

VARIABLES

VAR_GLOBAL

SMSOn : BOOL; (* Invio SMS *)

Phone : STRING[32]; (* Phone number *)

Message : STRING[32]; (* SMS text *)

SMSSent : BOOL; (* Messaggio SMS inviato *)

END_VAR

	Project : SMSMachine	
	VARIABLES :	
	Release : SMSMachine	Ver :1.00
	Author :	Date:25/02/2012
	Note :	Page:1 of 1

```

VAR
TPs : ARRAY[ 0..1 ] OF UINT; (* Token position *)
Fp : FILEP; (* File pointer *)
SktLsn : SysSktListen; (* FB socket listen *)
SktRx : SysUDPSktRcv; (* FB socket receive *)
RxCommand : STRING[ 32 ]; (* Rx command string *)
RxPeerIP : STRING[ 32 ]; (* IP da cui abilito ricezione UDP *)
RxPeerPort : UINT; (* Porta da cui abilito ricezione UDP *)
TxPeerIP : STRING[ 32 ]; (* IP verso cui invio risposte UDP *)
TxPeerPort : UINT; (* Porta verso cui invio risposte UDP *)
TxDot : BOOL; (* Tx Dot flag *)
Dummy : INT; (* Dummy variable *)
END_VAR

```

```

1 (* ***** *)
2 (* UDPInterface *)
3 (* ***** *)
4 (* Questo programma esegue gestione della comunicazione UDP, viene eseguito *)
5 (* il token della stringa di comando rilevando numero di telefono e testo del *)
6 (* messaggio da inviare. *)
7 (* *)
8 (* Stringa comando: *)
9 (* [+3912345678],(Testo messaggio) *)
10 (* ----- *)
11
12 (* ----- *)
13 (* ESEGUO INIZIALIZZAZIONI *)
14 (* ----- *)
15 (* Eseguo inizializzazioni al primo loop. *)
16
17 IF (SysFirstLoop) THEN
18
19     RxPeerIP:='255.255.255.255'; (* IP da cui abilito ricezione UDP *)
20     RxPeerPort:=65535; (* Porta da cui abilito ricezione UDP *)
21
22     TxPeerIP:='192.168.0.45'; (* IP verso cui invio risposte UDP *)
23     TxPeerPort:=1000; (* Porta verso cui invio risposte UDP *)
24
25     (* Eseguo apertura socket UDP. *)
26
27     Fp:=Sysfopen('UDPSKT', 'rw'); (* File pointer *)
28     RETURN;
29 END_IF;
30
31 (* ----- *)
32 (* GESTIONE INVIO RISPOSTE *)
33 (* ----- *)
34 (* Durante invio messaggio viene ritornata una serie di caratteri '.'. *)
35
36 IF (SMSOn AND SysClock1000 AND NOT(TxDot)) THEN
37     Dummy:=SysUDPSktSend(Fp, ADR('.'), 1, TxPeerIP, TxPeerPort);
38 END_IF;
39
40 TxDot:=SysClock1000; (* Tx Dot flag *)
41
42 (* Su SMS inviato eseguo report. *)
43

```

Project : SMSMachine

PROGRAM : UDPInterface

Release : SMSMachine

Ver :1.00

Author :

Date:25/02/2012

Note :

Page:1 of 2

PROGRAM UDPInterface

```

44 IF (SMSSent) THEN
45     Dummy:=SysUDPSktSend(Fp, ADR('$rSMS Sent$r'), 32, TxPeerIP, TxPeerPort);
46     SMSSent:=FALSE; (* Messaggio SMS inviato *)
47 END_IF;
48
49 (* ----- *)
50 (* GESTIONE RICEZIONE COMANDI *)
51 (* ----- *)
52 (* Forzo socket in condizione di listening. *)
53
54 SktLsn.File:=Fp; (* Flusso dati stream *)
55 SktLsn.MyPort:=1000; (* Porta in ascolto su socket *)
56 SktLsn(Enable:=TRUE); (* Forzo socket in listening *)
57
58 (* Eseguo ricezione dati da socket UDP. Accetto connessioni da tutti *)
59 (* gli IP e da tutte le porte. *)
60
61 SktRx.File:=Fp; (* Flusso dati stream *)
62 SktRx.PeerIP:='255.255.255.255'; (* IP da cui accetto connessioni *)
63 SktRx.PeerPort:=65535; (* Porta da cui accetto connessioni *)
64 SktRx.RxData:=ADR(RxCommand); (* Buffer dati ricevuti *)
65 (* SktRx.Size:=SIZEOF(RxCommand); (* Dimensione buffer dati ricevuti *)
66 SktRx.Size:=32; (* Dimensione buffer dati ricevuti *)
67 SktRx(Enable:=TRUE); (* Ricezione dati da socket *)
68 IF NOT(SktRx.Done) THEN RETURN; END_IF;
69
70 (* ----- *)
71 (* ESEGUO PARSING COMANDO RICEVUTO *)
72 (* ----- *)
73 (* Ricevuto comando estraggo da stringa numero di telefono. *)
74
75 TPs[0]:=FIND(RxCommand, '['); (* Token position *)
76 TPs[1]:=FIND(RxCommand, ']'); (* Token position *)
77 Phone:=MID(RxCommand, TPs[1]-TPs[0]-1, TPs[0]+1); (* Phone number *)
78
79 (* Ricevuto comando estraggo da stringa messaggio. *)
80
81 TPs[0]:=FIND(RxCommand, '('); (* Token position *)
82 TPs[1]:=FIND(RxCommand, ')'); (* Token position *)
83 Message:=MID(RxCommand, TPs[1]-TPs[0]-1, TPs[0]+1); (* SMS text *)
84
85 (* Invio risposta di invio SMS in corso. *)
86
87 Dummy:=SysUDPSktSend(Fp, ADR('SMS sending'), 32, TxPeerIP, TxPeerPort);
88 SMSOn:=TRUE; (* Invio SMS *)
89
90 (* [End of file] *)
91

