



## **SICAKLIK KONTROL SİSTEMİ**

### **MİKROİŞLEMCİLER 13. PROJE RAPORU**

**CEREN İŞBİLEN**  
**210207012**

**Bölümü: Elektronik ve Haberleşme Mühendisliği**

**KOCAELİ, 2024**

# İçindekiler

1.GİRİŞ .....	3
1.1 Sıcaklık kontrol otomasyonu hakkında.....	3
1.2 Projenin amacı.....	3
2.DONANIM.....	4
2.1 Kullanılan Malzemeler.....	4
2.2 Altium çizimi & bağlantılar.....	5
2.3 Breadboard üzerinde denenmesi.....	6
2.4 Baskı alınmış devre .....	6
3.YAZILIM.....	7
3.1 Algoritma .....	7
3.2 Kod.....	7
4.KAYNAKÇA.....	9

# 1.GİRİŞ

## 1.1 Sıcaklık kontrol otomasyonu

Sıcaklık kontrolü, hem evlerde hem de endüstriyel alanlarda önemli bir konudur. Ortam sıcaklığının istenilen seviyede tutulması, hem enerji tasarrufu sağlar hem de rahat bir ortam yaratır. Sıcaklık kontrol otomasyonu, bu süreci otomatik hale getirir ve insan müdahalesi olmadan sıcaklığı ayarlar.

## 1.2 Projenin amacı

Bu projede, MSP430 mikrodenetleyici kullanarak bir sıcaklık kontrol sistemi prototipi geliştirilmiştir. Kullanım kolaylığı açısından MCP9700 sıcaklık sensörü ile ortam sıcaklığı algılanmıştır. Algılanan sıcaklık için baz alınan değer, oda sıcaklığı yani 25 derece olarak belirlenmiştir.

Sıcaklık sensöründen algılanan ortam sıcaklığı, oda sıcaklığından düşük olduğunda ISITMA sistemi devreye girecek. Isıtma sistemi temsili olarak basit Led komponent ile temsil edilmiştir.

Sıcaklık sensöründen algılanan ortam sıcaklığından yüksek, ancak 34 dereceden de düşük ise SOĞUTMA'yı temsilen Fan devreye girecektir. Eğer ölçülen ortam sıcaklığı 34 dereceden de yüksek ise, ortamın gereğinden fazla sıcak olup tehlike göstergesi olduğunu temsilen Led komponenti yanıp sönecek şekilde ayarlanmıştır.

## 2.DONANIM

### 2.1 Kullanılan malzemeler

#### *MCP9700 Sıcaklık sensörü*

LM35 ve DS18B20 de projede kullanıldı ancak hassas sonuç ve kullanım kolaylığı açısından en son MCP9700’de karar kılındı.

#### *5V Fan ve Röle*

MSP430’un kontrol pinlerinden 3.3 V geldiği için ve fanın çalışması için 5V gerektiğinden, BC547 transistörü ile anahtarlama yapılarak 5V röle ile fan tetiklendi.

