

1
2

```

3 ORGANIZATION_BLOCK    Modulo_Principal // OB1  // OBxxx
4
5 VAR_TEMP
6 // reservado
7 info : ARRAY [0..19] OF BYTE;
8 // Variables temporales
9 TIM_P_Carga :INT ; // Tiempo de Rearme Seguridades Puente de Carga
10 TIM_P_Descarga :INT ; // Tiempo de Rearme Seguridades Puente de Descarga
11 END_VAR
12
13 //
14 // Generador de Pulsos
15 //
16
17 Generador(TIM_PULSO_1SEG := TIM_1SEG // IN: TIMER
18 ,TIM_PULSO_01SEG := TIM_01SEG // IN: TIMER
19 ,Pulso_1Seg := Pulso_1Seg // OUT: BOOL
20 ,Pulso_01Seg := Pulso_01Seg // OUT: BOOL
21 ); // VOID
22
23 //
24 // Rearme Potencia Cuadro Puente de Carga
25 //
26 IF (PC_KSP_MARCHA_GENERAL AND NOT PC_SEGURIDADES_REARMADAS) OR PC_K23_MARCHA_POTENCIA THEN
27 PC_K23_MARCHA_POTENCIA := TRUE ; // Marcha Potencia >> Serie del Poten
28 END_IF ;
29
30 // Ciclo de rearme del poten
31 // Si PC_K23_MARCHA_POTENCIA es uno pone a cero TIM_P_CARGA
32 IF PC_K23_MARCHA_POTENCIA THEN
33 IF Pulso_01Seg AND TIM_P_Carga < 25 THEN
34 TIM_P_Carga := TIM_P_Carga + 1;
35 END_IF ;
36 PC_K24_REARME_POTENCIA := (TIM_P_Carga >10) AND (TIM_P_Carga < 20) ; // Pulso de rearme P
37 oten
38 ELSE
39 TIM_P_Carga := 0;
40 PC_K24_REARME_POTENCIA := FALSE ; // Pulso de rearme Poten
41 END_IF ;
42
43 // Desconexion del los elementos de potencia por parada de emergencia
44 IF NOT PC_KSP_MARCHA_GENERAL AND NOT PC_SEGURIDADES_REARMADAS AND PC_K23_MARCHA_POTENCIA THEN
45 PC_K23_MARCHA_POTENCIA := FALSE ; // Marcha Potencia >> Serie del Poten
46 PC_K24_REARME_POTENCIA := FALSE ; // Pulso de rearme Poten
47 TIM_P_Carga := 0 ;
48 END_IF ;
49 //
50 // Bloque de Funcion Puente de Carga
51 //
52 Prg_Puente_Carga(); // VOID
53
54 //
55 // Rearme Potencia Cuadro Puente de Descarga
56 //
57 IF (PD_KSP_MARCHA_GENERAL AND NOT PD_SEGURIDADES_REARMADAS) OR PD_K23_MARCHA_POTENCIA THEN
58 PD_K23_MARCHA_POTENCIA := TRUE ; // Marcha Potencia >> Serie del Poten
59 END_IF ;
60 // Ciclo de rearme del poten
61 IF PD_K23_MARCHA_POTENCIA THEN
62 IF Pulso_01Seg AND TIM_P_Descarga < 25 THEN
63 TIM_P_Descarga := TIM_P_Descarga + 1;
64 END_IF ;
65 PD_K24_REARME_POTENCIA := (TIM_P_Descarga >10) AND (TIM_P_Descarga < 20) ; // Pulso de re
66 arme Poten
67 ELSE
68 TIM_P_Descarga := 0;
69 PD_K24_REARME_POTENCIA := FALSE ; // Pulso de rearme Poten
70 END_IF ;
71
72 // Desconexion del los elementos de potencia por parada de emergencia
73 IF NOT PD_KSP_MARCHA_GENERAL AND NOT PD_SEGURIDADES_REARMADAS AND PD_K23_MARCHA_POTENCIA THEN
74 PD_K23_MARCHA_POTENCIA := FALSE ; // Marcha Potencia >> Serie del Poten
75 PD_K24_REARME_POTENCIA := FALSE ; // Pulso de rearme Poten
76 TIM_P_Descarga := 0 ;
77 END_IF ;
78 //
79 // Bloque de Funcion Puente de Descarga
80 //
81 Prg_Puente_Descarga(); // VOID
82
83 //
84 // Zona libre de trabajo entre los dos puentes, debe de haber un minimo de 3 puestos entre ambos
85 // para que el puente de carga pueda realizar su trabajo ( Carga de horno )
