

CrtCom

KNIHOVNY PRO LADICÍ VÝPISY NA CRT A VSTUP Z KLÁVESNICE PŘES SÉRIOVOU KOMUNIKACI

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: 23.08.2004

Datum posledního uložení dokumentu: 23.08.2004

(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.	Výhody a nevýhody	6
5.	Knihovny balíku CrtCom	7
5.1.	Přilinkování knihoven do řídicí aplikace	7
5.2.	Inicializace ladicích výpisů přes komunikaci	8
5.3.	Obsluha komunikačního automatu	8
5.3.1.	Dostupné procedury a proměnné	10
5.3.1.1.	InitCrtCom	10
5.3.1.2.	CrtComMake	10
5.3.1.3.	CrtComAllChanges	10
5.3.1.4.	Typové konstanty	10
6.	Jak upravit stávající aplikace pro vzdálené ladicí výpisy	11
7.	Několik slov k implementaci přenosu ladicích výpisů	11
7.1.	Požadavky	11
7.2.	Komunikační protokol	11
7.3.	Komunikační automat	12
8.	Crt terminál	13
9.	Příklad	13

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Wil	23.08.2004	První vydání.

1.2. Účel dokumentu

Tento dokument slouží jako popis balíku knihoven CrtCom, které se používají pro zobrazování ladicích výpisů na vzdálený CRT terminál a pro vzdálený vstup z klávesnice.

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.

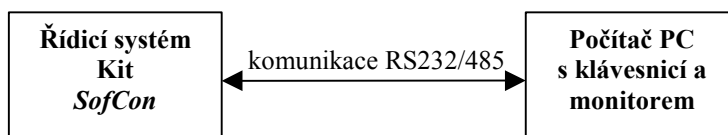
Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu „LibVer“.

2. Termíny a definice

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

3. Úvod

Balík knihoven CrtCom, který je součástí systémových knihoven LIB firmy SofCon spol. s r.o., se používá pro zobrazování ladicích výpisů řídicí aplikace na vzdálený CRT terminál a pro vzdálený vstup z klávesnice. Pojem „vzdálený“ znamená přenos dat mezi řídicím systémem a Crt terminálem po sériové komunikaci RS232 nebo RS485 (viz následující obrázek).



Na počítači PC je spuštěna aplikace „Crt terminál“, která řídí přenos zobrazovaných dat z řídicího systému a je schopna mu posílat kódy stisknutých kláves na klávesnici PC. Počítač PC je v postavení Master (vysílá požadavky), kdežto řídicí systém je v postavení Slave (odpovídá na požadavky).

Řídicí aplikace může po inicializaci vzdálených ladicích výpisů (viz dále) využívat standardních zobrazovacích prostředků knihovny Crt jazyka Borland Pascal nebo podobných či rozšířených knihoven (např. CrtWin z balíku systémových knihoven LIB SofCon spol. s r.o.). Tj. může volat například procedury a funkce TextMode, ClrScr, Write(ln), Read(ln), Window, Gotoxy, Readkey, Keypressed ... Akce týkající se zobrazení se poté projeví na obrazovce připojeného PC ve spuštěné aplikaci Crt terminal. Akce týkající se klávesnice se vztahují na klávesnici připojeného PC, tj. např. funkce Readkey čeká na stisk klávesy z klávesnice PC.

4. Výhody a nevýhody

Výhody:

- Malé nároky na paměť kódu: Obsluha standardní VGA zabírá 160KB kódu, kdežto knihovny CrtCom méně než 30KB.
 - Pozn. 1: Pokud vaše aplikace již používá některé komunikační knihovny (zejména ChnVirt a ChnCom), zabírají knihovny CrtCom necelých 15KB.
 - Pozn. 2: V případě řídicích systémů KitV40 a Kit386EXR nemusí být v kódové paměti rezervován prostor pro VGA, tj. pro aplikaci je k dispozici větší část paměti.
- Řídicí systém nemusí obsahovat VGA kartu ani klávesnici.
- Možnost připojení Crt terminálu k řídicímu systému kdykoliv během chodu aplikace bez nutnosti vypínání.
- Knihovny CrtCom podporují veškerý komfort zobrazovacích knihoven Crt a CrtWin standardně používaných pro ladicí výpisy v řídicích aplikacích. Tj. přechod (úprava aplikace) od zobrazení na standardní VGA kartu ke vzdálenému zobrazení je velice snadný (viz dále v tomto dokumentu).

