

PRAKTIKUM ZA LABORATORIJSKE VJEŽBE IZ MIKROPROCESORSKIH SISTEMA

Prof. dr Slobodan Lubura

Nikola Kukrić

Verzija 0.001

Contents

1. Uvod	4
2. Laboratijske vježbe.....	4
2.1. Povezivanje LCD display-a sa PIC16F887	4
2.2. Povezivanje KeyPad-a sa PIC16F887	6
2.3. Multipleksiranje 7Seg display-a.....	8
2.4. Konfiguracija timer0-a kao brojača.....	10
2.5. Konfiguracija Timer0-a kao timer u interrupt modu.....	12
2.6. Konfiguracija Timer1-a kao timer u interrupt modu.....	14
2.7. Mjerenje perioda sa CCP modulom	19
2.8. CCP – mjerenje faktora ispune.....	26
3. Pitanja i zadaci.....	32

1. Uvod

2. Laboratijske vježbe

2.1. Povezivanje LCD display-a sa PIC16F887

Uvod

PIC mikrokontroler može jednostavno upravljati LCD modulom koristeći naredbe u LCD biblioteci mikroC PRO. Podatke za prikaz na LCD modulu PIC mikrokontroler može slati preko 4 ili 8 linija, pa govorimo o 4-bitnom ili 8-bitnom prenosu podataka. Osnovna razlika između 4-bitnog ili 8-bitnog povezivanja je u brzini prenosa podataka koji se šalju na LCD modul. Kod 8-bitnog prenosa, 8-bitni ASCII podatak se u jednom ciklusu šalje preko linija DB0- DB7 ka LCD modulu, dok kod 4-bitnog prenosa 8-bitni ASCII podatak se pri slanju dijeli na dva dijela, te se preko linija DB4 – DB7 se šalje u dva ciklusa. Pored pomenutih linija za podatke pri povezivanju PIC mikrokontroler sa LCD modulom potrebne su još tri dodatne linije E – početak upisa/čitavanja podatka, R/W – izbor upis/čitanje i RS – izbor registra: instrukcija/podatak. Osnovna ideja postojanja 4-bitnog povezivanja je smanjeni broj linija između mikrokontrolera i LCD modula

LCD Library mikroC PRO

LCD biblioteka mikroC PRO podržava komunikaciju sa HD44780 (Hitachi) kompatibilnim LCD modulima preko 4 linije za podatke i tri dodatne linije: E, R/W i RS za kontrolu komunikacije. Prije samog korišćenja naredbi iz LCD biblioteke mikroC PRO moraju biti definisane

```
// Lcd pinout settings
sbit LCD_RS at RD4_bit;
sbit LCD_EN at RD5_bit;
sbit LCD_D7 at RD3_bit;
sbit LCD_D6 at RD2_bit;
sbit LCD_D5 at RD1_bit;
sbit LCD_D4 at RD0_bit;

// Pin direction
sbit LCD_RS_Direction at TRISD4_bit;
sbit LCD_EN_Direction at TRISD5_bit;
sbit LCD_D7_Direction at TRISD3_bit;
sbit LCD_D6_Direction at TRISD2_bit;
sbit LCD_D5_Direction at TRISD1_bit;
sbit LCD_D4_Direction at TRISD0_bit;
```

promjenljive kojima se konfigurira željeni port PIC mikrokontrolera čije linije se koriste za povezivanje sa LCD modulom. Specifikacija promjenljivih data je u tabeli:

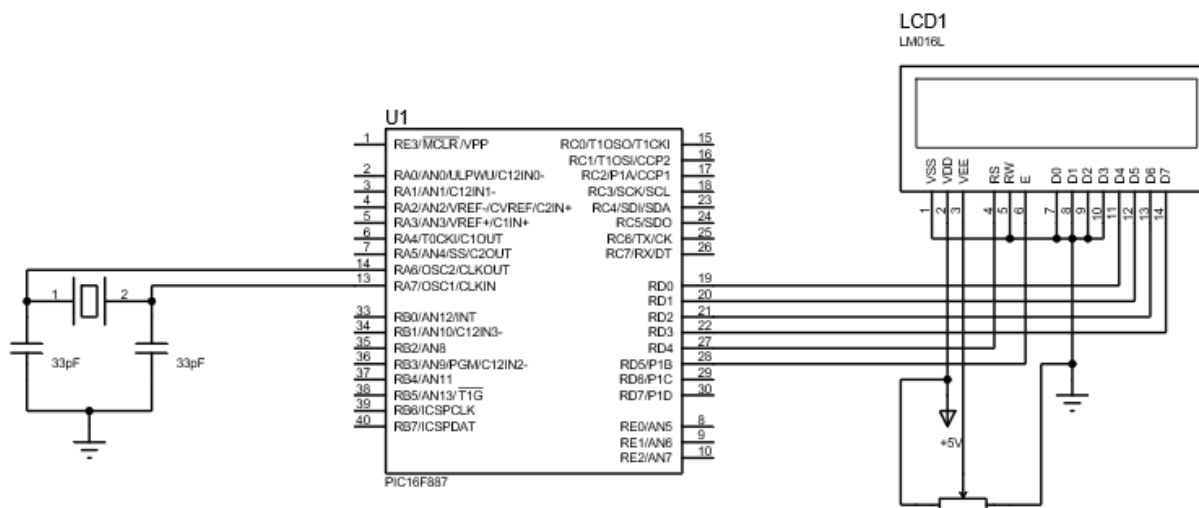
Library Routines;

- Lcd_Init
- Lcd_Out
- Lcd_Out_Cp
- Lcd_Chr
- Lcd_Chr_Cp
- Lcd_Cmd

Opis zadatka

PIC mikrokontroler 16F887 povezati sa LCD displejom preko 4 linije za podatke D4-D7 i tri linije za tri dodatne linije E, R/W i RS za kontrolu komunikacije. Na displeju prvo u prvom redu ispisati "Lcd_4bit_komun.", a u drugom "primjer". Zatim ispisati svoje ime i prezime ("Markovic Marko") i 7 puta napisani tekst pomjerati prvo u lijevu, a zatim u desnu stranu.

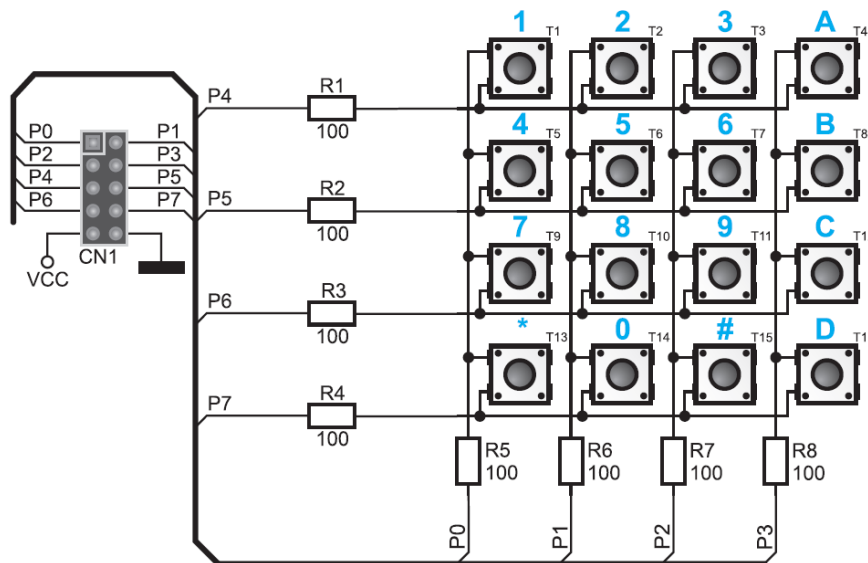
Šema spajanja



2.2. Povezivanje KeyPad-a sa PIC16F887

Uvod

Matrična tastatura je često koristi u aplikacijama gdje je potrebno unijeti brojeve i alfanumeričke karaktere kao što su kalkulator, telefon i sl. Ove tastature formirane su kao matrica, od kolona i redova u čijim presjecima postavljeni su tasteri kao što je prikazano na slici



