Ve file PPS.asm je program pro demonstraci PPS. Piny procesoru jsou nastaveny následovně:

* Výstup modulu PWM3 bude na pinu 12 – RB5
* Vstup pro hodiny čítače TMR0 bude na pinu 7 – RC3

Budeme generovat PWM s dobou periody 12 ms. Velikost „Duty cycle“ bude 3,234 milisekund, pokud možno co nejpřesněji.

Použijeme čítač TMR4

vnitřní oscilátor bude mít kmitočet 4MHz – kmitočet instrukcí 1MHz

Na začátku konfigurace nastavíme pacičky procesoru pomocí PPS. Výstup PWM má být na pinu 12. Tento pin ovládá registr RB5PPS (viz soubor PPS.doc). K němu najdeme periferii – PWM3. Ta odpovídá číslu 01110 , viz strana 6 ( čísla stran podle file PPS.doc, v datasheetu procesoru je to od strany 136 ) . Toto číslo tedy uložíme do registru RB5PPS.

Dále máme nastavit vnější vstup hodin pro čítač TMR0 . U vstupů patří ovládací registr k periferii. V tabulce na straně 3 najdeme registr pro čítač T0CKIPPS . K němu najdeme v tabulce na straně 2 číslo pro vstupní pin 7- RC3, to je 0x13. Toto číslo uložíme do registru T0CKIPPS . Nezapomeneme přidal „lock“ a „unlock“ sekvenci.

**Vlastní nastavení PWM**

Nejprve určíme parametry čítače TMR4

12ms periody (zadáno, viz výše) odpovídá kmitočtu 83.333 Hz

1MHz tedy musíme vydělit číslem 12000 , abychom dostali tento kmitočet. (jiná možnost: perioda instrukcí je 1 mikrosekunda, my potřebujeme periodu 12 milisekund, tedy periodu musíme násobit 12000)

Dělící poměr 12000 se skládá z děličky a z čísla v registru PR4

zkusíme dát děličku na 64

12000 / 64 = 187.5

to je dost nepříjemné, jsme právě uprostřed. Zaokrouhlíme nahoru na 189. Toto číslo dáme do registru PR4. Dělička bude dělit 64 – registr T4CON.

Přesný kmioočet PWM bude:

1000000 / 64 / 189 = 81.4 Hz, to odpovídá periodě 12.285 ms.

**Číslo pro „Duty cycle“.**

Popužijeme přímou úměru.

V registru PR4 je 189 , odpovídá 12.285 ms

V registru PWMDC je ???? , odpovídá 3,234 ms

Do registru PWM3DC uložíme číslo tedy 49,754 (pokud neumíte přímou úměru, pak vizte, že i na gymnáziu se její znalost předpokládá jako samozřejmost)

Číslo 49.754 je rozděleno na dvě části. V registru PWM3DCH bude 49 , desetinná část je v PWM3DCL. Nejvyšší bit zde má váhu 0.5 , ten další 0.25 ( 2-1 a 2-2 ) . Nejbližší možné číslo je tedy 11 binárně, odpovídá 0.75 desítkově. Toto číslo uložíme na bity 7 a 6 registru PWM3DCL

Do přerušení od TMR4 si dáme breakpoint. Pomocí STOPWATCH si změříme dobu od jednoho přerušení k druhému. Nezapomeneme, že tato doba je čtyřnásobkem periody PWM (postscaller u TMR4) Důrazně připomínáme, že nastavení postscalleru nemá vliv na periodu PWM. Ta je určena prescallerem a čístem v registru PR4. Postscaller pouze prodlouží dobu pro zavolání přerušení od TMR4, takže na jedno přerušení se vygenerují 4 periody PWM .

**U MPLAB XIDE verse 3.15 i verse 3.20 nefunguje simulace přiřazení čítačů ke generátorům PWM pomocí registru CCPTMRS . Ať už nastavíte do tohoto registru jakékoli číslo, PWM bude vždy odvozeno od čítače TMR2 . Tohle platí pro simulátor, překladač kód přeloží správně.**

V simulátoru si s tím poradíme jednoduše. Buď máme čítač TMR2 puštěný a tento čítač běží, v tom případě bude simulátor PWM simulovat. Musíme ovšem počítat s tím, že doba periody je jiná, než bude potom ve výsledném \*.hex file . Nebo TMR2 nepotřebujeme a máme ho vypnutý. V tom případě ve Variables ručně zapíšeme do T2CON a PR2 stejné číslo, jako máme u čítače TMR4 nebo TMR6. Tím pro účely simulace pustíme čítač TMR2 a PWM bude fungovat.

Pokud se vám v Logic Analyzer nezobrazují impulzy správně, zkuste změnit Logic Analyzer settings – Buffer size . Dále je užitečný signál PWM3OUT v Channel definitions. To je vnitřní výstup PWM, a měl by fungovat, i když se signál nepodaří díky chybnému nastavení propojit až na výstupní pacičku.

Příklad PPS.asm funguje s následujícím nastavením:

Buffer size 100000 records

Project Properties – Similator – Trace – Data buffer maximum size 5460000

V Analyzer nastaveny dva kanály – RB5, PWM3OUT

**Simulace vstupu TMR0**

Čítač TMR0 máme naprogramovaný na vnější vstup hodinového signálu, ten je na pinu 7 –RC3. Zapneme si stimulus. Do Asynchronous – Pin se nastavíme RC3, jako Action dáme tentokrát Toggle. To znamená, že při cvaknutí myší na Fire se pin zneguje. No a teď už jenom cvakáme na Fire a díváme se do Variables na registr TMR0, jak čítač krásně čítá. !!!! Mezi jednotlivými „Fire“ musí uplynout alespoň jeden instrukční cyklus, takže mezi nimi musíte alespoň jednou udělat F7. Dále nezapomeňte, že čítač čítá jenom na vzestupnou hranu ( viz nastavení registru OPTION\_REG, strana 244 datasheetu 16F1708.pdf ).