POLITECHNIKA WARSZAWSKA

Wydział Mechatroniki

Praca magisterska

Konrad Traczyk

Projekt urządzenia do lokalizacji pojazów w trybie on i offline

Opiekun pracy:
prof. dr hab. Michał Bartyś

Warszawa, 2017

Spis treści

Spis rysunków

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), a także całego systemu informatycznego, który pozwoliłby na obsłużenie pozyskanych danych. W jego skład wchodzą:

- Aplikacja mobilna na telefon z systemem operacyjnym Android.
- Strona WWW, umożliwiająca zdalny podgląd danych z przypisanych do użytkownika urządzeń.
- Aplikacja serwerowa, która obsługuje zapytania użytkownika oraz zapisująca napływające dane do bazy danych SQLite.

Do dodatkowych wymagań stawianych urządzeniu należą:

- Zapewnienie bezpiecznej wymiany danych.
- Posiadanie zastępczego źródła zasilania, umożliwiającego pracę przy wyłączonym silniku pojazdu, bądź w razie odłączenia akumulatora.
- Niewielkie wymiary w celu umożliwienia łatwego ukrycia urządzenia 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

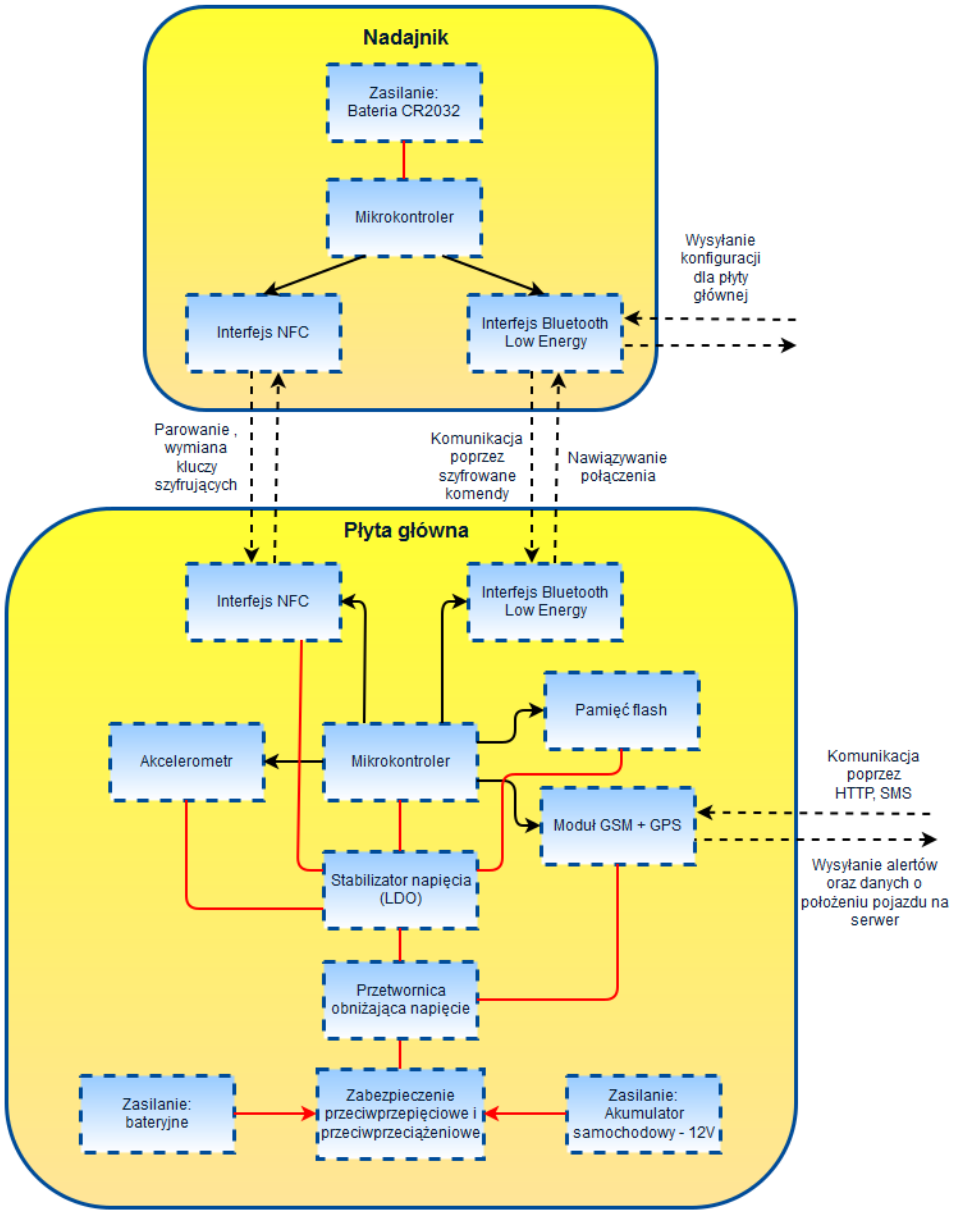
nad flotą pojazdów. Ponadto, dane te zapisywane są również w pamięci nieulotnej urządzenia, co pozwala na ograniczenie kosztów związanych z transmisją bezprzewodową i posiadaniem karty SIM od operatorów GSM.

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 główna stanowi rdzeń systemu, umożliwiający lokalizację pojazdu oraz wysyłanie danych na serwer. Drugi moduł stanowi układ deaktywujący, którego zadaniem jest dezaktywacja trybu alarmu po odpaleniu samochodu przez upoważnioną do tego osobę. Obie płytki komunikują się ze sobą poprzez protokół Bluetooh Low Energy, zapewniający energooszczędną wymianę danych, co pozwoli na zasilenie układu dezaktywującego 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 5 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

1.3 Istniejące rozwiązania

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

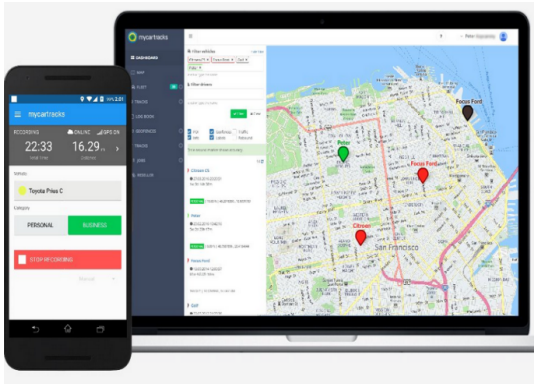
- Spark Nano 5.0 GPS Tracker

  To przenośne urządzenie do śledzenia pozycji geograficznej przy pomocy systemu GPS posiada zasilane bateryjnie. Zapewnia zdalne powiadamianie 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 smartphony 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

