

# PocketQube műhold fedélzeti számítógépének tervezése

Önálló laboratórium beszámoló

*Készítette:* Kristóf Timur, Budapest, 2015. május 08.

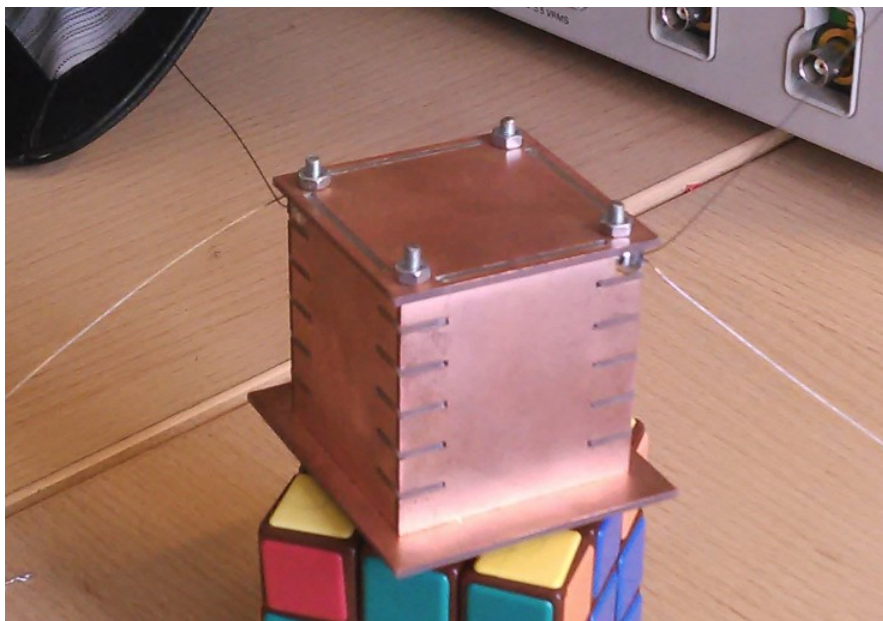
*Konzulens:* Dudás Levente

## Tartalomjegyzék

Bevezető.....	3
Küldetés.....	4
Mire jó ez?.....	4
Feladatkiírás.....	5
Kihívások.....	6
Alkatrészek.....	6
Energia.....	6
Redundancia.....	7
Mechanikai terhelés és termikus viszonyok.....	8
Rendszerterv.....	8
A SMOG-1 teljes rendszerterve.....	10
Fedélzeti számítógép részei.....	11
Mikrokontroller.....	11
Háttértár (flash memória).....	11
Mekkora a kellő tárolókapacitás?.....	12
RTCC (real-time clock & calendar).....	12
Szoftver felépítése.....	14
Megvalósítás.....	14
Analógia operációs rendszerekkel.....	14
Fájlrendszer, tömörítés.....	14
Folyamatábra.....	15
Energiagazdálkodás.....	15
Alvó mód (sleep mode).....	16
Készenléti mód (idle mode).....	16
Szunyókáló mód (doze mode).....	16
Összeköttetések.....	16
Memória, RTCC és szenzorok.....	16
PCU (power control unit).....	17
COM (rádiókommunikáció és spektrumanalizátor).....	17
Áramkorlátozó kapcsolók.....	17
MPPT telemetria.....	18
Összeköttetések blokkvázlata.....	19
Összefoglaló.....	20

## Bevezető

A PocketQube a CubeSat típusú műholdak egy új osztálya, amelyet a Morehead State University és a Kentucky Space szabványosított. Fő jellemzője, hogy mérete nem haladhatja meg az 5×5×5 centimétert, és tömege legfeljebb 180 gramm.



1. ábra: SMOG-1 PocketQube műhold

Magyarország első műholja a MASAT-1 volt, amely az eredeti CubeSat szabvány szerint 10×10×10 cm-es volt. Ennek egyik utódprojektje a SMOG-1, amely a már az újabb, kisebb méretű szabvány szerint készül a Budapesti Műszaki és Gazdaságtudományi Egyetemen a Villamosmérnöki és Informatikai Kar és a Gépészmérnöki kar együttműködésével. [1] A műhold elektronikáját a Szélessávú Hírközlés és Villamosságatan Tanszék hallgatói tervezik meg és rakják össze. A projekt 2014-ben kezdődött, és jelenlegi terveink szerint 2015 decemberében várható a mérnöki példány elkészülte. Ez korántsem jelenti azonban azt, hogy ezzel a műholdat befejeztük; ezt a példányt alapos tesztelésnek fogjuk alávetni és a tesztek eredménye szerint fogjuk továbbfejleszteni, esetleg egyes kritikus részeket újratervezni, ha úgy ítéljük szükségesnek. A mérnöki példányt egy kvalifikációs és egy repülő (végleges) példány fogja követni.

Ebben a félévben az önálló laboratórium feladatom az volt, hogy a SMOG-1 műhold feldélzeti számítógépét (on-board computer, OBC) tervezzem meg. Jelen dokumentumban erről a munkáról számolok be. Először szeretnék egy átfogó képet adni a SMOG-1-ről általában, beleértve a műhold teljes rendszertervét is, aztán pedig a saját

feladataimat és közreműködésemet fogom részletezni. Végül kitérek arra, hogy a jövőben merre fog folytatódni a fejlesztés, és milyen feladatok várnak még megoldásra, mielőtt a fedélzeti számítógép mérnöki példánya elkészülhet.

## Küldetés

A SMOG-1 a magyarországi CubeSat műholdak oktatási vonalát képviseli. Ez azt jelenti, hogy nagy részét a Műegyetem hallgatói tervezik, és a költségvetése is az egyetemi kasszához mért, vagyis drága, úrrinósított eszközök helyett a piacon széles körben elérhető ipari alkatrészekből építkezünk.

Műholdunk fő küldetése az elektromágneses spektrum monitorozása a Föld körüli pályáról DVB-T sávban. Vagyis azt mérjük, hogy a földi DVB-T műsorszóró adók jele mennyire vehető odafönről. [1]

## Mire jó ez?

Természetesen adódik a kérdés, hogy miért érdekes bárki számára is az az információ, hogy a földi digitális műsorszórás vajon vehető-e az űrből. Hiszen odafönt senki sem szeretne tévét nézni. Pontosan ez a válasz a kérdésre. Ugyanis a műsorszóró adók célpontja a földi lakosság, ezért minden olyan elektromágneses jel, ami vehető ezektől az adóktól az űrben, valójában elpazarolt, kidobott teljesítmény.

