A Rosetta leszállóegység fedélzeti szoftverrendszere

Baksa Attila, Balázs András, Pálos Zoltán, Spányi Péter, Szalai Sándor, Várhalmi László – *KFKI Részecske- és Magfizikai Kutatóintézet*

Az előző, 2002/12. számunkban jelent meg a Szerzők cikke a Wirtanen-üstökös vizsgálatára készült Rosetta űrszonda feladatairól és a vele szemben támasztott – sokszor egymással ellentétben álló – műszaki követelményekről.

A követelmények összegzése nyilvánvalóvá tette, hogy egy sokfeladatos (multitasking), valós idejű (real-time) operációs rendszer használata elengedhetetlen. Ez nemcsak a véletlenszerűen bekövetkező kiszolgálási kéréseket tudja minimális időkésleltetéssel kiszolgálni, hanem egyben meg is könnyíti a feladatkezelő taszkok párhuzamos fejlesztését több programozó számára.

A hardverelemek egységes kezelését és hozzáférésük vezérlését eszközvezérlő modulok biztosítják, amelyek szintén az operációs rendszer részei. A szoftverrendszer szerkezetét az 1. ábra mutatja.

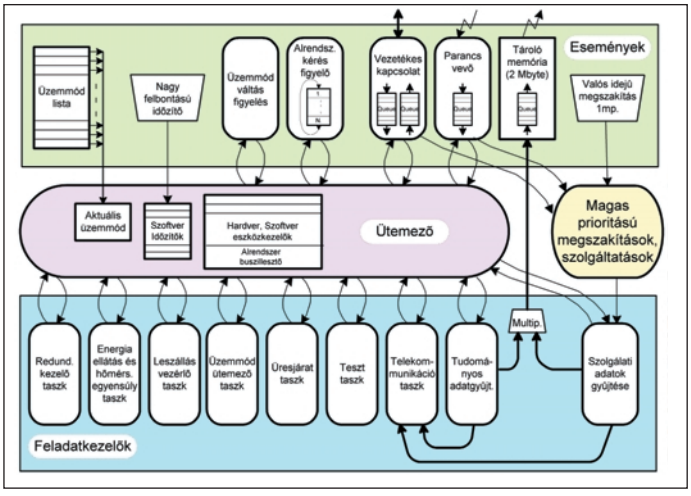
Az operációs rendszer magja és néhány feladatkezelő taszk az SGF Kft. közreműködésével készült. Az operációs rendszer ütemezi, valamint szolgáltatásokat nyújtva segíti a feladatkezelő taszkok munkáját. Az ütemezés szigorúan időosztásos (preemptive), de bármely taszk lemondhat felesleges processzoridő igényéről. Ez az ún. körbejárásos (Round-robin scheduling) megoldás. Időkritikus feladatok elvégzésének idejére az egyes taszkok magas futási prioritást kaphatnak az operációs rendszertől, ennek használata azonban veszélyei miatt feltételekhez kötött.

Az operációs rendszer és a feladatkezelő taszkok futása előtt a rendszerindító és inicializáló programmodul állítja be a processzorok sorrendjét, valamint választja ki a PROM és EEPROM memóriákban tárolt rendszerprogramok közül a futtatandót. A 2. ábra a különböző memóriákban tárolt szoftverelemek kapcsolatát és betöltésük variációs lehetőségeit mutatja.

