



POLITECHNIKA ŚLĄSKA W GLIWICACH

WYDZIAŁ ELEKTRYCZNY

Katedra Energoelektroniki, Napędu Elektrycznego i Robotyki

## **PRACA DYPLOMOWA MAGISTERSKA**

**Układ zdalnej lokalizacji obiektów ruchomych**

**Remote location aquisition system of moving objects**

Student: **inż. Bartosz Andrzej BĄTOREK**

Nr albumu: 205986/242065

Studia: Stacjonarne II stopnia

Kierunek: Elektronika i Telekomunikacja

Specjalność: Komputerowe Systemy Sterowania

Promotor: dr inż. Marian HYLA



## Spis treści

<b>1. WSTĘP .....</b>	<b>2</b>
1.1. MOTYWACJA .....	2
1.2. CEL I ZAKRES PRACY .....	2
1.3. UKŁAD PRACY.....	3
1.4. ZAŁOŻENIA .....	3
1.5. STRUKTURA SYSTEMU LOKALIZACJI.....	3
<b>2. WYKORZYSTANE TECHNOLOGIE .....</b>	<b>5</b>
2.1. PODSTAWY TEORETYCZNE .....	5
2.1.1. <i>Pozycjonowanie GPS [11]</i> .....	5
2.1.2. <i>Użytkowanie GPS</i> .....	7
2.1.3. <i>Transmisja GSM/GPRS</i> .....	9
2.1.4. <i>GPRS jako dodatek do GSM</i> .....	12
2.1.5. <i>Klasy i tryby pracy urządzeń GSM/GPRS</i> .....	14
2.2. TECHNOLOGIE SPRZĘTOWE.....	15
2.2.1. <i>Wybór odbiornika GPS [12]</i> .....	16
2.2.2. <i>Odbiornik Racewood RGPSM002</i> .....	22
2.2.3. <i>Transmisja danych do systemu nadzoru</i> .....	23
2.2.4. <i>Wybór modułu GPRS [12]</i> .....	24
2.2.5. <i>Moduł Siemens TC65</i> .....	30
2.2.6. <i>Budowa modułu TC65</i> .....	33
2.3. OPROGRAMOWANIE.....	35
2.3.1. <i>Język programowania Java</i> .....	35
2.3.2. <i>Java Micro Edition (JavaME) [2]</i> .....	35
2.3.3. <i>Profil IMP-NG</i> .....	37
2.3.4. <i>Java Enterprise Edition (JavaEE) [5]</i> .....	39
<b>3. REALIZACJA UKŁADU ZDALNEJ LOKALIZACJI OBIEKTÓW RUCHOMYCH .....</b>	<b>41</b>
3.1 UKŁAD ELEKTRONICZNY Z MODUŁEM. ....	41
3.2. OPROGRAMOWANIE UKŁADU .....	46
2.3.5. <i>Najważniejsze klasy programu</i> .....	46
2.4. ZASADA DZIAŁANIA SYSTEMU NADZORU .....	52
2.5. WIZUALIZACJA DANYCH.....	53
2.6. PREZENTACJA PRZYKŁADOWEGO URUCHOMIENIA OPRACOWANEGO UKŁADU .....	56
<b>4. PODSUMOWANIE I WNIOSKI.....</b>	<b>58</b>
<b>BIBLIOGRAFIA .....</b>	<b>60</b>

# Rozdział 1

## 1. Wstęp

W rozdziale przedstawiono cel i zakres pracy oraz jej strukturę. Omówiona została motywacja i założenia dotyczące realizowanego układu.

W ciągu ostatnich 2 dekad można było obserwować znaczny postęp w rozwoju technologii cyfrowych i telekomunikacyjnych. Objawiało się to między innymi bardzo szybkim powiększaniem oferty urządzeń do transmisji danych, zarówno w sektorze przemysłowym, jak np. moduły GSM/GPRS, jak i w sektorze konsumenckim, jak np. telefony komórkowe, smartfony czy tablety. Mnogość dostępnych aplikacji i urządzeń tego typu pociąga za sobą chęć stosowania ich w najróżniejszych miejscach i dziedzinach życia. Trudno wyobrazić sobie w dzisiejszych czasach podróż bez nawigacji samochodowej, czy oczekiwanie na paczkę kurierską bez dostępu do systemu śledzenia przesyłek. Wszystkie z przedstawionych rozwiązań wymagają do swojego działania transmisji danych oraz nierzadko – informacji o swojej aktualnej pozycji. Niniejsza praca łączy w sobie te dwie technologie – pozycjonowanie oraz transmisję danych.

### *1.1.Motywacja*

Główną motywacją była chęć wykorzystania dedykowanego mikroprocesorowego modułu przemysłowego do transmisji danych w systemie GPRS, dodatkowo wyposażonego w maszynę wirtualną Javy. Próba utworzenia i oprogramowania systemu wbudowanego, wykorzystującego pozycjonowanie GPS w sposób inny niż tylko nawigowanie i prowadzenie do celu, jest dodatkowym czynnikiem motywującym, zwłaszcza w dobie szerokiego zainteresowania systemem GPS. Praca jest częścią programu rozwoju Laboratorium Podstaw Telekomunikacji.

### *1.2.Cel i zakres pracy*

Celem pracy było zaprojektowanie, wykonanie i uruchomienie prototypowego układu umożliwiającego określenie położenia obiektu wraz z możliwością transmisji do zdalnego systemu nadzoru.

Zakres pracy obejmuje:

- przedstawienie działania technologii GPS oraz GPRS,
- opis technologii sprzętowych i programowych użytych w pracy,
- zestawienie dostępnych na rynku odbiorników GPS,
- zestawienie dostępnych na rynku modułów do transmisji GPRS,
- przedstawienie budowy i możliwości modułu Siemens TC65,
- omówienie języka programowania Java, w szczególności Java Micro Edition,
- fizyczną realizację układu zdalnej lokalizacji,
- opracowanie oprogramowania układu w języku Java, umożliwiającego przesyłanie lokalizacji do zdalnego systemu nadzoru,
- opracowanie oprogramowania występującego w roli systemu nadzoru urządzeń do zdalnej lokalizacji,

- przedstawienie możliwych zastosowań układu,
- prezentację działania układu w jednym z wybranych zastosowań.

### ***1.3. Układ pracy***

W pierwszym rozdziale pracy zawarto opis jej struktury, cel i zakres, a także motywację oraz założenia dotyczące wykonywanego układu. W rozdziale drugim przedstawiono użyte w pracy technologie sprzętowe, takie jak pozycjonowanie GPS czy transmisja GPRS, oraz programowe, a wśród nich między innymi język programowania Java, w kontekście platformy Java Micro Edition, oraz system bazodanowy PostgreSQL. Rozdział trzeci zawiera opis realizacji układu zdalnej lokalizacji obiektów ruchomych, zarówno od strony fizycznej, jak i programowej. Rozdział ten poświęcony jest także realizacji części stacjonarnej, próbie utworzenia systemu nadzoru i kontroli, oraz magazynu danych lokalizacyjnych, pochodzących z części mobilnej. W rozdziale czwartym przedstawiono przykładowe zastosowania omawianego układu zdalnej lokalizacji, oraz zaprezentowano działanie utworzonego układu w terenie. Ostatni rozdział zawiera podsumowanie i ostateczne wnioski.

### ***1.4. Założenia***

Przy realizacji pracy przyjęto następujące założenia:

- do określenia pozycji wykorzystany zostanie system lokalizacji GPS,
- transmisja danych do systemu nadzoru odbywać się będzie z wykorzystaniem technologii GPRS,
- w celu minimalizacji kosztów wykorzystana zostanie istniejąca, prototypowa płytką z modułem TC65, będącą przedmiotem innej pracy dyplomowej [13],
- komunikacja modułu TC65 z odbiornikiem GPS odbywać się będzie z wykorzystaniem transmisji szeregowej,
- oprogramowanie modułu napisane zostanie w języku Java dla platformy Java Micro Edition, w środowisku NetBeans,
- system nadzoru i kontroli zrealizowany zostanie jako dodatek, umożliwiając prezentowanemu układowi na działanie autonomiczne.

### ***1.5. Struktura systemu lokalizacji***

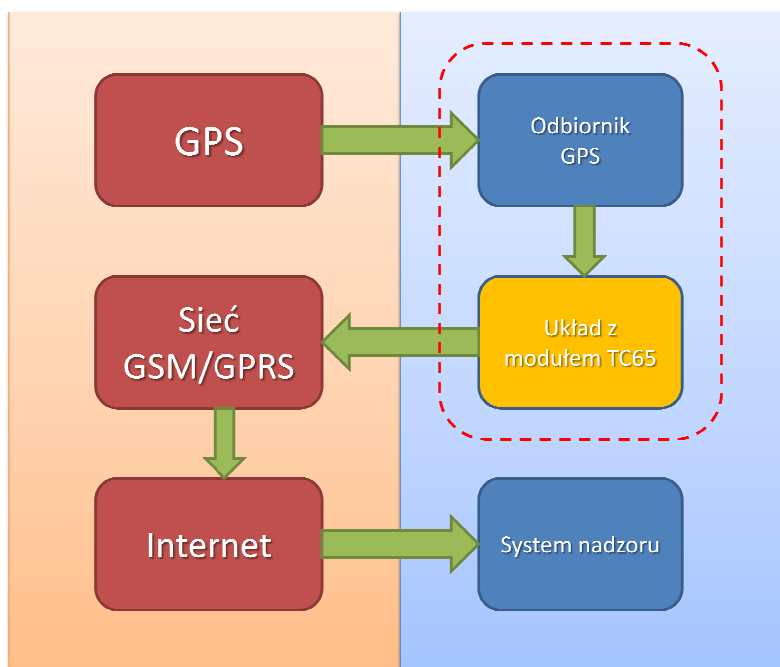
System został podzielony na dwie zasadnicze części:

- część ruchomą,
- część stacjonarną.

Podział ten wynika ze specyfiki systemu. Do części ruchomej należy prototypowa płytką z modułem Siemens TC65, wraz ze wszystkimi niezbędnymi do jej działania układami i elementami, oraz dostawka zawierająca odbiornik GPS wraz z anteną. Moduł TC65 do swojego działania wymaga odpowiedniego napięcia zasilania, które zapewnia zainstalowany na płycie drukowanej stabilizator

LM350, anteny GSM zainstalowanej na płycie i połączonej z modulem za pomocą odpowiedniej przejściówki, oraz aktywnej karty SIM, montowanej w specjalnym gnieździe. Dostawka GPS składa się ze stabilizatora liniowego LE33CZ pełniącego rolę zasilacza, oraz modułu RGPSM002 połączonego taśmą z odpowiednim gniazdem na płycie drukowanej dostawki. Obie płytki połączone są ze sobą przewodami, które zapewniają zasilanie oraz komunikację odbiornika GPS z modulem TC65.

Część stacjonarna składa się podłączonego do Internetu komputera z zainstalowanym oprogramowaniem należącym do systemu nadzoru. W celu realizacji tej części projektu posłużono się darmową usługą chmury obliczeniowej firmy Amazon – AWS (*Amazon Web Services*) [7]. Oprogramowanie systemu nadzoru to serwer HTTP z możliwością uruchamiania aplikacji sieciowych Java, aplikacja pełniąca rolę usługi oczekującej na połączenie ze strony układu zdalnej lokalizacji, oraz baza danych PostgreSQL, będąca magazynem dla danych konfiguracji i lokalizacji urządzeń. Schemat struktury systemu przedstawiono na rysunku 1.1.



Rys.1.1. Schemat struktury systemu lokalizacji

Układ zdalnej lokalizacji z modulem TC65 ustala swoją pozycję za pomocą dołączonego odbiornika GPS. Odbiornik komunikuje się z modulem poprzez protokół NMEA [6]. Po określeniu swojej pozycji, moduł inicjuje połączenie z systemem nadzoru, wykorzystując pakietową transmisję danych GPRS. Podczas transmisji przesyłane są dane takie jak: identyfikator urządzenia, określona pozycja, data, czas oraz aktualna prędkość. Pozycja określana i wysyłana jest co pewien czas, wedle ustawień użytkownika. Transmisja lokalizacji trwa tak długo, jak długo włączony jest układ.

Aplikacja systemu nadzoru oczekuje połączenia z wykorzystaniem protokołu HTTP ze strony urządzenia lokalizującego. Po nawiązaniu takiego połączenia odbiera przesyłane przez moduł dane, oraz zapisuje je w odpowiednich tabelach bazy danych PostgreSQL. Aplikacja ta jest także interfejsem sterującym pracą układu lokalizacji. Umożliwia ona także podgląd aktualnej pozycji układu lokalizacji, oraz wizualizację zapisanej trasy ruchu modułu, jeśli taki nastąpił. Jest to wygodny w użyciu interfejs dla osoby pełniącej rolę administratora systemu lokalizacji.

## Rozdział 2

### 2. Wykorzystane technologie

W rozdziale tym przedstawione zostaną technologie, które posłużyły do realizacji celów pracy. Rozdział podzielony został na trzy części. Pierwszy podrozdział dotyczy szeroko rozumianej teorii związanej z prezentowanymi tutaj technologiami. Przedstawione zostaną technologie GPS oraz GPRS. W drugim podrozdziale przedstawiono informacje dotyczące sprzętu używanego w pracy. Dokonano prezentacji oraz wyboru modułów GPS i GPRS. Przedstawiono także realizację fizycznej infrastruktury sieciowej za pomocą usługi AWS firmy Amazon. Ostatni podrozdział zawiera opis technologii programowych i oprogramowania wykorzystanego w pracy. Przedstawiono język programowania Java, w szczególności platformy Java Micro Edition i Java Enterprise Edition, system relacyjnych baz danych PostgreSQL, oraz protokół HTTP.

#### 2.1. Podstawy teoretyczne

Niniejsza praca opiera się na technologii pozycjonowania GPS, oraz pakietowej transmisji danych w sieciach GSM – GPRS. W rozdziale tym przedstawiono podstawy teoretyczne pozycjonowania GPS, podział na segmenty oraz sposób użytkowania systemu. Omówiono także technologię GPRS, strukturę jej sieci, protokołów oraz klasy i tryby urządzeń pracujących w tej technologii.

##### 2.1.1. Pozycjonowanie GPS [11]

GPS, pełna nazwa to GPS-NAVSTAR (ang. *Global Positioning System – NAVigation Signal Timing And Ranging*) – system nawigacji satelitarnej, jeden z kilku, obejmujący swoim zasięgiem całą kulę ziemską. Zbudowany przez Departament Obrony USA. Na system GPS składają się trzy segmenty:

- segment kosmiczny,
- segment kontroli,
- segment użytkownika.

Działa na zasadzie pomiaru czasu dotarcia sygnału radiowego, wysyłanego z satelitów, do odbiornika. Znając prędkość fal elektromagnetycznych oraz dokładny czas wysłania sygnału można obliczyć odległość satelitów od odbiornika. W sygnale GPS znajduje się informacja o układzie satelitów na niebie (tzw. almanach) oraz informacja o ich teoretycznej drodze i odchyleniach od niej (tzw. efemeryda). Po włączeniu odbiornik GPS aktualizuje te informacje w pamięci wewnętrznej. Następnie, wykorzystując zgromadzone dane, ustala swoją odległość od poszczególnych satelitów. Wykonując odpowiednie przekształcenia matematyczne, mikroprocesor odbiornika oblicza pozycję geograficzną (długość, szerokość oraz wysokość elipsoidalną) i następnie podaje ją w wybranym układzie odniesienia. Zwykle jest to WGS-84.

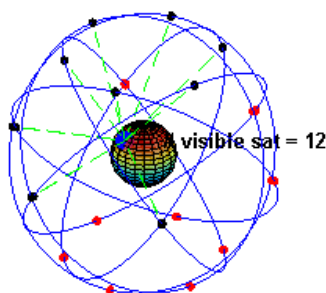
Utrzymaniem i zarządzaniem systemu GPS zajmuje się Departament Obrony Stanów Zjednoczonych. Korzystać z jego usług może każdy, kto posiada odpowiedni odbiornik GPS.

Odbiorniki takie produkowane są przez niezależne firmy trzecie. System GPS jest darmowy i wedle polityki Stanów Zjednoczonych taki ma pozostać.

## Segmenty systemu GPS:

### 1. segment kosmiczny

Składa się z około 28 aktywnych satelitów (pozostałe to satelity zapasowe bądź zużyte), okrążających Ziemię w ciągu 12 godzin. Ich układ zapewnia widoczność co najmniej 4 z nich, dzięki czemu system pracuje poprawnie na całej kuli ziemskiej. Na rysunku 2.1 przedstawiono poglądowy schemat rozmieszczenia satelitów wokół Ziemi.



Rys 2.1 – Poglądowy schemat rozmieszczenia satelitów GPS wokół Ziemi

### 2. segment kontroli

Pięć stacji nadzorczych, umieszczonych w miarę możliwości równomiernie wokół Ziemi, w pasie równikowym. Do zadań stacji kontroli należą:

- ciągły odbiór sygnałów nadawanych z satelitów,
- kontrola poprawności działania satelitów,
- komunikacja z satelitami,
- ustalanie poprawek jonosferycznych i pomiaru czasu.

### 3. segment użytkownika

Segment ten składa się z najróżniejszych odbiorników GPS, zarówno wojskowych jak i cywilnych.

System GPS pracuje na dwóch nośnych o częstotliwościach  $L1 = 1575,42$  MHz oraz  $L2 = 1227,6$  MHz. Porównanie różnicy faz odebranych sygnałów umożliwia dokładne wyznaczenie czasu propagacji sygnałów, który ulega zmianom w wyniku różnorodnego wpływu jonosfery, jednak nie w stopniu tak dużym aby uniemożliwić określenie współrzędnych. Dzięki systemowi DGPS (Differential GPS) użytkownicy cywilni otrzymują przybliżoną poprawkę jonosferyczną z naziemnych nadajników DGPS. Poprawka ta jest także zawarta w depeszy nawigacyjnej przesyłanej przez satelity GPS.

System DGPS wykorzystuje do działania naziemne stacje bazowe (o znanej i stałej pozycji). Znajdująca się niedaleko odbiornika stacja przesyła do niego dane różnicowe, na podstawie których odbiornik może dokonać poprawy niedokładnej lokalizacji wynikającej z błędów propagacji sygnału na drodze między satelitami a odbiornikiem.

Podobnym systemem korekcji błędów pozycji jest system WAAS/EGNOS. Działa on na zasadzie zbliżonej do DGPS z tą różnicą, że poprawki do odbiorników przesyłane są przez geostacjonarne satelity.



Niedokładność wyznaczenia pozycji względem położenia satelitów nazywa się „rozmyciem dokładności” DOP (ang. Dilution of Precision). Wyróżnia się kilka rodzajów rozmyć:

- GDOP – wyraża dokładność lokalizacji punktu w 4 wymiarach (3 wymiary przestrzenne oraz czas),
- HDOP – dokładność określenia współrzędnych płaskich,
- VDOP – dokładność określenia wysokości,
- TDOP – dokładność pomiaru czasu,
- PDOP – współczynnik określający stosunek pomiędzy błędem wyznaczania pozycji użytkownika, a błędem aktualnej pozycji satelity.

Jeśli wartość któregoś z parametrów wynosi 0, oznacza to, że w danej chwili określenie lokalizacji jest niemożliwe z powodu zakłóceń, słabego sygnału z satelitów, lub za małą liczbę widocznych aktualnie satelitów. Czym mniejsza wartość tego współczynnika (ale jednak większa od zera) tym określenie lokalizacji uznaje się za dokładniejsze. Określono, że dla  $DOP < 2,3$  sygnał jest znakomity, dla  $< 4,6$  dobry, a dla  $< 7,8$  umiarkowany. Dla wartości powyżej 8, sygnał uznaje się za słaby bądź bardzo słaby.

