

```

1  ....// *****
2  ....// PROGRAMMA ORIGINALE
3  ....// *****
4
5  ....// -----
6  ....// INITIALIZATION
7  ....// -----
8  ....// Program initializations.
9
10 ....IF (SysFirstLoop) THEN
11
12 .....// Serial port settings.
13
14 .....Sp.COM:=ADR('COM2'); //COM port definition
15 .....Sp.Baudrate:=9600; //Baudrate
16 .....Sp.Parity:='N'; //Parity
17 .....Sp.DataBits:=8; //Data bits
18 .....Sp.StopBits:=1; //Stop bits
19 .....Sp.DTRManagement:=DTR_AUTO_WO_TIMES; //DTR management
20 .....Sp.DTRComplement:=FALSE; //DTR complement
21 .....Sp.EchoFlush:=FALSE; //Received echo flush
22 .....Sp.DTROnTime:=0; //DTR On time delay (mS)
23 .....Sp.DTROffTime:=0; //DTR Off time delay (mS)
24 .....Sp.FlushTm:=0; //Flush time (mS)
25 .....Sp.RxSize:=0; //Rx buffer size
26 .....Sp.TxSize:=0; //Tx buffer size
27
28 .....// TCP client settings.
29 .....// Port 502 to connect to embed Modbus slave.
30 .....// Port 3000 to connect to ModbusSlave FB example.
31
32 .....TCPClient.PeerAdd:=ADR('127.0.0.1'); //Peer address
33 .....TCPClient.PeerPort:=3000; //Peer port
34 .....TCPClient.LocalAdd:=ADR('0.0.0.0'); //Local address
35 .....TCPClient.LocalPort:=0; //Local port
36 .....TCPClient.FlushTm:=50; //Flush time (mS)
37 .....TCPClient.LifeTm:=20; //Life time (S)
38 .....TCPClient.RxSize:=128; //Rx buffer size
39 .....TCPClient.TxSize:=128; //Tx buffer size
40
41 .....// Modbus master settings.
42
43 .....MMdb.Absolute:=FALSE; //Absolute addressing
44 .....MMdb.Timeout:=T#1s; //Timeout time
45 .....MMdb.Delay:=T#100ms; //Delay time
46 ....END_IF;
47
48 ....// -----
49 ....// MODBUS MANAGEMENT
50 ....// -----
51 ....// Eseguo selezione canale comunicazione.
52
53 ....IF NOT(SerialOrTCP) THEN
54
55 .....// Serial port management.
56
57 .....Sp(Open:=TRUE); //Serial port management
58 .....MMdb.File:=Sp.File; //File pointer
59 .....MMdb.Type:=MODBUS_PROTOCOL#MDB_RTU; //Modbus protocol type
60 .....// MMdb.Type:=MODBUS_PROTOCOL#MDB_ASCII; //Modbus protocol type
61 ....ELSE
62
63 .....// TCP client management.
64
65 .....TCPClient(Connect:=TRUE); //TCPClient management
66 .....MMdb.File:=TCPClient.File; //File pointer
67 .....MMdb.Type:=MODBUS_PROTOCOL#MDB_TCP; //Modbus protocol type
68 ....END_IF;
69
70 ....// Manage the modbus master communication.
71
72 ....MMdb(); //Modbus master
73 ....IF NOT(SysFIsOpen(MMdb.File)) THEN MMdb.Enable:=FALSE; CaseNr:=0; RETURN; END_IF;

```

```

74
75 // -----
76 // PROGRAM CASES
77 // -----
78 // Manage the program cases.
