



Enfora GSM know-how

GSM modem alapok

1.00 verzió, 2007-06-10

Jelen dokumentum az **Enfora** gyártmányú GSM/GPRS/GPS modemekkel ismerkedők számára lett összeállítva, és tartalmazza azokat a legfontosabb információkat, amelyek szükségesek a GSM modem megfelelő használatához. További információkat az Enfora honlapján és az Enfora CD-ken található PDF adatlapokban lehet találni, ill. a felmerülő kérdésekkel kapcsolatban lehet keresni a szerzőt, Varsányi Pétert az alábbi elérhetőségeken:

Mobil: (20)-942-7232

Fax: (1)-303-3433

E-mail: peter.varsanyi@componex-electronics.com

Témakörök:

1. Enfora képességek dióhéjban
2. Fejlesztői panel és használata
3. SIM és PIN kezelés
4. Hálózati információk lekérdezése
5. Hívás indítás és fogadás
6. SMS küldés és fogadás
7. GPS parancsok
8. GPRS kapcsolódás - Alapok
9. GPRS kapcsolódás - PPP
10. GPRS kapcsolódás - Windows[®]-al
11. GPRS kapcsolódás - IP cím kiosztás
12. GPRS kapcsolódás - UDP
13. GPRS kapcsolódás - TCP
14. EDGE modemek
15. EVENT funkciók **Bővült!**
16. CMUX működése

1. Enfora képességek dióhéjban

Az Enfora modemek az alábbi képességekkel rendelkeznek (más GSM-ekkel összehasonlítva):

- Egységes bekötésű 60 pólusú csatlakozó az összes modemhez, beleértve a GPS verziót is! Külön kérésre létezik 80 pólusú verzió is, megnövelt számú I/O lábbal. Bekötése igen logikus, magától értetődő, "felesleges" perifériákat az alapverzió nem tartalmaz, csak 2 analóg bemenetet, 1 analóg kimenetet (ez egyedi!), egy 8 bites szabadon programozható digitális I/O-t mindenféle adatrány megkötés nélkül. (Az új modellek pl. már 4 A/D-t és 80 pólusú csatlakozót tartalmaznak majd alapból!)
- Modulra forrasztott, belső SIM kártya foglalat és/vagy külső SIM kivezetések opcióban.
- Megnövelt hőmérséklet-tartomány más ipari modemekhez képest: -30 °C – +70 °C
- GSM Voice, Data (=CSD) és GPRS adatátviteli képességek, ez utóbbi Class 10 szinten, ez a jelenlegi maximális adatátviteli sebességet jelenti. Adatátviteli sebességre lefordítva ez max. 14,4 kbps CSD sebességet, max. 85,6 kbps GPRS letöltési és max. 42,8 kbps GPRS feltöltési sebességet jelent. Ennél nagyobb sebességek elérésére az EDGE verzió javasolt.
- SMS küldés-fogadás TXT (szöveges) és PDU (bináris) módon
- Beépített, fejlett GPS modul, állíthatóan közösített vagy külön soros vonali kapcsokkal.
- Beépített esemény-kezelő rendszer, amellyel nagyon sok programozást és külső hardvert igénylő alkalmazás mindössze néhány konfiguráló parancs egyszeri kiadására redukálódik; akár olyan szintű programozás is megoldható, mint pl. GPS pozíció megváltozásakor az új pozíció automatikus elküldése SMS-ben, GPRS-en. (Ez egy nagyon hasznos és eddig még a piacon egyedülálló tulajdonsága az Enforának; és mivel kész, tokozott modemek is kaphatóak, e funkcióit felhasználva akár komplex felügyeleti rendszerek készíthetők percek alatt!)
- Fejlett UDP/TCP kommunikáció GPRS alatt; az AT parancsokat UDP csomagként is ki lehet adni akár lokálisan, akár távolról; ez lehetővé teszi a modemek távprogramozását mindenféle külső/belső szoftverfejlesztés nélkül. Ugyancsak jelenleg egyedülálló az a 4 szintű IP cím kiosztó/meghatározó algoritmus, amellyel dinamikus IP címekkel is tudunk egy egyedileg címezhető GPRS rendszert felépíteni úgy, hogy közben nem nekünk kell a címek kiosztását lekezelnünk.

Az Enfora modemek az alábbi képességekkel NEM rendelkeznek más modemekhez képest:

- Beépített (embedded) program futtatási lehetőség; ezt bőven kárpótolja viszont az Event Engine. Külső processzorral ráadásul gyorsabb a fejlesztés. Egy megvalósult fejlesztés a Wavecom-on Embedded programként 44 kByte (a kötelező időosztásos működés miatt), és mindössze 4 kByte külső AVR processzorral!
- IIC/TWI/SPI interfész LCD meghajtásra, ill. keyboard mátrix szkennelő I/O-k. A 80 pól. csatlakozó opcióban azonban ezek is kérhetőek.
- Beépített Li-Ion akkumulátor-töltő
- GSM Státusz visszajelző LED
- Aktív GPS antennák meghajtásához feszültség kimenet; csak ajánlott külső áramkör van.
- Alkalmazói szintű TCP/IP hozzáférés, azaz POP3/SMTP, HTTP/FTP szolgáltatások. Csak a TCP szintig van megvalósítva az Internet, az azonban felár nélkül mindegyikben.
- Normál furatszerelt csatlakozási opció; csak egy fajta SMD csatlakozóval van gyártva.

2. Fejlesztői panel és használata

A fejlesztői panel **J202** csatlakozójára 4,5V–9V közötti, min. 1A-es tápot adunk. (Legegyszerűbb az 5VDC/1A dugasztáp, ez szinte mindenhol elérhető, pl. a Componex Electronics-nál kapható PHIHONG PSA05R-050 tápegység pl. ideális és nagyon megbízható típus ilyen célra). A **J204** a GSM modem 3,3V–4,5V közötti direkt tápja; itt akár 2A-es áramlökések is előfordulhatnak! A **J202** csatlakozón át a fejlesztő panel áramkörei (RS-232 leválasztás, LED-ek, stb.) is táplálást kapnak, míg a **J204** csak a GSM modult táplálja, így ha azt használjuk - mert pl. a saját GSM tápukat kell tesztelnünk - akkor is be kell kötni a **J202**-t! Ebben az esetben természetesen a **J204**-en keresztül csak a GSM modem fogyasztása fog folyni, míg a LED-ek és egyéb áramkörök energia-felvételét a **J202** fogja fedezni. A **D206** jelzi a **J202**-n lévő táp meglétét, a **D209** pedig a GSM modem tápjának meglétét. Ez utóbbit az **SW202** választja ki, amit a használt tápcsatlakozó irányába kell kapcsolni.

Az **RS232 Modem** feliratú csatlakozót a PC-hez kötjük; sima 1:1-es, 9 eres ún. soros vonali toldó kábellel tudjuk ezt megtenni. A **Windows® Hyperterminál®** programját elindítva az alábbiakat állítsuk be: Csatlakozás fül, Eszköz: *Közvetlenül a COMx portra*, Beállítás...: 115200 bit/sec, 8 bit, nincs paritás, 1 stopbit, hardver átvitelvezérlés. Ezután az **SW201** kapcsolót **ON** állásba kapcsoljuk, mire a LED-ek világítani kezdenek. Ezt követően az **SW203** kapcsolót **OFF** állásba kell kapcsolni az **EDGE** modemek (EDG0100-0x) esetén, míg **ON** állásba az összes többi modem esetén, mire a GSM modem elindul, és a terminál képernyőn rövidesen megjelenik az '*AT-Command Interpreter ready*' szöveg. A modem máris működik. (Ha periodikusan zagyva szöveg jelenik meg, akkor az **RS232 Debug** csatlakozóra csatlakoztunk!) A kommunikáció és a modem ellenőrzése: az ATI vagy ATIO parancsra az '*Enfora, L.P.*' szöveg, az AT11 parancsra a modem típusa (pl. '*Enabler-II G Modem*') jelenik meg, míg az AT&V parancsra az alábbi gyári beállítások láthatóak:

Enfora Enabler-IIG Dual (GSM0108-0x) és Quad (GSM0116-0x) verzió esetén:

```

E1 Q0 V1 X0 &C1 &D0 S0:000 S3:013 S4:010 S5:008 +CBST: 7,0,1 +FCLASS:0
+CRLP: 61,61,48,6 +CR:0 +CRC:0 +CMGF:1 +CSCS:"PCCP437" +CNMI:1,1,0,0,0
+ILRR:0 +IPR:115200 +CMEE:0 +CSMS:0,1,1,1 +CFUN:1 +CREG:0 +CGREG:0
+CLIP:0,2 +CAOC:1 +COPS:0,0 +CSNS:0 +CSSN:0,0 +CSDH:0 +CMUT:0
+COLP:0 +CPBS:"AD" +CGAUTO:3 +CGSMS:3 +CGEREP:0,0 +CGCLASS:"B"
+CSVM:0,"",129 +CGQREQ +CGQMIN
+IFC:2,2 +ICF:3 %CGAATT:0,1 %CGPPP:3 $AREG:1 $HOSTIF:0
$VGR: 20 $VGT: 8 $VLVL: 4 $VST: 0 $VSELECT: 0
$PADBLK: 512 $PADBS: 08 $PADCMD: 001b $PADDST: "000.000.000.000", 0
$PADFWD: 0d $PADSRC: 0 $PADTO: 50 $UDPAPI: "199.245.180.013", 1720
$ACTIVE: 1 $CONNTO: 60 $IDLETO: 120 $GATEWAY:000.000.000.000
$MSCLS: 8 %SLEEP: 2 %CPI: 0 %CGREG: 0 $ACKTM: 0, 0, 0
$MDMID: ""
$EVTIM1: 0, $EVTIM2: 0, $EVTIM3: 0, $EVTIM4: 0
$WAKEUP: 0, 0 $NETMON: 0, 0, 0, 0
$IOCFG: 11111111, $IOGPA: 11111111
$EVENT:  evgp evtyp evcat    p1    p2

```

OK

Enfora Enabler-IIG-A-GPS (MLG0208-0x) verzió esetén:

```
E1 Q0 V1 X0 &C1 &D0 S0:000 S3:013 S4:010 S5:008 +CBST: 7,0,1 +FCLASS:0
+CRLP: 61,61,48,6 +CR:0 +CRC:0 +CMGF:1 +CSCS:"PCCP437" +CNMI:1,1,0,0,0
+ILRR:0 +IPR:115200 +CMEE:0 +CSMS:0,1,1,1 +CFUN:1 +CREG:0 +CGREG:0
+CLIP:0,2 +CAOC:1 +COPS:0,0 +CSNS:0 +CSSN:0,0 +CSDH:0 +CMUT:0
+COLP:0 +CPBS:"AD" +CGAUTO:3 +CGSMS:3 +CGEREP:0,0 +CGCLASS:"B"
+CSVM:0,"",129 +CGQREQ +CGQMIN
+IFC:2,2 +ICF:3 %CGAATT:0,1 %CGPPP:3 $AREG:1 $HOSTIF:0
$VGR: 20 $VGT: 8 $VLVL: 4 $VST: 0 $VSELECT: 0
$PADBLK: 512 $PADBS: 08 $PADCMD: 001b $PADDST: "000.000.000.000", 0
$PADFWD: 0d $PADSRC: 0 $PADTO: 50 $UDPAPI: "199.245.180.013", 1720
$ACTIVE: 1 $CONTO: 60 $IDLETO: 120 $GATEWAY:000.000.000.000
$MSCLS: 8 %SLEEP: 2 %CPI: 0 %CGREG: 0 $ACKTM: 0, 0, 0
$MDMID: "" $MSGLOGEN: 0 $PKG: 19
$EVTIM1: 0, $EVTIM2: 0, $EVTIM3: 0, $EVTIM4: 0
$WAKEUP: 0, 0 $NETMON: 0, 0, 0, 0
$IOCFG: 11110111, $IOGPA: 11111111
$GPSLCL: 0, 0, $GOPMD: 1, 0, 0, $GPSSRC: 1721
$GPSDST: "000.000.000.000", "000.000.000.000", 0, 1, ""
$EVENT: evgp evtyp evcat p1 p2
```

OK

Enfora Enabler-IIE EDGE (EDG0100-0x) verzió esetén:

```
E1 Q0 V1 X0 &C1 &D0 S0:000 S3:013 S4:010 S5:008 +CBST: 7,0,1 +FCLASS:0
+CRLP: 61,61,48,6 +CR:0 +CRC:0 +CMGF:1 +CSCS:"PCCP437" +CNMI:1,1,0,0,0
+ILRR:0 +IPR:115200 +CMEE:0 +CSMS:0,1,1,1 +CFUN:1 +CREG:0 +CGREG:0
+CLIP:0,2 +CAOC:1 +COPS:0,0 +CSNS:0 +CSSN:0,0 +CSDH:0 +CMUT:0
+COLP:0 +CPBS:"MT" +CGAUTO:3 +CGSMS:3 +CGEREP:0,0 +CGCLASS:"B"
+CSVM:0,"",129 +IFC:2,2 +ICF:3 %CGAATT:0,1 %CGPPP:3 $AREG:1
$VGR:20 $VGT:8 $VLVL:4 $VST:0 $VSELECT:0
$MSCLS:10,10,1 %SLEEP:2 %CPI:0 %CGREG:0 $OFFDLY:800
```

OK

Javasolt további beállítások az összes verzió esetén:

AT&F	Gyári alapbeállítások visszatöltése a biztonság kedvéért.
ATX1	Kiterjesztett formátumú CONNECT üzenet (Tényleges sebesség kijelzése.)
AT&D2	DTR vonallal vezérelt vonalbontás (HyperTerminal [®] kompatibilitás miatt)
AT+CMEE=2	Kiterjesztett hibáüzenetek kijelzése számkóddal és magyarázó szöveggel.
AT+CREG=1	GSM hálózati regisztráció állapota.
AT+CGREG=1	GPRS hálózati regisztráció állapota.
AT+CGDCONT=1,"IP","internet"	GPRS alapbeállítása HyperTerminal [®] -hoz
AT&W	Eltároljuk ezeket, hogy a következő bekapcsolásra is így maradjanak.

3. SIM és PIN kezelés

A modem bekapcsolását követően az első feladat a SIM beírása, ha szükséges. Az AT+CPIN? parancsra a modem válasza négyféle lehet:

ha '+CPIN: SIM PIN', akkor várja a PIN kódot;

ha '+CPIN: SIM PUK', akkor várja a PUK kódot;

ha '+CPIN: READY', akkor a SIM PIN megadás nem kell, vagy már meg van adva; míg

ha 'ERROR', akkor nincs vagy hibás a SIM kártya, vagy annak csatlakozása nincs rendben.

A SIM PIN megadás parancsai:

SIM PIN: AT+CPIN="IIII" Az idézőjel itt fontos; más GSM-ek ezt nem igénylik!

SIM PUK: AT+CPIN="UUUUUUUU", "IIII" Ezzel meg lehet változtatni a PIN-t is.

