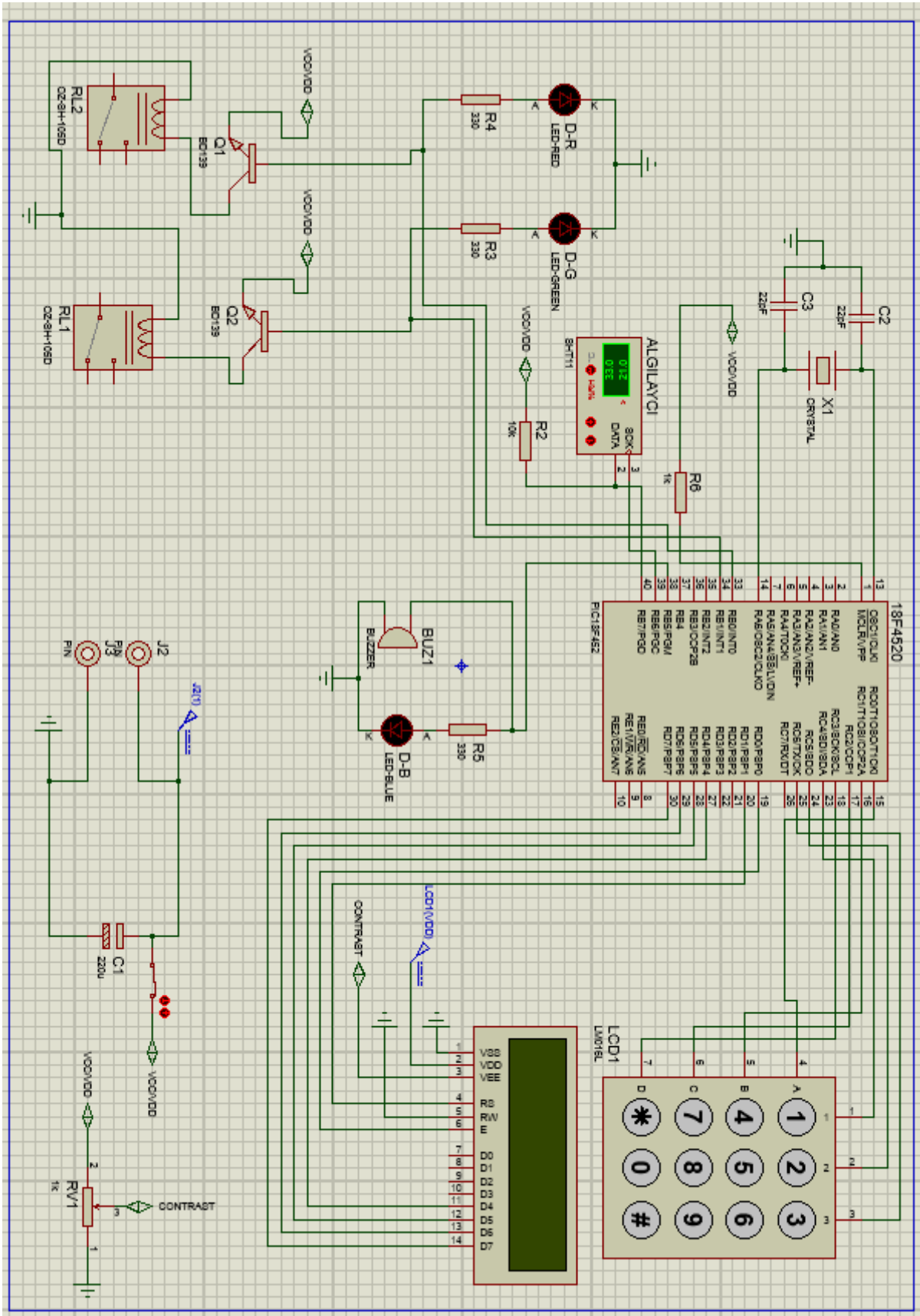
