram AD9833 (sinus; 1kHz)

Pro sinusový a trojúhelníkový signál jsem to provedl jednoduše ss předpětím přibližně 300mV (aritmetický průměr hodnot V_{\max} a V_{\min}), přivedeným na vstup operačního zesilovače pracujícího v invertujícím zapojení. Předpětí je nastaveno děličem z rezistorů a trimru (R_{14} R_{17} R_{18}). Pro obdélníkový signál bylo potřeba snížit jeho amplitudu tak, aby její střední hodnota byla stejná jako u předešlých průběhů. Toho bylo docíleno rozepnutím kontaktu relé K_1 . Tím byl pro všechny průběhy získán střídavý signál symetrický kolem nulové osy. Další funkcí prvního OZ je předzesílit signál pro další zpracování.

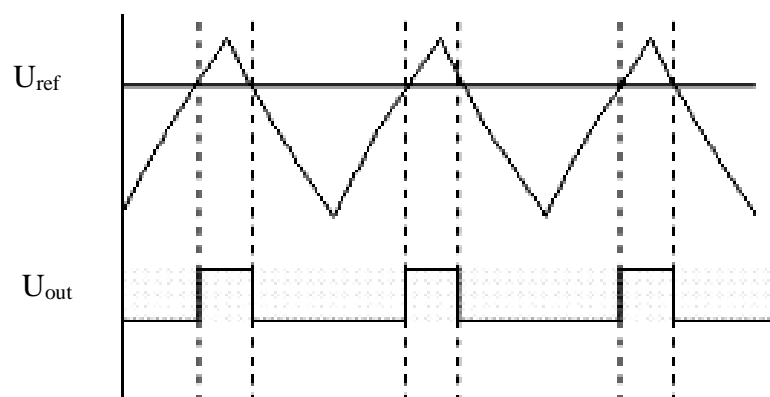
Druhá část úpravy signálu se týkala nastavení amplitudy a ss posunu signálu. Jako první se nastavuje amplituda. Tu nastavuji potenciometrem zapojeným mezi výstup prvního OZ a analogovou zem. Z jezdcy potenciometru se pak snímá signál a pokračuje dál na invertující vstup druhého OZ. Posun se nastavuje obdobně jako u prvního stupně. Mezi +12V a -12V je

zapojen potenciometr tak, aby na něm vznikl úbytek napětí odpovídající maximálnímu posunu ± 3 V. Výsledný posun signálu závisí na poloze jezdce potenciometru. Napětí z něj se přivádí na invertující vstup druhého OZ v součtu s napětím z jezdce potenciometru pro nastavení amplitudy.

Pro obě dvě části bylo nutné vybrat OZ, který bude mít dostatečnou kvalitu. Jelikož signál generovaný AD9833 může dosahovat kmitočtu až 12,5 MHz, bylo nutné vybrat OZ pro tento kmitočtový rozsah. Dalším důležitým parametrem byl nízký šum OZ, což se dá u zařízení jako je funkční generátor očekávat. Těmto parametrům vyhovovalo více operačních zesilovačů. Těmi byly například OPA2134, TLE2071C a OPA317. Nakonec jsem se po dohodě s konzultantem práce rozhodl pro operační zesilovač TLE2071C.

