Termíny a definice

TERMÍNY A DEFINICE POUŽÍVANÉ V MANUÁLECH FIRMY SOFCON

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 1.30

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: 12.08.2004
Datum posledního uložení dokumentu: 12.08.2004
(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

1.1. Revize dokumentu

Verze dokumentu	Autor	Datum vydání	Popis změn
1.00	Hv	05.03.2003	První vydání.
1.10	Tu	22.05.2003	Úprava dokumentu dle ISO9000.
1.30	Wil	12.08.2004	Upřesnění a doplnění termínů.

1.2. Účel dokumentu

Tento dokument slouží jako seznam termínů a definic používaných v manuálech SofCon týkajících se zejména softwarové podpory řídicích systémů KitXxx.

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.

2. Termíny a definice

Položka je používána pro jednotlivé proměnné ve strukturách, záznamech (RECORD) a objektech (OBJECT).

Jednotka programu je soubor definovaný příkazem UNIT v Borland Pascal. Tento soubor obsahuje seskupení funkcí, proměnných a typů týkajících se zpravidla řešení nějaké problematiky.

Knihovna je jednotka programu, která je ovšem použitelná pro více aplikací, tj. zabývá se nějakou konkrétní problematikou, která není pevně spjata s danou aplikací.

Řídicí systém je tvořen základní procesorovou deskou KitXxx (např. KitV40, Kit386EXR apod.), ke které je možno připojit další periferní desky (např. IOPCOM, IODXO01 apod).

Terminál je speciální případ řídicího systému, který obsahuje navíc modul pro grafické či textové zobrazení dat. Tento modul je zpravidla k řídicí jednotce připojen přes sběrnici IOBus.

PC-Kit je speciální deska do ISA sběrnice stolního PC, která umožňuje ladění aplikací s reálným hardware (desky periferií) přímo na počítači PC. To znamená, že funkci řídicí jednotky KitXxx nahrazuje procesor PC a přes desku PC-Kit jsou připojeny zbývající periferní desky řídicího systému. S tímto ohledem je nutno provést překlad aplikace (zpravidla podmíněný), jelikož PC mívá odlišné parametry a možnosti než řídicí jednotka KitXxx (např. jiná adresa sběrnice IOBus, PC může k ukládání dat využívat pevný disk kdežto KitXxx má k dispozici paměť EEPROM).

LPT-Kit je obdobou desky PC-Kit s tím rozdílem, že se k PC místo na ISA sběrnici připojuje přes LPT, tj. paralelní port.

Systémové knihovny (LIB) je balík obecných systémových knihoven používaných pro tvorbu aplikací pro terminály a řídicí systémy KitXxx.

Terminálové knihovny (LIBT) je balík starších knihoven používaných pro grafické a textové terminály. Je určen pouze pro starší aplikace. Nové aplikace by měly používat nové technologie řešení obsluhy terminálů, kterou definují **vizualizační knihovny**.

Vizualizační knihovny (LIBV) je balík knihoven používaných pro grafické a textové terminály s procesorovou deskou KitXxx. Kromě toho jsou tyto knihovny použitelné i pro zobrazování na monitoru PC.

Kernel je knihovna obsahující hlavní kód operačního systému reálného času ReTOS. Někdy se používá ve významu ReTOS.

ReTOS je operační systém reálného času. Podrobný popis najdete v samostatném manuál „ReTOS“.

Atomická operace je instrukce (příkaz), který nelze přerušit pomocí přerušení. Na řídicích systémech KIT lze takto používat přístupy do proměnných typu Byte a Word. Přístupy k proměnným typu LongInt, Real a složitějších struktur jsou přeloženy pomocí více instrukcí procesoru a proto mohou být přerušitelné.

Segmentová adresa je označení používané při přístupu k paměti v reálném režimu. Tato adresa je tvořena dvěma částmi nazývanými segment a offset. Tuto adresu převedete na lineární adresu následujícími příkazy:

(LongInt(Segment) shl 4) + Offset.

Protože Segment a Offset není v jazyce PASCAL normován, lze vytvořit různé kombinace, které odkazují na stejnou paměťovou buňku. Proto je lepší při zadávání adres používat lineární adresu (pokud to lze), kde žádná víceznačnost nehrozí.

Lineární adresa je označení používané při přístupu k paměti v chráněném režimu. Pomocí této adresy se lze i v reálném režimu jednoznačně odkazovat na paměťovou buňku.

