Sveučilište u Rijeci
Tehnički fakultet
Preddiplomski sveučilišni studij računarstva

Ugradbeni računalni sustavi

Završni projekt

Vedran Matić
Borna Sila
Luka Županović

Rijeka, travanj 2021.

Sadržaj

1. Uvod

Za završni projekt iz ovog kolegija ideja nam je bila simulirati mikrokontroler za upravljanje rashladnim uređajima. Korisnik pomoću definiranog izbornika, koji će detaljnije biti opisan u idućem poglavlju, ovisno o svojim željama bira željenu temperaturu odnosno želi li pokrenuti hlađenje ili grijanje ili pak želi balansiranje trenutne temperature unutar postavljene temperaturne razlike. Ukoliko se korisnik odluči za hlađenje, pokreće se mali ventilator (kojim upravlja DC motor), dok se odabirom grijanja pali žarulja od 0.75 A. Pritom smo za očitavanje temperature koristili tmp35 temperaturni senzor iz kutije. Od komponenti nam je, uz još žice i otpornike, bio potreban i LCD preko kojeg se ostvaruje interakcija s korisnikom. Za upravljanje žaruljom i DC motorom koristili smo 5V relaj, pošto se i na referentnom uređaju koristi relaj (na 220V), te tranzistor uz diodu koji je služio kao prekidač. I naravno, korišten je mikrokontroler Atmega16, osnova cjelokupnog projekta. Atmega16 je 8-bitni mikrokontroler s 32 8-bitna registra opće namjene temeljen na AVR RISC arhitekturi s 1KB SRAM memorije i 16 KB samoprogramirajuće Flash memorije od čega je iskorištenost bila 1/3. Uređaj kojim smo se vodili u izradi ovog projekta, pokušajući ga simulirati, je IDPlus 902/961.

2. Opis projekta

Postoji 5 osnovnih stanja odnosno *displaya* (u kodu varijabla *dMode*):

- početno stanje u kojem se ispisuje početna poruka
- stanje za prikaz temperature
- stanje za konfiguriranje načina rada(poziv showMenu funkcije)
- postavljanje lozinke
- provjera lozinke

Stanja se mijenjuju generiranjem vanjskog prekida INT0. Temperatura sa senzora se čita sa PORTA (PA0). Očitavanje temperature nije bio prevelik izazov s obzirom na znanja stećena na kolegiju. Valja napomenuti kako, s obzirom da temperatura izmjerena senzorom varira, koristili smo usrednjavanje vrijednosti(uzimali smo posljednja 32 uzorka temperature i uzeli njihovu srednju vrijednost).

```
typedef struct{
  int8_t    samIdx;
  uint32_t sum;
  uint16_t samples[TOT_SAMPLES];
}movAvg_t;
```

Navedena struktura inicijalizirana je funkcijom init_temp_ma(). *Display* na LCD-u se osvježuje svake sekunde generiranjem prekida (CTC način rada). Osnovni i jedan od najizazovnijih

zadataka u ovom projektu bio je implementacija *showMenu* funkcije. Ona započinje provjerom zastavice *submenu*. Ukoliko korisnik nije ni u jednom submeniju ispisuje se u kojem je načinu rada izbornik. Tri su načina rada moguća: varijable, mode i alarmi, a korisnik bira željeni pritiskom na tipku KEY1. Ukoliko je pak u sporednom izborniku onda provjeravamo u kojem je submeniju. Korisnik u submeni ulazi pritiskom gumba KEY2. Ukoliko je u submeniju za varijable na LCD-u se prikaže trenutna varijabla na kojoj se korisnik nalazi. Detaljniji opis varijabli sa zadanim vrijednostima dan je u nastavku.

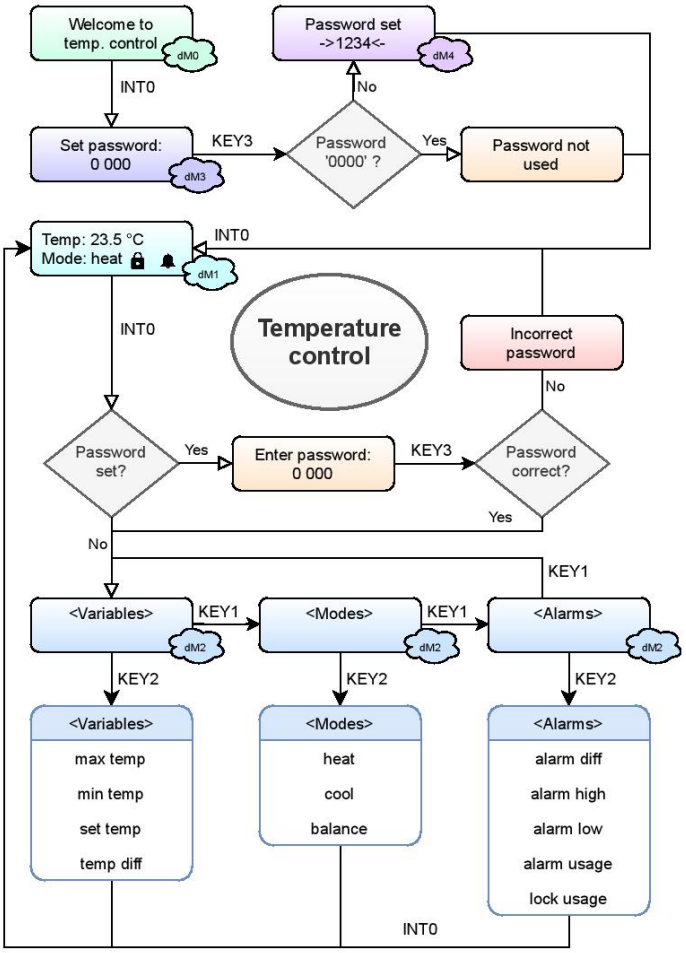
- variables[0] - maksimalna *set* temperatura (*99°C*)
- variables[1] - minimalna *set* temperatura (*0°C*)
- variables[2] - željena temperatura (*0°C*)
- variables[3] - temperaturna razlika na kojoj se pali grijanje/hlađenje (*2°C*)

