



SAKARYA ÜNİVERSİTESİ

NESNELERİN İNTERNETİ VE UYGULAMALARI

OTOPARK PROJESİ

Teslim Tarihi: 26.12.2021

RABİA ABDİOĞLU

1-B

B201210302

İÇİNDEKİLER

1. PROJE HAKKINDA	3
1.1 Projenin Amacı	2
1.2 Businnes Canvas Modeli ve Açıklaması	3
1.3 Uml Diyagramı	4
1.4 Gerekli Ekipmanlar.....	4
1.5 Gerçeklenmiş Prototipin Görselleri	5
1.6 Kodlar ve Açıklamaları	5

1. PROJE HAKKINDA

1.1 Projenin Amacı

Yapılan otopark projesinin çözmeyi amaçladığı sorunların en başında kaldırımların işgal edilmesi yer alıyor. Kaldırımları kullanması gereken yayalar, yoldan geçmek zorunda kalabiliyor. Bu durum yayalar ve yolda araba kullanan sürücüler içinde tehlike teşkil edebiliyor. Aynı zamanda engelli vatandaşların hayatını zorlaştırabiliyor. Çok büyük bir sorun gibi görünmeme sebebi toplumun durumu normalleştirilmesi veya alışmasından kaynaklanıyor olabilir.

Büyük cadde ve sokaklarda çok dikkat çekmeyen bu durum, dar sokaklarda daha maddi veya manevi kazalara sebep olabilir. Büyük şehirlerde kaldırımların doluluğu ve kişisel araç kullanımının fazla olması görüntü kirliliğine yol açabiliyor.

Aşağıda verilen görsellerde hem görüntü kirliliği hem de ilk görseldeki vatandaş gibi yayaların yolu kullanarak canını tehlikeye attığı bariz bir şekilde görülüyor.



Çözüm olarak geniş, büyük ve sağlam bir bina yapısında otopark yapılabilir. Yapılacak otopark akıllı sistemler ile döşenebilir. Güvenlik ve ücretin az olması, aracı olan vatandaşlara daha cazip gelir ve şehirlerin kaldırımları temiz ve güvenli hale getirilebilir.

1.2 Businnes Canvas Modeli



Problem	Çözüm
<div>Park yeri bulma zorluğu</div> <div>Park yerlerine fazla ücretler verilmesi</div> <div>Sokaklara park edilen çokça aracın yayalara kaldırım alanı bırakılmaması</div>	<div>Büyük, çok katlı otopark binaları yapılması</div>
Varolan Alternatifler	Kilit Metrikler
<div>Alternatif yok</div>	<div>İlk olarak küçük şehirlerde başlanabilir.</div> <div>Hedef 81 şehir</div>
Maliyet Yapısı	