86 // Tiene siempre preferencia la descarga
87 //
88 IF DB_PuenteCarga.Posicion_Carga <= 26 THEN
89 // Casos de P.Carga entre el 9 y el 26, con 2 Huecos de diferencial

```

```

90      IF ((DB_PuenteDescarga.Posicion_Descarga - DB_PuenteCarga.Posicion_Carga) <= 2) AND NOT PD
_Habilitar_A1_A2_A3_A4 THEN
91          PC_Diferencial_Puestos := TRUE ;
92      ELSE
93          PC_Diferencial_Puestos := FALSE;
94      END_IF ;
95      ELSE // Para el resto de los casos ( >26 ), solo 3 huecos
96      IF ((DB_PuenteDescarga.Posicion_Descarga - DB_PuenteCarga.Posicion_Carga) <= 3) AND NOT P
D_Habilitar_A1_A2_A3_A4 THEN
97          PC_Diferencial_Puestos := TRUE ;
98      ELSE
99          PC_Diferencial_Puestos := FALSE;
100      END_IF ;
101      END_IF ;
102
103 //
104 // Inhibir Giro del Horno desde los Puentes de Carga/Descarga
105 //
106 Inhibir_Giro_Horno(Clock      := Pulso_01Seg           // IN: BOOL
107                    ,Horno_Girando := PC_R2_HORNO_GIRANDO // IN: BOOL
108                    ,PC_Modo        := PC_Modo_Automatico  // IN: BOOL
109                    ,PC_FC_Arriba   := PC_FC1_MAXIMO_ELEVACION // IN: BOOL
110                    ,PC_Pos_Referen := PC_VF_A1_Ok_Referenc AND PC_VF_A2_Ok_Referenc AND
111                                     PC_VF_A3_Ok_Referenc AND PC_VF_A4_Ok_Referenc //
IN: BOOL
112                    ,PC_Pos_Origen := PC_VF_A2_Pos_Origen AND PC_VF_A2_Pos_Origen AND
113                                     PC_VF_A3_Pos_Origen AND PC_VF_A4_Pos_Origen //
IN: BOOL
114                    ,PC_Pos_Destino := PC_VF_A1_Pos_Destino OR PC_VF_A2_Pos_Destino OR
115                                     PC_VF_A3_Pos_Destino OR PC_VF_A4_Pos_Destino //
IN: BOOL
116                    ,PD_Modo        := PC_Modo_Automatico OR PD_Modo_Automatico // IN: BOOL
117                    ,PD_FC_Arriba   := PD_FC1_MAXIMO_ELEVACION // IN: BOOL
118                    ,PD_Pos_Referen := PD_VF_A1_Ok_Referenc AND PD_VF_A2_Ok_Referenc AND
119                                     PD_VF_A3_Ok_Referenc AND PD_VF_A4_Ok_Referenc //
IN: BOOL
120                    ,PD_Pos_Origen := PD_VF_A2_Pos_Origen AND PD_VF_A2_Pos_Origen AND
121                                     PD_VF_A3_Pos_Origen AND PD_VF_A4_Pos_Origen //
IN: BOOL
122                    ,PD_Pos_Destino := PD_VF_A1_Pos_Destino OR PD_VF_A2_Pos_Destino OR
123                                     PD_VF_A3_Pos_Destino OR PD_VF_A4_Pos_Destino //
IN: BOOL
124                    ,Inhibir_GHorno := PC_K15_CONSENSO_HORNO // INOUT: BOOL
125                    ,PC_Inh_GHorno := PC_Mem_Inhibicion_GH // INOUT: BOOL
126                    ,PD_Inh_GHorno := PD_Mem_Inhibicion_GH // INOUT: BOOL
127                    ); // VOID
128
129 //
130 // Enlaces de A1, A2, A3, A4 Ajuste Manual de Ejes Velocidad muy Lenta
131 //
132 PC_AJUSTE_MAN_A1 := PC_Ajuste_Ejes_Man ;
133 PC_AJUSTE_MAN_A2 := PC_Ajuste_Ejes_Man ;
134 PC_AJUSTE_MAN_A3 := PC_Ajuste_Ejes_Man ;
