



Monitor poziomu CO₂ w powietrzu

W dobie pandemii powstało wiele urządzeń, które mają się przyczynić (pośrednio i bezpośrednio) do minimalizacji przenoszenia groźnego koronawirusa. Niektóre z nich działają w dosyć nieoczywisty sposób, na przykład przypominają nam o wietrzeniu pomieszczeń, co jest zaskakująco ważnym środkiem zapobiegawczym.

Od początku 2020 roku pandemia covid-19 wywarła głęboki wpływ na codzienne życie miliardów ludzi na całym świecie. Perspektywa dostępności szczepionek rzuciła promyk światła na tę sytuację. Warunkiem jest, aby zaszczepiło się ok. 70% populacji kraju, wtedy możliwe będzie powstrzymanie rozprzestrzeniania się wirusa. Nawet dla krajów posiadających doskonałą infrastrukturę medyczną osiągnięcie tego celu stanowi trudne wyzwanie. W takich okolicznościach różne środki, które zostały podjęte na całym świecie przeciwko rozprzestrzenianiu się wirusa (takie jak dystans społeczny, częste mycie rąk, noszenie masek itp.), będą utrzymywane przez długi czas. Każda nowa strategia czy rozwiązania technologiczne, które pomogą w walce z wirusem, mogą mieć istotny wpływ na nasz świat.

W poniższym artykule opisujemy otwarty projekt przenośnego, autonomicznego,

taniego i łatwego w obsłudze urządzenia do monitorowania i rejestrowania stężenia CO₂ w otaczającym powietrzu w czasie rzeczywistym. Autorem tej konstrukcji jest Pierre Carlès, pracujący na co dzień na paryskiej Sorbonie jako dyrektor wydziału inżynierii mechanicznej. W ostatnim czasie, po pracy często zajmuje się zagadnieniami, związanymi z pandemią koronawirusa, statystyką zachorowań itp. Teraz zaproponował proste urządzenie, które wykorzystuje stężenie CO₂ jako estymator zastępujący pomiar ilości innych aerozoli oddechowych. Urządzenie to może pomóc w zdefiniowaniu najlepszych praktyk w zakresie wentylacji przestrzeni zamkniętych, co jak wspomniano, jest ważnym środkiem zapobiegającym rozprzestrzenianiu się covid-19. To prosty pomysł, który pozwala rozwiązać dosyć złożone zagadnienie.

Wirus odpowiedzialny za covid-19 przenosi się między innymi drogą powietrzną (kropelkową), poprzez tzw. aerozole oddechowe – mikroskopijne kropelki, które wydychane są jako normalny produkt uboczny oddychania. Na zewnątrz lub w dużych, otwartych przestrzeniach najlepszą strategią łagodzenia skutków skażenia z powietrza jest zachowanie dystansu. Utrzymanie odległości od jednego do dwóch metrów między ludźmi jest uważane za łatwy sposób na zabezpieczenie przed wymianą wirusów poprzez aerozole. Z kolei w zamkniętych przestrzeniach aerozole oddechowe przemieszczają się na duże odległości i równomiernie rozpraszają w całej objętości pomieszczenia, jeśli mają wystarczająco dużo czasu. W takich okolicznościach zachowanie dystansu społecznego traci skuteczność. W takim pomieszczeniu powietrze może stać się wektorem przenoszącym patogeny. Rozwiązanie tego problemu jest w zasadzie łatwe i polega na zapewnieniu dobrej wentylacji zamkniętych pomieszczeń. Jednak utrudnieniem jest brak odpowiednich mierników, które pozwoliłyby wybrać najlepszą strategię działania wentylacji. W uproszczeniu chodzi o to, jak

często należy otwierać okna lub drzwi oraz na jak długo?

