



# ZONGULDAK BÜLENT ECEVİT ÜNİVERSİTESİ

## ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

### EEM419 Mikroİşlemciler

Güz 2022

<b>Proje Adı</b>	Yüksek Sıcaklık ve Nemde Uyarım ve Optimal Sıcaklık İçin İstenilen Zamanda Ortam Soğutucu		
<b>Proje Grp No</b>	I-8	<b>Tarih</b>	10.01.2023
<b>Proje Grup (No, Ad-Soyad)</b>	1) 180106106006, Doğan-Nalçacı 2) 180106106061, Uğurcan-Doğan 3) 180106106062, Erkan-Doğan		

	<b>Talimat Veren</b>	<b>Talimat Alan</b>
<b>Malzeme Temini</b>	Uğurcan Doğan	Erkan Doğan
<b>Tasarım Yapılması</b>	Doğan Nalçacı	Uğurcan Doğan
<b>Yazılım yapılması</b>	Erkan Doğan	Doğan Nalçacı
<b>Sonuç Kaydetme</b>	Doğan Nalçacı	Erkan Doğan

	<b>Talimat Alan/Veren</b>
<b>Uygulama Genel Değerlendirmesi ve Yorumlanması</b>	Grupta görev dağılımı eşit bir şekilde yapılarak gerekli uygulamalar gerçekleştirildi ve rapor haline getirildi. Projemiz ilk düşünüldüğünde sıcaklık ölçümü ve fan ile manuel kontrol ile soğutulması olarak planlandı. Son şekline getirildiğinde sıcaklık artışıyla anlık fan hızı otomatik olarak yükseltildi ve kesme olarak acil durum butonu yerleştirilerek fan durdurulup sesli ve ışıklı uyarı verilmesi sağlandı. Sistem yazılımı ve tasarımı planlandığı gibi yapılarak proje istenilen şekilde gerçekleştirildi.

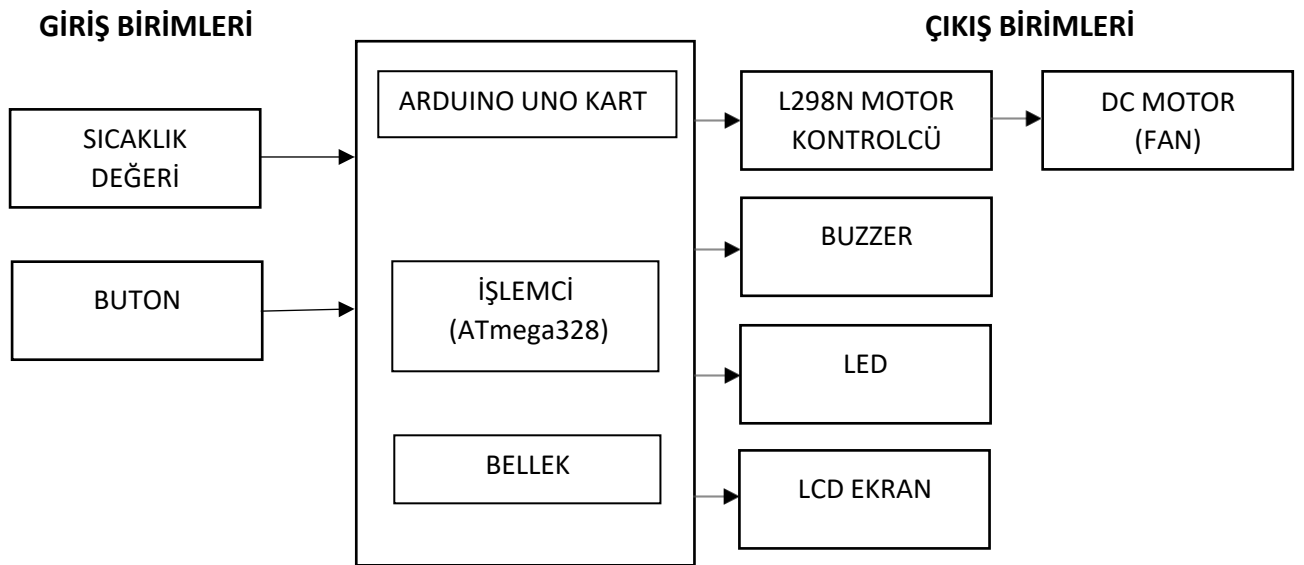
**PROJE ADI:** Yüksek Sıcaklık ve Nemde Uyarım ve Optimal Sıcaklık İçin İstenilen Zamanda Ortam Soğutucu