Küldetésünk tehát alkalmas lesz arra, hogy a méréseinkből arra lehessen következtetni, pontosan melyik földi adó mennyi felesleges teljesítményt sugároz fölfelé. Ezen információ birtokában pedig az adott adó üzemeltetői továbbfejleszthetik, vagy hangolhatják az antennáik iránykarakterisztikáját olyan módon, hogy a földi lakosság felé jutó teljesítmény a lehető legnagyobb, az űrbe kisugárzott elpazarolt teljesítmény pedig a lehető legkisebb legyen.

A küldetés legfontosabb eleme egy folytonosan hangolható spektrumanalizátor, amely a műhold kommunikációs rendszerének (COM) részét képezi. Ezt a spektrumanalizátort és a felbocsátásához használt magaslégköri ballont Dudás Levente és a Mikrohullámú Távérzékelés Laboratórium munkatársai készítették és már számos légballoon kísérleten bizonyított. [1]

A spektrumanalizátoron kívül néhány egyéb kísérlet fog még helyet kapni a fedélzeten, többek között egy totáldózismérő RAD FET van tervbe véve, valamint egy különleges hőszigetelő anyag tesztje, ami az akkumulátort fogja védeni.

## Feladatkiírás

Erre a félévre a feladataim az alábbiak voltak:

- A SMOG-1 küldetésének és felépítésének megismerése
- Ismerkedés az űreszközök tervezése során felmerülő kihívásokkal
- Részvétel a műhold rendszertervének kidolgozásában
- Fedélzeti számítógép (on-board computer, OBC) működésének, blokkvázlatának megtervezése
- Kitalálni, hogy a fedélzeti számítógép milyen alrendszerekkel hogyan legyen összekötöttségben
- Megismerni a témához kapcsolódó kommunikációs protokollokat és kidolgozni, hogy a fedélzeti számítógép hogyan kommunikáljon az egyes alrendszerekkel
- Eldönteni, hogy melyik alrendszereknek van szüksége helyi intelligenciára
- Alkatrészek megválasztása a fedélzeti számítógéphez
- Fedélzeti számítógép prototípusának megtervezése (kapcsolási rajz és nyomtatott áramköri terv)
- Fedélzeti számítógép szoftveres architektúrájának megtervezése
- A Silicon Laboratories fejlesztői környezetének (Simplicity Studio) megismerése
- A Microchip fejlesztői környezetének (MPLAB X) megismerése

## Kihívások

### Alkatrészek

Általában – mint ahogyan a CubeSat műholdak esetén is – a PocketQube-ok tervezésénél a mérnökök a piacon széles körben elérhető ipari alkatrészekből építkeznek. Ennek fő oka az, hogy ezeket a műholdakat zömében oktatási céllal, vagy rádióamatőrök készítik, emiatt pedig az úrminősített alkatrészek használata pénzügyileg megfizethetetlen terheket róna a tervezőkre.

Természetesen nem mindegy, hogy milyen alkatrészeket használunk, hiszen egy 5 cm-es kocka meglehetősen korlátozott méretű. Így az egyik legfontosabb szempont, hogy mindenből a lehető legkisebbet válasszuk. A SMOG-1 csapat jelentős időt fordított arra, hogy különféle katalógusokat böngészett, kutatva azt, hogy egy adott feladatra mik a lehető legkisebb elérhető áramkörök, illetve, hogy az adott alrendszer hogyan lehet a lehető legkisebb méretűre zsugorítani. (Ennek kiemelkedő példája az RTCC, a szenzorika és az EPS-ben használt áramkorlátozó kapcsolók.)

Másik fő szempont – amely sajnos kissé ellentmond az előzőnek – az, hogy olyan alkatrészeket válasszunk, amik „már jártak az űrben”, vagyis volt már olyan csapat, akik sikeresen használták az adott alkatrészt a saját amatőr műholdjukban. Sajnos, mivel a CubeSat műholdak még mindig nem terjedtek el széleskörűen, nehéz ilyen alkatrészeket találni, főleg a méretkorlátok miatt, hiszen a SMOG-1 a korábbi PocketCube-ok térfogatának mindössze egy nyolcadával rendelkezik. A fentiek miatt többnyire megelégszünk azzal is, ha olyan alkatrészt találunk, ami valamilyen ipari minősítéssel rendelkezik (pl. a szokásosnál szélesebb hőmérséklettartományban ígér jó működést a gyártó).

### Energia

Elsődleges energiaellátásunkat napelemek biztosítják, amelyek a kocka 5 cm-es oldallapjain helyezkednek el. Ez azt jelenti, hogy a napelemek számára rendelkezésre álló felület kb. 1/4-e a MASAT-1 felületének, vagyis negyedakkora bejövő teljesítményre számíthatunk, amely előzetes becslésünk szerint kb. 0,3 W (körátlag). Ekkora teljesítménnyel kell gazdálkodnunk, és a fedélzeti alrendszereknek is ebből kell működniük. [3]

Természetesen a SMOG-1-ben is lesz egy akkumulátor, azonban a hely szűke miatt ez is lényegesen kisebb lesz, mint a MASAT-1 akkumulátora volt. Optimális esetben az akkumulátort a napelemek töltik, amikor a műhold a Föld körüli pályája napos részén tartózkodik, és amikor földárnyékban van, akkor pedig teljes egészében az

akkumulátorról működnek majd a fedélzeti rendszerek.

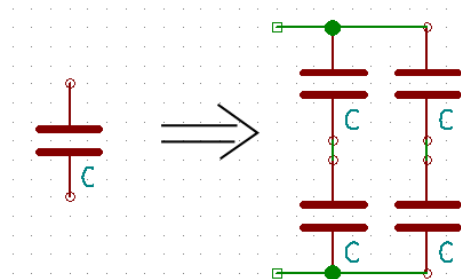
Nem mindegy, hogy milyen akkumulátort használunk, mert bizonyos fajták nem működnek jól az űrben. (pl. a Lítium-polimer akkumulátorok néhány kivételtől eltekintve alkalmatlanok a feladatra.) Azonban nem feltételezhetjük azt sem, hogy az akkumulátor a SMOG-1 küldetésének teljes időtartama alatt hibátlan lesz, mint ahogy azt sem, hogy már a pályára állításkor egyáltalán működni fog. (ld. később.) Emiatt a műholdat úgy tervezzük meg, hogy olyan esetekben is működjön (legalább a pályája napos oldalán), amikor az akkumulátor tönkrement. [2, 3]

## Redundancia

Hogy ellenálljon a világűr viszontagságainak, a SMOG-1 alrendszereit redundáns módon kell megtervezni, mégpedig úgy, hogy bármely pont meghibásodása esetén a teljes rendszer zökkenőmentesen működhessen tovább.

