

# uFilter

## JEDNOTKA DIGITÁLNÍCH FILTRŮ

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: 16.05.2003

Datum posledního uložení dokumentu: 16.05.2003

(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	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
3.1. Filtr na odstranění zákmitů	6
3.2. Sumarizační filtr	6
4.Popis konstant a typů	6
5.Popis objektů	7
5.1. tFiltr	7
5.1.1. tFilterBit	7
5.1.1.1. FSetParam	7
5.1.1.2. FGetParam	8
5.1.1.3. FreadVAL	8
5.1.2. tFilterByte	8
5.1.2.1. FSetParam	8
5.1.2.2. FGetParam	8
5.1.2.3. FreadVAL	8
5.1.3. tFiltrWord	8
5.1.3.1. FSetParam	8
5.1.3.2. FGetParam	9
5.1.3.3. FreadVAL	9
5.1.4. tFilterInteger	9
5.1.4.1. FSetParam	9
5.1.4.2. FGetParam	9
5.1.4.3. FReadVAL	9
5.1.5. tFilterLongint	9
5.1.5.1. FSetParam	9
5.1.5.2. FGetParam	10
5.1.5.3. FreadVAL	10
5.2. tFilterSuma	10
5.2.1. tFilterSumaByte	11
5.2.1.1. FSetParam	11
5.2.1.2. FGetParam	11
5.2.1.3. FSetWrite	11
5.2.1.4. FreadVAL	11
5.2.2. tFilterSumaWord	12
5.2.2.1. FSetParam	12
5.2.2.2. FGetParam	12
5.2.2.3. FSetWrite	12
5.2.2.4. FreadVAL	12
5.2.3. tFilterSumaInteger	12
5.2.3.1. FSetParam	13
5.2.3.2. FGetParam	13
5.2.3.3. FSetWrite	13
5.2.3.4. FreadVAL	13

5.2.4.	tFilterSumaLongint	13
5.2.4.1.	FSetParam	14
5.2.4.2.	FGetParam	14
5.2.4.3.	FSetWrite	14
5.2.4.4.	FreadVAL	14
5.2.5.	tFilterSumaReal	14
5.2.5.1.	FSetParam	14
5.2.5.2.	FGetParam	15
5.2.5.3.	FSetWrite	15
5.2.5.4.	FreadVAL	15
6.	Příklady	15
6.1.	Vychytávání zákmitů	15
6.2.	Sumarizační filtr	15

## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Su		První vydání.
1.10	1.XX	Tu	16.05.2003	Úprava dokumentu dle ISO9000.

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis jednotky digitálních filtrů.

### 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 orientovat se 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 uFilter je jednotka digitálních filtrů. Obsahuje dva typy filtrů. Je to filtr pro vyloučení krátkodobých zákmitů signálu měnícího se skokem a sumarizační filtr pro počítání plovoucího průměru.

#### 3.1. Filtr na odstranění zákmitů

---

Tento filtr změni svou výstupní hodnotu až po načtení CT konstantních hodnot ze vstupu. Dokud toto množství nenačte, dává na výstup svou poslední načtenou vnitřní hodnotu. Filtr může potlačovat zakmitávání signálu vlivem nežádoucích přechodových jevů. Dá se použít například pro odstranění poruch vzniklých zákmitu kontaktu membránové klávesnice.

Jednotlivé Filtry tohoto typu obsažené v jednotce pracují se vstupními hodnotami typu BIT, BYTE, WORD, INTEGER, LONGINT a na výstup vrací hodnotu stejného typu.

Doba prodlevy filtru CT a jeho vnitřní hodnota je nastavitelná pomocí stringu s parametry.

#### 3.2. Sumarizační filtr

---

Sumarizační filtr slouží k vytvoření plovoucí průměrné hodnoty ze vstupních hodnot, které na filtr postupně přicházejí. Filtr počítá vážený průměr své vnitřní hodnoty a nově příšlého údaje. Na výstupu filtru se objevuje jeho vnitřní hodnota. Nastavením váhy nového vzorku se mění rychlost odezvy výstupu filtru na změnu vstupního signálu. Součet vah nového vzorku a vnitřní hodnoty je jedna.

