بنام خدا

پروژه ترمستات دیجیتال

(مدار کنترل دمای محیط با میکروکنترلر AVR)

نگارنده : میلاد جهاندیده

Melec.ir

مقدمه

پروژه ترمستات یا کنترل دمای محیط یک پروژه کاملا کاربردی هست که برای ثابت نگه داشتن دمای محیط در یک بازه مشخص طراحی شده است.

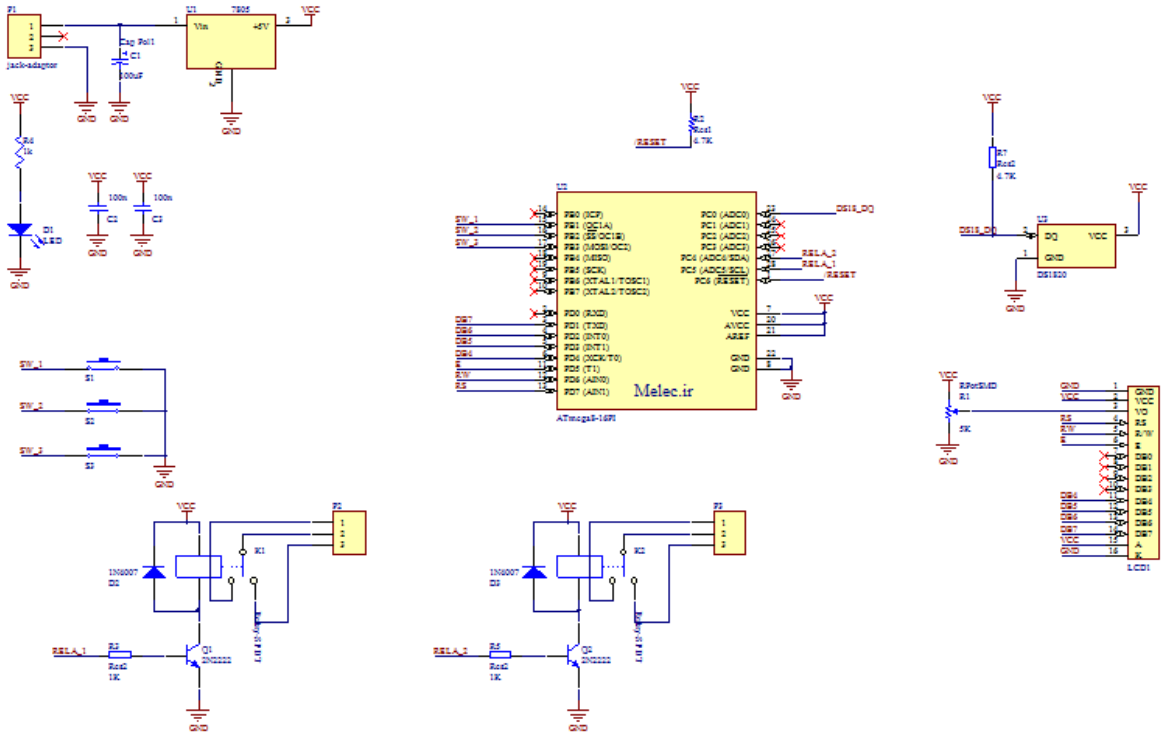
در این پروژه از یک سنسور دما برای اندازه گیری دما ، میکروکنترلر برای کارهای پردازشی ، دو عدد رله یکی برای فن و دیگری برای هیتر و یک عدد نمایشگر برای نمایش دما ، سه عدد پوش باتن برای تنظیم مقدار ماکزیمم و مینمیم دما طراحی شده است.

سنسور این پروژه DS18B20 می‌باشد ولی یک ویژگی خوب این پروژه این هست که با تغییر کوچک در سخت افزار و تغییر کد میکروکنترلر می توانید سنسور LM35 را نیز روی پروژه سوار کنید. نحوه انجام کار در ادامه توضیح داده خواهد شد.

