

ANDROID mobiltelefon alkalmazása GSM alapú helymeghatározó rendszerben

Maczák Balázs
Mérnök Informatikus BSc.



Témavezetők:
Tihanyi Attila
Takács György
Feldhoffer Gergely
2010

1. Nyilatkozat

Alulírott Maczák Balázs, a Pázmány Péter Katolikus Egyetem Információs Technológiai Karának hallgatója kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, és a diplomamunkában csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a Szakdolgozatot más szakon még nem nyújtottam be.

.....

Maczák Balázs

2. Tartalomjegyzék

1. Nyilatkozat.....	2
3. Tartalmi összefoglaló	5
4. Abstract.....	6
5. Bevezetés.....	7
6. Helymeghatározás.....	7
7. Módszerek.....	8
GPS alapú helymeghatározás.....	8
Leszármazott autonóm helymeghatározás.....	8
GSM alapú helymeghatározás.....	9
Helymeghatározás a szolgáltató által.....	9
GSM alapú helymeghatározás Cella ID segítségével.....	9
Helymeghatározás ívmetszéssel	10
Helymeghatározás a háromszögelés elvével.....	13
Helymeghatározás elve az észlelt időkülönbség mérése alapján.....	14
Az egyetemi megoldás	15
8. A feladat.....	16
Android és HTC G1 (<i>Dream</i>).....	16
9. Fejlesztés Android környezetben.....	17
10. A mért adatok tárolására használt adatbázis	18
11. A feladat megoldása.....	19
GPS adatok.....	19
GSM adatok.....	20
12. A kész program	21
13. Eltérések a Samba és az Android között.....	22
Az Android által készített mérések	26
14. Klasszifikáció.....	28

15.	Konklúzió.....	30
16.	Problémák, további feladatok.....	31
17.	Javaslatok.....	32
18.	Függelék.....	33
	Rövidítések jegyzéke.....	33
	Felhasznált irodalom.....	34
	Falcom Samba 75 specifikáció.....	35
	ADP specifikáció.....	36

Törölt:	Nyilatkozat	. 2¶
	Tartalmi összefoglaló	. 5¶
	Abstract	. 6¶
	Bevezetés	. 7¶
	Helymeghatározás	. 7¶
	Módszerek	. 8¶
	GPS alapú helymeghatározás	. 8¶
	Leszármazott autonóm helymeghatározás	. 8¶
	GSM alapú helymeghatározás	. 9¶
	Helymeghatározás a szolgáltató által	. 9¶
	GSM alapú helymeghatározás Cella ID segítségével	. 9¶
	Helymeghatározás ívmetszéssel	. 10¶
	Helymeghatározás a háromszögélés elvével	. 13¶
	Helymeghatározás elve az észlelt időkülönbség mérése alapján	. 14¶
	Az egyetemi megoldás	. 15¶
	A feladat	. 16¶
	Android és HTC G1 (<i>Dream</i>)	. 16¶
	Fejlesztés Android környezetben	. 17¶
	A mért adatok tárolására használt adatbázis	. 18¶
	A feladat megoldása	. 19¶
	GPS adatok	. 19¶
	GSM adatok	. 20¶
	A kész program	. 21¶
	Eltérések a Samba és az Android között	. 22¶
	Az Android által készített mérések	. 26¶
	Klasszifikáció	. 28¶
	Konklúzió	. 30¶
	Problémák, további feladatok	. 31¶
	Javaslatok	. 32¶
	Függelék	. 33¶
	Rövidítések jegyzéke	. 33¶
	Felhasznált irodalom	. 34¶
	Falcom Samba 75 specifikáció	. 35¶
	ADP specifikáció	. 36¶

Formázott: Címsor 1, Térköz
Utána: 12 pt

3. Tartalmi összefoglaló

Korunk fejlett technológiájának köszönhetően egyre elterjedtebbek az olyan alkalmazások, rendszerek, amelyek valamilyen módon egy mobil készülék pozíciójának meghatározására épülnek. A legelterjedtebb módszer a helymeghatározásra a GPS (Global Positioning System), mely a föld körül keringő 24 műhold segítségével, a műholdaktól való távolság alapján számolja a pozíciót. Ez a rendszer valóban hibátlanul működik abban az esetben, ha a föld körül keringő műholdak láthatók. Ellenkező esetben viszont (például beltéren, esetleg szűk utcákban) a műholdaktól való távolság ismeretének hiányában a GPS alapú helymeghatározás sem lehetséges.

Erre a problémára kívánunk megoldást nyújtani egy alternatív helymeghatározási lehetőséggel, mely nem a GPS műholdakra, hanem a GSM (Global System for Mobile Communications) adótornyokra épít, így ugyanúgy működik, akár kültéren van a készülék, akár beltéren. Ezt a rendszert a korábbi években már elkészítették az Egyetemen. Ez állt egy laptopból, amely a számolásokat végezte, valamint egy GSM adóvevő készülékből, amely a laptopra volt kötve.

A rendszer jól működik, egyetlen hibája a nagy méret, ebből adódóan a hordozhatatlanság. Az én feladatomból volt ennek a rendszernek a helyettesítése egy Android alapú mobiltelefonnal oly módon, hogy az eredeti funkciókat a lehető legnagyobb mértékben megtartsam. Ennek a feladatnak két része van: az első egy olyan szoftver készítése, amely képes egy terület GSM adatainak rögzítésére, a második pedig annak a megbecslése, hogy a megírt szoftverrel milyen eredmények várhatók a korábbi megoldáshoz képest.

4. Abstract

Korunk fejlett technológiájának köszönhetően egyre elterjedtebbek az olyan alkalmazások, rendszerek, amelyek valamilyen módon egy készülék pozíciójának meghatározására épülnek. A legelterjedtebb módszer a helymeghatározásra a GPS (Global Positioning System), mely a föld körül keringő 24 műhold segítségével, a műholdaktól való távolság alapján számolja a pozíciót. Ez a rendszer valóban hibátlanul működik abban az esetben, ha a föld körül keringő műholdak láthatók. Ellenkező esetben viszont (például beltéren, esetleg szűk utcákban) a műholdaktól való távolság ismeretének hiányában a GPS alapú helymeghatározás sem lehetséges. Erre a problémára kívánunk megoldást nyújtani egy alternatív helymeghatározási lehetőséggel, mely nem a GPS műholdakra, hanem a GSM adótornyokra épít, így ugyanúgy működik, akár kültéren van a készülék, akár beltéren. Ezt a rendszert a korábbi években már elkészítették az Egyetemen. Ez állt egy laptopból, amely a számolásokat végezte, valamint egy GSM adóvevő készülékből, amely a laptopra volt kötve. A rendszer jól működik, egyetlen hibája a nagy méret, ebből adódóan a hordozhatatlanság. Az én feladatomból volt ennek a rendszernek a helyettesítése egy Android alapú mobiltelefonnal oly módon, hogy az eredeti funkciókat a lehető legnagyobb mértékben megtartsam. Ennek a feladatnak két része van: az első egy olyan szoftver készítése, amely képes egy terület GSM adatainak rögzítésére, a második pedig annak a megbecslése, hogy a megírt szoftverrel milyen eredmények várhatók a korábbi megoldáshoz képest.

