DOKUMENTACJA PROJEKTU

Stacja Meteo z wykorzystaniem technologii IoT

Aleksandra Matysik 7424

Spis treści

Wstęp

OPIS PROJEKTU

Projekt **Stacji Meteo z wykorzystaniem Technologii IoT** opiera się na zaawansowanych technologiach mikrokontrolerów, czujników oraz komunikacji bezprzewodowej. Ma na celu dostarczenie precyzyjnych danych meteorologicznych w czasie rzeczywistym, które mogą być wykorzystane w różnych aplikacjach.

CEL PROJEKTU I JEGO ZNACZENIE

Głównym celem projektu jest stworzenie systemu umożliwiającego monitorowanie i analizę warunków atmosferycznych w sposób dokładny, niezawodny i łatwy w użyciu. Dzięki temu użytkownicy będą mogli uzyskać dostęp do aktualnych danych meteorologicznych na podstawie pomiarów przeprowadzanych przez stację meteo. Istotą projektu jest dostarczenie użytkownikom informacji na temat temperatury, wilgotności, natężenia światła itp., które są kluczowe dla wielu dziedzin życia, od codziennego planowania aktywności po profesjonalne aplikacje związane z monitorowaniem środowiska.

ZAKRES DOKUMENTACJI

Niniejsza dokumentacja techniczna obejmuje szczegółowy opis projektu stacji meteo opartej na technologii IoT. Zawiera ona opis sprzętu i oprogramowania użytych w projekcie, implementację poszczególnych komponentów, instrukcję użytkowania oraz podsumowanie projektu. Celem dokumentacji jest zapewnienie kompleksowego zrozumienia projektu dla osób zainteresowanych jego realizacją, dalszym rozwojem lub zastosowaniem w praktyce.

Opis Technologiczny

OPIS SPRZĘTOWY

System głównie opiera się na mikrokontrolerze STM32F303RE, który pełni rolę centralnej jednostki zarządzającej procesem zbierania i przetwarzania danych z różnych czujników oraz komunikacji z modułem WiFi. Do pomiaru parametrów atmosferycznych wykorzystywane są specjalistyczne czujniki, takie jak DS18B20 do pomiaru temperatury, oraz moduł ESP-01s do komunikacji bezprzewodowej.

Podstawowym założeniem projektu jest wykorzystanie architektury IoT, umożliwiającej zdalne monitorowanie i zarządzanie stacją meteo.

Specyfikacja Mikrokontrolera STM32F303RE

Mikrokontroler STM32F303RE to zaawansowany układ z rodziny mikrokontrolerów STM32F3xx, produkowany przez firmę STMicroelectronics.

Posiada rdzeń ARM Cortex-M4F z zegarem o maksymalnej częstotliwości pracy 72 MHz. Wyposażony jest w 512 KB pamięci Flash oraz 64 KB pamięci RAM. Posiada liczne interfejsy komunikacyjne, w tym UART, SPI, I2C, oraz interfejsy analogowe ADC i DAC. Posiada wbudowany moduł DMA, który umożliwia przesyłanie danych bez udziału procesora.

Zintegrowane peryferia umożliwiają wykorzystanie mikrokontrolera w różnorodnych aplikacjach, w tym w systemach IoT, kontrolerach urządzeń, czy systemach sterowania.

Specyfikacja Czujnika DS18B20

DS18B20 to cyfrowy czujnik temperatury, wyprodukowany przez firmę Maxim Integrated.

Mierzy temperaturę w zakresie od -55°C do +125°C z dokładnością do 0.0625°C. Komunikuje się za pomocą protokołu 1-Wire, co umożliwia obsługę wielu czujników na jednej linii danych. Posiada unikalny 64-bitowy numer seryjny, co umożliwia identyfikację każdego czujnika.

Dostępny jest w różnych obudowach, w tym w formie termometru cyfrowego, wodoodpornej kapsułki i w formie układu scalonego.

Specyfikacja Modułu WiFi ESP-01s

Moduł ESP-01s to moduł WiFi oparty na układzie ESP8266, produkowany przez firmę Espressif Systems.

Posiada wbudowany układ ESP8266EX, który zapewnia łączność WiFi w standardzie 802.11 b/g/n. Obsługuje różne tryby pracy, w tym klienta i punktu dostępowego (Access Point). Wyposażony jest w interfejsy UART, SPI i GPIO, umożliwiające łatwą integrację z innymi urządzeniami.

Może być programowany za pomocą polecenia AT lub za pomocą oprogramowania zewnętrznego, takiego jak ESP-IDF lub Arduino IDE.

Schemat Połączeń Mikrokontrolera z Peryferiami

Mikrokontroler STM32F303RE jest złączem kabla micro-AB USB.

Podłączenie czujnika DS18B20

Pin danych (DQ) czujnika DS18B20 jest podłączony do pinu GPIO **PC3** mikrokontrolera STM32F303RE.

Rezystor podciągający o wartości 4.7k ohm jest podłączony między linią danych czujnika a zasilaniem (**3.3V**), co zapewnia odpowiednie warunki dla komunikacji 1-Wire.

Zasilanie (VDD) oraz masa (**GND**) czujnika są podłączone odpowiednio do napięcia zasilania (3.3V) oraz masy mikrokontrolera.

Podłączenie modułu WiFi ESP-01s

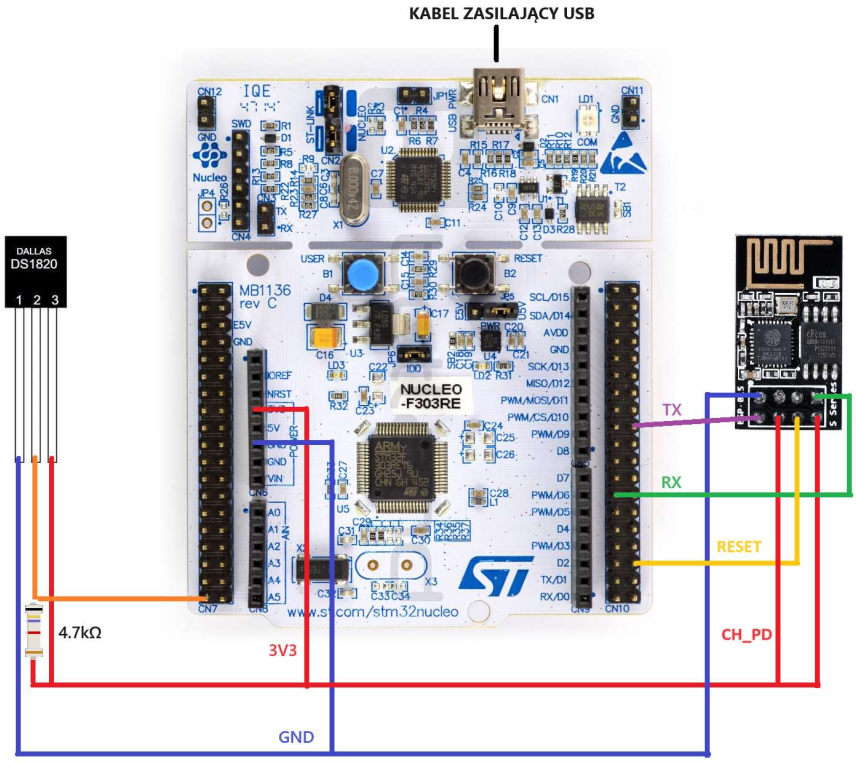
Moduł ESP-01s jest również zasilany z mikrokontrolera STM32F303RE poprzez napięcie zasilania (**3.3V**) oraz masę (**GND**).

