



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

Projekt inżynierski

System lokalizacji pojazdu

Autor: Dawid Jaraczewski

Kierujący pracą: dr inż. Krzysztof Tokarz

Gliwice, grudzień 2020

Oświadczenie

Wyrażam zgodę / Nie wyrażam zgody* na udostępnienie mojej pracy dyplomowej / rozprawy doktorskiej*.

Gliwice, dnia 30 grudnia 2020

.....
(podpis)

.....
(poświadczenie wiarygodności
podpisu przez Dziekanat)

* podkreślić właściwe

Spis treści

1	Wstęp	1
1.1	Wprowadzenie w zagadnienie	2
1.2	Cel pracy	3
1.3	Zakres pracy	3
1.4	Charakterystyka rozdziałów	4
2	Analiza tematu	5
2.1	Założenia realizacyjne	5
2.2	Schemat blokowy urządzenia	6
3	Wymagania i narzędzia	7
3.1	Opis narzędzi	7
3.1.1	Microsoft Visual Studio Code	7
3.1.2	PlatformIO IDE	8
3.1.3	Autodesk Fusion 360	8
3.1.4	ThingSpeak	8
3.1.5	Leaflet	9
3.1.6	GitHub	9
3.1.7	GitKraken	9
3.2	Wymagania funkcjonalne i нефункционалне	10
3.2.1	Wymagania funkcjonalne	10
3.2.2	Wymagania нефункционалне	11
3.3	Przypadki użycia	12
3.3.1	Przypadki użycia dla urządzenia	12
3.3.2	Przypadki użycia dla witryny internetowej	13

3.4	Proces pracy nad projektowaniem i implementacją	14
4	Specyfikacja zewnętrzna	15
4.1	Opis urządzenia	15
4.2	Kategorie użytkowników	16
4.3	Scenariusze korzystania z systemu	17
4.3.1	Konfiguracja urządzenia	17
4.3.2	Obsługa za pomocą wiadomości tekstowych	17
4.3.3	Odczyt danych za pomocą wiadomości tekstowych	17
4.3.4	Odczyt danych za pomocą witryny internetowej	18
4.4	Instrukcja obsługi	19
4.4.1	Pierwsze uruchomienie i konfiguracja	19
4.4.2	Tryb normalnej pracy	23
4.4.3	Zarządzanie urządzeniem	26
4.4.4	Sposób instalacji	27
4.5	Kwestie bezpieczeństwa	27
5	Specyfikacja wewnętrzna	29
5.1	Architektura urządzenia	29
5.2	Wybrane komponenty	30
5.2.1	AiThinker ESP12S A9G GPRS Node v1.0	30
5.2.2	Źródło zasilania	32
5.3	Schemat blokowy urządzenia	33
5.4	Obudowa urządzenia	34
5.5	Rozmieszczenie elementów w obudowie	34
5.6	Biblioteki	36
5.6.1	SoftwareSerial	36
5.6.2	ESPAsyncWebServer	36
5.6.3	Configurator	36
5.6.4	ConfigurationServer	37
5.7	Klasy	37
5.8	Tworzenie oprogramowania urządzenia	40
5.9	Algorytm pracy urządzenia	41

5.10	Konfiguracja	43
5.11	Witryna internetowa	44
5.12	Wykorzystywane komendy AT	45
5.13	Integracja urządzenia z witryną internetową	46
6	Weryfikacja i walidacja	47
6.1	Testowanie konfiguracji	47
6.2	Testowanie zarządzania	48
6.3	Testy niezawodności	48
6.4	Testowanie dokładności	49
6.5	Testy pracy na zasilaniu awaryjnym	49
6.6	Testy odporności na upadki	50
6.7	Wykryte i usunięte błędy	50
7	Podsumowanie i wnioski	53
7.1	Podsumowanie	53
7.2	Wnioski	53
7.3	Możliwe kierunki rozwoju	54

Rozdział 1

Wstęp

W ciągu ostatniej dekady inteligentne urządzenia, pomagające w codziennym życiu i pracy, znacznie zyskały na popularności i dostępności. Stały się nieodzownym elementem życia wielu ludzi i funkcjonowania przedsiębiorstw. Rosnące znaczenie komunikacji między tymi urządzeniami spowodowało ogromną popularność i rozwój koncepcji IoT (ang. *Internet of Things*), które z kolei mogą pobudzać wzrost rozwoju np. sztucznej inteligencji [10]. Internet Rzeczy to nie tylko urządzenia pomagające monitorować funkcje życiowe podczas uprawiania sportów, informujące nas o temperaturze i wilgotności, czy umożliwiające zarządzanie oświetleniem w mieszkaniu. To także przyrządy, które pozwalają na detekcję utraty zasilania elektrycznego i wysłanie stosownego powiadomienia, wykrycie upadku człowieka i poinformowanie o tym bliskich, czy umożliwiające ciągłe monitorowanie lokalizacji pojazdów. Wszystko to dostępne z ekranu komputera lub smartfona. To jedynie kilka przykładów ich zastosowań. Te inteligentne urządzenia pozwalają na zarządzanie coraz większą ilością sprzętu domowego, automatyzację procesu sortowania w magazynach, czy sterowanie przełącznikami bazując na odczytach przeróżnych czujników. Urządzenia IoT pozwalają także zwiększyć bezpieczeństwo w wielu gałęziach gospodarki, między innymi w medycynie, przy organizacji ruchu lub w sektorze budowlano-montażowym [11]. Jest to niewątpliwie obiecujący kierunek rozwoju technologii, a liczba dostępnych urządzeń i rozwiązań IoT rośnie z każdym dniem. W wielu branżach urządzenia Internetu Rzeczy znalazły zastosowanie i zostały wprowadzone na stałe. Najprawdopodobniej kwestią czasu

jest wdrożenie ich w kolejnych sektorach gospodarki. Niniejszą pracę poświęcono na realizację budowy urządzenia IoT służącego do monitorowania lokalizacji pojazdu (w dalszej części pracy nazywanego lokalizatorem) oraz systemu prezentacji danych zgromadzonych przez urządzenie.

1.1 Wprowadzenie w zagadnienie

Szacowana liczba zarejestrowanych w Polsce pojazdów wynosi około 25 milionów i stale się zwiększa [7]. Parkując pojazd, najczęściej na niestrzeżonych publicznych parkingach, szczególnie wzdłuż zaciemnionych ulic lub na dużych osiedlach, właściciel jest narażony na utratę swojego mienia [13]. Zjawisko kradzieży pojazdów jest w Polsce nadal często występującym problemem. Według raportów policyjnych na temat kradzieży pojazdów z 2019 roku wszczęto aż 8706 postępowań, a stopień wykrycia sprawcy to jedynie 20.5% [8]. Względem poprzednich dwudziestu lat trend jest raczej spadkowy, aczkolwiek liczba utraconych pojazdów przez ostatnie kilka lat utrzymuje się na stałym poziomie [3]. Na ten stan ma wpływ stosowanie coraz bardziej zaawansowanych zabezpieczeń pojazdów. Dużą rolę odgrywa także instalowanie w pojazdach systemów monitoringu, które co prawda nie stanowią zabezpieczenia przed kradzieżą, lecz pomagają w szybkim zorientowaniu się o kradzieży, a także w znalezieniu skradzionego pojazdu [9].

Systemy lokalizacji są obecne wszędzie wokoło. W obecnych czasach każdy posiadacz smartfona może bez problemu w krótkim czasie uzyskać informację o swoim położeniu geograficznym. Umożliwia to wykorzystanie przybliżonej pozycji, ustalonej za pomocą sieci komórkowej i systemów nawigacji satelitarnej – GPS (ang. *Global Positioning System*). Globalny System Pozycjonowania jest najczęściej używaną technologią do określania położenia geograficznego obiektów. System powstał już w latach siedemdziesiątych dla celów militarnych amerykańskiego wojska. Obecnie dostępny jest dla każdego, lecz z mniejszą dokładnością, a do jego użycia wystarczy jedynie odbiornik GPS [12].

Osoby prywatne czy przedsiębiorcy mają realną potrzebę monitorowania swoich dóbr lub floty. Tę możliwość oferuje wykorzystanie nowoczesnych lokalizatorów i dedykowanych systemów lokalizacyjnych.

1.2 Cel pracy

Celem projektu jest wykonanie pełnego systemu lokalizacji pojazdu, składającego się z konfigurowalnego kompaktowego urządzenia udostępniającego położenie pojazdu cyklicznie lub na żądanie, oraz witryny internetowej pozwalającej na prezentację danych o położeniu geograficznym pojazdu w czytelny dla użytkownika sposób.

1.3 Zakres pracy

Zakres pracy obejmuje:

- Przegląd istniejących rozwiązań;
- Wybór komponentów do budowy urządzenia;
- Wybór sposobu wysyłania danych;
- Wybór sposobu przechowywania i prezentacji danych;
- Zapoznanie się z komendami AT;
- Stworzenie oprogramowania;
- Wybór sposobu zasilania urządzenia;
- Testy urządzenia pod kątem skuteczności działania i czasu pracy na baterii;
- Zaprojektowanie trójwymiarowej obudowy do wykonania za pomocą technologii druku przestrzennego;
- Wykonanie instrukcji obsługi urządzenia.

1.4 Charakterystyka rozdziałów

W skład dalszej części pracy wchodzi zagadnienia związane z realizacją celu projektu. W następnym rozdziale przeprowadzono analizę tematu na podstawie istniejących rozwiązań dostępnych na rynku oraz sporządzono założenia realizacyjne i wstępny schemat urządzenia. W kolejnym rozdziale ustalono wszelkie wymagania oraz opisano przypadki użycia i narzędzia wykorzystywane do realizacji projektu. Rozdział "Specyfikacja zewnętrzna" zawiera prezentację i opis urządzenia, informacje dla kogo jest przeznaczone, scenariusze korzystania z systemu, a także obszerną instrukcję użytkownika. Poruszono również kwestie związane z bezpieczeństwem. W następnym rozdziale przedstawiono architekturę urządzenia, wybrane komponenty i użyte biblioteki. Opisano również proces tworzenia oprogramowania oraz algorytm pracy urządzenia, a także budowę i działanie procesu konfiguracji oraz prezentacji danych poprzez witrynę internetową. W rozdziale "Weryfikacja i walidacja" opisano przeprowadzone testy oraz wykryte i usunięte błędy. W ostatnim rozdziale podsumowano pracę oraz przedstawiono wnioski, a także opisano możliwe kierunki rozwoju.

Rozdział 2

Analiza tematu

Na rynku istnieje wiele gotowych komercyjnych urządzeń umożliwiających lokalizowanie pojazdu, przeanalizowano rozwiązania dostępne w popularnym polskim serwisie aukcyjnym. W segmencie cenowym do 100 zł wszystkie przeglądane urządzenia wyposażono w moduł GSM/GPS. Urządzenia realizują tę funkcjonalność, wykorzystując sygnał satelitarny i telefonię komórkową. Większość z przeglądanych urządzeń umożliwia prezentację wyników pracy poprzez wysłanie wiadomości tekstowej zawierającej współrzędne geograficzne lub hiperłącze do mapy z zaznaczoną lokalizacją. Przeglądane urządzenia wysyłają wiadomość zwrotną z ustaloną lokalizacją po wykonaniu połączenia głosowego na numer użytej karty SIM. Zdecydowana większość tego typu urządzeń musi być na stałe podłączona do instalacji elektrycznej i nie jest wyposażona w zasilanie awaryjne. W przypadku kradzieży mienia wyposażonego w wyżej opisane lokalizatory, odłączenie akumulatora pojazdu całkowicie pozbawia je funkcjonalności.

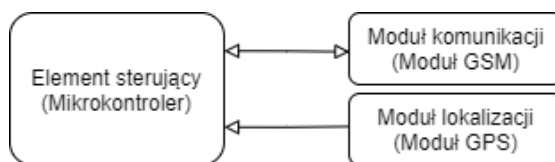
2.1 Założenia realizacyjne

Analizując istniejące rozwiązania, zdecydowano się zapewnić dwa sposoby dostarczania danych do użytkownika oraz przechowywać ostatnio ustaloną lokalizację w pamięci urządzenia. Ponadto zdecydowano się również wyposażyć lokalizator w awaryjne źródło zasilania. Ustalono też, że w skonfigurowanym urządzeniu proces komunikacji powinien być zrealizowany tak, by umożliwić zarządzanie urządzeniem

na odległość. Konfiguracja urządzenia powinna odbywać się z wykorzystaniem technologii łączności bezprzewodowej, a jej wykonanie ma być możliwe, używając w tym celu przykładowo smartfona. Urządzenie powinno cyklicznie sprawdzać swoje położenie geograficzne i wysyłać je do serwera zlokalizowanego w chmurze internetowej. Co więcej, lokalizator powinien wysyłać ostatnio znaną lokalizację na żądanie użytkownika – po odebraniu ustalonej komendy urządzenie powinno wysłać informację zwrotną z odpowiednimi danymi na wcześniej skonfigurowany numer telefonu komórkowego. Urządzenie powinno mieć także możliwość zdalnego restartu, zmiany okresu cyklicznego sprawdzania lokalizacji oraz resetowania konfiguracji do ustawień początkowych.