Jednotlivé Filtry tohoto typu obsažené v jednotce pracují se vstupními hodnotami typu BYTE, WORD, INTEGER, LONGINT, REAL a na výstup vrací hodnoty stejného typu.

### 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.

Jednotka definuje konstanty pro omezení maximálních LongInt čísel zpracovávaných filtrem tFilterSumaLongint.

```
MaxLong = 20000000;  
MinLong = -20000000;
```

## 5. Popis objektů

---

Jednotka **uFilter** definuje dva rodičovské objekty pro tvoření sumarizačního filtru a filtru pro odstranění zákmitů:

**tFilter** je typ objektu filtru na vychytání zákmitů .

**tFilterSuma** je typ objektu sumarizačního filtru.

### 5.1. tFilter

---

Virtuální objekt **tFilter** je dědicem Standardního **tObject** a je vytvořen pouze jako rodičovský objekt pro objekty reálných filtrů. V této jednotce to jsou: **tFilterBit**, **tFilterByte**, **tFilterWord**, **tFilterInteger**, **tFilterLongint**.

Objekt tFilter obsahuje proměnné:

```
fCt   : Word;      {počet konstantních vstupních vzorků,
                  aby se dostaly na výstup}
fC    : Word;      {počítadlo vstupních vzorků}
fF    : Boolean;   {vnitřní obsah filtru typu bit}
fByte : Byte;      {vnitřní obsah filtru typu byte}
fWord : Word;      {vnitřní obsah filtru typu word}
fInte : Integer;  {vnitřní obsah filtru typu Integer}
fLong : Longint;  {vnitřní obsah filtru typu LongInt}
```

Objekt obsahuje tyto metody:

```
constructor Init;
function FGetParam:tParamStr; virtual; {dodání parametrů }
```

Metoda **FGetParam** je prázdná.

```
procedure FSetParam(S: tParamStr); virtual; {nastavení parametrů}
```

Metoda **FSetParam** nastavuje proměnné objektu podle inicializačního stringu, který se jí předává jako parametr. String musí vyhovovat konvencím pro dekódování podle jednotky uString. Obsahuje klíčová slova, rovnítko a obsah proměnné:

CT	počet vstupních konstantních hodnot, které se projeví na výstupu filtru.
IBI	Nastavuje inicializační hodnotu filtru typu Bit.
IBY	Nastavuje inicializační hodnotu filtru typu Byte.
IWO	Nastavuje inicializační hodnotu filtru typu Word.
IIN	Nastavuje inicializační hodnotu filtru typu Integer.
ILO	Nastavuje inicializační hodnotu filtru typu Longint.

#### 5.1.1. tFilterBit

---

Objekt tFilterBit je dědicem objektu tFilter a pracuje se vstupní a výstupní proměnnou typu boolean, tedy vlastně s jedním bitem. Oproti rodičovskému objektu obsahuje navíc metody FGetParam a FreadVal. Metodu FSetParam objekt dědí.

##### 5.1.1.1. FSetParam

```
procedura FSetParam(S:tParamStr);virtual;
```

Metoda nastavuje hodnotu parametru CT = Byte a IBI = Boolean. Parametr CT nastavuje, kolik konstantních vstupních hodnot musí být zadáno, aby se vstupní hodnota projevila na výstupu tohoto filtru. Parametr IBI je počáteční nastavení výstupní hodnoty tohoto filtru. Standardní nastavení parametru je 'CT=0 IBI=False'

#### 5.1.1.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru CT = Byte a IBI = Boolean.

#### 5.1.1.3. FreadVAL

function FReadVal(F:Boolean):Boolean;virtual;

Vstupní hodnota této funkce se na jejím výstupu projeví, až po zadaném CT počtu zavolání této funkce s konstantní vstupní hodnotou. Počáteční výstupní hodnota se nastavuje parametrem IBI. Vstup a výstup je typu Boolean.

### 5.1.2. tFilterByte

---

Objekt tFilterByte je dědicem objektu tFilter a pracuje se vstupní a výstupní proměnnou typu byte. Oproti rodičovskému objektu obsahuje navíc metody FGetParam a FreadVal. Metodu FSetParam objekt dědí.

