**Čítače**

Čítač je zařízení, které slouží pro odměření přesných časových úseků. Náš procesor má celkem pět čítačů: TMR1 , TMR2 ,TMR3 ,TMR4 ,TMR5

TMR1 je nejsložitější, takže ho vynecháme. Pokud chcete, přečtěte si o něm v datasheetu

Zbylé čítače jsou rozděleny do dvou skupin. Čítače TMR2 a TMR4 tvoří čítače typu B , čítače TMR3 a TMR5 tvoří čítače typu C

**Čítače typu C**

TMR3 a TMR5

datasheet strana 176 , FIGURE 13-2:

datasheet je (již bylo mnohokrát řečeno, ale pro vaše pohodlí to zde uvádím ještě jednou ) <http://ozeas.sdb.cz/panska/mikroproc/33EV32/pdf/33EV32.pdf> Opravdu si ho otevřete, jinak je další výklad zcela nesrozumitelný.

Obrázek vypadá šíleně, ale neděste se. Úplně nalevo je vstup hodin TxCK. Ten můře být leccos, ale my budeme používat vnitřní oscilátor procesoru. Hodinový signál je roven frekvenci instrukcí, tedy to je 7.37 / 2 = 3.685MHz. Hodinový signál prochází děličkou (Prescaller), která může dělit 1 , nebo 8, nebo 64 , nebo 256 . Dále hodinový signál vchází do vlastního čítače ( TMRx , tedy buď TMR3 nebo TMR5 ). Čítač je šestnáctibitový. Čítač čítá a čítá. Úplně všechny čítače procesoru jsou čítače se zkráceným cyklem ( pokud nevíte, co to je, přestupte na gymnasium). Zkrácení je dáno číslem, které je uloženo v registru PRx ( tedy PR3 nebo PR5). Zkrácení funguje tak, že čítač čítá od 0 až do toho čísla, které je v registru PR. Toto číslo se v čítači ještě objeví. V další periodě hodin se čítač vynuluje.

Pokud nastavíme PR = 5 , pak čítač TMR čítá

0,1,2,3,4,5,0,1,2,3,4,5,0,1,2,3,4,5,0,1,2,3,4 ,5, ..........

Stav, kdy se čítač a registr periody rovnají ( TMR == PR ) se v datasheetu nazývá „match“ . Čítač budeme často používat jako děličku. Je zjevné, že čítač čítne o 1 více, než je číslo nastavené v registru PR. Proto jeho dělící poměr je o 1 větší než číslo v PR ( číslo 0 se samozřejmě také počítá )

Když čítač udělá „match“ , to znamená, že se vynuluje, oznámí to okolnímu světu tím, že nastaví bit TxIF do 1 ( úplně napravo nahoře, Set TxIF Flag). Bit se jmenuje buď T3IF nebo T5IF. Programátor tento bit testuje, a když zjistí, že má hodnotu 1 , ví, že uplynul čas, který jsme nastavili v čítači. Čítač samozřejmě čítá dál a dál a dál, udělá „match“ , nastaví bit IF, čítá , udělá „match“ , nastaví bit IF . **Bit IF musí nulovat programátor, čítač ho pouze nastavuje do 1 .**

**Ještě jednou: bit IF čítač nastaví do 1 . Nastavit do 0 bit IF musí programátor.**



Na obrázku výše máte zjednodušený obrázek z datasheetu.

Zcela zásadní je pro činnost čítače bit TxIF. Ten oznamuje okolí, že čítač udělal match, a tím pádem že uplynula nastavená doba. Bity TxIF jsou v registrech IFS0 a IFS1 . Podívejte se do datasheetu na stanu 55 , TABLE 4-23. Udělejte to, napište si, kde je který bit.

Překladač samozřejmě umí použít i písmenka, pomocí .equ k nim přiřadí čísla. Takže například instrukce

bset IFS0, #T3IF se přeloží jako

bset 0x0800, #8

Pokud uděláte chybu a napíšete

bclr IFS0, #T5IF ; překladač to přeloží jako

bclr 0x0800 , #12 **; tady se na to dívejte do té doby, dokud vám nedojde, co je na tom špatně a který bit jste pravděpodobně chtěli vynulovat** . Podívejte se do datasheetu na stranu 55, najděte registr s adresou 0x0800 , ukažte si na bit 12 , a dojde vám, co je špatně.

Před použitím musíme čítač naprogramovat. **Naprogramovat procesor znamená uložit řídící slovo do řídícího registru,** tedy uložit číslo do buňky. Samozřejmě, správné číslo do správné buňky .

Řídící registr pro náš čítač máme v datasheetu na straně 179 , REGISTER 13-2:. Díváme se na obrázek .

bit 15 – TON: zapíná a vypíná čítač. 1 – čítač čítá, 0 – čítač nečítá.

bit 13 – TSIDLE – určuje, zda má čítač čítat ve sleep módu proceroru. Tento mód nebudeme používat, takže je jedno, jakou hodnotu do tohoto bitu napíšeme, nejlépe 0

bit 6 – TGATE – umožňuje ovládat čítač dále ještě vnějším pinem procesoru – zapnout/vypnout. Zatím to nebudeme používat, takže 0

bity 5,4 – nastavení děličky – prescaller

bit1 – TCS – zdroj hodin. U nás vnitřní oscilátor, takže 0 .

Při konstrukci čísla pro registr TyCON postupujeme tak, že si na papír uděláme škatulky pro 16 bitů, vodorovně, tak, aby se výsledné binární číslo dalo snadno přečíst. Do škatulek pak napíšeme 1 nebo 0 podle toho, jak chceme čítač nastavit. Pokud je bit „Unimplemented“ , napíšeme 0 (je samozřejmě možno napsat cokoli, ale pro přehlednost je nejlepší 0). Binární číslo pak převedeme do hexadecimálního, aby se nám s ním snáze pracovalo. Pokud neumíte převody, přestupte na gymnasium.

Dále uvádíme zjednodušený výtah z datasheetu, kde jsou uvedeny bity, které budeme (téměř vždy) nastavovat. B znamená, že bit nastavíme na nějakou vhodnou hodnotu , 1 – nastavíme do 1, 0 – nastavíme do 0 , - je to jedno ( unimplemented ), ale doporučujeme 0.

**Registry T3CON a T5CON**



Takže: bit 15 do 1 , bit 6 do 0 , bit 1 do 0, bity 5 4 udávají dělící poměr děličky .

**Čítače typu B**

TMR2 a TMR5

datasheet strana 176 , FIGURE 13-2:

Tyto čítače jsou téměř stejné jako čítače typu C. V konfiguračním registru mají navíc bit T32, který umožňuje spojit dohromady čítač TMR2 a TMR3 ( nebo TMR4 a TMR5 ) a udělat z těchto dvou čítačů jeden obří 32bitový čítač. Pokud ovšem je bit T32 = 0, chová se čítač typu B stejně jako čítač typu C

Řídící registr máme na straně 178, REGISTER 13-1

Opět následuje výtah

**Registry T2CON a T4CON**



Pokud tedy chceme zapnout čítač do obyčejného 16-ti bitového módu, nastavíme

bit 15 do 1 , bit 6 do 0 , bit 3 do 0 , bit 1 do 0 , bity 54 určují děličku

**32-bitový mód**

V tomto případě „sežere“ čítač TMR2 čítač TMR3 (čítač TMR4 sežere čítač TMR 5) a nastavení celé dvojice se ovládá jenom registrem T2CON ( nebo T4CON). Čítače dávají dohromady obří čítač následovně



Vidíme, že TMR3 je vyšší polovina obřího čítače, TMR2 je nižší poloviny obřího čítače. Obdobně se skládají dohromady registry periody PR3 a PR2. **Ale POZOR ! „Match“ čítače nastaví bit T3IF .**

**Příklad**

Nastavíme čítač TMR2 tak, aby generoval čas 56ms ( tedy aby vždy po 52ms nastavil bit T2IF do 1)

