

Gra w statki

Adrian Furman

Bartosz Piętka

1 Opis projektu

Celem projektu było stworzenie elektronicznej wersji gry strategiczno-planszowej dla dwóch osób znanej pod nazwami "gra w statki" lub "okręty".

1.1 Założenia techniczne

Projekt zrealizowano przy użyciu dwóch mikrokontrolerów LPC 1768 z wyświetlaczami LCD 3,2" TFT (320x240) z zintegrowanymi z rezystancyjnymi ekranami dotykowymi.

W projekcie zostały przyjęte następujące założenia:

- a) Komunikacja pomiędzy mikrokontrolerami odbywa się poprzez interfejs UART
- b) Obie płytki zostaną oprogramowane tą samą wersją kodu

1.2 Zasady gry

Gra toczy się na stworzonej przez nas planszy. Każdy z graczy posiada dwie plansze o wielkości 10x10 pól oznaczonych poprzez współrzędne literami od A do J i cyframi 0 do 9:

- a) Plansza użytkownika na której przechowujemy własne statki i strzały przeciwnika
- b) Plansza przeciwnika na której przechowujemy oddane przez nas 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, w projekcie przyjęto następującą liczbę statków:

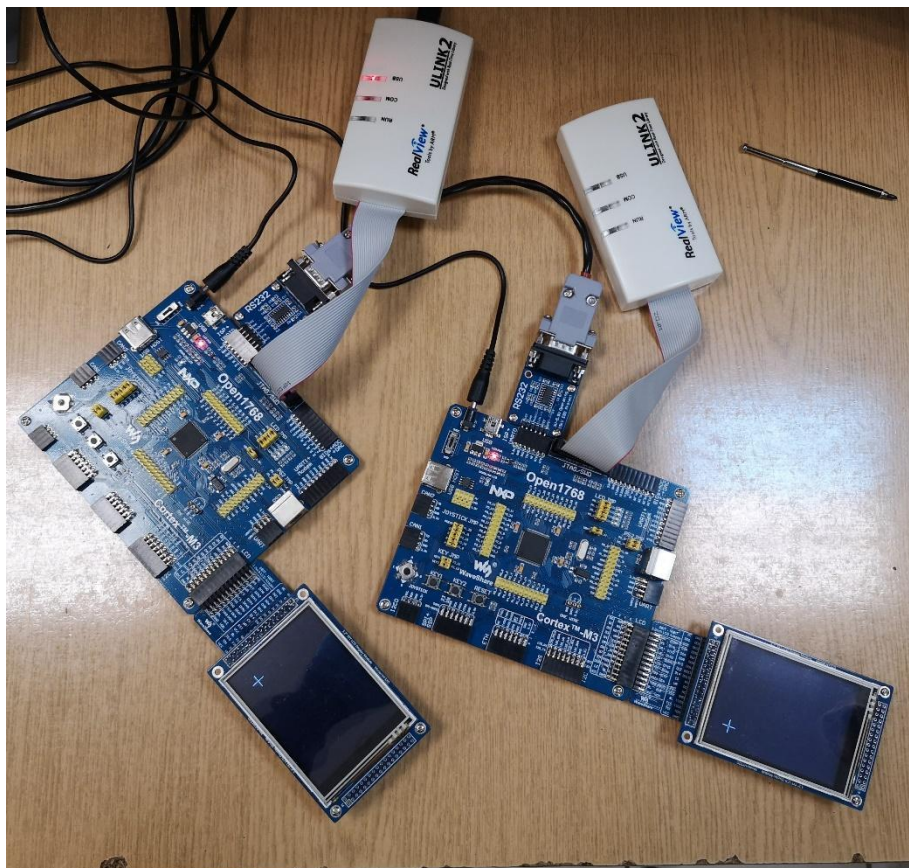
- jeden czteromasztowiec wielkości czterech pól,
- dwa trójmasztowce wielkości trzech pól
- trzy dwumasztowce o wielkości dwóch pól
- cztery jednomasztowce.

Trafienie okrętu przeciwnika polega na strzale, który jest odgadnięciem położenia statku przeciwnika. Strzały oddawane są naprzemiennie. W przypadku strzału trafionego, gracz kontynuuje strzelanie (czyli swój ruch) aż do momentu chybienia.

Wygrywa ten, kto pierwszy zatopi wszystkie statki przeciwnika.

