

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Mikroprocesorové a vestavěné systémy

**Simulace v CW: Konfigurovatelný DEC čítač**

## Obsah

Úvod .....	1
Uživatelské rozhraní .....	1
Datové typy a proměnné.....	1
Implementace.....	2
Důležité části kódu .....	3
Analýza paměťových nároků aplikace .....	4
Přílohy.....	5

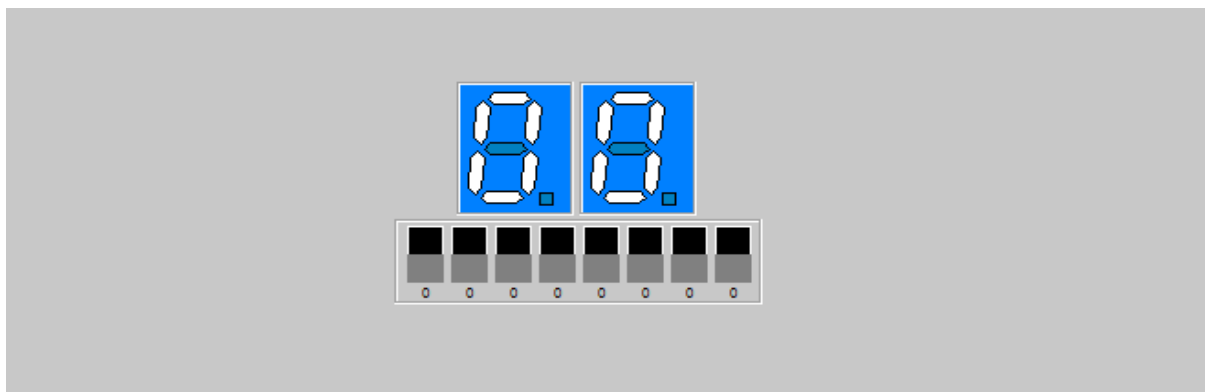
## Úvod

Projekt má za cíl vytvořit aplikaci pro mikrokontrolér řady HCS08, která bude sloužit jako čítač. Uživatelé bude umožňovat navolit si počáteční hodnotu, od které potom bude zahájen odpočet (funkcionalita podobná aplikaci typu minutka). Dále bude uživateli poskytovat volbu směru čítání a skončení odpočtu bude signalizovat blikáním displeje. Čítání je prováděno v dekadické číselné soustavě v rozsahu 9999 až 0.

Aplikace je tvořena a testována v prostředí Freescale Codewarrior verze 6.x. Při implementaci je použit jazyk C.

## Uživatelské rozhraní

Komunikace s uživatelem je prováděna prostřednictvím dvou sedmisegmentových displejů (levý L7seg, pravý R7seg) a osmi přepínačů (InputVec, DILswitch).



Přepínače po řadě zleva umožňují: spuštění aplikace, volbu mezi nastavováním a čítáním, volbu směru čítání, nastavování indexu nebo hodnoty a poslední čtyři přepínače slouží k zadávání čísla.

## Datové typy a proměnné

Proměnné jsou uloženy v paměti od adresy 0x100.

InputVec	byte, reprezentuje nastavení ovládacích přepínačů.
L7seg	byte, reprezentuje rozsvícené segmenty levého displeje
R7seg	byte, reprezentuje rozsvícené segmenty pravého displeje
Seg7	byte, horní 4 bity nesou hodnotu zobrazenou na levém displeji, dolní 4 bity nesou hodnotu zobrazenou na pravém displeji.
InnerVec	byte, slouží pro zajištění neměnné hodnoty převzaté z InputVec v rámci jednoho obslužného cyklu
Last	byte, obsahuje vnitřní stav čítače
Count	byte, počítadlo blikání displejů na konci odpočtu
Flash	byte, počítadlo problikávání displeje při zadávání hodnoty
CNT	int, 4 hodnoty po 4 bitech reprezentující dekadickou hodnotu čítače (0000 až 9999)

Konstanty reprezentující vnitřní stav: `LAST_NONE` počáteční stav, `LAST_LEFT` nastavování indexu, `LAST_RIGHT` nastavování hodnoty, `END_REACHED` konec odpočtu.

