



**WYŻSZA SZKOŁA  
INFORMATYKI I ZARZĄDZANIA**  
z siedzibą w Rzeszowie

## **Pomiary i sterowanie w sieciach sensorowych**

***Temat: Projekt „Inteligentny System Ostrzegawczy”***

Wykonawca: Kamil Markowski

Numer albumu: 64152

Prowadzący przedmiot: Dr inż. Janusz Kolbusz

Przedmiot: Pomiary i sterowanie w sieciach sensorowych

**Rzeszów 2022**

## Spis treści

<b>1. OPIS ZADANIA PROJEKTOWEGO.....</b>	<b>3</b>
<b>2. OPIS TECHNOLOGII ZWIĄZANYCH Z ROZWIĄZANIEM ZADANIA.....</b>	<b>3</b>
MIKROKONTROLER ARDUINO UNO R3 .....	3
DIODA LED RGB .....	3
DIODA LED .....	3
GŁOŚNIK PIEZO .....	3
CZUJNIK GAZU .....	3
CZUJNIK TEMPERATURY TMP36.....	3
REZYSTOR.....	3
PRZYCISK.....	4
<b>3. SCHEMAT SYSTEMU .....</b>	<b>4</b>
<b>4. PROGRAM SKRYPTU .....</b>	<b>5</b>
<b>5. SPOSÓB URUCHAMIANIA.....</b>	<b>6</b>
PRZYPADEK I DLA ZANIECZYSZCZEŃ .....	6
PRZYPADEK II DLA ZANIECZYSZCZEŃ .....	7
PRZYPADEK III DLA ZANIECZYSZCZEŃ.....	7
PRZYPADEK IV DLA TEMPERATURY .....	8
<b>6. ROZWIĄZANIE Z OPISEM POSZCZEGÓLNYCH LINII SKRYPTU.....</b>	<b>9</b>
FUNKCJA VOID SETUP() .....	9
FUNKCJA VOID LOOP() .....	9
<b>7. PODSUMOWANIE I WNIOSKI .....</b>	<b>11</b>
PODSUMOWANIE .....	11
WNIOSKI .....	11
<b>8. LITERATURA.....</b>	<b>11</b>

## **1. Opis zadania projektowego**

Projekt „Inteligentny system ostrzegawczy” oparty został o mikrokontroler Arduino Uno R3. Układ na bieżąco monitoruje stan powietrza (gazu, dwutlenku węgla) w pomieszczeniu oraz temperaturę, sygnalizując ich zmianę.

## **2. Opis technologii związanych z rozwiązaniem zadania**

### **Mikrokontroler Arduino Uno R3**

Ma 14 pinów wejścia/wyjścia cyfrowego (z których 6 można używać jako pinów PWM), 6 wejść cyfrowych, rezonator kwarcowy 16 MHz, złącze USB, gniazdo zasilania, listwę ICSP oraz przycisk resetowania.

### **Dioda LED RGB**

To dioda, która umożliwia świecenie w 3 kolorach (red green blue), emituje energię w postaci światła. Konieczne jest używanie rezystorów z diodą LED.

### **Dioda LED**

To dioda, która emituje energię w postaci światła. Konieczne jest używanie rezystorów z diodą LED.

### **Głośnik piezo**

Emituje sygnał dźwiękowy. Elementy piezoelektryczne są przydatne do wykrycia wibracji lub pukania. Można z nich korzystać w łatwy sposób, odczytując napięcie na wyjściu. Mogą być również wykorzystywane do bardzo małych przetworników audio takich jak buzzer.

### **Czujnik gazu**

Czujnik pozwala wykryć butan, propan, metan, LPG, alkohol, wodór i dym.

### **Czujnik temperatury TMP36**

Analogowy termometr w którym napięcie wyjściowe jest proporcjonalne do mierzonej temperatury z zakresu: od -40 °C do +120 °C. Zasilany jest napięciem od 2,7 do 5,5 V.

