

```

VAR
Fp : ARRAY[ 0..1 ] OF FILE; (* File pointer *)
i : INT; (* Auxiliary counter *)
RxStr : STRING[ 2048 ]; (* Rx string *)
TimeBf : UDINT; (* Buffer temporizzazione (uS) *)
CaseNr : USINT; (* Case gestione *)
RxChrs : UINT; (* Rx characters count *)
TxChrs : UINT; (* Tx characters count *)
j : INT; (* Auxiliary counter *)
Sm : SYSSERIALMODE; (* Serial mode *)
END_VAR

```

```

1  (* ----- *)
2  (* INIZIALIZZAZIONE *)
3  (* ----- *)
4  (* Eseguo inizializzazioni al primo loop di esecuzione. *)
5
6  IF (SysFirstLoop) THEN
7
8      (* Apro la COM0 in read/write ed imposto modo comunicazione. *)
9
10     Fp[0]:=Sysfopen('COM0', 'rw'); (* File pointer *)
11
12     i:=SysGetSerialMode(ADR(Sm), Fp[0]);
13     Sm.Baudrate:=115200;
14     i:=SysSetSerialMode(ADR(Sm), Fp[0]);
15
16     (* Apro il socket UDP. *)
17
18     Fp[1]:=Sysfopen('UDPSKT', 'rw'); (* File pointer *)
19 END_IF;
20
21
22 (* ----- *)
23 (* RICEZIONE DATI DA PORTA SERIALE *)
24 (* ----- *)
25
26 CASE (CaseNr) OF
27
28     (* ----- *)
29     (* RICEZIONE DATI DA PORTA SERIALE *)
30     (* ----- *)
31     (* Inizializzazione. *)
32
33     0:
34     RxChrs:=0; (* Rx characters count *)
35     CaseNr:=CaseNr+1; (* Case gestione *)
36
37     (* ----- *)
38     (* Eseguo ricezione dati da porta seriale. *)
39
40     1:
41     IF (SysGetIChars(Fp[0]) > 0) THEN
42         TimeBf:=SysGetSysTime(TRUE);
43         i:=Sysfread(ADR(RxStr)+RxChrs, 1, TO_INT(2048-RxChrs), Fp[0]);
44         RxChrs:=RxChrs+TO_UINT(i); (* Rx characters count *)
45     END_IF;

```

Project : SerialToUDP	
PROGRAM : RxSerial	
Release : RxSerialDa	Ver :1.00
Author :	Date:19/06/2012
Note :	Page:1 of 2

PROGRAM RxSerial

```

46
47 (* Eseguo attesa pausa ricezione. Trascorsi 100 mS senza ricevere *)
48 (* dati da porta seriale. Fine ricezione. *)
49
50 IF (RxChrs = 0) THEN RETURN; END_IF;
51 IF ((SysGetSysTime(TRUE)-TimeBf) < 1000000) THEN RETURN; END_IF;
52 TxChrs:=0; (* Tx characters count *)
53 CaseNr:=CaseNr+1; (* Case gestione *)
54
55 (* ----- *)
56 (* TRASMISSIONE DATI SU SOCKET UDP *)
57 (* ----- *)
58 (* Eseguo trasmissione stringa ricevuta su socket UDP, controllo *)
59 (* quanti caratteri devo ancora trasmettere. *)
60
61 2:
62 j:=TO_INT(RxChrs-TxChrs); (* Auxiliary counter *)
63 IF (j > 1000) THEN j:=1000; END_IF;
64
65 (* Eseguo trasmissione carateri. *)
66
67 i:=SysUDPSktSend(Fp[1], ADR(RxStr)+TxChrs, TO_UINT(j), '192.168.0.45', 1000);
68 TxChrs:=TxChrs+TO_UINT(i); (* Tx characters count *)
69
70 (* Eseguo controllo se trasmesso intera stringa. *)
71
72 IF (TxChrs < RxChrs) THEN RETURN; END_IF;
73 CaseNr:=0; (* Case gestione *)
74 END_CASE;
75
76 (* [End of file] *)
77
78

```

	Project : SerialToUDP	
	PROGRAM : RxSerial	
	Release : RxSerialDa	Ver :1.00
	Author :	Date:19/06/2012
	Note :	Page:2 of 2