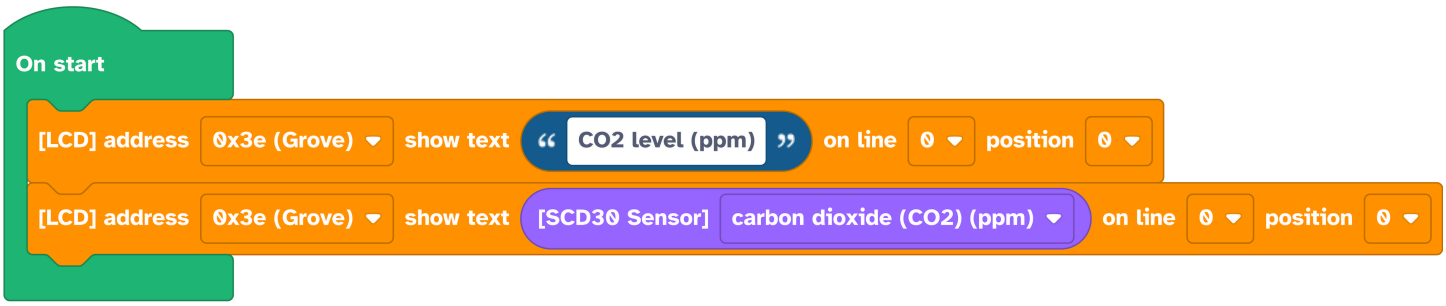
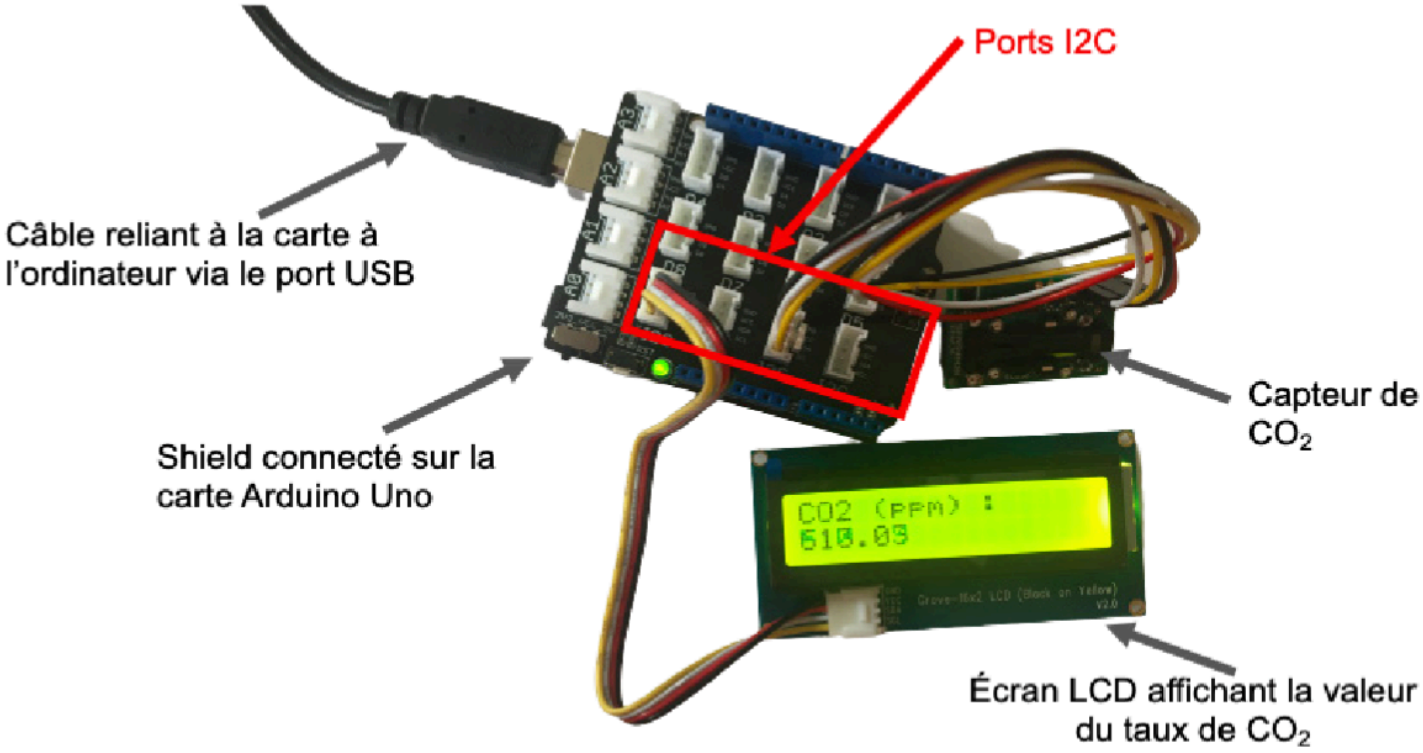
Datasheet - Programming & Sensors