RX modułu WiFi jest podłączony do pinu **PB10** czyli TX portu szeregowego UART3 mikrokontrolera, który służy do transmisji danych na modułu.

TX modułu WiFi jest podłączony do pinu **PB11** czyli RX portu szeregowego UART3 mikrokontrolera, który służy do odbierania danych z modułu.

Pin CH_PD modułu WiFi jest podłączony do napięcia zasilania (**3.3V**), ponieważ definiuje on stan aktywności modułu. W przyszłości zalecane podłączenie pinu CH_PD do któregoś z pinów GPIO, ponieważ umożliwi to wprowadzenie modułu w stan wysokiej impedancji (Hi-Z).

Dodatkowo, RESET pin i podłączony jest do pinu GPIO **PC4** i umożliwia reset modułu WiFi z poziomu programu. Przejście ze stanu niskiego w wysoki powoduje reset.



Rysunek 1. Schemat podłączenia peryferii do mikrokontrolera.

TECHNOLOGIE OPROGRAMOWANIA

Projekt wykorzystuje różnorodne technologie oprogramowania do zarządzania procesem zbierania, przetwarzania i prezentowania danych pomiarowych.

STM32CubeMX

STM32CubeMX jest narzędziem oferowanym przez STMicroelectronics do konfiguracji mikrokontrolerów STM32. Umożliwia ono graficzne konfigurowanie pinów mikrokontrolera, wybieranie odpowiednich peryferiów oraz generowanie kodu inicjalizacyjnego. W projekcie używamy STM32CubeMX do konfiguracji pinów mikrokontrolera STM32F303RE oraz do generowania kodu inicjalizacyjnego dla peryferiów takich jak UART i TIM.

Język programowania C/C++

Języki programowania C/C++ są powszechnie używane w programowaniu mikrokontrolerów STM32. C/C++ oferują elastyczność i wydajność potrzebną do tworzenia oprogramowania wbudowanego. W projektach mikrokontrolerowych języki te umożliwiają programistom pisanie kodu zarówno niskopoziomowego, jak i wysokopoziomowego, co pozwala na efektywne zarządzanie zasobami mikrokontrolera i realizację zaawansowanych funkcji. Programowanie w językach C/C++ pozwala na tworzenie wydajnego i niezawodnego oprogramowania, które może być łatwo

przenoszone między różnymi platformami sprzętowymi.

Biblioteka OneWire

Biblioteka OneWire to biblioteka programistyczna umożliwiająca komunikację z urządzeniami wykorzystującymi protokół 1-Wire. W projekcie korzystamy z tej biblioteki do obsługi komunikacji z cyfrowym czujnikiem temperatury DS18B20 poprzez interfejs GPIO mikrokontrolera STM32F303RE.

Język PHP

Język PHP (Hypertext Preprocessor) to język skryptowy ogólnego przeznaczenia, który jest powszechnie używany do tworzenia stron internetowych i aplikacji internetowych. W projekcie korzystamy z języka PHP do tworzenia prostych interfejsów użytkownika oraz do obsługi zapytań HTTP na serwerze, na którym hostowana jest aplikacja webowa prezentująca pomiary.

Baza Danych MySQL

MySQL to system zarządzania relacyjnymi bazami danych, który jest wykorzystywany do przechowywania, wyszukiwania i zarządzania danymi w bazach danych. W projekcie korzystamy z bazy danych MySQL do przechowywania historycznych danych pomiarowych, które są gromadzone przez system stacji meteo.

Struktura Oprogramowania

Oprogramowanie stacji meteo zostało zaprojektowane w sposób modułowy, co umożliwia łatwą modyfikację i rozszerzanie funkcjonalności. Główne zadania oprogramowania obejmują zbieranie danych z czujników, ich przetwarzanie oraz przesyłanie do zewnętrznego serwera. Następnie dane są przechowywane w bazie danych i prezentowane użytkownikowi za pomocą aplikacji webowej napisanej w PHP.

Oprogramowanie stacji meteo składa się z trzech głównych modułów.

MODUŁ POMIAROWY

Moduł Pomiarowy jest odpowiedzialny za interakcję z czujnikami i zbieranie danych pomiarowych. W tym module znajdują się procedury odczytu danych z czujników oraz ich konwersji do odpowiednich jednostek miar.

Inicjalizacja Czujników

Przy starcie systemu moduł pomiarowy inicjalizuje czujnik DS18B20.

Odczyt Danych

W regularnych odstępach czasu moduł odczytuje dane z czujników.

Przetwarzanie Danych

Surowe dane pomiarowe są przetwarzane i konwertowane na odpowiednie jednostki, (stopnie Celsjusza dla temperatury).

MODUŁ KOMUNIKACYJNY

Zarządza komunikacją między stacją meteo a zewnętrznymi serwisami internetowymi. Jest odpowiedzialny za przesyłanie danych pomiarowych na serwer.

Inicjalizacja Modułu WiFi

Moduł ESP-01s jest inicjalizowany i łączony z siecią WiFi.

Połączenie z serwerem

Konfiguracja połączenia z serwerem, umożliwiającym dostęp do bazy danych MySQL.

Przesyłanie Danych

Po przetworzeniu dane są przesyłane do bazy danych MySQL w określonych odstępach czasu.

