



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Mikroprocesorové a vestavěné systémy

Simulátor hudebního nástroje

2020/2021

Matej Otčenáš, xotcen01

December 5, 2020

Obsah

1	Úvod	2
2	Návod na spustenie programu	2
3	Implementácia	2
3.1	Podstata vytvárania tónin	2
4	Zvukový výstup	3
4.1	Pripojenie reproduktora	3
5	Návod na ovládanie programu	6
6	Odkaz na demonštračné video	7
	Zdroje	7

1 Úvod

Úlohou projektu bolo vytvoriť jednoduchý simulátor hudobného nástroja, ktorý bude schopný prehrávať cez pripojený *buzzer* reproduktor predpripravené demonštračné melódie na určitých frekvenciách. Povinný rozsah jednotlivých tónin je minimálne jedna oktáva.

2 Návod na spustenie programu

Na spustenie hudobného simulátoru je potrebné zariadenie *Arduino UNO* [1]. K jeho naprogramovaniu je rovnako použité vývojové prostredie od Arduina, ktoré umožňuje jednoduchú komunikáciu s hardwarovým kitom a súčasný preklad programu. Program bol vyvíjaný na platforme *Linux*, avšak s uvedeným vývojovým prostredím je možné simulátor spustiť na akejkoľvek inej platforme, ktorá toto prostredie poskytuje. Po naprogramovaní zariadenia je potrebné spustiť terminálové rozhranie v sekcii *Tools->Serial Monitor*, pomocou ktorého je možné s Arduino komunikovať v podobe **textových reťazcov**.

3 Implementácia

Program bol implementovaný pomocou programovacieho jazyka *Arduino Programming Language* [2]. Zdrojový súbor s názvom *main.ino* obsahuje hlavný program. Súbor s názvom *notes.h* je súbor dostupný z verejných príkladov Arduina a obsahuje definície názvov tónov s pevne stanoveným frekvenciami [3]. Zdrojový kód v súbore *main.ino* je ďalej rozdelený do častí **globálnych premenných**, ktoré sa skladajú predovšetkým z polí, ktoré obsahujú jednotlivé tóny melódií a ich dĺžky. Následne sú vytvorené **pomocné funkcie** na prehrávanie jednotlivých melódií. Na záver program obsahuje dve hlavné funkcie špecifické pre programovací jazyk Arduino. Prvou z nich je funkcia **setup()**, ktorá sa spúšťa po naprogramovaní kitu ako inicializačná funkcia. Druhou a zároveň poslednou funkciou je **loop()**, ktorá funguje na báze aktívneho čakania a spracováva terminálové vstupy.

3.1 Podstata vytvárania tónin

Jednou z najdôležitejších častí programu bola výsledná interpretácia výstupného tónu na pripojený reproduktor, kde bolo potrebné použiť správny prevod trvania jednej noty na časovú veličinu v milisekundách a následne sa interpretovala každá tónina jednotlivo v cykle podľa bližších špecifik. Aby bolo možné predávať na zvukový výstup správne dĺžky tónin, tak je nutné v cykle ošetriť samotnú dĺžku aktuálnej tóniny. V prípade, že takáto dĺžka má negatívnu hodnotu, tak vieme že ide o tzv. *polovú notu*¹, ktorej dĺžka je o polovicu dlhšia ako bežná nota.

¹https://en.wikipedia.org/wiki/Dotted_note

Po správnom prevedení postačuje použiť funkcie, ktorými disponuje samotný programovací jazyk pre Arduino a vie si ich korektne interpretovať. Pre lepšie pochopenie vytvárania tónin je použitý aj následovný pseudokód[4][3].

```
// dlzka celej noty v ms
// tempo udava rychlost prehratia noty
celaNota = (60000 * 4) / tempo;
for (i = 0; i < pocetNot; i++) {
    // normalna nota
    if (dlzkaNotyMelodie[i] > 0) {
        dlzkaNoty = (celaNota) / dlzkaNotyMelodie[i];
    }
    // polova nota
    else if (dlzkaNotyMelodie[i] < 0) {
        dlzkaNoty = (celaNota) / abs(dlzkaNotyMelodie[i]);
        dlzkaNoty *= 1.5;
    }

    //dlzkaNoty*0.9 -> zahrame 90% z noty a 10% je pauza
    tone(vystupnyPin, melodia[i], dlzkaNoty * 0.9);
    // nutne spozdenie o case dlzky noty
    delay(dlzkaNoty);
    // zastavenie generovania frekvenci na danom pine
    noTone(vystupnyPin);
}
```

Pre prevod hudobných nôt do frekvencií na konkrétne melódie boli použité dostupné zdroje a spolupráca s blízkym hudobníkom[5][6].