Projekt zakłada również wykonanie prostej witryny internetowej, która w sposób klarowny będzie prezentowała ostatnio znaną pozycję lokalizatora na mapie. Zdecydowano, że w treści witryny powinna znajdować się informacja o szczegółowej dacie ustalenia wyświetlanej lokalizacji oraz ostatnio odczytany stan poziomu baterii urządzenia.

2.2 Schemat blokowy urządzenia



Rysunek 2.1: Schemat blokowy urządzenia

Wstępny schemat blokowy urządzenia z pominięciem sekcji zasilania zakłada użycie trzech modułów: sterującego, komunikacji i lokalizacji. W znacznym uproszczeniu moduł lokalizacji ma wysyłać do elementu sterującego dane o lokalizacji, następnie moduł sterujący ma za zadanie przetworzyć te dane i wysłać używając modułu komunikacji. Moduł komunikacji ma pośrednio służyć również do zdalnego zarządzania elementem sterującym za pomocą wiadomości tekstowych. Każdy z przedstawionych na schemacie elementów zostanie dokładnie opisany w dalszej części pracy, przeprowadzone zostaną również rozważania na temat doboru elementów konstrukcyjnych urządzenia.

Rozdział 3

Wymagania i narzędzia

3.1 Opis narzędzi

Podczas wykonywania projektu posiłkowano się sporą ilością narzędzi oferowanych przez zewnętrzne firmy. Poniżej krótko opisano niektóre z nich.

3.1.1 Microsoft Visual Studio Code

Microsoft Visual Studio Code – to w ostatnich czasach bardzo popularny międzyplatformowy darmowy edytor kodu źródłowego udostępniający mnogą ilość funkcji ułatwiających pisanie oraz zarządzanie kodem m.in.:

- Kolorowanie składni;
- Znaczniki błędów i ostrzeżeń;
- Wizualizator struktury;
- Śledzenie zmian;
- Zaawansowane formatowanie tekstu;
- IntelliSense dla różnych języków programowania (uzupełnianie kodu, informacje o parametrach, podpowiedzi).

3.1.2 PlatformIO IDE

PlatformIO IDE – środowisko deweloperskie dla programowania systemów wbudowanych instalowane jako rozszerzenie dla Microsoft Visual Code. Środowisko umożliwia programowanie 40 różnych platform m.in.: Atmel AVR, Espressif 32, Espressif 8266, Microchip PIC32, czy ST STM32. Dodatkowo jest wyposażone w zbiór ponad 10 000 bibliotek, które dołącza się używając wbudowanego menedżera bibliotek. PlatformIO umożliwia nie tylko kompilację i wgrywanie oprogramowania, ale również wgrywanie plików do pamięci FLASH oraz odczyt portu szeregowego. Niestety dla mikrokontrolera Espressif 8266 nie jest dostępne narzędzie do debugowania w czasie rzeczywistym, a co za tym idzie, nie ma możliwości ustawienia punktu przerwania i ponownego wznowienia działania programu.

3.1.3 Autodesk Fusion 360

Autodesk Fusion 360 – zaawansowane oprogramowanie działające w chmurze internetowej zawierające narzędzia niezbędne do stworzenia prototypu i przetestowania jego działania. W tym przypadku potencjał oprogramowania został wykorzystany w minimalnym stopniu, gdyż użyto go jedynie do zaprojektowania trójwymiarowego modelu. Mimo braku wcześniejszego doświadczenia w modelowaniu obiektów trójwymiarowych oprogramowanie jest na tyle intuicyjne, iż po zapoznaniu się z krótkim filmem instruktażowym, bez problemu udało się zaprojektować obudowę dla urządzenia. Należy wspomnieć, że środowisko jest płatne dla użytku komercyjnego, aczkolwiek dla studentów oraz na potrzeby edukacyjne dostępna jest odnawialna roczna licencja.

3.1.4 ThingSpeak

ThingSpeak – to platforma analityczna Internetu rzeczy udostępniająca interfejs API (ang. *Application Programming Interface*) do wysyłania i pobierania danych za pomocą protokołów HTTP (ang. *Hypertext Transfer Protocol*) i MQTT (ang. *Message Queue Telemetry Transport*). Platforma pozwala zbierać, wizualizować oraz przetwarzać dane w czasie rzeczywistym za pomocą chmury internetowej. Dodatkowo platforma współpracuje z MATLAB'em oraz wspiera popularne mo-

duży takie jak: Arduino Uno, ESP8266 oraz Raspberry Pi. Dla komercyjnego użytku usługa jest płatna, aczkolwiek dla małych niekomercyjnych projektów (mniej niż 8200 wiadomości na dzień) jest darmowa. Przetwarzanie danych w chmurze odbywa się za pomocą MATLABa i nie jest wymagana do tego procesu dodatkowa licencja MathWorks.

3.1.5 Leaflet

Leaflet [6] – otwarta biblioteka JavaScript używana do tworzenia interaktywnych map internetowych wykorzystująca mapy OpenStreetMap. Biblioteka jest dobrze udokumentowana, a na stronie twórców dostępna jest instrukcja użytkownika i wiele przykładów. Dodatkowo biblioteka jest lekka, a jej funkcjonalność może być rozszerzana dodatkowymi wtyczkami. Ponadto działa wydajnie na wszystkich głównych platformach stacjonarnych, jak i urządzeniach mobilnych.

3.1.6 GitHub

GitHub – internetowy serwis hostingowy dla projektów informatycznych wykorzystujących system kontroli wersji Git. GitHuba używano do przechowywania i tworzenia kopii zapasowych plików projektów, a także do sprawdzania zmian i testowania poszczególnych wersji.

3.1.7 GitKraken

GitKraken – to narzędzie posiadające prosty interfejs graficzny, przeznaczone do zarządzania repozytoriami np. z platformy GitHub. Znacznie ułatwia i przyspiesza wszelkie prace wykonywane na repozytorium. Umożliwia połączenie konta z platformą GitHub, pozwalając na szybkie klonowanie i późniejsze aktualizowanie zmian zawartości repozytorium.

3.2 Wymagania funkcjonalne i нефunkcjonalne

3.2.1 Wymagania funkcjonalne

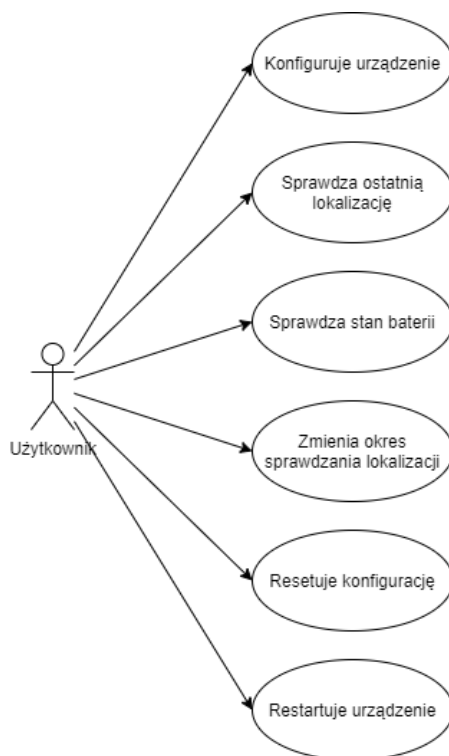
- Urządzenie cyklicznie sprawdza swoją lokalizację;
- Urządzenie pobiera aktualną datę i godzinę;
- Urządzenie umożliwia dwustronną komunikację wykorzystując wiadomości tekstowe;
- Urządzenie cyklicznie wysyła dane do serwera MQTT;
- Urządzenie raportuje o niskim stanie baterii;
- Urządzenie umożliwia konfigurację za pomocą interfejsu WiFi;
- Urządzenie sygnalizuje poprawną pracę za pomocą diod LED;
- Urządzenie sygnalizuje podłączenie do zasilania za pomocą diody LED;
- Użytkownik ma możliwość uzyskania informacji o lokalizacji urządzenia za pomocą wiadomości tekstowej;
- Użytkownik ma możliwość resetowania konfiguracji za pomocą wiadomości tekstowej;
- Użytkownik ma możliwość zmiany okresu sprawdzania lokalizacji;
- Użytkownik ma możliwość restartu urządzenia zdalnie lub za pomocą fizycznego przycisku;
- Użytkownik ma możliwość przybliżania i oddalania mapy na stronie internetowej;
- Użytkownik ma możliwość sprawdzenia dokładnej lokalizacji na mapie (współrzędnych geograficznych).

3.2.2 Wymagania нефunkcjonalne

- Urządzenie powinno być proste w konfiguracji, by nawet osoba średnio obeznana z technologią mogła je skonfigurować z pomocą instrukcji;
- Urządzenie powinno być jak najprostsze w instalacji, by nie wprowadzać bezpowrotnych zmian w konstrukcji wnętrza pojazdu;
- Urządzenie po podłączeniu zasilania powinno być bezobsługowe i działać stabilnie;
- Użytkownik powinien mieć możliwość uzyskania informacji o lokalizacji urządzenia nie mając dostępu do sieci internetowej;
- Strona internetowa jest przejrzysta i w prosty sposób przekazuje lokalizację pojazdu oraz informacje o baterii;
- Lokalizacja pojazdu na stronie internetowej jest prezentowana w centralnej części strony na mapie administracyjnej;
- Motyw kolorystyczny strony internetowej to odcienie zielonego;
- Urządzenie powinno działać na zasilaniu awaryjnym co najmniej przez 12 godzin.

3.3 Przypadki użycia

3.3.1 Przypadki użycia dla urządzenia



Rysunek 3.1: Diagram przypadków użycia

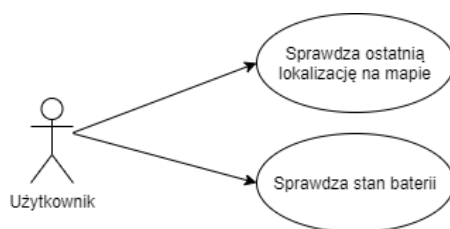
Aktorem dla wszystkich przypadków użycia jest użytkownik urządzenia – tym mianem określa się osobę, która korzysta z funkcjonalności urządzenia (lokalizatora) zgodnie z jego przeznaczeniem i opisem zawartym w instrukcji obsługi dołączonej do urządzenia. Na potrzeby spełnienia wymagań projektu przewidziano sześć różnych przypadków użycia:

- Konfiguracja urządzenia – urządzenie uruchamia się w trybie konfiguracji, a użytkownik za pomocą zewnętrznego urządzenia wprowadza wymagane dane, które zostają zapisane w pamięci urządzenia;
- Sprawdzenie ostatniej lokalizacji – urządzenie jest uruchomione w trybie normalnej pracy, użytkownik wysyła wiadomość z żądaniem informacji, urządze-

nie odbiera żądanie i wysyła wiadomość zwrotną zawierającą informację o ostatniej lokalizacji;

- Sprawdzenie stanu baterii – urządzenie jest uruchomione w trybie normalnej pracy, użytkownik wysyła wiadomość z żądaniem informacji, urządzenie odbiera żądanie i wysyła wiadomość zwrotną zawierającą informację o stanie baterii;
- Zmiana okresu sprawdzania lokalizacji – urządzenie jest uruchomione w trybie normalnej pracy, użytkownik wysyła wiadomość z komendą zmiany częstotliwości sprawdzania lokalizacji, urządzenie odbiera żądanie i zmienia okres sprawdzania ostatniej lokalizacji;
- Reset konfiguracji – urządzenie jest uruchomione w trybie normalnej pracy, użytkownik wysyła wiadomość z komendą resetowania konfiguracji, urządzenie odbiera żądanie i usuwa aktualną konfigurację i uruchamia się ponownie w trybie konfiguracji;
- Restart urządzenia – urządzenie jest uruchomione w trybie normalnej pracy, użytkownik wysyła wiadomość z komendą restartu, urządzenie odbiera żądanie, a następnie wyłącza się i włącza ponownie nie zmieniając trybu pracy.

3.3.2 Przypadki użycia dla witryny internetowej



Rysunek 3.2: Diagram przypadków użycia witryny internetowej

Aktorem dla wszystkich przypadków użycia jest użytkownik witryny internetowej – tym mianem określa się osobę, która odwiedza stronę i korzysta z dostępnych funkcjonalności. Na potrzeby spełnienia wymagań projektu przewidziano dwa przypadki użycia:

- Sprawdzenie ostatniej lokalizacji na mapie – użytkownik odwiedza stronę w celu sprawdzenia informacji o ostatniej lokalizacji urządzenia, strona prezentuje położenie geograficzne lokalizatora w formie markera na odpowiednim obszarze mapy administracyjnej;
- Sprawdzenie stanu baterii – użytkownik odwiedza stronę w celu sprawdzenia informacji o stanie poziomu baterii lokalizatora, strona prezentuje stan baterii urządzenia w postaci paragrafu z ostatnio odczytanym procentowym zużyciem poziomu baterii lokalizatora.

