Project dokumentation
ARMFITkit3 či jiný HW: Hra HAD
IMP Mikroprocesorové a vestavěné systémy

November 11, 2022
Zdeněk Lapeš (xlapes02)
lapes.zdenek@gmail.com

Contents

1 Introduction

The task was to create a Snake Game for the ARMFITkit3 board based on microcontroller Kinetis K60 (with ARM CortexM4 core) and 2x matrix displays (type: KWM30881AGB, decoder: 74HCT154).

The result is a Snake game that can be played on the board. Snake is displayed on the matrix displays and the player can control the snake using the *Fitkit3* builtin buttons. Implemented game is a single player game, written in C language using KDS IDE[1].

2 Preparation

2.1 Hardware

The provided HW for this project is *Fitkit3* board and one board with 2x matrix displays. The board with two matrix displays 8x8 is connected to the *Fitkit3* board using the connectors P1 (placed on the Fitkit3 board) and connector P3 (placed on the matrix display's board).
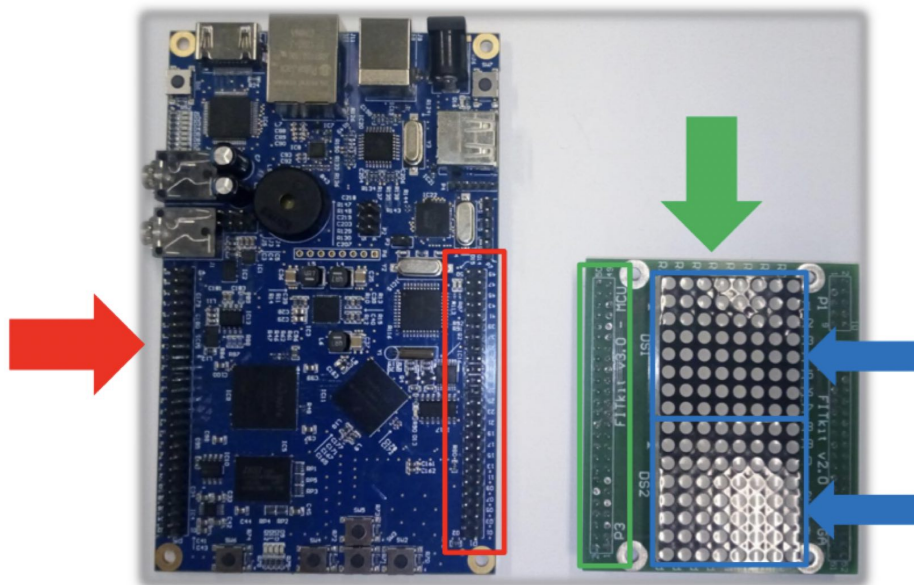

Figure 1: Fitkit3 board & matrix display board

3 Implementation

Implementation of the game is placed in only one file: `Sources/main.c`. The other files were created by the KDS IDE for MCU K60.

3.1 Implementation of the game

Implemantation is divided into 2 parts:

- Button interrupt handling
- Timer interrupt handling

In the projects are used these component of MCU K60 (*K60P100M100SF2V2*):

- **GPIO** (General Purpose Input/Output) Setting in/out pins for controlling LED's matrix displays.

- **PIT** (Periodic Interrupt Timer) Timer for controlling snake moves. Interrupt is generated every 0.24575 ms.

- **PORTx** (Port Control and Interrupts) **PORTA** and **PORTE**. PORTA is used for controlling LED matrix display a PORTB for controlling button interrupts.

The entry point of the program is the function `main()` located in `Source/main.c` file. At the beginning of the program, the function `SystemConfig()` is called where the clock for the **GPIO**, **PIT** and **PORTx** is enabled and pins for the **PORTx** are set as needed for all interrupts and display control. After initialization the structure `snake_t` is initialized and then is continued to endless loop where the all incoming interrupts from **PIT** timer and buttons generated on **PORTE** are handled.

3.2 How to play

The game is a single player game.

3.2.1 Game Control

The game is controlled by the *Fitkit3* builtin buttons:

- **SW2** Snake speed up.

- **SW3** Snake turn right.

- **SW4** Snake speed down.

- **SW5** Snake turn left.

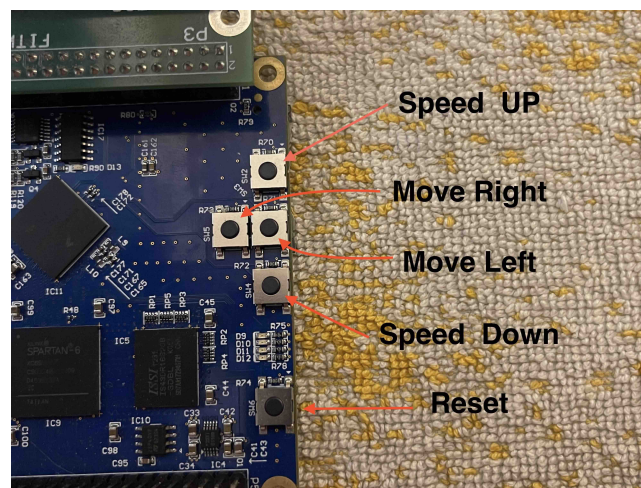- **SW6** Snake reset (default speed and starting position).

Figure 2: Fitkit3 Buttons

The Snake starts moving to the right with normal speed at the beginning of the game. By pressing the coressponding button, you can change the direction of the snake and his speed. If you want to reset the game, press the button **SW6**, which sets the speed to the default value and places snake to the starting position.

4 Conclusion

I managed to implement whole functionality of the game with some additional features like speed up/down and reset3.2.1. Video of the game is available on YouTube.

Literature

[1] Company, N.: Design Studio Integrated Development Environment (IDE). [online].
    URL https://www.nxp.com/design/designs/design-studio-integrated-development-envi
    KDS_IDE