KOCAELİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

KAPALI ORTAMDA YÖNLENDİRME İÇİN WEB TABANLI ROBOT KONTROLÜ

MÜHENDİSLİK TASARIMI 2 FİNAL RAPORU

CELİL KOÇ

160207055

KOCAELİ 2020

İÇİNDEKİLER TABLOSU

ŞEKİLLER TABLOSU…..……..……...2

ÖZET……4

1.GENEL BİLGİLER…….…....5

1.1.DONANIM……..……...5

1.1.1.ESP32-S…….……5

1.1.2.L293D…….....5

1.1.3.LM2596……...……...5

1.1.4.ATMEGA328P…….……...…….5

1.1.5.MPU6050…….……...…...5

1.1.6.OPTİK ENKODER…….……...….5

1.1.7.REDÜKTÖRLÜ TEKERLEK……..……..…….....6

1.1.8.SARHOŞ TEKER……6

1.1.9.18650 PİL0...…….....6

1.2.YAZILIM……..……...…….....6

1.2.1.C……..…...……....6

1.2.2.PHP……...……....6

1.2.3.JAVASCRIPT……..7

1.2.4.MySQL……...…….……...…...7

2.MALZEME……7

2.1.DONANIM……...…7

2.1.1.ESP32-S…….……...…….....7

2.1.2.L293D……..……...7

2.1.3.LM2596……...……...7

2.1.4.ATMEGA328P…….……...……..8

2.1.5.MPU6050…….……...…...8

2.1.6.OPTİK ENKODER…….……...….8

2.1.7.REDÜKTÖRLÜ TEKERLEK……..……..…….....8

2.1.8.SARHOŞ TEKER……8

2.1.9.18650 PİL....……8

2.2.YAZILIM……..……...……9

2.2.1.C……..…...……....9

2.2.2.PHP……...……....9

2.2.3.JAVASCRIPT.....……..9

2.2.4.MySQL……...…….……...….9

3.KAPALI ORTAMDA YÖNLENDİRME İÇİN WEB TABANLI ROBOT GERÇEKLENMESİ……..10

3.1. ROBOTUN İNTERNETE BAĞLANMASI……..……..10

3.2. ROBOTUN KONUM KESTİRİMİ…….……...11

3.3. ROBOTUN DOĞRUSAL HAREKETİ……...……...12

3.4. ROBOTUN PİNG YÖNETİMİ……...13

3.5. WEB ARAYÜZ TASARIMI……...14

3.6. WEB HAREKET……..…….....15

3.6.1. KULLANICI TARAFI…….…….....15

3.6.2. SUNUCU TARAFI…...…….…...17

3.7. WEB KOORDİNAT GRAFİKLERİ……..…….....18

3.7.1. KULLANICI TARAFI…….…….....18

3.7.2. SUNUCU TARAFI…...…….…...18

3.8. ROBOT VE KULLANICI ETKİLEŞİMLERİ…...……...……...19

4.DENEYSEL SONUÇLAR……...……..20

5.SONUÇLAR VE ÖNERİLER….……...……...21

6.KAYNAKÇA……...…….……...22

ŞEKİLLER TABLOSU

Şekil 1.ESP-32S 5
Şekil 2.L293D 5
Şekil 3.LM2596 5
Şekil 4.ATMEGA328P 5
Şekil 5.MPU6050 5
Şekil 6.OPTİK ENKODER 5
Şekil 7.TEKERLEK 6
Şekil 8.SARHOŞ TEKER 6
Şekil 9.18650 6
Şekil 10.C LOGO 6
Şekil 11.PHP LOGO 6
Şekil 12.JS LOGO 7
Şekil 13.MYSQL LOGO 7
Şekil 14.ROBOT ESP-32S 7
Şekil 15.ROBOT L293D 7
Şekil 16.ROBOT LM2596 7
Şekil 17.ROBOT ATMEGA328P 8
Şekil 18.ROBOT MPU6050 8
Şekil 19.ROBOT ENKODER 8
Şekil 20.ROBOT TEKERLEK 8
Şekil 21.ROBOT SARHOŞ TEKER 8
Şekil 22.18650 8
Şekil 23.Manuel Kontrol Web Cevabı 10
Şekil 24.Otomatik Kontrol Web Cevabı 10
Şekil 25.Konum Hesaplayan Fonksiyon 11
Şekil 26.Doğrusal Hareket 12
Şekil 27.Doğrusal Hareket Kodu 12
Şekil 28.Alçak Ping 13
Şekil 29.Yüksek Ping 13
Şekil 30.Ping Yönetimi Kodu 13
Şekil 31.Giriş Ekranı 14
Şekil 32.Kontrol Paneli 14
Şekil 33.Manuel Kontrol Tuş Kontrol Kodu 15

Şekil 34. Otomatik Kontrol Koordinatları Gönderme Kodu 15
Şekil 35.Otomatik Kontrol Mouse Koordinat Tespit Kodu 16
Şekil 36.Otomatik Kontrol Sunucu Tarafı 17
Şekil 37.Manuel Kontrol Sunucu Tarafı 17
Şekil 38.Konum grafiği kodu 1 18
Şekil 39.Konum grafiği kodu 2 18
Şekil 40.Konum Grafiği Sunucu Tarafı Kodu 18
Şekil 41.Etkileşim 1 19
Şekil 42.Etkileşim 2 19
Şekil 43.Etkileşim 3 19
Şekil 44.Etkileşim 4 19
Şekil 45 Test Kontrol Paneli 20
Şekil 46.Test Konum 3 20
Şekil 47.Test Konum 2 20
Şekil 48.Test Konum 1 20
Şekil 49.Test Konum 5 20
Şekil 50.Test Konum 4 20

KAPALI ORTAMDA YÖNLENDİRME İÇİN WEB TABANLI ROBOT KONTROLÜ

ÖZET

Kapalı ortamda yönlendirme için web tabanlı robot kontrolü, gps benzeri konum algılama teknolojilerinin çalışmadığı ve hassasiyet konusunda yetersiz kaldığı kapalı ortamlarda robotun internet üzerinden kontrolünü ve konum takibini amaç edinen bir çalışmadır.

