



POLITECHNIKA WARSZAWSKA

Wydział Mechatroniki

Praca magisterska

Konrad Traczyk

Projekt urządzenia do lokalizacji pojazdów  
w trybie on i off-line

Opiekun pracy:  
prof. dr hab. Michał Bartysi

Warszawa, 2017

# Spis treści

<b>Spis treści</b>	<b>2</b>
<b>Spis rysunków</b>	<b>4</b>
<b>Spis tabel</b>	<b>6</b>
<b>1 Wstęp</b>	<b>8</b>
1.1 Zakres pracy . . . . .	8
1.2 Schemat blokowy urządzeń . . . . .	10
1.3 Istniejące rozwiązania . . . . .	11
<b>2 Wstęp teoretyczny</b>	<b>13</b>
2.1 Zastosowane protokoły i systemy . . . . .	13
2.2 System GSM . . . . .	14
2.3 System GPS . . . . .	17
2.4 Protokół NMEA 0183 . . . . .	22
2.5 Protokół Bluetooth Low Energy . . . . .	26
2.6 Interfejs NFC . . . . .	29
2.7 Podsumowanie . . . . .	34
<b>3 Schematy elektroniczne urządzeń</b>	<b>35</b>
3.1 Urządzenie lokalizujące . . . . .	35
3.1.1 Schemat zasilania . . . . .	38
3.1.2 Moduł mikrokontrolera . . . . .	41
3.1.3 Moduł GSM i GPS . . . . .	42
3.1.4 Moduł pamięci flash . . . . .	44
3.1.5 Moduł akcelerometru . . . . .	45
3.1.6 Moduł NFC . . . . .	46
3.2 Urządzenie deaktywujące . . . . .	48

<b>4 Schematy płytEK drukowanych</b>	<b>50</b>
4.1 Urządzenie deaktywujące . . . . .	50
4.2 Urządzenie lokalizujące . . . . .	52
<b>5 Bezpieczeństwo komunikacji</b>	<b>57</b>
5.1 AES . . . . .	58
5.2 Dodatkowe warianty szyfrowania AES . . . . .	62
5.3 Realizacja szyfrowania komunikacji w projekcie . . . . .	63
<b>6 Oprogramowanie</b>	<b>64</b>
6.1 Urządzenie lokalizujące . . . . .	64
6.2 Urządzenie deaktywujące . . . . .	71
6.3 Aplikacja serwerowa . . . . .	74
6.4 Strona internetowa . . . . .	77
<b>7 Analiza stylu jazdy</b>	<b>82</b>
<b>8 Podsumowanie</b>	<b>83</b>
<b>Bibliografia</b>	<b>84</b>
<b>Wykaz skrótów</b>	<b>86</b>

# Spis rysunków

1.1	Schemat blokowy urządzeń wchodzących w skład systemu. Źródło: Twórczość własna . . . . .	10
1.2	Spark Nano 5.0 GPS Tracker. Źródło: [1] . . . . .	11
1.3	Aplikacja MyCarTracks. Źródło: [2] . . . . .	12
1.4	Urządzenie STI GL300. Źródło: [3] . . . . .	12
2.1	Podział pasma częstotliwości na kanały i okna czasowe. Źródło: [4] . . . . .	15
2.2	Podział obszaru na komórki. Źródło: [6] . . . . .	16
2.3	Model działania systemu GPS. Źródło: [7] . . . . .	18
2.4	Zbiór częstotliwości wykorzystywanych w systemie GPS. Źródło: [8] . . . . .	19
2.5	Zasada działania systemu GPS. Źródło: [8] . . . . .	20
2.6	Zasada działania systemu GPS - sygnał referencyjny. Źródło: [8] . . . . .	21
2.7	Poziomy napięć i kolejność bitów w interfejsach RS232 i RS422. Źródło: [9] . . . . .	23
2.8	Struktura pasma 2.4 ISM wykorzystywanego przez Bluetooth Low Energy. Źródło: [8] . . . . .	28
2.9	Model komunikacji rozgłoszeniowej. Źródło: [8] . . . . .	28
2.10	Model struktury danych serwera GATT. Źródło: [8] . . . . .	29
2.11	Budowa tag'a RFID. Źródło: [13] . . . . .	30
2.12	Zasada działania komunikacji pomiędzy urządzeniem aktywnym i pasywnym. Źródło: [14] . . . . .	31
3.1	Schemat modułu zasilania urządzenia lokalizującego. Źródło: Twórczość własna . . . . .	36
3.2	Schemat modułu funkcjonalnego urządzenia lokalizującego. Źródło: Twórczość własna . . . . .	37
3.3	Schemat modułu NFC urządzenia lokalizującego. Źródło: Twórczość własna . . . . .	38
3.4	Schemat modułu zasilania wejściowego urządzenia lokalizującego. Źródło: Twórczość własna . . . . .	39

3.5 Schemat przetwornicy impulsowej modułu zasilania urządzenia lokalizującego. Źródło: Twórczość własna . . . . .	40
3.6 Schemat stabilizatora napięcia modułu zasilania urządzenia lokalizującego. Źródło: Twórczość własna . . . . .	41
3.7 Schemat modułu mikrokontrolera w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	42
3.8 Schemat modułu układu GSM i GPS w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	43
3.9 Schemat modułu anten dla GSM i GPS w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	44
3.10 Schemat modułu karty SIM w urządzeniu lokalizującym. Źródło: Twórczość własna	44
3.11 Schemat modułu pamięci flash w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	45
3.12 Schemat modułu akcelerometru w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	46
3.13 Schemat części cyfrowej modułu NFC w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	47
3.14 Schemat części analogowej modułu NFC w urządzeniu lokalizującym. Źródło: Twórczość własna . . . . .	47
3.15 Schemat modułu zasilania urządzenia deaktywującego. Źródło: Twórczość własna	49
4.1 Wygląd górnej warstwy płytki urządzenia dezaktywującego oraz jej wizualizacja. Źródło: Twórczość własna . . . . .	51
4.2 Wygląd dolnej warstwy płytki urządzenia dezaktywującego oraz jej wizualizacja. Źródło: Twórczość własna . . . . .	51
4.3 Tabela opisująca korelację między grubością ścieżek, a maksymalnym dopuszczalnym natężeniem prądu. Źródło: [18] . . . . .	54
4.4 Wygląd górnej warstwy płytki urządzenia lokalizującego oraz jej wizualizacja. Źródło: Twórczość własna . . . . .	55
4.5 Wygląd dolnej warstwy płytki urządzenia lokalizującego oraz jej wizualizacja. Źródło: Twórczość własna . . . . .	56
5.1 Wykonanie operacji Substitute Bytes. Źródło: [20] . . . . .	59
5.2 Wykonanie operacji Shift Rows. Źródło: [20] . . . . .	59
5.3 Wykonanie operacji Mix Columns. Źródło: [20] . . . . .	60
5.4 Wykonanie operacji Add Round Key. Źródło: [20] . . . . .	60
5.5 Operacja szyfrowania metodą ECB. Źródło: [19] . . . . .	61

5.6	Operacja deszyfrowania metodą ECB. Źródło: [19] . . . . .	61
5.7	Operacja szyfrowania metodą CFB. Źródło: [19] . . . . .	62
5.8	Operacja deszyfrowania metodą CFB. Źródło: [19] . . . . .	62
6.1	Główny algorytm działania urządzenia. Źródło: Twórczość własna . . . . .	67
6.2	Przepływ sterowania w przypadku wykrycia ruchu pojazdu. Źródło: Twórczość własna . . . . .	70
6.3	Przepływ sterowania w trakcie parowania urządzenia lokalizującego z urządzeniem deaktywującym. Źródło: Twórczość własna . . . . .	71
6.4	Przepływ sterowania w momencie deaktywacji alarmu. Źródło: Twórczość własna . . . . .	73
6.5	Schemat relacji między tabelami w bazie danych. Źródło: Twórczość własna . . . . .	75
6.6	Strona logowania. Źródło: Twórczość własna . . . . .	78
6.7	Ekran rejestracji użytkownia. Źródło: Twórczość własna . . . . .	78
6.8	Strona główna. Źródło: Twórczość własna . . . . .	79
6.9	Okno trasy. Źródło: Twórczość własna . . . . .	79

# Spis tabel

2.1	Najczęściej wykorzystywane wiadomości NMEA0183 w odbiornikach GPS. Źródło: [8] . . . . .	24
2.2	Struktura wiadomości GGA. Źródło: Twórczość własna . . . . .	25
2.3	Struktura wiadomości VTG. Źródło: Twórczość własna . . . . .	26
2.4	Podsumowanie cech systemów i protokołów GSM, GPS, BLE oraz NFC. Źródło: Twórczość własna . . . . .	34

# Rozdział 1

## Wstęp

### 1.1 Zakres pracy

Celem pracy było zaprojektowanie, wykonanie i oprogramowanie urządzenia stanowiącego dodatkowe zabezpieczenie antykradzieżowe pojazdu, w postaci lokalizatora wykorzystującego system GNSS (*ang. Global Navigation Satellite System*) oraz GSM (*ang. Global System for Mobile Communications*), zdolnego do analizy stylu jazdy kierowcy, a także całego systemu informatycznego, który pozwoliłby na obsłużenie pozyskanych danych. W jego skład wchodzą:

- Strona WWW, umożliwiająca zdalny podgląd danych z przypisanych do użytkownika urządzeń.
- Aplikacja serwerowa, która obsługuje zapytania użytkownika oraz zapisującą napływające dane do bazy danych SQLite.

Do dodatkowych wymagań stawianych urządzeniu należą:

- Zapewnienie bezpiecznej komunikacji dezaktywującej tryb alarmu.
- Posiadanie zastępczego źródła zasilania, umożliwiającego pracę przy wyłączonym silniku pojazdu, bądź w razie odłączenia akumulatora.
- Niewielkie wymiary urządzenia w celu umożliwienia łatwego ukrycia w pojeździe.

Moduł umożliwia działanie w dwóch trybach. Pierwszy z nich polega na cyklicznym wysyłaniu na serwer pozycji samochodu w trakcie ruchu wraz z m.in. jego prędkością i przyspieszeniem. Dzięki temu możliwy jest zdalny podgląd stylu jazdy kierowcy, co ułatwia sprawowanie kontroli nad flotą pojazdów. Ponadto, dane te zapisywane są również w pamięci nieulotnej urządzenia,

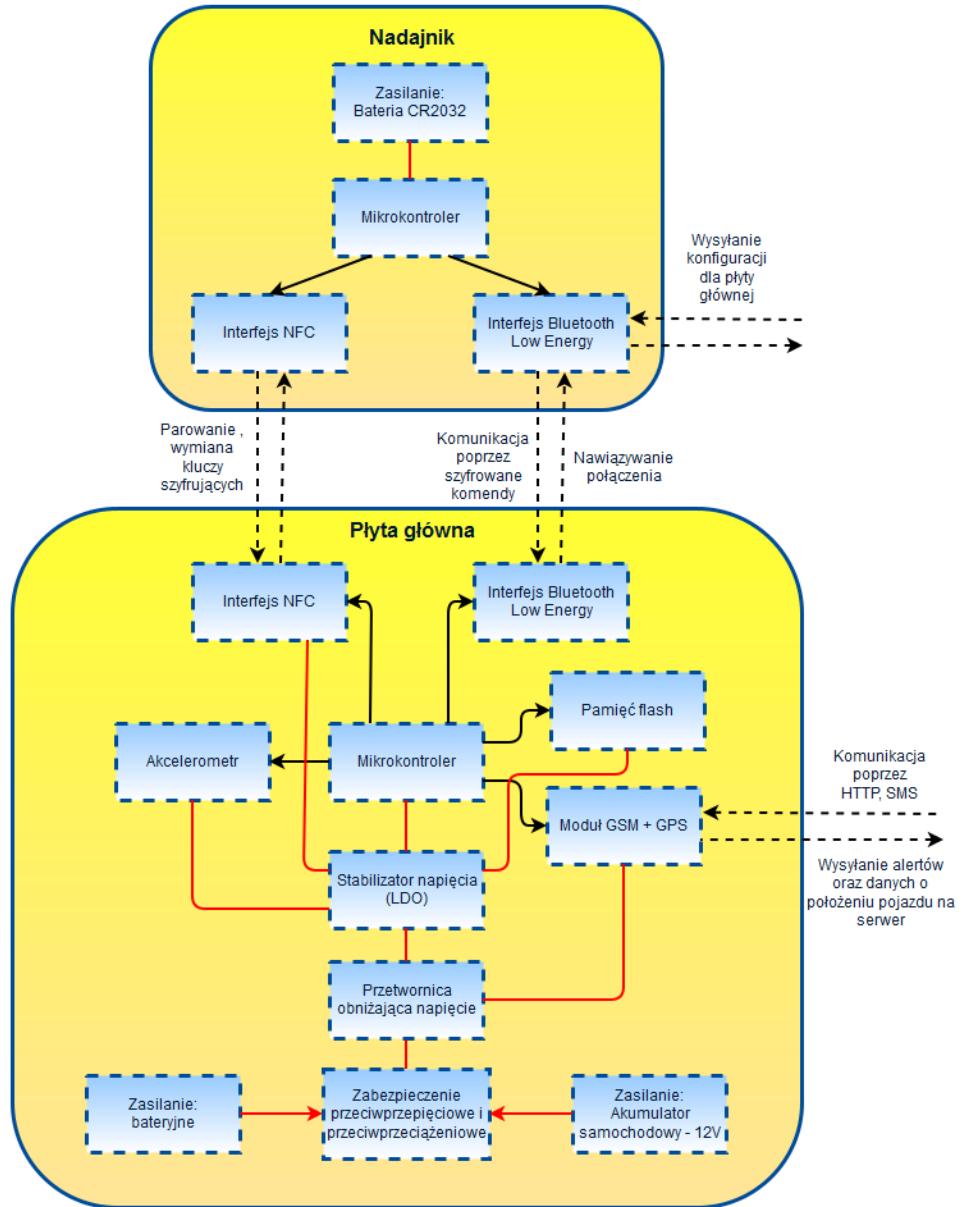
co pozwala na ograniczenie kosztów związanych z transmisją bezprzewodową i posiadaniem karty SIM od operatorów GSM.

Drugi tryb uaktywnia się w trakcie postoju i stanowi system alarmowego powiadamiania właściciela pojazdu o nieautoryzowanym jego przemieszczeniu w przypadku kradzieży.

W celu zapewnienia bezpieczeństwa, postanowiono rozbić projekt na dwa urządzenia. Jedno z nich – płytka lokalizatora, stanowi rdzeń systemu umożliwiający lokalizację pojazdu oraz wysyłanie danych na serwer. Drugi moduł stanowi układ deaktywujący, którego zadaniem jest wyłączenie trybu alarmu po odpaleniu samochodu przez upoważnioną do tego osobę. Obie płytki komunikują się ze sobą poprzez protokół Bluetooth Low Energy, zapewniający energooszczędną wymianę danych. Pozwoli to na zasilenie układu deaktywującego z niewielkiej baterii i jego nieprzerwaną pracę nawet przez kilka lat bez konieczności wymiany źródła zasilania. Ponadto, aby umożliwić bezpieczną transmisję niezbędne jest zastosowanie szyfrowania komunikacji. W celu eliminacji ryzyka podsłuchania procesu wymiany klucza szyfrującego, oba urządzenia zostały wyposażone w moduł NFC (*ang. Near Field Communication*), zapewniającego bezkontaktową komunikację na odległość do 10 cm.

## 1.2 Schemat blokowy urządzeń

Na przedstawionym poniżej rysunku 1.1 zaprezentowano schemat blokowy urządzeń, które stanowią główną część projektu - moduł płyty głównej oraz moduł dezaktywatora.



Rysunek 1.1: Schemat blokowy urządzeń wchodzących w skład systemu.

Źródło: Twórczość własna

## 1.3 Istniejące rozwiązania

W ramach pracy przeprowadzono analizę rynkową pod kątem istniejących, ciekawych rozwiązań. Poniżej zaprezentowano trzy najbardziej charakterystyczne z nich.

- Spark Nano 5.0 GPS Tracker

To przenośne urządzenie do śledzenia pozycji geograficznej przy pomocy systemu GPS posiada zasilane baterijnie. Zapewnia zdalne powiadomianie użytkownika o lokalizacji urządzenia poprzez sieć CDMA, z dokładnością do 2m. Wymiary urządzenia: 64,5 x 40 x 20,5 mm. Urządzenie pozwala na działanie przez ok. 2 tygodnie, przy założeniu pracy przez 1 godzinę dziennie. Producent udostępnia platformę online oraz aplikacje na smartfony z systemem Android oraz IOS, do przedstawiania danych użytkownikowi. Urządzenie domyślnie raportuje położenie co minutę, lecz producent umożliwia zdalne zwiększenie częstotliwości w razie chęci użytkownika. Cena urządzenia: 129,99\$. Wizualizację urządzenia przedstawiono na rysunku 1.2.



Rysunek 1.2: Spark Nano 5.0 GPS Tracker. Źródło: [1]

- MyCarTracks - aplikacja mobilna

Jest to aplikacja na smartphona, która dodaje do niego funkcjonalność trackera GPS. Stanowi rozwiązanie typowo programowe, które wykorzystuje zasoby zawarte w telefonie – moduł GPS, GSM oraz internet. Jest ono proste i tanie, lecz nie pozbawione wad. Ponieważ to aplikacja na telefon, a nie osobne urządzenie, konieczne jest umieszczenie smartphone'a w pojeździe na stałe, jeśli użytkownik chciałby użytkować ją jako zabezpieczenie antykradzieżowe. Ponadto, telefony pobierają stosunkowo dużo energii co wymusza częste ich ładowanie. W rezultacie efektywne ukrycie urządzenia jest utrudnione. Do kosztów rozwiązania należy wliczyć cenę telefonu (używane urządzenie kosztuje ok. 200-300zł) oraz 7\$ za każdy pojazd miesięcznie. Aplikację przedstawiono na rysunku 1.3.



Rysunek 1.3: Aplikacja MyCarTracks. Źródło: [2]

- STI GL300

Jest to kolejne niewielkie, przenośne urządzenie wykorzystujące moduł GPS do lokalizacji. Przekazuje ono informacje o położeniu w czasie rzeczywistym (co 60, 10 lub 5 sekund w zależności od wykupionej taryfy). Producent nie przedstawił informacji o sposobie komunikacji z serwerem, lecz najprawdopodobniej również wykorzystuje sieć GSM. Urządzenie to posiada baterię pozwalającą na ciągłą pracę do 2 tygodni. Urządzenie to nie ogranicza się do lokalizacji pojazdów dzięki niewielkim wymiarom. Producent wprowadza ciekawe funkcjonalności: powiadamianie poprzez wiadomość sms o osiągnięciu przez pojazd danej pozycji geograficznej, wejście w zdefiniowany obszar czy osiągnięcie pewnej prędkości. Aktualne oraz historyczne dane są przedstawiane użytkownikowi poprzez stronę internetową na mapach od firmy Google. Wymiary urządzenia to zaledwie ok. 5 cm x 2,5 cm x 2 cm. W opcji dodatkowej można dokupić wodoodporną obudowę, pozwalającą na zamontowanie urządzenia na zewnątrz pojazdu. Cena urządzenia to 70\$ oraz od 25\$ do 40\$ miesięcznej opłaty. Wygląd urządzenia pokazano na rysunku 1.4.



Rysunek 1.4: Urządzenie STI GL300. Źródło: [3]

# Rozdział 2

## Wstęp teoretyczny

### 2.1 Zastosowane protokoły i systemy

Zgodnie z założeniami pracy, aby zrealizować postawiony urządzeniu cel, zdecydowano się wykorzystać wymienione poniżej protokoły i interfejsy:

- **GSM** - Wykorzystywany do komunikacji zdalnej dalekiego zasięgu (wysyłanie danych na serwer oraz komunikacja z użytkownikiem)
- **GPS** - Wykorzystywany do wyznaczenia lokalizacji pojazdu, a także jego prędkości oraz azymutu.
- **NMEA 0183** - Jest to standard w jakim moduł GPS wysyła dane do mikrokontrolera
- **Bluetooth Low Energy** - Wykorzystywany do komunikacji bliskiego i średniego zasięgu (komunikacja z użytkownikiem oraz z urządzeniem deaktywującym)
- **NFC** - Wykorzystywany do komunikacji bardzo bliskiego zasięgu (do wysłania klucza szyfrującego do- oraz komendy deaktywującej od urządzenia deaktywującego)

Wszystkie z powyższych protokołów zostały pokrótko opisane, a następnie krótko zestawione w tym rozdziale.

## 2.2 System GSM

Poniższy podrozdział powstał na podstawie źródeł [4], [5] oraz [6].

System GSM (*ang. Global System for Mobile Communication*) jest obecnie najpowszechniej wykorzystywanym systemem służącym do komunikacji bezprzewodowej dalekiego zasięgu. System ten wykorzystywany jest do przesyłania głosu oraz serwisów danych. Pomysł na stworzenie sieci umożliwiającej komunikację głosową wyłonił się we wczesnych latach 70. ubiegłego wieku z opracowywanej w siedzibie Bell Laboratories mobilnej sieci radiowej. Jednakże dopiero dwanaście lat później, w 1982 roku powstał oficjalny komitet normalizacyjny nazwany *Groupe Spécial Mobile*, którego zadaniem było utworzenie jednolitego, otwartego standardu dla telefonii komórkowej.

Pierwotna wersja standardu działała w paśmie 900 MHz (880 - 960 MHz) i umożliwiała jedynie transmisję głosową. Jego kolejna wersja została opublikowana w 1990r. i definiowała ona dodatkowe pasmo 1800 MHz (1710 - 1880 MHz). Ponadto, umożliwiała przesyłanie krótkich wiadomości SMS (*ang. Short Message System*), a także faxu czy transmisję danych. Dalsze prace nad systemem wprowadziły do standardu techniki zwiększające przepustowość transmisji (maksymalna prędkość odbioru - 57.6 kb/s, maksymalna prędkość nadawania - 14.5 kb/s oraz 30 - 80 kb/s przy transmisji GPRS) oraz mechanizm przesyłania danych w pakietach GPRS (*ang. General Packet Radio Service*).

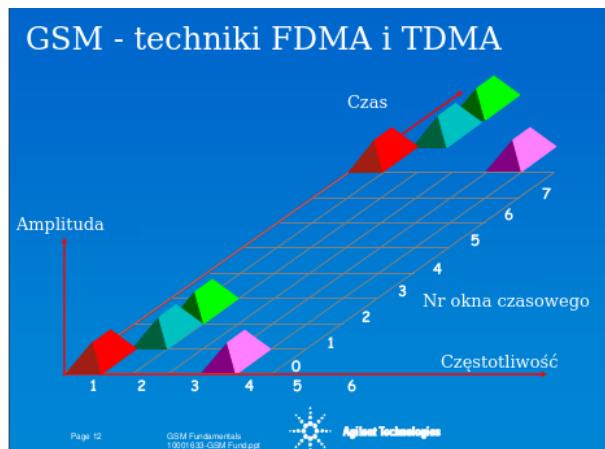
Pomimo pojawienia się na świecie nowszych rozwiązań, takich jak sieci UMTS i LTE, ze względu na ogólną popularność, architektura sieci GSM wciąż jest rozwijana.

System GSM umożliwia skorzystanie z następujących usług:

- Połączenia głosowe - Stanowią one sztandardową funkcjonalność sieci GSM. Jej standard definiuje kodek GSM, który służy do zamiany głosu (skonwertowanego do napięciowego sygnału analogowego przez mikrofon) na postać cyfrową, która jest następnie kompresowana stratnie i transmitowana do odbiorcy. Stosowana jest kompresja na podstawie algorytmu LPC (*ang. Linear Predictive Coding*). Po stronie odbiorcy sygnał jest dekodowany, lecz ze względu na stratność LPC, słyszalny jest zniekształcony, nienaturalny głos rozmówcy.
- Transmisja danych - Umożliwia dostęp do internetu z urządzenia GSM, a także korzystanie z transmisji strumieniowej.
- Wiadomości tekstowe i multimedialne - Usługa przesyłania krótkich wiadomości tekstowych, o długości do 160 znaków, pod warunkiem korzystania jedynie z alfabetu łacińskiego. W przypadku stosowania znaków diakrytycznych maksymalny rozmiar wiadomości

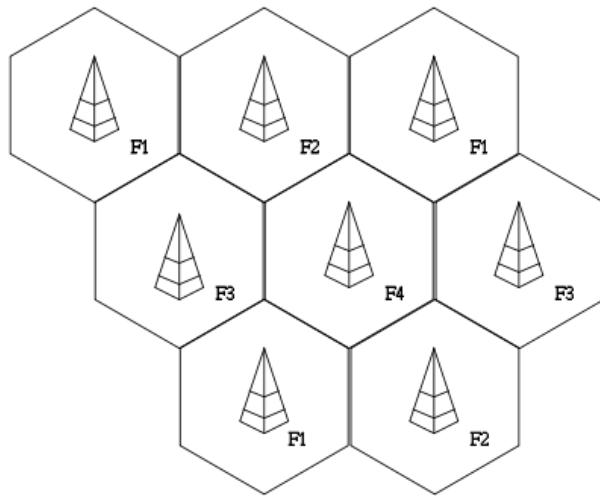
spada do 70 znaków. Wiadomości multimedialne (inaczej MMS), umożliwiają przesyłanie zdjęć, filmów czy dźwięków. Ich rozmiar maksymalny jest uzależniony od ograniczeń telefonu oraz operatora.

Jednym z głównych założeń systemu jest możliwość korzystania z niego przez wielu użytkowników jednocześnie. Aby rozwiązać ten problem, postanowiono zastosować technikę zmiany częstotliwości FDMA (*ang. Frequency Division Multiple Access*) oraz okien czasowych TDMA (*ang. Time Division Multiple Access*). Oznacza to, że pasmo częstotliwości GSM jest podzielone na wąskie kanały, o szerokości 200 kHz każdy. Czas użytkowania każdego kanału podzielony jest na 8 okien czasowych. Każde urządzenie ma zatem dostęp do sieci dostrajając się do odpowiedniego kanału w czasie trwania przydzielonego okna czasowego. Przedstawiono to na rysunku 2.1.



Rysunek 2.1: Podział pasma częstotliwości na kanały i okna czasowe. Źródło: [4]

Architektura sieci GSM powstała w oparciu o komórkowy system radiowy, skąd powszechnie stosowana nazwa - sieć komórkowa. Charakteryzuje się ona tym, że obszar terenu, na którym ma być prowadzona komunikacja radiowa dzieli się na tzw. komórki. Każdej komórce przypisana jest stacja bazowa (*ang. Base Transceiver Station*), która stanowi bramę dostępową do sieci. Urządzenie mobilne GSM, takie jak na przykład telefon, znajdując się na obszarze komórki najczęściej odbiera sygnał z więcej niż jednej stacji bazowej, jednakże zawiera połączenie z tą, której sygnał jest najsilniejszy. W razie spadku mocy sygnału stacji z którą urządzenie jest połączone, możliwa jest dynamiczna zmiana połączenia do innego BTS'a.