**1-AMAÇ VE HEDEFLER:** Sıcaklık faktörünün önemli olduğu spesifik ortamlarda optimal sıcaklık değerinin sağlanması için fan yardımıyla ortam ısısının sabit tutulması amaçlanmaktadır. Projemizde kullanılan DHT sıcaklık sensörü ile anlık değerler alınmaktadır. Sıcaklığın yükselmesi durumunda soğutucu olarak kullanılan fan, kontrolcü yardımıyla hızını sıcaklık değeriyle doğru orantılı olarak arttırıp azaltmaktadır. Sıcaklık değeri belirlediğimiz kritik sıcaklık limitine ulaştığında, sistem ışıklı uyarı verip, fan maksimum düzeyde çalışmaktadır. Sistemimizde yangın durumunda acil durum butonu kullanılmaktadır. Acil durum butonuna basıldığında yangının yayılmasını önlemek amaçlı, fan durmakta ve sistem sesli ve ışıklı uyarı vermektedir. Butona tekrar basıldığında sistem normal işleyişine geri dönmektedir.

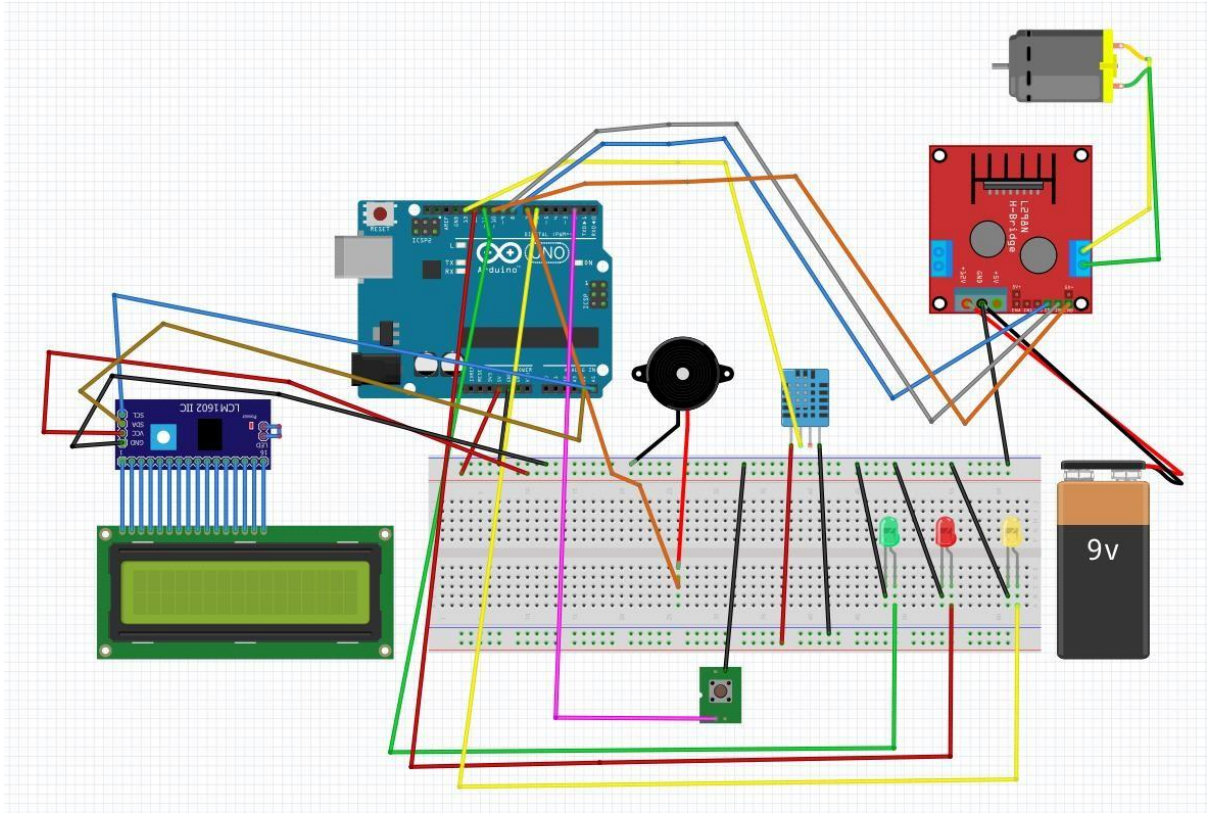
- Projemizde, Arduino Uno kartının ATmega328 (8-bit) SRAM: 2KB işlemcisi kullanılmıştır. Bulundukları elektronik yapıların beyni olan mikroişlemciler, bağlı oldukları mekanizmanın kontrolünü sağlamakla görevlidirler. Sistemimizde mikroişlemci kullanılmasının nedeni, sıcaklığa bağlı anlık olarak fan hızı artışı ve bu artışa bağlı uyarım ve kontrolün yapılmasıdır.
- Sistemimizde en az iki giriş ve çıkış kullanılmıştır. Giriş elemanlarımız; sıcaklık değeri ve butondur. Çıkış elemanlarımız; dc motor(fan), led, buzzer ve lcd ekrandır.
- Sistemimizde kesme olarak buton kullanılmıştır. Nedeni yangın gibi acil bir durumda butona basılarak, alevlerin yayılmasını önlemek için fan durdurulmuştur ve tahliye için aynı anda sesli ve ışıklı uyarı verilmiştir.

