

Začínáme s OS FreeDos na Kit188ER

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 01.00

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: 03.05.2005

Datum posledního uložení dokumentu: 03.05.2005

(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.....	4
1.1.	Revize dokumentu.....	4
1.2.	Účel dokumentu	4
1.3.	Rozsah platnosti	4
1.4.	Související dokumenty	4
2.	Termíny a definice	4
3.	Úvod.....	5
4.	FreeDos	5
4.1.	Vytvoření CF pro spuštění OS FreeDos.....	5
4.2.	Konfigurace Kit188ER pro OS FreeDos.....	6
4.3.	Textový výstup	6
4.4.	Knihovny SofCon.....	6
5.	Ukázkové aplikace pod OS FreeDos	6
5.1.1.	Př. 1. Příkazový řádek	6
5.1.2.	Př. 2. Práce se soubory v prostředí Borland Pascal.....	7
5.1.3.	Př. 3. Residentní program pomocí příkazu Keep	8

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
01.00	YY.XX	Net	3.5.2005	První vydání

1.2. Účel dokumentu

Tento dokument slouží k základnímu seznámení s OS FreeDos a postup pro spuštění na procesorové jednotce Kit188ER.

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 se orientovat v používání programového vybavení SofCon.

2. Termíny a definice

Compact FLASH karta – paměťové blokové medium s náhodným přístupem. Rozhraní kompatibilní s pevnými disky z PC (IDE). Data jsou organizována pomocí souborového systému (FAT). Princip ukládání dat podobný FLASH paměti, tzn. omezený počet zapisovacích cyklu (milióny).

Write Protect mód – pokud je paměť Kit188 ve stavu *Write Protect*, tak nelze zapisovat do určité paměťové oblasti (dáno konfigurací). Zpravidla se tak chrání paměť kódu.

Další termíny a definice používané v terminologii SofCon jsou popsány v samostatném dokumentu „Termíny a definice“.

3. Úvod

Dokument popisuje možnosti Kit188ER při použití operačního systému FreeDos, a základny pro sestavení aplikace pod překladačem Borlad Pascal 7.0.

4. FreeDos

FreeDos je volně šiřitelný (licence GNU General Public License) operační systém, který poskytuje stejnou množinu funkcí jako známý MSDOS. Lze ho zdarma získat na stránkách <http://www.freedos.org>. Pro Kit188ER se používá upravený FreeDos firmou SofCon, hlavní rysy tohoto systému zůstali zachovány. Hlavním přínosem FreeDos pro Kit188ER:

- práce se souborovým systémem (otevření souboru, zápis, přepis, čtení...)
- přímé spouštění exe a com aplikací
- plná podpora knihoven SofCon

Pod systémem FreeDos pro Kit188ER lze provozovat aplikace pouze v reálném režimu.

4.1. Vytvoření CF pro spuštění OS FreeDos

Compact Flash kartu vložíme do Kit188ER. Do paměti FLASH nahrajeme aplikace FDISK188. Tato aplikace zkontroluje MBR a BootSector, pokud MBR chybí, bude vytvořen (vytvoří se jedna primární partition o maximální velikosti CF). Po skončení programu vložíme CF do čtečky Compact Flash karet PC. Pokud se Compact Flash kartu nepovede otevřít, musíme jí zformátovat (tento stav nastane pokud neexistoval MBR sektor, nebo byl poškozen). Po těchto operacích můžeme na CF vytvořit potřebné soubory. Pak vrátíme CF zpět do Kit188ER a opět zpustíme aplikaci FDISK188 (z důvodu korekce BootSektoru, který byl s největší pravděpodobností formátem přepsán).

Pro účely knihovny CFlash je vhodné hned po zformátování vytvořit soubor LINEAR.DAT pro odkládání dat na Compact Flash kartu v lineárním módu bez použití operačního systému FreeDos na Kit188ER. Soubor může mít libovolnou velikost s ohledem potřeby vaší aplikace (jedná se o obdobu swapovacího souboru OS).

Na Compact Flash kartu nahrajeme běžným způsobem jádro operačního systému FreeDos (*kernel.sys*), příkazový interpret (*command.com*) a dávkový soubor *autoexec.bat*, který je zpracován při startu systému. Tyto soubory musí být umístěny v kořenovém adresáři Compact Flash karty. Do souboru *autoexec.bat* uvedeme jméno spouštěné aplikace (stejná syntaxe jako u „klasického autoexecu“ v systému DOS).

Soubor *kernel.sys* je součástí softwarové podpory firmy SofCon, soubor *command.com* je součástí projektu FreeDos a je volně ke stažení ze stránek OS FreeDos pod názvem FreeCom.

4.2. Konfigurace Kit188ER pro OS FreeDos

Procesorovou jednotku Kit188ER zkonfigurujeme pomocí aplikace KitLoader. Jméno souboru v konfigurační tabulce nastavíme na *kernel.sys*, položku „segment“ nastavíme na hodnotu \$100. Propojku pro volbu Apl/BMon zkratujiem (standartní spuštění aplikace). Po resetu bude Kit188ER nejprve hledat aplikaci (tj. *kernel.sys*) na Compact Flash kartě, pokud selže (ve slotu nebude Compact Flash) bude hledat aplikaci v paměti FLASH.