5. Bevezetés

Dolgozatomban először ismertetem a feladatot, bemutatom a lényegesebb technikákat, majd rátérek arra, hogy mivel foglalkoztam az elmúlt egy évben. Végül összefoglalom a tapasztalataimat, és fölvázolom, hogy jelenleg a problematika milyen stádiumban van, illetve teszek néhány javaslatot is, hogy milyen irányba lenne érdemes tovább haladni.

Megjegyzés [A1]: AZ egyetemen foglalkozunk alternatív helymeghatározási módszerekkel, mint pl a GSM alapú rendszer. Az én feladatom a továbbfejlesztéssel kapcsolatba: „ide kell írni a kiírásban szereplő feladatokat és azok magyarázatát. Azért van rá szükség, hogy az összefoglalóban még egyszer mint elvégzett feladatot leírassuk.

6. Helymeghatározás

A technológia fejlődésének köszönhetően egyre elterjedtebbé válnak az olyan szolgáltatások, alkalmazások, melyek valamilyen módon vagy a saját, vagy mások pozíciójának ismeretén alapulnak. Gondoljunk például egy autóban a GPS alapú navigációs segédeszközre, vagy, hogy a Google Maps is mutatja a készülék pozícióját a megjelenített térképen. Ezek az alkalmazások mind-mind ugyanarra az alapvető kérdésre vezethetők vissza: hogyan tudjuk megmondani egy készülék helyzetét?

7. Módszerek

Most ismertetném a helymeghatározásban eddig használt főbb megoldásokat, ezek előnyeit, illetve hiányosságait.

GPS alapú helymeghatározás

A ma használt talán legelterjedtebb helymeghatározási módszer a GPS. Ebben az esetben a helymeghatározás 24 db műhold segítségével történik, amiből egyszerre általában 7-12 látható. A műholdas helymeghatározó rendszer időmérésre visszavezetett távolságmérésen alapul. Mivel, ismerjük a rádióhullámok terjedési sebességét, és ismerjük a rádióhullám kibocsátásának és beérkezésének idejét, ezek alapján meghatározhatjuk a forrás távolságát. A háromdimenziós térben három ismert helyzetű ponttól mért távolság pontos ismeretében már meg tudjuk határozni a pozíciót. Ennek a módszernek a pontossága általában méteres nagyságrendben mozog, de léteznek olyan mérnöki felszerelések, melyek 0.5 cm-es pontosságot is adnak, ám ezek a műszerek sem méretük, sem áruk miatt nem alkalmasak átlagfelhasználók általi használatra.

A másik probléma a GPS rendszerrel, hogy mivel műholdakkal történő kommunikáción alapszik, beltéren, vagy szűk utcákban megbízhatatlan, vagy egyáltalán nem működik, ezen kívül használata speciális készüléket igényel, tekintve, hogy a telefonok többsége még nincs felszerelve az adott technológiához szükséges hardware-el.

Leszármazott autonóm helymeghatározás

Ez a módszer a GPS alapú helymeghatározásra épít, de előnye vele szemben, hogy nem igényel folyamatos GPS jelet. Lényege, hogy a mobiltelefon kezdeti helyzetét meghatározzuk GPS rendszer segítségével, majd a későbbi pozíciókat a mobiltelefon maga számolja oly módon, hogy a kiindulási ponttól számítva folyamatosan méri a gyorsulásának nagyságát és irányát, amelyből a pontos eltelt idő ismeretében a sebesség, és abból az elmozdulás számolható. Nagy előnye, hogy a mobiltelefon akkor is képes a pozíciójának meghatározására, ha megszűnt a kapcsolata a GPS műholdakkal.

GSM alapú helymeghatározás

Egy másik lehetséges megközelítése a helymeghatározásnak nem a GPS technológián alapul, hanem azokon az információkon, melyeknek minden telefon a birtokában van, mivel nélkülözhetetlenek a mobiltelefonos kommunikációhoz.

Helymeghatározás a szolgáltató által

Néhány mobiltelefon szolgáltató nyújt szolgáltatásokat helymeghatározásra, ám ez a szolgáltatás igen ritka, használata nehézkes, és ha el is érhető, igen drága. A Cell ID-hez hasonló technikán alapszik, de mivel a szolgáltató számára több információ áll rendelkezésre, valamivel pontosabb.

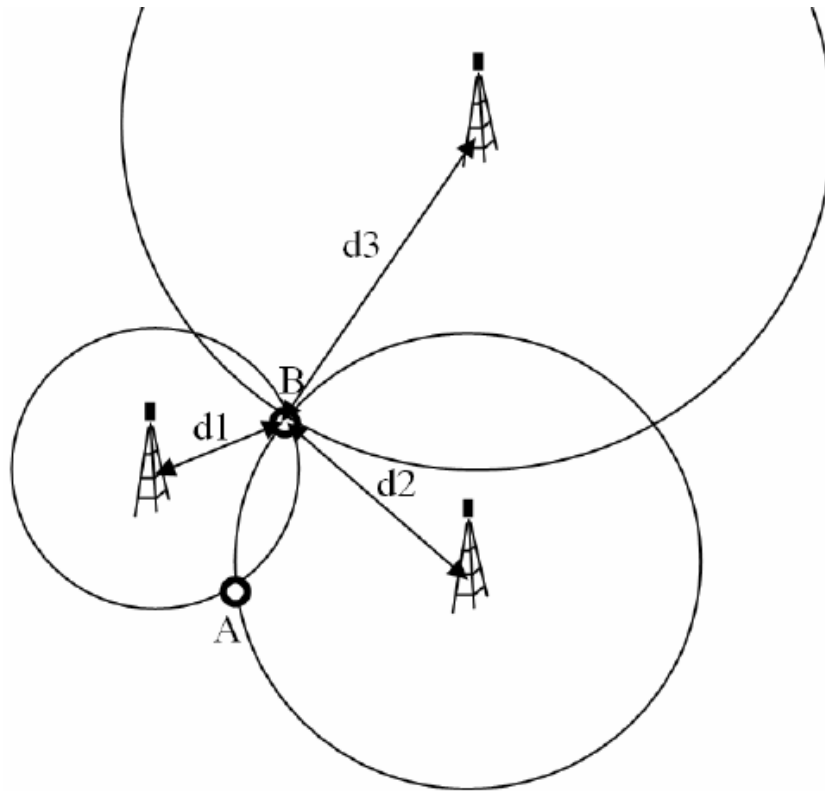
GSM alapú helymeghatározás Cella ID segítségével

