

## Popis komunikačního protokolu, pro komunikaci procesorů v mp3 přehrávači.

Řídící procesor má na svých portech připojenou klávesnici pro řízení přehrávače a display pro zobrazování údajů (o přehrávaném souboru, listování složkami..)

Obslužný procesor je připojen k mp3 dekodéru a přes IDE sběrnici k pevnému disku.

Aby byla umožněna spolupráce obou zařízení, komunikují spolu oba procesory USARTem.

Řídící procesor musí vždy po odeslání příkazu počkat na odpověď! Některé příkazy musejí po sobě následovat v přesném pořadí. Příkaz nebude proveden, dokud nebudou odeslány všechny parametry. Pokud obslužný procesor na příkaz dlouho nereaguje, lze příkaz stornovat posláním min. osmi nulovými byty. Pokud ani to nepomůže, musí se procesor restartovat.

Toto je zatím jen návrh, ne popis, tak to skutečně pracuje...

Např. pokud má příkaz jeden 4 bytový parametr, jsou po USARTu odeslána data:

číslo příkazu, 4. byt parametru (nejnižší), 3., 2. byt, 1 byt (nejvyšší byt)

### 01h – vrať vlastnosti disku (bez parametrů)

*Musí být 1. příkaz ve vzájemné komunikaci. Pokud je v odpovědi nastaven 7. bit, tak nelze znova tento příkaz volat.*

#### Parametry:

žádné

#### Odpověď:

- 1. byte obsahuje vlastnosti disku připojeného k řadiči. Byty mají následující význam:

```
0. bit = 1 -> disk je přítomen, zapnutý a funkční :-)  
1. bit = 1 -> podpora LBA (pokud není nastaven bit2, tak pouze LBA28  
-> disk do 120GB)  
2. bit = 1 -> podpora LBA 48 (disk je zřejmě větší jak 120GB)  
3. bit = 1 -> velikost sektoru je větší než 256 !!!  
-> (s takovým diskem program neumí)  
4. bit -> 0  
5. bit -> 0  
6. bit -> 0  
7. bit = 1 -> disk je pro nás použitelný (LBA, sektor o 512B)
```

*Po řídící procesor je významný zvláště 7. bit. (Pokud je nastaven, může mp3 přehrávač pokračovat v práci.) Pokud není žádný disk k procesoru připojen, tak procesor na žádný příkaz nereaguje! Proto by měl řídící procesor vypsat něco jako „SEARCH ATA...“ a poté čekat na odpověď příkazu 01h.*

- 2. – 41. byte - model disku (40 ASCII znaků obsahující jméno disku)

např.:

„SAMSUNG SV0511D

“

## 02h – vrat' oddíly se systémem FAT32

*Každý disk musí být rozdělen alespoň na jeden oddíl. Program v obslužném procesoru umí pracovat pouze s oddíly typu 0Bh nebo 0Ch (FAT32 s LBA adresací). Program dokáže najít až čtyři oddíly se systémem souborů FAT32, takový oddíl však musí být vždy na prvním záznamu MBR, či BR rozšířeného oddílu. (Jeden primární oddíl s FAT32 a poté vždy jeden logický oddíl v rozšířeném oddílu...)*

### Parametry:

- **číslo požadovaného oddílu (0-3)**

*Vrátí adresu a jmenovku jednoho z nalezených oddílů se systémem souborů FAT32. Hodnota parametru může být v rozsahu 0-3, víc oddílů program nevyhledává.*

### Odpověď:

- **1. byte – signalizace oddílu**

*Pokud tento byte se rovná 00h, tak požadovaný záznam neobsahuje adresu oddílu. Pokud se rovná FFh, tak záznam obsahuje adresu oddílu se systémem FAT32 a jmenovku tohoto oddílu.*

- **2. - 5. byte – adresa prvního sektoru oddílu**

*2. byte je nejnižší část adresy a 5. byte nejvyšší část adresy. Jedná se o LBA adresu sektoru, na kterém se nachází spouštěcí záznam svazku se systémem souborů FAT32.*

