**Sumo robot Project code:**

#include <Arduino.h>

Code for Pushing Five Objects Out of the Platform

// Pin definitions

#define LEFT\_MOTOR\_PIN1 2

#define LEFT\_MOTOR\_PIN2 3

#define RIGHT\_MOTOR\_PIN1 4

#define RIGHT\_MOTOR\_PIN2 5

#define DISTANCE\_SENSOR\_PIN A3

#define MODE\_SWITCH\_PIN 6

// Constants

#define OBJECT\_REMOVING 1

#define TIME\_LIMIT 60000 // 60 seconds

// Variables

unsigned long startTime;

void setup() {

pinMode(LEFT\_MOTOR\_PIN1, OUTPUT);

pinMode(LEFT\_MOTOR\_PIN2, OUTPUT);

pinMode(RIGHT\_MOTOR\_PIN1, OUTPUT);

pinMode(RIGHT\_MOTOR\_PIN2, OUTPUT);

pinMode(DISTANCE\_SENSOR\_PIN, INPUT);

pinMode(MODE\_SWITCH\_PIN, INPUT);

// Determine mode based on switch position

if (digitalRead(MODE\_SWITCH\_PIN) == OBJECT\_REMOVING) {

startTime = millis();

}

}

void loop() {

if (digitalRead(MODE\_SWITCH\_PIN) == OBJECT\_REMOVING) {

removeObjects();

}

}

void removeObjects() {

unsigned long currentTime = millis();

if (currentTime - startTime >= TIME\_LIMIT) {

stopMotors();

return;

}

// Read distance sensor

int distance = analogRead(DISTANCE\_SENSOR\_PIN);

if (distance < 200) { // Example threshold

pushObject();

} else {

searchForObject();

}

}

void moveForward() {

digitalWrite(LEFT\_MOTOR\_PIN1, HIGH);

digitalWrite(LEFT\_MOTOR\_PIN2, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN1, HIGH);

digitalWrite(RIGHT\_MOTOR\_PIN2, LOW);

}

void turnRight() {

digitalWrite(LEFT\_MOTOR\_PIN1, HIGH);

digitalWrite(LEFT\_MOTOR\_PIN2, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN1, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN2, HIGH);

}

void stopMotors() {

digitalWrite(LEFT\_MOTOR\_PIN1, LOW);

digitalWrite(LEFT\_MOTOR\_PIN2, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN1, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN2, LOW);

}

void pushObject() {

moveForward();

delay(1000); // Push for 1 second

stopMotors();

delay(500); // Wait for a moment

}

void searchForObject() {

turnRight();

delay(500); // Turn for 0.5 seconds

stopMotors();

delay(500); // Wait for a moment

}

Code for Line Following

#include <Arduino.h>

// Pin definitions

#define LEFT\_MOTOR\_PIN1 2

#define LEFT\_MOTOR\_PIN2 3

#define RIGHT\_MOTOR\_PIN1 4

#define RIGHT\_MOTOR\_PIN2 5

#define LEFT\_SENSOR\_PIN A0

#define MIDDLE\_SENSOR\_PIN A1

#define RIGHT\_SENSOR\_PIN A2

#define MODE\_SWITCH\_PIN 6

// Constants

#define LINE\_FOLLOWING 0

void setup() {

pinMode(LEFT\_MOTOR\_PIN1, OUTPUT);

pinMode(LEFT\_MOTOR\_PIN2, OUTPUT);

pinMode(RIGHT\_MOTOR\_PIN1, OUTPUT);

pinMode(RIGHT\_MOTOR\_PIN2, OUTPUT);

pinMode(LEFT\_SENSOR\_PIN, INPUT);

pinMode(MIDDLE\_SENSOR\_PIN, INPUT);

pinMode(RIGHT\_SENSOR\_PIN, INPUT);

pinMode(MODE\_SWITCH\_PIN, INPUT);

}

void loop() {

if (digitalRead(MODE\_SWITCH\_PIN) == LINE\_FOLLOWING) {

followLine();

}

}

void followLine() {

int leftSensor = analogRead(LEFT\_SENSOR\_PIN);

int middleSensor = analogRead(MIDDLE\_SENSOR\_PIN);

int rightSensor = analogRead(RIGHT\_SENSOR\_PIN);

if (middleSensor < 500) { // Black line detected

moveForward();

} else if (leftSensor < 500) { // Line detected on left

turnLeft();

} else if (rightSensor < 500) { // Line detected on right

turnRight();

} else {

stopMotors(); // Line lost

}

}

void moveForward() {

digitalWrite(LEFT\_MOTOR\_PIN1, HIGH);

digitalWrite(LEFT\_MOTOR\_PIN2, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN1, HIGH);

digitalWrite(RIGHT\_MOTOR\_PIN2, LOW);

}

void turnLeft() {

digitalWrite(LEFT\_MOTOR\_PIN1, LOW);

digitalWrite(LEFT\_MOTOR\_PIN2, HIGH);

digitalWrite(RIGHT\_MOTOR\_PIN1, HIGH);

digitalWrite(RIGHT\_MOTOR\_PIN2, LOW);

}

void turnRight() {

digitalWrite(LEFT\_MOTOR\_PIN1, HIGH);

digitalWrite(LEFT\_MOTOR\_PIN2, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN1, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN2, HIGH);

}

void stopMotors() {

digitalWrite(LEFT\_MOTOR\_PIN1, LOW);

digitalWrite(LEFT\_MOTOR\_PIN2, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN1, LOW);

digitalWrite(RIGHT\_MOTOR\_PIN2, LOW);

}

Switch Between Modes

void loop() {

if (digitalRead(MODE\_SWITCH\_PIN) == OBJECT\_REMOVING) {

removeObjects();

} else if (digitalRead(MODE\_SWITCH\_PIN) == LINE\_FOLLOWING) {

followLine();

}

}