MODUŁ AGREGACJI I PREZENTACJI

Opowiada za odbiór danych pomiarowych, ich przechowywanie oraz prezentację

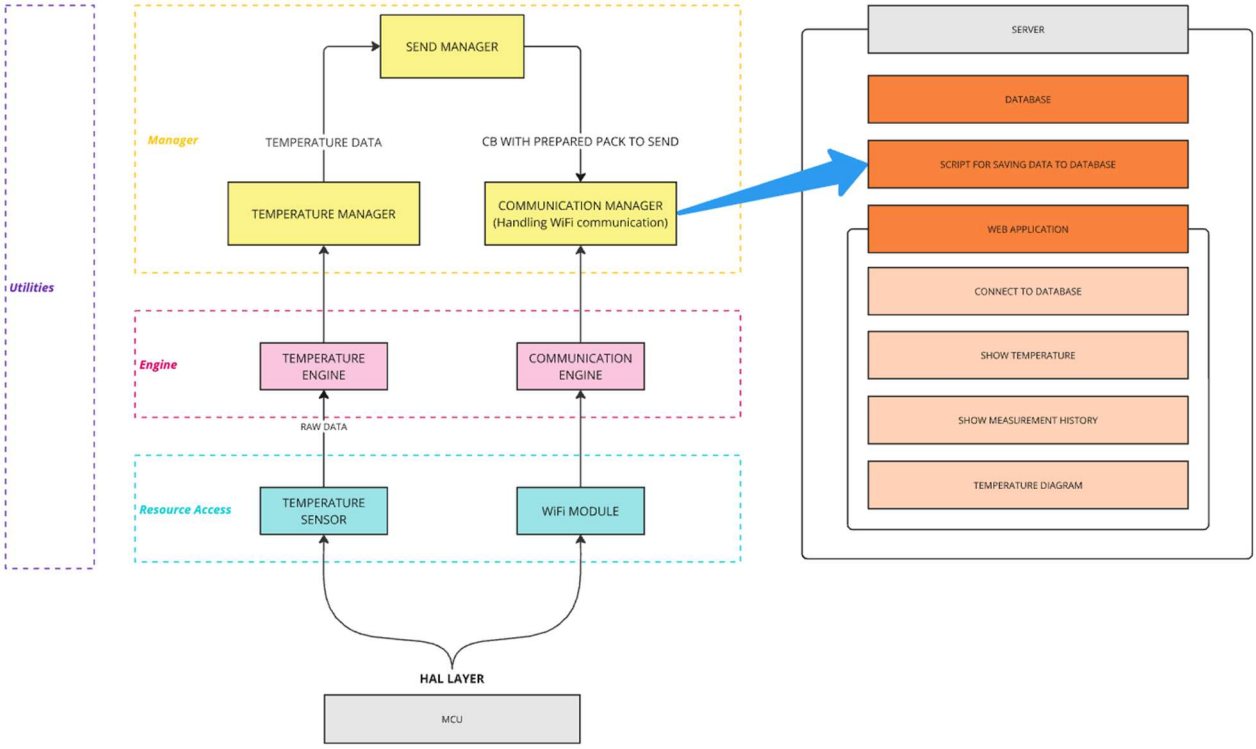
użytkownikowi.

Baza Danych MySQL

Otrzymane dane są zapisywane do bazy danych MySQL, która przechowuje historyczne dane pomiarowe.

Aplikacja PHP

Aplikacja webowa napisana w PHP pobiera dane z bazy danych MySQL i prezentuje je w przystępnej formie. Umożliwia użytkownikom przeglądanie aktualnych oraz historycznych danych pomiarowych za pomocą interfejsu webowego.



Rysunek 2. Uproszczony diagram projektu.

Implementacja

IMPLEMENTACJA ŚRODOWISKA

Konfiguracja środowiska obejmuje przygotowanie niezbędnych narzędzi i środowisk programistycznych do pracy z mikrokontrolerem STM32F303RE, modułem WiFi ESP-01s oraz serwerem i bazą danych.

Środowisko Embedded – Konfiguracja Projektu

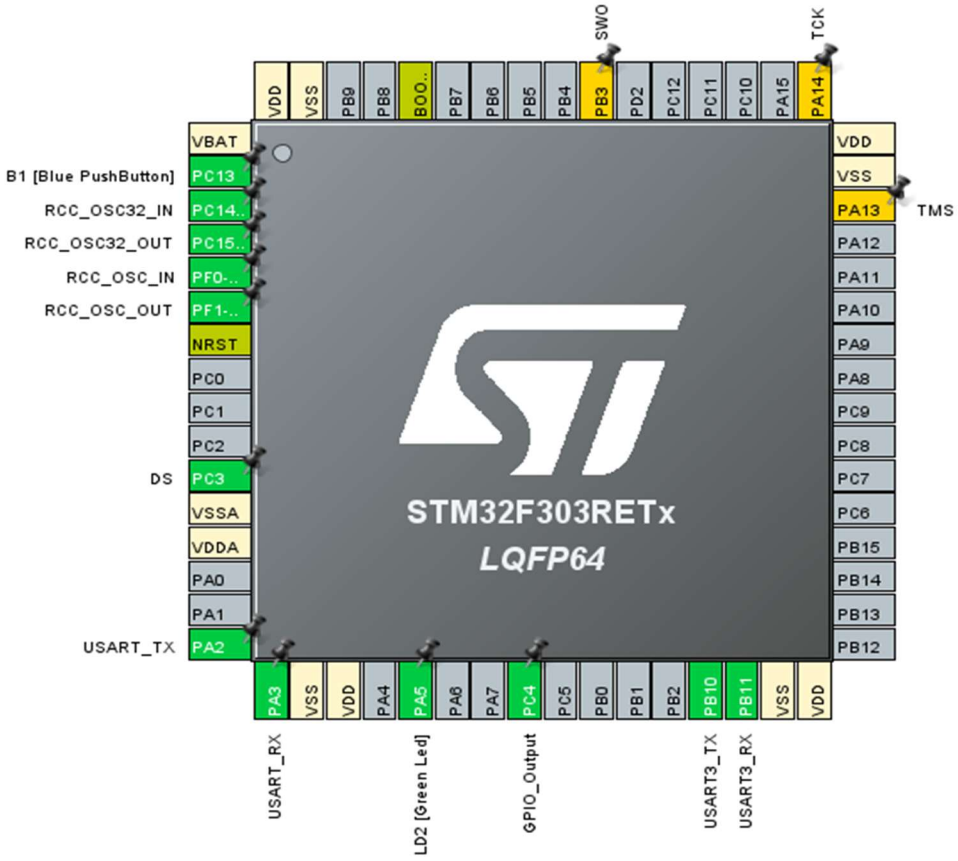
Konfiguracja części embedded projektu została wykonana za pomocą STM32CubeMX.