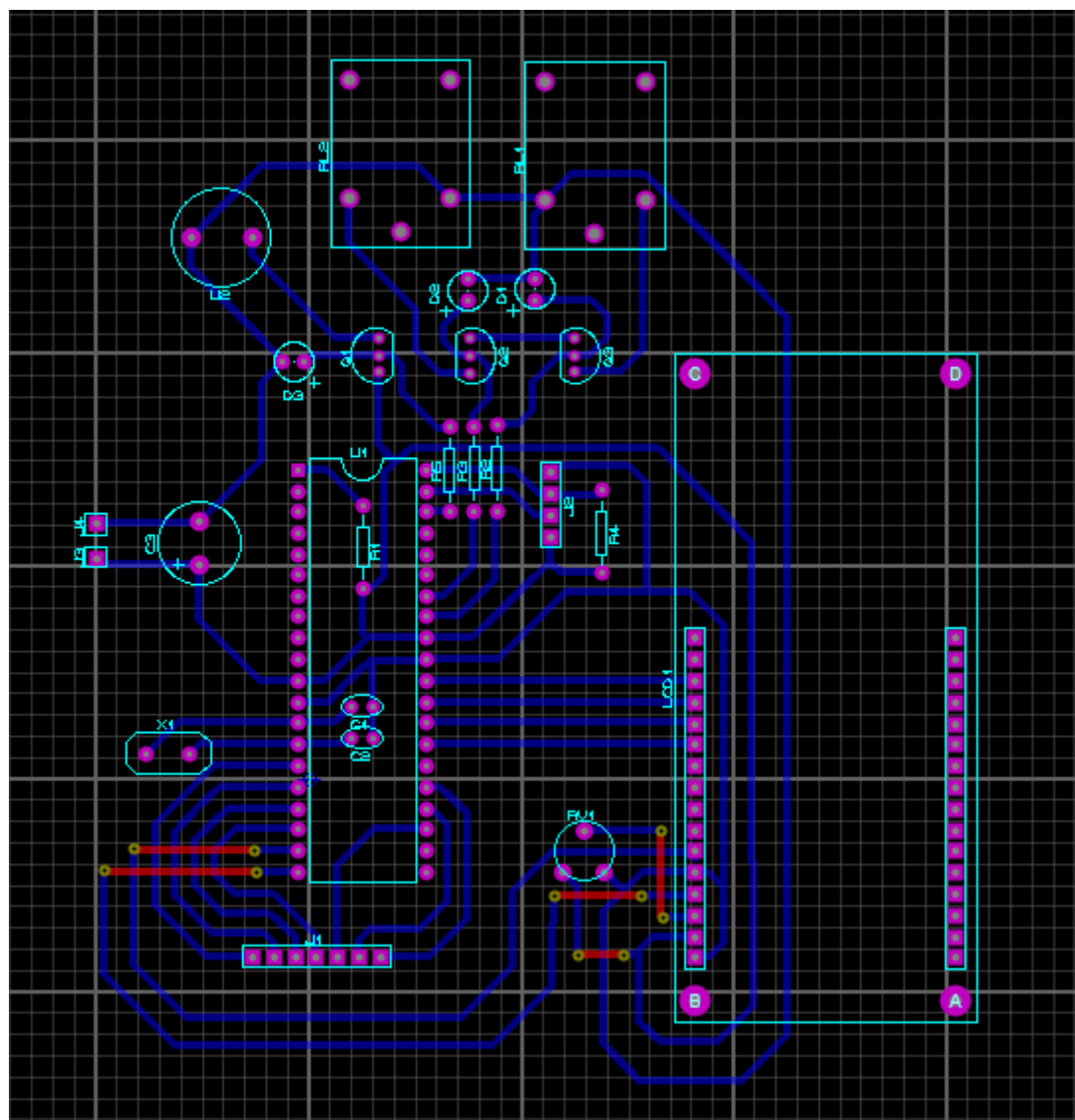


