

# Engel Kaldırıcı Sumo Robot

İrem Kalkanlı, Özlem Çalı, Deniz Uzun, Ceyda Uymaz, İrem Bozkurt

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: { irem.kalkanli , ozlem.cali, deniz.uzun, ceyda.uymaz,irem.bozkurt }@stu.fbu.edu.tr,

**Özetçe**— Sumo robot gerçekleştirilip, verilecek 5 adet hedefin 60 saniye içerisinde platformdan atabilecek bir sumo robot algoritması geliştirilecektir. Robot'un diğer bir modu ise çizgi izlemektir. Bu modda ise kensine bağlı olan sensörleri kullanarak çizgi üzerinde takip etmesi beklenmektedir.

## Anahtar Kelimeler — Sumo, Arduino

**Abstract**— Sumo robot is a sumo robot that can be implemented and thrown from the platform within 60 seconds of 5 targets to be given. algorithm will be developed. Another mode of Robot is to follow lines. In this mode, the sensors connected to it. It is expected to follow the line using.

## Keywords — Sumo, Arduino

### I. Giriş

Sumo robot gerçekleştirilip, verilecek 5 adet hedefin 60 saniye içerisinde platformdan atabilecek bir sumo robot algoritması geliştirilecektir. Robot'un diğer bir modu ise çizgi izlemektir. Bu modda ise kensine bağlı olan sensörleri kullanarak çizgi üzerinde takip etmesi beklenmektedir.

### II. SİSTEM MİMARİSİ

Proje kapsamında 1 araç kullanılacaktır.

#### 1) Arduino IDE

Arduino için Entegre Geliştirme Ortamı, C ve C++ dilleri ile yazılmış bir platformlar arası uygulamadır. Arduino uyumlu kartlara program yazmak ve yüklemek için kullanılır, aynı zamanda 3. taraf çekirdekler ve satıcıların geliştirme kartları içinde kullanılabilir

### III. KULLANILAN YAZILIM

#### 1) Engel Kaldırıcı Sumo Robot

```
//MOTOR CONTROL
int RPwm = 11;//sağ motorun hızı
int RDir = 13;//sağ motorun yönü
int LPwm = 3;//sol motorun hızı
int LDir = 12;//sol motorun yönü

//OPPONENT SENSORS
int LSens = A4;
int RSens = A2;
int MSens = A3;

int LFSens = A5;
int RFSens = 4;

//EDGE & CONTRAST SENSORS
int Redge = A1;//saha dışına çıkmaması için gereken sensörler
int Ledge = A0;
```

```
//TRIMPOTS//potasyometre
int SPD = A7;//direnci değiştirerek motorun hızını ayarlar.
int TRN = A6;//direnç açıya göre değiştirilir.

int Button = 10;
int ArduLed = 8;
int Buzzer = 9;

int Speed =50;
int MaxSpeed = 50; // Idle Speed while no sensor giving data.
int TurnSpeed = 55; // Left and Right Forward Turning Speed
int EdgeTurn = 15; // Turning Time variable when minisumo sees white line ->190
int Duration; // Turning Time at minisumo starting.
int LastValue = 5; // Last Value Variable for remembering last Opponent sensor sense.
```

Sensörler, ledler, buzzer ve motor bilgileri ilgili pinlere tanımlanmıştır. Bu tanımlamalar sonucu robot ile kod arasında bağlantılar yapılmıştır.

```
void setup() {
pinMode(LSens, INPUT); // Left Opponent Sensor Input
pinMode(RSens, INPUT); // Right Opponent Sensor Input
pinMode(MSens, INPUT); // Middle Opponent Sensor Input
pinMode(Buzzer, OUTPUT); // Buzzer Declared as Output
pinMode(ArduLed, OUTPUT); // ArduLed Declared as Output
pinMode(Button, INPUT); // Button Input

pinMode(RPwm, OUTPUT); // Four PWM Channel Declared as Output
pinMode(RDir, OUTPUT);
pinMode(LPwm, OUTPUT);
pinMode(LDir, OUTPUT);

digitalWrite(RFSens, HIGH);
digitalWrite(MSens, HIGH);

digitalWrite(Buzzer, LOW);
digitalWrite(ArduLed, LOW); // Arduino Mode Led Made Low

Serial.begin(9600);
tone(9, 523, 300);
delay(300);
noTone(9);
}
```

Sensörler, ledler, buzzer ve motor değişkenlerinin input output bilgileri ayarlanmıştır. Digital Write ile sensörlerin çalışma şekilleri low ve high olarak belirlenmiştir. Ardından robot başladığında ses çıkartması sağlanmıştır.

```
void Set_Motor (float Lval, float Rval, int timex){
Lval = Lval*2.5;
Rval = Rval*2.5;
```

Motorun hız değerlerinin ataması yapılır.

```
if (Lval >=0) {
```

```

    analogWrite(LPwm, Lval);
    digitalWrite(LDir, LOW);
  } else {
    Lval=abs(Lval);
    digitalWrite(LDir, HIGH);
    analogWrite(LPwm, Lval);
  }
  if (Rval >=0) {
    analogWrite(RPwm, Rval);
    digitalWrite(RDir, HIGH);
  } else {
    Rval=abs(Rval);
    digitalWrite(RDir, LOW);
    analogWrite(RPwm, Rval);
  }

  delay(timex);
}

```

Motorun hız değerine göre yön belirlenir. Motor hızının pozitif veya negatif olma durumuna göre her iki motora da yön ataması yapılır.