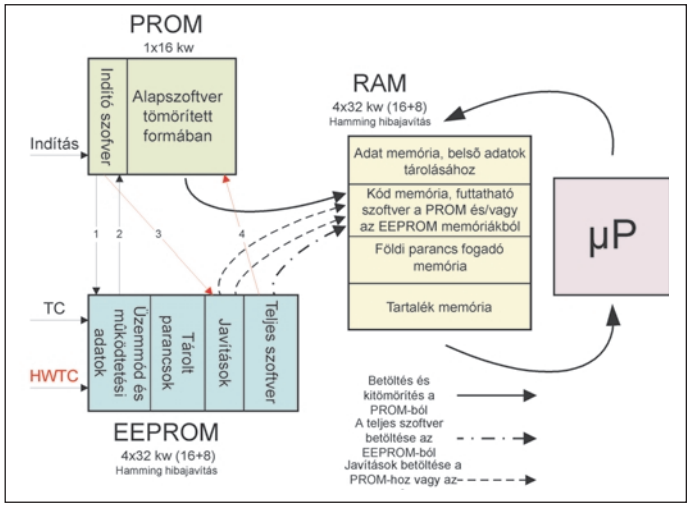
A megbízható PROM memóriában a leszállóegység létfontosságú feladatainak és vész üzemmódjainak ellátására képes szoftverrendszer található. Ennek a szoftvernek a mérete kb. 40 kilobyte, de az ennél kisebb kapacitású PROM memóriában csak tömörített formában fér el, ami külön érdekesség. Ezt a programot ki kell tömöríteni futtatás előtt, ami automatikusan megtörténik a fő szoftverrendszer meghibásodása esetén, amely a sérülékenyebb EEPROM memóriában található. A PROM tartalma a számítógép szétszedése nélkül nem cserélhető, ezért az űrszonda hosszú tesztelése miatt már jóval korábban el kellett készíteni ezt a szoftvert, mint ahogy a számítógép feladatait véglegesíteni lehetett volna.

Az EEPROM-ban tárolt, 62 kilobyte méretű fő szoftverrendszer képes a leszállóegység valamennyi feladatának elvégzésére, és a rádiórendszeren keresztül akár 10 év múlva, az üstökös felszínén is kicserélhető vagy javítható. Javításokat (patch-ek) földi parancsok formájában lehet a fedélzetre küldeni, amelyekkel a szoftver bármely része, még az operációs rendszer is módosítható. A kiválasztott szoftvert az indító modul áttölti a RAM memóriába, ott módosítja azt az esetleges javító csomagokkal, majd átadja a vezérlést a betöltött operációs rendszer inicializáló részének. Tesztelési célokra akár közvetlenül a RAM memóriába is fel lehet a földről küldeni egy teljes szoftvert. Futás közben, szintén földi paranccsal, lehetőség van a processzor átkapcsolására, ezáltal az új szoftver futtatására. Ez lehetővé teszi egy új szoftververzió kipróbálását anélkül, hogy közben vállalni kellene a korábbi verzió felülírásának kockázatát az EEPROM memóriákban.