Nevýhody:

- Přenos dat po sériové komunikaci je pomalejší než standardní přístup k VGA kartě a výpis na monitor, tj. pomalejší odezva příliš rychlých změn obrazovky, což pro řídicí aplikace není závažný nedostatek.
- Pouze textový režim 80x25 /16barev nebo 80x50 /16barev. Ostatní režimy a ani grafika nejsou implementovány.
- Řídicí systém musí obsahovat pro vzdálené ladicí výpisy a vzdálenou klávesnici jeden volný COM port.
- Pro pracovní videopaměť se na HEAPu řídicího systému alokuje 20kB.

5. Knihovny balíku CrtCom

Balík knihoven CrtCom je určen pro tvorbu řídicí aplikaci, která obsluhu videa a klávesnice provádí přes sériovou komunikaci. Balík obsahuje následující knihovny:

- **CrtType** - Základní typy a struktury pro video a klávesnici.
- **Crt2Mem** - Obslužná rutina přerušení Video INT 10h a Keyboard INT 16h.
- **ChnCrt** - Komunikační protokol pro přenos dat po sériové lince (linková vrstva podle ISO/OSI modelu). Fyzický přenos dat probíhá přes nižší komunikační vrstvu (např. ChnCom).
- **CrtComSl** - Komunikační automat Slave strany zajišťující příjem požadavků od Crt terminálu a odeslání odpovědi (síťová až prezentační vrstva podle ISO/OSI modelu).

5.1. Přilinkování knihoven do řídicí aplikace

Abyste mohli používat služby vzdálených ladicích výpisů, musíte do své aplikace v části USES přilinkovat knihovny balíku CrtCom a některou komunikační knihovnu fyzické vrstvy (např. ChnCom).

```
Př.      Uses
          CrtType,
          Crt2Mem,
          ChnVirt,
          ChnCom,
          ChnCrt,
          CrtComSl,
          ...
          Crt,      {funkce pro standardní výpisy}
          CrtWin,   {rozšířené funkce pro ladicí výpisy}
          ...;
```

Důležité: Knihovnu CRT2MEM je nutno přilinkovat PŘED knihovnou CRT nebo jinou knihovnou, která knihovnu Crt používá. Toto platí v celé vaší aplikaci.

Například pokud vaše knihovna uMyDeb.pas používá knihovnu Crt (příkaz uses Crt), není přípustné následující linkování:

```
Uses
  uMyDeb,
```

```
Crt2Mem,  
Crt,  
...;
```

Překladač Vám sice žádnou chybu nenahlásí, ale přeložená aplikace nebude správně fungovat (pravděpodobně bude docházet k opakovanému reset).

Jak zjistit pořadí přilinkovaných knihoven ve Vaší aplikaci? Překladač ukládá pořadí přilinkovaných knihoven do .MAP souboru (pokud máte nastaveno jeho ukládání). Tento .MAP soubor má jednoduchý textový formát, tj. je zobrazitelný v jakémkoliv textovém prohlížeči či editoru. Jednotlivé přilinkované knihovny jsou v .MAP souboru seřazeny tak, že první přilinkované jsou ukládány pod později přilinkované. To znamená, že knihovna Crt2Mem musí být vždy níže než knihovna Crt.

5.2. Inicializace ladicích výpisů přes komunikaci

První část inicializace ladicích výpisů spočívá v nastavení speciální přerušovací rutiny od Video BIOS a Keyboard BIOS. Jak již bylo zmíněno výše, provede se toto automaticky po přilinkování knihovny Crt2Mem.

Druhá část inicializace spočívá v nainicializování komunikačního kanálu, po kterém se budou data přenášet. To se provede zavoláním funkce **InitCrtCom**. Tato funkce má jako parametr textový řetězec obsahující parametry komunikačního kanálu.

Příklad pro KitV40: `InitCrtCom('NAM=CRT MAS=SLAVE LSB=200 NAM=V40_`
`BD=38400 PAR=E LRB=500')`

Pro přenos je použit kanál V40 s komunikační rychlostí 38400Bd.

Příklad pro Kit386EXR: `InitCrtCom('NAM=CRT MAS=SLAVE LSB=500 NAM=COM`
`COM=2 IRQ=3 BD=115200 PAR=E LRB=500')`

Pro přenos je použit port COM 2 na procesorové desce s komunikační rychlostí 115200Bd.

Příklad pro Kit188ER: `InitCrtCom('NAM=CRT MAS=SLAVE LSB=500 NAM=COM`
`ADD=$2300 IRQ=$E BD=115200 PAR=E`
`LRB=500')`

Pro přenos je použit port COM A na procesorové desce s komunikační rychlostí 115200Bd.