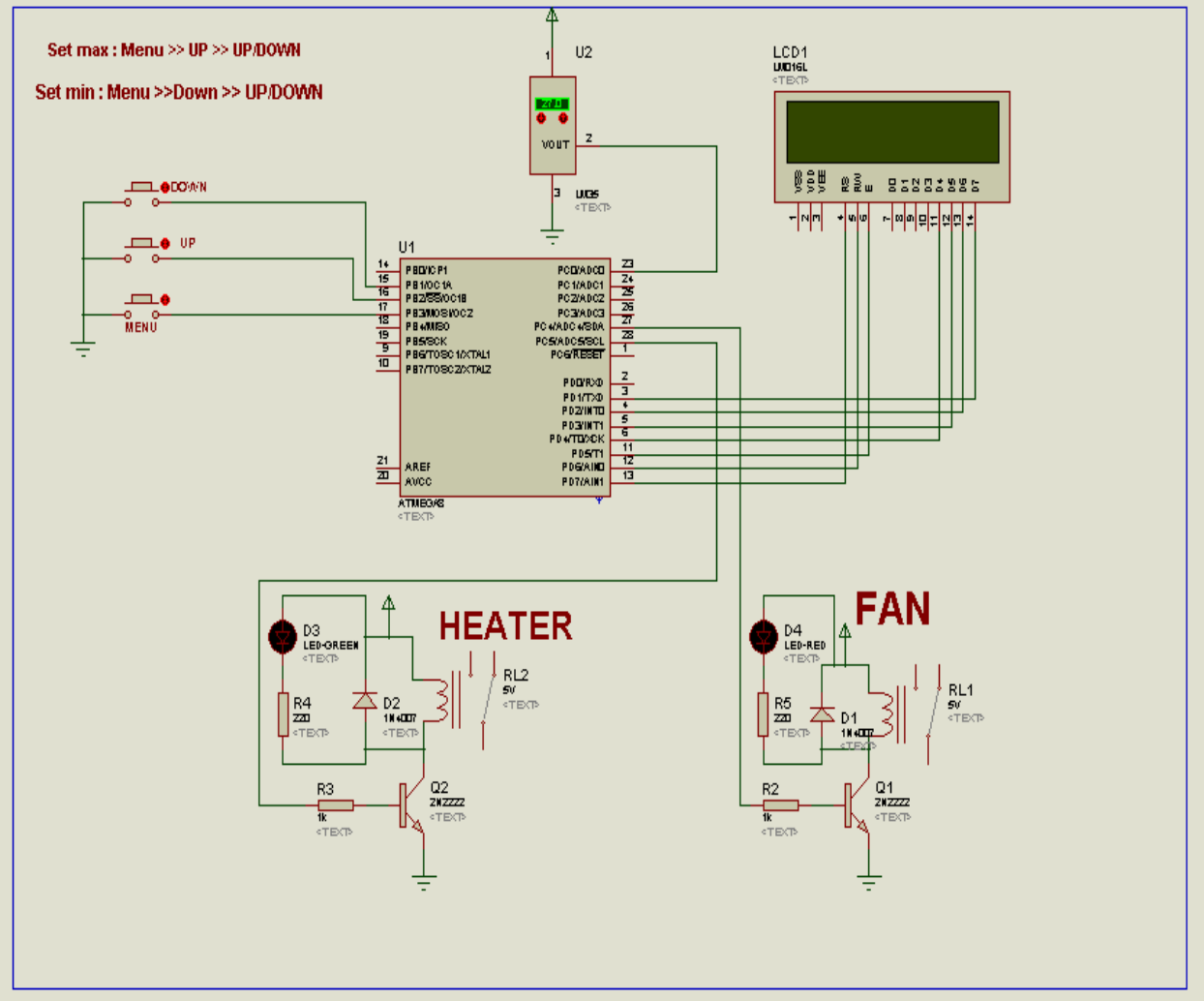
تذکر: شماتیک بصورت با کیفیت در فایل جداگانه بصورت PDF ضمیمه شده است.

