

# CFLASH

## JEDNOTKA PRO LINEÁRNÍ PŘÍSTUP K COMPACTFLASH PAMĚTEM NA KIT188ER

Příručka uživatele a programátora



**SofCon<sup>®</sup> spol. s r.o.**  
Střešovická 49  
162 00 Praha 6  
tel/fax: +420 220 180 454  
E-mail: [sofcon@sofcon.cz](mailto: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: 10.05.2005

Datum posledního uložení dokumentu: 10.05.2005

(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	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.	Popis konstant, typů a proměnných	7
4.1.	Informace o CF kartě	7
4.2.	Kódy výsledků funkcí přístupujících k CF kartě	7
5.	Objekt TCFlash	8
5.1.	Položky	8
5.2.	Funkce CFInit	8
5.3.	Funkce CFRead	8
5.4.	Funkce CFWrite	8
5.5.	Funkce CFGetFileSize	9
5.6.	Procedura CFSetVerify	9
6.	Příklad	9



## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Wil	10.05.2005	První vydání.

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis jednotky pro práci s paměti COMPACT FLASH pro absolutní přístup do lineárně vyhrazeného prostoru.

### 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.

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

---

Knihovna byla vytvořena za účelem možnosti přístupu k pamětem COMPACT FLASH (dále jen CF) libovolných výrobců a velikostí<sup>1</sup> bez nutnosti využívání služeb DOSu na řídicím systému **Kit188ER**. Pro jednoduchou implementaci přístupu k CF byl zvolen následující způsob: Na CF kartě je standardní souborový systém FAT16 s Boot sektorem a MBR (Master Boot Record, který určuje rozdělení na logické disky) nebo i bez MBR. Kdekoliv v Root (tj. v hlavním) adresáři musí být nahrán příslušný **!spojitý soubor!**, ke kterému se bude přistupovat. Těchto souborů může být i více podle potřeb dané aplikace. Vždy však musí platit, že soubor musí být umístěn ve spojitých sektorech od začátku do konce (tj. nesmí být fragmentovaný), jelikož knihovna používá pro přístup k CF kartě služeb BIOSu Int 13H (čtení a zápis sektoru v LBA režimu). Tento přímý přístup k sektorům je velice jednoduchý, nenáročný na výpočty, paměť a zpracovávání FAT. Na druhou stranu je nevýhodou výše zmíněná nutnost spojitosti souboru pro lineární přístup a jeho umístění v hlavním adresáři CF karty. Velikost daného souboru záleží na potřebách dané aplikace, tj. může být libovolná až do velikosti 2GB. Knihovna poté definuje objekt **TCFlash**, který umožní přistupovat do jednoho takového lineárního souboru absolutním (lineárním) způsobem. To znamená, že pokud je soubor pro lineární přístup velký například 1 234 567 byte, potom absolutní přístup do souboru se provádí od adresy 0 do adresy 1234566 (velikost souboru - 1). Adresa 0 přistupuje k prvnímu byte v souboru, naopak adresa 1234566 přistupuje k poslednímu byte v souboru. Tento způsob adresování je obdobou k absolutnímu přístupu do paměti RAM.

Po přilinkování této knihovny se provede v inicializační části detekce paměti CF. Aplikace pro přístup do daného souboru musí zavolat funkci **CFInit** s názvem souboru. Ta provede nalezení tohoto souboru pro lineární přístup a kontrolu jeho spojitosti. Poté lze využívat funkci **CFRead** pro čtení z absolutní adresy a funkci **CFWrite** pro zápis na absolutní adresu v souboru pro lineární přístup. Objekt dále obsahuje proměnnou informující o velikosti souboru pro lineární přístup a nastavitelnou proměnnou povolující verifikaci (tj. zpětné čtení s ověřením obsahu) zápisu dat na CF kartu.

**Při používání systému reálného času ReTOS** je nutno si uvědomit, že **čtení ani zápis** do CF karty **nejsou reentrantní**. To znamená, že pokud k CF kartě přistupuje více procesů (ať už do jednoho nebo více souborů), musí se tyto přístupy provádět v chráněné sekci ReTOSu (viz příkazy Lock/UnLock nebo LockKernel/UnLockKernel).