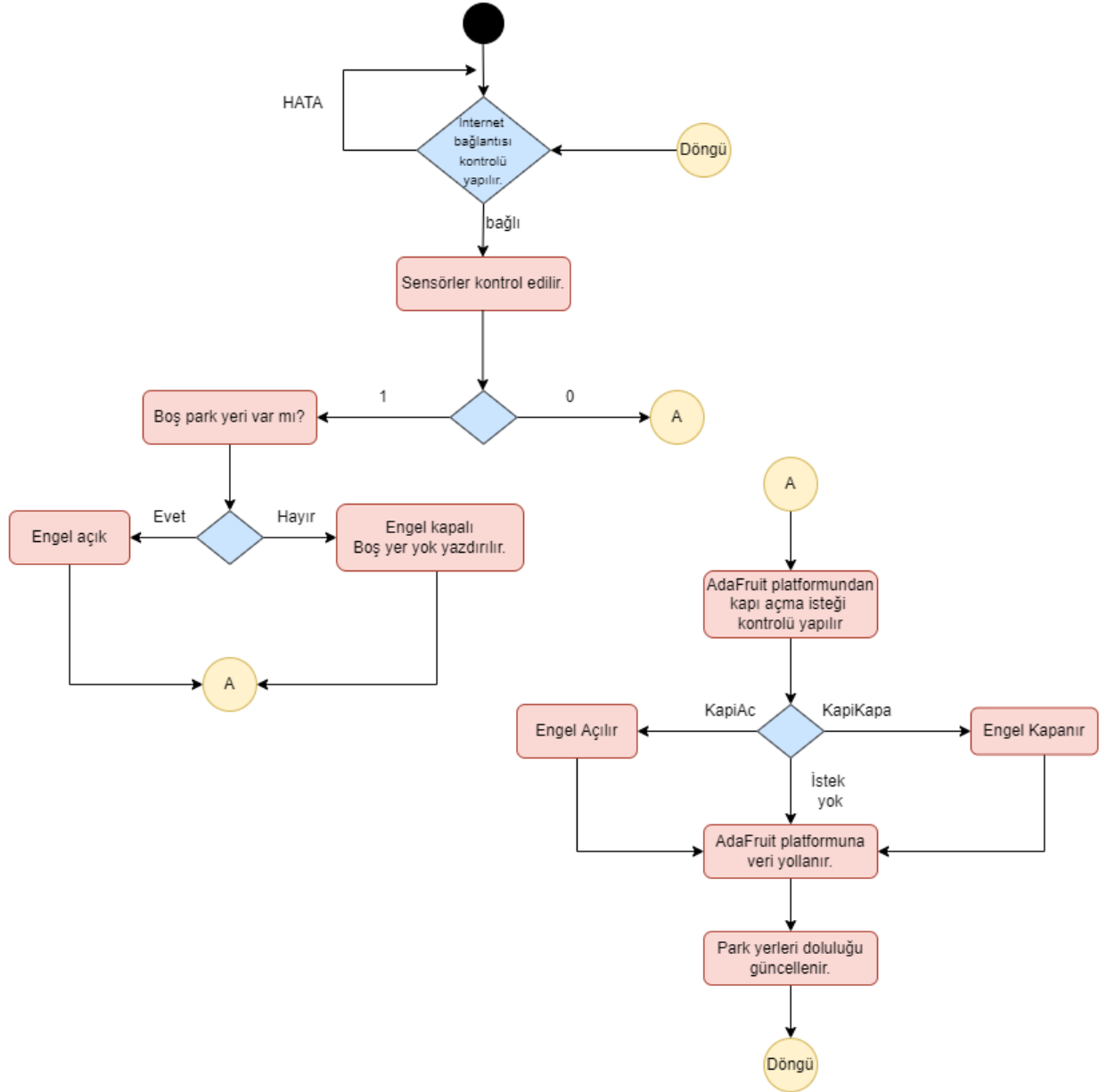
Otopark
Otopark Yönetim Sistemi

Benzersiz Değer Teklifi	Rekabet Avantajı	Hedef Kitle
<div>Güvenlik</div> <div>Düşük ücretler</div> <div>Boş park yeri için otopark içinde gezilmesine gerek yok.</div> <div>Mahallenin kaldırımları artık boş ve temiz</div>	<div>Fazla ve gereksiz ücretler alınmıyor</div> <div>Kanallar</div> <div>Reklam panoları ve inşaat önü afişler</div>	<div>Arabası olan herkes</div>
	<div>Gelir Kaynakları</div> <div>Otopark içi reklam alınabilir.</div>	

Model görseli sığmadığından iki parça olarak sunulmuştur.

Maliyet yapısı yatırımcılar ve desteklere bağlı olduğundan tam ücret çıkarılamamıştır.

