

Projekt i realizacja autonomicznego rejestratora parametrów ruchu GPS z interfejsem bezprzewodowym opartym o układ ESP32

Paweł

January 9, 2026

Contents

1	Wstęp	2
1.1	Koncepcja architektury systemu telemetrycznego	2
2	Projekt sprzętowy	2
2.1	Elementy układu	2
2.2	Konfiguracja interfejsów i połączeń	3
2.3	Projekt obudowy	4
3	Przegląd technologii i narzędzi	6
3.1	Rozwój IoT i miniaturyzacja	6
3.2	Telemetria i przetwarzanie danych	6
3.3	Architektura ESP32	6
3.4	System GNSS i protokół NMEA 0183	6
3.5	Magistrale komunikacyjne w systemach wbudowanych	7
4	Implementacja oprogramowania	8
4.1	Opis funkcjonalności i obsługa aplikacji	8
4.1.1	Ekran główny i parametry na żywo	8
4.1.2	Zarządzanie plikami i historia tras	11
4.2	Szczegóły techniczne implementacji	15
5	Testy i analiza wyników	16
5.1	Testy modułu GPS (Cold Start vs Hot Start)	16
5.2	Analiza jakości śladu w ruchu	17
5.3	Bilans energetyczny i czas pracy	17
6	Podsumowanie i wnioski	17
6.1	Osiągnięcia	17
6.2	Ograniczenia i plany rozwojowe	18

1 Wstęp

Rozwój technologii Internetu Rzeczy (IoT – Internet of Things) pozwolił na zmniejszenie systemów telemetrycznych. Jeszcze dziesięć lat temu potrzebna była duża infrastruktura sprzętowa. Celem tej pracy jest zaprojektowanie i zbudowanie urządzenia typu “GPS Tracker”. Urządzenie ma rejestrować trasę, monitorować parametry ruchu i pokazywać dane w czasie rzeczywistym.

Głównym założeniem projektu było stworzenie systemu autonomicznego, zasilanego bateryjnie, który łączy w sobie funkcjonalność klasycznego rejestratora danych (tzw. "czarnej skrzynki") z nowoczesnym interfejsem użytkownika dostępnym przez przeglądarkę internetową (Web Interface).

1.1 Koncepcja architektury systemu telemetrycznego

Największy problem przy tworzeniu przenośnych rejestratorów danych to znalezienie balansu między siłą obliczeń a zużyciem prądu. Stare, proste rejestratory danych tylko zapisują dane i nic więcej nie robią. Natomiast nowe rozwiązania IoT chcą przetwarzać dane tam, gdzie je zbierają, i nie potrzebują zewnętrznych serwerów. Z mojego doświadczenia widzę, że ten balans decyduje o tym, czy rejestrator będzie działał długo i dobrze.

W niniejszym projekcie przyjęto założenie, że urządzenie nie tylko gromadzi surowe dane pomiarowe, ale działa jako niezależny serwer aplikacji i udostępnia pełny graficzny interfejs. To wymusiło rezygnację z prostych mikrokontrolerów 8-bitowych na rzecz wydajniejszych jednostek obliczeniowych, łączących wiele funkcji (w tym bezprzewodową łączność) w jednym układzie scalonym.

Do realizacji zadania wybrano platformę **ESP32**, kierując się jej unikalną architekturą dwurdzeniową. Pozwala ona na separację procesów krytycznych (obsługa przerwania GPS i IMU) od obsługi stosu sieciowego WiFi, co jest ważne, żeby dane były zapisywane ciągle. Dodatkowo, wysoki stopień integracji w jednym układzie pozwolił zminimalizować wymiary urządzenia oraz uprościć układ oraz obniżyć cenę

2 Projekt sprzętowy

Warstwa sprzętowa urządzenia została zaprojektowana w sposób modułowy, co ułatwia diagnostykę oraz ewentualną wymianę komponentów. System składa się z następujących komponentów:

2.1 Elementy układu

- **Mikrokontroler:** ESP32 DevKit V1 WROOM-32 – dwurdzeniowy procesor z WiFi i Bluetooth
- **Moduł GPS:** NEO-6M V2 – odbiornik satelitarny (komunikacja UART)
- **IMU (czujnik ruchu):** MPU6050 – akcelerometr i żyroskop 6-osiowy (I2C)
- **Wyświetlacz:** OLED 0.96" – monochromatyczny ekran 128x64 px (I2C)
- **Pamięć masowa:** Micro SD Card Module – czytnik kart MicroSD (SPI)
- **Zasilanie:** Ładowarka Li-Ion z wejściem USB-C i zabezpieczeniami, współpracująca z ogniwem 18650. Układ umożliwia ładowanie baterii lub zasilanie systemu bezpośrednio z portu USB.

2.2 Konfiguracja interfejsów i połączeń

Wszystkie moduły peryferyjne zostały połączone z mikrokontrolerem zgodnie z załączonym schematem elektrycznym. Kluczowym wymogiem poprawności działania układu jest połączenie wszystkich mas (GND) w jeden wspólny potencjał.

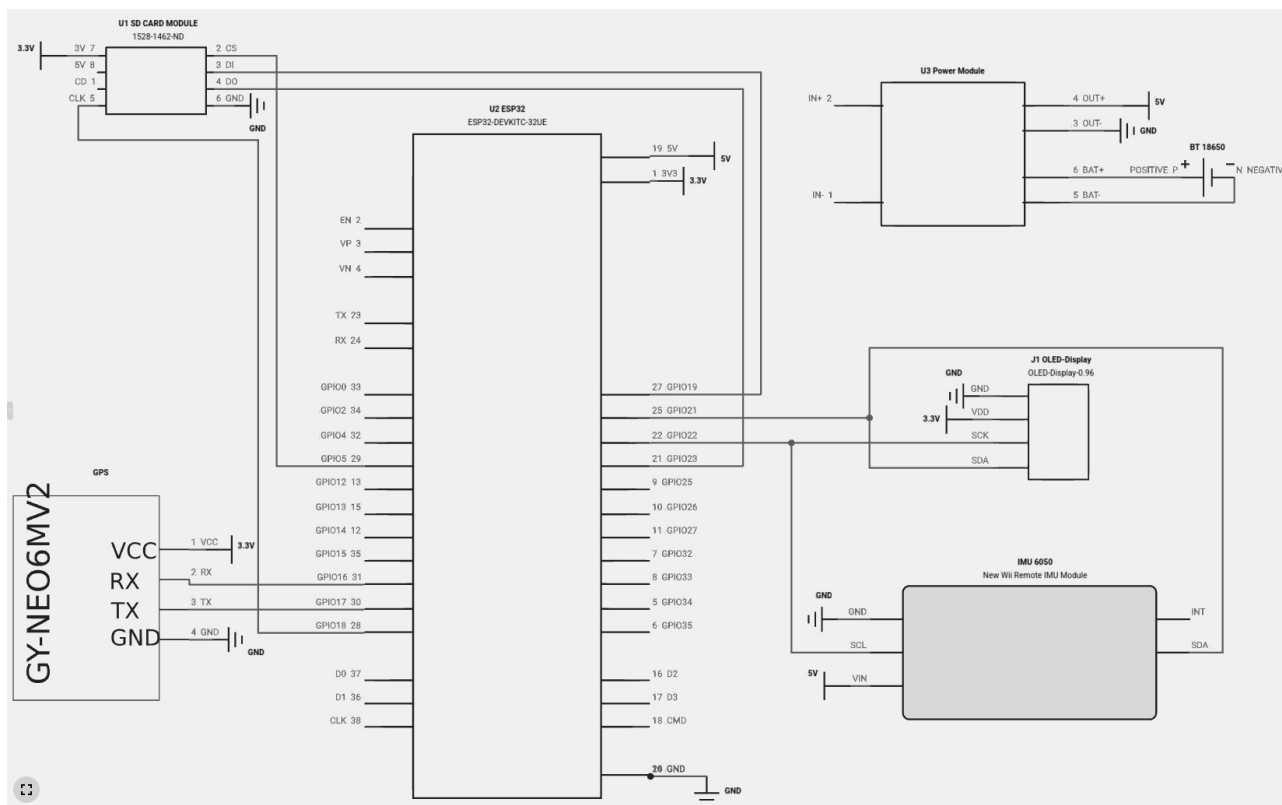


Figure 1: Schemat elektryczny systemu

Magistrala I2C (OLED + MPU6050): Urządzenia podłączone równolegle do wspólnych linii SDA i SCL.

Urządzenie	Pin Modułu	Pin ESP32	Uwagi
OLED 0.96"	SDA	GPIO 21	Adres I2C: 0x3C Z pinu 3V3 ESP32 Wspólna masa
	SCL	GPIO 22	
	VCC	3.3V	
	GND	GND	
MPU6050 (IMU)	SDA	GPIO 21	Adres I2C: 0x68 Z pinu 5V OUT power module Wspólna masa
	SCL	GPIO 22	
	VCC	5V	
	GND	GND	

Table 1: Konfiguracja magistrali I2C

Moduł GPS NEO-6M V2 (UART): Wykorzystuje sprzętowy port szeregowy UART2.

Pin Modułu GPS	Pin ESP32	Funkcja
TX	GPIO 16 (RX2)	Transmisja z GPS do ESP32
RX	GPIO 17 (TX2)	Konfiguracja GPS (opcjonalna)
VCC	3.3V	Z pinu 3V3 ESP32
GND	GND	Wspólna masa

Table 2: Połączenie modułu GPS

Micro SD Card Module (SPI): Wykorzystuje standardową magistralę VSPI.

Pin Modułu SD	Pin ESP32	Funkcja
CS	GPIO 5	Chip Select
SCK	GPIO 18	Zegar synchronizacji
MISO	GPIO 19	Dane (Master In, Slave Out)
MOSI	GPIO 23	Dane (Master Out, Slave In)
VCC	3.3V	Z pinu 3V3 ESP32
GND	GND	Wspólna masa

Table 3: Połączenie czytnika kart SD

2.3 Projekt obudowy

Obudowa została zaprojektowana w Autodesk Inventor z przeznaczeniem do druku 3D. Składa się z podstawy, ramki montażowej i pokrywy z wzorem plastra miodu. Podstawa zawiera komory na poszczególne moduły oraz schowek na ogniwo 18650. Ramka dystansowa tworzy okno dostępu do ekranu i GPS. Pokrywa z ażurowym wzorem zapewnia podgląd wnętrza układu.



Figure 2: Widok złożonej obudowy

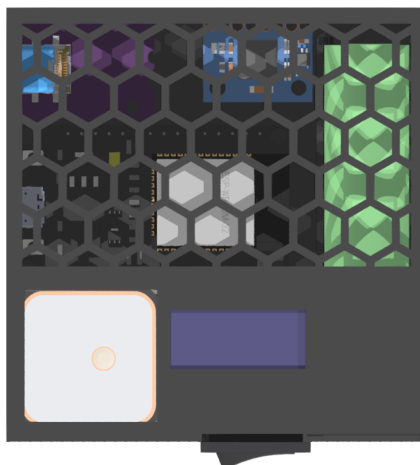


Figure 3: RWidok z góry

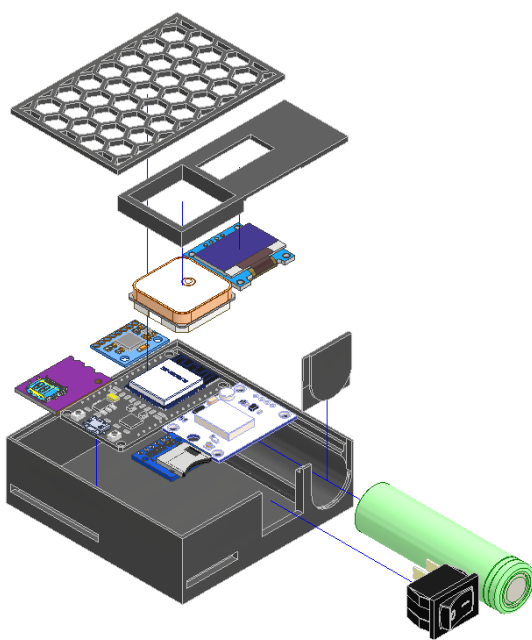


Figure 4: Widok eksplodowany konstrukcji

3 Przegląd technologii i narzędzi

Budowa trackera GPS wymaga połączenia kilku różnych technologii komunikacyjnych. Poniżej opisano kluczowe elementy systemu wykorzystane w projekcie.

3.1 Rozwój IoT i miniaturyzacja

Internet Rzeczy (IoT) to świat połączonych urządzeń – od zegarków, przez czujniki temperatury, aż po inteligentne zamki [3, 6]. Miniaturyzacja, czyli proces zmniejszania rozmiarów komponentów elektronicznych, umożliwiła budowę tak małych i wydajnych urządzeń.

Moduł odbiornika GPS NEO-6M ma wymiary zaledwie $16 \times 12,2 \times 2,4$ mm [1], co pozwala zbudować kompaktowy tracker. Mniejsze układy zużywają też znacznie mniej energii – urządzenie może pracować na baterii przez wiele godzin lub dni. Integracja wielu funkcji w jednym układzie (System on Chip, w skrócie SoC) obniża koszty produkcji i zwiększa niezawodność systemu.

3.2 Telemetria i przetwarzanie danych

Telemetria to technologia zbierania danych z odległych miejsc i przesyłania ich w czasie rzeczywistym. W przypadku trackera GPS oznacza to automatyczne odczytywanie pozycji i prędkości bez potrzeby fizycznego dostępu do urządzenia.

Przetwarzanie na miejscu (Edge Computing) to koncepcja, w której dane nie są od razu wysyłane do internetu, ale wstępnie analizowane bezpośrednio w urządzeniu [2]. Mikrokontroler ESP32 odbiera surowe dane z modułu GPS, przetwarza je – rozpoznaje współrzędne, oblicza prędkość – i zapisuje gotowe wyniki na karcie pamięci.

Dlaczego to ważne? Po pierwsze, oszczędza to energię baterii – zamiast ciągle wysyłać dane przez sieć (co zużywa dużo prądu), urządzenie analizuje je lokalnie i wysyła tylko niezbędne informacje. Po drugie, zwiększa to szybkość reakcji – decyzje podejmowane są natychmiast, bez czekania na odpowiedź z serwera.

3.3 Architektura ESP32

ESP32 to mikrokontroler firmy Espressif Systems, który posiada dwa niezależne rdzenie procesora Xtensa LX6 pracujące z częstotliwością do 240 MHz [5, 6]. W praktyce oznacza to, że może wykonywać dwa zadania jednocześnie, bez wzajemnego spowalniania się.

Jak to działa? Jeden rdzeń może zajmować się obliczeniami – odbieraniem danych GPS, zapisywaniem ich na kartę pamięci, wyświetlaniem informacji na ekranie [4]. W tym samym czasie drugi rdzeń obsługuje komunikację bezprzewodową – Wi-Fi lub Bluetooth. Dzięki temu oba procesy działają płynnie i nie przeszkadzają sobie nawzajem.

ESP32 posiada także wbudowane moduły Wi-Fi i Bluetooth, co czyni go idealnym rozwiązaniem dla projektów wymagających bezprzewodowej komunikacji [3]. Dodatkowo, układ obsługuje zaawansowane tryby oszczędzania energii – może automatycznie "zasypiać" gdy nie wykonuje zadań, pobierając mniej niż 10 mikroamperów prądu.

3.4 System GNSS i protokół NMEA 0183

Odbiornik GPS komunikuje się z mikrokontrolerem za pomocą protokołu NMEA 0183 – standardu opracowanego przez National Marine Electronics Association, używanego w nawigacji morskiej i lądowej [7, 9]. Jest to protokół tekstowy, w którym dane przesyłane są w postaci czytelnych dla człowieka zdań ASCII.

Moduł u-blox NEO-6M wysyła dane przez port szeregowy z prędkością 9600 bitów na sekundę [1]. Dane te mają postać tekstowych linii rozpoczynających się od znaku dolara (\$), zawierających kod identyfikacyjny oraz pola oddzielone przecinkami.

Najważniejsze typy ramek wykorzystywane w projekcie to:

- **\$GPGGA** – zawiera podstawowe dane o pozycji: szerokość geograficzną, długość geograficzną, wysokość nad poziomem morza oraz liczbę widocznych satelitów. To te informacje, które odpowiadają na pytanie "gdzie jestem?" [8].
- **\$GPRMC** – dostarcza dane o ruchu: prędkość przemieszczania się oraz kierunek jazdy (kurs). Dzięki tym danym można wyświetlić strzałkę wskazującą kierunek poruszania się.

ESP32 posiada sprzętowe porty UART, które automatycznie odbierają dane w tle [5]. Oznacza to, że procesor nie musi ciągle sprawdzać, czy nadeszły nowe dane – specjalny układ UART robi to automatycznie i informuje procesor, gdy cała ramka została odebrana. To znacznie odciąża główny procesor i zapewnia stabilność działania.

3.5 Magistrale komunikacyjne w systemach wbudowanych

W projekcie wykorzystano trzy różne standardy komunikacji, każdy wybrany do konkretnego zadania:

I2C (Inter-Integrated Circuit) to magistrala, która wykorzystuje tylko dwa przewody: jeden do przesyłania danych (SDA), drugi do synchronizacji czasowej (SCL) [11]. Największą zaletą I2C jest możliwość podłączenia wielu urządzeń do tych samych dwóch przewodów – każde urządzenie ma swój unikalny adres i reaguje tylko na komunikaty skierowane do niego.

W trackerze na jednej magistrali I2C pracują jednocześnie wyświetlacz OLED oraz czujnik ruchu IMU. Mikrokontroler rozróżnia je po adresach. I2C jest idealne dla urządzeń znajdujących się blisko siebie, które nie wymagają bardzo szybkiego przesyłu danych – typowa prędkość to do 400 tysięcy bitów na sekundę, co w zupełności wystarcza dla wyświetlaczy i czujników [12].

SPI (Serial Peripheral Interface) to magistrala synchroniczna zaprojektowana do szybkiego przesyłu danych. Wykorzystuje cztery linie komunikacyjne i może przysyłać dane jednocześnie w obu kierunkach (tzw. komunikacja dwukierunkowa) [11].

SPI zostało wybrane do obsługi karty pamięci SD, ponieważ zapis danych wymaga szybkiego transferu – karta SD może przyjmować dane z prędkością kilku do kilkudziesięciu megabitów na sekundę [12]. To znacznie szybciej niż I2C, dlatego dla karty pamięci jest to najlepszy wybór.

UART (Universal Asynchronous Receiver-Transmitter) to port szeregowy wykorzystujący dwa przewody: jeden do wysyłania danych (TX), drugi do odbierania (RX) [10]. Komunikacja odbywa się bez wspólnej linii zegarowej – oba urządzenia muszą być po prostu skonfigurowane na tę samą prędkość transmisji.

