

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Mikroprocesorové a vestavěné systémy
Simulace v CW: Světelné noviny

Obsah

<i>Úvod</i>	3
<i>Implementace</i>	3
<i>Významné části kódu</i>	3
<i>Proměnné a jejich datové typy</i>	5
<i>Analýza paměťových nároků aplikace</i>	5
<i>Přílohy</i>	6

Úvod

Cílem projektu je vytvořit pro prostředí Freescale Codewarrior verze 6.x aplikaci, která bude schopna zobrazit na LED displeji část loginu a po stisku příslušného tlačítka začít rotovat uložený text ve vertikálním či horizontálním směru, přičemž rychlost rotace je možno měnit pomocí táhla.

Implementace

Tvorba celého programu se dá rozdělit na několik částí – definice pole obsahujícího login připravený pro zobrazování, konfigurace prvků displeje v souboru LEDnoviny32cx8r.vtl, inicializace a zobrazení prvních čtyřech znaků loginu a nakonec vytvoření rotace, rychlosti pohybu textu na displeji a detekce stisknutých tlačítek. Před samotnou implementací bylo ještě třeba zvolit si programovací jazyk, ve kterém bude aplikace napsána. Já zvolila jazyk C, protože se nejedná o aplikaci s vysokými nároky na hardware.

Mapování loginu je možno provést po sloupcích nebo po řádcích, v mém projektu jsem se rozhodla jej namapovat po sloupcích, jelikož se takto snáz získává číselná reprezentace rozsvícených diod a při úpravě tvaru písmen jsou změny jednodušší a zůstávají v rámci jednoho znaku. Další výhodou je práce s menším rozsahem čísel, největší možná hodnota je 255, zvolila jsem tedy datový typ `unsigned char`. Pole má délku 64, jelikož na každý znak připadá šířka osm sloupců a login sestává z osmi znaků. Jednotlivé hodnoty pole pak představují součet poloh všech diod, které mají být v daném sloupci rozsvíceny.

Dále bylo potřeba provést konfiguraci prvků displeje. Úseky paměti, na které lze zapisovat, jsou uvedeny v souboru Project.prm. Jelikož se pole loginu automaticky ukládá od adresy 0x0100, prvky displeje jsou mapovány do části Z_RAM od adresy 0x00B0. Od tohoto místa v paměti začíná mapování jednotlivých diod displeje. Poté následuje mapování jednotlivých tlačítek a posuvníku. Později bylo do tohoto souboru potřeba přidat i nastavení obnovování displeje.

Počáteční inicializace zahrnuje vynulování paměti, kam jsou mapována tlačítka, a nastavení počáteční hodnoty posuvníku rychlosti. Inicializace displeje probíhá ve `for` cyklu iterujícím přes všechny sloupce diod.

Poslední fází implementace byl druhý cyklus `for`, ve kterém probíhá detekce stisku tlačítek pomocí logického součinu s jedničkou, rotace loginu v horizontálním i vertikálním směru, a to o počet bitů, který je inkrementován v každém průchodu cyklem po stisku příslušného tlačítka. Před dalším průchodem cyklu je realizováno zpoždění s využitím hodnoty získané z posuvníku rychlosti rotace.

Významné části kódu

Jádrem celého programu je funkce `main`. V ní se nachází deklarace lokálních proměnných, a to proměnné `direction`, která slouží k ukládání směru rotace a informace, jaké tlačítko bylo stisknuto, dále proměnné `coord_x` a `coord_y`, které slouží k čítání, o kolik bitů má login orotovat v horizontálním a vertikálním směru. Poté jsou inicializovány globální proměnné na implicitní hodnoty, ty jsou u tlačítek 0, jelikož nejsou stlačeny, a u posuvníku jsem jako počáteční hodnotu zvolila polovinu rozsahu, tedy 127. Následně se program zanořuje do prvního nekonečného cyklu `for`, který představuje stav inicializace, tedy stav po stisku tlačítka *init*. V této části jsou inicializovány proměnné `coord_x` a `coord_y` a na displeji jsou zobrazeny první čtyři znaky loginu.