Measure indoor air quality

1 Display CO2 level on LCD screen using SCD30 sensor

Editor used: vittascience.com/l476; vittascience.com/Arduino or vittascience.com/microbit





Code to copy into the editor

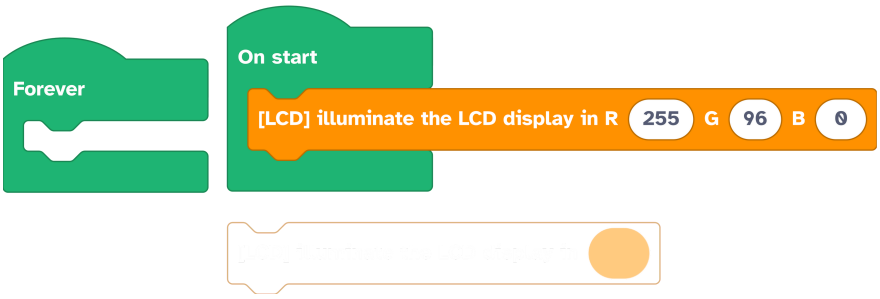
```
#include <Wire.h>
#include <rgb_lcd.h>
#include <SCD30.h>

rgb_lcd lcdRgb;

float t_scd;
float scd30 co2 = 0;
```

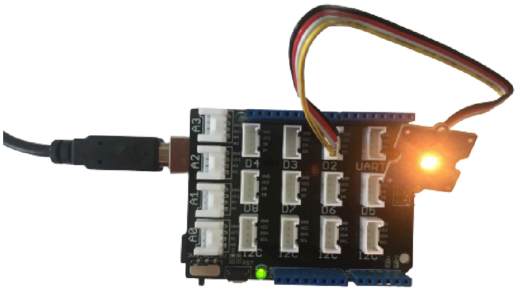
```
float scd30_t = 0;
float scd30_h = 0;

void serial_setupConnection(int baudrate) {
     Serial.begin(baudrate);
     while (!Serial) {
     Serial.println("En attente de l'ouverture du port série...");
        delay(1000);
     }
     Serial.println("Port série activé. Baudrate: " + String(baudrate));
     delay(50);
}

float scd30_read(uint8_t dataSelect) {
     t_scd = millis() - t_scd; if (t_scd > 1000 && scd30.isAvailable()) {
        float result[3] = {0};
        scd30.getCarbonDioxideConcentration(result);
        scd30_co2 = result[0];
        scd30_t = result[1];
        scd30_h = result[2];
     }
     switch (dataSelect) {
        case 0: return scd30_co2;
        case 1: return scd30_t;
        case 2: return scd30_h;
     }
}

void setup() {
     lcdRgb.begin(16, 2);
     serial_setupConnection(9600);
     Wire.begin();
     scd30.initialize();
     t_scd = millis();
     lcdRgb.setCursor(0, 0);
     lcdRgb.print(String("CO2 level (ppm)"));
     lcdRgb.setCursor(0, 0);
     lcdRgb.print(String(scd30_read(0)));
}

void loop() {}
```