W projekcie UART obsługuje moduł GPS. ESP32 posiada sprzętowe porty UART, co oznacza, że odbiór danych odbywa się automatycznie przez dedykowany układ elektroniczny [5]. Procesor nie musi ciągle sprawdzać, czy nadeszły nowe dane – układ sam go powiadomi. To zapewnia stabilność odbioru danych nawet przy intensywnej pracy procesora.

UART jest prosty, niezawodny i działa nawet na większych odległościach (przy odpowiednio niższych prędkościach). Jest idealny dla urządzeń, które wymagają prostego, dwukierunkowego połączenia między dwoma urządzeniami.

4 Implementacja oprogramowania

4.1 Opis funkcjonalności i obsługa aplikacji

Aplikacja webowa stanowi główne centrum sterowania urządzeniem. Poniżej przedstawiono szczegółowy opis interfejsu oraz dostępnych funkcji, które zostały zaimplementowane w celu zapewnienia intuicyjnej obsługi.

4.1.1 Ekran główny i parametry na żywo

Po nawiązaniu połączenia z siecią Wi-Fi urządzenia, w prawym górnym rogu aplikacji widoczny jest status "POŁĄCZONO". W przypadku utraty zasięgu lub zerwania połączenia, komunikat natychmiast zmienia się na "ROZŁĄCZONO", informując użytkownika o braku aktualnej synchronizacji danych.

Ekran główny został podzielony na dwie sekcje logiczne: mapę z aktualną pozycją (wykorzystującą bibliotekę Leaflet) oraz panel parametrów telemetrycznych.

Na panelu dolnym wyświetlane są kluczowe parametry odświeżane w czasie rzeczywistym:

- **Prędkość:** Aktualna prędkość przemieszczania się wyrażona w km/h.
- **Wysokość:** Wysokość nad poziomem morza w metrach.
- **Dystans:** Całkowita długość trasy przebyta w bieżącej sesji.
- **Satelitey:** Liczba widocznych i używanych do nawigacji satelitów, co pozwala ocenić jakość "fixa".
- **HDOP:** Parametr precyzji geometrycznej (im niższy, tym lepiej).
- **Bateria:** Napięcie ogniwa zasilającego, pozwalające monitorować stan naładowania.

W sytuacji, gdy urządzenie nie posiada jeszcze ustalonej pozycji GPS (np. zaraz po włączeniu wewnątrz budynku), na ekranie widnieje wyraźny komunikat o szukaniu sygnału GPS.

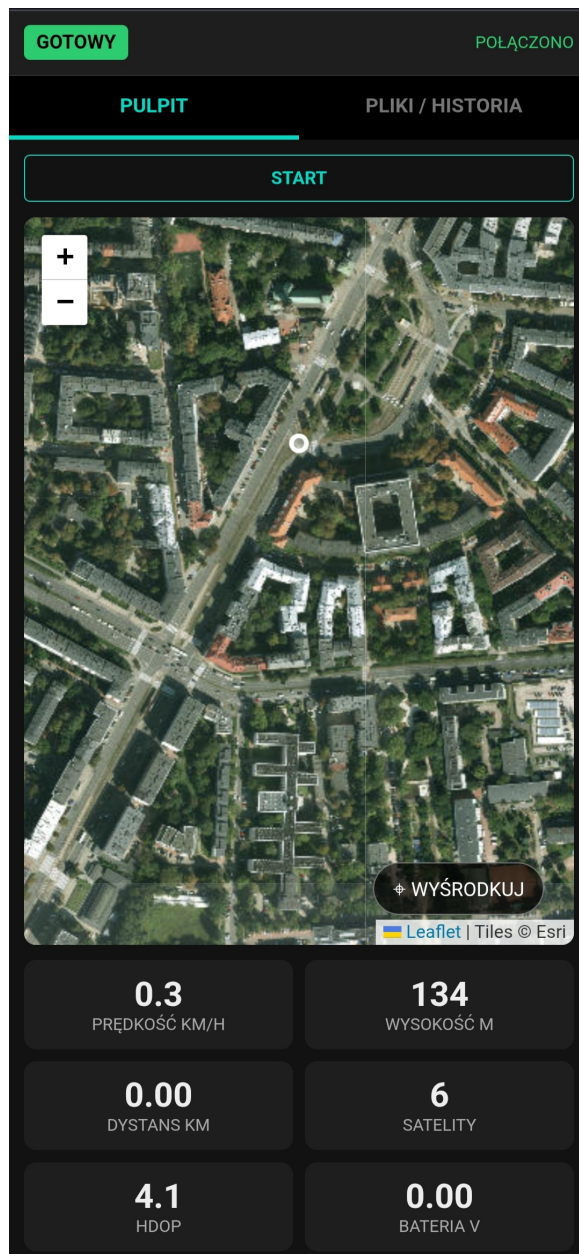


Figure 5: Ekran startowy aplikacji - system gotowy do pracy

Proces nagrywania trasy

Sterowanie procesem rejestracji odbywa się za pomocą centralnego przycisku "START". Rozpoczęcie nagrywania aktywuje rysowanie śladu na mapie oraz zliczanie statystyk sesji. Aplikacja udostępnia również funkcję centrowania mapy – specjalny przycisk pozwala jednym kliknięciem przywrócić widok na aktualną pozycję GPS, co jest przydatne po ręcznym przesuwaniu mapy w celu podejrzenia okolicy.

System obsługuje zaawansowane funkcje pauszowania:

- **Pauza manualna:** Użytkownik może w każdej chwili ręcznie wstrzymać zapis przyciskiem "PAUZA".
- **Auto-Pauza:** System automatycznie wykrywa brak ruchu i wstrzymuje zapis, co jest sygnalizowane pomarańczowym komunikatem "PAUZA / AUTO-PAUZA" na mapie. Zapobiega to rejestrowaniu błędnych punktów podczas postojów.

Po zakończeniu aktywności użytkownik ma do wyboru trzy opcje:

1. **WZNÓW:** Powrót do nagrywania w ramach tej samej sesji.
2. **ZAPISZ:** Trwałe zapisanie zgromadzonych danych do pliku w pamięci urządzenia.
3. **ODRZUĆ:** Anulowanie sesji i usunięcie tymczasowych danych bez zapisu.

