

**TÜBİTAK**  
**45. ORTAÖĞRETİM ÖĞRENCİLERİ**  
**ARAŞTIRMA PROJELERİ YARIŞMASI**  
**-2014-**

**PROJE RAPORU**

Proje Adı

**Android Söz Dinleyen Ev**  
**(AndSDev)**

Yarışma Kategorisi

**Bilgisayar**

**Ocak 2014**

## İçindekiler

Projenin Adı .....	3
Projenin Amacı.....	3
Proje Hedefleri .....	3
Giriş .....	3
Yöntem ve Teknikler .....	4
1. Farklı Söylem Tanımlama Algoritmalarının Oluşturulması .....	8
1.1 Genel İşlevler Tanımlama Algoritmasının Oluşturulması .....	9
1.2 İkili İşlev İşlemlerinin Tanımlama Algoritmasının Oluşturulması .....	11
1.3 Zamanlı İşlev İşlemlerinin Tanımlama Algoritmasının Oluşturulması .....	12
2. Model Evde Bulunan Elektronik Devreler .....	13
3. Proje Kapsamında Geliştirilen veya Kullanılan Sınıflar ve Fonksiyonlar .....	15
3.1 “MainActivity.java” Sınıfı .....	17
3.2 “Listener.java” Sınıfı .....	17
3.3 “cBluetooth.java” Sınıfı .....	17
3.4 “ZamanServisi.java” Sınıfı .....	18
3.5 “BluetoothDeneme.java” Sınıfı .....	18
3.6 “KomutIslemleri.java” Sınıfı .....	18
3.7 “SrStrIslem.java” Sınıfı .....	18
3.8 “SalonIslemleri.java” Sınıfı .....	18
3.9 “IkiliIslev.java” Sınıfı .....	19
3.10 “AnahtarListe.java” Sınıfı .....	19
Sonuçlar .....	19
Öneriler .....	20
Kaynaklar .....	21
Ekler .....	22
Ek1 : “MainActivity.java” Sınıfına ait bazı fonksiyonlar .....	22
Ek2 : “Listener.java” Sınıfına ait bazı fonksiyonlar .....	24
Ek3 : “KomutIslemleri.java” Sınıfına ait bazı fonksiyonlar .....	24
Ek4 : “SrStrIslem.java” Sınıfına ait bazı fonksiyonlar .....	26
Ek5 : “SalonIslemleri.java” Sınıfına ait bazı fonksiyonlar .....	28
Ek6 : “IkiliIslev.java” Sınıfına ait bazı fonksiyonlar .....	29
Ek7 : “AnahtarListe.java” Sınıfına ait sınıfına bazı fonksiyonlar .....	31
Ek8 : “Salon Lambasını Yak” temel komutu için örnek komut söylemleri .....	31

## Projenin Adı

Android Söz Dinleyen Ev (AndSDev)

## Projenin Amacı

Android işletim sisteminin ses tanıma ve bluetooth özelliğinden faydalanılarak bir evde kullanılan nesne veya cihazların(lambalar, perdeler, kombi, vb.) işlevlerine ait komutların konuşma diline yerleşmiş farklı söylem şekillerini tanıyabilen ve bu cihazların sözel komutlarla kontrolünü sağlayan bir program hazırlamak.

## Proje Hedefleri

Bu çalışmada amaca yönelik olarak aşağıdaki hedefler yapılan problem analizine göre belirlenmiş ve çalışmalar sırasıyla bu hedeflere göre yapılmıştır.

- Bluetooth modüller ile bir mikrodenetleyiciyi haberleştirme.
- Android işletim sisteminde Bluetooth haberleşmesinin yapılması.
- Android işletim sisteminde ses tanıma işleminin yapılması.
- Söylenmiş bir ses komutu ifadesinden işleve ait temel komut nasıl anlaşılır? Algoritmasını oluşturma ve yazma.
- Bir işleve ait farklı komut söylem şekillerinden aynı komut kodu nasıl üretilir? Algoritmasını oluşturma ve yazma.
- Komut söylenirken ortaya çıkan farklı sonuçlardan en doğru olanına nasıl ulaşılır? Algoritmasını oluşturma ve yazma.
- Sözel komutları model eve uygulama.

## Giriş

Android işletim sistemi 2009 yılından günümüze hızlı bir ivme kazanarak gelişmiştir ve kullanımı artmaktadır. Her gün dünya üzerinde bir milyondan fazla cihaz aktif hale getirilmektedir. Her ay Google Play Android uygulamaları yaklaşık 1,5 milyardan fazla indirmeye sahiptir[1][7]. Google'ın satın alması ve desteği ile birçok özellik eklenmiştir[2]. Bunlardan en önemlisi çoklu dil desteğine sahip ses tanıma özelliğidir. Ses tanıma, konuşmacı tarafından söylenen sözleri tanımlamak için işitsel sinyallerin analizinin yapılarak sonuçların değerlendirilmesidir. Sesli aramalarda, sms, sesli not alma gibi uygulamalarda sesi yazıya çevirerek rahatlıkla kullanılabilir [3][7].

Android işletim sisteminin ses tanıma özelliği incelendiğinde ses tanıma başarısının oldukça iyi olduğu görülmektedir[4]. Gürültüden etkilense de uygun kulaklık ve mikrofon kullanımı ile temiz bir ses kaydı elde edilebilmektedir. Özellikle ev gibi gürültülü sayılabilecek seslerden uzak ortamlarda verimli çalışmaktadır. Böylelikle her android mobil telefon kullanıcısı cebinde bir ses tanıma motoru taşımaktadır. Bu özelliğin akıllı ev sistemleri ile birleştirilmesiyle de filmlerde izlediğimiz bir hayal gerçekleşebilecektir.

Bu tür uygulamaları incelediğimizde iSpeech Home uygulaması ön plana çıkmaktadır. Ses tanıma teknolojileri üzerinde 2009'dan beri çalışan iSpeech firması birçok ev cihazı üreten firmayla görüşmeler yaparak birkaç ses komutuyla cihazların kontrolünü yapmayı amaçlamaktadır. Firma geliştirdiği mobil telefon platformlarına ait SDK'ları (Software Development Kit) ücretsiz olarak dağıtmaktadır[5].

Günümüzde mobil telefonlar üzerinde çalışan birçok uygulama kullanıcıya sunduğu grafik ara yüzlerini (GUI) ses ara yüzleri (VUI) ile desteklemektedir. Ses komutları kullanılarak giriş yapılan uygulamalarda kullanılan ara yüzler VUI (Voice User Interface) olarak adlandırılmaktadır [6]. Bu tür

uygulamalarda başarılı olabilmesi için kullanıcının komutlar konusunda eğitilmesi gerekmektedir. Aksi halde kullanıcı ne söyleyeceğini bilemeyebilir. Bunun için değişik yöntemler kullanılmaktadır. Google Play Store'daki birçok uygulama kullanıcıya komutları görsel olarak listelemekte veya kullanıcı uygulama tarafından yönlendirilmektedir [8].

Ses tanıma sistemlerinin gelişmesi ve akıllı evlerde kullanılan cihazların her eve girecek hale getirilmesiyle de bu teknolojiler günlük hayatımızda kullanılabilecek hale gelecektir. Fakat her bir cihaza ait komutların aynı dilde söylene bile son kullanıcı için bölgeye ve kişilere göre farklı söylem biçimleri ortaya çıkacaktır. Bir işleve ait yüzlerce komut kullanılabilecektir. Dolayısıyla her bir işleve ait komutun farklı söylem biçimlerinin tanınmasını sağlayan çalışmalara ihtiyaç vardır. Bu çalışmalarla son kullanıcı çeşitliliği ve komut zenginliğinin artması sağlanacaktır.

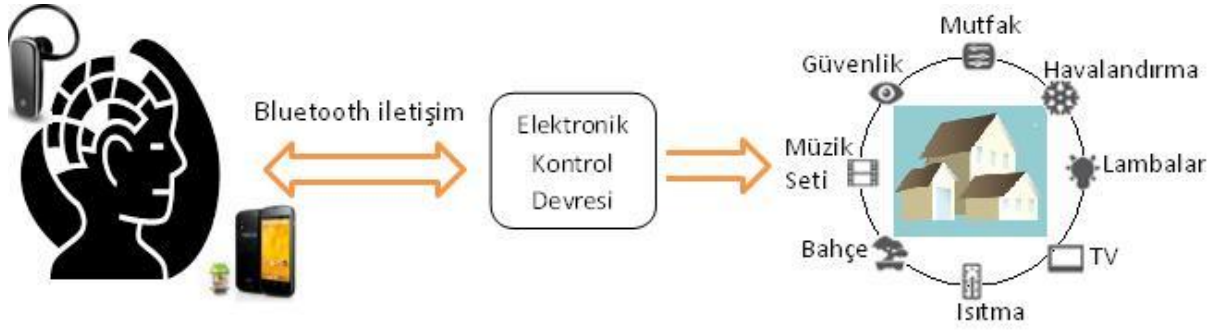
Biz de çalışmamıza bu noktadan yola çıkarak başladık. Okulumuzda önceki yıllarda yapılmış ve bilgisayar seri portu ile bir evdeki cihazların kontrolünü amaçlayan model ev projesi üzerinden bir evdeki cihazların kontrolü ile ilgili temel sözel komutları belirledik. Oluşturduğumuz bu komutların farklı söylem tarzları için okulumuz, ailemiz ve dershanemizdeki değişik yaş ve meslek gruplarındaki kişilerle beraber bir anket çalışması yaptık. Elde ettiğimiz verileri kullanarak bir cihaza/nesneye ait işlev komutunun farklı şekillerde söylene bile aynı işlevi gerçekleştirebilen bir programı Android işletim sisteminde çalışacak şekilde Eclipse ADT ortamında geliştirmeyi ve programın çıktılarını model ev üzerinde gözlemlemeyi amaçladık.

### **Yöntem ve Teknikler**

Genel olarak ses tanıma süreci mikrofon ile algılanan sesin kaydedilmesi ile başlar. Ses tanıma algoritmalarıyla ses işlenir, öznitelikleri çıkartılıp kaydedilir ve sonuçta karşılaştırma işlemi yapılarak sesin tanınması ile son bulur. Tanıma işleminin çıktısı sesin yazıya dönüştürülmüş şeklidir. Yazılım dilinde bu karakterlerden oluşmuş diziye string ifade adı verilir. Ses tanıma sonucunda elde edilen string ifadeden yararlanarak bir evdeki nesne veya cihazlara ait işlevlerin komutları yazılım tarafından tanınabilmektedir. Komutlara ait farklı söylem algoritmaları da bu ifade işlenerek elde edilecektir.

Sistemin genel çalışmasına baktığımızda mobil telefona direkt ya da bluetooth kulaklık vasıtasıyla gönderilen sesli komut ifadeleri yazılımda kullanılan ses tanıma API'leri ile Google ses tanıma motoruna internet üzerinden gönderilir. Belirlenen sayıdaki ses tanıma sonucu, string dizi olarak, tanımlama skoru da virgüllü sayı (float) dizisi olarak geri alınır [12][6]. Mobil telefonda hazırlanan yazılımla da alınan ses tanımlama sonucundan, belirlenen temel komut, farklı söylene bile yapılması istenen işleve ait gerekli komut kodu oluşturulur. Mobil telefon bluetooth modülü üzerinden elektronik devreye uygun formatta gönderilir. Sistemin genel çalışması Şekil 1'de gösterilmiştir.

Ses tanıma işlemi, "RecognitionListener" adı verilen ve ses tanıma işleminden oluşan bildirimleri alan genel ara yüzden [13] geliştirilen sınıf yardımıyla gerçekleştirilecektir. Bu sınıftaki "onResults" olayı ile ses tanıma sonuçları yakalanıp satırlara ayrılarak string değer şeklinde ana sınıfa gönderilecektir. Oluşturulan bu string ifadeden temel komutlara ait farklı söylemlerin tanınması java dilinin string işlemleri ile ilgili metotlarını kullanan yeni sınıf ve fonksiyonlar oluşturularak yapılacaktır.



