AUTOMATYCZNA REGULACJA TEMPERATURY

**Mikroprocesorowy system sterowania i pomiaru. Regulator PID.**

|  |  |
| --- | --- |
| **AUTORZY PROJEKTU** | **DOMINIKA FILIPIAK**  **WOJCIECH FLORCZAK** |
| **GRUPA** | **GRUPA DZIEKAŃSKA – A1**  **GRUPA LABORATORYJNA L2** |
| **POLITECHNIKA POZNAŃSKA**  **WYDZIAŁ WARIE**  **AUTOMATYKA I ROBOTYKA** | |

**Spis treści**

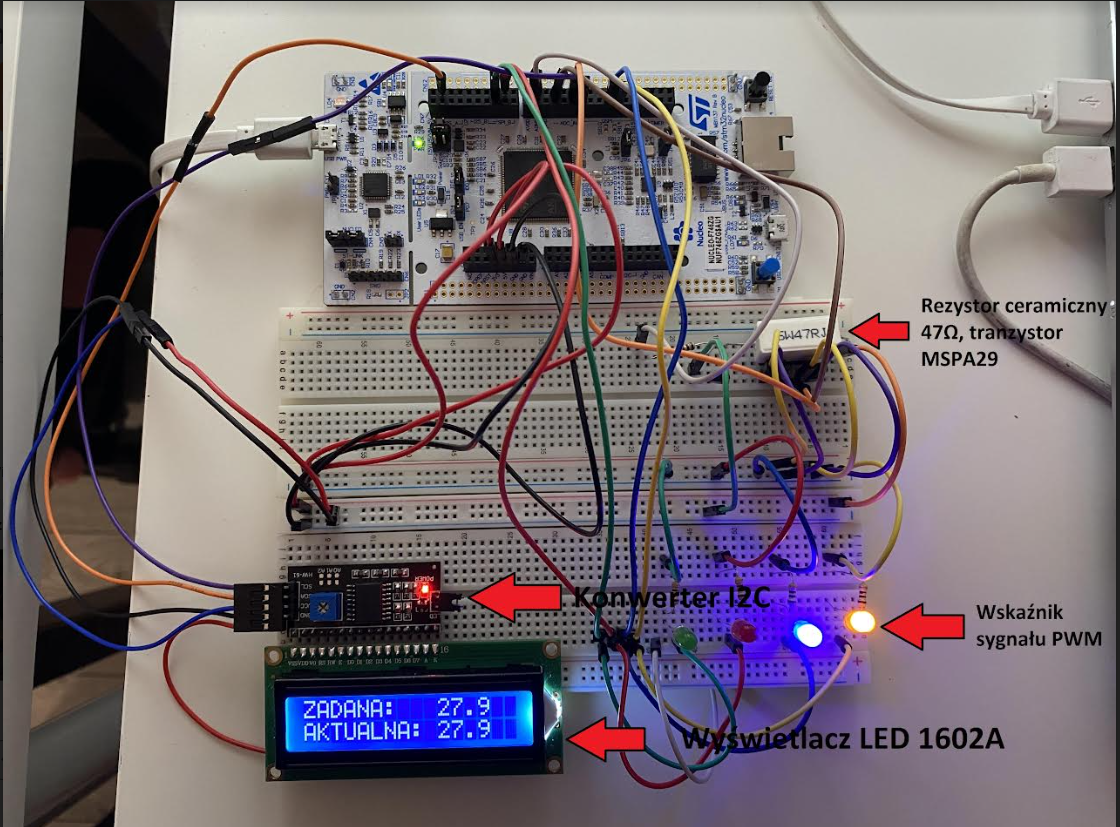
1. **Założenia projektowe oraz wykorzystane komponenty1**
2. **Realizacja układu w formie graficznej3**
3. **Analiza działania układu regulatora temperatury 1**
4. **Schemat elektroniczny złożonego układu projektowego1**
5. **Model zaprojektowany w środowisku Matlab – Simulink1**
6. **Graficzna wizualizacja – środowisko STM32CubeMonitor1**
7. **Opis rozwiązań zastosowanych w programie mikroprocesora1**
8. **Założenia projektowe**

Zadaniem projektowym było stworzenie układu automatycznej regulacji temperatury rezystora sterowanego za pomocą tranzystora.