### **Rezystor**

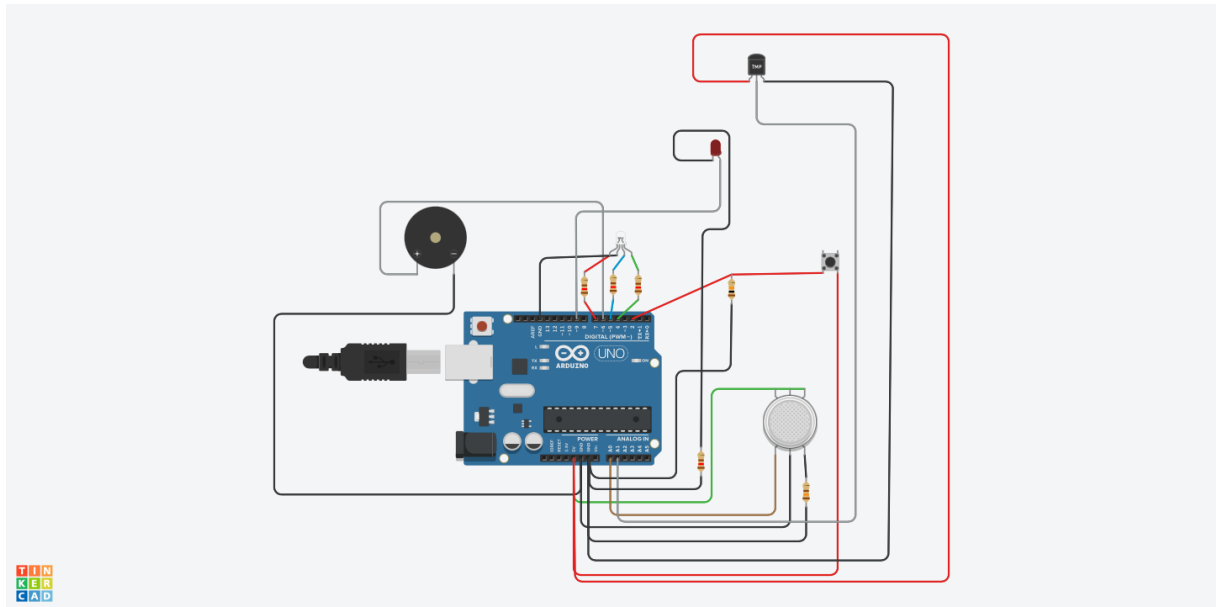
Ogranicza prąd, tworzone są z materiału niezbyt dobrze przewodzącego prąd ich konduktywność powinna być niska tj. nie przewodzić prądu, to dwójnik posiada dwa zaciski

Jest elementem pasywnym, przemiana energii elektryczną na energię ciepłą,

## Przycisk

Przycisk monostabilny montowany w sposób przewlekany

### 3. Schemat systemu



Rys. 1 Schemat inteligentnego systemu ostrzegawczego

Projekt przedstawiony na Rys. 1 składa się z:

- Mikrokontrolera Arduino Uno,
- przewodów,
- sześciu rezystorów
- czerwona diod LED (jedna sztuka),
- dioda LEG RGB (jedna sztuka),
- buzzer,
- przycisk,
- czujnik gazu,
- czujnik temperatury TMP36.