3.4 Proces pracy nad projektowaniem i implementacją

Przed przystąpieniem do właściwej realizacji projektu wgrywano na urządzenie przykłady oprogramowania dostarczone przez producenta modułu i testowano ich działanie. Przykłady pozwoliły na przetestowanie włączania i wyłączania urządzenia, a także na sprawdzenie komunikacji i zarządzanie modułem GSM/GPS za pomocą komend AT. W celu ułatwienia procesu testowania, po skompletowaniu wszystkich komponentów zaprojektowano i oddano do wydruku obudowę dla urządzenia. W pierwszej kolejności zaprojektowano wstępne wymagania działania oprogramowania. Następnie kolejno implementowano poszczególne moduły funkcjonalne, rozwijając oprogramowanie. Każda kolejna funkcjonalność po implementacji była wgrywana na urządzenie i niezwłocznie testowana, co pozwoliło na szybkie wykrycie i eliminację ewentualnych błędów.

Moduł debugowano za pomocą wysyłania odpowiednich komunikatów poprzez port szeregowy, wykorzystując uniwersalny asynchroniczny nadajnik-odbiornik i transceiver UART-USB. Wiadomości odbierano na komputerze za pomocą konsoli portu szeregowego dostępnej w PlatformIO IDE.

Rozdział 4

Specyfikacja zewnętrzna

4.1 Opis urządzenia

Lokalizator to urządzenie mające funkcję ustalania własnej lokalizacji. Urządzenie konfiguruje się, a następnie instaluje w pojeździe, umożliwiając sprawdzanie jego aktualnego położenia geograficznego.

W zaprojektowanym urządzeniu przewidziano zarówno komunikację za pomocą wiadomości tekstowych (zarządzanie urządzeniem i informacja zwrotna o lokalizacji i stanie baterii), jak i cykliczne wysyłanie informacji o ustalonej lokalizacji oraz stanie baterii do serwera, skąd nawet w razie awarii urządzenia istnieje możliwość pobrania ostatnio zapisanej lokalizacji. Dzięki zaimplementowaniu dwóch sposobów prezentacji danych użytkownik może sprawdzić lokalizację pojazdu zarówno za pomocą dowolnego urządzenia z dostępem do Internetu używając witryny internetowej (aplikacji webowej), jak i za pomocą telefonu wysyłając wiadomość tekstową o odpowiedniej treści, otrzymując odpowiedź zwrotną na wcześniej skonfigurowany numer karty SIM. Dodatkowo urządzenie wyposażono w zasilanie awaryjne, dzięki czemu jest w stanie nieprzerwanie monitorować lokalizację pojazdu nawet w przypadku odłączenia zasilania w pojeździe.



Rysunek 4.1: Lokalizator w zamkniętej obudowie podłączony do zasilania

4.2 Kategorie użytkowników

Urządzenie jest przeznaczone dla szerokiego grona użytkowników, w dowolnym przedziale wiekowym. Do konfiguracji i późniejszego użytkowania lokalizatora wystarczą podstawowe umiejętności korzystania z telefonu komórkowego oraz podstawowe umiejętności obsługi komputera personalnego wraz z obeznaniem w zakresie przeglądania witryn internetowych. Lokalizator może być używany zarówno poprzez osoby prywatne, chcące w pewien sposób monitorować i kontrolować swój pojazd podczas nieobecności, jak i przez przedsiębiorców chcących przykładowo monitorować pojazd firmowy udostępniony pracownikowi w ramach wykonywanej pracy. Właściciel pojazdu, w którym zainstalowano urządzenie, ma możliwość

zdalnego sprawdzenia ostatniej lokalizacji w dowolnym momencie za pomocą wiadomości tekstowej lub używając witryny internetowej.

4.3 Scenariusze korzystania z systemu

4.3.1 Konfiguracja urządzenia

W procesie konfiguracji urządzenia użytkownik zobowiązany jest do podłączenia urządzenia do źródła zasilania, a następnie konfiguracji podstawowych parametrów wymaganych do prawidłowej pracy lokalizatora. Proces konfiguracji odbywa się poprzez stworzony przez urządzenie punkt dostępowy, do którego użytkownik musi się połączyć, używając dowolnego urządzenia obsługującego sieci bezprzewodowe i umożliwiającego przeglądanie witryn internetowych.

4.3.2 Obsługa za pomocą wiadomości tekstowych

Użytkownik ma możliwość obsługi urządzenia za pomocą komend wysyłanych jako wiadomości tekstowe z poziomu telefonu komórkowego. Dostępne komendy to: zmiana częstotliwości sprawdzania lokalizacji i wysyłania jej do serwera MQTT, ponowne uruchomienie urządzenia, żądanie przesłania danych ostatniej lokalizacji oraz zresetowanie urządzenia do ustawień fabrycznych (wymagana ponowna konfiguracja). Aby użyć danej komendy (spis komend wraz z przykładem użycia dostępny w instrukcji obsługi urządzenia) należy wysłać wiadomość tekstową o odpowiedniej treści na numer karty SIM pracującego lokalizatora.

4.3.3 Odczyt danych za pomocą wiadomości tekstowych

Po wysłaniu żądania przesłania informacji o ostatniej lokalizacji w postaci wiadomości tekstowej, urządzenie po przetworzeniu żądania wyśle zwrotną wiadomość tekstową z: ostatnio zapisaną dokładną lokalizacją w formie współrzędnych geograficznych, dokładną datą tej lokalizacji z uwzględnieniem strefy czasowej oraz stanem baterii urządzenia.

4.3.4 Odczyt danych za pomocą witryny internetowej

Stronę internetową do odczytu danych z lokalizatora użytkownik otrzymuje w formie adresu witryny internetowej. Po załadowaniu skrypt pobiera dane z zewnętrznego serwera ThingSpeak, a następnie prezentuje je na witrynie internetowej – lokalizację w formie znacznika na mapie administracyjnej, a pozostałe dane w postaci akapitów zlokalizowanych pod mapą. Dane są automatycznie odświeżane.

4.4 Instrukcja obsługi

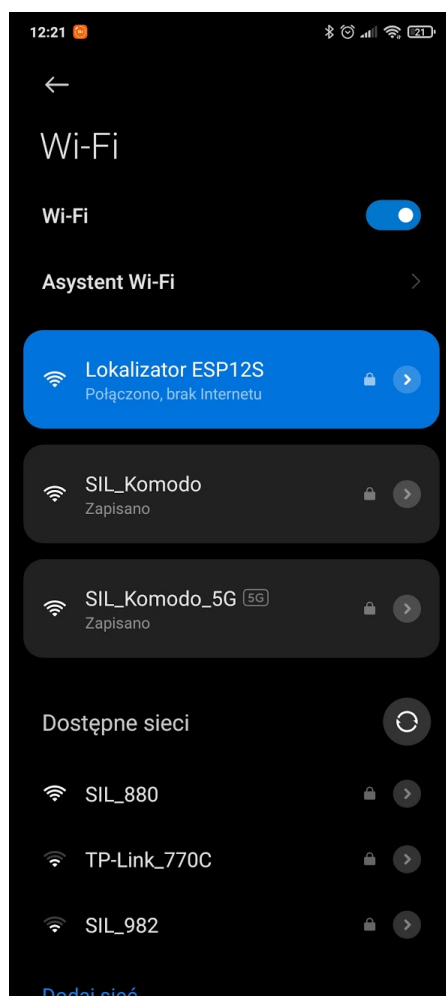
4.4.1 Pierwsze uruchomienie i konfiguracja

Przed pierwszym uruchomieniem należy otworzyć górną pokrywę urządzenia i wyciągnąć plastikową zawleczkę separującą awaryjne źródło zasilania, po czym z powrotem zamknąć pokrywę.



Rysunek 4.2: Lokalizator z otwartą górną pokrywą podłączony do zasilania

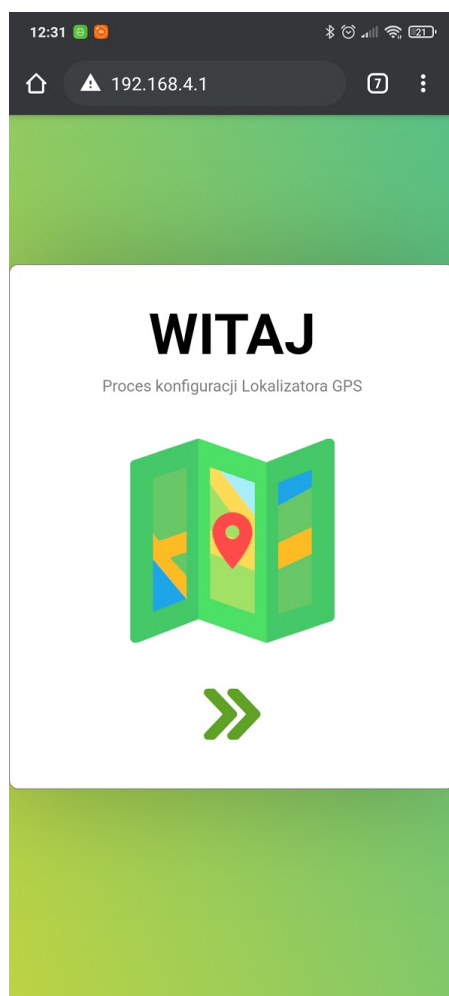
Moduł należy podłączyć do zasilania poprzez złącze MicroUSB. Można w tym celu użyć zwykłego gniazda USB dostępnego w komputerze, laptopie, samochodzie lub użyć ładowarki telefonu komórkowego. Po prawidłowym podłączeniu zasilania, na urządzeniu zaświeci się czerwona dioda zlokalizowana obok złącza zasilania.



Rysunek 4.3: Utworzony punkt dostępowy

Następnie należy połączyć dowolne urządzenie obsługujące WiFi 2.4GHz (np. smartfon lub laptop) do wcześniej stworzonego punktu dostępowego o identyfikatorze "Lokalizator ESP12S" używając hasła: "L0k4liz@t0r#01".

Kolejnym krokiem jest konfiguracja urządzenia za pomocą przeglądarki internetowej. W pole adresowe przeglądarki należy wpisać adres witryny – "http://192.168.4.1/" (domyślny adres IP urządzenia). Po prawidłowym połączeniu powinien załadować się ekran powitalny kreatora konfiguracji.



Rysunek 4.4: Ekran powitalny kreatora konfiguracji

Podczas konfiguracji należy podążać za wskazówkami kreatora. By przejść do kolejnego etapu, wystarczy nacisnąć przycisk znajdujący się w dolnej części witryny (podwójny grot strzały skierowany w prawą stronę).

12:32 192.168.4.1/config.html

KONFIGURACJA

Potrzebujemy kilku danych wymaganych do konfiguracji Lokalizatora GPS

Numer telefonu
+48515303896

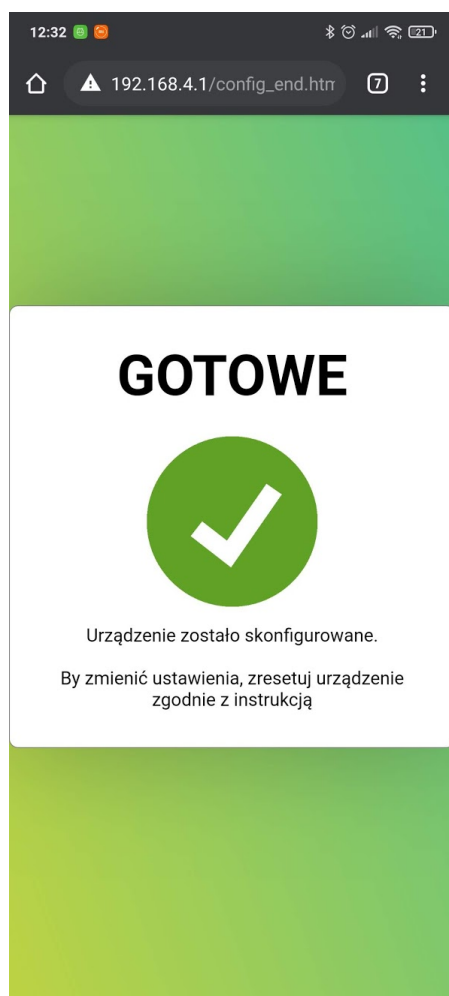
MQTT WriteAPIKey
HRDDRV3GPRPS

MQTT channelID
1048994

>>

Rysunek 4.5: Formularz do uzupełnienia wymaganych danych

W tym kroku należy wprowadzić dane wymagane do prawidłowego działania urządzenia. Wymagane dane to: numer telefonu użytkownika, klucz API zapisu danych dla kanału MQTT oraz identyfikator kanału MQTT. Po uzupełnieniu danych należy ponownie nacisnąć przycisk znajdujący się w dolnej części witryny. Jeżeli dane są prawidłowe, zostanie wyświetlona informacja o prawidłowym zakończeniu procesu konfiguracji. W przeciwnym wypadku użytkownik zostanie poproszony o poprawienie nieprawidłowych danych.