- 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

- 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

Rozdział 2

Wstęp teoretyczny

2.1 System GSM

2.2 System GPS

2.3 Protokół NMEA

2.4 Protokół Bluetooth Low Energy

2.5 Interfejs NFC

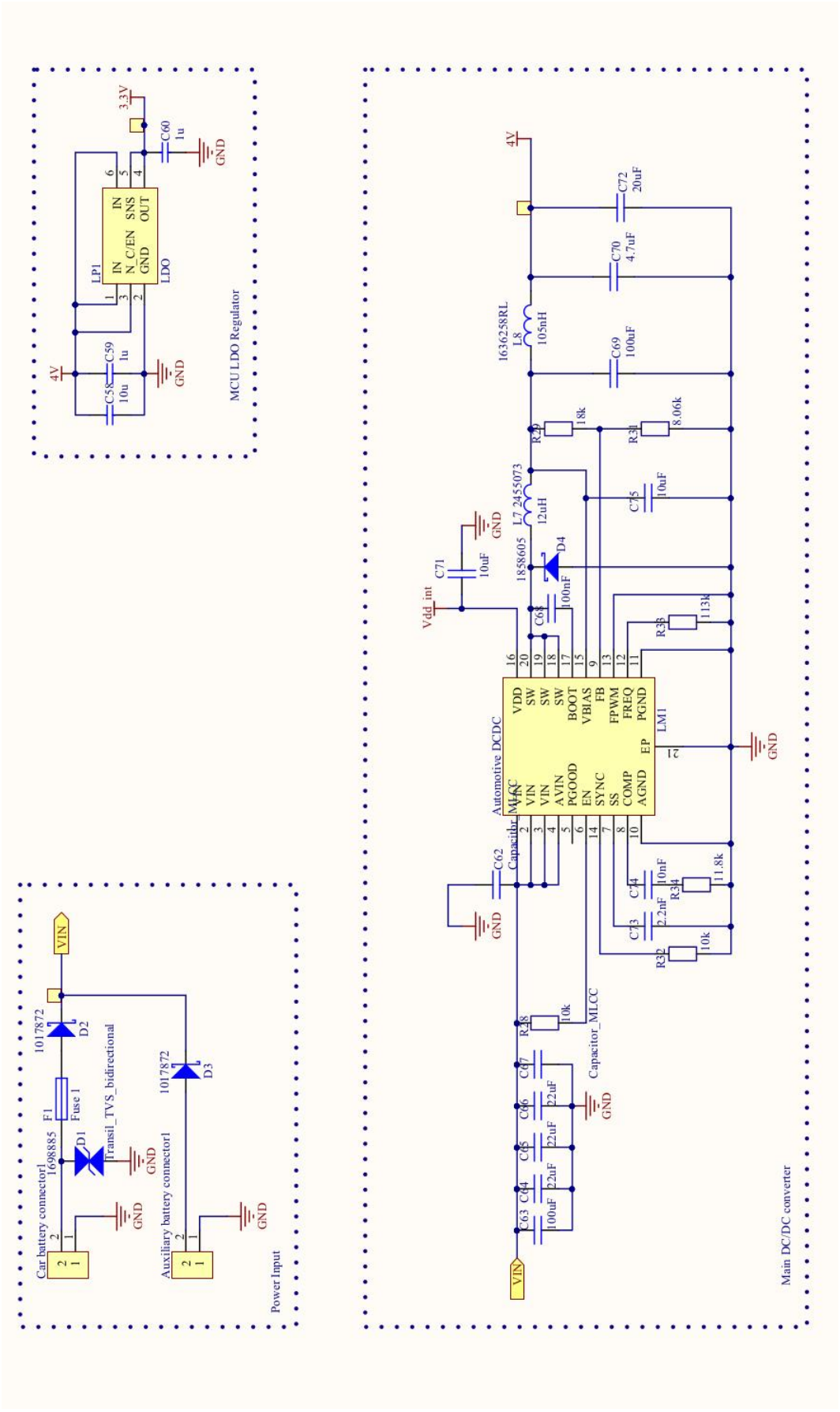
Rozdział 3

Schematy elektroniczne urządzeń

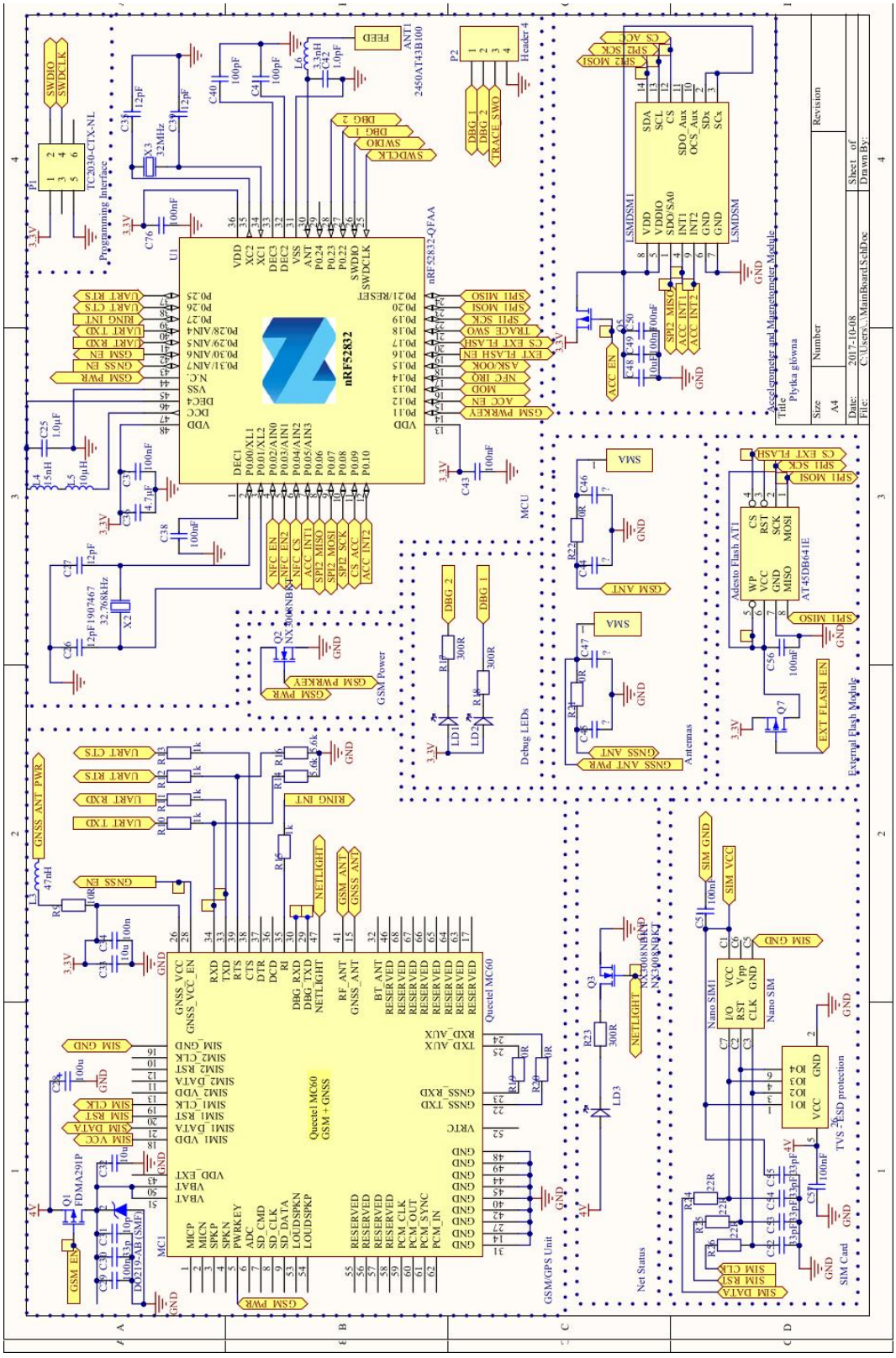
3.1 Urządzenie lokalizujące

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:

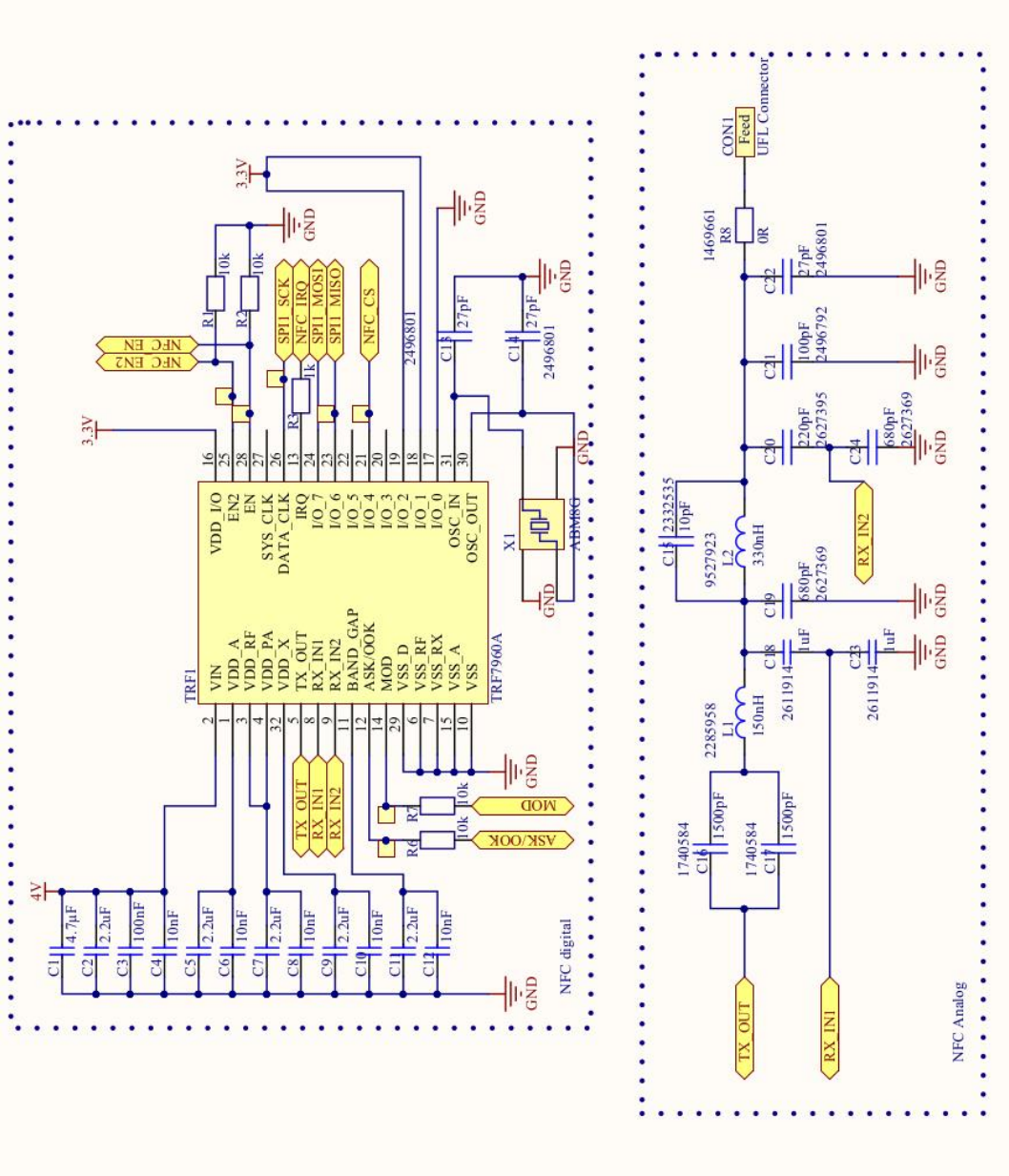
- Moduł zasilania, przedstawiony na rysunku 3.1
- Moduł funkcjonalny, przedstawiony na rysunku 3.2
- Moduł NFC, przedstawiony na rysunku 3.3



Rysunek 3.1: Schemat modułu zasilania urządzenia lokalizującego



Rysunek 3.2: Schemat modułu funkcjonalnego urządzenia lokalizującego

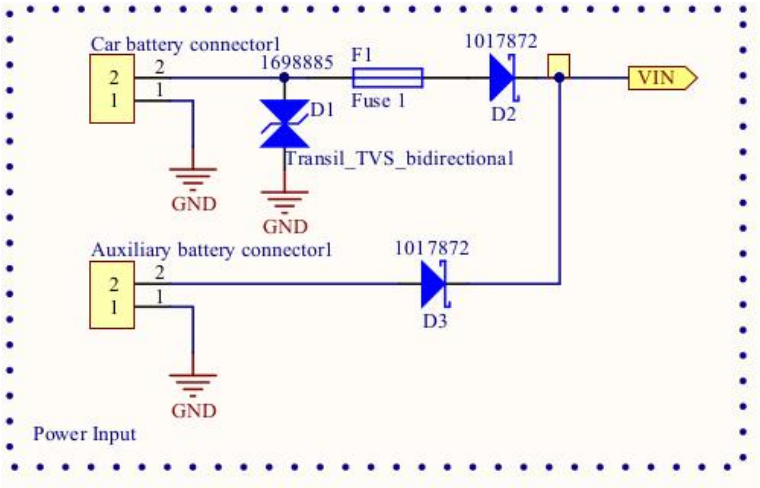


Rysunek 3.3: Schemat modułu NFC urządzenia lokalizującego

3.1.1 Schemat zasilania

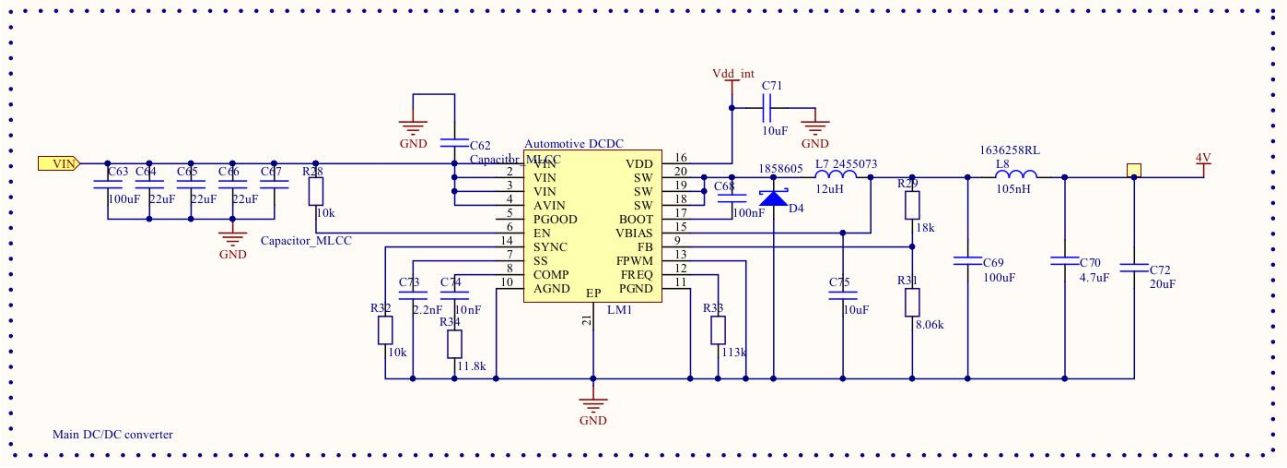
Akumulator samochodowy jest bardzo wygodnym źródłem zasilania układów elektronicznych, lecz gdy są one niewłaściwie zaprojektowane, bywa on dla nich zabójczy. Bliska odległość do alternatora i innych urządzeń indukcyjnych powoduje generowanie silnych zakłóceń na linii zasilającej. Niekiedy "szpilki" 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 wyno-

si ono 24.4V. Zabezpieczenie przeciążeniowe stanowi bezpiecznik samochodowy o wartości 4A. Ponieważ jednym z wymagań układu jest możliwość zasilania bateryjnego, konieczne jest zastosowanie dodatkowego przyłącza zasilania. Urządzenie można zasilić dowolną baterią o napięciu od 4 do 38V i wydajności prądowej co najmniej 3A w szczycie. Ze względu na prawdopodobieństwo wystąpienia różnic napięć pomiędzy dodatkowa 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