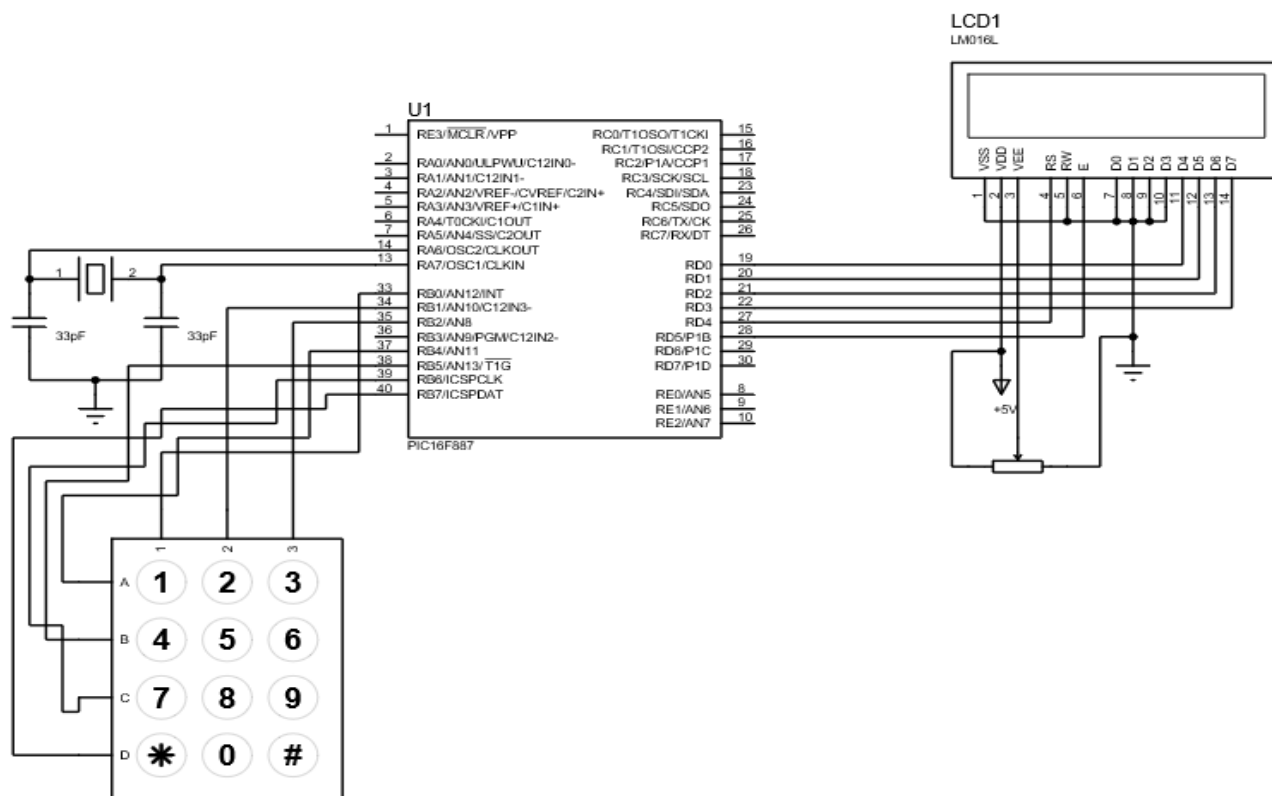
Obično su ove tastature 4x3 ili 4x4, sa 12 odnosno 16 tastera. Iz toga slijedi da je za dekodiranje tastature, odnosno prihvatanja pritisnutog tastera potrebno 12 odnosno 16 ulaznih linija PIC mikrokontrolera. Međutim, da bi se redukovao broj linija za povezivanje tastature sa PIC mikrokontrolera koristi se softverska tehnika skeniranja pritisnutog tastera, tako da se broj linija smanji na sedam za 4x3 odnosno osam za 4x4 matrične displeje. Skeniranje pritisnutog tastera izvodi se u sljedećim koracima (za matričnu tastaturu 4x4):

1. Četiri pina porta PIC mikrokontrolera definišu se kao izlazni, dok se ostala četiri definišu kao ulazni. Na izlazne pinove porta obično se priključuju “kolone” tastature, dok se “redovi” tastature priključuju na ulazne pinove porta.
2. Signal logičke “1” (HIGH) dovodi se sukcesivno na pojedine “kolone” tastature. Ukoliko je pritisnut taster u “koloni” koja je u tom trenutku logička “1”, to će se stanje preko pritisnutog tastera prenijeti u odgovarajući “red” tastature, odnosno na odgovarajući ulazni pin PIC mikrokontrolera,
3. Dekodiranjem logičkog stanja pinova koji su povezani na “kolone” i “redove” tastature identifikuje se pritisnuti taster

Opis zadatka

PIC mikrokontroler 16F887 povezati sa 4x4 (4x3 display-om iz Proteus biblioteke). Na drugi port PIC mikrokontrolera povezati LCD displej i to sa 4 linije za podatke D4-D7 i tri linije za tri dodatne linije E, R/W i RS za kontrolu komunikacije. Pritisnuti taster na tastaturi prvo detektovati, a zatim ga ispisati na tastaturi, kao i broj puta koliko je pritisnut (otpušten).

Šema spajanja:



2.3. Multipleksiranje 7Seg display-a

Uvod

Sedam segmentni display (SSD – Seven Segment Display) je jedan od najčešće korišćenih LED display-a. U suštini LED display čini više LED sijalica oblikovanih da prilikom svjetljenja pojedinih imamo željeni prikaz. Sedam segmentni display se sastoji od 7 segmenata (LED sijalica) koji su raspoređeni kao na slici ispod, i eventualno postoji dodatni segment za prikazivanje decimalnih tačaka. Radi pojednostavljenja rada anode ili katode su povezane sa zajedničkim pinom tako da imamo SSD sa zajedničkim anodama odnosno katodama. Segmenti su označeni slovima od a do g, plus dp, kao što je prikazano na slici ispod.

S obzirom da pin mikrokontrolera ima maksimalno ograničenje struje koje može da primi ili dati, poželjno je koristiti takozvane „low current“ LED kojima je potrebno samo 2mA za rad. U slučaju da unutar kućišta display nisu ugrađeni otpornici za ograničavanje struje, potrebno ih je postaviti između svakog pina mikrokontrolera i LED-a unutar display-a.



Primjetimo da spajanjem 7Seg display-a zauzimamo veliki broj izlaznih pinova mikrokontrolera. Na primjer ako bi želili da prikazemo 4-cifreni broj bilo bi nam potrebno 28 pinova mikrokontrolera. Da bi riješili ovaj problem koristimo se takozvanim multipleksiranjem.

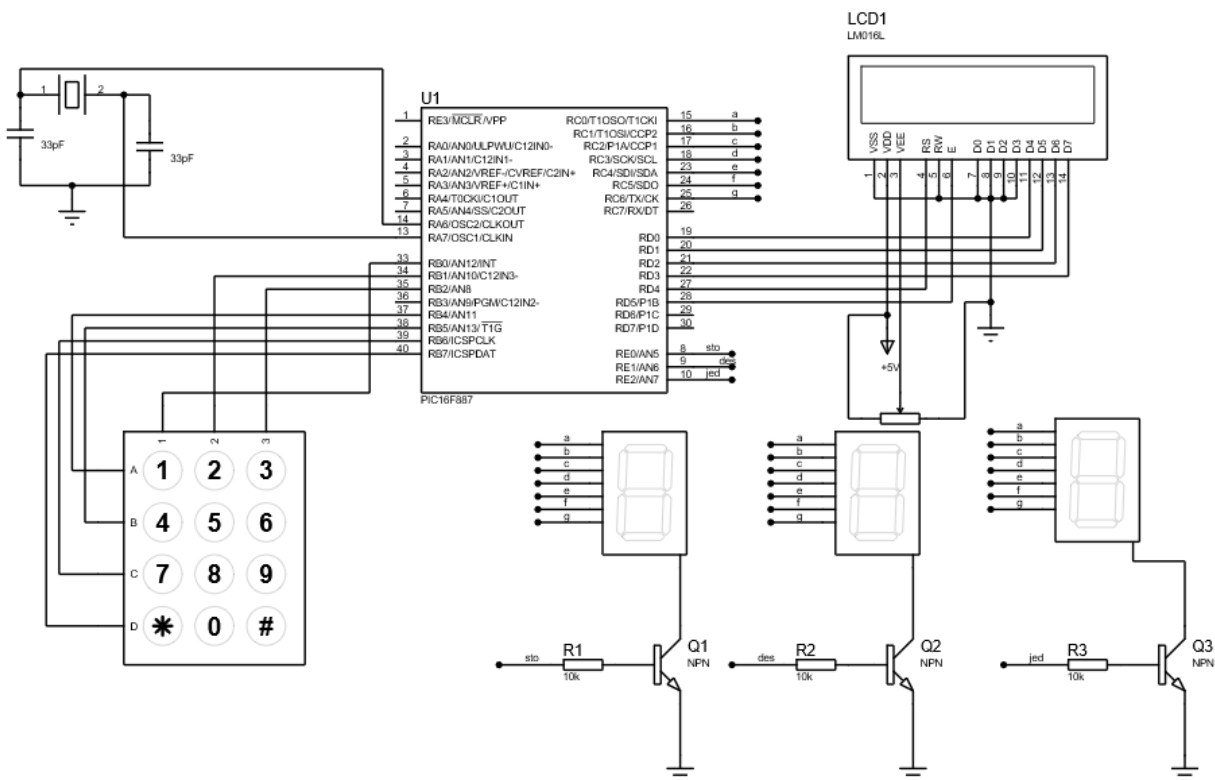
Multipleksiranje se zasniva na optičkoj iluziji. U jednom trenutku, aktivna je samo jedna cifra ali stanje cifri se tako brzo mjenja da stvara optičku iluziju kao da su sve cifre aktivne istovremeno. Na šemi spajanja sljedećeg zadatka prikazana su 3 7Seg display-a. U jednom trenutku aktivan je samo onaj display kod kojeg tranzistor spojen na zajedničku katodu provodi tj ako je na bazu tranzistora doveden pozitivan napon. Tada su na ostale baze tranzistora dovedene logičke nule te oni ne provode a samim ti i njihovi display-i ne rade. Nakon nekog trenutka uključujemo sljedeći u nizu display dovodeći mu na bazu tranzistora pozitivan napon, a na ostale tranzistore nulu, te

prikazujemo željenu cifru. Navedeni proces ciklično ponavljamo i stim da se sve realizuje brzo u našim očima se stvara optička iluzija da svi display-i rade istovremeno.

Opis zadatka

Na PORTB PIC mikrokontrolera 887 povezati KeyPad 4x3 iz Proteus biblioteke, na PORTD povezati LCD 16x2. Ostale portove mikrokontrolera iskoristiti za povezivanje tri 7Seg display-a. Prilikom pokretanja miktrokontrola na LCD display se ispisuje tekst „Unesite željeni broj:“. Omogućiti prikaz teksta u jednoj liniji LCD display-a. Korisnik unosi željeni broj te ako je on u formatu *123#, broj ispisati na LCD display-u te na 7Seg-metnim display-ima.