فایل های ضمایم : فایل شماتیک،فایل شبیه سازی پرتئوس ،برنامه نویسی، فایل PCB

شماتیک پروژه



شبیه سازی و برنامه میکروکنترلر



توضیحات برنامه نویسی پروژه برای سنسور **DS18B20**

#include <mega8.h>

فراخوانی کتابخانه میکروکنترلر ATMEGA8

#include <delay.h>

فراخوانی کتابخانه ایجاد تاخیر

```
#include <stdio.h>
```

فراخوانی کتابخانه استاندارد ورودی و خروجی های زبان C

```
#include <alcd.h>
```

فراخوانی کتابخانه نمایشگر LCD 2*16

```
#include <1wire.h>
```

فراخوانی کتابخانه ارتباط یک سیمه برای سنسور

```
#include <ds1820.h>
```

فراخوانی کتابخانه DS18B20

```
#define MAX_DS1820     1
```

حداکثر تعداد سنسور های DS18B20 موازی شده (این سنسور ها را میتوان تا 128 تا با هم به صورت موازی به میکروکنترل وصل کرد)

```
unsigned char ds1820_devices;
```

متغیر تعداد سنسور های موجود روی پروژه

```
unsigned char ds1820_rom_codes[MAX_DS1820][9];
```

آرایه دو بعدی نگهداری کدهای انحصاری هر سنسور

```
signed char min_temp,max_temp;
```

متغییر حداقل و حداکثر دما

```
char setting(void);//setting menu
```

تابع تنظیمات دما

```
char s_max(void);  //max temp setting
```

تابه تنظیم مقدار حداکثر

```
char s_min(void);  //min temp setting
```

تابع تنظمیم مقدار حداقل

```
void main(void){
```

تابع اصلی پروژه

```
unsigned char lcd_buffer[17],lcd_buffer_2[17];
```

آرایه های نگهداری اطلاعاتی که قرار است روی ال_سی_دی چاپ شود برای هر سطر یک آرایه که رشته هم گفته می شود.

```
Long temp=0;
```

متغییر نگهداری دما

```
w1_init();
```

راه اندازی اولیه ارتباط تک سیمه

```
ds1820_devices=w1_search(0xf0,ds1820_rom_codes);
```

تابع سرچ سنسور های متصل

این تابع تعداد سنسور های متصل روی پایه میکرو را سرچ یا اسکن میکند و تعداد سنسور ها را بهمراه کد آنها برمیگرداند.

```
DDRC=(1<<DDC4)|(1<<DDC5);
```

پایه های رله ها بصورت خروجی

```
DDRB=(0<<DDB1)|(0<<DDB2)|(0<<DDB3);
```

پایه های کلید ها یا شستی ها بصورت ورودی

```
PORTB.1=1;PORTB.2=1;PORTB.3=1;
```

پول آپ پایه های ورودی یا شستی ها روشن

```
lcd_init(16);
```

راه اندازی ال_سی_دی

```
lcd_clear();
```

پاک کردن ال_سی_دی

```
lcd_putsf("Termostat Pr");
```

نوشتن نوشته Termostat Pr روی ال سی دی

```
delay_ms(500);
```

تاخیر 500 میلی ثانیه تا نوشته روی نمایشگر دیده شود

```
sprintf(lcd_buffer, "%d Sensors Found", ds1820_devices);
lcd_clear();
lcd_puts(lcd_buffer);
```

در سه خط بالا ابتدا نوشته ها فرمت بندی یا قالب بندی می شود یعنی حروف و اعداد و نحوه نمایش قالب بندی می شود و در سطر بعدی نمایشگر پاک و در سط سوم نوشته نشان داده می‌شود.

```
delay_ms(1000);
```

تاخیر یک ثانیه ای برای نمایش تعداد سنسور های یافته شده

```
min_temp=20;
max_temp=30;
```

تعیین مقدار حداقل و حداکثر دما بصورت پیش فرض

```
temp = ds1820_temperature_10(ds1820_rom_codes[0]);
```

