

VARIABLES

```
VAR_GLOBAL
Di00M00 : BOOL;
Di01M00 : BOOL;
Do00M00 : BOOL;
Do01M00 : BOOL;
Start : BOOL; (* Test baud rate start *)
END_VAR
```

	Project : ModemSSetFind	
	VARIABLES :	
	Release : BaudChange	Ver :1.00
	Author :	Date:31/05/2012
	Note :	Page:1 of 1

```

VAR
Fp : ARRAY[ 0..1 ] OF FILEP; (* File pointer *)
Sm : SYSSERIALMODE; (* Serial mode *)
i : INT; (* Auxiliary counter *)
TimeBf : UDINT; (* Time buffer *)
CaseNr : USINT; (* Program case *)
BaudRates : ARRAY[ 0..7 ] OF UDINT := 1200,2400,4800,9600,19200,38400,57600,115200; (* Baud rates t
able *)
BRIdx : UINT; (* Baud rate index *)
PtIdx : UINT; (* Parity index *)
Parity : STRING[ 3 ] := 'EN0'; (* Parity table  E, N, 0  *)
RxTxString : STRING[ 16 ]; (* Rx/Tx string *)
StartPulse : BOOL; (* Start pulse *)
BaudFind : BOOL; (* Bud rate find command *)
END_VAR

```

```

1 (* ***** *)
2 (* COMMUNICATION MODE FIND *)
3 (* ***** *)
4 (* Questo programma esegue ricerca modo di comunicazione a cui è impostato un *)
5 (* Modem. Basta connettere il modem alla porta COM0 della CPU, il programma *)
6 (* invia la stringa "AT<CR>" e controlla ricezione da parte del modem della *)
7 (* stringa di risposta corretta "OK<CR>". *)
8 (* Il test viene eseguito inviando la stringa con diversi baud rates e parità *)
9 (* fino a trovare quella corretta a cui è impostato il modem. Su porta COM1 *)
10 (* viene riportato l'esito del test. *)
11 (* Il test inizia attivando l'ingresso I00 del modulo CPU. *)
12 (* ----- *)
13
14 (* ----- *)
15 (* INIZIALIZZAZIONE *)
16 (* ----- *)
17 (* Apro le COM0 e COM1 port in read/write ed inizializzo variabili. *)
18
19 IF (SysFirstLoop) THEN
20     Fp[0]:=Sysfopen('COM0', 'rw'); (* File pointer *)
21     Fp[1]:=Sysfopen('COM1', 'rw'); (* File pointer *)
22 END_IF;
23
24 (* Eseguo controllo se abilitazione ricerca modo comunicazione. *)
25
26 IF NOT(BaudFind) AND (Start <> StartPulse) THEN
27     StartPulse:=Start; (* Start pulse *)
28
29     IF (Start) THEN
30         BaudFind:=TRUE; (* Bud rate find command *)
31         CaseNr:=0; (* Program case *)
32         BRIdx:=0; (* Baud rate index *)
33         PtIdx:=0; (* Parity index *)
34     END_IF;
35 END_IF;
36
37 (* Eseguo controllo se comando ricerca baud rate in corso. *)
38
39 IF NOT(BaudFind) THEN RETURN; END_IF;
40
41 (* ----- *)

```

Project : ModemSSetFind

PROGRAM : SCommunication

Release : BaudChange

Ver :1.00

Author :

Date:31/05/2012

Note :