Šema spajanja



2.4. Konfiguracija timer0-a kao brojača

Uvod

Mikrokontroler PIC16F887 posjeduje tri potpuno nezavisna tajmera/brojača označenih kao TMR0, TMR1 i TMR2. Kada je u pitanju Timer0 on ima široku primjenu u praksi. Omogućava nam da generišemo impulse proizvoljnog trajanja, mjerenje spoljasnog impulsa.

Timer0 je 8-bit timer/brojač koji ima sljedeće funkcije:

- 8-bit tajmer/brojač
- 8-bit prescaler
- Programabilni interni ili spoljašnji sat
- Interrupt na prelivu

TIMER0 konfiguriše se preko **OPTION_REG** registra, čiji je izgled prikazan na slici.

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)
OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **T0CS – Clock Select bit**

Izbor načina rada TMR0:

1 – **TMR0** radi kao **brojač** impulsa dovedenih na I/O pin RA4

0 – **TMR0** radi kao **tajmer** i odbrojava taktne impulse $F_{osc}/4$

- **T0SE – TMR0 Source Edge Select bit**

TMR0 radi kao brojač

1 – Sadržaj registra TMR0 se uvećava nailaskom **silazne ivice** impulsa na I/O pinu RA4

0 – Sadržaj registra TMR0 se uvećava nailaskom **uzlazne ivice** impulsa na I/O pinu RA4

- **PSA – Prescaler Assignment bit**

Dodjela djelitelja:

1 – djelitelj dodjeljen **voćdog tajmeru** (WDT)

0 – djelitelj dodjeljen **tajmeru/brojaču** TMR0

- **PS2, PS1, PS0 – Prescaler Rate Select bits**

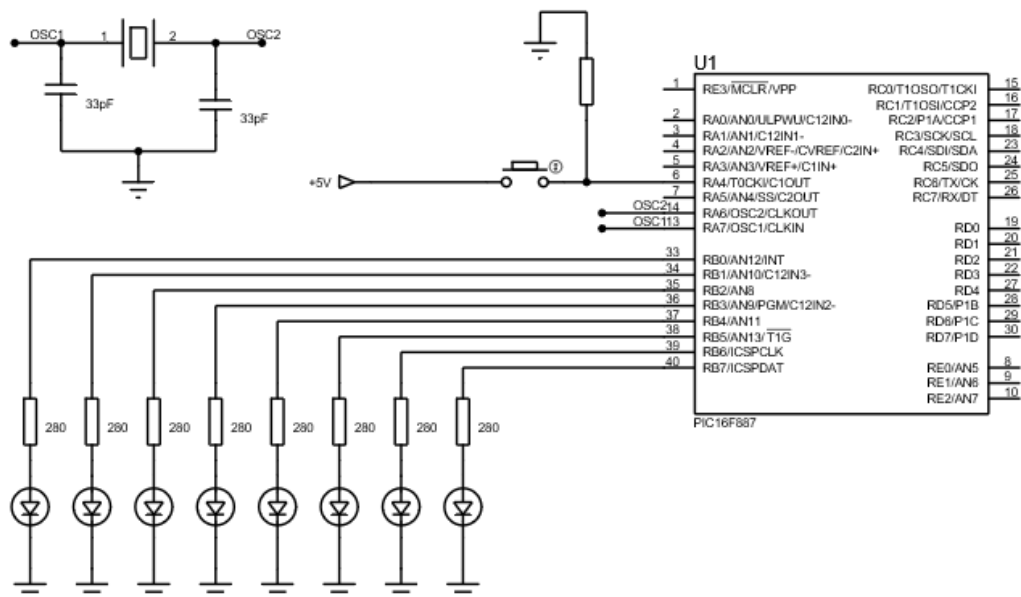
Kombinacija ovih bitova određuje faktor dijeljenja djelitelja

PS2	PS1	PS0	TMR0	WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Opis zadatka

Timer TMR0 konfigurirati da radi u brojačkom modu. Na brojački ulaz tajmera TMR0 (RA4) spojen je taster čiji broj pritisaka/otpuštanja treba odrediti. Na PORTB spojiti 8 LED dioda. Na početku je prva dioda u nizu uključena. Na svakom trećem pritisku (otpuštanju) tastera uključiti sljedeću diodu u nizu.

Šema spajanja



2.5. Konfiguracija Timer0-a kao timer u interrupt modu

Opis zadatka

Timer TMR0 konfigurirati da radi tajmerskom modu i napraviti zadržku od $T_{\text{delay}} = 0.025 \text{ s}$. .
Pretpostaviti da je frekvencija oscilatora $F_{\text{OSC}} = 8 \text{ MHz}$. Koristiti interrupt metod za opsluživanje
TIMER0. Nakon svake protekle sekunde upaliti LED koje su povezane na PORTB

Početna vrijednost TMR0

Da bi dobili potrebnu vremensku zadržku od $T_{\text{delay}} = 0.025 \text{ s}$ potrebno je odrediti faktor dijeljenja (PS2, PS1, PS0 bitove) i početnu vrijednost koju treba upisati u TMR0. Vrijeme zadržke jednog ciklusa tajmera TMR0 dato je izrazom:

$$T_{\text{ciklusa}} = \text{Prescaler} * [256 - (\text{TMR0} - 2)] * 4 * T_{\text{osc}} \quad (1)$$

Ukupno željeno vrijeme zadržke T_{delay} dobije se zbrajanjem pojedinačnih ciklusa zadržke T_{ciklusa} ako je $T_{\text{delay}} > T_{\text{ciklusa}}$.

$$T_{\text{delay}} = \text{Count} * T_{\text{ciklusa}} \quad (2)$$

Prvo odredimo najduže trajanje 1 ciklusa tajmera TMR0 koje se dobije se za: Prescaler= 256 (PS2= 1, PS1=1 , PS0= 1), vrijednost TMR0=0 i $F_{\text{OSC}} = 8 \text{ MHz}$.

Uvrštavanjem brojnih vrijednosti u (1) dobija se:

$$T_{\text{ciklusa max}} = 256 * [256 - (0 - 2)] * 4 * 125 \text{ ns}$$

$$T_{\text{ciklusa max}} = 256 * 258 * 500 \text{ ns}$$

$$T_{\text{ciklusa max}} = 33024 \mu\text{s} \approx 33 \text{ ms}$$

Pošto je $T_{\text{ciklusamax}} > T_{\text{delay}}$, onda vrijednost T_{ciklusa} treba korigovati upisom u TMR0 odgovarajuće vrijednosti. Vrijednost koju treba upisati u registar TMR0 možemo dobiti na osnovu jednačine:

$$\text{TMR0} = 2 + 256 - \frac{T_{\text{ciklusa}}}{4 * T_{\text{OSC}} * \text{Prescaler}} \quad (3)$$

Uvrštavanjem brojnih vrijednosti u (3) dobija se tražena vrijednost:

$$\text{TMR0} = 2 + 256 - \frac{0.025}{4 * 125 \text{ ns} * 256}$$

$$\text{TMR0} = 2 + 256 - \frac{50000}{256} = 62.68$$

Zakružimo vrijednost koju treba upisati u registar TRM0 na veću cjelobrojnu vrijednost:

TMR0= 63 (3Fh)

Provjerimo sada dobijeno vrijeme zadržske T_{delay} :

$$T_{\text{delay}} = \text{Prescaler} * [256 - (TMR0 - 2)] * 4 * T_{\text{osc}}$$

$$T_{\text{delay}} = 2 * 256 * [256 - (63 - 2)] * 4 * 125 \text{ ns}$$

$$T_{\text{delay}} = 256 * 195 * 500 \text{ ns} = 0.02496 \text{ s}$$

Dobijena vrijednost je bliska traženoj vrijednosti zadržske od $T_{\text{delay}} = 0.025 \text{ s}$.

Da bi TIMER0 bio konfigurisan po uslovima zadatka izgled **OPTION_REG** registar je kao na slici.

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)
OPTION_REG	X	X	0	X	0	1	1	1
	PBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

`OPTION_REG = 0x07h; // TMR0 u tajmer modu sa djeliteljem 1:256 od Fosc/4`

Podešavanje interapt sistema za TMR0

Da bi se tajmer TMR0 kontrolisao u interapt režimu potrebno je konfigurisati interapt sistem mikrokontrolera sa **INTCON** registrom.

INTCON	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bitovi **INTCON** registra koje se odnose na tajmer TMR0 su:

- **GIE – Global Interrupt Enable bit**

1 – Omogućava **sve interapte** koji nisu zabranjeni svojim lokalnim kontrolnim IE (Interrupt Enable) bitom

0 – Zabranjuje sve interapte

- **TMR0IE – TMR0 Overflow Interrupt Enable bit**

1 – Omogućava interapt od strane Tajmera0 (TMR0) pri njegovom prekoračenju

0 – Zabranjen je ovaj interapt

- **TMR0IF – TMR0 Overflow Interrupt Flag bit**

1 – Označava da je došlo do prekoračenja Timera 0 (TMR0)

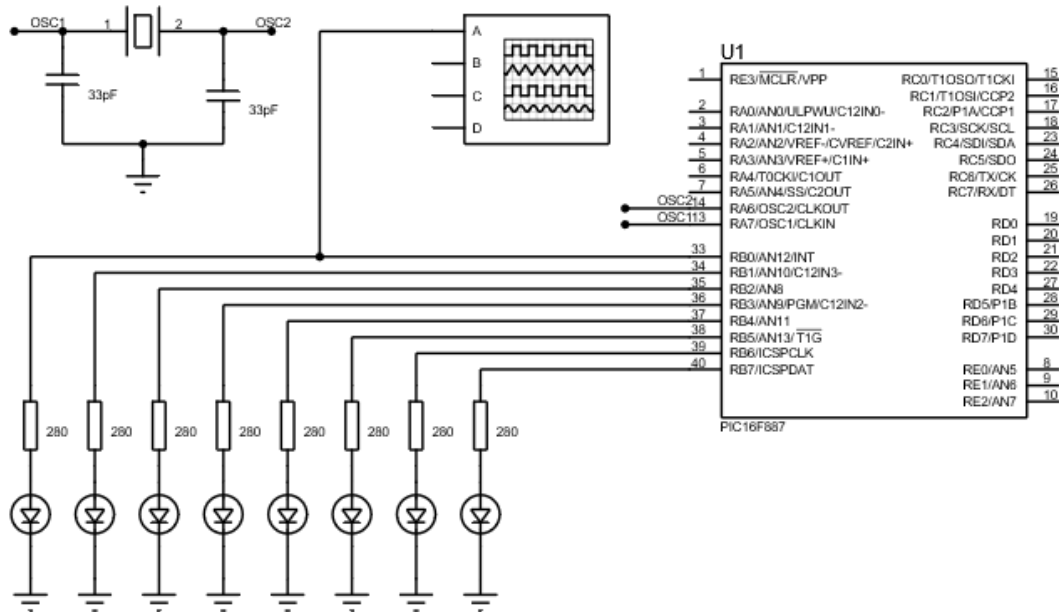
0 – Označava da nije došlo do prekoračenja Timera 0 (TMR0)

Da bi se omogućio interapt od strane TIMER0 prema uslovima zadatka izgled **INTCON** registar je kao na slici.

