

[**Mühendislik Fakültesi**](https://www.konya.edu.tr/muhendislikvemimarlik)

[**Bilgisayar Mühendisliği Bölümü**](https://www.konya.edu.tr/bilgisayarmuhendisligi)

**Gömülü Sistemleri**

|  |
| --- |
| **Uygulama Konusu** |
| Soğutma Sistemi |

|  |  |
| --- | --- |
| **Öğrenci Bilgileri** | |
| **Ögr. No** |  |
| **Ad Soyad** |  |

**Dersin Hocası**

**Doç. Dr. Muhammed KARAALTUN**

**Haziran 2023**

**Konya**

|  |
| --- |
| **Projede Aranan Özellikler** |
| Gömülü Sistemler dersi Proje Ödevi olarak her öğrenci Arduino tabanlı basit bir sistem geliştirecektir. Sistemde mutlaka görsel yazılım kullanılacaktır.    **Genel Özellikler:**           Arduino Tabanlı Sistem bağlantı şeması için fritzing paket programı kullanılacaktır.           ADC, DAC, Timer, (Zamanalayıcı), EEPROM, Kütüphane. Projenizde bu özelliklerden en az 3 adet kullanılmalı           Arduino yazılımı geliştirmek için Arduino veya (visual studio, visual C++, vb.) programlama dili kullanılabilir.  Kodlar düzgün bir şekilde raporun Ekler bölümünün altında yer alacaktır.    **Önerilen Projeler:**         Şehir aydınlatma sistemi veya Ev otomasyon sistemi, bu sistemde (ADC, DAC, Timer, (Zamanalayıcı), EEPROM) kullanılacaktır, arduinoya bağlı olan aygıtların bilgileri veri tabanına kaydedilecektir ve aygıtların bilgileri görsel olarak görsel yazılım kullanarak tasarlanacaktır. |
| **Proje Raporu Hazırlama Kuralları** |
| Her öğrenci proje ödevini aşağıda belirlenen kurallara göre hazırlayacaktır.  **1.**       **Rapor Bölümleri**           Kapak           içindekiler           Özet           Projede kullanılan donanım ve yazılımla ilgili bilgileri (vermiş olduğum notlardaki gibi)  Donanımda kullanılan malzemelerin açıklaması, özelikleri ve kullanım şekli gibi bilgiler.  Yazılımda kullanılan komutların ve fonksiyonların açıklaması.           Kaynaklar           Ekler (Kodlar)  **2.**       **Konu Anlatımı**           Etik kurallara uygun olarak, öğrenci konuyu kendi cümleleri ile sade bir şekilde anlatmalıdır. Yararlanılan kaynaklar belirtilmelidir.  **3.**       **Sayfa Düzeni**           Kenar Boşlukları: 2.5 cm           Sayfa Numarası: Sağ Alt Köşede (“Sayfa No/Toplam Sayfa” şeklinde. Örn: 1/10, 2/10 şekilde.)  **4.**       **Metin Özellikleri**           Paragraf Girintisi Yok           Metin iki yana yaslı           Tek satır aralıklı           12 Punto ve Normal           Paragraflar arasında 1 boşluk           Ana Başlıklar: 14 Punto ve Kalın, Alt Başlıklar:12 Punto ve Kalın           Tüm Metin Fontu: Times New Roman  **5.**       **Şekiller ve Tablolar**           Şekiller sayfa içerisinde ortalı olmalıdır.           Şekil açıklama metni şekil altında tek satıra sığıyorsa ortalı, sığmıyorsa iki yana yaslı sol kenara yaslı olmalıdır.           Tablolar sayfa içerisinde sola yaslı olmalıdır           Tablo açıklama metni tablo üstünde ve sola yaslı olmalıdır.  **6.**       **Kaynaklar**           Araştırma ödevinin hazırlanmasında yararlanılan kaynaklar, metin içerisinde kullanım sırasına göre rapor sonunda ve “Kaynaklar” başlığı altında sıra numaraları verilerek listelenecektir.  **Gömülü** **sistemleri** **projesi** **raporu** Word dosyası formatında sisteme yüklenecektir. Dosya adı öğrenci numarası olacaktır; son gönderme tarihi 6 Haziran Salı günü saat 17:00.  Başarılar Dilerim… |