**Wykorzystane komponenty:**

* Nucleo F746ZG
* Rezystor ceramiczny 47Ω
* Tranzystor bipolarny NPN Darlington MPSA29
* Rezystor 220Ω
* Czujnik BMP280
* Wyświetlacz LCD 1602A z konwerterem I2C
* Przewody połączeniowe
* Diody LED (żółta, niebieska, czerwona, zielona)

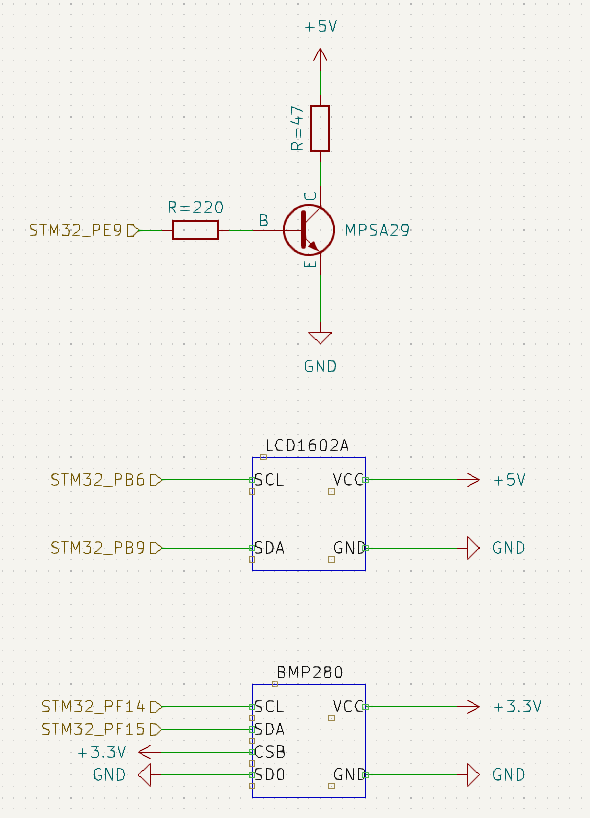
1. **Realizacja układu**

****

1. **Analiza działania układu**

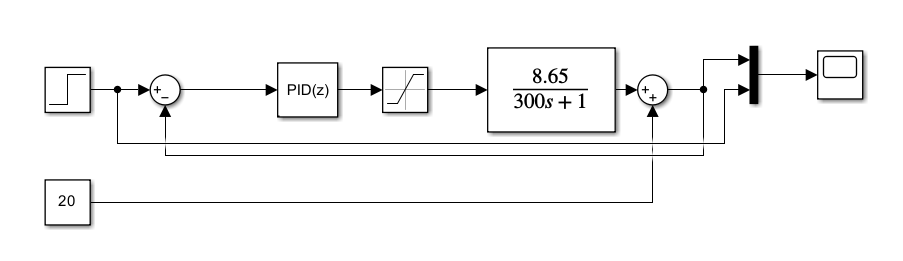
W projekcie zajmujemy się regulacją temperatury rezystora ceramicznego 47 [Ω], który został umieszczony tuż nad czujnikiem temperatury i ciśnienia BMP280. Przepływem prądu przez wspomniany rezystor steruje tranzystor NPN, do którego dobrano rezystor 220 [Ω]. Użyto również wyświetlacza LCD wraz z konwerterem I2C, który umożliwia proste przesyłanie danych. Na wspomnianym wyświetlaczu możliwy jest odczyt temperatury zadanej (jej zmiana jest możliwa dzięki wbudowanemu w płytkę Nucleo przyciskowi USER z krokiem 0.1 [⸰C], natomiast resetowanie tej wartości dokonujemy poprzez również wbudowany przycisk RESET) oraz aktualna temperatura odczytywana z czujnika BMP280. Układ zasilany jest bezpośrednio z płytki Nucleo – wykorzystujemy zarówno napięcie 3.3 [V], jak i 5 [V]. Algorytm działania zadania regulacji stałowartościowej realizowany jest na płytce NucleoF746ZG opartej na mikrokontrolerze STM32F746ZGT6. Dodatkowo stworzyliśmy wizualizację wypełnienia sygnału PWM (diody obok wyświetlacza). Dioda żółta migająca oznacza, że wartość sygnału PWM jest równa 0. Natomiast zapala się w momencie gdy wartość wypełnienia PWM oscyluje w zakresie <0 ; 25> [%]. Dioda niebieska informuje o wypełnieniu z zakresu <25 ; 50> [%].   
I kolejno diody czerwona <50 ; 75> [%], zielona <75 ; 100 > [%].