Şekil

1: Sistemin genel çalışması

Komutlara Ait Kategoriler			LAMBA KOMUTLARI		
NO	ADI		NO	ADI	KOD
1	Lamba Komutları		1	Salon Lambasını Yak	001
2	Tv Müzik Seti			Salon Lambasını Söndür	002
3	Mutfak Komutları		2	Oturma Odası Lambasını Yak	005
4	Pencere ve Perdeler			Oturma Odası Lambasını Söndür	006
6	Güvenlik Komutları		3	Çocuk Odası Lambasını Yak	007
7	Bahçe Komutları			Çocuk Odası Lambasını Söndür	008
8	Isıtma/Isınma Komutları		4	Mutfak Lambasını Yak	009
9	Telefon Komutları			Mutfak Lambasını Söndür	016
			5	Yatak Odası Lambasını Yak	017
				Yatak Odası Lambasını Söndür	018
			6	Antre Lambasını Yak	023
				Antre Lambasını Söndür	024
TELEVİZYON MÜZİK SETİ			7	Tüm Lambaları Söndür	000
NO	ADI	KOD	MUTFAK KOMUTLARI		
1	Televizyonu Aç	030			
	Televizyonu Kapat	031	NO	ADI	KOD
2	Televizyon Program Arttır	032	1	Kombiyi Aç	068
	Televizyon Program Azalt	033		Kombiyi Kapat	069
3	TRT Bir	040	2	Kombi Sıcaklığını Arttır	060
4	TRT İki	041		Kombi Sıcaklığını Azalt	061
5	NTV	042	3	Çamaşır Makinasını Çalıştır	062
6	ATV	043		Çamaşır Makinasını Kapat	063
7	Kanal D	044	4	Ocağı Yak	064
8	Star TV	045		Ocağı Kapat	065
9	STV	046	5	Fırını Aç	066
10	Kanal 26	047		Fırını Kapat	067
11	Televizyon Sesini Arttır	034	PENCERE VE PERDELER		
	Televizyon Sesini Azalt	035			
12	Müzik Setini Aç	036	NO	ADI	KOD
	Müzik Setini Kapat	037	1	Perdeyi Aç	090
13	Müzik Setinin Sesini Arttır	038		Perdeyi Kapat	091
	Müzik Setinin Sesini Azalt	039			

Tablo 1: Model evdeki cihazların kontrolünde kullanılan temel komutlar ve kodları.

Tablo 1'deki temel komutlara ait farklı söylem şekillerini belirlemek amacıyla, kişilerin cihazlara/nesnelere ait işlevlerin yapılmasını bir başkasından ya da kendisinden nasıl istendiğini sorgulayan Tablo 2'de bir kısmı verilen anket hazırlanmıştır.

İş veya eylem	Aileniz size nasıl söyler.	Kardeşiniz veya çocuğunuza nasıl söylersiniz.
Salon Lambasını Yak		
Salon Lambasını Söndür		
Oturma Odası Lambasını Yak		
Oturma Odası Lambasını Söndür		
Tüm Lambaları Söndür		
.....		

**Tablo2:** Uygulanan ankete ait örnek.

Hazırlanan anket yaş ve meslek gruplarına göre yaklaşık 150 kişiye uygulandı. Bunlardan 75'i öğrenci (14-18 yaş), 25'i öğretmen (26-40 yaş), 20'si ev hanımı (30-50 yaş), 30'u diğer kişiler şeklinde dağılım oluşmuştur. Excel programında temel komutlara ait kodlara göre sayfalar oluşturulmuş, yapılan anket ayrı ayrı incelenerek her bir temel komutun farklı söylem biçimlerine ait komutlar tablo 3'te verilen örnekte olduğu gibi yeniden yazılarak anahtar kelimeler belirlenmiş ve anahtar kelime listeleri oluşturulmuştur.

	Temel Komut : "Salon Lambasını Yak"			
1	salonun lambasını yak	salon ceyranını aydınlat	salondaki şavkı aç	büyük odadaki ceyranı on
2	salondaki lambayı yak	salonun ceyranını aydınlat	salon şavkını yak	büyük odanın ceyranını aç
3	salon ışığını yak	salondaki ceyranı aydınlat	salonun şavkını yak	büyük odadaki ceyranı aç
4	salondaki ışığı yak	salon ceyranını on	salondaki şavkı yak	büyük odanın ceyranını yak
5	salonun ışığını yak	salonun ceyranını on	salon şavkını aydınlat	büyük odadaki ceyranı yak
6	salon lambasını aç	salondaki ceyranı on	salonun şavkını aydınlat	büyük odanın ceyranını aydınlat
7	salonun lambasını aç	salon ceyranını aç	salondaki şavkı aydınlat	büyük odadaki ceyranı aydınlat
8	salondaki lambayı aç	salonun ceyranını aç	salon şavkını on	büyük odanın ceyranını on
9	salon ışığını aç	salondaki ceyranı aç	salonun şavkını on	büyük odadaki ceyranı on
10	salonun ışığını aç	salon ceyranını yak	salondaki şavkı on	büyük odanın ceyranını çak

**Tablo3:** Farklı söylem biçimlerine ait bazı örnek komutlar.

Anahtar kelime listelerinin oluşturulmasını anlatmak için lamba komutları ile ilgili yapılan çalışmalardan bazı örnekler aşağıdaki gibidir;

**Örnek 1-** Salon lambasının yakılması ile ilgili bazı komutlar şunlardır;

- 1- "salon lambasını yak"
- 2- "salondaki lambayı yak"
- 3- "salon ışığını çak"
- 4- "salondaki elektriği aç"
- 5- "salon ceyranını aydınlat"
- 6- .....

Salon Lambasını Yak			
Yer	Cihaz	İşlev	Açıklama
salon	lamba	Yak	
büyük oda	Işık	Aç	
misafir odası	Işığ	düğmeye bas	Lamba yanık ise söndür değilse yak yada sönük ise yak
yemek odası	elektrik	aydınlat	
sabah	elektriğ	On	
	anahtar	Çak	
	ceyran		
	ceryan		
	ceryan		
	Şavk		
	kuruşka		

**Tablo 4 :** “Salon Lambasını Yak” temel komutu için hazırlanan anahtar liste tablosu.

**Örnek 2-** Salon lambasının söndürülmesi ile ilgili bazı komutlar şunlardır;

- 1- “salonun lambasını söndür”
- 2- “salondaki ışığı söndür”
- 3- “salon lambasını karart”
- 4- “büyük odadaki ışık of”
- 5- “misafir odası elektriğini kes”
- 6- .....

Salon Lambasını Söndür			
Yer	Cihaz	İşlev	Açıklama
salon	Lamba	söndür	
büyük oda	Işık	kapat	
misafir oda	Işığ	düğmeye bas	Lamba yanık ise söndür değilse yak yada sönük ise yak
sabah	elektrik	karart	
	elektriğ	Kapa	
	anahtarı	Off	
	ceyran	Of	
	ceryan	Kes	
	ceryan		
	Şavk		
	kuruşka		

**Tablo 5 :** “Salon Lambasını Söndür” temel komutu için hazırlanan anahtar liste tablosu.

Anket yolu ile elde edilen bu komutlardan altı çizili olan kelimeler komutun tanımlanabilmesi için anahtar kelimelerdir. Bu kelimelere ayrıca komut söylenirken ses tanıma sonucunda tanımlanan bazı yakın kelimelerde eklenmiştir. Örneğin “off” ve “of” veya “salon” ve “sabah” gibi.

Elde edilen bu anahtar listeden yer olarak evdeki salonu, cihaz olarak lambayı ve işlev olarak da yak/söndür ifade eden anahtar kelimeler “AnahtarListe.java” isimli sınıfta string sabiti olarak aşağıdaki gibi tanımlanmıştır. Böylelikle kodlama işlemi esnasında veri tabanı işlemleri ile uğraşılmamıştır. Fakat hazırlanan yapı veri tabanı işlemleri ile çalışmaya uygundur.

```
public final static String Salon="salon, büyük oda, yemek odası, misafir odası, sabah";  
public final static String Lamba="lamba, ışık, ışık, elektrik, elektrikli, anahtar, ceyran, ceryan, cereyan, şavk";  
public final static String LambaYak="yak, aç, düğmeye bas, aydınlat, on, çak";  
public final static String LambaSondur="söndür, sondur, kapat, düğmeye bas, karart, kapa, off, of, kes";
```

Bu tanımlamalar diğer sınıflar tarafından da kullanılacaktır.

Elde edilen verilerin bu şekilde incelenmesinden sonra bir eve ait gerçekleştirilmesi düşünülen işlemler/eylemler belirlenmiştir. İşlem sırası model evde uygulanabilirliğe göre tercih edilmiştir.

Odalara ait belirlenen ortak işlemler;

- Lamba işlemleri
  - Lambanın yakılması.
  - Lambanın söndürülmesi.
  - Bir odanın lambasının yakılıp diğerinin söndürülmesi.
  - Bir odanın lambasının söndürülüp diğerinin yakılması.
  - Lambanın zamanla yakılması.
  - Lambanın zamanla söndürülmesi.
  - Lambaların hepsinin söndürülmesi.
- Perde/pencere işlemleri
  - Perdenin/pencerenin açılması.
  - Perdenin/pencerenin kapatılması.
- Cihazlara ait işlemler
  - Cihazların açılıp kapatılması.
  - Cihaz türüne göre oluşturulan temel komutlarda belirlenen işlemler şeklinde oluşturulmuştur.

### **1. Farklı Söylem Tanımlama Algoritmalarının Oluşturulması**

Anket sonuçlarından elde ettiğimiz ve oluşturabildiğimiz komutlara göre yukarıda belirlediğimiz işlemler incelendiğinde üç ana işlem kategorisinin olduğu görülmektedir.

1-“<yer> + <nesne/cihaz> + <işlev>” komut söylemi yapısındaki genel işlemler.

2-“<yer1> + <nesne1/cihaz1> + <işlev1> + <yer2> + <nesne2/cihaz2> + <işlev2>” komut söylemi yapısındaki ikili işlev işlemleri.

3-“<yer> + <nesne/cihaz> + <zaman> + <işlev>” komut söylemi yapısındaki zamanlı işlev işlemleri.

Burada öncelikli olarak işlem yapılacak olan kategorinin hangi yere/cihaza uygulanacağını belirlenmesi gerekir. Bunun için hazırlanan anahtar kelime listeleri içerisinden yer/cihaz bilgisinin ses tanıma sonucundan gelen ifadede bulunup bulunmadığı araştırılır. Eğer varsa bulunan yere/cihaza göre uygun işlem kategorileri sorgulanır. Tanımlanmış yer/cihaz yoksa hata kodu oluşturulur.

Buna göre üç farklı kategori için temel olarak aynı fakat ayrı işlevlerin yapılmasını sağlayan algoritmaların hazırlanması gerekir.



### 1.1 Genel İşlemler Tanımlama Algoritmasının Oluşturulması

Bir genel işlem komut söylemi ifadesi, “<yer>+<nesne/cihaz>+<işlev>” öğelerinden oluşmaktadır. Örneğin;

**“büyük odanın ceryanını aç”** komut ifadesinde,

Yer : büyük oda => salon

Nesne : ceryan => lamba

İşlev : aç => yak şeklinde olduğu görülmektedir.

Bir diğer örnek olarak da;

**“mutfağın ışığını kapat”** komut ifadesinde,

Yer : mutfak => mutfak

Cihaz : ışık => lamba

İşlev : kapat => söndür şeklindedir.

Bir diğer örnek olarak da;

**“mutfaktaki televizyonu kapa”** komut ifadesinde,

Yer : mutfak => mutfak

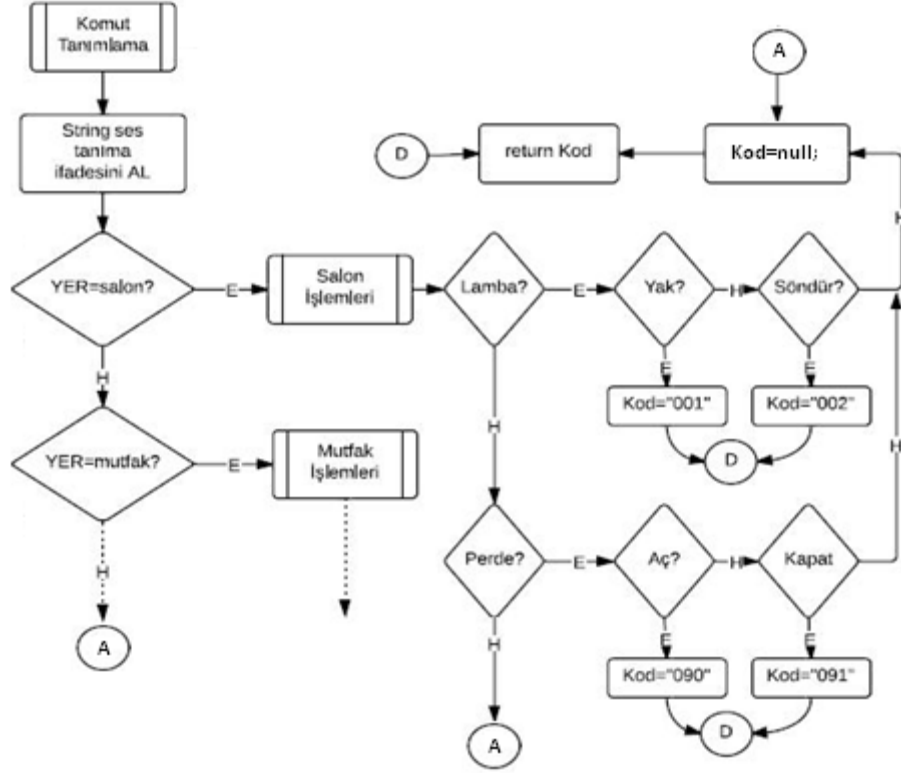
Cihaz : televizyon => televizyon

İşlev : kapa => kapat şeklindedir.

Verilen örneklerdeki gibi oluşturulan veya anket ile elde edilen birçok komut incelendiğinde genel işlemlere ait tanımlama algoritması için aşağıdaki gibi bir yol izlenmesi düşünülmüştür.