2 LED color configuration

Editor used: vittascience.com/l476 ; vittascience.com/Arduino or vittascience.com/microbit

Block Overview





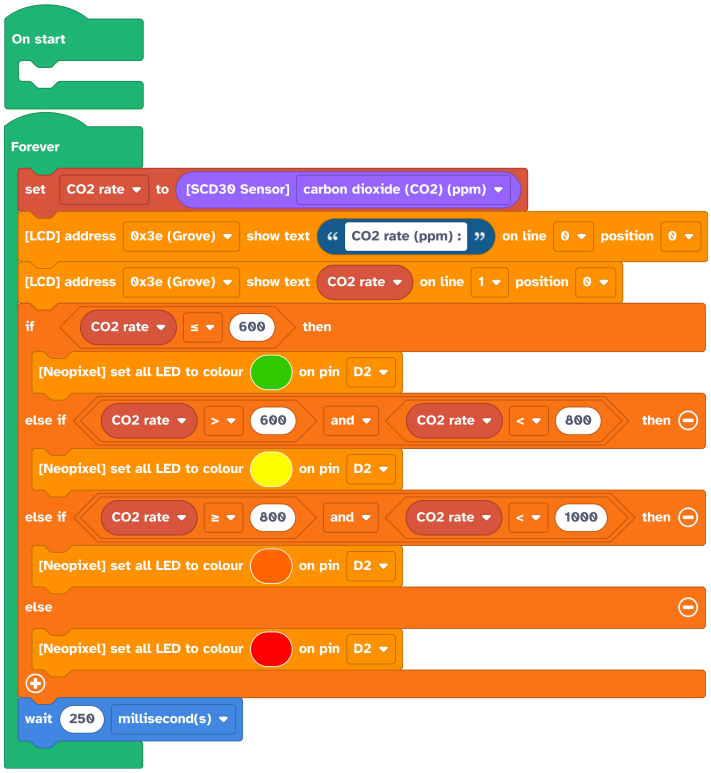
Code to copy into the editor

```
#include <Wire.h>
#include <rgb_lcd.h>

rgb_lcd lcdRgb;

void setup() {
    lcdRgb.begin(16, 2);
    lcdRgb.setRGB(255, 96, 0);
}

void loop() { }
```

