

## Arduino projesi için araştırma

Proje başlangıcında belirli bir alana giren cismin hızının ölçülmesi, ölçülen değerlerin tablet veya telefon aracılığıyla kullanıcıya gösterilmesi amaçlandı. Örneğin bir futbolcu kaleye şut çekiyor ve antrenör futbol topunun kaleye hangi hızla girdiğini elindeki tabletten görebiliyor olacak. Bu projenin hayata geçirilebilmesi için hangi malzemelerin gerektiğini ve nasıl bir yol izlenmesi gerektiğinin belirlenmesi için araştırmalar yaptım

Araştırmalarım sonucunda yapılan kodlamaların derlenebilmesi için işlemci(Board) olarak arduino uno'yu seçtim. Herkesin de bildiği üzere herhangi bir cimin hızının ölçülebilmesi için cismin aldığı yolun ve bu yolu aldığı zamanın bilinmesi gerekmektedir. Kısacası cismin aldığı yolu ve zamanı hesaplayabileceğim sensörleri araştırmaya başladım.

Sensör araştırmalarım doğrultusunda Hc-Sr04 (ultrasonik mesafe sensörü) (Şekil 4.1), Ldr (foto direnç sensörü) (Şekil 4.2) ve HB100 (mikrodalga doppler radar sensörü) (Şekil 4.3) sensörlerinin projede kullanılabileceğini düşündüm. Cihaz ile tablet arasındaki iletişimi sağlamak için wi-fi ve bluetooth olmak üzere 2 seçenekten bluetooth modülünün proje için daha verimli olabileceğini düşündüm. Bluetooth modüllerini araştırdığımda Hc05 ve Hc06 olarak iki farklı modeli olduğunu gördüm. Bu modüller arasındaki farkları araştırdığımda Hc05 modülünün Hc06 modülünden farklı olarak kendi kendine iletişim başlatabileceğini bu yüzden Hc05 modülünün (Şekil 4.4) projeye daha uygun olacağını gördüm.

Bütün bu araştırmalarımın sonuçlarını raporlayarak, raporumu yetkili kişiye sundum. Verdiğim bilgiler doğrultusunda projeye nasıl devam etmemiz gerektiği, ne gibi malzemeler kullanmamız gerektiğini ve bu malzemelerin maliyetleri hakkında fikir alışverişinde bulunduk. Yaptığımız toplantı sonucunda 2 adet Hc-Sr04 (ultrasonik mesafe sensörü) ile 2 farklı noktada geçen cismi algılayıp bu iki nokta arasında geçen süreyi bulmak amaçlandı. Sensörler arasındaki mesafe bilindiği için buradan hız bulunabileceği sonucuna varıldı. Seçtiğim board ve bluetooth modülü uygun bulundu.



(Şekil 4.1)



(Şekil 4.2)



(Şekil 4.3)



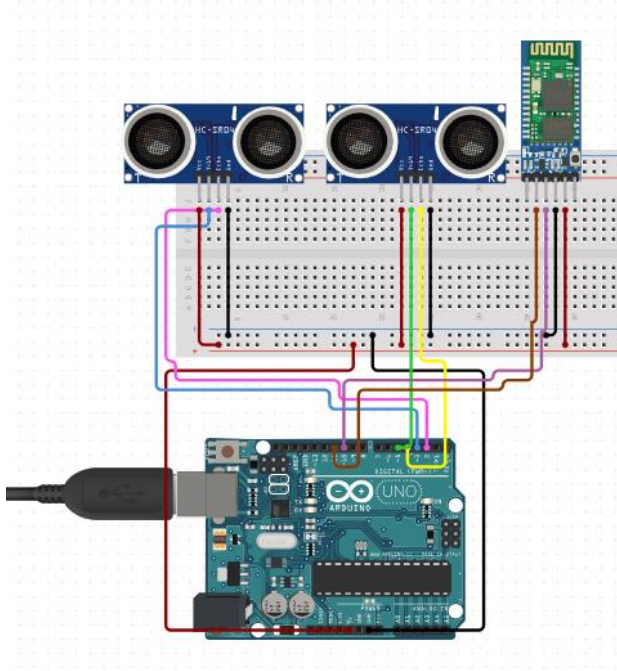
(Şekil 4.4)

## Devre şemasının çizilmesi ve malzemelerin alınması

Gerekli Malzemelerin belirlenmesinden sonra devre şemasını sanal ortamda modelledim (Şekil 5.1). Alınması gereken malzemeleri listeledim ve temin ettim. Alınan malzemeler;

- Arduino uno
- Hc-Sr04 (ultrasonik mesafe sensörü) x 2 adet
- Hc-05 (bluetooth modülü)
- Jumper erkek-erkek kablo
- Breadboard
- Dirençler

Alınan malzemeler, oluşturulan modelleme doğrultusunda breadboard üzerinde bağlantılarını gerçekleştirdim. Ardından arduino'nun programlanabilmesi için gereken yazılımsal ortamın yani Arduino IDE'nin kurulumunu yaptım. Sensörlerin ilk testi için gereken kodlamaları yaptım (Şekil 5.2). Yapılan kodlamaları arduino'ya yüklenip sensörlerin doğru çalışıp çalışmadığını, hassasiyetleri ve algı mesafelerini test ettim. Sorunsuz çalışan sensörler yetkili kişi tarafından da teste tabi tuttu. Yapılan testler sonucunda yüksek hızlarda sensörlerin topu görmediğini fark ettik.