#### *LCD 2x16 Ekran*

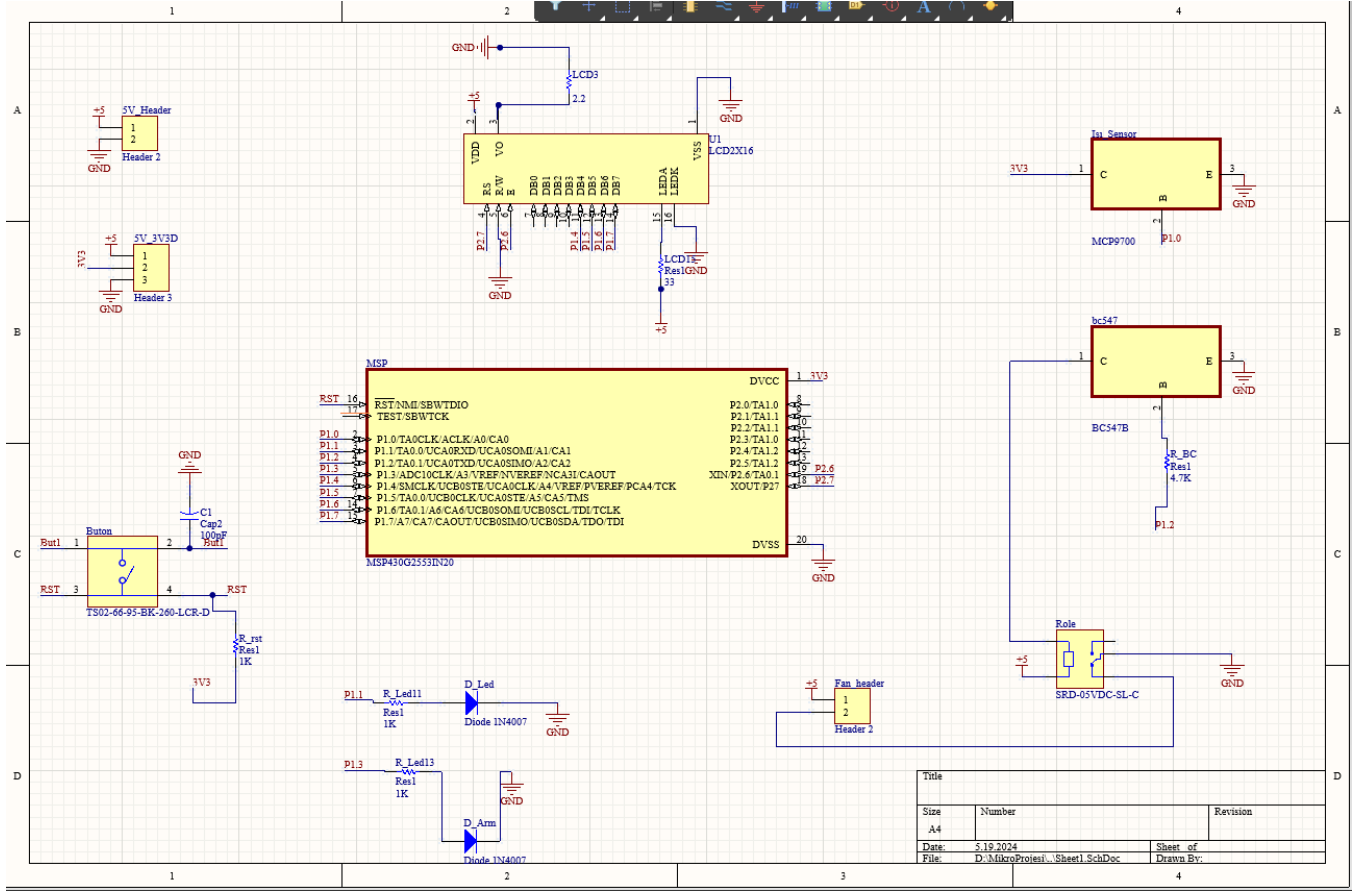
Anlık ortam sıcaklığını ve ortam koşulunu görebilmek için LCD 2x16 ekran kullanıldı.

