

VARIABLES

VAR_GLOBAL

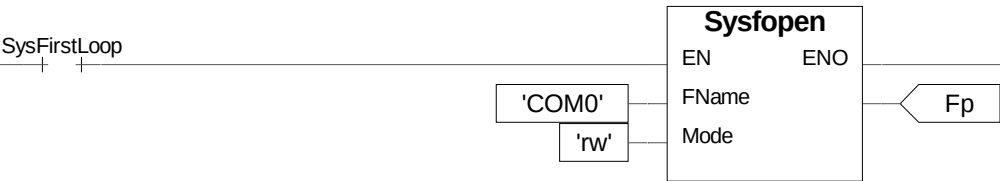
Ready AT %QX255.0 : BOOL; (* Ready output *)
Do01CPU AT %QX255.1 : BOOL; (* Do01 modulo CPU *)
Di00CPU AT %IX255.0 : BOOL; (* Di00 modulo CPU *)
Fp : FILEP; (* File pointer *)
CmActive : BOOL; (* Communication active *)
END_VAR

	Project : EthPCOutInterface	
	VARIABLES :	
	Release : SendDStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:1 of 1

```
VAR
SMode : SetSMode; (* Set serial mode *)
END_VAR
```

1

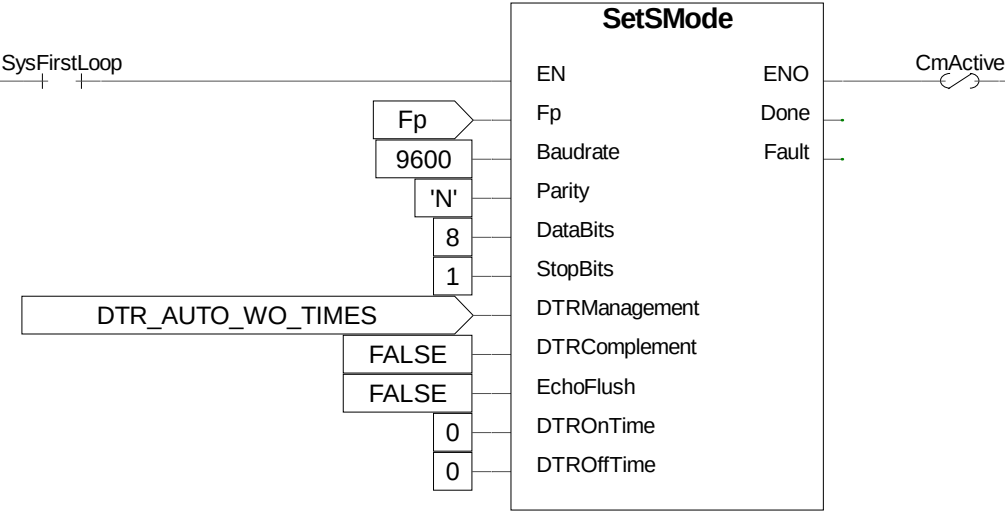
Open the serial COM



2

Set communication mode

SMode



Project : EthPCOutInterface

PROGRAM : SerialCom

Release : SendDlStat

Ver :1.00

Author :

