

Organy - Dokumentacja

Alicja Kałuża i Katarzyna Żuchowska

21 stycznia 2024

1 Opis projektu

Celem projektu było stworzenie działających organów składających się z ośmiu klawiszy, wyświetlających się na dotykowym ekranie. Dodatkowo znajduje się w nich możliwość zapisu maksymalnie 100 dźwięków oraz ich odtworzenia.

Komunikacja mikrokontrolera z głośnikiem jest przeprowadzona przez zewnętrzny przetwornik cyfrowo-analogowy.

Na wyświetlaczu pokazywana jest klawiatura złożona z 8 klawiszy organek. Naciskanie klawiszy na ekranie dotykowym powoduje odegranie dźwięku o częstotliwości odpowiadającej danemu klawiszowi, a dłuższe przytrzymanie go spowoduje dłuższy czas odgrywania dźwięku. Po naciśnięciu przycisku "REC" rozpocznie się zapisywanie melodii o maksymalnej liczbie dźwięków (100). Naciśnięcie przycisku "PLAY" spowoduje odtworzenie tej melodii.

2 Instrukcja użytkowania

Po naciśnięciu oraz puszczeniu jednego z 8 klawiszy zostanie odgany dźwięk o odpowiedniej częstotliwości przez czas równy czasowi naciskania tego klawisza.

Po naciśnięciu znajdującego się po lewej stronie organów przycisku "REC" rozpocznie się nagrywanie. Oznacza to że kolejne odgrywane przez nas dźwięki zostaną również zapisane. Takich dźwięków możemy zapisać maksymalnie 100. Następnie po naciśnięciu przycisku "PLAY", który pojawi się na miejscu przycisku "REC" nastąpi odegranie wcześniejszej melodii. Przycisk "PLAY" będzie również zamieniał się ponownie w przycisk "REC".

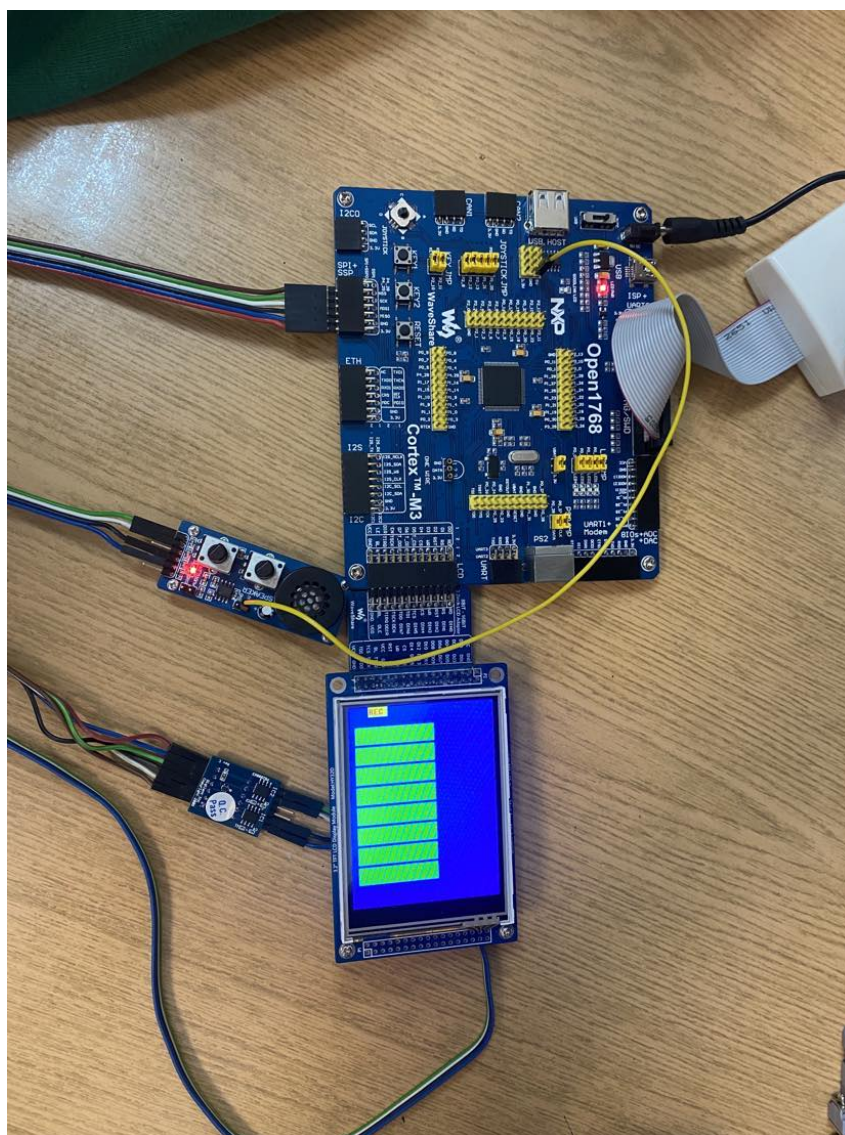
Używany przez nas ekran nie jest dokładny. Z tego powodu przyciski mogą nie zawsze działać poprawnie ponieważ miejsce dotyku może być błędnie odczytane. Należy więc spodziewać się, że czasami działanie organów może być nieprzewidywane. Ekran lepiej działa na środku, dlatego przyciski znajdujące się bliżej krawędzi częściej nie działają w pełni poprawnie.

Sygnał, który generujemy, a następnie wysyłamy na głośnik jest sygnałem prostokątnym dlatego dźwięk jaki słyszymy nie brzmi "najładniej".

