

VARIABLES

```

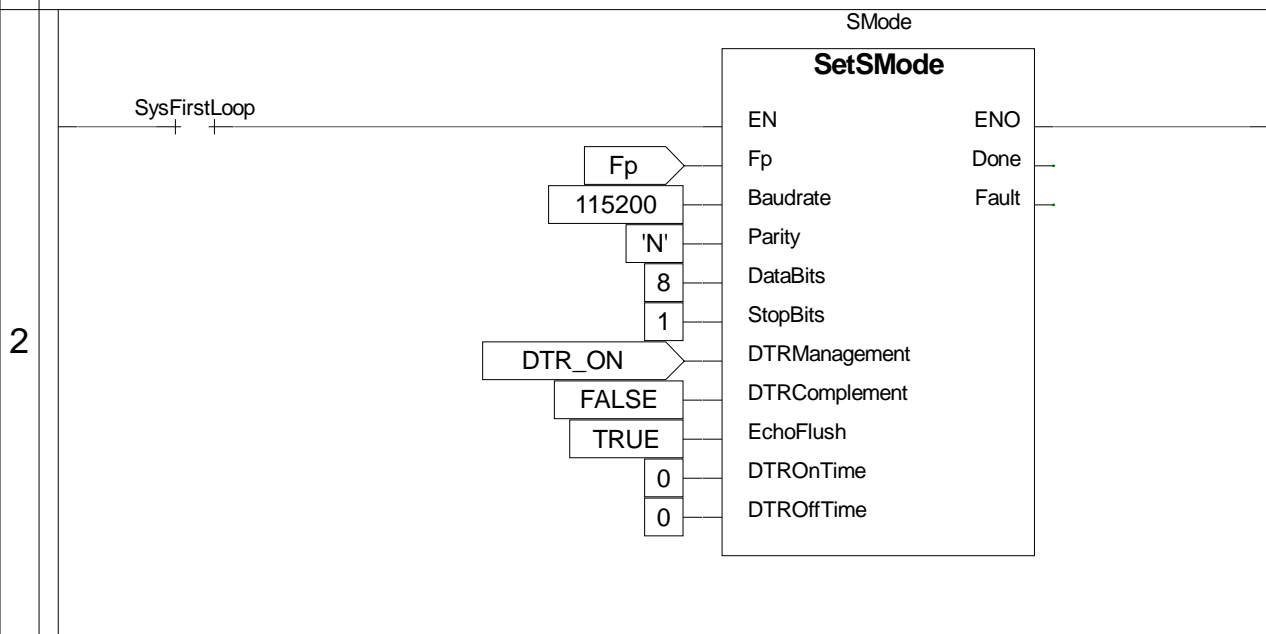
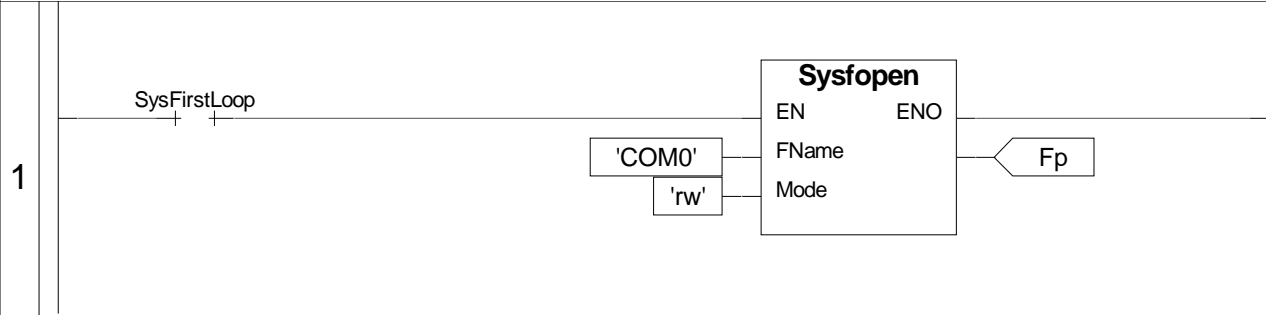
VAR_GLOBAL
Di00CPU AT %IX255.0 : BOOL;
Do00M00 AT %QX0.0 : BOOL;
Messaggi : ARRAY[ 0..15 ] OF @USINT;
Telefoni : ARRAY[ 0..15 ] OF @USINT;
MsgSelector : DWORD; (* Message selector *)
PNrSelector : DWORD; (* Phone number selector *)
END_VAR

