



Pázmány Péter Katolikus Egyetem Információs Technológiai Kar

GSM alapú helymeghatározás

Nagy mennyiségű adat rögzítése, tárolása, valamint megjelenítése GSM alapú helymeghatározáshoz

Készítette: Kelemen Mihály

Konzulens: Dr. Takács György, Tihanyi Attila

PPKE-ITK
2009.

Tartalomjegyzék

Bevezetés	3
Jelenleg használt mérőeszközök és sajátosságai	3
Korábbi „notebook mérés”	3
Helyhez kötött hosszú mérés	4
Robot alkalmazása beltéri mérésekhez	4
Mobil mérőeszköz autós és elemes kivitelben.....	5
Adatgyűjtő firmware újratervezése, megújult funkciói	6
Felújított, kibővített firmware	6
Bluetooth modul.....	7
Szoftvermódosítások	7
Xmodem1k protokoll	8
Rövid összefoglaló.....	9
Grafikus megjelenítés	10
Kültéri mérés	10
Robot mérés	11
Adatok tárolása központi szerveren.....	12
Felmerülő problémák	12
Megoldás	12
Intelligens feltöltő felület.....	13
Rövid összefoglaló.....	14
További lehetőségek	15
Összefoglaló.....	15
Távlatok a csapat későbbi munkájában	15
Felhasznált irodalom.....	16

Bevezetés

A tavaly elkészített adatgyűjtő eszköz hasznosnak bizonyult a kültéri mérések elvégzésében, valamint új ötletekkel, és ambícióval ellátva a GSM alapú helymeghatározás témakörében folytattam munkámat.

Félév elején elsősorban cellainformációk gyűjtésére volt a témával foglalkozó csapatnak szüksége, majd ez az igény átalakult, és a figyelem egyre inkább az adatok tömör, mégis veszteségmentes rögzítésére, később grafikus megjelenítésére irányult. A robotika labor is jelezte érdeklődését, és egy kölcsönös együttműködés kezdődött, amely reményteljes eredményekkel kecsegtet mindkét fél számára.

A csapat igényeihez igazodva a feladatom először új adatgyűjtő eszközök készítése volt, az Önálló laboratórium I. tárgyban végzett munkámra alapozva, utána a mért adatsorok tárolásának egyszerűsítése, és átláthatóvá tétele, valamint megjelenítése.

Jelenleg használt mérőeszközök és sajátosságaik

Idén több mérőeszközt is használtunk. Ezek között a legfontosabb különbség a felvett adatsorokban mutatkozik! Korábbi mérések során ill. a robot mérésnél például tárolunk C1 és C2 adatokat, amikből elméletben a bázisállomás választás algoritmusá számolható, míg a mobil mérőeszközök ezeket a paramétereket nem közli velünk, több információt közöl viszont az aktuális celláról melyenek elsősorban hívás közben van jelentősége. A másik leglényegesebb különbség, a rögzített sorok hossza. Az utóbb említett mérések során változó hosszúságú sorok keletkeznek: Az aktív cella 13 paramétere, továbbá vett jelerősség szerint csökkenő sorban maximum 6 észlelhető szomszéd cella, egyenként 7 paraméterrel. Összehasonlításképp a „notebookmérés” során rögzített adatsorban minden esetben 7 cellára vonatkozó információt rögzítünk, viszont amikor nem vehető ennyi cella jele akkor ezen cellainformációk között néhány helyén csak nullák találhatók.

Korábbi „notebook mérés”

Legelső méréseinket a SAMBA75 GSM/GPRS modem segítségével rögzítettük egy notebook segítségével. Helyileg rengeteg helyen használtuk az eszközt: A laborban egy helyen történő hosszú mérésre, az iskola épületében a 4. emeleti folyosón, az iskola mögötti parkolóban, de mértünk vele a Margit-szigeten, vagy akár az Árpád utca környékén.

Formailag minden sor egy adott másodpercben mért GPGBA helyzetinformáció, és utána 7 bázisállomás 9-9 vételi paramétere, végül a pontos idő, és egyéb információk. Ha egy pontban nem fogható 7 bázisállomás jele, például egy ház árnyékában kiesik pár, akkor ott csupa nulla paraméterek vehetők. Egy tipikus rögzített adatsor hasonlóképpen néz ki:

```
$GPGBA,194651.000,4735.2415,N,01902.9096,E,1,07,1.1,107.2,M,41.2,M,,0000*5D,216,30,0032,1432,00,742,22,12,28,216,30,0032,143E,15,111,26,22,22,216,30,0032,1420,37,114,25,21,21,216,30,0032,13A6,14,110,24,20,20,216,30,0032,13CE,20,118,15,11,11,000,000,0000,0,0000,00,0,0,-,-,216,30,0032,141F,31,109,15,11,11,21:47:04,742 19 -91 216 30 0032 1432 0 0 30 -101 9 I No connection
```

