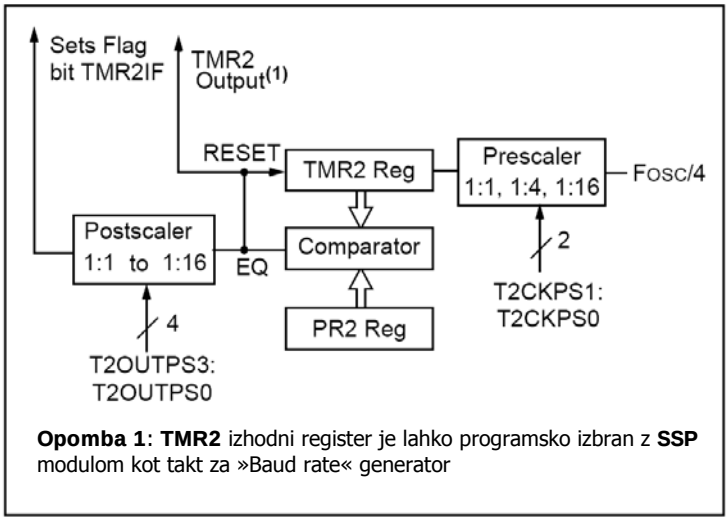
7. Generiranje pulzno-širinskih signalov (PWM)

Spoznali boste vmesnik mikrokrmilnika, ki omogoča generiranje pulznoširinskih signalov (PWM). Naučili se boste konfigurirati registre za izbiro periode (frekvence) in širine pulza vmesnikov PWM1 in PWM2.

Modul **Timer2** je 8-bitni časovnik/števec z vgrajenim preddelilnikom (Prescaler) in podelilnikom (Postscaler). Lahko se **uporablja tudi kot PWM časovna baza za generiranje dveh PWM signalov (z enako periodo) v okviru CCP podsklopa**.

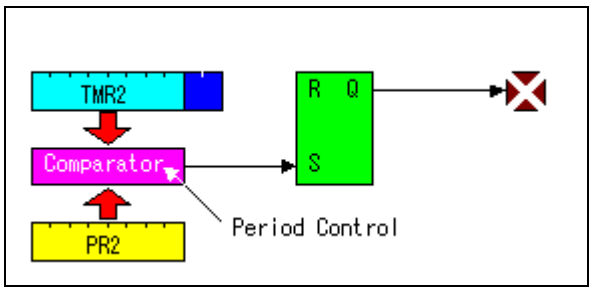


Slika 7-1: Zgradba modula Timer2

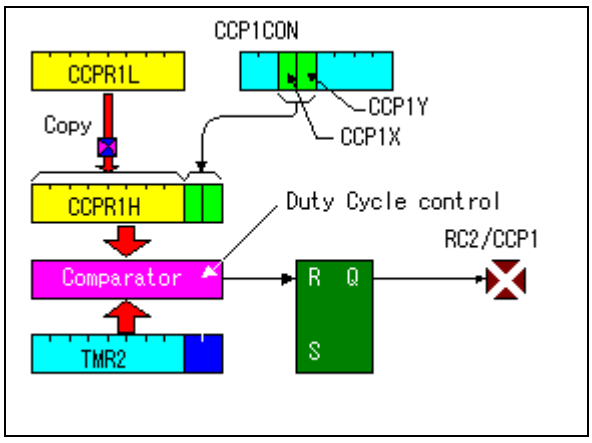
7.1. Zgradba in delovanje PWM vmesnika

CAPTURE/COMPARE/PWM (**CCP1** in **CCP2**) je podsklop modula **Timer2**, ki vsebuje 16-bitne registre in omogoča delovanje v naslednjih načinih:

- 16-bitni zajemalni (Capture) način;
- 16-bitni primerjalni (Compare) način;
- **PWM register za nastavitev širine pulza** (Master/Slave Duty cycle).

