



Technika mikroprocesorowa 2 - projekt

Temat projektu: Wykonanie kontrolera DJ MIDI w oparciu o mikrokontroler Cortex-M4

Spis treści

1. Wstęp	3
Terminologia.....	3
2. Kontrolery MIDI oraz związane protokoły.....	4
2.2 Wiadomości MIDI	4
2.2 Wiadomości System Exclusive	4
2.3 Kontrolery DJ MIDI	5
3. Realizacja sprzętowa	6
4. Struktura kodu	8

1. Wstęp

Końcowym celem projektu jest zaprojektowanie kontrolera MIDI, który będzie dobrym wyborem w sytuacjach, gdzie funkcje profesjonalnego kontrolera nie będą miały zastosowania. Z tego powodu po finalizacji kodu oraz dokumentacji projekt zostanie przeniesiony z płytki prototypowej na płytkę drukowaną, w tym z przeniesieniem wszystkich wykorzystanych układów scalonych. Zostanie również wykonana obudowa i wydrukowana za pomocą drukarki 3D.

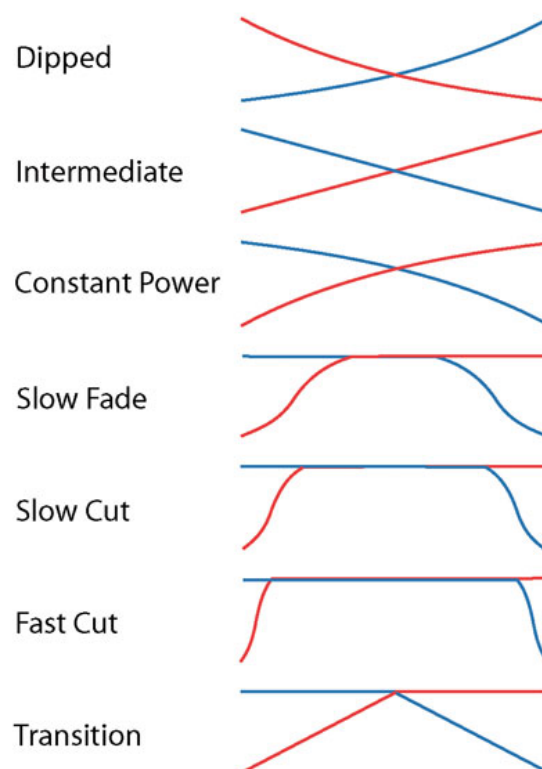
Ważnym elementem naszego projektu jest działanie bez sterowników pod każdym systemem operacyjnym – zarówno jak i jako urządzenie MIDI, jak i interfejs USB audio.

Terminologia

BPM – Beats Per Minute – określa tempo piosenki,

Deck – element programów DJ i kontrolerów do którego ładuje się ścieżki. Typowo używa się 2 lub 4 i wszystkie są od siebie niezależne,

Crossfader – element określający poziom mieszania ścieżek z 2 różnych decków zgodnie z zadaną charakterystyką:



Rysunek 1 – najpopularniejsze charakterystyki crossfaderów

Źródło: <https://math.stackexchange.com/questions/4621/simple-formula-for-curve-of-dj-crossfader-volume-dipped>

2. Kontrolery MIDI oraz związane protokoły

2.2 Wiadomości MIDI

Na profesjonalnym poziomie prezentacji muzycznych protokołów MIDI – Musical Instrument Digital Interface – jest podstawą tworzenia urządzeń do zarządzania wieloma związanymi peryferiami, tj. światła, wytwornice dymów i tym podobne. Uniwersalność protokołu sprawia, że każde urządzenie implementujące go jest plug-and-play, a konfiguracja wygląda tak samo.

Protokół został stworzony z myślą o cyfryzacji sygnałów z instrumentów muzycznych, więc używana terminologia może wydawać się <dziwna>.

Każde urządzenie implementujące protokół MIDI wysyła i odbiera pakiety o strukturze:

Typ wiadomości	Identyfikator wiadomości	Wartość
Rozróżniamy 3 podstawowe typy wiadomości: <ul style="list-style-type: none">• Control Change (CC)• Note• Pitch Bend Wiadomości Pitch Bend pozbawione są wartości, natomiast CC oraz Note nie różnią się niczym. Pozwalają natomiast na podwojenie możliwości wysłania unikatowych wiadomości.	Identyfikują konkretną funkcjonalność urządzenia wysyłającego wiadomość. Ze względu na zakres możliwości wykorzystania urządzeń MIDI mapowanie jest arbitralne, lecz istnieją standardy, do których można się dostosować.	Z zakresu od 0 do 127.

