

PROJEKT

STEROWNIKI ROBOTÓW

Sterownik wentylatora z czujnikiem DHT11 z interfejsem webowym

Skład grupy:
Piotr KOCZY, 259367
Michał RAGAN, 259364

Termin: wtTN19

Prowadzący:
dr inż. Wojciech DOMSKI

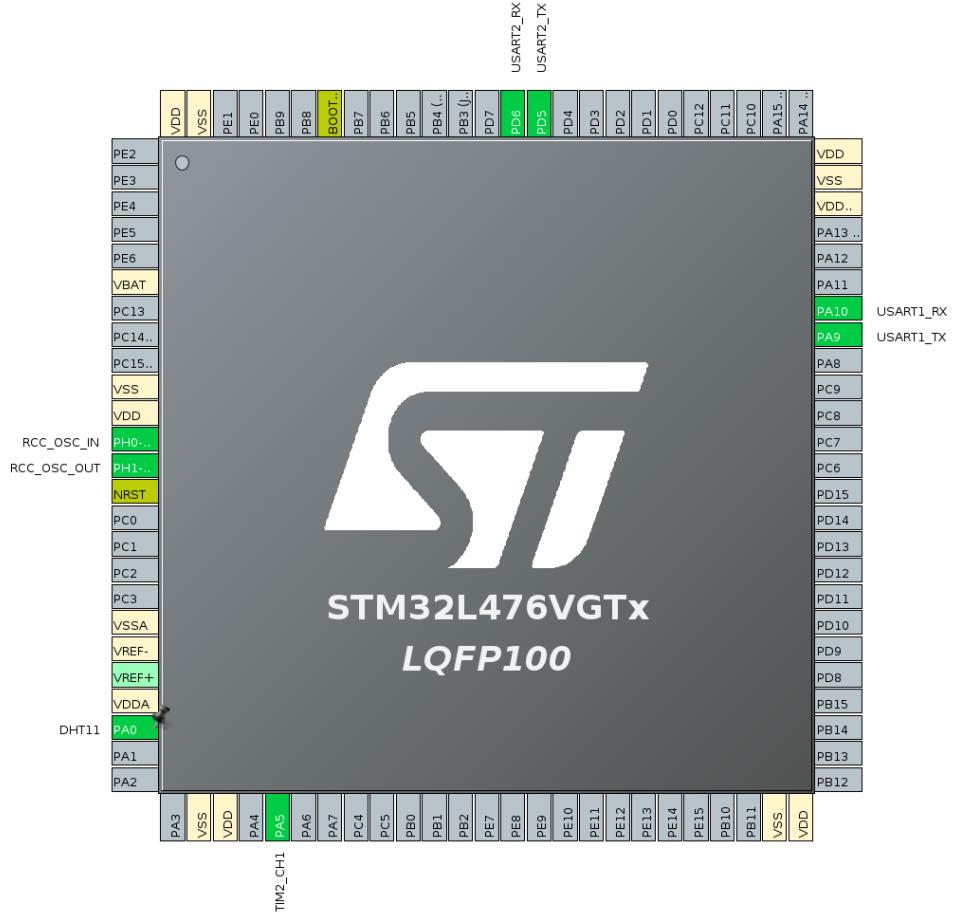
Spis treści

1 Opis projektu	2
2 Konfiguracja mikrokontrolera	2
2.1 Konfiguracja pinów	4
2.2 USART	4
2.3 Timery	4
2.3.1 RCC	4
2.3.2 TM2	4
2.3.3 TM1	4
3 Urządzenia zewnętrzne	5
3.1 Mikrokontroler STM32L476	5
3.2 Czujnik DHT11	5
3.3 Moduł WiFi ESP8266	5
3.4 Konwerter PWM na napięcie DC 0-10V	5
3.5 Wentylator	5
4 Projekt elektroniki	6
5 Konstrukcja mechaniczna	7
6 Opis działania programu	7
6.1 Odczytywanie danych z czujnika DHT11	7
6.2 Generowanie PWM	7
6.3 Interfejs webowy	8
6.3.1 Inicjalizacja	8
6.3.2 Praca systemu	8
6.4 Interfejs	8
7 Repozytorium	9
8 Harmonogram pracy	9
8.1 Podział pracy	10
9 Podsumowanie	10
Bibliografia	11