Dále je nutné si uvědomit, že přístup (čtení i zápis) k CF kartě trvá řádově jednotky milisekund na 512 byte a je závislý i na výrobcí a typu CF karty. Zápis je pomalejší než čtení (zhruba 2x), jelikož se při něm provádí verifikace zapsaných dat. Programátor aplikace by měl uvážit, ve kterých časově kritických sekcích řídicího a regulačního algoritmu přistupovat k CF kartě. **Rozhodně by se k CF kartě nemělo přistupovat v přerušovací rutině nebo pod zakázaným přerušením (instrukce CLI).**

Knihovna je použitelná i pro DOS na PC a i pro FreeDOS na Kit188ER. Pro tyto účely se podmíněným překladem nepoužívá přímého přístupu čtení a zápisu

---

<sup>1</sup> Jelikož Borland Pascal 7.0 má omezenou velikost lineárního adresování 2GB (hodnota High(Longint)) použije se v případě osazení větších CF karet maximální přístup do 2GB.

sektorů, ale služeb DOSu. Tyto služby umožňují kompletní správu FAT, což má následující výhody: soubor nemusí být spojitý (může být fragmentovaný) a nemusí ležet v Root adresáři disku, ale na libovolné cestě.

Na závěr jedno upozornění: Paměť CF má omezený počet zápisů do jednoho sektoru. Toto omezení závisí na typu a výrobci dané CF karty (řádově sta tisíce až milion zápisů).

## 4. Popis konstant, typů a proměnných

---

```
cVerNo = např. $0251; { BCD formát }
cVer   = např. '02.51,07.08.2003';
```

Číslo verze této jednotky v BCD tvaru a v textové podobě včetně datumu poslední změny.

```
tFileName = {$ifdef DOS} string; {$else} string[12]; {$endif}
```

Typ pro jméno souboru. Pro Kit188ER musí být jméno složeno z názvu (max. 8 znaků) a koncovky (max. 3 znaky) oddělené tečkou. Pro DOS může být ve jménu obsažena i kompletní či částečná cesta k souboru. Jméno může být uvedeno velkými i malými znaky (např. 'Events.DAT').

### 4.1. Informace o CF kartě

---

```
CFSize      : Longint = 0;
```

Proměnná informuje o velikosti CF karty v bytech (rozsah do 2GB). Konkrétní hodnota této proměnné se naplní v inicializační části jednotky. Hodnota 0 znamená žádnou CF kartu.

```
fCFAccess   : Boolean = False;
```

Proměnná informuje o práve probíhajícím přístupu (čtení/zápis) k CF kartě. Pro aplikaci je readonly.

### 4.2. Kódy výsledků funkcí přistupujících k CF kartě

---

```
resCF_Ok           = 0; {vsechno v poradku}
resCF_NoCF         = 1; {CF karta není přístupna}
resCF_FormatCFErr = 2; {CF karta má špatný formát (MBR,Boot,FAT
                        apod.)}
resCF_NoLinFile    = 3; {na CF kartě nebyl nalezen zadaný soubor pro
                        lineární přístup}
resCF_LinFileErr   = 4; {zadaný soubor pro lineární přístup není
                        spojitý}
resCF_RWOverflow   = 5; {při čtení/zápisu by došlo k přístupu mimo
                        lineární prostor}
resCF_ReadErr      = 6; {chyba při čtení sektoru z CF}
resCF_WriteErr     = 7; {chyba při zápisu sektoru do CF}
resCF_VerifyErr    = 8; {chyba při verifikaci zápisu sektoru do CF}
resCF_Busy         = 10; {k CF kartě již někdo přistupuje
                        (fCFAccess=True), zkuste službu později}
```

---

## 5. Objekt TCFflash

---

---

### 5.1. Položky

---

CF\_FileName : tFileName;

Název souboru pro lineární přístup. Nastaví se ve funkci CFInit.

CF\_FileSize : Longint;

Proměnná informuje o velikosti souboru pro lineární přístup (rozsah do 2GB). Konkrétní hodnota této proměnné se naplní ve funkci CFInit.

CF\_FileBase : Longint;

Proměnná informuje o absolutní adrese začátku souboru pro lineární přístup na disku. Položka se naplní ve funkci CFInit. Vydělíme-li tuto hodnotu velikostí sektoru (512), získáme absolutní adresu počátečního sektoru v LBA režimu.

