**MEKATRONİK PROGRAMI**

**PROJE RAPORU**

**logo, simge, sembol, ticari marka, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**DERS ADI**

**MEKATRONİK SİSTEM TASARIMI**

**PROJE ADI**

**PUL TOPLAYAN ROBOT**

**PROJE EKİBİ**

**Görkem Çağkan Sevinç Umut Ergen Uğur Avcu**

**221098058 221098008 221098001**

**İÇİNDEKİLER**

Kullanılan Malzemeler

Pul Toplayan Robotun Amacı ve Özeti

Pul Toplayan Robotun Teknik Çizimi

Arduino Nano

MZ80 Kızılötesi Sensör

Arduino Shield

12V 500RPM DC Motor

L298N Motor Sürücü

TCS3200 Renk Sensörü

1.5V Alkalin Pil

SG90 Servo Motor

Bağlantı Şeması

Pul Toplayan Robotun Kodu

**Kullanılan Malzemeler**

Arduino Nano

Mz80 Kızılötesi Mesafe Sensörü x5

Arduino Shield

12V 500 RPM Motor x2

L298N Motor Sürücü

TCS3200 x2

1,5V Alkalin Pil x6

SG90 Servo Motor

Stop Butonu

Rgb Stick

Jumper Kablo

Silikon Tekerlek

**Pul Toplayan Robotun Özeti**

Arduino tabanlı pul toplayan robot, (4 cm x 2 cm) boyutlardaki pulları algılayıp toplama kabiliyetine sahip bir robottur. Sensörler ve motorlar kullanılarak tasarlanan bu robot, pul toplama görevini otomatik olarak gerçekleştirecek şekilde programlanmıştır. Robot, sensörleri aracılığıyla pulları algılar ve konumlarını belirleyerek toplama mekanizmasını devreye sokar. Arduino'nun esnek programlama yetenekleri sayesinde, robotun hareketleri ve toplama işlemi hassas bir şekilde kontrol edilmiştir. Bu proje, robotik uygulamalar için Arduino'nun pratikliğini ve kullanım kolaylığını gösterirken, eğitim ve endüstriyel alanlarda farklı amaçlarla kullanılabilir.

elektronik donanım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Pul Toplayan Robotun Amacı**

Pul toplayan robot, (4 cm x 2 cm) boyutlardaki pulları otomatik olarak algılayıp toplamak ve belirlenen bir alanda biriktirmektir. Bu robot, sensörler ve motorlar kullanarak pulların konumunu tespit eder ve toplama işlemini gerçekleştirir. Bu proje, otomatik toplama süreçlerini optimize etmek, iş gücünü azaltmak ve verimliliği artırmak için tasarlanmıştır.

