

Data oddania: _____

Ocena: _____

Krzysztof Klimczak 143001

Dominik Żychliński 143153

Michał Kawski 142997

Odtwarzacz Wave

Spis treści

| | |
|--|----|
| 1. Opis działania programu..... | 3 |
| 1.1. Sterowanie aplikacją..... | 3 |
| 2. Wykorzystane możliwości mikrokontrolerów..... | 4 |
| 2.1. Wykorzystanie portów I/O..... | 4 |
| 2.1.1. Inicjalizacja..... | 5 |
| 2.1.2. Przesyłanie danych..... | 5 |
| 2.2. Obsługa szyny danych SPI..... | 5 |
| 2.2.1. Inicjalizacja..... | 5 |
| 2.2.2. Przesyłanie danych..... | 6 |
| 2.3 Przetwornik cyfrowo-analogowy DAC..... | 6 |
| 2.3.1. Inicjalizacja..... | 6 |
| 2.3. Przetwornik analogowo-cyfrowy ADC..... | 6 |
| 2.3.1. Inicjalizacja..... | 7 |
| 2.4 Timer, czyli zegar..... | 7 |
| 2.4.1. Inicjalizacja..... | 7 |
| 2.5 UART, czyli konsola..... | 7 |
| 2.5.1. Inicjalizacja..... | 7 |
| 3. Wykorzystane urządzenia peryferyjne..... | 8 |
| 3.1. Ekran LCD..... | 8 |
| 3.2. Joystick..... | 8 |
| 3.3. Akcelerometr..... | 9 |
| 3.4. Karta pamięci micro SD..... | 9 |
| 4. Algorytm programu..... | 9 |
| 5. Analiza FMEA/FMECA..... | 9 |
| 6. Wykorzystany sprzęt i oprogramowanie..... | 10 |
| 7. Dołączone noty katalogowe..... | 10 |
| 8. Podział pracy..... | 10 |

1. Opis działania programu

Po uruchomieniu aplikacji następuje inicjalizacja wszystkich używanych komponentów. Po zakończeniu tej czynności następuje próba dostępu do czytnika kart pamięci mikro SD oraz do samej karty. Jeśli aplikacja nie napotka żadnego błędu, wykonywany jest odczyt zawartości karty, tworzona jest lista plików (obsługiwane są przy tym tylko pliki z rozszerzeniem „wav” znajdujące się w katalogu głównym karty. W aplikacji jest obsługiwane do 256 plików wav (tyle może maksymalnie pozycji liczyć lista plików). Po wykonaniu tej czynności na ekranie pojawi się nazwa pierwszego pliku (ze względu na zastosowaną bibliotekę nazwa składa się z 8 znaków + 3 znaki + znak kropki pomiędzy nazwą a rozszerzeniem). W tym momencie aplikacja oczekuje na interakcje użytkownika. Co do samych plików wave, wymagane jest by były próbkowane 8000Hz, zaś wartość była zapisana w 1 bajcie (inne pliki są obsługiwane, ale nie odtwarzane płynnie, następuje „czkawka” gdyż odczyt z karty pamięci trwa za długo), ilość kanałów 1 (mono), stereo nie jest obsługiwane, dodatkowo ważne jest by plik nie zawierał żadnych dodatkowych atrybutów, np. informacji o wykonawcy (czyli najprostsza postać formatu) [jeśli będzie zawierać zostaną załadowane nieprawidłowe wartości odnośnie prędkości odtwarzania i bajcie od którego należy zacząć grać, w takim wypadku uzyskamy nieskładne dźwięki na wyjściu.

Warto też wspomnieć że mając dostęp do jakiejś aplikacji terminalowej, dzięki konfiguracji UART na mikrokontrolerze możliwy jest odczyt bardziej szczegółowych informacji zwracanych przez „płytkę” (np. listę plików, czy tekstowy komunikat o napotkanym błędzie). Parametry do podłączenia:

- prędkość (baud rate): 115200 bps
- bity stopu: 1
- parzystość: brak
- długość słowa: 8N1 (8 bitów)

1.1. Sterowanie aplikacją

Sterowanie aplikacją odbywa się za pomocą przycisków na płytce LPC2138 lub dżojstika na Expansion Board. Czynności są wykonywane oczywiście po wciśnięciu danego przycisku, dodatkowo fakt wciśnięcia przycisku jest zgłaszany zapaleniem się diody umieszczonej nad przyciskiem:

Przyciski:

P1.20 → poprzedni, wybór pliku o pozycji wcześniej niż aktualny na liście, w przypadku początku listy, następuje przeskok na jej koniec (zapala się dioda P1.16)

P1.21 → play(graj), następuje odtworzenie pliku, bądź wznowie grania w przypadku wciśnięcia pauzy (zapala się dioda P1.17)

P1.22 → pauza, zatrzymanie aktualnie odtwarzanego utworu, pozycja zatrzymania jest zapamiętywana (zapala się dioda P1.18)

P1.23 → stop, definitywne zatrzymanie odtwarzania, po wciśnięciu play granie zaczyna się od początku pliku (zapala się dioda P1.19)

P0.14 → następny, przechodzi do następnego pliku (w przypadku końca listy, następuje przeskok

do jej początku) (zapala się dioda P0.14)

Dżojstik:

lewo → poprzedni, czynność jak dla odpowiedniego przycisku (zapala się dioda P1.16)

prawo → następny, czynność jak dla odpowiedniego przycisku (zapala się dioda P0.14)

góra → zmiana trybu pokazywania czasu, pokazywa jest czas który upłynął

dół → zmiana trybu pokazywania czasu, pokazywa jest czas który pozostał

wciśnięcie → gdy gra muzyka oznacza „stop”, gdy nic nie gra oznacza „play” (stosowne diody są zapalane identycznie jak dla odpowiednich przycisków)

Dodatkowo w prawym górnym rogu wyświetlacza pojawiają się symbole określające określające stan odtwarzania:

> - aktualnie trwa odtwarzanie

- odtwarzanie jest zatrzymane

I – pauza

X – błąd nie można otworzyć pliku

2. Wykorzystane możliwości mikrokontrolerów

Wykorzystane elementy MCU:

- UART
- GPIO
- TIMER
- ADC
- DAC
- SPI

2.1. Wykorzystanie portów I/O

Porty I/O zostały wykorzystane do manipulowania diodami czy odczytu stanu przycisków oraz joysticka znajdującymi się na kontrolerach w celu sygnalizacji ewentualnych błędów, lub działań programu zakończonych sukcesem. Poniżej w tabeli zaprezentowane są numery diod, zdarzenia w programie jakie muszą zajść, aby dioda ta się zaświeciła.

| Dioda | Zdarzenie |
|-------|--|
| P1.17 | mruga , gdy aplikacja dojdzie do końca kodu (nie powinno to nigdy nastąpić) |
| P1.19 | mruga w trakcie odtwarzania (symbolizuje również moment kiedy następuje uzupełnienie buforu odczytu) |
| P0.14 | mruga , gdy aplikacja napotka błąd z systemem plików(błąd inicjalizacji, błąd odczytu) |

Tabela 1.

2.1.1. Inicjalizacja

Inicjalizacja GPIO odbywa się poprzez ustawienie stanu wysokiego na pinie SCK, MISO, MOSI, SSEL na szynie SPI, (przykład dla czytnika kart) tak więc dla SCK wartość jedności musi zaistnieć na miejscu 4-tym, dla MISO – 5-tym, MOSI – 6-tym oraz dla SSEL na miejscu 13-tym.

2.1.2. Przesyłanie danych

Przesyłanie danych polega na ustawieniu odpowiedniej wartości na danej linii. Odbywa się to poprzez dwa miejsca: IOSET oraz IOCLR. IOSET powoduje ustawienie stanu wysokiego w odpowiednim miejscu, a IOCLR stanu niskiego w danym miejscu.

2.2. Obsługa szyny danych SPI

Szyna danych SPI (ang. *Serial Peripheral Interface*) jest interfejsem komunikacyjnym pomiędzy mikroprocesorami a układami peryferyjnymi. Używa się jej do komunikacji z takimi urządzeniami jak: przetworniki ADC/DAC, układy RTC, pamięć EEPROM, pamięć flash, karty MMC/SD itp.

W naszym programie komunikacja za pomocą SPI odbywać się będzie pomiędzy kontrolerem, a ekranem LCD, kartą pamięci micro SD.