Page:1 of 3

PROGRAM SCommunication

```

42  (* CASES PROGRAMMA *)
43  (* ----- *)
44  (* Apro la COM0 port in read/write ed imposto parametri. *)
45
46  CASE (CaseNr) OF
47
48      (* ----- *)
49      (* Imposto parametri seriali. *)
50
51      0:
52      i:=SysGetSerialMode(ADR(Sm), Fp[0]);
53      Sm.Baudrate:=BaudRates[BRIdx];
54      Sm.Parity:=MID(Parity, 1, PtIdx+1);
55      Sm.DTRManagement:=DTR_ON;
56      i:=SysSetSerialMode(ADR(Sm), Fp[0]);
57
58      (* Stampo parametri con cui è eseguito il test. *)
59
60      i:=SysVarfprintf(Fp[1], 'Test, Baud is:%04d ', UDINT_TYPE, ADR(Sm.Baudrate));
61      i:=SysVarfprintf(Fp[1], 'Parity is:%c$r$n', USINT_TYPE, ADR(Sm.Parity));
62
63      (* Invio stringa di richiesta "AT<CR><LF>". *)
64
65      i:=SysFIBfClear(Fp[0]);
66      RxTxString:='AT$r$n'; (* Rx/Tx string *)
67      i:=Sysfwrite(ADR(RxTxString), TO_INT(LEN(RxTxString)), 1, Fp[0]);
68      TimeBf:=SysGetSysTime(TRUE); (* Time buffer *)
69      CaseNr:=CaseNr+1; (* Program case *)
70
71      (* ----- *)
72      (* Attendo tempo di pausa per ricezione risposta (1 Sec). *)
73
74      1:
75      IF ((SysGetSysTime(TRUE)-TimeBf) < 1000000) THEN RETURN; END_IF;
76
77      (* Durante tempo pausa dovrei sicuramente avere ricevuto risposta, *)
78      (* se il baud rate era corretto la risposta ha il corretto numero *)
79      (* di caratteri (3 caratteri, "OK<CR>"): *)
80
81      IF (SysGetIChars(Fp[0]) <> 3) THEN CaseNr:=10; RETURN; END_IF;
82
83      (* Numero caratteri ricevuti corretti, controllo stringa. *)
84
85      i:=Sysfread(ADR(RxTxString), 3, 1, Fp[0]);
86      IF ((FIND(RxTxString, 'ok$r')+FIND(RxTxString, 'OK$r')) = 0) THEN CaseNr:=10; RETURN; END_IF;
87
88      (* Arrivo se ricevuto risposta corretta. Stampo rapporto valori *)
89      (* di comunicazione corretti. *)
90
91      i:=SysVarfprintf(Fp[1], 'Baud rate found, Baud is:%04d ', UDINT_TYPE, ADR(BaudRates[BRIdx]));
92      i:=SysVarfprintf(Fp[1], 'Parity is:%c$r$n', USINT_TYPE, ADR(Parity[PtIdx]));
93      BaudFind:=FALSE; (* Bud rate find command *)
94
95      (* ----- *)
96      (* Arrivo se non ricevuto risposta, cambio modo di comunicazione. *)
97
98      10:
99      CaseNr:=0; (* Program case *)
100     IF (PtIdx < (SIZEOF(Parity)-2)) THEN PtIdx:=PtIdx+1; RETURN; END_IF;
101     PtIdx:=0; (* Parity index *)

```

Project : ModemSSetFind

PROGRAM : SCommunication

Release : BaudChange

Ver :1.00

Author :

Date:31/05/2012

Note :

