

DETEKTOR KODU MORSE'A

Z WYKORZYSTANIEM MIKROPROCESORA

KL05Z

Autorzy:

Adam Zielina
Jakub Guza

Prowadzący:

Mariusz Sokołowski

Akademia Górniczo-Hutnicza

Kraków (C) 2022

1. WSTĘP

Ten dokument dotyczy opracowania technicznego raportu programu, który służy jako detektor kodu Morse'a.

Zadanie nasze sprowadziło się do opracowania algorytmów, które będą odpowiednio zliczały czas świecenia, czas nieświecenia, wypisywały oraz wpisywały odpowiednie wartości na UART.

Nasz program posiada kodowanie liter, przejście do nowej litery, przejście do nowego słowa oraz znak końca nadawania.

Peryferia zewnętrzne jakie wykorzystaliśmy to:

- klawiatura klawiszowa
- głośnik WSR-04489
- analogowy czujnik światła ALS-PT19
- alfanumeryczny wyświetlacz LCD1602

Podstawowe założenia projektu:

1. Przygotowanie płytki do pracy z modułami zewnętrznymi.
2. Opracowanie sposobu kodowania liter, oraz algorytmów przetwarzających.
3. Napisanie algorytmu, który koduje kropki i litery.
4. Połączenie się z UART'em oraz umożliwienie komunikacji z PC.
5. Napisanie algorytmu na odtwarzanie liter poprzez głośnik.
6. Nauka i poznanie środowiska Keil uVision oraz programowanie mikrokontrolera ARM.

Dzięki temu projektowi, nauczyliśmy się programowania mikrokontrolera rodziny ARM, KL05Z.

Stworzyliśmy działający projekt, który posiada wiele ciekawych funkcjonalności, które zostaną opisane w kolejnych punktach.

2. FUNKCJONALNOŚĆ

Głównym założeniem projektu, jest nadawanie kodu morse'a za pomocą sygnałów świetlnych. Lecz urozmaiciliśmy projekt o nowe funkcjonalności. Zarys wygląda następująco:

Po włączeniu mikrokontrolera, należy najpierw skonfigurować napięcie referencyjne tła, wyświetlany jest komunikat „skonfiguruj, naciśnij S2”.

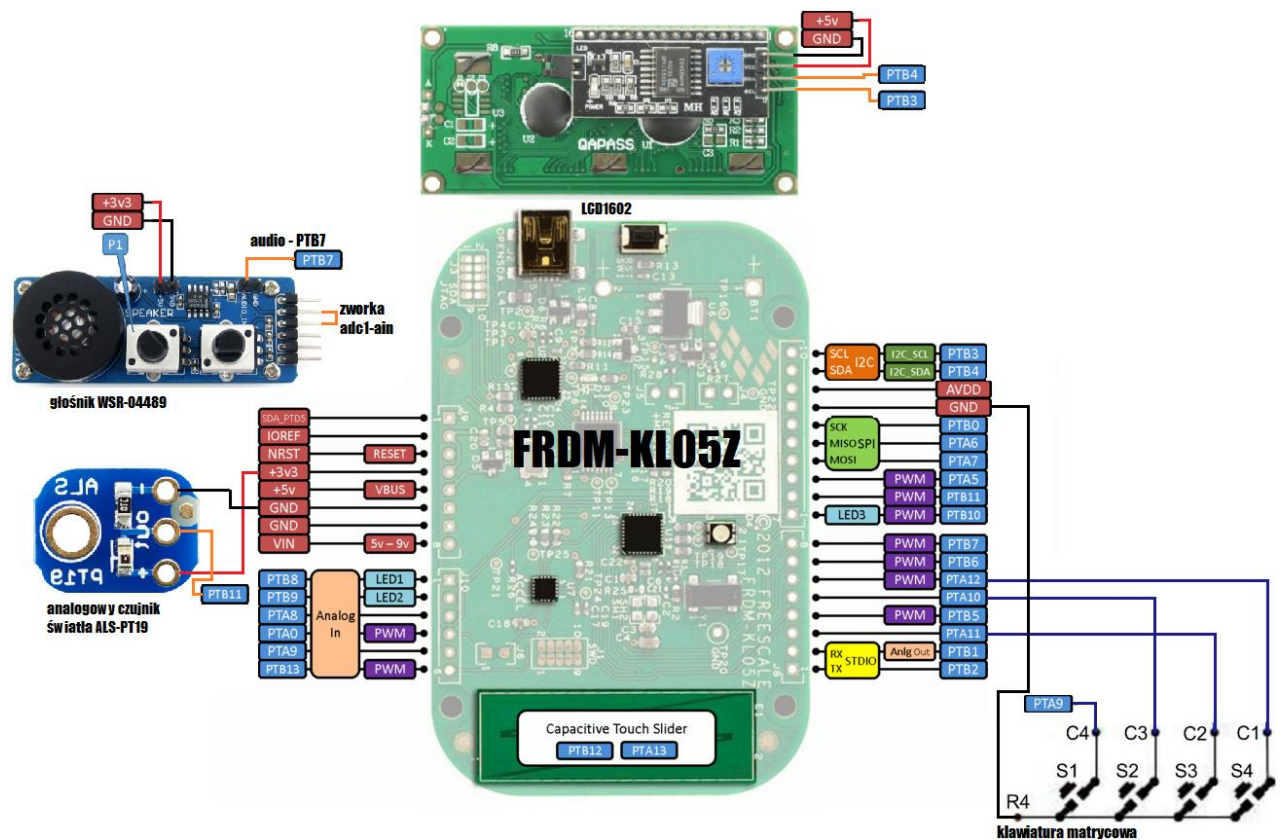
Jednocześnie poprzez komunikację z PC przez PuTTY, wybieramy tryb, w jakim ma pracować mikrokontroler.