```

	Project : SMSMachine	
	PROGRAM : UDPInterface	
	Release : SMSMachine	Ver :1.00
	Author :	Date:25/02/2012
	Note :	Page:2 of 2

```

VAR
Fp : FILEP; (* File pointer *)
Sm : SYSSERIALMODE; (* Serial mode data struct *)
Dummy : BOOL; (* Dummy variable *)
Mdm : ModemCore; (* FB gestione modem *)
SMSSend : ModemSMSSend; (* FB invio SMS *)
Tx : STRING[ 32 ]; (* Tx string monitor *)
Rx : STRING[ 32 ]; (* Rx string monitor *)
END_VAR

```

```

1 (* ***** *)
2 (* SMSSending *)
3 (* ***** *)
4 (* Questo programma esegue gestione dell'invio del messaggio SMS. *)
5 (* *)
6 (* Stringa comando: *)
7 (* [+3912345678],(Testo messaggio) *)
8 (* ----- *)
9
10 (* ----- *)
11 (* ESEGUO INIZIALIZZAZIONI *)
12 (* ----- *)
13 (* Eseguo inizializzazioni al primo loop. *)
14
15 IF (SysFirstLoop) THEN
16
17     (* Eseguo apertura porta seriale. *)
18
19     Fp:=Sysfopen('COM0', 'rw'); (* File pointer *)
20
21     (* Eseguo parametrizzazione porta seriale. *)
22
23     Dummy:=SysGetSerialMode(ADR(Sm), Fp); (* Get serial mode *)
24     Sm.Baudrate:=9600; (* Baudrate *)
25     Sm.Parity:='N'; (* Parity *)
26     Sm.DataBits:=8; (* Data bits *)
27     Sm.DTRManagement:=DTR_AUTO_WO_TIMES; (* DTR management mode *)
28     Dummy:=SysSetSerialMode(ADR(Sm), Fp); (* Set serial mode *)
29     RETURN;
30 END_IF;
31
32 (* ----- *)
33 (* GESTIONE MODEM *)
34 (* ----- *)
35 (* Gestione modem. *)
36
37 Mdm.File:=Fp; (* Flusso dati stream *)
38 Mdm(Enable:=TRUE); (* Gestione modem *)
39 Rx:=Mdm.ModemRx; (* Rx string monitor *)
40 Tx:=Mdm.ModemTx; (* Tx string monitor *)
41
42 (* ----- *)
43 (* GESTIONE INVIO SMS *)
44 (* ----- *)
45 (* Gestione invio SMS. *)
46
47 SMSSend.ModemID:=Mdm.ModemID; (* ID modem *)

```

Project : SMSMachine

PROGRAM : SMSSending

Release : SMSMachine

Ver :1.00

Author :

Date:25/02/2012

Note :

Page:1 of 2

PROGRAM SMSSending

```

48     SMSSend.Number:=Phone; (* Numero di telefono *)
49     SMSSend.Text:=Message; (* Messaggio da inviare *)
50     SMSSend(Send:=SMSOn); (* Eseguo invio SMS *)
51
52     IF (SMSSend.Done) THEN
53         SMSOn:=FALSE; (* Invio SMS *)
54         SMSSent:=TRUE; (* Messaggio SMS inviato *)
55     END_IF;
56
57 (* [End of file] *)
58

```

	Project : SMSMachine	
	PROGRAM : SMSSending	
	Release : SMSMachine	Ver :1.00
	Author :	Date:25/02/2012
	Note :	Page:2 of 2