PROGRAMACIÓN DE UNA CELDA DE MECANIZADO

Jorge Bengoa Pinedo	EE309
Luis Miguel Muro García	EE309
Gil Carvalho Tavares	EE309

ÍNDICE

1. Layout esqumático de la celda
    a. Descripción
2. Catálogo de comunicaciones
    a. Metacontrol
    b. UP_Robot
    c. UP_CNC
    d. UP_Simple_Efecto
    e. UP_Palets_Vacios
3. Gafcet nivel 2
    a. GEMER_Global
    b. GEMER_Local
    c. GMM
    d. UP_Simple_Efecto (G1)
    e. UP_CNC (G2)
    f. UP_Robot (G3)
    g. UP_Palets_Vacios (G4)
    h. SFC_Cilindro
    i. SFC_Programa
4. Programación
    a. Init (OB100)
    b. Main (OB1)
        i. GEMER_Global
        ii. GEMER_Local
        iii. Rearmes
        iv. GMM
        v. Unidades productivas
    c. Metacontrol

i. GEMER_Global
    1. Transiciones de estados
    2. Emergencia
ii. GEMER_Local
    1. Plantilla de forzado – Emergencia
    2. Resto de Segmentos
iii. GMM
    1. Plantilla de conexión
    2. Plantilla de forzado – Error asíncrono
    3. Tratamiento tamaño lotes
    4. Resto de segmentos

d. Unidades productivas
    i. UP_Simple_Efecto
        1. Plantilla de desconexión – STOP
        2. Plantilla de conexión
        3. Plantilla de congelación
        4. Entrada en estado de emergencia
        5. Llamada a la maniobra de expansión
        6. Llamada a la maniobra de compresión
        7. Resto de segmentos
    ii. UP_Robot
        1. Plantilla de desconexión – STOP
        2. Plantilla de conexión
        3. Plantilla de congelación
        4. Entrada en estado de emergencia
        5. Llamada a las maniobras
        6. Resto de segmentos
    iii. UP_CNC

1. Plantilla de desconexión – STOP
2. Plantilla de conexión
3. Plantilla de congelación
4. Entrada en estado de emergencia
5. Llamada a las maniobras
6. Resto de segmentos

e. Maniobras

i. SFC_Programa

1. Plantilla de desconexión – Error asíncrono
2. Error síncrono por tiempo
3. Error síncrono por mala configuración inicial
4. Plantilla de desconexión – Fin de maniobra
5. Tratamiento etapas fugaces
6. Resto de segmentos

ii. SFC_Cilindro

f. Señales analógicas

i. alturaPalets

1. Plantilla de desconexión
2. Llamada a la función SCALE
3. Error de límites
4. Comparación

ii. UP_Palets Vacíos

1. Plantilla de desconexión – Error asíncrono
2. Plantilla de desconexión – STOP
3. Plantilla de conexión
4. Plantilla de congelación
5. Salida de emergencia
6. Llamada a alturaPalets

7. Resto de segmentos

g. Test unitarios

i. UT_SFC_Programa

1. Tabla de símbolos
2. Llamada al bloque
3. Llamada a SFC_Programa
4. Resto de segmentos

ii. UT_SFC_Cilindro

iii. UT_UP_Simple_Efecto

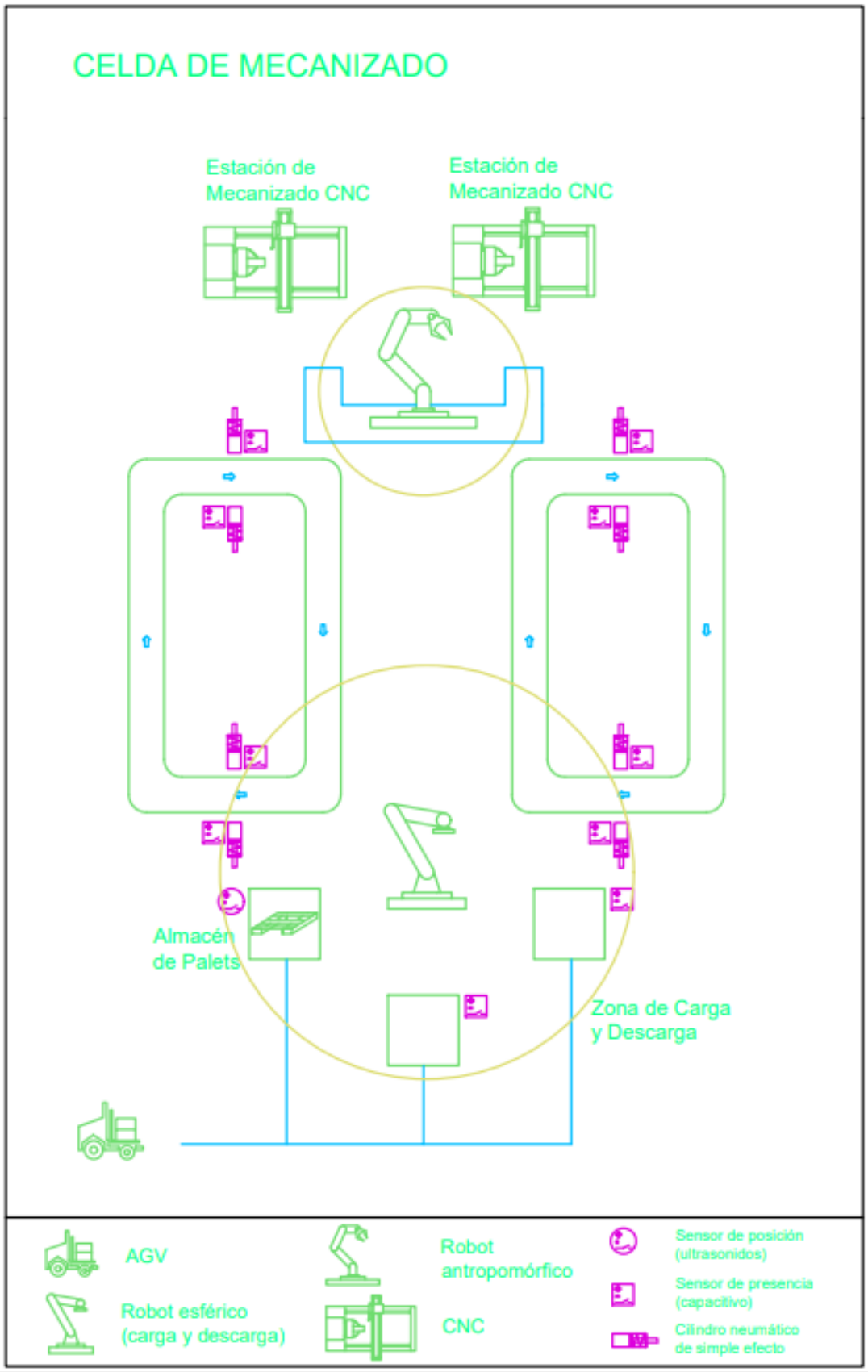
iv. UT_UP_Robot

v. UT_UP_CNC

vi. UT_UP_Palets_Vacios

LAYOUT ESQUEMÁTICO DE LA CELDA

A continuación se presenta el layout de la celda de mecanizado.



Descripción

Los AGV llegan cargan y descargan los palets en los buffers de carga y descarga. Un robot esférico se encarga de trasladar las cajas contenidas en los palets a una plataforma, desde la cual un cilindro neumático de simple efecto las empujará dentro de la cinta. La cinta sirve como buffer y transporta los productos hasta el otro lado de la celda, donde un robot antropomórfico se encarga de colocar las cajas en la CNC. La CNC mecaniza el contenido de las cajas. Cuando finaliza, envía una señal al robot antropomórfico para que traslade el producto mecanizado a una plataforma, desde la cual un cilindro neumático de simple efecto se encargará de empujar las cajas de vuelta a la cinta. Finalmente, el robot esférico descarga las cajas de la cinta a un palet situado en el buffer de descarga. Los AGVs se llevarán dicho palet a la siguiente estación.

Los palets vacíos que no son necesarios en el proceso se acumulan en un buffer de palets vacíos, y cuando alcanzan una determinada altura se manda una señal a los AGVs para que los recojan y los lleven al almacén o a otra estación donde sean necesarios.

CATÁLOGO DE COMUNICACIONES

En esta sección se expone el catálogo de comunicaciones. Como hay muchas señales se van a analizar las unidades productivas por separado.

METACONTROL

Señales del proceso al PLC:

- On: señal digital que indica que la celda pasa de estar parada a estado de marcha.
- Lote_pequeño: señal digital que indica que los lotes a mecanizar son pequeños.
- Lote_mediano: señal digital que indica que los lotes a mecanizar son medianos.
- Lote_grande: señal digital que indica que los lotes a mecanizar son grandes.
- Mcong: señal digital que indica que el modo de operación es con congelación.
- Cong: señal digital que indica que se congele la celda.
- Emer: señal digital que indica la entrada en estado de emergencia. Habrá señales globales y locales.
- Rearme: señal digital que indica la salida del estado de emergencia. Habrá señales globales y locales.

UP_ROBOT

Señales de la controladora del robot al PLC:

- Errores síncronos: señal digital que indicará que el robot está en error síncrono. Habrá 4 señales de este tipo (una por cada posible error síncrono):
    - Error tiempo carga
    - Error tiempo descarga
    - Error configuración carga
    - Error configuración descarga
- Error asíncrono: señal digital que indicará que el robot se encuentra en un estado de error asíncrono. No se distingue entre error local y global porque la entrada en este estado se indica desde fuera.
- Standby: señal digital indicando que el robot se encuentra en estado de standby.
- Ocupación: señal digital indicando que el robot se encuentra ocupado (en medio de una maniobra).
- Home: señal digital indicando que el robot se encuentra en posición fetal.

Señales del proceso al PLC:

- dc: señal digital que indica que el robot debe realizar una maniobra de carga.
- fin_p: señal digital que indica que el robot debe realizar una maniobra de descarga.

- Emer: señal digital que indica que el robot tiene que entrar en estado de emergencia. Habrá 2:
    - Emer_global
    - Emer_local
- Rearme: señal digital que indica que el robot tiene que salir del estado de emergencia. Habrá 2:
    - Rearme_local
    - Rearme_global
- Congelación: señal digital que indica que el robot tiene que parar su actividad manteniendo el estado en el que se encuentre.
- Reset: señal digital que desconecta el robot.

Señales del PLC a la controladora del robot:

- Cargar: señal digital que ordena ejecutar la maniobra de carga.
- Descargar: señal digital que ordena ejecutar la maniobra de descarga.

UP_CNC

Señales de la controladora del CNC al PLC:

- Errores síncronos: igual que en UP_Robot. Habrá 8 (una por cada posible error).
- Errores asíncronos: igual que en UP_Robot.
- Standby: igual que UP_Robot.
- Ocupación: igual que UP_Robot.
- Home: igual que UP_Robot.

Señales del proceso al PLC:

- Lápiz: señal digital que indica que el CNC va a fabricar lápices.
- Sacapuntas: señal digital que indica que el CNC va a fabricar sacapuntas.
- Escuadra: señal digital que indica que el CNC va a fabricar escuadras.
- Cartabón: señal digital que indica que el CNC va a fabricar cartabones.
- Emer: igual que UP_Robot.
- Rearme: igual que UP_Robot.
- Congelación: igual que UP_Robot.
- Reset: igual que UP_Robot.

Señales del PLC a la controladora del CNC:

- Cnc_lapiz: señal digital que ordena mecanizar lápices.

- Cnc_sacapuntas: señal digital que ordena mecanizar sacapuntas.
- Cnc_escuadra: señal digital que ordena mecanizar escuadras.
- Cnc_cartabon: señal digital que ordena mecanizar cartabones.

UP_SIMPLE_EFECTO

Señales de la controladora del cilindro al PLC:

- Errores síncronos: igual que en UP_Robot. Habrá 4 (una por cada posible estado de defecto).
- Errores asíncronos: igual que en UP_Robot.
- Standby: igual que en UP_Robot.
- Ocupación: igual que en UP_Robot.
- Scomp: señal digital que indica que el cilindro está comprimido.
- Sexp: señal digital que indica que el cilindro está expandido.

Señales del proceso al PLC:

- Push: señal digital que indica que el cilindro tiene que expandirse.
- Emer: igual que UP_Robot.
- Rearme: igual que UP_Robot.
- Congelación: igual que UP_Robot.
- Reset: igual que UP_Robot.

Señales del PLC a la controladora cilindro:

- Expandir: señal digital que indica que el cilindro tiene que expandirse.
- Comprimir: señal digital que indica que el cilindro tiene que comprimirse.

UP_PALETS_VACIOS

Señales de la controladora de la unidad al PLC:

- Errores asíncronos: igual que UP_Robot.
- Standby: igual que UP_Robot.
- H: señal analógica que indica la altura (magnitud muestreada) de la pila de palets vacíos.

Señales del proceso al PLC:

- H>Hmín: señal digital que indica que la altura de la pila de palets vacíos es mayor que la altura mínima para que se puedan recoger.
- H>Hmáx: señal digital que indica que la altura de la pila de palets vacíos ha alcanzado la altura de saturación.

