

VARIABLES

VAR_GLOBAL

Di00CPU AT %IX255.0 : BOOL; (* Input 00 CPU module *)
 Di01CPU AT %IX255.1 : BOOL; (* Input 01 CPU module *)
 Di02CPU AT %IX255.2 : BOOL; (* Input 02 CPU module *)
 Di03CPU AT %IX255.3 : BOOL; (* Input 03 CPU module *)
 Di04CPU AT %IX255.4 : BOOL; (* Input 04 CPU module *)
 Di05CPU AT %IX255.5 : BOOL; (* Input 05 CPU module *)
 Do00CPU AT %QX255.0 : BOOL; (* Output 00 CPU module *)
 Do01CPU AT %QX255.1 : BOOL; (* Output 01 CPU module *)
 Do02CPU AT %QX255.2 : BOOL; (* Output 02 CPU module *)
 Do03CPU AT %QX255.3 : BOOL; (* Output 03 CPU module *)
 END_VAR

Project : MQTTEExchange	
VARIABLES :	
Release : Xtarget	Ver :1.00
Author :	Date:28/12/2016
Note :	Page:1 of 1

```

VAR
MQTT : MQTTClient; (* MQTT client FB *)
TopicRxD : STRING[ 64 ]; (* Topic received buffer *)
ValueRxD : STRING[ 64 ]; (* Value received buffer *)
CaseNr : USINT; (* Program case *)
Topic : STRING[ 32 ] := 'IOCommand'; (* Topic name *)
DiPulse : BOOL; (* Digital input pulse *)
END_VAR
    
```

```

1 (* ***** *)
2 (* PROGRAM "MQTTEExchange" *)
3 (* ***** *)
4 (* Questo programma si connette ad un broker gratuito, esegue pubblicazione *)
5 (* dello stato di un ingresso logico sul topic "IOCommand". *)
6 (* Il programma si sottoscrive allo stesso topic così ne riceve dal broker lo *)
7 (* stato ad ogni variazione, e lo stato ricevuto comanda una uscita logica. *)
8 (* In questo modo lo stato dell'ingresso è riportato sull'uscita passando dal *)
9 (* broker. Realizzando un esempio di scambio dati tra diversi sistemi. *)
10 (* ----- *)
11
12 (* ----- *)
13 (* INIZIALIZZAZIONI *)
14 (* ----- *)
15 (* Eseguo inizializzazioni. *)
16
17 IF (SysFirstLoop) THEN
18     MQTT.SpyOn:=TRUE; (* Spy active *)
19     MQTT.Server:=ADR('broker.mqttdashboard.com'); (* Broker URL *)
20     MQTT.Port:=1883; (* Broker TCP port *)
21     MQTT.Username:=NULL; (* Broker username *)
22     MQTT.Password:=NULL; (* Broker password *)
23     MQTT.ClientID:=ADR('Elsist'); (* Client identifier *)
24     MQTT.KeepAlive:=180; (* Keep alive time (S) *)
25     MQTT.Flags:=16#00000000; (* Connection/Publish/Subscribe flags *)
26     MQTT.QoS:=1; (* Quality of Service *)
27     MQTT.TBufferRxD:=ADR(TopicRxD); (* Topic buffer (Received) *)
28     MQTT.TBLengthRxD:=SIZEOF(TopicRxD); (* Topic buffer length (Received) *)
29     MQTT.VBufferRxD:=ADR(ValueRxD); (* Value buffer (Received) *)
30     MQTT.VBLengthRxD:=SIZEOF(ValueRxD); (* Value buffer length (Received) *)
31 END_IF;
32
33 (* ----- *)
34 (* MQTT CLIENT MANAGEMENT *)
35 (* ----- *)
36 (* Gestione FB MQTTClient. *)
37
38 MQTT(); (* MQTTClient management *)
39 IF (MQTT.Fault) THEN CaseNr:=0; END_IF;
40 IF NOT(MQTT.Connected) THEN CaseNr:=0; END_IF;
41 Do01CPU:=MQTT.Connected; (* Connected to broker *)
42
43 (* ----- *)
44 (* GESTIONE RICEZIONE TOPIC *)
45 (* ----- *)
46 (* Gestione ricezione topic da broker. *)
47
48 IF (MQTT.TopicRxD) THEN
    
```

Project : MQTTEExchange	
PROGRAM : MQTTEExchange	
Release : Xtarget	Ver :1.00
Author :	Date:28/12/2016
Note :	Page:1 of 3