3 CO2 LED indicator

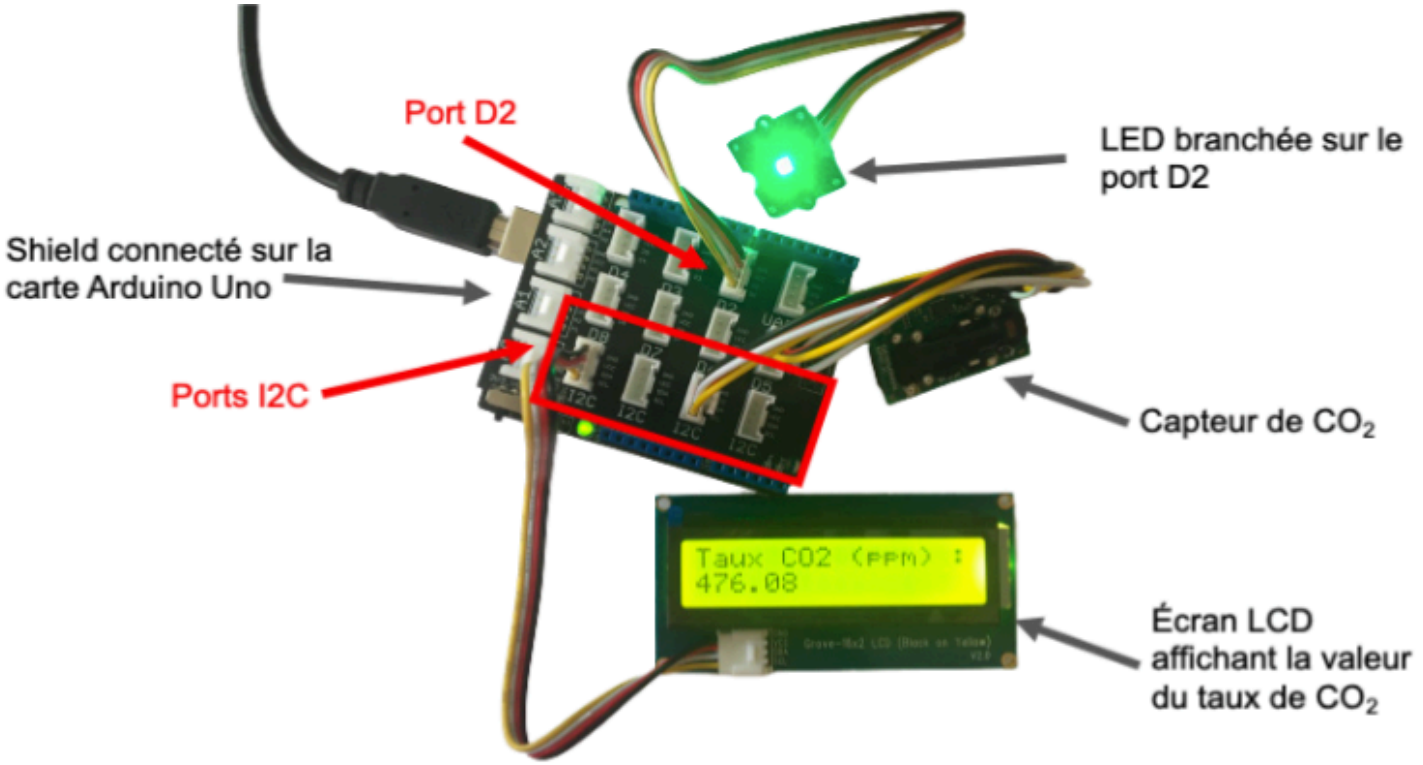
Editor used: vittascience.com/l476 ; vittascience.com/Arduino or vittascience.com/microbit

In the program, we then need to add "conditional structure" blocks. These blocks have a nomenclature such as: if this condition is met, then execute an instruction, otherwise execute another instruction. In this case, there are four conditions to anticipate, because there are four CO2 rate intervals. This program combines the programs created in steps 1 and 2, with the addition of a conditional structure block.



In this program, the carbon dioxide value is compared to different thresholds. To avoid having to take multiple measurements of the rate, which will then be compared to the different thresholds, it is possible to store the measured value in a variable. We suggest naming this variable "CO2 Rate." To create this variable, click on the Variables section.

A side panel opens: click "Create a variable". Name the variable "CO2 rate" for example, then click OK. Blocks specific to this new variable are created and accessible from the Variables section. To use this variable, click the Variables section.