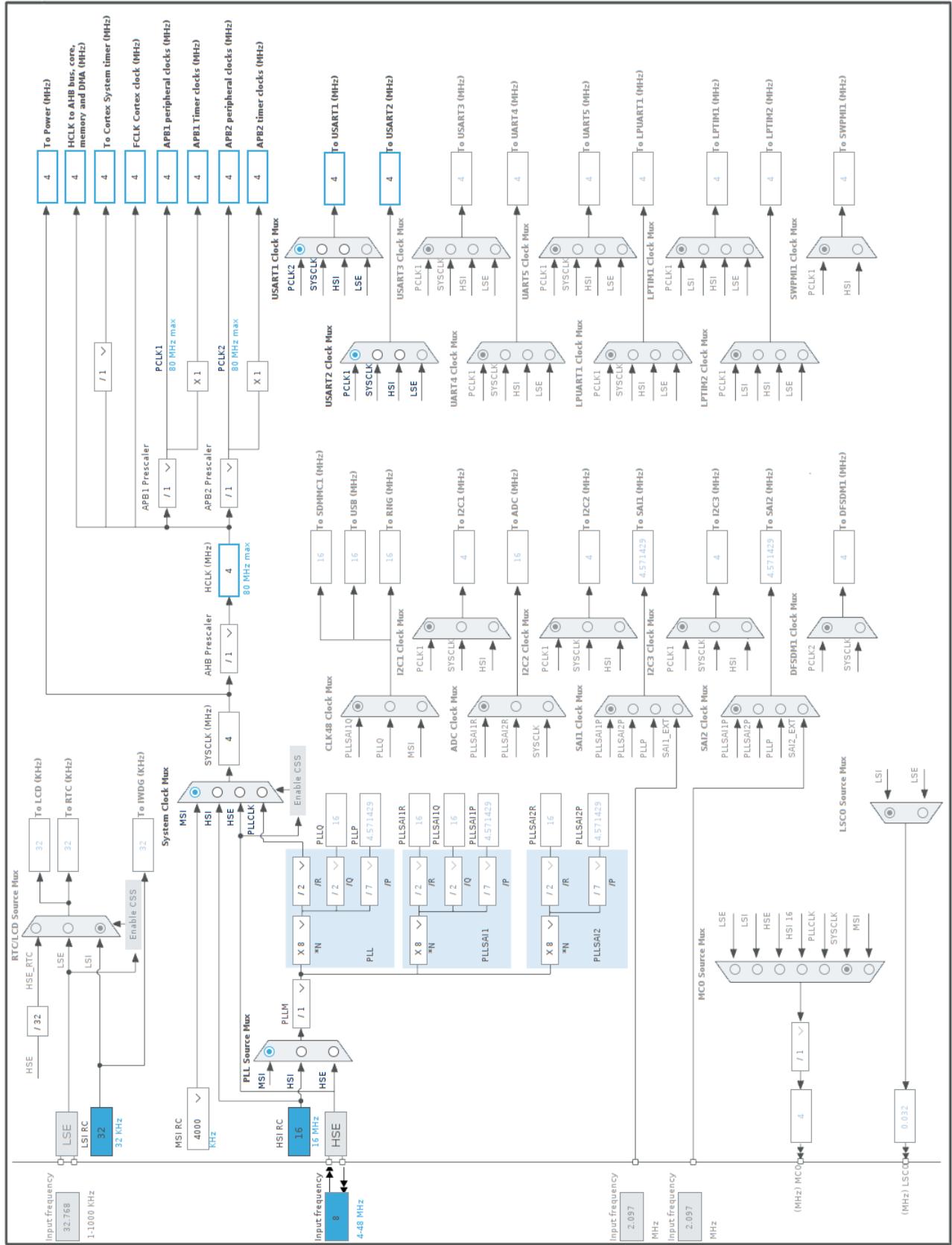
1 Opis projektu

Projekt „Sterownik wentylatora z czujnikiem DHT11 z interfejsem webowym” polega na stworzeniu urządzenia umożliwiającego automatyczne sterowanie wentylatorem na podstawie wskazań czujnika temperatury i wilgotności DHT11 oraz zdalne sterowanie poprzez interfejs webowy.

