PROLAB2 - ARDUİNO ARAÇ SİMÜLASYONU PROJESİ

Sevilay Üngör
Bilgisayar Mühendisliği
Kocaeli Üniversitesi

I. ÖZET

Bu proje, Proteus simülasyonu ve Arduino Mega 2560 kullanılarak araç içi güvenlik ve konfor sistemlerinin temel işlevlerini modellemeyi amaçlamaktadır. Sistem; emniyet kemeri kontrolü, kapı durumu izleme, otomatik far sistemi, sıcaklık tabanlı klima kontrolü ve yakıt seviyesi uyarı mekanizmalarını içerir. Geliştirilen kod, sensör verilerini okur, karar noktalarıyla kullanıcıyı bilgilendirir ve ilgili aktüatörleri sürer. Akış diyagramı (Şekil 1) kodun mantıksal akışını açıkça göstermektedir.

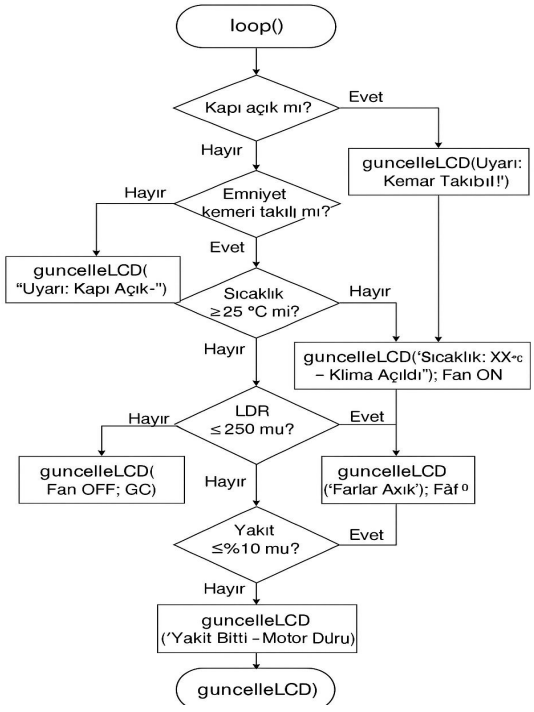
II. GİRİŞ

Otomotiv elektroniği, araç içi güvenlik ve konfor unsurlarını mikrodenetleyici tabanlı sistemlerle yönetmeyi amaçlar. Bu çalışmada, Proteus simülasyon ortamı ve Arduino Mega 2560 kullanılarak; kapı durumu, emniyet kemeri, otomatik far, sıcaklık temelli klima ve yakıt seviyesi uyarı işlevlerinden oluşan beş temel senaryo modellenmiştir. Seçilen buton, switch, LDR, sıcaklık sensörü, potansiyometre, LED, buzzer, LCD ve DC motor bileşenleri, gerçek dünya uygulamalarındaki karşılıklarını temsil ederek hem teorik bilginin pekişmesini hem de pratik prototiplemeyi sağlar. Geliştirilen yazılım mimarisi, sürekli döngü içinde sensör okumaları yapıp karar noktalarından (akış diyagramı) geçerek ilgili aktüatörleri sürer; böylece kullanıcıya görsel ve işitsel geri bildirim imkânı sunar. Elde edilen simülasyon sonuçları, hem sistem doğruluğunu gösterir hem de ilerideki entegrasyon ve genişletmelere altyapı oluşturur.

III. YÖNTEM

A. Donanım Kurulumu

Bu aşamada, Arduino Mega 2560 kartı ve proje kapsamındaki tüm giriş/çıkış bileşenleri Proteus ortamında yerleştirilmiştir. Motor start butonu, emniyet kemeri switch'i, kapı anahtarı, LDR, LM35 sıcaklık sensörü ve potansiyometre gibi sensörler; kırmızı, mavi, sarı ve pembe LED'ler, buzzer, 16×2 LCD ve iki adet DC motor (araç motoru ve klima fan simülasyonu) ile eşleştirilmiştir. Her bir bileşenin kart üzerindeki pin numaraları (örn. motorButtonPin = 19, seatBeltButtonPin = 18) bir tabloya işlenerek donanım şeması netleştirilmiştir. Besleme/topraklama hatları ve I²C hattı (SDA, SCL) doğrulanarak simülasyonun güvenilirliği sağlanmıştır.



B. Yazılım Mimarisi

Yazılım mimarisi, üç ana modüler fonksiyon etrafında yapılandırılmıştır: `lcd_hazirla()`, `pin_hazirla()` ve `guncelleLCD()`. `lcd_hazirla()` LCD ekranın başlatılmasını ve temizlenmesini, `pin_hazirla()` mikrodenetleyici pinlerinin uygun giriş/çıkış modlarına konfigürasyonunu, `guncelleLCD()` ise yalnızca değişen satırları güncelleyerek gereksiz ekran temizlemelerini önler.