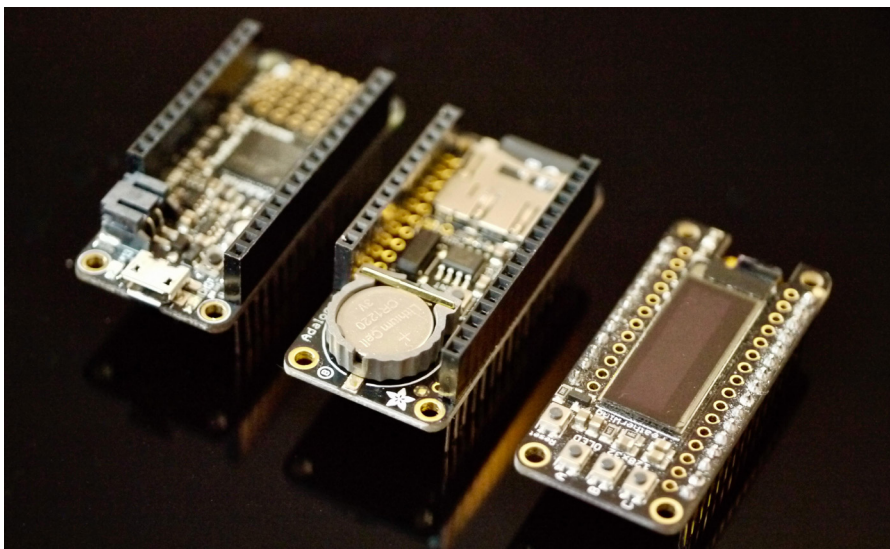
Autor konstrukcji proponuje użycie łatwego sposobu, który polega na monitorowaniu poziomu CO₂ w pomieszczeniu. Pełni on funkcję wskaźnika zastępczego, aby zapewnić ilościowy pomiar, ile powietrza w pomieszczeniu jest potencjalnie wypełnione aerozolami oddechowymi. W tym celu zaprojektował prosty system do pomiaru zawartości dwutlenku węgla w powietrzu.

Potrzebne elementy

Do zmontowania systemu użyto szeregu gotowych elementów. Dzięki temu, że układ oprogramowany jest w Arduino IDE, łatwo jest wymieniać je na inne, w razie potrzeby. Całkowity koszt elementów to około 100 euro. Autor zastosował następujące moduły:

- sensor dwutlenku węgla SEN0219, osadzony na module Gravity,
- moduł Adafruit Feather 328P lub dowolny inny moduł, kompatybilny z Arduino (Feather M0, M4, 32u4 itp.),
- moduł Adafruit Featherwing Adalogger lub dowolny inny moduł (moduły), umożliwiający podłączenie karty SD poprzez interfejs SPI i zegara czasu rzeczywistego przez I²C,
- moduł Adafruit Featherwing OLED z ekranem 128×32 pikseli lub dowolny inny ekran OLED, sterowany poprzez I²C,
- karta microSD,
- diody LED RGB o średnicy 5 mm (wersja ze wspólną katodą),
- opornik 120 Ω
- power bank z wyjściem USB o pojemności 1500 mAh lub więcej,
- krótki kabel USB A – micro-USB,
- bateria CR1220,
- piny i złącza (męskie i żeńskie),
- przewody do wykonania połączeń, koszulka termokurczliwa,
- dowolna obudowa, jej wymiary podyktowane są głównie wielkością power banku.

Wymieniony wcześniej moduł z mikrokontrolerem – Feather 328P, jest odpowiednikiem płytki Arduino, dostosowanej do formatu Feather i działającej na poziomie logicznym 3,3 V. Jeśli wymiary nie są problemem, to można go zastąpić dowolną płytką z rodziny Arduino. Wymogiem dla mikrokontrolera do tego projektu jest możliwość komunikowania się z urządzeniami peryferyjnymi zarówno przez SPI, jak i I²C oraz co najmniej jeden pin wejściowy wyposażony w przetwornik analogowo-cyfrowy (ADC). Zamiast wymienionego tutaj modelu 328P można zastosować dowolny inny moduł kompatybilny z Arduino, jak na przykład płytki Feather M0, M4 lub 32u4 albo dowolny inny model kompatybilny z tym środowiskiem programistycznym. Autor uzasadnia wybór modelu 328P niskim kosztem – moduł



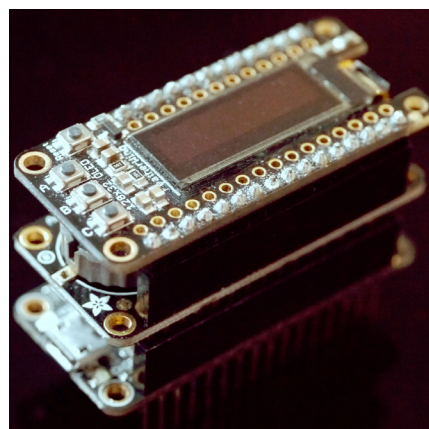
Fotografia 1. Moduły składające się na miernik poziomu CO₂

ten jest najtańszą dostępną na rynku płytką w formacie Feather.