#### 5.1.2.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Nastaví hodnotu parametru CT = Byte a IBY = Byte. Parametr CT nastavuje, kolik konstantních vstupních hodnot musí být zadáno, aby se vstupní hodnota projevila na výstupu tohoto filtru. Parametr IBY je počáteční nastavení výstupní hodnoty tohoto filtru. Standardní nastavení parametru je 'CT=0 IBY=0'

#### 5.1.2.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru CT = Byte a IBY = Byte.

#### 5.1.2.3. FreadVAL

function FReadVal(B:Byte):Byte;virtual;

Vstupní hodnota této funkce se na jejím výstupu projeví, až po zadaném CT počtu zavolání této funkce s konstantní vstupní hodnotou. Počáteční výstupní hodnota se nastavuje parametrem IBY. Vstup a výstup je typu Byte.

### 5.1.3. tFiltrWord

---

Objekt tFiltrWord je dědicem objektu tFiltr a pracuje se vstupní a výstupní proměnnou typu Word. Oproti rodičovskému objektu obsahuje navíc metody FGetParam a FreadVal. Metodu FSetParam objekt dědí.

#### 5.1.3.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Metoda nastavuje hodnotu parametru CT = Byte a IWO = Word. Parametr CT nastavuje, kolik konstantních vstupních hodnot musí být zadáno, aby se vstupní



hodnota projevila na výstupu tohoto filtru. Parametr IWO je počáteční nastavení výstupní hodnoty tohoto filtru. Standardní nastavení parametru je 'CT=0 IWO=0'.

### 5.1.3.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru CT = Byte a IWO = Word.

### 5.1.3.3. FreadVAL

function FReadVal(W:Word):Word;virtual;

Vstupní hodnota této funkce se na jejím výstupu projeví, až po zadaném CT počtu zavolání této funkce s konstantní vstupní hodnotou. Počáteční výstupní hodnota se nastavuje parametrem IWO. Vstup a výstup je typu Word.

## 5.1.4. tFilterInteger

---

Objekt tFilterInteger je dědicem objektu tFilter a pracuje se vstupní a výstupní proměnnou typu Integer. Oproti rodičovskému objektu obsahuje navíc metody FGetParam a FreadVal. Metodu FSetParam objekt dědí.

### 5.1.4.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Metoda nastavuje hodnotu parametru CT = Byte a IIN = Integer. Parametr CT nastavuje, kolik konstantních vstupních hodnot musí být zadáno, aby se vstupní hodnota projevila na výstupu tohoto filtru. Parametr IIN je počáteční nastavení výstupní hodnoty tohoto filtru. Standardní nastavení parametru je 'CT=0 IIN=0'.

### 5.1.4.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru CT = Byte a IIN = Integer.

### 5.1.4.3. FreadVAL

function FReadVal(I:Integer):Integer;virtual;

Vstupní hodnota této funkce se na jejím výstupu projeví, až po zadaném CT počtu zavolání této funkce s konstantní vstupní hodnotou. Počáteční výstupní hodnota se nastavuje parametrem IIN. Vstup a výstup je typu Integer.

## 5.1.5. tFilterLongint

---

Objekt tFilterLongint je dědicem objektu tFilter a pracuje se vstupní a výstupní proměnnou typu LongInt. Oproti rodičovskému objektu obsahuje navíc metody FGetParam a FreadVal. Metodu FSetParam objekt dědí.

### 5.1.5.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Metoda nastavuje hodnotu parametru CT = Byte a ILO = Longint. Parametr CT nastavuje, kolik konstantních vstupních hodnot musí být zadáno, aby se vstupní hodnota projevila na výstupu tohoto filtru. Parametr ILO je počáteční nastavení výstupní hodnoty tohoto filtru. Standardní nastavení parametru je 'CT=0 ILO=0'.

### 5.1.5.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru CT = Byte a ILO = Longint.

### 5.1.5.3. FreadVAL

function FReadVal(L:Longint):Longint;virtual;

Vstupní hodnota této funkce se na jejím výstupu projeví, až po zadaném CT počtu zavolání této funkce s konstantní vstupní hodnotou. Počáteční výstupní hodnota se nastavuje parametrem ILO. Vstup a výstup je typu Longint.