Ez a technológia kizárólag olyan információkon alapul, melyeknek minden mobiltelefon a birtokában van. Ebben az esetben azt használjuk ki, hogy a telefon mindig a hozzá legközelebb lévő toronyhoz van csatlakozva (persze ezen még lát több tornyot is), így pontosan tudjuk, hogy melyik cellában vagyunk. Egy ilyen cella mérete országonként változhat, Magyarországon az átlagos cellaméret 1191 m, de ez persze lakott területen jóval kisebb, és a forgalomsűrűség növekedésével szükségszerűen zsugorodik. Ezen kívül a szomszédos cellák figyelésével és követésével a pontosság tovább javítható (ez a tapasztalatok alapján nagyvárosban olyan 20 méteres pontosságot jelent).

Megjegyzés [A2]: Itt meg kellene hivatkozni a Bányai Balázs diploma tervét.

Helymeghatározás ívmetszéssel

Ennek a módszernek a lényege, hogy ha meg tudjuk határozni a távolságunkat két adótoronytól, akkor a lehetséges pozícióink két körív mentén adódik, amelyek rendszerint két pontban metszik egymást. Ha ezek után vesszük a távolságunkat egy harmadik toronytól, akkor a pozícióinkat egyértelműen meg tudjuk határozni igen jó közelítéssel. Az adótoronytól való távolság mérésére két alapvető módszert ismertetnék.



1. ábra: Az ívmetszés elve

1. Távfolságmérés a vett jel szintje alapján

A távfolságmérés alapulhat a jel szintjén, mivel a jelszint a következőktől függ:

- az adó és a vevő távfolságától
- az adó teljesítményétől
- az adóantenna nyereségétől
- iránykarakterisztikájától
- a vevőantenna nyereségétől
- a terjedést befolyásoló tényezőktől

Városi környezetben a veszteség számítására használható az Okumura-Hata modell:

$$L_U = 69.55 + 26.16 \log f - 13.82 \log h_B - C_H + [44.9 - 6.55 \log h_B] \log d$$

Ahol:

L_U = Veszteség városi területen. Mértékegysége: decibel (dB)

h_B = Adó antenna magassága. Mértékegysége: méter (m)

h_M = Vevő antenna magassága. Mértékegysége: méter (m)

f = Átviteli frekvencia. Mértékegysége: megahertz (MHz)

C_H = Antenna magasságának korrekciós faktora

d = Távfolság az adó és a vevő között. Mértékegység: kilométer (km)

Valamint kis városokban:

$$C_H = 0.8 + (1.1 \log f - 0.7) h_M - 1.56 \log f$$

Nagy városokban:

$$C_H = \begin{cases} 9.29 (\log(1.54h_M))^2 - 1.1 & , \text{ if } 150 \leq f \leq 200 \\ 3.2 (\log(11.75h_M))^2 - 4.97 & , \text{ if } 200 < f \leq 1500 \end{cases}$$

2. Távolagszmérés a vett jel késleltetése alapján

Abban az esetben, ha az adó és a vevő szinkronizált, a vett jel késleltetése az adó és a vevő távolagsától függ (tegyük fel, hogy a terjedési sebesség ismert). GPS műholdak esetében a szinkronizált adó a műholdakon van, a vevő órája nem szinkronizált. Az ebből fakadó hibákat a négy műhold alapján számolt egyenletekkel küszöbölik ki.

Elvben semmilyen akadály nincs annak, hogy ugyanezen az elven telefonhálózatoknál is meg tudjuk határozni a távolagsot például az alábbi módon: a mobiltelefon-készülék egy bázisállomás utasítására egy jelcsomagot küld ki, amely tartalmazza a küldés időpontját. A bázisállomás veszi a jelet és a küldési és beérkezési idő különbsége alapján jó közelítéssel számolni tudja a távolagsot a készülék és maga között. A harmadik generációs (3G) mobil rendszerekben az ilyen elven működő helymeghatározások elterjedt nevei:

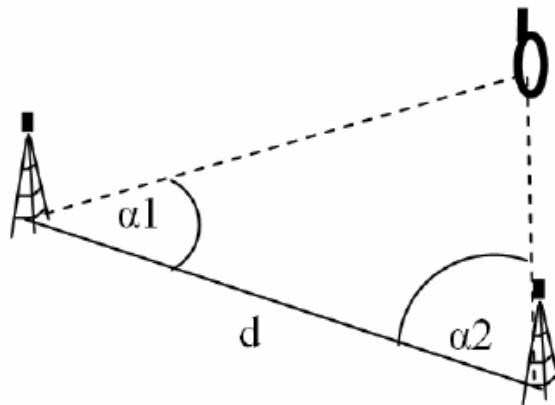
- AFLT – Advanced Forward Link Trilateration
- EFLT – Enhanced Forward Link Trilateration.

Az, hogy egy adott esetben a felmenő vagy a lemenő irányú terjedés ideje alkalmasabb a helymeghatározásra, a műszaki kérdéseken kívül számos jogi és biztonságtechnikai megfontolás alapján dönthető el.

Helymeghatározás a háromszögelés elvével

Ez a rádióállomások (például tipikusan a kalózkodók) bemérésének hagyományos módszere. Működésének alapelve az, hogy különböző helyeken megméri, hogy a keresett adó jele milyen irányból érkezik. Elméletben két mérési pont alapján már meghatározható az adó pozíciója oly módon, hogy a két mérőpontból képzeletben kibocsájtunk egy-egy egyenest a vett jel érkezési irányába, és ezek metszéspontja adja ki a keresett helyet.

Természetesen a gyakorlatban itt is megjelenik egy bizonyos hibafaktor, ám ez több mérés készítésével csökkenthető. Több, egymás melletti vevőantennát antennavektorként használva, az iránymérés az időkülönbségek alapján oldható meg.

