COETH01

OVLADAČ ETHERNETOVÉ KARTY IOETH01

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

Verze dokumentu 1.20

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 popsané. 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: 21.04.2004
Datum posledního uložení dokumentu: 21.04.2004
(Datum vydání a posledního uložení dokumentu musí být stejné)

Obsah :

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Čr		První vydání
1.10	1.XX	Tu	16.05.2003	Úprava dokumentu dle ISO9000
1.20	1.XX	Wil,Bin	21.04.2004	

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky pro ovládání ethernetové karty IOETH01.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba se seznámit s manuálem CoBase.

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

3.1. Účel knihovny CoEth01

Knihovna **CoEth01** slouží k ovládání síťové karty IOETH01 (resp. ethernetového řadiče LAN91C96 připojeného na sběrnici IOBUS). Knihovna ke své funkci nevyžaduje zapojení přerušovacích signálů na kartě IOETH01.

Ovladač síťové karty vychází ze třídy TCoDevice detailně popsané v dokumentaci ke knihovně CoBase. V této dokumentaci naleznete pouze jen odlišnosti a speciality týkající se implementace ovladače karty IOETH01.

Knihovna CoEth01 využívá knihovnu CoEthDef, což je knihovna společných definicí pro veškeré knihovny zajišťující přístup k médiu typu Ethernet. Některé zde uváděné konstanty jsou nadefinovány v knihovně CoEthDef, na což bude v příslušném místě upozorněno.

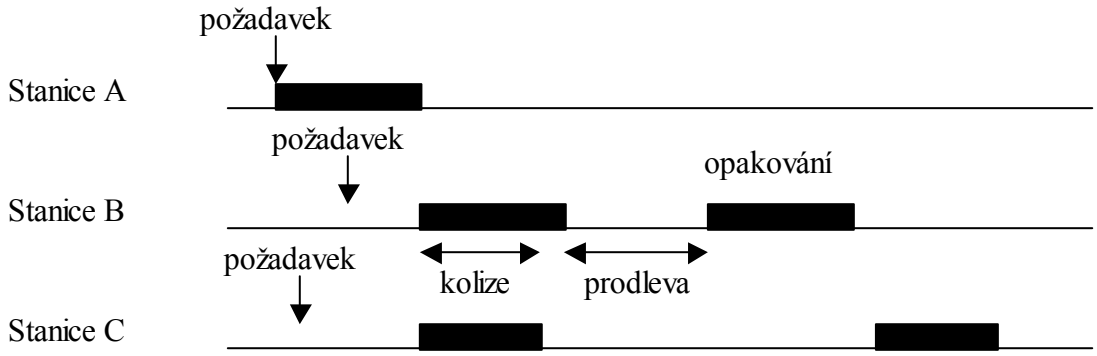
Z následujícího obrázku je zřejmé umístění vrstvy rozhraní sítě v rámci TCP/IP architektury. Stručný popis architektury TCP/IP je uveden v dokumentaci ke knihovně CoBase.

3.2. CSMA/CD

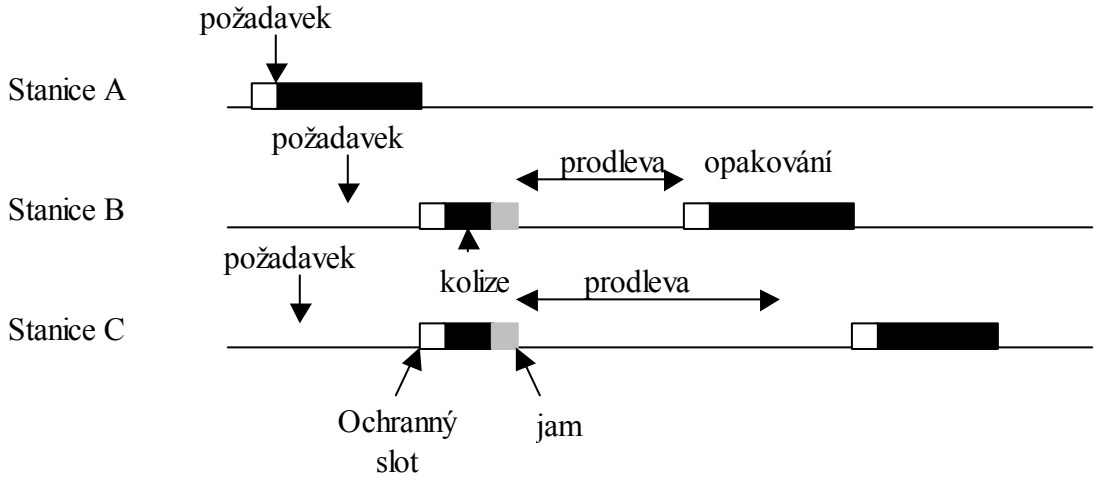
K řízení přístupu na sběrnici Ethernet se používá metoda CSMA/CD (Carries Sense Multiple Access/ Collision Detection).

Metody CSMA jsou založeny na schopnosti stanic rozpoznat, zda je **kanál obsazený nebo volný** obvykle pomocí sledování nosné. Existuje několik variant této metody. V případě nejjednodušší z nich stanice testuje před vysíláním stav kanálu. Je-li kanál obsazen, stanice vysílání odloží na okamžik, kdy se kanál uvolní. Protože stanice nedokáží rozpoznat kolizi, spolehlivé doručení rámce musí zajišťovat vyšší protokolová vrstva např. opakováním rámce po nedoručení potvrzení od protější

stanice. Z tohoto důvodu vykazují tyto metody relativně nízkou propustnost kanálu a vysoké zpoždění a tedy i nízkou efektivitu.



Výrazné zlepšení lze dosáhnout pomocí metody CSMA/CD, která je založena na **detekci kolize** dvou nebo více vysílajících stanic. Tato metoda vyžaduje přenosové médium, které detekci kolize umožňuje (tj. např. sběrnice typu otevřený kolektor, speciální vodič vedený paralelně se sběrnicí nebo sledování napětí na proudově buzené sběrnici).



