Crc16

JEDNOTKA PRO VÝPOČET CYKLICKÉHO REDUNDANTNÍHO KÓDU CRC 16

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 2.00

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 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: 03.06.2003
Datum posledního uložení dokumentu: 03.06.2003
(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	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	We		První vydání.
1.10	1.XX	Tu	22.05.2003	Úprava dokumentu dle ISO9000.
2.00	2.XX	Tu	03.06.2003	Doplněné funkce CalculateCrc a CheckCrc.

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky pro výpočet cyklického redundantního kódu CRC 16.

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

Jednotka **CRC16** je navržena pro vypočtení cyklického redundantního kódu CRC 16 z bloku dat o libovolné délce. Tento kód slouží pro detekci integrity bloku dat.

4. Popis konstant a typů

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

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

5. Popis objektu CRC 16

Objekt **tCrc16** je jediným objektem této jednotky.

```
pCrc16 = ^ tCrc16;
tCrc16 = object(tObject)
```

Objekt se používá takto:
1. Objekt vytvoříme konstruktorem.
2. Pomocí metody SetResidue vynulujeme hodnotu Residue.
3. Celý blok dat proženeme v pevném pořadí po bytech metodou MakeCrc.
4. Pomocí metody GetResidue vyzdvihneme z Residue spočítaný CRC celého bloku dat.

5.1. Pole

```
Residue : Word;
```

Residue je proměnná ve které se uchovávají zbytky po dělení generujícím polynomem. Zde se také nachází výsledný poslední zbytek a tedy kýžený CRC.

5.2. Metody

```
constructor Init;
```

Construktor **Init** vytvoří objekt.

```
procedure SetResidue(Res: Word);
```

SetResidue přiřazuje Res do proměnné Residue a tím nastavuje hodnotu zbytku. Před začátkem výpočtu se touto metodou Residue nuluje.

```
function GetResidue: Word;
```

GetResidue dodává hodnotu Residue. Na konci výpočtu dodá hodnotu výsledku.

```
procedure MakeCrc(B: Byte);
```

MakeCrc připočte k předchozímu zbytku v proměnné Residue hodnotu zbytku po dělení vytvářecím polynomem. Metoda po bytech počítá hodnotu c(x)=xrd(x) mod

(x16+x15+x2+1). Výpočet polynomu probíhá v jednom průchodu výpočetním algoritmem. Nepoužívá se cyklus. Nejnižší bit vstupujícího byte **B** se účastní první výpočtu cyklického polynomu.

```
function  CalculateCrc(Block:Pointer; Size:Word; Res:Word):Word;
```

Funkci **CalculateCrc** je vhodné použít při výpočtu CRC souvislého bloku dat. Ukazatel **Block** definuje začátek bloku paměti a parametr **Size** udává délku dat v bytech, pro které bude vypočtena hodnota CRC. Parametrem **Res** je možné nastavit počáteční velikost proměnné **Residue**. Uvnitř funkce je opakovaně volána procedura **MakeCrc**.

```
function CheckCrc(Block:Pointer; size:Word; Res:Word;
                  Crc:Word):Boolean;
```

Funkce **CheckCrc** spočítá CRC souvislého bloku dat od ukazatele **Block** o délce **Size**. Výslednou hodnotu CRC porovná s zadaným parametrem **Crc**. Pokud se obě hodnoty shodují (CRC je správně), vrátí funkce hodnotu TRUE. V opačném případě vrátí hodnotu FALSE.