Codes finaux

Code Arduino

```
const int PIEZO1_PIN = A0; // Sortie du piezo 1
const int PIEZO2_PIN = A1; // Sortie du piezo 2
const int PIEZO3_PIN = A2; // Sortie du piezo 3
float previousPiezo1ADC = 0; // Stocker la valeur précédente
float previousPiezo2ADC = 0; // Stocker la valeur précédente
float previousPiezo3ADC = 0; // Stocker la valeur précédente

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Lecture de la valeur ADC actuelle du piezo
  float piezo1ADC = analogRead(PIEZO1_PIN);
  float piezo2ADC = analogRead(PIEZO2_PIN);
  float piezo3ADC = analogRead(PIEZO3_PIN);

  // Calculer la différence entre la valeur actuelle et la précédente
  float diff1 = piezo1ADC - previousPiezo1ADC;
  float diff2 = piezo2ADC - previousPiezo2ADC;
  float diff3 = piezo3ADC - previousPiezo3ADC;

  // Mettre à jour la valeur précédente pour la prochaine itération
  previousPiezo1ADC = piezo1ADC;
  previousPiezo2ADC = piezo2ADC;
  previousPiezo3ADC = piezo3ADC;

  // Affichage de la différence
  Serial.print(diff1); Serial.print(",");
  Serial.print(diff2); Serial.print(",");
  Serial.print(diff3); Serial.print(",");

  Serial.print(-20); Serial.print(","); // To freeze the lower limit
  Serial.print(20); Serial.print(","); // To freeze the upper limit
  Serial.println();

  // Attente avant la prochaine lecture
  delay(5);
```

```
}
```

Code pour le support cubic

```
// Valeurs en mm
Eb = 6;
Es = 3;
L = 75;
R = 2.5;
Et = 100;

// Base

union(){
    cube([L, L, Eb]);
    cube([6, 6, 69]);
    translate([69, 69, 69])cube([6, 6, 6]);
    translate([69, 0, 69])rotate([0, 90, 0])difference(){
        cube([63, L, Es]);
        translate([11, 15, 0]) cylinder(Et, R, R, true);
        translate([51, 15, 0]) cylinder(Et, R, R, true);
        translate([11, 60, 0]) cylinder(Et, R, R, true);
        translate([51, 60, 0]) cylinder(Et, R, R, true);
        translate([15.5, 45, -50]) cube([30, 35, Et]);
        translate([20.5, 11, -50]) cube([20, 40, Et]);
        translate([15.5, 11, 2.5]) cube([5, 40, 2]);
        translate([40.5, 11, 2.5]) cube([5, 40, 2]);
    }
    translate([69, 69, 6])rotate([90, 0, 180])difference(){
        cube([69, 69, Es]);
        translate([11, 15, 0]) cylinder(Et, R, R, true);
        translate([56, 15, 0]) cylinder(Et, R, R, true);
        translate([11, 54, 0]) cylinder(Et, R, R, true);
        translate([56, 54, 0]) cylinder(Et, R, R, true);
        translate([17.5, 39, -50]) cube([30, 30, Et]);
        translate([22.5, 5, -50]) cube([20, 40, Et]);
        translate([17.5, 5, 2.5]) cube([5, 40, 2]);
        translate([42.5, 5, 2.5]) cube([5, 40, 2]);
    }
    translate([0, 69, 69])rotate([0, 0, 270])difference(){
        cube([69, 75, Es]);
        translate([11, 15, 0]) cylinder(Et, R, R, true);
        translate([51, 15, 0]) cylinder(Et, R, R, true);
        translate([11, 60, 0]) cylinder(Et, R, R, true);
```

```
        translate([51, 60, 0]) cylinder(Et, R, R, true);
        translate([15.5, 45, -50]) cube([30, 35, Et]);
        translate([20.5, 11, -50]) cube([20, 40, Et]);
        translate([15.5, 11, 2.5]) cube([5, 40, 2]);
        translate([40.5, 11, 2.5]) cube([5, 40, 2]);
    }
}

rotate([0, 180, 0])translate([-140, 0, -3])difference(){
    cube([63, L, Es]);
    translate([11, 15, 0]) cylinder(Et, R, R, true);
    translate([51, 15, 0]) cylinder(Et, R, R, true);
    translate([11, 60, 0]) cylinder(Et, R, R, true);
    translate([51, 60, 0]) cylinder(Et, R, R, true);
    translate([15.5, 45, -50]) cube([30, 30, Et]);
    translate([20.5, 11, -50]) cube([20, 40, Et]);
    translate([15.5, 11, -1.5]) cube([5, 40, 2]);
    translate([40.5, 11, -1.5]) cube([5, 40, 2]);
}

rotate([0, 180, 0])translate([-69, 80, -3])difference(){
    cube([69, 69, Es]);
    translate([11, 15, 0]) cylinder(Et, R, R, true);
    translate([56, 15, 0]) cylinder(Et, R, R, true);
    translate([11, 54, 0]) cylinder(Et, R, R, true);
    translate([56, 54, 0]) cylinder(Et, R, R, true);
    translate([17.5, 39, -50]) cube([30, 30, Et]);
    translate([22.5, 5, -50]) cube([20, 40, Et]);
    translate([17.5, 5, -1.5]) cube([5, 40, 2]);
    translate([42.5, 5, -1.5]) cube([5, 40, 2]);
}

rotate([0, 180, 0])translate([-146, 80, -3])difference(){
    cube([69, 75, Es]);
    translate([11, 15, 0]) cylinder(Et, R, R, true);
    translate([51, 15, 0]) cylinder(Et, R, R, true);
    translate([11, 60, 0]) cylinder(Et, R, R, true);
    translate([51, 60, 0]) cylinder(Et, R, R, true);
    translate([15.5, 45, -50]) cube([30, 30, Et]);
    translate([20.5, 11, -50]) cube([20, 40, Et]);
    translate([15.5, 11, -1.5]) cube([5, 40, 2]);
    translate([40.5, 11, -1.5]) cube([5, 40, 2]);
}
```

Revision #1
Created 28 May 2024 07:24:25 by Pierre Thery
Updated 28 May 2024 07:24:26 by Pierre Thery