Stanice, která má připravený rámec k vysílání a detekuje klid na sdíleném kanále po definovanou dobu označovanou jako kolizní slot, zahájí vysílání synchronizační posloupnosti a odešle vlastní rámec. Stanice, která chce vysílat, ale indikuje provoz na sdíleném médiu, musí počkat na uvolnění média a uplynutí ochranného intervalu (kolizního slotu). Teprve potom může zahájit vysílání.

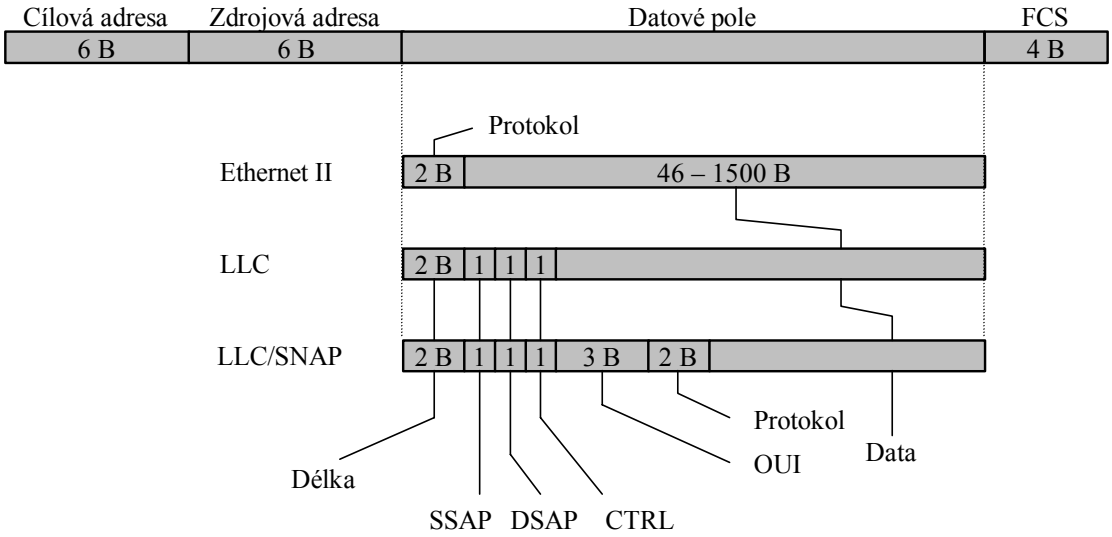
Pokud stanice začne vysílat a vstoupí do kolize (a tuto skutečnost rozpozná), přeruší vysílání rámce, ale ještě před uvolněním média odešle kolizní posloupnost (jam). O opakované vysílání se stanice pokusí po určité, náhodně zvolené době. Náhodná volba pomlky brání periodickému opakování kolize.

V případě sítě Ethernet: Pokud k opakované kolizi dojde, stanice prodlužuje střední doby opakovaní prodlevy na dvojnásobek. Po deseti neúspěšných pokusech přestane prodlevu prodlužovat a po šestnácti hlásí závadu vyšším vrstvám obsluhy. Tento postup je označován jako exponenciální ustupování (Exponential Back-off) a je navržen tak, aby zajistil stabilitu sítě pro alespoň 1024 stanic.

3.3. Formát ethernetového rámce

Formát rámců v lokální síti Ethernet je naznačen na následujícím obrázku. Rámec předchází ještě tzv. preambule, což je posloupnost osmi oktetů (jeden oktet je osm bitů), tj. 64 jedniček a nul. Preambule slouží k synchronizaci přijímače.

Formát rámce vrstvy rozhraní sítě



Cílová adresa a **zdrojová adresa** mají délku 48 bitů. Zdrojová adresa je vždy individuální a cílová může být skupinová (multicast – rámec je určen pro určitou skupinu stanic) nebo všeobecná (broadcast – rámec je určen pro všechny stanice v lokální síti). Rámec je zabezpečen 32 bitovým cyklickým kódem v poli **FCS** (Frame Check Sequence). **Datové pole** je dlouhé minimálně 48 oktetů, maximálně však 1502 oktetů. Maximální velikost datového pole se obvykle nazývá MTU (Maximum Transmittion Unit). Dekódování těchto položek realizuje zčásti již síťová karta (přesněji Ethernetový řadič).

Na spojové vrstvě (v terminologii OSI) jsou definovány další položky. V praxi se velmi často vyskytuje sdílení různých spojových protokolů na stejné lokální sítí. Z tohoto důvodu je potřeba rámce různých protokolů rozlišit dalšími položkami. Pro tyto účely se v současné době se používají tři formáty rámce na spojové vrstvě:

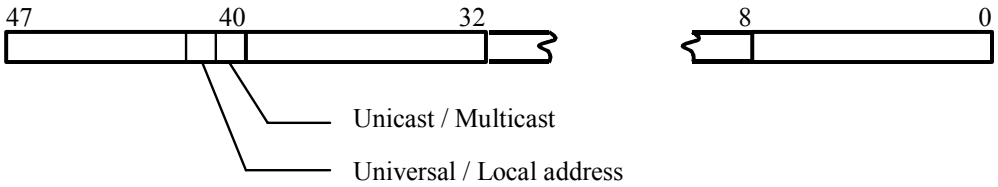
- Ethernet II
- LLC
- LLC/SNAP

Rámec typu **Eternet II** obsahuje kromě cílové a zdrojové adresy navíc 2 oktety specifikující síťový protokol. Všechny identifikátory síťových protokolů, tzv. ETHER TYPES, jsou uvedeny v RFC-1700 na str.168. Např. IP protokol používá identifikátor 800h, protokol IPX 8137h, atd. Zbytek rámce za identifikátorem tvoří datagram mezisíťové vrstvy.
Dalším formát rámce - **LLC** (Logical Link Control) definuje IEEE 802.3. První položkou LLC rámce je jeho celková délka. Následují položky SSAP (Source Service Access Point) a DSAP (Destination Service Access Point) označují zdrojovou, resp.

cílovou aplikaci, která rámec zpracovává. Poslední pole CTRL specifikuje typ odesílaného rámce. Rezervované hodnoty polí SSAP, DSAP, CTRL jsou uvedeny v RFC-1700.
Protože formát rámce LLC nedává dostatečný adresový prostor všem existujícím síťovým protokolům, byl vytvořen alternativní protokol **SNAP** (Subnetwork Access Protocol), který rozšiřuje záhlaví LLC o položky OUI (Identifikátor organizace), a identifikátor protokolu, který je shodný s identifikátorem protokolu v rámci Ethernet II. Při použití SNAP rámce mají položky SSAP, DSAP hodnotu 170 a položka CTRL hodnotu 3. Více informací lze nalézt v RFC-1042.
Všechny tři typy rámců lze současně provozovat na jedné lokální síti. Aby se rozlišil formát rámce Ethernet II od rámce LLC, má identifikátor protokolu v rámci formátu Ethernet II číselnou hodnotu vždy větší, než je maximální možná délka rámce v síti Ethernet. Identifikátory začínají hodnotou 800h.

Stanice, která chce po síti předat blok dat, jej opatří fyzickou adresou příjemce a svou vlastní fyzickou adresou, tzv. MAC (Media Access Control) adresu. MAC adresa je 48-bitový identifikátor. Každý síťový adaptér na stejné lokální síti musí mít přidělenou jednoznačnou MAC adresu. Formát této adresy je vyznačen na obrázku níže:

Formát MAC adresy

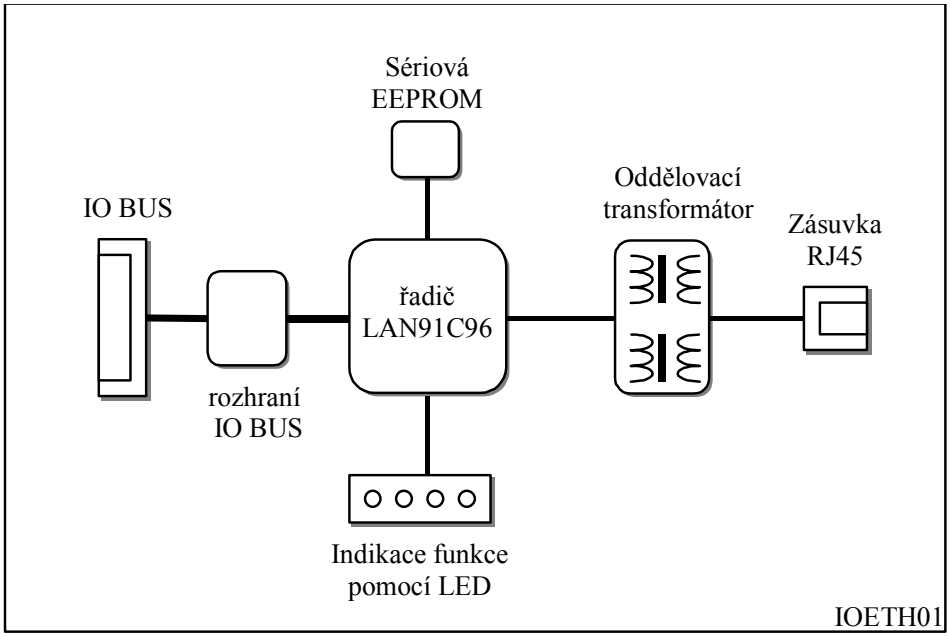


Dva nejnižší bity prvního oktetu adresy mají speciální význam. První bit určuje, zda se jedná o adresu konkrétní stanice (hodnota 0), nebo skupinovou, resp. všeobecnou adresu (multicast resp. broadcast). Všeobecná adresa obsahuje ve všech bitech hodnotu 1.
Druhý bit určuje zda se jedná o adresu přidělenou výrobcem, tj. teoreticky celosvětově unikátní adresu (hodnota 0), nebo o *lokální* adresu, přidělenou obvykle správcem sítě. MAC adresy, které mají v tomto bitu 0 jsou spravovány mezinárodní organizací IEEE (viz. RFC-1700 str. 172].

3.4. Síťová karta IOETH01

Karta IOETH01 je rozšiřující modul, který umožňuje připojení řídícího systému KIT na síť Ethernet. Připojuje se standardním způsobem na sběrnici IOBUS. Karta je osazena řadičem ethernetu LAN91C96, který je připojen prostřednictvím 8-bitové sběrnice. Na čipu je integrováno 6kB vyrovnávací paměti, která je dynamicky přidělována podle velikosti vysílaných nebo přijímaných rámců

Na následujícím obrázku je naznačeno blokové schéma síťové karty IOETH01.



Základním stavebním prvkem síťové karty je **ethernetový řadič**. Ethernetový řadič implementuje funkce fyzické vrstvy potřebné pro příjem a vysílání rámců po sběrnici Ethernet. Obvykle také obsahuje vyrovnávací paměť RAM pro pokrytí určité latence systému.
Oddělovací transformátor zajišťuje přizpůsobení signálu sběrnice Ethernet a galvanické oddělení sběrnice od ostatních částí systému.
Obvykle má Ethernetový řadič jeden nebo více výstupů pro indikace různých funkcí pomocí **LED** (např. příjem, vysílání nebo připojení ke sběrnici Ethernet).
Rozhraní IOBUS zajišťuje transformaci signálů řadiče na signály sběrnice řídícího systému, oddělení řadiče a posílení výstupních signálů.
Sériová EEPROM, která je nezbytnou součástí každé síťové karty obsahuje adresu řadiče (MAC adresu) a příp. další nastavení řadiče.
Zásuvka RJ-45 zajišťuje standardní připojení síťové karty do sběrnice Ethernet např. pomocí kabelu UTP.

MAC adresa karty je uložena v sériové paměti EEPROM. MAC adresa je nastavena z výroby a je odvozena ze sériového čísla desky.

Na desce jsou čtyři indikační led diody, určené k signalizaci provozních stavů. LED signalizují:

- příjem rámců
- vysílání rámců
- připojení kabelu
- přístup ke kartě ze sběrnice IOBUS.

Pro správnou funkci karty je potřeba správně nastavit konfigurační propojky pro určení bázové adresy v I/O prostoru. Knihovna CoEth01 nevyžaduje pro svou funkci přerušovací signály vyvedené z řadiče.

3.4.1. Obsah paměti EEPROM

V paměti EEPROM je uloženo implicitního nastavení řadiče. Po resetu je část paměti EEPROM řadičem automaticky překopírována do vnitřních registrů pro bázovou adresu, konfiguraci řadiče a MAC adresu.

Každá ethernetová karta v lokální síti musí mít vlastní jednoznačnou a unikátní MAC adresu. Aby nedocházelo ke kolizím, jsou všichni velcí výrobci ethernetových karet registrováni u mezinárodní organizace IEEE a mají přidělen určitý rozsah MAC adres.

MAC adresa je u každé vyrobené karty automaticky nastavena v rámci procesu testování podle následujícího klíče:

02-MM-TT-TT-SS-SS

> **MM** je „magické“ číslo, stejné pro všechny karty tohoto typu (např. 3Ch)
> **TT** je hodnota odvozená z aktuální času a náhodného čísla
> **SS** je sériové číslo karty (1 až 65535)

První bajt adresy má nastavený 2.bit na jedničku. To znamená, že se nejedná o globálně registrovanou MAC adresu, ale o lokálně zvolenou adresu, a proto není potřeba tyto adresy registrovat u IEEE.

4. Konstanty a jednoduché typy

4.1. Konstanty

4.1.1. Návratové kódy CST_ETH01_xxx

Knihovna CoEth01 definuje několik nových návratových kódu s prefixem CST_ETH01_:

CST_ETH01_ERR_IOBASE
Neplatná bázová adresa karty IOETH01 v I/O prostoru. Tato chyba může nastat při volání metody **CoBind**. Při výskytu této chyby zkontrolujte, zda je karta IOETH01 připojena a zda má nastavení propojek souhlasí se zadanou bázovou adresou.

CST_ETH01_ERR_TXUNRN
Podtečení vysílací fronty LAN91C96. Tento chybový kód může vrátit metoda **CoSendBuffer** nebo **CoSendBufferTo**. Chyba tohoto typu by však se současným software neměla nikdy nastat.

CST_ETH01_ERR_TXLINK
Chyba detekce linkového signálu na médiu při vysílání. Tento chybový kód může vrátit metoda **CoSendBuffer** nebo **CoSendBufferTo**. Chyba může být způsobena rušením nebo závadou na sběrnici.

CST_ETH01_ERR_TXLATECOL
Pozdní kolize při vysílání. Tento chybový kód může vrátit metoda **CoSendBuffer** nebo **CoSendBufferTo**, pokud vysílač detekuje kolizi po více než 64. bajtu vysílaného rámce. Chyba může být způsobena rušením nebo závadou na sběrnici.

CST_ETH01_ERR_TX16COL
Při vysílání bylo dosaženo 16 kolizí. Rámec nebyl odvysílán. Tento chybový kód může vrátit metoda **CoSendBuffer** nebo **CoSendBufferTo**, v případě poruchy nebo extrémním zatížení sběrnice.

CST_ETH01_ERR_TX
Nespecifikovaná chyba při vysílání. Tento chybový kód může vrátit metoda **CoSendBuffer** nebo **CoSendBufferTo**.

CST_ETH01_ERR_HARDWARE
Chyba hardware desky nebo samotného řídícího systému. Tato chyba může vzniknout i v případě náhlého rušení. Pokud ovladač síťové karty detekuje takový stav, provede reinicializaci řadiče.

CST_ETH01_ERR_NOEEPROM
Chyba detekce sériové paměti EEPROM. V případě výskytu této chyby je tato paměť EEPROM buď odpojena (propojka na desce IOETH01) nebo poškozena.

Tyto návratový kódy je možné přeložit pomocí metody **CoStatusToStr** na textový řetězec.

4.1.2. Identifikátory tříd zařízení DEV_CLASS_xxx

Třída TCoEth01 (ovladač síťové karty) má přidělen identifikátor **DEV_CLASS_NIC**. Tento identifikátor určuje zařízení v řetězci při volání funkcí **CoIoctl**, **CoGetOption**, **CoSetOption** apod. Zaregistrované jméno třídy pro použití s funkcí **CoCreateDevice** je **ETH01**.

4.1.3. Identifikátory řídících operací IOCTL_ETH01_xxx

Identifikátory s prefixem **IOCTL_** specifikují operaci prováděnou funkcí **CoIoctl** v řetězci zařízení. Knihovna CoEth01 definuje několik nových konstant s prefixem IOCTL_ETH01_.

IOCTL_NIC_GETMACADDRESS
Čtení aktuální adresy síťového adaptéru. Je vrácena adresa z paměti řadiče, nikoli adresa z paměti EEPROM. Po inicializaci zařízení metodou **CoBind** je vždy do paměti řadiče nastavena adresa z paměti EEPROM. Deklarace identifikátoru IOCTL_NIC_GETMACADDRESS je uvedena v knihovně CoEthDef.

Formát dat: TEthMacAddress

IOCTL_ETH01_SETMACADDRESS
Nastavení MAC adresy síťového adaptéru. Tímto příkazem je nastavena pouze adresa v řadiči a po resetu nebo výpadku napájení je adresa obnovena z paměti EEPROM.

Formát dat: TEthMacAddress

IOCTL_ETH01_STOREMACADDRESS
Uložení aktuální MAC adresy síťového adaptéru do EEPROM. Tento příkaz se používá k trvalé změně adresy síťového adaptéru.

Formát dat: žádná data se nepředávají

IOCTL_ETH01_RELOADMACADDRESS
Obnovení MAC adresy síťového adaptéru z EEPROM.

Formát dat: žádná data se nepředávají

IOCTL_ETH01_RESETCOUNTERS

Nulování čítačů událostí. Použijte tento příkaz, pokud chcete vynulovat hodnoty všech čítačů události.

