**Dual-axis-power-system**

#include <Servo.h>

// Define servos

Servo servoX; // Horizontal

Servo servoY; // Vertical

// LDR pins

const int ldrTopLeft = A0;

const int ldrTopRight = A1;

const int ldrBottomLeft = A2;

const int ldrBottomRight = A3;

// Servo positions

int posX = 90; // Initial position for horizontal servo

int posY = 90; // Initial position for vertical servo

void setup() {

// Attach servos to pins

servoX.attach(9);

servoY.attach(10);

// Initialize servos to mid position

servoX.write(posX);

servoY.write(posY);

// Begin serial communication for debugging

Serial.begin(9600);

}

void loop() {

// Read LDR values

int topLeft = analogRead(ldrTopLeft);

int topRight = analogRead(ldrTopRight);

int bottomLeft = analogRead(ldrBottomLeft);

int bottomRight = analogRead(ldrBottomRight);

// Debugging: Print LDR values

Serial.print("TL: "); Serial.print(topLeft);

Serial.print(" TR: "); Serial.print(topRight);

Serial.print(" BL: "); Serial.print(bottomLeft);

Serial.print(" BR: "); Serial.println(bottomRight);

// Calculate average light intensity for top and bottom pairs

int topAvg = (topLeft + topRight) / 2;

int bottomAvg = (bottomLeft + bottomRight) / 2;

int leftAvg = (topLeft + bottomLeft) / 2;

int rightAvg = (topRight + bottomRight) / 2;

// Adjust vertical servo (Y axis)

if (abs(topAvg - bottomAvg) > 50) { // Threshold to prevent constant movement

if (topAvg > bottomAvg) {

posY += 1; // Tilt up

} else {

posY -= 1; // Tilt down

}

posY = constrain(posY, 0, 180); // Limit servo position

servoY.write(posY);

}

// Adjust horizontal servo (X axis)

if (abs(leftAvg - rightAvg) > 50) { // Threshold to prevent constant movement

if (leftAvg > rightAvg) {

posX += 1; // Turn right

} else {

posX -= 1; // Turn left

}

posX = constrain(posX, 0, 180); // Limit servo position

servoX.write(posX);

}

// Small delay to allow servo to move

delay(20);

}