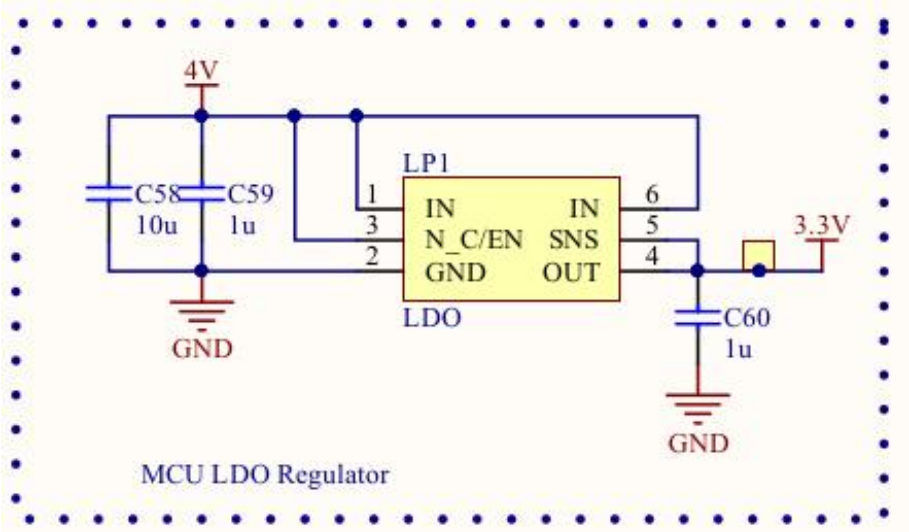
Niestety, często napięcie wejściowe, nawet po zadziałaniu zabezpieczenia w postaci transila, jest nadal zbyt duże dla zwykłych układów zasilających. Stąd konieczne jest stosowanie przetwornic impulsowych klasy automotive, które umożliwiają zasilanie napięciem wejściowym do kilkudziesięciu woltów. Schemat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



Rysunek 3.5: Schemat przetwornicy impulsowej modułu zasilania urządzenia lokalizującego

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.

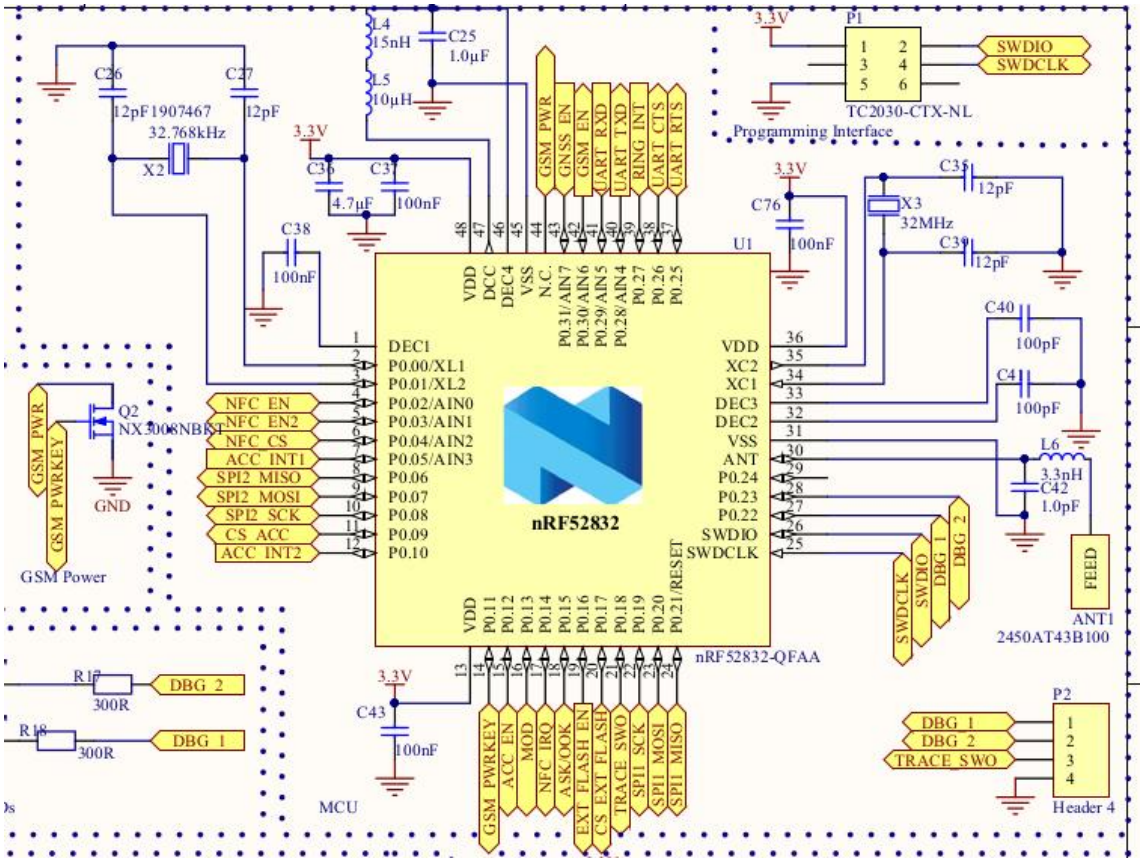
Ostatni stopień zasilania generuje z napięcia wyjściowego z przetwornicy napięcie o wartości 3.3V. Jest ono niezbędne do zasilania układów mikrokontrolera, pamięci flash, akcelerometru oraz układu GPS. Szacowany pobór prądu przez te układy wynosi ok. 200mA w szczycie, stąd dla bezpieczeństwa wykorzystano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



Rysunek 3.6: Schemat stabilizatora napięcia modułu zasilania urządzenia lokalizującego

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 oraz 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 NFCT, 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 μA /MHz gdy kod wykonywany jest z pamięci flash, natomiast w trybie oszczędzania energii pobór spada do ok 1.9 μA. Ostatnim i być może najważniejszym czynnikiem decydującym na wybranie tego układu jest posiadane 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.



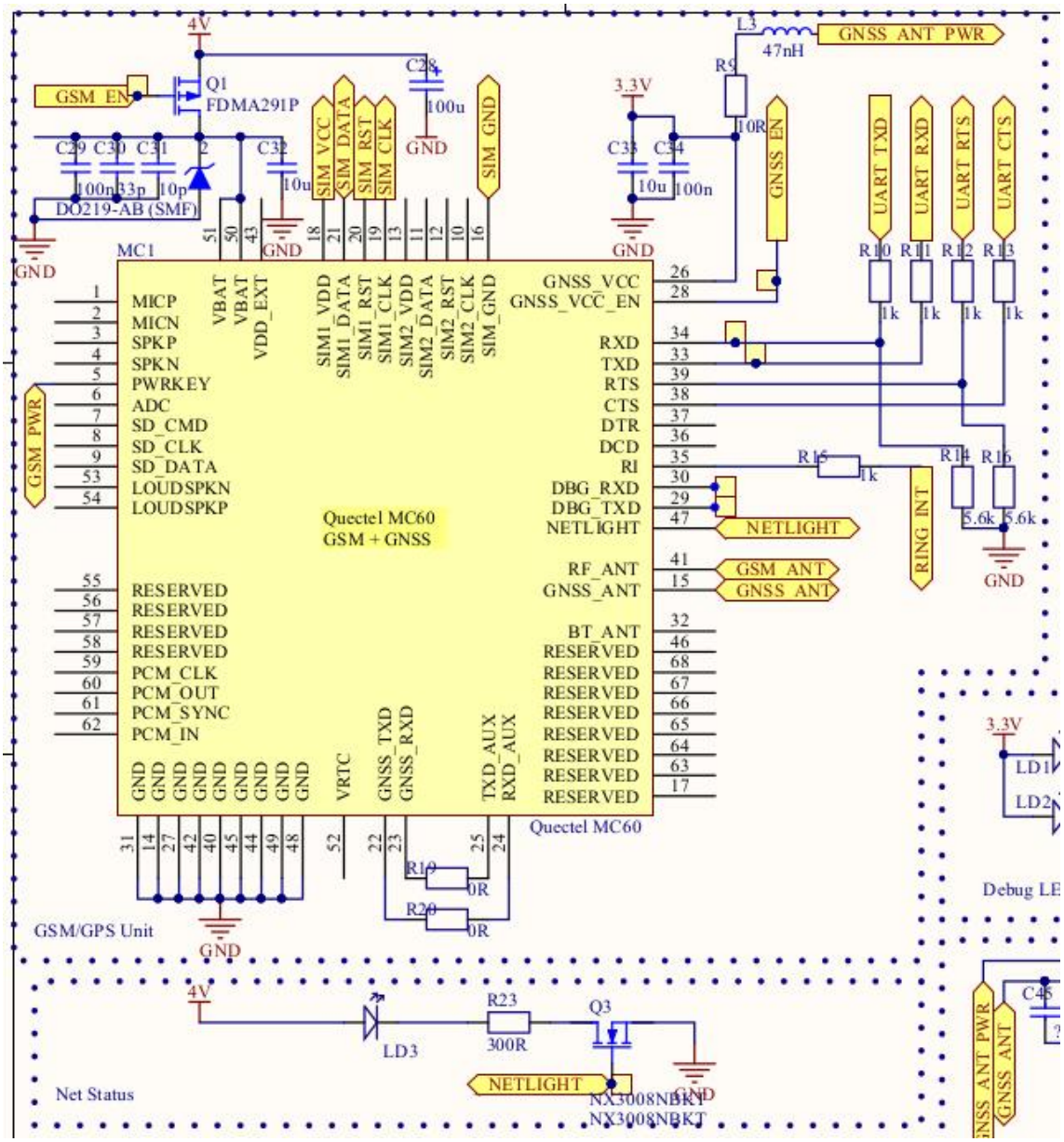
Rysunek 3.7: Schemat modułu mikrokontrolera w urządzeniu lokalizującym

