

PROJEKT: CZUJNIK HAŁASU

KURS: INTERNET RZECZY I INTELIGENTNE SYSTEMY WBUDOWANE

Authors

Michał Czosnyka
Przemysław Domagała



EAIiIB / Katedra Informatyki Stosowanej
Akademia Górniczo-Hutnicza im. Stanisława Staszica w
Krakowie
Kraków, Polska

20 Stycznia 2024

Spis treści

1	Założenia projektu	1
1.1	Funkcjonalności	1
2	Wykorzystane czujniki i elementy wykonawcze	2
2.1	ESP32-WROOM-32	2
2.2	Mikrofon MAX9814	2
2.3	Wyświetlacz OLED	3
2.4	Przycisk	3
2.5	Dioda LED	3
2.6	Pozostałe komponenty	3
3	BLE	4
4	MQTT	5
5	Aplikacja Serwerowa	6

1. Założenia projektu

Założeniem projektu jest stworzenie czujnika hałasu, pozwalającego mierzyć obecny, średni oraz maksymalny hałas w danych jednostkach czasowych.

1.1 Funkcjonalności

- Pomiar aktualnego, średniego i maksymalnego poziomu hałasu w dB.
- Wyświetlanie aktualnego oraz średniego poziomu hałasu na wyświetlaczu OLED.
- Konfiguracja urządzenia za pomocą BLE (Bluetooth Low Energy).
- Podgląd i wizualizacja pomiarów oraz konfiguracja urządzenia przy pomocy aplikacji serwerowej.
- Komunikacja urządzenie-serwer przy pomocy protokołu MQTT.
- Świecąca się dioda LED po przekroczeniu danego progu hałasu.
- Wyświetlenie adresu MAC na wyświetlaczu po naciśnięciu przycisku w celu sparowania urządzenia z aplikacją serwerową.

2. Wykorzystane czujniki i elementy wykonawcze

W tej sekcji szczegółowo opisano każdy komponent użyty w projekcie.

2.1 ESP32-WROOM-32

- Główna jednostka obliczeniowa
- Obsługa BLE, WiFi, protokołu MQTT, nadzorowanie pozostałych komponentów oraz obliczanie hałasu na podstawie napięcia wytworzonego przez mikrofon.
- Software napisany przy pomocy frameworku ESP-IDF dla języka C.
- Zasilanie na USB-C
- Przycisk BOOT uruchamiający tryb konfiguracyjny urządzenia oraz przycisk EN resetujący urządzenie.
- Dwie wbudowane diody. Czerwona - urządzenie podłączone do zasilania, niebieska - urządzenie rozłączone z WiFi, próbuje ponownie nawiązać połączenie.

2.2 Mikrofon MAX9814

- Moduł wyposażony w mikrofon elektretowy o zakresie od 20 Hz do 20 kHz oraz wzmacniacz z automatyczną regulacją wzmocnienia.
- Moduł w zależności od dźwięku w otoczeniu generuje odpowiednie napięcie na pinie, które przy pomocy odpowiedniej funkcji przeliczane jest na decybele.
- Protokół: analogowy.

2.3 Wyświetlacz OLED

- Rozmiar wyświetlacza: 128x64px.
- Sterownik: SSD1306.
- Domyślnie na ekranie wyświetla się informacja o aktualnym poziomie hałasu oraz o średnim poziomie hałasu od ostatniego resetu urządzenia. Po kliknięciu przycisku na ekranie wyświetla się informacja o adresie MAC urządzenia. Po ponowne naciśnięcie guzika powoduje powrót do wyświetlania informacji o hałasie.
- Istnieje możliwość konfiguracji wyświetlacza przez aplikację serwerową: kontrast, inwersja kolorów, przesunięcie tekstu w poziomie oraz pionie.
- Protokół: I2C.

2.4 Przycisk

- Przycisk, którego naciśnięcie zmienia zawartość wyświetlaną na ekranie: hałas/adres MAC (szczegółowo opisane w punkcie Wyświetlacz Oled).
- Protokół: Cyfrowy sygnał na jednym z pinów.