1.3 Uml Diyagramı



1.4 Gerekli Ekipmanlar

Otopark sisteminin bir prototipi için gerekli ekipmanlar.

a) Donanım

- a.1. NodeMcu(esp8266)
- a.2. IR alıcı verici sensör
- a.3. Servo motor
- a.4. Jumper kablo
- a.5. Breadboard
- a.6. Rfid (kullanılmadı)

b) Yazılım & kütüphaneler

- b.1. Arduino Ide
 - b.1.1. Servo.h
 - b.1.2. Esp8266Wifi.h
 - b.1.3. Adafruit_MQTT.h ve Adafruit_MQTT_Client.h
 - b.1.4. WifiUdp.h
- b.2. Frizting
 - b.2.1. Fc-51.fzpz (IR sensör için)
 - b.2.2. NodeMCUv3Lolin.fzpz

c) Platform

- c.1. AdaFruit.io

1.5 Gerçeklenmiş Prototipin Görselleri

Adafruit platformunda yapılan, otoparkın yönetim ve takip sistemi görseli. Boş park yeri varsa giriş ekranında gözükür. Gözüken boş park yerleri her kat girişlerindedir. Park yeri numaraları araç sahiplerince görülebilir yerlerde olur. Böylece araç sahibinin, boş olan park yerini otopark boyunca aramasına gerek kalmaz.

Örnek park binası görseli :



AdaFruit.io Dashboard - Otopark yönetimi ve takibi için arayüz.

Kapat-Aç butonları devredeki servo motorlarının açılıp kapanmasını sağlıyor.

A1 ve A2 text alanları , park yerleri dolu ise “DOLU”, boş ise “ BOŞ “ yazısını gösteriyor.

BOS YER yazan gauge, boş park yeri sayısını gösteriyor. Otoparkın doluluk oranı, iki değil daha büyük sayılar ile daha rahat anlaşılabilir.

rabiaabdioglu > Dashboards > Otopark



Kapat

A1
🚗

A2
🚗

Aç



Fritzing devre tasarımı:

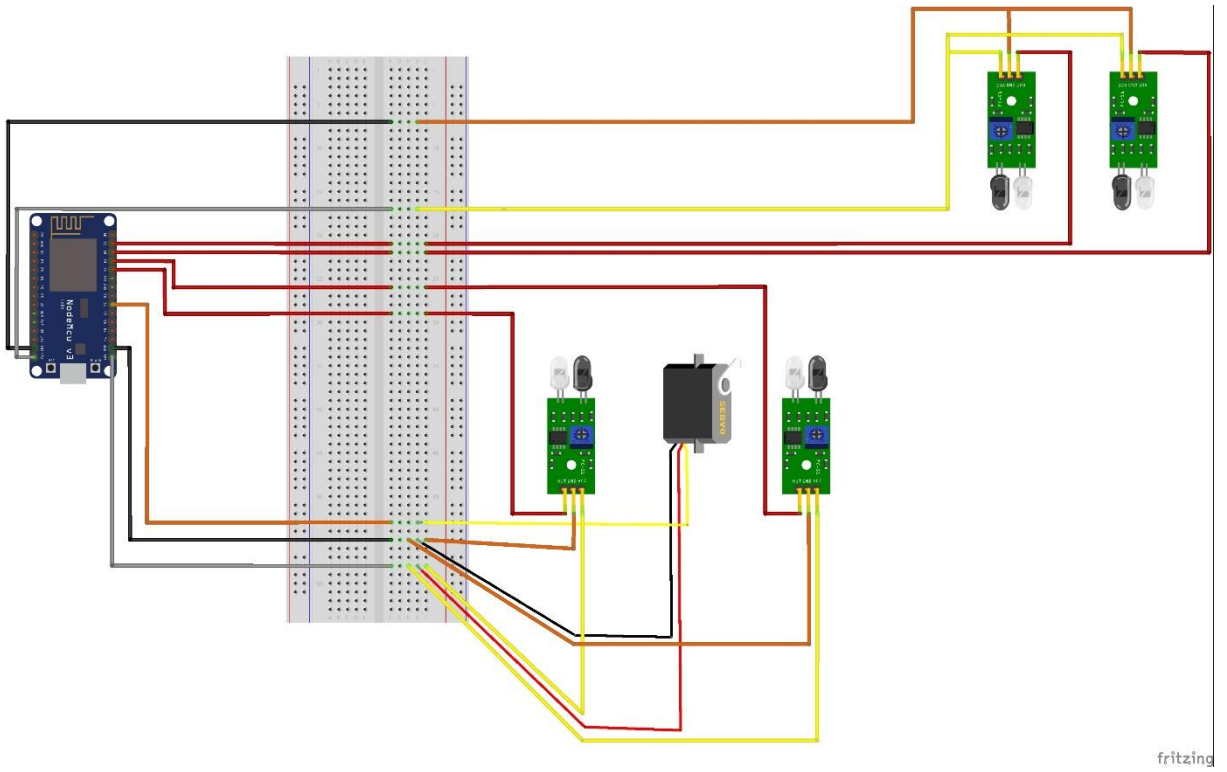
NodeMcu'da:

D1-D2-D3-D4 pinlerine, IR sensörler **kırmızı** kablo ile bağlıdır.

D6 pinine Servo motor **turuncu** kablo ile bağlıdır.

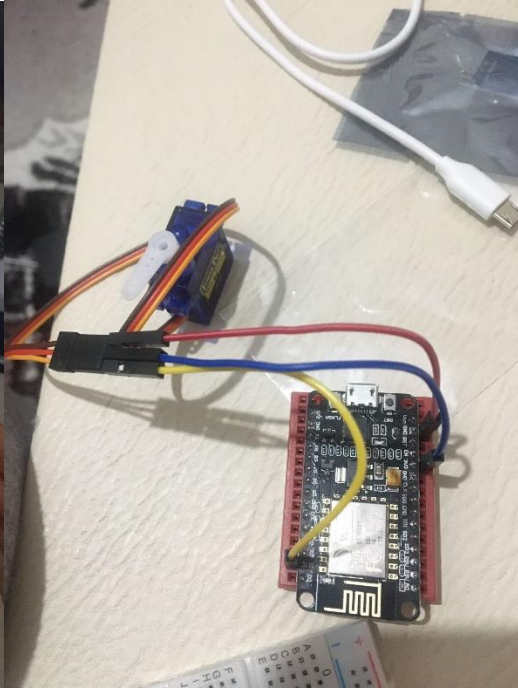
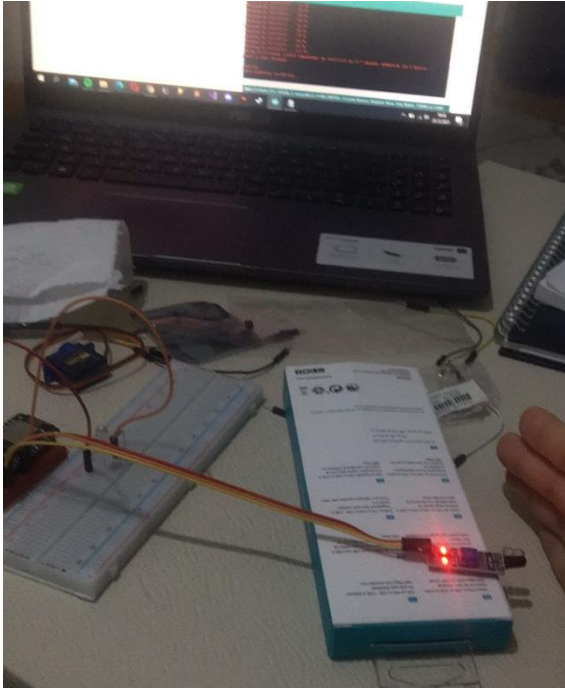
Servo motor, giriş ve çıkış sensörlerinin vcc ve gnd yolları birleştirilmiştir.

Park yerleri olan D1-D2 ye bağlı IR sensörlerin gnd ve vcc yolları birleştirilmiştir.