1. İşleme göre, yer ve cihazlara ait anahtar kelimeler ifade de var mı?
  - Eğer yoksa boş kod döndür.
  - Eğer varsa, yere ve cihaza ait işlemlere git.
2. Cihazın/nesnenin işlevini tanımlayan anahtar kelimeler ifade de var mı?
  - Eğer yoksa boş kod döndür.
  - Eğer varsa işleme uygun komut kodunu üret ve kodu döndür.

Buna göre salon işlemleri ile ilgili örnek akış diyagramı Şekil 2’deki gibi oluşturulmuştur.



Şekil 2: Genel işlemler tanımlama algoritmasına ait akış diyagramı.

Buradan yola çıkarak işlemi gerçekleştirmek için basitleştirilmiş örnek fonksiyonlar aşağıdaki gibidir;

```

String KomutIslem (String sonuc) {

    if (listeVarMI(sonuc,AnahtarListe.SALON) return SalonIslemleri (sonuc);
    else
    if (listeVarMI(sonuc,AnahtarListe.MUTFAK) return MutfakIslemleri(sonuc);
    ...
    else return null;//İşlem yok.
}

String SalonIslemleri(String sonuc){
    String kod=null;
    if(listeVarMI (sonuc,AnahtarListe.LAMBA)){
        if(listeVarMI (sonuc,AnahtarListe.YAK)) kod="001";
        if(listeVarMI (sonuc,AnahtarListe.SONDUR)) kod="002";
    }else
    if(listeVarMI (sonuc,AnahtarListe.PERDE)){
        if(listeVarMI(sonuc,AnahtarListe.AC)) kod="090";
        if(listeVarMI(sonuc,AnahtarListe.KAPAT)) kod="091";
    }
    ...
    return kod;//Hiçbiri yoksa işlem yok kodu döndürür.
}

boolean listeVarMI(String sonuc, String liste){
    String[] ayir=liste.split(",");
    for (int i = 0; i < ayir.length; i++) {
        if(sonuc.contains(ayir[i])) return true;
    }
    return false;
}

```

Ortak olarak kullanılan “listeVarMI” fonksiyonu ses tanımlamadan gelen string içerisinde elde ettiğimiz verilerden oluşturduğumuz anahtar kelimeleri arar ve bulursa “true”, bulamazsa “false” değerini geri döndürür. Fonksiyon isimleri anlatım için kullanılmıştır. Yazılımdaki isimlere benzer, fakat farklıdır.

### 1.2 İkili İşlev İşlemlerinin Tanımlama Algoritmasının Oluşturulması

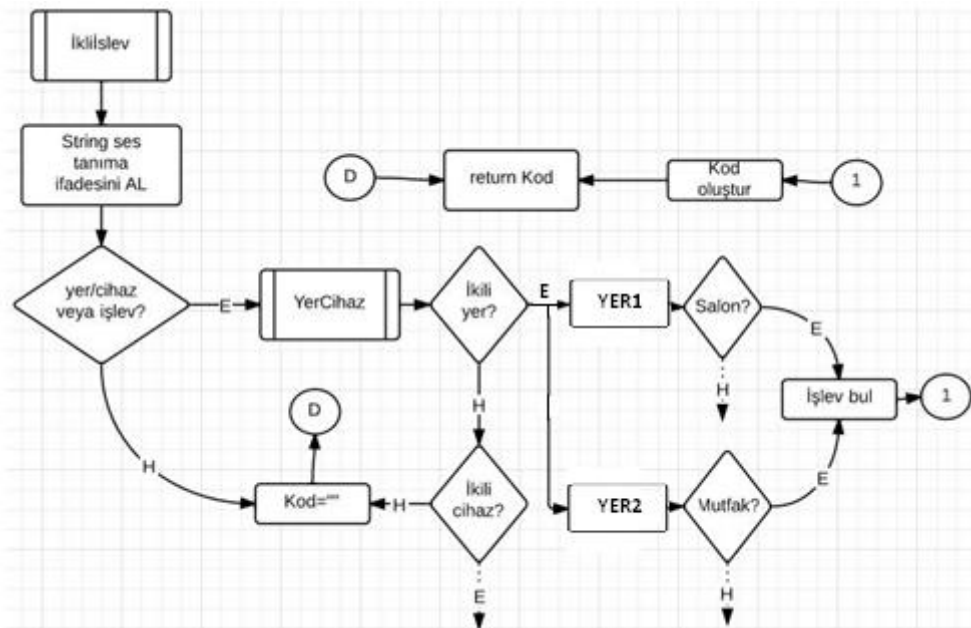
Bir ikili işlev işlemi komut söylemi ifadesi “<yer1> + <nesne1/cihaz1> + <işlev1> + <yer2> + <nesne2/cihaz2>+<işlev2>” şeklindedir. Bir cihazın/nesnenin açılıp/yakılıp bir diğerinin kapatılması /söndürülmesi şeklindeki komutları kapsar. Bu komutlara örnek verilecek olursa;

- “salon lambasını yak mutfak lambasını söndür”
- “mutfak ışığını kapat antrenin ışığını aç”
- “büyük odanın elektrikliğini aç çocuk odasının ışığını kes”
- “müzik setini kapa televizyonu aç” şeklinde olduğu görülür.

Anahtar olarak kullanılacak kelimelerin altı çizilmiş ve renklendirilmiştir. Komutlara bakıldığında ikili ifadeler halinde yer veya cihaz içermektedirler. Buna göre algoritma şu şekilde hazırlanmıştır.

1. İkili yer/cihaz veya işlem var mı?
  - Eğer yoksa boş kod döndür.
  - Eğer varsa ikili yer veya cihaz işlemlerine git.
    - İfade ilk söylenen yeri/cihazı bul.
    - İlk söylenen yer/cihaza ait işlev bul.
    - İlk söylenen yer/cihaza ait işlevin kodunu al.
    - İkinci söylenen yer/cihaza ait işlev bul.
    - İkinci söylenen yer/cihaza ait işlevin kodunu al.
    - İkili işlev kodunu oluştur ve geri döndür.

Buna göre akış diyagramı şekil 3’teki gibi oluşmuştur.



Şekil 3: İkili işlev işlemlerinin tanımlama algoritmasına ait akış diyagramı.

Akış diyagramına göre hazırlanan çözümle ilgili kodlar raporun ekler kısmında Ek6'da "IkiliIslev.java" başlıklı sayfada yer almaktadır.

### 1.3 Zamanlı İşlev İşlemlerinin Tanımlama Algoritmasının Oluşturulması

Bir zamanlı işlev komut söylemi ifadesi "<yer> + <nesne/cihaz> + <zaman> + <işlev>" şeklindedir. Bir cihazın/nesnenin belirli bir süre içerisinde açılması/yakılması veya kapatılması/ söndürülmesi şeklindeki komutları kapsar. Bu komutlara örnek verilecek olursa;

- "salon lambasını 10 dakika sonra yak"
- "televizyonu 1 saat sonra kapat"
- "mutfak lambasını aç 3 saat sonra karart"
- "bahçe sulamasını 20 dakika sonra bitir" şeklinde olur.

Komutlar incelendiğinde ise tek bir yer veya cihaza ait işlevin zamanlama ve süre bilgisini içerdiği görülmektedir. Öncelikli olarak süre bilgisinin sayısal olarak elde edilmesi gereklidir. Ayrıca istenen işlev hemen değil de daha sonra gerçekleştirileceğinden dolayı belirlenen işlemin uygun formatta bir depo alanına kaydının yapılması gerekmektedir. Bu tür bir kayıt için veri tabanı gibi karmaşık işlemlerin yerine küçük yapıdaki verileri kaydedebileceğimiz "SharedPreference" alanı uygundur[15]. Bu alana zamanlama işlevi ile ilgili uygun formatta kaydedilen komutlar daha sonra yazılımın arka planında çalışan bir servis[16] yardımıyla belli sürelerle okunarak vakti gelen işlevin yapılması sağlanmalıdır.

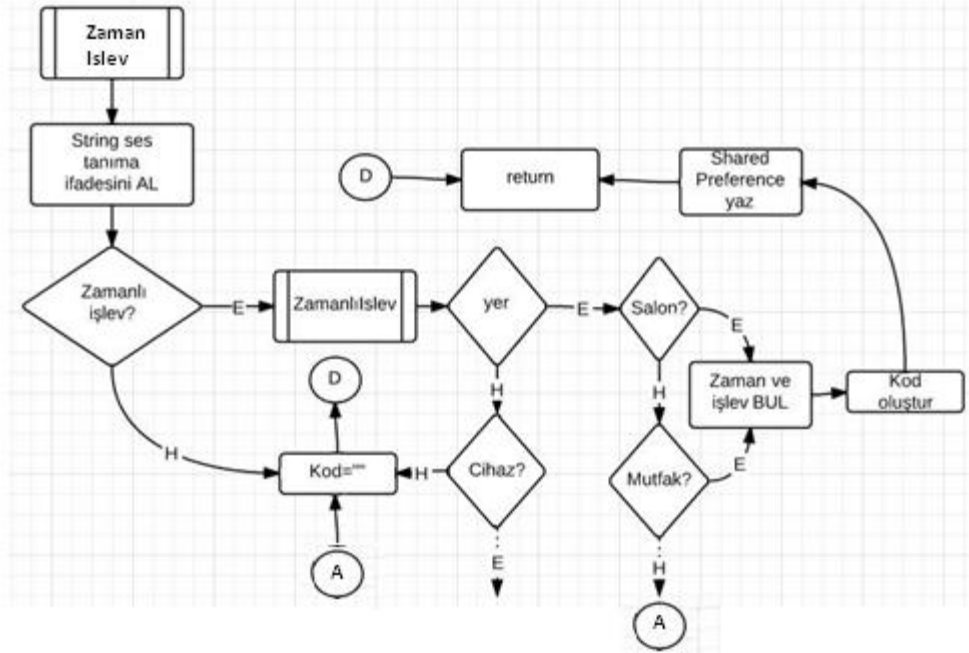
Buna göre algoritma şu şekilde olmaktadır;

1. Zamanlı işlev var mı?
  - Eğer yoksa boş kod döndür.
  - Eğer varsa zamanlı işleve git.
    - Sonuç içerisinde yer bul.
    - Sonuç içerisinde zaman bul.
    - Sonuç içerisinde süre bul.
    - Uygun kodu oluştur.
    - "SharedPreference" alanına yaz.

Akış diyagramı şekil 4'teki gibidir.

Depo alanına kaydedilen işlemlerle ilgili bilgilerin belli bir periyotta okunarak zamanı gelmiş işlemlerin kontrol edilmesi gerekmektedir. Bununla ilgili hazırlanan servis algoritması şu şekildedir;

1. Servisi oluştur.
2. Zamanlayıcıyı oluştur ve ayarla.
3. "SharedPreference" alanını zamanlayıcı değerinde oku (Örneğin, her saniyede).
4. Zamanlanmış işlev var mı?
  - Eğer yoksa işlem yapma.
  - Eğer varsa ve süresi geldiyse işlem için kod üret.
    - Kodu ana sınıfa gönder.
    - İşlevi "SharedPreference" alanından sil.
5. Zamanlı işlev için taramaya devam et.

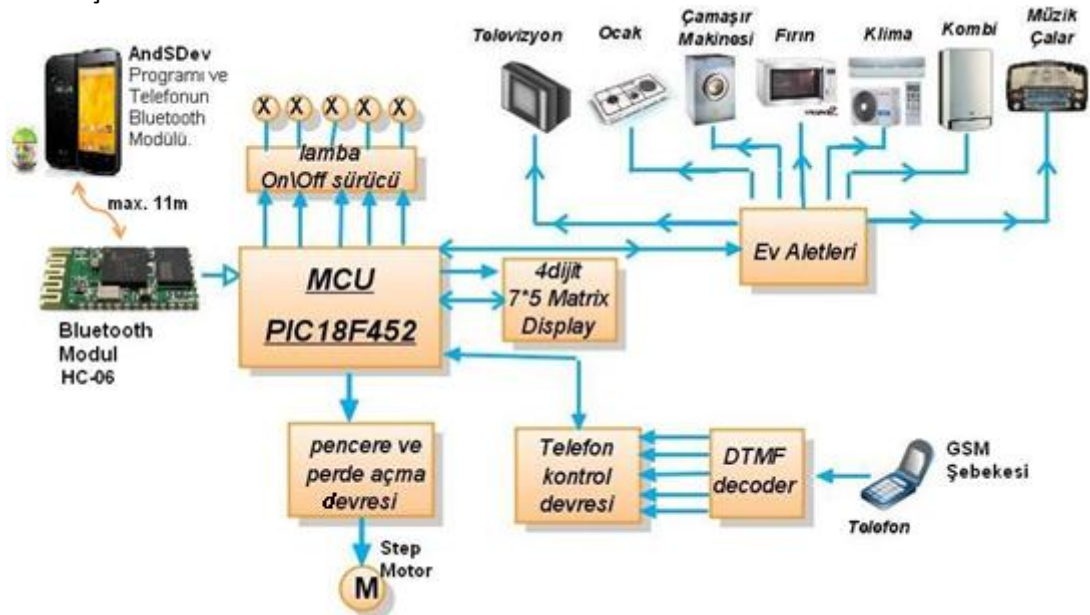


Şekil 4: Zamanlı işlev işlemlerinin tanımlanmasına ait akış diyagramı.