4.5.2 Výstup pro TTL/CMOS

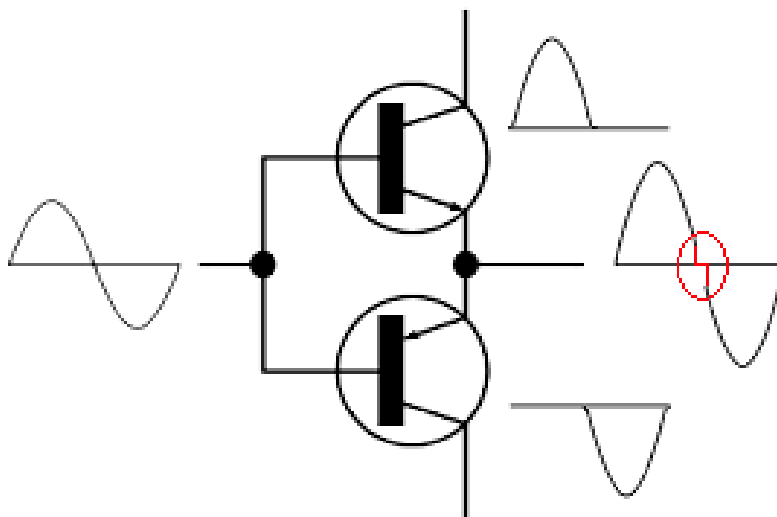
Po návrhu základních analogových výstupních obvodů jsem se rozhodl vytvořit na generátoru ještě druhý výstup pro TTL a CMOS obvody. Ten jsem opět realizoval pomocí operačního zesilovače TLE2071C. Nyní však v zapojení invertujícího komparátoru s hysterezí. Ta byla vytvořena zavedením kladné zpětné vazby a její velikost jsem zvolil cca 6mV (tzn. ± 3 mV). Tento výstup má ještě druhou funkci, a to regulaci střidy výstupního digitálního signálu. Ta vychází z porovnávání trojúhelníkového signálu s hodnotou referenčního napětí. Tento princip je naznačen na obrázku obr. 4-6. Kdykoli, když hodnota napětí trojúhelníkového signálu přesáhne referenční napětí, komparátor přepoklopí do logické jedničky a pokud poklesne pod hodnotu referenčního napětí, přepoklopí do logické nuly. Takže změnou referenčního napětí můžeme regulovat střidu signálu v rozmezí 0 až 100%. Podmínkou je však nastavení trojúhelníkového či pilového průběhu na jednom ze vstupů OZ. Výstup OZ je poté proudově zesílen tranzistorem BCX5516. Antiparalelně v přechodu báze-emitor tranzistoru je zapojena dioda, která ho chrání před průrazem v závěrném směru, protože na výstupu komparátoru je napětí buď +5 V, anebo -5 V.



obr. 4-6: Regulace střidy

4.5.3 Koncový stupeň

Stejně tak jako bylo potřeba proudově zesílit výstup pro TTL/CMOS obvody, bylo nutné zesílit i výstup analogových signálů. Zde jsem využil zapojení s komplementární dvojicí tranzistorů BCX5216 a BCX5516. Tím však vzniká přechodové zkreslení. Tento nežádoucí jev je vyznačen na obrázku obr. 4-7 červenou čarou a je nutné ho eliminovat. Abychom se ho zbavili, je nutné tranzistory i při klidovém stavu pootevřít tak, že vytvoříme předpětí pro báze obou tranzistorů. Předpětí je získáno součtem napětí na přechodech báze-emitor tranzistorů T_4 , T_5 a T_6 (jde o zapojení tranzistorů jako diod se strmou charakteristikou), kterými prochází konstantní proud z tzv. proudového zrcadla. Toto zapojení je vytvořeno tranzistory T_7 a T_8 . Jelikož mají oba tranzistory téměř stejné vlastnosti přechodu báze-emitor, teče oběma tranzistory přibližně stejný kolektorový proud. Řídící tranzistor T_7 je proudově napájen z +12 V přes rezistor. Tranzistorem T_8 protéká shodný proud téměř bez ohledu na velikosti jeho napětí mezi kolektorem a emitorem. Chová se tak jak proudový zdroj pro tranzistory T_4 , T_5 a T_6 .



