

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



PROJENİN ADI:

RASPBERRY Pİ İLE AKILLI EV OTOMASYONU

TASARIM PROJESİ

259139 SAMET DİNÇER

259185 MEHMET ENES HATUN

259187 SEFA DURMUŞ

2015-2016 GÜZ DÖNEMİ

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

PROJENİN ADI:

RASPBERRY Pİ İLE AKILLI EV OTOMASYONU

TASARIM PROJESİ

259139 SAMET DİNÇER

259185 MEHMET ENES HATUN

259187 SEFA DURMUŞ

Bu projenin teslim edilmesi ve sunulması tarafımdan uygundur.

Danışman : YRD. DOÇ. DR. GÜZİN ULUTAŞ

2015-2016 GÜZ DÖNEMİ



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırimcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

Üniversite döneminde edindiğimiz teknik ve yazılımsal deneyimleri pratik de uygulayabilmek mühendisliğin önemli bir gayesidir. Bu amaçla üniversite hayatı boyunca yapmış olunan projeler gelecek dönemde ki iş hayatı için bir ön hazırlık olarak katkı sağlar. Tasarım Projesi sayesinde Bilgisayar Mühendisliği alanında önemli iki dal olan Bilgisayar Yazılımı ve Donanımı dallarını bir arada bulundurduğundan önemli bir tecrübe edinilmesini sağlar. “**RASPBERRY Pİ İLE AKILLI EV OTOMASYONU**” projesi ile kullandığımız bazı sensörler sayesinde dış dünyadan alınan verilerin internet üzerinden iletilerek çalışan bir proje tasarımı gerçekleştirildi.

Gerçekleştirilen projenin başlangıcından itibaren sonuna kadar her safhasında bize yol gösteren , projenin yapılmasında fikirleriyle bizi teşvik eden çok değerli hocamız **YRD. DOÇ. DR. Güzin ULUTAŞ** ‘a teşekkür ederiz.

Proje’nin gerçekleştirilmesin de karşılaşılan sıkıntılarda yardımını hiç esirgemeyen arkadaşlarımıza çok teşekkür ederiz. Ayrıca üniversite hayatımız boyunca bizlerden desteklerini hiçbir zaman ayırmayan ailelerimize gönülden saygı ve sevgilerimizi sunarız.

SAMET DİNÇER

MEHMET ENES HATUN

SEFA DURMUŞ

TRABZON 2015

İÇİNDEKİLER

ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	VI
1-GİRİŞ ve GENEL BİLGİLER	1
1.1 Giriş.....	1
1.2 Raspberry Pi Kurulum ve Genel Ayarlarının Yapılması.....	2
1.2.1 Raspberry Pi Nedir ?.....	2
1.2.2 Raspberry Pi İşletim Sistemi Kurulumu ve İnternet Bağlantısı	2
2- SENSÖRLER'İN KONTROLLERİ ve UYGULAMALARI	3
2.1 HC-SR501 Ayarlanabilir IR Hareket Algılama Sensör'ü – PIR.....	3
2.1.1 PIR Sensörü.....	3
2.1.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması.....	4
2.2 DHT11 Sıcaklık ve Nem Sensörü.....	5
2.2.1 DHT11 Sıcaklık ve Nem Sensörü.....	5
2.2.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması.....	6
2.3 Su Seviye Sensör'ü (Water Level Sensörü)......	7
2.3.1 Su Seviye Sensör'ü7	7
2.3.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması.....8	8
2.4 Işık Sensör'ü (LDR- Light Dependent Resistors).....9	9
2.4.1 Işık Sensör'ü9	9
2.4.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması.....10	10

2.5	Servo Motor Kontrol'ü.....	11
2.5.1	Servo Motor	11
2.5.2	Board Üzerine Devrenin Kurulması ve Kodunun Yazılması	12
2.6	Hareket , Led , Buzzer ve Buton Kontrol'ü.....	13
2.6.1	Buzzer Kontrolü.....	13
2.6.2	Board Üzerine Devrenin Kurulması ve Kodunun Yazılması.....	14
2.7	Işık Sensörü Değerine Göre Pencere Kontrolü.....	15
2.7.1	Board Üzerine Devrenin Kurulması ve Kodunun Yazılması.....	16
3-	ANDROİD ARAYÜZ ve WEB SAYFASI TASARIMI	17
3.1	Android Studio da Webview Arayüzünün Tasarımı	18
3.2	Uygulamanın Android Cihazda ki Görüntüleri.....	19
4-	MYSQL VERİTABANI	20
4.1	MYSQL Workbench de Tabloların Oluşturulması.....	21
4.2	Veritabanının Web Sayfası ile Bağlantısının Yapılması.....	22
5-	SONUÇLAR	26
6-	ÖNERİLER	27
7-	KAYNAKLAR	27
	STANDARTLAR VE KISITLAR	28

ÖZET

Bu proje de günümüzde herkesin teknolojiyi aktif olarak kullanması göz önünde bulundurularak ev ortamında bulunan lambaların durumunu , evin o anki ısı durumunun , ev de herhangi bir hareket olup olmadığının kontrolü , siz evde yokken evde oluşabilecek su basmasına yönelik evdeki su seviyesinin öğrenilmesi vb. gibi birçok olayların önceden durumlarının bilinmesi ve gerekirse kontrolleri sağlanılmalıdır. Akıllı ev otomasyonunda Raspberry Pi kiti ile bazı sensörlerin Python programlama dili kullanılarak kodlarının yazılması ve projenin Android platformda çalışması amaçlanmıştır.

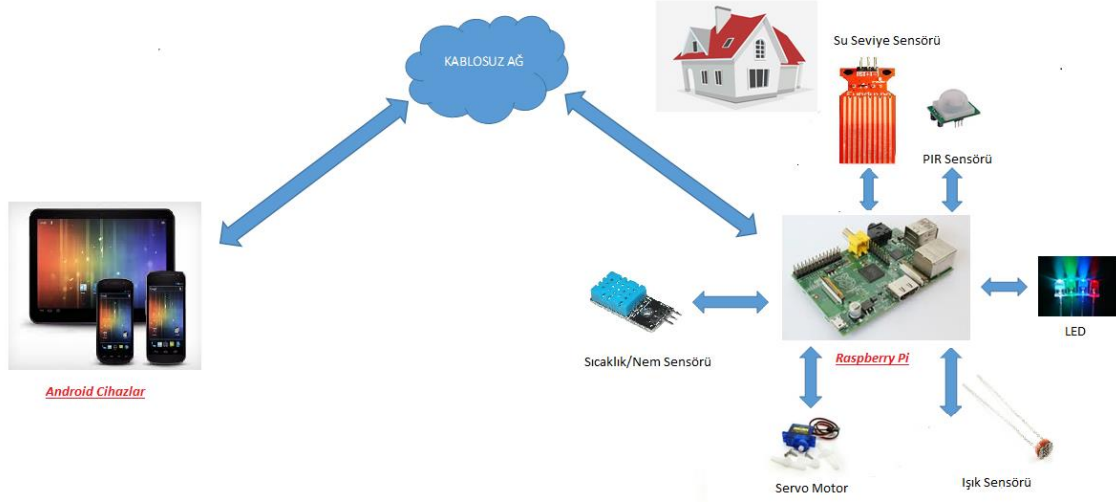
İki bölümden oluşacak projenin ilk bölümünde Raspberry Pi ile sensörlerin kontrolleri , tanımlamaları , bağlantıları ve Python kodları yer almaktadır , ikinci bölümde ise Android ile beraber sensörlerin çalışması denetlenmiştir.

Sonuç olarak Raspberry Pi kullanımı , sensörlerin kit üzerin de kontrolü , Android kodlamanın nasıl yapıldığının öğrenilmesi , ASP.NET ile web sayfası geliştirme öğrenildi.

.

1-GİRİŞ ve GENEL BİLGİLER

1.1 Giriş

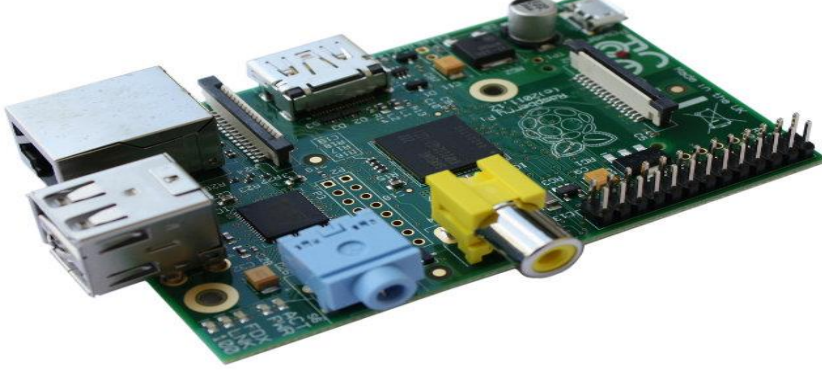


Şekil 1: Proje Diyagramı

Şekil 1’de ki diyagramda görüldüğü gibi "Akıllı Ev Projesi " ev de kullanılan sensörlerin durumlarına bağlı olarak onlar için yazılmış programları Android cihaz üzerinden kontrol edilebilmekte. Akıllı denilmesinin nedeni tasarlanan projenin üretilebilirlik , enerji tasarrufu ve yönetilebilirlik açısından ciddi kazançlar elde edilmesini sağlamaktadır. Bütün akıllı ev sistemlerinin amacı insanların hayatını kolaylaştırmaktır. Günümüzde nüfusun çoğunluğu aktif olarak teknolojinin içindedirler. Bundan dolayı her şeyi bulundukları yerden kontrol edebilme isteği oluşuyor. Bu nedenle ev otomasyon sistemleri geliştirilmiştir. Akıllı sistemler , kullanıcıya teknik konulara hakim olmadan projenin bütün özelliklerini kullanma ve kontrol edebilme imkanı sağlar. Ayrıca güvenliği sağlamak ve tehlikeli bir durum olduğunda kullanıcıyı uyarabilir. Bu çalışmada insanların hayatını kolaylaştıracak bir sistem tasarlanması hedeflenmiştir.