Lehetséges meghibásodások az alábbiak:

- Ellenállás: szakadássá változhat, ezért kritikus helyekre legalább két párhuzamos ellenállást helyezünk el.
- Kondenzátor: rövidzárrá vagy szakadássá is változhat, ezért kritikus helyre négy kondenzátort kell elhelyezni. Ha ezek közül bármelyik bárhogyan elromlik, a kapcsolás még mindig kondenzátorként működik, csak a kapacitása változhat meg.



2. ábra: Kondenzátorok elrendezése

- Diódák, tranzisztorok: rövidzárrá vagy szakadássá változhatnak
- Integrált áramkörök: bármelyik alkatrészláb rövidre záródhat akár a föld (logikai 0) vagy a tápfeszültség (logikai 1) felé, vagy tirisztorhatás jöhet létre. Ezt általában funkcionális redundanciával küszöböljük ki, vagyis az adott funkcionális egységből, amelynek része az adott áramkör, legalább kettőt helyezünk el a fedélzeten.

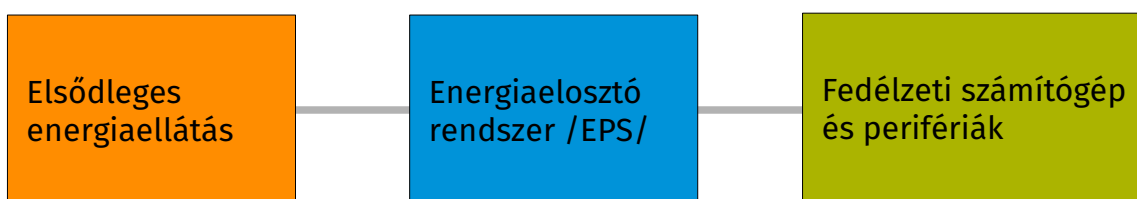
Ez azt jelenti, hogy a tervezés során különös gondot kell fordítani arra, hogy bármelyik alkatrész, ha elromlik, azzal ne veszélyeztesse a többi alrendszert. (Pl. Ha egy integrált áramkör valamelyik kivezetése rövidre záródik a földponttal, az ne okozhassa más áramkörök tápellátásának megszűnését, és ne akadályozza azok között a kommunikációt.)

## Mechanikai terhelés és termikus viszonyok

A műholdat rázópad és termovákuum kamra segítségével fogjuk tesztelni, hogy ellenálljon az űrben uralkodó ellenséges viszonyoknak.

## Rendszerterv

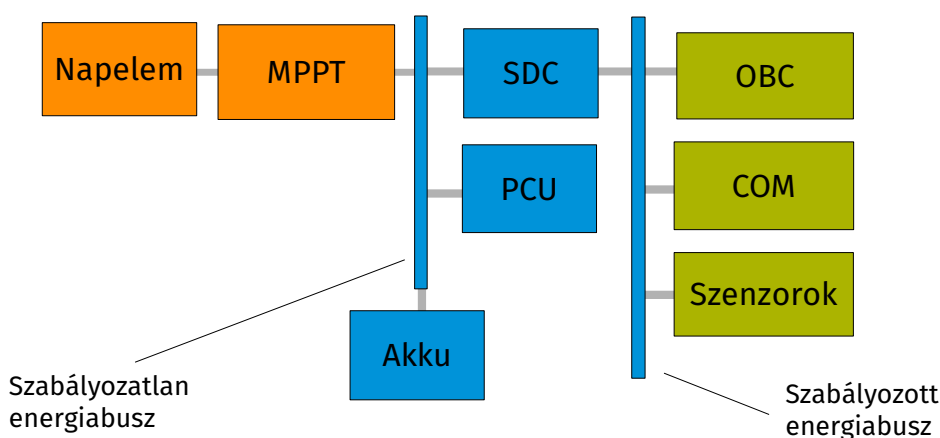
Műholdunk alapvetően három jól elkülöníthető egységből épül fel. Az elsődleges energiaellátásért a kocka oldalain levő napelemek felelősek, melyek MPPT áramkörrel ellátva táplálják a SMOG-1-et. Ezt az energiát kezeli, raktározza és átalakítja az energiaelosztó rendszer (EPS). Ez állítja elő azt a feszültséget, amiről a fedélzeti számítógép és annak perifériái működnek.



3. ábra: SMOG-1 magas szintű blokkvázlata

Az energiaelosztó rendszer tartalmaz egy akkumulátort az energia tárolására, egy PCU (power control unit) egységet a programozott logikák megvalósítására és egy step-down konvertert, amely a fedélzeti számítógép és perifériái részére állít elő megfelelő feszültséget. Ezen áramkörök mindegyike megfelelő védelemmel (túláram és túlfeszültség, redundancia, stb.) vannak ellátva.

Az alábbi ábrán (a redundáns párjuk nélkül) látható a műhold blokkvázlata a lényeges funkcionális egységekkel.