## Implementace

Implementace se skládá z několika částí. První je konfigurace prvků z uživatelského rozhraní a jejich namapování na paměť mikrokontroléru, dále inicializace displeje a ovládacích prvků, hlavní cyklus pro nastavování počáteční hodnoty a čítání a nakonec detekce konce čítání. Součástí je také nastavení modulu RTC a obsluhy přerušení

Přepínače jsou namapovány na prvních 8 bitů RAM na adresu `0x100`, tomu odpovídá proměnná `InputVec`, levý sedmisegmentový displej na adresu `0x101`, proměnná `L7seg`, a pravý sedmisegmentový displej na adresu `0x102`, proměnná `R7seg`.

Při spuštění aplikace nebo po resetu je vynulována hodnota `InputVec`, čítače `CNT` a počítadla blikání displejů `Count` a `Flash`. Sedmisegmentové displeje jsou inicializovány hodnotou `0x3F` (zobrazení číslice 0). Vnitřní stav čítače je nastaven na hodnotu `LAST_NONE`.

Nastavení počáteční hodnoty je rozděleno na zadání indexu číslice a hodnoty číslice.

Přepnutím `DILswitch` na hodnotu `10X0XXXX` zadává uživatel index číslice. Vnitřní stav je nastaven na hodnotu `LAST_LEFT` a dolní 4 bity `DILswitch` jsou nastaveny na hodnotu odpovídající právě zobrazované číslici na `L7seg`. To zaručí, že hodnota, která zůstala nastavena v `DILswitch` po předchozím zadávání nebude zaměněna se vstupem uživatele. Rozsah hodnot indexu je 0 až 3, ale prvek `DILswitch` umožňuje zadávat hodnoty v rozsahu 0 až 15, proto pokud uživatel nastaví hodnotu mimo platný rozsah je tato hodnota ignorována a jako vstup je brána poslední zadaná platná hodnota.

Přepnutím `DILswitch` na hodnotu `10X1XXXX` zadává uživatel hodnotu číslice a vnitřní stav je nastaven na hodnotu `LAST_RIGHT`. Zadávání hodnoty se řídí stejnými pravidly jako v případě indexu s rozdílem v rozsahu platných hodnot (0 až 9).

Zadávání hodnot indikuje aplikace problikáváním displeje s nastavovanou hodnotou, řešené použitím proměnné `Flash`. V závislosti na její hodnotě je na displeji zobrazena příslušná hodnota nebo nic. Řízení problikávání (nastavování `Flash`) probíhá pomocí časovače v rámci obsluhy přerušení. Zadaná hodnota je uložena do paměti pomocí funkce `set_cnt()`.

Čítání je zahájeno nastavením `DILswitch` na `110XXXXX` (čítání dolů) nebo `111XXXXX` (čítání nahoru)

V rámci hlavního cyklu jsou při čítání zobrazovány aktuální hodnoty na displeje pomocí funkcí `update_7Seg_down()` a `update_7Seg_up()`. Změna hodnoty `CNT` probíhá při obsluze přerušení vyvolané časovačem.

Konec odpočtu indikuje nastavení vnitřního stavu na hodnotu `END_REACHED`, to je provedeno v rámci obsluhy přerušení v případě, že hodnota `CNT` je 0000 (nebo 9999). Když je aplikace ve stavu konce odpočtu jsou ignorovány vstupy od uživatele a displej problikává. Obsluha přerušení v tomto

stavu inkrementuje hodnotu Count, po dosažení Count=3 je aplikace uvedena do počátečního stavu.

Měření intervalu čítání provádí modul RTC, který periodicky vyvolává přerušení. Přerušení je využíváno při nastavování hodnot pro problikávání displeje periodickým přenastavováním hodnoty Flash, při čítání je v přerušení volána funkce `dec_cnt()` dekrementující hodnotu čítače (nebo `inc_cnt()` při čítání nahoru) a je kontrolována hodnota CNT na dosažení koncové hodnoty.

