



BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

Nesnelerin İnterneti ve Uygulamaları

Proje Ödevi

Dersi Verenler : Prof. Dr. Cüneyt BAYILMIŞ
Doç. Dr. Murat İSKEFİYELİ

Dersi Alan : Şimal Ece KAZDAL

Öğrenci No : G221210068

Ders Grubu : 2A

Dersi Alan : Okan BAŞOL

Öğrenci No : G211210083

Ders Grubu : 2B

İçindekiler

Problemin Tanımı.....	3
Problemin Çözümü.....	3
Big Data (Büyük Veri).....	3
Business Canva İş Modeli.....	5
Kullanılan Malzemeler ve Teknolojiler.....	6
NodemCU.....	6
Arduino Uno R3.....	6
DHT11.....	6
MQ2.....	6
Arduino 6V 250RPM Redüktörlü Araba Motoru.....	6
Arduino Lazer Modül.....	6
Buzzer.....	6
Şasi.....	6
Tekerlek.....	6
Motor Sürücü Kartı L298n.....	6
HC-SR04 Arduino Ultrasonik Mesafe Sensörü.....	6
Arduino IDE.....	6
MIT Inventor App.....	6
Maliyet Tablosu.....	7
Devre Şeması.....	8
Devre Tasarımı.....	9
Kaynak Kodları.....	10
Uygulama Ekran Görüntüleri.....	18
Kaynakça.....	20

Problemin Tanımı

Çocukların bulunduğu ortamların güvenilirliğini kontrol etmek.

Problemin Çözümü

Ortam Kontrol Aracı ile çocuk örneğin parkta oyun oynarken ortamda yanıcı bir gaz tespit edilirse buzzer devreye girerek çocuğa ve çevrediklere haber vermiş olacak. Aynı zamanda ortam sıcaklığı kaydedilerek uzaktan ebeveynler tarafından kontrol edilmesi sağlanacak. Böylece hava koşulları olumsuz şekilde değişirse buna göre tedbir alınabilecek. Araç engel görürse lazer devreye girecek ve böylece çocukların yerdeki taş veya engelleri görmeden takılmasının önüne geçilmiş olacak. Güçlü bir ışık kaynağı olduğundan fark edilmesi daha kolay olacaktır.

Big Data (Büyük Veri)

Bu projede Big Data kavramı, ThingSpeak platformu ve sensörlerden gelen verilerin IoT (Nesnelerin İnterneti) bağlamında işlenmesi ve anlamlandırılması ile ilişkilidir.

Bu proje, aşağıdaki Big Data süreçlerini uygulamaktadır:

a. Veri Toplama

- **Kaynaklar:**
 - **DHT11 Sensörü:** Sıcaklık ve nem verileri sağlar.
 - **WiFi Modülü:** ESP8266, sensörden gelen verileri bir platforma (ThingSpeak) aktarır.
 - **Web Sunucusu:** Motor komutlarını almak için kullanılır.
- **Örnek Veri:**
 - Sıcaklık: 25.6 °C
 - Nem: 60%
 - Kullanıcıdan gelen motor komutları (örneğin, ileri git, sağa dön).

b. Veri Gönderimi ve Saklama

- Sensörlerden okunan veriler, ThingSpeak platformuna aktarılır:
 - **Sıcaklık ve Nem Değerleri:** Kanalin ilgili alanlarına yazılır (field 1 ve field 2).
 - **API Kullanımı:** ThingSpeak.writeField() ile sıcaklık ve nem değerleri gönderilir.

- **ThingSpeak'in Rolü:**

- Verileri saklar.
- Grafiksel görselleştirme ve analiz için geçmiş veri saklama sağlar.
- Büyük hacimli IoT verilerini yönetir.

c. Veri İşleme

- Veri gönderilirken ve okunurken anlamlı bir bağlamda işlenir:
 - **Gönderim:** Sensör verileri ThingSpeak'e yazılır.
 - **Okuma:** ThingSpeak'ten geçmiş sıcaklık ve nem verileri okunur.
- Kullanıcıya ThingSpeak'ten alınan veriler sunulur. Bu işlem, geçmiş verilerle analiz yapmayı mümkün kılar.

d. Veri Analizi

- ThingSpeak, bir veri analitik platformu olarak kullanılır:
 - Sıcaklık ve nem değişimleri, zaman serisi grafikleri ile analiz edilir.
 - Anlık veri uyarıları veya geçmiş veri trendlerini tespit etmek mümkündür.

Business Canvas Model

Key Partners

Şimal Ece KAZDAL
Okan BAŞOL

Key Activities

- Sistem Tasarımı: Donanım ve yazılımın entegrasyonu ile bir robotik sistem oluşturulması.
- Yazılım Geliştirme: Robotun hareketi, engellerden kaçınması, gaz algılaması ve IoT entegrasyonu için kod yazılması.
- Test ve Hata Ayıklama: Sistemin farklı koşullarda güvenilir çalışması için test edilmesi ve sorunların giderilmesi.
- Dokümantasyon Hazırlığı: Kullanım kılavuzları, montaj talimatları ve sorun giderme rehberlerinin oluşturulması.