`INTCON = 0xA0;`

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	1	X	1	X	X	0	X	X
	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

Šema spajanja



2.6. Konfiguracija Timer1-a kao timer u interrupt modu

Opis zadatka

Tajmer **TMR1** konfigurisati da radi tajmerskom modu i napraviti zadržku od $T_{\text{delay}} = 0.25 \text{ s}$. . Pretpostaviti da je frekvencija oscilatora $F_{\text{OSC}} = 8 \text{ MHz}$. Koristiti interrupt metod za opsluživanje tajmera **TMR1**. Nakon svake proteklih 2s naizmjenično paliti i gasiti prve 4 odnosno zadnje 4 LED diode spojene na PORTB.

Analiza rješenja zadatka

Konfiguracija TMR1

TMR1 konfigurirše se preko **T1CON** registra, čiji je izgled prikazan na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1

Bitovi **T1CON** registra za konfiguraciju **TMR1** u ovom zadatku imaju sljedeća značenja:

- **T1CKPS1, T1CKPS0 – Timer 1 Clock Prescaler Select bits**

Binarna kombinacija ovih bitova određuje faktor dijeljenja signala $F_{OSC}/4$ prema datoj tabeli.

T1CKPS1	T1CKPS0	TMR1
0	0	1:1
0	1	1:2
1	0	1:4
1	1	1:8

- **TMR1CS - Clock Source Select bit**

1 – Broje se impulsi dovedeni na I/O pin T1CKI (uzlazna ivica impulsa)

0 – Broje se impulsi internog oscilatora $F_{OSC}/4$

- **TMR1ON – Timer 1 On bit**

1 – Rad tajmera TMR1 je omogućen

0 - Rad tajmera TMR1 nije omogućen

Početna vrijednost TMR1

Da bi dobili potrebnu vremensku zadržku od $T_{delay} = 0.25$ s potrebno je odrediti faktor dijeljenja (**T1CKPS1, T1CKPS0** bitove) i početnu vrijednost koju treba upisati u **TMR1**. Vrijeme zadržke jednog ciklusa tajmera **TMR1** dato je izrazom:

$$T_{ciklusa} = Prescaler * [65536 - (TMR1 - 2)] * 4 * T_{osc} \quad (1)$$

Ukupno željeno vrijeme zadržke T_{delay} dobije se zbrajanjem pojedinačnih ciklusa zadržke $T_{ciklusa}$ ako je $T_{delay} > T_{ciklusa}$.

$$T_{delay} = Count * T_{ciklusa} \quad (2)$$

Prvo odredimo najduže trajanje 1 ciklusa tajmera TMR1 koje se dobije se za: $Prescaler = 8$ ($T1CKPS1 = 1$, $T1CKPS0 = 1$), vrijednost $TMR1 = 0$ i $F_{OSC} = 8$ MHz.

Uvrštavanjem brojnih vrijednosti u (1) dobija se:

$$T_{ciklusa \max} = 8 * [65536 - (0 - 2)] * 4 * 125 \text{ ns}$$

$$T_{ciklusa \max} = 8 * 65538 * 500 \text{ ns}$$

$$T_{ciklusa \max} = 0.2621 \text{ s}$$

Pošto je $T_{ciklusamax} > T_{delay}$, onda vrijednost $T_{ciklusa}$ treba korigovati upisom u **TMR1** odgovarajuće vrijednosti. Vrijednost koju treba upisati u registar **TMR1** možemo dobiti na osnovu jednačine:

$$TMR1 = 2 + 65536 - \frac{T_{ciklusa}}{4 * T_{OSC} * Prescaler} \quad (3)$$

Uvrštavanjem brojnih vrijednosti u (3) dobija se tražena vrijednost:

$$TMR1 = 2 + 65536 - \frac{0.25}{4 * 125ns * 8}$$

$$TMR1 = 2 + 65536 - 62500 = 3038$$

Vrijednost koju treba upisati u registar **TMR1** je **TMR1= 3038 (0x0BDEh)**

Provjerimo sada dobijeno vrijeme zadržske T_{delay} :

$$T_{delay} = Prescaler * [65536 - (TMR0 - 2)] * 4 * T_{osc}$$

$$T_{delay} = 8 * [65536 - (3038 - 2)] * 4 * 125ns$$

$$T_{delay} = 8 * 62496 * 500ns = 0.249984s$$

Dobijena vrijednost je bliska traženoj vrijednosti zadržske od $T_{delay} = 0.25s$.

Da bi tajmer **TMR1** bio konfigurisan po uslovima zadatka izgled **T1CON** registar je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
T1CON	X	X	1	1	X	X	0	1
	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

`T1CON = 0x31; // TMR1 u tajmer modu sa djeliteljem 1:8 od Fosc/4`

Podešavanje interapt sistema za TMR1

Da bi se tajmer **TMR1** kontrolisao u interapt režimu portebno je konfigurisati interapt sistem mikrokontrolera sa tri registra: **INTCON**, **PIE1** i **PIR1**.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bitovi **INTCON** registra koje se odnose na tajmer **TMR1** su:

- **GIE – Global Interrupt Enable bit**

1 – Omogućava **sve interapte** koji nisu zabranjeni svojim lokalnim kontrolnim IE (Interrupt Enable) bitom
0 – Zabranjuje sve interapte

- **PEIE – Peripheral Interrupt Enable bit**

1 – Omogućava **sve interapte izazvane periferijskim modulima**, a koji koji nisu zabranjeni svojim lokalnim kontrolnim IE bitom

0 – Zabranjuje sve interapte koji su izazvani od strane periferijskih modula

Bitovi **PIE1** registra koje se odnose na tajmer **TMR1** su:

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIE1	-	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **TMR1IE – Timer1 Overflow Interrupt Enable bit**

1 - Omogućava se zahtijev za interapt od strane Timer1 (TMR1), koji se generiše pri njegovom prekoračenju

0 – Zabranjuje sve ovaj intarapt

Bitovi **PIR1** registra koje se odnose na tajmer **TMR1** su:

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIR1	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **TMR1IF - Timer1 Overflow Interrupt Flag bit**

1 – Označava da je došlo do prekoračenja Tajmera 1 (TMR1)

0 – Označava da nije došlo do prekoračenja Tajmera 1 (TMR1)

Da bi se omogućio interapt od strane tajmera **TMR1** prema uslovima zadatka izgled **INTCON**, **PIE1** i **PIR1** registara je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	1	1	X	X	X	X	X	X
	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIE1	-	X	X	X	X	X	X	1
	-	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIR1	-	X	X	X	X	X	X	0
	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

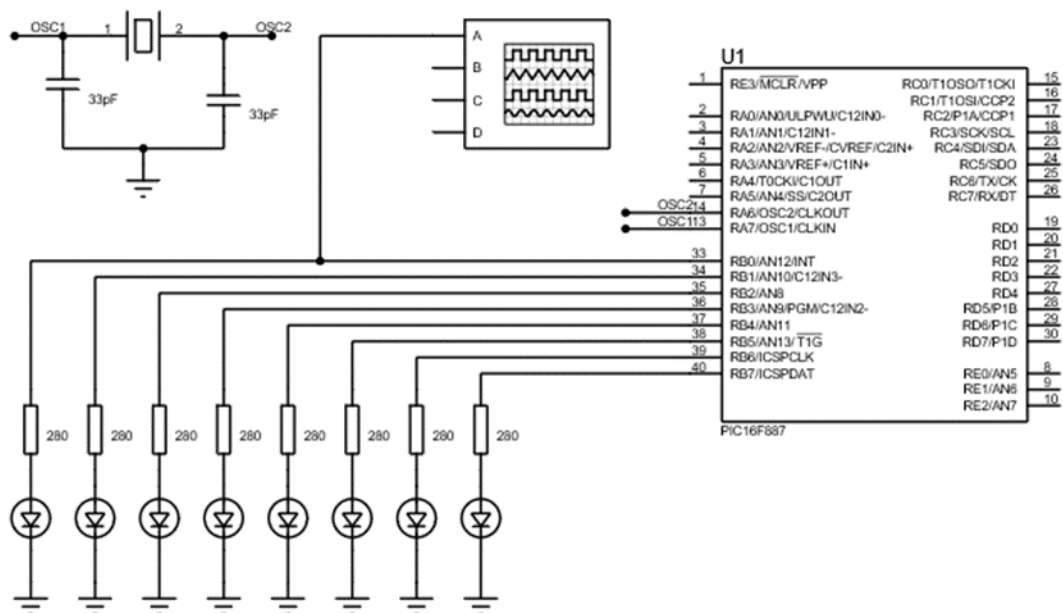
`INTCON.GIE = 1; // Omogućiti interapte, bit GIE u INTCON registru`

