**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 chete, 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:

Obrázek vypadá šíleně, ale neděste se. Úplně nalevo je vstup hodin. 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, proto )

Když čítač udělá „match“ , to znamená, že se vynuluje, oznámí to okolnímu světu tím, že nastaví bit TxIF do 1 ( úplně napravo). 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 .**



Na obrázku výše máte zjednodušený obrázek z datasheetu.

Zcela zásadní je pro činnsot 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 písm,enka, 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**

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:. Přečtěte si to tam. 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 ).

**Registry T3CON a T5CON**



**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 těcchto dvou čítačů jeden obří 32bitový čítač. Pokud ovšem je bit T32 = 0, chová se čítač typu B stejně jako čítač typu B

Řídící registr máme na straně 178, REGISTER 13-1

Opět následuje výtah

**Registry T2CON a T4CON**



**32-bitový mód**

V tomto případě „sežere“ čítač TMR2 čítač TMR3 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 druhá straně ( u bitu T2IF) má vzcházet 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 bzchom 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 hodnotz, 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 .

Do podprogramů si dejte breakpoint ( vždy jenom jeden breakpoint do jednoho podprogramu) , pusťte si stopwatch, mačkejte F5 a dívejte se, co to dělá. Také zkuste mačkat F7. Neustávejte, dokud vám nebude úplně jasné, co se děje.