Value Propositions

Projeimiz parklarda, ev, sınıt gibi ortamlarda daha çok çocukların sağlık ve güvenliğini sağlamak amacıyla geliştirilmiştir. Çocuk parkta oynarken araç da yanında giderek ortamda oluşması halinde gerekli uyarıları verecektir. Bunun dışında ortam sıcaklığı ve nemli uzaktan kontrol edilebilmektedir.

Customer Relationships

Öncelikli olarak ebeveynlere bu projenin ne amaçla yapıldığını detaylıca anlatacağız. Burada amaç anne ve babaların çocuklarının yanında olmaları ve kontrol edebilmelerini sağlamaktır. Bu detaylıca anlatılacaktır. Bazı okullarla da bağışta gezmesi için okul müdürleri ile görüşmeler sağlanacaktır.

Customer Segments

- Eğitim Kurumları: Robotik ve IoT konularında uygulamalı eğitim vermek isteyen okullar ve üniversiteler.
- Maker Topluluğu: DIY projelerine ve robotik sistemlere ilgi duyan hobbiseverler.
- IoT Meraklıları: IoT çözümlerini robotik uygulamalara keşfetmek isteyen bireyler.
- STEM Eğitim Programları: Bilim, teknoloji, mühendislik ve matematik odaklı eğitim girişimleri.

Key Resources

- NodeMCU
- Arduno uno r3
- Hava Sıcaklık ve Nem Sensörü (DHT11)
- Gaz Sensörü (MQ2)
- Lazer Modülü
- Buzzer
- Motor Sürücü Kartı (L298n)
- Güç Kaynağı
- Redüktörlü Araba Motoru

Channels

Okullara, bazı sosyal tesislere ve belediyelere parklar için reklam verilecek, sosyal medya ve web sitelerinden duyurulacaktır. Daha cazibedici olması için başka bir müşterinin referansı ile gelen müşterilere indirim uygulanacaktır.

Cost Structure

- Donanım Maliyetleri: Sensörler, motorlar, kontrolcüler ve diğer bileşenlerin maliyetleri.
- Geliştirme Maliyetleri: Yazılım geliştirme ve test süreçlerinde harcanan zaman ve emek.
- Tanıtım Maliyetleri: Eğitim etkinlikleri düzenlemek, çevrimiçi içerik üretmek ve reklam faaliyetleri.
- Bakım ve Destek: Kullanıcılara sağlanan teknik destek hizmetleri.
- bunlara bağlı olarak ürün fiyatı belirlenecektir.

Revenue Streams

- Ürün Satışı: Robotik sistemin önceden monte edilmiş veya DIY kiti şeklinde satılması.
- Eğitim Atölyeleri: Eğitim kurumlarına yönelik uygulamalı robotik ve IoT eğitimleri düzenlenmesi.
- Premium İçerik: Detaylı rehberler, ileri düzey modüller veya özel işlevler için ücretli erişim.
- Ortaklık ve Sponsorluk: Eğitim veya teknoloji alanındaki şirketlerle iş birliği ve sponsorluklar.
- Bu şekilde satış stratejileri düşünülmüştür.

Business Canvas İş Modeli

Kullanılan Malzemeler ve Teknolojiler

NodeMCU

ESP8266 tabanlı bir Wi-Fi modülü olan NodeMCU, IoT projelerinde kullanılan bir mikrodenetleyicidir. Kablosuz bağlantı desteği ve GPIO pinleriyle çeşitli sensör ve cihazları kontrol edebilir.

Arduino Uno R3

Atmega328 mikrodenetleyiciye sahip popüler bir geliştirme kartıdır. 14 dijital giriş/çıkış pini, 6 analog giriş pini ve basit programlama desteği sunar.

DHT11

Sıcaklık ve nem ölçümü yapabilen bir sensördür. Düşük maliyetlidir ve temel projelerde kullanılabilir, ancak yüksek doğruluk gerektiren durumlar için sınırlıdır.

MQ2

Gaz algılama sensörüdür. LPG, karbon monoksit, metan gibi gazları tespit edebilir. Ev güvenlik sistemlerinde ve gaz sızıntı alarmı projelerinde kullanılır.

Arduino 6V 250RPM Redüktörlü Araba Motoru

6V giriş voltajıyla çalışan ve 250RPM hızında dönebilen redüktörlü DC motor. Robot projelerinde tekerlekleri hareket ettirmek için kullanılır.

Arduino Lazer Modül

Lazer ışını üreten küçük bir modüldür. Robotik projelerde işaretleme, mesafe algılama veya görsel efektler için kullanılır.

Buzzer

Ses çıkaran bir modüldür. Alarm, uyarı veya bildirim gibi sesli geri bildirim vermek için kullanılır.

Şasi

Robotik sistemlerde tüm bileşenlerin monte edildiği temel yapıdır. Genellikle plastik veya metalden yapılır.

Tekerlek

Robotik araçlarda hareketi sağlamak için motorlarla birlikte kullanılan mekanik bileşenlerdir. Projeye uygun boyut ve malzemede seçilir.

Motor Sürücü Kartı L298n

Çift H-köprüsü motor sürücüsü, DC motorların hızını ve yönünü kontrol etmek için kullanılır. 2 motoru bağımsız olarak kontrol edebilir.