obr. 4-7: Komplementární dvojice

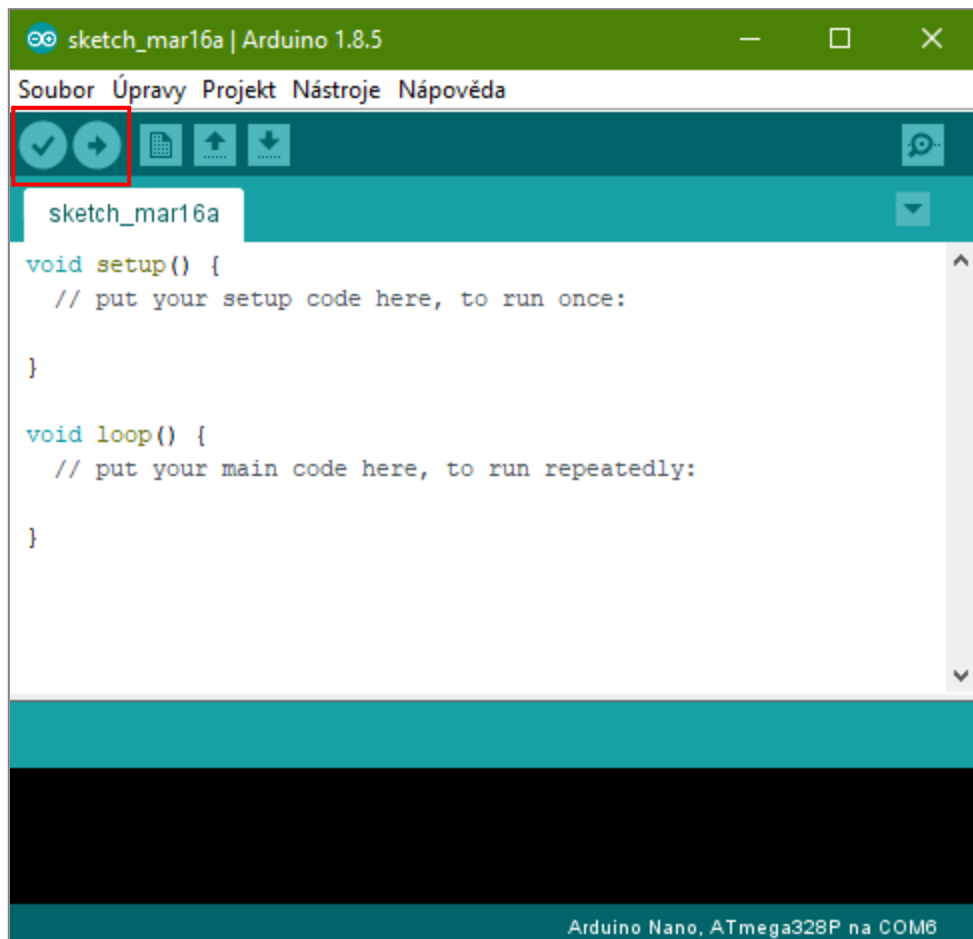
Zpětná vazba pro OZ_2 není vedena přímo z jeho výstupu, ale až z výstupu koncového stupně. Zamezí se tak zkreslení, které by vzniklo tranzistory komplementární dvojice, a tak bude výstupní signál přesně v poměru se vstupním napětím OZ_2 .

5 SOFTWARE GENERÁTORU

Stejně tak jako byl v předešlé kapitole popsán hardware funkčního generátoru, bude v této kapitole popsán software. Kapitulu jsem rozdělil do tří podkapitol. V první se lze seznámit s vývojovým prostředím, druhá se zabývá částí programu obsluhující uživatelské vstupy a třetí komunikací mezi řídicí jednotkou a ostatními bloky zapojení. Výpis zdrojového textu programu a kompletní vývojový diagram jsou uvedeny v přílohách 5 a 6 na CD.