2 Konfiguracja mikrokontrolera



Rysunek 1: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX



Rysunek 2: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

Numer pinu	Pin	Tryb pracy	Etykieta
23	PA0	GPIO_Output	DHT11
30	PA5	TM2_CH1	
68	PD5	USART1_TX	
69	PD6	USART1_RX	
86	PD5	USART2_TX	
87	PD6	USART2_RX	

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 USART

Wykorzystane zostały 2 interfejsy UART. USART1 jest wykorzystywany do komunikacji z modułem WiFi, a USART2 służy głównie do debugowania programu, oba interfejsy mają skonfigurowane parametry w ten sam sposób 2.

Baud Rate	11520
Word Length	8 Bits (including parity)
Parity	None
Stop Bits	1

Tabela 2: Konfiguracja peryferium USART

2.3 Timery

2.3.1 RCC

Do sterowania wentylatorem zostanie wykorzystany sygnał PWM generowany z wykorzystaniem RCC (High Speed Clock).

HSI Calibration Value	16
MSI Calibration Value	0
MSI Auto Calibration	Disabled

Tabela 3: Konfiguracja peryferium RCC

2.3.2 TM2

Sygnał PWM do sterowania wentylatorem jest generowany w tym timerze o parametrach:

Prescaler (PSC - 16 bits value)	0
Counter Mode	Up
Counter Period	65535
Internal Clock Division (CKD)	No Division
auto-reload preload	Enable

Tabela 4: Konfiguracja peryferium TM2

2.3.3 TM1

Do komunikacji z czujnikiem za pomocą protokołu 1-WIRE wykorzystany jest timer TM1, ma on rozdzielcość 1us.

Prescaler (PSC - 16 bits value)	4
Counter Mode	Up
Counter Period	0xffff-1
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

Tabela 5: Konfiguracja peryferium TM1

3 Urządzenia zewnętrzne

3.1 Mikrokontroler STM32L476

Mikrokontroler STM32L476 będzie sterować wentylatorem na podstawie odczytanych wartości z czujników. [1] [3]

3.2 Czujnik DHT11

Czujnik temperatury i wilgotności DHT11 posłuży do odczytu tych wartości. Komunikacja jest przy użyciu protokołu 1-wire z wykorzystaniem rezystora PULL-UP [2]

3.3 Moduł WiFi ESP8266

Zewnętrzny moduł WiFi ESP8266 zapewni łączność bezprzewodową przez interfejs webowy.

