

# Przemysł 4.0

## Laboratorium 2

### *Obsługa czujników i wizualizacja danych*

prowadzący: *Dr inż. Radosław Idzikowski*

---

## 1 Wprowadzenie

Celem laboratorium jest zapoznanie się z obsługą czujników dedykowanego modułu bazującego na popularnym mikrokontrolerze z rodziny **Raspberry Pi CM4** (*Compute Module 4*). Zadaniem będzie odczytanie wartości zmierzonych przez różne czujniki oraz ich wizualizacja. Programy będą pisane w języku programowania **PYTHON** z użyciem połączenia zdalnego (*Remote Control*) z poziomu komputera klasy PC (preferowany system **Linux**) z wykorzystaniem **MS Visual Studio Code**. Czas przewidziany na wykonanie zadania to 1 termin wraz z ocenieniem pracy. Praca będzie oceniana na bieżąco w trakcie zajęć wraz z postępem programowania i testowania kolejnych przykładów. **Po ukończeniu każdego zadania należy zwołać prowadzącego w celu zaakceptowania etapu i odnotowania postępów.**

## 2 Zadania

W ramach zajęć należy w zespołach wykonać następujące zadania:

1. Obsługa zewnętrznego czujnika temperatury.
2. Obsługa jednego z wbudowanych czujników.
3. Wizualizacja danych pomiarowych.

Za wykonanie zadania nr 1 jest ocena dostateczna, za każde kolejne zadanie jest +1 do oceny. Na ocenę bardzo dobrą (5.0) należy wykonać wszystkie trzy zadania. Po zakończeniu zajęć należy kody źródłowe napisanych programów (w szczególności zadań nr 2 i 3) zgrać i przesłać do prowadzącego. Można dołączyć zdjęcia potwierdzające poprawne działanie.

## 3 Opis zadań

### 3.1 Zadanie 1

Zadanie polega na odczytaniu temperatury z czujnika i wyświetleniu jej wartości w konsoli oraz zapaleniu odpowiednich diod **LED**. W pierwszej kolejności należy podłączyć sondę z termistorem, pokazaną na Rysunku 1, do złącza oznaczonego **NTC 10k $\Omega$**  (sondę na przewodzie należy pobrać od prowadzącego). Interfejs **SPI** do czujnika **jest już podpięty** poprzez umieszczenie 3 zworek w odpowiedniej listwie kołkowej znajdującej się w tylnej części urządzenia. Przykładowy kod do obsługi termistora został umieszczony w Listingu 1. Ponadto należy dodać prostą funkcjonalność - w zależności od temperatury powinna zmieniać się liczba zapalonych diod **LED**. W tym celu należy przyjąć odpowiednie progi, zakładając że dla najwyższej temperatury mają się palić wszystkie diody oraz dla najniższej wszystkie mają być wygaszone. Podczas realizacji zadania mogą pojawić się problemy z biblioteką **pigpio**, w takim wypadku należy wymusić jej uruchomienie jako usługi **sudo pigpiod**.



Rysunek 1: Termistor NTC 110 10kΩ z przewodem

```
1 import time
2 import pigpio # http://abyz.co.uk/rpi/pigpio/python.html
3 pi = pigpio.pi()
4 if not pi.connected:
5     exit(0)
6 sensor = pi.spi_open(1, 1000000, 0) # CE1, 1Mbps, main SPI
7 stop = time.time() + 600
8 while time.time() < stop:
9     c, d = pi.spi_read(sensor, 2)
10    if c == 2:
11        sign = (d[0] & 0x80) >> 7
12        value = (((d[0] & 0x7f) << 8) | d[1]) >> 5
13        temp = value * 0.125
14        if sign == 1:
15            temp = temp * -1
16        print("{:.2f}".format(temp))
17        time.sleep(0.25) # Don't try to read more often than 4 times a second.
18 pi.spi_close(sensor)
19 pi.stop()
```

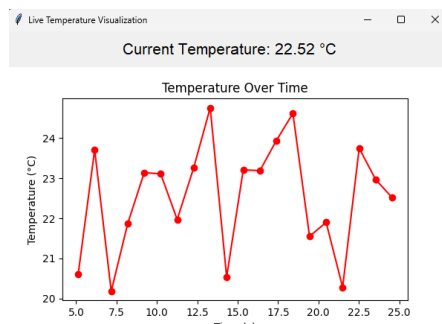
Listing 1: Obsługa termistora

## 3.2 Zadanie 2

Moduł został wyposażony zestaw czujników VL6180X, w którego skład wchodzi: ToF (*time-of-flight*) do pomiaru odległości oraz ALS (ambient light sensing) do pomiaru natężenia światła. Ponadto moduł ma możliwość świecenia. Oba czujniki znajdują się poniżej kamery. W ramach zadania należy obsłużyć jeden ze wspomnianych czujników. Czujniki są podłączone do interfejsu I<sup>2</sup>C. Przykładowy kod uruchomienia czujnika znajduje się [repozytorium github leachj](#). Oficjalna dokumentacja znajduje się na [stronie producenta](#).

## 3.3 Zadanie 3

Celem zadania jest stworzenie prostego GUI, które będzie pokazywać bieżącą wartość bieżąco z czujnika oraz 20 historycznych. Warto skorzystać z bibliotek `tkinter` oraz `matplotlib` do rysowania wykresów. Program należy **uruchomić bezpośrednio z poziomu modułu!**



Rysunek 2: Przykładowe GUI