

Projekt zaliczeniowy

**Regulator-PID-ogrzewanie**

Julia Wałowska 153302

Józef Wiatr 51088

1. **Opis Projektu**

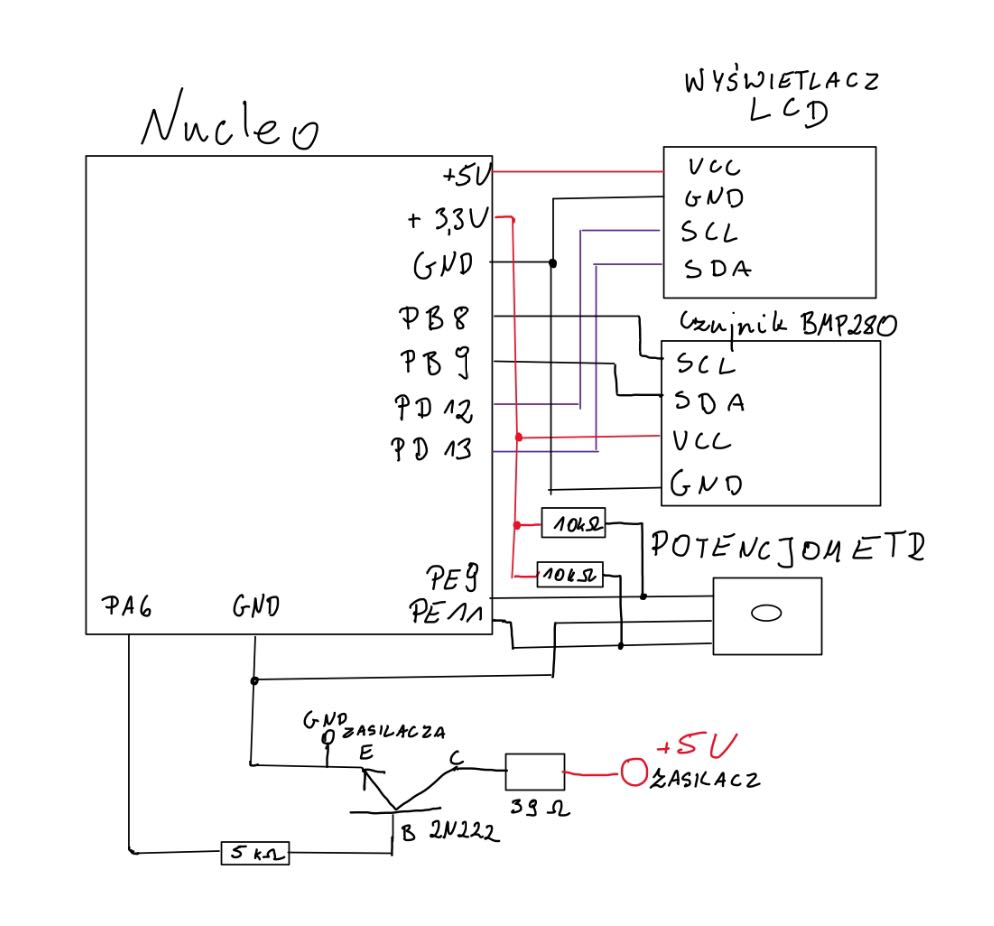
Celem projektu było zbudowanie układu automatycznej regulacji temperatury w oparciu o:

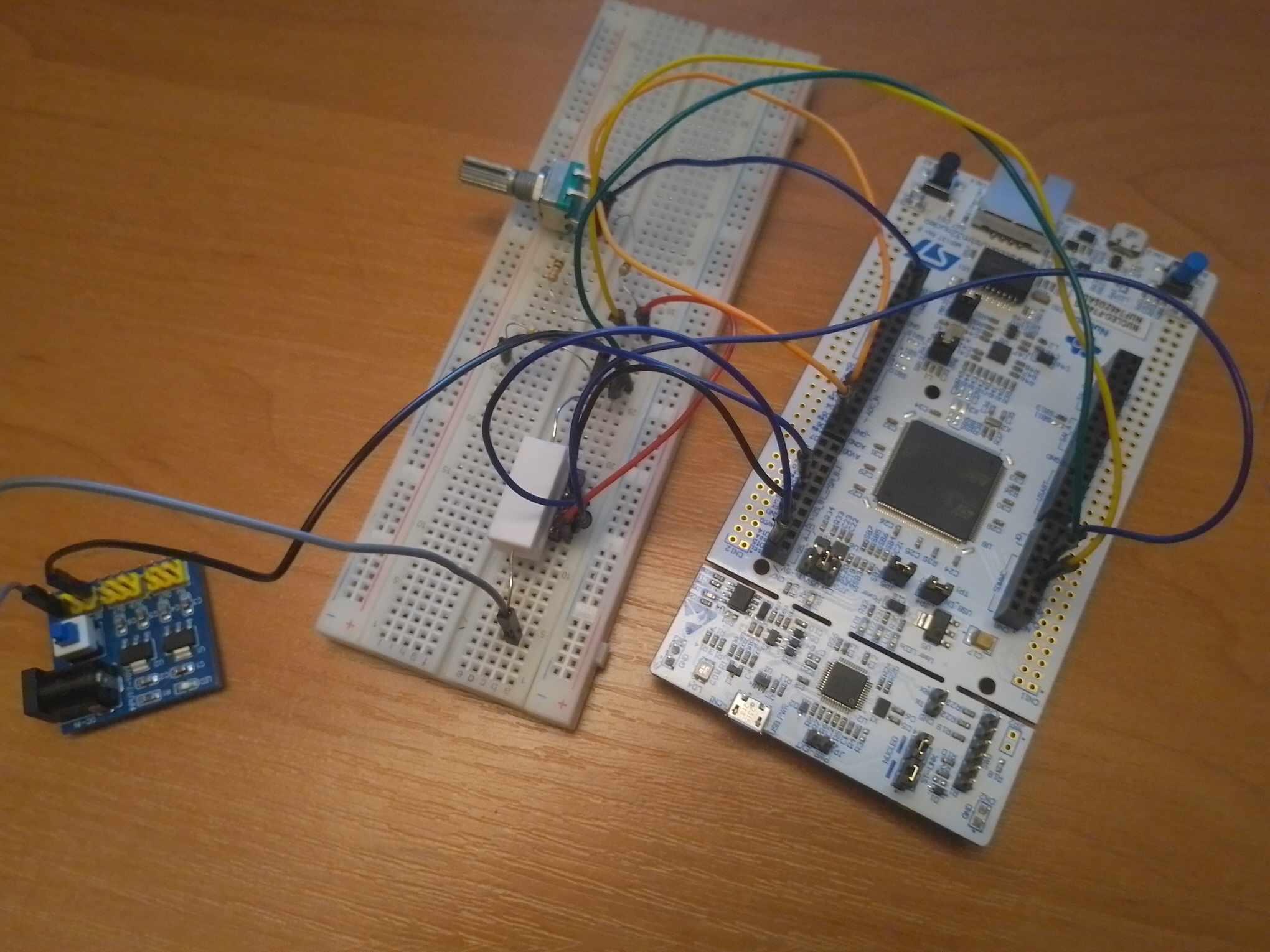
* płytkę NUCLEO 144
* regulator PID
* rezystor 39Ω
* czujnik BMP280

Poza elementami podstawowymi wykorzystano również:

* enkoder
* zasilacz 5W
* przewody
* płytka stykowa
* rezystory

1. **Schemat układu**



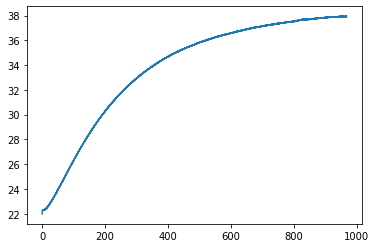


Zdjęcie 1: Połączony układ

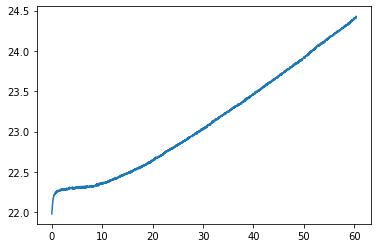
1. **Analiza modelu układu**

Do przeanalizowania układu na początku wykonano obliczenia analityczne:

Wniosek z obliczeń - nie można lokować biegunów.

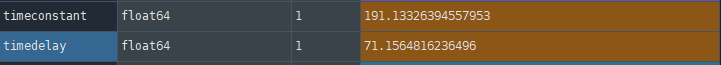


Wykres 1: Odpowiedź skokowa na podstawie której wyznaczono parametry.



Wykres 2: Zbliżenie na początek wykresu.

