



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

December 10, 2022

Zdeněk Lapeš (xlapes02)  
*lapes.zdenek@gmail.com*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preparation</b>	<b>1</b>
2.1	Hardware . . . . .	1
<b>3</b>	<b>Implementation</b>	<b>1</b>
3.1	Timer interrupt Handler: . . . . .	1
3.2	Button interrupt Handler . . . . .	2
<b>4</b>	<b>Functionality</b>	<b>2</b>
4.1	How to play . . . . .	2
4.1.1	Game Control . . . . .	2
<b>5</b>	<b>Conclusion</b>	<b>3</b>
<b>6</b>	<b>Autoevaluation</b>	<b>3</b>

## 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 written in C language using KDS IDE[1]. Snake is displayed on the matrix displays and the player can control the snake using the *Fitkit3* 5 builtin buttons.

## 2 Preparation

### 2.1 Hardware

The provided HW for this project is *Fitkit3* board and one board with 2x matrix displays. The board with 2 matrix displays 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). One matrix display is (8,8) so the total size of the display is (16,8).

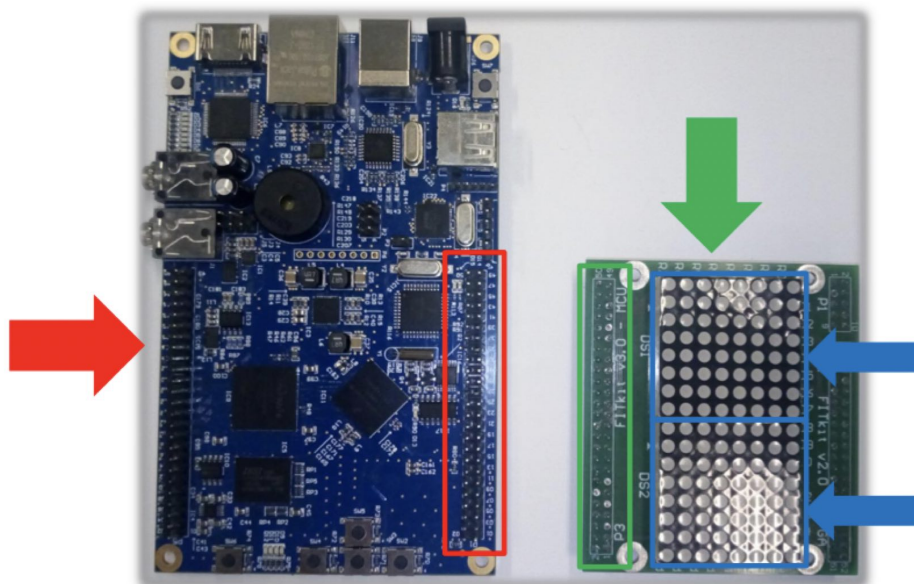


Figure 1: Fitkit3 board & matrix display board

## 3 Implementation

Implementation of the game is inside file: `Sources/main.c`. All other files were created by the KDS IDE for MCU K60.

Implementation is divided into 2 parts:

### 3.1 Timer interrupt Handler:

Snake next position is calculated here and display is updated by calling one of the functions: `move_up()`, `move_down()`, `move_left()`, `move_right()`.

### 3.2 Button interrupt Handler

Snake direction, speed and position is changed here according to the pressed button.

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 of the `snake_t` structure is continued to endless loop while the all incoming interrupts by **PIT** timer and buttons are handled, using the functions `PIT0_IRQHandler()` and `PORTE_IRQHandler()`.

## 4 Functionality

### 4.1 How to play

The game is a single player game.

#### 4.1.1 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).

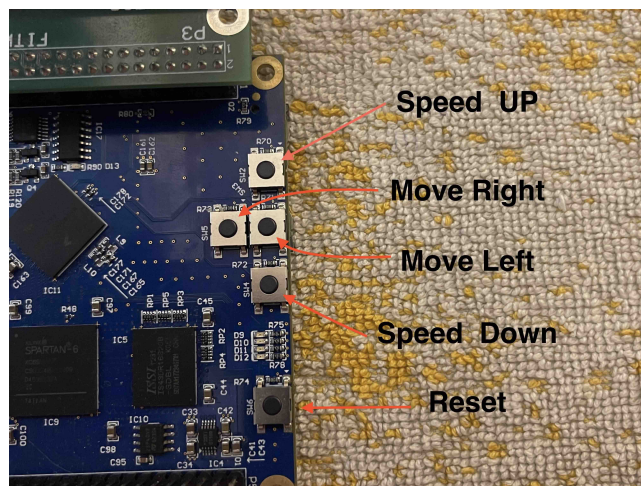


Figure 2: Fitkit3 Buttons

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

## 5 Conclusion

The project was successfully completed and the game is working as expected. I managed to implement whole functionality of the game with some additional features like speed up/down and reset??. In project I used the timer interrupt for controlling the snake moves and button interrupts for controlling the snake direction and speed.

## 6 Autoevaluation

Table 1: Autoevaluation

Task	Points	Description
E	1	I began long before the deadline and afterward I needed to fix some special cases, what I have learned from laboratories.
F	5	The whole functionality requirements were covered.
Q	3	Code should be straightforward to understand almost everybody.
P	1	Illustration of functionality can be watched on youtube <a href="#">click here to watch</a>
D	4	All documentation requirements are covered.
Total	14	

## Literature

[1] Company, N.: Design Studio Integrated Development Environment (IDE). [online].

URL [https://www.nxp.com/design/designs/design-studio-integrated-development-environment-for-kds\\_ide](https://www.nxp.com/design/designs/design-studio-integrated-development-environment-for-kds_ide)