

```
// Çığor Çizgi İzleyen Robot Kodu - Optimize Edilmiş Versiyon
```

```
// Rokart Dipswitch Hız Kontrolü (%25, %50, %75, %100)
```

```
// --- SENSÖR PİNLERİ ---
```

```
#define sag_cizgi 7
```

```
#define orta_cizgi 6
```

```
#define sol_cizgi 5
```

```
// --- MOTOR PİNLERİ ---
```

```
#define sol_motor1 A4
```

```
#define sol_motor2 A3
```

```
#define sol_pwm 9
```

```
#define sag_motor1 A1
```

```
#define sag_motor2 A2
```

```
#define sag_pwm 10
```

```
// --- GÖSTERGE VE KONTROL PİNLERİ ---
```

```
#define D3 13
```

```
#define D2 11
```

```
#define D1 12
```

```
#define buzzer 8
```

```
#define switch1 4
```

```
#define switch2 3
```

```
#define switch3 2
```

```
#define buton A0
```

```
// --- DEĞİŞKENLER ---
```

```
bool son_sag = 0;
bool son_sol = 0;
int hiz = 0;

void setup() {
    // Sensör Pinleri
    pinMode(sag_cizgi, INPUT);
    pinMode(orta_cizgi, INPUT);
    pinMode(sol_cizgi, INPUT);

    // Motor Pinleri
    pinMode(sol_motor1, OUTPUT);
    pinMode(sol_motor2, OUTPUT);
    pinMode(sol_pwm, OUTPUT);

    pinMode(sag_motor1, OUTPUT);
    pinMode(sag_motor2, OUTPUT);
    pinMode(sag_pwm, OUTPUT);

    // LED ve Buzzer Pinleri
    pinMode(D1, OUTPUT);
    pinMode(D2, OUTPUT);
    pinMode(D3, OUTPUT);
    pinMode(buzzer, OUTPUT);

    // Switch ve Buton (Dahili Pull-up dirençleri aktif)
    pinMode(switch1, INPUT_PULLUP);
    pinMode(switch2, INPUT_PULLUP);
    pinMode(switch3, INPUT_PULLUP);
}
```

```
pinMode(buton, INPUT_PULLUP);

// Başlangıçta motorları durdur
motorSur(0, 0, 0, 0, 0, 0);

// -----
// MOTOR HIZ AYARLARI (Modlar arası geçişte Nano'yu resetleyiniz)
// -----

// Mod 1: %25 Hız
if (digitalRead(switch1) == 1 && digitalRead(switch2) == 1 && digitalRead(switch3) == 1) {
    hiz = 64;
    digitalWrite(D1, HIGH); digitalWrite(D2, HIGH); digitalWrite(D3, HIGH);
}

// Mod 2: %50 Hız
else if (digitalRead(switch1) == 0 && digitalRead(switch2) == 1 && digitalRead(switch3) == 1) {
    hiz = 127;
    digitalWrite(D1, HIGH); digitalWrite(D2, LOW); digitalWrite(D3, LOW);
}

// Mod 3: %75 Hız
else if (digitalRead(switch1) == 0 && digitalRead(switch2) == 0 && digitalRead(switch3) == 1) {
    hiz = 191;
    digitalWrite(D1, LOW); digitalWrite(D2, HIGH); digitalWrite(D3, LOW);
}

// Mod 4: %100 Hız
else if (digitalRead(switch1) == 0 && digitalRead(switch2) == 0 && digitalRead(switch3) == 0) {
    hiz = 255;
    digitalWrite(D1, LOW); digitalWrite(D2, LOW); digitalWrite(D3, HIGH);
}
```

```
else {  
    // Tanımsız bir switch durumu olursa varsayılan %50 hız olsun  
    hiz = 127;  
}  
  
// Başlama uyarısı  
digitalWrite(buzzer, HIGH);  
delay(200);  
digitalWrite(buzzer, LOW);  
}  
  
void loop() {  
    // Sensör okumaları  
    int sol = digitalRead(sol_cizgi);  
    int orta = digitalRead(orta_cizgi);  
    int sag = digitalRead(sag_cizgi);  
  
    // SİYAH ZEMİN - BEYAZ ÇİZGİ MANTIĞI  
    // Eğer zemin beyaz, çizgi siyah ise sensör okumaları (0 ve 1'ler) tersine dönecektir.  
  
    // 1. Durum: Sadece ORTA sensör çizgiyi görüyor (Düz git)  
    if (sol == 0 && orta == 1 && sag == 0) {  
        motorSur(HIGH, LOW, hiz, HIGH, LOW, hiz);  
        son_sol = 0; son_sag = 0;  
    }  
  
    // 2. Durum: Sadece SOL sensör çizgiyi görüyor (Sola dön)  
    else if (sol == 1 && orta == 0 && sag == 0) {  
        // Keskin sola dönüş: Sol motor geri, sağ motor ileri
```

```
motorSur(LOW, HIGH, hiz, HIGH, LOW, hiz);  
son_sol = 1; son_sag = 0;  
}
```

```
// 3. Durum: Sadece SAĞ sensör çizgiyi görüyor (Sağa dön)
```

```
else if (sol == 0 && orta == 0 && sag == 1) {  
    // Keskin sağa dönüş: Sol motor ileri, sağ motor geri  
    motorSur(HIGH, LOW, hiz, LOW, HIGH, hiz);  
    son_sol = 0; son_sag = 1;  
}
```

```
// 4. Durum: Çizgi tamamen kaybedildi (Son duruma göre etrafında dönerek çizgiyi ara)
```

```
else if (sol == 0 && orta == 0 && sag == 0) {  
    if (son_sol == 1) {  
        // En son çizgiyi solda görmüştü, sola doğru kendi etrafında dön  
        motorSur(LOW, HIGH, hiz, HIGH, LOW, hiz);  
    }  
    else if (son_sag == 1) {  
        // En son çizgiyi sağda görmüştü, sağa doğru kendi etrafında dön  
        motorSur(HIGH, LOW, hiz, LOW, HIGH, hiz);  
    }  
    else {  
        // Başlangıçta çizgi yoksa dur  
        motorSur(LOW, LOW, 0, LOW, LOW, 0);  
    }  
}
```

```
// 5. Durum: T kavşak veya tam kesişim (Bütün sensörler çizgiyi görüyor)
```

```
else if (sol == 1 && orta == 1 && sag == 1) {
```

```
// T kavşakta düz gitmeyi tercih et
motorSur(HIGH, LOW, hiz, HIGH, LOW, hiz);
}
}

// --- YARDIMCI FONKSİYON: MOTOR SÜRME ---
// Bu fonksiyon kod tekrarını önler ve motorları tek satırda kontrol etmeni sağlar.
void motorSur(bool sol_1, bool sol_2, int sol_hiz, bool sag_1, bool sag_2, int sag_hiz) {
    digitalWrite(sol_motor1, sol_1);
    digitalWrite(sol_motor2, sol_2);
    analogWrite(sol_pwm, sol_hiz);

    digitalWrite(sag_motor1, sag_1);
    digitalWrite(sag_motor2, sag_2);
    analogWrite(sag_pwm, sag_hiz);
}
```