

## Sprawozdanie z zadania laboratoryjnego z przedmiotu

### **Monitoring i sterowanie w inżynierii środowiska**

studia stacjonarne, kierunek AiR, 7 semestr, data: [20.12.2023]

#### Wykonawcy:

1. Jerzy Zawieja, 147571

#### **I. Wstęp.**

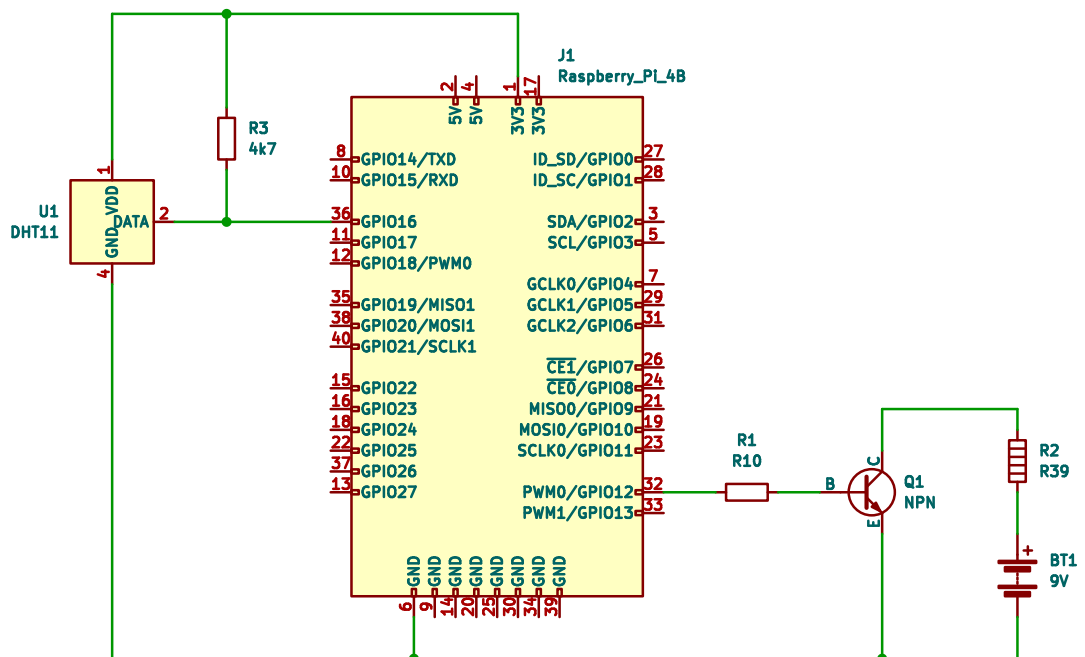
Zadanie laboratoryjne miało na celu przedstawić dowolny system monitoringu i sterowania w inżynierii środowiska. W tym celu został zaprojektowany układ elektroniczny do sterowania temperaturą na rezystorze ceramicznym. Całość została zbudowana na płycie prototypowej i połączona przewodami z mikrokontrolerem i zewnętrznym źródłem zasilania oraz niezależną aplikacją mobilną służącą do wprowadzania danych i monitoringu efektów. Jednym z celów aplikacji było też założenie, że temperaturą można będzie sterować zdalnie – operując raczej w sieci WAN niż LAN.

#### **II. Opis części sprzętowej.**

Logiką układu wraz z przesyłem danych zajmuje się mikrokontroler Raspberry PI 4B, który został użyty do tego układu, ponieważ może jednocześnie zapewnić zasilanie układu, sterowane wyjście PWM, oraz połączenie internetowe do przesyłu danych.

