

VAR

```
i : UDINT; (* Auxiliary variable *)
Result : STRING[ 64 ]; (* Result string *)
CaseNr : USINT; (* Case gestione *)
HTTPRq : HTTPClient; (* HTTP client *)
Request : STRING[ 64 ];
IdProbe : INT := 10;
Temperature : REAL := 20.5;
Di00CPU AT %IX255.0 : BOOL;
END_VAR
```

```
1 // *****
2 // PROGRAMMA SU SERVER HTTP
3 // *****
4 // Esempio di utilizzo HTTPClient con dati JSON. Ecco il programma sul server
5 // HTTP copiare le righe e salvarle nel server nel file "Test.php".
6
7 // Per testare lo script da browser è possibile passare i parametri in GET,
8 // per questo motivo ho gestito la variabile "Post".
9 // Da browser digitare http://URL/Test.php?Post={"IdProbe":10, "Temperature":20.5}
10
11 /*
12 if (isset($_REQUEST['Post']))
13     $RxMessage=$_REQUEST['Post']; //Messaggio ricevuto
14 else
15     $RxMessage=file_get_contents("php://input"); //Messaggio ricevuto
16
17 //echo $RxMessage;
18
19 // -----
20 // CONTROLLO RICHIESTA IN ARRIVO
21 // -----
22 // Acquisisco richiesta JSON ricevuta.
23
24 $ARx=json_decode($RxMessage, true); //Rx data array
25 echo "IdProbe=".$ARx['IdProbe'].", Temperature=".$ARx['Temperature'];
26 */
27
28 // *****
29 // PROGRAMMA SU SLIMLINE
30 // *****
31
32 (* ----- *)
33 (* GESTIONE SERVIZIO HTTP *)
34 (* ----- *)
35 (* Eseguo inizializzazione variabili. *)
36
37 IF (SysFirstLoop) THEN
38     HTTPRq.SpyOn:=TRUE; (* Spy On *)
39     HTTPRq.Method:=TRUE; (* POST *)
40
41     HTTPRq.HostAddress:=ADR('URL...'); (* Host address servizio REST *)
42     HTTPRq.HostName:=HTTPRq.HostAddress; (* Host name servizio REST *)
43     HTTPRq.Page:=ADR('Test.php'); (* Pagina servizio REST *)
44
45     HTTPRq.HostPort:=80; (* Porta servizio REST *)
46
```

Project : HTTPClientExample

PROGRAM : Main

Release : Xtarget

Ver :1.00

Author :

Date:10/07/2018

Note :

Page:1 of 2

PROGRAM Main

```

47      (* Eseguo compilazione dati da inviare al server in JSON. *)
48      i:=SysVarsnprintf(ADR(Request), SIZEOF(Request), '{"IdProbe":%d, ', INT_TYPE, ADR(IdProbe));
49      i:=SysLWVarsnprintf(ADR(Request), SIZEOF(Request), '"Temperature":%.1f}', REAL_TYPE, ADR(Tempe
ature));
50      HTTPRq.Request:=ADR(Request);
51
52      HTTPRq.DBSize:=512;
53      HTTPRq.Timeout:=5000; (* Timeout 5 s *)
54  END_IF;
55
56  HTTPRq(); (* HTTP client *)
57
58  (* ----- *)
59  (* GESTIONE SERVIZIO HTTP *)
60  (* ----- *)
61  (* Case gestione sequenze programma. *)
62
63  CASE (CaseNr) OF
64
65      (* ----- *)
66      (* Se comando attivo aspetto si disattivi. *)
67
68      0:
69      HTTPRq.Enable:=FALSE; (* HTTP get page enable *)
70      IF NOT(Di00CPU) THEN CaseNr:=CaseNr+1; END_IF;
71
72      (* ----- *)
73      (* Attesa comando, controllo fronte salita. *)
74
75      1:
76      IF NOT(Di00CPU) THEN RETURN; END_IF;
77      i:=Systemset(ADR(Result), 0, SIZEOF(Result)); (* Empty result *)
78      HTTPRq.Enable:=TRUE; (* HTTP get page enable *)
79      CaseNr:=CaseNr+1; (* Case gestione *)
80
81      (* ----- *)
82      (* Acquisizione pagina, su ricezione eseguo trasferimento in stringa. *)
83
84      2:
85      IF (HTTPRq.DBChars <> 0) THEN
86          IF ((Sysstrlen(ADR(Result))+HTTPRq.DBChars) < SIZEOF(Result)) THEN
87              i:=Systemmove(ADR(Result)+Sysstrlen(ADR(Result)), HTTPRq.DBAddress, HTTPRq.DBChars);
88              END_IF;
89      END_IF;
90
91      (* Se Done fine esecuzione. *)
92      IF (HTTPRq.Done) THEN
93          CaseNr:=0; (* Case gestione *)
94          IF NOT(HTTPRq.PLoad) THEN RETURN; END_IF; (* Se Done ma non Ok la richiesta HTTP è in error
e. *)
95      END_IF;
96  END_CASE;
97
98  (* [End of file] *)
99

```

	Project : HTTPClientExample	
	PROGRAM : Main	
	Release : Xtarget	Ver :1.00
	Author :	Date:10/07/2018
	Note :	Page:2 of 2