Arduino ile Kendi Kendine Park Eden Araba Yapımı

Kendi kendine park etme işleminde belirli varsayımlara göre algoritmayı oluşturup sensörleri konumlandırmamız gerekiyor. Varsayımlarımız bu projemizde şu şekilde olacak. Senaryoda yolun sol kısmı duvar ve park alanlarından oluşacak. Videoda da göreceğiniz üzere arabanın sol tarafında 2 tane, arka ve ön kısmında da birer tane olmak üzere toplamda 4 tane sensörle etrafı algılamaktadır. Araba harekete başladığında sol taraftaki sensörler sürekli mesafe ölçerek park alanına uygun genişlik ararlar. Her iki sol sensör de park için uygun mesafe tespit ederse park işlemi algoritması devreye girer. Bu projede arabanın nasıl park edeceği iki farklı yöntemden oluşuyor.

**Sistemin Algoritması:**

Arabanın sol kenarındaki iki sensörün ölçtüğü değer 15cm den küçük ise duvar olduğunu anlar ve ileri hareket eder.Bunu hafızasına kaydeder. Daha sonra park aramaya başlar.İleri giderken Kenardaki iki sensör sürekli ölçüm yapar ve bu ölçümler sonucu ikisinin de değeri aynı anda 15cm den büyük ise, nasıl park edeceğine karar vermesi gerekiyor.

**Park Yöntemi Seçimi Algoritması**

* **1. durum :** Eğer ölçülen değer arabanın eninden büyük ve arabanın uzunluğundan küçük ise paralel park sistemini çalıştıracak.
* **2. durum :** Eğer ölçülen değer arabanın uzunluğundan büyük ise robot dik şekilde park edecektir.

**Paralel Park Etme Algoritması:**

Bu durumda araba park alanını geçerek, kenardaki iki sensör tekrar duvar gördüğünde araba durur. Biraz geri gelir. ve sağa 45 derece döner. Geriye hareket ederken arka sensör ölçüm yaparak park alanına girer ve sola doğru dönemeye başlar. Sola hareketi sırasında kenarlardaki sensörler sürekli ölçüm yapar ve iki sensörün ölçtüğü değer birbirine eşit oluncaya kadar sola dönmeye devam eder. Eşit olduğunda durur. Ön sensör ölçüm yapar ve 10 cm den küçük olana kadar ileri gider. ve 10 cm den küçük ise durur. ve park etme işlemi sona erer.

**Dik Park Etme Algoritması :**

Kenarlardaki sensörler aynı anda arabanın uzunluğundan fazla değer ölçmüş ise araba durur. ve sola 90 derece dönüş yapar. Park alanına doğru hareket etmeye başlar. Bu esnada ön sensör sürekli ölçüm yapar ve ölçülen değer 10cm den küçük ise araba durur. park işlemi tamamlanmış olur.

**Malzemeler:**

* Arduino Mega
* Adafruit Motor Shield
* 4 Dc Motorlu Robot Kit
* 4 Adet HC-SR04 Ultrasonik Sensör
* LM 393 Kızılötesi Hız Sensörü
* Lipo Pil ( 7.4V 850 mah yeterlidir)
* Jumper Kablolar

**Mekanik Kısım:**

Sistemde bulunan kızılötesi sensör motorun devrini ölçmektedir. Bu sayede park ederken tekerleklerin tur sayısını ölçerek hatasız park etmesini sağlamaktır. Robot kitinizde enkoder disk yoksa ek olarak alıp takabilirsiniz. Burada dikkat edilmesi gereken nokta enkoder disk üzerindeki delik sayısıdır. Bu projedeki enkoder diskin delik sayısı 20 dir. Sizdeki farklı sayıda ise arabanın dönüşlerini tekrar ayarlamanız gerekiyor.





LM393 Hız sensörünü yukarıdaki şekildeki gibi yerleştiriniz. Enkoder disk deliklerinin hızasında olmasına dikkat ediniz.