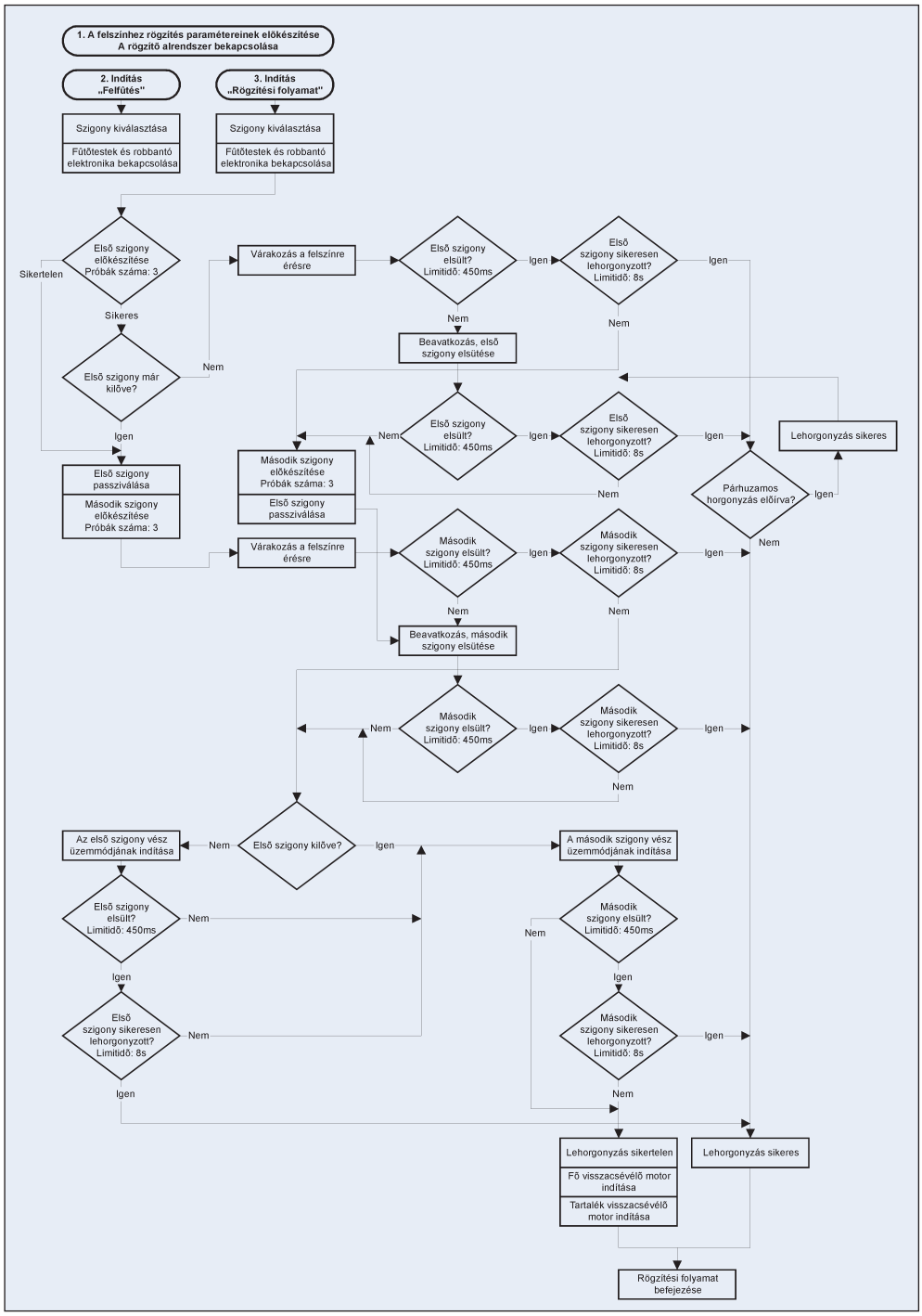
1. ábra A szoftverrendszer szerkezete



2. ábra A szoftverrendszer indításának lehetőségei és memóriaképe



A multitaszkos környezet kialakítását a processzor csak annyiban támogatja, hogy a verem (stack) taszkok közti felosztását és szigorú védelmét biztosítja – ugyanis maga a verem a processzorral egyben, integráltan helyezkedik el –, viszont a memória taszkonként felosztott védelmét nem támogatja, ezért ezt az operációs rendszer szolgáltatásaként kellett megvalósítani. A taszkok közti kommunikáció lehetőségét szintén az operációs rendszer végzi jelzések és üzenetek formájában, de egy kijelölt



3. ábra A felszínre érés felügyeletének folyamata

memóriaterületen keresztül akár nagy mennyiségű adatcsere is lehetséges.

A perifériák (közös alrendszeri busz, rádiórendszer, telemetria-memória) fizikai szintű kezelése hardver megszakításokkal történik, az adatok szavankénti vétele és küldése, valamint az üzenetstruktúra felismerése és létrehozása az operációs rendszer feladata. Az üzenetek magasabb szintű kezelését a feladatkezelő taszkok végzik.

A szoftver folyamatosan ellenőrzi a hardver működését, és hiba észlelése esetén átvált a meghibásodott hardverelem tartalék példányára. Ez igaz a processzorkártyákra is, tehát a szoftver kezdeményezheti saját processzorának leváltását a tartalék egységre. A futó szoftver helyes működését hardver (watch-dog időzítő) is ellenőrzi, ezáltal a szoftverrendszer hibás működése esetén is van lehetőség a tartalék processzor és szoftver aktiválására. A processzorok kapcsolatban állnak egymással, a fő egység folyamatosan továbbítja a küldetés aktuális állapotának legfontosabb változásait a tartalék egységnek, s egy esetleges szerepváltás után a korábban tartalék processzor zökkenőmentesen képes átvenni a küldetés irányítását.

Mivel a Harris RTX2010-es processzorhoz csak szerény képességekkel rendelkező szoftverfejlesztő eszközök álltak rendelkezésre, ezért első lépésként egy keresztfejlesztő környezetet kellett kialakítani, amely a fejlesztés alatt álló szoftver letöltése és futtatása mellett a cél rendszeren (target) képes segíteni a hibakeresést és a valós idejű, párhuzamos rendszer tesztelését. A fedélzeti szoftver egyik feladatkezelő taszkjába került egy hibakereső és monitorozó modul, amelynek egyszerű parancskészlete segítségével a fejlesztők egy asztali PC számítógépen, annak soros vonalán keresztül követhetik nyomon vagy befolyásolhatják a fedélzeti szoftver működését.

