```
1
2
3
4
5 //
6 // --------------    GENERADOR DE PULSOS      -----------------
7 //
```

```
 8 FUNCTION  Generador : VOID //FC1 : VOID
 9 // ******************************
10 // ***   GENERADOR DE PULSOS   ***
11 // ******************************
12 VAR_INPUT
13   TIM_PULSO_1SEG   : TIMER;  // Temporizador pulso 1.0 Seg.
14   TIM_PULSO_01SEG  : TIMER;  // Temporizador pulso 0.1 Seg.
15 END_VAR
16
17 VAR_OUTPUT
18   Pulso_1Seg   : BOOL;  // Salida pulso 1.0 Seg.
19   Pulso_01Seg  : BOOL;  // Salida pulso 0.1 Seg.
20 END_VAR
21
22 VAR
23   CTime   : S5TIME;
24 END_VAR
25
26
27 //Temporizador Pulso 1 Seg
28    CTime:=S_ODT(T_NO := TIM_PULSO_1SEG,
29                 S    := NOT Pulso_1Seg ,
30                 TV   := T#1.0S,
31                 Q    := Pulso_1Seg);
32 //Temporizador Pulso 0.1 Seg
33    CTime:=S_ODT(T_NO := TIM_PULSO_01SEG,
34                 S    := NOT Pulso_01Seg ,
35                 TV   := T#0.1S,
36                 Q    := Pulso_01Seg);
37
38 END_FUNCTION
39
40 //
41 // --------------    GENERADOR DE PULSOS      -----------------
42 //
43
```

```
44 FUNCTION  Flip_Flop : VOID   //FC2 : INT
45 // ******************************
46 // ***   FUNCION FLIP-FLOP     ***
47 // ******************************
48 VAR_INPUT
49   TIMER_OFF : TIMER  ;  // Temporizador Retardo OFF
50   TIMER_ON  : TIMER  ;  // Temporizador Retardo ON
51   T_OFF     : S5TIME ;  // Tiempo OFF
52   T_ON      : S5TIME ;  // Tiempo ON
53 END_VAR
54
55 VAR_IN_OUT
56   Out_FF    : BOOL   ;  // Salida Flip-Flop
57 END_VAR
58
59 VAR
60   TM_ON   :  BOOL   ;
61   CTime   :  S5TIME ;
62 END_VAR
63
64  // Área de instrucciones
65
66  CTime := S_ODT(T_NO := TIMER_OFF,
67                 S    := NOT Out_FF ,
68                 TV   := T_OFF,
69                 Q    := TM_ON);
70
71  CTime := S_PEXT(T_NO := TIMER_ON,
72                  S    := TM_ON,
73                  TV   := T_ON,
74                  Q    := Out_FF);
75
76
77 END_FUNCTION
78
79
80 //
81 // --------------    FRENO TRANSLACION      -----------------
82 //
83
```

```
 84 FUNCTION  Freno_Translacion : VOID   //FC3 : INT
 85 // ***************************************
 86 // ***   FUNCION FRENO TRANSLACION     ***
 87 // ***************************************
 88 VAR_INPUT
 89   Modo_Freno    : BOOL   ;  // Modo de trabajo OFF Man, ON Auto
 90   Marcha_Paro   : BOOL   ;  // Orden Marcha/Parada Freno
 91   TIMER_ON_OFF  : TIMER  ;  // Temporizador Retardo ON
 92   T_ON          : S5TIME ;  // Tiempo Retardo OFF Freno
 93 END_VAR
 94
 95 VAR_IN_OUT
 96   Out_FRENO : BOOL   ;  // Salida FRENO
 97 END_VAR
 98
 99 VAR
100   TM_ON   :  BOOL   ;
101   CTime   :  S5TIME ;
102 END_VAR
103
104  // Área de instrucciones
105
106  // Activacion y desactivacion del Freno en Modo Automatico
107  // Out = ON a la activacion y tras la señal de OFF, permanece activo el tiempo T_ON
108  CTime := S_OFFDT(T_NO  := TIMER_ON_OFF,
109                   S     := (Marcha_Paro AND Modo_Freno) OR NOT  Modo_Freno,
110                   R     := FALSE,
111                   TV    := T_ON,
112                   Q     := Out_FRENO);
113
114
115 END_FUNCTION
116
117 //
118 // --------------    Numero de Hilos segun Programa     -----------------
119 //
120
121
```

```
122 FUNCTION  Programas_Descargas: VOID      // FCx04 : INT
123
124 VAR_INPUT
125   // Variables de Entrada
126   Carga_Descarga        : BOOL  ;  // OFF [ Puente Carga ] ... ON [ Puente Descarga ]
127   Puesto_Min_Horno      : INT   ;  // Puesto Minimo del Horno, para la Carga /Descarga
128   Puesto_Max_Horno      : INT   ;  // Puesto Maximo del Horno, para la Carga /Descarga
129   Puente_Arriba         : BOOL  ;  // Puente Arriba
130   Horno_Girando         : BOOL  ;  // Señal de Horno Hirando
131   Limite_Prg            : INT   ;  // Limite del Numero de Programas Maximos a introducir
132
133 END_VAR
134
135 VAR_IN_OUT
136   // Variables Entradas/Salidas
137
138   NPuesto_Des_Horno     : INT   ;  // PV Puesto actual de Carga/Descarga del Horno
139   Hilo_Curso            : INT   ;  // Numero de Hilo en Curso
140
141   Tipo_Programa         : INT   ;  // Seleccion del Numero de Programa para el Puente de Carga/Des
    carga
142   Num_Hilos_Prg         : INT   ;  // Numero de Descargas/ Cargas, por puesto del horno, para el p
    rograma seleccionado
143
144   Fin_Ciclo_Car_Des     : BOOL  ;  // Estado de Final de ciclo de Carga/Descarga del Hilo en el Ho
    rno, Descuenta 1 Hilo
145   Memoria_Fin_Ciclo     : BOOL  ;  // Memoria de Final Ciclo Carga/Descarga en el Horno
146   Memoria_Giro_Horno    : BOOL  ;  // Memoria girando el Horno
147   Inhibir_Puente        : BOOL  ;  // Memoria inhibir Puente por fin de Carga/Descarga
148
149 END_VAR
150
151   //
152   //   Control puestos limite de Horno, para el caso de cada puente
153   //
154       //  Fin_Ciclo_Car_Des = Ciclo_Elevacion = 120 _Orden Abrir ... [Tras descargar, da orden d
    e Abrir]
155       IF Fin_Ciclo_Car_Des AND NOT Memoria_Fin_Ciclo AND NOT Puente_Arriba THEN
156          Hilo_Curso        := Hilo_Curso - 1 ;
157          Memoria_Fin_Ciclo := TRUE ;           // Memoria Fin ciclo de Descarga de un Hilo en el
    Horno
158          // Control de Hilo_Curso, tras la descarga
159          IF Hilo_Curso <= 0 THEN
160             NPuesto_Des_Horno  := NPuesto_Des_Horno + 1 ;    // Si al descargar el hilo se ha cmp
    letado el paquete se pasa a otro puesto
161             Hilo_Curso         := Num_Hilos_Prg ;            // Inicializa el Numero de Hilos que
     tiene que llevar el nuevo paquete
162          END_IF ;
163       END_IF ;
164
165       // Una vez el Puente Arriba, y con memoria de descarga, Reset la memoria de descarga
166       IF Puente_Arriba AND Memoria_Fin_Ciclo THEN
167          Memoria_Fin_Ciclo := FALSE ;  // para descontar solo un hilo por ciclo
168       END_IF ;
169
170       // Limite minimo del Hilo.
171       IF Hilo_Curso <= 0 THEN
172          Hilo_Curso := Num_Hilos_Prg ;            // Inicializa el Numero de Hilos
173       END_IF ;
174
175       // Control limites de la posicion de Carga para el giro del Horno
176       IF Horno_Girando AND NOT Memoria_Giro_Horno  THEN
177          Memoria_Giro_Horno := TRUE ;
178          IF NOT Inhibir_Puente THEN
179             NPuesto_Des_Horno  := NPuesto_Des_Horno - 1 ;  // al girar el horno el puente debe se
    guir completabdo el paquete en el puesto anterior
180          END_IF ;
181          // Anula la Inhibicion, tras Giro del Horno
182          // IF Inhibir_Puente AND NPuesto_Des_Horno = Puesto_Max_Horno THEN
183             Inhibir_Puente    := FALSE ;
184          // END_IF;
185       ELSIF  NOT Horno_Girando AND Memoria_Giro_Horno THEN
186             Memoria_Giro_Horno := FALSE;
187       END_IF ;
188
189      // Control limites de la posicion de Descarga del Horno
190         IF NPuesto_Des_Horno < Puesto_Min_Horno THEN      // El puente esta en el puesto Minimo y
     debe de retroceder y caragar el minimo que puede
191            NPuesto_Des_Horno := Puesto_Min_Horno ;
192         ELSIF  NPuesto_Des_Horno > Puesto_Max_Horno THEN  // El puente esta en el puesto Maximo y
     no debe de Avanzar, debe esperar a que gire el horno
193             NPuesto_Des_Horno := Puesto_Max_Horno ;
194             Inhibir_Puente    := TRUE             ;
195         END_IF ;
196
197   //
198   //   Control limites de seleccion de programas
199   //
200        IF Tipo_Programa > Limite_Prg THEN
```

```
201           Tipo_Programa := Limite_Prg;       // Programa = Limite, siempre que la seleccion sea > a limite
202         ELSIF  Tipo_Programa <= 0 THEN
203               Tipo_Programa := 5;             // Si no hay seleccion, Programa = 5
204         END_IF ;
205
206   //
207   //   Obtener Hilos en funcion al programa seleccionado
208   //
209         CASE  Tipo_Programa + 1 OF
210               1 :   // Programa 1 Rasillon 4 Hilos
211                     Num_Hilos_Prg := 4 ;
212               2 :   // Programa 2 Cerramiento 4 Hilos
213                     Num_Hilos_Prg := 4 ;
214               3 :   // Programa 3 Ladrillo    5 Hilos
215                     Num_Hilos_Prg := 5 ;
216               4 :   // Programa 4 Ladrillo    4 Hilos
217                     Num_Hilos_Prg := 4 ;
218               5 :   // Programa 5 Termoarcilla 3 Hilos
219                     Num_Hilos_Prg := 3 ;
220               6 :   // Programa 6 Termoarcilla 4 Hilos
221                     Num_Hilos_Prg := 4 ;
222               7 :   // Programa 7 Vobedilla    3 Hilos
223                     Num_Hilos_Prg := 3 ;
224               8 :   // Programa 8 Vobedilla    4 Hilos
225                     Num_Hilos_Prg := 4 ;
226               9 :   // Programa 9
227                     Num_Hilos_Prg := 3 ;
228               10:   // Programa 10
229                     Num_Hilos_Prg := 3 ;
230               11:   // Programa 11
231                     Num_Hilos_Prg := 3 ;
232               12:   // Programa 12
233                     Num_Hilos_Prg := 3 ;
234               13:   // Programa 13
235                     Num_Hilos_Prg := 3 ;
236               14:   // Programa 14
237                     Num_Hilos_Prg := 3 ;
238               15:   // Programa 15
239                     Num_Hilos_Prg := 3 ;
240               16:   // Programa 16
241                     Num_Hilos_Prg := 3 ;
242               17:   // Programa 17
243                     Num_Hilos_Prg := 3 ;
244               18:   // Programa 18
245                     Num_Hilos_Prg := 3 ;
246               19:   // Programa 19
247                     Num_Hilos_Prg := 3 ;
248               20:   // Programa 20
249                     Num_Hilos_Prg := 3 ;
250         END_CASE ;
251   ;
252 END_FUNCTION
253
254
255 //
256 // --------------  Comuniciacion PLC <> UNI en Modo Posicionado   -----------------
257 //
258
259
```

```
260 FUNCTION  Enlace_PLC_UNI: VOID       // FC008 : INT
261
262 VAR_INPUT
263   // Variables de Entrada
264    Habilitacion     :BOOL ;      // Habilitacion del Sistema