- **6. - 16. byte – 11 ASCII znaků obsahujících jmenovku svazku**

---

## 03h – nastav oddíl

*Načte spouštěcí záznam vybraného oddílu se systémem FAT32. Parametr pro tento příkaz lze zjistit voláním příkazu 02h. Pokud jako odpověď je úspěch, nelze již znova tento příkaz (ani předchozí) volat. (Do restartu procesoru.)*

### Parametry:

- **adresa požadovaného oddílu (4byty)**

*Adresa prvního sektoru oddílu, na kterém se nachází spouštěcí záznam FAT32. (Adresa by měla odpovídat jednomu ze záznamů získaných příkazem 02h.)*

### Odpověď:

- **1 byte signalizující úspěch**

*Pokud je nastaven 4. bit vráceného bytu, znamená to, že spouštěcí záznam svazku byl korektně načten. Pokud ne, došlo k některé z chyb:*

- *adresa sektoru neobsahuje spouštěcí záznam svazku s FAT32*
- *velikost sektoru není 512B*
- *cluster FAT je větší jak 16KB (více jak 32 sektorů na cluster)*
- *svazek neobsahuje dvě kopie FAT tabulky*

## 04h – vrat' velikost clusteru

### parametry:

- bez parametru

### odpověď:

- velikost clusteru

vrátí jeden byte s velikostí FAT clusteru (kolik sektorů má cluster) Vždy se jedná o exponenty 2 (1, 2, 4, 8, 16, 32...) Program umí pracovat pouze s clusterem o velikosti do 32 sektorů (16kB).

---

## 05h – vrat' jméno a adresu souboru

*Záznamy v adresáři nemusejí být v systému FAT(32) nijak řazeny (podle abecedy)!*

### parametry:

- 1. až 4. byte je číslo clusteru na kterém se nachází část adresáře  
*Další cluster adresáře zjistíte voláním funkce 06h.*
- 5. byte = část adresáře  
(Může být v rozsahu 0..[velikostClusteru - 1] – maximálně tedy 31.)
- 6. byte = číslo záznamu  
(Může být v rozsahu 0..15) *Pokud chcete soubory vyčítat tak, jak jdou po sobě v adresáři, tak nejdříve inkrementujte číslo záznamu, poté část adresáře, a nakonec čtete další cluster...*

### odpověď:

- první byte signalizuje stav odpovědi:
    - 00h => záznam v adresáři je prázdný (následujících 15 bytů je k ničemu)
    - 01h => záznam je adresář
    - 02h => záznam je soubor
    - 06h => záznam je soubor s příponou MP3
    - 80h => ukazatel na záznam (5. nebo 6. byte parametru) je mimo rozsah (Zvětšete část adresáře a číslo záznamu zvolte 0, nebo volejte příkaz 06h a zjistíte číslo dalšího clusteru v alokačním řetězci.)
  - 2. až 9. byte --> 8 znaků dlouhé jméno  
("DOSovský" tvar - např. složka "dokumenty" = "DOKUME~1" )
  - 10..12 byte -> u souboru tři znaky přípony  
(u adresářů většinou mezery - 20h 20h 20h) (Pokud se jedná o adresář, a záznam o příponě se nerovná 0x202020 (tri mezery), tak skutečné jméno adresáře je : ADRESAR.PŘI)
  - 13..16 byte -> 1. cluster souboru
- 

## 06h – vrat' číslo dalšího clusteru v alokačním řetězci

### parametry:

- 1. až 4. byte je číslo clusteru od kterého chcete zjistit pokračování.  
Program nekontroluje, zda číslo nepřevyšuje počet clusterů ve FAT!!! Pokud byl zadán parametr mimo tento rozsah, nevrátí program platné informace!

### odpověď:

- 1. byte signalizuje stav výsledku:
  - 00h => další 4 byty odpovědi obsahují číslo následujícího clusteru v alokačním řetězci
  - FFh => předaný cluster je poslední v řetězci, nebo je prázdný. (další čtyři byty odpovědi obsahují neplatnou hodnotu)
- 2. až 5. byte obsahují číslo clusteru, který následuje v alokačním řetězci za předaným clusterem

## 07h – čti cluster

(Přečte a odešle všechna data obsažená v požadovaném clusteru.)

