

Vysokého učení technického v Brně

Fakulta informačních technologií



Projekt do předmětu IMP

Zadání: Simulace v CW: Světelné noviny

Obsah

1. Úvod	3
1.1. Popis zadání	3
2. Popis řešení	3
2.1. void tisk_znaku(const byte bitmapa_znaku[8][8], short start_x, short start_y)	4
2.2. void tisk_login(short start)	4
2.3. void clear_display(void)	4
2.4. void zpozdeni(unsigned int n)	4
3. Grafický popis činnosti aplikace - vývojový diagram	5
4. Paměťové nároky aplikace	6

1. Úvod

Cílem projektu bylo vytvořit ve vývojovém prostředí CodeWarrioru simulaci světelných novin. K zobrazení je použit maticový display o rozměrech 8x32 LED diod, který v simulaci zobrazujeme pomocí nástroje VisualisationTool. Program pro platformu HC(S)08 bylo možné napsat v jazyce C nebo Assembler.

1.1. Popis zadání

Program po inicializaci zobrazí první 4 písmena loginu autora. Pod LED displejem se nachází tlačítka init, horiz, vert a horizontální posuvník. Stisk tlačítka init je možné provést kdykoliv během běhu programu a slouží k návratu do výchozího stavu. Stiskem tlačítka horiz dochází k rotaci loginu autora to směrem zleva doprava. Stisknutím tlačítka vert dojde k rotaci loginu autora směrem zdola nahoru. Posledním ovládacím prvkem je horizontální posuvník, který slouží k nastavení rychlosti rotací. Jednotlivé LED diody písmen loginu jsou vysvíceny modrou barvou.

2. Popis řešení

Program je napsán v jazyce C a prostředí CodeWarrioru verze V6.3.

První úkolem bylo správně namapovat LEDky a tlačítka. Vybral jsem mapování po bytech, kvůli pohodlnějšímu ovládání LEDek, ovšem vzhledem k paměťovým nárokům by bylo lepší zvolit mapování po bitech. Tlačítka se v paměti RAM nachází na adresách 0x200 – 0x202. Horizontální posuvník je namapován na adresu 0x203. LED diody se nachází na adresách 0x300 – 0x3ff. Adresy jsou v 16čkové soustavě.

Dalším krokem bylo vytvoření bitmapy jednotlivých písmen zobrazovaného loginu. K tomu jsem využil globální konstanty typu **byte**, která zde identifikuje trojrozměrné pole. První rozměr je počet písmen v loginu, v našem případě se jedná o 8 písmen. Optimalizace zde nejde provést, jelikož se zde žádné písmeno neopakuje. Druhý a třetí rozměr pole reprezentuje velikost matice písmene loginu. V našem případě se jde o rozměr 8x8.

Pro zapínání/vypínání jednotlivých diod slouží funkce *tisk_znaku()*, která na požadovaném místě (podle souřadnic X a Y) rozsvítí požadovaný znak. Tato funkce je následně volána funkcí *tisk_loginu()*.

Jelikož nástroj VisualisationTool neumožňuje generovat přerušení, bylo nutné kontrolovat stisk tlačítka v hlavní funkci (main) programu. Občas se stávalo, že bylo potřeba tlačítko zmáčknout víckrát. Při reálném nasazení, by ovšem bylo pro kontrolu stisku tlačítka použito přerušení.

2.1.void tisk_znaku(const byte bitmapa_znaku[8][8], short start_x, short start_y)

Funkce slouží pro výpis jednoho znaku na vypočítaných souřadnicích maticového displeje. Proměnná *bitmapa_znaku* slouží k předání bitmapy s namapovanými vlastnostmi daného písmene. Proměnné *start_x* a *start_y* slouží pro předání souřadnice prvního bodu znaku.

2.2.void tisk_login(short start)

Funkce sloužící pro korektní vykreslení celého loginu. Stará se o správné rozložení viditelných písmen. Pro každý znak loginu volá funkci *tisk_znaku()*. Proměnná *start* slouží pro předání počátečního bodu loginu.

