

KARABÜK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
UZAKTAN EĞİTİM BÖLÜMÜ



UZAKTAN KONTROLLÜ BEŞİK VE AKILLI BEŞİK

BİTİRME PROJESİ 2

MUHAMMET MUSTAFA GÜL

2019-2020 BAHAR

İçindekiler

1.Genel Bilgiler.....	4
1.1. Giriş.....	4
2. Kullanılan Malzemeler.....	5
2.1 Kontrolcü Seçimi	5
2.1.1. Arduino Uno	5
2.1.2. Microdenetleyici ATmega328P Pin Yapısı	7
2.3 Komponentler.....	8
2.4. Çıkış Aygıtları.....	10
2.5. Diğer Kullanılan Malzemeler	12
2.6. İletişim.....	13
2.7. Mobil Uygulama	14
3. Yapılan Çalışmalar	16
3.1 Android Uygulama	16
3.2. Tasarım.....	17
3.2.1 Blok Diyagramı.....	17
3.2.2. Fritzing Tasarımı	18
3.2.3. Programlama Akış Diyagramları	19
4. Arduino Kodları	22
5. Android Uygulama	26
5.1. Android Uygulama Kod Kısmı	28
Kaynakça	32

ÖZET

Projemiz android telefona yazdığımız uygulama sayesinde bluetooth ile haberleşmeyi sağlıyor. Bebeğimiz ağladığında ya da hareket ettiğinde bunu sensörler yardımı ile algılayıp sallanmaya başlıyor. Havadaki nem ve sıcaklık değerlerini okuyarak uygulamamıza bildirimde bulunuyor. Sensörlerden veri okunmadan da manuel şekilde sallama işlemi gerçekleştiriliyor. Uyandırma seçeneği kullanılarak bebeğimiz istediğimiz zaman butonu aktif ederek uyandırabiliriz. Beşiğimizde gece lambamızda vardır. Yine uygulamadaki buton aktif edilerek kontrol sağlanır.

Tüm aşamalardaki kodlama işlemleri arduino IDE ve MIT APP İNVERTER kullanılarak belirtilen elemanların istenilen şekilde görev yapmaları sağlanmıştır.

1.Genel Bilgiler

1.1. Giriş

Projenin konusu Uzaktan Kontrollü Beşik ve Akıllı Beşik 'tir. Üç adımda olduğu söylenebilir. İlk adım breadboard üzerine monte edilen Bluetooth, ses, buzzer, hareket, nem ve sıcaklık sensörleri denemek amacıyla çalıştırılmıştır.

Projenin ikinci aşamasında android uygulama kısmı MIT APP INVERTER kullanılarak tasarımı yapılmış ve denenmiştir. Projemiz sağlıklı çalışması sağlanmıştır.

Projemizin son aşaması kontrol ünitemizin beşiğe montaj kısmını oluşturmaktadır.

2. Kullanılan Malzemeler

2.1 Kontrolcü Seçimi

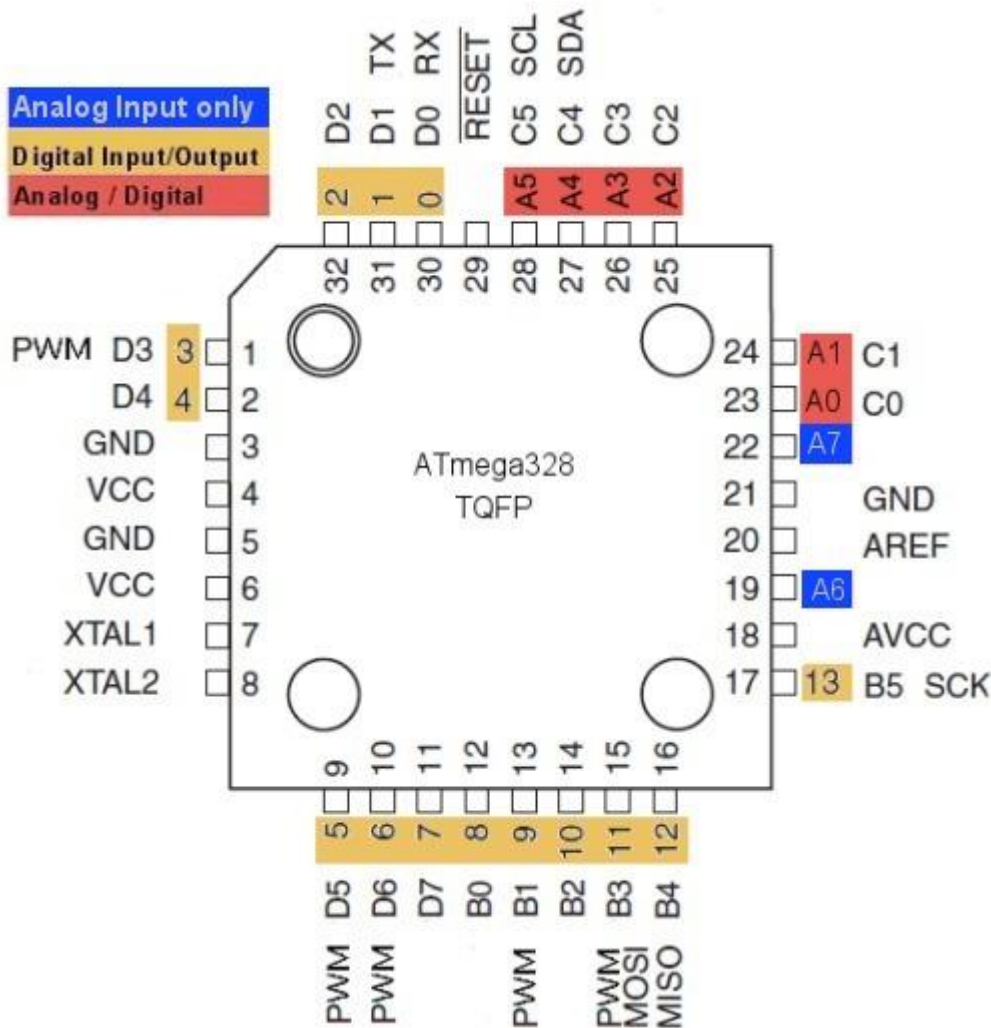
Arduino, bilgisayarınız aracılığıyla programlayarak çeşitli elektronik projeler yapabileceğiniz bir mikro kontrolcü platformudur. Yeterli sayıda digital giriş ve çıkışa sahip olması başlıca seçme sebeplerimden oldu.

Boards	Microcontroller	Operating Voltage/s (V)	Digital I/O Pins	PWM Enabled Pins	Analog I/O Pins	DC per I/O (mA)	Flash Memory (KB)	SRAM (KB)	EEPROM (KB)	Clock (MHz)	Length (mm)	Width (mm)	Cable	Native Network Support
Uno	ATmega328	5	14	6	6	20	32	2	1	16	68.6	53.4	USB A-B	None
Leonardo	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	micro-USB	None
Micro	ATmega32u4	5	20	7	12	40	32	2.5	1	16	48	18	micro-USB	None
Nano	ATmega328	5	22	6	8	40	32	2	0.51	16	45	18	mini-B USB	None
Mini	ATmega328	5	14		6	20	32	2	1	16	30	18	USB-Serial	None

Şekil 1

2.1.1. Arduino Uno

Arduino, bilgisayarınız aracılığıyla programlayarak çeşitli elektronik projeler yapabileceğiniz bir mikro kontrolcü platformudur. Arduino ile robotik projeler, akıllı ev sistemleri, müzik aletleri gibi aklınıza gelebilecek neredeyse tüm elektronik projeleri gerçekleştirmeniz mümkündür. Arduino üzerindeki elektronik bağlantıları giriş olarak kullanarak butonlar, uzaktan kumandalar ve farklı çeşitlerde sensörler bağlayabilir; bu girişlerden elde edilen verilere göre elektronik çıkışlar elde ederek motorların, robotik mekanizmaların, rölelerin, hoparlörlerin, ışık ve göstergelerin ve hatta Ethernet veya Wi-Fi üzerinden internetteki herhangi bir verinin elde ettiğiniz giriş değerlerine göre değişimini yazdığınız program aracılığıyla kontrol edebilirsiniz. ATmega328 mikro kontrolcü bulunmaktadır.



Şekil 3

Arduino Uno’ da SMD ve DİP model olarak 2 farklı çipli model mevcuttur. Bu modellerin farkları DİP soketli modelde mikrodenetleyiciyi sökebilir dışarıda kullanabilirsiniz. Bunun yanı sıra DİP soket modellerinde dijital pinlerin üzerinde sol tarafta 6 adet pin bulunur. Bu pinler SPI haberleşmede kullanılan MOSI(Master Out Slave In), MISO (Master In Slave Out), SCK(Serial Clock), Reset pinleridir. Bu pinler digital 11, 12, 13 pinlerinden farklı olarak kullanılabilirler. Kalan 2 pin ise 5 volt ile GND’ dir.

