

*Příručka uživatele*

**u@speccyFTP**

**(L1: M-8255-TP)**

*Utilita pro **MDOS File Commander***

*(c) 2003, 2007, 2020 MTs*

## Základní informace

**Utilita je určena pouze pro MFC verze 2 a vyšší.** Jedničkové verze MFC byly testovací a věci okolo utilit ještě nebyly pořádně vyřešeny.

U utility **je přiložen její zdrojový text** ve formátu do Promethea (maska *p \*.b*). Máte tak možnost podívat se, jak to vlastně funguje a větší šanci k tvorbě své vlastní a funkční utility. Pokud hodláte v tomto zdrojáku něco měnit (opravit, vylepšit atd.), pak pamatujte, že je slušnost **informovat o tom původního autora** (zvláště, když to pak hodláte šířit dál).

Kdybyste náhodou zapomněli, tak utilita se do MFC nahrává stiskem **SS+U** a spouští se klávesou **3** nebo **U**.

**Utilita je uložena v paměti počítače pouze dočasně.** MFC utilitu za určitých okolností (kopírování souborů) maže a je tedy třeba ji pak nahrát znovu.

**Je možné pracovat pouze s JEDNOU konkrétní disketou a mechanikou.** Nejde číst soubory např. z mechaniky A a zapisovat je na B. Nelze ani použít 2 různé diskety na stejné mechanice (tj., že z jedné diskety by se četlo a na druhou se zapisovalo). **Pokud disketu během práce v utility módu vyměníte za jinou, následky mohou být katastrofické (v nejhorším případě až ztráta veškerých dat na disketě)!**

**Zápis dat** (podprogram U\_WBYTE) **funguje pouze na zx disketě.** Toto omezení lze však obejít tak, že se soubor zkopíruje z pc na zx disketu a budete pracovat tam.

## u@speccyFTP

(c) 2003, 2007, 2020 MTs

Tato MFC utilita slouží pro přenos souborů mezi dvěma počítači dle protokolu *speccyFTP*. Tento protokol jsem zavedl já a jeho popis naleznete ke konci této příručky.

Dále utilita obsahuje na nejnižší vrstvě 512 bytes dlouhý ovladač *M-8255-TP (MTs-8255-Transfer Protocol)*, který umožňuje přes obvod 8255 (po kabelu) opravdu RYCHLÝ přenos souborů. Ovladač ale lze úplně jednoduše vyměnit za jiný, např. sériový RS-232 (*speccyFTP* nenařizuje přenos jen jedním konkrétním hardwarem...).

Myšlenka přenosu souborů po kabelu mě chytla v roce 2003 na vysoké škole při hrabání se v síťových protokolech PC a učení se obecného modelu ISO/OSI. Nejprve jsem chtěl přenést na ZXS pravý FTP z PC, ale toto jsem byl donucen nakonec zavrhnout jako zbytečně složité řešení vyžadující spoustu paměti. Vytvořil jsem tedy naprosto novou věc a pojmenoval sFTP (*speccy File Transfer Protocol*). Ta zkratka se mi ovšem později ukázala jako hodně nešťastná (použili ji na PC), takže od roku 2007 raději používám slovo *speccyFTP*, což je ve skutečnosti i významem té původní zkratky.

V letech 2003-2007 existovala tato utilita ve verzi (resp. s pořadovým číslem) 10 a byla napsána natvrdo s nutností použít IO 8255. Na konci roku 2006 jsem projekt znovu otevřel a udělal modulárním (dovolil jednoduchou výměnu za jinou než 8255 vrstvu – tzv. L1 ovladač). Následně kontaktoval tehdejší vývojáře (Baze, Tritol, Shrek, aj.), poslal jim novou dokumentaci a snažil se vyjednat podporu i v jejich systémech. Nic se však neudělalo, protože ti lidé už víceméně se ZXS končili a nic nového se od nich neobjevilo. Tak (z)byl pouze můj software a nesl pořadové číslo 24. Na webu však tato utilita nikdy zveřejněna nebyla a stejně tak ani nová dokumentace s modulární aktualizací.

Aktuální verze je z roku 2020 a nese pořadové číslo 28. Za znovuootevření projektu se zasloužil *Chip128*, který zkoušel soubory přenášet, ale ač MFC hrdě hlásil úspěšný přenos, tak data byla chybná. Nakonec se zjistilo, že chyba není v software ale v hardware - byla „nakopnutá“ (vadná) jedna 8255ka a XOR ochrana toto nepoznala. Obohatil jsem tedy *speccyFTP* ještě o sadu kontrolních bytes v InfoPacketu, které případnou hardwarovou chybu odhalí ještě před vlastním přenosem souboru.

Do budoucna se chystám napsat L1 pro RS-232 a pokusit se podporu *speccyFTP* přidat i do utility .sercp od Pavla Vymetálka.

## Specifikace propojovacího kabelu (L0: 8255)

Spojit dvě 8255 jde mnoha způsoby. Já jsem se snažil vymyslet takové, které by fungovalo i na Didaktiku GAMA (tam jak známo je C0 použito pro stránkovací účely, takže brána C3...0 nepřipadá v úvahu) a také, aby to pak bylo i přehledné v samotném programovém kódu. Potřebnými dráty jsem nešetřil.

|     |         |     |                        |
|-----|---------|-----|------------------------|
| A0  | o-----o | A0  |                        |
| A1  | o-----o | A1  |                        |
| A2  | o-----o | A2  |                        |
| A3  | o-----o | A3  |                        |
| A4  | o-----o | A4  |                        |
| A5  | o-----o | A5  |                        |
| A6  | o-----o | A6  |                        |
| A7  | o-----o | A7  | A0...7 – data (1 byte) |
| B4  | o-----o | B4  | busy/ready bit         |
| B5  | o-----o | B5  | XOR error bit          |
| B7  | o-----o | B7  | BREAK bit              |
| C4  | o-----o | C4  | busy/ready bit         |
| C5  | o-----o | C5  | XOR error bit          |
| C7  | o-----o | C7  | BREAK bit              |
| GND | o-----o | GND |                        |

**Rozmístění signálů na konektoru interface:**

| spoje      | č. vývodu | součástky  |
|------------|-----------|------------|
| +5V        | 1         | PC0        |
| IN1        | 2         | OUT1       |
| IN0        | 3         | OUT0       |
| výřez      | 4         | výřez      |
| PC6        | 5         | GND        |
| <b>PC5</b> | 6         | <b>PC7</b> |
| PC1        | 7         | <b>PC4</b> |
| PC3        | 8         | PC2        |
| PB6        | 9         | <b>PB7</b> |
| <b>PB4</b> | 10        | <b>PB5</b> |
| PB2        | 11        | PB3        |
| PB0        | 12        | PB1        |
| <b>PA6</b> | 13        | <b>PA7</b> |
| <b>PA4</b> | 14        | <b>PA5</b> |
| <b>PA2</b> | 15        | <b>PA3</b> |
| <b>PA0</b> | 16        | <b>PA1</b> |

*Kompakt, D40/80, M/P*

| spoje      | č. vývodu | součástky  |
|------------|-----------|------------|
| IN2        | 1         | OUT2       |
| IN1        | 2         | OUT1       |
| IN0        | 3         | OUT0       |
| výřez      | 4         | výřez      |
| PC6        | 5         | GND        |
| <b>PC5</b> | 6         | <b>PC7</b> |
| PC1        | 7         | <b>PC4</b> |
| PC3        | 8         | PC2        |
| PB6        | 9         | <b>PB7</b> |
| <b>PB4</b> | 10        | <b>PB5</b> |
| PB2        | 11        | PB3        |
| PB0        | 12        | PB1        |
| <b>PA6</b> | 13        | <b>PA7</b> |
| <b>PA4</b> | 14        | <b>PA5</b> |
| <b>PA2</b> | 15        | <b>PA3</b> |
| <b>PA0</b> | 16        | <b>PA1</b> |

*Gama '89*

Máte-li tedy dvě 255ky propojeny, pak Vám již nic nebrání úspěšnému bleskurychlému přenosu souborů.

Snažte se ale zabránit situaci, aby se po propojení dvou 8255 (zapnutím počítače) začaly obvody "přetlačovat", tj. aby byly stejné porty na obou 8255 nastaveny jako výstupní. Mohlo by dojít až ke zničení obvodů. Majitelé D40/80 mohou být celkem v klidu, protože obě verze MDOSů nechávají interface zablokovaný, a na Kompaktu se vše nastavuje jak má (všechno vstupní). Pouze Didaktik Gama má bránu C(0-3) defaultně nastavenou směrem ven. Ale tyto dráty z brány C nepoužíváme, takže opět nemůže k ničemu dojít.

**Funkční klávesy:**

- CS+1** návrat do MFC  
**I** příjem souborů z druhého počítače  
**O** odeslání označených souborů na druhý počítač  
**B** odblokování interface (OUT 153,16 nutný u D40/80)

- L** povolení vypínání mechaniky při LOADu
- P** změna délky paketu (defaultně 256 bytes)
- R** nebo **T** změna počtu opakování po chybě (defaultně 5)

Po spuštění utility musíte zvolit, jestli budete data vysílat ven (file OUT) nebo je přijímat (file IN). Jakmile zvolíte příslušnou volbu (nebo klasika CS+1 pro návrat do MFC), dojde k nastavení obvodu 8255 a zobrazí se hláška "Ready to work. Press any key.". Pouze tehdy, když je toto hlášení zobrazeno **na obou** počítačích, můžete stisknout libovolnou klávesu (kromě CS+1 samozřejmě) pro zahájení přenosu. Přenos se dá kdykoliv přerušit stiskem SPACE (příp. BREAKu). MFC přenesou všechny označené soubory, rozhodně tedy nemusíte přenášet jen po jednom. Celé to pracuje obdobně jako např. ve staříčké M602 na PC...

**Pouze u 8255 v řadiči D40/80 ze Skalice je třeba ještě před přenosem odblokovat interface v disketové jednotce** (pokud jste tak už neučili v Basicu před spuštěním MFC...), proto je zde ona volba pod klávesou **B**.

*Poznámka k 8255 u D40/80 (resp. k MDOSům):*

*D40/80 (s MDOS 2) má pro blokování své 8255 dva porty: 151 a 153. OUT 151,32 povoluje port 153 a teprve s takto povoleným portem 153 můžete použít OUT 153,16. Teprve toto zajistí odblokování 8255 a možnost jejího použití! Toto šílené řešení je k zajištění nekolidování, když má počítač již nějakou svoji 8255 (GAMA například). Pokud MDOS 2 při resetu detekuje jinou 8255, pak OUT 151,32 neudělá. Když jinou 8255 nedetekuje, pak OUT 151,32 udělá, ale svou 8255 i tak nechá ještě pro jistotu zablokovanou a uživatel si ji musí povolit sám tím OUT 153,16.*

*D40/80 (s MDOS 1) to má díky chybě v sobě ještě komplikovanější. Tam jsou porty 145 a 153, ovšem ta 145ka nikdy port 153 nepovolí. Uživatel tedy musí pro odblokování 8255 provést OUTy dva: OUT 145,32 a OUT 153,16*

*Didaktik Kompakt má vlastní 8255ku (podobně jako Didaktik GAMA) a výše uvedené OUTování na ni nemá žádný vliv! Po resetu je vždy 8255 povolena a okamžitě připravena k použití.*

*S hardwarovým projektem d80mini a rozjžděním MDOSu 2 tam se zjistilo, že port 153 koliduje se ZXS 128K (porty #xxFD kde musí stovacetosmy mít bit 1 v jedničce). Proto tam (v ROMce) mají OUT 153 nahrazen za OUT 155. D80mini však nemá svoji 8255ku, takže outováním se ničeho nedosáhne (max. tak*

zhroucení systému na ZXS 128) a ony instrukce se měly z ROM lépe úplně odstranit.

Pomocí klávesy **L** můžete povolit, aby při načítání souborů z diskety mechanika zhasínala. Moc Vám to však nedoporučuji, protože paralelní přenos je opravdu tak rychlý, že vypínání disketovky ho neúměrně zbrzdí. Při jiném L1 ovladači, který má výrazně delší dobu přenosu, to však může být užitečné.

Klávesou **P** (jako paket) si můžete změnit délku přenosových paketů. Doporučuji nesahat a nechat výchozí hodnotu 256. Změna velikosti slouží spíše vývojářům k testovacím účelům. Maximální velikost si program hlídá (a když tak automaticky sníží), nedovolí Vám nastavit více, než co k tomuto účelu má v utilitě volného místa.

Klávesou **R** (repeat) se dá zvolit počet opakování při neúspěšném odeslání paketu. Výchozí hodnota je 5, tedy pět pokusů paket odeslat než se vyvolá chyba a přenos se předčasně ukončí. Můžete schválně vyzkoušet nastavit na ideální hodnotu 1. Pokud je HW plně funkční a kabel odolný proti rušení, mělo by to fungovat a můžete si tak změřit maximální možnou rychlost přenosu. Ona ale bude stejně nuceně brzděna zapisovacím procesem na cílové médium...

Pokud nastane při ukládání dat nějaká chyba, utilita ji ohlásí a vrátí se do MFC. Může se Vám tak stát, že naposledy ukládaný soubor nebude zcela v pořádku (nebude celý). Takovýto **chybný soubor poznáte podle toho, že nebude mít nastaven žádný atribut** (HSAPRWED) a také by měl být vždy na poslední pozici ve výpisu. Tento soubor ručně odstraňte (vymažte).

**Chybová (a jiná) hlášení:**

| Zpráva (hláška)               | Význam                                                                      |
|-------------------------------|-----------------------------------------------------------------------------|
| 8255 unblocked (OUT 153,16) . | Informace, že interface v D40/80 byl odblokován.                            |
| Disk I/O error!               | Chyba při práci s disketou.                                                 |
| Hardware link error!          | Detekována chyba na hardwarovém propojení počítačů (Level 0 není v pořádku) |
| OK. Stop drive when reading.  | Informace, že mechanika se bude při LOADu vypínat.                          |
| Ready to work. Press any key. | Očekává se stisk libovolné klávesy (kromě CS+1). Poté začne samotný přenos. |
| Receiving file: <i>name</i>   | Právě přijímaný soubor.                                                     |
| Sending file: <i>name</i>     | Právě vysílaný soubor.                                                      |
| Transfer error!               | Chyba při přenosu (break nebo error).                                       |
| Waiting (InfoPacket)...       | Přijímací počítač čeká na úvodní paket z vysílacího počítače.               |

Při jakémkoliv návratu do MFC, jsou porty 8255 nastaveny jako vstupní mimo bránu C3-0. Ta je kvůli Didaktiku Gama neustále držena jako výstupní! Totéž platí při spuštění této utility nebo po OUTu 153,16.



## **L1: M-8255-TP (MTs-8255-Transfer Protocol)**

Následuje výpis zdrojového kódu rutiny, která zajišťuje spolehlivý přenos dat přes obvod 8255 na základě výše popsaného propojovacího kabelu (L0).

Spolehlivost přenosu je řešena přidáním jednoho byte, tzv. XORu, vysílající stranou na konec každého paketu. Příjímací strana jej porovná se svým, který si tvoří během příjmu bajtů. Liší-li se, došlo při přenosu k chybě. Bohužel trvalým poškozením nějakého bitu brány A (např. odpojený jeden drát), přičemž na tom „nakopnutém“ bitu bude stále stejná hodnota, dojde k situaci, že přijímač chybu na základě XORu neodhalí a bude považovat data za správná. Je to z principu fungování instrukce XOR a přenosu celého byte najednou. Takovouto chybu však zjistí vyšší level (aplikace) už na samém počátku přenosu v příchozím InfoPacketu (je v něm uložena speciální sekvence kontrolních bytes, která je předem známá a tak se jen hledá, jestli tam skutečně je).

Dále jsem L1 pro 8255 napsal tak, bych se vyhnul jakékoliv potřebě něco časovat a spoléhat se na to. Kabel je definovaný tak (bitů na 8255 je dost :), že samo zařízení si řekne, zda už je připraveno byte přijmout/odeslat. Pokud tedy uděláte kabel delší než je normálně zdrávo (u paralelního víc než 2 metry), nemuselo by se to nějak zvláště negativně projevit. Ale nezkoušel jsem to. U jiných L1 ovladačů (např. pro RS-232) se už ale časování vyhnout zaručeně nepůjde...

L1 v utilitě pro MFC začíná na adrese 57600 a má délku 512 bajtů. Tento kód můžete klidně vyměnit za jiný (pokud nějaký jiný máte...):

```
CLEAR 49999
LOAD * "u@sFTP" CODE 55296
LOAD * "jiný L1 ovladac" CODE 57600 ;nesmí být delší než 512 bytes !!!
SAVE * "u@sFTP-my" CODE 55296,2816
```

```

;(c)MTs M-8255-TP ;IN: HL adresa s daty
;(MTs-8255-Transfer Protocol) ; BC delka dat
; ; E pocet pokusu

;delka ovladace max 512 bytes !

;A0-7 odesilany byte

;C7 (1) break u vysilani
;C5 (1) ted se posila xor-CRC
;C4 wait smycka pro prijimac

;B7 (1) break u prijimani
;B5 (1) prijmac hlasi chybu
;B4 wait smycka pro vysilac

PORTA equ 31
PORTB equ 63
PORTC equ 95
PORTCOMM equ 127

X1_INIT jr SET_ALIN
X2_SETOU jr SET_OUT
X3_SETIN jr SET_IN
X4_BOUT jr BYTE_OUT
X5_BIN jr BYTE_INX
X6_ENDOU jr SET_ALIN
X7_ENDIN jr SET_ALIN
X8_REOUT jr RESERVED
X9_REIN jr RESERVED

RESERVED ret

;C0-3 u D Gama musi byt vystup!
;nezapojovat tyto bity v kabl!

SET_ALIN ld a,%10011010
out (PORTCOMM),a
ret

SET_OUT ld a,%10000010
out (PORTCOMM),a
ex (sp),hl
ex (sp),hl
xor a
out (PORTC),a
ret

SET_IN ld a,%10011000
out (PORTCOMM),a
ex (sp),hl
ex (sp),hl
xor a
out (PORTB),a
ret

BYTE_INX jr BYTE_IN

;OUT: c break/CRC errorry
; nc odeslanoOK a prijatoOK
; HL,BC beze zmeny

BYTE_OUT di
inc bc
OUT_AG ld d,0
push bc
push hl
GO_OUT xor a
out (PORTC),a
WAIT_O1 ld a,127
in a,(254)
rra
jr nc,BREA_O
in a,(PORTB)
bit 7,a
jr nz,BREA_O
bit 4,a
jr z,WAIT_O1
dec bc
ld a,b
or c
ld a,(hl)
jr nz,OUT1
ld a,d
OUT1 out (PORTA),a
xor d
ld d,a
inc hl
ld a,b
or c
ld a,16
jr nz,OUT2
ld a,16+32
OUT2 out (PORTC),a
WAIT_O2 ld a,127
in a,(254)
rra
jr nc,BREA_O
in a,(PORTB)
bit 7,a
jr nz,BREA_O
bit 4,a
jr nz,WAIT_O2
bit 5,a
jr nz,OUT_ERR
ld a,b
or c
jr nz,GO_OUT
xor a
out (PORTC),a
OUT_END pop hl
pop bc
ret

```

```

OUT_ERR  dec  e
          jr   z,BREA_O
          pop  hl
          pop  bc
          jr   OUT_AG

BREA_O   ld   a,128
          out  (PORTC),a

PAUS_O   ex   (sp),hl
          ex   (sp),hl
          dec  a
          jr   nz,PAUS_O
          scf
          jr   OUT_END

;IN: HL kam ukladat data
;      BC max delka bufferu

;OUT: c break/CRC errorry
;      nc prijato bez chyby
;      BC delka prijatych dat
;      HL beze zmeny

BYTE_IN  di
          push hl
          push bc
          ld   bc,0
          ld   d,c

GO_IN    ld   a,127
          in   a,(254)
          rra
          jr   nc,BREA_I
          in   a,(PORTC)
          bit  7,a
          jr   nz,BREA_I
          bit  4,a
          jr   nz,GO_IN
          ld   a,16
          out  (PORTB),a

WAIT_I1  ld   a,127
          in   a,(254)
          rra
          jr   nc,BREA_I
          in   a,(PORTC)
          bit  7,a
          jr   nz,BREA_I
          bit  4,a
          jr   z,WAIT_I1
          bit  5,a
          in   a,(PORTA)
          jr   nz,IN_ALL

;porovnani na max delku volného
mista pro packet
          ex   af,af'
          ld   a,d
          pop  de
          push de

          push hl
          or   a
          ld   h,b
          ld   l,c
          sbc  hl,de
          pop  hl
          jr   z,BREA_I
          ld   d,a
          ex   af,af'

          ld   (hl),a
          inc  hl
          inc  bc
          xor  d
          ld   d,a
          xor  a
          out  (PORTB),a
          jr   GO_IN

IN_ALL   cp   d
          ld   a,0
          jr   z,IN_END
          ld   a,32
          out  (PORTB),a
          pop  bc
          pop  hl
          jr   BYTE_IN

BREA_I   ld   a,128
          out  (PORTB),a

PAUS_I   ex   (sp),hl
          ex   (sp),hl
          dec  a
          jr   nz,PAUS_I
          ld   a,128
          scf

IN_END   out  (PORTB),a
          pop  hl
          pop  hl
          ret

A08255   equ  $-X1_INIT
A08255FR equ  512-A08255
          defs A08255FR

A08255   00250 ;délka ovladače
A08255FR 00262 ;volné místo
A0FTPADR 57600 ;zde je L1 ovladač
(utilita MFC = CODE 55296,2816)

X1_INIT 57600
X2_SETOU 57602
X3_SETIN 57604
X4_BOUT  57606
X5_BIN   57608
X6_ENDOU 57610
X7_ENDIN 57612
X8_REOUT 57614
X9_REIN  57616

```

## Podrobnosti k protokolu *specckyFTP*

Popis dále slouží především pro vývojáře, kteří chtějí protokol *specckyFTP* podpořit ve vlastních programech.

Každá síťová komunikace se dá rozdělit do několika vrstev. Pro naše účely postačí definovat si 4 vrstvy (úrovně):

### Level 0 (nultá úroveň)

Jde o fyzický hardware (např. RS-232, 8255) a definování propojení počítačů mezi sebou. Nejčastěji propojovací kabel.

