



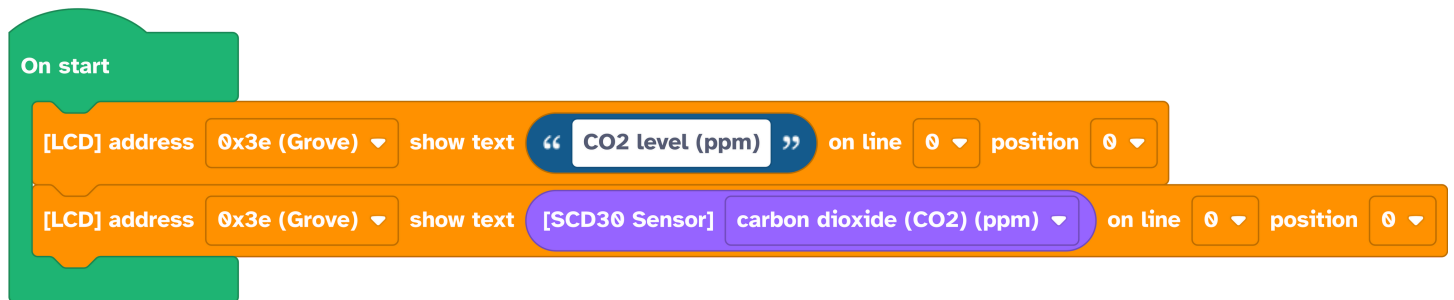
Scheda tecnica - Programmazione e sensori

Misurare la qualità dell'aria interna

1

Visualizza il livello di CO2 sullo schermo LCD utilizzando il sensore SCD30

Editor utilizzato: vittascience.com/l476; vittascience.com/Arduino o vittascience.com/microbit