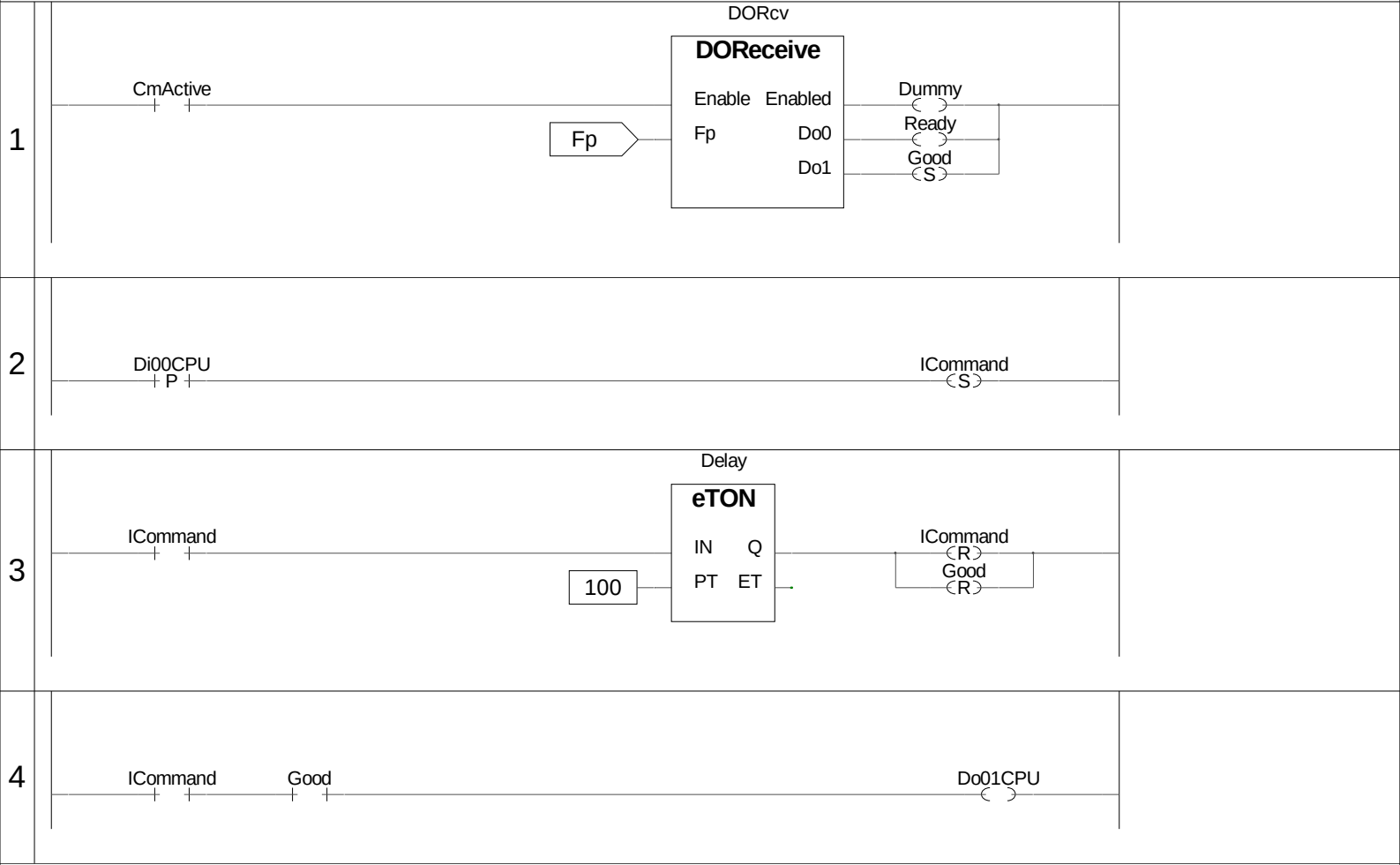
Date:09/12/2014

Note :

Page:1 of 1

PROGRAM Logic

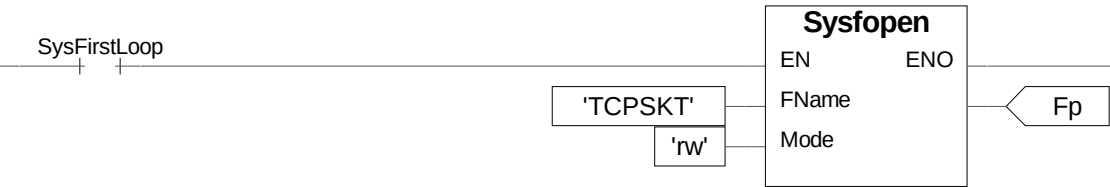
```
VAR
Dummy : BOOL; (* Dummy variable *)
DORcv : DOReceive; (* DOReceive FB instance *)
Good : BOOL; (* Good signal *)
Delay : eTON; (* Tempo di ritardo *)
ICommand : BOOL; (* Input command *)
END_VAR
```



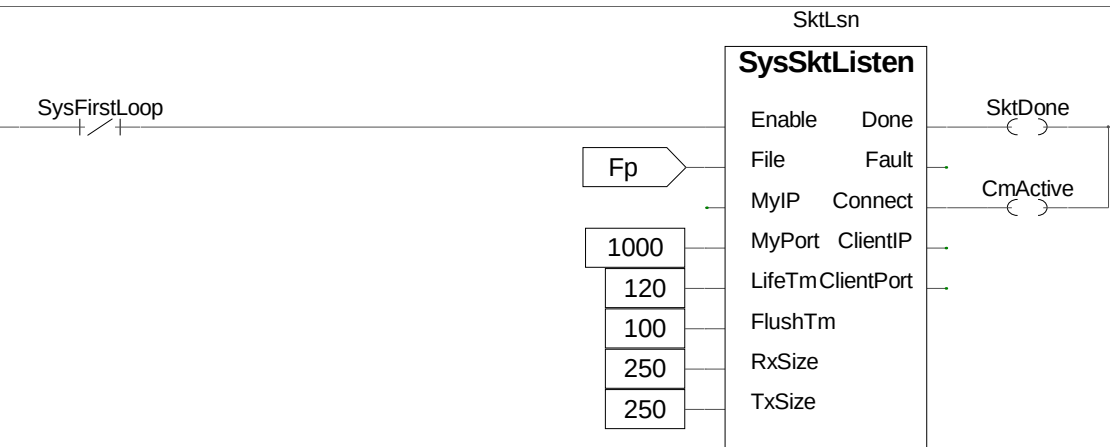
	Project : EthPCOutInterface	
	PROGRAM : Logic	
	Release : SendDStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:1 of 1

```
VAR
SktLsn : SysSktListen; (* FB Socket listen *)
SktDone : BOOL; (* Socket done *)
END_VAR
```

1



2



	Project : EthPCOutInterface	
	PROGRAM : TCPCom	
	Release : SendDlStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:1 of 1

(SFR054B000) Set the serial communication mode
ENCRYPTED CODE

```
VAR_INPUT
Fp : FILEP;
Baudrate : UDINT; (* Baudrate *)
Parity : STRING[ 1 ]; (* Parity type *)
DataBits : USINT; (* Nr of data bits *)
StopBits : USINT; (* Nr of stop bits *)
DTRManagement : USINT; (* DTR management type *)
DTRComplement : BOOL; (* Complement the DTR signal *)
EchoFlush : BOOL; (* Flush the echo *)
DTROnTime : UINT; (* DTR On wait time *)
DTROffTime : UINT; (* DTR Off wait time *)
END_VAR

VAR_OUTPUT
Done : BOOL; (* Execution done *)
Fault : BOOL; (* Execution fault *)
END_VAR
```

1

	Project : EthPCOutInterface	
	FUNCTION BLOCK : SetSMode	
	Release : SendDStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:1 of 1

VAR_INPUT

```
Enable : BOOL; (* FB Enable *)
Fp : FILEP; (* File pointer *)
END_VAR
```

VAR_OUTPUT

```
Enabled : BOOL := FALSE; (* FB Enabled *)
Do0 : BOOL; (* Do0 output *)
Do1 : BOOL; (* Do1 output *)
END_VAR
```