### Level 1 (první úroveň)

Zde patří ovladač (software) pro hardware z úrovně 0. Ovladač musí umět pracovat s pakety a měl by (tj. nemusí, ale opravdu je dobré, aby to uměl a nenechal to až na posouzení konečným uživatelem...) zajistit spolehlivý přenos. **Ovladač musí být relokovatelný** (protože je nutné mít možnost jej umístit a používat kdekoliv v paměti), **mít délku maximálně 512 bytes, a obsahovat 9 podprogramů**, které je možné volat z vyšších úrovní. Dobrou vizitkou programátora také je, že počítá s umístěním L1 i v ROM, kde jak víme se nedá zapisovat, a tedy se nemohou použít techniky zápisu do vlastního kódu.

*Paket = blok bytes o určité velikosti (0 - 65535)*

*Spolehlivý přenos = data jsou přenesena bez chyb (resp. umět rozeznat, že přijatá data jsou chybná)*

Je-li ovladač nahrán na adresu *ADR*, pak musí platit:

**ADR+0** (X1) - počáteční inicializaci hardware

**ADR+2** (X2) - inicializaci vysílání

**ADR+4 (X3)** - inicializaci příjmu**ADR+6 (X4)** - odeslání jednoho paketu

Vstup:     *HL* = adresa se začátkem dat, která se mají odeslat  
              *BC* = délka dat  
              *E* = počet opakování (jednička znamená pouze 1 pokus data odeslat)

Výstup:     *carry* = CRC error s vyčerpáním všech pokusů, nebo break  
              *non carry* = odesláno OK  
              *HL, BC* stejné jako na vstupu

**ADR+8 (X5)** - příjem jednoho paketu

Vstup:     *HL* = adresa pro uložení dat  
              *BC* = maximální délka bufferu pro paket (256 default)

Výstup:     *carry* = CRC error (a vysílač vyčerpал všechny pokusy),  
                  break, a nebo přetečení bufferu  
              *non carry* = data přijatá bez chyby  
              *HL* = stejné jako na vstupu, *BC* = délka přijatých dat

**ADR+10 (X6)** - ukončení celého vysílání (navrácení hardware do nějakého bezpečného stavu)

**ADR+12 (X7)** - ukončení celého příjmu ((navrácení hardware do nějakého bezpečného stavu)

**ADR+14 (X8)** - zatím nepoužito, rezervováno do budoucna

**ADR+16 (X9)** - zatím nepoužito, rezervováno do budoucna

Všechny podprogramy mohou při své práci používat libovolné registry Z80 včetně alternativních!

Autor aplikace by měl vždy uživatele informovat v manuálu o tom, kde přesně se level 1 v paměti nachází. To proto, aby uživatel měl možnost vzít úplně jiný ovladač hardware a úplně jednoduše výměnou 512 bytes provozovat *specyFTP* s ním.

Tímto je zajištěna modularita (oddělenost) a zcela zrušena pracnost spojená s přepisováním celé aplikace pro jiný typ hardware.

## Level 2 (druhá úroveň)

Samotný protokol *specyFTP*. Definuje pakety, dává jim význam a logiku. Mohou existovat tyto pakety:

**InfoPacket** (128 bytes) – tzv. hlavička souboru

| Offsest | Délka | Význam                                                                                                                                                                                                                                                                                       |
|---------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0       | 1     | info byte (identifikace programu, který data posílá)<br>nutné uvést pokud se používá offset 21-127<br><br>0 - k dispozici pouze povinné info o souboru<br>1 - MFC (offset 32 až 63 = 32 bytes z DIR - MDOS)<br>2 - MFC (offset 32 až 63 = 32 bytes z DIR - MSDOS)<br>3-255 - zatím nevyužito |
| 1       | 1     | přípona souboru:<br><br>"P" Basic Program<br>"B" Bytes<br>"N" Number array<br>"C" Character array<br>může být ale i jakýkoliv jiný kód ASCII (0-255);<br><br>pokud bodyflag<>0, jde o bezhlavičkový blok dat bez ohledu na příponu!                                                          |
| 2-11    | 10    | jméno souboru<br>může obsahovat kódy 0-255<br>"file1" a "file1 " jsou 2 naprosto rozdílné soubory (nesmí se tedy doplňovat mezerami jako v TAP, je třeba použít nuly)!                                                                                                                       |
| 12-15   | 4     | délka souboru ( <b>32bit</b> )                                                                                                                                                                                                                                                               |
| 16,17   | 2     | délka Basic ("P") programu bez proměnných (jinde nemá význam = 32768)                                                                                                                                                                                                                        |
| 18,19   | 2     | startovací adresa/řádek                                                                                                                                                                                                                                                                      |
| 20      | 1     | BodyFlag (nenulové pouze u bezhlavičkového souboru!)                                                                                                                                                                                                                                         |