## 2. Model Evde Bulunan Elektronik Devreler

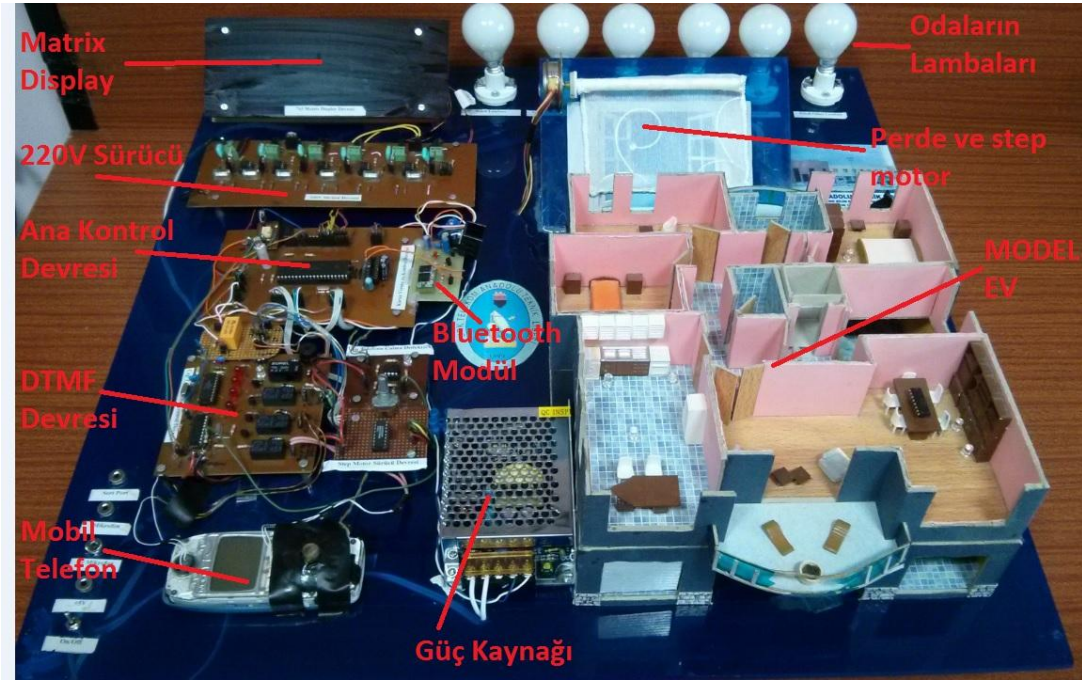
Model evde bulunan cihazların kontrolünü sağlamak için MCU(Mikrocontroller Unit) tabanlı bir elektronik devre vardır. Devrenin merkezinde PIC18F452 [9] mikro denetleyicisi bulunmaktadır. Devrenin blok şeması Şekil 5’de görülmektedir. Devre aşağıdaki bölümlerden oluşur.

- Ana kontrol devresi.
- 220V lamba sürücü devresi.
- Bluetooth modül devresi.
- Dört dijit (7x5) matris display devresi.
- Step motor sürücü devresi.
- DTMF kod çözücü ve telefon kontrol devresi.



Şekil 5: Model evdeki elektronik devrelere ait blok şema.

Kullanılan devre ile 6 adet odanın lambası, perdesi, mutfakta kullanılan cihazlar, TV, müzik seti kontrolü yapılabilir. Devrede bulunan cep telefonu dışarıdan aranarak da girilen uygun kodlarla sistem çalıştırılabilir. Ana devreye bluetooth modül [11] devresi PCB'si hazırlanarak eklenmiştir.



**Resim 1:** Projede kullanılan elektronik devreler ve model ev.

Devrenin kontrol mantığı PIC18F452 içerisine yüklenen yazılımla oluşturulmuştur. Yazılım CCS C derleyicisinde yazılmıştır[10]. Genel olarak MCU seri port pinlerine (Rx, Tx) bağlanan bluetooth modülden veri bekler, veri geldiğinde gelen komut kodunu çözer ve ilgili işlemi yapar. İletişimde string olarak seri veri alınır ve gönderilir. Seri veri iletişim hızı 9600 baud'dur. Veri formatı **"\*KOD#"** şeklindedir. Komut kodu üç tane rakamdan oluşur. Oluşturulan temel komutlara ait komut kodları Tablo 1'de verilmiştir.

MCU yazılımının algoritması;

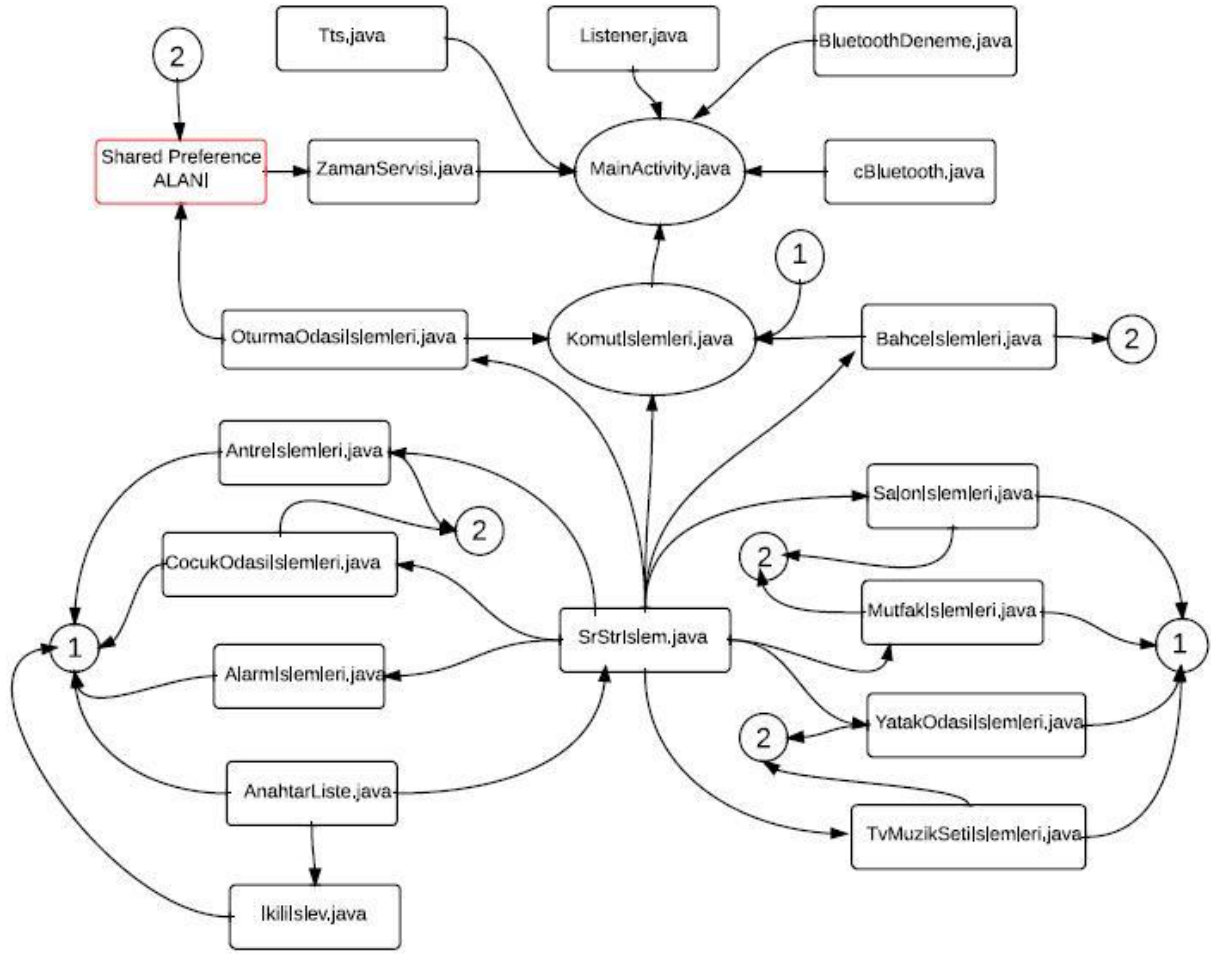
1. MCU'nun ilk kurulum ayarlarını yap.
2. Seri port ve harici kesmeyi kur ve aktifleştir.
3. Seri port ve DTMF devreden veri bekle.
4. Veri geldi ise ;
  - string dizeden kodu ayırıştır.
  - String kodu sayısal değere çevir.
  - Koda uygun işleme ait fonksiyonu çağır.
5. Seri porttan DTMF devreden tekrar veri bekle.

Model evde cihazlara ait gösterilemeyen bazı işlevler matriks led display üzerinde gösterilmeye çalışılmıştır.



### 3. Proje Kapsamında Geliştirilen veya Kullanılan Sınıflar ve Fonksiyonlar

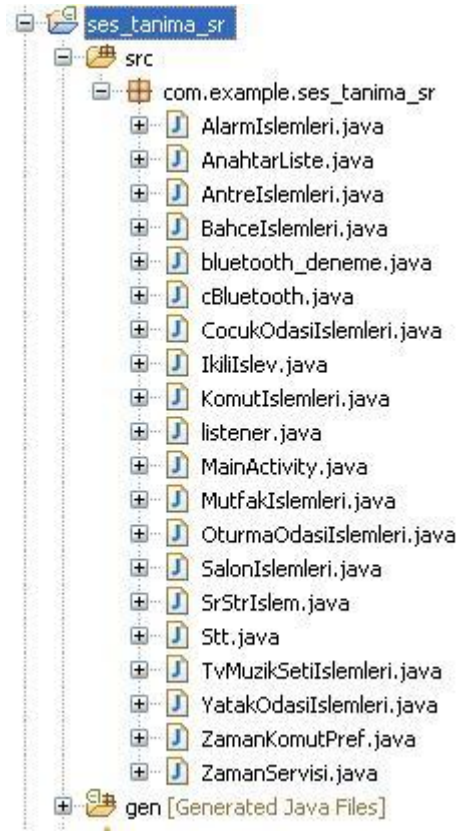
Bir evde bulunan birçok nesne ve cihaz model evde de bulunmaktadır. Bu cihazların tamamının kontrol edilmesi düşünüldüğünde, kapsamı oldukça büyük bir proje karşımıza çıkacaktır. Bu nedenle öncelik olarak amaçta belirtilen farklı söylemleri tanıma ile ilgili algoritmaların koda yansıtılması düşünülmüştür. Özellikle de lambda komutları üzerinde durulmuş ve burada farklı söylem tanıma algoritmalarına uygun olarak geliştirilen kod yapıları diğer komut kategorilerine uygulanmaya çalışılmıştır. Fakat hazırlanan sınıf diyagramına bir evdeki neredeyse bütün kontrolleri yapabilecek yapılar da eklenmiştir. Programa ait çizilen sınıf diyagramı Şekil 6'da verilmiştir.



Şekil

6: Programa ait çizilen sınıf diyagramı.

Çizilen sınıf diyagramında ok yönleri veri gönderimini ifade etmektedir. Yani bir sınıf diğer bir sınıf tarafından oluşturulup kullanılıyorsa, kullanılan sınıf oluşturulduğu sınıfa yaptığı işlemle ilgili veri göndermektedir. Yer ve cihazlara ait işlemleri gerçekleştiren bütün sınıflar "KomutIslemleri.java" sınıfı tarafından oluşturulmaktadır. "KomutIslemleri.java" sınıfı ana sınıfla bu sınıflar arasında köprü gibidir. İşlemin yapılacağı yer ve cihaz bu sınıfta belirlenir ve işlemin sonucu ana sınıfa gönderilir. Zamanlı işlev gerektiren bütün sınıflar da depo alanına veri kaydetmektedirler. Depo alanından veri okuma işlemini ise "ZamanServisi.java" sınıfı yapmaktadır.



**Resim 2:** Eclipse Package Explorer görüntüsü.

“MainActivity.java” isimli sınıf ana sınıftır. Sistem bu noktadan itibaren mobil telefonda çalışmaya başlar. Activity sınıfından geliştirilmiştir [7][18]. Programın görsel öğeleri burada bulunmaktadır. “Listener.java” ise ses tanıma işleminin gerçekleştirildiği sınıftır. “RecognitionListener” sınıfından uygulanmıştır [13]. “onResults” olayıyla ses tanıma sonuçları alınır ve “MainActivity.java” sınıfına handler nesnesi [19][6] yardımıyla satırlara ayrılarak string mesaj olarak gönderilir. Gelen mesaj artık ses tanıma sonucumuzdur. Farklı söylem tanımlama algoritmalarıyla geliştirdiğimiz diğer sınıflarca string ifade işlenir ve ilgili işlemlere ait işlev kodları üretilerek tekrar ana sınıf üzerinden bluetooth modüle gönderilir.