HC-SR04 Arduino Ultrasonik Mesafe Sensörü

Yüksek frekanslı ses dalgaları yayarak ve alarak ve aradaki zaman. atlamasını ölçerek nesnelerin mesafesini algılar ve ölçebilir.

Arduino IDE

Arduino ve diğer mikrodenetleyicilerin programlanması için kullanılan ücretsiz bir geliştirme ortamıdır. Basit bir arayüz sunar ve geniş bir kütüphane desteği vardır.

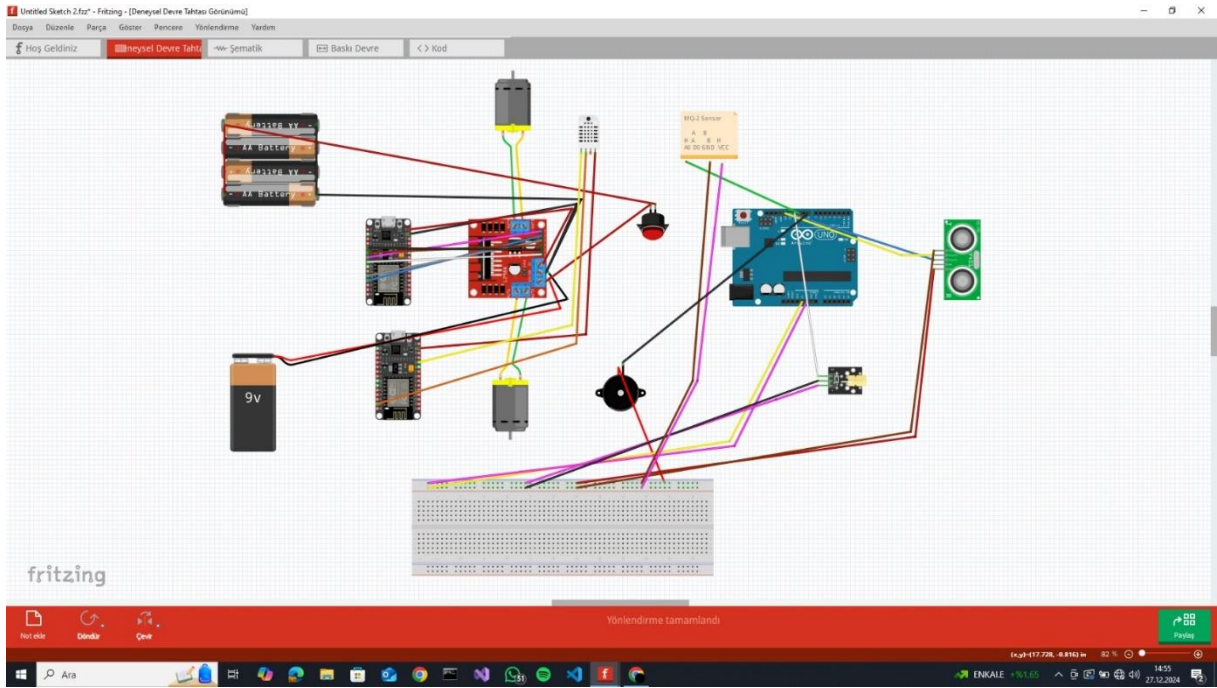
MIT App Inventor

Google tarafından ortaya çıkarılan ve sonrasında Massachusetts Institute of Technology (MIT) tarafından geliştirilen, özgür bir uygulama

