```
#include <Keyboard.h>
#include <Mouse.h>

// === SIP & PUFF SETTINGS ===
const int pressurePin1 = A2;
const int pressurePin2 = A3;
const float VCC = 5.0;
const float sensitivity = 0.009;
float baselineVoltage1 = 0.0;
float baselineVoltage2 = 0.0;
float suctionThreshold_kPa = -0.5;
float blowThreshold_kPa = 10.5;
const float noiseThreshold_kPa = 0.3;  // Dead zone to avoid false triggers
unsigned long lastPressureTime1 = 0;
unsigned long lastPressureTime2 = 0;
const unsigned long debounceDelay = 300;

// === JOYSTICK SETTINGS ===
const int xPin = A0;
const int yPin = A1;
const int buttonPin = 2;
const int threshold = 40;
const int maxMove = 15;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(pressurePin1, INPUT);
  pinMode(pressurePin2, INPUT);
  Keyboard.begin();
  Mouse.begin();
  Serial.begin(9600);

  // Baseline calibration for both pressure sensors
  float total1 = 0, total2 = 0;
  for (int i = 0; i < 10; i++) {
    total1 += analogRead(pressurePin1) * (VCC / 1023.0);
    total2 += analogRead(pressurePin2) * (VCC / 1023.0);
    delay(10);
  }
  baselineVoltage1 = total1 / 10;
  baselineVoltage2 = total2 / 10;
  Serial.print("Baseline Voltage 1: ");
  Serial.println(baselineVoltage1, 3);
  Serial.print("Baseline Voltage 2: ");
```

```
  Serial.println(baselineVoltage2, 3);
}

void loop() {
  // === JOYSTICK LOGIC ===
  int xVal = analogRead(xPin);
  int yVal = analogRead(yPin);
  int btn = digitalRead(buttonPin);

  int xMove = 0;
  int yMove = 0;

  if (abs(xVal - 512) > threshold) {
    xMove = map(xVal, 0, 1023, -maxMove, maxMove);
  }
  if (abs(yVal - 512) > threshold) {
    yMove = map(yVal, 0, 1023, -maxMove, maxMove);
  }

  if (xMove != 0 || yMove != 0) {
    Mouse.move(xMove, yMove);
  }

  if (btn == LOW) {
    Mouse.press(MOUSE_LEFT);
  } else {
    Mouse.release(MOUSE_LEFT);
  }

  // === SIP & PUFF LOGIC FOR SENSOR 1 ===
  float sensorVoltage1 = 0;
  for (int i = 0; i < 5; i++) {
    sensorVoltage1 += analogRead(pressurePin1) * (VCC / 1023.0);
    delay(5);
  }
  sensorVoltage1 /= 5;

  float pressure_kPa1 = (sensorVoltage1 - baselineVoltage1) / sensitivity;
  if (pressure_kPa1 > blowThreshold_kPa) {
    if (millis() - lastPressureTime1 > debounceDelay) {
      Keyboard.write('g');  // puff 1
      lastPressureTime1 = millis();
    }
  } else if (pressure_kPa1 < suctionThreshold_kPa) {
```

```
    if (millis() - lastPressureTime1 > debounceDelay) {
      Keyboard.write('e');  // sip 1
      lastPressureTime1 = millis();
    }
  }

  // === SIP & PUFF LOGIC FOR SENSOR 2 ===
  float sensorVoltage2 = 0;
  for (int i = 0; i < 5; i++) {
    sensorVoltage2 += analogRead(pressurePin2) * (VCC / 1023.0);
    delay(5);
  }
  sensorVoltage2 /= 5;

  float pressure_kPa2 = (sensorVoltage2 - baselineVoltage2) / sensitivity;
  if (pressure_kPa2 > blowThreshold_kPa) {
    if (millis() - lastPressureTime2 > debounceDelay) {
      Keyboard.write('h');  // puff 2
      lastPressureTime2 = millis();
    }
  } else if (pressure_kPa2 < suctionThreshold_kPa) {
    if (millis() - lastPressureTime2 > debounceDelay) {
      Keyboard.write('f');  // sip 2
      lastPressureTime2 = millis();
    }
  }

  delay(10);
}
```