

Kurs: **Systemy Wbudowane (2021/2022)**
Kierunek studiów: **Informatyka Stosowana (stopień pierwszy)**
Uczelnia: **Akademia Górnictwo-Hutnicza im. Stanisława Staszica w Krakowie**

Temat projektu: Gra w statki	
Autorzy: Leśnicki Bartłomiej Wiktoria Szewczyk	Data oddania: 21/01/2022
	Nadzorowany przez: dr inż. Krzysztof Świentek

1. Opis projektu

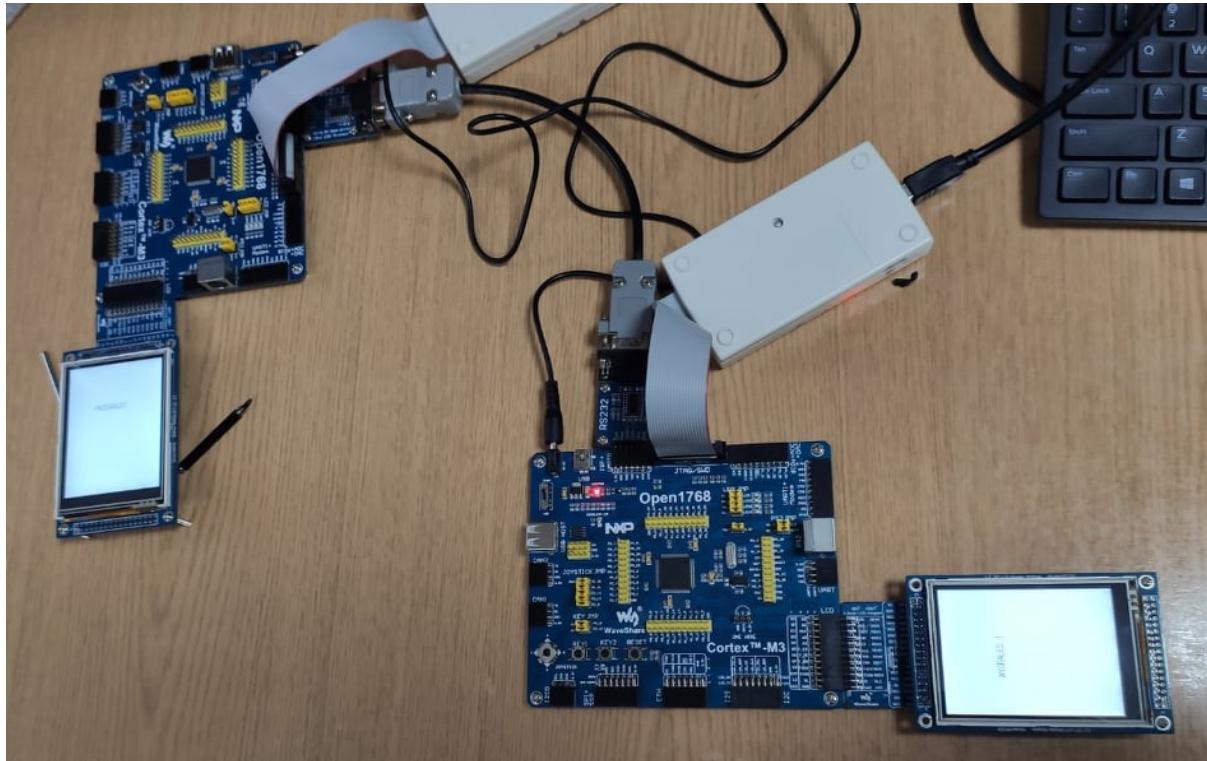
Celem projektu było zaprojektowanie gry w statki na mikrokontrolerach LPC1768. Komunikację obsługuje interfejs UART. Za warstwę graficzną odpowiada wyświetlacz LCD. Sterowanie odbywa się za pomocą touchpada.

2. Przebieg gry

Każdy z graczy ma dwie plansze o wielkości 10 na 10 pól. Na jednej planszy ustawia 10 różnych statków (jeden czteromasztowiec, dwa trójmasztowce, trzy dwumasztowce i cztery jednomasztowce), tak aby nie stykały się ze sobą. Na drugiej planszy zaznaczane są rezultaty strzałów w okręty przeciwnika - trafiony, pułło, pole nie sprawdzone. Grę rozpoczyna osoba, która pierwsza ułoży swoje statki na planszy. Przebieg gry odbywa się turowo. Jeżeli gracz trafi w statek przeciwnika to może strzelić jeszcze raz, jeżeli spudłował zaczyna się tura przeciwnika. Grę wygrywa gracz, który jako pierwszy zatopi wszystkie statki przeciwnika.