Implementacja STM32CubeMX

- Pobranie i zainstalowanie STM32CubeMX ze strony STMicroelectronics.
- Stworzenie nowego projekt dla mikrokontrolera **STM32F303RE**.
- Ustawienie taktowania procesora na **72 MHz**.
- Konfiguracja peryferiów
  - **UART2** (PA2 i PA3) - Ustawienia obejmują prędkość transmisji (*baud rate*) na 38400, 8 bitów danych, brak parzystości, 1 bit stopu. UART2 służy do debugu i przesyłania komunikatów diagnostycznych, które możemy odczytywać programem Hterm.
  - GPIO dla DS18B20 (**PC3**) - Konfiguracja pinu jako wejścia/wyjścia cyfrowego do komunikacji z DS18B20. Czujnik wymaga jednego pinu danych, który jest podłączony do portu GPIO mikrokontrolera. Ponadto, rezystor podciągający (pull-up) 4.7k ohm jest użyty między linią danych a zasilaniem (3.3V).
  - **UART3** (PB10 i PB11) - Ustawienia obejmują prędkość transmisji na 115200, 8 bitów danych, brak parzystości, 1 bit stopu. UART3 służy do komunikacji z modułem WiFi ESP-01s. Skonfigurowany do komunikacji z modułem WiFi ESP-01s. UART3 zapewnia dwukierunkową komunikację szeregowa, umożliwiając przesyłanie zebranych danych na serwer. Włączono również przerwania dla UART3, aby umożliwić odbieranie danych z modułu w trybie interrupt-driven, co poprawia responsywność systemu.
  - GPIO dla RESET Pin od ESP-01s (**PC4**) - Dodatkowy pin do zarządzania stanem resetu modułu ESP-01s. Aby wykonać reset modułu pin musi przejść z niskiego stanu logicznego w wysoki.
- Konfiguracja timerów MCU
  - **TIM4** (Prescaler = 71, Period = 0xffff-1) - Używany do generowania precyzyjnych opóźnień w mikrosekundach potrzebnych do komunikacji z czujnikiem DS18B20.
  - **TIM16** (Prescaler = 1151, Period = 62499) – Jego przerwania są używane do periodycznej pracy menedżera projektu.

- **TIM17** (Prescaler = 0, Period = 49999) – Dzięki przerwaniom jest używany do mierzenia czasu od ostatniej wiadomości wysłanej do modułu WiFi ESP-01s, co pozwala na implementację mechanizmu timeout, aby monitorować stabilność połączenia i reagować na potencjalne problemy z komunikacją.

- Po zakończeniu konfiguracji projektu w STM32CubeMX, wygenerowano kod startowy, który został zaimportowany do środowiska programistycznego STM32CubeIDE.



Rysunek 3. Wizualizacja konfiguracji mikrokontrolera za pomocą programu STM32CubeMX.

Serwer i Baza Danych

Proces konfiguracji serwera i bazy danych obejmuje instalację XAMPP, konfigurację bazy danych MySQL oraz stworzenie skryptu PHP, który będzie odpowiadał za odbieranie i przechowywanie danych z mikrokontrolera.

Wymagania wstępne

Przed przystąpieniem do konfiguracji, należy upewnić się, że na urządzeniu będącym jednostką przetwarzającą zainstalowane jest następujące oprogramowanie:

- **XAMPP** - Zintegrowane środowisko, które zawiera serwer Apache, bazę danych MySQL oraz interpreter PHP.

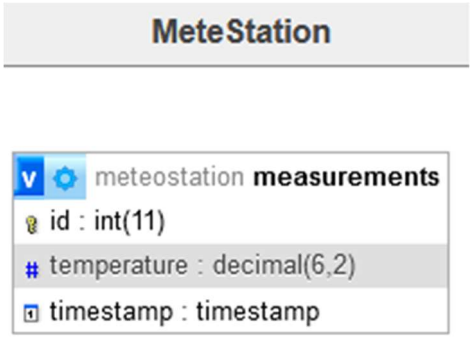
Implementacja Serwera

- Włączenie modułów **Apache** i **MySQL** za pomocą środowiska XAMPP.