### 2.1.2. Użytkowanie GPS

Korzystanie z systemu pozycjonowania GPS jest darmowe i wymaga jedynie posiadania specjalnego odbiornika. Można wyróżnić dwa typy odbiorników GPS: wewnętrzne i zewnętrzne. Odbiorniki wewnętrzne to urządzenia zintegrowane z innymi urządzeniami, takimi jak nawigacje samochodowe, smartfony czy laptopy. Zaletą takiego rozwiązania jest przede wszystkim wygoda. Urządzenie, które wymaga wiedzy o aktualnej pozycji ma wbudowany odbiornik, więc nie trzeba zaopatrzć go w odbiornik zewnętrzny, ani konfigurować połączenia. Wadą takiego rozwiązania jest możliwość wykorzystania odbiornika jedynie przez urządzenie w które został wbudowany. Niedogodność ta nie dotyczy odbiorników zewnętrznych. Są to zazwyczaj urządzenia niewiele większe od pudełka zapalek. Połączenie z urządzeniem głównym realizowane jest najczęściej za pomocą bezprzewodowego łącza Bluetooth bądź przewodowo z wykorzystaniem standardu RS-232 lub USB. Taki sposób połączenia wymaga określenia sposobu wymiany informacji między odbiornikiem a urządzeniem. Sposób przekazywania danych nosi nazwę protokołu komunikacji. Odbiorniki GPS najczęściej komunikują się z pozostałymi urządzeniami poprzez protokół NMEA 0183. Jest to protokół opublikowany przez National Marine Electronics Association, oraz wykorzystywany m.in. podczas komunikacji między morskimi urządzeniami elektronicznymi w które wyposażone są nowoczesne jednostki żeglugi. NMEA jest protokołem całkowicie tekstowym gdzie dane przesyłane są za pomocą kodu ASCII, w postaci pakietów zwanych „zdaniami”. Pojedyncze zdanie to sekwencja znaków oddzielonych od siebie przecinkami, rozpoczynająca się od znaku „\$” oraz identyfikatora zdania, który określa jego typ i przeznaczenie. Sekwencja kończy się symbolami powrotu karetki i przesuwu strony (<CR><LF>).

Na rysunku 2.2 przedstawiono przykładowe zdanie protokołu NMEA.

```
$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
```

Rys 2.2 – Przykładowa sekwencja NMEA z odbiornika GPS

Kolejnymi danymi tej przykładowej sekwencji są:

- \$GPGGA – identyfikator zdania (*Global Positioning System Fixed Data*),
- 123519 – czas UTC w formacie GGMMSS,
- 4807.038 – szerokość geograficzna w postaci DDMM.MMM (tutaj oznaczająca 48° 7,038'),
- N – kierunek szerokości geograficznej,
- 01131.000 – długość geograficzna w postaci DDDMM.MMM (tutaj oznaczająca 11° 31'),
- E – kierunek długości geograficznej,
- 1 – oznaczenie trybu odbiornika, tzw. fix (tutaj: tryb GPS SPS, podana pozycja jest poprawna),
- 08 – liczba satelitów użytych do określenia pozycji,
- 0.9 – wartość współczynnika HDOP,
- 545.4 – wysokość nad poziomem morza (geoidalna),
- M – jednostka wysokości npm. (tutaj: metry),
- 46.9 – różnica między wysokością geoidalną a elipsoidalną,
- M – jednostka różnicy wysokości,
- \*47 – suma kontrolna całego zdania,

gdzie:

G – godzina,

M – minuta,

S – sekunda,

D – stopień.

Opis poszczególnych części protokołu został opracowany na podstawie dokumentu [6].

Istnieją także inne protokoły komunikacji z odbiornikami GPS takie jak np. binarny protokół firmy SiRF, wykorzystywany do komunikacji z chipsetami SiRF Star III, jednak w praktyce są one niezmiernie rzadko wykorzystywane i zwykle można je wyłączyć, umożliwiając tym samym komunikację w protokole NMEA.

Ważnym czynnikiem dotyczącym wszystkich systemów bezprzewodowych jest zasięg oraz poziom mocy odbieranego sygnału. Odbiorniki GPS transmitują dane lokalizacyjne niezależnie od tego czy znajdują się w zasięgu satelitów nawigacyjnych oraz potrafią poprawnie określić swoją pozycję czy też nie. W związku z tym zaszła potrzeba informowania o tym, czy przesłana pozycja jest pozycją prawidłową. Prawidłowość określenia pozycji nosi miano „fixu” GPS. Fix jest informacją określającą tryb pracy modułu GPS. Podczas gdy odbiornik znajduje się w zasięgu mniej niż 3 satelitów nawigacyjnych lub moc odbieranego z satelitów sygnału jest zbyt mała, nie można poprawnie określić położenia. W takim przypadku dane lokalizacyjne pochodzące z odbiornika uznaje się za nieprawidłowe. Potocznie określa się taką sytuację „brakiem fixu”, lub „brakiem prawidłowego fixu” (*ang. no valid fix*). Odbiornik który znajduje się w zasięgu 3 satelitów, gdy ich poziom sygnału jest odpowiednio wysoki, potrafi określić swoją pozycję jedynie w dwóch wymiarach (2D), tj. długość oraz szerokość geograficzną. Dane lokalizacyjne zawierające współrzędne geograficzne uznaje się wtedy za prawidłowe, natomiast dane o wysokości za nieprawidłowe. Ostatnią możliwą do uzyskania sytuacją jest czas, w którym odbiornik pozostaje w zasięgu 4 lub więcej satelitów. Umożliwia to pełne określenie pozycji w trzech wymiarach (3D).

Praca zawiera prostą obsługę kilku typów zdań protokołu NMEA, która zostanie przedstawiona w jednym z kolejnych rozdziałów.

### 2.1.3. Transmisja GSM/GPRS

Wprowadzenie technologii GPRS przez operatorów sieci telefonii komórkowych miało na celu wyjście naprzeciw coraz większym zapotrzebowaniom na mobilny dostęp do Internetu. Technologia ta stała się początkiem sieci określanej jako 2,5G i będącej podstawą do wprowadzenia sieci trzeciej generacji (3G), jaką jest UMTS (*ang. Universal Mobile Telecommunications System*).

Technologia GPRS korzystająca z pakietowej transmisji danych, przyspieszyła prędkość przesyłania danych z 14,4 kbit/s dla transmisji z komutacją łączy (CSD) do nawet 171,2 kbit/s przy założeniu idealnych warunków połączenia z siecią GPRS. W praktyce prędkość transmisji oscyluje przy 40 kbit/s wykorzystując trzy szczeliny czasowe dla odbierania danych oraz jedną szczelinę czasową dla wysyłania. Najważniejszą cechą i zaletą transmisji pakietowej jest rozliczanie się z operatorem za ilość przesyłanych danych, a nie, jak było to w przypadku transmisji z komutacją łączy, za czas połączenia. Zaleta ta zadecydowała o sukcesie transmisji pakietowej i przyczyniła się do powstawania kolejnych generacji tej technologii, m.in. technologii EDGE (*ang. Enhanced Data rates for GSM Evolution*).

Usługi transmisji danych w sieci GPRS można podzielić na dwa rodzaje. Pierwszym jest grupa usług polegająca na transmisji danych od punktu do punktu i oznaczana jest symbolem PTP (*ang. Point To Point*). Wśród usług PTP można wyróżnić usługi połączeniowe PTP-CONS (*ang. Point To Point Connection Oriented Network Service*) oraz na bezpołączeniowe usługi sieciowe PTP-CLNS (*ang. Point To Point Connectionless Network Service*).

Usługi PTP-CONS polegają na ustanowieniu logicznego połączenia pomiędzy użytkownikami. Przykładem wykorzystania takich połączeń mogą być operacje związane z obsługą kart kredytowych, czy monitoring elektroniczny.

Usługi PTP-CLNS związana są z przesyłaniem pakietów IP pomiędzy użytkownikami. Przykładem może być przesyłanie poczty elektronicznej lub korzystanie z sieci WWW (*ang. World Wide Web*).

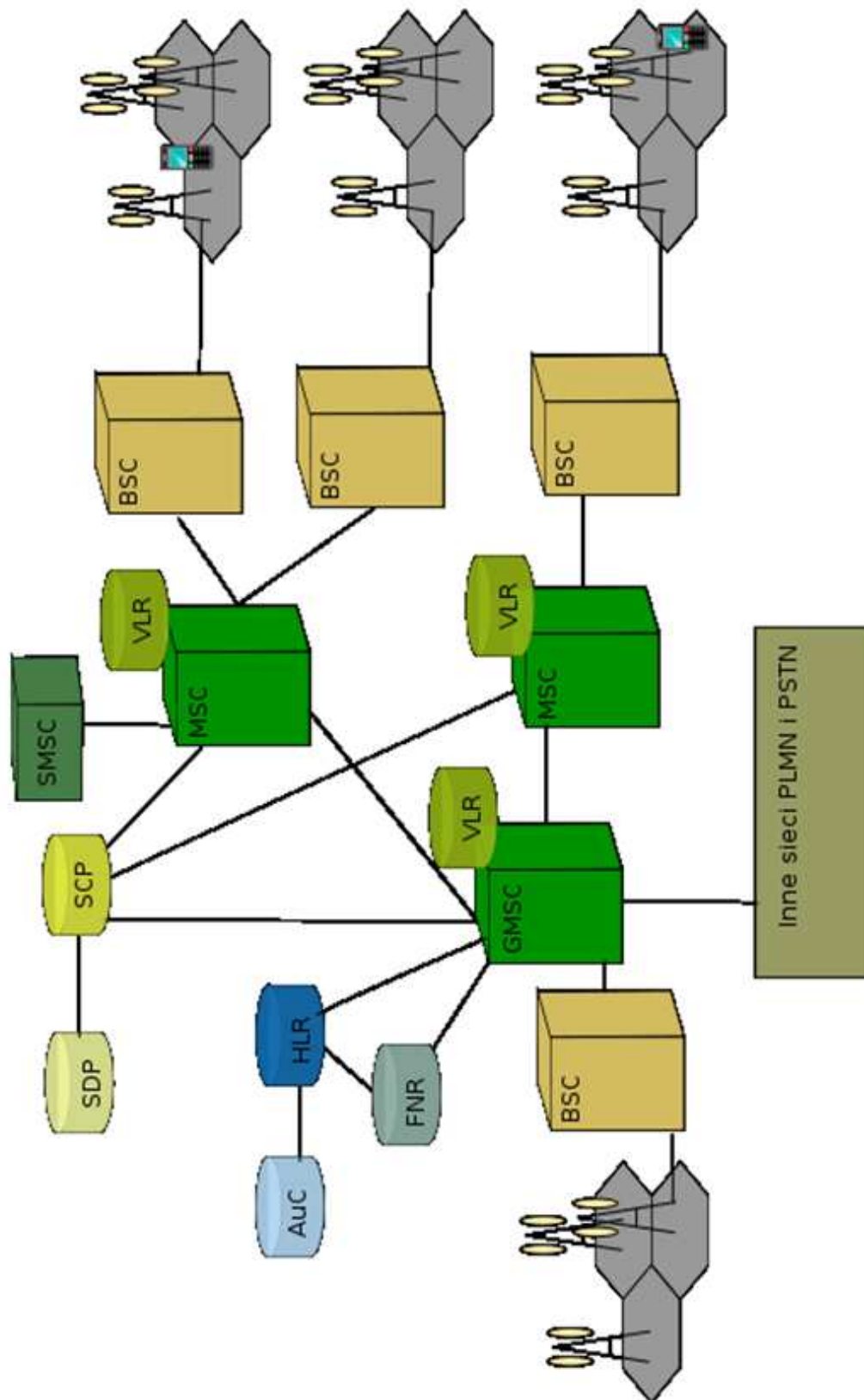
Druga grupa to usługi związane z transmisją od jednego użytkownika do wielu i oznaczana jest symbolem PTM (*ang. Point To Multipoint*). Usługę PTM można podzielić na usługi PTM-M (*ang. Point To Multipoint Multicast*) oraz PTM-G (*ang. Point To Multipoint Group*).

W usłudze PTM-M użytkownicy, lub grupa użytkowników, którzy mają być adresatami danych muszą zostać zdefiniowani. Przykładem takich usług są np.: informacje na temat pogody czy ruchu drogowego.

W przypadku usługi PTM-G dane przesyłane są do wszystkich użytkowników w czasie rzeczywistym. Dodatkowo użytkownicy muszą wyrazić zgodę na uczestnictwo w takim połączeniu. Usługę tą wykorzystuje się np.: podczas połączeń konferencyjnych. Standard GPRS jest aktualnie podstawowym standardem przesyłania danych w sieciach komórkowych. Każdy nowo wyprodukowany telefon komórkowy oraz każda stacja bazowa posiadają zaimplementowaną obsługę GPRS, dzięki czemu jest to standard dostępny na całym świecie.

Wprowadzenie usługi GPRS nie wymaga tworzenia infrastruktury sieci komórkowej od zera. Wykorzystuje się istniejącą sieć GSM rozbudowaną o komponenty odpowiedzialne za obsługę pakietowej transmisji danych GPRS.

Na rys.2.3 przedstawiono schemat struktury sieci GSM.



*Rys 2.3 – Struktura sieci GSM*

Stacja Bazowa jest elementem sieci (tzw. stacją przekąźnikową), będącym interfejsem pomiędzy telefonem komórkowym użytkownika a siecią GSM. Dzięki systemowi anten transmituje ona i odbiera na kilku częstotliwościach (liczba ta zależy od konfiguracji sprzętowej i oprogramowania) sygnał zakodowany cyfrowo. Częstotliwości używane przez stacje bazowe obsługujące sąsiadujące ze sobą komórki różnią się, tak aby nie dochodziło do interferencji fal radiowych między nimi. Zwykle od kilkudziesięciu do kilkuset stacji bazowych jest podłączonych i obsługiwanych przez jeden Kontroler Stacji Bazowych.

Kontroler Stacji Bazowych (BSC) jest elementem sieci, który odpowiada za zarządzanie stacjami bazowymi, oraz transmisję danych pomiędzy stacjami bazowymi a resztą sieci. Z poziomu Kontrolera Operator zarządza radiową częścią sieci, poprzez zmianę parametrów poszczególnych stacji bazowych. BSC odpowiada także za przydzielanie telefonowi komórkowemu wolnej szczeliny czasowej na odpowiedniej częstotliwości oraz za śledzenie jakości rozmowy. W przypadku jej pogorszenia, np.: podczas gdy abonent oddala się od obsługującej go stacji bazowej, zostanie przydzielona mu inna częstotliwość, obsługiwana przez inną stację bazową bliżej abonenta, oraz odpowiednia szczelina czasowa. Mechanizm ten nosi nazwę „handover”. Kilka BSC jest podłączonych do jednego MSC.

Mobile Switching Centre (MSC) to cyfrowa centrala telefoniczna stworzona do pracy w sieci GSM. Jest odpowiedzialna za zestawianie połączeń oraz koordynuje współpracę pomiędzy elementami sieci. Ilość MSC w sieci zależy od ilości abonentów i generowanego przez nich obciążenia sieci.

Gateway Mobile Switching Centre (GMSC) jest centralą MSC z dodatkową funkcjonalnością odpowiedzialną za kontaktowanie się z elementem sieci zwanym HLR. Każda rozmowa podczas zestawiania połączenia do abonenta danej sieci musi być odpowiednio trasowana do jednego z GMSC należącego do niej (nawet gdy abonent jest w tym czasie w roamingu w innej sieci) w celu uzyskania informacji o użytkowniku, którego numer wybrano w celu rozpoczęcia rozmowy.

Home Location Register (HLR - rejestr stacji własnych) to baza danych, w której przechowywane są informacje o abonentach, należących do danej sieci. Między innymi numery IMSI, MSISDN, informacje o subskrybowanych przez użytkownika usługach, informacje o MSC, które aktualnie obsługuje abonent czy informacje o jego statusie (np. telefon jest wyłączony, telefon jest włączony do sieci itp.).

Authentication Centre (AuC) jest elementem sieci, który przechowuje dane abonentów danej sieci, na podstawie których dokonuje się procesu uwierzytelnienia numeru IMSI i zezwala danemu abonentowi logującemu się do sieci na korzystanie z zasobów radiowych. Authentication Centre powiązany jest z rejestrem HLR, a ich liczba zależy od ilości użytkowników danej sieci.

Visitor Location Register (VLR - rejestr abonentów przyjezdnych) jest bazą danych związaną z centrum MSC. W sieci istnieją zawsze pary MSC-VLR. W bazie tej znajdują się informacje o abonentach, którzy w danym momencie są na obszarze obsługiwanym przez to właśnie MSC. Część z tych informacji jest kopiowana z HLR w momencie, gdy abonent pojawia się w "zasięgu" obejmowanym przez dany MSC, inne, takie jak jego lokalizacja są przenoszone później.

Flexible Number Register (FNR) jest opcjonalnym elementem sieci, wykorzystywanym podczas wykorzystywania mechanizmu „Number Portability”, znanym jako "przenoszenie numeru pomiędzy operatorami". Jest to baza danych zawierająca informacje o wszystkich abonentach w sieciach GSM danego kraju. Zawiera informacje o numerze abonenta, który dzięki temu można zachować, przechodząc do innej sieci; aktualnym operatorze, którego abonent jest klientem; oraz numer, pod który należy przekierować rozmowę do tego operatora.

SMS Center (SMSC) - element sieci uczestniczący w przesyłaniu SMS-ów pomiędzy abonentami i przechowujący te wiadomości, które nie mogą być w danej chwili dostarczone. Np.: gdy abonent jest poza zasięgiem sieci lub ma wyłączony telefon.

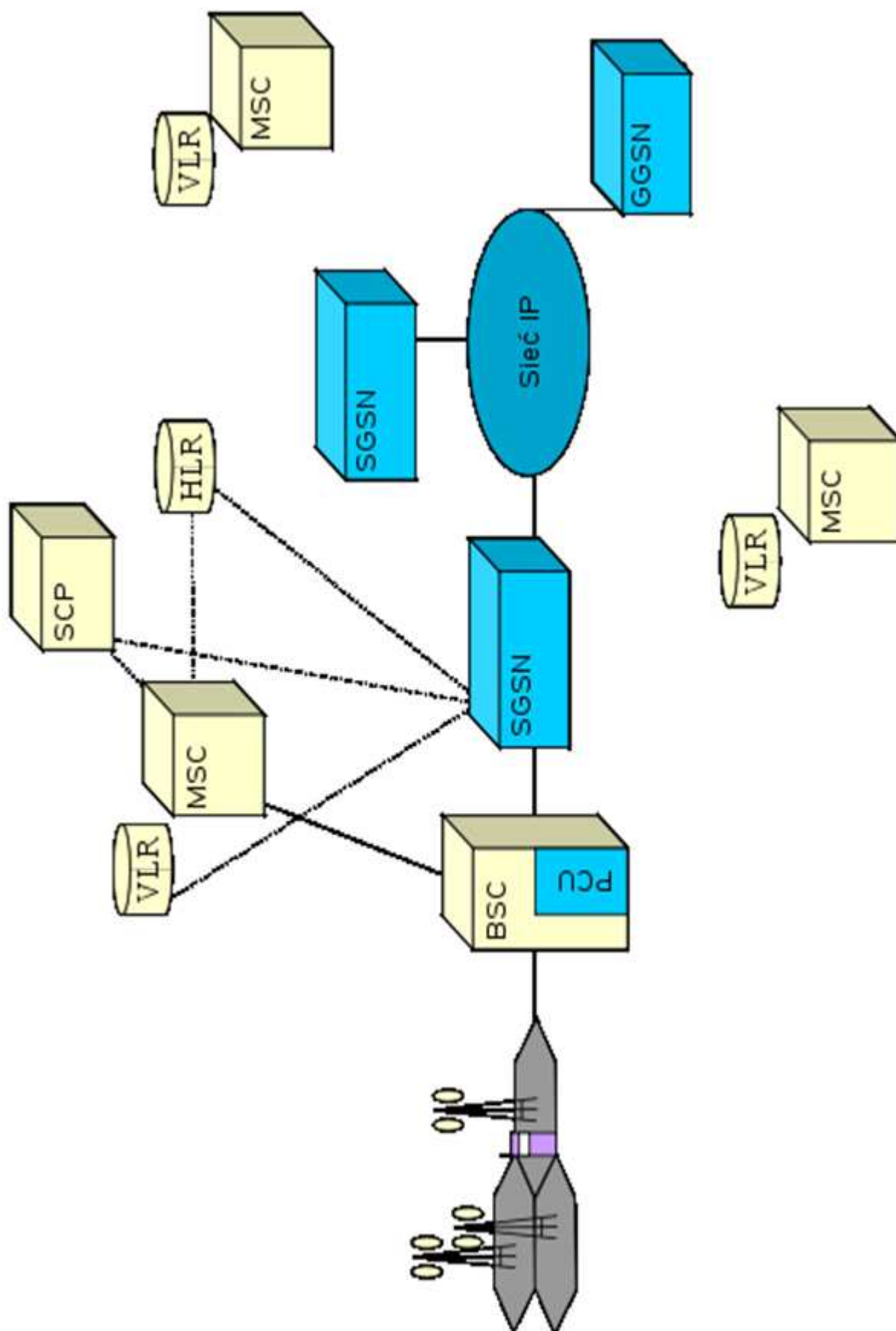
Service Control Point (SCP) jest to element sieci, na którym oparte są sieci inteligentne. Działają na nim serwisy związane są z usługami dodatkowymi, które mogą być wykupione przez abonenta, np. Virtual Private Network (VPN) lub Prepaid. SCP potrafi komunikować się z MSC dzięki protokołom SS7 i może wpływać na zestawianą rozmowę, np. zmieniając sposób naliczania opłat, przekierowując do innego numeru, lub dołączając do rozmowy dodatkowego abonenta. MSC może informować SCP o różnych zdarzeniach w sieci związanych z rozmową (np. abonent, do którego kierowana jest rozmowa, aktualnie rozmawia, odrzucił połączenie, nie podnosi słuchawki itp.) i na tej podstawie, podejmowane są dalsze decyzje w stosunku do tej rozmowy.