3. Wykorzystane urządzenia i peryferia

Do realizacji projektu wykorzystaliśmy dwa mikrokontrolery LPC1768. Dołączliśmy do nich ekrany LCD o nazwie 3.2inch 320x240 Touch LCD składające się z dwóch części które są niezależnymi urządzeniami: wyświetlacza LCD wykonanego w technologii TFT i rezystancyjnego ekranu dotykowego. Z wyświetlaczem komunikujemy się przez 16-bitowy interfejs równoległy i8080, natomiast komunikacja z ekranem dotykowym odbywa się z użyciem protokołu SPI. Mikrokontrolery komunikują się przez interfejs UART (UART0). Podłączenie peryferiów i mikrokontrolerów przedstawia zdjęcie 1.



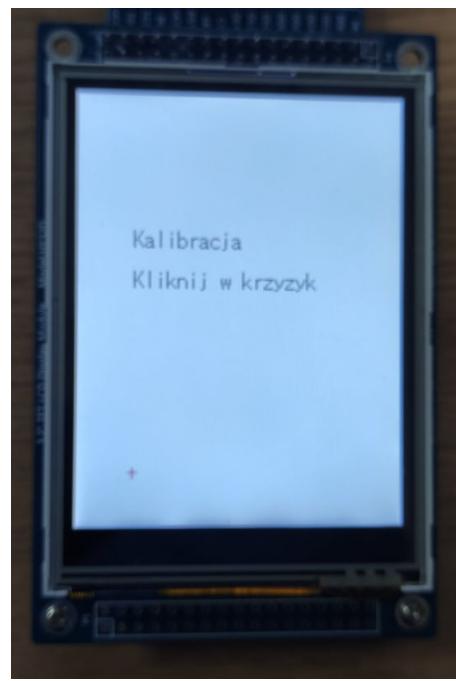
Zdjęcie 1: Podłączenie peryferiów i mikrokontrolerów.

4. Opis działania programu

Aby rozpocząć grę do obu mikrokontrolerów należy załadować ten sam kod oraz nacisnąć przycisk RESET na dwóch płytach ewaluacyjnych. Następnie przechodzimy przez następujące etapy:

I. Kalibracje

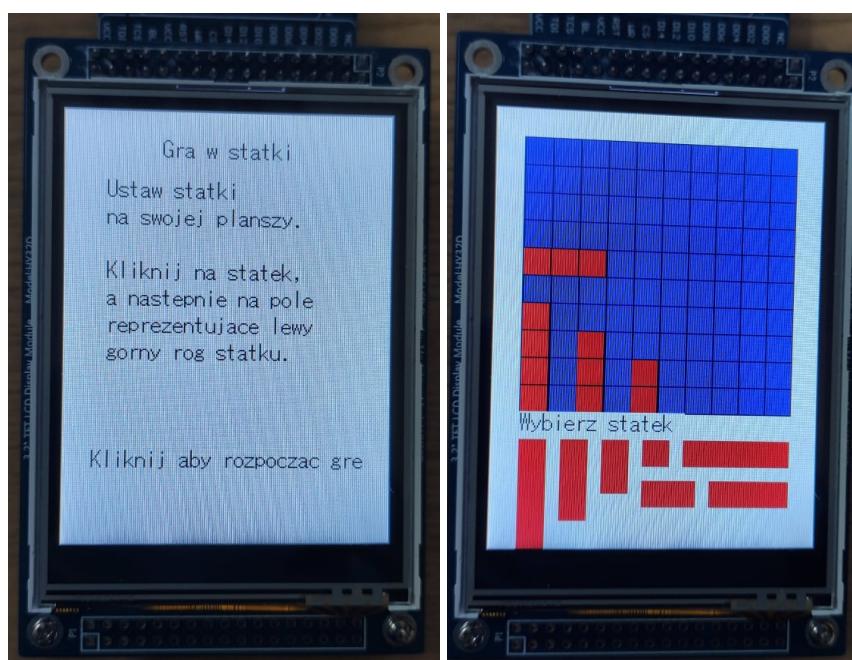
W rogach ekranu zostaną wyświetcone po kolej 3 krzyżyki. Użytkownik powinien kliknąć jak najdokładniej w środek krzyżka. Po tej procedurze następują odpowiednie obliczenia przeprowadzające kalibrację. Dzięki temu po każdym kliknięciu współrzędne ekranu dotykowego zamieniane są na współrzędne wyświetlacza LCD. Proces kalibracji przedstawiony jest na zdjęciu 2.