Çalışma , iki motorlu ve bir sarhoş tekerli robot üzerinde gerçeklenmişdir. Robotun internet üzerinden kontrol edilebilir ve izlenebilir olması için robota şu özellikler kazandırılmıştır ; Hareket kontrolü , Konum Kestirimi.

Robotun internet ile arayüz bağlantısı JSON kodu kullanılarak sağlanmıştır. Konum kestirimi için ise gyro sensörü ve optik enkoder kullanılmıştır.

Anahtar Kelimeler: Konum Kestirimi , Nesnelerin İnterneti , Robot Konumu , Robot Kontrolü

KISALTMALAR

SOC : System on Chip,

DMP : Digital Motion Processor ,

JSON: JavaScript Object Notation ,

IMU : Inertial Measurement Unit,

PWM : Pulse Width Modulation

1.GENEL BİLGİLER

1.1. DONANIM

1.1.1. ESP-32S

ESP-32S düşük maliyetli, düşük güç tüketen WIFI ve Bluetooth entegreli bir SOC'tur. Espressif Systems adlı firma tarafından üretilir.

Şekil 1.ESP-32S

1.1.2. L293D

L293D dörtlü yarım H köprüsüdür. Lojik sinyale göre çıkış pinlerinin polaritesini değiştirir. Robotik uygulamalarda sıkça kullanılır.

Şekil 2.L293D

1.1.3. LM2596

LM2596 bir DC-DC step-down çeviricidir. Verimi %90'ın üstüne çıkabilmektedir.

Şekil 3.LM2596

1.1.4. ATMEGA328P

ATMEGA328P Atmel firması tarafından üretilen , 8 Bit Harvard Mimarisi kullanan bir mikrodenetleyicidir.

Şekil 4.ATMEGA328P

1.1.5. MPU6050

MPU6050 tek bir chip üzerinde 3 eksen ivme ölçer , 3 eksen gyro ve bir DMP bulunduran hareket tespit sensörüdür.

Şekil 5.MPU6050

1.1.6 OPTİK ENKODER

Optik Enkoder , dönüş hızı tespiti için ışık,ışık sensörü ve delikli disk kullanan elektromekanik bir sensördür.

Şekil 6.OPTİK ENKODER

1.1.7 REDÜKTÖRLÜ TEKERLEK

Redüktörlü tekerlek içinde 250rpm redüktöre bağlı bir dc motor olan bloktur.

Şekil 7.TEKERLEK

1.1.8. SARHOŞ TEKER

Her yöne dönebilen bir tekerlek türüdür.

Şekil 8.SARHOŞ TEKER

1.1.9. 18650 PİL

18650 pili bir lithium-ion pildir. Şarj olabilme özelliğine sahiptir. Azami gerilimi 4.2V, ortalama gerilimi ise 3.7V civarındadır.Yüksek kapasite ve yüksek akım verebilme özelliklerine sahiptir.

Şekil 9.18650

1.2. YAZILIM

1.2.1 C

AT&T Bell laboratuvarlarında, Ken Thompson ve Dennis Ritchie tarafından UNIX İşletim Sistemi' ni geliştirebilmek amacıyla B dilinden türetilmiş yapısal bir programlama dilidir.

Şekil 10.C LOGO

1.2.2 PHP

PHP, internet için üretilmiş, sunucu taraflı, çok geniş kullanımlı, genel amaçlı, içerisine HTML gömülebilen betik ve programlama dilidir.

Şekil 11.PHP LOGO

1.2.3 JAVASCRİPT

JavaScript, yaygın olarak web tarayıcılarında kullanılmakta olan,kullanıcı taraflı dinamik bir programlama dilidir.

Şekil 12.JS LOGO

1.2.4 MySQL

MySQL, altı milyondan fazla sistemde yüklü bulunan çoklu iş parçacıklı,çok kullanıcılı hızlı ve sağlam bir veri tabanı yönetim sistemidir.

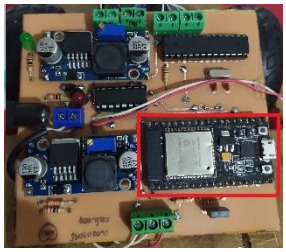
Şekil 13.MYSQL LOGO

2. MALZEME

2.1 DONANIM

2.1.1. ESP-32S

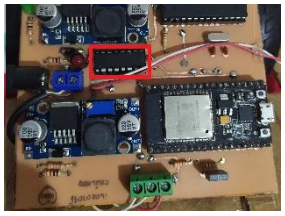
ESP-32S kablosuz internet üzerinden veri aktarımı sağlamak , hareketi kontrol etmek ve konum takibi yapmak gibi görevleri yerine getirirmek için kullanılmıştır.