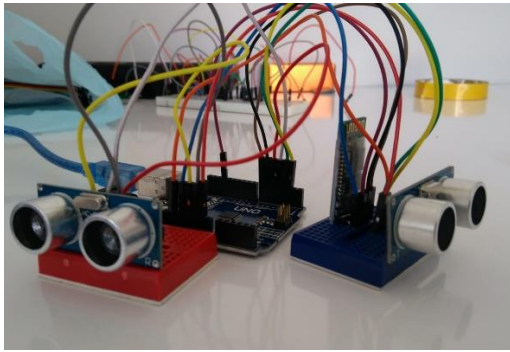
(Şekil 5.1)

```
void setup() {  
  pinMode(trigger, OUTPUT);  
  // Sensörün Trigger bacağına gerilim  
  //uygulayabilmemiz için OUTPUT yapıyoruz.  
  pinMode(echo, INPUT);  
  // Sensörün Echo bacağındaki gerilimi  
  //okuyabilmemiz için INPUT yapıyoruz.  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(trigger, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigger, LOW);  
  
  Time = pulseIn(echo, HIGH);  
  // Dalga üretildikten sonra geri yansıyor Echo  
  //bacağına HIGH duruma geçireceği süreyi pulseIn  
  //fonksiyonu ile kaydediyoruz.  
  TotalRoad = (double)Time*0.034;  
  Distance = TotalRoad / 2;  
  
  //Serial.println(Distance);  
  if(Distance < 150){  
    Serial.println(Distance);  
  }  
}
```

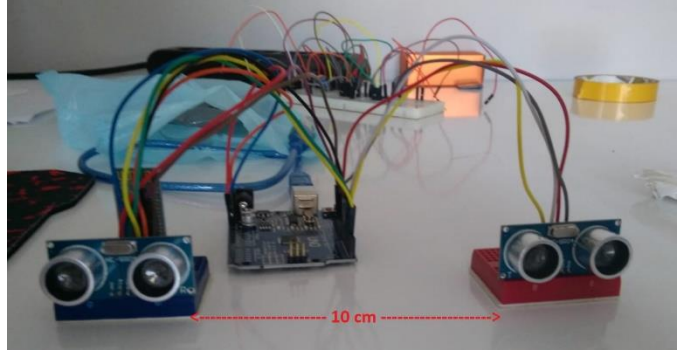
(Şekil 5.2)

## Çift sensör ile yatay düzlemde hız hesaplama

Yapılan testler sonucu sensörlerin cisimleri yüksek hızlarda göremediği tespit edilmişti. Bu hassasiyetin daha net ölçülebilmesi ve hangi hızlara kadar görüş sağladığını görmek amacıyla hız hesaplamalarına geçildi. Bunun için sensörlerin arasının ölçülüp 10cm'de sabitledikten sonra (Şekil 6.2) zaman tespiti için gerekli kodlarını yazdım. Milisaniye cinsinden bulunan zaman değerini saniyeye çevirip, kendi belirlediğim 10cm uzunluğa bölünmesi ile ölçülen hız değerlerini ekrana yazdırdım. Yapılan bu kodlamaların bir kısmı (Şekil 6.3 ve Şekil 6.4)'de görülebilir.



(Şekil 6.1)



(Şekil 6.2)

```
unsigned long Time[2];
double TotalRoad[2];
double Distance[2];
double Speed;

int maxDistance = 200;

void setup() {
  // put your setup code here, to run once:

  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);

  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  for(int i=0; i<2; i++){

    digitalWrite(trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigger, LOW);

    Time[i] = pulseIn(echo, HIGH);
    TotalRoad[i] = (double)Time[i]*0.034;
    Distance[i] = TotalRoad[i] / 2;
```

(Şekil 6.3)

```
    if(i==1 && Distance[i]>0){
      Measurement();
    }
  }

void Measurement(){

  if(Distance[1]<maxDistance && Distance[0]<maxDistance){
    double distance = -(Distance[1]-Distance [0]);
    double time = (Time[1]-Time[0])/2;
    time = time/1000; // milisaniye - saniye dönüşümü

    Speed = (distance)/(time);

    Serial.print("Hız : ");
    Serial.print(Speed);
    Serial.println(" cm/sn");

    Speed = (Speed*0.036); // cm/sn - Km/h dönüşümü

    Serial.print("Hız : ");
    Serial.print(Speed);
    Serial.println(" km/h");
    return;
```

(Şekil 6.4)

## Tek sensör ile karşıdan gelen cismin hızını hesaplama

Yapılan sistem tekrar teste tabi tutuldu. Yapılan testler sonucu daha net görülmekte olan sensör hassasiyetinin, topu 15 km/h hızdan sonrasında nadiren gördüğü, 20 km/h hızdan sonra hiç görmediği fark edildi. Bunun üzerine Hc-Sr04 ultrasonik mesafe sensörünün gelişmiş versiyonlarını araştırarak, bu sensörlerin fiyatları tartışıldı. Fiyatları fazla görülen bu sensörler projeye uygun görülmedi.

Sonuca ulaşmak için sensör değişikliği yerine sistemin çalışma prensibinin değiştirilebilirliği hakkında fikirler ürettim. Öne sürdüğüm bu fikirlerin içerisinde sensörün karşıdan gelen cisme odaklanması ve bunun üzerinden hız ölçümü yapmasını kararlaştırdım. Bu uygulamayı hayata geçirebilmek için araştırmalar yaptım.

Araştırmalarımın sonucunda sensörün karşıdan gelen cismi 2 kez algılayıp, cismin mesafelerini bir diziye atadıktan sonra uzaklıklar arasındaki farkı mesafe olarak kabul etmek ve cismin bu mesafeyi ne kadar zamanda geçtiğini ölçerek, cismin hızının bulmayı amaçladım. Bu doğrultuda gereken kodlar yazdım (Şekil 7.1 ve Şekil 7.2) ve devre üzerinde yapılması gereken değişiklikleri (Şekil 7.3) yaptım.

```
unsigned long Time[2];
double TotalRoad[2];
double Distance[2];
double DistanceTime[2];
double distance;
double TimeS = 100;

int maxDistance = 30;

void setup() {
    // put your setup code here, to run once:

    pinMode(trigger, OUTPUT);
    pinMode(echo, INPUT);

    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:

    for(int i=0; i<2; i++){

        digitalWrite(trigger, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigger, LOW);
```