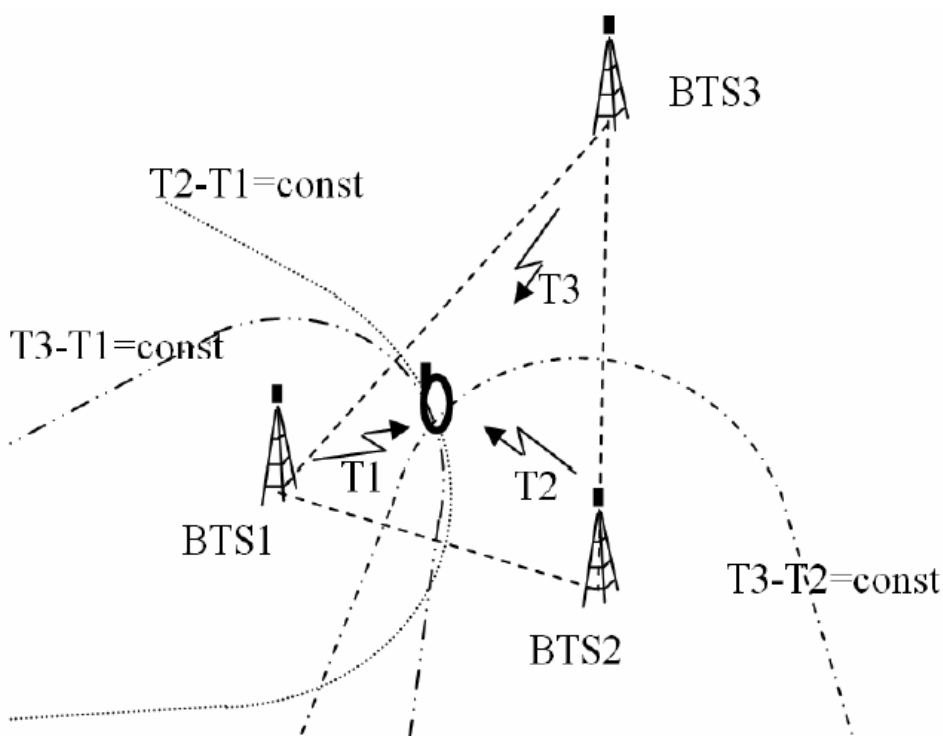


2. ábra: Helymeghatározás a vett rádiójelek alapján

Helymeghatározás elve az észlelt időkülönbség mérése alapján

Abban az esetben, ha a bázisállomások órái szinkronizálva működnek, egy több adótól egyszerre elküldött jel beérkezési idejének időkülönbségei a bázisállomástól való távolságtól függenek. A mobiltelefon saját pontos, de nem szinkronizált órája alapján ugyanazt a jelsorozatot generálva korrelációs módszerrel azt tudja kellő pontossággal mérni, hogy különböző bázisállomásokról egyszerre küldött kódsorozatot milyen időeltéréssel veszi ($T_3 - T_1$, $T_3 - T_2$, $T_2 - T_1$).

A bázisállomások helye a hálózatüzemeltető számára ismert. Azok a pontok, amelyeknek távolságkülönbsége egy bázisállomás-pártól állandó, egy hiperbola mentén helyezkednek el. A mobilkészülék helye tehát a hiperbolák metszéspontjában van.

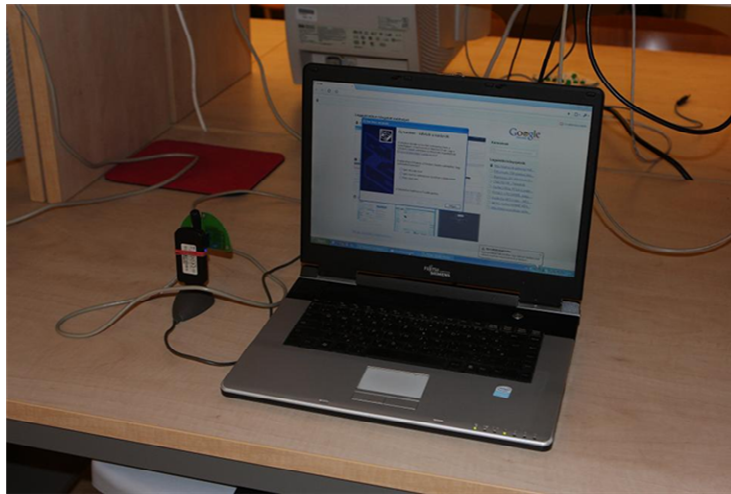


3. ábra: Helymeghatározás az észlelt időkülönbség mérése alapján

Az egyetemi megoldás

A saját módszerünk lényege, hogy az alkalmazás várható helyének területén előzetes méréseket végzünk és ezeket a pozícióval együtt egy adatbázisban eltároljuk. Ezután ezen a területen a cellainformációkat lekérve az eredményeket összevetjük az adatbázisban letároltakkal, és megkapjuk az aktuális pozíciónkat. Ezzel a megközelítéssel természetesen nagyságrendekkel jobb pontosság érhető el, mint a „*real time*” megoldások segítségével, ám előzetes mérést igényel.

A rendszer már korábban kidolgozásra került az Egyetemen, ekkor egy laptop-ra csatolt USB-s GSM adó-vevő (*Falcom Samba 75*) és egy USB-s GPS vevő segítségével mértek, és határozták meg a pozíciót.



4. ábra: Korábbi mérőállomás (bal oldalon a Falcom Samba 75)

Az én feladatom egy mérőalkalmazás készítése volt Android telefonra, mely hordozhatóbb, mint a fent említett mérőrendszer, valamint olyan szempontból is előnyösebb, hogy ez esetben az adatok egyből egy, a későbbi végfelhasználó által használt készüléken vannak.

8. A feladat

A feladatom a félév során egy olyan mérőalkalmazás készítése volt, mely Android mobiltelefonon futva információkat szolgáltat a különböző cellainformációkról.

Android és HTC G1 (*Dream*)

Az Android a Google vezetésével fejlesztett nyílt forráskódú Linux alapú mobil platform, mely mögé több tucat világhírű cég sorakozott fel. A fejlesztők reményei szerint ez a rendszer fogja meghatározni a jövőben a mobil telekommunikációt.

A HTC G1 (más néven *Dream*) egy a HTC által nemrég forgalomba hozott okos telefon Android operációs rendszerrel. Egy ilyen telefon fejlesztők számára kiadott változatán történt a fejlesztés (*ADP*).



5. ábra: HTC G1 (*Dream*)

9. Fejlesztés Android környezetben

Android operációs rendszerre a fejlesztés történhet Java, vagy C++ nyelven. A Java nyelv használata esetén használható az alkalmazásfejlesztők számára készített eclipse-plugin, mely nagyban megkönnyíti a fejlesztést. A rendszer fejlesztői a rendszerbe integráltak egy saját maguk által készített Java virtuális gépet (*Dalvik*). Az sdk-hoz készített referencia –bár minden funkcióra kitér- sajnálatos módon igen szűkszavú, és sok esetben hiányos.

