

Komunikace satelitu CubeSAT

Verze 3.0

napsal Ing. David Holas

prosinec 2005 – leden 2006



Obsah

1	Princip komunikace satelitu	3
1.1	Komunikační kanály satelitu	3
1.2	Palubní modul a palubní sběrnice	3
2	Implementace linkové vrstvy palubní komunikace	4
2.1	Komunikační zprávy	4
2.2	Příjem a vysílání komunikačních zpráv	4
2.3	Komunikační buffery	5
2.4	Malá a velká varianta implementace linkové vrstvy	6
3	Rozhraní linkové vrstvy palubní komunikace	7
3.1	Datové typy rozhraní linkové vrstvy	7
3.2	Funkce rozhraní linkové vrstvy	7
4	Aplikační vrstva z hlediska struktury programu a komunikace	8
5	Obsluha komunikace v hlavní smyčce	9
5.1	Malá varianta komunikace	9
5.2	Velká varianta komunikace	10
6	Vysílání zpráv a výměna dat mezi moduly na palubě	11
7	Logické adresy palubních modulů	11
7.1	Použití přidělené logické adresy v modulu	11
7.2	Moduly s jednou komunikační adresou	11
7.3	Moduly podporující více komunikačních adres	12
7.4	Rozdělení adres mezi moduly	12
8	Komunikace ze Země	12
8.1	Princip komunikačního spojení	12
8.2	Diagnostický rámec majáku	12
8.3	Vrstvy komunikace	13
9	Obecné datové zprávy podporované všemi moduly	13
9.1	Datová zpráva „Status modulu“	13
9.2	Datová zpráva „Odešli datovou zprávu“	13
9.3	Datová zpráva „Záznamy o význačných událostech - skupina“	14
9.4	Datová zpráva „Záznamy o význačných událostech - jedna“	14
9.5	Datová zpráva „Záznamy o průběhu veličiny“	15
9.6	Datová zpráva „Statistiky komunikace - skupina“	15
10	Datové zprávy specifické pro modul Napájecího zdroje	16
10.1	Datová zpráva „Stav a výhled palubního napájení“	16
10.2	Datová zpráva „Oznámení o plánu vypnutí“	16
11	Datové zprávy specifické pro modul Stabilizace	16
12	Datové zprávy specifické pro modul Fotoaparát	16
13	Datové zprávy specifické pro modul Tether	16
14	Datové zprávy specifické pro modul Maják	16
15	Datové zprávy specifické pro modul Komunikace	16

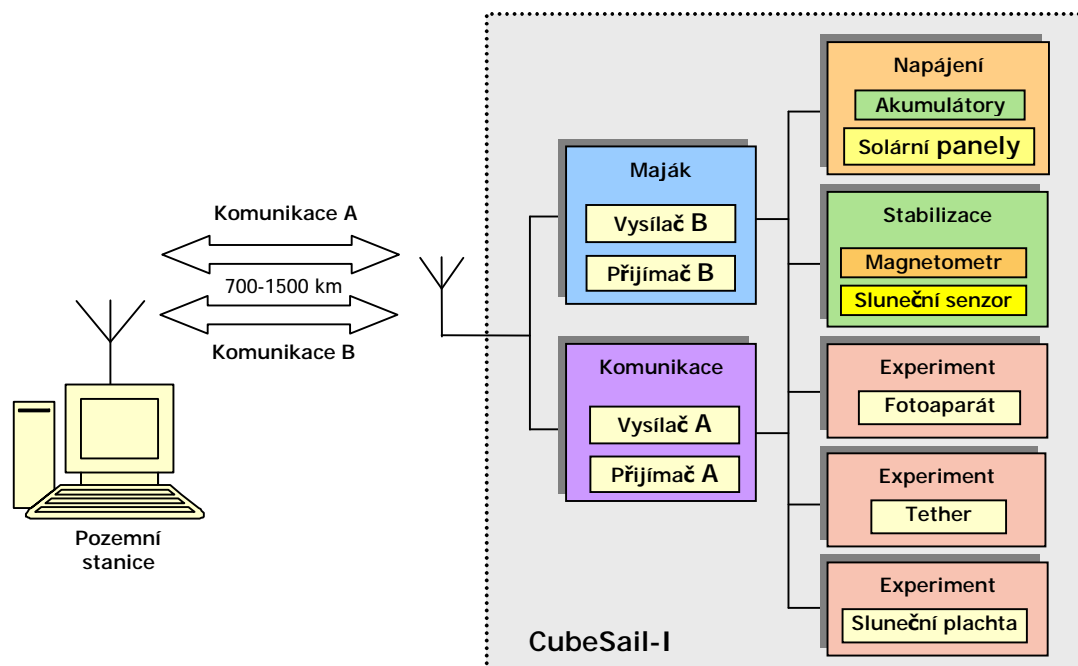
1 Princip komunikace satelitu

Satelit se skládá z nezávislých inteligentních částí, tzv. **modulů**. Moduly mají společné napájení, které spravuje jeden z nich (Power Module). Mezi sebou jsou moduly propojeny elektricky palubní komunikační sběrnici. Sběrnice slouží k výměně dat mezi jednotlivými moduly na palubě a především k **přístupu k datům modulů ze Země**.

1.1 Komunikační kanály satelitu

Komunikačním prostředkem mezi pozemním střediskem a palubními systémy jsou dva komunikační moduly.

1. **hlavní komunikační modul**, který poskytuje komunikační kanál s větší šířkou pásma. Hlavní komunikace, zejména přenosy náročné na objem dat (přenos záznamů průběhu stabilizace, přenos dat fotografií a podobně).
2. **záložní komunikační modul**, který je integrován společně s majákem. Kromě periodického vysílání stavového rámce (radiomaják) slouží jako záložní komunikační kanál s nízkou šířkou pásma. V tomto režimu se principiálně neliší od hlavní komunikační linky.



Palubní propojení modulů zajišťuje **palubní komunikační sběrnice**. Každý z palubních modulů je na ni elektricky připojen, což mu umožňuje příjem a vysílání dat z této sběrnice.

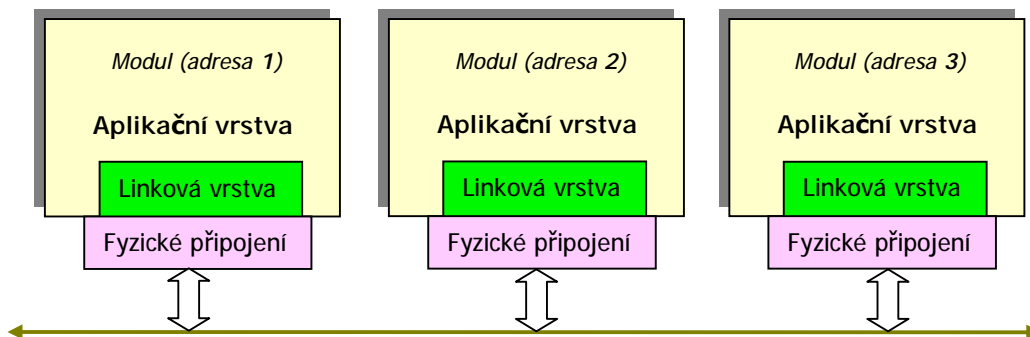
1.2 Palubní modul a palubní sběrnice

Obsluhu této sběrnice a řízení jejího komunikačního protokolu zajišťuje v každém připojeném modulu softwarová část, zvaná **linková vrstva**. Ta poskytuje rozhraní pro hlavní výkonnou část programu konkrétního modulu, tj. pro jeho **aplikační vrstvu**. Při startu modulu provede aplikační vrstva inicializaci linkové vrstvy a poté periodicky volá funkce rozhraní pro zpracování přijatých dat a podle potřeby pro odeslání potřebných dat.