#### 4. Program skryptu

```
void setup()
{
  pinMode(1, OUTPUT);
  pinMode(A0, INPUT);
  pinMode(2, INPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(A1, INPUT);
  pinMode(9, OUTPUT);
}

void loop()
{
  if (analogRead(A0) <= 20 && digitalRead(2) == 0) {
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
  } else {
    if (analogRead(A0) >= 21 && analogRead(A0) <= 29) {
      digitalWrite(5, HIGH);
      digitalWrite(4, LOW);
      digitalWrite(7, LOW);
    }
    if (analogRead(A0) > 29) {
      digitalWrite(4, LOW);
      digitalWrite(5, LOW);
      digitalWrite(7, HIGH);
      tone(6, 523, 1000); // play tone 60 (C5 = 523 Hz)
    }
  }
  if ((-40 + 0.488155 * (analogRead(A1) - 20)) > 40) {
    digitalWrite(9, HIGH);
  } else {
    digitalWrite(9, LOW);
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```

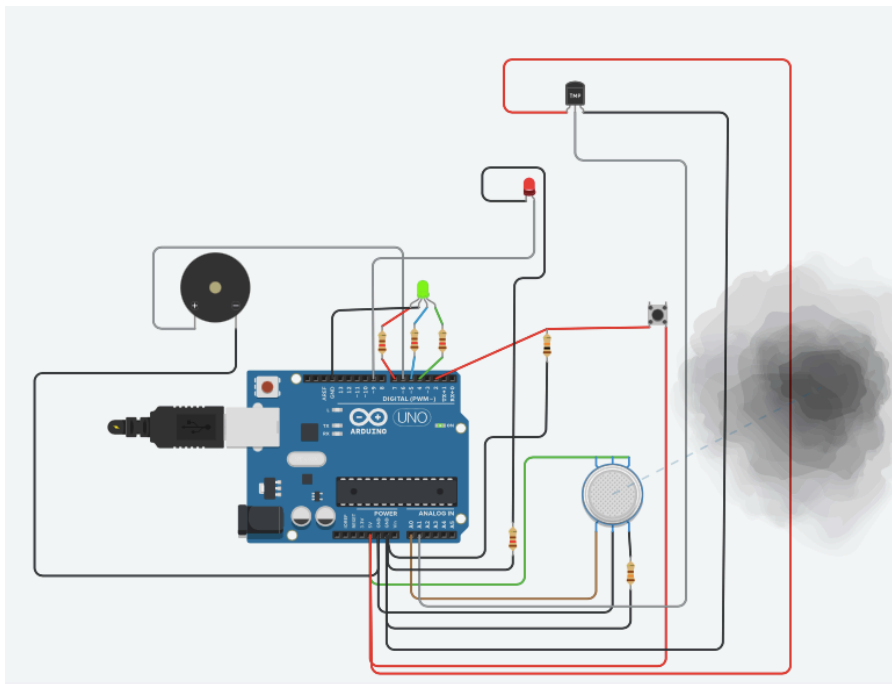
## 5. Sposób uruchamiania

W celu uruchomienia programu należy zalogować / zarejestrować się na platformie thinkercad.com. Następnie należy wczytać projekt i kliknąć lewym przyciskiem myszy na ikonkę „uruchom symulację”. Kliknięcie w czujnik gazu spowoduje powstanie chmury zanieczyszczeń, im bliżej przesunięta zostanie tym większe zanieczyszczenie wykryje czujnik, co przekłada się na sposób ostrzegania. (Rys. 2, Rys. 3, Rys. 4).