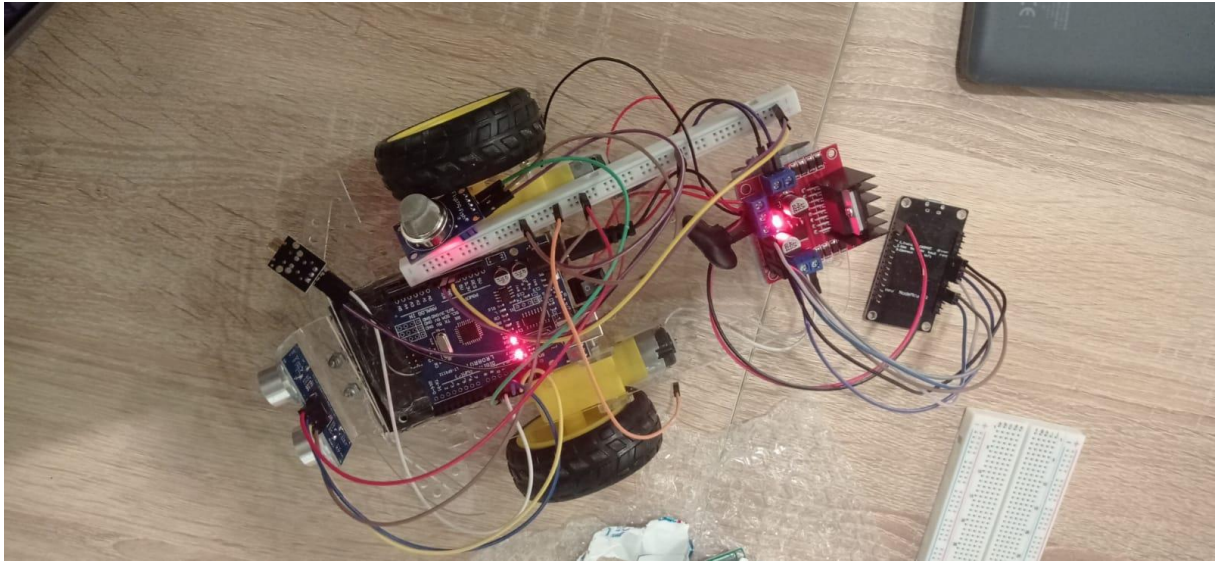
Maliyet Tablosu

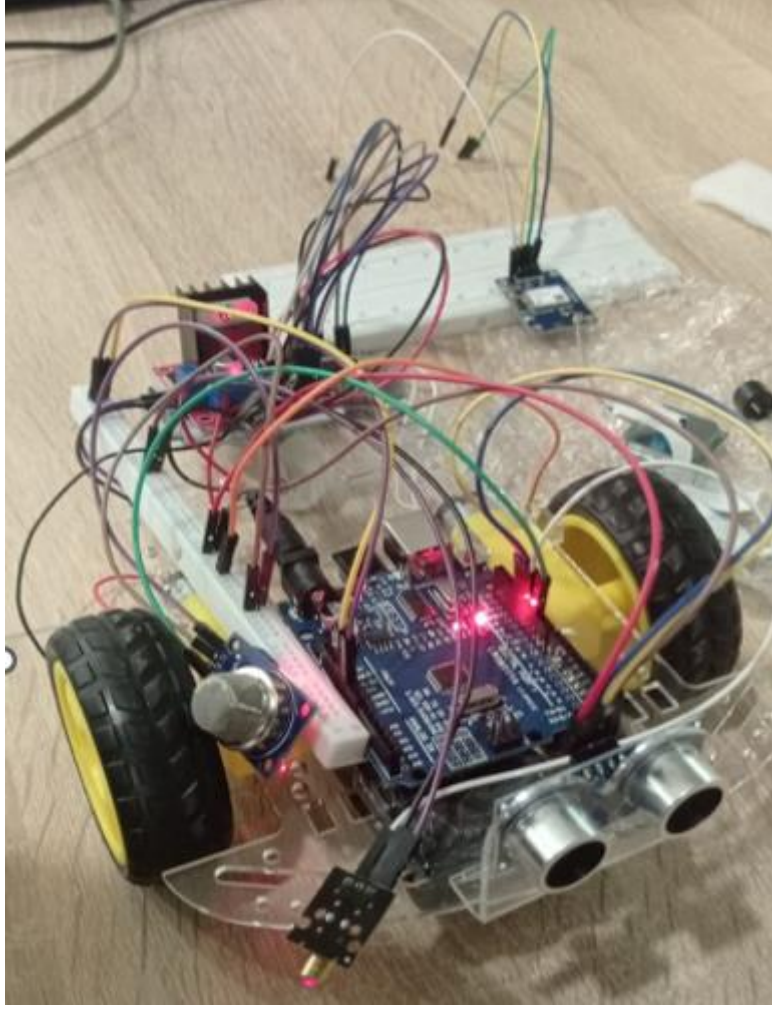
Arduino Uno R3	1 adet	183 ₺
DHT11	1 adet	35 ₺
MQ2	1 adet	51 ₺
Arduino 6V 250RPM Redüktörlü Araba Motoru	2 adet	100 ₺
NodeMCU	1 adet	202 ₺
Arduino Lazer Modül	1 adet	16 ₺
Buzzer	1 adet	48 ₺
Şasi	1 adet	33 ₺
Tekerlekler	3 adet	125 ₺
Motor Sürücü Kartı L298n	1 adet	75 ₺
Breadbord + kablolar	belirsiz	250 ₺
PowerBank	1 adet	450 ₺
Bant	1 adet	35 ₺
9V Pil	1 adet	49 ₺
1,5V Pil	4 adet	147 ₺
HC-SR04 Arduino Ultrasonik Mesafe Sensörü	1 adet	56 ₺
Toplam		1855 ₺