Jako główne urządzenie wykonawcze tj. grzałkę, został wykorzystany rezystor ceramiczny 5W39Ω, który poprzez zwiększenie na nim spadku napięcia, zwiększał straty energii, jako emisję ciepła do otoczenia. Do sterowania spadkiem napięcia na rezystorze został wykorzystany układ z tranzystorem w roli wzmacniacza. Użyty został tranzystor NPN 2N2222 0.8A, z racji jego łatwej dostępności i możliwości wszechstronnego użycia. Do kolektora i emitera tranzystora zostało podłączone zewnętrzne źródło zasilania 6x1.5 V DC, (6 szeregowo połączonych baterii AA). Za sterowanie na bazie tranzystora odpowiada wyjście PWM z Raspberry Pi 4B, które posiada też podłączony rezystor 10 Ω, ograniczający maksymalny prąd na wyjściu z mikrokontrolera (maksymalny prąd wynosi ) Za pomiar temperatury odpowiada czujnik DHT11, który ma podłączony także rezystor podciągający, by ustabilizować napięcie sygnału DATA z czujnika.

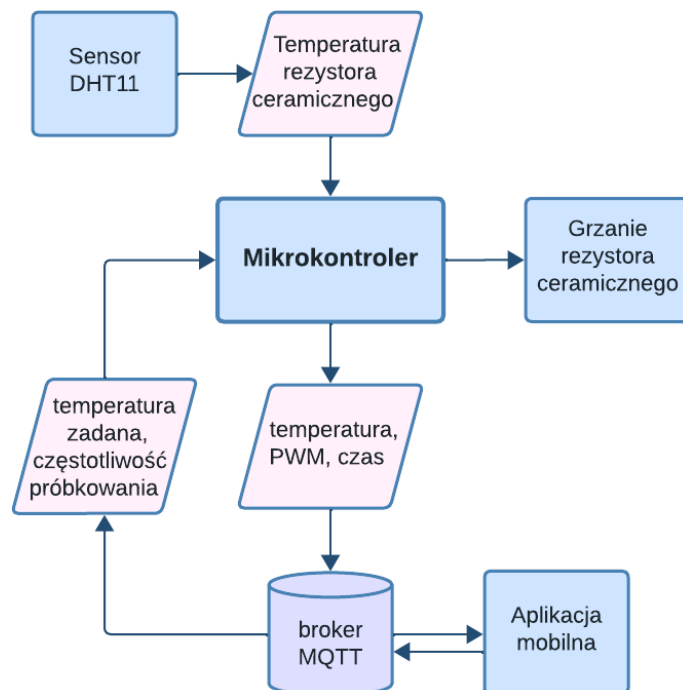
Poniżej przedstawiono schemat połączeń układu.



Rysunek 1 - schemat połączeń układu

### III. Opis części programistycznej.

Cała logika sterowania wykonywana jest przez mikrokontroler Raspberry PI. Przekazywanie danych w układzie następuje według poniższego schematu.



Rysunek 2 - diagram przepływu danych

Raspberry PI po otrzymaniu informacji o temperaturze aktualnej i zadanej korzystając ze skryptu napisanego w języku Python oblicza wypełnienie sygnału PWM za pomocą zdefiniowanego wcześniej regulatora PID.

Wartości regulatora zostały wyznaczone przez narzędzie programu Matlab do identyfikacji obiektu i doboru nastaw regulatora. Najpierw przeprowadzono pomiar odpowiedzi skokowej obiektu, po podaniu maksymalnego wypełnienia sygnału PWM (100%). Po analizie próbek sygnału wejściowego PWM jak i wyjściowej temperatury obiektu, został wyznaczony dyskretny regulator PID o parametrach:

**kp = 16.4663; ki = 0.058784; kd = 1153.1284**

Jak widać, tak naprawdę człon różniczkujący nie jest tu bardzo potrzebny, gdyż układ sam w sobie ma dużą stałą czasową.

Poniżej przedstawiono fragment kodu odpowiadający za przeliczanie kolejnych składowych dla danej wartości PID.

```
# Compute the proportional term
# Regular proportional-on-error, simply set the proportional term
self._proportional = self.Kp * error

# Compute integral and derivative terms
self._integral += self.Ki * error * dt
self._integral = _clamp(self._integral, self.output_limits) # Avoid integral windup

self._derivative = -self.Kd * d_input / dt

# Compute final output
output = self._proportional + self._integral + self._derivative
output = _clamp(output, self.output_limits)
```

*Rysunek 3 - fragment kodu z regulatorem PID*

Oprócz obliczania wartości PID wykorzystywane jest także ograniczenie sygnału wyjściowego (by maksymalne sterowanie wynosiło 100% wypełnienia sygnału PWM, a nie więcej).

