



# Bezprzewodowy rozproszony system pomiaru warunków środowiskowych

inż. Piotr Gajcy

WEL19EG1N4

Studia II stopnia – niestacjonarne

Promotor pracy:  
ppłk dr inż. Tadeusz Sondej

# Plan prezentacji

- Sensory do pomiaru warunków środowiskowych
- Sposoby komunikacji w rozproszonych systemach pomiarowych
- Projekt i wykonanie modelu systemu
- Wykonanie badań testowych
- Podsumowanie

# Sensory do pomiaru warunków środowiskowych

- Czujniki ciśnienia**

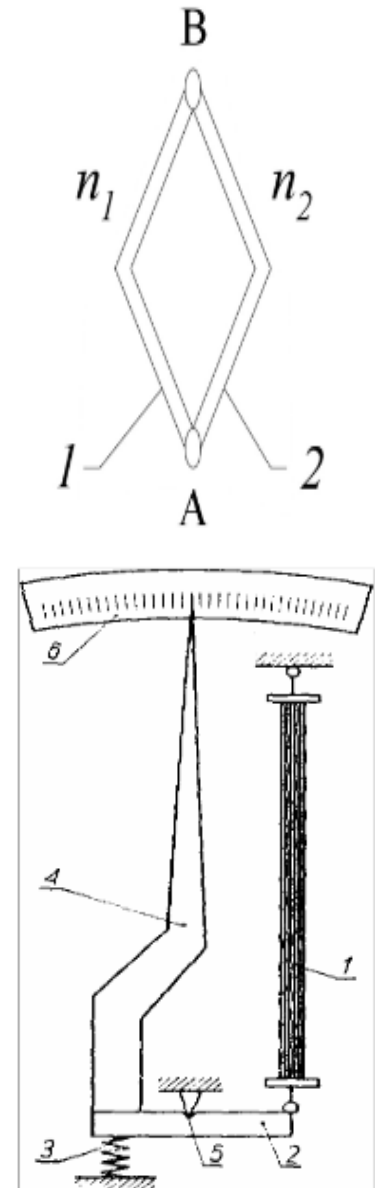
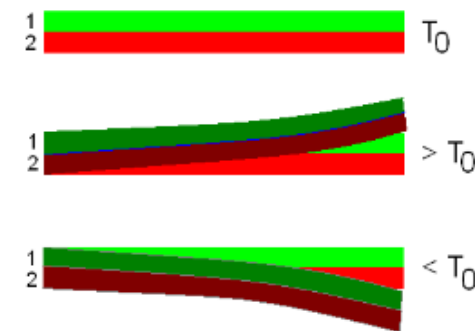
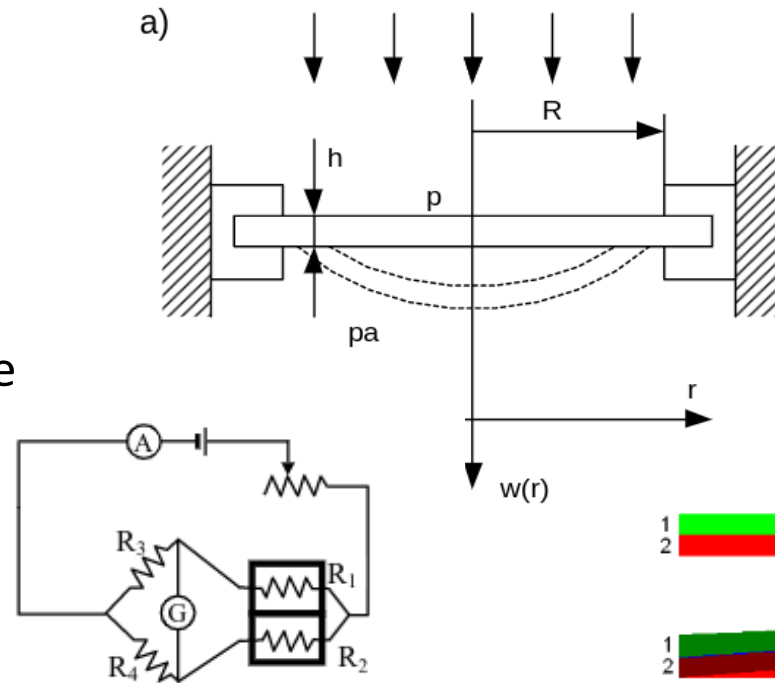
- Sprężysta membrana

- Czujniki temperatury**

- Czujniki rezystancyjne
- Czujniki półprzewodnikowe
- Termopary
- Czujniki bimetaliczne

- Czujniki wilgotności**

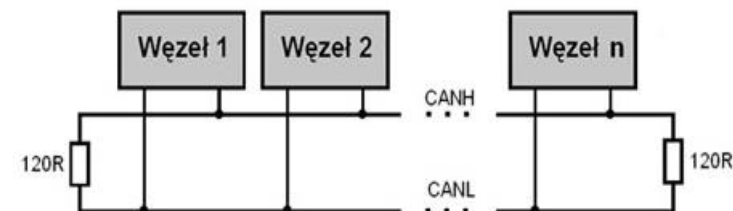
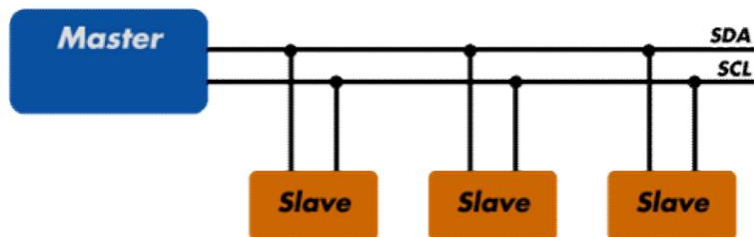
- Włosowe
- Oparte na zasadzie przewodnictwa cieplnego
- Pojemnościowe



# Sposoby komunikacji w rozproszonych systemach pomiarowych

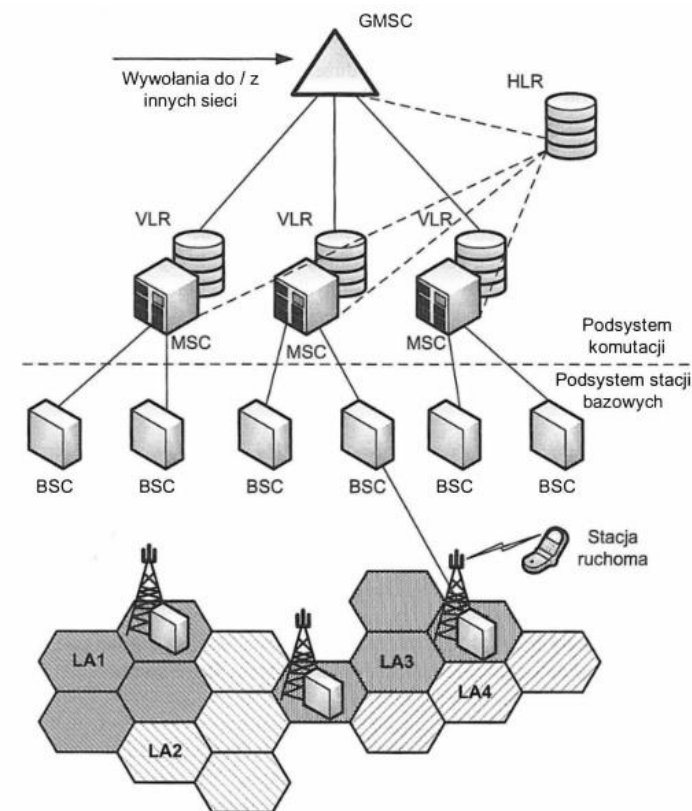
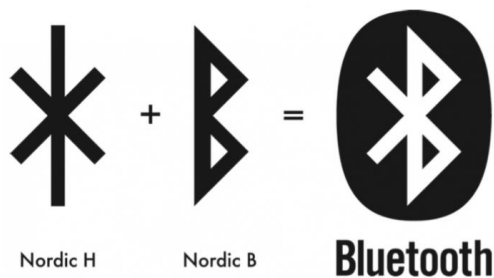
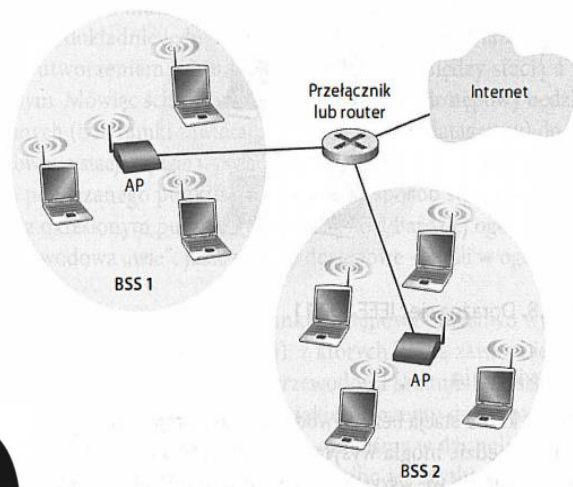
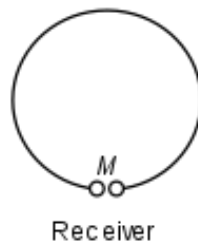
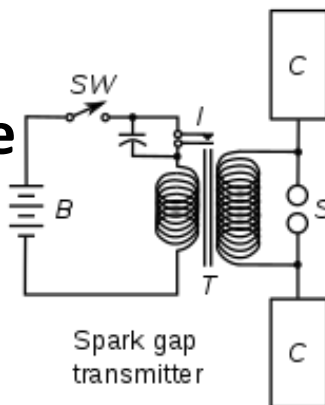
## Przewodowe