Płytki Featherwings są odpowiednikiem shieldów, znanych z klasycznych modułów ekosystemu Arduino. Jeśli moduł z mikrokontrolerem zastąpimy typową płytką ekosystemu Arduino, to wtedy wymienione płytki Featherwings muszą zostać zastąpione równoważnymi shieldami.

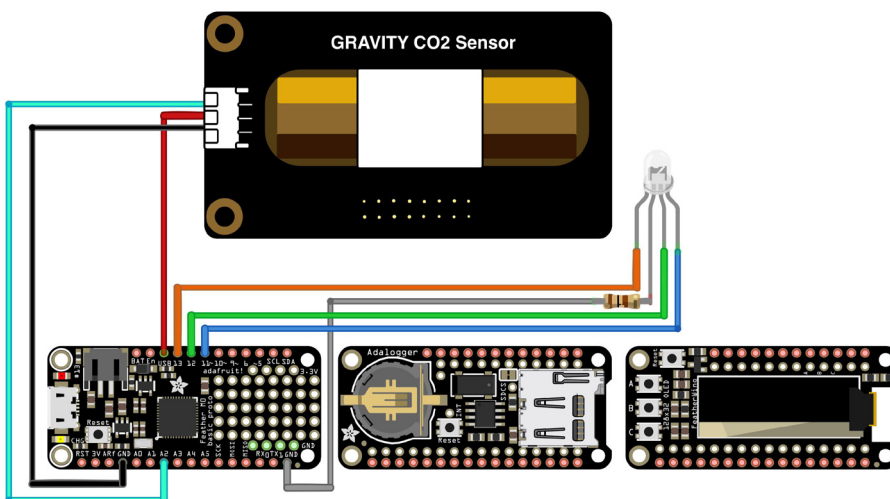
Opis modułów

Moduły zastosowane w urządzeniu zostały pokazane na **fotografii 1**. Featherwing OLED, jak sama nazwa wskazuje, to mały ekran OLED, który komunikuje się przez I²C z mikrokontrolerem. Ciekawą cechą zalecanego tutaj modelu wyświetlacza jest obecność trzech przycisków (A, B i C), które służą w systemie do interakcji z interfejsem urządzenia. W razie potrzeby można je łatwo zastąpić niezależnymi przyciskami. Featherwing Adalogger to połączenie zegara czasu rzeczywistego z interfejsem I²C i czytnika kart SD SPI. Służy do rejestrowania pomiarów oraz odpowiadającej im godziny i daty.



Fotografia 2. Zmontowane moduły

Power bank można zastąpić dowolną ładowarką ze złączem micro-USB w przypadku, gdy urządzenie ma być używane jako urządzenie stacjonarne zasilane z gniazdka. Należy pamiętać, że power banki często są dostarczane z własnym kablem USB. W wielu przypadkach kable takie mają tylko przewody zasilające i nie można ich używać do programowania ani przesyłania danych.



