**Hardware code:**

**// Pin definitions**

**const int trigPin = 12; // Trigger pin for HC-SR04**

**const int echoPin = 13; // Echo pin for HC-SR04**

**const int motor1Pin1 = 7; // IN1 for Motor 1**

**const int motor1Pin2 = 8; // IN2 for Motor 1**

**const int motor2Pin1 = 5; // IN3 for Motor 2**

**const int motor2Pin2 = 6; // IN4 for Motor 2**

**const float obstacleDistance = 20.0; // Distance in cm to detect obstacle**

**long duration; // Duration of the ultrasonic signal**

**float distance; // Measured distance**

**unsigned long startTime; // Time tracking for forward movement**

**unsigned long moveDuration; // Total forward movement duration**

**void setup() {**

**// Initialize pins**

**pinMode(trigPin, OUTPUT);**

**pinMode(echoPin, INPUT);**

**pinMode(motor1Pin1, OUTPUT);**

**pinMode(motor1Pin2, OUTPUT);**

**pinMode(motor2Pin1, OUTPUT);**

**pinMode(motor2Pin2, OUTPUT);**

**// Initialize serial communication**

**Serial.begin(9600);**

**// Ensure motors are off at the start**

**stopMotors();**

**}**

**void loop() {**

**// Move forward until an obstacle is detected**

**moveForward();**

**startTime = millis(); // Record start time of movement**

**while (true) {**

**distance = measureDistance();**

**if (distance > 0 && distance <= obstacleDistance) {**

**// Obstacle detected: Stop and calculate movement time**

**stopMotors();**

**delay(500); // Short pause**

**moveDuration = millis() - startTime; // Calculate forward movement time**

**// Reverse back to the starting position**

**moveBackward(moveDuration);**

**stopMotors();**

**delay(500); // Short pause**

**// Break the loop and restart the process**

**break;**

**}**

**}**

**}**

**// Function to measure distance using HC-SR04**

**float measureDistance() {**

**digitalWrite(trigPin, LOW);**

**delayMicroseconds(2);**

**digitalWrite(trigPin, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(trigPin, LOW);**

**duration = pulseIn(echoPin, HIGH);**

**return (duration \* 0.034) / 2; // Convert to distance in cm**

**}**

**// Function to stop motors**

**void stopMotors() {**

**digitalWrite(motor1Pin1, LOW);**

**digitalWrite(motor1Pin2, LOW);**

**digitalWrite(motor2Pin1, LOW);**

**digitalWrite(motor2Pin2, LOW);**

**Serial.println("Motors stopped");**

**}**

**// Function to move forward**

**void moveForward() {**

**digitalWrite(motor1Pin1, HIGH);**

**digitalWrite(motor1Pin2, LOW);**

**digitalWrite(motor2Pin1, HIGH);**

**digitalWrite(motor2Pin2, LOW);**

**Serial.println("Moving forward");**

**}**

**// Function to move backward for a specific duration**

**void moveBackward(unsigned long duration) {**

**Serial.println("Moving backward");**

**digitalWrite(motor1Pin1, LOW);**

**digitalWrite(motor1Pin2, HIGH);**

**digitalWrite(motor2Pin1, LOW);**

**digitalWrite(motor2Pin2, HIGH);**

**delay(duration); // Move backward for the same time as forward**

**}**