

Fakulta Informačních technologií

Vysoké Učení Technické v Brně

IMP - Mikroprocesorové a vstavané systémy

Varianta termínu - B :Simulátor hudebního nástroje

V Brně dňa: 19.12.2021

Marek Németh
xnemet05

Obsah

1. Úvod	3
2. Návod	3
2.1. Spustenie dema	3
2.2. Pridanie a zahranie vlastnej melódie	4
2.3. Prehratie celej oktávy	5
3. Implementácia	5
3.1. Základné funkcie	5
3.2. Komunikacia s MCU	6
4. Zhrnutie	6
5. Schéma zapojenia	7
6. Odkaz na video	7
7. Zdroje	7

1. Úvod

Zadaním projektu bolo navrhnuť a implementovať systém pre prehrávanie zvolenej melódie na reproduktore cez ESP32.

Systém mal podporovať nasledovné funkcionality:

- Zahranie zvolenej melódie užívateľom pomocou terminálu s vhodným vstupným reťazcom
- Prehratie tónov aspoň jednej oktávy
- Prehratie demo-skladby

2. Návod

Pre spustenie systému je nevyhnutný os Windows linux alebo macOS. Pre naflashovanie systému do MCU môžete použiť nástroj 'idf.py flash'[1] alebo naflashovať cez arduino IDE[2]. Jednotlivé funkcionality fungujú cez komunikáciu cez Serial(viac v kap.3.2):

2.1. Spustenie dema

príkaz 'demo' ktorý sa odošle cez Serial prehrá jednoduchú a krátku demo-skladbu

2.2. Pridanie a zahranie vlastnej melódie

Užívateľ môže cez serial posielat' jednotlivé tóny (v rozsahu oktávy C_5) s dĺžkou od 'štvrtinovej noty' (150 ms) po štvoritú dĺžku celej noty (2400 ms) vo formáte : '<nota> <dĺžka>'

Príklad pridania noty:

```
User>>'C 1'  
MCU>>'C 1'  
User>>'a 5'  
MCU>>'a 5'
```

System nerozlišuje veľké a malé písmená, dĺžka je predaná ako index (viac v kap. 3.)

Prehratie uloženej melódie je vykonané zaslaním príkazu 'end'. Pre vyčistenie pamäte pre uloženú melódiu slúži príkaz 'clear'.

2.3. Prehratie celej oktávy

Pre prehratie oktávy C_5 použite príkaz 'octave'.

3. Implementácia

Väčšinu implementácie tvoria jednoduché funkcie na rozoznávanie príkazov od užívateľa a statické polia s koštantami. Tieto polia obsahujú nasledovné:

demo - obsahuje jednotlivé frekvencie tónov

demo-skladby

demo_dels - obsahuje dĺžky jednotlivých tónov v demo-skladbe

octaveC4,C5 - nachádzajú sa v nich frekvencie jednotlivých tónov v oktávach C_4 a C_5 . [3]

delays - obsahuje v sebe dĺžky jednotlivých nôt

3.1. Základné funkcie

V kóde sú implementované nasledovné funkcie:

void playOctave - prehrá celú oktávu pre užívateľa

void play_demo - prehrá demo-skladu

Tieto funkcie využívajú priamo zápis frekvencie cez PWM na výstupný port GPIO16 pomocou `ledcWriteTone(ledChannel, <freq>)[4]`

dálej funkcie

`void setup` - inicializácia pinov a `Serial` komunikácie

`void loop` - ktorá zaručuje beh programu

`bool read_in` - funkcia ktorá prečíta a spracuje vstup od užívateľa vracia `true` ak používateľ zadal tón na pridanie alebo využil nejakú implementovanú funkcionálnosť, `false` ak zadal príkaz `end`, potom nasleduje prehratie melódie vo funkcii `loop`.