خواندن دما از سنسور

```
while(1){
```

حلقه بینهایت

```
temp = ds1820_temperature_10(ds1820_rom_codes[0]);
```

خواندن دما از سنسور

```
sprintf(lcd_buffer, "  T= %d.%d%cC",temp/80,temp%80,223);
```

قالب بندی اطلاعات سطر اول ال_سی_دی

```
sprintf(lcd_buffer_2," %d%cC <T<
%d%cC",min_temp,223,max_temp,223);
```

قالب بندی اطلاعات سطر دوم ال_سی_دی

```
lcd_clear();
lcd_puts(lcd_buffer);
delay_ms(1);
lcd_gotoxy(0,1);
lcd_puts(lcd_buffer_2);
```

در چند سطر بالا اطلاعات روی نمایشگر نشان داده می شود که در خطوط بالا تک تک خطوط توضیح داده شد.

```
If((temp/80)>max_temp){  // FAN
```

```
PORTC.4=1;
 }
else{
PORTC.4=0;
}
```

شرط در صورت بالا بودن دما

```
if((temp/80)<min_temp){   // HEATER
PORTC.5=1;
}
else{
PORTC.5=0;
}
```

شرط در صورت پایین بودن دما

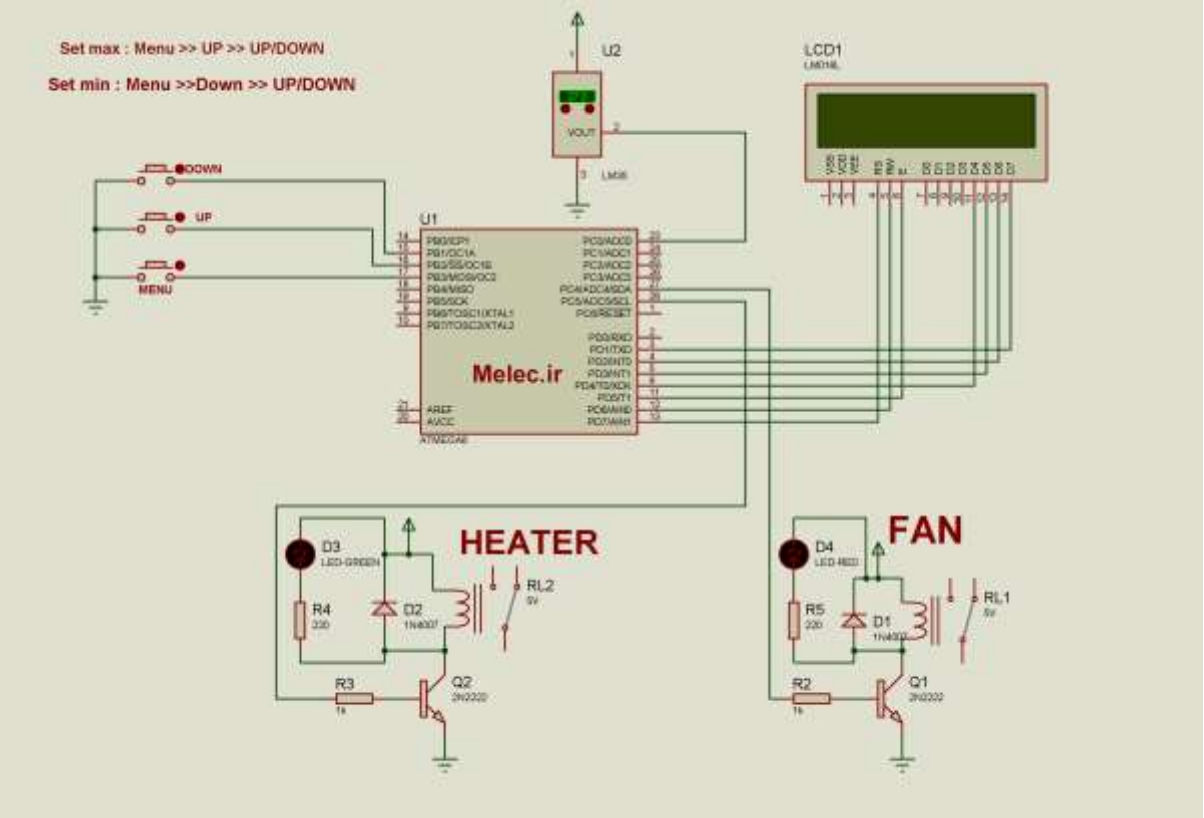
```
if(PINB.3==0){ // setting pin
while(PINB.3==0);
setting();
   }
```