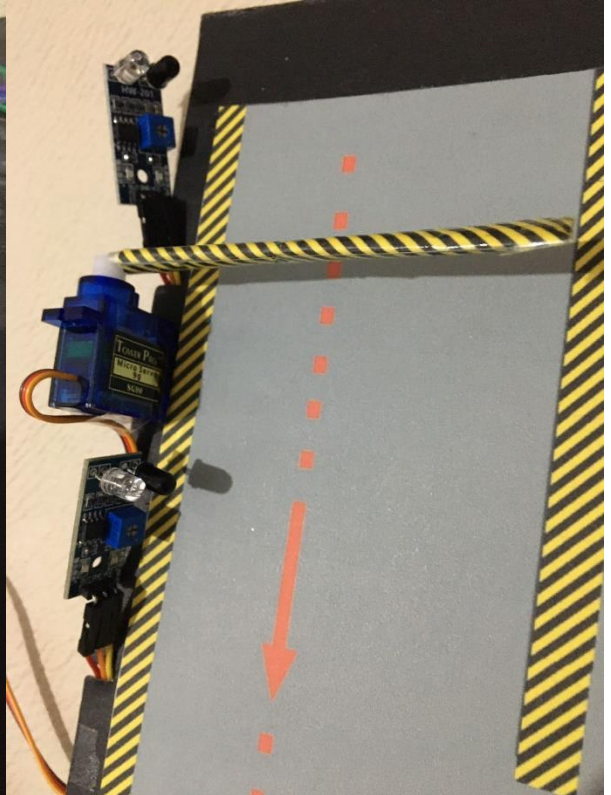
Prototip otopark projesi görselleri:

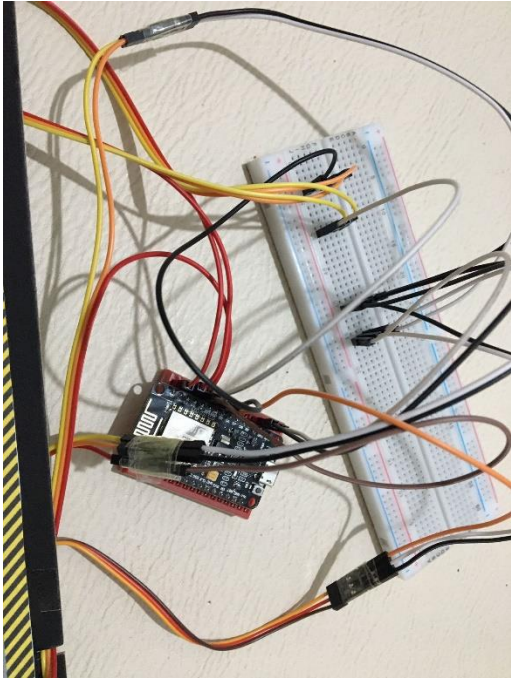
- IR sensör ve servo motor denemeleri



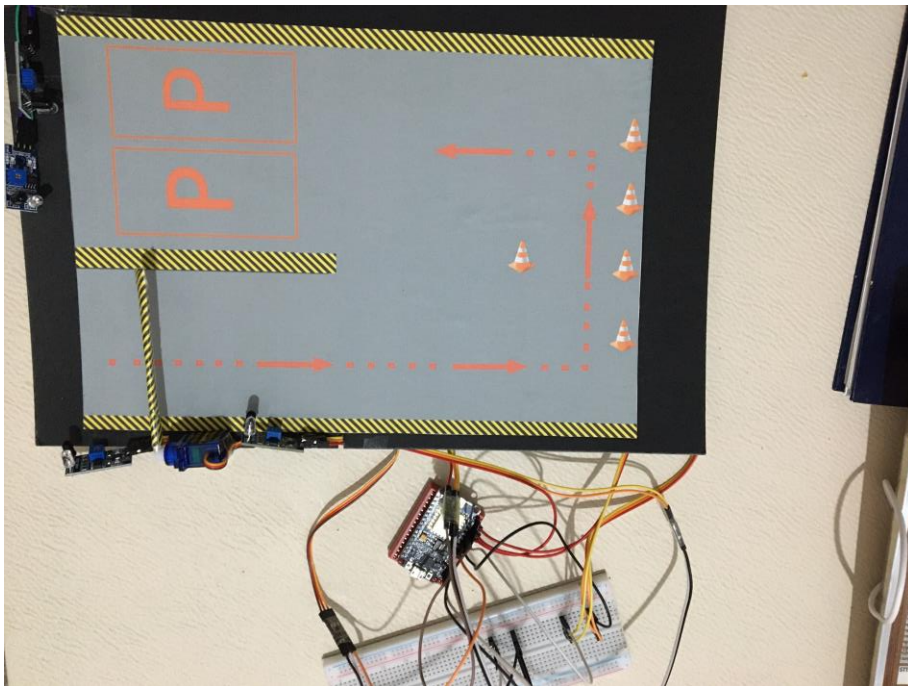
- Maket üzerinde ır sensör ve servo motor son hal

A close-up view of the servo motor and breadboard assembly. The servo motor is labeled "Tosin Pro Micro Servo". A yellow and black striped caution tape is visible on the right side.





- Son hal



1.6 Kodlar ve Açıklamaları

Kütüphane ekleme, wifi bağlantısı ayarlamaları, başlangıç değerleri ve AdaFruit için feed tanıtımı.

```
//kütüphaneler
#include <ESP8266WiFi.h>
#include <Servo.h>
#include <WiFiUdp.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

//
const char *ssid = "2.KAT"; // Enter your WiFi Name
const char *pass = "ccNC54rQ"; // Enter your WiFi Password

#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "rabiaabdioglu"
#define MQTT_PASS "aio_TvtQ03oBRJ8cWGjexnhWzFRp4mxJ"

WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME, MQTT_PASS);

//Feedler|

//buton
Adafruit_MQTT_Subscribe KapiAc = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME "/f/KapiAc");
Adafruit_MQTT_Subscribe KapiKapa = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME "/f/KapiKapa");

const char BosYerFeed[] PROGMEM = MQTT_NAME "/feeds/bosyer";

//park yerleri
Adafruit_MQTT_Publish ParkA1_Dolu = Adafruit_MQTT_Publish(&mqtt, MQTT_NAME "/f/ParkA1_Dolu");
Adafruit_MQTT_Publish ParkA2_Dolu = Adafruit_MQTT_Publish(&mqtt, MQTT_NAME "/f/ParkA2_Dolu");
Adafruit_MQTT_Publish BosYer = Adafruit_MQTT_Publish(&mqtt, BosYerFeed);

//başlangıç değerleri

int ir_cikis = 2; // D4 pini çıkış sensörü
int ir_giris = 0; // D3 pini giriş sensörü
int araba_say=0;
int park1 = 5; // D1 pini çıkış sensörü
int park2 = 4; // D2 pini giriş sensörü
String Dolu="DOLU";
String Bos="BOS";

Servo servo;
```

Setup içerisinde servo motorun pin değeri, mqtt ile feed , Ir sensörlerinin input pini ayarlaması ve wifi bağlantısı yapılır.

```

void setup()
{
  Serial.begin(9600);
  mqtt.subscribe(&KapiAc);
  mqtt.subscribe(&KapiKapa);
  servo.attach(12); //servo motor D6 pininde
  servo.write(0); //servo motor 0 dan başlasın
  pinMode(ir_giris, INPUT); // giriş sensörü
  pinMode(ir_cikis, INPUT); // çıkış sensörü

  //wifi bağlantısı
  WiFi.begin(ssid, pass);
  Serial.print("Baglaniliyor.... : ");
  Serial.print(ssid);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Baglandi. : ");
  Serial.println(ssid);
  Serial.print("IP Adresi : ");
  Serial.println(WiFi.localIP());
}