(Az IIII helyett természetesen a PIN kód, az UUUUUUUU helyett a PUK kód áll.)

Az Enfora GSM-ek sajnos nem rendelkeznek olyan üzemállapot visszajelző LED-del, mint pl. a Wavecom vagy SimCom modemek, amely a villogási frekvenciájával jelezné a hálózatra történő felcsatlakozás állapotát. Ehelyett az AT+CREG=1 majd AT&W parancsokkal eltárolhatjuk a hálózati regisztráció kijelzését. Ebben az esetben az alábbi üzenetek jelennek meg menet közben:

+CREG: 2 GSM hálózat keresése elkezdődött

+CREG: 3 GSM hálózat keresése sikertelen, mert nincs SIM, vagy nincs PIN megadva

+CREG: 1 GSM hálózat keresése sikerült, bejelentkezés rendben.

A SIM PIN kérés kikapcsolása véglegesen: AT+CLCK="SC",0,"IIII". Visszakapcsolás: AT+CLCK="SC",1,"IIII" Az IIII helyett természetesen a PIN kód álljon!

Vigyázat! A GSM modemek sok parancsot mindaddig nem hajtanak végre, amíg a SIM PIN nincs megadva! Ha a parancsokra ok nélkül ERROR-t kapunk, kérdezzük le, jó-e a SIM az AT+CPIN? paranccsal! Amennyiben itt is csak egy ERROR-t kapunk, adjuk ki az AT+CMEE=1 vagy AT+CMEE=2 parancsot, és ismételten adjuk ki az AT+CPIN? parancsot. Ha a kapott válasz +CME ERROR: 10 (+CMEE=1-nél) vagy +CME ERROR: SIM not inserted (+CMEE=2-nél), akkor ez valószínűleg SIM kompatibilitási hiba. Ezt kérjük jelezze a Componex Electronics-nak vagy közvetlenül nekem.

4. Hálózati információk lekérdezése

Az alábbiakban pár fontosabb hálózati/vételi információ lekérdezése következik:

GSM térerő és jelminőség: AT+CSQ Válasz: +CSQ: xx,99
Aktuális GSM szolgáltató: AT+COPS? Válasz: +COPS: 0,0,"T-Mobile H"
Lehetséges GSM szolgáltatók: AT+COPS=? Válasz:
+COPS: (2,"T-Mobile H","TMO H","21630"),
(1,"vodafone HU","voda HU","21670"),
(1,"H PANNON GSM","PANNON","21601")

Aktuális GSM torony adatai: AT%EM=2,1
Környező GSM tornyok adatai: AT%EM=2,3
Válasz: Összetett, lásd alábbi dokumentáció: *GSM0000TN012 - Engineering Mode Manual.pdf*

A GSM típusának azonosítója: AT+CGMM Válasz: *Enabler-II G Modem*
Aktuális modem verziószám: AT+CGMR Válasz: *S/W v0.6.1*
GSM modem IMEI azonosítója: AT+CGSN Válasz: *IMEI szám*
SIM kártya azonosítója: AT+CIMI Válasz: *SIM szám*

Részletes hibaüzenet számmal: AT+CMEE=1
Részletes hibaüzenet szöveggel: AT+CMEE=2

5. Hívás indítás és fogadás

Hívás indítása: ATD<hívószám> vagy ATDL (Utolsó szám újrAhívása)
Hívás felvétele automatikusan: ATSO=x ahol x a csengetések száma; 0 = nincs felvétel.
Hívás felvétele manuálisan: ATA a RING üzenetek megjelenése után működik

A hívószámok beírásánál kerüljük a vezetékes modemeknél megszokott ATDT ill. ATDP formát, bár az előbbi még elfogadja a modem. Ugyancsak kerüljük a zárójeleket, kötőjeleket, szóközőket, mert bizonyos esetekben elfogadja az ilyesmit a modem, bizonyos esetekben viszont nem!

Ha a hívószámot pontosvessző (;) karakterrel zárjuk le, akkor a hívás nem CSD, hanem VOICE módban fog kimenni, a modem pedig visszatér a parancs-módba.

Hívástípus kijelzés bekapcsolása: AT+CRC=1 Válasz:
+CRING: REL ASYNC Normál adathívás (==CSD)
+CRING: VOICE Voice (audió) hívás
Hívószám kijelzés bekapcsolása: AT+CLIP=1,1

Ha a hívószámot nagy "I" betűvel (I) zárjuk le, akkor a hívott fél telefonján történő hívószám-kijelzést le tudjuk tiltani. Ha a hívószámot kis "i" betűvel (I) zárjuk le, akkor ezzel engedélyezzük a hívószám-kijelzést; ez amúgy az alapértelmezett eset.

Ismeretlen hívószámú SIM kártya hívószámának megállapítására egyetlen mindig járható út van: küldeni kell egy SMS-t egy ismert hívószámra, pl. saját mobilunkra pl. az alábbi paranccsal:

```
AT+CMGS="06209427232" <CR>  
> Teszt <CTRL-Z>
```

A kapott SMS fejlécében pedig mindig szerepelni fog az SMS küldő telefonszáma is.

7. GPS parancsok

A GPS verzió jelenleg még csak fejlesztési szakaszban van, de már működőképes és használható. A modembe beépített GPS nem tekinthető professzionális célú megoldásnak: az érzékenysége sem a legideálisabb, a teljesítmény-felvétele sem túl alacsony, ill. aktív (azaz külső) GPS antennát is csak az ajánlott kiegészítő táp-áramkör külső megvalósítása után lehet csatlakoztatni hozzá. Ezért főképp az alábbi alkalmazásokra ajánljuk:

1. Terepi mérési adatoknál a mérés helyének automatikus rögzítéséhez, csak nyílt terepen.
2. Pontos időadatok megállapításához (A GPS atomóra-pontosságú időadatot szolgáltat!)
3. Olcsó GPS megoldás olcsó/kis processzor mellé, amely nem rendelkezik több soros vonali csatlakozással, így egyetlen soros vonalon keresztül kezelhető GSM, SMS, GPRS és GPS.
4. Riasztókba, járművekbe történő telepítésre célszerűbb a μ -Blox különálló GPS moduljait felhasználni, melyeket szintén a *Componex Electronics* forgalmaz, és melyek aktív antenna csatlakozással, széles méret-, teljesítmény-felvét- és érzékenység-választékban készülnek.

A GPS modulnál először meg kell határozni, hogy milyen módon kívánjuk a NMEA formátumú adatokat elérni:

1. Kérhetjük azt, hogy a Modem RS-232 csatlakozásán keresztül, a modem adatokkal szimultán férhessünk hozzá az adatokhoz. Ebben az esetben az `AT$GPSLCL=1,XX` parancsot kell kiküldeni.
2. Kérhetjük azt, hogy a Debug RS-232 csatlakozásán keresztül, a modem adataitól függetlenül és folyamatosan férhessünk hozzá az adatokhoz. Ebben az esetben az `AT$GPSLCL=5,XX` parancsot kell kiküldeni, és alapesetben 4800 baud, 8N1 formátumban fogjuk az adatokat változatlan formában megkapni a Debug porton. Az adatátviteli sebességet NMEA paranccsal lehet megváltoztatni ezen a porton.
3. Végezetül kérhetjük azt is, hogy a GPRS kapcsolatot felhasználva, UDP csomagokba rakva a modem automatikusan küldje el egy megadott szerver számára a mindenkori GPS koordináta adatokat, ill. az Event funkciók felhasználásával az automatikus SMS küldés is megoldható bizonyos események bekövetkeztekor, pl. a alappozíció megváltozásakor, külső hardver jelek megváltozásakor, stb. (Erről majd lásd később.) Ez a funkció lehetővé teszi egyszerű és olcsó felügyeleti-riasztó funkciók megvalósítását!

Másodszor pedig azt is meg kell határozni, hogy a GPS milyen adatokat szolgáltatson számunkra:

Bináris maszk:	NMEA kód:	Megjegyzés:
1	GGA	Global Positioning System Fixed Data (Idő, hosszúság, szélesség, kiszámítási módszer, pontossági adatok)
2	GLL (Jön!)	Geographic Position - Latitude/Longitude (Hosszúság, szélesség, idő, érvényesség, üzemmód)
4	GSA	GNSS DOP and Active Satellites (2D/3D mód, max. 12 követett műhold adatai, pontossági adatok)
8	GSV	GNSS Satellites in View (Max. 3 aktív műhold részletes adatai külön adatsorokban, amik alapján a pozíció ki lett számolva)
16	RMC	Recommended Minimum Specific GNSS Data (Dátum és idő, érvényesség, hosszúság, szélesség, magasság, sebesség)
32	VTG (Jön!)	Course Over Ground and Ground Speed (Dőlésszögek és vízszintes sebességek a mértékegységeikkel és az üzemmóddal)

A kívánt adatcsomagok kódjait összesíteni kell, és azt kell beírni az XX helyére; azaz pl. az AT\$GPSLCL=1,63 parancs az összes létező adatot kiadja, míg a AT\$GPSLCL=1,16 parancs csak a leghasznosabb RMC csomagokat, amelyekben minden lényeges GPS információ szerepel. Lehetőség van egyszeri, célzott tartalmú adatkiolvasásra is az AT\$GPSRD=XX parancs kiadásával. (A NMEA szabvány leírása jelen leírásnak nem tárgya; amúgy is tele van vele az Internet.)

Második lépésben meg kell határozni, hogy milyen üzemmódban működjön a GPS, és mikor adjon adatot: csak most egyszer; mindig, amikor jön valami; vagy adott időközönként adjon egy GPS fotót. Pl. a folyamatos adatkiadást az alábbi paranccsal tudjuk bekapcsolni: AT\$GOPMD=1,2

Ezek után, ha nem egy föld alatti bunkerben vagyunk, hanem célszerűen a nyílt terepen, de legalább az ablak közelében, akkor pár perc után már kezdenek érkezni a műhold adatok, és előbb az időadatok fixálódnak, aztán 8-10-12 műhold pályája is megállapításra kerül, végül, ha elegendő adat áll a GPS vevő rendelkezésére, megjelenik az első "A" státuszú csomag is az első pozíció-adatokkal. A GPS működéséből adódóan ez még közel sem lesz pontos, hiszen a pontos mérést lehetővé tevő korrekciós adatokat csak pár óránként küldik le a műholdak. Ezért lehetőség van pl. egy szerverről automatikusan letölteni ezeket a korrekciós adatokat, amelyek segítségével így a több óra helyett percek alatt megállapítható a pontos koordináta. (De ez már megint nem ennek a leírásnak a része...)

A kapott adatcsomagok mintája egy teljes körű adatszolgáltatás esetén:

```
$GPGGA,172851.37,4729.725436,N,01904.609031,E,1,03,3.0,142.0,M,,M,,*71
$GPGSA,A,3,07,,16,22,21,,,,,0.0,0.0,0.0*31
$GPGSV,3,1,12,07,65,034,31,03,59,034,,18,58,095,,16,49,217,39*7D
$GPGSV,3,2,12,22,49,163,41,21,44,064,35,19,28,034,,26,12,034,*75
$GPGSV,3,3,12,29,10,034,,27,06,034,,06,04,034,,08,02,034,*7B
$GPRMC,172851.37,A,4729.725436,N,01904.609031,E,0.0,264.0,011106,,,*1C
```

Ebből számunkra lényeges a legutolsó sor, mely szerint az alábbi dátum/idő/GPS adatunk van:

```
$GPRMC,172851.37,A,4729.725436,N,01904.609031,E,0.0,264.0,011106,,,*1C
```

<i>Név:</i>	<i>Minta:</i>	<i>M.e.:</i>	<i>Magyarázat:</i>
Message ID	\$GPRMC		RMC protocol fejléc
UTC Time	172851.37		hhmmss.sss, azaz 17:28:51, 37 századmásodperc
Status	A		A = Pozíció érvényes / V = Pozíció nem érvényes
Latitude	4729.725436		ddmm.mmmm, szélességi fok, <i>lásd megjegyzés!</i>
N/S Indicator	N		N = északi szélesség / S = déli szélesség
Longitude	01904.609031		dddmm.mmmm, hosszúsági fok, <i>lásd megjegyzés!</i>
E/W Indicator	E		E = keleti hosszúság / W = nyugati hosszúság
Speed Over Ground	0.0	csomó	
Course Over Ground	264.0	fok	
Date	011106		ddmmyy, azaz 2006-11-01
Nem használt			
Nem használt			
Checksum	*1C		

Megjegyzés:

Gyakori tévedés, hogy a NMEA formátumban kapott szélességi és hosszúsági koordinátákat nem értelmezik megfelelően. A koordináták ugyanis nem *egészfok.törtfok* formátumban vannak megadva, hanem *ddmm.mmmm* formátumban, így felhasználás előtt előbb át kell őket számolni a megszokott formátumra. A fenti példánál maradva, a kapott 4729.725436 szélességi koordinátáról le kell vágni a szögperc részét, azaz a 29.725436-ot, majd el kell osztani 60-al, így 0.495424-et kapunk tizedes résznek; majd ehhez hozzá kell adni az egész részt, a 47-et, tehát a végeredmény 47.495424 fok lesz. Hasonlóképp a 01904.609031 hosszúsági koordinátáról is le kell vágni a szögperc részét, azaz itt a 04.609031-et, és ezt is el kell osztani 60-al, így 0.076817-ot kapunk tizedes résznek; majd ehhez is hozzá kell adni az egész részt, a 019-et, tehát a végeredmény 19.076817 fok lesz. Hogy az élet még szebb legyen, itt van egy másik minta adat, ahol egy harmadik féle kiírási mód is látható. Ennél nem osztani, hanem épp szorozni kell, mégpedig a töredék szögperceket 60-al, hogy szögmásodperceket kapjunk. Azaz a $0.979' * 60 = 58.74''$ ill. $0.491' * 60 = 29.46''$

Enfora által szolgáltatott NMEA formátum: N 4729.979 E 1906.491 (ddmm.mmm)

PDA-s térkép-programok által kiírt formátum: N 47.49965 E 19.10818 (dd.ddddd)

Google által kiírt szögmásodperces formátum: +47° 29' 58.74" +19° 6' 29.45" (dd° mm' ss.ss")

8. GPRS kapcsolódás - Alapok

A GSM-ek világában kétségtelenül a GPRS a legbonyolultabb, ámde mostanában a legkeresettebb szolgáltatás. Ezért erről - mivel sokan kevésbé ismerik - szeretnék kicsit bővebben szólni, és többféle irányból is megmutatni, hogy mi is ez, mire is jó ez.