**İÇİNDEKİLER**

* **Proje hakkında ……………………………………………………………………..4**
* **Projede kullandığım donanım ve yazılım ile ilgili bilgiler………………………..5**
* **Yazılımda kullandığım kodlar ve açıklamaları……………………………………6**
* **Fritzing………………………………………………………………………………7**
* **Kaynaklar…………………………………………………………………………...7**

**Proje Hakkında Özet**

Bu Arduino projesinin amacı, bir LM35 sıcaklık sensörü kullanarak ortam sıcaklığını ölçmek ve bu sıcaklık değerine bağlı olarak bir fanın hızını kontrol etmektir.

Projenin başlangıcında, TimerOne kütüphanesi dahil edilmekte ve gerekli değişkenler tanımlanmaktadır. LM35 sıcaklık sensörü, Arduino'nun A0 pinine bağlıdır ve sensörden okunan sıcaklık değeri temp\_adc\_val değişkenine atanır. Ardından, analog değeri gerilim değerine dönüştürerek sıcaklık değerini temp\_val değişkeninde saklarız.

Ana döngü (loop) içinde, sıcaklık değeri kontrol edilir. Eğer sıcaklık değeri 30 dereceden yüksekse, fanın çalışması gerekmektedir. Timer1.pwm() fonksiyonu kullanılarak fanın hızı ayarlanır ve fan çalışır duruma getirilir. Aynı zamanda, sıcaklık değeri seri haberleşme üzerinden görüntülenir.

Eğer sıcaklık değeri 30 dereceden düşükse, fan durdurulur. Timer1.disablePwm() fonksiyonu kullanılarak fanın hız kontrolü kapatılır ve seri haberleşme üzerinden fanın kapalı olduğu bilgisi görüntülenir.

Ana döngüdeki son işlem, 500 milisaniye (delay(500)) beklemektir. Bu bekleme süresi, sıcaklık değerinin belirli aralıklarla kontrol edilmesini sağlar.

Bu şekilde, projenin amacı, ortam sıcaklığını ölçerek sıcaklık değerine bağlı olarak bir fanın hızını kontrol etmek ve sıcaklık değerini seri haberleşme üzerinden görüntülemektir.

Projede belirlenen özellikler arasından ADC , Timer ve Timer için TimerOne Kütüphanesini kullandım .

Projemde amacım bir nevi kalorifer , klima sistemi gibi sıcaklığı bir seviyenin altında tutmaya çalışmak sıcaklık arttığı taktirde fanı çalışmasını sağlayıp sıcaklığı düşürmek.

**Projemde Kullandığım Donanım ve Yazılım Hakkında**

**Donanım:**

Arduino: Bu projede Arduino mikrodenetleyici kullanılıyor. Arduino, bir geliştirme platformudur ve kolayca programlanabilir. Bu projede Arduino, sensörlerden veri okuyup fanın kontrolünü gerçekleştiren işlevleri yerine getiriyor.

LM35 Sıcaklık Sensörü: Projede LM35 adında bir analog sıcaklık sensörü kullanılıyor. Bu sensör, ortam sıcaklığını ölçmek için kullanılır. Arduino'nun analog giriş pinine bağlanarak sıcaklık değerini sağlar.

L298N Motor Sürücü: Fanın hızını kontrol etmek için L298N adlı bir motor sürücü kullanılıyor. Bu sürücü, Arduino'dan gelen sinyalleri alarak fan motorunu belirli bir hızda çalıştırır. İki adet IN1 ve IN2 pinleri, fanın dönüş yönünü belirlerken, EN1 pinine bağlanan PWM sinyali ile fanın hızı kontrol edilir.