Jak je z příkladů patrné, první část řetězce (za NAM=CRT) se týká protokolu ChnCrt. Druhá část řetězce (za druhým NAM=) se týká fyzického komunikačního portu.

Parametr LSB u protokolu ChnCrt určuje velikost vysílacího bufferu, tj. jak maximálně velké bloky dat ladicích výpisů bude řídicí systém posílat Crt terminálu na zobrazení. Jak optimálně nastavit velikost tohoto parametru bude popsáno v dalších kapitolách.

5.3. Obsluha komunikačního automatu

Po inicializaci komunikačního kanálu pro ladicí výpisy je dále nutné periodicky volat komunikační automat, který zajišťuje komunikaci s Crt terminálem.

Tento automat je volán prostřednictvím procedury **CrtComMake**, tj. je nutno periodicky volat proceduru CrtComMake.

Pozn: Jedno volání CrtComMake neznamená odvysílání všech dosud zobrazených změn. Tj. pokud například napíšete `writeln('xxxx')` a potom CrtComMake, vůbec to nemusí znamenat, že po zavolání CrtComMake se Váš nápis 'xxxx' objeví na obrazovce Crt terminálu. Komunikační automat CrtComMake je naprosto průchozí, tj. nemá v sobě žádné čekací smyčky (nečeká na příjem zprávy od Crt terminálu a nečeká ani na konec odvysílání odpovědi). Tím je režie tohoto automatu minimální a aplikace se může věnovat důležitější činnosti, během níž se může odpověď s ladicími výpisy Crt terminálu vysílat na pozadí pod přerušením.

Uvedme si příklad:

```
{inicializace komunikačního automatu}
  InitCrtCom(...);

{hlavní smyčka programu či procesu}
Repeat
  {řídící činnost, výpočty, apod.}
  ...

  {ladicí výpisy}
  Writeln('xxx');
  Gotoxy(...);
  ...
  writeln('yyy');

{krok komunikačního automatu ladicích výpisů}
  CrtComMake;

{v případě použití o.s.ReTOS uvolnění dalších procesů}
  Wait(1);
Until FlEndApp;
```

V CrtComMake se přijímají požadavky od Crt terminálu a nastavují se odpovědi pro vysílání. Vlastní odvysílání odpovědi poté probíhá na pozadí pod přerušením, tj. již po skončení procedury CrtComMake (v příkladu výše během Wait(1) nebo dalších algoritmů či jiných procesů).

Aby se maximálně využilo volného procesorového času a refresh obrazovky ladicích výpisů byl co nejrychlejší, je výhodné posílat Crt terminálu tak velké bloky dat, jejichž vysílání trvá po dobu danou mezi dvěma příkazy CrtComMake. Resp. aby se odvysílaná odpověď stihla odvysílat před dalším voláním CrtComMake, ale zároveň aby co nejlépe vyplnila tuto dobu. K tomuto účelu slouží již dříve zmíněný parametr LSB, který by měl při dané komunikační rychlosti zhruba určovat počet odvysílaných byte mezi dvěma příkazy CrtComMake. Pokud tento parametr nebude takto optimálně nastaven, nejedná se o žádnou chybu. Komunikace bude nadále fungovat, jenom refresh obrazovky ladicích výpisů bude možná o něco pomalejší.

5.3.1. Dostupné procedury a proměnné

Komunikační automat (implementovaný v knihovně CrtComSI) má následující veřejné procedury a proměnné:

5.3.1.1. InitCrtCom

```
function InitCrtCom(const S:string):tChResult;
```

Funkce provede inicializaci komunikačního kanálu s parametrizačním řetězcem *S*. Jako výslednou funkční hodnotu vrátí případný kód chyby (viz chybové kódy *resErr_Xxx* používaných komunikačních knihoven – *ChnVirt*, *ChnCrt*, *ChnCom* apod.).

5.3.1.2. CrtComMake

```
procedure CrtComMake;
```

Procedura provede krok komunikačního automatu. Tato procedura by se měla volat zpravidla na konci těla procesu s ladicími výpisy.

5.3.1.3. CrtComAllChanges

```
procedure CrtComAllChanges;
```

Procedura volá *CrtComMake* tak dlouho, dokud se do Crt terminálu nepřenesou veškeré změny Video paměti. Pokud Crt terminál není připojen, skončí procedura po čase *CrtComAllChangesTOut* (viz níže) na timeout a předá řízení zpět aplikaci. Tuto proceduru může programátor zavolat, pokud například požaduje vidět změny po každém *Write* – většinou to však není potřeba a aplikace používá proceduru *CrtComMake*.