(Şekil 7.1)

```
Time[i] = pulseIn(echo, HIGH);
//Distance[i] = (Time[i]/2)/29.1;
TotalRoad[i] = (double)Time[i]*0.034;
Distance[i] = TotalRoad[i] / 2;

delay(TimeS);

if(Distance[i]<maxDistance && Distance[i+1]<maxDistance){

    distance = (double)(Distance[i] - Distance[i+1]);
    TimeS = TimeS/1000;    // milisaniye - saniye dönüşümü

    double Speed = (double)(distance)/(TimeS);

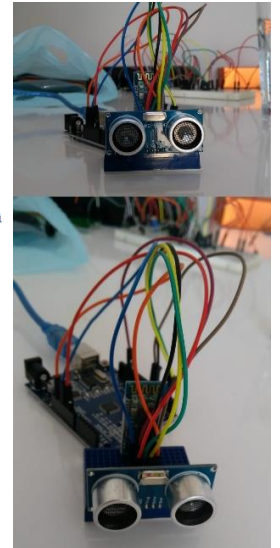
    Serial.print("Hareket algılandı\n");
    Serial.print("Hız : ");
    Serial.print(Speed);
    Serial.println(" cm/sn");

    Speed = (double)(Speed*0.036) // cm/sn - Km/h dönüşümü

    Serial.print("Hız : ");
    Serial.print(Speed);
    Serial.println(" km/h\n");

}
```

(Şekil 7.2)



(Şekil 7.3)

## Giriş-Çıkış sensör kontrol düzeni

X-bandında hız ölçen sistemin son düzenlemelerini yaptıktan sonra top ile test etme aşamasına geçtim. Yapılan testler doğrultusunda sensörün bu şekilde kullanılmasının algılamadaki hassasiyetini arttırdığı gözlemlendim. Sensörün 20 km/h'den sonrasını nadiren, 30 km/h'den sonrasını hiç algılamadığını tespit ettim. Hassasiyetin artırılması doğrultusunda algılama alanını daralttığını gözlemlendim. Alanın daralması proje kapsamında maliyeti arttırmamasından ve uygulanabilirliği azaltmasından dolayı bu yöntemden vazgeçildi.

Yapılan testler sonrasında projeye yeni problemlerin eklendiği ve bu problemler doğrultusunda nasıl bir çözüm yolu uygulanması gerektiği konusunda araştırmalar yaptım. Problemlerden biri olan algılama alanının nasıl artırılabilceği yönünde sensörlerin arkalı önlü koyularak topun hem geliş yönünde hem de çıkış yönünde algılanmasıyla hassasiyetin ve alanın artırılabilceğini ön gördüm.

Böylelikle sensörlerden biri topun ilk girişini algılayacak ve diğerinde çıkış yönünde bir algılama sağlayarak aradaki hız hesaplanacaktı. Bu doğrultuda proje için çalışmaya başladım. Gerekli devre düzenini kurup kodlar üzerindeki değişiklikleri (Şekil 8.1 ve Şekil 8.2) yaptım. Problemlerden bir diğeri olan uygulanabilirlik projenin ilerleyen aşamalarına bırakıldı.

```
void loop() {  
  
  Sensor1();  
  Sensor2();  
  
  if(endFlag == true){  
    Calculate();  
  }  
}  
  
void Sensor1(){  
  
  digitalWrite(trigger, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigger, LOW);  
  
  Time = pulseIn(echo, HIGH);  
  TotalRoad = (double)Time*0.034;  
  Distance = TotalRoad / 2;  
  
  if(Distance<maxDistance){  
    DistanceTime = millis();  
    endFlag = false;  
  }  
}
```

(Şekil 8.1)

```
void Sensor2(){  
  
  digitalWrite(trigger2, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigger2, LOW);  
  
  Time2 = pulseIn(echo2, HIGH);  
  TotalRoad2 = (double)Time2*0.034;  
  Distance2 = TotalRoad2 / 2;  
  
  if(Distance2<maxDistance && endFlag == false){  
    DistanceTime2 = millis();  
    endFlag = true;  
  }  
}  
  
void Calculate(){  
  
  double CalculateTime = (DistanceTime - DistanceTime2) ;  
  double CalculateDistance = (Distance + Distance2);  
  double Speed = (CalculateDistance/(CalculateTime/1000));  
  
  Serial.print("Speed : ");  
  Serial.print(Speed);  
  Serial.println(" cm/sn");  
  
  Speed = (Speed/100)*(36/10); // cm/sn to km/h
```

(Şekil 8.2)



## Arduino projesinin minyatür kalede test edilmesi

Yapılan giriş-çıkış algılama sisteminin hassasiyetinin ölçülebilmesi için minyatür kale üzerinde testlerine başladım. Sistemin minyatür kaleye montajından sonra topun hızı ölçülmeye başladım. Testler sonrasında kalenin en alt kısmından geçmekte olan cisimleri görmediği ve algı alanının yetersiz olmasından dolayı sistemin iyileştirilmesi için çalışmalar yapılması istenildi. Testler bitirilmeden önce sistem x-bandında ölçüm yapan tek sönör haline çevrilerek kalenin arkasına koyulmasına ve test edilmesine karar verildi. Bu yöntem verimli olmadığı için iptal edildi.