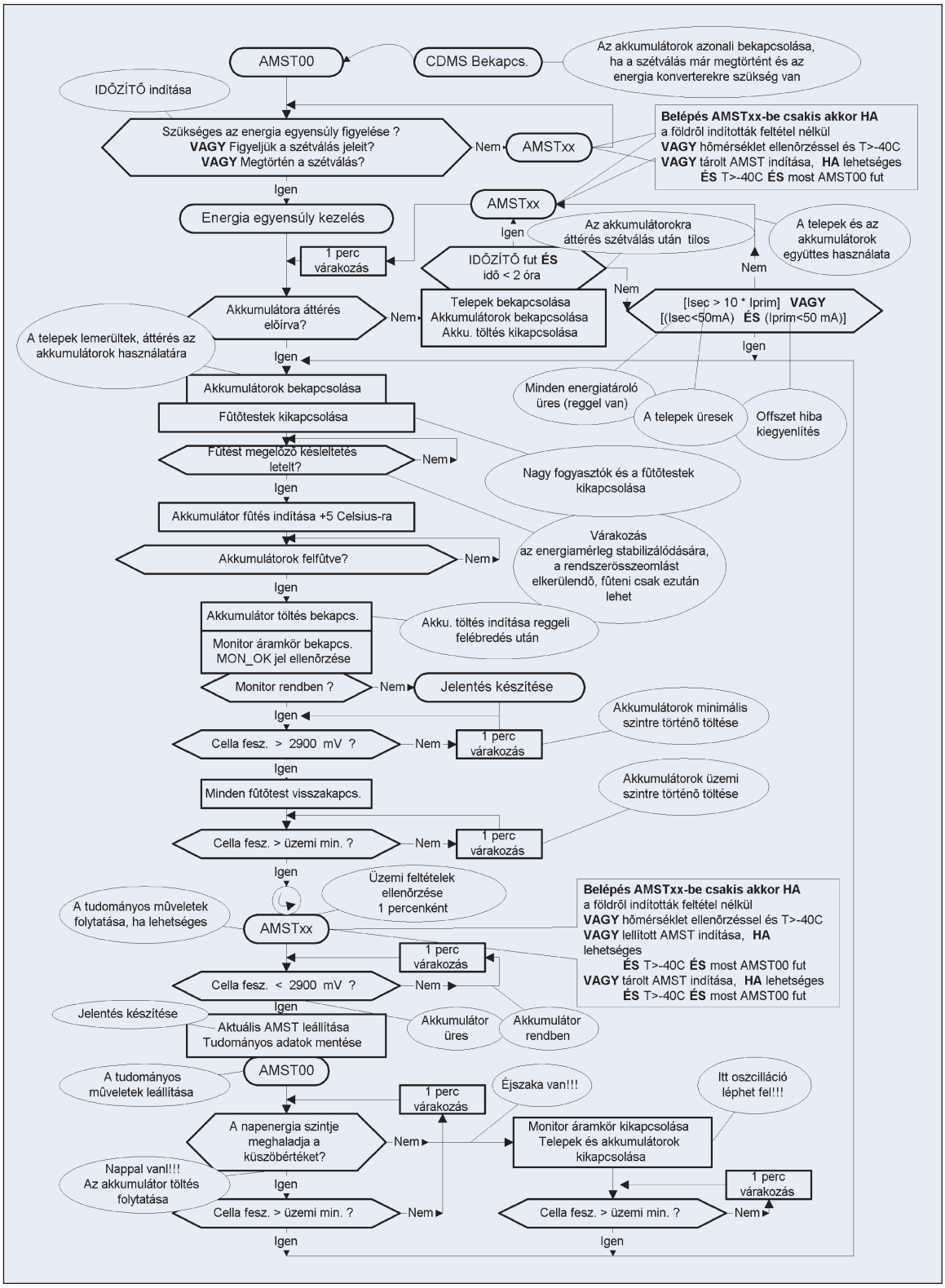
Ezért a fejlesztés korai fázisában a fedélzeti elektronika processzorának buszára egy soros interfészt (UART) tartalmazó, kiegészítő kártya csatlakozott. A fejlesztés későbbi szakaszában a minősítő példányok teszteléséhez már más típusú monitorozó rendszer készült, mert a soros vonali kiegészítő kártya ebben a hardverkonfigurációban már nem volt alkalmazható. Ez a közös alrendszeri buszon (SSIF) keresztül képes egy leegyszerűsített parancskészlet végrehajtására.