- RS
- CAN
- I2C



## Bezprzewodowe

- IRDA
- Radio
- ZigBee
- GSM
- WiFi
- LoRa
- Bluetooth

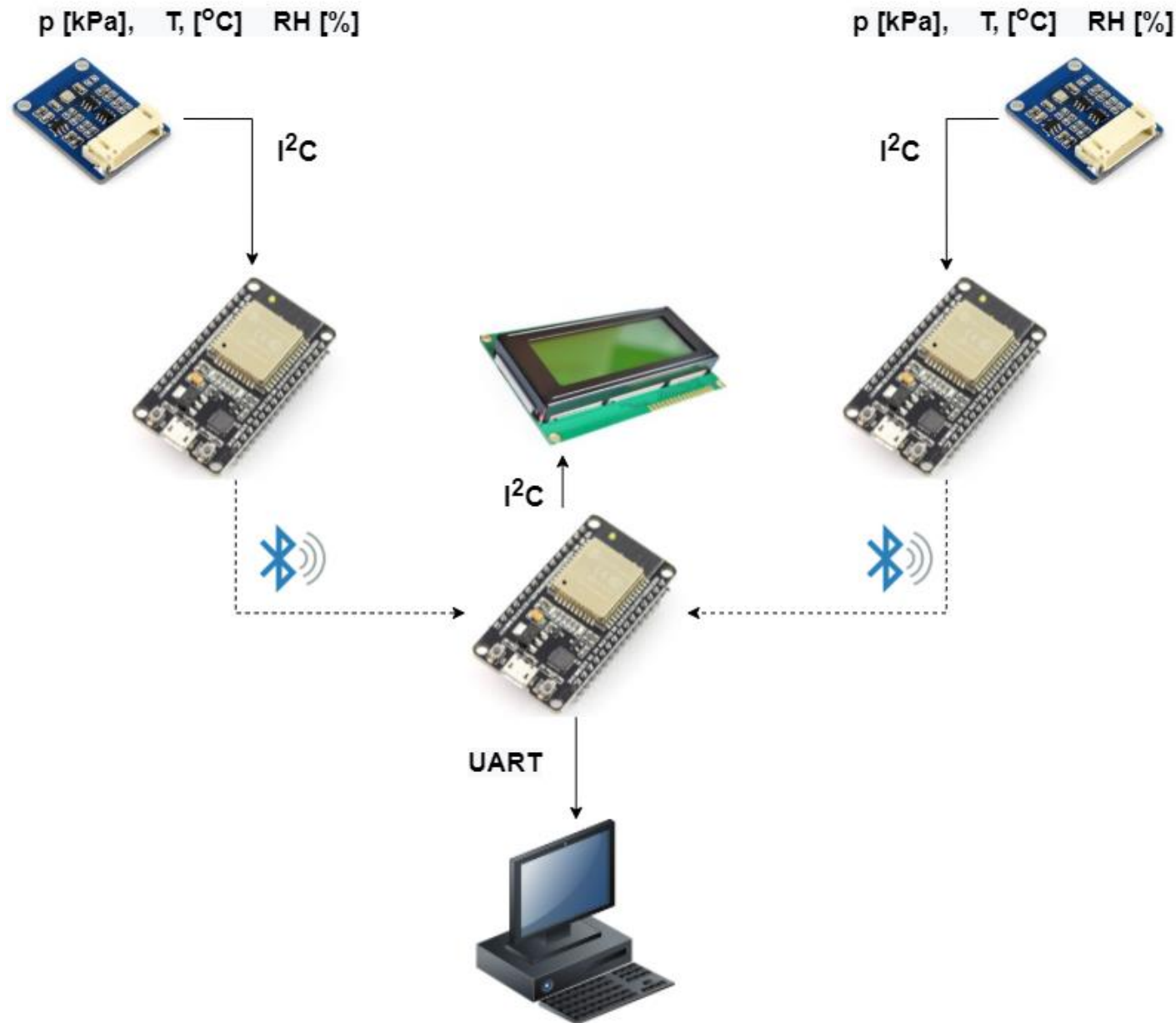


# Ogólne założenia systemu

- Pomiar ciśnienia, temperatury oraz wilgotności
- Bezprzewodowe przesyłanie danych
  - Protokół Bluetooth Low Energy
- Przesyłanie zebranych danych do komputera poprzez UART
- Prezentacja zebranych danych na wyświetlaczu LCD

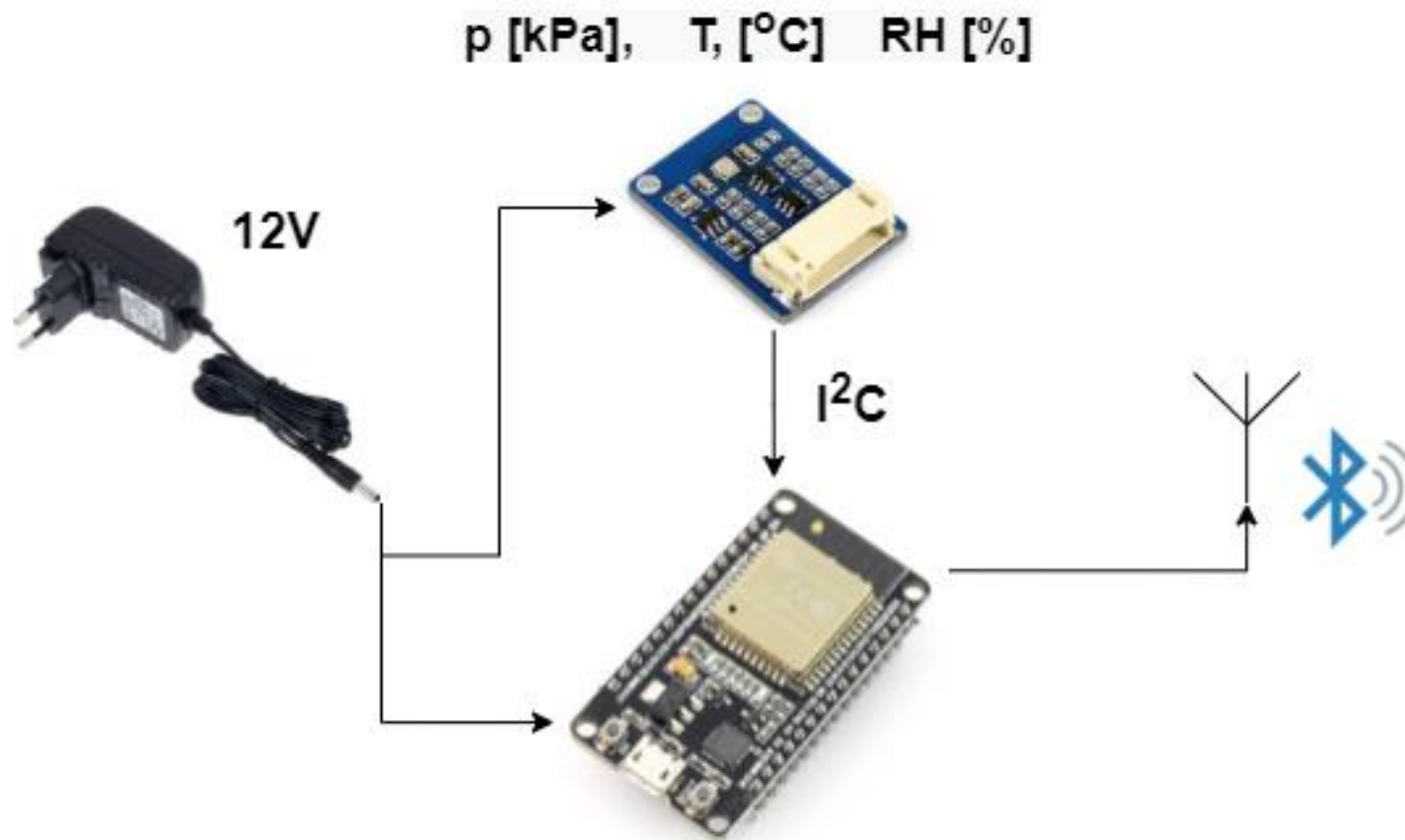
# Projekt i wykonanie modelu systemu pomiarowego