Konkrétní specifikace rozhraní linkové vrstvy a princip práce s ním je popsán dále.

Každý modul na palubě má dohodou pevně přidělenou logickou adresu. Kromě toho existují **dvě virtuální adresy**, pro pozemní stanici komunikující prostřednictvím hlavního a záložního komunikačního kanálu. Data (žádosti a povely) vysílané z pozemní stanice se na vnitřní sběrnici projeví, jako **by byla zaslána virtuální adresou**, podle toho, kterým komunikačním kanálem byla přijata. Za to je zodpovědný komunikační modul, který rádiovou výzvu přijal.

Odpověď na tyto žádosti je na vnitřní sběrnici adresována příslušné virtuální adrese, kterou zachytí příslušný komunikační modul a zařadí ji k odeslání svým rádiovým kanálem k Zemi.



Data si moduly mezi sebou vyměňují prostřednictvím datových zpráv. Každá zpráva je určena svou délkou, adresou odesílatele, adresou adresáta, svým kódem a datovým obsahem. Z pohledu aplikační vrstvy se jedná o pole bajtů.

Konkrétní specifikace zpráv, jejich kódů a struktury datových částí je uvedena v další části tohoto popisu. Zpráva může být po sběrnici zaslána na konkrétní cílovou adresu, nebo na všeobecnou adresu jako **broadcast**. V takovém případě ji přijmou všichni účastníci, s výjimkou virtuálních adres (broadcasty se nepřenášejí do pozemního střediska).

Moduly, které měří hodnoty a veličiny, zajímavé i pro ostatní palubní moduly, tyto hodnoty periodicky zasílají všem palubním modulům ve formě broadcastu. Například modul stabilizace, jehož součástí je magnetometr pro měření vektoru vnějšího magnetického pole, zasílá periodicky aktuálně měřený vektor ve speciální datové zprávě.

2 Implementace linkové vrstvy palubní komunikace

Linková vrstva je implementována v assembleru procesoru x51. Je samozřejmě možné ji přepisem upravit do jazyka jiného mikroprocesoru, vyžaduje pouze ovládání jednoho sériového kanálu v přerušení a jeden časovač 1000 Hz.

2.1 Komunikační zprávy

Z hlediska linkové vrstvy je jednotkou komunikace jedna zpráva, tj. jeden obsah datového rámce. Strukturu obsahu datové části zprávy linková vrstva nezkoumá, pouze kontroluje délku a hlídá správný formát rámce. Dále je linková vrstva zodpovědná za to, že přijme a do aplikační vrstvy předá jen ty zprávy, jejichž cílová adresa odpovídá adrese modulu, nebo zprávy určené všem adresám.

Z pohledu aplikační vrstvy je datová zpráva pole bajtů s touto strukturou:

Offset 0:	1 x BYTE	délka zprávy
Offset 1:	1 x BYTE	adresa, které je zpráva určena
Offset 2:	1 x BYTE	adresa, která zprávu odeslala
Offset 3:	1 x BYTE	kód datové zprávy
Offset 4:	N x BYTE	obsah datové části zprávy

Kód datové zprávy má u žádostí (tj. přijatých zpráv, na které je požadována odpověď) **nejvyšší bit nulový**, kód datové zprávy odpovědi (tj. datové zprávy odesílané na základě přijetí žádosti nebo spontánně jako broadcast) má v **nejvyšším bitu jedničku** (tj. kód žádosti + 128).

2.2 Příjem a vysílání komunikačních zpráv

Celý proces komunikace a příjmu a odesílání zpráv probíhá **na pozadí** vykonávání hlavního programu modulu. Úkolem aplikační vrstvy pro příjem je pouze se periodicky dotazovat, zda není k dispozici nová zpráva.

Když aplikační vrstva přijímá vstupní zprávu, dostává od funkce rozhraní linkové vrstvy pointer na paměť, kde se nachází datová zpráva s výše uvedenou strukturou. Tuto paměťovou strukturu může uživatelská vrstva zpracovávat. Když zpracování dokončí, oznámí to linkové vrstvě.

Podobně při odesílání si aplikační vrstva vyžádá od linkové vrstvy pointer na paměť, v níž vytvoří zprávu s touto strukturou a oznámí linkové vrstvě odeslání. Linková vrstva posléze na pozadí zajistí ve vhodném okamžiku odeslání zprávy.

Paměťové úseky, které linková vrstva předává hlavnímu programu s přijatou zprávou, nebo pro přípravu zprávy k vyslání, se nazývají komunikační buffery.

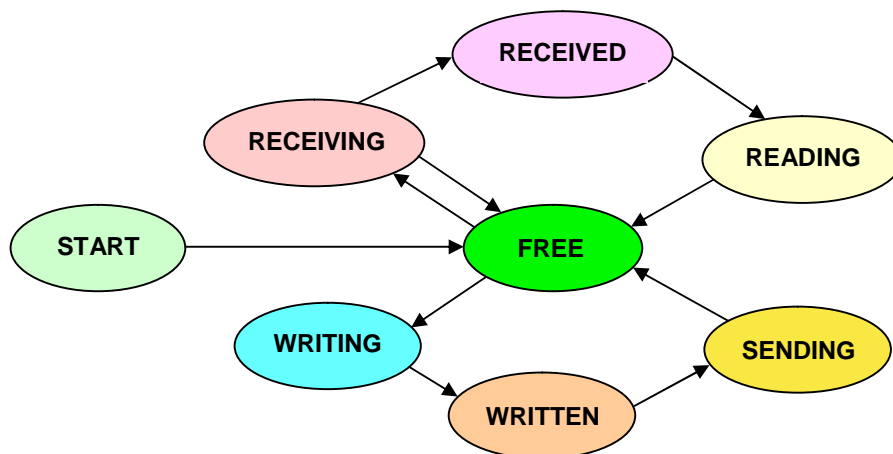
2.3 Komunikační buffery

Linková vrstva implementuje jeden nebo více těchto bufferů. Jeden komunikační buffer je dimenzován tak, aby mohl obsahovat právě jednu komunikační zprávu. Každý buffer linkové vrstvy se může nacházet v jednom z následujících stavů:

- FREE** buffer je prázdný, neobsahuje žádnou zprávu. Může se použít buď k přijímání nové zprávy ze sběrnice (obsadí jej linková vrstva a označí jako RECEIVING), nebo aplikační vrstvou k přípravě nové zprávy k odeslání (označen jako WRITING).
- RECEIVING** do bufferu se právě přijímá nová zpráva ze sběrnice. Pokud se příjem zprávy dokončí úspěšně, bude buffer označen jako RECEIVED a bude k dispozici aplikační vrstvě k vyzvednutí. Pokud příjem skončí neúspěšně (teoreticky timeout, například vypnutí napájení odesílacího modulu a podobně), bude blok označen jako FREE a zpráva nebude přijata.
- RECEIVED** buffer obsahuje novou zprávu přijatou ze sběrnice. Buffer je připraven k vyzvednutí a zpracování aplikační vrstvou (připraven k přechodu do READING).
- READING** aplikační vrstva tento buffer vyzvedla ke zpracování, ale ještě zpracování neukončila. Zpracování právě probíhá. Jakmile aplikační vrstva zpracování dokončí a signalizuje toto linkové vrstvě, bude buffer označen jako FREE a bude znovu k dispozici.
- WRITING** buffer byl ze stavu FREE vyzvednut aplikační vrstvou, která potřebuje připravit novou zprávu k odeslání (buď jako odpověď na dříve přijatou zprávu, nebo sama aktivně potřebuje odeslat novou). Probíhá zápis do bufferu a příprava odesílací zprávy. Jakmile aplikační vrstva přípravu zprávy dokončí a signalizuje to linkové vrstvě, bude buffer označen jako WRITTEN a připraven k odeslání.
- WRITTEN** buffer obsahuje zprávu připravenou k odeslání do sběrnice. Jakmile nastane okamžik, kdy tento modul bude smět vysílat (dle pravidel definovaných linkovou vrstvou), začne jeho odesílání a přejde do stavu SENDING.
- SENDING** buffer se právě odesílá do sběrnice. Pokud vysílání skončí úspěšně, blok se označí jako FREE a zpráva bude vyslána. Pokud dojde při odesílání k chybě (například teoreticky timeout přerušení), odesílání se přeruší, buffer přejde do stavu FREE a zpráva nebude vyslána.

Buffer přechází mezi stavy na základě akcí linkové vrstvy (obsluha zahájí příjem rámce, zahájí vysílání rámce) a volání funkcí rozhraní aplikační vrstvou (vyzvednutí bufferu z příjmu, vyzvednutí nového bufferu pro přípravu odesílané zprávy).

Diagram stavových přechodů bufferů je znázorněn na následujícím obrázku:



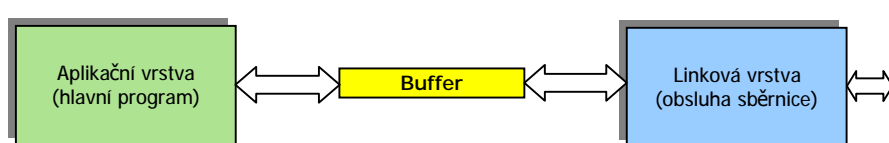
2.4 Malá a velká varianta implementace linkové vrstvy

Originální implementace linkové vrstvy je zapsána v syntaxi makroassembleru Keil ASM51. Na začátku zdrojového textu se nacházejí makra, pomocí kterých je možné ovlivnit překlad.

Zásadní makra jsou SMALL_KOMBUS a LARGE_KOMBUS.

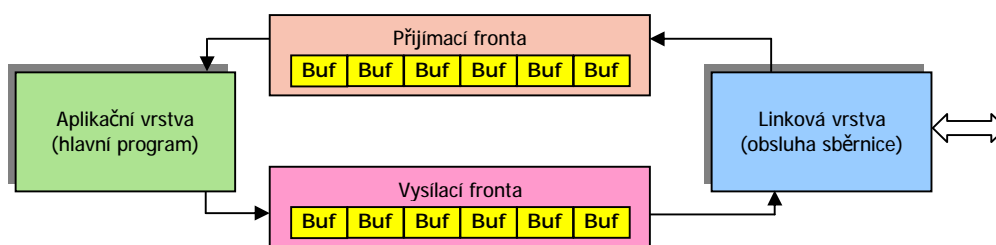
Jestliže je aktivní SMALL_KOMBUS, je implementována tzv. **malá varianta komunikace**. Ta se vyznačuje tím, že obsahuje pouze **jeden malý buffer** (max. délka dat XX bajtů), který je sdílen podle potřeby pro příjem i vysílání zpráv. Malou variantu komunikace může použít palubní modul, u něhož nepřipadá v úvahu intenzivní komunikace. Je totiž zřejmé, že například v okamžiku, kdy aplikační vrstva obsadí buffer a připravuje zprávu k odeslání, nemůže začít do tohoto bufferu příjem. Pokud by právě v tomto okamžiku začala ze sběrnice přicházet zpráva pro tento modul, došlo by k její ztrátě (linková vrstva zprávu sice přijme, ale nikam neuloží, tj. z hlediska aplikační vrstvy jako by zpráva vůbec nepřišla).

Výhoda malé varianty komunikace je v tom, že je menší z hlediska velikosti kódu (s jedním bufferem se manipuluje jednodušeji) a vyžaduje jen málo paměti (v případě procesoru x51 si vystačí s bufferem v interní RAM a nepotřebuje externí paměť).



Další omezení malé varianty komunikace je v tom, že nemůže přijmout dlouhé datové zprávy, protože je na jejich uložení nemá buffer dostatečnou velikost. Pokud přichází ze sběrnice zpráva, která by se do rezervovaného prostoru nevešla, uloží se pouze prvních XX bajtů, další bajty se neuloží, ale zpráva se přijme. Aplikační vrstva musí respektovat to, že ačkoliv je u přijímané zprávy vyplněna správná délka, platný je pouze začátek zprávy.

Varianta LARGE_KOMBUS znamená velkou variantu linkové vrstvy, která se vyznačuje tím, že místo jednoho společného bufferu implementuje **fronty bufferů pro vysílání a pro příjem**. Příchozí buffery se řadí do fronty, odkud je vyzvedává aplikační vrstva a po zpracování je opět uvolňuje. Podobně vysílací fronta umožňuje, aby v jednom okamžiku aplikační vrstva vyzvedla a začala připravovat k odeslání více zpráv, nebo aby je připravovala rychleji (po omezenou krátkou dobu danou kapacitou fronty), než je bude linková vrstva v daném okamžiku schopna odesílat do sběrnice.



Nevýhodou této varianty je větší velikost kódu (nutnost implementace práce s frontami a synchronizace) a větší spotřeba paměti RAM. Každý buffer musí být dimenzován na maximální možnou velikost zprávy

3 Rozhraní linkové vrstvy palubní komunikace

Hlavní část palubního SW (aplikační vrstvy jednotlivých modulů satelitu) bude napsána v jazyce C. Tomu je přizpůsobeno i rozhraní linkové vrstvy. Datové typy a prototypy funkcí rozhraní jsou uvedeny v hlavičkovém souboru `KomBus.h`

3.1 Datové typy rozhraní linkové vrstvy

`KOM_MSGPTR` je datový typ deklarovaný v `KOMBUS.H` jako pointer na bajty zprávy. Pointery tohoto typu vystupují v parametrech a návratových hodnotách funkcí rozhraní. Pro x51 implementaci je tento typ "BYTE idata *" pro malou variantu a "BYTE xdata *" pro velkou variantu komunikace.

3.2 Funkce rozhraní linkové vrstvy

`KOM_Inicializace` tuto funkci musí aplikační vrstva vyvolat na začátku, ještě před voláním některé z následujících funkcí. Způsobí inicializaci hardwarového rozhraní (naprogramování fyzického sériového kanálu) a inicializaci všech interních proměnných a datových struktur linkové vrstvy. Funkci není možné volat opakovaně. Funkce nemá parametry a nevrací žádnou hodnotu.

