

Bazy Danych

prowadzący: dr inż. Grażyna Kurpińska

Projekt: Baza wycieczkowa z GUI

O projekcie

Celem projektu było stworzenie programu z graficznym interfejsem, który korzystając z bazy danych (utworzonej także w ramach tego projektu) umożliwia użytkownikowi zaplanowanie trasy wycieczkowej oraz zapisanie utworzonej wycieczki. Projekt posiada także opcję edycji zaplanowanych wycieczek. Projekt korzysta z bazy ElephantSQL, co umożliwia użytkownikowi dostęp do danych na różnych urządzeniach.

Wygląd aplikacji (GUI):



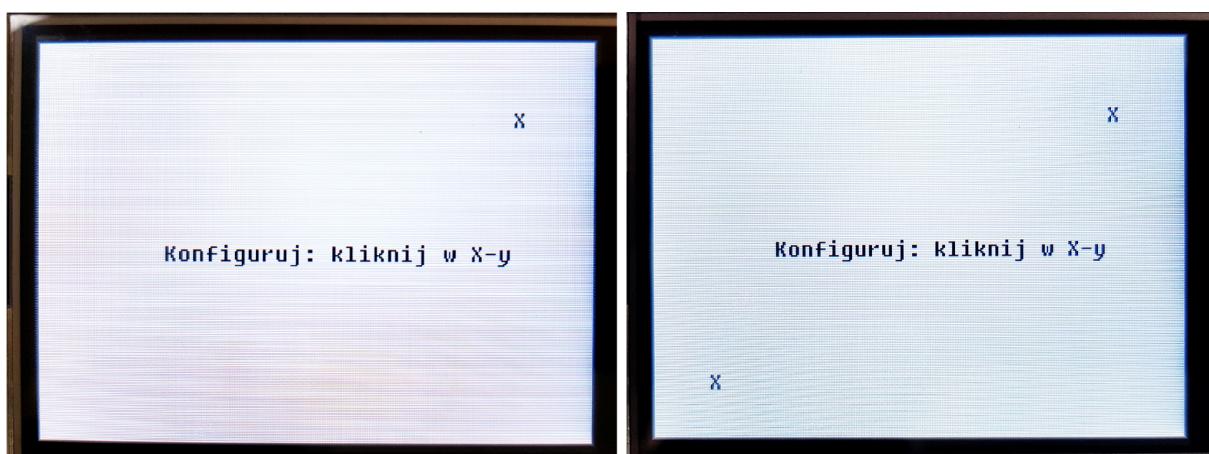
Rysunek.1.

Opis:

1. Przycisk przycisania
2. Przycisk podgłasznania
3. Klawiatura grająca

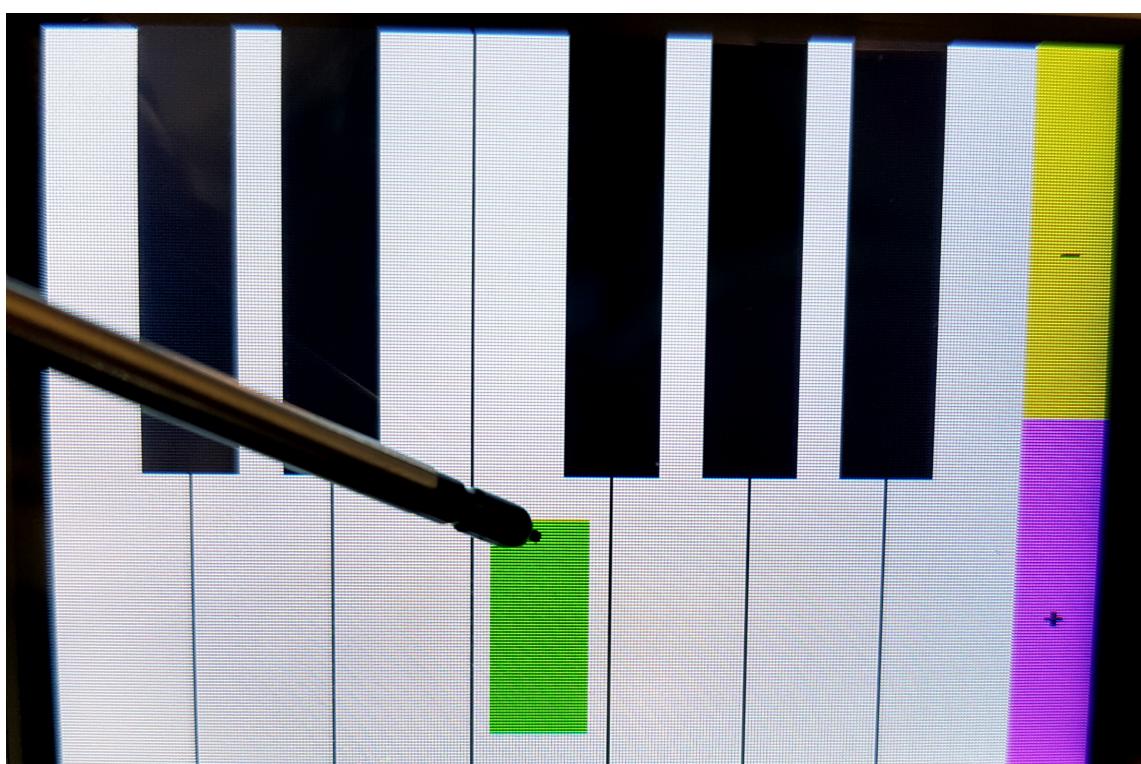
Instrukcja

Po uruchomieniu należy skalibrować ekran dotykowy poprzez dotknięcie kolejno 2 miejsc oznaczonych literą 'X' na wyświetlaczu, jak przedstawiono poniżej (Rys.2.). Aby dokonać poprawnej konfiguracji należy przytrzymać każde z miejsc co najmniej przez 1 sekundę.



Rysunek.2.

Czynność ta pozwala przejść do ekranu głównego aplikacji (Rysunek.1). Za pomocą rysika użytkownik może wybrać, który dźwięk ma być zagrany, naciskając na odpowiedni klawisz. Po dotknięciu, wybrany dźwięk (klawisz) zostanie podświetlony kolorem zielonym, jak widać to na poniższym rysunku (Rysunek.3.).



Rysunek.3.

Dłuższe przytrzymanie klawisza powoduje dłuższą generację dźwięku - podniesienie rysika od ekranu aplikacji powoduje przerwanie wydobywania danego dźwięku.

Nie ma możliwości wygenerowania dwóch (lub więcej) dźwięków równocześnie.

Nie ma możliwości przesuwania rysika wzdłuż klawiatury w celu generowania tzw. glissanda.

Po stronie prawej, dotknięcie przycisku '-' (Rysunek.1, nr 1) powoduje zmniejszenie głośności wydobywania dźwięków z dotykowego keyboardu, natomiast przycisk '+' (Rysunek.1, nr 2)- zwiększenie głośności aplikacji. Istnieje 10 poziomów głośności - wyjściowo aplikacja przyjmuje poziom średni (poziom 5). Jedno kliknięcie danego przycisku powoduje zmniejszenie lub zwiększenie głośności o jeden poziom.

Opis sytuacji problematycznych

Program losowo może nie wygenerować przerwania dźwięku podczas oderwania rysika. W takim przypadku, należy ponownie nacisnąć zacięty przycisk celem jego odblokowania.

Jest to błąd sprzętowy, który (mimo konfiguracji na zbocze narastające i opadające) losowo ma problem z generacją przerwania.

Szczegółowy opis algorytmu

Projekt zrealizowany został w środowisku µVision5.

Aplikacja bazuje głównie na układzie przerwań NVIC (Nested Vectored Interrupt Controller), peryferyjnym liczniku TIMER0 oraz peryferyjnym układzie DAC (Digital-to-Analog Converter). Do obsługi zdarzeń na dołączonym dotykowym wyświetlaczu LCD został wykorzystany układ przerwań zewnętrznych EINT3 (obsługa przerwań na pinie P0.19 dla zbocza opadającego i narastającego). Dodatkowo w projekcie wykorzystany został peryferium UART0 w celu łatwiejszego debugowania programu dzięki ciągłej komunikacji z komputerem.

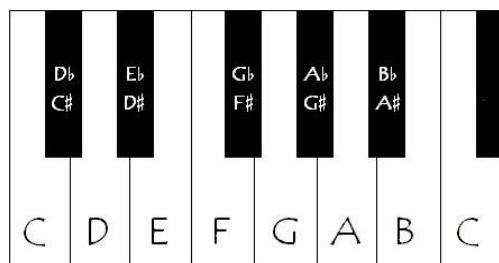
Działanie:

Program poprzez generowanie przerwania EINT3 przy dotknięciu dołączonego ekranu dotykowego (lub odjęciu rysika od ekranu) uruchamia licznik TIMER0, który w równych odstępach czasowych generuje przerwania wysyłające do DACa kolejne wartości komórek tablicy SoundWave ('fala dźwiękowa'), czyli kolejne wartości funkcji cos, której zadaniem jest imitowanie fali dźwiękowej.

Aplikacja wykorzystując funkcje wbudowane jest w stanie odczytać współrzędne przerwania (dotknięcie ekranu), implementuje też funkcję uśredniania odczytu w celu zminimalizowania niepoprawnych odczytów.