265    Inhibir          :BOOL ;      // Inhibir Ciclo Posicionado Destino OFF >> Inhibir, ON >> OK
266    Modo             :BOOL ;      // Modo de Trabajo  OFF >> Manual, ON >> Automatico
267    Hab_Destino      :BOOL ;      // Permiso de Destino... P Descarga Si A1-A2 > 3000.0 mm, Salida
   en Recto
268    Carga_Descarga   :BOOL ;      // Tipo de Puente   OFF >> Carga,  ON >> Descarga
269    FC_Lento_Origen  :BOOL ;      // FC. Origen Lento
270    FC_Max_Origen    :BOOL ;      // FC. Maximo Origen
271    FC_Max_Destino   :BOOL ;      // FC. Maximo Destino
272    FC_Elevacion     :BOOL ;      // FC. Maximo Elevacion
273    DT_Atras_Pinzas  :BOOL ;      // DT. Pinza Atras
274    Tipo_Maniobra    :INT  ;      // Tipo de Manionbra a realizar  (0 > Nada, 10 > Origen, 20 > Des
   tino )
275    Ciclo_GHorno     :BOOL ;      // Orden de Ciclo Giro del Gorno
276    Sis_Posicionado  :BOOL ;      // Sistema total Posicionado .... Puente, Carro, Giro ( Evita con
   tinuar )
277   // Ordenes PLC > UNI
278    Inicializar_Pos  :BOOL ;      // Orden de Inicializar el Control de Posicionado
279
280   // Enlaces UNI >> PLC
281    Referenciado     :BOOL ;      // Referenciado% Estado Referenciado del equipo a Origen
282    POrden_Destino   :BOOL ;      // Cal_Destino%  Estado Peticion Orden Destino Confirmada por el
   UNI
283    OK_Origen        :BOOL ;      // Pos_Origen%   Estado Equipo situado en Origen
284    OK_Destino       :BOOL ;      // Pos_Destino%  Estado Equipo situado en Destino
285    POrden_Origen    :BOOL ;      // Cal_Origen%   Estado Peticion Orden Origen Confirmada por el U
   NI
286    POrden_Giro_H    :BOOL ;      // Cal_GiroH%    Estado Peticion Orden Giro Horno Confirmada por
   el UNI
287
288 END_VAR
289
290 VAR_IN_OUT
291   // Variables de Entrada/Salida
292    Ciclo_Posicion    :INT  ;      // Ciclo de Posicionado Automatico
293
294   // Ordenes PLC > UNI
295    Rapido_Lento     :BOOL ;      // LR_Auto% Orden Rapido/Lento (con cero se desplaza a velocidad
   lenta)  Automatico
296    Avance           :BOOL ;      // Av_Auto% Orden de Avance                        Automatico
297    Retroceso        :BOOL ;      // Rt_Auto% Orden de Retroceso                     Automatico
298    Pet_Origen       :BOOL ;      // PO_Auto% Orden de Peticion Posicion a Origen    Automatico
299    Pet_Inicializar  :BOOL ;      // IN_Auto% Orden de Inicializar Ciclo          No Automatico, Ma
   nual
300    Pet_Destino      :BOOL ;      // PD_Auto% Orden de Peticion Posicion a Destino   Automatico
301    Pet_Ref_Inicio   :BOOL ;      // RF_Auto% Orden de Peticion Referencia a Origen  Automatico
302    Modo_Auto        :BOOL ;      // MD_Auto% Orden Modo del UNI en                  Automatico
303    Pet_Giro_H       :BOOL ;      // GH_Auto% Orden de Peticion Giro Horno, reajuste Automatico de
   los parametros de la posicion de destino
304    Modo_Bruto       :BOOL ;      // MB_Auto% Orden Modo Posicionado en Bruto        Automatico
305    Ciclo_Origen     :INT  ;      // Ciclo Origen
306    Ciclo_Destino    :INT  ;      // Ciclo Destino
307    Ciclo_GiroHorno  :INT  ;      // Ciclo Giro Horno
308
309 END_VAR
310
311 //
312 // Condiciones Para el Ciclo Automatico
313 //
314 IF NOT Carga_Descarga THEN     // Puente de CARGA
315    //
316    // Condiciones generales de Inicio Ciclo de Origen
317    // Condiciones en las que el puente debe regresar a su posicion de origen
318  //IF FC_Elevacion AND DT_Atras_Pinzas(pinzas abiertas) AND Modo_Auto THEN
319    IF FC_Elevacion AND DT_Atras_Pinzas THEN
320       // Inicio Ciclo de Origen
321       IF Tipo_Maniobra = 10 AND Ciclo_Origen = 0 AND NOT OK_Origen THEN
322          Ciclo_Origen := 10 ;   // Orden de Inicio de Ciclo a Origen
323       // Una vez atendida la peticion, anula la orden de Inicio Ciclo
324       ELSIF  Ciclo_Origen = 10 AND (Pet_Ref_Inicio OR Pet_Origen OR Pet_Destino) THEN
325              Ciclo_Origen := 20 ;
326       // Una vez Posicionado a Origen y dejar de estar arriba(empieza a bajar a recoger material),
    anula la Memoria de orden origen
327       ELSIF  Ciclo_Origen >= 10 AND OK_Origen AND Tipo_Maniobra <> 10 THEN
328             Ciclo_Origen := 0;  // Finaliza la Secuencia de Peticion Origen
329       END_IF ;
330    // Si la Pinza no esta Abierta o no esta Arriba, Anula las Ordenes de Origen, porque puede esta
   r cargada de material
331    ELSE
332        Ciclo_Origen := 0;    // Inicializa la Secuencia de Peticion Origen
333    END_IF ;
334    //
335    // Condiciones generales de Inicio Ciclo de Destino
336    //
337  //IF FC_Elevacion AND NOT DT_Atras_Pinzas (pinzas cerradas) AND Eje referenciado THEN
```

```
338    IF FC_Elevacion AND NOT DT_Atras_Pinzas AND Inhibir THEN
339       // Inicio Ciclo de Destino
340     //IF Tipo_Maniobra = 20 AND Ciclo_Destino = 0 AND NOT OK_Destino AND Habilitacion destino THEN
341       IF Tipo_Maniobra = 20 AND Ciclo_Destino = 0 AND NOT OK_Destino AND Hab_Destino THEN
342          Ciclo_Destino := 10 ;       // Orden de Inicio de Ciclo a Destino
343       // Una vez atendida la peticion, anula la orden de Inicio Ciclo
344       ELSIF  Ciclo_Destino = 10  AND (Pet_Ref_Inicio OR Pet_Origen OR Pet_Destino) THEN
345             Ciclo_Destino := 20 ;    // La Peticion Destino ha sido atendida
346       // Una vez Posicionado en Destino, anula la Memoria de orden Destino
347       ELSIF  Ciclo_Destino >= 10  AND OK_Destino AND Tipo_Maniobra <> 20 THEN
348             Ciclo_Destino := 0 ;    // Finaliza la Secuencia de Peticion Destino
349       END_IF ;
350    // Si la Pinza no esta Abierta y no esta Arriba o el puente no esta en automatico, Anula las Or
   denes de Destino
351    ELSIF  (NOT FC_Elevacion AND NOT DT_Atras_Pinzas AND Ciclo_Destino >= 10) OR NOT Modo_Auto THEN
352        Ciclo_Destino := 0 ;    // Inicializa la Secuencia de Peticion Destino
353    END_IF ;
354
355 END_IF ;
356
357 //
358 // PUENTE DESCARGA
359 //
360 IF Carga_Descarga THEN    // Puente de Descarga
361    //
362    // Condiciones generales de Inicio Ciclo de Origen
363    //
364    IF FC_Elevacion AND NOT DT_Atras_Pinzas THEN
365       // Inicio Ciclo de Origen
366       IF Tipo_Maniobra = 10 AND Ciclo_Origen = 0 AND NOT OK_Origen THEN
367          Ciclo_Origen := 10 ;   // Orden de Inicio de Ciclo a Origen
368       // Una vez atendiendo la peticion, anula la orden de Inicio Ciclo
369       ELSIF  Ciclo_Origen = 10 AND (Pet_Ref_Inicio OR Pet_Origen OR Pet_Destino) THEN
370             Ciclo_Origen := 20 ;
371       // Una vez Posicionado a Origen y dejar de esta arriba, anula la Memoria de orden origen
372       ELSIF  Ciclo_Origen >= 10 AND OK_Origen AND Tipo_Maniobra <> 10 THEN
373             Ciclo_Origen := 0;  // Finaliza la Secuencia de Peticion Origen
374       END_IF ;
375    // Si la Pinza no esta Abierta y no esta Arriba, Anula las Ordenes de Origen
376    ELSIF  NOT FC_Elevacion AND NOT DT_Atras_Pinzas AND Ciclo_Origen >= 10 THEN
377        Ciclo_Origen := 0;  // Finaliza la Secuencia de Peticion Origen
378    END_IF ;
379    //
380    // Condiciones generales de Inicio Ciclo de Destino
381    //
382    IF FC_Elevacion AND DT_Atras_Pinzas AND Inhibir THEN
383       // Inicio Ciclo de Destino
384       IF Tipo_Maniobra = 20 AND Ciclo_Destino = 0 AND NOT OK_Destino AND Hab_Destino THEN
385          Ciclo_Destino := 10 ;       // Orden de Inicio de Ciclo a Destino
386       // Una vez atendiendo la peticion, anula la orden de Inicio Destino
387       ELSIF  Ciclo_Destino = 10 AND (Pet_Ref_Inicio OR Pet_Origen OR Pet_Destino) THEN
388             Ciclo_Destino := 20 ;    // La Peticion Destino ha sido atendida
389       // Una vez Posicionado a Destino y dejar de esta arriba, anula la Memoria de orden Destino
390       ELSIF  Ciclo_Destino >= 10  AND OK_Destino AND Tipo_Maniobra <> 20 THEN
391             Ciclo_Destino := 0 ;    // Finaliza la Secuencia de Peticion Destino
392       END_IF ;
393    // Si la Pinza no esta Abierta o no esta Arriba, Anula las Ordenes de Destino
394    ELSIF  NOT FC_Elevacion OR NOT DT_Atras_Pinzas THEN
395        Ciclo_Destino := 0 ;    // Inicializa la Secuencia de Peticion Destino
396    END_IF ;
397
398 END_IF ;
399
400 //
401 // Inicio Ciclo de GIRO DEL HORNO
402 //
403 IF Ciclo_GHorno AND Ciclo_GiroHorno = 0 AND NOT Pet_Giro_H  THEN
404    Ciclo_GiroHorno := 10   ;       // Inicio Secuencia del Ciclo Giro de Horno
405 ELSIF  Ciclo_GiroHorno = 10 THEN
406    // Solicitud de Giro del Horno, Recalculo
407    IF Pet_Giro_H AND POrden_Giro_H THEN
408       Ciclo_GiroHorno   := 20 ;      // Solicitud Atendida, espera a ser Finalizada
409       Pet_Giro_H        := FALSE;    // GH_Auto% Orden de Peticion Giro Horno, reajuste
410    ELSIF  NOT Pet_Giro_H AND NOT POrden_Giro_H THEN
411       Pet_Giro_H   := TRUE ;        // GH_Auto% Orden de Peticion Giro Horno, reajuste
412    END_IF ;
413 ELSIF  Ciclo_GiroHorno = 20 THEN
414    IF NOT Ciclo_GHorno AND NOT Pet_Giro_H AND NOT POrden_Giro_H THEN
415       Ciclo_GiroHorno   := 0 ;      // Solicitud Finalizada
416    END_IF ;
417 ELSIF  Pet_Ref_Inicio OR Pet_Origen THEN
418     // Inicializa el sistema si hay peticion a Referencia Origen
419       Ciclo_GiroHorno   := 0 ;       // Solicitud Finalizada
420       Pet_Giro_H        := FALSE;    // GH_Auto% Orden de Peticion Giro Horno, reajuste
421 END_IF ;
422
423 //
424 // ------------ Control del Sistema de posicionado ---------------------------
425 //
```

```
426
427 // Ciclo Automatico
428 // Si seguridades rearmadas, contactores de potencia activos y puente en automatico THEN
429 IF Habilitacion AND Modo_Auto THEN
430    // Proceso Automatico de Posicionado
431    CASE Ciclo_Posicion OF
432         0  :  // Inicio
433               IF Referenciado THEN            // Si esta Referenciado ( Datos de Posicion en Orig
    en - Zona de Carga/Descarga )
434                  Rapido_Lento     := FALSE;   // Anula la Velocidad Rapida
435                  Pet_Ref_Inicio   := FALSE;   // Anula Peticion de Posicionado
436                  // Segun los Casos ....
