Arduino HIL Test Uygulaması Raporu

1. Proje Tanımı

Bu proje, fiziksel bir Arduino kartı ile Proteus ortamında simüle edilmiş bir Arduino'nun, grafik kullanıcı arayüzü (GUI) aracılığıyla entegre şekilde haberleşmesini sağlayan bir sistemdir. Proje, "Hardware-in-the-Loop (HIL)" prensibiyle çalışarak eş zamanlı simülasyon ve kontrol imkanı sunmaktadır.

2. Proje Amacı

Projenin temel amacı, simüle edilen araç sistemlerinden gelen verilerin GUI aracılığıyla analiz edilerek, gerçek Arduino kartına uygun komutların iletilmesi ve buna karşılık gelen donanımsal çıktılarla sürecin izlenmesidir.

3. Sistem Gereksinimleri ve Bileşenler

3.1 Proteus Simülasyonu (Fig.1)

- 5 adet buton: Motor, klima ve acil durum gibi araç fonksiyonlarını tetikler.
- 2 adet potansiyometre: Araç hızı ve yakıt seviyesi girişlerini temsil eder.
- Simüle edilen Arduino kartı: Buton ve potansiyometre verilerini UART üzerinden GUI'ye gönderir.

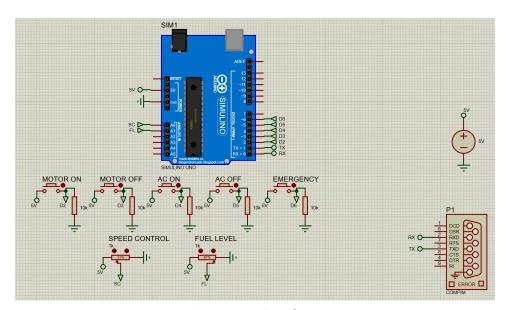


Fig.1 - Proteus Simülasyonu

3.2 Grafik Kullanıcı Arayüzü (GUI) (Fig.2)

- Proteus'tan gelen verileri UART ile okur ve anlık olarak kullanıcıya sunar.
- Gerçek Arduino'ya uygun komutları göndererek kontrol sağlar.
- Sistem durumlarını simgeler, göstergeler ve grafikler ile görselleştirir.
- Mesajları zaman damgalı şekilde kaydeder ve mesaj monitörü panelinde listeler.

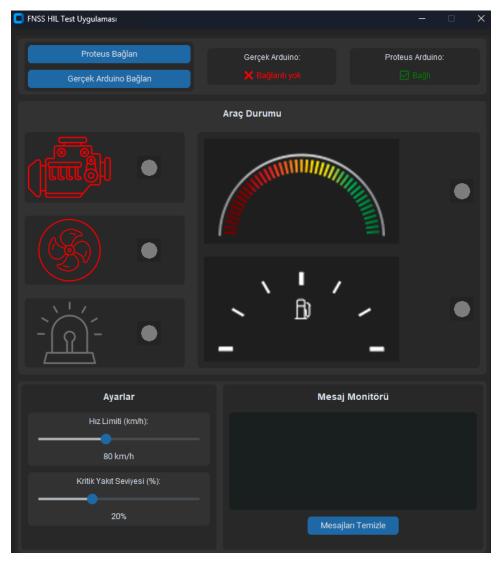


Fig.2 - Python GUI Uygulaması

3.3 Gerçek Arduino (Fig.3-6)

GUI'den gelen komutlara göre dijital çıkış pinlerinden LED kontrolü sağlar.

-Sistem çıktıları:

- Motor açık: Yeşil LED yanar (Fig.3)
- Motor kapalı: Yeşil LED söner

- Klima açık: Mavi LED yanar (Fig.4)
- Klima kapalı: Mavi LED söner
- Acil durum: Kırmızı LED sürekli yanar (Fig.5-6)
- Hız limiti aşıldıysa: Hız uyarı LED'i yanar (Fig.6)
- Yakıt seviyesi kritik altına düştüyse: Yakıt uyarı LED'i yanar (Fig.6)