79
80 CASE (CaseNr) OF
81
82     // LEGGI
83     // -----
84     // TERMOSTATO
85     // TMST_Temp_ambiente
86     0:
87     MMdb.Node := 4;
88     MMdb.Enable:=TRUE; //Modbus enable
89     MMdb.FCode:=16#03; //Modbus function code
90     MMdb.Address:=49; //Modbus register address
91     MMdb.Points:=1; //Modbus register points
92     MMdb.Buffer:=ADR(TMST_Temp_ambiente); //Memory buffer address
93     IF NOT(MMdb.Done) THEN RETURN; END_IF;
94     MMdb.Enable:=FALSE; //Modbus enable
95     CaseNr:=CaseNr+1; //Program case
96
97     // TMST_Setpoint
98     1:
99     MMdb.Node := 4;
100    MMdb.Enable:=TRUE; //Modbus enable
101    MMdb.FCode:=16#03; //Modbus function code
102    MMdb.Address:=385; //Modbus register address
103    MMdb.Points:=1; //Modbus register points
104    MMdb.Buffer:=ADR(TMST_Setpoint); //Memory buffer address
105    IF NOT(MMdb.Done) THEN RETURN; END_IF;
106    MMdb.Enable:=FALSE; //Modbus enable
107    CaseNr:=CaseNr+3; //Program case
108
109    // PDC
110    // PDC_stagione_da
111    //2:
112    //MMdb.Node := 5;
113    //MMdb.Enable:=TRUE; //Modbus enable
114    //MMdb.FCode:=16#03; //Modbus function code
115    //MMdb.Address:=384; //Modbus register address
116    //MMdb.Points:=1; //Modbus register points
117    // MMdb.Buffer:=ADR(PDC_stagione_da); //Memory buffer address
118    //IF NOT(MMdb.Done) THEN RETURN; END_IF;
119    //MMdb.Enable:=FALSE; //Modbus enable
120    //CaseNr:=CaseNr+1 //Program case
121
122    // PDC_allarme
123    //3:
124    //MMdb.Node := 5;
125    //MMdb.Enable:=TRUE; //Modbus enable
126    //MMdb.FCode:=16#03; //Modbus function code
127    //MMdb.Address:=385; //Modbus register address
128    //MMdb.Points:=1; //Modbus register points
129    // MMdb.Buffer:=ADR(PDC_allarme); //Memory buffer address
130    //IF NOT(MMdb.Done) THEN RETURN; END_IF;
131    //MMdb.Enable:=FALSE; //Modbus enable
132    //CaseNr:=CaseNr+1; //Program case
133
134    // CAPP
135    // CAPP_allarme
136    4:
137    MMdb.Node := 2;
138    MMdb.Enable:=TRUE; //Modbus enable
139    MMdb.FCode:=16#03; //Modbus function code
140    MMdb.Address:=2910; //Modbus register address
141    MMdb.Points:=1; //Modbus register points
142    MMdb.Buffer:=ADR(CAPP_registro_allarme); //Memory buffer address
143    IF NOT(MMdb.Done) THEN RETURN; END_IF;
144    MMdb.Enable:=FALSE; //Modbus enable
145    CaseNr:=CaseNr+1; //Program case
146

```

```

147 ..... // WILO
148 ..... // WILO_allarme
149 ..... 5:
150 ..... MMdb.Node := 3;
151 ..... MMdb.Enable:=TRUE; //Modbus enable
152 ..... MMdb.FCode:=16#04; //Modbus function code
153 ..... MMdb.Address:=941; //Modbus register address
154 ..... MMdb.Points:=1; //Modbus register points
155 ..... MMdb.Buffer:=ADR(POMPA_allarme); //Memory buffer address
156 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
157 ..... MMdb.Enable:=FALSE; //Modbus enable
158 ..... CaseNr:=CaseNr+1; //Program case
159 .....
160 ..... // -----
161 ..... // SCRIVI
162 ..... // -----
163 ..... // TERMOSTATO
164 ..... // TMST_Stagione
165 ..... 6:
166 ..... MMdb.Node := 4;
167 ..... MMdb.Enable:=TRUE; //Modbus enable
168 ..... MMdb.FCode:=16#06; //Modbus function code
169 ..... MMdb.Address:=99; //Modbus register address
170 ..... MMdb.Points:=1; //Modbus register points
171 ..... MMdb.Buffer:=ADR(TMST_Stagione); //Memory buffer address
172 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
173 ..... MMdb.Enable:=FALSE; //Modbus enable
174 ..... CaseNr:=CaseNr+1; //Program case
175 .....
176 ..... // POMPA
177 ..... // POMPA_on_off
178 ..... 7:
179 ..... MMdb.Node := 3;
180 ..... MMdb.Enable:=TRUE; //Modbus enable
181 ..... MMdb.FCode:=16#06; //Modbus function code
182 ..... MMdb.Address:=41; //Modbus register address
183 ..... MMdb.Points:=1; //Modbus register points
184 ..... MMdb.Buffer:=ADR(POMPA_on_off); //Memory buffer address
185 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
186 ..... MMdb.Enable:=FALSE; //Modbus enable
187 ..... CaseNr:=CaseNr+1; //Program case
188 .....
189 ..... // POMPA_setpoint
190 ..... 8:
191 ..... MMdb.Node := 3;
192 ..... MMdb.Enable:=TRUE; //Modbus enable
193 ..... MMdb.FCode:=16#06; //Modbus function code
194 ..... MMdb.Address:=2; //Modbus register address
195 ..... MMdb.Points:=1; //Modbus register points
196 ..... MMdb.Buffer:=ADR(POMPA_setpoint); //Memory buffer address
197 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
198 ..... MMdb.Enable:=FALSE; //Modbus enable
199 ..... CaseNr:=CaseNr+1; //Program case
200 .....
201 .....