`INTCON.PEIE = 1; // Omogućiti interapte periferala, bit PEIE u INTCON registru`

`PIR1.TMR1IF = 0; // resetovati interapt fleg TMR1IF`

`PIE1.TMR1IE = 1; // Omogućiti TMR1 interapt`

Šema spajanja



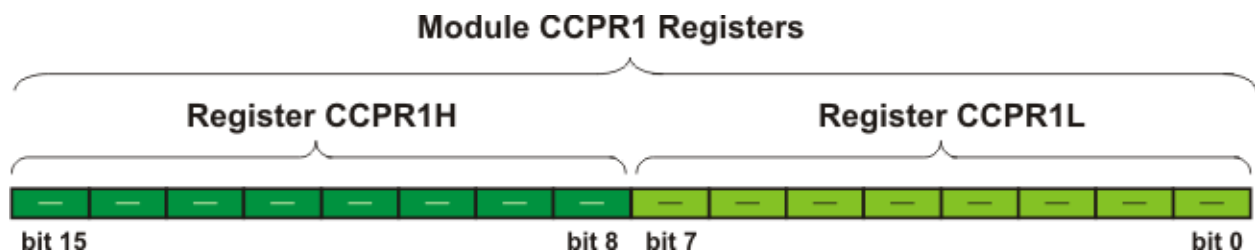
2.7. Mjerenje perioda sa CCP modulom

CCP moduli

Skraćenica CCP označava Capture/Compare/PWM. Zahvaljujući CCP modulu imamo mogućnost određivanja trajanja nekog događaja ili iniciranje nekog događaja nakon isteklog vremena.

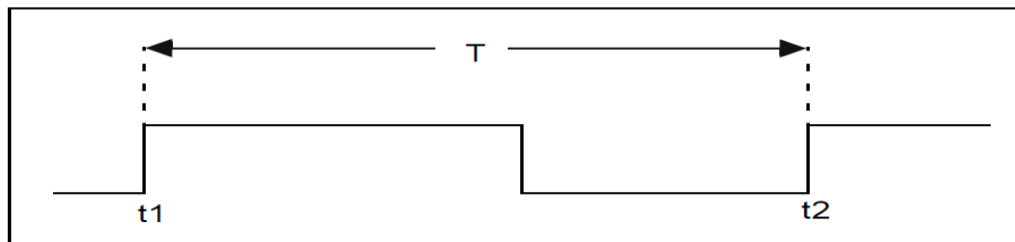
- Capture (prihvatni) mod – određuje vrijeme trajanja nekog događaja
- Compare (poređenje) mod – Upoređuje vrijednosti sadržane u dva registra od kojih je jedan TMR1. Omogućava aktiviranje nekog eksternog događaja nakon isteka unaprijed određenog vremena
- PWM – Pulse Width Modulation – omogućava kreiranje signala željene frekvencije i faktora ispunje

Glavni dio CCP modula je 16-bitni registar CCPR1, koji se sastoji od CCPR1H i CCPR1L registara. Koristi se za poređenje sa sadržajem registra TMR1 (TMR1H i TMR1L)



Opis zadatka

Pomoću **CCP1** modula odrediti period pravougaonih impulsa koji se dovode na pin RC2/P1A/CCP1 PIC mikrokontrolera. **CCP1** modul konfigurirati da radi u režimu hvatanja (capture) na svaku rastuću ivicu pravougaonog impulsa na pin RC2/P1A/CCP. Pretpostaviti da je frekvencija oscilatora $F_{osc}=20$ MHz. Koristiti interrupt metod za opsluživanje **CCP1** modula i tajmera **TMR1**. Izmjereni period/frekvenciju prikazati na LCD displeju koji treba biti povezan na PORTD.

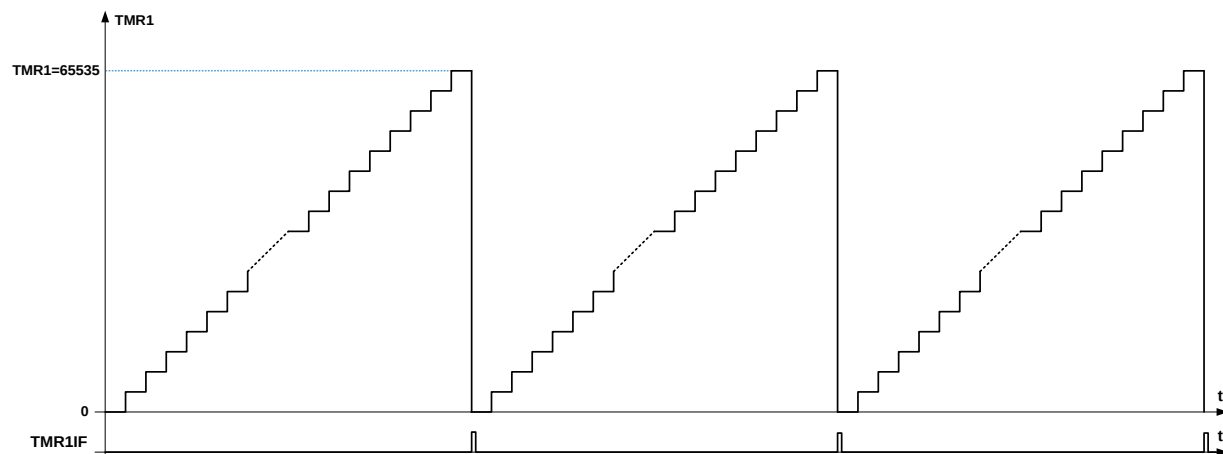


Lista povezivanja	
Port mikrokontrolera	Vanjske komponente
RD0	LED0
RD1	LED1
RD2	LED2
RD3	LED3
RD4	LED4
RD5	LED5
RD6	LED6
RD7	LED7

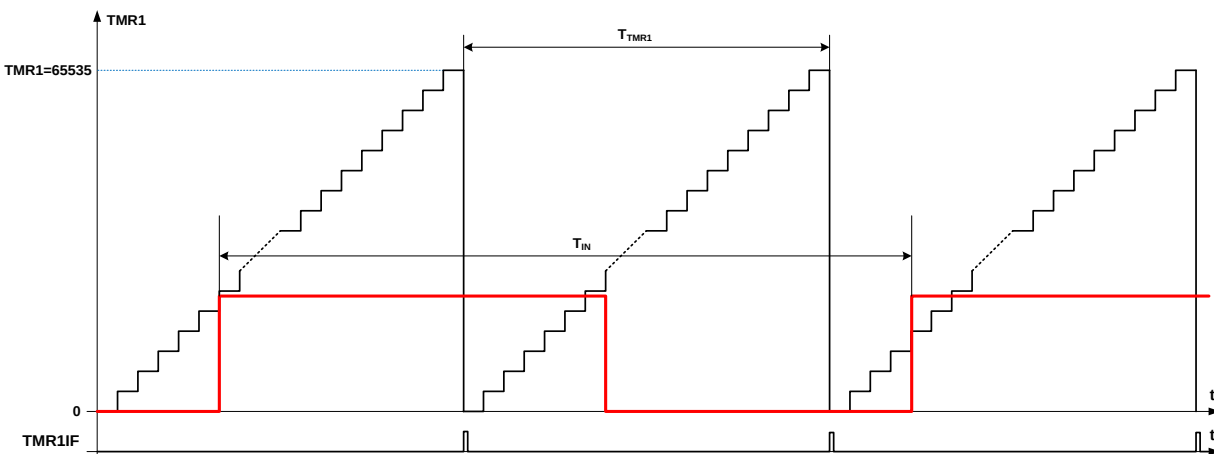
Analiza rješenja zadatka

Algoritam rada

TMR1 je sastavni dio **CCP1** modula i modu "hvatanja" treba ga postaviti da radi u **free - running** (kružni tajmer) modu kao što je prikazano na slici

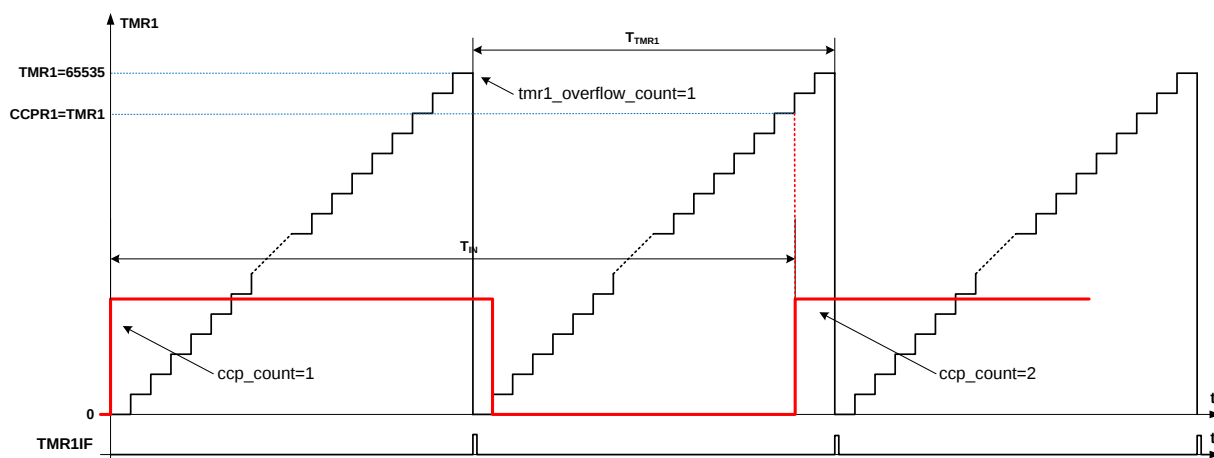


Prilikom mjerenja perioda pravougaonih impulsa koji se dovode na pin RC2/P1A/CCP1 PIC mikrokontrolera mogu se javiti dva slučaja. U prvom slučaju period pravougaonih impulsa T_{IN} duži je od perioda **free - running** (kružni tajmer) tajmera T_{TMR1} , kao što je prikazana na slici



Za mjerenja perioda pravougaonih impulsa T_{IN} u ovom slučaju primjenjen je sljedeći algoritam:

1. Sa kontrolnim bitovima **CCP1M3:CCP1M0** podesiti da **CCP1** modul "hvata" svaku rastuću ivicu ulaznih pravougaonih impulsa i omogućiti interapt od strane **CCP1** modula
2. Konfigurisati **TMR1** tako da ima najduži priod T_{TMR1} sa djeliteljem 1:1 (bitovi **T1CKPS1**, **T1CKPS0**) i omogućiti njegov interapt
3. Pri prvoj rastućoj ivici pravougaonog impulsa u registre **TRM1H** i **TRM1L** upisati **TRM1H = 0x00h** i **TRM1L = 0x00h**, tako da se početak rada tajmera sinhronizuje sa prvom rastućom ivicom ulaznog pravpugaonog impulsa kao što je prikazano na slici.