- H<Hmin: señal digital que indica que la altura de la pila de palets vacíos es menor que la altura mínima para que se puedan recoger, por lo que la unidad puede volver al estado de standby.
- Emer: igual que UP_Robot.
- Rearme: igual que UP_Robot.
- Congelación: igual que UP_Robot.
- Reset: igual que UP_Robot.

Señales del PLC a los AGVs (las señales para recoger la pila no se mandan a la unidad, evidentemente):

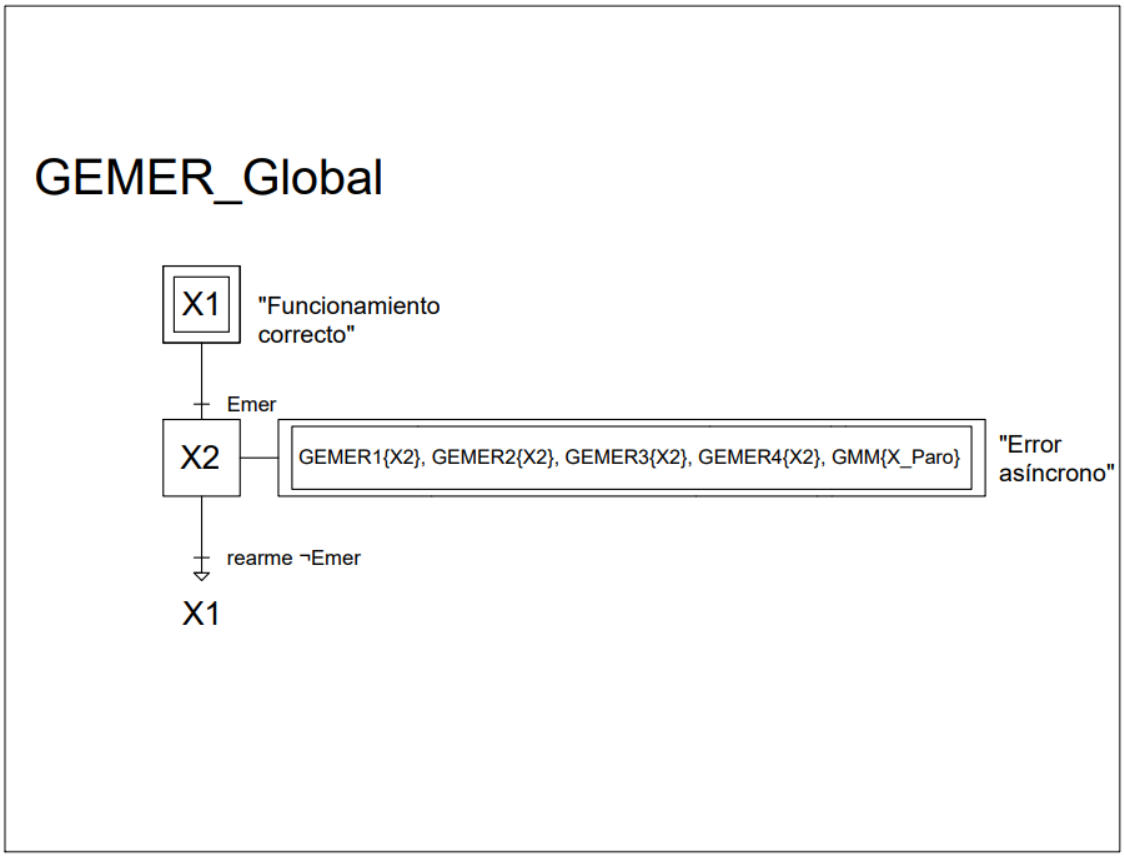
- Recoger pila: señal digital que ordena a los AGVs a recoger la pila.
- Pila saturada: señal digital que ordena a los AGVs a recoger la pila (mayor prioridad).

GRAFCET NIVEL 2

A continuación, se presenta el Grafcet de nivel 2 de la celda de mecanizado. En cada Grafcet se incluye una explicación del funcionamiento del mismo.

GEMER_GLOBAL

Controla la emergencia global. Se inicia con el inicio del programa en un estado de funcionamiento correcto. Si se pulsa la seta de emergencia, pasa a un estado de defecto global. En dicho estado, todas las unidades productivas del proceso deben cesar su funcionamiento. Para ello, el grafcet de emergencia global fuerza a emergencia a los grafcet de emergencia local.

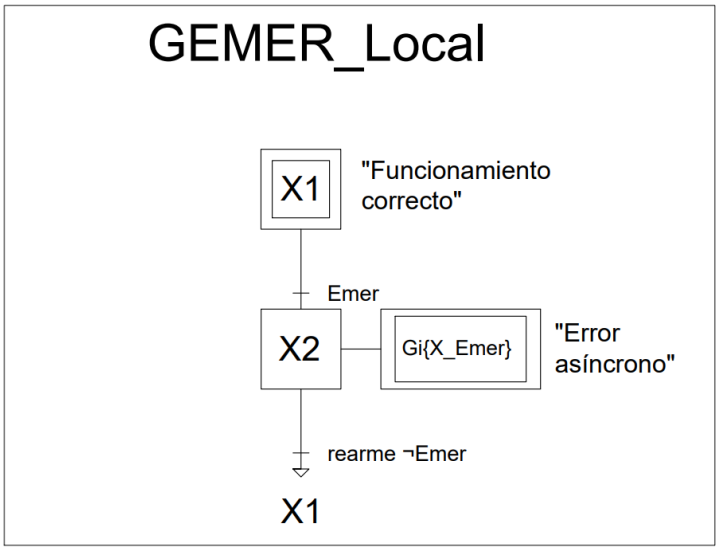


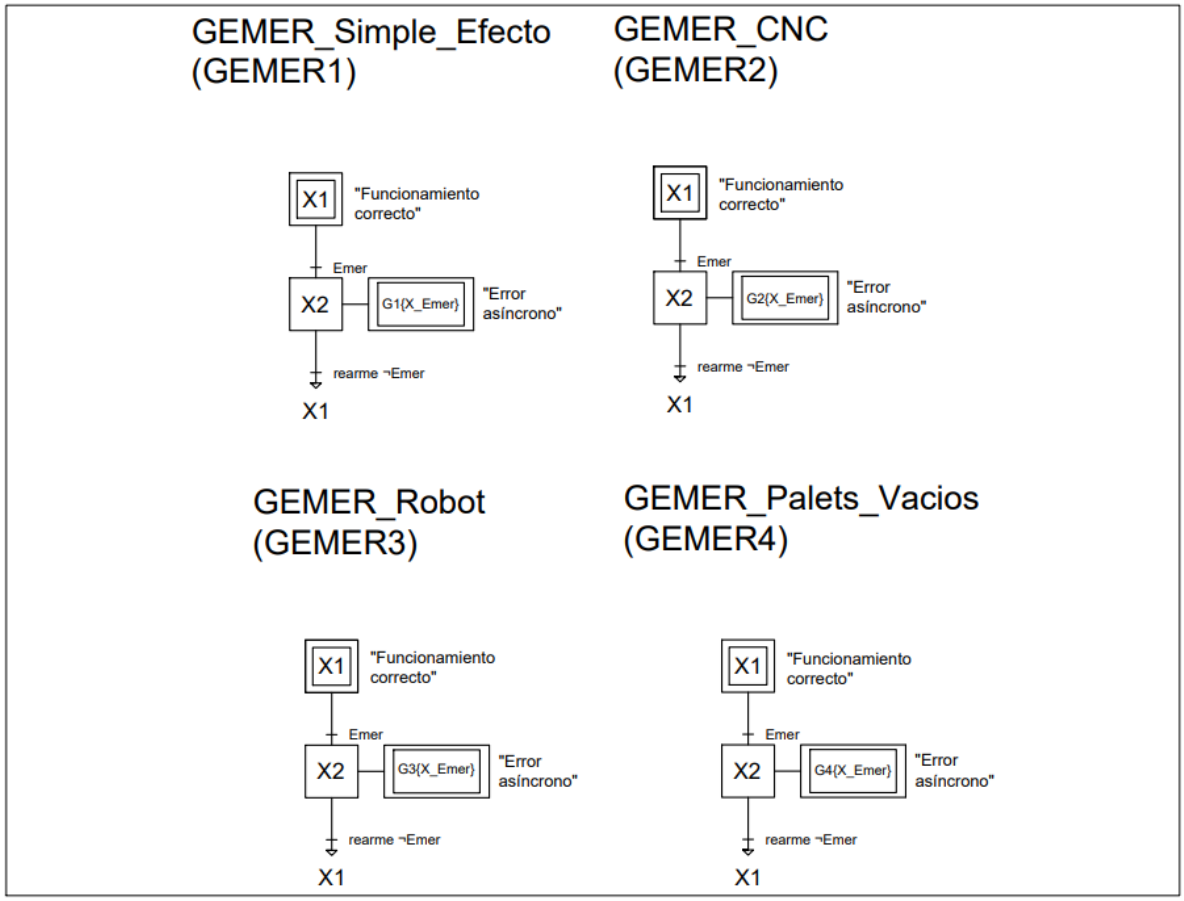
GEMER_LOCAL

Controla la emergencia local. Se inicia al inicio del programa en un estado de funcionamiento correcto. Si se pulsa la seta de emergencia local, fuerza a la unidad productiva que controla al estado de defecto. Es controlado por el GEMER_Global.

En caso de entrar en defecto local, la unidad productiva correspondiente se irá al estado de emergencia, pero las demás seguirán en funcionamiento. Se rearman con una señal de rearme local.

Existe un grafcet de emergencia local por cada unidad productiva.





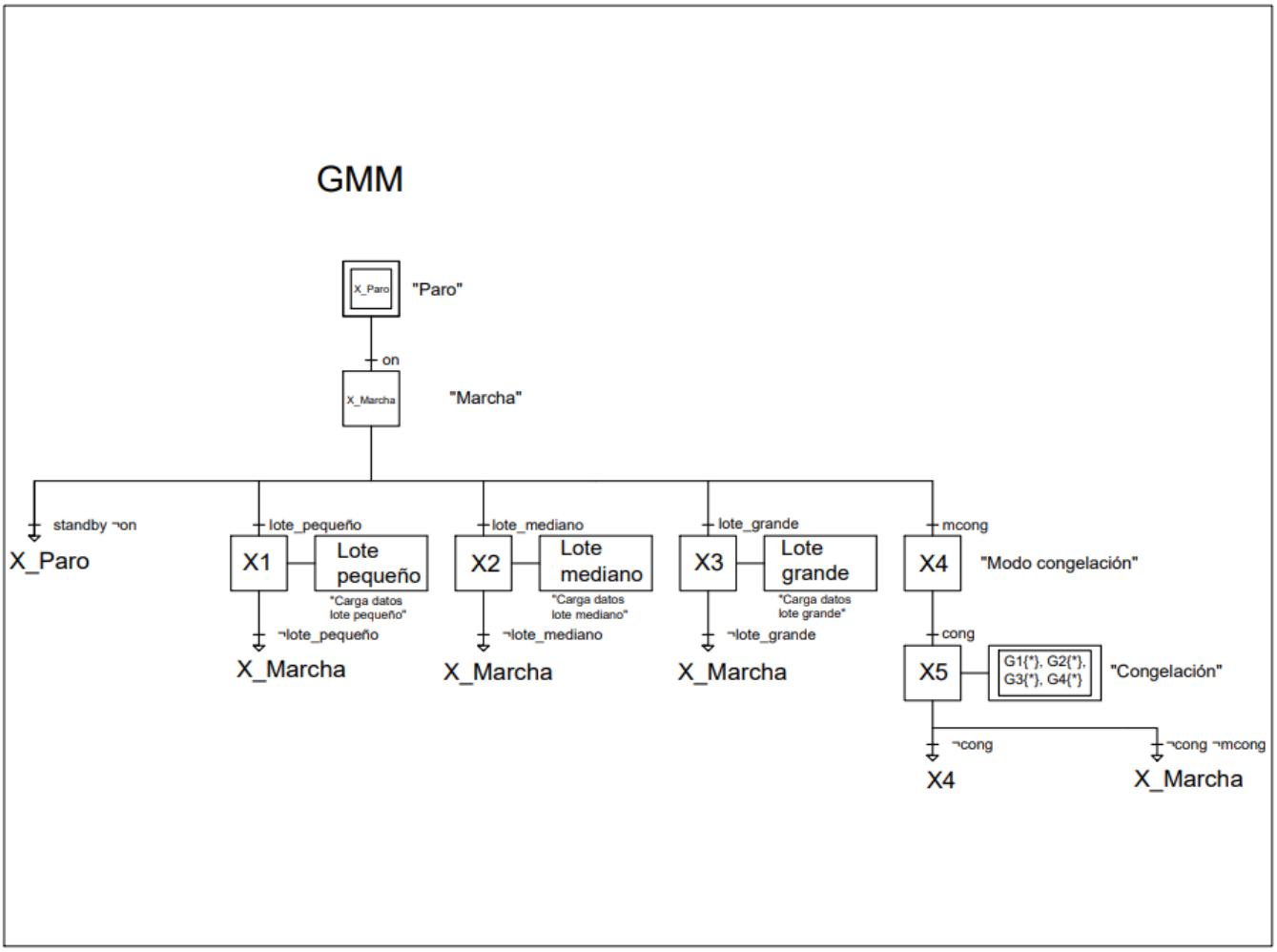
GMM

Controla los modos de marcha. Se inicia en el estado de parada al iniciar el programa. Al activar la señal “on”, inicia todas las unidades productivas. Si todas las unidades productivas están en standby y se desactiva la señal “on”, vuelve a la parada (desconexión de las unidades productivas).

