

```
1 FUNCTION_BLOCK FB1200
2
3
4 TITLE = '提升门屏蔽光栅子程序'
5 // INR0-INR7 右侧P1-P5提升门
6 // INL0-INL7 左侧卷帘门
7 // S1, S2 是静态门 不检测是否打开. 只要是关闭状态即可.
8 // 其余门必须完成东走 [关闭-打开-关闭] 之后才有输出.
9 // OUTR 右侧状态输出.
10 // OUTL 左侧状态输出.
11 // OUTR 在 INR 完成动作后 即输出 OUTR=TRUE.
12 // OUTL 必须要S1, S2, OUTR都为真且INL完成动作后才输出 OUTL=TRUE.
13
14 // KEY为假时: 复位所有信号.
15 VERSION: '1.0'
16 AUTHOR: xy_z
17 NAME: SZHONG6
18 FAMILY: VOLVO
19
20
21 VAR
22     INR : STRUCT           //右侧
23         IN00 : BOOL;
24         IN01 : BOOL;
25         IN02 : BOOL;
26         IN03 : BOOL;
27         IN04 : BOOL;
28         IN05 : BOOL;
29         IN06 : BOOL;
30         IN07 : BOOL;
31     END_STRUCT;
32
33     INL : STRUCT           //左侧
34         IN00 : BOOL;
35         IN01 : BOOL;
36         IN02 : BOOL;
37         IN03 : BOOL;
38         IN04 : BOOL;
39         IN05 : BOOL;
40         IN06 : BOOL;
41         IN07 : BOOL;
42     END_STRUCT;
43     R AT INR : BYTE;
44     L AT INL : BYTE;
45
46     RC : BYTE;
47     LC : BYTE;
48
49     StepR : INT;
50     StepL : INT;
51 END_VAR
52
53 VAR_INPUT
54     INR00 : BOOL;
55     INR01 : BOOL;
56     INR02 : BOOL;
57     INR03 : BOOL;
58     INR04 : BOOL;
59     INR05 : BOOL;
60     INR06 : BOOL;
61     INR07 : BOOL;
62
63     INL00 : BOOL;
64     INL01 : BOOL;
65     INL02 : BOOL;
66     INL03 : BOOL;
67     INL04 : BOOL;
68     INL05 : BOOL;
69     INL06 : BOOL;
70     INL07 : BOOL;
71
72     S1 : BOOL;
73     S2 : BOOL;
```

```
74
75     Key : BOOL;
76 END_VAR
77
78 VAR_OUTPUT
79     OUTR : BOOL;
80     OUTL : BOOL;
81 END_VAR
82
83 BEGIN
84     INR.IN00 := INR00;
85     INR.IN01 := INR01;
86     INR.IN02 := INR02;
87     INR.IN03 := INR03;
88     INR.IN04 := INR04;
89     INR.IN05 := INR05;
90     INR.IN06 := INR06;
91     INR.IN07 := INR07;
92
93     INL.IN00 := INL00;
94     INL.IN01 := INL01;
95     INL.IN02 := INL02;
96     INL.IN03 := INL03;
97     INL.IN04 := INL04;
98     INL.IN05 := INL05;
99     INL.IN06 := INL06;
100    INL.IN07 := INL07;
101
102
103
104
105
106
107 IF KEY THEN
108     // 右侧安全门,全部 关闭-打开-关闭返回true
109     CASE StepR OF
110         0 : //判断全部信号true
111             IF R = 16#FF THEN
112                 StepR :=1;
113             END_IF;
114         1 : //判断全部信号false一次
115             R := R XOR 16#FF; //取反
116             RC := R OR RC; //存储到LR
117             IF RC = 16#FF THEN //全为0时下一步
118                 StepR := 2;
119             END_IF;
120         2 : //再次判断全部信号true
121             IF R = 16#FF THEN
122                 StepR :=3;
123             END_IF;
124         3 :
125             OUTR := TRUE;
126     END_CASE;
127
128     // 左侧安全门,全部 关闭-打开-关闭 且 右侧完成动作 且 S1,S2全部关闭
129     CASE StepL OF
130         0 : //判断全部信号true
131             IF L = 16#FF THEN
132                 StepL :=1;
133             END_IF;
134         1 : //判断全部信号false一次
135             L := L XOR 16#FF; //取反
136             LC := L OR LC; //存储到LC
137             IF LC = 16#FF THEN //全为0时下一步
138                 StepL := 2;
139             END_IF;
140         2 : //再次判断全部信号true
141             IF L = 16#FF AND OUTR AND S1 AND S2 THEN
142                 StepL :=3;
143             END_IF;
144         3 :
145             OUTL := TRUE;
146     END_CASE;
```

```
147     ELSE
148         // RESET
149         STEPR := 0;
150         STEPL := 0;
151         OUTR  := 0;
152         OUTL  := 0;
153         RC   := 16#0;
154         LC   := 16#0;
155     END_IF;
156 END_FUNCTION_BLOCK
157
158
```