



# **IMP Projekt 2021/2022**

## **Hra Tamagotchi**

# Obsah

<b>1. Úvod</b>	<b>3</b>
<b>2. Implementácia</b>	<b>3</b>
2.1. Inicializácia	3
2.2. Tlačidlá	4
2.3. Bzučiak	4
2.4. Časovač životného cyklu	4
2.5. Displej	4
2.5.1. Vysvecovanie jednotlivých LED	4
2.5.2. Výber postavy	4
2.5.3. Vykresľovanie postavy	5
2.5.4. Pohyb postavy	5
<b>3. Funkcionalita a návod použitia</b>	<b>5</b>
<b>4. Záver</b>	<b>5</b>
<b>5. Odkaz na demo video</b>	<b>6</b>
<b>6. Zdroje</b>	<b>6</b>

# 1. Úvod

Cieľom projektu bolo vytvoriť zjednodušenú verziu hry Tamagotchi. Táto hra spočíva v staraní sa o elektronickú postavičku, ktorú je potrebné "kŕmiť", aby nezomrela. Hra je implementovaná v jazyku C, pomocou prostredia Kinetis Studio Design 3.0, na prípravku FITkit v.3.0. Je využívaný rozširujúci modul maticové LED displeje.

Pre ovládanie hry boli využité tlačidlá, pre vykreslenie postavy maticové displeje a nakoniec pre zvukové efekty zvukový generátor. Ovládanie životného cyklu postavy je implementované pomocou časovača.

Samotná hra má 3 fázy. V tej prvej je možné zvoliť postavu, s ktorou bude hra prebiehať. Je možné zvoliť z 3 rôznych postáv. Každá má inú vizuálnu reprezentáciu a takisto má vlastný pohyb po displeji.

Následne sa spustí životný cyklus tejto postavy. Postava žije, pokiaľ je v nejakom časovom intervale nakŕmená. Pokiaľ sa tak nestane, postava zomrie a hra skončí. Po konci hry je možné ju opäť spustiť a zvoliť inú postavu.

## 2. Implementácia

Zdrojový kód aplikácie sa nachádza v module **main.c**, ostatné súbory boli vygenerované pri vytváraní projektu. Kód je rozdelený do jednotlivých častí - makrá, globálne premenné, inicializačné funkcie, handler na ovládanie tlačidiel, handler na obsluhu životného cyklu postavy, handler na generovanie zvuku, funkcie na výber, vykresľovanie a pohyb postavy. Jadrom aplikácie je funkcia **main**, v ktorej sa nachádza nekonečný cyklus riadiaci celú hru a jej jednotlivé fázy.

### 2.1. Inicializácia

Funkcie **SystemConfig()**, **TimerConfig()**, **LPTMR0Init()** sa používajú na konfiguráciu potrebných periférií, ktoré sa využívajú pri implementácii. Vo funkcii **SystemConfig()** sa aktivujú moduly a porty ktoré sú potrebné. Následne prebehne konfigurácia jednotlivých pinov. Funkcia **TimerConfig()** sa stará o inicializáciu časovača zodpovedného za životný cyklus postavy. Táto inicializácia prebehne pri každom novom životnom cykle postavy. **LPTMR0Init()** sa stará o inicializáciu časovača zodpovedného za generovanie zvuku. Inicializačné funkcie boli prebraté z poskytnutých demo aplikácií a upravené pre potreby projektu.

### 2.2. Tlačidlá

Reagovanie na stlačenie tlačidiel je implementované v handleri **PORTE\_IRQHandler()**, ktorý kontroluje prerušenie na **PORTE** a porovná ho s makrami jednotlivých tlačidiel. Tento handler sa stará o to, keď si užívateľ vyberá postavu a je

potrebné sledovať, do ktorej strany posúva ponuku postáv. Následne sa stará o spustenie hry, kŕmenie postavy a nakoniec takisto o resetovanie hry.

## 2.3. Bzučiak

Bzučiak je spínaný vo funkcii **beep()** v momente, keď je nastavený flag na jeho spustenie - ten sa nastavuje pri stlačení tlačidla **SW6** vo fáze životného cyklu postavy. V tom momente je nastavený na logickú jednotku. Zvuk má statickú dĺžku a frekvenciu a je spustený pri kŕmení postavičky.

## 2.4. Časovač životného cyklu

Dĺžka životného cyklu je fixne určená v globálnej premennej, je nastavená na približne 20 sekúnd. Cyklus je vždy obnovený po nakŕmení postavy, ale začne až po skončení celého predchádzajúceho cyklu. Teda ak je postava nakŕmená aspoň raz za jeden životný cyklus, tak ho dožije v plnej dĺžke, začína sa nový životný cyklus a následne musí byť opäť v časovom úseku dĺžky života nakŕmená.

O tento cyklus sa stará časovač **PIT**, ktorý vyvolá prerušenie v pravidelných intervaloch. Tieto prerušenia značia koniec životného cyklu. Časovač je obnovený vo funkcii **main** po tom, ako nastane prerušenie tlačidla **SW6**, ktoré značí kŕmenie.