Do obsługi pinów raspberry oraz przetwarzania informacji z termometru wykorzystywane są ogólnodostępne biblioteki w języku Python. Obsługa MQTT także wykorzystuje gotową bibliotekę w tymże języku.

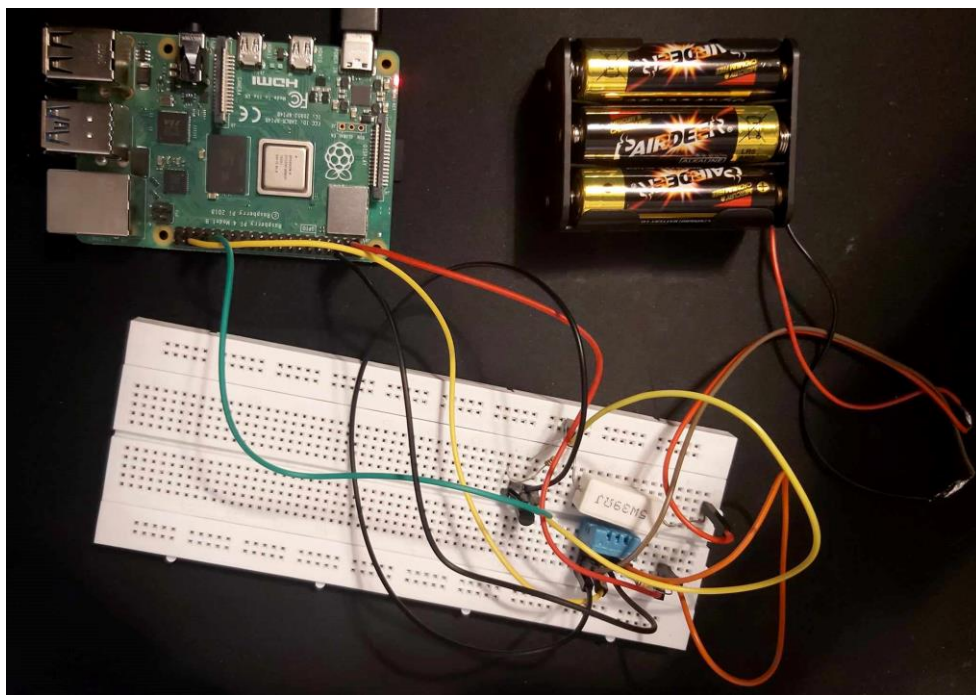
By uzyskać możliwość zdalnego sterowania, tj. sterowania podczas gdy użytkownik i grzałka znajdują się w różnych sieciach LAN, komunikacja MQTT wykorzystuje ogólnodostępny broker danych o adresie test.mosquitto.org. Do przesyłu danych wykorzystywane są struktury JSON i kilka różnych tematów subskrypcji przez MQTT. Na początku funkcjonowania, mikrokontroler sprawdza także czy posiada połączenie internetowe oraz czy ma dostęp do używanego brokera danych.

Kod sterowania jest zbudowany w ten sposób, że aktywna jest wciąż pętla programu, która wykonuje czynności sterowania tylko co ustalony wcześniej interwał zbierania danych. Po każdym otrzymaniu nowych danych pomiarowych regulator PID oblicza nową wartość wypełnienia sygnału PWM.

Jako Frontend w układzie wykorzystywana jest aplikacja mobilna napisana za pomocą platformy programistycznej Flutter, która buduje aplikację na zasadzie drzewa widżetów. W aplikacji mobilnej wykonywane są jedynie procesy pobierania i wysyłania danych za pomocą MQTT, zapisywania i wyświetlania danych oraz obliczanie prostych danych statystycznych na podstawie otrzymanych pomiarów.