Ukoliko je u submeniju za mode onda se prikažu trenutni modalitet rada, te ukoliko je u alarmima onda se prikaže trenutni alarm na kojem se korisnik nalazi. Detaljniji opis alarma sa zadanim vrijednostima dan je u nastavku.

- alarms[0] - temperaturna razlika na kojoj se pali alarm (*99°C*)
- alarms[1] - gornja temperaturna granica (*50°C*)
- alarms[2] - donja temperaturna granica (*0°C*)
- alarms[3] - zastava za korištenje alarma (*on*/***off***)
- alarms[4] - zastava za korištenje zaključavanja (*on*/***off***)

S obzirom da su isti gumbovi za mijenjanje varijable i za prolazak kroz izbornik, koristimo *mSelect* pomoćnu zastavicu. Ukoliko je ona postavljena (1) onda se varijabla na kojoj se korisnik nalazi mijenja. Drugi način rada je tzv. *Modes* submeni. Oni su spremljeni u polje u kodu nazvano *mode*. Moguće vrijednosti su ovdje, kao što je već rečeno heating, cooling i balance. Posljednji način rada su alarmi. Također, ukoliko je *mSelect* postavljen korisniku

je omogućeno mijenjanje. Sljedeća slika radnog dijagrama dodatno pojašnjava funkcioniranje našeg projekta te promjene stanja na LCD zaslonu.



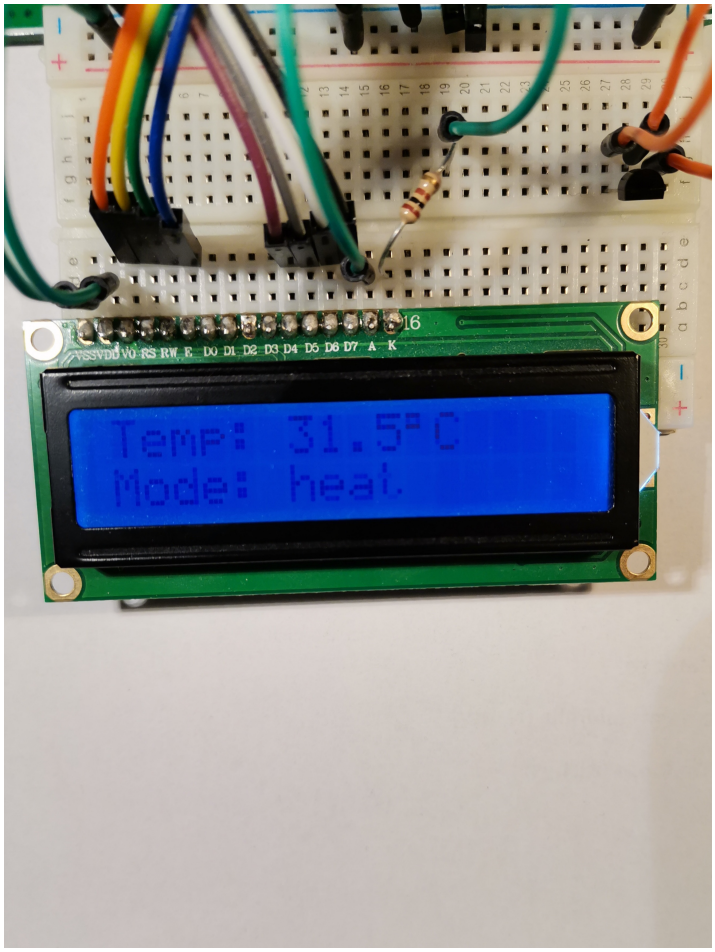
Slika 2.1: *Radni dijagram sustava za upravljanje temperaturom*

