



MIKROPROCESOROVÉ A VESTAVĚNÉ SYSTÉMY
2020/2021

Dokumentace hry HAD

Šesták Pavel(xsesta07)

Brno, December 19, 2021

Contents

1	Úvod	2
2	Implementace	2
2.1	Makra	2
2.2	Definované typy	2
2.3	Globální proměnné	2
2.4	Inicializace MCU	2
2.5	Inicializace časovače	3
2.6	Multiplexing maticového displeje	3
2.7	Obsluha přerušení časovače pro posun hada	3
2.8	Obsluha přerušení tlačítek pro změnu směru hada	3
3	Shrnutí	4
4	Použitá literatura	5

1 Úvod

Cílem tohoto projektu je vytvořit zjednodušenou verzi hry had na výukovém kitu Minerva s reprezentací na maticovém displeji. Pohyb hada po ploše je realizován díky hardwarovým tlačítkům dostupným na výukovém kitu. Zjednodušená verze hada umožňuje pohyb hada po bezbarierové herní ploše o velikosti 8*16. Had může přecházet z jedné strany herní plochy na druhou přes stěny herní plochy. Had se v průběhu hry zrychluje. V případě, že se had kousne, tak se hra automaticky restartuje a pípne bzučák. Demonstrační video k projektu je na adrese: https://www.stud.fit.vutbr.cz/~xsesta07/Personal/projects/IMP_snake/snake.php

2 Implementace

2.1 Makra

V rámci projektu byly definovány makra pro slovní zastoupení bitových pozic v rámci zdrojového textu pro lepší čitelnost. Jsou definovány pro tlačítka i piny pro obsluhu displeje. Dále jsou zde makra pro nastavení velikosti herního pole a délky hada.

2.2 Definované typy

V rámci projektu byla definovaná struktura Cell, která reprezentuje jednu buňku na displeji. Pro přehlednější práci s pohybem hada byl definován výčetový typ, reprezentující směr pohybu hada.

```
1 typedef struct {
2     int col;
3     int row;
4 } Cell;
5
6 typedef enum {
7     UP,
8     DOWN,
9     LEFT,
10    RIGHT
11 } Direction;
```

2.3 Globální proměnné

Celkový stav hry je reprezentován globálními proměnnými. Dále mezi globálními proměnnými jsou i pomocná pole, která slouží pro iteraci, přes maticový displej a tlačítka pro jejich inicializaci a snížení redundance kódu.

```
1 unsigned int compare; //Compare value for low power timer
2 Direction direction; //Represent current direction of snake
3 char currentCol = 0; //Represent index for column multiplexing
4 Cell snake[SNAKE_LEN]; //Snake representation
5 int head; //Index in array of snake head
6 int gameRun; //indicate if is game, or game over
7 const unsigned int rows[8] = {ROW_0, ROW_1, ROW_2, ROW_3, ROW_4, ROW_5, ROW_6,
8     ROW_7 }; //Rows on matrix display
9 const unsigned int cols[4] = {COL_0, COL_1, COL_2, COL_3 }; //Columns on matrix
10    display
11 const unsigned int buttons[4] = { 10, 12, 27, 26 }; // Buttons to control snake
```

2.4 Inicializace MCU

V rámci funkce MCUInit se nastaví parametry hodinového signálu a vypne se watchdog, který pro tuto aplikaci nebudeme potřebovat. Dále se provede inicializace portů na mikrokontroleru. V rámci inicializace se povolí hodinový signál pro port A a port E. Na portu A máme připojený maticový displej a na portu E jsou připojena tlačítka. Tlačítka nastavíme jako GPIO piny a zapneme jim pull-up rezistory na mikrokontroleru. Dále je nutné povolit přerušení na portu E, pro odchyt stisknutí tlačítka. Dále nastavíme porty na portu A pro displej jako GPIO výstupní piny. V poslední řadě je portu A nastaven jako výstupní port 4 pro obsluhu bzučáku.

2.5 Inicializace časovače

Časovač je nastaven, aby generoval přerušení, které budeme v obslužné rutině obsluhovat. Před dělička je nastavena na 2.

2.6 Multiplexing maticového displeje

Rotace samotného displeje probíhá ve funkci rotateDisplay. Napřed se vyčistí output registr, dále se pro iteruje celý řádek v daném sloupci a zkontroluje se podmínkou, zda se na dané pozici nenachází had, v případě, že ano tak pomocí globálního pole rows se zapíše do výstupního registru. Následně se do output registru nastaví stávající index sloupce pomocí vzorce:

$\text{GPIOA_PDOR} |= (\text{currentCol} \ll i) \gg i \ll \text{cols}[i]$. V poslední řadě se zvýší index sloupce a spustí se delay pro další překreslení. Rychlá změna sloupce spolu se setrvačností lidského oka působí, že displej svítí.

2.7 Obsluha přerušení časovače pro posun hada

Nejprve se sníží počítadlo pro další vyvolání přerušení, to slouží pro průběžné zrychlování hada a zapíše se do clear příznaku, že bylo dané přerušení obslouženo. Nejnižší hodnota komparátoru je 0x20. Díky globálním proměnným snake a head získáme aktuální buňku, která reprezentuje hlavu hada. Na základě směru hada modifikujeme index sloupce či řádku. Tyto modifikace jsou implementovány v separátních funkcích a řeší přechody mezi stěnami. Dále se zkontroluje, zda na nové pozici hlavy již není jiná část hada, v případě že ano došlo ke kolizi a hra je restartována. Restart je doprovázen zvukem bzučáku. V případě, že kolize nenastala tak se posune index hlavy na poslední článek hada a vloží se na jeho pozici nová hlava.

2.8 Obsluha přerušení tlačítek pro změnu směru hada

Obsluha přerušení tlačítek probíhá na portu E. Napřed spustíme delay, pro odfiltrování zákmitu na tlačítku. Dále se kontroluje přes ISFR registr jaké konkrétní tlačítko bylo stisknuto a zda je pořád stisknuto pomocí input registru. V případě shody se nastaví příslušný směr do globální proměnné direction. Následně se vyčistí příznaky přerušení.

3 Shrnutí

Při testování byl otestován průchod skrz hrany displeje a restart hry při "kousnutí" se hadem. Dále by bylo možné doimplementovat ovoce a na konci hry zobrazit na maticovém displeji skóre. Testováno bylo na poskytnutém kitu Minerva s dvojicí maticových displejů.

4 Použitá literatura

Při vypracovávání řešení bylo čerpáno z vlastních zdrojových textů napsaných v rámci laboratorních cvičení. Dále bylo čerpáno z referenčního manuálu podrodiny K60 mikrokontrolerů od NXP dostupný na adrese: <https://www.ic-components.cz/files/ad/MK60FX512VMD12.pdf>. Dále bylo čerpáno ze schéma obvodového zapojení výukového kitu Minerva, který je dostupný na adrese: <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fexcs%2FFITkit3-schema.pdf>. V poslední řadě pro ucelení informací byla využita prezentace k danému tématu od pana Ing. Václava Šimka Ph.D., který je i autorem tohoto zadání, dostupná na adrese: https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fprojects%2FIMP_projekt+-+had.pdf.