Digital Pinler : 0 veya 5 volt giriş, çıkış (input,output) verebilen pinlerdir. Bu pinlerden bazılarının ekstra özellikleri vardır. Mesala D13 pini Serial Clock pini olarak kullanılır yani SPI haberleşmede 2 cihazın saat hızlarını eşitleyip senkronizasyon sağlamak amaçlı kullanılır. Bunun yanı sıra PWM (pulse with modulation) dalga modülasyonu olarak kullanılabilir yani normalde digital pinler 0 veya 5 volt gerilim verebilirken PWM pinleri analogWrite komutu ile 0,019 volt

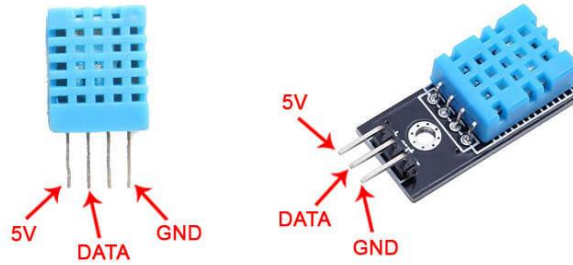
hassasiyetle 0-5 volt arası gerilim verebilirler. Bu pinler hassas motor sürme gibi amaçlarla kullanılabilir.

PWM Pinleri : Yukarıda digital pin tanımında da belirttiğim gibi 0,5 volt arasında 0,019 volt hassasiyetle voltaj alabilirsiniz. Bu pinler digital pinlerden ~işaretleli olanlardır. Uno için; 3,5,6,9,10,11 numaralı digital pinlerdir.

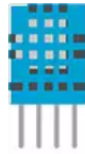
Analog Pinler : A0-A5 Arasındaki pinlerdir, analog giriş çıkış amaçlı kullanılırlar, Analog olarak 0-5 volt arası gerilimi 5/1024 hassasiyet ile alıp verebilir. IN olarak kullanıldığında cihazların verdiği gerilimi ölçebilirsiniz.

2.3 Komponentler

➤ Isı ve Nem Sensörü [DHT11]



Şekil 4



DHT11

Çalışma Voltajı	3 – 5V
Akım mAh Max.	2.5mAh
Nem	20-80% / 5%
Sıcaklık	0-50°C / $\pm 2^\circ\text{C}$
Ölçüm Hızı	1 Hz (1 Saniyede Yenileme)
Fiziki Ölçüler	15.5mm x 12mm x 5.5mm
Avantajlar	Ucuz

DHT11 sıcaklık aralığı +2 derece hassasiyetle 0 ° C ile 50 ° C arasındadır. DHT11 nem aralığı ise% 5 ile% 20 doğrulukla ölçüm yapmaktadır. Fiyat uygunluğu ve sadece +2 derece hassasiyet farkı ile ölçüm yapması bizim için tercih sebebi olmuştur.

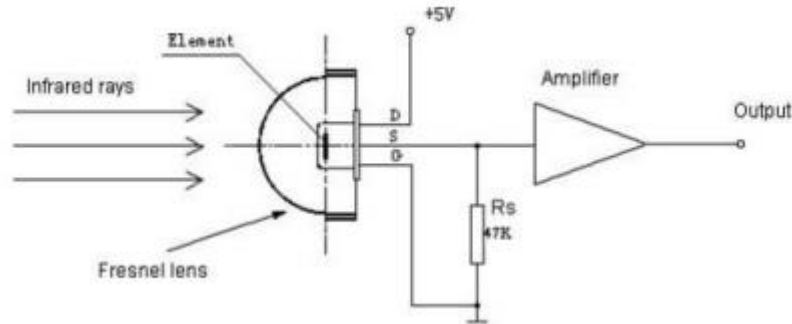
Şekil 5

➤ Hareket Sensörü [PIR]



Şekil 6

PIR ismi Passive Infra-Red kelimelerinin baş harflerinden gelmektedir. Bu da bu sensörün kızılötesi dalgalarla çalıştığı anlamına gelir. Hareket sensörü ortamdaki sıcaklık ve kızılötesi dalga değişimlerini algılamaya yarayan sensörlerdir. Yapılarında bir fresnel lens bulunur. Bu lens sayesinde ortamdaki nesnelerden gelen ışınlar sensörün odaklanmasını sağlar. Ortamda bir dalga değişimi olduğunda sensör algılama işlemi gerçekleştirir.



Şekil 7

Özellikleri:

- Çalışma Voltajı: 5-12V
- Lojik Sinyal Çıkış Seviyesi: 3,3V
- Algılama Alanı: 3-5 metre
- Algılama Açısı: 140°
- Bekleme Süresi: 5-200 saniye
- Ürün Boyutları: 33x25x24 mm

➤ Ses Sensörü



Şekil 8

Ses sensörü kartı, üzerinde mikrofon bulunan ve ortamdaki ses seviyesine göre dijital çıkış veren bir karttır. Bir el çırpmasını veya bir kornayı; ani ve yüksek şekilde çıkan bir sesi veya ortam gürültüsünü sürekli olarak okuyabileceğiniz kart üzerinde potansiyometre yer almaktadır.

Bu sayede ortam gürültüsünden sıyrılarak anlık gelen ses çıkışını elde edebilirsiniz.

2.4. Çıkış Aygıtları

➤ Buzzer



Şekil 9

Buzzer Arduino devrelerinde ses elde etmek amacıyla kullanılan bir ekipmandır. Buzzerı günlük hayatta bize uyarı veren birçok yerde görebiliriz. Örneğin araçlarda ki park sistemlerinde, alarmlarda vb. alanlarda görme şansımız vardır. Kullanımı oldukça kolay olduğu için çok kullanışlı bir ekipmandır. Sinyal ucuna dijital çıkış vermemiz yeterli olacaktır ses çıkışı için.

➤ Role [Tek Kanallı]



Şekil 10

Üzerinde bir adet röle bulunan bir modüldür. İki adet gösterge ledi bulunur. Kart üzerindeki röleyi kontrol etmeniz için 5 volt sinyal giriş pini bulunmaktadır. Buradan vereceğiniz sinyal ile rölenin anahtar konumunu ayarlayabilirsiniz. Böylelikle üzerinden hem ac hem dc akım geçirebilir. Mikrodenetleyeciden tetik sinyali sırasında 20mA"lık bir akım çekmektedir. Röle basit bir açma/kapama anahtarıdır. Giriş voltajı 5 Volt olduğunda çalışır, giriş voltajı 0 Volt olduğunda kapanır.

➤ Servo Motor



Şekil 11

Servo motor 0 ila 180 derece arasında 1 derece hassasiyetle dönebilen motor çeşididir. Tam tur atamaz. Genellikle robot kol gibi tam tur dönmesine gerek olmayan, hassas açılı yerlerde kullanılır. Servo motor içerisinde bir adet DC motor bulunur. DC motorun ucuna bağlı dişli sisteminin yardımıyla servo mili daha fazla yük kaldırabilmektedir. Bu işlem sırasında servonun dönüş

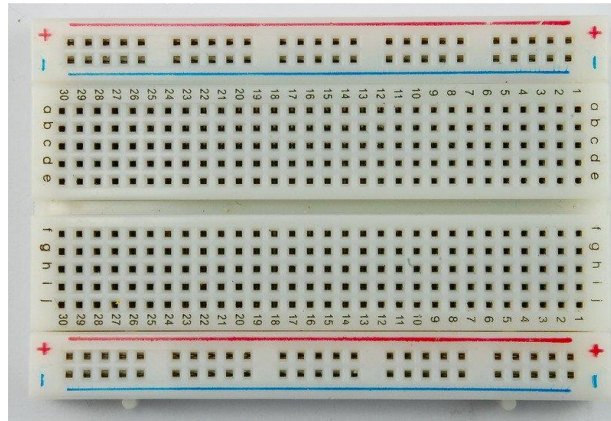
hızı da yavaşlamış olur. Kullanılan dişli sistemine göre servo motorların kaldırabileceği yük değişir.

Servoların kaldırabileceği yük tork gücü üzerinden ifade edilir. Servo motorların torku, motor miline bağlı 1 cm uzunluğundaki çubuğun kaldırabileceği maksimum yük olarak tarif edilir. Piyasada bulunan servolar genellikle 1,4 kgf.cm torka sahiptir. Bu da demek oluyor ki, motor milinize bağlı 1 cm uzunluğunda bir çubuk varsa ve bu çubuğun ucuna bağlı yük 1,4 kilogramdan fazlaysa motorunuzun gücü mili döndürmeye yetmez. Eğer çubuğun uzunluğu 10 cm ise en fazla 140 gram kaldırabilirsiniz.

Servo motorun üç adet bağlantı kablosu bulunmaktadır. Bu kablolar genellikle kırmızı, turuncu (bazen sarı) ve siyah (bazen kahverengi) olmaktadır. Bu renkler kabloların görevini göstermektedir. Kırmızı renk besleme (genellikle 5 volt) bağlantısını, siyah veya kahverengi renk de toprak bağlantısını göstermektedir. Geriye kalan turuncu kablo ise motorun açısını belirleyecek olan veri bağlantısıdır. Motorun dönüş açısının belirlenmesi için veri hattı üzerinden PWM adı verilen özel kare dalga sinyalleri yollanmaktadır. PWM sinyali belirli bir süre 5 volt, belirli bir süre 0 volt düzeyinde verilen gerilimdir.