Przepływ danych w tej szynie danych odbywa się za pomocą 4-ech pinów:

- MOSI (ang. *Master Output Slave Input*) – służy do wysyłania danych do urządzenia peryferyjnego
- MISO (ang. *Master Input Slave Output*) – służy do przyjmowania danych z urządzenia peryferyjnego
- SCK (ang. *Serial Clock*) – służy jako sygnał taktujący
- SS (ang. *Slave Select*) – pin, którego rola polega na wskazywaniu urządzenia peryferyjnego do/z którego mają być wysyłane/pobierane dane

2.2.1. Inicjalizacja

Inicjalizacja szyny danych SPI odbywa się poprzez wykonanie następujących czynności:

- ustawienie GPIO, co zostało opisane wcześniej w punkcie 2.1.
- zresetowanie urządzenia wejściowego – SSEL

- zresetowanie wszystkich pinów funkcyjnych: MOSI, MISO, SCK
- nadanie pinom SSEL, MOSI, MISO, SCK stanu wysokiego
- ustawienie szyny danych SPI jako master poprzez nadanie wartości jedności na 5-tym miejscu w rejestrze S0SPCR
- ustawienie odpowiedniej wartości w rejestrze S0SPCCR, który ustawia wartość SCK dla urządzenia ustawionego jako master (ilość cykli PCLK, które taktują zegar SPI), określa to między innymi prędkość transmisji, przyjmuje wartości od 8 do 256 (tylko parzyste). W naszym przypadku zawsze 8.

2.2.2. Przesyłanie danych

Do przesyłania danych wykorzystujemy rejestr S0SPDR, do którego wpisujemy dane, które chcemy przesłać, a następnie w pętli czekamy aż, w rejestrze S0SPSR pojawi się wartość jedności na 7-mym miejscu od prawej strony – świadczy to o tym, że przekazanie danych dobiegło końca, a w rejestrze S0SPDR znajduje się flaga, wskazuje na to, czy przesyłanie danych powiodło się, czy też nie.

2.3 Przetwornik cyfrowo-analogowy DAC

Przetwornik DAC (ang. Digital to Analog Converter) jest to urządzenie elektroniczne przetwarzające sygnał cyfrowy do analogowego. W naszej aplikacji wykorzystany został do skonwertowania utworu wave znajdującego się na karcie do sygnału analogowego i odtworzeniu tego utworu na wyjściu audio – w naszym przypadku na słuchawkach.

2.3.1. Inicjalizacja

Inicjalizacja odbyła się poprzez przypisanie przetwornikowi odpowiedniego adresu, co odbyło się za pomocą jednego makra:

```
#define DACR (*(volatile unsigned long *) 0xE006C000)
```

2.3. Przetwornik analogowo-cyfrowy ADC

Przetwornik ADC (ang. Analog to Digital Converter) to urządzenie elektroniczne zamieniające sygnał analogowy na cyfrowy, odbywa się to przez pomiar przyłożonego na wejściu napięcia i zamiany jego wartości na określoną wartość cyfrową (np. liczba z pewnego zakresu który jest obsługiwany przez przetwornik). W przypadku naszego projektu jest on wykorzystywany do odczytania wartości z akcelerometru, który to na 3 oddzielnych wyjściach, w zależności od przyspieszenia które mierzy nadawane jest odpowiednie napięcie.

2.3.1. Inicjalizacja

Inicjalizacja odbywa się poprzez włączenie odpowiedniego pinu (urządzenia), po czym wprowadzenia odpowiednich wartości do rejestru CR danego przetwornika:

- zegar przetwornika ustawiamy na 4.5MHz
- ustawiamy dokładność 10-bitową
- aktywujemy przetwornik
- wykonujemy testowy odczyt

2.4 Timer, czyli zegar

W projekcie jest on wykorzystany do opóźniania czynności np. odczytu danych z akcelerometru w trakcie potrząsania, bądź odczytu stanu przycisków (użytkownik nie jest w stanie bowiem nacisnąć go na okres paru mikro sekund). Dodatkowym zadaniem Timera jest sterowaniem pętli odtwarzającej sam plik wave, gdzie kolejne próbki muszą być wysłane do przetwornika DAC w odpowiednich odstępach czasu (w naszym przypadku powinno to być 8000 próbek na sekundę).