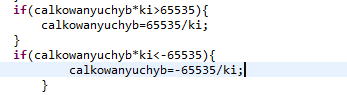
Następnie na podstawie odpowiedzi skokowej i przygotowanego skryptu wyliczono T0 , T oraz wzmocnienie obiektu k Analizy dokonano na podstawie odpowiedzi skokowej i dopasowania obiektu inercyjnego z opóźnieniem transportowym, a wyniki zapisano w pliku. Wyliczone wartości zaimplementowano w modelu.



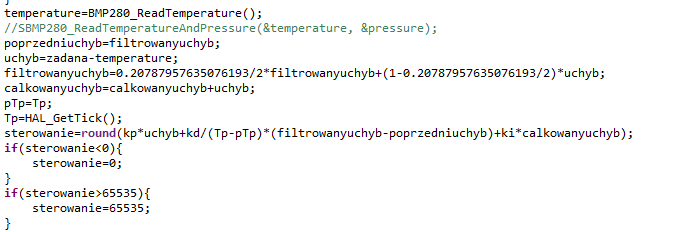


Wyliczone parametry są dla transmitancji pomiędzy sygnałem sterującym tj. wypełnieniem(konwencjonalnym 0-100%) razy 65536(w takiej formie jest ono podawane w kodzie) a temperaturą odczytywaną z czujnika.

W celu zapewnienia lepszej regulacji zaimplementowano także wind-up:



Oraz poddano sygnał prostej filtracji przed poddaniem go różniczkowaniu:



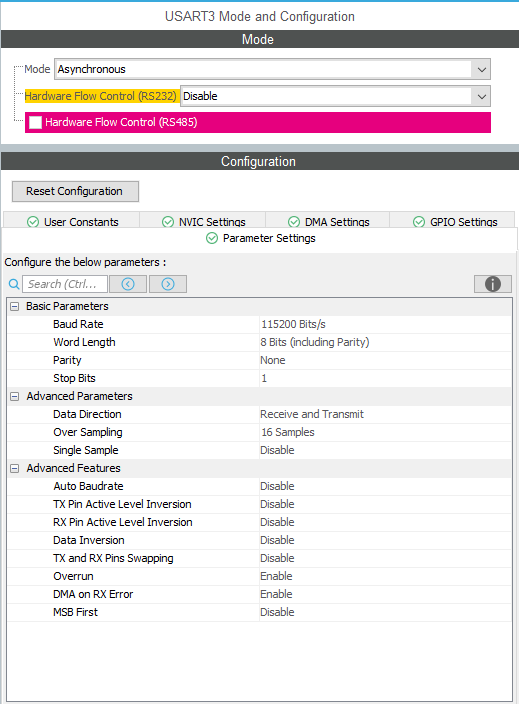




Warto też zauważyć, że konieczne było wprowadzenie nasycenia sygnału sterującego w przypadku, z powodu ograniczenia jakie nakłada płytka nucleo na sterowanie wypełnieniem sygnału PWM dawanego na bazę tranzystora.

1. **Komunikacji szeregowa**

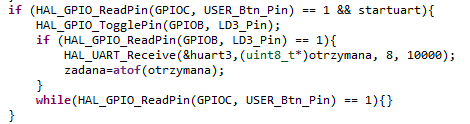
W celu zrealizowania zadania zdecydowaliśmy się korzystać z komunikacji szeregowej USART3. Początkowo skonfigurowano ioc:



Zdjęcie 2: Konfiguracja ioc dla komunikacji szeregowej

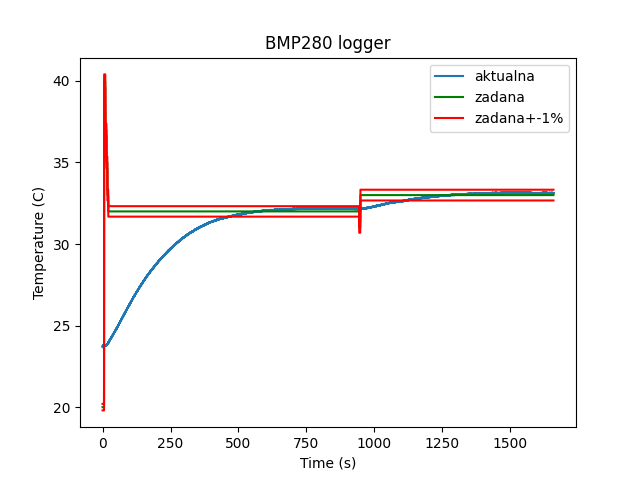
Następnie za pomocą kodu zainicjalizowano dwukierunkową komunikację szeregową:

* zadawanie wartości:



1. **Wizualizacja GUI - otrzymany uchyb**

Wizualizację GUI przygotowano przy pomocy języka python. Aby umożliwić wizualizację należało “wysłać” informacje z płytki:



Wykres 2: Wizualizacja GUI z zaznaczonym 1% uchybem

Na wykresie widać, że uzyskany uchyb regulacji mieści się w 1%.