Service Data Point (SDP) to baza danych, zawierająca informacje o abonentach, które są wykorzystywane przez programy działające na platformie Sieci Inteligentnych.: np. program Prepaid przechowuje tam informacje o ilości dostępnych impulsów.

#### **2.1.4. GPRS jako dodatek do GSM**

GPRS jako sposób pakietowego przesyłania danych w sieciach GSM, jest jej rozwinięciem, niewymagającym kompletnej zmiany infrastruktury sieciowej, a jedynie dołożenie sprzętu dedykowanego do obsługi wspomnianego standardu. Jest to swojego rodzaju ewolucja GSM (sieci 2G), nazywana często siecią 2,5G, ponieważ stanowi ona drogę w kierunku sieci w standardzie trzeciej generacji (3G). Specyfikacja GPRS rozwijana jest jako wydzielona część standardu GSM przez konsorcjum standaryzacyjne 3GPP.

Na rys. 2.4 przedstawiono strukturę sieci GPRS.



Rys 2.4 – Struktura sieci GPRS

SGSN (*ang. Serving GPRS Support Node*) to element sieci GPRS odpowiedzialny za zarządzanie terminalami znajdującymi się na kontrolowanym przez siebie obszarze. Teren ten podzielony jest na części zwane „Routing Area” (obszar kilkudziesięciu komórek GSM). Jeśli terminal zmienia lokalizację i znajdzie się w innym Routing Area, to fakt ten zostanie odnotowany w SGSN. Element ten jest także odpowiedzialny za uwierzytelnianie terminali chcących włączyć się do sieci. Podczas transmisji uczestniczy on w przesyłaniu pakietów pomiędzy terminalem a innymi elementami sieci GPRS. Ilość SGSN w sieci zależy od ruchu pakietowego generowanego przez abonentów.

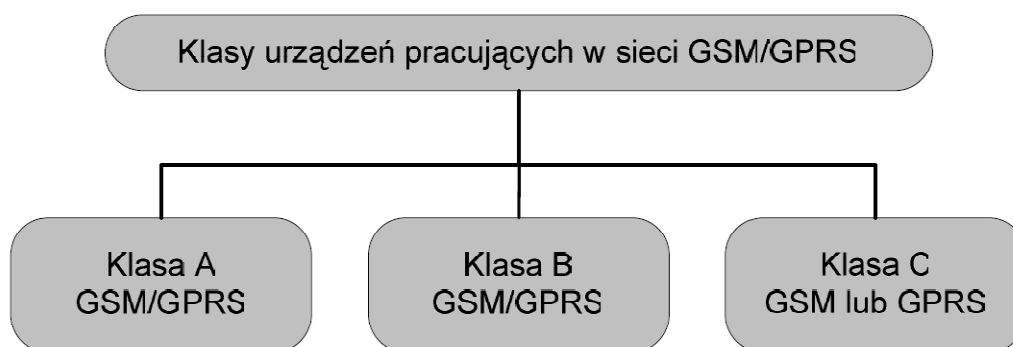
GGSN (*ang. Gateway GPRS Support Node*) jest elementem sieci działającym jak router, który łączy sieć GPRS i sieć zewnętrzną (np. Internet). W czasie gdy użytkownik terminala chce uzyskać dostęp do zasobów zewnętrznej sieci, GGSN przydziela mu adres IP (z własnej puli adresów lub z listy dostarczonej przez serwer z zewnętrznej sieci), dodatkowo na czas sesji aktywuje tzw. protokół „PDP context”, który zawiera numer IMSI terminala, przydzielony mu adres IP oraz adres IP SGSN, które kontroluje teren, na którym aktualnie znajduje się terminal. PDP context jest wykorzystywany podczas trasowania pakietów przychodzących z zewnętrznych sieci.

PCU (*ang. Packet Control Unit*) to element odpowiedzialny za prawidłową obsługę ruchu pakietowego w radiowej części sieci. Przydziela on terminalom GPRS wolne kanały radiowe, buforuje dane przesłane przez SGSN, przekazuje je do odpowiedniej stacji bazowej, dodając informację, która umożliwia terminalowi identyfikację własnych danych. PCU może być zaimplementowany jako dodatkowy sprzęt w BSC lub jako niezależny element sieci obsługujący jedno lub więcej BSC.

### 2.1.5. Klasy i tryby pracy urządzeń GSM/GPRS

Europejski Instytut Norm Telekomunikacyjnych ETSI (*ang. European Telecommunications Standards Institute*) definiuje trzy zasadnicze klasy urządzeń pracujących w sieci GSM/GPRS. Klasy te oznaczane są symbolami A, B i C.

Na rys.2.5 przedstawiono podział klas urządzeń sieci GSM/GPRS.



Rys 2.5 – Klasy urządzeń sieci GSM/GPRS

Urządzenia klasy A mogą być podłączone do sieci GPRS i GSM jednocześnie. Mogą one w tym samym czasie odbierać w sieci GSM rozmowy, dane, wiadomości SMS oraz korzystać z sieci GPRS odbierając lub wysyłając dane. Aby było to możliwe urządzenie musi monitorować obie sieci jednocześnie.

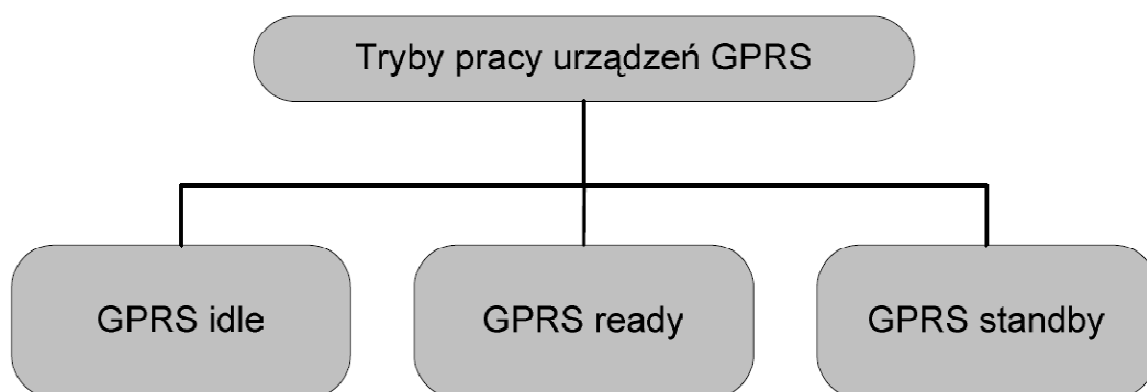


Urządzenia klasy B mogą działać zarówno w sieci GSM jak i GPRS, lecz nie wspierają współdzielenia transmisji dla obu sieci. Jeżeli połączenie GPRS jest aktywne, urządzenie nie ma możliwości odebrania połączenia GSM i vice versa.

Urządzenia klasy C posiadają możliwość obsługi obu sieci, jednakże w danej chwili włączona może być tylko jedna. Przez to, jeżeli urządzenie zostało podłączone do sieci GPRS jest jednocześnie odłączone od sieci GSM.

Urządzenia GPRS mogą pracować w jednym z trzech trybów: spoczynku, gotowości i oczekiwania.

Na rysunku 2.6 przedstawiono schemat blokowy trybów pracy w sieci GPRS.



Rys 2.6 – Tryby pracy urządzeń w sieci GPRS

Urządzenia korzystające z sieci GPRS mogą pracować w kilku trybach pracy.

Pierwszym trybem pracy urządzenia GPRS jest tryb bezczynności (ang. Idle) W tym trybie urządzenie jest podłączone do sieci GSM, nastawione jest na odbieranie danych w trybie CSD (ang. Circuit Switched Data), jednakże nie współpracuje z siecią GPRS przy jednoczesnym posiadaniu pełnej funkcjonalności tej sieci.

W trybie gotowości (ang. Ready) urządzenie może aktywować obsługę protokołu PDP (ang. Packet Data Protocol), który pozwala na ustanowienie sesji połączenia pomiędzy urządzeniem mobilnym, a zewnętrzną siecią w celu przesłania danych. Kiedy PDP zostanie aktywowany następuje transmisja danych, po której urządzenie przechodzi w tryb Standby.

Trzecim trybem pracy jest tryb oczekiwania (ang. Standby), w którym urządzenie jest podłączone do sieci GPRS, jednak nie realizuje transmisji danych. W sytuacji, kiedy dane oczekują na odebranie przez urządzenie sieć informuje o tym MS, które przechodzi w tryb gotowości i aktywuje protokół PDP.

## 2.2. Technologie sprzętowe

W niniejszym rozdziale przedstawiono technologie sprzętowe, wykorzystane do realizacji układu zdalnej lokalizacji obiektów ruchomych. Przedstawiono technologie takie jak sprzętowy odbiórnik systemu GPS, urządzenia nadawczo-odbiorcze pakietowej transmisji danych GPRS oraz infrastrukturę serwerowo-sieciową obsługującą aplikację systemu nadzoru i magazynu danych na potrzeby realizacji pracy.

### 2.2.1. Wybór odbiornika GPS [12]

Przegląd odbiorników GPS wykonany został w oparciu o ofertę hurtowni elektronicznej TME.eu[12] na bazie dostępnych aktualnie urządzeń.

#### ACTE ACTivE AGPS50A

Odbiornik AGPS50A to urządzenie w formie anteny odbiorczej w obudowie zakończonej wtykiem u-Molex.



Rys. 2.7 – Odbiornik AGPS50A

Parametry odbiornika AGPS50A:

- Dokładność pomiaru pozycji  $\pm 3\text{m}$
- Dokładność pomiaru czasu  $\pm 60\text{ns}$
- Dokładność pomiaru prędkości  $\pm 0,1\text{m/s}$
- Czułość odbiornika  $-165\text{dBm}$
- Napięcie zasilania  $5\text{V DC}$
- Protokół komunikacyjny NMEA 0183
- Liczba kanałów 22
- Rodzaj złącza u-Molex
- hot start  $<1\text{s}$
- cold start  $<36\text{s}$

#### HOPE GPS01

GPS01 to prosty miniaturowy odbiornik GPS charakteryzujący się niewielką ceną.



Rys. 2.8. – Odbiornik Hope GPS01

Parametry odbiornika GPS01:

- Dokładność pomiaru pozycji  $\pm 5\text{m}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Dokładność pomiaru prędkości  $\pm 0,1\text{m/s}$
- Interfejs UART
- Czułość odbiornika  $-157\text{dBm}$
- Napięcie zasilania 3.3V DC
- wyjście PPS
- Liczba kanałów 65

### HOPE GPS02

Odbiornik GPS02 to miniaturowy układ scalony do zastosowań w przemyśle.



Rys. 2.9 – Odbiornik Hope GPS02

Parametry odbiornika GPS02:

- Dokładność pomiaru pozycji  $\pm 5\text{m}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Dokładność pomiaru prędkości  $\pm 0,1\text{m/s}$
- Interfejs UART
- Czułość odbiornika  $-160\text{dBm}$
- Napięcie zasilania 3.3V DC
- Liczba kanałów 65
- filtr SAW
- wyjście PPS

### HOPE GPS03

GPS03 to odbiornik bardzo zbliżony parametrami do odbiornika GPS02.



Rys. 2.10 – Odbiornik Hope GPS03

Parametry odbiornika GPS03:

- Typ modułu komunikacyjnego      GPS
- Dokładność pomiaru pozycji       $\pm 5\text{m}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Dokładność pomiaru prędkości     $\pm 0,1\text{m/s}$
- Czułość odbiornika                 $-160\text{dBm}$
- Napięcie zasilania     $3.3\text{V DC}$
- Liczba kanałów      65
- filtr SAW
- wyjście PPS

#### **HOPE GPS04**

Odbiornik GPS04 to najnowszy odbiornik z rodziny GPS firmy Hope.



*Rys. 2.11. – Odbiornik Hope GPS04*

Parametry odbiornika GPS04:

- Dokładność pomiaru pozycji       $\pm 5\text{m}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Dokładność pomiaru prędkości     $\pm 0,1\text{m/s}$
- Interfejs      UART
- Czułość odbiornika                 $-160\text{dBm}$
- Napięcie zasilania     $3.3\text{V DC}$
- Liczba kanałów      65
- filtr SAW
- wyjście PPS

#### **FIBOCOM GTS-4E-60**

Fibocom GTS-4E-60 to miniaturowy odbiornik charakteryzujący się dużym zakresem napięcia zasilającego.



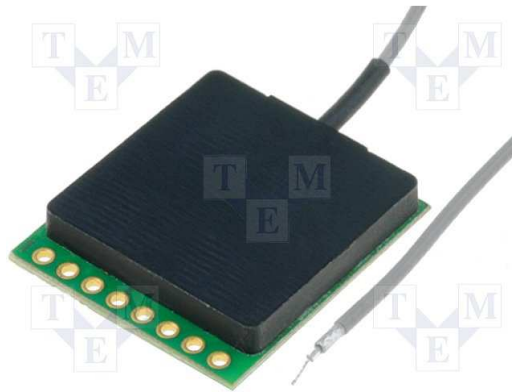
*Rys. 2.12. – Odbiornik Fibicom GTS-4E-60*

Parametry odbiornika GTS-4E-60:

- Interfejs UART
- Czułość odbiornika -161dBm
- Napięcie zasilania 2.7...5.5V DC
- Montaż SMD
- Obudowa LCC24
- Liczba kanałów 48
- chipset SiRF Star IV
- cold start <35s
- hot start <1s

### **ORIGIN ORG1306-R01**

ORG1306-R01 to najprostszy odbiornik z rodziny GPS firmy Origin.



*Rys. 2.13. – Odbiornik ORG1306-R01*

Parametry odbiornika ORG1306-R01:

- Dokładność pomiaru pozycji  $\pm 2\text{m}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Dokładność pomiaru prędkości  $\pm 0,01\text{m/s}$
- Czułość odbiornika -159dBm
- Napięcie zasilania 3.3...5.5V DC
- Protokół komunikacyjny NMEA 0183, SiRF
- Temperatura pracy -40...85°C
- Montaż SMD
- Liczba kanałów 12

### **ORIGIN ORG1308-R01-TR**

Odbiornik ORG1308-R01-TR to miniaturowy odbiornik GPS ze zintegrowanym złączem antenowy U.FL.



*Rys. 2.14. – Odbiornik Origin ORG1308-R01-TR*

Parametry odbiornika ORG1308-R01-TR;

- Dokładność pomiaru pozycji  $\pm 2\text{m}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Dokładność pomiaru prędkości  $\pm 0,01\text{m/s}$
- Czułość odbiornika  $-159\text{dBm}$
- Napięcie zasilania  $3.3...5.5\text{V DC}$
- Temperatura pracy  $-40...85^\circ\text{C}$
- Protokół komunikacyjny NMEA 0183, SiRF
- Liczba kanałów 12
- A-GPS
- SPI
- UART
- chipset SiRF Star III
- cold start  $<40\text{s}$
- warm start  $<32\text{s}$
- hot start  $<1\text{s}$

### **ORIGIN ORG1315-R01**



*Rys. 2.15. – Odbiornik Origin ORG1315-R01*

Parametry odbiornika ORG1315-R01:

- Dokładność pomiaru pozycji  $\pm 2\text{m}$
- Dokładność pomiaru prędkości  $\pm 0,01\text{m/s}$
- Dokładność pomiaru czasu  $\pm 1\mu\text{s}$
- Napięcie zasilania 3.3...5.5V DC
- Czułość odbiornika -159dBm
- Protokół komunikacyjny NMEA 0183, SiRF
- Temperatura pracy -40...85°C
- Liczba kanałów 12
- A-GPS
- SPI
- UART
- chipset SiRF Star III
- hot start <1s
- warm start <32s
- cold start <35s
- wbudowana antena

**RACEWOOD RGPSM002**



*Rys. 2.16. – Odbiornik Racewood RGPSM002*

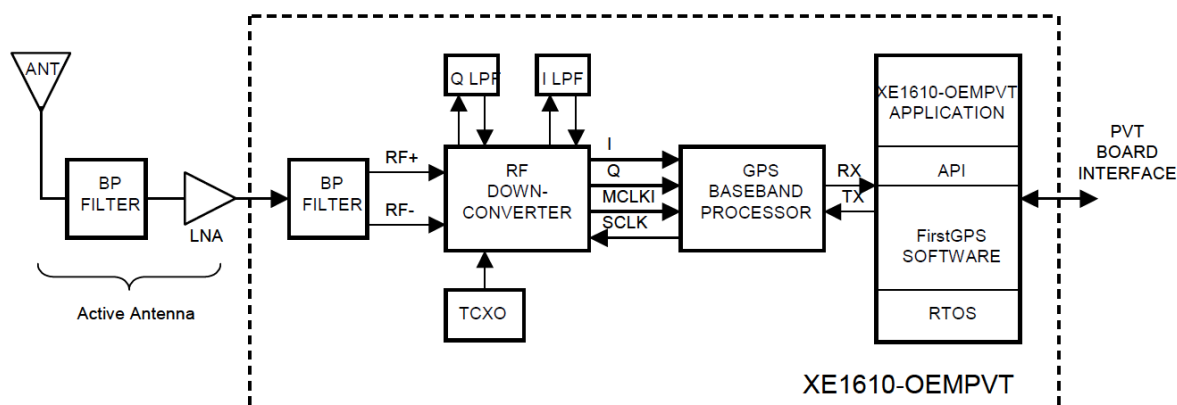
Parametry odbiornika RGPSM002:

- Dokładność pomiaru pozycji <5m
- Dokładność pomiaru czasu  $\pm 100\text{ns}$
- Czułość odbiornika -143dBm
- Napięcie zasilania 3...3.65V DC
- Protokół NMEA 0183
- Liczba kanałów 8
- UART
- chipset FirstGPS
- cold start <120s
- warm start <32s
- hot start <12s
- wyjście PPS

## 2.2.2. Odbiornik Racewood RGPSM002

Moduł RGPSM002 firmy Racewood to miniaturowy, w pełni autonomiczny odbiornik GPS zapewniający informacje o pozycji, prędkości oraz dokładnym czasie. Odbiornik charakteryzuje się bardzo niskim poborem prądu rzędu 19 mA, przy zasilaniu 3 V, oraz dużą dokładnością przesyłanych danych. Dodatkową cechą odbiornika jest także duża czułość, zapewniająca poprawne działanie nawet w silnie zabudowanych obszarach miejskich. Odbiornik jest wyposażony w prosty interfejs szeregowy w standardzie LVTTL 3V. Zaprojektowany został w celu uproszczenia integracji systemu GPS z systemami wbudowanymi. Znajduje zastosowanie w branży motoryzacyjnej, przy zarządzaniu flotą pojazdów, w urządzeniach PDA oraz usługach lokalizacyjnych.

Na rysunku 2.17 przedstawiono schemat blokowy odbiornika RGPSM002.