Funkcjonalności:

- tryb wysyłania kodu morse'a z komputera poprzez PuTTY, oraz klasyczne nadawanie sygnałami świetlnymi
- tryb odbierania i na komputerze w terminalu, oraz na wyświetlaczu LCD
- wyświetlenie tablicy znaków morse'a w terminalu dla użytkownika
- możliwość odsłuchania nadanej wiadomości poprzez terminal jak i poprzez sygnały świetlne

3. SCHEMAT POŁĄCZEŃ

Poniżej prezentuje się schemat podłączeń peryferiów zewnętrznych naszego projektu.



4. REALIZACJA PROJEKTU

Projekt jest podzielony na pliki nagłówkowe jak i pliki źródłowe.

W podfolderach projektu odpowiednio irc oraz src, są umieszczone pliki .h oraz .c

ADC – kalibracja oraz ustawianie przetwornika analogowo-cyfrowego

i2c – konfiguracja portu szeregowego, potrzebny m.in. do działania wyświetlacza LCD

lcd1602 – konfiguracja wyświetlacza LCD

pit – konfiguracja licznika PIT, wykorzystujemy go do uśredniania wyniku napięcia z przetwornika

klaw – maski oraz inicjalizacja klawiatury matrycowej

uart0 – inicjalizacja portu szeregowego UART0, potrzebny do komunikacji z PC baud rate = 28800

tpm_cm – inicjalizacja licznika TPM0, który jest potrzebny do obsługi głośnika

music.samples.h – plik nagłówkowy, w którym zamieszczamy próbki dźwięków

Główny element naszego projektu, czyli odbiornik kodu Morse’a, gdy nadajnikiem jest latarka, działa na zmianach napięcia, które odbiera fototranzystor.

Najpierw należy skonfigurować odbiornik, który zbiera próbki przez 2s i uśrednia napięcie referencyjne tła.

Kolejnym etapem, jest podział sygnału odpowiednio napięcie niskie(brak świecenia) oraz napięcie wysokie(świecenie).

Licznik Systick, zlicza „ticki” które trwają 400ms, a potem w zależności od zliczonych „ticków” wykrywa ile czasu trwało napięcie niskie lub wysokie.