Şekil 14.ROBOT ESP-32S

2.1.2. L293D

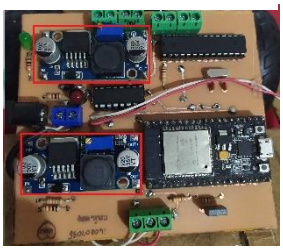
DC motorları sürmek için kullanılmıştır.



Şekil 15.ROBOT L293D

2.1.3. LM2596

8.4V olan batarya voltajını ATMEGA328P ve ESP-32S mikrodenetleyicileri için uygun olan gerilimi indirmek ve bataryayı şarj etmek için kullanılmıştır.



Şekil 16.ROBOT LM2596

2.1.4. ATMEGA328P

MPU6050 sensöründen sağlıklı bir şekilde açı verisi okuyabilmek için en az 500hz frekansında veri örneklenmelidir. ESP32 denetleyecisinin ana programındaki internete bağlanma komutları ve diğer komutlar 10ms den fazla sürdüğü için , mpu6050 ikinci bir denetleyiciye(Atmega328p) bağlanmış ve ESP32'ye bu veriler UART haberleşmesi ile aktarılmıştır.

Şekil 17.ROBOT ATMEGA328P

2.1.5. MPU6050

MPU6050 'nin sadece gyro sensörü kullanılarak robotun kendi etrafındaki dönüş açısını hesaplanır.

Şekil 18.ROBOT MPU6050

2.1.6 OPTİK ENKODER

Optik Enkoder , robotun ileri ve geri hareketinde katettiği mesafeyi hesaplar.

Şekil 19.ROBOT ENKODER

2.1.7 REDÜKTÖRLÜ TEKERLEK

Robotun hareketini sağlar.

Şekil 20.ROBOT TEKERLEK

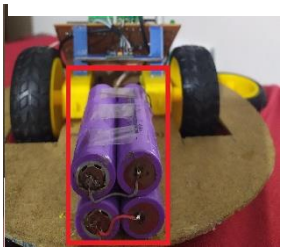
2.1.8. SARHOŞ TEKER

Robotun hareketini sağlar.

Şekil 21.ROBOT SARHOŞ TEKER

2.1.9 18650 PİL

Robotun güç kaynağıdır. 2 seri ve 2 paralel olmak üzere 4 tane 18650 pil kullanılmıştır. Böylece 8.4V ve ~4000mAh kapasitesine sahip bir batarya elde edilmiştir.



Şekil 22.18650

2.2. YAZILIM

2.2.1 C

C dili ESP-32S ve ATMEGA328P mikrodenetleyicilerini programlamak için kullanılmıştır. Robotun internet bağlantısı , konum kestirimi ve hareketi C dili tarafından gerçeklenir.

2.2.2 PHP

Robotun bağlandığı websitesi üzerinde bulunan giriş ekranını ve veritabanına veri yazma ve okuma işlemlerini gerçeklemek için sunucu tarafında kullanılmıştır.

2.2.3 JAVASCRİPT

Websitesinin kullanıcı paneli kısmında koordinat grafiklerini,tuşları,batarya göstergesini ve sistem uyarılarını gerçeklemek için kullanılmıştır.

2.2.4 MySQL

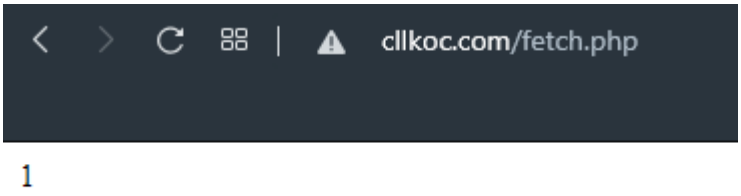
MySQL, panele giriş için gerekli olan şifreyi ve kullanıcı adını ,robotun poziyon bilgilerini ,robotun hareket durumunu, şarj durumunu ve otomatik mod için gidilmesi gereken koordinatları veritabanına kaydetmek için kullanılmıştır.

3. KAPALI ORTAMDA YÖNLENDİRME İÇİN WEB TABANLI ROBOT KONTROLÜ GERÇEKLENMESİ

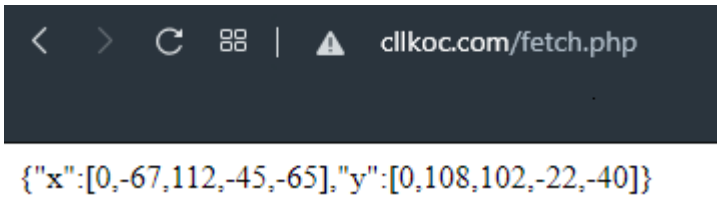
3.1. ROBOTUN İNTERNETE BAĞLANMASI

ESP-32S önceden tanımlanan bir wifi ağına bağlanır, ESP-32S HTTP İstemcisi olarak davranabilir ve websitesinden verileri polling yöntemi ile almaktadır. Robot kendi pozisyon bilgisini,batarya seviyesini GET methodu kullanarak websayfasına gönderir , cevap olarakta sayfadan hareket kodu (dur =1, ileri=2, geri=3,vb…) veya otomatik mod için koordinatları JSON formatında alır , işler ve harekete geçer.

```
readPosition();
readMPU();
sprintf(sent,"http://cllkoc.com/fetch.php?x=%d&y=%d&p=%d&a=%d&v=%d&f=%d",x,y,gec,pos,volt,busyFlag);
portEXIT_CRITICAL(&externalMux);
http.setReuse(true);
http.begin(sent); //HTTP
```



Şekil 23.Manuel Kontrol Web Cevabı



Şekil 24.Otomatik Kontrol Web Cevabı

3.2 ROBOTUN KONUM KESTİRİMİ

Robotun kendi etrafındaki dönüş açısını hesaplamak için MPU6050 gyro sensörü ve ileri-geri doğrusal hareketini hesaplamak için optik enkoder kullanılmıştır.