```

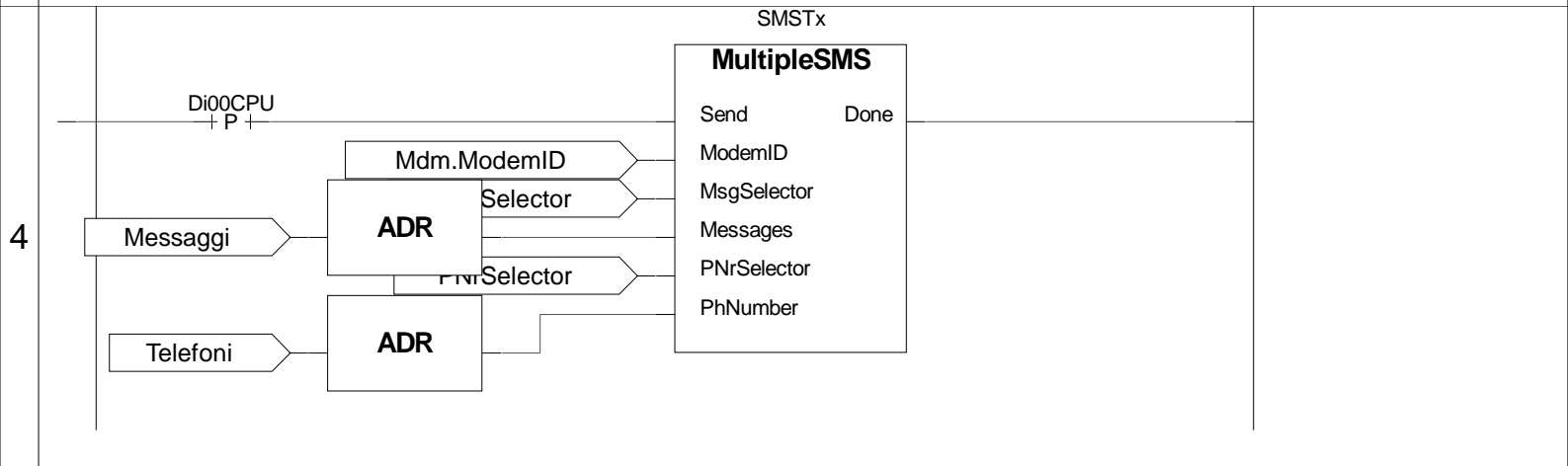
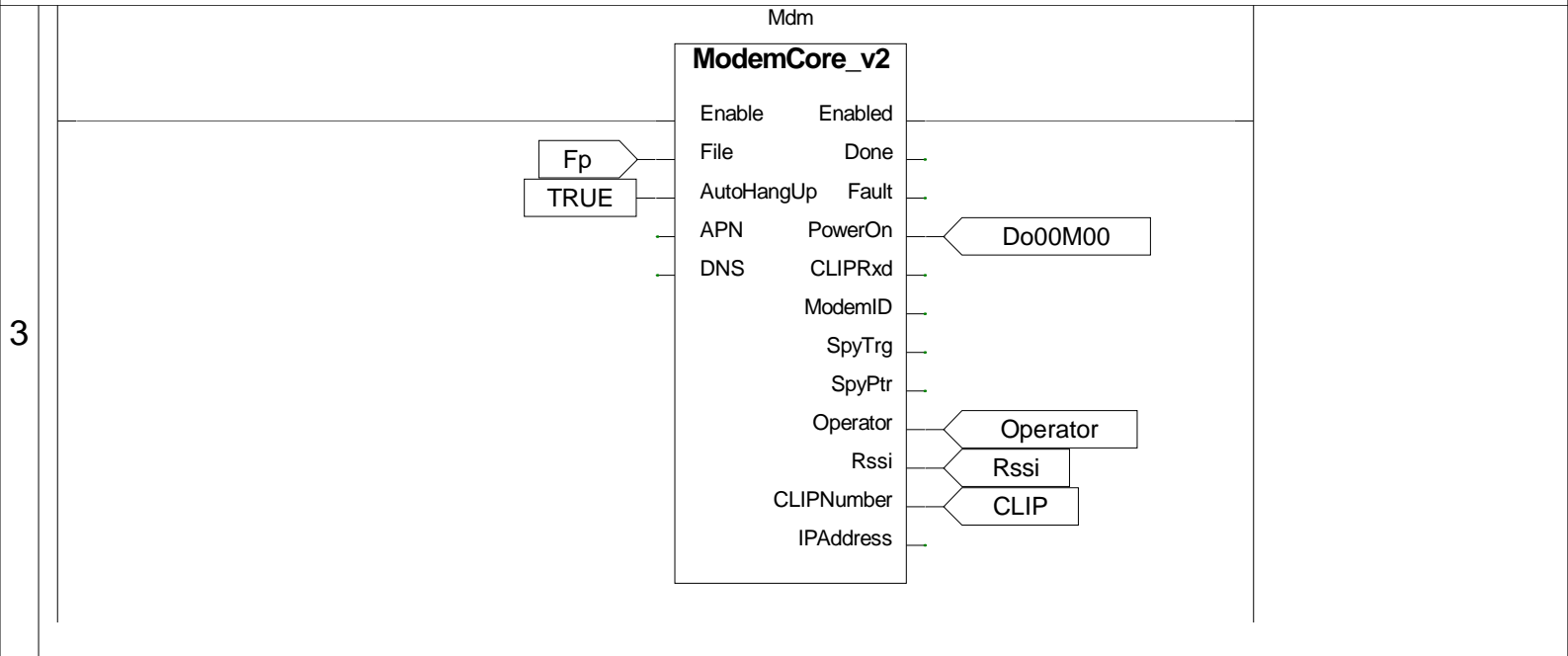
	Project : MultipleSMS	
	VARIABLES :	
	Release : MultipleSM	Ver :1.00
	Author :	Date:07/10/2013
	Note :	Page:1 of 1

```

VAR
Rssi : USINT; (* RSSI value *)
CLIP : STRING[ 16 ]; (* CLIP received number *)
Operator : STRING[ 16 ]; (* Operator value *)
Fp : FILEP; (* Serial pointer *)
Mdm : ModemCore_v2; (* Modem core FB *)
SMode : SetSMode; (* Serial mode FB *)
SMSTx : MultipleSMS;
END_VAR
    
```