Implementacja Bazy Danych MySQL

- Otwarcie aplikacji *phpMyAdmin*, dostępnej pod adresem http://localhost/phpmyadmin.
- Utworzenie nowej bazę danych o nazwie *'metoestation'* oraz tabelę *'measurements'*, używając poniższych zapytań SQL:

```
CREATE DATABASE meteostation;
CREATE TABLE measurements (
      id INT AUTO_INCREMENT PRIMARY KEY,
      temperature DECIMAL(6, 2) NOT NULL,
      timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```



Rysunek 4. Diagram bazy danych 'meteostation'.

Implementacja Skryptu Obsługi Danych

- Utworzenie pliku PHP *insert_data.php* w katalogu *htdocs*, który będzie odbierał dane z mikrokontrolera i wstawiał je do bazy danych.
- Testowanie skryptu PHP za pomocą narzędzia Postman, które wysyła żądań POST z przykładowymi danymi na serwer.

Użycie Postman do testowania skryptu

- Utworzenie nowego żądanie POST.
- Ustawienie adresu URL na *http://localhost/insert_data.php*.
- W zakładce "Body" wybranie opcję *x-www-form-urlencoded* i dodanie parametru *temperature*.
- Wysłanie żądania i sprawdzenie czy dane zostały poprawnie zapisane w bazie danych.

IMPLEMENTACJA CZĘŚCI EMBEDDED

Czujnik Temperatury DS18B20

DS18B20 to cyfrowy czujnik temperatury, szeroko stosowany w aplikacjach IoT i monitoringu środowiska ze względu na jego precyzję, prostotę obsługi i możliwość komunikacji za pomocą pojedynczej linii danych (1-Wire). Mierzy temperaturę w zakresie od -55°C do +125°C. Możliwość ustawienia dokładności pomiaru. Każdy czujnik posiada unikalny numer seryjny, co umożliwia użycie wielu czujników na jednej linii danych.

Podłączenie czujnika:

- Pin danych (DQ) - Podłączony do pinu PC3 mikrokontrolera Nucleo-F303RE.
- Rezystor podciągający - Rezystor 4.7k ohm podłączony między linią danych a zasilaniem (3.3V), co zapewnia odpowiednie warunki dla komunikacji 1-Wire.
- Zasilanie (VDD) i masa (GND) - Podłączone odpowiednio do 3.3V i GND.

Ze względu na złożoność komunikacji OneWire wykorzystana została gotowa biblioteka odpowiedzialnej za ten typ komunikacji.

Implementacja oprogramowania dla DS18B20:

1. **Obsługa opóźnień w mikrosekundach**

Korzystanie z przerwań TIM1 do generowania precyzyjnych opóźnień w mikrosekundach. Opóźnienia te są niezbędne do prawidłowej komunikacji OneWire z DS18B20, który wymaga dokładnych czasów sygnałów.

2. **Inicjalizacja czujnika**

Ustawienie rozdzielczości pomiarowej na 11 bitów, co daje precyzję 0,125°C i maksymalny czas konwersji 375 ms.

Wysyłanie sygnału resetu do DS18B20, a następnie oczekiwanie na odpowiedź (*presence pulse*), aby upewnić się, że czujnik jest podłączony i gotowy do komunikacji.

Wysłanie komendy *Skip ROM*. Ponieważ używamy tylko jednego czujnika, możemy pominąć identyfikację za pomocą unikalnego numeru seryjnego i bezpośrednio wysłać komendę pomiaru temperatury.

3. **Odczyt temperatury**

Wysłanie komendy *Convert T* do czujnika, co inicjuje proces pomiaru temperatury. Proces ten trwa około 750 ms przy maksymalnej rozdzielczości, przy ustawionej przez nas 375ms.

Po zakończeniu konwersji, mikrokontroler wysyła komendę *Read Scratchpad*, aby odczytać zapisane dane temperatury z pamięci czujnika.

Odczytane dane są konwertowane na temperaturę w stopniach Celsjusza. DS18B20 zapisuje

wynik jako 16-bitową liczbę, którą następnie należy przetworzyć na wartość temperatury.

4. Status pomiaru

Podczas pomiaru temperatury mogą wystąpić różne sytuacje, które wpływają na wynik. Dlatego wprowadzone zostały statusy pomiarów, aby lepiej zarządzać wynikami.

Podczas odczytu danych z czujnika, mikrokontroler sprawdza poprawność sumy kontrolnej (CRC) danych. Jeśli suma kontrolna jest niepoprawna, wynik pomiaru jest oznaczany jako *WRONG_CRC_CODE*.

Jeżeli zmierzona temperatura jest poza zakresem -55°C do +125°C, wynik jest oznaczany jako *OVER_TEMP*.

W sytuacji, gdy temperatura została zmierzona pomyślnie status zostanie ustawiony na *CORRECT_TEMP*.

5. Ustawienie stanu Hi-Z

Podczas dezaktywacji czujnika DS18B20, linia danych (DQ) zostaje ustawiona w stan Hi-Z (*High Impedance*). Stan Hi-Z oznacza, że pin jest wyłączony z obwodu i zachowuje się jak wejście o wysokim oporze, co oznacza brak aktywnego źródła sygnału.

Ustawienie linii danych w stanie Hi-Z eliminuje aktywne źródło sygnału oraz pozwala na zmniejszenie poboru prądu przez czujnik, co przyczynia się do oszczędności energii.

Kod odpowiedzialny za obsługę czujnika DS18B20 można znaleźć w plikach:

- *RA_ds18b20.hpp*,
- *RA_ds18b20.cpp*,
- *E_TemperatureSensor.hpp*,
- *E_TemperatureSensor.cpp*.

Moduł WiFi ESP-01s

ESP-01s to moduł WiFi oparty na chipsecie ESP8266, który zapewnia bezprzewodową komunikację internetową. Jest wysoce popularny w projektach IoT ze względu na swoją niską cenę i niewielkie rozmiary.

Obsługuje protokoły TCP/IP, umożliwiając integrację z różnymi usługami internetowymi. W projekcie zdecydowaliśmy się na korzystanie z protokołu MQTT do komunikacji z brokerem. MQTT jest lekkim protokołem komunikacyjnym, który umożliwia przesyłanie danych w formie wiadomości publikacji-subskrypcji.

Moduł ESP-01s jest konfigurowany i kontrolowany za pomocą poleceń AT. Te proste komendy umożliwiają zmianę ustawień, inicjowanie połączenia z siecią WiFi oraz wysyłanie danych.

Podłączenie modułu:

- Piny TX i RX - Podłączone do pinów PB10 i PB11, czyli pinów RX i TX UART3 mikrokontrolera. Odpowiednio na krzyż RX-TX, TX-RX.
- Pin CH_PD - Podłączony do pinu PC4. Pin ten odpowiada za włączanie/wyłączanie modułu. Przejście pinu na stan LOW skutkuje wyłączeniem modułu.
- Zasilanie (VDD) i masa (GND) - Podłączone odpowiednio do 3.3V i GND.

Implementacja oprogramowania dla ESP-01s:

1. **Obsługa przerwań UART**

Obsługa asynchronicznej komunikacji z mikrokontrolerem przez interfejs UART3 i jego przerwania jest kluczowym elementem w komunikacji między mikrokontrolerem Nucleo-F303RE a modułem WiFi ESP-01s.

Kiedy moduł WiFi ESP-01s wysyła dane, mikrokontroler przechodzi w stan przerwania UART3, co wyzwala odpowiednią funkcję obsługi przerwań. Jej głównym zadaniem jest monitorowanie, analiza i odpowiedź na dane przychodzące, zapewniając niezawodną i efektywną wymianę informacji między urządzeniami.

2. **Obsługa TimeOut dla Braku Aktywności**

Implementacja mechanizmu timeout'u, który ustawia flagę przetrzwarzania danych, jeśli dostatecznie długo nic nie przyszło z Modułu. Zapobiega to czekaniu w nieskończoność, w przypadku braku odpowiedzi z modułu.

3. **Inicjalizacja modułu poleceniami AT**

Inicjalizacja modułu WiFi ESP-01s wykonywana jest za pomocą poleceń AT. Poprzez wykonanie odpowiednich poleceń AT, konfigurujemy parametry modułu oraz nawiązujemy połączenie z siecią WiFi, co umożliwia późniejszą wymianę danych.

Polecenie *AT+CWMODE* ustawiamy tryb pracy modułu jako klient.

Ustawiamy adres MAC modułu na określoną wartość. Jest to istotne w przypadku konieczności unikalnej identyfikacji modułu w sieci. Umożliwia to polecenie *AT+CIPSTAMAC*.

Po każdym wykonaniu polecenia AT sprawdzamy stan sukcesu wykonania. Jeśli stan ten jest negatywny, wykonujemy reset modułu.

4. **Inicjalizacja połączenia z siecią WiFi**

Za pomocą polecenia *AT+CWJAP* nawiązujemy połączenie z określoną siecią WiFi. W ciągu znaków są zawarte nazwa SSID sieci oraz hasło do niej. Te dane są wcześniej zdefiniowane w kodzie. Jednak docelowo warto zawrzeć je w pliku konfiguracyjny.

Jeśli moduł pomyślnie nawiąże połączenie z siecią WiFi, następuje pobranie lokalnego adresu IP modułu ESP-01s w sieci WiFi (*AT+CIFSR*).

Następnie moduł ESP-01s jest ustawiany w tryb multiplex, co pozwala na jednoczesne

obsługiwania wielu połączeń. Służy do tego polecenie *AT+CIPMUX*.

5. **Inicjalizowanie połączenia z serwerem**

Po nawiązaniu połączenia z siecią WiFi, moduł ESP-01s musi nawiązać połączenie z serwerem, do którego będą wysyłane dane. Wykorzystujemy polecenie *AT+CIPSTART* do inicjalizacji połączenia TCP z serwerem.

6. **Wysyłanie danych na serwer**

Po nawiązaniu połączenia z serwerem, dane z czujników mogą być wysyłane za pomocą żądań **HTTP POST**. Wykorzystujemy polecenie *AT+CIPSEND* do określenia długości wiadomości, a następnie wysyłamy właściwą wiadomość zawierającą dane.

7. **Reset modułu**

Program zapewnia funkcję resetującą moduł ESP-01s, która steruje napięciem pinu RESET modułu, oraz resetuje wszystkie zmienne stanu związane z połączeniem Wi-Fi i serwerem.

Kod odpowiedzialny za obsługę modułu WiFi można znaleźć w plikach:

- *RA_esp_01s.hpp*,
- *RA_esp_01s.cpp*,
- *E_WiFiModule.hpp*,
- *E_WiFiModule.cpp*.

Menedżer

Menedżer Pomiarów pełni kluczową rolę w projekcie, zarządzając periodycznym zbieraniem danych z różnych czujników, ich przetwarzaniem i kierowaniem do modułu ESP-01s w formacie JSON.

Jest on odpowiedzialny za efektywne zarządzanie procesem pomiarów oraz agregację zebranych danych w odpowiedni format, co umożliwia ich dalszą transmisję poprzez moduł ESP-01s.

Implementacja Menedżera obejmuje

1. **Cykliczne pobieranie danych**

Menedżer cyklicznie pobiera dane z czujników, aktualnie tj. czujnika temperatury.

Menedżer wyposażony jest w mechanizmy obsługi błędów, które pozwalają na skuteczną reakcję na ewentualne problemy podczas zbierania danych, takie jak utrata połączenia z czujnikiem lub niestabilność transmisji. Jeśli trzykrotnie z rzędu zostaną odczytane nieprawidłowe dane, Menedżer automatycznie zleca zresetowanie czujnika, aby przywrócić prawidłowe działanie systemu.

2. **Przetwarzanie danych**

Zebrane dane są przetwarzane w celu ich dostosowania do formatu JSON, który jest akceptowany przez moduł ESP-01s.

3. Kierowanie Danych do Modułu WiFi

Po przetworzeniu, dane w formacie JSON są kierowane do modułu ESP-01s, skąd zostaną wysłane na serwer.

Kod odpowiedzialny za obsługę modułu WiFi można znaleźć w plikach:

- *Manager.hpp*,
- *Manager.cpp*.

Końcowe działanie embedded

Dzięki narzędziom diagnostycznym takim jak Hterm, można zobaczyć poprawne działanie projektu embedded. Na załączonych zrzutach ekranu widać poprawne wykonanie programu.