5.1 Arduino IDE

Název kapitoly je zároveň názvem použitého vývojového prostředí. Programy se v tomto vývojovém prostředí vytváří v programovacím jazyce podobnému jazyku C a C++. Po spuštění se otevře okno základní obrazovky, viz obrázek obr. 5-1. Jsou zde připraveny dvě prázdné programové smyčky. První se nazývá *setup()* a po zapnutí napájení proběhne pouze jednou. Slouží k úvodnímu nastavení desky (chcete-li její inicializaci). Druhá se nazývá *loop()* a vykonává se opakovaně až do vypnutí napájení. Proto se v ní zpravidla vykonává hlavní programová smyčka. Mimo tyto hlavní smyčky se píše kód podprogramů nebo se zde inicializují globální proměnné. V dolní části obrázku můžete vidět konzoli, kde se vypisují chybová hlášení a podobně. Poslední důležitou informací pro začátek je účel dvou kulatých tlačítek v levém horním rohu obrázku označených červeným rámečkem. První tlačítko *Verify* slouží k ověření syntaxe popř. přeložení programu do strojového kódu. Druhým tlačítkem *Send* se kód nahraje do desky připojené k PC pomocí zvoleného USB portu.

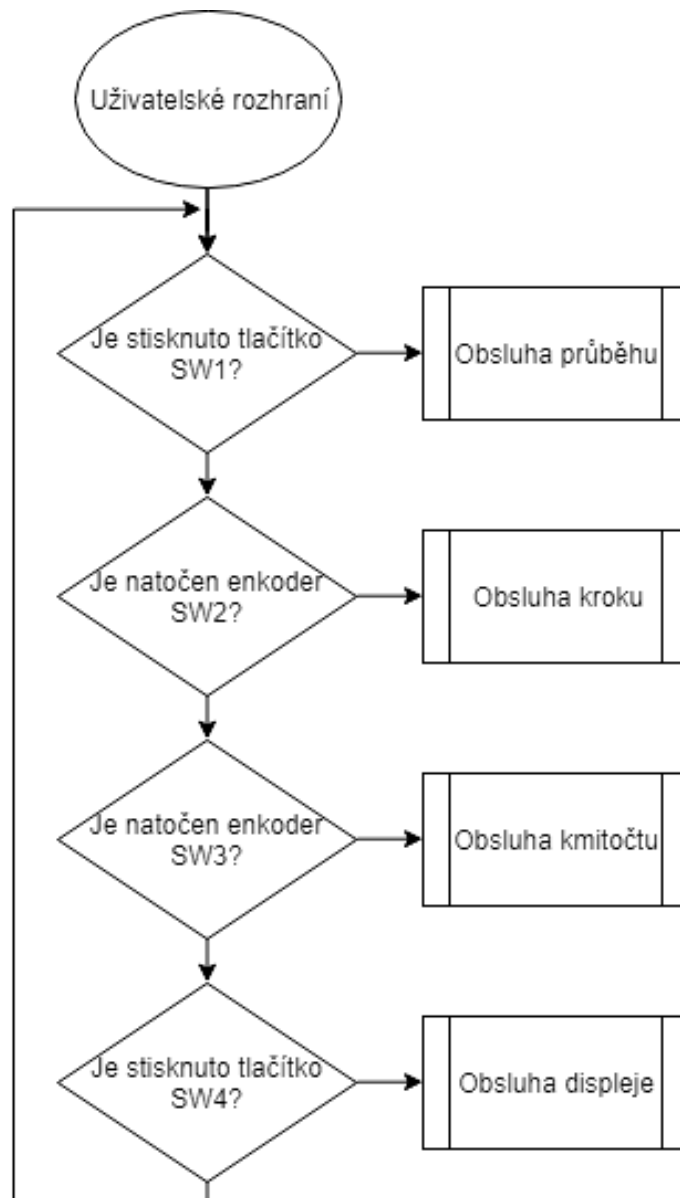


obr. 5-1: Vývojové prostředí Arduino IDE

Hlavní výhodou tohoto vývojového prostředí je podle mě jeho komunitní vývoj. Jednotliví uživatelé poskytují své knihovny k dispozici ostatním uživatelům. Díky tomu v dnešní době existuje mnoho knihoven, které práci programátora dosti zjednoduší. Já jsem například využil knihovny *SPI.h* a *LiquidCrystal.h*, abych si práci usnadnil.