Devre Şeması



Devre Tasarımı





Kaynak Kodları

```
#include <ESP8266WiFi.h>

#include "ThingSpeak.h"

#include "DHT.h" // DHT11 Sensör için kütüphane

#define WLAN_SSID "Okan"

#define WLAN_PASSWORD "okan12345"

unsigned long channelID = 2795688; // ThingSpeak kanal ID

const char* writeAPIKey = "CW4RIARGZQ9W48OI"; // ThingSpeak yazma API anahtarı

const char* serverTS = "api.thingspeak.com"; // ThingSpeak yazma API anahtarı

#define DHTPIN D1 // DHT11 Sensörünün bağlı olduğu pin (D1 örnektir, kendi pininizi yazın)

#define DHTTYPE DHT11 // Sensör tipi (DHT11 için sabit)

DHT dht(DHTPIN, DHTTYPE);

WiFiClient client;

void wifiSetup() {

    Serial.println("\nWiFi Bağlantısı Başlatılıyor...");
```

```

WiFi.begin(WLAN_SSID, WLAN_PASSWORD);

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print("."); }

Serial.println("\nWiFi Bağlantısı Sağlandı!");

Serial.print("IP Adresi: ");

Serial.println(WiFi.localIP());}

void setup() {

    Serial.begin(9600);          // Seri Haberleşme başlatılıyor

    wifiSetup();                // WiFi kurulum fonksiyonu

    ThingSpeak.begin(client);   // ThingSpeak başlatılıyor

    dht.begin();                // DHT Sensörü başlatılıyor

    Serial.println("Kurulum Tamamlandı.");}

#define ENA  14      // Enable/speed motors Right    GPIO14(D5)

#define ENB  12      // Enable/speed motors Left     GPIO12(D6)

#define IN_1 15      // L298N in1 motors Right      GPIO15(D8)

#define IN_2 13      // L298N in2 motors Right      GPIO13(D7)

#define IN_3 2       // L298N in3 motors Left       GPIO2(D4)

#define IN_4 0       // L298N in4 motors Left       GPIO0(D3)

#include <ESP8266WebServer.h>

String command;               //String to store app command state.

int speedCar = 800;           // 400 - 1023.

int speed_Coeff = 3;


const char* ssid = "NodeMCU Car";

ESP8266WebServer serverMotor(80);

void setupMotorServer() {

    pinMode(ENA, OUTPUT);

    pinMode(ENB, OUTPUT);

    pinMode(IN_1, OUTPUT);

    pinMode(IN_2, OUTPUT);

    pinMode(IN_3, OUTPUT);

    pinMode(IN_4, OUTPUT);

    Serial.begin(115200);

    WiFi.mode(WIFI_AP);

    WiFi.softAP(ssid);

    IPAddress myIP = WiFi.softAPIP();

    Serial.print("AP IP address: ");

    Serial.println(myIP);

```

```

serverMotor.on ( "/", HTTP_handleMotorRoot );

serverMotor.onNotFound ( HTTP_handleMotorRoot );

serverMotor.begin();  }

void goAhead(){
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, HIGH);
  analogWrite(ENA, speedCar);
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, HIGH);
  analogWrite(ENB, speedCar);}

void goBack(){
  digitalWrite(IN_1, HIGH);
  digitalWrite(IN_2, LOW);
  analogWrite(ENA, speedCar);
  digitalWrite(IN_3, HIGH);
  digitalWrite(IN_4, LOW);
  analogWrite(ENB, speedCar);}

void goRight(){
  digitalWrite(IN_1, HIGH);
  digitalWrite(IN_2, LOW);
  analogWrite(ENA, speedCar);
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, HIGH);
  analogWrite(ENB, speedCar);}

void goLeft(){
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, HIGH);
  analogWrite(ENA, speedCar);
  digitalWrite(IN_3, HIGH);
  digitalWrite(IN_4, LOW);
  analogWrite(ENB, speedCar);}

void stopRobot(){
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, LOW);
  analogWrite(ENA, speedCar);
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, LOW);
  analogWrite(ENB, speedCar);}

void loop() {

```

```

serverMotor.handleClient(); // Motor serverini dinleyin

sicaklikVeNem();

delay(2000); // 2 saniye bekle (ThingSpeak limiti için)

void sicaklikVeNem() {

    float sicaklik = dht.readTemperature(); // Santigrat sıcaklık okuma

    float nem = dht.readHumidity(); // Nem okuma

    if (isnan(sicaklik) || isnan(nem)) { // Sensörden değer okunamıyorsa hata mesajı

        Serial.println("DHT11'dan değer okunamadı! Lütfen bağlantıyı kontrol edin.");

        return;

    }

    Serial.print("Hesaplanan Sıcaklık: ");

    Serial.print(sicaklik);

    Serial.println(" °C");

    Serial.print("Hesaplanan Nem: ");

    Serial.print(nem);

    Serial.println(" %");

    Serial.println("ThingSpeak'e sıcaklık ve nem değerleri gönderiliyor...");

    int sicaklikDurum = ThingSpeak.writeField(channelID, 1, sicaklik, writeAPIKey); // 1. field'a sıcaklık

    int nemDurum = ThingSpeak.writeField(channelID, 2, nem, writeAPIKey); // 2. field'a nem

    if (sicaklikDurum == 200 && nemDurum == 200) {

        Serial.println("Veriler başarıyla gönderildi!");

    } else {

        Serial.print("Veri gönderme başarısız! Sıcaklık Hata Kodu: ");

        Serial.println(sicaklikDurum);

        Serial.print("Nem Hata Kodu: ");

        Serial.println(nemDurum);

    }

    float okunanSicaklik = ThingSpeak.readFloatField(channelID, 1); // 1. field'dan sıcaklık

    float okunanNem = ThingSpeak.readFloatField(channelID, 2); // 2. field'dan nem

    Serial.print("ThingSpeak'ten Okunan Sıcaklık: ");

    Serial.println(okunanSicaklik);

    Serial.print("ThingSpeak'ten Okunan Nem: ");

    Serial.println(okunanNem);

}

void HTTP_handleMotorRoot(void) {

    if (serverMotor.hasArg("State")) {

        Serial.println(serverMotor.arg("State"));

    }

    serverMotor.send(200, "text/html", "");

    delay(1);
}