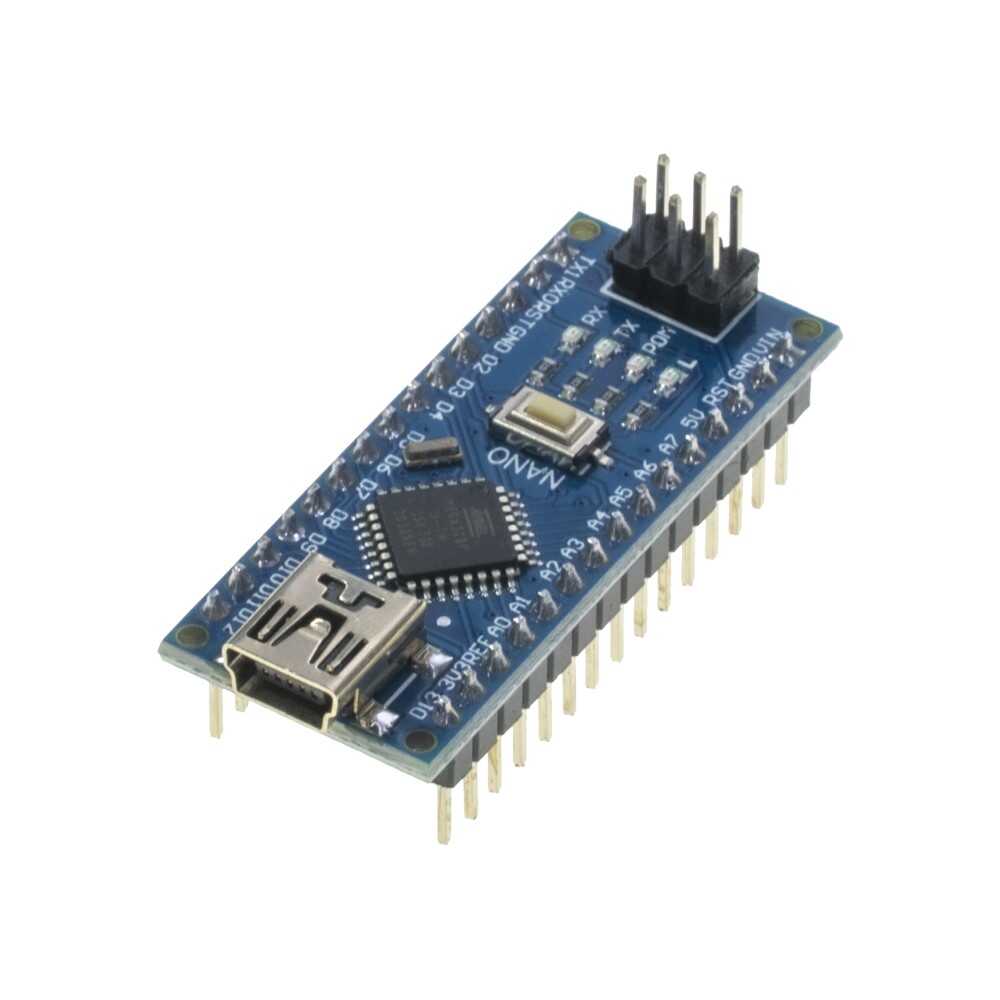
**Pul Toplayan Robotun Teknik Çizimi**

**taslak, çizim, diyagram, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Arduino Nano**

Arduino Nano, Atmega328P mikrodenetleyici çipine sahiptir ve Arduino.cc tarafından geliştirilen açık kaynaklı bir mikrodenetleyici kartıdır. Arduino Nano, diğer devrelere arayüzlenebilen dijital ve analog giriş/çıkış pinleri ile donatılmıştır. Kartın 14 dijital pini (altı PWM çıkışı), 8 analog pini vardır ve mini-B tipi USB kablosu ile Arduino IDE ile programlanabilmektedir. 7 ile 12V arasındaki voltajları kabul etse de USB kablo veya harici bir güç kaynağı ile güçlendirilebilmektedir. Küçük boyutları ve düşük maliyeti sayesinde kompakt projeler ve taşınabilir elektronik uygulamalar için ideal bir seçenektir.



**Özellikleri**

* Dijital G / Ç: 14 adet pin deliği (6 PWM Destekli)
* Analog G / Ç: 8 adet pin deliği
* DC Akım G/Ç Pinleri: 40mA
* Giriş Voltajı: DC 7V ~ 12V (Tavsiye)
* Girişi Voltajı: 6V-20V (Sınırları)
* Çalışma Voltajı (Lojik): 5V
* Atmega328 mikroişlemci denetleyici
* Flash Hafıza: 32KB
* SRAM: 2KB (ATmega328)
* EEPROM: 1KB (ATmega328)
* Çalışma Frekansı: 16MHz
* Boyutlar: 18.5mm x 43.2mm

**MZ80 Kızılötesi Sensörü**

MZ80 kızıl ötesi mesafe sensörüdür, NPN çıkışlı olup 5V ile çalışır. 3 ila 80cm aralığındaki mesafeyi algılamaktadır. Arkasındaki pot ile ayarlanabilir mesafe özelliğiyle sınıfına göre çok kullanışlıdır. Ayarlanan mesafe aralığına bir engel geldiği zaman dijital voltaj çıkışı verir. Otomasyon sistemleri, cisim sayma, güvenlik sistemleri gibi birçok alanda kullanılır.

tasarım, tarak içeren bir resim

Açıklama otomatik olarak orta güvenilirlik düzeyiyle oluşturuldu

**Özellikleri**

* Besleme voltajı: 5v
* Akım Tüketimi : > 25mA (min) ~ 100mA (max)
* Boyutlar : 1.7cm (çap) x 4.5cm (uzunluk)
* Kablo uzunluğu : 45cm
* Nesne algılaması : saydam ya da mat
* Ölçüm aralığı : 3cm  - 80cm
* Çalışma sıcaklık aralığı : -25 °C ~ 55 °C

**Bağlantı**

* Kahverengi: +5V
* Mavi: GND
* Siyah: DATA

**Arduino Shield**

Arduino pinleri 3 pin formatında (GND 5V Signal) olduğundan sensörler ve çeşitli giriş çıkışlar kolaylıkla bağlanabilir. Reset pini kart üzerinde taşınmaktadır ve DC adaptör girişi diğer Arduino kartlarında olduğu gibi 2.1 mm merkez pozitif girişlidir. 5V, 3.3V ve GND güç pinleri çoğaltılarak karta alınır.