Jeśli świecenie trwało od 400 do 2399ms, to kodujemy kropkę. (od 1 do 5 ticków)

Jeśli świecenie trwało od 2400 do 4399ms to kodujemy kreskę. (od 6 do 10 ticków)

Jeśli przerwa trwała od 2400 do 3599ms, oznacza to koniec litery. (od 6 do 9 ticków)

jeśli przerwa trwała od 4400 do 5999ms, oznacza to koniec słowa. (od 10 do 15 ticków)

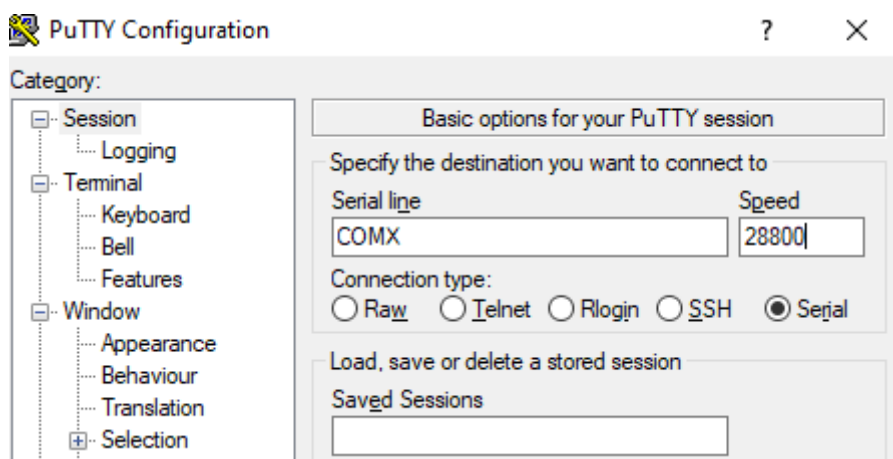
Aby zakończyć nadawanie, należy odczekać więcej niż 6400ms, i nadać krótkotrwały impuls latarką.

Powyższe czynności wyświetlają wynik naszego nadawania na LCD, oraz na terminalu na PC, oraz możliwość odtworzenia nadanej wiadomości przez głośnik.

Kolejne funkcjonalności, jakie oferuje projekt, tyczą się komunikacji PC z mikroprocesorem poprzez port szeregowy UART. Możemy przez terminal nadać kod do przetworzenia przez nasz mikroprocesor i wyświetlenie wyniku na ekranie komputera jak i na wyświetlaczu LCD.

5. PODRĘCZNIK UŻYTKOWNIKA

Aby zacząć użytkowanie projektu, należy podłączyć mikroprocesor kablem USB do komputera. Przechodzimy do PuTTY:



W miejscu „Speed” wpisujemy szybkość transmisji w naszym przypadku 28800.

W miejscu „Serial line” wpisujemy COMX:

gdzie X jest numerem portu szeregowego UART, który przydzielił nasz komputer.

Aby otrzymać ten numer, należy wejść w:

Panel sterowania -> Sprzęt i dźwięk -> Urządzenia i drukarki, w urządzeniach wybrać J-link i kliknąć „właściwości”, w zakładce „Sprzęt” możemy odczytać numer portu szeregowego.

Po włączeniu programu, wyświetli nam się napis „skonfiguruj wciśnij S2”.

Należy na klawiaturze matrycowej nacisnąć przycisk S2, by skonfigurować napięcie referencyjne tła.

Przez dwie sekundy wyświetli się napis „konfigurowanie”, po czym w terminalu następujące opcje:

```
COM3 - PuTTY
skonfiguruj
wcisnij S2

wybierz tryb pracy: odbieranie "rcv" lub wysyłanie "snd"
wyswietl tabele morse'a: "morse"
█
```

Po wpisaniu opcji „rcv”, możemy nadawać sygnały świetlne na czujnik światła,

a terminal staje się „odbiorcą” i wyświetli wynik nadawania kodu Morse’a.

```
rcv
wybrano tryb odbierania
wcisnij s3 aby zaczac odbierac lub wpisz "exit" by wyjsc:

zacznij nadawac

koniec odbierania
wiadomosc: k s
wcisnij s2 aby odtworzyc wiadomosc lub s3 aby wyjsc
```