Rysunek 1. Schemat połączeń modułów w systemie

Listing 1. Kod szkicu dla Arduino IDE

```

#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <RTClib.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

Adafruit_SSD1306 display = Adafruit_SSD1306(128, 32, &Wire);
RTC_PCF8523 rtc;
// Piny, do których dołączone są przyciski
#define BUTTON_A 9
#define BUTTON_B 6
#define BUTTON_C 5
// Piny do których dołączone są diody LED
#define Blue 13
#define Red 12
#define Green 11
// Poziomy napięcia w systemie
float CO2_V = 0.;
float logic_V = 3.3;

unsigned int CO2_ppm;
// Domyślny próg CO2
unsigned int Threshold = 1000;
// Czas rzgrzewania sensora
unsigned int preheat = 120;
unsigned int Count = 0;
// Pin wejścia CS karty SD
const int chipSelect = 10;

unsigned long t,t0;
boolean Status_A = LOW;

String dataString = "";
String fileString = "LOG";

void setup() {
  SD.begin(chipSelect);
  // Konieczne by używać 12 bit ADC
  analogReadResolution(12);
  pinMode(Red, OUTPUT);
  pinMode(Green, OUTPUT);
  pinMode(Blue, OUTPUT);
  pinMode(BUTTON_A, INPUT_PULLUP);
  pinMode(BUTTON_B, INPUT_PULLUP);
  pinMode(BUTTON_C, INPUT_PULLUP);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.display();
  // Czas na uruchomienie modułów
  delay(1000);
  analogWrite(Red,0);
  analogWrite(Green,0);
  analogWrite(Blue,5);
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE,0x0000);
  display.setTextSize(2);
  display.println("Preheating");
  display.setTextSize(1);
  display.println();
  display.println("Please wait ...");
  display.display();
  analogWrite(Blue,0);
  analogWrite(Red,2);
  display.clearDisplay();
  display.display();
}

void loop () {
  // Pobiera czas z RTC
  DateTime now = rtc.now();
  // Nadaje unikatowe nazwy logom
  fileString += String(now.unixtime()).substring(4,9);
  fileString += ".txt";
  while (HIGH) {
    t0 = millis();
    CO2_V = 0;
    CO2_ppm = 0;
    Count = 0;
    while (millis() - t0 < 10000) {
      // 4095 należy zasrtać 1023 przy 10-bit ADC
      CO2_V = logic_V*analogRead(A2)/4095.;
      CO2_ppm += int((CO2_V - 0.4)*3125);
      Count += 1;
      Serial.print(Count);
      Serial.print(' ');
      Serial.println(CO2_ppm);
      t = millis();
    }
    while (millis() - t < 2000) {
      if (digitalRead(BUTTON_A)==LOW) {
        Status_A = HIGH;
        displaythreshold();
      }
      if (Status_A == HIGH) {
        if (digitalRead(BUTTON_B)==LOW && Threshold < 5000) {
          Threshold = Threshold + 250;
          displaythreshold();
          delay(200);
        }
        else if (digitalRead(BUTTON_C)==LOW && Threshold > 0) {
          Threshold = Threshold - 250;
          displaythreshold();
          delay(200);
        }
      }
    }
  }
  Status_A = LOW;
  CO2_ppm = CO2_ppm / Count;
}

```

Montaż modułów

Pierwszym zadaniem jest zainstalowanie goldpinów w poszczególnych modułach – mikrokontrolerze Feather i modułach Featherwings. Trzy PCB można następnie ułożyć w stos lub umieścić obok siebie na płytce prototypowej lub w module Featherwing Tripler. W przypadku tych ostatnich rozwiązań wystarczy wyposażyć każdą płytkę w dołączone do niej proste goldpiny. Do układania płytek w bardziej zwartą i bardziej praktyczną konstrukcję należy stosować żeńskie piny do układania w stopy. Trzeba tylko pamiętać, że niezależnie od wybranej konfiguracji, OLED Featherwing zawsze wymaga męskich złączy, znajdując się na szczycie stosu.

Złącza są instalowane poprzez wlutowanie ich do odpowiednich otworów w każdej płytce. Niektórzy dystrybutorzy oferują nawet płytki z lutowanymi złączami, co znacznie upraszcza montaż. Należy jednak zachować ostrożność, jeśli chodzi o typ goldpinów, które są instalowane domyślnie. W większości przypadków są to zwykłe męskie piny, przystosowane do płytek stykowych i nie umożliwiające instalowania wielu płytek jedna na drugiej.

Gdy wszystkie trzy moduły zostaną wyposażone w odpowiednie złącza, wystarczy je ułożyć w stos i cały zestaw będzie gotowy. Zmontowane moduły Feather i Featherwing pokazano na **fotografii 2**.