202 ..... // FANCOIL
203 ..... // FC_1_Setpoint_summer
204 ..... 9:
205 ..... MMdb.Node := 6;
206 ..... MMdb.Enable:=TRUE; //Modbus enable
207 ..... MMdb.FCode:=16#06; //Modbus function code
208 ..... MMdb.Address:=4142; //Modbus register address
209 ..... MMdb.Points:=1; //Modbus register points
210 ..... MMdb.Buffer:=ADR(FC_Setpoint_summer); //Memory buffer address
211 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
212 ..... MMdb.Enable:=FALSE; //Modbus enable
213 ..... CaseNr:=CaseNr+1; //Program case
214 .....
215 ..... // FC_1_Setpoint_winter
216 ..... 10:
217 ..... MMdb.Node := 6;
218 ..... MMdb.Enable:=TRUE; //Modbus enable
219 ..... MMdb.FCode:=16#06; //Modbus function code

```

```

220 ..... MMdb.Address:=4143; //Modbus register address
221 ..... MMdb.Points:=1; //Modbus register points
222 ..... MMdb.Buffer:=ADR(FC_Setpoint_winter); //Memory buffer address....
223 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
224 ..... MMdb.Enable:=FALSE; //Modbus enable
225 ..... CaseNr:=CaseNr+1; //Program case
226 .....
227 ..... // FC_1_on_off
228 ..... 11:
229 ..... MMdb.Node := 6;
230 ..... MMdb.Enable:=TRUE; //Modbus enable
231 ..... MMdb.FCode:=16#06; //Modbus function code
232 ..... MMdb.Address:=4189; //Modbus register address
233 ..... MMdb.Points:=1; //Modbus register points
234 ..... MMdb.Buffer:=ADR(FC_on_off); //Memory buffer address....
235 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
236 ..... MMdb.Enable:=FALSE; //Modbus enable
237 ..... CaseNr:=CaseNr+1; //Program case
238 .....
239 ..... // FC_1_FC_stagione
240 ..... 12:
241 ..... MMdb.Node := 6;
242 ..... MMdb.Enable:=TRUE; //Modbus enable
243 ..... MMdb.FCode:=16#06; //Modbus function code
244 ..... MMdb.Address:=4190; //Modbus register address
245 ..... MMdb.Points:=1; //Modbus register points
246 ..... MMdb.Buffer:=ADR(FC_stagione); //Memory buffer address....
247 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
248 ..... MMdb.Enable:=FALSE; //Modbus enable
249 ..... CaseNr:=CaseNr+1; //Program case
250 .....
251 ..... // FC_2_Setpoint_summer
252 ..... 13:
253 ..... MMdb.Node := 7;
254 ..... MMdb.Enable:=TRUE; //Modbus enable
255 ..... MMdb.FCode:=16#06; //Modbus function code
256 ..... MMdb.Address:=4142; //Modbus register address
257 ..... MMdb.Points:=1; //Modbus register points
258 ..... MMdb.Buffer:=ADR(FC_Setpoint_summer); //Memory buffer address....
259 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
260 ..... MMdb.Enable:=FALSE; //Modbus enable
261 ..... CaseNr:=CaseNr+1; //Program case
262 .....
263 ..... // FC_2_Setpoint_winter
264 ..... 14:
265 ..... MMdb.Node := 7;
266 ..... MMdb.Enable:=TRUE; //Modbus enable
267 ..... MMdb.FCode:=16#06; //Modbus function code
268 ..... MMdb.Address:=4143; //Modbus register address
269 ..... MMdb.Points:=1; //Modbus register points
270 ..... MMdb.Buffer:=ADR(FC_Setpoint_winter); //Memory buffer address....
271 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
272 ..... MMdb.Enable:=FALSE; //Modbus enable
273 ..... CaseNr:=CaseNr+1; //Program case
274 .....
275 ..... // FC_2_on_off
276 ..... 15:
277 ..... MMdb.Node := 7;
278 ..... MMdb.Enable:=TRUE; //Modbus enable
279 ..... MMdb.FCode:=16#06; //Modbus function code
280 ..... MMdb.Address:=4189; //Modbus register address
281 ..... MMdb.Points:=1; //Modbus register points
282 ..... MMdb.Buffer:=ADR(FC_on_off); //Memory buffer address....
283 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
284 ..... MMdb.Enable:=FALSE; //Modbus enable
285 ..... CaseNr:=CaseNr+1; //Program case
286 .....
287 ..... // FC_2_FC_stagione
288 ..... 16:
289 ..... MMdb.Node := 7;
290 ..... MMdb.Enable:=TRUE; //Modbus enable
291 ..... MMdb.FCode:=16#06; //Modbus function code
292 ..... MMdb.Address:=4190; //Modbus register address

```

```

293 ..... MMdb.Points:=1; //Modbus register points
294 ..... MMdb.Buffer:=ADR(FC_stagione); //Memory buffer address.....