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 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:

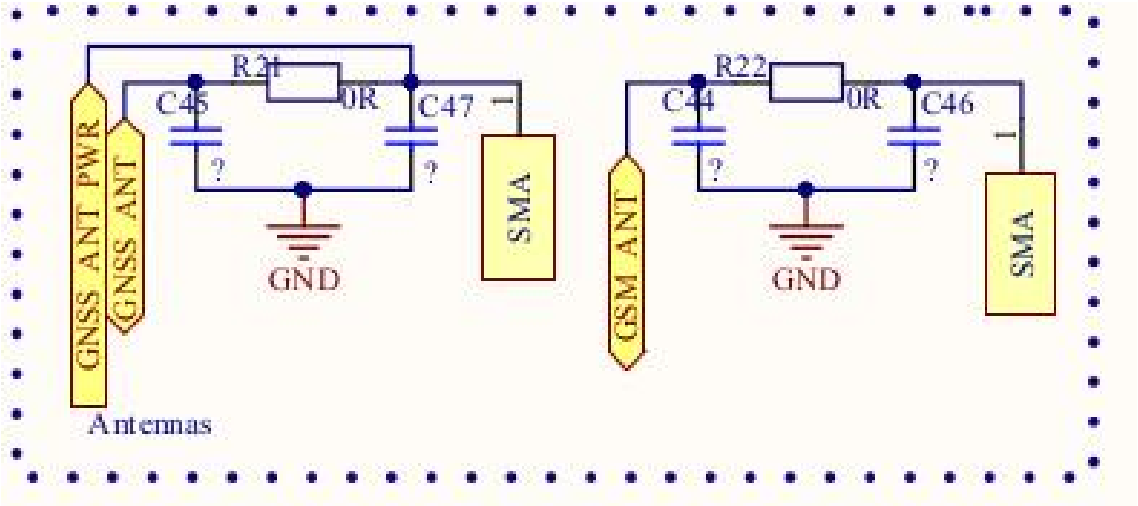
- Około 25 mA gdy działa jedynie moduł GPS
- Do 1.6 A w trakcie transmisji danych poprzez sieć GSM

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.



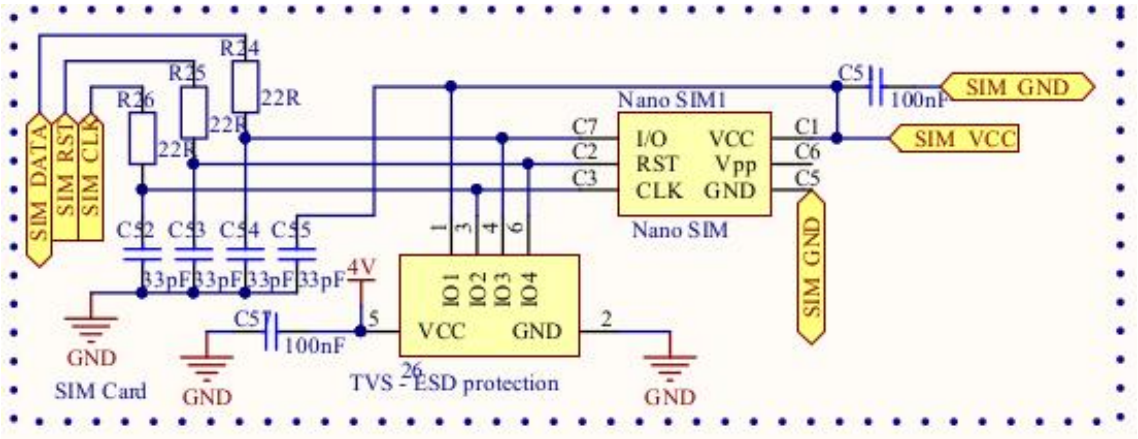
Rysunek 3.8: Schemat modułu układu GSM i GPS w urządzeniu lokalizującym

W celu zwiększenia niezawodności działania urządzenia, zdecydowano zastosować zewnętrzne anteny GSM i GPS poprawiające jakość sygnału. Dodatkowo, w celu zwiększenia jakości sygnału, antena GPS jest anteną aktywną. Oznacza to, że dostarczane jest do niej dodatkowe zasilanie, co powoduje wzmocnienie 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

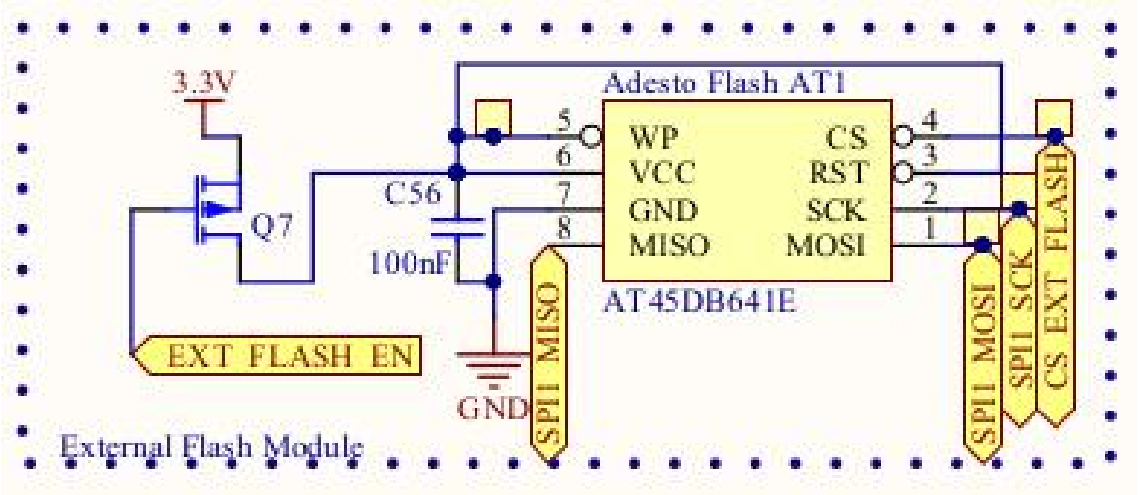
Ostatnią częścią układu GSM jest połączenie modułu z kartą SIM, umożliwiającą zalogowanie do sieci. Przedstawiono je na rysunku 3.10. Widać na nim układ TVS, który jest odpowiedzialny za zabezpieczenie wrażliwej elektroniki w karcie SIM przed wyładowaniami statycznymi ESD (*ang. Electrostatic discharge*).