1.2 Raspberry Pi Kurulum ve Genel Ayarlarının Yapılması

1.2.1 Raspberry Pi Nedir ?



Şekil 2: Raspberry Pi

Şekil 2’de gösterilen Raspberry Pi, ARM1176JZF-S 700 MHz merkezi işlem birimini içeren Broadcom BCM2835 mikroçipi üzerine kurulmuştur. VideoCore 4 GPU grafik kartına ve 512 megabayt RAM e sahiptir. Kart üzerinde herhangi bir ROM veya SSD hard disk olmayıp, sistem boot etme ve uzun dönemli depolamada çözüm olarak SD kart kullanımı düşünülmüş ve bir SD kart okuyucu PCB ' nin arkasına yerleştirilmiştir. SD kart vasıtasıyla Linux işletim sistemini Raspberry Pi'ye kurabilirsiniz. Raspberry Pi Model B , üzerinde Linux dağıtımlı işletim sistemleri çalışmaktadır. Python programlama dili ile programlanabildiği gibi C , Java ve Perl programlama dilleri de kullanılabilir. Rahat kullanımı ve uygun fiyatından dolayı tercih edilmektedir.

1.2.2 Raspberry Pi İşletim Sistemi Kurulumu ve İnternet Bağlantısı

Raspberry pi için işletim sistemi olarak Debian tabanlı Raspian işletim sistemi <https://www.raspbian.org/RaspbianImages> internet sitesinden indirildi. 8 GB’lık hafıza kartına Raspian imajı yazıldı. Hafıza kartı Raspberry pi’ye takıldı. Wi-fi adapter kullanarak internet bağlantısı gerçekleştirildi.

2- SENSÖRLER'İN KONTROLLERİ ve UYGULAMALAR

2.1 HC-SR501 Hareket Algılama Sensör'ü – PIR

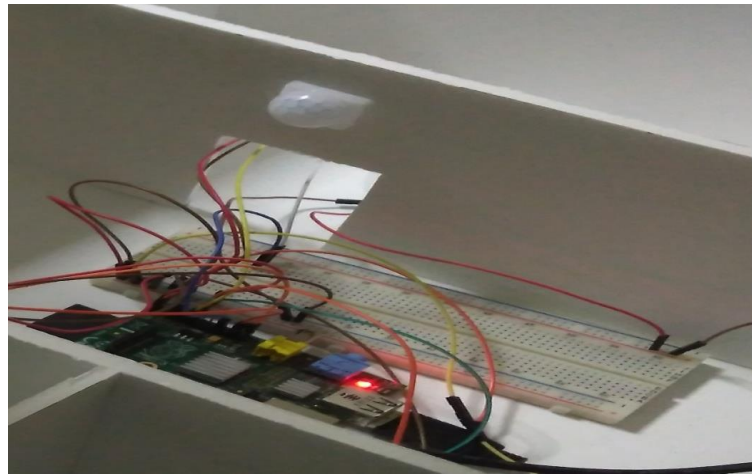
2.1.1 PIR Sensörü



Şekil 3: Hareket Algılama Sensörü

Hareket sensörleri, bir ortamda oluşan canlı hareketini algılamak için kullanılan sensörlerdir. Dijital çıkışlı olan modül, ortamda hareket olmadığı zaman lojik 0 , hareket olduğu zaman ise lojik 1 çıkışı vermektedir. Genellikle hareket sensörü hırsız alarmları , otomatik aydınlatma gibi sistemlerde kullanılır. Sensör çalışır durumda iken ve güvenlik aktif ise evde oluşacak her hareket de alarm sistemi devreye girecektir ve kullanıcının sisteminde evde hareket var uyarısı verilecektir. Butona basılması durumunda alarm devre dışı kalacak. Eğer güvenlik aktif değilse alarm çalmayacaktır.

2.1.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması



Şekil 4: Hareket Algılama Devresi

Şekil 4’de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi ve Python ile kodu yazıldı. Kod yazıldıktan sonra test edildi.

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.IN)
GPIO.setup(3, GPIO.OUT)
while True:
    i=GPIO.input(11)
    if i==0:
        print "HAREKET ALGILANMADI",i
        time.sleep(0.1)
    elif i==1:
        print "HAREKET ALGILANDI!!!",i
        time.sleep(0.1)
```

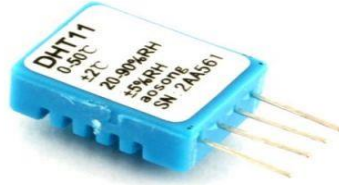
Şekil 5: Hareket Algılama Sensör Kodu

```
pi@raspberrypi:~ $ sudo python hareket.py
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANMADI 0
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
HAREKET ALGILANDI!!! 1
```

Şekil 6: Hareket Algılama Sensör Testi

2.2 DHT11 Sıcaklık ve Nem Sensörü

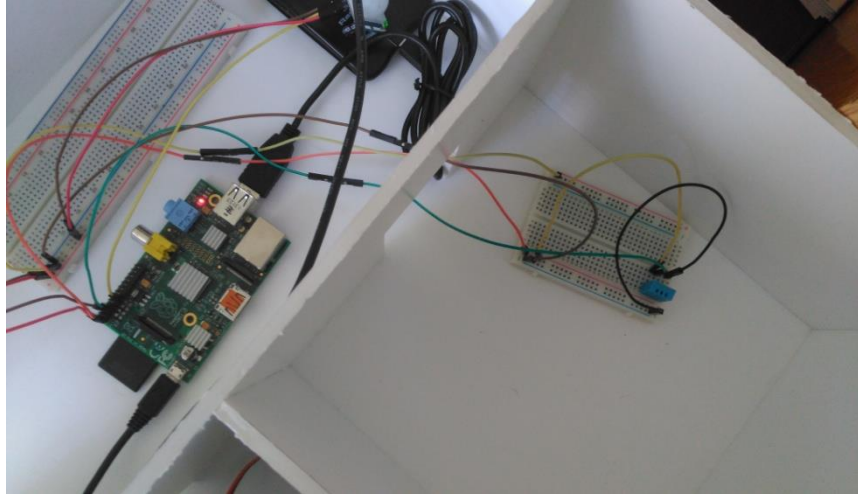
2.2.1 DHT11 Sıcaklık ve Nem Sensörü



Şekil 7: DHT11 Sıcaklık ve Nem Sensörü

DHT11 Sıcaklık ve Nem Sensörü dijital sinyal çıkışı veren sıcaklık ve nem ölçümü yapabilen bir modüldür. Kaliteli ölçüm kapasitesine sahip olan sensör uzun dönem çalışmalarda dengeli çıktılar vermektedir.. 0 ile 50°C arasında 2°C hata payı ile sıcaklık değeri ölçmektedir.

2.2.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması



Şekil 8: DHT11 Sıcaklık/Nem Sensörü

Şekil 8’de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi. Fakat bu sensörü diğer sensörler ile çalıştırdığımız zaman programın

yavaşlamasına neden oluyor. Ayrıca DHT11 sensörü python kodu ile yavaş çalışmakta bu yüzden bu sensör için C kodu yazıldı. Kod yazıldıktan sonra test edildi.