## 2-TASARIM:

Aşağıda Şekil.2.1 ve Şekil.2.2’de proje akış şeması ve Fritzing devre şeması verilmektedir.



**Şekil.2.1** Proje akış şeması



Şekil.2.2 Fritzing devre şeması

### 3-DONANIM:

**Arduino uno:** Arduino / Genuino Uno, bir bilgisayar, başka bir Arduino/Genuino kartı veya diğer mikrodenetleyicilerle iletişim kurmak için bir dizi özelliğe sahiptir. Atmega328, 0 (RX) ve 1 (TX) dijital pinlerinde bulunan UART TTL (5V) seri iletişimi sağlamaktadır. Arduino.cc tarafından geliştirilen açık kaynaklı bir mikrodenetleyici kartıdır. Arduino Uno, diğer devrelere arayüzlenebilen dijital ve analog giriş/çıkış pinleri ile donatılmıştır.

**ATmega328 işlemci:** ATmega328p sistemi ile FLASH belleğe yaklaşık olarak 10000 defa yazma ya da okuma yapılabilir. 131 adet tek çevrimde çalışan oldukça güçlü bir komut setine sahip olup 20 MHZ'e kadar da hızı desteklemektedir. Tamamı ile statik mikrodenetleyici olarak karşımıza çıkar. ATmega328, Atmel tarafından megaAVR ailesinde oluşturulan tek çipli bir mikro denetleyicidir. Değiştirilmiş bir Harvard mimarisi 8-bit RISC işlemci çekirdeğine sahiptir.

**Arduino bellek:** Projelere göre değişen Arduino UNO boyutları ile birlikte bellekler de değişebilir. Genel olarak 32 KB flash belleğe sahiptir ve 2 KB SRAM'i vardır. Statik Rasgele Erişim Belleği, taslağın çalışırken değişkenleri oluşturduğu ve değiştirdiği yerdir.

**DHT sensör:** DHT11, ortamdaki sıcaklığı ve nemi ölçmek amacıyla kullanılan bir sensördür. Ortam sıcaklığını çeşitli ısı birimlerinde (celcius, farhenayt, kelvin) elde etmemizi ve ortamdaki nemi ölçmek için kullanılmaktadır. Ortamdaki havayı ölçmek için bir kapasitif nem sensörü ve bir termistör kullanır ve veri pini üzerinde dijital bir sinyal çıkarır.

**Buton:** Buton Arduino'ya veri girişı yapılmasını saęlayan bir aratır. Buton sayesinde Arduino zerindeki dijital pinlerle veri girişı yapılabilir. Dijital pin ile 1 ve 0 deęerleri gnderilebilir. Butonun soldaki iki bacağı kendi aralarında, saędaki iki bacağı da kendi aralarında kısa devre halindedir. Butona basıldıęı anda tm bacaklar kısa devre durumuna geer.

**L298N motor src:** L298N DC Step Motor Src 4.8V-46V arasındaki iki motoru srmek iin hazırlanmıř bir motor src kartıdır. İki kanallı bu motor src kanal başına 2A akım vermektedir. Kart zerinde L298N motor src entegresi kullanılmıřtır.

**Buzzer:** Buzzer, piezo etkisi ile ses ıkarmaya yarayan bip sesi ıkaran bir komponenttir. Buzzerlar genelde zerinden geen elektrik akımı ile tepkimeye girmesi sonucu kristalize bir ses ıkarmaktadırlar.

**Led:** LED (Light Emitting Diode) ıřık yayan diyot anlamına gelmektedir. Arduino iin kullanılan LED'ler 5V ile alıřır. LED'lerin anot ve katot olmak zere iki adet bacağı vardır. Anot bacağı + kutbuna, katot bacağı ise – kutbuna denk gelir.

