

Systemy Mikroprocesorowe i Wbudowane - Projekt

Szafa grająca

Przedstawienie projektu

Układ ma za zadanie odgrywać wybraną przez użytkownika melodię.

Urządzenia wejściowe stanowią 3 przyciski.
Urządzenia wyjściowe to wyświetlacz LCD oraz buzzer.

Przedstawienie projektu cd.

Melodie zostały skonstruowane po stronie programowej.

W pamięci programu mieści się 16 melodii.

Założenie projektu

Ilość melodii początkowo miała być uzależniona od ilości przycisków, jednak bardziej optymalnym jakościowo rozwiązaniem okazało się użycie dwóch przycisków jako nawigacyjnych, a tylko jednego jako inicjalizującego odegranie melodii

Tone()

Opis funkcji Tone():

- ❖ Funkcja Tone() umożliwia wygenerowanie przebiegu prostokątnego o wypełnieniu 50%. Na zadanym pinie. Długość trwania może zostać zadana, w przeciwnym razie funkcja działa do momentu wywołania noTone(). Pin może zostać podłączony do buzzera lub innego głośnika.

Składnia funkcji:

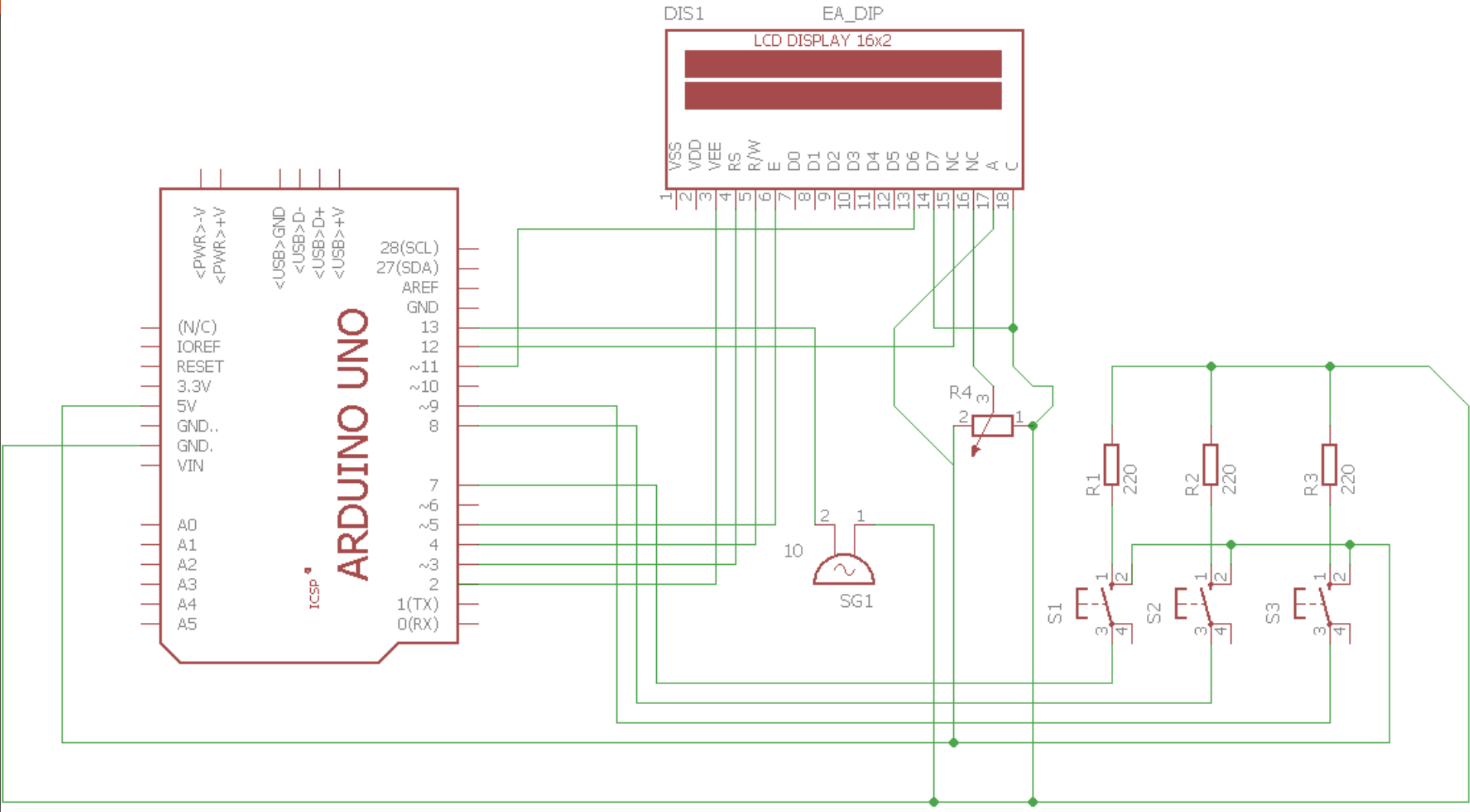
Tone (pin, częstotliwość)