W algorytmie przyjęte zostało założenie, iż współrzędne dotyku oraz współrzędne wyświetlacza są uzależnione relacją liniową, w związku z czym została zaimplementowana kalibracja dwupunktowa.

Regulowanie częstotliwości dźwięku odbywa się poprzez zwiększenie lub zmniejszenie częstotliwości generowania przerwań przez TIMER0. Aplikacja posiada możliwość wytworzenia 12 dźwięków w zakresie 262 Hz - 494 Hz, które odpowiadają kolejnym tonom w zakresie jednej oktawy: od C4 do H4. Kolejne częstotliwości odpowiadają kolejnym klawiszom jak poniżej:



Rysunek.4.

Nazwa dźwięku	Częstotliwość (Hz)
C4	262
C#4	277
D4	294
D#4	311
E4	330
F4	349
F#4	370
G4	392
G#4	415
A4	440

A#4	466
H4	494

Tablica.1.

W aplikacji wykorzystana została obserwacja, iż częstotliwości kolejnych tonów określone są przez zależność: $f(n+1) = \sqrt[12]{2} * f(n)$ - zatem $266 * \sqrt[12]{2} \approx 277$, itd.

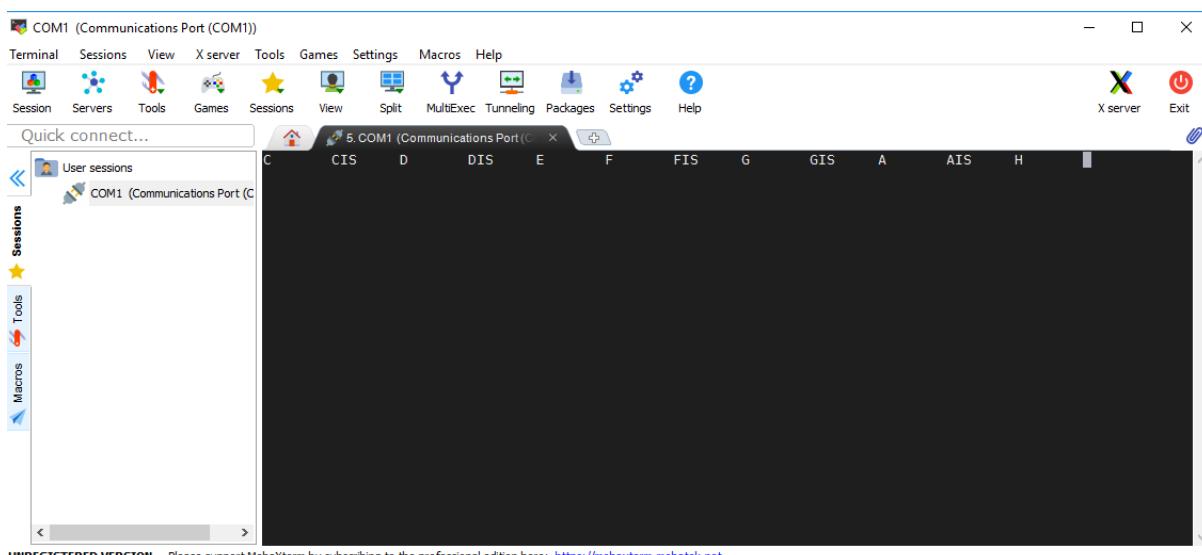
'Match Register' dla licznika TIMERo, czyli wartość odpowiadająca za częstotliwość generowania przerwań i wysyłania wartości do DAC, ustalona została poprzez podzielenie PCLK (częstotliwość zegara peryferyjnego tj. domyślnie częstotliwość zegara procesora - CCLK/4) przez wartość żądanej częstotliwości dźwięku pomnożonej przez rozmiar tablicy *SoundWave*, co pozwala na generowanie przez DAC całej fali (wszystkich wartości z tablicy) z częstotliwością zgodną z powyższą tabelą.

Podgłaśnianie i przyciszanie wykorzystane w aplikacji polega na przemnożeniu poszczególnych wartości wysyłanych do DACa, tak aby spowodować rozciągnięcie funkcji cos z tablicy *SoundWave* w pionie, tym samym zwiększenie (lub zmniejszenie) amplitudy nadawanej fali.

Pozostałe obserwacje

Docelowo istnieje możliwość ulepszenia kalibracji, tak aby uwzględnić ewentualne rotacje dotyku względem wyświetlacza.