Először nézzük meg GSM hardver irányból, mi is a GPRS? A GSM hálózatról annyit illik tudni, hogy 2 frekvencia-tartománya létezik: egy alsó és egy felső tartomány. Európában az alsó tartomány 900 MHz-en, a felső tartomány 1800 MHz-en van, míg Amerikában 850 MHz ill. 1900 MHz a két használatos frekvencia-tartomány. Ennek megfelelően egy kétsávós (ún. Dual Band) GSM Európa területén használható, míg egy négysávós (ún. Quad Band) az egész világon. Ezek a frekvenciasávok tovább vannak osztva egy alsó és egy felső félre; ezek az ún. UpLink és DownLink frekvenciák; az egyikben a GSM modem küld adatot a toronynak, a másikban a torony küld adatot a GSM modemnek. Ezek a frekvencia-sávok további, 200 kHz széles frekvenciasávokra vannak osztva, ezek az ún. csatornák. A 900/850 MHz-es frekvencia-sávban 124 csatorna van, míg a 1800/1900 MHz-es frekvencia-sávban 374 csatorna. Egy-egy csatorna további 8 db időszeletre vagy időrésre (ún. Time Slot) van osztva, és egy-egy ilyen időszellet összesen 15625 bitnyi információt szállít. A GSM hálózat - pl. hogy ne legyen annyira érzékeny a rádiófrekvencia-zavarokra - ezen csatornák és időszeltek között folyamatosan ugrál, tehát ilyen 4,615 ms-onként ismétlődő, 577 μ s hosszúságú adatcsomagok formájában történik MINDIG az adatok továbbítása, legyen szó VOICE vagy CSD hívásról, vagy a GPRS-ről. (És ez adja pl. a GSM-ek jellegzetes brekegő hangját hangerősítők közelében.)

A GPRS a **G**lobal **P**acket **R**adio **S**ystem-nek a rövidítése, és a fent leírt csatornák és időszeltek jobb kihasználása érdekében készítették el. A VOICE és CSD hívások esetén ugyanis, ha egy adott frekvencián éppen adatátviteli probléma áll fent, akkor a rendszer gyorsan keres egy másikat, és ott próbálja meg újra átvinni az adatokat. CSD hívások esetén pl. állítható is, hogy hányszor próbáljon egy adatcsomagot átküldeni, mielőtt bontja a hívást. Ebből adódóan a csatornák és időszeltek által biztosított elméleti 172 kbit/sec (!!!) adatátviteli sebesség soha sincs kihasználva. Ezen tud segíteni a GPRS azzal, hogy ezeket az amúgy üresen maradó adatátviteli lyukakat betömökdi adatcsomagokkal és így jobban kihasználja a szolgáltató rendelkezésére álló nagy adatátviteli kapacitását. Ez a logika a GPRS három fő tulajdonságát rögtön érthetővé teszi:

- Míg a CSD hívások átlagsebessége kötött, addig a GPRS sebessége pillanatról-pillanatra elég jelentős mértékben meg tud változni. (Azért az átlag szó kiemelése, mert a CSD hívások is csomagkapcsolt adatátvitel, tehát a rendszer működéséből adódóan akár 2,88 másodperces adatátviteli kimaradás is felléphet elvben, amely után olyan nagy sebességű adatátviteli lökés léphet fel, amely a 14,4 kbit/sec többszörösét is kiteheti. A kommunikáció egészére nézve azonban - több perces átlagban - pontosan igaz lesz a 14.4 kbit/sec-es átlagsebesség.)
- Mivel a VOICE és CSD hívások esetén fent kell tartani egy biztonsági tartalékot, és elvileg az adatátvitel folyamatos, ezért ezek idő alapon vannak számlázva. A GPRS esetén viszont, mivel az időzített adatátvitel elvileg sem garantált, kizárólag forgalom alapú a számlázás! A GPRS hálózaton töltött idő így teljesen lényegtelen; akár bekapcsolástól-kikapcsolásig aktív lehet a GPRS kapcsolat, miközben tudunk VOICE és CSD hívásokat is indítani vagy fogadni!
- Maga a "GPRS képesség" csak adatcsomagok elküldését és fogadását, továbbá pár rendszer-üzenet kezelését jelenti. A GPRS nem azonos a mobil Internettel! A GPRS maga még azzal sem foglalkozik, hogy elvesznek-e a csomagok adás közben, vagy sem! Ezért a GPRS-re, mint alap szolgáltatásra egy több rétegű szolgáltatás-csomag épülhet. (Az ún. TCP/IP verem.)

Mire használható tehát a GPRS? Íme néhány példa a teljesség igénye nélkül:

- CSD hívások helyett adatok küldésére költség-hatékony módon. Ugyanannyi adatot töredék áron tudunk átküldeni egyik helyről a másikra GPRS kapcsolaton, mint CSD kapcsolaton.
- 9600/14400 bit/sec-nél nagyobb vagy kisebb sebességű adatátvitel megvalósítására. Mivel a CSD hívások mindig idő alapon vannak számlázva, ezért az a célszerű, ha kihasználjuk a max. sebességgel - pl. csak 2400 bit/sec-el - küldi az adatokat. Másrészt ismerte a csatornák elvi maximális sebességét (172 kbit/sec), akár a CSD átviteli sebesség többszöröse is létrejöhet, különösen a ritkán használt időszakokban, pl. éjszaka, amikor nincs VOICE hívás, így sok csatorna és idő-szelet van szabadon. (Sajnos nehéz a GPRS-re jellemző adatátviteli sebességet mondani, mert ez nagyon sok paramétertől függ. Kb. 40-60 kbit/sec mindenesetre reálisan elérhető a napi gyakorlatban; EDGE modemekkel ennek háromszorosa is könnyen elérhető.)
- SMS-ek vagy ahhoz hasonló rövid vezérlő üzenetek küldése töredék költségen. Érdemes egy kicsit elgondolkodni, hogy mekkora az árkülönbség egy kb. 10 Ft-os, max. 160 karakteres SMS (1120 bit/10 Ft = 112 bit/Ft) ill. egy kb. 10 Ft/Mbit díjtételű (100.000 bit/Ft) GPRS ára között: potom kilencszázszoros!!!
- Készenléti üzemű készülékek adatátviteli megoldása. Sok esetben fordul elő, hogy egy adott készülék csak megadott időközönként - pl. percenként, negyedóránként - gyűjt egy pár byte méretű adatsomagot. Ezeket normál esetben bufferelni kell, és általában adott időszakonként egyszer ki kell olvasni. A központban így az utolsó időszak adata mindig hiányozni fog, mert még a távoli egység bufferében lesz, így nem lesz naprakész az adatbázis. GPRS kapcsolat esetén - mivel egy adatsomag alig 61 byte méretű - ez a kis mennyiségű adat is könnyedén elküldhető azonnal a központba, így elmarad a bufferelés és az ezzel járó nagy adatvesztési kockázat. (Ki ne járt volna úgy, hogy egy memóriahiba miatt törlődött több napi/heti fontos adata? Nem véletlenül olvassák ki naponta a fontos adatokat, miközben elvileg több megabyte méretű memóriákkal akár hetekre is tárolható lenne minden.)
- Folyamatos adatátviteli lehetőség VOICE hívás közben! Pl. flotta-követő rendszerekben akár az is megvalósítható, hogy miközben a sofőr a központtal beszél élőben, addig a központból elküldik neki a térkép-adatokat, ami rögtön megjelenik a sofőr képernyőjén is.
- Enfora GSM modemek esetében újszerű felhasználási mód lehet az ún. központi alkalmazás-fejlesztés lehetősége. Ez annyit jelent, hogy az **Enfora** GSM modemek - bár nem tartalmaznak belső programfuttatási lehetőséget - tartalmaznak egy igen rugalmasan és jól programozható Event Engine-t, ill. képesek mindennemű adat-kommunikációjukat remote módban önállóan is végrehajtani. Tehát pl. a GPS adatokat folyamatosan, vagy egy esemény bekövetkeztére egy adott címre el tudják küldeni GPRS kapcsolattal. Így az adatokat feldolgozó-kiértékelő rutint nem a GSM modem, vagy a mellé telepített távoli processzor fogja futtatni, hanem egy GPRS kapcsolattal rendelkező központi szerver, amely így könnyen, kényelmesen fejleszthető, nincs memória- és erőforrás korlát, nincs megnövekedett áramfelvétel, geometriai méret probléma, stb. Az Enfora modemek a fenti logika szerinti felépítésükkel jelenleg egyedül állnak a piacon!
- Bár az előző oldalon még azt írtam, hogy a GPRS nem azonos a mobil Internettel, mégis a GPRS kapcsolat az, amellyel megvalósítható Internet kapcsolatot tudunk létrehozni mindkét irányban. Azaz mint kliens, mobilan el tudunk érni fixen vagy mobilan telepített szervereket, ill. mint szerver, mi is tudunk olyan információkat feltenni az Internetre, amely a világ bármely pontjáról akár egy egyszerű böngésző programmal elérhető. Ez utóbbi viszont mindenképpen igényli egy külső mikroprocesszor alkalmazását, mely a gyűjtött adatokat HTML formátumba konvertálja, hogy azt a böngészők meg tudják jeleníteni. (Ilyen szempontból ma még egyedül a **Wavecom** cég az, amely teljes körű szolgáltatást tud biztosítani egy GSM modemben belül.) Az Enfora ezt még nem tudja, de minden egyéb feltételt biztosít - kb. féláron!!!

9. GPRS kapcsolódás - PPP

Ennyi bevezető után lássuk végre a GPRS-t a gyakorlatban is! A GPRS kapcsolódásnak több szintje/fázisa is van. Igaz ez magára a GPRS-re is, meg az utána következő szolgáltatásokra is. A GPRS kapcsolódás alap parancsai szerencsére szabványosítva vannak, AT+ -szal kezdődnek, míg az **Enfora** kiegészítései AT\$ vagy AT% kezdetűek. Így az alábbiakban leírtak elvben az összes GSM modemre igazak, míg a parancsok közül csak az AT+ -szal kezdődőek használhatók máshol:

A PIN kód beírása után a GPRS - általában ún. DETACH állapotban van, azaz a GSM modem nem figyeli a GSM hálózaton lévő GPRS vezérlő üzeneteket. Általában - de nem az Enforánál! Az előző oldalon írtam, hogy ma az **Enfora** az egyetlen olyan GSM modem, amely önmagában alkalmas a központi alkalmazás-fejlesztésre. Ennek megfelelően az **Enfora** modem bizonyos felprogramozás esetén már rögtön a bekapcsolás után fel szeretne lépni a GPRS hálózatra. Mivel elég rejtélyes problémákat okozhatnak a fenti beállítások, gyorsan adjuk ki őket; később még úgyis hallunk róluk:

AT\$AREG=1	Nem kérünk automatikus GPRS regisztrációt, csak GSM-et.
AT\$HOSTIF=0	Nem kérünk UDP/TCP szintű GPRS-t, csak sima modemet.
AT&W	Eltároljuk, hogy a következő bekapcsolásra is így maradjon.

A szabványos AT+CGREG=1 majd AT&W parancsokkal tároljuk el a GRS hálózati regisztráció kijelzését. **Enfora** esetén használhatjuk még az AT%CGREG=1 majd AT&W parancsokat is, ami kicsit bővebb kijelzést ad. Ebben az esetben az alábbi üzenetek jelennek meg menet közben:

+CGREG: 2	GPRS hálózat keresése elkezdődött
+CGREG: 3	GPRS hálózat keresése sikertelen, mert nincs SIM, vagy nincs PIN megadva
+CGREG: 1	GPRS hálózat keresése sikerült, bejelentkezés rendben.

Adjuk ki az AT+CGATT=0 parancsot, amely lecsatlakoztat minket a GPRS kapcsolatról. Ezzel a GPRS-t teljesen kikapcsoltuk. (Le is kérdezhajtuk az aktuális állapotot az AT+CGATT=? parancssal, amire 0 lesz a helyes válasz, a GPRS nem aktív. Most tudjuk a GPRS paramétereit beállítani. Az első és legfontosabb paraméter a GPRS-ben, az ún. APN szerver neve. Ezt az AT+CGDCONT parancssal tudjuk állítani. Az APN az **Access Point Name** rövidítése, azaz azt határozza meg, hogy melyik kommunikációs szerveren át kívánunk csatlakozni a GPRS-hez. Ezek a szerverek egyben azt is meghatározzák, hogy mit érhetünk el GPRS-el. Három jellegzetes értéke lehet:

- **"internet"** vagy **"net"** Ha ezt a beállítást használjuk, akkor a nagybetűs Internetet érjük el. Ekkor a mi GSM modemünknek is lesz egy - általában - dinamikus IP címe, és elérhetjük bármelyik Internetre kötött szervert, pl. a <http://www.google.com/>-ot. Elvileg ezt a beállítást kell akkor is használnunk, ha mondjuk beágyazott WEB szervert szeretnénk létrehozni a GSM modemmel, bár ebben az esetben olyan előfizetésre van szükségünk, ami statikus IP címet ad, különben nem tudunk nevet regisztrálni a WEB szerverünknek.
- **"wap"** Ha ezt a beállítást használjuk, akkor a mobilos Internetet érjük el. Ez annyiban különbözik az előzőtől, hogy itt olyan módosított tartalmú WEB oldalakat fogunk elérni, amik a mobiltelefonok kicsi kijelzőjén is megjeleníthető; amolyan lebutított Internet. Ezért főleg a mobilok GPRS beállítása szokott ez lenni.

- **"cégnév.hu"** Ha ezt a beállítást használjuk, akkor egy, a GSM szolgáltatónál vagy nálunk elhelyezett saját szervert érhetünk el, és ezzel létrehoztunk egy saját, a világtól teljesen független hálózatot. Ez a hálózat jellemzően egy VPN hálózat szokott lenni, azaz egy saját jelszóval ÉS titkosítással rendelkező külön magán-hálózat. Ennek főleg biztonsági okai lehetnek; pl. egy céges flottakövető rendszernél igencsak bajos lenne, ha valaki nyomon tudná követni kívülről, hogy melyik kamionunk éppen merre jár, és mi a rakománya.

(Érdekességként megjegyzem, hogy a *T-Mobile* ez utóbbi, egyedi céges APN üzemmódot két különböző név alatt oldja meg a számlázás megoldásának függvényében. Ha az APN nevének formája **cégnév.gr.hu**, akkor a megrendelő cég számára egyben történik az egyes SIM-eken zajló forgalom mérése és számlázása, míg a **cégnév.co.hu** forma esetén az egyes SIM-ek forgalma külön-külön kerül mérésre és számlázásra. Erről, és a céges APN-ek létrehozásának és üzemeltetésének mikéntjéről egyébként részletes dokumentáció kérhető a *T-Mobile*-től, ill. a *Componex Electronics Hungary Kft*-től.)

A többi beállítás szolgáltató- ill. GSM függő; itt és most nem térek ki rá. Annyit jegyzek csak meg, hogy 4 különböző környezetet (=context) tudunk előre definiálni, és köztük tudunk váltogatni. A tesztelés céljából most állítsuk be az Internetet:

Beállítás:	AT+CGDCONT=1,"IP","internet"
Törlés (felül is írhatjuk az új értékkel):	AT+CGDCONT=x ahol x=1..4
Visszaellenőrzés:	AT+CGDCONT?
Válasz:	+CGDCONT: 1,"IP","internet","",0,0

Beállíthatjuk még a minimális és az elvárt szolgáltatási színvonalat is. Az előbbi azt a minimális adatátviteli sebességet jelenti, ami alatt nekünk nem felel meg a GPRS kapcsolat; az utóbbi pedig azt az ideális szintet, amit szeretnénk használni. Ez arra kell, hogy a GSM hálózat "lefoglalja" nekünk a kért mennyiségű szabad csatornát. Persze ha az időközben mások által bonyolított VOICE és CSD hívások lefoglalják előttünk a szabad slotokat, akkor nincs más megoldás, mi leszünk elsőnek kidobva! A két parancs: AT+CGQMIN és AT+CGQREQ, a tesztelés szempontjából ezek beállításának nincs jelentősége.