En función de la cantidad de material que vaya en las cajas, se elige un tamaño de lote. Los lotes pueden ser: pequeño, mediano, grande. Cuando se elige un lote, se transfieren los datos de producción asociados a dicho lote a los datos de producción globales (p.ej. el tiempo de procesamiento de un lote de sacapuntas grande es de 30s mientras que para un lote pequeño es de 10s).

Incluye un modo de congelación. Cuando se está en este modo, las unidades productivas funcionan de la forma habitual con los valores de lote seleccionados hasta que se activa la señal de congelación. En ese momento, todas las unidades productivas se detienen en el estado en el que estén. Al descongelar, retoman el funcionamiento normal desde el punto en el que estaban.

IMPORTANTE: no se usan los temporizadores modernos, por lo que después de descongelar, estos se resetean.



UP_SIMPLE_EFECTO (G1)

Unidad productive encargada de controlar los cilindros de simple efecto para la carga de cajas en la cinta.

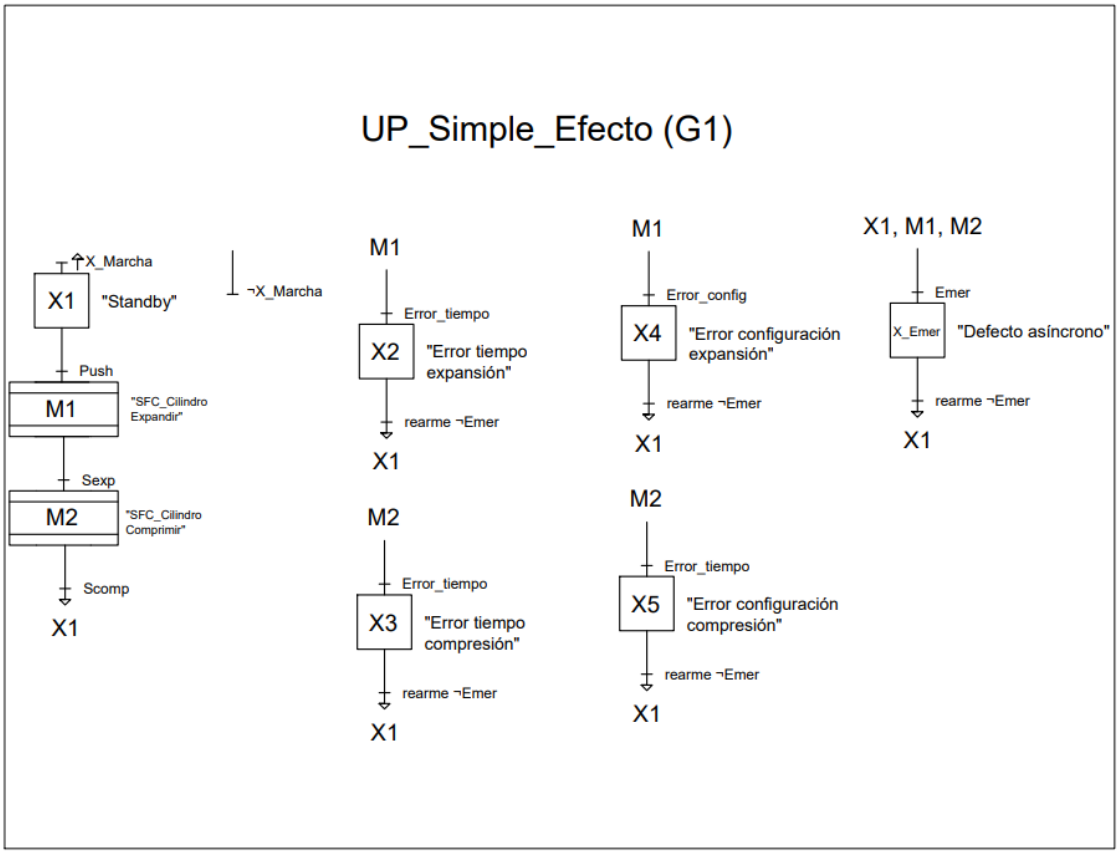
Se inicia cuando se llega al estado de marcha en GMM. Comienza en un estado de standby. Cuando se activa la señal “push”, se produce la expansión del cilindro del émbolo que empuja la caja dentro de la cinta. Cuando llega al final de carrera (Sexp) se activa la maniobra de compresión. Una vez comprimido, vuelve al estado de standby.

La unidad productiva usa maniobras del tipo SFC_Cilindro, las cuales controlan un actuador, incluyendo errores por tiempo y por mala configuración inicial.

Se incluyen estados de error síncrono: uno por cada posible defecto (mala configuración inicial o tiempo límite alcanzado). Además, se distingue en qué maniobra se produjo el error para tener la máxima información posible en cuanto al mismo (no es lo mismo que el error se produzca en la expansión que en la compresión: p.ej. podría deberse a que un sensor está averiado y de esta forma podemos saber cuál es con mayor facilidad).

Cuando se activa la señal de emergencia (ya sea global o local) se va a un estado de defecto asíncrono.

Cuando el estado de marcha de GMM deja de estar activo, se desconecta la unidad productiva.

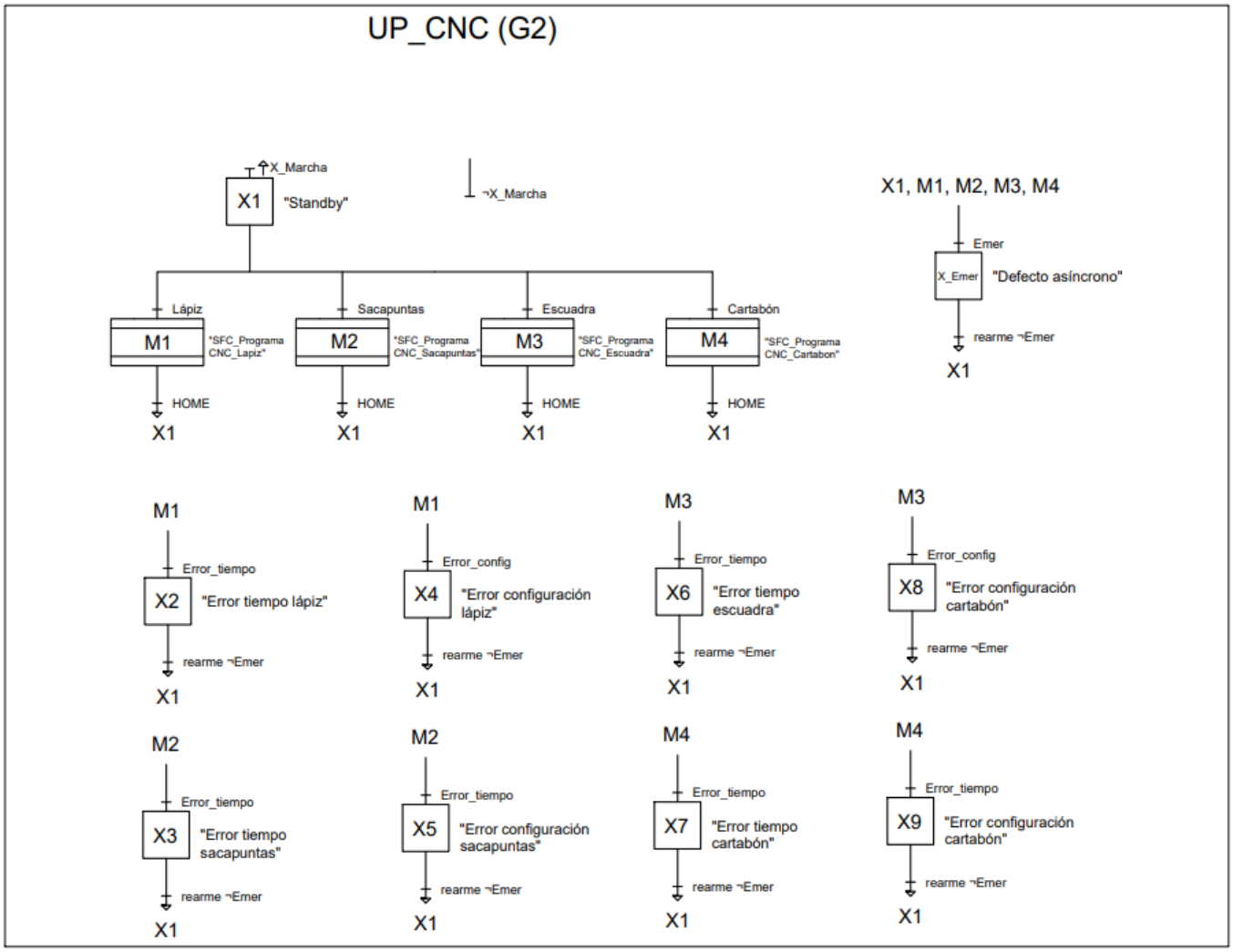


UP_CNC (G2)

Controla la ejecución de los programas de las CNC. Se inicia con el flanco positivo del estado de marcha de GMM.

La filosofía usada es la misma que en el cilindro de simple efecto:

- Maniobras: usa maniobras (SFC_Programa) para la fabricación de los diferentes productos.
- Errores síncronos: un estado de error por cada posible defecto síncrono en cada maniobra.
- Errores asíncronos: entrada en emergencia en caso de emergencia global o local.
- Desconexión: con nivel bajo del estado de marcha del GMM.

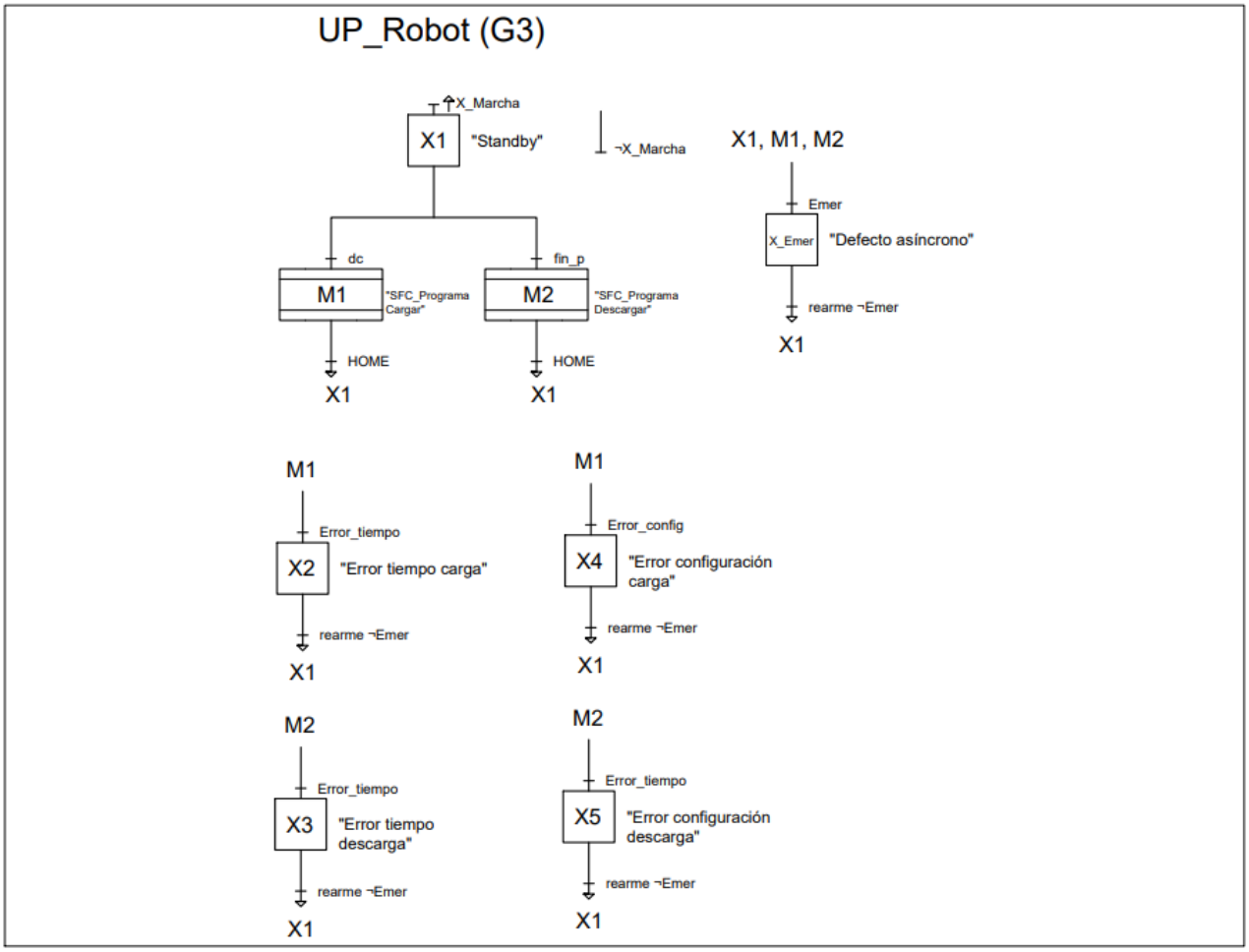


UP_ROBOT (G3)

Controla el robot de carga y descarga de palets. Se inicia con el flanco positivo del estado de marcha de GMM.