**Devre Şeması:**



**Ultrasonik Sensörlerin Pin Bağlantıları:**

**Ön Sensör** => **Trig Pin** : D34 , **Echo Pin** : D35 **VCC** : +5V **GND** : GND

**Sol Ön Sensör** => **Trig Pin** : D36 , **Echo Pin** : D37 **VCC** : +5V **GND** : GND

**Sol Arka Sensör** => **Trig Pin** : D38 , **Echo Pin** : D39 **VCC** : +5V **GND** : GND

**Arka Sensör** => **Trig Pin** : D40 , **Echo Pin** : D41 **VCC** : +5V **GND** : GND

**Motor Shield Dc Motor Pin Bağlantıları:**

**Sol Ön Motor** => **M4**

**Sağ Ön Motor** => **M3**

**Sol Arka Motor** => **M1**

**Sağ Arka Motor** => **M2**

**LM393 Hız Sensörü Pin Bağlantıları:**

**VCC** => **5V** : **OUT** => **D21** : **GND** => **GND**

Ve bir anolog ucu anolog girişe bağlıyoruz

**ARDUİNO YAZILIM (KOD) KISMI :**

// Kendi kendine park eden araba

// Kaan\_POLAT\_21652912

// Yusuf\_ŞİMŞEK\_21984397

// BaranCan\_YÜCEL\_21984508

//motorların tanımları

#include <AFMotor.h>

AF\_DCMotor Sol\_On\_Motor(4);

AF\_DCMotor Sag\_On\_Motor(3);

AF\_DCMotor Sol\_Arka\_Motor(1);

AF\_DCMotor Sag\_Arka\_Motor(2);

#include <Ultrasonic.h>

Ultrasonic ultrasonic\_arka(40,41),ultrasonic\_sol\_arka(38,39),ultrasonic\_sol\_on(36,37),ultrasonic\_on(34,35);

// ultrasomik sensörlerin tanımlamamalrı

#define Sol 0 //sol yön komutu

#define Sag 1 //sağ yön komutu

#define Ileri 2 //ileri yön komutu

#define Geri 3 //geri yön komutu

#define minimum\_limit 15 //Arabanın genişliği (cm)

#define minimum\_limit1 28 //arabanın uzunluğu (cm)

byte park\_durumu = 0;

**int** sinyalpin = 21;

**volatile** **int** val;

**int** sayac = 0;

**int** guncel\_durum = 0;

**int** onceki\_durum = 0;

**void** say(**int** saydir)

{

**for** (**int** i = 0 ; i <= saydir; i+1)

{

val = digitalRead(sinyalpin);

**if** (val == LOW) {

guncel\_durum = 0;

}

**else** {

guncel\_durum = 1;

}

**if**(guncel\_durum != onceki\_durum)

{

**if**(guncel\_durum == 1)

{

sayac = sayac + 1;

Serial.println(sayac);

i = i+1;

}

**else**

{

i = i ;

}

onceki\_durum = guncel\_durum;

}

**if** (i == saydir)

{

Sol\_On\_Motor.run(RELEASE);

Sag\_On\_Motor.run(RELEASE);

Sol\_Arka\_Motor.run(RELEASE);

Sag\_Arka\_Motor.run(RELEASE);

}

}

}

**void** motor\_pinSetup()

{

Sol\_On\_Motor.run(RELEASE);

Sag\_On\_Motor.run(RELEASE);

Sol\_Arka\_Motor.run(RELEASE);

Sag\_Arka\_Motor.run(RELEASE);

}

// Hareket fonksiyonları

**void** Robot\_Hareket(byte motor, byte spd)

{

**if** (motor == Ileri)

{

Sol\_On\_Motor.setSpeed(spd);

Sag\_On\_Motor.setSpeed(spd);

Sol\_Arka\_Motor.setSpeed(spd);

Sag\_Arka\_Motor.setSpeed(spd);

Sol\_On\_Motor.run(FORWARD);

Sag\_On\_Motor.run(FORWARD);

Sol\_Arka\_Motor.run(FORWARD);

Sag\_Arka\_Motor.run(FORWARD);

}

**if** (motor == Geri)

{

Sol\_On\_Motor.setSpeed(spd);

Sag\_On\_Motor.setSpeed(spd);

Sol\_Arka\_Motor.setSpeed(spd);

Sag\_Arka\_Motor.setSpeed(spd);

Sol\_On\_Motor.run(BACKWARD);

Sag\_On\_Motor.run(BACKWARD);

Sol\_Arka\_Motor.run(BACKWARD);

Sag\_Arka\_Motor.run(BACKWARD);

}

**if** (motor == Sol)

{

Sol\_On\_Motor.setSpeed(spd);

Sag\_On\_Motor.setSpeed(spd);

Sol\_Arka\_Motor.setSpeed(spd);

Sag\_Arka\_Motor.setSpeed(spd);

Sol\_On\_Motor.run(BACKWARD);

Sag\_On\_Motor.run(FORWARD);

Sol\_Arka\_Motor.run(BACKWARD);

Sag\_Arka\_Motor.run(FORWARD);

}

**if** (motor == Sag)

{

Sol\_On\_Motor.setSpeed(spd);

Sag\_On\_Motor.setSpeed(spd);