`KOM_TestVstupniZpravy` vrací příznak, zda má linková vrstva k dispozici novou zprávu z příjmu. Nemá parametry, návratová hodnota je typu `BOOL` (v případě x51 bit). Pokud vrátí hodnotu `TRUE`, znamená to, že je k dispozici nová zpráva a měla by být neprodleně vyzvednuta pomocí funkce `KOM_PripravitVstupniZpravu`, zpracována a uvolněna.

`KOM_PripravitVstupniZpravu` tato funkce se smí volat pouze v případě, že funkce `KOM_TestVstupniZpravy` vrátila `TRUE`. Nemá parametry a vrací ukazatel na novou, vstupní zprávu. Tento ukazatel použije aplikační vrstva ke zpracování příchozí zprávy. Až zpracování dokončí, musí vyvolat funkce `KOM_UvolnitVstupniZpravu`. Pokud by tuto funkci nevyvolala, buffer se zprávou by zůstal navždy ve stavu `READING` a pro další komunikaci by byl ztracen.

`KOM_UvolnitVstupniZpravu` aplikační vrstva tuto funkci volá **povinně** po dokončení zpracování zprávy, předané předchozí funkcí. Je žádoucí, aby etapa zpracování zprávy proběhla co nejrychleji, protože po dobu, kdy se vstupní zpráva zpracovává, není možné buffer použít k příjmu nové zprávy, takže pokud během zpracování vstupní zprávy začne ze sběrnice přicházet nová, musí se pro její příjem použít další buffer z přijímací fronty, pokud je k dispozici. Pokud k dispozici není (zpracování trvá tak dlouho, že se stihne přijímací fronta naplnit nebo je použita malá varianta s jedním bufferem), přijímaná zpráva se ignoruje a dojde k její ztrátě. Funkce `KOM_UvolnitVstupniZpravu` má jeden parametr, a to ukazatel na buffer, který se uvolňuje. Nevrací žádnou hodnotu.

`KOM_PripravitVystupniZpravu` tuto funkci aplikační vrstva volá v případě, že potřebuje připravit k odeslání novou zprávu. Funkce nemá parametry a vrací hodnotu typu `BOOL` (v případě x51 bit). Funkce nemusí skončit s úspěchem, buffer pro odeslání nemusí být v tomto okamžiku k dispozici, v takovém případě musí aplikační vrstva volání této funkce opakovat. Pokud funkce skončí úspěšně, pak vrátí `TRUE`. Pointer na paměťový buffer,

kam může aplikační vrstva připravit zprávu k odeslání, získá aplikační vrstva voláním funkce `KOM_PointerVystupniZpravy`. Platí, že po úspěšném volání této funkce musí být později vyvolána funkce `KOM_OdeslatVystupniZpravy`.

KOM_PointerVystupniZpravy funkce vrací pointer na datový buffer výstupní zprávy. S pamětí, na kterou ukazuje tento pointer, může aplikační vrstva přímo pracovat a sestavit výstupní zprávu. Funkce nemá parametry a vrací pointer.

KOM_PripravSablonuOdpovedi tato pomocná funkce slouží ke snazšímu vyplnění bufferu pro odpověď. Připraví hlavičku ve výstupním bufferu podle vstupní zprávy. Délku zachovává stejnou, jako vstupní zpráva, přehodí adresu odesílatele a adresáta a vyplní kód zprávy (= kód vstupní zprávy + 0x80). Funkce má dva parametry, a to pointer na vstupní zprávu, která je vzorem, a pointer na buffer výstupní zprávy, která je cílem. V případě malé varianty komunikace se pracuje s jedním bufferem, parametry jsou shodné. Funkce nevrací žádnou hodnotu.

KOM_UvolnitVstupniPripravitVystupniZpravy funkce slouží pro uvolnění vstupní a zároveň pro přípravu výstupní zprávy. Funkce je důležitá v případě malé varianty komunikace, protože zaručuje, že společný buffer, který je uvolňován, bude hned použit k odeslání a nestane se, že by jej hned obsadila přijímací rutina pro příjem ze sběrnice. Funkce má jeden parametr a nevrací žádnou hodnotu. Pointer na nově obsazený buffer se získá pomocí volání `KOM_PointerVystupniZpravy`.

KOM_OdeslatVystupniZpravy slouží pro oznámení linkové vrstvě, že výstupní buffer je vyplněn a připraven k odeslání. Funkce má jeden parametr, a to pointer na výstupní zprávu. Nevrací žádnou hodnotu.

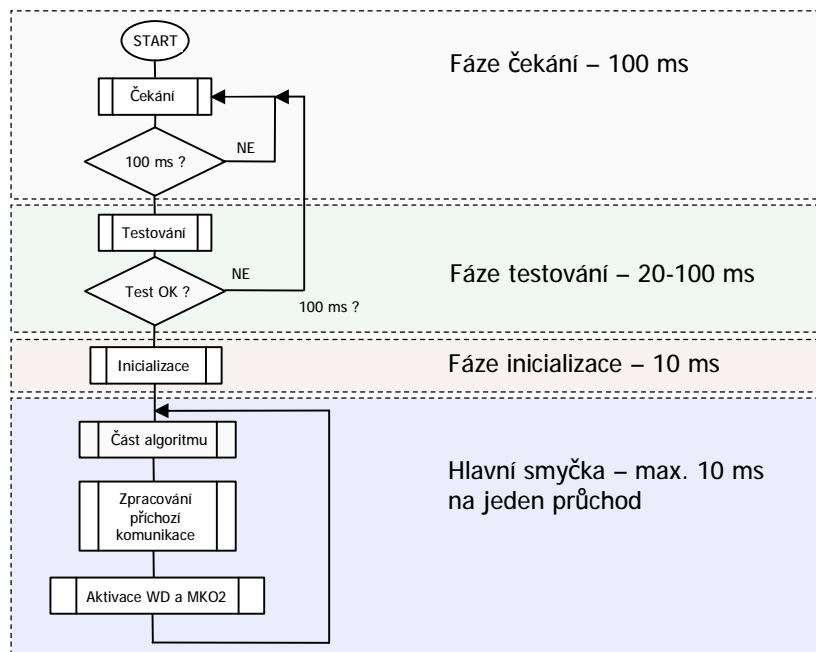
4 Aplikační vrstva z hlediska struktury programu a komunikace

Po zapnutí napájení modulu a nastartování řídicí jednotky modulu (řídicího procesoru) provede hlavní program modulu tyto činnosti:

1. **Fáze úvodního čekání** – do této fáze vstoupí program okamžitě po nastartování procesoru. Hned po startu musí procesor po nějakou dobu (například 100ms) pasivně čekat. Je to prevence proti kolísání napájecího napětí, kdy, pokud by došlo k přerušení dodávky těsně po zapnutí.
2. **Fáze testování** – vykonává se těsně před start-up kódem dodaným překladačem. Cílem je provést základní testy procesoru, funkčnost registrů, aritmetické jednotky, příznakového registru a podobně. Pokud by tyto základní testy skončily neúspěšně, nesmí procesor vůbec spustit hlavní program a musí přejít do nekonečné smyčky a očekávat RESET watch-dogem.