Codice da copiare nell'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)));
}
```

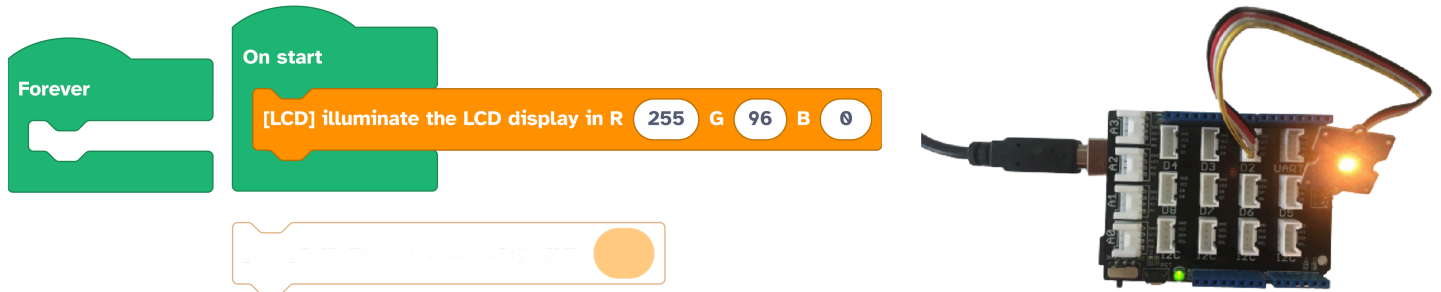
```
void loop() {}
```

2

Configurazione del colore del LED

Editore utilizzato: vittascience.com/l476 ; vittascience.com/Arduino o vittascience.com/microbit

Panoramica del blocco



Codice da copiare nell'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

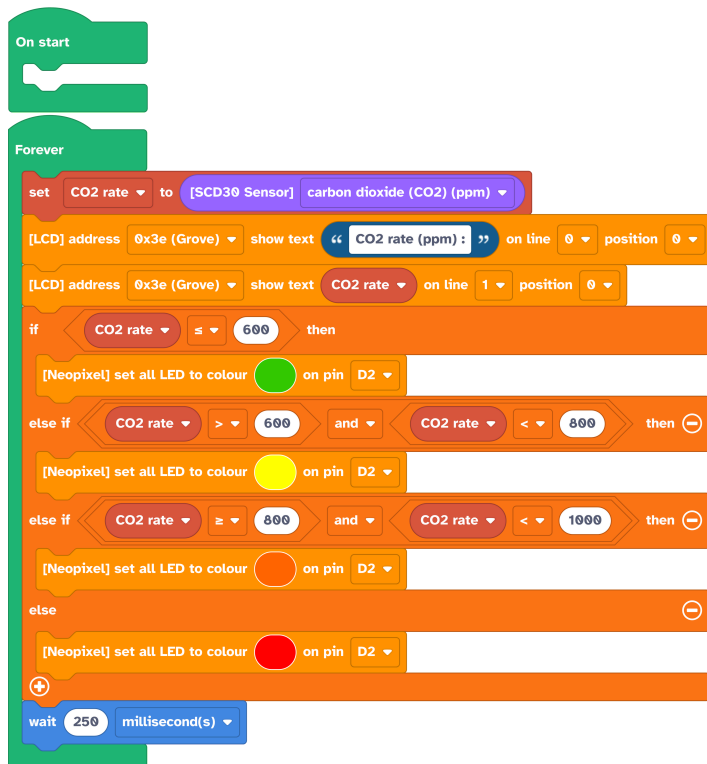
Indicatore LED CO2

Editore utilizzato: vittascience.com/l476 ; vittascience.com/Arduino o vittascience.com/microbit

Nel programma, dobbiamo quindi aggiungere blocchi di "struttura condizionale". Questi blocchi hanno una nomenclatura del tipo: se questa condizione è soddisfatta, allora esegui un'istruzione, altrimenti esegui un'altra istruzione. In questo caso, ci sono quattro condizioni da prevedere, perché ci sono quattro intervalli di velocità di CO2. Questo programma combina i programmi creati nei passaggi 1 e 2, con l'aggiunta di un blocco di struttura condizionale.

In questo programma, il valore di anidride carbonica viene confrontato con diverse soglie. Per evitare di dover effettuare più misurazioni del tasso, che verrà poi confrontato con le diverse soglie, è possibile memorizzare il valore misurato in una variabile. Suggeriamo di chiamare questa variabile "Tasso di CO2". Per creare questa variabile, cliccare sulla sezione Variabili.

Si apre un pannello laterale: clicca su "Crea una variabile". Assegna alla variabile il nome "tasso di CO2", ad esempio, quindi clicca su OK. Verranno creati blocchi specifici per questa nuova variabile, accessibili dalla sezione Variabili. Per utilizzare questa variabile, clicca sulla sezione Variabili.





Codice da copiare nell'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();
}

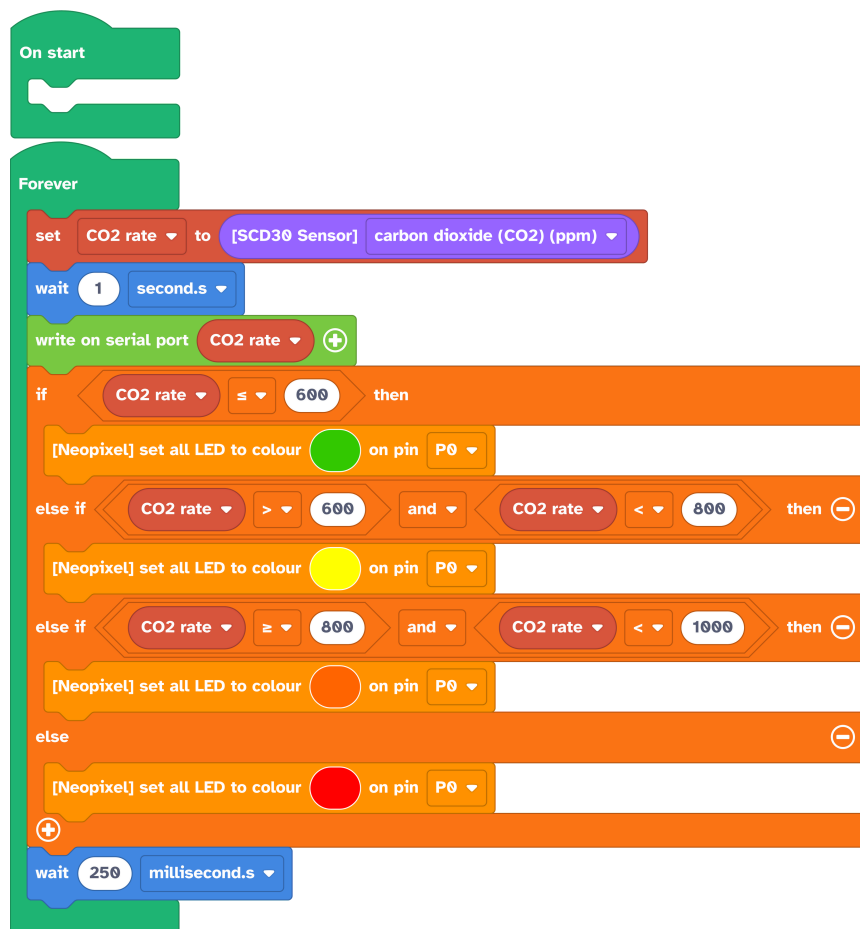
void loop() {
    CO2_rate = scd30_read(0);
    lcdRgb.setCursor(0, 0);
    lcdRgb.print(String("CO2 rate (ppm) :"));
    lcdRgb.setCursor(0, 1);
    lcdRgb.print(String(CO2_rate));
    if (CO2_rate <= 600) {
        neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
    }
    else if (CO2_rate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

Display CO2

Editore utilizzato: vittascience.com/l476 ; vittascience.com/Arduino o vittascience.com/microbit



Ricordatevi di aggiungere una pausa di un secondo per limitare la frequenza con cui i valori vengono visualizzati sullo schermo del computer. Questo facilita la lettura dei valori misurati.



Codice da copiare nell'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);