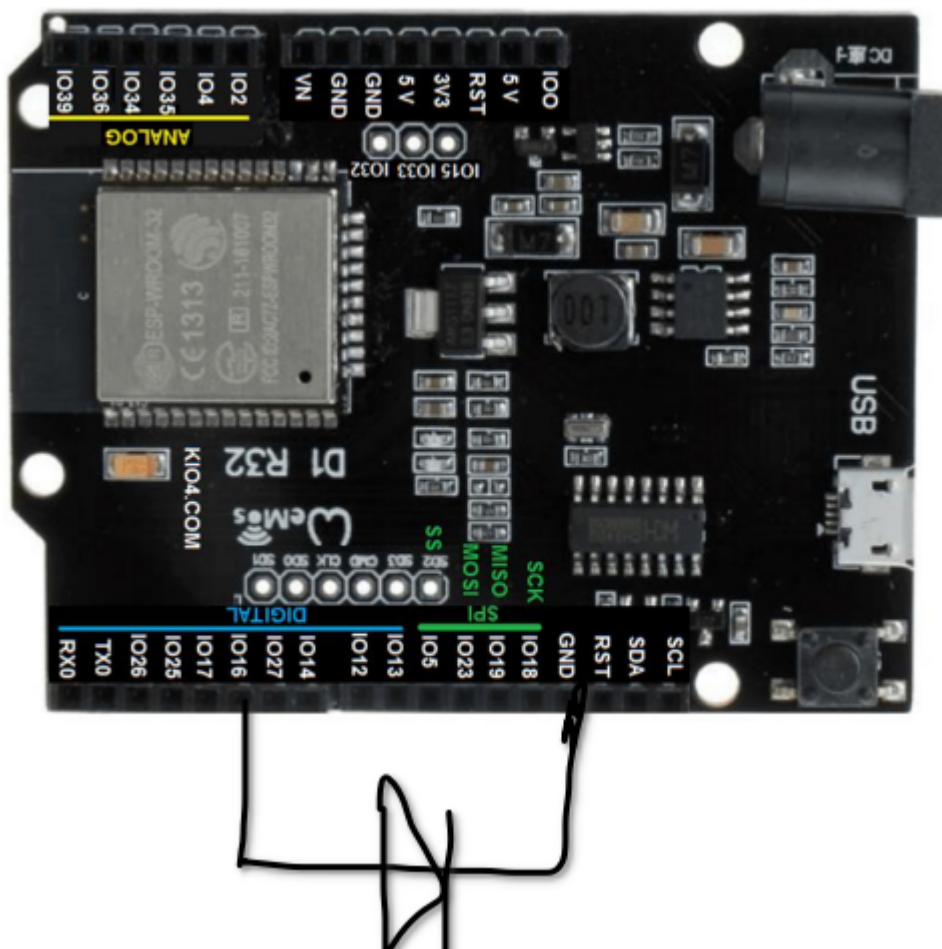
3.2. Komunikácia s MCU

Celá komunikácia s programom prebieha cez `Serial[5]` pomocou `Serial.readBytes` a `Serial.println`. Spracovávanie inputu os `Usera` je implementované pomocou regulárnych výrazov s knižnicou `regex`.

4. Zhrnutie

Svoje riešenie považujem za pomerne jednoduché no praktické z pohľadu prístupu k úlohe. Možné rozšírenie by bolo určite pridať možnosť prehrať akýkoľvek tón ktoré by bolo implementované pomocou ďalších polí pre jednotlivé oktávy.

5. Schéma zapojenia



6. Odkaz na video

<https://youtu.be/TZI52MdGZ0o>

5. Zdroje

- [1] <https://github.com/espressif/esp-idf>
- [2] <https://www.arduino.cc/en/software>
- [3] <https://pages.mtu.edu/~suits/notefreqs.html>
- [4] <https://techtutorialsx.com/2017/07/01/esp32-arduino-controlling-a-buzzer-with-pwm/>
- [5] <https://www.arduino.cc/reference/en/language/functions/communication/serial/>