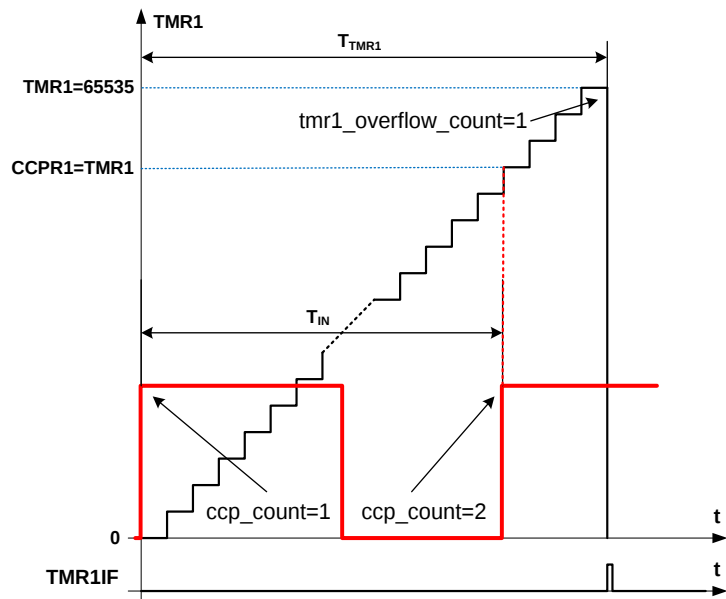


4. Sa promjenljivom **tmr1_overflow_count** broji se broj prekoračenja brojača (na slici **tmr1_overflow_count = 1**), a sa promjenljivom **ccp_count** broj rastućih ivica ulaznih pravougaonih impulsa. Pri drugoj rastućoj ivici ulaznog impulsa (**ccp_count = 2**) dolazi do njegovog "hvatanja", odnosno trenutna vrijednost tajmera **TMR1** kopira se u **CCPR1** registar CCP modula. Period ulaznih pravougaonih impulsa dobija se na osnovu izraza:

$$T_{IN} = (tmr1_overflow_count) * 65536 + CCPR1 - tmr1_overflow_count$$

gdje sa sa **tmr1_overflow_count** pravi korekcija vremena usljed latencije interuupt rutine za opsluživanje tajmera **TMR1**.

U drugom slučaju period pravougaonih impulsa T_{IN} kraći je od perioda **free - running** (kružni tajmer) tajmera T_{TMR1} , kao što je prikazana na slici



Lako je uočiti da se ovaj drugi slučaj mjerenja perioda pravougaonih impulsa samo poseban slučaj prethodno opisanog za **$tmr1_overflow_count = 0$** .

Konfiguracija TMR1

TMR1 konfiguriše se preko **T1CON** registra, čiji je izgled prikazan na slici.

T1CON	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	
	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bitovi **T1CON** registra za konfiguraciju **TMR1** u ovom zadatku imaju sljedeća značenja:

- **T1CKPS1, T1CKPS0 – Timer 1 Clock Prescaler Select bits**

Binarna kombinacija ovih bitova određuje faktor dijeljenja signala $F_{OSC}/4$ prema datoj tabeli.

T1CKPS1	T1CKPS0	TMR1
0	0	1:1
0	1	1:2
1	0	1:4
1	1	1:8

- **TMR1CS - Clock Source Select bit**

1 – Broje se impulsi dovedeni na I/O pin T1CKI (uzlazna ivica impulsa)

0 – Broje se impulsi internog oscilatora $F_{osc}/4$

- **TMR1ON** – Timer 1 On bit

1 – Rad tajmera TMR1 je omogućen

0 - Rad tajmera TMR1 nije omogućen

Početna vrijednost TMR1

Da bi **TMR1** radio kao kružni (*free-running*) tajmer sa najdužim periodom T_{TMR1} pri djeljitelju 1:1 (bitovi **T1CKPS1**, **T1CKPS0**) početna vrijednost tajmera TMR1 mora biti nula, tj. **TRM1H** = **0x00h** i **TRM1L** = **0x00h**.

Vrijeme jednog ciklusa T_{TMR1} tajmera **TMR1** dato je izrazom:

$$T_{TMR1} = Prescaler * [65536 - (TMR1 - 2)] * 4 * T_{osc} \quad (1)$$

Trajanje 1 ciklusa tajmera **TMR1** dobije se za: $Prescaler = 1$ (bitovi **T1CKPS1** = 0, **T1CKPS0** = 0), vrijednost **TMR1** = 0 i $F_{osc} = 20$ MHz.

Uvrštavanjem brojnih vrijednosti u (1) dobija se:

$$T_{TMR1max} = 1 * [65536 - (0 - 2)] * 4 * 50 \text{ ns}$$

$$T_{TMR1max} = 65538 * 500 \text{ ns}$$

$$T_{TMR1max} = 0.32769 \text{ s}$$

Dakle, ukoliko je period pravougaonih impulsa veći od $T_{TMR1max}$ imamo prvi slučaj mjerenja kada je potrebno brojati i prekoračenja tajmera *tmr1_overflow_count*, a u suprotnom imamo drugi slučaj gdje ne dolazi do njegovog prekoračenja.

Da bi tajmer **TMR1** bio konfigurisan po uslovima zadatka izgled **T1CON** registar je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
T1CON	X	X	0	0	X	X	0
	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS
							TMR1ON

`T1CON = 0x01; // TMR1 u tajmer modu sa djeljiteljem 1:1 od Fosc/4`

Konfiguracija CCP1 modula

CCP1 modul konfiguriše se preko **CCP1CON** registra, čiji je izgled prikazan na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
							CCP1M0
							Bit 0

Bitovi **CCP1CON** registra za konfiguraciju **CCP1** modula u ovom zadatku imaju sljedeća značenja:

- **CCP1M3 – CCP1M0 ECCP Mode Select bits**

CCP1 M3	CCP1 M2	CCP1 M1	CCP1 M0	Način rada CCP1 modula
0	0	0	0	Pad modula je onemogućen
0	0	0	1	Ne koristi se
0	0	1	0	Mod za poređenje (compare) Izlaz mijenja stanje i setuje se bit CCP1IF
0	0	1	1	Ne koristi se
0	1	0	0	Mod za prihvatanje (capture) Na svaku silaznu ivicu impulsa na RC2/CCP1
0	1	0	1	Mod za prihvatanje (capture) Na svaku uzlaznu ivicu impulsa na RC2/CCP1
0	1	1	0	Mod za prihvatanje (capture) Na svaku četvrtu uzlaznu ivicu impulsa na RC2/CCP1

Prema uslovu zadatka CCP1 modul treba da radi u modu za prihvatanje (*capture*) na svaku uzlaznu ivicu impulsa na **RC2/CCP1** pinu, što odgovara kombinaciji bitova **CCP1M3 = 0**, **CCP1M2 = 1**, **CCP1M1 = 0** i **CCP1M0 = 1**.

Da bi **CCP1** modul bio konfigurisan po uslovima zadatka izgled **CCP1CON** registra je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
CCP1CON	X	X	X	X	0	1	0	1
	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCPM1	CCPM0

CCP1CON = 0x05; // CCP1 modul u modu prihvatanja (capture) na svaku rastucu ivicu ulaznog impulsa

Podešavanje interapt sistema za TMR1 i CCP1 modul

Da bi se tajmer **TMR1** i **CCP1** modul kontrolisali u interapt režimu portebno je konfigurisati interapt sistem mikrokontrolera sa tri registra: **INTCON**, **PIE1** i **PIR1**.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bitovi **INTCON** registra koje se odnose na tajmer **TMR1** i **CCP1** modul su:

- **GIE** – Global Interrupt Enable bit

1 – Omogućava **sve interapte** koji nisu zabranjeni svojim lokalnim kontrolnim IE (Interrupt Enable) bitom

0 – Zabranjuje sve interapte

- **PEIE – Peripheral Interrupt Enable bit**

1 – Omogućava sve interapte izazvane perifernim modulima, a koji koji nisu zabranjeni svojim lokalnim kontrolnim IE bitom

0 – Zabranjuje sve interapte koji su izazvani od strane perifernih modula

Bitovi **PIE1** registra koje se odnose na tajmer **TMR1** i **CCP1** modul su:

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIE1	-	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **CCP1IE - CCP1 Interrupt Enable bit**