5.3.1.4. Typové konstanty

```
CrtComAllChangesTOut : word = 2000
```

Timeout [ms] pro proceduru *CrtComAllChanges*.

```
CounterRecMess : word = 0;
```

Počítadlo přijatých zpráv od Crt terminálu

```
CounterErrMess : word = 0;
```

Počítadlo chyb při příjmu zpráv od Crt terminálu

```
LastErrMess : tChResult = Res_Ok;
```

Kód poslední zjištěné chyby při příjmu zpráv.

6. Jak upravit stávající aplikace pro vzdálené ladicí výpisy

Postup je velice jednoduchý:

1. V části USES přilinkovat potřebné knihovny (viz kapitola „5.1 Přilinkování knihoven do řídicí aplikace“)
2. Nainicializovat komunikační kanál funkcí **InitCrtCom** (viz kapitola „5.2 Inicializace ladicích výpisů přes komunikaci“)
3. Periodicky volat proceduru **CrtComMake** (viz kapitola „5.3 Obsluha komunikačního automatu“)

7. Několik slov k implementaci přenosu ladicích výpisů

Tato kapitola je zde zařazena jako zajímavost nastiňující programátorovi algoritmy komunikačního automatu a vysvětlující některé způsoby implementace.

7.1. Požadavky

- Jednoduchá SW obsluha pro aplikačního programátora, tj. používání většiny standardních funkcí Pascalské jednotky Crt i SofCon jednotky CrtWin.
- Pomocí jednoduchých úprav možnost zahrnutí těchto vzdálených výpisů i do již naprogramovaných aplikací.
- Není nutnost implementovat grafické módy.
- Malé nároky na paměť kódu, aby algoritmy vzdálených ladicích výpisů zabíraly podstatně menší část paměti kódu než standardní VGA.

7.2. Komunikační protokol

Byl vytvořen komunikační protokol ChnCrt, který má následující vlastnosti:

- Komunikace Master-Slave, řídicí systém je Slave a Crt terminál je Master.
- Nepoužívá se žádná adresace zařízení na sběrnici – přenos probíhá pouze mezi dvěma zařízeními.
- Data zpráv jsou chráněna kontrolním součtem SUM8, který sice neposkytuje tak velké zabezpečení jako například CRC16, ale jeho výpočet je mnohem rychlejší. Ladicí výpisy navíc nevyžadují dokonalé zabezpečení přenosu dat.
- Protokol je binární s důsledným znakovým rozlišením začátků zpráv. V přenosu binárních dat se nepoužívají (nevkládají) žádné znaky transparence. To umožňuje přenos celých bloků Video RAM bez dalšího kódování.

Komunikace zvoleného typu Master (PC) - Slave (řídící systém) má oproti komunikaci z řídícího systému do pasivně zobrazujícího Crt terminálu (kdy řídící systém by zobrazovaná data vysílal a Crt terminál je jen pasivně přijímal a zobrazoval) následující výhody:

- Pokud není Crt terminál připojen, není řídící systém zatěžován žádnou komunikací ladicích výpisů.
- Komunikace Master-Slave umožňuje bezpečné řízení přenosu dat a opakování chybně přenesených zpráv.
- Komunikace Master-Slave umožňuje do budoucna naimplementování dalších ladicích prostředků, jako například čtení bloku paměti, obsahu globálních proměnných apod.
- Největší výhodou ovšem je, že Crt terminál nemusí být připojen k řídícímu systému od začátku, ale je jej možno připojit i v průběhu chodu řídící aplikace. Crt terminál si poté vyžádá obsah celé paměti RAM, tj. na obrazovce se zobrazí veškeré výpisy, dokonce i ty které řídící aplikace vypsal ještě před připojením Crt terminálu (pokud je samozřejmě později nepřepsala jinými výpisy).

7.3. Komunikační automat

1. Po připojení Crt terminálu k řídícímu systému se nejprve vyčtou základní informace o videomódu, tvaru a pozici kurzoru a další parametry.
2. Poté se postupně načtou všechny řádky video paměti.
3. Dále se Crt terminál dotazuje na změněné řádky. Řídící systém si uchovává změny své video paměti v pomocné struktuře a při příchodu požadavku na čtení dat změněných řádků pošle Crt terminálu změněné řádky. Pokud se změnilo pouze hodnoty znaků a jejich atributy (barvy) zůstaly zachovány, posílají se pouze hodnoty znaků. Při změně části řádku se posílá pouze změněný úsek řádku (od prvního změněného do posledního změněného). Tím dochází k rychlejší odezvě vzdálených výpisů. To znamená, že čím méně skutečných změn vaše aplikace do video paměti provádí, tím je jejich výpis na obrazovku Crt terminálu rychlejší.