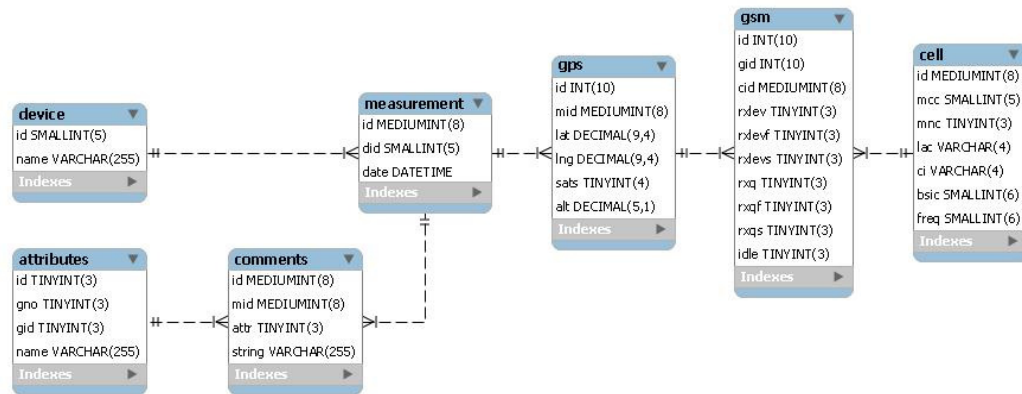
Az Android rendszer 1.6-os verzióján történt a fejlesztés, ebben a verzióban a következő cellainformációk voltak elérhetőek:

- A főcella esetében:
 - Cell ID (Cella azonosító)
 - LAC (Location Area Code)
 - MCC (Mobile Country Code)
 - MNC (Mobile Network Code)
- A mellékcellák esetében
 - Cell ID (Cella azonosító)
 - RSSI (Received Signal Strength Indicator)

10. A mért adatok tárolására használt adatbázis

Mint már írtam, saját helymeghatározó rendszerünk az előzetes mérésen, és a mérési eredmények adatbázisban történő eltárolásán alapszik. Az ehhez használt adatbázist Kelemen Mihály a korábbi években már elkészítette.

A központi tábla a measurement. Ez tárol egy mérési alkalmat (pl.: Ludovika tér, 2010. március 22. d.u.). Ehhez kapcsolódik a device tábla, ebben tároljuk el a mérőeszközt (Samba vagy Android). A méréshez lehet commenteket fűzni, mint például az időjárás, a helyszín, stb. (comments tábla). A gps táblában tárolunk el egy darab mérési pontot (egy időpillanat), ehhez kapcsolódnak a gsm sorok, amelyek leírják, hogy az adott pontban milyen toronyokat láttunk. a cell táblában van letárolva minden olyan torony amelyet láttunk a mérés során (helytakarékossági okokból).



6. ábra: Az adatbázis felépítése

11. A feladat megoldása

A feladatot elvégző programot, mint már említettem, Java nyelven írtam meg. Azzal kezdtem, hogy megpróbáltam kideríteni, hogy milyen osztályok nyújtják a szükséges szolgáltatásokat. Erre a fejlesztők számára kiadott referenciát használtam, ezen kívül több fórumon is találtam hasznos példaprogramokat.

GPS adatok

A GPS adatok kinyerésére az osztályunknak meg kell valósítania a `LocationListener` interfészt. Ehhez természetesen implementálnunk kell az interfész `GPS` pozícióváltozáskor hívott `callback` függvényét, jelen esetben az `onLocationChanged(Location arg0)` tagfüggvényt. Ennek a függvénynek egy `Location` típusú argumentuma van, melyből a `getLatitude()` és a `getLongitude()` függvényekkel kapjuk meg a szélességi, illetve hosszúsági fokokat.

Ahhoz viszont, hogy a pozíciófrissítéskor a függvény ténylegesen meg is hívódjon, regisztrálnunk kell az osztályunkat a `LocationManager` osztály `requestLocationUpdates` függvénye segítségével. Ez a következő paramétereket várja:

- `provider`: a szolgáltatás neve, amire regisztrálni szeretnénk (jelen esetben ez `LocationManager.GPS_PROVIDER`).
- `minTime`: két frissítés közötti eltelt minimum idő ezredmásodpercben.
- `minDistance`: két frissítés közötti minimum távolság méterben.
- `listener`: egy `LocationListener` objektum, melynek minden frissítéskor meghívódik az `onLocationChanged` függvénye.

A fenti dolgokat elvégezve azt értem el, hogy minden esetben, amikor a GPS pozíció változik, lefut a fő osztályom `onLocationChanged` függvénye, és frissíti saját a szélességi és hosszúsági fokokat tároló mezőit.

GSM adatok

A GSM adatok kinyerésével foglalkozó rész legfontosabb osztálya a `TelephonyManager`. Ahhoz hogy a `TelephonyManager` osztály egy példányát megkapjuk, amivel dolgozni fogunk, az alábbi dolgokat kell tennünk:

- kell szereznünk egy példányt a `Context` osztályból (a saját osztályunk megfelelő)
- ennek az objektumnak meg kell hívni a `getSystemService` tagfüggvényét `Context.TELEPHONY_SERVICE` paraméterrel
- az eredményt átkonvertáljuk `TelephonyManager` típusú objektummá

Ezek után feladat az MCC és az MNC kinyerése volt. Ehhez a `TelephonyManager` `getNetworkOperator` függvényét kell meghívni, amely egy `String` típusú objektummal tér vissza, melynek első 3 karaktere az MCC-t, a maradék pedig az MNC-t tartalmazza.

A főcella cellaazonosítójának, valamint a LAC kinyeréséhez először a `GsmCellLocation` osztály egy példányára van szükség. Ezt a `TelephonyManager` objektumunk `getCellLocation` függvényének hívásával érhetjük el. Ha ezzel megvagyunk, a `GsmCellLocation` példányunk `getCid`, illetve `getLac` függvénye visszaadja nekünk a szükséges adatokat.