Paměť RAM je prostor, odkud se hodnoty mohou číst a kam se mohou zapisovat bez žádných speciálních funkcí. Někdy se také označuje jako paměť **RWM**.

Paměť RWM, viz paměť RAM.

Paměť ROM je prostor, odkud se hodnoty mohou pouze číst. Zápis do této paměti je možný pouze pomocí speciálního přístupu podle typu paměti ROM: Do paměti

EPROM lze hodnoty zapsat vícekrát po jejich smazání pomocí programátoru. Do paměti PROM lze hodnoty zapsat pouze jednou pomocí programátoru. Do paměti FLASH, EEPROM lze hodnoty zapsat vícekrát dle typu buď pomocí programátoru nebo pomocí speciálních funkcí pro zápis.

Paměť EPROM, viz ROM.

Paměť PROM, viz ROM.

Paměť FLASH, viz ROM.

Paměť EEPROM, viz ROM.

RTD zkratka pro Retos Debugger. Program spustitelný pod DOS na PC umožňující převod přeložené aplikace (EXE) do binární formy (BIN soubory) a její nahrání do EPROM a FLASH pamětí řídicího systému KitXxx. V řídicím systému musí být spuštěn BiosMonitor, který je součástí BIOS. Podrobný popis RTD najdete v samostatném manuálu „Retos Debugger“.

KitLoader je vylepšená a přepracovaná verze programu RTD. Program KitLoadr je spustitelný pod Win95 a vyšší (jedná se o Win32 aplikaci). Oproti programu RTD byla přidána podpora řídicího systému Kit188ER. Podrobný popis KitLoader najdete v samostatném manuálu „Kit Loader“.

Modul aplikace je definován v RTD jako seskupení určitého počtu jednotek aplikace. Každý modul aplikace má své jedinečné jméno, počáteční adresu a maximální velikost. Jednotlivé moduly se nesmí překrývat. Rozlišujeme dva typy modulů aplikace: ROM a RAM (datový modul). ROM moduly se poté nahrávají každý ve formě jednoho BIN souboru do řídicího systému. Podrobný popis RTD a modulů aplikace najdete v samostatném manuálu „Retos Debugger“.

Datový modul je RAM modul aplikace, který smí obsahovat pouze jednotky DATA, STACK a HEAP, viz manuál „Retos Debugger“.

Jednotka HEAP je vytvořena v programu RTD a je to označení rezervované paměti, která je spravována systémovými knihovnami Borland Pascal. Někdy je tato paměť také označována jako halda případně dynamická paměť. Velikost této jednotky je určena v RTD ve zbytku datového modulu po odečtu velikosti jednotek Data a Stack. V žádném případě velikost HEAP **není** nastavena pomocí překladače Borland Pascal nebo direktivou programu.

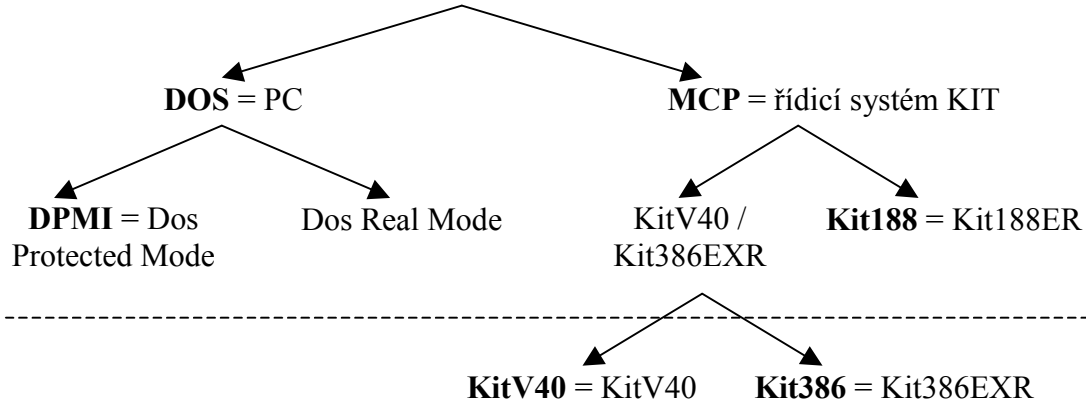
Jednotka DATA je vytvořena v programu RTD a je to označení paměti obsahující veškeré globální proměnné a inicializované proměnné jednotek a aplikace. Velikost této jednotky je dána počtem a typem těchto proměnných.