**LCD ekran:** LCD, Liquid Crystal Display yani Sıvı Kristal Ekran elektrikle kutuplanan sıvının ıřıęı tek fazlı geirmesi ve nne eklenen bir kutuplanma filtresi ile gzle grlebilmesi ilkesine dayanan bir grnt teknolojisidir.

**I2C:** I2C, Inter-integrated devre iletiřim protokol anlamına gelir. Kısa menzilli veri aktarım uygulamaları iin sadece 2 kablo gerektiren seri iletiřim protokoldr. I2C tabanlı sensrler, LCD/OLED ekranlar ve iletiřim modlleri iin arayz oluřturmak iin projelerde kullanılan ok popler bir iletiřim protokoldr.

**DC motor:** DC motorlar doęru akım enerjisini mekanik enerjiye eviren makinelerdir. Motorun iinde yer alan sargılara elektrik akımı uygulandıęında, yine motorun ierisinde bulunan sabit mıknatıslara zıt ynde oluřan manyetik kuvvetin etkisi ile hareket etmesi prensibine dayanmaktadır. Endstriyel alanda olduęu kadar ev aletlerinde de kullanılan DC motorlar, aspiratrler, su pompaları ve fanlar gibi pek ok alanda karřımıza ıkmaktadır.

#### 4-YAZILIM:

##### Proje kodları ve aıklamaları:

```
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#define yesilled 11 //Yeřil led ataması
#define kirmiziled 12 //Kırmızı led ataması
#define motor1hiz 10 //Byk motor hız pin giriři
#define motor1 8 //Byk motorun (+) sı
#define motor2 9 //Byk motorun (-) sı
#define DHTPIN 13 //DHT pin giriř pini
#define DHTTYPE DHT11 //Ktphaneden DHT 11 seimi
DHT proje(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2); //LCD Ataması
int length = 1;
float t = proje.readTemperature(); //Sıcaklık tanımlaması
```

```

const byte buzzerPin = 7; //Buzzer ataması
const byte ledPin = 6; //Sarı led ataması
const byte interruptPin = 2; //Kesme buton ataması
volatile byte state = HIGH; //Açılma
volatile byte state1 = LOW; //Kapanma
void setup() {
    pinMode(yesilled, OUTPUT);
    pinMode(kirmiziled, OUTPUT);
    pinMode(motor1hiz, OUTPUT);
    pinMode(motor1, OUTPUT);
    pinMode(motor2, OUTPUT);
    lcd.begin();
    proje.begin();
    pinMode(buzzerPin, OUTPUT);
    pinMode(ledPin, OUTPUT);
    pinMode(interruptPin, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(interruptPin), blink, RISING); //Gerilim Lojik-0' dan Lojik-1' e
}
void loop() {
    float t = proje.readTemperature();
    digitalWrite(buzzerPin, state);
    digitalWrite(ledPin, state);
    digitalWrite(motor1, state1);
    pinMode(motor1hiz, state1);
    if (10 <= t && t <= 20) { //Eğer sıcaklık 10 ile 20 arasındaysa
        lcd.setCursor(0, 0);
        lcd.print("Sicaklik: "); //Sıcaklık gösterilecek
        lcd.print(t);
        lcd.println(" *C ");
        lcd.setCursor(0, 1);
        lcd.println("Fan hizi: %25"); // Fan durumunun bilgisi
        delay(500);
        digitalWrite(yesilled, HIGH); // Yeşil led yanacak
        digitalWrite(kirmiziled, LOW); //Kırmızı led kapalı durumda olacak
        analogWrite(motor1hiz, 40); //Motor hızı 2.5 voltaj gönderip 40 rp de dönecek
        digitalWrite(motor1, 1); //Motor sol yönde açık
        digitalWrite(motor2, 0); //Motor sağ dönme kapalı
    }
    else if (20 <= t && t <= 25) { //Eğer sıcaklık 20 25 arasında olursa
        lcd.setCursor(0, 0);
        lcd.print("Sicaklik: "); //Sıcaklık gösterilecek
        lcd.print(t);
        lcd.println(" *C ");
        lcd.setCursor(0, 1);
        lcd.println("Fan hizi: %50");
        delay(500);
        digitalWrite(yesilled, HIGH); // Yeşil led yanacak
        digitalWrite(kirmiziled, LOW); //Kırmızı led kapalı durumda olacak
        analogWrite(motor1hiz, 80); //Motor hızı 4,48 voltaj gönderip 85 rp de dönecek
        digitalWrite(motor1, 1); //Motor sol yönde açık
        digitalWrite(motor2, 0); //Motor sağ yönme kapalı
    }
}

