

APS - prednaska, M.D. 2018-11-07

Projekt pouzity na demonstraciu:

- vyuzitia prerusenania od periferie (casovaca)
- porovnanie ASM a C programu s prakticky identickou cinnostou

Projekt je mozne simulovat v prostredii ARM Keil C51 (<http://www.keil.com/c51/>), ktore je mozne stiahnut na stranke:

<https://www.keil.com/c51/demo/eval/c51.htm>

vo forme evaluacnej verzie, ktora plne postacuje pre ucely demonstracie. Podobne simulacne prostredie (<http://www2.keil.com/mdk5>) je mozne stiahnut aj pre 32-bitove MCU s jadrami ARM na stranke:

<https://www.keil.com/demo/eval/arm.htm>

----- Povodne read.me v predmete Mikroprocesorova technika -----

Mikroprocesorova Technika, 6. tyzden

Obsluha prerusenania v 8051, vyuzitie internej periferie citaca/casovaca

Ciel:

Princip ovladania periferie a vyuzitie prerusenania 8051 od internej periferie citaca/casovaca T0 v ASM aj C.

CASOVAC_ASM (asm) + Simulacia v simulatore

Kod demonstuje vyuzitie jednej zo zakladnych periferii 8051, casovaca T0 na generovanie definovaneho oneskorenia. Obsluha T0 (reprogramovanie definovaneho casoveho intervalu a ovladanie zvoleneho pinu) je realizovana v preruseni. Vyznam jednotlivych bitov v riadiacich registroch bol prebrany v ramci prednasok.

Zdrojove kody pouzite v ramci cvicenia demonstuju aj pouzitie niekterych novych direktiv assemblera (napr. #HIGH a #LOW), ktore realizuju vyber hornych resp. spodnych 8 bitov zo 16-bitoveho cisla.

ULOHY:

-
- ODSIMULUJTE A OVERTE SPRAVNOST PROGRAMU
 - VYSVETLITE, PRECO JE V PRERUSENI CASOVAC NAJSKOR ZASTAVENY A AZ POTOM MODIFIKOVANY
 - JE PERIODA GENEROVANEHO SIGNALU NA VYSTUPNOM PORTE PRESNE 1 MS?

CASOVAC_C + Simulacia v simulatore

Uvedenu cinnost je mozne realizovat aj v jazyku C. Na definovanie obsluzneho programu pre casovac T0 je pouzite klucove slovo "interrupt"

s cislom prislusneho prerusenania, ktore jednoznacne definuje vektorovu adresu prerusenania. Klucove slovo interrupt je dalsim z rozsireni jazyka C51, standardny jazyk ANSI C ho neobsahuje.

Klasicky procesor 8051 ma nasledujuce zdroje prerusenani:

Prerusenanie	Vektorova adresa	Cislo prerusenania
External 0	0003h	0
Timer 0	000Bh	1
External 1	0013h	2
Timer 1	001Bh	3
Serial	0023h	4

Novsie derivaty (ako napr. ADuC83x a pod.) obsahuju dalsie zdroje prerusenani, ktore maju vektory prerusenani umiestnene na vyssich adresach, pricom tieto adresy su od seba vzdialene vzdy o 8 bajtov a kazdemu zdroju prerusenania tak je v C51 priradene jednoznacne cislo prerusenania.

V kode je pouzite klucove slovo volatile, ktore je sucastou ANSI C:

```
volatile int pocitadlo;
```

Uvedena konstrukcia zabezpeci, ze C prekladac pri analyze programu main() bude predpokladat, ze premenna pocitadlo moze byt zmenena aj mimo tela programu (v nasom pripade v prerusenani). Bez tohto predpokladu by kompilator pri analyze programu main (a pokuse o optimalizaciu kodu) mohol predpokladat, ze podmienka testovania

```
if( pocitadlo == STOP_CASOVACA) {  
    ...  
}
```

nikdy nenastane (kedze v tele main() programu nie je pocitadlo po inicializacii na hodnotu 0 ziadnym sposobom modifikovane) a teda prislusnu cast kodu by mohol uplne eliminovat (vyhodit), co by sposobilo nespravnu funkcnost programu.

ULOHY:

- ODSIMULUJTE A OVERTE SPRAVNOST PROGRAMU
- PREZRITE SI KOD GENEROVANY PREKLADACOM A ZHODNODTE JEHO EFEKTIVNOST
- AKA JE MAXIMALNA PERIODA, KTORU MOZEME VYTVORIT PRI 12 MHz HODINOVOM KMITOCTE?

Domaca uloha:

Upravte kod tak, aby perioda bola s co navacsou presnostou rovna 1 sekunde. Pomocka: kedze casovac je len 16 bitovy, musite softverovo (napr. pomocou pocitadla v pomocnej premennej) zvacsit rozsah pocitania.