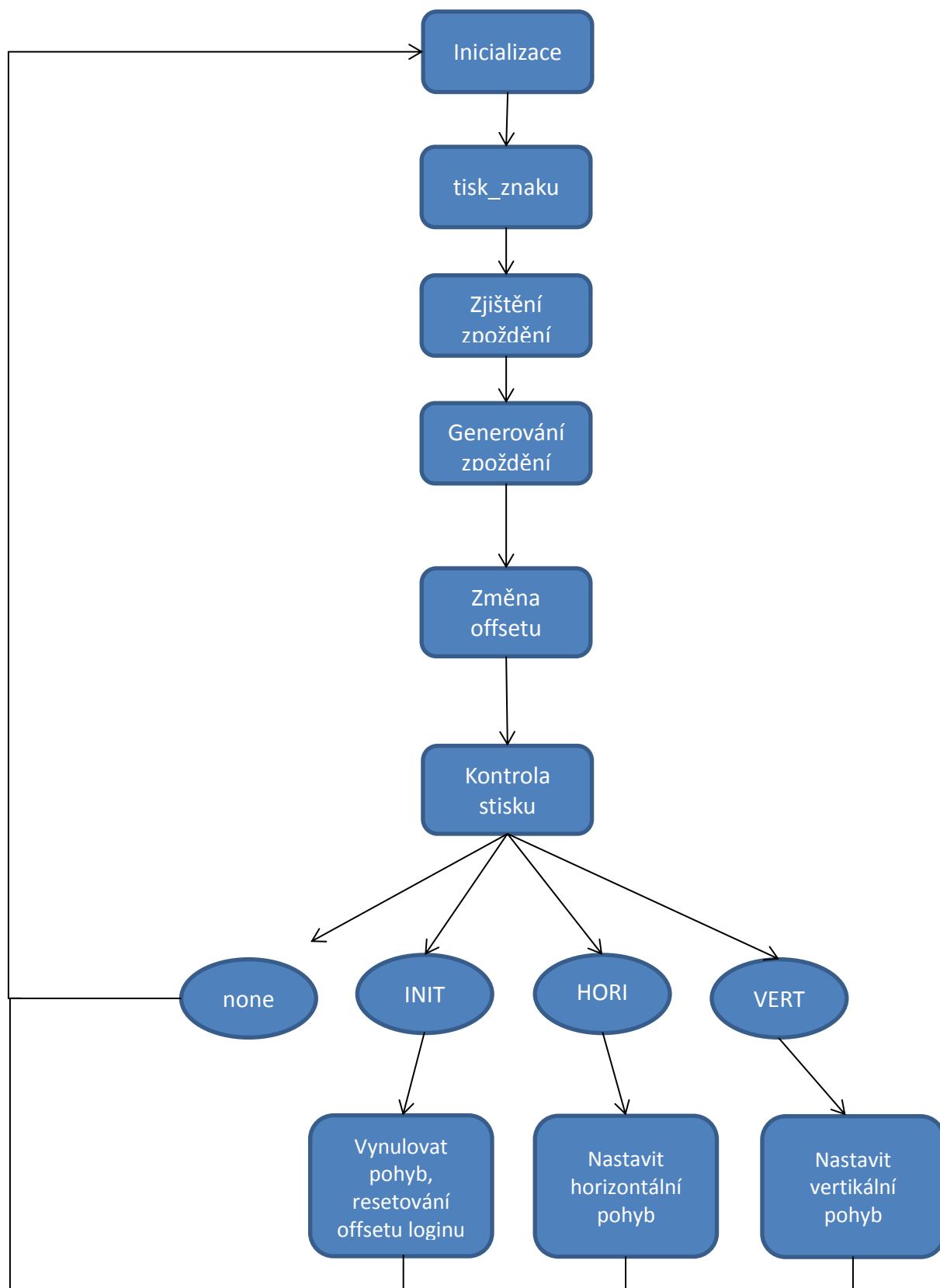
2.3.void clear_display(void)

Funkce starající se o vyčištění displeje. Obsahuje jednoduchý *for* cyklus, který nuluje byty LED displeje.

2.4.void zpozdeni(unsigned int n)

Funkce slouží ke generování zpoždění programu. Proměnná *n*, předává hodnotu, se kterou se počítá v cyklu. Určuje pouze počet opakování cyklu. Pro zpoždění se používá generování určitého počtu *nop* instrukcí z Assembleru.

3. Grafický popis činnosti aplikace - vývojový diagram



4. Paměťové nároky aplikace

Nároky aplikace na RAM paměť: 14072 Bytů

Nároky souboru *main.c* na RAM paměť: 1268 Bytů

Nároky aplikace na FLASH paměť: 2205 Bytů

Nároky souboru *main.c* na FLASH paměť: 9 Bytů

Velikost kódu: 279 řádků

Použité datové typy:

- unsigned char
- short int
- byte
- unsigned int

Vlastní makra:

- *DELKA_RADKU* – určuje délku řádku displeje
- *VYSKA_SLOUPCE* – určuje počet LED diod na výšku displeje
- *MAX_INDEX_LED* – poslední index pole LED
- *POCET_LED* – počet LED diod displeje
- *RADKY_PISMENE* – určuje počet diod na výšku znaku
- *SLOUPCE_PISMENE* - určuje počet diod na šířku znaku
- *PISMENA_LOGINU* – počet znaků v loginu