### parametry:

- 1. až 4. byte je číslo libovolného clusteru (na kterém se třeba nachází soubor).

### odpověď:

- *velikost\_clusteru* \* 512 bytů dat obsažených v požadovaném clusteru (pokud soubor v tomto clusteru zabírá třeba jen 100B a velikost clusteru je 8KB, odešle všech 8KB.)
- 

## 08h – zjistí velikost souboru

Tento příkaz není pro přehrávání mp3 moc důležitý, je přidám pouze pro účely ladění programu a pro použití v pokusném externím šuplíku přes RS232.

*Program nekontroluje zda požadovaný záznam obsahuje soubor, to lze zjistit voláním příkazu 05h*

### parametry:

1. až 4. byte je číslo clusteru na kterém se nachází část adresáře

### **5. byte = část adresáře**

(Může být v rozsahu 0..*velikostClusteru* - 1] – maximálně tedy 31.)

### **6. byte = číslo záznamu**

(Může být v rozsahu 0..15) *Pokud chcete soubory vyčítat tak, jak jdou po sobě v adresáři, tak nejdříve inkrementujte číslo záznamu, poté část adresáře, a nakonec čtete další cluster...*

### odpověď:

- první byte signalizuje stav odpovědi:
    - 00h => velikost záznamu byla přečtena
    - 80h => ukazatel na záznam (5. nebo 6. byte parametru) je mimo rozsah
  - 4 byty reprezentující velikost souboru v bytech.
- 

příkazy s kódem 80 a větším jsou pro práci s dekodérem

## 80h – hraj mp3

### parametry:

- 1. až 4. byte je číslo clusteru na kterém se nachází část adresáře
- 5. byte = část adresáře  
(Může být v rozsahu 0..*velikostClusteru* - 1] – maximálně tedy 31.)
- 6. byte = číslo záznamu obsahující soubor s příponou mp3  
(Může být v rozsahu 0..15)

### odpověď:

- první byte signalizuje stav odpovědi:
  - 06h => soubor se nejspíše podařilo začít přehrávat (nekontroluje zda jsou mp3 data platná)
  - not 06h => nějaká chyba...

## 81h – vrat' stav přehrávání

Příkaz je bez parametru

**odpověď:**

- Dva byty indikující stav přehrávání. Bity mají následující význam:

```
PREH_STAV0:
0. bit = 0 => stop nebo pauza (nic nehraje)
          1 => play (nejaky soubor je prehravan)
1. bit = 0 => neni zadny soubor k prehravani (nastaveni bitu play nema zadny
          ucinek)
          1 => je soubor pripraven k prehravani (muze byt ale pauza - nic nehraje)
2. bit = 1 => je nastaven, kdyz se zmeni prehravany soubor (bez volani prikazu)
          do doby, nez prijde prikaz na dotaz STAVU... (81h)
3. bit = 0 => po skonzeni souboru se pokracuje v prehravani (zavisi na nastaveni
          repeatu)
          1 => po skonzeni prehravani souboru se nic neprehlava
          (ceka se na prijeti prikazu k prehravani)
4. bit = 0 => repeat off
          1 => repeat on
5. bit = 0 => repeat adresare (prvni cluster adresare musi byt nastaven
          v PREH_ADDR_ZACATEK_CL[1-4] )
          1 => repeat souboru (po skonzeni prehravani souboru se ten samy soubor
          zacne prehravat znova)
6. bit = 0 =>
          1 => rezervovano (pro nahodny vyber, nebo jiny rezim prehravani...)
7. bit = 0 =>
          1 => rezervovano
PREH_STAV1:
0. bit = 0 => normalni prehravani
          1 => previjeni mp3 dopredu
1. bit = 0 =>
          1 => rezervovano (pro previjeni dozadu)
2. bit = 0 => zpusob previjeni,
          0 => previji se rychle, potichu
          1 => previji se pomalu, vzdy se po nejake dobe kratky usek souboru
          prehraje
3. bit = 0 => normalni nastaveni zvuku
          1 => zvýraznění basu a vysek (moznost dekodery vs1001 "bass/treble
          enhancer")
4. bit = 0 => NO MUTE
          1 => MUTE
5,6,7 => rezervovano
```

### • 2 byty obsahující odehraný čas mp3 v sekundách

Tato informace je získávána z registru dekodéru. Čas je platný i v případě rychlého převíjení. Pouze v případě neplatných dat, či skoků v souboru neobsahuje tento záznam pravdivou informaci.

