

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií



Mikroprocesorové a vestavěné systémy

Simulace v CW

Světelné noviny

OBSAH

1	ÚVOD	2
2	NÁVRH	2
2.1	Posuv	2
2.1.1	<i>Posuv vpravo</i>	2
2.1.2	<i>Posuv hore</i>	3
2.1.3	<i>Rýchlosť posuvu</i>	3
3	METRIKY	3
4	VÝVOJOVÝ DIAGRAM	4

1 ÚVOD

V tomto projekte sme mali vytvoriť Svetelné noviny pre True-Time Simulátor v prostredí Code-warrior v jazyku C alebo assembleru. Svetelné hodiny majú zobrazovať na displeji časť prihlasovacieho mena, ktoré je uložené v pamäti mikrokontroléru. Každý znak je reprezentovaný bitmapou v rastru o veľkosti 8 riadkov a 8 stĺpcov.

2 NÁVRH

Ako prvé a nastavíme ukazovatele na správnu pozíciu. Pole ukazovateľov „login“ sa naadresuje od pozície 612. Toto pole znázorňuje jednotlivé stĺpce. Pomocou funkcie setlogin() sa priradia do pamäti hodnoty, podľa ktorých sa určí ktoré ledky budú svietiť.

Ďalej sa priradia hodnoty ukazovateľov „init“, „RIGHT“, „UP“ a „bar“ na adresy jednotlivých tlačítok a baru.

2.1 POSUV

Login sa má pohybovať v dvoch smeroch, vertikálne a horizontálne. Posuv v ktorom smere bol zadáný podľa posledného dvojčísla v logine. Keďže sa môj login končí číslami 08, tak som za úlohu dostal posuv vpravo a posuv hore.



Obrázok 1 Tlačítka pre posuv

2.1.1 Posuv vpravo

Po stlačení tlačítka „horiz“ z obrázka [1], sa text začne pohybovať horizontálne do pravej strany. Posuv som vyriešil pomocou algoritmu [1], kde sa postupne presúvajú jednotlivé stĺpce o jedno doprava a posledný sa dostane na prvú pozíciu.

```
int i;
unsigned char tmp;
tmp = *login[dlzka-1];
for(i = dlzka-1; i > 0 ; i--)
{
    *login[i] = *login[i-1];
}
*login[0] = tmp;
```

Algoritmus 1 – Kód pre pohyb vpravo

2.1.2 Posuv hore

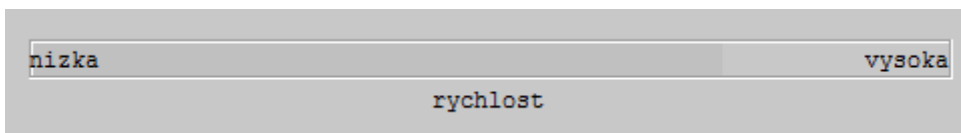
Po stlačení tlačítka „vert“ z obrázka [1], sa text začne pohybovať hore. Pre pohyb hore využívam algoritmus [2], v ktorom využívam binárny posun vpravo. Pri tomto posuve musím kontrolovať, či sa na poslednej pozícii nachádza 1 alebo 0. To zisťujem tak, že zistím či je číslo na danej pozícii párne alebo nepárne.

```
int i;
for(i = 0; i <dlzka; i++)
{
    if(*login[i] %2 ==0)
    {
        *login[i] = *login[i] >> 1;
    }
    else
    {
        *login[i] = *login[i] >> 1;
        *login[i] = *login[i] + 128;
    }
}
```

Algoritmus 2 – Kód pre posuv hore

2.1.3 Rýchlosť posuvu

Rýchlosť posuvu sa nastavuje podľa baru z obrázku [2]. Z testovania mi vyšlo, že ideálna podmienka cyklu, na spomalenie je $(3500 - (*bar * 10))$.



Obrázok 2 – Nastavenie rýchlosti posuvu

3 METRIKY

Veľkosť kódu: 166 riadkov

Nárok na RAM: 268B

Nárok na FLASH: 804B

4 VÝVOJOVÝ DIAGRAM