Rysunek 2.2: Podział obszaru na komórki. Źródło: [6]

Aby móc korzystać z sieci GSM, urządzenie muszą posiadać kartę SIM (ang. *Subscriber Identification Module*). Oprócz przydatnych dla użytkownika wbudowanej pamięci na wiadomości SMS i kontakty, posiada unikalny na całym świecie numer identyfikujący użytkownika w sieci. W celu zalogowania się do sieci, urządzenie GSM musi podać ten numer w trakcie nawiązywania połączenia ze stacją bazową.

Każda stacja bazowa wykorzystuje wiele kanałów GSM. Jednakże, aby nie dopuścić do wzajemnego zakłócania się, stacje bazowe z przylegającymi do siebie cel wykorzystują inne zbiory kanałów. Dodatkowo, w stacjach bazowych stosuje się dwa rodzaje anten - dookółne i kierunkowe o pokryciu  $120^\circ$ . Anteny dookółne pokrywają cały obszar komórki tym samym zbiorem kanałów, natomiast kierunkowe - dla każdego podobszaru wykorzystują ich inny zestaw. Ze względu na skończoną prędkość sygnału radiowego, istnieje również maksymalny promień pojedynczej komórki. W praktyce wynosi on około 35 km. Poniższa istnieje konieczność umożliwienia prowadzenia komunikacji na tak duży zasięg, standard ten nie należy do najbardziej energooszczędnego. W zależności od klasy urządzenia, minimalna moc nadawanego sygnału może wynosić od 1 do 20 mW, natomiast maksymalna nawet do 8 W. Urządzenia GSM mają możliwość dostosowywania mocy transmisji na podstawie mocy sygnału odebranego od stacji bazowej, w celu ograniczenia wysokiego zużycia energii.

Biorąc jednak pod uwagę fakt, iż transmisja przebiega w oknach czasowych, moc średnia jest niższa. Każde okno czasowe trwa  $577 \mu s$ , a w jego czasie można wysłać jedną z kilku ramek komunikacyjnych. W trakcie każdej z nich można wysłać 148 bitów danych. W przypadku ramki nadawanej w trakcie rozmowy, głos kodowany jest jedynie na 57 bitach. Każde z urządzeń w sieci otrzymuje okno czasowe co  $4.615 \text{ ms}$ , liczone od początku okna, do rozpoczęcia następnego. Stąd wynika, że w trakcie nadawania, urządzenie transmituje dane jedynie przez 12.5%

czasu. Pobór prądu, w zależności od odległości do nadajnika, a więc od mocy nadawania, może wówczas wynosić nawet do 1.5 A. Przyjmując napięcie zasilania układu GSM wynoszące 4 V, maksymalna pobierana moc średnia może wynosić:

$$P_{\text{sr}} = U \cdot I \cdot \tau \quad (2.1)$$

$$P_{\text{sr}} = 4V \cdot 1.5A \cdot 0.125 = 0.75W$$

gdzie:

$P_{\text{sr}}$  - moc średnia

$U$  - napięcie zasilania układu,

$I$  - natężenie prądu w momencie transmisji

$\tau$  - współczynnik wypełnienia impulsu (czas trwania okna czasowego podzielony przez czas pomiędzy oknami czasowymi)

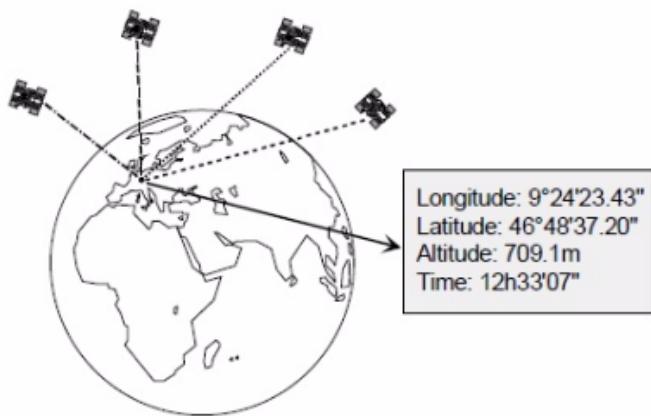
Wartość mocy średniej wynosząca 0.75 W odpowiada ciąglemu zużyciu prądu rzędu 187.5 mA przy napięciu zasilania układu rzędu 4 V.

## 2.3 System GPS

Poniższy podrozdział powstał na podstawie źródeł [7] oraz [8].

System GPS (*ang. Global Positioning System*) był historycznie pierwszym systemem nawigacji satelitarnej GNSS (*ang. Global Navigation Satellite System*). Powstał w wyniku prac w Departamencie Obrony Stanów Zjednoczonych i jest w pełni własnością rządu tego kraju. Oprócz niego istnieją jeszcze rosyjski GLONASS, a od niedawna europejski Galileo oraz chiński Beidou. Ostatnie dwa systemy satelitarne nie są jeszcze w pełni funkcjonalne, stanowią raczej systemy o zasięgu regionalnym niż globalnym.

Głównym elementem składowym systemów nawigacji satelitarnej, są jak sama nazwa wskazuje satelity. Poruszają się one po ściśle określonych, stałych orbitach, które są tak dobrane, aby z dowolnego punktu na globie, w dowolnym momencie była możliwość odebrania sygnału z co najmniej czterech z nich. Przedstawia to rysunek 2.3.



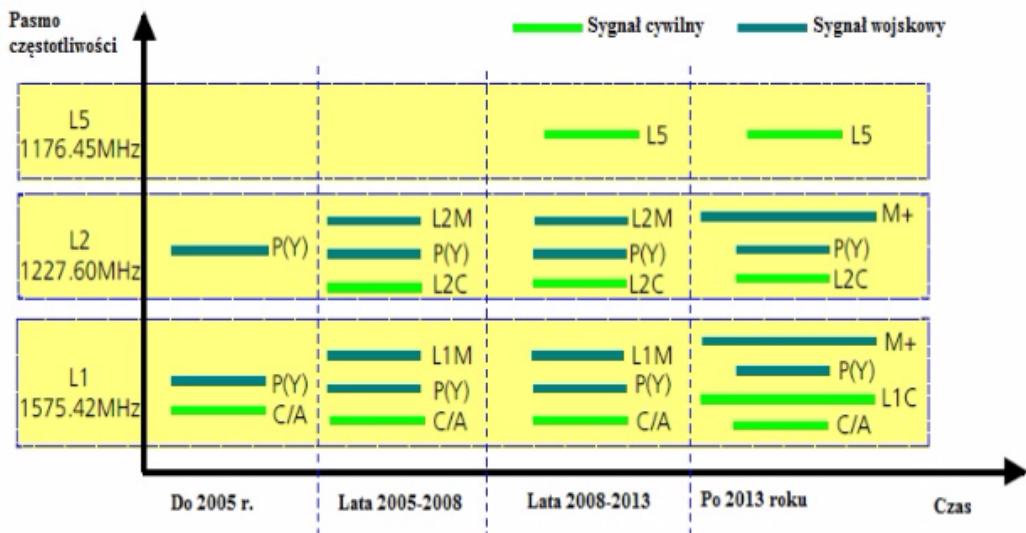
Rysunek 2.3: Model działania systemu GPS. Źródło: [7]

Podstawę w systemach GNSS stanowi czas. Każdy z satelitów posiada 4 zegary atomowe, które stanowią najdokładniejsze źródło czasu znane ludzkości. Zegary te posiadają błąd rzędu 1 sekundy po upływie najwcześniej 30000 lat. Dodatkowo, są one co pewien czas synchronizowane ze źródłami na Ziemi.

Zadaniem każdego z satelitów jest nadawanie w formie rozgłoszeniowej sygnału, w którym zawarta jest wiadomość o jego lokalizacji na orbicie oraz czasie w momencie wysyłania wiadomości. Sygnał ten nadawany jest drogą radiową więc jego prędkość jest równa prędkości światła. Po dotarciu do odbiornika jest on bardzo słaby, przez co praktycznie niemożliwe jest jego odebranie wewnętrz budynków, a w pobliżu wysokich obiektów dokładność lokalizacji spada. Najdokładniejsze wyniki wyznaczania pozycji można osiągnąć na otwartej przestrzeni. Podstawowy sygnał przesyłany jest na fali nośnej o częstotliwości 1575.42 MHz, która nosi nazwę L1.

Ponadto, sygnał GPS (a także pochodzący z innych systemów lokalizacji satelitarnej) łatwo poddaje się zakłóceniom w momencie przejścia przez jonosferę, bowiem fala elektromagnetyczna ulega na niej załamaniu, przez co zmienia swój tor i wydłuża drogę. W efekcie, pomiary odległości od odbiornika do satelity, niezbędne do wyznaczenia lokalizacji przestają być dokładne i pojawia się błąd lokalizacji. Problem ten rozwiązano na 2 sposoby. Pierwszym z nich jest nadawanie sygnału przez satelity na kilku częstotliwościach. Każda z nich, przechodząc przez jonosferę ulega załamaniu, lecz pod innym kątem, przez co pokonają różne długości drogi przebytej do odbiornika, a tym samym zostaną odebrane w różnych momentach. Dzięki temu, odbiornik jest w stanie wyznaczyć korektę i wyeliminować błąd. Pierwotnie, rozwiązanie to było dostępne jedynie w celach militarnych, lecz od 2005 roku, Departament Obrony Stanów Zjednoczonych udostępnił częstotliwość L2 (1227.60 MHz) do celów cywilnych, a od 2008r. -

również L5 (1176.45 MHz). Częstotliwości dostępne w systemie GPS pokazano na rysunku 2.4.

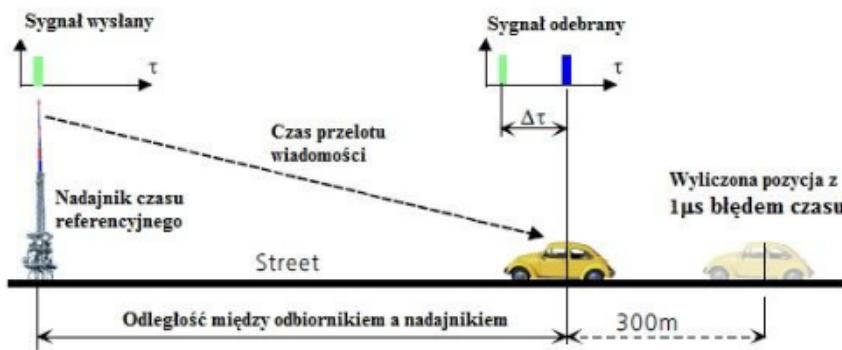


Rysunek 2.4: Zbiór częstotliwości wykorzystywanych w systemie GPS. Źródło: [8]

Druga metoda to wyznaczenie korekty dla przejścia przez jonusferę w stacjach naziemnych, a następnie rozgłoszanie jej w postaci depeszy poprzez sieć stacji bazowych. Rozwiązanie to nosi miano DGPS (ang. **Differential GPS**). Dzięki zastosowaniu tej techniki, dokładność lokalizacji wzrasta z nominalnych 15 m nawet do 10 cm.

Układy GPS, które umożliwiają skorzystanie z któregoś z tych dwóch rozwiązań są jednak kosztowne, więc na rynku cywilnym najpowszechniej stosowane są moduły wykorzystujące jedynie częstotliwość L1. Dzięki temu, za kilkadziesiąt złotych można uzyskać system, który w sprzyjających warunkach charakteryzuje się dokładnością do kilku metrów[8] (Rozdział 7 - Testy urządzenia). W pracy tej uzyskano dokładność systemu rzędu 1 - 2 metrów w terenie otwartym, oraz 5 - 7 metrów idąc chodnikiem wzdłuż wysokich budynków.

Aby zrozumieć zasadę działania systemu GPS proszę wyobrazić sobie sytuację jak na rysunku 2.5:



Rysunek 2.5: Zasada działania systemu GPS. Źródło: [8]

Założymy, że w przedstawionym pojeździe znajduje się odbiornik GPS. W pewnym momencie odbiera on sygnał z nadajnika referencyjnego. W sygnale znajduje się wartość czasu w momencie wysłania wiadomości. Ze względu na skończoną prędkość światła ( $c = 299792458\text{m/s}$ ), zostanie ona odebrana przez odbiornik z pewnym opóźnieniem. Wykorzystując ten fakt i znając prędkość transmisji (wartość prędkości światła), można wyznaczyć odległość do nadajnika:

$$D = c \cdot \Delta t \quad (2.2)$$

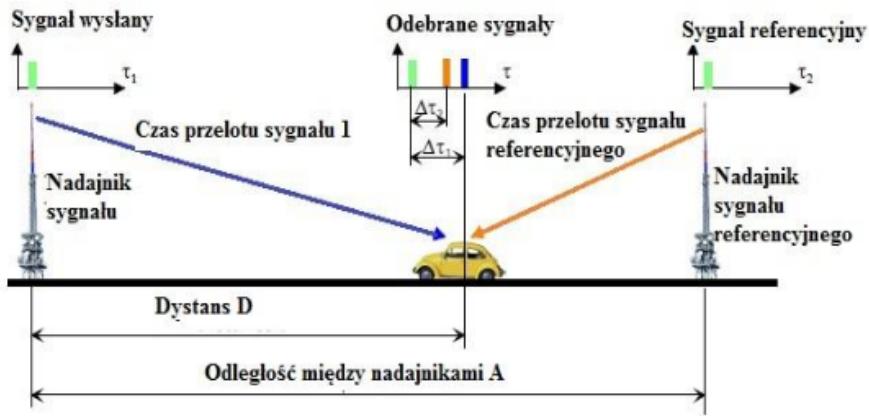
gdzie,

$D$  - odległość między nadajnikiem i odbiornikiem

$c$  - prędkość światła

$\Delta t$  - różnica czasu między wysłaniem i odebraniem wiadomości

Aby wyznaczyć różnicę czasu, odbiornik powinien mieć również własny zegar. Powinien on być przy tym niezwykle dokładny i zsynchronizowany z zegarem w nadajniku, bowiem błąd rzędu  $1\text{ }\mu\text{s}$  powoduje błąd lokalizacji rzędu 300 m. Ponieważ uzyskanie takiej dokładności oraz synchronizacji w każdym odbiorniku jest niemożliwe, należało znaleźć sposób umożliwiający rezygnację z konieczności posiadania przez nie zegara. Przedstawiono go na rysunku 2.6:



Rysunek 2.6: Zasada działania systemu GPS - sygnał referencyjny. Źródło: [8]

Polega on na zastosowaniu dodatkowego sygnału referencyjnego czasu. Wówczas po odebraniu obu sygnałów (które zostały wysłane w tym samym momencie) otrzymujemy:

$$\begin{cases} \Delta\tau_1 \cdot c = D \\ \Delta\tau_2 \cdot c = A - D \end{cases} \quad (2.3)$$

Gdzie:

$\tau_1$  - różnica czasu między wysłaniem sygnału z nadajnika, a momentem jego odebrania

$\tau_2$  - różnica czasu między wysłaniem sygnału z nadajnika referencyjnego, a momentem jego odebrania

$A$  - odległość między nadajnikami

$D$  - odległość od nadajnika do odbiornika

Po odjęciu stronami drugiego równania od pierwszego otrzymamy:

$$\begin{aligned} (\Delta\tau_1 - \Delta\tau_2) \cdot c &= 2D - A \\ D &= \frac{(\Delta\tau_1 - \Delta\tau_2) \cdot c + A}{2} \end{aligned} \quad (2.4)$$

Ponieważ jednak:

$$\begin{cases} \Delta\tau_1 = t - \tau_1 \\ \Delta\tau_2 = t - \tau_2 \end{cases} \quad (2.5)$$

to

$$(\Delta\tau_1 - \Delta\tau_2) = ((t - \tau_1) - (t - \tau_2)) = \tau_2 - \tau_1 \quad (2.6)$$

Gdzie:

$t$  - czas w momencie nadania sygnału przez nadajniki

$\tau_1$  - różnica czasu między wysłaniem sygnału z nadajnika, a momentem jego odebrania

$\tau_2$  - różnica czasu między wysłaniem sygnału z nadajnika referencyjnego, a momentem jego odebrania

Powyższe równania prowadzą do wniosku, że zastosowanie dodatkowego nadajnika referencyjnego, zsynchronizowanego z satelitami powoduje eliminację konieczności posiadania zegara w odbiorniku.

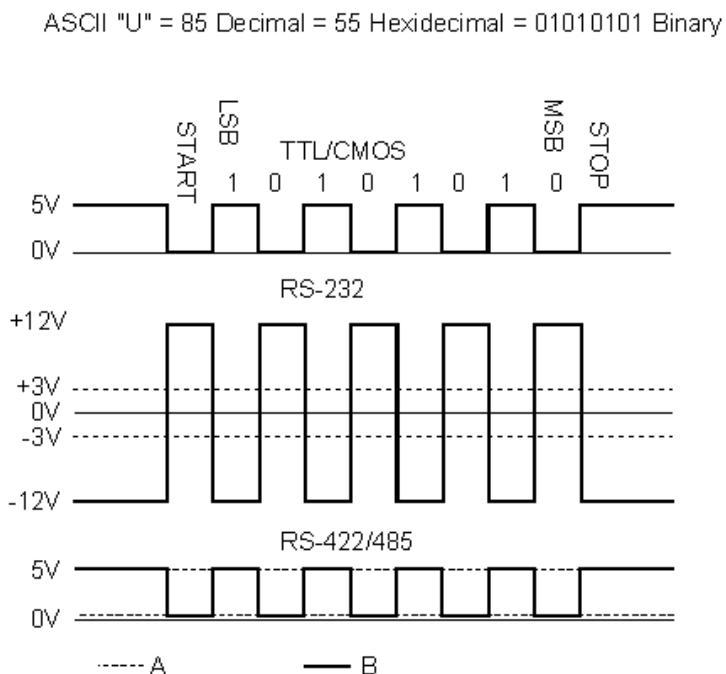
W rzeczywistości, takim nadajnikiem referencyjnym jest inny satelita GPS, bowiem są one ze sobą zsynchronizowane i nadają w dokładnie tym samym momencie.

Wyznaczona powyżej odległość  $D$  dotyczy odcinka (jednej osi współrzędnych), a w rzeczywistości do wyznaczenia lokalizacji niezbędne są trzy osie. Uwzględniając zatem satelitę referencyjnego, do lokalizacji odbiornika potrzeba sygnału z co najmniej 4 satelitów.

## 2.4 Protokół NMEA 0183

Niniejszy podrozdział powstał na podstawie źródeł [8] oraz [10].

Protokół ten stanowi standard komunikacji między urządzeniami elektronicznymi wykorzystywanymi w urządzeniach morskich, zwłaszcza urządzeniami do lokalizacji i nawigacji. Zawiera on specyfikację elektryczną oraz opis ramek (wiadomości) wymienianych między modułami. Powstał w Stanach Zjednoczonych w *National Marine Electronics Association*. Jego najnowsza, czwarta wersja pochodzi z listopada 2008 roku. Protokół ten pierwotnie wykorzystywał interfejs RS232, lecz w wersji drugiej dokonano jego zmiany na RS422. Interfejsy te różnią się jedynie poziomami napięć przypisanym logicznym wartościom bitów 0 i 1. Oba z nich umożliwiają wysłanie danych bajt po bajcie. Każdy z nich rozpoczyna się bitem START (stan niski w RS422), który umożliwia odbiornikowi wykrycie początku bajtu i synchronizację. Następnie następuje wysłanie kolejno 8 bitów danych, po których przesyłany jest bit parzystości (0 gdy liczba bitów w bajcie danych o wartości 1 jest parzysta lub 1 gdy jest nieparzysta) i na koniec - bit stopu (stan wysoki). Przedstawiono to na rysunku 2.7.



Rysunek 2.7: Poziomy napięć i kolejność bitów w interfejsach RS232 i RS422. Źródło: [9]

Na podstawie tego interfejsu, NMEA nabudowała wyższą warstwę protokołu w postaci wiadomości. Każda wiadomość rozpoczyna się symbolem '\$', po którym występuje 2 literowy kod mówiący o typie urządzenia (GP - urządzenie GPS, GN - urządzenie GLONASS) i 3 literowy kod definiujący typ przesyłanej wiadomości. Po kodzie występuje przecinek, a następnie lista pól danych oddzielonych przecinkami. W obrębie wiadomości, każde pole ma ścisłe określona funkcję i w razie nie występowania, musi zostać przesłane jako puste. Za ostatnim polem występuje znak '\*', po którym występuje suma kontrolna, liczona jako funkcja Exclusive Or (XOR) ze wszystkich znaków między '\$', a '\*' bez ich uwzględnienia.

Lista zdefiniowanych wiadomości jest bardzo dłuża, jednak w tabeli 2.4 zestawiono najpowszechniej wykorzystywane w odbiornikach GPS.

*Tabela 2.1: Najczęściej wykorzystywane wiadomości NMEA0183 w odbiornikach GPS.**Źródło: [8]*

GGA	Dane związane z ustalaniem pozycji GPS
GGL	Pozycja geograficzna – długość, szerokość
GSA	Informacje o aktywnych satelitach oraz o jakości połączenia (DOP – ang. <i>Dilution of Position</i> )
GSV	Informacje o satelitach w zasięgu
RMC	Rekomendowane minimum danych GNSS
VTG	Dane o kursie oraz prędkości
ZDA	Wiadomość z aktualnym czasem i datą

W niniejszej pracy, aby zrealizować założenia projektu niezbędne było sparsowanie (przeanalizowanie) danych z dwóch wiadomości nadawanych przez moduł GPS: GGA i VTG. Z wiadomości GGA wyłoniono informacje o długości i szerokości geograficznej, wskaźnikach półkul, statusie wyznaczenia lokalizacji, liczbie satelitów w zasięgu oraz wysokości nad poziomem morza. Wiadomość VTG została wykorzystana w celu uzyskania informacji o kursie (azymucie) oraz prędkości odbiornika. Ich struktury przedstawiono kolejno w tabelach 2.2 i 2.3.

Tabela 2.2: Struktura wiadomości GGA. Źródło: Twórczość własna

Pole	Opis
\$	Symbol początku wiadomości
GP	Typ urządzenia
GGA	Typ wiadomości
130305.743	Czas UTC w formacie hhmmss.sss
4717.115	Szerokość geograficzna w formacie ddmm.mmm
N	Wskaźnik półkuli (N - północna, S - południowa)
00833.912	Długość geograficzna w formacie dddmm.mmm
E	Wskaźnik półkuli (W - zachodnia, E - wschodnia)
1	Wskaźnik informujący o statusie wyznaczania pozycji 0 - pozycja nieustalona 1 - pozycja ustalona na podstawie sygnału z satelitów 2 - pozycja ustalona przy pomocy DGPS 6 - pozycja wyestymowana za pomocą mechanizmu <i>Dead reckoning</i>
08	Liczba satelitów z których odebrano sygnał
0.94	Wskaźnik jakości sygnału (1 - najlepszy, 20 - bardzo słaby)
00499	Wysokość nad poziomem morza
M	Jednostki wysokości (M - metry)
047	Różnica w wysokości między geoidą (Ziemią), a elipsoidą (przybliżeniem Ziemi)
M	Jednostki wysokości (M - metry)
„	Dane GPS (pole puste)
0000	Numer identyfikacyjny stacji bazowej GPS
*	Znak końca danych
58	Suma kontrolna
<CR><LF>	Znak końca wiadomości

Tabela 2.3: Struktura wiadomości VTG. Źródło: Twórczość własna

Pole	Opis
\$	Symbol początku wiadomości
GP	Typ urządzenia
VTG	Typ wiadomości
227.15	Kurs (azymut) w stopniach
T	Pole stałe, zawierające symbol T
„	Kurs magnetyczny (nie zaimplementowany przez producenta)
M	Pole stałe, zawierające symbol M
0.00	Prędkość
N	Pole stałe opisujące jednostki prędkości (N - węzły, K - kilometry na godzinę)
0.00	Prędkość
K	Pole stałe opisujące jednostki prędkości (N - węzły, K - kilometry na godzinę)
A	Tryb pozycjonowania N - brak pozycji A - pozycja na podstawie sygnału z satelitów D - pozycja na podstawie DGPS
*	Znak końca danych
3E	Suma kontrolna
<CR><LF>	Znak końca wiadomości

## 2.5 Protokół Bluetooth Low Energy

Poniższy podrozdział powstał na podstawie źródeł [11] oraz [8].

Projekt protokołu BLE (*ang. Bluetooth Low Energy*) został zapoczątkowany przez firmę Wibree należącą do grupy Nokia. Celem nadzorowanym, który przyświecał jego autorom nie było utworzenie kolejnego protokołu, którego zastosowanie byłoby przesadnie szerokie. Zamiast tego, zdecydowali się oni na zaprojektowanie standardu radiowego umożliwiającego najniższe możliwe zużycie energii, a przy tym nieskomplikowanego, przez co możliwe byłoby zastosowanie go w systemach o niskich kosztach budowy. Innymi słowy są to założenia idealne dla rynku smartfonów oraz IoT (*ang. Internet of Things*), gdzie urządzenia zasilane są z niewielkich baterii (często CR2032). Dla przykładu, zakup w liczbie 1000 sztuk mikrokontrolera użytego w tej pracy - nRF52832 wyprodukowanego przez Nordic Semiconductor, który posiada wbudowany

stos BLE, kosztuje jedynie 9.78 zł za sztukę.

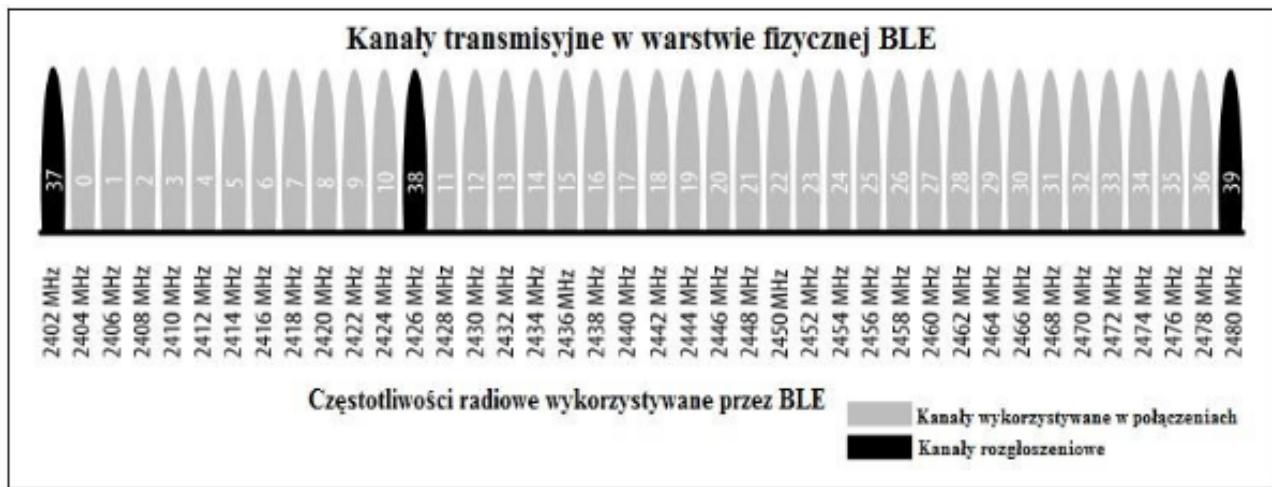
W trakcie prac, projekt został przejęty przez grupę Bluetooth SIG (*ang. Bluetooth Special Interests Group*), zrzeszającą dziesiątki firm i organizacji z wielu dziedzin przemysłu, zainteresowanych wykorzystywaniem i rozwojem protokołu Bluetooth. W roku 2010, Bluetooth Low Energy, znany również jako Bluetooth Smart został włączony do standardu Bluetooth 4.0, obok klasycznego protokołu Bluetooth Classic. Nie należy jednakże mylić tych dwóch protokołów komunikacyjnych, ponieważ poza warstwą fizyczną (interfejs radiowy o częstotliwości 2.4 GHz w pasmie ISM) różnią się w swych założeniach. Bluetooth Classic jest bowiem typowym protokołem umożliwiającym szybką, lecz energochłonną komunikację. W grudniu 2013 roku wprowadzono pierwszą dużą poprawkę do protokołu (Bluetooth 4.1), a rok później dalsze modyfikacje w postaci standardu Bluetooth 4.2.