437                  IF Ciclo_Origen = 10  THEN         // Si Se desea ir a Posicion de Origen  ( Zon
    a de Carga/Descarga )
438                     Ciclo_Posicion := 20 ;
439                  ELSIF  Ciclo_Destino = 10  THEN     // Si se desea ir a Posicion de Destino ( Paq
    uetes Horno )
440                     Ciclo_Posicion := 30 ;
441                  END_IF ;
442               ELSE                              // Para culaquier Caso, Se va a Referencia a Origen
443                  IF (Ciclo_Origen = 10 OR Ciclo_Destino = 10 OR Ciclo_GiroHorno = 10 OR NOT Refere
    nciado) AND FC_Elevacion THEN
444                     Ciclo_Posicion := 10 ;
445                  ELSE
446                     Ciclo_Posicion := 0 ;
447                  END_IF ;
448               END_IF ;
449
450         10 :  // Secuencia 1 .. Referenciado a Origen
451               Rapido_Lento := FC_Max_Origen AND NOT FC_Lento_Origen;  // Si FC de parada origen ac
    tivo (contacto NC) y no FC lento se pone en velocidad rapida
452               // Control Sobre el Final del Referenciado del Equipo
453               IF Pet_Ref_Inicio AND Referenciado THEN
454                  // Final de Referencia a Origen
455                  Pet_Ref_Inicio    := FALSE;    // RF_Auto% Orden de Peticion Referencia a Origen
456                  Ciclo_Posicion    := 0 ;       // Inicio del Control
457               ELSIF  NOT Referenciado THEN
458                  Pet_Ref_Inicio   := TRUE ;     // RF_Auto% Orden de Peticion Referencia a Origen
459               END_IF ;
460
461         20 :  // Secuencia 2 .. Posicion a Origen
462               // Control Sobre la peticion del Posicionado a Origen del Equipo
463               IF Pet_Origen AND POrden_Origen THEN
464                  Ciclo_Posicion    := 25 ;      // Solicitud Atendida, espera a ser Finalizada
465               ELSIF  NOT Pet_Origen AND NOT POrden_Origen AND NOT OK_Origen THEN
466                  Pet_Origen    := TRUE ;        // PO_Auto% Orden de Peticion Posicion a Origen
467               END_IF ;
468
469         25 :  // Secuencia 2.1 .. Posicion a Origen se esta ejecutando
470               // Control Sobre el Final del Posicionado a Origen del Equipo
471               Pet_Origen   := FALSE;           // PO_Auto% Orden de Peticion Posicion a Origen
472               IF NOT Pet_Origen AND NOT POrden_Origen AND OK_Origen THEN
473                  Ciclo_Posicion    := 0 ;      // Solicitud Finalizada
474               END_IF ;
475
476         30 :  // Secuencia 3 .. Posicion a Destino
477               // Control Sobre la peticion del Posicionado a Origen del Equipo
478               IF Pet_Destino AND POrden_Destino THEN
479                  Ciclo_Posicion    := 35 ;      // Solicitud Atendida, espera a ser Finalizada
480               ELSIF  NOT Pet_Destino AND NOT POrden_Destino AND NOT OK_Destino THEN
481                  Pet_Destino   := TRUE ;        // PD_Auto% Orden de Peticion Posicion a Destino
482               END_IF ;
483
484         35 :  // Secuencia 3.1 .. Posicion a Destino se esta ejecutando
485               // Control Sobre el Final del Posicionado a Destino del Equipo
486               Pet_Destino  := FALSE;           // PD_Auto% Orden de Peticion Posicion a Destino
487               IF NOT Pet_Destino AND NOT POrden_Destino AND OK_Destino THEN
488                  Ciclo_Posicion    := 0 ;      // Solicitud Finalizada
489               END_IF ;
490
491         ELSE  ; // Resto de los casos
492
493    END_CASE ;
494
495    // Ordenes Inhabilitada en control de Posicionado
496    Avance       := FALSE ;              // Anula orden de Avance    en Automatico Modo_Bruto
497    Retroceso    := FALSE ;              // Anula orden de Retroceso en Automatico Modo_Bruto
498
499 END_IF ;
500
501 // Control General del Sistema
502 IF Habilitacion THEN
503    // Modo de Control
504       Modo_Bruto := FALSE;
505
506    // Modo Man_Auto
507       Modo_Auto := Modo ;
508
509    // Modo de Control en Automatico
```

```
510    IF NOT Modo_Auto THEN
511         // Modo no Automatico, Se puede Inicializar el sistema en Automatico
512         IF Inicializar_Pos OR Pet_Inicializar THEN
513             Pet_Inicializar := Referenciado OR POrden_Destino OR OK_Origen OR POrden_Origen OR POr
    den_Giro_H ;
514             Modo_Bruto      := FALSE ;
515         END_IF ;
516         Ciclo_Origen      := 0    ;    // Inicio Ciclo Origen
517         Ciclo_Destino     := 0    ;    // Inicio Ciclo Destino
518         Ciclo_GiroHorno   := 0    ;    // Inicio Ciclo Giro del Horno
519         Pet_Origen        := FALSE;    // PO_Auto% Orden de Peticion Posicion a Origen    Automati
    co
520      // Pet_Inicializar   := FALSE;    // IN_Auto% Orden de Inicializar Ciclo          no Automati
    co
521         Pet_Destino       := FALSE;    // PD_Auto% Orden de Peticion Posicion a Destino   Automati
    co
522         Pet_Ref_Inicio    := FALSE;    // RF_Auto% Orden de Peticion Referencia a Origen  Automati
    co
523         Pet_Giro_H        := FALSE;    // GH_Auto% Orden de Peticion Giro Horno, reajuste Automati
    co
524         Ciclo_Posicion    := 0    ;
525    END_IF ;
526 ELSE // Inicializa el sistema por defecto
527       Ciclo_Posicion    := 0    ;
528       Modo_Bruto        := FALSE;
529       Modo_Auto         := FALSE;
530       Rapido_Lento      := FALSE;    // LR_Auto% Orden Rapido/Lento                     Automatico
531       Avance            := FALSE;    // Av_Auto% Orden de Avance                        Automatico
532       Retroceso         := FALSE;    // Rt_Auto% Orden de Retroceso                     Automatico
533       Pet_Origen        := FALSE;    // PO_Auto% Orden de Peticion Posicion a Origen    Automatico
534       Pet_Inicializar   := FALSE;    // IN_Auto% Orden de Inicializar Ciclo          no Automatico
535       Pet_Destino       := FALSE;    // PD_Auto% Orden de Peticion Posicion a Destino   Automatico
536       Pet_Ref_Inicio    := FALSE;    // RF_Auto% Orden de Peticion Referencia a Origen  Automatico
537       Modo_Auto         := FALSE;    // MD_Auto% Orden Modo del UNI en                  Automatico
538       Pet_Giro_H        := FALSE;    // GH_Auto% Orden de Peticion Giro Horno, reajuste Automatico
539       Ciclo_Origen      := 0    ;    // Inicio Ciclo Origen
540       Ciclo_Destino     := 0    ;    // Inicio Ciclo Destino
541       Ciclo_GiroHorno   := 0    ;    // Inicio Ciclo Giro del Horno
542 END_IF ;
543 ;
544 END_FUNCTION
545
546 //
547 // --------------    Control Tabla de mm Posicion Puente   -----------------
548 //
549
```

```
550 FUNCTION  Puntos_Posicion: VOID   // FC009 : INT
551
552 VAR_INPUT
553   // Variables de Entrada
554    Carga_Descarga   :BOOL ;      // OFF >> Puente Carga, ON >> Puente Descarga
555    Pos_Vl_mm        :BOOL ;      // Posicionado OFF >> Nº Vueltas encoder, ON >> Milimetros
556    SP_Puesto        :INT  ;      // Preseleccion numero de Puesto Carga/Descarga
557 END_VAR
558
559 VAR_IN_OUT
560   // Variables de Entrada/Salida
561    SP_Pos_Trans      :REAL  ;     // Valor de la Posicion para la translacion  Vueltas/mm
562    SP_Pos_Carro      :REAL  ;     // Valor de la Posicion para el Carro        Vueltas/mm
563    SP_Pos_Giro       :REAL  ;     // Valor de la Posicion para el Giro         Vueltas/mm
564 END_VAR
565
566 VAR_TEMP
567   // Variables Temporal
568    Indice            :INT   ;     // Indice resultante
569 END_VAR
570
571 //
572 // Obtener Los parametros de la posicion de Carga/Decarga en funcion al Horno
573 //
574
575    IF NOT Carga_Descarga THEN    // Puente de Carga
576       // Puntero Indice
577       IF SP_Puesto >= 4 AND SP_Puesto <=27 THEN     // Parametros para la Carga
578          Indice := SP_Puesto ;  // Puntero
579       ELSE
580          Indice := 4 ;          // Puntero por defecto
581       END_IF ;
582
583       // Valores resultantes en mm
584       SP_Pos_Trans := DB_PuenteCarga.mm_Translacion[Indice];  // Valor de la Posicion para la tran
    slacion mm
585       SP_Pos_Carro := DB_PuenteCarga.mm_Carro[Indice]      ;  // Valor de la Posicion para el Carr
    o       mm
586       SP_Pos_Giro  := DB_PuenteCarga.mm_Giro[Indice]       ;  // Valor de la Posicion para el Giro
          mm
587    ELSE   // Para el Puente de Descarga   ( Puestos del 30..9 )
588       // Puntero Indice
589       IF SP_Puesto >= 9 AND SP_Puesto <=30 THEN     // Parametros para la Descarga
590          Indice := SP_Puesto ;  // Puntero
591       ELSE
592          Indice := 30 ;          // Puntero por defecto
593       END_IF ;
594       // Valores resultantes en mm
595       SP_Pos_Trans := DB_PuenteDescarga.mm_Translacion[Indice];  // Valor de la Posicion para la t
    ranslacion mm
596       SP_Pos_Carro := DB_PuenteDescarga.mm_Carro[Indice]      ;  // Valor de la Posicion para el C
    arro       mm
597       SP_Pos_Giro  := DB_PuenteDescarga.mm_Giro[Indice]       ;  // Valor de la Posicion para el G
    iro        mm
598    END_IF ;
599
600 END_FUNCTION
601
602 //
603 // --------------    Conversion REAL to INT ( BYTE )    -----------------
604 //
605
```

```
606 FUNCTION  Fun_REAL_INT:VOID        // FC007 : INT
607
608 VAR_INPUT
609   // Variables de Entrada
610    DW_Real          :REAL ;      // Palabra Fuente Real
611    Numero_Base      :REAL ;      // Palabra Base para conversion
612 END_VAR
613
614 VAR_IN_OUT
615   // Variables de Entrada/Salida
616   D_U_Dd_Ud   :INT    ;   // Decenas_Unidades_DecenasDecimal_UnidadesDecimal
617   Cm_Dm_Um_C  :INT    ;   // CentenasMillar_DecenasMillar_UnidadMillar_Centenas
618 END_VAR
619
620 VAR_TEMP
621   // Variables termporales
622   Modulo   :DINT    ;   // Decenas_Unidades_DecenasDecimal_UnidadesDecimal
623   Entero   :DINT    ;   // DentenasMillar_DecenasMillar_UnidadMillar_Centenas
624 END_VAR
625
626 // Área de instrucciones   Convertir el valor 123456.78 >> en datos para el UNI
627 // Los parametros del variador Unidrive donde hay que introducir los valores de destino son el 19.