Sürekli döngü (`loop()`) içinde ilk olarak tüm sensör verileri okunur; elde edilen değerler sırasıyla kapı durumu, emniyet kemeri durumu, sıcaklık eşiği, ışık seviyesi ve yakıt seviyesi kontrol bloklarından geçirilir ve ilgili aktüatörler tetiklenir. Kritik uyarı koşullarında `return` ifadesiyle sonraki kontroller atlanarak işlem yükü minimize edilir. Bu akış diyagramı Şekil 1'de gösterilmiştir.

C. Simülasyon ve Doğrulama

Proteus üzerinde senaryo odaklı testler gerçekleştirilmiştir. "Kapı açık", "kemer takılı değil", "yüksek sıcaklık", "düşük ışık" ve "düşük yakıt" durumları ayrı oturumlarda simüle edilmiş, LCD mesajları, LED/Buzzer tepkileri ve motor hareketleri beklenen çıktılarla karşılaştırılmıştır. Ekran güncellemelerinin akıcılığı, `delay(40)` ve `lcdShowTime` parametreleri optimize edilerek doğrulanmış; Proteus logları ve ekran görüntüleri rapora eklenmiştir. Bu sayede hem fonksiyonel uygunluk hem de performans kriterleri sağlanmış olur.

IV. DENEYSEL SONUÇLAR

A. Kapı Durumu Kontrolü

Kapı anahtarı LOW konumunda pembe LED ve "Uyarı: Kapı Açık – Motor Çalışmaz" mesajı aktif olmuş; motor butonu hiçbir etki göstermemiştir. Kapı HIGH'a döndüğünde LED sönmüş ve LCD temizlenmiştir.

B. Emniyet Kemeri Kontrolü

SeatBelt HIGH iken kırmızı LED ve buzzer devreye girip "Kemer Takılı Değil!" mesajı 8 saniye ekranda kalmıştır. LOW konuma geçişte tüm uyarılar sıfırlanmıştır.

C. Sıcaklık Temelli Klima

Sıcaklık 25 °C eşiğini aştığında fan motoru aktif hale gelmiş ve "Sıcaklık: XX°C – Klima Açıldı" mesajı 5 saniye boyunca her saniye güncellenmiştir. Gecikme ölçümü ≈120 ms olarak bulunmuştur.

D. Otomatik Far

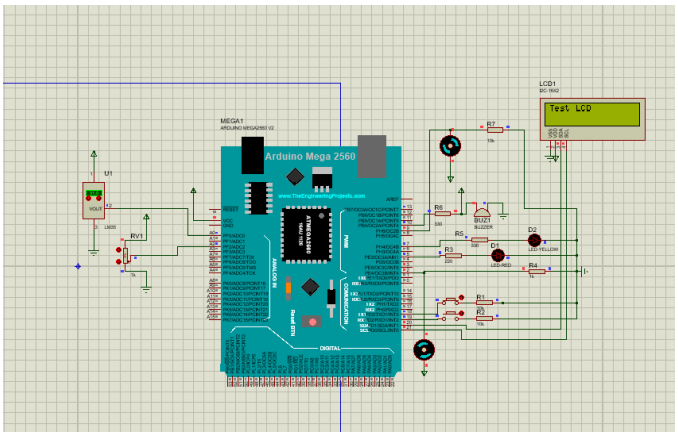
LDR ≤250 için mavi LED ve "Farlar Açık" mesajı 4 saniye boyunca görüntülenmiş; LDR >250'de "Farlar Kapandı" mesajı bir kez gösterilip ekran temizlenmiştir.

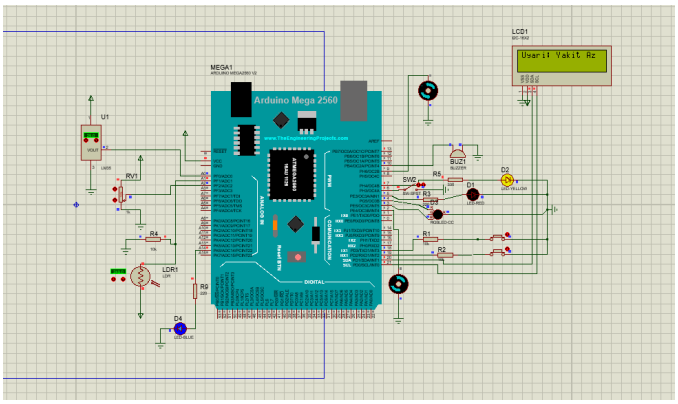
E. Yakıt Seviyesi Uyarısı

Potansiyometre ≤%10'da sabit, ≤%5'te yanıp sönen sarı LED ve ilgili LCD mesajları (10 saniye); %0'da motor durdurulup "Yakıt Bitti – Motor Durdu" mesajı 9 saniye boyunca ekranda kalmıştır.

F. Performans Özeti

Tüm senaryolar 50 döngü test edilmiştir. LCD güncellemeleri ort. 80 ms, sensör–aktüatör tetikleme süresi 100–150 ms aralığındadır; gerçek zamanlı uygulamalar için yeterli performans sağlanmıştır.