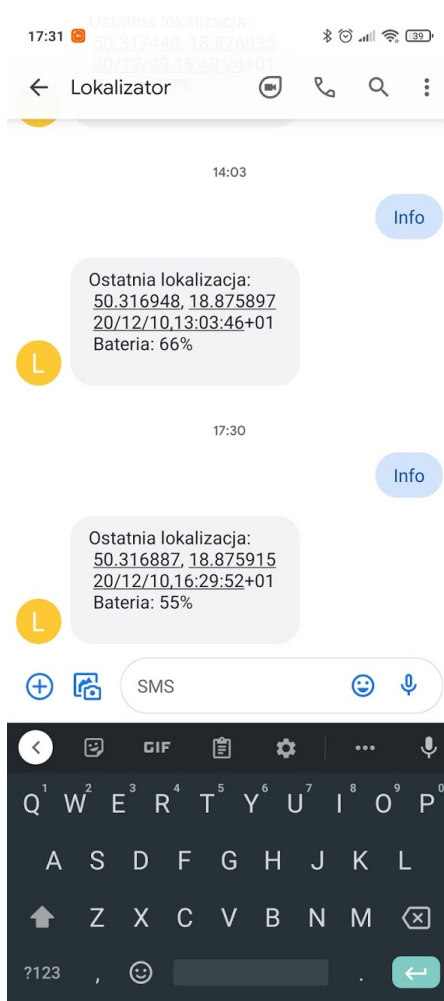
Rysunek 4.6: Zakończenie procesu konfiguracji

Po zakończeniu procesu konfiguracji urządzenie uruchomi się ponownie w trybie normalnej pracy.

4.4.2 Tryb normalnej pracy

W tym trybie pracy urządzenie, domyślnie co minutę, sprawdza swoją lokalizację. Gdy uda się ją ustalić, wysyła dane do wcześniej skonfigurowanego brokera. W każdym cyklu pracy urządzenie wysyła również do brokera informacje o stanie poziomu baterii. Użytkownik może uzyskać informację o ostatnio ustalonej lokalizacji urządzenia, jej szczegółowej dacie oraz stanie baterii urządzenia na dwa sposoby.

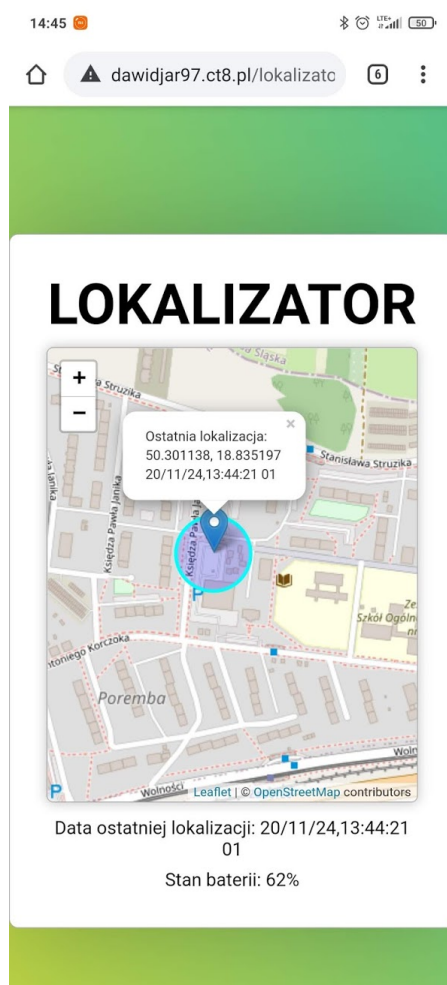
Pierwszym z nich jest wysłanie wiadomości tekstowej na numer karty SIM umieszczonej w module o treści "Info". Po odebraniu wiadomości, moduł powinien wysłać wiadomość zwrotną zawierającą: lokalizację w formie współrzędnych geograficznych, datę ustalonej lokalizacji z uwzględnieniem strefy czasowej oraz informację o stanie baterii.



Rysunek 4.7: Informacje uzyskane za pomocą komendy "Info"

W celu uzyskania informacji za pomocą tego sposobu użytkownik musi posiadać urządzenie podłączone do sieci komórkowej umożliwiające wysyłanie oraz odbieranie wiadomości tekstowych np. telefon komórkowy lub smartfon z aktywną kartą SIM.

Drugim sposobem na uzyskanie informacji jest odwiedzenie witryny internetowej dołączonej do urządzenia. Ostatnia lokalizacja powinna być widoczna na mapie administracyjnej w formie markera. Dodatkowo klikając na marker użytkownik może uzyskać szczegółowe współrzędne geograficzne, a pod mapą znajdują się informacje o dokładnej dacie lokalizacji oraz stanie poziomu baterii urządzenia.



Rysunek 4.8: Informacja uzyskane za pomocą witryny internetowej

Do odczytu danych za pomocą witryny internetowej wymagane jest dowolne urządzenie mające dostęp do sieci internetowej wyposażone w przeglądarkę umożliwiającą wyświetlanie zawartości stron internetowych (np. komputer stacjonarny z zainstalowaną przeglądarką Google Chrome podłączony do sieci Internet).

Wykorzystanie pierwszego sposobu gwarantuje uzyskanie najnowszych danych zebranych przez urządzenie.

W przypadku pracy na zasilaniu awaryjnym (po odłączeniu głównego źródła zasilania), gdy stan baterii osiągnie poziom poniżej wartości 20%, urządzenie poinformuje użytkownika o niskim stanie poziomu baterii za pomocą wiadomości tekstowej, wysłanej na wcześniej skonfigurowany numer telefonu.

4.4.3 Zarządzanie urządzeniem

Użytkownik ma możliwość zdalnego zarządzania lokalizatorem używając specjalnych komend wysyłanych za pomocą wiadomości SMS (ang. *Short Message Service*). Lista dostępnych komend:

- Zmiana częstotliwości sprawdzania lokalizacji urządzenia i wysyłania ich do serwera – za pomocą wiadomości tekstowej o treści: "ChangeTaskPeriod: wartość", gdzie wartość to nowy odstęp czasu w milisekundach między kolejnymi cyklami pracy urządzenia; Prawidłowe wartości do ustawienia to przedział zamknięty od 10000[ms] (10 sekund) do 900000[ms] (15 minut). Jeżeli użytkownik poda wartość spoza zakresu, to czas zostanie ustawiony odpowiednio na najwyższą lub najniższą możliwą wartość;
- Ponowne uruchomienie urządzenia – za pomocą wiadomości tekstowej o treści: "R3ST4RT", po odebraniu wiadomości urządzenie zostanie wyłączone, a następnie włączone ponownie;
- Reset konfiguracji urządzenia – za pomocą wiadomości tekstowej o treści: "?!R3S3T.", po odebraniu wiadomości urządzenie usunie aktualną konfigurację i zostanie przywrócone do stanu fabrycznego. Następnie lokalizator uruchomi się w trybie konfiguracji. Uwaga, po resecie konfiguracji urządzenia, należy je ponownie skonfigurować.

4.4.4 Sposób instalacji

Urządzenie należy zainstalować w pojeździe, który ma być lokalizowany. Poleca się montaż urządzenia w zamkniętym i niewidocznym miejscu za pomocą taśmy dwustronnej w schowku pojazdu, zazwyczaj w przedniej konsoli nad nogami pasażera lub między przednimi fotelami w tunelu środkowym. Do urządzenia należy doprowadzić kabel zasilający uprzednio podłączony do gniazda USB w pojeździe. Po podłączeniu należy upewnić się, że zasilanie działa prawidłowo – sygnalizowane czerwoną diodą przy wtyku zasilania. Warto również w tym momencie sprawdzić, czy komunikacja z urządzeniem za pomocą wiadomości tekstowych działa poprawnie.

4.5 Kwestie bezpieczeństwa

Kwestie bezpieczeństwa w urządzeniach tego typu są bardzo ważne. Osoby z zewnątrz mające dostęp do konfiguracji lub metod zarządzania urządzeniem mogłyby w łatwy sposób zaburzyć jego pracę na przykład poprzez usunięcie konfiguracji.

Aby żadna osoba trzecia nieposiadająca danych dostępowych do urządzenia nie mogła zmienić konfiguracji, punkt dostępowy lokalizatora został zabezpieczony hasłem z wykorzystaniem technologii szyfrowania WPA2 (ang. *Wi-Fi Protected Access II*), a sam reset konfiguracji nie jest możliwy za pomocą fizycznego przycisku.

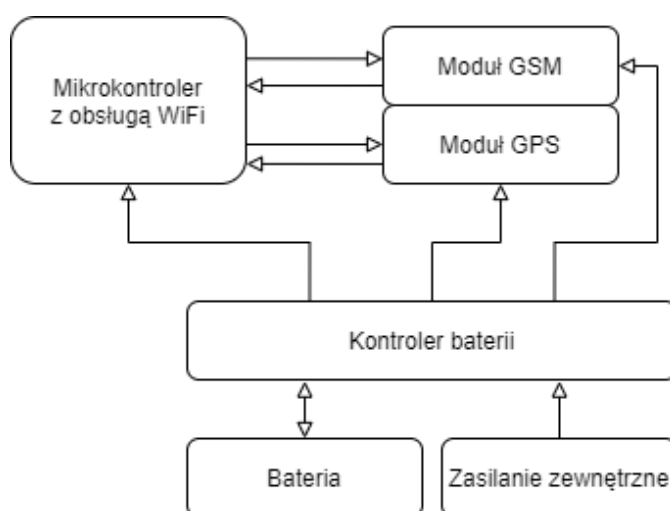
Ponadto osoba trzecia, nie znając numeru karty SIM, nie jest w stanie w żaden sposób kontrolować pracy urządzenia. Co więcej, nawet w przypadku gdy osoba trzecia spisałaby numer kontaktowy bezpośrednio z karty SIM, komenda resetowania konfiguracji urządzenia nie jest jej znana i jedynym sposobem przerwania pracy lokalizatora jest odłączenie zasilania – dla tego przypadku przewidziano zasilanie awaryjne.

Osoba działająca w złej intencji, próbująca przerwać pracę lokalizatora, może odłączyć przewód zasilania awaryjnego, w sposób mechaniczny odseparować ogniwo akumulatora lub po prostu uszkodzić urządzenie. Przed tego typu działaniami bardzo ciężko w pełni zabezpieczyć produkt, dlatego zaleca się montaż urządzenia w niewidocznym trudno dostępnym miejscu.

Rozdział 5

Specyfikacja wewnętrzna

5.1 Architektura urządzenia



Rysunek 5.1: Architektura systemu

Do realizacji projektu potrzebne były przede wszystkim: mikrokontroler zarządzający wyposażony w obsługę sieci WiFi, moduł GSM obsługujący transmisję danych w sieci Internet i komunikację SMS, moduł GPS pozwalający na uzyskanie dokładnej lokalizacji oraz akumulator pozwalający na awaryjne zasilanie układu. Dodatkowo do właściwego funkcjonowania urządzenia niezbędne były takie elementy jak: kontroler ładowania, regulator napięcia, konwerter USB-UART,

a także odpowiednie anteny.