## Důležité části kódu

Kostru programu tvoří nekonečný cyklus řízený hodnotou zadanou na přepínačích (`InnerVec`) a vnitřním satvem (`Last`). Pokud byl ukončen odpočet je tok řízení programu přesměrován mimo nastavování a odpočet. Jinak je v závislosti na hodnotě `InnerVec` provedena příslušná operace.

```
for(;;) {
    if (Last == END_REACHED)
        Konec odpočtu - problikávání a reset
    else {
        if (InnerVec < 0x80)
            Reset
        else if ((InnerVec & 0xD0) == 0x80)
            Nastavování indexu
        else if ((InnerVec & 0xD0) == 0x90)
            Nastavování hodnoty
        else if ((InnerVec & 0xE0) == 0xC0)
            update_7Seg_down();
        else if ((InnerVec & 0xE0) == 0xE0)
            update_7Seg_up();
    }
}
```

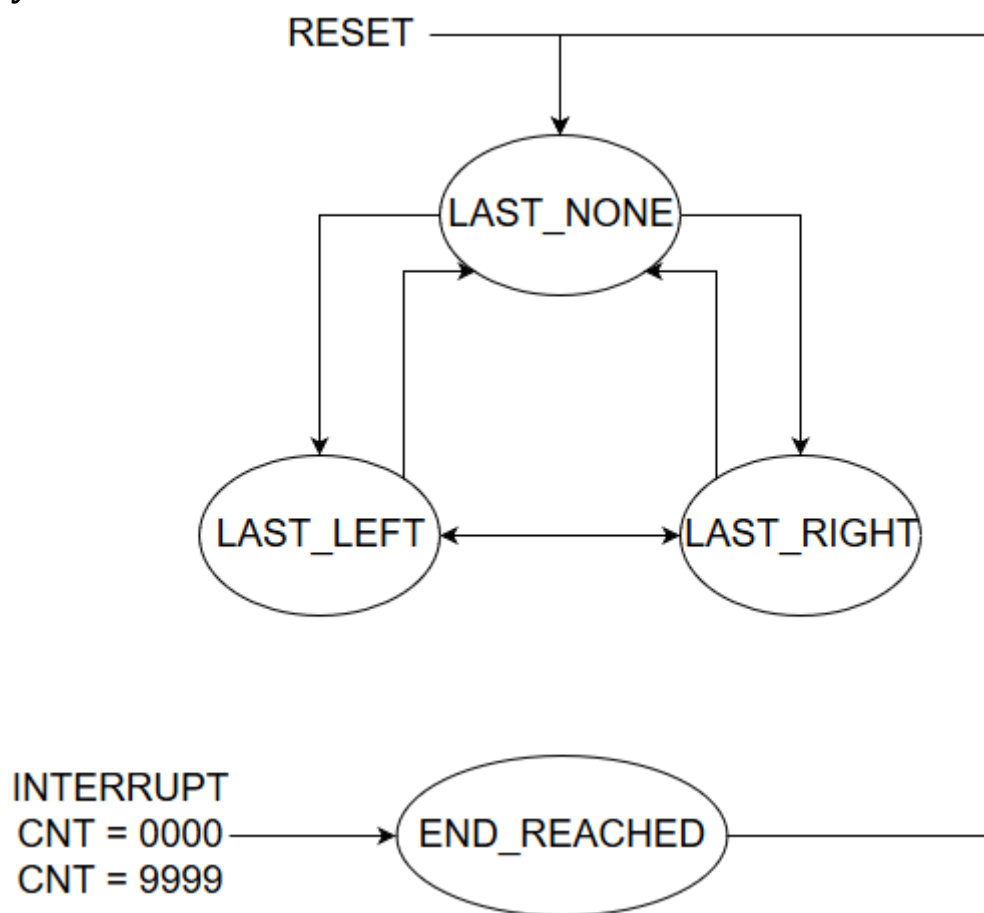
Další důležitou částí je obsluha přerušení. Ta řídí problikávání displeje při nastavování (Flash) a problikávání na konci odpočtu. Pokud probíhá čítání volá příslušnou funkci na změnu hodnoty čítače.

```
interrupt VectorNumber_Vrtc void RTC_interrupt(void) {
    Flash++;
    if (Last == END_REACHED)
        Problikávání displeje na konci odpočtu
    else {
        Kontrola konce odpočtu
        if ((InnerVec & 0xE0) == 0xC0)
            dec_cnt();
        else if ((InnerVec & 0xE0) == 0xE0)
            inc_cnt();
    }
    Nastavení dalšího přerušení
}
```

## **Analýza paměťových nároků aplikace**

READ_ONLY:	1288 B
READ_WRITE:	128 B
Velikost kódu:	253 řádků

## Přílohy



Obrázek 1 - Stavový diagram čítače