## Schemat ogólny



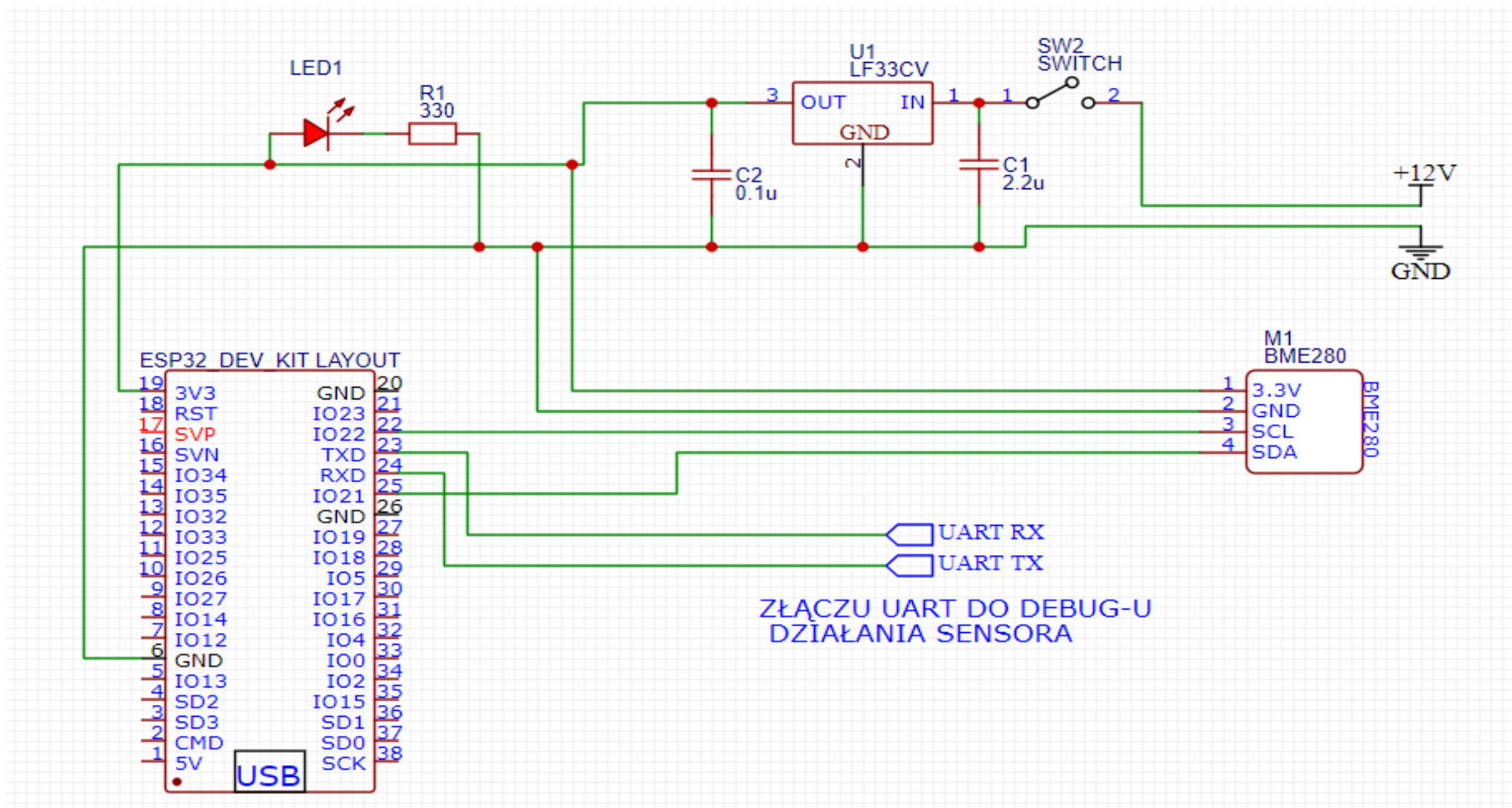
# Projekt i wykonanie modelu systemu pomiarowego