```
// -----
// DHT11 sensorunu oku
// -----
if (read_dht11_dat()) {

    // Sıcaklık veya nem değişmiş mi?
    if ((last_temp != dht11_dat[2]) || (last_rh != dht11_dat[0])) {

        // SQL sorgusunu çalıştır
        sprintf (sql, "update sicaklik set isi=%d , nem=%d where idsicaklik=1", dht11_dat[2], dht11_dat[0]);
        printf("%s\n", sql);
        if (mysql_query(conn, sql)) {
            fprintf(stderr, "%s\n", mysql_error(conn));
        }
        last_temp = dht11_dat[2];
        last_rh = dht11_dat[0];
    }
}

// Tekrar okumak için bekle
delay(1 * 1000 );
```

Şekil 9: Sıcaklık/Nem Sensörü Kodu

```
2015-12-12 11:26:13.126939
Sicaklik = 21.0*C
Nem Miktari = 46.0%
##### UPDATE ! ! ! #####
2015-12-12 11:26:15.186872
Sicaklik = 21.0*C
Nem Miktari = 46.0%
##### UPDATE ! ! ! #####
2015-12-12 11:26:22.306342
Sicaklik = 21.0*C
Nem Miktari = 47.0%
#####
2015-12-12 11:26:31.952288
Sicaklik = 21.0*C
Nem Miktari = 47.0%
#####
2015-12-12 11:26:34.008018
Sicaklik = 21.0*C
Nem Miktari = 46.0%
```

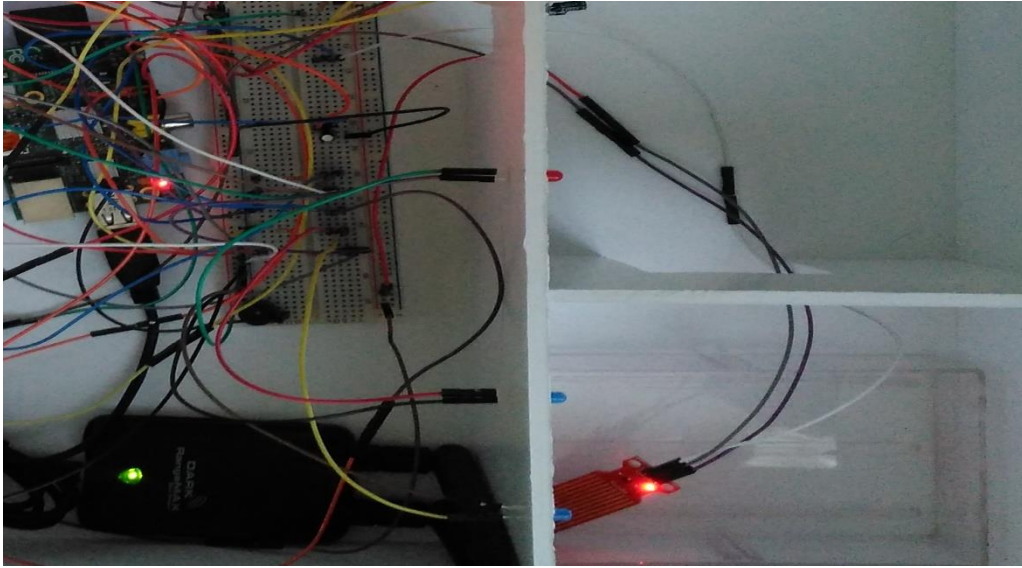
Şekil 10: Sıcaklık/Nem Sensörü Testi

2.3 Su Seviye Sensör'ü (Water Level Sensör))

2.3.1 Su Seviye Sensör'ü

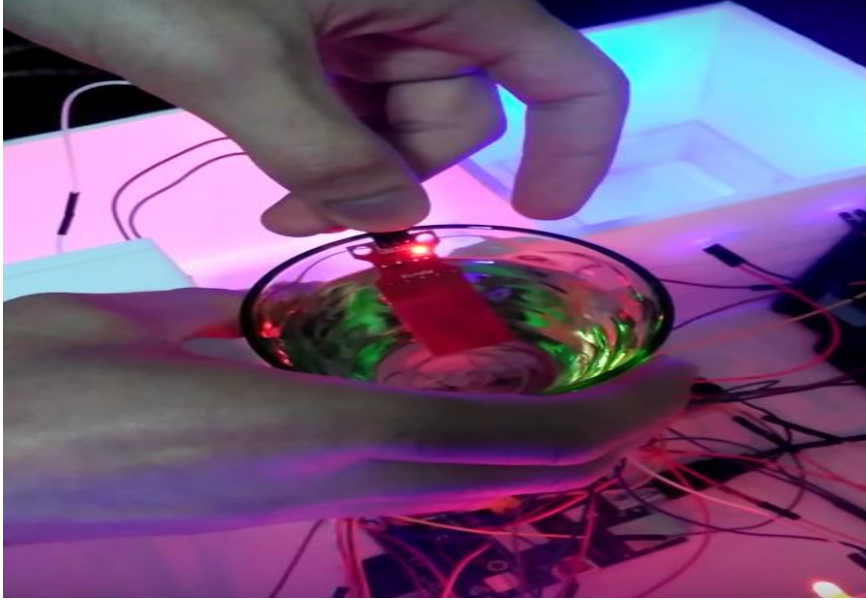
Su seviye sensörü sığ sularda ve yağmurlu havalarda biriken sularda kullanılabilir. Birbirine paralel olarak çekilmiş iletken direnç hatlarının su ile teması sonucu sensör çıkış pininde analog bir değer okunabilmektedir. Bir çok mikrodenetleyeci tarafından desteklenmektedir. Sensör kullanımı oldukça basittir. Besleme voltajı ve toprak bağlantısı yapılarak, sensör çıkış bacağından okuma yapılabilmektedir.

2.3.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması



Şekil 11: Su Seviye Sensörü

Şekil 11’de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi ve Python ile kodu yazıldı. Kod yazıldıktan sonra test edildi.



Şekil 12: Su Seviye Sensörü Testi

```
if (GPIO.input(10) == GPIO.HIGH) :
    cursor.execute("Update su Set baskin=1 where idsu=1 ")
    GPIO.output(8, GPIO.HIGH)
else(GPIO.input(10) == GPIO.LOW) :
    cursor.execute("Update su Set baskin=0 where idsu=1")
    GPIO.output(8, GPIO.LOW)
```

Şekil 13: Su Seviye Kodu

Kodu çalıştırdığınız zaman su seviyesi belirli direnç değerinin üstüne çıktığı zaman alarm çalmakta. Sensörü sudan çıkardığınız zaman alarm devre dışı kalmakta.

2.4 Işık Sensör'ü (LDR- Light Dependent Resistors)

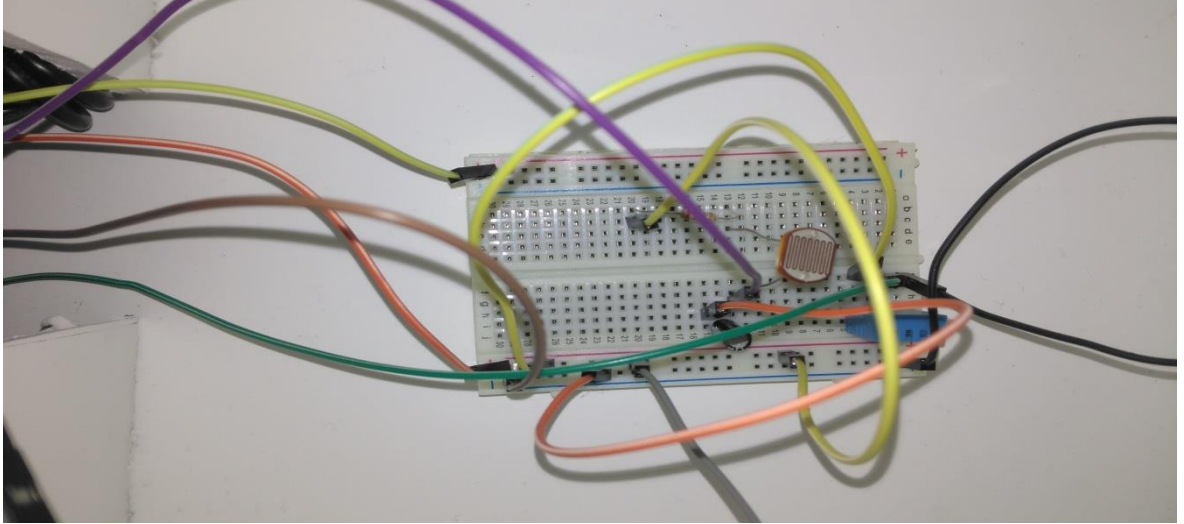
2.4.1 Işık Sensör'ü



Şekil 14: Işık Sensörü (LDR)

Bu sensör üzerine düşen ışığın miktarına bağlı olarak iki ucu arasındaki direnç değerinin değişimine göre çalışmaktadır. Işık sensöründen okunan değere göre pencere açma kapatma , evdeki ledi yakıp söndürme gibi uygulamalarda kullanılabilir.

2.4.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması



Şekil 15: Işık Sensörü

Şekil 15'de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi ve Python ile kodu yazıldı. Kod yazıldıktan sonra test edildi.


