

# Gra w statki

## Dokumentacja projektu

Damian Płociennik  
Grzegorz Litarowicz

22 stycznia 2020

## 1 Opis projektu

Celem wybranego przez nas projektu było stworzenie elektronicznej wersji gry znanej pod nazwami "gra w statki" lub "okręty".

### 1.1 Zasady gry

Gra toczy się na specjalnej planszy. Każdy z graczy posiada po dwie plansze o wielkości 10x10 pól. Na jednym z kwadratów gracz zaznacza swoje statki, których położenie będzie odgadywał przeciwnik. Na drugim zaznaczane są trafione statki przeciwnika i oddane przez siebie strzały. Statki ustawiane są w pionie lub poziomie, w taki sposób, aby nie stykały się one ze sobą ani bokami, ani rogami. Okręty są różnej wielkości i zazwyczaj więcej jest jednostek o mniejszej wielkości, np. gracze mogą posiadać po jednym czteromasztowcu wielkości czterech kratek, dwóch trójmasztowcach wielkości trzech kratek, trzech dwumasztowcach o wielkości dwóch kratek i po czterech jednomasztowcach.

Trafienie okrętu przeciwnika polega na strzale, który jest odgadnięciem położenia jakiegoś statku. Strzały oddawane są naprzemiennie. W przypadku strzału trafionego, gracz kontynuuje strzelanie (czyli swój ruch) aż do momentu chybienia. Zatopienie statku ma miejsce wówczas, gdy gracz odgadnie położenie całego statku.

Wygrywa ten, kto pierwszy zatopi wszystkie statki przeciwnika.

### 1.2 Założenia wstępne

Przystępując do realizacji zadania przyjęte zostały następujące założenia:

1. Do zrealizowania zadania zostaną wykorzystane używane na laboratorium dwa mikrokontrolery LPC 1768 z wyświetlaczami LCD 3,2" TFT (320x240) zintegrowanymi z rezystancyjnymi ekranami dotykowymi,
2. Obie płytki zostaną oprogramowane tą samą wersją kodu,
3. Komunikacja będzie odbywała się z wykorzystaniem interfejsu UART.

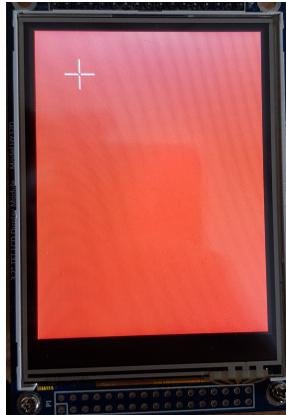
## 2 Użytkowanie aplikacji

W celu rozpoczęcia gry należy posiadać dwa mikrokontrolery LPC1768 z wpięтыm wyświetlaczem LCD ILI9325 (zintegrowanym z rezystancyjnym ekranem dotykowym XPT2046), które są połączone poprzez UART0. Na obu płytach powinien być wgrany stworzony przez nas program.

Aplikacja posiada intuicyjny graficzny interfejs użytkownika, który prezentowany jest na wyświetlaczu LCD a obsługiwany jest poprzez panel dotykowy. Aby uruchomić program i przejść do pierwszego etapu, należy nacisnąć przycisk RESET.

## 2.1 Kalibracja ekranu

W pierwszej fazie należy dokonać kalibracji ekranu dotykowego. W tym celu oczom użytkownika zostanie zaprezentowane czerwone tło z zaznaczanymi kolejno trzema punktami, znajdującymi się przy krawędziach i narożnikach. Zadaniem użytkownika jest jak najdokładniejsze dotknięcie środka wskazywanych punktów.



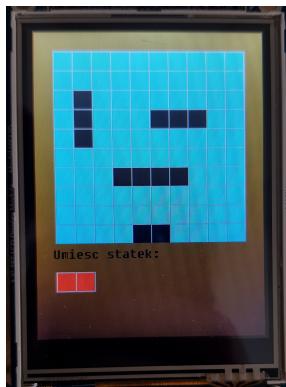
Rysunek 1: Ekran kalibracji.

## 2.2 Planowanie

Drugim etapem do jakiego przystępuje użytkownik po zakończeniu kalibracji jest planowanie. Na wyświetlaczu prezentowana jest początkowo pusta plansza, na której użytkownik rozmieszczać będzie swoje statki. W dolnej części ekranu prezentowany jest statek, który w danym momencie może zostać umieszczony na planszy. W tym celu należy wybrać kolejno pierwsze i ostatnie pole, które ma zająć okręt. Jeśli pola zostaną wybrane nieprawidłowo (niepoprawny rozmiar statku, okręt styka się z innym bokiem lub rogiem) proces rozmieszczania danej jednostki będzie ponawiany do skutku. Użytkownik ma do dyspozycji kolejno:

- jeden czteromasztowiec,
- dwa trójmasztowce,
- trzy dwumasztowce,
- jeden czteromasztowiec.

Po rozmieszczeniu wszystkich statków, należy poczekać aż ten etap zostanie również zakończony przez przeciwnika.



Rysunek 2: Ekran z widoczną fazą planowania.