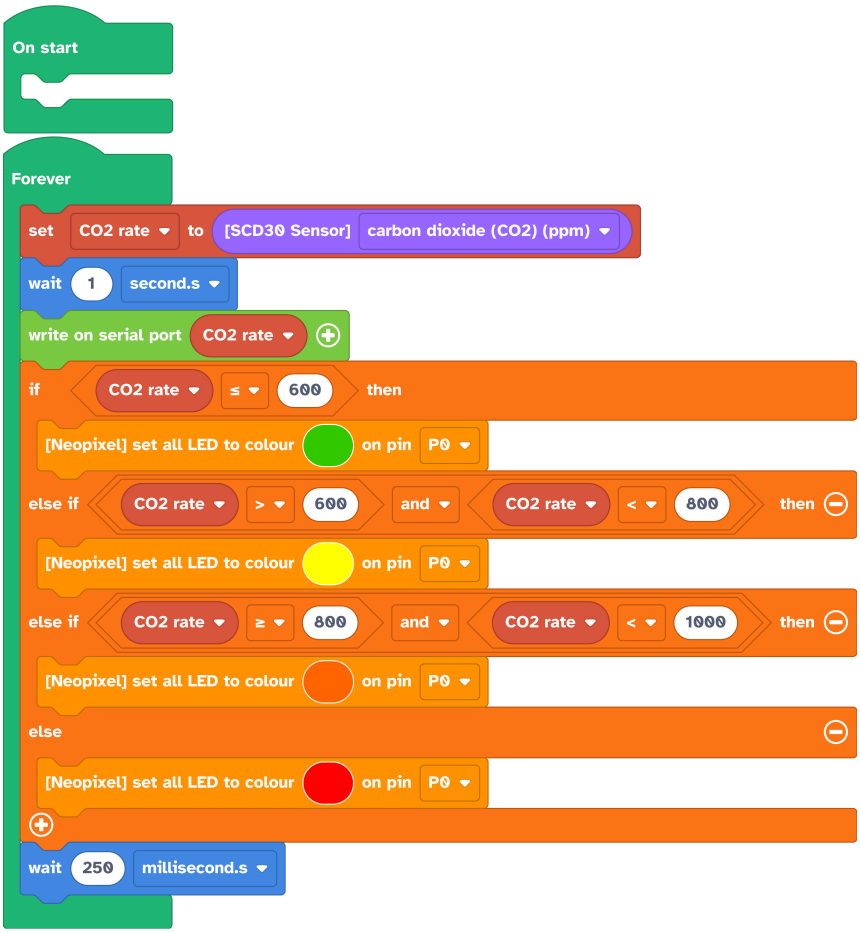
Code to copy into the editor

```
#include <Wire.h>
#include <SCD30.h>
#include <rgb_lcd.h>
#include <Adafruit_NeoPixel.h>

#define NP_LED_COUNT_2 30

rgb_lcd lcdRgb;
Adafruit_NeoPixel Neopixel_2(NP_LED_COUNT_2, 2, NEO_GRB + NEO_KHZ800);

float t_scd;
float scd30_co2 = 0;
float scd30_t = 0;
float scd30_h = 0;
float CO2_rate;

void serial_setupConnection(int baudrate) {
     Serial.begin(baudrate);
     while (!Serial) {
         Serial.println("En attente de l'ouverture du port série...");
         delay(1000);
     }
     Serial.println("Port série activé. Baudrate: " + String(baudrate));
     delay(50);
}

float scd30_read(uint8_t dataSelect) {
     t_scd = millis() - t_scd; if (t_scd > 1000 && scd30.isAvailable()) {
         float result[3] = {0};
         scd30.getCarbonDioxideConcentration(result);
         scd30_co2 = result[0];
         scd30_t = result[1];
         scd30_h = result[2];
     }
     switch (dataSelect) {
         case 0: return scd30_co2;
         case 1: return scd30_t;
         case 2: return scd30_h;
     }
}
```

```
void neopixel_showAllLed(Adafruit_NeoPixel *neoPx, uint8_t ledCount, uint8_t
r, uint8_t g, uint8_t b) {
     for (int i=0; i<ledCount; i++) {
         neoPx->setPixelColor(i, neoPx->Color(r, g, b));
     }
     neoPx->show();
}

void setup() {
     serial_setupConnection(9600);
     Wire.begin();
     scd30.initialize();
     t_scd = millis();
     lcdRgb.begin(16, 2);
     Neopixel_2.begin();
}

void loop() {
     CO2_rate = scd30_read(0);
     lcdRgb.setCursor(0, 0);
     lcdRgb.print(String("CO2 rate (ppm) :"));
     lcdRgb.setCursor(0, 1);
     lcdRgb.print(String(CO2_rate));
     if (CO2_rate <= 600) {
         neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
     }
     else if (CO2_rate > 600 && CO2_rate < 800) {
          neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 255, 0);
     }
     else if (CO2_rate >= 800 && CO2_rate < 1000) {
          neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 102, 0);
     }
     else {
         neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 0, 0);
     }
     delay(250);
}
```

4 CO2 display

Editor used: vittascience.com/l476 ; vittascience.com/Arduino or vittascience.com/microbit



Remember to add a one-second pause to limit the frequency with which the values are displayed on the computer screen. This makes it easier to read the measured values.