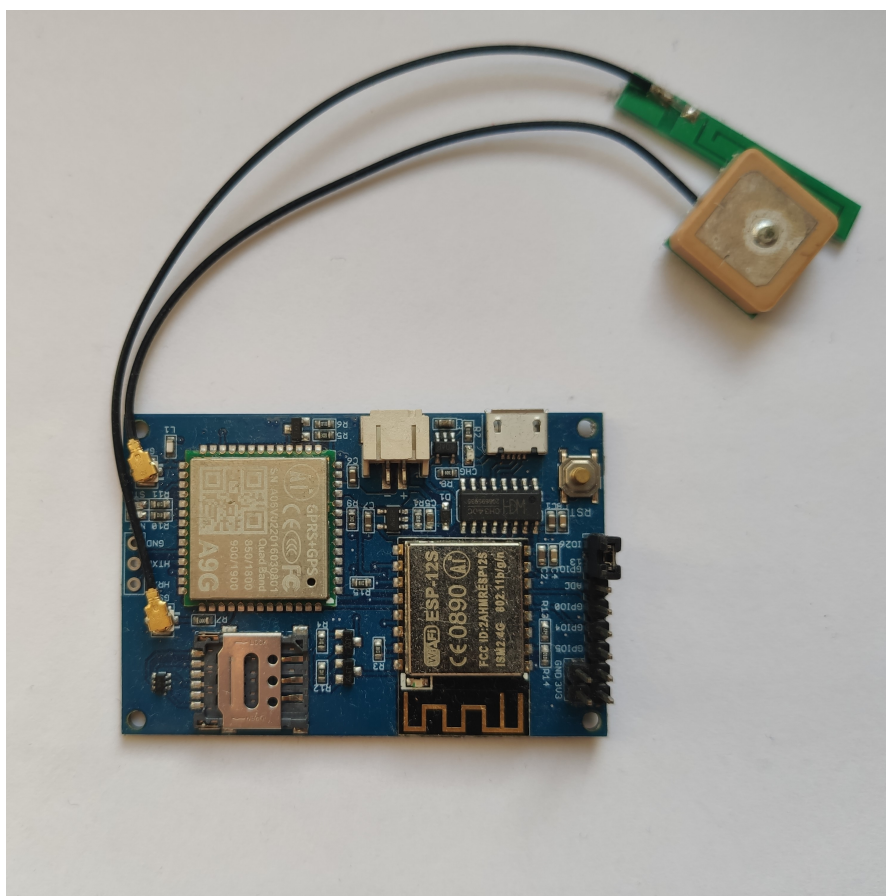
5.2 Wybrane komponenty

Przy poszukiwaniu potrzebnych elementów zwrócono uwagę na pobór energii elektrycznej, rozmiar komponentów, stopień trudności użycia w projekcie oraz cenę zakupu. Po rozeznaniu elementów dostępnych na rynku zdecydowano wybrać moduł posiadający prawie wszystkie wymagane elementy – AiThinker ESP12S A9G GPRS Node v1.0 oraz akumulator typu 18650 firmy Samsung jako zasilanie awaryjne.

5.2.1 AiThinker ESP12S A9G GPRS Node v1.0

Płytką wyposażoną jest w moduł Espressif ESP-12S, moduł GSM/GPS AiThinker A9G, kontroler ładowania MCP73831, liniowy regulator napięcia LDO AP2112, konwerter USB-UART, a także anteny GPS i GSM [4]. Oprócz wyżej wymienionych elementów moduł posiada złącze MicroUSB do programowania, ładowania baterii i zasilania urządzenia, a także złącze JST2 służące do podłączenia baterii. Ponadto na płytce znajdują się cztery dodatkowe wyprowadzenia ogólnego przeznaczenia, a jedno z nich to wyprowadzenie przetwornika analogowo-cyfrowego. Wyprowadzenia można również wykorzystać do obsługi magistrali I2C (ang. *Inter-Integrated Circuit*), co pozwala rozbudować układ o dodatkowe moduły. Dodatkowo na płytce znajdują się również diody sygnalizujące podłączenie zewnętrznego zasilania oraz pracę modułu GSM/GPS. Powodem wyboru właśnie tego urządzenia jest stosunkowo niska cena oraz ogromna możliwość tworzenia wszelakich funkcjonalności bez dołączania dodatkowych elementów. Moduł spełnia wszystkie wymagania projektu, a dodatkowo umożliwia jego przyszły rozwój.

ESP-12S bazuje na mikrochipie ESP8266 [2]. ESP8266 to bardzo popularny SoC (ang. *System-on-a-chip*) firmy Espressif wyposażony w moduł komunikacji bezprzewodowej WiFi stosowany w wielu urządzeniach Internetu Rzeczy. Układ oferuje ogromne możliwości w niskiej cenie, a dodatkowo umożliwia programowanie za pomocą Arduino IDE lub PlatformIO IDE.



Rysunek 5.2: Moduł AiThinker ESP12S A9G GPRS Node v1.0

AiThinker A9G to kompletny czterozakresowy moduł GSM/GPS obsługujący technologię GPRS (ang. *General Packet Radio Service*), pozycjonowanie GPS, połączenia głosowe i wiadomości SMS [1]. Komunikacja i zarządzanie modułem odbywa się za pomocą komend AT. Dokładność pozycjonowania poziomego urządzenia wynosi 2.5 m.

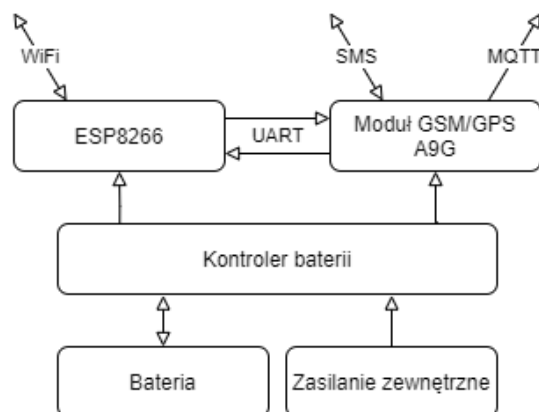
5.2.2 Źródło zasilania

Jako źródło awaryjnego zasilania wybrano akumulator litowo-jonowy 18650 Samsung INR18650-35E o napięciu nominalnym 3.6 V, pojemności 3500 mAh i maksymalnym prądzie rozładowania 8 A (impulsowy 13 A). Ogniwo zostało dobrane tak, by spełnić wymagania zasilania modułu GPS/GSM, który może potrzebować prądu tymczasowego rzędu 2 A. By nie lutować przewodów bezpośrednio do ogniwa, zdecydowano się zastosować koszyk na pojedynczy akumulator 18650.



Rysunek 5.3: Ogniwo akumulatorowe Samsung INR18650-35E

5.3 Schemat blokowy urządzenia

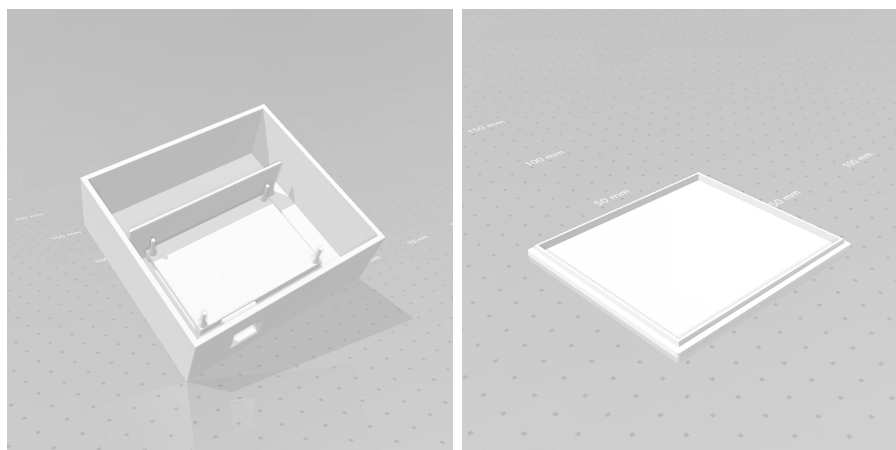


Rysunek 5.4: Schemat blokowy i opis urządzenia

Rolę modułu zarządzającego pełni mikrokontroler ESP8266. Do spełnienia funkcjonalności modułów komunikacji i lokalizacji użyto modułu GSM/GPS AiThin-ker A9G, z którym element zarządzający komunikuje się za pomocą komend AT wykorzystując port szeregowy. Do procesu konfiguracji, zgodnie z założeniami projektu, wykorzystany został moduł komunikacji bezprzewodowej WiFi układu ESP-12S. Do wysyłania danych i odbierania poleceń użytkownika wykorzystano usługę SMS. Do wysyłania danych do serwera użyto lekkiego protokołu MQTT, opartego o wzorzec publikacja/subskrypcja (ang. *publish/subscribe*).

5.4 Obudowa urządzenia

Na wstępnym etapie projektowania obudowy zwymiarowano wszystkie komponenty. Następnie stworzono wstępny szkic i model obudowy wycinając go ze styroduru. Kolejnym krokiem było wykonanie wirtualnego trójwymiarowego modelu, w którym przewidziano odpowiednie mocowania i odseparowanie komponentów urządzenia. Zaplanowano także miejsce na częściowe ukrycie przewodów zasilania oraz otwór na wtyk przewodu zasilania. Trójwymiarowy model obudowy zaprojektowano z użyciem oprogramowanie Autodesk Fusion 360, a następnie zlecono do wydrukowania z grafitowego PLA (ang. *Polylactic Acid*) w technologii druku przestrzennego.



Rysunek 5.5: Trójwymiarowy model obudowy i górnej pokrywy

5.5 Rozmieszczenie elementów w obudowie

Wszystkie elementy urządzenia zostały stabilnie zamocowane we wcześniej przygotowanych dla nich miejscach w obudowie. Do zamocowania dedykowanych anten wykorzystano fabrycznie zainstalowaną na nich taśmę dwustronną. Do zamocowania koszyka na baterię i dodatkowego podklejenia płytki użyto mocnej piankowej taśmy dwustronnej. Przewody zasilania częściowo ukryto w specjalnie zaprojektowanym schowku pod modułem.



Rysunek 5.6: Rozmieszczenie elementów w obudowie – widok z góry



Rysunek 5.7: Rozmieszczenie elementów w obudowie

5.6 Biblioteki

5.6.1 SoftwareSerial

Biblioteka umożliwiająca wykorzystanie programowo zrealizowanego portu szeregowego do komunikacji z zewnętrznymi urządzeniami używając dowolnych wyprowadzeń. Wykorzystywana ze względu na zastosowanie tylko jednego natywnego portu szeregowego w mikrokontrolerze ESP8266. Biblioteka jest częścią paczki ESP8266 dodawanej z menedżera platform PlatformIO. Biblioteka jest udostępniona na warunkach licencji GNU Lesser General Public License, opublikowanej przez Free Software Foundation.

5.6.2 ESPAsyncWebServer

Darmowa biblioteka, która umożliwia stworzenia lokalnego asynchronicznego serwera HTTP używanego w procesie konfiguracji. Ponadto oferuje wiele funkcjonalności takich jak: wiele połączeń w tym samym czasie, pracę poza pętlą loop programu, obsługę WebSocketów, czy łatwą obsługę zdarzeń serwera. Bibliotekę pozyskano z bazy bibliotek PlatformIO IDE, gdzie jest udostępniona na warunkach licencji GNU Lesser General Public License, opublikowanej przez Free Software Foundation.

5.6.3 Configurator

Biblioteka do obsługi pamięci mikrokontrolera, wykorzystuje zawartą w paczce ESP8266 bibliotekę "FS" – operującą na systemie plików SPIFFS (ang. *Serial Peripheral Interface Flash File System*). Została opracowana w celu zwiększenia przejrzystości kodu i zmniejszenia jego objętości, ponieważ jest wykorzystywana zarówno w procesie konfiguracji, jak i podczas właściwej pracy urządzenia. W bibliotece zdefiniowano ścieżkę do pliku zawierającego konfigurację, bit do debugowania urządzenia oraz przypisano wyprowadzenie wykorzystywanej diody LED.

Spis i krótki opis funkcji biblioteki:

- `deleteFile()` – Funkcja do usuwania plików, domyślnie wywoływana dla pliku konfiguracyjnego;

- `isConfigurationCompleted()` – Funkcja sprawdzająca czy plik konfiguracyjny jest dostępny w pamięci;
- `saveConfiguration()` – Funkcja do zapisu, jako parametry przyjmuje zmienne do zapisu w pliku konfiguracyjnym;
- `loadConfiguration()` – Funkcja do odczytu, jako parametry przyjmuje zmienne do których wczytuje dane z pliku konfiguracyjnego.

Dane przechowywane w pliku konfiguracyjnym to: numer telefonu, `writeAPIkey`, `channelID`, ostatnia lokalizacja oraz jej dokładna data.

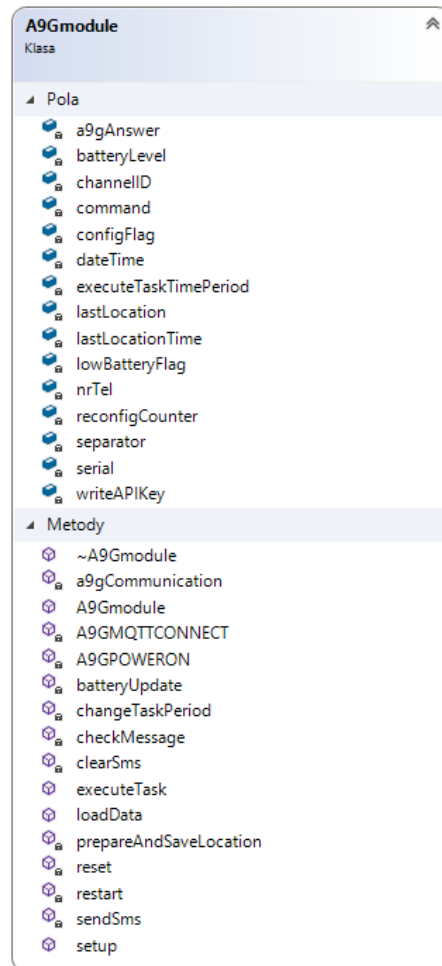
5.6.4 ConfigurationSerwer

Biblioteka opracowana do obsługi procesu konfiguracji. Wykorzystuje biblioteki `Configurator` oraz `ESPAsyncWebServer`. Została stworzona w celu odseparowania procesu konfiguracji i poprawienia czytelności pliku `main.cpp`. W bibliotece zdefiniowano port serwera HTTP, nazwę punktu dostępowego oraz hasło dostępu. W bibliotece tworzona jest instancja lokalnego asynchronicznego serwera HTTP. Biblioteka posiada jedną funkcję – `startServerConfiguration()`, która inicjalizuje punkt dostępowy i obsługuje zdarzenia serwera. Gdy proces konfiguracji zostaje ukończony, plik konfiguracyjny zostaje zapisany, a urządzenie zrestartowane.