Jednotka STACK je vytvořena v programu RTD a je to označení paměti používané pro zásobník, tj. lokální proměnné, návratové adresy apod. Velikost této jednotky je dána překladačem. V případě použití ReTOS a procesů určuje tato paměť Stack hlavního procesu. Stack dalších procesů je určen při jejich vytváření.

Zálohovaná paměť RAM je část paměti RAM, která zabírá zbývající prostor RAM nad datovým modulem, tj. nad STACK, DATA a HEAP. Obsah této paměti je zálohován baterií a při startu aplikace je zachován. Do této části paměti se zpravidla ukládají proměnné a struktury, které aplikace potřebuje zachovávat i po výpadku napájení (např. technologické parametry).

Jednotka –LOADER- je vytvořena v programu RTD a slouží ke spuštění aplikace. Velikost této jednotky je dána automaticky RTD. Umísťuje se do libovolného ROM modulu aplikace a tento modul poté nazýváme jako spouštěcí.

Podmíněné překlady jsou řešeny pomocí následujících příkazů překladače Borland Pascal,{$define XXX}, {$ifdef XXX}, {$else} a {$endif}. Tyto příkazy jsou s výhodnou používány pro psaní aplikací, které mohou být spouštěny jak na počítači PC tak na řídicích systémech KIT. Protože systémy KIT mají různé procesory, v dnešní době např. V40, I386EXR je opět výhodné použít podmíněné překlady, pokud zdrojové soubory aplikace mají být použitelné na obou procesorech. V současné době se používá zejména těchto ustálených podmíněných direktiv:



Pozn: Podmíněné direktivy nad čárkovanou čárou jsou na úrovni systémových knihoven LIB, tj. pro KitV40 i Kit386EXR jsou knihovny stejné. Podmíněný překlad pro KitV40 nebo Kit386EXR (pokud je potřeba) se provádí tedy až ve vlastní aplikaci doporučenými direktivami KitV40 a Kit386.

RunTime Error neboli chyba za běhu programu. Jednotlivé chyby jsou očíslovány od 1 do 255. Např. chyba 201 znamená Range Check Error neboli chybu rozsahu při přístupu k polím přes index, naplnění proměnné hodnotou mimo meze apod. Aplikace může chybu za běhu programu vyvolat i přímo příkazem RunError(číslo 0..255). Při vzniku či vyvolání chyby se aplikace ukončí a začne se zpracovávat řetězec Exit procedur.

Exit procedura (ExitProc) je procedura, která se zavolá při ukončení aplikace. Pokud aplikace skončila chybou za běhu programu (RunTime Error) je v proměnné ExitCode uložen nenulový kód chyby a v proměnné ErrorAddr adresa (pointer) místa, kde chyba vznikla. Velice se doporučuje kód a adresu chyby ukládat do restart struktury aplikace. Jednotlivé Exit procedury lze na sebe řetězit tak, že hodnotu proměnné ExitProc uložíte do svojí pomocné proměnné (např. SaveExitProc) a ExitProc naplníte adresou na vaši FAR proceduru (např. MyExitProc). V těle této vaší procedury nesmíte zapomenout provést opět obnovu proměnné ExitProc:=SaveExitProc. Tím se zajistí, že jako první Exit procedura se vyvolá vaše MyExitProc a po jejím skončení původní ExitProc, tj. SaveExitProc. Takto lze jednotlivé Exit procedury na sebe řetězit, přičemž platí pravidlo, že POSLEDNÍ nastavená Exit procedura se vyvolá jako PRVNÍ. Některé knihovny nastavují své vlastní Exit procedury.

Některé toto provádějí již v inicializační části knihovny, jiné při zavolání určité procedury.

Restart struktura je datová struktura v zálohované RAM, do které se ukládají jednotlivé chyby za běhu programu (RunTime Error). Záznam jedné chyby se skládá zpravidla z kódu chyby, adresy místa vzniku, datumu a času vzniku a textového popisu, který může například obsahovat název právě běžícího procesu (viz ReTOS). Zpětným prohlížením záznamů v restart struktuře může programátor odhalit případné chyby a opravit je. Proto by každá aplikace měla definovat a obsluhovat takovouto strukturu. Restart struktura může navíc obsahovat čítače pro počet startů aplikace apod.

Kruhový buffer je označení oblasti paměti, která po svém zaplnění začne data ukládat od začátku, tzn. přeteče na začátek. Při používání tohoto bufferu dochází po zaplnění k přepisu nejstarších dat.

