

VARIABLES

VAR_GLOBAL

(* Inputs *)

Di00CPU AT %IX255.0 : BOOL; (* Input 00 modulo CPU *)

Di01CPU AT %IX255.1 : BOOL; (* Input 01 modulo CPU *)

(* Outputs *)

Do00CPU AT %QX255.0 : BOOL; (* Output 00 modulo CPU *)

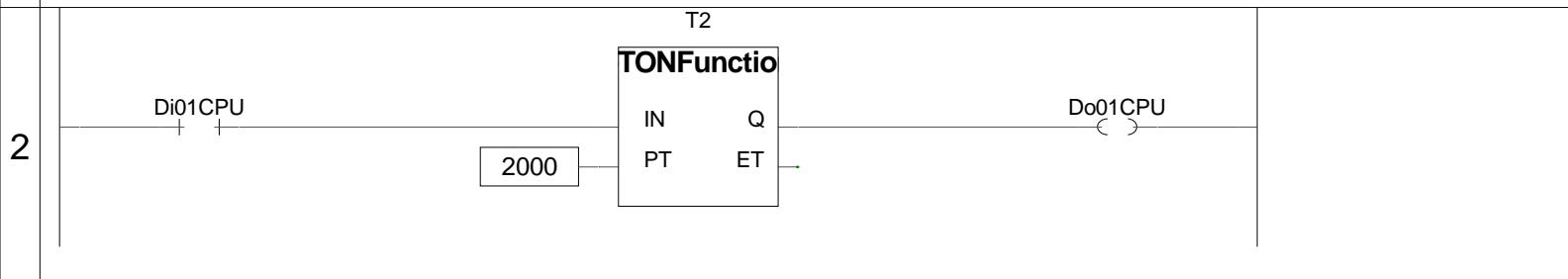
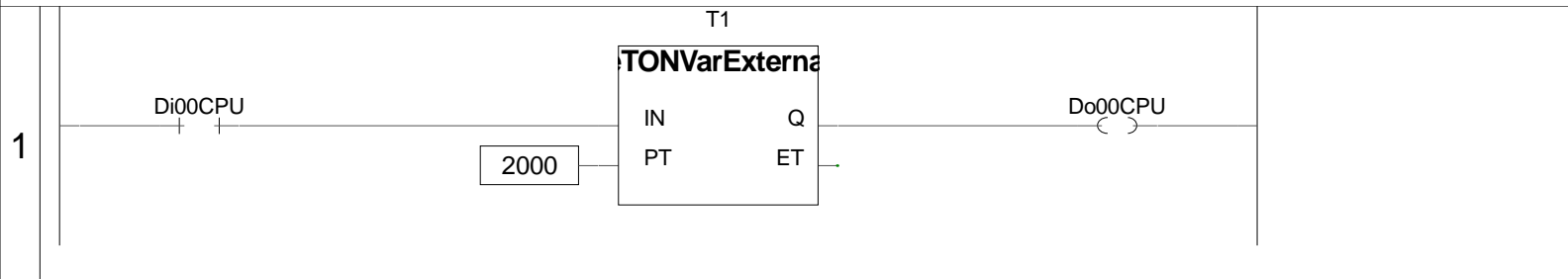
Do01CPU AT %QX255.1 : BOOL; (* Output 01 modulo CPU *)

END_VAR

	Project : FBTimer	
	VARIABLES :	
	Release : FBTimer	Ver :1.00
	Author :	Date:27/05/2015
	Note :	Page:1 of 1

```

VAR
T1 : eTONVarExternal; (* Timer con variabile esterna *)
T2 : eTONFunction; (* Timer con funzione *)
END_VAR
    
```