Rys. 2.17. – Schemat blokowy odbiornika RGPSM002

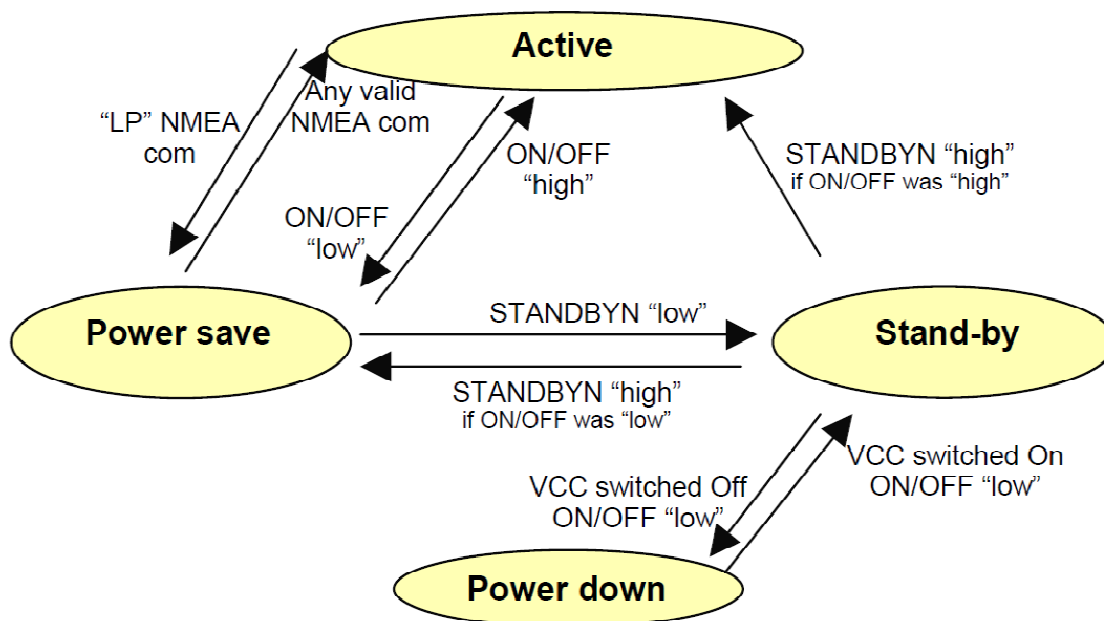
Na schemacie wyróżnić można kilka grup funkcyjnych. Pierwszą z nich jest grupa RF, aktywna antena oraz filtry wejściowe, odpowiedzialna za dostarczenie odpowiednio przygotowanego sygnału GPS do procesora sygnałowego. Kolejna grupa składa się przede wszystkim z procesora sygnałowego GPS, którego zadaniem jest przekształcenie odebranego sygnału GPS, do postaci protokołu NMEA. Procesor ten oparty jest na rdzeniu FirstGPS. Ostatnią grupą są interfejsy wejścia-wyjścia, służące do komunikacji odbiornika z systemem zewnętrznym.

Odbiornik RGPSM002 pracować może w czterech trybach zasilania:

- tryb Active – sekcja RF, system GPS, procesor oraz RTC włączone
- tryb Power Save – sekcja RF i GPS wyłączone
- tryb Stand-By – sekcja RF, GPS i zegar procesora wyłączone, RTC włączone na zasilaniu podstawowym
- tryb Power Down – sekcja RF, GPS, procesor wraz z zegarem wyłączone, główne zasilanie wyłączone, RTC włączony na zasilaniu podtrzymującym

Na rysunku 2.18 przedstawiono tryby zasilania odbiornika RGPSM002 oraz przejścia między nimi.





Rys. 2.18. – Tryby zasilania odbiornika RGPSM002

### 2.2.3. Transmisja danych do systemu nadzoru.

Transmisja zebranych danych do nadrzędnego systemu nadzoru to zadanie wymagające ciągłego lub okresowego połączenia z siecią, dającą dostęp do zasobów jakie oferuje system nadzoru. Najprostszym rozwiązaniem jest wykorzystanie sieci Internet, która w dobie bardzo szybkiego rozwoju telekomunikacji, dostępna jest prawie z każdego zakątka Ziemi. Możliwości dostępu do sieci Internet najczęściej sprowadzają się do połączeń kablowych, takich jak lokalne sieci osiedlowe (LAN) czy telewizyjne sieci kablowe TVK, jak również do dostępu bezprzewodowego – za pomocą technologii takich jak Wi-Fi, Wi-Max czy sieci telefonii komórkowej lub satelitarnej. Najpopularniejszym kanałem dostępu do sieci Internet na krótkich odcinkach od stacji nadawczej jest technologia Wi-Fi, natomiast w przypadku gdy wymagana jest duża mobilność, wykorzystuje się sieci telefonii komórkowej poprzez system GPRS, oraz jego pochodne: EDGE, UMTS lub HSDPA.

Moduły GPRS znalazły szerokie zastosowanie w mobilnej teledystrybucji i systemach powiadamiania alarmowego. Moduły te dają duże możliwości konfiguracji i scenariuszy użytkowania, przez co są bardzo uniwersalne, co jest niewątpliwą zaletą. Budowane są najczęściej przy użyciu popularnych mikrokontrolerów, opartych na rdzeniach takich jak ARMv7 czy ARMv9, które znacząco ułatwiają oprogramowanie takiego modułu, jak i wspomagają zadania obliczeniowe. Dzięki temu, komunikacja z modułem najczęściej odbywa się z wykorzystaniem magistral jakie oferuje mikrokontroler. Popularnymi sposobami komunikacji są w tym przypadku interfejsy RS-232, USB, JTAG oraz ostatnio nawet Ethernet. Dodatkowym atutem takiego rozwiązania jest możliwość wykorzystania przez moduł GPRS innych interfejsów jakie oferuje mikroprocesor, do zadań związanych z sygnalizacją czy komunikacją z innymi urządzeniami elektroniki przemysłowej. Mowa tutaj o interfejsach I2C, SPI czy portach ogólnego przeznaczenia GPIO.

## 2.2.4. Wybór modułu GPRS [12]

### Cinterion HC25

Moduł HC25 to modem HSDPA przeznaczony do zastosowania w urządzeniach takich jak routery, przenośne odtwarzacze multimediiów czy urządzenia PDA.



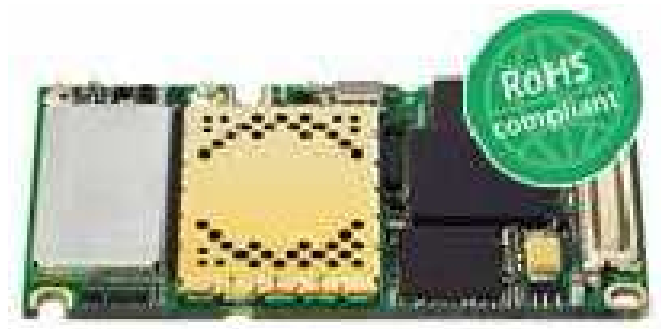
Rys.2.17. – Moduł Cinterion HC25

Parametry modułu HC25:

- obsługa kart SIM typu 3 V i 1,8 V,
- praca w systemie GSM 850/900/1800/1900 MHz,
- zgodność z systemami UMTS/HSPDA (WCDMA/FDD) 850/190/2100 MHz,
- klasa GPRS multislot 10,
- klasa EDGE multislot 10,
- stacja mobilna klasy B,
- transmisja danych CSD do 14,4 kbit/s,
- obsługa za pomocą komend AT,
- interfejs programowanie NDIS,
- obsługa protokołów TCP, UDP, HTTP, FTP, SMTP, POP3
- złącze komunikacyjne B2B firmy Molex, zawierające porty:
  - USB 2.0 full speed,
  - zasilanie,
  - interfejs audio,
  - zasilanie: od 3,2 – 4,2 V,
  - średni pobór prądu przy transmisji HSPDA < 970 mA.

### **Cinterion XT65**

Jest to połączenie czterzakresowego modułu GSM/GPRS oraz 16 kanałowego odbiornika GPS.



*Rys. 2.18. – Moduł Cinterion XT65*

Parametry modułu XT65:

- praca w systemie GSM 850/900/1800/1900 MHz,
- wewnętrzny procesor ARM® Core, Blackfin® DSP
- pamięć 400 kB RAM, około 1,2MB pamięci FLASH,
- klasa GPRS 12 multislots 12,
- obsługa protokołów TCP, UDP, HTTP, FTP, SMTP, POP3,
- złącze komunikacyjne B2B firmy Molex, zawierające porty:
- USB 2.0 full speed,
- dwa interfejsy szeregowy w standardzie ITU-T V.24,
- magistrale szeregowy I2C i SPI,
- dwa wejścia przetwornika A/D,
- jedno wyjście analogowe PWM,
- stacja mobilna klasy B,
- obsługa kart SIM typu 3 V i 1,8 V,
- obsługa za pomocą komend AT,
- transmisja danych CSD do 14,4 kbit/s,
- zasilanie: od 3,2 – 4,5 V,

Platforma Java:

- konfiguracja CLDC 1.1 HI,
- specyfikacja J2ME o profilu IMP – NG ,
- API JSR179 do obsługi GPS,
- obsługa usług kryptograficznych PKI, SSL oraz wspomaganie protokołu HTTPS,
- aktualizacja programu OTAP.

Odbiornik GPS:

- 16 kanałowy odbiornik GPS, częstotliwość pracy L1 1575,42 MHz,
- dokładność sferyczna (SEP): 5 m, dokładność koleisty (CEP): 2,5 m,
- ze wspomaganie DGPS dokładność sferyczna (SEP): 3 m, dokładność koleisty (CEP): 2,0 m,
- obsługa za pomocą komend AT lub interfejsu Java

### Siemens TC65

Siemens TC65 to programowalny w języku Java czterozakresowy modem GPRS.



Rys. 2.19. – Moduł TC65

Parametry modułu TC65:

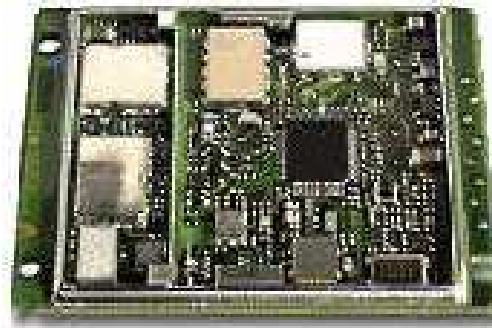
- praca w systemie GSM 850/900/1800/1900 MHz,
- procesor ARM© Core, Blackfin© DSP
- pamięć 400 kB RAM, około 1,7MB pamięci FLASH,
- klasa GPRS multislots 12,
- obsługa protokołów TCP, UDP, HTTP, FTP, SMTP, POP3,
- złącze komunikacyjne B2B firmy Molex, zawierające porty:
- USB 2.0 full speed,
- dwa interfejsy szeregowy w standardzie ITU-T V.24,
- magistrale szeregowy I2C i SPI,
- dwa wejścia przetwornika A/D,
- jedno wyjście analogowe PWM,
- stacja mobilna klasy B,
- obsługa kart SIM typu 3 V i 1,8 V,
- obsługa za pomocą komend AT,
- transmisja danych CSD do 14,4 kbit/s,
- zasilanie: od 3,2 – 4,5 V,
- pobór mocy w zależności od trybu:
- tryb power down – 100 mA,
- tryb idle < 3,5 mA,
- w czasie rozmowy średnio – 300 mA,
- w czasie transmisji GPRS, (klasa 12) średnio – 600 mA,
- kontrola ładowania ogniw litowo – jonowych.

Platforma Java:

- konfiguracja CLDC 1.1 HI,
- specyfikacja J2ME o profilu IMP – NG ,
- obsługa usług kryptograficznych PKI oraz wspomaganie protokołu HTTPS,
- aktualizacja programu możliwa poprzez OTAP.

### **Cinterion TC35i**

Moduł Cinterion TC35i to dwuzakresowy modem GSM, wykorzystywany najczęściej w małych aplikacjach, nie potrzebujących pakietowej transmisji danych.



*Rys. 2.20. – Moduł TC35i*

Parametry modułu TC35i:

- praca w systemie GSM 900/1800 MHz,
- stacja mobilna klasy B,
- SMS (PTP i PTM),
- obsługa kart SIM typu 3 V,
- obsługa za pomocą komend AT,
- transmisja danych CSD do 14,4 kbit/s,
- zasilanie: od 3,3 – 5,5 V,
- pobór mocy w zależności od trybu:
- tryb power down – 100 mA,
- tryb idle < 3,5 mA,
- w czasie rozmowy średnio – 300 mA (2,3 A w szczycie).

### **Wavecom Q24 plus**

Moduł Q24 plus to czterozakresowy modem GPRS ze zintegrowanym złączem na karty SIM.



*Rys. 2.21. – Moduł Q24 plus*

Parametry modułu serii Q24 plus:

- praca w systemie GSM 850/900/1800/1900 MHz,
- klasa GPRS multislot 10,
- procesor: ARM7, 32 bit, 52 MHz,
- dostępne interfejsy:
- dwa interfejsy UART,
- 11 wejść/wyjść GPIO,
- magistrale szeregowo I2C i SPI,
- obsługa klawiatury,
- wejście A/D,
- dwa wyjścia analogowe audio,
- stacja mobilna klasy B,
- obsługa kart SIM typu 3 V i 1,8 V,
- obsługa za pomocą komend AT,
- zasilanie: od 3,2 – 4,2 V.

### **Wavecom Q26 Extreme**

Q26 Extreme to modem HSDPA wyposażony w procesor ARMv9 programowany w języku C lub skryptami Lua.



*Rys. 2.22. – Moduł Q26 Extreme*

Parametry modułu Q26 Extreme:

- procesor ARM946/DSP, 32 bit, 26-104 MHz,
- standard HSPDA do 7,2 Mbit/s,
- praca w systemie GSM 850/900/1800/1900 MHz,
- praca w systemie UMTS/HSxDA (WCDMA/FDD) 850/190/2100 MHz,
- klasa EDGE multislot 12,
- klasa GPRS multislot 12,
- obsługa protokołów TCP, UDP, HTTP, FTP, SMTP, POP3
- dostępne interfejsy:
- do 45 wejść/wyjść GPIO (26 GPIO 2,8 V, 19 GPIO 1,8 V),
- dwie magistrale SPI,
- magistrala I2C,

- klawiatura 5 x 5 klawiszy,
- dwa asynchroniczne interfejsy szeregowo,
- USB 2.0 full speed,
- 16 bitowy port równoległy,
- interfejs audio,
- stacja mobilna klasy B,
- obsługa kart SIM typu 3 V i 1,8 V,
- obsługa za pomocą komend AT,
- transmisja danych CSD do 14,4 kbit/s,
- zasilanie: od 3,2 – 4,2 V,
- średni pobór prądu przy transmisji HSPDA < 970 mA.

### Motorola G24

Jest to moduł pracujący w standardach GPRS oraz EDGE programowany w języku Java.



Rys. 2.23. – Moduł G24

Parametry modułu G24 Java:

- praca w systemie GSM 850/900/1800/1900 MHz,
- klasa EDGE multislots 10,
- klasa GPRS multislots 10,
- obsługa protokołów TCP/IP, SMTP, POP3,
- dostępne interfejsy:
- dwa asynchroniczne interfejsy szeregowo,
- magistrala I2C
- USB 2.0 full speed,
- cyfrowe wejście audio,
- stacja mobilna klasy B,
- obsługa kart SIM typu 3 V,
- obsługa za pomocą komend AT,
- transmisja danych CSD do 14,4 kbit/s,
- zasilanie: od 3,3 – 4,2 V,

Platforma Java:

- konfiguracja CLDC 1.1 HI,
- specyfikacja MIDP 2.0,

- specyfikacja WMA 1.1/2.0 (SMS,MMS), systemu plików (JSR 75),
- specyfikacja API Motoroli, dla obsługi I2C, GPIO,
- aktualizacja programu możliwa poprzez OTAP.

### **Teltronika TM3**

Moduł TM3 to pracujący w czterech zakresach modem GPRS/EDGE.



*Rys. 2.24. – Moduł TM3*

Parametry modułu TM3:

- praca w systemie GSM 850/900/1800/1900 MHz,
- klasa EDGE multislot 12,
- klasa GPRS multislot 12,
- obsługa protokołów UDP/IP, TCP/IP, SMTP, POP3,
- dostępne interfejsy:
- 30 wejść/wyjść GPIO,
- dwa asynchroniczne interfejsy szeregowo,
- magistrala I2C,
- USB 2.0 full speed,
- dwa wejścia analogowe audio,
- stacja mobilna klasy B,
- obsługa kart SIM typu 3 V i 1,8V,
- obsługa za pomocą komend AT,
- transmisja danych CSD do 14,4 kbit/s,
- zasilanie: od 3,3 – 4,2 V,

### **2.2.5. Moduł Siemens TC65**



Modem Siemens TC65 to jeden z kilku przedstawicieli modułów programowanych w języku Java. Moduł posiada w swoim wewnętrznym oprogramowaniu zaimplementowaną wirtualną maszynę Javy (JVM – ang. Java Virtual Machine), dzięki czemu czas przeznaczony na zrozumienie działania modułu oraz na jego oprogramowanie ulega znacznemu skróceniu. W dodatku, zyskuje się możliwość programowania modułu korzystając ze wszystkich dobrodziejstw programowania zorientowanego obiektowo (OO – ang. Object Oriented programming), co przekłada się m.in. na czytelność oraz łatwość tworzenia kodu.

W tabeli przedstawiono parametry techniczne modemu TC65 [8].

Tab. 2.1 – Parametry modułu Siemens TC65

Parametr	Wartość
Zakres częstotliwości	GSM: 850/900/1800/1900 MHz
Moc wyjściowa	Klasa 4 (+33 dBm $\pm$ 2 dB) dla EGSM 850 Klasa 4 (+33 dBm $\pm$ 2 dB) dla EGSM 900 Klasa 1 (+30 dBm $\pm$ 2 dB) dla GSM1800 Klasa 1 (+30 dBm $\pm$ 2 dB) dla GSM1900
Zasilanie	od 3,2 V do 4,5 V
Zakres temperatur	Zakres normalny: od -30°C do +65°C Zakres ograniczony: od +65°C do 75°C
Transmisja danych	GPRS: • klasa multislots: 12 • klasa stacji mobilnej: B • schemat kodowania: CS1 CS4 CSD: • prędkość transmisji: 2,4 – 14,4 kbit/s • V.110, RLP, transmisja nieprzezroczysta
SMS	• tryb tekstowy i PDU • przechowywanie: karta SIM plus miejsce na 25 SMSy w pamięci modułu • transmisja SMS poprzez CSD lub GPRS
Audio	Tryb kodowanie głosu: • tryb HR (ang. <i>Half Rate</i> ) • tryb FR (ang. <i>Full Rate</i> ) • ulepszony tryb FR (ang. <i>Enhanced Full Rate</i> ) • tryb adaptacyjny AMR (ang. <i>Adaptive Multi-Rate</i> )
Komendy AT	AT-Hayes GSM 07.05 oraz 07.07, komendy AT firmy Siemens kompatybilne ze sterownikami RIL (ang. <i>Radio Interface Layer</i> )
Kompatybilność Microsoft	obsługa interfejsów RIL/NDIS (ang. <i>Network Driver Interface Specification</i> ) dla urządzeń Pocket PC oraz Smartphone

Platforma Javy	Wirtualna maszyna Javy wraz ze specyfikacjami API, parserem AT, interfejsem szeregowym, systemem plików Flash oraz stosem TCP/IP. Pamięć Flash: około 1,7 MB Pamięć RAM: około 400 kB Stos TCP/IP Dostęp poprzez komendy AT
Adresowanie IP	IP w wersji 6
Zdalny dostęp karty SIM	Moduł TC65 daje możliwość zewnętrznego dostępu do karty SIM poprzez interfejs szeregowy oprócz karty SIM podłączonej do dedykowanych linii modułu.
2 interfejsy szeregowy	<p>ASC0</p> <ul style="list-style-type: none"> <li>• 8 przewodowy interfejs modemowy z liniami kontrolnymi, niesymetryczny, asynchroniczny</li> <li>• prędkość transmisji: od 300 do 460800 bit/s w trybie stałym oraz od 1200 do 460800 bit/s w trybie automatycznym</li> <li>• linie RTS0/CTS0 oraz XON/XOFF kontrolujące przepływ</li> <li>• możliwość multipleksowania transmisji</li> </ul> <p>ASC1</p> <ul style="list-style-type: none"> <li>• 4 przewodowy interfejs, niesymetryczny asynchroniczny</li> <li>• prędkość transmisji: od 300 do 460800 bit/s w trybie stałym</li> <li>• linie RTS1/CTS1 oraz programowe XON/XOFF kontrolujące przepływ</li> </ul>
USB	USB w wersji 2.0 Full Speed
I2C	7 bitowe adresowanie i transmisja, prędkość transmisji do 400 kbit/s, alternatywnie linie magistrali I2C mogą zostać skonfigurowane jako linie magistrali SPI
SPI	Magistrala szeregowy do transmisji danych do 6,5 Mbit/s. W przypadku gdy interfejs SPI jest aktywny, nieaktywną jest magistrala I2C
Audio	<ul style="list-style-type: none"> <li>• 2 wejścia analogowe</li> <li>• 1 wejście cyfrowe PCM</li> </ul>
Interfejs kart SIM	3 V oraz 1,8V
Antena	50 $\Omega$ , podłączana do modułu poprzez gniazdo antenowe lub wlutowywana
Interfejs modułu	80 pinowe złącze typu B2B (ang. <i>board-to-board</i> )
GPIO	10 wejść/wyjść ogólnego przeznaczenia programowanych za pomocą komend AT. Linie te mogą zostać skonfigurowane alternatywnie jako licznik impulsów.