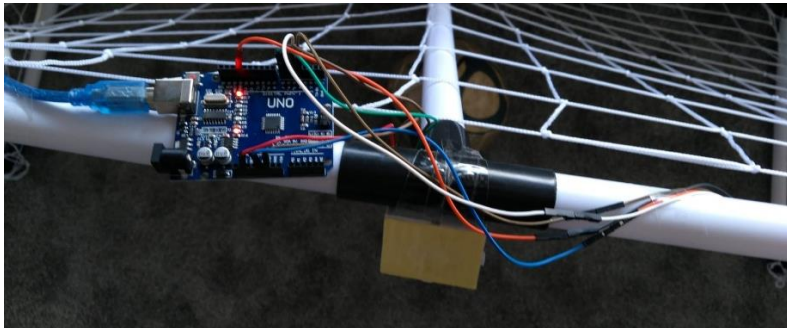
Algı alanını genişletmek adına kalenin üst direğine aralarındaki mesafe 33cm olacak şekilde 3 adet sensör monte ettim (Şekil 9.1, Şekil 9.2, Şekil 9.3 ve Şekil 9.4). Aynı zamanda kodları iyileştirmek için çalıştım. Çalışmalar sırasında zaman tespitinde sorunlarla karşılaştım. Bu sorunları çözmeye çalıştım. Zamanlama sorunu çözüldükten sonra test aşamasına geçtim. Test sonuçlarının doğrultusunda verimli sonuç alamadım. Bunun üzerine kullanılan Hc-Sr04 ultrasonik mesafe sensörünün projeye uygun olup olmadığı tartışıldı.



(Şekil 9.1)



(Şekil 9.2)



(Şekil 9.3)



(Şekil 9.4)

## Farklı sensörlerin araştırılması ve ldr sensörünün testi

Sensörler istenilen hızları ölçmek için yeterli olmadığından kullanılan sensörlerin proje kapsamında doğru seçim olup olmadığı gündeme getirilmişti. Sensörün proje için doğru olup olmadığı araştırırken bir yandan da farklı sensörleri araştırmaya devam ettim.

Proje başında araştırılan sensörlerden LDR (light dependent resistor) sensörü için devre oluşturulup gerekli kodlamalar (Şekil 10.1 ve Şekil 10.2) yapıldı. Sensörün ışık ortamıyla birebir etkileşimi olduğundan sensörün dış ortamda kullanılamayacağı kararlaştırıldı. Dış ortama uygun olması adına sistemin ilk 1 milisaniyesinde ortamın ışık oranının, ortalamasının alınıp hesaplamaya aktarılması fikrini öne sürdüm fakat yöntemin sağlıklı olacağı, bu yüzden ldr (Foto direnç) sensörünün projeye uygun olmadığı kararlaştırıldı. Bunun üzerine proje için tekrar yeni sensörler araştırmaya başladım.

```
void Sensor1(){
    sensor1Value = analogRead(A4);

    if(sensor1Value < average_Sensor1){
        startTime = millis();
        endFlag = false;
    }
}

void Sensor2(){
    sensor2Value = analogRead(A5);

    if(sensor2Value < average_Sensor2 && endFlag == false){
        endTime = millis();
        endFlag = true;
    }
}

void average(){
    for(int i=0; i<100; i++){
        average_Sensor1 += analogRead(A4);
        average_Sensor2 += analogRead(A5);
    }
    average_Sensor1 = (average_Sensor1/100);
    average_Sensor2 = (average_Sensor2/100);
}
```

(Şekil 10.1)

```
void Calculate(){
    double Time = (endTime) - (startTime);
    double TimeInSeconds = Time/1000;
    double Speed = distance/(TimeInSeconds);

    Speed = distance/(CalculateTime/1000);

    Serial.print("Speed: ");
    Serial.print(Speed);
    Serial.println(" cm/s ");

    Speed = (Speed*0.036);
    Serial.print("Speed: ");
    Serial.print(Speed);
    Serial.println(" km/h \n");
}

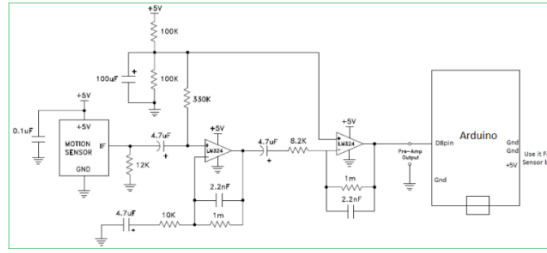
void MoveTest(){
    if(endFlag == true){
        endFlag = false;
        Serial.println("Hareket Algılandı");
    }
}
```

(Şekil 10.2)

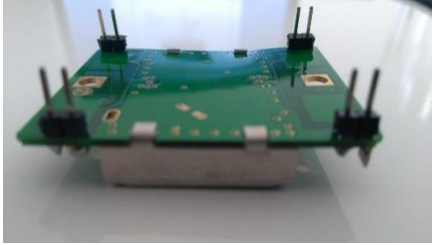
## Hb100 mikrodalga doppler sensörü ile hız ölçümü

Tekrar arduino projesine döndüm. Proje için uygun sensör aramaya başladım. Araştırmalarım sonrasında Hb100 mikrodalga doppler radar sensörünün kullanılmasına karar verdim. Hb100 ve arduino ile yapılmış olan radar silahı projelerini araştırdım ve örnek aldım. Radar silah projelerinin projemize uygulanabilirliğini kontrol ettim.

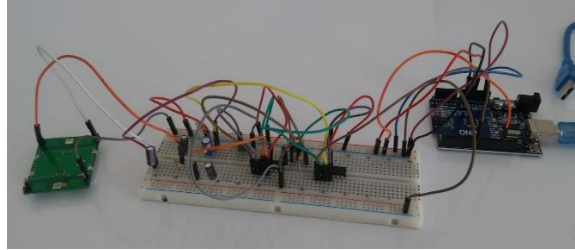
Hb100'ün sinyal yayması için gerekli amfi devresi tasarımlarını inceledim. Devre şeması çıkarttım (Şekil 13.1). Arduino'ya uyumlu versiyonunu bulamadığım için cihaza pinler lehimleyerek uygun hale getirdim (Şekil 13.2). Gerekli bağlantıları breadboard üzerinde oluşturduktan (Şekil 13.3) sonra kodlarını yazılmaya başladım (Şekil 13.4). Devrenin testini gerçekleştirdim. Testler sonucu verilerin yanlış geldiğini tespit ettim.



