

```

#include "arduino_secrets.h"

/* Smart Cane - Object & Gap Detection */

const int trigPin = 6;
const int echoPin = 7;
const int vibrationPin = 8;

int thresholdObstacle = 20; // Distance for obstacle detection
//int thresholdGap = 50;      // Distance for gap detection *I would
//need a second ultrasonic sensor
int stableLower = 21;        // Lower bound for stable zone
int stableUpper = 49;        // Upper bound for stable zone

// Vibration delay settings
int vibrationDelayMin = 10;
int vibrationDelayMax = 500;

void setup() {
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(vibrationPin, OUTPUT);
}

// Function to measure distance
float getDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    return duration * 0.034 / 2; // Convert microseconds to cm
}

// Function to filter noise by averaging multiple readings
float getFilteredDistance(int numReadings) {
    float sum = 0;
    for (int i = 0; i < numReadings; i++) {
        sum += getDistance();
        delay(10);
    }
    return sum / numReadings; // Return averaged distance
}

void loop() {
    float distance = getFilteredDistance(5); // Take 5 filtered readings

    // Obstacle Detection
    if (distance < thresholdObstacle) {
        int vibrationDelay = map(distance, 0, thresholdObstacle,
vibrationDelayMin, vibrationDelayMax);
    }
}

```

```

        digitalWrite(vibrationPin, HIGH);
        delay(vibrationDelay);
        digitalWrite(vibrationPin, LOW);
        delay(vibrationDelay);
    }
    /* In need of a second ultrasonic sensor
    // Gap Detection
    else if (distance > thresholdGap) {
        digitalWrite(vibrationPin, HIGH);
        delay(500); // Longer vibration for gaps
        digitalWrite(vibrationPin, LOW);

    }
    */
    // Safe Zone (No obstacles or gaps)
    else if (distance <= stableUpper) {
        digitalWrite(vibrationPin, LOW); // No vibration
    }

    // Print distance for debugging
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");

    delay(500);
}

```