



KARADENİZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
MİKROİŞLEMCİLER LABORATUVARI



AKILLI KAVŞAK KONTROLÜ

1. GİRİŞ

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

Bu deney föyünde, mikroişlemci veya mikrodenetleyici tabanlı bir geliştirme platformu kullanılarak akıllı kavşak kontrol sisteminin tasarlanması ve programlanması amaçlanmaktadır. Böylelikle temel C/Arduino programlama becerilerinin yanı sıra giriş/çıkış işlemleri, zamanlama teknikleri, durum makineleri ve buton tabanlı kontrol konularında uygulamalı deneyim kazandırılması hedeflenmektedir.

2. MİKROİŞLEMCİ VE MİKRODENETLEYİCİ TABANLI TRAFİK KONTROLÜ TEMELLERİ

Trafik ışıklarının kontrolü, şehir içi ulaşımın güvenli ve düzenli biçimde sürdürülebilmesi açısından önemli bir uygulama alanıdır. Geleneksel trafik kontrol sistemlerinde ışık süreleri çoğunlukla önceden belirlenen sabit zamanlara göre düzenlenirken, mikroişlemci ve mikrodenetleyici tabanlı sistemlerde trafik yoğunluğu, yaya talepleri, sensörlerden alınan bilgiler ve acil durumlar gibi farklı koşullar dikkate alınarak daha esnek kontrol mekanizmaları geliştirilebilir. Böylece trafik ışıklarının çalışma süreleri ve geçiş durumları değişen koşullara göre düzenlenebilir.

Mikroişlemciler temel olarak komutları işleyen, aritmetik ve mantıksal işlemleri gerçekleştiren ve sistemdeki diğer birimlerin çalışmasını yöneten işlem birimleridir. Mikrodenetleyiciler ise işlemci çekirdeğinin yanı sıra bellek, dijital ve analog giriş/çıkış birimleri, zamanlayıcılar ve çeşitli haberleşme arabirimlerini aynı yapı içerisinde barındırır. Bu özellikleri sayesinde mikrodenetleyiciler sensör, buton, LED, motor ve benzeri çevre birimlerinin doğrudan kontrol edildiği gömülü sistem uygulamalarında yaygın olarak kullanılmaktadır. Intel Galileo gibi geliştirme kartları mikroişlemci tabanlı bir yapı sunarken, Arduino Uno gibi kartlar mikrodenetleyici tabanlıdır. Her iki yaklaşımda da girişlerden alınan bilgiler işlenerek belirlenen kontrol algoritmasına göre çıkış elemanları yönetilebilir.

Giriş/çıkış (I/O) işlemleri, mikroişlemci ve mikrodenetleyici tabanlı sistemlerin fiziksel çevre ile etkileşim kurmasını sağlayan temel mekanizmalardan biridir. Çıkış birimleri aracılığıyla LED'ler, motorlar, röleler, buzzerlar, ekranlar ve çeşitli eyleyiciler kontrol edilebilirken; giriş birimleri üzerinden buton, anahtar, potansiyometre, sıcaklık, ışık, mesafe ve hareket sensörleri gibi farklı elemanlardan veri alınabilir. Örneğin bir otomasyon sisteminde sıcaklık sensöründen alınan bilgiye göre fan çalıştırılabilir, bir mesafe sensörünün algıladığı nesneye göre motor durdurulabilir veya bir buton aracılığıyla belirli bir işlem başlatılabilir. Girişlerden alınan bilgiler program içerisinde değerlendirilerek belirlenen koşullara göre çıkışların kontrol edilmesi, gömülü sistemlerin temel çalışma prensibini oluşturur. Bu yapı sayesinde donanım ve yazılım birlikte çalışarak gerçek hayattaki kontrol ve otomasyon sistemlerinin küçük ölçekli modellerinin oluşturulmasına olanak sağlar. Bu deneyde kullanılan temel komutlar Tablo 1'de verilmiştir.