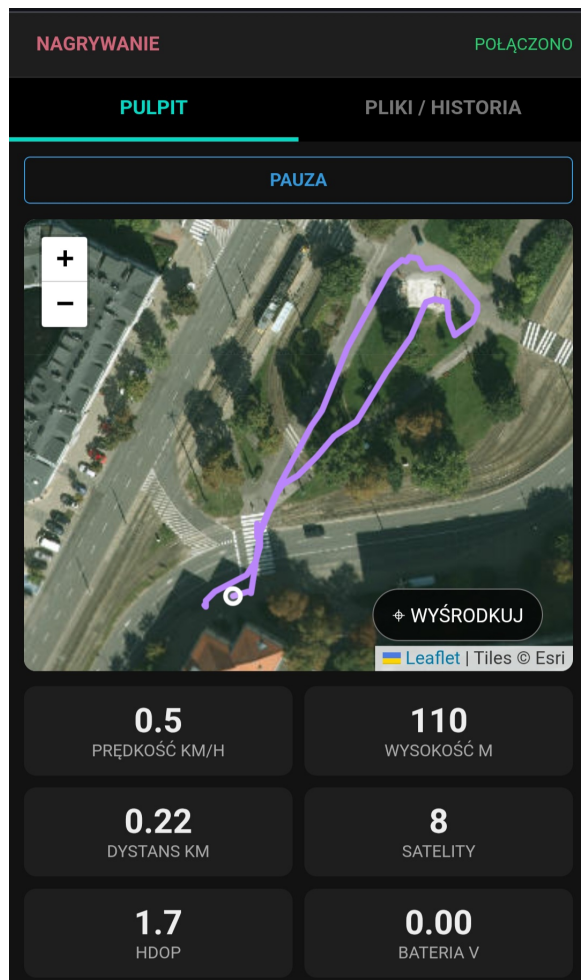


Figure 6: Trwająca sesja z rysowaną trasą

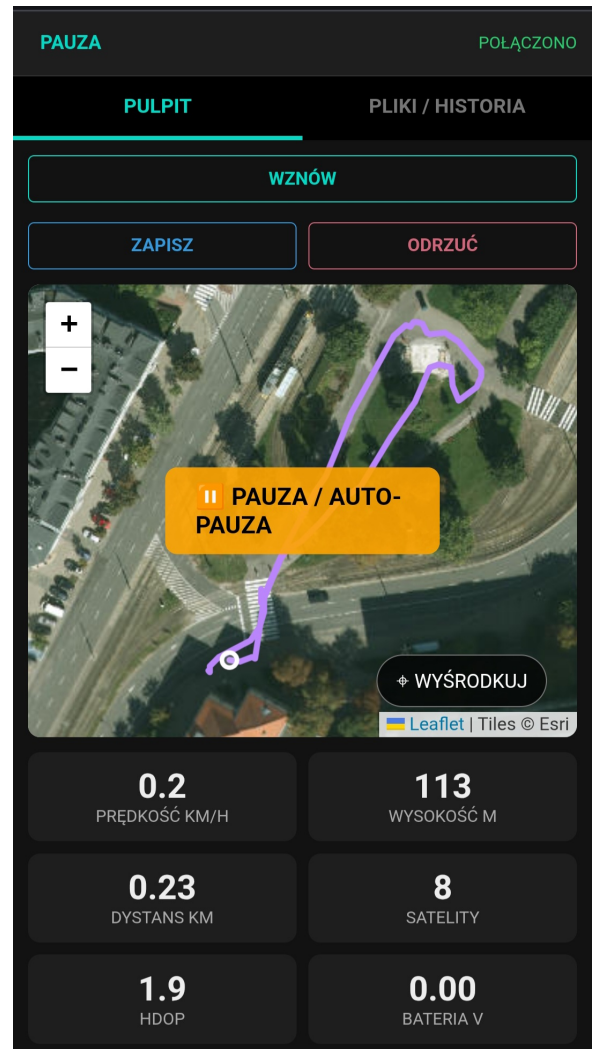


Figure 7: Stan wstrzymania nagrywania (Pauza)

Wykresy i analiza trendów

Dla bardziej wnikliwej analizy aktywności, aplikacja udostępnia sekcję wykresów rysowanych na żywo. Pozwala to na bieżąco śledzić historię parametrów:

- Profil prędkości w czasie.
- Profil wysokości n.p.m.
- Wykres precyzji GPS (HDOP).
- Odczyty z akcelerometru (siły przeciążeń).

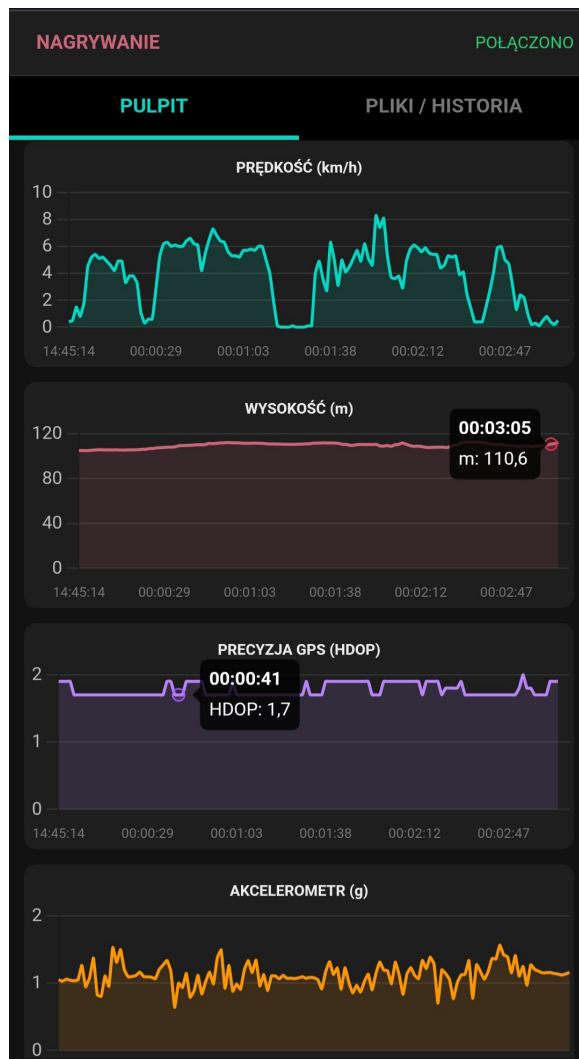


Figure 8: Wykresy parametrów telemetrycznych

4.1.2 Zarządzanie plikami i historia tras

Druga zakładka interfejsu, "PLIKI / HISTORIA", dedykowana jest zarządzaniu danymi zgromadzonymi na karcie SD. Użytkownik otrzymuje pełną kontrolę nad archiwum tras bez konieczności wyjmowania karty z urządzenia.