شرط فشرده شدن شستی تنظیمات که در این صورت توابع تنظیمات اجرا می شود.

```
 }
```
پایان حلقه بینهایت که تا زمانی که میکرو روشن هست این حلقه اجرا می شود

```
}
```
پایان تابع اصلی برنامه

شبیه سازی با سنسور LM35

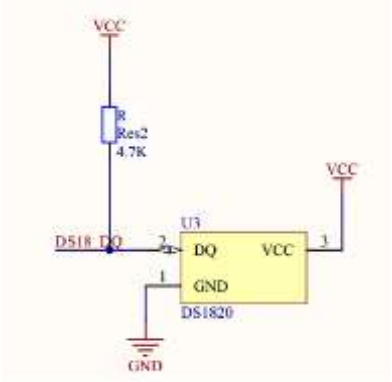


چطوری از سنسور LM35 به جای سنسور DS18B20 در پروژه

همانطور که در مقدمه هم گفتیم سسنور اصلی پروژه سنسور DS18B20 است. در اینجا ما هر کد را توضیح دادیم و بنا به نیازتون از هر دو می توانید استفاده و آن را تغییر دهید و همچنین فایل پروتئوس و کد هر دو پروژه در فایل های ضمیمه موجود است ولی شماتیک و PCB کشیده شده برای DS18B20 هست ولی در هنگام منتاژ با تغییرات زیر میتوانید سنسور LM35 را به جای DS18B20 جا بزنید.

مرحله اول

روی برد یک مقاومت پول آپ برای DS18B20 هست که در LM35 نیازی به این مقاومت نیست و باید آن را منتاژ نکنید پس مقاومت کنار سنسور را که مقدارش 4.7K هست را منتاژ نکنید یا اگر منتاژ کردید قطع کنید.



مرحله دوم

پایه های مثبت و منفی سنسور LM35 و DS18B20 برعکس هم هستن ولی دیتا هر دو پایه وسط هست که فرقی نمی کند. پس باید در نصب روی برد دقت کنید که معکوس DS18B20 باید نصب کنید حتما حتما با ولتمتر تست علامت گذاری بعد نصب کنید.

مرحله سوم

تغییر کد میکروکنترلر ، باید کد مخصوص LM35 که در ضمایم آمده است را روی آیسی پروگرام کنید.

توضیح برنامه نویسی برای سنسسور LM35

سعی شده برای هر خط برنامه نویسی یک جمله کلیدی نوشته شود تا بتوانید آن را درک یا تغییر دهید.

```
#include <mega8.h>
```

فراخوانی کتابخانه مگا8

```
#include <delay.h>
```

فراخوانی کتابخانه ایجاد تاخیر

```
#include <stdio.h>
```

فراخوانی کتابخانه ورودی و خروجی های استاندارد زبان C

```
#include <alcd.h>
```

فراخوانی کتابخانه نمایشگر LCD

```
#define ADC_VREF_TYPE 0x60
```

تعیین نوع رفرنس ADC میکروکنترلر

```
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
input voltage
delay_us(10);
ADCSRA|=0x40;
while ((ADCSRA & 0x10)==0);
```

```
ADCSRA|=0x10;
return ADCH;
}
```

تابع بالا برای خواندن مقدار ADC نوشته شده است و بطور کلی وقتی فراخوانی می شود این تیکه کد اجرا و مقدار ولتاژ روی پایه ADC میکروکنترلر را قرائت و به برنامه اصلی باز میگرداند.س

```
signed char min_temp,max_temp;
```

