

## Laboratorij računalnog inženjerstva

Računala i procesi - treća laboratorijska vježba

Pulsno širinska modulacija, PID regulacija

## 1. Priprema

Prije izrade vježbe obavezno:

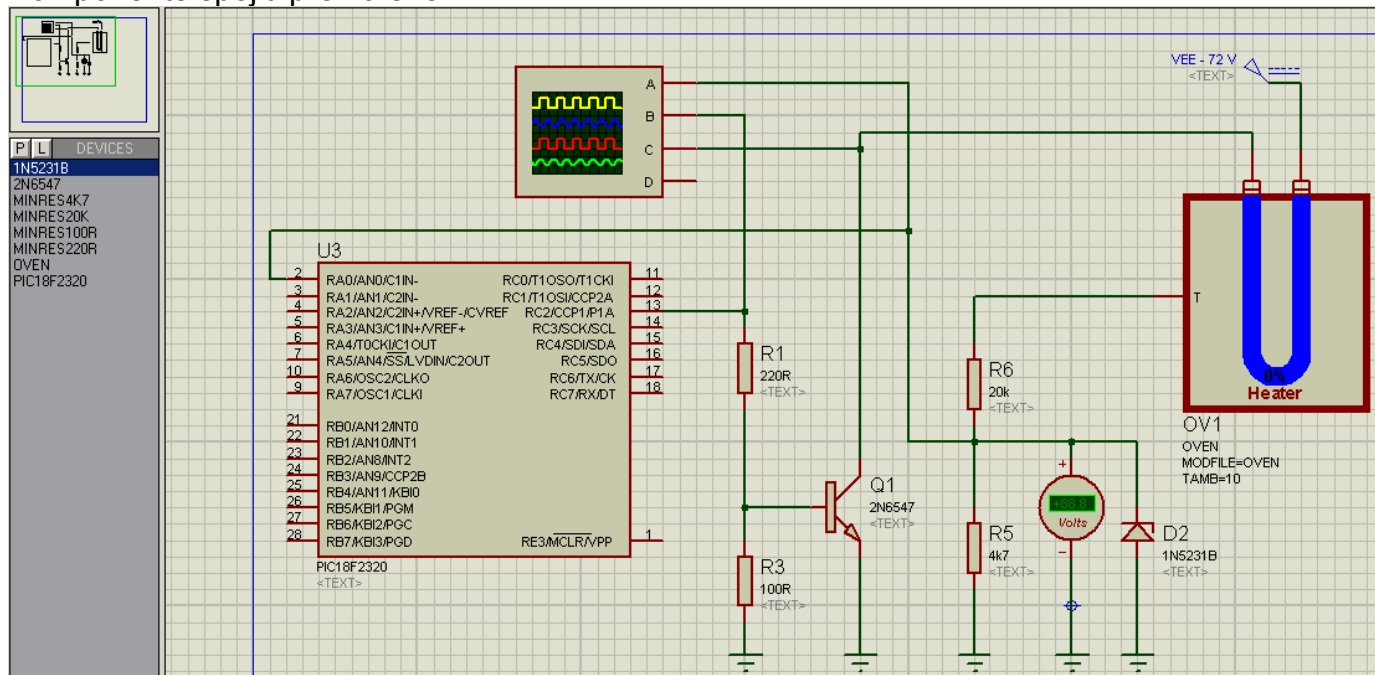
- pročitati poglavlje 14 (pogotovo 14.5) referentnog PIC priručnika (*PICmicro Mid-Range MCU Family Reference Manual*, link na stranicama predmeta)
- pročitati poglavlja 11, 13, 15 i 19 tvorničke dokumentacije mikrokontrolera PIC18F2320 (*PIC18F2220/2320/4220/4320 Data Sheet*, link na stranicama predmeta)
- pročitati poglavlja 2.2, 2.7 i 2.9 dokumentacije biblioteka funkcija prevoditelja C18 (*MPLAB C18 C Compiler Libraries*, link na stranicama predmeta)
- podsjetiti se gradiva trećeg predavanja iz RIP-a (RIP\_RIP, Računalo i proces).
- pročitati poglavlja koja se tiču implementacije PID algoritma iz dokumenta AN964, *Software PID Control of an Inverted Pendulum Using the PIC16F684*, <http://ww1.microchip.com/downloads/en/AppNotes/00964A.pdf>

## 1. Zadatok

Koristeći **PIC18F2320**, treba upravljati peći koja zagrijava prostoriju (primjer iz trećeg predavanja). Proteusova komponenta **Oven** enkapsulira grijač, prostoriju i osjetilo temperature (izlaz T).