4. ábra: a SMOG-1 vázlatos rendszerterve



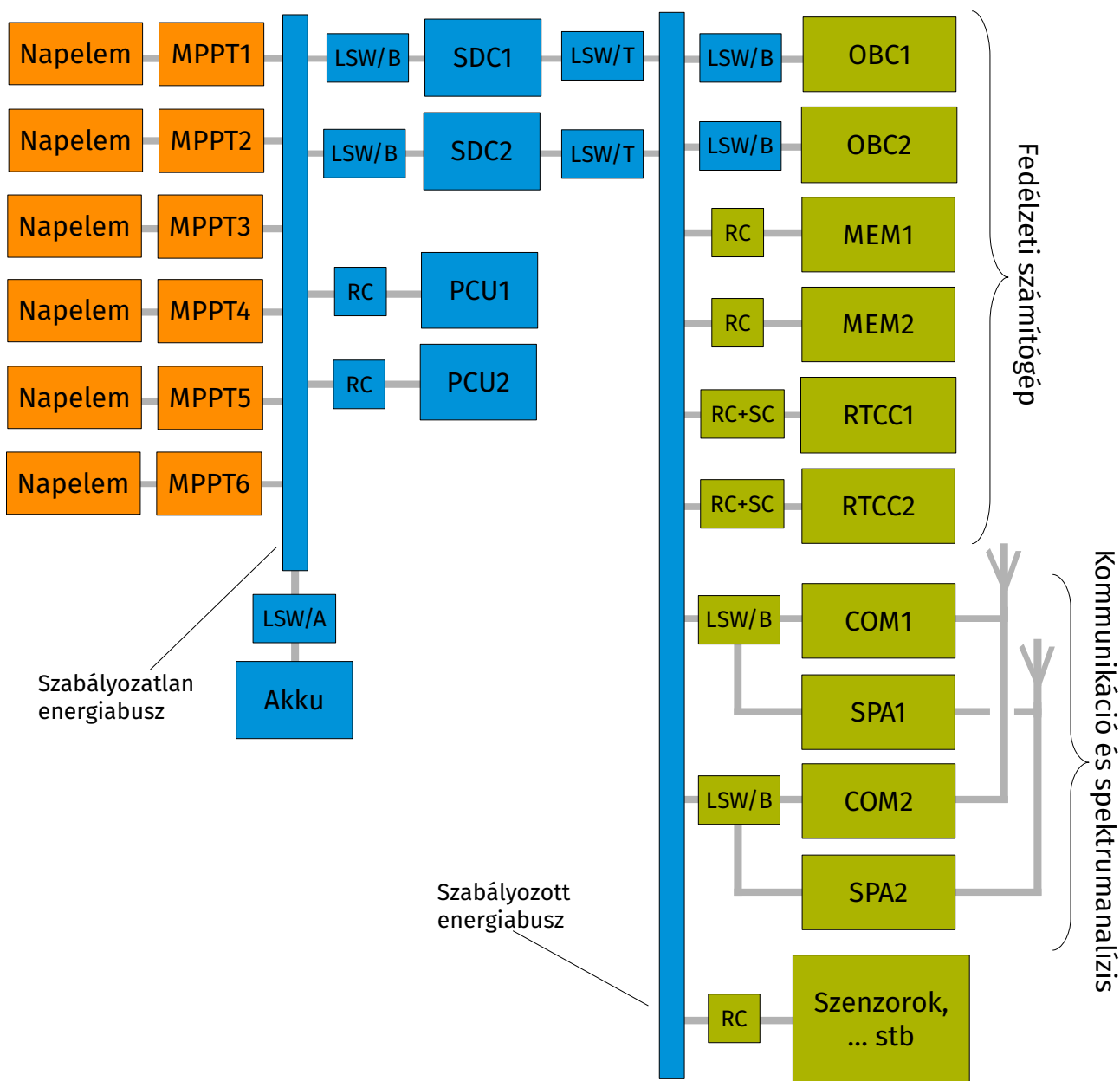
A napelemek és az akkumulátor egy ún. szabályozatlan energiabuszra (ahol akár 5V is lehet) vannak rákötve. Erről üzemel a PCU és az SDC, amely egy ún. szabályozott energiabuszt hajt meg a kimenetével (1,8V – 3,6V). Ezen a buszon olyan feszültség van, amiről már az OBC és perifériái működni tudnak.

Természetesen a fenti ábra még tovább finomítható, ha az egységeket további részekre bontjuk és figyelembe vesszük a redundáns párokat és a védelmi kapcsolásokat is. Többféle van: az *LSW/A* az akkumulátor túl- és alulfeszültség és -áram védelmére szolgál, az *LSW/B* egy áramkorlátozó kapcsoló, az *LSW/T* pedig egy túlfeszültségtől védő kapcsoló. [2] Az alacsony fogyasztású alkatrészeket áramkorlátozó kapcsoló helyett egyszerű RC tagokkal kötjük a szabályozott energiabuszra, az RTCC áramkör számára pedig szuperkapacitás biztosítja a redundáns energiaellátást. (Ezt a teljes rendszerterven az „RC+SC” rövidítés jelöli.)

Az OBC-t szétbonthatjuk háromfelé: a tényleges OBC-re (ami egy mikrokontroller), a hozzá tartozó flash memóriára és az RTCC-re. (ld. a fedélzeti számítógép részei c. fejezetet.) A COM rendszert még szétbonthatjuk a földi állomással kommunikáló egységre és a spektrumanalizátorra.

Megemlítendő még, hogy a két spektrumanalizátor és a két rádiókommunikációs rendszer egy-egy közös antennával rendelkezik. Ezen antennákat Dudás Levente készíti, és egymásra merőlegesen helyezkednek el a műhold külsején. [1]

### A SMOG-1 teljes rendszerterve



5. ábra: a SMOG-1 teljes rendszerterve

## Fedélzeti számítógép részei

### Mikrokontroller

A fedélzeti számítógép „lelke” egy PIC24 családba tartozó PIC24FJ256GA108 típusú mikrokontroller, melyet a Microchip gyárt. Ez tartalmaz egy saját kategóriájában jónak mondható RISC processzort (CPU) és megfelelő mennyiségű operatív memóriát (RAM). [4]

Ennek kiválasztásakor az egyik fő szempont az volt, hogy az OBC és az EPS-hez tartozó PCU (power control unit) ugyanolyan családba tartozzon, hogy a rajtuk futó szoftverek között lehessen kódot megosztani.

További szempontok az üzemi feszültségtartomány (2–5 V, bár az OBC energiaellátása maximum 3,6 V lesz, de a PCU-nak akár 5 V-ról is működnie kell), az alacsony fogyasztás (néhány mA) és a megfelelő teljesítmény (16 MIPS) voltak. Lehetséges alternatívák lehettek volna az ARM Cortex termékek (ilyeneket nagyon sok gyártó készít), vagy a Microchip PIC18 vagy PIC32 családjú termékei. A Cortex-M család ígéretesnek tűnt, azonban egyrészt sokkal szűkebb feszültségtartományban üzemel (nem sikerült belőle olyan modellt találni, ami 2–5 V-ig működik), valamint többet fogyaszt, mint a PIC24. A PIC18 belső architektúrájából kifolyólag ugyanazokat az utasításokat több órajelciklus alatt hajtja végre, mint a PIC24, valamint csak 8 bites, tehát nem lenne hatékony választás. A PIC32 pedig, habár nagy teljesítményű, a fogyasztása miatt nem megfelelő választás ehhez az alkalmazáshoz.