Sklad je označení oblasti paměti, která po svém zaplnění začne vracet chybový kód a při jejím používání nedochází ke ztrátě dat vlivem přetečení.

Instance objektu se nazývá oblast paměti typu objekt. Tato paměť může být např. vytvořena v jednotce Data, globální proměnná typu objekt, nebo v dynamické paměti pomocí funkce *new* při použití konstruktoru.

Záznam archivu je používán pro strukturu archivu, která se ukládá do rezervované paměti a pracuje se s ní pomocí objektů archivu. Tato struktura má po dobu existence archivu pevně daný formát. Změny této struktury bez konverze dat vedou ke ztrátě uložených dat.

Popisovač archivu je struktura, která je uložena zpravidla v zálohované paměti (nad datovým modulem) a obsahuje stav archivu.

Popisovač tabulky je struktura, která slouží pro zadání zobrazení záznamů archivu v programu TheKing v podobě tabulky. V této struktuře jsou definována jména sloupců a typy položek jednotlivých záznamů.

Popisovač grafu je struktura, která slouží pro zadání zobrazení záznamů archivu v programu TheKing v podobě grafu. Před definicí tohoto popisovače musí být vždy zadán popisovač tabulky, protože definuje typy a textové popisy jednotlivých položek záznamu.

Vizualizační programy TheKing, TheQueen jsou programy na PC (platforma Win32), které zajišťují vzdálenou vizualizaci dat připojeného řídicího systému. Pro PC s monitorem je určen obecný program TheKing. Pro PC s dotykovým LCD panelem je určen program TheQueen. Na přání zákazníka mohou být vytvořeny další podobné programy (v současné době např. TKingCon), které z výše uvedených vycházejí. Přenos dat mezi řídicím systémem a programem PC je zpravidla po sériové komunikační lince RS232 eventuelně i po RS485, Ethernet, modem apod.

Programy jsou schopny zobrazovat přijímaná data jednoduchého typu v uživatelsky vytvořených obrazovkách, dále strukturované záznamy archivů, komunikovat nestrukturované binární datové bloky a v neposlední řadě i nahrávat do řídicího systému nové verze aplikací. Při zobrazování záznamů archivů se použijí informace načtené po komunikaci z řídicího systému, jež definují způsob zobrazení. Tyto informace jsou nazvány popisovači a jejich

podrobný popis najdete v samostatném manuálu „Archivy“. Záznamy archivů je možno zobrazovat i formou grafů.

Knihovny pro vzdálenou vizualizaci dat je balík knihoven KitKing, který je součástí systémových knihoven LIB. Tento balík slouží pro implementaci komunikačního rozhranní v řídicím systému, které komunikuje s programem TheKing po sériové lince. Knihovny tvoří přímou nadstavbu objektům archivu odvozeným od tChArchiveVirt a jsou popsány v samostatném manuálu.

Objekt komunikující s programem TheKing je objekt tWinTickArchive z balíku knihoven KitKing. Tento objekt je popsán v samostatném manuálu popisujícím tyto knihovny.

Aplikace LOADER je malá aplikace nahraná do řídicího systému, která zajišťuje komfortnější nahrávání vlastní řídicí aplikace do řídicího systému, než jaký umožňuje RTD přes Bios Monitor. V řídicím systému je poté nahrán BIOS, LOADER a vlastní řídicí aplikace. Nový Loader lze nahrát pomocí RTD nebo formou speciální aplikace (PatchLdr), která po spuštění přepíše starý Loader novým a spustí ho. Mohou existovat různé aplikace LOADER podle potřeb a možností jak nahrávat řídicí aplikaci do systému. Například pro možnost nahrávání řídicí aplikace z programu TheKing byl vytvořen balík knihoven LdrLib, který je součástí systémových knihoven LIB. Pomocí tohoto balíku lze jednoduše vytvořit vlastní aplikaci LOADER pro potřeby konkrétního řídicího systému, která komunikuje s programem TheKing.

Tabulka modulů je vyhrazená část paměti EEPROM aplikace LOADER komunikující s programem TheKing, ve které jsou uloženy popisy jednotlivých modulů řídicí aplikace. Tato tabulka se vytváří při zápisu nové řídicí aplikace do systému.

Watch Dog (**WD**) je výraz pro algoritmus, který provádí hlídání chodu procesoru nebo běh aplikace či její části. WD může být řešen buď jako HW součástka nebo jako SW algoritmus.