Következő tehát a GPRS kapcsolódás parancsa:	AT+CGATT=1
A GPRS kapcsolat állapotának lekérdezése:	AT+CGATT=?
A GPRS kapcsolat automatikus elindítása bekapcsoláskor:	AT%CGAATT=0,1

Ezzel a GPRS kapcsolatunk átlépett a DETACH-ed állapotból az ATTACH-ed állapotba, tehát figyelni kezdi a GPRS kapcsolódásra alkalmas tornyokat, és ahogy mozgunk, ugyanúgy kilép-belép, ahogy azt a GSM tornyokkal is teszi. Ez még teljesen ingyen van; ha nincs GPRS előfizetésünk, akkor is működik.

A következő lépés az AT+CGDCONT paranccsal definiált 4 lehetséges környezet elindítása. Ezt az AT+CGACT parancs végzi; az első paraméter a ki-be kapcsolás, a második a környezet száma 1-4-ig:

A GPRS csatlakozás parancsa az adott szerverhez:	AT+CGACT=1,1
A GPRS csatlakozás(ok) állapotának lekérdezése:	AT+CGACT=?
A GPRS csatlakozás automatikus elindítása bekapcsoláskor:	AT\$AREG=2

Ezzel a GPRS csatlakozásunk átlépett a DEACTIVATE-d állapotból az ACTIVATE-d állapotba, tehát felvette a kapcsolatot az általunk megadott AT+CGDCONT paranccsal definiált APN szerverrel. A szerver tehát él és működik, de ehhez már előfizetés kell, és még továbbra sem fizetünk érte. Így ebben az állapotban tetszőleges ideig lehetünk. Természetesen azután, hogy egy környezetet (=context) aktiváltunk, az AT+CGDCONT paranccsal azt az adatsort többet nem tudjuk módosítani, amíg az adott környezet aktív. Tehát ha pl. az AT+CGDCONT=1... kezdetű parancsra mániákusan ERROR-t kapunk, akkor nézzük meg az AT+CGACT=? paranccsal, hogy nem aktív-e a megadott környezet.

Végezetül elindíthatjuk a GPRS kapcsolatunkat is: ATD*99#
Ugyanez context megadással kombinálva: ATD*99***1#

A válasz garantáltan nem fog nekünk tetszeni:

```
CONNECT
~}#R!}!}!} }3}"&} } } }#}%A#}%}'"}{ }"W}0~
~}#R!}!}!} }3}"&} } } }#}%A#}%}'"}{ }"W}0~ ...
```

A magyarázat az, hogy ezek ún. PPP adatsomagok, amelyek emberi fogyasztásra alkalmatlanok, ámde nagyon alkalmasak adatok adott címre történő elküldésére. A **PPP** ugyanis a **Point-to-Point Protocol** rövidítése, azaz egy olyan protokollnak a neve, amely képes két adott IP című számítógép közötti adatátvitelre. És ezzel áttértünk egy másik nagy-nagy témakörre, a TCP/IP veremre (=stack)!

Az Internet ugyanis épp olyan összetett, mint a GPRS. Az Internet is szolgáltatási rétegekből áll, ezeket a rétegeket hívjuk TCP/IP rétegeknek, azok egymásra épülését pedig TCP/IP veremnek. A verem alján a fizikai adatátviteli közeg áll; ez esetünkben a GPRS, de ez lehetne egy ethernet hálózat, egy vezetékös modem soros vonala, stb. A verem tetején pedig van maga az Internet böngésző, ahol pl. beírom, hogy <http://www.google.hu/>, és már kereshetem is, ami érdekelt. Tehát innentől fogva kezeljük már közösen ezt a két dolgot, és nézzük meg, milyen rétegekből építkezik a mobil Internet, melyek mind az előző rétegre épülnek.

A legelső réteg tehát a fizikai adatátviteli közeg; ez GSM esetén lehet CSD hívás ill. olcsó GPRS kapcsolat. Eddig erről volt szó. (Nem említettük meg, hogy a GPRS alapszintje csak GPRS üzenetek fogadásából és küldéséből áll; nem fehér embernek való ez a szint. Az AT+CGEREP parancs mutatja meg ezeket az üzeneteket, amikre vagy válaszolunk az AT+CGANS paranccsal, vagy indítunk egy saját adatátvitelt az AT+CGDATA paranccsal. Néhány GSM modemet úgy reklámoznak, hogy GPRS képes, miközben csak a legalapvetőbb, szabványosított GPRS parancsokat ismeri.

A következő réteg a csomagkapcsolati réteg. Többféle protokoll ismert, PC-n a PPP protokoll, míg régi UNIX gépeken a SLIP (ill még a CLSIP) protokoll terjedt el. Ez utóbbit még nem ismeri az **Enfora**, de nincs is jelentősége. A protokoll leírásától itt eltekintenek, mert nem témája jelen írásnak, és a neten sokfelé meg is lehet találni. Ami a lényeg: szinte minden Internetezésre alkalmas készülék/szoftver ismeri a PPP alap-protokollt, így az **Enfora** GSM modemek bármilyen szoftverrel képesek Internet elérésére. A későbbiekben pl. a *Windows*[®] részének, a *HyperTerminál*[®]-nak a segítségével fogom megmutatni a GPRS alapú Internet kapcsolat létrehozását.

A következő réteg az UDP csomagok szintje. Az **User Datagram Protokol** a rövid és ellenőrzést nem igénylő adatsomagok szintje. Kb. olyasmi, mint az SMS, vagy a postai levél: Amikor feladjuk őket, nem tudjuk biztosan, hogy megérkeztek-e, ill. hogy azonos sorrendben érkeznek-e meg, mint ahogy mi feladtuk őket. Ezért ezzel bajos dolog nagy fájlokat átküldeni, de pl. tökéletesen alkalmas pl. GPS pozíciók, AT parancsok, egyedi mérési adatpontok küldésére. A csomagok ellenőrzése a mi dolgunk: kell valamiféle visszajelző UDP csomagot kapnunk, hogy az ellenállomás vette a csomagot, és arra az alábbi a válasza. Az **Enfora** GSM modemek az UDP csomagokat használják arra, hogy a GPRS kapcsolat ideje alatt AT-parancsokat adhassunk ki neki, akár lokálisan, akár távolról. Erről a PAD, a **Packet Assembler-Disassembler** nevű modulja gondoskodik; később még erről is részletesen szólunk.

A 4. réteg a TCP csomagok szintje. A **Transmission Control Protokol** már kifejezetten a nagy adatsomagok átviteli protokollja. Kezeli a nagy adatoknál szükséges darabolást, és gondosan ügyel arra, hogy a csomagok megfelelő sorrendben érkezzenek meg az ellenállomásra, ne legyenek hiányzó és dupla csomagok, stb. Kb. olyasmi, mint a CSD hívás mobilon, vagy mint egy csővezeték: amit az egyik végén betöltünk, az a másik végén változatlan formában kijön. A számítógépek címzése egy egyedi, négyjegyű azonosítószámmal, az ún. IP címmel történik (Pl. 198.162.1.1), és egy ún. socket számmal, ami egy 0-tól 65535-ig terjedő szám. (Ezért is hívják TCP/IP-nek is az Internetet.) Minden Internetes szolgáltatás erre a TCP szintre épül, vagy fordítva megfogalmazva TCP szinten bármilyen Internetes szolgáltatás megvalósítható.

Végezetül az utolsó, 5. szint már maga az Internet. Ahogy említettem, az Internet maga különálló szolgáltatások sorából áll, melyeknek mindnek van egy-egy alapértelmezett UDP/TCP port száma az alábbiak szerint, a teljesség igénye nélkül:

Szolgáltatás:	Port/Protocol:	Funkció:
smtp	25/tcp	E-mail küldés
pop3	110/tcp	E-mail fogadás
imap	?/tcp	Intelligens E-mail fogadás
http	80/tcp	WEB, http oldalak kezelése
ftp	21/tcp	FTP, fájlok küldése-fogadása
dns	53/tcp	IP cím meghatározása név alapján

A fenti táblázatból látható, hogy ha pl. egy beágyazott WEB szervert szeretnénk megvalósítani, akkor csak annyi a dolgunk, hogy csinálunk a 80 portra egy TCP protokollt használó alkalmazást, ami az Interneten bőségesen dokumentált **http** protokoll szerint **html** fájlokat küld el, amit majd a mi mobil alkalmazásunk fog előállítani a lokálisan rendelkezésére álló mérési vagy egyéb adatokból.

10. GPRS kapcsolódás Windows®-al

Ha már ennyire belejöttünk az Internetbe, ne is hagyjuk abba. Ahogy említettem, a *Windows® HyperTerminál®*-ja segítségével pillanatok alatt csatlakozhatunk az Internethez GPRS-en keresztül az alábbi módon. (Különösen arra jó ez, hogy leteszteljük, tényleg működik-e a kártyánkon a GPRS szolgáltatás, amelyet külön kell aktiválni a GSM szolgáltatóknál, és hát bizony ez keveredésre adhat okot...) Először is indítsuk újra a modemet, vagy kapcsoljuk ki a GPRS-t teljesen az `AT+CGATT=0` paranccsal, ha korábban a GPRS-el játszottunk, hogy minden biztosan az alaphelyzetben legyen. Újraindítás után adjuk meg a PIN kódot is, ha szükséges.

Elő lépésben hozzunk létre egy általános modemet, hogy a *HyperTerminál®* tudjon majd mivel kapcsolatot létesíteni. Ennek menete a következő *WIN98SE* magyar nyelvű operációs rendszer alatt: (A *WIN2000* magyar nyelvű operációs rendszer beállításai a fejezet második felében találhatóak meg!)

1. *Start* menü, *Beállítások*, *Vezérlőpult*, majd *Modemek* ikon
2. Klikkeljünk a *Hozzáadás* gombra.
3. "*Ne ismerje fel a modemet. Modem kijelölése listából.*" bejelölés bejelölése, majd *Tovább*.
4. Válasszuk ki a "*Szabványos modem típusok*" közül valami gyorsabbat, pl. a "*Szabványos 33.600 bps modem*"-et, majd *Tovább*.
5. Válasszuk ki a használt soros vonalat, ahova a tesztpanel is csatlakoztattuk, pl. "*Kommunikációs port (COM1)*", majd *Tovább*. **Vigyázat!** Ha azt a soros vonalat valamelyik program használja, akkor azt előbb zárjuk be!
6. Várjuk meg, míg a Windows feltelepíti a modemet, majd *Befejezés*.
7. Visszajutottunk a modemek menübe, és már van egy általános soros modemünk. *Bezárás*.

Második lépésben hozzunk létre egy telefonos kapcsolatot a fenti modemmel az alábbiak szerint:

1. Klikkeljünk az asztalon a *Sajátgép* ikonra.
2. Nyissuk ki a *Telefonos hálózat* rendszermappát.
3. Klikkeljünk az *Új csatlakozás* ikonra.
4. Adjuk meg a csatlakozás nevét, pl. *Enfora GPRS teszt*
5. *Válasszon eszközt* menüben válasszuk ki az imént létrehozott *Szabványos 33.600 bps modem*-et, majd *Tovább*.
6. A telefonszám mezőbe írjuk be, hogy **99#* (vagy **99***1#*), majd *Tovább*.
7. *Befejezés*, mire a Windows létrehozza az új kapcsolatunkat *Enfora GPRS teszt* néven.
8. Válasszuk ki az új kapcsolatunkat, majd jobb klikk, aztán *Tulajdonságok*.
9. Itt kapcsoljuk ki a *Körzetszám és tárcsázási tulajdonságok használata* bejelölését, mire csak a telefonszám mező marad meg, a többi beszűrül.
10. Válasszuk a *Kiszolgáltató típusok* fület, ott kapcsoljuk ki a *NetBEUI* és *IPX/SPX kompatibilis* hálózati protokollokat (így talán kicsit kevesebben fognak meghekkelni ☺), majd *OK*.
11. Ismételten válasszuk ki az új kapcsolatunkat, majd jobb klikk, aztán *Csatlakoztatás*.
12. A megjelenő menüben klikk a *Csatlakoztatás* gombra.

Ha ügyesek voltunk mi is, és a GSM szolgáltató emberei is megtették, ami a dolguk volt, akkor másodperceken belül már meg is jelenik a jobb alsó sarokban a két kis számítógép-monitor formájú piktogram, ami jelzi, hogy már él a GPRS alapú mobil Internet kapcsolatunk. Behívhatjuk a kedvenc böngészőprogramunkat, és beírhatunk egy tetszőleges WEB címet, garantáltan be fog jönni, csak egy kicsit (nagyon) lassú lesz a 1 GBit/sec-es céges csatlakozásunk után. ☺ A csatlakozás bontása a szokásos módon történik, a DCD LED világítása pedig jelzi annak sikeres megtörténtét. (Sajnos a LED-ek pont fordítva világitanak a tesztpanelen, mint az logikus lenne... ☹)

Mint látható, minden az alapértelmezés szerinti volt, semmilyen különleges beállítást nem kellett tennünk, mégis működik a GPRS alapú Internet. Ha nem működne, a hibajelzéseket az alábbi módon kérném: a *HyperTerminál*[®] által kiírt hibakód száma és szövege, AT&V parancs teljes válasza. Ha a modemet most újra programozni szeretnénk, ne ijedjünk meg, hogy nem látunk semmit, csak gyorsan kapcsoljuk vissza a Command Echo-t az ATE1 parancssal, mert a *HyperTerminál*[®] kikapcsolta!

A WIN2000 magyar nyelvű operációs rendszer alatt az alábbira módosul a beállítások menete: Első lépésben hozzunk létre egy általános modemet, hogy a *HyperTerminál*[®] tudjon majd mivel kapcsolatot létesíteni.

1. *Start* menü, *Beállítások*, *Vezérlőpult*, majd *Telefon és modem beállításai* ikon, azon belül a *Modemek* fül kiválasztása.
2. Klikkeljünk a *Hozzáadás* gombra.
3. "*Ne cizsgálja meg a modemet. Modem kijelölése listából.*" bejelölés beixelése, majd *Tovább*.
4. Válasszuk ki a "*(Szabványos modem típusok)*" közül valami gyorsabbat, pl. a "*Szabványos 33.600 bps modem*"-et, majd *Tovább*.
5. Válasszuk ki a használt soros vonalat, ahova a tesztpanel is csatlakoztattuk, pl. "*Kommunikációs port (COM1)*", majd *Tovább*. **Vigyázat!** Ha azt a soros vonalat valamelyik program használja, akkor azt előbb zárjuk be!
6. Várjuk meg, míg a Windows feltelepíti a modemet, majd *Befejezés*.
7. Visszajutottunk a modemek menübe, és már van egy általános soros modemünk. *OK*.