PROGRAM MQTTEExchange

```

49
50 (* Eseguo controllo se topic corretto. *)
51
52 IF (SysStrFind(ADR(TopicRxD),ADR(Topic), FIND_DEFAULT) <> NULL) THEN
53
54     (* Eseguo controllo se comando "Off". *)
55
56     IF (SysStrFind(ADR(ValueRxD),ADR('Off'), FIND_DEFAULT) <> NULL) THEN
57         Do00CPU:=FALSE;
58     END_IF;
59
60     (* Eseguo controllo se comando "On". *)
61
62     IF (SysStrFind(ADR(ValueRxD),ADR('On'), FIND_DEFAULT) <> NULL) THEN
63         Do00CPU:=TRUE;
64     END_IF;
65 END_IF;
66 END_IF;
67
68 (* ----- *)
69 (* GESTIONE CASES PROGRAMMA *)
70 (* ----- *)
71 (* Gestione cases programma. *)
72
73 CASE (CaseNr) OF
74
75     (* ----- *)
76     (* CONNECTION TO BROKER *)
77     (* ----- *)
78     (* Eseguo connessione al broker. *)
79
80     0:
81     MQTT.Enable:=Di01CPU; (* Connection enable *)
82     IF NOT(Di01CPU) THEN RETURN; END_IF;
83
84     (* Se broker già connesso gestisco publish. *)
85
86     IF (MQTT.Connected) THEN CaseNr:=10; RETURN; END_IF;
87     CaseNr:=CaseNr+1; (* Program case *)
88
89     (* ----- *)
90     (* Eseguo attesa connessione al broker. *)
91
92     1:
93     IF NOT(MQTT.Connected) THEN RETURN; END_IF;
94     CaseNr:=10; (* Program case *)
95
96     (* ----- *)
97     (* TOPIC SUBSCRIBE *)
98     (* ----- *)
99     (* Eseguo sottoscrizione al topic. *)
100
101     10:
102     MQTT.Topic:=ADR(Topic); (* Topic name *)
103     MQTT.Subscribe:=TRUE; (* Topic subscribe *)
104     CaseNr:=CaseNr+1; (* Program case *)
105
106     (* ----- *)
107     (* Eseguo attesa sottoscrizione al topic. *)
108

```

Project : MQTTEExchange	
PROGRAM : MQTTEExchange	
Release : Xtarget	Ver :1.00
Author :	Date:28/12/2016
Note :	Page:2 of 3

PROGRAM MQTTEExchange

```

109      11:
110      MQTT.Subscribe:=FALSE; (* Topic subscribe *)
111      IF NOT(MQTT.Subscribed) THEN RETURN; END_IF;
112      CaseNr:=20; (* Program case *)
113
114      (* ----- *)
115      (* TOPIC PUBLISH *)
116      (* ----- *)
117      (* Eseguo pubblicazione evento pulsante. La pubblicazione avviene *)
118      (* sul cambiamento di stato del pulsante. *)
119
120      20:
121      IF (Di00CPU = DiPulse) THEN RETURN; END_IF;
122      DiPulse:=Di00CPU; (* Digital input pulse *)
123
124      (* Definisco nome topic da pubblicare. *)
125
126      MQTT.Topic:=ADR(Topic); (* Topic name *)
127
128      (* Definisco valore da pubblicare in base allo stato ingresso. *)
129
130      IF NOT(Di00CPU) THEN MQTT.Value:=ADR('Off'); END_IF;
131      IF (Di00CPU) THEN MQTT.Value:=ADR('On'); END_IF;
132
133      (* Definisco lunghezza valore da pubblicare. *)
134
135      MQTT.VLength:=Sysstrlen(MQTT.Value); (* Value length *)
136      MQTT.Publish:=TRUE; (* Topic publish *)
137      CaseNr:=CaseNr+1; (* Program case *)
138
139      (* ----- *)
140      (* Eseguo attesa pubblicazione topic. *)
141
142      21:
143      MQTT.Publish:=FALSE; (* Topic publish *)
144      IF NOT(MQTT.Published) THEN RETURN; END_IF;
145      CaseNr:=0; (* Program case *)
146      END_CASE;
147
148      (* [End of file] *)
149
150

```

	Project : MQTTEExchange	
	PROGRAM : MQTTEExchange	
	Release : Xtarget	Ver :1.00
	Author :	Date:28/12/2016
	Note :	Page:3 of 3

(eLLabNetworkLib_A400) Client for a MQTT server
 ENCRYPTED CODE

```

VAR_INPUT
Enable : BOOL;
SpyOn : BOOL; (* Spy active *)
Publish : BOOL; (* Publish command *)
Subscribe : BOOL; (* Subscribe command *)
Server : @USINT; (* Server URL or IP *)
Port : UINT; (* Server port *)
Username : @USINT; (* Username *)
Password : @USINT; (* Password *)
ClientID : @USINT; (* Client identifier string *)
KeepAlive : UINT; (* Keep Alive time (S) *)
Flags : DWORD; (* Protocol flags *)
QoS : USINT; (* Quality of service *)
Topic : @USINT; (* Topic name *)
Value : @USINT; (* Topic value *)
VLength : UDINT; (* Topic value length *)
TBufferRxD : @USINT; (* Topic buffer received *)
TLengthRxD : UDINT; (* Topic buffer length received *)
VBufferRxD : @USINT; (* Value buffer received *)
VLengthRxD : UDINT; (* Value buffer length received *)
END_VAR
    
```

```

VAR_OUTPUT
Connected : BOOL; (* Connected to server *)
Fault : BOOL; (* Execution fault *)
Published : BOOL; (* Topic published (Pulse) *)
Subscribed : BOOL; (* Topic subscribed (Pulse) *)
TopicRxD : BOOL; (* Topic received (Pulse) *)
VLengthRxD : UINT; (* Topic value length received *)
CTime : REAL; (* Communication time (S) *)
END_VAR
    
```

```

VAR_EXTERNAL
SysActTaskID : USINT; (* Task ID *)
END_VAR
    
```

1

Project : MQTTExchange	
FUNCTION BLOCK : MQTTClient	
Release : Xtarget	Ver :1.00
Author :	Date:28/12/2016
Note :	Page:1 of 1