La filosofía usada es la misma que en el cilindro de simple efecto:

- Maniobras: usa maniobras (SFC_Programa) para la carga y la descarga.
- Errores síncronos: un estado de error por cada posible defecto síncrono en cada maniobra.
- Errores asíncronos: entrada en emergencia en caso de emergencia global o local.
- Desconexión: con nivel bajo del estado de marcha del GMM.



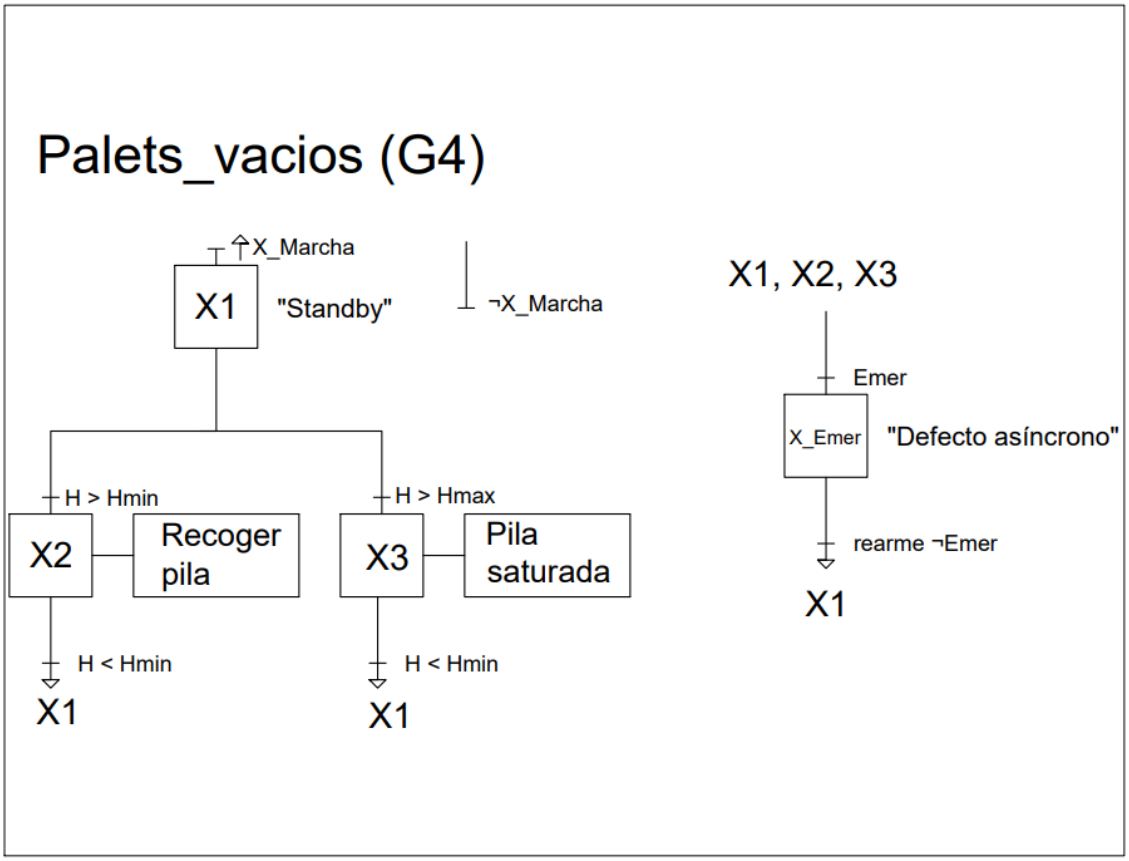
UP_PALETS_VACIOS (G4)

Controla la pila de palets vacíos mediante señales analógicas. Se inicia con el flanco positivo del estado de marcha de GMM. La desconexión es por nivel bajo de dicha señal.

Comienza en estado de standby. El robot de carga y descarga va cargando los palets que no son necesarios en la pila de palets vacíos. Si se sobrepasa el nivel mínimo (Hmin) se envía una señal a los AGVs que se encargarán de recoger la pila (todavía pueden situarse más palets sobre la pila). Si se sobrepasa el nivel máximo (Hmax), la pila estará saturada y no se podrán situar más palets sobre la misma. Cuando los AGVs recogen los palets y la pila tiene una altura menor que Hmin, se vuelve al estado de standby.

Se incluye un estado de defecto asíncrono para defectos globales y locales.

No hay defectos síncronos, ya que solo se usan sensores que están midiendo la altura de la pila de palets vacíos constantemente.



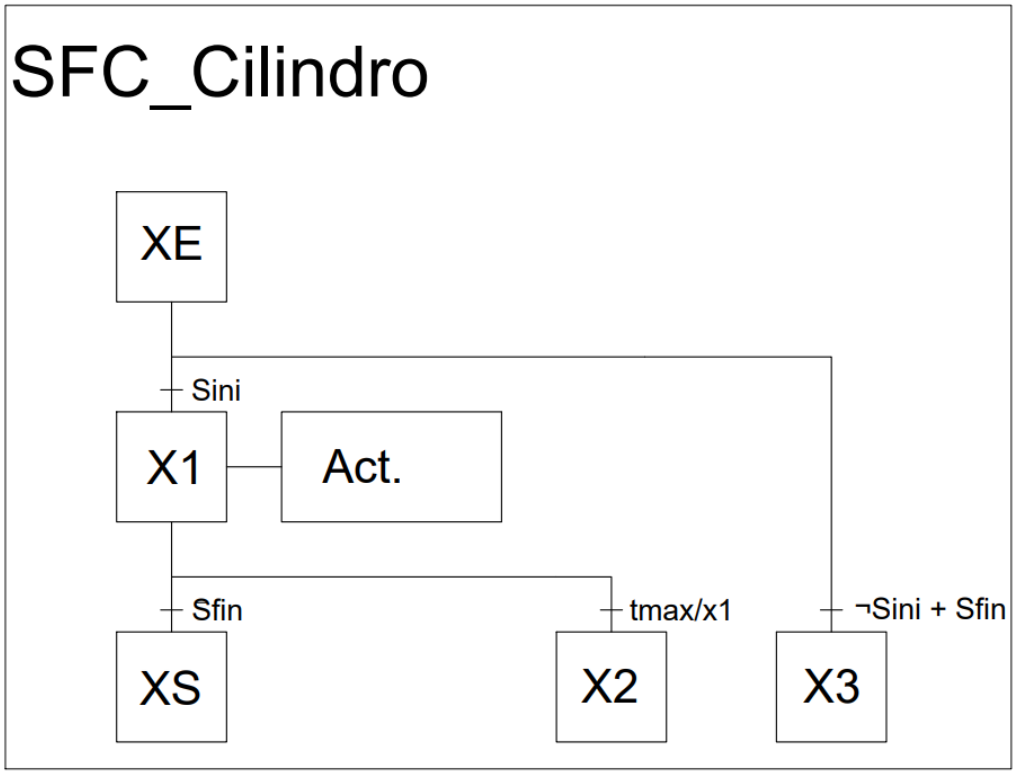
SFC_CILINDRO

Maniobras para la unidad productiva “UP_Simple_Efecto”.

Comienza en el estado XE (fugaz). Si la configuración inicial del cilindro es incorrecta, va a un estado de error síncrono por mala configuración inicial. En caso contrario, pasa al estado de ejecución de la maniobra (activa el actuador). Si el tiempo de maniobra supera un límite (tmax), irá a un estado de defecto por error de tiempo límite. En caso contrario, la maniobra finaliza con éxito.

Es necesario enfatizar que en el caso de la compresión no se le pasa ningún actuador a la maniobra porque se usa un cilindro de simple efecto y la compresión es por la no actuación sobre la expansión. A pesar de esto, se usa también la estructura de la maniobra para la compresión a modo de encapsulamiento (puede haber errores síncronos).

Esta maniobra podría usarse también para la expansión y compresión de cilindros de doble efecto, por ejemplo. En dicho caso, sí sería necesario pasar el actuador de compresión como parámetro.

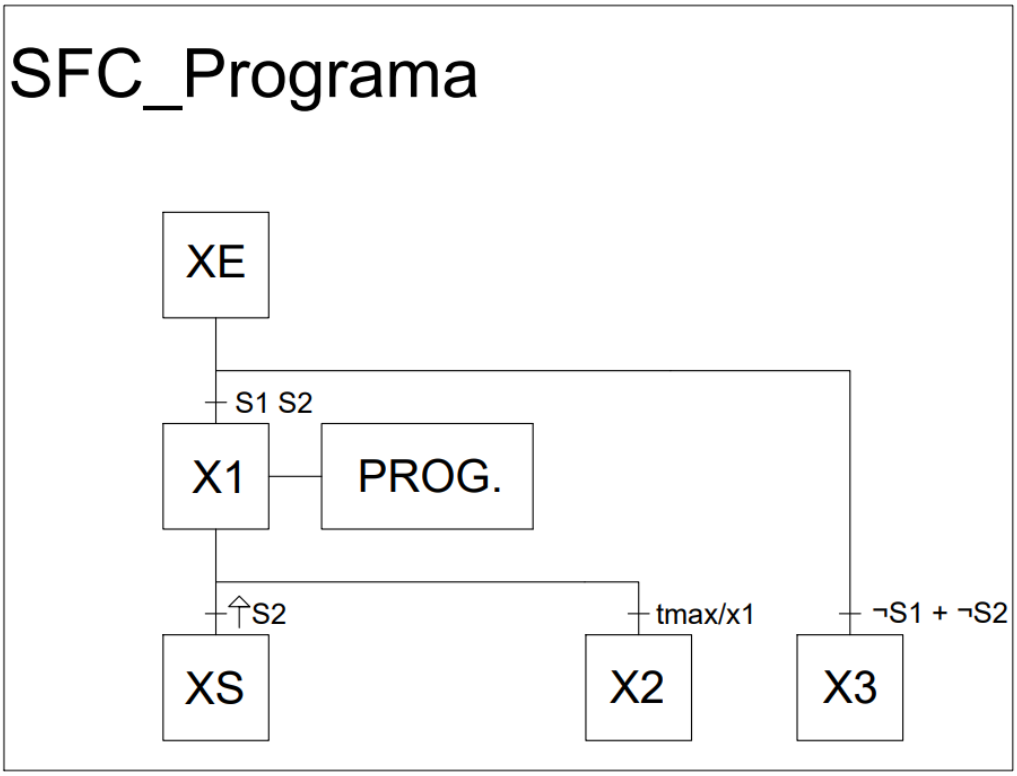


SFC_PROGRAMA

Maniobra usada por la CNC y el robot de carga y descarga para utilizar sus programas.

¿Por qué tener SFC_Programa y SFC_Cilindro? Aunque las maniobras son parecidas, no son iguales. Las receptividades de las maniobras son diferentes:

- Xe X1: es necesario tener los 2 sensores activos para no tener un error por configuración inicial. Para iniciar la maniobra, la unidad productiva debe estar en home y tiene que tener la señal que lo activa a nivel alto.
- X1 Xs: las unidades productivas inician y terminan el programa en home, por lo que no se pueden usar niveles para esta receptividad (si no se produciría una etapa fugaz). Durante la ejecución del programa, habrá un momento que la señal home esté a nivel bajo, por lo que se usa el flanco positivo de dicha señal para indicar el fin de la maniobra.



PROGRAMACIÓN

INIT (OB100)

Inicializa los estados globales (solo son globales los estados de emergencia global).

```
SET
S     "x_marcha"
R     "x_emer"
```

MAIN (OB1)

Se encarga de llamar a las unidades productivas, integrando los modos de marcha y las emergencias. Llama a los bloques en el siguiente orden:

- Emergencia global
- Emergencias locales
- Rearmes
- Modos de marcha
- Unidades productivas

Las salidas de los bloques de emergencia y modos de marcha determinarán las entradas “emer”, “rearme”, “cong” y “start” de las unidades productivas.