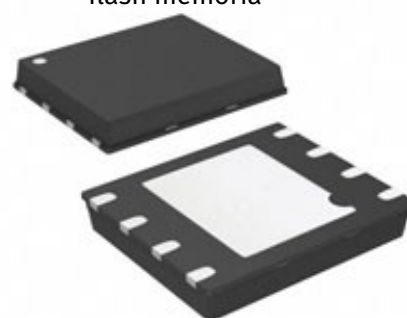
A mikrokontroller feladata, hogy az OBC fő szoftverét futtassa. Ez a szoftver fogja vezérelni az összes többi alrendszert (az energiaellátó rendszer kivételével), legfontosabbként említendő a kommunikációt, valamint gyűjt majd telemetriaadatokat az egész műhold összes alrendszere felől (beleértve az EPS-t és a napelem oldalakat is). Lásd bővebben a szoftver felépítéséről szóló fejezetet.

### Háttértár (flash memória)

Küldetéséből kifolyólag a SMOG-1 folyamatosan adatokat generál. Egyrészt a spektrumanalizátor kimenetét (ld. bevezető / küldetés), másrészt a műhold különféle alrendszereinek telemetriaadatait (ld.



6. ábra: a választott mikrokontroller



7. ábra: a választott flash memória

összeköttetések).

Fontos tehát, hogy legyen egy megbízható háttértár, ami kellő kapacitással rendelkezik. Szóba jönnek az EEPROM modulok és a flash memóriák. Lényeges szempont, hogy a memória ne fogyasszon sokat és lehetőleg a mikrokontrollerrel azonos (vagy tágabb) feszültségtartományban üzemeljen.

### Mekkora a kellő tárolókapacitás?

Először számoljuk ki, hogy a spektrumanalízátor méréseiből mekkora adatmennyiség keletkezik egy nap alatt. Tegyük fel, hogy egy mérés  $M = 2000$  byte adatmennyiséget jelent és  $T = 20$  sec ideig tart.

Az alábbi egyszerű számítással megkaphatjuk:

$$\frac{1 \text{ nap}}{T} \cdot M = \frac{24 \cdot 3600}{T} \cdot M \frac{\text{byte}}{\text{nap}} = \frac{24 \cdot 3600 \cdot 2000}{20} \frac{\text{byte}}{\text{nap}} = 8640000 \frac{\text{byte}}{\text{nap}} = 8.24 \frac{\text{Mbyte}}{\text{nap}}$$

Tehát egy nap alatt kicsit több, mint 8 Mbyte adat keletkezik (ha nem számítjuk a telemetriaadatokat). Vegyük figyelembe azt, hogy a SMOG-1 egy nap alatt körülbelül 4-6-szor halad el Magyarország felett, tehát ennyi lehetőségünk van rá, hogy az adatokat letöltsük. Arra nincs szükség, hogy hosszú ideig tároljuk őket a műholdon.

Így esett a választás az Adesto által gyártott AT45DB641E modulra [5], amely 64 Mbit (8 Mbyte) tárterülettel rendelkezik és képes az 1,7–3,6 V-os tartományban működni. További modulok, amiket találtunk, csak lényegesen nagyobb feszültségen (többnyire 3 V körül) kezdtek működni, vagy lényegesen szűkebb tartományban (pl. 1,65–1,95 V) működtek.

### RTCC (real-time clock & calendar)

A műholdon pozíció meghatározására való szenzor sajnos nem kaphat helyet, mert a kereskedelmi forgalomban kapható GPS áramköröket biztonsági okokból úgy készítik, hogy korlátokat építenek bele magasságra, sebességre és gyorsulásra vonatkozólag. Vagyis bizonyos magasság (altitude) felett „levágják” a mért értékeket és nem szolgálnak pontos adatokkal.

Ezen probléma azért jelentős, mert hiába mérünk a spektrumanalízátorral, ha nem tudjuk semmilyen módon megállapítani, hogy a mérést hol végezzük.

Erre ad megoldást az a megközelítés, hogy ha tudjuk az egyes mérések pontos idejét,



8. ábra: a választott RTCC áramkör

akkor a Földön a műhold pályájának kiszámolásával és a pontos idő figyelembevételével mégis meg tudjuk majd mondani, hogy melyik mérést hol végezte a műhold. Ezt a célt szolgálja az RTCC áramkör [6], amely SPI protokollon kommunikál a fedélzeti számítógép mikrokontrollerével, valamint bizonyos időközönként egy interrupt segítségével felébreszti azt.

**Megjegyzés.** Felmerülhet a kérdés, hogy miért nem használtuk valamely mikrokontrollerbe épített RTCC megoldást. Ennek oka az, hogy ezek a megoldások korántsem elég precízek, használatuk esetén a kontrollert nem lehet alvó módban üzemeltetni, valamint redundancia okokból ilyenkor a fedélzeti számítógép mindkét redundáns egységének egyszerre kellene működnie, hogy az egyikük meghibásodása esetén a másik se veszítse el a pontos időt. Ehelyett inkább egy rendkívül alacsony fogyasztású áramkört használunk, amelyet Géczy Gábor épített és amelynek redundáns energiaellátást is biztosítunk 11 mF-os szuperkapacitások segítségével.

## Szoftver felépítése

Az OBC szoftvere az ún. *event loop* mintát követi. (Elterjedt név még a *dispatcher loop*.) [7] Ennek lényege a következő: az OBC egy prioritásos sor (queue-t) tart fenn az elvégzendő feladatok számára. Valamely esemény bekövetkeztekor mindössze csak rögzíti, hogy újabb dolog vár feldolgozásra, aztán folytatja az addigi folyamatot. Amikor pedig éppen kiürült a prioritásos sor, akkor alvó üzemmódba helyezi magát a következő esemény bekövetkeztéig.

## Megvalósítás

Programozástechnikai szempontból a szoftver tulajdonképpen egy végtelen ciklus, ami azt figyeli, hogy milyen feladatok vannak a queue-ban, és sorban végrehajtja őket. Amikor elfogytak a feladatok, akkor alvó üzemmódba helyezi a processzort.

Új események beérkezése interrupt (megszakítás) által történik. Ilyenkor a CPU felébred az alvó üzemmódból (ha éppen aludt volna) és végrehajt egy ún. interrupt rutint (megszakításkezelő programrészletet). Ebben a bizonyos rutinban történik a feladatok hozzáadása a queue-hoz. Normál esetben az új feladatok a sor végére kerülnek, azonban bizonyos esetekben (például ha az adott feladat időzítése fontos) a sor elejére szűrődnek be.