Tekerin çevresi ~20.5 cm olarak ölçülmüştür. Enkoder 9° lik hassasiyetle ölçüm yapabilmektedir. Her 9° lik değişim robotun ortamda 0.51cm ileri veya geri gitmesini temsil eder.

2 boyutlu düzlemde konum tespiti için robotun y eksenine göre acısının sinüsü ,x ekseninde ki ilerleme ve bu açının kosinüsü, y ekseninde ki ilerlemedir. Bu bilgiler ışığında aşağıdaki ilişki ortaya çıkar ;

yeniXkonumu = eskiXkonumu + (enkoderDerece*0.51)*sin(mpuDerece)

yeniYkonumu = eskiYkonumu + (enkoderDerece*0.51)*cos(mpuDerece)

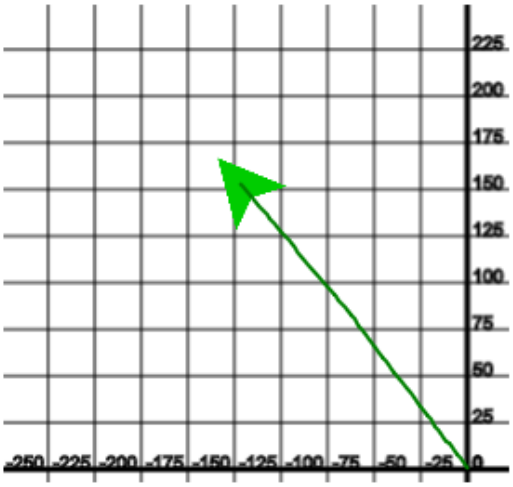
```
void readPosition(){

        ang = tik2*0.51;
        tik2=0;
        hypoNew = hypoOld + ang;
        Xnew = Xold + ang * sin(pos*0.0175); x = (int)Xnew;
        Ynew = Yold + ang * cos(pos*0.0175); y = (int)Ynew; distance = (int)hypoNew;
        Xold=Xnew;Yold=Ynew; hypoOld = hypoNew;

  }
```

Şekil 25.Konum Hesaplayan Fonksiyon

3.3. ROBOTUN DOĞRUSAL HAREKETİ

Robotun üzerindeki komponentlerin ağırlık merkezine göre eşit dağılmayışı, zemin bozukluğu ve motorlar arasındaki hız farkı robotun doğrusal hareket yapmasına engel olmaktadır. Bu durumu engellemek için robot ileri-geri hareket halindeyken sağ-sol dönüşü engellenmektedir. Sağa veya sola dönebilmek için robotun durur pozisyonda olması gerekir.

Robot ileri-geri hareket etmeye başladığında en son sahip olduğu açı değeri korunacak şekilde motorların hızları ESP32 tarafından pwm sinyal kullanılarak alçaltılır veya artırılır.



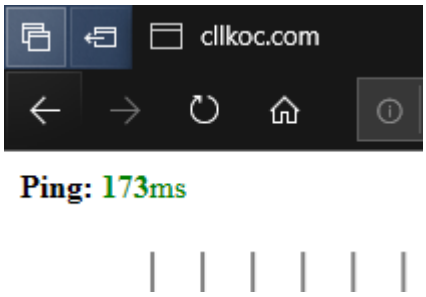
Şekil 26.Doğrusal Hareket

```
if(op==1)
{
 if(tik2%20==0){
 if(pos < 200 && pos > 160 )
 {

 }else{
 if(donaci > olddonaci)
 {
  pulse2--;
  pulse++;
 }else if(donaci < olddonaci)
 {
  pulse--;
  pulse2++;
 }else{

   }
 }
```

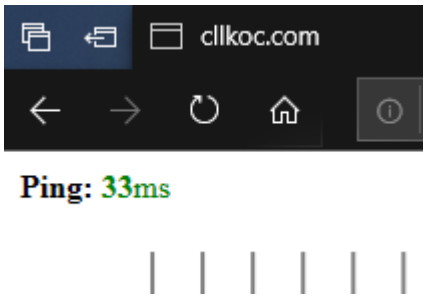
Şekil 27.Doğrusal Hareket Kodu

3.4. ROBOTUN PİNG YÖNETİMİ

ESP32, Bölüm 2.5'te belirtildiği üzere sunucuya bağlanabilir fakat sunucudan cevap alma süresi , sunucu meşguliyeti, robotun bağlı olduğu ağın hızı , vb. nedenlerden dolayı her zaman mükemmel olmayabilir.Yapılan denemelerde gecikme süresinin(Ping) 5saniyeye kadar çıktığı gözlemlenmiştir. Robotun istenmeyen bir şekilde çarpmasını veya hareket etmesini engellemek için ESP32 sunucuya bağlanma isteği attıktan sonra 1 sn lik bir zamanlayıcı başlatır. Sunucudan cevap süresi 1sn üzerinde olur ise Robotu olduğu yerde durdurur.