```



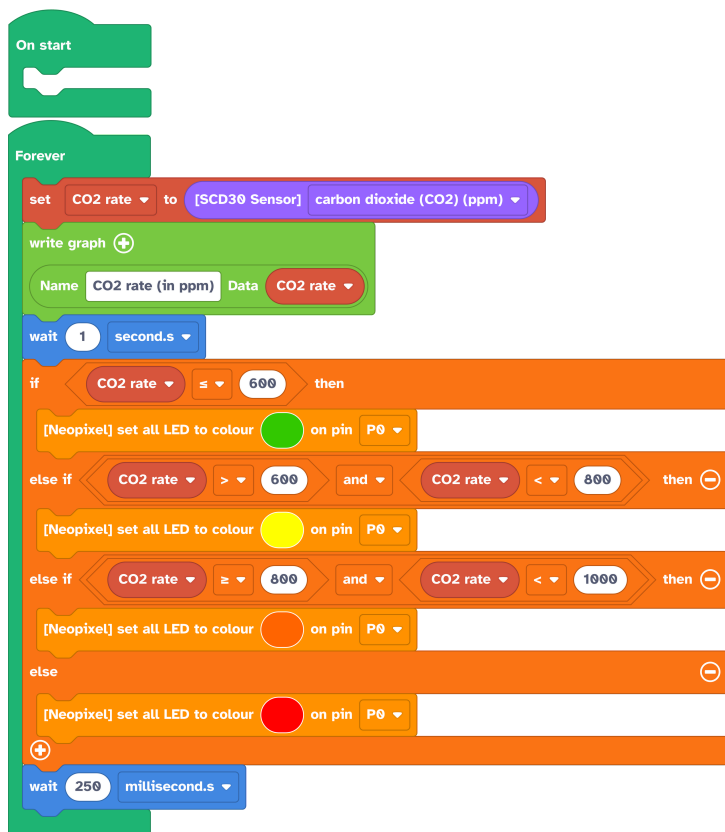
```
lcdRgb.print(String("CO2 rate (ppm)"));
lcdRgb.setCursor(0, 1);
lcdRgb.print(String(CO2_rate));
if (CO2_rate <= 600) {
    neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 51, 204, 0);
} else if (CO2_rate > 600 && CO2_rate < 800) {
    neopixel_showAllLed(&Neopixel_2, NP_LED_COUNT_2, 255, 255, 0);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

Visualizzazione dei dati

Editore utilizzato: vittascience.com/l476 ; vittascience.com/Arduino o vittascience.com/microbit



1. Per visualizzare il grafico, nella console di visualizzazione, selezionare "Modalità grafico" sulla destra.
2. Da questa rappresentazione grafica, è possibile esportare tutti i dati in un file .csv (leggibile da fogli di calcolo come Excel, Libre Office Calc, GoogleSheet, Numbers, ecc.). Per farlo, è sufficiente cliccare sul pulsante Esporta in basso nella finestra. I dati saranno quindi accessibili dal foglio di calcolo utilizzato di default sul computer. La funzione "grafico" del foglio di calcolo consente di disegnare un grafico che mostra l'evoluzione dei livelli di anidride carbonica nel tempo, che può poi essere stampato.



Codice da copiare nell'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() { }

```