2.4.1. Inicjalizacja

Jest ona wykonywana przy każdym użyciu zegara

- Zatrzymujemy zegar
- wpisujemy mu wartość jaką ma odliczać
- wyłączamy przerwania zegara
- ustawiamy zegar na tryb w którym po odliczeniu zegar jest zatrzymywany
- włączamy zegar

Informacja o tym, że odliczanie się zakończyła jest sygnalizowana wartością 0 na najmłodszym bicie rejestru CR danego timera (w naszym przypadku Timer 1, czyli T1TCR)

2.5 UART, czyli konsola

Aby umożliwić użytkownikowi lepszą komunikację z płytką skorzystaliśmy z wbudowanego w płytke układ UART (UART ang. Universal Asynchronous Receiver and Transmitter), układ scalony używany do asynchronicznego przekazywania i odbierania informacji poprzez port szeregowy (w naszym przypadku USB). W celu wysłania danych wystarczy je umieścić w rejestrze UART_THR (należy jednak poczekać aż bufor się opróżni, czyli poprzednie dane zostaną przesłane).

2.5.1. Inicjalizacja

- włączenie odpowiedniego PINSEL (P0.0 - wysyłanie, P0.1 - odbiór)
- ustawiamy prędkość działania układu, zgodnie ze wzorem (1) na stronie 92 załącznika nr 1
- parametry działania (ilość bitów, bity stopu parzystość)

- zerujemy przerwania

Informacja o tym, że odliczanie się zakończyła jest sygnalizowana wartością 0 na najmłodszym bicie rejestru CR danego timera (w naszym przypadku Timer 1, czyli T1TCR)

3. Wykorzystane urządzenia peryferyjne

Wykorzystane elementy zewnętrzne:

- diody LED na płytce LPC2138
- wyjście słuchawkowe
- ekran LCD
- akcelerometr
- czytnik kart mikro SD
- joystick

3.1. Ekran LCD

Ekran LCD jest wykorzystywany w aplikacji przez cały czas. To dzięki niemu użytkownik ma orientację co się w programie dzieje i co może zrobić w danym momencie. Z kontrolerem LPC2138 współpracuje on na następujących liniach:

P0.4 – SPI-SCK

P0.6 – SPI-MOSI

P0.7 – SPI-SSEL

Nie potrzebuje on dodatkowej linii MISO ponieważ komunikacja jest jednokierunkowa – tj. ekran nie wysyła do kontrolera żadnych danych, tylko je przyjmuje. Stąd zbędny jest nam port MISO (ang. *Master Input Slave Output*).

Na szynę SPI zawsze wysyłamy 9 bitów. Pierwszy z nich „mówi” o tym, czy dane przesłane po nim są zwykłymi danymi czy też poleceniem. Jeżeli pierwszy bit ma wartość 0 to jest to polecenie, jeżeli 1 wówczas przesyłane są dane do operacji.

Wszystkie piksele mieszczące się na ekranie traktowane są jako wektor pikseli a nie jako tablica dwuwymiarowa. Tak więc jeżeli chcemy zmienić wartość piksela znajdującego się pod pikselem o indeksie *i*, musimy do indeksu *i* dodać ilość pikseli mieszczących się w linii ekranu.

3.2. Joystick

Podczas komunikacji wykorzystuje bezpośrednio piny od P0.8 do P0.12. W celu sprawdzenia stanu joysticku musimy sprawdzić odpowiednie bity rejestru IOPIN. Wówczas na wybranym pinie pojawi się stan niski.

| Pin | Numer bitu rejestru | Efekt |
|-------|---------------------|----------------------------------|
| P0.8 | 9 | Joystick jest wciśnięty |
| P0.9 | 10 | Joystick jest skierowany w lewo |
| P0.10 | 11 | Joystick jest skierowany do góry |
| P0.11 | 12 | Joystick jest skierowany w prawo |

| | | |
|-------|----|----------------------------------|
| P0.12 | 13 | Joystick jest skierowany do dołu |
|-------|----|----------------------------------|

3.3. Akcelerometr

Akcelerometr wykorzystywany jest do zmiany utworów w sposób losowy, po potrząśnięciu urządzenia (osie X i Y), następuje zmiana utworu na losowy.

Urządzenie nie jest bezpośrednio podpięte pod piny IO, został natomiast podpięty pod odpowiednie linie przetwornika analogowo-cyfrowego, aby odczytać stan przyspieszenia na konkretnej osi wystarczy aktywować wejście przetwornika (Pinsel) po czym odczytać przetwornikiem z danego wejścia wartość napięcia nadawaną przez akcelerometr.

