

PROJE RAPORU

PROJENİN ADI:

Uzaktan Akıllı Kontrol ve Bilgilendirme Sistemleri: SMARTURKOD

ÖZET

Son yıllarda oldukça gelişen teknoloji otomatik kontrol sistemleri üzerinde de etkili olmuştur. Teknolojinin gelişmesi uzaktan erişimi oldukça kolay hale getirmiştir. İnsanların rahat, huzurlu, mutlu yaşayabilmelerini sağlamak amacıyla akıllı ev otomasyonu günümüzde oldukça ilgi görmektedir. Akıllı ev fikri 1800'lerde ortaya çıkarak teknolojinin ilerlemesiyle birlikte uygulanmaya başlanmıştır. Ancak bu teknolojinin evlerimize girebilmesi uzun yıllar almıştır. 1934-1939 yılları arasında yapılan dünya fuarlarında elektrikli evler sergilenirken, 1966 yılında Jim Sutherland, "ECHO-IV" isimli bir ev otomasyon sistemi geliştirmiştir. Mikroişlemcilerin bulunması ile birlikte düşük fiyata yapılabilen otomasyon sisteminin yapı hizmet sektöründeki kullanımı hızlanmıştır.

Her sistemin avantajları olacağı gibi akıllı ev sistemlerinin de avantajları vardır.

- Akıllı ev otomasyonunun en büyük avantajı güvenliğimizin sağlanmasıdır.
- Kontrolü sağlanan sistemler sayesinde daha rahat bir yaşam sürdürülmesini sağlar.
- Zaman ve enerji tasarrufu sağlar.
- Engelli hastaların daha rahat yaşamlarını sürdürmelerini sağlar.
- İnsanlara konfor sağlar.
- Acil durumlarda uyarı verir, gerekli işlemleri yapar.

Bu projede de, maliyeti düşük ve kolay kullanılabilir Arduino tabanlı bir akıllı ev sistemi tasarlanmıştır. Akıllı ev sisteminin kontrol ve denetleme mekanizması bulunmaktadır. C#, Bluetooth, Wi-Fi ve GSM ile hem kontrol hem de denetleme yapılmaktadır. Evin sıcaklığından, hırsız bilgisine kadar veri alımı mümkündür.

Projede çeşitli modüller kullanılmıştır: Arduino MEGA, DHT11 Isı ve sıcaklık modülü, ESP8266 İnternet modülü, HC-05 Bluetooth modülü, GSM modülü, MQ-2 Yanıcı gaz sensörü, Alev sensörü, PIR Hareket sensörü, Jumper, 4 kanallı röle, Buzzer, LED, IR LED, Mikrofon kartı, LDR. Projenin temel amacı bu modüller ve ikincil sistemlerin (bilgisayar, telefon, LCD) eşzamanlı çalışması ve kullanıcıyı bilgilendirip, kontrol yetkisi vermesidir. İlerleyen dönemlerde daha sistematik ve daha geniş bir sistem yapmak mümkündür.

İÇİNDEKİLER:

1) Projenin amacı

2) Giriş

3) Yöntem

- a. Arduino MEGA
- b. DHT11 Isı ve sıcaklık modülü
- c. ESP 8266 Internet modülü
- d. HC-05 Bluetooth modülü
- e. GSM modülü
- f. MQ-2 Yanıcı gaz sensörü
- g. Alev sensörü
- h. PIR Hareket sensörü
- i. Jumper
- j. 4 kanallı röle
- k. Buzzer zil
- l. LED
- m. IR LED
- n. Mikrofon kartı
- o. LDR
- p. Proje çalışma mantığı
- q. Bilgisayar uygulaması ve C# geliştirme ortamı
- r. Pushetta Bildirim Sistemi
- s. MIT © AppInventor 2 ©
- t. Arduino geliştirme ortamı
- u. Android uygulaması
- v. Proje Kodları

4) Proje Görselleri ve Projenin Yapım Aşamaları

5) Sonuç ve Öneriler

6) Kaynakça

1) PROJENİN AMACI:

Küresel çapta kullanılan akıllı ev sistemlerinin maliyetli ve çok zor kullanılıyor olmasından dolayı daha uygun ve daha ergonomik bir yapıda akıllı ev sistemi arayışına girmek ve yaygın sistemlerin dışında da akıllı ev teknolojisine katkı sağlamak amaçlanmıştır. Projemiz kapsamında Arduino ile geliştirilen sistem her an değişmeye ve gelişmeye uygun bir yapıdadır.

2) GİRİŞ:

Herkes evini uzaktan kontrol etmek ve evindeki durumu uzak bir yerden görüntülemek ister. Gerek Android cihazdan, gerek de bilgisayar ile bu durum sağlanabilmektedir. Bu sırada akıllı ev sistemi dediğimiz sistemler devreye girmektedir. Akıllı ev sistemleriyle ev kontrol edilebilmekte, evin durumu çeşitli kontrol mekanizmalarına iletilmektedir. Olası acil durumlarda da gerekli işlemler yapılabilmektedir.

Bu projede Arduino'nun kullanılma amacı, hem çeşitli modüller ve sensörlerle bu işlemi sağlayabilmemiz, hem de Arduino'nun geliştirilmeye açık yapısıdır. Ayrıca, çoğu web sitesinde Arduino hakkında sayfalarca kaynak bulunmaktadır. Arduino'nun kolay kullanımı da bu projeyi Arduino ile yapmamızdaki büyük etkenlerden biridir.

“Akıllı Ev” fikri ilk olarak 1980'lerin başında ortaya çıktı. İlk uygulamalarda, sıradan her hangi bir fiziksel engeli olmayan insanların ev konforu düşünülmüştü [6]. Üreticiler, bir sistemi bütününe entegre etmeyi amaçlamamışlardı. Ancak, bireysel olarak kontrol edilebilecek birçok ürün çeşidi ortaya çıktı. (enerji kontrol ünitesi, güvenlik sistemi, ışık kontrolcüler vs.) Günümüzde ise, ev teknolojileri, bizlerin hiç de yabancı olmadığı bir konular haline gelmiştir. Mikserler, kahve makineleri ve mutfak robotları, televizyonlar, müzik setleri, DVD oynatıcılar, videolar; buzdolapları, çamaşır ve bulaşık makineleri, otomatik garaj kapıları, ışık seviyesi ayarlanabilir lambalar, telsiz telefonlar, elektrik süpürgeleri ve daha bunlar gibi birçok cihaz ile hayatımızın ayrılmaz bir parçası olmuştur.

Daha sonraları bu cihazlar, yine gelişen teknolojiyle hayatlarımızı kolaylaştırmak için birçok değişim geçirdi; televizyonlar, müzik setleri ve hatta garaj kapıları için uzaktan kumandalar; kahve makineleri için zamanlayıcılar, kullanıcıya birçok seçenek sunan çamaşır ve bulaşık makineleri, buzdolapları geliştirildi. Bu sürecin bir sonraki adımı, tüm evin tek bir noktadan kontrol edilmesine imkân veren ve programlama imkânlarıyla bu kontrolü kendiliğinden sağlayan ev otomasyon teknolojilerini tüketicilerin hizmetine sunmaktadır. 2010 yılı itibarıyla ise, evler yeni bir cihazla donatılacaklar: ev içerisindeki birçok fonksiyonu birleştiren bir kontrol işlemcisi. Projemiz kapsamında yapmış olduğumuz literatür çalışmalarında akıllı ev sistemlerinin tarihçesine ulaştık. Tablo 1'de gösterilmiştir.

Tarih	Açıklama	Kaynak
1993	Akıllı Ev Sistemleri: Bu çalışmada, ev otomasyonu ürünleri sınıflandırılmıştır. CEBus nokta mimarisi verilmiş ve bunun detaylı açıklaması yapılmıştır.	Christos Douligeris [1]
1995	Telefon Hatlarını Kullanarak Araçların Uzaktan PC İle Kontrolü : Bu çalışmada, telefon kabloları ve bir PC kullanarak ev otomasyonu fikri gösterilmiştir. Sistem, bir PC ile birçok aracın açıp kapanmasını bir kablo yoluyla gönderilen DTMF sinyalleri ile yapıyor. Sistemin donanım ve yazılımı telefon standartlarına dayanıyor. Sistemin donanım kısmında DTMF alıcısı, on/off Hook algılayıcısı, zil algılayıcısı ve bir giriş/çıkış PC arayüz kartından oluşuyor. Çalışmada bir aracın kontrolü örneklendirilmiştir. Yazılım kontrol programı kısmı ise Turbo Basic ile yazılmış ve akış çizelgeleri gösterilmiştir.	Baki Koyuncu [2]

1998	Ofis ve Ev için Telefonla Uzaktan Kontrol: Bu çalışmada, ofis ve ev için bir uzaktan kontrol sistemi telefon kullanarak dizayn edilmiş ve uygulanmıştır. Uzaktan kontrolün fonksiyonu, uzaktaki bir yere sağlanan gücü telefon kablosuyla kontrol etmektir. Sistem DTMF telefon sistemine dayalı çalışmaktadır. Sistem telefon tuş takımını, veri ve yorumların girildiği girdi aleti olarak kullanıyor.	İsmail Coşkun, Hamid Ardām [3]
2004	DTMF Alıcısının Hızlı Fourier Transform İle Analizi Ve Sinyal Algılaması: Bu çalışmada, performans değerlendirmesi ve DTMF alıcısının analizi için hızlı fourier transform algoritması kullanarak, yeni bir metot düşünülmüştür.	Min Ju Park, Sang Jin Lee, Dal Hwan Yoon [4]
2004	Akıllı Ev Araştırması: Bu çalışmada akıllı evin bir tanımı yapılmış, ve daha sonra akıllı evin elementleri, araştırma projeleri, akıllı ev network durumu, akıllı ev uygulamaları ve son olarak da doğruluğu açıklanmıştır.	Li Jiang, Da-You Liu, Bo Yang [5]

2004	Yaşlı Ve Fiziksel Engelli İnsanlar İçin Akıllı Ev: Akıllı ev, yaşlı ve engelli insanlar için iyi bir alternatif olarak düşünülmüştür. Bu sayede, hem hareket kolaylığı sağlanmakta hem de 24 saat sağlık kontrolü yapılabilmektedir.	Dimitar H. Stefanov, Zeungnam Bien, Won-Chul Bang [6]
2006	Vücut Hareketlerini Kullanarak Akıllı Ev Kontrolü: Bu çalışma, akıllı evdeki ışık ve perde gibi nesnelerin vücut hareketleri ile kontrolünü önermektedir.	Daehwan Kim, Daijin Kim [7]

Tablo 1. Akıllı ev sistemlerinin kronolojik sıralaması

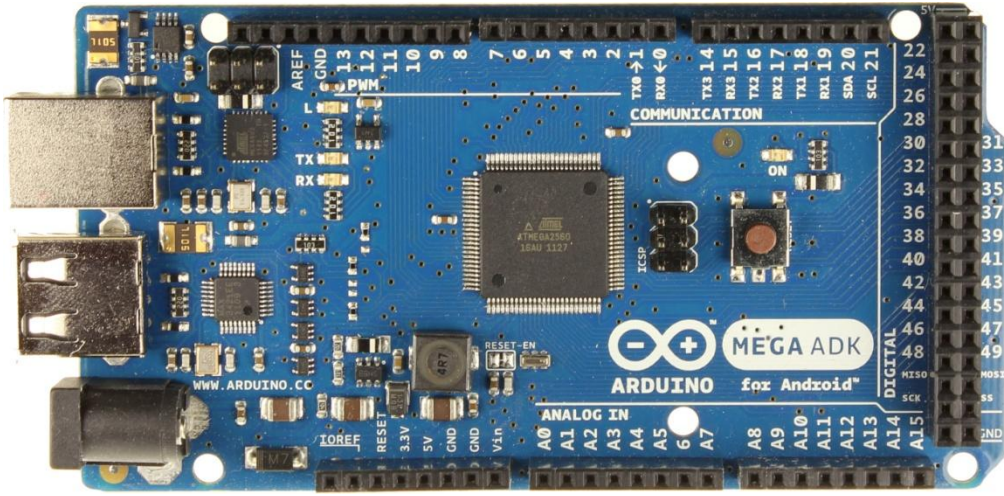
3) YÖNTEM:

Proje çerçevesinde geliştirilen akıllı ev sistemi, evin durumunu ve kritik bilgileri (hırsız, yangın, sel, elektrikli cihazların durumu gibi) Bluetooth, Wi-Fi ve seri portlar ile kullanıcıya iletmektedir. Bunlar da HC-05 Bluetooth modülü, DHT11 Isı ve sıcaklık modülü, ESP8266 Internet modülü, GSM modülü, MQ-2 Metan, bütan, duman ve yanıcı gaz sensörü, PIR hareket sensörü, alev sensörü, 4 kanallı röle, çeşitli jumperlar, bir Android cihaz ve Bluetooth Bağlantısı ile sağlanabilmektedir. Wi-Fi bağlantısı ve modülü sağlandığı takdirde dünyanın her yerinden evimiz kontrol edilebilmektedir.

Projenin temel parçaları aşağıdaki gibidir:

- Arduino MEGA Geliştirme Kartı
- DHT11 ısı ve sıcaklık modülü
- HC-05Bluetooth Modülü
- ESP8266 Internet Modülü
- GSM Modülü
- Alev sensörü
- PIR hareket sensörü
- MQ-2 Gaz sensörü
- Jumper Kablolar
- 4 Kanallı röle
- Buzzer Zil
- Çeşitli Ledler

a) **Arduino MEGA geliştirme kartı:**Arduino açık kaynak kodlu geliştirilebilir bir mikrodenetleyici kartıdır. ATMEL işlemcisini kullanan bu kartla elektronik ve robotik pek çok uygulamayı hayata geçirebiliriz. UNO, Mega, Leonardo, PRO Mini gibi pek çok çeşidi bulunur. Arduino IDE programıyla beraber kullanılır. Kartın üzerinde bulunan analog ve dijital girişler sayesinde analog ve dijital verileri rahatlıkla kullanabiliriz. Ayrıca sensörleri kullanarak da gelen verileri işleyebiliriz. Bu geliştirme kartı ile sensörlerden gelen verileri kullanabilmekteyiz. Bu verileri bilgisayar ortamında kaydedebilmemize imkan vermektedir. Bu kartı tercih etmemizin sebebi kolay bir şekilde çevresiyle etkileşime girebilmesi, analog ve dijital verileri işleyebilmesi ve internet üzerinde örnek projelerin yaygın olarak bulunmasıdır. Şekil 1’de projemizde kullandığımız Arduino MEGA kartı görülmektedir.

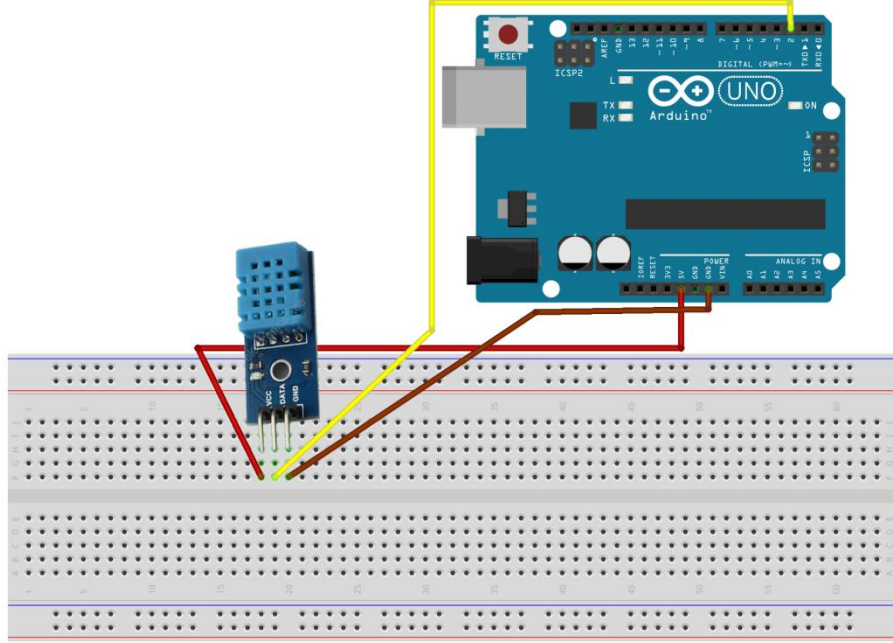


Şekil 1.Arduino MEGA geliştirme kartı.

Arduino’nun programlama dili C++’a çok benzemektedir. Kendine özgü kodlama diliyle programlanmaktadır. Kullandığımız ArduinoMEGA üzerinde 2 adet besleme ünitesi vardır. Biri USB, diğeri de harici güç girişidir. Ayrıca voltaj regülâtörü ve Atmega mikrodenetleyicisi ArduinoMEGA’nun temel parçalarındandır. Teknik Özellikleri:

- Mikrodenetleyici: ATmega2560
- Çalışma gerilimi : +5 V DC
- Tavsiye edilen besleme gerilimi: 7 - 12 V DC
- Besleme gerilimi limitleri: 6 - 20 V
- Dijital giriş / çıkış pinleri: 54 tane (15 tanesi PWM çıkışını destekler)
- Analog giriş pinleri: 16 tane
- Giriş / çıkış pini başına düşen DC akım: 40 mA
- 3,3 V pini için akım: 50 mA
- Flash hafıza: 256 KB (8 KB bootloader için kullanılır)
- SRAM: 8 KB
- EEPROM: 4 KB
- Saat frekansı: 16 MHz

b) DHT11 Isı ve Sıcaklık Modülü: Projede evin ısı ve sıcaklığını ölçmek için kullanılmıştır. Sensör üzerinde GND (toprak), VCC(5V giriş) ve çıkış pini bulunmaktadır. GND pini Arduino üzerindeki GND pinine, VCC pini Arduino üzerindeki 5V pinine, DATA pini ise Arduino üzerindeki herhangi bir digitalpinebaanlanması gerekmektedir. Şekil 2’de DHT11 ve şeması verilmiştir.

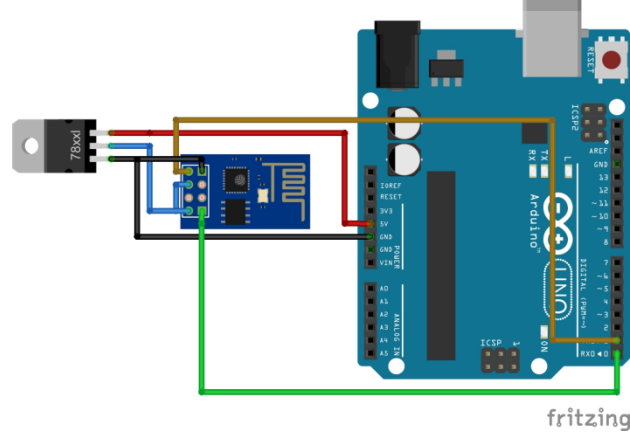


Şekil 2. DHT11 ve şema

Örnek kod aşağıdaki gibidir:

```
#include<dht11.h>
dht11 DHT11;
voidsetup()
{
  DHT11.attach(2);
  Serial.begin(9600);
}
voidloop()
{
  Serial.print("NEM (%): ");
  Serial.println((float)DHT11.humidity,DEC);
  Serial.print("SICAKLIK (°C): ");
  Serial.println((float)DHT11.temperature,DEC);
  delay(2000);
}
```

- c) **ESP8266 Internet Modülü:** Projede interneti kullanarak veri iletip almak için internetin gerekli olduğu durumlarda ESP8266 kullanılmıştır. Internet işlemleri projelerde en çok tercih edilen modüllerden biridir. Şekil 3'te ESP8266 ve şeması verilmiştir. ESP8266 ile dünyanın her noktasından kontrol mümkündür.



Şekil 3. ESP8266 ve şema

Örnek kod aşağıda verilmiştir.

```
#include<SoftwareSerial.h>

voidsetup()
{
  Serial.begin(9600);
  esp8266.begin(9600);
}

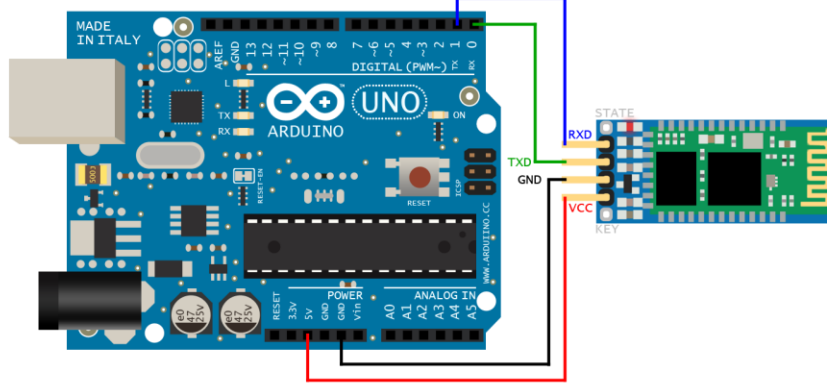
voidloop()
{
  if(esp8266.available()) {
    while(esp8266.available())
    {
      char c = esp8266.read();
      Serial.write(c);
    }
  }

  if(Serial.available())
  {
    delay(1000);

    Stringcommand="";

    while(Serial.available())
    {
      command+=(char)Serial.read();
    }
    esp8266.println(command);
  }
}
```


d) **HC-05 Bluetooth Modülü:** Android cihaz veya bilgisayar arasında Bluetooth kullanarak iletişim kurmak için gerekli modüldür. Modüldeki RX ve TX pinleri Arduino'ya ters bağlanmalıdır. 5V ile çalışır. Şekil 4'de HC-05 modül ve şeması verilmiştir.

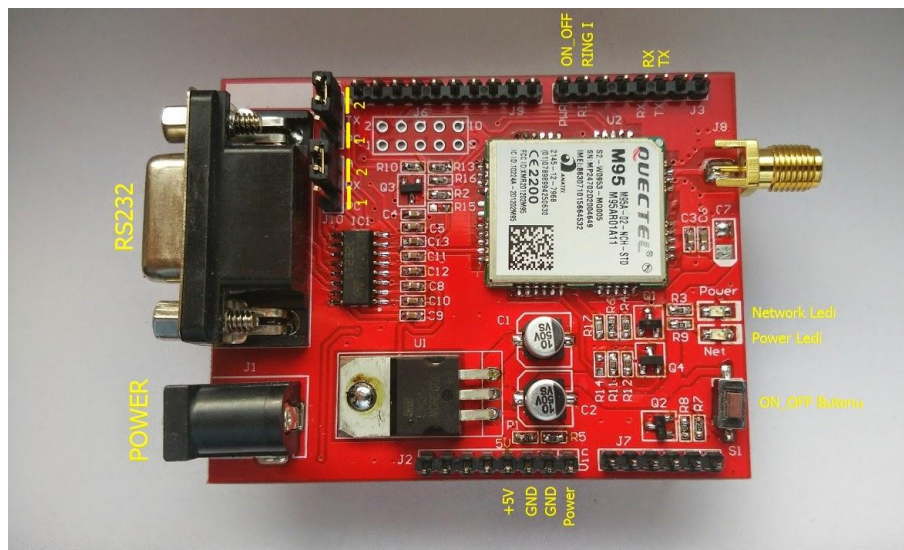


Şekil 4. HC-05 ve şema

Aşağıda örnek bir proje verilmiştir:

```
#include<SoftwareSerial.h>
SoftwareSerialmySerial(10, 11); // RX, TX
int led1 = 7;
int led2 = 6;
int led3 = 5;
int led4 = 4;
voidsetup()
{
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);
    mySerial.begin(9600);
    mySerial.println("LED uygulaması");
}
voidloop()
{
    charch=mySerial.read();
    if (ch=='q')
    {
        digitalWrite(led1,!digitalRead(led1));
        if (digitalRead(led1) ==HIGH)
        {
            mySerial.println("LED 1 yandı");
        }
        else
        {
            mySerial.println("LED 1 sondu");
        }
    }
}
```

e) **GSM Modülü:** SMS göndermek için GSM modülü kullanılmıştır.



Örnek kod aşağıdaki gibidir

```
#include <GSM.h>
#define PINNUMBER ""
GSM gsmAccess;
GSM_SMS sms;

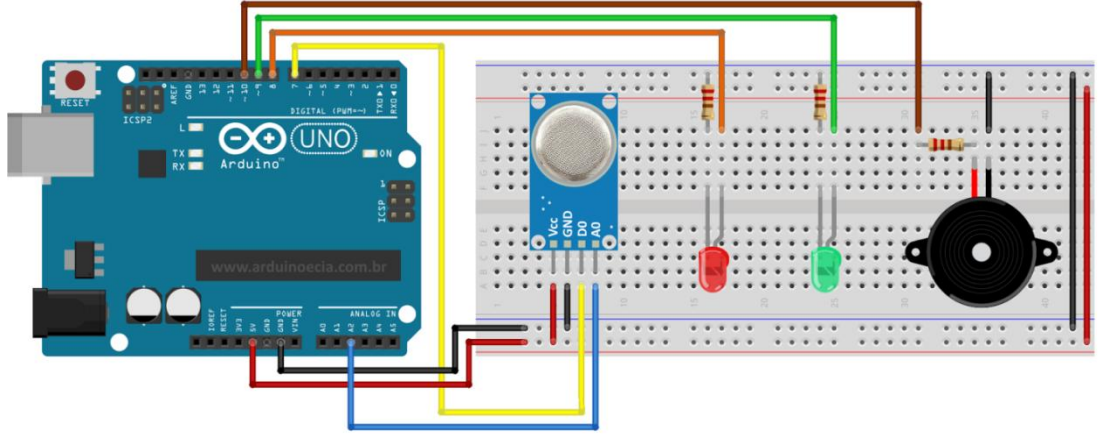
void setup() {
    Serial.begin(9600);
    while (!Serial) {}
    Serial.println("SMS Messages
Sender");
    boolean notConnected = true;
    while (notConnected) {
        if (gsmAccess.begin(PINNUMBER) ==
GSM_READY) {
            notConnected = false;
        } else {
            Serial.println("Not connected");
            delay(1000);
        }
    }
    Serial.println("GSM initialized");
}

void loop() {
    Serial.print("Enter a mobile number:
");
    char remoteNum[20]; // telephone
number to send sms
    readSerial(remoteNum);
    Serial.println(remoteNum);

    Serial.print("Now, enter SMS
content: ");
    char txtMsg[200];
    readSerial(txtMsg);
    Serial.println("SENDING");
    Serial.println();
    Serial.println("Message:");
    Serial.println(txtMsg);
    sms.beginSMS(remoteNum);
    sms.print(txtMsg);
    sms.endSMS();
    Serial.println("\nCOMPLETE!\n");
}

int readSerial(char result[]) {
    int i = 0;
    while (1) {
        while (Serial.available() > 0) {
            char inChar = Serial.read();
            if (inChar == '\n') {
                result[i] = '\0';
                Serial.flush();
                return 0;
            }
            if (inChar != '\r') {
                result[i] = inChar;
                i++;
            }
        }
    }
}
```

- f) **MQ-2 Yanıcı gaz sensörü:** Olası gaz sızıntısı vb. durumlarda kullanıcıyı bilgilendirmek için kullanılmıştır. Analog ve digital çıktı almak mümkündür. Şekil 6'te gaz algılandığında uyarı veren şema çizilmiştir.

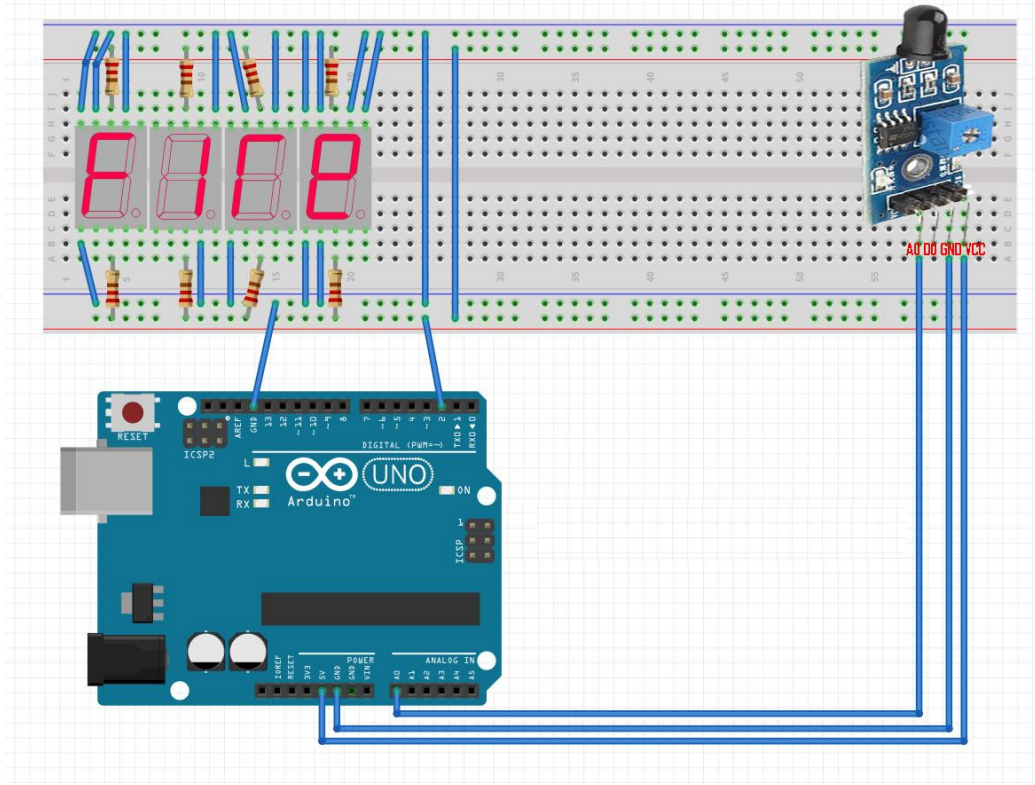


Şekil 6. Gaz algılandığında uyarı veren şema

Aşağıda örnek bir proje verilmiştir.

```
voidsetup()  
{  
  int zil = 9;  
}  
voidloop()  
{  
  floatsensorValue;  
  intdeger=500;  
  sensorValue=analogRead(A0);  
  if(sensorValue>deger)  
  {  
    digitalWrite(zil,HIGH);  
    delay(1000);  
    digitalWrite(zil,LOW);  
    delay(500);  
  }  
}
```

- g) **Alev sensörü:** Akıllı ev sistemi alev spektrumunu algılasa uyarı verecek ve gerekli işlemi yapacaktır. Şekil 5’te alev algılayınca yazı yazdıran devre verilmiştir.



Şekil 5. Alev algılayınca yazı yazdıran devre

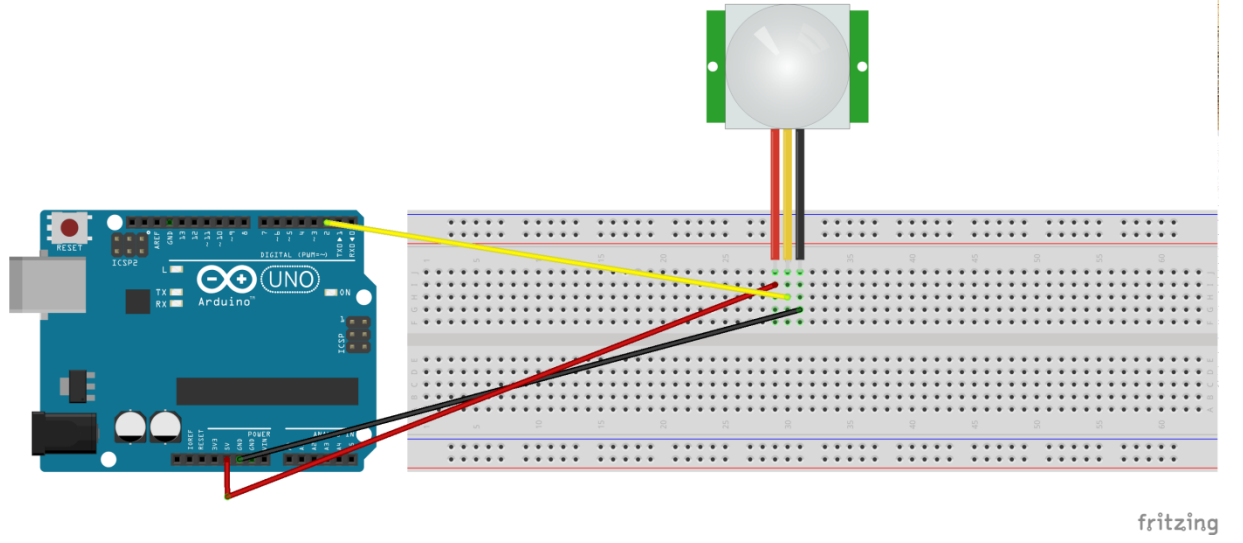
Örnek kod aşağıda verilmiştir:

```
const int sensorPin = 0;
const int pinSpeaker = 10;
int smoke_level;

void setup() {
  Serial.begin(115200);
  pinMode(sensorPin, INPUT);
  pinMode(pinSpeaker, OUTPUT);
}

void loop() {
  smoke_level = analogRead(sensorPin);
  Serial.println(smoke_level);
  if (smoke_level < 500) {
    playTone(300, 160); delay(150); } } void playTone(long duration, int freq) {
  duration *= 1000; int period = (1.0 / freq) * 1000000; long elapsed_time = 0;
  while (elapsed_time < duration) { digitalWrite(pinSpeaker, HIGH);
  delayMicroseconds(period / 2); digitalWrite(pinSpeaker, LOW);
  delayMicroseconds(period / 2); elapsed_time += (period); } }
```

h) **PIR Hareket sensörü:** Hırsız algılamada PIR sensör kullanılmıştır. Evlerdeki fotosel de PIR sensördür. Şekil 7’de sensörün şeması verilmiştir.



Şekil 7. PIR Hareket sensörü

Örnek kod aşağıda verilmiştir.

```
intpirPin= 3; // PIR pin
intledPin= 4; // LED pin
intdeger= 0;

voidsetup() {
  pinMode(pirPin,INPUT); // PIR Pin'i giriş yapılıyor
  pinMode(ledPin,OUTPUT); // LED Pin'i çıkış yapılıyor
  Serial.begin(9600); //Serial Porttan veri göndermek için
  baudrate ayarlanıyor.
}

voidloop(){
  deger=digitalRead(pirPin); // Dijital pin okunuyor
  Serial.println(deger); // Okunan değer seri porttan okunuyor.
  if (deger==HIGH) {
    digitalWrite(ledPin,HIGH); // Eğer okunan değer 1 ise LED yakılıyor.
  }
  else{
    digitalWrite(ledPin,LOW); // Eğer okunan değer 0 ise LED
    söndürülüyor.
  }
}
```

- i) **Jumper:** Arduinoyla sensörler, modüller ve breadboardı bağlamak için kullanılan bakır, dışı plastik kaplı kablolardır. Dişi erkek, erkek erkek ve dişi dişi şeklinde üç tipi bulunmaktadır. Şekil 8’te bir jumper örneği verilmiştir.



Şekil 8. Jumper örneği

- j) **4 Kanallı röle:** Televizyon, bilgisayar, lamba vs. elektrikli eşyaları çalıştırıp kapatmak için kullanılmaktadır. Şekil 9’da bir röle verilmiştir.



Şekil 9. Röle

Örnek proje

```
Voidsetup() {  
  int role=5;  
}  
voidloop() {  
  digitalWrite(5, LOW);  
  delay(3000);  
  digitalWrite(5, HIGH);  
  delay(3000);}
```

- k) **Buzzer zil:** Herhangi bir uyarı verileceğinde ses çıkartmak için buzzer ismi verilen ziller kullanılmaktadır. Güç, toprak ve çıkış pinleri bulunmaktadır. Şekil 10’da bir buzzer örneği verilmiştir.



Şekil 10. Buzzer.

- l) **Ledler:** Ledler projede gerekli alanlarda kullanılmıştır. Şekil 11’de kırmızı bir LED örneği verilmiştir.



Şekil 11. LED

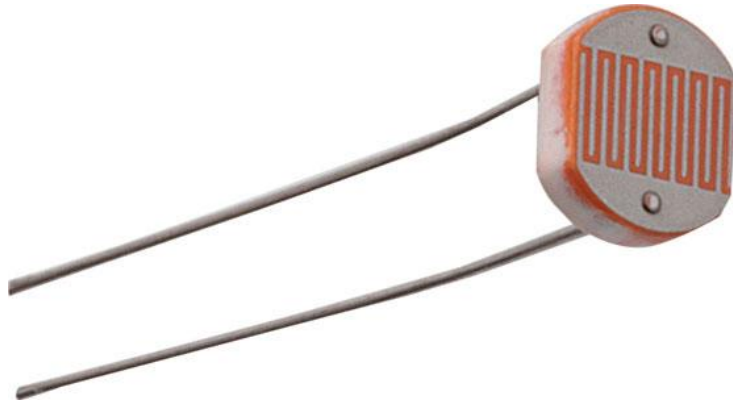
m) IR LED: Projede telefon ve bilgisayardan televizyonu kontrol etmek amacıyla oluşturulmuştur. IR sinyalleri gönderilmektedir. Telefon veya bilgisayardan gerekli komutu aldığı takdirde belirtilen sinyali gönderir.



n) Mikrofon kartı: Alkış sesi gibi yüksek seslerde lambaların yanması için kullanılmıştır.



o) LDR: Işık seviyesi düşünce lambaları yakmak için kullanılmıştır

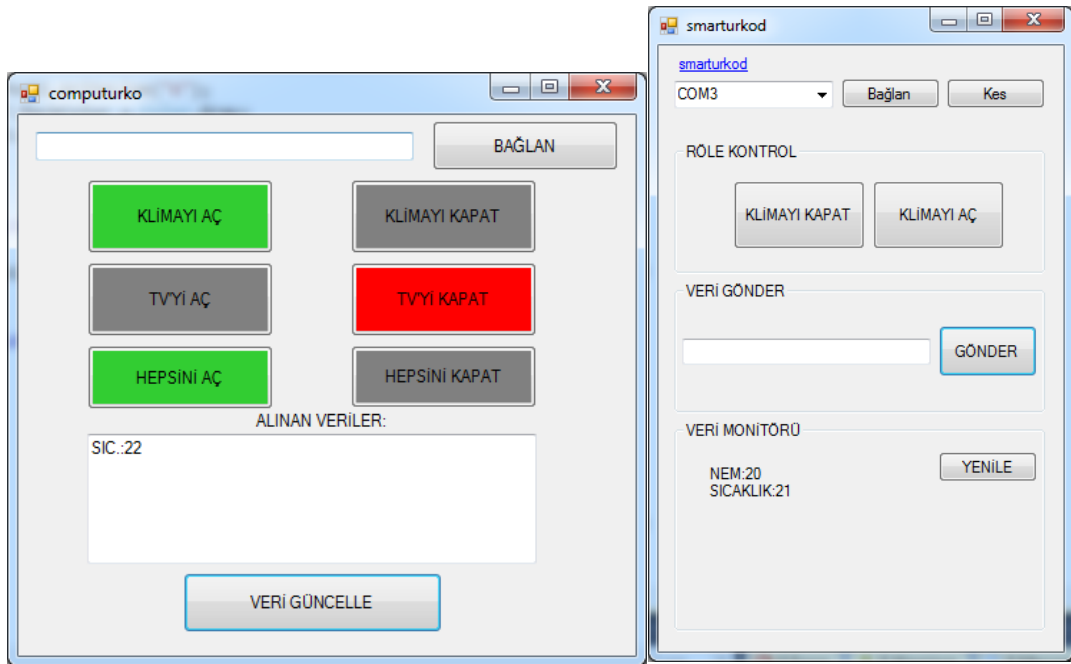


- p) **Projenin çalışma mantığı şu şekildedir:** Android veya bilgisayar üzerinden Bluetooth veya seri iletişimden alınan veri HC05 veya USB kablo ile Arduinoya aktarılmaktadır. Sistemin kodlarındaki işlemler gerçekleştirilmektedir ve uygulama üzerinden gönderilen istek kısa süre içerisinde gerçekleşmektedir. Şekil 12’de temel algoritma gösterilmektedir.



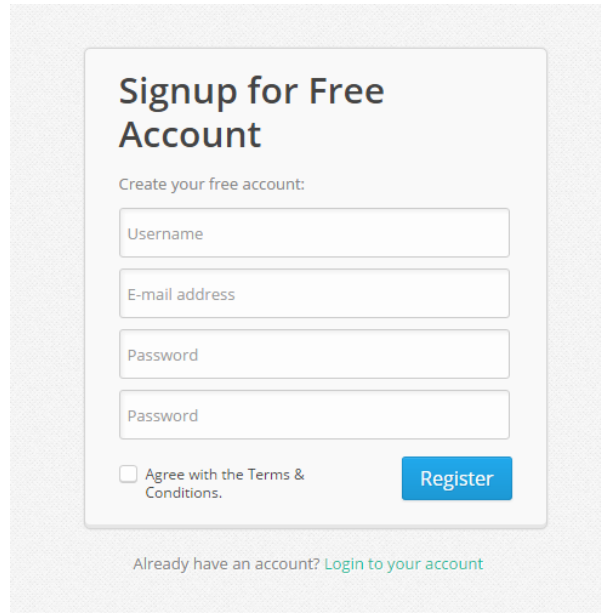
Şekil 12. Çalışma mantığı

- q) **Bilgisayar Uygulaması ve C# Dili kullanarak bilgisayardan sistem kontrolü:** Bilgisayardan kontrol için C# dilinde bir yazılım hazırladım. Visual Studio Geliştirme ortamı bu yazılımı hazırladığım ortamdı. C# dili çok geniş bir dildir. Bu dili kullanarak gömülü sistemlerden, mobil geliştirmeye, oyun geliştirmek vs. mümkündür. Bu projede de **bilgisayar ile sistem USB ile bağlı olmak şartıyla** bilgisayardan kontrol mümkündür. Şekil 13’dekendi oluşturduğum yazılım verilmiştir.

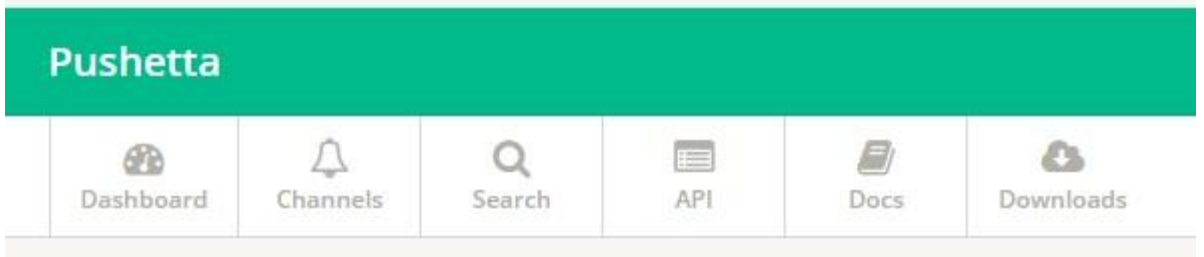


Şekil 13. C# uygulaması

r) **Pushetta bildirim sistemi:** Telefona ve bilgisayara bildirim yollamak için pushetta uygulamasını kullandım. Pushetta ile internet erişimi ve Pushetta uygulamasına sahip her telefonda Arduino'dan bildirim yollayabiliriz. Android telefona uygulamayı yüklemeliyiz, Chrome tarayıcısına da uzantıyı kurmalıyız. Aynı zamanda uygulamaya www.pushetta.com adresinden üye olup kanal açmamız gerekiyor. Siteye girdikten sonra sağ üstte **Signup** butonuna tıklayıp kullanıcı adı, e-posta gibi bilgileri girdikten sonra üyeliği tamamlıyoruz.(Şekil 14.)Üyeliğimiz ile giriş yaptıktan sonra menü olan bir sayfa gelecek.(Şekil 15.)

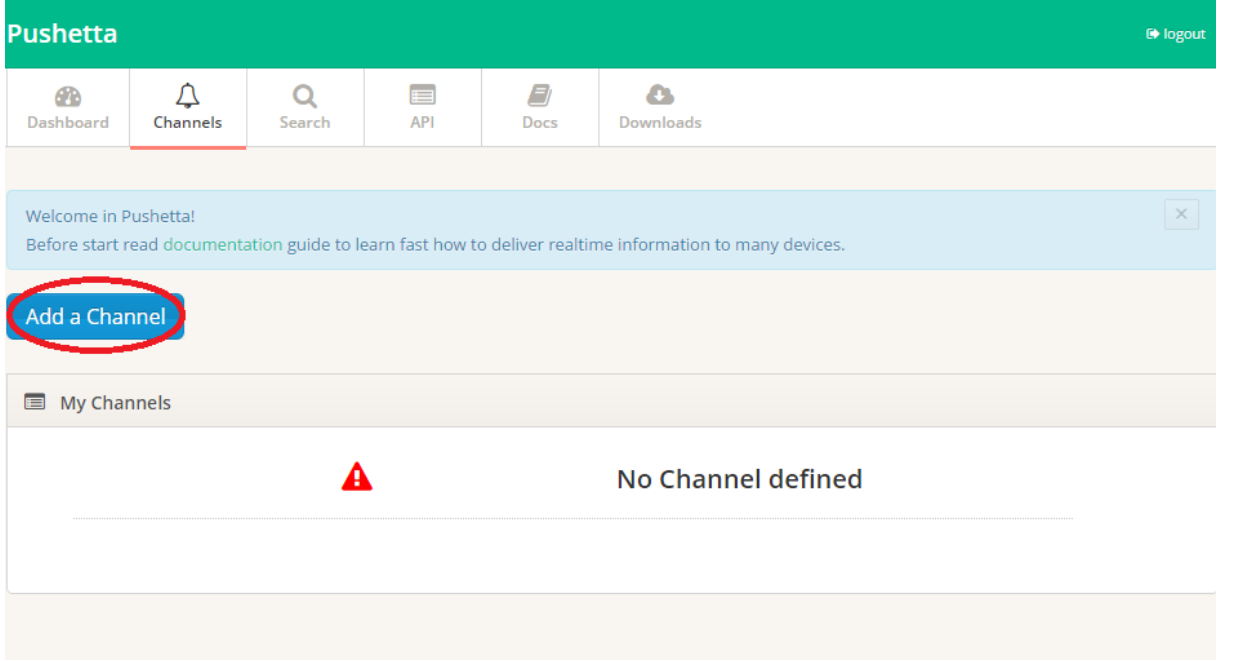
A screenshot of the Pushetta website's registration page. The page has a light gray background. In the center, there is a white box with a gray border. Inside the box, the text "Signup for Free Account" is written in a bold, dark font. Below this, it says "Create your free account:". There are four input fields: "Username", "E-mail address", "Password", and another "Password" field. Below the input fields, there is a checkbox labeled "Agree with the Terms & Conditions." and a blue button labeled "Register". At the bottom of the box, it says "Already have an account? [Login to your account](#)".

Şekil 14. Kayıt olma/giriş yapma ekranı



Şekil 15. Giriş yaptıktan sonra gelen yönetim ekranı

Gelen menüden ‘Channels’ kısmına tıklayarak kanallarımıza erişiyoruz. Hiçbir kanal yoksa Add a Channel butonu ile bir buton ekliyoruz. (Şekil 16.)



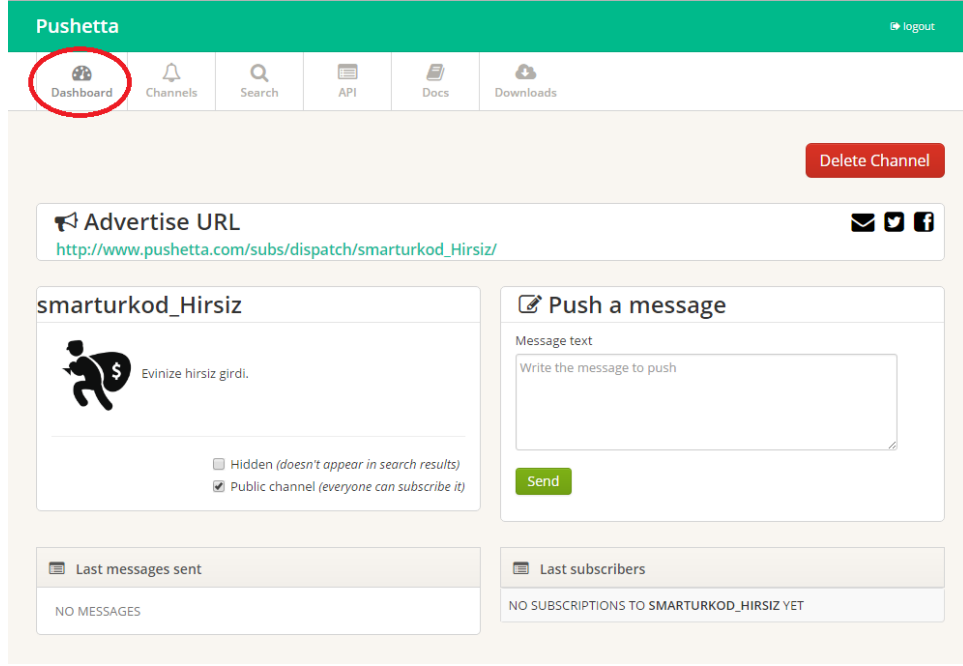
Şekil 16. Kanal eklemek.

Add a Channel butonuna tıkladıktan sonra kanalımızın ayarlarını yapacağımız form geliyor. Formdaki alanları projemize göre ayarladıktan sonra Create butonu ile kaydediyoruz. (Şekil 17.)

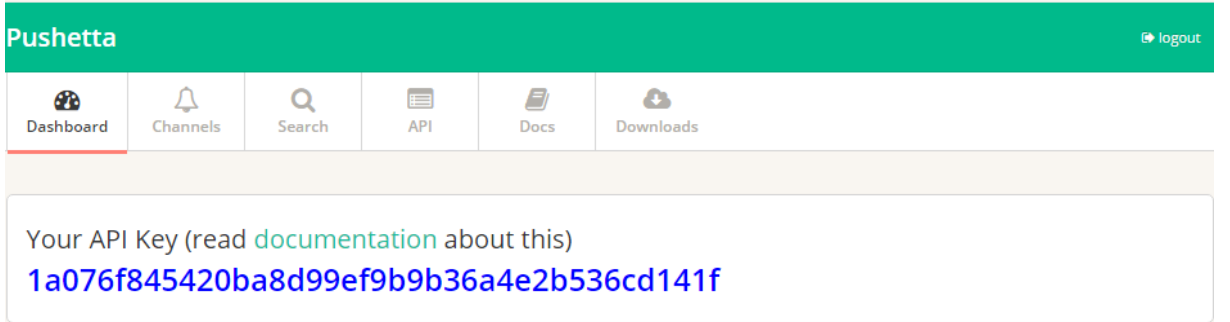
The image shows the 'Create a new Channel' form. It has a title bar with a plus icon and the text 'Create a new Channel'. The form contains several fields: 'Channel Icon' with a placeholder image of a person with a dollar sign and a button 'Select an image for Your channel' (annotated as 'Bildirim ikonu'); 'Channel Name' with the text 'smarturkod - Hırsız' (annotated as 'Bildirim ismi'); 'Description' with the text 'Evinize hırsız girdi. Acil durum.' (annotated as 'Açıklama'); 'Kind' with a dropdown menu set to 'Private' (annotated as 'Gizli veya herkese açık'); and 'Hidden' with a checkbox. At the bottom is a 'Create' button.

Şekil 17. Kanal ayarları

18.) Kanalı ayarladıktan ve oluşturduktan sonra Dashboard sekmesine tıklıyoruz. (Şekil

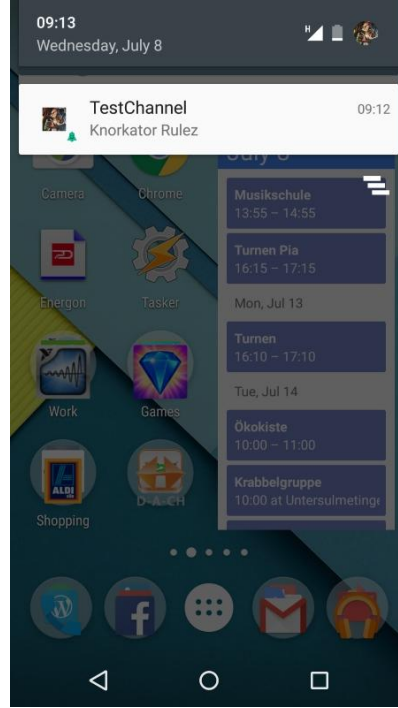


Dashboard sayfasına eriştiğimizde kanalımıza özgü 'Kanal API anahtarı' nı alıyoruz. Bu kodu Arduino koduna eklemeyi unutmamalıyız. (Şekil 19.)

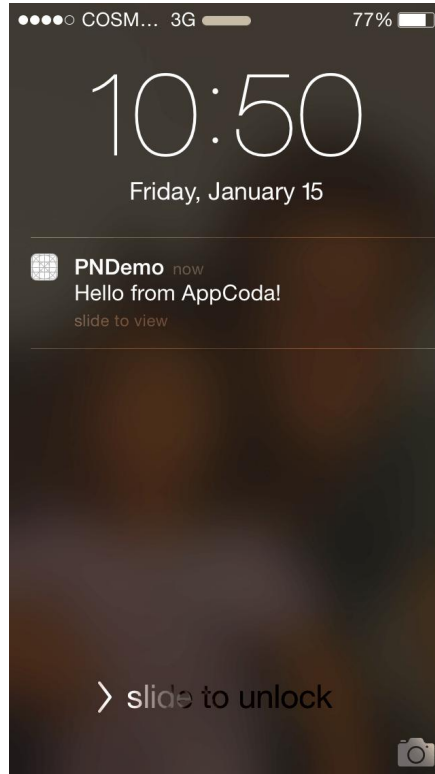


Şekil 19. Kanal API anahtarı.

Şekil 20 ve 21’de mobil cihazlar için gelen bildirim verilmiştir. Pushetta uygulaması iOS ve Android işletim sistemleriyle kullanılabilir.

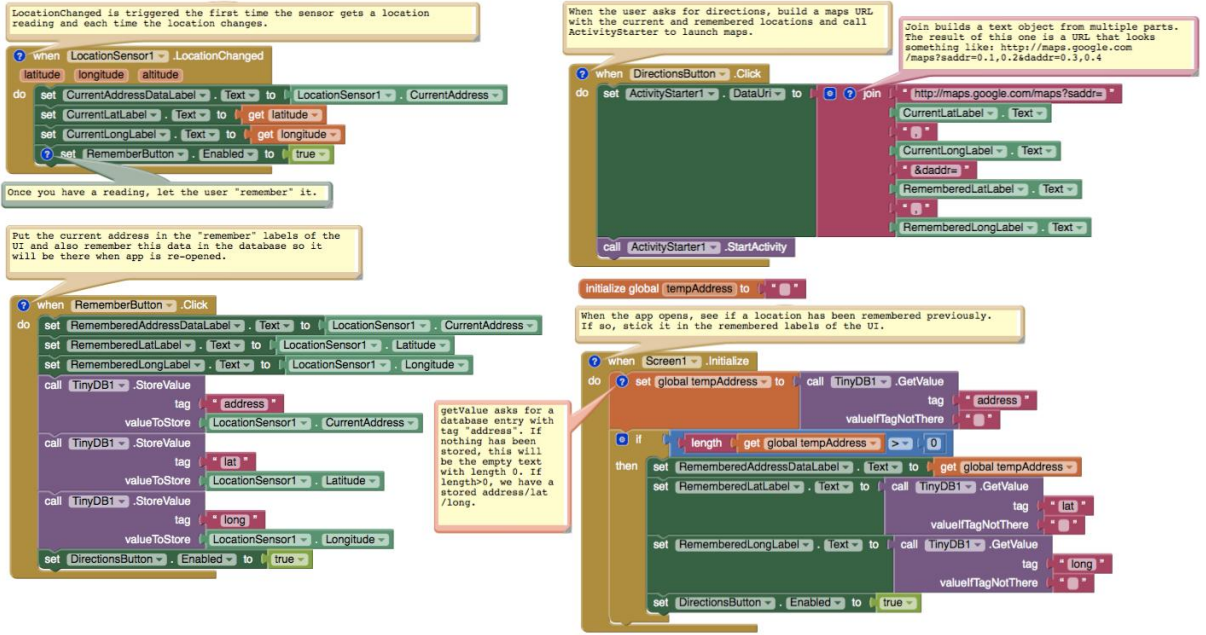


Şekil 20. Android bildirimi.



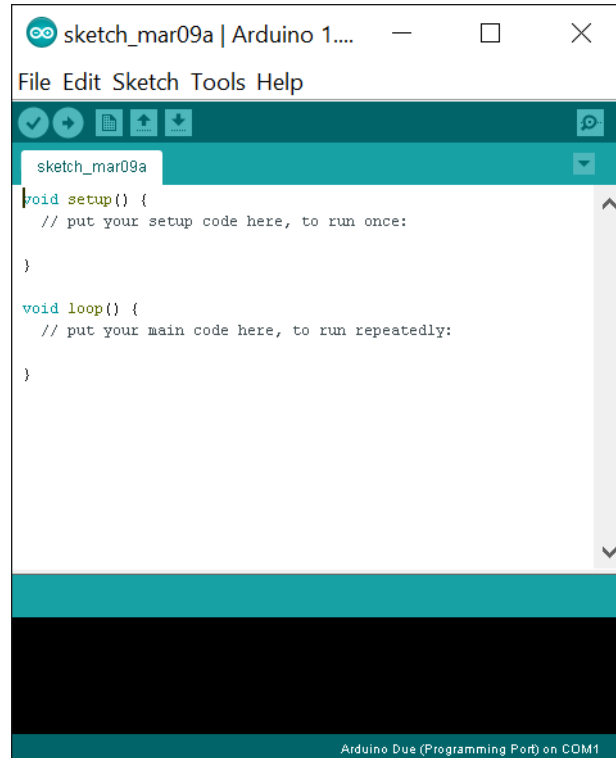
Şekil 21. iOS bildirimi

s) **MIT App Inventor 2:** Sistemin Android uygulaması yapılırken App Inventor 2 kullanılmıştır. Şekil 15’de App Inventor 2 blokları gösterilmiştir.



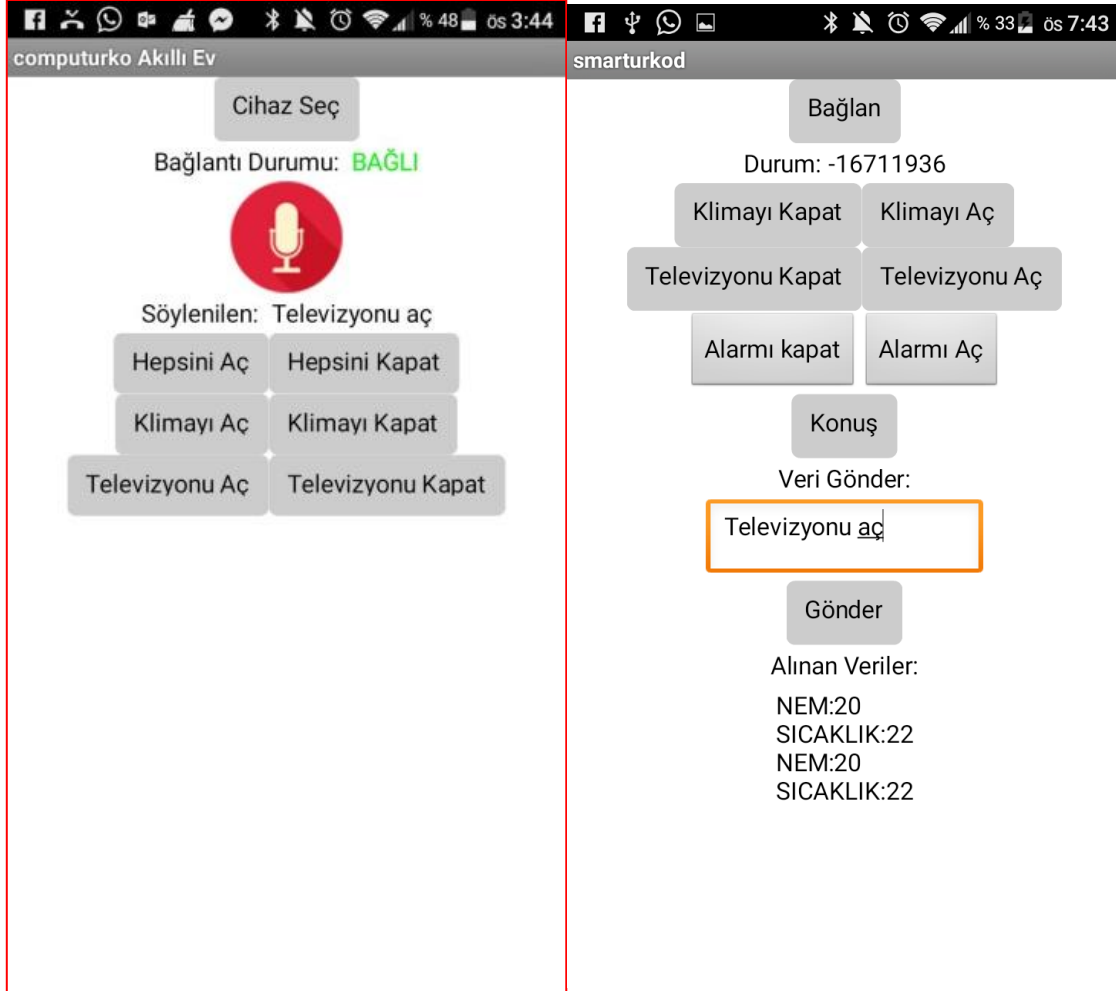
Şekil 15. App Inventor 2 blokları

t) **Arduino Geliştirme Ortamı:** Arduino kodları yazılırken kolaylık sağlaması açısından geliştirme ortamı (IDE) kullanılmaktadır. Şekil 16’de bu ortamdan bir görüntü verilmiştir.



Şekil 16. Arduino IDE

- u) **Android uygulaması ile kontrol mekanizmasının eş zamanlı kullanılması:** Bu projede kontrol noktası olarak C# dili ile üretilen programın yanında AppInventor 2 ile üretilen Android uygulaması da bulunmaktadır. Uygulamanın ekran görüntüsü Şekil 16'de verilmiştir.



Şekil 16. Ekran görüntüsü

v) Proje Kodları:

Arduino Tabanlı Kodlar:

```
#include <IRLib.h>
IRsend My_Sender;
#include <GSM.h>
#define PINNUMBER ""//simde pin varsa buraya yazın
GSM gsmAccess;
GSM_SMS sms;
int ses_sensoru = 24;
int alkis = 0;
int role = 18;
int ldr_pin = A4; //ldr'nin pinini tanımladık
int deger = 0; //ldr değerini tanımladık
long algilama_araligi_baslangic = 0;
long algilama_araligi = 0;
boolean isik_durumu = false;
char senderNumber[20];
#define ag_ismi "SSID"
#define ag_sifresi "password"
#define SERVER "api.pushetta.com"
String API("2828e318e8357eb7285ff2962642b132de28b1b");
String CHANNEL("SMARTURKOD");
#include <SoftwareSerial.h>
SoftwareSerial espSerial(14,15);
#include <LiquidCrystal.h>//LCD kütüphanesi
#include <SoftwareSerial.h>//Seri iletişim kütüphanesi
#include<dht11.h>
#define DHT11PIN 9 //DHT11 pini
SoftwareSerial kontroll1(0,1);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //LCD pinleri
dht11 DHT11; //DHT11 tanımlama
int role1=7;
int role2=8;
int zil=10;
int alarm
int degiskenDeger = "999999999999";
int sensorHirsiz=analogRead(A5);
int sensorYangin = analogRead(A3);
int sensorSu =analogRead(A2);
int veri;
void setup()
{
  pinMode(ses_sensoru, INPUT);
  pinMode(role, OUTPUT)
  pinMode(role1, OUTPUT);
  pinMode(role2, OUTPUT);
  digitalWrite(role1, HIGH);
  digitalWrite(role2, HIGH);
  Serial.begin(9600);
  kontroll1.begin(9600);
  lcd.begin(16,2);
  lcd.setCursor(3,0);
  lcd.print("SMARTURKOD");
  delay(3000);
  lcd.clear();
  espSerial.begin(115200);
  espSerial.println("AT");
  delay(5000);
  if(espSerial.find("OK"))
```



```

{
delay(2000);
String
baglantiKomutu=String("AT+CWJAP=\"") + ag_ismi+"\", \"\"+ag_sifresi+"\"";
espSerial.println(baglantiKomutu);
delay(5000);
}
sms1.begin(9600);
while (!sms1) {
;
}
boolean notConnected = true;
while (notConnected) {
if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
notConnected = false;
} else {
sms1.println("baglanamadi");
delay(100);
}
}
sms.print("AT+CMGD=1,4");//tüm smsleri sil
delay(1000);
sms1.println("GSM baglandi");
}
void seriYaz()
{
Serial.write("NEM:");
Serial.print(DHT11.humidity);
Serial.write("\n");
Serial.write("SICAKLIK:");
Serial.print(DHT11.temperature);
Serial.write("\n");
delay(100);
}
void veriPushetta(String CHANNEL,String text)
{
espSerial.println(String("AT+CIPSTART=\"TCP\", \"\" + SERVER + "\",80");
//server'a bağlanma
delay(3000);
String bildirim=("POST /api/pushes/");
bildirim+= (CHANNEL);
bildirim+= ("/ HTTP/1.1");
bildirim+= ("\r\n");
bildirim+= ("Host: ");
bildirim+= (SERVER);
bildirim+= ("\r\n");
bildirim+= ("Authorization: Token ");
bildirim+= (API);
bildirim+= ("\r\n");
bildirim+= ("Content-Type: application/json");
bildirim+= ("\r\n");
bildirim+= ("Content-Length: ");
bildirim+= (text.length() + 46);
bildirim+= ("\r\n\r\n");
bildirim+= ("{" + "\"body\" : \"";
bildirim+= (text);
bildirim+= ("\", \"message_type\" : \"text/plain\" }");
bildirim+= ("\r\n\r\n");
espSerial.print("AT+CIPSEND=");
delay(100);
espSerial.println(bildirim.length());
}

```

```

// Sorun oluşmadıysa veriyi gönder.
if(espSerial.find(">")){
    espSerial.println(bildirim);
    espSerial.println("AT+CIPCLOSE=0");
    delay(100);
}
else
{
    espSerial.println("AT+CIPCLOSE=0");
}
}

void smsYangin()
{
    String txtMsg = "";
    txtMsg = "YANGIN VAR!";
    Serial.println("Gonderiliyor");
    Serial.println(txtMsg);
    sms.beginSMS("05321234567");//sms atılacak telefon numarası
    sms.print(txtMsg);
    sms.endSMS();
    Serial.println("\nTAMAMLANDI!\n");
    delay(100);
}

void smsHirsiz() {
    String txtMsg = "";
    txtMsg = "HIRSIZ VAR!";
    Serial.println("Gonderiliyor");
    Serial.println(txtMsg);
    sms.beginSMS("05321234567");//sms atılacak telefon numarası
    sms.print(txtMsg);
    sms.endSMS();
    Serial.println("\nTAMAMLANDI!\n");
    delay(100);
}

void smsSu() {
    String txtMsg = "";
    txtMsg = "SU BASTI!";
    Serial.println("Gonderiliyor");
    Serial.println(txtMsg);
    sms.beginSMS("05321234567");//sms atılacak telefon numarası
    sms.print(txtMsg);
    sms.endSMS();
    Serial.println("\nTAMAMLANDI!\n");
    delay(100);
}

void loop()
{
    seriYaz();
    int chk= DHT11.read(DHT11PIN);
    lcd.setCursor(0,0);
    lcd.print("NEM :");
    lcd.print(DHT11.humidity);
    lcd.setCursor(0,1);
    lcd.print("SIC.:");
    lcd.print(DHT11.temperature);
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("NEM/SIC:");lcd.print(DHT11.humidity/DHT11.temperature);
    delay(2000);
}

```

```

lcd.clear();
if(sensorHirsiz>degiskenDeger)
{
digitalWrite(10,HIGH);
delay(500);
digitalWrite(10,LOW);
delay(500);
sms.flush();
smsHirsiz();
veriPushetta(CHANNEL,"HIRSIZ VAR!");
}
if(sensorSu<=degiskenDeger)
{
digitalWrite(10,HIGH);
delay(500);
digitalWrite(10,LOW);
delay(500);
sms.flush();
smsSu();
veriPushetta(CHANNEL,"SU BASTI!");
}
else
{
digitalWrite(10,LOW);
}
if(sensorYangin>degiskenDeger)
{
digitalWrite(10, HIGH);
delay(500);
digitalWrite(10, LOW);
delay(500);
sms.flush();
smsYangin();
veriPushetta(CHANNEL,"YANGIN VAR!");
}
if(Serial.available())
{
int pos = 0;
pos = Serial.read();
if(pos>0)
{
analogWrite(6, pos);
}
}
veri = Serial.read();
if(veri=='1')
{
digitalWrite(role1, 0);
}
if(veri=='2')
{
digitalWrite(role1, 1);
}
if(veri=='3')
{
digitalWrite(role2, 0);
}
if(veri=='4')
{
digitalWrite(role2, 1);
}
if(veri=="Klimayı aç")

```

```

{
digitalWrite(role1, 0);
}
if(veri=="Klimayı kapat")
{
digitalWrite(role1, 1);
}
if(veri=="Televizyonu aç")
{
digitalWrite(role2, 0);
}
if(veri=="Televizyonu kapat")
{
digitalWrite(role2, 1);
}
if(veri=="kanal1")
{
My_Sender.send(SONY,0xa8bca, 20);
}
}
int sensor_durumu = digitalRead(ses_sensoru);

if (sensor_durumu == 0)
{
if (alkis == 0)
{
algilama_araligi_baslangic = algilama_araligi = millis();
alkis++;
}
else if (alkis > 0 && millis()-algilama_araligi >= 50)
{
algilama_araligi = millis();
alkis++;
}
}

if (millis()-algilama_araligi_baslangic >= 400)
{
if (alkis == 2)
{
if (!isik_durumu)
{
isik_durumu = true;
digitalWrite(role, HIGH);
}
else if (isik_durumu)
{
isik_durumu = false;
digitalWrite(role, LOW);
}
}
alkis = 0;
}
if (deger > 0 && deger <= 100)
{
digitalWrite(ledler[0], HIGH); //eğer 0 ile 100 arasındaysan ledi yak
}
}
}

```

App Inventor 2 Blokları:

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL: ai2.appinventor.mit.edu/?locale=en#6695682629697536. The interface is divided into several panels:

- Palette:** Contains categories like User Interface, Layout, Media, Drawing and Animation, and Sensors. The User Interface section is expanded, showing components like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, TextBox, TimePicker, and WebView.
- Viewer:** Shows a preview of the app on a mobile device. The app's UI includes a status bar at the top with the time 9:48. The main screen has a title "smarturkod" and a series of buttons and labels: "Bağlan", "Durum:", "Klimayı Kapat", "Klimayı Aç", "Televizyonu Kapat", "Televizyonu Aç", "Alarmı Kapat", "Alarmı Aç", "Konuş", "Veri Gönder:", a text input field, "Gönder", and "Alınan Veriler:". Below the preview, there are "Non-visible components" including BluetoothClient1, SpeechRecognizer1, and Clock1.
- Components:** A list of components used in the app, including Screen1, ListPicker1, HorizontalArrangeme, Label1, Label2, HorizontalArrangeme, Button1, Button2, HorizontalArrangeme, Button3, Button4, HorizontalArrangeme, Button8, Button7, HorizontalArrangeme, Button6, Label3, TextBox1, and Button5.
- Properties:** A panel on the right showing the properties of the selected component (Screen1). Properties include AboutScreen, AlignHorizontal (Center: 3), AlignVertical (Top: 1), AppName (smarturkod), BackgroundColor (White), BackgroundImage (None...), CloseScreenAnimation (Default), Icon (None...), OpenScreenAnimation (Default), ScreenOrientation (Unspecified), Scrollable, and ShowStatusBar (checked).

The bottom of the interface shows a Windows taskbar with various application icons and a system clock indicating 19:31 on 15.01.2017.

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
set Label2 . Text to "Bağlı"
set Label2 . Text to

when Button8 .Click
do call BluetoothClient1 .SendText
text " 5 "

when Button1 .Click
do call BluetoothClient1 .SendText
text " 2 "

when Button7 .Click
do call BluetoothClient1 .SendText
text " 6 "

when Button6 .Click
do call SpeechRecognizer1 .GetText

when Button3 .Click
do call BluetoothClient1 .SendText
text " 4 "

when Button2 .Click
do call BluetoothClient1 .SendText
text " 1 "

when Button4 .Click
do call BluetoothClient1 .SendText
text " 3 "

when Button5 .Click
do call BluetoothClient1 .SendText
text TextBox1 . Text

when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then if call BluetoothClient1 .BytesAvailableToReceive
then set Label5 . Text to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

when SpeechRecognizer1 .AfterGettingText
result
do call BluetoothClient1 .SendText
text SpeechRecognizer1 . Result

C# Kodları:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.IO.Ports;

namespace SMARTURKOD
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            for (int i = 0; i <
System.IO.Ports.SerialPort.GetPortNames().Length; i++)
            {
                comboBox1.Items.Add(System.IO.Ports.SerialPort.GetPortNames()[i]);
            }
        }
        private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
        {
            if (serialPort1.IsOpen) serialPort1.Close();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            timer1.Start();
            try
            {
                serialPort1.PortName = comboBox1.Text;
                if (!serialPort1.IsOpen)
                    serialPort1.Open();
                MessageBox.Show("Bağlanılan port:
"+serialPort1.PortName);
            }
            catch { }
        }
    }
}
```

```

    }
    catch
    {
        MessageBox.Show("Zaten Seri Porta Bağlı:
"+serialPort1.PortName);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        if (serialPort1.IsOpen)
            serialPort1.Close();
        MessageBox.Show("Seri Port Kapatıldı");
    }
    catch
    {
        MessageBox.Show("Seri Port Kapalı");
    }
}

private void button3_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        serialPort1.WriteLine("2");
    }
    else
    {
        MessageBox.Show("Seri Porta Bağlan");
    }
}

private void button5_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        serialPort1.Write("1");
    }
    else
    {
        MessageBox.Show("Seri Porta Bağlan");
    }
}

private void button4_Click(object sender, EventArgs e)
{

```



```

        if (serialPort1.IsOpen)
        {
            serialPort1.WriteLine(textBox1.Text);
            textBox1.Text = "";
        }
        else
        {
            MessageBox.Show("Seri Porta Bağlan");
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {

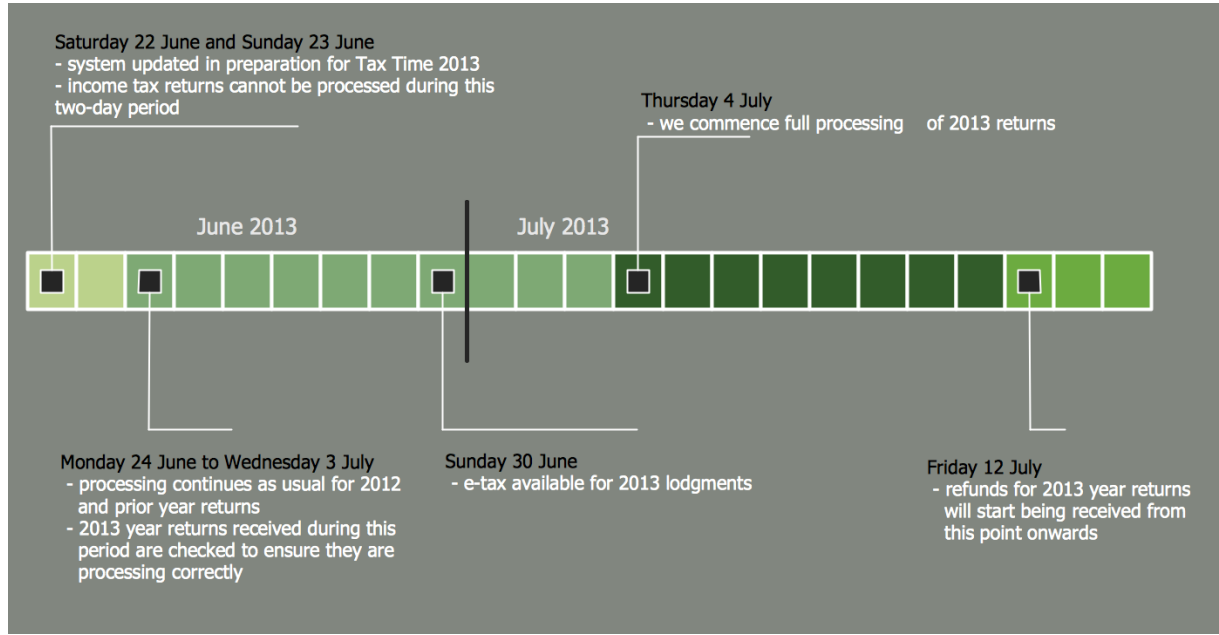
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        try
        {
            string sonuc =
serialPort1.ReadExisting();//Serial.print(sicaklik); ile gelen
sıcaklık değerini alıyoruz.
            label1.Text = sonuc + ""; //Labele yazdırıyoruz.
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

}
}

```

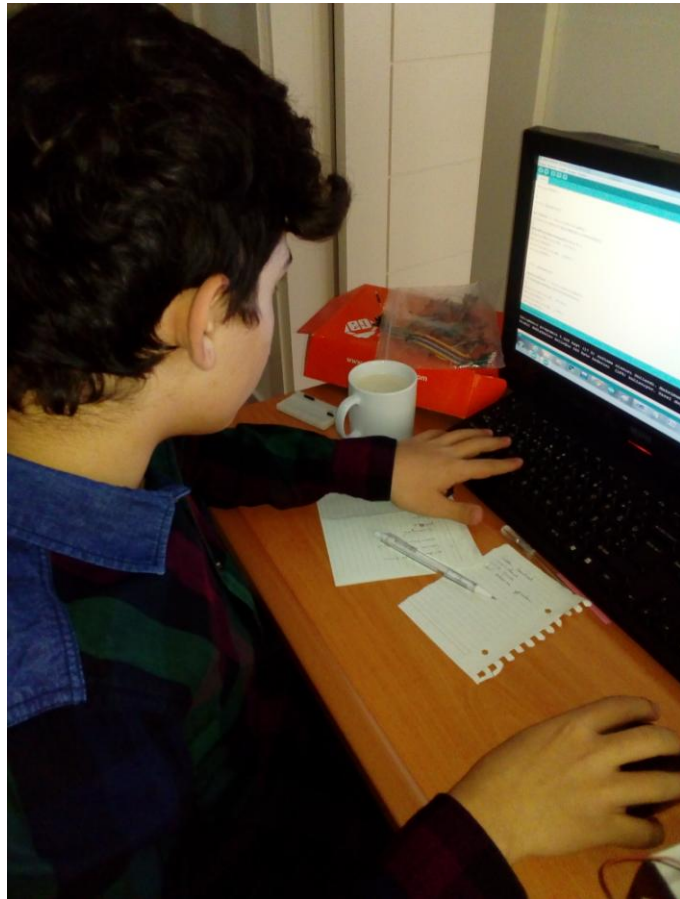
4) Proje ile ilgili Görseller ve Projenin Yapım Aşamaları

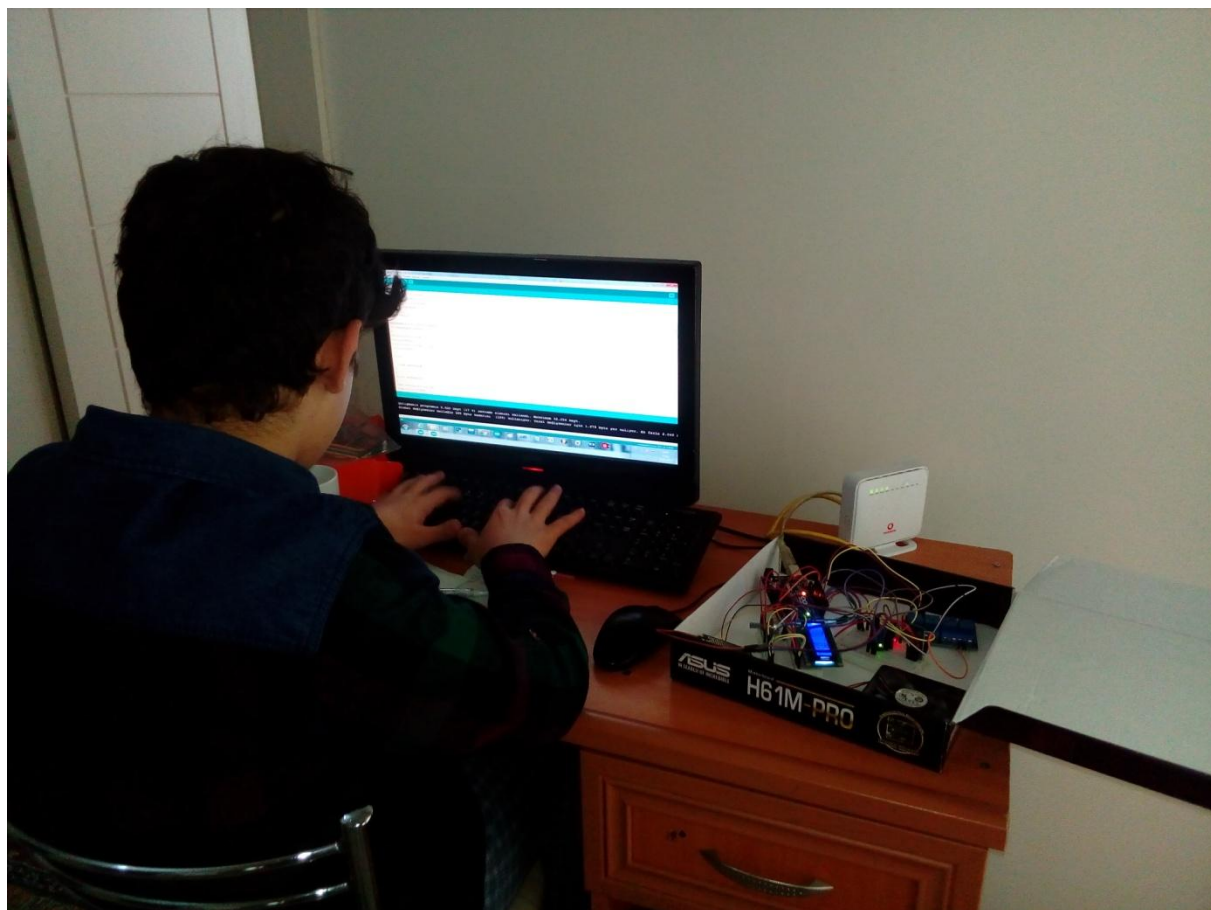


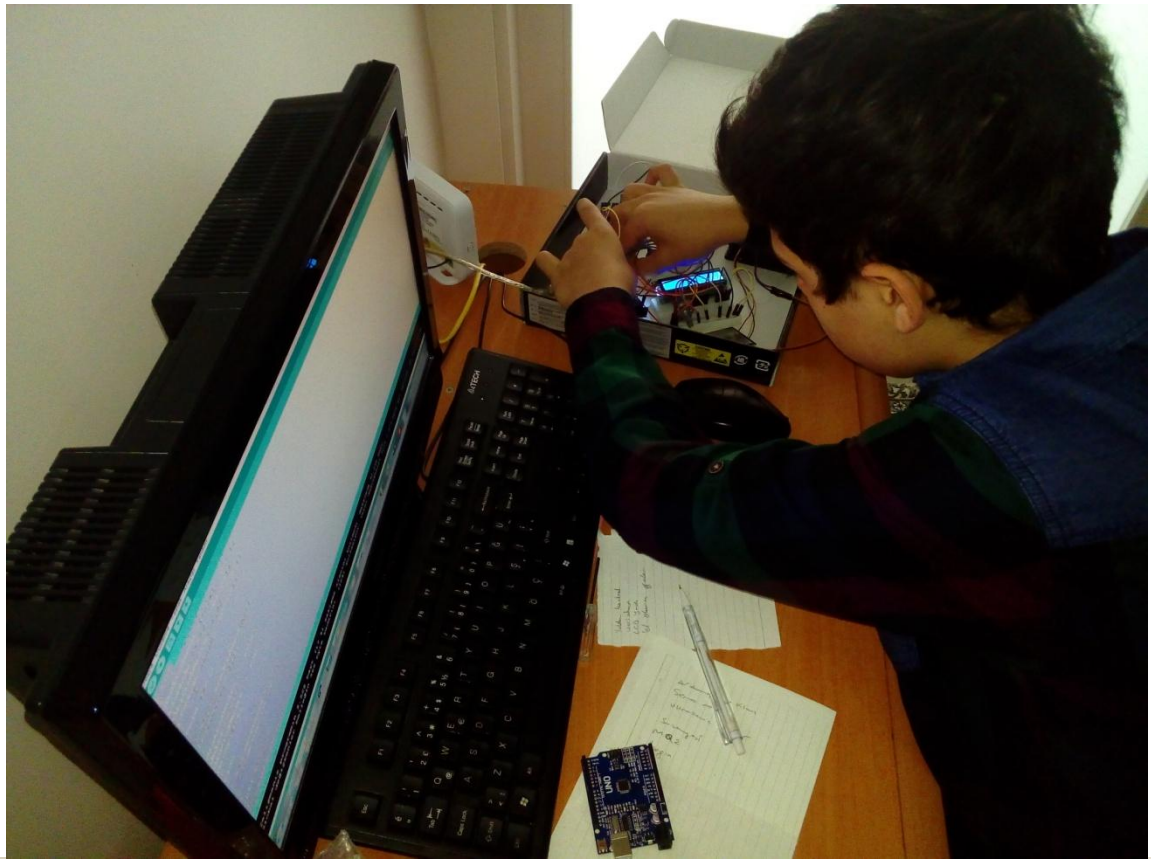
EVİNİZİ KONTROL EDİN

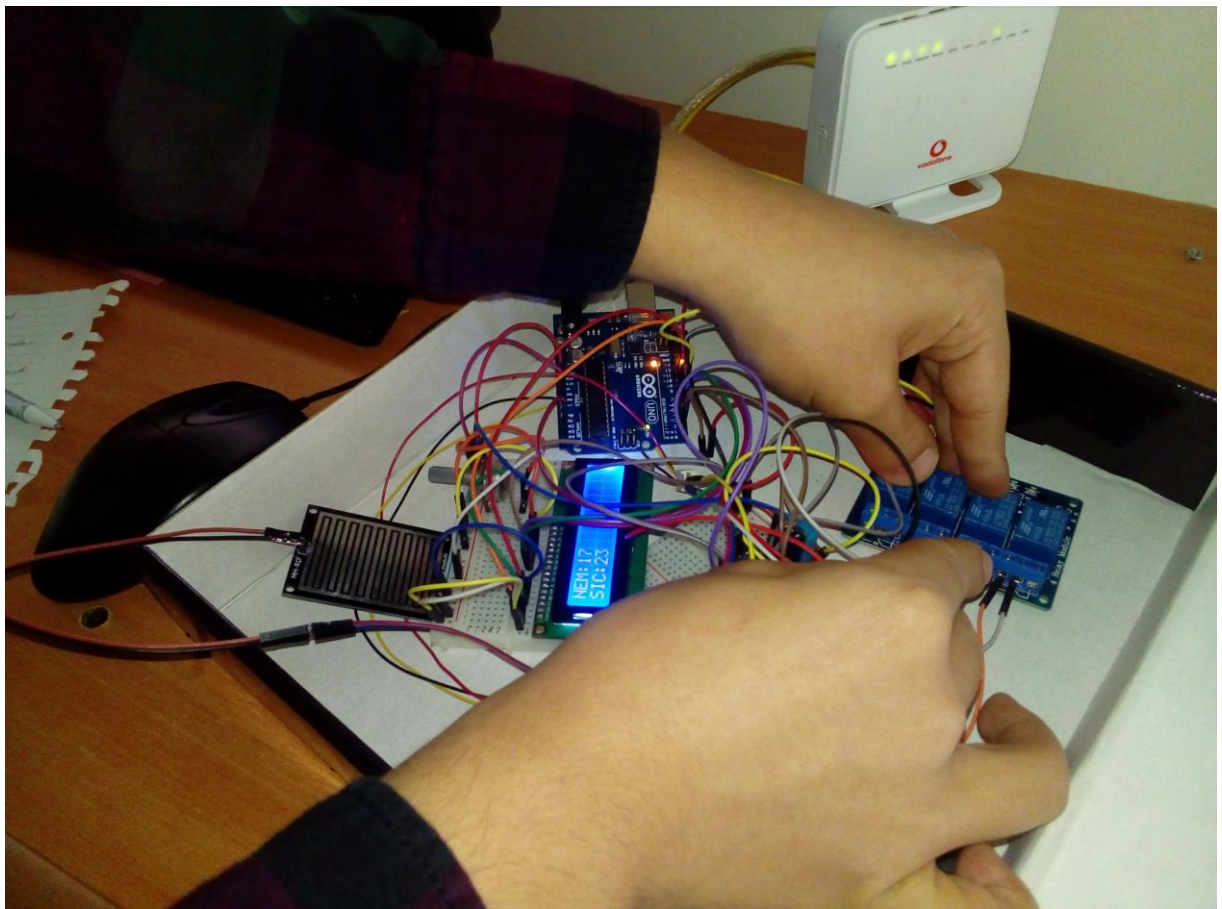
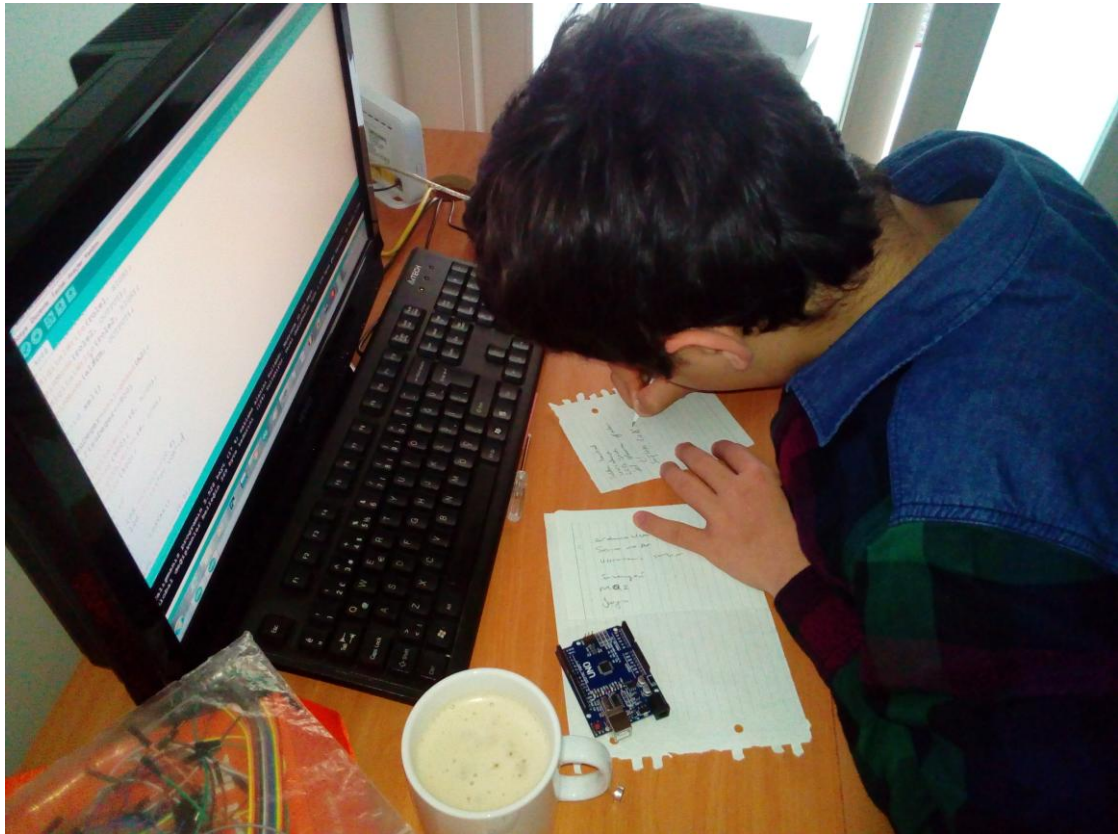


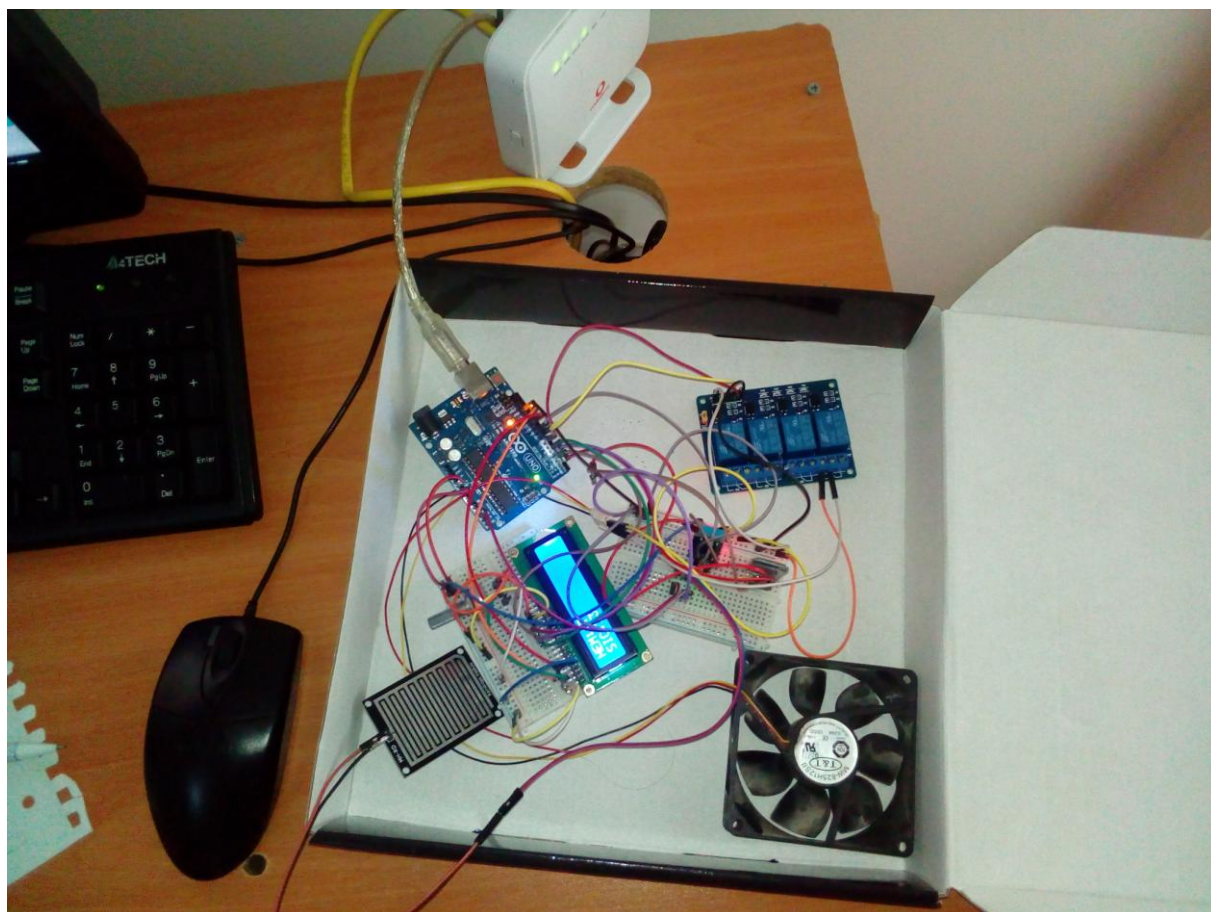












5) SONUÇ VE ÖNERİLER

Kullanıcının kolay bir şekilde sisteme erişebilmesi için mobil cihaz, C# destekli uygulama, Android uygulaması ve kontrol menüsü tasarlanmış ve bu arayüz ile kullanıcı ve SMARTURKOD sisteminin aktif etkileşimi sağlanmıştır.

Projemiz kapsamında yapmış olduğumuz akıllı ev sistemindeki algılayıcılar ile ev hakkında gerçek zamanlı olarak bilgi alma, sistemi kontrol etme ve kullanıcıya bu bilgileri gönderme gerçekleştirilmiştir. Bu kapsamda Bluetooth, Wi-Fi ve GSM tabanlı akıllı ev uygulamasının işlevleri şu şekilde sıralanabilir; kullanıcıdan veri alınır, veri gerekli algoritmalar ile kullanılabilir hale gelir ve sistemdeki uygulaması yapılır. Günümüzde akıllı ev sistemlerinin kurulum maliyetleri yüksektir. Gerçekleştirilen bu uygulama ile daha düşük maliyete etkin bir akıllı ev uygulaması gerçekleştirilmiştir.

Günümüzde konfor ve güvenlik ev tasarımında en önemli faktörlerin başındadır. Teknolojinin ve dolayısı ile otomasyon sistemlerinin her geçen gün gelişmesi bizleri hayal gücümüzün sınırlarını zorlamamıza sağlamaktadır.

Ev yaşantımızda kullanmakta olduğumuz birçok elektrikli alet ve eşyanın, farklı kontrol mekanizmalarına sahip olması ve programlanabilme özellikleri sayesinde geliştirilen teknolojik ev cihazları ile ciddi anlamda zamansal ve maddi anlamda tasarruf yapılmasını sağlamaktadır. Dolayısıyla teknoloji ve otomasyon evlerimize girmektedir. Ev otomasyonu ilk bakışta karmaşık gelmesine karşın bilgisayarın günümüzde hayatımızın her yerinde görülmekte adeta parçası olmaktadır. Teknolojinin ev otomasyonu ile ilgili kullanımlarının da sadeleştirilerek pratik ve kullanıcı kolaylığı sağlamasına hedeflenmektedir. Bu amaçla Arduino kullanılarak ‘SMARTURKOD’ sistemi oluşturulmuştur.

Uzaktan kontrollü akıllı ev sistemi yönetim ekranı ile üç farklı mekanizma ile kontrol edilmektedir. Android uygulaması, C# uygulaması, Internet kontrol sistemi ve GSM ile istenilen işlem kablosuz veya kablolu bir şekilde sisteme iletilmekte, gerekli işlem yapılmaktadır.

Günümüzde her evde bilgisayar olduğunu düşündüğümüzde insanların hayatlarını kolaylaştıracak bir proje olduğuna inanıyoruz.

Sonuç olarak, bu proje hem maliyeti az, hem de geliştirmeye ve yeniliklere açık bir akıllı ev sistemi tasarlanmıştır. Proje, gerek engelli insanlar için gerekse akıllı sistemlerinin kullanıldığı sahalarda tercih sebebi olabilmektedir. Teknoloji çapı dediğimiz bu çağda evlerimize kadar giren teknoloji, gelecekte ne getirecek bilinmez.

6) KAYNAKÇA:

- <http://maker.robotistan.com/arduino-dersleri-17-hc-05-bluetooth-modulu-kullanimi/>
- <http://arduinoturkiye.com/dht11-sicaklik-ve-nem-sensorunun-arduino-ile-kullanimi/>
- <https://create.arduino.cc/projecthub/nolan-mathews/connect-to-blynk-using-esp8266-as-arduino-uno-wifi-shield-m1-46a453>
- <http://playground.arduino.cc/Main/MQGasSensors>
- [https://msdn.microsoft.com/tr-tr/library/system.io.ports.serialport\(v=vs.110\).aspx](https://msdn.microsoft.com/tr-tr/library/system.io.ports.serialport(v=vs.110).aspx)
- <http://forum.mekatronikmuhendisligi.com/Konu-c-seri-port-kullanimi.html>
- <http://www.instructables.com/id/Connect-Arduino-Uno-to-Android-via-Bluetooth/>
- <http://www.instructables.com/id/Course-on-MIT-App-Inventor-and-Arduino/>
- <http://www.williamwhitepapers.com/pr/Chronology%20of%20SMART%20Recovery.pdf>
- <http://www.robotiksystem.com/arduino-mega2560-schematic.pdf>
- Douligeris, C., “Intelligent homesystems” , *IEEE Communications Magazine*, 52-61 (1993).
 - Koyuncu, B., “PC remote control of appliances by using telephoning lines” , *IEEE Transactions on Consumer Electronics*, 41 (1): 201-209 (1995).
 - Coskun, __, Ardam, H., “A remote controller for home and office appliances by telephone” , *IEEE Transactions on Consumer Electronics*, 44 (4): 1291-1297 (1998).
 - Park, M. J., Lee, S. J., Yoon, D. H. “Signal detection and analysis of DTMF receiver with quick fourier transform” , *The 30th Annual Conference of the IEEE Industrial Electronics Society*, Korea, 2058-2064 (2004).
 - Jiang, L., Liu, D. Y., Yang, B., “Smart home research” , *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai , 659-663 (2004).
 - Stefanov, D. H., Bien, Z., Chul Bang, W., “The smart house for older persons and persons with physical disabilities” , *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12 (2): 228-250 (2004)
 - Bindevre Ltd.Şti. Tasarım, proje ve teknik uygulama notları
 - Çetin, K. and Şahin, Ö., “Voice Integrated Home Automation and Security System through Telephone Line”, 5th International Conference on Electrical and Electronics Eng.(ELECO 2007), 5-9 Aralık 2007, Bursa, s.364-367.
 - 3.Görel, G. ve Küçüktepe, A., “Yeni Nesil Akıllı Ev Otomasyonu Uygulaması”, MKT2012, Proje Tabanlı Mekatronik Eğitim Çalıştayı, 25-27 Mayıs 2012, Çankırı.
 - 4.Bayer; S.E. ve Ertunç, H.M. “Akıllı Ev Otomasyon Uygulamaları”, III.Otomasyon Sempozyumu ve Sergisi, Pamukkale Üniversitesi, Denizli, s.6-10, .2005.
 - 5.Binaların Yangından Korunması Hakkında Yönetmelik