## Moduł sensora – schemat



# Projekt i wykonanie modelu systemu pomiarowego

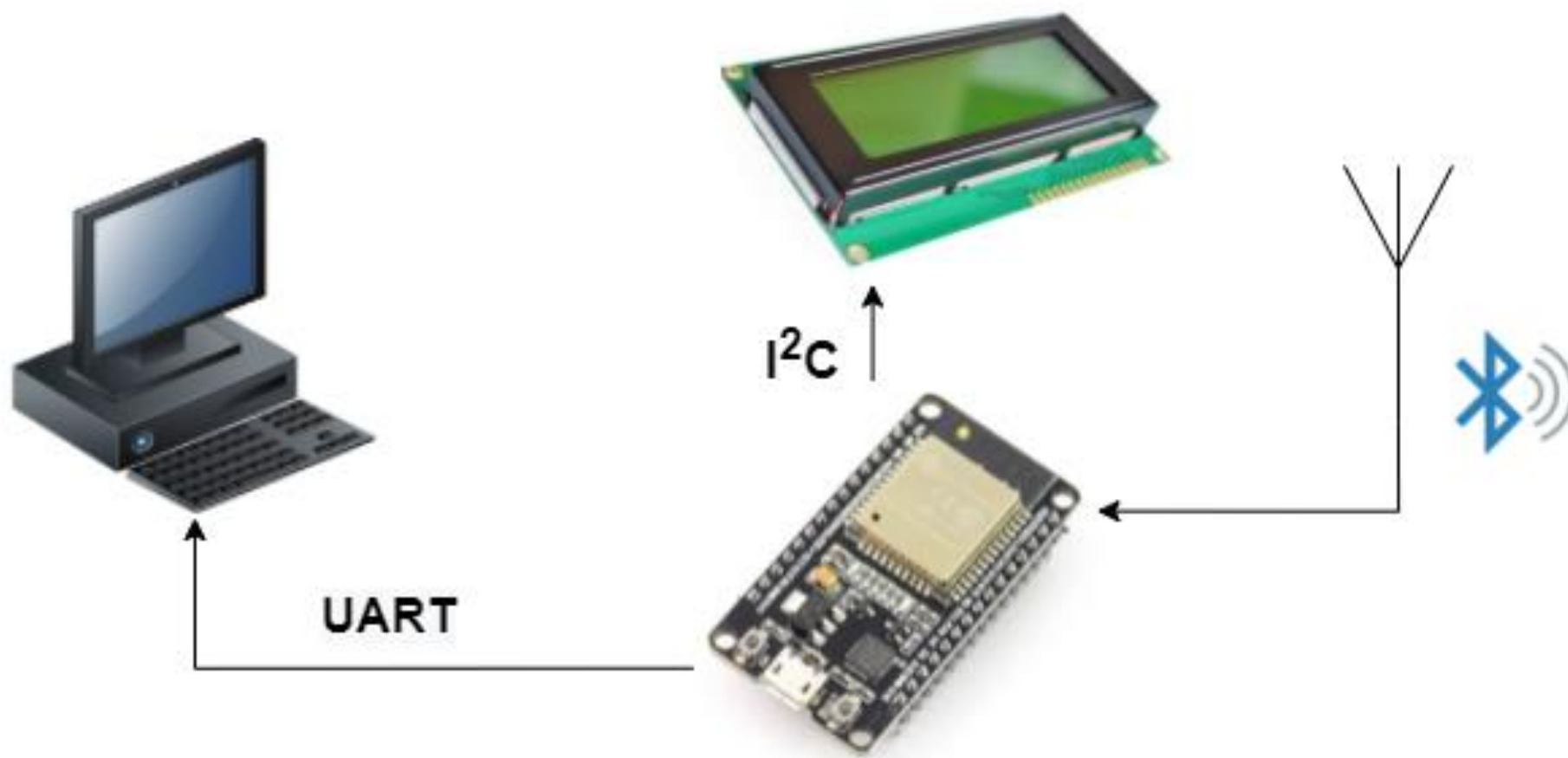
## Moduł sensora – schemat elektryczny





# Projekt i wykonanie modelu systemu pomiarowego

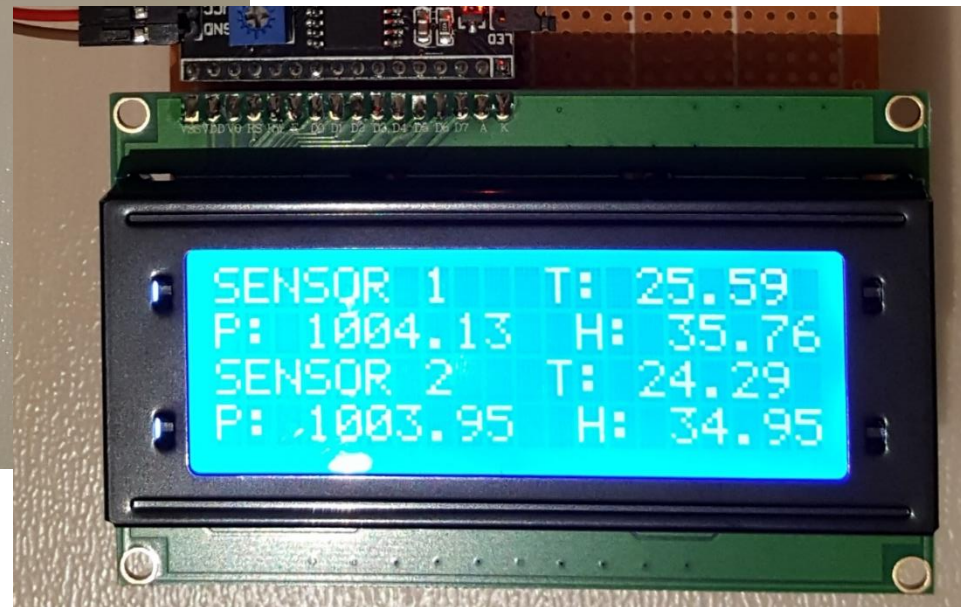
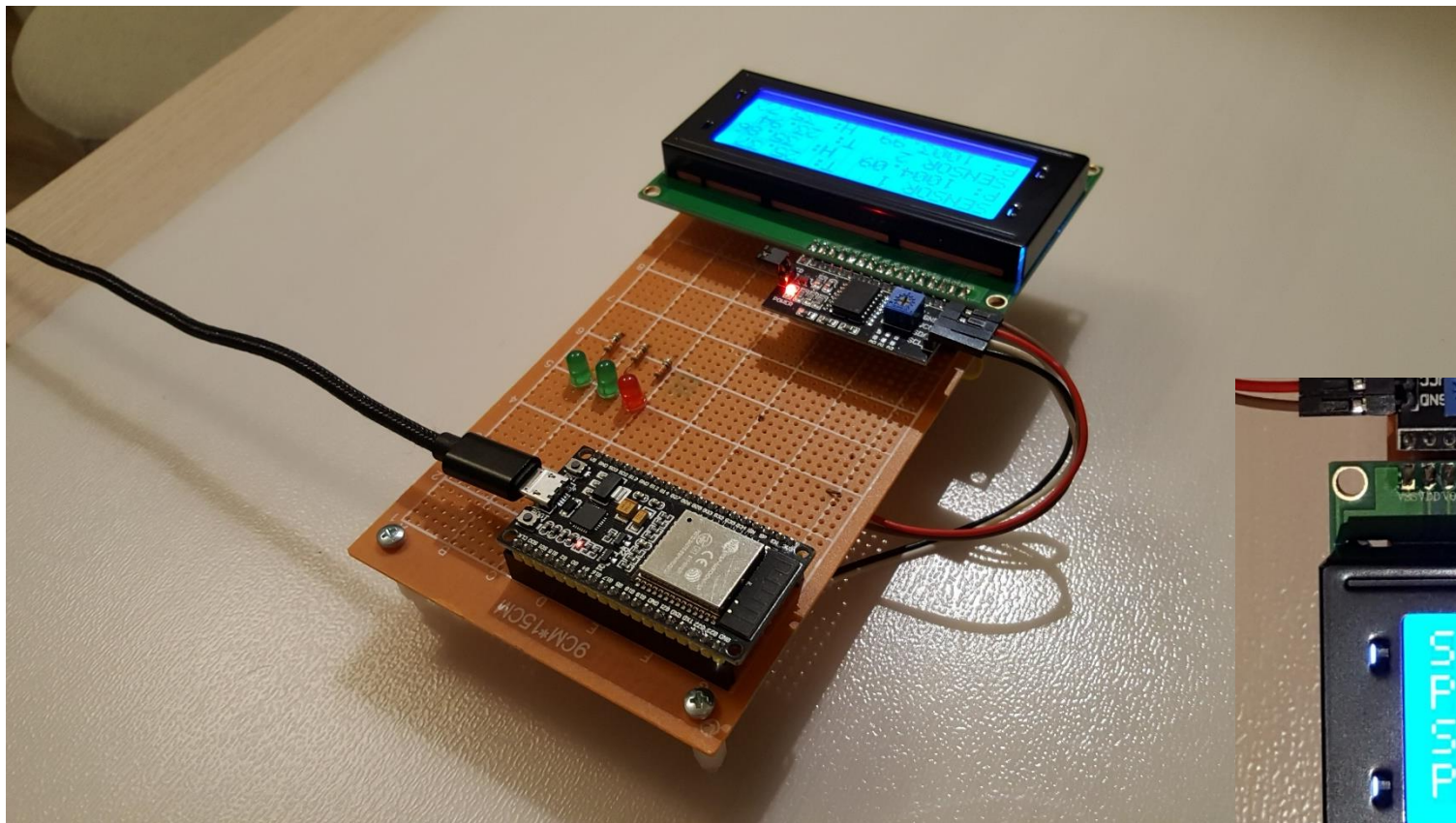
## Moduł centralny – schemat





# Projekt i wykonanie modelu systemu pomiarowego

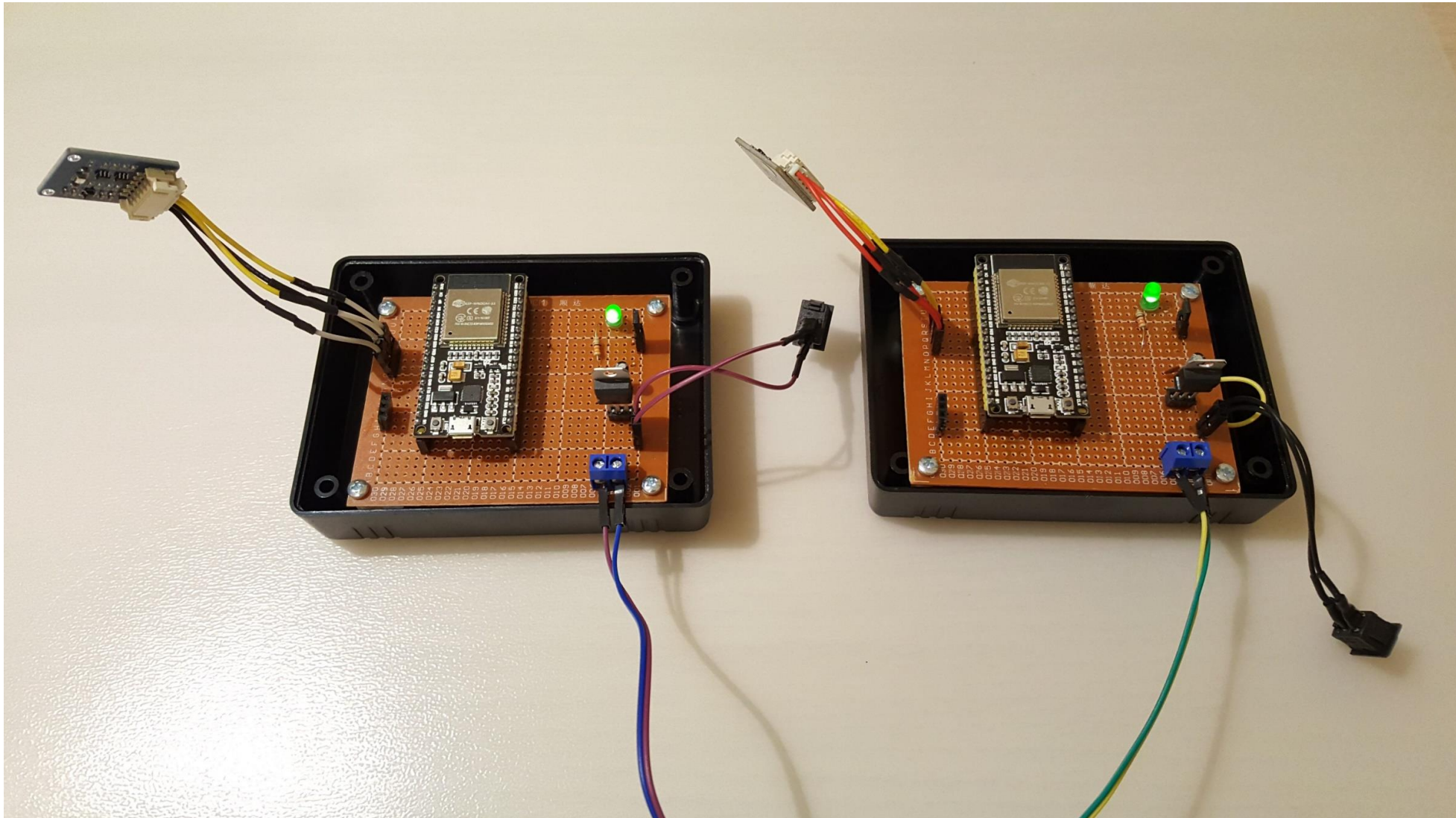
## Realizacja modułu centralki





# Projekt i wykonanie modelu systemu pomiarowego

## Realizacja modułów sensorów



# Projekt i wykonanie modelu systemu pomiarowego

## Główne moduły

### BME280:

- Napięcie pracy: 3,3 V lub 5 V
- Interfejs: I2C
- Temperatura:
  - zakres pomiarowy: od -40 °C do 85 °
  - rozdzielczość: 0,01 °C
  - dokładność:  $\pm 1$  °C
- Wilgotność:
  - zakres pomiarowy: od 0 % do 100 % RH
  - rozdzielczość: 0,008 % RH
  - dokładność:  $\pm 3$  % RH
  - czas odpowiedzi: 1 s
- Ciśnienie:
  - zakres pomiarowy: od 300 hPa do 1100 hPa
  - rozdzielczość: 0,18 Pa
  - dokładność:  $\pm 1$  hPa
- Pobór prądu: do 0,1 mA @ 1 Hz
- Wymiary: 27 x 20 mm