5.7 Klasy

Główną i jedyną klasą modułu jest klasa `A9Gmodule`, której instancja odpowiada modelowi rzeczywistego urządzenia. W klasie zawarte są metody odpowiadające za wszystkie funkcjonalności, komunikację, a także harmonogram cyklu pracy. Większość metod jest prywatna i są one wywoływane jedynie wewnątrz klasy poprzez inne metody. W tym rozwiązaniu zastosowano pełną hermetyzację klasy. Spoza klasy użytkownik może jedynie wywołać metody: `setup` – do konfiguracji urządzenia, `loadData` – do załadowania odpowiednich danych z pamięci, `executeTask` – metoda wywoływana w każdej iteracji pętli głównej służąca do uruchamiania kolejnego cyklu pracy modułu.

Klasa posiada wskaźnik do obiektu programowo zrealizowanej komunikacji szeregowej, zmienne do przechowywania danych oraz komend AT, i licznik rekonfiguracji.



Rysunek 5.8: Diagram klas urządzenia

Spis i krótki opis wszystkich metod prywatnych klasy:

- **A9GPOWERON()** – Metoda włączająca moduł GSM/GPS;
- **A9GMQTTCONNECT()** – Metoda inicjująca połączenie internetowe do serwera MQTT. Metoda sprawdza również poprawność zainicjowanego połączenia i restartuje urządzenie w przypadku wystąpienia problemów z łącznością;

- `clearSms()` – Metoda do usuwania wiadomości tekstowych z pamięci modułu GSM/GPS;
- `sendSms(String msg)` – Metoda do wysyłania wiadomości tekstowych, przygotowuje odpowiednią komendę i wysyła ją do modułu GSM/GPS;
- `a9gCommunication(String command, const int timeout)` – Metoda do komunikacji dwustronnej z modułem GSM/GPS. Jeżeli jako argument podano komendę, wysyła ją do modułu GSM/GPS, a następnie oczekuje na wiadomość zwrotną przez podany czas;
- `checkMessage()` – Metoda do sprawdzenia wiadomości zwrotnej. Treść wiadomości zostaje odpowiednio zinterpretowana, jeżeli rozpoznano polecenie, zostaje wywołana odpowiednia akcja;
- `batteryUpdate()` – Metoda do wysyłania żądania sprawdzenia stanu baterii;
- `changeTaskPeriod()` – Metoda do zmieniania częstotliwości wysyłania lokalizacji;
- `reset()` – Metoda do przywracania urządzenia do ustawień fabrycznych;
- `restart()` – Metoda do restartowania urządzenia.

Spis i krótki opis wszystkich metod publicznych klasy:

- Kontruktor – tworzy obiekt programowego zrealizowanego portu szeregowego;
- `setup()` – Metoda do ustawienia modułu w tryb pracy. Rozpoczyna komunikację między modułami i ustawia odpowiednio wyprowadzenia sterujące modułem GSM/GPS. Następnie wywołuje metodę `A9GPOWERON`. Ustawia kolejno kodowanie, tryb tekstowy, wywołuje metodę `A9GMQTTCONNECT`, czyści wiadomości tekstowe, włącza moduł GPS i ustawia pracę w trybie oszczędzania energii;
- `loadData()` – Metoda do wczytywania odpowiednich danych z pamięci urządzenia, wykorzystuje bibliotekę `Configurator`;

- `executeTask()` – Metoda zawierająca pojedynczy cykl pracy urządzenia, wywoływana cyklicznie w głównej pętli programu. W metodzie wywoływane są aktualizacje stanu baterii oraz daty, zapytanie o lokalizację, zapisywanie lokalizacji do pamięci urządzenia, wysyłanie pakietów MQTT oraz wywołanie oczekiwania na komunikację z użytkownikiem.

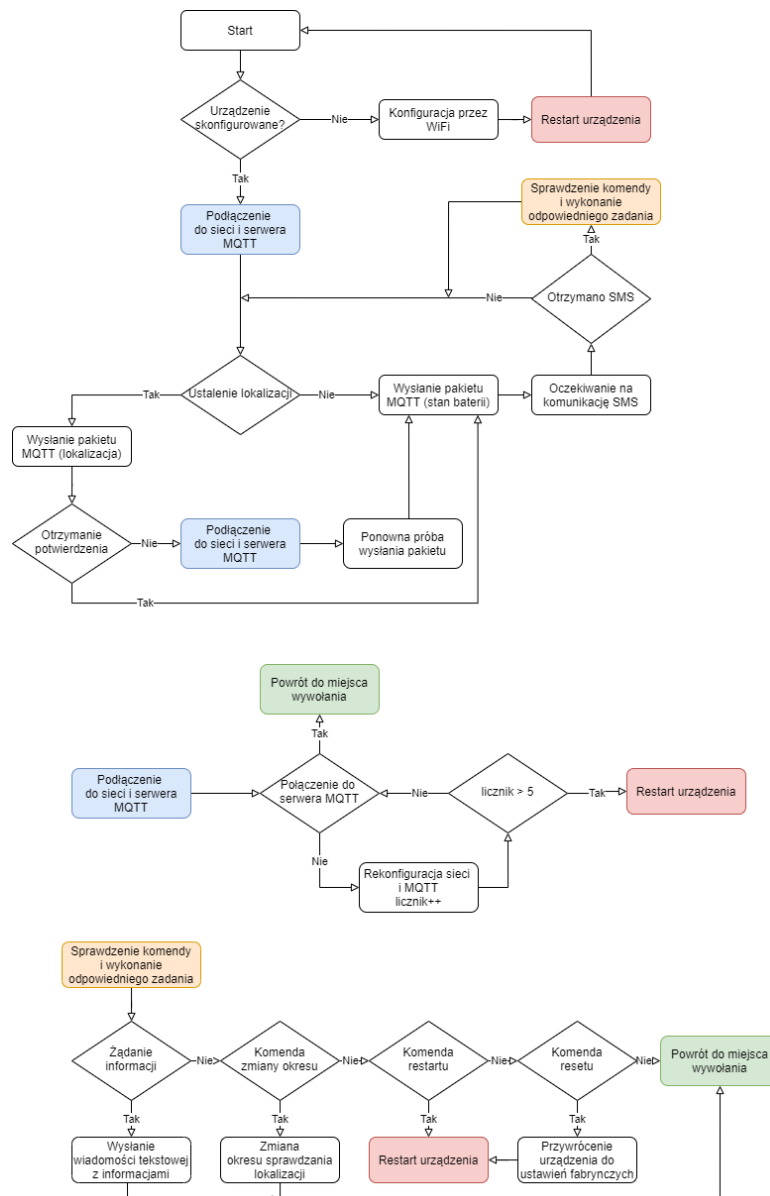
5.8 Tworzenie oprogramowania urządzenia

Obsługę modułu GSM/GPS i konfigurację urządzenia za pomocą WiFi w początkowym etapie zaimplementowano jako całkowicie osobne projekty, a następnie połączono je w współpracującą całość. W następnych etapach dodawano kolejne funkcjonalności urządzenia jako metody klasy do obsługi modułu GSM/GPS.

Do implementacji podstawowych funkcjonalności modułu GSM/GPS częściowo wykorzystano przykłady dostarczone przez producenta umieszczone na platformie GitHub [4]. W dużym stopniu wykorzystano również dokumentację dostarczoną przez producenta modułu, w szczególności tłumacząc jej rozszerzoną wersję z języka chińskiego.

5.9 Algorytm pracy urządzenia

Algorytm pracy urządzenia rozpoczyna się sprawdzeniem stanu konfiguracji. Jeżeli urządzenie nie jest skonfigurowane, uruchamia się konfiguracja poprzez WiFi, w przeciwnym razie lokalizator przechodzi do procesu podłączenia do sieci i serwera MQTT.



Rysunek 5.9: Diagram pracy urządzenia

Proces podłączenia do sieci i serwera MQTT – w tym etapie urządzenie sprawdza poprawność konfiguracji poprzez możliwość podłączenie do brokera MQTT. Jeżeli łączność została nawiązana, algorytm wraca do miejsca wywołania procesu, w przeciwnym razie proces wywoływany jest ponownie, a licznik zostaje zinkrementowany. Gdy rekonfiguracja następuje więcej niż pięć razy, urządzenie wyłącza się i włącza ponownie, rozpoczynając pracę od początku.

Gdy proces zakończy się pomyślnie, urządzenie sprawdza aktualną lokalizację. Jeżeli uda się ustalić współrzędne geograficzne zostaje wysłany pakiet danych z informacją o lokalizacji. Następnie moduł sprawdza, czy otrzymano potwierdzenie odbioru. Jeżeli wszystko poszło dobrze, przechodzi do następnego etapu. Jeżeli nie, ponownie wywoływany jest proces podłączenia do sieci i serwera MQTT, a następnie wykonywana jest kolejna próba wysłania pakietu. W przypadku gdy nie uda się ustalić aktualnej pozycji, urządzenie wysyła jedynie pakiet z informacją o stanie baterii i przechodzi do dalszego etapu. W kolejnym etapie lokalizator oczekuje na komunikację SMS przez ustalony czas, w przypadku gdy ta nastąpi, wywoływany jest proces sprawdzenia komendy i wykonania odpowiedniego zadania.

Proces sprawdzania komendy i wykonania odpowiedniego zadania – w tym etapie urządzenie sprawdza zawartość otrzymanej wiadomości tekstowej i porównuje ją ze znanymi komendami. W pierwszej kolejności moduł sprawdza, czy komenda jest żądaniem informacji. Jeżeli tak, wysłana zostaje wiadomość tekstowa ze wszystkimi aktualnymi informacjami i następuje powrót do miejsca wywołania procesu. W przeciwnym razie moduł przechodzi do kolejnego sprawdzenia, tym razem czy jest to komenda zmiany okresu. Jeżeli tak, nowa wartość jest sprawdzana i przypisywana, po czym następuje powrót do miejsca wywołania procesu. Jeżeli nie, moduł przechodzi do kolejnych sprawdzeń – czy zawartość jest komendą resetu, bądź restartu urządzenia. Jeżeli tak, moduł wywołuje odpowiednio restart urządzenia lub przywrócenie do ustawień fabrycznych. Jeżeli zawartość nie zostaje zinterpretowana jako żadna komenda, następuje powrót do miejsca wywołania procesu.

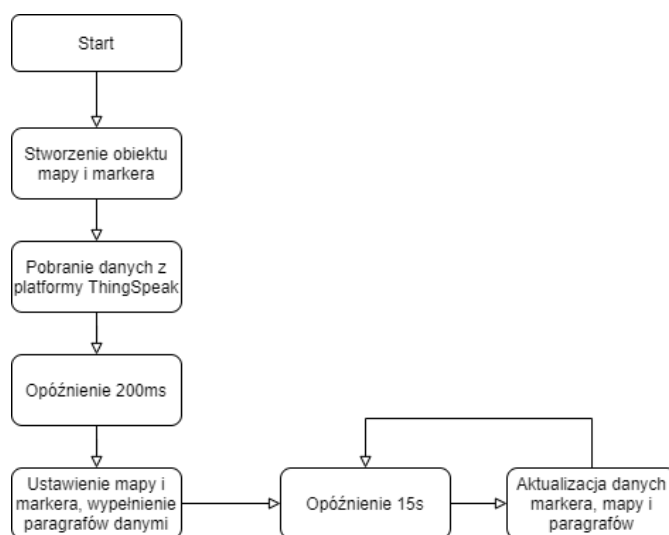
Po czasie oczekiwania na komunikację lub sprawdzeniu komendy wiadomości tekstowej następuje powrót do ustalenia lokalizacji, a algorytm zostaje zapętlony.