Nyolc feladatkezelő taszkot fejlesztettünk ki, amelyek sorban a következők.

- *Telekommunikációs taszk*, amely az üstökös körül „keringő" egységen (anyaszondán) keresztül a rádiólátható ság idején tartja a kapcsolatot a földdel, elvégzi a parancsok vételét, a telemetria adatok továbbítását, valamint a nagy kapacitású telemetria memóriakártyák kezelését. Feladata továbbá a tartalékolt rádiórendszer hibáinak kiküszöbölése és a vész üzemmódú rádióadások kezelése.
- *Üzemmód-ütemező taszk,* amely a földi parancsok értelmezését és végrehajtását, a működési üzemmód beállítását, valamint a tudományos küldetés ütemezését végzi. Azt, hogy a számítógépnek egy adott üzemmódban mi a feladata, milyen műszereket kell kiszolgálnia, és hogy az üzemmód mikor kezdődik, illetve meddig tart, egy üzemmódlista- táblázat írja le. Megadható az üzemmódok sorrendje, és lehetőség van külső események alapján feltételes elágazások beiktatására is. Ezzel a megoldással hosszabb mérési szekvenciák esetén is leírhatók az üzemmódok közötti átmenetek. Ezeket az adatgyűjtési szekvencia táblázatokat az angol nevének kezdőbetűi alapján (Acquisition Mode Sequencer Tables) AMST-nek röviditjük. Ennek a kötött struktúrájú táblázatnak a segítségével a szoftver rugalmasan és gyorsan konfigurálható anélkül, hogy magát a programkódot változtatni kellene.
- *Tudományos alrendszer kezelő taszk,* amely az intelligens tudományos berendezések igényeinek kiszolgálását, valamint a

szolgálati adatok és mérési eredmények összegyűjtését végzi. Lehetővé teszi a berendezések közti kommunikációt, és tárolási szolgáltatást ad azok belső működési adatai számára, emellett periodikusan továbbítja a fedélzeti időt számukra.

- *Redundancia kezelő taszk,* amelynek feladata a rendszerben található tartalékolások kezelése és a processzorok közti kommunikáció megvalósítása. A leszállóegység működése során fontos szerepe van a pontos fedélzeti időnek – például a mérések időpontja vagy a memóriában tárolt és időzítve kiküldendő parancsok is ehhez igazodnak –, ezért ezt három független időmérő forrás felhasználásával számítja ki. Folyamatosan szinkronizálja a számítógép-hardver két darab valós idejű óráját (RTC boards) és a szoftver valós idejű óráját, amit egy 32 Hz-es megszakítás ütemez. A szoftver fejlesztési fázisában ez a taszk tartalmazta a hibakereső és monitorozó modult.
- *Teszttaszk,* amely a számítógép hardver elemeinek folyamatos ellenőrzését végzi, és a tesztek eredményéről összesítést készít a földi személyzet számára. A memóriahibák detektálását Hamming kódoló és ellenőrző áramkör segítségével végzi, és a meghibásodott (de javítható) szavakat kijavítva visszaírja az EEPROM, ill. a RAM-memóriákba.
- *Aktív leszállító rendszert kezelő taszk,* amely az anyaszondától történő szétválás, az ereszkedés, az üstökös felszínére érés és a talajhoz rögzítés folyamatait felügyeli. Hibaesemény észlelése esetén azonnal beavatkozik a hibás folyamat menetébe és a tartalék rendszereket bekapcsolva folytatja a leszállást. Ez a taszk csak az üstökös megközelítése során fut magas prioritással a lehető leggyorsabb reakcióidő elérése érdekében. Egyik legfontosabb és legkritikusabb feladata a talajt érés gyors érzékelése és utasítás kiadása a leszállóegységre szerelt szigony kilövésére. Ha ez késve történne, az űreszköz – a kis gravitáció miatt – visszapattanhatna az üstökös felszínéről. A 3. ábra a felszínre érkezést felügyelő és vezérlő folyamatot mutatja be.
- *Tápellátás- és hőmérsékleti szabályozó taszk*, amelynek feladata az energiaellátó alrendszer vezérlése, az energiaforrások állapotának kezelése, valamint az előírt hőmérsékletek ellenőrzése a leszállóegység különböző zónáiban. A leszállóegység négy energiaforrással rendelkezik. A szétválást megelőzően az anyaszondától kapja az energiát. Önálló működésének első néhány napjában, amely a küldés elsődleges szakasza, a viszony-