Ładowanie	Moduł zawiera linie nadzorujące proces ładowania akumulatorów litowo-jonowych i litowo-polimerowych
Licznik impulsów	Impulsy mierzone są w zakresie od 0 do 1000 na sekundę. Jeżeli licznik impulsów jest aktywny, linie GPIO nie są dostępne.
Przetwornik ADC	Dwa wejścia symetryczne przetwornika analogowo-cyfrowego
Przetwornik DAC	Wyjście cyfrowe przetwornika, które dostarcza sygnał PWM

### 2.2.6. Budowa modułu TC65.

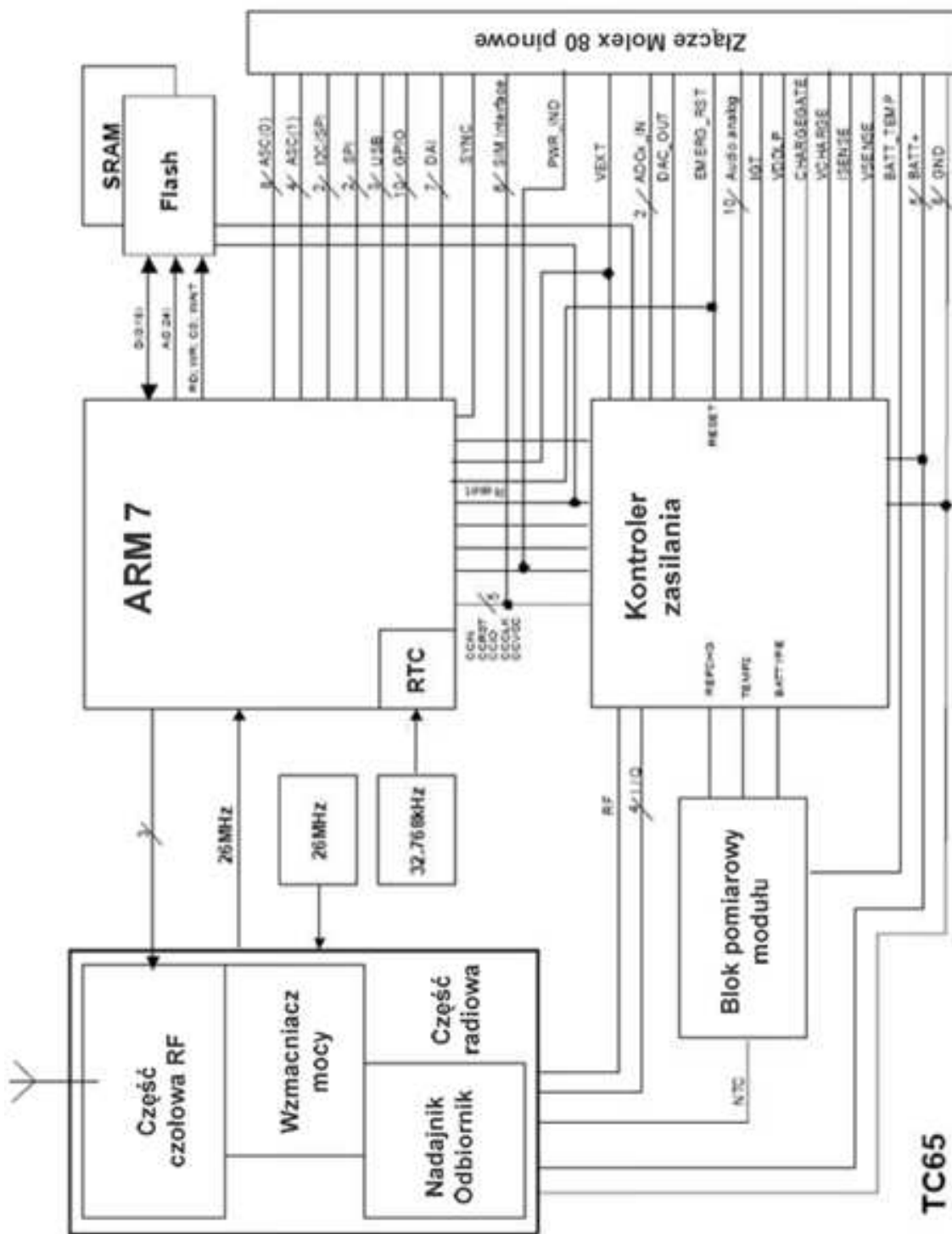
Budowę modułu TC65 można podzielić na trzy funkcjonalne grupy.

Pierwsza z nich obejmuje główny mikrokontroler ARM, wraz z jego peryferiami takimi jak pamięć RAM i Flash. Komunikacja z pamięcią odbywa się za pomocą linii adresowych A0 do A24, oraz linii danych D0 do D15. W strukturze mikrokontrolera znajduje się także zegar czasu rzeczywistego RTC, taktowany częstotliwością 32.768 kHz. Interfejsy komunikacyjne mikrokontrolera wyprowadzone są na złącze B2B.

Druga grupa zawiera kontroler zasilania i blok pomiarowy układu. Pierwszy element odpowiedzialny jest za zasilanie oraz jego obsługę, nadzoruje włączanie, wyłączanie i resetowanie modułu oraz proces ładowania ogniw. Jego zadaniem jest także obsługa przetworników analogowo-cyfrowych i cyfrowo-analogowych, zawartych w jego strukturze. Układ pomiarowy modułu dostarcza informacji o temperaturze części radiowej oraz ładowanego ogniwa.

Trzecią grupą jest część radiowa, która swoją strukturą obejmuje blok nadajnika i odbiornika, wzmacniacz mocy oraz blok czołowy RF. Blok nadajnika i odbiornika zajmuje się formowaniem oraz przetwarzaniem sygnałów częstotliwości radiowej odbieranych i nadawanych przez moduł. Kolejny blok wzmacnia sygnał do wartości potrzebnej na poprawne wyemitowanie lub odebranie sygnału radiowego. Ostatni blok odpowiedzialny jest za emisję i odbiór sygnałów za pomocą anteny nadawczo-odbiorczej.

Na rysunku 2.15 przedstawiono budowę modułu Siemens TC65.



Rys.. 2.25. – Budowa wewnętrzna modułu Siemens TC65

## 2.3 Oprogramowanie

### 2.3.1. Język programowania Java

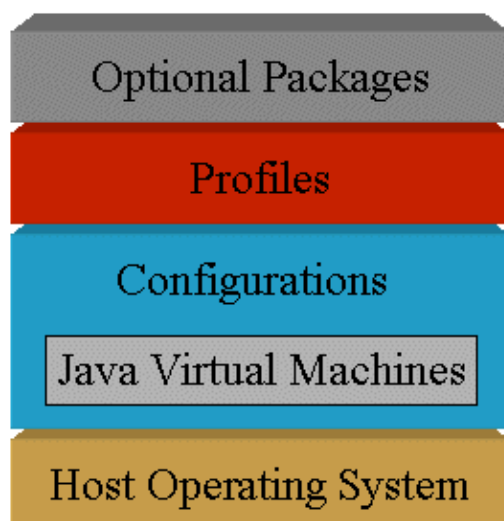
Java jest obiektywnym językiem programowania wypuszczonym na rynek przez firmę Sun Microsystems pod koniec 1995 roku [3]. Programy w nim napisane kompilowane są do tzw. kodu bajtowego, który wykonywany jest przez maszynę wirtualną. Składnia języka oraz słowa kluczowe zostały przejęte z języka C++. Główne koncepcje, takie jak zarządzanie pamięcią czy maszyna wirtualna, zostały zapożyczone z języka Smalltalk.

W 1996 roku firma Sun Microsystems przedstawiła wersję 1.0 języka Java. W krótkim czasie język ten został zaimplementowany w przeglądarkach Netscape, co przyczyniło się do zwiększenia popularności języka. W 1998 roku nastąpił przełom w rozwoju Javy, kiedy to pojawiła się wersja 2.0, nazywana wówczas J2SE 1.2 (ang. Java 1.2 Standard Edition). Wersja ta wprowadziła znaczące zmiany w porównaniu do poprzedniczki. Zaimplementowano m.in. metodę wykonywania programów poprzez kompilację do kodu maszynowego w locie – JIT (ang. Just In Time Compilation) oraz bibliotekę graficzną Swing. W wersji tej Java została podzielona na trzy platformy, różniące się od siebie przeznaczeniem: J2SE – dla aplikacji desktopowych, J2EE (Java Enterprise Edition) – dla aplikacji serwerowych, oraz J2ME (Java Micro Edition) – dla aplikacji mobilnych. W pracy opisano dwie z nich: JavaME oraz JavaEE.

### 2.3.2. Java Micro Edition (JavaME) [2]

Java Micro Edition jest technologią opracowaną przez firmę Sun Microsystems. Znana jest również pod nazwą Java 2 Platform, Micro Edition lub J2ME. Definiuje uproszczoną wersję Javy. Platforma ta została skonstruowana z myślą o uruchamianiu i tworzeniu aplikacji mobilnych na urządzenia o ograniczonych zasobach sprzętowych takich jak telefony komórkowe, palmtopy czy karty inteligentne. Ze względu na ograniczenia techniczne takich urządzeń, takie jak wolniejsze procesory, mniejsza ilość pamięci czy ekrany o niskiej rozdzielczości, platforma JavaME ma własny zestaw klas, znacznie ograniczony w stosunku do standardowej platformy Javy. Zestaw ten nazywa się konfiguracją (ang. *Configuration*). Dodatkową funkcjonalność zapewniają tzw. profile (ang. *Profiles*), zawierające klasy nieprzewidziane w konfiguracji.

Podstawową architekturę JavyME przedstawiono na rysunku 2.16.



Rys 2.26. – Ogólna architektura JavyME

Cechą charakterystyczną Javy jest wykorzystywanie przez nią maszyny wirtualnej. Wszystkie platformy tej technologii, łącznie z JavąME, wymagają maszyny wirtualnej do uruchamiania aplikacji. W przypadku JavyME, w odróżnieniu do reszty platform, maszynę tę tworzy producent urządzenia, na którym ma działać maszyna. Platforma JavaME jest więc niezależna od zastosowanego w urządzeniu systemu operacyjnego (jak SymbianOS czy Android), procesora czy architektury. Większość tradycyjnych telefonów komórkowych jest wyposażonych w autorskie systemy operacyjne producentów, co wręcz uniemożliwiłoby proces tworzenia i implementacji maszyny wirtualnej dla JavyME, przez jedną firmę. Dlatego to zadanie zostało powierzone producentom telefonów.

Wyróżnić można dwie podstawowe konfiguracje utworzone dla JavyME: CDC (Connected Device Configuration) oraz CLDC (Connected Limited Device Configuration). CDC jest stworzone dla zaawansowanych urządzeń, posiadających wydajne procesory i duże ilości pamięci, jak np.: nawigacje samochodowe czy odtwarzacze BluRay. CLDC przewidziane jest dla urządzeń prostszych i mniej wydajnych, takich jak np.: telefony komórkowe czy smartfony. Zdecydowanie większa część używanych obecnie konfiguracji JavyME to CLDC.

Rozszerzeniem możliwości konfiguracji są profile. Profil jest definicją zestawów interfejsów programistycznych (API) ułatwiających tworzenie i działanie aplikacji. Podstawowym profilem konfiguracji CLDC jest profil MIDP (Mobile Information Device Profile), dostarczający klas odpowiedzialnych za interfejs użytkownika (warstwa wizualna - GUI) oraz warstwę komunikacji sieciowej. Innym profilem konfiguracji jest profil IMP-NG, pozbawiony klas odpowiedzialnych za obsługę warstwy graficznej.

Najczęściej spotykaną kombinacją jest więc CLDC oraz MIDP. Istnieją dwie wersje konfiguracji CLDC oraz cztery wersje profilu MIDP. Konfiguracja CLDC 1.0 jest starsza i uboższa od CLDC 1.1. Z tego względu nie jest już praktycznie w ogóle wykorzystywana. Profil MIDP występuje w wersjach 1.0, 2.0, 2.1 oraz najnowszej – 3.0. Wersja 2.0 dodaje m.in. obsługę kolorowych wyświetlaczy oraz interfejsu audio. Jest najczęściej obecnie stosowaną wersją MIDP. W najnowszym wydaniu Javy (JavaMe 8) profile MIDP zostały zastąpione przez profil MEEP (JavaME Embedded Profile).

Dopełnieniem całości są pakiety opcjonalne (*ang. Optional Packages*), dające możliwość programowania dodatkowych funkcji, nie zawartych w podstawowych profilach. Wybór pakietów obsługiwanych w poszczególnych urządzeniach zależy najczęściej od producenta i modelu urządzenia. Pakiety dodatkowe oznaczane są akronimem JSR (*ang. Java Specification Request*) oraz numerem.

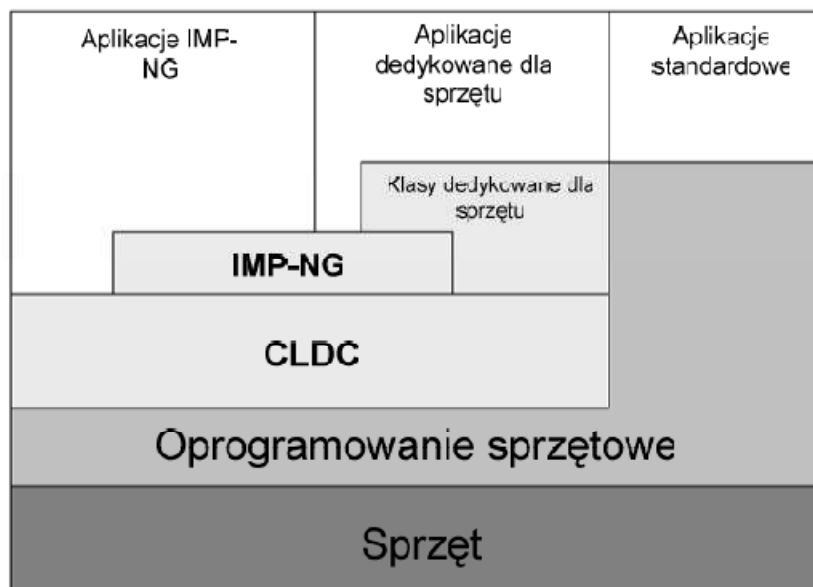
Dość popularnym pakietem implementowanym np.: w telefonach komórkowych, opartych o konfigurację CLDC i profil MIDP 2.0, jest pakiet JSR-82, dający możliwość oprogramowania łączności bezprzewodowej Bluetooth z poziomu aplikacji JavyME. Należy zauważyć, że sama obecność technologii Bluetooth w urządzeniu nie daje jeszcze możliwości jej programowania przez JavęME. Zapewnia to dopiero obsługa pakietu JSR-82.

Do tworzenia aplikacji na platformę JavaME niezbędne są odpowiednie narzędzia programistyczne. Narzędzia te zapewnia pakiet o nazwie WTK (*ang. Java Wireless Toolkit*), lub pochodne tego pakietu. Dostarcza on niezbędnych bibliotek i programów potrzebnych do skompilowania pisanego programu oraz narzędzi dodatkowych takich jak emulatorzy platformy czy definicje funkcji zawartych w API. Pakiet WTK jest bazowym pakietem dla producentów urządzeń programowanych w JavieME, którzy często wydają własne pakiety, rozszerzone o pewne specyficzne dla danego producenta rozwiązania. Najczęściej nie neguje to jednak kompatybilności między pakietami. Aplikacja napisana za pomocą pakietu jednego producenta, będzie z powodzeniem działać na urządzeniu innego, o ile obsługuje ono odpowiednią konfigurację i profil oraz ewentualnie wykorzystywane pakiety dodatkowe. W niniejszej pracy wykorzystywany jest pakiet SMTK „*TC65 Rel.3 Software Development Kit*”. Pakiet ten dostarcza m.in. emulatora, pozwalającego na uruchamianie i testowanie napisanych programów na komputerze klasy PC w połączeniu z rzeczywistym modulem Siemens TC65, na którym odbywa się proces debugowania programu. Środowiskiem programistycznym jest NetBeans IDE, który doskonale integruje się ze wspomnianym wyżej pakietem SDK.

### 2.3.3. Profil IMP-NG

Profil IMP-NG utworzony został na potrzeby zastosowania języka Java do obsługi urządzeń elektronicznych w zastosowaniach przemysłowych. Jest bardzo podobny do profilu MIDP pod względem zawartych klas i interfejsów, jednak nie implementuje w swojej strukturze klas do obsługi wyświetlaczy graficznych urządzeń. Profil IMP-NG działa w warstwie CLDC, która reprezentuje wirtualną maszynę Javy, oraz dołącza wymagane do działania biblioteki. Jest to podstawa do oprogramowania urządzenia, w oparciu o wspierane pakiety API. Wyróżnić można dwa rodzaje pakietów API. Pierwszy z nich jest pakietem standardowych klas dostarczonych wraz z pakietem IMP-NG. Drugi zestaw klas, to zbiór dedykowany do obsługi specyficznych zadań i procesów danego urządzenia, takich jak obsługa interfejsu Bluetooth czy przetworników analogowo-cyfrowych.

Na rysunku 2.17 przedstawiono hierarchię warstw korzystających z profilu informacyjnego IMP-NG.



Rys. 2.27. – warstwy profilu IMP-NG

Minimalne wymagania sprzętowe do obsługi profilu IMP-NG:

- 128 kB nie ulotnej pamięci przeznaczonej na komponenty profilu IMP,
- 8 kB nie ulotnej pamięci przeznaczonej na zarządzanie trwałymi danymi aplikacji,
- 128 kB nie ulotnej pamięci wykorzystywanej podczas pracy wirtualnej maszyny Java, np. wykorzystanie obsługi stosu,
- praca w sieci – dwukierunkowa, ciągła lub nieciągła,
- opcjonalnie możliwość obsługi dźwięków.

Aplikacje pisane w oparciu o konfigurację CLDC i profil informacyjny IMP-NG nazywane są IMletami. Nazwa ta pochodzi o MIDletu - nazwy aplikacji zbudowanej w oparciu o profil MIDP. IMlet jest bardzo podobny do MIDletu, dziedziczy bowiem z tej samej klasy.