**Yazılım:**

Arduino IDE: Arduino projelerini programlamak için Arduino IDE (Integrated Development Environment) kullanılıyor. Bu projede de Arduino IDE, programı yazmak ve Arduino'ya yüklemek için kullanılıyor.

TimerOne Kütüphanesi: Projede TimerOne adlı bir kütüphane kullanılıyor. Bu kütüphane, zamanlayıcı işlevlerini kolaylaştıran fonksiyonları sağlar. Projede zamanlama ve PWM işlevleri için kullanılıyor.

setup() Fonksiyonu: Arduino programının başlangıcında otomatik olarak çağrılan setup() fonksiyonunda, seri haberleşme başlatılıyor ve gerekli pinlerin modları ayarlanıyor.

loop() Fonksiyonu: Arduino programının sürekli olarak tekrarlanan bölümü olan loop() fonksiyonunda, sıcaklık okunuyor, değer hesaplanıyor ve fanın hızı kontrol ediliyor. Aynı zamanda, sıcaklık değeri seri haberleşme ile görüntüleniyor.

Bu proje, ortam sıcaklığını ölçmek ve fan hızını buna göre ayarlamak için kullanılan donanım ve yazılımı içermektedir. Arduino ve diğer bileşenler, sıcaklık sensöründen gelen verileri okur, işler ve fanın hızını belirli bir sıcaklık eşiği üzerinde çalışacak şekilde ayarlar. Bu şekilde, proje ortam sıcaklığını kontrol etmeyi ve fanın otomatik olarak devreye girmesini veya durmasını sağlamayı amaçlar.

**Not:** Hocam videoda çektim ama gönderirken kalite biraz düştü kusura bakmayın.

**Projemde Kullandığım Kodlar ve Açıklamaları**

#include <TimerOne.h>

const int lm35\_pin = A0; //a0 pininin lm35 sıcaklık sensörü için tanımlanması

float dutyCycle = 100; //darbe genişliği modülasyonunun sinyal genişliği(fanın hızı)

int temp\_adc\_val; // sensörden okunan analog değer değişkeni . 10 bitlik okuma ......

float temp\_val;

void setup() {

Serial.begin(9600); //seri haberleşme hızının 9600 olarak belirlenmesi(baudrate=9600)

pinMode(5, OUTPUT); //l298n motor surucusu IN1

pinMode(6, OUTPUT); //l298n motor surucusu IN2

pinMode(9, OUTPUT); //l298n motor surucusu EN1

//fan yönününü belirledim

digitalWrite(5, LOW);

digitalWrite(6, HIGH);

Timer1.initialize(40); // 40 us = 25 kHz

}void loop() {

temp\_adc\_val = analogRead(lm35\_pin); // sıcaklık okuma

temp\_val = (temp\_adc\_val \* 4.88); //adc değerini eşdeğer gerilime çevir

temp\_val = (temp\_val/10); // LM35, 10mv/°C çıkış veriyoruz

Serial.print("Temperature = ");

Serial.print(temp\_val);

if(temp\_val > 30)

{

Timer1.pwm(9, (dutyCycle / 100) \* 1023);

Serial.println("\t, FAN ÇALIŞIYOR");

}

else{

Timer1.disablePwm(9);

Serial.println("\t, FAN KAPALI");

}

delay(500);

}

**Fritzing**

**metin, dikdörtgen, ekran görüntüsü, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu**



1. **Arduino Uno**
2. **Fan**
3. **L298N Motor Sürücü**
4. **9V Pil**
5. **LM35 Sıcaklık Sensörü**
6. **Breadboard**

**Kaynaklar**

1. <https://www.tutorialspoint.com/timer1-based-pwm-in-arduino-uno>
2. <https://www.electronicwings.com/arduino/lm35-interfacing-with-arduino-uno>
3. <http://domoticx.com/arduino-pwm-frequency-and-timers/>
4. <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>