## Analógia operációs rendszerekkel

Párhuzamot találhatunk az OBC szoftvere és egy operációs rendszer között. Az előbb részletezett *event loop* megvalósítás tulajdonképpen úgy működik, mint egy nagyon kezdetleges ütemező. Ha így nézzük, akkor az elvégzendő feladatok processzeknek, az OBC szoftvere pedig egyfajta operációs rendszernek fogható fel. Sajnos azonban hardveres korlátok miatt nem lenne gazdaságos ennél bonyolultabb rendszert kialakítani, mert a PIC24 processzorában nincs sem MMU (memory management unit), sem elég operatív memória ahhoz, hogy érdemes legyen több folyamatot párhuzamosan futtatni. [9]

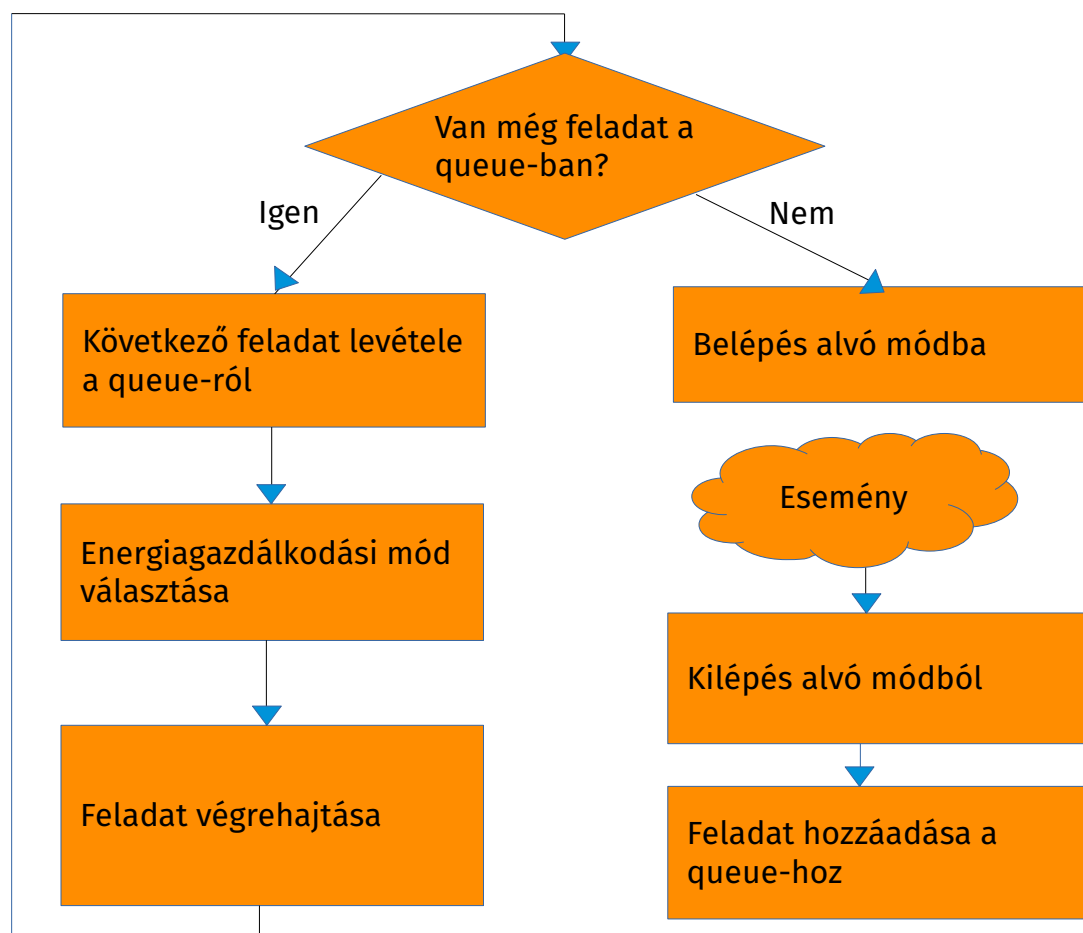
Természetesen lehetne preemptív multitaszkingot megvalósítani, úgy, hogy context switch esetén az operatív memória teljes tartalmát átmásoljuk a flash memóriába, de ez mikrokontrolleres hardveren lassú és gazdaságtalan folyamat lenne. [8, 9]

## Fájlrendszer, tömörítés

Az OBC-hez tartozik 8 Megabyte redundáns flash memória. Ezt a tárterületet használjuk majd a mért adatok és a telemetria tárolására. Amikor a műhold a földi állomással kommunikál, akkor innen olvassa majd ezeket az adatokat és küldi le őket.

Ehhez egy egyszerű fájlrendszert kell kialakítani, amely ACID tulajdonságokkal (atomicity, consistency, isolation, durability) rendelkezik majd, valamint a hely kihasználtság hatékonysága érdekében érdemes lesz az adatainkat tömöríteni.

## Folyamatábra



9. ábra: az OBC szoftverének magas szintű folyamatábrája

## Energiagazdálkodás

Az OBC szoftverét futtató mikrokontroller számos hasznos energiagazdálkodási lehetőséggel rendelkezik, amelyeknek mind jó hasznát vesszük a műholdon.

Mai mikrokontrollerekben már természetes lehetőség, hogy a bennük található integrált perifériákat tetszőlegesen ki lehet kapcsolni, igény szerint, amikor épp nincs rájuk szükség. [4]

Azonban a választott mikrokontroller a perifériák energiaellátásának szabályozása mellett három hasznos üzemmóddal is rendelkezik. [4]

## Alvó mód (sleep mode)

Ebben az üzemmódban a kontrollerben levő órajelgenerátor kikapcsol, vagyis megáll a CPU és minden olyan periféria is, amely ennek órajeléről jár. Ebben a módban lehet a legtöbb energiát megtakarítani, viszont egyúttal így a legkevésbé használható is a chip. [4]

Az OBC olyankor lesz ebben az üzemmódban, ha abszolút nincs semmi teendője és semelyik perifériájára sincsen szükség.

## Készenléti mód (idle mode)

A készenléti mód hasonlít az alvó módhoz abban, hogy a CPU ilyenkor kikapcsol, viszont az órajelgenerátora nem, ezért a kontrollerben levő perifériák továbbra is működőképesek maradnak ebben a módban. [4]

Az OBC-t akkor helyezzük ebbe az üzemmódba, ha éppen nincsen feldolgozható feladata, viszont valamelyik perifériájának működése még kívánatos. (pl. bizonyos időzítők vagy I/O portok működése folyamatban van, de nincs szükség a CPU mag bekapcsolására.)