#### IV. Opis działania opracowanego „zadania laboratoryjnego”.

Układ rzeczywisty funkcjonuje zgodnie z wcześniej przyjętymi założeniami. Pod podłączeniu zasilania do Raspberry PI, po chwili automatycznie uruchamia się skrypt, a płytka zaczyna wysyłać dane do brokera MQTT. Po uruchomieniu aplikacji mobilnej można od razu zobaczyć aktualną temperaturę układu i rozpocząć sterowanie grzania.

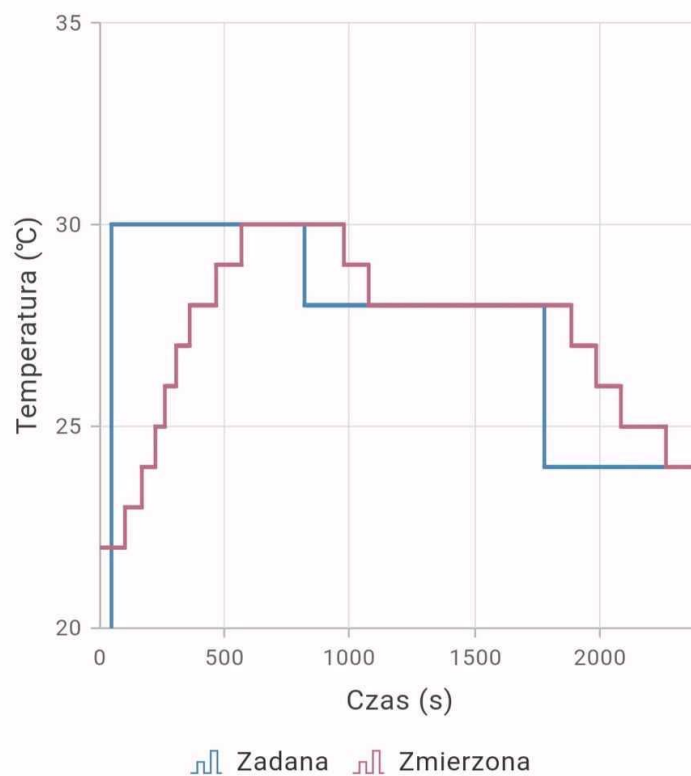


*Rysunek 4 - realizacja fizyczna układu sterowania*

Na Rysunku 4 można zauważyć, że rezystor ceramiczny i sensor DHT11 są umieszczone blisko siebie. Taka konfiguracja jest konieczna, by zapewnić poprawne działanie układu, gdyż temperatura mierzona powinna być jak najwierniejsza aktualnej temperaturze rezystora.

Wszystkie potrzebne informacje do sterowania i monitoringu można wyświetlać i wprowadzać korzystając z aplikacji mobilnej. Prezentuje się ona w następujący sposób.

### ≡ MiS temperaturą rezystora



#### MONITOROWANIE

Interwał zbierania danych: 2 s ✓  
Aktualna temperatura: 24 °C  
Prąd na rezystorze grzejącym: 16.1 mA