Şekil 29.Yüksek Ping



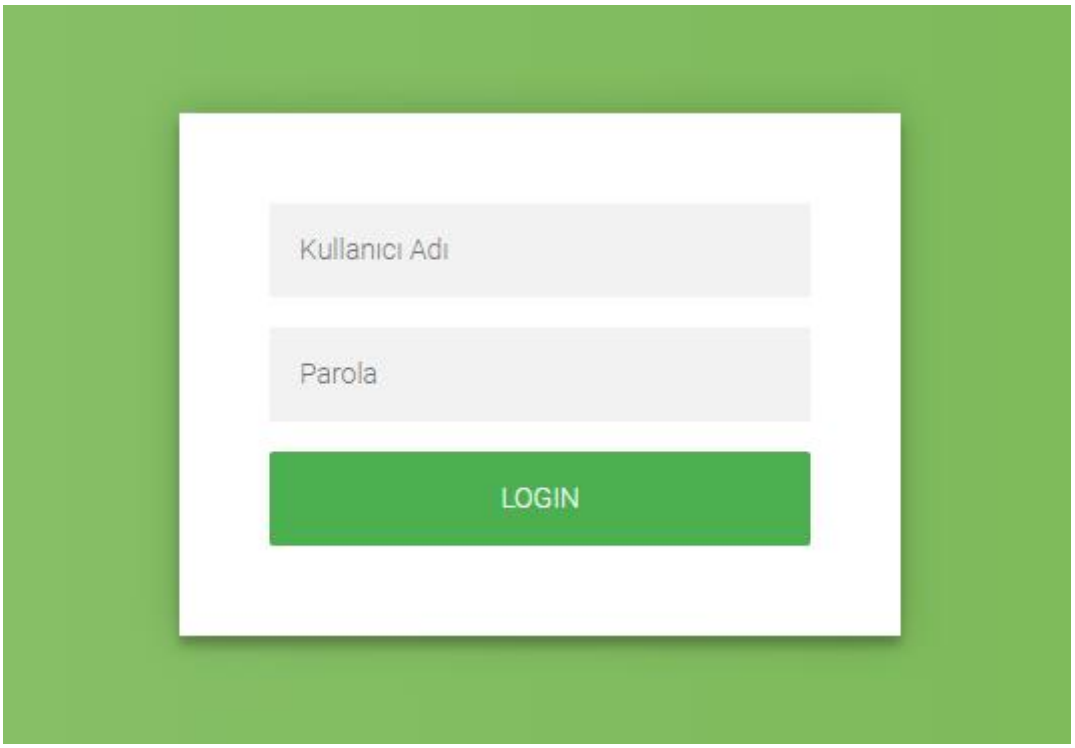
Şekil 28.Alçak Ping

```
 http.begin(sent); //HTTP
 timerAlarmEnable(timer);
 zaman = millis();
// USE_SERIAL.print("[HTTP] GET...\n");
 // start connection and send HTTP header

httpCode = http.GET();
if(httpCode > 0) {
 //file found at server.
 if(httpCode == HTTP_CODE_OK) {
        timerAlarmDisable(timer);
```

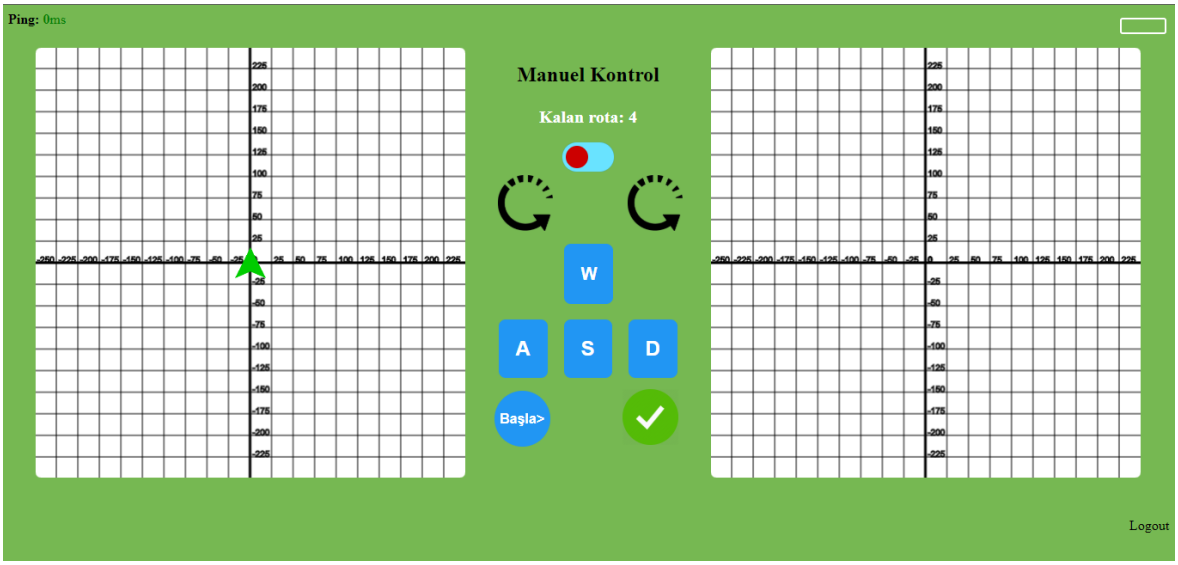
Şekil 30.Ping Yönetimi Kodu

3.5. WEB ARAYÜZ TASARIMI

Arayüz tasarımı HTML ve CSS kullanılarak yapıldı.Kontrol paneline erişim için şifre konuldu. Koordinat grafikleri için Canvas kullanıldı. Hareket kontrolü içinse klayedeki w,a,s,d tuşları ve otomatik kontrol içinse mouse ile çizim kullanıldı.