2.5. Diğer Kullanılan Malzemeler

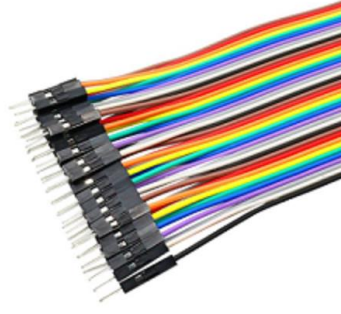
➤ BreadBoard



Şekil 12

Üzerinde devrelerimizi test ettiğimiz araçtır. Kurduğumuz devreleri birbirlerine lehimlemeden kolaylıkla test etmemizi sağlar. Tasarladığımız devreleri baskı devre veya delikli plakette üzerine aktarmadan önce denememize olanak sağlar. Bu sayede devre bağlantılarını kontrol ederek bir hata olup olmadığını gözlemlemiş oluruz. Devreleri tak-çıkartma şeklinde kurabildiğimiz için kullandığımız elektronik bileşenleri başka projelerde tekrar kullanma imkânı verir.

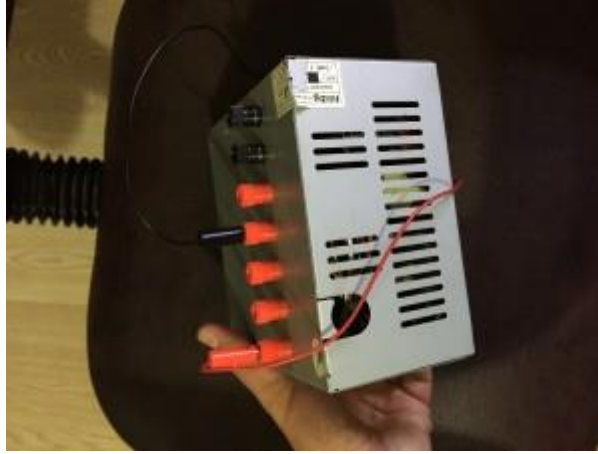
➤ Jumper Kablolar



Şekil 13

Breadborda bir eleman bağlarken veya Arduino ya bir modül bağlarken kullanılabilir. Üç tipi vardır; Dişi-Erkek, Dişi-Dişi, Erkek-Erkek

➤ Güç Kaynağı

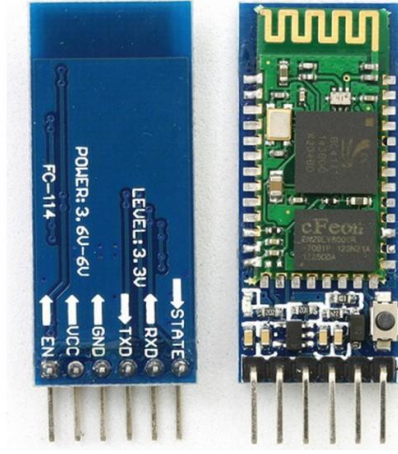


Şekil 14

Güç kaynakları sistemlerin ihtiyacı olan enerjiyi karşılamak amacıyla kullanılan araçlardır. Ben boşta olan Bilgisayar güç kaynağımı kullanarak resimdeki gibi çıkışlar elde ettim. Ve projem için gereken voltajı alabildim.

2.6. İletişim

➤ HC05 [Bluetooth Modülü]



Şekil 15

Bluetooth modülü arduino ile uzaktan yönetilebilir projeler üretmenize olanak sağlamaktadır. Bluetooth modülleri ile evinizdeki herhangi bir cihazı cep telefonu veya tablet ile kontrol edebilir, akıllı cihazlarla kontrol edebileceğiniz araçlar tasarlayabilirsiniz.

HC-05 Bluetooth modülünü bir diğer Bluetooth modülü olan HC-06 dan ayıran en önemli özelliği hc-05 modülün hem alıcı hem de verici modül olarak kullanılabilmesidir. Yani elinizdeki HC-05 Bluetooth modülünü başka Bluetooth modüllerine veri göndermek için kullanabilirsiniz. Bu özelliği ile akıllı cihazlara ihtiyaç olmadan iki arduino arasında Bluetooth üzerinden veri aktarımı yapabilirsiniz.

Özellikler:

- Çalışma Gerilimi: 3.3V
- Bluetooth Protokolü: Bluetooth 2.0+Edr(Gelişmiş Veri Hızı)
- 2.4Ghz Haberleşme Frekansı
- Hassasiyet: ≤ -80 Dbm
- Çıkış Gücü: $\leq +4$ Dbm
- Asenkron Hız: 2.1 Mbps/160 Kbps
- Senkron Hız: 1 Mbps/1 Mbps
- Güvenlik: Kimlik Doğrulama Ve Şifreleme
- Akım: 50 Ma
- Boyutları: 43X16X7Mm

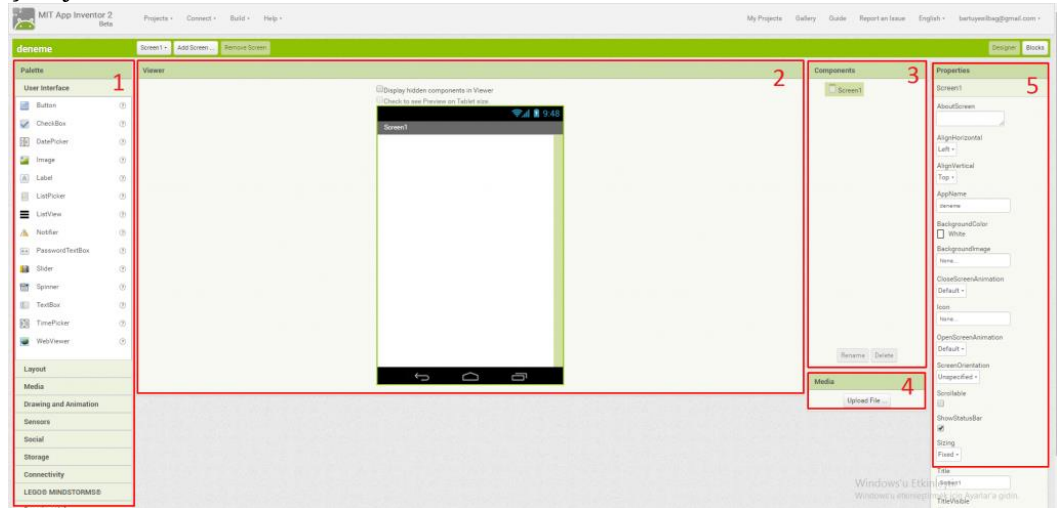
2.7. Mobil Uygulama

➤ MIT APP INVERTER



Şekil 16

App Inventor, Massachusetts Institute of Technology(MIT) tarafından geliştirilen, programlama konusunda tecrübesiz olmayan kimselerin kullanması için yapılmış olan ve Android uygulaması geliştirmesine olanak sağlayan bir uygulama geliştirme platformudur. En önemli özelliği drag and drop yani sürükle ve bırak şeklinde programlanabilmesidir. Şuan için sol üst köşeden Start New Project butonuna bastıktan sonra Şekil-17 karşımıza çıkıyor.



Şekil 17

1- Palette: Uygulamaya koyabileceğimiz görsel ya da işlevsel her şeyin bulunduğu kısım. Örneğin buton, onay kutusu, tarayıcı, bluetooth

2-Viewer: Uygulamamızın görüldüğü kısımdır. Uygulamayı bitirdiğimizde, görsel arayüz bu ekrandaki gibi gözükür

3-Components: Uygulamamızda kullanacağımız nesnelerin ismini görüp değiştirebileceğimiz kısım

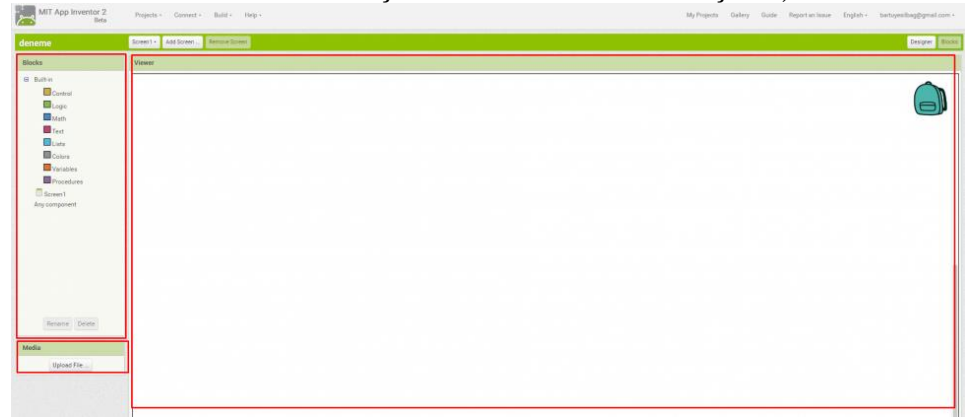
4-Media: Kullanacağımız ses, fotoğraf gibi dosyaların yükleneceği yer.