Code to copy into the editor

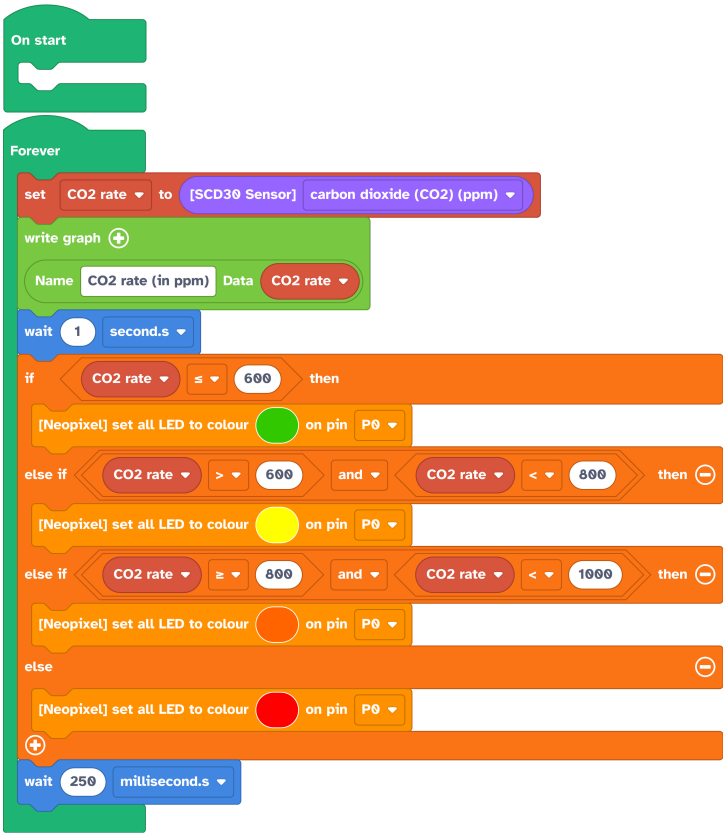
```
#include <Wire.h>
#include <SCD30.h>
#include <rgb_lcd.h>
#include <Adafruit_NeoPixel.h>

#define NP_LED_COUNT_2 30

rgb_lcd lcdRgb;
Adafruit_NeoPixel Neopixel_2(NP_LED_COUNT_2, 2, NEO_GRB + NEO_KHZ800);

float t_scd;
float scd30_co2 = 0;
float scd30_t = 0;
float scd30_h = 0;
float CO2_rate;
```

```
void serial_setupConnection(int baudrate) {
     Serial.begin(baudrate);
     while (!Serial) {
         Serial.println("En attente de l'ouverture du port série...");
         delay(1000);
     }
     Serial.println("Port série activé. Baudrate: " + String(baudrate));
     delay(50);
}

float scd30_read(uint8_t dataSelect) {
     t_scd = millis() - t_scd; if (t_scd > 1000 && scd30.isAvailable()) {
         float result[3] = {0};
         scd30.getCarbonDioxideConcentration(result);
         scd30_co2 = result[0];
         scd30_t = result[1];
         scd30_h = result[2];
     }
     switch (dataSelect) {
         case 0: return scd30_co2;
         case 1: return scd30_t;
         case 2: return scd30_h;
     }
}

void neopixel_showAllLed(Adafruit_NeoPixel *neoPx, uint8_t ledCount,
uint8_t r, uint8_t g, uint8_t b) { for (int i=0; i<ledCount; i++)
     {
         neoPx->setPixelColor(i, neoPx->Color(r, g, b));
     } neoPx->show();
}

void setup() {
     serial_setupConnection(9600);
     Wire.begin();
     scd30.initialize();
     t_scd = millis();
     lcdRgb.begin(16, 2);
     Neopixel_2.begin();
     CO2_rate = scd30_read(0);
     delay(1000*1);
     Serial.println(String(CO2_rate));
     lcdRgb.setCursor(0, 0);
```

```
    lcdRgb.print(String("CO2 rate (ppm)"));
    lcdRgb.setCursor(0, 1);
    lcdRgb.print(String(CO2_rate));
    if (CO2_rate <= 600) {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
    } else if (CO2_rate > 600 && CO2_rate < 800) {
         neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 255, 0);
    } else if (CO2_rate >= 800 && CO2_rate < 1000) {
         neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 102, 0);
    } else {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 0, 0); }
    delay(250); }

void loop() { }
```

