

**IOT UYGULAMASI İLE AKILLI EV
OTOMASYONU**
10. HAFTA İLERLEME RAPORU
14/04/2024 - 06/05/2024

Names : Emre DAĞLI
Numbers 184208025

1. Rapor konusu:
IOT Uygulaması kodlaması ve devrenin test edilmesi

2. Raporun Özeti:
Ara yüzü tasarlanan IOT uygulamasının kod kısmının yazılıp butonların işlevlerinin belirlenmesi.
Tasarladığımız devrenin IOT uygulaması ile test edilmesi

3. Rapor:
Devre Tasarımı

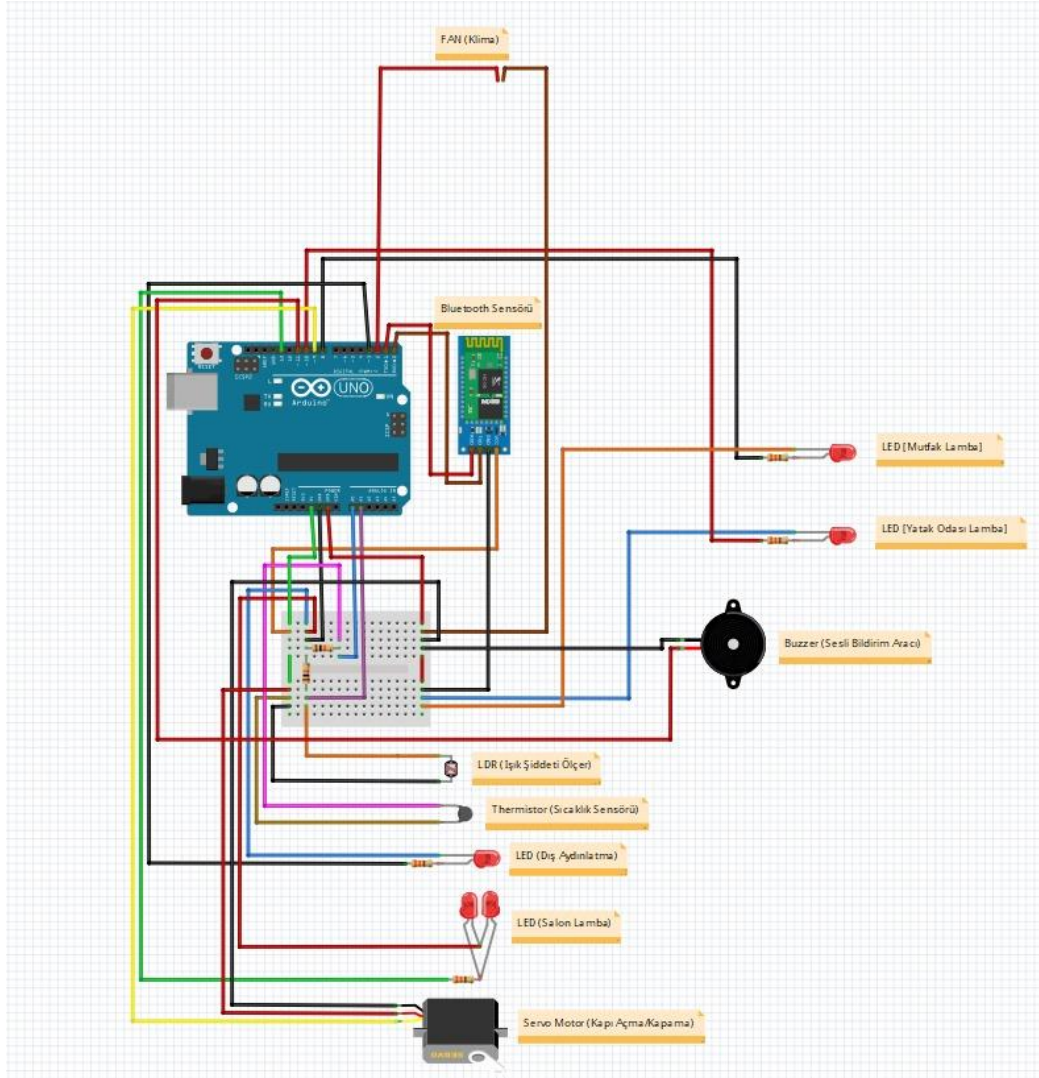


Fig. 1 Devre Tasarımı Örneği

Önceki raporda tasarladığımız devre üzerinde yapılan çalışmalar sonucunda devremize yeni elemanlar ekleyerek akıllı ev sistemimizin işlevlerini artırma üzerinde çalışmalar yaptık. Devreyi ilk tasarladığımızda her işlevden bir tane olarak tasarlamıştık. Ancak bizim akıllı ev sistemimizde her odada ayrı bir ışığı ve servo motorlar kapı/pencereleri kontrol edeceğimiz için önceki devreye ledler ve servo motorlar ekledik. Bu eklemeler sonucunda devre tasarımı yukarıdaki gibi oldu.

Bu devre tasarımına farklı işlevler üzerinde çalışma yaparak eklemeler yapmak istiyorum. Projeimin daha da kapsamlı olması için bu konu üzerinde çalışma gerçekleştireceğim.

IOT Uygulaması Kodlaması

Bu tasarladığım devreyi test edebilmek için IOT uygulamamın da hazır hale gelmesi gerekiyordu. Bunun üzerinde çalışmalar yaptım. Öncelikle yaptığım çalışmalara uygulama ve devrenin haberleşebilmesi için Bluetooth kısmından başladım. Önceki raporda paylaştığım uygulama arayüzünde Bluetooth modülüne bağlanmak için ‘Bağlan’ butonu koydum. Aynı zamanda bir de ‘Bağlantıyı kes’ butonu koydum. Bu butonların işlevlerini gerçekleştirebilmeleri için Android Stüdio’da kodlama yapmam gerekiyordu. Araştırmalar sonucu gerekli kodlamaları yaptım ve bunların sonucunda Bluetooth butonlarım aktif hale geldi. Bu kodlama kısmını aşağıda sizinle paylaştım.

```
// Set onClickListener for the connect button
buttonConnect.setOnClickListener(v -> connectToBluetoothDevice());