Şekil 31.Giriş Ekranı



Şekil 32.Kontrol Paneli

3.6. WEB HAREKET

3.6.1. KULLANICI TARAFI

Manuel modda ; W,A,S,D tuşlarından biri basılı olduğu sürece robot hareket eder ve tuş bırakılırsa robot durur. Bu özelliği kazanmak için keyUp ve keyDown fonksiyonlarından yararlanılmıştır. Hangi tuş basıldı tespiti için tuşların ALT CODE'u doğrulanır. Tuşların basılıp basılmaması durumana göre "sqlhandler.php" dosyası GET methodu ile çağrılır. GET methodundaki anahtar(code) hangi tuşun basıldığı bilgisini taşır.

Otomatik modda ise gidilmesi gereken koordinatları sqlhandler.php adlı dosyaya post methodu ile aktarır.Burda post methodu kullanmak zorunludur çünkü , x ve y ekseni koordinatları birden fazla olabilir ve başka bir php dosyasına dizi atmak için post methodu AJAX ile kullanılmalıdır.

```
if(event.keyCode == 68) {
    if(!ispressed)
    {
        xmlhttp.open("GET","sqlhandler.php?code=4",true);
        xmlhttp.send();
        buttonCode = 68;
        ispressed = 1;
       document.getElementById("right").classList.remove("button1");
        document.getElementById("right").classList.add("button1basti");
    }

  }
```

Şekil 33.Manuel Kontrol Tuş Kontrol Kodu

```
$.ajax({
        type: "POST",
        url: "sqlhandler.php",
        data: { xArray:JSON.stringify(positionsX) , yArray:JSON.stringify(positionsY)},
        success: function() {
              //alert("Koordinatlar başarılı bir şekilde aktarıldı.");
         }
});
```

Şekil 34. Otomatik Kontrol Koordinatları Gönderme Kodu

```
canvas2.addEventListener("click", function (evt) {
        if(remot==-1 && rota_sayi > 0){
         var mousePos = getMousePos(canvas2, evt);
         ctx2.beginPath();
         ctx2.strokeStyle = "blue";
         ctx2.lineWidth = 4;
         ctx2.lineJoin = "round";
         ctx2.moveTo(lastX, lastY);
         ctx2.lineTo(mousePos.x, mousePos.y);
         positionsX[a] = mousePos.x - 250;
         positionsY[a] = 250-mousePos.y;
         rota_sayi = rota_sayi - 1;
         rota_sinir.innerHTML = rota_sayi;
         a = a + 1;
         ctx2.closePath();
         ctx2.stroke();
         lastY = mousePos.y; lastX = mousePos.x;
}else if(remot!=-1){

    alert("Çizerek yönledirme için otomatik kontrol moduna geçiniz!");
}else if(!(rota_sayi > 0)){
    alert("Rota sayısı aşıldı.");
}else{}

}, false);

//Get Mouse Position
function getMousePos(canvas, evt) {
    var rect = canvas2.getBoundingClientRect();
    return {
        x: evt.clientX - rect.left,
        y: evt.clientY - rect.top
    };
}
```

Şekil 35.Otomatik Kontrol Mouse Koordinat Tespit Kodu

3.6.2. SUNUCU TARAFI

"Sqlhandler.php" adlı scriptin yaptığı iştir.Manuel modda tuşların durumuna göre veritabanının ilgili kısmına hareket kodunu yazar.Otomatik modda ise Koordinat grafiğinde gidilmesi istenen koordinatları veritabanına yazar.

```
if(isset($_GET["code"]))
{
    $int = (int)$_GET["code"];
}
if($int > 10)
 $int = 10;
// sql to create table
$sql = "UPDATE control SET CODE=$int WHERE id=1";
```

Şekil 37.Manuel Kontrol Sunucu Tarafı

```
if(isset($_POST))
{
   $sql = "DROP TABLE pozisyon";
   $conn->query($sql);
   $sql = "CREATE TABLE pozisyon(id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, x INT(16),y INT(16))";
   $conn->query($sql);
   $xArray = json_decode($_POST['xArray']);
   $yArray = json_decode($_POST['yArray']);
   $sql = "INSERT INTO pozisyon (x,y) VALUES";
   $multiple;
   for($i=0; $i < count($xArray) ; $i++)
   {
        $sayiX = (int)$xArray[$i];
        $sayiY = (int)$yArray[$i];
        $multiple=$multiple."($sayiX,$sayiY)";
        if($i!= count($xArray)-1)
        $multiple=$multiple.",";
  }
  $sql = $sql.$multiple;
}
```