Project : MultipleSMS	
PROGRAM : MultipleSMSSend	
Release : MultipleSM	Ver :1.00
Author :	Date:07/10/2013
Note :	Page:1 of 2



Project : MultipleSMS	
PROGRAM : MultipleSMSSend	
Release : MultipleSM	Ver :1.00
Author :	Date:07/10/2013
Note :	Page:2 of 2

PROGRAM StartUp

```

1 (* ***** *)
2 (* FUNCTION BLOCK "MultipleSMS" *)
3 (* ***** *)
4 (* Il blocco funzione esegue l'impostazione dei parametri di comunicazione *)
5 (* su porta seriale. *)
6 (* ----- *)
7
8 (* ----- *)
9 (* MESSAGGI DA INVIARE *)
10 (* ----- *)
11 (* Carico i messaggi da inviare. *)
12
13 MsgSelector:=16#00000003; (* Message selector *)
14 Messaggi[0]:=ADR('Messaggio 0');
15 Messaggi[1]:=ADR('Messaggio 1');
16 Messaggi[2]:=ADR('Messaggio 2');
17
18 (* ----- *)
19 (* NUMERI DI TELEFONO *)
20 (* ----- *)
21 (* Carico i numeri di telefono a cui inviare. *)
22
23 PNrSelector:=16#00000003; (* Phone number selector *)
24 Telefoni[0]:=ADR('+39335217595');
25 Telefoni[1]:=ADR('+39335217595');
26
27 (* [End of file] *)
28
29

```

	Project : MultipleSMS	
	PROGRAM : StartUp	
	Release : MultipleSM	Ver :1.00
	Author :	Date:07/10/2013
	Note :	Page:1 of 1

FUNCTION_BLOCK SetSMoDe

(SFR054B210) 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 : MultipleSMS	
	FUNCTION BLOCK : SetSMoDe	
	Release : MultipleSM	Ver :1.00
	Author :	Date:07/10/2013
	Note :	Page:1 of 1

```

VAR_INPUT
Send : BOOL; (* Send command *)
ModemID : UDINT; (* Modem ID *)
MsgSelector : DWORD; (* Message selector *)
Messages : @USINT; (* SMS messages *)
PNrSelector : DWORD; (* Phone number selector *)
PhNumber : @USINT; (* Phone numbers *)
END_VAR
    
```

```

VAR_OUTPUT
Done : BOOL; (* Invio eseguito *)
END_VAR
    
```

```

VAR
SdCmd : BOOL; (* Comando invio SMS *)
SMSSend : ModemSMSSend_v2; (* FB invio SMS *)
PNrIdx : USINT; (* Phone number index *)
TxtIdx : USINT; (* Text index *)
CaseNr : USINT; (* Case gestione *)
Ptr : @UDINT; (* Auxiliary pointer *)
END_VAR
    
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "MultipleSMS" *)
3 (* ***** *)
4 (* Il blocco funzione esegue l'impostazione dei parametri di comunicazione *)
5 (* su porta seriale. *)
6 (* ----- *)
7
8 (* ----- *)
9 (* INIZIALIZZAZIONI *)
10 (* ----- *)
11 (* Eseguo inizializzazioni. *)
12
13 SMSSend.ModemID:=ModemID; (* Modem ID *)
14 SMSSend.TLgt:=eLEN(SMSSend.Text); (* Message length *)
15 SMSSend(); (* Eseguo FB invio SMS *)
16 SMSSend.Send:=FALSE; (* Comando invio SMS *)
17
18 (* ----- *)
19 (* GESTIONE COMANDO INVIO *)
20 (* ----- *)
21 (* Eseguo controllo se comando invio. *)
22
23 IF NOT(Send) THEN Done:=FALSE; END_IF;
24
25 (* Se terminato acquisizione esco. Occorre disabilitare comando "Send" e *)
26 (* poi abilitare per effettuare una nuova esecuzione comando. *)
27
28 IF (Done) THEN RETURN; END_IF;
29
30 (* ----- *)
31 (* GESTIONE CASE PROGRAMMA *)
32 (* ----- *)
33 (* Eseguo gestione case programma *)
34
35 CASE (CaseNr) OF
    
```