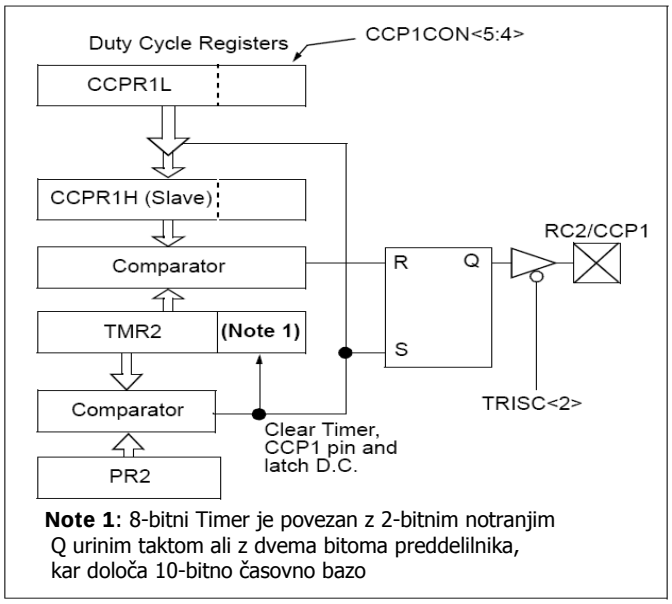


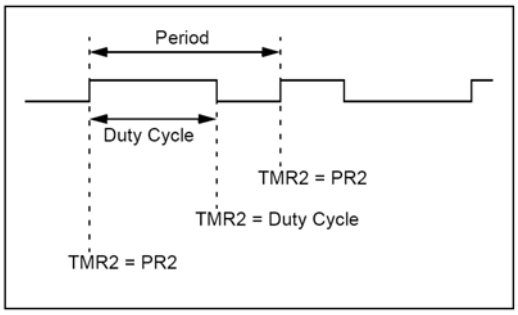
Slika 7-2: Nastavitev periode



Slika 7-3: Nastavitev širine pulza

PWM način delovanja omogoča, da na izhodih **CCP1** in **CCP2** programsko generiramo pulznoširinska signala z nastavljivo (največ 10-bitno) širino pulzov (angl.: Duty cycle) pri vnaprej izbrani frekvenci (le-ta je enaka za oba sklopa).





*Slika 7-5: **PWM** izhodni signal*

*Slika 7-4: Poenostavljena shema **PWM1** sklopa*

PWM1 izhod je na priključku **RC2/CCP1.**

PWM2 izhod je na priključku **RC1/CCP2.**

Periodo PWM signala nastavimo z vpisom 8-bitne vrednosti v register **PR2**. Izračunamo jo po naslednji formuli:

$PWM_{Perioda}$ = (**PR2**+1) * 4 * T_{osc} * $TMR2_{Prescaler}$, pri čemer je T_{osc} = 50 ns, če je frekvenca urinega takta 20 MHz.

Frekvenco (obratna vrednost $PWM_{perioda}$) lahko izbiramo od 1,22 kHz do nekaj 100 kHz (spodnja tabela), pri čemer se zmanjša ločljivost nastavitve širine pulza pod 10-bitov, če je frekvenca večja od 20 kHz.

PWM frekvenca	1.22 kHz	4.88 kHz	19.53 kHz	78.12kHz	156.3 kHz	208.3 kHz
Vrednost preddelilnika (1, 4, 16)	16	4	1	1	1	1
PR2 vrednost	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
Največja ločljivost (biti)	10	10	10	8	7	5.5

Tabela 7-1: Vrednosti ločljivosti pri različnih frekvencah in periodah

Širino pulza (Duty cycle) nastavimo z vpisom višjih 8-bitov v register CCPR1L in spodnjih dveh bitov 10-bitne vrednosti v register **CCP1CON** (bita 5 in 4) v skladu z naslednjo formulo:

$PWM_{Duty\ cycle}$ = (**CCPR1L:CCP1CON<**5:4**>**) * T_{osc} * $TMR2_{Prescaler}$

Za čim večjo ločljivost se priporoča upoštevanje vrednosti **PR2**, kot jih prikazuje Tabela 7-1!

7.2. Opis registrov in primer programa

7.2.1. Opis registrov CCP1CON oz. CCP2CON

Nahaja se na naslovih 17h oz. 1Dh.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	**-**	**CCPxX**	**CCPxY**	**CCPxM3**	**CCPxM2**	**CCPxM1**	**CCPxM0**
bit7	6	5	4	3	2	1	bit0

Legenda: R = omogočeno branje bita, W = omogočeno pisanje v bit, U = neuporabljen bit, beri kot '0', -n = vrednost ob POR resetu

bita 7-6: neuporabljena

bita 5-4: **CCPxX: CCPxY**: **spodnja dva bita 10-bitne vrednosti širine pulza PWM sklopa**

biti 3-0: **CCPxM3- CCPxM0**: izbira načina delovanja CCPx sklopa, za **PWM način: 11xx**