متغیر برای نگه داشتن مقدار حداقل و حداکثر دما

```
char setting(void);//setting menu
```

تابع تنظیمات

```
char s_max(void);  //max temp setting
```

تابع تنظیم مقدار حداکثر

```
char s_min(void);  //min temp setting
```

تابع تنظیم مقدار حداقل

```
void main(void){
```

تابع اصلی برنامه ما همانطور که میدانید برنامه های زبان C یک تابع اصلی به نام main دارند که برنامه از این قسمت شروع به اجرا میکند.

```
unsigned char lcd_buffer[17],lcd_buffer_2[17];
```

تعریف دو آرایه برای بافر های LCD یا به عبارتی نوشته های روی ال_سی_دی قبل از اینکه چاپ شوند در داخل این آرایه ها نگه داری می شوند.

```
long temp=0;
```

متغیر برای نگه داشتن مقدار دما

```
DDRC=(1<<DDC4)|(1<<DDC5);
```

پایه های رله ها به عنوان خروجی

```
DDRB=(0<<DDB1)|(0<<DDB2)|(0<<DDB3);
PORTB.1=1;PORTB.2=1;PORTB.3=1;
```

پایه های شستی ها به عنوان ورودی

```
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x86;
```

تنظیم رجیستر های ADC

```
lcd_init(16);
```

آماده سازی یا راه اندازی اولیه ال_سی_دی

```
lcd_clear();
```

پاک کردن ال سی دی

```
lcd_putsf("Termostat");
delay_ms(1);
```

تاخیر یک میلی ثانیه

```
lcd_gotoxy(0,1);
```

سطر دوم ال سی دی

```
lcd_putsf("Melec.ir");
delay_ms(2000);
```

چند خط بالا برای نمایش اسم روی نمایشگر است.

```
min_temp=20;
max_temp=30;
```

تعیین پیش فرض مقدار ماکزیمم و مینیمم

```
while(1){
```

حلقه بینهایت

```
temp = read_adc(0);
temp = temp*2 ;
```

خواندن دما از سنسور ، پایه خروجی سنسور به پایه صفر ADC وصل شده است.

علت اینکه مقدار خوانده شده ضرب در دو می شود همان تابع تبدیل ولتاژ به دما سنسور می باشد. که با توجه به مقدار رفرنس و هشت بیتی خواندن ADC و غیره حساب میشود .

```
sprintf(lcd_buffer, "  T= %d%cC",temp,223);
```

سطر اول ال سی دی دما را نمایش میدهیم

```
sprintf(lcd_buffer_2," %d%cC <T<
%d%cC",min_temp,223,max_temp,223);
```

سطر دوم مقدار ماکزیمم و مینمم

```
lcd_clear();
```

پاک کردن ال سی دی

```
lcd_puts(lcd_buffer);
```

ارسال اطلاعات سطر اول به ال سی دی

```
delay_ms(1);
```

تاخیر یک میلی ثانیه

```
lcd_gotoxy(0,1);
```

رفتن به سطر دوم ال سی دی

```
lcd_puts(lcd_buffer_2);
```

قرار دادن اطلاعات سطر دوم ال سی دی

```
delay_ms(100);
```

تاخیر 100 میلی ثانیه

```
if((temp)>max_temp){  // FAN
PORTC.4=1;
}
   else{
   PORTC.4=0;
    }
```

شرط اینکه اگر دما بیشتر از دمای ماکزیمم باشد که در این صورت فن روشن می شود.

```
if((temp)<min_temp){   // HEATER
PORTC.5=1;
}
 else{
```

```
  PORTC.5=0;
  }
```

شرط اینکه دما کمتر از دمای مینیمم باشد که در این صورت هیتر روشن می شود.

```
if(PINB.3==0){ // setting pin
while(PINB.3==0);
setting();
}
```

شرط اینکه کلید تنظیمات زده شود.

```
   }
```

پایان حلقه بینهایت

```
}
```

پایان تابع اصلی