    24 y el 19.23
628 // Valor del PLC  123456.78mm >> Uni   #19.24 = 1234*100, #19.23 = 5678/100
629 //
630   Modulo      := REAL_TO_DINT(DW_Real*100.0) MOD REAL_TO_DINT(Numero_Base);
631   Entero      := REAL_TO_DINT(((DW_Real*100.0)/ Numero_Base)-(DINT_TO_REAL(Modulo)/Numero_Base));
632
633   D_U_Dd_Ud   := DINT_TO_INT(Modulo) ;   // Resto    5678         >> 56.78
634   Cm_Dm_Um_C  := DINT_TO_INT(Entero);    // Entero   1234         >> 1234
635 ;
636 END_FUNCTION
637
638
639 //
640 // --------------    Conversion REAL to INT ( BYTE )    -----------------
641 //
642
```

```
643 FUNCTION  Tabla_Datos : VOID     //FCx15 : INT
644
645 VAR_INPUT
646   // Variables Entrada
647   mm_Puesto_04 :REAL ;  // Valor en mm del Puesto 4
648   mm_Puesto_05 :REAL ;  // Valor en mm del Puesto 5
649   mm_Puesto_06 :REAL ;  // Valor en mm del Puesto 6
650   mm_Puesto_07 :REAL ;  // Valor en mm del Puesto 7
651   mm_Puesto_08 :REAL ;  // Valor en mm del Puesto 8
652   mm_Puesto_09 :REAL ;  // Valor en mm del Puesto 9
653   mm_Puesto_10 :REAL ;  // Valor en mm del Puesto 10
654   mm_Puesto_11 :REAL ;  // Valor en mm del Puesto 11
655   mm_Puesto_12 :REAL ;  // Valor en mm del Puesto 12
656   mm_Puesto_13 :REAL ;  // Valor en mm del Puesto 13
657   mm_Puesto_14 :REAL ;  // Valor en mm del Puesto 14
658   mm_Puesto_15 :REAL ;  // Valor en mm del Puesto 15
659   mm_Puesto_16 :REAL ;  // Valor en mm del Puesto 16
660   mm_Puesto_17 :REAL ;  // Valor en mm del Puesto 17
661   mm_Puesto_18 :REAL ;  // Valor en mm del Puesto 18
662   mm_Puesto_19 :REAL ;  // Valor en mm del Puesto 19
663   mm_Puesto_20 :REAL ;  // Valor en mm del Puesto 20
664   mm_Puesto_21 :REAL ;  // Valor en mm del Puesto 21
665   mm_Puesto_22 :REAL ;  // Valor en mm del Puesto 22
666   mm_Puesto_23 :REAL ;  // Valor en mm del Puesto 23
667   mm_Puesto_24 :REAL ;  // Valor en mm del Puesto 24
668   mm_Puesto_25 :REAL ;  // Valor en mm del Puesto 25
669   mm_Puesto_26 :REAL ;  // Valor en mm del Puesto 26
670   mm_Puesto_27 :REAL ;  // Valor en mm del Puesto 27
671   mm_Puesto_28 :REAL ;  // Valor en mm del Puesto 28
672   mm_Puesto_29 :REAL ;  // Valor en mm del Puesto 29
673   mm_Puesto_30 :REAL ;  // Valor en mm del Puesto 30
674   Carga_Desc   :BOOL ;  // Tipo de Puente  OFF >> Carga  ON >> Descarga
675
676 END_VAR
677
678 VAR_IN_OUT
679    // Variables Entrada/Salida
680    Tran_Carr_Giro   :INT  ;  // Tipo de Dato  Tran >> Translacion [1] , Carr >> Carro [2] , Giro >
    > Giro [3]
681    Lim_Minimo       :INT  ;  // Limite Minimo para el Puente Carga/Descarga
682    Lim_Maximo       :INT  ;  // Limite Maximo para el Puente Carga/Descarga
683    Indice           :INT  ;  // Indice del Puntero
684    Dato_Trans       :REAL ;  // Dato a Transferir Real
685 END_VAR
686
687   // Reajuste de los Limites de Puestos
688   IF Lim_Minimo < 4  THEN
689      Lim_Minimo := 3;
690   END_IF ;
691   IF Lim_Maximo > 30 THEN
692      Lim_Maximo := 31;
693   END_IF ;
694   IF Indice < Lim_Minimo OR Indice > Lim_Maximo THEN
695      Indice     := Lim_Minimo ;
696   END_IF ;
697
698   // Control de Datos
699
700   CASE  Indice OF
701        3,4 : // Puesto 4
702              Dato_Trans := mm_Puesto_04;
703        5 : // Puesto 5
704              Dato_Trans := mm_Puesto_05;
705        6 : // Puesto 6
706              Dato_Trans := mm_Puesto_06;
707        7 : // Puesto 7
708              Dato_Trans := mm_Puesto_07;
709        8 : // Puesto 8
710              Dato_Trans := mm_Puesto_08;
711        9 : // Puesto 9
712              Dato_Trans := mm_Puesto_09;
713       10 : // Puesto 10
714              Dato_Trans := mm_Puesto_10;
715       11 : // Puesto 11
716              Dato_Trans := mm_Puesto_11;
717       12 : // Puesto 12
718              Dato_Trans := mm_Puesto_12;
719       13 : // Puesto 13
720              Dato_Trans := mm_Puesto_13;
721       14 : // Puesto 14
722              Dato_Trans := mm_Puesto_14;
723       15 : // Puesto 15
724              Dato_Trans := mm_Puesto_15;
725       16 : // Puesto 16
726              Dato_Trans := mm_Puesto_16;
727       17 : // Puesto 17
728              Dato_Trans := mm_Puesto_17;
729       18 : // Puesto 18
730              Dato_Trans := mm_Puesto_18;
```

```
731       19 : // Puesto 19
732            Dato_Trans := mm_Puesto_19;
733       20 : // Puesto 20
734            Dato_Trans := mm_Puesto_20;
735       21 : // Puesto 21
736            Dato_Trans := mm_Puesto_21;
737       22 : // Puesto 22
738            Dato_Trans := mm_Puesto_22;
739       23 : // Puesto 23
740            Dato_Trans := mm_Puesto_23;
741       24 : // Puesto 24
742            Dato_Trans := mm_Puesto_24;
743       25 : // Puesto 25
744            Dato_Trans := mm_Puesto_25;
745       26 : // Puesto 26
746            Dato_Trans := mm_Puesto_26;
747       27 : // Puesto 27
748            Dato_Trans := mm_Puesto_27;
749       28 : // Puesto 28
750            Dato_Trans := mm_Puesto_28;
751       29 : // Puesto 29
752            Dato_Trans := mm_Puesto_29;
753       30,31 : // Puesto 30
754            Dato_Trans := mm_Puesto_30;
755   END_CASE ;
756
757   // Memorizar Datos en la Tabla
758   IF NOT Carga_Desc THEN    // Puente Carga
759      IF Tran_Carr_Giro = 1 THEN    // Translacion
760         IF DB_PuenteCarga.mm_Translacion[Indice] <= 0.0 THEN
761            DB_PuenteCarga.mm_Translacion[Indice] := Dato_Trans;
762         END_IF ;
763      ELSIF Tran_Carr_Giro = 2 THEN    // Carro
764            IF DB_PuenteCarga.mm_Carro[Indice] <= 0.0 THEN
765               DB_PuenteCarga.mm_Carro[Indice] := Dato_Trans;
766            END_IF ;
767      ELSIF Tran_Carr_Giro = 3 THEN    // Giro
768            IF DB_PuenteCarga.mm_Giro[Indice] <= 0.0 THEN
769               DB_PuenteCarga.mm_Giro[Indice] := Dato_Trans;
770            END_IF ;
771      END_IF ;
772   ELSE                    // Puente Descarga
773      IF Tran_Carr_Giro = 1 THEN    // Translacion
774         IF DB_PuenteDescarga.mm_Translacion[Indice] <= 0.0 THEN
775            DB_PuenteDescarga.mm_Translacion[Indice] := Dato_Trans;
776         END_IF ;
777      ELSIF Tran_Carr_Giro = 2 THEN    // Carro
778            IF DB_PuenteDescarga.mm_Carro[Indice] <= 0.0 THEN
779               DB_PuenteDescarga.mm_Carro[Indice] := Dato_Trans;
780            END_IF ;
781      ELSIF Tran_Carr_Giro = 3 THEN    // Giro
782            IF DB_PuenteDescarga.mm_Giro[Indice] <= 0.0 THEN
783               DB_PuenteDescarga.mm_Giro[Indice] := Dato_Trans;
784            END_IF ;
785      END_IF ;
786    END_IF ;
787 ;
788 END_FUNCTION
789
790 //
791 //  Gestion Datos de las Tablas
792 //
793
794
```

```
795 FUNCTION  Gestion_Datos_Tablas : VOID    //FCx16 : INT
796
797 //
798 // Enlace sobre el Puente de Carga
799 //
800    DB_PuenteCarga.Lim_Minimo     := 4  ;
801    DB_PuenteCarga.Lim_Maximo     := 27 ;
802
803 //
804 // Datos de Puente Carga Translacion
805 //
806    DB_PuenteCarga.Tran_Carr_Giro := 1  ;
807
808      Tabla_Datos(mm_Puesto_04   :=    735.98    // IN: REAL   ***   62-25  Posicion Anterior - 23
    01/OCT/06
809                 ,mm_Puesto_05   :=   1979.98    // IN: REAL   ***   62-25  Posicion Anterior - 22
    01/OCT/06
810                 ,mm_Puesto_06   :=   3319.47    // IN: REAL   ***   62-25  Posicion Anterior - 21
    01/OCT/06
811                 ,mm_Puesto_07   :=   4715.11    // IN: REAL   ***   62-25  Posicion Anterior - 20
    01/OCT/06
812                 ,mm_Puesto_08   :=   6182.32    // IN: REAL   ***   62-25  Posicion Anterior - 19
    01/OCT/06
813                 ,mm_Puesto_09   :=   7717.29    // IN: REAL   ***   62-25  Posicion Anterior - 18
    01/OCT/06
814                 ,mm_Puesto_10   :=   9324.96    // IN: REAL   ***   62-25  Posicion Anterior - 17
    01/OCT/06
815                 ,mm_Puesto_11   :=  10927.10    // IN: REAL   ***   62-25  Posicion Anterior - 16
    01/OCT/06
816                 ,mm_Puesto_12   :=  12580.20    // IN: REAL   ***   62-25  Posicion Anterior - 15
    01/OCT/06
817                 ,mm_Puesto_13   :=  14286.70    // IN: REAL   ***   62-25  Posicion Anterior - 14
    01/OCT/06
818                 ,mm_Puesto_14   :=  16026.80    // IN: REAL   ***   62-25  Posicion Anterior - 13
    01/OCT/06
819                 ,mm_Puesto_15   :=  17754.80    // IN: REAL   ***   62-25  Posicion Anterior - 12
    01/OCT/06
820                 ,mm_Puesto_16   :=  19447.30    // IN: REAL   ***   62-25  Posicion Anterior - 11
    27/SEP/06
821                 ,mm_Puesto_17   :=  21205.70    // IN: REAL   ***   62-25  Posicion Anterior - 10
    27/SEP/06
822                 ,mm_Puesto_18   :=  22948.20    // IN: REAL   ***   62-25  Posicion Anterior - 09
    27/SEP/06
823                 ,mm_Puesto_19   :=  24684.40    // IN: REAL   ***   62-25  Posicion Anterior - 08
    27/SEP/06
824                 ,mm_Puesto_20   :=  26374.60    // IN: REAL   ***   62-25  Posicion Anterior - 07
    27/SEP/06
825                 ,mm_Puesto_21   :=  28038.00    // IN: REAL   ***   62-25  Posicion Anterior - 06