buttonDisConnect.setOnClickListener(v -> disconnectBluetooth());

toggleButtonSalon = findViewById(R.id.toggleButtonSalon);
muftakToggleButton = findViewById(R.id.muftakToggleButton);
yatakToggleButton = findViewById(R.id.yatakToggleButton);
kapiToggleButton = findViewById(R.id.kapiToggleButton);
pencereToggleButton = findViewById(R.id.pencereToggleButton);
klimaToggleButton = findViewById(R.id.klimaToggleButton);
```

Fig. 2 Android Bluetooth Kodlama

Yapmış olduğumuz kodlama sonucunda butonların aktif hale geldiğini test edebilmek için tasarladığımız uygulamayı Android telefonumuza kurup denemesini bu şekilde gerçekleştirdik. Bu işlemin olumlu sonuçlandığını gördükten sonra uygulamamızda bulunan diğer butonları da aktif hale getirmek için çalışmalar yaptım. Burada kullandığımız ToggleButton’lar aktif çalışabilmesi için Arduino kodunda her işleve tanımlanan değerleri Android Stüdio’da kodlama kısmında butonlara tek tek tanımlıyorum. Bu kodlamaları aşağıdaki şekilde detaylı inceleyebilirsiniz.

```

toggleButtonSalon.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            sendData("3"); // ToggleButton açıkken '1' gönder
        } else {
            sendData("4"); // ToggleButton kapalıyken '0' gönder
        }
    }
});

muftakToggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            sendData("5"); // ToggleButton açıkken '5' gönder
        } else {
            sendData("6"); // ToggleButton kapalıyken '6' gönder
        }
    }
});

yatakToggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            sendData("7"); // ToggleButton açıkken '7' gönder
        } else {
            sendData("8"); // ToggleButton kapalıyken '8' gönder
        }
    }
});

kapiToggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            sendData("1"); // ToggleButton açıkken '1' gönder
        } else {
            sendData("0"); // ToggleButton kapalıyken '0' gönder
        }
    }
});