Sol motor hızı pozitif ise sağa yönelmesi için LDir ataması LOW yapılır. Eğer negatif ise sola yönelmesi için LDir ataması HIGH yapılır.

Sağ motor hızı negatif ise sola yönelmesi için RDir ataması LOW yapılır. Eğer pozitif ise sağa yönelmesi için RDir ataması HIGH yapılır.

```

void loop() {
  digitalWrite(RPwm, LOW);
  digitalWrite(LPwm, LOW);
  tone(Buzzer, 18, 100);
}

```

Sağ ve sol motorun güçleri low olarak ve motorun sesi ayarlandı.

```

Start:
  /// Edge Sensor Control Routine ///
  digitalWrite(ArduLed, LOW);
  if (analogRead(Ledge)>100 && analogRead(Redge)< 100) {
    digitalWrite(Buzzer, LOW);
    digitalWrite(ArduLed, HIGH);
    Set_Motor(-50, -50, 350); // Geri
    Set_Motor(50, -50, EdgeTurn); // Left Backward, Right Forward, Turning Time Based
on ETRN Trimpot
    LastValue=5;
  }
  else if (analogRead(Ledge)< 100 && analogRead(Redge)> 100) {
    digitalWrite(Buzzer, LOW);
    digitalWrite(ArduLed, HIGH);
    Set_Motor(-50, -50, 350); // Back 35 Milliseconds
    Set_Motor(-50, 50, EdgeTurn); // Right Backward, Left Forward, Turning Time Based
on ETRN Trimpot
    LastValue=5;
  }
  else if (analogRead(Ledge)>100 && analogRead(Redge)> 100) {
    digitalWrite(Buzzer, LOW);
    digitalWrite(ArduLed, HIGH);
  }
}

```

```
Set_Motor(-50, -50,15); // Back 35 Milliseconds
Set_Motor(50, -50, EdgeTurn); // Right Backward, Left Forward, Turning Time Based
on ETRN Trimpot
```

Robotun ayarlanmış alandan çıkmaması için control sensörleri ile kontrol yapılmıştır ve çıkmaması sağlanmıştır.

```
LastValue=5;

}else
/// Opponent Sensor Control Routine ///
//while (digitalRead(Button)==LOW) {Set_Motor(0, 0, 20); digitalWrite(Buzzer, LOW);
LastValue=3;} digitalWrite(Buzzer, LOW);
if (digitalRead(MSens)==LOW) {Set_Motor(50, 50,1); LastValue=5;} else
if (digitalRead(LSens)== LOW) {Set_Motor(-50, 50,1); LastValue=7;} else
if (digitalRead(RSens)==LOW) {Set_Motor(50, -50,1); LastValue=3;} else
{

//Speed=(analogRead(SPD)/10.3); Speed=100-Speed;
if (LastValue==5) { Set_Motor(20, 20,1);} else // Forward, Based on SPD (A7) Trimpot
if (LastValue==7) { Set_Motor(-20, 40,2);} else // Left Turning Based on SPD (A7)
Trimpot
if (LastValue==3) { Set_Motor(40,-20,2);} // Right Turning Based on SPD (A7) Trimpot
}
goto Start;
}
```

Bu kısımda robotun bir cisim gördüğünde nasıl tepkiler vereceği ayarlanmıştır.

## 2)Çizgi takip

```
//MOTOR CONTROL
int RPwm = 11;//sağ motorun hızı
int RDir = 13;//sağ motorun yönü
int LPwm = 3;//sol motorun hızı
int LDir = 12;//sol motorun yönü

//EDGE & CONTRAST SENSORS
int Redge = A1;
int Ledge = A0;
```

Saha dışına çıkmaması için gereken sensörler.

```
void Set_Motor (float Lval, float Rval){
    Lval = Lval*2.5;
    Rval = Rval*2.5;
```

Motorun hız değerlerinin ataması yapıldı.

```
if (Lval >=0) {
    analogWrite(LPwm, Lval);
    digitalWrite(LDir, LOW);
} else {
    Lval=abs(Lval);
    digitalWrite(LDir, HIGH);
    analogWrite(LPwm, Lval);
}
if (Rval >=0) {
```

```

    analogWrite(RPwm, Rval);
    digitalWrite(RDir, HIGH);
  } else {
    Rval=abs(Rval);
    digitalWrite(RDir, LOW);
    analogWrite(RPwm, Rval);
  }
}

```

Motorun hız değerine göre yön belirlenir. Motor hızının pozitif veya negatif olma durumuna göre her iki motora da yön ataması yapılır.

Sol motor hızı pozitif ise sağa yönelmesi için LDir ataması LOW yapılır. Eğer negatif ise sola yönelmesi için LDir ataması HIGH yapılır.