1.3 Przed rozpoczęciem

Przed rozpoczęciem gry należy zaopatrzyć się w dwa mikrokontrolery LPC1768 z wpiętym wyświetlaczem LCD ILI9325 (zintegrowanym z rezystancyjnym ekranem dotykowym XPT2046), które są połączone poprzez UART0. Następnie na obie płytki należy załadować skompilowany w standardzie C99 program. Aby uruchomić program i przejść do pierwszego etapu, należy nacisnąć przycisk RESET. Realizacja połączenia wszystkich modułów przedstawiona jest na zdjęciu poniżej:



Rysunek 1: Przykładowe połączenie modułów

2 Gra

2.1 Kalibracja

Po wciśnięciu przycisku RESET pojawia się ekran kalibracji ekranu dotykowego:



Rysunek 2: Ekran kalibracji

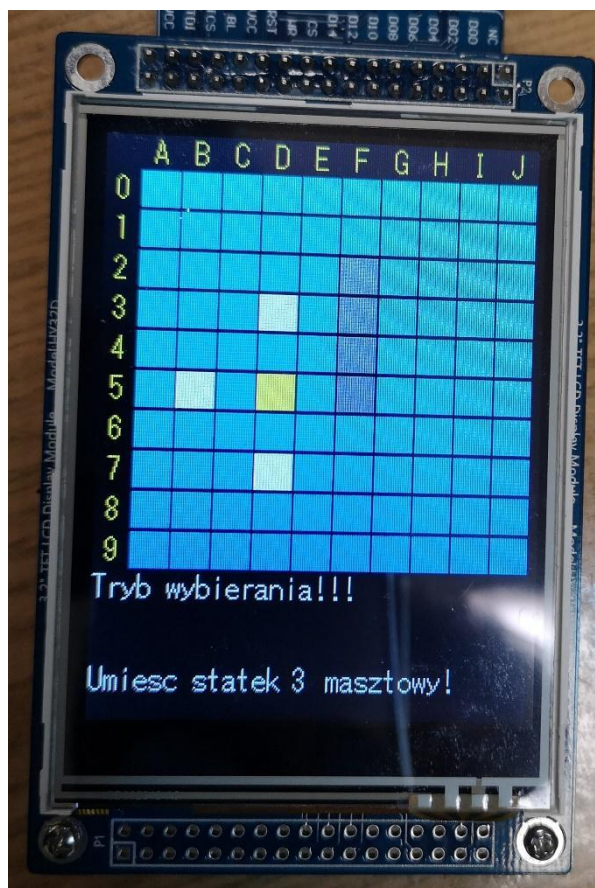
Kalibracja polega na jak najdokładniejszym dotknięciu środka wyświetlanych krzyżyków. Aby osiągnąć satysfakcjonującą dokładność należy użyć rysika, a dotknięcia wykonywać możliwie najbardziej prostopadle.

2.2 Planowanie

Po ukończonej kalibracji pojawia się pusta plansza do gry:

- kolorem szarym oznaczane są statki już wybrane
- kolorem żółtym podświetlone jest aktualnie wybrany początek statku
- kolorem białym podświetlają się wszystkie możliwe zakończenia statku

Wybranie wyświetlonego zakończenia statku nie jest równoważne z jego umiejscowieniem, umiejscowienie jest możliwe tylko jeśli statki spełniają określone wcześniej zasady. Jeśli statku nie da się umieścić pola związane z aktualnym wyborem (żółte i białe) są czyszczone. Prawidłowo umieszczony statek pojawia się w kolorze szarym.