Pozostała część obwodu jest dosyć prosta. Trzy ułożone w stos płytki łączą się ze sobą w naturalny dla siebie sposób. Do nich zostaje nam podłączyć tylko diodę LED RGB oraz czujnik CO₂ i zasilanie. Dioda LED jest podłączona do pinów I/O nr 13 (czerwona), 12 (niebieska) i 11 (zielona), a jej wspólna katoda jest podłączona do masy. Można wybrać inne piny, ale ten konkretny wybór był uzasadniony potrzebą uniknięcia stosowania pinu 10, na który wpływa komunikacja sprzętowa SPI między mikrokontrolerem a płytką Adaloggera. Poza tym na płytkach Feather (i w większości płytek Adafruit) pin numer 13 jest połączony z czerwoną diodą LED SMD znajdującą się na samej płytce. Przy tym wyborze połączenia czerwony kanał zewnętrznej diody LED RGB zawsze naśladuje czerwoną diodę LED na płycie, co może być wygodne przy debugowaniu. Ważne jest, aby dokładnie sprawdzić, które piny są domyślnie używane przez shieldy lub moduły Featherwings, z którymi łączy się z płytą główna, aby uniknąć konfliktów.

Wybrany przez autora czujnik CO₂ działa na zasadzie pomiarów w bliskiej podczerwieni (NDIR) do estymacji poziomu CO₂ w powietrzu. Moduł produkowany jest przez firmę Gravity. Na rynku dostępnych jest kilka innych modeli tego rodzaju, jednakże większość z nich jest droższa. Ten konkretny model ma wyjście analogowe, z sygnałem

zawierającym się w przedziale 0...2 V, w zależności od stężenia dwutlenku węgla w analizowanym powietrzu. Wyjście miernika jest podłączone do pinu A2 na płytce Feather. Ten pin jest używany jako wejście analogowe, a instrukcja w kodzie aktywuje podłączony do niego 12-bitowy przetwornik analogowo-cyfrowy. Wybór pinu A2 ma na celu uwolnienie innych potencjalnie ważnych pinów analogowych. Autor wskazuje tutaj szczególnie na piny A0 i A1, które są powiązane z dwoma przetwornikami cyfrowo-analogowymi na wielu płytkach Feather. Jest to istotne, na wypadek gdyby potrzebne było podłączenie innych modułów do układu.

Na koniec należy podłączyć zasilanie do wszystkich modułów – do płytki Feather, z USB, a do pozostałych modułów zasilanie 3,3 V, które jest dostępne na płytce z mikrokontrolerem. Na **rysunku 1** zaprezentowano schemat połączeń poszczególnych modułów systemu, ale bez źródła zasilania. Dla uproszczenia budowy systemu zasilanie podawane jest np. z power banku wprost do portu micro-USB na płytce Feather. W ten sposób nie jest potrzebny żaden dodatkowy moduł do np. zwiększania napięcia z 3,3 V do 5 V, wymagane przez niektóre moduły w systemie.

Układ charakteryzuje się stosunkowo niewielkim zużyciem prądu, dzięki czemu power bank o pojemności 2000 mAh może zasilać urządzenie, zapewniając jego nieprzerwaną pracę przez ponad 24 godziny.

Ponieważ płytka z mikrokontrolerem i dwa podłączone do niej moduły ułożone są w stos, styki na płytkach nie są bezpośrednio dostępne do dolutowania przewodów. Istnieje jednak kilka sposobów, które pozwalają rozwiązać ten problem. Jeśli na płycie Feather zostały zainstalowane długie żeńskie złącza do łączenia pionowego (jak na fotografii 2), to połączenia można łatwo przyłutować do pinów wystających z dolnej części płytki. Większość płytek Featherwing jest wyposażona w podwójny rząd pinów, które mają umożliwić użytkownikowi podłączenie przewodów do pinów, a jednocześnie umożliwiają układanie modułów w stos. Samą płytkę Feather można przyłutować na małej płytce uniwersalnej, do której można łatwo przyłutować inne przewody.

Obudowa

Gdy wszystko jest już okablowane i podłączone, jedyne, co pozostaje do zrobienia, to znaleźć pudełko o odpowiednich wymiarach, np. takie jak na **fotografii 3**, aby zmieścić w nim wszystkie elementy. Następnie należy wykonać w nim wymagane otwory tam, gdzie jest to potrzebne i na koniec zabezpieczyć wszystkie elementy elektroniczne i power bank w środku. Łatwym i praktycznym sposobem ich mocowania jest użycie dwustronnej piankowej taśmy samoprzylepnej. Pozwala ona na połączenie ze sobą