Komunikační automat řídícího systému posílá Crt terminálu najednou tolik změněných řádků, kolik se mu jich vejde do pomocného vysílacího bufferu (velikost tohoto bufferu je dána parametrem LSB o kterém jsme se zmínili dříve).

V nastavených časových intervalech může Crt terminál provádět refresh všech řádků video paměti (viz bod 2) a poté pokračuje dále čtením pouze změněných řádků.

4. Pokud byla v Crt terminálu stisknuta nějaká klávesa (klávesy) příslušející řídícímu systému, pošle se přednostně do řídícího systému.
5. Pokud řídící systém změní videomód, odpoví na první jakýkoliv požadavek Crt terminálu zprávou se základními informacemi o videomódu (viz bod 1). Tímto způsobem Crt terminál provede patřičnou změnu své obrazovky a pokračuje dále načtením všech řádků video paměti (viz bod 2).

8. Crt terminál

V současné době byla pro tyto účely vytvořena provizorní DOS aplikace **CrtTerm.Exe**, která na celé obrazovce vypisuje totéž, co by se vypisovalo na obrazovku připojenou přímo k řídicímu systému se standardní VGA.

Nastavení parametrů komunikace (COM, komunikační rychlost apod.) je možné v souboru **CrtTerm.Ini**. Změny se samozřejmě projeví až po novém spuštění CrtTerm.Exe.

Po spuštění CrtTerm.Exe jsou stisknuté klávesy zpracovány tímto programem, tj. není aktivována vzdálená klávesnice řídicího systému. Pokud ale uživatel stiskne klávesu F12, budou se poté všechny stisknuté klávesy posílat do řídicího systému, tj. aktivuje se vzdálená klávesnice řídicího systému. Klávesou F11 dojde opět k přepnutí na klávesnici pro Crt terminál, tj. stisknuté klávesy opět přísluší Crt terminálu. Aplikace CrtTerm se ukončuje klávesou Esc – samozřejmě není-li aktivní vzdálená klávesnice řídicího systému.

Je připravována podobná aplikace na platformě Win32, která bude uživateli umožňovat komfortnější ovládání a bude případně rozšiřitelná o další funkce (jako například sledování obsahu paměti).

9. Příklad

Následující příklad je určen pro Kit386EXR. Na obrazovku vypisuje hodnotu neustále se inkrementujícího čítače. Komunikace s Crt terminálem probíhá na portu COM 2 rychlostí 115200 Bd. Jednoduchou úpravou parametrizačního řetězce funkce InitCrtCom a přilinkováním správných komunikačních knihoven lze příklad použít i pro KitV40 nebo Kit188ER.

```

program Example_CrtCom_1;

uses
  CrtType,           {typy Video a Keyboard BIOSu}
  {$ifdef MCP}
  Crt2Mem,           {obsluhy Video a Keyboard BIOSu pro komunikaci na
                    KitXxx}
  {$endif}
  Crt,               {Pascalske funkce pro VGA a klavesnici}
  ChnVirt,           {jednotne rozhranni vseh komunikacnich knihoven}
  ChnCom,            {komunikacni knihovna pro fyzicke rozhranni UART
                    i8250}
  ChnCrt,            {komunikacni protokol pro prenos Video a Keyboard}
  CrtComSl;         {komunikacni automat pro prenos Video a Keyboard}

  {-----}
const
  cAppVer = '01.00,07.06.2004'; {verze tohoto prikkladu}

  Counter : word = 0; {citac pro prikklad}

Begin
  {Inicializace komunikacniho automatu}
  if InitCrtCom('NAM=CRT MAS=SLAVE LSB=200 '+
               'NAM=COM COM=2 IRQ=3 BD=115200 PAR=E STO=1 LRB=500')
    <> res_Ok then RunError(10);

```

```
{vypis na VGA}
  TextMode(CO80);
  TextAttr:=LightGray;
  ClrScr;
  writeln('CrtCom Example 1, ver "',cAppVer,'"');

{hlavni cyklus}
  repeat
    {periodicke vypisy na VGA}
      Inc(Counter);
      gotoxy(1,3); write('Counter ',Counter:5);
    {krok komunikacniho automatu}
      CrtComMake;
  until keypressed; {ukoncit na stisk klavesy z pripojeneho CRT
                    terminalu}

End.
```