```

Motor

```

#define ENA 14 // Enable/speed motors Right GPIO14(D5)

#define ENB 12 // Enable/speed motors Left GPIO12(D6)

```

```

#define IN_1 15      // L298N in1 motors Right    GPIO15(D8)
#define IN_2 13      // L298N in2 motors Right    GPIO13(D7)
#define IN_3 2       // L298N in3 motors Left     GPIO2(D4)
#define IN_4 0       // L298N in4 motors Left     GPIO0(D3)

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

String command;      //String to store app command state.

int speedCar = 800;   // 400 - 1023.

int speed_Coeff = 3;

const char* ssid = "NodeMCU Car";

ESP8266WebServer server(80);

void setup() {

  pinMode(ENA, OUTPUT);

  pinMode(ENB, OUTPUT);

  pinMode(IN_1, OUTPUT);

  pinMode(IN_2, OUTPUT);

  pinMode(IN_3, OUTPUT);

  pinMode(IN_4, OUTPUT);

  Serial.begin(115200);

  WiFi.mode(WIFI_AP);

  WiFi.softAP(ssid);


  IPAddress myIP = WiFi.softAPIP();

  Serial.print("AP IP address: ");

  Serial.println(myIP);

  server.on ( "/", HTTP_handleRoot );

  server.onNotFound ( HTTP_handleRoot );

  server.begin(); }

void goAhead(){

  digitalWrite(IN_1, LOW);

  digitalWrite(IN_2, HIGH);

  analogWrite(ENA, speedCar);

  digitalWrite(IN_3, LOW);

  digitalWrite(IN_4, HIGH);

  analogWrite(ENB, speedCar);}

void goBack(){

  digitalWrite(IN_1, HIGH);

  digitalWrite(IN_2, LOW);

```

```

    analogWrite(ENA, speedCar);

    digitalWrite(IN_3, HIGH);

    digitalWrite(IN_4, LOW);

    analogWrite(ENB, speedCar); }

void goRight(){

    digitalWrite(IN_1, HIGH);

    digitalWrite(IN_2, LOW);

    analogWrite(ENA, speedCar);

    digitalWrite(IN_3, LOW);

    digitalWrite(IN_4, HIGH);

    analogWrite(ENB, speedCar);}

void goLeft(){

    digitalWrite(IN_1, LOW);

    digitalWrite(IN_2, HIGH);

    analogWrite(ENA, speedCar);

    digitalWrite(IN_3, HIGH);

    digitalWrite(IN_4, LOW);

    analogWrite(ENB, speedCar); }

void goAheadRight(){

    digitalWrite(IN_1, LOW);

    digitalWrite(IN_2, HIGH);

    analogWrite(ENA, speedCar/speed_Coeff);

    digitalWrite(IN_3, LOW);

    digitalWrite(IN_4, HIGH);

    analogWrite(ENB, speedCar); }

void goAheadLeft(){

    digitalWrite(IN_1, LOW);

    digitalWrite(IN_2, HIGH);

    analogWrite(ENA, speedCar);

    digitalWrite(IN_3, LOW);

    digitalWrite(IN_4, HIGH);

    analogWrite(ENB, speedCar/speed_Coeff);}

void goBackRight(){

    digitalWrite(IN_1, HIGH);

    digitalWrite(IN_2, LOW);

    analogWrite(ENA, speedCar/speed_Coeff);

    digitalWrite(IN_3, HIGH);

    digitalWrite(IN_4, LOW);

    analogWrite(ENB, speedCar); }

```

```

void goBackLeft(){
    digitalWrite(IN_1, HIGH);
    digitalWrite(IN_2, LOW);
    analogWrite(ENA, speedCar);
    digitalWrite(IN_3, HIGH);
    digitalWrite(IN_4, LOW);
    analogWrite(ENB, speedCar/speed_Coeff); }

void stopRobot(){
    digitalWrite(IN_1, LOW);
    digitalWrite(IN_2, LOW);
    analogWrite(ENA, speedCar);
    digitalWrite(IN_3, LOW);
    digitalWrite(IN_4, LOW);
    analogWrite(ENB, speedCar); }

void loop() {
    server.handleClient();

    command = server.arg("State");
    if (command == "F") goAhead();
    else if (command == "B") goBack();
    else if (command == "L") goLeft();
    else if (command == "R") goRight();
    else if (command == "I") goAheadRight();
    else if (command == "G") goAheadLeft();
    else if (command == "J") goBackRight();
    else if (command == "H") goBackLeft();
    else if (command == "0") speedCar = 400;
    else if (command == "1") speedCar = 470;
    else if (command == "2") speedCar = 540;
    else if (command == "3") speedCar = 610;
    else if (command == "4") speedCar = 680;
    else if (command == "5") speedCar = 750;
    else if (command == "6") speedCar = 820;
    else if (command == "7") speedCar = 890;
    else if (command == "8") speedCar = 960;
    else if (command == "9") speedCar = 1023;
    else if (command == "S") stopRobot();}

void HTTP_handleRoot(void) {
    if( server.hasArg("State") ){
        Serial.println(server.arg("State")); }
}

```

```
server.send ( 200, "text/html", "" );

delay(1);}

```

Arduino Uno

#define echoPin 12 // Ultrasonik sensörün echo pini Arduino kartımızın 12. pinine bağladık

#define trigPin 13 // Ultrasonik sensörün trig pini Arduino kartımızın 13. pinine bağladık.