## 5.2. tFilterSuma

Objekt je dědicem Standardního **TObject**. Je to virtuální rodičovský objekt pro konstrukci reálných sumačních filtrů, pracujících nad různými vstupními a výstupními typy. V jednotce uFiltr to jsou objekty : **tFilterSumaByte**, **tFilterSumaWord**, **tFilterSumaInteger**, **tFilterSumaLongint**, **tFilterSumaReal**.

Objekt tFilterSuma obsahuje proměnnou **fMReal** :REAL, ve které se uschovává jeho vnitřní hodnota. Dále objekt obsahuje konstruktor **Init** a dvě prázdné metody:

```
procedure FSetParam(S: tParamStr); virtual;
function FGetParam:tParamStr; virtual;
```

Výpočet filtru je implementován až u dědiců objektu tFiltSuma.

U filtru **tFilterSumaReal** probíhá výpočet podle Standardního vzorce pro tento typ filtru :

$$\mathbf{V_{n+1}}=\mathbf{M*V_{in}}+(1-\mathbf{M})*\mathbf{V_n} .$$

Kde **M** je váhová konstanta filtru v intervalu (0;1). Tato konstanta se uschovává ve vnitřní proměnné objektu a dá se nastavit příslušným parametrem. **Vin** je vstupní hodnota zadávaná filtru. **Vn** je vnitřní hodnota filtru, je to vlastně předchozí výstupní hodnota, uschovává se ve vnitřní proměnné objektu. **Vn+1** je výsledná výstupní hodnota sumarizačního filtru, která se stává novou vnitřní hodnotou filtru.

U ostatních typu filtrů probíhá výpočet podle upraveného vzorce filtru:

$$\mathbf{V_{n+1}}=((\mathbf{M*cMul*V_{in}})+((\mathbf{cMUL}-(\mathbf{M*cMul}))*\mathbf{V_n})) \text{ div } \mathbf{cMul} .$$

Kde **M** je váhová konstanta filtru v intervalu (0;1). Tato konstanta se uschovává ve vnitřní proměnné objektu. **cMul** je přesnost filtru - určuje s jakou přesností, neboli na kolik desetinných míst bude během výpočtu zadaná konstanta **M** zaokrouhlena. Implicitně nastavená hodnota je 100, to způsobí, že konstanta **M** bude zaokrouhlena na dvě desetinná místa.

**Vin** je vstupní hodnota zadávaná filtru. **Vn** je předchozí výstupní hodnota filtru, uschovává se ve vnitřní proměnné objektu. **Vn+1** je výsledná hodnota sumarizačního filtru.

Vzorec filtru byl takto upraven proto, aby se mohly při výpočtech používat celočíselné operace, které jsou rychlejší než operace s typem Real. Konstanty **M** a

**cMul** jsou po zadání nejdřív vynásobeny a zaokrouhleny na celé číslo a s ním se teprve pracuje v celočíselných operacích. Z toho plyne, že se zpracovává počet desetinných míst odpovídající velikosti zadané konstanty **M**. Při implicitním nastavení konstanty **cMul** na 100, je konstanta **M** zaokrouhlována na dvě desetinná místa.

### 5.2.1. tFilterSumaByte

Objekt pracuje nad proměnnými typu Byte. Objekt obsahuje proměnné:

```
fByte : Byte;      {vnitřní hodnota filtru}
fM : Byte;      {konstanta M Sumarizačního filtru}
cMul : Byte;      {konstanta cMul Sumarizačního filtru}
```

Objekt obsahuje nové metody:

```
constructor Init
procedure FSetParam(S: tParamStr); virtual;      {nastavení proměnných
objektu}
function FGetParam:tParamStr;virtual;          {čtení proměnných objektu}
procedure FSetWrite(B:Byte);virtual;           {nastaví aktuální hodnotu
filtru}
function FReadVal(B:Byte):Byte; virtual;       {nový vzorek do filtru}
```

#### 5.2.1.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Metoda nastavuje proměnné objektu podle inicializačního stringu, který se jí předává jako parametr. String musí vyhovovat konvencím pro dekódování podle jednotky uString. Obsahuje klíčová slova, rovnítko a obsah proměnné:

**IBY** Nastavuje inicializační vnitřní hodnotu filtru fByte. Je to předcházející výstupní hodnota filtru.

**MUL** Nastavuje konstantu filtru **cMul**, která určuje přesnost výpočtu filtru. Pro tento typ filtru jsou přípustné hodnoty konstanty MUL: 1,10,100.

