

VYSOKÉ UČENIE TECHNICKÉ V BRNE FAKULTA INFORMAČNÝCH TECHNOLOGIÍ



Mikroprocesorové a vstavané systémy (IMP) 2016/2017

Simulácia v CodeWarrior

Konfigurovateľný OCT čítač

Obsah

1	Úvod	2
2	Návrh aplikácie	2
3	Popis aplikácie	2
4	Významné úseky kódov	4
5	Analýza pamäťových nárokov	4
6	Vývojový diagram	5
7	Záver	5

1 Úvod

Úlohou tohto projektu bolo v jazyku C, alebo Assembler napísať Real-Time simulátor konfigurovateľného OCT čítača v prostredí CodeWarrior verzia 6.x. Aplikácia má byť schopná zobrazíť, inkrementovať, dekrementovať a upravovať hodnoty čítacieho registra.

2 Návrh aplikácie

Pre implementáciu zadania som zvolil jazyk C, vzhľadom na komplexnosť a jednoduchosť riešenia úlohy, kde nie je až taká podstatná pamäťová náročnosť.

Vnútrotný stav čítača (CNT), je implementovaný pomocou pol'a integerov o štyroch prvkoch, kde každý reprezentujem jeden zo štyroch nibblov. Dohromady je jeho veľkosť 16.

Čítač bude prechádzať medzi stavmi pomocou premennej MODE, ktorá bude označovať číselnú hodnotu aktuálneho stavu aplikácie.

3 Popis aplikácie

Program po spustení na začiatku zobrazuje vynulované všetky prepínače a zhasnuté obe segmentové displeje. Nainicializuje vnútrotný stav čítača CNT na hodnotu nula. Čítač je v režime STOP.

Po zapnutí(prenutí prepínača 7 - prvý zľava) samotného čítača vo Vizualization tool sa rozsvietia obe displeje.

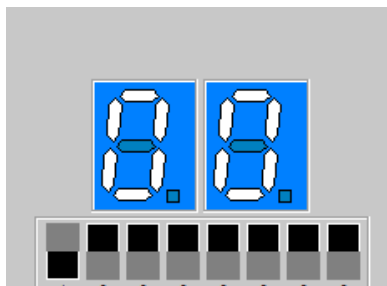


Figure 1: Spustenie čítača

V tomto stave (šiesty prepínač v dolnej pozícii) je čítač v režime SET-ľavý-Seg, kde blinká ľavý segmentový displej a pomocou posledných dvoch prepínačov je možno meniť hodnotu ľavého displeja, ktorý reprezentuje index CNT rozdeleného na 4 nibbly, CNTN[0-3].

Po prepnutí štvrtého prepínača čítač prejde do stavu SET-pravý-seg, kedy hodnota na ľavom seg7 reprezentujúca index jedného zo štyroch niblov CNT. Pravý-seg7 sa rozblinká a reprezentuje hodnotu indexu niblu CNT svietiacom na ľavom seg7. Poslednými tromi prepínačmi môžeme túto hodnotu na pravom seg7 meniť.

V prípade prepnutia tretieho prepínača sa CNT nastaví na maximálnu hodnotu.

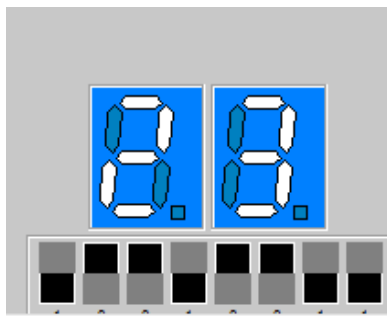


Figure 2: Čítač v režime SET

Naopak po prepnutí šiesteho prepínača do hornej pozície sa dostaneme do stavu čítania START. V prípade prepínača päť v dolnej pozícii to je čítanie smerom dole a v hornej pozícii čítanie smerom hore.

Pri režime START čítač inkrementuje/dekrementuje stav CNT. Na ľavom seg7 sa zobrazuje index najpomalšie sa meniaceho nibblu CNT a na pravom seg7 hodnota daného indexu CNT v osmičkovej sústave.

	0.	1.	2.	3.
Krok 1	0000	0000	0000	0000
			
Krok x	0000	0000	0000	0111
Krok x+1	0000	0000	0001	0000
			
Krok y	0000	0000	0111	0111
Krok y+1	0000	0001	0000	0000

Figure 3: Príklad inkrementovania hodnoty v CNTN znázornený v dvojkovej sústave

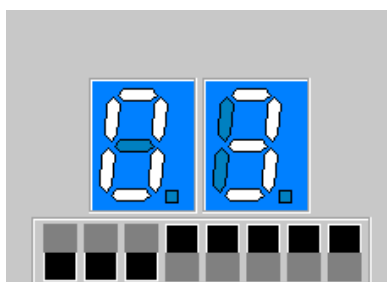


Figure 4: Čítač v režime START

Čítač využíva modul Real-Time Counter (RTC) dostupný na čipe HCS08, pre meranie času a správu prerušenia. Jeho funkcionality bola zistená z datasheetu HCS08[1]. Pre jeho činnosť bolo nutné v projektovom súbore .prm povoliť prerušenie a zadeklarovať ho na prvej voľnej adrese 0xFFC4.