#define buzzer 8 // Buzzer pinini 8'e bağlıyoruz

#define laserPin 11 // Lazer pinini 11'e bağlıyoruz

#define gasSensor A0 // Gaz sensörünü analog A0 pinine bağlıyoruz

long sure, uzaklik; // süre ve uzaklık adında iki değişken tanımlıyoruz

float h, t; // Sıcaklık ve nem değişkenleri

int gasValue = 0; // Gaz sensörü değeri

```
void setup() {

  pinMode(echoPin, INPUT);

  pinMode(trigPin, OUTPUT);

  pinMode(buzzer, OUTPUT); // Buzzer'ı çıkış olarak tanımlıyoruz

  pinMode(laserPin, OUTPUT); // Lazer pinini çıkış olarak tanımlıyoruz

  Serial.begin(9600); // Seri monitörü başlatıyoruz}

void loop() {

  digitalWrite(trigPin, LOW);

  delayMicroseconds(5);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  sure = pulseIn(echoPin, HIGH);

  uzaklik = sure / 29.1 / 2;

  Serial.print("Mesafe: ");

  Serial.println(uzaklik);

  gasValue = analogRead(gasSensor);

  Serial.print("Gaz Sensörü Değeri: ");

  Serial.println(gasValue);

  if (gasValue > 400) {

    digitalWrite(buzzer, HIGH); // Gaz algılandı, buzzer'ı çal

    Serial.println("UYARI: Gaz algılandı!");

  } else {

    digitalWrite(buzzer, LOW); // Gaz yok, buzzer'ı kapat  }

  if (uzaklik < 5) { // Mesafe 5 cm'den küçükse lazeri aç

    digitalWrite(laserPin, HIGH); // Lazer aktif

    Serial.println("UYARI: Lazer aktif! Çok yakına gelindi.");
  }
}
```



```

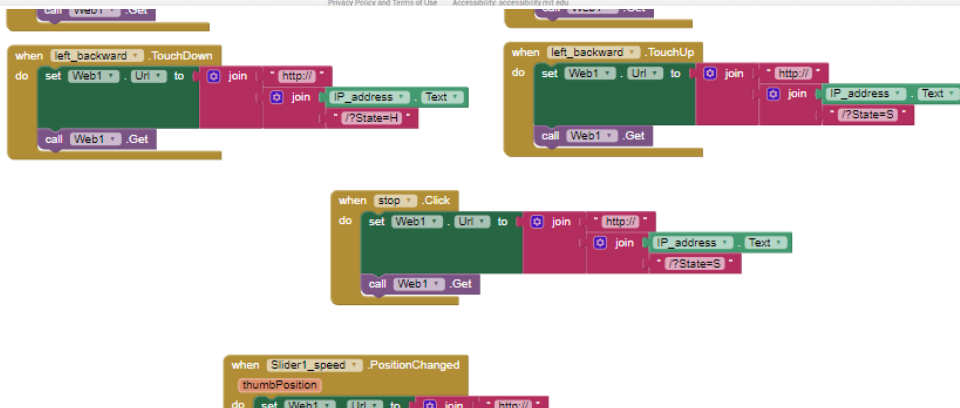
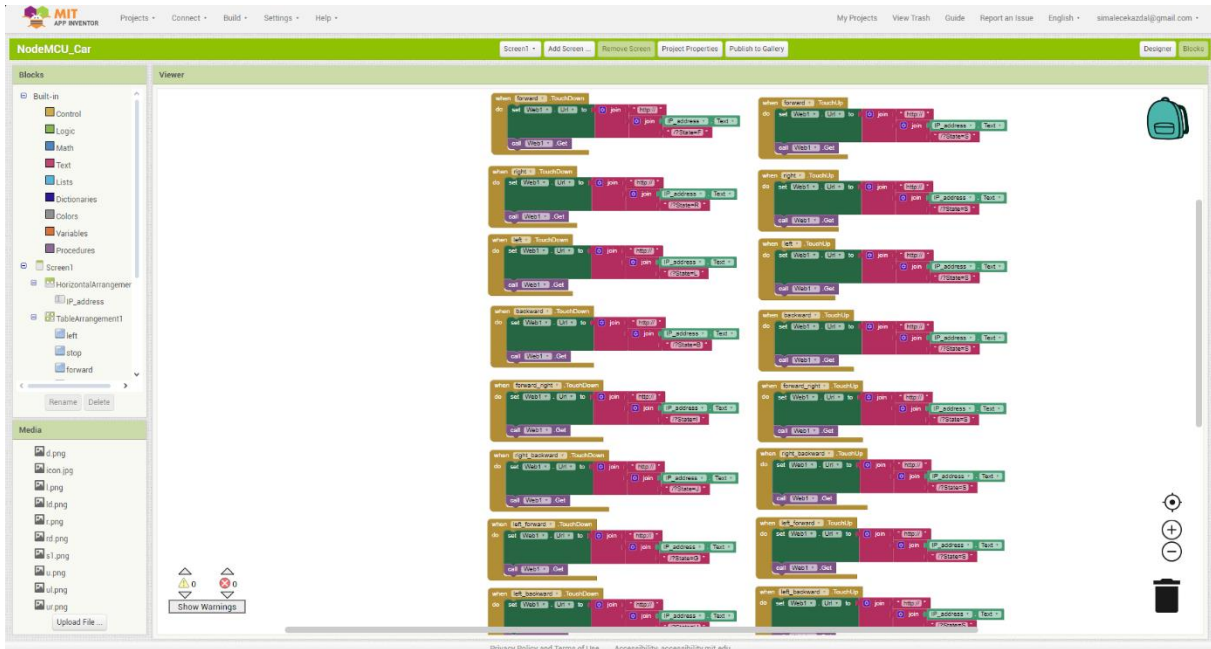
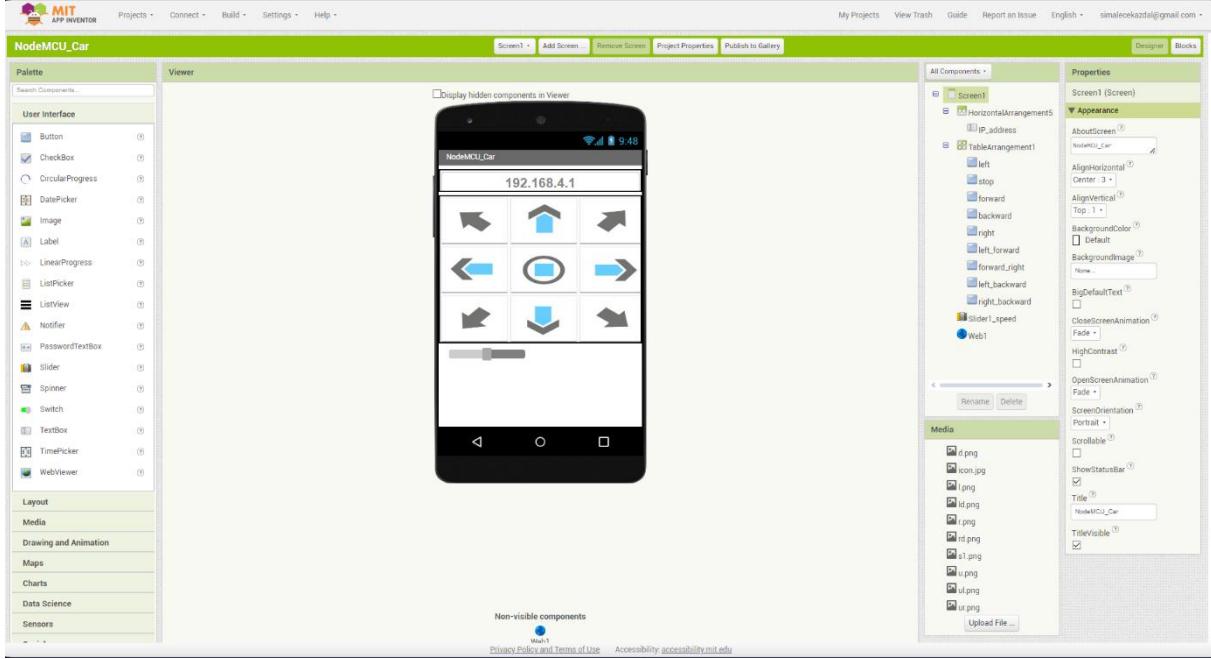
} else {

    digitalWrite(laserPin, LOW); // Lazer kapalı}

    delay(100); // Bir sonraki ölçüme geçmeden önce kısa bir gecikme}