## 2.3 Gra właściwa

Gracz, który zakończył fazę planowania jako pierwszy rozpoczyna rozgrywkę w trybie aktywnym (ataku), natomiast przeciwnik w trybie biernym ("pod ostrzałem"). W pierwszym z nich prezentowana jest plansza przeciwnika (początkowo pusta) na której kolejno należy zaznaczać pola, na które ma zostać oddany strzał. Jeśli atak został dokonany celnie (na wybranym polu znajdował się statek), to pole zostanie oznaczone kolorem czerwonym i użytkownik może wykonać kolejny ruch. W przeciwnym wypadku pole kolorowane jest na szaro i następuje zamiana ról. W trybie biernym gracz widzi swoją planszę z zaznaczonymi wszystkimi statkami oraz trafieniami i "pudłami" przeciwnika.



(a) Tryb aktywny (atak).

(b) Tryb bierny ("pod ostrzałem").

Rysunek 3: Ekrany w czasie trwania gry właściwej

Gra kończy się w momencie, gdy wszystkie okręty jednego z graczy zostaną zatopione.

## 3 Użytkowanie aplikacji od strony sprzętowo-programistycznej

### 3.1 Obsługa wyświetlacza i panelu dotykowego

Konfiguracji początkowej można dokonać wykorzystując metodę `LCDconfig()`, która inicjuje zarówno wyświetlacz LCD, jak i rezystancyjny ekran dotykowy. Plik `LCD_utils.h` zawiera również inne funkcje przydatne do obsługi wyświetlacza LCD:

- `drawRectangle()` - pozwalającą na narysowanie prostokąta i wypełnienie go kolorem,
- `drawLine()` - pozwalającą na narysowanie prostej linii o zadanym kolorze,
- `background()` - pozwalającą na wypełnienie całego obszaru wyświetlacza jednym kolorem,
- `clearInfoPanel()` - pozwalającą na wypełnienie jednym kolorem części, w której wypisywane są komunikaty,
- `printString()` - pozwalającą na wypisanie na ekranie tekstu.

W pliku `calibrationTp.h` znajdują się procedury odpowiadające za prawidłową kalibrację ekranu. Ekran dotykowy obsługiwany jest poprzez system przerwań, obsługa których dokonywana jest w handlerze `EINT3_IRQHandler()`.

Prawidłowe pobranie współrzędnych z ekranu dotykowego powinno odbywać się w poniżej zaprezentowany sposób:

```
int x;  
int y;  
  
//czeka na dotknietcie (wygenerowanie przerwania)  
tp_wait4touch();
```

```

//przenosi wspolrzedne z panelu dotykowego na wspolrzedne wyswietlacza LCD
recalculateCoords();
//zapisuje aktualne wspolrzedne wyswietlacza do zmiennych x, y
saveCurrentCoords(&x, &y);
//czeka na pusczenie ekranu
tp_wait4release();

```

### 3.2 Obsługa komunikacji poprzez interfejs UART

Konfiguracji początkowej interfejsu UART można dokonać wykorzystując metodę `UARTconfig()`. Plik `uart.h` zawiera również inne metody potrzebne do prawidłowej obsługi komunikacji:

- `sendInfo()` - pozwalającą na przesłanie informacji do drugiego mikrokontrolera,
- `readInfo()` - pozwalającą na odczytanie informacji od drugiego mikrokontrolera,
- `waitForACK()` - oczekującą na przesłanie do mikrokontrolera informacji `CONFIG_READY`.

Zdefiniowane również zostały trzy dyrektywy `#define`:

- `CONFIG_READY` - oznaczającą informację jaka zostanie przesłana/odczytana jako sygnał ACK,
- `GOOD_SHOOT` - oznaczającą informację jaka zostanie przesłana/odczytana przy trafionym strzałe,
- `BAD_SHOOT` - oznaczającą informację jaka zostanie przesłana/odczytana przy chybionym strzałe.

W celu wysłania wiadomości do drugiego mikrokontrolera należy nadać kolejno informację `CONFIG_READY` a po niej właściwą informację w postaci liczby typu `uint8_t`. Odczyt odbywa się kolejno poprzez oczekiwanie na sygnał `CONFIG_READY` (procedura `waitForACK()`), a następnie odczyt informacji.

### 3.3 Obsługa gry

Metody odpowiedzialne za obsługę gry można znaleźć odpowiednio w:

- `field.h` - definicja i obsługa pojedynczego pola,
- `board.h` - definicja i obsługa planszy gry,
- `game.h` - ustawienia początkowe i obsługa gry.

### 3.4 Optymalizacja

Kod komplikowany jest w standardzie C99 z najwyższym trzecim poziomem optymalizacji (-O3). Kompilacja odbywa się bez żadnych ostrzeżeń.

## 4 Bibliografia

- Okręty - Wikipedia, wolna encyklopedia. [online]. [dostęp: 22.01.2020]. Dostęp w Internecie: <https://pl.wikipedia.org/wiki/Okr%C4%99ty>
- STM32 – nieblokująca obsługa panelu dotykowego z układem XPT2046. [online]. [dostęp: 22.01.2020]. Dostęp w Internecie: <https://mikrokontroler.pl/2018/09/17/stm32-nieblokujaca-obsluga-panelu-dotykowego-ukladem-xpt2046/>