1. **Schemat elektroniczny**

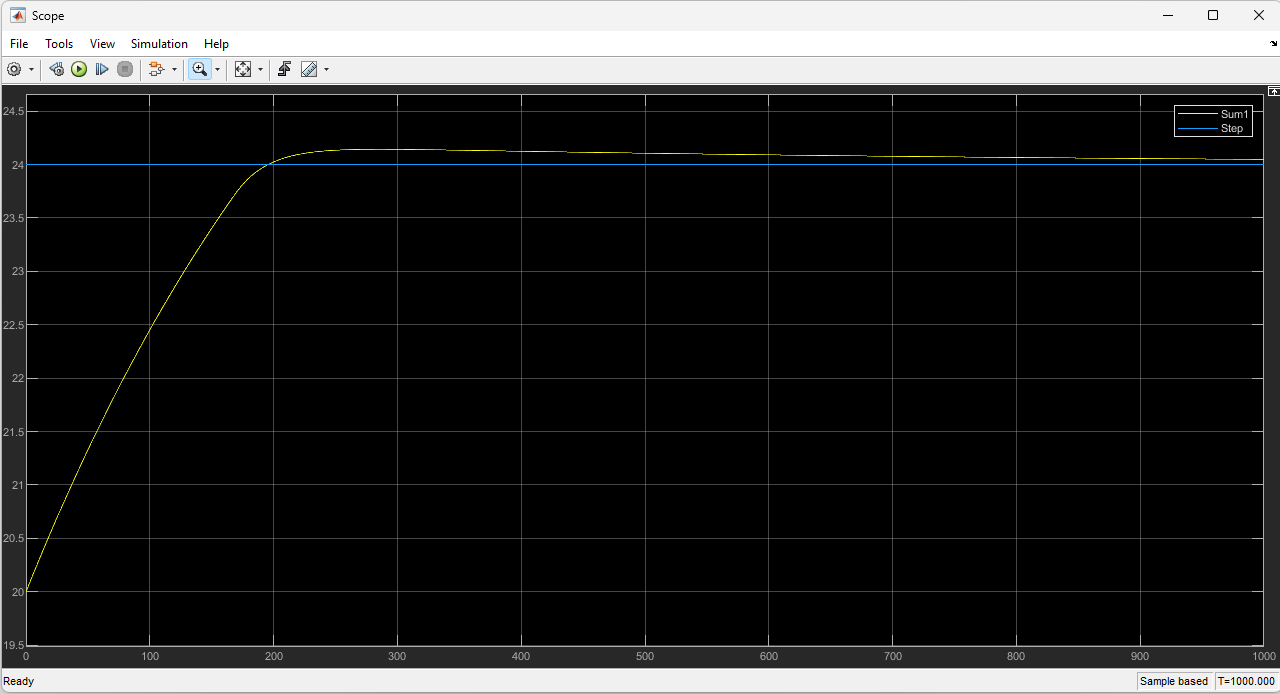
****

1. **Model Matlab – Simulink**

**Do wykonania modelu posłużyliśmy się transmitancją rezystora, która została podana na jednym z wykładów. Zaprojektowaliśmy model URA z wykorzystaniem Matlab -Simulink.**

****

**Rys. 1 Schemat URA wykonany w Simulink**

****

**Rys. 2 Wykres przebiegu temperatury rezystora ceramicznego oraz wartości zadanej (tutaj na wartości 24)**

Widzimy, że na powyższej symulacji występuje minimalne przeregulowanie rzędu 0,5%, jest więc ono pomijalnie małe. Jeśli chodzi o wartości uchybu ustalonego to również mieści się ona w zakresie tolerancji ( >1%).