Formát dat: žádná data se nepředávají

IOCTL_ETH01_GETCOUNTERS

Čtení stavu čítačů událostí. Po provedení příkazu je naplněna předaná struktura stavem všech čítačů událostí. viz. kapitola 4.2.4

Formát dat: TEthCounters

4.1.4. Identifikátory parametrů zařízení COPT_ETH01_xxx

Identifikátory s prefixem **COPT_** specifikují parametr zařízení ve funkci **CoGetOption** příp. **CoSetOption**. Knihovna CoEth01 definuje několik nových konstant s prefixem COPT_ETH01_:

COPT_ETH01_IOBASE

Bázová adresa karty IOETH01 v I/O prostoru. Pokud není bázová adresa nastavena správně, metoda **CoBind** vrátí chybový kód. Tento parametr lze měnit pouze v neaktivním stavu zařízení.

Formát dat: Word Implicitní hodnota: 300h

COPT_ETH01_TXPAGES

Počet rezervovaných paměťových stránek (256B) ve vnitřní paměti ethernetového řadiče pro vysílání. Ethernetový řadič LAN91C96 má celkem 6kB vnitřní vyrovnávací paměti. Část této paměti může být vyhrazena pro vysílání, což zajistí možnost kdykoli odeslat jeden nebo více rámců o minimální zadané velikosti, přestože např. jiné stanice naprosto zahlcují přijímač řadiče. Implicitní hodnota je 6, což postačuje na odeslání jednoho ethernetového rámce o maximální délce (1536B).

Formát dat: Word Implicitní hodnota: 6 (min: 0, max: 24)

COPT_ETH01_PRMS

Povolení promiskuitního módu řadiče. V tomto módu řadič přijímá všechny rámce bez ohledu na cílovou adresu stanice. Tento mód slouží spíše pro testovací účely a běžně je vypnutý.

Formát dat: Boolean Implicitní hodnota: False (vypnuto)

COPT_ETH01_MONCSN

Povolení sledování nosné během vysílání. Pokud je sledování zapnuto, ethernetový řadič během vysílání sleduje svou vlastní nosnou. Pokud je signál ztracen, vysílání se ukončí a metoda **CoSendBuffer** nebo **CoSendBufferTo** vrátí chybový kód CST_ETH01_TX a dojde ke zvýšení čítače ETHCTR_TX_LOSTCARRIER.

Formát dat: Boolean Implicitní hodnota: True (zapnuto)

COPT_ETH01_SQET
Povolení tzv. SQET (Signal quality error test). Řadič testuje klid na lince po dobu několika mikrosekund po ukončení vysílání. Pokud zjistí závadu, pak metody **CoSendBuffer** nebo **CoSendBufferTo** vrátí chybový kód.

Formát dat: Boolean Implicitní hodnota: False (vypnuto)

COPT_ETH01_PADFRAMES
Povolení zarovnávání datových rámců na minimální délku (64 bajtů). Tento přepínač lze např. vypnout v případě, že provozujeme řadič na dvoubodovém spoji v plně duplexní režimu.

Formát dat: Boolean Implicitní hodnota: True (zapnuto)

COPT_ETH01_MODE
Mód ethernetového řadiče. viz. konstanty ETH_MODE_ v kapitole 4.1.5

Formát dat: Word Implicitní hodnota: ETH_MODE_CSMACD

4.1.5. Identifikátory módů řadiče ETH_MODE_xxx

Konstanty ETH_MODE_ jsou určeny k nastavení režimu práce ethernetového řadiče. Použijí se společně s přepínačem **COPT_ETH01_MODE** a metodou **CoSetOption** nebo **CoGetOption**.

ETH_MODE_CSMACD
Standardní provoz řadiče s zapnutým naslouchánim nosné a detektorem kolizi. Tento režim je určen pro práci na sdíleném médiu.

ETH_MODE_DRIVER
Nedokumentováno.

ETH_MODE_FDSE
Plně duplexní přepínaný ethernet (Full Duplex Switched Ethernet). Řadič má v tomto režimu odpojený detektor kolizí a naslouchání nosné a může vysílat a přijímat zároveň. Tento režim je určen k efektivnějšímu propojení dvou stanic, nelze jej použít na sdíleném médiu.

ETH_MODE_LOOPBACK
Vysílané rámce jsou interně v řadiči posílány zpět do přijímače. Na vnějším rozhraní se nevysílá ani nepřijímá. Tento režim je určen pro testování funkčnosti řadiče.

4.1.6. Identifikátory síťových protokolů ETH_TYPE_xxx

ETH_TYPE_IP	= $0800	IP protokol verze 4
ETH_TYPE_ARP	= $0806	ARP protokol
ETH_TYPE_RARP	= $8035	RARP protokol
ETH_TYPE_IPX	= $8137	Novell IPX/SPX protokol

Konstanty ETH_TYPE_ identifikují protokol síťové (mezisíťové) vrstvy (tj. položka Protocol struktury **TEthAddress**). Uvedený výčet není samozřejmě úplný, kompletní seznam čísel všech registrovaných protokolů lze nalézt v RFC1700, str. 168 – ETHER TYPES. Konstanty ETH_TYPE_ jsou deklarovány v knihovně CoEthDef.

4.1.7. Identifikátory formátu rámce ETH_FORMAT_xxx

ETH_FORMAT_ETHII	Formát Ethernet II
ETH_FORMAT_LLC	Formát IEEE 802.3, LLC
ETH_FORMAT_SNAP	Formát LLC/SNAP

Tyto konstanty identifikují formát rámce na spojové vrstvě (viz. kapitola 3.3), konstanty lze použít při určení formátu rámce odchozího nebo příchozího rámce (viz. položka LinkProto struktury **TEthAddress**) Konstant ETH_FORMAT_ jsou deklarovány v knihovně CoEthDef.

4.2. Typy

4.2.1. Třída TCoEth01

```
TCoEth01 = object( TCoDevice );
```

Pokud použijeme knihovnu CoEth01 (uvedeme ji někde v seznamu za klíčovým slovem **uses**), bude tato třída zaregistrovaná v globálním seznamu zařízení pod jménem **ETH01**. Toto jméno lze použít v případě vytváření třídy funkcí **CoCreateDevice**. Číselný identifikátor třídy je **DEV_CLASS_NIC** (Network Interface Card).