    27/SEP/06
826                 ,mm_Puesto_22   :=  29687.80    // IN: REAL   ***   62-25  Posicion Anterior - 05
    27/SEP/06
827                 ,mm_Puesto_23   :=  31280.20    // IN: REAL   ***   62-25  Posicion Anterior - 04
    26/SEP/06
828                 ,mm_Puesto_24   :=  32810.70    // IN: REAL   ***   62-25  Posicion Anterior - 03
    26/SEP/06
829                 ,mm_Puesto_25   :=  34309.20    // IN: REAL   ***   62-25  Posicion Anterior - 02
    26/SEP/06
830                 ,mm_Puesto_26   :=  35741.20    // IN: REAL   ***   62-25  Posicion Anterior - 01
    26/SEP/06
831                 ,mm_Puesto_27   :=  37103.40    // IN: REAL   ***   62-25  Posicion Origen
    26/SEP/06
832                 ,mm_Puesto_28   :=   1000.00    // IN: REAL    Solo para Pruebas de Ajueste 1.0 Mt
    s
833                 ,mm_Puesto_29   :=   1500.00    // IN: REAL    Solo para Pruebas de Ajueste 1.5 Mt
    s
834                 ,mm_Puesto_30   :=   2000.00    // IN: REAL    Solo para Pruebas de Ajueste 2.0 Mt
    s
835                 ,Carga_Desc     :=     FALSE    // INOUT: BOOL   Puente Carga
836                 ,Tran_Carr_Giro := DB_PuenteCarga.Tran_Carr_Giro   // INOUT: INT  1[Puente], 2[Car
    ro], 3[Giro]
837                 ,Lim_Minimo     := DB_PuenteCarga.Lim_Minimo       // INOUT: INT
838                 ,Lim_Maximo     := DB_PuenteCarga.Lim_Maximo       // INOUT: INT
839                 ,Indice         := DB_PuenteCarga.Posicion_Carga   // INOUT: INT
840                 ,Dato_Trans     := DB_PuenteCarga.Dato_Trans       // INOUT: REAL
841                                 ); // VOID
842 //
843 // Datos de Puente Carga Carro
844 //
845      DB_PuenteCarga.Tran_Carr_Giro := 2  ;
846
847      Tabla_Datos(mm_Puesto_04   :=   8669.00    // IN: REAL   ***   62-25  Posicion Anterior - 23
    01/OCT/06
848                 ,mm_Puesto_05   :=   7457.63    // IN: REAL   ***   62-25  Posicion Anterior - 22
    01/OCT/06
849                 ,mm_Puesto_06   :=   6322.86    // IN: REAL   ***   62-25  Posicion Anterior - 21
    01/OCT/06
850                 ,mm_Puesto_07   :=   5290.30    // IN: REAL   ***   62-25  Posicion Anterior - 20
    01/OCT/06
851                 ,mm_Puesto_08   :=   4358.91    // IN: REAL   ***   62-25  Posicion Anterior - 19
```

```
    01/OCT/06
852                 ,mm_Puesto_09   :=   3533.51    // IN: REAL    ***   62-25  Posicion Anterior - 18
    01/OCT/06
853                 ,mm_Puesto_10   :=   2791.96    // IN: REAL    ***   62-25  Posicion Anterior - 17
    01/OCT/06
854                 ,mm_Puesto_11   :=   2164.24    // IN: REAL    ***   62-25  Posicion Anterior - 16
    01/OCT/06
855                 ,mm_Puesto_12   :=   1647.95    // IN: REAL    ***   62-25  Posicion Anterior - 15
    01/OCT/06
856                 ,mm_Puesto_13   :=   1251.90    // IN: REAL    ***   62-25  Posicion Anterior - 14
    01/OCT/06
857                 ,mm_Puesto_14   :=    942.22    // IN: REAL    ***   62-25  Posicion Anterior - 13
    01/OCT/06
858                 ,mm_Puesto_15   :=    763.75    // IN: REAL    ***   62-25  Posicion Anterior - 12
    01/OCT/06
859                 ,mm_Puesto_16   :=    697.41    // IN: REAL    ***   62-25  Posicion Anterior - 11
    27/SEP/06
860                 ,mm_Puesto_17   :=    745.34    // IN: REAL    ***   62-25  Posicion Anterior - 10
    27/SEP/06
861                 ,mm_Puesto_18   :=    929.23    // IN: REAL    ***   62-25  Posicion Anterior - 09
    27/SEP/06
862                 ,mm_Puesto_19   :=   1210.71    // IN: REAL    ***   62-25  Posicion Anterior - 08
    27/SEP/06
863                 ,mm_Puesto_20   :=   1623.83    // IN: REAL    ***   62-25  Posicion Anterior - 07
    27/SEP/06
864                 ,mm_Puesto_21   :=   2120.75    // IN: REAL    ***   62-25  Posicion Anterior - 06
    27/SEP/06
865                 ,mm_Puesto_22   :=   2745.56    // IN: REAL    ***   62-25  Posicion Anterior - 05
    27/SEP/06
866                 ,mm_Puesto_23   :=   3501.67    // IN: REAL    ***   62-25  Posicion Anterior - 04
    26/SEP/06
867                 ,mm_Puesto_24   :=   4335.62    // IN: REAL    ***   62-25  Posicion Anterior - 03
    26/SEP/06
868                 ,mm_Puesto_25   :=   5275.25    // IN: REAL    ***   62-25  Posicion Anterior - 02
    26/SEP/06
869                 ,mm_Puesto_26   :=   6312.99    // IN: REAL    ***   62-25  Posicion Anterior - 01
    26/SEP/06
870                 ,mm_Puesto_27   :=   7422.66    // IN: REAL    ***   62-25  Posicion Origen
    26/SEP/06
871                 ,mm_Puesto_28   :=   1000.00    // IN: REAL     Solo para Pruebas de Ajueste 1.0 Mt
s
872                 ,mm_Puesto_29   :=   1500.00    // IN: REAL     Solo para Pruebas de Ajueste 1.5 Mt
s
873                 ,mm_Puesto_30   :=   2000.00    // IN: REAL     Solo para Pruebas de Ajueste 2.0 Mt
s
874                 ,Carga_Desc     :=     FALSE    // INOUT: BOOL
875                 ,Tran_Carr_Giro := DB_PuenteCarga.Tran_Carr_Giro   // INOUT: INT  1[Puente], 2[Car
ro], 3[Giro]
876                 ,Lim_Minimo     := DB_PuenteCarga.Lim_Minimo       // INOUT: INT
877                 ,Lim_Maximo     := DB_PuenteCarga.Lim_Maximo       // INOUT: INT
878                 ,Indice         := DB_PuenteCarga.Posicion_Carga   // INOUT: INT
879                 ,Dato_Trans     := DB_PuenteCarga.Dato_Trans       // INOUT: REAL
880                                 ); // VOID
881 //
882 // Datos de Puente Carga Giro
883 //
884    DB_PuenteCarga.Tran_Carr_Giro := 3  ;
885
886      Tabla_Datos(mm_Puesto_04   :=    276.06    // IN: REAL    ***   62-25  Posicion Anterior - 23
    01/OCT/06
887                 ,mm_Puesto_05   :=    315.58    // IN: REAL    ***   62-25  Posicion Anterior - 22
    01/OCT/06
888                 ,mm_Puesto_06   :=    351.09    // IN: REAL    ***   62-25  Posicion Anterior - 21
    01/OCT/06
889                 ,mm_Puesto_07   :=    390.57    // IN: REAL    ***   62-25  Posicion Anterior - 20
    01/OCT/06
890                 ,mm_Puesto_08   :=    430.94    // IN: REAL    ***   62-25  Posicion Anterior - 19
    01/OCT/06
891                 ,mm_Puesto_09   :=    469.23    // IN: REAL    ***   62-25  Posicion Anterior - 18
    01/OCT/06
892                 ,mm_Puesto_10   :=    504.64    // IN: REAL    ***   62-25  Posicion Anterior - 17
    01/OCT/06
893                 ,mm_Puesto_11   :=    546.56    // IN: REAL    ***   62-25  Posicion Anterior - 16
    01/OCT/06
894                 ,mm_Puesto_12   :=    583.33    // IN: REAL    ***   62-25  Posicion Anterior - 15
    01/OCT/06
895                 ,mm_Puesto_13   :=    623.54    // IN: REAL    ***   62-25  Posicion Anterior - 14
    01/OCT/06
896                 ,mm_Puesto_14   :=    660.18    // IN: REAL    ***   62-25  Posicion Anterior - 13
    01/OCT/06
897                 ,mm_Puesto_15   :=    697.02    // IN: REAL    ***   62-25  Posicion Anterior - 12
    01/OCT/06
898                 ,mm_Puesto_16   :=    742.18    // IN: REAL    ***   62-25  Posicion Anterior - 11
    27/SEP/06
899                 ,mm_Puesto_17   :=    775.16    // IN: REAL    ***   62-25  Posicion Anterior - 10
    27/SEP/06
900                 ,mm_Puesto_18   :=    814.01    // IN: REAL    ***   62-25  Posicion Anterior - 09
    27/SEP/06
901                 ,mm_Puesto_19   :=    852.45    // IN: REAL    ***   62-25  Posicion Anterior - 08
```

```
    27/SEP/06
902                 ,mm_Puesto_20   :=    893.07    // IN: REAL   ***   62-25  Posicion Anterior - 07
    27/SEP/06
903                 ,mm_Puesto_21   :=    927.03    // IN: REAL   ***   62-25  Posicion Anterior - 06
    27/SEP/06
904                 ,mm_Puesto_22   :=    966.38    // IN: REAL   ***   62-25  Posicion Anterior - 05
    27/SEP/06
905                 ,mm_Puesto_23   :=   1000.88    // IN: REAL   ***   62-25  Posicion Anterior - 04
    26/SEP/06
906                 ,mm_Puesto_24   :=   1043.56    // IN: REAL   ***   62-25  Posicion Anterior - 03
    26/SEP/06
907                 ,mm_Puesto_25   :=   1082.53    // IN: REAL   ***   62-25  Posicion Anterior - 02
    26/SEP/06
908                 ,mm_Puesto_26   :=   1117.33    // IN: REAL   ***   62-25  Posicion Anterior - 01
    26/SEP/06
909                 ,mm_Puesto_27   :=   1157.16    // IN: REAL   ***   62-25  Posicion Origen
    26/SEP/06
910                 ,mm_Puesto_28   :=    500.00    // IN: REAL    Solo para Pruebas de Ajueste 0.5 Mt
    s
911                 ,mm_Puesto_29   :=    750.00    // IN: REAL    Solo para Pruebas de Ajueste 0.75Mt
    s
912                 ,mm_Puesto_30   :=   1000.00    // IN: REAL    Solo para Pruebas de Ajueste 1.0 Mt
    s
913                 ,Carga_Desc     :=     FALSE    // INOUT: BOOL
914                 ,Tran_Carr_Giro := DB_PuenteCarga.Tran_Carr_Giro   // INOUT: INT  1[Puente], 2[Car
    ro], 3[Giro]
915                 ,Lim_Minimo     := DB_PuenteCarga.Lim_Minimo       // INOUT: INT