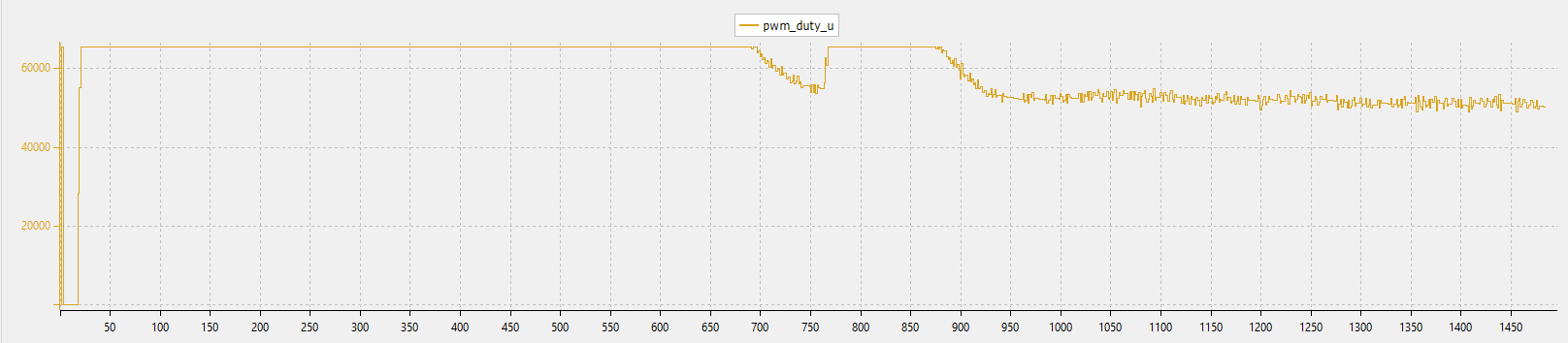
Doboru nastaw regulatora dokonano przy pomocy bloku Discrete PID Controller. Blok ten pomógł nam tak dobrać nastawy, aby spełnione były założenia projektowe, czyli odpowiednio niski uchyb ustalony. Wyznaczone nastawy przyjęły wartości:

**Kp= 1.2**

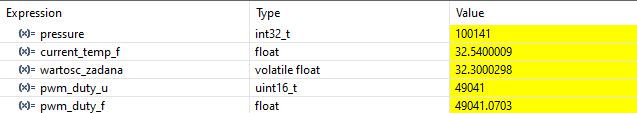
**KI = 0.008**

**KD = 0**

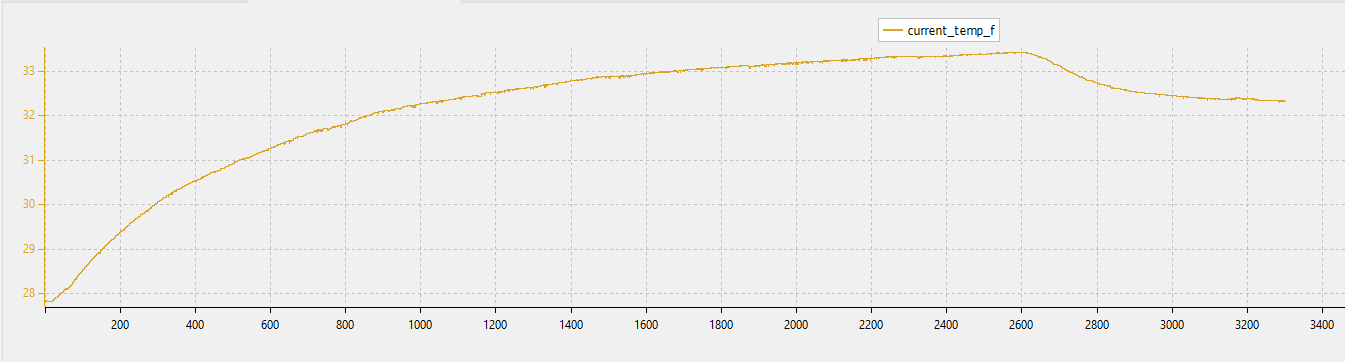
**Następnie nastawy zostały zaimplementowane w programie.**