```

import RPi.GPIO as GPIO, time

GPIO.setmode(GPIO.BCM)

def Rctime (PiPin):
    deger = 0

    GPIO.setup(PiPin, GPIO.OUT)
    GPIO.output(PiPin, GPIO.LOW)
    time.sleep(0.1)

    GPIO.setup(PiPin, GPIO.IN)

    while (GPIO.input(PiPin) == GPIO.LOW):
        deger += 1

    return deger

while True:
    print Rctime(18)

```

Şekil 16: Işık Sensörü Kodu

```

pi@raspberrypi:~ $ sudo python ısıık2.py
ısıık2.py:8: RuntimeWarning: This channel is
  GPIO.setup(PiPin, GPIO.OUT)
323
298
324
315
317
309
316
308
317
317
323
294
326
328
314
327
320
327

```

Şekil 17: Işık Sensörü Testi

2.2 Servo Motor Kontrol'ü

2.5.1 Servo Motor

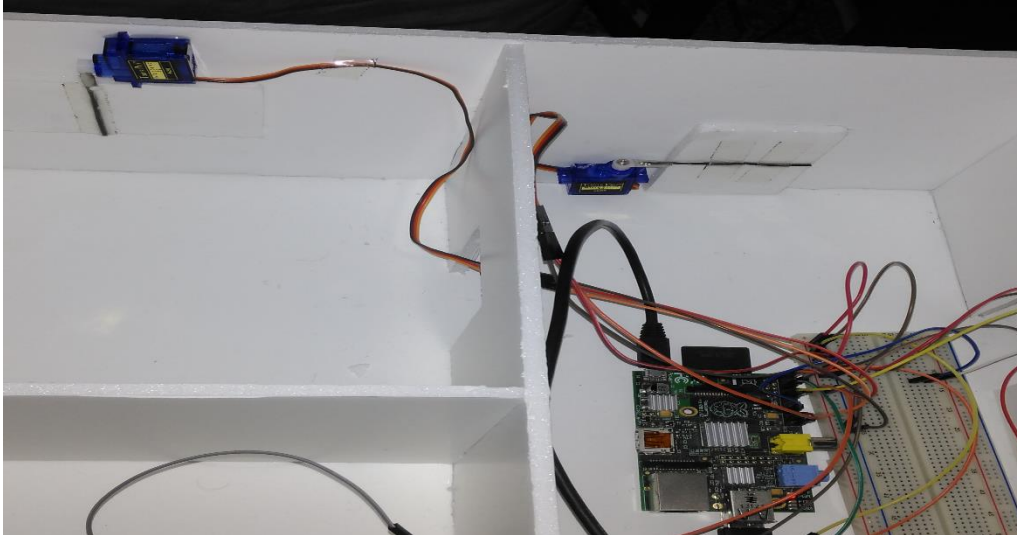


Şekil 18 : Servo Motor

Servo motorlar kullanım alanı olarak daha çok robotik uygulamalarda karşımıza çıkar. Biz tasarımıımızda servo motor ile kapı ve pencere kontrolünü gerçekleştirdik. Çalışma prensibi olarak verilen girdi değerine göre motorun dönme açısı ayarlanmaktadır. Ayrıca pencere için uygulamamızda *Sürekli Açık* ve *Sürekli Kapalı* butonları sayesinde pencerenin sürekli açık veya sürekli kapalı tutulması sağlanmıştır.

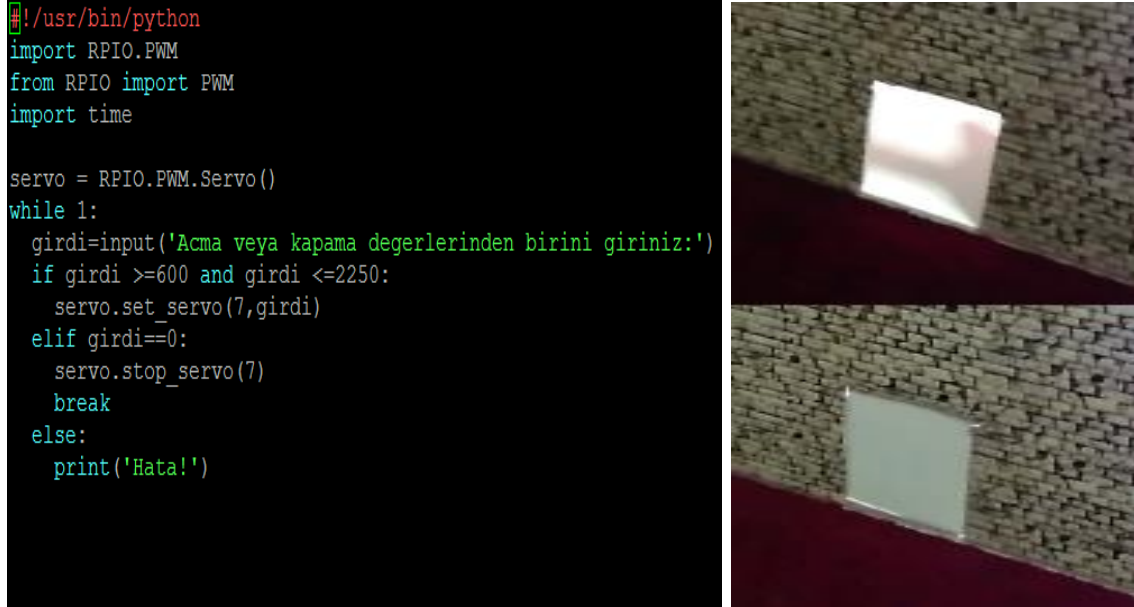
Kapı Aç	600	////	Kapı Kapat	1050
Pencere Aç	1800	/////	Pencere Kapat	1050

2.5.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması

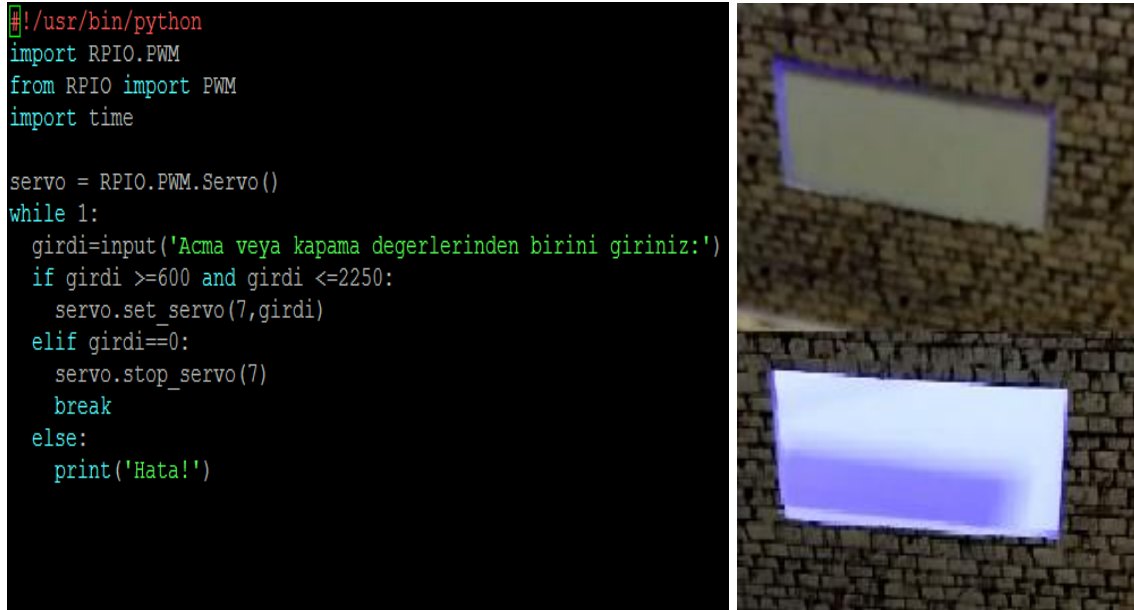


Şekil 19: Servo Motor Devresi

Şekil 19’de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi ve Python ile kodu yazıldı. Kod yazıldıktan sonra test edildi.



Şekil 20: Servo Motor Kapı Kodu ve Testi



Şekil 21: Servo Motor Pencere Kodu ve Testi

-12-

2.6 Hareket , Led , Buzzer ve Buton Kontrol'ü

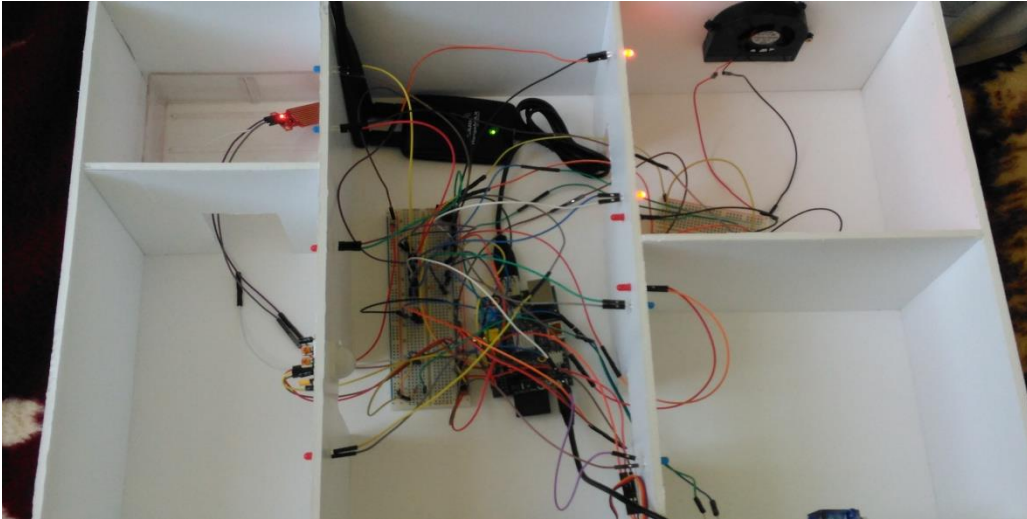
2.6.1 Buzzer Kontrolü



Şekil 22: Buzzer

Şekil 22’te gösterilen Buzzer modülü, üzerinde ses çıkışını sağlayan buzzer bulunan basit bir elektronik modüldür. Uyarı sistemlerinde veya farklı mikrodenetleyeci çalışmalarında kullanılabilir. I/O pini 5V veya 3.3V ile tetiklendiğinde buzzer ses çıkarmaya başlayacaktır. Bu modül görme engellileri uyarmak için kullanılabilir. Uygulamada ise güvenlik aktif ise evde herhangi bir hareket algılandığı zaman hem evde bulunan ledler sırayla yanmaya başlayacak hem de alarm çalmaya başlayacak. Butona basarak alarm devre dışı bırakılır.

2.6.2 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması



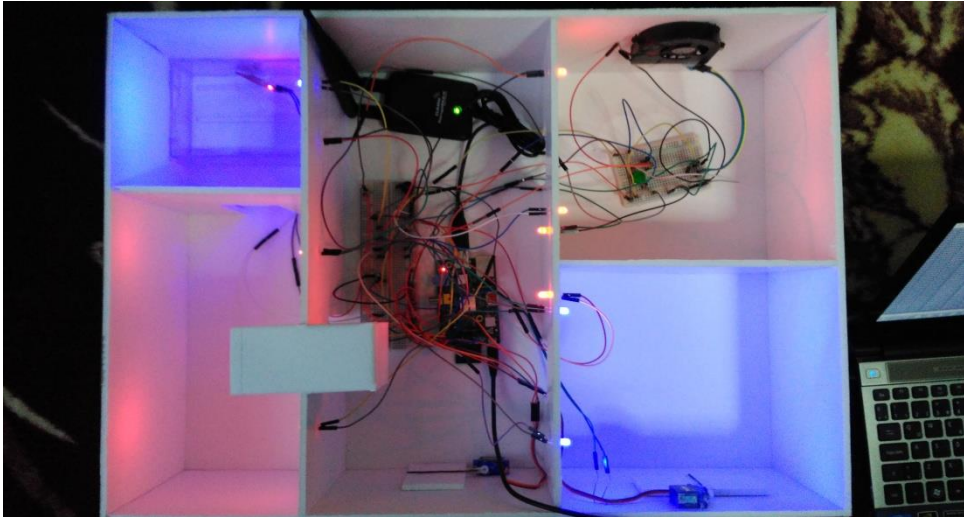
Şekil 23: Hareket + Buzzer + Led

-13-

Şekil 23’de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi ve Python ile kodu yazıldı. Kod yazıldıktan sonra test edildi.