5-Properties: Eklediğimiz nesnelerin ayarları ile bu kısımdan oynayabiliriz. Örneğin bir buton eklediğimizde, butonun üzerindeki yazıyı, boyutunu veya rengini buradan değiştirebiliriz.

Kısımları bu şekilde kısaca tanıttıktan sonra bir diğer önemli bölüme geçiyoruz.

Sağ üst kısımda designer/block isminde iki buton bulunmaktadır. Bu butonlardan designer kısmı şuna kadar gördüğümüz kısımken block kısmı ise parçaları birleştirerek, kodlama yaptığımız kısımdır. Buradan, kontrol yapılarını, mantıksal işlemleri, matematiksel işlemleri yapabiliriz.

Block kısmına tıklarsak Şekil-18 ekran biri karşılar;



3. Yapılan Çalışmalar

3.1 Android Uygulama

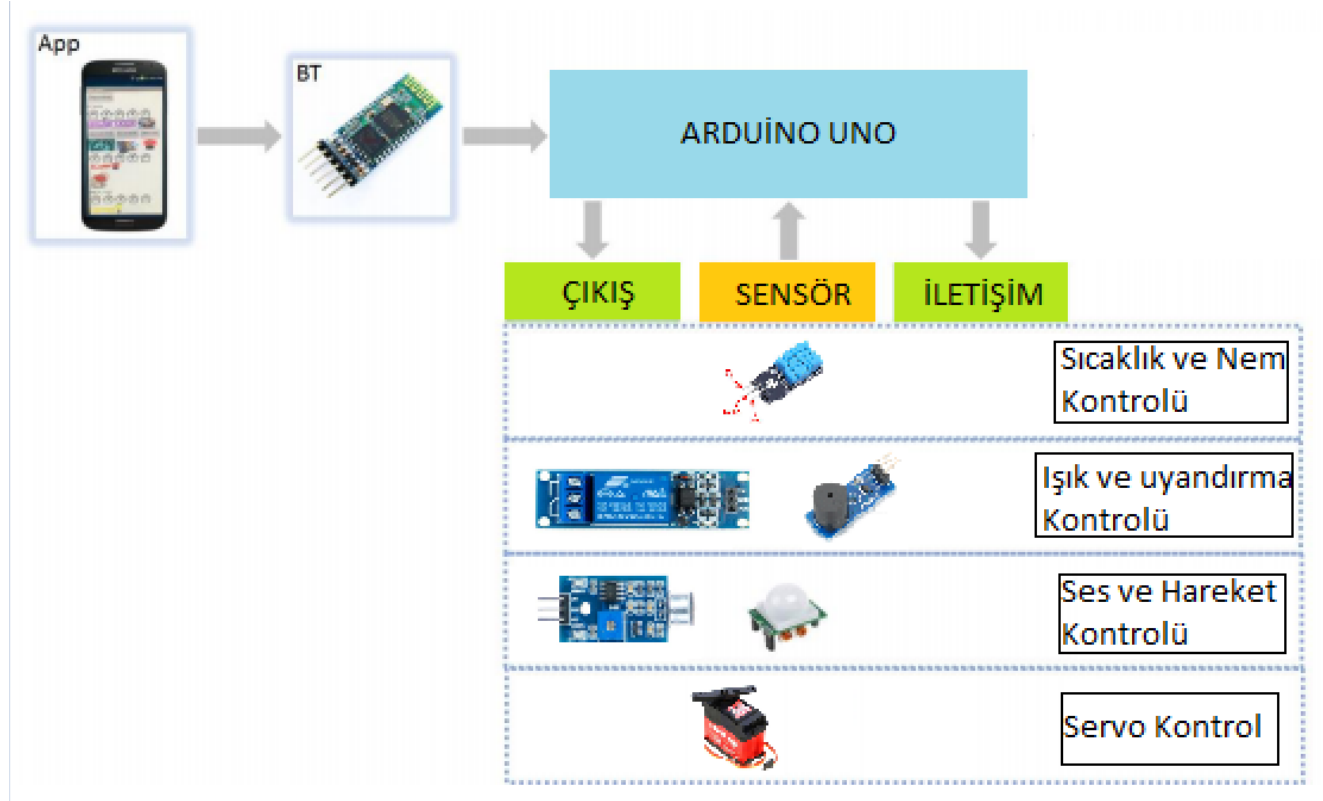
Android uygulamamız Bluetooth kullanarak Arduino ile haberleşme sağlamaktadır. Çeşitli komutlar gönderip ve almaktadır. Çocuğumuz için ortamın sıcaklık ve nem değerleri önemli olduğu için okuma işlemi yapıp android uygulamamızda bir satırda bu değerleri göstermektedir. Uygulamamızda motor hız kontrolü ve üç adet mod tasarlanmıştır.

- **Manuel Mod**
Bu Modda belirlenen bir hızda sallama işlemi gerçekleştirilir.
- **Ağlama Modu**
Bu Modda ses sensöründen gelen veri eğer eşik değerini aşıyorsa sallama işlemi gerçekleştirir.
- **Sessiz Mod**

Bu Modda ise ses sensöründen veya hareket sensöründen gelen bir veri varsa sallama işlemi gerçekleşir.