Helyhez kötött hosszú mérés

Az előző méréshez nagyon hasonló a helyhez kötött hosszú mérés, de jellemző rá néhány sajátosság. A helyhez kötött hosszú mérést a laborban végeztük, egy ponton lehelyezett SAMBA75 modem hetekig rögzítette a cellainformációkat ugyanazon a ponton, hogy később megállapításokat tehessünk a hálózat helyzetfüggetlen viselkedéseiről. Tipikusan ezek az adatsorok nagy méretű állományokat képeznek, és nincs benne GPS koordináta, mivel beltéri mérés lévén felesleges lett volna azt is mérni. Az egyhelyben végzett mérés sorai a következőképpen alakultak:

```
$GPGBA,,,,,,,,,1E,23C5,24,116,43,39,39,216,30,001E,2B21,10,741,33,24,32,216,30,001E,0EFC,57,115,30,26,26,000,000,0000,0000,00,0,0,-,-,216,30,001E,FFFF,13,112,22,18,18,216,30,001E,23D8,55,110,14,10,10,000,000,0000,0000,00,0,0,-,-,15:19:32:881
```

Az első cella adatai előtti karakterek az előző fejezetben tárgyalt adattípussal való kompatibilitás szempontjából szerepelnek a sorban.

Robot alkalmazása beltéri mérésekhez

A félév második felében a robotika laborral karöltve elkezdünk azon munkálkodni, hogy beltéri méréseinket egy robottal végeztethessük. A módszer legnagyobb előnyének számítana, hogy nem igényel emberi aktivitást, valamint a mért cellainformációk helyéről is tud relatív pozíciót szolgáltatni. Mérés közben továbbá automatikusan térképet is csinál a környezetéről ugyanebben a relatív koordinátarendszerben, így a mérés során bejárt útvonal és a tereptárgyakhoz képesti viszonya később grafikusán ábrázolható, nyomon követhető. Egy robottal rögzített log file:

7 1242126327: 216 30 001E 2B24 12 729 13 3 17 1242126327: 216 30 001E 23C7 13 112
16 12 12 1242126327: 216 30 001E 0F1B 55 739 14 10 10 1242126327: 216 30 001E 0F1A
56 735 12 8 8 1242126327: 216 30 001E FFFF 10 741 11 1 9 1242126327: 216 30 001E
23C5 24 116 9 5 5 1242126327: 216 30 001E 2B74 14 738 7 -3 13

0 0 0

Látható, hogy a cellainformációk, és a robot pozíciója (első két paraméter a robot relatív pozíciójának koordinátái, harmadik pedig a jelenlegi irány a kezdeti irányához képesti szögelfordulásban) külön sorban szerepel.

A robot pozícióját x , y koordinátával, és θ szögelfordulással adja meg. Mérés kezdetekor a (0,0) koordinátában áll. Minden pontban mért helyzetét, ehhez a ponthoz méri egy relatív koordináta rendszerben. Térkép készítéséhez szükséges minden pillanatban a pontos irány ismerete, amit induláshoz képesti szögelfordulásban ad meg.

Mobil mérőeszköz autós és elemes kivitelben

Az előző félévben tervezett, és elkészített adatgyűjtő az eddig tárgyalt eszközökhöz képest eltérő GSM modemmel (Wavecom Q2687) lett felszerelve. Ennek két következménye van a rögzített adatokra nézve:

1. Mobil mérőeszközünkkel rögzített cellainformációinkban nem szerepel a C1 valamint C2 paraméter, viszont a főcelláról új többletinformációkat közöl: Az RxLevFull, RxLevSub, RxQual, RxQualFull, RxQualSub, Idle TS paramétereket, melyek elsősorban hívás közben kapnak szerepet.
2. A szomszédos cellák RxLevel szerinti csökkenő sorrendben követik egymást, ha nem látható 6 szomszédos cella (például vidéken), akkor azok kinullázása helyett rövidebb sort kapunk.

\$GPGGA,080150.000,4713.4240,N,02044.4692,E,1,06,2.4,90.1,M,40.5,M,,0000*61,216,7
0,0017,a02b,18,526,32,,,0,,,0,216,70,9d76,,20,2,1,216,70,9d81,a176,18,17,12

Ebben az adatsorban például csak 3 cella jele volt vehető, a többi helyén nincsenek felesleges nullák, mint a SAMBA esetén!

Adatgyűjtő firmware újratervezése, megújult funkciói

A félév elején több mobil adatgyűjtő eszköz összeállítására került sor, hogy minél több mérés rögzítésére adjunk lehetőséget. Az eszközök dobozolása viszont új problémát vetett fel: A memóriakártya hozzáférhetőségének kérdését, ugyanis az csak a doboz ismételt szétszedésével vehető ki az adatgyűjtőből. Ezen túl számos funkcionális hiányosság is sürgette az eszköz módosítását.