4 Zvukový výstup

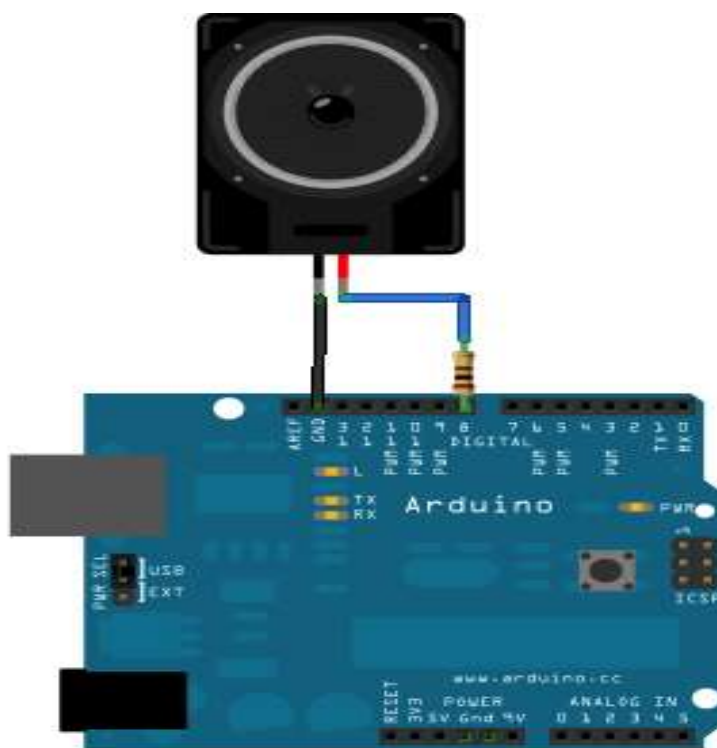
Softwarová implementácia pre žiadaný zvukový výstup bola vysvetlená v časti 3.1, avšak aby bolo možné interpretovať zvuk aj na hardwarovom výstupe, tak je nutné aj správne zapojenie, ktoré popisuje sekcia 4.1.

4.1 Pripojenie reproduktora

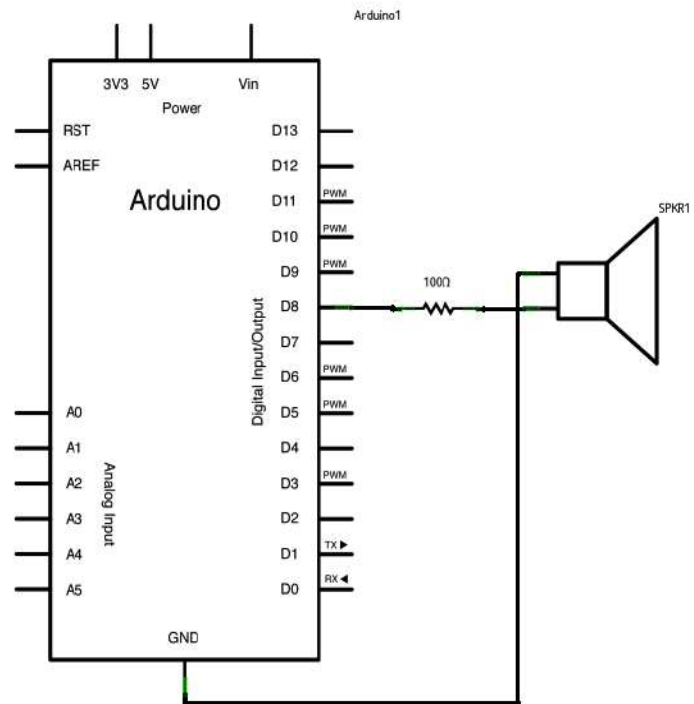
Z pohľadu logiky zapojenia je samotný Arduino kit pomerne jednoduchý na pochopenie a jediné čo stačí je prepojiť slaboprúdové dátové káble na kladný a záporný pól 8 Ohm-ového reproduktora a súčasne ich napojiť na vybraný pin² a uzemniť. Aby bola konštrukcia plne korektná, mal by byť použitý napríklad 100 Ohm-ový rezistor, ktorý je veľmi dobrou prevenciou pred poškodením samotného kitu či reproduktora, avšak v uvedenej implementácii tento rezistor