4 Významné úseky kódov

Funkcia zachytávania prerušenia z modulu RTC.

```
interrupt 29 void RTC_ISR(void) {  
    run();  
}
```

Funkcia samotného čítača, kde inkrementuje hodnoty CNTN rozdeleného na 4 nibbly.

```
void countUp() {  
    if(CNTN[3] < 7) {  
        CNTN[3] = CNTN[3] + 1;  
    } else if(CNTN[2] < 7) {  
        CNTN[2] = CNTN[2] + 1;  
        CNTN[3] = 0;  
    } else if(CNTN[1] < 7) {  
        CNTN[1] = CNTN[1] + 1;  
        CNTN[3] = 0;  
        CNTN[2] = 0;  
    } else if(CNTN[0] < 7) {  
        CNTN[0] = CNTN[0] + 1;  
        CNTN[3] = 0;  
        CNTN[2] = 0;  
        CNTN[1] = 0;  
    } else if(CNTN[0] == 7) {  
        blikanie();  
    }  
    for(i=0; i<4; i++) {  
        if(CNTN[i] > 0) {  
            segShow(i, seg1);  
            segShow(CNTN[i], seg2);  
            break;  
        }  
    }  
}
```

5 Analýza pamäťových nárokov

Nároky na RAM	1335B
Nároky na pamäť Flasg	154B
Veľkosť kódu	394 riadkov

Table 1

6 Vývojový diagram

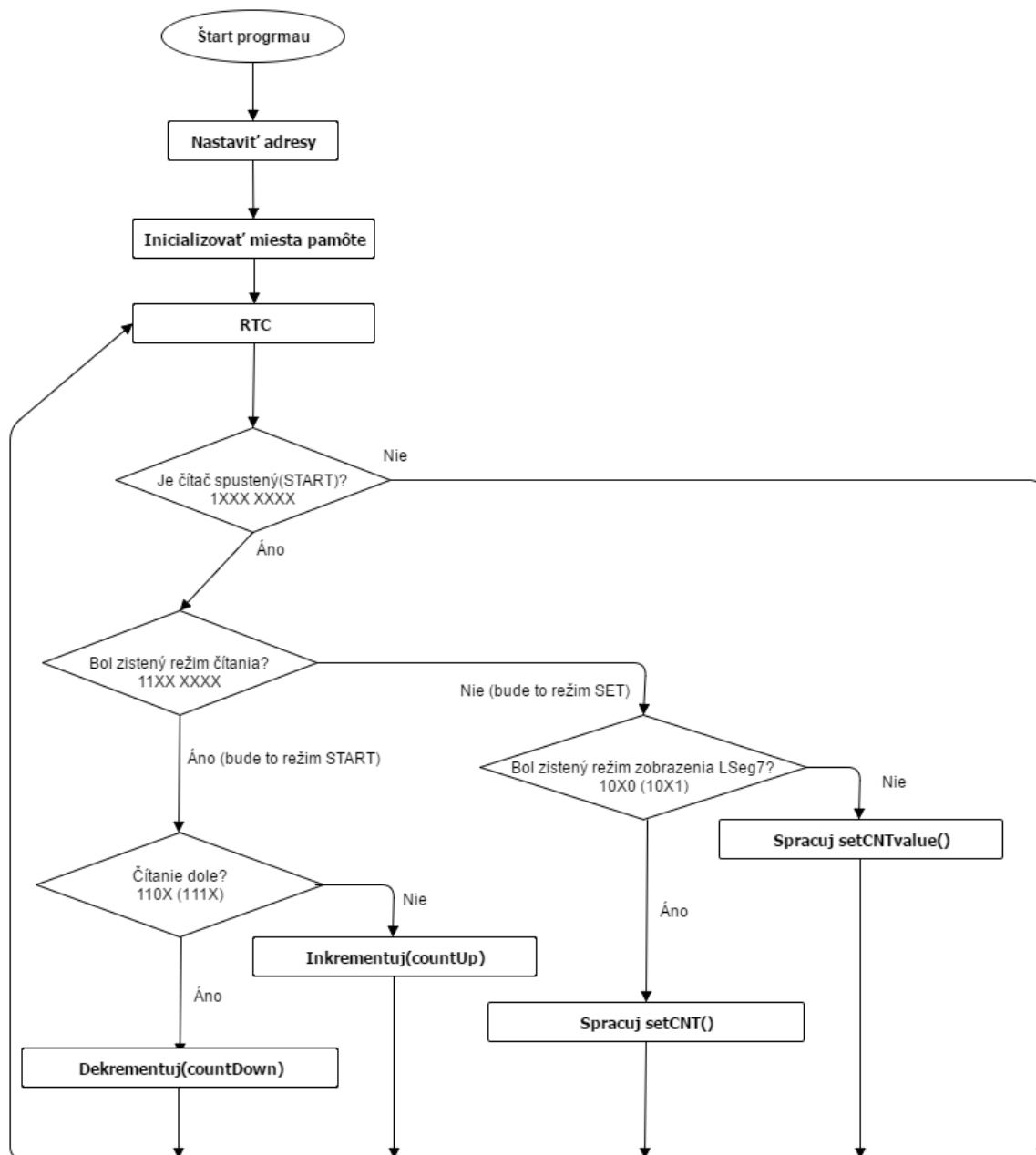


Figure 5

7 Záver

Program bol riadne otestovaný, behom tohto testovania sa nevyskytli, žiadne závažné chyby. Zadanie projektu bolo dodržané, pričom bolo navyše pridané správanie prepínača, ktorý nemal v zadaní žiadnu funkciu (prepínač tri).

Program bol vytvorený v aplikácii CodeWarrior od firmy Freescale, ktorý nám poskytlo VUT v Brne, Fakulta Informačných technológií ako verziu pre účely školských projektov a za pomociu školou vytvoreného a mnou upraveného Visualization Tool.

References

- [1] Freescale. *MC9S08JM60 MC9S08JM32 Data Sheet [online] [HCS08 Microcontrollers]*. 1/2009 [cit.2016-12-01].
https://www.fit.vutbr.cz/study/courses/IMP/private/VYUKA/CVICENI_A_LABORATORE/mc9s08jm60.pdf