Szablon aplikacji IMletowej musi zawierać w swojej strukturze metody odpowiedzialne za kontrolę i sterowanie cyklem działania aplikacji. Szkielet taki można przedstawić w następującej postaci:

```
public class IMletMain extends MIDlet {
    public void startApp() {
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```

Utworzona klasa IMletMain dziedziczy po klasie MIDlet, przez co ma dostęp do interfejsów i funkcji zdefiniowanych w klasach MIDletów.



Metoda `startApp()` jest uruchamiana automatycznie podczas startu działania `IMletu`.

Dwie kolejne metody związane są ze stanami, w jakich może znaleźć się `IMlet`. Pierwszym ze stanów jest stan wstrzymania. Stan taki może zostać wywołany przez sytuacje zewnętrzne, jak np. rozmowa głosowa, lub poprzez świadome działanie programisty, wywołując metodę `notifyPaused()`. Stan aktywny można przywrócić wywołując metodę `resumeRequest()`, która informuje maszynę wirtualną Javy o chęci powrotu do stanu aktywnego.

Kolejnym stanem jest stan zakończenia. Można do niego doprowadzić wywołując metodę `destroyApp()`. Metoda ta powinna zawierać instrukcje zapisujące wszelkie niezapisane dane oraz kończące działanie `IMletu`.

### 2.3.4. Java Enterprise Edition (JavaEE) [5]

Java Platform, Enterprise Edition (dawniej Java Platform 2, Enterprise Edition lub J2EE) jest standardem tworzenia usługowych aplikacji biznesowych, utworzonym na podstawie platformy Java 2 Standard Edition. Jest to zbiór różnorodnych technologii umożliwiających tworzenie wyżej wymienionych aplikacji. Technologie te tworzone są także przez firmy trzecie. Aplikacje JavyEE oparte są o wielowarstwową architekturę komponentową. Komponenty są zwykle osadzone na serwerze aplikacji obsługującym poszczególne części JavyEE.

Mimo iż JavaEE jest czasem określana mianem korporacyjnej, nadaje się nie tylko do aplikacji stricte biznesowych. Dzięki mnogości różnorodnych technologii oraz łatwości z jaką pisze się programy, idealnie nadaje się do tworzenia aplikacji webowych wszędzie tam, gdzie standardowe strony internetowe nie wystarczają. W związku z tym wytworzyło się określenie platformy jako „Java Web”, w przeciwieństwie do JavaEE. Przyjęło się, że jeżeli aplikacja korzysta z technologii EJB (ang. Enterprise JavaBeans) to mówimy o aplikacji biznesowej (JavaEE), w przeciwnym razie o aplikacji webowej (definicja za [5]).

Najważniejsze z technologii JavyEE to m.in. serwlety (nazwa pochodzi od apletów, aplikacji użytkowych). Są podstawą wszystkich aplikacji webowych. Według najprostszej definicji jest to klasa, która może przyjmować żądania i generować odpowiedzi. Java jako technologia bardzo uniwersalna teoretycznie udostępnia mechanizm serwletów działających dla dowolnych protokołów sieciowych. W praktyce nie spotyka się jednak innych niż te, które wykorzystują protokół HTTP. Kolejną ważną technologią jest technologia JSP (Java Server Pages). Jest to połączenie kodu opisu stron internetowych HTML (HyperText Markup Language) z językiem Java. Konstrukcja JSP przypomina konstrukcję innych technologii internetowych, takich jak PHP czy ASP.NET. Pliki JSP składają się ze zwykłych znaczników HTML oraz z konstrukcji charakterystycznych dla Javy. Dzięki wykorzystaniu Javy, zawartość stron internetowych jest dynamicznie przekształcana po stronie serwera, w wyniku czego użytkownik dostaje jedynie kod HTML zrozumiały dla przeglądarek. Technologią wykorzystywaną równie często co JSP jest JPA (Java Persistence API). Jest to technologia dostępu do źródeł danych, bez względu na faktycznie stosowaną technologię bazodanową. JPA samo w sobie jest jedynie opisem który określa sposób komunikacji ze źródłami danych. W aplikacjach korzysta się z

istniejących implementacji tego opisu zwanych frameworkami. Najczęściej stosowanym frameworkiem JPA jest Hibernate. Kolejna technologia wykorzystywana w ramach JavyEE to JSF (Java Server Faces). Jest ona odpowiedzialna za kwestie prezentacji danych, pozwala rozdzielić czynności związane z konstruowaniem widoku aplikacji od mechanizmów tzw. logiki aplikacji. JSF kontroluje to, co dzieje się w warstwie prezentacji i łączy ją z pozostałymi komponentami aplikacji, jednak za wyświetlanie i prezentację odpowiedzialna jest inna technologia. W poprzedniej wersji JSF były to strony JSP, w aktualnej kontroluje to framework Facelets.

Technologia JDBC (Java DataBase Connectivity) oraz baza danych PostgreSQL to technologie nie należące do zbioru JavaEE, lecz warte wspomnienia w tym punkcie. Technologia JDBC jest dostępna nie tylko dla platformy JavaEE, można z niej korzystać także na platformie JavaSE (Java Standard Edition). Jest to interfejs programistyczny, umożliwiający łatwy i niezależny od platformy dostęp do baz danych za pomocą języka SQL. Platformy JavaSE oraz JavaEE zawierają API JDBC, jednak nie jest to wystarczające aby móc połączyć się z bazą danych. Konieczne jest dołączenie do aplikacji specjalnego sterownika, odpowiadającego za faktyczną komunikację interfejsu JDBC z bazą danych. Sterownik ten dostarcza najczęściej producent bazy danych [9]. Zaletą takiego rozwiązania jest przenośność kodu aplikacji. W przypadku zmiany bazy danych wystarczy dołączyć do aplikacji inny sterownik JDBC oraz dokonać niewielkich korekt w kodzie aby, całość działała jak wcześniej.

PostgreSQL jest to system zarządzania relacyjnymi bazami danych, działający na zasadach wolnego oprogramowania. Jest jednym z najpopularniejszych systemów bazodanowych, zaraz obok MySQL.

Niniejszy projekt wykorzystuje technologie serwletów oraz JSP ze zbioru JavaEE w celu obsługi żądań napływających z urządzenia mobilnego oraz generowania odpowiedzi w postaci wizualizacji przebiegu trasy śledzonego pojazdu. Rolę magazynu gromadzącego zapis przebytych tras pełni baza danych w systemie PostgreSQL, do której dostęp realizowany jest za pomocą technologii JDBC.

## Rozdział 3

### 3. Realizacja układu zdalnej lokalizacji obiektów ruchomych

Część mobilna układu zdalnej lokalizacji obiektów ruchomych składa się z modułu TC65 firmy Siemens, osadzonego na prototypowej płytce drukowanej, będącej przedmiotem pracy dyplomowej studenta Łukasza Kasperczyka [13] w roku 2008. Posłużono się gotową płytką, zawierającą układ zasilania, układy wejścia-wyjścia oraz slot karty SIM i antenę, w celu minimalizacji kosztów układu. Moduł połączony jest z odbiornikiem GPS RGPSM002 firmy Racewood znajdującym się na osobnej płytce PCB, wraz z dedykowanym układem zasilania. Układ zasilania modułu GPS składa się ze stabilizatora liniowego typu Low-Dropout LE33, dostarczającego napięcia 3,3 V, przy zasilaniu własnym powyżej 3.5 V. W związku z tym, do zasilania odbiornika GPS, można wykorzystać napięcie zasilania modułu TC65, wynoszące 4,1 V.

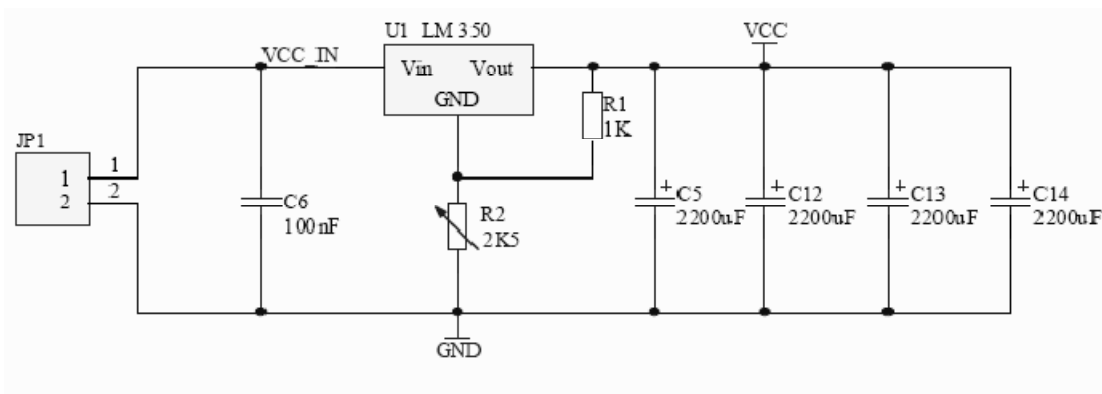
Aplikacja zawarta w oprogramowaniu modułu TC65 na bieżąco odbiera dane przesyłane przez odbiornik GPS do modułu oraz analizuje zawarte w nich informacje. Jeśli odbiornik GPS prześle do modułu poprawne dane lokalizacyjne, moduł zestawia połączenie GPRS z siecią Internet, oraz przesyła za jej pośrednictwem swoją lokalizację do zdalnego systemu nadzoru. Operacja ta powtarzana jest w pętli, aż do zakończenia działania modułu.

#### 3.1 Układ elektroniczny z modułem.

Jako interfejs sprzętowy modułu TC65 została wykorzystana płytka PCB będąca częścią innej pracy dyplomowej [13]. Interfejs ten zawiera w sobie następujące układy:

- układ zasilania modułu,
- slot karty SIM,
- antenę nadawczo-odbiorczą,
- układy komunikacyjne RS-232 i USB.

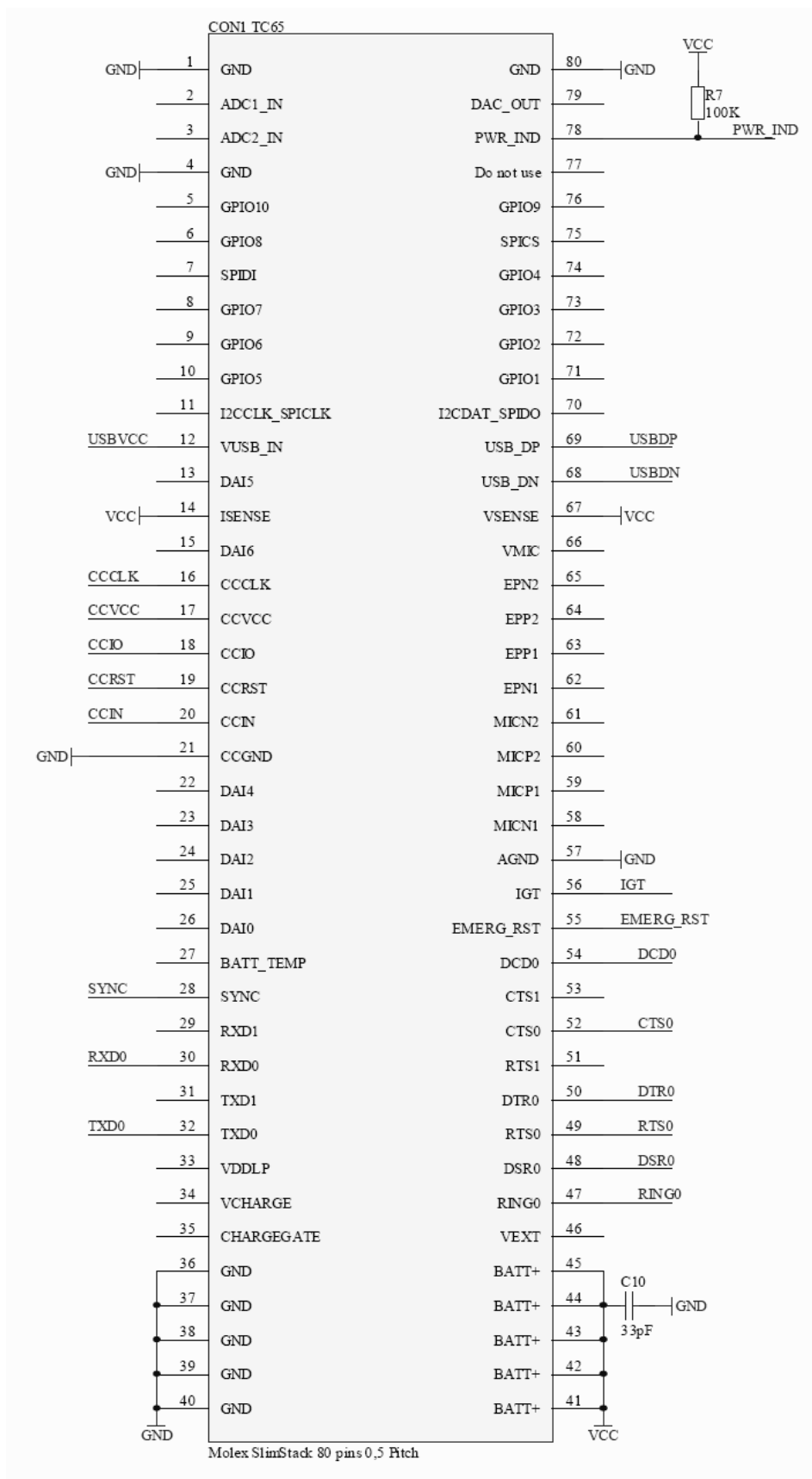
Na rysunku 3.1 przedstawiono układ zasilania modułu TC65. Układ działa w oparciu o liniowy stabilizator napięcia LM350 o baterię kondensatorów w roli filtra zmniejszającego tętnienia napięcia.



Rys. 3.1 – Układ zasilania modułu TC65

Stabilizator LM350 wraz z rezystorami konfiguracyjnymi dostarcza napięcia zasilania o wartości 4,1 V, które dodatkowo może być regulowane za pomocą potencjometru R2 w zakresie od 3,2 V do 4,5 V.

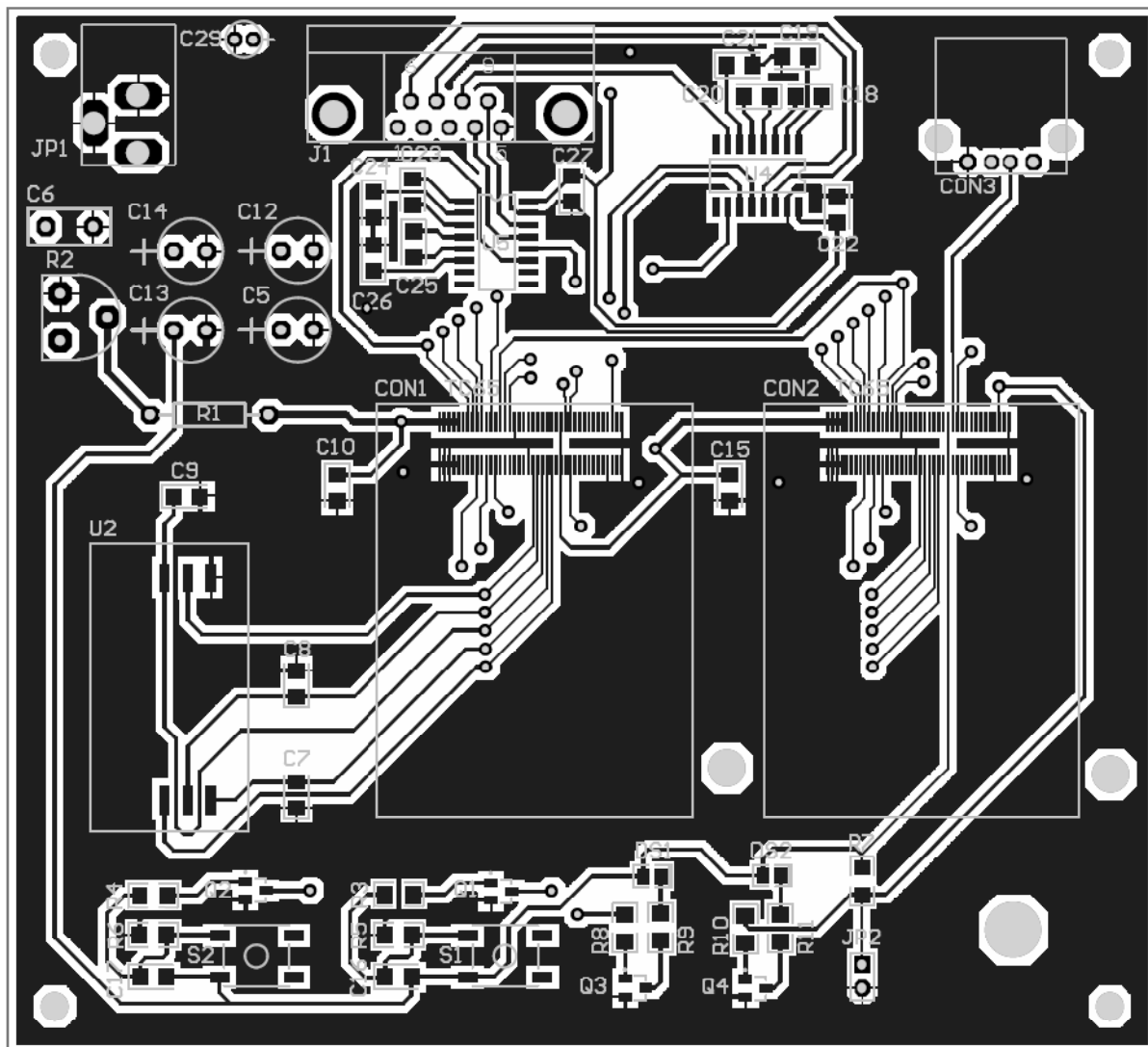
Rysunek 3.2 przedstawia złącze B2B firmy Molex, łączące interfejs użytkownika z modułem TC65. Złącze dostarcza niezbędnych napięć, sygnałów sterujących oraz sygnałów interfejsów komunikacyjnych.



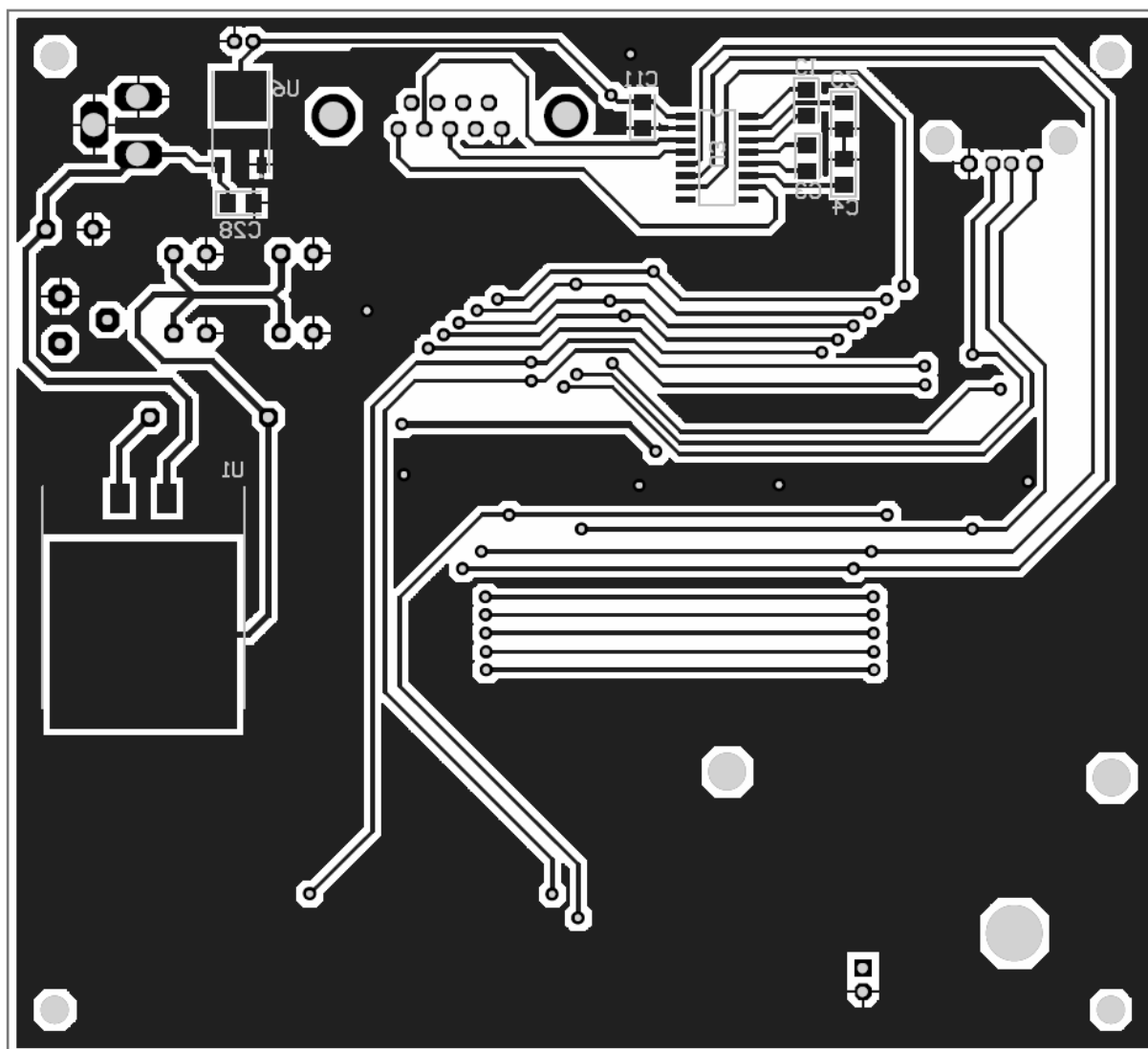
Rys.3.2 – Schemat złącza B2B modułu TC65

Projekt obwodu drukowanego utworzony został w programie Protel DXP. Wykonano płytkę o wymiarach 123 mm na 113 mm.

Rysunki 3.3 oraz 3.4 przedstawiają projekt interfejsu od strony górnej oraz dolnej płytki.



Rys.3.3 – Górna część płytki PCB interfejsu



*Rys.3.4 – Dolna część płytki PCB interfejsu*

### 3.2. *Oprogramowanie układu*

Aplikacja działająca na module TC65 została napisana w języku Java, dla konfiguracji CLDC oraz profilu IMP-NG za pomocą zintegrowanego środowiska programistycznego NetBeans IDE 8.1. W projekcie wykorzystano dostarczony przez producenta modułu TC65, firmę Siemens, pakiet programistyczny SDK o nazwie „*TC65 Rel.3 Software Development Kit*”.

Kod aplikacji podzielony został na funkcjonalne grupy, zwane klasami, zgodnie z zasadami programowania zorientowanego obiektowo. Każda z klas odpowiada zwykle za jedno, bądź kilka zadań z podobnej grupy funkcjonalnej. Klasami podstawowymi występującymi w aplikacji są:

- `IMletMain.java`,
- `ModuleInitializer.java`,
- `TC65GPRSSender.java`,
- `TC65GPS.java`,
- `TC65Location.java`,
- `TC65Preferences.java`,
- `TC65Serial.java`.

Klasy dodatkowe, pomocnicze, to klasy:

- `Logger.java`
- `StrSplit.java`
- `TC65Class.java`
- `TC65Runnable.java`

Aplikacja zawarta w pamięci modułu TC65 po starcie, wywołuje funkcję `startApp()` klasy `IMletMain`, w której definiowane są pozostałe moduły, tworzone obiekty zdefiniowanych wcześniej klas, oraz przeprowadzana jest wstępna inicjalizacja modułu. Następnie uruchamiane są dwa wątki programu, zawarte w klasach `TC65Serial` i `TC65GPRSSender`. Pierwszy z nich odpowiedzialny jest za ciągłe odczytywanie danych napływających z odbiornika GPS poprzez interfejs szeregowy. Dane te są odpowiednio przetwarzane, a wynik działania w postaci aktualnej lokalizacji wpisany zostaje do odpowiedniego obiektu aplikacji, zawartego w klasie `TC65Location`. Drugi z wątków, `TC65GPRSSender`, odpowiada za okresowe przysyłanie aktualnej lokalizacji, odczytanej z obiektu `TC65Location` do zdalnego systemu nadzoru poprzez sieć GPRS/GSM.

#### 2.3.5. Najważniejsze klasy programu

##### **IMletMain**

Klasa `IMletMain` to główna klasa programu lokalizacyjnego. Zawiera metodę startową aplikacji `startApp()`, oraz szereg wywołań powodujących wczytanie do pamięci programu pozostałych klas, utworzenie obiektów tych klas, wczytanie do pamięci ustawień modułu, oraz ogólną inicjalizację modułu do działania.

Listing przedstawia kod klasy `IMletMain` odpowiedzialny za wczytanie do pamięci i utworzenie odpowiednich obiektów wczytanych klas.