4.2.2. Struktura TEthMacAddress

```
PEthMacAddress = ^TEthMacAddress;
TEthMacAddress = packed record
  case Integer of
    0: (B0, B1, B2, B3, B4, B5: Byte);
    1: (Bytes: array[0..5] of Byte);
    2: (Words: array[0..2] of Word);
end;
```

Struktura **TEthMacAddress** reprezentuje MAC adresu síťového adaptéru. Je definována v knihovně CoEthDef.

4.2.3. Struktura TEthAddress

```
TEthAddress = record
  Size      : Byte;
  Mac       : TEthMacAddress;        { MAC adresa }
  Format    : Byte;                  { ETH_FORMAT_xxx }
  case Integer of
    0 : ( Protocol  : Word );
    1 : ( SSAP: Byte; DSAP: Byte );
end;
```

Struktura **TEthAddress** popisuje adresu na spojové vrstvě, která se použije při volání aplikačních metod třídy TCoEth01 (např. CoSendBufferTo, CoRecvBufferFrom, CoBind a CoConnect). Je definována v knihovně CoEthDef.

Položka **Size** udává velikost struktury TEthAddress. Před použitím struktury je nutné se vždy ujistit, zda položka **Size** obsahuje správnou hodnotu, tedy SizeOf(TethAddress). V položce **Mac** je obsažena a MAC adresa stanice. Položka **Format** identifikuje použitý formát rámce a může obsahovat jednu z konstant ETH_FORMAT_ z kapitoly 4.1.7. V případě použití konstant ETH_FORMAT_ETHII nebo ETH_FORMAT_SNAP, musí být vyplněna položka **Protocol**. V případě ETH_FORMAT_LLC, musí být vyplněny obě položky **SSAP** a **DSAP**.

4.2.4. Struktura TEthCounters

```
TEthCounterType = (
  ETHCTR_TX_FRAMES,       { Frames sent }
  ETHCTR_RX_FRAMES,       { Frames received  }
  ETHCTR_TX_SINGLECOL,    { Single collision       }
  ETHCTR_TX_MULTIPLECOL, { Mutliple collision     }
  ETHCTR_TX_DEFFERAL,     { Defferal               }
  ETHCTR_TX_EXCDEFFERAL, { Excessive Defferal     }
  ETHCTR_TX_LOSTCARRIER, { Lost Carrier           }
  ETHCTR_TX_SQET,         { Signal Quality Err     }
  ETHCTR_TX_LINKERROR,    { Link Error             }
  ETHCTR_TX_UNDERRUN,     { FIFO Underrun          }
  ETHCTR_TX_LATECOL,      { Late Collision         }
  ETHCTR_TX_16COL,        { 16 Collisions          }
  ETHCTR_RX_TOOSHORT,     { Frame too short        }
  ETHCTR_RX_TOOLONG,      { Frame too long         }
  ETHCTR_RX_OVERRUN,      { RX Overrun             }
  ETHCTR_RX_BADCRC,       { RX Bad CRC             }
  ETHCTR_RX_INVHEADER,    { RX Invalid Header      }
  ETHCTR_RX_DISCARDS      { RX Discards            }
ETHCTR_REINITIALIZE
);

  TEthCounters = array[TEthCounterType] of Longint;
```

Tato struktura slouží k předání aktuálních hodnot čítačů událostí. Čítače událostí jsou 32 bitové proměnné, které se zvyšují o 1 pokaždé, co nastane při přijmu nebo vysílání určitá událost.

ETHCTR_TX_FRAMES

Zvyšuje se s každým úspěšně odeslaným rámcem.

ETHCTR_RX_FRAMES

Zvyšuje se s každým úspěšně přijatým rámcem.

ETHCTR_TX_SINGLECOL

Počet jednoduchých kolizí na sběrnici při vysílání. Jednoduchá kolize je detekována tehdy, když vysílač musí odložit vysílání na pozdější dobu v důsledku kolize na sdílené sběrnici, a při dalším pokusu o vysílání ke kolizi nedojde. Jednoduché kolize nejsou na závadu. Jejich počet svědčí pouze o jistém zatížení sběrnice.

ETHCTR_TX_MULTIPLECOL

Počet násobných kolizí na sběrnici při vysílání. Násobná kolize je detekována tehdy, když vysílač musí opakovaně (alespoň 2-krát po sobě) odložit vysílání stejného rámce v důsledku kolize na sdílené sběrnici.

ETHCTR_TX_DEFFERAL

Počet vysílání rámcu odložených na pozdější dobu, v důsledku vysílání jiné stanice. Pokud stanice na sběrnici vysílají nezávisle na sobě, je zvyšování tohoto čítače celkem běžné.

ETHCTR_TX_EXCDEFFERAL

Vysílání rámce bylo odložen na dobu větší, než čas potřebný pro odeslání 2 * 1518 B dat. Taková situace by se neměla na sběrnici vyskytovat. Pokud ano, svědčí to o nějaké závadě (pravděpodobně některé ze stanic).

ETHCTR_TX_LOSTCARRIER

Ztráta nosné na začátku vysílání. Tento čítač se zvyšuje jen tehdy pokud je zapnuta kontrola sledování nosné. Viz. přepínač COPT_ETH01_MONCSN v kapitole 4.1.4

ETHCTR_TX_SQET

Počet zjištěných chyb testu SQET. Aby se tento čítač zvyšoval, musí být testování SQET povolené. Viz. COPT_ETH01_SQET v kapitole 4.1.4

ETHCTR_TX_LINKERROR

Počet chyb nosné při vysílání. Zvýšený výskyt těchto chyb může svědčit o významném rušení nebo poruše sběrnice.

ETHCTR_TX_UNDERRUN

Podtečení interní FIFO v řadiči při vysílání. Tento čítač by měl při současném hardware zůstat nulový.