5.2 Obsluha uživatelského rozhraní

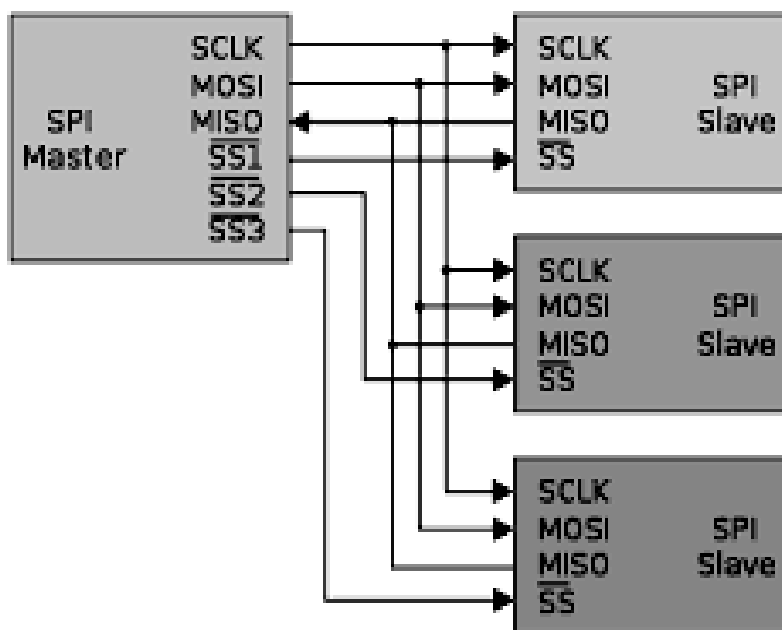
Obsluha uživatelských vstupů je velmi jednoduchá, protože chceme, aby pracovala co možná nejrychleji. Její zjednodušený vývojový diagram je možné vidět na obrázku obr. 5-2. Je založena na nekonečném cyklu, ve kterém stále dokola zjišťujeme, zda-li se nezměnil stav některého z uživatelských vstupů (stisk tlačítka nebo otočení ekoderu). Když se nějaká tato testovací podmínka splní, dojde k obslužení příslušného vstupu. Nekonečný cyklus, ve kterém se tyto podmínky vyhodnocují, je hlavní programová smyčka *loop()*, popsána v předešlé kapitole.



obr. 5-2: Zjednodušený VD - uživatelské rozhraní

5.3 Komunikace mezi řídicí jednotkou a ostatními bloky

Při inicializaci řídicí jednotky po zapnutí napájení se současně provede i reset ostatních bloků. Dále se čeká na pokyny uživatele. Jelikož mi přišlo zbytečné podmiňovat zahájení komunikace s IO AD9833 speciálním uživatelským vstupem, použil jsem jako podmínku pro její zahájení uživatelský vstup pro změnu průběhu analogového signálu. Komunikace probíhá po synchronní sériové lince SPI (obecně znázorněné na obrázku obr. 5-3). Master je v mém případě arduino a Slave integrovaný obvod AD9833, jehož komunikaci s řídicí jednotkou řeší dva podprogramy.



obr. 5-3: SPI komunikační protokol

První podprogram si připraví data pro vysílání do IO AD9833 (hodnotu kmitočtu, fáze a tvaru signálu) tak, aby byla zachována posloupnost bitů pro sériový přenos. Toto pořadí je uvedeno v datasheetu integrovaného obvodu. Poté se začne volat druhý podprogram, který už takto připravené bity postupně posílá do integrovaného obvodu. Po ukončení přenosu se program vrací zpět do hlavní programové smyčky *loop()*, kde čeká na další pokyn.

Druhým zařízením, se kterým řídicí jednotka komunikuje, je displej. Pro zjednodušení programu i usnadnění práce jsem využil základně implementovanou knihovnu *LiquidCrystal*. Takže namísto tvoření již existujícího programu pro paralelní přenos dat stačilo řídicí jednotce definovat, kam jsou připojené vývody displeje a použít sadu příkazů z knihovny.