**M** Nastavuje konstantu filtru, reálné číslo v intervalu (0;1)

Standardní nastavení parametru je 'M=0,01 MUL=100 IBY=0'

#### 5.2.1.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru M=Real, MUL=Byte, IBY=Byte.

#### 5.2.1.3. FSetWrite

procedure FSetWrite(B:Byte);virtual;

Tato procedura nastaví počáteční hodnotu sumarizačního filtru stejně jako při použití parametru IBY. Vstupní hodnota pro počáteční nastavení filtru je typu Byte.

#### 5.2.1.4. FreadVAL

function FReadVal(B:Byte):Byte;Virtual;

Tato funkce sumarizačního filtru ze vstupní hodnoty a zadané konstanty M vypočítá s použitím vzorce pro sumarizační filtr výstupní hodnotu filtru, kterou vrátí. Počáteční nastavení výstupní hodnoty funkce se zadává parametrem IBY. Vstup a výstup je typu Byte.

## 5.2.2. tFilterSumaWord

---

Objekt pracuje nad proměnnými typu Word. Objekt obsahuje proměnné:

```
fByte : Word;      {vnitřní hodnota filtru}
fM : LongInt;     {konstanta M Sumarizacního filtru}
cMul : Word;      {konstanta cMul Sumarizacního filtru}
```

Objekt obsahuje nové metody:

```
constructor Init
procedure FSetParam(S: tParamStr); virtual;      {nastavení proměnných
objektu}
function FGetParam:tParamStr;virtual;          {čtení proměnných objektu}
procedure FSetWrite(B: Word);virtual;          {nastaví aktuální hodnotu
filtru}
function FReadVal(B: Word): Word; virtual;      { nový vzorek do
filtru}
```

### 5.2.2.1. FSetParam

```
procedura FSetParam(S:tParamStr);virtual;
```

Metoda nastavuje proměnné objektu podle inicializačního stringu, který se jí předává jako parametr. String musí vyhovovat konvencím pro dekódování podle jednotky uString. Obsahuje klíčová slova, rovnítko a obsah proměnné:

**M** nastavuje konstantu filtru, reálné číslo v intervalu od 0 do 1.

**MUL** nastavuje konstantu filtru **cMul**, která určuje přesnost výpočtu filtru. Pro tento typ filtru jsou přípustné hodnoty konstanty: 1,10,100,1000,10000. Implicitní nastavení je 100, tj 2 desetinná místa.

**IWO** nastavuje počáteční hodnotu filtru. Používá se pro výpočet následující výstupní hodnoty filtru. Je to vlastně předcházející výstupní hodnota filtru.

Standardní nastavení parametru je 'M=0,01 MUL=100 IWO=0'

### 5.2.2.2. FGetParam

```
procedura FGetParam:tParamStr;virtual;
```

Metoda vrátí nastavenou hodnotu parametru M=Real, MUL=Byte, IWO=Word.

### 5.2.2.3. FSetWrite

```
procedure FSetWrite(W:Word);virtual;
```

Tato procedura nastaví počáteční hodnotu sumarizačního filtru stejně jako při použití parametru IWO. Vstupní hodnota pro počáteční nastavení filtru je typu Word.

### 5.2.2.4. FreadVAL

```
function FReadVal(W:Word):Word;Virtual;
```

Tato funkce sumarizačního filtru ze vstupní hodnoty a zadané konstanty M vypočítá výstupní hodnotu filtru, kterou vrátí. Počáteční nastavení výstupní hodnoty funkce se zadává parametrem IWO. Vstup a výstup je typu Word.

