

ChnSof

POPIS KOMUNIKAČNÍHO PROTOKOLU FIRMY SOFCON SPOL. S R.O.

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 24.08.2004

Datum posledního uložení dokumentu: 24.08.2004

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.	O dokumentu	5
1.1.	Revize dokumentu	5
1.2.	Účel dokumentu	5
1.3.	Rozsah platnosti	5
1.4.	Související dokumenty	5
2.	Termíny a definice	5
3.	Úvod	6
4.	Fyzická vrstva	6
5.	Linková vrstva	6
5.1.	Základní struktura	6
5.1.1.	Pravidla	7
5.2.	Výpočet CRC16	7
5.2.1.	Výpočet CRC16 v kódu 8086	7
5.2.2.	Výpočet CRC16 v kódu 8080 a Z80	8
5.2.3.	Výpočet CRC16 v kódu MCS51	9
5.3.	Zajištění transparence přenášených dat	11
6.	Síťová vrstva	11
6.1.	Základní struktura	11
6.2.	Pravidla	12
6.3.	Potvrzování rámců	12
6.4.	Opakování rámců	12
7.	Obsah datové části CDATA	13
7.1.	Data formát 0	13
7.2.	Data formát 1	13
7.3.	Data formát 2	14
7.3.1.	Obsah pole DEST a SOUR	14
7.4.	Obsah pole CMD	15
7.5.	Obsah datové části REC	15
7.5.1.	Identifikátor typů TREC	15
7.5.2.	Typy datové části RDATA	16
7.6.	Pravidla	17
7.7.	Potvrzování zpráv	17
7.8.	Opakování zpráv	17
8.	KIT-BUILDER a PASCAL	18
8.1.	KIT-BUILDER	18
8.2.	KIT-Pascal	18
9.	Příklady zpráv	18
9.1.	Příklady zpráv	18
9.1.1.	Přenosový rámec - linková vrstva	18
9.1.2.	Test komunikace - linková i síťová vrstva	19
9.1.3.	Číslování a potvrzování rámce - síťová vrstva	20
9.1.3.1.	MASTER ani SLAVE nežádají potvrzení rámce	20
9.1.3.2.	MASTER nežádá a SLAVE žádá potvrzení zprávy	20
9.1.3.3.	MASTER žádá a SLAVE nežádá potvrzení zprávy	21
9.1.3.4.	MASTER i SLAVE žádají potvrzení zprávy	21
9.1.4.	Zaslání zprávy formátu DF2 s číslováním a potvrzováním zprávy - linková, síťová vrstva a aplikační proces	21

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX			První vydání
1.10	1.XX	Tu	9.6.2003	Úprava dokumentu dle ISO9000. Oprava kapitoly 9 – jiné délky.
1.11	1.XX	Hv	12.09.2003	Změna formulací definic Oprava chyb případně doplnění textů u příkladů
1.12	1.XX	Tes	24.08.2004	Doplněn algoritmus výpočtu CRC16 pro 8080, Z80, MCS51

1.2. Účel dokumentu

Tento dokument slouží jako popis komunikačního protokolu firmy SofCon pro připojení informačních systémů.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba orientovat se v používání programového vybavení SofCon.

Tuto definici síťové vrstvy doplňují manuály popisující konkrétní knihovny fyzické vrstvy a linkové vrstvy. Vlastní síťová vrstva je popsána v manuálu ChnSofs2.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

V dokumentu jsou popsány komunikační vrstvy fyzické, linkové a síťové pro přenos zpráv mezi paralelně běžícími procesy používanými ve firmě SofCon[®] s.r.o. .

Fyzická vrstva popisuje, přes která rozhraní je možno komunikaci provést.

Linková vrstva popisuje základní rámec přenášených dat. Obsahuje definici řídicích znaků, způsob adresace v komunikační síti, zabezpečení přenášených dat a zajištění transparence dat.

Síťová vrstva popisuje sémantiku přenosu libovolných zpráv oběma směry po síti Master - Slave resp. Master - množina Slave. Definuje formáty a obsahy identifikačních a datových polí, způsoby potvrzování, zabezpečení opakování, ...

V poslední kapitole jsou pak omezení jednotlivých zpráv při komunikaci s KIT-BUILDER a při programování v PASCALu.

4. Fyzická vrstva

Fyzická komunikační vrstva může být tvořena rozhraním RS232, RS485, RS422, telefonním modemem, GSM modemem, radiovým modemem a jiným.

Pro tyto fyzické vrstvy jsou v programovém vybavení firmy SofCon vytvořeny komunikační knihovny ChnCom, ChnCom2, ChnV40, ChnComM, ChnV40M, ChnRacom.

5. Linková vrstva

Linková vrstva definuje pravidla přenosu mezi dvěma uzly, délku bloků, způsob zabezpečení dat zajištění transparence přenášených dat. Je zde popsán linkový protokol používaný ve firmě SofCon[®] s.r.o.

5.1. Základní struktura

Všechny přenosy se uskutečňují pomocí zpráv mající níže popsaný rámec, zabezpečení a transparentnost.

Linková vrstva je v programovém vybavení firmy SofCon implementována např. v knihovně ChnPrt.

		Rámec zprávy						
délka v bytech		1	1	1	2	LEN	2	1
obsah		SOH	DNODE	SNODE	LEN	DATA	CRC	ETX

SOH počátek zprávy, 001h

DNODE fyzická adresa příjemce v síti <0,255>

Je-li adresa rovna 0, pak přijímají všichni ale na zprávu žádný neodpovídá.

SNODE fyzická adresa odesílatele v síti <0,255>

LEN délka zprávy DATA v bytech <0, 32000>
 DATA vlastní datová zpráva
 CRC zabezpečení dat $CRC16 = x^{16} + x^{15} + x^2 + 1$
 ETX konec zprávy, 003h

5.1.1. Pravidla

- 1) Přenosový rámec je shodný při komunikaci Master - Slave i Slave - Master
- 2) Zbytek po dělení je generován z SOH, DNODE, SNODE, LEN a DATA.
- 3) Pole SNODE je plněno číslem uzlu, který zprávu odesílá.
- 4) U polí LEN a CRC je nižší byte je zaslán první.
- 5) Je-li adresa adresáta DNODE rovna 0, pak přijímají všichni účastníci na komunikační lince, ale na zprávu žádný neodpovídá.

5.2. Výpočet CRC16

5.2.1. Výpočet CRC16 v kódu 8086

```

unit Crcl6;
interface
uses
  Objects;
  {
  {=====
  {
  {   unit uCrcl6 - jednotka pro vypocet Crcl6
  {
  {   (C)1992 Vladimir Kastner, Na Vlcovce 6, Praha 6
  {
  {=====
  {
  { (C) P.Tesar, T.Pecina, LP Praha, December 1989 }
  { Vypocet CRC-16 (IEEE MICRO 83) }
  { generacni polynom = x16 + x15 + x2 + 1 }
  { nejnizsi bit B je prvni prijaty nebo zasilany }
  {
type
  pCrcl6 = ^ tCrcl6;
  tCrcl6 = object(tObject)
    Residue : Word;           { x15 je umisten v bitu B0 }
    constructor Init;         { vytvoreni objektu }
    procedure SetResidue(Res: Word); { definuje hodnotu zbytku po deleni }
    function GetResidue: Word;   { navrati hodnotu zbytku po deleni }
    procedure MakeCrc(B: Byte);  { vypocet zbytku a jeho navraceni }
  end;

implementation

constructor tCrcl6.Init; { vytvoreni objektu }
begin
end;

procedure tCrcl6.SetResidue(Res: Word); {definuje hodnotu zbytku po deleni}
begin
  Residue:=Res;
end;

function tCrcl6.GetResidue: Word; { navrati hodnotu zbytku po deleni }
begin
  GetResidue:=Residue;
end;

```

```

procedure tCrc16.MakeCrc(B: Byte); { vypocet zbytku a jeho navraceni }
label Nc1,Nc2;
begin
  asm
    mov al,B
    les bx,Self
    xor al,byte ptr es:[bx].Residue
    mov dl,al
    add dl,dl
    xor dl,al
    add al,al
    mov dh,2
    jc Nc1
    mov dh,0
  Nc1:
    or al,al
    jpe Nc2
    xor dh,3
  Nc2:
    mov al,dh
    rcr al,1
    add dx,dx
    add dx,dx
    add dx,dx
    add dx,dx
    add dx,dx
    add dx,dx
    or dl,al
    xor dl,byte ptr es:[bx+1].Residue
    mov es:[bx].Residue,dx
  end;
end;
end.

```

5.2.2. Výpočet CRC16 v kódu 8080 a Z80

```

;POPIS CINNOSTI:
;   PODPROGRAM CRCF JE EFEKTIVNI REALIZACI RUTINY
;PRO VYPOCET CRC-16 (CYCLIC REDUNDANCY CHECK).
;GENERACNI POLYNOM:
;   X^16 + X^15 + X^2 + 1
;PRIRAZENI BITU:
; ADR                ADR+1                ADR+2
; MSB                LSB                MSB                LSB                MSB                LSB
; X^16.....X^23    X^8.....X^15    X^0.....X^7
;
;ZAKLADNI VLASTNOSTI CRC JE:
;   NECHT CRC(AAAA,BBBB)=CCDD
;
;KDE AAAA ... POCATECNI ADRESA OBLASTI PAMETI
;   BBBB ... KONCOVA ADRESA OBLASTI PAMETI
;   CCDD ... CRC-16 Z OBLASTI AAAA-BBBB VCETNE
;
;POTOM, ZAPISEME-LI NA ADRESU BBBB+1 BYTE DD
;   A NA ADRESU BBBB+2 BYTE CC
;
;MUSI VYJIT: CRC(AAAA,BBBB+2)=0000H
;
;CRC-16 POUZIVAME NAPR. PRI KONTROLE OBSAHU PAMETI EPROM
;PRO VYPOCET CRC Z OBLASTI PAMETI JE TREBA PODPROGRAM
;CRCF UZAVRIT VNEJSI SMYCKOU PRES ADRESY OBLASTI.
;
;VSTUPNI REGISTRY: A - BYTE
;                   BC - CRC Z PREDCHOZIHO KROKU
;VYSTUPNI REGISTRY:BC - CRC VYPOCTENE V PROVEDENEM KROKU
;ZMENENE REGISTRY: A, FLAGS, HL, BC
;
;PRIKLAD: PRO SAPI-1
;   ORG                7000H
ADRPOC EQU                1000H                ;NAPRIKLAD
ADRKON EQU                1FFFH                ;NAPRIKLAD
HILO EQU                  136H                ;SAPI-1 MONITOR
;   LXI                H,ADRPOC
;   LXI                D,ADRKON
;   LXI                B,0000H                ;CRC=0000
CYKL: MOV                A,M

```



```

        PUSH      H
        CALL     CRCF      ;VYPOCET CRC-16
        POP      H
        CALL     HILO      ;HL+1 : DE SET CY DLE POROVNANI
        JNC      CYKL
;      ...      BC=CRCF-16 Z OBLASTI
;      ADRPOC -- ADRKON
;

;PODPROGRAM PRO VYPOCET CRC-16
; ( IEEE MICRO 83 )
;-----
CRCF: XRA      C
      MOV      L,A
      MOV      H,A
      ADD      A
      XRA      L
      MOV      L,A
      MOV      A,H
      ADD      A
;=====
;      MVI      A,0      ;ORIGINAL 8080
;      JPE      CRCF1
;      MVI      A,03H
;CRCF1:JNC     CRCF2
;      XRI      02H
;CRCF2:MOV    H,A
;=====
; UPRAVA PRO 8080 & Z80
      MVI      H,10B
      JC       CRCF1
      MVI      H,00B
CRCF1:ORA     A
      JPE      CRCF2
      MVI      A,11B
      SUB     H
      MOV     H,A
CRCF2:MOV    A,H      ;CY=0
;-----
      RAR
      DAD     H
      DAD     H
      DAD     H
      DAD     H
      DAD     H
      DAD     H
      DAD     H
      ORA     L
      XRA     B
      MOV     C,A
      MOV     B,H
      RET
      END

```

5.2.3. Výpočet CRC16 v kódu MCS51

```

;=====
;§
;§ implementace CRC pro mikroradice typu i51 §
;§
;§ (C)2001 Viktor Pohorely, vytvoreno pro SofCon s.r.o. §
;§
;§ Stresovicka 49, Praha 6 - Stresovice §
;§
;§
;=====a
;§
;§ vychozi knihovna : unit uCrcl6.pas §
;§
;§ (C)1992 Vladimřr Kastner, Na Vlřtovce 6, Praha 6 §
;§
;§
;=====L
; Vypocet CRC-16 (IEEE MICRO 83)
; generujici polynom = x16 + x15 + x2 + 1
; nejnizsi bit B je prvni prijaty nebo zasilany
; revision 1.0
; otestovano 9.2.2001

```

```

;funkce CRC16 zajistuje vypocet zbytku po deleni generujicim
polynomem

;uzivatel musi zajistit vynulovani promenne Residue pri vypoctu CRC
;pro novy balik dat

;naroky na pamet
;2B RAM - ulozeni zbytku (Residue)
;Stack - 2B navratova adresa + 1B na prubezne ukladani
;nici registry A,R0,R1

;!!Residue data 32h      ;zbytek po deleni (word)

;procedura pro vypocet CRC-16
;parametry predavany v dvojici ACC (Data:Byte) a Residue
(Zbytek:Word)
CRC16:
    xrl A,Residue      ; xor al,byte ptr es:[bx].Residue
    mov R1,A
    clr C
    rlc A              ; add dl,dl
    xrl A,R1          ; xor dl,al
    mov R0,A
    mov A,R1
    clr C
    rlc A              ; add al,al ;nastaveni carry
    mov R1,#2         ; mov dh,2

    jc Nc1            ; jc  Nc1
    mov R1,#0         ; mov dh,0
Nc1:
    mov C,P            ; or  al,al
    mov A,R1
    jnc Nc2            ; jpe Nc2
    xrl A,#3          ; xor dh,3
Nc2:
    mov R1,A          ; mov al,dh
    clr C
    rrc A              ; rcr al,1
    push ACC
    mov A,R0
    mov R0,#6
Adding:  clr C
    rlc A
    xch A,R1
    rlc A
    xch A,R1
    djnz R0,Adding
    mov R0,A
    pop ACC           ; 6 * add dx,dx

    orl A,R0          ; or  dl,al
    xrl A,Residue+1  ; xor dl,byte ptr es:[bx+1].Residue
    mov Residue,A
    mov Residue+1,R1; mov es:[bx].Residue,dx

    ret
;end

```

5.3. Zajištění transparence přenášených dat

Při vysílání zprávy jsou jednotlivé byte zakódovány takto:
 SOH - zakódován jako dvojice DLE,SOH -> 010h,001h
 ETX - zakódován jako dvojice DLE,ETX -> 010h,003h
 ostatní jednotlivé byte zprávy (DNODE, SNODE, LEN, DATA, CRC), mající hodnotu DLE jsou zakódovány jako DLE,DLE -> 010h,010h.

Tím je umožněno zkonstruovat přijímač, aby byl v proudu znaků schopen poznat počátek zprávy (vždy kombinace DLE,SOH). Jako další byte zprávy následuje DNODE=adresa přijímače. (ve výjimečném případě DNODE=DLE zakódovaném jako DLE,DLE). Pokud zpráva není určena pro příslušný přijímač, může být kompletně příjem ignorován až do opětovného nalezení počátku zprávy.

6. Síťová vrstva

6.1. Základní struktura

Protokol musí být schopen přenášet obousměrně zprávy CDATA mezi dvěma komunikujícími procesy.

Pro tuto linkovou vrstvu je v programovém vybavení firmy SofCon vytvořena komunikační knihovna ChnSofs2

Význam pole DATA linkové vrstvy

	DATA							
délka v bytech	1							LEN-1
obsah	CODE							CDATA
bit	7	6	5	4	3	2	1	0
obsah	FRM		DATx			ACKx		

význam pole FRM

FRM		
hodnota	název	formát přenášených dat
00	TST	test spojení, data CDATA nejsou obsažena
01	DF1	data formát 1, bez logických adres
10	DF2	data formát 2, s logickými adresami
11	DF0	data formát 0, FRM(11)+DATx+ACKx ve významu identifikátoru zprávy

význam pole DAT

DAT		
hodnota	název	číslo zprávy
000	DAT0	není zasílána žádná zpráva CDATA
001 až 110	DATx	je zasílána zpráva CDATA číslo 1 až 6, vyžadováno ACKx
111	DAT7	je zasílána zpráva CDATA číslo 7, nevyžadováno ACK

Pro číslování zpráv platí :
 $NEW\text{číslo} = (OLD\text{číslo} \bmod 6) + 1$

význam pole ACK

ACK		
ACK	název	číslo potvrzení ACK
000	ACK0	neposíláno žádné potvrzení
001 až 110	ACKx	posíláno potvrzení ACK na došlou zprávu č.1 až 6
111	NACK	posíláno negativní potvrzení

6.2. Pravidla

- 1) Komunikace na komunikační síti je MASTER - SLAVE.
- 2) MASTER zasílá rámce, SLAVE zasílá rámce jako odpovědi.
- 3) SLAVE komunikuje pouze jedním typem formátu dat DF1 nebo DF2.
- 4) MASTER komunikuje s daným uzlem pouze formátem, který daný uzel zaslal jako odpověď na test spojení v poli FRM.
- 5) Při číslování a potvrzování rámců MASTER vysílá rámce s číslem v poli DAT a potvrzuje SLAVE uzlu naposledy přijatý rámec v poli ACK. SLAVE přijme rámec, v odpovědi zašle rámec s číslem v poli DAT a potvrdí MASTER uzlu příjem rámce v poli ACK.
- 6) Při zaslání odpovědi SLAVE uzlem je pole DNODE linkové vrstvy naplněno hodnotou přijatou v poli SNODE.

6.3. Potvrzování rámců

- 1) Pro potvrzování rámců slouží pole DAT a ACK. Uzel čísluje vysílané rámce od 1 do 6. Pokud uzel neobdrží rámec s potvrzením zaslání rámce v poli ACK, je rámec znovu odvílán.
- 2) Pokud nechceme využívat potvrzování rámců nastavíme hodnotu pole DAT = DAT7. Hodnotu DAT = DAT0 použijeme při testu komunikace s daným uzlem. Odpověď bez potvrzení obsahuje v poli ACK hodnotu ACK0.

6.4. Opakování rámců

- 1) MASTER uzel udržuje přehled o SLAVE uzlech, které jsou na dané komunikační síti připojeny. K tomuto účelu využívá zprávu pro test spojení, pole FRM = 00. Pokud SLAVE uzel do časového intervalu odpoví, je veden jako dostupný a je možné na něj zasílat zprávy. SLAVE uzel v odpovědi na příkaz test pošle v poli FRM typ formátu přenášených dat. Tímto formátem se poté bude s daným uzlem komunikovat. V případě, že uzel není dostupný je na něj zasílaná zpráva zrušena. Test spojení MASTER uzlu i s nepřipojenými SLAVE uzly, je prováděn periodicky, v jednom cyklu je testováno několik SLAVE uzlů a poté následuje pauza, aby nedocházelo k zahlcení komunikační sítě.
- 2) Pokud MASTER uzel do časového intervalu nedostane potvrzení na odeslaný rámec, je rámec znovu odvílán. Nezávisle-li MASTER uzel ani po x opakováních odpovědi, je uzel sítě prohlášen za nedostupný.

- 3) Odpověď zasílaná od SLAVE uzlu musí mít formát stejný, jako byl použit ve zprávě z MASTER uzlu.

7. Obsah datové části CDATA

Při formátu DF0 přenášených dat zajišťuje aplikační proces naplnění obsahu datové části DATA a adresu cílového uzlu DNODE.

Při formátu DF1 přenášených dat zajišťuje aplikační proces naplnění obsahu datové části CDATA a adresu cílového uzlu DNODE.

Při formátu DF2 přenášených dat zajišťuje aplikační proces naplnění obsahu datové části CDATA, adresu cílového uzlu DNODE odvozuje komunikační proces z hodnoty LOGA.

7.1. Data formát 0

Protokol formátu 0 - DF0 musí být schopen přenášet zprávy CDATA mezi dvěma uzly sítě, z nichž jeden je ve funkci MASTER a druhý ve funkci SLAVE. Není zde zavedeno potvrzování zasílaných a přijímaných rámců (komunikace na úrovni linkové vrstvy). Potvrzování a zabezpečení může být realizováno přímo komunikujícím procesem v MASTER uzlu.

	CDATA formát 0 - DF0	
délka v bytech	LEN-1	
obsah	REC	

Identifikátor zprávy je dán polem CODE = FRM(11) + DATx + ACKx. Vyjadřuje zároveň 2 informace :

- typ rekordu přenášených dat (vyjadřuje počet, délku a sémantický význam jednotlivých položek data-rekordu).

- sémantiku, která se má provést po příjmu zprávy (test komunikace, žádost o data, zaslání dat).

REC vlastní přenášená data. Význam jednotlivých položek rekordu a jejich celkový počet je určen konkrétní hodnotou typu TREC.

Délka REC je dána velikostí LEN-1, sémanticky musí souhlasit s předpokládanou délkou v závislosti na použitém identifikátoru zprávy.

7.2. Data formát 1

Protokol formátu 1 - DF1 musí být schopen přenášet obousměrně zprávy CDATA mezi dvěma komunikujícími procesy v Master uzlu a Slave uzlu. Logické adresy, číslování a potvrzování zpráv není zavedeno.

	CDATA formát 1 - DF1	
délka v bytech	1	LEN-2
obsah	CMD	REC

CMD sémantika, která se má provést po příjmu zprávy.
 REC vlastní přenášená data.

7.3. Data formát 2

Protokol formátu 2 - DF2 musí být schopen přenášet obousměrně zprávy DATA mezi dvěma nebo více procesy v Master uzlu a Slave uzlu. Jsou zavedeny logické adresy odesílatele a příjemce, číslování a potvrzování zpráv. Odesílatelem zprávy může být proces v MASTER uzlu i ve SLAVE uzlu. Odesílatel zprávu odešle a příjemce přijetí zprávy potvrdí.

	CDATA formát 2 - DF2						
délka v bytech	13			1 LEN-15			
obsah	HEADER					CMD	REC
délka v bytech	2	1	4	4	1	1	
obsah	LEN1	DUMMY	DEST	SOUR	MNO	ANO	

význam pole HEADER

LEN1 délka části CDATA formátu 2, $LEN1 = LEN - 1$
 DUMMY k libovolnému využití. Proces, kterému je zpráva určena, v odpovědi na zprávu zasílá hodnotu pole DUMM zpět.
 DEST jednoznačný identifikátor cílového procesu, pro který je zpráva určena. V odpovědi na zprávu je hodnota přijatého DEST zkopírována do pole SOUR.
 SOUR jednoznačný identifikátor zdrojového procesu, ze kterého je zpráva vysílána. V odpovědi na zprávu je hodnota přijatého SOUR zkopírována do pole DEST.
 MNO číslo zprávy. Čísluje odesílající proces např. $NEWčíslo = (OLDčíslo + 1) \text{ mod } 256$
 ANO potvrzení poslední zprávy. Proces, který přijal zprávu, v odpovědi potvrzuje číslo přijaté zprávy. Hodnota je rovna číslu MNO zprávy, kterou potvrzujeme.
 CMD sémantika, která se má provést po příjmu zprávy.

7.3.1. Obsah pole DEST a SOUR

	DEST a SOUR		
délka v bytech	1	1	2
obsah	IDENT	INST	LOGA

IDENT identifikátor procesu
 INST instance procesu (význam pouze pro procesy, které mají více instancí)
 LOGA jedinečná logická adresa v síti. Z LOGA se odvozuje přenosový kanál,
 DNODE pro přenos po přenosovém kanálu

7.4. Obsah pole CMD

CMD	
hodnota	sémantika zprávy
003h	potvrzení zprávy
040h	žádost o čtení dat
041h	zápis dat
x	další CMD

7.5. Obsah datové části REC

délka v bytech	REC			
	1	2	2	RSIZE * RCNT
obsah	TREC	RADDR	RCNT	RDATA

TREC určuje typ dat přenášených v RDATA
 RADDR logická adresa RDATA, případně prvního parametru v poli typu TREC
 RCNT počet rekordů, v případě RBIT je RCNT = 1
 RDATA vlastní přenášená data, rekordy, pole dat, pole rekordů

7.5.1. Identifikátor typů TREC

TREC			
Hodnota	název	Význam	velikost prvku v bytech
1	RBYTE	pole BYTE	1
2	RWORD	pole WORD	2
3	RINTEGER	pole INTEGER	2
4	RLONGINT	pole LONGINT	4
5	RDWORD	pole DWORD	4
6	RTEXT	pole CHAR	až 256
7	RREAL	pole REAL (formát Pascal, 6 Byte)	6
8	RDATETIME	komprimovaný DATUM a ČAS (formát MS-DOS)	4
9	RBIT	BIT	1
X	y	nové typy dat, rekordů, polí dat, polí rekordů	y

7.5.2. Typy datové části RDATA

pole BYTE, TREC = 1

délka v bytech	RBYTE 1 * RCNT
obsah	BYTE[1] .. BYTE[RCNT]

pole WORD, TREC = 2, nižší byte je zasílán první

délka v bytech	RWORD 2 * RCNT
obsah	WORD[1] .. WORD[RCNT]

pole INTEGER, TREC = 3, nižší byte je zasílán první

délka v bytech	RINTEGER 2 * RCNT
obsah	INTEGER[1] .. INTEGER[RCNT]

pole LONGINT, TREC = 4, nižší byte jsou zasílány první

délka v bytech	RLONGINT 4 * RCNT
obsah	LONGINT[1] .. LONGINT[RCNT]

pole DWORD, TREC = 5, nižší byte jsou zasílány první

délka v bytech	RDWORD 4 * RCNT
obsah	DWORD[1] .. DWORD[RCNT]

pole CHAR, TREC = 6

délka v bytech	RTEXT 1 * RCNT, max 256
obsah	CHAR[1] .. CHAR[RCNT]

pole DREAL, TREC = 7, nižší byte jsou zasílány první

délka v bytech	RREAL 6 * RCNT
obsah	REAL[1] .. REAL[RCNT]

Komprimovaný DATUM a ČAS, TREC = 8, nižší byte jsou zasílány první

délka v bytech	RDATETIME 4					
bit	31..25	24..21	20..16	15..11	10..5	4..0
obsah	rok 0 ~ 1980	měsíc	den	hodina	minuta	sekunda (sudé)

BIT, TREC = 9

délka v bytech	RBIT 1
obsah	hodnota 0 nebo 1

7.6. Pravidla

- 1) Komunikační proces naplní pole DNODE adresou cílového uzlu. Pokud je ve formátu DF2 je adresa uzlu naplněna hodnotou odvozenou z pole LOGA.
- 2) Pole LEN1 obsahuje délku pole CDATA, naplní proces, který zprávu odesílá.
- 3) Pole DUMMY může odesílající proces použít k libovolnému účelu. Proces, který zprávu přijal, naplní pole DUMMY hodnotou získanou z přijaté zprávy.
- 4) Pole DEST a SOUR slouží jako jednoznačné identifikátory adresáta a odesílatele zprávy, jsou naplněny procesem, který zprávu odesílá. Pole jsou složeny z identifikátoru procesu IDENT, instance procesu INST a jedinečné adresy v komunikační síti LOGA. Komunikační proces, naplní pole DNODE hodnotou odvozenou z pole LOGA (transformační funkce $f(\text{LOGA})=\text{DNODE}$). Proces, který odpovídá na přijatou zprávu, naplní pole DEST hodnotou pole SOUR získanou z přijaté zprávy a pole SOUR hodnotou DEST získanou z přijaté zprávy. Komunikační proces, který odpověď odvysílá, naplní pole DNODE hodnotou odvozenou z pole LOGA odpovědi.
- 5) Pole CMD vyjadřuje sémantiku zprávy. Obsahuje příkazy Zápis zprávy, Žádost o čtení zprávy, Potvrzení zprávy atd.
- 6) Pole REC je složeno z pole TREC, RADDR a RCNT. Pole TREC informuje o typu parametrů v poli RDATA. Pole RADDR označuje logickou adresu prvního parametru v poli RDATA, v případě žádosti o čtení označuje logickou adresu prvního parametru, který má být v odpovědi zaslán. Pole RCNT označující počet parametrů v poli RDATA, v případě žádosti o data označuje počet parametrů, které mají být zaslány. RDATA obsahuje vlastní data přenášené zprávy. Může obsahovat pole prvků typu určeného polem TREC. V případě žádosti o data je pole RDATA prázdné.

7.7. Potvrzování zpráv

- 1) Pro potvrzování zpráv slouží pole MNO a ANO. Číslování a kontrolu potvrzení musí zajistit proces, který zprávu odesílá. Proces, který zprávu odesílá, naplní pole MNO číslem odesílané zprávy, pole ANO číslem přijaté zprávy, kterou potvrzuje. Proces, který odpovídá na přijatou zprávu, naplní pole ANO hodnotou pole MNO.
- 2) Pole MNO určuje číslo odesílané zprávy, pole ANO potvrzuje číslo přijaté zprávy. Proces, který zprávu odesílá, naplní pole MNO číslem odesílané zprávy, pole ANO číslem zprávy, kterou potvrzuje.

7.8. Opakování zpráv

- 1) Pokud proces, který odeslal zprávu, do časového intervalu nedostane potvrzení na odeslanou zprávu, pak zprávu znovu odvysílá. Po x opakováních může vyhlásit chybu komunikace s procesem adresáta.

8. KIT-BUILDER a PASCAL

8.1. KIT-BUILDER

Všechny zprávy projektu KIT-BUILDER komunikují pomocí protokolu síťové vrstvy ChnSof formátem 2 - DF2, vyžadují potvrzování rámců. Odesílatelem zprávy je pouze nadřazený systém. Logická adresa je IDENT = 1, INST = 1, LOGA v nižším byte obsahuje číslo komunikačního uzlu.

Omezení rozsahu adres

TADDR	
název	rozsah adres
RBYTE	0..1999
RWORD	0..999
RINTEGER	0..999
RLONGINT	0..499
RDWORD	0..499
RSTRING	0..1999
RREAL	0..199
RDATETIME	0
RBIT	0..15999

8.2. KIT-Pascal

Všechny zprávy používané při programování v jazyku Pascal, mají-li být připojitelné k nadřazenému počítači, budou nuceni komunikovat pomocí síťové vrstvy ChnSof formátem 2 - DF2, identifikátory zpráv a ostatní omezení budou stejná jako v projektu KIT-BUILDER.

9. Příklady zpráv

9.1. Příklady zpráv

9.1.1. Přenosový rámeček - linková vrstva

Zaslání rámečku z uzlu 5 do uzlu 6, pole DATA obsahuje 6 byte „1,2,3,4“

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	6	5	4	0	1,2,3,4	?	?	003h

9.1.2. Test komunikace - linková i síťová vrstva

Test komunikace formát 1 - DF1.

MASTER má adresu uzlu 5 a zasílá test komunikace na uzlu 6.

MASTER vyšle SLAVE

linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	6	5	1	0		?	?	003h

síťová vrstva, ChnSof, DF1

CODE		
FRM	DAT _x	ACK _x
00	000	000

SLAVE vyšle MASTER

SLAVE nemá zprávu pro MASTER

linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	5	6	1	0		?	?	003h

síťová vrstva, ChnSof, DF1

CODE		
FRM	DAT _x	ACK _x
aa=01	000	000

nebo

SLAVE zasílá zprávu pro MASTER

linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	5	6	bbb	bbb		?	?	003h

síťová vrstva, ChnSof, DF1

CODE			CDATA
FRM	DAT _x	ACK _x	
aa=01	ccc	000	Xxx

aa - označuje, jaký formát přenášených dat uzel podporuje. Tímto formátem je nutno poté vždy komunikovat. Může nabývat hodnot 01 pro formát DF1, 10 pro DF2 a 00 pro test.

Nepřijme-li MASTER odpověď do časového intervalu, je daný uzel prohlášen za nedostupný.

Test dostupných i nedostupných uzlů je prováděn opakovaně.

bbb - délka části DATA

ccc - číslo zasílané zprávy

9.1.3. Číslování a potvrzování rámce - síťová vrstva

Zasílání zpráv formát 1 a 2 - DF1 a DF2, bez přenosového rámce

Nepřijme-li MASTER odpověď do časového intervalu, je daný uzel prohlášen za nedostupný a zpráva zrušena.

Je-li vyžadováno potvrzování zpráv a je-li přijata zpráva stejného čísla, je zasláno potvrzení zprávy, ale vlastní zpráva CDATA není předána ke zpracování. Odstranění znovu zasílaných zpráv při chybách na komunikační lince.

Dále budou uvedeny příklady komunikace pomocí formátu DF2.

9.1.3.1. MASTER ani SLAVE nežádají potvrzení rámce

MASTER zasílá

CODE			CDATA
FRM	DAT _x	ACK _x	
10	111	000	xxx

MASTER přijme od SLAVE

CODE			CDATA
FRM	DAT _x	ACK _x	
10	111	000	xxx

9.1.3.2. MASTER nežádá a SLAVE žádá potvrzení zprávy

MASTER zasílá

CODE			CDATA
FRM	DAT _x	ACK _x	
10	111	aaa	xxx

MASTER přijme od SLAVE

CODE			CDATA
FRM	DAT _x	ACK _x	
10	bbb	000	xxx

MASTER zasílá

CODE			CDATA
FRM	DAT _x	ACK _x	
10	000	ccc (bbb)	xxx

aaa - MASTER potvrzuje přijetí rámce č. aaa

bbb - SLAVE zasílá rámec č. bbb = <1,6>

ccc (=bbb) - MASTER potvrzuje přijetí rámce č. bbb

9.1.3.3. MASTER žádá a SLAVE nežádá potvrzení zprávy

MASTER zasílá

CODE			CDATA
FRM	DAT _x	ACK _x	
10	aaa	000	xxx

MASTER přijme od SLAVE

CODE			CDATA
FRM	DAT _x	ACK _x	
10	000	bbb (aaa)	xxx

aaa - MASTER zasílá rámeček č. aaa = <1,6>

bbb (= aaa) - SLAVE potvrzuje přijetí rámečku č. aaa

9.1.3.4. MASTER i SLAVE žádají potvrzení zprávy

MASTER zasílá

CODE			CDATA
FRM	DAT _x	ACK _x	
10	aaa	bbb	xxx

MASTER přijme od SLAVE

CODE			CDATA
FRM	DAT _x	ACK _x	
10	ccc	ddd (aaa)	xxx

MASTER zasílá

CODE			CDATA
FRM	DAT _x	ACK _x	
10	eee	fff (ccc)	xxx

aaa - MASTER zasílá rámeček č. aaa = <1,6>

bbb - MASTER potvrzuje přijetí rámečku č. bbb

ccc - SLAVE zasílá rámeček č. ccc = <1,6>

ddd (= aaa) - SLAVE potvrzuje přijetí rámečku č. aaa

eee - MASTER zasílá rámeček č. eee = <1,6>

fff (= ccc) - MASTER potvrzuje přijetí rámečku č. ccc

9.1.4. Zaslání zprávy formátu DF2 s číslováním a potvrzováním zprávy - linková, síťová vrstva a aplikační proces

Zaslání zprávy z procesu 9, instance 10, uzlu 5 do procesu 8, instance 7, uzlu 6, čísla zprávy ddd, příkazu zápis dat, data typu BYTE, od logické adresy 11, počtu 3, hodnoty 2, 4, 6.

MASTER zasílá
linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	6	5	23	0		?	?	003h

síťová vrstva, ChnSof

DATA			
CODE			CDATA
FRM	DAT _x	ACK _x	
10	aaa	bbb	

data aplikace, vyplní proces který zprávu zasílá

CDATA formát 2 - DF2											
2	1	4	4	1	1	1	1	2	2	3	
LEN1	DUMM	DEST	SOUR	MNO	ANO	CMD	TREC	RADDR	RCNT	RDATA	
22	ccc	8,7,6	9,10,5	ddd	eee	040h	1	11	3	2, 4, 6	

aaa - číslování rámce, provádí síťová vrstva

bbb - potvrzení přijatého rámce, provádí síťová vrstva

ccc - k libovolnému využití odesílajícího procesu. Proces, který zprávu přijal, v odpovědi DUMM zkopíruje

ddd - číslo zprávy zasílané zprávy, čísluje odesílající proces

eee - potvrzení přijaté zprávy, potvrzuje příjemce zprávy

MASTER přijme od SLAVE

v případě, že SLAVE uzel má odpověď na zprávu od procesu, který zprávu obdržel. Proces pouze potvrzuje přijetí zprávy ddd.

linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	5	6	15	0		?	?	003h

síťová vrstva, ChnSof

DATA			
CODE			CDATA
FRM	DAT _x	ACK _x	
10	fff	aaa	

data aplikace, vyplnil proces který zprávu odeslal

CDATA formát 2 - DF2						
2	1	4	4	1	1	1
LEN1	DUMMY	DEST	SOUR	MNO	ANO	CMD
14	ccc	9,10,5	8,7,6	ggg	ddd	003h

nebo

v případě, že SLAVE uzel má odpověď na zprávu od procesu, který zprávu obdržel. Proces potvrzuje přijetí zprávy ddd a navíc zasílá zprávu ggg.

linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	5	6	xx	xx		?	?	003h

síťová vrstva, ChnSof

DATA			
CODE			CDATA
FRM	DATx	ACKx	
10	fff	aaa	

data aplikace, vyplnil proces který zprávu odeslal

CDATA formát 2 - DF2										
LEN1	DUMM	DEST	SOUR	MNO	ANO	CMD	TREC	RADDR	RCNT	RDATA
xx	ccc	9,10,5	8,7,6	ggg	ddd	040h	xx	xx	xx	xx

nebo

v případě, že SLAVE nemá žádnou zprávu od k odpovědi, pak síťová vrstva pouze potvrdí příjem rámce

linková vrstva, ChnPrt

SOH	DNODE	SNODE	LEN		DATA	CRC		ETX
			LO	HI		LO	HI	
001h	5	6	1	0		?	?	003h

síťová vrstva, ChnSof

DATA		
CODE		
FRM	DATx	ACKx
10	000	fff

V případě, že proces, který zprávu přijal, hned neodpoví, je zasláno síťovou vrstvou pouze potvrzení rámce nebo zpráva od jiného procesu. Vlastní odpověď může přijít v jiném rámci při testu komunikace nebo zaslání další zprávy.