2.5 Dioda LED

- Czerwona dioda LED, zaświecająca się, gdy hałas w danym momencie w otoczeniu przekroczy próg hałasu. Domyślnie próg hałasu ustawiony jest na 60dB, jednak może być skonfigurowany za pomocą aplikacji serwerowej.
- Protokół: Cyfrowy sygnał z jednego z pinów.

2.6 Pozostałe komponenty

- Breadboard.
- Kable męsko-męskie.
- Kabel USB-C do zasilania.

3. BLE

BLE (Bluetooth Low Energy) w projekcie służy do konfiguracji urządzenia. Użytkownik za pomocą aplikacji mobilnej lub innego klienta BLE może przesłać następujące dane:

- SSID sieci WiFi.
- Hasło sieci WiFi.

Proces konfiguracji:

1. Po uruchomieniu urządzenia użytkownik ma 20 sekund by wejść w tryb konfiguracyjny.
2. Naciśnięcie przycisku BOOT na płycie uruchamia tryb konfiguracyjny.
3. Przy pomocy klienta BLE należy podać informacje o SSID lub/i hasle Wifi. Dodatkowo można je też odczytać.
4. Po podaniu obu parametrów lub ponownym naciśnięciu przycisku BOOT urządzenie wychodzi z trybu konfiguracji i zapisuje parametry w pamięci NVS.
5. Odliczanie się resetuje i po 20 sekundach urządzenie przechodzi w stan pracy, łącząc się z WiFi i wysyłając informacje o pomiarach na broker MQTT.

4. MQTT

Czujnik wykorzystuje protokół MQTT do komunikacji z aplikacją serwerową. Wymiana danych opiera się na trzech topicach, które umożliwiają odczyt informacji w aplikacji, pobieranie bieżącej konfiguracji oraz wysyłanie nowych ustawień do czujnika.

Struktura topiców

- **noise_detector_web/<mac_address>/noise_data** Topic służy do przesyłania danych pomiarowych. Czujnik wysyła:
 - Średni poziom hałasu z ostatnich 15 sekund (w decybelach).
 - Maksymalny poziom hałasu z ostatnich 15 sekund (w decybelach).
 - Znacznik czasu pomiaru (timestamp).
- **noise_detector_web/<mac_address>/sensor_info** Topic zawiera informacje o aktualnej konfiguracji czujnika, w tym:
 - Próg hałasu (w decybelach), po przekroczeniu którego zapala się dioda.
 - Ustawienia wyświetlacza:
 - * Inwersja kolorów.
 - * Przesunięcie tekstu w osiach X i Y (w pikselach).
 - * Poziom kontrastu.
- **noise_detector_web/<mac_address>/configuration_data** Topic przeznaczony do wysyłania nowych ustawień czujnika z poziomu aplikacji serwerowej. Parametry konfigurowane za pomocą tego tematu odpowiadają informacjom dostępnym w topicu **noise_detector_web/<mac_address>/sensor_info**.

5. Aplikacja Serwerowa

Aplikacja serwerowa pozwala właścicielom czujników hałasu na zarządzanie urządzeniami i ich konfigurację. Użytkownicy mogą dodawać czujniki za pomocą adresu MAC, przekazywać prawa do zarządzania innym osobom oraz przeglądać dane pomiarowe w czasie rzeczywistym lub z wybranego okresu w przeszłości. Dostępne informacje obejmują aktualny poziom hałasu, średnie i maksymalne wartości z ostatnich 15 sekund, a także z wybranego zakresu czasu. Użytkownicy mają wgląd w pełną historię komunikatów zarejestrowanych przez czujnik. Aplikacja umożliwia zarządzanie wieloma urządzeniami za pośrednictwem jednego konta, a konfiguracja obejmuje ustawienia progu hałasu oraz parametry wyświetlacza, takie jak kontrast, pozycjonowanie tekstu czy inwersja kolorów.

Wykorzystanie technologie:

- Backend
 - Python(Flask)
- Frontend
 - HTML
 - CSS
 - Javascript
- Funkcję Bazy danych spełniają pliki JSON