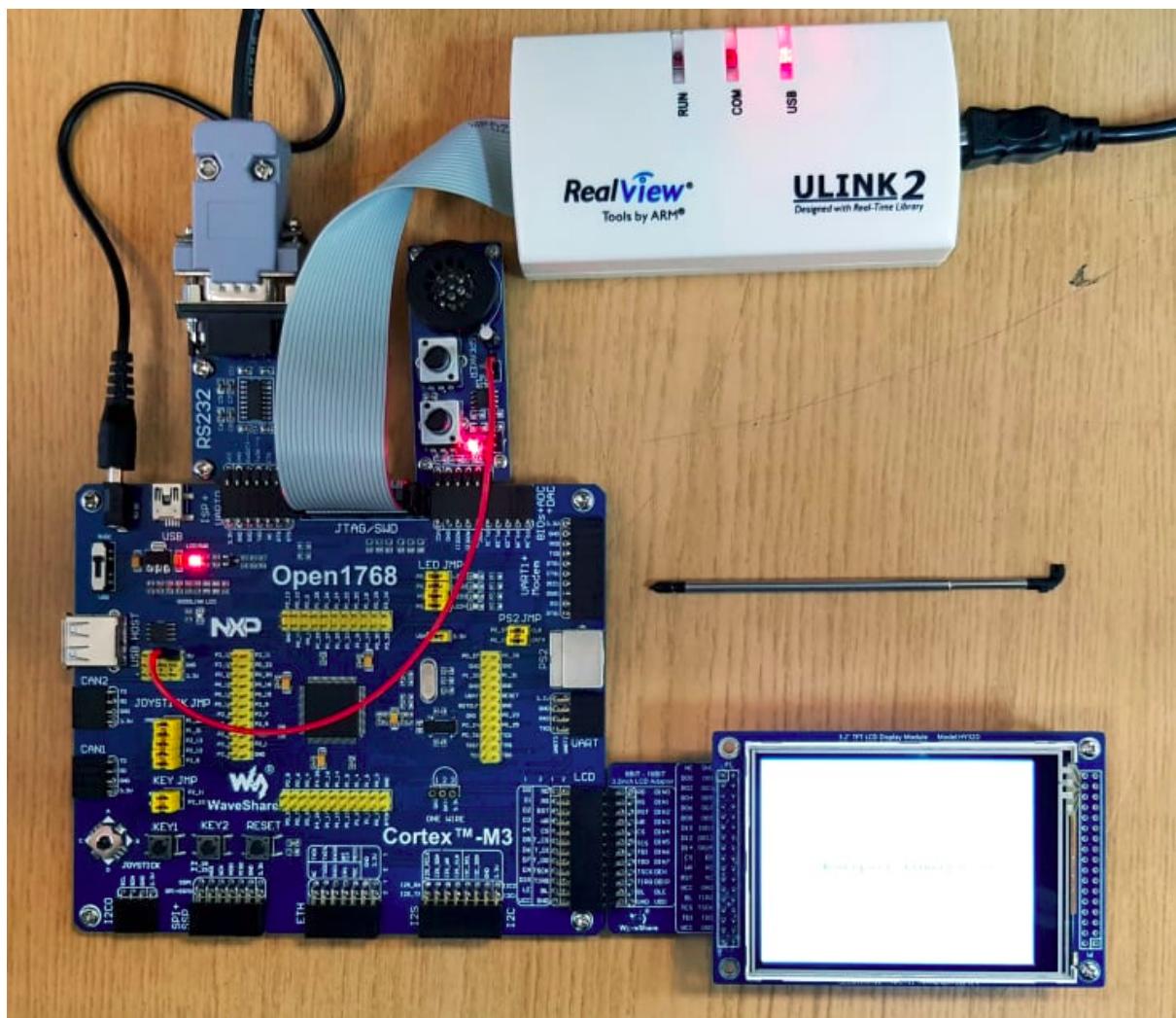
Aplikacja posiada dodatkową funkcjonalność - możliwość śledzenia granych klawiszy na podłączonym komputerze. Pierwotnie peryferium UART zostało skonfigurowane w celu łatwiejszego debugowania programu, jednak z czasem stało się wartościowym dodatkiem do projektu. Dzięki tej funkcji użytkownik po podłączeniu z komputerem jest w stanie zobaczyć i odtworzyć swoją twórczość. Poniżej (Rysunek.5.) zaprezentowane zostało działanie dodatkowej funkcjonalności.



Rysunek.5.

Wykorzystany sprzęt

Projekt wykorzystuje mikrokontroler LPC1768, programator i debugger ULINK2, moduł 3.2 cal ekranu dotykowego LCD, peryferyjny głośniczek (Analog Test Board), peryferium UART (RS232 Board), rysik - do obsługi wyświetlacza.



Rysunek.6.

Twórcy projektu

Aleksandra Krzemińska
Przemysław Nowak