4. ábra Az energiaellátás felügyeletének algoritmusa

lag nagy kapacitású telepek szolgáltatják az energiát. A küldetés további hónapjai alatt pedig napcellákról tölthető akkumulátorok szolgáltatják az energiát. Az üstökösön a földhöz hasonlóan nappalok és éjszakák váltogatják egymást, ezért ez a taszk képes kezelni az akkumulátorok lemerülésének tényét. Elmenti a rendszer aktuális állapotát, majd éjszakára kikapcsolja a leszállóegységet. A reggel érkező napenergia hatására megkezdi a belső tér felfűtését a –40 °C üzemi hőmérsékletre, ami az éjszakai hőveszteség miatt szükséges (éjszaka az üstökös felszíne –200 °C-ra is lehűlhet). Ezután engedélyt ad az üzemmód-ütemező taszk számára az előző estén félbehagyott tudományos kísérletek folytatására.

A taszk feladatköréhez tartozik a következő folyamatok irányítása:

– A telepek terheléssel történő felkészítése használatba vételük előtt, ami a leszállást megelőző napokban történik majd meg. Az előkészítés menetéről adatsorokat továbbít a földre részletes elemzés céljából.
– Az akkumulátorok teljes feltöltése az anyaszonda energiájából, szintén a leszállást megelőzően. A Li-ion akkumulátorok túltöltése robbanásveszélyes, ezért az akkumulátorok hét-hét cellájának energiaszintjét külön-külön, többféle algoritmussal ellenőrzi a szoftver. Az energiaszintek ellenőrzése még többszörös áramköri meghibásodás ellenére is biztosítva van.
– A telepek, az akkumulátorok és a napelemtáblák folyamatos felügyelete és vezérlése. Ez a feladatkör a szétválást követően a taszk egyik legfontosabb funkciója, és összetettségét a 4. ábra mutatja be.
– A fedélzeti műszerek és alrendszerek be-, illetve kikapcsolása az üzemmód ütemező taszk utasításai alapján.
– Az esetleges rövidzárlatok és túlfogyasztások érzékelése és a hibás fedélzeti alrendszer kiiktatása a további munkából.
– Minden észlelt hibaeseményről jelentés készítése a földi személyzet számára.

• *Üresjárati (Idle) taszk*, amelynek feladata a többi taszk inicializálását követően a rendszer üresjárati idejében a futási aktivitás fenntartása.

Az Európai Űrhivatal (ESA) szoftverfejlesztésre és tesztelésre vonatkozó követelményei szigorú szabályokat írnak elő a forrásprogramok írására vonatkozólag, valamint megkövetelik verziókövető és archiváló rendszer használatát.

Több száz oldalas dokumentációk készítésére volt szükség a rendszer ismeretanyagának rögzítéséhez:

– Követelmények összefoglaló leírása (Requirements Document)
– Alrendszer működési leírás (Sub-System Specification)
– Szoftvertervezési leírás (Detailed Design Document)
– Részletes szoftverleírás (Detailed Software Specification)
– Szoftververzió-jelentés (Version Control Report)
– Felhasználói leírás (User Interface Specification)
– Tesztelési eljárások a funkciók ellenőrzésére (Functional Test Plan).

Cikksorozatunk következő és egyben utolsó része a fedélzeti számítógép funkcióinak ellenőrzését segítő automatizált tesztelő rendszer bemutatásával foglalkozik majd.

Ezúttal szeretnénk kifejezni köszönetünket a Magyar Űrkutatási Irodának, amely támogatásával lehetővé tette a Rosetta-programban való részvételt.

KFKI RMKI Űrtechnológiai Osztály
Dr. Szalai Sándor
1121 Budapest, Konkoly Thege út 29-33.
Tel.: 392-2523, fax: 395-9151
E-mail: szalai@rmki.kfki.hu, www.rmki.kfki.hu