916                 ,Lim_Maximo     := DB_PuenteCarga.Lim_Maximo       // INOUT: INT
917                 ,Indice         := DB_PuenteCarga.Posicion_Carga   // INOUT: INT
918                 ,Dato_Trans     := DB_PuenteCarga.Dato_Trans       // INOUT: REAL
919                                 ); // VOID
920
921 //
922 // ------------------------------------------------------------------------------------------
    -----------------
923 //
924 //
925 // Enlace sobre el Puente de Descarga
926 //
927    DB_PuenteDescarga.Lim_Minimo     := 9  ;
928    DB_PuenteDescarga.Lim_Maximo     := 30 ;
929
930 //
931 // Datos de Puente Descarga Translacion
932 //
933    DB_PuenteDescarga.Tran_Carr_Giro := 1  ;
934
935      Tabla_Datos(mm_Puesto_04   :=   2000.00    // IN: REAL    Solo para Pruebas de Ajueste 2.0 Mt
    s
936                 ,mm_Puesto_05   :=   1500.00    // IN: REAL    Solo para Pruebas de Ajueste 1.5 Mt
    s
937                 ,mm_Puesto_06   :=   1000.00    // IN: REAL    Solo para Pruebas de Ajueste 1.0 Mt
    s
938                 ,mm_Puesto_07   :=   1500.00    // IN: REAL    Solo para Pruebas de Ajueste 1.5 Mt
    s
939                 ,mm_Puesto_08   :=   2000.00    // IN: REAL    Solo para Pruebas de Ajueste 2.0 Mt
    s
940                 ,mm_Puesto_09   :=  38573.80    // IN: REAL   ***   62-25  Posicion Anterior - 18
    01/OCT/06
941                 ,mm_Puesto_10   :=  36963.90    // IN: REAL   ***   62-25  Posicion Anterior - 17
    01/OCT/06
942                 ,mm_Puesto_11   :=  35528.10    // IN: REAL   ***   62-25  Posicion Anterior - 16
    01/OCT/06
943                 ,mm_Puesto_12   :=  33682.40    // IN: REAL   ***   62-25  Posicion Anterior - 15
    01/OCT/06
944                 ,mm_Puesto_13   :=  31961.40    // IN: REAL   ***   62-25  Posicion Anterior - 14
    01/OCT/06
945                 ,mm_Puesto_14   :=  30222.30    // IN: REAL   ***   62-25  Posicion Anterior - 13
    01/OCT/06
946                 ,mm_Puesto_15   :=  28491.90    // IN: REAL   ***   62-25  Posicion Anterior - 12
    01/OCT/06
947                 ,mm_Puesto_16   :=  26759.40    // IN: REAL   ***   62-25  Posicion Anterior - 11
    27/SEP/06
948                 ,mm_Puesto_17   :=  25020.00    // IN: REAL   ***   62-25  Posicion Anterior - 10
    27/SEP/06
949                 ,mm_Puesto_18   :=  23247.00    // IN: REAL   ***   62-25  Posicion Anterior - 09
    27/SEP/06
950                 ,mm_Puesto_19   :=  21520.40    // IN: REAL   ***   62-25  Posicion Anterior - 08
    27/SEP/06
951                 ,mm_Puesto_20   :=  19809.00    // IN: REAL   ***   62-25  Posicion Anterior - 07
    27/SEP/06
952                 ,mm_Puesto_21   :=  18126.70    // IN: REAL   ***   62-25  Posicion Anterior - 06
    27/SEP/06
953                 ,mm_Puesto_22   :=  16512.10    // IN: REAL   ***   62-25  Posicion Anterior - 05
    27/SEP/06
954                 ,mm_Puesto_23   :=  14886.60    // IN: REAL   ***   62-25  Posicion Anterior - 04
    26/SEP/06
955                 ,mm_Puesto_24   :=  13351.20    // IN: REAL   ***   62-25  Posicion Anterior - 03
    26/SEP/06
```

```
956                 ,mm_Puesto_25   :=  11870.00    // IN: REAL   ***   62-25  Posicion Anterior - 02
     26/SEP/06
957                 ,mm_Puesto_26   :=  10419.70    // IN: REAL   ***   62-25  Posicion Anterior - 01
     26/SEP/06
958                 ,mm_Puesto_27   :=   9107.14    // IN: REAL   ***   62-25  Posicion Origen
     26/SEP/06
959                 ,mm_Puesto_28   :=   7830.85    // IN: REAL   ***     62-25  Posicion Provisional
     02/AGO/06
960                 ,mm_Puesto_29   :=   6614.40    // IN: REAL   ***     62-25  Posicion Provisional
     02/AGO/06
961                 ,mm_Puesto_30   :=   5518.09    // IN: REAL   ***     62-25  Posicion Provisional
     02/AGO/06
962                 ,Carga_Desc     :=      TRUE    // INOUT: BOOL
963                 ,Tran_Carr_Giro := DB_PuenteDescarga.Tran_Carr_Giro   // INOUT: INT  1[Puente], 2[
     Carro], 3[Giro]
964                 ,Lim_Minimo     := DB_PuenteDescarga.Lim_Minimo       // INOUT: INT
965                 ,Lim_Maximo     := DB_PuenteDescarga.Lim_Maximo       // INOUT: INT
966                 ,Indice         := DB_PuenteDescarga.Posicion_Descarga// INOUT: INT
967                 ,Dato_Trans     := DB_PuenteDescarga.Dato_Trans       // INOUT: REAL
968                                 ); // VOID
969 //
970 // Datos de Puente Descarga Carro
971 //
972      DB_PuenteDescarga.Tran_Carr_Giro := 2  ;
973
974      Tabla_Datos(mm_Puesto_04   :=   2000.00    // IN: REAL    Solo para Pruebas de Ajueste 2.0 Mt
     s
975                 ,mm_Puesto_05   :=   1500.00    // IN: REAL    Solo para Pruebas de Ajueste 1.5 Mt
     s
976                 ,mm_Puesto_06   :=   1000.00    // IN: REAL    Solo para Pruebas de Ajueste 1.0 Mt
     s
977                 ,mm_Puesto_07   :=   1500.00    // IN: REAL    Solo para Pruebas de Ajueste 1.5 Mt
     s
978                 ,mm_Puesto_08   :=   2000.00    // IN: REAL    Solo para Pruebas de Ajueste 2.0 Mt
     s
979                 ,mm_Puesto_09   :=   9018.75    // IN: REAL   ***   62-25  Posicion Anterior - 18
     01/OCT/06
980                 ,mm_Puesto_10   :=   9807.02    // IN: REAL   ***   62-25  Posicion Anterior - 17
     01/OCT/06
981                 ,mm_Puesto_11   :=  10455.10    // IN: REAL   ***   62-25  Posicion Anterior - 16
     01/OCT/06
982                 ,mm_Puesto_12   :=  11009.10    // IN: REAL   ***   62-25  Posicion Anterior - 15
     01/OCT/06
983                 ,mm_Puesto_13   :=  11421.40    // IN: REAL   ***   62-25  Posicion Anterior - 14
     01/OCT/06
984                 ,mm_Puesto_14   :=  11728.30    // IN: REAL   ***   62-25  Posicion Anterior - 13
     01/OCT/06
985                 ,mm_Puesto_15   :=  11902.30    // IN: REAL   ***   62-25  Posicion Anterior - 12
     01/OCT/06
986                 ,mm_Puesto_16   :=  11960.50    // IN: REAL   ***   62-25  Posicion Anterior - 11
     27/SEP/06
987                 ,mm_Puesto_17   :=  11912.30    // IN: REAL   ***   62-25  Posicion Anterior - 10
     27/SEP/06
988                 ,mm_Puesto_18   :=  11739.00    // IN: REAL   ***   62-25  Posicion Anterior - 09
     27/SEP/06
989                 ,mm_Puesto_19   :=  11435.80    // IN: REAL   ***   62-25  Posicion Anterior - 08
     27/SEP/06
990                 ,mm_Puesto_20   :=  11020.90    // IN: REAL   ***   62-25  Posicion Anterior - 07
     27/SEP/06
991                 ,mm_Puesto_21   :=  10479.90    // IN: REAL   ***   62-25  Posicion Anterior - 06
     27/SEP/06
992                 ,mm_Puesto_22   :=   9827.01    // IN: REAL   ***   62-25  Posicion Anterior - 05
     27/SEP/06
993                 ,mm_Puesto_23   :=   9032.26    // IN: REAL   ***   62-25  Posicion Anterior - 04
     26/SEP/06
994                 ,mm_Puesto_24   :=   8181.42    // IN: REAL   ***   62-25  Posicion Anterior - 03
     26/SEP/06
995                 ,mm_Puesto_25   :=   7205.00    // IN: REAL   ***   62-25  Posicion Anterior - 02
     26/SEP/06
996                 ,mm_Puesto_26   :=   6119.84    // IN: REAL   ***   62-25  Posicion Anterior - 01
     26/SEP/06
997                 ,mm_Puesto_27   :=   4925.59    // IN: REAL   ***   62-25  Posicion Origen
     26/SEP/06
998                 ,mm_Puesto_28   :=   3768.62    // IN: REAL   ***   62-25  Posicion Provisional 02
     /AGO/06
999                 ,mm_Puesto_29   :=   2388.03    // IN: REAL   ***   62-25  Posicion Provisional 02
     /AGO/06
1000                ,mm_Puesto_30   :=    959.99    // IN: REAL   ***   62-25  Posicion Provisional 02
     /AGO/06
1001                ,Carga_Desc     :=      TRUE    // INOUT: BOOL
1002                ,Tran_Carr_Giro := DB_PuenteDescarga.Tran_Carr_Giro   // INOUT: INT  1[Puente], 2[
     Carro], 3[Giro]
1003                ,Lim_Minimo     := DB_PuenteDescarga.Lim_Minimo       // INOUT: INT
1004                ,Lim_Maximo     := DB_PuenteDescarga.Lim_Maximo       // INOUT: INT
1005                ,Indice         := DB_PuenteDescarga.Posicion_Descarga// INOUT: INT
1006                ,Dato_Trans     := DB_PuenteDescarga.Dato_Trans       // INOUT: REAL
1007                                ); // VOID
1008 //
1009 // Datos de Puente Descarga Giro
```

```
1010 //
1011    DB_PuenteDescarga.Tran_Carr_Giro := 3  ;
1012
1013      Tabla_Datos(mm_Puesto_04   :=   1000.00    // IN: REAL    Solo para Pruebas de Ajueste 1.0 Mt
     s
1014                 ,mm_Puesto_05   :=    750.00    // IN: REAL    Solo para Pruebas de Ajueste 0.75Mt
     s
1015                 ,mm_Puesto_06   :=    500.00    // IN: REAL    Solo para Pruebas de Ajueste 0.5 Mt
     s
1016                 ,mm_Puesto_07   :=    750.00    // IN: REAL    Solo para Pruebas de Ajueste 0.75Mt
     s
1017                 ,mm_Puesto_08   :=   1000.00    // IN: REAL    Solo para Pruebas de Ajueste 1.0 Mt
     s
1018                 ,mm_Puesto_09   :=    980.03    // IN: REAL   ***   62-25  Posicion Anterior - 18