****

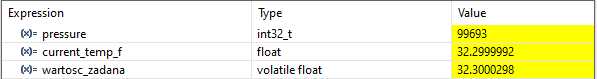
**Rys. 3 Przebieg sygnału sterującego PWM w Data Trace Timeline Graph**

****

Na powyższym wykresie widoczny jest przebieg sygnału sterującego PWM na przestrzeni 1450 [s]. Zadana przez nas wartość wynosiła 32.3⸰C, zaś najwyższa osiągnięta wartość temperatury to 32.54⸰C, zatem uchyb ustalony był na poziomie 0,74% co spełnia założenie projektowe (eust < 1%). Widoczny na wykresie spadek wartości sygnału PWM w przedziale czasowym <700 ; 750> [s] wynika z osiągnięcia przez rezystor zadanej temperatury, a następnie nagły wzrost wartości wynika z celowego ochłodzenia układu. Dowodzi to prawidłowego działania układu regulacji.

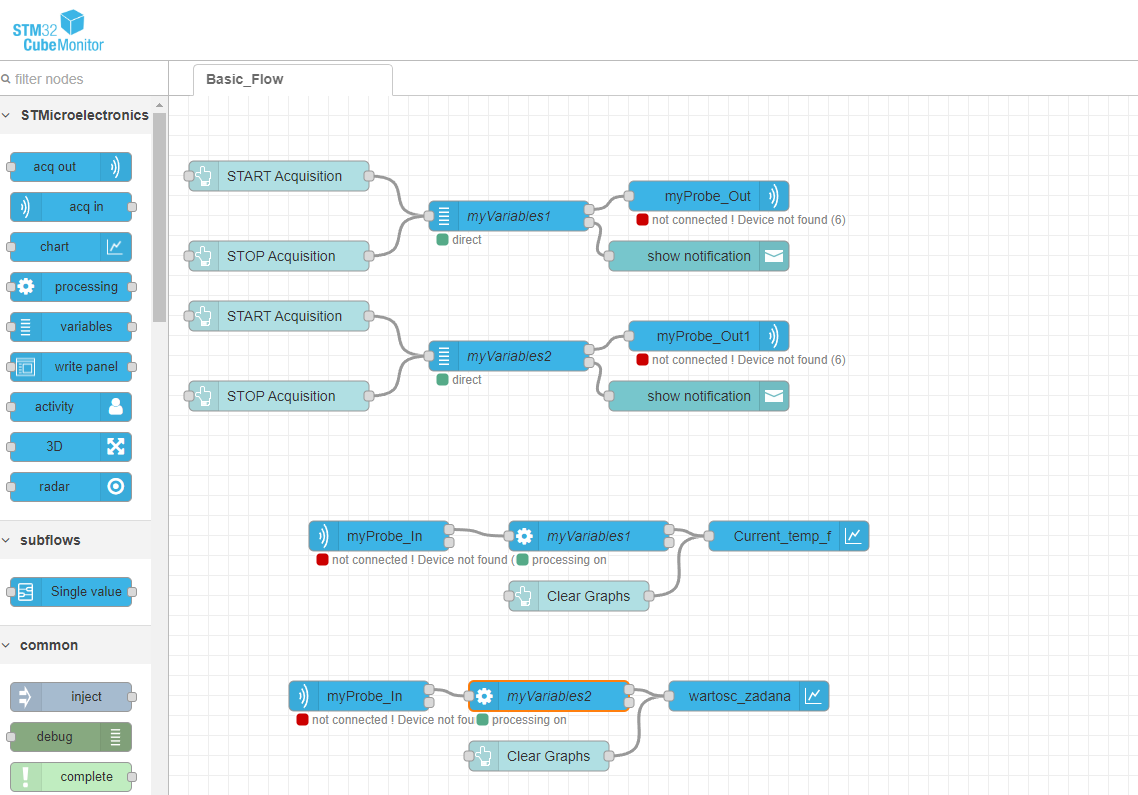
****

**Rys. 4 Przebieg temperatury układu rzeczywistego**

****

Przy dłuższym czasie nagrzewnia możemy zauważyć, że uchyb maleje jeszcze bardziej. Aktualnie jest na poziomie 0,003% co oczywiście nadal spełnia założenie projektowe.