```

Fig. 3 Android Buton Kodlama

Bu kodlamaları yaptıktan sonra ToggleButtonları denemek için Android uygulamayı tekrar yeni hali ile telefona kuruyorum. Ardından butonları tek tek denediğimde servo motorlar dışında hepsi sorunsuz çalıştığını gözlemliyorum. Android Stüdio kod kısmında herhangi bir sorun yok ancak Arduino kod kısmında servolar ile ilgili bir sorun olabilir bunun için çalışmalar yapacağım.

Android Uygulama Ara Yüzü

Android uygulamasının ara yüzü hakkında bilgi vermek gerekirse ilk başta tasarladığım ara yüz devre üzerinde deneme yapabilmek için şu anda bu şekilde ilerliyorum. Aşağıda Android telefonda uygulamanın görünüşünü sizinle paylaştım. Aşağıda da görüldüğü gibi bu ekrana bir logo eklenecek. Ardından her bir işlevin resimleri eklenecek. Uygulamanın daha güzel olması için Togglebuttonları da switchlerle değiştirmeyi düşünüyorum. Bunların dışında bazı butonlar ekrana sığmadı ve aşağıya doğru taşıyor. Bunun için ekranın pixel ayarlarından boyutunu ayarlayacağım. Ufak tefek eklemeler veya çıkarmalar olabilir. Ancak aşağıda görüldüğü şekilde uygulama sorunsuz bir şekilde devre ile haberleşerek çalışabiliyor.

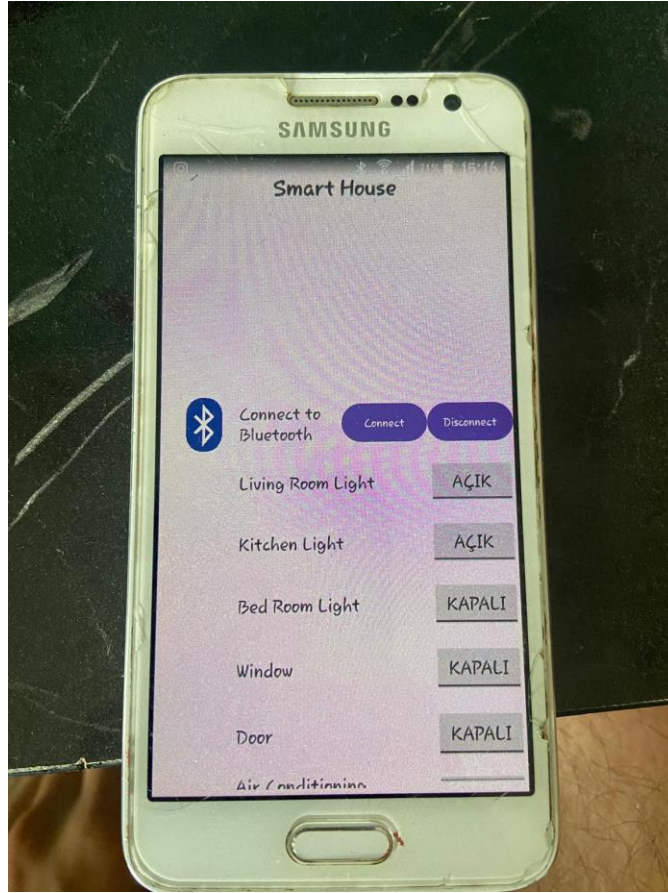


Fig. 4 Android Uygulama Ara Yüzü

4. Gelecek Hafta :

- Devreye farklı işlevler eklemek için çalışmalar yapılacak
- IOT uygulamasının daha da geliştirilmesi

5. The Resources Used in This Report:

[1] www.robotistan.com

[2] <https://gelecegiyazanlar.turkcell.com.tr>

[3] www.projehocam.com

[4] www.wikipedia.com