(Şekil 13.1)



(Şekil 13.2)



(Şekil 13.3)

```
#include <FreqMeasure.h>

double lfrq;
long int pp;

void setup() {
  Serial.begin(9600);
  Serial.println("FreqPeriod Library Test");
  FreqMeasure.begin();
}

double sum=0;
int count=0;

void loop() {
  if(FreqMeasure.available()){
    sum=sum+FreqMeasure.read();
    count+=1;
    if(sum>15){
      float frekans=FreqMeasure.countToFrequency(sum/count);
      Serial.print("Frekans : ");
      Serial.println(frekans);
      Serial.print("HIZ : ");
      Serial.println(frekans/19.49);
      sum=0;
      count=0;
    }
  }
}
```

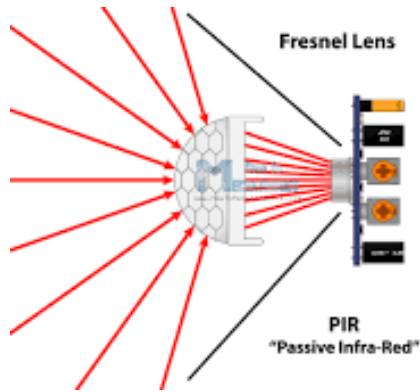
(Şekil 13.4)



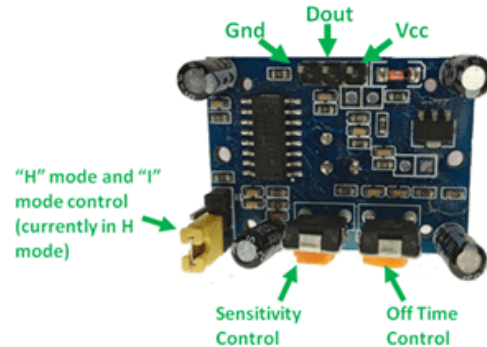
## Hb100 hatalarının tespiti ve Pır Sensörü

Hb100 mikrodalga doppler sensöründen veri alamadım. Zaman zaman aldığım verilerin ise hatalı olduğunu belirledim. Bunun sebeplerini araştırdım. Devre bağlantılarını kontrol ettim. Kodları kontrol ettim. Bir sorunla karşılaşmadım ve hatanın nedeni bulamadım. Bu yüzden devreyi breadboard üzerinde sıfırdan yapmaya başladım. Bağlantıları tekrar kontrol ettim. Fakat sonuç değişmedi. Multimetre ile bağlantıları kontrol ettim. Verilerin yanlış gelmesinin nedenini bulamadım. Bu sorunun nedeni üzerine araştırmamalar yaptım. Araştırmalar sonrasında nedeni bulunamayan hatalardan dolayı Hb100 projeden çıkarıldı.

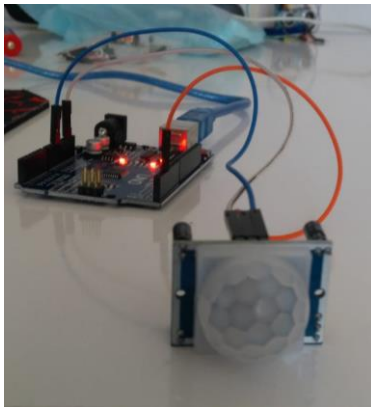
Hb100 kullanılmaktan vazgeçilerek yerine HC-SR501 ayarlanabilir IR hareket sensörü (PIR) kullanılmasına karar verildi. Bunun için gerekli araştırmaları yaptım. Çalışma prensibini araştırdım (Şekil 14.1 ve Şekil 14.2). Sensörü tedarik ettim. Sensör zaman ve hassasiyet anahtarlarını en düşüğe alarak "I" moduna ayarladım. İlk testler için gerekli bağlantıları yaptım (Şekil 14.3) ve sensörün algılama kapasitesini ölçmek için gerekli kodları yazdım (Şekil 14.3).



(Şekil 14.1)



(Şekil 14.2)



(Şekil 14.3)

```
int pirPin = 3; // PIR pin
int deger = 0;

void setup() {
  pinMode(pirPin, INPUT); // PIR Pin'i giriş yapılıyor
  Serial.begin(9600); //Serial Porttan veri göndermek için baudrate ayarlanıyor.
}

void loop(){
  deger = digitalRead(pirPin); // Dijital pin okunuyor

  if (deger == HIGH) {
    Serial.println("Hareket Algılandı");
  }
  else{
    Serial.println("Hareket Yok");
  }
}
```

(Şekil 14.3)

## Hareketin süresini tutma ve Monster Notebook akışı

PIR Sensörü ile algılama kapasitesini ölçtükten sonra önünden geçen cismin hareketini ne kadar sürede bitirdiğini hesaplamaya geçtim. Bunun için kodlamalar yaptım (Şekil 15.1). Fakat PIR sensörünün çalışma prensibinden kaynaklanan sayma özelliği hareketin bitmesinden sonra 4 saniye daha high konumunda kalmasına neden olmaktaydı. Bu durum hız hesaplarken sorun çıkartacağından dolayı hesaplama sırasında bulunan süreden 4 saniye çıkartarak işlem yaptım. Hesaplamada uyguladığım bu yöntem cihazın 4 saniyelik bir gecikme ile cevap vermesine yol açtı. Gecikmeyi nasıl önleyebileceğim hakkında çalışmalara başladım. Aynı zamanda projenin nasıl devam etmesi gerektiğine ilişkin yetkili kişilerden fikirler aldım.

```
void loop() {  
  deger = digitalRead(pirPin);  
  
  if (deger == HIGH){  
    startTime = millis();  
    Time = startTime - endTime;  
    Time= Time-DefaultTime;  
  }  
  
  else{  
    endTime = millis();  
  
    Speed = (distance/(Time/1000));    // cm/sn  
  
    if(Speed < 9999999 ){  
      Serial.print("Speed : ");  
      Serial.print(Speed);  
      Serial.print(" cm/sn\n");  
  
      Speed = (Speed/100)*(36/10);    //cm/sn to km/h  
  
      Serial.print("Speed : ");  
      Serial.print(Speed);  
      Serial.print(" km/h\n\n");  
  
      //delay(500);  
      return;  
    }  
  }
```