Page:2 of 3

PROGRAM SCommunication

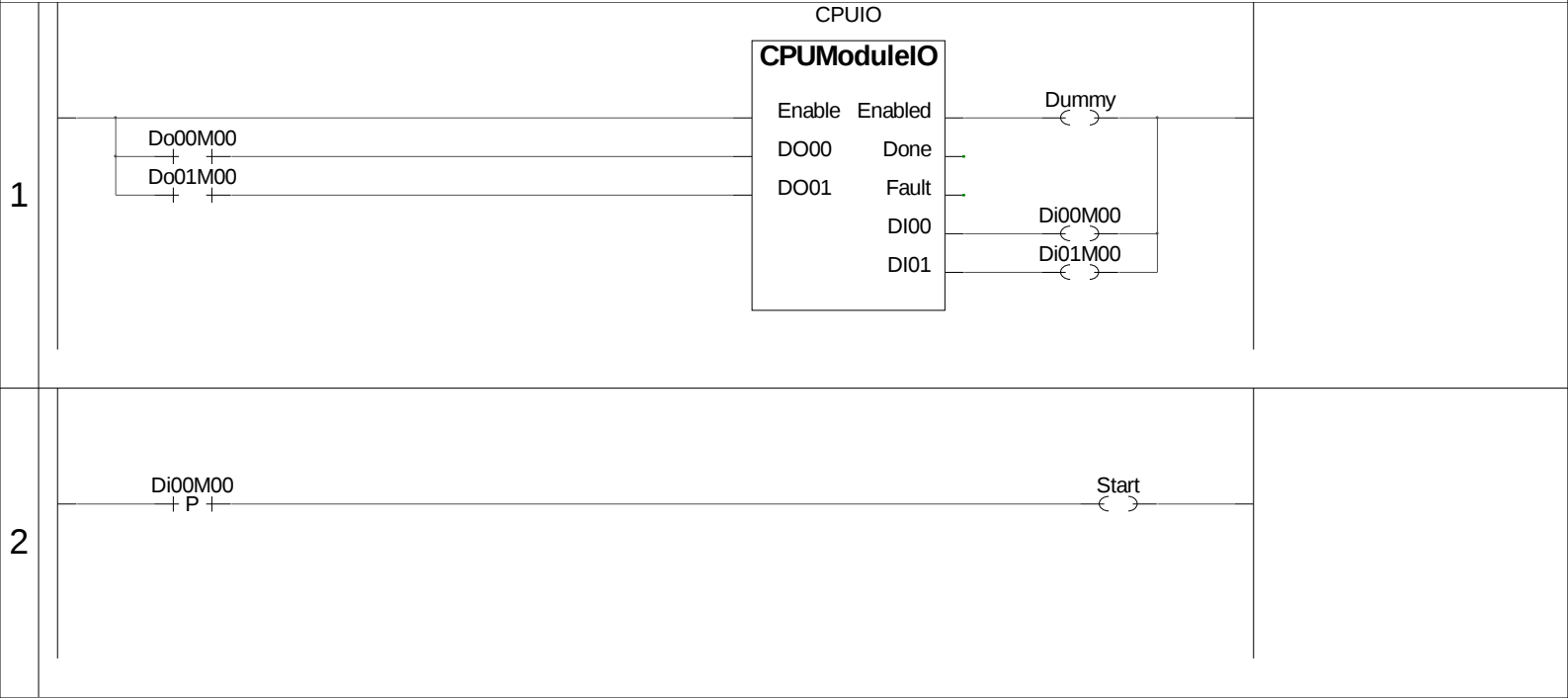
```

102
103      (* Cambio baud rate comunicazione. *)
104
105      IF (BRIdx < ((SIZEOF(BaudRates)/4)-1)) THEN BRIdx:=BRIdx+1; RETURN; END_IF;
106
107      (* Ho eseguito il test in tutti i modi e baud rate possibili. *)
108
109      RxCtxString:='Test ended.$r$n'; (* Rx/Tx string *)
110      i:=Sysfwrite(ADR(RxCtxString), TO_INT(LEN(RxCtxString)), 1, Fp[1]);
111      BaudFind:=FALSE; (* Bud rate find command *)
112      END_CASE;
113
114      (* [End of file] *)
115
116

```

	Project : ModemSSetFind	
	PROGRAM : SCommunication	
	Release : BaudChange	Ver :1.00
	Author :	Date:31/05/2012
	Note :	Page:3 of 3

```
VAR
CPUIO : CPUModuleIO; (* CPUModuleIO function block *)
Dummy : BOOL; (* Dummy variable *)
END_VAR
```



	Project : ModemSSetFind	
	PROGRAM : Logic	
	Release : BaudChange	Ver :1.00
	Author :	Date:31/05/2012
	Note :	Page:1 of 1

FUNCTION_BLOCK CPUModuleIO

(SFR054B100) Manages the logic I/O on the CPU module
 ENCRYPTED CODE

```
VAR_INPUT
Enable : BOOL; (* Function enable *)
D000 : BOOL; (* Digital output 0 *)
D001 : BOOL; (* Digital output 1 *)
END_VAR

VAR_OUTPUT
Enabled : BOOL; (* Function enabled *)
Done : BOOL; (* Function done *)
Fault : BOOL; (* Function fault *)
DI00 : BOOL; (* Digital input 0 *)
DI01 : BOOL; (* Digital input 0 *)
END_VAR
```

1

	Project : ModemSSetFind	
	FUNCTION BLOCK : CPUModuleIO	
	Release : BaudChange	Ver :1.00
	Author :	Date:31/05/2012
	Note :	Page:1 of 1