Második lépésben hozzunk létre egy telefonos kapcsolatot a fenti modemmel az alábbiak szerint:

1. *Start* menü, *Beállítások*, *Vezérlőpult*, majd *Hálózati és telefonos kapcsolatok* ikon.
2. Klikkeljünk az *Új kapcsolat létrehozása* ikonra.
3. A megjelenő *Új kapcsolat varázsló*-ban klikkeljünk a *Tovább* gombra.
4. Válasszuk ki a *Privát hálózat elérése telefonos hálózaton keresztül* sort, majd *Tovább* gomb.
5. *A Válassza ki a kapcsolathoz használni kívánt eszközöket:* beállításnál válassza ki a *Modem - Szabványos 33600 bps modem (COM1)* sort, majd *Tovább* gomb.
6. *A Telefonszám:* mezőbe írjuk be, hogy *99# (vagy *99***1#), majd *Tovább*.
7. *A kapcsolat létrehozása: Minden felhasználónak*, majd *Tovább*.
8. *Írja be a kapcsolat nevét:* pl. *Enfora GPRS teszt*
9. *Parancsikron elhelyezése az asztalon* beállítás beixelése.
10. *Befejezés*, mire a Windows létrehozza az új kapcsolatunkat *Enfora GPRS teszt* néven.
11. A szokások szerint a *Felhasználónév* és a *Jelszó* mezőket üresen kell hagyni, mert GPRS alatt a felhasználókat is a SIM kártya azonosítja, így nem kell külön jelszavas azonosítás, mint a vezetékes modemek esetében.

11. GPRS kapcsolódás - IP cím kiosztás

Murphy szerint ami jó csak akad az életben, az vagy törvénytelen, vagy erkölcstelen, vagy hízlal. Én kiegészíteném azzal, hogy "és kevés van belőle". Ilyen sajnos az IP cím is, amit annak idején arra találtak ki, hogy pár katonai bázis néhány száz számítógépét kösse össze. (Ez volt az ARPANET, az Internet őse.) Ezt követően elég pazarlóan osztották ki a címtartományokat, így mára, amikor már egy drágább kávéfőző is saját WEB szerverén jelzi a kávé hőfokát, bizony a statikus IP cím egy ritka, és így drága dolog lett. Nem volt más megoldás, mint az elfogyott IP címeket újra kiosztani, ami egy sor új problémát hozott be az életbe. Ezek az ún. dinamikus IP címek. (Helyesbítenék: a statikus IP címek egy csoportja felhasználható lenne oly módon is, hogy mindig csak azoknak a felhasználónak osztanák ki, akik éppen Internetezni szeretnének; azonban a tapasztalatok szerint egyszerűbb ezt más módon megoldani, ezért nem is ezt a módszert alkalmazzák a gyakorlatban.)

Az újra kiosztott IP címek legnagyobb problémája, hogy egy IP cím azonosít egy számítógépet az egész világon. Ha kettő van belőle, már nem tudnánk azonosítani, melyik gépről is van szó a kettő közül. Ezért az újra kiosztott IP címeket csak magánhálózatokon, ill. az Internettől fizikailag külön álló hálózatokon szabad alkalmazni, a két hálózat közötti csatlakozási ponton pedig ún. router-eket, útvonal-választó eszközöket kell használni. Ezek az útvonal-választó eszközök a belső hálózatról érkező kérésekben kicserélik az IP címet a sajátjukra, majd elküldik a maguk nevében a külső hálózat irányába. A külső hálózatról kapott válaszban pedig visszacseréli az IP címet az eredetire, lévén az IP csomagok tartalmaznak olyan információkat, melyek alapján megállapítható, hogy ez a válasz melyik kérésre jött válaszul. Ez az IP cím csere a NAT, azaz a **N**etwork **A**ddress **T**ranslation. Ez a módszer nagyon jól működik mindaddig, amíg a belső hálózat irányából jönnek a kérések, hiszen mindegyik számítógép azt hiszi, hogy ő direktben kommunikál a külső hálózattal. Azonban ha kívülről meg szeretnénk szólítani a belső hálózat bármelyik gépét, ezt nem tehetjük meg, hiszen mi csak a router IP címét látjuk; sőt, még azt sem tudhatjuk, hogy a belső hálózaton szereplő számítógépek milyen IP címeket használnak. Nem beszélve arról, hogy az Interneten annyi rosszindulatú támadás - betörési- és adatlopási kísérlet - van, hogy minden egyes Internetre csatlakozó eszköz - így a router is - egy tűzfal (Firewall) mögé van dugva, ami direkt nem engedi meg, hogy kívülről, előzetes kérés nélkül ott adatcsomagok tudjanak bemenni.

Ezt a bevezetőt csak azért írtam le, mert a GSM modemeken zajló GPRS hálózat általában pont ugyanilyen belső hálózat, amelyet az APN szerver - mint router és tűzfal egy személyben - köt össze az Internettel. Bár technikailag meg lehetne úgy is oldani, hogy van egy belső hálózat, amely csak egyetlen egy ponton csatlakozik az Internetre, mégis biztonsági és gazdasági okok miatt inkább több kapcsolódási pont van. A **T-Mobile** GPRS szolgáltatásánál pl. két egymás mellett lévő, dinamikus IP című modem sem látja egymást, hanem mindegyik csak a router IP címét látja a másik modem saját IP címe helyett. A **Pannon GSM** GPRS-e ilyen szempontból már korrektebb, ott a dinamikus IP című kártyák is látják egymást. Tehát amit eddig és eztán leírtam a modemek közötti UDP és TCP kommunikációról, az nem mindig működik a gyakorlatban. Pár biztos megoldás van csak:

1. Kérhetünk - felár ellenében persze - statikus IP címet a GPRS kapcsolatunkhoz, mely esetben nincs route-olás, címfordítás (NAT) és így minden úgy működik, ahogy eddig leírtam. Sajnos ezt minden olyan modemre meg kell tennünk, amelyiket önállóan meg szeretnénk szólítani, így egyenrangú hálózat esetén, ahol bármelyik modem beszélgethet bármelyik másik modemmel, ez jelentős többletköltséget jelent. Emiatt max. saját, céges APN-nél szokták használni, amit a következő pontban írok le, hogy egyszerűsítsék a GSM-ek nyilvántartását és beazonosítását.

(Innentől fogva ugyanis a statikus, fix IP cím azonos lesz egyfajta egyedi telefonszámmal.)

2. Kérhetünk céges magánhálózatot - azaz saját APN szervert -, ebben az esetben a saját magánhálózatunkon sincs route-olás, címfordítás (NAT) és így minden úgy működik, ahogy leírtam. Ebben az esetben azonban az Internetről nem tudjuk elérni a modemjeinket, és a modemekről sem tudjuk elérni az Internetet. Ez sok esetben nem hátrány, hanem előny, hiszen nem kell az adatlopástól ill. az illegális behatolási kísérletektől tartanunk, ezért ez az elterjedt megoldás. A címkiosztás egy ilyen hálózaton lehet az olcsóbb, dinamikus IP kiosztás, mely esetben nekünk kell a modemek azonosítását megoldanunk, de lehet a drágább statikus IP cím is.
3. Végezetül ha beérjük egy kliens-szerver konfigurációval, ahol az egyes modemek nem látják egymást, csak a szervert, továbbá csak a kliensek indítanak kéréseket a szerver felé, akkor az is elégséges megoldás, ha csak a szerver(ek) számára kérünk statikus IP címe(ke)t. A módszer előnye, hogy maga a szerver gép lehet egy fixen telepített, védett asztali számítógép is, amely nem a bizonytalan térerejű, GSM zavaróval blokkolható GSM hálózatot használja, hanem bérelt vonali, nagy sebességű Internet-kapcsolattal rendelkezik, akár több irányba is, így maga a központi gép nem blokkolható, és akár egyedül képes lekezelni több tízezer egyidejű GSM kommunikációt is. Mivel az internetes szerverek számára a statikus IP cím kötelező, így egy nem GPRS alapú statikus IP cím fenntartása olcsóbb is, és lehet hozzá saját DNS nevet kérni. Ekkor már csak egyetlen egy probléma vár megoldásra: hogyan azonosítsuk a modemjeinket? A következőkben erről lesz szó...

Először is meg kell állapítanunk, hogy a kapott SIM kártyához milyen IP címet állítottak be - mert sajnos néha ezt a GSM szolgáltatók sem oldják meg elsöre és tökéletesen, így bizony tesztelni illik. Ehhez nem kell mást tenni, mint a 10. fejezetben leírt módon, GPRS-en keresztül fel kell jelentkezni az internetre, majd ha ez sikerült, kiadni *Windows*[®] 98 alatt az **IPCONFIG** parancsot pl. a *Total Commander*-ben vagy az *MS-DOS parancssor* futtatása után, és az kiírja a mi IP címünket az IP-cím sorban. *Windows*[®] XP esetén a jobb alsó sarokban látható *Kapcsolat* ikonra kattintva előjön a kapcsolat állapotának menüje, és ott is kiírja valahol a mi IP címünket. Ezt írjuk fel, majd menjünk el a <http://www.ip-adress.com/> oldalra. (Vigyázat, az *address*-ben csak egy D betű van szándékosan!) Ez az oldal kiírja, hogy az Internet felől milyen IP cím alatt látnak minket. (Ha esetleg ez az oldal épp nem működne, a *Google*-ba beírt *My IP address* kérdésre számszámra találunk olyan találatot, ahol az IP címünket kiírják.)

Ha ez a két IP cím – a helyi és a távoli – nem azonos, akkor mi egy NAT, azaz egy IP címfordítás mögött vagyunk, így esélyünk sincs a GPRS modemünknek távolról üzenetet küldeni, max. mi kezdeményezhetünk TCP kapcsolatot egy másik, nem NAT-os IP-címmel. Ilyen pl. a **T-Mobile** GPRS szolgáltatása; ott nem tudunk a GPRS modemek között beszélgetni, még akkor sem, ha az IP címüket ismerjük. A Pannon GSM GPRS modemek viszont nincsenek NAT-olva a tesztek szerint.

Ezt követően bontsuk el az Internet kapcsolatot, majd egy perc múlva csatlakozzunk újra, és ekkor is kérdezzük le az **IPCONFIG** paranccsal a saját IP címünket. Ha az IP címünk ugyanaz, amit az előbb felírtunk, akkor nagy valószínűséggel fix IP címünk van; ez nagy mértékben megkönnyíti a munkánkat, mert a jövőben az eddigi telefonszám helyett az új, fix IP cím alapján tudjuk azonosítani a modemjeinket.

Fontos megérteni a NAT és a fix IP cím közötti különbséget, ezért ismételten elmagyarázom: egy GPRS modemnek lehet dinamikus és fix IP címe, ill. valódi IP-címe és NAT-olt (lokális) IP címe. Ez 4 különböző kombinációt jelent:

A NAT-olt IP cím a legrosszabb: távolról a modemet nem tudjuk megszólítani, max. a modemről indíthatunk TCP kapcsolatot a távoli irányba. Ilyen szempontból mindegy, hogy a címünk dinamikus vagy fix IP cím. Ilyenek a **T-Mobile** GPRS előfizetései.

Ha az IP címünk valódi, fix IP cím, akkor bárki és bárhonnán elérhet minket; kezdeményezhetünk és fogadhatunk UDP és/vagy TCP kapcsolatokat bárkivel. Viszont a pénztárcánkat ez az állapot meg fogja viselni...

A legoptimálisabb az, ha az IP címünk valódi, ámde dinamikusan kiosztott: ahányszor belépünk, annyiszor kapunk egy új címet egy adott tartományból. Ilyenek a **Pannon GSM** GPRS előfizetései. Itt már csak egy dolgot kellene megoldanunk: a mi saját IP címünket valahogy mások tudomására kellene hozni minden egyes bekapcsoláskor, hogy tudjanak velünk mások is kommunikálni. Ebben az **Enfora** verhetetlen: szabadalmaztatott technikájuknak köszönhetően önállóan képesek egy IP címtár jellegű szolgáltatásra felcsatlakozni minden GPRS csatlakozáskor, oda elküldeni a mostani dinamikus IP címüket, az esetleges nyugtákat kezelni, ill. ha kérjük, bizonyos időnként még "életjelet" is adni.

Folytatjuk...

12. GPRS kapcsolódás - UDP

Az UDP kommunikáció a leghasznosabb szolgáltatás az **Enfora** modemeknél, ámde a leginkább összetett is. Röviden és tömören egy olyan módszer, amellyel - az SMS-hez hasonlóan - rövid, max. 512 byte-os üzeneteket tudunk küldeni automatikusan a szerver(ek) felé, vagy visszafelé. Ezek az üzenetek lehetnek GPS koordináták, események (I/O aktivitás, limit-túllépés, stb.), ill. válasz-üzenetek a szerver felől küldött AT parancsokra. Kezdjük ez utóbbival.