Dostępne funkcjonalności obejmują:

- **Odświeżanie listy:** Pobranie aktualnego wykazu plików z karty SD.
- **Pobieranie:** Transfer wybranego pliku .csv/.txt bezpośrednio do pamięci telefonu/komputera.
- **Usuwanie:** Trwałe usunięcie niepotrzebnych plików.
- **Podgląd:** Specjalny tryb wizualizacji archiwalnej sesji.

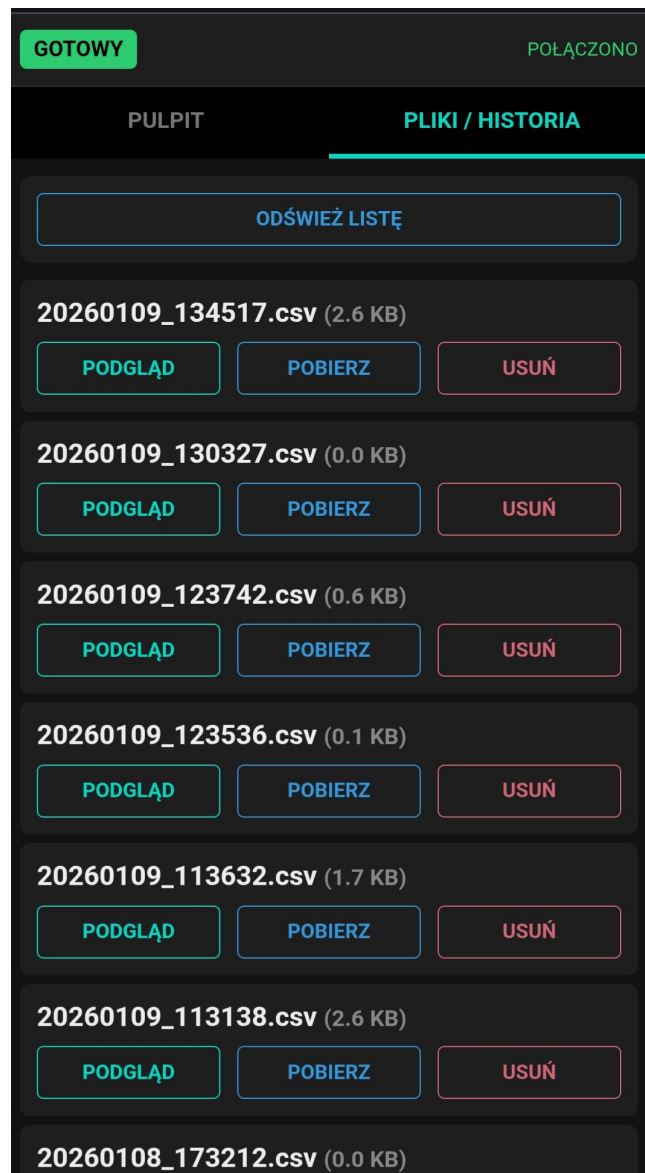


Figure 9: Lista zapisanych plików

W trybie **podglądu pliku** interfejs prezentuje kompletny ślad trasy na mapie oraz rozszerzony zestaw statystyk, niedostępnych w trybie na żywo, takich jak:

- **Średnia prędkość:** Wartość wyliczona dla całej trasy, z uwzględnieniem postojów (auto-pauza).
- **Suma wzniosów i spadków:** Łączna różnica wysokości pokonana podczas sesji.
- **Maksymalna wysokość:** Najwyższy punkt osiągnięty na trasie (max AMSL).
- **Dystans:** Całkowita długość trasy.
- **Średnie HDOP:** Średnia wartość precyzji pomiaru z całego zapisu, pozwalająca ocenić wiarygodność danych.

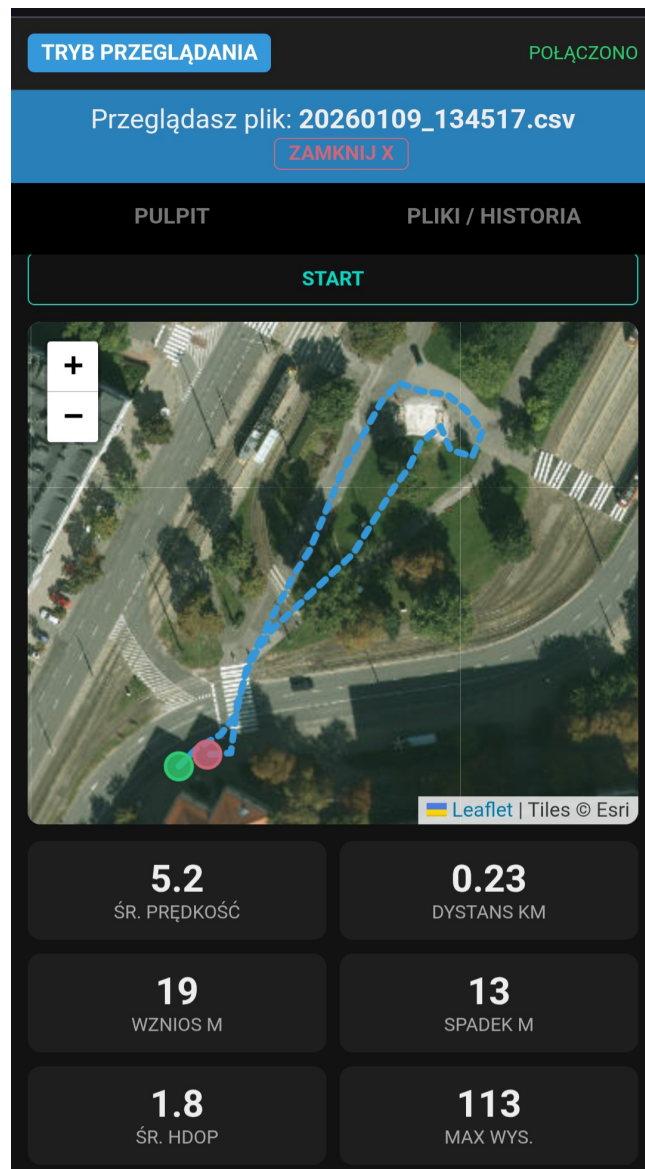


Figure 10: Tryb podglądu historii z dodatkowymi parametrami

Dodatkowo, w trybie podglądu użytkownik ma dostęp do szczegółowych wykresów analizujących przebieg całej trasy. Są to te same wykresy co w trybie na żywo (prędkość, wysokość, HDOP), ale przedstawiające kompletne dane z całego nagrania, co pozwala na łatwą identyfikację kluczowych momentów wycieczki.