## 2.5. Displej

### 2.5.1. Vysvecovanie jednotlivých LED

Vysvecovanie na displeji implementujú dve funkcie a to **column\_select()** a **print\_row()**. Funkcia na výber stĺpca je prebratá z dema poskytnutého k projektom. Funkcia sa stará o zvolenie konkrétneho stĺpca v ktorom sa bude vykresľovať. Následne na to, je volaná funkcia pre výber riadku, ktorá je volaná pre každý riadok v ktorom je žiadúce vykresľovať v danom stĺpci.

### 2.5.2. Výber postavy

V prvej fáze života sa postavy vykresľujú staticky bez pohybu. Toto sa vykonáva pomocou pamätania si aktuálne vykresľovanej postavy, následného kontrolovania prerušenia tlačidlami pre zmenu postavy. Zmena aktuálnej postavy sa vykonáva vo funkcii **choose\_character()**, ktorá určí dizajn a pohyb aktuálnej postavy. Dizajn sa kopíruje z globálnych premenných - matic - ktoré fixne udávajú východzí dizajn postavy.

### 2.5.3. Vykresľovanie postavy

Vykreslenie postavy sa vykonáva funkciou **draw\_character()**, ktorá vykreslí maticu ktorá sa práve nachádza v premennej **current\_character**.

### 2.5.4. Pohyb postavy

Pohyb postavy je implementovaný v dvoch funkciách. Prvou je **character\_movement()**, ktorá definuje jednotlivé špecifické spôsoby pohybu. Kontroluje aktuálnu pozíciu postavy na displeji a podľa toho určí jej nasledujúcu polohu.

Druhou funkciou je **move\_character\_one\_led()**, ktorá posunie postavu na displeji určeným smerom o jednu pozíciu. Táto funkcia je volaná vo funkcii **character\_movement()**, na vykonanie jednotlivých zmien postavy.

Samotný pohyb postavy je implementovaný využitím funkcie **delay()**, ktorá je volaná medzi vykresľovaním a zmenou polohy postavy.

## 3. Funkcionalita a návod použitia

Pre spustenie hry je potrebné pripojiť mikrokontrolér k počítaču. Po pripojení a otvorení aplikácie Kinetis Design Studio je potrebné skontrolovať debug nastavenia a následne spustiť samotný debug. Po nahratí aplikácie do zariadenia je možné hru začať.

Na začiatku hry si hráč pomocou šípok doprava a doľava vyberá postavu, s ktorou chce hrať. Stlačením tlačidla **SW6** sa potvrdí výber aktuálnej postavy a začne sa hra. Po začiatku hry je potrebné postavu kŕmiť stlačením tlačidla **SW6**. Pokiaľ sa tak nestane raz za nejaký určený čas, postava umrie a hra končí zobrazením prehry na displeji.

Po prehre je možné hru opäť spustiť a to stlačením tlačidla **SW6**.

## 4. Záver

Riešenie projektu spĺňa požiadavky zadania. Pri implementácii boli použité prvky a komponenty tak, ako bolo určené v zadaní. Aplikácia bola testovaná na FITkit v.3.0. a neprejavili sa žiadne chyby.

Hra reaguje na tlačidlá, vykresľuje postavu a jej pohyb a takisto generuje zvukový efekt pri kŕmení postavy. Časovače sú využité pre signalizovanie životného cyklu a generovanie zvuku.

Pre zlepšenie riešenia by bolo možné pridať vizuálnu indikáciu kŕmenia, to by však vyžadovalo upraviť dizajn postáv, pretože inak by bolo nutné zablokovať nejakú časť displeja. Takisto by bolo možné pridať viac zvukových efektov, toto nebolo vylepšené z časových dôvodov.

## 5. Odkaz na demo video

Demo aplikácie (1:40)

<https://drive.google.com/file/d/1Qi0yODnX1lZKsOwS9CzbSdJcp2-liRP5/view?usp=sharing>

Popis zdrojového kódu (2:20)

<https://drive.google.com/file/d/1egT4mmDX71dkOAZAoc3nAxbU7S5rRbiK/view?usp=sharing>

## 6. Zdroje

1. Text k prednáškam kurzu Mikroprocesorové a vestavěné systémy na VUT FIT v Brně. [online] 2021 [cit. 18.12.2021]. Dostupné z:

<https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Flectures&cid=14662>

2. KL05 Sub-Family Reference Manual [online] Rev 3.1. Freescale Semiconductor, Inc © 2012 [cit. 18.12.2021]. Doc. Number: KL05P48M48SF1RM. Dostupné z:

[http://dl.btc.pl/kamami\\_wa/kl05\\_reference\\_manual.pdf](http://dl.btc.pl/kamami_wa/kl05_reference_manual.pdf)

3. Schéma zapojenia FITkit v3.0. [online] [cit. 18.12.2021] Dostupné z:

<https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fexcs%2FFITkit3-scheme.pdf&cid=14662>