Şekil 36.Otomatik Kontrol Sunucu Tarafı

3.7. WEB KOORDİNAT GRAFİKLERİ

3.7.1. KULLANICI TARAFI

Grafikleri güncellemek için 100ms periyotlarla çağrılan fonksiyon startLiveUpdate() , "viewcount.php" adlı scriptten robotun x eksenindeki konumu , y eksenindeki konumu , dönüş açısı ,batarya seviyesini ve ping verilerini JSON formatında alarak ayrıştırır ve grafikleri günceller.

```
function startLiveUpdate(){

 setInterval(function(){
 fetch('viewcount.php').then(function(response){
     return response.json();

 }).then(function(data){
  x = data.viewX;
  y = data.viewY;
  volt = data.volt;
  degree = parseInt(data.Angle);
  busy = parseInt(data.f);
 if(busy==1)
    document.getElementById('busycheck').src='images/busy.gif';
  else
    document.getElementById('busycheck').src='images/free.png';
  if(volt > 100)
   {
     volt = volt - 100;
     isCharging(1);
   }else{isCharging(0);}
```

Şekil 38.Konum grafiği kodu 1

```
  if(volt>=75)
      btx.fillStyle = "#c8f86b";
  if(volt >=50 && volt < 75)
      btx.fillStyle = "#6bf887";
  if(volt < 50 && volt >= 25)
      btx.fillStyle = "#f1c631";
  if(volt <25)
      btx.fillStyle = "#e43c39";
  if(volt < 10)
      if(kont)
     { alert("Batarya kritik seviyede! ,robotu durdurup şarj etmeyi düşünebilirsiniz."); kont=0;}
  btx.clearRect(1,1,48,13);
  btx.fillRect(1, 1, volt/100*batarya.width, 13);
  ctx.clearRect(0,0,canvas.width,canvas.height);
  ctx.save();
  ctx.drawImage(car,0,0,500,500);
  ctx.translate(x+250,250-y); // change origi
  ctx.rotate(degree*Math.PI/180);
  ctx.drawImage(base_image,-18,-18,36,36);
  ctx.restore();
  ctx.lineTo(x+250,250-y);
  ctx.lineWidth = 3;
  ctx.strokeStyle = "#008000";
  ctx.stroke();

 }).catch(function (error){
   console.log(error);
 });



},100);
}
document.addEventListener('DOMContentLoaded',function(){
startLiveUpdate();
});
```

Şekil 39.Konum grafiği kodu 2

3.7.2. SUNUCU TARAFI

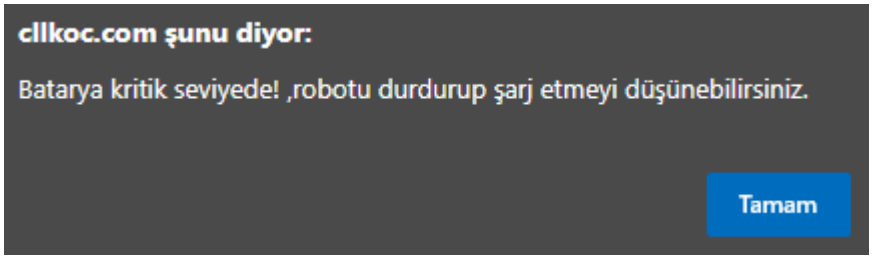
"viewcount.php" adlı scriptin yaptığı iştir.Veritabanındaki verilerini okur ve onları JSON formatında paketleyerek Kontrol panelindeki canvas grafiğe aktarır.

```
$data = [
    'viewX' => (int) $keywords[0],
    'viewY'=> (int)$keywords[1],
    'Ping'=> (int)$keywords[2],
    'Angle'=>(int)$keywords[3],
    'volt' => (int)$keywords[4],
    'f' => (int)$keywords[5]
    ];
$conn->close();
echo json_encode($data);
```

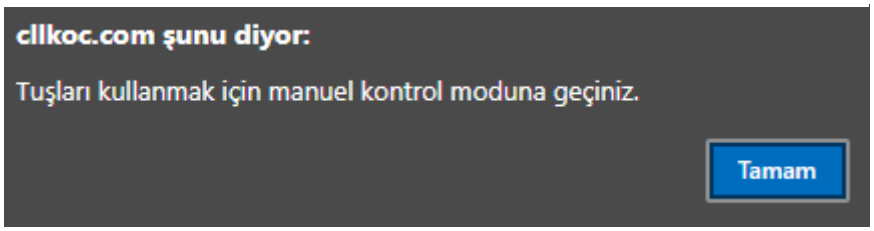
Şekil 40.Konum Grafiği Sunucu Tarafı Kodu

3.8. ROBOT VE KULLANICI ETKİLEŞİMLERİ

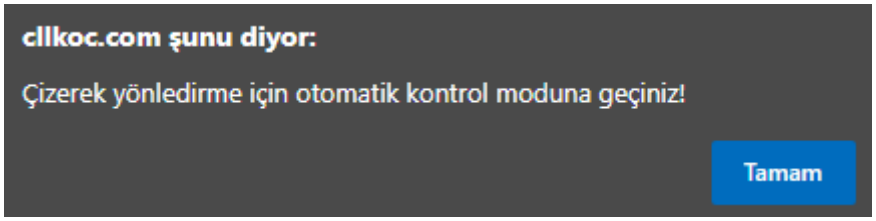
Robotu uzaktan kontrol ederken en iyi deneyimi sağlamak için kullanıcıyı uyaran bazı etkileşimler kullanılmıştır.