Rysunek 3.10: Schemat modułu karty SIM w urządzeniu lokalizującym

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 umożliwi przechowywanie wielu długich tras oraz dokładne profilowanie statystyczne stylu jazdy kierowcy. Schemat pamięci w urządzeniu lokalizującym pokazano na rysunku 3.11.



Rysunek 3.11: Schemat modułu pamięci flash w urządzeniu lokalizującym

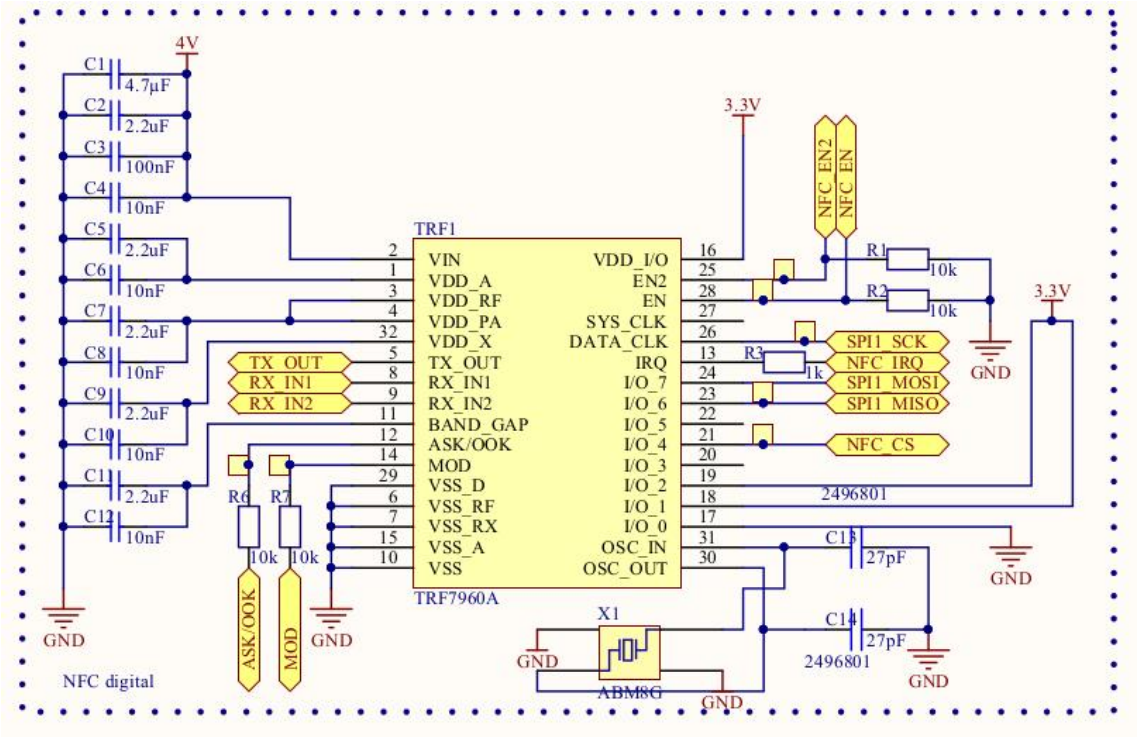
3.1.5 Moduł akcelerometra

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 akcelerometra przedstawiono na rysunku **??**.

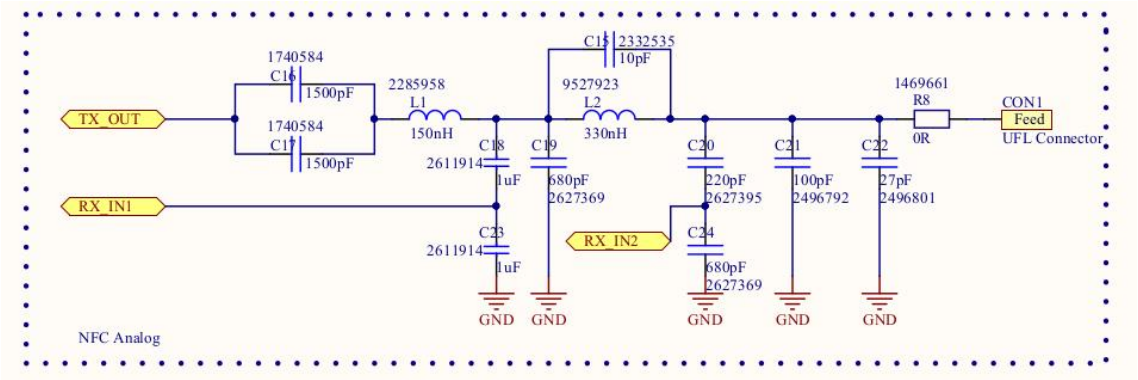
3.1.6 Moduł NFC

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 5 cm. Komunikacja odbywa się pomiędzy dwoma urządzeniami. Ze względu na sposób transmisji, jedno z urządzeń inicjuje komunikację. Inicjator generuje zmienne pole magnetyczne, w który może (lecz nie musi) zawrzeć dane wysyłane do urządzenia docelowego. Urządzenie docelowe wykrywa to pole i może odpowiedzieć poprzez odpowiednie zniekształcenie go, które jest wykrywane przez inicjator. Urządzenie docelowe nie generuje żadnego pola magnetycznego. Może jedynie zniekształcać 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 3.12 - część cyfrowa oraz 3.13 - część analogowa.



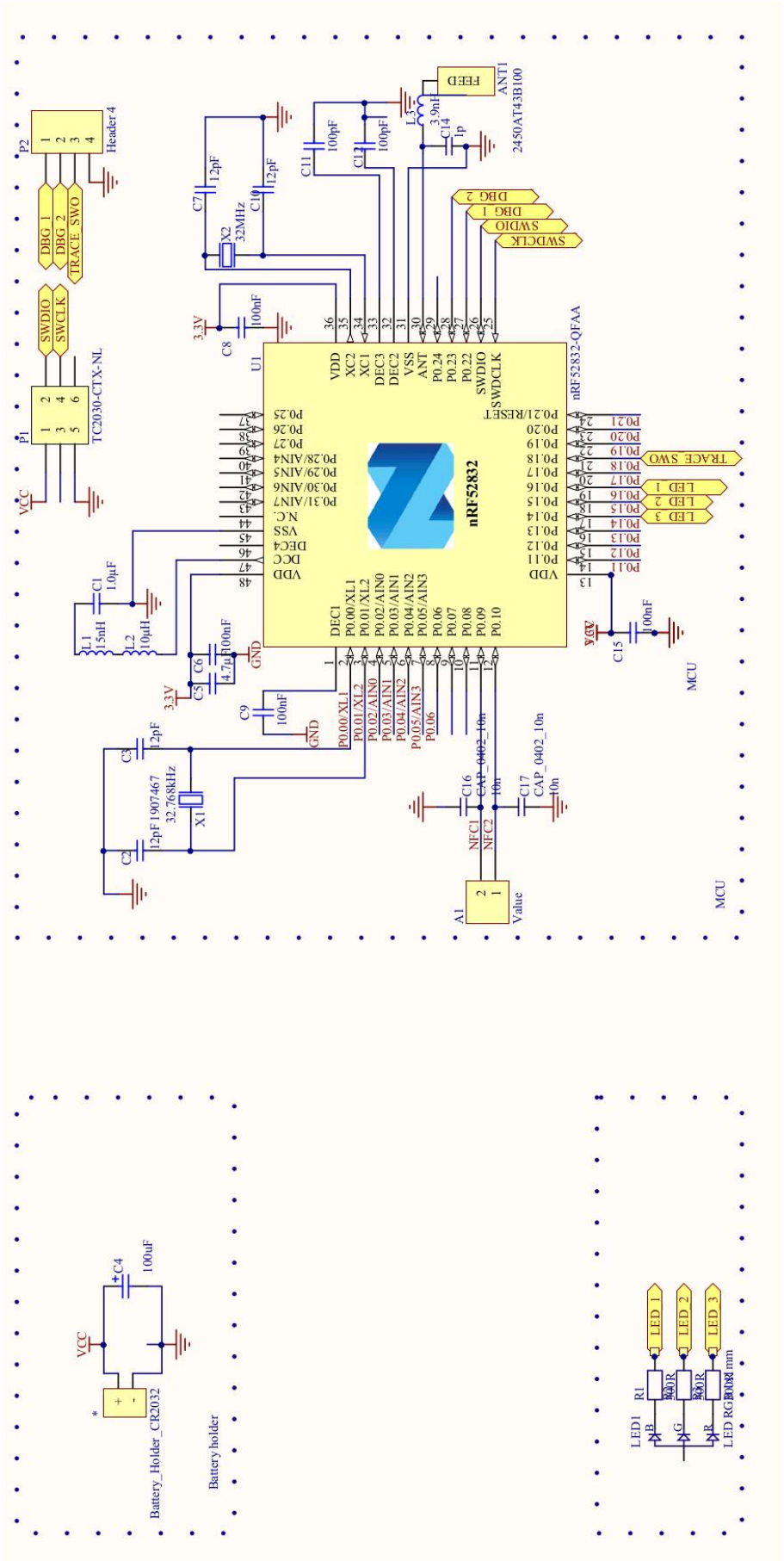
Rysunek 3.12: Schemat części cyfrowej modułu NFC w urządzeniu lokalizującym