Kako bi se tijekom rada onemogućilo slučajno mijenjanje postavki, uobičajeno je da ovakvi upravljači imaju lozinku. Ukoliko ju korisnik postavi, mijenjanje postavki zahtjeva unos lozinke. Lozinka se postavlja nakon prvog pritiska INT0 ($fMode = 3$ u ISR za INT0). Sljedeći isječak koda prikazuje našu implementaciju funkcije za postavljanje lozinke.

```
void setPsw() {
  if (!pswSet) {
    lcd_clrscr();
    lcd_gotoxy(1, 0);
    lcd_puts("Set password:");
    lcd_gotoxy(4, 1);

    for (uint8_t i = 0; i < 4; i++){
      if (mVar == i) {
        lcd_putc(mSelect ? '<' : ' ');
        lcd_putc(password[i]);
        lcd_putc(mSelect ? '>' : ' ');
      } else lcd_putc(password[i]);
    }
  } else if (pswUse) {
    lcd_clrscr();
    lcd_gotoxy(2, 0);
    lcd_puts("Password set");
    lcd_gotoxy(4, 1);
    lcd_puts("->");
    for (uint8_t i = 0; i < 4; i++){
      lcd_putc(password[i]);
    }
    lcd_puts("<-");
  } else {
    lcd_clrscr();
    lcd_gotoxy(2, 0);
    lcd_puts("Password not");
    lcd_gotoxy(6, 1);
    lcd_puts("used");
  }
}
```

Slika 2.2: *Početno stanje*



Slika 2.3: *Stanje za prikaz temperature*

2.1 lcd.h

Ovu knjižicu smo koristili zbog mnoštva predefiniranih korisnih funkcija. One koje smo mi koristili jesu lcd_init() za inicijalizaciju LCD-a prije korištenja, lcd_clrscr() za brisanje teksta na ekranu,lcd_putc(char) i lcd_puts(string) za ispis charactera i stringa na ekran te lcd_gotoxy(x,y) kojom definiramo na kojoj poziciji na ekranu želimo nešto ispisati. Također, ovdje smo definirali izlazne pinove LCD-a na sljedeći način:

```
#define LCD_PORT          PORTB        /**< port for the LCD lines   */
#define LCD_DATA0_PORT    LCD_PORT     /**< port for 4bit data bit 0 */
#define LCD_DATA1_PORT    LCD_PORT     /**< port for 4bit data bit 1 */
```

```
4  #define LCD_DATA2_PORT   LCD_PORT     /**< port for 4bit data bit 2 */
5  #define LCD_DATA3_PORT   LCD_PORT     /**< port for 4bit data bit 3 */
6  #define LCD_DATA0_PIN    4            /**< pin for 4bit data bit 0  */
7  #define LCD_DATA1_PIN    5            /**< pin for 4bit data bit 1  */
8  #define LCD_DATA2_PIN    6            /**< pin for 4bit data bit 2  */
9  #define LCD_DATA3_PIN    7            /**< pin for 4bit data bit 3  */
10 #define LCD_RS_PORT      PORTD     /**< port for RS line         */
11 #define LCD_RS_PIN       5            /**< pin  for RS line         */
12 #define LCD_RW_PORT      PORTD     /**< port for RW line         */
13 #define LCD_RW_PIN       6            /**< pin  for RW line         */
14 #define LCD_E_PORT       PORTD     /**< port for Enable line     */
15 #define LCD_E_PIN        7            /**< pin  for Enable line     */
```

2.2 Hardver i spajanje

Temperaturni senzor spojen je, kao što je već rečeno na PORTA, LCD je spojen na PORTD (PD4-PD7) za prijenos naredbi te PORTB (PB4-PB7) za prijenos podataka. PORTB je također korišten i za gumbove. Na PORTA su spojeni i izlazi za žarulju, ventilator te zujalicu (alarm) redom od PA1-PA3.

Žarulja je spojena na bateriju s relejem kao prekidačem unutar strujnog kruga. Spoj je na COM i NO kako bi strujni krug bio zatvoren tek kada relej dobije signal s mikrokontrolera. Za prekidač DC motora korišten je 2n2222 tranzistor koji može podnijeti do 1A struje te dioda koja spriječava povratak struje u mikrokontroler kako nebi došlo do oštećenja. U istu svrhu mogla se koristiti i IC l298. Za rad zujalice je korišten otpornik od 100

3. Zaključak

Cilj projekta bio je produbiti znanja stečena na kolegiju Ugradbeni računalni sustavi. Tijekom izrade ovog projekta susreli smo se s brojnim izazovima koji su nam pomogli da pobolje upoznamo rad s mikrokontrolerom te nekim dodatnim komponentama kao šta je relay i zujalica. Također smo proširili poznavanje rada LCD zaslona implementirajući vlastite simbole (ikone) za prikaz. Kroz timski rad uspjeli smo prebroditi izazove na koje smo naišli i simulirati navedeni uređaj na način na koji smo prvotno zamislili.