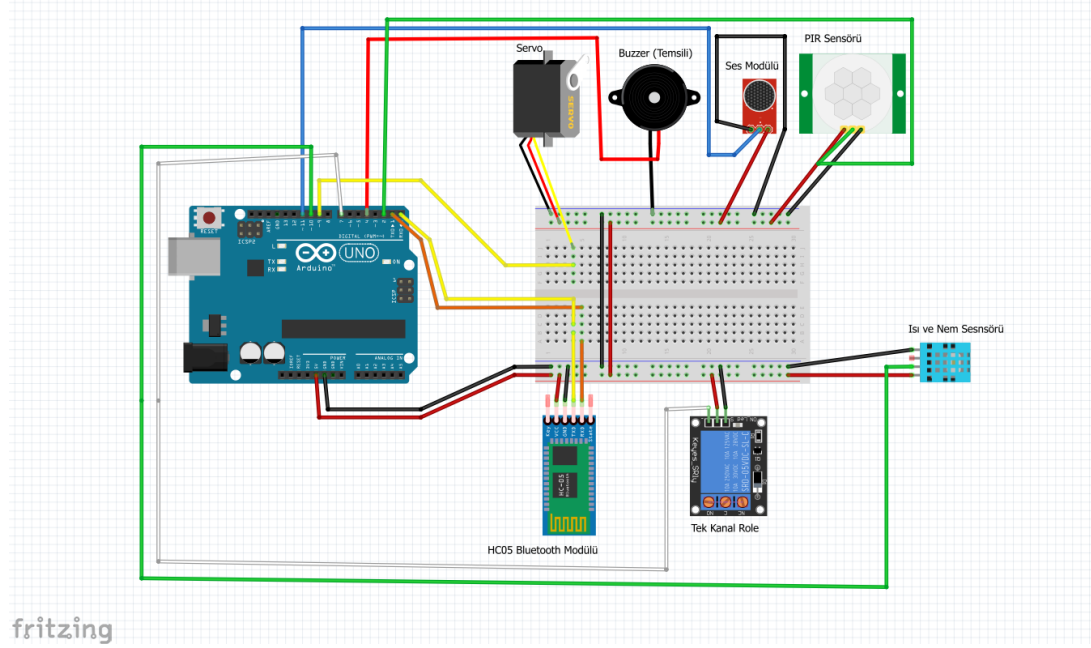
3.2. Tasarım

3.2.1 Blok Diyagramı



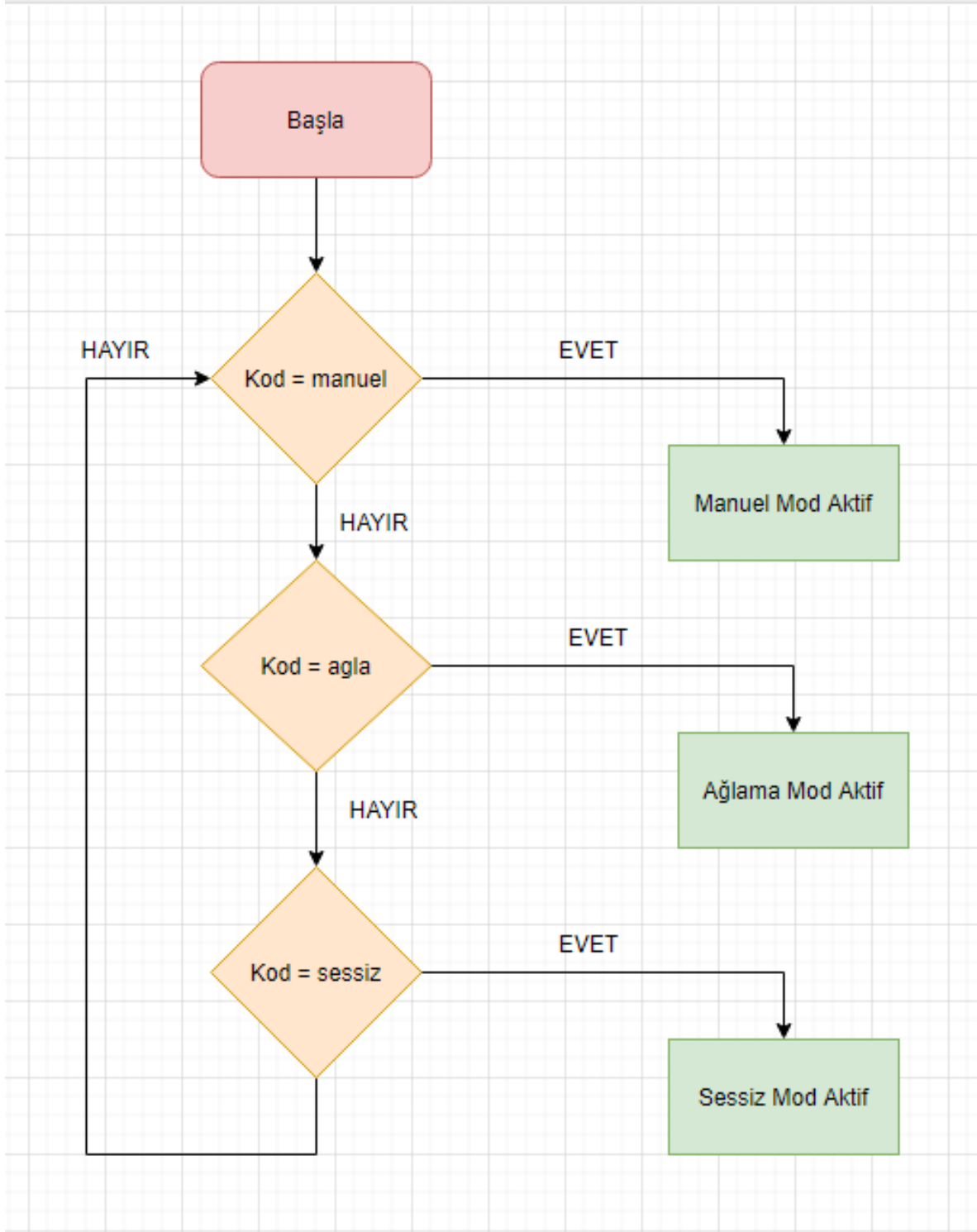
Şekil 18

3.2.2. Fritzing Tasarımı



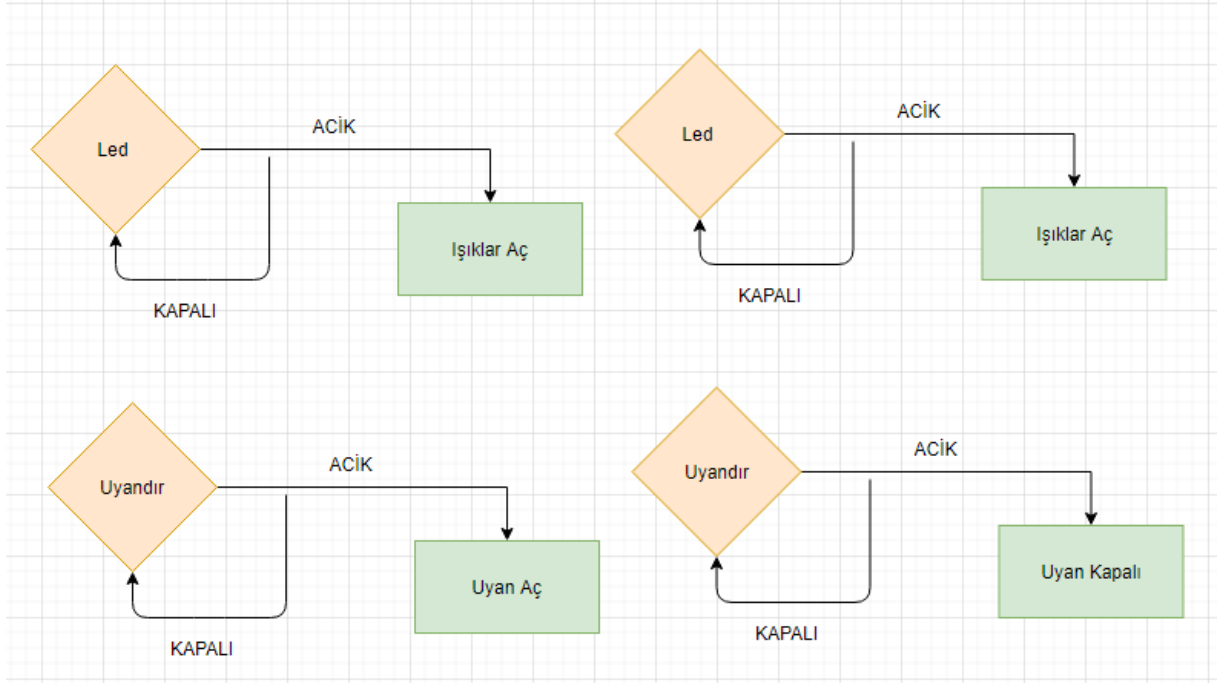
3.2.3. Programlama Akış Diyagramları

1.Mod Seçimi

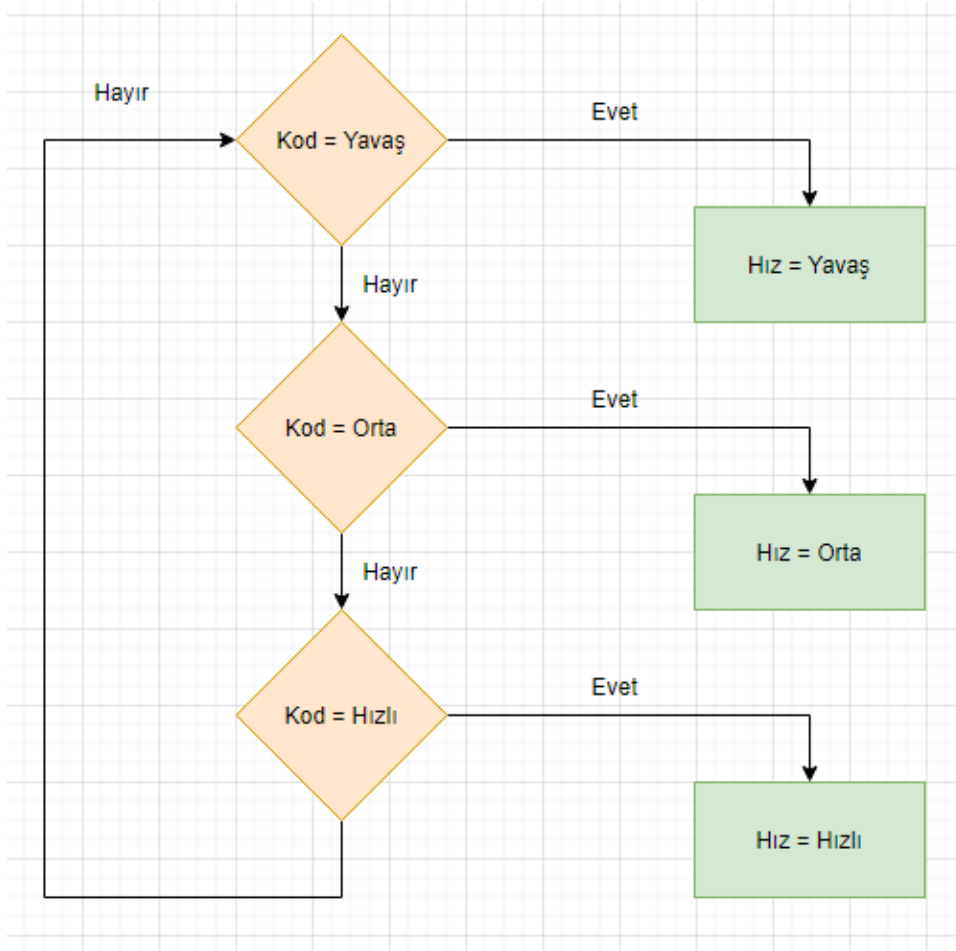


Şekil 19

2. Kontrol edilen diğer aygıtların şemaları



Şekil 20



Şekil 21

4. Arduino Kodları

```
ebesik
#include "DHT.h" // ısı ve nem sensörü kütüphanesi
#include <Servo.h> // servo kütüphanesi
#define DHTPIN 10 // Isı ve Nem sensörü için tanımladığımız pin
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

Servo myServo;

String veri = ""; // bluetooth tan gelen veri

const int ses = 11; // ses sensöründen gelecek olan veri okumak için
const int pir = 2; // hareket sensöründen gelen veriyi okumak için
const int buzzer = 4; // uyandırmak için buzzer pini
const int role = 7; // role kartı için
int hiz=150; // motor hareketi için ilk değer atadık
String hizdeger; // okunan veriyi değişkene atadık
int seviye=0; // ses sensöründen gelen veriyi kayıt eden değişken
int hareket=0; // hareket sensöründen gelen veriyi kayıt eden sensör
bool veridurumu; // bluetooth okuma yapılırken veri geldiyse eğer veridurumunu değiştiren değişken
String verieski; // bluetooth okuma yapıldıktan sonra bu değişkene atılır ve kıyaslama yapılır.
//unsigned yenisizaman,eskizaman;

void setup() {
  Serial.begin(9600); // seri haberleşmeyi başlattık
  pinMode(ses,INPUT); // ses sensörünün bağlı olduğu pin giriş olarak ayarlandı
  pinMode(pir,INPUT); // hareket sensörünün bağlı olduğu pin giriş olarak ayarlandı
  pinMode(buzzer,OUTPUT); // buzzerın bağlı olduğu pin çıkış olarak ayarlandı
  pinMode(role,OUTPUT); // rolenin olduğu pin çıkış olarak ayarlandı
  myServo.attach(9); // servo pin aktif edildi
}
```