EK-1 (Devrenin Proteus Çizimi)





EK-2 (Kaynak Kod)

```
#include <18F452.h>
#device adc=8

#FUSES NOWDT //No Watch Dog Timer
#FUSES WDT128 //Watch Dog Timer uses 1:128 Postscale
#FUSES XT //Crystal osc <= 4mhz for PCM/PCH , 3mhz
to 10 mhz for PCD
#FUSES NOPROTECT //Code not protected from reading
#FUSES NOOSCSEN //Oscillator switching is disabled, main
oscillator is source
#FUSES NOBROWNOUT //No brownout reset
#FUSES BORV20 //Brownout reset at 2.0V
#FUSES NOPUT //No Power Up Timer
#FUSES NOSTVREN //Stack full/underflow will not cause
reset
#FUSES NODEBUG //No Debug mode for ICD
#FUSES NOLVP //No low voltage prgming, B3(PIC16) or
B5(PIC18) used for I/O
#FUSES NOWRT //Program memory not write protected
#FUSES NOWRTD //Data EEPROM not write protected
#FUSES NOWRTB //Boot block not write protected
#FUSES NOWRTC //configuration not registers write
protected
#FUSES NOCPD //No EE protection
#FUSES NOCPB //No Boot Block code protection
#FUSES NOEBTR //Memory not protected from table reads
#FUSES NOEBTRB //Boot block not protected from table
reads

#use delay(clock=4000000)

#include "SHT11.c"
#include "new_lcd.c"

//3x4 tuş takımı
#define sat1 pin_c0
#define sat2 pin_c1
#define sat3 pin_c2

#define sat4 pin_c3
#define sut1 pin_c4
#define sut2 pin_c5
#define sut3 pin_c6

// ısıtıcı ve soğutucu
#define isitici pin_b0
#define sogutucu pin_b1

//sht11 algılayıcı
#define sht_data_pin PIN_b7// (Data pin veya input)
#define sht_clk_pin PIN_b6

// sensor kontrol
#define ses_sensor pin_b5

float temp=0,humid=0,sicaklik=0,nem=0;
int i=1, negatif=0, flag=0, derece=25, tus=99, pres = 99;

int anticounter=10000, counterpid=0, flagmode=0, mode=0;
```

```

//fonksiyonlar
int keypad_oku() // Fonksiyon ismi
{
    output_c(0x00); // D portu çıkışı sıfırlanıyor

    output_high(sat1);
    if (input(sut1))
        { delay_ms(20); tus=1;output_low(sat1);}
    else if (input(sut2))
        { delay_ms(20); tus=2;output_low(sat1);}
    else if (input(sut3))
        { delay_ms(20); tus=3;output_low(sat1);}
    else
        output_low(sat1);

    output_high(sat2);
    if (input(sut1))
        { delay_ms(20); tus=4;output_low(sat2);}
    else if (input(sut2))
        { delay_ms(20); tus=5;output_low(sat2);}
    else if (input(sut3))
        { delay_ms(20); tus=6;output_low(sat2);}
    else
        output_low(sat2);

    output_high(sat3);
    if (input(sut1))
        { delay_ms(20); tus=7;output_low(sat3);}
    else if (input(sut2))
        { delay_ms(20); tus=8;output_low(sat3);}
    else if (input(sut3))
        { delay_ms(20); tus=9;output_low(sat3);}
    else
        output_low(sat3);

    output_high(sat4);
    if (input(sut1))
        { delay_ms(20); tus=10;output_low(sat4);}
    else if (input(sut2))
        { delay_ms(20); tus=0;output_low(sat4);}
    else if (input(sut3))
        { delay_ms(20); tus=19;output_low(sat4);}
    else
        output_low(sat4);

    return tus;
}

void allkeypadlow()
{
    output_low(sat1);
    output_low(sat2);
    output_low(sat3);
    output_low(sat4);
    output_low(sut1);
    output_low(sut2);
    output_low(sut3);
    tus=99;
}

```

```

void pid()
{
    if(negatif == 1 && derece != 0)
    {
        output_low(isitici);
        output_high(sogutucu);
    }
    else if(derece >= sicaklik-0.2 && derece <= sicaklik+0.2)
    {
        output_low(isitici);
        output_low(sogutucu);
    }
    else if(derece > sicaklik )
    {
        output_high(isitici);
        output_low(sogutucu);
    }
    else if(derece < sicaklik)
    {
        output_low(isitici);
        output_high(sogutucu);
    }
    counterpid -= 1;
}

float absdiff(float x,float y)
{
    if(x>y)
        return x-y;
    else
        return y-x;
}

// ana program
void main()
{
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_CLOCK_DIV_2);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_ccp1(CCP_OFF);

    lcd_init();
    delay_ms(100);
    sht_init();
    delay_ms(100);

    set_tris_a(0x00);    // A portu komple çıkış
    set_tris_b(0x00);
    set_tris_c(0x0f);
    set_tris_d(0x00);

    lcd_init();    // LCD hazırlanıyor