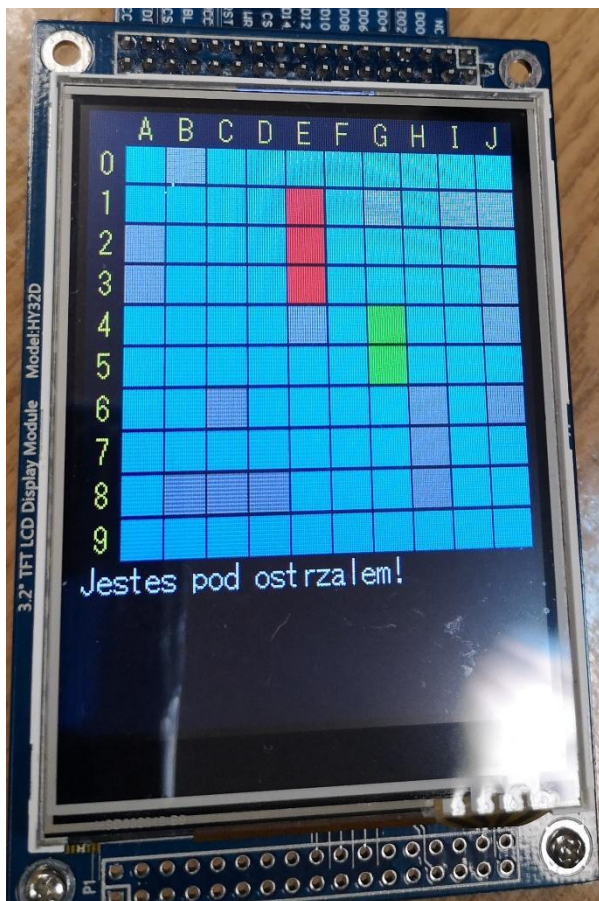


Rysunek 3: Tryb wybierania statków

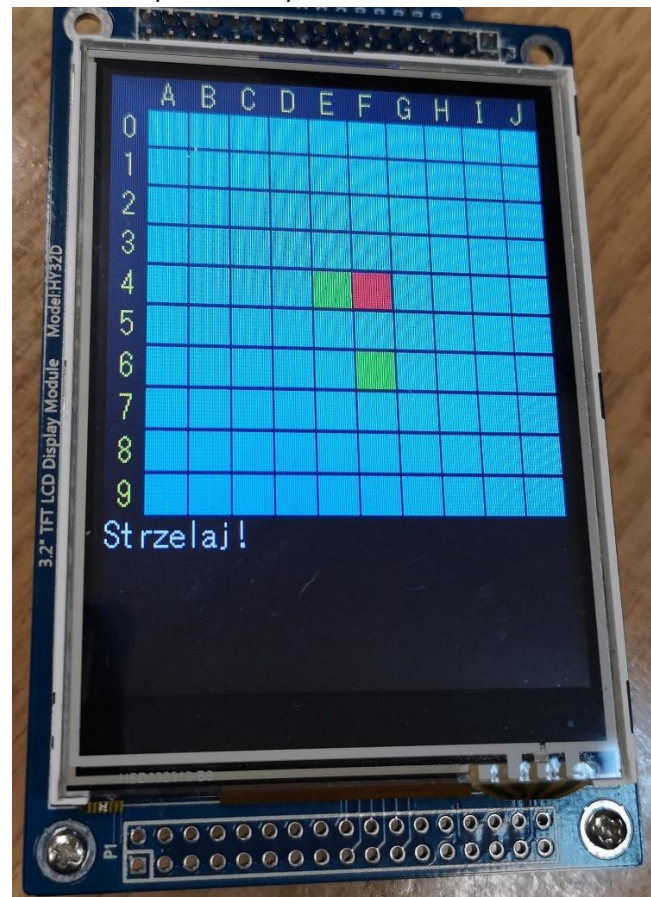
Zadaniem użytkownika jest wybranie kolejno początku oraz końca statku o długości wyświetlanej u dołu ekranu. Po zakończeniu fazy planowania należy potwierdzić swoją gotowość poprzez kliknięcie w ekran.

2.3 Rozgrywka

Gracz, który zakończył fazę planowania jako pierwszy rozpoczyna rozgrywkę w trybie ataku, natomiast przeciwnik w trybie ostrzału. W pierwszym z nich prezentowana jest plansza przeciwnika (początkowo pusta) na której kolejno należy zaznaczać pola, na które ma zostać oddany strzał. Strzał celny oznaczany jest kolorem czerwonym i użytkownik może wykonać kolejny ruch. W przeciwnym wypadku pole kolorowane jest na zielono i następuje zamiana ról.



Rysunek 4: Tryb ostrzału



Rysunek 5: Tryb ataku

Gra kończy się gdy jeden z graczy zatopi wszystkie okręty przeciwnika. Wyświetlany jest wtedy przycisk NOWA GRA. Jego naciśnięcie powoduje przejście do fazy planowania z pominięciem kalibracji.

3 Dokumentacja techniczna

Konfiguracja startowa dokonywana jest poprzez wywołanie metody *void initPeripherals(void)*, która inicjalizuje ekran dotykowy, wyświetlacz LCD, UART oraz włącza przerwania.

3.1 UART

Plik Uart.h zawiera metody potrzebne do prawidłowej obsługi komunikacji:

- *sendInfo()* - pozwalająca na przesyłanie informacji do drugiego mikrokontrolera,
- *readInfo()* - pozwalająca na odczytanie informacji od drugiego mikrokontrolera,
- *waitForACK()* - oczekująca na przesłanie do mikrokontrolera informacji CONFIG READY.

Zdefiniowane również zostały trzy dyrektywy *#define*:

- *READY* - oznaczająca informacje jaka zostanie przesłana/odczytana jako sygnał ACK,
- *HITTED* - oznaczająca informacje jaka zostanie przesłana/odczytana przy trafionym strzale,
- *MISSED* - oznaczająca informacje jaka zostanie przesłana/odczytana przy chybnym strzale.

