

```

VAR
UDPSk : SysSktListen; (* UDP socket listen *)
TimeBf : UDINT; (* Time buffer (uS) *)
StringTx : STRING[ 32 ]; (* Buffer stringa da trasmettere *)
Vars : ARRAY[ 0..1 ] OF REAL; (* Variabili da trasmettere *)
i : INT; (* Auxiliary variable *)
END_VAR

```

```

1 (* ***** *)
2 (* PROGRAM "UDPTxmit" *)
3 (* ***** *)
4 (* Questo programma esegue l'invio di due variabili REAL su UDP. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* INIZIALIZAZIONI *)
9 (* ----- *)
10 (* Eseguo inizializzazioni. *)
11
12 IF (SysFirstLoop) THEN
13     UDPSk.File:=Sysfopen('UDPSKT', 'rw'); (* File pointer *)
14     UDPSk.MyPort:=1000; (* Porta UDP *)
15     TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
16 END_IF;
17
18 (* ----- *)
19 (* GESTIONE TRASMISSIONE *)
20 (* ----- *)
21 (* Attivo socket UDP in ascolto. *)
22
23 UDPSk(Enable:=TRUE); (* Socket UDP in ascolto *)
24
25 (* Eseguo temporizzazione invio valore variabili (1 Sec). *)
26
27 IF ((SysGetSysTime(TRUE)-TimeBf) < 1000000) THEN RETURN; END_IF;
28 TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
29
30 (* Eseguo incremento variabili per test. *)
31
32 Vars[0]:=Vars[0]+0.1; (* Variabili da trasmettere *)
33 Vars[1]:=Vars[1]+0.2; (* Variabili da trasmettere *)
34 IF (Vars[0] > 999.9) THEN Vars[0]:=0.0; END_IF;
35 IF (Vars[1] > 999.9) THEN Vars[1]:=0.0; END_IF;
36
37 (* Creo messaggio da inviare con valore variabili. *)
38
39 i:=SysVarsnprintf(ADR(StringTx), 16+1, '[%.2f - ', REAL_TYPE, ADR(Vars[0]));
40 i:=SysVarsnprintf(ADR(StringTx)+LEN(StringTx), 16+1, '%.2f]', REAL_TYPE, ADR(Vars[1]));
41 i:=SysUDPSktSend(UDPSk.File, ADR(StringTx), LEN(StringTx), '192.168.0.42', 2000);
42
43 (* [End of file] *)
44

```

Project : VarsToUDP

PROGRAM : UDPTxmit

Release : VarsToUDP

Ver :1.00

Author :

Date:11/04/2014

Note :

Page:1 of 1