Klikając w czujnik temperatury TMP pokazuje się suwak, którym można przesuwając w prawo zwiększając temperaturę lub w lewo zmniejszając. (Rys. 5, Rys. 6).

### Przypadek I dla zanieczyszczeń

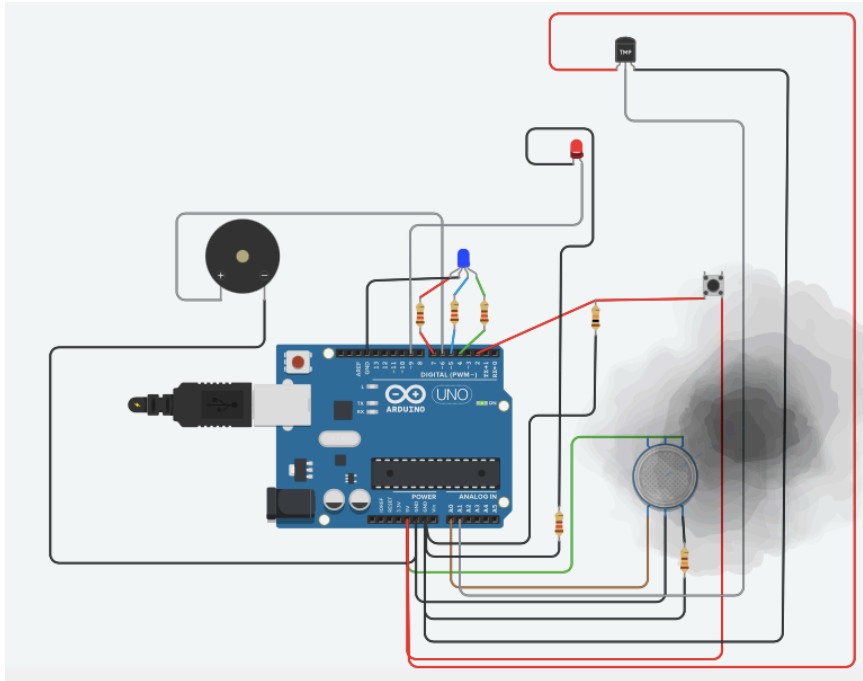
Zanieczyszczenia są daleko od czujnik – brak alarmu dźwiękowego, dioda RGB świeci na zielono. (Rys. 2).



Rys. 2 Jakość powietrza bardzo dobra

## Przypadek II dla zanieczyszczeń

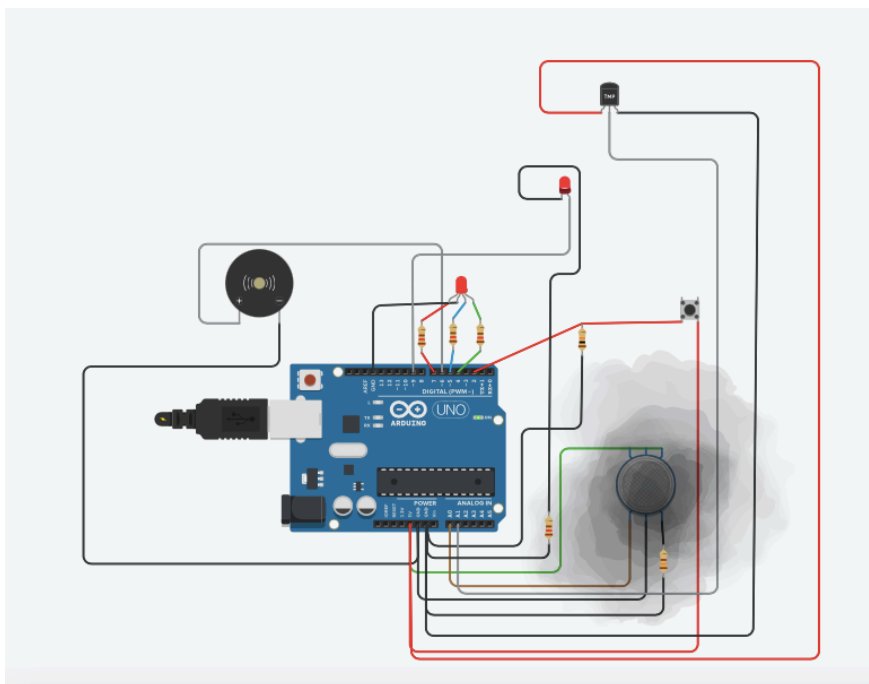
Czujnik wykrywa lekkie zanieczyszczenia – brak alarmu dźwiękowego, dioda RGB świeci na niebiesko. (Rys. 3).



Rys. 3 Jakość powietrza dobra

## Przypadek III dla zanieczyszczeń

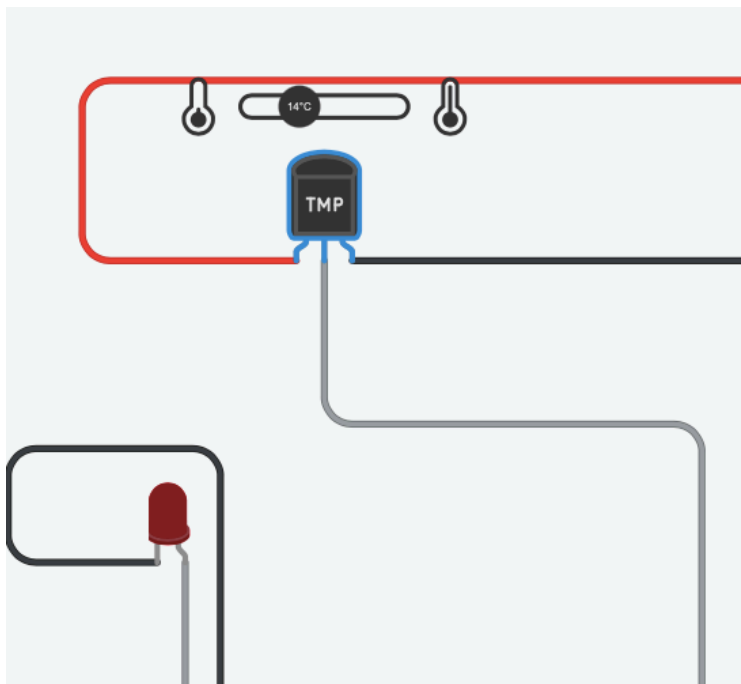
Czujnik wykrywa duże zanieczyszczenia –uruchamiany jest alarm dźwiękowy, dioda RGB świeci na czerwono. (Rys. 3).



Rys. 4 Jakość powietrza zła

#### Przypadek IV dla temperatury

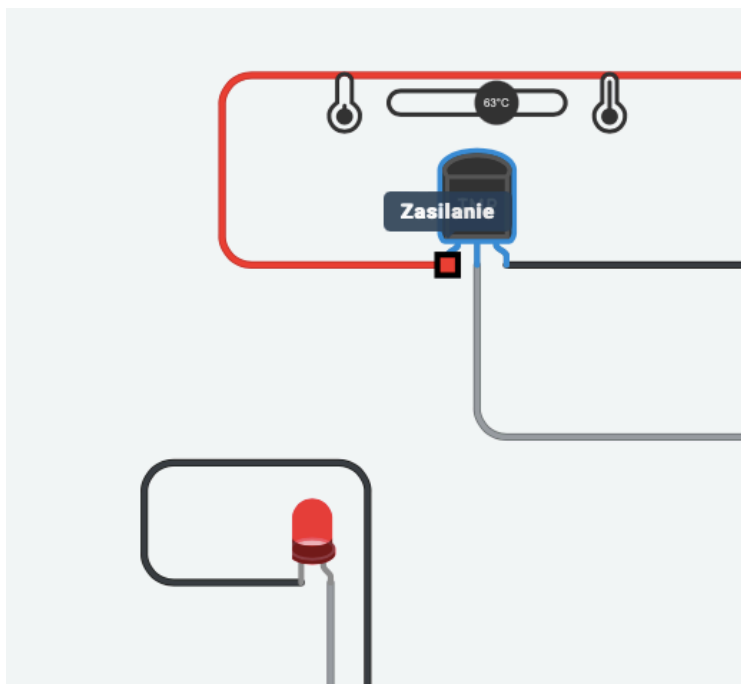
Gdy czujnik temperatury wykryje temperaturę poniżej 40 stopni Celsjusza to nie jest uruchamiana dioda.



Rys. 5 Czujnik temperatury dioda wyłączona

#### Przypadek V dla temperatury

Gdy czujnik temperatury wykryje temperaturę 40 lub więcej stopni Celsjusza to nie jest uruchamiana dioda.



Rys. 6 Czujnik temperatury dioda włączona



## 6. Rozwiązanie z opisem poszczególnych linii skryptu

### Funkcja void setup()