Tablo 1. Temel Komutları

Komut	Görevi
pinMode(pin, OUTPUT/INPUT)	Pin yönünü belirleme
digitalWrite(pin, HIGH/LOW)	Dijital çıkışı kontrol etme
digitalRead(pin)	Dijital giriş değerini okuma
delay(ms)	Milisaniye cinsinden bekleme
Serial.begin()	Seri haberleşmeyi başlatma
Serial.print() / Serial.println()	Seri porta veri gönderme

Trafik kontrol uygulamalarında zamanlama önemli bir diğer konudur. Işıkların belirli sürelerle açık veya kapalı tutulması için en temel yöntemlerden biri delay(ms) fonksiyonudur. Bu fonksiyon programın belirtilen süre boyunca beklemesini sağlar. Örneğin delay(1000) komutu programın yaklaşık bir saniye beklemesine neden olur. Ancak delay() bloklayıcı bir fonksiyondur; bekleme süresi boyunca programın diğer işlemleri gerçekleştirmesi mümkün değildir. Bu nedenle aynı anda birden fazla olayın takip edilmesi gereken sistemlerde yalnızca delay() kullanılması bazı girişlerin zamanında algılanamamasına neden olabilir.

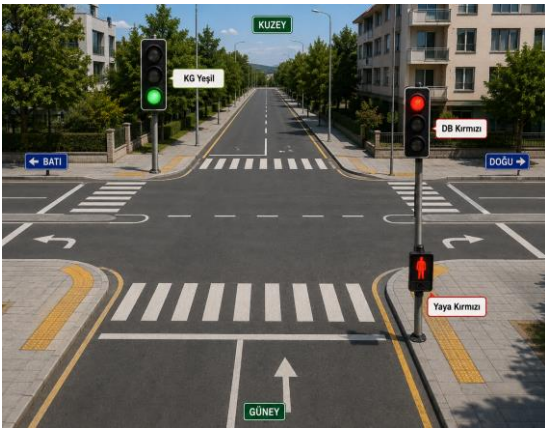
Daha esnek zamanlama gerektiren uygulamalarda millis() fonksiyonundan yararlanılabilir. millis(), sistem çalışmaya başladığından itibaren geçen süreyi milisaniye cinsinden verir. Bu değer kullanılarak programın çalışması durdurulmadan belirli zaman aralıkları takip edilebilir. Böylece trafik ışıklarının süreleri kontrol edilirken aynı zamanda buton veya sensör girişlerinin okunması gibi başka işlemler de sürdürülebilir. Bu tür bloklamayan zamanlama yöntemleri, gerçek zamanlı tepki gerektiren kontrol uygulamalarında daha esnek bir yapı sağlar.

Trafik ışığı gibi birden fazla çalışma durumuna sahip sistemlerin kontrolünde durum makinesi (FSM – Finite State Machine) yaklaşımından yararlanılabilir. FSM, sistemin belirli durumlara ayrılmasını ve bu durumlar arasındaki geçişlerin belirli kurallara göre gerçekleştirilmesini sağlar. Bir kavşak sistemi, geçen süre, buton girişi veya sensör bilgisi gibi

koşullara göre bir durumdan diğerine geçer. Bu yaklaşım, kontrol algoritmasının daha düzenli ve anlaşılır biçimde oluşturulmasına ve farklı çalışma koşullarının yönetilmesine yardımcı olur.

Sistemin çalışması sırasında değişkenlerin ve giriş/çıkış durumlarının izlenmesi için seri haberleşmeden de yararlanılabilir. Serial.begin() ile seri haberleşme başlatılırken, Serial.print() veya Serial.println() fonksiyonları kullanılarak sistemin çalışma durumu bilgisayara aktarılabilir. Bu yöntem özellikle programın test edilmesi, hata ayıklama ve buton gibi girişlerin doğru algılanıp algılanmadığının kontrol edilmesi sırasında yararlıdır.

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.