#### *Led ve diğer komponentler*

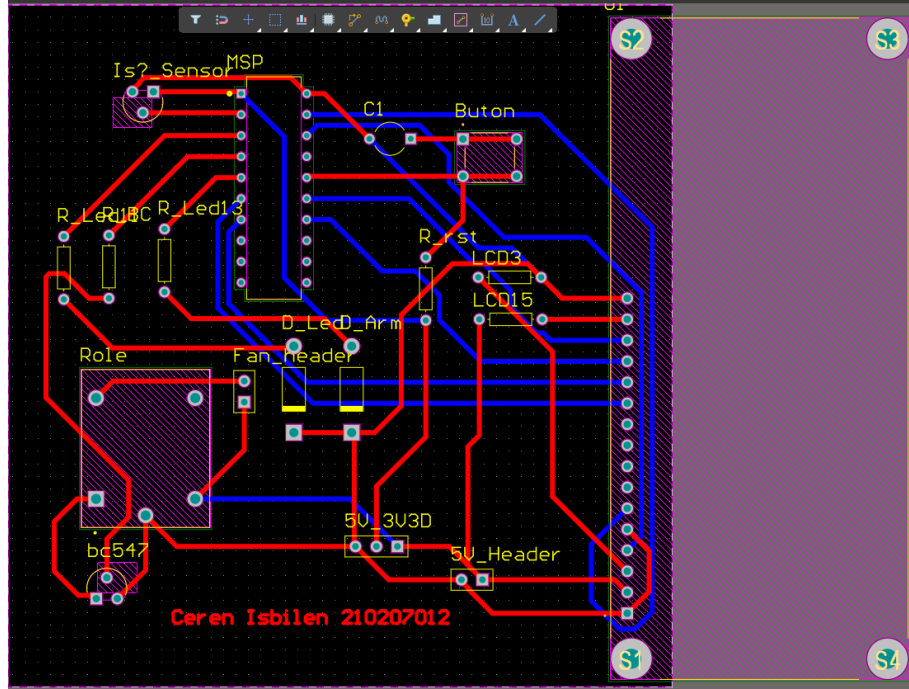
Eğer ortam sıcaklığı “soğuk” koşulunda ise, ısıtma temsili olarak; ortam sıcaklığı anormal yüksek ise alarm olarak kırmızı led kullanılmıştır.

Harici olarak gerekli direnç, header, kapasitör kullanılmıştır.

## 2.2 Altium çizimi & bağlantılar



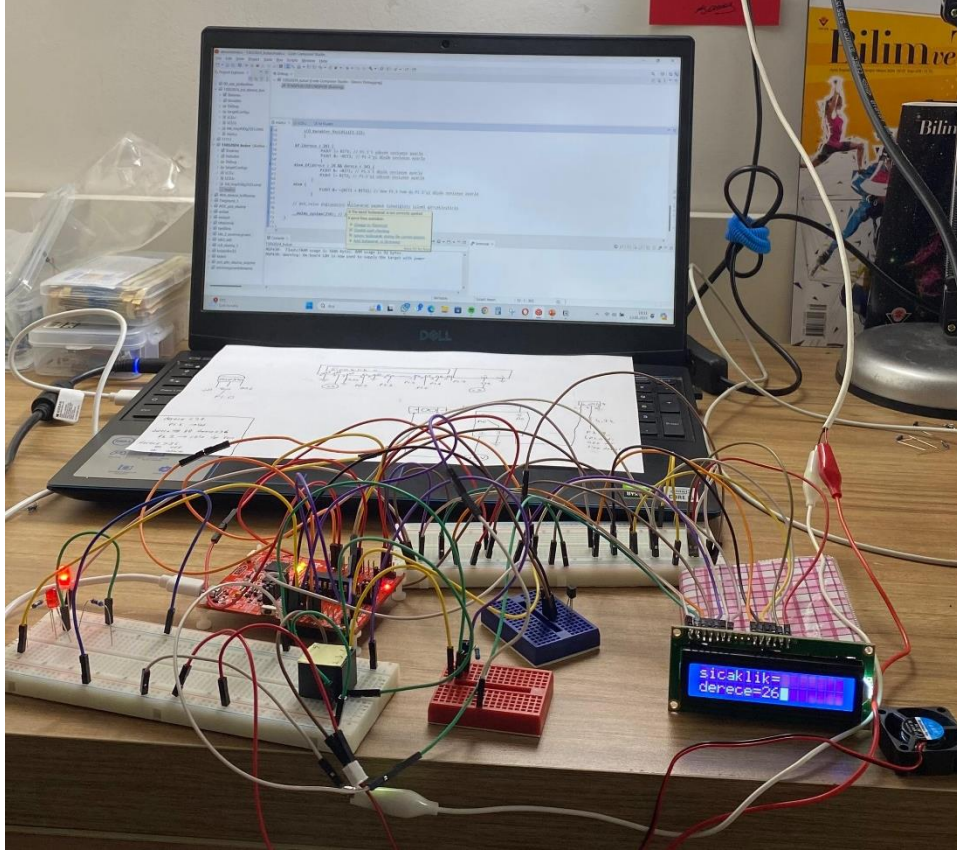
Şekil 2.1 (Devre çizimi)



