**Havalimanlarında Bagaj Ve Yük Taşıyabilen Robot**

Bu çalışma havalimanlarında bagaj ve yük taşımayı kolaylaştırmak ve insanların bu konudaki iş yükünü azaltmak için yaptığım robot çalışmasıdır. Robotu çizgi izleyen sensörler kullanılarak havalimanına giriş yapan insanların bagajlarının robot tarafından teslim alınıp ilgili yerlere götürülmesi planlanmaktadır. Çizgi izleyen robot, gideceği yolu otonom olarak takip eden robot tipidir. Robot bu takip etme işlemini sensörleri vasıtasıyla renk farkını algılayarak yapmaktadır. Havalimanına oluşturacağımız yollar sayesinde bu işlemi yapacak, insanların üzerinde ki bu yükü alacak ve bu alanda hayatı kolaylaştıracaktır.

**Ardunio Yazılımı:**

#define SensorSol 10

#define SensorOrta 11

#define SensorSag 12

void setup() {

pinMode(SensorSol, INPUT);

pinMode(SensorOrta, INPUT);

pinMode(SensorSag, INPUT);

Serial.begin(9600);

Serial.println("Cizgi Sensoru Testi");

delay(2000);

}

void loop() {

Serial.print(digitalRead(SensorSol));

Serial.print(",");

Serial.print(digitalRead(SensorOrta));

Serial.print(",");

Serial.println(digitalRead(SensorOrta));

delay(100);

}

#define SensorSol 10

#define SensorOrta 11

#define SensorSag 12

#define MotorR1 6

#define MotorR2 7

#define MotorRE 9

#define MotorL1 5

#define MotorL2 4

#define MotorLE 3

void setup() {

pinMode(SensorSol, INPUT);

pinMode(SensorOrta, INPUT);

pinMode(SensorSag, INPUT);

pinMode(MotorR1, OUTPUT);

pinMode(MotorR2, OUTPUT);

pinMode(MotorL1, OUTPUT);

pinMode(MotorL2, OUTPUT);

}

void loop() {

if (digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 0) {

ileri();

}

if (digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 1) {

sag();

}

if (digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0) {

sol();

}

}

void ileri() {

digitalWrite(MotorR1, HIGH);

digitalWrite(MotorR2, LOW);

analogWrite(MotorRE, 150);

digitalWrite(MotorL1, HIGH);

digitalWrite(MotorL2, LOW);

analogWrite(MotorLE, 150);

}

void sag() {

digitalWrite(MotorR1, HIGH);

digitalWrite(MotorR2, LOW);

analogWrite(MotorRE, 0);

digitalWrite(MotorL1, HIGH);

digitalWrite(MotorL2, LOW);

analogWrite(MotorLE, 150);

}

void sol() {

digitalWrite(MotorR1, HIGH);

digitalWrite(MotorR2, LOW);

analogWrite(MotorRE, 150);

digitalWrite(MotorL1, HIGH);

digitalWrite(MotorL2, LOW);

analogWrite(MotorLE, 0);

}

void dur() {

digitalWrite(MotorR1, HIGH);

digitalWrite(MotorR2, LOW);

digitalWrite(MotorRE, LOW);

digitalWrite(MotorL1, HIGH);

digitalWrite(MotorL2, LOW);

digitalWrite(MotorLE, LOW);

}

**Robotun Görüntüleri:**