²V prípade implementácie sa jedná o pin č.8

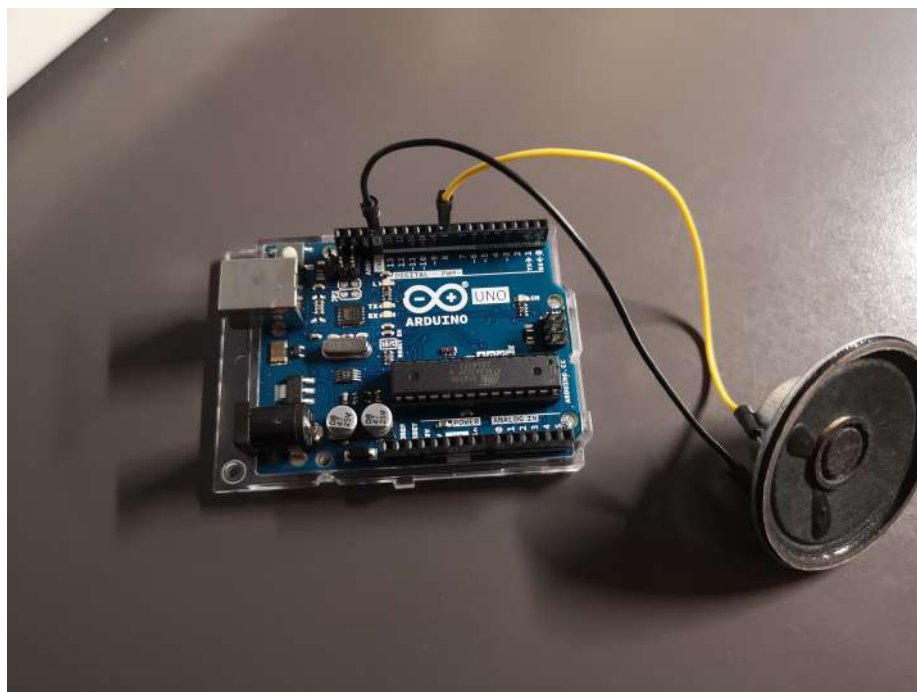
použitý nie je, čo ale zvukový výstup neovplyvňuje. Obrázok 1 názorne zobrazuje pripojenie reproduktora na Arduino v plnej korektnosti. Obrázok 2 je identické zobrazenie prevedené v schematickej podobe. Na záver obrázok 3 je výsledné zapojenie reproduktora bez použitia ochranného rezistoru.



Obrázok 1: Nákres zapojenia



Obrázok 2: Schematický náčrt zapojenia



Obrázok 3: Praktické zapojenie reproduktora

5 Návod na ovládanie programu

V sekcii 2 je uvedený presný postup na spustenie terminálu vo vývojovom prostredí Arduina a rovnako tak forma vstupných príkazov. Keďže hardwarové zariadenie nedisponuje žiadnymi vstupnými perifériami vo forme tlačítka (odhliadnuc od tlačítka *RESET*), tak je možné program ovládať **iba** pomocou vstupných terminálových reťazcov. Aby bolo možné počuť hudobný výstup či inak komunikovať so zariadením, tak program očakáva jeden z nasledovných reťazcov:

- **merry christmas** - spustí predpripravenú melódiu o dĺžke približne 20 sekúnd
- **silent night** - spustí obdobnú melódiu podobného rozsahu
- **star wars** - melódia z filmu *Star Wars*
- **help** - vypíše dostupnú nápovedu k programu

6 Odkaz na demonštračné video

<https://vimeo.com/487927184>

Zdroje

- [1] *Arduino Uno Rev3*. URL: <https://store.arduino.cc/arduino-uno-rev3>.
- [2] *Language Reference*. URL: <https://www.arduino.cc/reference/en/>.
- [3] The Arduino Team. *Play a Melody using the tone() function*. URL: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody>.
- [4] *Melody*. URL: <https://www.arduino.cc/en/tutorial/melody>.
- [5] Joe Wolfe. *Music Acoustics*. URL: <http://newt.phys.unsw.edu.au/music/note/>.
- [6] B. H. Suits. *Tuning*. URL: <https://pages.mtu.edu/~suits/notefreqs.html>.