

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy - semestrální projekt

M - ARM-FITkit3: Hra HAD

Obsah

1	Úvod	2
2	Návod	2
3	Implementace	2
3.1	Technické detaily implementace	2
3.2	Programová část implementace	2
3.2.1	Funkce <code>main</code>	2
3.2.2	Handler pro časovač PIT - <code>PIT0_IRQHandler</code>	2
3.2.3	Handler pro časovač NVIC na Portu E - <code>PORTE_IRQHandler</code>	2
3.2.4	Funkce <code>SystemConfig</code>	3
3.2.5	Funkce <code>display_snake</code>	3
3.2.6	Funkce <code>move_snake</code>	3
3.2.7	Funkce <code>column_select</code> a <code>row_select</code>	3
3.2.8	Funkce <code>light_up</code>	3
3.2.9	Zpomalovací funkce	3
3.2.10	Funkce <code>rip</code>	3
4	Literatura a zdroje	4

1 Úvod

Zadáním projektu bylo vytvořit program, který bude na malém maticovém displeji, který je řešený pomocí tzv. multiple-xingu realizovat hru "HAD". Zároveň tuto hru bude možné ovládat pomocí tlačítek umístěných na FITkit3.

Tato hra je však lehce omezená oproti běžnému formátu a to tak, že had nebude sbírat ovoce a tudíž nebude ani růst, jde jen a pouze o pohyb hada.

2 Návod

Program je nutné přeložit a spustit v Kinetis Design Studio. Po spuštění se hra ovládá tlačítky nahoru, dolů, doprava a doleva, které mění směr pohybu hada a tlačítkem center, které restartuje hru.

3 Implementace

3.1 Technické detaily implementace

Jak už bylo zmíněno výše, program bude realizován na zařízení FITkit3 a na maticovém displeji. Zároveň je tento projekt napsán ve vývojovém prostředí Kinetis Design Studio.

3.2 Programová část implementace

K úvodnímu pochopení, jak zařízení FITkit3 funguje byly použity doložené materiály, které nám byly poskytnuty zadavateli. Hlavní nápomocí byly schémata přístroje FITkit3 [1], které sloužili k zjištění na jakých pinech můžeme pracovat s jakými částmi zařízení a velikou pomocí při pochopení funkcionality byl demostrační program [2], který nám byl poskytnut.

Dále k implementaci projektu byly využity materiály a kódy z 2. cvičení [3], odkud jsem načerpal informace, jak je možné spravovat signály tlačítek a používat tlačítka.

Dále jsem čerpal z prezentace k projektu - [4].

Celý program je realizován v souboru `main.c`, který je rozdělen na další podfunkce a makra.

3.2.1 Funkce `main`

V hlavní funkci programu dojde nejdříve k zavolání funkce `SystemConfig`, což je funkce, která byla převzatá z demostračního programu [2].

Dále je potřeba inicializovat každou část hada, který je reprezentovaný polem struktury typu `snake_part` a obsahuje 4 integerové hodnoty. Jedna značí pozici x (tedy číslo řady), druhá značí pozici y (tedy číslo sloupce) a třetí značí směr, kterým se had pohybuje (tyto směry jsou potom reprezentovány makry - `UP`, `DOWN`, `LEFT` a `RIGHT`).

Nadále dojde k zavolání funkce `display_snake`.

Následně se dostává program do nekonečného cyklu, přes který iteruje celý svůj běh programu.

3.2.2 Handler pro časovač PIT - `PIT0_IRQHandler`

Tento handler ošetřuje přerušení při pohybu hada tak, aby nemohly proběhnout žádné jiné akce v programu.

Vykonává dva typy příkazů na základě toho, v jakém stavu se program ocitne. V případě, že je had živý, tak se může volně pohybovat a zavolá se tedy funkce `move_snake`, která hadem pohne nebo zjistí, zda had nezemřel. Pokud nezemřel, tak je zavolána funkce `delay_snake` a had je vykreslován po určitý čas.

Pokud je had mrtvý, je zavolána funkce `delay_rip`, která zobrazí nápis na maticovém displeji.

Poté je vyčištěn řádek, aby nebyly žádné diody aktivovány a flag pro přerušení je vymazán.