5 Data visualization

Editor used: vittascience.com/l476 ; vittascience.com/Arduino or vittascience.com/microbit



1. To view the graph, in the display console, select "Graph Mode" on the right.
2. From this graphical representation, it is possible to export all the data as a .csv file (readable by spreadsheets such as Excel, Libre Office Calc, GoogleSheet, Numbers, etc.). To do this, simply click on the Export button at the bottom of the window. The data will then be accessible from the spreadsheet used by default on the computer. The "graphing" function of the spreadsheet allows you to draw a graph showing the evolution of carbon dioxide levels over time, which can then be printed.

Code to copy into the editor

```
#include <Wire.h>
#include <SCD30.h>
#include <rgb_lcd.h>
#include <Adafruit_NeoPixel.h>

#define NP_LED_COUNT_2 30

rgb_lcd lcdRgb;
Adafruit_NeoPixel Neopixel_2(NP_LED_COUNT_2, 2, NEO_GRB + NEO_KHZ800);

float t_scd;
float scd30_co2 = 0;
float scd30_t = 0;
float scd30_h = 0;
float CO2_rate;

void serial_setupConnection(int baudrate) {
```

```
    Serial.begin(baudrate);
    while (!Serial) {
        Serial.println("En attente de l'ouverture du port série...");
        delay(1000);
    }
    Serial.println("Port série activé. Baudrate: " + String(baudrate));
    delay(50);
}

float scd30_read(uint8_t dataSelect) {
    t_scd = millis() - t_scd;
    if (t_scd > 1000 && scd30.isAvailable()) {
        float result[3] = {0};
        scd30.getCarbonDioxideConcentration(result);
        scd30_co2 = result[0];
        scd30_t = result[1];
        scd30_h = result[2];
    }
    switch (dataSelect) {
        case 0: return scd30_co2;
        case 1: return scd30_t;
        case 2: return scd30_h;
    }
}

void neopixel_showAllLed(Adafruit_NeoPixel *neoPx, uint8_t ledCount, uint8_t
 r, uint8_t g, uint8_t b) {
    for (int i=0; i<ledCount; i++) {
        neoPx->setPixelColor(i, neoPx->Color(r, g, b));
    } neoPx->show();
}

void setup() {
    serial_setupConnection(9600);
    Wire.begin();
    scd30.initialize();
    t_scd = millis();
    lcdRgb.begin(16, 2);
    Neopixel_2.begin();
    CO2_rate = scd30_read(0);
    delay(1000*1);
    Serial.print("@Graph:");
    Serial.print("CO2 rate (in ppm):");
```

```
    Serial.print(CO2_rate); Serial.print("|");
    Serial.print("\n");
    delay(50);
    lcdRgb.setCursor(0, 0);
    lcdRgb.print(String("CO2 rate (ppm)"));
    lcdRgb.setCursor(0, 1);
    lcdRgb.print(String(CO2_rate));
    if (CO2_rate <= 600) {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
    }
    else if (CO2_rate > 600 && CO2_rate < 800) {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 255, 0);
    } else if (CO2_rate >= 800 && CO2_rate < 1000) {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 102, 0);
    } else {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 0, 0);
    }
    delay(250);
}

void loop() { }
```