**L298N Özellikleri**

* Arduino Nano ile uyumludur, tüm pinlerin düzenini shield ile aynı yapar.
* Arduino Uno, Leonardo ve Mega ile çalışabilmektedir.
* Sensörler ve servo motorlar gibi çeşitli giriş ve çıkışları kolaylıkla kullanabilmek için 3 pinli düzene sahiptir. Her pimin yanında topraklama ve besleme pimleri bulunur.
* Reset pini kart üzerinde taşınır.
* 7-12V arası voltajı beslemek için kart üzerinde adaptör bulunmaktadır.
* Tahta boyutları: 58x59mm

**12V 500 RPM Motor**

12V 500 RPM motor, nominal olarak 12 volt gerilimde çalışan ve boşta 500 devir/dakika dönüş hızına sahip bir elektrik motorudur.



**Özellikleri**

* Çalışma Voltajı: 12V
* Devir: 500 Rpm/Dk
* Boşta Çektiği Akım: 180mA ± 40mA
* Zorlama Akımı: 0.9A
* Motor Çapı: 12mm
* Tork: 2 kg/cm
* Redüktör Çapı: 12mm
* Mil: 3mm D Şaft
* Mil Uzunluğu: 10mm
* Uzunluk (Mil Hariç): 24mm
* Ağırlık: 9.5gr

**L298N Motor Sürücü Devresi**

L298N voltaj regülatörlü çift motor sürücü kartı, 4.8V ile 24V’a kadar olan motorları sürmek için hazırlanmış olan bir motor sürücü kartıdır. Bu motor sürücü kartı iki kanalı olup, kanal başına 2A akım vermektedir. Kart üzerinde L298N motor sürücü entegresi kullanılmıştır.

metin, ekran görüntüsü, grafik, grafik tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**L298N Özellikleri**

* Birbirinden bağımsız olarak iki ayrı motoru kontrol edebilir.
* Kanal başına 2A akım verebilmektedir.
* Üzerinde dahili regülatörü vardır.
* Yüksek sıcaklık ve kısa devre koruması vardır.
* Motor dönüş yönüne göre yanan ledler vardır.
* Kart üzerinde dahili soğutucu vardır.

**Pin Bağlantıları**

* ENA: Sol motor kanalını aktif etme pini
* IN1: Sol motor 1. girişi
* IN2: Sol motor 2. gitişi
* IN3: Sağ motor 1. girişi
* IN4: Sağ motor 2. girişi
* ENB: Sağ motor kanalını aktif etme pini
* MotorA: Sol motor çıkışı
* MotorB: Sağ motor çıkışı
* VCC: Besleme voltaj girişi (4.8V-24V)
* GND: Toprak bağlantısı
* 5V: 5V çıkış

**TCS3200 (Renk Sensörü)**

TCS3200 Renk Sensörü Kartı, üzerinde TAOS TCS3200 çip ve 4 adet beyaz led bulunan bir renk sensörüdür. Ortamdaki ışığı algılama, test şeridi okuma, renge göre sıralama ve kalibrasyon gibi pek çok renk uygulamalarında kullanılabilir. Hemen hemen tüm renkleri algılayabilir. Çıkış frekansı 2Hz – 500kHz arasıdır. Ürün üzerinde bulunan sensör yuvası sayesinde dış ortama karşı etkiler minimuma indirilmiştir, daha iyi bir renk algılama performansı sergiler.



**TCS3200 Özellikleri**

* Giriş voltajı: 2.7 V-5 V
* Arayüz: Dijital TTL
* Sensör yuvası sayesinde daha sağlıklı ölçüm
* Programlanabilir ve tam ölçekli çıkış frekansı
* Mikrodenetleyiciler ile doğrudan iletişim kurar.
* Boyutları: 28.4x28.4mm

**1.5V Alkalin Pil**

1.5V pilleri, enerji depolama ve verimlilik konusunda üstün performans sunan bileşenlerdir. Bu piller, hem alkalin hem de lityum kimyasal yapılarında üretilir ve geniş bir uygulama yelpazesi için uygundur.