#### STEROWANIE

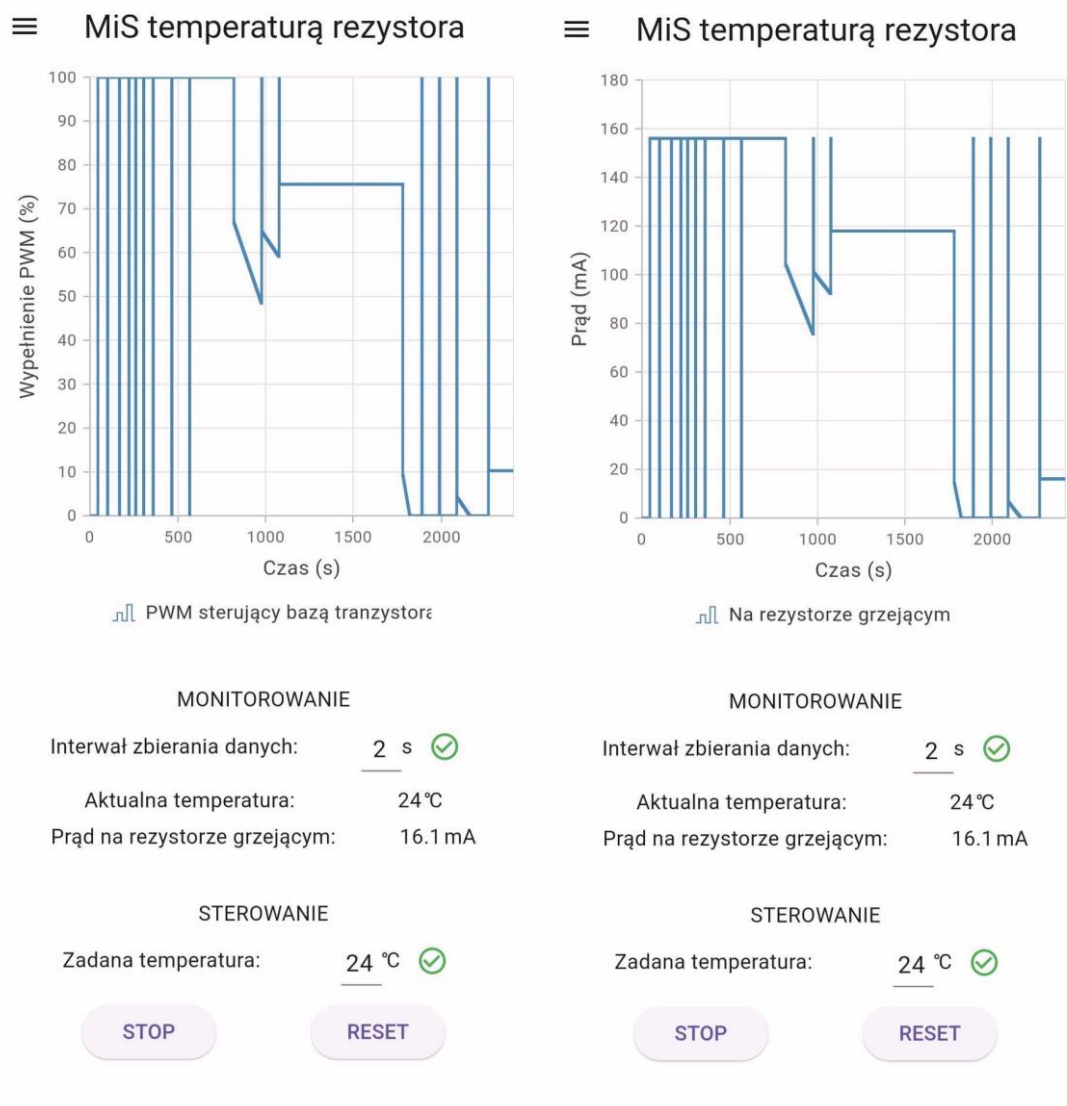
Zadana temperatura: 24 °C ✓

STOP

RESET

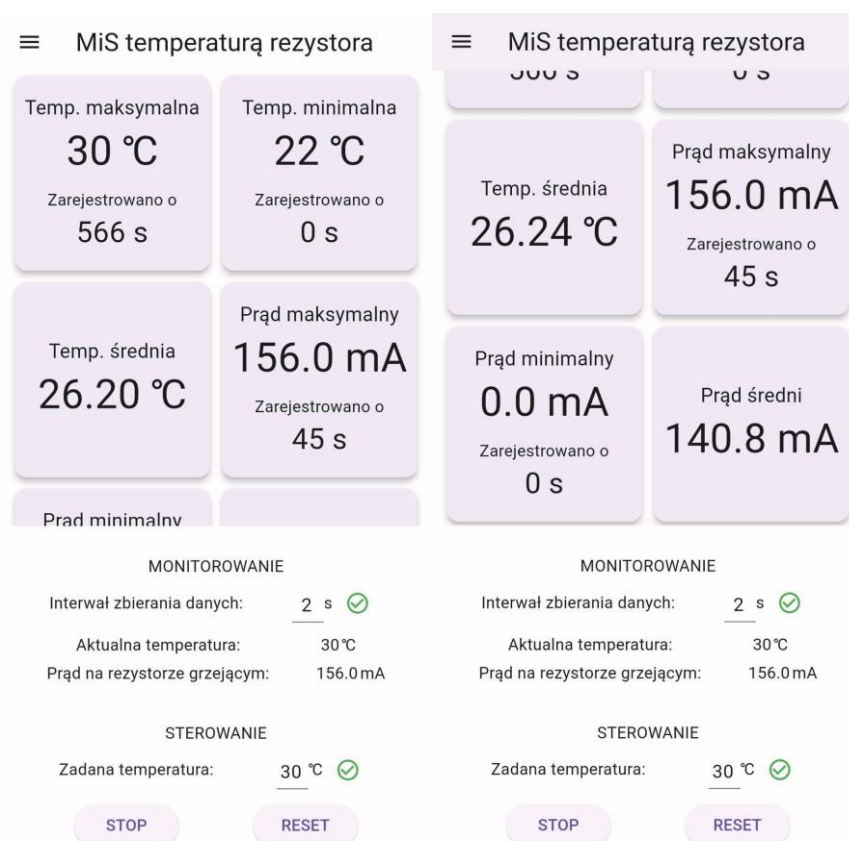
Rysunek 5 - główny ekran aplikacji mobilnej

Można też wybrać inne ekrany, by śledzić zmiany sygnału PWM, oraz różnych danych statystycznych.



Rysunek 6 - wykres sygnału PWM/prądu w aplikacji

Wykres PWM jest faktycznym odwzorowaniem procentowego wypełnienia sygnału, natomiast wartości prądu na rezystorze ceramicznym są proporcjonalnie przeliczane w zależności od wartości sygnału PWM. Występujące na wykresach skoki wartości, są spowodowane ponownym naliczaniem sygnału sterowania po każdej zmianie temperatury. Jest to błąd synchronizacji przesyłu danych z obliczaniem nowych wartości sterowania, jednak same wartości, mimo że widoczne na wykresie, nie wpływają na działanie układu, gdyż ten nie zdąży nawet zareagować na krótki impuls sygnału.



Rysunek 7 - wyświetlanie danych statystycznych w aplikacji

