

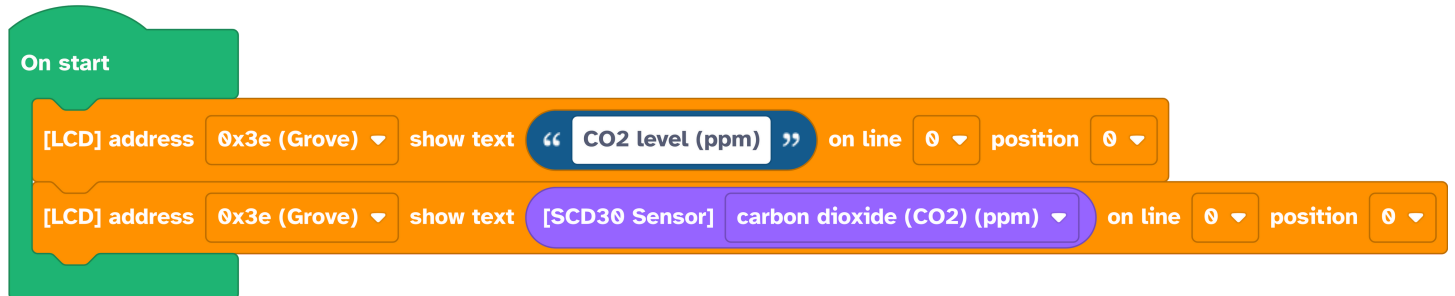


## Измерете качеството на въздуха в помещенията

1

### Показване на нивото на CO2 на LCD екрана с помощта на сензор SCD30

Използван редактор: [vittascience.com/l476](http://vittascience.com/l476); [vittascience.com/Arduino](http://vittascience.com/Arduino) или [vittascience.com/microbit](http://vittascience.com/microbit)



### Код за копиране в редактора

```
#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() {}

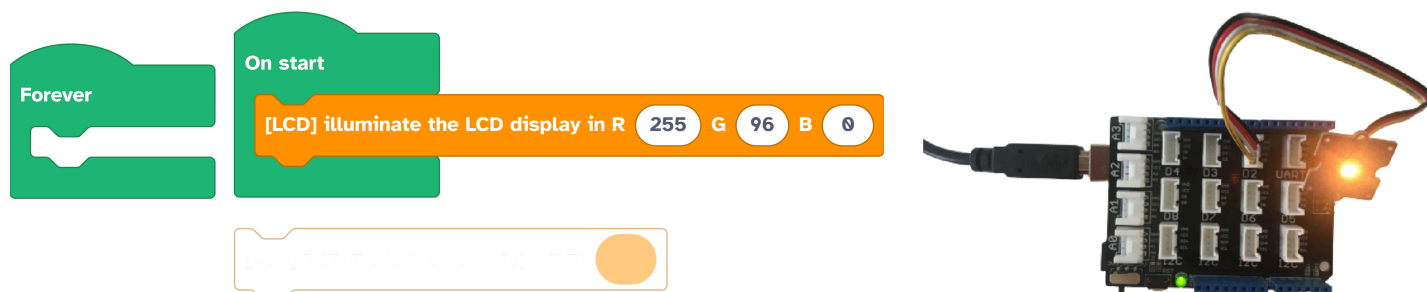
```

## 2

# Конфигурация на цветовете на светодиодите

Използван издател: [vittascience.com/l476](http://vittascience.com/l476); [vittascience.com/Arduino](http://vittascience.com/Arduino) или [vittascience.com/microbit](http://vittascience.com/microbit)

## Aperçu des blocs



## Код за копиране в редактора

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

rgb_lcd lcdRgb;