## 5.2.3. tFilterSumaInteger

---

Objekt pracuje nad proměnnými typu Word. Objekt obsahuje proměnné:

```
fInte : Integer;
```

```
fM : Longint; {konstanta M Sumarizacniho filtru}
cMul : Word; {konstanta c Mul Sumarizacniho filtru}
```

Objekt obsahuje metody:

```
constructor Init;
procedure FSetParam(S: tParamStr); virtual;
function FGetParam:tParamStr;virtual;
procedure FSetWrite(I:Integer);virtual;
function FReadVal(I:Integer):Integer; virtual;
```

### 5.2.3.1. FSetParam

```
procedura FSetParam(S:tParamStr);virtual;
```

Metoda nastavuje proměnné objektu podle inicializačního stringu, který se jí předává jako parametr. String musí vyhovovat konvencím pro dekódování podle jednotky uString. Obsahuje klíčová slova, rovnítko a obsah proměnné:

**M** - Konstanta filtru - reálné číslo v intervalu od 0 do 1.

**MUL** - Konstanta filtru **cMul**, která určuje přesnost výpočtu filtru. Pro tento typ filtru jsou přípustné hodnoty konstanty: 1,10,100,1000,10000. Implicitní nastavení je 100, tj 2 desetinná místa.

**IIN** - počáteční nastavení filtru. Používá se pro výpočet následující výstupní hodnoty filtru. Je to vlastně předcházející výstupní hodnota filtru.

Standardní nastavení parametru je 'M=0,01 MUL=100 IIN=0'

### 5.2.3.2. FGetParam

```
procedura FGetParam:tParamStr;virtual;
```

Metoda vrátí nastavenou hodnotu parametru M=Real, MUL=Byte, IIN=Integer.

### 5.2.3.3. FSetWrite

```
procedure FSetWrite(I:Integer);virtual;
```

Tato procedura nastaví počáteční hodnotu sumarizačního filtru stejně jako při použití parametru IIN. Vstupní hodnota pro počáteční nastavení filtru je typu Integer.

### 5.2.3.4. FreadVAL

```
function FReadVal(I:Integer):Integer;Virtual;
```

Tato funkce sumarizačního filtru ze vstupní hodnoty a zadané konstanty M vypočítá výstupní hodnotu filtru, kterou vrátí. Počáteční nastavení výstupní hodnoty funkce se zadává parametrem IIN. Vstup a výstup je typu Integer.

## 5.2.4. tFilterSumaLongint

---

Objekt pracuje nad proměnnými typu Word. Objekt obsahuje proměnné:

```
fLong : Longint;
fM : Byte; {konstanta M Sumarizacniho filtru}
cMul : Byte; {konstanta cMul Sumarizacniho filtru}
```

Objekt obsahuje nové metody:

```
constructor Init;
procedure FSetParam(S: tParamStr); virtual;
function FGetParam:tParamStr;virtual;
procedure FSetWrite(L:Longint);virtual;
function FReadVal(L:Longint):Longint; virtual;
```

### 5.2.4.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Metoda nastavuje proměnné objektu podle inicializačního stringu, který se jí předává jako parametr. String musí vyhovovat konvencím pro dekódování podle jednotky uString. Obsahuje klíčová slova, rovnítko a obsah proměnné:

**M** - Konstanta filtru - reálné číslo v intervalu od 0 do 1.

**MUL** - Konstanta filtru **cMul**, která určuje přesnost výpočtu filtru. Pro tento typ filtru jsou přípustné hodnoty konstanty: 1,10,100. Implicitní nastavení je 100, tj 2 desetinná místa.

**ILO** - počáteční nastavení filtru. Používá se pro výpočet následující výstupní hodnoty filtru. Je to vlastně předcházející výstupní hodnota filtru.

Standardní nastavení parametru je 'M=0,01 MUL=100 ILO=0'

### 5.2.4.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru M=Real, MUL=Byte, ILO = Longint.

### 5.2.4.3. FSetWrite

procedure FSetWrite(L:Longint);virtual;

Tato procedura nastaví počáteční hodnotu sumarizačního filtru stejně jako při použití parametru ILO. Vstupní hodnota pro počáteční nastavení filtru je typu Longint.

