



KARADENİZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
MİKROİŞLEMCİLER LABORATUVARI



AKILLI KAVŞAK KONTROLÜ

1. GİRİŞ

Akıllı kavşak sistemleri günümüz şehir içi ulaşımının en önemli unsurlarından biridir. Artan araç trafiği ve yaya yoğunluğu, kavşakların yalnızca sabit zamanlamalarla değil, aynı zamanda sensörler ve mikrodnetleyiciler yardımıyla yönetilmesini gerekli kılmıştır. Bu nedenle trafik ışıklarının belirli koşullara göre uyarlanabilmesi hem güvenlik hem de verimlilik açısından önemli bir konudur.

Bu deney föyünde, Arduino simülatörü (Wokwi, Tinkercad veya Proteus) kullanılarak akıllı kavşak modelinin mikrodnetleyici tabanlı programlanması amaçlanmaktadır. Böylelikle temel C/Arduino programlama becerilerinin yanı sıra durum makineleri, zamanlama teknikleri ve giriş/çıkış işlemleri konusunda da uygulamalı deneyim kazandırılması hedeflenmektedir.

2. MİKROİŞLEMCİ TABANLI TRAFİK KONTROLÜ TEMELLERİ

Trafik ışıklarının kontrolü, şehir içi ulaşımın en kritik bileşenlerinden biridir. Geleneksel sistemlerde sabit zamanlı denetim yapılırken, akıllı kavşak uygulamaları sayesinde trafik yoğunluğu, yaya geçişleri ve acil durumlar mikrodnetleyiciler (Arduino) ve sensörler yardımıyla dinamik ve güvenli biçimde yönetilebilir. Böylece yaya talebi, araç yoğunluğu ve acil durumlar gibi koşullar gerçek zamanlı olarak dikkate alınır.

Mikroişlemciler, merkezî işlem birimi (CPU) işlevi gören, bellekten komutları alıp işleyen ve giriş/çıkış birimleriyle haberleşmeyi sağlayan entegre devrelerdir. Arduino, üzerinde mikrodnetleyici bulunan ve dijital/analog pinleri üzerinden çevre birimlerini kontrol etmeyi sağlayan bir geliştirme kartıdır. Bu deneyde, Arduino IDE ve/veya Wokwi/Tinkercad/Proteus gibi simülasyon ortamlarında; `pinMode()`, `digitalWrite()`, `digitalRead()` ve `INPUT_PULLUP` ile buton okuma; `delay()` ve `millis()` ile zamanlama; `Serial.begin()` ve `Serial.print()` ile seri haberleşme; ayrıca durum makinesi (FSM) mantığı ile faz geçişlerini yönetme konularını uygulamalı olarak öğreneceklerdir.

Arduino kartlarında giriş/çıkış (I/O) işlemleri, harici donanımların kontrol edilmesini ve sistemden veri alınmasını sağlayan temel mekanizmadır. Çıkış pinleri üzerinden LED'ler, motorlar veya ekranlar gibi elemanlar çalıştırılabilirken; giriş pinleri sayesinde buton, sensör veya potansiyometrelerden veri okunabilir. Bu yapı, deneylerde fiziksel bileşenlerin kolayca yönetilmesine olanak tanır ve karmaşık devreler basit kod satırlarıyla

kontrol edilebilir hale gelir. Örneğin, bir LED'i yakmak için yalnızca ilgili pine çıkış yönü atanır ve dijital değer gönderilir, bir butondan okuma yapmak içinse giriş olarak tanımlanmış pine sinyalin ulaşip ulaşmadığı kontrol edilir. Böylece hem donanım kurulumu hem de yazılım tarafı öğrenciler için anlaşılır ve yönetilebilir bir bütünlük oluşturur. Kodlanarak yapılan bu giriş/çıkış işlemleri, gerçek hayatta kullanılan otomasyon sistemlerinin küçük bir prototipi niteliğinde olup, deneysel öğrenmeyi kolaylaştırır. Trafik ışığı simülasyonunda LED'ler, yaya butonları ve ekranda seri port çıktıları kullanılabilir. Tablo 1'de Arduino temel komutları verilmektedir.

Tablo 1. Arduino Temel Komutları

Arduino Komutu	Görevi
pinMode(pin, OUTPUT/INPUT)	Pin yönü ayarlama
digitalWrite(pin, HIGH/LOW)	LED veya çıkış kontrolü
digitalRead(pin)	Buton/sensör okuma
delay(ms)	Milisaniye cinsinden bekleme
Serial.print()	Seri haberleşme