### ESP32:

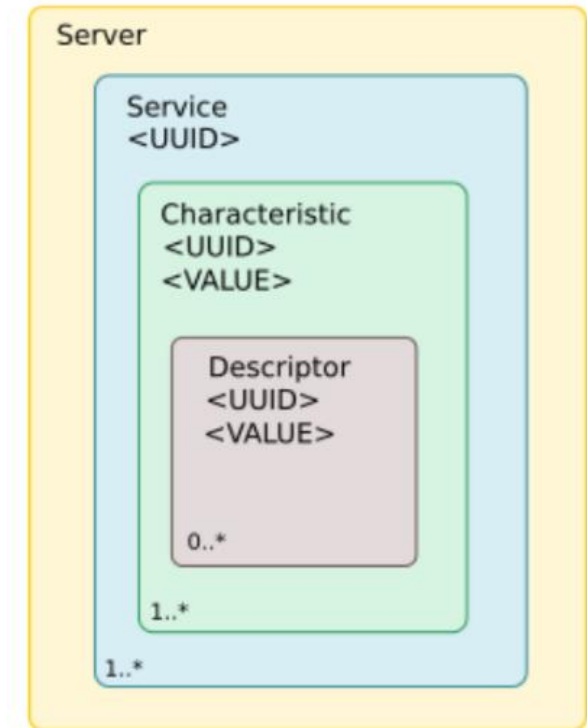
- Chip ESP32
- Pamięć SRAM 520 KB
- Pamięć Flash: 4 MB
- Wbudowany moduł Bluetooth BLE
- 30 wyprowadzeń GPIO w tym:
  - UART
  - I2C
- Wymiary: 55 x 28 x 8 mm



# Projekt i wykonanie modelu systemu pomiarowego

## Oprogramowanie

- Centralka oraz sensory
  - Arduino IDE
  - C/C++
  - Biblioteki ESP32 BLE -> obsługa BLE
  - Biblioteki Adafruit BME280 -> obsługa czujnika BME280
- Oprogramowanie analizujące
  - SerialPlot -> odbiór i rejestracja danych z centralki
  - Matlab -> parametry statystyczne, wykresy



## Centralka

```
if (connection_status[i] == true && pRemoteCharacteristics[i]->canRead()) {  
    digitalWrite(led_gpio[i], HIGH);  
    value = pRemoteCharacteristics[i]->readValue();  
    char * strtokIndx;  
    strtokIndx = strtok((char *)value.c_str(), ",");  
    temperature = atof(strtokIndx);  
    strtokIndx = strtok(NULL, ",");  
    pressure = atof(strtokIndx);  
    strtokIndx = strtok(NULL, ",");  
    humidity = atof(strtokIndx);  
    Serial.print(i); Serial.print(", "); Serial.print(temperature); Serial.print(", ");  
        Serial.print(pressure); Serial.print(", "); Serial.println(humidity);  
    lcd.setCursor(0, 2*i);  
    lcd.print("SENSOR "); lcd.print(i+1); lcd.print("    T: "); lcd.print(temperature);  
    lcd.setCursor(0, 2*i+1);  
    lcd.print("P: "); lcd.print(pressure); lcd.print("    H: "); lcd.print(humidity);  
}
```

## Sensor

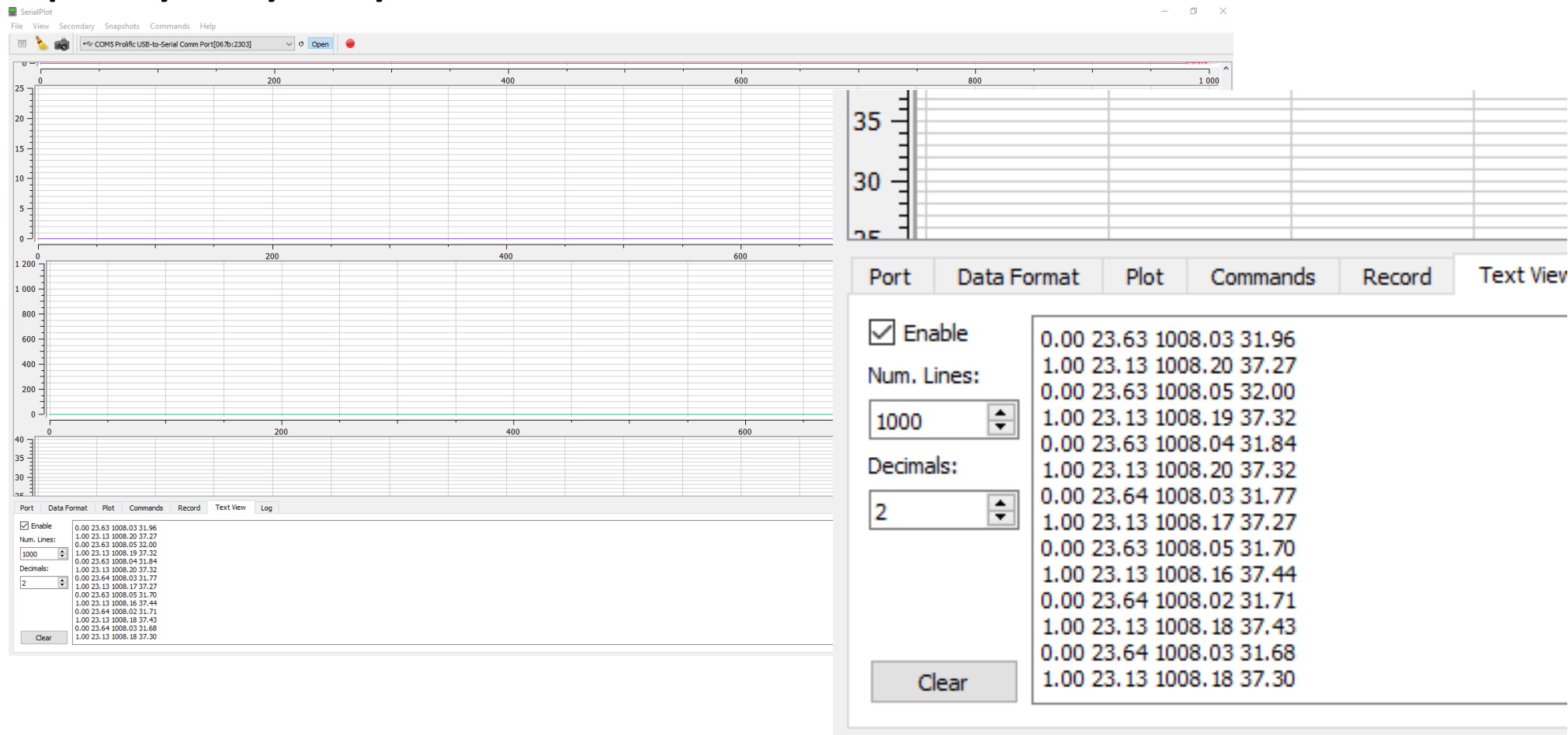
```
float temperature, pressure, humidity;
BLEDevice::init("Sensor_1");
BLEServer *pServer = BLEDevice::createServer();
BLEService *pService = pServer->createService(SERVICE_UUID);
BLECharacteristic *pCharacteristic = pService->createCharacteristic(CHARACTERISTIC_UUID,
                                                                    BLECharacteristic::PROPERTY_READ |
                                                                    BLECharacteristic::PROPERTY_WRITE
                                                                    );
pCharacteristic->setValue("test sensora 1");
pService->start();
while (1) {
    temperature = bme.readTemperature();
    pressure=bme.readPressure() / 100.0F;
    humidity=bme.readHumidity();
    char buffer_data[24];
    snprintf(buf2, 24, "%.2f, %.2f, %.2f\r\n", temperature, pressure, humidity);
    pCharacteristic->setValue(buffer_data);
    delay(delayTime);
}
delay(delayTime);
}
```