```

```

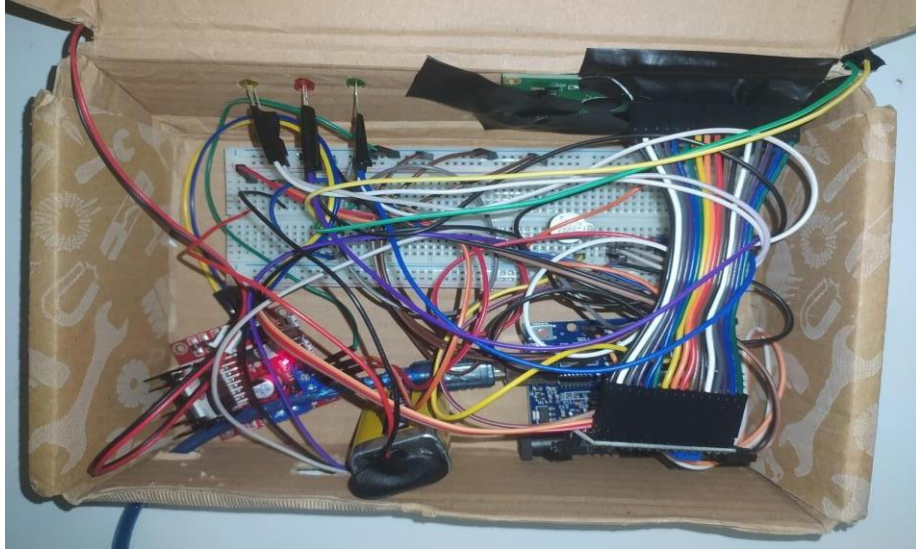
else if (26 <= t && t <= 30) { //Eğer sıcaklık 26 30 arasında olursa
    lcd.setCursor(0, 0);
    lcd.print("Sicaklik: "); //Sıcaklık gösterilecek
    lcd.print(t);
    lcd.println(" *C ");
    lcd.setCursor(0, 1);
    lcd.println("Fan hizi: %75");
    delay(500);
    digitalWrite(yesilled, HIGH); // Yeşil led yanacak
    digitalWrite(kirmiziled, LOW); //Kırmızı led kapalı durumda olacak
    analogWrite(motor1hiz , 150); //Motor hızı 7,5 voltaj gönderip 150 rp de dönecek
    digitalWrite(motor1, 1);
    digitalWrite(motor2, 0);
}
else if (31 <= t && t <= 60) { //Eğer sıcaklık 31 60 arasında olursa
    lcd.setCursor(0, 0);
    lcd.print("Sicaklik: "); //Sıcaklık gösterilecek
    lcd.print(t);
    lcd.println(" *C ");
    lcd.setCursor(0, 1);
    lcd.println("Fan hizi: %100");
    delay(500);
    digitalWrite(yesilled, LOW);
    digitalWrite(kirmiziled, HIGH);
    analogWrite(motor1hiz , 255); //Motor hızı 10,5 voltaj gönderip son hızda dönecek
    digitalWrite(motor1, 1);
    digitalWrite(motor2, 0);
}
else { // yukarıdakiler gerçekleşmediği her durumda
    digitalWrite(kirmiziled, LOW); //Kırmızı led kapalı durumda olacak
    digitalWrite(yesilled, HIGH); // Yeşil led yanacak
    Serial.println(" .... Kapalı .... ");
    analogWrite(motor1hiz , 0); //Motor kapalı konumda ve 0 voltaj da
    digitalWrite(motor1, 0); //Motorun sağ yada sola hareketi kapalı
    digitalWrite(motor2, 0);
}
}
void blink() { //Kesme algoritması ataması
    state = !state;
    state1 = !state1;
}

```

## 5-UYGULAMA SONUÇ VE YORUMLARI:

Planlamakta olduğumuz projeyi isterlere göre gerçekleştirdik ve başarılı bir şekilde sonuçlandırdık. Projemizin amacını gerçekleştirdik ve ortamın optimal sıcaklığı sağlanarak sistem tarafından uyarılar verildi ve kesme işlemi buton yardımıyla gerçekleştirildi. Projemizdeki devre bir kutu içine yerleştirildi. Sensörler, LCD ekran ve uyarıcı sistemler dışarıdan müdahale gerektirdiği için kutu dışına çıkarıldı. Aşağıda sistemimizin uygulama sonucundaki sistem iç ve dış son donanım ve tasarımına ait görseller belirtilmiştir.





**Şekil.5.1** Sistem iç donanımı



**Şekil.5.2** Normal çalışma durumu



**Şekil.5.3** Uyarım ve kesme durumu