Şekil 22

```

void loop() {
    oku(); // bluetooth verisi okumayı sağlıyor
    delay(50);
    isioku(); // ısı ve nem değerlerini okuyup android uygulamaya yolluyor
    hizoku(); // android uygulamadan gelen hız değerleri okunuyor

    if(veri == "manuel") // mod seçimi sadece sallama işlemi yapılıyor
    {
        mansal(); // sadece sallama işlemi
    }
    else
    if(veri == "agla") // sese duyarlı moda geçiliyor
    {
        agsal(); // ses algılandığında aktif oluyor
    }
    else
    if(veri == "sessiz") // ses ve hareket duyarlı mod
    {
        sezsas(); // ses veya hareket varsa aktif oluyor
    }

    else if(veri == "uyanac"){ // uyandırmak istendiğinde aktif yada pasif olabiliyor
        digitalWrite(buzzer,HIGH);
    }
    else if(veri == "uyankap"){
        digitalWrite(buzzer,LOW);
    }
    //uyan ac

    else if(veri == "isikac"){ // ışık açılmak istendiğinde aktif yada pasif olabiliyor
        digitalWrite(role,HIGH);
    }
    else if(veri == "isikap"){
        digitalWrite(role,LOW);
    }
}

```

Şekil 23

```

}
void isioku(){ // ısı ve nem okuma
float t = dht.readTemperature(); // sıcaklık değeri okunup değişkene atılıyor
float h = dht.readHumidity(); // nem değeri hesaplanıp yüzde olarak değişkene atılıyor
Serial.print("Nem Oranı:% "); // uygulamada yazdırmak için
Serial.print((int)h); // uygulama okunan veri gönderiliyor
Serial.print("|"); // aandroid kısımda gelen veriyi ayırabilmek için bu işareti kullandık
Serial.print("Oda Sıcaklığı: ");
Serial.print((int)t);
Serial.print(" C");
Serial.print("|");
for(int i=0;i<50;i++){ // bu döngü bize yukarda yapılan işlem esnasında eğer veri gelirse kaçırmamak
    delay(10); // için hem bekleme yapıyor hemde verinin gelip gelmediğinin kontrolünü sağlıyor.
    oku(); // bluetooth veri okuma
    if(veridurumu == 1) // yeni veri kontrolü
        break;
}
}
void oku() // bluetooth veri okuma
{
    while(Serial.available() > 0) // veri gelirse oku
    {
        veri = Serial.readString(); // veri değişkenine at
    }
    if(verieski != veri){ // eski ve yeni veri kıyaslama
        veridurumu=1;
        verieski = veri;
    }else
    {
        veridurumu=0;
    }
}
void hizoku(){ // gelen hız değerleri atamaları yapılıyor
    if(veri == "yavas")
        hiz = 10;
    else if(veri == "orta"){
        hiz = 100; }
    else if(veri == "hizli")
        hiz = 180;
    else if(veri == "stop") // beşik resmine basıldığında durdurma işlemi yapılıyor
        {hiz = 0;
}
}

```

Şekil 24


```

/*-----manuel sallama-----*/
int mansal(){ // beşiğimizin hareket fonk.
    hizoku(); // hız değeri okunuyor.
    int i;
    if(veridurumu == 1)
        return;
    myServo.write(0);
    for(i=0;i<5;i++){ // servo hareketini tamamlayabilmesi için geçen zaman içerisinde veri gelip gelmedi
        delay(10); // kontrolü sağlanıyor
        if(veridurumu == 1)
            break;
        oku();
    }
    isioku();
    if(veridurumu == 1)
        return;
    myServo.write(hiz);
    for(i=0;i<5;i++){
        delay(10);
        if(veridurumu == 1)
            break;
        oku();
    }
}

/*-----manuel sallama bitis-----*/
/*-----agla ac-----*/
void agsal(){ // aglama modu aktif olduğunda çalışıyor
    oku();
    seviye = digitalRead(ses); // ses algılanırsa değişkene atamıyor
    if(seviye == LOW){ // ses varsa kod çalışıyor
        mansal();
    }
    else{
        myServo.write(0);
        delay(15);
    }
}

/*-----agla ac bitis-----*/

```

Şekil 25

```

/*-----sessiz mod -----*/
void sezsal(){ // sessiz modu
    oku();
    hareket = digitalRead(pir); // hareket algılandığında değeri atanıyor
    seviye = digitalRead(ses); // ses algılandığında değeri atanıyor
    if(seviye == LOW || hareket == HIGH){ // ses veya hareket varsa salla yoksa sallama
        mansal();
    }
    else
    {
        myServo.write(0);
        delay(15);
    }
}

/*-----sessiz mod bitis-----*/

```

Şekil 26

5. Android Uygulama

Mıt App İnvertor kullanarak tasarlanmıştır. Bluetooth ikonu üzerine tıklanarak aygıt seçilir ve bağlantı kurulur. Bağlantı yapılırsa butonun altında bağlantı yapıldı yazı belirir. E-beşik yazan fotoğrafa tıklandığında ise bütün basılı butonlar sıfırlanır ve arduino ya sallanmayı durdurmasını söyler.

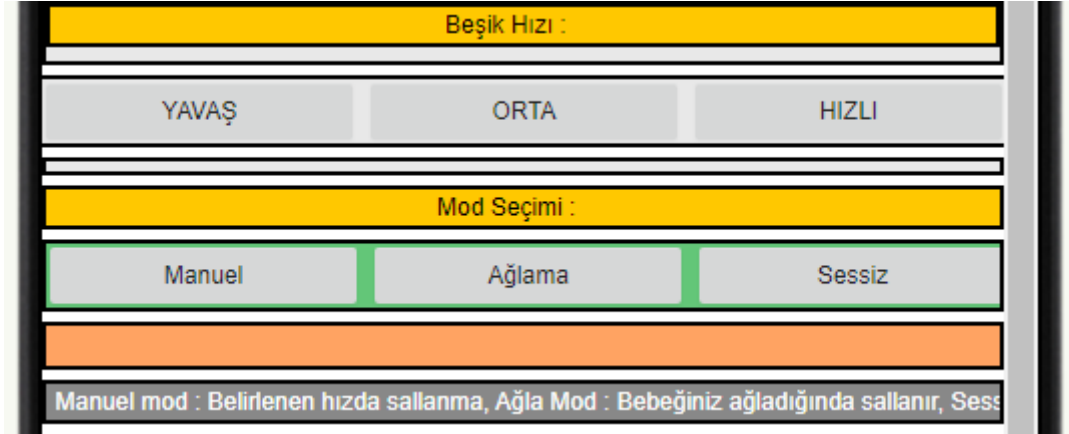


Şekil 27

Bu kısımda Beşik hızı ve mod seçim işlemi yapılır. Beşik hızı seçiminde sadece bir seçme işlemi yapılabilir. Basılan buton pasif olur diğer iki seçenek aktif şekilde kalır.

Mod seçim işlemi de hız seçim işlemi gibi butonlar aktif yada pasif olur. Ve bir alt satırında sıcaklık ve nem oranı bilgisi kullanıcıyla paylaşılır.

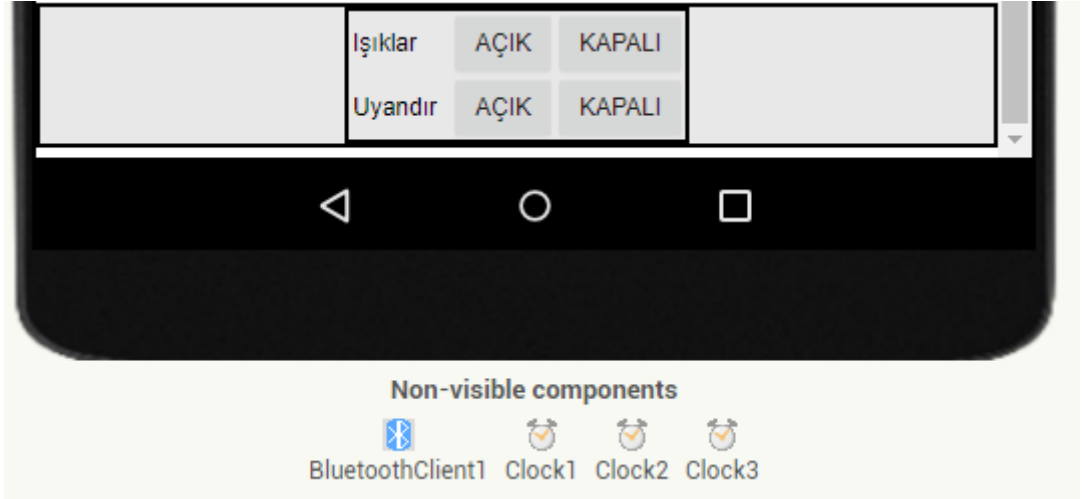
En alt kısımda da kullanılacak olan modların kullanıcıya açıklaması yapılır.