      01/OCT/06
1019                 ,mm_Puesto_10   :=    947.75    // IN: REAL   ***   62-25  Posicion Anterior - 17
      01/OCT/06
1020                 ,mm_Puesto_11   :=    910.61    // IN: REAL   ***   62-25  Posicion Anterior - 16
      01/OCT/06
1021                 ,mm_Puesto_12   :=    882.56    // IN: REAL   ***   62-25  Posicion Anterior - 15
      01/OCT/06
1022                 ,mm_Puesto_13   :=    844.95    // IN: REAL   ***   62-25  Posicion Anterior - 14
      01/OCT/06
1023                 ,mm_Puesto_14   :=    817.13    // IN: REAL   ***   62-25  Posicion Anterior - 13
      01/OCT/06
1024                 ,mm_Puesto_15   :=    784.04    // IN: REAL   ***   62-25  Posicion Anterior - 12
      01/OCT/06
1025                 ,mm_Puesto_16   :=    744.73    // IN: REAL   ***   62-25  Posicion Anterior - 11
      27/SEP/06
1026                 ,mm_Puesto_17   :=    714.66    // IN: REAL   ***   62-25  Posicion Anterior - 10
      27/SEP/06
1027                 ,mm_Puesto_18   :=    685.24    // IN: REAL   ***   62-25  Posicion Anterior - 09
      27/SEP/06
1028                 ,mm_Puesto_19   :=    656.11    // IN: REAL   ***   62-25  Posicion Anterior - 08
      27/SEP/06
1029                 ,mm_Puesto_20   :=    618.48    // IN: REAL   ***   62-25  Posicion Anterior - 07
      27/SEP/06
1030                 ,mm_Puesto_21   :=    588.39    // IN: REAL   ***   62-25  Posicion Anterior - 06
      27/SEP/06
1031                 ,mm_Puesto_22   :=    557.86    // IN: REAL   ***   62-25  Posicion Anterior - 05
      27/SEP/06
1032                 ,mm_Puesto_23   :=    533.16    // IN: REAL   ***   62-25  Posicion Anterior - 04
      26/SEP/06
1033                 ,mm_Puesto_24   :=    499.87    // IN: REAL   ***   62-25  Posicion Anterior - 03
      26/SEP/06
1034                 ,mm_Puesto_25   :=    468.05    // IN: REAL   ***   62-25  Posicion Anterior - 02
      26/SEP/06
1035                 ,mm_Puesto_26   :=    437.16    // IN: REAL   ***   62-25  Posicion Anterior - 01
      26/SEP/06
1036                 ,mm_Puesto_27   :=    406.90    // IN: REAL   ***   62-25  Posicion Origen
      26/SEP/06
1037                 ,mm_Puesto_28   :=    375.69    // IN: REAL   ***    62-25  Posicion Provisional 0
     2/AGO/06
1038                 ,mm_Puesto_29   :=    341.58    // IN: REAL   ***    62-25  Posicion Provisional 0
     2/AGO/06
1039                 ,mm_Puesto_30   :=    312.92    // IN: REAL   ***    62-25  Posicion Provisional 0
     2/AGO/06
1040                 ,Carga_Desc     :=      TRUE    // INOUT: BOOL
1041                 ,Tran_Carr_Giro := DB_PuenteDescarga.Tran_Carr_Giro   // INOUT: INT  1[Puente], 2[
     Carro], 3[Giro]
1042                 ,Lim_Minimo     := DB_PuenteDescarga.Lim_Minimo       // INOUT: INT
1043                 ,Lim_Maximo     := DB_PuenteDescarga.Lim_Maximo       // INOUT: INT
1044                 ,Indice         := DB_PuenteDescarga.Posicion_Descarga// INOUT: INT
1045
1046                 ,Dato_Trans     := DB_PuenteDescarga.Dato_Trans       // INOUT: REAL
1047                                 ); // VOID
1048
1049 END_FUNCTION
1050
1051
1052 //
1053 //   ------------------- Gestion datos Pantalla <> PLC  -----------------------------
1054 //
1055
```

```
1056 FUNCTION  Gestion_Pantalla: VOID       //FCx17 : INT
1057
1058 VAR_TEMP
1059   // Variables temporales
1060   Indice   :INT ;       // Indice de Tablas
1061 END_VAR
1062
1063 //
1064 //  Datos de Posicion de Ejes Puente de Carga
1065 //
1066 //  Validar Datos Manuales desde la Pantalla a Posicion Destino.
1067 // Se lleva el puente manual a la posicion deseada y pulsando actualizar, se fija esa posicion com
     o la de destino de ese puesto.
1068 IF PC_Mem_Act_Datos THEN
1069    DB_PuenteCarga.mm_Translacion[DB_PuenteCarga.Posicion_Carga] := DB_PuenteCarga.SP_mm_Puente_Pan
     ;
1070    DB_PuenteCarga.mm_Carro[DB_PuenteCarga.Posicion_Carga]       := DB_PuenteCarga.SP_mm_Carro_Pan
     ;
1071    DB_PuenteCarga.mm_Giro[DB_PuenteCarga.Posicion_Carga]        := DB_PuenteCarga.SP_mm_Giro_Pan
     ;
1072    PC_Mem_Act_Datos                                             := FALSE ;
1073 END_IF ;
1074
1075 // Igualar Datos Posicion Actual a Posicion destino   (Puente de Carga )
1076 // los datos se introducen numericamente desde la pantalla y se le da a igualar
1077 IF PC_Mem_Igualar_Datos THEN
1078    DB_PuenteCarga.mm_Translacion[DB_PuenteCarga.Posicion_Carga] := DB_PuenteCarga.mm_Real_Puente
     ;
1079    DB_PuenteCarga.mm_Carro[DB_PuenteCarga.Posicion_Carga]       := DB_PuenteCarga.mm_Real_Carro
     ;
1080    DB_PuenteCarga.mm_Giro[DB_PuenteCarga.Posicion_Carga]        := DB_PuenteCarga.mm_Real_Giro
     ;
1081    DB_PuenteCarga.SP_mm_Puente_Pan                              := DB_PuenteCarga.mm_Real_Puente
     ;
1082    DB_PuenteCarga.SP_mm_Carro_Pan                               := DB_PuenteCarga.mm_Real_Carro
     ;
1083    DB_PuenteCarga.SP_mm_Giro_Pan                                := DB_PuenteCarga.mm_Real_Giro
     ;
1084    PC_Mem_Igualar_Datos                                         := FALSE ;
1085 END_IF ;
1086
1087 //
1088 //  Datos de Posicion de Ejes Puente de Descarga
1089 //
1090 //  Validar Datos Manuales desde Pantalla a Posicion Destino
1091 IF PD_Mem_Act_Datos THEN
1092    DB_PuenteDescarga.mm_Translacion[DB_PuenteDescarga.Posicion_Descarga] := DB_PuenteDescarga.SP_m
     m_Puente_Pan ;
1093    DB_PuenteDescarga.mm_Carro[DB_PuenteDescarga.Posicion_Descarga]       := DB_PuenteDescarga.SP_m
     m_Carro_Pan  ;
1094    DB_PuenteDescarga.mm_Giro[DB_PuenteDescarga.Posicion_Descarga]        := DB_PuenteDescarga.SP_m
     m_Giro_Pan   ;
1095    PD_Mem_Act_Datos                                                      := FALSE ;
1096 END_IF ;
1097
1098 // Igualar Datos Posicion Actual a Posicion destino   (Puente de Descarga )
1099 IF PD_Mem_Igualar_Datos THEN
1100    DB_PuenteDescarga.mm_Translacion[DB_PuenteDescarga.Posicion_Descarga] := DB_PuenteDescarga.mm_R
     eal_Puente   ;
1101    DB_PuenteDescarga.mm_Carro[DB_PuenteDescarga.Posicion_Descarga]       := DB_PuenteDescarga.mm_R
     eal_Carro    ;
1102    DB_PuenteDescarga.mm_Giro[DB_PuenteDescarga.Posicion_Descarga]        := DB_PuenteDescarga.mm_R
     eal_Giro     ;
1103    DB_PuenteDescarga.SP_mm_Puente_Pan                                    := DB_PuenteDescarga.mm_R
     eal_Puente   ;
1104    DB_PuenteDescarga.SP_mm_Carro_Pan                                     := DB_PuenteDescarga.mm_R
     eal_Carro    ;
1105    DB_PuenteDescarga.SP_mm_Giro_Pan                                      := DB_PuenteDescarga.mm_R
     eal_Giro     ;
1106    PD_Mem_Igualar_Datos                                                  := FALSE ;
1107 END_IF ;
1108
1109 //
1110 //  Datos de Estado UNI Puente de Carga
1111 //
1112
1113   CASE DB_PuenteCarga.SP_Estado_Uni OF
1114        0  : // Estado UNI - A1   EB40 , EB41   >>Parametro #10.01 del variador
1115             DB_PuenteCarga.PV_Estado_UNI := WORD_TO_INT(PEB40)*256 + WORD_TO_INT(PEB41);
1116        1  : // Estado UNI - A2   EB60 , EB61   >> #10.01
1117             DB_PuenteCarga.PV_Estado_UNI := WORD_TO_INT(PEB60)*256 + WORD_TO_INT(PEB61);
1118        2  : // Estado UNI - A3   EB80 , EB81   >> #10.01
1119             DB_PuenteCarga.PV_Estado_UNI := WORD_TO_INT(PEB80)*256 + WORD_TO_INT(PEB81);
1120        3  : // Estado UNI - A4   EB100, EB101  >> #10.01
1121             DB_PuenteCarga.PV_Estado_UNI := WORD_TO_INT(PEB100)*256 + WORD_TO_INT(PEB101);
1122   ELSE :     // Instrucciones_ELSE
1123     ;
1124   END_CASE ;
1125
```

```
1126 //
1127 //  Datos de Estado UNI Puente Descarga
1128 //
1129   CASE  DB_PuenteDescarga.SP_Estado_Uni OF
1130         0  : // Estado UNI - A1   EB140 , EB141   >> #10.01
1131              DB_PuenteDescarga.PV_Estado_UNI := WORD_TO_INT(PEB140)*256 + WORD_TO_INT(PEB141);
1132         1  : // Estado UNI - A2   EB160 , EB161   >> #10.01
1133              DB_PuenteDescarga.PV_Estado_UNI := WORD_TO_INT(PEB160)*256 + WORD_TO_INT(PEB161);
1134         2  : // Estado UNI - A3   EB180 , EB181   >> #10.01
1135              DB_PuenteDescarga.PV_Estado_UNI := WORD_TO_INT(PEB180)*256 + WORD_TO_INT(PEB181);
1136         3  : // Estado UNI - A4   EB200, EB201    >> #10.01
1137              DB_PuenteDescarga.PV_Estado_UNI := WORD_TO_INT(PEB200)*256 + WORD_TO_INT(PEB201);
1138   ELSE :     // Instrucciones_ELSE
1139     ;
1140   END_CASE ;
1141
1142 ;
1143 END_FUNCTION
1144
1145 //
1146 //  Funcion de Reiniciar el Posicionado Automatico de un Eje si en Origen, no esta Posicionado
1147 //
1148
```

```
1149 FUNCTION  Referenciar_Ejes : VOID       // FCx18 : INT
1150
1151 VAR_INPUT
1152     // Variables de Entrada
1153     Modo            :BOOL  ;     // Modo del Sistema
1154     Clock           :BOOL  ;     // Pulso 0.1 Seg.
1155     FC_Arriba       :BOOL  ;     // Estado que permite controlar el proceso, la pinza debe de esta
     r arriba para no colisionar.