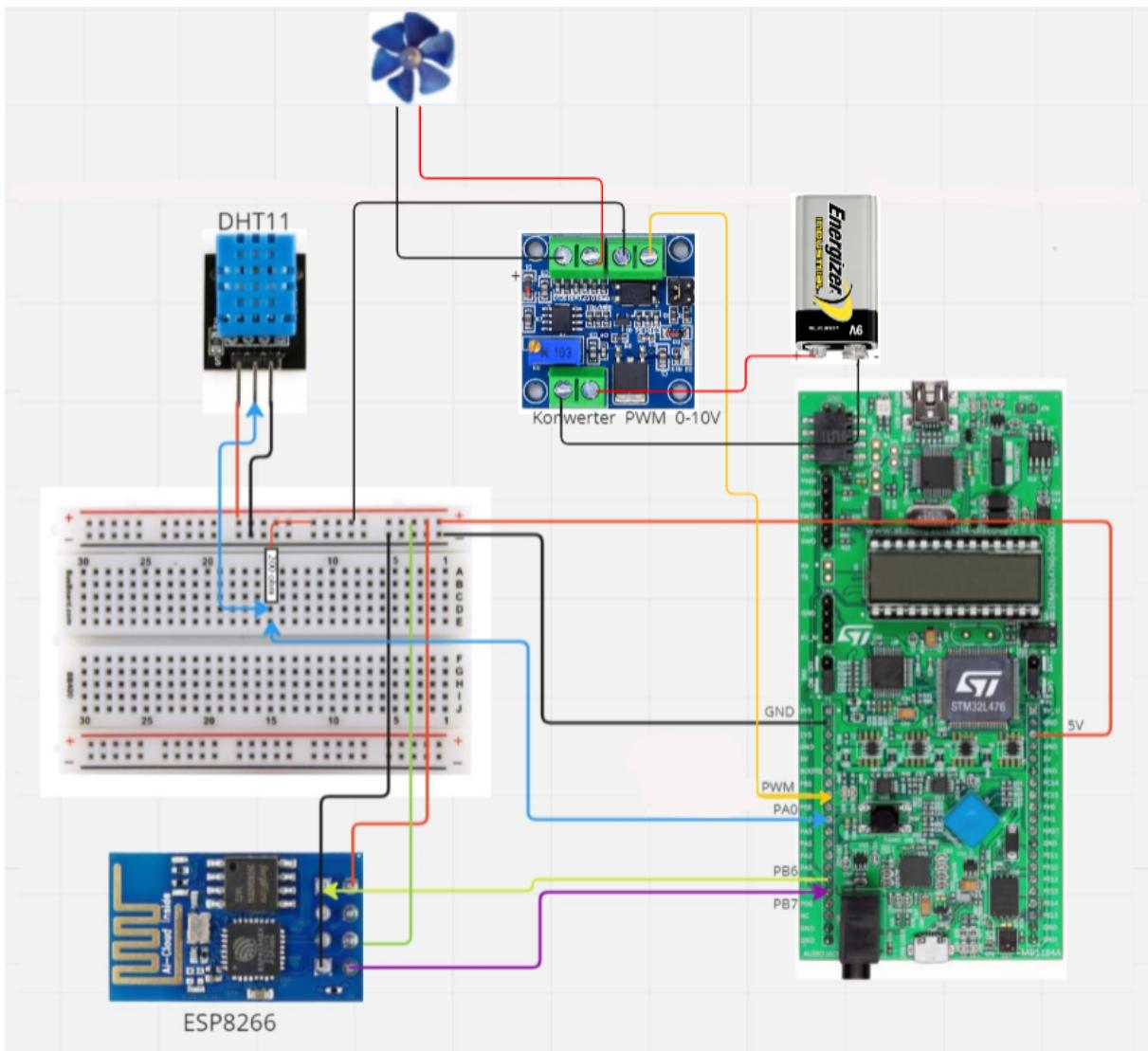
3.4 Konwerter PWM na napięcie DC 0-10V

Mikrokontroler wysyła sygnał do konwertera, który steruje wentylatorem.

3.5 Wentylator

Obroty wentylatora sterowane są napięciem, dlatego wymagany jest konwerter.