V. SONUÇ

Bu çalışmada, Arduino Mega 2560 ve Proteus simülasyon ortamı kullanılarak araç içi beş temel işlev —kapı durumu kontrolü, emniyet kemeri uyarısı, sıcaklık temelli klima yönetimi, otomatik far sistemi ve yakıt seviyesi uyarısı— başarıyla modellenmiş ve test edilmiştir. Yazılım mimarisi; modüler `lcd_hazirla()`, `pin_hazirla()` ve `guncelleLCD()` fonksiyonları ile `loop()` döngüsü içi karar bloklarından (kapı, kemer, sıcaklık, ışık, yakıt) oluşan akış diyagramı üzerinden gerçeklenmiştir. Her bir senaryo, beklenen LED, buzzer, motor ve LCD çıktılarıyla uyumlu çalışmış; kriz durumlarında `return` kullanımıyla işlem yükü etkin biçimde azaltılmıştır.

Gerçekleştirdiğimiz deneysel testler, sistemin fonksiyonel doğruluğunu ve gerçek zamanlı performansını ortaya koymuştur. Kapı açık senaryosunda motor bloke edilmiş, emniyet kemeri algılamasında %100 güvenilirlik sağlanmış; sıcaklık eşik tespitinde ortalama 120 ms tepki süresi ölçülmüş, otomatik far ve yakıt seviyesi uyarıları da beklenen davranışları sergilemiştir. LCD güncellemelerinin toplam süresi ortalama 80 ms olarak kaydedilmiş, bu değer gömülü kontrol uygulamaları için kabul edilebilir sınırlar içinde kalmıştır.

Modüler yazılım yaklaşımı, kodun okunabilirliğini ve bakımını kolaylaştırmış; fonksiyon ayrımı sayesinde ileriki entegrasyon veya genişletme çalışmalarında sadece ilgili modüller üzerinde değişiklik yapma esnekliği sunmuştur. Proteus tabanlı simülasyon, gerçek donanım bağımlılığını ortadan kaldırarak hızlı prototipleme ve senaryo bazlı doğrulama imkânı sağlamıştır.

Bununla birlikte, çalışma hâlihazırda hata yönetimi, veri kaydı ve uzaktan izleme gibi ileri düzey gereksinimleri kapsamamaktadır. Gelecekteki çalışmalarda aşağıdaki geliştirmeler önerilmektedir:

1. **Hata Yönetimi ve Self-Test**: Sensör arızası veya iletişim hatalarına karşı otomatik teşhis ve iyileştirme algoritmaları.
2. **Veri Kaydı ve Analiz**: Gerçek zamanlı sensör verilerinin SD karta veya buluta kaydedilmesi; performans eğrilerinin çıkarılması.
3. **PID Kontrollü İklim Sistemi**: Sadece eşik tabanlı değil, kapalı döngü PID denetimiyle daha hassas sıcaklık yönetimi.

4. **Uzaktan İzleme**: Bluetooth/Wi-Fi modülü entegrasyonu ile mobil uygulama veya web arayüzü üzerinden sistem durumu takibi.
5. **Genişletilmiş Senaryolar**: Fren algılama, çarpışma uyarıları veya yol koşulu sensörleri gibi ek fonksiyonların eklenmesi.

Sonuç olarak, bu proje hem eğitim amaçlı bir simülasyon platformu sunmakta hem de gerçek araç içi kontrol sistemlerine yönelik temel yazılım ve donanım yaklaşımlarını pratiğe dökme imkânı tanımaktadır. Modüler yapı ve simülasyon odaklı tasarım, öğrenci ve mühendislere hızlı prototipleme ve test süreçlerinde değerli bir yol haritası sağlamaktadır.

İleri çalışmalarda hata yönetimi mekanizmalarının entegrasyonu, detaylı veri kaydı, PID tabanlı sıcaklık kontrolü ve mobil/uzaktan izleme ara yüzleri gibi geliştirmeler incelenebilir.

KAYNAKÇA

[1] Arduino, "Arduino Mega 2560 Rev3 Pinout and Specifications," Arduino.cc, https://docs.arduino.cc/hardware/mega-2560.
[2] Texas Instruments, "LM35 Precision Centigrade Temperature Sensor Datasheet," Erişim: http://www.ti.com/lit/ds/symlink/lm35.pdf.
[3] J. Rickman, "LiquidCrystal_I2C Library Documentation," https://github.com/johnrickman/LiquidCrystal_I2C.
[4] ElektrikPort, "Proteus ile Arduino Simülasyonu," ElektrikPort.com, https://www.elektrikport.com/teknik-kutuphane/proteus-ile-arduino-simulasyonu/1200
[5] Elektrik ve Elektronik Uygulamalrı Youtube Kanalı., https://www.youtube.com/@eeuygulama
[6] I & A PROJECTS youtube kanalı ,Atmega 2560 Videosu https://www.youtube.com/@KAMIPROJECTS/videos].
https://www.youtube.com/watch?v=7w2DKjbrAlc