Rysunek 3.13: Schemat części analogowej modułu NFC w urządzeniu lokalizującym

3.2 Urządzenie deaktywujące

Głównym zadaniem tego urządzenia jest cykliczne rozgłaszanie swej obecności. Po wykryciu przez urządzenie lokalizujące, łączy się ona z deaktywatorem oraz bezpiecznym kanałem dokonywane jest wyłączenie funkcji alarmu. Dzięki temu, że urządzenie to ma tak proste zadanie, nie pobiera ona dużo poboru 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 wymiany baterii. 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 3.14: Schemat modułu zasilania urządzenia deaktywującego

Rozdział 4

Schematy płytek drukowanych

4.1 Urządzenie deaktywujące

4.2 Urządzenie lokalizujące

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 zmiennego 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 dokonałem, poniważ zastosowany przeze mnie mikrokontroler nRF52832 firmy Nordic Semiconductor posiada sprzetowe wsparcie szyfrowania danych wykorzystując właśnie AES128. Algorytm ten powstał w 2001 roku w Stanach Zjednoczonych w ośrodku NIST (ang. National Institude 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 deszyfracji). Z tego powodu, klucz szyfrujący stanowi ekstremalnie wrażliwą daną, której pod żadnym pozorem nie powinno się przesyłać poprzez ogólno 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:

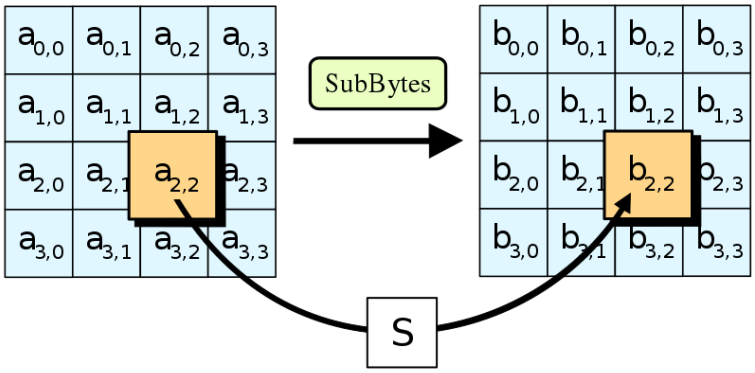
1. Utworzenie podkluczy – Etap ten polega na wygenerowaniu w sposób losowy klucza pierwotnego, a następnie na jego podstawie - po jednym podkluczu dla każdej z rund szyfrujących. Ich liczba jest uzależniona od rozmiaru klucza. Dla klucza 128-bitowego występuje 10 rund, dla klucza 192-bitowego – 12 cykli, 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.

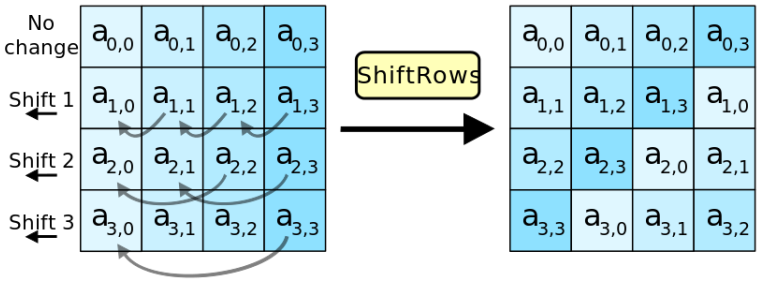
    - W pierwszym z nich, każdy bajt danych jest zastępowany innym bajtem pobranym w z góry zdefiniowanej tablicy (ang. lookup table) nazywanej S-Boxe'em Rijndael'a.

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



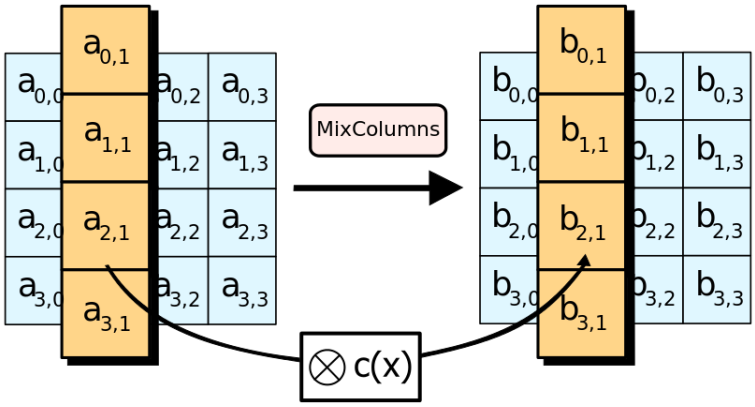
Rysunek 5.1: Wykonanie operacji Substitute Bytes

- 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.



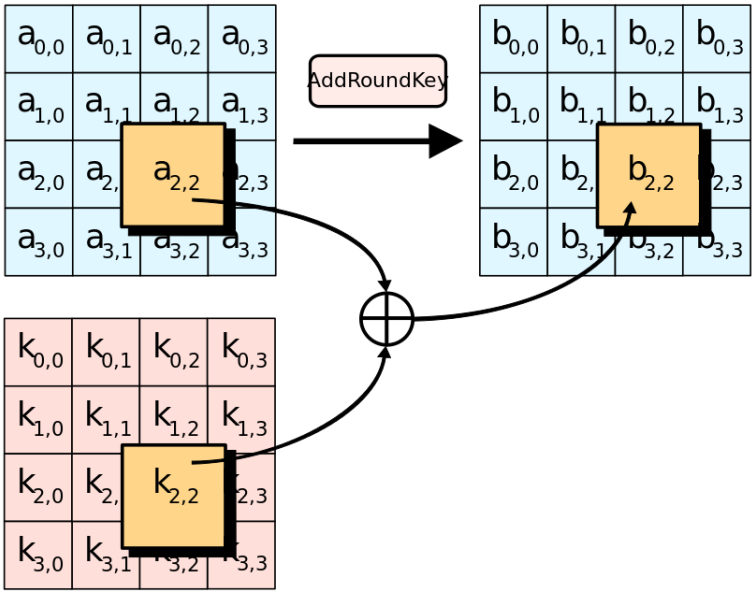
Rysunek 5.2: Wykonanie operacji Shift Rows

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



Rysunek 5.3: Wykonanie operacji Mix Columns

- 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 przypisanym do danej rundy. Wizualizację kroku przedstawiono na rysunku 5.4.