Koristeći PID algoritam, treba održavati temperaturu stalnom. Budući da nemamo D/A pretvornik na raspolaganju, treba koristiti PWM signal i varirati popunjenost ovisno o trenutnoj izlaznoj vrijednosti PID algoritma, te na taj način regulirati struju kroz grijač. Konstante ( $K_p$ ,  $K_i$ ,  $K_d$ ) podesiti tako da proces dostiže željenu temperaturu maksimalno brzo i s minimalnim nadvišenjem, te da su odstupanja u stacionarnom stanju također minimalna.

Komponente spojiti prema shemi:



Peć se napaja iz istosmjernog izvora napona 72 V. Na pinu *RC2* generiramo PWM signal, koji se dovodi na naponsko djelilo tranzistorske sklopke Q1, koja uključuje i isključuje grijač. Izlazni napon peći (ovisan o temperaturi) preko naponskog djelila spojen je na pin *RA0* mikrokontrolera. Dioda D2 štiti ulaz mikrokontrolera od napona viših od 5 V.

Parametri peći su (popuniti ih u Proteusu – desni klik na peć, *Properties*, ...):

- temperatura okoline ( $^{\circ}\text{C}$ ) - postaviti  $T_A = 10\text{ }^{\circ}\text{C}$
- termički otpor (*thermal resistance to ambient*,  $^{\circ}\text{C/W}$ ) - postaviti  $T_{RA} = 10\text{ }^{\circ}\text{C/W}$
- vremenska konstanta peći (sec) - postaviti 1 sec
- vremenska konstanta grijača (sec) - postaviti na 3 sec
- temperaturni koeficijent ( $\text{V}/^{\circ}\text{C}$ ) - postaviti  $T_C = 0.1\text{ V}/^{\circ}\text{C}$
- snaga (W) - postaviti  $P = 2\text{ W}$

Izlaznu temperaturu signalizira napon na pinu T peći. Temperaturu želimo održavati na 21 °C, a toj temperaturi odgovara napon na pinu *RA0* od 2.5 V.

Koristiti PID algoritam s predavanja, uzimajući u obzir vremensku komponentu:

```
prosla_greska = 0
I_komponenta = 0
start:
    greska = postavljena_vrijednost - izmjerena_vrijednost
    I_komponenta = I_komponenta + (greska*dt)
    D_komponenta = (greska - prosla_greska)/dt
    izlaz = (Kp * greska) + (Ki*I_komponenta) + (Kd*D_komponenta)
    prosla_greska = greska
    cekaj(dt)
    goto start
```

Umjesto u beskonačnoj petlji, izvoditi PID algoritam u prekidnoj rutini vremenskog sklopa *Timer0*, koji treba podesiti da generira prekid svakih *dt* sekundi. Za *dt* odabrati broj milisekundi koji je višekratnik broja 2, te za množenje i dijeljenje s *dt* koristiti posmak ulijevo i udesno. Najbolji iznos konstante *dt* utvrditi eksperimentalno.

Pri postavljanju popunjenosti, voditi računa o opsegu (popunjenost je 10 bitni broj). Također, pri izračunu voditi računa o vrijednosti / opsegu koji ste odabrali za varijable (*P*, *I*, *D\_komponenta*). Pomak udesno (operator >>) kod C18 nije aritmetički pomak!

Mikrokontroler radi na frekvenciji od 4 MHz. Frekvenciju rada PWM jedinice i parametre A/D jedinice odabrati po volji poštujući ograničenja mikrokontrolera.

### Pulsno-širinska modulacija

Trajanje periode izlaznog PWM signala računa se po formuli:

$$\text{period PWM-a} = [(\mathbf{PR2}) + 1] * 4 * T_{osc} * \text{TMR2 preddjelilo}$$

Trajanje visoke razine signala (u sekundama) računa se po formuli:

$$\text{popunjenost PWM-a} = (\mathbf{CCPR1L:CCP1CON<5:4>}) * T_{osc} * \text{TMR2 preddjelilo}$$

Pomoćne funkcije za rad sa modulima *PWM* i *Timer* navedene su u tablici. Detaljnije opise možete pronaći u dokumentaciji.