135 PC_AJUSTE_MAN_A4 := PC_Ajuste_Ejes_Man ;
136
137 //
138 // Enlaces de A1, A2, A3, A4 Ajuste Manual de Ejes Velocidad muy Lenta
139 //
140 PD_AJUSTE_MAN_A1 := PD_Ajuste_Ejes_Man ;
141 PD_AJUSTE_MAN_A2 := PD_Ajuste_Ejes_Man ;
142 PD_AJUSTE_MAN_A3 := PD_Ajuste_Ejes_Man ;
143 PD_AJUSTE_MAN_A4 := PD_Ajuste_Ejes_Man ;
144
145 //
146 // Datos de la Tablas Puente, Carro y Giro
147 //
148 //
149 Gestion_Datos_Tablas(); // VOID
150
151 //
152 // Gestion Datos Pantalla PLC <> Pantalla
153 //
154 Gestion_Pantalla(); // VOID
155
156 //
157 // Tiempo de Ciclo Translacion Puente de Carga
158 //
159
160 Tiempos_Ciclo(Habilitacion := PC_Modo_Automatico // IN: BOOL
161               ,Clock        := Pulso_01Seg // IN: BOOL
162               ,Ciclo_Carga  := DB_PuenteCarga.Ciclo_Auto_Elevac > 0 AND
163                               DB_PuenteCarga.Ciclo_Auto_Elevac <= 100 // IN: BOOL
164               ,Ciclo_Desc   := (DB_PuenteCarga.Ciclo_Auto_Elevac > 100 AND
165                               DB_PuenteCarga.Ciclo_Auto_Elevac <150) OR
166                               (DB_PuenteCarga.Ciclo_Auto_Elevac = 0 AND NOT (PC_VF_A3_Pos_
Origen AND PC_VF_A4_Pos_Origen
167                               AND PC_VF_A1_Pos_Origen AND PC_VF_A2_Pos_Origen )) // IN:
BOOL
168               ,Mem_Ciclo_Carga := PC_Mem_TCiclo_Dest // INOUT: BOOL

```

```
169      ,Mem_Ciclo_Desc      := PC_Mem_TCiclo_Origen // INOUT: BOOL
170      ,PV_Ciclo_Carga      := DB_PuenteCarga.TCiclo_D_Puente[0] // INOUT: INT
171      ,PV_Ciclo_Desc      := DB_PuenteCarga.TCiclo_O_Puente[0] // INOUT: INT
172      ,TIM_Ciclo_Carga     := DB_PuenteCarga.TCiclo_D_Puente[1] // INOUT: INT
173      ,TIM_Ciclo_Desc     := DB_PuenteCarga.TCiclo_O_Puente[1] // INOUT: INT// INOUT:
INT
174      ); // VOID
175
176 //
177 // Tiempo de Ciclo Translacion Puente de Descarga
178 //
179
180      Tiempos_Ciclo(Habilitacion := PD_Modo_Automatico // IN: BOOL
181      ,Clock := Pulso_01Seg // IN: BOOL
182      ,Ciclo_Carga := DB_PuenteDescarga.Ciclo_Auto_Elevac > 0 AND
183      DB_PuenteDescarga.Ciclo_Auto_Elevac <= 100 // IN: BOOL
184      ,Ciclo_Desc := (DB_PuenteDescarga.Ciclo_Auto_Elevac > 100 AND
185      DB_PuenteDescarga.Ciclo_Auto_Elevac <150) OR
186      (DB_PuenteDescarga.Ciclo_Auto_Elevac = 0 AND NOT (PD_VF_A3_P
os_Origen AND PD_VF_A4_Pos_Origen
187      AND PD_VF_A1_Pos_Origen AND PD_VF_A2_Pos_Origen )) // IN:
BOOL
188      ,Mem_Ciclo_Carga := PD_Mem_TCiclo_Dest // INOUT: BOOL
189      ,Mem_Ciclo_Desc := PD_Mem_TCiclo_Origen // INOUT: BOOL
190      ,PV_Ciclo_Carga := DB_PuenteDescarga.TCiclo_D_Puente[0] // INOUT: INT
191      ,PV_Ciclo_Desc := DB_PuenteDescarga.TCiclo_O_Puente[0] // INOUT: INT
192      ,TIM_Ciclo_Carga := DB_PuenteDescarga.TCiclo_D_Puente[1] // INOUT: INT
193      ,TIM_Ciclo_Desc := DB_PuenteDescarga.TCiclo_O_Puente[1] // INOUT: INT// INOU
T: INT
194      ); // VOID
195
196 END_ORGANIZATION_BLOCK
197
198
```