Şekil 28

Uygulamamızın son kısmında ışık ve uyandırma kontrolü yapılıyor. Işıklar kontrolü; açık butonuna tıklandığında, açık butonu pasif diğer kapalı butonu ise aktif konumda kalıyor. Eğer kapalı butonuna basılır ise, kapalı butonu pasif açık butonu ise aktif olarak kalıyor.

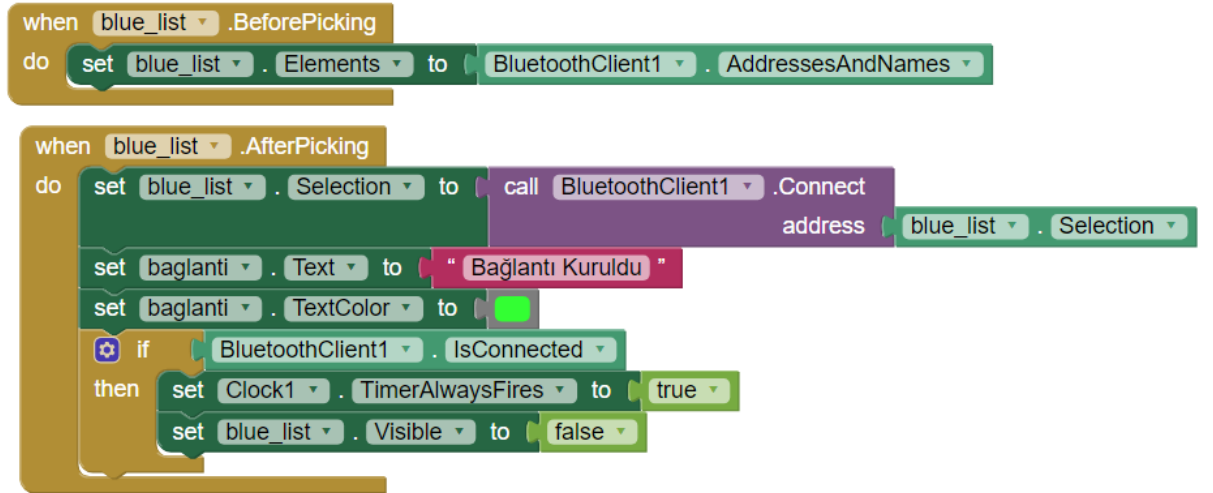
Uyandır butonunun açık butonuna basılır ise, açık butonu pasif kapalı butonu aktif konumda kalır. Kapalı butonun basılırsa az önceki işlemler tekrar eder.



Şekil 29

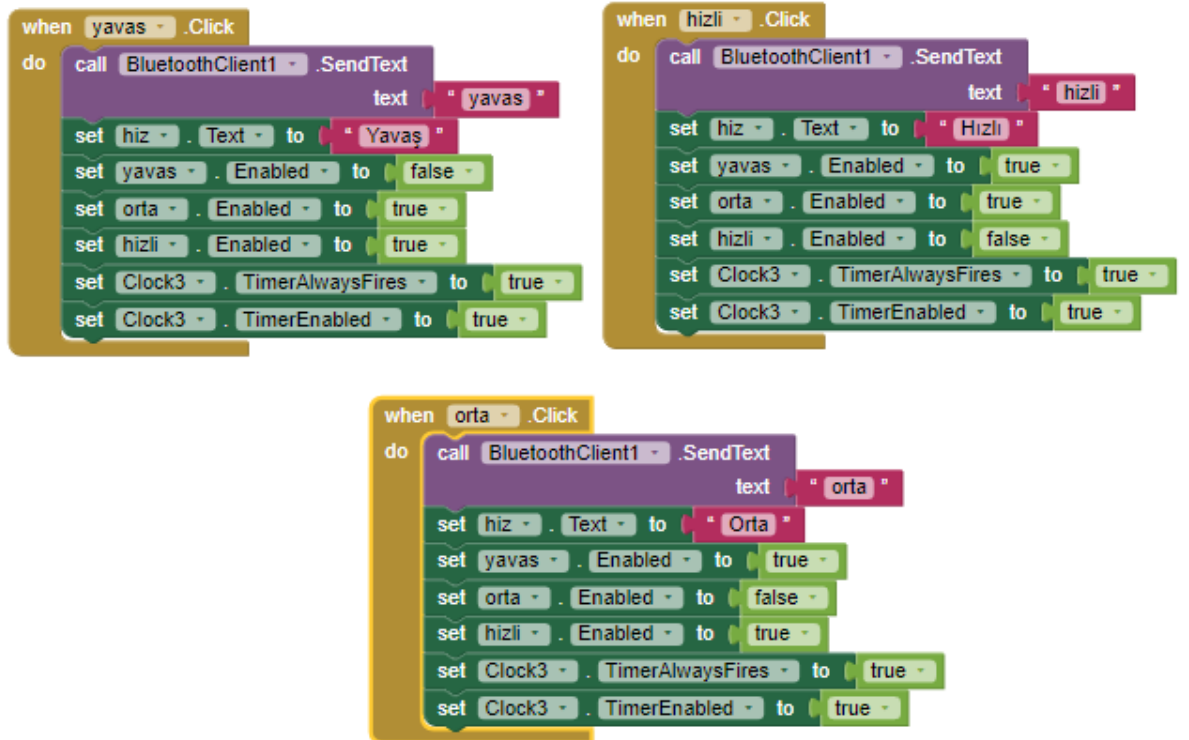
5.1. Android Uygulama Kod Kısım1

Bluetooth bağlantısının yapıldığı kod kısmı. Aygıt seçiminin yapıldığı ve sonucunda bağlantı yapılırsa “Bağlantı Kuruldu” yazan ve renginin yeşil olduğunu belirttiğimiz kısım.



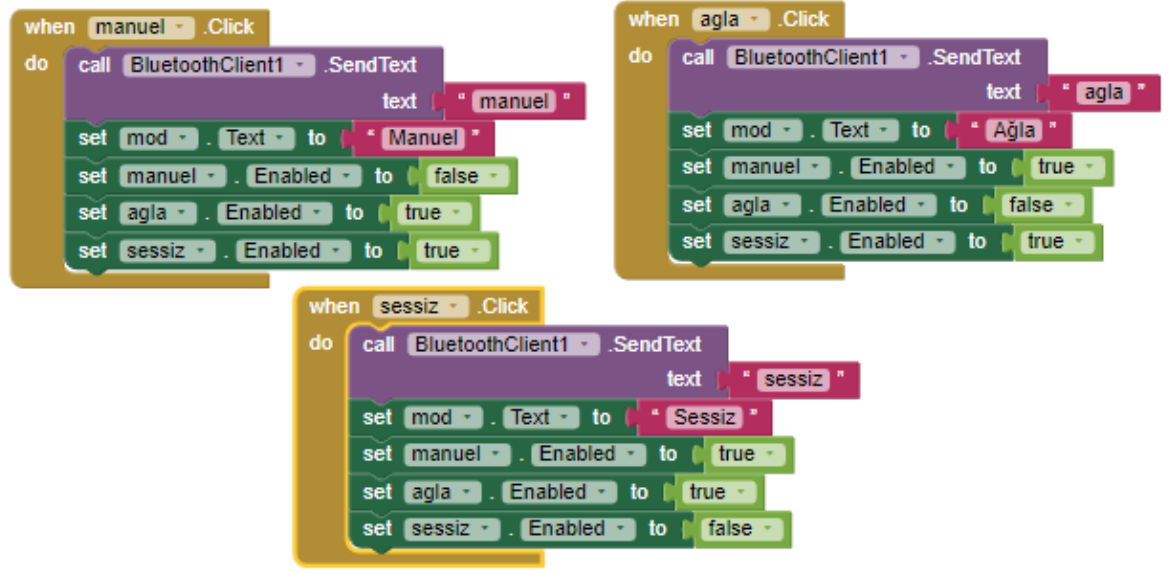
Şekil 30

Hız seçimi yapılırken basılacak butonların ayarlarını ve göndermesi gereken ifadeyi tanımladık.