“cBluetooth.java” bluetooth işlemlerinin yapıldığı sınıftır. Bluetooth modüle bağlanır, veri alır ve veri gönderme işlemlerini yapar. CxemCar1 [17] çalışmasını anlatan internet sitesinden alınmış ve elimizdeki bluetooth modülle [11] haberleşme “BluetoothDeneme.java” sınıfıyla denenmiştir. Deneme sonucunda mikro denetleyiciden gelen ifadelerin tam alınabilmesi için bazı değişiklikler yapılmıştır. Bütün haberleşme string değerlerle yapılmaktadır.

“KomutIslemleri.java” ise evdeki kontrol işlemleri için hazırlanmış sınıflarla ana sınıf arasında köprü işlevi görmektedir. Elde edilen ses tanıma sonucundan yer veya cihaza göre yapılacak işlem burada belirlenmekte ve ilgili sınıfa ait fonksiyon çağrılarak işlem yapıp, gerekli işlem kodu ana sınıfa gönderilmektedir.

Raporun buradan sonraki kısmında Ocak 2014 tarihine göre çalışmamızda kullandığımız veya yazdığımız bazı sınıflar-fonksiyonlar ve bunlara ait işlevler anlatılmaktadır.



### 3.1 “MainActivity.java” Sınıfı

Programa ait ana sınıftır. Android Activity sınıfına ait metotları ve aşağıdaki fonksiyonları kullanır (Bkz. Ek1).

**public void sr\_baslat():** Intent sınıfı [\[20\]](#)[\[6\]](#)[\[7\]](#) yardımıyla ses tanıma işlemini başlatılmasını sağlar. Program ilk çalıştığında buton yardımıyla bu fonksiyon çağırılarak ses tanıma başlatılır. Daha sonra her hata oluştuğunda veya yeni ses tanıma istendiğinde tekrar başlar. İnternet bağlantı hatası ve kullanıcı istediğinde ise ses tanıma işlemi durdurulur.

**public void zmnServisiDurdur():** “ZamanServisi.java” adlı sınıfın intent sınıfı yardımıyla durdurulmasını sağlar. Program çalıştırıldığında eğer servis çalışmıyorsa servis çalıştırılır. Bu fonksiyon yardımıyla da durdurulur.

**private final handler srHandler=new Handler():** Sınıflardan gönderilen mesajların yakalanmasında kullanılır. Gelen mesajlara göre programın çalışması yönlendirilir. Ses tanıma, bluetooth ve zaman servisi ile ilgili mesajlar yakalanır.

**private void komut Gönder (String kmt\_kodu):** Üretilen komut kodlarının bluetooth modül üzerinden string olarak donanımın alabileceği biçimde gönderilmesini sağlar. Komut donanım tarafından alınıp işlenebilmişse geriye “OK” string ifadesi gönderilir. Komut gönderim formatı seri olarak “\*KOD#” şeklindedir.

### 3.2 “Listener.java” Sınıfı

Android “RecognitionListener” sınıfından uygulanmış bir sınıftır [\[13\]](#). Bu sınıfın metotlarını kullanır. Ana sınıfa hata mesajları ile ses tanıma sonucunu string biçimde handler nesnesi kullanarak mesaj olarak gönderir. “onResults” olayı ile yakalanan ses tanıma sonucu “/n” karakteri ile satırlara ayrılır. Bu sınıf ana sınıfta tanımlanan SpeechRecognition [\[12\]](#) sınıfına parametre olarak bildirilir (Bkz Ek2).

**private string hata\_kodu\_gönder(int error):** “onError” olayıyla üretilen hata koduna göre string ifade oluşturur ve ana sınıfa handler nesnesi ile mesaj olarak gönderir.

### 3.3 “cBluetooth.java” Sınıfı

Mobil telefonun bluetooth modülüyle ilişkili işlemleri yapmak için kullanılmıştır [\[17\]](#) .

**public void checkBTState():** Bluetooth durumunu kontrol eder. Bluetooth cihazda yoksa “Bluetooth mevcut değil” mesajını ana sınıfa gönderir. Bluetooth açık değilse açma isteğini göndererek açılmasını sağlar.

**public void BT\_Connect (String Address):** Adresi girilen bluetooth modüle bağlantı yapılmasını sağlar.

**public void sendData (String message):** String ifadenin gönderilmesini sağlar.

**private class Connected Thrend extends Thread:** Bu sınıf bir iş parçacığı oluşturarak bluetooth bağlı iken elektronik devrede bulunan bluetooth modülden gelen bilgileri yakalar ve ana sınıfa string olarak gönderir.

### 3.4 “ZamanServisi.java” Sınıfı

Android Service sınıfından geliştirilmiştir. Servis yordamı olarak çalışır ve istenen sürede periyodik olarak Shared Preference[15] alanını okuyarak zamanlanmış işlem var mı kontrol eder. Eğer buldu ise ve işlem zamanı gelmişse işlem kodunu ana sınıfa gönderir. Zamanlama işlemi Timer nesnesi ile yapılır.

**private void IslemYap():** Shared Preference alanını okur ve işlem var mı kontrol eder. İşlem var ise işlem kodunu oluşturur ve ana sınıfa gönderir.

### 3.5 “BluetoothDeneme.java” Sınıfı

“cBluetooth.java” sınıfının çalışmasını denemek için oluşturulmuştur. Android Activity sınıfından geliştirilmiştir. Butonlar yardımıyla string ifadeleri donanıma gönderir.

### 3.6 “KomutIslemleri.java” Sınıfı

Ana sınıfa gelen ses tanıma sonucundan istenen işlemin ait olduğu sınıfın bulunmasını ve bu sınıfı oluşturarak ilgili işlemlerin yapılmasını sağlar (Bkz Ek3).

**public string islem(string sonuc):** İşlemin yapılacağı sınıflara dallanmayı sağlayan fonksiyondur. Gelen ses tanıma sonucundan istenen işlemi bulur ve bu işleme ait sınıfın metodunu çağırır.

### 3.7 “SrStrIslem.java” Sınıfı

String ifade işlemlerini içeren sınıftır. Genel olarak projede kullanılacak bazı ortak fonksiyonlar bu sınıfta tanımlanmıştır. Dolayısıyla neredeyse bütün kontrol sınıfları tarafından oluşturulmuş ve bu sınıfın metotları kullanılmıştır (Bkz Ek4).

**public boolean Uygula(String sonuc, int islem):** Tam sayı sabit olarak belirlenen işlem kodlarına göre “sonuc” ifadesinden yapılacak olan işlemin (yak, söndür, zamanlı yak vb.) belirlenmesini sağlar.

**public boolean ListeVarMı (String sonuc, String Liste):** “sonuc” ifadesinde anahtar listede belirlenen kelimelerin olup olmadığına bakar. Varsa mantıksal “true”, yoksa “false” değerini döndürür.

**public String StrSayiBul (string s):** Girilen string ifade içerisinde bulunan sayısal değeri bulur ve geri döndürür. Zamanlı işlemlerin yapılacağı komut ifadelerinde kullanılmıştır.

**Public String zamanAyarla (int deger, int zmn):** “StrSayiBul” fonksiyonunca bulunan sayısal değer bu fonksiyon tarafından mevcut saniye, dakika, saat değerlerine eklenir. Oluşan yeni değer “saat: dakika: saniye” formatında geri döndürülür. Bu veri zamanlı işlev için shared preference alanına kaydedilecek zaman değeridir.

### 3.8 “SalonIslemleri.java” Sınıfı

Gelen ses tanıma sonucundan salona ait yapılacak işlemlerin belirlendiği sınıftır (Bkz Ek5 ). Yapılacak işleme göre komut kodu oluşturulur ve geri döndürülür.

**public String SalonIslem (String Sonuc):** Salonda yapılacak olan işleme göre komut kodunun oluşturulduğu fonksiyondur. Bu kodların bazıları Tablo 1’de verilmiştir. Bunlara ek olarak da işleme göre uygun string ifadeler oluşturulur.

### 3.9 "IkiliIslev.java" Sınıfı

Ses tanıma sonucunda ikili işlev işlemlerinin var olup olmadığına bakar. Eğer varsa ikili işlev işlem kodunu oluşturur ve geri döndürür. İkili işlev kodu yapılacak olan işlemlerin komut kodunu virgülle ayrılarak yan yana yazılması şeklindedir. Örneğin "001, 016" gibi (Bkz Ek6).

**public String IkiliIslev (String sonuc):** İkili işlevi bulur ve kod oluşturulması için "yer1, yer2, islev" yapısını oluşturur. Buradaki işlev için iki farklı durum söz konusudur. Örneğin "aç, kapat" veya "kapat, aç" şeklindedir.

**public int IslevYeri (String sonuc, String Liste1, String Liste2):** Ses tanıma sonucunda işlevin "aç, kapat" veya "kapat, aç" işlevlerinin hangisi olduğunu bulur. Liste1 ve Liste2 "yak" ve "söndür" işlevlerine ait anahtar kelimelerdir. 1 sayısal değeri "aç, kapat", 2 sayısal değeri ise "kapat, aç" işlevini ifade eder. -1 ise işlev yok demektir.

**public String IslevYerKomut (String Islev\_yer):** "yer1, yer2, islev" şeklinde gelen ifadeden ikili işlev kodunu "kod1, kod2" şeklinde oluşturur ve geri döndürür.

**public void bekle (final Long ms):** Milisaniye cinsinden zaman gecikmesinin oluşmasını sağlar. Ana sınıfta ikili işlev gerçekleştirilirken donanım komut kodu "kod1, kod2" şeklindeki ifadeden "kod1" ve "kod2" ayrıştırılarak belli bir zaman gecikmesi sağlanarak gönderilir.

### 3.10 "AnahtarListe.java" Sınıfı

Hazırlanan anahtar kelime listelerinin string sabitler halinde tutulduğu sınıftır. Diğer sınıflar tarafından anahtar listeler bu sınıftan çekilir (Bkz. Ek7).

**public String getir (int Liste):** Kullanılacak olan anahtar listenin geri döndürüldüğü fonksiyondur. Hazırlanan her bir listeye tam sayı olarak bir isim verilmiştir. Bu sayısal değere göre ifadeler string olarak geri döndürülür.

## Sonuçlar

- ✓ Yaptığımız anketlerden elde edilen komutlar ve kendi türettiğimiz komutların farklı söylem biçimleri Android mobil telefon ve tabletlerde denenmiş ve başarılı ses tanıma sonuçlarında istenen işlemin farklı söylem biçimlerini tanıyarak bir işlev için aynı işlem kodu üretilmiştir. Örneğin "salon lambasını yak" komutunun yaklaşık 350 biçimi ve "salon lambasını söndür" komutunun yine yaklaşık 500 biçimi elde edilmiş ve denenmiştir (Bkz Ek8). Bu durumda sadece lamba işlemleri için 850 farklı söylem geliştirilmiş ve 6 farklı oda düşünüldüğünde bu kategoride 5000'e yakın farklı söylemin tanınabildiği görülmüştür. Bu durum diğer kategorilere uygulandığında ise komut sayısı 10.000'i bulmaktadır.
- ✓ Eskişehir'de farklı milletlerden insanların bulunması komutların kullanımı konusunda çeşitliliğe yol açmıştır. Bu tür söylem biçimleri de denenmiştir. Örneğin macur olarak tabir edilen insanlar tarafından kullanılan "salon kuruşkasını aç" veya ege yöresinde kullanılan "salonun şavkını kapa" ifadelerindeki "kuruşka" ve "şavk" anahtar kelimelerinden "kuruşka" ses tanıma motoru tarafından tanınmazken "şavk" kelimesi zorda olsa tanınmıştır. Bu yönüyle bu tür söylem biçimleri ses tanıma motorunun tanımamasından dolayı başarısızdır. Fakat aynı komut cümleleri string ifade olarak giriş yapıldığında, komutlar için aynı işlem kodu üretilmiştir. Bundan dolayı hazırlanan programa klavye ile komut girişi de eklenmiştir. Bu durumda da program konuşmasında problem olan kişilerce de rahatlıkla kullanılabilecektir.

- ✓ Çalışma bir eve uygulandığında burada yaşayan ve konuşmasında problem olmayan, yaşlı, hasta veya bakıma muhtaç insanlar bu programla evlerindeki cihazların kontrolünü rahatlıkla yapabileceklerdir.
- ✓ Bluetooth modüller için yaklaşık 11 metre olan iletişim mesafesinin bir ev için yeterli olmadığı görülmüştür.
- ✓ Son olarak da yapılan bu çalışmayla hazırlanan programın rahatlıkla uygun donanımın olduğu evlere uygulanabileceği görülmüştür.