Arduino'da zamanlama işlemleri, devrelerde olayların belirli aralıklarla gerçekleşmesini sağlamak için kullanılır. En temel yöntem, delay(ms) fonksiyonudur ve belirtilen süre boyunca programın çalışmasını durdurarak istenen beklemeyi gerçekleştirir. Örneğin, bir LED'in 1 saniye yanıp sönmeye için delay(1000) komutu kullanılabilir. Ancak bu yöntem bloklayıcı (blocking) olduğu için, bekleme süresi boyunca başka hiçbir işlem yapılamaz. Daha gelişmiş ve çoklu görev gerektiren senaryolarda ise millis() fonksiyonu tercih edilir. millis(), kart açıldığından beri geçen süreyi milisaniye cinsinden döndürür ve bu sayede kesintisiz (non-blocking) zamanlama yapılabilir. Böylece bir yandan LED yanıp sönerken aynı anda buton girişleri kontrol edilebilir veya seri port üzerinden veri okunabilir. Bu yaklaşım, özellikle trafik ışığı simülasyonlarında hem LED'lerin hem de yaya butonlarının eşzamanlı yönetilmesine olanak tanır ve gerçek hayattaki sistemlere daha yakın bir çalışma mantığı sağlar.

Durum makineleri (FSM – Finite State Machine), bir sistemin farklı durumlarını ve bu durumlar arasındaki geçişleri tanımlamak için kullanılan mantıksal bir yaklaşımdır. FSM'de sistem her an yalnızca bir durumda bulunur ve belirli girdilere, yani sensör, zamanlayıcı, buton veya başka bir dış etkiye bağlı olarak bir durumdan diğerine geçiş yapar. Bu yaklaşım özellikle karmaşık süreçleri basitleştirerek anlaşılır hale getirmesi ve kontrolü kolaylaştırması açısından önemlidir. Trafik ışıkları buna en güzel örneklerden biridir; kırmızı, yeşil ve sarı ışıkların her biri bir durumu temsil eder ve zamanlayıcıya bağlı olarak sırayla diğer duruma geçiş yaparlar. Mikrodenetleyici tabanlı sistemlerde FSM yaklaşımı kullanılarak, her adım net bir şekilde tanımlanır ve sistemin güvenilir bir şekilde çalışması sağlanır.

3. DENEY TASARIMI VE UYGULANMASI



Sistem Tanımı

- Tek kavşak: Kuzey–Güney (KG) ve Doğu–Batı (DB).
- Her yönde: Kırmızı, Sarı, Yeşil LED.
- Yaya ışığı: Kırmızı, Yeşil LED.
- Yaya butonu.

Pin Tanımları

- KG ışıkları: 2 (Kırmızı), 3 (Sarı), 4 (Yeşil)
- DB ışıkları: 5 (Kırmızı), 6 (Sarı), 7 (Yeşil)
- Yaya ışıkları: 8 (Kırmızı), 9 (Yeşil)
- Yaya butonu: 10

Uygulama

- Arduino simülatöründe (Tinkercad, Wokwi veya Proteus) devre kurulacaktır.
- LED ve buton bağlantıları pin tanımlarına göre yapılacaktır.
- Kod yüklenerek simülasyon başlatılacaktır.
- Gecikme süreleri delay() değerleri değiştirilerek ayarlanabilir. Işıklar için bekleme sürelerini ayarlayınız.
- Yaya butonuna basıldığında sistemin kırmızı ışık verip yaya ışığının yeşil yanmasını test ediniz.

4. DENEYE HAZIRLIK SORULARI

1. Arduino'da giriş/çıkış işlemleri hangi fonksiyonlarla yapılır? Günlük hayattan örnek veriniz.
2. Arduino'da gecikme oluşturmak için hangi yöntemler kullanılır?
3. Trafik ışıkları kontrolünde neden durum makinesi (FSM) yaklaşımı tercih edilir?
4. Yaya ve araç ışıklarının aynı anda yeşil yanmasının önlenmesi için nasıl bir mantık geliştirilmelidir?
5. Fail-safe (hata güvenli) çalışma kavramı nedir, trafik ışığı sistemlerinde neden önemlidir?

5. DENEY RAPORU

1. Hazırlık sorularının cevaplanması
2. Deney sorusunun çözümünü içeren Arduino kodu
3. Simülasyon ekranından alınan gözlem sonuçları
4. Deney esnasında karşılaşılan zorluklar
5. Rapor 1 hafta içinde deney sorumlusuna uzem veya deney sorumlusunun belirttiği platform üzerinden teslim edilir. Raporlar, bir sonraki hafta deneyine kadar teslim edilebilir. Kopya rapor veren gruplar gerekli sorumluluğu üstlenmiş olurlar.

6. DENEY SORUSU

Trafik ışığı FSM'inize "fail-safe" modu ekleyiniz. Yaya butonunu (D10, INPUT_PULLUP) hata butonu olarak kullanın. Butona basıldığında sistem normal döngüyü durduracak ve KG/DB sarı ışıkları 500 ms aralıklarla yanıp sönecek (üniversal uyarı modu). Fail-safe aktifken diğer ışıklar sönmük kalmalıdır.

Trafik yoğunluğuna göre ışıkların sürelerini dinamik olarak ayarlamak için nasıl bir yaklaşım geliştirilmelidir? Verilen Arduino kodunu güncelleyerek yoğunluğu manuel olarak ayarlayınız ve ışık sürelerini buna göre değiştiriniz.