Bluetooth Low Energy, ze względu na swoje założenie o onergoszczędności posiada pewne ograniczenia. Pierwszym z nich jest przepustowość danych. Góra granica prędkości transmisji wynosi 1 Mb/s, jednakże jest to wartość jedynie teoretyczna. W praktyce jest ona obwarowana wieloma ograniczeniami sprzętowymi producentów mikrokontrolerów. W standardzie zawarte jest, iż pojedynczy pakiet danych może zawierać maksymalnie 20 bajtów. Ograniczenie sprzętowe wynika tu z częstotliwości wysyłania pakietów. Dla mikrokontrolera Nordic Semiconductor z rodziny nRF51, wynosi ona do 6 pakietów na każdy interwał połączenia. Jest to konfigurowalny parametr, określający odcinek czasu w obrębie którego jeśli nie dojdzie do transmisji pakietu, połączenie zostanie uznane za zerwane. Interwał połączenia może wynosić od 7.5 ms do 4 s. Przy założeniu najkrótszej wartości tego parametru otrzymujemy przepustowość:

$$\text{Przepustowość} = 6 \text{ pakietów/interwał} \cdot \frac{1000 \text{ ms}}{7.5 \text{ ms}} \cdot 20 \text{ bajtów} = 15960 \text{ bajtów/s} \approx 128 \text{ kb/s} \quad (2.7)$$

Jak widać jest to wartość znacznie odbiegająca od 1 Mb/s, jednakże w porównaniu do zysku na zużyciu energii jest to i tak bardzo dobry wynik.

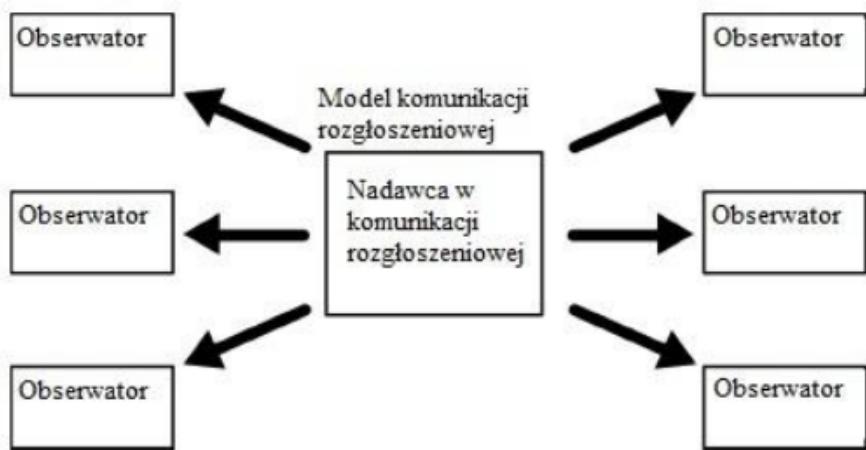
Kolejne ograniczenie to zasięg komunikacji. Oficjalnie, Bluetooth Low Energy posiada zasięg rzędu 50 m. Jest to jednak wartość trudna do uzyskania, silnie zależna od otoczenia (między urządzeniami nie może być przeszkód), mocą transmisji (rekonfigurowalna, im mniejsza tym zasięg mniejszy) oraz liczbą innych urządzeń znajdujących się w pobliżu, która to decyduje o zajętości kanałów. Pasmo częstotliwości (od 2.402 GHz do 2.480 GHz) wykorzystywanej przez protokół podzielone jest bowiem na 40 kanałów. Przedstawiono to na rysunku 2.8.



Rysunek 2.8: Struktura pasma 2.4 ISM wykorzystywanego przez Bluetooth Low Energy.

Źródło: [8]

Protokół Bluetooth Low Energy oferuje dwie możliwości wysyłania danych. Pierwszym z nich jest bezpołączeniowe rozgłaszczenie (ang. *Advertising*). Wówczas, każde z urządzeń wysyła cyklicznie w eter pakiet danych o pojemności do 31 bajtów. Interwał między pakietami może wynosić od 20 ms do 10.24 s. Jest to jednak komunikacja jednokierunkowa. Wysłane w ten sposób dane może odebrać każde urządzenie będące w zasięgu.

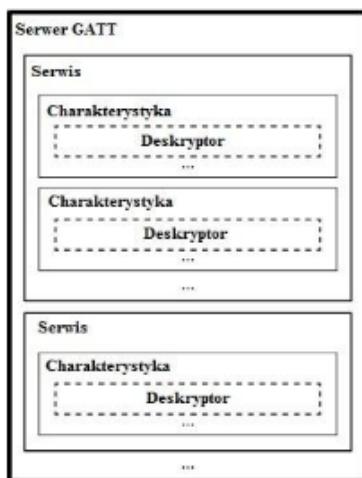


Rysunek 2.9: Model komunikacji rozgłoszeniowej. Źródło: [8]

Drugą metodą jest wysyłanie danych będąc w połączeniu. Wówczas komunikacja może być obustronna. W tym przypadku, BLE definiuje dwa możliwe typy urządzeń. Jedno z nich, które inicjuje połączenie określone jest jako *Central*, natomiast urządzenie akceptujące połączenie - *Peripheral*. Nie występuje przy tym ograniczenie, że urządzenie może mieć tylko jedną rolę.

Może ono będąc w połączeniu urządzeniem typu *Peripheral* zainicjować samodzielnie połączenie z innym odbiornikiem, a więc stać się dla tego połączenia *Central'em*. Ważne jest, że urządzenie posiada jedną rolę dla danego połączenia, które jest realizowane typu punkt - punkt.

Standard Bluetooth Low Energy definiuje logiczny podział struktur danych. Główną jednostką są tak zwane serwisy. Są to zgrupowania pewnych funkcjonalności, zwanych charakterystykami. Charakterystyki stanowią podstawowe jednostki komunikacji. Mogą one zawierać tzw. deskryptory, które są krótkimi, zrozumiałymi dla ludzi informacjami jak na przykład nazwa charakterystyki. Można to porównać do definicji klasy (serwis), zawierającej definicje metod (charakterystyk). Struktura ta zarządzana jest przez tak zwany serwer GATT (*ang. Generic Attribute Server*). Przedstawiono to na rysunku 2.10.



Rysunek 2.10: Model struktury danych serwera GATT. Źródło: [8]

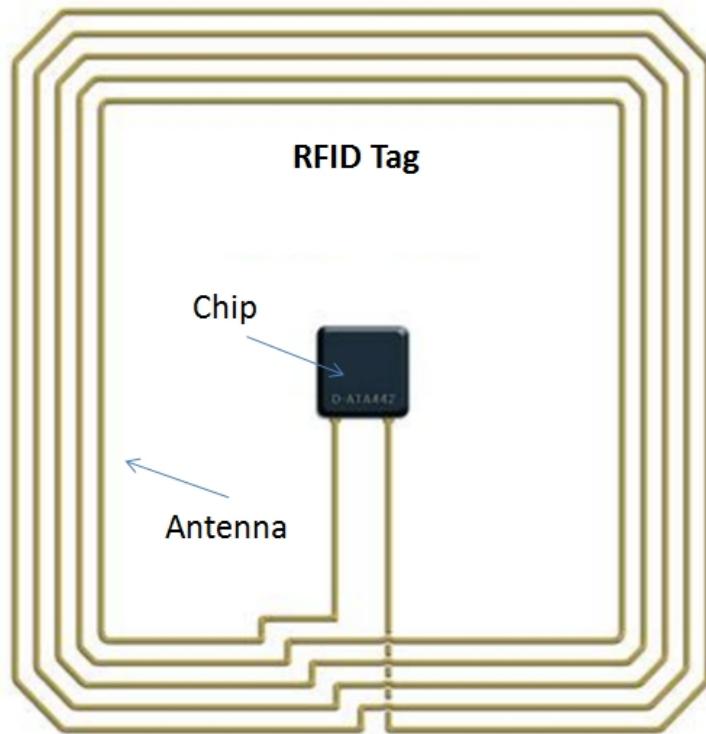
Dla charakterystyk zostały zdefiniowane 4 metody komunikacji. Pierwsza z nich - *Write*, polega na wysłaniu danych z urządzenia typu *Central* do *Peripheral*. Druga - *Read*, umożliwia inicjatorowi połączenia odczytanie danych z urządzenia podległego. Pozostałe 2 metody - *Notify* oraz *Indicate* polegają na wysłaniu danych z urządzenia typu *Peripheral* do urządzenia typu *Central* lub odwrotnie, bez żadnego rządzenia transmisji ze strony odbiorcy. Różnica polega na tym, że *Indicate* wymaga od odbiorcy wysłania potwierdzenia odbioru, a *Notify* nie.

## 2.6 Interfejs NFC

Poniższy rozdział powstał na podstawie źródeł [12] i [14].

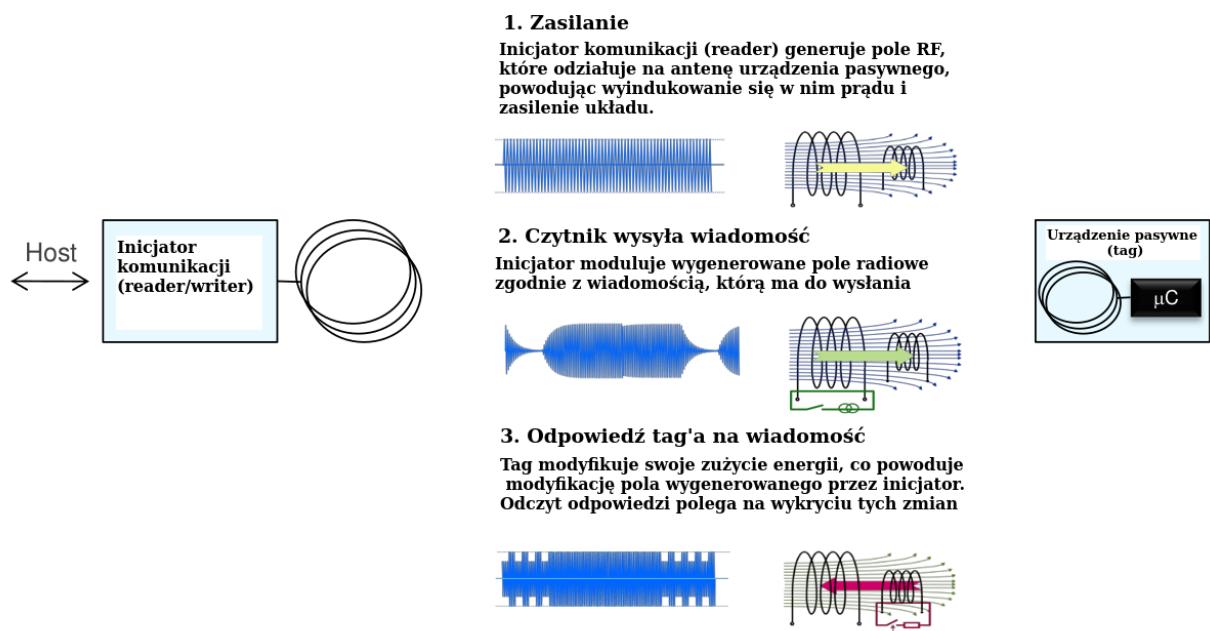
Near Field Communication to protokół radiowy, stanowiący rozszerzenie swego starszego brata - interfejsu RFID. Mimo, iż jest do niego bardzo podobny w wielu aspektach, różni

się znacznie pod względem założeń. RFID (*ang. Radio Frequency Identification*), nie stanowi prawdziwego protokołu komunikacyjnego, bowiem pozwala jedynie na wymianę bardzo krótkich informacji, zwanych identyfikatorami. Urządzenia RFID stanowią bardzo proste układy. Składają się one zazwyczaj z niewielkiego chipa, zawierającego pamięć nieulotną (zazwyczaj do 1 kB) oraz anteny. Przedstawiono to na rysunku 2.11.



Rysunek 2.11: Budowa tag'a RFID. Źródło: [13]

W odróżnieniu od tego, NFC (*ang. Near Field Communication*) stanowi pełnoprawny protokoł komunikacyjny. Umożliwia on wymianę długich wiadomości. Został on zbudowany na podstawie RFID i wykorzystuje jego warstwę fizyczną. Tak samo jak w RFID, w NFC można wyróżnić 2 typy urządzeń - pasywne oraz aktywne. Urządzenie pasywne nie generuje swojego własnego pola elektromagnetycznego, w przeciwieństwie do urządzenia aktywnego, które inicjuje komunikację. Ponadto, urządzenie bierne, tak samo jak w przypadku RFID nie posiada nawet własnego źródła zasilania. Gdy znajdzie się ono w polu wygenerowanym przez urządzenie aktywne, w jego antenie wyindukuje się prąd, który jest w stanie zasilić niewielki moduł. Komunikacja zwrotna odbywa się poprzez modyfikację zużycia energii tag'a (urządzenia pasywnego) zgodnie z bitowym wzorcem, który należy wysłać. Dynamiczne zmiany parametrów zużycia powodują pewne zaburzenia wygenerowanego przez inicjator pola RF. Odczyt danych przez nie polega na odczycie zmian tego pola. Przedstawiono to na rysunku 2.12.



Rysunek 2.12: Zasada działania komunikacji pomiędzy urządzeniem aktywnym i pasywnym.

Źródło: [14]

Istnieje również możliwość komunikacji pomiędzy dwoma urządzeniami aktywnymi, wówczas zamiast modyfikować pole wyindukowane, urządzenie odpowiada swoim własnym, wygenerowanym z energii źródła zasilania polem. Dzięki temu zasięg komunikacji jest większy.

Na tym w zasadzie podobieństwa między NFC i RFID się kończą. RFID pozwala bowiem na odpowiedź w postaci swojego unikalnego numeru identyfikacyjnego UID (*ang. Unique Identifier Number*, natomiast moduły NFC stanowią najczęściej urządzenia programowalne, co pozwala na przesłanie dowolnej wiadomości. Ponadto, kolejną różnicą jest fakt, że RFID nie posiada jednego wspólnego standardu komunikacji. Co więcej, nie posiada nawet stałej częstotliwości komunikacji, lecz zależy ona od producenta sprzętu. Zasięg komunikacji w przypadku RFID również jest zmienny i zależy od częstotliwości sygnału. Dla wartości rzędu 125 - 134.3 kHz wynosi ona do 30 cm (zazwyczaj około 10 cm), dla częstotliwości 13.56 MHz - do 1.5 metra, a w przypadku 433 MHz - nawet do 500 metrów. Ta różnorodność i brak pojedynczego standardu komunikacji stała się przyczyną do powstania protokołu NFC.

NFC pracuje na pojedynczej częstotliwości o wartości 13.56 MHz. Zasięg komunikacji jest niewielki (rzędu 10 cm), a urządzenia mają możliwość emulowania tagów RFID, czyli zachowania się jak one gdy wykryte zostanie pole RF. Dodatkowym atutem NFC jest zdefiniowanie formatu komunikacji pomiędzy urządzeniami - NDEF (*ang. NFC Data Exchange Format*). Istnieją pewne dobrze znane struktury danych, możliwe do wysłania poprzez NFC. Są to:

- Wiadomości tekstowe
- Adresy internetowe URI
- Proste komendy
- Podpisy cyfrowe

Organizacją zajmującą się standaryzacją i rozwijaniem NFC jest NFC Forum. Definiuje ona 4 rodzaje urządzeń pasywnych:

#### 1. Typ 1

- Bazuje na specyfikacji ISO-14443A
- Może być tylko do odczytu lub mieć zdolność do zapisu i odczytu
- Rozmiar pamięci od 96 B do 2 kB
- Prędkość komunikacji - 106 kb/s
- Brak ochrony przed kolizją pól

#### 2. Typ 2

- Bazuje na specyfikacji ISO-14443A
- Może być tylko do odczytu lub mieć zdolność do zapisu i odczytu
- Rozmiar pamięci od 96 B do 2 kB
- Prędkość komunikacji - 106 kb/s
- Zapewnia mechanizm ochrony przed kolizją

#### 3. Typ 3

- Bazuje na specyfikacji ISO-18092 i JS-X-6319-4
- Może być tylko do odczytu lub mieć zdolność do zapisu i odczytu
- Rozmiar pamięci do 1 MB
- Prędkość komunikacji - 212 lub 424 kb/s
- Zapewnia mechanizm ochrony przed kolizją

#### 4. Typ 4

- Bazuje na specyfikacji ISO-18092 i JS-X-6319-4

- Może być tylko do odczytu lub mieć zdolność do zapisu i odczytu
- Rozmiar pamięci: 2, 4 lub 8 kB
- Prędkość komunikacji - 106, 212 lub 424 kb/s
- Zapewnia mechanizm ochrony przed kolizją

## 2.7 Podsumowanie

We wcześniejszych podrozdziałach dokonano krótkiej analizy każdego z protokołów i interfejsów wykorzystanych w pracy. Niniejszy podrozdział stanowi ich podsumowanie ze wskazaniem najważniejszych cech, ograniczeń i możliwości wykorzystania. Z rozważań wykluczono jednakże protokół NMEA 0183 ze względu na fakt, iż jest to jedynie narzucony przez producentów modułów GPS standard komunikacji.

*Tabela 2.4: Podsumowanie cech systemów i protokołów GSM, GPS, BLE oraz NFC.*

*Źródło: Twórczość własna*

Parametr	GSM	GPS	BLE	NFC
<b>Właściwości</b>	<ul style="list-style-type: none"> <li>- Ogromny zasięg, dzięki rozbudowanej sieci stacji naziemnych</li> <li>- Możliwość odbioru i transmisji</li> <li>- Prędkość rzędu 57.6 kb/s (odbiór) i 14.5 kb/s (transmisijsa)</li> <li>- Wysoki pobór prądu (w szczytce do 1.5 A)</li> </ul>	<ul style="list-style-type: none"> <li>- Zasięg globalny</li> <li>- Tylko do odczytu</li> <li>- 50 bit/s</li> <li>- Średni pobór prądu (ok. 30 mA w trakcie śledzenia pozycji)</li> </ul>	<ul style="list-style-type: none"> <li>- Średni zasięg (do 50 m)</li> <li>- Możliwość odbioru i transmisji</li> <li>- Około 125 kb/s</li> <li>- Niski pobór prądu (ok. 7 - 14 mA w trakcie transmisji)</li> </ul>	<ul style="list-style-type: none"> <li>- Bardzo bliski zasięg (do 10 cm)</li> <li>- Możliwość odbioru i transmisji</li> <li>- Około 106 kb/s</li> <li>- Pobór energii rzedu 100 mA w trybie inicjatora, 0 mA w trybie pasywnym</li> </ul>
<b>Możliwości wykorzystania</b>	<ul style="list-style-type: none"> <li>- Rozmowy głosowe</li> <li>- Wiadomości SMS</li> <li>- Dostęp do internetu</li> </ul>	<ul style="list-style-type: none"> <li>- Odczyt lokalizacji</li> <li>- Bardzo dokładny odczyt prędkości</li> <li>- Bardzo dokładne źródło czasu</li> </ul>	<ul style="list-style-type: none"> <li>- Energooszczędna transmisja danych</li> </ul>	<ul style="list-style-type: none"> <li>- Bezpieczna, bezkontaktowa transmisja danych</li> <li>- Parowanie urządzeń</li> <li>- Wymiana kluczy szyfrujących</li> </ul>

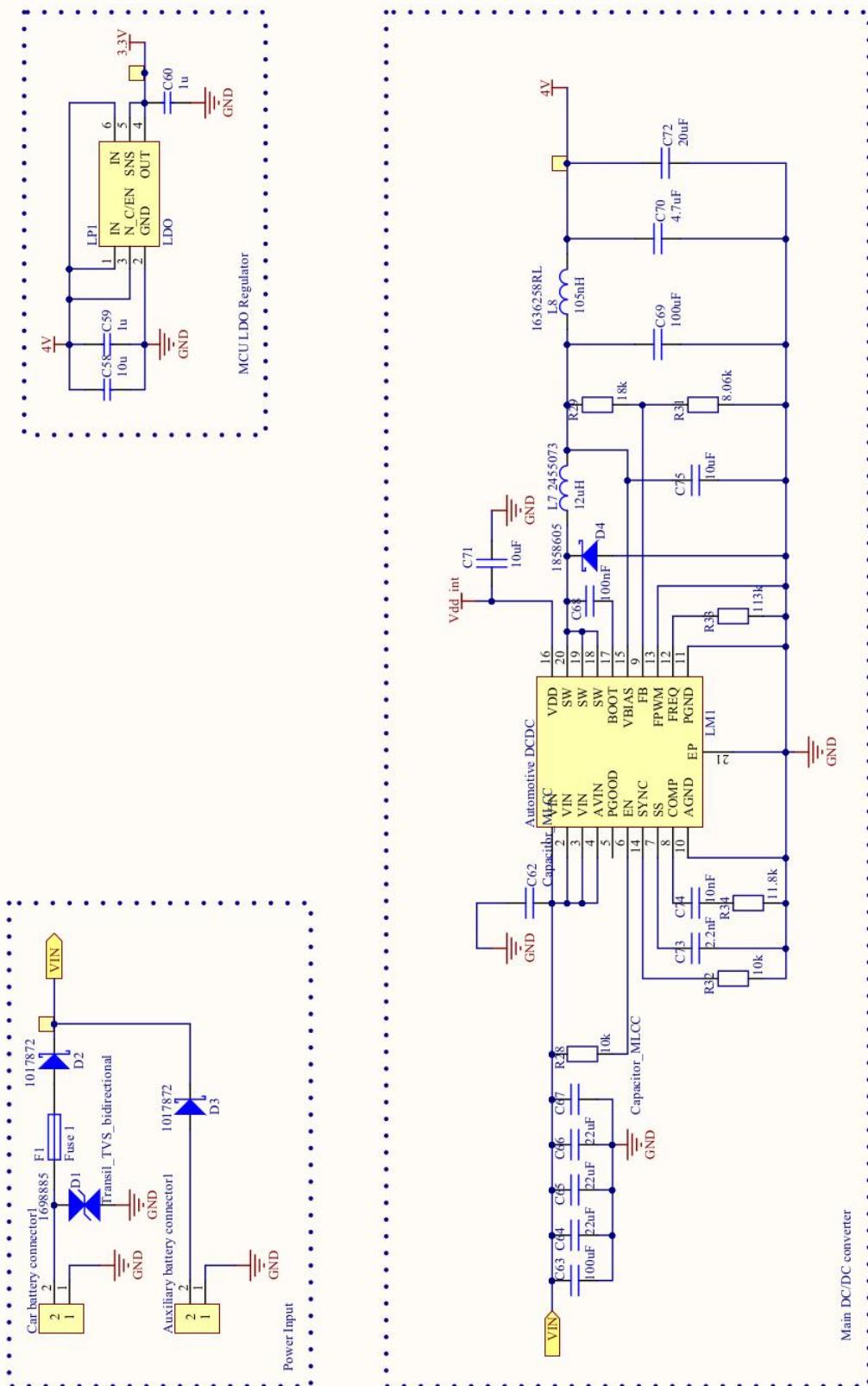
# Rozdział 3

## Schematy elektroniczne urządzeń

### 3.1 Urządzenie lokalizujące

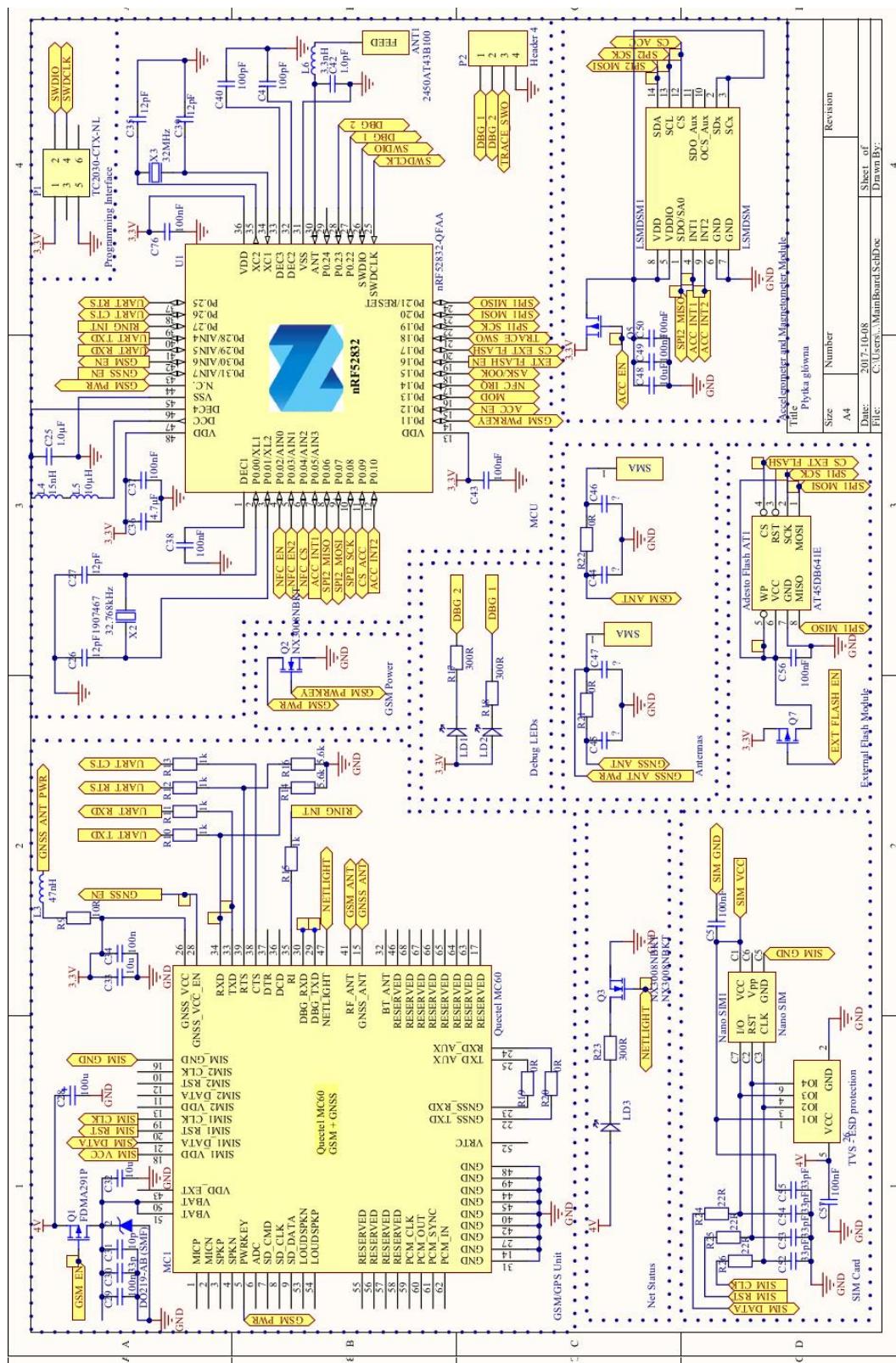
Ze względu na poziom skomplikowania układu, schemat elektroniczny musiał zostać rozbity na podschematy. W urządzeniu lokalizującym można wyróżnić trzy znaczące moduły elektroniczne, realizujące odpowiednie funkcje. Są to:

- Moduł zasilania, przedstawiony na rysunku 3.1
- Moduł funkcjonalny, przedstawiony na rysunku 3.2
- Moduł NFC, przedstawiony na rysunku 3.3



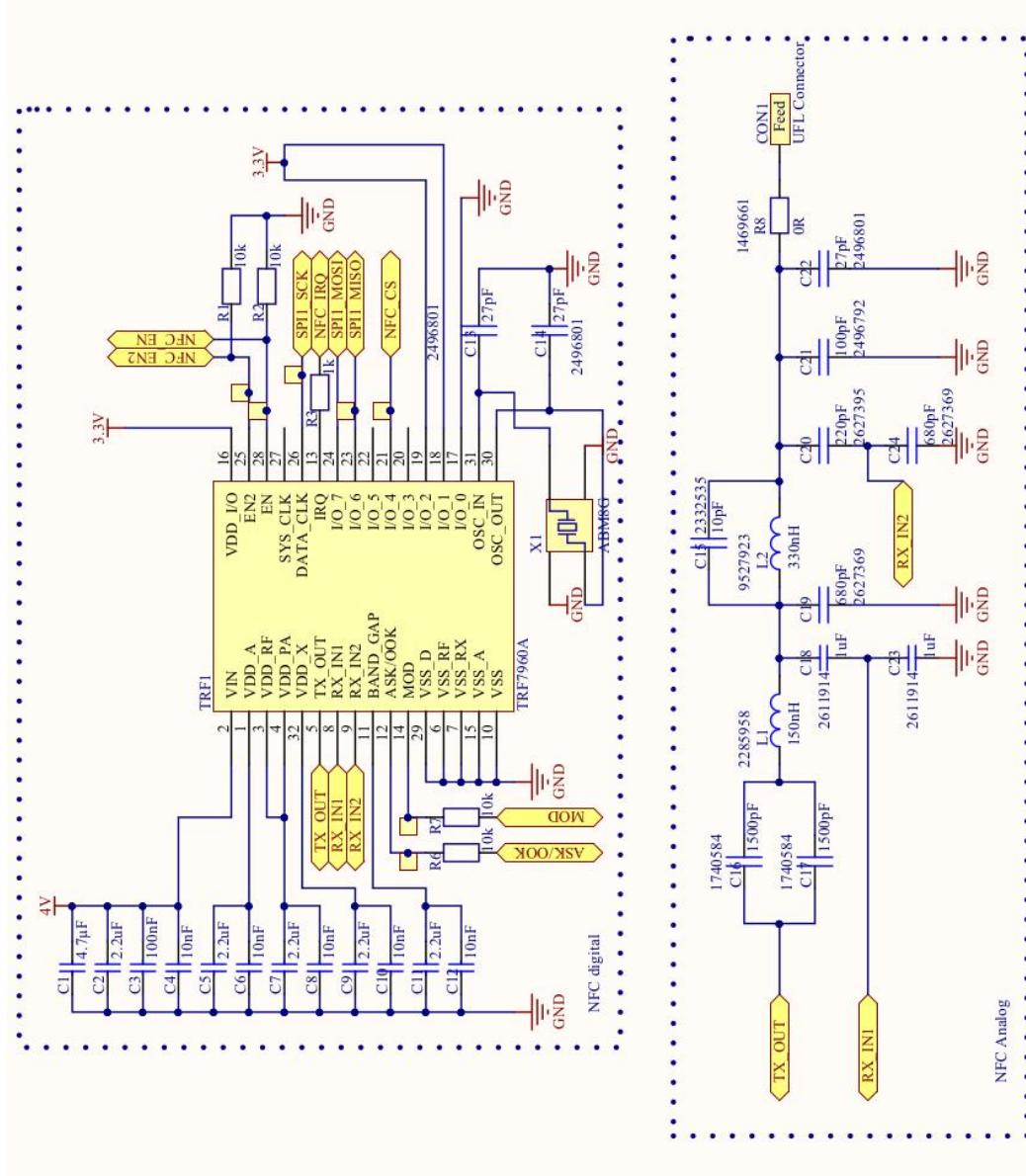
Rysunek 3.1: Schemat modułu zasilania urządzenia lokalizującego.

Źródło: Twórczość własna



Rysunek 3.2: Schemat modułu funkcjonalnego urządzenia lokalizującego.

Źródło: Twórczość własna



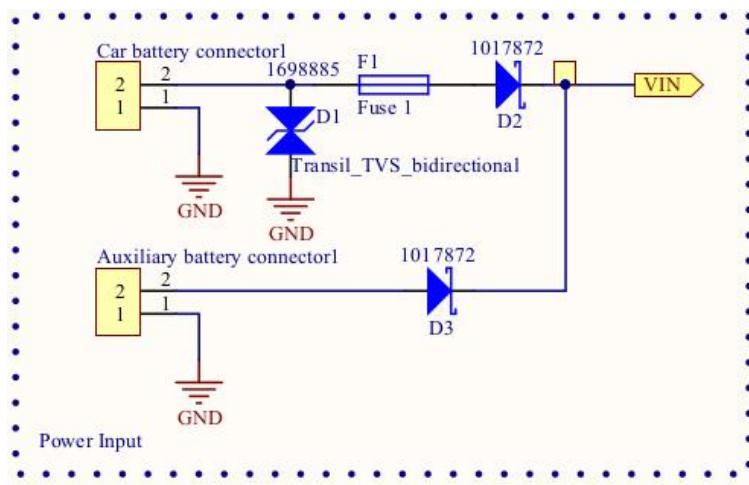
Rysunek 3.3: Schemat modułu NFC urządzenia lokalizującego.

Źródło: Twórczość własna

### 3.1.1 Schemat zasilania

Akumulator samochodowy jest bardzo wygodnym źródłem zasilania układów elektronicznych, lecz gdy są one niewłaściwie zaprojektowane, bywa on dla nich zabójczy. Bliska odległość do alternatora i innych urządzeń indukcyjnych powoduje generowanie silnych zakłóceń na linii zasilającej. Niekiedy "szpilki" napięciowe osiągają wartość rzędu 100V. Z tego powodu należy stosować transile – diody zabezpieczające. Powodują one ograniczenie zbyt dużego napięcia do pewnej maksymalnej wartości. W przypadku zastosowanego przeze mnie komponentu wynosi

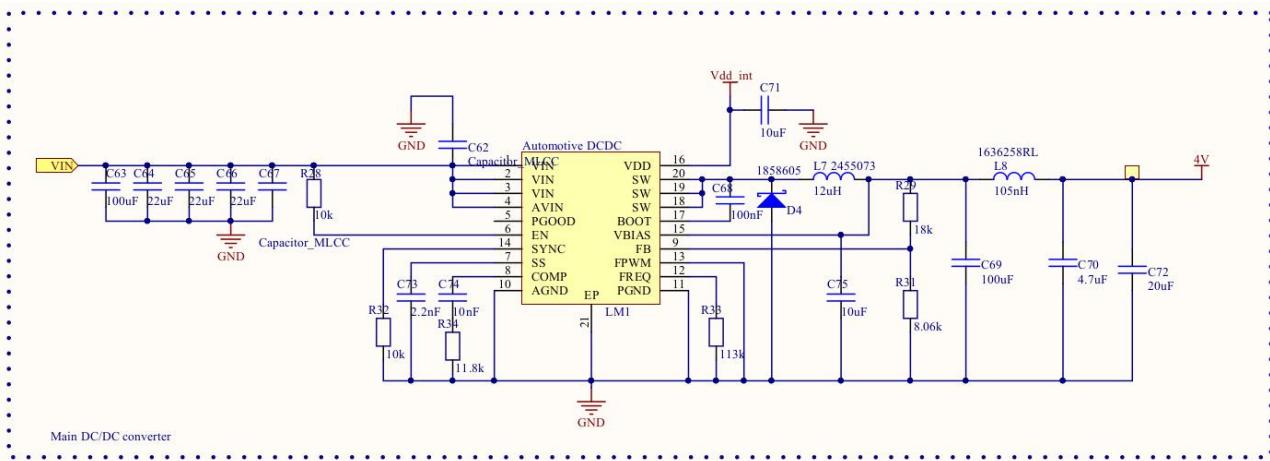
si ono 24.4V. Zabezpieczenie przeciążeniowe stanowi bezpiecznik samochodowy o wartości 4A. Ponieważ jednym z wymagań układu jest możliwość zasilania baterii samochodowej, konieczne jest zastosowanie dodatkowego przyłącza zasilania. Urządzenie można zasilić dowolną baterią o napięciu od 4 do 38V i wydajności prądowej co najmniej 3A w szczytowym. Ze względu na prawdopodobieństwo wystąpienia różnic napięć pomiędzy dodatkową baterią, a akumulatorem samochodu i wiążącym się z tym przepływem prądu z jednego źródła do drugiego, konieczne jest zastosowanie diód zabezpieczających przed rozładowaniem baterii przez akumulator (gdy napięcie akumulatora niższe niż napięcie baterii) lub mogącym doprowadzić baterię do zniszczenia doładowywaniem jej bezpośrednio z akumulatora (gdy napięcie baterii jest od niższe napięcia akumulatora). W trakcie projektowania, zdecydowano się na zastosowanie diód Schottky'ego ze względu na ich niski spadek napięcia (0.2 - 0.55V w zależności od natężenia prądu) oraz szybki czas przełączania ze stanu zaporowego do przewodzenia (ograniczenie krótkotrwałych zaników zasilania przy wyłączaniu samochodu). Na rysunku 3.4 przedstawiono część wejściową dla zasilania całej płytki.



Rysunek 3.4: Schemat modułu zasilania wejściowego urządzenia lokalizującego.

Źródło: Twórczość własna

Niestety, często napięcie wejściowe, nawet po zadziałaniu zabezpieczenia w postaci transila, jest nadal zbyt duże dla zwykłych układów zasilających. Stąd konieczne jest stosowanie przetwornic impulsowych klasy automotive, które umożliwiają zasilanie napięciem wejściowym do kilkudziesięciu woltów. Schemat wykorzystanej przetwornicy przedstawiono na rysunku 3.5.

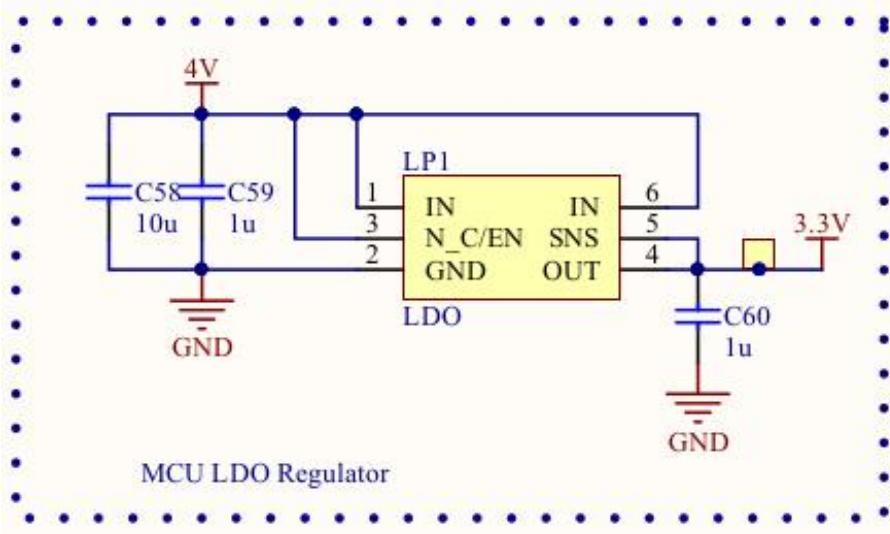


Rysunek 3.5: Schemat przetwornicy impulsowej modułu zasilania urządzenia lokalizującego.

Źródło: Twórczość własna

Zastosowana w urządzeniu przetwornica umożliwia zasilanie napięciami od 4 do 38V. Wybrano ją ze względu na niewielką liczbę zewnętrznych komponentów, niezbędnych do jej działania w porównaniu do innych modułów, a także wysoką sprawność rzędu od 85% do 90% w zależności od chwilowego natężenia prądu. Wytwarza ona na wyjściu napięcie o wartości 4V, którym zasilany jest moduł GSM oraz dalszy stopień obniżania napięcia.

Ostatni stopień zasilania generuje z napięcia wyjściowego z przetwornicy napięcie o wartości 3.3V. Jest ono niezbędne do zasilania układów mikrokontrolera, pamięci flash, akcelerometru oraz układu GPS. Szacowany pobór prądu przez te układy wynosi ok. 200mA w szczytce, stąd dla bezpieczeństwa wykorzystano stabilizator napięcia LDO (ang. Low Dropout Stabilizer) o maksymalnym natężeniu wyjściowym 0,5A. Jego schemat przedstawiono na rysunku 3.6.

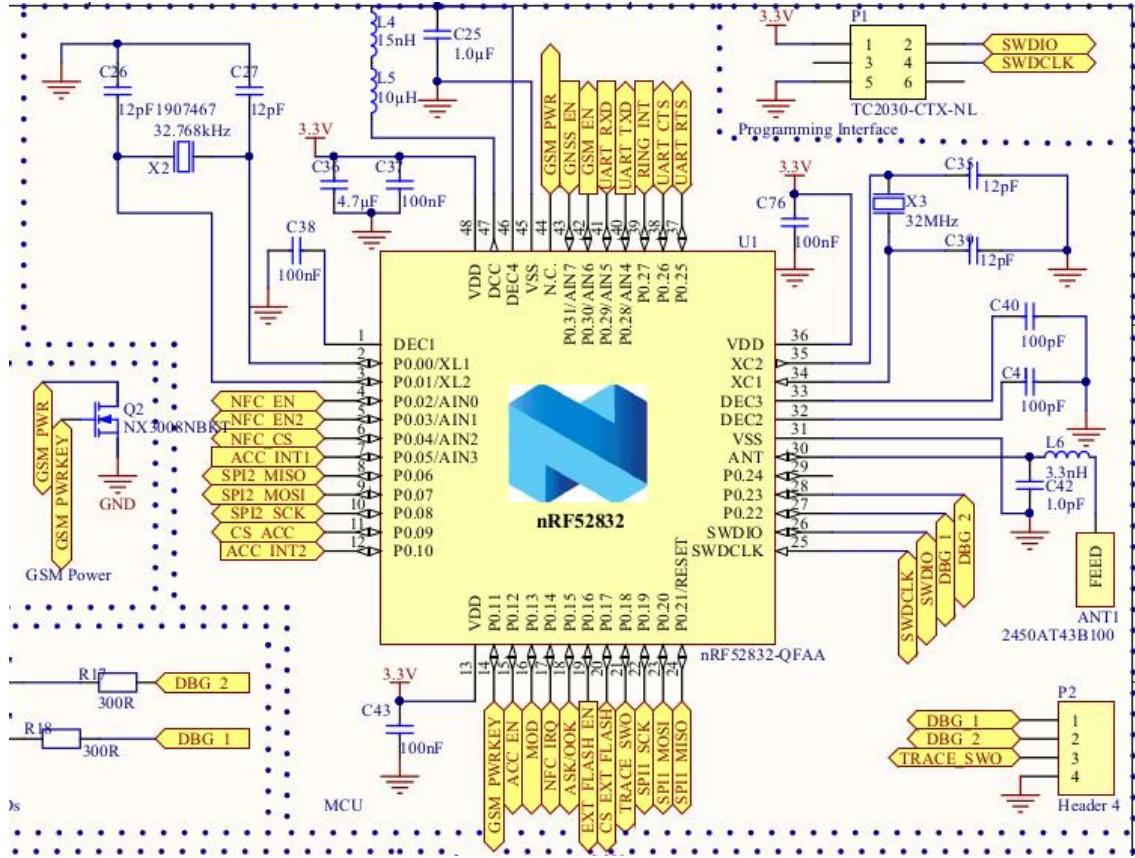


Rysunek 3.6: Schemat stabilizatora napięcia modułu zasilania urządzenia lokalizującego.

Źródło: Twórczość własna

### 3.1.2 Moduł mikrokontrolera

Serce urządzenia stanowi mikrokontroler nRF52832 firmy Nordic Semiconductor. Układ ten posiada 32 bitowy rdzeń Cortex-M4 zaprojektowany przez firmę ARM, sprzętową jednostkę FPU, 512kB wewnętrznej pamięci Flash oraz 64kB pamięci RAM. Zdecydowano się na wykorzystanie tego mikrokontrolera ze względu na kilka czynników. Pierwszym z nich jest jego wyposażenie - posiada wbudowany układ radiowy działający na częstotliwości 2.4 GHz i umożliwiający komunikację w standardzie Bluetooth Low Energy, ANT lub wykorzystanie własnego protokołu. Dodatkowym atutem tego mikrokontrolera jest wyposażenie w sprzętowy interfejs NFC, umożliwiający wykorzystanie modułu jako tag (urządzenie podrzędne) w komunikacji poprzez interfejs NFC. Ponadto ma bardzo duże możliwości obliczeniowe – wewnętrzny zegar 64 MHz umożliwia bardzo szybkie wykonywanie zaprogramowanych zadań i szybki powrót do trybu oszczędzania energii. Zużycie energii przez ten procesor jest bardzo niewielkie. W trakcie wykonywania programu pobór prądu wynosi  $58 \mu A /MHz$  gdy kod wykonywany jest z pamięci flash, natomiast w trybie oszczędzania energii pobór spada do ok  $1.9 \mu A$ . Ostatnim i być może najważniejszym czynnikiem decydującym na wybór tego układu jest posiadanego przez autora doświadczenie zawodowe w programowaniu układów od tego producenta, a zatem bardzo dobra znajomość jego możliwości i SDK (*ang. Software Development Kit*). Schemat mikrokontrolera przedstawiono na rysunku 3.7.



Rysunek 3.7: Schemat modułu mikrokontrolera w urządzeniu lokalizującym.

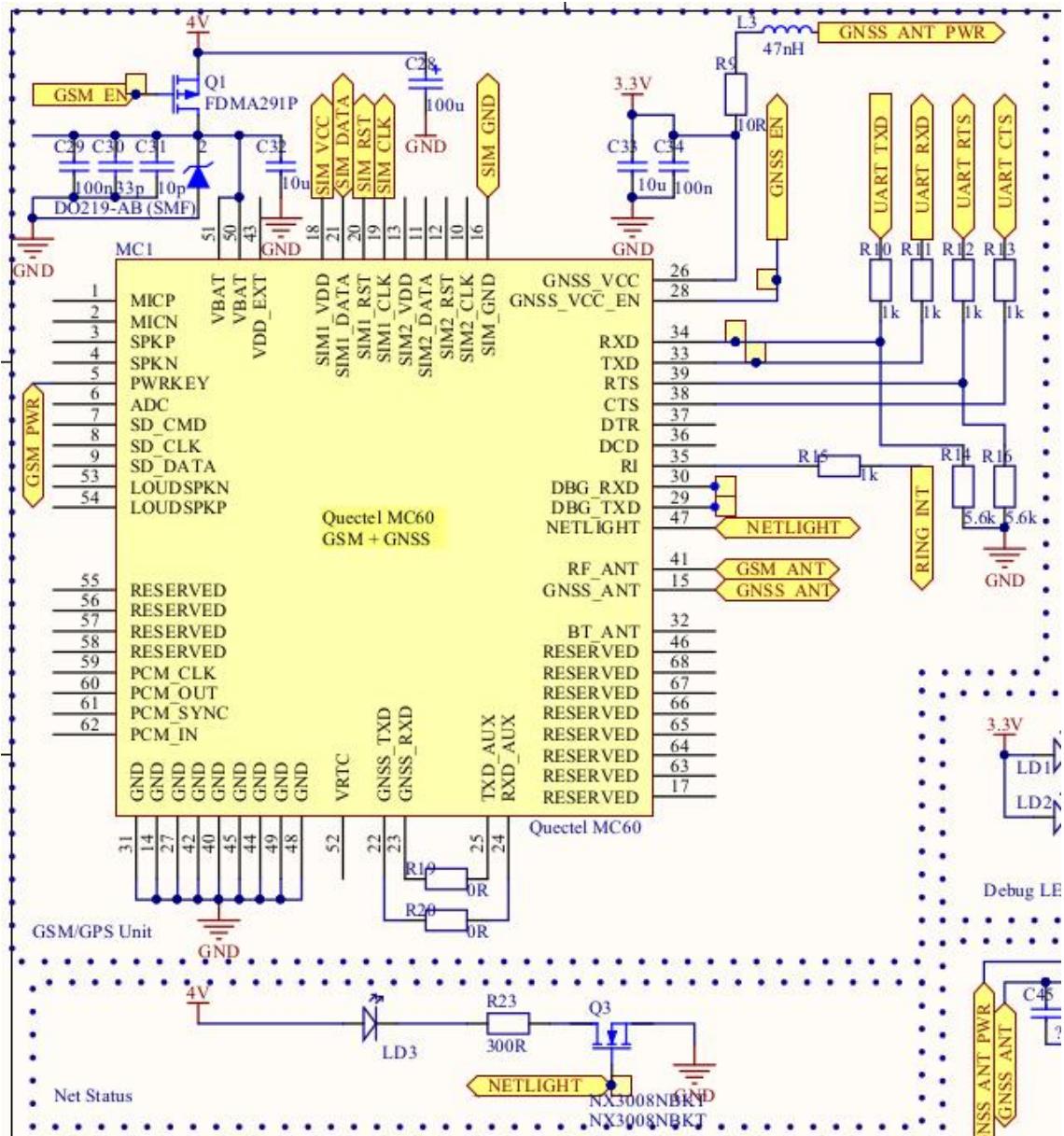
Źródło: Twórczość własna

### 3.1.3 Moduł GSM i GPS

Jako moduł realizujący główną funkcję urządzenia wybrano układ Quectel MC60. Stanowi on połączenie modułu GSM oraz GPS w jednym chipie. Umożliwia transmisję w wielu protokołach, takich jak: TCP/IP, UDP, FTP, PPP, HTTP czy NTP. Ponadto możliwe jest odbieranie i wysyłanie danych w postaci krótkich wiadomości SMS. Układ posiada niewielkie wymiary: 18.7 mm x 16 mm x 2.1 mm dzięki czemu możliwe będzie zmniejszenie całego urządzenia. Zużycie energii wynosi:

- Około 25 mA gdy działa jedynie moduł GPS
  - Do 1.6 A w trakcie transmisji danych poprzez sieć GSM

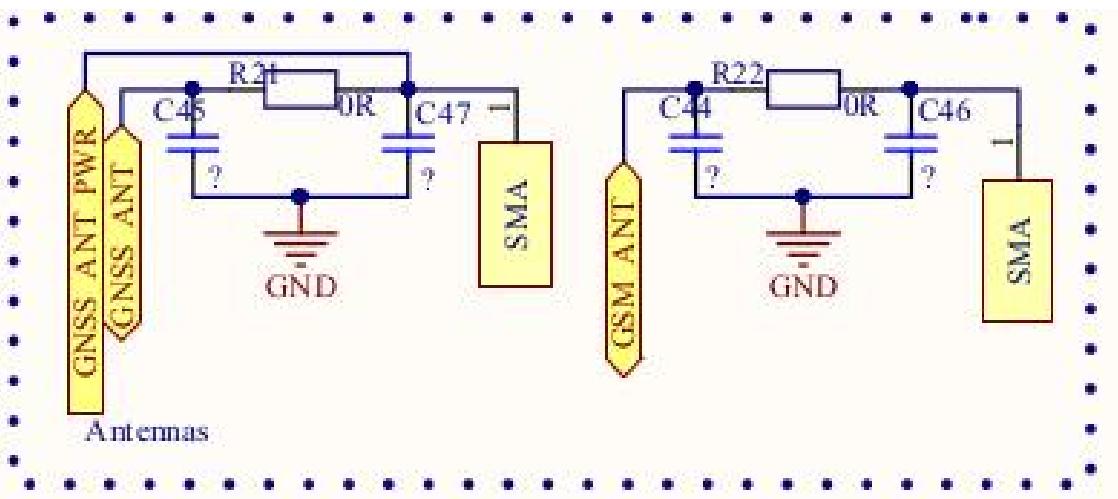
Ponadto, kombinacja tych dwóch systemów umożliwia wykorzystanie funkcjonalności AGPS. Polega ona na podaniu do modułu GPS zgrubnych danych o położeniu satelitów, pobranych z sieci GSM. Dzięki temu, ustalenie własnej lokalizacji, nawet po długotrwałym wyłączeniu, trwa ok. sekundy (tzw. warm start). Schemat modułu GSM i GPS przedstawiono na rysunku 3.8.



Rysunek 3.8: Schemat modułu układu GSM i GPS w urządzeniu lokalizującym.

Źródło: Twórczość własna

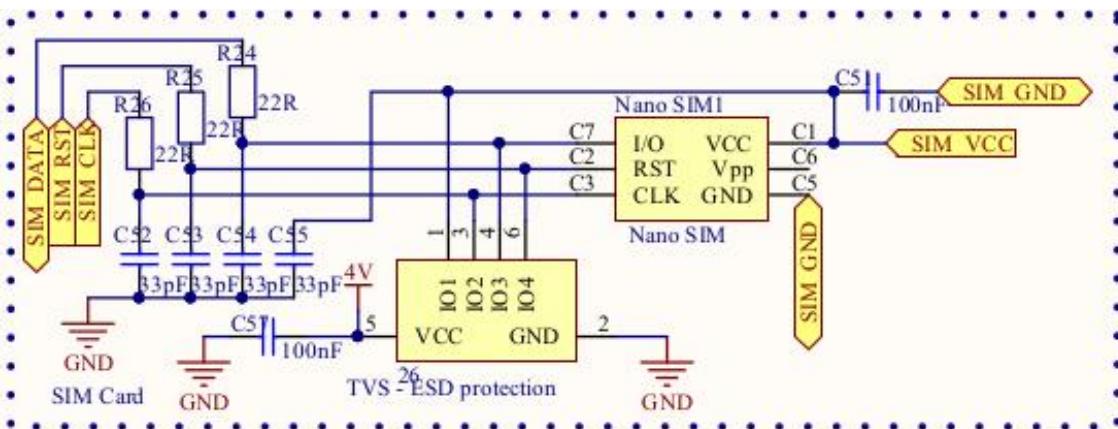
W celu zwiększenia niezawodności działania urządzenia, zdecydowano zastosować zewnętrzne anteny GSM i GPS poprawiające jakość sygnału. Dodatkowo, w celu dalszej jego poprawy, antena GPS jest anteną aktywną. Oznacza to, że dostarczane jest do niej dodatkowe zasilanie, co powoduje wzmacnienie odebranego sygnału. Schemat anten przedstawiono na rysunku 3.9. Zawarte na nim znaki zapytania, zamiast wartości kondensatorów oznaczają, że kondensatory należy dobrać po złożeniu płytki i przebadaniu jej pod kątem jak najlepszego dopasowania impedancji.



Rysunek 3.9: Schemat modułu anten dla GSM i GPS w urządzeniu lokalizującym.

Źródło: Twórczość własna

Ostatnią częścią układu GSM jest połączenie modułu z kartą SIM, umożliwiającą zalogowanie do sieci. Przedstawiono je na rysunku 3.10. Widać na nim układ TVS, który jest odpowiedzialny za zabezpieczenie wrażliwej elektroniki w karcie SIM przed wyładowaniami statycznymi ESD (ang. **Electrostatic discharge**).



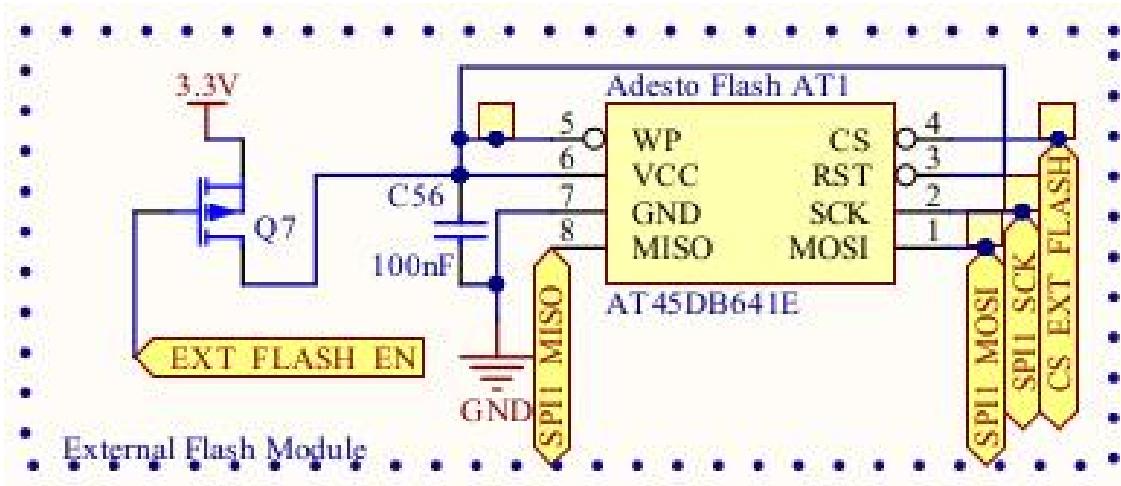
Rysunek 3.10: Schemat modułu karty SIM w urządzeniu lokalizującym.

Źródło: Twórczość własna

### 3.1.4 Moduł pamięci flash

Wewnętrzna pamięć flash mikrokontrolera jest niewystarczająca, aby przechowywać w niej trasy wraz z parametrami jazdy. Stąd też pojawia się konieczność zastosowania zewnętrznego układu pamięci nieulotnej. Zastosowana w urządzeniu pamięć flash posiada pojemność 8 MB, co umożliwia przechowywanie wielu długich tras oraz dokładne profilowanie statystyczne stylu

jazdy kierowcy. Schemat podłączenia pamięci w urządzeniu lokalizującym pokazano na rysunku 3.11.

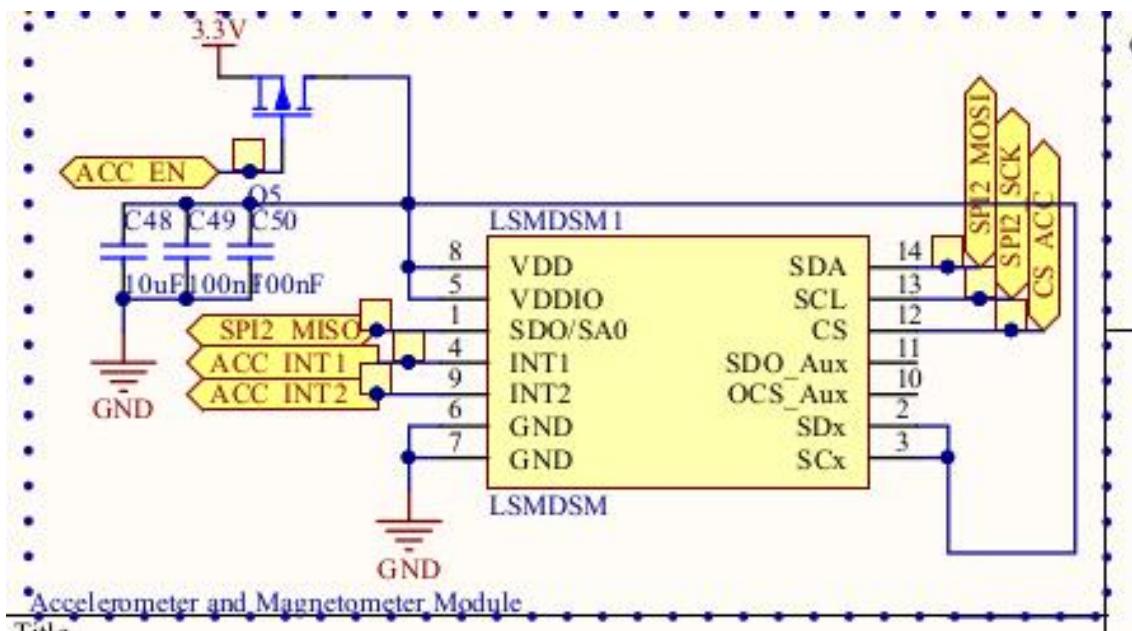


Rysunek 3.11: Schemat modułu pamięci flash w urządzeniu lokalizującym.

Źródło: Twórczość własna

### 3.1.5 Moduł akcelerometru

Kolejną ważną częścią urządzenia jest moduł akcelerometru. Pozwala on na wybudzenie urządzenia w momencie przemieszczenia pojazdu, a w razie braku dezaktywacji - uruchomienie procedury alarmowej. Ponadto, dzięki jego wskazaniom możliwe jest wyznaczenie przyspieszenia pojazdu pozwalające na profilowanie stylu prowadzenia pojazdu przez kierowcę. Wbudowany żyroskop pozwoli na dokładniejsze profilowanie stylu jazdy kierowcy w trakcie pokonywania zakrętów oraz zmiany pasa. Schemat modułu akcelerometru przedstawiono na rysunku 3.12.

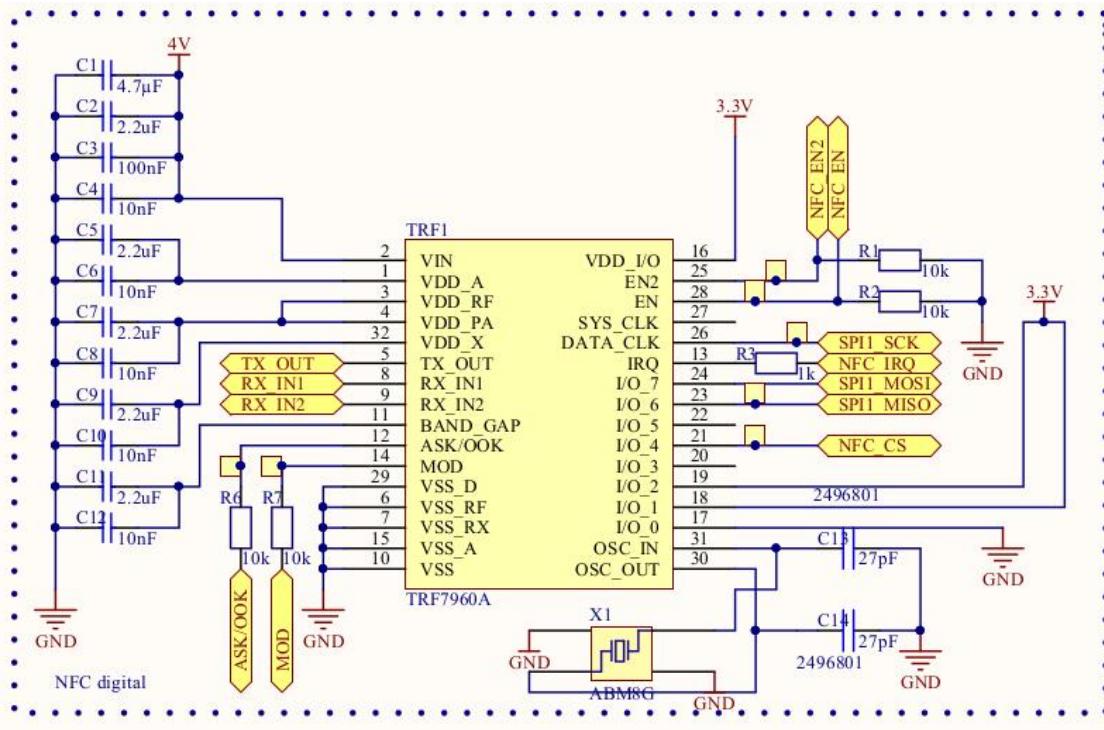


Rysunek 3.12: Schemat modułu akcelerometru w urządzeniu lokalizującym.

Źródło: Twórczość własna

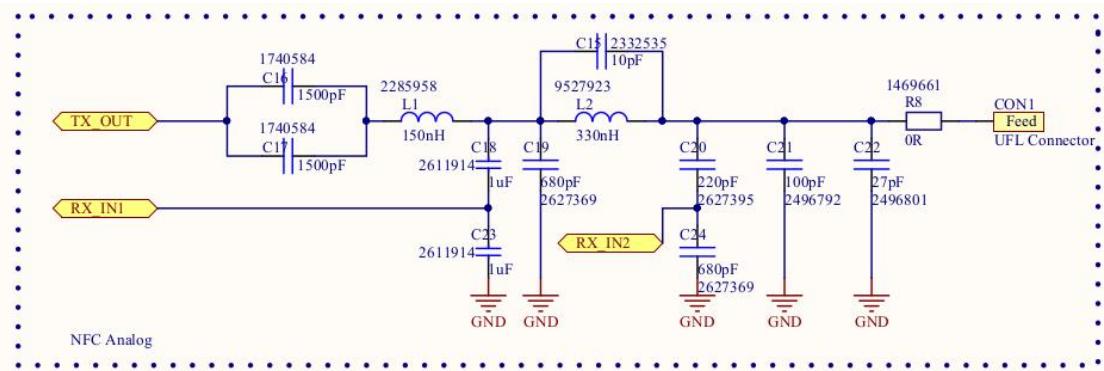
### 3.1.6 Moduł NFC

Moduł ten stanowi istotną część z punktu widzenia bezpieczeństwa komunikacji bezprzewodowej. Jest ono zapewnione poprzez zastosowanie szyfrowania wiadomości. Jeśli jednak ktoś podsłucha transmisję inicjalizacji urządzenia, w której przekazywane są klucze szyfrujące, cały koncept traci sens. Dzięki zastosowaniu modułu NFC, możliwość podsłuchania transmisji wymiany kluczy szyfrujących zostaje zniwelowana poprzez fizyczne ograniczenia zasięgu komunikacji. NFC posiada zasięg maksymalny do 10 cm. Komunikacja odbywa się pomiędzy dwoma urządzeniami. Ze względu na sposób transmisji, jedno z urządzeń inicjuje komunikację. Inicjator generuje zmienne pole magnetyczne, w którym może (lecz nie musi) zawrzeć dane wysyłane do urządzenia docelowego. Urządzenie docelowe wykrywa to pole i może odpowiedzieć poprzez odpowiednie zwiększenie go, które jest wykrywane przez inicjator. Urządzenie docelowe nie generuje żadnego pola magnetycznego. Może jedynie zwiększać pole generowane przez inicjator. Stąd wynika, że inicjator musi mieć znacznie większe zużycie energii niż urządzenie docelowe – tag. W urządzeniu lokalizacyjnym zastosowano moduł inicjatora NFC, którego schemat przedstawiono na rysunkach 3.13 - część cyfrowa oraz 3.14 - część analogowa.



Rysunek 3.13: Schemat części cyfrowej modułu NFC w urządzeniu lokalizującym.

Źródło: Twórczość własna

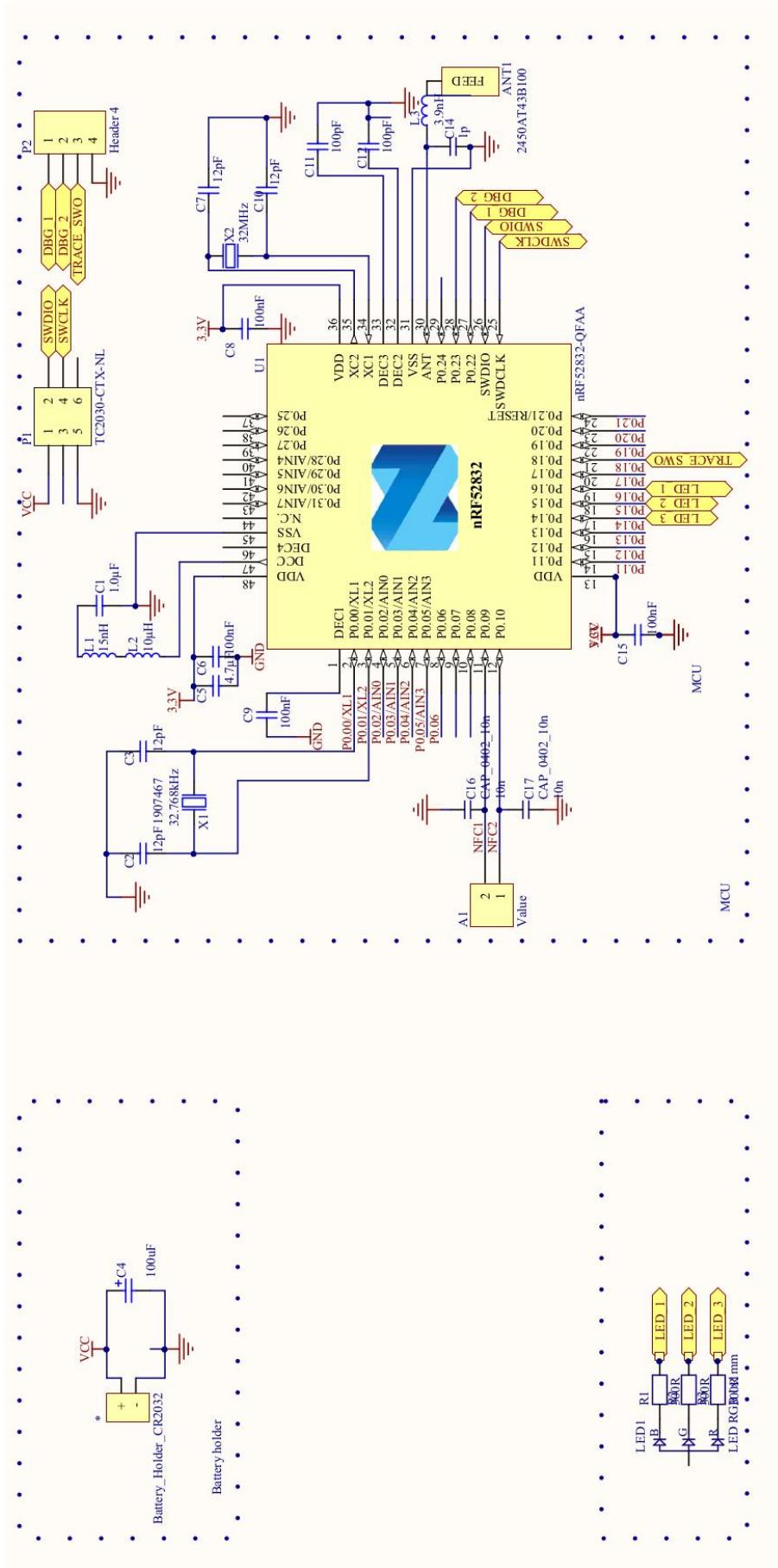


Rysunek 3.14: Schemat części analogowej modułu NFC w urządzeniu lokalizującym.

Źródło: Twórczość własna

## 3.2 Urządzenie deaktywujące

Główym zadaniem tego urządzenia jest cykliczne rozgłaszenie swej obecności. Po wykryciu przez urządzenie lokalizujące, łączy się ono z deaktywatorem oraz bezpiecznym kanałem dokonywane jest wyłączenie funkcji alarmu. Dzięki temu, że urządzenie to ma tak proste zadanie, nie pobiera ona dużo energii, więc możliwe jest zasilenie go ze standardowej baterii CR2032 o promieniu 20 mm i grubości 3.2 mm. Urządzenie to, przy odpowiedniej konfiguracji parametrów transmisji może działać kilka lat bez konieczności jej wymiany. Zastosowanie wspomnianego źródła zasilania stanowi kompromis pomiędzy czasem działania i rozmiarem urządzenia, które docelowo powinno być umieszczone przy kluczach samochodowych. Schemat deaktywatora przedstawiono na rysunku 3.3.



Rysunek 3.15: Schemat modułu zasilania urządzenia deaktywującego.

Źródło: Twórczość własna

# Rozdział 4

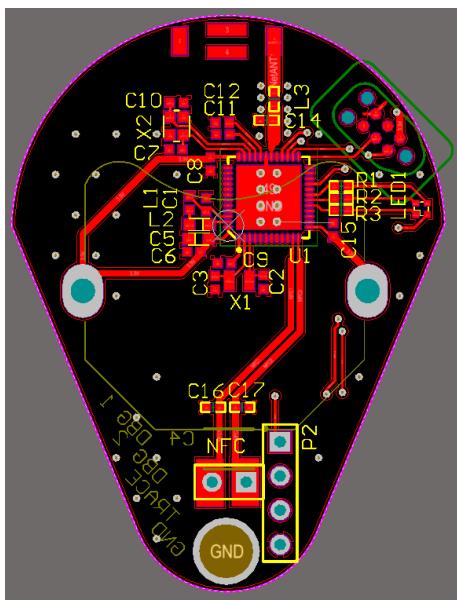
## Schematy płytEK drukowanych

### 4.1 Urządzenie deaktywujące

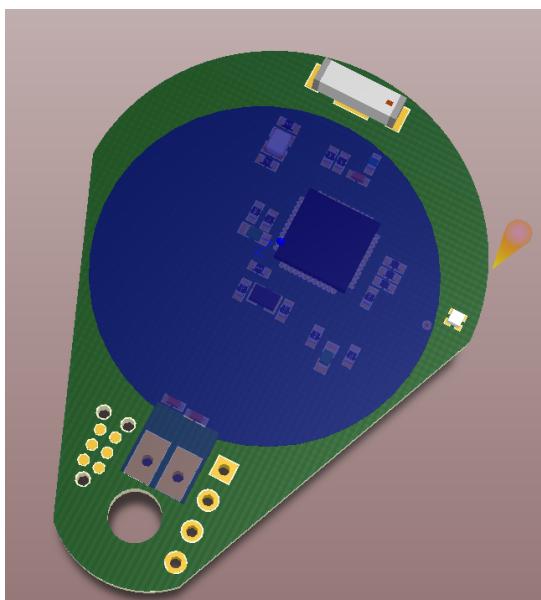
Ze względu na pełniony przez urządzenie cel, powinno ono zawsze towarzyszyć osobie upoważnionej do uruchomienia pojazdu. Biorąc pod uwagę przykład zastosowania urządzenia we flotach pojazdów, szybko można zauważyc, że zazwyczaj do pojazdu nie jest przypisana jedna osoba, lecz może być on używany przez wielu kierowców. Stąd też logiczny staje się wniosek, że urządzenie nie może być przyporządkowane do kierowcy, lecz do pojazdu. Idealnym rozwiązaniem wydaje się umieszczenie go przy kluczykach lub karcie umożliwiającej uruchomienie pojazdu bezkluczykowo, jako dodatkowy brelok. Z tego powodu ważne stają się wymiary samego urządzenia. Nie powinno być ono zbyt grube, aby nie przeszkadzało w kieszeni, ani zbyt duże, aby nie obijało się o nogi, a tym samym nie rozpraszało kierującego w trakcie jazdy. Rozmiar płytki urządzenia dezaktywującego wynosił odpowiednio 32 mm x 43 mm szerokości i wysokości.

Ze względu na prostotę urządzenia, składa się ono z bardzo niewielu modułów. Na górnej warstwie płytka znajduje się serce układu - mikrokontroler nRF52832 wraz z anteną 2.4 GHz ISM do komunikacji poprzez Bluetooth Low Energy. Dodatkowo, znajdują się tam złącze do programowania, złącze debugowe oraz antena NFC, zwizualizowana jako koło w kolorze niebieskim. Górną warstwę płytki przedstawiono na rysunku 4.1.

Centralne miejsce na dolnej warstwie płytki zajmuje bateria litowa CR2032, która zapewni kilkuletnią pracę dezaktywatora. Posiada ona średnicę 20mm oraz grubość 3.2 mm co stanowi idealny kompromis pomiędzy wymiarami urządzenia, a czasem jego pracy, bowiem mniejsze baterie oferują mniejszą pojemność liczoną w miliampero godzinach. Wygląd oraz wizualizację dolnej warstwy płytki przedstawiono na rysunku 4.2.



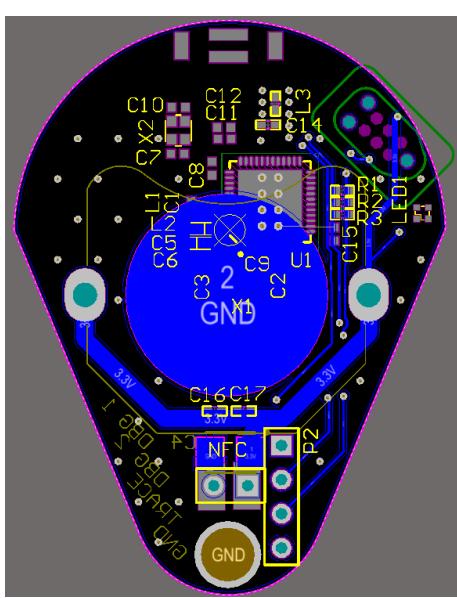
(a) Wygląd górnej warstwy płytki



(b) Wizualizacja górnej warstwy płytki

Rysunek 4.1: Wygląd górnej warstwy płytki urządzenia dezaktywującego oraz jej wizualizacja.

Źródło: Twórczość własna



(a) Wygląd dolnej warstwy płytki



(b) Wizualizacja dolnej warstwy płytki

Rysunek 4.2: Wygląd dolnej warstwy płytki urządzenia dezaktywującego oraz jej wizualizacja.

Źródło: Twórczość własna

## 4.2 Urządzenie lokalizujące

Płytkę lokalizującą jest znacznie bardziej skomplikowana od urządzenia dezaktywującego. Wynika to głównie z faktu, iż stanowi podstawę funkcjonalności całego systemu, a zatem posiada wiele modułów realizujących określone zadania. Płytkę ma wymiary 50 mm x 50 mm, co powinno umożliwić ukrycie jej w większości miejsc w pojeździe (na przykład pod plastikowymi zabudowami kokpitu).

Wygląd i wizualizacje urządzenia przedstawiono na rysunkach 4.4 oraz 4.5.

Ze względu na użycie kilku układów radiowych, wykorzystujących częstotliwości od 13.56 MHz (NFC), poprzez 900 MHz/ 1800 MHz (GSM) i 1575.42 MHz (GPS) aż po 2.4 GHz (Bluetooth), a także przetwornicy impulsowej o znacznym szczytowym natężeniu prądu (aż do 2.5 A), niezbędne jest odpowiednie rozłożenie elementów na płytce, które zminimalizowałoby ich wzajemny wpływ. W związku z tym, postanowiono umieścić kluczowe elementy zasilające oraz radiowe w rogach płytki, aby zmaksymalizować wzajemne odległości. W ten sposób, w lewym górnym rogu płytki umieszczono złącza zasilania wejściowego, w prawym górnym - cewkę indukcyjną, stanowiącą główny element impulsowej stabilizacji napięcia. Cewka ta przy okazji stanowi główne źródło zakłóceń sygnałów. W lewym dolnym rogu znajduje się antena Bluetooth Low Energy, natomiast w prawym dolnym rogu - złącze anteny GPS.

Przewody anten GPS oraz GSM są dodatkowo ekranowane, dzięki czemu znacznie zmniejszona jest podatność tych sygnałów na zakłócenia w trakcie przepływu od anteny do płytki. Jednakże na samej płytce sygnały te nie posiadają ekranu elektromagnetycznego, przez co są podatne na szумy. Z tego względu niezbędna jest minimalizacja długości ścieżek między złączem anteny oraz wejściami układów. W dodatku, sygnał GPS stanowi najsłabszy ze wszystkich sygnałów radiowych, wykorzystywanych w urządzeniu, przez co niezbędne staje się jak największe oddalenie toru GPS od pozostałych układów. Ze względu na fakt, iż sygnał GSM jest znacznie mocniejszy, znajduje się on na środku prawego boku płytki, bliżej cewki indukcyjnej przetwornicy impulsowej.

Wszystkie sygnały radiowe użyte w urządzeniu są sygnałami analogowymi. Są one podatne na zjawisko odbicia fali elektromagnetycznej, które polega na odbiciu sygnału na końcu przewodu, bądź ścieżki elektrycznej i nałożeniu się na sygnał pierwotny. Wprowadza to dodatkowe zakłócenia w transmisji sygnału, a spowodowane jest niedopasowaniem impedancji toru transmisyjnego. Aby zminimalizować ten efekt, należy zaprojektować ścieżki po których przesyłany jest sygnał wysokiej częstotliwości tak, aby miały odpowiednią impedancję, zgodną z impedancją anteny. Dokonuje się to poprzez dobór grubości (wynika ona z grubości warstwy miedzi, zazwyczaj  $35 \mu\text{m}$ ) oraz szerokości (wybór pod kątem optymalności zużycia miejsca na PCB) ścieżek radiowych. Na podstawie tak wyznaczonych parametrów wylicza się ich niezbędną dłuż-

gość, aby osiągnąć założoną impedancję. W przypadku sygnałów GPS, GSM oraz bluetooth wynosi ona  $50 \Omega$ . Ostateczną impedancję, uwzględniającą pojemności i indukcyjności pasożytnicze między ścieżkami, zmierzoną po złożeniu płytka można jeszcze skorygować poprzez dobór elementów w filtrach przyantenowych (filtry: C45, R21, C47 oraz C44, R22, C46). Dodatkowo, ścieżki wysokiej częstotliwości prowadzi się łagodnymi łukami, bez ostrych załamań, które mogłyby zwiększać pojemność, a tym samym mogącą zmienić impedancję.

W dodatku, w trakcie projektowania urządzenia należy pamiętać o wysokim chwilowym poborze prądu (aż do około 2.5 A). Z tego względu, trzeba zaprojektować odpowiednio grube ścieżki zasilające. Jest to wymagane z dwóch powodów. Pierwszym z nich jest fakt oporności ścieżki.

$$R = \rho \cdot l / S \quad (4.1)$$

gdzie:

R - oporność ścieżki,

$\rho$  - oporność właściwa materiału, z którego wykonano ścieżkę,

l - długość ścieżki,

S - powierzchnia (liczona jako iloczyn grubości i szerokości) ścieżki,

Jak widać, im większa szerokość ścieżki, tym większa jej powierzchnia, a więc mniejsza rezystancja. Im mniejsza rezystancja, tym straty napięcia na samej ścieżce będą mniejsze, a tym samym mniejsze napięcie dostarczane do układów funkcjonalnych i większe straty na ciepło, generujące niepotrzebne zużycie energii.

$$U = R \cdot I \quad (4.2)$$

gdzie:

U - strata napięcie na ścieżce,

R - opór ścieżki,

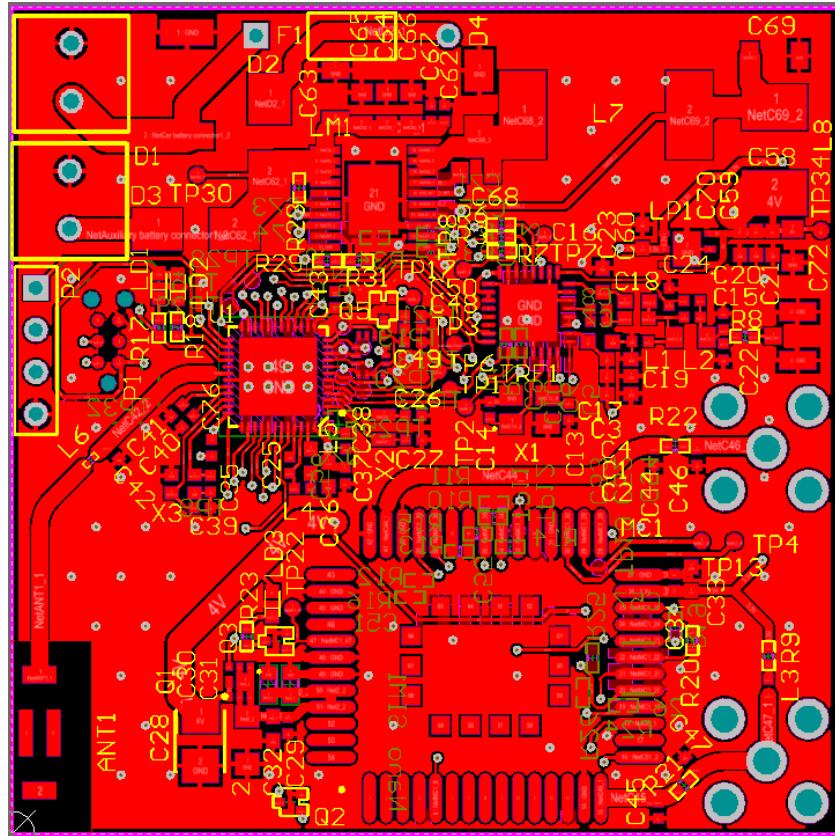
I - prąd płynący przez ścieżkę

Drugi przypadek wynika niejako z pierwszego. Gdy opór ścieżki jest zbyt duży, energia tracona na ścieżce jest tak duża, że ulega ona przepaleniu i całe urządzenie przestaje działać. Aby się przed tym ustrzec, ścieżki zasilające mają grubość 2 mm, co pozwala na przepływ około 3,5 A natężenia ciągłego prądu w temperaturze 20 stopni Celsjusza. Ze względu jednak na fakt, iż główne obciążenie urządzenia stanowi prąd chwilowy, trwający bardzo krótko, średnie natężenie prądu będzie dużo niższe od tej wartości. Tabela zestawiająca zależność między grubością ścieżek na płytce PCB od wartości maksymalnego dopuszczalnego prądu ciągłego, przepływającego przez nią, przedstawiono na rysunku 4.3.

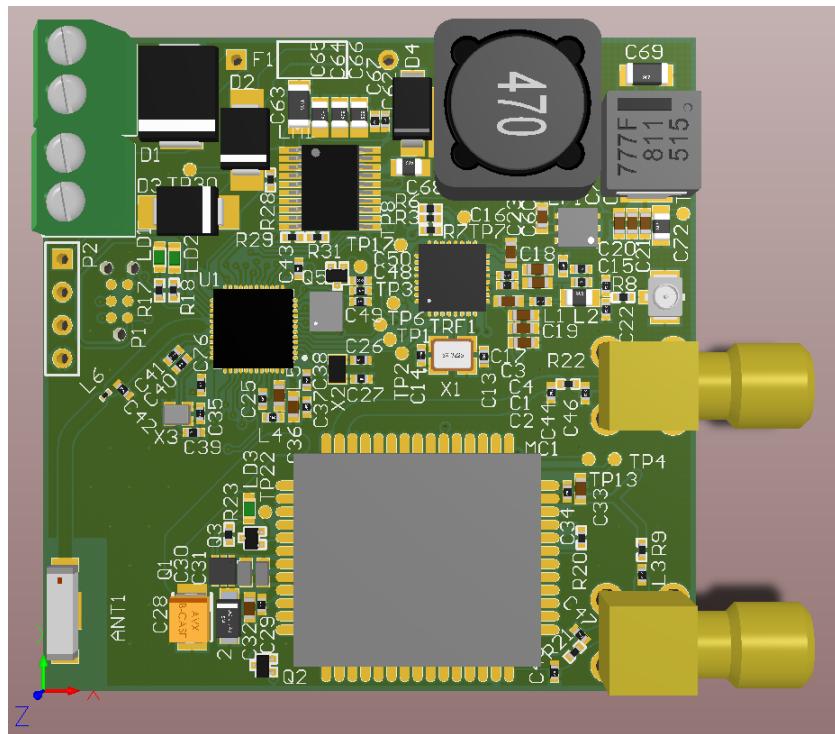
Szerokość ścieżki	Dopuszczalny prąd		
	$\Delta T=20^\circ\text{C}$	$\Delta T=80^\circ\text{C}$	prąd niszczący
0,5mm (20mil)	1,5A	3,5A	6A
1mm (40mil)	2,5A	5A	8A
2mm (80mil)	3,5A	7A	12A
3mm (120mil)	5A	10A	18A

*Uwaga! dotyczy typowej płytki drukowanej o grubości miedzi 0,035...0,038mm*

Rysunek 4.3: Tabela opisująca korelację między grubością ścieżek, a maksymalnym dopuszczalnym natężeniem prądu. Źródło: [18]



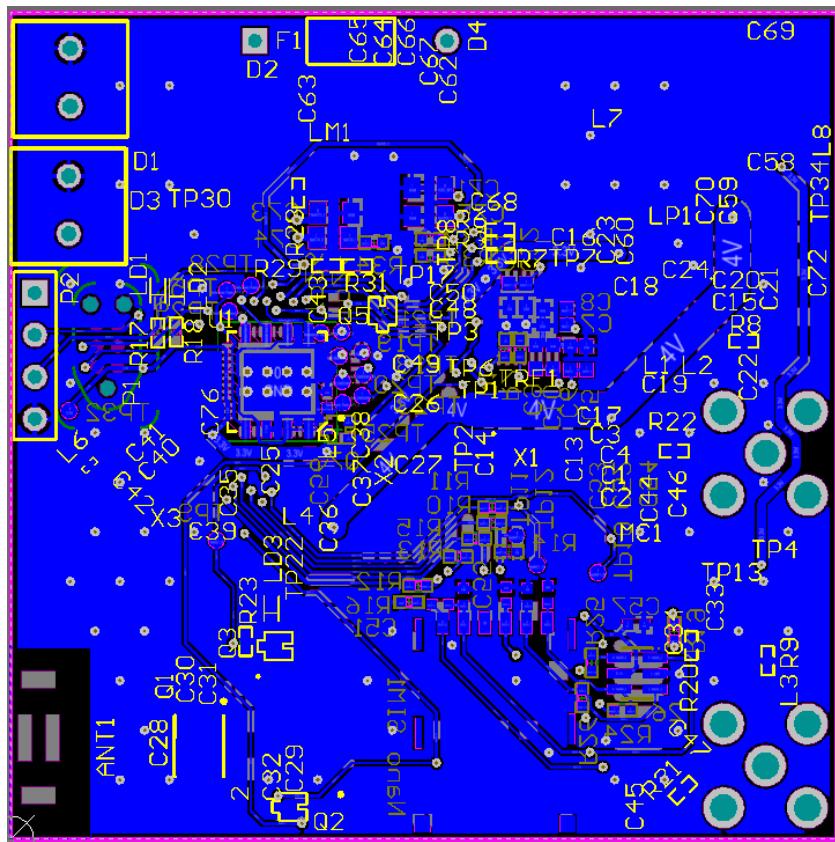
(a) Wygląd górnej warstwy płytki



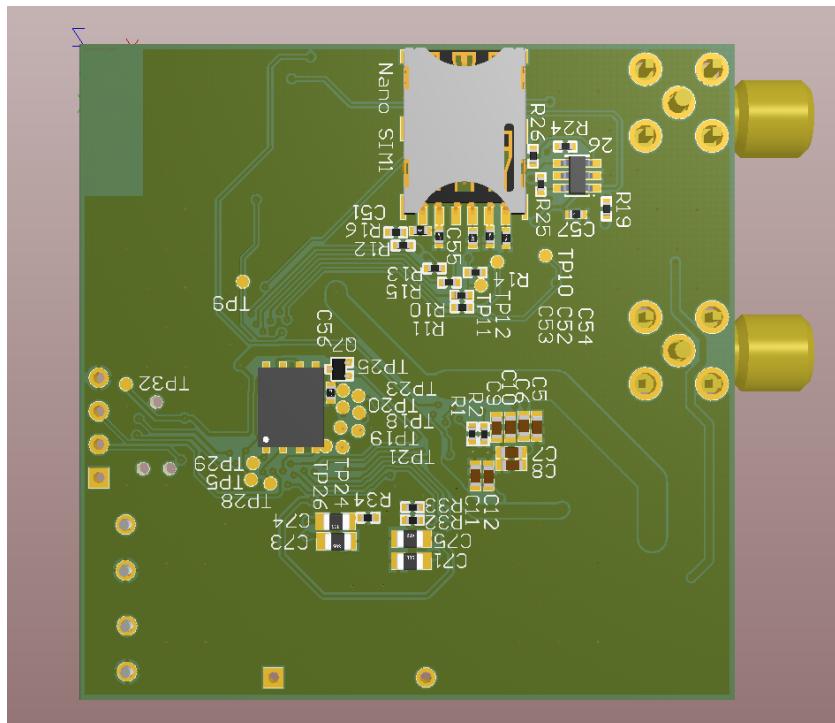
(b) Wizualizacja górnej warstwy płytki

Rysunek 4.4: Wygląd górnej warstwy płytki urządzenia lokalizującego oraz jej wizualizacja.

*Źródło: Twórczość własna*



(a) Wygląd dolnej warstwy płytki



(b) Wizualizacja dolnej warstwy płytki

Rysunek 4.5: Wygląd dolnej warstwy płytki urządzenia lokalizującego oraz jej wizualizacja.

Źródło: Twórczość własna

# Rozdział 5

## Bezpieczeństwo komunikacji

Jednym z podstawowych wymagań dotyczących tej pracy jest bezpieczna wymiana komunikatów poprzez Bluetooth Low Energy. Jest to tak kluczowe, ponieważ za pomocą tego protokołu, poprzez bezprzewodowe medium, przesyłane są kluczowe dane, zwłaszcza komendy dezaktywujące tryb alarmowy urządzenia. Transmisja fizycznie jest zawsze realizowana rozgłoszeniowo, co powoduje, że jej podsłuchanie nie jest trudnym zadaniem. Jest to niebezpieczne z dwóch powodów. Pierwszym z nich jest fakt wysyłania wrażliwych danych, jak na przykład dane lokalizujące pojazd. Dzięki nim, potencjalny złodziej mógłby po krótkiej analizie bezproblemowo określić miejsca, w których regularnie przebywa pojazd, a następnie wybrać dla niego najbardziej korzystne i przygotować się do kradzieży. Po drugie, co ważniejsze, będąc w pobliżu pojazdu w trakcie dezaktywacji trybu alarmowego, byłby w stanie podsłuchać komendę dezaktywującą, a następnie zapisać ją w celu późniejszego odtworzenia, umożliwiającego późniejszą bezproblemową kradzież pojazdu.

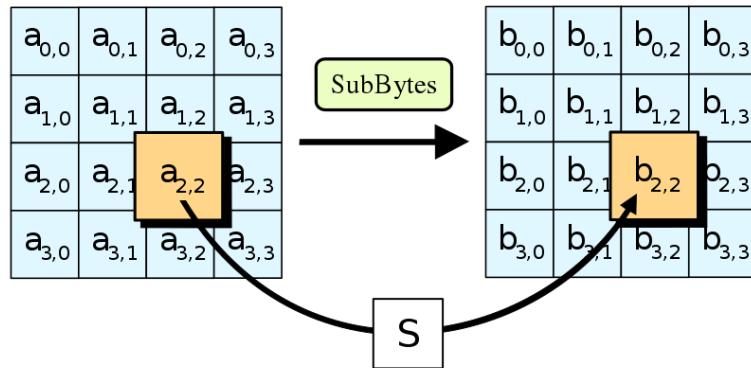
Z przytoczonych powyżej powodów komunikacja bezprzewodowa musi być szyfrowana. Jednakże operacja ta sama w sobie nie zabezpiecza tak naprawdę komendy dezaktywującej, a jedynie wrażliwe dane. Wynika to z faktu, iż w przypadku przechwycenia danych przesyłanych bezprzewodowo, dzięki szyfrowaniu są one nadal bezpieczne, ponieważ są one kompletnie niezrozumiałe. Inaczej ma się to do komendy deszyfrującej. Wynika to z faktu, że komenda ta tak naprawdę nie musi być zrozumiała dla potencjalnego złodzieja. Wystarczy, że jedynie ją odtworzy, nawet w formie zaszyfrowanej. Urządzenie wówczas ją zdeszyfruje i wykona deaktywację alarmu. Wszystko przez fakt, że komenda ta jest stała, nie zawiera elementu zmennego w czasie. W wyniku szyfrowania stałej komendy stałym kluczem szyfrującym, uzyskamy oczywiście stały i powtarzalny pakiet zaszyfrowanych danych, które mogą być bezcenne w ręku potencjalnego złodzieja. W celu zabezpieczenia się przed tym, do komunikacji należy wprowadzić element zmienności w czasie.

## 5.1 AES

Jako główny algorytm szyfrowania w niniejszej pracy wykorzystano algorytm AES (ang. Advanced Encryption Standard) w wersji ze 128-bitowym kluczem szyfrującym. Wyboru tego dokonalem, ponieważ zastosowany przezem mnie mikrokontroler nRF52832 firmy Nordic Semiconductor posiada sprzętowe wsparcie szyfrowania danych wykorzystując właśnie AES128. Algorytm ten powstał w 2001 roku w Stanach Zjednoczonych w ośrodku NIST (ang. National Institute of Standards and Technology) w wyniku prac badawczych dwóch belgijskich kryptografów - Vincenta Rijmena i Joan'a Daemen, od których nazwisk powstała oryginalna nazwa algorytmu – Rijndael. Stanowi on jeden z najpopularniejszych na świecie szyfrów symetrycznych, a o jego skuteczności stanowi fakt, że w 2002r. Został przyjęty jako federalny standard szyfrowania w Stanach Zjednoczonych. Pojęcie szyfr symetryczny oznacza, że do zaszyfrowania oraz zdeszyfrowania stosuje się ten sam klucz szyfrujący (w przeciwieństwie do algorytmów asymetrycznych, gdzie stosuje się dwa klucze, jeden do szyfrowania, a drugi do deszyfrowania). Z tego powodu, klucz szyfrujący stanowi ekstremalnie wrażliwą daną, której pod żadnym pozorem nie powinno się przesyłać poprzez ogólnie dostępne medium komunikacyjne. Wyciek klucza szyfrującego powoduje kompromitację całej komunikacji w wyniku czego przestaje ona być uznawana za bezpieczną. Proces szyfrowania składa się z kilku kroków. Pierwszym z nich jest podzielenie danych wejściowych (zwyczajowo nazywanych tekstem jawnym) na bloki o rozmiarze 128 bitów, czyli szesnasty bajtów. Każdy blok przedstawiany jest jako macierz o wymiarach 4 bajty x 4 bajty, szeregowana kolumnami. Macierze te nazywają się macierzami stanu. Następnie, na każdej z tych macierzy (bloku danych) kolejne operacje:

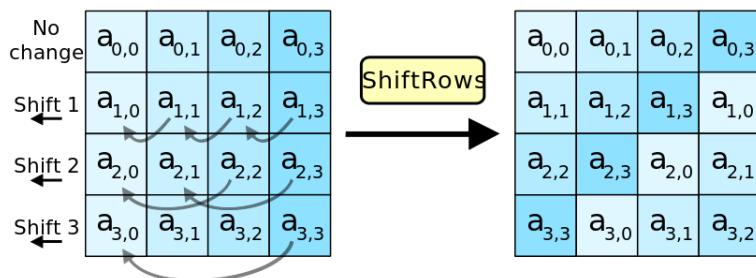
1. Utworzenie podkluczy – Etap ten polega na wygenerowaniu w sposób losowy klucza pierwotnego, a następnie na jego podstawie - po jednym podkluczu dla każdej z rund szyfrujących. Ich liczba jest uzależniona od rozmiaru klucza. Dla klucza 128-bitowego występuje 10 rund, dla klucza 192-bitowego – 12 cykli, a dla klucza 256-bitowego – 14 powtórzeń, wliczając klucz pierwotny.
2. Wykonanie rundy wstępnej (inicjującej) – Polega na wykonaniu operacji alternatywy wyłączającej – XOR (ang. Exclusive Or) dla każdego bajtu z bloku danych oraz odpowiadającego mu bajtu w kluczu pierwotnym.
3. Wykonanie rund szyfrujących – Etap ten jest wykonywany kilkukrotnie, w zależności od liczby cykli. Każda runda składa się z kilku kroków.
  - W pierwszym z nich, każdy bajt danych jest zastępowany innym bajtem pobranym w z góry zdefiniowanej tablicy (ang. lookup table) nazywanej S-Boxe'm Rijndael'a.

Operacja ta nazywa się w skrócie SB (ang. Substitution Bytes) i przedstawiono ją na rysunku 5.1. Zgodnie z zamysłem twórców, tablica ta gwarantuje nielinowość przekształcenia, a w efekcie i całego szyfrowania.



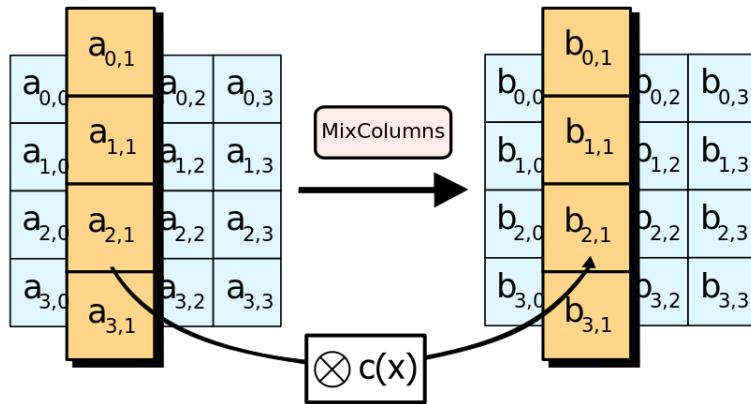
Rysunek 5.1: Wykonanie operacji Substitute Bytes. Źródło: [20]

- Kolejny krok to zamiana wierszy. Polega na przesunięciu bajtów w trzech ostatnich wierszach bloku. Pierwszy wiersz pozostaje bez zmian, w drugim wierszu bajty są przesuwane o jeden w lewo, w trzecim o dwie pozycje w lewo, a w ostatnim o 3 miejsca w tym samym kierunku. Każdy bajt, który w wyniku przesunięcia znajdzie się poza wierszem, zostaje umieszczony na jego ostatniej pozycji (wiersze w wyniku rotacji się zawijają). Operacja ta nosi miano SR (ang. Shift Rows). Przedstawiono ją na rysunku 5.2.



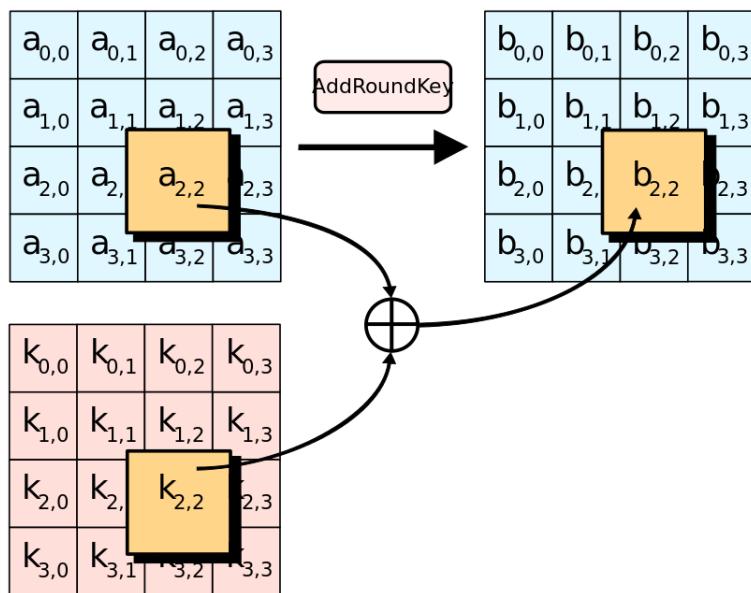
Rysunek 5.2: Wykonanie operacji Shift Rows. Źródło: [20]

- Trzecim z kolei krokiem jest operacja mieszania kolumn – MC (ang. Mix Columns). W tym etapie, każda z kolumn jest przemnażana lewostronnie przez stałą macierz o wymiarach  $4 \times 4$ , w wyniku czego powstaje kolumna z nowymi wartościami. Operacja ta przedstawiona jest na rysunku 5.3.



Rysunek 5.3: Wykonanie operacji Mix Columns. Źródło: [20]

- Ostatni krok nazywany jest AR (ang. Add Round Key) i polega na wykonaniu operacji XOR na każdym bajcie bloku danych i odpowiadającym mu bajcie w kluczu przypisany do danej rundy. Wizualizację kroku przedstawiono na rysunku 5.4.



Rysunek 5.4: Wykonanie operacji Add Round Key. Źródło: [20]

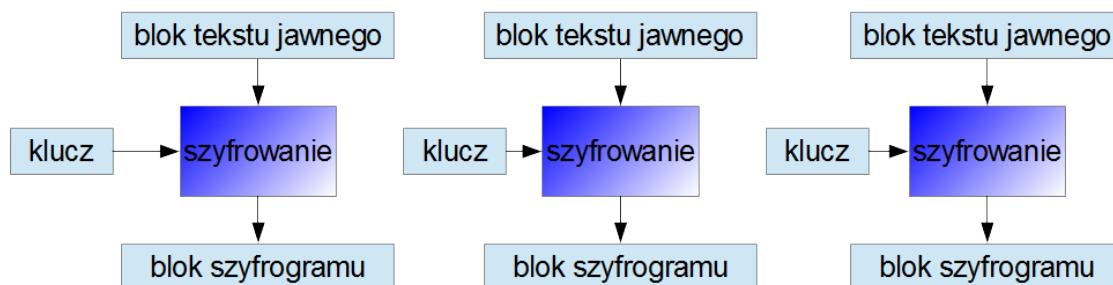
4. Ostatni etap to runda kończąca – W jej trakcie wykonywane są operacje identyczne jak w rundach szyfrujących, za wyjątkiem mnożenia kolumn, która nie występuje.

Deszyfrowanie jest operacją odwrotną do szyfrowania i polega na przekształceniu danych zaszyfrowanych na tekst jawnny. Tak samo jak w przypadku szyfrowania, tekst dzieli się na 16-bajtowe bloki. W jego trakcie wykonuje się analogiczne operacje co w przypadku szyfrowania.

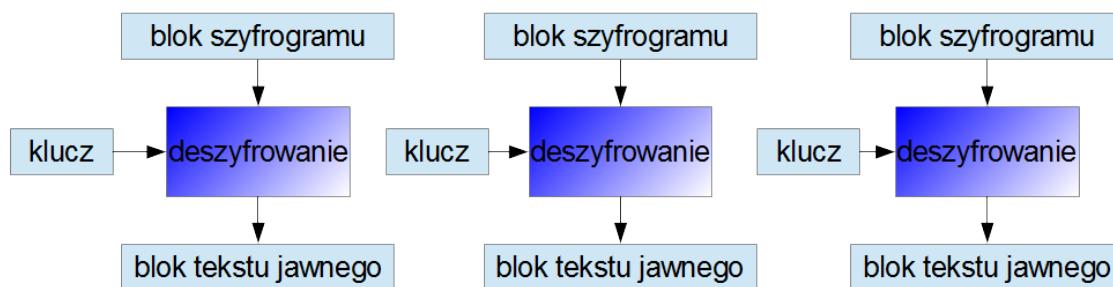
1. Odwrotne podstawianie bajtów – polega na ponownym zastosowaniu tablicy S-Box w celu podmiany bajtów.

2. Przesuwanie bajtów w wierszach w prawo. Zasada jest taka sama jak w operacji SR, zmienia się jedynie kierunek.
3. Wykonanie operacji XOR dla każdego bajtu bloku danych z odpowiadającym mu bajtem w podkluczu przypisany do danej rundy deszyfrującej. Podklucze są takie same jak w trakcie szyfrowania, lecz powinny być brane w kolejności odwrotnej (zaczynając od ostatniego, a kończąc na kluczu pierwotnym).
4. Ostatnia operacja to odwrócone mnożenie kolumn.

W efekcie uzyskujemy blok danych zdeszyfrowanych. Przedstawiony tutaj wariant algorytmu szyfrowania nosi miano ECB (ang. Electronic Codebook) i stanowi najprostrzą metodę szyfrowania. Można go przedstawić na rysunkach 5.5 oraz 5.6.



Rysunek 5.5: Operacja szyfrowania metodą ECB. Źródło: [19]

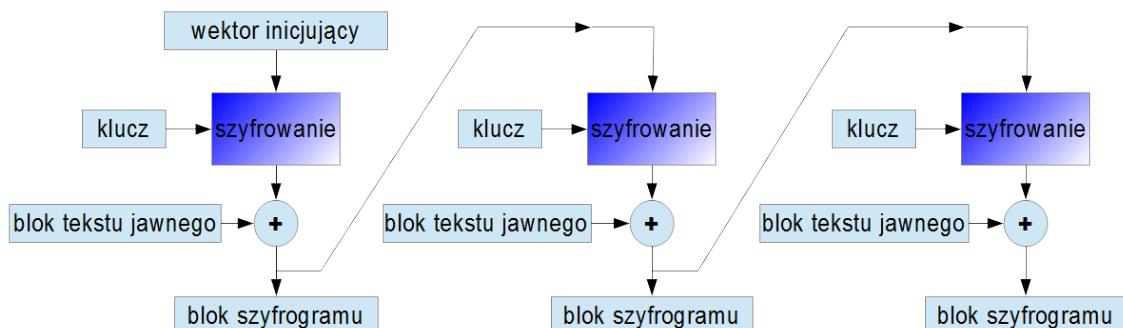


Rysunek 5.6: Operacja deszyfrowania metodą ECB. Źródło: [19]

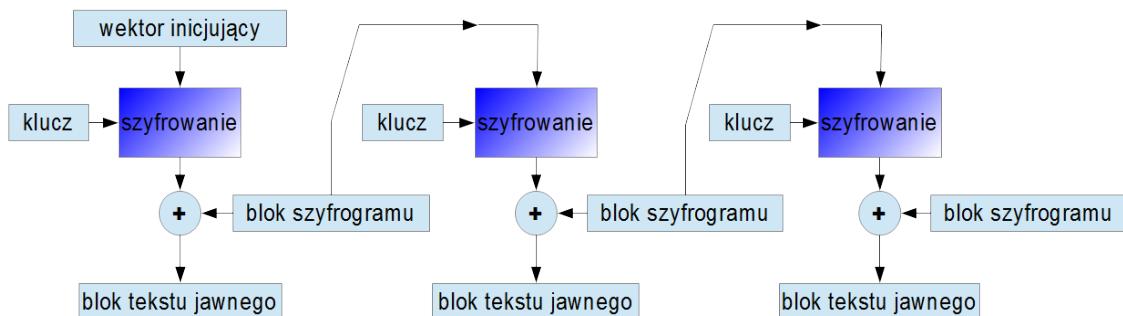
Czas trwania szyfrowania pojedynczego bloku danych o długości szesnastu bajtów na mikrokontrolerze nRF52832 wynosi około  $30 \mu\text{s}$ . Deszyfrowanie trwa zaś około  $60 \mu\text{s}$ .

## 5.2 Dodatkowe warianty szyfrowania AES

Jak zostało przedstawione wcześniej, mechanizmy szyfrowania doskonale działają w przypadku wrażliwych danych. Nie sprawdzają się natomiast w przypadku przesyłania komend, ze względu na brak zmienności pakietów w czasie i możliwości zwyczajnego odtworzenia zaszyfrowanego pakietu przez niepowołane osoby. Z tego powodu, do komunikacji należy wprowadzić element zmienności. Jednym z wariantów algorytmu AES jest tzw. CFB (ang. Cipher Feedback), przedstawiony na rysunkach 5.7 oraz 5.8. Stanowi on wysokopoziomowy algorytm, który bazuje na wariantie ECB, zmieniając jedynie logiczną strukturę informacji niezbędnych do szyfrowania. Przede wszystkim, wprowadza pojęcie wektora inicjującego (ang. initializing vector), który stanowi niezbędny dodatkowy element zmienności. Klucz główny bowiem zazwyczaj jest niezmienny na przestrzeni życia komunikujących się ze sobą urządzeń, co wynika z faktu konieczności nieupubliczniania go. Wektor inicjujący jest natomiast generowany przy każdej nowej komunikacji.



Rysunek 5.7: Operacja szyfrowania metodą CFB. Źródło: [19]



Rysunek 5.8: Operacja deszyfrowania metodą CFB. Źródło: [19]

W odróżnieniu od wariantu ECB, zamiast tekstu jawnego, szyfrowaniu ulega wektor inicjujący. Jego postać zaszyfrowana jest następnie poddawana operacji XOR z blokiem danych tekstu jawnego, a powstały w ten sposób szyfrogram stanowi nowy wektor inicjujący dla na-

stępnego bloku danych. W przypadku deszyfrowania korzysta się oczywiście z tego samego wektora inicjującego oraz klucza szyfrującego. Co ciekawe, w odróżnieniu od wariantu ECB, w metodzie CFB deszyfrowanie jest to tak naprawdę szyfrowanie. Oznacza to, że wystarczy zaimplementować jedynie mechanizm szyfrowania w algorytmie AES, aby móc zarówno szyfrować jak i deszyfrować wiadomości. Zaszyfrowany wektor inicjujący jest poddawany operacji XOR z blokiem tekstu zaszyfrowanego w efekcie czego uzyskujemy blok tekstu jawnego. Natomiast blok tekstu zaszyfrowanego stanowi wektor inicjujący dla kolejnych bloków szyfru.

### 5.3 Realizacja szyfrowania komunikacji w projekcie

W pracy zdecydowano się na wykorzystanie zarówno metod ECB oraz CFB. Pierwszym, a zarazem najbardziej podstawowym etapem jest generowanie klucza szyfrującego. Operacja ta jest realizowana przez płytę główną systemu lokalizującego. Następnie, klucz jest przekazywany po ten na żądanie użytkownika przez interfejs NFC (ang. Near Field Communication) do urządzenia deaktywującego, pełniącego rolę beacona (rozgłosni). Zastosowanie NFC jest po-wszechnie uważane za bezpieczną metodę komunikacji, ze względu na jej bardzo niską moc transmisji, a tym samym bardzo niewielki zasięg (do 5 cm). Ogranicza to zatem możliwość podsłuchania klucza szyfrującego do zera. Przy pomocy tego klucza, za każdym razem gdy płyta główna systemu połączy się z urządzeniem deaktywującym w celu uzyskania od niego komendy deaktywującej, wpierw wyśle mu zaszyfrowany, nowo wygenerowany na potrzeby danego połączenia wektor inicjalizacyjny. Umożliwi to dalszą komunikację wykorzystując wariant CFB oraz niezbędną zmienność zaszyfrowanych pakietów, praktycznie niwelującą skuteczność podsłuchiwanego transmisji.

# Rozdział 6

## Oprogramowanie

### 6.1 Urządzenie lokalizujące

Urządzenie to stanowi trzon całego systemu i to na nie spada największy ciężar funkcjonalny. Do jego zadań należą:

- Wykrywanie ruchu pojazu
- Alarmowe powiadamianie właściciela w przypadku nieautoryzowanego przemieszczenia pojazdu
- Pobieranie próbek lokalizacji, prędkości, przyspieszenia pojazdu po wykryciu ruchu i gromadzenie ich na zdalnym serwerze
- Analiza stylu jazdy kierowcy
- Wysyłanie poprzez SMS lokalizacji pojazdu na żądanie użytkownika

Duża liczba zadań do wykonania przez procesor i konieczność wywoływania ich po ścisłe określonym czasie, spowodowała, że niezbędnym stało się wprowadzenie modułu planisty. Stanowi on bardzo prostą funkcjonalność, bez możliwości wywieszczania zadań i zmiany kontekstu, więc nie wprowadza funkcjonalności wielowątkowości znanej z pełnopranych systemów operacyjnych. Jego celem jest zakolejkowanie zadań i oznaczenie ich jako do wykonania, po wymaganym czasie. Kod służący kolejkowaniu przedstawiono na listingu 6.1. Na listingu 6.2 przedstawiono funkcję, która wywyływana jest cyklicznie co 10 ms w przerwaniu od energooszczędnego timera, służącą do oznaczania zadań, które można już wykonać. Na ostatnim listingu, oznaczonym numerem 6.3, przedstawiono funkcję wykonującą zadania. Ograniczeniem funkcjonalnym jest tutaj fakt, iż żadne z zadań nie może przyjmować argumentów, ani zwracać wartości.

*Listing 6.1: Funkcja do kolejkowania zadań*

```
scheduler_error_code_e SchedulerAddOperation(void (*callback)(void),
                                              volatile uint32_t timeMsFromNow,
                                              volatile uint8_t* taskIndex,
                                              bool isCyclic){
    scheduler_entry_t entry;

    // Just safe guard not to miss the time
    if (timeMsFromNow < 2)
    {
        timeMsFromNow = 2;
    }

    for (uint8_t i=0; i< SCHEDULER_BUFFER_SIZE; ++i)
    {
        if (_scheduleBuffer[i].isInProgress == false)
        {
            entry.isInProgress = true;
            entry.isTimedOut = false;
            entry.callback = callback;
            entry.timePeriodMs = timeMsFromNow;
            entry.triggerTime = scheduler_current_time_ms + timeMsFromNow;
            entry.isCyclic = isCyclic;
            memcpy(&_scheduleBuffer[i], &entry, sizeof(scheduler_entry_t));
            if (taskIndex != NULL)
                *taskIndex = i;
            return E_SCHEDULER_OK;
        }
    }

    return E_SCHEDULER_NO_RESOURCES;
}
```

Listing 6.2: Funkcja do sprawdzania czy nie należy wykonań zadania

```
scheduler_error_code_e SchedulerCheckOperations(){
    scheduler_current_time_ms += 10;
    for (uint8_t i=0; i< SCHEDULER_BUFFER_SIZE; ++i)
    {
        if (_scheduleBuffer[i].isInProgress == true &&
            _scheduleBuffer[i].triggerTime <= scheduler_current_time_ms)
        {
            // If it is cyclic task - reschedule the next cycle
            if (_scheduleBuffer[i].isCyclic)
            {
                _scheduleBuffer[i].triggerTime = scheduler_current_time_ms
                    +_scheduleBuffer[i].timePeriodMs;
            }

            _scheduleBuffer[i].isTimedOut = true;
        }
    }

    return E_SCHEDULER_OK;
}
```

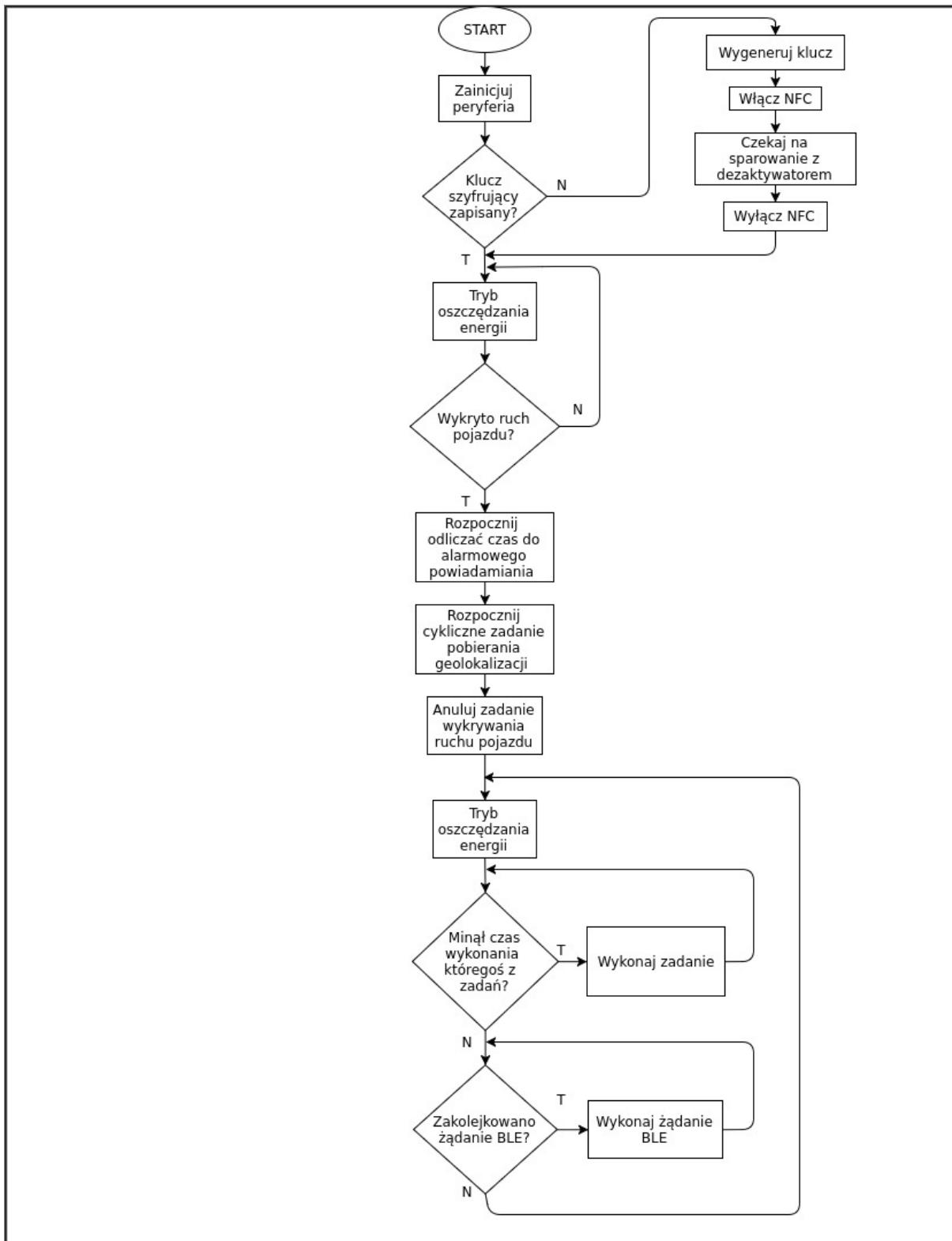
Listing 6.3: Funkcja do wykonywania zadań

```
scheduler_error_code_e ScheduleExecutePendingOperations(){
    for (uint8_t i=0; i< SCHEDULER_BUFFER_SIZE; ++i)
    {
        if (_scheduleBuffer[i].isInProgress == true &&
            _scheduleBuffer[i].isTimedOut == true)
        {
            if (_scheduleBuffer[i].isCyclic == false)
            {
                _scheduleBuffer[i].isInProgress = false;
            }

            _scheduleBuffer[i].callback();
            _scheduleBuffer[i].isTimedOut = false;
        }
    }

    return E_SCHEDULER_OK;
}
```

Główny cykl działania urządzenia został przedstawiony na rysunku 6.1.



Rysunek 6.1: Główny algorytm działania urządzenia.

Źródło: Twórczość własna

Jak widać na rysunku 6.1, po uruchomieniu i zainicjalizowaniu peryferiów i modułów na płytce, mikrokontroler dokonuje sprawdzenia czy wygenerowany został klucz szyfrujący, służący do zabezpieczenia komunikacji z dedykowanym urządzeniem deaktywującym. Jeśli klucza nie ma, oznacza to, że urządzenie nie zostało jeszcze sparowane. Wówczas, włączony zostaje układ NFC, procesor wchodzi w tryb oszczędzania energii w trakcie oczekiwania na parowanie, a całe urządzenie staje się nieoperacyjne, dopóki nie zostanie powiązane z modułem deaktywującym. Gdy zostanie to zrobione, moduł NFC zostaje wyłączony (i nie jest używany aż do momentu powrotu do ustawień fabrycznych), a mikrokontroler wchodzi w tryb oszczędzania energii w oczekiwaniu na nadchodzące zadania.

Pierwszym z nich jest wykrywanie ruchu pojazdu. Zostało to zrealizowane poprzez wykorzystanie funkcjonalności akcelerometru - wybudzenia w razie wykrycia przyspieszenia powyżej progu wynoszącego  $1.2 \frac{m}{s^2}$  i utrzymującego się przez więcej niż 1 sekundę. Jest on cykliczne, co 5 sekund, odpytywany, czy nie nastąpiło takie zdarzenie. Jeśli nie nastąpiło, cykl się powtarza. Jeśli wykryto ruch, zadanie okresowego sprawdzania ruchu jest wyłączone, natomiast do kolejki zadań ładowane są 3 najważniejsze z punktu widzenia całego systemu - zadanie alarmu, skanowania w poszukiwaniu *Key Tag'a* oraz cyklicznego pobierania próbek lokalizacji. Pierwsze z nich, stanowi zegar, który odlicza 30 sekund. Jeśli w tym czasie, alarm zostanie deaktywowany poprzez nawiązanie połączenia z urządzeniem dezaktywującym i nadanie odpowiedniego komunikatu, wówczas zadanie zostaje anulowane i jedynym cyklicznym zadaniem jest próbkowanie lokalizacji. W przeciwnym razie, uruchamiane jest zadanie cykliczne, co 10 minutowego powiadamiania właściciela o lokalizacji pojazdu poprzez wiadomości SMS. Okres ten został dobrany w ten sposób, aby nie wyczerpać za szybko środków na koncie karty SIM, użytej w module, a przy tym uzyskać rozsądną częstotliwość wysyłania SMS'ów. Zadanie to można anulować, w przypadku fałszywego alarmu, wysyłając odpowiednią komendę poprzez SMS z numeru właściciela.

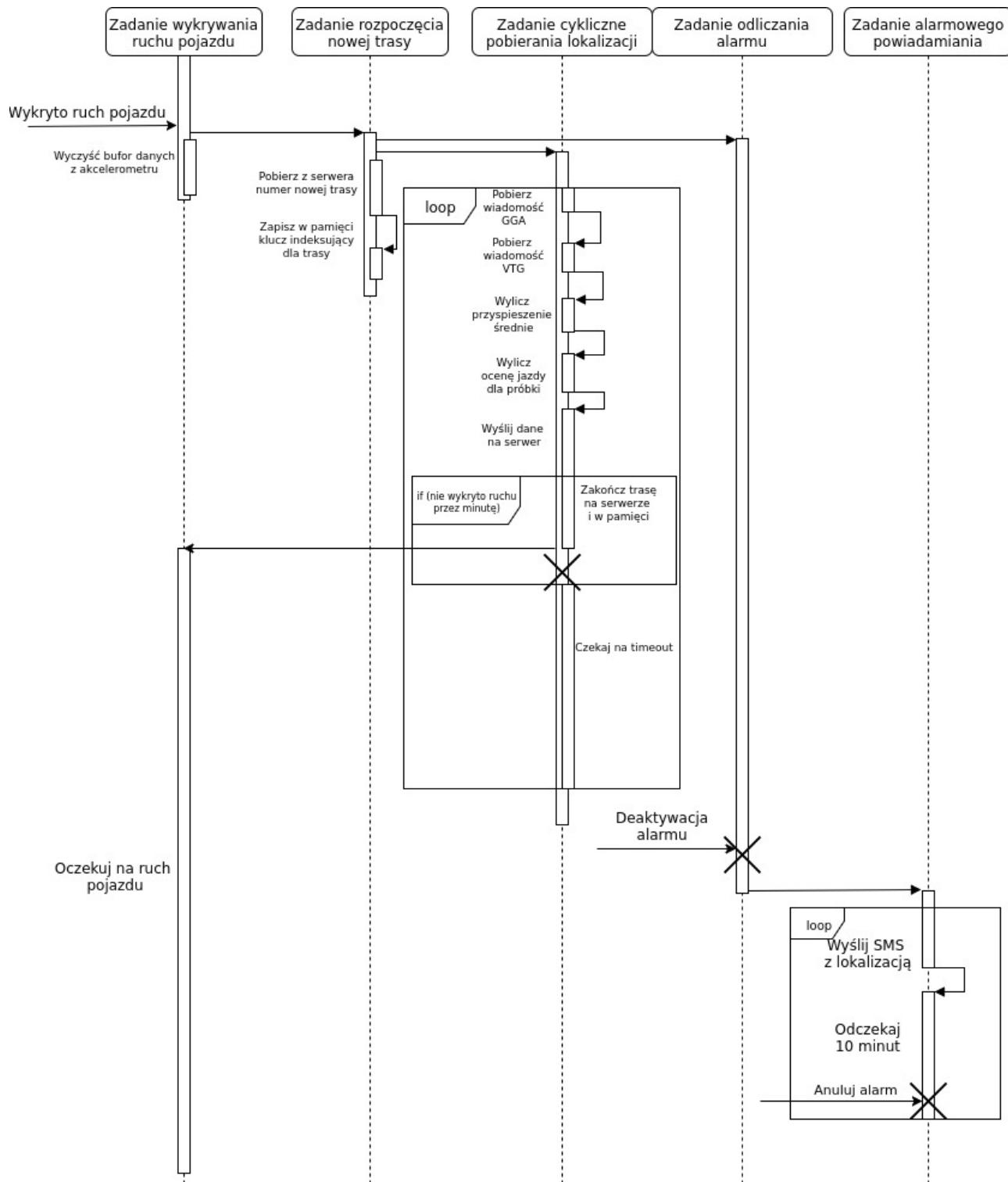
Pierwszy krok w procesie deaktywacji alarmu jest wykonywany przez płytke główną. Ze względu na fakt, iż urządzenie powinno być ukryte, nie powinno ono rozwijać żadnych pakietów w sposób ciągły, co zminimalizuje jego wykrywalność w aplikacjach skanujących poprzez Bluetooth Low Energy. Z tego powodu, w momencie wykrycia ruchu pojazdu, to ono nawiązuje połączenie z *Key Tag'iem*. W tym celu dokonuje wykrycia urządzeń posiadających odpowiedni zestaw serwisów i charakterystyk. Dla każdego z nich, sprawdza ono jego nazwę. Jeśli urządzenie nazywa się "KeyTag", wówczas dopiero nawiązywane jest z nim połączenie. W trakcie połączenia, urządzenie peryferyjne musi przesyłać kod deaktywujący (wygenerowany w trakcie parowania), zaszyfrowany kluczem głównym (wylosowanym przez płytke główną w trakcie parowania). Jeśli zostanie on poprawnie odszyfrowany, alarm jest deaktywowany. W przeciwnym razie - połączenie zostaje zerwane.

Ponadto, równolegle, niezależnie od tego czy alarm został deaktywowany czy nie, uruchamiane jest zadanie cyklicznego pobierania próbek lokalizacji. Okres wynosi 10 sekund i w momencie próbkowania pobierane są dane takie jak:

- Status lokalizacji (brak sygnału z satelitów, lokalizacja wyznaczona)
- Lokalizacja pojazdu
- Prędkość pojazdu
- Średnie przyspieszenie pojazdu z okresu pomiędzy próbками
- Azymut ruchu
- Ocena jazdy z okresu pomiędzy próbками
- Parametr HDOP informujący o dokładności wyznaczenia lokalizacji (idealnie, wartość mniejsza od 1)
- Liczba satelitów z których odebrano sygnał
- Czas pobrania próbki

Na etapie testów, okazało się, że parametr określający azymut pojazdu w momencie poboru próbki jest praktycznie bezużyteczny, ze względu na znaczące wahania.

Na rysunku 6.2 przedstawiono diagram interakcji, który pokazuje przeływ sterowania pomiędzy zadaniami.



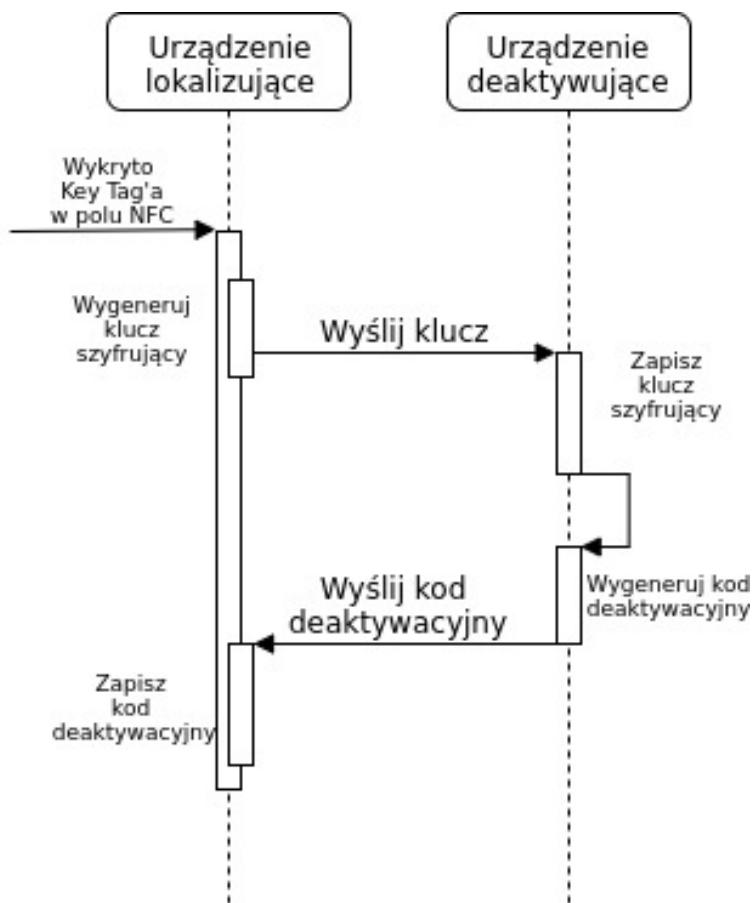
Rysunek 6.2: Przepływ sterowania w przypadku wykrycia ruchu pojazdu.

Źródło: Twórczość własna

## 6.2 Urządzenie deaktywujące

Urządzenie deaktywujące stanowi najprostszy moduł w całym systemie. Z racji tego, że posiada on tylko jedną funkcję, ale jakże istotną funkcję - deaktywację alarmu, jego główną cechą powinna być energooszczędność. Z tego powodu, urządzenie to pozbawione jest zewnętrznych układów poza anteną BLE i przez większość czasu znajduje się w trybie oszczędzania energii.

Aby system w samochodzie stał się operacyjny, pierwszym krokiem jest sparowanie ze sobą modułu lokalizującego z przeznaczonym dla niego urządzeniem deaktywacyjnym. Polega ona na wygenerowaniu 16 - bajtowego klucza szyfrującego AES128 po stronie pierwszego urządzenia, oraz 16 - bajtowej komendy deaktywującej po stronie *Key Tag'a*. Losowanie komendy deaktywującej zamiast wprowadzenie jej stałej dla wszystkich urządzeń stanowi dodatkowe zabezpieczenie. Jak zostało przedstawione we wcześniejszych rozdziałach, operacja parowania realizowana jest poprzez interfejs NFC. Przedstawiono ją na rysunku 6.3.