295 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
296 ..... MMdb.Enable:=FALSE; //Modbus enable
297 ..... CaseNr:=CaseNr+1; //Program case
298
299 ..... // CANALA
300 ..... // CNL_stagione_a
301 ..... 17:
302 ..... MMdb.Node := 1;
303 ..... MMdb.Enable:=TRUE; //Modbus enable
304 ..... MMdb.FCode:=16#06; //Modbus function code
305 ..... MMdb.Address:=4185; //Modbus register address
306 ..... MMdb.Points:=1; //Modbus register points
307 ..... MMdb.Buffer:=ADR(CNL_stagione_a); //Memory buffer address.....
308 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
309 ..... MMdb.Enable:=FALSE; //Modbus enable
310 ..... CaseNr:=CaseNr+1; //Program case
311
312 ..... // CNL_setpoint_winter
313 ..... 18:
314 ..... MMdb.Node := 8;
315 ..... MMdb.Enable:=TRUE; //Modbus enable
316 ..... MMdb.FCode:=16#06; //Modbus function code
317 ..... MMdb.Address:=4143; //Modbus register address
318 ..... MMdb.Points:=1; //Modbus register points
319 ..... MMdb.Buffer:=ADR(CNL_setpoint_inverno); //Memory buffer address.....
320 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
321 ..... MMdb.Enable:=FALSE; //Modbus enable
322 ..... CaseNr:=CaseNr+1; //Program case
323
324 ..... // CNL_setpoint_summer
325 ..... 19:
326 ..... MMdb.Node := 8;
327 ..... MMdb.Enable:=TRUE; //Modbus enable
328 ..... MMdb.FCode:=16#06; //Modbus function code
329 ..... MMdb.Address:=4142; //Modbus register address
330 ..... MMdb.Points:=1; //Modbus register points
331 ..... MMdb.Buffer:=ADR(CNL_setpoint_estate); //Memory buffer address.....
332 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
333 ..... MMdb.Enable:=FALSE; //Modbus enable
334 ..... CaseNr:=CaseNr+1; //Program case
335
336 ..... // CNL_on_off_set
337 ..... 20:
338 ..... MMdb.Node := 1;
339 ..... MMdb.Enable:=TRUE; //Modbus enable
340 ..... MMdb.FCode:=16#06; //Modbus function code
341 ..... MMdb.Address:=4184; //Modbus register address
342 ..... MMdb.Points:=1; //Modbus register points
343 ..... MMdb.Buffer:=ADR(CNL_on_off_set); //Memory buffer address.....
344 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
345 ..... MMdb.Enable:=FALSE; //Modbus enable
346 ..... CaseNr:=CaseNr+1; //Program case
347
348 ..... // CNL_fan_speed_set
349 ..... 21:
350 ..... MMdb.Node := 1;
351 ..... MMdb.Enable:=TRUE; //Modbus enable
352 ..... MMdb.FCode:=16#06; //Modbus function code
353 ..... MMdb.Address:=4186; //Modbus register address
354 ..... MMdb.Points:=1; //Modbus register points
355 ..... MMdb.Buffer:=ADR(CNL_fan_speed_set); //Memory buffer address.....
356 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
357 ..... MMdb.Enable:=FALSE; //Modbus enable
358 ..... CaseNr:=CaseNr+1; //Program case
359
360 ..... // CAPPA
361 ..... // CAPPA_on_off_set
362 ..... 22:
363 ..... MMdb.Node := 2;
364 ..... MMdb.Enable:=TRUE; //Modbus enable
365 ..... MMdb.FCode:=16#06; //Modbus function code

```

```
366 .....MMdb.Address:=2810; //Modbus register address
367 .....MMdb.Points:=1; //Modbus register points
368 .....MMdb.Buffer:=ADR(CAPPA_mode_set); //Memory buffer address.....
369 .....IF NOT(MMdb.Done) THEN RETURN; END_IF;
370 .....MMdb.Enable:=FALSE; //Modbus enable
371 .....CaseNr:=0; //Program case
372 .....END_CASE;
373
```

```

1  ....// *****
2  ....// PROGRAMMA MODIFICATO
3  ....// *****
4
5  ....// -----
6  ....// INITIALIZATION
7  ....// -----
8  ....// Program initializations.
9
10 IF (SysFirstLoop) OR FirstLoop THEN
11
12 .....// Serial port settings.