Rysunek 5.4: Wykonanie operacji Add Round Key

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

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

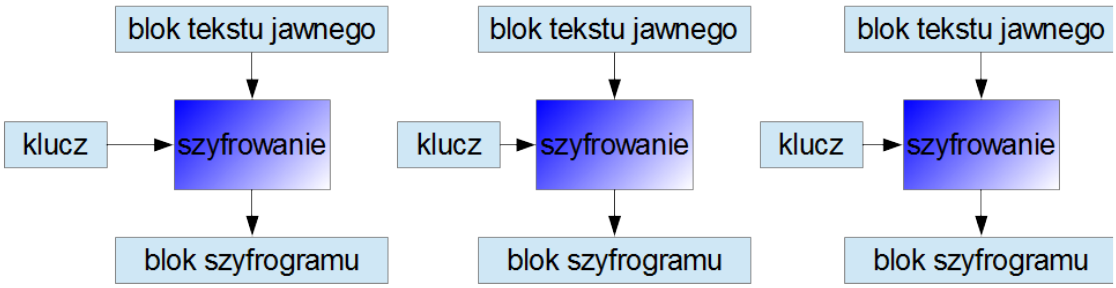
1. Odwrotne podstawianie bajtów – polega na ponownym zastosowaniu tablicy S-Box w celu podmiany bajtów.

2. Przesuwanie bajtów w wierszach w prawo. Zasada jest taka sama jak w operacji SR, zmienia się jedynie kierunek.

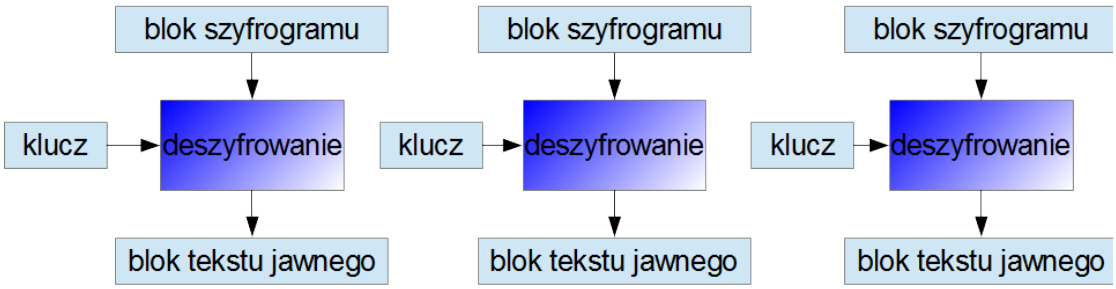
3. Wykonanie operacji XOR dla każdego bajtu bloku danych z odpowiadającym mu bajtem w podkluczu przypisanym do danej rundy deszyfrującej. Podklucze są takie same jak w trakcie szyfrowania, lecz powinny być brane w kolejności odwrotnej (zaczynając od ostatniego, a kończąc na kluczu pierwotnym).

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

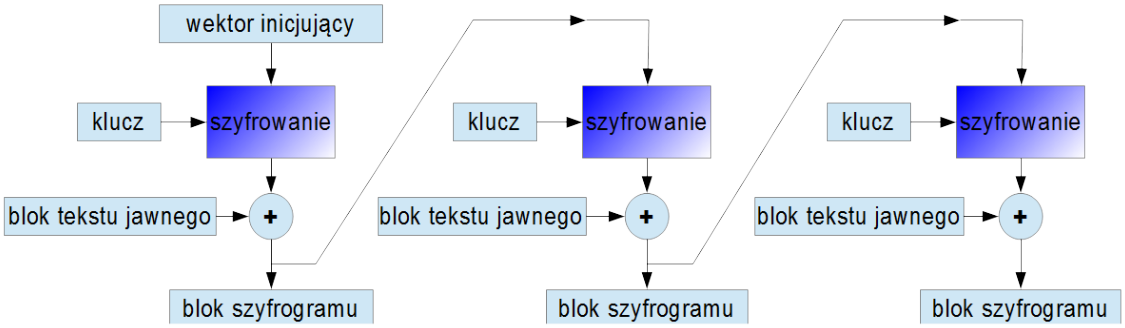


Rysunek 5.6: Operacja deszyfrowania metodą ECB

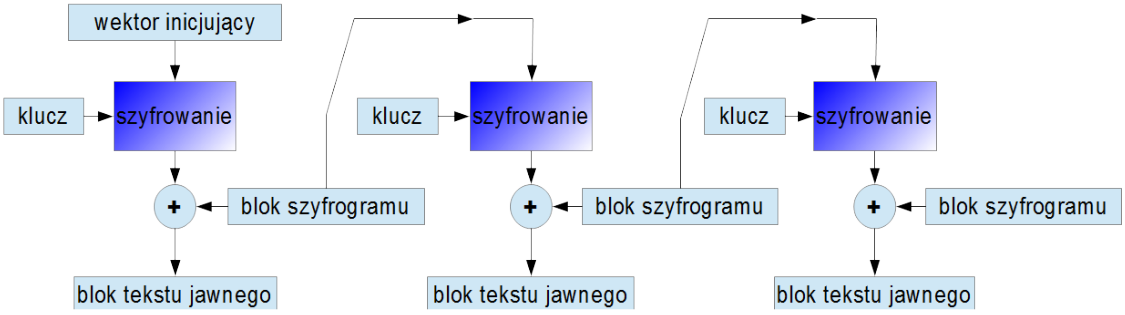
Czas trwania szyfrowania pojedynczego bloku danych o długości szesnastu bajtów na mikrokontrolerze nRF52832 wynosi około 30 μs. Deszyfrowanie trwa zaś około 60 μs.

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 wariancie 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



Rysunek 5.8: Operacja deszyfrowania metodą CFB

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 następnego 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 następnych 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 p ten na żądanie użytkownika oprzez interfejs NFC (ang. Near Field Communication) do urządzenia deaktywującego, pełniącego rolę beacona (rozgłośni). Zastosowanie NFC jest powszechnie 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 podsłuchiwania transmisji.

Bibliografia

[1] Aswin N Raghavan, Harini Ananthapadmanaban, Manimaran S Sivamurugan, and Balaraman Ravindran. Accurate mobile robot localization in indoor environments using bluetooth. *ANRH Ananthapadmanaban*, 2010.

[2] Bing-Fei Wu, Cheng-Lung Jen, and Kuei-Chung Chang. Neural fuzzy based indoor localization by kalman filtering with propagation channel modeling. *IEEE International Conference on Systems, Man and Cybernetics*, 2007.

[3] Jason M. O'Kane. *A Gentle Introduction to ROS*. University of South Carolina, 2013.

[4] Aaron Martinez and Enrique Fernández. *Learning ROS for Robotics Programming*. Packt Publishing, 2013.

[5] Dokumentacja online pakietu ROS. `wiki.ros.org`. Dostęp: 2017-07-11.

[6] Dokumentacja sdk mikrokontrolerów rodziny nrf51. `https://infocenter.nordicsemi.com/index.jsp`. Dostęp: 2017-07-11.

[7] Specyfikacja standardu Bluetooth. `https://www.bluetooth.com/specifications/bluetooth-core-specification`. Dostęp: 2017-07-11.

[8] Dokumentacja biblioteki PyBlueZ. `https://github.com/karulis/pybluez/wiki`. Dostęp: 2017-07-11.

[9] Dokumentacja biblioteki SciPy. `https://docs.scipy.org/`. Dostęp: 2017-07-11.

Wykaz skrótów

AES	Advanced Encryption Standard
API	Application Programming Interface
BLE	Bluetooth Low Energy
GATT	Generic Attibute
GCC	GNU Compiler Collection
ISM	Industrial, Scientific, Medical (pasmo częstotliwości)
MAC	Media Access Control
RAM	Random Access Memory
RSSI	Radio Signal Strength Indicator
SDK	Software Development Kit
UHF	Ultra High Frequency