Digitalno procesiranje v mehatroniki 1

Vaja 5 – Uporaba pulzno-širinske modulacije (PWM)

Napišite program, ki bo spreminjal vklopno razmerje tranzistorja pretvornika navzdol in s tem krmilil vrtljaje enosmernega motorja. Spreminjanje vklopnega razmerja izvedite s pomočjo PWM modula mikroprocesorja. Pri tem uporabite tipko PORTB,0 za start in tipko PORTA,4 za stop vrtenja motorja ter s potenciometrom AN0 nastavljajte vklopno razmerje (od 0% do 100%) in posledično hitrost vrtenja motorja.

Potek vaje:

1. Na začetku programa opravimo inicializacijo uporabljenih vhodov in izhodov, izbor PWM načina delovanja mikroprocesorja ter nastavitev delovanja A/D pretvorbe.

   - Nožica RB0 (start tipka) bo logični vhod (v registru **TRISB** postavimo ničti bit na ena)
   - Nožica RA4 (stop tipka) bo logični vhod (v registru **TRISA** postavimo četrti bit na ena)
   - Določimo začetno stanje izhoda RB1 (npr.: v **LATB** registru prvi bit postavimo na vrednost nič)
   - Nožica RB1 (LED indikator delovanja PWM modula) bo logični izhod (v registru **TRISB** postavimo prvi bit na nič)

   - Določimo začetno stanje izhoda RC2 (npr.: v **LATC** registru drugi bit postavimo na vrednost nič)
   - Nožica RC2 (PWM izhod) bo logični izhod (v registru **TRISC** postavimo drugi bit na nič)
   - Določimo PWM režim delovanja (biti <3:0> **CCP1CON** register)
   - Nastavimo pripadajoči časovnik (Timer 2) PWM modula – izberemo ustrezni delilnik frekvence (Prescaler in Postscaler)
   - Postavimo vrednost števca časovnika (register **TMR2**) na nič
   - Postavimo register **PR2** (vrednost PWM periode) na predhodno preračunano vrednost, ki odraža izbrano frekvenco 10kHz pulzno-širinske modulacije

   - Nožica RA0 (potenciometer) bo logični vhod (v registru **TRISA** postavimo ničti bit na ena)
   - Dodelitev kanalu AN0 funkcijo analognega vhoda v mikroprocesor. To opravimo z vpisom ustreznih vrednosti bitov v register **ADCON1**
   - Izbor kanala, na katerem želimo izvesti A/D pretvorbo (**ADCON0**)
   - Izbira takta za izvedbo A/D pretvorbe (**ADCON0**)
   - Vklop A/D modula (**ADCON0**)

   - Start A/D pretvorbe s postavitvijo bita GO/DONE na logično 1 – za podrobnejši opis A/D pretvorbe glej Vajo 2!

2. Glavni program naj se izvaja v obliki neskončne zanke.

3. V podprogramu naj se izvaja preverjanje končanja A/D pretvorbe in osvežitev spremenljivk vklopnega časa. Dalje naj se izvaja preverjanje stanja tipk start in stop ter glede na njuni stanji omogoči oz. ustavi delovanje PWM. V primeru delovanja PWM naj se ob osveženi A/D pretvorbi spremeni vklopno razmerje pulzno-širinske modulacije.

Pri izvedbi vaje lahko uporabite naslednje ukaze:

• **MOVLW**	**k**	Naloži konstanto *k* v delovni register.
• **MOVF**	**f,d**	Naloži vrednost spremenljivke *f* v delovni register (*d*=W) ali v spremenljivko *f* (*d*=F).
• **MOVFF**	**fs,fd**	Naloži vrednost spremenljivke *fs* v spremenljivko *fd.*
• **MOVWF**	**f**	Shrani vrednost iz delovnega registra v spremenljivko *f.*
• **BCF**	**f,b**	Postavi *b*-ti bit v registru *f* na nič.
• **BSF**	**f,b**	Postavi *b*-ti bit v registru *f* na ena.
• **BTG**	**f,b**	Spremeni logično stanje bita *b* v registru *f*.
• **BTFSC**	**f,b**	Preskoči naslednji ukaz, če je bit *b* v registru *f* postavljen na nič.
• **BTFSS**	**f,b**	Preskoči naslednji ukaz, če je bit *b* v registru *f* postavljen na 1.
• **BRA**	**n**	Brezpogojno skoči z izvajanjem programa na mesto labele *n*.
• **CLRF**	**f**	Postavi vrednost spremenljivke *f* na nič.
• **CALL**	**n**	Klicanje podprograma z oznako *n*.
• **RETURN**		Konec podprograma in vrnitev v glavni program.

14.0 CAPTURE/COMPARE/PWM (CCP) MODULES

Each CCP (Capture/Compare/PWM) module contains a 16-bit register which can operate as a 16-bit Capture register, as a 16-bit Compare register or as a PWM Master/Slave Duty Cycle register. Table 14-1 shows the timer resources of the CCP Module modes.

The operation of CCP1 is identical to that of CCP2, with the exception of the special event trigger. Therefore, operation of a CCP module in the following sections is described with respect to CCP1.

Table 14-2 shows the interaction of the CCP modules.

REGISTER 14-1: CCP1CON REGISTER/CCP2CON REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DCxB1:DCxB0**: PWM Duty Cycle bit1 and bit0

Capture mode:
Unused

Compare mode:
Unused

PWM mode:
These bits are the two LSbs (bit1 and bit0) of the 10-bit PWM duty cycle. The upper eight bits (DCx9:DCx2) of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0**: CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)
0001 = Reserved
0010 = Compare mode, toggle output on match (CCPxIF bit is set)
0011 = Reserved
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode,
Initialize CCP pin Low, on compare match force CCP pin High (CCPIF bit is set)
1001 = Compare mode,
Initialize CCP pin High, on compare match force CCP pin Low (CCPIF bit is set)
1010 = Compare mode,
Generate software interrupt on compare match (CCPIF bit is set, CCP pin is unaffected)
1011 = Compare mode,
Trigger special event (CCPIF bit is set)
11xx = PWM mode

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

14.5 PWM Mode

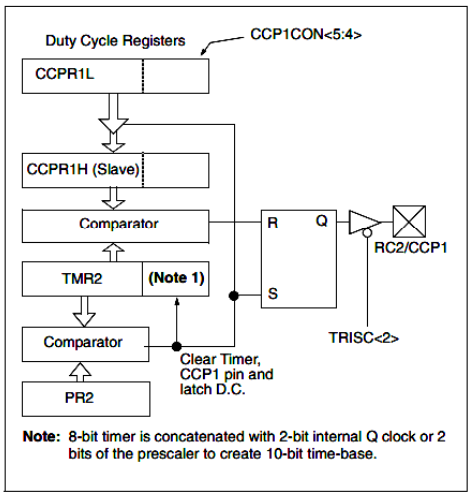
In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

Note:	Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 14-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 14.5.3.

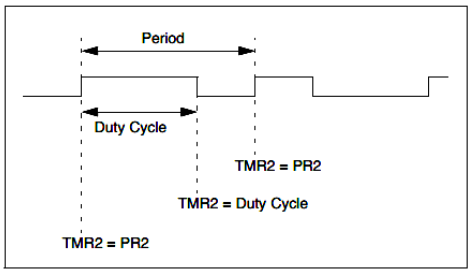
FIGURE 14-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time-base.

A PWM output (Figure 14-4) has a time-base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 14-4: PWM OUTPUT



14.5.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.

14.5.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$\text{PWM period} = (PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 prescale value})$$

PWM frequency is defined as 1 / [PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

Note:	The Timer2 postscaler (see Section 12.0) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

14.5.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

REGISTER 12-1: T2CON: TIMER2 CONTROL REGISTER

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
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown