

IOT PROJE
AKILLI EV OTOMASYONU

HAZIRLAYANLAR:

AD VE SOYAD: AYŞEGÜL ÖZSOY, ZEHRA BEGÜM AKTOLGA

NUMARA: B191210098, B191210062

ŞUBE: 1. ÖĞRETİM B GRUBU

İÇİNDEKİLER:

- İŞ MODELİ
- PROJE AÇIKLAMASI
- DEVRE KURULUMU
- KOD

Temel Özellikler	Temel Faaliyetler	Değer Önerisi	Müşteri İlişkileri	Müşteri Segmenti
Özellikler Yatırımcılar, Ev sahibi olan insanlar	İşik Süddet Ayarı Gün ışığına göre ışık şiddetini ayarlama	Müşteriler için Ev içinde ki yaşam kalitesini artırmak	Ev Evde yaşayan herkes	Ev Evde yaşayan herkes
	Temel Kaynaklar Kaynaklar Teknolojik altyapı, Ev sahibi insanlar		Kanallar Kanal Evdere uzaktan kontrol için akıllı telefonlar	
Maliyet Yapısı Maliyet Teknolojik altyapı			Gelir Akışı Gelir Evde kullanılan her lamba için belirli ücret alınabilir.	

İŞ MODELİ

AKILLI EV OTOMASYONU

Bu projedeki amacımız gnlk yařamı biraz daha kolaylařtırmak ve konforlu hale getirebilmek.

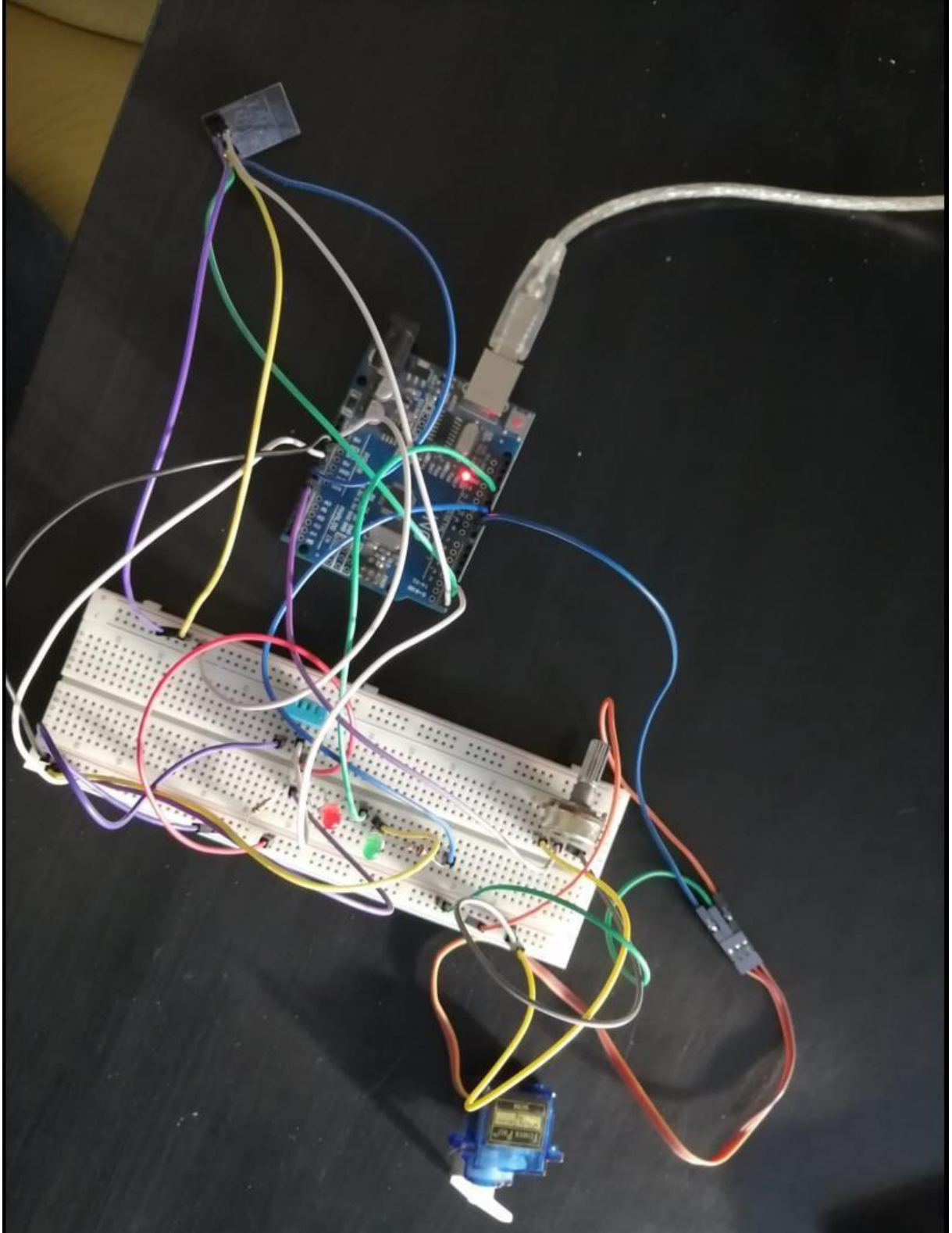
Bunu saęlamak iin projemizde evde odaların ierisindeki ıřık miktarına gre ledlerin aılıp kapatılmaya karar verilmesi veya ıřık durumuna gre evde yapılabilcek etkinliklerin seimine karar verilmesi ayrıca odanın sıcaklık ve nem deęeri llp telefon uygulaması aracılıęıyla gsterilmesi saęlanır. Mesela gn ıřıęı fazlaysa ledin snk olması ve evde kitap okumanın daha rahat olduęuna karar verilmesi. Ya da ıřık miktarı yetersizse ledin parlaklıęını arttırıp kitap okunması gibi.

Projeyi gerekleřtirebilmek iin ıřık řiddetini arttırıp azaltabilmek amacıyla potansiyometre ve servo motor kullandık. Telefonda bulunan uygulama sayesinde ledi aıp kapatma kontrol ve ıřık řiddetinin ayarlanması gibi zellikler telefon zerinden saęlanır.

DEVRE KURULUMU

- Arduino UNO
- ESP8266 (ESP-01)
- Potansiyometre
- Direnç
- Jumper Kablolar
- Breadboard
- Ledler
- Servo Motor
- DHT11(Sıcaklık ve nem sensörü)

DEVRE FOTOĞRAFI



KULLANILAN KOD

```
#include "DHT.h"
```

```
#define DHTPIN 2
```

```
#include <Servo.h>
```

```
DHT dht(DHTPIN,DHT11);
```

```
Servo otomasyon;
```

```
/*
```

```
-- New project --
```

This source code of graphical user interface

has been generated automatically by RemoteXY editor.

To compile this code using RemoteXY library 3.1.6 or later version

download by link <http://remotexy.com/en/library/>

To connect using RemoteXY mobile app by link <http://remotexy.com/en/download/>

- for ANDROID 4.8.01 or later version;

- for iOS 1.5.1 or later version;

This source code is free software; you can redistribute it and/or

modify it under the terms of the GNU Lesser General Public

License as published by the Free Software Foundation; either

version 2.1 of the License, or (at your option) any later version.

```
*/
```

```
////////////////////////////////////
```

```
// RemoteXY include library //
```

```
////////////////////////////////////
```

```
// RemoteXY select connection mode and include library
```

```
#define REMOTEXY_MODE_ESP8266_HARDSERIAL_POINT
```

```
#include <RemoteXY.h>
```

```
// RemoteXY connection settings
```

```
#define REMOTEXY_SERIAL Serial
```

```
#define REMOTEXY_SERIAL_SPEED 115200
```

```
#define REMOTEXY_WIFI_SSID "aysebegm"
```

```
#define REMOTEXY_WIFI_PASSWORD
```

```
"123456789"
```

```
#define REMOTEXY_SERVER_PORT 6377
```

```
// RemoteXY configurate
```

```
#pragma pack(push, 1)
```

```
uint8_t RemoteXY_CONF[] =
```

```
{ 255,4,0,33,0,118,0,13,13,1,
```

```
1,0,8,6,12,12,37,31,88,0,
```

```
2,0,30,5,22,11,106,26,31,31,
```

```
79,78,0,79,70,70,0,67,4,43,
```

```
37,20,5,64,26,11,129,0,5,37,
```

```
35,5,94,112,111,116,97,115,121,111,
```

```
32,100,101,196,159,101,114,105,0,5,
```

```
0,10,44,30,30,2,26,31,129,0,
```

```
1,78,18,6,17,83,196,177,99,97,
```

```
107,108,196,177,107,0,129,0,31,78,
```

```
13,6,17,78,101,109,0,67,4,3,
```

```
87,20,5,2,26,11,67,4,29,87,
```



```
20,5,2,26,11 };
```

```
// this structure defines all the variables and events of your control interface
```

```
struct {
```

```
    // input variables
```

```
    uint8_t button_1; // =1 if button pressed, else =0
```

```
    uint8_t switch_1; // =1 if switch ON and =0 if OFF
```

```
    int8_t joystick_1_x; // =-100..100 x-coordinate joystick position
```

```
    int8_t joystick_1_y; // =-100..100 y-coordinate joystick position
```

```
    // output variables
```

```
    char text_1[11]; // string UTF8 end zero
```

```
    char text_2[11]; // string UTF8 end zero
```

```
    char text_3[11]; // string UTF8 end zero
```

```
    // other variable
```

```
    uint8_t connect_flag; // =1 if wire connected, else =0
```

```
} RemoteXY;
```

```
#pragma pack(pop)
```

```
////////////////////////////////////
```

```
//      END RemoteXY include      //
```

```
////////////////////////////////////
```

```
#define PIN_BUTTON_1 13
```

```
#define PIN_SWITCH_1 10

int otomasyonacisi=1500;

void setup()
{
    RemoteXY_Init ();

    pinMode (PIN_BUTTON_1, OUTPUT);
    pinMode (PIN_SWITCH_1, OUTPUT);
    otomasyon.attach(9);
    // TODO you setup code

}

void loop()
{
    RemoteXY_Handler ();

    digitalWrite(PIN_BUTTON_1, (RemoteXY.button_1==0)?LOW:HIGH);
    digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIGH);
    int potasyookunan=analogRead(A0);
    dtostrf(potasyookunan,0,2,RemoteXY.text_1);
    otomasyon.writeMicroseconds(RemoteXY.joystick_1_x*10 + otomasyonacisi);
    float temp=dht.readTemperature();
    float hum=dht.readHumidity();
    dtostrf(temp,0,1,RemoteXY.text_2);
    dtostrf(hum,0,1,RemoteXY.text_3);
```

```
// TODO you loop code

// use the RemoteXY structure for data transfer

// do not call delay()


}
```