```
1. void setup()
2. {
3.   pinMode(1, OUTPUT);
4.   pinMode(A0, INPUT);
5.   pinMode(2, INPUT);
6.   pinMode(4, OUTPUT);
7.   pinMode(5, OUTPUT);
8.   pinMode(7, OUTPUT);
9.   pinMode(6, OUTPUT);
10.    pinMode(A1, INPUT);
11.    pinMode(9, OUTPUT);
12. }
```

Funkcja void setup() odpowiada za jednorazowe wykonanie instrukcji, ustawienie poszczególnych gniazd pinów:

- analogowy A0, czujnik gazu, **(linia 4)**
- analogowy A1, czujnik temperatury **(linia 10)**,
- wejście cyfrowe 2, przycisk (symulacja otwarcia, zamknięcia okna) **(linia 5)**,
- wyjścia cyfrowe 4,5,7 są od diody LED RGB (3 stopniowy monitor jakości powietrza) **(linie 6,7,8)**,
- wyjście cyfrowe 6 to buzzer **(linia 9)**,
- wyjście cyfrowe 9 to czerwona dioda LED (sygnalizacja zbyt dużej temperatury 40 lub więcej stopni Celsjusza) **(linia 11)**.

### Funkcja void loop()

```
1. void loop()
2. {
3.   if (analogRead(A0) <= 20 && digitalRead(2) == 0) {
4.     digitalWrite(4, HIGH);
5.     digitalWrite(5, LOW);
6.     digitalWrite(7, LOW);
7.   } else {
8.     if (analogRead(A0) >= 21 && analogRead(A0) <= 29) {
9.       digitalWrite(5, HIGH);
10.      digitalWrite(4, LOW);
11.      digitalWrite(7, LOW);
12.    }
13.    if (analogRead(A0) > 29) {
14.      digitalWrite(4, LOW);
15.      digitalWrite(5, LOW);
16.      digitalWrite(7, HIGH);
17.      tone(6, 523, 1000); // play tone 60 (C5 = 523 Hz)
18.    }
19.  }
```

Funkcja `void loop()` to pętla nieskończona, odpowiada za logikę wyświetlania alarmów w zależności od wartości odczytanych z czujnika temperatury oraz czujnika zanieczyszczeń.

- Pierwszy `if` odpowiada za ustawienie diody LED RGB na kolor zielony jeżeli czujnik zanieczyszczeń przyjmuje wartości mniejsze lub równa 20 i jednocześnie wejście cyfrowe 2 jest równe 0 to dioda świeci na zielono. **(linie 3-6)**.
- Następnie występuje dalsza część instrukcji warunkowej `if else`, która sprawdza czy wartość z czujnika zanieczyszczeń jest w przedziale od 21 włącznie do 29 włącznie to wtedy ustawia diodę LED RGB na kolor niebieski. **(linie 7-12)**.
- Potem gdy wartość z czujnika zanieczyszczeń jest większa lub równa 29 to ustawia diodę LED RGB na kolor czerwony oraz uruchamia buzzer. **(linie 13-18)**.