Az előzőekben láthattuk, hogy tudunk PPP kapcsolatot létrehozni a *Windows® HyperTerminál®*-ja segítségével. Most erre lesz szükségünk, és egy apró kis segédprogramocskára, a *HW-Group* nevű német cég egyik ingyenes (freeware) szoftverére, a *Hercules SETUP* programra, amelyről az alábbi linken olvashatunk: http://www.hw-group.com/products/hercules/index_en.html ill. az alábbi linkről tudjuk letölteni: <http://www.hw-group.com/download/sw/HerculesSetup.zip> (~ 500 kByte ZIP-ben, kibontás után ~ 1,2 MByte, telepíteni nem kell, csak elindítani.) Bár a program nem kifejezetten a mi céljainkra való, és sok - számunkra - felesleges funkció is van benne, de a kis mérete és egyszerű installja miatt választottam pont ezt. Bármilyen másik programot is választhatunk, amely képes adott tartalmú IP/UDP csomagokat készíteni. Próbáljunk ki egy másik **Enfora** specialitást, az ún. local PPP üzemmódot. Ekkor GPRS kommunikációra alkalmatlan SIM kártyával is ki tudjuk próbálni az eddig leírtakat. Küldjük el a modemnek az alábbi parancsot: `AT$HOSTIF=3` Ezzel bekapcsoljuk a local PPP üzemmódot a modemem, majd a korábban *Enfora GPRS teszt* névvel létrehozott GPRS kapcsolatunkat csatlakoztatva létrejön egy virtuális PPP kapcsolat, amely Internettel való kapcsolatra nem, de a modemünkkel történő kommunikációra már alkalmas. Indítsuk el a *Hercules SETUP*-ot, és válasszuk ki az *UDP* fület. Itt írjuk be az *UDP Module IP* helyre a 199.245.180.013 címet, a *Port* helyre a 1720 portcímét. A többi beállítás lényegtelen; majd klikkeljünk a *Listen* gombra, ami *Close* feliratúra változik. A használt 199.245.180.013/1720 IP/portcím az *Enfora* modemek alapértelmezett API portja, amit amúgy az `AT$UDPAPI` paranccsal tudunk megváltoztatni, de nincs rá szükség. Az alsó Send ablak első sorába írjuk be az alábbi adatot, majd klikkeljünk a mellette lévő Send gombra:

```
$00$00$04$00ATI<CR><LF>
```

Az alsó, *Sent Data* feliratú ablakban megjelent az imént kiküldött `ATI` parancs, míg a felső, *Received data* feliratú ablakban megjelent a válasz: *Enfora, L.P. OK* Ezzel kiküldtünk egy parancsot az API-nak UDP csomagként, majd a választ szintén UDP csomagként visszakaptunk. Közben akár internetezhettük volna, ha az `AT$HOSTIF=3` helyett a `AT$HOSTIF=0` parancsot adtuk volna ki korábban. A kiküldött adatsomag első 4 byte-ja speciális, ezeket ezért hexadecimális karakterekként küldtük ki, ezt jelzi a \$ jel: az első két byte értéke lényegtelen, a harmadik byte 4-es értéke jelzi, hogy a kiküldött UDP csomag AT parancsot jelent, a negyedik byte kötelezően 0 értékű. A kapott válasz UDP csomag szintén tartalmazza ezt a 4 byte-os fejléceket, csak sajnos ezt a *Hercules SETUP* levágja. Ha nem tenné, akkor látnánk, hogy ott a harmadik byte értéke 4-ről 5-re változott, ami az AT parancs választ tartalmazó UDP csomagot jelzi. Mindig ez a harmadik byte mutatja meg, hogy a csomag mit tartalmaz. Kipróbálhatjuk még az alábbi parancsokat is (Ne felejtjük el, itt a \$ jel a hexa-decimális karakterek jelzője; ezért ha mint \$ jelet akarjuk használni, akkor dupláznunk kell!):

Lekérdezzük, hogy mi a mi IP címünk:

```
Send:      $00$00$04$00AT$NETIP?<CR><LF>
Válasz:    $NETIP: "000.000.000.000", "000.000.000.000", "000.000.000.000"
```

Nincs hálózati IP címünk. Akkor mi a local (virtuális) IP címünk?

Send: \$00\$00\$04\$00AT\$\$LOCIP?<CR><LF>
Válasz: \$LOCIP: "001.002.003.004", "002.003.004.001", "003.004.001.002"

Csatlakozzunk fel a hálózatra:

Send: \$00\$00\$04\$00AT\$\$CONN<CR><LF>
Válasz: CONNECT

Sikerült! Mi most az IP címünk? (Közben akár internetezhetünk is, míg ezt kiadjuk...)

Send: \$00\$00\$04\$00AT\$\$NETIP?<CR><LF>
Válasz: \$NETIP: "123.123.123.123", "123.123.123.123", "123.123.123.123"

Kapcsolódjunk le a hálózatról:

Send: \$00\$00\$04\$00AT\$\$DISC<CR><LF>
Válasz: NO CARRIER

Sikerült! Mi most az IP címünk?

Send: \$00\$00\$04\$00AT\$\$NETIP?<CR><LF>
Válasz: \$NETIP: "000.000.000.000", "000.000.000.000", "000.000.000.000"

Tehát újra a local PPP kapcsolatban vagyunk.

Az eddigiekben tehát saját magunkkal tudtunk beszélgetni. Ennek főleg akkor lehet haszna, ha már GPRS/UDP kapcsolatban vagyunk, és szeretnénk pár modem-parancsot kiadni. Ez más modemeken úgy szokott történni, hogy +++-al bontjuk a DATA módot, és átlépünk COMMAND módba, majd ott kiadjuk a kívánt parancsot, majd visszalépünk DATA módba, és folytatjuk a kommunikációt. Sajnos az **Enfora** modemek esetében más a koncepció: a +++ nem COMMAND módra vált, hanem ténylegesen bontja a GPRS/UDP vagy GPRS/TCP kapcsolatot. Ezenkívül használhatjuk még a CMUX üzemmódot is, sőt igazából ma már a CMUX üzemmód használata a preferált, ha üzem közben modem-parancsokat szeretnénk kiadni. Erről lásd jelen dokumentáció legvégén a CMUX fejezetet.

Sokkal lényegesebb, hogy más modemekkel tudjunk kommunikálni, sőt, akár távoli parancsokat is ki tudjunk adni. Ehhez nem kell mást tenni, mint a megcélzott modemem beállítani az AT\$PADSRC paranccsal egy tetszőleges, általunk választott portszámot – két érték kivételével. Az első a nullás érték, ez ugyanis a TCP/IP kommunikációban tiltott, tehát ezzel tudjuk letiltani, hogy mások távolról a GSM modemünket megszólíthassák. A másik tiltott érték az AT\$UPDAPI portcíme, amivel az előbb is játszottunk. Ennek alapértéke 1720, tehát ezt se használjuk. A fentiek után tehát adjuk ki mondjuk az AT\$PADSRC=1234 parancsot mindkét modemem: azon is, amelyiket UDP-n hívni akarjuk, különben nem tudjuk megszólítani; továbbá a hívón is, különben a visszaérkező választ meg mi nem fogjuk látni. Ezt követően indítsuk el a PPP kapcsolatot a **hívott** modemünkön, majd a *Hercules SETUP*-ot is, ott válasszuk ki az *UDP* fület. Itt írjuk be az *UDP Module IP* helyre a **hívó** modem IP címét, a *Port* helyre az előbb választott 1234 portcímet, az eddig nem használt *Local port* helyre pedig szintén 1234 kerül. A többi beállítás lényegtelen; majd klikkeljünk a *Listen* gombra, ami *Close* feliratúra változik.

Ezután a **hívó** modemem állítsuk be az UDP protokoll szerinti üzemmódot az `AT$HOSTIF=1` paranccsal. Ezt követően indíthatjuk az UDP kapcsolat aktiválását, mely során a hívó modem is fellép a GPRS hálózatra, és az IP cím alapján megkeresi a hívott modemet. (Az UDP kommunikációban nincs szerver és kliens oldal; a hívó és hívott elnevezést is csak az egyes modemek könnyebb azonosítása miatt használom. Az UDP ugyanis egyenrangú felek kommunikációja, mint ahogy SMS küldésnél is csak a küldés pillanatában különböztetünk meg SMS-t küldő és SMS-t fogadó modemet, egy pillanat múlva az SMS küldés iránya is megfordulhat.) Az IP cím megadására kétféle módszer is létezik: vagy az `AT$PADDST` paranccsal adhatjuk meg a tárcsázás előtt, vagy magával a tárcsázó paranccsal, mintha csak egy telefonszámot adnánk meg. (Ez természetesen fix, ún. statikus IP címeket feltételez; sajnos a valóságban azonban dinamikus IP címeket szokás használni; az ezzel kapcsolatos tudnivalókról külön fejezet fog szólni.) Az első módszer:

Előbb beállítjuk a hívott oldal IP címét: `AT$PADDST="123.123.123.123",1234`
Majd elindíthatjuk a GPRS-t a "szokásos" tárcsázó paranccsal: `ATD#99*`
Vagy GPRS csatlakozás automatikus elindítása bekapcsoláskor: `AT$AREG=2`

A második módszer:

Tárcsázunk az **Enfora** saját tárcsázó utasításával: `ATDP 123.123.123.123/1234`

A válasz rövidesen megérkezik: `CONNECT` ha sikerült, vagy
`NO CARRIER` ha nem sikerült.

Ezt követően amit láthatatlanul beírunk a **hívó** oldal terminál képernyőjén, az megjelenik a **hívott** oldalon, esetünkben a *Hercules SETUP* program felső, *Received data* képernyőjén; amit pedig beírunk az alsó, *Sent data* képernyőn, az megjelenik a **hívó** oldal terminál képernyőjén.

Most csináljunk egy kis trükköt – nem véletlenül használjuk épp a *Hercules SETUP* programot az egyik oldalon. Megcsinálhattunk volna az UDP kapcsolatot két *Windows® HyperTerminál®* között is, két külön soros vonal és két GSM modem segítségével. A *Hercules SETUP* jobb felső sarkában látható *Close* felírra klikkelve az visszaváltozik *Listen*-re, így már átírhatjuk a Port helyén látható 1234 címet 1720-ra, majd ismét klikkeljünk a *Listen* gombra, ami *Close* felírrá változik. Most küldük el az alsó *Send* sorban már korábban beírt alábbi parancsot a *Send* gombbal:

Send: `$00$00$04$00AT$NETIP?<CR><LF>`
Válasz: `$NETIP: "123.123.123.123", "000.000.000.000", "000.000.000.000"`

A válaszban szereplő IP cím már nem a **hívott** modemé, hanem a **hívó** modem IP címe! A kiküldött parancs ugyanis most nem az 1234-es portcímre ment, hanem a 1720-asra, ami a modem API portja, és észlelve, hogy érvényes parancsot kapott, a választ is ugyanarra az IP címre küldi vissza, ahonnan a kérést is kapta. Így tudunk tehát távoli parancsokat kiadni egy modemnek, és ezért nem lehetett azonos az `AT$UDPAPI`-ban szereplő 1720-as portcím, és az `AT$PADSRC`-ben általunk önkényesen választott portcím: ha ugyanis azonos lenne, akkor a modem nem tudná, hogy távoli parancsot kapott, vagy csak adatot.

Az UDP csatlakozás bontása természetesen a megszokott módon, a +++ karakterekkel vagy a DTR vonallal történik, és a *NO CARRIER* üzenet megjelenése jelzi, hogy az UDP kapcsolat megszakadt. Ezt – mivel az UDP csatlakozás egyenrangú – mind a két oldalon meg kell tenni, mert a másik oldal nem bont automatikusan. Nincs is mit: az UDP csomagkapcsolt adatátvitel – csak egy adatsomag átküldési idejére van kapcsolat a két oldal között; a következő pillanatban akár másik címről is kaphat adatokat.

Akinek esetleg az előző játékomban kicsit bonyolult volt, annak itt egy egyszerűbb: nem kell hozzá más, mint két *Windows® HyperTerminál®*, két külön soros vonal és két GSM modem – és persze két fix IP című SIM kártya.

Az alábbi parancsokat kell kiadni **mindkét** modemen:

AT+CGDCONT=... (Ha eddig még nem tettük volna meg...)
AT\$HOSTIF=1 (UDP protokoll beállítása)
AT\$PADSRC=XXXXX Választunk egy tetszőleges portcímet 1–65535 között, kivéve a 1720-at.

Ez után már jöhet is a csatlakozás. Természetesen mindegyik modemen a **másik** modem IP-címét kell megadni, különben nem fog működni semmi.

Az első módszer:

Előbb beállítjuk a másik modem IP címét: AT\$PADDST="123.123.123.123",XXXXX
Majd elindíthatjuk a GPRS-t a "szokásos" tárcsázó paranccsal: ATD#99*
Vagy GPRS csatlakozás automatikus elindítása bekapcsoláskor: AT\$AREG=2

A második módszer:

Tárcsázunk az **Enfora** saját tárcsázó utasításával: ATDP 123.123.123.123/XXXXX

A válasz rövidesen megérkezik: *CONNECT* ha sikerült, vagy
NO CARRIER ha nem sikerült.

Ha ezt mindkét modemen megcsináltuk, akkor amit az egyik ablakban beírunk, azt a másik ablakban látjuk. A csatlakozás bontása a megszokott módon, a +++ karakterekkel vagy a DTR vonallal történik.

A bátrabbak választhatnak két különböző portcímet is a két oldalon. Ebben az esetben is létre fog jönni UDP kapcsolat a két modem között, de ehhez az kell, hogy az AT\$PADSRC és AT\$PADDST portcímeiket jól állítsuk be. Azaz ha az egyik oldalon az AT\$PADSRC-ben beállítunk "A"-t, akkor a másik oldalon az AT\$PADDST-ben kell "A"-nak állnia, a másik oldali AT\$PADSRC-ben viszont "B"-nek kell állnia, akárcsak az első oldali AT\$PADDST-ben. Természetesen az AT\$PADDST alkalmazása helyett mindenhol használhatjuk a sokkal logikusabb ATDP... kezdetű tárcsázó parancs teljes alakját is, ami egyszerűen csak felülírja az AT\$PADDST-ben lévő értékeket.

13. GPRS kapcsolódás - TCP

Ahogy arról már többször szó volt, a GPRS alatti TCP kapcsolat megfelel a hagyományos modem CSD üzemmódjának: a modem ugyanis egy olyan környezetet hoz létre, amely lekezeli mindazokat a rendszereeményeket, amelyek ahhoz szükségesek, hogy az általunk feladott adatok a címzetthez az eredetivel megegyező módon megérkezzenek. Tehát miután felépítettünk egy TCP kapcsolatot, a modem egy "csövet" épít ki az ellenállomásig: amit és ahogy betöltünk ebbe a csőbe, az és úgy fog a cső túoldalán kijönni. Lehet, hogy nem azonnal - mert éppen valami adatátviteli hiba miatt megakad a kommunikáció, és csak másodpercek múlva indul újra - de ki fog jönni minden! Ha a hibajavítás nem oldható meg valami permanens hiba miatt, akkor maga a kapcsolat fog megszakadni ugyanúgy, mint egy CSD hívásnál - és mi nem fogjuk tudni, hogy mi okozta a hibát. Ilyen szempontból a TCP socket tökéletesen alkalmas pl. gyűjtött mérési adatok időszaki távkiolvasására, de nem előnyös pl. a riasztórendszerekben, ahol egy szabotázs (pl. a GSM antenna letörése) oly mértékben leronthatja az adatátviteli képességeket, hogy bár néha-néha sikeresen kimegy egy adatcsomag, de a nyugta már nem érkezik vissza, így a TCP socket nem mutat meg egyetlen átvitt adatyte-ot sem, miközben lehet, hogy már többször sikeresen elküldte a riasztórendszer a legutolsó GPS koordinátákat, csak nyugta és ellenőrzés nélkül ezt a rendszer eldobja. Tehát ilyen szituációk felmerülésekor az UDP csomagok jelentik a jó megoldást - azon az áron, hogy a nyugtákat nekünk kell majd visszaküldözgetnünk.

A másik fontos megjegyzés a TCP socket-tel kapcsolatban az, hogy az Enfora GSM modemek nem teszik lehetővé a GPRS kommunikáció felfüggesztését pl. egy AT parancs kiadásának idejére. A vonalbontás parancs (+++ vagy DTR jel) ugyanis mindig bontja a TCP kapcsolatot, és vele együtt a GPRS kapcsolatot is. De mivel a GPRS kapcsolatnál nincs kapcsolási díj, a fel-le kapcsolódás nem okoz többletkiadást, tehát országon belül ez is járható út. Azonban országon kívül, roaming esetén ez már nincs így; ott lényeges a GPRS kapcsolat folyamatos fenntartása. Ilyen esetben egyedül a CMUX technika használható, amely max. 6 db virtuális soros vonalat hoz létre egyetlen egy fizikai soros vonalon keresztül, és ezek a virtuális soros vonalak mint külön-külön modemek funkcionálnak: amíg az egyiket él egy TCP kapcsolat, a másikon SMS-eket küldhetünk-fogadhatunk, a harmadikon nézhetjük a pillanatnyi térerőt AT-parancsokkal, míg a negyediken vehetjük a bejövő VOICE hívás hívószám-kijelzését. Erről részletesen lásd az erről szóló későbbi fejezetet.