Nyní aplikace vstupuje do druhého, zanořeného nekonečného cyklu `for`, ve kterém je testován stisk tlačítka a případně uložen nový směr rotace. Pokud bylo stisknuto tlačítko *init*, je využito příkazu `break` pro přechod do vnějšího cyklu `for` a stavu inicializace. Když je nastaven směr rotace pomocí jednoho z tlačítek *horiz* a *vert*, je inkrementována proměnná `coord_x`, nebo `coord_y` a je provedena rotace loginu o zvýšený počet bitů. Jako poslední je zavolána funkce zajišťující zpoždění před další rotací, a to na základě hodnoty odečtené z posuvníku.

```
void main(void) {
    unsigned int direction = 0;
    unsigned int coord_x = 0;
    unsigned int coord_y = 0;

    set_global_variables();

    for (;;) {
        initialize_variables(&coord_x, &coord_y);
        reset_display();

        for (;;) {
            __RESET_WATCHDOG(); /* feeds the dog */

            get_direction(&direction);

            if (direction == 0) {
                break;
            }
            increment_rotation_bits(direction, &coord_x, &coord_y);

            rotate_login(coord_x, coord_y);
            delay_next_rotation();
        }
    }
}
```

Další funkce, která stojí za zmínku, je `rotate_login`, ve které jsou diody procházeny po sloupcích a pro každý sloupec je nejprve provedena horizontální rotace. Ta je realizována pouze pomocí indexování pole loginu tak, aby došlo k rotaci ve směru zleva doprava. Poté je nad získanou hodnotou sloupce provedena i vertikální rotace. K její realizaci je použit bitový posuv a logický součet.

```
void rotate_login(unsigned int coord_x, unsigned int coord_y) {
    int i;
    for (i = 0; i < 32; i++) {
        display[i] = login[(i - (coord_x % 64) + 64) % 64];
        display[i] = (display[i] << (coord_y % 8)) | (display[i] >> (8 -
(coord_y % 8)));
    }
}
```

Proměnné a jejich datové typy

Tři proměnné deklarované na začátku funkce `main` jsou již popsány výše. Co se týče hodnot, kterých nabývají, u proměnné `direction` jsou to tři hodnoty – 0 při inicializaci a po stisku tlačítka `init`, 1 po stisku tlačítka `horiz` a 2 po zmáčknutí tlačítka `vert`. Proměnné `coord_x` a `coord_y` jsou na počátku inicializovány na 0 a po stisku příslušného tlačítka inkrementovány až do přechodu zpět do stavu inicializace. V případě dlouhého běhu programu a rotování loginu by mohlo dojít k přetečení, to by však uživatel neměl nijak zaregistrovat, rotace přirozeně přejde na rotaci o 0 bitů. Všechny tyto proměnné mají datový typ `unsigned int` právě kvůli možnosti dlouhodobé inkrementace. U proměnné `direction` by byl zcela dostačující i datový typ `char`.