Podíváme se na obrázek na straně 2. Vidíme, že do čítače vstupuje kmitočet Fosc 7.37MHz/2, tedy 3,685MHz. Na druhé straně ( u bitu T2IF) má vycházet čas 56ms, to odpovídá kmitočtu 17,85Hz (pokud nevíte, jak se z času udělá kmitočet, gymnázium!). Celý blok čítač TMR2 a prescaller tedy musí zajistit dělení číslem 3.685MHz / 17,85 = 206360 . Tohle číslo bychom měli vložit do registru PR2 , ale je moc velké ( jaké je maximální možné číslo ?) Nezbývá tedy než nastavit prescaller. Součin hodnoty, kterou dělí prescaler, a hodnoty, kterou dělí čítač TMR2 , musí dát 206360 . Tak to zkusíme:

Prescaller dělí 8 , na TMR2 zbývá 25795 . Do registru PR2 uložíme číslo o 1 menší, tedy 25798

Na nastavení se podívejte do programu citac01.s .

Pustíme si simulátor, dále PAUSE a RESET. Mačkáme F7 a díváme se , co všechno program dělá. Po nějaké době program začne procházet cyklem od návěští cyklus . Nyní si dáme breakpoint na instrukci call odTMR2 a zmáčkneme F5 . Program běží a běží, až se kousne na breakpointu. Pustíme si STOPWATCH a díváme se, jaký čas uplynul mezi jednotlivými voláními. Nezapomeneme nastavit správně v simulátoru periodu instrukcí – v okně SPOPWATCH vpravo dole, 1MHz je špatně. U verse MPLABX 5.10 se po nastavení frekvence hodin někdy číslo neobjeví v okně stopwatch, pomůže celý MPLABX vypnout a opět zapnout.

Zmáčkneme několikrát F7 , díváme se, jak program vstupuje do podprogramu. Potom opět mačkáme F5 , program se zastaví na breakpointu. Mačkáme F8 – podprogram se udělá, jako by to byla jedna instrukce. Několikrát zopakujeme, až nám bude jasné, jaký je rozdíl mezi F7 a F8 .

Potom zrušíme breakpoint u call odTMR2 a nastavíme breakpoint na call odTMR3 . Opět se podíváme, jak dlouhý čas čítač generuje, a porovnáme s naším výpočtem.

A totéž ještě jednou s call odTMR4

Neustávejte, dokud vám nebude úplně jasné, co se děje.

Dopište si čítač TMR5 podle svého uvážení a zkuste generovat nějaký čas.

Samozřejmě, do podprogramů odTMR2 , odTMR3 , odTMR4 si můžete dopsat nějakou akci, která se má dělat vždy po zvoleném čase – blikání s diodou, generování tónu, .... , .... ,

Pokud netušíte, o čem se mluvilo v minulých odstavcích, zřejmě neumíte ovládat simulátor. To jsme samozřejmě dělali už mnohokrát, nicméně si to přečtěte zde:

všeobecný návod na simulátor je v

<http://ozeas.sdb.cz/panska/mikroproc/33EV32/MPLABX/pouziti_MPLABX_16F.docx>

od strany 12 , je to pro jiný procesor, ale ovládání simulátoru je stejné

stopwatch

<http://ozeas.sdb.cz/panska/mikroproc/33EV32/MPLABX/stopwatch_33EV32.docx>

A celé se to naučte. Na gymnasiu byste mohli v zeměpise úplně vypustit Severní Ameriku a naučit se jenom Jižní. Z prvního byste dostali 5 , z druhého 1. a výsledek by byl 3 . Nebo byste se nějak vyhnuli zkoušení a vůbec nikdo by po celou zbývající dobu studia nepoznal, že jste Severní Ameriku úplně vypustili. Tak tomu není v našem předmětu. Potřebujete neustále znát úplně všechno, co jsme dosud probrali. Pokud máte nějaké mezery, doučte se, vše máte napsané na Ozeášovi.