```
DS18B20 INIT

ONEWIRE INIT

SEARCHING ONEWIRE DEVICE

DS18B20 INIT

ONEWIRE INIT

SEARCHING ONEWIRE DEVICE

AT+RST
{}

{AT+RST␍␍

OK
WIFI DISCONNECT
}

{
 ets Jan  8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 27728, room 16
tail 0
chksum 0x2a
load 0x3ffe8000, len 2124, room 8
tail 4
chksum 0x07
load 0x3ffe8850, len 9276, room 4
tail 8
chksum 0xba
csum 0xba
}

AT
{ooonololooo␍oorlooooololooo␍␍l`ooooool␍oooo␍␍ll`orlooob
ready
}
```

```
AT+CWMODE=1
{AT␍␍

OK
}

AT+CWQAP
{AT+CWMODE=1␍␍

OK
}

AT+RST
{AT+CWQAP␍␍

OK
}

{AT+RST␍␍

OK
}
{
 ets Jan  8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 27728, room 16
tail 0
chksum 0x2a
load 0x3ffe8000, len 2124, room 8
tail 4
chksum 0x07
load 0x3ffe8850, len 9276, room 4
tail 8
chksum 0xba
csum 0xba
}
AT+CIPSTAMAC="A4:CF:12:EF:A7:40"
{▫▫s▫p▫▫▫▫▫▫▫▫␍▫l`▫▫▫r▫p▫▫▫▫▫▫▫␍␍␍l`▫▫▫▫▫▫▫l␍▫▫▫▫␍␍ll`▫rl▫▫▫▫▫

ready
}

Trying to connect WIFI network...

AT+CWJAP="HUAWEI-2.4G-h5Z9","EW8ksB63"
{AT+CIPSTAMAC="A4:CF:12:EF:A7:40"␍␍

OK
}

{AT+CWJAP="HUAWEI-2.4G-h5Z9","EW8ksB63"␍␍
}

{WIFI CONNECTED
}

{WIFI GOT IP
}
```

```
AT+CIFSR
{
OK
}

AT+CIPMUX=1
{AT+CIFSR
+CIFSR:STAIP,"192.168.18.18"
+CIFSR:STAMAC,"a4:cf:12:ef:a7:40"

OK
}
WIFI init done!

Trying to connect to server...

AT+CIPSTART=0,"TCP","192.168.18.5",80
{AT+CIPMUX=1

OK
}

Connection to the server completed!