```
if(guvenlik == ((1L,,))):
    if(hareket(17)):
        try:
            while 1:
                if ( GPIO.input(23) == False ):
                    GPIO.output(8,GPIO.LOW)
                    cursor.execute("Update motor set alarm = 0 where idmotor=1")
                    break
                else:
                    cursor.execute("Update motor set alarm = 1 where idmotor=1")
                    #Dot Dot Dot
                    GPIO.output(8,GPIO.HIGH)
                    GPIO.output(2,True)# 12 nolu pini +5v cikis verdik
                    if ( GPIO.input(23) == True ):
                        time.sleep(.1)
                    GPIO.output(2,False)#12 nolu pini 0v'a dusurduk.
                    GPIO.output(3,True)
                    GPIO.output(8,GPIO.LOW)
                    if ( GPIO.input(23) == True ):
                        time.sleep(.1)
                    GPIO.output(3,False)
                    GPIO.output(14,True)
                    GPIO.output(8,GPIO.HIGH)
                    if ( GPIO.input(23) == True ):
                        time.sleep(.1)
                    GPIO.output(14,False)#12 nolu pini 0v'a dusurduk.
                    GPIO.output(15,True)
                    GPIO.output(8,GPIO.LOW)
                    if ( GPIO.input(23) == True ):
                        time.sleep(.1)
                    GPIO.output(15,False)
                    GPIO.output(25,True)
                    GPIO.output(8,GPIO.HIGH)
                    if ( GPIO.input(23) == True ):
                        time.sleep(.1)
                    GPIO.output(25,False)
            except KeyboardInterrupt: #hata yakaladigimizda calisacak komut
```

Şekil 24 : Hareket + Buzzer + Led Kodu



Şekil 25: Hareket + Buzzer + Led Test

2.7 Işık Sensörü Değerine Göre Pencere Kontrolü

Kullanılan ışık sensöründen okunan ortamın ışık değerine göre pencere servo motor yardımıyla açılıp kapanmaktadır. Ayrıca uygulama da oda 5 de bulunan pencere kontrolü otomatik çalışsın seçeneğinin seçilmiş olmalı. Bu sayede kullanıcı istediği zaman pencereyi kontrol edebilmektedir. Sürekli açık seçeneği ile pencereyi sürekli açık tutabilir , sürekli kapalı kalsın seçeneği ile tamamen kapalı tutulabilir. Son olarak otomatik seçeneği ile ışık değerine bağlı olarak 1000 olarak belirlenmiş seviyeyi geçince pencere otomatik kapanmakta , 500 ile 1000 arasında bir değerde ise belli açıda açılmakta , son olarak 500 derecenin altına düştüğünde tamamen açılmaktadır.

2.7.1 Board Üzerine Devrenin Kurulması ve Kodunun Yazılması



Şekil 26: Işık Sensörü Değerine Göre Pencere Kontrolü

Şekil 26'de ki gibi board üzerine devre kuruldu ve Raspberry Pi ile bağlantısı gerçekleştirildi ve Python ile kodu yazıldı. Kod yazıldıktan sonra test edildi.