## Szunnyókáló mód (doze mode)

Bizonyos feladatok esetén (pl. szinkron kommunikáció) szükség van arra, hogy a CPU és a perifériák órajele egymással szinkronban legyen, viszont nincs szükség rá, hogy a CPU ugyanolyan magas órajelről fusson, mint a perifériák. [4]

Ennek az üzemmódnak jó hasznát fogja venni az OBC olyan helyzetekben, amikor valamilyen lassú, szinkron kommunikáció van folyamatban.

# Összeköttetések

## Memória, RTCC és szenzorok

Az OBC-hez fizikailag is legközelebb eső áramkörök (ezek vele azonos nyomtatott huzalozású lemezen helyezkednek el) két SPI buszon fognak az OBC-vel kommunikálni. Ez úgy értendő, hogy egy OBC két különböző SPI buszon is master módban működik, és a redundáns párja is rajta lesz mindkét SPI buszon szintén master módban. Fontos megemlíteni, hogy habár mindkét OBC nem lesz egyszerre bekapcsolva (kettejük között ún. hideg redundancia valósul meg), a busz el lesz látva védődiódákkal és ellenállásokkal, hogy a kettejük egyidejű bekapcsolt állapota biztosan ne okozza a slave eszközök meghibásodását.

A memória, RTCC és szenzorok egyik példánya az egyik, másik példánya a másik SPI



buszon foglal majd helyet slave módban. Így, ha akár mindegyik fajta egység közül egy megsérül, a teljes rendszer még mindig üzemképes marad. [2]

Az RTCC egyúttal megszakítást is tud küldeni az OBC felé, hogy így segítse az egyes feladatok időzítését.

**Megjegyzés.** Habár a rendszerterven szerepelnek szenzorok, még nem döntött el, hogy pontosan milyen gyártmányú integrált áramköröket használunk majd erre a célra, és jelenleg még egyikünk sem dolgozik ezen a témán.

### PCU (power control unit)

A PCU szintén egy PIC24 családba tartozó mikrokontroller, ami az EPS-hez tartozik. Az EPS Géczy Gábor munkája, és a PCU feladatai közé tartozik egyes elemekről telemetriaadatok gyűjtése, az OBC-t ellátó LSW-k vezérlése, és annak eldöntése, hogy a redundáns pár közül melyik OBC üzemeljen. [2]

Az OBC és a PCU között módosított 1-wire digitális összeköttetés lesz, amelyen telemetriaadatokat és egyéb információkat cserélhetnek egymással.

### COM (rádiókommunikáció és spektrumanalizátor)

Rádiókommunikációért egy Silicon Laboratories gyártmányú Si1062 integrált áramkör lesz a felelős. Ez a chip tartalmaz a rádiófrekvenciás áramkörön kívül egy beépített 8051-es mikrokontrollert is, amely által „helyi intelligencia” valósítható meg a COM rendszerben. Ez az alrendszer Dudás Levente munkája. [1]

A spektrumanalizátor egy Si4464 típusú IC, amelyet az Si1062-ben levő mikrokontroller fog vezérelni SPI buszon keresztül. Ebben az elrendezésben tehát a földi állomással való kommunikáció és a spektrumanalizátor az OBC szemszögéből egyetlen egységes egészként viselkedik.

A COM és az OBC között 1-wire digitális összeköttetés lesz, amelyen spektrum mérési adatokat, földről vett, és a földre küldendő információkat cserélhetnek egymással.

### Áramkorlátozó kapcsolók

A fedélzeti számítógéphez tartozik két darab limiter switch (áramkorlátozó kapcsoló), amelyeket Géczy Gábor tervezett. Ezek a kapcsolók a COM rendszer egy-egy redundáns egységét védik és felügyelik. Afféle „okos biztosítóként” egy bizonyos megengedett áramérték felett kioldanak, így megakadályozva, hogy az adott áramkörben egy rövidzárlat energiaveszteséget okozzon. [2]

Ezek a kapcsolók nem csak az automatikus kioldásra valók, hanem ennek megtörténte esetén képesek digitális jelet adni a mikrokontroller felé, valamint áram- és

feszültségmérő kapcsolásokkal is el vannak látva.

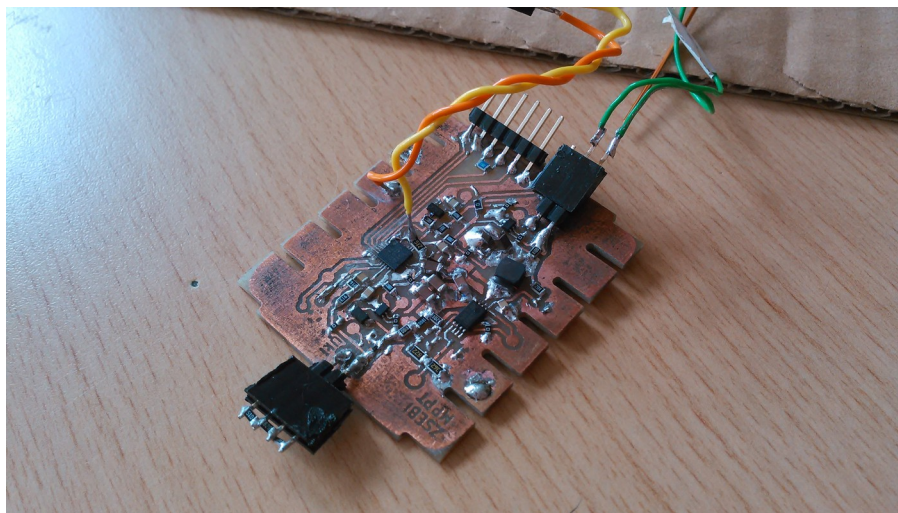
Az OBC-nek feladata két ilyen kapcsoló vezérlése és tőlük áram- és feszültségtelemetria mérése (analóg-digitál átalakítók segítségével) és rögzítése.

**Megjegyzés.** A fedélzeti számítógép mikrokontrollerei (a redundáns pár mindkét tagja) is ilyen kapcsolókkal vannak megvédve, de azokat a kapcsolókat az EPS vezérli.