7.2.2. Opis registra T2CON

Nahaja se na naslovu 12h.

u-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	**TOUTPS3**	**TOUTPS2**	**TOUTPS1**	**TOUTPS0**	**TMR2ON**	**T2CKPS1**	**T2CKPS0**
bit7	6	5	4	3	2	1	bit0

Legenda: R = omogočeno branje bita, W = omogočeno pisanje v bit, U = neuporabljen bit, beri kot '0', -n = vrednost ob POR resetu

bit 7: neuporabljen

biti 6-3: **TOUTPS3: TOUTPS0**: delilnik frekvence za izhod Timer2 (od 0000 - 1:1 do 1111 - 1:16) – **nima vpliva na PWM**

biti 2: **TMR2ON**: **omogočitev Timer2** (**1 - Timer2 aktiven**, 0 - neaktiven)

bita 1-0: **T2CKPS1- T2CKPS0**: **izbira preddelilnika** za Timer2 v **PWM načinu: 00 – 1x,** 01 – 4x, 10 ali 11 – 16x

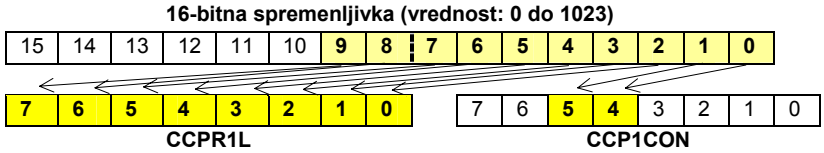
Postopek konfiguriranja CCP modula za PWM način delovanja:

1. Nastavitev PWM periode (frekvence) z vpisom **8-bitne vrednosti** v register **PR2**;
2. Nastavitev PWM širine pulza z vpisom višjih 8-bitov v register **CCPR1L** in spodnjih dveh bitov **10-bitne vrednosti** v register **CCP1CON** (bita 5 in 4)
3. Nastavitev **CCP1 priključka kot izhod** z brisanjem bita 2 (RC2=0) v registru **TRISC**;
4. Nastavitev **TMR2** preddelilnika in omogočitev Timer2 z vpisom ustrezne vrednosti (npr.: 04h) v register **T2CON**;
5. Konfiguriranje CCP1 modula za PWM način delovanja z vpisom (npr.: 0Ch) v register **CCP1CON**;
6. Ponavljanje (izvajanje) točke 2, ko želimo spremeniti širino pulza.

Navedeni postopek velja za **PWM1** sklop, ki ima izhod na priključku **CCP1/RC2**. Če želimo uporabiti tudi **PWM2** sklop, dobimo izhodni signal na priključku **CCP2/RC1**, registra v točki 2 in 5 (oz. formuli) pa nadomestimo z **CCPR2L** in **CCP2CON**.

V kolikor smo zadovoljni z **8-bitno ločljivostjo**, je dovolj da 8-bitno vrednost (med 0 in 255) prenesemo v register **CCPR1L**, medtem ko bita 5 in 4 v registru CCP1CON pustimo na izhodiščni vrednosti (0).

Če želimo generirati PWM signal s polno **10-bitno ločljivostjo** (med 0 in 1023), naložimo vrednost iz 16-bitne spremenljivke **v oba registra po postopku, kot je to prikazano** (Slika 7-6)**:**