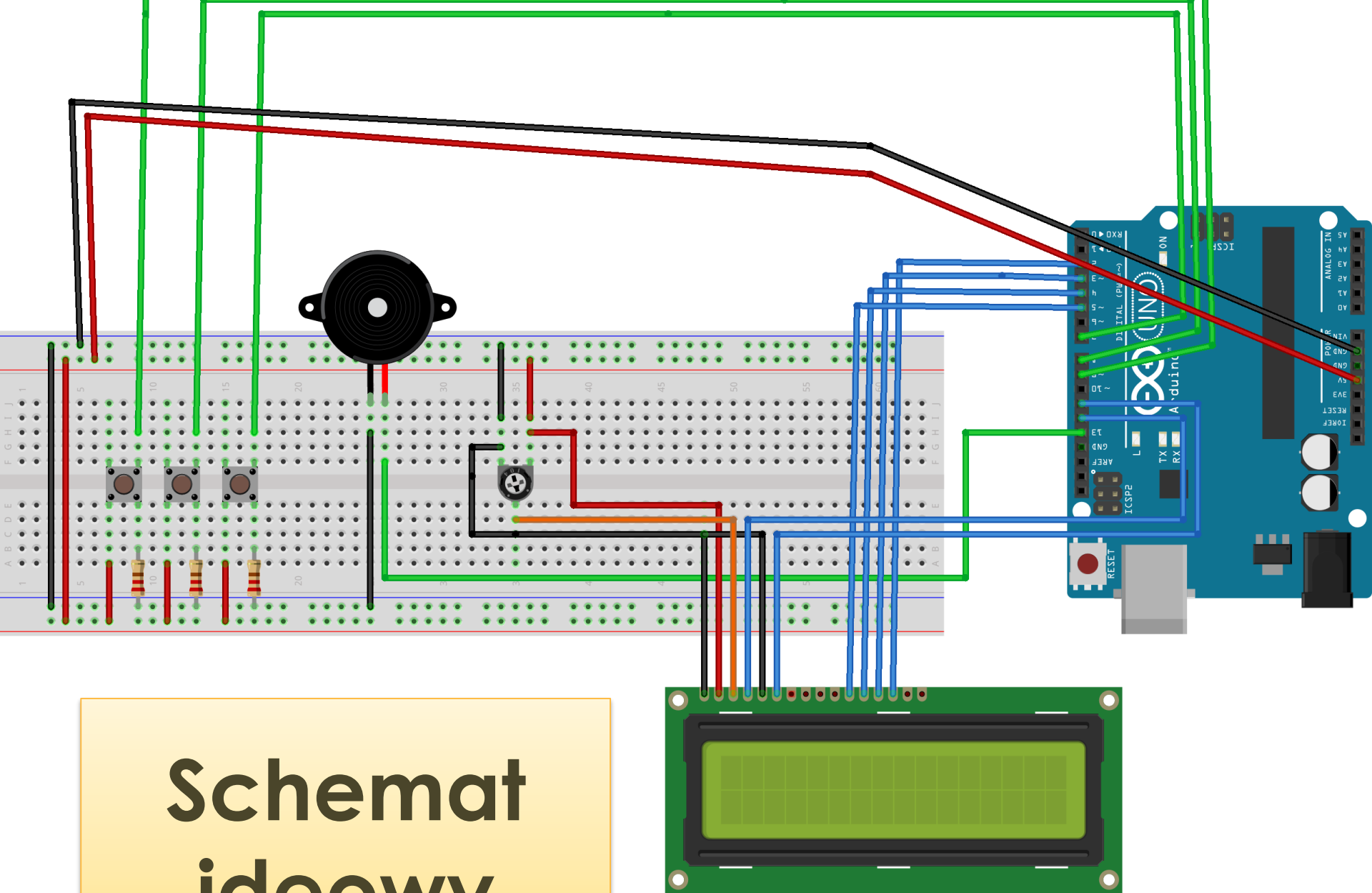
- ❖ Zakres możliwych do wygenerowania częstotliwości to 31 do 65535Hz.

Lista części użytych w projekcie

- ☐ Arduino Uno
- ☐ Zworki
- ☐ Buzzer
- ☐ Wyświetlacz LCD (w typie HD44780)
- ☐ 3 przyciski
- ☐ 3 rezystory
- ☐ Potencjometr liniowy

Schemat - Eagle





Schemat
ideowy



AN OPEN PROJECT WRITTEN, DEBUGGED AND SUPPORTED
BY MASSIMO BANZI, DAVID CUARTIELLES, TOM IGOE,
GIANLUCA MARTINO AND DAVID MELLIS



Projekt od strony kodu

Początek kodu

```
#include <LiquidCrystal.h>
#include "pitches.h"

#define OCTAVE_OFFSET 0
#define MAX_SONG_NUMBER 16

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const char* tytuły[] = {"Game of Thrones", "Wśród nocnej ciszy", "Ona t

const int playPin = 8;
const int prevPin = 9;
const int nextPin = 7;
```

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
```

pitches.h

Zapis melodii

```
int thrones_melody[] = {  
NOTE_G3 , NOTE_C3, NOTE_DS3, NOTE_F3, NOTE_G3 ,NOTE_C3, NOTE_DS3, NOTE_F3, NOTE_D3,  
NOTE_F3 , NOTE_B2, NOTE_D3, NOTE_DS3, NOTE_F3, NOTE_B2, NOTE_DS3, NOTE_D3,NOTE_C3 }; //18  
  
int thrones_duration[] = {  
3, 3, 1, 1, 2, 2, 1, 1, 9,  
3, 3, 1, 1, 2, 2, 1, 1, 9};
```

Sekcja setup()

```
void setup()  
{  
    //buzzer.begin(16, 2);  
    //tone.begin(BUZZER_PIN);  
    pinMode(playPin, INPUT);  
    pinMode(nextPin, INPUT);  
    pinMode(prevPin, INPUT);  
    Serial.begin(9600);  
    lcd.print("Starting up!");  
    lcd.autoscroll();  
}
```

Sekcja loop() z kodem umożliwiaj ącym odegranie melodii

```
void loop()  
{  
    int playState = digitalRead(playPin);  
    int prevState = digitalRead(prevPin);  
    int nextState = digitalRead(nextPin);  
  
    if (playState) //jesli wcisniety przycisk play  
    {  
        lcd.begin(16, 2);  
        lcd.print("hello, world!");  
        switch(song)  
        {  
            case 0:  
            {  
                lcd.begin(16, 2);  
                lcd.print("Wlasnie gram:");  
                lcd.setCursor(0, 1);  
                lcd.print(tytuly[song]);  
                for (int thisNote = 0; thisNote < 18; thisNote++)  
                {  
                    int noteDuration = 120 * noteDurations[thisNote];  
                    tone(10, melody[thisNote],noteDuration);  
                    delay(noteDuration +30);  
                }  
                break;  
            }  
        }  
    }  
}
```



```
if (prevState) //jesli wcisniety przycisk play
{
    lcd.begin(16, 2);
    lcd.print("Numer melodii:");
    song = --song%MAX_SONG_NUMBER;
    if (song<0)
    {
        song=MAX_SONG_NUMBER + song;
    }
    lcd.print(song);
    lcd.setCursor(0, 1);
    lcd.print(tytuly[song]);
    delay(400);
}
```

```
if (nextState) //jesli wcisniety przycisk play
{
    lcd.begin(16, 2);
    lcd.print("Numer melodii:");
    song = ++song%MAX_SONG_NUMBER;
    lcd.print(song);
    lcd.setCursor(0, 1);
    lcd.print(tytuly[song]);
    delay(400);
}
```

Obsługa
przycisków
prevState i
nextState

Podsumowanie

Układ działa zgodnie z początkowymi założeniami.

Podczas wykonywania projektu musiałem się zmierzyć m.in. z takimi problemami jak:

- Przechowywanie piosenek w formie dostępnej do ich odtworzenia
- Podłączenie całego układu, dopasowanie rezystorów, rozmieszczenie logistyczne podzespołów na płytce stykowej