```
1. if ((-40 + 0.488155 * (analogRead(A1) - 20)) > 40) {  
2.     digitalWrite(9, HIGH);  
3. } else {  
4.     digitalWrite(9, LOW);  
5. }  
6.     delay(10);
```

Ostatnia instrukcja warunkowa znajduje się poza główną instrukcją warunkową, w funkcji `void loop()`. Instrukcja ta odpowiada za logikę obsługi czujnika temperatury [TMP36] oraz czerwonej diody LED.

- Instrukcja warunkowa dokonuje obliczeń, z czujnika na stopnie Celsjusza i dla wartości większych niż 40 uruchamiana jest dioda czerwona dioda LED, ostrzegająca przed zbyt wysoką temperaturą. **(linie 1-2)**.
- W przeciwnym wypadku dioda ustawiona jest na pozycji wyłączonej. **(linie 3-4)**.
- Opóźnienie w celu poprawienia wydajności symulacji. **(linia 6)**.

## **7. Podsumowanie i wnioski**

### **Podsumowanie**

Arduino to narzędzie zarówno dla amatorów jak i profesjonalistów. Dzięki platformie tinkercad, która pozwala symulować online stworzony wirtualny schemat. Posiada rozbudowany zestaw komponentów, dający możliwość stworzenia prostych jak i bardziej skomplikowanych projektów. Jedną z największych zalet tej platformy jest to, że przed fizycznym zbudowaniem układu Arduino można sprawdzić i przetestować działanie wirtualnego układu, pozwala to na uniknięcie przepalenia komponentów. Program sam ostrzega, że w przypadku użycia rezystora o zbyt małym oporze istnieje możliwość spalania diody.

### **Wnioski**

W projekcie występuje zależność między poziomem zanieczyszczenia a rodzajem uruchomieniem sygnałów ostrzegawczych w postaci dźwięki i koloru diody oraz zależność pomiędzy temperaturą a zapaleniem się ostrzegawczej czerwonej diody LED. W Arduino to, można zaprogramować pod własne potrzeby całą logikę obsługi dodatkowych komponentów, takich jak diody, głośniki, wyświetlacze oraz wiele innych.

## **8. Literatura**

- <https://pl.wikipedia.org/wiki/Arduino>, z dnia 22.11.2022
- <https://botland.com.pl/czujniki-nacisku/11152-membrana-piezoelektryczna-z-przewodami-20mm-sparkfun-sen-10293-5903351249775.html>, z dnia 22.11.2022
- <https://pl.wikipedia.org/wiki/Rezystor>, z dnia 22.11.2022
- <https://physics.uwb.edu.pl/wf/fi-bot/?p=1184>, z dnia 22.11.2022
- <https://forbot.pl/blog/kurs-arduino-podstawy-programowania-porty-io-id3648>, z dnia 22.11.2022
- <https://www.arduino.cc/en/Guide>, z dnia 22.11.2022