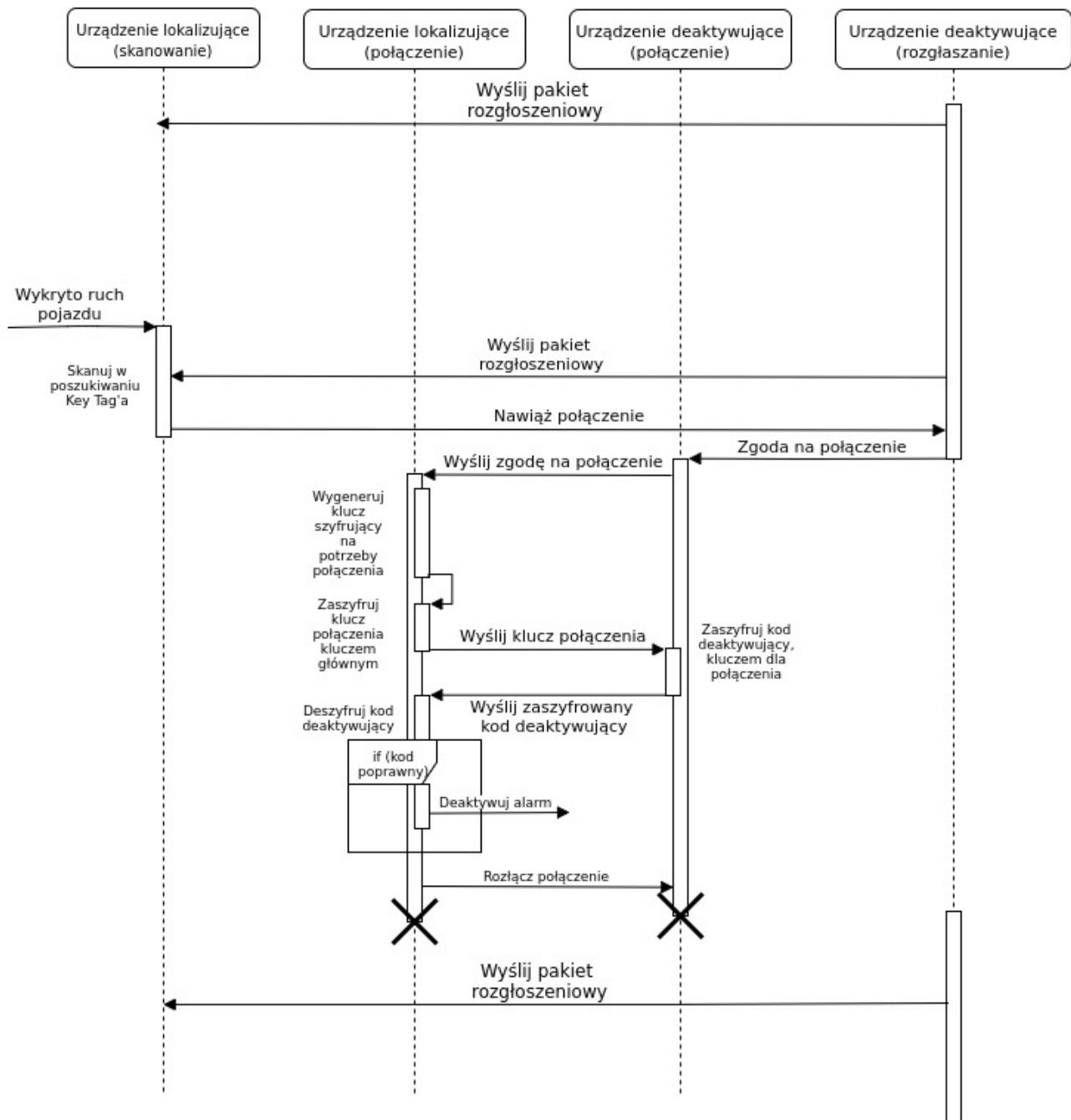


Rysunek 6.3: Przepływ sterowania w trakcie parowania urządzenia lokalizującego z urządzeniem deaktywującym.

Źródło: Twórczość własna

W momencie wykrycia ruchu pojazdu, urządzenie lokalizujące uaktywnia mechanizm skanowania urządzeń wykorzystujących BLE w poszukiwaniu *Key Tag'a*. Gdy znajdzie urządzenie o pasującej specyfikacji (nazwa oraz struktura serwisów i charakterystyk), łączy się z nim i generuje ono tymczasowy 16 - bajtowy klucz szyfrujący, na potrzeby aktualnego połączenia, który jest następnie wysyłany do urządzenia deaktywującego. Następnie, *Key Tag* dokonuje szyfrowania tym kluczem kodu deaktywacyjnego wylosowanego na etapie parowania i przesyła spowrotem do urządzenia lokalizującego. Ono sprawdza czy klucz jest poprawny i jeśli tak - deaktywuje alarm. Jeśli nie, alarm pozostaje aktualny. W każdym wypadku, po przesłaniu klucza, połączenie zostaje przerwane, a urządzenie lokalizujące wyłącza skanowanie. Zostanie ono włączone dopiero gdy zostanie wykryty nowy ruch pojazdu. Schemat operacji deaktywacji przedstawiono na rysunku 6.4.

Dzięki takiemu rozwiązaniu, następuje minimalizacja nawiązywania połączeń. Jest to korzystne z punktu widzenia obu urządzeń, ponieważ połączenie poprzez Bluetooth Low Energy stanowi najbardziej energochłonny element komunikacyjny pomiędzy urządzeniami.



Rysunek 6.4: Przepływ sterowania w momencie deaktywacji alarmu.

Źródło: Twórczość własna

## 6.3 Aplikacja serwerowa

Kolejnym etapem pracy było zaprogramowanie aplikacji serwerowej, która obsługiwałaby zapytania HTTP (*ang. Hyper Text Transfer Protocol*) użytkownika oraz zapytania do bazy danych. Aplikacja została napisana przy pomocy framework'a Qt. Jest to ostatnimi czasy jedno z najpopularniejszych zbiorów bibliotek do języka C++. Wybór środowiska nastąpił z kilku powodów. Pierwszym i jednocześnie najważniejszym z nich jest wbudowany moduł obsługi relacyjnych baz danych. Dzięki temu, wykorzystując kilka wysokopoziomowych funkcji można szybko operować na danych. Ponadto, aplikacja napisana w C++ statystycznie zapewnia większą wydajność niż napisana w języku Java. Ostatnim z powodów jest wieloletnia znajomość tej biblioteki przez autora pracy.

Aplikacja składa się z dwóch głównych modułów: manager'a bazy danych oraz serwera HTTP.

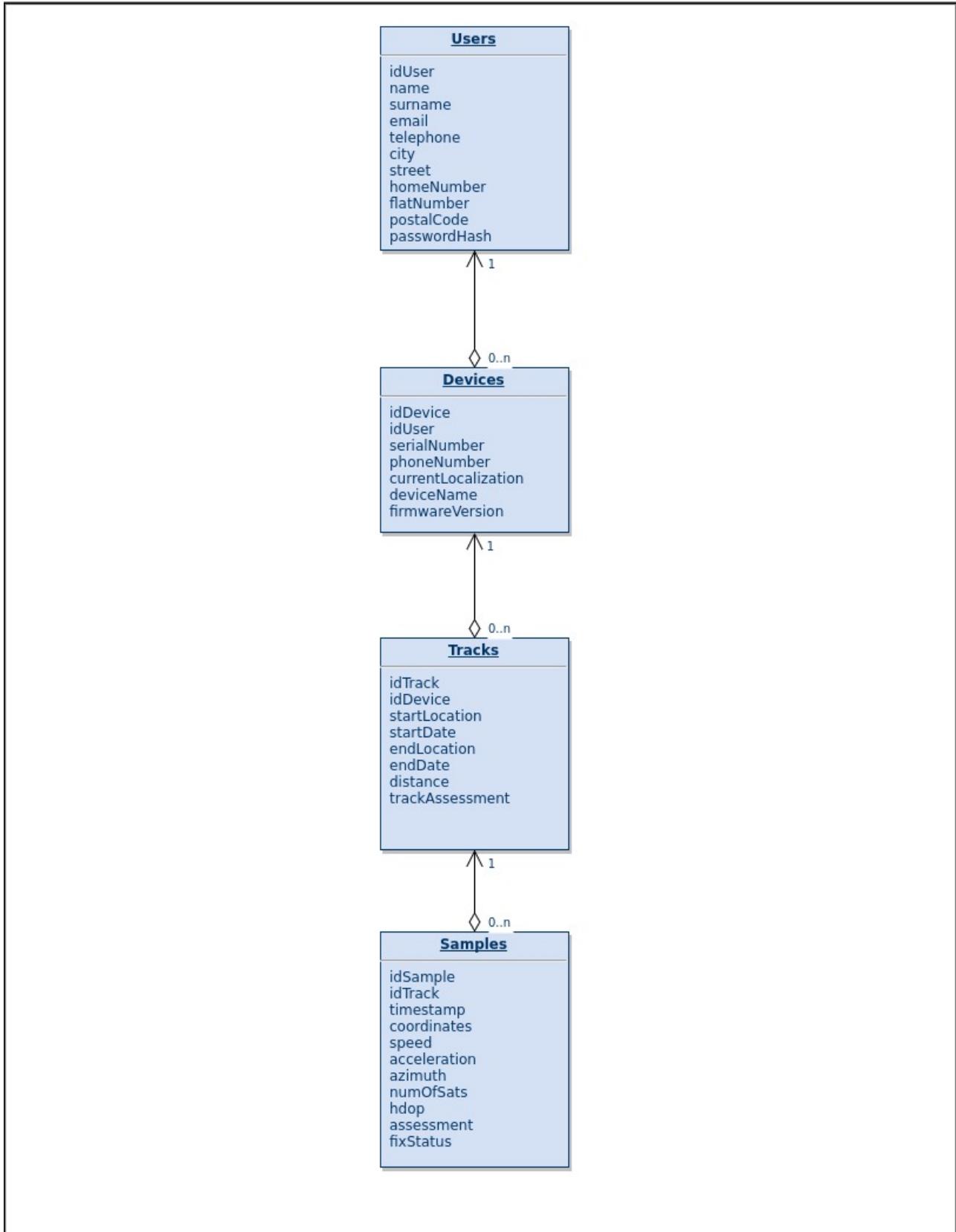
Baza danych składa się z 4 tabel. Są to :

- Tabela użytkowników
- Tabela urządzeń
- Tabela tras
- Tabela próbek

Jako silnik bazodanowy, zdecydowano się wykorzystać system SQLite. Stanowi on uproszczony, lecz bardzo wydajny sterownik opierający się na zapytaniach SQL, służący do obsługi relacyjnych baz danych. Kolejną zaletą jest fakt, iż jest on obsługiwany wewnętrznie przez bibliotekę Qt. Relacje pomiędzy nimi zostały przedstawione na rysunku 6.5.

Założenia struktury bazy danych są następujące:

- Każdy zarejestrowany użytkownik może posiadać więcej niż jedno urządzenie
- Każde urządzenie może otrzymać opisującą je nazwę
- Do każdego z urządzeń może być przypisana więcej niż jedna trasa
- Trasa posiada wpisy o miejscach i czasach ich rozpoczęcia i zakończenia



Rysunek 6.5: Schemat relacji między tabelami w bazie danych.

Źródło: Twórczość własna

Jako serwer HTTP zastosowano bibliotekę QttpServer użytkownika supamii[21]. Została ona napisana pod licencją MIT, co zapewnia swobodę użytkowania i modyfikacji kodu źródłowego, a nawet komercyjne zastosowanie pod warunkiem umieszczenia oryginalnych warunków licencyjnych i informacji o autorze. Biblioteka umożliwia komunikację poprzez zarówno zapytania HTTP typu GET jak i POST.

Przy pomocy aplikacji, można wykonać następujące operacje:

- Logowanie użytkownika
- Wylogowywanie użytkownika
- Rejestracja nowego użytkownika
- Pobieranie danych o użytkowniku z bazy danych
- Zmianę danych użytkownika
- Zmianę hasła użytkownika
- Kasowanie użytkownika
- Dodawanie urządzenia do konta użytkownika
- Pobieranie listy urządzeń przypisanych do użytkownika
- Pobieranie informacji o urządzeniu
- Usuwanie urządzenia z bazy danych
- Dodawanie nowej trasy do urządzenia
- Pobieranie listy tras przypisanej do urządzenia
- Pobieranie informacji o trasie
- Dodawanie próbek do trasy
- Zakończanie trasy
- Usuwanie trasy

## 6.4 Strona internetowa

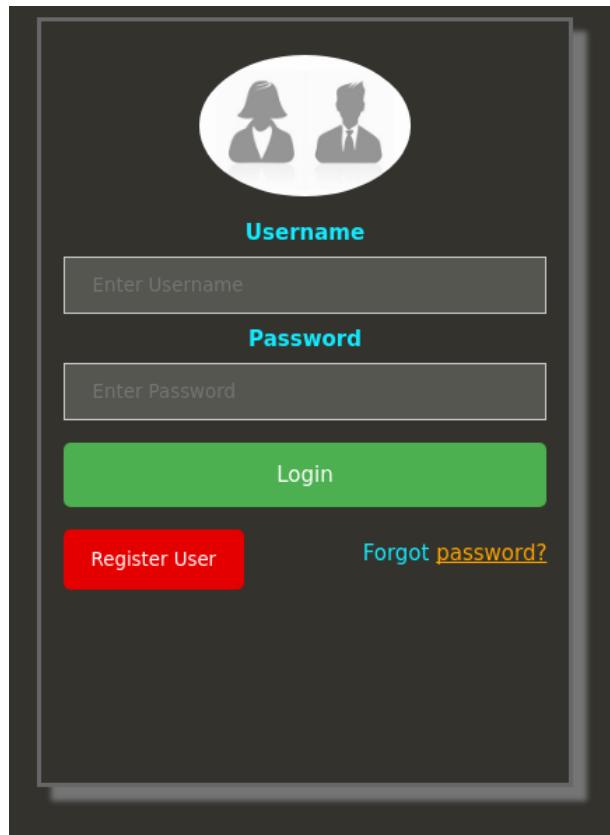
Ostatnim elementem systemu opisywanego w tej pracy jest strona internetowa. Dzięki niej, możliwy staje się zdalny i czytelny podgląd danych, które napływają praktycznie w czasie rzeczywistym. Z tego względu stanowi ona równie istotną część pracy co pozostałe. Dzięki wykorzystaniu strony internetowej, możliwe staje się stworzenie konta użytkownika, przypisanie do niego urządzeń, a następnie podgląd tras - zarówno aktualnie przebywanej jak i historycznych wraz z ich parametrami, zapisanymi na serwerze.

Szkielet strony internetowej został zaprojektowany w języku HTML (*ang. Hypertext Markup Language*), natomiast jej część funkcjonalna przy użyciu języka *JavaScript*. Jest to język skryptowy, wykorzystywany głównie w przeglądarkach i charakteryzuje się tym, że jest jednowątkowy.

Dzięki wykorzystaniu *JavaScript*, możliwe staje się dodanie wielu funkcjonalności do struktury strony internetowej, które stanowią jej połoczenie ze światem zewnętrznym, oraz umożliwiają wykorzystanie wielu efektownych rozwiązań wizualnych, jak na przykład animowane przewijanie strony, wyskakujące okienka czy interaktywne wykresy.

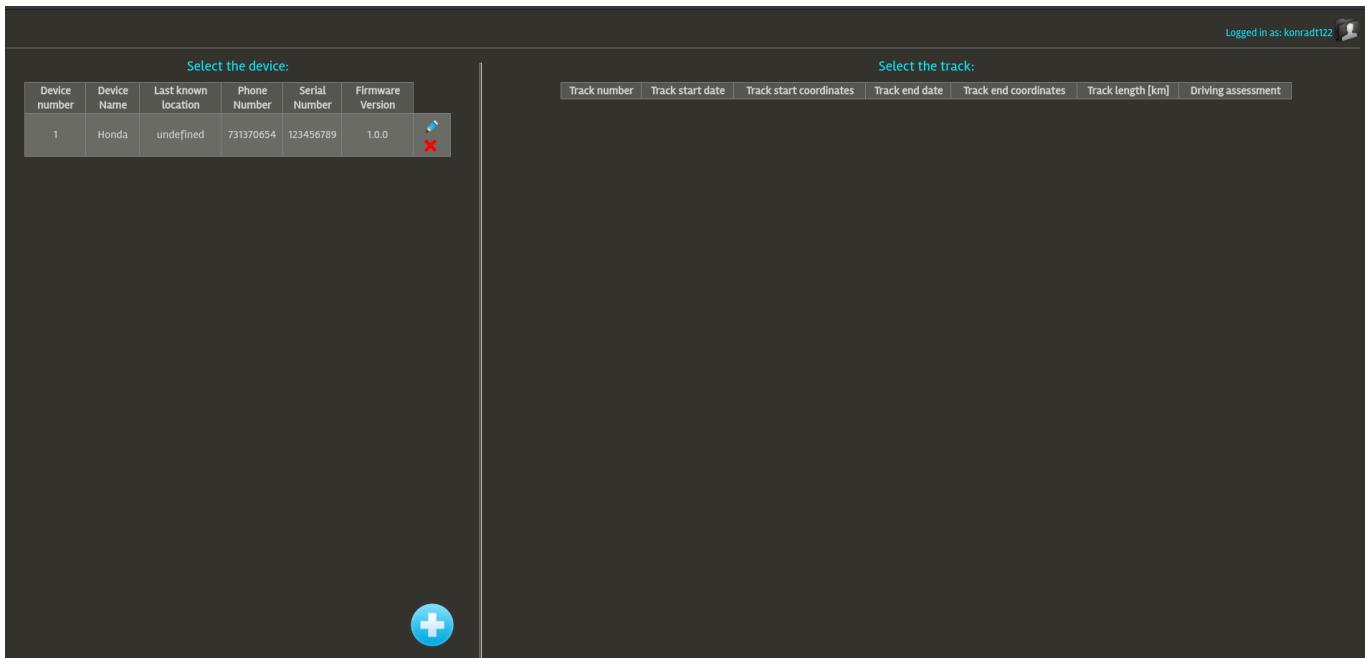
W niniejszej pracy, dzięki wykorzystaniu *JavaScript* możliwe stało się połączenie strony internetowej z bazą danych w celu użycia wymienionych w poprzednim podrozdziale funkcji.

Na rysunkach 6.6, 6.7, 6.8 oraz 6.9 przedstawiono podstawowe ekranы: logowania, rejestracji użytkownika, ekran główny oraz ekran trasy.

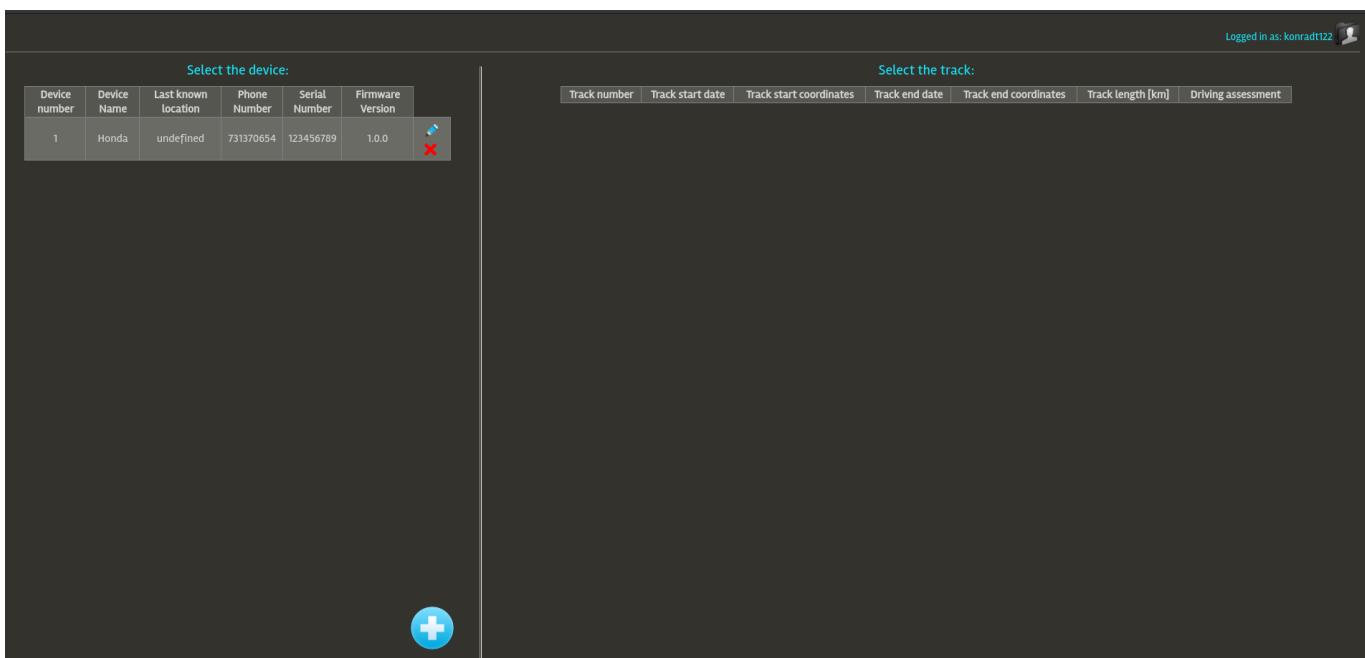


Rysunek 6.6: Strona logowania. Źródło: Twórczość własna

Rysunek 6.7: Ekran rejestracji użytkownika. Źródło: Twórczość własna



Rysunek 6.8: Strona główna. Źródło: Twórczość własna



Rysunek 6.9: Okno trasy. Źródło: Twórczość własna

Ekran trasy przedstawia poszczególne próbki lokalizacji, należące do trasy, na mapie od firmy Google. W tym celu, niezbędne staje się wykorzystanie API producenta dla modułu Google Maps. Pozwala ono na wyświetlenie mapy, przybliżanie i oddalenie, odnalezienie lokalizacji, nanieśenie znaczników (w tym autorskich - zdefiniowanych w formacie grafiki wektorowej - .svg).

Ponadto, dzięki zastosowaniu modułu *Info Bubble*, możliwe staje się wyświetlenie na mapie interaktywnego okienka informacyjnego, powiązanego z konkretnym znacznikiem geolokalizacyjnym.

Aby wykorzystać API Google Maps, należy zarejestrować aplikację na stronie <https://developers.google.com/maps/documentation/javascript/get-api-key>. Po rejestracji, do projektu przypisany zostanie klucz, który należy zawsze wewnątrz zapytania HTTP wykonywanego przy ładowaniu strony, umożliwiającego ściągnięcie z internetu plików źródłowych zawierających kod obsługujący mapy. Przedstawiono to na listingu 6.4.

*Listing 6.4: Fragment kodu pozwalający na użycie API Google Maps*

```
<script async defer src="https://maps.googleapis.com/maps/api/js?key=TWoj_KLUCZ_API&callback=initMap" type="text/javascript"></script>
```

Na listingach 6.5 oraz 6.6 przedstawiono fragmenty kodu, które umożliwiają wysłanie zapytań HTTP typu POST oraz GET. Stanowią one podstawową formę komunikacji między stroną internetową, a serwerem bazodanowym.

*Listing 6.5: Fragment kodu pozwalający na użycie zapytań HTTP typu POST*

```
function http_post(_url , _arguments , _successCallback)
{
    var _args = JSON.stringify(_arguments);
    // Send the data using post
    $.ajax({
        url: _url ,
        type: "POST",
        dataType: "json", // expected format for response
        contentType: "application/json", // send as JSON
        data: _args ,

        complete: function() {
            //called when complete
        },
        success: function(msg) {
            _successCallback(msg);
        },
        error: errorCallback ,
    });
}
```

*Listing 6.6: Fragment kodu pozwalający na użycie zapytań HTTP typu GET*

```
function http_get(_url, _arguments, _successCallback)
{
    //var _args = JSON.stringify(_arguments);
    var _args = _arguments;
    // Send the data using post
    $.ajax({
        url: _url,
        type: "GET",
        dataType: "json", // expected format for response
        contentType: "application/json", // send as JSON
        data: _args ,
        complete: function() {
            //called when complete
        },
        success: function(msg) {
            _successCallback(msg);
        },
        error: errorCallback,
    });
}
```

## Rozdział 7

### Analiza stylu jazdy

## Rozdział 8

### Podsumowanie

# Bibliografia

- [1] Spark Nano. *Spark Nano 5.0 GPS Tracker*. [https://ii.brickhousesecurity.com/fcgi-bin/iipsrv.fcgi?FIF=/images/brickhousesecurity//source/GPS-SN5\\_6.tif&wid=335&cvt=jpeg](https://ii.brickhousesecurity.com/fcgi-bin/iipsrv.fcgi?FIF=/images/brickhousesecurity//source/GPS-SN5_6.tif&wid=335&cvt=jpeg).
- [2] My Car Tracks. *Aplickacja na smartphone MyCarTracks*. <https://www.mycartracks.com/features;jsessionid=3D9E99856BB52CB325C6D699AA3EF5F0.mct-node-one>.
- [3] STI. *STI GL300 GPS Tracker*. [https://images-na.ssl-images-amazon.com/images/I/81Xv5REeNxL.\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/81Xv5REeNxL._SL1500_.jpg).
- [4] Agilent Technologies. *GSM presentation*.  
[http://mars.merhot.dk/w/images/8/88/GSM\\_praesentation\\_noter.pdf](http://mars.merhot.dk/w/images/8/88/GSM_praesentation_noter.pdf), 2000.
- [5] Tutorials Point Pvt. Ltd. *GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS (GSM) TUTORIAL*.  
<https://www.tutorialspoint.com/gsm/gsmTutorial.pdf>, 2014.
- [6] *Cellular network architecture image*.  
[https://en.wikipedia.org/wiki/Cellular\\_network](https://en.wikipedia.org/wiki/Cellular_network).
- [7] Zogg J. M. U-Blox AG. *GPS Essentials of Sattelite Navigation Compendium*, 2009.
- [8] Konrad Traczyk. *Projekt personalnego urządzenia śledzącego pozycję geograficzną, sprzązonego z aplikacją mobilną*, 2016.
- [9] Obrazek prezentujący strukturę bitów i poziomy napięć w interfejsach rs232 i rs422.  
<http://www.bb-elec.com/Images/whitepaper-images/DataByte.aspx>.
- [10] Tiger Chen. *MC60 Series Hardware Design*. Quectel, Shanghai, China, May 2017.
- [11] Townsend K., Cuff C., Davidson A., and R. *Getting started with Bluetooth Low Energy*. O'Reilly Media, 2014.

- [12] Igoe T., Coleman D., and Jepson B. *Beginning NFC. Near Field Communication with Arduino, Android and PhoneGap*. O'Reilly Media, 2014.
- [13] *Obrazek przedstawiający budowę tag'a RFID.*  
<http://1oomzzme3s617r8yzr8qutjk.wpengine.netdna-cdn.com/wp-content/uploads/2017/04/RFID-basics-Fig-3.jpg>.
- [14]
- [15] *nRF52832 Product Specification v1.4.*
- [16] Nordic Semiconductor. *Dokumentacja SDK mikrokontrolerów rodziny nRF5x.*  
<https://infocenter.nordicsemi.com/index.jsp>.
- [17] Monika Jaworowska. *Projektowanie płytka PCB dla linii szybkiej komunikacji.*  
<https://elektronikab2b.pl/technika/19012-projektowanie-plytek-drukowanych-z-ukladami-WgYv75-YUW1>, 2013.
- [18] Piotr Górecki. *O paskudztwach i czarodziejach, czyli zakłócenia w układach elektrycznych.*  
[https://elportal.pl/pdf/2003/edw\\_2003\\_09\\_s20.pdf](https://elportal.pl/pdf/2003/edw_2003_09_s20.pdf), 2003.
- [19] Krzysztof Kowalczyk. *Advanced Encryption Standard.*  
<http://www.crypto-it.net/pl/symetryczne/aes.html?tab=1>.
- [20] en.wikipedia.org. *Advanced Encryption Standard.*  
[https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard).
- [21] Son-Huy Pham. *QtppServer.*  
<https://github.com/supamii/QtppServer/blob/QTTPv1.0.0/LICENSE>.

# Wykaz skrótów

AES	Advanced Encryption Standard
API	Application Programming Interface
BLE	Bluetooth Low Energy
BTS	Base Tranceiver Station
ESD	Electrostatic Discharge
GATT	Generic Attibute
GCC	GNU Compiler Collection
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISM	Industrial, Scientific, Medical (pasmo częstotliwości)
SDK	Software Development Kit
SMS	Short Message System
SIM	Subscriber Identification Module
UHF	Ultra High Frequency