**1,5V Alkalin Pil Özellikleri**

* Voltaj: 1.5V
* Kimyasal Yapı: Alkalin (çinko-mangan dioksit)
* Kapasite: AA pillerde yaklaşık 2000-3000 mAh, AAA pillerde yaklaşık 1000-1200 mAh
* Raf Ömrü: Yaklaşık 5-10 yıl (kullanılmadığında)
* Kullanım Alanları: Oyuncaklar, uzaktan kumandalar, saatler, el fenerleri, küçük elektronik cihazlar
* Çalışma Sıcaklığı Aralığı: -20°C ila 54°C
* Boyutlar:
* AA: 50.5 mm uzunluk, 14.5 mm çap
* AAA: 44.5 mm uzunluk, 10.5 mm çap

**SG90 (Servo Motor)**

Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüye ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.

**metin, plastik içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**SG90 Özellikleri**

* Çalışma gerilimi: 4.8 - 6.0 VDC
* Hız :4.8V: 0.1 sn/60°
* Zorlanma Torku :6V: 1.8 kg.cm
* Dişli kutusu: Plastik
* Dönüş açısı: 0-180°
* Kablo Uzunluğu: 15 cm
* Boyutlar: 23.1 x 12.2 x 29 mm
* Ağırlık: 9 g

**PUL TOPLAYAN ROBOTUN BAĞLANTI ŞEMASI**

**diyagram, devre, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**PUL TOPLAYAN ROBOTUN KODLARI**

#include <Servo.h>

#include <EEPROM.h>

#define ENA 6

#define IN1 3

#define IN2 2

#define IN3 1

#define IN4 0

#define ENB 5

#define S3 9

#define sensorfloor 7 //iç sensör D7

#define S2 8    //s0 5v  --- s1 gnd

#define sensoregg 4 //  Dış sensör D4

int hiz = 210;

int frequency = 0;

#define bsens   A5

#define lsens   A1

#define rsens   A4

#define flsens  A0

#define frsens  A3

Servo servo;

int Regg = 0, Cegg = 0, Begg = 0;

int Rfloor = 0, Cfloor = 0, Bfloor = 0;

int corner, eggr, eggb, eggc, whitefloor = 0;

bool eggflag = 0;

/////////////////////////////////////////////////////////////motor

void motorcontrol(int left , int  right)

{

  if (left < 0)

  {

    digitalWrite(IN1, 1);

    digitalWrite(IN2, 0);

    analogWrite(ENA, abs(left));

  }

  else if (left > 0)

  {

    digitalWrite(IN1, 0);

    digitalWrite(IN2, 1);

    analogWrite(ENA, left);

  }

  else

  {

    digitalWrite(IN1, 1);

    digitalWrite(IN2, 1);

    analogWrite(ENA, 0);

  }

  if (right < 0)

  {

    digitalWrite(IN3, 1);

    digitalWrite(IN4, 0);

    analogWrite(ENB, abs(right));

  }

  else if (right > 0)

  {

    digitalWrite(IN3, 0);

    digitalWrite(IN4, 1);

    analogWrite(ENB, right);

  }

  else

  {

    digitalWrite(IN3, 1);

    digitalWrite(IN4, 1);

    analogWrite(ENB, 0);

  }

}

////////////////////////////////////////////////////////////////////sensor okuma

void rgbsens()

{

  digitalWrite(S2, LOW);

  digitalWrite(S3, LOW);

  Regg = pulseIn(sensoregg, LOW);

  delay(10);

  digitalWrite(S2, HIGH);

  digitalWrite(S3, LOW);

  Cegg = pulseIn(sensoregg, LOW);

  delay(10);

  digitalWrite(S2, LOW);

  digitalWrite(S3, HIGH);

  Begg = pulseIn(sensoregg, LOW);

  delay(10);

  digitalWrite(S2, LOW);

  digitalWrite(S3, LOW);

  Rfloor = pulseIn(sensorfloor, LOW);

  delay(10);

  digitalWrite(S2, HIGH);

  digitalWrite(S3, LOW);

  Cfloor = pulseIn(sensorfloor, LOW);

  delay(10);

  digitalWrite(S2, LOW);

  digitalWrite(S3, HIGH);

  Bfloor = pulseIn(sensorfloor, LOW);

  delay(10);

}

void setup() {

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);

  pinMode(IN3, OUTPUT);

  pinMode(IN4, OUTPUT);

    pinMode(ENB, OUTPUT);

  pinMode(ENA, OUTPUT);

  pinMode(bsens , INPUT);

  pinMode(lsens , INPUT);

  pinMode(rsens , INPUT);

  pinMode(flsens , INPUT);

  pinMode(frsens , INPUT);

  pinMode(S2, OUTPUT);

  pinMode(S3, OUTPUT);

  pinMode(sensorfloor, INPUT);

  pinMode(sensoregg, INPUT);

  servo.attach(13);

  motorcontrol(0, 0);

  if (Rfloor < Bfloor)