## V. Podsumowanie.

Podczas realizacji zadania laboratoryjnego został zrealizowany układ automatycznej regulacji temperatury rezystora wraz z możliwością zmiany temperatury zadanej i interwału zbierania danych. Układ ma możliwości sterowania zdalnego, a jego stała czasowa zależy głównie od zewnętrznego źródła zasilania podłączonego do tranzystora. W aplikacji mobilnej można łatwo wyświetlać różne dane w układzie, śledzić ich historię i odczytywać proste dane statystyczne.

Całość została zaprojektowana w ten sposób, by dalsze kierunki rozwoju były jak najbardziej uproszczone. Ponieważ system już jest w stanie funkcjonować całkowicie zdalnie, jako potencjalny kierunek rozwoju można obrać system zdalnego ustalania temperatury w pomieszczeniu, zmieniając element grzejący na kaloryfer w pokoju i sterować przepływem wody w głowicy i umieszczając termometr w drugiej części pomieszczenia.

Sama aplikacja mobilna jest też tak zbudowana, by można było dodać kolejny ekran bez dużych zmian struktury aplikacji. Można w prosty sposób dodać wykres innych mierzonych wartości lub innych danych statystycznych.

W razie posiadania kilku takich stacji sterowania, można też podłączać się do każdej z nich zmieniając jedynie przedrostek tematu MQTT.