# Projekt i wykonanie modelu systemu pomiarowego

## Poprawny odczyt danych:



## Analiza danych

```
function data_analyser
```

```
format bank %format - dwa miejsca dziesiętne  
disp("Welcome!")
```

```
%-----Pobranie danych-----
```

```
filename="pomiar.txt"; % plik z danymi
```

```
if isfile(filename)
```

```
    % File exists.
```

```
    disp("Znaleziono plik... Analizuję...");
```

```
else
```

```
    warningMessage = sprintf('Warning: file does not exist:\n%s', filename);
```

```
    uiwait(msgbox(warningMessage));
```

```
    return;
```

```
end
```

```
B=readmatrix(filename); % wczytanie pliku
```

```
[rows, ~]=size(B);%rozmiar wczytanego pliku
```

```
sensor1_data=zeros(rows,3); % alokacja pamięci na dane z serwerów
```

```
indexS1=1; % indeks pomocniczy do rozdzielania danych
```

```
sensor2_data=zeros(rows,3);
```

```
indexS2=1;
```

```
%-----Rozdzielenie danych-----
```

```
for i = 1:1:rows
```

```
    if B(i,1) == 0.00
```

```
        sensor1_data(indexS1, 1)= B(i, 2);
```

```
        sensor1_data(indexS1, 2)= B(i, 3);
```

```
        sensor1_data(indexS1, 3)= B(i, 4);
```

```
        indexS1=indexS1+1;
```

```
    elseif B(i,1) == 1.00
```

```
        sensor2_data(indexS2, 1)= B(i, 2);
```

```
        sensor2_data(indexS2, 2)= B(i, 3);
```

```
        sensor2_data(indexS2, 3)= B(i, 4);
```

```
        indexS2=indexS2+1;
```

```
    else
```

```
        disp("Błąd parsowania linii:\n")
```

```
        disp(B(i, :));
```

```
end
```

```
end
```

```
sensor1_data( ~any(sensor1_data,2), : ) = []; %usuwanie rzędów wypełnionych samymi zerami
```

```
sensor2_data( ~any(sensor2_data,2), : ) = []; %
```

## Najważniejsze badania dotyczyły pomiarów rzeczywistych warunków środowiskowych

- Data pomiarów: 01.05.2021
- 12:00 a 23:00 (11 godzin)
- Przykład uzyskanych danych:

1.000, 24.630, 1002.560, 46.230

0.000, 23.240, 1002.510, 47.240

1.000, 24.630, 1002.590, 46.590

0.000, 23.250, 1002.530, 47.300

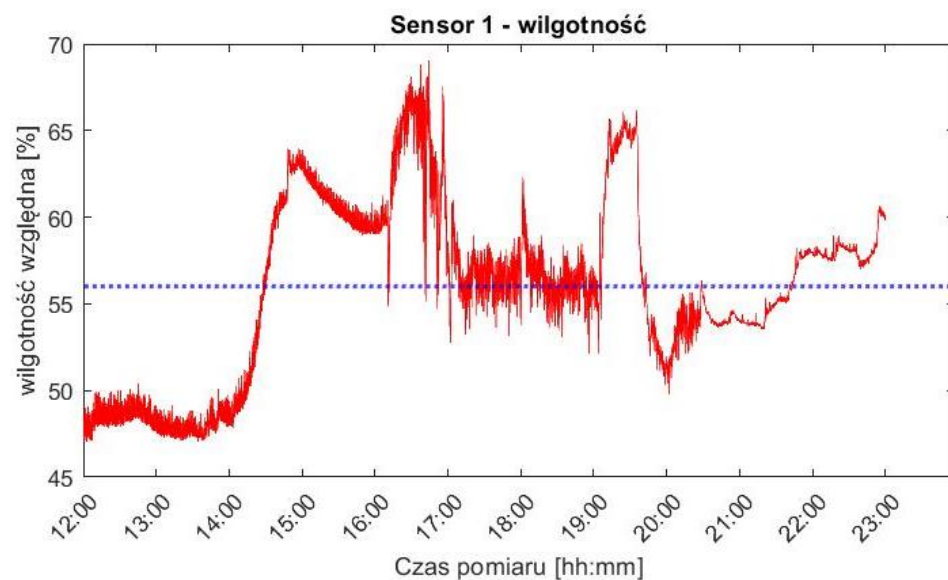
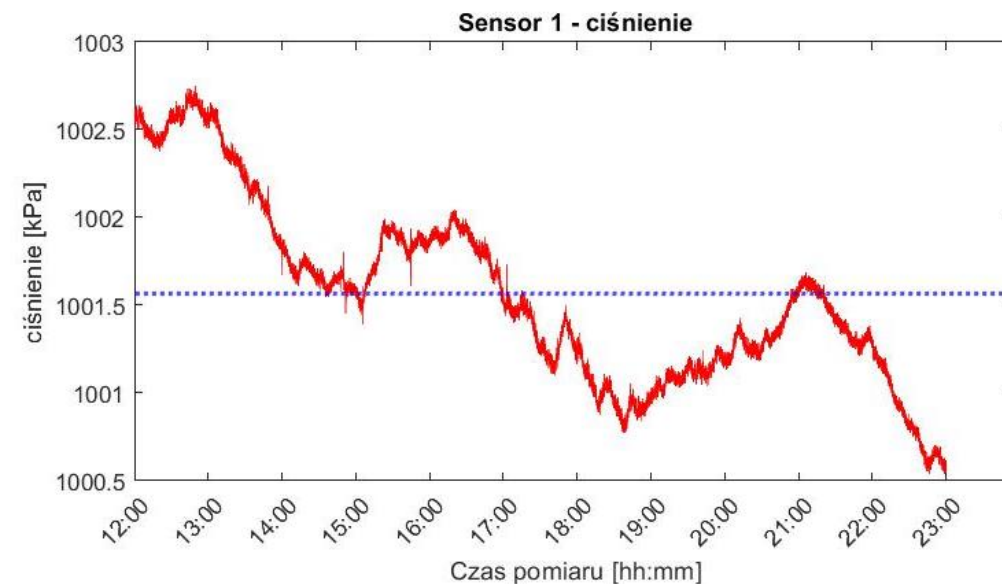
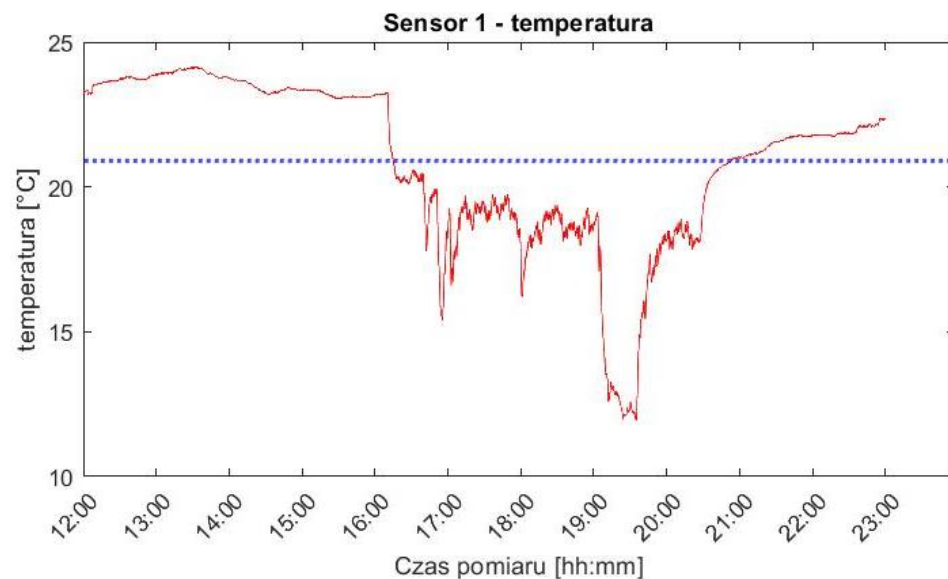
1.000, 24.640, 1002.590, 48.080

0.000, 23.250, 1002.550, 47.370

...

- odebrano 82818 linii pomiarów (po 41409 na sensor),
- W sumie 2,29 MB (czyli 2 401 753 bajtów) uzyskanych danych.