```

## pencere + ıřık
deger = Rctime(18)
cursor.execute("Select oto from motor where idmotor=1")
oto = cursor.fetchall()

if(oto == ((1L,),)):
    if(deger < 500):
        servo.set_servo(7,1800)
    elif (deger >= 500 and deger <= 1000 ):
        servo.set_servo(7,1400)
    else:
        servo.set_servo(7,1050)
elif(oto == ((0L,),)):
    cursor.execute("Select pencere from motor where idmotor=1")
    pencere = cursor.fetchall()
    if(pencere == ((1L,),)):
        servo.set_servo(7,1800)
    elif(pencere == ((0L,),)):
        servo.set_servo(7,1050)

```

řekil 27 : ıřık Sensörü Deęerine Göre Pencere Kontrolü Kodu

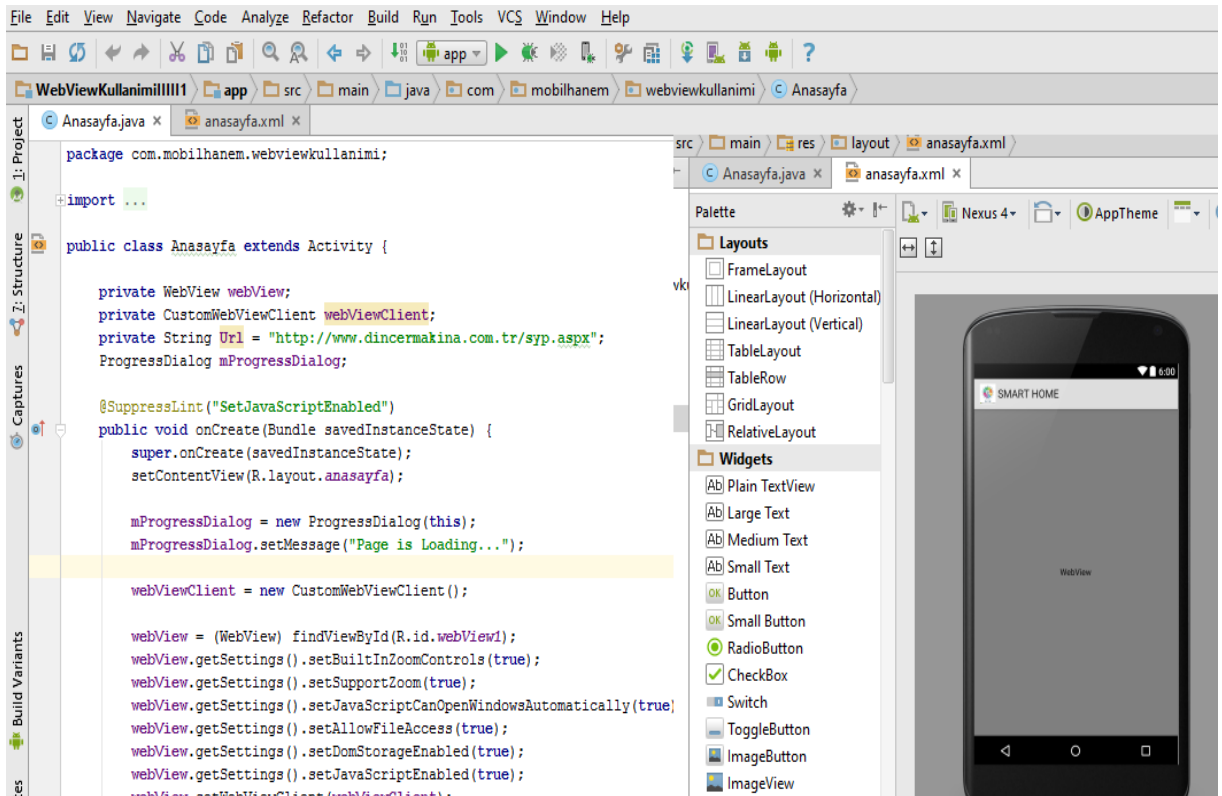


řekil 28 : ıřık Sensörü Deęerine Göre Pencere Kontrolü Testi

3- ANDROID ARAYÜZ ve WEB SAYFASI TASARIMI

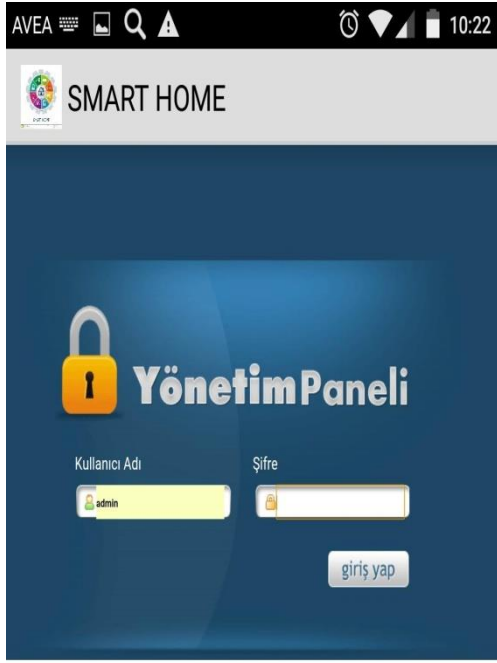
3.1 Android Studio da Webview Arayüzünün Tasarımı

Projede Android programlama için Android Studio programı kullanıldı. Android Studio, android OS a sahip cihazlar a uygulama geliştirmek için kullanılan bir uygulamadır. Telefon arayüzünden uygulamanın bulunduğu web sayfasına bağlanmak için bir tane web view uygulaması yazıldı. Bu sayede bilgisayardan girmenize gerek kalmıyor. Uygulama sizi direk o sayfaya yönlendirmekte.

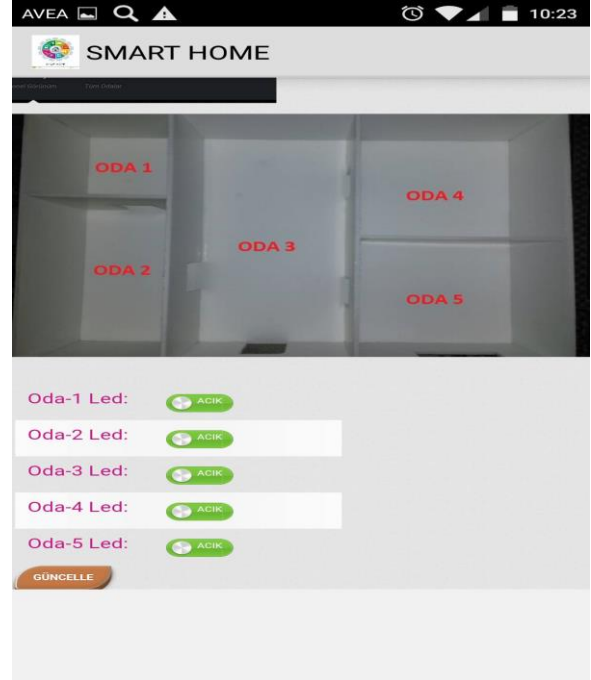


Şekil 29 : Web View Arayüzü Tasarımı

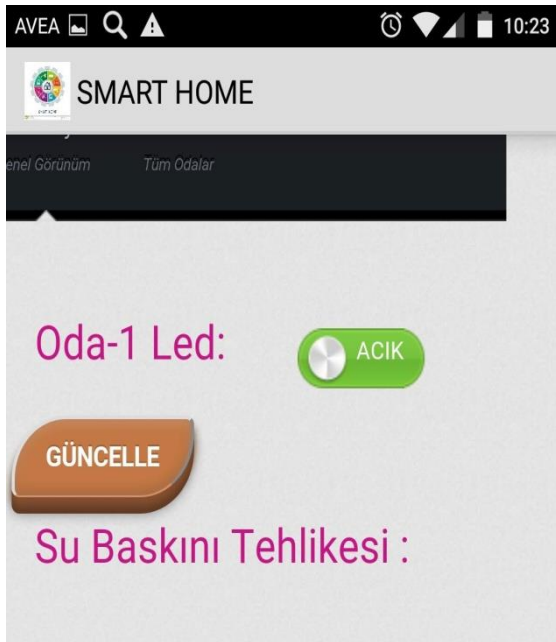
3.2 Uygulamanın Android Cihazda ki Görüntüleri



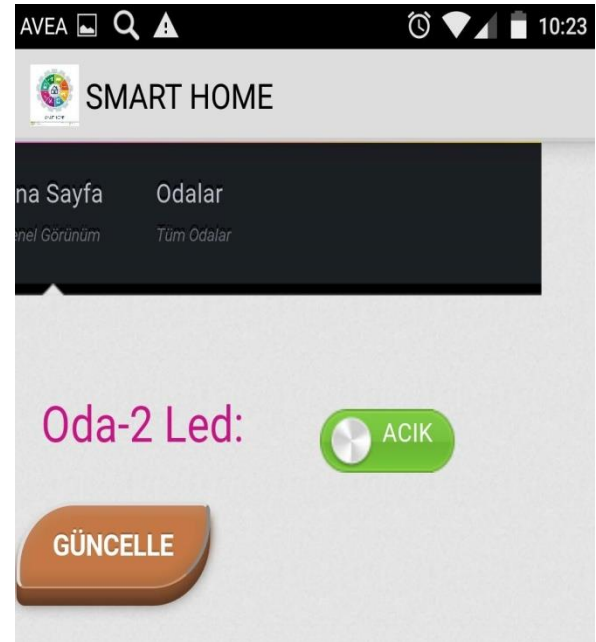
Şekil 30 : Kullanıcı Giriş Kısmı



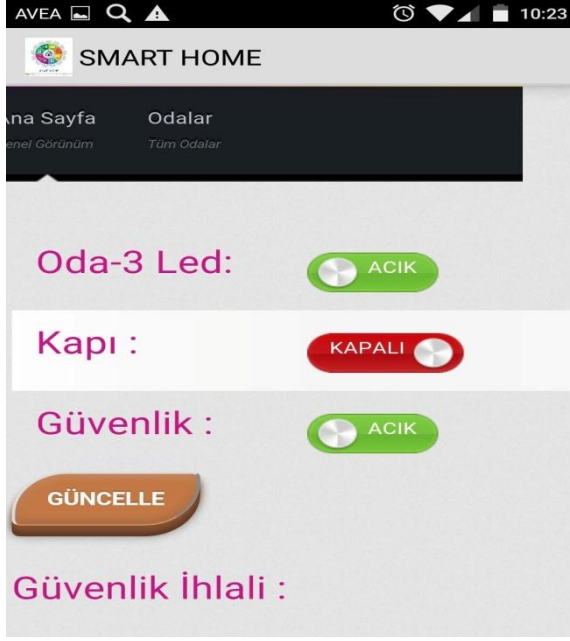
Şekil 31 : Uygulama Ana Sayfa



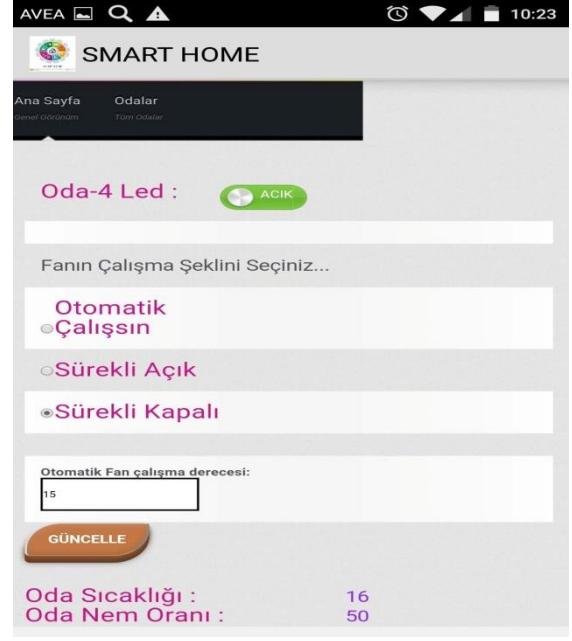
Şekil 32: Oda-1 Kontrolleri



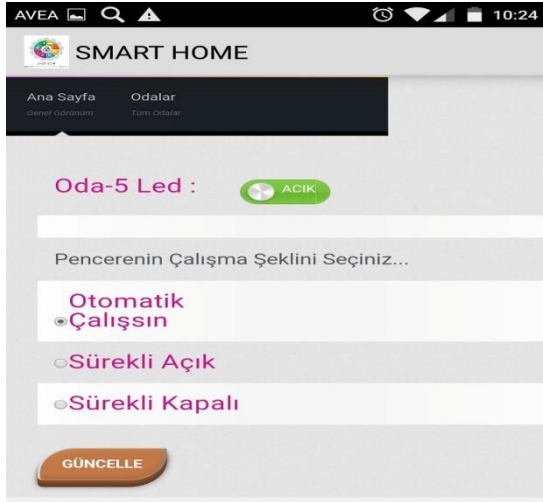
Şekil 33: Oda-2 Kontrolleri



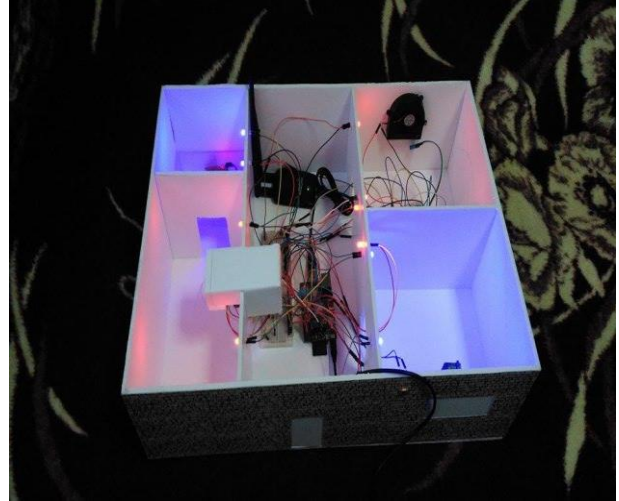
Şekil 34: Oda-3 Kontrolleri



Şekil 35: Oda-4 Kontrolleri



Şekil 36: Oda-5 Kontrolleri



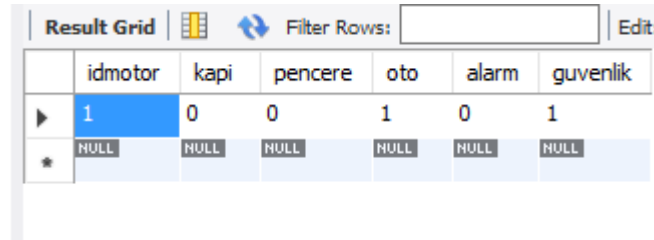
Şekil 37: Genel Görünüm

4- MYSQL VERİTABANI

MYSQL multithread (çoklu iş parçacıklı) , multiuser (çok kullanıcı) , hızlı ve sağlam bir veritabanı yönetim sistemidir. UNIX , OS/2 , WINDOWS gibi işletim sistemleri için ücretsiz dağıtılmaktadır. Ancak Linux altında diğer platformlara göre daha hızlı çalışmaktadır. Web sunucularında en çok kullanımı tercih edilen veritabanıdır . ASP PHP gibi birçok web programlama dili ile kullanılabilir.

Akıllı Ev Projesinde de Raspberry Pi dan okunan değerler veritabanına yazılmaktadır. Ve herhangi bir değişim anında sürekli veritabanı güncellenmektedir. Web uygulamamız üzerinden de veritabanına bağlanıp güncel durumlar görülebilir. Kullanıcı isterse bazı durumları değiştirebilir.

4.1 MYSQL Workbench de Tabloların Oluşturulması



	idmotor	kapi	pencere	oto	alarm	guvenlik
▶	1	0	0	1	0	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 38 : Servo Motor ve Alarm Sistemi Kontrol Tablosu

Şekil 38‘ de ki **motor** adında ki veritabanı tablosu kapı ve pencerelere takılmış servo motorların kontrolü , **pencere** sütunun aldığı değer eğer 1 olarak değiştirilirse (uygulamada pencerenin olduğu oda 5 de ki sürekli açık değerine denk gelmektedir.) pencere sürekli açık konumda kalacaktır. Değer 0 olarak ayarlanırsa pencerenin konumu sürekli kapalı konumuna gelecektir. Eğer **oto** değişkeni 1 ‘ e setlenirse ışık sensöründen okunan ışık değerine göre pencerenin konumu ayarlanmaktadır.

Güvenlik sütunu evde ki hareket sensörü ile birlikte çalışmaktadır. Eğer güvenlik durumu açık ise evde ki hareket sensöründen herhangi bir hareketlilik algılandığı durumunda web uygulamasına uyarı gelmektedir ve aynı zaman da evde ki alarm sistemi devreye girmektedir. **Güvenlik** aktif değilse evde alarm sistemi devreye girecek fakat uygulamaya uyarı gelmeyecektir.

	idsicaklik	isi	nem	fan	oto	derece
▶	1	16	50	0	0	15
★	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 39 : Sıcaklık/Nem ve Fan Kontrolü

DHT11 sıcaklık sensöründen okunan değerler kontrol edilmekte. Tabloda ki **isi** ve **nem** değerleri sıcaklık ve nem değerlerine karşılık gelmekte. Uygulama da fan 3 farklı değere sahip. Sürekli açık durumu ile hep çalışır durumda olması , sürekli kapalı durumu ile hep kapalı kalsun durumu sağlanmış oldu. **Oto** değişkeni ise sıcaklık sensöründen okunan değer eğer derece değişkeni ile setlenen değer üstüne çıkarsa fanın otomatik çalışması , eğer altında kalırsa da otomatik kapanması sağlanmıştır.

	idsu	baskin
▶	1	0
★	NULL	NULL

Şekil 40 : Su Seviye Kontrol Tablosu

Uygulamada aynı zaman da evde siz olmadığınız durumda su taşkını olup olmadığının kontrolü yapılabilenmektedir. Ve evde hem alarm çalmakta hemde web uygulamasında uyarı vermektedir.

Result Grid						
	idled	led1	led2	led3	led4	led5
▶	1	1	1	1	1	1
★	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 41 : Led Kontrol Tablosu

Kullanıcı evde bulunmadığı zaman uygulama sayesinde evde ki lambaların hangisinin açık olup olmadığını yada hangisini açmak istediğini kontrol edebilmekte.

4.2 Veritabanının Web Sayfası ile Bağlantısının Yapılması

4.2.1 Oda-1 İçin Gerekli Bağlantının Kurulması

Oda 1 için gerekli Su Baskını var veya yok durumlarının değerlendirilmesi ve gerekli veritabanı bağlantılarının kurulması gerçekleştirildi.