Felújított, kibővített firmware

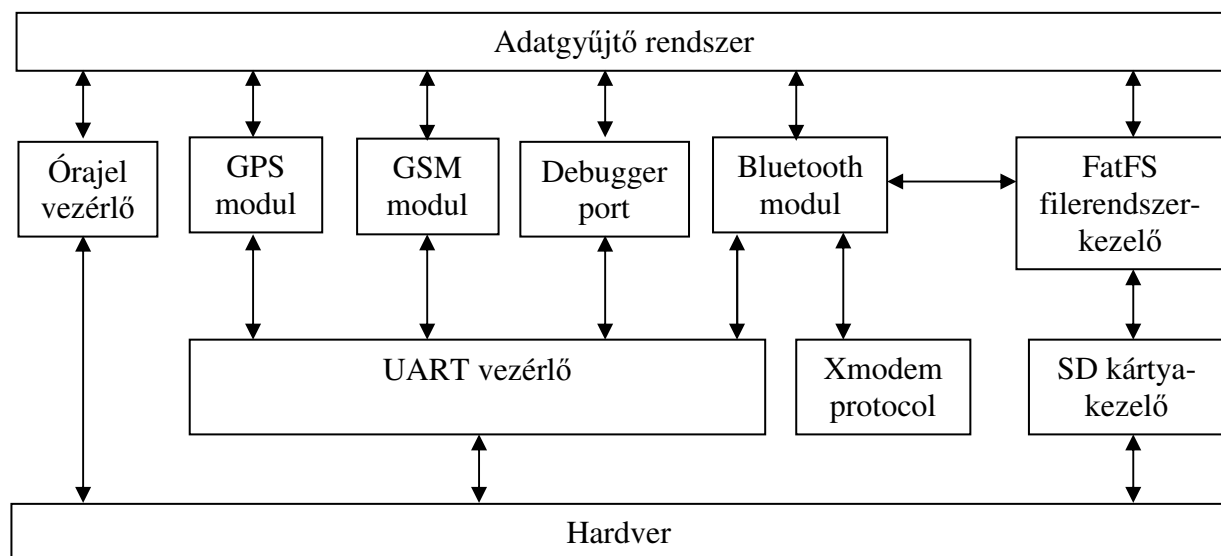
Az adatgyűjtő szoftverétől idővel egyre több funkciót vártunk a kezdeti elképzelésekhez képest. Ilyen szolgáltatás például a dátum és idő kezelés, mivel kényelmesebb az eszközzel tároltatni a pontos időt minden mérés elején, mint ezt papíron dokumentálni minden egyes alkalommal.

A firmware bővítése új funkciókkal a kód méretének nagyságából kifolyólag egyre nehezebbé vált. Egyes fejlesztések majdhogynem az egész szoftver átírását igényelték, és minden esetben irreálisan sok ráfordított időt igényelt ahhoz képest, hogy egy attól alig különböző funkció már volt a rendszerben.

Az adatok átvitele számítógépre csakugyan problémás, mivel minden esetben az eszköz teljes szétszerelését igényelte, hogy a kártyához hozzáférhessünk. Az adatátvitel megkönnyítésére a vezeték nélküli kapcsolatot választottam, amit egy Bluetooth modul segítségével valósítottam meg. Az új modul beépítése, illetve az adatgyűjtő és az adatok továbbításáért felelős üzemmódok egyidejű kezelése is sürgette a firmware újratervezését.

A cél az volt, hogy a hálózati progradelemeket sikerüljön minél jobban elválasztani a program törzsétől. Ehez több egymásra épülő interfészt valósítottam meg, melyek közül a hardverspecifikus elemek helyezkednek el legalul. Ezek csupán az eszközök közötti kommunikációért felelősek. Felettük helyezkednek el azok a függvények, amik az adott modulok üzeneteit értelmezni tudják, és az adatgyűjtő rendszer számára érthető nyelven továbbítja ezeket.

A megvalósított interfészek viszonyát leginkább egy blokkdiagramon lehet megmutatni:



1. ábra: Adatgyűjtő rendszer blokkdiagramja

Az új firmware segítségével egyszerűbbé vált új hardvereszközök integrálása a meglévő rendszerbe, valamint szükség esetén lehetőséget biztosít egy esetleges későbbi platform-váltáshoz.