Figure 11: Szczegółowa analiza wykresów w trybie przeglądania historii

Dane zapisywane są na karcie SD w formacie CSV (Comma Separated Values), co umożliwia ich łatwy import do arkuszy kalkulacyjnych (np. Excel) lub specjalistycznego oprogramowania GIS. Każda linia w pliku odpowiada jednemu punktowi pomiarowemu i zawiera następujące pola, oddzielone przecinkami:

- **Czas (ms):** Czas systemowy w milisekundach od uruchomienia urządzenia.
- **Szerokość i Długość geograficzna:** Współrzędne GPS (format dziesiętny).
- **Prędkość (km/h):** Chwilowa prędkość ruchu.
- **Wysokość (m n.p.m.):** Wysokość nad poziomem morza.
- **HDOP:** Precyzja pomiaru poziomego (im mniejsza wartość, tym lepiej).
- **Satelitey:** Liczba widocznych satelitów użytych do ustalenia pozycji.
- **Akcelerometr (ax, ay, az):** Surowe dane z czujnika przeciążeń dla trzech osi.
- **Bateria (V):** Napięcie zasilania w momencie pomiaru.

Przykładowa struktura linii danych:


```
millis,lat,lon,speed,alt,hdop,sats,ax,ay,az,batt  
34500,52.2297,21.0122,15.5,112.4,1.2,8,0.02,-0.98,0.15,4.12
```

Integracja z wyświetlaczem OLED

Niezależnie od aplikacji mobilnej, urządzenie wyposażone jest w fizyczny ekran OLED. Pełni on funkcję pomocniczą, prezentując najważniejsze informacje w zwięzłej formie:

- Aktualny status: Oczekiwanie / Nagrywanie / Pauza.
- Liczba widocznych satelitów (ikona "SAT").
- Bieżąca prędkość oraz przebyty dystans.
- Adres IP oraz nazwa sieci WiFi (SSID) w celu ułatwienia połączenia.



Figure 12: Przykładowy wygląd ekranu OLED prezentujący status urządzenia

4.2 Szczegóły techniczne implementacji

Architektura wielowątkowa

Oprogramowanie zostało zrealizowane w środowisku PlatformIO (C++) na układzie ESP32, który posiada dwa rdzenie. Pozwoliło to na podział zadań:

- **Rdzeń 0 (Komunikacyjny):** Obsługuje WiFi i serwer strony internetowej. Dzięki temu, gdy użytkownik przegląda mapę na telefonie, nie obciąża to głównego procesu zapisu.
- **Rdzeń 1 (Logiczny):** Zajmuje się tylko tym, co najważniejsze - odbieraniem danych z GPS i zapisywaniem ich na kartę.

Bezpieczeństwo i wielozadaniowość (Mutex)

Aby dwa rdzenie nie próbowały jednocześnie pisać na tę samą kartę SD (co mogłoby ją uszkodzić), zastosowano mechanizm Mutex. Działa on jak cyfrowy "klucz" - tylko jeden proces może mieć do niego dostęp w danym momencie. Jeśli WiFi chce odczytać plik, musi poczekać, aż GPS skończy zapamiętywać pozycję.

Logika sterowania (Maszyna Stanów)

System działa w trzech, ściśle określonych trybach, co zapobiega błędom:

- **IDLE (Czuwanie):** Urządzenie działa i pokazuje dane, ale nie zapisuje plików.

- **RECORDING (Nagrywanie):** Główny tryb wycieczkowy - dane są zapisywane na kartę.
- **PAUSED (Pauza):** Wstrzymanie zapisu. Ważne jest to, że przejście w tryb pauzy wymusza "zrzut" (Flush) wszystkich danych na kartę, więc nagłe wyłączenie urządzenia nie spowoduje utraty ostatniego fragmentu trasy.

Interfejs WWW (SPA)

Strona internetowa ładuje się z pamięci urządzenia tylko raz. Potem telefon pobiera w tle już tylko same liczby (prędkość, pozycję). Dzięki temu interfejs działa płynnie nawet na słabszym zasięgu, a bateria trackera jest oszczędzana.

Bezpieczeństwo Zapisu

Dane zbierane są w pamięci RAM, a na kartę SD zrzucane są w większych paczkach co 10 sekund. Chroni to kartę przed zużyciem, a nazwy plików tworzone są automatycznie z daty i godziny.

Stabilność i Watchdog

System ignoruje "pływanie" GPS na postoju (Filtr Histerezy). Dodatkowo, funkcja "Watchdog" restertuje moduł WiFi w razie problemów z połączeniem, nie przerywając przy tym nagrywania trasy.

5 Testy i analiza wyników

W celu weryfikacji założeń projektowych przeprowadzono serię testów sprawdzających niezawodność urządzenia w warunkach rzeczywistych. Skupiono się na trzech krytycznych aspektach: czasie akwizycji sygnału GPS, odporności na błędy zapisu oraz wydajności energetycznej.

5.1 Testy modułu GPS (Cold Start vs Hot Start)

Jednym z kluczowych parametrów użytkowych lokalizatora jest czas oczekiwania na ustalenie pierwszej pozycji (Time To First Fix - TTFF). Procedura testowa obejmowała uruchomienie urządzenia w otwartej przestrzeni po długim okresie nieaktywności (powyżej 4 godzin).

- **Cold Start (Zimny start):** Moduł uruchamia się "na ślepo" - nie wie, gdzie się znajduje, ani gdzie na niebie szukać satelitów. Musi dopiero pobrać te dane z sygnału GPS. aobserwowane czasy oczekiwania wynosiły od 1 do 5 minut.
- **Hot Start:** Po krótkim zaniku zasilania (poniżej 30 minut) moduł odzyskuje fix niemal natychmiastowo (zazwyczaj poniżej 5 sekund), co potwierdza poprawność działania układu zasilania.

Jak wskazują badania porównawcze [13], moduł NEO-6M, mimo bycia rozwiązaniem budżetowym, oferuje dokładność porównywalną do droższych sensorów (np. Walksnail WS-M181). Głównymi wadami wymienianymi w literaturze, a potwierdzonymi w niniejszym projekcie, są długi czas zimnego startu (wyszukiwania satelitów) oraz niższa częstotliwość odświeżania. Jednakże, jak zauważają autorzy badania, wady te nie są krytyczne w typowych zastosowaniach amatorskich i przy pomiarach w punktach statycznych.