ETHCTR_TX_LATECOL

Počet pozdních kolizí při vysílání. Tento čítač je zvýšen pokaždé když řadič při vysílání detekuje kolizi po více než 64. bajtu vysílaného rámce. Častý výskyt svědčí o závadě na sběrnici (např. překročena maximální povolená délka segmentu apod.)

ETHCTR_TX_16COL

Počet rámců, které nebyly odvysílány v důsledku toho, že při jejich vysílání bylo dosaženo 16 kolizí. Pokud se tato chyba vyskytuje častěji, svědčí to o extrémním zatížení sběrnice.

ETHCTR_RX_TOOSHORT

Délka přijatého rámce je menší než 64 B.

ETHCTR_RX_TOOLONG

Počet přijatých rámců delších než 1518 B. Výskyt těchto chyb svědčí o špatné funkci některé ze stanic na síti.

ETHCTR_RX_OVERRUN,

Tento čítač je zvýšen pokaždé, kdy je zjištěno přetečení přijímací vyrovnávací paměti řadiče. Hodnota nemusí (a s nejvyšší pravděpodobností nebude) odpovídat počtu ztracených paketů, ale bude o něco menší. Velká hodnota tohoto čítače svědčí o tom, že stanice nestačí dostatečně rychle zpracovávat příchozí pakety.

ETHCTR_RX_BADCRC

Počet přijatých rámcu s neplatným CRC. Tento čítač se zvyšuje jen v případě, že je knihovna CoEth01 přeložena bez symbolu ETH_DISCARD_BAD_CRC. Tato funkce je k dispozici jen na vyžádání.

ETHCTR_RX_INVHEADER

Počet zahozených přijatých paketů kvůli neplatnému formátu nebo chybě v záhlaví rámce.

ETHCTR_RX_DISCARDS

Počet zahozených přijatých paketů, z jiných důvodů (např. vyšší protokolová vrstva rámec nepřijala.)

ETHCTR_REINITIALIZE

Počet reinicializací ovladače ethernetové karty.

Textový konfigurační řetězec

Parametry zařízení TCoEth01 lze nastavovat pomocí textového konfiguračního řetězce metodami **CoSetOptionString**.

V následujícím seznamu jsou uvedeny všechny povolené identifikátory parametrů:

IOBASE Bázová adresa ethernetové karty v I/O prostoru (viz. také COPT_ETH01_IOBASE). Tento parametr lze měnit pouze v případě, že zařízení je v neaktivním stavu.

Formát dat:	číslo
Rozsah hodnot:	$0000-$FFFF
Implicitní nastavení:	$0300

PRMS Povolení promiskuitního režimu řadiče, tj. režimu, kdy přijímá všechny rámce bez ohledu na cílovou adresu v příchozím rámci (viz. také COPT_ETH_PRMS).

Formát dat:	číslo
Rozsah hodnot:	0 (ne), 1 (ano)
Implicitní nastavení:	0

TXPAGES Počet rezervovaných paměťových stránek (256B) ve vnitřní paměti ethernetového řadiče pro vysílání. (viz. také COPT_TX_PAGES)

Formát dat: číslo
Rozsah hodnot: 0-24
Implicitní nastavení: 6

MONCSN Povolení sledování nosné během vysílání. Pokud je sledování zapnuto, ethernetový řadič během vysílání sleduje svou vlastní nosnou. Pokud je signál ztracen, vysílání se ukončí a metoda **CoSendBuffer** nebo **CoSendBufferTo** vrátí chybový kód CST_ETH01_TX a dojde ke zvýšení čítače ETHCTR_TX_LOSTCARRIER. (viz. také COPT_ETH01_MONCSN)

Formát dat: číslo
Rozsah hodnot: 0 (ne), 1 (ano)
Implicitní nastavení: 1

SQET Povolení tzv. SQET (Signal quality error test). Řadič testuje klid na lince po dobu několika mikrosekund po ukončení vysílání. Pokud zjistí závadu, pak metody **CoSendBuffer** nebo **CoSendBufferTo** vrátí chybový kód. (viz. také COPT_ETH01_SQET)

Formát dat: číslo
Rozsah hodnot: 0 (ne), 1 (ano)
Implicitní nastavení: 0

PADFRAMES Povolení zarovnávání datových rámcu na minimální délku (64 bajtů). Tento přepínač lze např. vypnout v případě, že provozujeme řadič na dvoubodovém spoji v plně duplexní režimu. (viz. také COPT_ETH01_PADFRAMES)

Formát dat: číslo
Rozsah hodnot: 0 (ne), 1 (ano)
Implicitní nastavení: 1

MODE Režim práce ethernetového řadiče (viz.také COPT_ETH01_MODE a konstanty ETH_MODE_xxx)

Formát dat: symbol
Povolené hodnoty: CSMACD, FDSE, DRIVER, LOOPBACK
Implicitní nastavení: CSMACD

Příklad:

```
var
  Dev : PCoDevice;

Dev^.CoSetOptionString( DEV_CLASS_NIC,  'IOBASE=$310 TXPAGES=12' );
```

5. Poznámky

5.1. Inicializace zařízení pomocí CoBind

Metoda **CoBind** provede inicializaci instance zařízení ETH01 a provede inicializaci ethernetového řadiče.

První parametr metody **CoBind**, tj. AAddress je ukazatel na lokální adresu (strukturu TEthAddress). Pokud adresu nespecifikujeme, tj. uvedeme zde konstantu **nil**, bude MAC adresa síťové karty převzata z EEPROM na desce. V opačném případě bude lokální MAC adresa nastavena podle položky MAC struktury TEthAddress. Ostatní položky (Tj. položky Format, Protocol, SSAP, DSAP) struktury TEthAddress je nutné nastavit pouze v případě, že bychom chtěli později „navázat spojení“ s jinou stanicí. Tyto položky budou sloužit k filtrování příchozích rámců.