Şekil 41.Etkileşim 1



Şekil 42.Etkileşim 2

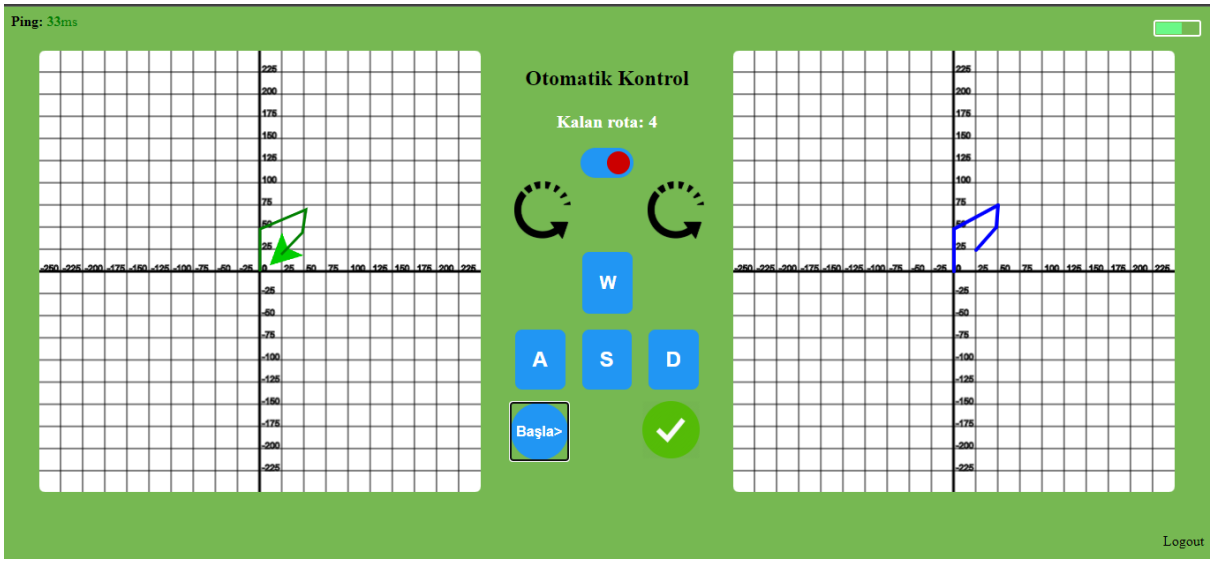


Şekil 43.Etkileşim 3

Şekil 44.Etkileşim 4

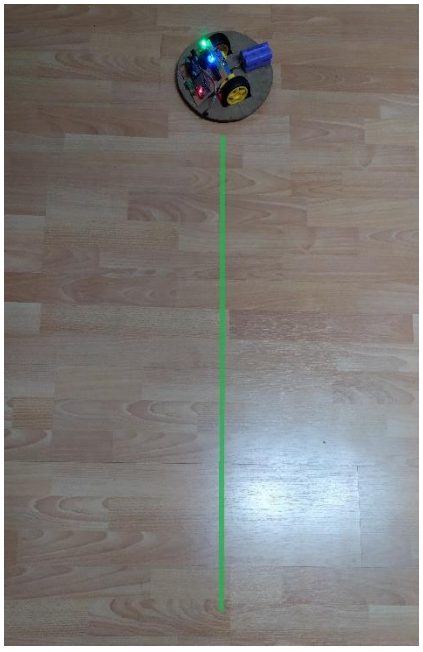
4.DENEYSEL SONUÇLAR

Yapılan deneyeler sonucunda robot başarılı bir şekilde hareket ettirilmiş ve konumu kestirilmiştir. Aşağıdaki şekiller otomatik modda yönlendirme testi görüntüleridir.



Şekil 45 Test Kontrol Paneli

Şekil 48.Test Konum 1



Şekil 47.Test Konum 2

Şekil 46.Test Konum 3

Şekil 50.Test Konum 4

Şekil 49.Test Konum 5

5.SONUÇ VE ÖNERİLER

Robotun internet üzerinden yönledirmesi ve kapalı ortamda konum takibi başarı ile gerçeklenmiştir.Yapılan deneylerde , robotun gittiği zemine bağlı olarak konum kestiriminde hatalar ~10%'a kadar çıkabilmektedir.

Hata oranı ve deney sonuçları düşünülerek, konum kestirimi için enkoder kullanmanın çok verimli bir yöntem olmadığı saptanmıştır. En iyi hassasiyeti almak için sistem tamamen IMU sensörlerden oluşabilir.

Robotun bağlandığı sayfa ise herkes tarafından görüntülenebilir ve hareket komutları manipüle edilebilir bu yüzden güvenli bağlantı için bir şifreleme sistemi kullanılabilir.

6.KAYNAKÇA

https://www.instructables.com/id/Navigate-Robot-With-Shoe-Sensors-Wo-GPS-Wo-Map/

https://github.com/xioTechnologies/Oscillatory-Motion-Tracking-With-x-IMU

http://mikrobotik.com/wp2/2018/08/10/esp8266-01-ile-dunyadan-evinize-web-sitenizle-baglanti-kurun/

https://stackoverflow.com/questions/7496674/how-to-rotate-one-image-in-a-canvas?rq=1

https://www.w3schools.com/php/php_mysql_intro.asp

https://elektronikhobi.net/arduino-optik-enkoder/

https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/timer.html

https://randomnerdtutorials.com/esp32-pwm-arduino-ide/

https://stackoverflow.com/questions/9001526/send-array-with-ajax-to-php-script