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

void loop() { }
```

## 3

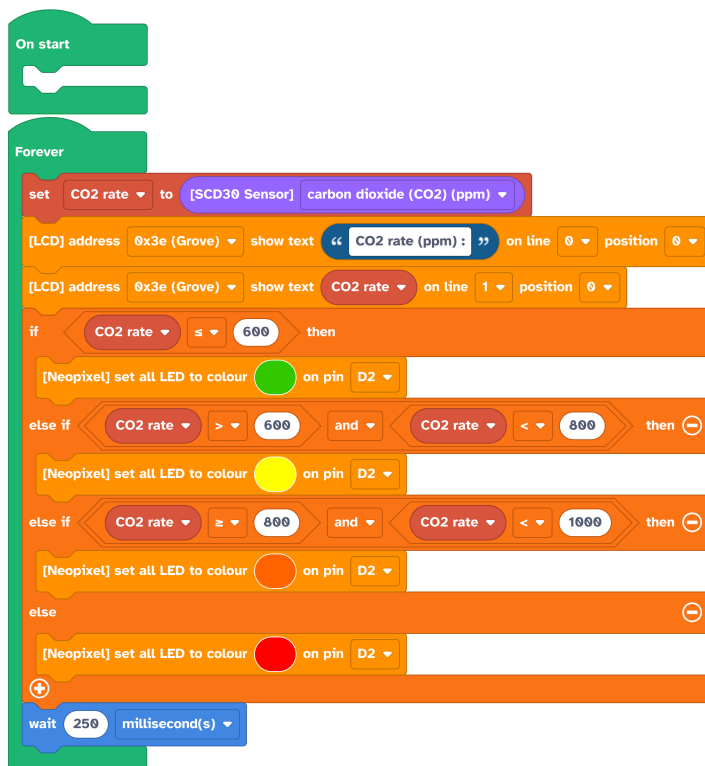
## LED индикатор за CO2

Използван издател: [vittascience.com/l476](http://vittascience.com/l476); [vittascience.com/Arduino](http://vittascience.com/Arduino) или [vittascience.com/microbit](http://vittascience.com/microbit)

В програмата след това трябва да добавим блокове с „условна структура“. Тези блокове имат номенклатура, като например: ако това условие е изпълнено, тогава се изпълнява инструкция, в противен случай се изпълнява друга инструкция. В този случай има четири условия, които трябва да се предвидят, защото има четири интервала на скорост на CO2. Тази програма комбинира програмите, създадени в стъпки 1 и 2, с добавяне на блок с условна структура.

В тази програма стойността на въглеродния диоксид се сравнява с различни прагове. За да се избегне необходимостта от множество измервания на скоростта, които след това ще бъдат сравнени с различните прагове, е възможно измерената стойност да се съхрани в променлива. Предлагаме да наменувате тази променлива „Скорост на CO2“. За да създадете тази променлива, щракнете върху секцията „Променливи“.

Отваря се страничен панел: щракнете върху „Създаване на променлива“. Наименувайте променливата например „CO2 rate“ (скорост на CO2), след което щракнете върху ОК. Блокове, специфични за тази нова променлива, се създават и са достъпни от секцията „Променливи“. За да използвате тази променлива, щракнете върху секцията „Променливи“.





## Код за копиране в редактора

```
#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();
}

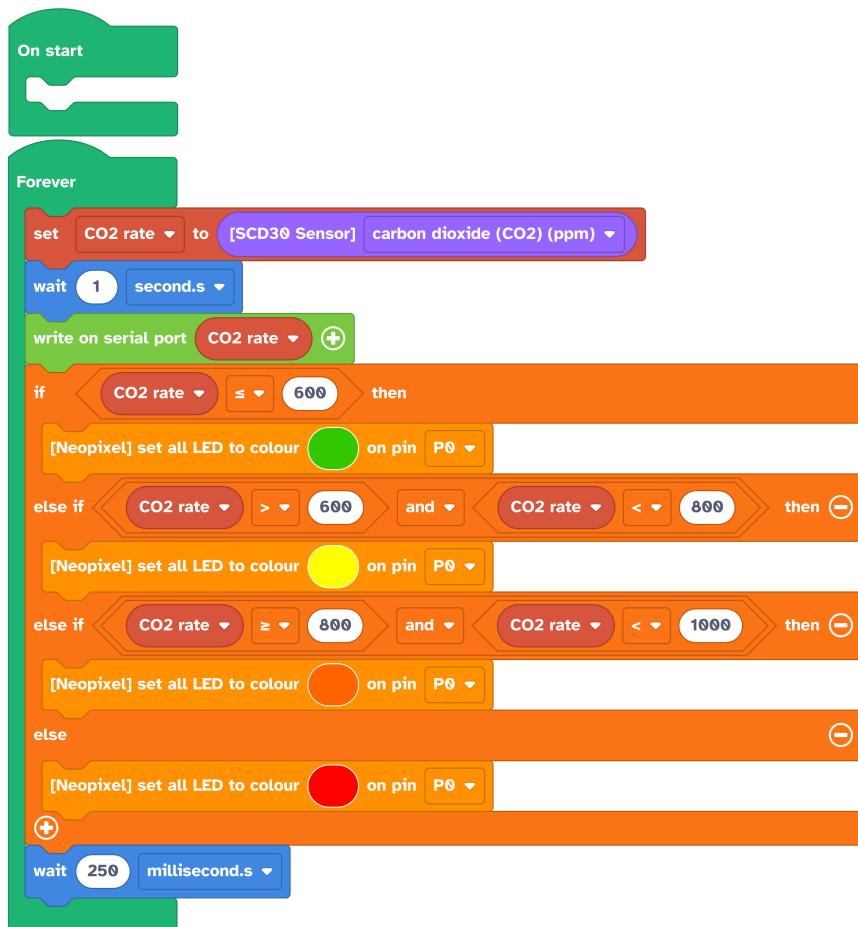
void loop() {
    CO2_rate = scd30_read(0);
    lcdRgb.setCursor(0, 0);
    lcdRgb.print(String("CO2 rate (ppm) :"));
    lcdRgb.setCursor(0, 1);
    lcdRgb.print(String(CO2_rate));
    if (CO2_rate <= 600) {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
    }
    else if (CO2_rate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