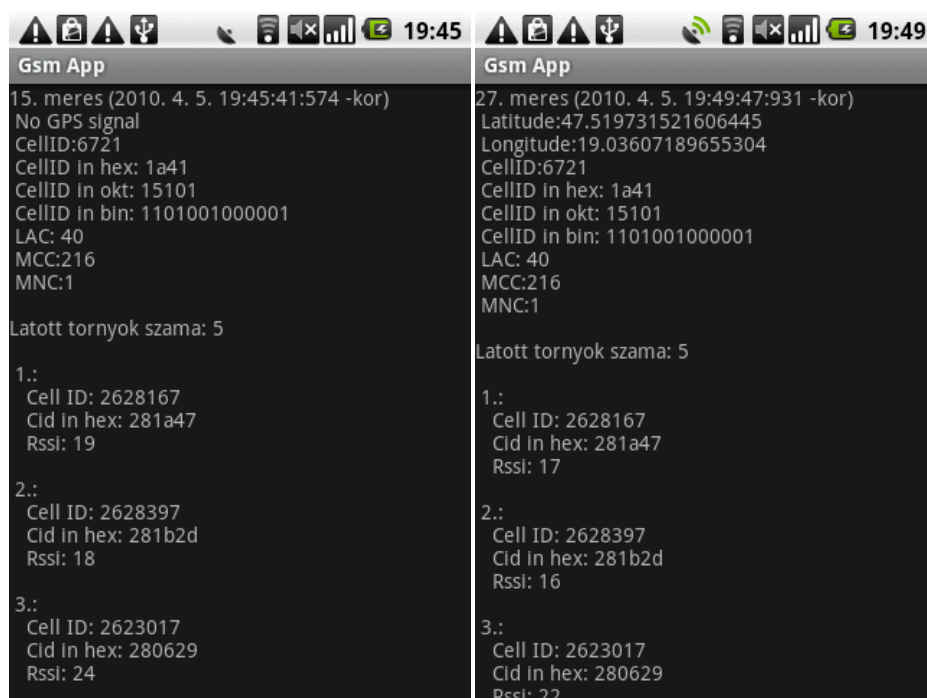
A környező cellák adatainak kinyeréséhez a következőt kell tenni:

- meg kell hívni a `TelephonyManager` objektumunk `getNeighboringCellInfo` függvényét, mely egy `NeighboringCellInfo` típusú objektumokat tartalmazó listát ad vissza.
- végigiterálunk a listán, és minden elemére alkalmazzuk a következőt:
 - a cellaazonosító kinyeréséhez meghívjuk a `getCid` függvényét
 - az RSSI megszerzéséhez meghívjuk a `getRssi` függvényét

12. A kész program

Ez a szoftver a telefonon futtatva másodpercenként lekérdezi az aktuális cellaadatokat (*Cell_ID*, *LAC*, *MCC*, *MNC*), valamint az elérhető információkat a környező cellákról (*Cell_ID*, *Rssi*). Ezekhez az adatokhoz hozzáteszi a mérés sorszámát, az aktuális időt, valamint a GPS adatokat, ha azok elérhetőek.

Az így kapott mérést megjeleníti a kijelzőn, valamint egy szöveges file-ban is tárolja a memóriakártyán. A cella azonosítóját a program kiírja hexadecimális, oktális és bináris formában is a korábbi mérésekkel való könnyebb összehasonlítás érdekében.



7. ábra: A kész program futás közben

Formázott: Címsor 1

13. Eltérések a Samba és az Android között

Az első és talán legszembetűnőbb különbség a rendelkezésre álló adatok számában van. Míg Samba esetén szinte az összes adat lekérhető volt, az Android ezzel szemben igen szűkszavú volt.

	Samba	Android
Cell ID	+	+
LAC	+	+
MCC	+	+
MNC	+	+
BSIC	+	-
Chanel (frequency)	+	-
RSSI	+	-
C1	+	-
C2	+	-

8. ábra: A Samba és az Android által a főcelláról szolgáltatott adatok

	Samba	Android
Cell ID	+	+
LAC	+	-
MCC	+	-
MNC	+	-
BSIC	+	-
Chanel (frequency)	+	-
RSSI	+	+
C1	+	-
C2	+	-

9. ábra: A Samba és az Android által a mellékcellákról szolgáltatott adatok

A következő eltérés az volt, hogy míg a Samba csak 2G-s hálózaton működik, az Android 2G-s és 3G-s hálózattal egyaránt tud dolgozni. Ez két okból is probléma volt:

- A Samba jelzett FFFF cellaazonosítóval rendelkező állomásokat. Ennek a jelenségnek az értelmezése még nyitott, de nagy valószínűséggel annak tulajdonítható, hogy nincs felkészítve a 3G-s hálózatok használatára
- Az Android sok esetben a cellaazonosító elejére egy E hexadecimális karaktert fűzött, az alább látható mérési pont esetében éppen a LAC-al egyenlő, ám ez lehet a pusztán véletlen műve is

Az alábbi táblázatban látható egy összehasonlítás a korábbi mérőállomás (Samba és külső USB –s GPS vevő), és az Android által egy időben készített mérési pontokról. Mint az jól látható, a főcelláról gyűjtött adatok (már amelyik lekérhető volt Androidon) megegyeznek.

A két eszköz eredményei közötti eltérés a mellékcellákban mutatkozik meg, ami viszont normális, tekintve, hogy két GSM eszköz nem feltétlenül látja ugyanazokat a tornyokat. Megfigyelhető továbbá, hogy a Samba két cella esetében is FFFF Cell ID – t adott, egynél pedig 0-t, így lehetséges, hogy azok az Android számára is látható cellák voltak. A táblázatban utolsóként szerelő cellát viszont mindketten látták, azzal a különbséggel, hogy mint már írtam, az Android a Cell ID elé fűzött egy E hexa karaktert.

A GPS vevők által szolgáltatott szélességi és hosszúsági fokok közti eltérés abból adódik, hogy míg a korábban használt eszköz fok – fokperc – fokmásodperc alakban dolgozik, az Android a fokot adja meg tizedes alakban. Ez különösebb problémát nem okozott, az adatbázisba való feltöltésnél kell figyelni rá, hogy az adatok a megfelelő formátumba legyenek hozva.

	Samba	Android
Latitude	4728.8534,N	47.48194456100464
Longitude	01905.0988,E	19.084598422050476
Cell ID	1272	1272
LAC	000E	000E
MCC	216	216
MNC	1	1
BSIC	17	??
Chanel (frequency)	53	??
RSSI	46	??
C1	41	??
C2	41	??
Cell ID	FFFF	e0245
LAC	000E	??
MCC	216	??
MNC	1	??
BSIC	14	??
Chanel (frequency)	792	??
RSSI	52	13
C1	32	??
C2	22	??
Cell ID	714	e255a
LAC	000E	??
MCC	216	??
MNC	1	??
BSIC	22	??
Chanel (frequency)	55	??
RSSI	39	24
C1	34	??
C2	34	??

	Samba	Android
Cell ID	FFFF	e1277
LAC	000E	??
MCC	216	??
MNC	1	??
BSIC	57	??
Chanel (frequency)	42	??
RSSI	33	28
C1	28	??
C2	28	??
Cell ID	0	e04f4
LAC	0	??
MCC	0	??
MNC	0	??
BSIC	0	??
Chanel (frequency)	0	??
RSSI	0	15
C1	??	??
C2	??	??
Cell ID	0c15	1273
LAC	000E	??
MCC	216	??
MNC	1	??
BSIC	77	??
Chanel (frequency)	56	??
RSSI	25	99
C1	20	??
C2	20	??
Cell ID	2559	e2559
LAC	000E	??
MCC	216	??
MNC	1	??
BSIC	57	??
Chanel (frequency)	42	??
RSSI	33	13
C1	28	??
C2	28	??