13
14 .....Sp.COM:=ADR('COM2'); //COM port definition
15 .....Sp.Baudrate:=9600; //Baudrate
16 .....Sp.Parity:='N'; //Parity
17 .....Sp.DataBits:=8; //Data bits
18 .....Sp.StopBits:=1; //Stop bits
19 .....Sp.DTRManagement:=DTR_AUTO_WO_TIMES; //DTR management
20 .....Sp.DTRComplement:=FALSE; //DTR complement
21 .....Sp.EchoFlush:=FALSE; //Received echo flush
22 .....Sp.DTROnTime:=0; //DTR On time delay (mS)
23 .....Sp.DTROffTime:=0; //DTR Off time delay (mS)
24 .....Sp.FlushTm:=0; //Flush time (mS)
25 .....Sp.RxSize:=0; //Rx buffer size
26 .....Sp.TxSize:=0; //Tx buffer size
27
28 .....// TCP client settings.
29 .....// Port 502 to connect to embed Modbus slave.
30 .....// Port 3000 to connect to ModbusSlave FB example.
31
32 .....TCPClient.PeerAdd:=ADR('127.0.0.1'); //Peer address
33 .....TCPClient.PeerPort:=3000; //Peer port
34 .....TCPClient.LocalAdd:=ADR('0.0.0.0'); //Local address
35 .....TCPClient.LocalPort:=0; //Local port
36 .....TCPClient.FlushTm:=50; //Flush time (mS)
37 .....TCPClient.LifeTm:=20; //Life time (S)
38 .....TCPClient.RxSize:=128; //Rx buffer size
39 .....TCPClient.TxSize:=128; //Tx buffer size
40
41 .....// Modbus master settings.
42
43 .....MMdb.Absolute:=FALSE; //Absolute addressing
44 .....MMdb.Timeout:=T#1s; //Timeout time
45 .....MMdb.Delay:=T#100ms; //Delay time
46 END_IF;
47
48 IF SecondLoop THEN
49
50 .....// Serial port settings.
51
52 .....Sp(Open:=FALSE);
53 .....Sp.Baudrate:=19200; //Baudrate
54
55 .....SecondLoop := FALSE;
56 END_IF;
57
58 ....// -----
59 ....// MODBUS MANAGEMENT
60 ....// -----
61 ....// Eseguo selezione canale comunicazione.
62
63 IF NOT(SerialOrTCP) THEN
64
65 .....// Serial port management.
66
67 .....Sp(Open:=TRUE); //Serial port management
68 .....MMdb.File:=Sp.File; //File pointer
69 .....MMdb.Type:=MODBUS_PROTOCOL#MDB_RTU; //Modbus protocol type
70 .....// MMdb.Type:=MODBUS_PROTOCOL#MDB_ASCII; //Modbus protocol type
71 ELSE
72
73 .....// TCP client management.

```

```

74
75 ..... TCPClient(Connect:=TRUE); //TCPClient management
76 ..... MMdb.File:=TCPClient.File; //File pointer
77 ..... MMdb.Type:=MODBUS_PROTOCOL#MDB_TCP; //Modbus protocol type
78 ..... END_IF;
79
80 ..... // Manage the modbus master communication.
81
82 ..... MMdb(); //Modbus master
83 ..... IF NOT(SysFIsOpen(MMdb.File)) THEN MMdb.Enable:=FALSE; CaseNr:=0; RETURN; END_IF;
84
85 ..... // -----
86 ..... // PROGRAM CASES
87 ..... // -----
88 ..... // Manage the program cases.
89
90 ..... CASE (CaseNr) OF
91 .....
92 .....     // LEGGI
93 .....     // -----
94 .....     // TERMOSTATO
95 .....     // TMST_Temp_ambiente
96 .....     0:
97 .....     MMdb.Node := 4;
98 .....     MMdb.Enable:=TRUE; //Modbus enable
99 .....     MMdb.FCode:=16#03; //Modbus function code
100 .....     MMdb.Address:=49; //Modbus register address
101 .....     MMdb.Points:=1; //Modbus register points
102 .....     MMdb.Buffer:=ADR(TMST_Temp_ambiente); //Memory buffer address
103 .....     IF NOT(MMdb.Done) THEN RETURN; END_IF;
104 .....     MMdb.Enable:=FALSE; //Modbus enable
105 .....     CaseNr:=CaseNr+1; //Program case
106 .....
107 .....     // TMST_Setpoint
108 .....     1:
109 .....     MMdb.Node := 4;
110 .....     MMdb.Enable:=TRUE; //Modbus enable
111 .....     MMdb.FCode:=16#03; //Modbus function code
112 .....     MMdb.Address:=385; //Modbus register address
113 .....     MMdb.Points:=1; //Modbus register points
114 .....     MMdb.Buffer:=ADR(TMST_Setpoint); //Memory buffer address
115 .....     IF NOT(MMdb.Done) THEN RETURN; END_IF;
116 .....     MMdb.Enable:=FALSE; //Modbus enable
117 .....     CaseNr:=CaseNr+3; //Program case
118 .....
