

# PORTFOLIO

(na wniosek firmy F16)

## Analiza możliwości pozyskania i interpretacji surowych danych pochodzących z odbiorników GPS w urządzeniach mobilnych

*Autor: Sebastian Ernst*

Centrum Inteligentnych Systemów Informatycznych Akademia Górniczo-Hutnicza im. Stanisława Staszica al. Mickiewicza 30, 30-059 Kraków  
budynek C-2 pokój 426 tel.: 12 617 44 53 www.isi.agh.edu.pl [isi@agh.edu.pl](mailto:isi@agh.edu.pl)

# 1 Wprowadzenie

Urządzenia mobilne – przede wszystkim tzw. smartfony i tablety – posiadają w zdecydowanej większości możliwość określania swojej lokalizacji w otwartych przestrzeniach. Głównym narzędziem wykorzystywanym do pozycjonowania jest odbiornik systemu GPS (ang. Global Positioning System) [6]. W nowszych modelach stosowane są odbiorniki będące w stanie odbierać również sygnały z satelitów systemu GLONASS [3].

## 1.1 Problemy pozycjonowania w urządzeniach mobilnych

Problemem systemów pozycjonowania przy pomocy satelitów jest długi czas określania lokalizacji. W zależności od stanu urządzenia i wykonywanej procedury inicjalizacyjnej (cold start, warm start, hot start) wynosi zazwyczaj ok. 30–40 sekund, lecz może trwać nawet kilkanaście minut, co jest wartością niedopuszczalną w przypadku typowych aplikacji na urządzenia mobilne. Dlatego też stosowane są techniki, które mają na celu skrócenie czasu pozyskiwania lokalizacji, określane mianem Assisted GPS. Polegają one na dostarczeniu do urządzenia obliczonych już dla danego okresu czasu (typowo: 1 miesiąc) orbit satelitów oraz wartości funkcji ich pozycji. Wymagają one jednak połączenia z serwerem udostępniającym te dane lub umieszczenia w urządzeniu zbioru danych aktualnych na chwilę określania pozycji.

Odbiorniki GPS wspomagane są często zgrubną pozycją, pozyskiwaną przy pomocy innych modułów komunikacji radiowej, najczęściej radia GSM/UMTS oraz karty sieciowej Wi-Fi. Określenie pozycji możliwe jest dzięki oszacowaniu odległości od stacji bazowych odpowiedniego typu sieci poprzez pomiar siły ich sygnału. Oczywiście, wymagane jest posiadanie bazy lokalizacji stacji bazowych, co z uwagi na dużą dynamikę oraz rozmiar danych oznacza również konieczność wykorzystania łącza danych oraz posiadania dostępu do serwera udostępniającego te dane.

Ostatnim czujnikiem wykorzystywanym do określania pozycji urządzeń mobilnych jest magnetometr, czyli cyfrowy kompas. Obecnie stosowane implementacje obciążone są dość dużą niedokładnością, związaną przede wszystkim z pracą w bezpośredniej bliskości źródeł promieniowania elektromagnetycznego.

## 1.2 Cel raportu

Celem raportu jest określenie możliwości narzędzi pozyskiwania informacji o lokalizacji urządzenia w najpopularniejszych obecnie platformach mobilnych. Opracowanie pełni rolę raportu wspomagającego w ramach projektu związanego z badaniem możliwości określania lokalizacji urządzeń mobilnych wewnątrz budynków.



Rozważane podejścia obejmują symulowanie sygnałów GPS wewnątrz budynków oraz wykorzystanie technik Differential GPS. Implementacja tych podejść wymagałaby z pewnością dostępu do surowych danych z satelitów, istotna jest więc ocena możliwości otrzymania takich danych w wybranych platformach mobilnych.

Do porównania wykorzystano dwie najbardziej rozpowszechnione platformy mobilne: iOS firmy Apple oraz Android, rozwijany przez firmę Google. Z uwagi na konsumencki charakter docelowych aplikacji, w opracowaniu wzięto pod uwagę tylko techniki zgodne z zasadami publikacji aplikacji poprzez oficjalne kanały dystrybucyjne, odpowiednio App Store dla systemu iOS oraz Google Play dla systemu Android.