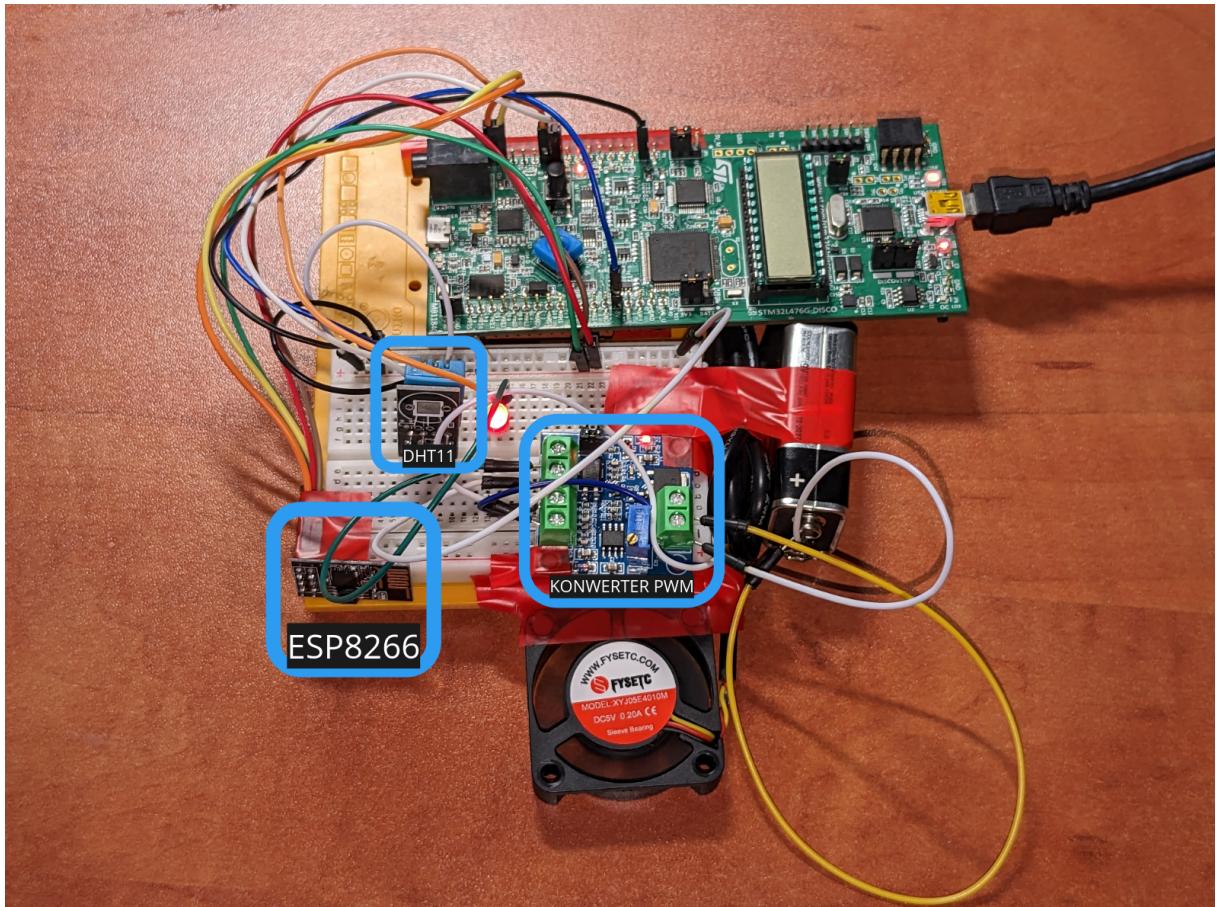
4 Projekt elektroniki



Rysunek 3: Schemat elektroniki

5 Konstrukcja mechaniczna

W naszym projekcie skupialiśmy się na software'owej części bardziej, niż na aspekcie mechanicznej budowy urządzenia.



Rysunek 4: Zdjęcie układu

6 Opis działania programu

6.1 Odczytywanie danych z czujnika DHT11

Komunikacja z czujnikiem odbywa się z wykorzystaniem biblioteki STM32Cube HAL i protokołu 1-wire. Program wysyła żądanie wysłania danych do czujnika, a ten wysyła sekwencję bitów, które zostają odkodowane i zapisane w pamięci jako aktualna temperatura i wilgotność. Czujnik temperatury ma stały błąd pomiaru, dlatego został zastosowany `offset`, który zwiększa odczytaną temperaturę o 8°C, bo tyle wynosi błąd czujnika.

6.2 Generowanie PWM

Sygnał PWM o okresie 65535 i zegarze 80Mhz jest aktualizowany z częstotliwością w przybliżeniu 1220 Hz, co mieści się w wymaganiach zastosowanego konwertera PWM (oczuje on 1-3 kHz). Funkcja generująca stopień wypełnienia sygnału odczytuje aktualną wartość temperatury i wilgotności, a następnie w zależności od zadanych progowych wartości tych parametrów steruje wypełnieniem PWM. Minimum określa próg poniżej którego wypełnienie PWM wynosi 0%, a maximum próg powyżej którego wypełnienie PWM wynosi 100%.

Parametry te zostały ustawione następująco:

- minimum temperatury = 18°C
- maximum temperatury = 30°C
- minimum wilgotności = 70%
- maximum wilgotności = 90%

Po obliczeniu wartości wypełnienia PWM na podstawie odczytu temperatury i wilgotności następuje porównanie ich ze sobą - większa zostaje wybrana jako nowa wartość wypełnienia PWM.