W celu wysłania wiadomości do drugiego mikrokontrolera należy nadać kolejno informacje *READY*, a po niej właściwa informacja w postaci liczby typu *uint8_t*. Odczyt odbywa się kolejno poprzez oczekiwanie na sygnał *READY* (procedura *waitForACK()*), a następnie odczyt informacji.

3.2 Wyświetlacz i panel dotykowy

Pliki:

- *calibrationTp.h*
- *Drawings.h*
- *Printings.h*

Interakcja pomiędzy ekranem dotykowym a resztą programu oparta jest na systemie przerwań, których obsługa zawarta jest w funkcji *EINT3_IRQHandler()*. Współrzędne dotyku panelu zapisywane są jako średnia z 15 odczytów w celu zredukowania rozrzutu pomiędzy kolejnymi pomiarami przy pojedynczym dotknięciu.

Proces kalibracji, który przeprowadzany jest na początku każdego uruchomienia programu ma na celu wyznaczenie macierzy, która przekształca współrzędne z panelu dotykowego na współrzędne znajdującego się pod tym panelem ekranu LCD. Wypełnienie macierzy przekształcenia oparte jest na klasycznym algorytmie 3 punktowym zaczerpniętym z 4 strony

dokumentu [1]. Każdy późniejszy odczyt współrzędnych zawiera przemnożenie wejściowego wektora współrzędnych odczytanych z panelu dotykowego przez wspomnianą macierz przekształcenia w celu otrzymania współrzędnych ekranu LCD odpowiadającym dotkniętemu miejscu.

Z perspektywy reszty programu oczekujemy dotknięć ekranu w konkretnych momentach (fazach gry/turach) co zapewnia funkcja *getTouch()*, która czeka na dotyk, przekształca współrzędne wykorzystując macierz uzyskaną w procesie kalibracji, a następnie czeka na zakończenie dotyku. Czekanie na start/koniec dotyku uwarunkowane jest globalną flagą *touched* która zamieniana jest w handlerze przerwań wywoływanym z momencie rozpoczęcia oraz zakończenia dotyku.

W plikach Drawings.h oraz Printings.h znajdują się funkcje pomocnicze, odpowiedzialne za rysowanie kształtów, z których składa się interfejs (min. algorytm rysowania linii [2]) oraz wypisywanie napisów/liter pozwalające na przedstawianie rozmaitych komunikatów czy etykiet planszy:

3.3 Gra

Pliki:

- Field.h – definicja pojedynczego pola planszy
- Board.h – definicja planszy wraz z funkcjami obsługującymi plansze
- Game.h – definicja gry i funkcje związane z rozgrywką
- Ship_utils.h – funkcje związane z wybieraniem statków
- main.c - inicjalizacja peryferiów + główna pętla gry

Rozgrywka rozpoczyna się od fazy inicjalizacji planszy i wyboru pozycji swoich okrętów, następnie w momencie obustronnego potwierdzenia gotowości (wymiana komunikatów READY) rozpoczyna się rozgrywka, która przez specyfikę gry oparta jest o tury. W naszej turze wyświetlana jest plansza oddanych strzałów oraz rozpoczyna się oczekiwanie na strzał. Po oddaniu strzału na płytkę przeciwnika wysyłany jest komunikat READY, oraz kolejne dwie ramki ze współrzędnymi oddanego strzału (w tym miejscu współrzędne rozumiane są jako indeksy pól planszy 10x10). Następnie, po otrzymaniu komunikatu READY oraz kolejnych współrzędnych, płytkę przeciwnika sprawdza czy pod w danym miejscu znajduje się okręt i wysyła adekwatną informację (READY + HITTED/MISSED). Po oddaniu strzału i wymianie informacji następuje zmiana tury.

Gra toczy się do momentu zatopienia wszystkich statków jednego z graczy co sprowadza się do chwili uzyskania wartości zero w zmiennej przechowującej sumę nieuszkodzonych

fragmentów floty naszej lub przeciwnika ($4*1 + 3*2 + 3*2 + 1*4$), która dekrementowana jest po każdym celnym strzale.

Inicjalizacja gry (która składa się min. z rozmieszczenia okrętów) oraz sama pętla gry znajdują się w pętli *do while* aby umożliwić ponowne rozpoczęcie rozgrywki po zakończonej partii bez konieczności ponownej kalibracji panelu dotykowego z ekranem LCD oraz inicjalizacji reszty peryferiów.

3.4 Źródła

[1] - Algorytm kalibracji - [AN-1021 APPLICATION NOTE](#),

[2] - Algorytm Bresenhama (rysowanie linii) - [Algorytm Bresenhama – Wikipedia](#),

[3] - Dokumentacja mikrokontrolera LPC1768 - [UM10360 LPC176x/5x User manual](#)