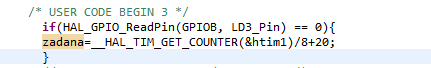
1. **Implementacja enkodera**

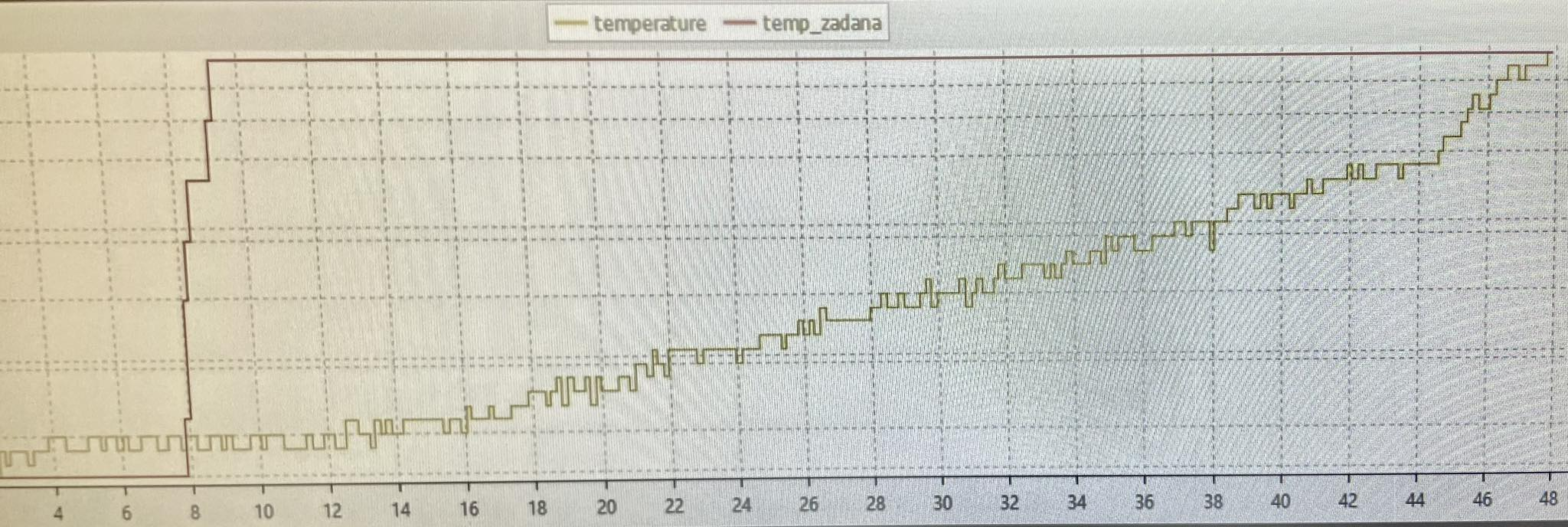
Konfiguracja ioc:



Zdjęcie 3: Konfiguracja ioc dla enkodera

Przygotowano kod, którego zadaniem jest przypisanie wartości ustawionej za pomocą enkodera jako wartość ustaloną:

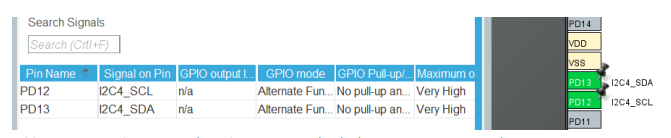




Wykres 3: Grzanie rezystora po zmienieniu nastaw prz użycii enkodera.