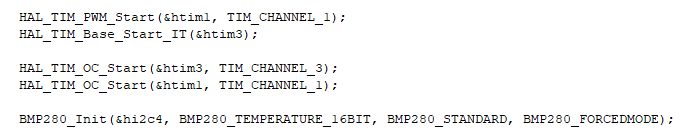
1. **Graficzna wizualizacja za pomocą środowiska STM32CubeMonitor.**

****

**Rys. 4 Schemat blokowy w STM32CubeMonitor**

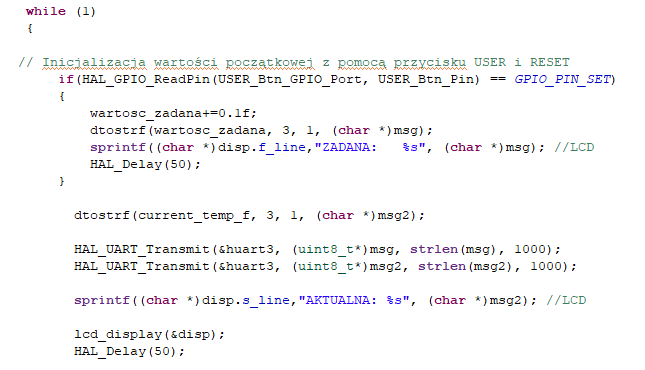
1. **Opis rozwiązań zastosowanych w programie mikroprocesora**

* **Skonfigurowanie interfejsu I2C** do obsługi czujnika oraz wyświetlacza LED (za pośrednictwem konwertera I2C)
* **Konfiguracja timerów:**
* TIM1 został skonfigurowany do obsługi sygnału PWM sterującego bramką użytego tranzystora NPN;
* TIM3 został użyty w trybie przerwaniowym NVIC do wysyłania danych i obliczania sygnału sterującego.

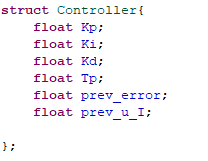
****

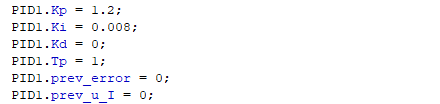
* **Konfiguracja wyświetlacza LCD z wykorzystaniem konwertera I2C**

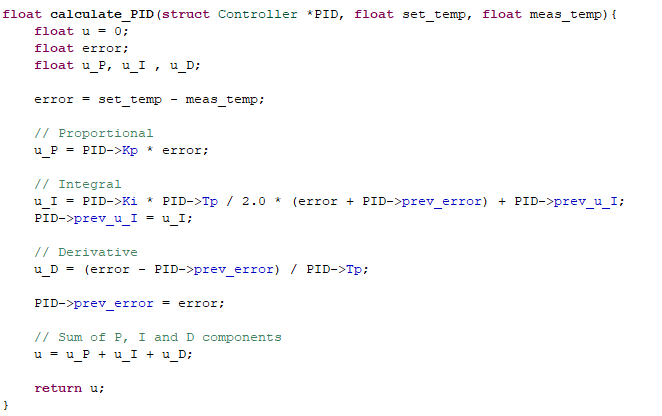
****

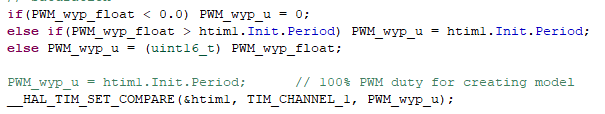
****

* **Implementacja regulatora PID**

****

****

****

****