```
public class IMletMain extends MIDlet {

    public TC65Preferences tc65Preferences;
    public ModuleInitializer moduleInitializer;
    public TC65Location tc65Location;
    public TC65GPS tc65GPS;
    public TC65Serial tc65Serial;
    public TC65GPRSSender tc65GPRSSender;

    public void startApp() {

        tc65Preferences = new TC65Preferences();

        try {
            moduleInitializer = new ModuleInitializer(this);
            moduleInitializer.InitializeModule();
        } catch (ATCommandFailedException ex) {
            ex.printStackTrace();
        }

        if (moduleInitializer.isInitialized()) {
            tc65Location = new TC65Location();
            tc65GPS = new TC65GPS(this);
            tc65Serial = new TC65Serial(this);
            tc65GPRSSender = new TC65GPRSSender(this);
            tc65SMSSupport = new TC65SMSSupport(this);

            // start threads
            tc65Serial.start();
            tc65GPRSSender.start();
        }
    }
}
```

### **TC65Preferences**

Klasa TC65Preferences powstała w celu przetrzymywania aktualnych ustawień modułu oraz odczytu i zapisu tych ustawień w pamięci układu. Składa się z publicznych pól różnego typu, przechowujących poszczególne ustawienia konfiguracyjne. Konfiguracji podlega m.in. dostęp do sieci GPRS, zależny od operatora sieci komórkowej, który dostarcza dane takie jak APN (*ang. Access Point Name*), nazwa użytkownika i hasło, potrzebne do zalogowania się do sieci GPRS.

Listing przedstawia sposób definicji pól publicznych, służących do przechowywania ustawień konfiguracyjnych.

```
public class TC65Preferences {
    public String serverUrl;
    public String gprsApn;
    public String gprsUser;
    public String gprsPass;
    public String serialPort;
    public int serialBaudrate;
}
```

### ModuleInitializer

Klasa ta została utworzona jako wydzielona część kodu aplikacji, odpowiedzialna za inicjalizację i wstępną konfigurację modułu. Zawiera w sobie funkcje konfigurujące m.in. połączenie z siecią GPRS.

Listing przedstawia funkcję odczytującą ustawienia z moduły ustawień i konfigurującą połączenie GPRS na podstawie uzyskanych ustawień.

```
private void InitializeGPRS() throws ATCommandFailedException {
    String apn = iMletMain.tc65Preferences.gprsApn;
    String user = iMletMain.tc65Preferences.gprsUser;
    String pass = iMletMain.tc65Preferences.gprsPass;

    String resp = this.atc.send("AT^SJNET=\"gprs\", \"\" + apn +
    "\",\"" + user + "\",\"" + pass + "\",\"\",0\r");
    // GPRS conf
    System.out.println("GPRS CONF: " + resp);
    resp = this.atc.send("AT+CGATT=1\r");
    // Attach GPRS
    System.out.println("GPRS Attach: " + resp);
}
```

### TC65Serial

Niniejsza klasa odpowiada za obsługę odczytywania danych z portu szeregowego modułu, do którego podłączony jest odbiornik GPS. Klasa ta, będąca w rzeczywistości wątkiem, nieustannie odczytuje dane napływające z odbiornika GPS znak po znaku, po czym składa je w tzw. zdania (*ang. sentences*) protokołu NMEA, oraz przekazuje obiektowi TC65GPS w celu weryfikacji oraz odczytania aktualnej lokalizacji.

Listing przedstawia główną funkcję wątku, odpowiedzialną za odczytywanie znaków poprzez interfejs szeregowy oraz składanie ich w zdania protokołu NMEA.

```
public void mainLoop() {
    try {

        if (is.available() > 0) {
            input = is.read();
            //read data from COM port

            if (input != 13 && input != 10 && input != -1) {
                //catch CR, LF or end of stream
                sentence = sentence + (char) input;
                //collect all letters (bytes) until CR, LF
                or end of stream is reached
            } else if (sentence != "") {
                sentence2 =
                iMletMain.tc65GPS.validateSentence(sentence);
                if (sentence2 != null) {
                    // checks if sentence is valid

                    iMletMain.tc65GPS.parseNMEA(sentence2);
                }
                sentence = "";
                sentence2 = "";
            }
        }
    }
}
```

```

        //clear sentence for new data
    }
}

```

### TC65GPS

Klasa ta zawiera w sobie obsługę weryfikacji poprawności odebranego zdania protokołu NMEA, oraz analizator składni (*ang. parser*) tego protokołu. Metoda ta działa na zasadzie rozdzielania łańcuchów znakowych w miejscach występowania znaku przecinka, który oddziela od siebie kolejne informacje w zdaniu NMEA. Protokół NMEA jest tak skonstruowany, że konkretnym danym odpowiadają z góry ustalone miejsca, więc w celu wydobycia określonej danej wystarczy znać jej pozycję w zdaniu. Listing przedstawia część metody parsującej, odpowiedzialnej za analizę zdania typu *\$GPGGA*.

```

public void parseNMEA(String sentence) throws Throwable{
    String[] sentenceData;
    sentenceData = strSplit.split(',', sentence);
    // split NMEA sentence to components

    if (sentenceData[0].equals("$GPGGA")) {
iMletMain.tc65Location.setTime(sentenceData[1]);
        //UTC Time HHMMSS.SSS
iMletMain.tc65Location.setLatitude(sentenceData[2]);
        //Latitude
iMletMain.tc65Location.setLatitudeDir(sentenceData[3]);
        //Latitude direction
iMletMain.tc65Location.setLongitude(sentenceData[4]);
        //Longitude
iMletMain.tc65Location.setLongitudeDir(sentenceData[5]);
        //Longitude direction
if (sentenceData[6].endsWith("0")) {
//Fix 0:No Fix, 1:SPS Mode, 2:DiffGPS, SPS Mode, 3:PPS Mode
iMletMain.tc65Location.setFix("No valid Fix");
}
if (sentenceData[6].endsWith("1")) {
iMletMain.tc65Location.setFix("GPS SPS Mode");
}
if (sentenceData[6].endsWith("2")) {
iMletMain.tc65Location.setFix("DiffGPS SPS Mode");
}
if (sentenceData[6].endsWith("3")) {
iMletMain.tc65Location.setFix("GPS PPS Mode");
}
iMletMain.tc65Location.setSatellitesUsed(sentenceData[7]);
//Satellites used 0-12
iMletMain.tc65Location.setHDOP(sentenceData[8]);
//HDOP
iMletMain.tc65Location.setAltitude(sentenceData[9]);
//Altitude in meters
    }
}

```

### TC65Location

Klasa TC65Location definiuje obiekt służący do przechowywania aktualnej lokalizacji, oraz metod dostępu do tych danych. Klasa składa się z prywatnych pól przechowujących poszczególne części pozycji, jak np.: długość czy szerokość geograficzną. Dopełnieniem klasy są metody dostępu do tych pól, pozwalające wpisać aktualną lokalizację, oraz ją pobrać. Metody te odpowiednio przetwarzają dane zawarte w polach prywatnych, tak aby przesyłane dane zgodne były z oczekiwaniami systemu nadzoru, np. aby prędkość wyświetlana była w kilometrach na godzinę, a nie w węzłach, jak to przedstawia odbiornik GPS.

Listing zawiera definicję pól prywatnych klasy TC65GPS służących do przechowywania aktualnej pozycji.

```
public class TC65Location {  
    private String latitude = "";  
    private String latitudeDir = "";  
    private String longitude = "";  
    private String longitudeDir = "";  
    private float altitude = 0;  
    private float speed = 0;  
    private String date = "";  
    private String time = "";  
    private String fix = "No valid Fix";  
    private String fixD = "No valid Fix";  
    private int satellitesUsed = 0;  
    private float hdop = 0.0f;  
}
```

Listing przedstawia przykładowy sposób dostępu do pola przechowującego aktualną prędkość, wraz z jej konwersją.

```
public float getSpeed() {  
    return (float) (int) (((speed * 1.852f) + 0.005f) * 100.0f)  
    / 100.0f;  
}
```

### TC65GPRSSender

Klasa ta, także będąca wątkiem, odpowiada za przesyłanie danych lokalizacyjnych do systemu nadzoru oraz za generowanie unikalnego identyfikatora trasy UID. Identyfikator ten jest tworzony przy każdym starcie aplikacji i pozostaje niezmienny podczas całego jej działania. Jest on wykorzystywany podczas wizualizacji przebiegu trasy do rozpoznania do której trasy należy dany rekord w bazie danych, w przypadku gdy istnieje więcej niż jedna trasa. Przesyłanie danych do serwera odbywa się za pośrednictwem protokołu HTTP. Wywoływany jest odpowiedni adres, zapisany w konfiguracji programu, pod którym znajduje się serwet obsługujący żądanie. Dane lokalizacyjne, numer UID, data oraz czas przesyłane są jako parametry żądania HTTP. Listing przedstawia główną pętlę wątku, metodę wysyłającą dane za pośrednictwem protokołu http, oraz funkcję generującą unikalny identyfikator.

```

public void mainLoop() {
    try {
        if (iMletMain.tc65Location.isFixValid()) {
            sendData();
        }
        Thread.sleep(1000);
    }

private void sendData() {
    String url = iMletMain.tc65Preferences.serverUrl;
    String time = iMletMain.tc65Location.getTime();
    String date = iMletMain.tc65Location.getDate();
    String lat =
String.valueOf(iMletMain.tc65Location.getLatDD());
    String lat_d = iMletMain.tc65Location.getLatDir();
    String lon =
String.valueOf(iMletMain.tc65Location.getLonDD());
    String lon_d = iMletMain.tc65Location.getLonDir();
    int resp = 0;
    String url_out = "";

    UID = checkLength(UID, 20);
    date = checkLength(date, 6);
    time = checkLength(time, 6);
    lat = checkLength(lat, 9);
    lat_d = checkLength(lat_d, 1);
    lon = checkLength(lon, 10);
    lon_d = checkLength(lon_d, 1);

    try {
        do {
            url_out = (url + "?uid=" + UID + "&date=" + date +
"&time=" + time + "&lat=" + lat + "&lat_d=" + lat_d + "&lon=" + lon
+ "&lon_d=" + lon_d);

            HttpURLConnection httpConn = (HttpURLConnection)
Connector.open(url_out);
            resp = httpConn.getResponseCode();
            if (resp == 200) {
                System.out.println("WYSLANO! HTTP:200");
            } else {
                System.out.println("ERROR");
            }
        } while (resp != 200); // do in loop until resp =
HTTP:200
    }

    private void genUID() {
        Random rnd = new Random();
        long lUID = 0;

        do {
            rnd.setSeed(System.currentTimeMillis());
            lUID = rnd.nextLong();
        } while (lUID == 0);
    }
    // generate until non-zero

```

```

        if (lUID < 0) {
            lUID = lUID * -1;
// change sign if under 0
        }
        this.UID = String.valueOf(lUID);
    }

```

## 2.4. Zasada działania systemu nadzoru

Część stacjonarna systemu to aplikacja webowa pełniąca rolę usługi, uruchomiona na odpowiednim serwerze aplikacyjnym JavaEE. W ramach projektu wykorzystany został serwer „GlassFish Server Open Source Edition v 4” firmy Oracle.

Aplikacja składa się z dwóch stron Java ServerPages:

- index.jsp,
- map.jsp,

oraz czterech serwletów:

- SaveData.do,
- ShowRoutes.do,
- GetXml.do,
- DelRoute.do.

Ponadto część ta zawiera bazę danych w postaci serwera PostgreSQL.

Dane lokalizacyjne przesłane z układu zdalnej lokalizacji trafiają do serwletu SaveData.do w postaci parametrów wywołania żądania. Parametry te są odczytywane zgodnie z listingiem.

```

String uid = request.getParameter("uid");
String date = request.getParameter("date");
String time = request.getParameter("time");
String lat = request.getParameter("lat");
String lat_d = request.getParameter("lat_d");
String lon = request.getParameter("lon");
String lon_d = request.getParameter("lon_d");

```

Tak pobrane dane są następnie zapisywane w bazie danych za pomocą technologii JDBC oraz języka SQL, co prezentuje listing.