	Project : FBTimer	
	PROGRAM : LD	
	Release : FBTimer	Ver :1.00
	Author :	Date:27/05/2015
	Note :	Page:1 of 1

```
VAR_INPUT
IN : BOOL; (* Timer input *)
PT : UDINT; (* Preset time value (mS) *)
END_VAR
```

```
VAR_OUTPUT
Q : BOOL; (* Delayed output *)
ET : UDINT; (* Executing time (mS) *)
END_VAR
```

```
VAR_EXTERNAL
SysTime : UDINT; (* System time (mS) *)
END_VAR
```

```
VAR
TimeBf : UDINT; (* Time buffer (mS) *)
END_VAR
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "eTONVarExternal" *)
3 (* ***** *)
4 (* Il blocco funzione esegue temporizzazione uscita su attivazione ingresso. *)
5 (* Se ingresso "IN" non è attivo l'uscita non è attiva ed il valore "ET" è 0 *)
6 (* Attivando ingresso "IN", il valore tempo in "ET" inizia ad incrementarsi e *)
7 (* raggiunto il valore di preset si attiva l'uscita. *)
8 (* ----- *)
9 (* Se input non attivo salvo tempo ed eseguo reset uscita. *)
10
11 IF NOT(IN) THEN
12     TimeBf:=SysTime; (* Time buffer (mS) *)
13     Q:=FALSE; (* Delayed output *)
14     RETURN;
15 END_IF;
16
17 (* Arrivo se input attivo controllo stato uscita. *)
18 (* Se PT inferiore tempo scansione, timer può bloccarsi in On, per sbloccarlo *)
19 (* eseguo controllo se variazione di PT. *)
20
21 IF (Q) THEN
22     IF (PT > ET) THEN Q:=FALSE; END_IF;
23     RETURN;
24 END_IF;
25
26 (* Eseguo controllo se tempo trascorso. *)
27
28 ET:=SysTime-TimeBf; (* Executing time (mS) *)
29
30 IF (ET >= PT) THEN
31     Q:=TRUE; (* Delayed output *)
32     ET:=PT; (* Executing time (mS) *)
33 END_IF;
34
35 (* [End of file] *)
36
37
```

Project : FBTimer	
FUNCTION_BLOCK : eTONVarExternal	
Release : FBTimer	Ver :1.00
Author :	Date:27/05/2015
Note :	Page:1 of 1

```

VAR_INPUT
IN : BOOL; (* Timer input *)
PT : UDINT; (* Preset time value (mS) *)
END_VAR

VAR_OUTPUT
Q : BOOL; (* Delayed output *)
ET : UDINT; (* Executing time (mS) *)
END_VAR

VAR
TimeBf : UDINT; (* Time buffer (uS) *)
END_VAR
    
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "eTONFunction" *)
3 (* ***** *)
4 (* Il blocco funzione esegue temporizzazione uscita su attivazione ingresso. *)
5 (* Se ingresso "IN" non è attivo l'uscita non è attiva ed il valore "ET" è 0 *)
6 (* Attivando ingresso "IN", il valore tempo in "ET" inizia ad incrementarsi e *)
7 (* raggiunto il valore di preset si attiva l'uscita. *)
8 (* ----- *)
9 (* Se input non attivo salvo tempo ed eseguo reset uscita. *)
10
11 IF NOT(IN) THEN
12     TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
13     Q:=FALSE; (* Delayed output *)
14     RETURN;
15 END_IF;
16
17 (* Arrivo se input attivo controllo stato uscita. *)
18 (* Se PT inferiore tempo scansione, timer può bloccarsi in On, per sbloccarlo *)
19 (* eseguo controllo se variazione di PT. *)
20
21 IF (Q) THEN
22     IF (PT > ET) THEN Q:=FALSE; END_IF;
23     RETURN;
24 END_IF;
25
26 (* Eseguo controllo se tempo trascorso. *)
27
28 ET:=(SysGetSysTime(TRUE)-TimeBf)/1000; (* Executing time (mS) *)
29
30 IF (ET >= PT) THEN
31     Q:=TRUE; (* Delayed output *)
32     ET:=PT; (* Executing time (mS) *)
33 END_IF;
34
35 (* [End of file] *)
36
37
    
```

Project : FBTimer	
FUNCTION BLOCK : eTONFunction	
Release : FBTimer	Ver :1.00
Author :	Date:27/05/2015
Note :	Page:1 of 1