```

```

while(1)
{
    allkeypadlow();
    keypad_oku();
    pres = tus;
    allkeypadlow();

    //kare
    if(pres == 19)
    {
        flagmode = 1;
        printf(lcd_putc, "\f");
    }

    while(flagmode==1)
    {
        lcd_gotoxy(1,1);
        printf(lcd_putc, "Mod gir");
        keypad_oku();
        pres = tus;
        if(pres == 19)
            pres = 99;
        allkeypadlow();

        if(pres == 1)
        {
            lcd_gotoxy(2,2);
            printf(lcd_putc, "%d", pres);
            mode = 0;
            delay_ms(1000);
            printf(lcd_putc, "\f");
            lcd_gotoxy(1,1);
            printf(lcd_putc, "On-Off Ayarlandi.");
            delay_ms(1000);
            flagmode = 0;
        }

        if(pres == 2)
        {
            lcd_gotoxy(2,2);
            printf(lcd_putc, "%d", pres);
            mode = 1;
            delay_ms(1000);
            printf(lcd_putc, "\f");
            lcd_gotoxy(1,1);
            printf(lcd_putc, "PID Ayarlandi.");
            delay_ms(1000);
            flagmode = 0;
            counterpid = absdiff(sicaklik, derece) * 10000;
        }

        if(pres == 3)
        {
            lcd_gotoxy(2,2);
            printf(lcd_putc, "%d", pres);
            mode = 1;
            delay_ms(1000);
            printf(lcd_putc, "\f");
            lcd_gotoxy(1,1);

```

```

        printf(lcd_putc,"P Ayarlandi.");
        delay_ms(1000);
        flagmode = 0;
        counterpid = absdiff(sicaklik,derece) * 10000;
    }

    if(pres == 4)
    {
        lcd_gotoxy(2,2);
        printf(lcd_putc,"%d",pres);
        mode = 1;
        delay_ms(1000);
        printf(lcd_putc,"\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc,"PI Ayarlandi.");
        delay_ms(1000);
        flagmode = 0;
        counterpid = absdiff(sicaklik,derece) * 10000;
    }

    if(pres == 5)
    {
        flagmode = 0;
        printf(lcd_putc,"\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc,"CikiS.");
        delay_ms(1000);
    }

}

//yıldız
if(pres == 10)
{
    flag = 1;
    printf(lcd_putc,"\f");
}

while(flag==1)
{
    lcd_gotoxy(1,1);
    printf(lcd_putc,"SIC gir");
    keypad_oku();
    pres = tus;
    if(pres == 10 && i == 1)
        pres = 99;
    allkeypadlow();
    if( pres >= 0 && pres <= 9)
    {
        lcd_gotoxy(i,2);
        if(i==1)
            derece = 0;
        printf(lcd_putc,"%d",pres);
        delay_ms(1000);
        derece = (derece*10) + pres;
        i += 1;
    }
    else if(pres == 19)
    {
        printf(lcd_putc,"\f%d derece \nayarlandi.",derece);
        flag = 0;
    }
}

```

```

        negatif = 0;
        delay_ms(1000);
        i=1;
        printf(lcd_putc, "\f");
    }
    else if(pres == 10)
    {
        printf(lcd_putc, "\f-%d derece \nayarlandi.", derece);
        flag = 0;
        delay_ms(1000);
        negatif = 1;
        i=1;
        printf(lcd_putc, "\f");
    }
    else{;}
}

sht_rd (temp, humid);
sicaklik=(float)temp;
nem=(float)humid;
lcd_gotoxy(1,1);
printf(lcd_putc, "SICAKLIK %2.2f C", sicaklik);
lcd_gotoxy(1,2);
printf(lcd_putc, "NEM %2.2f RH", nem);

//sensor kontrol
output_low(ses_sensor);
    delay_ms(1000);
if(nem < 1)
{
    output_high(ses_sensor);
    delay_ms(1000);
}

if(negatif == 1 && derece != 0 && mode == 0)
{
    output_low(isitici);
    output_high(sogutucu);
    delay_ms(1000);
}
else if(derece >= sicaklik-0.2 && derece <= sicaklik+0.2 && mode ==
0)
{
    output_low(isitici);
    output_low(sogutucu);
    delay_ms(1000);
}
else if(derece > sicaklik && mode == 0)
{
    output_high(isitici);
    output_low(sogutucu);
    delay_ms(1000);
}

else if(derece < sicaklik && mode == 0)
{
    output_low(isitici);
    output_high(sogutucu);
    delay_ms(1000);
}

```



```
    if(mode == 2 && counterpid != 0)
    {
        pid();
        anticounter = 10000; //bekleme süresi
    }
    else if(mode == 2 && counterpid == 0)
    {
        if(anticounter == 0)
            counterpid = absdiff(sicaklik,derece)* 50000; //2 derece
arasındaki farkla orantılı çalışma süresi
        else
        {
            anticounter -= 1;
            output_low(isitici);
            output_low(sogutucu);
        }
    }
}
```