|        |    |                                                                                                                                                                                                                                                                                                                                                   |
|--------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21-63  | 43 | reserved (nepoužito)<br>může libovolně používat aplikační program<br><br>32-63 využívá MFC pro kopii z původního DIR přenášeného souboru                                                                                                                                                                                                          |
| 64-85  | 22 | speciální sekvence kontrolních bytes díky kterým lze snadno odhalit HW závadu na propojení počítačů, neboť nebudou-li v přijatém infopacketu tyto hodnoty, tak je z hardwarového hlediska něco špatně a přenos nemůže pokračovat (22 bytes oné kontrolní sekvence:<br>0,1,2,4,8,16,32,64,128,255,254,253,251,247,239,223,191,127,129,153,126,102) |
| 86-127 | 42 | reserved (nepoužito)<br>může libovolně používat aplikační program                                                                                                                                                                                                                                                                                 |

Ať už existuje jakýkoliv systém souborů, aplikační software musí vždy zajistit transformaci vlastností souboru do povinného formátu uvedeného výše (offset 1-20). Dobrovolné informace (offset 21-63 a 86-127) používat nemusí, ale jako vývojáři jistě chápete, že jsou neméně důležité, protože se tak alespoň při přenosu mezi stejným operačním systémem neztratí...

Teoreticky by jediná povinná informace mohla být jen délka dat (anebo ani ta ne), vše ostatní by mohlo být dobrovolné. Ovšem k takovému přenosu už nepotřebujeme „protokol“ (protože jde jen o sekvenci bajtů bez informací o nich). Přeci jen se snažím jít se *specyFTP* „výše“ a vědět s jakým souborem a operačním systémem mám co dočinění. Bude-li se například jednat o nejhorší možný scénář, tj. soubor z PC, u kterého samozřejmě znáte jen jeho jméno a velikost, pak ostatní povinné informace si musíte vymyslet. Stejně, pouze uživatel ví, čeho tím přenosem chce docílit a on si po přenosu ty soubory už ručně upraví... Mezi systémy ZX Spectra byste však s vyplněním povinných informací neměli mít problémy. Spíše Vám bude ještě místo na uložení chybět. Proto je v InfoPacketu na prvním místě ten infobyte, kterým si lze definovat operační systémy a tedy i informace navíc, které nechcete přenosem ztratit.

Lze předpokládat, že v budoucnu bude existovat více druhů InfoPacketu (tzn. s jinou délkou než 128 bytes), které budou řešit narůstající požadavky uživatelů (např. podpora adresářů,

diagnostika přenosové cesty, přenos celého image diskety). Proto InfoPacket o velikosti 128 bytes berte jako takovou první vlašťovku. Počet druhů InfoPacketů je samozřejmě omezený – celkem jich může být maximálně 255.

### **DataPacket** (1...65535 bytes)

Vlastní data patřící souboru. Musí následovat po InfoPacketu o velikosti 128 bytes. Délka DataPacketu může být dlouhá až 65535 bytes. Příjmací počítač ovšem není povinen velké pakety přijmout. **Velikost, kterou musí přijmout je 256 bytes**, cokoliv nad to, může odmítnout (vyhodit uživateli error). Proto je dobré, aby aplikační software umožnil uživateli nastavit maximální velikost odesílaného paketu. Počet DataPacketů je samozřejmě dán pouze délkou souboru, přičemž ale není nikde podmíněno, že všechny DataPakety patřící jednomu souboru musí mít pokaždé stejnou velikost (tj. klidně pošlete různé délky, jestliže vám to pro level 1 hraje do karet). Důležité je i zmínit, že ten minimální počet 256 bytes jsou **čistá data souboru** ukládána do paměti. Nepatří tam kontrolní mechanismy (např. XOR byte navíc) – to si řeší L1 v jiném paměťovém prostoru.

### **Close/EndPacket** (0 bytes)

Speciální varianta InfoPacketu. Říká, že soubor se má uzavřít, tedy že se už celý přenesl (ClosePacket). V určitých případech také znamená korektní ukončení celého přenosu, tj. všechny soubory už byly přeneseny (EndPacket).

Po všech DataPacketech patřících jednomu souboru musí vždy následovat CloseEndPacket. Kdyby totiž ne a místo něj přišel InfoPacket, tak to aplikační soft bude chápat jako další DataPacket a data přidá k souboru!



### Level 3 (třetí úroveň)

Znázorňuje obrázek (viz. dále). Jde v podstatě o chování aplikačního software. Pouze on pracuje se souborem jako celkem a řídí jeho vysílání/příjem.