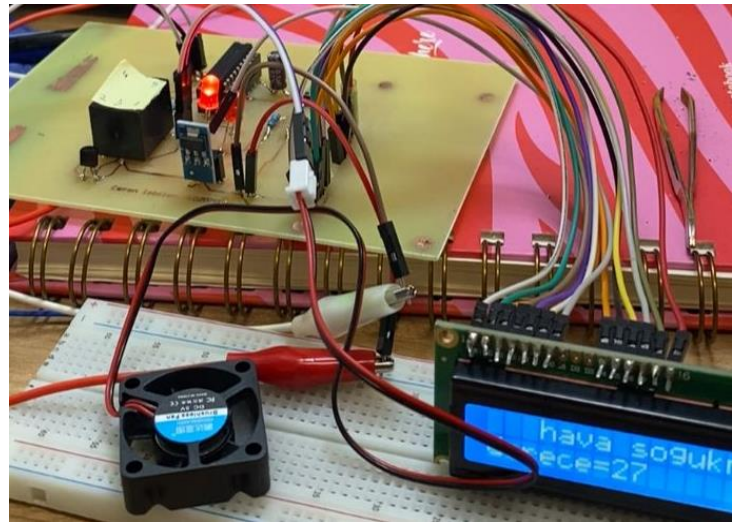
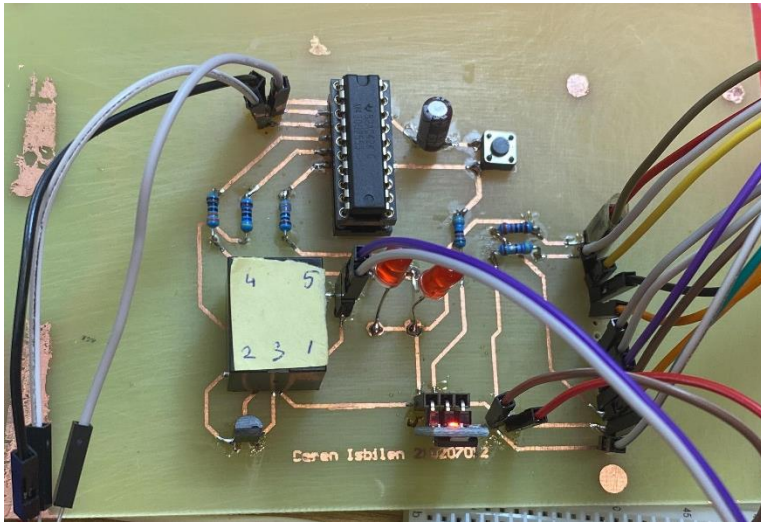
Şekil 2.2 (PCB Bağlantıları)

Sıcaklık kontrol sistemi prototipi devremizin önce breadboardda denemesi yapıldıktan sonra altium ortamına birebir şekilde aktarılmıştır.

### 2.3 Breadboard üzerinde denenmesi



Şekil 2.3 (breadboard üzerinde çalışan sistem)



Şekil 2.4 (PCB basılmış hali)

### 3. YAZILIM

#### 3.1 Algoritma



#### 3.2 Kod

Giriş ayarları&tanımlamaları:

```

main.c × LCD.c LCD.h
1 #include <msp430.h>
2 #include "LCD.h"
3
4 unsigned int adc_value = 0; // ADC degerini saklamak için degisken
5 int pot_value = 0;
6 float derece; // Dereceyi saklamak için degisken
7 unsigned char dizi[2]; // ekrana yazılacak degeri saklamak için dizi
8
9 int i,j; // Döngü için sayac
10

```



Pin atamaları ve ADC ayarları:

```

11 void main(void)
12 {
13     WDTCTL = WDTPW | WDTHOLD; // Watchdog timer'ı durdur
14     BCSCTL1 = CALBC1_1MHZ; // DCO'yu 1MHz'e ayarla
15     DCOCTL = CALDCO_1MHZ; // DCO kontrol register'ını 1MHz'e ayarla
16     LCD_Ayarla(); // Başlangıçta LCD ayarlarını yap
17
18     P1SEL |= BIT0; // A0 pini olarak ayarla
19     P1DIR |= BIT1 + BIT2 + BIT3; // P1.1, P1.2 ve P1.3 çıkış olarak ayarlandı
20
21     // ***** ADC ayarları ***** //
22     ADC10CTL1 = INCH_0; // A0 pini için ADC kanalını seç
23     ADC10CTL0 = SREF_0 + ADC10SHT_2 + ADC10ON; // Referans seviyesi besleme ve toprak arasında,
24     ADC10AE0 |= BIT0; // A0 pini için analog giriş etkinleştir
25

