

# Hra HAD: ARM – Fitkit3

Mikroprocesorové a vestavěné systémy – Semestrální  
projekt

# 1 Úvod

Cílem projektu je vytvořit program, který bude fungovat na zařízení Fitkit3 a maticovém displeji, který nám byl poskytnut zadavatelem. Programem má být omezená verze hry „HAD“, známou především pro její slávu v 90. letech, kdy ji firma Nokia, tehdejší největší prodejce telefonů na našem území vložil do svých zařízení. Naše verze bude fungovat pouze pro procházení herního pole s hadem o fixní velikosti. Pohyb bude řešen pomocí tzv. multiplexingu.

## 2 Návrh a implementace

### 2.1 Použité nástroje

Samotnou implementaci, v mém případě provázely problémy, které začaly jak nepodporovaným systémem a následného odmítnutí prostředí MCUXpresso, které jsem považoval za lepší využít, protože podle NXP se jedná o nástupce také doporučeného KDS. Po vyzkoušení nejnovějších instalací obou doporučených prostředí, vyzkoušených na několika operačních systémech jsem nicméně zkusil nainstalovat doporučovanou verzi KDS 3.0.0 u které jsem už nenarazil na jediný problém. Prostedí jsem využíval na operačním systému macOS, pomocí virtuálního obrazu systému Windows 7 Professional.

Mé řešení implementace částečně vychází ze zadaného testovacího programu, ze kterého jsem převzal některé funkce. A který mi pomohl na začátku práce pochopit některé funkcionality.

### 2.2 Použité funkce

#### 2.2.1 Handlers pro časovače:

Funkce jsou inspirovány funkcemi z laboratorního cvičení o přerušeních, kde jsme podobnou funkci vytvářeli.

##### 2.2.1.1 Časovač PIT

Handler „PITO\_IRQHandler“ se stará o to aby nemohly během pohybu hada proběhnout jiné akce v programu. Zajišťuje to, že se had neustále vykresluje.

##### 2.2.1.1 Časovač NVIC na portu E

Handler „PORTE\_IRQHandler“ slouží k vytváření přerušení během stisku tlačítek změny směru.

### 2.2.2 Funkce SystemConfig

Tato funkce je z velké části přejata z daného testovacího programu. Je zde přidáno několik příkazů, které jsem našel buď ve cvičení nebo manuálu, většina z nich slouží k prvotnímu nastavení potřebných flagů a nastavení při spuštění.

### 2.2.3 Funkce row\_select

Tato funkce slouží k usnadnění přístupu k jednotlivým pozicím na displeji. V kombinaci s funkcí „column\_select“, kterou jsem si ve svém projektu také nechal z testovacího programu velmi ulehčuje práci a zvyšuje přehlednost kódu v místech, kde je zajištěn pohyb hada.

### 2.2.4 Funkce flash\_body

Jedná se o další obslužnou funkci, která jednoduše zvyšuje přehlednost kódu. Její použití vyžaduje pouze funkce „row\_select“ a „column\_select“ s nimiž v kombinaci dokáže rozsvítit dané tělo hada jen pomocí současné pozice dané v systému.

### 2.2.4 Funkce moves

Tato funkce se stará o pohyb hada, respektive jeho návaznost na sebe a přemístění mezi hranami.

Dále jsem převzal funkci delay() z původního testovacího programu, která má za úkol to aby displej vypadal jako, že svítí více bodů najednou.

## 3 Obhajoba

- <https://youtu.be/PurBTAVSZD4> - Funkčnost mého řešení.
- <https://youtu.be/qcPEXLMbfbM> - Jednotlivé části kódu

## 4 Zdroje

- <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fhw%2FLab2&cid=14662> – cvičení 2
- <https://www.nxp.com/docs/en/user-guide/KDSUG.pdf> - KDS 3.0.0
- [https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fprojects%2FIMP\\_projekt+-had.pdf&cid=14662](https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fprojects%2FIMP_projekt+-had.pdf&cid=14662) – zadání
-