Součástí testovací fáze je i test kontrolních součtů paměti programu. Program musí počítat se situací, kdy dojde k porušení hlavního programového kódu. Pokud tato situace nastane, nesmí program postoupit do další fáze a musí přejít do nekonečné smyčky a očekávat watch-dog.

3. **Fáze inicializace** – nachází se na začátku uživatelského programu. Program provede:
 - a) inicializaci všech svých proměnných a datových struktur
 - b) poté inicializaci pomocných procesů vykonávaných v přerušení (jedním z těchto procesů je linková vrstva komunikační sběrnice, tj. aplikační vrstva vyvolá funkci `KOM_Inicializace`).
 - c) povolení všech přerušení
4. **Hlavní smyčka programu** – představuje hlavní předmět činnosti modulu. Hlavní smyčka se skládá z těchto částí:
 - a) vykonání části algoritmu
 - b) zpracování příchozích datových zpráv z linkové vrstvy
 - c) aktivace Watch-Dog a bezpečnostního MKO fyzického rozhraní sběrnice



Algoritmus modulu je třeba navrhnout tak, aby jeho část, vykonávaná v jednom průběhu hlavní smyčky, netrvala příliš dlouho. Procesor by měl být schopen vykonat alespoň řádově stovky průběhů hlavní smyčkou za sekundu. Tím se minimalizuje doba odezvy na příchozí komunikační zprávu.

Důležité je, že aktivace Watch-Dog a aktivace bezpečnostního MKO je v celém programu umístěna na jednom místě, a to na konci hlavní smyčky. Tímto opatřením se minimalizuje šance na to, že v případě neplánované havárie programu se při nekontrolovaném vykonávání instrukcí bude Watch-Dog periodicky vybavovat a procesor skončí ve stavu, kdy bude muset zasáhnout Power Module a resetovat jej vypnutím a zapnutím napájení.

5 Obsluha komunikace v hlavní smyčce

Úkolem obsluhy komunikace v hlavní smyčce je vyzvedávání příchozích zpráv z linkové vrstvy a případně odpověď na ně. V jednom průchodu hlavní smyčkou smí program obsloužit jen určitý maximální počet příchozích zpráv, a zároveň strávit obsluhou komunikace jen určitý maximální čas. V případě malé varianty komunikace nemůže být k dispozici nikdy více než jeden buffer, proto žádné omezení není třeba.

5.1 Malá varianta komunikace

Obsluha se liší pro malou a velkou variantu komunikace. Rozdíl spočívá v tom, že malá varianta má pouze jeden buffer. Paměť s příchozí zprávou a paměť pro přípravu odesílané zprávy je identická. Dokud aplikační vrstva neuvolní buffer s příchozí zprávou, nemůže úspěšně vyzvednout buffer pro odchozí zprávu. Mohlo by se ovšem stát, že okamžitě po uvolnění vstupní zprávy začne přicházet nová zpráva ze sítě a linková vrstva by uvolněný buffer obsadila ještě dříve, než by aplikační vrstva stihla buffer rezervovat pro vysílání. Pokud ovšem aplikační vrstva buffer vyzvednout stihne, může se zase stát, že během přípravy odesílací zprávy se nová příchozí zpráva nebude moci přijmout a ztratí se.

Tato situace nemá ze své podstaty ideální řešení. Preferována je varianta, kdy se podaří odpovědět na první vstupní zprávu. Proto je v malé variantě komunikace k dispozici funkce rozhraní, která uvolňuje příchozí buffer a zároveň alokuje buffer pro odesílací zprávu. Vše se děje v rámci jedné operace linkové vrstvy, takže buffer pro odesílání je zaručeně k dispozici. Funkce se jmenuje KOM_UvolnitVstupniPripavitVystupniZpravu.

Postup zpracování zpráv v hlavní smyčce pro malou variantu komunikace:

1. Začátek hlavní smyčky

2. Vykonání části algoritmu aplikační vrstvy, max. cca 10 ms
3. Test, zda je k dispozici vstupní zpráva, pokud ne, pokračuje se na bodu 11
4. Vyzvednutí vstupní zprávy
5. Test podporovaných kódů vstupních zpráv tohoto modulu. Pokud to není některý z podporovaných kódů, modul zprávu ignoruje. Uvolnění vstupní zprávy a pokračuje se na bodu 11
6. Zpracování obsahu příchozí zprávy. Pokud je obsah nějak nekonzistentní nebo nedává smysl, zprávu aplikační vrstva ignoruje. Uvolnění bufferu a pokračuje se na bodu 11.
7. Pokud je zpráva broadcast nebo zpráva nevyžaduje potvrzovací odpověď, neodpovídáme. Uvolnění vstupní zprávy a pokračuje se na bodu 11
8. Uvolnění vstupní a příprava bufferu pro výstupní zprávu jedním voláním speciální funkce rozhraní XX.
9. Příprava (vyplnění) zprávy k odeslání v paměti.
10. Uvolnění zprávy k odeslání do linkové vrstvy
11. Konec zpracování komunikace, pokračuje se bodem 1.

5.2 Velká varianta komunikace

V případě velké varianty komunikace je situace odlišná. Buffery pro odesílání a pro příjem jsou odděleny ve zvláštních frontách, nijak se nemísí. Může se stát, že se během jednoho cyklu hlavní smyčky nahromadí více vstupních zpráv, proto je potřebná kontrola na maximální počet zpracovaných zpráv a na maximální čas strávený komunikací v hlavní smyčce.

Dále platí, že buffer pro odesílání lze vyzvednout nezávisle na tom, zda je právě vyzvednutý nějaký buffer z příjmu. Díky tomu může být během přípravy výstupní zprávy neustále k dispozici pointer na vstupní zprávu a není nutné si významné hodnoty ze vstupní zprávy poznamenávat do zvláštních proměnných.

Postup zpracování zpráv v hlavní smyčce pro malou variantu komunikace:

1. Začátek hlavní smyčky
2. Vykonání části algoritmu aplikační vrstvy, max. cca 10 ms
3. Nulování pomocného počítadla zpracovaných zpráv, začátek počítání času stráveného zpracováním vstupních zpráv.
4. Test, zda je k dispozici vstupní zpráva, pokud ne, pokračuje se na bodu 15.
5. Vyzvednutí vstupní zprávy
6. Test podporovaných kódů vstupních zpráv tohoto modulu. Pokud to není některý z podporovaných kódů, modul zprávu ignoruje. Přejít na bod 12.
7. Zpracování obsahu příchozí zprávy. Pokud je obsah nějak nekonzistentní nebo nedává smysl, zprávu aplikační vrstva ignoruje. Pokračuje se bodem 12.
8. Pokud je zpráva broadcast nebo zpráva nevyžaduje potvrzovací odpověď, neodpovídáme. Pokračuje se bodem 12.
9. Vyzvednutí bufferu pro potvrzovací odpověď. Pokud není buffer k dispozici, nelze v tomto okamžiku odpovědět, ale povel bude zpracován. Pokračuje se bodem 12.
10. Příprava (vyplnění) zprávy k odeslání v paměti, pointer na vstupní zprávu je stále k dispozici.
11. Uvolnění výstupní zprávy k odeslání do linkové vrstvy
12. Uvolnění vstupní zprávy, která se právě zpracovává.
13. Zvýšení počtu zpracovaných vstupních zpráv v tomto cyklu hlavní smyčky.
14. Kontrola počtu zpracovaných zpráv a kontrola celkového času stráveného komunikací v tomto průchodu hlavní smyčkou. Pokud nebyl ještě zpracován maximální povolený počet a zároveň ještě neuplynul maximální čas, pokračujeme bodem 4.
15. Konec zpracování komunikace, pokračuje se bodem 1.