(Şekil 15.1)

## Android uygulaması ile bluetooth cihazına bağlanma

Android Studio yükleyerek SmartBall adlı bir uygulama oluşturmaya başladım. Açılış sayfasında butonlar koyarak butonların farklı activity'lere geçiş yapmalarını sağladım. Arduino'dan gelen verileri işleme sokmak için ilk olarak uygulama ile Arduino'nun bluetooth modülünün (Hc-05) telefon ile eşleşmesini sağlamak için gerekli kodları yazdım (Şekil 16.1). Uygulamanın bluetooth özelliğini kullanabilmesi için gerekli izinleri kodladım. Yapılan işlemler sonucunda uygulamanın çalışma esnasında telefonun önceden eşleştirilmiş bluetooth cihazlarını görebilmesi ve bu cihazlara bağlanmasını sağladım. Oluşabilecek hatalardan bluetooth cihazına bağlanamama, telefonun bluetooth özelliğinin kapalı olması veya uygulamanın çalışması sırasında bağlantının kopması gibi durumların hata vermesi durumlarında uygulanılacak senaryolar oluşturdum ve bu senaryoları kodladım (Şekil 16.2).

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.bt_control);

    //Calling widgets
    btnPaired = (Button)findViewById(R.id.PairedDevices);
    deviceList = (ListView)findViewById(R.id.ListView);
    HomePage = (Button)findViewById(R.id.ResultPage);

    //if the device has bluetooth
    myBluetooth = BluetoothAdapter.getDefaultAdapter();

    if(myBluetooth == null)
    {
        //Show a msg, that the device has no bluetooth adapter
        Toast.makeText(getApplicationContext(), "Bluetooth cihazı bulunamadı.",
            Toast.LENGTH_LONG).show();

        //finish apk
        finish();
    }
    else if(!myBluetooth.isEnabled())
    {
        //Ask to the user turn the bluetooth on
        Intent turnBtOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(turnBtOn, REQUEST_CODE);
    }

    btnPaired.setOnClickListener((v) -> { pairedDevicesList(); });
    HomePage.setOnClickListener((view) -> { finish(); });

    private void pairedDevicesList()
    {
        pairedDevices = myBluetooth.getBondedDevices();
        ArrayList list = new ArrayList();

        if (pairedDevices.size() > 0)
        {
            for(BluetoothDevice bt : pairedDevices)
            {
                list.add(bt.getName() + "\n" + bt.getAddress());
            }
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Eşleşmiş Bluetooth cihazı bulunamadı.",
                Toast.LENGTH_LONG).show();
        }

        final ArrayAdapter adapter =
            new ArrayAdapter<context, this, android.R.layout.simple_list_item_1, list>;
        deviceList.setAdapter(adapter);
        deviceList.setOnItemClickListener(myItemClickListener);

    private AdapterView.OnItemClickListener myItemClickListener = (av, w, arg2, arg3) -> {
        // Get the device MAC address, the last 17 chars in the View
        String info = ((TextView) w).getText().toString();
        String address = info.substring(info.length() - 17);

        // Make an intent to start next activity.
        Intent i = new Intent(getApplicationContext(), Training.class);
        //Change the activity.
        i.putExtra(EXTRA_ADDRESS, address);
        startActivity(i);
    };
}
```

(Şekil 16.1)

```
private class ConnectBT extends AsyncTask<Void, Void, Void> // UI thread
{
    private boolean ConnectSuccess = true; //if it's here, it's almost connected

    @Override
    protected void onPreExecute()
    {
        progress = ProgressDialog.show(context, Training.this, "Bağlantı Kuruluyor...",
            "Lütfen Bekleyiniz!!!"); //show a progress dialog
    }

    @Override
    protected Void doInBackground(Void... devices) //while the progress dialog is shown
    {
        try
        {
            if (btSocket == null || !isBtConnected)
            {
                myBluetooth = BluetoothAdapter.getDefaultAdapter();
                BluetoothDevice dispositivo = myBluetooth.getRemoteDevice(address);
                btSocket = dispositivo.createInsecureRfcommSocketToServiceRecord(myUUID);
                BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                btSocket.connect(); //start connection
            }
        }
        catch (IOException e)
        {
            ConnectSuccess = false; //if the try failed, you can check the exception here
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void result)
    {
        super.onPostExecute(result);

        if (!ConnectSuccess)
        {
            msg(R.string.Baglanti_basari_siz_oldu, "Lütfen tekrar deneyiniz.");
            finish();
        }
        else
        {
            msg(R.string.Baglanti_basari_siz_bir_gokilde_kuruldu);
            isBtConnected = true;
            progress.dismiss();
        }
    }

    private void Disconnect()
    {
        if (btSocket != null) //If the btSocket is busy
        {
            try
            {
                btSocket.close(); //close connection
            }
            catch (IOException e)
            {
                msg(R.string.Error);
            }
        }
        finish(); //return to the first layout
    }
}
```

(Şekil 16.2)

## Android uygulamasında SQLite işlemleri

Uygulama içerisinde SQLite ile yeni bir veri tabanı ve bu veri tabanına bağlı 2 adet tablo oluşturdum (Şekil 17.1). Bu tablolardan ilki uygulama içerisindeki kişilerin verilerini, diğeri ise bu kişilerin antrenman bilgilerini tutmaktadır. Bu tablolara veri ekleme, veri silme, güncelleme ve listeleme işlemlerini gerçekleştirmek için gerekli kodlamaları yaptım. Daha sonra bu işlemlerin kullanıcı tarafından görülebilecek ve kullanılabilecek haline getirdim (Şekil 17.2). Bu işlemlerin ön taraftaki activity tasarımlarını yaptım (Şekil 17.3).