A továbbiak jobb megértéséhez nézzük röviden, hogyan is működik a CSD hívás: a hívott oldalon először is be kell állítani, hogy automatikusan vegye fel a beérkező hívást valahányadik csengetésre a modem; ezt az AT\$S0=x paranccsal tudjuk megtenni. Ezt követően a hívó modemem kiadjuk a tárcsázás parancsot, majd sikeres kapcsolódás esetén megjelenik a CONNECT szöveg mindkét oldalon, és onnantól fogva amit beírnak az egyik modemem, azt viszont látjuk a másikon és fordítva. (Sajátunkon természetesen nem, mert itt nincs ECHO funkció, mint a COMMAND módnál.) A hívás végét bármelyik oldalon kezdeményezhetjük a +++ karakterek kiadásával, vagy hardveresen a DTR vonal lekapcsolásával, ha az AT&D2 konfiguráló parancsot korábban már kiadtuk.

A helyzet pontosan ugyanez TCP socket esetén is, csak a parancsok kicsit mások. Először is be kell kapcsolnunk a TCP socket üzemmódot mindkét modemem, majd az egyiket vételre kell állítani:

Hívó oldal:

AT\$HOSTIF=2 (TCP socket üzemmód)

AT\$ACTIVE=1 (Aktív/kliens mód)

Hívott oldal:

AT\$HOSTIF=2 (TCP socket üzemmód)

AT\$ACTIVE=0 (Passzív/szerver mód)

A második lépés a TCP socket aktiválása a hívó oldalán, mely során a modem fellép a GPRS hálózatra, és figyelni kezdi az adatátvitelt kezdeményező aktív, kliens oldal kéréseit. A TCP socket ún. portszámokkal dolgozik, azaz minden egyes IP címhez tartozik 65535 lehetséges portcím, amely egy-egy különálló, független és egyidejű kommunikációs csatornát jelent; erre az Internetnél szükség is van, hiszen egyszerre több program is futhat a számítógépen. Az **Enfora** modemek esetén azonban csak egy csatorna lehet aktív egyszerre. A csatorna száma teljesen tetszőleges a nullát kivéve, ami az összes lehetséges csatornát jelenti. Ha Internetre csatlakozunk, akkor azonban célszerű az első 1024 csatornát nem használni saját célra, mert azok az Interneten általában definiált funkcióval bírnak, és esetleg problémát okozhatunk vele. Az `AT$PADSRC` paranccsal adhatjuk meg, hogy melyik portot kívánjuk figyelni; a nullás értéke bármely, a mi IP címünkre szóló kérést elfogad. Ez az alapeset. Azt, hogy milyen IP címről fogadunk el hívást, azt pedig a `AT$PADDST` paranccsal adhatjuk meg, külön az IP címet, és külön a portcímet is. Mindkettőnél használható a nullás "joker" szám, sőt, az IP címet megadhatjuk tartományként is, pl. 198.162.1.0 formában, amely az összes 198.162.1... kezdetű IP címet jelenti. Természetesen itt is a nullás beállítás az alapértelmezett.

A harmadik lépés magának a csatlakozásnak az elindítása. Erre háromféle módszer létezik:

Elindíthatjuk a GPRS-t a "szokásos" tárcsázó paranccsal: `ATD#99*`
Elindíthatjuk az **Enfora** saját GPRS tárcsázó utasításával is: `ATDT`
A GPRS csatlakozás automatikus elindítása bekapcsoláskor: `AT$AREG=2`

A válasz rövidesen megérkezik: *LISTEN*

Ezt követően indíthatjuk a TCP socket aktiválását a hívó oldalon, mely során a hívó modem is fellép a GPRS hálózatra, és az IP cím alapján megkeresi az adatátvitelt kiszolgáló szerver modemet. Az IP cím megadására kétféle módszer is létezik: vagy az `AT$PADDST` paranccsal adhatjuk meg a tárcsázás előtt, vagy magával a tárcsázó paranccsal, mintha csak egy telefonszámot adnánk meg. (Ez természetesen fix, ún. statikus IP címeket feltételez; a valóságban azonban dinamikus IP címeket szokás használni; az ezzel kapcsolatos tudnivalókról külön fejezet fog szólni.) Az első módszer:

Előbb beállítjuk a szerver IP címét: `AT$PADDST="67.103.214.214",1728`
Majd elindíthatjuk a GPRS-t a "szokásos" tárcsázó paranccsal: `ATD#99*`
Vagy GPRS csatlakozás automatikus elindítása bekapcsoláskor: `AT$AREG=2`

A második módszer:

Tárcsázunk az **Enfora** saját tárcsázó utasításával: `ATDT 67.103.214.214/1728`

A válasz rövidesen megérkezik: *CONNECT* ha sikerült, vagy
NO CARRIER ha nem sikerült.

A TCP socket bontása a megszokott módon, a +++ karakterekkel vagy a DTR vonallal történik, és a *NO CARRIER* üzenet megjelenése jelzi mindkét oldalon, hogy a TCP kapcsolat megszakadt.

A fenti példákban szereplő IP cím speciális: ez az **Enfora** cég TCP teszt-szerverének az IP címe, mely mindig működik, és nem csinál mást, mint kiír egy üdvözlő szöveget a mi IP címünkkel, majd az összes neki elküldött adatot egy fejléc után változatlan formában visszaküldi az alábbiak szerint:

```
>> You are IP 123.123.123.123 port 12345  
>> 123.123.123.123/12345> A mi üzenetsorunk, amit Enterrel zártunk le, vagy vártunk 5 sec-et.
```

14. EDGE modemek

Az EDGE technológiáról röviden annyit kell tudni, hogy míg a GPRS kapcsolat csak a GSM hálózat szabad csatornáit használta ki, addig az EDGE modemeknél magát a GSM hálózatot is elég érdekes módon megváltoztatták. Az alap GSM ugyanis 200 KHz-es frekvenciával küldi ki a jeleket, ahol 1 jel-idő 1 bitet jelent; ez mára kissé elavult és gazdaságtalan megoldás. Az EDGE technológia egy újfajta kódolási eljárással 1 jel-idő alatt 3 bitnyi információt képes átküldeni, tehát helyből tripla lesz az adatátviteli sebesség. Ehhez természetesen mind a GSM modemnek, mind pedig a GSM toronynak ismernie kell ezt az új kódolási rendszert. Ezért kell az EDGE technológiához teljesen más - és kissé drágább - GSM modem, és ezért korlátozódik az EDGE hozzáférés jelenleg még egy szűk vételem területre. A háromszoros adatátviteli sebesség különösen a GPRS kommunikációban jelent egy nagy sebesség-ugrást, ezért a két fogalom szorosan összekapcsolódik.

Az EDGE modemek szinte teljesen lábkompatibilisek a többi *Enabler-II* sorozatba tartozó Enfora modemmel, két apró eltérés mégis van: az egyik a 40. láb, amely általában nem használt, így gondot sem okoz; míg a másik a 22. láb, amely a többi Enfora modemnél a Remote ON láb, itt viszont a Reset láb lett. Ez gyakorlatilag "csak" egy szintváltást jelent, ami miatt a fejlesztői kártyán az SW203 kapcsolót pont fordított állásba kell kapcsolni, ahogy ez a 2. fejezetben, ill. az *Enfora Enabler II-G to Enabler II-E Transition Guide EDG0100PB002MAN.pdf* dokumentumban is le van írva.

Sokkal lényegesebb változás, hogy bár bejött pár 3G mobilszolgáltatásokkal kapcsolatos parancs, az Enfora modemek "esszenciájának" számító PAD/UDP/Event Engine/Dinamic IP kezelő parancscsoport még nincsen megvalósítva az EDGE modemeken. Ezért főleg olyan helyre javasolt a használata, ahol csupán PPP protokollal történő Internet csatlakozásra kívánják felhasználni.

15. EVENT funkciók

Talán az egyik legkülönlegesebb szolgáltatása az **Enfora** modemeknek az *Event Engine*; ez egy olyan szoftveres megoldás, amely átmenetet képez a "buta" modemek egyszerűsége és a piacvezető **Wavecom** modemekben futtatható *OpenAT*-ban megírt szoftverek között. Mindkét megoldásnak van előnye és hátránya: míg az *OpenAT* előnye a széleskörű programozhatóság, addig az *Event Engine* a könnyű programozhatóságban és a nagy megbízhatóságban jeleskedik: nincsenek bugok, nincsen több száz megabájtos pilótavizsgás fejlesztőkörnyezet! Sokszor azonban nincs is igény komolyabb funkciók végzésére; egyszerűen csak SMS-t kell küldeni egy jel megérkezésekor, vagy fel kell hívni egy számot egy paraméter túllépése esetén. Hogy ez mennyire így van, mi sem bizonyítja jobban, mint hogy az **Enfora** cég külön termékskálát gyárt *MobileTracker (MT)* néven, melyek olyan, **Enfora** modemmel ellátott, dobozolt modemek, amelyek tartalmazznak pár I/O csatlakozási pontot, és alkalmasak kisebb vezérlési feladatok megoldására.

Az *Event Engine* használatáról külön dokumentáció is szól *GSM0000AN015: Application Note – Event Monitor and Reporting Overview* címmel, ill. van egy letölthető Event konfiguráló program is *DIOEventTool.EXE* néven, melynek dokumentációját a *GSM0116PB001MAN: Enfora Event Tool User Manual* néven kell keresni az **Enfora** honlapján. Angolul kevésbé tudók részére igyekszem más, érdekesebb példákkal szolgálni az alábbiakban.

Az *Event Engine* programozásához az `AT$EVENT` paranccsal kell létrehozni legalább egy bemeneti és egy kimeneti eseményt. A bemeneti események lehetnek változás jellegűek (*Transition trigger*), esemény jellegű (*Occurrence trigger*) vagy bemeneti állapot (*Input*); kimeneti esemény csak egyféle van. (*Output*). Összesen 100 db különböző esemény-csoportunk lehet; egy esemény csoport pedig akár tucatnyi egymással *ÉS* kapcsolatban álló feltételből és független hatásból állhat. Ebből is látható, hogy igen komoly automatikák hozhatók létre a modemmel komoly programozói tudás nélkül is. Aki kicsit járatosabb az automatikák világában, az bizonyára találkozott már az ún. OPLC-kkel vagy más néven intelligens reléekkel. (Omron ZEN, Siemens LOGO!, stb.) Ezek a "nagy" ipari PLC-k olyan visszabutított verziói, melyeket komoly programozói tudás nélkül gyakorlatilag bárki tud programozni alig egy órányi ismerkedés után, hasonlóan egyszerű módon.

Az első példában leszimuláljuk a **Wavecom** modemek – az **Enfora** modemekből sajnos hiányzó – GSM LED-jét a modem GPIO2-es jelével. Ezt legkönnyebben az általam pont ilyen célra kifejlesztett *MTENFORA TEST BOARD*-al tudjuk letesztelni. Adjuk ki a következő parancsokat a modemnek:

```
at$evdela
at$event=1,0,9,1,1
at$event=1,3,33,65538,0
at$event=2,0,9,2,4
at$event=2,3,9,0,0
at&w
at$event?
$EVENT:      evgp  evtyp  evcat      p1      p2
              1A    0      9           1        1
              1B    3     33         65538    0
              2A    0      9           2        4
              2B    3      9           0        0
OK
at$reset
```

Az első sorban létrehozunk 1. sorszám alatt egy *Transition* trigger, amely akkor lesz igaz, ha az AT+CREG parancs értéke 1 lesz, azaz a GSM modem rendben feljelentkezik a hálózatra. A 2. sorban a hozzá tartozó *Output* hatás a GPIO2 kimenet villogtatása: a P1 paraméter alsó 16 bitje (2) a H szintet, azaz – a fordítva bekötött LED miatt – a sötét LED-et jelenti, a felső 16 bitje (1) az L szintet, azaz a világító LED-et jelenti. (65538 = 1*65536 + 2) A P2-es paraméter nullás értéke pedig nonstop villogást jelent. Tehát ha a GSM modem sikeres feljelentkezése esetén a LED villogni fog; $2^{1/4} = 1/2$ másodpercig sötét lesz, majd $1/4$ másodpercig világít. A 3. sorban az AT+CREG parancs 2 és 4 közötti egyéb értékeit figyeljük, és ebben az esetben a 4. sorban fixen alacsony szintre kapcsoljuk a GPIO2-t, azaz a LED folyamatosan fog világítani, ha nem vagyunk a GSM hálózatra kapcsolódva, pontosan úgy, ahogy azt a **Wavecom** modem is jelzi. (A hívás jelzésével még adós vagyok...)

A második példában az eddig méltánytalanul hanyagolt analóg bemenetek kezelését mutatom meg, ez azonban már kifejezetten az *MTENFORA TEST BOARD*-ot igényli, azon van ugyanis csak olyan könnyen kezelhető beállító potenciométer, amivel az analóg jel tesztelhető. A kiadandó parancsokat nem részletezem, csak az eredményt:

\$EVENT:	evgp	evtyp	evcat	p1	p2
1A	1	18	0	500	
1B	3	9	0	0	
2A	1	18	501	1760	
2B	3	17	0	0	
3A	1	18	0	250	
3B	3	19	0	0	
4A	1	18	251	1000	
4B	3	11	0	0	
5A	1	18	1001	1760	
5B	3	19	0	0	
6A	1	18	0	750	
6B	3	21	0	0	
7A	1	18	751	1500	
7B	3	13	0	0	
8A	1	18	1501	1760	
8B	3	21	0	0	
9A	1	18	0	1250	
9B	3	22	0	0	
10A	1	18	1251	1760	
10B	3	14	0	0	

A program működésének megértéséhez annyit kell tudni, hogy az analóg jelek bemeneti tartománya 0 és 1760 közé esik; ezt a tartományt én felosztottam 7 db 250-es résztartományra, és attól függően, hogy a bemenő feszültség melyik tartományba esik, az *MTENFORA TEST BOARD*-on található 4 db LED – sorban a GPIO7, GPIO6, GPIO4 és GPIO2 – különböző kombinációkban világít az alábbiak szerint:

	0-250	251-500	501-750	751-1000	1001-1250	1251-1500	1500-1760
GPIO2	●	●	○	○	○	○	○
GPIO4	○	●	●	●	○	○	○
GPIO6	○	○	○	●	●	●	○
GPIO7	○	○	○	○	○	●	●

Az első esemény-négyes első fele a GPIO2 kimenetet kezeli: 0 és 500 között világít a LED, felette nem. A 2. hatos esemény-csoport a GPIO4 kimenetért felel: 251 és 1000 között világít a LED, ezen kívül nem. A 3. hatos csoport a GPIO6-ot, még az utolsó négyes csoport a GPIO7-et kezeli. A páros (B jelű) *Output* sorok a kimeneteket kapcsolják alacsony vagy magas szintre, míg a páratlan (A jelű) esemény jellegű (*Occurrence trigger*) sorok definiálják az analóg jel értéktartományát.