Nastavení parametru maximálního počtu zpracovaných zpráv a maximálního času stráveného komunikací je závislé na výkonu řídicího procesoru. Konstanty je třeba volit tak, aby maximální čas byl pokud možno do 10 milisekund a zároveň počet zpráv byl alespoň 1-2. Obecně platí, že čím větší čas stráve-

ný komunikací a čím větší počet zpráv zpracovávaných najednou (a čím kratší trvání jednoho cyklu hlavní smyčky), tím nižší latence a vyšší komunikační propustnost modulu.

6 Vysílání zpráv a výměna dat mezi moduly na palubě

Mezi moduly na palubě probíhají tyto komunikační procesy:

1. rozesílání stavových a diagnostických informací
2. přenos dat na vyžádání mezi moduly
3. komunikace virtuálních adres, tj. zprostředkovaná komunikace pozemní stanice

Vznikne-li v modulu **požadavek spontánního odeslání zprávy** (tedy nikoliv odpověď na přijatou žádost, ale samovolně vzniklá žádost do jiného modulu), je nastaven SW příznak odeslání zprávy. Je-li tento příznak nastaven, pak se v každém cyklu hlavní smyčky pokouší program vyzvednout z linkové vrstvy buffer zprávy pro odeslání.

Pokud se vyzvednutí nepodaří (všechny buffery jsou obsazené), pak program v hlavní smyčce **opakuje pokus** tak dlouho, dokud se vyzvednutí nepodaří (za normálního průběhu komunikace k tomu dojde brzy, maximálně v řádu desetin sekundy).

Modul může implementovat **bezpečnostní funkci**, kdy průběh času, po který se nedaří vyzvednout buffer k odeslání, sleduje pomocným časovačem. Pokud se nedaří odeslat zprávu po určitý dlouhý čas (např. 2 sekundy), znamená to poruchu, a to buď softwarovou (poškození datových struktur seznamu bufferů linkové vrstvy), nebo hardwarovou poruchu (nefunkční sériové rozhraní).

Bude-li takováto situace detekována, je jedním z možných řešení samočinný reset modulu, například tím, že zakáže veškerá přerušeni, zahájí nekonečnou smyčku a přestane vybavovat externí watch-dog. Tím po krátké době dojde k jeho aktivaci a procesor modulu dostane signál RESET.

7 Logické adresy palubních modulů

Logické adresy modulů jsou určeny **závaznou dohodou**. Každý modul a jeho konstrukční skupina má **pevně přidělené číslo adresy**, které musí **bezpodmínečně dodržet**. Konflikt adres na palubní sběrnici má za následek trvalou kolizi vysílání těchto dvou konfliktních modulů a celkovou špatnou funkci systému.

7.1 Použití přidělené logické adresy v modulu

Hodnota přidělené adresy (nebo více adres, protože linková vrstva podporuje obsluhu více logických adres jedním modulem) je **pevně vložena** jako konstanta do zdrojového textu linkové vrstvy, jehož překlad je součástí programu příslušného modulu. Hodnota adresy modulu je tedy pevně dána překladem a není určena například aplikační vrstvou v rámci inicializace komunikace (například jako parametr inicializační funkce rozhraní). Je tomu tak právě proto, aby hodnota adresy nemusela být v linkové vrstvě uložena v proměnné. Tím se zmenšuje riziko, že poškozením obsahu této proměnné dojde ke změně adresy a tím k logické kolizi na palubním sběrnici.

Rozsah logických adres, které modul obsluhuje, se ve zdrojovém textu linkové vrstvy definuje jako hodnota **dvou konstant**, a to THIS_ADDR a THIS_ADDR_LAST. Linková vrstva se chová tak, že přijímá všechny datové zprávy, které mají jako adresu adresáta vyplněnou hodnotu ležící v intervalu <THIS_ADDR; THIS_ADDR_LAST>.

Při odesílání datových zpráv je adresa odesílatele vyplněna podle hodnoty, kterou ve vyplněném bufferu zprávy určí aplikační vrstva. Tato vlastnost je důležitá pro moduly s více logickými adresami (komunikace), protože jejich aplikační vrstva potřebuje z hlediska ostatních modulů vystupovat pod různými adresami.

7.2 Moduly s jednou komunikační adresou

Modul vystupuje na palubní síti jako **jedna adresa**, konstanty THIS_ADDR a THIS_ADDR_LAST mají stejnou hodnotu. Pro tuto adresu modul přijímá datové zprávy, jeho aplikační vrstva je zpracovává a s touto adresou odesílatele na ně posílá odpovědi.

7.3 Moduly podporující více komunikačních adres

Moduly, které podporují více komunikačních adres, jsou dva. Jedná se o moduly komunikační, a to **modul Komunikace** a **modul Maják**. Oba dva umožňují funkci zprostředkování komunikace ze Země. Každý z nich má přiděleny dvě adresy. Jedna přísluší samotnému modulu (aplikační vrstva Majáku a Komunikace), druhá je tzv. virtuální adresa, která zastupuje vysílání ze Země. Z pohledu ostatních palubních modulů vystupuje modul Maják a modul Komunikace jako dvě nezávislé adresy. Přenos zpráv z/do pozemní stanice je popsán v další kapitole.

7.4 Rozdělení adres mezi moduly

Logické adresy jsou mezi moduly stanoveny dohodou takto:

Název modulu	Adresa modulu	Virtuální adresa	Poznámka
Napájení	1	-	
Maják	2	3	adr. 3 je pozemní stanice přes maják
Komunikace	4	5	adr. 5 je poz. stanice přes hlavní kanál
Stabilizace	6	-	
Exp.- Fotoaparát	7	-	
Exp.- Tether	8	-	
Exp.- Plachta	9	-	

8 Komunikace ze Země

8.1 Princip komunikačního spojení

Za spojení s pozemní stanicí a zprostředkování komunikace mezi pozemním střediskem a zbytkem satelitu jsou zodpovědné dva moduly – Komunikace a Maják. Jejich úloha se liší:

Komunikace – představuje hlavní datový kanál s velkou šířkou pásma, určený pro běžné spojení a přenos větších objemů dat. Modul má implementován pouze proprietární protokol s možností šifrování, spojení a datový přenos je umožněn pouze autorizovanému řídicímu středisku. Tento modul na svém rádiovém kanálu žádné informace spontánně neodesílá.

Maják – pomocný datový kanál, jehož hlavní úlohou je periodické vysílání speciálního rámce se základními diagnostickými daty satelitu. Tento rámec je vyslán ve veřejném protokolu a umožňuje příjem a dekodování všem příslušně vybaveným stanicím. Kromě toho má modul majáku implementovaný proprietární protokol s možností šifrování, a tím umožňuje alternativní přístup na vnitřní sběrnici satelitu.