Sağ motor hızı negatif ise sola yönelmesi için RDir ataması LOW yapılır. Eğer pozitif ise sağa yönelmesi için RDir ataması HIGH yapılır.

```

void setup() {

pinMode(RPwm, OUTPUT);
pinMode(RDir, OUTPUT);
pinMode(LPwm, OUTPUT);
pinMode(LDir, OUTPUT);

}

```

Setup fonksiyonunda pinMode komutuyla değişkenlere output bilgileri yollanmıştır.

```

void loop() {
  Start:
  if (analogRead(Ledge)>100 && analogRead(Redge)>100) { //Sol ve sağ sensör siyah zeminde
    olduğu için motorun düz gitmesi istenir.
    forward();
  } if (analogRead(Ledge)<100 && analogRead(Redge)>100) { //Sol sensör beyaz zeminde, sağ
    sensör siyah zeminde olduğu için motorun sağa gitmesi istenir.
    right();
  } if (analogRead(Ledge)>100 && analogRead(Redge)<100) { //Sol sensör siyah zeminde, sağ
    sensör beyaz zeminde olduğu için motorun sola gitmesi istenir.
    left();
  }

  if (analogRead(Ledge)<100 && analogRead(Redge)<100) { //Sol ve sağ sensör beyaz zeminde
    olduğu için motorun geri gitmesi istenir.
    back();
  }
  goto Start;
}

```

Zeminin renk kontrolleri olasılıklara göre okunur.

Eğer edge değerleri 100'den büyük ise siyah zeminde bulunduğu, 100'den küçükse beyaz zeminde bulunduğu anlamına gelir.

Değerlere uygun şekilde robotun hareketi belirlenir.

```

void forward() {
  Set_Motor(30,30);
}

```

```
void left() {  
    Set_Motor(0,30);  
}  
void right() {  
    Set_Motor(30,0);  
}  
void back() {  
    Set_Motor(-30,-30);  
}
```

Motorların uygun yöne ilerleyebilmesi için gerekli fonksiyonlar yazılarak, sağ ve sol yön atamaları yapılmıştır.

#### IV. SONUÇLAR

Sumo robot gerçekleştirilip, verilecek hedeflerin 60 saniye içerisinde platformdan atabilecek bir sumo robot algoritması geliştirilmiş ve gözlenmiştir.

Robot'un diğer bir modunda ise çizgi takibi yapılmıştır. Bu modda ise kensine bağlı olan sensörleri kullanarak çizgi üzerinde takip ettiği gözlenmiştir.

#### PROJE EKİBİ

İREM KALKANLI (PROJE EKİP SORUMLUSU):

OKUL NUMARASI:190301007

DOĞUM TARİHİ:15.01.2000

DOĞUM YERİ: İSTANBUL

MEZUN OLDUĞU LİSE: ATAŞEHİR 3 DOĞA KOLEJİ

DENİZ UZUN:

OKUL NUMARASI:190301015

DOĞUM TARİHİ:08.04.2001

DOĞUM YERİ: İSTANBUL

MEZUN OLDUĞU LİSE: KAVACIK UĞUR ANADOLU LİSESİ

ÖZLEM ÇALI:

OKUL NUMARASI:190301002

DOĞUM TARİHİ:19.05.2000

DOĞUM YERİ: HATAY

MEZUN OLDUĞU LİSE: NECMİ ASFUROĞLU ANADOLU LİSESİ

İREM BOZKURT:

OKUL NUMARASI:190302010

DOĞUM TARİHİ:04.11.1998

DOĞUM YERİ: ADIYAMAN

MEZUN OLDUĞU LİSE: ÖZEL BİL KOLEJİ FEN LİSESİ

CEYDA UYMAZ:

OKUL NUMARASI:200301503

DOĞUM TARİHİ:26.08.2000

DOĞUM YERİ: İSTANBUL

MEZUN OLDUĞU LİSE: CELAL ARAS ANADOLU LİSESİ

#### REFERANS DOSYALAR

<https://youtu.be/ifEdgAdM9xk>

[https://github.com/iremkalkanli/Engel\\_kaldirici\\_sumo\\_robot](https://github.com/iremkalkanli/Engel_kaldirici_sumo_robot)

[https://github.com/iremkalkanli/Engel\\_kaldirici\\_sumo\\_robot](https://github.com/iremkalkanli/Engel_kaldirici_sumo_robot)KAYNAKLAR

- [1] Levent, Vecdi Emre (2020) “ZNYQ Mimarisi”, *System on Chip (SOC) Design -Ders Notları*.
- [2] Levent, Vecdi Emre (2020) “PL/PS CoProcessing”, *System on Chip (SOC) Design -Ders Notları*.
- [3] Levent, Vecdi Emre (2020) “Donanım Hızlandırıcı Projesi”, *System on Chip (SOC) Design-Ders Notları*.
- [4] Levent, Vecdi Emre (2020) “Interrupt’lar”, *System on Chip (SOC) Design-Ders Notları*.
- [5] Levent, Vecdi Emre (2020) “Interfaces II”, *System on Chip (SOC) Design-Ders Notları*.