```

MySQLCommand komut = new MySqlCommand("Select baskin from su where idsu=1", baglanti);
MySQLCommand komut2 = new MySqlCommand("Select led1 from led where idled=1", baglanti);
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd = komut.ExecuteReader();
dtrd.Read();
string a;
a = dtrd["baskin"].ToString();

if (a == 1.ToString())
{
    Label1.Text = "VAR-ALARM DURUMU";
}
else
    Label1.Text = "YOK";
baglanti.Close();
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd2 = komut2.ExecuteReader();
dtrd2.Read();
a = dtrd2["led1"].ToString();

if (a == 1.ToString())
{
    cb_oda1.Checked = true;
}
else
    cb_oda1.Checked = false;

dtrd.Close();
dtrd2.Close();
baglanti.Close();

```

Şekil 42 :Su Baskını kontrolü Veritabanı Bağlantısı

4.2.2 Oda-2 İçin Gerekli Bağlantının Kurulması

Oda 2 Led Kontrolü yapılması için gerekli veritabanı bağlantısı kurulması gerçekleştirildi.

```

MySQLCommand komut = new MySQLCommand("Select led2 from led where idled=1", baglanti);
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd = komut.ExecuteReader();
dtrd.Read();
string a;
a = dtrd["led2"].ToString();

if (a == 1.ToString())
{
    cb_oda2.Checked = true;
}
else
    cb_oda2.Checked = false;

dtrd.Close();
baglanti.Close();

```

Şekil 43: Led Kontrolü Veritabanı Bağlantısı

4.2.3 Oda-3 İçin Gerekli Bağlantının Kurulması

Oda 3 Güvenlik seçeneği aktif durumda iken hareket algılandığında alarm sistemi devreye girmektedir. Butona basıldığında alarmsistemi devre dışı bırakmak için gerekli veritabanı bağlantısı kurulması gerçekleştirildi.