```

Uygulama Ekran Görüntüleri



ThingSpeak™ Channels • Apps • Devices • Support • Commercial

Channel ID: 2795688
Author: mwa0000036369931
Access: Private

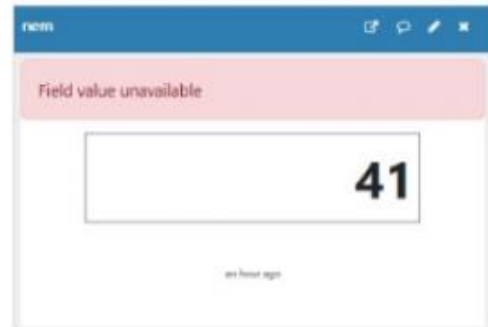
Private View Public View Channel Settings Sharing API Keys Data Import / Export

+ Add Visualizations + Add Widgets + Export recent data

MATLAB Analysis

Channel Stats

Created: about 8 hours ago
Last entry: about an hour ago
Entries: 106



Kaynakça

<https://www.nodemcu.com/>

<https://www.espressif.com/en/products/socs/esp8266>

<https://www.adafruit.com/>

<https://www.digikey.com/>

<https://www.robotistan.com/>

<https://www.st.com/en/motor-drivers/l298.html>

<https://www.arduino.cc/en/software>

<https://www.sparkfun.com/products/9403>

<https://www.st.com/en/motor-drivers/l298.html>