# Wykonanie badań testowych.



Średnia temperatura: 20.89°C

Temperatura zawiera się w przedziale: 11.94°C do 24.14°C

Średnie ciśnienie: 1001.56 kPa

Ciśnienie zawiera się w przedziale: 1000.54 kPa do 1002.75 kPa

Średnia wilgotność: 55.98 %

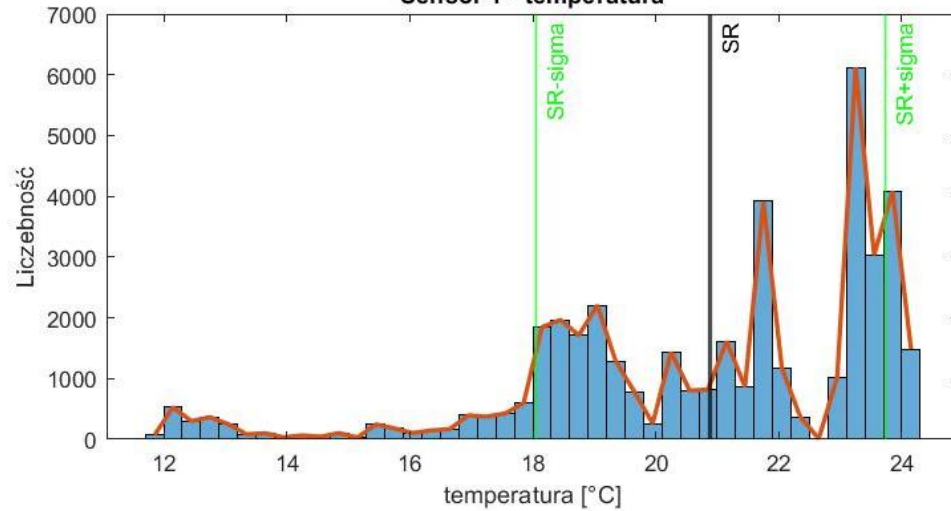
Wilgotność zawiera się w przedziale: 47.04% do 69.05%

Odchylenie standardowe:

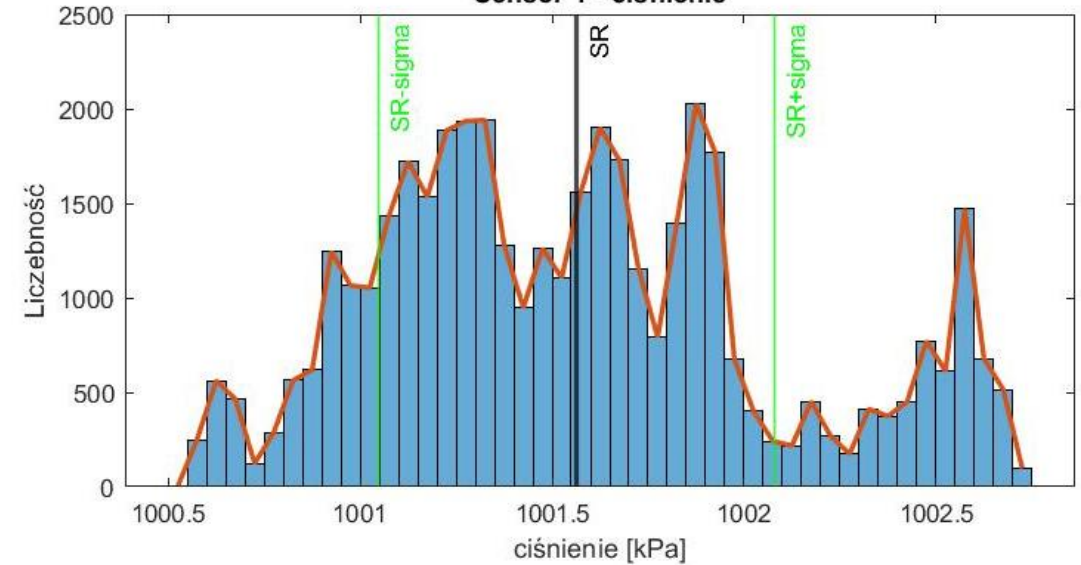
Temperatura: 2.84°C Ciśnienie: 0.52 kPa Wilgotność: 5.11%