```
public DataBase_SQLite(Context context) {
    super(context, DATABASE_NAME, factory, null, DATABASE_VERSION);
}

@Override
public void onCreate(SQLiteDatabase db) {

    db.execSQL("CREATE TABLE " + TABLE_PERSONS + "("
        + ROW_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + ROW_NAME + " TEXT NOT NULL,"
        + ROW_SURNAME + " TEXT NOT NULL,"
        + ROW_AGE + " TEXT NOT NULL,"
        + ROW_FOOT + " TEXT NOT NULL,"
        + ROW_POSITION + " TEXT NOT NULL)");

    db.execSQL("CREATE TABLE " + TABLE_TRAINING_DATA + "("
        + ROW_TRAINING_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + ROW_PERSONS_ID + " INTEGER NOT NULL,"
        + ROW_TRAINING_DATE + " TEXT NOT NULL,"
        + ROW_TOTAL_SHOTS + " INTEGER NOT NULL,"
        + ROW_SUCCESSFUL_SHOTS + " INTEGER NOT NULL,"
        + ROW_PERCENT_SHOTS + " INTEGER NOT NULL)");

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_PERSONS);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_TRAINING_DATA);

    onCreate(db);

}
```

(Şekil 17.1)

```
lastPage.setOnClickListener(view) -> {
    Intent i = new Intent( packageContext, EditPlayer.class);
    startActivity(i);
});

playerDelete.setOnClickListener(view) -> {

    DataBase_SQLite vt = new DataBase_SQLite( context, EditPlayer.this);
    vt.DeletePlayer_TablePerson(idBul);
    Intent i = new Intent( packageContext, EditPlayer.class);
    startActivity(i);
});

editPlayer.setOnClickListener(view) -> {

    String cmgName = Name.getText().toString();
    String cmgSurname = Surname.getText().toString();
    String cmgAge = Age.getText().toString();
    String cmgFoot = Foot.getText().toString();
    String cmgPosition = Position.getText().toString();

    if (cmgName.length() > 0 && cmgSurname.length() > 0 && cmgAge.length() > 0
        && cmgFoot.length() > 0 && cmgPosition.length() > 0) {
        // Veritabanı bağlantımızı açalım ver ardından gerekli bilgileri VeriDuzenle() gönderelim
        DataBase_SQLite vt = new DataBase_SQLite( context, EditPlayer.this);
        vt.EditPlayer_TablePerson(idBul, cmgName, cmgSurname, cmgAge, cmgFoot, cmgPosition);

        msg( R.string.Degisiklikler_kaydedildi.);
    } else {
        msg( R.string.Lutfen_gerekli_alanlari_doldurunuz.);
    }
}

});
```

(Şekil 17.2)

The screenshot displays the SmartBall application interface. On the left, a sidebar menu contains options: ANTREMENTA BAŞLA, İSTATİSTİKLERİ GÖR, OYUNCU DÜZENLE, AYARLAR, and ÇIKIŞ YAP. The main area shows a list of players with their names, ages, and positions. A player's details are shown in a modal view, including their name, surname, age, foot, and position. The list of players includes: 8 - Fernando - Muslera - 32 - sağ - Kaleci, 9 - Mariano - Filho - 32 - Sağ - Sağ bek, 10 - Maicon - Roque - 29 - Sağ - Stoper, 11 - Serdar - Aziz - 27 - Sağ - Stoper, 12 - Lionel - Carole - 27 - Sol - Sol bek, 13 - Fernando - Reges - 31 - Sağ - önlibero, 14 - Bafetimbi - Gomis - 33 - sağ - Santrafor, 15 - Eren - Derdiyok - 30 - Sağ - Santrafor, 16 - Sinan - Gümüş - 24 - sol - Sağ kanat, and 17 - Sofiane - Feghouli - 28 - sağ - Sağ kanat. The modal view for Fernando Muslera shows his age as 32, foot as sağ, and position as Kaleci. At the bottom, there are buttons for OYUNCU EKLE, OYUNCUYU SİL, and GERİ DÖN.

(Şekil 17.3)

## Arduino verilerinin işlenmesi ve uygulama tasarımı

Arduino ile oluşturduğum sistem algı alanından bir nesne geçmesini algılamaktaydı. Bu cihazı kalenin üst kısmına ekleyerek kaleye giren topun hareketini algılayıp, verileri android projemize aktarıp oyunculara göre tutmak amaçlanmıştı. Bunun için oyuncuları tutma işlemlerini daha önceden gerçekleştirdik.

Sırada gelen verileri oyunculara göre tutma ve bunları listeleme işlemi kaldı. Bir panel oluşturarak (Şekil 18.1) antrenmanı başlatıp bitirdiğimizde toplam isabetli şut sayısını, toplam şut sayısını, antrenmanın yapıldığı tarihi, oyuncuyu ve isabet yüzdesi verilerini işlemlere sokup genel sonuçları kullanıcıya göstermek için gerekli kodlamaları yaptım (Şekil 18.2). Aynı zamanda kullanıcının bu zamana kadar yapılan tüm antrenmanları oyunculara göre gözlemleyebileceği bir panel de hazırladım. Uygulamanın genel tasarımı ile ilgili iyileştirilmeler yaptım.



(Şekil 18.1)

```
private void msg(String s)
{
    Toast.makeText(getApplicationContext(), s, Toast.LENGTH_LONG).show();
}

private void DataRead()
{
    trainingView.setMovementMethod(new ScrollingMovementMethod());
    trainingView.setMaxLines(Integer.MAX_VALUE);
    final Handler handler = new Handler();
    stopThread = false;
    Thread thread = new Thread((Runnable) () -> {
        while(!Thread.currentThread().isInterrupted() && !stopThread)
        {
            try
            {
                inputStream = btSocket.getInputStream();
                int byteCount = inputStream.available();
                if(byteCount > 0)
                {
                    byte[] rawBytes = new byte[byteCount];
                    inputStream.read(rawBytes);
                    final String string = new String(rawBytes, charsetName: "UTF-8");
                    handler.post(() -> {
                        if(string == "Isabet Algılandı."){
                            Hit = Hit + 1;
                            trainingView.append("\n" + string
                                + " Isabet sayısı : " + Hit);
                        }
                    });
                }
            }
            catch (IOException ex)
            {
                stopThread = true;
            }
        }
    });

    thread.start();
}

public void TrainingDataSave(){
    int TotalShots = 10;
    int percentShots = Hit/TotalShots*100;
    String date = df.format(Now) + "/" + hour + "/" + minute;