GEMER_GLOBAL

Llama a la emergencia global.

```
1     CALL  "GEMER_Global"
```

GEMER_LOCAL

Si no hay emergencia global (prioritaria frente a emergencias locales), se llama a las emergencias locales:

```
1     UN    "x_emer"
2
3     SPBN  _emer_robot1
4
5     CALL  "GEMER_Local", "GEMER_Robot1_DB"
6        emer     :="emer_local_robot1"
7        rearme   :="rearme_local_robot1"
8        bit_emer :="robot1_emer"
9
10 _emer_robot1 : NOP 0
```

```
1     UN    "x_emer"
2
3     SPBN  _emer_robot2
4
5     CALL  "GEMER_Local", "GEMER_Robot2_DB"
6        emer     :="emer_local_robot2"
7        rearme   :="rearme_local_robot2"
8        bit_emer :="robot2_emer"
9
10 _emer_robot2 : NOP 0
```

```
UN    "x_emer"

SPBN  _emer_cnc1

CALL  "GEMER_Local", "GEMER_CNC1_DB"
   emer     :="emer_local_cnc1"
   rearme   :="rearme_local_cnc1"
   bit_emer :="cnc1_emer"

_emer_cnc1 : NOP 0
```

```
UN    "x_emer"

SPBN  _emer_cnc2

CALL  "GEMER_Local", "GEMER_CNC2_DB"
   emer     :="emer_local_cnc2"
   rearme   :="rearme_local_cnc2"
   bit_emer :="cnc2_emer"

_emer_cnc2 : NOP 0
```

```
      UN    "x_emer"

      SPBN  _emer_simple_efectol

      CALL  "GEMER_Local", "GEMER_Simple_Efectol_DB"
         emer     :="emer_local_simple_efectol"
         rearme   :="rearme_local_simple_efectol"
         bit_emer :="sefl_emer"

_emer_simple_efectol : NOP 0
```

```
      UN    "x_emer"

      SPBN  _palets_vaciosl

      CALL  "GEMER_Local", "GEMER_Local_Palets_Vacios_DB"
         emer     :="emer_local_palets_vacios"
         rearme   :="rearme_local_palets_vacios"
         bit_emer :="pa_emer"

_palets_vaciosl : NOP 0
```

REARMES

Mira si hay rearmes (si un robot está en emergencia global no podrá rearmarse localmente porque antes se estará ejecutando la plantilla de desconexión por error asíncrono).

Network 8: Rearme robot1

Network 9: Rearme robot2

Network 10: Rearme cnc1

Network 11: Rearme cnc2

Network 12: Rearme simple efecto1

Network 13: Rearme palets vacíos

GMM

Si no hay emergencia global, se ejecutan los modos de marcha. En caso contrario, se ejecutarán desde GEMER_Global, que activará la plantilla de desconexión:

```
      UN    "x_emer"

      SPBN  _gmm_emer

      CALL  "GMM", "GMM_DB"
         on              :="gmm_on"
         lote_pequenio   :="gmm_lote_pequenio"
         lote_mediano    :="gmm_lote_mediano"
         lote_grande     :="gmm_lote_grande"
         mcong           :="gmm_mcong"
         cong            :="gmm_cong"
         emer            :=FALSE
         bit_paro        :="gmm_bit_paro"
         bit_marcha      :="gmm_bit_marcha"
         bit_congelacion :="gmm_bit_congelacion"

_gmm_emer : NOP 0
```

UNIDADES PRODUCTIVAS

Se llama a las unidades productivas con los bits de control determinados por las anteriores llamadas a los bloques de emergencia y modos de marcha:

```
CALL "UP_Robot", "UP_Robot1_DB"
   start                    :="gmm_bit_marcha"
   emer                     :="robot1_emer"
   rearme                   :="robot1_rearme"
   cong                     :="gmm_bit_congelacion"
   dc                       :="robot1_dc"
   fin_p                    :="robot1_fin_p"
   home                     :="robot1_home"
   tiempo_max               :="datosProduccion".tmax_carga_descarga
   temp_max                 :="robot1_temp"
   cargar                   :="robot1_cargar"
   descargar                :="robot1_descargar"
   bit_standby              :="robot1_b_standby"
   bit_ocupacion            :="robot1_b_ocupacion"
   bit_error_tiempo_carga   :="robot1_bdt_carga"
   bit_error_tiempo_descarga :="robot1_bdt_descarga"
   bit_error_config_carga   :="robot1_bdc_carga"
   bit_error_config_descarga :="robot1_bdc_descarga"
   bit_emer                 :="robot1_b_emer"
```

```
CALL "UP_Robot", "UP_Robot2_DB"
   start                    :="gmm_bit_marcha"
   emer                     :="robot2_emer"
   rearme                   :="robot2_rearme"
   cong                     :="gmm_bit_congelacion"
   dc                       :="robot2_dc"
   fin_p                    :="robot2_fin_p"
   home                     :="robot2_home"
   tiempo_max               :="datosProduccion".tmax_carga_descarga
   temp_max                 :="robot2_temp"
   cargar                   :="robot2_cargar"
   descargar                :="robot2_descargar"
   bit_standby              :="robot2_b_standby"
   bit_ocupacion            :="robot2_b_ocupacion"
   bit_error_tiempo_carga   :="robot2_bdt_carga"
   bit_error_tiempo_descarga :="robot2_bdt_descarga"
   bit_error_config_carga   :="robot2_bdc_carga"
   bit_error_config_descarga :="robot2_bdc_descarga"
   bit_emer                 :="robot2_b_emer"
```

```
CALL "UP_CNC", "UP_CNC1_DB"
   start                    :="gmm_bit_marcha"
   emer                     :="cnc1_emer"
   rearme                   :="cnc1_rearme"
   cong                     :="gmm_bit_congelacion"
   lapiz                    :="cnc1_lapiz"
   sacapuntas               :="cnc1_sacapuntas"
   escuadra                 :="cnc1_escuadra"
   cartabon                 :="cnc1_cartabon"
   home                     :="cnc1_home"
   tmax_cnc_lapiz           :="datosProduccion".tmax_cnc_lapiz
   tmax_cnc_sacapuntas      :="datosProduccion".tmax_cnc_sacapuntas
   tmax_cnc_escuadra        :="datosProduccion".tmax_cnc_escuadra
   tmax_cnc_cartabon        :="datosProduccion".tmax_cnc_cartabon
   temp_max                 :="cnc1_temp"
   cnc_lapiz                :="cnc1_cnc_lapiz"
   cnc_sacapuntas           :="cnc1_cnc_sacapuntas"
   cnc_escuadra             :="cnc1_cnc_escuadra"
   cnc_cartabon             :="cnc1_cnc_cartabon"
   bit_standby              :="cnc1_b_standby"
   bit_ocupacion            :="cnc1_b_ocupacion"
   bit_error_tiempo_lapiz   :="cnc1_bdt_lapiz"
   bit_error_tiempo_sacapuntas :="cnc1_bdt_sacapuntas"
   bit_error_tiempo_cartabon :="cnc1_bdt_cartabon"
   bit_error_config_lapiz   :="cnc1_bdc_lapiz"
   bit_error_config_sacapuntas :="cnc1_bdc_sacapuntas"
   bit_error_config_escuadra :="cnc1_bdc_escuadra"
   bit_error_config_cartabon :="cnc1_bdc_cartabon"
   bit_emer                 :="cnc1_b_emer"
```

```
CALL "UP_CNC", "UP_CNC2_DB"
   start                    :="gmm_bit_marcha"
   emer                     :="cnc2_emer"
   rearme                   :="cnc2_rearme"
   cong                     :="gmm_bit_congelacion"
   lapiz                    :="cnc2_lapiz"
   sacapuntas               :="cnc2_sacapuntas"
   escuadra                 :="cnc2_escuadra"
   cartabon                 :="cnc2_cartabon"
   home                     :="cnc2_home"
   tmax_cnc_lapiz           :="datosProduccion".tmax_cnc_lapiz
   tmax_cnc_sacapuntas      :="datosProduccion".tmax_cnc_sacapuntas
   tmax_cnc_escuadra        :="datosProduccion".tmax_cnc_escuadra
   tmax_cnc_cartabon        :="datosProduccion".tmax_cnc_cartabon
   temp_max                 :="cnc2_temp"
   cnc_lapiz                :="cnc2_cnc_lapiz"
   cnc_sacapuntas           :="cnc2_cnc_sacapuntas"
   cnc_escuadra             :="cnc2_cnc_escuadra"
   cnc_cartabon             :="cnc2_cnc_cartabon"
   bit_standby              :="cnc2_b_standby"
   bit_ocupacion            :="cnc2_b_ocupacion"
   bit_error_tiempo_lapiz   :="cnc2_bdt_lapiz"
   bit_error_tiempo_sacapuntas :="cnc2_bdt_sacapuntas"
   bit_error_tiempo_escuadra :="cnc2_bdt_escuadra"
   bit_error_tiempo_cartabon :="cnc2_bdt_cartabon"
   bit_error_config_sacapuntas :="cnc2_bdc_sacapuntas"
   bit_error_config_escuadra :="cnc2_bdc_escuadra"
   bit_error_config_cartabon :="cnc2_bdc_cartabon"
   bit_emer                 :="cnc2_b_emer"
```

```
CALL "UP_Simple_Efecto", "UP_Simple_Efecto1_DB"
   start                    :="gmm_bit_marcha"
   emer                     :="sefl_emer"
   cong                     :="sefl_rearme"
   rearme                   :="gmm_bit_congelacion"
   scomp                    :="sefl_scomp"
   sexp                     :="sefl_sexp"
   push                     :="sefl_push"
   tiempo_max               :="datosProduccion".tmax_simple_efecto
   temp_max                 :="sefl_temp"
   expandir                 :="sefl_expandir"
   bit_standby              :="sefl_b_standby"
   bit_ocupacion            :="sefl_b_ocupacion"
   bit_error_tiempo_expansion :="sefl_bdt_expansion"
   bit_error_tiempo_compresion :="sefl_bdt_compresion"
   bit_error_config_expansion :="sefl_bdc_expansion"
   bit_error_config_compresion :="sefl_bdc_expansion"
   bit_emer                 :="sefl_b_emer"
```

```
CALL "UP_Palets_Vacios", "UP_Palets_Vacios_DB"
   start        :="gmm_bit_marcha"
   emer         :="pa_emer"
   rearme       :="pa_rearme"
   cong         :="gmm_bit_congelacion"
   hraw         :="pa_hraw"
   LO_LIM       :="datosProduccion".LO_LIM
   HI_LIM       :="datosProduccion".HI_LIM
   hmin         :="datosProduccion".hmin
   hmax         :="datosProduccion".hmax
   recoger_pila :="pa_recoger_pila"
   pila_saturada :="pa_pila_saturada"
   error_medida :="pa_error_medida"
   h_real       :="pa_hreal"
   h>hmin       :="pa_h>hmin"
   h>hmax       :="pa_h>hmax"
   h<hmin       :="pa_h<hmin"
```

METACONTROL

GEMER_GLOBAL

Se encarga de coordinar la emergencia global en toda la celda, llamando a los bloques de emergencia locales. También fuerza a GMM. Usa variables absolutas (FC sin parámetros y que controla estados globales). Se inicia directamente en estado de marcha con el OB100.

TRANSICIONES DE ESTADOS

Segmentos encargados de la transición entre estados. Cabe destacar que se usan variables absolutas. Al rearmarse llama a los bloques de emergencia globales para que se rearmen.

IMPORTANTE: no fuerza directamente a ninguna unidad productiva, si no a los bloques que las controlan.

Network 2: x_emer -> x_marcha

Comment

```
1      U     "x_emer"
2      UN    "emer"
3      U     "rearme"
4      S     "x_marcha"
5      R     "x_emer"
6
7      SPBN  _rearme
8
9      CALL  "GEMER_Local", "GEMER_Robot1_DB"
10        emer     :=FALSE
11        rearme   :=TRUE
12        bit_emer :="robot1_emer"
13
```

Network 1: x_marcha -> x_emer

Comment

```
1      U     "x_marcha"
2      U     "emer"
3      S     "x_emer"
4      R     "x_marcha"
```

```
14     CALL  "GEMER_Local", "GEMER_Robot2_DB"
15        emer     :=FALSE
16        rearme   :=TRUE
17        bit_emer :="robot2_emer"
18
19     CALL  "GEMER_Local", "GEMER_CNC1_DB"
20        emer     :=FALSE
21        rearme   :=TRUE
22        bit_emer :="cnc1_emer"
23
24     CALL  "GEMER_Local", "GEMER_CNC2_DB"
25        emer     :=FALSE
26        rearme   :=TRUE
27        bit_emer :="cnc2_emer"
28
29     CALL  "GEMER_Local", "GEMER_Simple_Efector1_DB"
30        emer     :=FALSE
31        rearme   :=TRUE
32        bit_emer :="sef1_emer"
33
34  _rearme : NOP 0
```

EMERGENCIA

Se encarga de llamar a los bloques a los que fuerza con la señal de emergencia a nivel alto.

```
1   U     "x_emer"
2
3   SPBN  _emergencia
4
5   CALL  "GEMER_Local", "GEMER_Robot1_DB"
6      emer     :=TRUE
7      rearme   :=FALSE
8      bit_emer :="robot1_emer"
9
10  CALL  "GEMER_Local", "GEMER_Robot2_DB"
11     emer     :=TRUE
12     rearme   :=FALSE
13     bit_emer :="robot2_emer"
14
15  CALL  "GEMER_Local", "GEMER_CNC1_DB"
16     emer     :=TRUE
17     rearme   :=FALSE
18     bit_emer :="cnc1_emer"
19
```

```
20  CALL  "GEMER_Local", "GEMER_CNC2_DB"
21     emer     :=TRUE
22     rearme   :=FALSE
23     bit_emer :="cnc2_emer"
24
25  CALL  "GEMER_Local", "GEMER_Simple_Efector1_DB"
26     emer     :=TRUE
27     rearme   :=FALSE
28     bit_emer :="sef1_emer"
29
30  CALL  "GMM", "GMM_DB"
31     on              :=
32     lote_pequenio   :=
33     lote_mediano    :=
34     lote_grande     :=
35     mcong           :=
36     cong            :=
37     emer            :=TRUE
38     bit_paro        :="gmm_bit_paro"
39     bit_marcha      :="gmm_bit_marcha"
40     bit_congelacion :="gmm_bit_congelacion"
41
42  _emergencia : NOP 0
```

GEMER_LOCAL

Se encarga de forzar al estado de emergencia a las unidades productivas a las que controle. Se ha implementado en un FB porque tiene estados internos locales (los estados de GEMER_Global son globales y, además, es requisito del trabajo implementar dicho bloque como FC sin parámetros).