```

MySQLCommand komut = new MySQLCommand("Select leds from led where idled=1", baglanti);
MySQLCommand komut2 = new MySQLCommand("Select kapi,alarm,guvenlik from motor where idmotor=1", baglanti);
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd = komut.ExecuteReader();
dtrd.Read();
string a,b,c;
a = dtrd["led5"].ToString();

if (a == 1.ToString())
{
    cb_oda3.Checked = true;
}
else
    cb_oda3.Checked = false;

baglanti.Close();
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd2 = komut2.ExecuteReader();
dtrd2.Read();
a = dtrd2["kapi"].ToString();
b = dtrd2["alarm"].ToString();
c = dtrd2["guvenlik"].ToString();

if (a == 1.ToString())
{
    cb_kapi.Checked = true;
}
else
    cb_kapi.Checked = false;
if (b == 1.ToString())
{
    Label1.Text = "VAR-Alarm Durumu!";
}
else
    Label1.Text = "YOK";
if (c == 1.ToString())
{
    cb_guvenlik.Checked = true;
}
else
    cb_guvenlik.Checked = false;

dtrd.Close();
dtrd2.Close();

```

Şekil 44: Hareket Algılama ve Alarm Kontrolü Veritabanı Bağlantısı

4.2.4 Oda-4 İçin Gerekli Bağlantının Kurulması

Oda 4 Sıcaklık ve Nem değerleri göz önüne alınarak sıcaklık belirlenen değerin üzerine çıktığında oda içerisindeki fanın çalışması ve kontrolü için gerekli veritabanı bağlantısı kurulması gerçekleştirildi.

```

MySQLCommand komut = new MySQLCommand("Select led3 from led where idled=1", baglanti);
MySQLCommand komut2 = new MySQLCommand("Select isi,nem,fan,oto,derece from sicaklik where idsicaklik=1", baglanti);
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd = komut.ExecuteReader();
dtrd.Read();
string a, b, c,f,g ,l;
l = dtrd["led3"].ToString();

if (l == 1.ToString())
{
    cb_oda4.Checked = true;
}
else
    cb_oda4.Checked = false;

baglanti.Close();
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd2 = komut2.ExecuteReader();
dtrd2.Read();
a = dtrd2["isi"].ToString();
b = dtrd2["nem"].ToString();
c = dtrd2["fan"].ToString();
f = dtrd2["oto"].ToString();
g = dtrd2["derece"].ToString();

Label1.Text = a;
Label2.Text = b;
TextBox1.Text = g;
if (f == 1.ToString())
{
    RadioButton1.Checked = true;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
}
else {
    if (c == 1.ToString())
    {
        RadioButton1.Checked = false;
        RadioButton2.Checked = true;
        RadioButton3.Checked = false;
    }
    else
    {
        RadioButton1.Checked = false;
        RadioButton2.Checked = false;
        RadioButton3.Checked = true;
    }
}

dtrd.Close();
dtrd2.Close();
baglanti.Close();

```

Şekil 45: Sıcaklık Değerine Göre Fan Kontrolü Veritabanı Bağlantısı

4.2.4 Oda-5 İçin Gerekli Bağlantının Kurulması

Oda 5 LDR ışık sensörü ortam aydınlık(gündüz) ya da ortam karanlık (gece) ise percerenin açılması veya kapatılması için gerekli veritabanı bağlantısı kurulması gerçekleştirildi.

```

MySQLCommand komut = new MySQLCommand("Select led4 from led where idled=1", baglanti);
MySQLCommand komut2 = new MySQLCommand("Select pencere,oto from motor where idmotor=1", baglanti);
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd = komut.ExecuteReader();
dtrd.Read();
string a, b, l;
l = dtrd["led4"].ToString();

if (l == 1.ToString())
{
    cb_oda5.Checked = true;
}
else
    cb_oda5.Checked = false;

baglanti.Close();
if (baglanti.State == ConnectionState.Closed)
    baglanti.Open();

MySQLDataReader dtrd2 = komut2.ExecuteReader();
dtrd2.Read();
a = dtrd2["pencere"].ToString();
b = dtrd2["oto"].ToString();

if (b == 1.ToString())
{
    RadioButton1.Checked = true;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
}
else {
    if (a == 1.ToString())
    {
        RadioButton1.Checked = false;
        RadioButton2.Checked = true;
        RadioButton3.Checked = false;
    }
    else
    {
        RadioButton1.Checked = false;
        RadioButton2.Checked = false;
        RadioButton3.Checked = true;
    }
}

dtrd.Close();
dtrd2.Close();
baglanti.Close();

```

Şekil 46: Ortam Işığına Göre Pencere Kontrolü Veritabanı Bağlantısı

SONUÇLAR

- Raspberry Pi kullanımı ve işlevselliği kavrandı.
- Python programlama dili öğrenildi
- PIR(Hareket Algılama) sensörü testi yapıldı.
- DHT11(Sıcaklık ve Nem) sensörü testi yapıldı.
- Su seviye sensörü kontrol testi yapıldı.
- LDR(Light Dependent Resistors) ışık sensörü testi yapıldı.
- Servo motor kontrol testi yapıldı.
- Led ve Buzzer etkileşimli alarm testi yapıldı.
- Android Studio da Webview Arayüzünün Tasarımı yapıldı.
- Web sayfasının tasarımı yapıldı.
- MYSQL Workbench de tablolar oluşturuldu.
- Veritabanının Web Sayfası ile bağlantısı yapıldı.

ÖNERİLER

Projemizde uygulanması ve esnekliği bakımından Raspberry Pi kullandık. Bu modülün boyutunun küçük olması ve ayrı modül veya sensörlerin kullanımını sağlayan birçok pin bulundurması uygulama geliştirme bakımından tercih edilme sebeplerinden birkaçı olmuştur. Ayrıca modüle kurulan Linux tabanlı işletim sistemi sayesinde temel seviyede internette gezinme, oyun oynama, yazılım geliştirme, fotoğraf görüntüleme ve düzenleme , video izlemek ve müzik dinlemek mümkün olabiliyor.

KAYNAKLAR

- [1] <https://www.raspberrypi.org/forums/viewtopic.php?f=32&t=98140>
- [2] <http://forum.raspi.gen.tr/konu-proje-raspberry-pi-ile-isi-ve-nem-sensoru-dht11-kullanimi>
- [3] <http://diyhacking.com/raspberry-pi-gpio-control/>
- [4] <http://lombarditech.com/raspberry/simple-code-for-reading-the-temperature-sensor/>
- [5] <http://roboturka.com/2015/10/09/raspberry-pi-ile-pir-hareket-sensoru/>
- [6] <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-12-sensing-movement/software>
- [7] http://pi.cs.man.ac.uk/download/Raspberry_Pi_Education_Manual.pdf

STANDARTLAR ve KISITLAR FORMU

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Akıllı ev otomasyonu projesinin gerçekleşmesi için Raspberry Pi modülünden faydalanılmıştır. Bu sayede evin farklı bölümlerinin amaca uygun sensörlerle kontrol edilmesi hedeflenmiştir. Akıllı ev otomasyonu yeni bir proje olmamasına rağmen kullanımı ve fonksiyonellik açısından geleceği olan bir yönelim alanı olmaya adaydır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Projenin tasarlanması ve sensörlerin kullanılmasında bazı zorluklarla karşılaşılmış ve bu sorunların çözümüne ulaşılmıştır. Örnek olarak DHT11(Sıcaklık ve Nem) sensörünün Python dilinde çalışmasında ki sorunların onun yerine C dili ile yazılarak giderilmesi sağlanmıştır.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Hazırladığımız projenin yazılımın yanı sıra donanımı ilgilendiren kısımlarının olması sebebiyle Elektronik devre bilgisi ve yorumlanması için Bilgisayar Mimarisi ve Elektrik Devreleri derslerinden faydalanılmıştır. Ayrıca bu projenin tasarımı boyunca PYTHON programlama dilinden kullanılmıştır.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

Mühendislik standartları ele alındığında tasarlanan proje kapsamında uygulamanın web üzerinden internet aracılığı ile bir çok platformdan çalıştırılması sağlamıştır. Bu kapsamda PYTHON, Android Studio, ASP. NET ve MYSQL Workbench kullanılmıştır.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

Tasarımın en önemli ekonomik yükü kullanılan Raspberry Pi kitinin maliyetidir.

b) Çevre sorunları:

Gerçeklenen projenin çevreye olumsuz bir etkisi bulunmamaktadır.

c) Sürdürülebilirlik:

Günlük hayatımızda teknolojinin hızla gelişmesi ve insan yaşamına olumlu etkileri etkin bir şekilde hissedilmektedir. Akıllı ev otomasyonu bu teknolojik gelişmeler sayesinde insan yaşamını kolaylaştıran sürdürülebilir bir alandır.

d) Üretilirlik:

Birçok firma akıllı ev otomasyonu ilgilenmekte bu yüzden üretilebilirlik açısından önemli bir alandır.

e) Etik:

Tasarlanan projenin etik açıdan bir sakıncası yoktur.

f) Sağlık:

Çevremizdeki mobil telefonlar, TV vb. cihazların etkisi ile yayılan radyasyona bakıldığında uygulamasını geliştirdiğimiz tasarım sağlık açısından bir sakıncası bulunmaktadır.

g) Güvenlik:

Uygulamanın internet tabanlı çalışması güvenlik açısından çeşitli problemler oluşturabilir.

h) Sosyal ve politik sorunlar:

Tasarımın herhangi bir sosyal ve politik bir sorunu bulunmamaktadır.