6 STAVBA A ZPROVOZNĚNÍ GENERÁTORU

Tato kapitola se zabývá výrobou funkčního generátoru. Základem pro jeho tvorbu bylo vytvoření schémat a jiné výkresové dokumentace. Pro další postup bylo nutné spočítat hodnoty jednotlivých pasivních součástek a nakoupit je. Když byly všechny podklady a materiály pohromadě, začal jsem se samotnou stavbou.

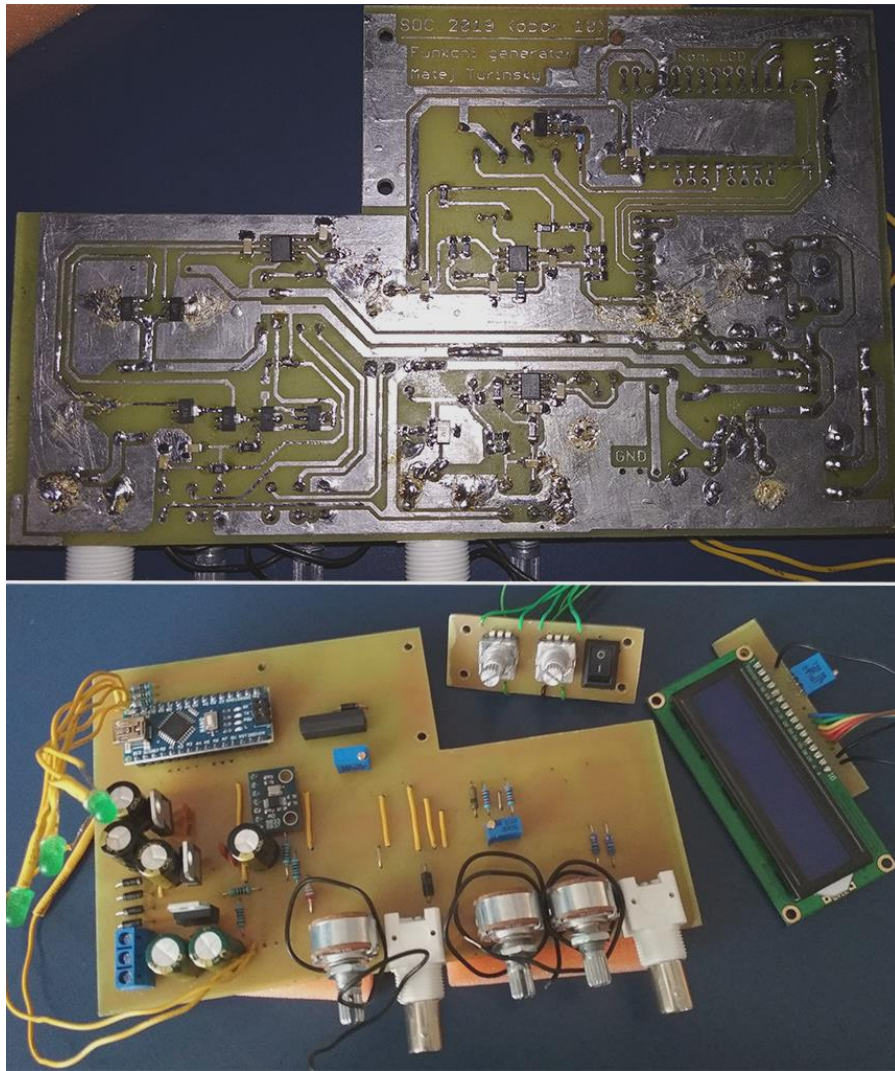
Nejprve jsem na nepájivém poli testoval zapojení, převážně pak software. Když byl software nazákladě uživatelských vstupů schopen ovládat IO AD9833, bylo potřeba změřit, jak výstupní signál integrovaného obvodu vypadá. Následně jsem začal s realizací výstupních obvodů tak, jak je to popsáno v kapitolách 4.5.1, 4.5.2 a 4.5.3. Při dosažení stanovených požadavků na výstupní signály z funkčního generátoru jsem začal pracovat na konečném provedení stavby generátoru.