## Öneriler

- ✓ Wifi (Kablosuz) ile haberleşmenin yapılarak haberleşmenin şifrelenmesi daha uygun olacaktır. Elektronik sisteme internet üzerinden erişim de eklenerek mekâna olan bağımlılık ortadan kaldırılabilir. Ses tanıma olmadığı durumlarda da cihazların kontrolü telefon üzerinden manuel yapılabilir.
- ✓ Belirlenen komut işlem kodlarının model evdeki donanıma göre olduğu unutulmamalıdır. Bir akıllı ev sistemi kontrol paneliyle haberleşme durumunda ise burada kullanılan işlem kodlarına ihtiyaç vardır. Program bu işlem kodlarını kaydederek kullanabilmelidir.
- ✓ Programın esnek bir yapıda olması açısından anahtar kelimelerin bulunduğu sınıf dosyası veri tabanı oluşturularak geliştirilmelidir. Bu veri tabanına erişim internet üzerinden sağlanarak yeni tanınacak komut söylemleri anahtar kelimelerinin de veri tabanına eklenmesi sağlanmalıdır.
- ✓ Ülkemize ait şive ve lehçe farklılıklarının ses tanıma motoru tarafından yeterince tanınmadığı görülmüştür. Bu konuda ses tanıma motorunun ses tanımlamasının eğitilmesi gerektiği önerilmektedir.
- ✓ Türkçe başarılı ve hızlı bir ses tanıma için mobil telefon üzerinde (local) ses tanıma işleminin yapılabilmesine ihtiyaç vardır. Ülkemizde bu konuyla ilgili çalışma yapan firmaların verilerini son kullanıcılarla paylaşmadığı görülmüştür. Bu tür çalışmaların yapılması önerilmektedir.

## Kaynaklar

- [1] <http://developer.android.com/about/index.html> , (Erişim, 05.12.2013)
- [2] <http://developer.android.com/guide/index.html> , (Erişim, 05.12.2013)
- [3] <http://android-developers.blogspot.com/2010/03/speech-input-api-for-android.html> , (Erişim, 05.12.2013)
- [4] <http://shiftdelete.net/jelly-beanin-ses-tanima-performansi-38366.html> , (Erişim, 12.10.2013)
- [5] <http://www.ispeech.org/apps/ispeechhome> , (Erişim, 10.12.2013)
- [6] Professional Android Sensor Programming, G. Milette, A. Stroud, 2012
- [7] Merhaba Android, M.Önder, A. O. Mermerkaya, 2012
- [8] <http://ab.org.tr/ab13/bildiri/285.pdf> (Erişim, 05.12.2013)
- [9] [http://www.robotiksystem.com/pic18f452\\_mikrodenetleyici\\_ozellikleri.html](http://www.robotiksystem.com/pic18f452_mikrodenetleyici_ozellikleri.html) , (Erişim, 05.10.2013)
- [10] CCS C ile PIC Programlama, S. Çiçek, 2012
- [11] <http://www.direnc.net/HC06-BLUETOOTH-MODUL,PR-7874.html> , (Erişim, 05.10.2013)
- [12] <http://developer.android.com/reference/android/speech/SpeechRecognizer.html> , (Erişim, 15.10.2013)
- [13] <http://developer.android.com/reference/android/speech/RecognitionListener.html> , (Erişim, 15.10.2013)
- [14] <http://developer.android.com/reference/java/lang/String.html> , (Erişim, 15.10.2013), <http://www.elektrikport.com/teknik-kutuphane/javada-stringler-bolum-1-elektrikport-akademi/6874#ad-image-0> (Erişim, 15.10.2013)
- [15] <http://developer.android.com/reference/android/content/SharedPreferences.html> , (Erişim, 22.10.2013), <http://www.programlamadefteri.com/mobil/shared-preferences-ile-androidde-veri-saklama-519> (Erişim, 22.10.2013)
- [16] <http://developer.android.com/guide/components/services.html> , (Erişim, 25.10.2013)
- [17] <https://github.com/cxemnet/CxemCar1> , (Erişim, 10.10.2013)
- [18] <http://developer.android.com/reference/android/app/Activity.html> , (Erişim, 05.10.2013) , <http://muhendisiz-biz.blogspot.com/2012/01/android-activity.html> (Erişim, 05.10.2013)
- [19] <http://developer.android.com/reference/android/os/Handler.html> , (Erişim, 12.10.2013)
- [20] <http://developer.android.com/reference/android/content/Intent.html> , (Erişim, 12.10.2013)

## **Ekler**

### ***Ek1 : "MainActivity.java" Sınıfına ait bazı fonksiyonlar***

```
public void sr_baslat() {
    sr.cancel();
    sr_basla_dur=true;
    pbar.setVisibility(pbar.VISIBLE);

    Intent intent= new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.ACTION_VOICE_SEARCH_HANDS_FREE);

    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.LAN
        GUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
        this.getPackageName());
    intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS,10);
    kulaklikBagli=false;
    try { sr.startListening(intent);} catch (Exception e) { }
} // sr_baslat()

private final Handler srHandler=new Handler() {
    @Override
    public void handleMessage(Message msg) {
        String str = new String();
        switch (msg.what) {
            case ON_ERROR:
                Log.d("SR","HATA");
                if(msg.obj.toString().contains("ERROR_SERVER")){
                    Toast.makeText(getApplicationContext(),
                        msg.obj.toString()+" \n Ağ Bağlantısı
                        yokk.",Toast.LENGTH_SHORT).show();
                    sr.cancel();
                    pbar.setVisibility(pbar.INVISIBLE);
                }
                else{
                    Toast.makeText(getApplicationContext(),msg.obj.toString(
                        )+"\nSes tanıma tekrar başlatıldı.",
                        Toast.LENGTH_SHORT).show();
                    pbar.setVisibility(pbar.INVISIBLE);
                    sr.cancel();sr_baslat();
                }
                break;
            case ON_RESULT:
                str=msg.obj.toString();
                sonuc_text.setText("");
                if(komut.islem(str)=="dur"){
                    sr.cancel();
                    pbar.setVisibility(pbar.INVISIBLE);
                    sr_basla_dur=false;
                    Toast.makeText(getApplicationContext(), "Ses tanıma
                        işlemi durduruldu...",Toast.LENGTH_SHORT).show();
                }else
                if(komut.islem(str)=="bağlan"){
                    bl.BT_Connect(bl_adres);
                    Toast.makeText(getApplicationContext(),
                        "Bluetooth bağlantı yapıldı...",
                        Toast.LENGTH_SHORT).show();
                }else{
```

```

        if(komut.islem(str).contains("zaman")){
            Toast.makeText(getApplicationContext(),
                "Komut="+komut.islem(str), Toast.LENGTH_SHORT).show();
        }else
        if( komut.islem(str).length()>3){
            IkiliIslev islev=new IkiliIslev();
            String[]k=komut.islem(str).split(",");
            //Log.d("Main Aactivitiy", "K0:"+k[0]);
            komutGonder(k[0]);
            islev.bekle(1000);
            komutGonder(k[1]);
        }else
            komutGonder(komut.islem(str));
        if(komut.islem(str)==""){ //Hata uyarısı
            try {
                Uri notification =
                    RingtoneManager.getDefaultUri(RingtoneManager.TYPE
                        _NOTIFICATION);
                Ringtone r =
                    RingtoneManager.getRingtone(getApplicationContext(
                        ), notification);
                r.play();
            } catch (Exception e) {}
        }

        if(sr_basla_dur==true){
            Log.d(TAG, "SR Baslatma islemi");
            sr.cancel();sr_baslat();
            sonuc_text.setText(str);
        }

        Toast.makeText(getApplicationContext(),
            "Komut="+komut.islem(str), Toast.LENGTH_SHORT).show();

        break;
    case ON_PARTIAL_RESULT:

        Toast.makeText(getApplicationContext(),msg.obj.toString(),Toas
            t.LENGTH_SHORT).show();
        break;
    case ON_RMS:
        konus_btn.setImageResource(R.drawable.accept);
        break;
        ///////
    case cBluetooth.BL_NOT_AVAILABLE:
        Log.d(cBluetooth.TAG, "Bluetooth mevcut değil. Bitir");
        Toast.makeText(getApplicationContext(), "Bluetooth mevcut değil",
            Toast.LENGTH_SHORT).show();
        finish();
        break;
    case cBluetooth.BL_INCORRECT_ADDRESS:
        Log.d(cBluetooth.TAG, "MAC adres yanlış");
        Toast.makeText(getApplicationContext(), "Bluetooth adresi yanlış",
            Toast.LENGTH_SHORT).show();
        break;
    case cBluetooth.BL_REQUEST_ENABLE:
        Log.d(cBluetooth.TAG, "Bluetooth açılması
            gerekli");
        BluetoothAdapter.getDefaultAdapter();
        Intent enableBtIntent = new
            Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

```

```

        startActivityForResult(enableBtIntent, 1);
        break;
    case cBluetooth.BL_SOCKET_FAILED:
        Toast.makeText(getBaseContext(), "Socket başarısız! İletişim hatası.", Toast.LENGTH_SHORT).show();
        finish();
        break;
    case cBluetooth.RECIEVE_MESSAGE:

        String msj_oku= msg.obj.toString();
        Log.d("SR",msj_oku);
        msj_oku.replace("cavap", "");

        sonuc_text.setText(sonuc_text.getText().toString()+"\n"+msj_oku+"\n");
        if(!msj_oku.contains("OK")){//Hata uyarısı
            try {
                Uri notification =
                    RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
                Ringtone r =
                    RingtoneManager.getRingtone(getApplicationContext(), notification);
                r.play();
            } catch (Exception e) {}
        }

        break;
    }
}
};//srHandler sonu

```

### ***Ek2 : “Listener.java” Sınıfına ait bazı fonksiyonlar***

```

public void onError(int error)
{
    Log.d(TAG, hata_kodu_gonder(error));

    srHandler.obtainMessage(MainActivity.ON_ERROR,hata_kodu_gonder(error).length(),-1,hata_kodu_gonder(error)).sendToTarget();
}

public void onResults(Bundle results){
    String str = new String();
    ArrayList<String> data =
        results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
    float[] skor= results.getFloatArray(SpeechRecognizer.CONFIDENCE_SCORES);
    for (int i = 0; i < data.size(); i++){
        str+="->"+data.get(i)+"\n";
    }
    Log.d(TAG,str);
    srHandler.obtainMessage(MainActivity.ON_RESULT,str.length(),-1,str).sendToTarget();
}

```

### ***Ek3 : “KomutIslemleri.java” Sınıfına ait bazı fonksiyonlar***

```

//Bu sınıf tarafından kullanılan sınıflar
SalonIslemleri salon=new SalonIslemleri();
SrStrIslem islem=new SrStrIslem();
IkiliIslev lambalar=new IkiliIslev();

```



```

AnahtarListe anahtar=new AnahtarListe();

public String islem(String sonuc){
    String komut="";
    //Telefona verilecek komutlarla ilgili işlemler.
    if(sonuc.contains("neksus")){
        sonuc.replace("neksus","");
        if(islem.listeVarMI(sonuc,
            anahtar.getir(anahtar.BLBAGLAN)) komut="bağlan";

        if(islem.listeVarMI(sonuc,anahtar.getir(anahtar.SRDUR)) komut="dur";

    }
    //Lambaların hepsini söndürme işlemi
    if(islem.listeVarMI(sonuc,anahtar.getir(anahtar.LAMBALAR)) &&
        islem.listeVarMI(sonuc,anahtar.getir(anahtar.SONDUR))) komut="000";
    //Salon ile ilgili işlemler
    if((lambalar.ikiliIslev(sonuc)==null) &&
        islem.listeVarMI(sonuc, anahtar.getir(anahtar.SALON))) {
        sonuc=sonuc.replace("salon","");
        komut=salon.SalonIslem(sonuc);
    }
    if(lambalar.ikiliIslev(sonuc)!=null){
        komut=lambalar.ikiliIslev(sonuc);
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("oturma
        oda") && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.YAK))) {
        komut="005";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("oturma oda")
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.SONDUR))) {
        komut="006";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("çocuk
        oda") && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.YAK))) {
        komut="007";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("çocuk
        oda") && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.SONDUR))) {
        komut="008";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("mutfak")
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.YAK))) {
        komut="009";
    }
    if((lambalar.ikiliIslev(sonuc)==null) &&
        sonuc.contains("mutfak") && islem.listeVarMI(sonuc,anahtar.getir(anaht
            ar.LAMBA)) && islem.listeVarMI(sonuc, anahtar.getir(anahtar.SONDUR))) {
        komut="016";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("yatak
        oda") && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.YAK))) {
        komut="017";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("yatak
        oda") && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.SONDUR))) {
        komut="018";
    }
    if((lambalar.ikiliIslev(sonuc)==null) && sonuc.contains("antre")
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
        && islem.listeVarMI(sonuc, anahtar.getir(anahtar.YAK))) {
        komut="023";
    }
}

```

```

        if((lambalar.ikiliIslev(sonuc)==null) &&
            (sonuc.contains("antre") || sonuc.contains("hol"))
            &&islem.listeVarMI(sonuc, anahtar.getir(anahtar.LAMBA))
            &&islem.listeVarMI(sonuc, anahtar.getir(anahtar.SONDUR))) {
            komut="024"; }
        if(sonuc.contains("kombi aç")){ komut="068"; }
        if(sonuc.contains("kombi kapat")){ komut="069"; }
        if(sonuc.contains("perdeyi aç")){ komut="090"; }
        if(sonuc.contains("perdeyi kapat")){ komut="091"; }
        return komut;
    } //String islem