Sol\_Arka\_Motor.setSpeed(spd);

Sag\_Arka\_Motor.setSpeed(spd);

Sol\_On\_Motor.run(FORWARD);

Sag\_On\_Motor.run(BACKWARD);

Sol\_Arka\_Motor.run(FORWARD);

Sag\_Arka\_Motor.run(BACKWARD);

}

}

**void** Robot\_Dur()

{

Sol\_On\_Motor.run(RELEASE);

Sag\_On\_Motor.run(RELEASE);

Sol\_Arka\_Motor.run(RELEASE);

Sag\_Arka\_Motor.run(RELEASE);

}

// Park yeri araması

**bool** Park\_Yeri\_Kontrol()

{

**long** on\_Sensor = ultrasonic\_on.Ranging(CM);

**long** sag\_Sensor = ultrasonic\_sol\_on.Ranging(CM);

**long** sag\_arka\_Sensor =ultrasonic\_sol\_arka.Ranging(CM);

**if**( (sag\_Sensor <= minimum\_limit)&&(sag\_arka\_Sensor <= minimum\_limit)&&(park\_durumu == 0))

{

Robot\_Hareket(Ileri, 100);

park\_durumu = 1; Serial.println(park\_durumu);

}

**if**((sag\_Sensor > minimum\_limit)&&(sag\_Sensor < minimum\_limit1)&&(sag\_arka\_Sensor > minimum\_limit)&&(sag\_arka\_Sensor < minimum\_limit1)&&(park\_durumu == 1))

{

Robot\_Hareket(Ileri, 100);

park\_durumu = 2;Serial.println(park\_durumu);

}

**if**((sag\_Sensor >= minimum\_limit1)&&(sag\_arka\_Sensor >= minimum\_limit1)&&(park\_durumu == 1))

{

/\* Dik Park Etme Kararı \*/

Robot\_Dur() ;

delay(500);

park\_durumu = 10;Serial.println(park\_durumu);

}

**if**((sag\_Sensor <= minimum\_limit)&&(sag\_arka\_Sensor <= minimum\_limit)&&(park\_durumu == 2))

{

/\* Paralel Park Etme Kararı \*/

park\_durumu = 3; Serial.println(park\_durumu);

}

**return** park\_durumu;

}

**void** Park\_bul()

{

Park\_Yeri\_Kontrol();

**if**(park\_durumu == 3 )

{

Robot\_Dur();Serial.println(park\_durumu);

delay(400);

park\_durumu = 4;

}

**if**(park\_durumu == 4 )

{

Robot\_Hareket(Geri,120);

say(18);

Robot\_Dur();Serial.println(park\_durumu);

delay(500);

Robot\_Hareket(Sag,150);

say(9);

Robot\_Dur();

delay(500);

park\_durumu = 5;

}

**if**(park\_durumu == 5)

{

Robot\_Hareket(Geri,120);

**long** arka\_Sensor = ultrasonic\_arka.Ranging(CM);Serial.println(arka\_Sensor);

**if**(arka\_Sensor>0 && arka\_Sensor <= 13)

{

Robot\_Dur();

delay(400);

park\_durumu = 6;

}

**return** arka\_Sensor;

}

**if**(park\_durumu == 6)

{

Robot\_Hareket(Sol,150);

**long** sag\_Sensor = ultrasonic\_sol\_on.Ranging(CM); Serial.println(sag\_Sensor);

**long** sag\_arka\_Sensor = ultrasonic\_sol\_arka.Ranging(CM); Serial.println(sag\_arka\_Sensor);

**if**(sag\_Sensor == sag\_arka\_Sensor)

{

Robot\_Dur();

park\_durumu = 7;

}

**return** sag\_Sensor,sag\_arka\_Sensor;

}

**if**(park\_durumu == 7)

{

**long** on\_Sensor = ultrasonic\_on.Ranging(CM);

**if**(on\_Sensor<=6)

{

Robot\_Dur();

park\_durumu = 8;

}

**else**

{

Robot\_Hareket(Ileri,100);

}

**return** on\_Sensor;

}

**if** (park\_durumu ==10)

{

Robot\_Hareket(Sol,180);

say(14);

Robot\_Dur();

delay(500);

park\_durumu = 7;

}

}

**void** setup()

{

Serial.begin(9600);

attachInterrupt(5, say, CHANGE);

pinMode (sinyalpin, INPUT) ;

motor\_pinSetup();

}

**void** loop()

{

Park\_bul();

}

Kaan Polat 21652912

Baran Can Yücel 21984508

Yusuf Şimşek 21984397