Fyzické rozhraní, použité frekvence, antény, způsob modulace se pro modul Komunikace a Maják liší. Detailní popis rádiové části není předmětem tohoto dokumentu. Kromě periodického odesílání diagnostického rámce majákem se funkce obou komunikačních modulů shodují.

8.2 Diagnostický rámec majáku

Struktura diagnostického rámce majáku, způsob jeho příjmu a dekodování je **popsán a zveřejněn pro světovou radioastronomickou komunitu**, protože se předpokládá, že vysílané majákové informace budou zachytávány radioastronomickými stanicemi po celé Zemi a hodnoty přijatých dat budou poslány (například elektronickou poštou) k vyhodnocení do pozemního řídicího střediska.

Přijem a report diagnostického rámce bude **motivován zasláním certifikátu příjemci**. Přijetí a ohlášení správného, kompletního a dekodovatelného rámce do řídicího střediska bude motivováno hodnotnějším certifikátem.

Rámec je formátován jako UI rámec protokolu AX.25. Detailní popis rámce je uveden v jiném dokumentu.

8.3 Vrstvy komunikace

Fyzickou vrstvu komunikace tvoří rádiové modemy na příslušných frekvencích, koncové stupně a antény na satelitu a v pozemním středisku. Přesný popis fyzické vrstvy komunikace, frekvencí, modulace a podobně je obsažen v jiném dokumentu.

Formátování sekvence bitů je pro diagnostický rámec majáku dáno normou AX.25 (FSK, NRZI, bit stuffing), pro proprietární komunikaci je formátování a bitové kódování popsáno v jiném dokumentu.

Není dokončeno

9 Obecné datové zprávy podporované všemi moduly

Tyto zprávy podporují všechny palubní moduly, kromě virtuálních adres. Pokud obdrží modul některou z těchto zpráv, podle možnosti odpoví. Formát zprávy, kódy a struktura jsou shodné pro všechny moduly, datový obsah se může samozřejmě lišit.

Modul nemusí nutně podporovat všechny zprávy, pokud bude v jeho procesorovém jádru problém s místem pro kód. To je zajištěno podmíněným překladem, kdy je možné implementaci jednotlivých komunikačních vlastností vypnout.

9.1 Datová zpráva „Status modulu“

Zpráva umožňuje zjistit obecný status modulu. Obsah a interpretace dat statutu se liší podle konkrétního modulu. Zprávu je možné získat na vyžádání, jako kteroukoliv jinou. Kromě toho odpověď na tuto zprávu odesílají všechny moduly jako broadcast periodicky na palubní sběrnici (pro komunikační moduly).

Žádost: (délka je pevná)

Byte 0 ... délka zprávy = $3+1 = 4$

Byte 1 ... adresa, pro kterou je zpráva určena (adresát)

Byte 2 ... adresa, která zprávu odeslala (odesílatel)

Byte 3 ... kód zprávy = 1

Odpověď: (délka je pevná)

Byte 0 ... délka zprávy = $3+1+32 = 35$

Byte 1 ... adresa, pro kterou je zpráva určena (adresát = odesílatel ze žádosti)

Byte 2 ... adresa, která zprávu odeslala (odesílatel = tato adresa)

Byte 3 ... kód zprávy = $1+128 = 129$

Byte 4 až 34 ... data statutu zprávy. Interpretace je odlišná pro jednotlivé moduly. Každý modul definuje svou vlastní strukturu stavových dat.

9.2 Datová zpráva „Odešli datovou zprávu“

Zpráva umožňuje vynutit na konkrétním modulu, aby odeslal datovou zprávu daného tvaru. Žádost má proměnnou délku, podle délky zprávy, která je v ní obsažena. Výsledkem je „retranslace“ zprávy do palubní sběrnice.

Žádost: (délka je proměnná)

Byte 0 ... délka zprávy = $3+1+<délka zprávy> = 4+<délka zprávy>$

Byte 1 ... adresa, pro kterou je zpráva určena (adresát)

Byte 2 ... adresa, která zprávu odeslala (odesílatel)

Byte 3 ... kód zprávy = 2

Byte 4 ... délka vnořené zprávy

Byte 5 ... adresát vnořené zprávy, komu má být poslána

Byte 6 ... odesílatel vnořené zprávy, co má být vyplněno jako odesílatel

Byte 7 ... kód vnořené zprávy

Byte 8 až $4+délka-1$... data vnořené zprávy

Odpověď: (délka je proměnná)

Byte 0 ... délka zprávy = $3+1+<délka zprávy> = 4+<délka zprávy>$

Byte 1 ... adresa, pro kterou je zpráva určena (adresát = odesílatel ze žádosti)
 Byte 2 ... adresa, která zprávu odeslala (odesílatel = tato adresa)
 Byte 3 ... kód zprávy = $2+128 = 130$
 Byte 4 až 4+délka-1 ... data vnořené zprávy

9.3 Datová zpráva „Záznamy o význačných událostech - skupina“

Každý modul vede záznamy o svých význačných událostech. Záznamy slouží k diagnostice a sledování činnosti modulu z pozemního střediska. Události se v paměti modulu ukládají do datové struktury typu kruhová fronta. Po naplnění fronty se nejstarší záznamy přepisují postupně novými. Struktura události sestává z identifikátoru skupiny, z časové značky a parametrů. Modul eviduje hodnotu posledního čísla skupiny (1 BYTE), a toto číslo ukládá jako identifikátor skupiny do nově uložených událostí. Po uložení 8 událostí zvyšuje toto číslo o 1.

Zpráva umožňuje načíst celou skupinu událostí najednou, ale neodkazuje se na absolutní číslo skupiny, ale na relativní číslo skupiny, s tím, že 0=nejmladší, nejnovější skupina událostí, 1=druhá nejmladší skupina ... atd. až 255=dvěšestpadesátá pátá nejstarší skupina událostí. Pokud v seznamu neexistuje žádná skupina s požadovaným číslem, pošle se odpověď s nulovým počtem událostí.

Datová struktura jedné události v komunikační zprávě je tato:

- Offset 0 = identifikátor skupiny (absolutní podle interní evidence)
- Offset 1 = BYTE 1 časové značky
- Offset 2 = BYTE 2 časové značky
- Offset 3 = BYTE 3 časové značky
- Offset 4 = kód typu události
- Offset 5 = BYTE 1 parametrů události (0, pokud není použit)
- Offset 6 = BYTE 2 parametrů události (0, pokud není použit)
- Offset 7 = BYTE 3 parametrů události (0, pokud není použit)
- Offset 8 = BYTE 4 parametrů události (0, pokud není použit)
- Offset 9 = BYTE 5 parametrů události (0, pokud není použit)

Parametry události jsou specifické pro konkrétní událost. Seznam typů událostí pro jednotlivé palubní moduly, jejich parametry a jejich význam není součástí tohoto dokumentu.

Délka datové struktury jednoho záznamu události je tedy 10 bajtů.