```

#### ***Ek4 : “SrStrIslem.java” Sınıfına ait bazı fonksiyonlar***

// Bu sınıf tarafından kullanılan sınıflar  
 AnahtarListe anahtar=new AnahtarListe();

```

public boolean Uygula(String sonuc, int islem){
    String liste="";
    String listeZaman="saat,dakika,dakka,saniye";
    String listeBos="";
    switch (islem) {
        case LAMBA:
            liste=anahtar.getir(anahtar.LAMBA);
            return listeVarMI(sonuc, liste);
        case YAK:
            liste=anahtar.getir(anahtar.YAK);
            return listeVarMI(sonuc, liste);
        case SONDUR:
            liste=anahtar.getir(anahtar.SONDUR);
            return listeVarMI(sonuc, liste);
        case ZAMAN_YAK:
            liste=anahtar.getir(anahtar.YAK);
            return (listeVarMI(sonuc, liste)&&listeVarMI(sonuc,
                listeZaman));
        case ZAMAN_SONDUR:
            liste=anahtar.getir(anahtar.SONDUR);
            return (listeVarMI(sonuc, liste)&&listeVarMI(sonuc,
                listeZaman));
        case ZAMAN_YAK_SONDUR:
            liste=anahtar.getir(anahtar.YAK);
            listeBos=anahtar.getir(anahtar.SONDUR);
            if(listeVarMI(sonuc,liste)&&listeVarMI(sonuc,listeBos)){
                String[] s1=liste.split(",");
                String[] s2=listeBos.split(",");
                for (int i = 0; i < s1.length; i++) {
                    for (int j = 0; j < s2.length; j++) {
                        if(sonuc.indexOf(s1[i])<sonuc.indexOf(s1[j]))
                            return true;
                    }
                }
            }
            break;
    }
    return false;
} //Uygula

public boolean listeVarMI(String sonuc, String liste){
    String[] ayir=liste.split(",");
    for (int i = 0; i < ayir.length; i++) {

```

```

        if(sonuc.contains(ayir[i]))return true;
    }
    return false;
} //listeVarMI

public String StrSayiBul(String s){
    String sayi = "";
    for (int i = 0; i < s.length(); i++) {

        if(Character.isDigit(s.charAt(i)))sayi+=String.valueOf(s.charAt(i));
    }
    //ses tanıma sonuçlarından 1-?? arasında farklı sonuc geldiği için
    //bunlar arasından aynı olan 1, 2, 3 basamaklı sayı döndürülür.
    try {
        if(sayi.length()%2==0)sayi=sayi.substring(0,2); //Sayı iki basamaklı
ise
        else if(sayi.length()%3==0)sayi=sayi.substring(0,3); //Sayı üç
basamaklı ise
        else if(sayi.length()<6)sayi=sayi.substring(0,1); //Sayı bir
basamaklı ise
    } catch (Exception e) {}
    return sayi;
} //StrSayiBul

public String zamanAyarla(int deger, int zmn){
    int saniye=0,dakika=0,saat=0;
    Calendar takvim = Calendar.getInstance();
    saniye=takvim.get(Calendar.SECOND);
    dakika=takvim.get(Calendar.MINUTE);
    saat=takvim.get(Calendar.HOUR_OF_DAY);
    switch (zmn) {
        case 0: //Saniye
            if((saniye+deger)>59){
                dakika++;
                if(dakika>59){saat++;dakika=dakika-60;}
                saniye=(saniye+deger)-60;
                return
                String.valueOf(saat)+":"+String.valueOf(dakika)+":"+String.
                valueOf(saniye);
            }else return
            String.valueOf(saat)+":"+String.valueOf(dakika)+":"+String.val
            ueOf(saniye+deger);
        case 1: //Dakika
            if((dakika+deger)>59){
                dakika=(dakika+deger)-60;saat++;
                if(saat>23)saat=saat-24;
                return
                String.valueOf(saat)+":"+String.valueOf(dakika)+":"+String.
                valueOf(saniye);
            }else return
            String.valueOf(saat)+":"+String.valueOf(dakika+deger)+":"+String.
            valueOf(saniye);
        case 2: //Saat
            if((saat+deger)>23){
                saat=(saat+deger)-24;
                return
                String.valueOf(saat)+":"+String.valueOf(dakika)+":"+String.
                valueOf(saniye);
            }else return
            String.valueOf(saat+deger)+":"+String.valueOf(dakika)+":"+String.
            valueOf(saniye);
    }
}

```

```

        default:
            return null;
    }
} //zamanAyarla

```

### Ek5 : “SalonIslemleri.java” Sınıfına ait bazı fonksiyonlar

```

// Bu sınıf tarafından kullanılan sınıflar
private SrStrIslem islem=new SrStrIslem();

public String SalonIslem(String sonuc){
    String komut="";
    if(islem.Uygula(sonuc, islem.LAMBA) || islem.Uygula(sonuc, islem.YAK) ||
        islem.Uygula(sonuc, islem.SONDUR)) {
        if(islem.Uygula(sonuc, islem.YAK)) komut="001";
        if(islem.Uygula(sonuc, islem.SONDUR)) komut="002";

        //Salon Lambasını ... saniye/dakika/saat sonra aç
        if(islem.Uygula(sonuc, islem.ZAMAN YAK))
        {
            if(sonuc.contains("saniye")) {
                komut="zaman salon saniye="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),0);
            }
            if(sonuc.contains("dakika") || sonuc.contains("dakka")) {
                komut="zaman salon dakika="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),1);
            }
            if(sonuc.contains("saat")) {
                komut="zaman salon saat="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),2);
            }
        }
        //Salon Lambasını ... saniye/dakika/saat sonra kapat
        if(islem.Uygula(sonuc, islem.ZAMAN SONDUR)) {
            if(sonuc.contains("saniye")) {
                komut="zaman salon saniye="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),0);
            }
            if(sonuc.contains("dakika") || sonuc.contains("dakka")) {
                komut="zaman salon dakika="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),1);
            }
            if(sonuc.contains("saat")) {
                komut="zaman salon saat="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),2);
            }
        }
        //Salon lambasını aç .... saniye/dakika/saat sonra söndür
        if(islem.Uygula(sonuc, islem.ZAMAN YAK SONDUR)) {
            if(sonuc.contains("saniye")) {
                komut="zaman salon saniye="+
                    islem.zamanAyarla(Integer.parseInt
                        (islem.StrSayiBul(sonuc)),0);
            }
        }
    }
}

```

```

        if (sonuc.contains("dakika") || sonuc.contains("dakka")) {
            komut = "zaman salon dakika=" +
            islem.zamanAyarla(Integer.parseInt
                (islem.StrSayiBul(sonuc)), 1);
        }
        if (sonuc.contains("saat")) {
            komut = "zaman salon saat=" +
            islem.zamanAyarla(Integer.parseInt
                (islem.StrSayiBul(sonuc)), 2);
        }
    }
} //Lamba
if (sonuc.contains("perde")) {
    if (sonuc.contains("aç")) komut = "090";
    if (sonuc.contains("kapat")) komut = "091";
}

return komut;
} //SalonIslem

```

### ***Ek6 : "IkiliIslev.java" Sınıfına ait bazı fonksiyonlar***

// Bu sınıf tarafından kullanılan sınıflar  
 AnahtarListe anahtar=new AnahtarListe();

```

public String ikiliIslev(String sonuc){
    boolean dur=false;
    String liste=anahtar.getir(anahtar.ODA);
    String komut="", liste2="";
    String[] yer=liste.split(",");

    for (int i = 0; i < yer.length; i++) {
        for (int j = 0; j < yer.length; j++) {
            if (sonuc.contains(yer[i]) && sonuc.contains(yer[j])) {
                if (sonuc.indexOf(yer[i]) != sonuc.indexOf(yer[j])) {
                    liste=anahtar.getir(anahtar.YAK);
                    liste2=anahtar.getir(anahtar.SONDUR);
                    if (listeVarMI(sonuc, liste) && listeVarMI(sonuc, liste2)) {
                        Log.d("ikili islev", "işlev=" +
                            String.valueOf(islevYeri(sonuc, liste, liste2)));
                        dur=true;

                        if (sonuc.indexOf(yer[i]) < sonuc.indexOf(yer[j])) {
                            Log.d("ikili islev", "işlev="+yer[i]+yer[j]);
                            //komut="yer1,yer2,islev"
                            //islev 1: aç,kapat islev 2:kapat,aç
                            komut=yer[i]+"," +yer[j]+"," +
                            String.valueOf(islevYeri(sonuc, liste,
                                liste2));
                            return islevYerKomut(komut);
                        } else {
                            Log.d("ikili islev", "işlev="+yer[j]+yer[i]);
                            komut=yer[j]+"," +yer[i]+"," +
                            String.valueOf(islevYeri(sonuc, liste,
                                liste2));
                            return islevYerKomut(komut);
                        }
                    }
                }
            }
        }
    } //if(listeVarMI...
    dur=true;
    break;
} //if(sonuc.indexOf(yer[i])...

```

```

        }//if(sonuc.contains...
    }//for j
    if(dur)break;
}//for i
return null;
}//ikiliIslev

public String islevYerKomut(String islev_yer){
    String[] islev_yer_ayir=islev_yer.split(",");
    String komut="";
    if(listeVarMI(islev_yer, anahtar.getir(anahtar.SALON))){
        switch (Integer.parseInt(islev_yer_ayir[2])) {
            case 1:
                if(islev_yer_ayir[1].contains("mutfağ")||
                    islev_yer_ayir[1].contains("mutfak"))
                    komut="001,016";//salon aç mutfağı kapat
                if(islev_yer_ayir[1].contains("oturma"))
                    komut="001,006";//salon aç oturma odası kapat
                if(islev_yer_ayir[1].contains("yatak"))
                    komut="001,018";//salon aç yatak odası kapat
                if(islev_yer_ayir[1].contains("çocuk"))
                    komut="001,008";//salon aç çocuk odası kapat
                if(islev_yer_ayir[1].contains("antre"))
                    komut="001,024";//salon aç antre kapat
                break;
            case 2:
                if(islev_yer_ayir[1].contains("mutfağ")||
                    islev_yer_ayir[1].contains("mutfak"))
                    komut="002,009";//salon kapat mutfağı aç
                if(islev_yer_ayir[1].contains("oturma"))
                    komut="002,005";//salon kapat oturma odası aç
                if(islev_yer_ayir[1].contains("yatak"))
                    komut="002,017";//salon kapat yatak odası aç
                if(islev_yer_ayir[1].contains("çocuk"))
                    komut="002,007";//salon kapat çocuk odası aç
                if(islev_yer_ayir[1].contains("antre"))
                    komut="002,023";//salon kapat antre aç
                break;
        }//Switch
        //.....//
    }//if(listeVarMI(islev_yer, anahtar.getir(anahtar.SALON)))...
    return komut;
}//islevYerKomut

```

```

public int islevYeri(String sonuc,String listel,String liste2){
    String[] sonuclar=sonuc.split("\n");
    for(int k=0;k<sonuclar.length;k++){ //Bütün sonuçlarda ARA
        if(listeVarMI(sonuclar[k],listel)&&
            listeVarMI(sonuclar[k],liste2)){
            //herhangi bir sonuçta var ise
            String[] l1s=listel.split(",");
            String[] l2s=liste2.split(",");
            for (int i = 0; i < l1s.length; i++) {
                for (int j = 0; j < l2s.length; j++) {
                    if(sonuclar[k].contains(l1s[i])&&
                        sonuclar[k].contains(l2s[j])) {
                        //iki işlev bulunduysa
                        if(sonuclar[k].indexOf(l1s[i]) <
                            sonuclar[k].indexOf(l2s[j])){
                            return 1;
                        }
                    }
                }
            }
        }
    }
}

```

```

        if (sonuclar[k].indexOf(l1s[i]) >
            sonuclar[k].indexOf(l2s[j])) {
            return 2;
        }
    }
} //for j
} //for i
} //if(listeVarMI
} //for k
return -1;
} //islevYeri

```

### Ek7 : "AnahtarListe.java" Sınıfına ait sınıfına bazı fonksiyonlar

```

//Döndürülecek listeyi tutan sabitler
public final static String SrDur="dur,bitir,son,sonlandır,iptal,kes,yapma";
public final static String BlBaglan="bağlan,bağlantı,bul,kur,oluştur";
public final static String Odalar="salon,sabah,mutfak,mutfak,oturma,yatak,
çocuk,antre,hol";
public final static String Salon="salon,büyük oda,yemek odası,misafir
odası,sabah";
public final static String Lamba="lamba,ışık,ışığı,elektrik,elektriği,
anahtar,ceyran,ceyran, cereyan,şavk";
public final static String LambaYak="yak,aç,düğmeye bas,aydınlat,on,çak";
public final static String LambaSondur="söndür,sondur,kapat,düğmeye
bas,karart,kapa,off,of";
public final static String Lambalar="lambaları,ışıkları,elektrikleri,
hepsini,tüm,tümünü,tamamını";
//.....//
public String getir(int liste){
    String anahtar="";
    switch (liste) {
        case SRDUR:anahtar=SrDur;break;
        case BLBAGLAN:anahtar=BlBaglan;break;
        case SALON:anahtar=Salon;break;
        case LAMBA:anahtar=Lamba;break;
        case YAK:anahtar=LambaYak;break;
        case SONDUR:anahtar=LambaSondur;break;
        case LAMBALAR:anahtar=Lambalar;break;
        case ODA:anahtar=Odalar;break;
        //.....//
    }
    return anahtar;
}