Slika 7-6: Vrednost 16-bitne spremenljivke shrani v CCPR1L in CCP1CON

Primer programa v C-jeziku za generiranje signalov na izhodih PWM1 in PWM2:

```
// Demo program za prenos 10-bitne vred. na PWM1 in PWM2 (LD2 na RC1)
// Vrednost povecuj za 1 kv. (~5mV) ob vsakem pritisku na tipke

#include <pic1687x.h> // vkljucitev datoteke z definicijami simbolov
unsigned int Vredn;  // sprem. za vrednost PWM: 10-bit 0-1023

void main(void)
{
unsigned i;    // stevec za zakasnitev
//--------- Inicializacija registrov vhodno/izhodnih vmesnikov
TRISC=0;       // vsi pini porta C so izhodni
TRISA=0x10;    // vsi pini porta A so izhodni, razen RA4
INTCON=0;      // prekinitev ne bomo uporabljali
PORTC=0;       // izhode porta C postavi na 0
PORTA=0;       // izhode porta A postavi na 0

//--- inicializacija PWM1 ali/in PWM2
PR2=249;       // Perioda=(249+1)*4*50ns*1=50us ali 20kHZ pri 20Mhz takt, preddelilnik=1
CCPR1L=0;      // zacetna vrednost Duty cikla=0, moznost nastav: 0 do 1023 po 50ns
T2CON=0x04;    // bit 2: TMR2ON=1, aktiviraj Timer2, preddel.=1, b'00000100'
CCP1CON=0X0C; // za Timer 2 izberemo PWM nacin bit3=1, bit2=1, b'00001100'

Vredn=0;       // zacetna vrednost PWM1
// -------- jedro programa
       while(1)
       {
       PORTC=PORTC|0xF0;     // postavi RC7 do RC4 za test tipk
         if(RA4==1)          // ali je pritisnjena katerakoli tipka ?
         {
         Vredn=Vredn+1;      // DA, povecaj vrednost za 1 kvant
         RC0=1;              // vklopi LD1
  // ---- prenesi 10-bitno vrednost iz 16-bitne sprem. Vredn v PWM1 in PWM2 registre
         CCPR2L=CCPR1L=(unsigned char)(Vredn>>2); //zgornjih 8 bitov -> CCPR1L in CCPR2L
         CCP2CON=CCP1CON=(CCP1CON&0xCF)|((unsigned char)(Vredn<<4)); //spodnja 2 bita ->
       // CCPxCON<5,4>
         }
         else
         RC0=0;              // NE, izklopi LD1
       PORTC=PORTC&0x0f;           // brisi RC7 do RC4 za test tipk
       for(i=0; i<3333; i++);      // zakasnitev: 3333*3,0us=~10ms
       }
}
```

Program 7-1: Popoln primer programa v C-jeziku za vmesnik PWM

Za razumevanje tematike je potrebno tudi predznanje iz osnov programiranja in številskih sistemov. Dodatna znanja iz programiranja v zbirnem jeziku so v: [2][4][6][7]. Za razumevanje programov, ki so kodirani v C-jeziku, je potrebno poznavanje osnov ANSI C strukturnega programiranja: [15][11][13][5]. Veliko rešitev praktičnih nalog z uporabo PIC mikrokrmilnikov je na spletnem portalu http://www.interq.or.jp/japan/se-inoue/e_pic6.htm. Podrobnejši opis delovanja časovnika Timer2 in PWM vmesnika s primeri uporabe v PIC mikrokrmilnikih je v: [16][17].

Vprašanja za utrjevanje:

Napišite podprogram iz imenom InPWM1 za inicializacijo izhoda PWM1 s frekvenco 2,00 kHz.

Rešitev:

```
InPWM1 bsf    STATUS,RP0     ; 1 -> RP0 , izbira Bank1 za dostop do TRISC in PR2
       bcf    TRISC,2        ; RC2/CCP1 port za PWM1 definiramo kot izhod
       movlw  d'155'         ; desetisko 155 -> Za nastavitev frekvence signala 2kHz
       movwf  PR2            ; Perioda=(155+1)0.2us*16 = 499,2us -> f=1/499,2 = 2,003KHz
       bcf    STATUS,RP0     ; Nazaj v banko0
       clrf   CCPR1L         ; zacetna vrednost sirine pulza: CCPR1L = 0
       bsf    CCP1CON,3      ; 1 -> CCP1CON.3
       bsf    CCP1CON,2      ; 1 -> CCP1CON.2, bit3 in 2 na 1 v CCP1CON-PWM nacin delovanja
       movlw  b'00000111'    ; vrednost za T2CON, bit1,0 -> 1, preddelilnik je 1:16
       movwf  T2CON          ; 1 -> bit2, start casovnika TMR2
       return                ; konec podprograma
```

Napišite podprogram z imenom InPWM_2 za inicializacijo časovnika izhoda PWM2 s frekvenco 5,00 kHz.

1. *Koliko bitno vrednost lahko vpišemo kot vrednost za širino pulza ?*
2. *Kakšne vrednosti preddelilnika lahko izbiramo ?*
3. *Kakšna je ločljivost nastavitve širine pulza (najkrajša in najdaljša širina pulza v ns) pri frekvenci 20 kHz ?*
4. *Ali PWM vmesnik proži prekinitve ?*
5. *Kakšna je najmanjša širina pulza (razen 0) v zgornjem (rešenem) primeru ?*