6.1 Stavba funkčního generátoru

Prvním bodem stavby generátoru bylo vyrobit DPS. Pro jeho návrh jsem využil návrhové prostředí Eagle. Pro výrobu jsem zvolil fotocestu a jednovrstvý plošný spoj. Celé zapojení je rozděleno do tří celků, které jsem ale umístil na jednu desku a poté cuprexit rozdělit na tři části. Při návrhu jsem pracoval jak se zemí digitální (GND), tak i se zemí analogovou (AGND). Proto jsem se musel v zapojení vyvarovat nechtěných proudových smyček a země propojit jen v jednom bodě.

Takto vyrobený plošný spoj jsem poté osadil. Při osazení jsem pracoval s vývodovými i SMD součástkami. Vývodové součástky jsou osazeny obvyklým způsobem, a to za pomoci páječky a cínu. Pro osazení součástek SMD jsem zvolil metodu pájení horkým vzduchem s použitím pájecí pasty. Ta byla ale zdlouhavá, proto jsem pro součástky se dvěma vývody použil opět páječku a cín. Takto osazený plošný spoj můžete vidět na obrázku obr. 6-1.

Posledním krokem bylo takto osazenou desku vložit společně s toroidním transformátorem do plastové krabičky a vše mechanicky zajistit pomocí šroubků a vrutů. Následně připojit pomocí vodičů oddělenou část desky, na které je displej a část ovládacích prvků, a také je mechanicky zajistit.



obr. 6-1: Osazený plošný spoj

6.2 Test funkčnosti

Po dokončení stavby generátoru bylo nutné zjistit, zda jeho kvalita odpovídá zadání. Před uvedením do provozu jsem provedl finální kontrolu DPS. Dále jsem otestoval chod programu mimo zapojení. Když jsem si byl jistý, že DPS i program jsou v pořádku, začal jsem se samotným testem funkčnosti zapojení. Ten se sestával s naměřením hodnot výstupního signálu a porovnání se zvoleným normálem GFG-8015G. Naměřené časové průběhy obou funkčních generátorů jsou součástí přílohy 7 (Oscilogramy AD9833 vs. GFG-8015G) a přílohy 8 (Oscilogramy sestaveného funkčního generátoru).

Při prvním spuštění zařízení byly zjištěny dvě závady. První byla chyba v komunikaci mezi řídicí jednotkou a integrovaným obvodem AD9833. Ta byla způsobena špatným zapojením vývodů arduina, určených pro SPI komunikaci. Tato chyba byla následně hardwarově odstraněna. Druhá chybou bylo špatné zapojení tranzistoru PNP (T_9). Chyba byla způsobena špatně zvoleným typem tranzistoru v návrhu plošného spoje. Bylo tedy nutné opět provést další hardwareové úpravy, které tuto chybu eliminovali.

7 ZÁVĚR

Výstupem této práce je funkční generátor, který umožňuje na prvním ze dvou výstupů generovat analogový signál s požadovaným tvarem, kmitočtem, amplitudou a stejnosměrným posunem. Zároveň na druhém výstupu generuje digitální signál uzpůsobený obvody TTL a CMOS s nastavitelnou střídou signálu. Dosáhnul jsem tak toho, že jsem si vytvořil další přístroj pro svoji dílničku.

Cíl této práce byl splněn, protože výstup práce odpovídá mnou zadaným požadavkům na parametry funkčního generátoru, jak vyplývá z provedených měření. Při návrhu jsem se jistě dopustil některých chyb a zapojení má tak nejspíš své nedostatky. Zatím jsem však spokojený s výsledkem své práce, ale plánuji ho doplnit o další funkce. Jednou z nich by mělo být rozmítání signálu.