Se reutilizará esta función con cada una de las unidades productivas, con la diferencia de que se llamará cada vez a un bloque de datos diferente.

Presenta la siguiente interfaz:

- Emer: informa de emergencia global o local.
- Rearme: rearme global o local.
- Bit_emer: informa que la función se encuentra en estado de emergencia.
- X_marcha: estado de marcha.
- X_emer: estado de emergencia.
- Bit_trabajo: controla la conexión del bloque.

PLANTILLA DE CONEXIÓN

Solamente se ejecuta una vez (la función no se desconecta) con el inicio del programa (OB100).

```
SET
FP    #bit_trabajo
S     #x_marcha
R     #x_emer
```

PLANTILLA DE FORZADO - EMERGENCIA

En caso de emergencia (local o global), se establece el estado de emergencia y se informa al exterior de dicho estado mediante “bit_emer”.

```
1    U    #emer
2    S    #x_emer
3    R    #x_marcha
4    =    #bit_emer
5    BEB
```

RESTO DE SEGMENTOS

Los demás segmentos tratan la evolución de las etapas del grafcet y las salidas (informa del estado de emergencia).

Network 3: x_marcha -> x_emer

Network 4: x_emer -> x_marcha

Network 5: Tratamiento estado de emergencia

GMM

Se encarga de determinar los modos de marcha (lote pequeño, lote mediano, lote grande o congelación). Cuando se encuentra en marcha las unidades productivas están activas; cuando está en parada se desconectan. Se inicia con el OB100 y permanece siempre activo. Presenta la siguiente interfaz:

- On: pasa al estado de marcha.
- Lote_pequenio: los lotes a procesar son pequeños.
- Lote_mediano: los lotes a procesar son medianos.
- Lote grande: los lotes a procesar son grandes.
- Mcong: modo de marcha con congelación.
- Cong: señal de congelación.
- Emer: señal de emergencia.
- Bit_paro: indica que el bloque se encuentra en paro.
- Bit_marcha: indica que el bloque está en marcha (si no está en parada).
- Bit_congelacion: indica que el bloque está en estado de congelación.

- X_parada: estado de parada.
- X_marcha: estado de marcha.
- X_lote_pequenio: estado de producción de lotes pequeños.
- X_lote_mediano: estado de producción de lotes medianos.
- X_lote_grande: estado de producción de lotes grandes.
- X_mcong: estado de marcha con posibilidad de congelación.
- X_cong: estado de congelación.
- Bit_trabajo: controla la conexión del bloque.

PLANTILLA DE CONEXIÓN

Conecta el bloque con el inicio del programa en el OB100. Solo se ejecuta una vez la plantilla de conexión.

```
SET
FP    #bit_trabajo
S     #x_parada
R     #x_marcha
R     #x_lote_pequenio
R     #x_lote_mediano
R     #x_lote_grande
R     #x_cong
```

PLANTILLA DE FORZADO – ERROR ASÍNCRONO

En caso de emergencia global, se fuerza al estado de parada y se reinician los demás estados.

```
U     #emer
S     #x_parada
R     #x_marcha
R     #x_lote_grande
R     #x_lote_mediano
R     #x_lote_pequenio
R     #x_mcong
R     #x_cong
R     #bit_marcha
R     #bit_congelacion
BEB
```

TRATAMIENTO TAMAÑO LOTES

Carga los datos de producción con los datos adecuados según el tamaño de lote.

Solamente varía el tiempo de procesado en la CNC.

```
      U     #x_lote_pequenio

      SPBN  _lotePequenio

      L     "datosLotePequenio".tmax_cnc_lapiz
      T     "datosProduccion".tmax_cnc_lapiz
      L     "datosLotePequenio".tmax_cnc_sacapuntas
      T     "datosProduccion".tmax_cnc_sacapuntas
      L     "datosLotePequenio".tmax_cnc_escuadra
      T     "datosProduccion".tmax_cnc_escuadra
      L     "datosLotePequenio".tmax_cnc_cartabon
      T     "datosProduccion".tmax_cnc_cartabon

_lotePequenio : NOP 0
```

```
      U     #x_lote_mediano

      SPBN  _loteMediano

      L     "datosLoteMediano".tmax_cnc_lapiz
      T     "datosProduccion".tmax_cnc_lapiz
      L     "datosLoteMediano".tmax_cnc_sacapuntas
      T     "datosProduccion".tmax_cnc_sacapuntas
      L     "datosLoteMediano".tmax_cnc_escuadra
      T     "datosProduccion".tmax_cnc_escuadra
      L     "datosLoteMediano".tmax_cnc_cartabon
      T     "datosProduccion".tmax_cnc_cartabon

_loteMediano : NOP 0
```

```
      U     #x_lote_grande

      SPBN  _loteGrande

      L     "datosLoteGrande".tmax_cnc_lapiz
      T     "datosProduccion".tmax_cnc_lapiz
      L     "datosLoteGrande".tmax_cnc_sacapuntas
      T     "datosProduccion".tmax_cnc_sacapuntas
      L     "datosLoteGrande".tmax_cnc_escuadra
      T     "datosProduccion".tmax_cnc_escuadra
      L     "datosLoteGrande".tmax_cnc_cartabon
      T     "datosProduccion".tmax_cnc_cartabon

_loteGrande : NOP 0
```

RESTO DE SEGMENTOS

El resto de segmentos se encargan de las transiciones entre estados y la activación de los actuadores:

Network 3: x_parada -> x_marcha

Network 4: x_marcha -> x_parada

Network 5: x_marcha -> x_lote_pequenio

Network 6: x_marcha -> x_lote_mediano

Network 7: x_marcha -> x_lote_grande

Network 8: x_marcha -> x_mcong

Network 9: x_mcong -> x_cong

Network 10: x_lote_pequenio -> x_marcha

Network 11: x_lote_mediano -> x_marcha

Network 12: x_lote_grande-> x_marcha

Network 13: x_cong -> x_marcha

Network 14: x_cong -> x_mcong

Network 18: Bit de marcha

Network 19: Bit de parada

Network 20: Bit de congelación

UNIDADES PRODUCTIVAS

UP_SIMPLE_EFECTO

Gestiona el cilindro de simple efecto.
Presenta los siguientes parámetros:

- start: inicia la unidad productiva.
- emer: señal de emergencia (error asíncrono).
- cong: señal de congelación.
- rearme: vuelta al estado de standby de un estado de error.
- scomp: cilindro comprimido.
- sexp: cilindro expandido.
- push: autorización para expandirse.
- tiempo_max: tiempo máximo para las maniobras.
- temp_max: temporizador de las manioabras.
- expandir: expande el cilindro (no hace falta actuador para comprimir porque es un simple efecto).
- bit_standby: estado de standby.
- bit_ocupación: el robot se encuentra en medio de una maniobra.
- bit_error_tiempo_expansión: error síncrono en la expansión.
- bit_error_tiempo_compresión: error síncrono en la compresión.
- bit_error_config_expansión: error síncrono por mala configuración inicial en la expansión.
- bit_error_config_compresión: error síncrono por mala configuración inicial en la compresión.
- bit_emer: estado de emergencia.
- bit_trabajo: se encarga de la conexión de la unidad productiva.
- x_expansión: estado de expansión.
- x_compresión: estado de compresión.
- x_standby: estado de standby.
- x_error_tiempo_expansión: estado de error síncrono en la expansión.
- x_error_tiempo_compresión: estado de error síncrono en la compresión.
- x_error_config_expansión: estado de error síncrono por mala configuración inicial en la expansión.
- x_error_config_compresión: estado de error síncrono por mala configuración inicial en la compresión.
- x_emer: estado de error asíncrono.
- SFC_Expansion: maniobra de expansión.
- SFC_Compresión: maniobra de compresión (sin actuador).
- bit_fin_expansión: finalización de la expansión.
- bit_fin_compresión: finalización de la compresión.

No se tiene un actuador "compresion" porque el cilindro es un simple efecto. No actuar sobre el cilindro supone la compresión. Si se definiese un estado de compresión, habría conflictos porque al entrar en emergencia, se llamaría al mismo actuador con "expansion" = 1 y "compresion" = 1.

Sin embargo, es necesario llamar a la maniobra de compresión para encapsularla y gestionar errores por tiempo y mala configuración inicial.

PLANTILLA DE DESCONEXIÓN - STOP

Desconexión de la unidad productiva.

Reinicia todos los parámetros para futuras conexiones.

Llamada a las maniobras usando la entrada asíncrona para indicar su pausa.

Después de la caja de código se especifica la instrucción “UN #start”, que coincide con la instrucción con la que se activó la plantilla. Esto es porque después de la caja, la RLO = 1 y /ER = 0. Por lo tanto, si la plantilla no se ejecuta, tras saltar la caja de código la RLO = 1 cuando debería ser 0 (UN #start no se cumple).

```
1      UN    #start
2      R     #bit_trabajo
3      R     #bit_emer
4      R     #bit_error_config_expansion
5      R     #bit_error_config_compresion
6      R     #bit_error_tiempo_expansion
7      R     #bit_error_tiempo_compresion
8      R     #bit_standby
9      R     #bit_ocupacion
10     R     #temp_max
11     FR    #temp_max
12
13     SPBN  stop
```

```
41
42     UN    #start
43     BEB
```

```
13     SPBN  stop
14
15     CALL  #SFC_Expansion
16        sini            :=
17        sfin            :=
18        reset           :=TRUE
19        tiempo_error    :=
20        temp_error      :=
21        bit_fin         :=
22        bit_error_tiempo :=
23        bit_error_config :=
24        bit_ocupacion   :=
25        actuador        :=#expandir
26
27
28     CALL  #SFC_Compresion
29        sini            :=
30        sfin            :=
31        reset           :=TRUE
32        tiempo_error    :=
33        temp_error      :=
34        bit_fin         :=
35        bit_error_tiempo :=
36        bit_error_config :=
37        bit_ocupacion   :=
38        actuador        :=
39
40  stop: NOP 0
```

PLANTILLA DE CONEXIÓN

Conexión de la unidad productiva.

```
1   U    #start
2   FP   #bit_trabajo
3   S    #x_standby
4   R    #x_expansion
5   R    #x_compresion
6   R    #x_emer
7   R    #x_error_config_expansion
8   R    #x_error_config_compresion
9   R    #x_error_tiempo_expansion
10  R    #x_error_tiempo_compresion
```

PLANTILLA DE CONGELACIÓN

Congela la unidad productiva: habilitación de componentes y reset de actuadores.

Los estados no cambian y no se va a un estado especial, de forma que cuando se quite la congelación la unidad productiva vuelva a la etapa en la que estaba.

Se repite el estímulo de activación para saltar el resto de segmentos.

No se llama al actuador de compresión porque no existe.

```
1   U    #cong
2
3   SPBN _cong
4
5   R    #temp_max            //Habilitación de componentes
6   FR   #temp_max
7
8   CALL #SFC_Expansion
9      sini             :=
10     sfin             :=
11     reset            :=TRUE
12     tiempo_error     :=
13     temp_error       :=
14     bit_fin          :=
15     bit_error_tiempo :=
16     bit_error_config :=
17     bit_ocupacion    :=
18     actuador         :=#expandir
```

```
20     CALL #SFC_Compresion
21        sini             :=
22        sfin             :=
23        reset            :=TRUE
24        tiempo_error     :=
25        temp_error       :=
26        bit_fin          :=
27        bit_error_tiempo :=
28        bit_error_config :=
29        bit_ocupacion    :=
30        actuador         :=
31
32 _cong: NOP 0
33
34     U    #cong                //Repetición de
35     BEB
```

ENTRADA EN ESTADO DE EMERGENCIA

No se entrará si se está en error síncrono para evitar perder la información de error. Resetea los actuadores.

```
1      O     #x_standby
2      O     #x_expansion
3      O     #x_compresion
4      U     #emer
5      S     #x_emer
6      R     #x_standby
7      R     #x_compresion
8      R     #x_expansion
9
10     SPBN  _emer
11
12     CALL  #SFC_Expansion
13        sini             :=
14        sfin             :=
15        reset            :=TRUE
16        tiempo_error     :=
17        temp_error       :=
18        bit_fin          :=
19        bit_error_tiempo :=
20        bit_error_config :=
21        bit_ocupacion    :=
22        actuador         :=#expandir
23
```

```
25     CALL  #SFC_Compresion
26        sini             :=
27        sfin             :=
28        reset            :=TRUE
29        tiempo_error     :=
30        temp_error       :=
31        bit_fin          :=
32        bit_error_tiempo :=
33        bit_error_config :=
34        bit_ocupacion    :=
35        actuador         :=
36
37  _emer: NOP 0
38
39     O     #x_standby
40     O     #x_expansion
41     O     #x_compresion
42     U     #emer
43     BEB
```