    DataBase SQLite dbSQL = new DataBase_SQLite( context Training.this);
    dbSQL.AddData_TableTrainingData(idBul, date, TotalShots, Hit, percentShots);
}
```

(Şekil 18.2)



## SmartBall uygulamasının son haline getirilmesi

Uygulamanın son halini alması için bütün ekran geçişlerini test ettim ve erken geçişleri test ederken fark edilen tasarımsal hataları düzelttim. Uygulamanın menü iconlarını tasarladım ve 4 farklı boyut için uygulamanın içine gömdüm. Oyuncu ekleme ve düzenleme kısımlarında oyuncunun fotoğrafının eklenebileceği bir alan yaptım. Bu alana telefonun galerisinden veya kamerasından fotoğraf eklenebilmesini (Şekil 19.1), eklenen fotoğrafın kullanıcı tarafından kırılmasını (Şekil 19.2) sağladım. Kırpılmış şekilde alınan fotoğrafın son halini byte tipinde bir değişkene çevirerek veri tabanına kaydettim. Tekrar fotoğrafın gösterileceği zaman ise byte tipine çevirme işleminin tam tersini uygulayarak fotoğrafın ekrana yazdırılmasını sağladım. Fotoğraf seçiminde hangi işlemin yapılması istenildiğini soran (Şekil 19.3) bir alan oluşturdum. Bluetooth işlemleri sırasında bekletilen kısımlar için (Şekil 19.4) veya bluetooth seçeneğinin açılması için (Şekil 19.5) bilgiler gönderdim.

```
final String[] items = new String[]{"Camera", "Galeriden Seç"};
ArrayAdapter<String> adapter = new ArrayAdapter<>(context, this,
    android.R.layout.select_dialog_item, items);
android.support.v7.app.AlertDialog.Builder builder =
    new android.support.v7.app.AlertDialog.Builder(context, this);
builder.setTitle("Resim seç");
builder.setAdapter(adapter, (dialog, item) -> { //pick from camera
    if (item == 0) {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        mImageCaptureUri = Uri.fromFile(new File(Environment.getExternalStorageDirectory(),
            "tmp_avatar_" + String.valueOf(System.currentTimeMillis()) + ".jpg"));
        intent.putExtra(android.provider.MediaStore.EXTRA_OUTPUT, mImageCaptureUri);
        try {
            //intent.putExtra("return-data", true);
            startActivityForResult(intent, PICK_FROM_CAMERA);
        } catch (ActivityNotFoundException e) {
            e.printStackTrace();
        }
    } else { //pick from file
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        Intent mIntent = new Intent(Intent.ACTION_PICK,
            android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(mIntent, PICK_FROM_FILE);
    }
});
```

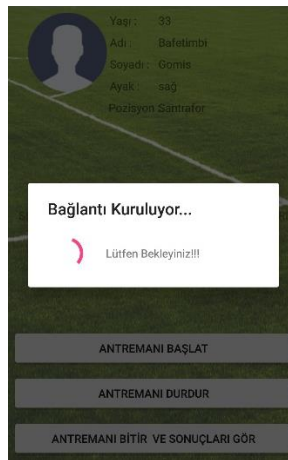
(Şekil 19.1)

```
List<ResolveInfo> list = getPackageManager().queryIntentActivities(intent, 0);
int size = list.size();
if (size == 0) {
    Toast.makeText(context, "Can not find image crop app", Toast.LENGTH_SHORT).show();
    return;
} else {
    intent.setData(mImageCaptureUri);
    mImageCaptureUri2 = Uri.fromFile(new File(Environment.getExternalStorageDirectory(),
        "tmp_avatar_" + String.valueOf(System.currentTimeMillis()) + ".jpg"));
    intent.putExtra(android.provider.MediaStore.EXTRA_OUTPUT, mImageCaptureUri2);
    intent.putExtra("name", "output", value: 200);
    intent.putExtra("name", "output", value: 200);
    intent.putExtra("name", "aspect", value: 1);
    intent.putExtra("name", "aspect", value: 1);
    intent.putExtra("name", "scale", value: true);
    intent.putExtra("name", "return-data", value: true);
    if (size == 1) {
        Intent i = new Intent(intent);
        ResolveInfo res = list.get(0);
        i.setComponent(new ComponentName(res.activityInfo.packageName, res.activityInfo.name));
        startActivityForResult(i, CROP_FROM_CAMERA);
    } else {
        for (ResolveInfo res : list) {
            final CropOption co = new CropOption();
            co.title = getPackageManager().getApplicationLabel(res.activityInfo.applicationInfo);
            co.icon = getPackageManager().getApplicationIcon(res.activityInfo.applicationInfo);
            co.appIntent = new Intent(intent);
            co.appIntent.setComponent(new ComponentName(res.activityInfo.packageName, res.activityInfo.name));
            cropOptions.add(co);
        }
        CropOptionAdapter adapter = new CropOptionAdapter(getApplicationContext(), cropOptions);
    }
}
```

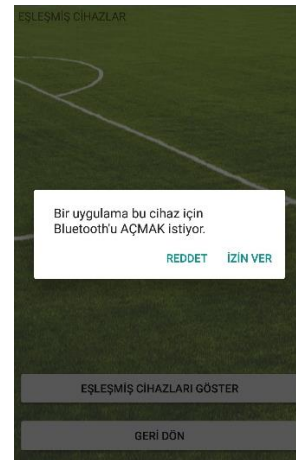
(Şekil 19.2)



(Şekil 19.3)



(Şekil 19.4)



(Şekil 19.5)