1 – Omogućava se zahtijev za interapt od strane **CCP1** modula, koji se generiše pri promjeni stanja na I/O pinu **RC2/CCP1**

0 – Zabranjuje sve ovaj intarapt

- **TMR1IE – Timer1 Overflow Interrupt Enable bit**

1 - Omogućava se zahtijev za interapt od strane Timer1 (**TMR1**), koji se generiše pri njegovom prekoračenju

0 – Zabranjuje sve ovaj intarapt

Bitovi **PIR1** registra koje se odnose na tajmer **TMR1** i **CCP1** modul su:

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIR1	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **CCP1IF - CCP1 Interrupt Flag bit**

Capture (prihvatni) mod

1 – Došlo je do promjene logičkog stanja na I/O pinu **RC2/CCP1** i sadržaj tajmera **TMR1** je preslikan u **CCPR1** registar

0 – Nije došlo do preslikavanja sadržaj tajmera **TMR1** u **CCPR1** registar

- **TMR1IF - Timer1 Overflow Interrupt Flag bit**

1 – Označava da je došlo do prekoračenja Tajmera 1 (**TMR1**)

0 – Označava da nije došlo do prekoračenja Tajmera 1 (**TMR1**)

Da bi se omogućio interapt od strane tajmera **TMR1** prema uslovima zadatka izgled **INTCON**, **PIE1** i **PIR1** registara je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	1	1	X	X	X	X	X
	GIE	PEIE	TOIE	INTE	RBIE	TOIF	RBIF

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIE1	-	X	X	X	X	1	X
	-	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE

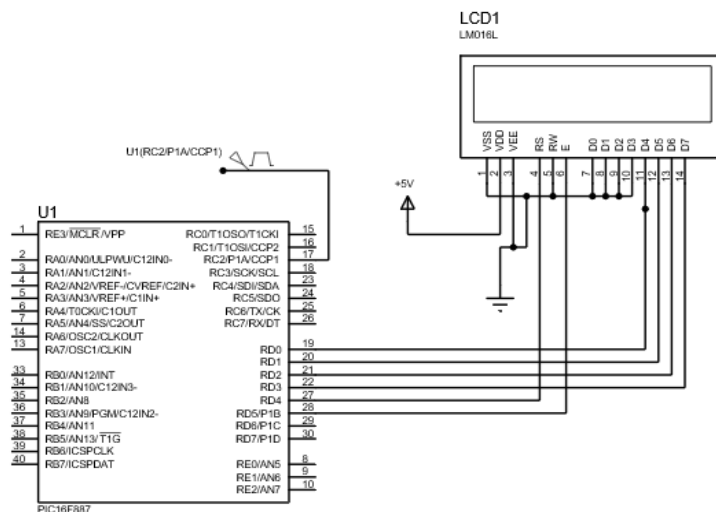
		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIR1	-	X	X	X	X	0	X	0
	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

```

INTCON.GIE = 1; // Omoguciti interapte, bit GIE u INTCON registru
INTCON.PEIE = 1; // Omoguciti interapte periferala, bit PEIE u INTCON registru
PIE1.CCP1IE = 1; // Omoguciti interapt od strane CCP1 modula
PIE1.TMR1IE = 1; // Omoguciti interapt od strane tajmera TMR1
PIR1.CCP1IF = 0; // Obrisati CCP1IF
PIR1.TMR1IF = 0; // Obrisati TMR1IF

```

Šema spajanja



2.8. CCP – mjerenje faktora ispunje

Opis zadatka

Pomoću **CCP1** modula odrediti faktor ispunje (duty cycle) pravougaonih impulsa koji se dovode na pin RC2/P1A/CCP1 PIC mikrokontrolera. **CCP1** modul konfigurisati da radi u režimu hvatanja (capture) prvo na rastuću ivicu, a zatim na opadajuću ivicu pravougaonog impulsa na pin RC2/P1A/CCP. Pretpostaviti da je frekvencija oscilatora $F_{OSC}=20$ MHz. Koristiti interapt

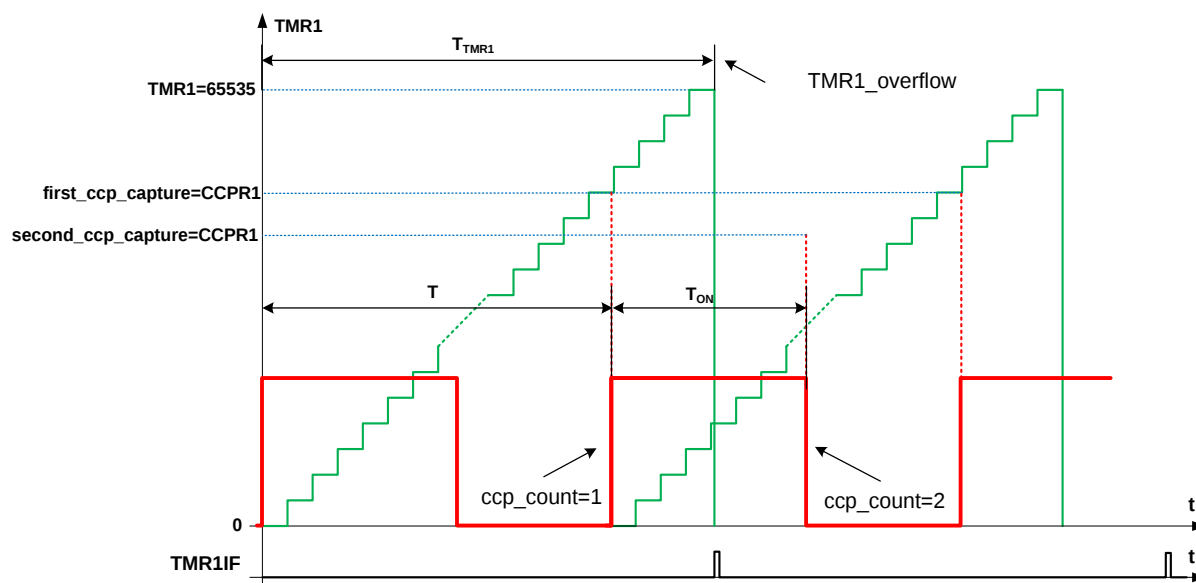
metod za opsluživanje **CCP1** modula. Postaviti da tajmer **TMR1** ima najduži ciklus. Izmjereni faktor ispune prikazati na LCD displeju koji treba biti povezan na PORTD.

Analiza rješenja zadatka

Algoritam rada

Za mjerenja faktora ispune pravougaonih impulsa T_{IN} primjenjen je sljedeći algoritam:

1. Sa kontrolnim bitovima **CCP1M3:CCP1M0** podesiti da **CCP1** modul prvo "hvata" svaku rastuću ivicu ulaznih pravougaonih impulsa i omogućiti interapt od strane **CCP1** modula
2. Konfigurisati **TMR1** tako da ima najduži priod T_{TMR1} sa djeliteljem 1:1 (bitovi **T1CKPS1, T1CKPS0**) i omogućiti njegov interapt
3. uvećati brojač ivica ulaznog signala **ccp_count** za jedan
4. obrisati **CCP1IF**
4. U dijelu interapt rutine koja opslužuje CCP modul i koji se izvršava pri svakoj rastućoj ivici pravougaonog impulsa uraditi sljedeće:
 - u promjenjivu **first_ccp_capture** kopirati trenutnu vrijednost registra **CCPR1** koja odgovara periodu ulaznih impulsa T_{ON}
 - u registre **TRM1H** i **TRM1L** upisati **TRM1H = 0x00h** i **TRM1L = 0x00h** čime se početak rada tajmera **TMR1** sinhronizuje sa rastućim ivicama ulaznog pravougaonog impulsa kao što je prikazano na slici.
 - promijenti sadržaj registra **CCP1CON** tako da se omogući "hvatanje" na opadajuću ivicu ulaznih impulsa



5. U dijelu interapt rutine koja opslužuje CCP modul i koji se izvršava pri svakoj opadajućoj ivici pravougaonog impulsa uraditi sljedeće:

- u promjenjivu **second_ccp_capture** kopirati trenutnu vrijednost registra **CCPR1** koja odgovara trajanju periodu ulaznih impulsa T_{ON}
- promijenti sadržaj registra **CCP1CON** tako da se omogući "hvatanje" na svaku rastuću ivicu ulaznih impulsa
- obrisati brojač ivica ulaznog signala **ccp_count**

Konfiguracija TMR1

TMR1 konfigurira se preko **T1CON** registra, čiji je izgled prikazan na slici.

T1CON	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	
	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bitovi **T1CON** registra za konfiguraciju **TMR1** u ovom zadatku imaju sljedeća značenja:

- **T1CKPS1, T1CKPS0 – Timer 1 Clock Prescaler Select bits**

Binarna kombinacija ovih bitova određuje faktor dijeljenja signala $F_{osc}/4$ prema datoj tabeli.

T1CKPS1	T1CKPS0	TMR1
0	0	1:1
0	1	1:2
1	0	1:4
1	1	1:8

- **TMR1CS - Clock Source Select bit**

1 – Broje se impulsi dovedeni na I/O pin T1CKI (uzlazna ivica impulsa)

0 – Broje se impulsi internog oscilatora $F_{osc}/4$

- **TMR1ON – Timer 1 On bit**

1 – Rad tajmera TMR1 je omogućen

0 - Rad tajmera TMR1 nije omogućen

Početna vrijednost TMR1

Da bi **TMR1** radio kao sa najdužim periodom T_{TMR1} pri djelitelju 1:1 (bitovi **T1CKPS1, T1CKPS0**) početna vrijednost tajmera TMR1 mora biti nula, tj. **TRM1H = 0x00h** i **TRM1L = 0x00h**.