Érdeemes kipróbálni a fenti programot, mert nagyon látványosan megmutatja, hogy az **Enfora** modem által biztosított analóg bemenet meglehetősen lomha: kell pár másodperc, mire a beállított érték beáll. Tehát az 1 mV-os mérési felbontású analóg bemenet csak lassú jelek figyelésére alkalmas, arra viszont kifejezetten alkalmas, mert nagyon jól védett az elektromos zajok és zavarok ellen!

A harmadik példám működik az eredeti, **Enfora**-féle fejlesztői panelen is; csak annyit csinál, hogy a GPIO2 bemenet 5. bekapcsolása után bekapcsolja a GPIO7-es kimenetet. Főleg az esemény-számoló trigger működésmódja miatt érdekes. A szükséges beállítások:

```
$EVENT:      evgp evtyp evcat      p1      p2
              1A    0      1          0        0
              1B    3     47          0        5
              2A    0     51          0        0
              2B    3     14          0        0
```

Az 1. sorban figyeljük a bemenetet, amit a 2. sorban számolunk az erre a célra szolgáló 47-es *Event Counter*-rel, melynek P2-es paramétere, azaz az 5-ös érték a maximuma. A maximum elérésekor a 3. sorban lévő 51-es speciális változás jellegű (*Transition trigger*) trigger lesz aktív, és az kapcsolja át a 4. sorban a GPIO7 kimenetet alacsony szintre, hogy a LED-je világítson.

Végül az utolsó példában egy egyszerű kis riasztót csinálunk; adjuk ki az alábbi parancsokat:

```
at$stoatev=1,atd06209427232v
at$stoatev?
$STOATEV:      AT Event#      AT Cmds
              1              ATD06209427232;
              2
              i..
              i5
OK
at$event=1,0,1,0,0
at$event=1,3,44,1,0
at$event?
$EVENT:      evgp evtyp evcat      p1      p2
              1A    0      1          0        0
              1B    3     44          1        0

at&w
```

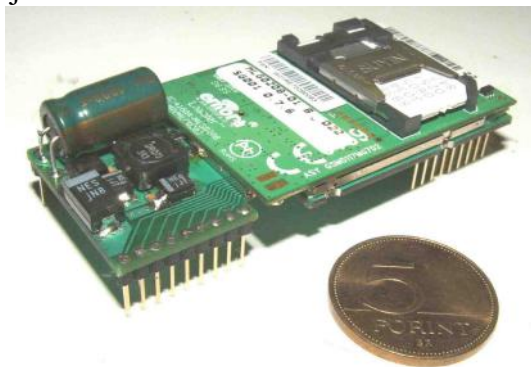
A fenti programban először definiáltunk egy AT parancsot, amelyek majd időzítve végre akarunk hajtani. A parancs megcsörget egy mobilszámot – az enyémet. A végén lévő "v" betű igen fontos, az a VOICE hívást generáló lezáró pontosvessző helyett áll! Tehát ez egy azonnali GSM VOICE hívást eredményez, amelyről minden esetben azonnal értesülök, ellentétben egy SMS-el, ami néha órákig tud keringeni a szerverek dzsungelében. Ezek után létrehoztunk egy változás jellegű (*Transition trigger*) trigger, amely a 44-es futtatás paranccsal futtatja az előzőleg eltárolt parancsot, amint a GPIO2-n egy alacsony szint van, pl. egy betörő kinyitotta a – sajnos nem létező – nyaralóm ajtaját. Azaz ekkor felhív engem az imént percek alatt létrehozott GSM modemes riasztóm...

Az *Event Engine* felhasználási köre elképesztően tág: van 4 időzítőnk, eseményszámlálónk, két analóg bemenetünk nagy érzékenységgel, 8 db digitális bemenet és/vagy kimenet; a GPS-es verzió (MLG0208) akár olyat is meg tudunk adni, hogy ha egy adott körön kívül került a készülék, akkor riasszon minket, pl. ha a kikötőben horgonyzó hajónkat elsodorta a víz! (Az **Enfora** tulajdonosa egy hajózási társaság; eredetileg az **Enfora** modemek még ilyen célra készültek; a *MobileTracker* egységek kifejezetten ilyesmi felhasználásra vannak reklámozva.) A kimeneti események jelentős része GPRS alapú üzenetküldést jelent, ami kezdőknek ugyan nem a legegyszerűbb, de a legsokoldalúbb és egyben a legolcsóbb mobil kommunikációs megoldás.

Pontosan a GSM témában kezdő, vagy éppen egyedi vagy kis sorozatú fejlesztésekkel foglalkozó kollégáim segítségére fejlesztettem ki egy *MTENFORA* nevű modult, amely tulajdonképpen az ipari szabványnak tekinthető, a **MultiTech** cég által kifejlesztett bekötéssel és méretekkel kompatibilis GSM megoldás az **Enfora** modemek felhasználásával. Sokan nem szeretnek ugyanis bajlódni az SMD GSM modem csatlakozók forrasztásával, a GSM-ek nagy áramlökésekkel produkáló tápegységeinek fejlesztésével, vagy épp a 2,8V-os jelszintű jelek szintillesztésével. Az általam kifejlesztett modulok az összes ilyen problémát megoldják:

- Széles bemenő feszültségtartomány 3 – 5V-ig (max. 2 – 6V-ig), 0,5A-re korlátozott maximális bemenő árammal, így akár 2 db ceruzaelemről is járatható a GSM modem!
- A teljes tápegysége lekapcsolható, így kikapcsolt állapotban az áramfelvétele mindössze μ A-es nagyságrendű!
- A tápfeszültséggel együtt változó bemenő-kimenő jelszintek: 5V-os tápfeszültség esetén TTL / 5V-CMOS, míg 3,3V-os tápfeszültség esetén LV-TTL / 3V-CMOS jelszintek.
- 2 mm-es lábtávolságú tűskesoros csatlakozás, amely akár foglalatba rakható, közvetlenül be is forrasztható, de akár drótokkal is behuzalozható.
- Szoftverfejlesztés vagy külső mikroprocesszor nem kell hozzá, hanem elég felprogramozni az *MTENFORA TEST BOARD*-al, majd a foglalatból kivéve átrakni a végleges áramkörébe.
- Az *MTENFORA TEST BOARD* önállóan felhasználható mint soros vonali GSM/GPRS modem PC-hez/PLC-hez, vagy mint soros vonali GPS vevő.
- A **MultiTech**-féle socket-technológiának köszönhetően csak egy áramkört kell kifejleszteni és egy NYÁK-ot kell legyártani; a behelyezett modultól függően mégis lehet PSTN modem, GSM modem, Ethernet, Bluetooth vagy Wi-Fi interfészünk ugyanahhoz a készülékhez, ahogy azt a Megrendelő kéri.
- Az eredeti socket opcionálisan ki van egészítve külső SIM foglalathoz kivezetésekkel, továbbá tartalmaz 2 analóg bemenetet és 4 digitális I/O-t, beépített soros ellenállással, így direkt LED vagy tranzisztor meghajtásra is alkalmas külső ellenállások nélkül.

Az *MTENFORA* modul fotója:



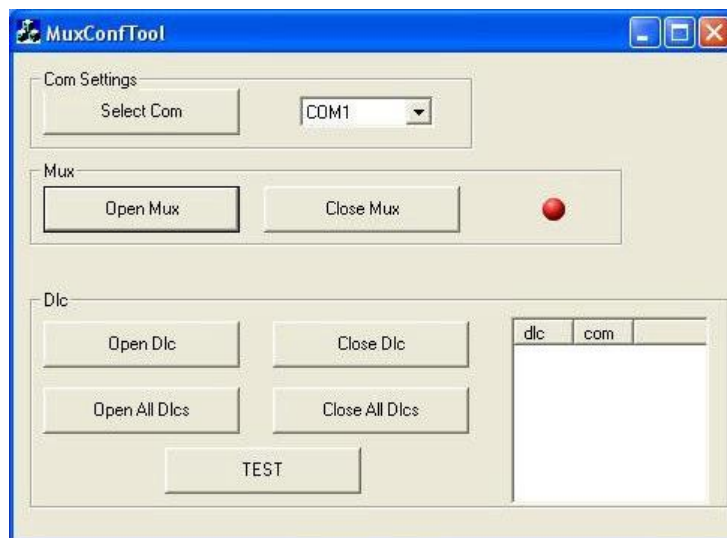
16. CMUX működése

CMUX technika az **Enfora** modemeken max. 6 db virtuális soros vonalat hoz létre egyetlen egy fizikai soros vonalon keresztül, és ezek a virtuális soros vonalak mint külön-külön modemek funkcionálnak: amíg az egyiket él egy TCP kapcsolat, a másikon SMS-eket küldhetünk-fogadhatunk, a harmadikon nézhetjük a pillanatnyi térerőt AT-parancsokkal, míg a negyediken vehetjük a bejövő VOICE hívás hívószám-kijelzését. Erről részletesen lásd az *GSM0000TN013 - Enabler IIG and IIE CMUX Description.pdf* fájlt.

Sajnos a CMUX protokoll bonyolultabb annál, semhogy pár parancsral el lehetne intézni, hiszen pl. CRC számítások is vannak benne. Ezért a bájt-szintre lebontott részletes kommunikáció leírásától eltekintenek. Szerencsére az **Enfora** cég készített egy kissé butucska, de azért jól használható PC-s programot ennek tesztelésére, amivel PC-n le tudjuk szimulálni a CMUX protokoll működését, majd tetszőleges soros vonali logger programmal – pl. a <http://www.compt.ru/> címről ingyen letölthető *COM Port Toolkit* -el – le tudjuk menteni a bájt-szintű kommunikációt. (Akinek ez esetleg gondot okozna, szívesen elküldöm a saját naplózásaimat.) A *PC_MUX* program sajnos csak *Windows*[®] XP-n (esetleg *Windows*[®] 2000 alatt) működik.

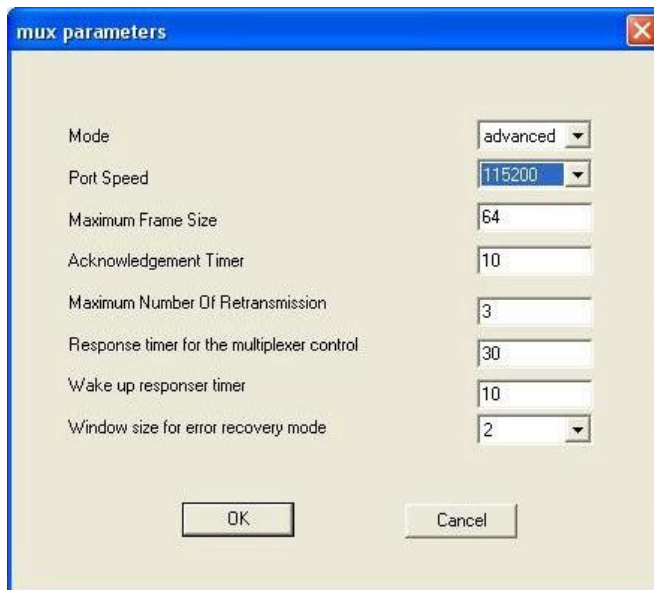
A *PC_MUX.ZIP* két fájlt tartalmaz; az egyiket - a Windows drivert - *MUXGSM2710.SYS*-nek hívják, és a *C:\windows\System32\Drivers* útvonalra kell másolni. (Vigyázat: a **Wavecom** hasonló szoftvere is ide teszi a sajátját, és bár azonos a kettő mérete, tartalmilag jócskán eltér tőle.) A másik fájl a *MuxConfTool.EXE*, amit bárhová másolhatunk.

A program elindítása előtt a GSM modemet 115200 (esetleg 57600) baud-ra kell beállítani az *AT+IPR* parancsral, ha nem ott lenne; a szoftver más sebességet sajnos nem támogat. Ezek után pl. a *Windows*[®] *HyperTerminál*[®]-jából ki kell adni az alábbi inicializáló parancsot: *AT+CMUX=1,0,5* Ezt követően a modem semmiféle parancsra nem reagál; ez normális! Lépünk ki a *HyperTerminál*[®]-ból, majd indítsuk el rögtön a *MuxConfTool.EXE*-t, ahol az alábbiakat láthatjuk:



Válasszuk ki előbb a COM1 vagy COM2 portot (többet nem támogat sajnos), majd **Select Com**.

Ezt követően klikkeljünk az **Open Mux** gombra, amire megjelenik az alábbi almenü:



Elsőként a késsel kijelölt **Port Speed** beállítását állítsuk át 115200-ra az eredeti 57600-ról (ha kell.) Az egyes paraméterek az AT+CMUX paramétereit jelentik sorban; részletes leírásuk a CMUX parancsnál található meg. Az **OK** gombra történő kattintás után máris sikeresnek kell(ene) lennie a csatlakozásnak, és a **Close Mux** gomb melletti, addig piros jelzőlámpának zöld színre kell váltania.

Ezt követően az **Open Dlc** gombbal tudunk új csatornákat létrehozni, míg a **Close Dlc** gombbal lezárni. Összesen 6 DLC csatornát tudunk megnyitni az **Enfora** modemeken (A **Wavecom** csak 4-et enged) Ne használjuk az **Open All Dlcs**, **Close All Dlcs** és a **TEST** gombokat, mert nem működnek; az **Open All Dlcs** pl. több, mint 6 DLC-t szeretne megnyitni egyszerre, és ez nem igazán működik, így belefagy... Az **Open Dlc** gombra történő kattintás után amúgy az alábbi "félelmetes" menü jön be a DLC csatorna paramétereivel:



Ebből a legelső, **DLCI** sor (a DLC csatorna száma) az érdekes; ha több csatornát szeretnénk megnyitni, ide kell 1-6-ig beírni a számokat, majd az **OK**-ra kattintani.

Miután megnyitottunk egy DLC csatornát, a hozzá rendelt soros vonalat - ami a gombok mellett lévő táblázatban látható - használhatjuk pl. a *HyperTerminál*[®] segítségével. Minden DLC csatornán elsőnek az *AT-command Interpreter Ready* üzenet fogad minket, mintha most kapcsoltuk volna be a modemet.

A DLC csatornák lezárását követően a **Close Mux** gombbal bonthatjuk végleg a CMUX kommunikációt. Ezt követően a modem visszatér a normál üzemmódba, az AT parancsokat újra elfogadja, így ha ismételten el szeretnénk indítani a CMUX programot, akkor előtte ismételten ki kell adni az AT+CMUX=1,0,5 parancsot.