**Centrum Inteligentnych Systemów Informatycznych** Akademia Górniczo-Hutnicza im. Stanisława Staszica al. Mickiewicza 30, 30-059 Kraków  
budynek C-2 pokój 426 tel.: 12 617 44 53 [www.isi.agh.edu.pl](http://www.isi.agh.edu.pl) [isi@agh.edu.pl](mailto:isi@agh.edu.pl)



## 2 Implementacja pozycjonowania w platformach mobilnych

Z uwagi na złożoność problemu pozyskiwania lokalizacji, opisaną w sekcji 1.1, producenci platform mobilnych dostarczają narzędzi w postaci bibliotek bądź frameworków, opakowujących poszczególne urządzenia lokalizacyjne i izolujące twórcę aplikacji od danych pochodzących z nich.

W niniejszej sekcji opisano techniki pozyskiwania lokalizacji w dwóch rozważanych systemach mobilnych.

### 2.1 Pozycjonowanie w systemie Apple iOS

Za określanie pozycji i towarzyszących jej parametrów odpowiada w systemie iOS framework Core Location [1]. Jego trzon stanowi klasa CLLocationManager, wysokopoziomowy interfejs do wszystkich narzędzi pozycjonujących urządzeń iPhone, iPad oraz iPod touch.

Klasa ta pozwala na śledzenie następujących parametrów:

- współrzędne geograficzne (w układzie odniesienia WGS-84),
- wysokość nad poziomem morza (w metrach),
- prędkość (w m/s),
- kierunek (w stopniach; tylko dla urządzeń wyposażonych w kompas),
- dokładność pozycji w poziomie i w pionie (w metrach).

Wersja: 6f0efb8 2 2014-01-21 17:57:18 +0100

Jedną z najważniejszych kwestii przy projektowaniu urządzeń mobilnych jest zarządzanie energią. Praktycznie wszystkie moduły służące do pozyskiwania lokalizacji wiążą się z dużym wydatkiem energetycznym. W związku z tym, producenci starają się z poziomu oprogramowania ograniczyć wpływ energii, poprzez implementację odpowiednich strategii lokalizacyjnych.

Klasa CLLocationManager pozwala programiście na określenie kilku parametrów pozwalających systemowi operacyjnemu na lepsze gospodarowanie zasobami energii. Parametr activityType pozwala na określenie celu, w którym aplikacja pobiera lokalizację:

- nawigacja pojazdów kołowych (CLActivityTypeAutomotiveNavigation),
- nawigacja dla innych środków transportu (CLActivityTypeOtherNavigation),
- aplikacje „sportowa” (CLActivityTypeFitness),
- inne zastosowania (CLActivityTypeOther).

Centrum Inteligentnych Systemów Informatycznych Akademia Górniczo-Hutnicza im. Stanisława Staszica al. Mickiewicza 30, 30-059 Kraków  
budynek C-2 pokój 426 tel.: 12 617 44 53 www.isi.agh.edu.pl [isi@agh.edu.pl](mailto:isi@agh.edu.pl)

Śledzenie pozycji polega w przypadku frameworku Core Location na wywoływaniu funkcji typu callback przy każdej aktualizacji lokalizacji bądź kierunku – osobno dla każdego z tych dwóch parametrów. Domyślnie aktualizacje te następują co sekundę.