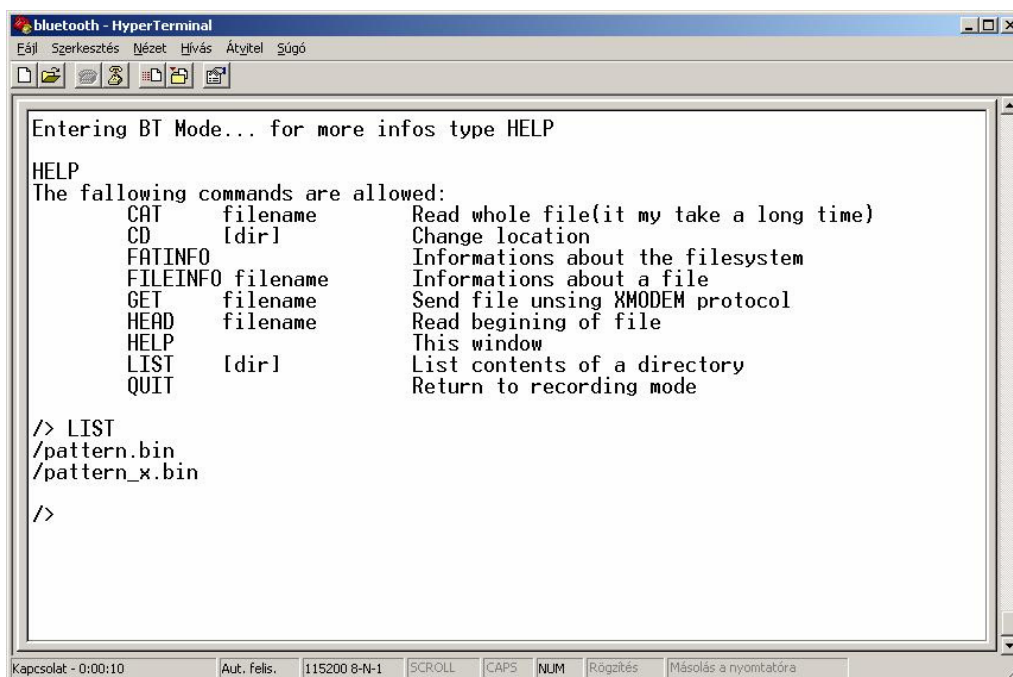
Bluetooth modul

Mint azt már korábban említettem, az adatok kinyeréséhez vezeték nélküli adatátviteli módot választottuk, hogy ne kelljen mindig kiszedni az eszközt a dobozából. Az adatátvitelért jelen esetben a Bluegiga WT32 Bluetooth Audio Module a felelős, ami az adott hardverkörnyezetben maximálisan 1.8Mbit/s adatátviteli sebességre képes soros porton keresztül. A modul kapcsolatfelvételhez Serial Port Profile(SPP) és Hands-free Audio Profile(HFP) módot támogat maximálisan 2.4Mbit/s adatátviteli sebességgel. A tervezett adatátviteli sebesség kinyeréséhez a mikrokontroller 32 Mhz-es órajelének finomhangolása szükséges, hogy a soros port sebessége a maximális hibaküszöbön belül maradjon.

Szoftvermódosítások

Soros adatátvitel segítségével Hyperterminal-ban Xmodem protokollon keresztül anélkül küldhetünk át adatokat, hogy saját megvalósított fájl fogadó alkalmazást írtunk volna a számítógépre. A soros port másik előnye, hogy nem csak fájlokat küldhetünk és fogadhatunk rajta, hanem a Hyperterminál segítségével szöveges karaktereket is küldhetünk. Ezt kihasználva egy parancsértelmezőt is implementáltam ugyanazon a kommunikációs csatornán, amely segítségével egyszerűen megtudhatjuk milyen file-ok vannak az adattárolón,

vagy akár bele is olvashatunk egy adott file tartalmába, mielőtt letöltenénk azt számítógépünkre. A megvalósított parancsok a CAT, CD, FATINFO, FILEINFO, GET, HEAD, HELP, LIST, QUIT



2. ábra: Soros porti parancsértelmező

Xmodem1k protokoll

Soros porton a fájlküldés egyik legegyszerűbb, és leggyorsabb módja az Xmodem protokoll használata. Tudni kell róla, hogy csak pont-pont összeköttetés esetén alkalmazható, mivel nincs benne címzett, de jelen esetben ez nem jelent problémát. Az Xmodem 128 bájtos adatcsomagokat tud egyszerre küldeni, amihez hozzáadódik még 4 bájt payload, ami a csomagok szinkronizálásért, és hibaellenőrzésért felelősek. Az adatátvitel minden esetben a fogadó fél NAK (0x21) parancsával kezdődik, ezt követően küldi el a másik fél az első 132 bájt adatot. A küldött adatformátum a következőképpen nézett ki:

1. SOH: A küldendő blokk elejét jelzi, mérete 1 bájt
2. A küldendő bájt sorszámát jelzi moduló osztva 256-al, mérete 1 bájt
3. Az előző bájt bitenkénti negáltja.
4. A küldendő 128 bájtos adatcsomag
5. Ellenőrző bájt: Az adatcsomag bájtjainak összege moduló osztva 256-al

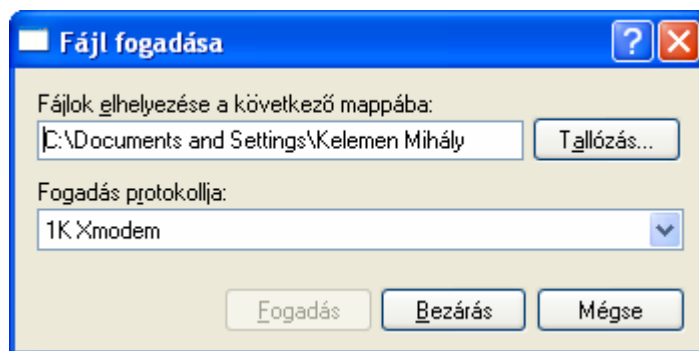
Minden csomag elküldése után egy bájtos válasz érkezik, az adatátvitel sikerességétől függően lehet ACK(sikeres adatátvitel), NAK(sikertelen adatátvitel), valamint CAN(másik fél

megszakította az adatátvitelt). Abban az esetben, ha nem érkezik válasz, a küldőnek újra kell küldenie az aktuális csomagot.