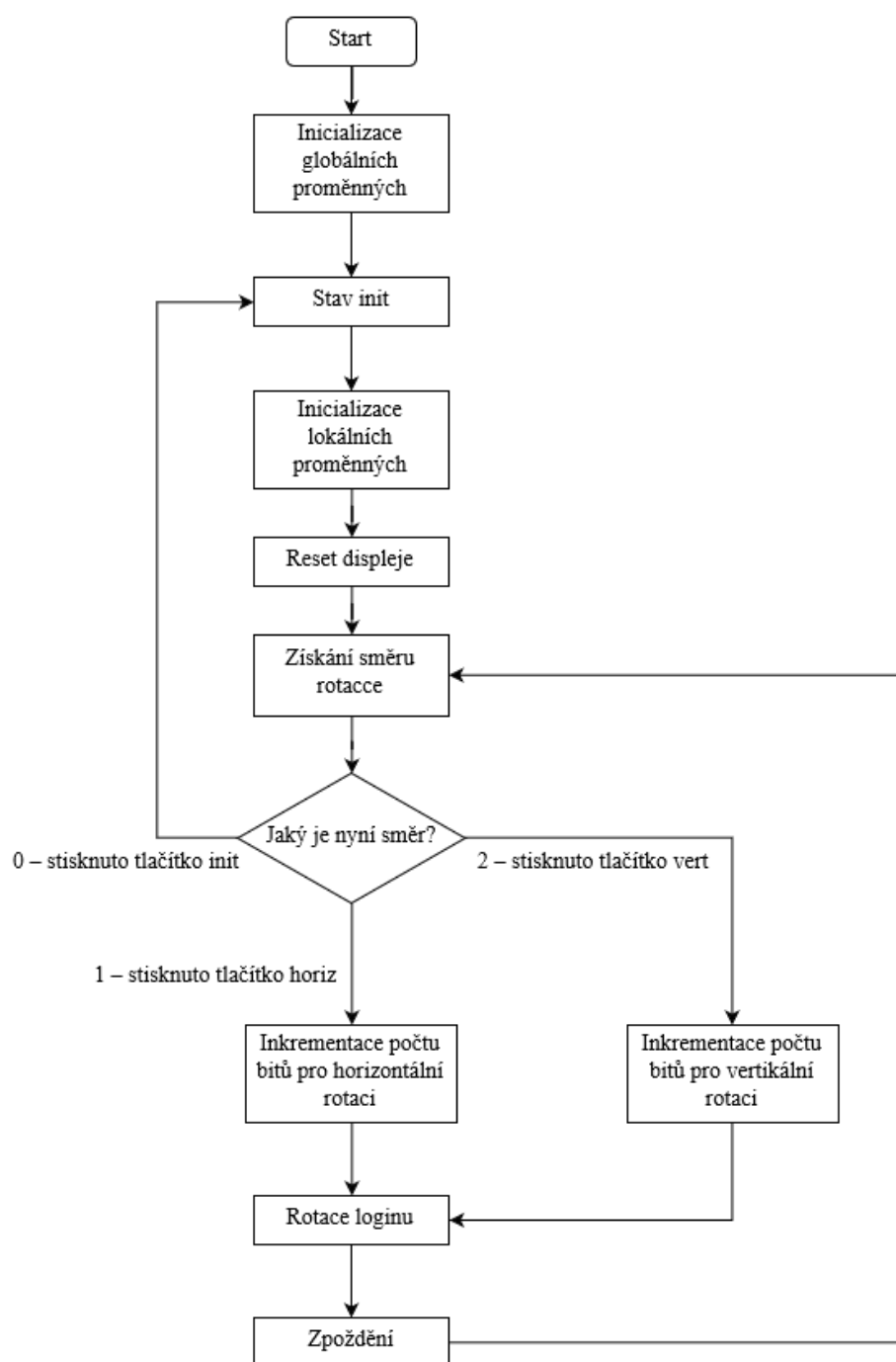
V programu je dále deklarováno šest globálních proměnných. První je pole `login` obsahující 64 prvků typu `unsigned char`, z nichž každý představuje jeden sloupec diod. Tyto prvky tedy nabývají hodnot v rozsahu 0 až 255, podle polohy rozsvícených diod. Ostatní globální proměnné jsou naplněny odkazy na jiné části paměti, kam jsou namapovány prvky nástroje Visualization Tool. Tlačítka jsou odkazována třemi ukazateli na `unsigned char` `init`, `horiz` a `vert`. Odkazovaná paměť může nabývat pouze hodnot 0 a 1 v závislosti na tom, zda je stlačeno příslušné tlačítko. Další proměnná stejného typu má název `speed` a odkazuje místo v paměti, na které se mapuje hodnota odečítaná z posuvníku rychlosti rotace. Poslední globální proměnnou je proměnná `display` ukazující na první místo v paměti, od kterého se mapují jednotlivé diody na základě konfigurace v souboru `LEDnoviny32cx8r.vtl`.

Ve funkci `delay_next_rotation` jsou vytvořeny dvě lokální proměnné, `counter`, která je inkrementována ve `while` cyklu při realizaci zpoždění, a `wait`, do které je uložena délka zpoždění, jenž je nepřímo úměrná k rychlosti rotace. Jelikož obě tyto proměnné mohou nabývat vysokých a pouze kladných hodnot, jsou taktéž datového typu `unsigned int`. Aby byla hodnota proměnné `wait` vždy kladná, je hodnota proměnné `speed` odečítána od čísla 256, tedy o 1 vyšší než je maximální možná hodnota proměnné `speed`.

Analýza paměťových nároků aplikace

Nároky na RAM	414 B
Nároky na Flash	74 B
Velikost kódu	157 řádků

Přílohy



Vývojový diagram popisující činnost aplikace

```

*****
DEPENDENCY TREE
*****
main and _Startup Group
|-- main
|   |-- set_global_variables
|   |-- initialize_variables
|   |-- reset_display
|   |-- get_direction
|   |-- increment_rotation_bits
|   |-- rotate_login
|   |-- delay_next_rotation
|       |-- _IMUL_STAR08
|-- _Startup
|   |-- Init
|       |-- loadByte
|-- main                                     (see above)

```

Popis aplikace pomocí stromu závislostí