6.3 Interfejs webowy

6.3.1 Inicjalizacja

W pierwszej kolejności następuje inicjalizacja polegająca na zresetowaniu modułu ESP8266 i przekazaniu mu SSID i hasła sieci, z którą ma spróbować się połączyć. Jeśli zakończy się to sukcesem program wyśle poprzez UART do konsoli adres strony.

```
RESETTING..RESETTING.....AT---->OK  
CW MODE---->1  
connecting... to the provided AP  
Connected to,"*****"  
IP ADDR: 192.168.0.192  
CIPMUX---->OK  
CIPSERVER---->OK  
Now Connect to the IP ADDRESS|
```

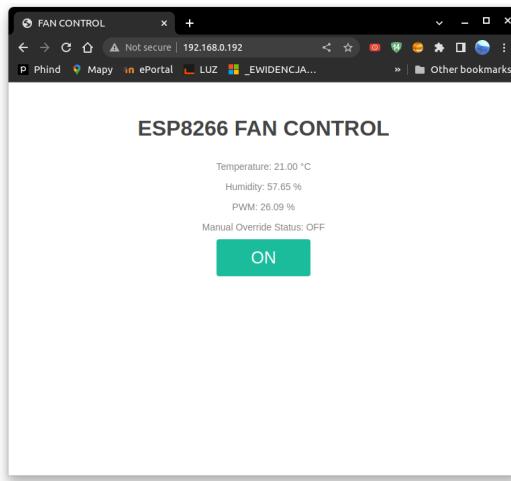
Rysunek 5: Komunikaty wysyłane przez moduł WiFi

6.3.2 Praca systemu

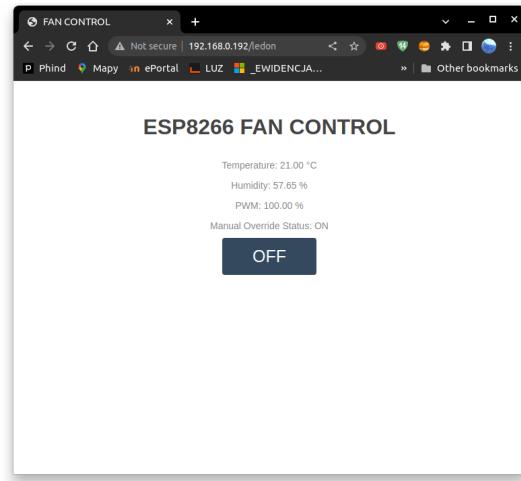
Po udanej inicjalizacji główna pętla programu odczytuje dane z czujnika, oblicza wartość wypełnienia PWM, a następnie czeka na odpowiedź modułu ESP8266. Dlatego w stronie został zawarty skrypt automatycznie odświeżający ją co kilka sekund, aby co ten zadany czas dokonywać pomiaru oraz wyświetlać nowe informacje na stronie.

6.4 Interfejs

Na stronie znajdują się 4 wskazania: temperatury, wilgotności, wypełnienia PWM i kontrolka trybu manualnego. Dane te są aktualizowane z każdym odświeżeniem strony. Poniżej widoczny jest duży przycisk służący do przełączania pomiędzy trybami automatycznym i manualnym. W trybie automatycznym wypełnienie PWM jest obliczane na podstawie aktualnych warunków, a w trybie manualnym zostaje ustawione na 100% do momentu powrotu do trybu automatycznego.