10. ábra: A Samba és az Android összehasonlítása egy mérési pont alapján

Megjegyzés [A4]: Mérési eredmények elhelyezhetők az adatbázisban, inden szükséges adatm megvan és összeilleszthető

Az Android által készített mérések

A feladat megoldása során készítettem egy programot, mellyel ellenőrizhető a mért adatok pozíciójának helyessége. Az alábbi ábrán ennek a megjelenítőnek egy pillanatképe látható.



11. ábra: A Ludovika téri mérés pontjai

Formázott: Normál

Megjegyzés [A5]: Továbbá egy kis Heuréka, működik, használható stb. legalább 2-3 oldal.
Jó lenne, ha a feladatkiírás minden egyes pontjára lenne válasz!

Mivel minden ehhez szükséges adat rendelkezésre áll Androidon, és a mérések néhány transzformáció után összeilleszthetők a korábbi eredményekkel, az Android által készített mérések adatbázisba helyezhetők. Az ehhez szükséges transzformációk:

- az Android által szolgáltatott GPS adatok átváltása fok – fokperc - fokmásodperc alakba
- A sats (szatelliteszám) oszlopba nem a látott szatellitek száma, hanem 1, vagy 0 kerül attól függően, hogy a GPS szolgáltatás elérhető-e
- Az alt (magasság) oszlopba mindig 0 kerül
- A GSM táblában az RXLEV oszlop után minden NULL

például:

```
id      gid cid rxlev rxlevf rxlevs rxq rxqf rxqs idle
294162  0 1935 16      NULL  NULL  NULL NULL NULL NULL
```

14. Klasszifikáció

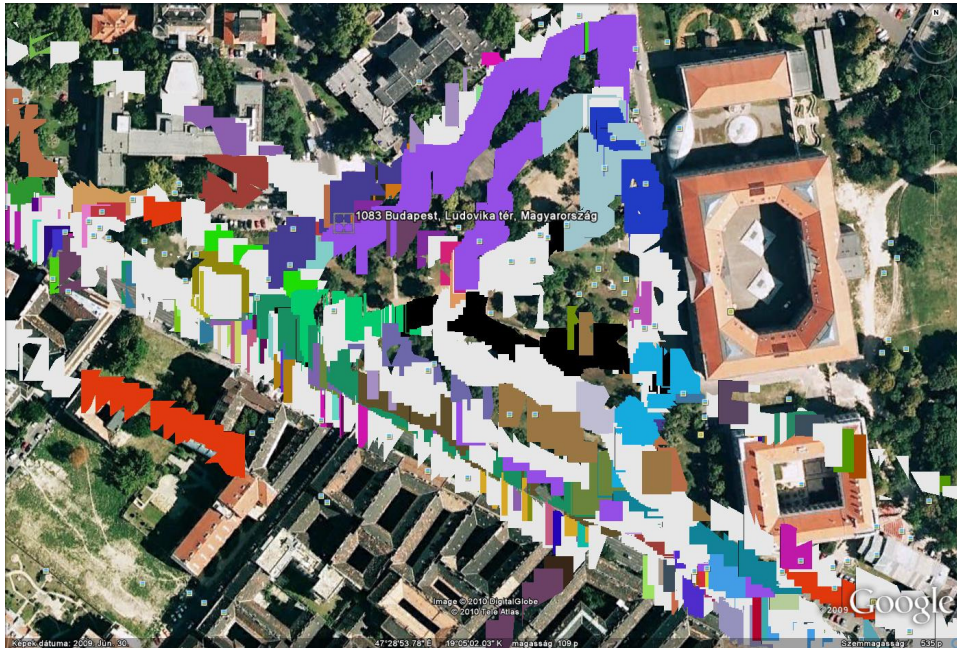
A következő feladat annak megbecslése volt, hogy az Androiddal készített mérések alapján történő helymeghatározástól milyen eredmények várhatók. Ennek becslésére egy Kelemen Mihány által készített klasszifikációs algoritmust használtunk, amely aszerint csoportosítja a mérési pontokat, hogy mely tornyok voltak láthatók.

Az algoritmus működése a következő:

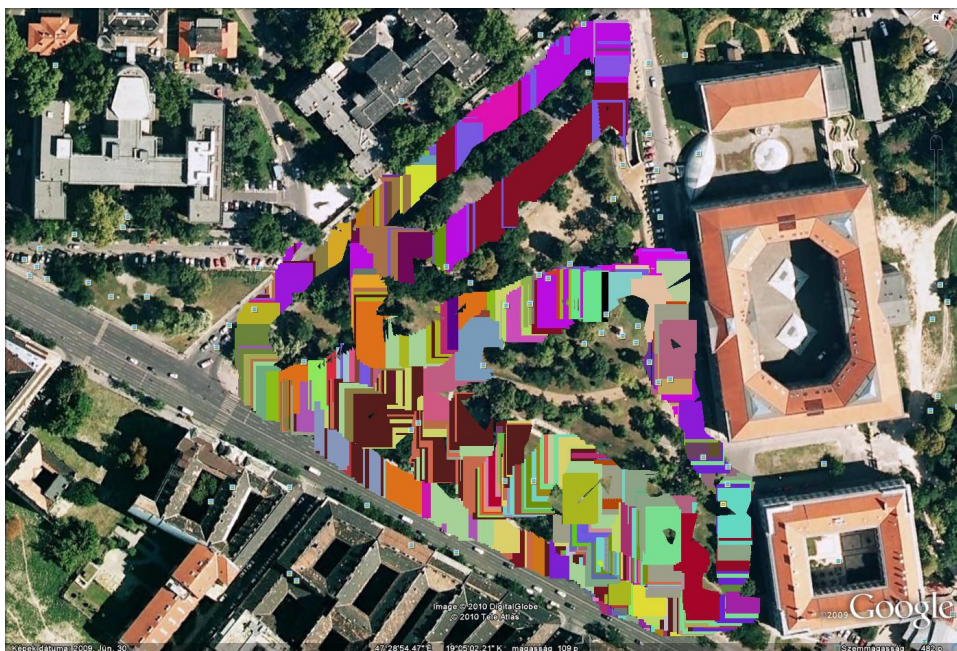
- A program elvégzi az adatbázis-lekérdezést, és eltárolja az eredményt.
- Minden mérésben rendezi a látott tornyokat cellaazonosító alapján.
- Ezután rendezi a mérési sorokat egymáshoz képest a látott tornyok cellaazonosítója alapján. Ekkor az azonos tornyokat látó mérési pontok egymás mellé kerültek.
- Ezután végigjárja a mérési pontokat, és olyan módon színezi, hogy az egy klaszterbe tartozó pontok (melyek mérésekor pontosan azonos tornyok voltak láthatók) azonos színűek legyenek.
- Az így készült pontokat kimentti egy .kml (Keyhole Markup Language) fájlba helyezi, mely alkalmas térbeli pontok tárolására, és megnyitható a Google Earth nevű programmal.