LLAMADA A LA MANIOBRA DE EXPANSIÓN

Llama a SFC_Cilindro como bloque multiinstancia (mejor portabilidad y no hace falta usar una gran cantidad de bloques de datos). Si se produce un error por tiempo o mala configuración inicial se va a un estado de error. Si la maniobra finaliza exitosamente se va al estado de compresión.

```
1      U     #x_expansion
2
3      SPBN  _exp
4      CALL  #SFC_Expansion
5         sini             :=#scomp
6         sfin             :=#sexp
7         reset            :=FALSE                                  FALSE
8         tiempo_error     :=#tiempo_max
9         temp_error       :=#temp_max
10        bit_fin          :=#bit_fin_expansion
11        bit_error_tiempo :=#bit_error_tiempo_expansion
12        bit_error_config :=#bit_error_config_expansion
13        bit_ocupacion    :=#bit_ocupacion
14        actuador         :=#expandir
15
16
17     U     #bit_error_tiempo_expansion
18     S     #x_error_tiempo_expansion
19     R     #x_expansion
20
21     U     #bit_error_config_expansion
22     S     #x_error_config_expansion
23     R     #x_expansion
24
25     U     #bit_fin_expansion
26     S     #x_compresion
27     R     #x_expansion
28
29  _exp: NOP 0
```

LLAMADA A LA MANIOBRA DE COMPRESIÓN

No se indica actuador por ser un cilindro de simple efecto. Si no se actúa sobre el cilindro, este se comprime.

```
1        U     #x_compresion
2
3        SPBN  _comp
4        CALL  #SFC_Compresion
5           sini              :=#sexp
6           sfin              :=#scomp
7           reset             :=FALSE
8           tiempo_error      :=#tiempo_max
9           temp_error        :=#temp_max
10          bit_fin           :=#bit_fin_compresion
11          bit_error_tiempo :=#bit_error_tiempo_compresion
12          bit_error_config :=#bit_error_config_compresion
13          bit_ocupacion     :=#bit_ocupacion
14          actuador          :=
15
16       U     #bit_error_tiempo_compresion
17       S     #x_error_tiempo_compresion
18       R     #x_compresion
19
20       U     #bit_error_config_compresion
21       S     #x_error_config_compresion
22       R     #x_compresion
23
24       U     #bit_fin_compresion
25       S     #x_standby
26       R     #x_compresion
27
28 _comp: NOP 0
```

RESTO DE SEGMENTOS

El resto de segmentos indican la evolución del grafcet y la activación de los bits de estado:

Network 5:	Salida del estado de emergencia
Network 6:	Rearme errores síncronos
Network 7:	x_standby -> x_expansion
Network 8:	Llamada a la maniobra de expansión
Network 9:	Llamada a la maniobra de compresión
Network 10:	Bit de standby
Network 11:	Bit de emergencia
Network 12:	Bit error tiempo expansión
Network 13:	Bit error tiempo compresión
Network 14:	Bit error config expansión
Network 15:	Bit error config compresión

UP_ROBOT

Gestiona el robot pick and place.
Presenta los siguientes parámetros:

- start: inicia la unidad productiva.
- emer: señal de emergencia (error asíncrono).
- cong: señal de congelación.
- rearme: vuelta al estado de standby de un estado de error.
- dc: señal de carga.
- fin_p: señal de descarga.
- home: indica el estado en la posición fetal.
- tiempo_max: tiempo máximo para las maniobras.
- temp_max: temporizador de las manioabras.
- cargar: programa de carga.
- descargar: programa de descarga.
- bit_standby: estado de standby.
- bit_ocupación: el robot se encuentra en medio de una maniobra.
- bit_error_tiempo_carga: error síncrono en la carga.
- bit_error_tiempo_descarga: error síncrono en la descarga.
- bit_error_config_carga: error síncrono por mala configuración inicial en la carga.
- bit_error_config_descarga: error síncrono por mala configuración inicial en la descarga.
- bit_emer: estado de emergencia.
- bit_trabajo: se encarga de la conexión de la unidad productiva.
- x_carga: estado de carga.
- x_descarga: estado de descarga.
- x_standby: estado de standby.
- x_error_tiempo_carga: estado de error síncrono en la carga.
- x_error_tiempo_descarga: estado de error síncrono en la descarga.
- x_error_config_carga: estado de error síncrono por mala configuración inicial en la carga.
- x_error_config_descarga: estado de error síncrono por mala configuración inicial en la descarga.
- x_espera_home_carga: estado transitorio entre la maniobra de carga y el estado de standby.
- x_espera_home_descarga: estado transitorio entre la maniobra de descarga y el estado de standby.
- SFC_Programa_Carga: maniobra de carga de tipo SFC_Programa con bloque de datos multiinstancia.
- SFC_Programa_ Descarga: maniobra de descarga de tipo SFC_Programa con bloque de datos multiinstancia.
- bit_fin_carga: finalización de la carga.
- bit_fin_descarga: finalización de la descarga.

Las variables “bit_fin_carga” y “bit_fin_descarga” son temporales porque las unidades productivas están pensadas para estar trabajando constantemente.

PLANTILLA DE DESCONEXIÓN - STOP

Desconecta la unidad productiva y reinicia los parámetros para futuras conexiones. Igual que la de la UP_Simple_Efecto.

PLANTILLA DE CONEXIÓN

Conexión de la unidad productiva. Igual que la de la UP_Simple_Efecto.

PLANTILLA DE CONGELACIÓN

Congela la unidad productiva. Igual que la de la UP_Simple_Efecto.

ENTRADA EN ESTADO DE EMERGENCIA

Entrada en defecto por error asíncrono. Igual que la de la UP_Simple_Efecto.

LLAMADAS A LAS MANIOBRAS

Usa las maniobras SFC_Programa para cargar en el robot el programa correspondiente (carga o descarga). Hay 2 segmentos: uno para la llamada a la maniobra de carga y otro para la de descarga. Se usan bloques multiinstancia para mejorar la portabilidad del código y no definir un bloque de datos por cada maniobra. Igual que las de la UP_Simple_Efecto.

RESTO DE SEGMENTOS

Los segmentos incluidos en el FB son los siguientes:

Network 1:	Plantilla de desconexión - STOP
Network 2:	Plantilla de conexión
Network 3:	Plantilla de congelación
Network 4:	Entrada en estado de emergencia
Network 5:	Salida del estado de emergencia
Network 6:	Rearme errores síncronos
Network 7:	x_standby -> x_carga
Network 8:	x_standby -> x_descarga
Network 9:	x_espera_home_carga -> x_standby
Network 10:	x_espera_home_descarga -> x_standby
Network 11:	Llamada a la maniobra de carga
Network 12:	Llamada a la maniobra de descarga
Network 13:	Bit de standby
Network 14:	Bit de emergencia
Network 15:	Bit error tiempo carga
Network 16:	Bit error tiempo descarga
Network 17:	Bit error config carga
Network 18:	Bit error config descarga

UP_CNC

Gestiona la CNC.

Presenta los siguientes parámetros:

- start: inicia la unidad productiva.
- emer: señal de emergencia (error asíncrono).
- cong: señal de congelación.
- rearme: vuelta al estado de standby de un estado de error.
- lapiz: indica que se van a procesar lápices.
- sacapuntas: indica que se van a procesar sacapuntas.
- escuadra: indica que se van a procesar escuadras.
- cartabon: indica que se van a procesar cartabones.
- home: indica el estado en la posición fetal.
- tiempo_lápiz: tiempo máximo para fabricar el lápiz.
- tiempo_sacapuntas: tiempo máximo para fabricar el sacapuntas.
- tiempo_escuadra: tiempo máximo para fabricar el sacapuntas.
- tiempo_cartabon: tiempo máximo para fabricar el cartabón.
- temp_max: temporizador de las manioabras.
- cnc_lapiz: programa del lápiz.
- cnc_sacapuntas: programa del sacapuntas.
- cnc_escuadra: programa de la escuadra.
- cnc_cartabon: programa del cartabón.
- bit_standby: estado de standby.
- bit_ocupación: el robot se encuentra en medio de una maniobra.
- bit_error_tiempo_lapiz: error síncrono en la lápiz.
- bit_error_tiempo_sacapuntas: error síncrono en la sacapuntas.
- bit_error_tiempo_escuadra: error síncrono en las escuadras.
- bit_error_tiempo_cartabon: error síncrono en los cartabones.
- bit_error_config_lapiz: error síncrono por mala configuración inicial en la fabricación del lápiz.
- bit_error_config_sacapuntas: error síncrono por mala configuración inicial en la fabricación del sacapuntas.
- bit_error_config_escuadras: error síncrono por mala configuración inicial en la fabricación de la escuadra.
- bit_error_config_cartabon: error síncrono por mala configuración inicial en la fabricación del cartabón.
- bit_emer: estado de emergencia.
- bit_trabajo: se encarga de la conexión de la unidad productiva.
- x_carga: estado de carga.
- x_descarga: estado de descarga.
- x_standby: estado de standby.
- x_error_tiempo_lapiz: estado de error síncrono en la fabricación del lápiz.
- x_error_tiempo_sacapuntas: estado de error síncrono en la la fabricación del sacapuntas.
- x_error_tiempo_escuadra: estado de error síncrono en la fabricación de la escuadra.
- x_error_tiempo_cartabon: estado de error síncrono en la la fabricación del cartabón.
- x_error_config_lapiz: estado de error síncrono por mala configuración inicial en la fabricación del lápiz.
- x_error_config_sacapuntas: estado de error síncrono por mala configuración inicial en la fabricación del sacapuntas.

- x_error_config_escuadra: estado de error síncrono por mala configuración inicial en la fabricación del lápiz.
- x_error_config_cartabon: estado de error síncrono por mala configuración inicial en la fabricación del cartabon.
- x_emer: estado de emergencia por error asíncrono.
- SFC_lapiz: maniobra para la fabricación del lápiz de tipo SFC_Programa con bloque de datos multiinstancia.
- SFC_sacapuntas: maniobra para la fabricación del sacapuntas de tipo SFC_Programa con bloque de datos multiinstancia.
- SFC_escuadra: maniobra para la fabricación de la escuadra de tipo SFC_Programa con bloque de datos multiinstancia.
- SFC_cartabon: maniobra para la fabricación del cartabon de tipo SFC_Programa con bloque de datos multiinstancia.
- bit_fin_lápiz: finalización de la fabricación del lápiz.
- bit_fin_sacapuntas: finalización de la fabricación del sacapuntas.
- bit_fin_escuadra: finalización de la fabricación de la escuadra.
- bit_fin_cartabón: finalización de la fabricación del cartabón.

Los bits de fin son temporales porque las unidades productivas están pensadas para estar trabajando constantemente.

PLANTILLA DE DESCONEXIÓN - STOP

Desconecta la unidad productiva y reinicia los parámetros para futuras conexiones. Igual que la de la UP_Simple_Efecto.

PLANTILLA DE CONEXIÓN

Conexión de la unidad productiva. Igual que la de la UP_Simple_Efecto.

PLANTILLA DE CONGELACIÓN

Congela la unidad productiva. Igual que la de la UP_Simple_Efecto.

ENTRADA EN ESTADO DE EMERGENCIA

Entrada en defecto por error asíncrono. Igual que la de la UP_Simple_Efecto.

LLAMADA A LAS MANIOBRAS

Usa las maniobras SFC_Programa (se reutiliza) para cargar en el robot el programa correspondiente (lápiz, sacapuntas, escuadra, cartabón). Hay 4 segmentos: uno para cada producto. Se usan bloques multiinstancia para mejorar la portabilidad del código y no definir un bloque de datos por cada maniobra. Igual que las de la UP_Simple_Efecto.

RESTO DE SEGMENTOS

Los segmentos incluidos en el FB son los siguientes:

Network 1:	Plantilla de desconexión - STOP
Network 2:	Plantilla de conexión
Network 3:	Plantilla de congelación
Network 4:	Entrada en estado de emergencia
Network 5:	Salida del estado de emergencia
Network 6:	Rearme errores síncronos
Network 7:	x_standby -> x_lapiz
Network 8:	x_standby -> x_sacapuntas
Network 9:	x_standby -> x_escuadra
Network 10:	x_standby -> x_cartabon
Network 11:	Llamada a la maniobra de fabricación del lápiz
Network 12:	Llamada a la maniobra de fabricación del sacapuntas
Network 13:	Llamada a la maniobra de fabricación del sacapuntas
Network 14:	Llamada a la maniobra de fabricación del sacapuntas
Network 15:	Bit de standby
Network 16:	Bit de emergencia
Network 17:	Bit error tiempo lápiz
Network 18:	Bit error tiempo sacapuntas
Network 19:	Bit error tiempo escuadra
Network 20:	Bit error tiempo cartabón
Network 21:	Bit error config lápiz
Network 22:	Bit error config sacapuntas
Network 23:	Bit error config escuadra
Network 24:	Bit error config cartabón

MANIOBRAS

SFC_PROGRAMA

Ejecuta los programas de la unidad productiva con control de tiempo y configuración inicial.