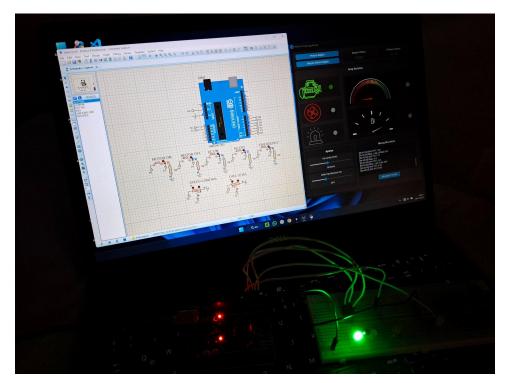


Fig.3 - Motor Açık

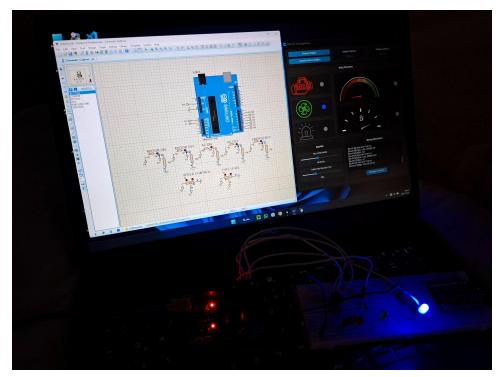


Fig.4 - Klima Açık

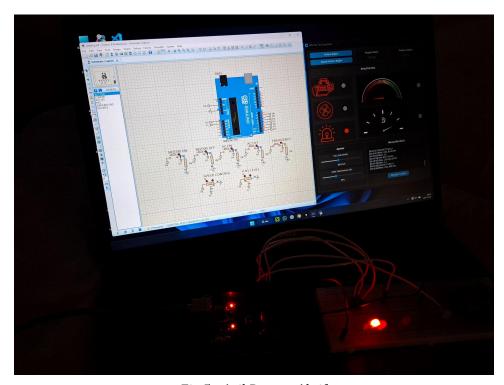


Fig.5 - Acil Durum Aktif

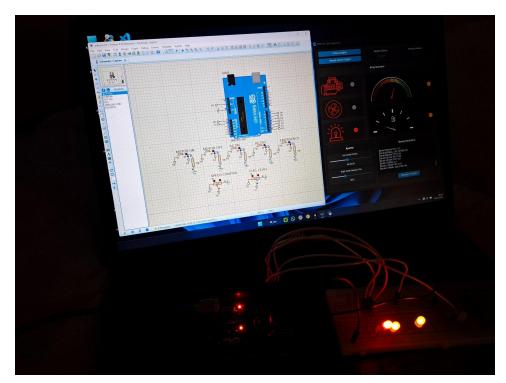


Fig.6 - Acil Durum, Yakıt Uyarısı ve Hız Uyarısı Aktif

4. Test Senaryoları

4.1 Dijital Girişler (Butonlar)

- Buton 1: Motor Başlatıldı
- Buton 2: Motor Durduruldu
- Buton 3: Klima Açıldı
- Buton 4: Klima Kapatıldı
- Buton 5: Acil Durum

Öncelik kuralı: Buton 2 ve 4, buton 1 ve 3'e göre önceliklidir.

Örneğin: Buton 1 aktifken Buton 2'ye basılırsa yeşil LED sönmelidir. (Fig.7)

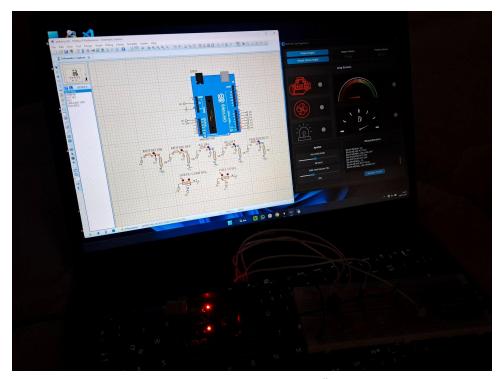


Fig.7 - Buton 2, Buton 1'e Göre Öncelikli

4.2 Analog Girişler (Potansiyometreler)

- Potansiyometre 1: Hız kontrolünü temsil eder. Hız limiti GUI'den belirlenir. Aşılırsa LED yanar. (Fig.6)
- Potansiyometre 2: Yakıt seviyesini temsil eder. Kritik eşik GUI'de belirlenir. Altına inerse LED yanar. (Fig.6)

4.3 UART Haberleşme (Fig.8)

- Simüle Arduino ve Gerçek Arduino GUI ile UART üzerinden haberleşir.
- GUI her Arduino'ya farklı portlardan bağlanır ve gelen veriye göre aksiyon alır.
- Her mesaj GUI'de saat bilgisiyle kaydedilir.

```
[06:45:44] REAL: LED_GREEN:OFF
[06:45:48] REAL: 2,1
[06:45:48] SIM: BUTTON:2:0
[06:45:48] REAL: LED_GREEN:ON
[06:45:50] REAL: 2,0
[06:45:50] SIM: BUTTON:1:0
[06:45:50] REAL: LED_GREEN:OFF
[06:45:51] REAL: 3,1
[06:45:51] SIM: BUTTON:3:1
[06:45:51] REAL: LED_BLUE:ON
[06:45:55] REAL: 3,0
[06:45:55] SIM: BUTTON:3:0
[06:45:55] REAL: LED BLUE:OFF
[06:45:58] OLAY: Emergency (buton basili - latch)
[06:45:58] REAL: 4,1
[06:45:58] SIM: BUTTON:5:1
[06:45:58] REAL: LED_RED:ON
[06:46:03] OLAY: Emergency (buton birakildi)
[06:46:03] SIM: BUTTON:5:0
[06:46:05] REAL: 5,1
[06:46:05] SIM: POT:1:91
[06:46:05] REAL: LED_SPEED:ON
[06:46:08] SIM: POT:2:30
[06:46:10] SIM: POT:2:25
[06:46:10] SIM: POT:2:21
[06:46:11] REAL: 6,1
[06:46:11] SIM: POT:2:15
[06:46:11] REAL: LED_FUEL:ON
[06:46:18] REAL: Serial thread hatası: Bağlantı koptu
```

Fig.8 - Seri Monitör Log Kayıt Örneği