Listing 1. cd.

```

DateTime now = rtc.now();
display.setCursor(0,0);
display.clearDisplay();
display.display();
display.setTextSize(1);
if (CO2_V==0) {
display.println("Error");
display.println(" ");
}
else if (CO2_V < 0.4) {
display.println("Preheating...");
display.println(" ");
display.setTextSize(2);
display.print(CO2_V);
display.println(" V");
}
else {
displayhour(now.day());
display.print('/');
displayhour(now.month());
display.print('/');
display.print(now.year(), DEC);
display.print(" ");
displayhour(now.hour());
display.print(':');
displayhour(now.minute());
display.println();
display.println(" ");
display.setTextSize(2);
display.print(CO2_ppm);
display.println(" ppm");
dataString = String(now.day());
dataString += '/';
dataString += String(now.month());
dataString += '/';
dataString += String(now.year());
dataString += '/';
dataString += String(now.hour());
dataString += ':';
dataString += String(now.minute());
dataString += '.';
dataString += String(now.second());
dataString += ' ';
// Zapisuje aktualny czas w sekundach
dataString += String(now.unixtime());
dataString += ' ';
dataString += String(CO2_ppm);
File dataFile = SD.open(fileString, FILE_WRITE);
// Zapisuje wartość CO2 w ppm na SD
dataFile.println(dataString);
dataFile.close();
}
display.display();
if (CO2_ppm <= Threshold) {
analogWrite(Red,0);
analogWrite(Green,3);
}
else {
analogWrite(Green,0);
analogWrite(Red,2);
}
}
}

void displayhour(int number) {
// Funkcja zapisania godziny,
// pozwala na wyświetlenie '0'
if (number < 10) {
display.print('0');
display.print(number);
}
else {
display.print(number);
}
}

void displaythreshold() {
display.setCursor(0,0);
display.clearDisplay();
display.setTextSize(1);
display.println("Choose Threshold");
display.println(" ");
display.setTextSize(2);
display.print(Threshold);
display.println(" ppm");
display.display();
}
}

```

nawet nierównych elementów, takich jak płytki drukowane. Na **fotografii tytułowej** pokazano układ w obudowie autora.

Oprogramowanie

Ostatnim etapem realizacji projektu jest załadowanie do modułu z mikrokontrolerem odpowiedniego oprogramowania. Wystarczy podłączyć płytkę z mikrokontrolerem do komputera i wgrać szkic, który został pokazany na **listingu 1**. Należy to zrobić za pomocą Arduino IDE.

Adalogger Featherwing ma dwie funkcje – obsługuje kartę micro-SD i dostarcza czas z układu RTC. Przed załadowaniem kodu programu do płytki Feather ważne jest, aby zainicjować ten zegar aktualnym czasem. Należy to zrobić tylko raz w całym okresie eksploatacji urządzenia (a dokładniej w okresie życia baterii Adaloggera, czyli kilku lat). Aby to zrobić, należy włożyć baterię do modułu, a następnie wgrać i uruchomić specjalny skrypt na mikrokontrolerze w układzie. Można go znaleźć w Arduino



Fotografia 3. Elektronika w obudowie

IDE w przykładach, a dokładniej mówiąc, w menu RTCLib. Nosi on nazwę układu zegara czasu rzeczywistego, jaki chcemy zaprogramować. W przypadku Adaloggera jest to PCF8523. Po załadowaniu i uruchomieniu tego programu (może być konieczne otwarcie w tym celu monitora szeregowego) zegar w Adaloggerze jest ustawiony na aktualny czas zdefiniowany przez komputer. Pozostanie zsynchronizowany z tym czasem tak długo, jak działa bateria.

Po zainicjowaniu zegara czasu rzeczywistego można już wgrać docelowy szkic Arduino do płytki Feather. Proponowany kod jest napisany dla Arduino IDE. Bazuje na następujących bibliotekach, które należy najpierw zaimportować do Arduino IDE za pomocą menedżera bibliotek:

- SPI,
- SD,
- Wire,
- RTCLib,
- Adafruit_GFX,
- Adafruit_SSD1306.