A mérési pontoknak ilyen módon történő osztályozása azért érdekes, mert képet ad arról, hogy jelenlegi állapotában a helymeghatározó algoritmustól milyen pontosság, és egyéb jellemzők várhatók.

Ez azért lehetséges, mert amikor a rendszert ténylegesen helymeghatározásra használjuk, akkor azok a területek, melyek különböző klaszterbe estek megkülönböztethető lesznek egymástól, ami azt jelenti, hogy a pontosság körülbelül akkorára becsülhető, amekkora az átlagos klaszterméret.



12. ábra: A Ludovika téri mérés klaszterizációja (Samba)



13. ábra: A Ludovika téri mérés klaszterizációja (Android)

15. Konklúzió

A feladat megoldása során arra jutottam, hogy az Android készülék alkalmas GSM alapú helymeghatározásra mind beltéren, mind kültéren, tekintve, hogy a rendelkezésre álló adatok száma ezt lehetővé teszi, valamint, hogy ezek az adatok összefüggésben vannak a telefon pozíciójával.

Ezen kívül a használt algoritmussal átlagosan 10 méteres klasztereket sikerült előállítani, ami azt jelenti, hogy ettől az algoritmustól körülbelül ilyen pontosság várható el. Ez elég jó pontosságot jelent, főleg, ha összehasonlítjuk a GPS eszközök méteres nagyságrendjével.

Megjegyzés [A6]: Nagyon kérem, hogy örüljön annak, hogy megoldotta a feladatot, és az elejére visszahivatkozva mutasson rá, hogy minden részfeladatot megcsinált.

16. Problémák, további feladatok

Mivel jelen dolgozatban én csak kültéri helymeghatározással foglalkoztam, azért a mérési pontok pozíciója adódott a GPS műholdak alapján. Viszont ahhoz, hogy a rendszer beltéren is alkalmazható legyen, találni kell egy módszert arra, hogy miként vihető be a pozíció a mérési adatbázisba GPS használata nélkül.

17. **Javaslatok**

A rendszer további fejlesztésére a következő javaslatokat tenném:

- Amint lehetséges, áttérés az Android 2.1-ás verziójára, amelyben jóval több információ lekérdezhető a környező cellákról (*hálózattípus, PCS, LAC, stb.*)
- Mérési pozíció bevitelére egy érintőképernyős beviteli lehetőség fejlesztése az alkalmazáshoz, amelynek egy megjelenített térképre, épületalaprajzra „*kattintva*” lehetne beadható a pozíció.
- Annak további vizsgálata, hogy többször megismételve ugyanazt a mérési sorozatot az eredmény mennyiben lesz különböző a két esetben.
- A klaszterizáló algoritmus továbbfejlesztése és pontosítása több cellainformáció figyelembe vételével

18. Függelék

Rövidítések jegyzéke

- GPS – Global Positioning System
- ADP – Android Developer Phone
- GSM – Global System for Mobile communications
- Cell ID – Cella azonosító
- MCC – Mobile Country Code
- MNC – Mobile Network Code
- LAC – Location Area Code
- BSIC – Base Station Identity Code
- RSSI – Received signal strength indication
- KML - Keyhole Markup Language

Felhasznált irodalom

Wikipedia (<http://en.wikipedia.org>)

Bányai Balázs, Feldhoffer Gergely, Tihanyi Attila - Helymeghatározás GSM hálózat felhasználásával 2008 08 híradástechnika

Takács György - Helymeghatározás mobiltelefonnal és mobil hálózattal (2008/08)

Google Android Reference Manual (1.5)

HTC honlapja (<http://www.htc.com>)

Falcom honlapja (<http://www.falcomusa.com>) Samba 75 használati utasítás és AT parancskészlet

OpenCellID (<http://www.opencellid.org/>)

Falcom Samba 75 specifikáció

Integrated Quad Band GSM/GPRS/EDGE Engine

SAMBA 75: 850/900/1800/1900 MHz

Data, SMS, Fax, MMS, Downloads

TCP/IP stack implemented

Audio/Video streaming over TCP/IP connection

GPRS Multi-Slot class 12

EDGE Multi-Slot class 10-B

Integrated USB-Hub

Internal GSM antenna

USB serial link

World-wide applicable

BVRP Connection Software

Windows 2000, XP, Linux, Mac OS

ADP specifikáció

Processor	Qualcomm® MSM7201A™, 528 MHz
Operating System	Android™
Memory	ROM: 256 MB RAM: 192 MB
Dimensions (LxWxT)	117.7 mm x 55.7 mm x 17.1 mm (4.60 in x 2.16 in x 0.62 in)
Weight	158 grams (5.60 ounces) with battery
Display	3.2-inch TFT-LCD flat touch-sensitive screen with 320 x 480 (HVGA) resolution
Network	HSPA/WCDMA: Europe: 2100 MHz US:1700/2100 MHz Up to 7.2 Mbps down-link (HSDPA) and 2 Mbps up-link (HSUPA) speeds Quad-band GSM/GPRS/EDGE: 850/900/1800/1900 MHz (Band frequency, HSUPA availability, and data speed are operator dependent.)
Device Control	Trackball with Enter button
Keyboard	Slide-out 5-row QWERTY keyboard

GPS	GPS navigation capability with Google Maps™
Connectivity	Bluetooth® 2.0 with Enhanced Data Rate Wi-Fi®: IEEE 802.11b/g HTC ExtUSB™ (11-pin mini-USB 2.0 and audio jack in one)
Camera	3.2 megapixel color camera with auto focus
Audio	Built-in microphone and speaker Ring tone formats: AAC, AAC+, AMR-NB, MIDI, MP3, WMA, WMV 40 polyphonic and standard MIDI format 0 and 1 (SMF)/SP MIDI
Battery	Rechargeable Lithium-ion battery Capacity: 1150 mAh Talk time: Up to 350 minutes for WCDMA Up to 406 minutes for GSM Standby time: Up to 402 hours for WCDMA Up to 319 hours for GSM (The above are subject to network and phone usage.)
Expansion Slot	microSD™ memory card (SD 2.0 compatible)
AC Adapter	Voltage range/frequency: 100 ~ 240V AC, 50/60 Hz DC output: 5V and 1A
Special Features	Digital Compass, Motion Sensor