3.2.3 Handler pro časovač NVIC na Portu E - `PORTE_IRQHandler`

Tato obslužná funkce se stará o bezpečné stisknutí tlačítek. Pokud je stisknuté tlačítko, je zastaven chod programu, handler zpracuje jaké tlačítko bylo stisknuté a provede požadovanou činnost. Pokud je zmáčknuto tlačítko `BUTTON_CENT_MASK`, tak je program resetován a je hra začíná odznova.

Funkce je inspirována kódem z [3].

3.2.4 Funkce `SystemConfig`

Tato funkce je převzatá z demostračního programu [2] a stará se o konfiguraci všech hlavních komponent programu. Nastaví PTA piny (pro výběr sloupce a řady) a nastaví potřebné PTE piny pro ovládání pomocí tlačítek (upraven kód z [3]).

Také se postará o konfiguraci časovače na PORTu B a o konfiguraci časovače PIT.

3.2.5 Funkce `display_snake`

Tato funkce se za pomoci funkcí `column_select` a `row_select` stará o zobrazení hada na maticovém displeji.

3.2.6 Funkce `move_snake`

Se stará o pohyb hada. Funguje na základní logice posouvání hada. Tělo této funkce je složené z hlavního `for` cyklu, který iteruje od posledního článku hada až po první (tedy hlavu). Každý článek (až na hlavu) si převezme souřadnice a směr toho předcházejícího článku. Pro hlavu platí to, že jako jediná používá ještě jednu svoji proměnnou a tou je `direction_prev`, která uchovává informace o předchozím směru pohybu hada a tedy je-li potřeba změnit směr, je druhému článku předána hodnota této proměnné, aby zachoval pohyb hada a hlava se pohne už ve směru, který jí byl zvolen jako poslední.

Může se však stát, že had se plazí nesprávným směrem, který vyústí v jeho smrt. V tu chvíli navrátí tato funkce hodnotu `1 - true`, jinak vrací `0 - false`.

3.2.7 Funkce `column_select` a `row_select`

Tyto zastávají roli výběru sloupce a řádku na maticovém displeji. V případě `column_select` jde o 4-16 dekodér, který na základě vloženého decimálního čísla dekoduje signály A0, A1, A2 a A3, které aktivují příslušný sloupec. Funkce `column_select` byla přejatá z demonstračního programu [2]. `row_select` vybere v příslušném sloupci diodu v řádku a vrátí její binární reprezentaci, pomocí které jsme schopni diodu rozsvítit.

3.2.8 Funkce `light_up`

Použitím funkcí `column_select` a `row_select` rozsvítí diodu.

3.2.9 Zpomalovací funkce

Funkce `delay`

Tato funkce slouží pouze pro zpomalení chodu programu. Využívá se například při vykreslování hada, aby to vypadalo, že jsou všechny diody hada rozsvíceny naráz, přitom se vždy rozsvěcuje pouze jeden sloupec naráz. Napětí v diodách však chvíli zůstane a tedy když jsou sloupce rozsvěčovány v krátkých intervalech, tak to vypadá, že jsou rozsvíceny naráz.

Funkce `delay_snake`

Principiálně stejná funkce jako funkce `delay`, jediný rozdíl je v tom, že při každé iteraci dojde k vykreslení hada.

Funkce `delay_letter`

Je úplně stejná funkce, pouze vypisuje část písmena při každé iteraci.

3.2.10 Funkce `rip`

Tato funkce byla implementována v rámci dobrovolného rozšíření a stará se o výpis velkého nápisu `RIP` na maticový displej v případě, že had zemře. Má parametr `i`, díky kterému je možné realizovat i pohyb tohoto nápisu.

4 Literatura a zdroje

(1) Šimek V. *Minerva.PrjPeb* [online]. 16.17.2013 [cit. 18.12.2020]. Dostupné z: <http://www.fit.vutbr.cz/simekv/schematics%20-%20FITkit%20v3.0.pdf>.

(2) Mrázek V. *Bez názvu, demonstrace programu* [online]. neznámé [cit. 8.11.2020]. Dostupné z: http://www.fit.vutbr.cz/simekv/IMP_projekt%20-%20had_tabule_test.zip

(3) Neznámý. *Lab2* [online]. 16.5.2015 [cit. 18.12.2020]. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fhwlab%2FLab2&cid=13997>

(4) Mrázek V. *Pár slov k projektům* [online]. 22.10.2020 [cit. 18.12.2020]. Dostupné z: http://www.fit.vutbr.cz/simekv/IMP_projekt%20-%20had.pdf