Funkcija	Primjer korištenja	Opis
<i>OpenPWM1(char period)</i>	<i>OpenPWM1(0xFF)</i>	Inicijalizira PWM, kanal 1 s periodom
<i>ClosePWM1(void)</i>	<i>ClosePWM1()</i>	Isključuje PWM, kanal 1
<i>SetDCPWM1(unsigned int pp)</i>	<i>SetDCPWM1(0x3FF)</i>	Postavlja popunjenost kanala 1 (10 bita)
<i>OpenTimer2(unsigned char cfg)</i>	<i>OpenTimer2(T2_PS_1_1)</i>	Uključuje i podešava Timer2
<i>CloseTimer2(void)</i>	<i>CloseTimer2()</i>	Isključuje Timer2
<i>ReadTimer2()</i>	<i>T2 = ReadTimer2()</i>	Čita vrijednost brojala Timer2
<i>WriteTimer2()</i>	<i>WriteTimer2(0xFF)</i>	Postavlja vrijednost brojala Timer2

Na predavanjima je prikazan primjer asemblerskog programa koji kontinuirano smanjuje popunjenost signala mikrokontrolera *PIC18F2320*.

Isti primjer napisan u jeziku C:

```
#include <p18f2320.h>
#include <timers.h>
#include <pwm.h>

#pragma config WDT = OFF
#pragma config PWRT= ON
#pragma config MCLRE=OFF

unsigned long dcPWM1;
```

```

void main(void)
{
    OpenPWM1(100);           // PR2 = 100
    dcPWM1 = 320;            // 320=80<<2, koristimo samo CCPR1L
    SetDCPWM1(dcPWM1);       // CCPR1L = 80 (gornjih 8 bita CCPR1)

    OpenTimer2(TIMER_INT_OFF & T2_PS_1_4); // Ukljuci timer2, preddjelilo = 4

    do
    {
        Delay10TCYx(28);     // pricekaj dok Timer2 skoro ne završi
        dcPWM1 = dcPWM1 - 40; // CCPR1L -= 10 (40=10<<2)
        SetDCPWM1(dcPWM1);    // promijeni popunjenost
    }
    while(dcPWM1 != 0);

    ClosePWM1();             // Iskljuci PWM
    CloseTimer2();           // Iskljuci Timer2
}

```

U konfiguracijskim bitovima sklopa *Timer2* postavili smo preddjelilo na 1:4, a za period PWM modula 100 taktova sklopa *Timer2*.

**Napomena:** u ovom primjeru koristi se samo registar *CCPR1L*, pa se zato faktor popunjenosti (*duty cycle*) množi sa 4 (posmak ulijevo za 2) kod konfiguracije PWM modula da bi se vrijednost upisala u registar *CCPR1L*. Faktor popunjenosti je 10-bitni broj zapisan u paru registara *CCPR1L<7:0>:CCP1CON<5:4>* - u registru *CCPR1L* zapisano je 8 najznačajnijih bitova, dva preostala najmanje značajna bita treba zapisati u bitove 5 i 4 registra *CCP1CON*.

Pomoćne funkcije koje definira C18 olakšavaju upis vrijednosti faktora popunjenosti preko dva registra.

Detalje o konfiguraciji modula *Timer2* funkcijom *OpenTimer2()* pročitajte u dokumentaciji biblioteke funkcija C18, str. 60.

### Podešavanje PID regulatora

Pri podešavanju PID regulatora za bilo koji sustav, do željenog odziva najlakše je doći sljedećim postupkom:

1. Na osnovu odziva sustava u otvorenoj petlji, utvrditi koji parametre treba poboljšati
2. Dodati proporcionalnu komponentu – skratiti vrijeme porasta
3. Dodati derivacijsku komponentu – smanjiti nadvišenje
4. Dodati integracijsku komponentu – smanjiti grešku u stacionarnom stanju
5. Podesiti konstante (*Kp*, *Kd*, *Ki*) dok se ne dobije željeni odziv

Često nije ni potrebno implementirati sve tri komponente (P, I i D), ako jednostavniji algoritam (npr. samo PI ili samo PD) daje dobre rezultate.