Příklad:

```
{ Inicializace zařízení a nastavení implicitní adresy z EEPROM }
Status := Sock^.CoBind( nil, 0, nil );


{ Inicializace zařízení a explicitní nastavení adresy }
const
  LocalMacAddress : TEthAddress = (
    Size:  SizeOf( TEthAddress );
    Mac: Bytes: ( $02, $00, $00, $00, $00, $01 );
   { Na ostatních položkách nezáleží }
  );

Status := Sock^.CoBind( @LocalMacAddress, 0, nil );
```

5.2. Navazování spojení pomocí CoConnect

Na úrovni spojových protokolů, které implementuje tato knihovna jako jednoduchou nadstavbu nad ovladačem karty IOETH01 není možné navazovat spojení. Přesto zařízení TCoEth01 implementuje metodu **CoConnect**. Metoda **CoConnect** neprovádí navazování spojení, pouze si uloží zadanou adresu protistanice. Uložená adresa protistanice se použije na filtrování příchozích datagramů (jsou přijaty pouze datagramy od stanice, se kterou je „navázáno spojení“.). Po zavolání metody **CoConnect** je možné volat metody **CoSendBuffer** bez určení adresy cílové stanice a **CoRecvBuffer**, kdy adresa zdrojové stanice je jednoznačná.

První parametr metody **CoConnect** je ukazatel na adresu protistanice (strukturu TEthAddress). Struktura TEthAddress musí mít před voláním metody **CoConnect** vyplněné všechny položky. *Tato verze knihovny neumožňuje volání metody CoConnect s nespecifikovanou adresou, tedy s hodnotou* ***nil*** *místo ukazatele na adresu.*

5.3. Posílání a příjem dat

K posílání a příjmu dat jsou určené metody **CoSendBufferTo** a **CoRecvBufferFrom** (příp. **CoSendBuffer** a **CoRecvBuffer** v případě „navázaného spojení“). K předávání adresy cílové příp. zdrojové adresy je určena struktura TEthAddress. V případě metod pro odesílání a příjem dat jsou využity všechny položky struktury TEthAddress.

Karta IOETH01 umožnuje posílat data na všeobecnou adresu (broadcast). Všeobecná adresa je FF:FF:FF:FF:FF:FF.

6. Příklady

Následující příklady ukazují jak přistupovat přímo k zařízení ethernetové karty. Přímé ovládání ethernetové karty tímto způsobem je vhodné použít pouze ve speciálních případech (např. monitorování lokální sítě na nejnižší úrovni, rychlá komunikace mezi dvěmi jednotkami mimo lokální síť, apod.)

6.1. Vytvoření zařízení síťové karty

```
uses
  CoBase,
  CoEthDefs,
  CoEth01;

var
  Status : TCoStatus;  { Návratový kód volaných funkcí }
  Dev    : PCoDevice;  { Ukazatel na instanci zařízení }
  ..
  ..

begin
  ..
  ..
  Status := CoCreateDevice( 'ETH01', '', 'IOBASE=$310', nil, Dev );

  if Status <> CST_SUCCESS then
  begin
    { Při vytváření instance zařízení nastala chyba }
    WriteLn( 'CoCreateDevice() failed: ', CoStatusToStr( Status ) );
    Exit;
  end;

  { V tomto okamžiku máme v proměnně Dev uložen ukazatel na instanci
     zařízení ethernetové karty }
  ..
  ..
```

6.2. Inicializace síťové karty

```
  { Inicializace zařízení, bude použita adresa z paměti EEPROM }
  Status := Dev^.CoBind( nil, 0, nil );
```

nebo

```
const

  MyAddr : TEthAddress =
           ( Size: SizeOf( TEthAddress );
             Mac:  Bytes: ($00, $02, $00, $00, $00, $01)
             { Ostatní položky jsou pro CoBind nevýznamné }
           );

  ..
  ..
  { Inicializace zařízení, adresa síťové karty je uvedená výše }
  Status := Dev^.CoBind( @MyAddr, 0, nil );
```

```
  if Status <> CST_SUCCESS then
  begin
    WriteLn( 'CoBind() failed: ', Dev^.CoStatusToStr( Status ) );
    Exit;
  end;
```

6.3. Posílání rámců

```
const
  { Nejprve si musíme nadefinovat adresu cílové stanice }

  DestAddr : TEthAddress =
              ( Size:   SizeOf( TEthAddress );
                Mac:    Bytes: ($XX, $XX, $XX, $XX, $XX, $XX);
                Format: ETH_FORMAT_ETHII;
                Proto:  $YYYY
              );

var
  Buffer  : array[0..1499] of Byte;
  BuffLen : Word;
  ..
  ..

  { Zde je potřeba naplnit buffer a proměnnou BuffLen }

  ..
  ..
  { Odeslání rámce na adresu DestAddr }

  Status := Dev^.CoSendBufferTo( @Buffer, BuffLen, 0, @DestAddr,
                                 nil );

  if Status <> CST_SUCCESS then
  begin
    WriteLn( 'CoSendBufferTo() failed: ',
                Dev^.CoStatusToStr( Status );

    Exit;
  end;
  ..
  ..
```

6.4. Příjem rámců

```
var

  Buffer   : array[0..1499] of Byte;
  RecvLen  : Word;
  RecvAddr : TEthAddress;
  ..
  ..

  Status := Dev^.CoRecvBufferFrom( @Buffer, SizeOf( Buffer ),
                                   @RecvLen, 0, @RecvAddr, nil );

  if Status <> CST_SUCCESS then
  begin
    WriteLn( 'CoRecvBufferFrom() failed: ',
                Dev^.CoStatusToStr( Status );
```

```
    Exit;
  end;

  { V proměnné RecvLen je délka přijatých dat překopírovaných do pole
    Buffer. RecvAddr obsahuje adresu odesílatele dat }
```

6.5. Uvolnění prostředků síťové karty

```
{ Přechod zařízení do neaktivního stavu }

Status := Dev^.CoUnbind( 0, nil );

if Status <> CST_SUCCESS then
begin
  WriteLn( 'CoUnbind() failed: ', Dev^.CoStatusToStr( Status ) );
  Exit;
end;

{ Uvolnění instance zařízení z paměti }

CoReleaseDevicePtr( Dev );
```