Project : MultipleSMS	
FUNCTION_BLOCK : MultipleSMS	
Release : MultipleSM	Ver :1.00
Author :	Date:07/10/2013
Note :	Page:1 of 2

```

36
37 (* ----- *)
38 (* ATTESA COMANDO INVIO *)
39 (* ----- *)
40 (* Eseguo attesa comando invio. *)
41
42 0:
43 IF NOT(Send) THEN RETURN; END_IF;
44 TxtIDx:=16#FF; (* Text index *)
45 CaseNr:=10; (* Case gestione *)
46
47 (* ----- *)
48 (* SELEZIONE MESSAGGIO *)
49 (* ----- *)
50 (* Eseguo selezione messaggio da inviare, invio solo i messaggi che *)
51 (* hanno il relativo bit in "TxtSelector" attivo. *)
52
53 10:
54 TxtIDx:=TxtIDx+1; (* Text index *)
55 IF (TxtIDx > 31) THEN Done:=TRUE; CaseNr:=0; RETURN; END_IF;
56 IF NOT(VBitTest(MsgSelector, TxtIDx)) THEN RETURN; END_IF;
57
58 (* Trasferisco indirizzo testo messaggio da inviare. *)
59
60 Ptr:=Messages+(TxtIDx*4); (* Auxiliary pointer *)
61 SMSSend.Text:=@Ptr;
62 PNrIDx:=16#FF; (* Phone number index *)
63 CaseNr:=20; (* Case gestione *)
64
65 (* ----- *)
66 (* SELEZIONE NUMERO DI TELEFONO *)
67 (* ----- *)
68 (* Eseguo selezione numero di telefono a cui inviare il messaggio, *)
69 (* invio ai numeri che hanno il relativo bit in "PNrSelector" attivo. *)
70
71 20:
72 PNrIDx:=PNrIDx+1; (* Phone number index *)
73 IF (PNrIDx > 31) THEN CaseNr:=10; RETURN; END_IF;
74 IF NOT(VBitTest(PNrSelector, PNrIDx)) THEN RETURN; END_IF;
75
76 (* Trasferisco numero telefono a cui inviare SMS. *)
77
78 Ptr:=PhNumber+(PNrIDx*4); (* Auxiliary pointer *)
79 SMSSend.Number:=@Ptr;
80 SMSSend.Send:=TRUE; (* Comando invio SMS *)
81 CaseNr:=CaseNr+1; (* Case gestione *)
82
83 (* ----- *)
84 (* Attesa fine invio messaggio, *)
85
86 21:
87 IF ((SMSSend.Done) OR (SMSSend.Fault)) THEN CaseNr:=CaseNr-1; END_IF;
88 ELSE
89 CaseNr:=0; (* Case gestione *)
90 END_CASE;
91
92 (* [End of file] *)
93
94

```

	Project : MultipleSMS	
	FUNCTION BLOCK : MultipleSMS	
	Release : MultipleSM	Ver :1.00
	Author :	Date:07/10/2013
	Note :	Page:2 of 2

FUNCTION VBitTest

(SFR058A200) Tests a bit of a variable
ENCRYPTED CODE

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

1

	Project : MultipleSMS	
	FUNCTION : VBitTest	
	Release : MultipleSM	Ver :1.00
	Author :	Date:07/10/2013
	Note :	Page:1 of 1

FUNCTION eLEN

(SFR053A500) String length calculation
ENCRYPTED CODE

```
VAR_INPUT  
Str : @USINT; (* String pointer *)  
END_VAR
```

1

	Project : MultipleSMS	
	FUNCTION : eLEN	
	Release : MultipleSM	Ver :1.00
	Author :	Date:07/10/2013
	Note :	Page:1 of 1