Използван издател: vittascience.com/l476; vittascience.com/Arduino или vittascience.com/microbit



Не забравяйте да добавите еднoсекундна пауза, за да ограничите честотата, с която стойностите се показват на екрана на компютъра. Това улеснява отчитането на измерените стойности.



### Код за копиране в редактора

```
#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);

```



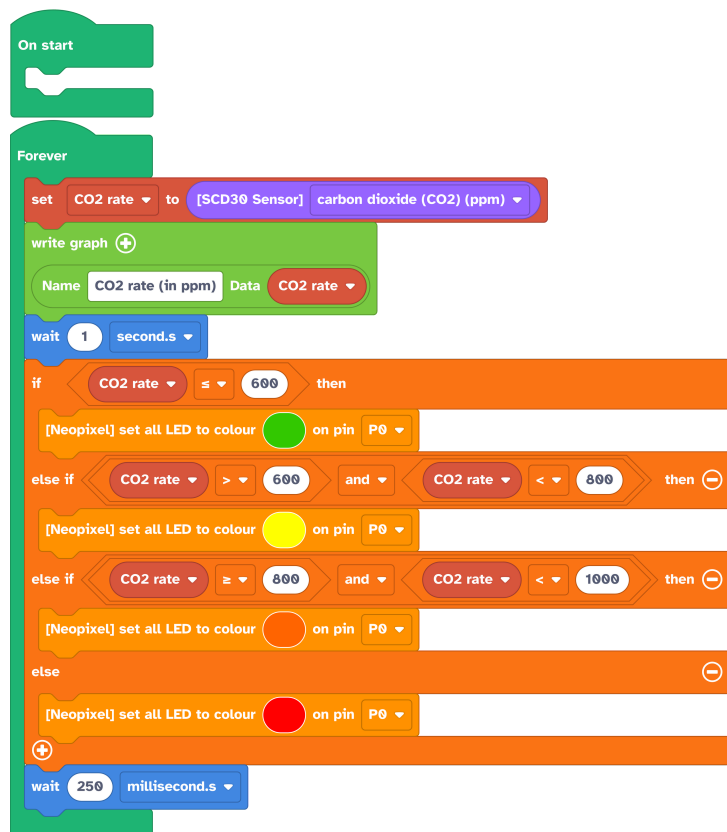
```
lcdRgb.print(String("CO2 rate (ppm)"));
lcdRgb.setCursor(0, 1);
lcdRgb.print(String(CO2_rate));
if (CO2_rate <= 600) {
    neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
} else if (CO2_rate > 600 && CO2_rate < 800) {
    neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 255, 0);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

## Визуализация на данни

Използван издател: [vittascience.com/l476](http://vittascience.com/l476); [vittascience.com/Arduino](http://vittascience.com/Arduino) или [vittascience.com/microbit](http://vittascience.com/microbit)



1. За да видите графиката, в конзолата за показване изберете „Режим на графиката“ отдясно.
2. От това графично представяне е възможно да експортирате всички данни като .csv файл (четим от електронни таблици като Excel, Libre Office Calc, GoogleSheet, Numbers и др.). За да направите това, просто кликнете върху бутона „Експортиране“ в долната част на прозореца. След това данните ще бъдат достъпни от електронната таблица, използвана по подразбиране на компютъра. Функцията за „графично представяне“ на електронната таблица ви позволява да начертаете графика, показваща еволюцията на нивата на въглероден диоксид във времето, която след това може да бъде отпечатана.



## Код за копиране в редактора

```
#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() { }

```