

Engelden Kaçan Robot

ÖĞRETMEN KILAVUZU

Kategori	Robotik ve Kodlama	Süre	Uzun proje (5–7 ders)
Seviye	Orta	Hedef	Temel bilgisi olan orta seviye lise öğrencileri

Amaç

- Algıla-karar-ver-hareket et döngüsünü güvenli robot davranışına dönüştürmek.

Problem durumu

- Hareketli robot çarpışmadan ilerlemek için önündeki engeli algılamalıdır

Kazanımlar

- Mesafe verisini hareket kararına bağlar.
- Motor hareketlerini fonksiyonlaştırır.
- Kaçınma davranışını test eder.

Hazırlık

- Engelden Kaçan Robot için örnek girdileri, güvenlik sınırlarını ve test ölçütlerini dersten önce kontrol etmek.

Araç / malzemeler

- Robot şasesi
- Arduino
- HC-SR04
- Motor sürücü
- DC motorlar
- Uygun batarya

Uygulama

- Şaseyi kur.
- Motorları ayrı test et.
- Sensörü doğrula.
- Hareket fonksiyonlarını yaz.
- Kaçınma akışını ekle.

Test aşaması

- Düz duvar
- Dar köşe
- Sensörün algılamadığı yumuşak/eğik yüzey

Olası hatalar ve çözümler

- Motor yönü ters: Girdiyi veya bağlantıyı tek tek doğrula; değişiklikten sonra aynı testi yeniden çalıştır.
- Batarya zayıf: Girdiyi veya bağlantıyı tek tek doğrula; değişiklikten sonra aynı testi yeniden çalıştır.
- Sensör kör noktası: Girdiyi veya bağlantıyı tek tek doğrula; değişiklikten sonra aynı testi yeniden çalıştır.

- Dönüş süresi yetersiz: Girdiyi veya bağlantıyı tek tek doğrula; değişiklikten sonra aynı testi yeniden çalıştır.

Güvenlik

- İlk motor testinde tekerlekleri kaldırın.
- Batarya kutuplarını ve sürücü akım sınırını kontrol edin.

Öğretmen yönergeleri

- Hazır çözümü vermeden önce öğrenciden tahminini ve denediği adımı açıklamasını isteyin.
- Sonucu yalnız çalışıyor/çalışmıyor olarak değil; test kanıtı ve açıklamayla değerlendirin.

Geliştirme önerileri

- Servo üstünde sensör
- Rastgele yön seçimi
- Hız kontrolü

Kaçınma algoritması

Motor sürücü modeli pin ve yön fonksiyonlarını değiştirir; kod iskeleti kullanılan karta göre tamamlanmalıdır.

```
// L298N örneği: başka sürücülerde pin ve lojik değişebilir.
const int trig = 12, echo = 11, ena = 5, enb = 6;
const int in1 = 7, in2 = 8, in3 = 9, in4 = 10;
void hareket(int sol1, int sol2, int sag1, int sag2, int hiz) {
  digitalWrite(in1, sol1); digitalWrite(in2, sol2);
  digitalWrite(in3, sag1); digitalWrite(in4, sag2);
  analogWrite(ena, hiz); analogWrite(enb, hiz);
}
float mesafeOku() {
  digitalWrite(trig, LOW); delayMicroseconds(2);
  digitalWrite(trig, HIGH); delayMicroseconds(10); digitalWrite(trig, LOW);
  long sure = pulseIn(echo, HIGH, 30000);
  return sure == 0 ? -1 : sure * 0.0343 / 2;
}
void setup() {
  pinMode(trig, OUTPUT); pinMode(echo, INPUT);
  pinMode(ena, OUTPUT); pinMode(enb, OUTPUT);
  pinMode(in1, OUTPUT); pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT); pinMode(in4, OUTPUT);
}
void loop() {
  float cm = mesafeOku();
  if (cm > 0 && cm < 20) {
    hareket(LOW, LOW, LOW, LOW, 0); delay(200);
    hareket(LOW, HIGH, LOW, HIGH, 140); delay(400);
    hareket(HIGH, LOW, LOW, HIGH, 140); delay(450);
  } else { hareket(HIGH, LOW, HIGH, LOW, 150); }
}
```