  {

    corner = 1; //corner is Red

  }

  else if (Rfloor > Bfloor)

  {

    corner = 0; //corner is Blue

  }

 // Serial.begin(9600);

  servo.write(0);

for(int i=0;i<20;i++)

{

    rgbsens();

    eggr = Rfloor;

    eggb = Bfloor;

    whitefloor = 15;

  }

   motorcontrol(hiz, hiz);

     while(Cegg > whitefloor +3)

  {

    //  Serial.println("Renkli Bolgede");

    rgbsens();

  }

}

void loop() {

/\*

  // Test

Serial.print(digitalRead(bsens));

Serial.print("  ");

  Serial.print(digitalRead(lsens));

  Serial.print("  ");

  Serial.print(digitalRead(flsens));

  Serial.print("  ");

  Serial.print(digitalRead(frsens));

  Serial.print("  ");

  Serial.println(digitalRead(rsens));

  delay(50);

  \*/

 if (digitalRead(lsens) == 1&& digitalRead(rsens) == 1)

  {

    motorcontrol(-1\*hiz, -1\*hiz);

    delay(600);

     motorcontrol(-1\*hiz, hiz);

     delay(600);

  }

else if (eggflag == 1 &&digitalRead(flsens) == 1&&( digitalRead(bsens) == 1 || digitalRead(rsens) == 1 ) )

  {

    motorcontrol(-1\*hiz, hiz);

    delay(50);

  }

 else if (digitalRead(frsens) == 1 )

  {

    motorcontrol(-1\*hiz, hiz);

    delay(50);

  }

  else if ( digitalRead(flsens) == 1)

  {

    motorcontrol(hiz, -1\*hiz);

    delay(50);

  }

  else if (eggflag == 1 && digitalRead(bsens) == 1 && digitalRead(rsens) == 1)

  {

  }

  else if (eggflag == 1 && digitalRead(rsens) == 1)

  {

    motorcontrol(0, hiz);

    while(digitalRead(rsens) == 1&& digitalRead(bsens) == 0){delay(10);}

    delay(1);

  }

   else if (eggflag == 1 && digitalRead(bsens) == 1)

  {

    motorcontrol(hiz, 0);

    while(digitalRead(bsens) == 1&& digitalRead(rsens) == 0){delay(10);}

    delay(1);

  }

/\*else if (digitalRead(lsens) == 0&&digitalRead(frsens) == 0 )

  {

    motorcontrol(hiz, -1\*hiz);

    delay(200);

  }

else if (digitalRead(rsens) == 0&&digitalRead(flsens) == 0)

  {

    motorcontrol(-1\*hiz, hiz);

    delay(200);

  }

  \*/

  else if (digitalRead(rsens) == 1)

  {

    motorcontrol(-1\*hiz, hiz);

  }

  else if (digitalRead(lsens) == 1)

  {

    motorcontrol(hiz, -1\*hiz);

  }

  else

  {

    motorcontrol(hiz, hiz);

  }

  rgbsens();

     if (Cegg > whitefloor)

  {

  delay(10);

  rgbsens();

  if ((eggflag == 1)&&(Rfloor > eggr - 30 && Rfloor < eggr + 30) && (Bfloor > eggb - 30 && Bfloor < eggb + 30) )

  { motorcontrol(0, 0);

    delay(200);

    if ((eggflag == 1)&&(Rfloor > eggr - 30 && Rfloor < eggr + 30) && (Bfloor > eggb - 30 && Bfloor < eggb + 30) )

  {

    servo.write(0);

    eggflag = 0;

    motorcontrol(-0.8\*hiz, -0.8\*hiz);

    delay(700);

    motorcontrol(-1\*hiz, hiz);

    delay(500);

  }

  }

 else if ((Regg > eggr - 30 && Regg < eggr + 30) && (Begg > eggb - 30 && Begg < eggb + 30))

  {

    servo.write(100);

    eggflag = 1;

  }

     if (eggflag == 0 && Cegg > whitefloor + 10)

  {

    delay(200);

    if (eggflag == 0 && Cegg > whitefloor + 10)

  {

    motorcontrol(-1\*hiz, -1\*hiz);

    delay(200);

    motorcontrol(-1\*hiz, hiz);

    delay(200);

  }

  }

  }

}