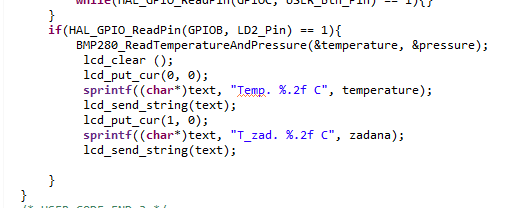
1. Implementacja wyświetlacza

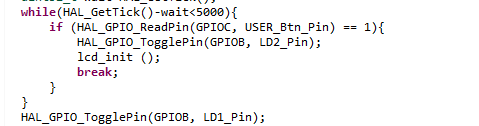
Konfiguracja ioc:



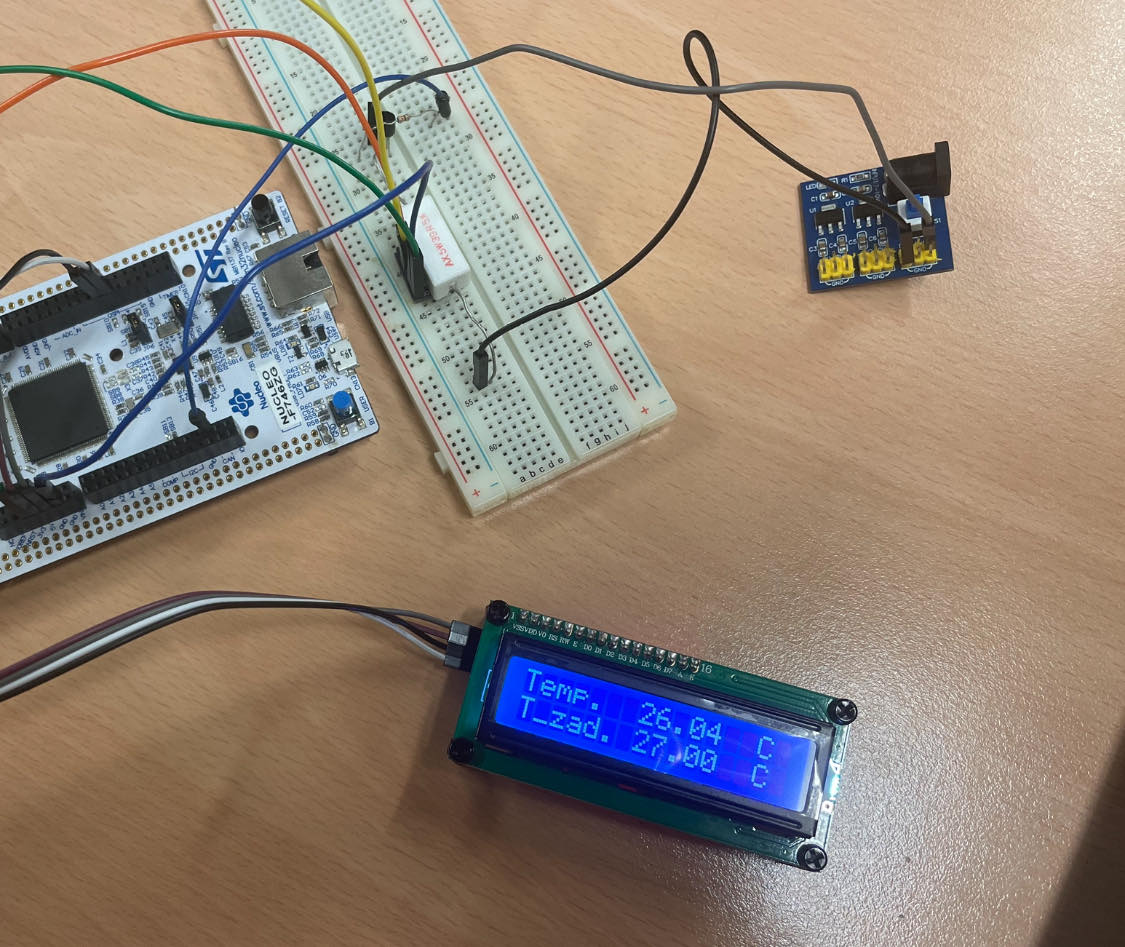
Zdjęcie 4: Konfiguracja ioc dla wyświetlacza LCD

Przygotowano kod, którego zadaniem jest wyświetlenie wartości zadanej i odczytanej na ekranie. Dodatkowo ekran jest uruchamiany za pomocą przycisku USER:





Niestety ze względu na brak wyświetlacza w domu, nie udało się korzystać z niego w trakcie nagrania. Przygotowany kod wykorzystano jednak w trakcie laboratoriów dodatkowych).



Zdjęcie 5: Wyświetlana temperatura zadana i odczytana.

1. **Kontrola wersji**

Projekt realizowano z wykorzystaniem kontroli wersji github.

Link do repozytorum: https://github.com/juliawalowska/Regulator-PID-ogrzewanie