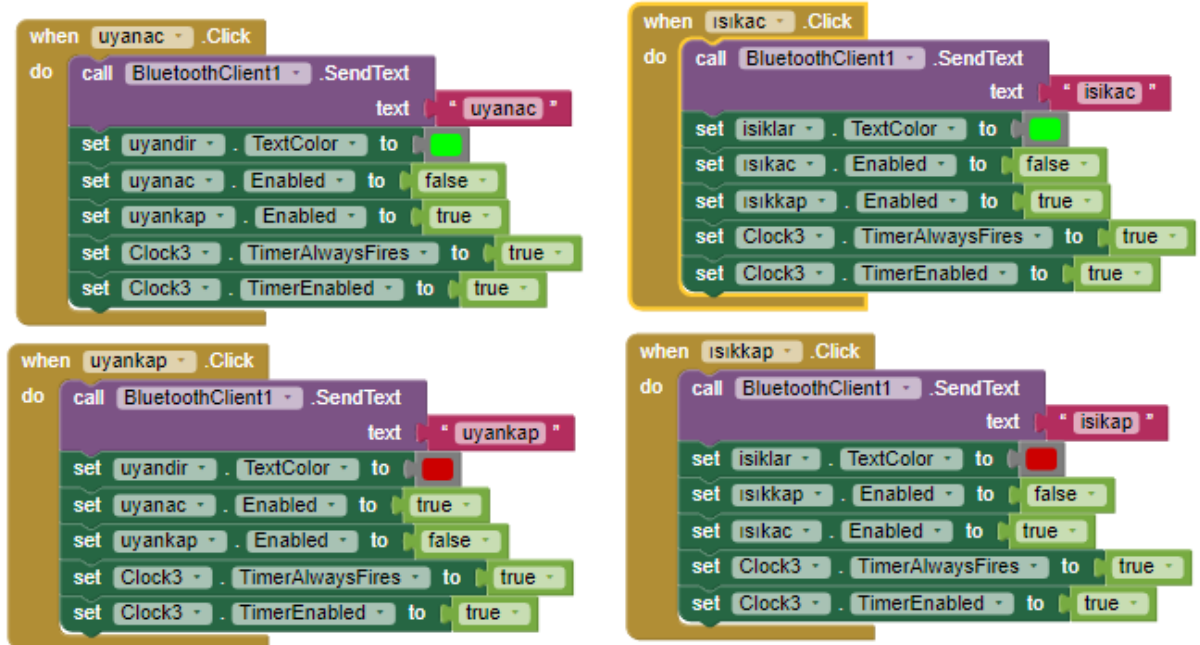
Şekil 31

Hız seçimi yapılırken basılacak butonların ayarlarını ve göndermesi gereken ifadeyi tanımladık.



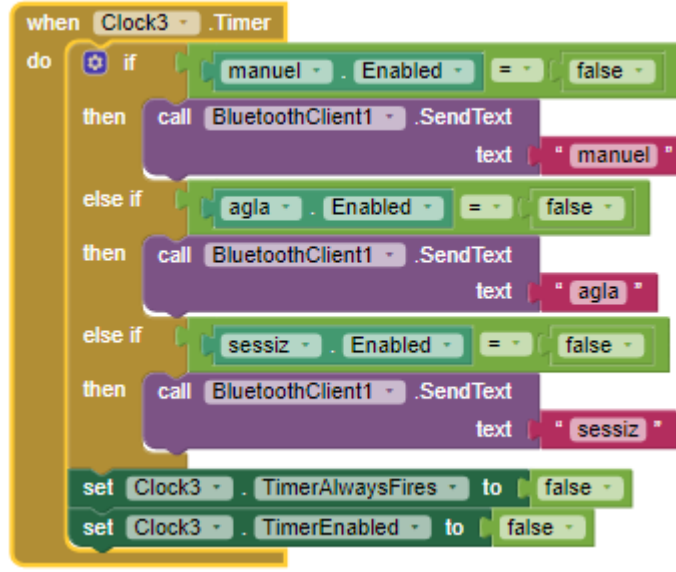
Şekil 32

Uygulamamızın son kısmında bulunan ışık ve uyandırma butonlarını ne göndermesi gerektiğini ve birbiri ile ilişkisini tanımladık. Clock3 aktif yapıyoruz bu bizim butona basıldığında mod kısmımızın gönderdiği veriyi kesiyordu bende belli bir saniye sonra tekrardan basılı olan mod kısmının veri göndermesini başlatıyorum.



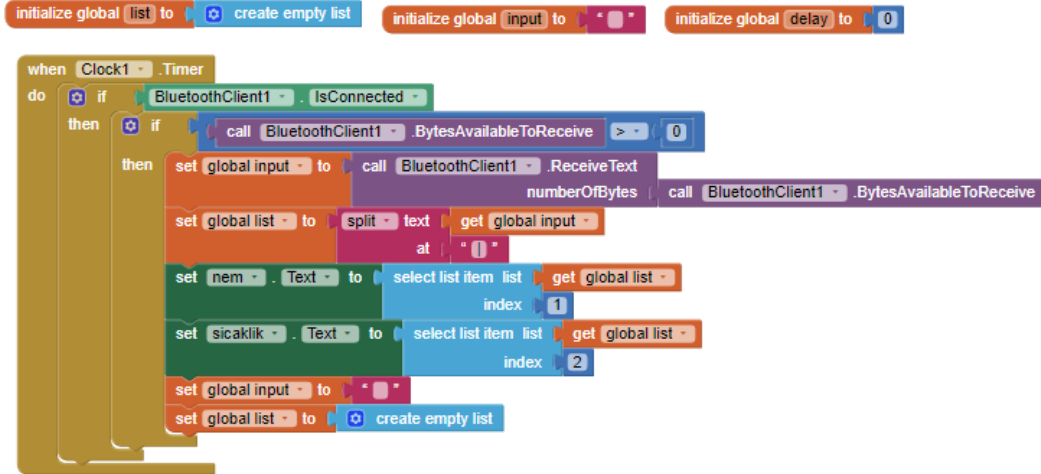
Şekil 33

Yukarda Şekil-33 de bahsettiğim Clock3 kısmının kod kısmı Şekil-34 de gösterdiğim gibi olmaktadır.



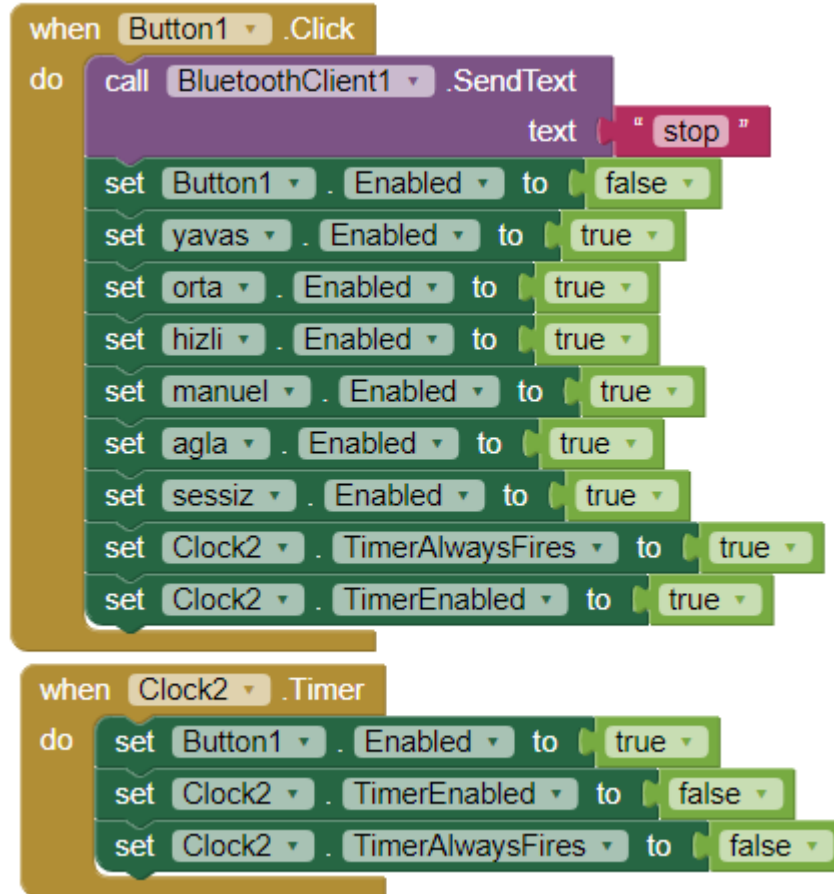
Şekil 34

Bu kısım bizim arduinomuzun bize göndermiş olduğu sıcaklık ve nem verilerini “|” bu işaret baz alınarak global değişkenimize atıyoruz. Ve gelen veri ilk parçası nem labelında ikinci veri ise sıcaklık labelında sergileniyor. Sonra hafıza boşaltılıp çıkılıyor.



Şekil 35

Bu kısımda uygulamamızın en üst kısmında yer alan resimli buton bizim açma kapama tuşumuz. Veri gönderimini kesmek bu tuşa basılması gerekir. Bu diğer butonları aktif eder ve arduinoya “stop” komutunu yollar. Arduinomuz motor dönüşünü kapatarak sallanmayı bitirir. Kullanıcı tekrar veri girene kadar.



Şekil 36

Kaynakça

- 1- <https://www.robotistan.com/sensor>
- 2- <https://www.robotistan.com/bluetooth-1>
- 3- <http://ai2.appinventor.mit.edu/>
- 4- <https://www.arduino.cc/>
- 5- <https://maker.robotistan.com/appinventor-ile-arduino-kontrolu/>
- 6- <https://howtomechatronics.com/tutorials/arduino/how-to-build-custom-android-app-for-your-arduino-project-using-mit-app-inventor/>
- 7- <https://appinventor.mit.edu/>