```

### Ek8 : "Salon Lambasını Yak" temel komutu için örnek komut söylemleri

	"Salon Lambasını Yak"			
1	salonun lambasını yak	salon ceyranını aydınlat	salondaki şavkı aç	büyük odadaki ceyranı on
2	salondaki lambayı yak	salonun ceyranını aydınlat	salon şavkını yak	büyük odanın ceyranını aç
3	salon ışığını yak	salondaki ceyranı aydınlat	salonun şavkını yak	büyük odadaki ceyranı aç
4	salondaki ışığı yak	salon ceyranını on	salondaki şavkı yak	büyük odanın ceyranını yak
5	salonun ışığını yak	salonun ceyranını on	salon şavkını aydınlat	büyük odadaki ceyranı yak
6	salon lambasını aç	salondaki ceyranı on	salonun şavkını aydınlat	büyük odanın ceyranını

				aydınlat
7	salonun lambasını aç	salon ceryanını aç	salondaki şavkı aydınlat	büyük odadaki ceryanı aydınlat
8	salondaki lambayı aç	salonun ceryanını aç	salon şavkını on	büyük odanın ceryanını on
9	salon ışığını aç	salondaki ceryanı aç	salonun şavkını on	büyük odadaki ceryanı on
10	salonun ışığını aç	salon ceryanını yak	salondaki şavkı on	büyük odanın ceryanını çak
11	salondaki ışığı aç	salonun ceryanını yak	salon şavkını çak	büyük odadaki ceryanı çak
12	salon lambasını aydınlat	salondaki ceryanı yak	salonun şavkını çak	büyük odanın lambasını çak
13	salonun lambasını aydınlat	salon ceryanını aydınlat	salondaki şavkı çak	büyük odadaki lambayı çak
14	salondaki lambayı aydınlat	salonun ceryanını aydınlat	büyük odanın lambasını yak	büyük odadaki ışığı çak
15	salon ışığını aydınlat	salondaki ceryanı aydınlat	büyük odadaki lambayı yak	büyük odanın ışığını çak
16	salonun ışığını aydınlat	salon ceryanını on	büyük odanın ışığını yak	büyük odadaki elektriğini çak
17	salondaki ışığı aydınlat	salonun ceryanını on	büyük odadaki ışığı yak	büyük odadaki elektriği çak
18	salon lambasını on	salondaki ceryanı on	büyük odanın lambasını aç	büyük odanın ceyranını çak
19	salonun lambasını on	salon ceryanını aç	büyük odadaki lambayı aç	büyük odadaki ceyranı çak
20	salondaki lambayı on	salonun ceryanını aç	büyük odanın ışığını aç	büyük odanın ceryanını çak
21	salon ışığını on	salondaki ceryanı aç	büyük odadaki ışığı aç	büyük odadaki ceryanı çak
22	salonun ışığını on	salon ceryanını yak	büyük odanın lambasını aydınlat	büyük odanın şavkını aç
23	salondaki ışığı on	salonun ceryanını yak	büyük odadaki lambayı aydınlat	büyük odadaki şavkı aç
24	salon elektrik aç	salondaki ceryanı yak	büyük odanın ışığını aydınlat	büyük odanın şavkını yak
25	salonun elektrik aç	salon ceryanını aydınlat	büyük odadaki ışığı aydınlat	büyük odadaki şavkı yak
26	salondaki elektrik aç	salonun ceryanını aydınlat	büyük odanın lambasını on	büyük odanın şavkını aydınlat
27	salon elektrik yak	salondaki ceryanı aydınlat	büyük odadaki lambayı on	büyük odadaki şavkı aydınlat
28	salonun elektrik yak	salon ceryanını on	büyük odanın ışığını on	büyük odanın şavkını on
29	salondaki elektrik yak	salonun ceryanını on	büyük odadaki ışığı on	büyük odadaki şavkı on
30	salon elektrik aydınlat	salondaki ceryanı on	büyük odanın elektriğini aç	büyük odanın şavkını çak
31	salonun elektrik aydınlat	salon ceryanını çak	büyük odadaki elektriği aç	büyük odadaki şavkı çak
32	salondaki elektrik aydınlat	salonun ceryanını çak	büyük odanın elektriğini yak	oğlum salon lambasını yak
33	salon elektrik on	salondaki ceryanı çak	büyük odadaki elektriği yak	oğlum salonun lambasını yak
34	salonun elektrik on	salon lambasını çak	büyük odanın elektriğini on	oğlum salondaki lambayı yak
35	salondaki elektrik on	salonun lambasını çak	büyük odadaki elektriği on	oğlum salon ışığını yak
36	salon elektriğini yak	salondaki lambayı çak	büyük odanın ceyranını aç	oğlum salondaki ışığı yak
37	salonun elektriğini yak	salon ışığını çak	büyük odadaki ceyranı aç	oğlum salonun ışığını yak
38	salondaki elektriği yak	salondaki ışığı çak	büyük odanın ceyranını yak	oğlum salon lambasını aç
39	salon elektriğini aç	salonun ışığını çak	büyük odadaki ceyranı yak	oğlum salonun lambasını aç
40	salonun elektriğini aç	salon elektriğini çak	büyük odanın ceyranını aydınlat	oğlum salondaki lambayı aç
41	salondaki elektriği aç	salonun elektriğini çak	büyük odadaki ceyranı aydınlat	oğlum salon ışığını aç
42	salon elektriğini on	salondaki elektriği çak	büyük odanın ceyranını on	oğlum salonun ışığını aç
43	salonun elektriğini on	salon ceyranını çak	büyük odadaki ceyranı on	oğlum salondaki ışığı aç
44	salondaki elektriği on	salonun ceyranını çak	büyük odanın ceryanını aç	oğlum salon lambasını aydınlat
45	salon ceyranını aç	salondaki ceyranı çak	büyük odadaki ceryanı aç	oğlum salonun lambasını aydınlat
46	salonun ceyranını aç	salon ceryanını çak	büyük odanın ceryanını yak	oğlum salondaki lambayı aydınlat
47	salondaki ceyranı aç	salonun ceryanını çak	büyük odadaki ceryanı yak	oğlum salon ışığını aydınlat
48	salon ceyranını yak	salondaki ceryanı çak	büyük odanın ceryanını aydınlat	oğlum salonun ışığını aydınlat
49	salonun ceyranını yak	salon şavkını aç	büyük odadaki ceryanı aydınlat	oğlum salondaki ışığı aydınlat
50	salondaki ceyranı yak	salonun şavkını aç	büyük odanın ceryanını on	oğlum salon lambasını on
1	oğlum salonun lambasını on	oğlum salon ceryanını çak	oğlum büyük odadaki elektriği aç	
2	oğlum salondaki lambayı on	oğlum salonun ceryanını çak	oğlum büyük odanın elektriğini yak	
3	oğlum salon ışığını on	oğlum salondaki ceryanı çak	oğlum büyük odadaki elektriği yak	
4	oğlum salonun ışığını on	oğlum salon lambasını çak	oğlum büyük odanın elektriğini on	
5	oğlum salondaki ışığı on	oğlum salonun lambasını çak	oğlum büyük odadaki elektriği on	
6	oğlum salon elektriğini yak	oğlum salondaki lambayı çak	oğlum büyük odanın ceyranını aç	
7	oğlum salonun elektriğini yak	oğlum salon ışığını çak	oğlum büyük odadaki ceyranı aç	
8	oğlum salondaki elektriği yak	oğlum salondaki ışığı çak	oğlum büyük odanın ceyranını yak	



9	oğlum salon elektriğini aç	oğlum salonun ışığını çak	oğlum büyük odadaki ceyranı yak
10	oğlum salonun elektriğini aç	oğlum salon elektriğini çak	oğlum büyük odanın ceyranını aydınlat
11	oğlum salondaki elektriği aç	oğlum salonun elektriğini çak	oğlum büyük odadaki ceyranı aydınlat
12	oğlum salon elektriğini on	oğlum salondaki elektriği çak	oğlum büyük odanın ceyranını on
13	oğlum salonun elektriğini on	oğlum salon ceyranını çak	oğlum büyük odadaki ceyranı on
14	oğlum salondaki elektriği on	oğlum salonun ceyranını çak	oğlum büyük odanın ceyranını aç
15	oğlum salon ceyranını aç	oğlum salondaki ceyranı çak	oğlum büyük odadaki ceyranı aç
16	oğlum salonun ceyranını aç	oğlum salon ceyranını çak	oğlum büyük odanın ceyranını yak
17	oğlum salondaki ceyranı aç	oğlum salonun ceyranını çak	oğlum büyük odadaki ceyranı yak
18	oğlum salon ceyranını yak	oğlum salondaki ceyranı çak	oğlum büyük odanın ceyranını aydınlat
19	oğlum salonun ceyranını yak	oğlum salon şavkını aç	oğlum büyük odadaki ceyranı aydınlat
20	oğlum salondaki ceyranı yak	oğlum salonun şavkını aç	oğlum büyük odanın ceyranını on
21	oğlum salon ceyranını aydınlat	oğlum salondaki şavkı aç	oğlum büyük odadaki ceyranı on
22	oğlum salonun ceyranını aydınlat	oğlum salon şavkını yak	oğlum büyük odanın ceyranını aç
23	oğlum salondaki ceyranı aydınlat	oğlum salonun şavkını yak	oğlum büyük odadaki ceyranı aç
24	oğlum salon ceyranını on	oğlum salondaki şavkı yak	oğlum büyük odanın ceyranını yak
25	oğlum salonun ceyranını on	oğlum salon şavkını aydınlat	oğlum büyük odadaki ceyranı yak
26	oğlum salondaki ceyranı on	oğlum salonun şavkını aydınlat	oğlum büyük odanın ceyranını aydınlat
27	oğlum salon ceyranını aç	oğlum salondaki şavkı aydınlat	oğlum büyük odadaki ceyranı aydınlat
28	oğlum salonun ceyranını aç	oğlum salon şavkını on	oğlum büyük odanın ceyranını on
29	oğlum salondaki ceyranı aç	oğlum salonun şavkını on	oğlum büyük odadaki ceyranı on
30	oğlum salon ceyranını yak	oğlum salondaki şavkı on	oğlum büyük odanın ceyranını çak
31	oğlum salonun ceyranını yak	oğlum salon şavkını çak	oğlum büyük odadaki ceyranı çak
32	oğlum salondaki ceyranı yak	oğlum salonun şavkını çak	oğlum büyük odanın lambasını çak
33	oğlum salon ceyranını aydınlat	oğlum salondaki şavkı çak	oğlum büyük odadaki lambayı çak
34	oğlum salonun ceyranını aydınlat	oğlum büyük odanın lambasını yak	oğlum büyük odadaki ışığı çak
35	oğlum salondaki ceyranı aydınlat	oğlum büyük odadaki lambayı yak	oğlum büyük odanın ışığını çak
36	oğlum salon ceyranını on	oğlum büyük odanın ışığını yak	oğlum büyük odanın elektriğini çak
37	oğlum salonun ceyranını on	oğlum büyük odadaki ışığı yak	oğlum büyük odadaki elektriği çak
38	oğlum salondaki ceyranı on	oğlum büyük odanın lambasını aç	oğlum büyük odanın ceyranını çak
39	oğlum salon ceyranını aç	oğlum büyük odadaki lambayı aç	oğlum büyük odadaki ceyranı çak
40	oğlum salonun ceyranını aç	oğlum büyük odanın ışığını aç	oğlum büyük odanın ceyranını çak
41	oğlum salondaki ceyranı aç	oğlum büyük odadaki ışığı aç	oğlum büyük odadaki ceyranı çak
42	oğlum salon ceyranını yak	oğlum büyük odanın lambasını aydınlat	oğlum büyük odanın şavkını aç
43	oğlum salonun ceyranını yak	oğlum büyük odadaki lambayı aydınlat	oğlum büyük odadaki şavkı aç
44	oğlum salondaki ceyranı yak	oğlum büyük odanın ışığını aydınlat	oğlum büyük odanın şavkını yak
45	oğlum salon ceyranını aydınlat	oğlum büyük odadaki ışığı aydınlat	oğlum büyük odadaki şavkı yak
46	oğlum salonun ceyranını aydınlat	oğlum büyük odanın lambasını on	oğlum büyük odanın şavkını aydınlat

47	oğlum salondaki cereyanı aydınlat	oğlum büyük odadaki lambayı on	oğlum büyük odadaki şavkı aydınlat
48	oğlum salon cereyanını on	oğlum büyük odanın ışığını on	oğlum büyük odanın şavkını on
49	oğlum salonun cereyanını on	oğlum büyük odadaki ışığı on	oğlum büyük odadaki şavkı on
50	oğlum salondaki cereyanı on	oğlum büyük odanın elektriğini aç	oğlum büyük odanın şavkını çak

Örnek olması için **Toplam 350** komut yazılmıştır, fakat yeni komutlar eklenerek sayı daha da arttırılabilir.