```

Connection dbConn;
dbConn = DriverManager.getConnection("jdbc:postgresql://" + DB_URL +
    "/" + DB_NAME, DB_USER, DB_PASS);
Statement stat = dbConn.createStatement();
stat.executeUpdate("INSERT INTO " + DB_TABLE_NAME + " VALUES(default,
    '" + uid + "', '" + date + "', '" + time + "', '" + lat + "', '" +
    lat_d + "', '" + lon + "', '" + lon_d + "')");
dbConn.close();

```

Na rysunku 3.5 przedstawiono dane zapisane już w tabeli bazy danych.

SQL Editor

Graphical Query Builder

Previous queries

▼

Delete

Delete All

SELECT \* FROM locations

Output pane

Data Output

Explain

Messages

History

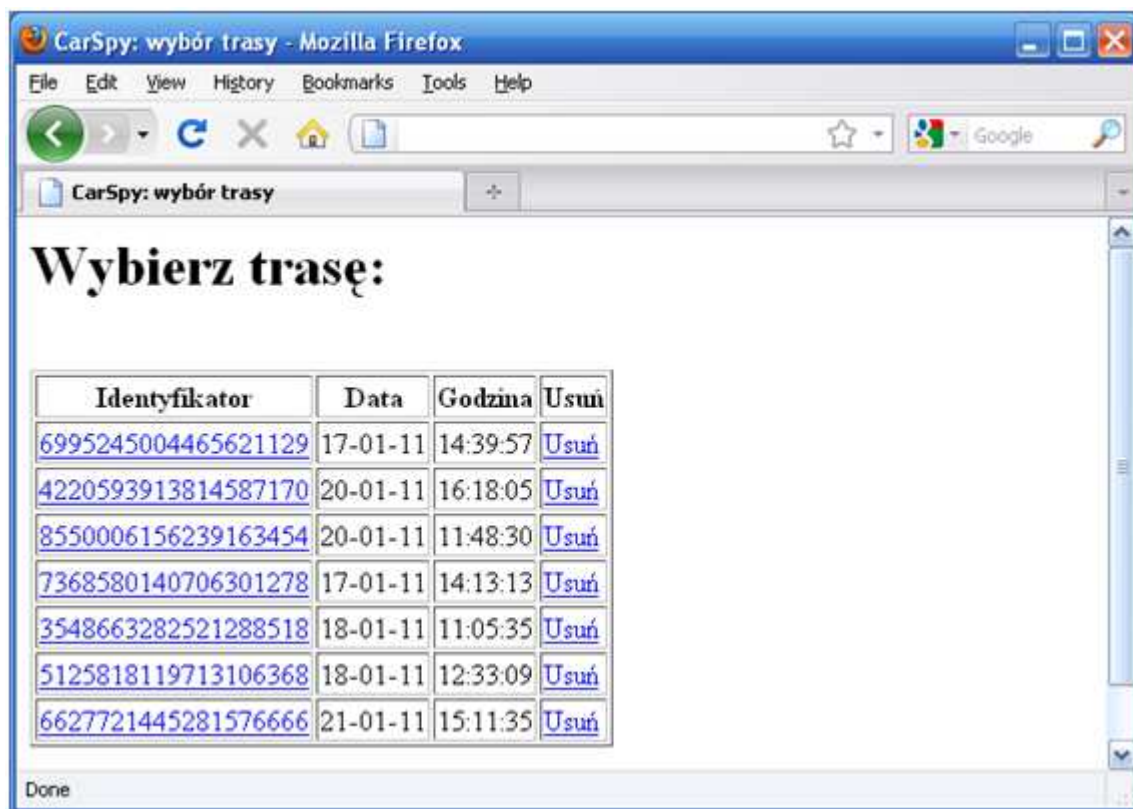
	id integer	uid character varying(36)	date character	time character	lat character	lat_d chara	lon character	lon_d chara
1067	1117	304475050908392511	151214	020404	50.29661	N	18.96099	E
1068	1118	304475050908392511	151214	020411	50.29662	N	18.96099	E
1069	1119	304475050908392511	151214	020419	50.29662	N	18.96099	E
1070	1120	304475050908392511	151214	020426	50.29661	N	18.961	E
1071	1121	304475050908392511	151214	020431	50.29661	N	18.961	E
1072	1122	304475050908392511	151214	020436	50.29662	N	18.961	E
1073	1123	304475050908392511	151214	020445	50.29662	N	18.961	E
1074	1124	304475050908392511	151214	020450	50.29662	N	18.961	E
1075	1125	304475050908392511	151214	020453	50.29662	N	18.961	E
1076	1126	304475050908392511	151214	020500	50.29661	N	18.961	E
1077	1127	304475050908392511	151214	020509	50.29661	N	18.96099	E
1078	1128	304475050908392511	151214	020516	50.29661	N	18.96099	E
1079	1129	304475050908392511	151214	020525	50.2966	N	18.96099	E
1080	1130	304475050908392511	151214	020532	50.29662	N	18.96099	E
1081	1131	304475050908392511	151214	020541	50.29662	N	18.96098	E
1082	1132	304475050908392511	151214	020548	50.29661	N	18.96098	E
1083	1133	304475050908392511	151214	020550	50.29661	N	18.96098	E
1084	1134	304475050908392511	151214	020555	50.29661	N	18.96098	E
1085	1135	304475050908392511	151214	020558	50.29661	N	18.96098	E

Rys 3.5 – Dane lokalizacyjne zapisane w bazie danych (zrzut ekranu aplikacji pgAdmin III)

## 2.5. Wizualizacja danych

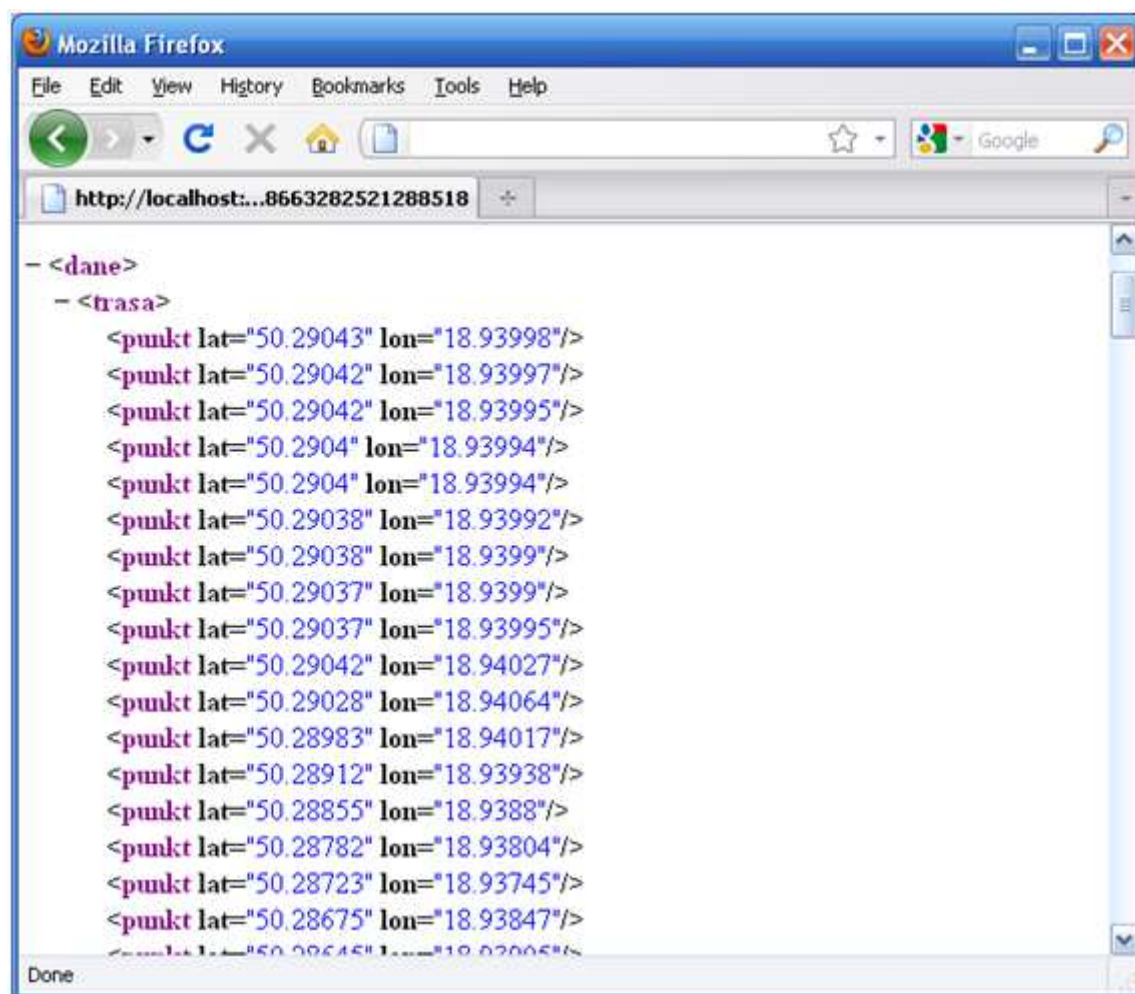
Dane lokalizacyjne stanowiące punkty o współrzędnych geograficznych zapisanych w bazie danych są mało przystępną formą dla człowieka. Dlatego też aplikacja umożliwia wizualizację zebranych punktów na mapie. Wykorzystana do tego celu została usługa Google Maps firmy Google. Jest to usługa bezpłatna dla użytku domowego jak i komercyjnego, polegająca na możliwości wstawienia map na własną stronę internetową. Mapy można rozszerzać o dodatkowe funkcjonalności, jak markery, trasy czy geokodowanie adresów. Google utworzyło specjalne API ułatwiające dostęp do wyżej wspomnianych usług. W niniejszym projekcie zostało wykorzystane najnowsze dostępne obecnie API (ang. Application Programing Interface) w wersji trzeciej [10].

Dostęp do mapy z wyrysowaną nań trasą odbywa się poprzez serwlet ShowRoutes.do, do którego można się dostać klikając odpowiedni odnośnik na dole strony głównej projektu (index.jsp). Serwlet pobiera z bazy danych informacje o zapisanych trasach, oraz wyświetla je w formie tabeli (Rys. 3.6).

*Rys. 3.6 – Wybór trasy do wyświetlenia*

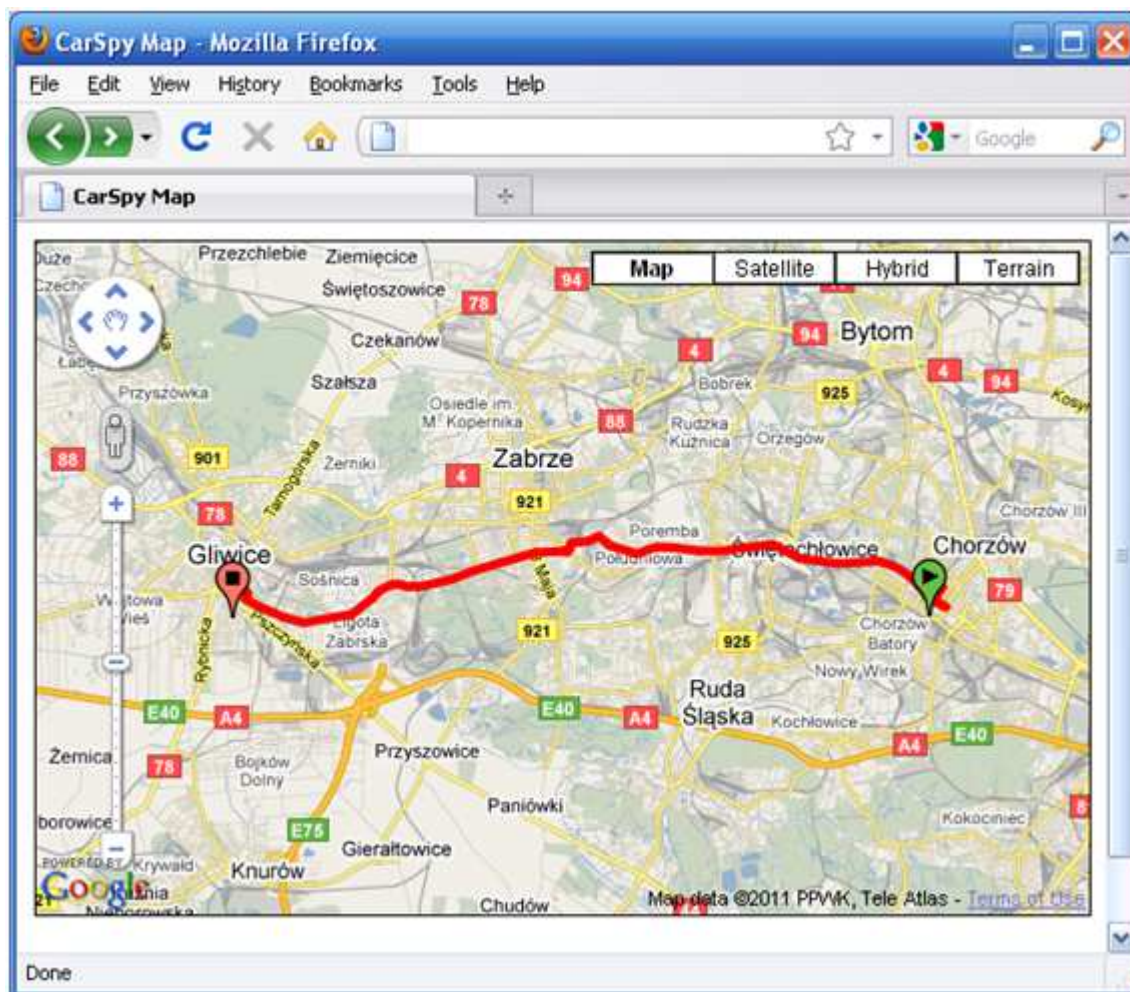
Identyfikatory tras stanowią łącza do strony dynamicznej `map.jsp` zawierające informacje niezbędne do wyświetlenia wybranej trasy. Strona `map.jsp` zawiera obsługę mechanizmu Google Maps API V3. Do wyświetlenia trasy na mapie potrzebny jest zbiór punktów należących do wybranej trasy. Zbiór ten przekazywany jest w postaci dynamicznie generowanego pliku XML (ang. Extensible Markup Language) przez serwlet `GetXml.do` (Rys. 3.7).





Rys. 3.7 – Dynamicznie wygenerowany plik XML (zrzut ekranu przeglądarki internetowej z wygenerowanym dokumentem XML)

Na podstawie powyższego pliku XML, strona dynamiczna map.jsp tworzy tabelę współrzędnych geograficznych oraz poprzez język skryptowy JavaScript przekazuje ją do mechanizmu Google Maps, który wyświetla w przeglądarce mapę z zaznaczoną trasą (Rys. 3.8).



Rys. 3.8 – Mapa z zaznaczoną trasą (zrzut ekranu przeglądarki internetowej z wczytaną interaktywną mapą)

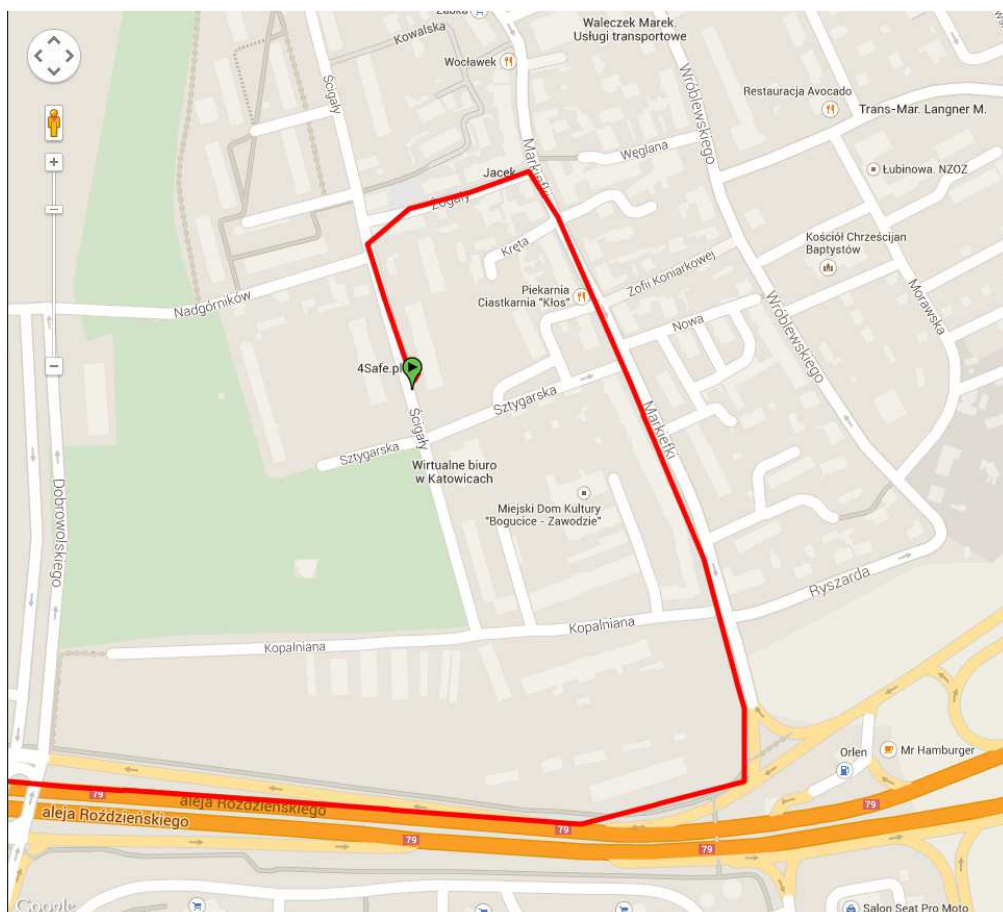
Mapa wyświetlana przez silnik Google Maps jest interaktywna. Możliwa jest zmiana przybliżenia, przesuwanie mapy za pomocą myszy czy zmiana widoku na zdjęcia satelitarne bądź mapę terenową.

## 2.6. Prezentacja przykładowego uruchomienia opracowanego układu

W niniejszym rozdziale znajdują się zrzuty ekranów przedstawiające wizualizacje przykładowych tras, zarejestrowanych za pomocą działającego systemu. Zaprezentowano najciekawsze możliwości jakie daje usługa Google Maps wykorzystana w projekcie.

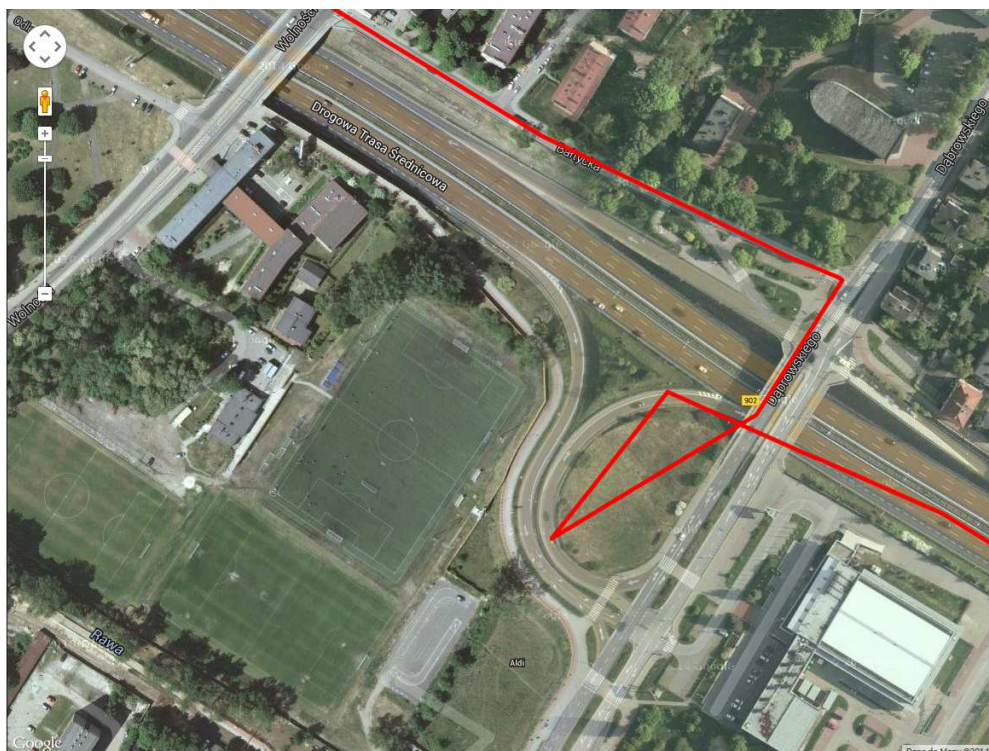


Rys. 3.9 – Mapa przedstawiająca trasę z Chorzowa do Gliwic wyświetloną na widoku drogowym



Rys. 3.10 – Szczegóły wyjazdu z Katowic





Rys. 3.11 – Szczegóły wjazdu na trasę szybkiego ruchu w Chorzowie na zdjęciu satelitalnym

## 4. Podsumowanie i wnioski

W pracy przedstawiono podstawy lokalizacji za pomocą technologii GPS oraz omówiono system komunikacji bezprzewodowej GPRS. Przybliżono język programowania Java, ze szczególnym uwzględnieniem zastosowania go w urządzeniach wbudowanych, za pomocą platformy Java Micro Edition, oraz w zastosowaniach serwerowych, za pomocą platformy Java Enterprise Edition. Dokonano przeglądu dostępnych na rynku modułów do transmisji GPRS, oraz wybrano moduł najbardziej odpowiadający założeniom pracy.

Zaprojektowano sprzęgnięcie odbiornika GPS RGPSM002 firmy Racewood z modułem GPRS TC65 firmy Siemens, przy pomocy istniejącego interfejsu sprzętowego, będącego przedmiotem innej pracy dyplomowej.

Zaprojektowano oraz napisano oprogramowanie obsługujące moduł TC65 w języku Java, mające na celu pobieranie aktualnej lokalizacji oraz przesyłanie jej do zdalnego systemu nadzoru. Wykonano również implementację systemu nadzoru w oparciu o platformę webową Java Enterprise Edition z wykorzystaniem bazy danych PostgreSQL jako magazynu danych.

Wykonano próbne uruchomienie modułu wraz z napisanym oprogramowaniem, oraz zarejestrowano kilka przykładowych tras przemieszczenia obiektu.

W ramach rozwoju projektu z wykorzystaniem modułu do transmisji GPRS, takiego jak TC65, można:

- zaimplementować obsługę portów ogólnego przeznaczenia GPIO w celu rejestracji stanów w jakich znajduje się aktualnie lokalizowany obiekt,

- wykorzystać interfejs audio modułu TC65 w celu podsłuchu otoczenia obiektu,
- utworzyć klasę sterującą modułem za pomocą wiadomości SMS,
- wprowadzić możliwość konfiguracji modułu za pomocą interfejsu WWW.

Wspomniana rozbudowa układu pozwoli wykorzystać wszystkie dostępne możliwości modułu Siemens TC65 oraz platformy JavaME. Przykładem zastosowania takiego układu może być w pełni autonomiczny system pozycjonowania np. floty samochodów firmowych, bądź pomoc w lokalizowaniu zaginionych alpinistów oraz ratowników w terenach górskich.

## Bibliografia

- [1] Rychlicki-Kicior K.: *Ćwiczenia J2ME. Java dla urządzeń mobilnych*, Helion, Gliwice 2006.
- [2] Rychlicki-Kicior K.: *J2ME Praktyczne projekty*, Helion, Gliwice 2007
- [3] Horstmann C. S., Cornell G.: *Java Podstawy*, Helion, Gliwice 2008
- [4] Horstmann C. S., Cornell G.: *Java Techniki zaawansowane*, Helion, Gliwice 2009
- [5] Rychlicki-Kicior K.: *Java EE 6. Programowanie aplikacji WWW*, Helion, Gliwice 2010
- [6] SiRF Technology, Inc.: *NMEA Reference Manual*, San Jose 2008
- [7] <http://aws.amazon.com>
- [8] *TC65 Hardware Interface Description*. Siemens 2006
- [9] <http://jdbc.postgresql.org/>
- [10] <http://code.google.com/apis/maps/documentation/javascript/>
- [11] [http://pl.wikipedia.org/wiki/Global\\_Positioning\\_System](http://pl.wikipedia.org/wiki/Global_Positioning_System)
- [12] <http://tme.eu>
- [13] Kasperczyk Ł.: *Transmisja GPRS z wykorzystaniem modułu TC65 z technologią Java*, Praca magisterska, Politechnika Śląska, Wydział Elektryczny, Gliwice 2008