(a) Tryb automatyczny



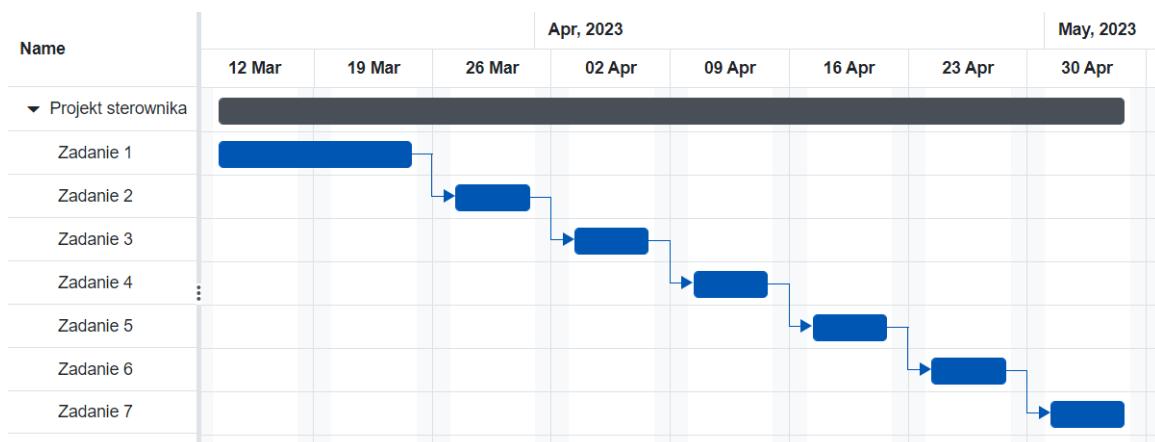
(b) Tryb manualny

7 Repozytorium

Link do repozytorium: https://github.com/kopiotrek/Sterowniki_robotow

8 Harmonogram pracy

1. Zaplanowanie projektu i zdefiniowanie wymagań użytkownika, na podstawie których powstanie sterownik wentylatora.
2. Wykonanie schematu podłączenia mikrokontrolera STM32L476, czujnika DHT11, wentylatora i modułu WiFi.
3. Zaimplementowanie komunikacji z czujnikiem DHT11 bez wykorzystania gotowych bibliotek
4. Wybór odpowiedniego modułu WiFi i sprawdzenie połączenia z mikrokontrolerem.
5. Zaprogramowanie mikrokontrolera STM32L476 umożliwiające odczyt temperatury i wilgotności z czujnika DHT11, sterowanie wentylatorem oraz przesyłanie danych do modułu WiFi.
6. Utworzenie interfejsu sieciowego, aby umożliwić zdalne sterowanie wentylatorem i wyświetlanie danych z czujnika.
7. Implementacja funkcji automatycznego sterowania wentylatorem na podstawie wskazań czujnika.



Rysunek 6: Diagram Gantta

8.1 Podział pracy

Michał Ragan	Piotr Koczy
Zaplanowanie projektu i zdefiniowanie wymagań użytkownika, na podstawie których powstanie sterownik wentylatora.	Zaplanowanie projektu i zdefiniowanie wymagań użytkownika, na podstawie których powstanie sterownik wentylatora.
Wybór odpowiedniego modułu WiFi i sprawdzenie połączenia z mikrokontrolerem.	Wykonanie schematu podłączenia mikrokontrolera STM32L476, czujnika DHT11, wentylatora i modułu WiFi.
Utworzenie interfejsu sieciowego, aby umożliwić zdalne sterowanie wentylatorem i wyświetlanie danych z czujnika.	Zaprogramowanie mikrokontrolera STM32L476 umożliwiające odczyt temperatury i wilgotności z czujnika DHT11, sterowanie wentylatorem oraz przesyłanie danych do modułu WiFi.
Implementacja funkcji automatycznego sterowania wentylatorem na podstawie wskazań czujnika.	Zaimplementowanie komunikacji z czujnikiem DHT11 bez wykorzystania gotowych bibliotek

9 Podsumowanie

Wszystkie zadania zostały pomyślnie zrealizowane. Użytkownik łączy się z układem poprzez WiFi, gdzie ma podgląd aktualnych wskazań temperatury, wilgotności i wypełnienia sygnału PWM. Przy pomocy przycisku może uruchomić tryb manualny, który ustawia pełną moc wentylatora aż do wyłączenia tego trybu. Dodatkowo włączenie trybu manualnego sygnalizowane jest zapaleniem się czerwonej diody na układzie.

Literatura

- [1] W. Domski. Sterowniki robotów, Laboratorium – Wprowadzenie, Wykorzystanie narzędzi STM32CubeMX oraz SW4STM32 do budowy programu mrugającej diody z obsługą przycisku. Mar. 2017.
- [2] M. Electronics. DHT11 Temperature Humidity Sensor features temperature humidity sensor complex with a calibrated digital signal output. 2023.
- [3] M. Szumski. Mikrokontrolery STM32 w systemach sterowania i regulacji. Sty. 2017.