

**FB3 - <offline>**

"FB3:Derivate calc" Derivate calculation  
**Nome:** Derivate **Famiglia:** General  
**Autore:** SimoneS **Versione:** 0.1  
**Versione blocco:** 2  
**Data e ora Codice:** 01/02/00 12:12:35  
**Interfaccia:** 14/04/99 17:39:01  
**Lunghezze (blocco / codice / dati):** 00474 00324 00000

Indirizzo	Dichiarazione	Nome	Tipo	Valore iniziale	Commento
	in				
	out				
0.0	in_out	Input_value	REAL	0.000000e+000	Value to derive
4.0	in_out	Reset	BOOL	FALSE	Reset command
4.1	in_out	Derive_spl_CMD	BOOL	FALSE	Timing step
6.0	in_out	Derive_value	REAL	0.000000e+000	Calculated value
10.0	in_out	Enable	BOOL	FALSE	Calculation finish
12.0	stat	Samples	ARRAY[0..9]		Calculation apply items
*4.0	stat		REAL		
52.0	stat	Nr_samples	INT	0	# of samples (2 - 20)
54.0	stat	Funct_pointer	DINT	L#0	Derive function pointer
58.0	stat	Derive_calc_ON	BOOL	FALSE	Derive calculation start
60.0	stat	Nr_of_shf	INT	0	# of shifted samples
62.0	stat	Loop	INT	0	Loop index
64.0	stat	Index	INT	0	Array index
66.0	stat	Prev_sum	REAL	0.000000e+000	First 10 items sum
70.0	stat	Next_sum	REAL	0.000000e+000	Second 10 items sum
74.0	stat	Dummy_bit_1	BOOL	FALSE	Dummy bit
74.1	stat	Calc_ON	BOOL	FALSE	Calculation ON
74.2	stat	Init_end	BOOL	FALSE	Array init end
	temp				

**Blocco:FB3 1st derive**

This function calculate the 1st derive of [Input\_value]  
The derive is calculate on 20 samples of [Input\_value] the samples are take each  
time [CLK] is TRUE  
The output value is the sum of 1st 10 samples with 2nd 10 samples.

## Parameters:

.[Input\_value]: Value to derive  
.[Reset]: This command reset the act derive value. The calculation restart when  
all the items of samples array are refreshed  
.[Derive\_spl\_CMD]: Timing step. The samples are taken when  
[Derive\_spl\_CMD] = TRUE  
.[Derive\_value]: Derivated output value  
.[Enable]: This output is energized when the calculation of act value is  
finished  
.[Samples]: Samples container.  
.[Nr\_samples]: # of samples to take for calculation  
.[Derive\_calc\_ON]: System variable  
.[Nr\_of\_shf]: System variable  
.[Loop]: System variable  
.[Index]: System variable  
.[Prev\_sum]: System variable  
.[Next\_sum]: System variable  
.[Dummy\_bit\_1]: System variable  
.[Calc\_ON]: System variable  
.[Init\_end]: System variable

Segmento: 1 Reset

This network manages the reset command

```
U   #Reset           // Reset command
R   #Derive_calc_ON
```

Segmento: 2	Timing
This network manage the timing of function	

```

U    #Derive_spl_CMD
BLD  100
FP    #Dummy_bit_1
=    #Calc_ON          // One-shot clock
UN   #Calc_ON          // Timng calculation
R    #Enable
BEB          // End block

```

Segmento: 3	1st derive calculation
This network manages the calculation of [Input_value] 1st derive	

```

// ***** Derive calculation *****
**
//
U    #Derive_calc_ON // **** Reset derive value ****
SPB  Calc
L    0
T    #Derive_value
SPA  Shf          // Jump derive calculation
Calc: L 0
T    #Index       // Init array index
T    #Prev_sum    // Init 1st 10 samples sum
T    #Next_sum    // Init 2nd 10 samples sum
L    5
Lab: T #Loop      // Load loop index
L    #Funct_pointer // Load derive function pointer
LAR1 // Load pointment register
L    P##Samples   // Load array pointer
+AR1 // ADD to pointment register
L    #Index       // Load pointer of array index
SLD  3
+AR1 // Sum the array pointer to inex pointer
L    DID [AR1,P#0.0] // Load pointed sample
L    #Prev_sum
+R
T    #Prev_sum    // Add the act sample to prev sum
L    DID [AR1,P#20.0] // Load pointed sample plus 5
L    #Next_sum
+R
T    #Next_sum    // Add the act sample plus 5 to next sum
L    4
L    #Index
+I          // Array index increment
T    #Index
L    #Loop
LOOP Lab        // Loop end
L    #Prev_sum    // 1st derive calculation
L    #Next_sum
-R
//L    5.000000e+000
//R          // ([Prev_sum]-[Next-sum])/25
T    #Derive_value
//
// ***** 1 item to right array shifting *****
**
//
Shf: L 32
T    #Index       // Load array index whit address of 9th item
L    9
Lab1: T #Loop     // Load loop index
L    #Funct_pointer // Load the pointer to derive function
LAR1 // Load pointment register
L    P##Samples   // Load the pointer of array samples
+AR1 // ADD to pointment register
L    #Index       // Load the pointer of act sample
SLD  3
+AR1 // Sum the pointer of array with the pointer of 19th sample
L    DID [AR1,P#0.0]
T    DID [AR1,P#4.0] // Shift the samples of 1 item to right
L    #Index
L    4
-I          // Array index decrement
T    #Index
L    #Loop

```

```
LOOP Lab1          // Loop end
```

```
Segmento: 4      Add new sample to array
```

```
This network add a new sample to array
```

```
L      #Funct_pointer
LAR1
L      P##Samples
+AR1
L      #Input_value
T      DID [AR1,P#0.0] // Insert new sample to 1st array field
SET
SAVE
S      #Enable
L      #Nr_of_shf      // # of shifted samples calculation
L      1
+I
T      #Nr_of_shf
```

```
Segmento: 5      Array init end calculation start
```

```
This network manages the activation of derive calculation start (when array
initialization is finish)
```

```
L      #Nr_of_shf      // # of shifted samples
L      9
>=I
=      #Init_end
U      #Init_end
S      #Derive_calc_ON // Set derive calculation start
U(
O      #Init_end      // Reset coomand
O      #Reset
)
SPBNB Jump
L      0              // Reset # of shifted samples
T      #Nr_of_shf
Jump:  NOP 0
```