5.2 Analiza jakości śladu w ruchu

Testy wykazały, że o ile w warunkach statycznych moduł radzi sobie bardzo dobrze, to podczas aktywności dynamicznych (np. bieg) lub przy okresowym przesłanianiu anteny, jakość sygnału zauważalnie spada.

W momentach, gdy antena jest zasłaniana (np. przez ciało biegacza lub gęste korony drzew), na mapie mogą pojawiać się nieprawdziwe trasy (artefakty), odbiegające od rzeczywistego toru poruszania się. Towarzyszą temu również błędne odczyty prędkości chwilowej. Jest to zjawisko wynikające z chwilowej utraty widoczności satelitów i jest typowe dla odbiorników GPS tej klasy bez wspomagania systemami inercyjnymi.

5.3 Bilans energetyczny i czas pracy

Urządzenie zasilane jest ogniwem Li-Ion 18650 o pojemności nominalnej 2600 mAh. Szacunkowy pobór prądu przez poszczególne moduły:

- ESP32 (z WiFi w trybie AP+STA): ok. 150-200 mA [14]
- Moduł GPS: ok. 40 mA [15]
- OLED i czujniki: ok. 10-15 mA [16]

Łączny pobór prądu oscyluje w granicach 250 mA. Uwzględniając sprawność przetwornicy Step-Up (ok. 85%), realny czas pracy wynosi:

$$T \approx \frac{2600mAh}{250mA} \cdot 0.85 \approx 8.8h$$

Jest to wynik satysfakcjonujący dla jednodniowej wycieczki rowerowej lub pieszego monitoringu.

6 Podsumowanie i wnioski

Zrealizowany projekt inżynierski spełnił postawione przed nim założenia. Udało się zbudować autonomiczny, miniaturowy system rejestracji trasy, który z powodzeniem zastępuje komercyjne rozwiązania w zastosowaniach amatorskich i półprofesjonalnych.

6.1 Osiągnięcia

Względem pierwotnych prototypów, obecna wersja urządzenia charakteryzuje się:

- **Pełną wielozadaniowością:** Rozdzielenie logiki zapisu danych od obsługi interfejsu użytkownika wyeliminowało problemy z "przycinaniem się" strony WWW.
- **Jakością danych:** Zastosowanie filtra *Auto-Pause* znacząco poprawiło czytelność generowanych śladów GPX/TXT, eliminując szum pomiarowy na postojach.
- **Ergonomią:** Interfejs WWW dostępny na smartfonie eliminuje konieczność montowania fizycznych przycisków i dużych wyświetlaczy LCD na obudowie.

6.2 Ograniczenia i plany rozwojowe

Obecna rewizja urządzenia posiada pewne ograniczenia, które wyznaczają kierunki dalszych prac rozwojowych:

1. **Brak łączności globalnej:** Urządzenie wymaga połączenia WiFi z telefonem użytkownika. Dodanie modułu GSM (np. SIM800L) pozwoliłoby na wysyłanie pozycji do chmury niezależnie od obecności operatora.

Podsumowując, zaprojektowany system stanowi solidną bazę do dalszego rozwoju systemów IoT przeznaczonych do monitoringu floty i aktywności fizycznej.

References

- [1] u-blox AG. *NEO-6 u-blox 6 GPS Modules – Data Sheet* (GPS.G6-HW-09005). 2011.
- [2] Design and Implementation of ESP32-Based Edge Computing for Object Detection. *Sensors*, vol. 25, no. 6, p. 1656, 2025. DOI: 10.3390/s25061656.
- [3] Botland. *ESP32 - Co to jest i do czego służy?* 2024. [Online]. Dostępne: <https://botland.com.pl/blog/esp32-co-to-jest-i-do-czego-sluzyl>
- [4] Performance Evaluation of Programming Languages on ESP32 Microcontroller. *Electronics*, vol. 12, no. 1, p. 143, 2022.
- [5] Espressif Systems. *ESP32 Technical Reference Manual*. 2024. [Online]. Dostępne: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- [6] MSalamon.pl. *ESP32 od Podstaw – Przegląd Rodzin Układów i Ich Najważniejszych Cech*. 2025. [Online]. Dostępne: <https://sklep.msalamon.pl/blog/esp32-od-podstaw-przegląd-rodzin-układów-i-ich-najważniejszych-cech/>
- [7] Wikipedia. *NMEA 0183 – Wikipedia, wolna encyklopedia*. 2005. [Online]. Dostępne: https://pl.wikipedia.org/wiki/NMEA_0183
- [8] Elektro-Praktyk. *System nawigacji satelitarnej GPS, część 8 – Protokół NMEA-0183*. 2013.
- [9] Geoforum.pl. *Uniwersalny język nawigacji – Standard NMEA-0183 w odbiornikach GPS*. (b.d.).
- [10] Elektro-Praktyk. *UART na SPI/I2C – Zestawienie cech i parametrów mostków komunikacyjnych*. 2013.
- [11] WonderfulPCB. *Wybór właściwego protokołu komunikacyjnego: I2C, SPI lub UART*. 2025. [Online]. Dostępne: <https://www.wonderfulpcb.com/pl/blog/i2c-communication-protocol-vs-spi-vs-uart-comparison-guide/>
- [12] OurPCB. *I2C vs SPI: Różnice, które powinieneś znać*. (b.d.). [Online]. Dostępne: <https://ourpcb.pl/i2c-vs-spi.html>
- [13] Comparative Experimental Evaluation of the Error of GPS Modules. *Automatic Control and Power Systems*, vol. 10, no. 2, 2025. [Online]. Dostępne: <https://science.lpnu.ua/acps/all-volumes-and-issues/volume-10-number-2-2025/comparative-experimental-evaluation-error-gps>
- [14] DeepBlueEmbedded. *ESP32 Sleep Modes & Power Consumption*. (b.d.). [Online]. Dostępne: <https://deepbluembedded.com/esp32-sleep-modes-power-consumption-in-each-mode/>
- [15] Robotics Bangladesh. *GPS Module NEO-6M Satellite Positioning Module*. (b.d.). [Online]. Dostępne: <https://roboticsbd.com/electronics-module/2397-gps-module-neo-6m-satellite-positioning-module-for-arduino-with-microusb-robotics.html>
- [16] BitBank Software. *How much current do OLED displays use?*. 2019. [Online]. Dostępne: <https://bitbanksoftware.blogspot.com/2019/06/how-much-current-do-oled-displays-use.html>