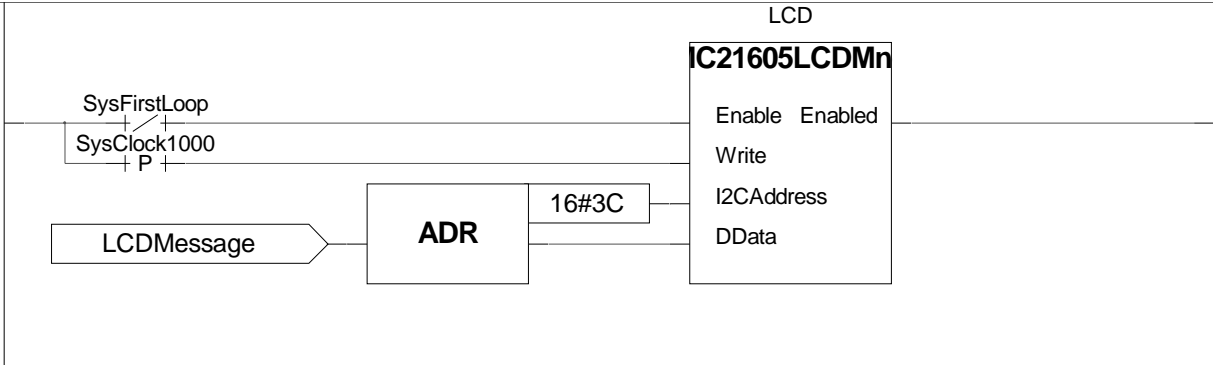


```

VAR
LCD : MC21605LCDMng;
LCDMessage : STRING[ 32 ] := 'Hello !           I am a SlimLine!';
END_VAR
    
```

1



Project : LCDTest	
PROGRAM : LD	
Release : LCDTest	Ver :1.00
Author : Sergio Bertana	Date:04/09/2013
Note :	Page:1 of 1

```

VAR_INPUT
Enable : BOOL; (* FB enable *)
Write : BOOL; (* Display write command *)
I2CAddress : USINT; (* LCD I2C address *)
DData : @USINT; (* Display data buffer pointer *)
END_VAR

VAR_OUTPUT
Enabled : BOOL; (* FB enabled *)
END_VAR

VAR
CaseNr : ARRAY[ 0..1 ] OF USINT; (* Case gestione *)
DWrBuffer : ARRAY[ 0..1 ] OF BYTE; (* Display write buffer *)
TimeBf : UDINT; (* Time buffer (uS) *)
DDIDx : USINT; (* Display data index *)
Ptr : @USINT; (* Auxiliary pointer *)
IWrite : BOOL; (* Display write command (Internal) *)
END_VAR
    
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "MC21605LCDMng" *)
3 (* ***** *)
4 (* Questo blocco funzione gestisce il display LCD MC21605 della Midas. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* GESTIONE ABILITAZIONE *)
9 (* ----- *)
10 (* Gestione abilitazione blocco funzione. *)
11
12 IF NOT(Enable) THEN Enabled:=FALSE; RETURN; END_IF;
13
14 (* Gestione primo loop abilitazione. *)
15
16 IF NOT(Enabled) THEN
17     Enabled:=TRUE; (* FB enabled *)
18     CaseNr[0]:=16#00; (* Case gestione *)
19     CaseNr[1]:=16#FF; (* Case gestione *)
20 END_IF;
21
22 (* ----- *)
23 (* GESTIONE COMNDO WRITE *)
24 (* ----- *)
25 (* Controllo se attivo il comando write. *)
26
27 IF (Write) THEN IWrite:=TRUE; END_IF;
28
29 (* ----- *)
30 (* TEMPORIZZAZIONE TRA CASES PROGRAMMA *)
31 (* ----- *)
32 (* Siccome non è possibile leggere dal display il segnale di busy, viene *)
33 (* eseguita una temporizzazione tra i vari cases pari al tempo necessario *)
34 (* al display per eseguire il comando più lento (Clear display 760 uS). *)
35
36 IF (CaseNr[0] <> CaseNr[1]) THEN
37     IF ((SysGetSysTime(TRUE)-TimeBf) < 1000) THEN RETURN; END_IF;
    
```

