

Timer

JEDNOTKA PRO ODMĚŘOVÁNÍ ČASOVÝCH INTERVALŮ

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: 18.06.2004

Datum posledního uložení dokumentu: 18.06.2004

(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.	Popis konstant a typů	6
5.	Objekt tTimer	6
5.1.	Metody	7
5.1.1.	SaveTime procedura	7
5.1.2.	TstTime funkce	7
5.1.3.	ReadTime funkce	7
5.1.4.	SetCount procedura	7
5.1.5.	TstCount funkce	7
5.1.6.	ReadCount funkce	7
6.	Příklad	8

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Wi		První vydání.
1.10	2.XX	Tu	22.05.2003	Úprava dokumentu dle ISO9000.
1.20	2.XX	Wil	18.06.2004	Úprava popisu dle změn v jednotce HWsyst. Oprava popisu metody ReadCount.

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky pro odměřování časových intervalů.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není nezbytně nutné číst žádný další manuál, ale pro pochopení funkce se doporučuje seznámit s manuály „HWsyst“ popřípadě „Tick“. Dále 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

Jednotka **Timer** definuje objekt **tTimer**, jehož instance slouží k odměřování časových intervalů. Pro měření času používá služby BIOSu **Int 1Ah**, která dodává počet přerušení systémového časovače (**Int 08h**) od spuštění systému. Předpokládá se, že perioda přerušení Int 08h je **ActIRQTime** = 55 ms. Pokud se změní perioda volání obsluhy přerušení Int 08h, která provádí čítání počtů přerušení systémového časovače, musí se změnit patřičně i proměnná ActIRQTime, aby jednotka **Timer** mohla časy vyhodnocovat správně. Toto se děje zejména při použití knihovny **Tick** (funkce **SetTimeUser1**) pro obsluhu uživatelských systémových časovačů, přičemž jednotka Tick byla upravena tak, že korekci proměnné ActIRQTime provádí automaticky. Jednotka Timer používá pro svou činnost konstanty a proměnné z jednotky **HWSyst** (např. již zmíněná proměnná ActIRQTime).

Pozn1: Na různých procesorech není perioda přerušení systémového časovače Int 08h přesně 55ms. Například na KitV40 je zhruba 54,926ms. Proto při měření časů pomocí objektu **tTimer** může vzniknout drobná odchylka.

Pozn2: Jelikož čítač počtu přerušení systémového časovače Int 08h se jednou za den nuluje, lze objekt **tTimer** použít pro měření časových intervalů kratších než jeden den.

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. Objekt tTimer

```
pTimer = ^tTimer;  
tTimer = object(TObject)
```

Objekt **tTimer** definuje metody, které zajišťují odměřování časových intervalů. Metody lze rozdělit podle způsobu použití do dvou skupin. Jednu skupinu tvoří metody **SaveTime**, **TstTime** a **ReadTime** a druhou skupinu tvoří metody **SetCount**, **TstCount** a **ReadCount**. Příklad použití metod z první skupiny: Voláním metody **SaveTime** je zaznamenána hodnota aktuálního času a periodickým voláním metody **TstTime** se otestuje, zda požadovaný interval již uplynul, popřípadě voláním metody **ReadTime** se zjistí již uplynulý čas od zaznamenání. Příklad použití metod z druhé skupiny: Voláním metody **SetCount** je zaznamenána hodnota aktuálního času a zároveň je nastaven požadovaný interval. Dále periodickým voláním metody **TstCount** se otestuje, zda požadovaný interval již uplynul, popřípadě voláním metody **ReadCount** se zjistí již uplynulý čas od zaznamenání. Metody obou skupin používají pro svou činnost odlišné lokální proměnné, proto je lze v aplikaci používat současně ve více odměřovacích algoritmech. V jednom odměřovacím algoritmu však musí být použity metody jen jedné skupiny.

5.1. Metody

5.1.1. SaveTime procedura

```
procedure SaveTime;
```

Metoda **SaveTime** zaznamená hodnotu aktuálního času. Hodnota takto zaznamenaného času je poté využívána např. metodou **TstTime**.

5.1.2. TstTime funkce

```
function TstTime(Ms:LongInt): Boolean;
```

Metoda **TstTime** provádí test, zda uplynulo **Ms** milisekund od času zaznamenaného metodou **SaveTime**. Je-li časový interval od volání metody **SaveTime** delší nebo roven **Ms**, vrátí metoda **TstTime** hodnotu True, v opačném případě vrátí hodnotu False. Hodnota parametru Ms by neměla být delší než jeden den (tj. 3600*24*1000).

5.1.3. ReadTime funkce

```
function ReadTime: Longint;
```

Metoda **ReadTime** vrátí počet milisekund, které uběhly od zaznamenaného času metodou **SaveTime**.

5.1.4. SetCount procedura

```
procedure SetCount(Ms:Longint);
```

Metoda **SetCount** zaznamená hodnotu aktuálního času a zároveň provede nastavení časového intervalu na hodnotu **Ms** milisekund. Hodnota takto zaznamenaného času a nastavený interval je poté využívána např. metodou **TstCount**. Hodnota parametru Ms by neměla být delší než jeden den (tj. 3600*24*1000).

5.1.5. TstCount funkce

```
function TstCount: Boolean;
```

Metoda **TstCount** provádí test, zda nastavený interval od času zaznamenaného metodou **SetCount** již uplynul. Je-li časový interval od volání metody **SetCount** delší nebo roven nastavenému intervalu, vrátí metoda **TstCount** hodnotu True, v opačném případě vrátí hodnotu False.

5.1.6. ReadCount funkce

```
function ReadCount: Longint;
```

Metoda **ReadCount** vrátí počet milisekund, které zbývají do uplynutí zaznamenaného času metodou **SetCount**. Vrátí-li funkce zápornou hodnotu, nastavený čas již uběhl.

6. Příklad

Ukázkový program vytvoří instanci objektu tTimer pro odměřování časových intervalů. Barvou atributu jsou označovány hodnoty intervalů, které již uplynuly.

```
uses
  Crt,
  Timer;
var
  Tm : tTimer;
  i   : LongInt;
  y,x : Word;
begin
  WriteLn;
  WriteLn('Test jednotky Timer. ');
  WriteLn;
  x:=WhereX;
  y:=WhereY;
  with Tm do
    repeat
      SaveTime;
      repeat
        GotoXY(x,y);
        TextAttr:=LightGray;
        Write('Cas v milisec = ');
        for i:=0 to 10 do
          begin
            if TstTime(i*1000) then
              TextAttr:=LightRed
            else
              TextAttr:=LightBlue;
            Write(i*100,' ');
          end;
          Delay(500);
        until TstTime(11000);
      until KeyPressed;
      TextAttr:=LightGray;
      ReadLn;
    end.
```