A fájl végét minden esetben egy EOT(0x04) karakter jelzi. Erre a parancsra ugyanúgy sikeres választ vár a küldő, hogy tudja a másik fél értesült az adatátvitel végéről.

Későbbiekben módosították a protokollt: Lehetőség nyílt 1024 bájtos csomagok egyidejű továbbítására, valamint az egyszerű hibaszámítási algoritmust leváltotta egy 16 bites CRC algoritmus, ami sokkal megnövelte az adatátvitel megbízhatóságát. A Hyperterminal Xmodem1k protokollja ezeket a módosításokat is támogatja, ezért implementáltam a protokoll átdolgozott verzióján fájlok küldését a számítógépre.

Fájlok áttöltése számítógépre nagyon egyszerű, egyszerűen csak csatlakozni kell a mérőeszközhöz Bluetooth-on keresztül, aztán a kijelölt COM porton felvenni a kapcsolatot vele a Hyperterminál segítségével (1843200 bps, 8 adatbit, nincs paritás, 1 stopbit, soremelés+kocsivissza karakterek küldése). Ezt követően a megvalósított parancsértelmező segítségével ki kell választani a megfelelő fájlt, és a GET <fájlnév> parancs kiadását követően a rendszer várja, hogy a fogadó fél felkészüljön a fájl fogadására. Ehhez az „Átvitel” menü „Fájl fogadása” pontjában kell kiválasztani az „1K Xmodem” protokollt, és a kívánt fájlnevet, és már indul is a fájlátvitel.



3. ábra: Fájl fogadása Hyperterminál segítségével

Rövid összefoglaló

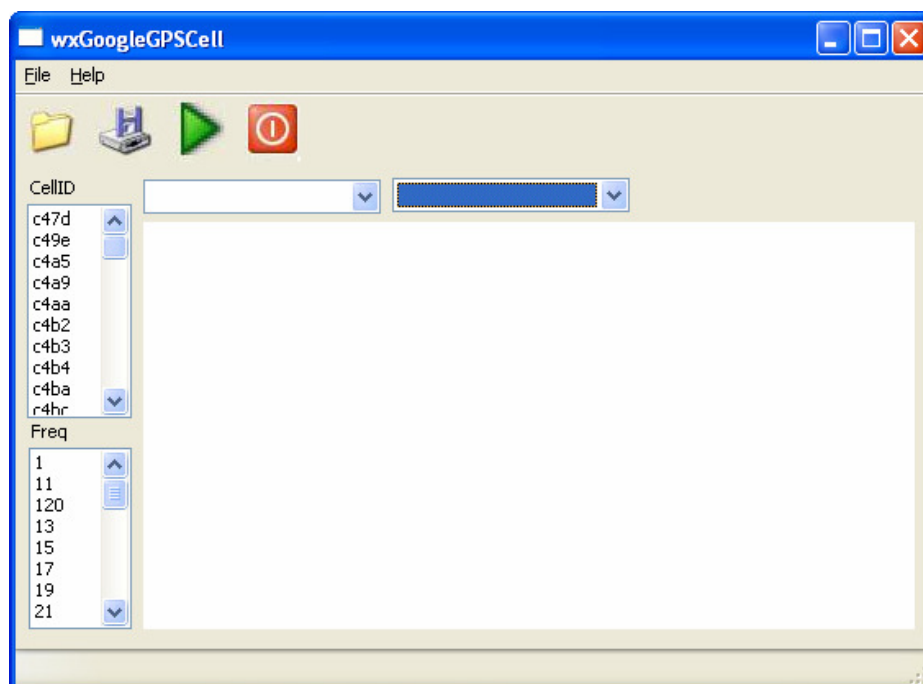
Elkészült az adatgyűjtő legújabb verziója, amelyik az eddigieknél sokkal kényelmesebb mérést tesz lehetővé. Rengeteg mérést rögzítésére adott lehetőséget, amelyek fontos információkkal látják el a csoport további munkáját. Igény esetén új funkciók implementálása lényegesen rövidebb időt vesz igénybe, mint eszközünk korábbi verziójánál.