# Wykonanie badań testowych.

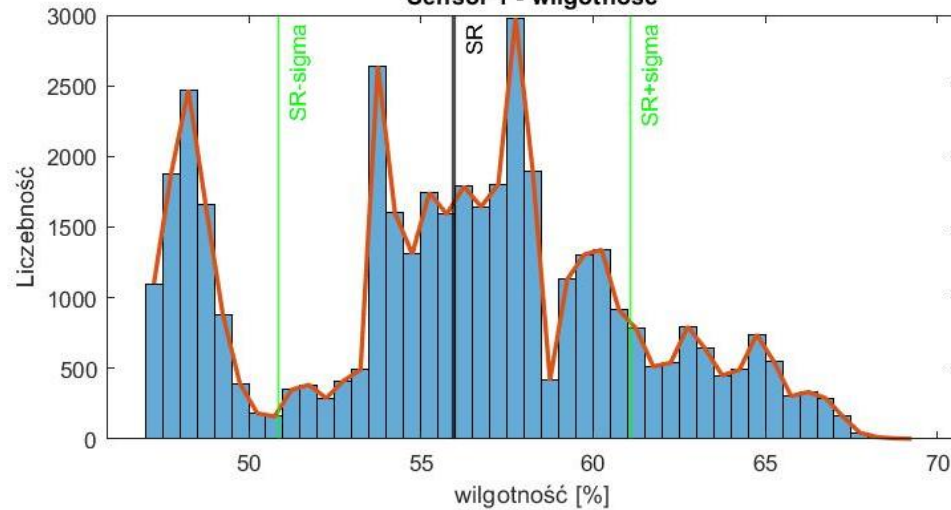
Sensor 1 - temperatura



Sensor 1 - ciśnienie



Sensor 1 - wilgotność

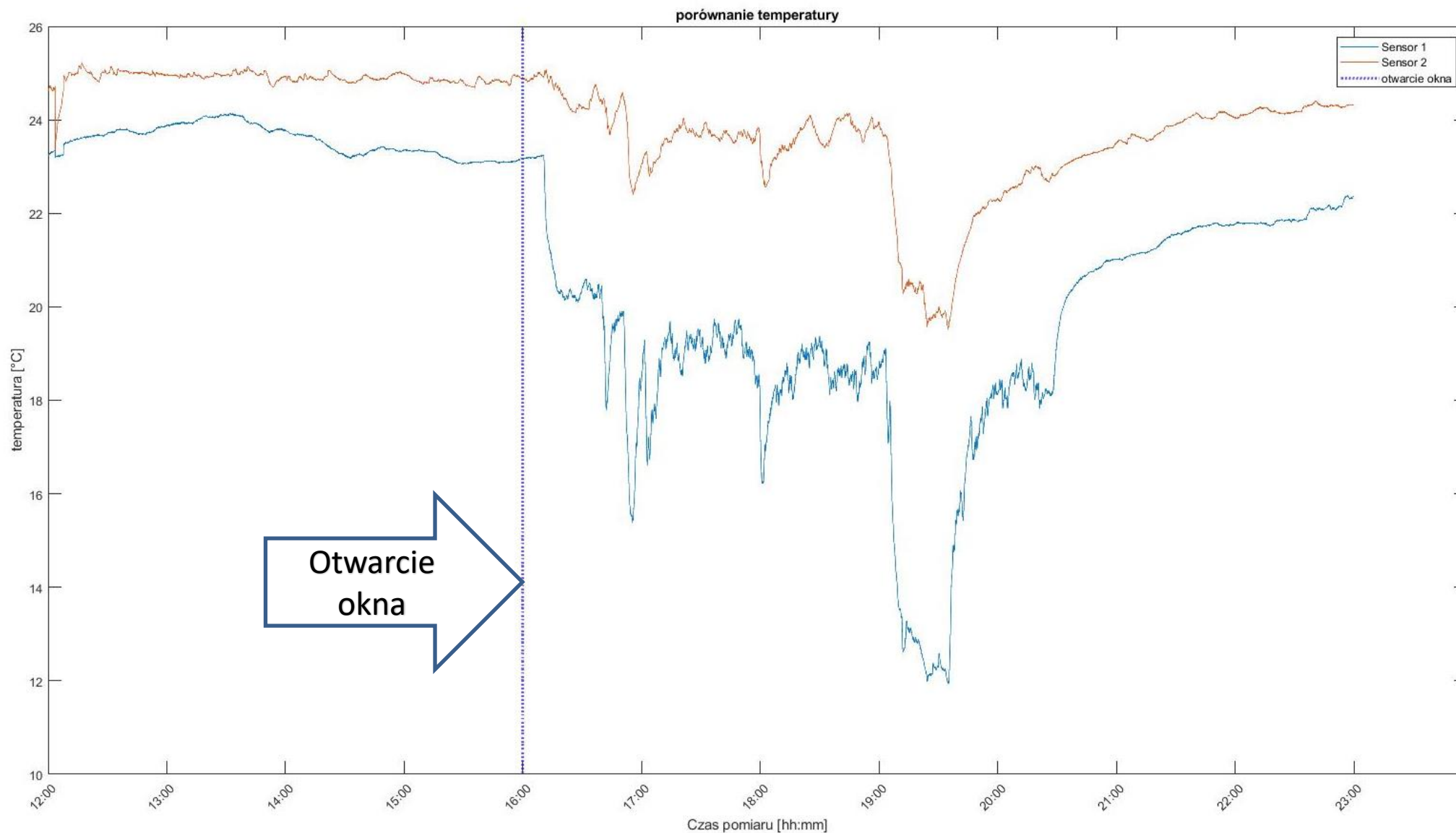


Temperatura:  $(20.89 \pm 2.84)^{\circ}\text{C}$

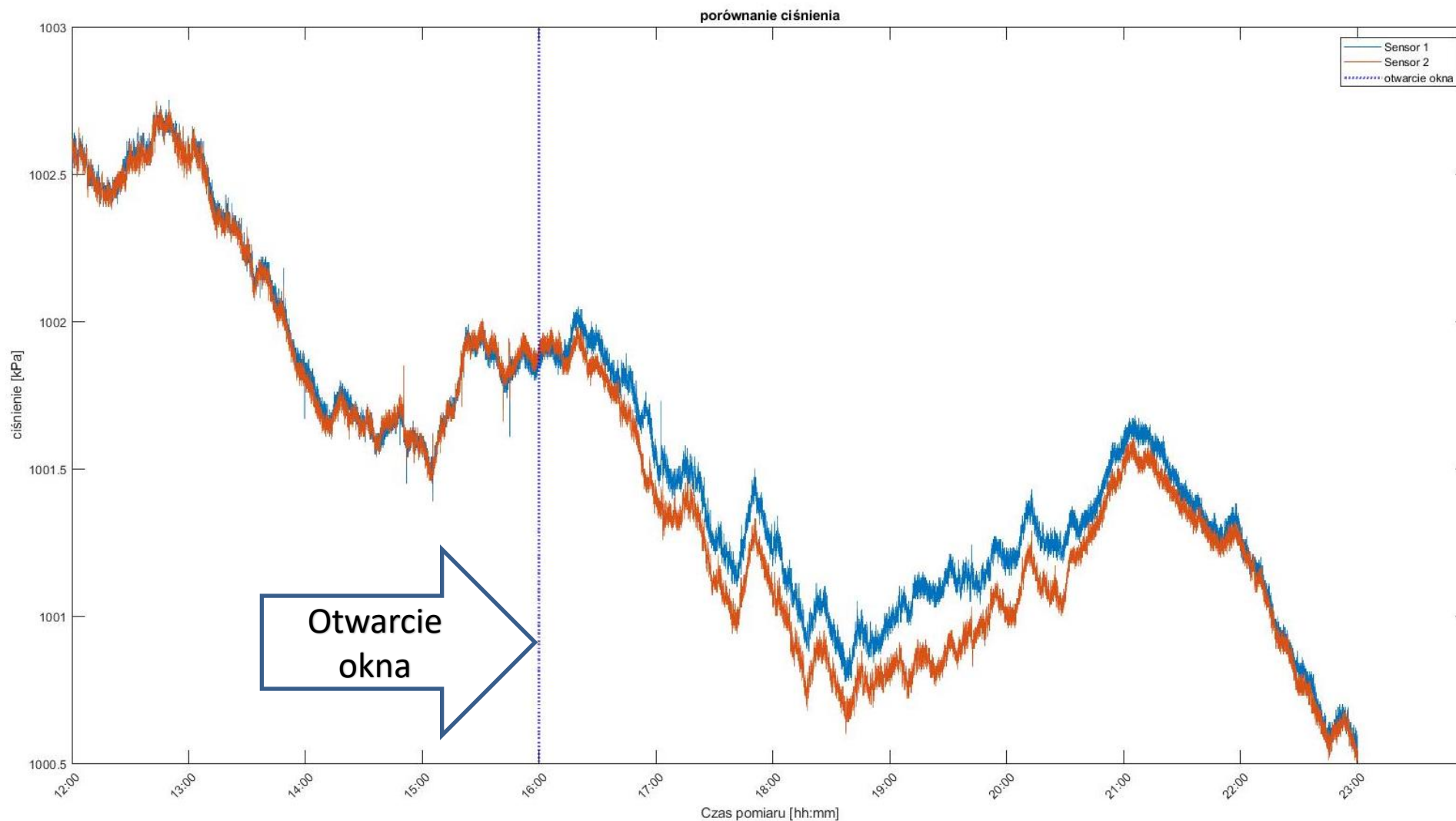
Ciśnienie:  $(1001.56 \pm 0.52) \text{ kPa}$

Wilgotność:  $(55.98 \pm 5.11) \%$

# Opracowanie wniosków. Redakcja tekstu pracy.

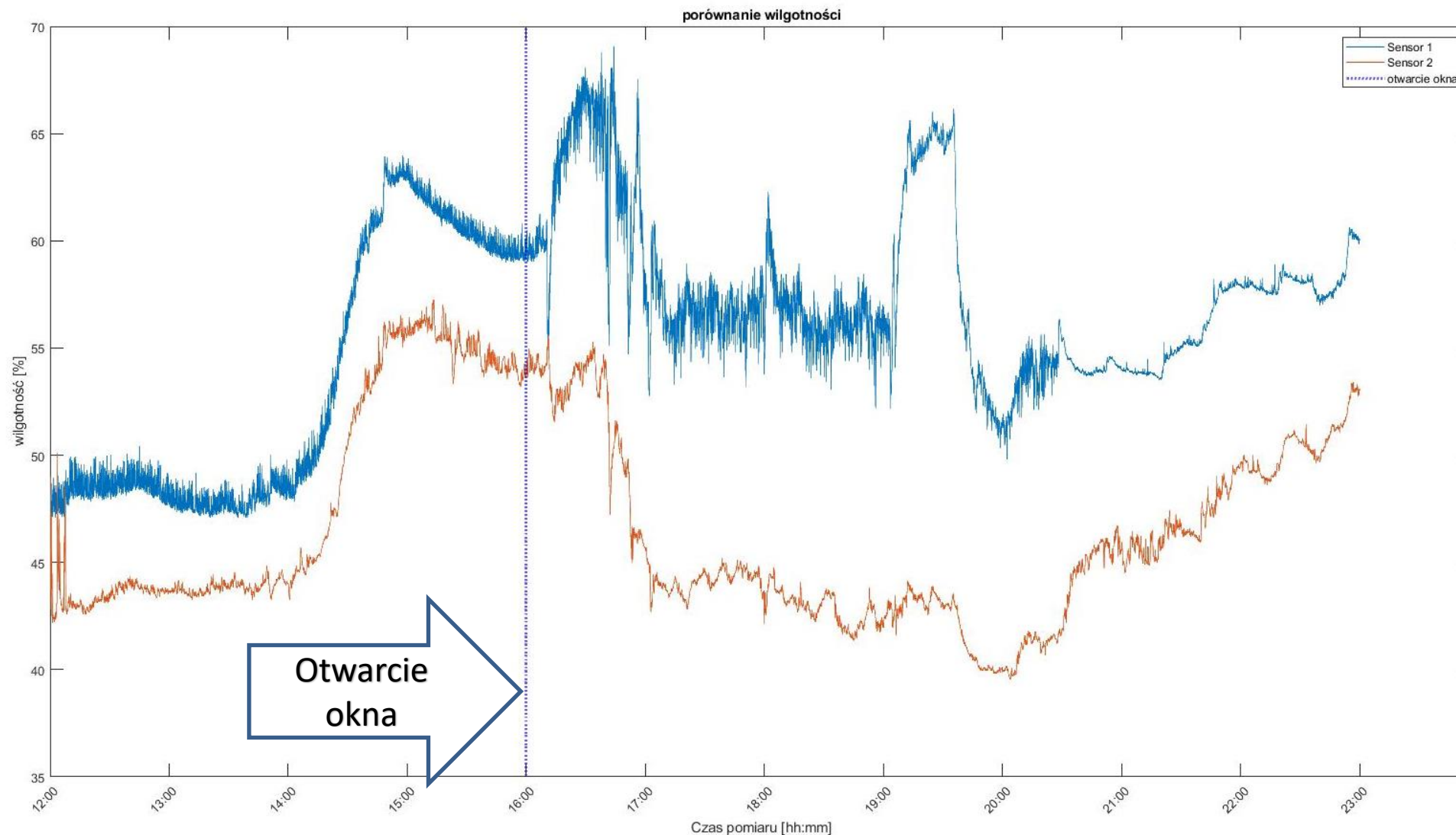


# Opracowanie wniosków. Redakcja tekstu pracy.





# Opracowanie wniosków. Redakcja tekstu pracy.





- Zgodnie z założeniem pracy zbudowano bezprzewodowy rozproszony system pomiaru warunków środowiskowych.
- Dodatkowo dokonano przeglądu protokołów komunikacyjnych, a także różnego rodzaju czujników środowiskowych.
- Zbudowane urządzenie korzysta z modułów z chipsetem ESP32, oraz z czujników BME280.
- Do zbudowania oprogramowania sterującego posłużyło Arduino IDE, wykorzystano biblioteki *ESP32 BLE* oraz *Adafruit BME280*.
- Wykonano 11-godzinne pomiary temperatury, ciśnienia oraz wilgotności przez dwa, niezależne sensory.
- Zbudowane urządzenie poprawnie pracowało przez czas trwania badań
- Udowodniono przydatność użytych komponentów do budowy zadanego systemu, a także zaprezentowano przykładową realizację akwizycji i analizy danych.



Dziękuję za uwagę.