CF\_VerifyWr : Boolean;

Příznak požadavku verifikace zpětného čtení zapsaných dat.

---

### 5.2. Funkce CFInit

---

```
function CFInit(const FileName:tFileName):word;
```

Funkce provede nalezení souboru *FileName* pro lineární přístup a naplní položky *CF\_FileName*, *CF\_FileSize* a *CF\_FileBase*. Jako funkční hodnotu vrací kód výsledku úspěšnosti operace (viz konstanty *resCF\_xxx*). Funkce není reentrantní pro ReTOS, proto je ji nutno volat v chráněné sekci (viz příkazy Lock/UnLock nebo LockKernel/UnLockKernel).

---

### 5.3. Funkce CFRead

---

```
function CFRead(aSou:longint; aDest:pointer; aLen:word):word;
```

Funkce přečte ze souboru pro lineární přístup CF karty od absolutní adresy *aSou* *aLen* bytů a uloží je na adresu v RAM danou ukazatelem *aDest*. Parametr *aSou* by měl nabývat hodnot 0 až *CF\_FileSize-1*. Jako funkční hodnotu vrací kód výsledku úspěšnosti operace (viz konstanty *resCF\_xxx*). Funkce není reentrantní pro ReTOS, proto je ji nutno volat v chráněné sekci (viz příkazy Lock/UnLock nebo LockKernel/UnLockKernel).

---

### 5.4. Funkce CFWrite

---

```
function CFWrite(aSou:pointer; aDest:longint; aLen:word):word;
```

Funkce zapíše z adresy v RAM dané ukazatelem *aSou* *aLen* bytů do souboru pro lineární přístup CF karty na absolutní adresu *aDest*. Parametr *aDest* by měl nabývat hodnot 0 až *CF\_FileSize-1*. Jako funkční hodnotu vrací kód výsledku úspěšnosti operace (viz konstanty *resCF\_xxx*). Funkce není reentrantní pro ReTOS, proto je ji nutno volat v chráněné sekci (viz příkazy Lock/UnLock nebo LockKernel/UnLockKernel).



## 5.5. Funkce CFGetFileSize

---

```
function CFGetFileSize : longint;
```

Funkce vrátí velikost souboru pro lineární přístup. Vrátí-li hodnotu 0 znamená to, že soubor, který byl zadán ve funkci CFinIt, neexistuje nebo není spojitý.

## 5.6. Procedura CFSetVerify

---

```
procedure CFSetVerify(fVerify:boolean);
```

Procedura nastaví příznak požadavku verifikace zpětného čtení zapsaných dat.

## 6. Příklad

---

Předpokládejme velikost souboru pro lineární přístup alespoň 2\*SizeOf(tData). Příklad provede přečtení struktury tData ze začátku souboru a zkopíruje její obsah za tuto strukturu, tj. v souboru budou dvě identické struktury tData.

### Uses

```
CFlash;
```

### Type

```
pData = ^tData;
tData = {nejaka struktura, např. record}
```

### Var

```
Res :word; {kod vysledku operace}
Data:pData; {ukazatel na data}
CF :tCFflash; {objekt pro linearni pristup k CF karte}
```

### Begin

```
New(Data);

if CFSize=0 then
  writeln('No CF')
else
  writeln('CF size = ',CFSize shr 20,'MB');

{inicializace souboru pro linearni pristup}
Res:=CF.CFInit('EVENTS.DAT');
if Res<>resCF_Ok then
  Halt; {soubor zadaneho nazvu neexistuje nebo je nespravny}
if CF.CFGetFileSize<2*SizeOf(tData) then
  Halt; {soubor je moc maly}

{precteni prvnioho zaznamu tData}
Res:=CF.CFRead(0{abs.adresa v souboru}, Data; SizeOf(tData));
if Res<>resCF_Ok then
  Halt; {chyba pri cteni ze souboru}

{zkopirovani do druheho zaznamu tData}
Res:=CF.CFWrite(Data, SizeOf(tData){abs.adresa v souboru},
  SizeOf(tData));
if Res<>resCF_Ok then Halt; {chyba pri zapisu do souboru}

Dispose(Data);
```

### End.

Pozn: Pro procesor Kit188ER je třeba tento příklad opatřit příkazy pro výpis (příkazy write/writeln) na vzdálený CrtTerminál (viz knihovny CrtCom a příslušné příklady).