Donanım Bağlantıları

1. Breadboard üzerine Kuzey–Güney, Doğu–Batı ve yaya trafik ışıklarını temsil edecek LED'ler yerleştirilir. Her LED için 220 Ω direnç kullanılır.
2. LED'lerin anot (+) uçları 220 Ω direnç üzerinden Galileo kartının ilgili dijital pinlerine aşağıdaki şekilde bağlanır:
KG ışıkları: Dijital Pin 2 (Kırmızı), Dijital Pin 3 (Sarı), Dijital Pin 4 (Yeşil)
DB ışıkları: Dijital Pin 5 (Kırmızı), Dijital Pin 6 (Sarı), Dijital Pin 7 (Yeşil)
Yaya ışıkları: Dijital Pin 8 (Kırmızı), Dijital Pin 9 (Yeşil)
3. LED'lerin katot (–) uçları Galileo kartının GND hattına bağlanır.
4. Yaya geçişi için dört bacaklı push button breadboard üzerine yerleştirilir. Butonun bir tarafı Dijital Pin 10'a, karşı tarafı ise GND'ye bağlanır.
5. Intel Galileo kartının güç bağlantısı yapılır. Öncelikle kartın güç adaptörü bağlanarak Galileo'nun açılması beklenir. Daha sonra USB kablosu ile Galileo kartı bilgisayara bağlanır.

Programın Yüklmesi ve Test Edilmesi

1. Galileo kartı USB kablosu ile bilgisayara bağılyken Arduino IDE açılır.
2. Arduino IDE içerisinde Intel Galileo kartına uygun kart seçimi yapılır:
Tools → Board → Intel Galileo / Galileo Gen 2
3. Galileo'nun bağılı olduğı haberleşme portu seçilir:
Tools → Port → ilgili COM portu
4. Deney sayfasında verilen kaynak kodu Arduino IDE'ye aktarılır ve Upload komutu kullanılarak Galileo kartına yüklenir.
5. Yükleme işleminin başarıyla tamamlandığı Arduino IDE'nin Output bölümünden kontrol edilir. Program Galileo üzerinde çalışmaya başladığında trafik ışıklarının sıralı olarak değişmesi gözlemlenir.
6. Trafik ışıklarının yanma süreleri program içerisinde bulunan delay() değerleri değiştirilerek ayarlanır. Farklı bekleme süreleri denenerek sistemin davranışı gözlemlenir.
7. Normal trafik çevriminin farklı aşamalarında yaya butonuna basınız. Buton isteğinin sistem tarafından nasıl algılandığını ve sistemin verdiği tepkiyi gözlemleyiniz.

4. DENEYE HAZIRLIK SORULARI

1. Arduino'da dijital giriş/çıkış işlemleri hangi fonksiyonlarla gerçekleştirilir? Bir giriş ve bir çıkış elemanına örnek veriniz.
2. Arduino'da zamanlama işlemleri için delay() ve millis() fonksiyonları nasıl kullanılır? Bu iki yöntemin temel farkı nedir?
3. Durum makinesi (FSM – Finite State Machine) nedir? Trafik ışıklarının kontrolünde kullanılmasının avantajı nedir?
4. Akıllı kavşak sistemlerinde sabit zamanlı kontrol yerine sensör destekli mikroişlemci/mikrodenetleyici tabanlı kontrol kullanılmasının avantajları nelerdir?

5. DENEY RAPORU

1. Hazırlık sorularının cevapları
2. Deney sorularının çözümünü içeren Arduino kaynak kodu
3. Deney sırasında gözlemlenen sistem davranışları ve sonuçların yorumlanması
4. Deney sırasında karşılaşılan zorluklar

Rapor Teslimi: Rapor, deney tarihinden itibaren en geç bir hafta içinde UZEM veya deney sorumlusunun belirttiğı platform üzerinden teslim edilmelidir. Kopya rapor veren gruplar gerekli sorumluluğı üstlenmiş olurlar.

6. DENEY SORULARI

Verilen trafik ışığı kontrol programına bir fail-safe (hata güvenli) çalışma modu ekleyiniz. Bu aşamada D10 pinine bağlı yaya butonunu fail-safe modunu tetiklemek amacıyla kullanınız. Butona basıldığında normal trafik çevrimi durdurulmalı ve KG ile DB yönlerindeki sarı ışıklar 500 ms aralıklarla sürekli yanıp sönmelidir. Fail-safe modu aktifken diğer tüm ışıklar sönmük kalmalıdır. Sistem yeniden başlatılıncaya kadar fail-safe modu aktif kalmalıdır.

Trafik yoğunluğuna göre ışıkların sürelerini dinamik olarak ayarlamak için nasıl bir yaklaşım geliştirilebilir? Farklı trafik yoğunluğu seviyelerini temsil eden değişkenler tanımlayarak, verilen Arduino kodunda ışık sürelerinin bu yoğunluk seviyelerine göre değişmesini sağlayınız.