READ TEMPERATURE

START

READ SCRATCHPAD

26.38

Send...

AT+CIPSEND=0.156
{AT+CIPSEND=0,156

OK
> }

POST /meteo_station/insert_data.php HTTP/1.1
Host: 192.168.18.5
Content-Length: 17
Content-Type: application/x-www-form-urlencoded

temperature=26.38

{
Recv 156 bytes

SEND OK

+IPD,0,229:HTTP/1.1 200 OK
Date: Wed, 03 Jul 2024 19:05:49 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
X-Powered-By: PHP/8.2.12
Content-Length: 31
Content-Type: text/html; charset=UTF-8

New record created successfully}
```

IMPLEMENTACJA APLIKACJI WEBOWEJ

Aplikacja webowa składa się z dwóch głównych elementów: skryptów PHP odpowiedzialnych za komunikację z bazą danych oraz frontendu opartego na HTML/CSS/JavaScript do prezentacji danych pomiarowych.

Struktura Aplikacji

Aplikacja webowa składa się z następujących głównych komponentów:

- Interfejs użytkownika:
  index.php: Główny plik HTML generujący interfejs użytkownika. Wyświetla aktualne pomiary temperatury oraz historię pomiarów w formie listy. Zawiera skrypty JavaScript do inicjalizacji wykresu oraz obsługi dynamicznego odświeżania danych.
  style.css: Arkusz stylów odpowiedzialny za estetykę i układ interfejsu użytkownika.
- Komunikacja z backendem:
  db_connect.php: Plik PHP odpowiedzialny za nawiązanie połączenia z bazą danych MySQL oraz pobieranie danych pomiarowych.
  insert_data.php: Plik PHP odpowiedzialny za przyjmowanie nowych danych pomiarowych wysyłanych z mikrokontrolera STM32F303RE przez moduł WiFi ESP-01s. Obsługuje żądania POST, w których przesyłane są pomiary temperatury. Po odebraniu danych, plik ten dokonuje insercji nowego pomiaru do bazy danych MySQL.
- Wykres Temperatury:
  Wykorzystanie biblioteki **Chart.js** do generowania interaktywnego wykresu liniowego prezentującego historię pomiarów temperatury.

Zasada Działania

Aplikacja webowa pobiera dane pomiarowe z bazy danych MySQL za pomocą zapytań PHP. Dane są następnie przetwarzane i prezentowane w interfejsie użytkownika w postaci aktualnych wartości pomiarowych oraz listy ostatnich pomiarów. Dodatkowo, wykorzystanie wykresu liniowego pozwala na wizualizację zmian temperatury w czasie.

Integracja Z Układem Mikrokontrolera

Aplikacja webowa integruje się z modułem mikrokontrolera STM32F303RE oraz modułem WiFi ESP-01s poprzez bazę danych MySQL. Dane pobierane przez mikrokontroler są regularnie aktualizowane w bazie danych, co umożliwia ich natychmiastowe odwzorowanie w interfejsie użytkownika aplikacji webowej.

Debugowanie

Podczas implementacji projektu wykorzystywane są różne narzędzia do debugowania, które umożliwiają monitorowanie działania mikrokontrolera, analizę poprawności sygnałów oraz działanie skryptów PHP.

WYSYŁANIE KOMUNIKATÓW PRZEZ UART2

Komunikaty diagnostyczne są wysyłane przez UART2, co pozwala na monitorowanie działania programu w czasie rzeczywistym. Makro odpowiedzialne za wysyłkę tekstów na port szeregowy znajduje się w pliku *utilities.hpp*. Funkcję można wywołać w dowolnym miejscu kodu, aby wysłać komunikat do narzędzi diagnostycznych, takich jak Hterm.

UŻYCIE ANALIZATORA STANÓW LOGICZNYCH

Analizator stanów logicznych, np. Saleae Logic Analyzer, jest wykorzystywany do weryfikacji poprawności sygnałów na pinach mikrokontrolera. Umożliwia on obserwację zmian stanów logicznych, co jest pomocne w diagnozowaniu problemów z komunikacją i przepływem danych.

Rysunek 5. Przykład konfiguracji programu Hterm.

POSTMAN DO DEBUGOWANIA ŻĄDAŃ POST ZE SKRYPTÓW PHP

Podczas tworzenia skryptu PHP odpowiedzialnego za obsługę żądań HTTP typu POST, używany był program Postman do testowania i debugowania. Postman umożliwił symulację żądań POST przy różnych temperaturach, aby zweryfikować funkcjonalność skryptu i obsłużyć wszelkie błędy, które mogą wystąpić podczas operacji na bazie danych.