```

Loop içinde program düzeni daha iyi anlaşılması için fonksiyonlara parçalanmış.

```

void loop() {

  MQTT_connect(); //Mqtt bağlantısı yapılır.

  Sensor_Kontrol(); //Sensör kontrolüne göre engel açılıp kapatılır.
  AdaFruit_VeriYolla(); //Adafruit platformuna park yeri dolu-boş bilgisi yollanır
  AdaFruit_Kontrol(); // AdaFruit platformundan buton kontrolü

  int bosyer=digitalRead(park1)+digitalRead(park2);
  BosVer.publish((char*)bosyer); // Her döngü sonunda Adafruitte boş yer bilgisi yollanır

}

//gerekli fonksiyonlar
void Sensor_Kontrol(){
  if (digitalRead(ir_giris) == 0 )
  {
    if(ParkYeriSayisi()!=0)
    {
      Serial.print("Giris yapildi ");
      servo.write(120); delay(500);}
    else{ Serial.println("Boş park yeri yok"); }
  }

  else if (digitalRead(ir_cikis)==0)
  { // check if the input is HIGH
    Serial.print("Cikis yapildi");
    servo.write(120); delay(500);
  }
  else
  {
    Serial.println("Kapi kapaniyor..");

    | servo.write(0);
  }
}

```


Sensör kontrolü fonksiyonu ile engel otomatik olarak açılıp kapanıyor. Eğer boş yer yoksa giriş sensörü okusa bile engel açılmaz.

```
void AdaFruit_VeriYolla(){

////////// Park Yerleri boş dolu verisi yolla

    if (digitalRead(park1)==0){

        ParkA1_Dolu.publish((char*)Dolu.c_str());

    }
    else{ParkA1_Dolu.publish((char*)Bos.c_str());}

    if (digitalRead(park2)==0){

        ParkA2_Dolu.publish((char*)Dolu.c_str());

    }
    else{ParkA2_Dolu.publish((char*)Bos.c_str());}

int ParkYeriSayisi()
{
    int bosyer=digitalRead(park1)+digitalRead(park2);

    return bosyer;
}
```

VeriYolla fonksiyonu , Adafruit arayüzündeki A1 ve A2 park alanları için Dolu-Boş bilgisini yolluyor.

Kontrol fonksiyonu, platform üzerinden manuel engel aç veya kapa isteğini kontrol ediyor.

```
void AdaFruit_Kontrol(){
    Adafruit_MQTT_Subscribe * subscription;
    while ((subscription = mqtt.readSubscription(5000)))
    {
        //Kapi ac butonu
        if (subscription == &KapiAc)
        {
            Serial.println((char*) KapiAc.lastread);

            if (!strcmp((char*) KapiAc.lastread, "1"))
            {
                servo.write(120);
            }
        }

        //Kapi kapa butonu
        if (subscription == &KapiKapa)
        {
            Serial.println((char*) KapiKapa.lastread);

            if (!strcmp((char*) KapiKapa.lastread, "1"))
            {
                servo.write(0);
            } }
    }
}
```

```
//mqtt bağlantısı
void MQTT_connect()
{
    int8_t ret;

    if (mqtt.connected()) //bağlı mı ?
    {
        return;
    }

    uint8_t retries = 3;

    while ((ret = mqtt.connect()) != 0)
    {
        mqtt.disconnect();
        delay(3000);
        retries--;
        if (retries == 0)
        {
            while (1);
        }
    }
}
```

KAYNAK

adafruit ve kodlama kısmı için:

https://arduino-esp8266.readthedocs.io/en/latest/exception_causes.html

<https://learn.adafruit.com/mqtt-adafruit-io-and-you?view=all>

<https://learn.adafruit.com/mqtt-adafruit-io-and-you?view=all>

<https://www.etkilesimliogrenme.com/nodemcu-esp32-pin-giris-ve-cikislari>