Przyciski mogą nie zawsze działać. Jeżeli przycisk nie zadziałał należy go przycisnąć ponownie do skutku.

3 Sposób podłączenia elementów

Urządzenie składa się z następujących komponentów: - płytki LPC 1768, - wyświetlacza LCD, - głośnika, - zewnętrznego przetwornika cyfrowo-analogowego obsługiwane go protokołem SPI.



Rysunek 1: Odpowiednio podłączone elementy

Przetwornik cyfrowo analogowy podłączony jest do płytki za pomocą pinów P12, P13, P14, P15, P16, P17, P18. Piny po jego jednej stronie łączy się kolejno z następującymi pinami na płytce: "3,3V", "GND", "MISO", "MOSI", "SCK", "NSS".

Na prztworniku znajdują się w rzeczywistości dwa przetworniki z tego powodu używamy jedynie dwóch z czterech dostępnych pinów. Są one podłączone "GND" do "GND" oraz "VCC" do "VCC", a sposób podpięcia pozostałych dwóch przewodów zaprezentowany jest na Rysunku 1.

4 Opis algorytmu

Z ekranu dotykowego pobieramy pięć kolejnych par współrzędnych, które są odczytywane jako dotknięcie z ekranu dotykowego. Następnie obliczamy ich średnią. Robimy tak z powodu niedokładności odczytu z ekranu. Kolejnym krokiem jest sprawdzenie miejsca dotyku po uprzednim przeliczeniu współrzędnych. Jeżeli następuje dotknięcie jednego z klawiszy następuje włączenie timera oraz po puszczeniu następuje odegranie odpowiedniego dźwięku o odpowiedniej długości trwania. Jeżeli uprzednio zostało włączone nagrywanie zostanie dodatkowo zapisany dźwięk który zagrano do tablicy. Natomiast jeżeli naciśnięto przycisk "REC" zostaje uruchomiona opcja nagrywania. Po naciśnięciu przycisku "PLAY" zostają kolejno odegrane dźwięki zapisane w tablicy. W sytuacji, gdy dotknięto w miejscu gdzie nie znajduje się żaden z przycisków nic się nie wydarzy.

5 Opis kodu

5.1 int main()

Zostają zainicjalizowane wszystkie elementy znajdujące się w programie. następnie na ekranie rysuję się niebieski prostokąt będący tłem, po nim pojawia się napis "REC" oraz 8 klawiszy organów. W nieskończonej pętli while(1) zostaje wywoływana funkcja "check()" która obsługuje działanie organów.

5.2 void TIMER0_IRQHandler(void)

Działanie jednego zegara. Jest to jeden z elementów potrzebnych do odegrania dźwięku o odpowiedniej długości i częstotliwości.

5.3 void TIMER1_IRQHandler(void)

Działanie drugiego zegara. Służy do liczenia czasu nacisku klawisza.

5.4 void initSPI(void)

Uruchomienie interfejsu SPI.

5.5 void sendByteSPI(uint8_t info)

Funkcja obsługująca komunikację przez SPI, wysyła ona 1 bajt. Wysyła ona kolejno bity kontrolne i bity danych.

5.6 void SysTick_Handler(void)

Obsługuje działanie zegara za pomocą zegaru systemowego.

5.7 void conf(void)

Konfiguruje zegar systemowy.

5.8 void delay2(int d)

Obsługa opóźnień za pomocą msTicks2.

5.9 void delay(int d)

Obsługa opóźnień za pomocą msTicks.

5.10 void zagraj2(int f,int time)

Wywoływanie funkcji wysyłającej informację poprzez SPI. Generuje sygnał prostokątny o odpowiedniej amplitudzie odpowiadającej odpowiedniej częstotliwości przez odpowiedni czas.

5.11 void rysuj(char literka, uint16_t x, uint16_t y)

Rysowanie odpowiedniej litery na odpowiednich współrzędnych podanych jako argumenty.

5.12 void rysujprostokat(uint16_t x, uint16_t y,uint16_t xx, uint16_t yy,uint16_t color)

Rysowanie prostokąta. Jako argumenty są przekazywane współrzędne jednego z rogów, wymiary oraz kolor.

5.13 void EINT3_IRQHandler() include "LPC17xx.h"

Nie używana przez nas funkcja do obsługi organów. Nie obsługuje ona czasu nacisku klawisza, każdy dźwięk jest odgrywany tak samo długo.

5.14 void init_GPIO()

Inicjalizacja GPIO.

5.15 void initTimer0_2()

Inicjalizacja timera 2.

5.16 void initTimer0_1()

Inicjalizacja timera 1.

5.17 void check()

Implementacja wyżej opisanego algorytmu.

6 Użyte biblioteki

- LPC17xx.h - obsługa płytki open1768
- LCD_ILI9325.h - obsługa wyświetlacza dotykowego
- Open1768_LCD.h - obsługa wyświetlacza na płytce open1768
- asciiLib.h - biblioteka obsługująca rysowanie liter na ekranie
- PIN_LPC17xx.h - obsługa pinów
- stdlib.h - standardowa biblioteka dla języka C
- TP_Open1768.h - kolejna biblioteka potrzebna do obsługi płytki
- Driver_SPI.h - obsługa SPI

7 Druga wersja projektu

Posiadamy również drugą wersję projektu (funkcja void EINT3_IRQHandler()). Aktualna wersja kodu nie jest idealna, pojawiają się w niej błędy częściej niż byśmy chcieli. Z tego powodu zostawiliśmy drugą wersję. Jest ona w pełni funkcjonalna, oprócz czasu odgrywania dźwięku - każdy dźwięk jest odgrywany tak samo długo.