5.10 Konfiguracja

Proces konfiguracji urządzenia został zrealizowany w oparciu o uruchomiony na urządzeniu punkt dostępowy i wykorzystanie internetowego serwera asynchronicznego. Przygotowany system konfiguracji zawiera trzy witryny, które realizują kolejne etapy procesu. W pierwszym kroku, przy połączeniu użytkownika do głównej ścieżki serwera, zostaje wysłana witryna z informacją powitalną. Po naciśnięciu przez użytkownika odnośnika prowadzącego do kolejnego etapu urządzenie wysyła właściwą witrynę zawierającą formularz z wymaganymi danymi. Po uzupełnieniu i naciśnięciu przycisku następuje walidacja danych po stronie przeglądarki użytkownika. Jeżeli wprowadzone informacje są błędne, użytkownik zostanie poinformowany o tym fakcie i poproszony o ich ponowne wprowadzenie. Jeżeli wprowadzone informacje są poprawne, formularz z danymi jest wysyłany do urządzenia, a użytkownik jest kierowany do ostatniej witryny, gdzie dostaje informację o poprawnie ukończonym procesie konfiguracji. W tym czasie urządzenie zapisuje odebrane dane konfiguracyjne do pamięci, a następnie uruchamia się ponownie w trybie normalnej pracy.

5.11 Witryna internetowa

Witryna internetowa do odczytywania danych zebranych przez urządzenie została zbudowana w oparciu język HTML (ang. *HyperText Markup Language*) w wersji 5. Stronę wykonano w sposób umożliwiający jak najprostszą obsługę. Witryna zawiera jedynie mapę administracyjną, umieszczoną w centralnej części i dwa akapity z informacją o dacie ustalenia lokalizacji oraz stanie poziomu baterii, zlokalizowane w kontenerze pod mapą. Do stylizacji strony użyto kaskadowych arkuszy stylów, dla przyjemniejszej interakcji serwisu z użytkownikiem użyto dodatkowych cieniowań oraz gradientu tła. Funkcjonalności witryny realizuje skrypt napisany w języku JavaScript z wykorzystaniem wtyczki Leaflet [6] do obsługi mapy i jQuery [5] do pobierania zewnętrznych danych.



Rysunek 5.10: Diagram pracy skryptu witryny

Skrypt pobiera dane poprzez API z chmury ThingSpeak, a następnie prezentuje lokalizację w postaci markera o odpowiednich współrzędnych geograficznych umieszczonego na mapie administracyjnej. Pozostałe dane umieszczane są we wcześniej przygotowanych akapitach. Skrypt uruchamia się cyklicznie, odświeżając dane i pozycję markera na mapie.

5.12 Wykorzystywane komendy AT

W projekcie wykorzystano wymienione niżej i krótko opisane komendy AT dla modułu AiThinker A9G. Spis wszystkich komend wraz z opisem jest dostępny w dokumentacji dostarczonej przez producenta [4].

- AT+CSCS=GSM – ustawienie kodowania GSM;
- AT+CMGF=1 – ustawienie trybu tekstowego;
- AT+CPMS – ustawienie miejsca przechowywania wiadomości SMS;
- AT+CMGD=1,4 – usunięcie wszystkich wiadomości SMS;
- AT+CGATT=1 – włączenie GPRS;
- AT+CIPMUX=0 – tryb pracy dla jednego połączenia;
- AT+CSTT – ustawienie APN (ang. *Access Point Name*);
- AT+CHCR – rozpoczęcie połączenia GPRS;
- AT+CIFSR – wyświetlenie adresu IP;
- AT+CGACT=1,1 – aktywacja kontekstu PDP (ang. *Packet Data Protocol*);
- AT+MQTTDISCONN – rozłączenie połączenia z serwerem MQTT;
- AT+MQTTCONN – połączenie z serwerem MQTT;
- AT+MQTTPUB – wysłanie pakietu MQTT;
- AT+CBC? – zapytanie o stan baterii;
- AT+CCLK? – zapytanie o aktualną datę i godzinę;
- AT+LOCATION=2 – zapytanie o aktualną lokalizację;
- AT+GPS=1 – włączenie modułu GPS;
- AT+GPSLP=1 – włączenie trybu niskiego zużycia energii dla modułu GPS;
- AT+SLEEP – obsługa trybu niskiego zużycia energii.

5.13 Integracja urządzenia z witryną internetową

Wymiana danych, zebranych przez lokalizator, z witryną internetową odbywa się poprzez broker MQTT zlokalizowany w chmurze ThingSpeak. W brokerze dodano kanał dla urządzenia, a następnie odpowiednio go skonfigurowano, dodając również cztery pola o ustalonym identyfikatorze przeznaczone na dane, kolejno:

- Lat – dla szerokości geograficznej;
- Long – dla długości geograficznej;
- Data – dla daty określenia lokalizacji;
- Battery – dla wskazania poziomu baterii.

Kanał ustawiono jako prywatny i wygenerowano klucze API dla procesów odczytu i zapisu danych kanału. Urządzenie działa jako wydawca (ang. *publisher*) i wysyła pakiety MQTT do brokera, który zapisuje i przechowuje zebrane dane. Dane wysyłane są z wykorzystaniem pierwszego trybu QoS (ang. *Quality of Service*), który wymaga wysłania potwierdzenia odbioru danych przez odbiorcę (broker). Najnowsze dane można pobrać z chmury wykorzystując oferowane przez ThingSpeak API odczytu danych, a następnie wykorzystać w dowolny sposób.

Rozdział 6

Weryfikacja i walidacja

W celu weryfikacji oprogramowania przeprowadzono szereg testów poszczególnych scenariuszy użytkownika. Każda pojedyncza funkcjonalność była testowana jednostkowo przed dodaniem do właściwego oprogramowania, a następnie przeprowadzono testy integracyjne funkcjonalności z pozostałymi modułami. Testy przeprowadzono głównie samodzielnie, jednak dla potrzeby szerszych danych, urządzenie udostępniono do przetestowania testerom. Testerzy mieli za zadanie przeprowadzić proces konfiguracji oraz wypróbować zarządzanie lokalizatorem. Pierwszy tester w średnim wieku jest obeznany z aktualną technologią, na co dzień posługuje się smartfonem i w pełni korzysta z jego możliwości. Drugi tester jest w podeszłym wieku, a jego stopień obeznania z technologią można uznać za niski, natomiast na co dzień posługuje się smartfonem i potrafi korzystać z zainstalowanej na nim przeglądarki internetowej, a także rozumie pojęcia takie jak: aplikacja, ekran główny, czy pole adresowe.

6.1 Testowanie konfiguracji

Urządzenie było wielokrotnie konfigurowane w celu sprawdzenia występowania możliwych błędów. Przetestowano rozłączenie się z siecią w trakcie procesu, konfigurację na kilku urządzeniach jednocześnie, odłączenie zasilania podczas konfiguracji oraz konfigurację z użyciem błędnych danych.

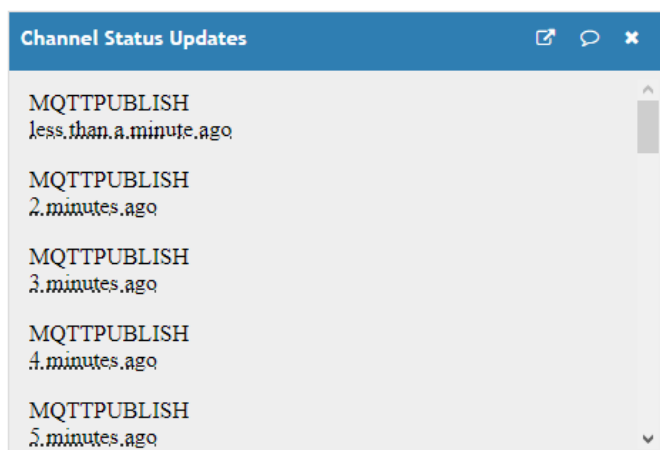
Dla pierwszego testera konfiguracja nie sprawiła żadnych kłopotów, przebiegła szybko i sprawnie. Dla drugiego testera proces konfiguracji był nieco utrudniony z powodu rozłączenia z siecią WiFi w trakcie wprowadzania danych. Tester przypadkowo nacisnął przycisk alertu na smartfonie odpowiadający za przełączenie sieci na tę posiadającą dostęp do Internetu, co spowodowało zerwanie połączenia z lokalizatorem. Po ponownym podłączeniu urządzenia do sieci lokalizatora i rozpoczęciu procesu od nowa konfiguracja odbyła się bez żadnych przeszkód.

6.2 Testowanie zarządzania

Przez zastosowanie krótkich prostych komend, proces zarządzania został mocno uproszczony i ograniczony do minimum. Testerzy nie mieli żadnych problemów z ustawieniem częstotliwości lokalizacji czy restartem urządzenia. Moduł każdorazowo odpowiadał na wysyłane żądania danych informacyjnych. Po przeprowadzeniu pomyślnie zakończonych właściwych testów, przy próbie uzyskania informacji o lokalizacji, moduł nagle przestał odpowiadać na jakiekolwiek komendy. Źródłem problemu okazała się zapełniona pamięć przeznaczona na przechowywanie wiadomości tekstowych. Problem udało się rozwiązać i opisano go w sekcji "Wykryte i usunięte błędy".

6.3 Testy niezawodności

Test niezawodności polegał na sprawdzeniu nieprzerwanej pracy urządzenia przez pięć dni. Urządzenie zainstalowano w często użytkowanym pojeździe i przynajmniej dwa razy dziennie monitorowano poprawność jego pracy poprzez sprawdzenie lokalizacji na żądanie za pomocą wiadomości tekstowej. Dodatkowo sprawdzano również częstotliwość odbierania danych przez broker MQTT oraz sprawdzano ostatnią lokalizację na stronie internetowej. Urządzenie przeszło pomyślnie test. Na koniec testu odłączono zewnętrzne zasilanie i sprawdzono, czy lokalizator działa na zasilaniu awaryjnym. Ten test również zakończono wynikiem pozytywnym.



Rysunek 6.1: Częstotliwość odbierania danych przez broker

6.4 Testowanie dokładności

Testy przeprowadzono, porównując lokalizację ustaloną przez urządzenie do lokalizacji ustalonej przez aplikację Mapy Google na smartfonie. W otwartym terenie miejskim urządzenie uzyskuje bardzo dobrą dokładność – około 1 m. Problemy z poprawnym i szybkim określeniem lokalizacji występują w parkingach podziemnych oraz tunelach. Problem częściowo rozwiązano, przechowując ostatnio ustaloną poprawną lokalizację w pamięci urządzenia.

6.5 Testy pracy na zasilaniu awaryjnym

Przeprowadzono testy czasu pracy urządzenia na zasilaniu awaryjnym. Testy przeprowadzono dla dwóch trybów oszczędzania energii modułu GPS. Każdorazowo ładowano urządzenie do poziomu 100% naładowania baterii, a następnie odłączano zewnętrzne źródło zasilania. W pierwszym etapie przeprowadzono po dwa testy dla każdego trybu oszczędzania energii. Dla pierwszego trybu uzyskane czasy pracy wyniosły 20 i 19.5 godzin. Dla drugiego trybu uzyskane czasy pracy wyniosły 13 i 16 godzin. Zdecydowano się na wybór pierwszego trybu oszczędzania energii i przeprowadzono trzy kolejne testy. Uzyskane czasy to: 18, 21 i 20.5 godzin pracy na zasilaniu awaryjnym.

Dodatkowo określono również średni czas raportowania urządzenia o niskim stanie naładowania baterii, który wyniósł około 4 godzin przed końcem pracy urządzenia, wynikającym z rozładowania akumulatora.

6.6 Testy odporności na upadki

Zamknięte w obudowie urządzenie upuszczano z wysokości około półtora metra. Przeprowadzono trzy próby.

Po pierwszym upadku klapka obudowy otworzyła się, a płytką PCB (ang. *Printed Circuit Board*) delikatnie podniosła się. Urządzenie w dalszym ciągu funkcjonowało poprawnie i reagowało na wysyłane komendy. Zdecydowano o dodatkowym podklejeniu płytki PCB do elementów montażowych (wcześniej płytka była zamontowana "na wcisk").

Podczas drugiego upadku urządzenie spadło klapką obudowy na ziemię, po oględzinach nie stwierdzono żadnych uszkodzeń lub przemieszczeń elementów. Urządzenie pracowało poprawnie i reagowało na wysyłane komendy.

Przy trzecim upadku urządzenie pozostawiono podłączone do zasilania, po upadku urządzenie zrestartowało się. Po oględzinach w tym przypadku również nie stwierdzono żadnych uszkodzeń lub przemieszczeń elementów. Urządzenie w dalszym ciągu pracowało poprawnie i reagowało na wysyłane komendy.

6.7 Wykryte i usunięte błędy

W początkowym etapie przygotowywania oprogramowania dla urządzenia występował problem z komunikacją między modulem zarządzającym a modulem GSM/GPS. Zarówno dane wysyłane, jak i odbierane przez mikrokontroler, były bardzo często przekłamywane, co powodowało nieprawidłową i niestabilną pracę urządzenia. Źródło problemu zlokalizowano w bibliotece używanej do stworzenia programowego portu szeregowego. Problem udało się rozwiązać, obniżając szybkość transmisji oraz zwiększając pojemność buforów na wiadomości przychodzące i wychodzące.