Zdjęcie 2: Kalibracja.

II. Rozstawianie Statków

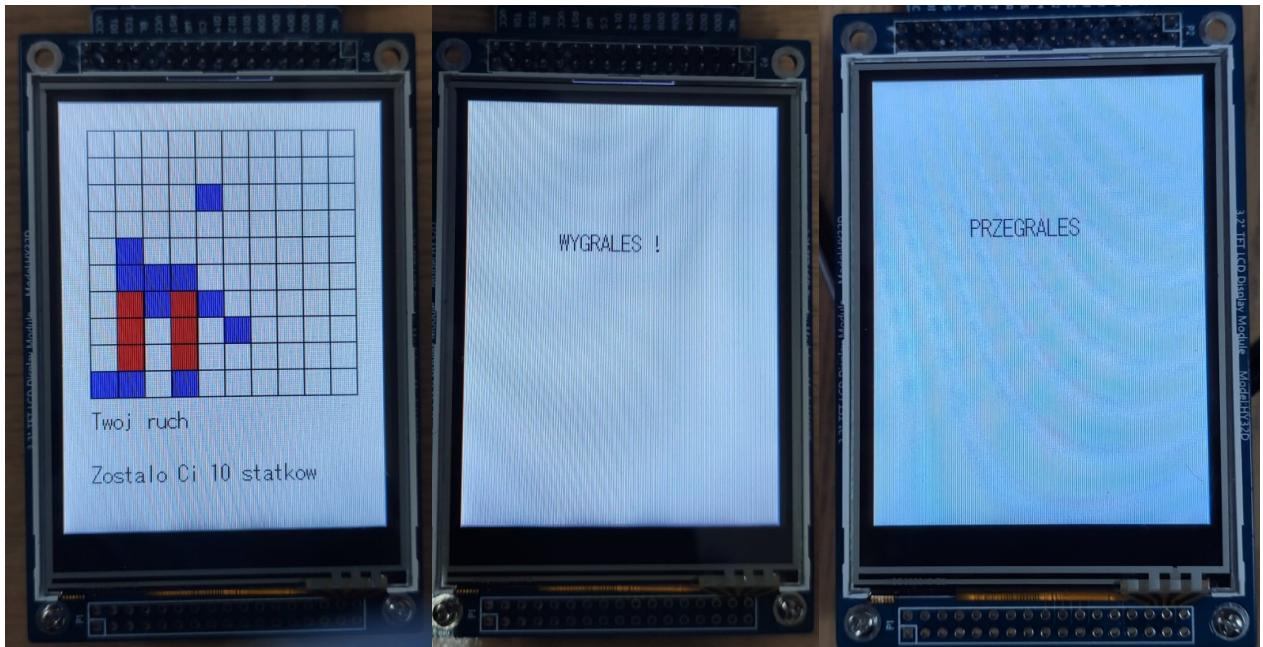
W celu rozstawienia statków wyświetlna zostaje pusta plansza oraz statki do wyboru. Użytkownik wybiera najpierw statek klikając na jego model a następnie wybiera pole na planszy, które będzie reprezentować lewy górny róg statku. Jeśli statek nie spełnia wymogów gry nie zostanie postawiony (liczba statków jest ograniczona, statek nie może leżeć w bezpośrednim sąsiedztwie innego, statek musi się zmieścić na planszy w całości). Jeśli ułożymy wszystkie statki, czekamy na przeciwnika oraz jako pierwi rozłączamy strzelanie. Proces rozpoczęcia gry i rozstawiania statków przedstawia zdjęcie 3.



Zdjęcie 3: Rozstawianie statków.

III. Gra

Gry na ekranie zostanie wyświetlony komunikat "Twój ruch" należy kliknąć na pole, które chcemy ostrzelać. Jeśli trafiliśmy w statek to pole zamaluje się na czerwono i możemy strzelać dalej. W przeciwnym wypadku pole zamaluje się na niebiesko, a na ekranie pojawi się komunikat "Tura przeciwnika" i rozpoczyna się tura przeciwnika. Pola w które nie próbowaliśmy strzelać mają kolor biały. Jeżeli trafiliśmy we wszystkie statki przeciwnika wygrywamy grę. Gra zostaje zakończona, na naszym ekranie wyświetla się napis "WYGRAŁEŚ!" a u przeciwnika napis "PRZEGRAŁEŚ". Proces gry i jej zakończenia przedstawia zdjęcie 4.