```

ADC değeri okuma ve LCD'ye basmaya hazırlamak için diziye atama:

```

25
26 while(1){
27     ADC10CTL0 &= ~ENC; // Dönüşüm bitini sıfırla
28     while (ADC10CTL1 & BUSY); // Eğer ADC mesgulse bekle
29     ADC10CTL0 |= ENC + ADC10SC; // Dönüşümü başlat
30     while (ADC10CTL1 & BUSY); // Dönüşüm tamamlanana kadar bekle
31
32     adc_value = ADC10MEM; // ADC sonucunu oku
33
34     // ADC değerini 0-40 arasında bir sayıya dönüştür
35     pot_value = (unsigned int)(adc_value);
36     derece = (pot_value * 3.3) / 1023; // ADC değerini voltaja dönüştür
37     derece = derece / 0.0322; // Voltajı dereceye dönüştür (burada gerçek volt değeri kullanılmadı)
38     int derece_int = (int)derece; // Dereceyi int türüne dönüştür
39     int x = 1;
40
41     for(i = 0; i <= 1; i++){
42         dizi[i] = ((derece_int / x) % 10) + 48; // Dereceyi karakter dizisine dönüştür
43         x = x * 10;
44     }
45

```

Hangi komponentin hangi koşulda çalışacağı:

```

46     LCD_Git_XY(2, 1); // Kursörü 2.satır 1.sütuna götür
47     LCD_Yazi_Yaz("derece="); // Ekrana "derece=" yazdır
48     for(i = 0; i <= 1; i++){
49         LCD_Karakter_Yaz(dizi[1 - i]); // Dereceyi ekrana yazdır
50     }
51
52     if (derece < 28.00) {
53         P1OUT |= BIT1; // P1.1'i yüksek seviyeye ayarla
54         P1OUT &= ~(BIT2 + BIT3); // P1.2 ve P1.3'ü düşük seviyeye ayarla
55         LCD_Git_XY(1, 1); // Kursörü 1.satır 1.sütuna götür
56         LCD_Yazi_Yaz("hava soguk"); // Ekrana "hava soguk" yazdır
57     } else if (derece > 28.00 && derece < 35.00) {
58         LCD_Git_XY(1, 1); // Kursörü 1.satır 1.sütuna götür
59         LCD_Yazi_Yaz("hava orta sicak"); // Ekrana "hava orta sicak" yazdır
60         P1OUT &= ~(BIT1 + BIT3); // P1.1 ve P1.3'ü düşük seviyeye ayarla
61         P1OUT |= BIT2; // P1.2'yi yüksek seviyeye ayarla
62     } else {
63         LCD_Git_XY(1, 1); // Kursörü 1.satır 1.sütuna götür
64         LCD_Yazi_Yaz("hava cok sicak"); // Ekrana "hava cok sicak" yazdır
65         P1OUT ^= BIT3; // P1.3'ü toggle yap
66
67         // Bekleme döngüsünü 17 kez tekrar ediyoruz (yaklaşık 17 saniye)
68         for (j = 0; j < 17; j++){
69             // Yaklaşık 1 saniyelik bekleme döngüsü
70             for(i = 4000000; i > 0; i--); // yaklaşık 1 saniye bekle
71         }
72         P1OUT &= ~(BIT1 + BIT2); // Hem P1.1 hem de P1.2'yi düşük seviyeye ayarla
73     }
74

```



#### **4.KAYNAKÇA**

1. [ADC in MSP430 Microcontroller - Part 2: Multiple Conversion \(youtube.com\)](#)
2. [MSP430 Denetleyiciler ile Karakter LCD Kullanımı – Elektronik Devreler Projeler \(320volt.com\)](#)