Grafikus megjelenítés

A mérések megjelenítése is szükséges különböző megfigyelések szemléltetéséhez, például a frekvenciák többszöri kihasználása egy adott területen belül, vagy akár a vett jelerősség változása egy útvonalon. Ezek vizuális megjelenítésére különböző megjelenítő eszközök készültek, amelyeket elsősorban az különböztet meg, hogy a mérést kültérben, vagy beltérben készítettük. Kültéri méréseknél ugyanis a Google térképei jól felhasználhatóak a mérés helyének megjelenítésére, míg beltérben ezek a térképek nem használhatóak, mivel nem tartalmaznak az épületekről alaprajzot, valamint legtöbbször nincs is pozíció-adatunk.

Kültéri mérés

Kültéri mérések megjelenítésére a Google Maps API-t használtam fel. Előnye, hogy könnyen kezelhető, és a térkép-kezeléssel (nagyítás, mozgatás) nem kell foglalkozni, elég csak a megfelelő színű pontokat elhelyezni a térképen. Pontok elhelyezése JavaScript nyelven történik. Az általam írt program a wxWidgets library segítségével lett megírva C++ nyelven. Egy kiválasztott bemeneti fájl alapján készít html kimenetet, ami a Google Maps API szerint helyes térképet jelenít meg. A program jelenleg kizárólag a „Mobil mérőeszközök” által felvett méréseket támogatja, mivel ezeket nem tudtuk eddig megjeleníteni. Program tesztelésekor kiderült, hogy az általunk választott API sok megjelenítendő pontot lassan kezel, így a program további fejlesztése nem ajánlott.



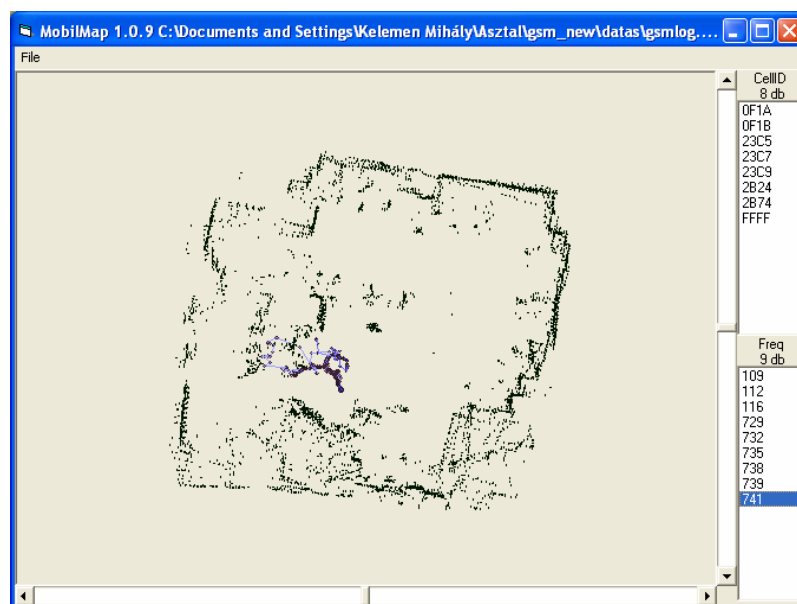
4. ábra: Térkép generálása Google Maps API számára



5. ábra: Mobil mérés megjelenítése Google Maps API segítségével

Robot mérés

Reményeink szerint a beltéri méréseinket robot fogja végezni. Ennek az előnye, hogy a robotunk beltérben is tud pozíció adatokat tárolni, ami alapján a pozíciója ábrázolható az általa készített térképen. Ezekhez az új adatokhoz szintén nem létezett megjelenítő program eddig. Egy régebbi Visual Basic-ben készült megjelenítőt fejlesztettem tovább, ami kültéri „notebook” mérések megjelenítésére volt alkalmas térkép nélkül. A megjelenítő képes kirajzolni egy „notebook” mérés során megtett útvonalat, vagy egyszerre egy robot mérés során megtett útvonalat egy másik robot útvonallal és egy általa készített beltéri térképpel. A másik robot útvonal a Robotika labor számára érdekes, mivel így térképen ábrázolható egy robot javított útvonala a nyers adatokhoz képes.



6. ábra: Robot mérés megjelenítése

Adatok tárolása központi szerveren

A mérések rögzítése minden esetben fájlalba történik. A helymeghatározással foglalkozó csapat minden tagjának szüksége van ezen mérésekre valamilyen formában. Vannak akik statisztikákat próbálnak kinyerni az összesített adatsorból, olyanok is akik helymeghatározó algoritmusokat próbálnak ki és az általuk detektált koordinátát hasonlítják a GPS koordinátáihoz. Az adatsorok megosztása a csoport tagjai között eleinte FTP serveren történt, minden mérés egy „TXT” fájlban volt, melynek a neve hordozott információt a mérés helyét illetően, ám ez a mérések számának növekedésével egy új, informatívabb tárolási módszer kidolgozását sürgette.