---

## 82h – nastav hlasitost

**parametry**

- 1.byte: hlasitost pro levý kanál
- 2. byte: hlasitost pro pravý kanál

*Hodnota 0 je maximální hlasitost, 255 minimální hlasitost.*

*Výchozí hodnota po startu přehrávače je 0x4040.*

**Odpověď:**

- vždy FFh

## 83h – vrat' informace o přehrávaném souboru

*Program netestuje zda se vůbec nějaký soubor přehrává. To lze zjistit testováním 1. bitu 1. bytu odpovědi na*

příkaz 81h.

### **bez parametru**

#### **odpověď:**

- 2 byty získané z registru AUDATA dekodéru vs1001. Bližší informace o jejich obsahu najdete na straně 26 datasheetu k vs1001. (Obsahuje informace o datovém toku, vzorkovací frekvenci a počtu kanálů.)
- 2 byty získané z registru HDAT0
- 2 byty získané z registru HDAT1

Registry HDAT0, HDAT1 obsahují informace získané z hlavičky mp3 souboru. Jejich bližší popis najdete na straně 27 datasheetu k vs1001. Nejdůležitější informaci obsahují asi bity 15 až 5 registru HDAT1. Pokud tyto byty obsahují konstantu 2047, jsou přehrávaná data platná. (Mp3 hraje.)

---

## **84h – nastav stav přehrávání**

### **parametry:**

- 2 byty nastavující vlastnosti přehrávání

```
PREH_STAV0:
  0. bit = 0 => pauza
           1 => play (pokud není připraven soubor k přehrávání, nemá nastavení
                tohoto bitu žádný účinek)
  1. a 2. bit nelze příkazem nastavit (mohou mít libovolnou hodnotu)
  3. bit = 0 => po skončení souboru se pokračuje v přehrávání (závisí na nastavení
                repeatu)
           1 => po skončení přehrávání souboru se nic nepřehrává
                (čeká se na přijetí příkazu k přehrávání)
  4. bit = 0 => repeat off
           1 => repeat on
  5. bit = 0 => repeat adresare (první cluster adresare musí být nastaven
                v PREH_ADDR_ZACATEK_CL[1-4] )
           1 => repeat souboru (po skončení přehrávání souboru se ten samý soubor
                začne přehrávat znovu)
  6. bit - rezervováno (pro nahodný vyber, nebo jiný režim přehrávání...)
  7. bit - rezervováno
PREH_STAV1:
  0. bit = 0 => normalní přehrávání
           1 => prevíjení mp3 dopředu
  1. bit = 0 =>
           1 => rezervováno (pro prevíjení dozadu)
  2. bit = zpusob prevíjení,
           0 => prevíjí se rychle, potichu
           1 => prevíjí se pomalu, vždy se po nějaké době kratky usek souboru
                přehraje
  3. bit = 0 => normalní nastavení zvuku
           1 => zvýraznění basu a vysek (možnost dekodéru vs1001 "bass/treble
                enhancer")
  4. bit = 0 => NO MUTE
           1 => MUTE
  5,6,7 => rezervováno
```

I když výrobce dekodéru píše, že by měl být čas MP3 správně vypočítáván i během prevíjení, u některých souborů tomu tak není. (Při začátku prevíjení se registr s odehraným časem vynuluje.) Zatím se mi nepodařilo zjistit, zda tato chyba je u dekodéru vs1001k běžná, či se jedná o chybu mého firmwaru. Při pomalém prevíjení je poměr přehraných sektorů ku přeskočeným 5:20. Při pomalém prevíjení může docházet k velmi hlasitým defektům výstupního signálu! Doporučuji nepřevíjet při maximální hlasitosti.

#### **odpověď:**

- vždy FFh