

```
1 #include <ESP32Servo.h>
2 #include <SoftwareSerial.h>
3
4 #define LED 27 // led pin
5 #define BUT 15 // button pin
6 #define pot A3 // potentiometer pin
7
8 int counter = 0;
9 int ventilateTime = 7; // how many seconds the window should be opened
10 bool windowOpened = false;
11
12 Servo servo;
13 SoftwareSerial OpenLCD(0, 21); // TX (not used), RX pins
14
15 void setup() {
16     // init pins
17     pinMode(BUT, INPUT);
18
19     // define the pereiod (50Hz = standard)
20     servo.setPeriodHertz(50);
21     // attach the servo to pin 33,
22     // tell it that minimum pulsewidth is 700us (0°)
23     // and that the maximum is 2500us (180°)
24     //servo.attach(33, 700, 1600); //1600us (90°)
25     servo.attach(33); // servo pin
26     Serial.begin(9600);
27     OpenLCD.begin(9600);
28
29     servo.write(0); //initially window closed
30
31     setContrast(2);
32     clearScreen();
33     OpenLCD.print("CO2: ");
34     OpenLCD.write(254);
35     OpenLCD.write(128 + 0 + 10);
36     OpenLCD.print("ppm");
37
38     //changeBackgroundColor(0, 255, 0);
39 }
40
41 void loop() {
42     int tmp = analogRead(pot); // using potentiometer for testing purposes
43     int ppm = map(tmp, 0, 4095, 500, 1500); // convert range of values from
44     //potentiometer (0 - 4095)to ppm (500 - 1500)
45     Serial.println(ppm);
46
47     if (ppm >= 1000 ) { // from 1000 ppm: move servo to open window
48         int pos = 0;
49         if (windowOpened == false) {
50             windowOpened = true;
51
52             servo.write(180);
53             changeBackgroundColor(255, 0, 0);
54
55             clearScreen();
56             OpenLCD.print("CO2: ");
57             OpenLCD.write(254);
58             OpenLCD.write(128 + 0 + 10);
59             OpenLCD.print("ppm");
```

```
59     }
60
61     if (counter == ventilateTime) { // close window after ventilateTime seconds
62         servo.write(0);
63         counter = 0;
64         delay(5000);
65     }
66     else {
67         counter++;
68     }
69
70     //Serial.println(counter);
71
72     OpenLCD.write(254);
73     OpenLCD.write(128 + 0 + 5);
74     OpenLCD.print(ppm);
75
76
77 }
78
79 else if (ppm < 1000 ) {
80     if (windowOpened == true) {
81         windowOpened = false;
82         servo.write(0);
83         changeBackgroundColor(0, 255, 0);
84
85         clearScreen();
86         OpenLCD.print("CO2: ");
87         OpenLCD.write(254);
88         OpenLCD.write(128 + 0 + 10);
89         OpenLCD.print("ppm");
90     }
91
92     counter = 0;
93
94     OpenLCD.write(254);
95     OpenLCD.write(128 + 0 + 5);
96     OpenLCD.print(" ");
97     OpenLCD.write(254);
98     OpenLCD.write(128 + 0 + 5);
99     OpenLCD.print(ppm);
100 }
101
102 delay (1000);
103 }
104
105 void clearScreen() {
106     OpenLCD.write('|'); // Beginning of a new command (Setting mode)
107     OpenLCD.write('-'); // Clear screen and reset cursor position (1. Këscht)
108 }
109
110 void setContrast(int contrastValue) {
111     OpenLCD.write('|'); // Beginning of a new command (Setting mode)
112     OpenLCD.write(24); // Send contrast command
113     OpenLCD.write(contrastValue); // Set contrast value [0;255]
114 }
115
116 void changeBackgroundColor(int r, int g, int b) {
117     // change range of values to be between 0 an 29
118     r = map (r, 0, 255, 0, 29);
```

```
119 g = map (g, 0, 255, 0, 29);
120 b = map (b, 0, 255, 0, 29);
121
122 // control backlight
123 OpenLCD.write('|');
124 OpenLCD.write(128 + r); // set white/red backlight amount to 0%
125
126 OpenLCD.write('|');
127 OpenLCD.write(158 + g); //set green backlight amount to 0%
128
129 OpenLCD.write('|');
130 OpenLCD.write(188 + b); //set bluebacklight amount to 0%
131 }
132
```