Kod pokazany na listingu 1 można pobrać z następującego repozytorium GitHub: <http://bit.ly/301mdmA>. Struktura tego programu jest bardzo prosta. Po włączeniu urządzenia rozpoczyna dwuminutową fazę nagrzewania wstępnego, podczas której dioda LED świeci na niebiesko. Jest to konieczne, aby sensor CO₂ działał poprawnie. W tym czasie

tworzony jest też plik na karcie microSD, na której zapisywane będą poziomy dwutlenku węgla. Nazwa pliku jest generowana automatycznie podczas rozruchu jako: LOG****.TXT, gdzie gwiazdki są zastępowane unikalnym, 5-cyfrowym indeksem. Ułatwia to porządkowanie plików w czasie i chroni starszych plików w przypadku ponownego uruchomienia urządzenia.

Następnie rozpoczyna się normalny cykl pracy, w którym pomiar stężenia CO₂ dokonywany jest co dwie sekundy. Po serii pięciu takich pomiarów szacunkowa wartość stężenia CO₂ jest obliczana jako średnia. Następnie jest ona zapisywana na karcie microSD wraz z datą i godziną. Te same dane są również wyświetlane na ekranie wyświetlacza OLED.

Zmierzone stężenie dwutlenku węgla jest porównywane z progami zdefiniowanymi przez użytkownika. Jeśli wartość przekroczy ten próg, dioda LED zmienia kolor na czerwony, w przeciwnym razie świeci w kolorze zielonym. Następnie rozpoczyna się nowy cykl pomiarowy.

Pomiędzy dwoma pomiarami program sprawdza jeszcze, czy przycisk A jest wciśnięty. Jeśli tak, przechodzi do menu, w którym użytkownik może zwiększać lub zmniejszać próg sygnalizacji z krokiem ± 250 ppm za pomocą przycisków B i C.

Domyślny próg jest ustalony na 1000 ppm, zgodnie z zaleceniami kilku międzynarodowych organizacji zdrowotnych.

Uwagi

Zaprezentowane urządzenie jest niewielkie, ponieważ monitor CO₂ ma być przenośny, umożliwiając użytkownikowi łatwe używanie go w różnych pomieszczeniach, a nawet na zewnątrz. Umieszczenie magnesów po stronie przeciwnej do ekranu OLED pozwala na łatwe ustawienie urządzenia na metalowych powierzchniach. Zaleca się, aby czujniki CO₂ były umieszczane na wysokości od 1 do 1,5 metra, z dala od drzwi i okien.

Przy tak niewielkich rozmiarach wizualna sygnalizacja informująca użytkownika o nadmiernym stężeniu CO₂ (tj. czerwona dioda LED i liczba na ekranie) jest stosunkowo dyskretna. Może się to wydawać złym wyborem projektowym, jeśli celem jest ostrzeżenie o potrzebie przewietrzenia pomieszczenia. Jednak w istocie jest wręcz przeciwnie. Dyskretne wskazania są celowym wyborem projektowym. Autor wskazuje, że działania profilaktyczne powinny być przemyślane, z uwzględnieniem potencjału w przypadku nieodpowiednich i wywołujących niepokój wiadomości. Duży, migający wyświetlacz informujący, że poziom CO₂ jest zbyt wysoki, z pewnością kwalifikowałby się jako urządzenie wywołujące niepokój, które mogłoby przynieść efekt przeciwny do zamierzonego, powodując niepotrzebne napięcie. Wentylacja pomieszczenia trwa kilka minut i musi być powtarzana co kilkadziesiąt minut, jeśli w pomieszczeniu przebywają ludzie. W takich warunkach i biorąc pod uwagę, że ryzyko skażenia powietrznego jest niewielkie, nawet w zamkniętych przestrzeniach (pod warunkiem, że ludzie noszą maseczki) traktowanie wentylacji jako natychmiastowej konieczności awaryjnej byłoby nadmierne.

Przedstawione urządzenie należy postrzegać raczej jako sposób określenia ilościowych wytycznych dotyczących wentylacji, aby zapewnić możliwie najwyższą jakość powietrza w pomieszczeniach zamkniętych. Tego rodzaju układ powinien być raczej monitorowany przez jedną wyznaczoną osobę, a nie używany jako publiczny sygnał ostrzegawczy.

Nikodem Czechowski, EP

Źródła:

- <http://bit.ly/37VvF8a>
- <https://bit.ly/301mdmA>

O projektach, mini, soft i wielu innych diskutuj
na <https://forum.ep.com.pl>