Kolejnym napotkanym problemem był błąd podczas próby połączenia do brokera MQTT. Po analizie problemu ustalono, że problem występuje przy jakiegokolwiek próbie połączenia z Internetem. Źródłem problemu okazała się nieprawidłowa konfiguracja usługi GPRS u operatora. Po prawidłowej konfiguracji usługi problem przestał występować, a urządzenie każdorazowo łączyło się z brokerem.

Następny problem został zauważony już po pomyślnym przejściu właściwych testów. Urządzenie nagle przestało reagować na wszelkie próby wysyłania komend tekstowych. Po podłączeniu urządzenia do konsoli PlatformIO IDE i prześledzeniu jego pracy zdiagnozowano źródło problemu, którym okazała się pełna pamięć do przechowywania wiadomości tekstowych. Po usunięciu przechowywanych wiadomości problem ustał. W celu rozwiązania problemu zaimplementowano cykliczne czyszczenie pamięci przeznaczonej na przechowywanie wiadomości tekstowych.

Ostatni napotkany problem został zauważony podczas ekstremalnych testów urządzenia w ruchu. Lokalizator był luźno umieszczony w plecaku testera przemieszczającego się po leśnej trasie rowerowej z licznymi wzniesieniami. Podczas testu urządzenie nagle przestało raportować lokalizację i reagować na wysyłane polecenia. Po otwarciu obudowy i szybkiej analizie zlokalizowano źródło problemu. Powodem brakiem funkcjonowania lokalizatora okazał się poluzowany styk akumulatora w koszyczku. Problem udało się rozwiązać, odpowiednio regulując docisk ogniwa akumulatorowego.

Rozdział 7

Podsumowanie i wnioski

7.1 Podsumowanie

Wykonane urządzenie jest w pełni zgodne z założeniami i realizuje postawione wymagania dotyczące funkcjonalności. Uzyskane wyniki pracy są satysfakcjonujące. W kombinacji z wykonaną witryną internetową i wykorzystaniem platformy ThingSpeak projekt składa się na pełny system lokalizacji pojazdu mający wiele praktycznych zastosowań. Użytkownik może śledzić zmiany lokalizacji pojazdu używając wykonanej aplikacji webowej lub wysyłając odpowiednie żądanie. Przeprowadzone długotrwałe testy pozwalają stwierdzić, iż urządzenie jest w pełni funkcjonalne i sprawdza się w codziennym użytkowaniu. Podsumowując, cel niniejszej pracy został zrealizowany. W przyszłości projekt może zostać zmodyfikowany i rozbudowany o dodatkowe funkcjonalności.

7.2 Wnioski

Podczas procesu testowania urządzenia zdiagnozowano szereg niezamierzonych działań lub nieprawidłowości opisanych w poprzednim rozdziale. Wczesne wykrycie błędów pozwoliło na udoskonalenie pracy urządzenia i poprawienie finalnego projektu. Pracę, w miarę możliwości, starano się wykonać przy jak najniższym budżecie, stąd decyzja o wykorzystaniu gotowego układu AiThinker ESP12S A9G GPRS Node v1.0 zamówionego z popularnego chińskiego serwisu sprzedażowego.

Biorąc pod uwagę całkowity koszt elementów potrzebnych do zbudowania urządzenia, lokalizator sprawuje się nad wyraz dobrze, a efekty jego pracy są miarodajne.

7.3 Możliwe kierunki rozwoju

Projekt można rozbudować w wielu kierunkach, zarówno rozbudowując samo urządzenie, jak i system prezentacji czy integracji danych. Poniżej przedstawiono kilka możliwości rozwoju:

- Dodanie bezpiecznej strefy dla pojazdu, po przekroczeniu której urządzenie będzie raportowało alertem w formie wiadomości tekstowej;
- Detekcja utraty zasilania głównego raportowana alertem w formie wiadomości tekstowej;
- Wzbogacenie urządzenia o akcelerometr i kolejne funkcjonalności wraz z systemem powiadomień;
- Możliwość zdalnego odcięcia zapłonu pojazdu;
- Dostęp do szerszych informacji o pojeździe poprzez komunikację urządzenia z komputerem pokładowym pojazdu poprzez magistralę CAN;
- Odtworzenie historii lokalizacji pojazdu za pomocą witryny internetowej;
- Stworzenie większego systemu prezentacji i przechowywania danych z możliwością podłączenia kilku lokalizatorów dla jednego użytkownika;
- Stworzenie własnego brokera MQTT oraz bazy danych, rezygnacja z usług platformy ThingSpeak.

Bibliografia

- [1] Dokumentacja - a9g gprs/gsm+gps/bds module. https://ai-thinker-open.github.io/GPRS_C_SDK_DOC/en/hardware/a9g.html, 2017. [data dostępu: 2020-10-01].
- [2] Dokumentacja - esp12s. https://www.exp-tech.de/media/pdf/1c/99/0a/a020ps01a0_esp-12s_product_specification.pdf, 2018. [data dostępu: 2020-10-01].
- [3] Czy twój samochód jest bezpieczny? sprawdź, gdzie kradnie się najwięcej aut w polsce - natemat.pl. <https://natemat.pl/276457,gdzie-kradna-najczesciej-samochody-zobacz-zestawienie-statystyk-policji>, 2019. [data dostępu: 2020-12-06].
- [4] Dokumentacja - aithinker esp12s a9g gprs node v1.0. <https://github.com/IOT-MCU/ESP-12S-A9-A9G-GPRS-Node-v1.0>, 2019. [data dostępu: 2020-10-01].
- [5] Dokumentacja - jquery. <https://api.jquery.com>, 2020. [data dostępu: 2020-11-25].
- [6] Dokumentacja - leaflet. <https://leafletjs.com/reference-1.7.1.html>, 2020. [data dostępu: 2020-11-25].
- [7] Liczba samochodów w polsce – ile aut jeździ po polskich drogach? - smartdriver.pl. <https://www.smartdriver.pl/liczba-samochodow-w-polsce-ile-aut-jezdzi-po-polskich-drogach>, 2020. [data dostępu: 2020-12-06].

- [8] Raport 2019 utracone pojazdy - statystyka.policja.pl. <https://statystyka.policja.pl/st/wybrane-statystyki/kradzieze-samochodow/190273,Raport-2019-utracone-pojazdy.html>, 2020. [data dostępu: 2020-12-06].
- [9] Mateusz Olewiński, Tomasz Perzyński, Daniel Pietruszczak. Wybrane zabezpieczenia pojazdów samochodowych przed kradzieżą. *Autobusy – Technika, Eksploatacja, Systemy Transportowe*, 226(12):168–177, 2018.
- [10] Senkus Piotr, Skrzypek Adam, Łuczak Miłosz, Malinowski Artur. Internet of things: przeszłość – teraźniejszość – przyszłość. *Zeszyty Naukowe Uniwersytetu Przyrodniczo-Humanistycznego w Siedlcach, Seria: Administracja i Zarządzanie*, 103(30):163–172, 2014.
- [11] Piotr Smejda. Internet rzeczy (iot) we współczesnej gospodarce. rola, zadania i bariery rozwoju. *Zeszyty Naukowe, Organizacja i Zarządzanie*, 1208(64):43–55, 2016.
- [12] Krzysztof Tokarz, Jarosław Paduch, Łukasz Herb. Influence of receiver parameters on gps navigation accuracy. *Man-Machine Interactions 2*, 103:85–93, 2011.
- [13] Stanisław Wieteska. Kradzieże samochodów w polsce jako element ryzyka w ubezpieczeniach auto casco. *Annales. Etyka w życiu gospodarczym*, 2(18):115–126, 2015.

Spis skrótów i symboli

API zbiór reguł opisujących sposób komunikacji między programami lub podprogramami (ang. *Application Programming Interface*)

APN adres lub nazwa punktu dostępu między siecią komórkową operatora a zewnętrzną siecią (ang. *Access Point Name*)

Broker pełni rolę serwera, z którym łączą się urządzenia MQTT

CAN szeregową różnicową magistrala komunikacyjna stosowana w przemyśle (ang. *Controller Area Network*)

Git rozproszony system kontroli wersji

GitHub platforma umożliwiająca obsługę systemu kontroli wersji Git

GNU GPL licencja dla wolnego i otwartego oprogramowania

GPS amerykański system nawigacji satelitarnej (ang. *Global Positioning System*)

GPRS technika pakietowego przesyłania danych w sieciach GSM (ang. *General Packet Radio Service*)

HTML hipertekstowy język znaczników, wykorzystywany do tworzenia stron internetowych (ang. *HyperText Markup Language*)

HTTP protokół do przesyłania hipertekstowych dokumentów (ang. *Hypertext Transfer Protocol*)

IDE zintegrowane środowisko programistyczne (ang. *Integrated Development Environment*)

I2C dwukierunkowa szeregową magistrala służąca do przesyłania danych w urządzeniach elektronicznych (ang. *Inter-Integrated Circuit*)

IoT popularna koncepcja, według której jednoznacznie identyfikowalne przedmioty mogą wymieniać, gromadzić lub przetwarzać dane między sobą (ang. *Internet of Things*)

MathWorks amerykańska korporacja specjalizująca się w tworzeniu oprogramowania do obliczeń naukowych i inżynierskich

MQTT lekki protokół do przesyłania danych pakietowych (ang. *Message Queue Telemetry Transport*)

OpenStreetMap projekt mający na celu stworzenie darmowej mapy całej kuli ziemskiej

PDP struktura danych zawierająca informację o sesji abonenta w GPRS (ang. *Packet Data Protocol*)

PLA polilaktyd – biodegradowalny polimer, stosowany jako filament do druku przestrzennego (ang. *Polylactic Acid*)

PCB płytki obwodu drukowanego Printed (ang. *Printed Circuit Board*)

QoS całość charakterystyk potrzebnych do spełnienia wyrażonych potrzeb użytkownika usługi telekomunikacyjnej (ang. *Quality of Service*)

SMS usługa wysyłania wiadomości tekstowych w cyfrowych sieciach telefonii komórkowej (ang. *Short Message Service*)

SoC układ scalony zawierający kompletny system elektroniczny (ang. *System-on-a-chip*)

SPIFFS prosty lekki system plików zapisywanych w wydzielonym obszarze pamięci Flash (ang. *Serial Peripheral Interface Flash File System*)

Styrodur polistyren ekstrudowany, materiał służący do ocieplania powierzchni

Transceiver urządzenie elektroniczne pełniące funkcję nadajnika i odbiornika

UART układ scalony służący do asynchronicznej wymiany informacji poprzez port szeregowy (ang. *Universal Asynchronous Receiver-Transmitter*)

USB komputerowe złącze komunikacyjne (ang. *Universal Serial Bus*)

WebSocket protokół zapewniający pełny dwukierunkowy kanał komunikacji za pomocą jednego gniazda TCP

WiFi standardy stworzone do budowy komputerowych sieci bezprzewodowych

WPA2 standard szyfrowania stosowany w sieciach bezprzewodowych (ang. *Wi-Fi Protected Access II*)

Spis rysunków

2.1	Schemat blokowy urządzenia	6
3.1	Diagram przypadków użycia	12
3.2	Diagram przypadków użycia witryny internetowej	13
4.1	Lokalizator w zamkniętej obudowie podłączony do zasilania	16
4.2	Lokalizator z otwartą górną pokrywą podłączony do zasilania	19
4.3	Utworzony punkt dostępowy	20
4.4	Ekran powitalny kreatora konfiguracji	21
4.5	Formularz do uzupełnienia wymaganych danych	22
4.6	Zakończenie procesu konfiguracji	23
4.7	Informacje uzyskane za pomocą komendy "Info"	24
4.8	Informacja uzyskane za pomocą witryny internetowej	25
5.1	Architektura systemu	29
5.2	Moduł AiThinker ESP12S A9G GPRS Node v1.0	31
5.3	Ogniwo akumulatorowe Samsung INR18650-35E	32
5.4	Schemat blokowy i opis urządzenia	33
5.5	Trójwymiarowy model obudowy i górnej pokrywy	34
5.6	Rozmieszczenie elementów w obudowie – widok z góry	35
5.7	Rozmieszczenie elementów w obudowie	35
5.8	Diagram klas urządzenia	38
5.9	Diagram pracy urządzenia	41
5.10	Diagram pracy skryptu witryny	44
6.1	Częstotliwość odbierania danych przez broker	49

Zawartość dołączonej płyty

Do pracy dołączona jest płyta CD z następującą zawartością:

- praca (źródła \LaTeX owe i końcowa wersja w `pdf`),
- źródła programu,
- pliki trójwymiarowego modelu obudowy urządzenia (`stl`),
- witryny internetowe do konfiguracji urządzenia i prezentacji danych.