*Poznámka:*

*Chybové hlášení je znázorněno před X6/X7. Pozor však, aby X6/X7 nedostalo slovo až po reakci uživatele (např. čekání až stiskne libovolnou klávesu). To by bylo špatně, protože HW by měl být převeden po chybě do tzv. bezpečného (výchozího) stavu co nejdříve.*

**Nesmírně důležité je mít jednotlivé vrstvy oddělené. Jedině tak je zajištěna modulárnost a tedy hardwarová nezávislost:**

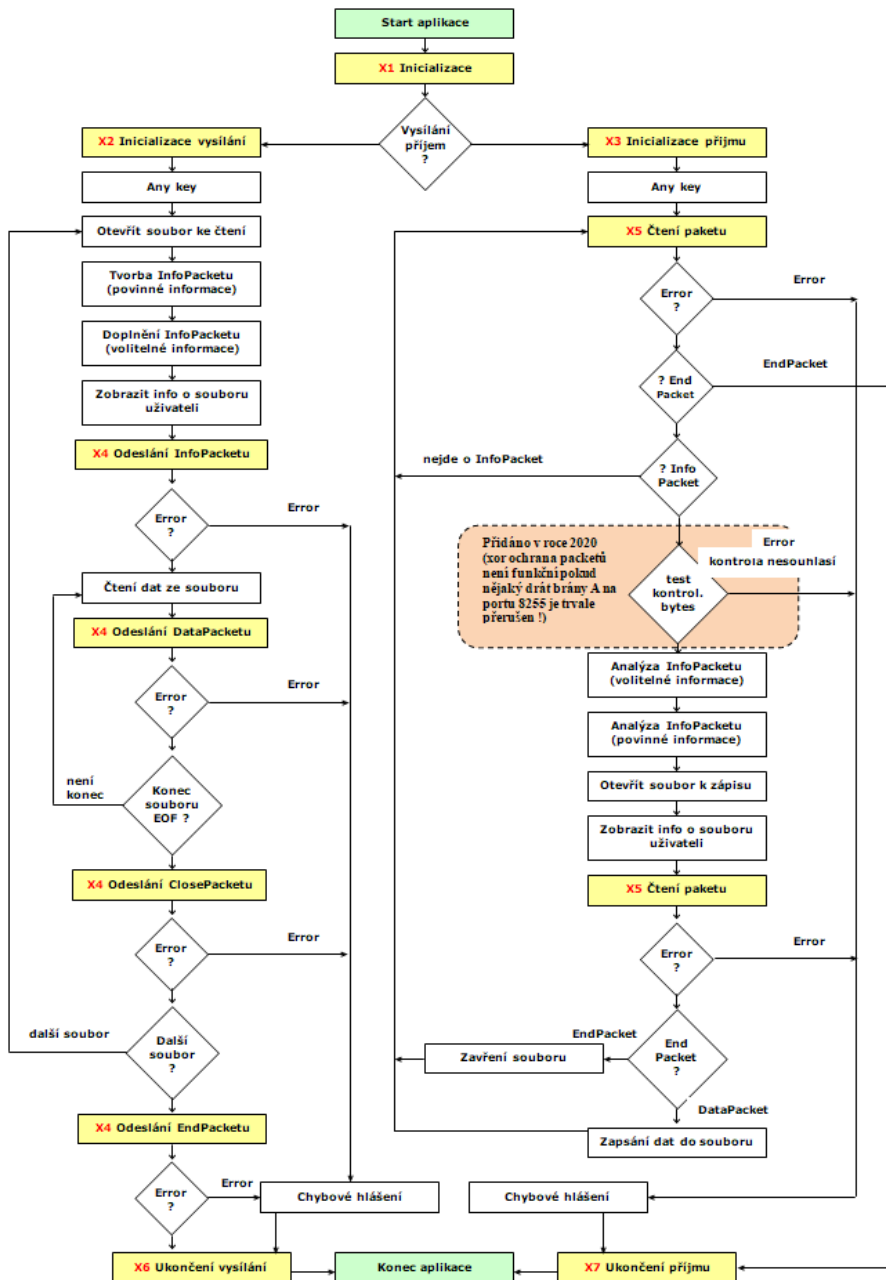
Level 0 pouze říká, jakým železem a drátem má být přenos uskutečněn

Level 1, jaký relokovatelný ovladač se má pro přenos paketů použít

Level 2 je souborový protokol

Level 3 pak konečná aplikace

Pouze level 3 leží na bedrech programátorů, kteří chtějí *specyFTP* do svých výtvorů zabudovat. Level 0 je defakto nezajímá, na level 1 vyčlení pouze 512 volné místo, a level 2 je teorie, dle které vytvoří level 3 aplikaci.



[mts.zxs@tiscali.cz](mailto:mts.zxs@tiscali.cz)

(připomínky jsou vítány)

© MTs