Programista ma jednak możliwość wpłynięcia na dodatkowe parametry:

- pożądana dokładność pozycji (`desiredAccuracy`),
- odległość, której pokonanie wyzwała wywołanie funkcji callback dla lokalizacji (`distanceFilter`),
- kąt, o który musi zmienić się kierunek, aby została wywołana funkcja callback dla kierunku (`headingFilter`).  
Parametr `desiredAccuracy`, określający pożądaną przez aplikację dokładność pozycji, może przyjmować wartości:
- najlepsza (`kCLLocationAccuracyBest`): najwyższa dokładność możliwa do osiągnięcia w danych warunkach,
- najlepsza, dla nawigacji (`kCLLocationAccuracyBestForNavigation`): j.w., ale wykorzystywane są też dodatkowe czujniki ruchu; wiąże się z wyższym o ok. 20% zużyciem energii,
- cztery opcje, ustawiające dokładność na 10 m, 100 m, 1 km lub 3 km.  
Wysokopoziomowe API systemu iOS nie daje niestety dostępu do surowych danych GPS, nie udostępnia ono też możliwości uzyskania informacji o satelitach „widzianych” w danej chwili przez odbiornik GPS. Pozyskanie surowych danych byłoby prawdopodobnie możliwe w przypadku wykonania tzw. operacji jailbreak, czyli usunięcia ograniczeń nałożonych przez producenta. Aplikacja taka jednak nie mogłaby być dystrybuowana oficjalnymi kanałami, więc zgodnie z wyjaśnieniami w sekcji 1.2 – wariant ten nie podlegał analizie.

Wersja: 6f0efb8 3 2014-01-21 17:57:18 +0100

## 2.2 Pozycjonowanie w systemie Google Android

Sposób pozyskiwania lokalizacji w aplikacjach dla systemu Google Android ewoluował w czasie. We wcześniejszych wersjach systemu dostępny był framework `android.location`, który pozwalał na dość dokładną interakcję z urządzeniami pozycjonującymi, wymagał jednak pewnego nakładu pracy. W nowszych wersjach systemu, Google forsuje interfejs Google Location Services API, będący częścią Google Play Services [4]. Można się więc spodziewać, że framework `android.location` może przestać być dostępny w którejś z kolejnych wersji systemu operacyjnego.

### 2.2.1 Framework `android.location`

Centralnym elementem systemu jest klasa `LocationManager`, która pozwala na odpytywanie tzw. dostawców lokalizacji (ang. `location providers`). Każdy z tych abstrakcyjnych „dostawców” posiada następujące cechy:

- dokładność (wysoka/niska),
- wymagania dotyczące zasilania (poziom wysoki, średni lub niski),
- wymagane urządzenia fizyczne lub usługi, takie jak:
  - dostęp do sieci komórkowej,
  - dostęp do połączenia internetowego,
  - dostęp do satelitarnego systemu pozycjonowania (np. GPS),
- informacja, czy korzystanie związane jest z opłatami,
- wsparcie dla pozycji 3D (zawierającej wysokość n.p.m.),
- wsparcie dla pobierania kierunku,
- wsparcie dla pomiaru prędkości.

Po wybraniu dostawcy, aplikacja może zasubskrybować zmiany dostarczanej przez niego lokalizacji przy pomocy klasy `LocationListener` oraz metody `requestLocationUpdates` klasy `LocationManager`.

Istnieją także klasy `GpsStatus.Listener` oraz `GpsStatus.NmeaListener`, obsługujące ściśle dostawców skojarzonych z odbiornikiem GPS. Pierwsza z nich dostarcza zdarzeń związanych z aktualizacją stanu odbiornika GPS, druga natomiast pozwala na dostęp do surowych danych GPS. Działanie ich nie jest jednak identyczne na wszystkich urządzeniach; testy na nowszych urządzeniach z systemem Android wskazują, że na dużej ich części metody nie działają wcale.

### 2.2.2 Google Location Services API

Google Location Services API jest częścią nowego pakietu Google Services [5]. Zawiera on proste w użyciu, wysokopoziomowe interfejsy do istotnych funkcjonalności systemu operacyjnego. Jest on częścią aplikacji Google Play Store, a więc aplikacje korzystające z Google Services mogą działać tylko na urządzeniach zdolnych do pobierania aplikacji ze sklepu Google Play.

Konstrukcja i funkcjonalność Location API przypomina framework Core Location z systemu iOS, jej funkcjonalność jest jednak (na chwilę przygotowania raportu) bardziej ograniczona.