Vedlejším přínosem bylo osvojení si práce s vývojovým prostředím pro programování vývojových desek Arduino a získání zkušeností s prací s operačními zesilovači a dalšími součástkami. Také jsem si poprvé vyzkoušel osazovat součástky SMD a myslím, že úspěšně.

8 POUŽITÁ LITERATURA

- [1] P. Jaroslav, Softwarový funkční generátor, Plzeň: Bakalářská práce: Západočeská univerzita v Plzni. Fakulta elektrotechnická, 2012.
- [2] R. H. serveru, „vyvoj.hw.cz,“ 3. 3. 2001. [Online]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/prima-cislicova-synteza-kmitoctu-dds-direct-digital-synthesis.html>. [Přístup získán 10. 2. 2019].
- [3] ANALOG_DEVICE, „Data Sheet: AD9833,“ [Online]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad9833.pdf>. [Přístup získán 17. 3. 2019].
- [4] „Arduino Forum,“ Arduino, 2019. [Online]. Dostupné z: <https://www.arduino.cc>. [Přístup získán 17. 3. 2019].

9 POUŽITÉ PROGRAMY

- EAGLE
- Arduino IDE
- Proteus
- Draw.io
- MS Word 2013

10 SEZNAM OBRÁZKŮ

| | |
|---|----|
| obr. 2-1: Periodické signály | 8 |
| obr. 2-2: Spojité a diskrétní signály | 9 |
| obr. 2-3: Invertující integrátor | 10 |
| obr. 2-4: Neinvertující komparátor s hysterézí | 10 |
| obr. 2-5: Blokové schéma analogového oscilátoru [1] | 10 |
| obr. 2-6: Princip DAS [1] | 11 |
| obr. 2-7: Blokové schéma PPL [1]..... | 11 |
| obr. 2-8: Blokové schéma DDS [2] | 12 |
| obr. 2-9: Průběhy signálu v DDS..... | 13 |
| obr. 2-10: DAC s váhovými odpory | 13 |
| obr. 4-1: AD9833 shield | 15 |
| obr. 4-2: Arduino NANO – pinout [4]..... | 16 |
| obr. 4-3: Rotační enkoder | 17 |

| | |
|---|----|
| obr. 4-4: Testovací zapojení LCD..... | 18 |
| obr. 4-5: Oscilogram AD9833 (sínus; 1kHz)..... | 19 |
| obr. 4-6: Regulace střídý..... | 20 |
| obr. 4-7: Komplementární dvojice..... | 21 |
| obr. 5-1: Vývojové prostředí Arduino IDE..... | 22 |
| obr. 5-2: Zjednodušený VD - uživatelské rozhraní..... | 23 |
| obr. 5-3: SPI komunikační protokol | 24 |
| obr. 6-1: Osazený plošný spoj..... | 26 |

11 SEZNAM PŘÍLOH

| | |
|-----------|---|
| Příloha A | Blokové schéma zapojení |
| Příloha B | Schéma zapojení napájecích obvodů |
| Příloha C | Schéma zapojení řídicích obvodů |
| Příloha D | Schéma zapojení analogového zesilovače |
| Příloha E | Schéma zapojení koncového stupně a výstupu TTL/CMOS |
| Příloha F | Motiv plošného spoje |
| Příloha G | Osazovací výkres horní strany |
| Příloha H | Osazovací výkres dolní strany |

12 SEZNAM PŘÍLOH NA CD

| | |
|-----------|--|
| Příloha 1 | Schémata zapojení |
| Příloha 2 | Plošný spoj |
| Příloha 3 | Výpočty hodnot součástek |
| Příloha 4 | Seznam součástek |
| Příloha 5 | Vývojový diagram |
| Příloha 6 | Výpis programu |
| Příloha 7 | Oscilogramy AD9833 vs. GFG-8015G |
| Příloha 8 | Oscilogramy sestaveného funkčního generátoru |