Felmerülő problémák

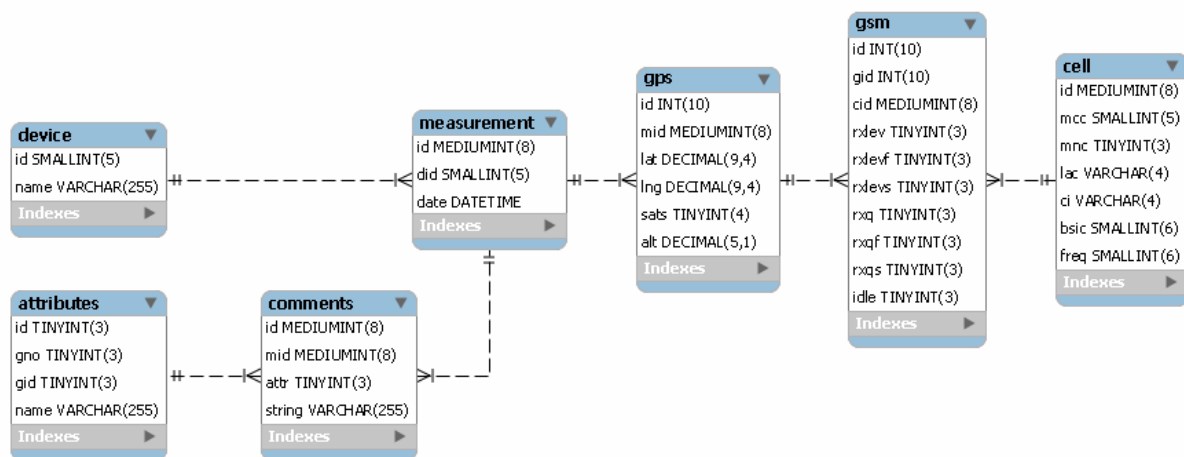
Az FTP-n tárolt fájlok elszaporodásával kiütköztek a módszer problémái. Az elsődleges probléma az volt, hogy egy régebben felvett mérésről idővel elfelejti az ember, hogy mikor, hol, és milyen környezetben készül. A felvétel dátumát tartalmazhatja a fájlnev ugyan, de ez nem minden mérésünknél szerepel, rendszerint csak a mérés helye van meg, és az is csak hozzávetőleges.

Fájljaink nem hordoznak továbbá információt a felvétel napszakáról, az aktuális időjárásról, és arról sem, hogy melyik adatgyűjtő eszközzel készült a felvétel, pedig ez meghatározza a tárolt adatsorok formátumát. Ebben az esetben, hogy egy mérés kültéri, vagy beltéri, csak a fájl beolvasásával dönthető el, az ugyanis csak kültéri mérésnél kezdődik NMEA üzenettel.

A mérések feldolgozását is nehezíti, hogy adatainkat fájlokban tároljuk, így ugyanis minden feldolgozó alkalmazás készítésekor ismernünk kell az adatsor pontos szerkezetét, legrosszabb esetben egy programot annyi verzióban kell megírni, amennyi adatformátumunk van. Több mérésről együttes statisztikák készítése is gondot okoz, ez ugyanis csak a fájlok összemácsolásával lehetséges, ami nagy fájl méretet, és nagy feldolgozási időt eredményez.

Megoldás

A felmerült problémák kiküszöbölésére az adatokat SQL adatbázisban tároljuk, aminek köszönhetően méréseink gyorsan, és egyszerűen hozzáférhetőek. Az adatok több, egymással kapcsolatban álló táblába kerülnek, ezzel is csökkentve az adatbázis helyfoglalását, és növelve a sebességét.



7. ábra: A megvalósított adatbázis szerkezeti felépítése

A mérésekkel kapcsolatos információk, mint például a helyszín, a napszak, vagy az időjárás egy külön táblában kaptak helyet, így korlátlan számú információt rögzíthetünk méréseinkről, és számuk bármelyik pillanatban zavartalanul növelhető. Minden mérési információ egy bizonyos kategóriát alkot (például időjárás), amelyen belül az adott mérés típusa kiválasztható (napos/esős/szeles). A kategóriák és a kategórián belüli típusok száma is tetszőleges, utólag megváltoztatható. Minden információhoz megadható továbbá egy szöveges kiegészítés, amivel az előre megadott besoroláson túli közlendőnket rögzíthetjük.

Az adatbázis tervezésével elértük, hogy a különböző formátumú adatsoraink ugyanabban a táblában kapjanak helyet maximális kompatibilitást garantálva, mégis eldönthető, hogy egy mérés melyik eszközzel lett rögzítve.

Táblázat változása az adatgyűjtő típusától függően:

	id	gid	cid	rxlev	rxlevf	rxlevs	rxq	rxqf	rxqs	idle
Régi „notebook” mérés	31	1	6	35	25	33	0	0	0	0
Mobil autós mérés (aktív cella)	2117	594	66	23	0	0	0	0	0	0
Mobil mérés hívás közben (aktív cella)	39	1	1	0	19	85	0	31	32	0
Mobil mérés (szomszédos cella)	2118	294	65	9	NULL	NULL	NULL	NULL	NULL	NULL

Intelligens feltöltő felület

Mérés során az adatokat fájlokban tároljuk, így nem szükséges aktív kapcsolat a szerver és az eszköz között, elég csak később feltölteni a gyűjtött adatokat a szerverre. A mérőműszerek

adatsora és a szerveren tárolt adatbázis eltérő, így szükség van egy feltöltő programra, ami biztosítja a konverziót a két szerkezet között.