### 5.2.4.4. FreadVAL

function FReadVal(L:Longint):Longint;Virtual;

Tato funkce sumarizačního filtru ze vstupní hodnoty a zadané konstanty M vypočítá výstupní hodnotu filtru, kterou vrátí. Počáteční nastavení výstupní hodnoty funkce se zadává parametrem IBY. Vstup a výstup je typu Longint.

**Pozor, Maximální** možná vstupní hodnota pro zpracování je 20 000 000 viz konstanta **MaxLong**. **Minimální** možná vstupní hodnota pro zpracování je -20 000 000 viz konstanta **MinLong**. Pokud vstupní hodnota překročí tyto meze je vždy vrácena pro větší kladná čísla hodnota 20 000 000 viz konstanta **MaxLong**. Pro větší záporná čísla je vrácena hodnota -20 000 000 viz konstanta **MinLong**.

### 5.2.5. tFilterSumaReal

Objekt pracuje nad proměnnými typu Word. Objekt obsahuje proměnné:

fReal : Real;

fM : Real; {konstanta M Sumarizacniho filtru}

Objekt obsahuje nové metody:

constructor Init;

procedure FSetParam(S: tParamStr); virtual;

function FGetParam:tParamStr;virtual;

procedure FSetWrite(R:Real);virtual;

function FReadVal(R:Real):Real; virtual;

#### 5.2.5.1. FSetParam

procedura FSetParam(S:tParamStr);virtual;

Metoda nastavuje proměnné objektu podle inicializačního stringu, který se jí předává jako parametr. String musí vyhovovat konvencím pro dekodování podle jednotky uString. Obsahuje klíčová slova, rovnítko a obsah proměnné:

**M** - Konstanta filtru, reálné číslo v intervalu od 0 do 1.

**IRE** - počáteční nastavení filtru. Používá se pro výpočet následující výstupní hodnoty filtru. Neboli je to předcházející výstupní hodnota filtru.

Standardní nastavení parametru je 'M=0,01 IRE=0'

### 5.2.5.2. FGetParam

procedura FGetParam:tParamStr;virtual;

Metoda vrátí nastavenou hodnotu parametru M=Real , IRE=Real.

### 5.2.5.3. FSetWrite

procedure FSetWrite(R:Real);virtual;

Tato procedura nastaví počáteční hodnotu sumarizačního filtru stejně jako při použití parametru Real. Vstupní hodnota pro počáteční nastavení filtru je typu Real.

### 5.2.5.4. FreadVAL

function FreadVal(R:Real):Real;Virtual;

Tato funkce sumarizačního filtru ze vstupní hodnoty a zadané konstanty M vypočítá výstupní hodnotu filtru, kterou vrátí. Počáteční nastavení výstupní hodnoty funkce se zadává parametrem IRE. Vstup a výstup je typu Real.

## 6. Příklady

---

### 6.1. Vychytávání zákmitů

---

Tento příklad ukazuje použití filtru na vychytání zákmitů na příkladu filtru **tFiltrByte**.

```
uses uFilter;
var tByte:tFilterByte;
    b:byte;
begin
  with tByte do
    begin
      init; {inicializace objektu tByte}
      FSetParam('CT=5 IBY=65'); {nastaví parametr CT na 5 a IBY na 65}
      b:=tByte.FreadVal(b); {vyvolá metodu objektu tByte.FreadVal}
    end;
end.
```

### 6.2. Sumarizační filtr

---

Tento příklad ukazuje použití sumarizačního filtru na příkladu filtru **tFilterSumaByte**.

```
uses uFilter;
var tSumaByte:tFilterSumaByte;2
    b:byte;
```

```
begin
with tSumaByte do
begin
  init; {inicializace objektu tSumaByte}
  FSetParam('M=0.5 IBY=200 MUL=10'); {nastavi hodnotu parametru:
      M=0.5,IBY=200 MUL=10}
  FSetWrite(85); {Změní hodnotu parametru IBY na 85}
  b:=FreadVal(b); {vyvolá metodu FreadVal objektu tSumaByte}
end;
end.
```