Žádost: (délka je pevná)

Byte 0 ... délka zprávy = $3+1+1 = 5$ bajtů
 Byte 1 ... adresa, pro kterou je zpráva určena (adresát)
 Byte 2 ... adresa, která zprávu odeslala (odesílatel)
 Byte 3 ... kód zprávy = 3
 Byte 4 ... identifikátor požadované skupiny událostí (0 = nejmladší skupina)

Odpověď: (délka je proměnná)

Byte 0 ... délka zprávy = $3+1+1+10*N = 5+10*N$, N=počet událostí skupiny, min. 0 max. 8
 Byte 1 ... adresa, pro kterou je zpráva určena (adresát = odesílatel ze žádosti)
 Byte 2 ... adresa, která zprávu odeslala (odesílatel = tato adresa)
 Byte 3 ... kód zprávy = $3+128 = 131$
 Byte 4 ... identifikátor požadované skupiny událostí (0 = nejmladší skupina), kopie ze žádosti
 Byte 5 až délka-1 ... data události požadované skupiny

Počet událostí obsažených v požadované skupině může být 0 až 8, tj. délka výsledné odpovědi může být pouze 5, 15, 25, 35, 45, 55, 65, 75 nebo 85.

9.4 Datová zpráva „Záznamy o význačných událostech - jedna“

Touto zprávou je možné načíst pouze jednu událost ze skupiny událostí. Zpráva je k dispozici jako havarijní náhrada, pokud bude přenos dlouhých zpráv nespolehlivý. V takovém případě má pozemní stanice možnost pokusit se načíst data událostí postupně po jedné.

V žádosti se uvádí identifikátor skupiny a pořadí události v této skupině. Událost může mít pořadí 0-7, čím větší číslo, tím starší událost. Např. událost č.0 ve skupině č.0 je nejnovější událost v systému.

V odpovědi se posílá kopie požadované identifikace události (skupina, pořadí) a také skutečné číslo skupiny a skutečné pořadí události ve skupině. Může se totiž (na rozdíl od skupinové žádosti, zpráva č. 3) stát, že mezi dvěma žádostmi dojde k uložení události, čímž se obecně mohou změnit skupiny, nebo, pokud jde o nejnovější nebo nejstarší skupinu, pořadí událostí. Tyto situace musí pozemní stanice ošetřovat.

Žádost: (délka je pevná)

Byte 0 ... délka zprávy = $3+1+1+1 = 6$ bajtů

Byte 1 ... adresa, pro kterou je zpráva určena (adresát)

Byte 2 ... adresa, která zprávu odeslala (odesílatel)

Byte 3 ... kód zprávy = 4

Byte 4 ... identifikátor požadované skupiny událostí (0 = nejmladší skupina)

Byte 5 ... číslo pořadí události ve skupině (0 = nejmladší událost ze skupiny)

Odpověď: (délka je proměnná)

Byte 0 ... délka zprávy = $3+1+2+1+10=17$ (událost existuje) nebo $3+1+2+0=6$ (událost není)

Byte 1 ... adresa, pro kterou je zpráva určena (adresát = odesílatel ze žádosti)

Byte 2 ... adresa, která zprávu odeslala (odesílatel = tato adresa)

Byte 3 ... kód zprávy = $4+128 = 132$

Byte 4 ... identifikátor požadované skupiny událostí (0 = nejmladší skupina), kopie ze žádosti

Byte 5 ... číslo pořadí události ve skupině (0 = nejmladší událost ze skupiny), kopie ze žádosti

Pokud požadovaná událost není, zpráva odpovědi má pouze 6 bajtů. Pokud byla událost nalezena, má odpověď délku 17 bajtů a následují tyto bajty:

Byte 6 ... identifikační číslo skupiny (absolutní)

Byte 7 až 16 (celkem 10 bajtů) ... data záznamu události

Stanice, která data požaduje (typicky pozemní stanice) a která načítá události po skupinách, musí kontrolovat a ošetřovat situaci, kdy začne stahovat události jedné skupiny a náhle se změní absolutní ID skupiny v přijaté události (posun logických skupin).

9.5 Datová zpráva „Záznamy o průběhu veličiny“

Pro účely vyhodnocení a sledování, popřípadě diagnostiky, je důležité, aby jednotlivé moduly vedly pravidelné záznamy o průběhu zásadních veličin. Modul řízení napájení například vede záznamy o průběhu palubního napětí, proudovém odběru, modul stabilizace vede záznamy o měřeném vektoru externího magnetického pole a podobně.

Způsob ukládání těchto dat je interní věc konkrétního modulu, ale modul musí být schopen transformovat data do podoby vyžadované touto zprávou.

Každá ukládaná veličina má v rámci modulu jedinečné identifikační číslo. Seznam ukládaných veličin v jednotlivých modulech a přiřazení jejich identifikačních čísel není součástí tohoto dokumentu.

Žádost: (délka je pevná)

Odpověď: (délka je proměnná)

9.6 Datová zpráva „Statistiky komunikace - skupina“

Každý modul by měl evidovat statistické údaje o probíhající komunikaci, například celkový počet bajtů přijatých ze sběrnice, celkový počet bajtů odeslaných do sběrnice, celkový počet zpráv přijatých, celkový počet zpráv odeslaných, počet zpráv přijatých za poslední minutu a podobně. Přesný seznam těchto statistik není obsahem tohoto dokumentu. Hodnoty statistik se posílají v této zprávě.

Žádost: (délka je pevná)

Odpověď: (délka je proměnná)

10 Datové zprávy specifické pro modul Napájecího zdroje

10.1 Datová zpráva „Stav a výhled palubního napájení“

Energie, která je na palubě satelitu k dispozici, je obecně nedostatek a moduly musí počítat se situací, kdy

10.2 Datová zpráva „Oznámení o plánu vypnutí“

Vzhledem k tomu, že se z důvodu hospodaření s energií je nutné moduly čas od času odpojit od napájení, o čemž rozhoduje modul řízení napájení, je nutné plánované vypnutí modulu oznámit předem, aby měl dotyčný modul případně šanci dokončit některé činnosti, než bude vypojen.

Proto každý modul eviduje interní proměnnou nazvanou TTD (Time To Die), která udává čas v sekundách do jeho vypnutí. Tuto hodnotu mu modul Napájení neustále aktualizuje rozesláním broadcastu této zprávy.

Žádost: (délka je pevná)

Odpověď: (délka je proměnná)

11 Datové zprávy specifické pro modul Stabilizace

Datová zpráva „Časová synchronizace“

Datová zpráva „Příkaz ke změně času“

Datová zpráva „Stav měření orientačních vektorů“

Datová zpráva „Stav algoritmu stabilizace“

Datová zpráva „Příkaz ke změně stavu algoritmu“

Datová zpráva „Příkaz ke změně elementů dráhy“

12 Datové zprávy specifické pro modul Fotoaparát

Datová zpráva „Stav adresáře souborů“

Datová zpráva „Položka adresáře souborů“

Datová zpráva „Rozvrh pořízení fotografií a uložení souborů“

Datová zpráva „Vymazání všech souborů“

Datová zpráva „Vymazání souboru“

Datová zpráva „Vytvořit prázdný soubor“

Datová zpráva „Zapsat data do souboru“

13 Datové zprávy specifické pro modul Tether

Datová zpráva „Stav tetheru“

14 Datové zprávy specifické pro modul Maják

Datová zpráva „Nastavení parametrů diagnostického rámce“

Datová zpráva „Zjištění stavu rádiového kanálu“

Datová zpráva „Nastavení parametrů rádiového kanálu“

15 Datové zprávy specifické pro modul Komunikace

Datová zpráva „Zjištění stavu rádiového kanálu“

Datová zpráva „Nastavení parametrů rádiového kanálu“