Instrukcja Użytkowania Aplikacji Webowej

Instrukcja użytkowania aplikacji skierowana jest do użytkowników końcowych, którzy będą korzystać z systemu monitorowania i prezentacji danych pomiarowych za pomocą aplikacji webowej. Aplikacja umożliwia śledzenie aktualnych pomiarów temperatury oraz przeglądanie historii pomiarów w formie wykresów i listy.

WYMAGANIA WSTĘPNE

Przed rozpoczęciem użytkowania aplikacji webowej, upewnij się, że spełnione są poniższe wymagania:

- **Urządzenia**
  Komputer stacjonarny, laptop, tablet lub smartfon z przeglądarką internetową.
- **Połączenie internetowe**
  Stabilne połączenie internetowe zapewniające dostęp do aplikacji webowej.
- **Konto użytkownika**
  Aplikacja webowa nie wymaga logowania, więc każdy użytkownik może uzyskać dostęp do danych pomiarowych bez konieczności posiadania konta.

KORZYSTANIE Z APLIKACJI WEBOWEJ

Aby rozpocząć korzystanie z aplikacji webowej, postępuj zgodnie z poniższymi krokami:
Otwarcie Strony Aplikacji:

Uruchom przeglądarkę internetową na swoim urządzeniu.

W pasku adresu wpisz adres URL aplikacji webowej lub kliknij na odpowiedni skrót na pulpicie.

Interfejs Użytkownika

Po załadowaniu strony głównej aplikacji zobaczysz aktualne pomiary temperatury wraz z wykresem prezentującym ostatnią godzinę pomiarów. Poniżej znajduje się lista ostatnich 30 pomiarów wraz z ich szczegółami czasowymi.

Aktualizacja Danych

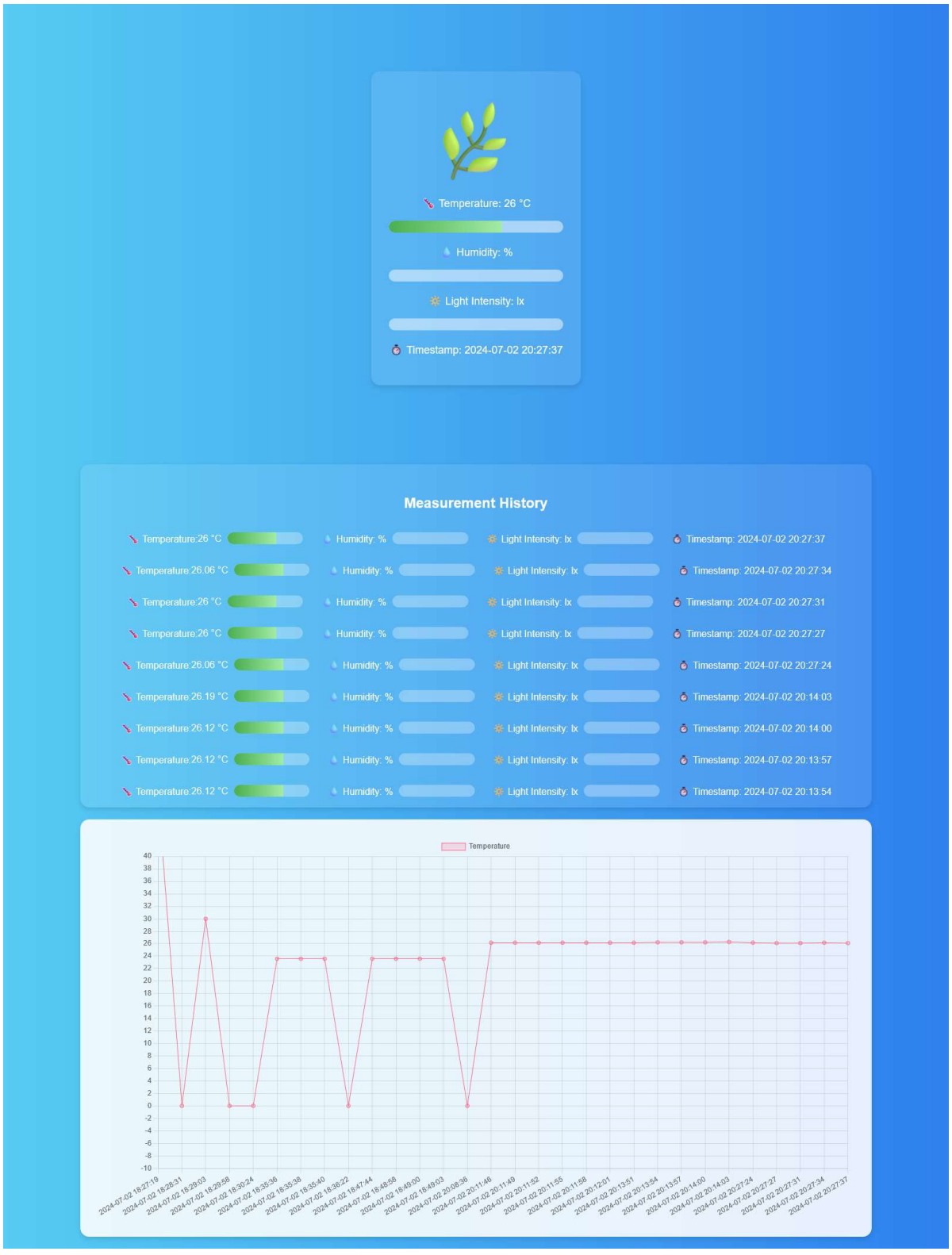
Aplikacja automatycznie odświeża dane co 20 sekund, zapewniając aktualne informacje bez konieczności ręcznego odświeżania strony.

Przeglądanie Historii Pomiarów

Aby zobaczyć szczegółową historię pomiarów, przewiń w dół strony do sekcji "Historia pomiarów". Tutaj możesz zobaczyć szczegółowe dane pomiarowe z ostatnich 30 odczytów wraz z czasem ich wykonania.

Interakcja z Wykresem

Wykres temperatury pozwala na przesuwać się po danych, w celu dokładniejszej analizy poszczególnych punkty pomiarowe.



Rysunek 6. Aplikacja webowa projektu.

Kierunek Dalszych Działań

INTEGRACJA DODATKOWYCH CZUJNIKÓW

Rozbudowa systemu o obsługę dodatkowych czujników pomiarowych, takich jak wilgotność, ciśnienie atmosferyczne, natężenie światła oraz czujniki wilgotności gleby, w celu uzyskania bardziej kompleksowej wiedzy na temat warunków atmosferycznych i środowiskowych.

UDOSKONALENIE MODUŁU WIFI ORAZ KOMUNIKACJI

Rozbudowa funkcjonalności ESP-01s. Umożliwienie przechodzenia w stan wysokiej impedencji (Hi-Z) dzięki podłączeniu pinu CH_PD do jednego z pinów PGIO mikrokontrolera.

Wprowadzenie komunikacji za pomocą protokołu MQTT.

POBIERANIE PROGNOZY POGODY

Implementacja mechanizmu pobierania prognozy pogody z zewnętrznych źródeł internetowych, co umożliwi użytkownikom uzyskanie bardziej wszechstronnych informacji na temat warunków pogodowych oraz ich prognozy na przyszłość.

ZABEZPIECZENIA SYSTEMU

Wprowadzenie mechanizmów zabezpieczeń, takich jak uwierzytelnianie, szyfrowanie danych oraz monitorowanie aktywności systemu, aby zapewnić bezpieczną i niezawodną komunikację między stacją meteo a serwerem oraz aplikacją mobilną.

PRZEJŚCIE NA TECHNOLOGIĘ NodeRED

Migracja części funkcjonalności systemu na technologię NodeRED, co umożliwi bardziej elastyczne i intuicyjne tworzenie oraz zarządzanie interfejsem użytkownika, automatyzację procesów oraz integrację z innymi platformami i usługami IoT.

OBSŁUGA EKRANU LCD

Rozszerzenie funkcjonalności stacji meteo o obsługę ekranu LCD umożliwi bezpośrednie wyświetlanie danych pomiarowych bez konieczności korzystania z zewnętrznego urządzenia. Ekran LCD zapewni użytkownikom szybki i wygodny dostęp do najważniejszych informacji dotyczących aktualnych warunków atmosferycznych.