119 .....     // PDC
120 .....     // PDC_stagione_da
121 .....     //2:
122 .....     //MMdb.Node := 5;
123 .....     //MMdb.Enable:=TRUE; //Modbus enable
124 .....     //MMdb.FCode:=16#03; //Modbus function code
125 .....     //MMdb.Address:=384; //Modbus register address
126 .....     //MMdb.Points:=1; //Modbus register points
127 .....     // MMdb.Buffer:=ADR(PDC_stagione_da); //Memory buffer address
128 .....     //IF NOT(MMdb.Done) THEN RETURN; END_IF;
129 .....     //MMdb.Enable:=FALSE; //Modbus enable
130 .....     //CaseNr:=CaseNr+1 //Program case
131 .....
132 .....     // PDC_allarme
133 .....     //3:
134 .....     //MMdb.Node := 5;
135 .....     //MMdb.Enable:=TRUE; //Modbus enable
136 .....     //MMdb.FCode:=16#03; //Modbus function code
137 .....     //MMdb.Address:=385; //Modbus register address
138 .....     //MMdb.Points:=1; //Modbus register points
139 .....     // MMdb.Buffer:=ADR(PDC_allarme); //Memory buffer address
140 .....     //IF NOT(MMdb.Done) THEN RETURN; END_IF;
141 .....     //MMdb.Enable:=FALSE; //Modbus enable
142 .....     //CaseNr:=CaseNr+1; //Program case
143 .....
144 .....     // CAPPA
145 .....     // CAPPA_allarme
146 .....     4:

```

```

147 ..... MMdb.Node := 2;
148 ..... MMdb.Enable:=TRUE; //Modbus enable
149 ..... MMdb.FCode:=16#03; //Modbus function code
150 ..... MMdb.Address:=2910; //Modbus register address
151 ..... MMdb.Points:=1; //Modbus register points
152 ..... MMdb.Buffer:=ADR(CAPPA_registro_allarme); //Memory buffer address
153 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
154 ..... MMdb.Enable:=FALSE; //Modbus enable
155 ..... CaseNr:=CaseNr+1; //Program case
156
157 ..... // WILO
158 ..... // WILO_allarme
159 ..... 5:
160 ..... MMdb.Node := 3;
161 ..... MMdb.Enable:=TRUE; //Modbus enable
162 ..... MMdb.FCode:=16#04; //Modbus function code
163 ..... MMdb.Address:=941; //Modbus register address
164 ..... MMdb.Points:=1; //Modbus register points
165 ..... MMdb.Buffer:=ADR(POMPA_allarme); //Memory buffer address
166 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
167 ..... MMdb.Enable:=FALSE; //Modbus enable
168 ..... CaseNr:=CaseNr+1; //Program case
169
170 ..... // -----
171 ..... // SCRIVI
172 ..... // -----
173 ..... // TERMOSTATO
174 ..... // TMST_Stagione
175 ..... 6:
176 ..... MMdb.Node := 4;
177 ..... MMdb.Enable:=TRUE; //Modbus enable
178 ..... MMdb.FCode:=16#06; //Modbus function code
179 ..... MMdb.Address:=99; //Modbus register address
180 ..... MMdb.Points:=1; //Modbus register points
181 ..... MMdb.Buffer:=ADR(TMST_Stagione); //Memory buffer address
182 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
183 ..... MMdb.Enable:=FALSE; //Modbus enable
184 ..... CaseNr:=CaseNr+1; //Program case
185
186 ..... // POMPA
187 ..... // POMPA_on_off
188 ..... 7:
189 ..... MMdb.Node := 3;
190 ..... MMdb.Enable:=TRUE; //Modbus enable
191 ..... MMdb.FCode:=16#06; //Modbus function code
192 ..... MMdb.Address:=41; //Modbus register address
193 ..... MMdb.Points:=1; //Modbus register points
194 ..... MMdb.Buffer:=ADR(POMPA_on_off); //Memory buffer address
195 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
196 ..... MMdb.Enable:=FALSE; //Modbus enable
197 ..... CaseNr:=CaseNr+1; //Program case
198
199 ..... // POMPA_setpoint
200 ..... 8:
201 ..... MMdb.Node := 3;
202 ..... MMdb.Enable:=TRUE; //Modbus enable
203 ..... MMdb.FCode:=16#06; //Modbus function code
204 ..... MMdb.Address:=2; //Modbus register address
205 ..... MMdb.Points:=1; //Modbus register points
206 ..... MMdb.Buffer:=ADR(POMPA_setpoint); //Memory buffer address
207 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
208 ..... MMdb.Enable:=FALSE; //Modbus enable
209 ..... CaseNr:=CaseNr+1; //Program case
210
211 ..... // FANCOIL
212 ..... // FC_1_Setpoint_summer
213 ..... 9:
214 ..... MMdb.Node := 6;
215 ..... MMdb.Enable:=TRUE; //Modbus enable
216 ..... MMdb.FCode:=16#06; //Modbus function code
217 ..... MMdb.Address:=4142; //Modbus register address
218 ..... MMdb.Points:=1; //Modbus register points
219 ..... MMdb.Buffer:=ADR(FC_Setpoint_summer); //Memory buffer address

```

```

220 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
221 ..... MMdb.Enable:=FALSE; //Modbus enable
222 ..... CaseNr:=CaseNr+1; //Program case
223
224 ..... // FC_1_Setpoint_winter
225 ..... 10:
226 ..... MMdb.Node := 6;
227 ..... MMdb.Enable:=TRUE; //Modbus enable
228 ..... MMdb.FCode:=16#06; //Modbus function code
229 ..... MMdb.Address:=4143; //Modbus register address
230 ..... MMdb.Points:=1; //Modbus register points
231 ..... MMdb.Buffer:=ADR(FC_Setpoint_winter); //Memory buffer address
232 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
233 ..... MMdb.Enable:=FALSE; //Modbus enable
234 ..... CaseNr:=CaseNr+1; //Program case
235
236 ..... // FC_1_on_off
237 ..... 11:
238 ..... MMdb.Node := 6;
239 ..... MMdb.Enable:=TRUE; //Modbus enable
240 ..... MMdb.FCode:=16#06; //Modbus function code
241 ..... MMdb.Address:=4189; //Modbus register address
242 ..... MMdb.Points:=1; //Modbus register points
243 ..... MMdb.Buffer:=ADR(FC_on_off); //Memory buffer address
244 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
245 ..... MMdb.Enable:=FALSE; //Modbus enable
246 ..... CaseNr:=CaseNr+1; //Program case
247
248 ..... // FC_1_FC_stagione
249 ..... 12:
250 ..... MMdb.Node := 6;
251 ..... MMdb.Enable:=TRUE; //Modbus enable
252 ..... MMdb.FCode:=16#06; //Modbus function code
253 ..... MMdb.Address:=4190; //Modbus register address
254 ..... MMdb.Points:=1; //Modbus register points
255 ..... MMdb.Buffer:=ADR(FC_stagione); //Memory buffer address
256 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
257 ..... MMdb.Enable:=FALSE; //Modbus enable
258 ..... CaseNr:=CaseNr+1; //Program case
259
260 ..... // FC_2_Setpoint_summer
261 ..... 13:
262 ..... MMdb.Node := 7;
263 ..... MMdb.Enable:=TRUE; //Modbus enable
264 ..... MMdb.FCode:=16#06; //Modbus function code
265 ..... MMdb.Address:=4142; //Modbus register address
266 ..... MMdb.Points:=1; //Modbus register points
267 ..... MMdb.Buffer:=ADR(FC_Setpoint_summer); //Memory buffer address
268 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
269 ..... MMdb.Enable:=FALSE; //Modbus enable
270 ..... CaseNr:=CaseNr+1; //Program case
271
272 ..... // FC_2_Setpoint_winter
273 ..... 14:
274 ..... MMdb.Node := 7;
275 ..... MMdb.Enable:=TRUE; //Modbus enable
276 ..... MMdb.FCode:=16#06; //Modbus function code
277 ..... MMdb.Address:=4143; //Modbus register address
278 ..... MMdb.Points:=1; //Modbus register points
279 ..... MMdb.Buffer:=ADR(FC_Setpoint_winter); //Memory buffer address
280 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
281 ..... MMdb.Enable:=FALSE; //Modbus enable
282 ..... CaseNr:=CaseNr+1; //Program case
283
284 ..... // FC_2_on_off
285 ..... 15:
286 ..... MMdb.Node := 7;
287 ..... MMdb.Enable:=TRUE; //Modbus enable
288 ..... MMdb.FCode:=16#06; //Modbus function code
289 ..... MMdb.Address:=4189; //Modbus register address
290 ..... MMdb.Points:=1; //Modbus register points
291 ..... MMdb.Buffer:=ADR(FC_on_off); //Memory buffer address
292 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;

```

```

293 ..... MMdb.Enable:=FALSE; //Modbus enable
294 ..... CaseNr:=CaseNr+1; //Program case
295
296 ..... // FC_2_FC_stagione
297 ..... 16:
298 ..... MMdb.Node := 7;
299 ..... MMdb.Enable:=TRUE; //Modbus enable
300 ..... MMdb.FCode:=16#06; //Modbus function code
301 ..... MMdb.Address:=4190; //Modbus register address
302 ..... MMdb.Points:=1; //Modbus register points
303 ..... MMdb.Buffer:=ADR(FC_stagione); //Memory buffer address