4.3. Textový výstup

Pomocí systémového komunikačního portu na desce Kit188ER můžeme propojit Kit188ER s klasickou ASCII terminálovou aplikací, např. HyperTerm spuštěnou na PC. Lze použít stejného komunikačního kabelu jako při komunikace mezi Kitem a KitLoaderem, před spuštěním terminálové aplikace musíme KitLoader ukončit (uvolnit COM port). Parametry komunikace terminálu nastavíme na: rychlost 115200Bd, 1 stop bit, 8 datových bitů, parita žádná. V okně terminálu se budou vypisovat informace o průběhu bootování os FreeDos. Po nahrání jádra operačního systému dojde ke spuštění příkazového interpreta *command.com*, který zpracovává příkazy v dávkovém souboru *autoexec.bat*. Proces bootování končí spuštěním aplikace uvedené v souboru *autoexec.bat* nebo příkazovým řádkem „C:\>“. Komunikace mezi PC a Kit188ER probíhá v obou směrech, tzn. stisknuté klávesy z klávesnice PC jsou posílány do Kit188ER a zpracovány, proto lze používat klasických DOS příkazů typu *DIR*, *COPY*, *MOVE*...apod.

4.4. Knihovny SofCon

Aby byla aplikace spustitelná z EXE souboru pod OS FreeDos stačí aplikaci přeložit s novými knihovnami SofCon bez dalších úprav programu. Tyto nové knihovny lze nalézt v adresáři \LIB\DOS188.

5. Ukázkové aplikace pod OS FreeDos

5.1.1. Př. 1. Příkazový řádek

Pokud máme připojený ASCII terminál na komunikačním portu (postup připojení popsán v kap. 4.3), měli by jsme po spuštění FreeDosu dostat hlášení o verzi OS FreeDos, licence pod kterou je operační systém šířen. Pak následují informace o připojené Compact Flash kartě a konečně příkazový řádek „C:\>“.

Můžeme vyzkoušet základní příkazy OS Freedos jako např.

C:\>DIR - *výpis obsahu adresáře*

C:\>COPY KERNEL.SYS KERNEL.BAK - *vytvoření kopie souboru*

C:\>TYPE AUTOEXEC.BAT - *výpis obsahu textového souboru*

Pro seznam všech příkazů podporovaných interpretem FreeCom použijte „?“.
Chování příkazů je dle očekávání stejné jako na OS MSDOS na běžném PC.

5.1.2. Příklad 2. Práce se soubory v prostředí Borland Pascal

Jako ukázkový příklad práce se soubory předkládáme jednoduchý program, který vypíše obsah souboru (pokud existuje), vyžádá si nový obsah od uživatele a uloží jej do stejného souboru.

```
Program Ex2_FileIO;
const FileName = 'c:\sample.txt';
var F: Text;
    s:string;
BEGIN
  {$I-} {Vypnout kontrolu vstupne/vystupnich operaci}
  Assign(F, FileName);
  Reset(F);      {Otevreni souboru pro cteni}
  {$I+}
  if IOResult=0 then    {vysledek operace otevreni souboru, existuje?}
  begin
    readln(F,s);  {precti radek ze souboru}
    writeln('Obsah Souboru ',FileName,': ',s);
    Close(F);
  end else
  begin
    writeln('Soubor Neexistuje');
  end;

  Rewrite(F);    {Otevreni souboru pro zapis}
  Write('Novy obsah souboru ',FileName,': ');readln(s);
  write(F,s);    {Zapis do souboru vstup od uzivatele}

  Close(F);
  Writeln('---- Konec Prikladu Ex2 ----')
END.
```

5.1.3. Př. 3. Residentní program pomocí příkazu Keep

Následující program je ukázkou použití příkazu Keep (ukončí aplikaci a zůstane rezidentní). Program se po spuštění vrátí zpět na příkazovou řádku, svůj kód „zavěsí“ na uživatelský časovač Int1c, který je volán s periodou 55ms. Při každém 16-tém vyvolání procedury invertuje výstup pro LED diodu D5.

```
program Ex2_Keep;
uses Dos;
var
  IntVec_1C : Procedure;
  Counter:word;

procedure XorLed;interrupt;
begin
  if (counter and $F) = 0 then
  begin
    port[$FF7A]:=port[$FF7A] xor $40; {blikani LED5}
  end;
  if counter = $FFFF then counter:=0      {osetreni pretecení}
  else inc(counter);
  asm { nasleduje volani puvodni obsluhy Int1C}
    {puvodni procedura byla prerusovaci rutina,
     a proto bude ukoncena instrukci iret, ktera vybira
     ze zasobniku i registr FLAGS, proto je musime rucne
     ulozit}
  pushf
    {Puvodni interupt rutina Int1C obsluhuje watchdog, podle
     hodnoty registru AX. Platna hodnota AX registru je ulozena v
     urovni 16 na zasobniku (8. word v poradi) }
  mov  ax, [ss:bp+16]
  call IntVec_1C {volani puvodni Int1C rutiny se spravnym reg. AX }
  end;
end;

BEGIN { main }
  GetIntVec($1c,@IntVec_1C); { ulozeni adresy puvodni obsluzne rutiny }
  SetIntVec($1c,Addr(XorLed)); { nastaveni nove obsluhy }
  Keep(0); { ukonci aplikaci ale nech v pameti kod }
END.
```