1156     FC_Origen       :BOOL  ;     // Estado del FC. de Origen del Motor
1157     Pos_Referencia  :BOOL  ;     // Eje Referenciado
1158     Pos_Origen      :BOOL  ;     // Eje Posicionado en Origen
1159     Pos_Destino     :BOOL  ;     // Eje Posicionado en Destino
1160     Eje_Parado      :BOOL  ;     // Estado del Eje Parado, Velocidad "0"
1161     Freno_Eje       :BOOL  ;     // Estado del Freno del Eje
1162     Ciclo_Maniobra  :INT   ;     // Tipo de Manionbra que esta realizando
1163                                  // (0 > Nada, 10 > Referenciando a Origen// , 20-25 > Posicionado
     se en Origen
1164                                  // 30-35 > Posicionadose en Destino )
1165 END_VAR
1166
1167 VAR_IN_OUT
1168     // Variables de Entrada/Salida
1169     Modo_Eje        :BOOL  ;    // Permite el cambio de Modo del Eje OFF [MAN], ON [AUT]
1170     Referenciado    :BOOL  ;    // Orden de Referenciado
1171     Situacion_Eje   :BOOL  ;    // Indica la situacion del Eje  OFF [ Sin Posicionar ] , ON [ Posi
     cionado ]
1172     PV_Tim_Sup      :INT   ;    // Contaje Tiempo de Supervision
1173     PV_Tim_OK       :INT   ;    // Contaje Tiempo de Sistema OK
1174 END_VAR
1175
1176
1177 //
1178 //   Cuerpo del Programa
1179 //
1180      Modo_Eje := Modo ;
1181
1182      // Control de la peticion automatico de Inicializar Eje
1183      //
1184      // Si el PC esta arriba y esta en automatico
1185      IF  FC_Arriba AND Modo THEN
1186         // Si el eje está referenciado y no está en el origen, ni en destino y se encuentra parado
1187         // y tiene activa la maniobra de regresar a origen, Entonces si transcurren 2.5 segundos d
     a la orden de referenciar el Eje
1188         // y pone el eje en manual, por lo que el puente queda parado hasta que se reactive por un
      operador.
1189         IF Pos_Referencia AND NOT Pos_Origen AND NOT Pos_Destino AND Eje_Parado AND NOT Freno_Eje
     AND Ciclo_Maniobra = 25 THEN
1190            PV_Tim_OK     := 0     ;
1191            Referenciado  := FALSE ;
1192            Situacion_Eje := FALSE ;
1193            IF Clock AND PV_Tim_Sup < 25 THEN
1194               PV_Tim_Sup := PV_Tim_Sup + 1;
1195            ELSIF  PV_Tim_Sup >= 25 THEN
1196                Modo_Eje         := FALSE ;
1197                Referenciado     := TRUE  ;
1198                Situacion_Eje    := FALSE ;
1199                PV_Tim_Sup       := 0     ;
1200         END_IF ;
1201         // Si el eje se encuentra parado en la posicion de origen y no tiene ninguna maniobra acti
     va,
1202         // transcurridos 0.8 sg el eje ya se encuentra referenciado, y se pone el eje en automatic
     o.
1203         ELSIF  Pos_Referencia AND Pos_Origen AND NOT Pos_Destino AND Eje_Parado AND NOT Freno_Eje
     AND Ciclo_Maniobra = 0 THEN
1204               PV_Tim_Sup    := 0     ;
1205               IF Clock AND PV_Tim_OK < 8 THEN
1206                  PV_Tim_OK := PV_Tim_OK + 1;
1207               ELSIF  PV_Tim_OK >= 8 THEN
1208                   Modo_Eje         := TRUE  ;
1209                   Referenciado     := FALSE ;
1210                   Situacion_Eje    := TRUE  ;
1211                   PV_Tim_OK        := 0     ;
1212               END_IF ;
1213         ELSIF  NOT Eje_Parado THEN
1214               PV_Tim_OK    := 0 ;
1215               PV_Tim_Sup   := 0 ;
1216         END_IF ;
1217      ELSE
1218            PV_Tim_OK    := 0 ;
1219            PV_Tim_Sup   := 0 ;
1220      END_IF ;
1221
1222 END_FUNCTION
1223
1224 //
1225 //    Funcion Inhibicion Hiro del Horno
1226 //
1227
```

```
1228 FUNCTION  Inhibir_Giro_Horno: VOID       //FCx19 : INT
1229
1230 VAR_INPUT
1231     // Variables de Entrada
1232     Clock           :BOOL  ;     // Pulso 1 Seg.
1233     Horno_Girando   :BOOL  ;     // Estado del Horno Girando
1234     // Variables Puente de Carga
1235     PC_Modo         :BOOL  ;     // Modo Puente de Carga
1236     PC_FC_Arriba    :BOOL  ;     // Estado que permite controlar el proceso
1237     PC_Pos_Referen  :BOOL  ;     // Eje Referenciado
1238     PC_Pos_Origen   :BOOL  ;     // Eje Posicionado en Origen
1239     PC_Pos_Destino  :BOOL  ;     // Eje Posicionado en Destino
1240     // Variables Puente de Descarga
1241     PD_Modo         :BOOL  ;     // Modo Puente de Descarga
1242     PD_FC_Arriba    :BOOL  ;     // Estado que permite controlar el proceso
1243     PD_Pos_Referen  :BOOL  ;     // Eje Referenciado
1244     PD_Pos_Origen   :BOOL  ;     // Eje Posicionado en Origen
1245     PD_Pos_Destino  :BOOL  ;     // Eje Posicionado en Destino
1246
1247 END_VAR
1248
1249 VAR_IN_OUT
1250     // Variables de Entrada/Salida
1251     Inhibir_GHorno  :BOOL  ;    // Salida Inhibicion Giro del Horno
1252     PC_Inh_GHorno   :BOOL  ;    // Condiciones de Inhibicion Giro del Horno P.Carga
1253     PD_Inh_GHorno   :BOOL  ;    // Condiciones de Inhibicion Giro del Horno P.Descarga
1254 END_VAR
1255
1256 // Condiciones con el Horno Parado
1257 IF  NOT Horno_Girando THEN
1258    // Puente de Carga
1259    IF  PC_Modo THEN
1260       IF  PC_Pos_Referen THEN
1261          IF  NOT PC_Pos_Origen AND (PC_Pos_Destino OR NOT PC_FC_Arriba) THEN
1262             PC_Inh_GHorno := TRUE ;
1263          ELSIF  (PC_Pos_Origen OR PC_FC_Arriba) AND NOT PC_Pos_Destino THEN
1264             PC_Inh_GHorno := FALSE;
1265        END_IF ;
1266       ELSE
1267           PC_Inh_GHorno   := FALSE;
1268       END_IF ;
1269    ELSE
1270       PC_Inh_GHorno   := FALSE;
1271    END_IF ;
1272
1273    // Puente de Descarga
1274    IF  PD_Modo THEN
1275       IF  PD_Pos_Referen THEN
1276          IF  NOT PD_Pos_Origen AND (PD_Pos_Destino OR NOT PC_FC_Arriba) THEN
1277             PD_Inh_GHorno := TRUE ;
1278          ELSIF  (PD_Pos_Origen OR PD_FC_Arriba) AND NOT PD_Pos_Destino THEN
1279             PD_Inh_GHorno := FALSE;
1280        END_IF ;
1281       ELSE
1282           PD_Inh_GHorno   := FALSE;
1283       END_IF ;
1284    ELSE
1285       PD_Inh_GHorno   := FALSE;
1286    END_IF ;
1287 ELSE
1288     PC_Inh_GHorno   := FALSE;
1289     PD_Inh_GHorno   := FALSE;
1290 END_IF ;
1291
1292 // Inhibicion Giro del Horno
1293 Inhibir_GHorno  := PC_Inh_GHorno OR PD_Inh_GHorno;
1294
1295 END_FUNCTION
1296
1297
1298
1299 //
1300 //  Funcion Tiempos de Ciclo para Translacion, Carro, y Giro en los puentes de carga y descarga
1301 //
1302
```

```
1303 FUNCTION  Tiempos_Ciclo : VOID   // FCx20 : INT
1304
1305 VAR_INPUT
1306   // Variables de entrada
1307   Habilitacion  : BOOL ;     // Habilitacion del sistema
1308   Clock         : BOOL ;    // Pulso de Tiempos
1309   Ciclo_Carga   : BOOL ;    // Ciclo de carga
1310   Ciclo_Desc    : BOOL ;    // Ciclo de descarga
1311 END_VAR
1312
1313 VAR_IN_OUT
1314   // Variables de Entrada/Salida
1315   Mem_Ciclo_Carga : BOOL ;   // Memoria en Ciclo de carga
1316   Mem_Ciclo_Desc  : BOOL ;   // Memoria en Ciclo de descarga
1317   PV_Ciclo_Carga  : INT  ;   // Contaje Ciclo Carga, parcial
1318   PV_Ciclo_Desc   : INT  ;   // Contaje Ciclo Descarga, parcial
1319   TIM_Ciclo_Carga : INT  ;   // Tiempo Total Ciclo de Carga
1320   TIM_Ciclo_Desc  : INT  ;   // Tiempo Total Ciclo de Descarga
1321 END_VAR
1322
1323 // Funcion de Contaje
1324 IF Habilitacion THEN
1325    //
1326    // Ciclo de Carga ( Hacia el Horno )
1327    IF Ciclo_Carga AND NOT Mem_Ciclo_Carga THEN
1328       Mem_Ciclo_Carga   := TRUE ;
1329       PV_Ciclo_Carga    := 0 ;
1330    ELSIF Ciclo_Carga AND Mem_Ciclo_Carga AND Clock THEN
1331       PV_Ciclo_Carga := PV_Ciclo_Carga + 1;
1332    ELSIF NOT Ciclo_Carga AND Mem_Ciclo_Carga THEN
1333       TIM_Ciclo_Carga   := PV_Ciclo_Carga ;
1334       Mem_Ciclo_Carga   := FALSE ;
1335       PV_Ciclo_Carga    := 0 ;
1336    END_IF ;
1337    //
1338    // Ciclo de Descarga ( Hacia el Origen )
1339    IF Ciclo_Desc AND NOT Mem_Ciclo_Desc THEN
1340       Mem_Ciclo_Desc   := TRUE ;
1341       PV_Ciclo_Desc    := 0 ;
1342    ELSIF Ciclo_Desc AND Mem_Ciclo_Desc AND Clock THEN
1343       PV_Ciclo_Desc := PV_Ciclo_Desc + 1;
1344    ELSIF NOT Ciclo_Desc AND Mem_Ciclo_Desc THEN
1345       TIM_Ciclo_Desc   := PV_Ciclo_Desc ;
1346       Mem_Ciclo_Desc   := FALSE ;
1347       PV_Ciclo_Desc    := 0 ;
1348    END_IF ;
1349 ELSE
1350    Mem_Ciclo_Carga := FALSE ;
1351    Mem_Ciclo_Desc  := FALSE ;
1352 END_IF ;
1353
1354 END_FUNCTION
1355
```