VAR

```
TimeBf : UDINT; (* Time buffer (uS) *)
RxData : STRING[ 32 ]; (* Rx data buffer *)
Ch : USINT; (* Rx character *)
SPtr : @USINT; (* String pointer *)
END_VAR
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "DOReceive" *)
3 (* ***** *)
4 (* Questa FB riceve i comandi attivazione uscite da stream di comunicazione. *)
5 (* *)
6 (* Su ricezione stringa S0<CR> attivo uscita Do0 *)
7 (* Su ricezione stringa R0<CR> disattivo uscita Do0 *)
8 (* Su ricezione stringa SA<CR> attivo per un loop uscita Do1 *)
9 (* ----- *)
10
11 (* ----- *)
12 (* FB ENABLE *)
13 (* ----- *)
14 (* Check if the FB is enabled. *)
15
16 IF NOT(Enable) THEN
17     Do0:=FALSE; (* Do0 output *)
18     Do1:=FALSE; (* Do1 output *)
19     SPtr:=ADR(RxData); (* String pointer *)
20     Enabled:=FALSE; (* FB Enabled *)
21     RETURN;
22 END_IF;
23
24 (* Azzerò uscite one shot. *)
25
26 Do1:=FALSE; (* Do1 output *)
27 Enabled:=TRUE; (* FB Enabled *)
28
29 (* ----- *)
30 (* GESTIONE RICEZIONE COMANDI *)
31 (* ----- *)
32 (* Check if there are any character in the receive buffer. *)
33
34 IF NOT(TO_BOOL(SysGetIChars(Fp))) THEN
35
36     (* If no characters are received for a 100 mS reception restarts. *)
37
38     IF ((SysGetSysTime(TRUE)-TimeBf) >= 100000) THEN SPtr:=ADR(RxData); END_IF;
39     RETURN;
```

Project : EthPCOutInterface

FUNCTION BLOCK : DOReceive

Release : SendDStat

Ver :1.00

Author :

Date:09/12/2014

Note :

Page:1 of 2

FUNCTION_BLOCK DOReceive

```

40  END_IF;
41
42  (* One or more characters are received store received character. *)
43
44  TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
45
46  (* Eseguo lettura caratteri ricevuti e controllo se ricevuto <CR>. *)
47
48  WHILE (TO_BOOL(SysGetIChars(Fp))) DO
49      Ch:=TO_USINT(Sysfgetc(Fp)); (* Rx character *)
50      @SPtr:=Ch; (* Salvo carattere ricevuto in stringa *)
51      SPtr:=SPtr+1; (* String pointer *)
52      @SPtr:=0; (* Termino stringa con codice tappo *)
53      IF (SPtr > ADR(RxData)+3) THEN SPtr:=ADR(RxData); RETURN; END_IF;
54
55      (* Controllo se ricevuto <CR>. *)
56
57      IF (Ch <> 16#0D) THEN RETURN; END_IF;
58
59      IF (FIND(RxData, 'S0') <> 0) THEN Do0:=TRUE; END_IF;
60      IF (FIND(RxData, 'R0') <> 0) THEN Do0:=FALSE; END_IF;
61      IF (FIND(RxData, 'SA') <> 0) THEN Do1:=TRUE; END_IF;
62
63      SPtr:=ADR(RxData); (* String pointer *)
64      RETURN;
65  END_WHILE;
66
67  (* [End of file] *)
68
69

```

	Project : EthPCOutInterface	
	FUNCTION BLOCK : DOReceive	
	Release : SendDStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:2 of 2

(ePLCStdLib_B000) On delay timer
ENCRYPTED CODE

```
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
```

1

	Project : EthPCOutInterface	
	FUNCTION BLOCK : eTON	
	Release : SendDStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:1 of 1

FUNCTION VBitTest

(SFR058A100) Tests a bit of a variable
ENCRYPTED CODE

```
VAR_INPUT
Variable : UDINT; (* Variable to test *)
Bit : USINT; (* Bit to test *)
END_VAR
```

1

	Project : EthPCOutInterface	
	FUNCTION : VBitTest	
	Release : SendDlStat	Ver :1.00
	Author :	Date:09/12/2014
	Note :	Page:1 of 1