### MPPT telemetria

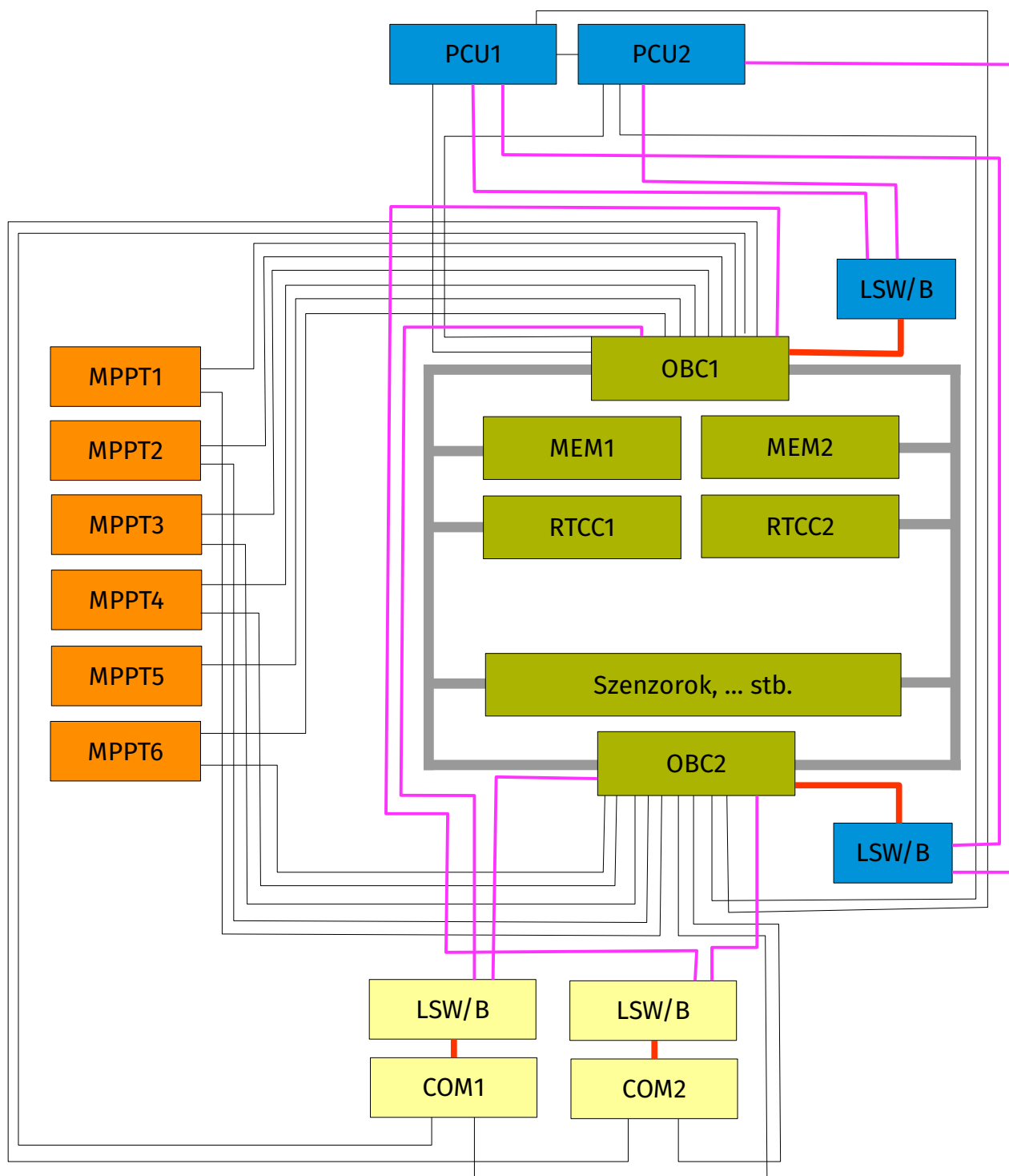
A műhold 5 cm-es oldalain napelemek vannak, amelyek optimális működéséről a Herman Tibor által tervezett MPPT áramkör gondoskodik. Ez az áramkör egyben el van látva feszültség- és árammérőkkel, amelyeket azok digitálisan rögzíteni képesek. [3]

Az OBC-nek feladata ezen MPPT áramkörök által mért telemetriaadatok lekérése és rögzítése.



10. ábra: egy napelemoldal belső részének prototípusa

## Összeköttetések blokkvázlata



- SPI digitális kommunikáció
- Energiaellátás
- Analóg telemetria és digitális vezérlés
- Módosított 1-wire digitális komm.

11. ábra: az OBC összeköttetései más alrendszerekkel

## Összefoglaló

A félév során tehát megismerkedtem a SMOG-1 küldetésével és tervezett felépítésével. Megismerkedtem az űreszközök tervezése során felmerülő kihívásokkal és ezek megoldásával, valamint a redundáns tervezés módszerével. Részt vettem a műhold rendszertervének kidolgozásában. Megterveztem a fedélzeti számítógép (on-board computer, OBC) működését, blokkvázlatát, és azt, hogy a műhold mely egyéb alrendszereivel hogyan lesz összeköttetésben. Megismertem a 1-wire és SPI protokollokat, valamint kidolgoztam, hogyan kommunikáljon az OBC a COM, MPPT, PCU, ... stb. rendszerekkel. Eldőlt, hogy a COM rendszernek helyi intelligenciára van szüksége, amit egy Si1062 áramkörrel valósítunk meg. Megválasztottam az alkatrészeket az OBC-hez, valamint megismerkedtem a szükséges fejlesztői eszközökkel (Simplicity Studio, MPLAB X) és megterveztem az OBC szoftveres architektúráját.

## Irodalomjegyzék

Jelen munkámhoz az alábbi publikációkat, írásokat használtam segítségül.

- [1] The Spectrum Monitoring System of Smog-1 Satellite – Dudás Levente, 2015
- [2] MSc Önálló laboratórium beszámoló – Géczy Gábor, 2015
- [3] MSc Önálló laboratórium beszámoló – Herman Tibor, 2015
- [4] PIC24FJ256GA108 adatlap – Microchip
- [5] AT45DB641E adatlap – Adesto Technologies
- [6] RV-3049-C3 adatlap – Micro Crystal Switzerland
- [7] An Object Behavioral Pattern for Demultiplexing Handles for Synchronous Events – Douglas C. Smith
- [8] Architektúrák jegyzet az Informatika 1. tárgyhoz – Dr. Móczár Géza
- [9] Operációs rendszerek jegyzet az Informatika 1. tárgyhoz – Dr. Kondorosi Károly