```
#include <ESP32Servo.h>
#include <SoftwareSerial.h>

#define LED 27  // led pin
#define BUT 15  // button pin
#define pot A3  // potentiometer pin

int counter = 0;
int ventilateTime = 7; // how many seconds the window should be opened
bool windowOpened = false;

Servo servo;
SoftwareSerial OpenLCD(0, 21); // TX (not used), RX pins

void setup() {
  // init pins
  pinMode(BUT, INPUT);

  // define the pereiod (50Hz = standard)
  servo.setPeriodHertz(50);
  // attach the servo to pin 33,
  // tell it that minimum pulsewidth is 700us (0°)
  // and that the maximum is 2500us (180°)
  //servo.attach(33, 700, 1600); //1600us (90°)
  servo.attach(33);   // servo pin
  Serial.begin(9600);
  OpenLCD.begin(9600);

  servo.write(0); //initially window closed

  setContrast(2);
  clearScreen();
  OpenLCD.print("CO2: ");
  OpenLCD.write(254);
  OpenLCD.write(128 + 0 + 10);
  OpenLCD.print("ppm");

  //changeBackgroundColor(0, 255, 0);
}

void loop() {
  int tmp = analogRead(pot); // using potentiometer for testing purposes
  int ppm = map(tmp, 0, 4095, 500, 1500); // convert range of values from potentiometer (0 - 4095)to ppm (500 - 1500)
  Serial.println(ppm);

  if (ppm >= 1000 ) { // from 1000 ppm: move servo to open window
    int pos = 0;
    if (windowOpened == false) {
      windowOpened = true;

      servo.write(180);
      changeBackgroundColor(255, 0, 0);

      clearScreen();
      OpenLCD.print("CO2: ");
      OpenLCD.write(254);
      OpenLCD.write(128 + 0 + 10);
      OpenLCD.print("ppm");
```

```
    }

    if (counter == ventilateTime) { // close window after ventilateTime seconds
      servo.write(0);
      counter = 0;
      delay(5000);
    }
    else {
      counter++;
    }

    //Serial.println(counter);

    OpenLCD.write(254);
    OpenLCD.write(128 + 0 + 5);
    OpenLCD.print(ppm);


  }

  else if (ppm < 1000 ) {
    if (windowOpened == true) {
      windowOpened = false;
      servo.write(0);
      changeBackgroundColor(0, 255, 0);

      clearScreen();
      OpenLCD.print("CO2: ");
      OpenLCD.write(254);
      OpenLCD.write(128 + 0 + 10);
      OpenLCD.print("ppm");
    }

    counter = 0;

    OpenLCD.write(254);
    OpenLCD.write(128 + 0 + 5);
    OpenLCD.print("    ");
    OpenLCD.write(254);
    OpenLCD.write(128 + 0 + 5);
    OpenLCD.print(ppm);
  }

  delay (1000);
}

void clearScreen() {
  OpenLCD.write('|'); // Beginning of a new command (Setting mode)
  OpenLCD.write('-');  // Clear screen and reset cursor position (1. Këscht)
}

void setContrast(int contrastValue) {
  OpenLCD.write('|'); // Beginning of a new command (Setting mode)
  OpenLCD.write(24);  // Send contrast command
  OpenLCD.write(contrastValue);   // Set contrast value [0;255]
}

void changeBackgroundColor(int r, int g, int b) {
  // change range of values to be between 0 an 29
  r = map (r, 0, 255, 0, 29);
```

```
  g = map (g, 0, 255, 0, 29);
  b = map (b, 0, 255, 0, 29);

  // control backlight
  OpenLCD.write('|');
  OpenLCD.write(128 + r); // set white/red backlight amount to 0%

  OpenLCD.write('|');
  OpenLCD.write(158 + g); //set green backlight amount to 0%

  OpenLCD.write('|');
  OpenLCD.write(188 + b); //set bluebacklight amount to 0%
}

```