Project : LCDTest	
FUNCTION BLOCK : MC21605LCDMng	
Release : LCDTest	Ver : 1.00
Author : Sergio Bertana	Date: 04/09/2013
Note :	Page: 1 of 4

```

38     TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
39     CaseNr[1]:=CaseNr[0]; (* Case gestione *)
40 END_IF;
41
42 (* ----- *)
43 (* GESTIONE CASE PROGRAMMA *)
44 (* ----- *)
45 (* Eseguo gestione case LCD. *)
46
47 CASE (CaseNr[0]) OF
48
49     (* ----- *)
50     (* INIZIALIZZAZIONE DISPLAY *)
51     (* ----- *)
52     (* Eseguo comando "Default function set". *)
53
54     0, 1, 2:
55     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
56     DWrBuffer[1]:=16#30; (* Default function set *)
57     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
58     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
59
60     (* ----- *)
61     (* Eseguo comando "Function set DL, N, F". *)
62
63     3:
64     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
65     DWrBuffer[1]:=16#38; (* 8 bits, 2 lines, 5*7 dots *)
66     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
67     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
68
69     (* ----- *)
70     (* Eseguo comando "Display on/off control". *)
71
72     4:
73     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
74     DWrBuffer[1]:=16#08; (* Display off *)
75     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
76     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
77
78     (* ----- *)
79     (* Eseguo comando "Clear display". *)
80
81     5:
82     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
83     DWrBuffer[1]:=16#01; (* Clear display e cursor home *)
84     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
85     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
86
87     (* ----- *)
88     (* Eseguo comando "Cursor and display shift". *)
89
90     6:
91     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
92     DWrBuffer[1]:=16#06; (* Cursor increment *)
93     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
94     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
95
96     (* ----- *)
97     (* Eseguo comando "Display on/off control". *)

```

Project : LCDTest	
FUNCTION BLOCK : MC21605LCDMng	
Release : LCDTest	Ver :1.00
Author : Sergio Bertana	Date:04/09/2013
Note :	Page:2 of 4

```

98
99      7:
100     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
101     DWrBuffer[1]:=16#0C; (* Display on *)
102     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
103     CaseNr[0]:=100; (* Case gestione *)
104
105     (* ----- *)
106     (* SCRITTURA DATI SU RIGA SUPERIORE *)
107     (* ----- *)
108     (* Eseguo comando "Set DDRAM address". *)
109
110
111     100:
112     IF NOT(IWrite) THEN RETURN; END_IF;
113     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
114     DWrBuffer[1]:=16#80; (* DDRAM address:=0 *)
115     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
116     DDIDx:=0; (* Display data index *)
117     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
118
119     (* ----- *)
120     (* Eseguo scrittura dato sul display. Il tempo necessario è 18.5 uS *)
121     (* quindi non eseguo temporizzazione, il tempo di loop del programma *)
122     (* PLC sarà sicuramente maggiore. *)
123
124     101:
125     Ptr:=TO_UDINT(DData)+DDIDx; (* Auxiliary pointer *)
126     DWrBuffer[0]:=16#40; (* Co:=FALSE, A0:=TRUE *)
127     DWrBuffer[1]:=@Ptr; (* Dato display *)
128     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
129
130     (* Eseguo controllo se terminato tutta la riga superiore. *)
131
132     DDIDx:=DDIDx+1; (* Display data index *)
133     IF (DDIDx > 15) THEN CaseNr[0]:=110; END_IF;
134
135     (* ----- *)
136     (* SCRITTURA DATI SU RIGA INFERIORE *)
137     (* ----- *)
138     (* Eseguo comando "Set DDRAM address". *)
139
140
141     110:
142     DWrBuffer[0]:=16#80; (* Co:=TRUE, A0:=FALSE *)
143     DWrBuffer[1]:=16#C0; (* DDRAM address:=0 *)
144     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
145     DDIDx:=16; (* Display data index *)
146     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
147
148     (* ----- *)
149     (* Eseguo scrittura dato sul display. Il tempo necessario è 18.5 uS *)
150     (* quindi non eseguo temporizzazione, il tempo di loop del programma *)
151     (* PLC sarà sicuramente maggiore. *)
152
153     111:
154     Ptr:=TO_UDINT(DData)+DDIDx; (* Auxiliary pointer *)
155     DWrBuffer[0]:=16#40; (* Co:=FALSE, A0:=TRUE *)
156     DWrBuffer[1]:=@Ptr; (* Dato display *)
157     IF NOT(SysI2CWrRd(I2CAddress,02, ADR(DWrBuffer), 0, 0)) THEN CaseNr[0]:=0; RETURN; END_IF;
158
159     (* Eseguo controllo se terminato tutta la riga inferiore. *)

```

Project : LCDTest	
FUNCTION BLOCK : MC21605LCDMng	
Release : LCDTest	Ver :1.00
Author : Sergio Bertana	Date:04/09/2013
Note :	Page:3 of 4

FUNCTION_BLOCK MC21605LCDMng

```

158
159     DDIDx:=DDIDx+1; (* Display data index *)
160     IF (DDIDx > 31) THEN IWrite:=FALSE; CaseNr[0]:=100; END_IF;
161     ELSE
162     CaseNr[0]:=0; (* Case gestione *)
163     END_CASE;
164
165
166 (* [End of file] *)
167
168

```

	Project : LCDTest	
	FUNCTION BLOCK : MC21605LCDMng	
	Release : LCDTest	Ver :1.00
	Author : Sergio Bertana	Date:04/09/2013
	Note :	Page:4 of 4