Vrijeme jednog ciklusa T_{TMR1} tajmera **TMR1** dato je izrazom:

$$T_{TMR1} = Prescaler * [65536 - (TMR1 - 2)] * 4 * T_{osc} \quad (1)$$

Trajanje 1 ciklusa tajmera **TMR1** dobije se za: $Prescaler = 1$ (bitovi **T1CKPS1 = 0, T1CKPS0 = 0**), vrijednost **TMR1 = 0** i $F_{osc} = 20 \text{ MHz}$.

Uvrštavanjem brojnih vrijednosti u (1) dobija se:

$$T_{TMR1max} = 1 * [65536 - (0 - 2)] * 4 * 50 \text{ ns}$$

$$T_{TMR1max} = 65538 * 500 \text{ ns}$$

$$T_{TMR1max} = 0.032769 \text{ s}$$

Dobijeni period $T_{TMR1max}$ predstavlja ograničenje, pa faktor ispunje ulaznih signala kojima je period manji od $T_{TMR1max}$ neće moći biti određen. U konkretnom slučaju za ulazne provougaoni signali sa frekvencijom manjom od 32 Hz faktor ispunje neće moći biti određen po predloženom algoritmu

Da bi tajmer **TMR1** bio konfigurisan po uslovima zadatka izgled **T1CON** registar je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
T1CON	X	X	0	0	X	X	0
	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS
							TMR1ON

`T1CON = 0x01; // TMR1 u tajmer modu sa djeliteljem 1:1 od Fosc/4`

Konfiguracija CCP1 modula

CCP1 modul konfigurise se preko **CCP1CON** registra, čiji je izgled prikazan na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
							Bit 0

Bitovi **CCP1CON** registra za konfiguraciju **CCP1** modula u ovom zadatku imaju sljedeća značenja:

- CCP1M3 – CCP1M0 ECCP Mode Select bits**

CCP1 M3	CCP1 M2	CCP1 M1	CCP1 M0	Način rada CCP1 modula
0	0	0	0	Pad modula je onemogućen
0	0	0	1	Ne koristi se
0	0	1	0	Mod za poređenje (compare)
				Izlaz mijenja stanje i setuje se bit CCP1IF
0	0	1	1	Ne koristi se
0	1	0	0	Mod za prihvatanje (capture)
				Na svaku silaznu ivicu impulsa na RC2/CCP1
0	1	0	1	Mod za prihvatanje (capture)
				Na svaku uzlaznu ivicu impulsa na RC2/CCP1
0	1	1	0	Mod za prihvatanje (capture)
				Na svaku četvrtu uzlaznu ivicu impulsa na RC2/CCP1

Prema uslovu zadatka CCP1 modul treba da radi u modu za prihvatanje (**capture**) na svaku uzlaznu i silaznu ivicu impulsa na **RC2/CCP1** pinu, što odgovara kombinaciji bitova **CCP1M3**

= 0, CCP1M2 = 1, CCP1M1 = 0 i CCP1M0 = 1 i CCP1M3 = 0, CCP1M2 = 1, CCP1M1 = 0 i CCP1M0 = 0.

Da bi CCP1 modul bio konfigurisan po uslovima zadatka izgled **CCP1CON** registra je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
CCP1CON	X	X	X	X	0	1	0	1
	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCPM1	CCPM0

CCP1CON = 0x05; // CCP1 modul u modu prihvatanja (capture) na svaku rastucu ivicu ulaznog impulsa

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
CCP1CON	X	X	X	X	0	1	0	0
	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCPM1	CCPM0

CCP1CON = 0x04; // CCP1 modul u modu prihvatanja (capture) na svaku opadajucu ivicu ulaznog impulsa

Podešavanje interapt sistema CCP1 modul

Da bi se CCP1 modul kontrolisao u interapt režimu portebno je konfigurisati interapt sistem mikrokontrolera sa tri registra: **INTCON**, **PIE1** i **PIR1**.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1

Bitovi **INTCON** registra koje se odnose na tajmer TMR1 i CCP1 modul su:

- **GIE – Global Interrupt Enable bit**

1 – Omogućava **sve interapte** koji nisu zabranjeni svojim lokalnim kontrolnim IE (Interrupt Enable) bitom
0 – Zabranjuje sve interapte

- **PEIE – Peripheral Interrupt Enable bit**

1 – Omogućava **sve interapte izazvane perifernim modulima**, a koji nisu zabranjeni svojim lokalnim kontrolnim IE bitom

0 – Zabranjuje sve interapte koji su izazvani od strane perifernih modula

Bitovi **PIE1** registra koje se odnose na CCP1 modul su:

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIE1	-	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1

- **CCP1IE - CCP1 Interrupt Enable bit**

1 – Omogućava se zahtijev za interapt od strane **CCP1** modula, koji se generiše pri promjeni stanja na I/O pinu **RC2/CCP1**

0 – Zabranjuje sve ovaj intarapt

Bitovi **PIR1** registra koje se odnose na **CCP1** modul su:

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIR1	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **CCP1IF - CCP1 Interrupt Flag bit**

Capture (prihvatni) mod

1 – Došlo je do promjene logičkog stanja na I/O pinu **RC2/CCP1** i sadržaj tajmera **TMR1** je preslikan u **CCPR1** registar

0 – Nije došlo do preslikavanja sadržaj tajmera **TMR1** u **CCPR1** registar

Da bi se omogućio interapt od strane **CCP1** modula prema uslovima zadatka izgled **INTCON**, **PIE1** i **PIR1** registara je kao na slici.

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (X)
INTCON	1	1	X	X	X	X	X	X
	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIE1	-	X	X	X	X	1	X	0
	-	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
PIR1	-	X	X	X	X	0	X	X
	-	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

`INTCON.GIE = 1; // Omoguciti interapte, bit GIE u INTCON registru`

`INTCON.PEIE = 1; // Omoguciti interapte periferala, bit PEIE u INTCON registru`

`PIE1.CCP1IE = 1; // Omoguciti interapt od strane CCP1 modula`

`PIE1.TMR1IE = 0; // Zabraniti interapt od strane tajmera TMR1`

`PIR1.CCP1IF = 0; // Obrisati CCP1IF`

3. Pitanja i zadaci

1. PIC mikrokontroler 16F887 povezati sa 4x4 (4x3 display-om iz Proteus biblioteke). Na drugi port PIC mikrokontrolera povezati LCD displej i to sa 4 linije za podatke D4-D7 i tri linije za tri dodatne linije E, R/W i RS za kontrolu komunikacije. Pritisnuti taster na tastaturi prvo detektovati, a zatim ga ispisati na LCD, kao i broj puta koliko je pritisnut (otpušten).
2. Na PORTB PIC mikrokontrolera 16F887 povezati KeyPad 4x3 iz Proteus biblioteke, a na PORTD povezati LCD 16x2. Ostale portove mikrokontrolera iskoristiti za povezivanje tri 7Seg display-a. Prilikom pokretanja miktrokontrola na LCD display se ispisuje tekst „Unesite željeni broj:“. Omogućiti prikaz teksta u jednoj liniji LCD display-a. Korisnik unosi željeni broj te ako je on u formatu *123#, broj ispisati na LCD display-u te na 7Seg-metnim display-ima.
3. Na PORTB PIC mikrokontrolera 16F887 povezati KeyPad 4x3 iz Proteus biblioteke, na PORTD povezati LCD 16x2. Ostale portove mikrokontrolera iskoristiti za povezivanje tri 7Seg display-a. Pritisnuti taster na tastaturi prvo detektovati, a zatim ga ispisati na LCD, kao i broj puta koliko je pritisnut (otpušten). Broj puta koliko je pritisnut taster uvecati za 120 te ispisati i na 7Seg-metnim display-ima.
4. Timer **TMR0** konfigurisati da radi u brojačkom modu. Na brojački ulaz tajmera TMR0 (RA4) spojen je taster čiji broj pritisaka/otpuštanja treba odrediti. Na PORTB spojiti 8 LED dioda. Na pocetku je prva dioda u nizu ukljucena. Na svakom trećem pritisku (otpuštanju) tastera uključiti sljedeću diodu u nizu. Na PORTE spojiti LCD display te ispisivati koliko je ukupno puta taster pritisnut (otpušten).
5. Tajmer **TMR1** konfigurisati da radi tajmerskom modu i napraviti zadržku od $T_{\text{delay}} = 0.25$ s. . Pretpostaviti da je frekvencija oscilatora $F_{\text{osc}} = 8$ MHz. Koristiti interapt metod za opsluživanje tajmera **TMR1**. Na PORTD spojiti LCD display. Na neki od pinova mikrokontrolera postaviti prekidač. Na display-u ispisivati vrijeme proteklo od zatvaranja prekidača.
6. Tajmer **TMR0** konfigurisati da radi tajmerskom modu i napraviti zadržku od $T_{\text{delay}} = 0.25$ s. . Pretpostaviti da je frekvencija oscilatora $F_{\text{osc}} = 8$ MHz. Koristiti interapt metod za opsluživanje tajmera **TMR0**. Na PORTD spojiti LCD display. Na neki od pinova mikrokontrolera postaviti taster. Na display-u ispisivati vrijeme proteklo od posljednjeg pritiska tastera.
7. Tajmer **TMR1** konfigurisati da radi tajmerskom modu i napraviti zadržku od $T_{\text{delay}} = 0.25$ s. Pretpostaviti da je frekvencija oscilatora $F_{\text{osc}} = 8$ MHz. Koristiti interapt metod za opsluživanje tajmera **TMR1**. Na **PORTD** spojiti LCD display. Na **PORTE** spojiti **KeyPad 4x3**. Omogcuti korisniku unos željenog broja preko **KeyPad**-a. Zatim koristeći napavljenu zadrsku pomoću **TMR1**, na LCD display-u odbrojavati od unijetog broja do nule.