Istnieją rozszerzenia protokołu wzbogacające go o nowe funkcjonalności, lecz ze względów kompatybilności podstawowa wersja jest najbardziej rozpowszechnioną.

2.2 Wiadomości System Exclusive

Ten typ wiadomości jest dla nas interesujący, ponieważ używany przez nas program oferuje informacje typowo niedostępne w programach DJ właśnie za pomocą wiadomości SysEx. Struktura wiadomości jest bardziej skomplikowana¹, lecz pozwala to na tworzenie funkcji nieosiągalnych przy użyciu protokołu MIDI.

Niestety oznacza to również, że nie mamy możliwości tworzenia własnych *mappingów* – przypisać funkcji do identyfikatora wiadomości. Te wiadomości są rezultatem współpracy producentów sprzętu oraz deweloperów oprogramowania. Na szczęście nasz program DJ, Native Instruments Traktor Pro 3, w przypadku jednego mappingu pozwala na przypisanie dowolnego urządzenia MIDI jako odbiornik. Mapping został stworzony do kontrolera Numark Mixdeck i nie jest dostępna jego dokumentacja, lecz udało nam się rozszyfrować kluczowe elementy, na których nam zależało.

¹ <https://www.midi.org/specifications-old/item/table-4-universal-system-exclusive-messages>

2.3 Kontrolery DJ MIDI

Głównym zadaniem kontrolerów DJ MIDI jest oferowanie fizycznej kontroli nad wybranym programem. Zazwyczaj mają one również interfejsy audio, eliminując potrzebę zewnętrznych wzmacniaczy i mikserów. Z tych powodów w naszym projekcie chcieliśmy zaimplementować:

- Przyciski play/pause,
- Przyciski cue – pozwalają na odstuch muzyki przez kanał dźwiękowy inny niż główny,
- Potencjometry wzmocnienia/głośności,
- Potencjometr crossfadera – kontroluje mieszanie muzyki z różnych decków,
- Przyciski ładowania – załadują wybrany plik do decku,
- Przyciski *modifier* – pozwalają na zmianę funkcjonalności innych wejść w zależności od stanu,
- Encoder przeglądarki – pozwala na szybkie i dokładne przeglądanie biblioteki muzyki,
- Diody play/pause,
- Diody końca ścieżki – ostrzegające o zbliżającym się końcu aktualnie odtwarzanego utworu,
- Wskaźniki głośności – pokazujące poziom głośności granej muzyki,
- Wskaźnik fazy – pokazujący synchronizację między dwoma grającymi deckami.

Powyższą funkcjonalność możemy osiągnąć bez użycia wiadomości z protokołu System Exclusive. Gdy go użyjemy, nasz kontroler zyskuje możliwości wyświetlania właściwości załadowanych ścieżek:

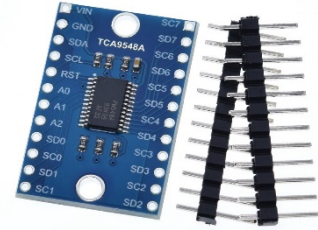
- Autorów i tytułu,
- BPM,
- Czas.

3. Realizacja sprzętowa

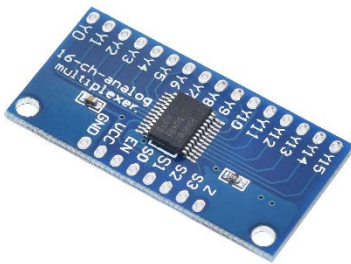
Do realizacji planowanych funkcji wykorzystaliśmy następujące elementy:



Wyświetlacz I2C SSD1306



Multiplexer I2C TCA9548A



Multiplexer CD74HC4067

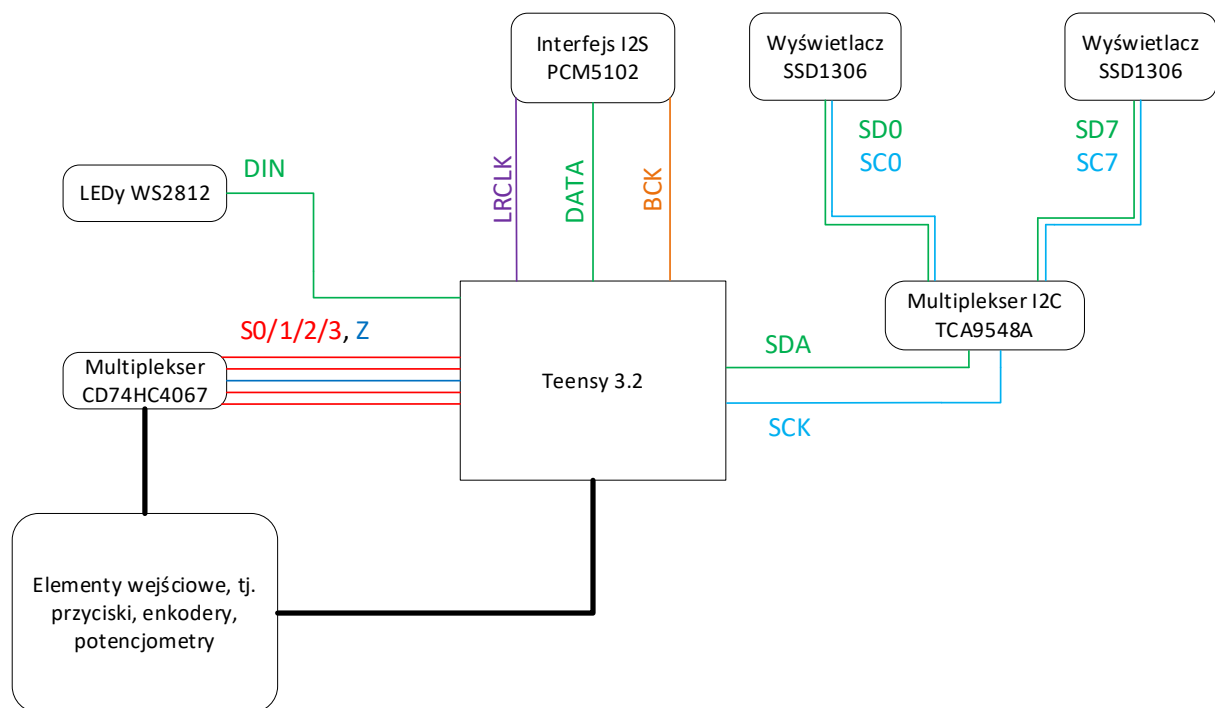


Interfejs I2S PCM5102



Adresowalne diody LED WS2812





Rysunek 2 – uproszczony schemat połączeń

4. Struktura kodu

Ważnym elementem naszego projektu jest biblioteka [Control Surface](#). Ułatwia ona tworzenie kontrolerów klasy MIDI, generalizując odbieranie i wysyłanie wiadomości z różnych elementów. Najważniejszą jej częścią jest możliwość dodania własnego handlera wiadomości System Exclusive, pozwalając nam na przetwarzanie i przechowywanie dodatkowych informacji. Tym się zajmuje klasa TrackDataHandler i jedna jej instancja zawiera informacje o jednym decku (A lub B). Diody WS2812 połączone szeregowo kontrolujemy jako jeden długi pasek za pomocą biblioteki [FastLED](#). Jest ona szybsza i oferuje więcej opcji od innych bibliotek, np. Adafruit_Neopixel.

Wyświetlacze sterowane są przez biblioteki Adafruit_SSD1306 oraz Adafruit_GFX ze względu na prostotę użycia. Wyświetlacze są odświeżane tylko wtedy, kiedy klasa TrackDataHandler zwróci informację, że nowe dane są dostępne. Robimy to, ponieważ pojedyncza klatka jednego wyświetlacza jest rysowana przez około 30 milisekund. Chcąc odświeżać je przy każdej pętli wprowadzamy bardzo duże opóźnienie w odbieraniu i wysyłaniu danych, ponieważ dzieje się to wszystko w jednym, liniowym wątku.

Kod obsługujący interfejs I2S został wygenerowany automatycznie przez [Audio System Design Tool for Teensy Audio Library](#). Jest to proste połączenie kanałów USB audio oraz sprzętowych portów I2S na płycie Teensy 3.2.