Zdjęcie 4: Przebieg i zakończenie gry.

5. Opis funkcji

W naszym projekcie użyliśmy następujących funkcji:

I. Funkcje do obsługi UART. :

- A. void `UART_konfiguracja()` - konfiguracja początkowa UART0,
- B. void `wyslij_info(const int info)` - funkcja do wysyłania danych do drugiego mikrokontrolera,
- C. void `czytaj_info(int *info)` -funkcja do odczytywania danych z drugiego mikrokontrolera.
- D. void `wypisz(char* tab, int size)` - funkcja do wypisywania tekstu na UART używana w procesie debugowania kodu.

II. Funkcje do obsługi wyświetlacza LCD i touchpanela:

A. Do konfiguracji tych peryferiów użyliśmy funkcji bibliotecznych:

- lcdConfiguration(),
- init_ILI9325(),
- touchpanelInit().

B. Do kalibracji wyświetlacza LCD z ekranem dotyковym użyliśmy następujących funkcji:

- void EINT3_IRQHandler(void) - funkcja/handler obsługujący przerwania panelu dotykowego
- void rysuj_X(int x, int y) - funkcja rysuje krzyżek w wybranym miejscu, używana w kalibracji.
- void kalibracja() - funkcja odpowiedzialna za kalibrację
- void touchpad_to_LCD_XY() - funkcja zmieniająca współrzędne punktu dotkniętego na współrzędne na wyświetlaczu LCD.

III. Funkcje odpowiedzialne za warstwę wizualną:

- A. void caly_ekran(int kolor) - zamalowuje cały ekran na wybrany kolor,
- B. void litera(int x0, int y0, char a) - wypisuje literę w danym punkcie na ekranie,
- C. void napis(int x0, int y0, char* text, int size) - wypisuje napis w danym punkcie na ekranie korzysta z funkcji void litera(int x0, int y0, char a),
- D. void rysuj_linie(const int x1, const int y1, const int x2, const int y2, uint16_t kolor) - rysuje linie o zadanym kolorze od punktu (x1,y1) do (x2,y2),
- E. void rysuj_kwadrat(int kolor, int x, int y, int szer) - funkcja rysująca kwadrat o zadanej szerokości boku w danym punkcie i o danym kolorze na wyświetlaczu LCD.
- F. void rysuj_plansze(char tab_statki[12][12]) - rysuje plansze na podstawie danych z tablicy.

IV. Funkcje odpowiedzialne za rozstawianie statków na swojej planszy

- A. void rysuj_rodzaje_statkow() - rysuje wszystkie rodzaje statków w dole ekranu (umożliwia wybór),
- B. void LCD_to_index(int *x, int *y) - zamienia współrzędne ekranu na indeksy w tabeli reprezentującej plansze,
- C. int wybor_statku() - na podstawie kliknięcia w statek w dole ekranu ustala jaki statek został wybrany, zwraca liczbę umożliwiającą identyfikację,
- D. void uzupełnianie_moich_statow() - umożliwia rozstawienie statków na swojej planszy, korzysta z powyższych funkcji.

V. Funkcje odpowiedzialne za obsługę logiczną gry.

- A. int strzelaj() - funkcja odpowiedzialna za obsługę pojedynczego strzału.
- B. bool strzelanie() - funkcja odpowiedzialna za obsługę procesu strzelania. Strzela i powtarza strzał jeżeli trafiliśmy w statek przeciwnika. Zwraca wartość true jeżeli wygraliśmy, zwraca wartość false jeśli gra jeszcze nie została zakończona.
- C. int czekaj_na_strzał() - funkcja odpowiedzialna za odbieranie strzału od przeciwnika. Funkcja zwraca flagę oznaczającą celny lub niecelny strzał.
- D. int czekaj_na_strzelanie() - funkcja odpowiedzialna za logikę odbierania i zliczania strzałów, korzysta z funkcji int czekaj_na_strzał(). Zwraca ilość trafionych pól.

6. Podsumowanie

W ramach projektu zaprojektowaliśmy i zrealizowaliśmy grę w statki na mikrokontrolerach LPC1768. Komunikacja odbywa się poprzez interfejs UART. Udało nam się poprawnie przeprowadzić kalibrację wyświetlacza i ekranu dotykowego. Mikrokontrolery poprawnie komunikują się poprzez interfejs UART z podstawową kontrolą błędów. Algorytmy wprowadzone do programu należycie odzwierciedlają grę w statki. Program realizuje wszystkie etapy gry.