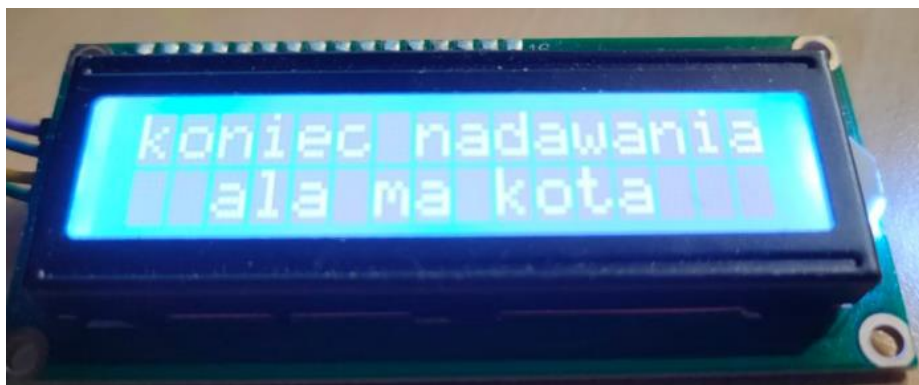
Aby rozpocząć odbieranie, należy wcisnąć s3, i zacząć nadawać sygnały świetlne.

Możemy też powrócić do głównego „menu” wpisując exit.

Po nadaniu znaku końca nadawania, odbieramy przetłumaczoną wiadomość w terminalu oraz na wyświetlaczu LCD. Teraz możemy zdecydować, czy chcemy ją odsłuchać, czy też powrócić do trybu odbierania.

Po wpisaniu opcji „snd”, możemy nadawać kod Morse’a z klawiatury, wpisując sekwencję kropek oraz kresek. Zostanie on odkodowany i wyświetlony na ekranie terminala oraz na wyświetlaczu LCD.

```
wybierz tryb pracy: odbieranie "rcv" lub wysylanie "snd"
wyswietl tabele morse'a: "morse"
snd
wybrano tryb nadawania morsem
wpisz "exit" aby wyjsc lub wpisz kod do nadania:
.- .-.. .- -- .- -. --- -.- ala ma kota
wcisnij s2 aby odtworzyc wiadomosc lub s3 aby wyjsc
```



Aby zakodować słowo, wpisujemy litera po literze oddziałając je spacją.

Aby przejść do kolejnego słowa, naciskamy TAB.

Z tego miejsca również możemy odtworzyć wiadomość, lub wpisać exit, by kodować kolejną sekwencję.

Ostatnią możliwością, jest wyświetlenie alfabetu kodu Morse'a, który służy jako pomoc w nadawaniu.

```
wybierz tryb pracy: odbieranie "rcv" lub wysyłanie "snd"
wyswietl tabele morse'a: "morse"
morse
A .-   B -... C -.-. D -..  E .    F ..-.
G ---. H .... I ..   J .--- K --.  L ....
M --   N -.   O ---  P .--- Q ---. R --.
S ...  T -    U ..-  V ...- W .--  X ---.
Y ---. Z ---.
```

6. DZIENNIK ZMIAN

REV	DATA	ZMIANY	OSOBA
0.0	10.12.2021	Koncepcja projektu, wymyślenie struktury projektu, wymyślenie głównego algorytmu	Jakub Guza Adam Zielina (guzajakub@student.agh.edu.pl) (adamzielina@student.agh.edu.pl)
0.1	12.12.2021	Wstępny kod, praca nad poprawną strukturą projektu	Jakub Guza Adam Zielina (guzajakub@student.agh.edu.pl) (adamzielina@student.agh.edu.pl)
0.2	14.12.2021	Konsultacje z prowadzącym, zmiana planu odnośnie wyszukiwania liter, stworzenie tablicy charów	Jakub Guza Adam Zielina (guzajakub@student.agh.edu.pl) (adamzielina@student.agh.edu.pl)
0.3	20.12.2021	Działający prototyp, poprawnie wypisuje litery na LCD	j.w.
0.4	02.01.2022	Dodanie funkcjonalności UART	j.w.
0.5	08.01.2022	Dodanie funkcjonalności głośnika, odtwarzanie wiadomości	j.w.
0.6	17.01.2022	Praca nad dobrym udokumentowaniem kodu, końcowa dokumentacja, poprawki	j.w.