304 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
305 ..... MMdb.Enable:=FALSE; //Modbus enable
306 ..... CaseNr:=CaseNr+1; //Program case
307
308 ..... // CNL_on_off_set
309 ..... 17:
310 ..... MMdb.Node := 1;
311 ..... MMdb.Enable:=TRUE; //Modbus enable
312 ..... MMdb.FCode:=16#06; //Modbus function code
313 ..... MMdb.Address:=4184; //Modbus register address
314 ..... MMdb.Points:=1; //Modbus register points
315 ..... MMdb.Buffer:=ADR(CNL_on_off_set); //Memory buffer address
316 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
317 ..... MMdb.Enable:=FALSE; //Modbus enable
318 ..... CaseNr:=CaseNr+1; //Program case
319
320 ..... // CNL_fan_speed_set
321 ..... 18:
322 ..... MMdb.Node := 1;
323 ..... MMdb.Enable:=TRUE; //Modbus enable
324 ..... MMdb.FCode:=16#06; //Modbus function code
325 ..... MMdb.Address:=4186; //Modbus register address
326 ..... MMdb.Points:=1; //Modbus register points
327 ..... MMdb.Buffer:=ADR(CNL_fan_speed_set); //Memory buffer address
328 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
329 ..... MMdb.Enable:=FALSE; //Modbus enable
330 ..... CaseNr:=CaseNr+1; //Program case
331
332 ..... // CAPPA
333 ..... // CAPPA_on_off_set
334 ..... 19:
335 ..... MMdb.Node := 2;
336 ..... MMdb.Enable:=TRUE; //Modbus enable
337 ..... MMdb.FCode:=16#06; //Modbus function code
338 ..... MMdb.Address:=2810; //Modbus register address
339 ..... MMdb.Points:=1; //Modbus register points
340 ..... MMdb.Buffer:=ADR(CAPPA_mode_set); //Memory buffer address
341 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
342 ..... MMdb.Enable:=FALSE; //Modbus enable
343 ..... CaseNr:=CaseNr+1; //Program case
344 ..... SecondLoop := TRUE;
345
346 ..... // CANALA
347 ..... // CNL_stagione_a
348 ..... 20:
349 ..... MMdb.Node := 8;
350 ..... MMdb.Enable:=TRUE; //Modbus enable
351 ..... MMdb.FCode:=16#06; //Modbus function code
352 ..... MMdb.Address:=1001; //Modbus register address
353 ..... MMdb.Points:=1; //Modbus register points
354 ..... MMdb.Buffer:=ADR(CNL_stagione_a); //Memory buffer address
355 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
356 ..... MMdb.Enable:=FALSE; //Modbus enable
357 ..... CaseNr:=CaseNr+1; //Program case
358
359 ..... // Richiesta TA (termostato che richiede)
360 ..... 21:
361 ..... MMdb.Node := 8;
362 ..... MMdb.Enable:=TRUE; //Modbus enable
363 ..... MMdb.FCode:=16#06; //Modbus function code
364 ..... MMdb.Address:=3001; //Modbus register address
365 ..... MMdb.Points:=1; //Modbus register points

```

```

366 ..... MMdb.Buffer:=ADR(TMST_Richiesta); //Memory buffer address .....
367 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
368 ..... MMdb.Enable:=FALSE; //Modbus enable
369 ..... CaseNr:=CaseNr+1; //Program case
370 .....
371 ..... // CNL_setpoint_winter
372 ..... 22:
373 ..... MMdb.Node := 8;
374 ..... MMdb.Enable:=TRUE; //Modbus enable
375 ..... MMdb.FCode:=16#06; //Modbus function code
376 ..... MMdb.Address:=1008; //Modbus register address
377 ..... MMdb.Points:=1; //Modbus register points
378 ..... MMdb.Buffer:=ADR(CNL_setpoint_inverno); //Memory buffer address .....
379 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
380 ..... MMdb.Enable:=FALSE; //Modbus enable
381 ..... CaseNr:=CaseNr+1; //Program case
382 .....
383 ..... // CNL_setpoint_summer
384 ..... 23:
385 ..... MMdb.Node := 8;
386 ..... MMdb.Enable:=TRUE; //Modbus enable
387 ..... MMdb.FCode:=16#06; //Modbus function code
388 ..... MMdb.Address:=1011; //Modbus register address
389 ..... MMdb.Points:=1; //Modbus register points
390 ..... MMdb.Buffer:=ADR(CNL_setpoint_estate); //Memory buffer address .....
391 ..... IF NOT(MMdb.Done) THEN RETURN; END_IF;
392 ..... MMdb.Enable:=FALSE; //Modbus enable
393 ..... CaseNr:=0; //Program case
394 ..... FirstLoop := TRUE;
395 ..... END_CASE;
396

```