A feltöltő programot PHP nyelven valósítottam meg, mivel HTML formon keresztül egyszerűvé teszi a mérés tulajdonságainak beállítását, és egyszerűen kezeli a MySQL adatbázisokat. Az „Upload” gombra kattintva a rendszer automatikusan feltölti a fájlt a szerverre, és sikeres feltöltés esetén elkezdi feltölteni az adatokat a központi adatbázisba. A program önállóan képes megkülönböztetni a különféle adatformátumokat. Minden eddigi adatformátumot gond nélkül kezel, és az adatok felvételét követően már egymással kompatibilisnek tekinthetők, így a gyakorlatilag nincs szükség az adatformátumok további megkülönböztetésére.

GSM upload

Measurement

Device

Date

Attributes

Data file:

8. ábra: Az intelligens feltöltő felület

Rövid összefoglaló

Minden jelenleg használt mérőeszköz által rögzített adatsor feltölthető egy központi adattárba, ami gyorsabb elérést, nagyobb megbízhatóságot. Az egységes adatszerkezet megkönnyíti a mérések kezelését, továbbá több mérést felhasználó statisztikák is zökkenőmentesen készíthetők. Nem csak MySQL adatbázis használható az adatok tárolására. A feltöltő program támogatja más ípusú adatbázis kezelését is, például Postgis, vagy Oracle.

További lehetőségek

Készülnek már az új adatbázist felhasználó megjelenítő eszközök. Előnyük, hogy a mérés típusától függetlenül képesek megjelenítésre. A későbbiekben ajánlott inkább azokra koncentrálni. A robot mérés megjelenítése egyelőre nem megoldható csupán adatbázis használatával, mivel a robot által készített térkép feltöltése egyelőre nem támogatott.

Összefoglaló

A félév során több adatgyűjtőt készítettem el, így lehetővé téve nagy mennyiségű adathalmaz felvételét. A mérőeszközöket új, és hasznos funkciókkal sikerült kiegészíteni, melyek lehetővé tették az adatok gyors áttöltését a PC rendszerbe valamint azok ellenőrzését az adatbázisba kerülésük előtt. Ez utóbbi funkció azt a lehetőséget is megadja, hogy régi már elfelejtett tartalmú adatállományokat könnyen, gyorsan tudunk azonosítani, és a tartamukat ismételten használhatóvá tenni. Egy reményekkel teli kezdeményezés, hogy a robotika labor egyik robotja automatikus méréseket végezzen az épületben éjszakánként. A félévben készített adatfeltöltő programomnak köszönhetően a mérések most már áttekinthetőbben tárolhatóak, és a kezelésük is kevesebb problémát vet fel, mint korábban. A nagy mennyiségű adatsor egy helyen tárolása növeli a nagy tanulóhalmazból dolgozó helymeghatározó algoritmusok hatékonyságát.

A félév végére elértük, hogy az összes eddigi mérés megjeleníthetővé váljon, és számos API-t, illetve néhány saját megoldást is kipróbáltunk az ábrázolás módjára. Kiderült, hogy a Google Maps API használata nem járható út, ellenben a Google Earth megbízhatónak bizonyult, amennyiben nem okoz problémát annak telepítése.

Távlatok a csapat későbbi munkájában

Tapasztalataink szerint a továbbiakban elsősorban az adatbázisból szabad dolgoznunk, mivel az elérése sokkal egyszerűbb a korábbi megoldástól, valamint a rendszert is kevésbé terheli. Méréseink számát elsősorban a robot mérés tovább fejlesztésével fokozhatjuk, mivel az nem igényel majd emberi felügyeletet mérés közben. A kültéri mérések tekintetében GSM alapú helymeghatározással elsősorban a magas házak közelében érhetünk el lényegesen jobb pontosságot a GPS alapú módszerekhez képest, mivel itt a GPS nagyon pontatlan, valamint az egy helyen vehető cellák száma sok, ezért célszerű a továbbiakban ezekre koncentrálni.

Felhasznált irodalom

- Wavecom Q2687:
http://www.kern.hu/KERN_HU/index.php?option=com_content&task=view&id=34&Itemid=58
- Samba75:
<http://www.falcom.de/products/mobile-data/samba75/>
- Bluegiga WT32 bluetooth modul:
http://www.bluegiga.com/WT32_Bluetooth_Audio_Module
- Xmodem protocol:
<http://www.programmersheaven.com/download/2167/Zipfilelist.aspx>
- Google Maps API:
<http://code.google.com/intl/hu-HU/apis/maps/>