Wersja: 6f0efb8 4 2014-01-21 17:57:18 +0100

Aplikacja może pobrać lokalizację jednorazowo (przy pomocy klasy LocationClient), lub zasubskrybować zmiany lokalizacji (przy pomocy klasy LocationRequest). W drugim przypadku, subskrypcja może określać następujące parametry:

- okres czasu ważności subskrypcji lub chwilę jej wygaśnięcia,
- częstotliwość aktualizacji,
- minimalne rejestrowane przesunięcie.

Podobnie jak Core Location w iOS, również Location API nie pozwala na dostęp do surowych danych z odbiornika GPS.



### 3 Podsumowanie

Obie rozważane platformy mobilne – Apple iOS oraz Google Android – posiadają interfejsy programistyczne służące do pobierania informacji o lokalizacji urządzenia. W przypadku iOS, technika pobierania informacji o lokalizacji nie uległa zasadniczym zmianom od chwili udostępnienia SDK w 2008 roku; framework był tylko wzbogacany w miarę wprowadzania w nowych modułach lokalizacyjnych (np. kompasu) lub udostępniania nowych funkcjonalności (jak np. geofencing – możliwość ustawienia alertów związanych z znalezieniem się urządzenia w określonym obszarze lub jego opuszczeniem).

W przypadku systemu Android rozwój jest bardziej burzliwy – „tradycyjny” mechanizm oparty o framework android.location jest aktualnie wygaszany na rzecz nowego Location Services API, przypominającego Core Location. „Stary” mechanizm dawał programiście bogatsze możliwości (m.in. pobieranie surowych danych GPS, które jednak nie działało na wielu modelach urządzeń). Było to jednak okupione większym nakładem pracy, i m.in. koniecznością samodzielnego określania strategii lokalizacji. Nowe API zwalnia programistę z tych obowiązków, ale niestety nie umożliwia dostępu do informacji o „widzianych” satelitach czy do surowych danych NMEA.

Podsumowując, wszelkie techniki polegające na wykorzystaniu surowych danych GPS są niepraktyczne w przypadku konsumenckich aplikacji dla wiodących platform mobilnych. Wiązałyby się one z koniecznością usuwania ograniczeń nakładanych przez producentów na urządzenia, co stoi w bezpośredniej sprzeczności z komfortem użytkowania oferowanym przez współczesne urządzenia mobilne.

Bardziej obiecującą drogą pozyskiwania informacji o lokalizacji wewnątrz budynków są działające z niską mocą nadajniki radiowe instalowane wewnątrz budynków (ang. beacons). Firma Apple opracowała i wprowadziła w wersji 7.0 systemu iOS technologię iBeacon [2], działającą w oparciu o system Bluetooth. Narzędzia programowe zewnętrznych dostawców (np. Radius Networks [7]) pozwalają na korzystanie z nadajników kompatybilnych z iBeacon również z poziomu aplikacji dla systemu Android.



## Literatura

- [1] Apple, Inc. iOS Developer Library. <http://developer.apple.com/library/ios/>.
- [2] Apple, Inc. iOS: Understanding iBeacon. <http://support.apple.com/kb/HT6048>. Wersja: 6f0efb8 5 2014-01-21 17:57:18 +0100
- [3] Federal Space Agency, Russian Federation. Information analytical centre of GLONASS and GPS controlling. <http://glonass-iac.ru/en/>.
- [4] Google, Inc. Android Developers, android.location: package summary. [http:// developer.android.com/reference/android/location/package-summary.html](http://developer.android.com/reference/android/location/package-summary.html).
- [5] Google, Inc. Google Location Services APIs. [http:// developer.android.com/google/ play-services/location.html](http://developer.android.com/google/play-services/location.html).
- [6] B.W. Parkinson and J.J. Spilker. Global Positioning System: Theory and Applications. Number v. 1 in Global Positioning System: Theory and Applications. American Institute of Aeronautics and Astronautics, 1996.
- [7] Radius Networks, Press Release. Radius Networks announces open source Android iBeacon SDK.