P0.21 (AIN1.6) – oś Y

P0.22 (AIN1.7) – oś X

3.4. Karta pamięci micro SD

Czytnik kart został podłączony za pomocą SPI na liniach (SCK P0.4, MISO P0.5, MOSI P0.6, SSEL P0.13). Komunikacja odbywa się za pomocą specjalnego protokołu, lecz wszystkie polecenia i dane, w obu kierunkach przesyłane są za pomocą 8 bitów.

Aby odczytać dane z karty należy wysłać (przez SPI, w rejestrze S0SPDR umieszczamy dane, gdy zostaną przesłane, zmienia się stan rejestru S0SPSR) najpierw nr polecenia odczytu (17) oraz numer sektora, który ma być odczytany. Przyjęcie polecenia oraz gotowość do nadawania danych jest potwierdzane 2 bajtami o określonej wartości. Po czym następuje dopiero przesłanie bajt po bajcie zawartości całego sektora (czyli 512B) oraz 2 bajtów sumy kontrolnej (dokładny opis w komentarzach w kodzie, oraz w dołączonej dokumentacji do czytnika SD)

4. Algorytm programu

1. Program wyświetla strukturę plików (terminal), przygotowywana jest lista plików i oczekuje na wybranie utworu wave, który ma zostać odtworzony
2. Zostaje zmieniona zawartość ekranu – program pokazuje nazwę pliku wave, czas trwania, czas jaki upłynął. Do tego włączona zostaje obsługa klawiszy umożliwiających manipulowanie utworem - play, pause, stop
3. Rozpoczyna się pętla, w której program buforuje utwór muzyczny aż do jego ukończenia, bądź naciśnięcia odpowiedniego przycisku, który pozwala na opuszczenie tej pętli (stop, next, prev)
4. Po zakończeniu odtwarzania lub interwencji użytkownika, następuje powrót do punktu 2.

5. Analiza FMEA/FMECA

Aplikacja nie wystartuje jeśli nie uda się zainicjować czytnika SD lub ustalić systemu plików na karcie, w takim wypadku zostanie wyświetlony komunikat na ekranie oraz na terminalu

pojawi się odpowiednia informacja, dodatkowo zacznie mrugać dioda P0.14, płyta jest blokowana, nie można wykonać już innych czynności poza restartem. Identyczna sytuacja jest w przypadku nie odnalezienia pliku na karcie, dioda P0.14 zaczyna mrugać, na terminalu i ekranie pojawia się komunikat, płyta jest blokowana.

Nie jesteśmy w stanie wykryć braku ekranu. Awaria akcelerometru, jeśli nastąpi, jest obsługiwana w ten sposób, że będziemy dostawać zawsze taką samą wartość, zatem wtedy utracimy po prostu zdolność do losowej zmiany utworów.

6. Wykorzystany sprzęt i oprogramowanie

Wykorzystany przez nas sprzęt to mikrokontrolery:

- mikrokontroler LPC2138
- płyta rozszerzająca Experimental Expansion Board

Wykorzystane oprogramowanie:

- biblioteka „Petit FAT File System Module” – wykorzystana do obsługi struktury katalogów na karcie micro SD
- GNU ARM – jako zestaw do kompilacji

7. Dołączone noty katalogowe

- Załącznik 1 - user.manual.lpc2131.lpc2132.lpc2134.lpc2136.lpc2138.pdf – dokumentacja wykorzystywanego mikrokontrolera
- Załącznik 2 - LPC2138 Education Board Users Guide-Rev A.pdf – dokumentacja wykorzystywanej płyty z mikrokontrolerem
- Załącznik 3 - Experiment Expansion Board Users Guide-Rev A.pdf – dokumentacja płyty Expansion Board
- Załącznik 4 - lds176 spec.pdf – dokumentacja ekranu lcd znajdującego się na płycie Expansion Board
- Załącznik 5 - MMA7260QT.pdf – dokumentacja akcelerometru znajdującego się na płycie Expansion Board.
- Załącznik 6 - ProdManualSDCardv1.9.pdf – dokumentacja do czytników kart SD, opisujący rozkazy karty, formaty odpowiedzi, sposób konfiguracji

8. Podział pracy

Krzysztof Klimczak:

- kierownik zespołu
- obsługa karty micro SD
- obsługa plików wave (odtwarzanie)

Michał Kawski:

- ekran

- przyciski

Dominik Żychliski:

- DAC
- ADC