Tiene los siguientes parámetros:

- sensor1: receptividad entre xe y x1.
- sensor2: receptividad entre xs y x1.
- reset: indica la desconexión por error asíncrono.
- tiempo_error: tiempo límite antes de error síncrono.
- temp_error: temporizador de error síncrono.
- bit_fin: la maniobra ha finalizado correctamente.
- bit_error_tiempo: tiempo límite agotado (error síncrono).
- bit_error_config: mala configuración inicial (error síncrono).
- bit_ocupacion: maniobra en ejecución.
- actuador1: programa a cargar en el robot.
- bit_trabajo: conecta el bloque maniobra.
- xe: estado inicial.
- x1: estado de ejecución del programa.
- x2: estado de error síncrono por tiempo.
- x3: estado de error síncrono por mala configuración inicial.
- xs: estado final.
- mf_sensor2: marca de flanco del sensor 2.
- mp_sensor2: merca de pulso del sensor 2.

PLANTILLA DE DESCONEXIÓN – ERROR ASÍNCRONO

En caso de error asíncrono, desconecta los actuadores y habilita los componentes para futuras conexiones. No es necesario notificar al exterior, ya que la señal de error asíncrono es externa (es el exterior quien solicita este estado).

```
1    U    #reset
2    R    #bit_fin
3    R    #bit_trabajo          //Futuros arranques
4    R    #bit_ocupacion
5    R    #actuador1
6    R    #temp_error           //Habilitación de temporizadores
7    FR   #temp_error
8    R    #mf_sensor2
9    BEB
```

ERROR SÍNCRONO POR TIEMPO

Desconexión por tiempo límite (actuadores, temporizadores, bit de fin y bit de trabajo).

Es necesario desconectar x2 porque si no no se podría volver a ejecutar la maniobra (se saldría continuamente

del bloque).

Se deja todo preparado para futuras conexiones.

```
1    U    #x2
2    S    #bit_error_tiempo
3    R    #x2
4    R    #bit_fin
5    R    #bit_trabajo          //Futuros arranques
6    R    #bit_ocupacion
7    R    #actuador1
8    R    #temp_error           // Habilitación de temporizadores
9    FR   #temp_error
10   R    #mf_sensor2
11   BEB
```

ERROR SÍNCRONO POR MALA CONFIGURACIÓN INICIAL

Desconexión por mala configuración inicial.

Es necesario desconectar x3 porque si no no se podría volver a ejectar la maniobra.

Se deja todo preparado para futuras conexiones.

```
1    U    #x3
2    S    #bit_error_config
3    R    #x3
4    R    #bit_trabajo
5    R    #bit_fin
6    R    #bit_ocupacion
7    R    #actuador1
8    R    #temp_error
9    FR   #temp_error
10   R    #mf_sensor2
11   BEB
```

PLANTILLA DE DESCONEXIÓN – FINAL DE MANIOBRA

Desconexión por fin de la maniobra. Se deja todo preparado para futuras conexiones y se notifica al exterior (set bit_fin).

```
1    U    #xs
2    S    #bit_fin
3    R    #xs
4    R    #bit_trabajo
5    R    #bit_ocupacion
6    R    #actuador1
7    R    #temp_error
8    FR   #temp_error
9    R    #mf_sensor2
10   [illegible]
```

Plantilla de conexión

Inicialización del bloque a valores iniciales.
Se resetean los bits de control y los estados.

```
1    SET
2    FP   #bit_trabajo
3    R    #bit_fin
4    R    #bit_error_config
5    R    #bit_error_tiempo
6    R    #bit_ocupacion
7    S    #xe
8    R    #x1
9    R    #x2
10   R    #x3
11   R    #xs
```

TRATAMIENTO ETAPAS FUGACES

Se altera el orden de los segmentos de transición entre etapas para evitar problemas de fugacidad.

Al entrar a la maniobra se produce una marca de pulso si la configuración inicial es correcta: "sensor2" está a nivel alto y la marca de flanco comienza el bloque siempre a nivel bajo. Esto provocaría que la maniobra sea fugaz.

Se cambia el orden de las etapas para arreglar este problema.

Network 8: x1 -> x2 (Error síncrono por tiempo)

Network 9: x1 -> xs - Estado fugaz

Network 10: xe -> x3 (Error síncrono por mala configuración inicial)

Network 11: xe -> x1

RESTO DE SEGMENTOS

Los demás segmentos se encargan de la ejecución del grafcet y no tienen gran importancia:

Network 6:	Configuración flancos
Network 7:	Configuración temporizador maniobra
Network 8:	x1 -> x2 (Error síncrono por tiempo)
Network 9:	x1 -> xs - Estado fugaz
Network 10:	xe -> x3 (Error síncrono por mala configuración inicial)
Network 11:	xe -> x1
Network 12:	Cargar programa

SFC_CILINDRO

Se encarga de ejecutar las maniobras de carga y descarga de la unidad productiva.

El funcionamiento es exactamente igual que SFC_Programa, salvo por la configuración inicial necesaria para su correcto funcionamiento (Sini a nivel alto) y la transición entre el estado de ocupación y el estado final (Sfin a nivel alto; antes era por flanco).

Cabe destacar el tratamiento de etapas fugaces: para evitar una configuración inicial errónea estando Sini en nivel alto (en principio transitaría al estado de ocupación), se altera el orden de los segmentos:

Network 7:	xe -> x3 (Error síncrono por mala configuración inicial)
Network 8:	xe -> x1

SEÑALES ANALÓGICAS

ALTURAPALETS [FC]

Gestiona las señales analógicas de la pila de palets vacíos. Determina cuándo la pila puede ser transportada a otro lugar en la fábrica.

Devuelve los conectores con las condiciones necesarias para la implemetación del grafcet, la señal física y una señal de error en caso de que la función falle.

Toma los siguientes parámetros:

- RAW: señal analógica muestreada.
- BIPOLAR: especifica si la señal puede tomar valores en un rango solo positivo o positivo y negativo.
- LO_LIM: altura mínima permitida para la interpolación.
- HI_LIM: altura máxima permitida para la interpolación.

- h_min: altura mínima de la pila de palets para ser transportada a otro lugar de la fábrica. Se pueden apilar más palets en la pila después de alcanzar este valor.
- h_max: altura máxima de la pila de palets para ser transportada a otro lugar de la fábrica. No se pueden apilar más palets en la pila si se alcanza este valor.
- reset: desconecta el bloque FC.
- REAL_VAL: valor de la magnitud física (altura).
- h>h_min: la altura de la pila ha superado h_min.
- h>h_max: la altura de la pila ha superado h_max.
- h<h_min: la altura de la pila es menor que h_min.
- error: la altura medida está fuera de límites.

PLANTILLA DE DESCONEXIÓN

En el caso de que se active la entrada "reset" se ejecuta la plantilla de desconexión (resetea las salidas) y se sale del bloque:

```
1    U    #reset
2    R    #"h>h_min"
3    R    #"h>h_max"
4    R    #"h<h_min"
5    R    #error
6    BEB
```

LLAMADA A LA FUNCIÓN SCALE

Convierte la magnitud muestreada (raw) en magnitud física (real_val). La señal bipolar será "false" (no hay alturas negativas), aunque se dejará como entrada a gusto del cliente. Los límites entre los que se interpola (lo_lim y hi_lim) y los valores de referencia de altura de la pila de palets (h_min y h_max) se guardarán en el bloque de datos de producción.

```
1    CALL  SCALE
2       IN      :=#RAW
3       HI_LIM  :=#HI_LIM
4       LO_LIM  :=#LO_LIM
5       BIPOLAR :=#BIPOLAR
6       RET_VAL :=#ret_val_1
7       OUT     :=#REAL_VAL
```

ERROR DE LÍMITES

Si la altura está fuera de los límites definidos por lo_lim y hi_lim, se activa una señal de error. Esto se puede saber al comparar la señal ret_val con el código de error de la función SCALE (W#16#0008).

```
1    L    #ret_val_1
2    L    W#16#0008
3    ==I
4    =    #error
```

COMPARACIÓN

Las operaciones de comparación de señales analógicas no tienen mucho que comentar. Simplemente, se cargan los valores de la señal física y el valor de referencia con el que se quiera comparar y se lleva a cabo la comparación. Es conveniente conocer en qué acumulador se almacena cada variable. Las comparaciones de distinto tipo (==, <>, <= …) que involucran las mismas variables pueden realizarse consultando los biestables A0 y A1 (es el caso de la comparación con h_min). El valor devuelto por las comparaciones es un conector booleano que se usará como condición lógica en el resto del código.

```
1   L    #REAL_VAL      //ACU2
2   L    #h_min         //ACU1
3   >R
4   =    #"h>h_min"
```

```
1   L    #REAL_VAL      // ACU2
2   L    #h_max         // ACU1
3   >R
4   =    #"h>h_max"
```

```
U <0
=    #"h<h_min"
```

UP_PALETS_VACIOS (G4)

Funciona como interfaz para conocer el estado de la pila de palets. Se sirve de la función alturaPalets. Tiene los siguientes parámetros:

- start: conecta y desconecta la unidad productiva.
- Emer: entra en estado de emergencia.
- Rearme: sale del estado de emergencia.
- Cong: entra en congelación.
- Hraw: señal de entrada analógica muestreada que representa la altura de la pila de palets vacíos.
- LO_LIM: límite mínimo de altura para la interpolación.
- HI_LIM: límite máximo de altura para la interpolación.
- Hmin: altura mínima (magnitud física) de la pila de palets vacíos para ser recogida.
- Hmax: altura máxima (magnitud física) de la pila de palets vacíos para saturarse (no se pueden colocar más palets sobre la misma).
- Recoger_pila: llamada a los AGVs para recoger la pila de palets vacíos.
- Pila_saturada: llamada a los AGVs para recoger la pila de palets vacíos (urgentemente).
- Error_medida: la magnitud medida está fuera de rango (LO_LIM – HI_LIM).
- H_real: altura de la pila de palets vacíos en forma física.

- H>hmin: conector indicando que la altura de la pila de palets es mayor que h_min.
- H<hmin: conector indicando que la altura de la pila de palets es menor que h_min.
- H>hmax: conector indicando que la altura de la pila de palets es mayor que h_max.
- X1: estado de standby.
- X2: estado indicando que la altura de la pila de palets vacíos es mayor que hmin.
- X3: estado indicando que la altura de la pila de palets vacíos es mayor que hmax.
- X_emer: estado de emergencia.
- Bit_trabajo: bit de control del encendido de la unidad productiva (flanco positivo de start).
- Bit_standby: bit indicando el estado de standby.
- Bit_emer: bit indicando el estado de emergencia.

PLANTILLA DE DESCONEXIÓN – ERROR ASÍNCRONO

Entra en error asíncrono por emergencia. Desconecta las salidas y activa el estado de emergencia.

```
1    U    #emer
2    S    #x_emer
3    R    #recoger_pila
4    R    #pila_saturada
5    R    #bit_standby
6    R    #error_medida
7    R    #"h<hmin"
8    R    #"h>hmax"
9    R    #"h>hmin"
10   BEB
```

PLANTILLA DE DESCONEXIÓN - STOP

Se activa al desconectar la unidad productiva. Resetea los bits de control (importante resetear el bit de trabajo para futuras conexiones) y las salidas.

```
UN    #start
R     #bit_trabajo
R     #bit_standby
R     #recoger_pila
R     #pila_saturada
R     #error_medida
R     #"h<hmin"
R     #"h>hmax"
R     #"h>hmin"
BEB
```

PLANTILLA DE CONEXIÓN

Conecta la unidad productiva por flanco positivo del bit de trabajo (debe ser reseteado adecuadamente después de la desconexión). Establece el estado de standby como estado inicial.

```
1        U     #start
2        FP    #bit_trabajo
3        S     #x1
4        R     #x2
5        R     #x3
6        R     #x_emer
```

PLANTILLA DE CONGELACIÓN

Resetea las salidas y mantiene el estado actual en el que estaba para poder seguir produciendo tras descongelar.

```
U     #cong
R     #recoger_pila
R     #pila_saturada
R     #error_medida
R     #"h<hmin"
R     #"h>hmax"
R     #"h>hmin"
BEB
```

SALIDA DE EMERGENCIA

Rearme desde el estado de emergencia.

```
1        U     #x_emer
2        UN    #emer
3        U     #rearme
4        S     #x1
5        R     #x_emer
```

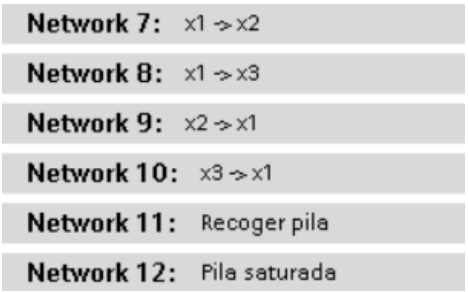
LLAMADA A ALTURAPALETS

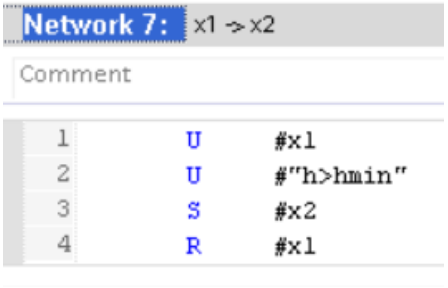
Llama al FC alturaPalets para conocer la altura de la pila de palets, así como los conectores usados para la transición entre estados (h<hmin, h>hmin, h>hmax). Los datos de los valores extremos, así como de hmin y hmax están en el DB datosProduccion.

```
1
2        CALL  "alturaPalets"
3           RAW      :=#hraw
4           BIPOLAR  :=FALSE
5           LO_LIM   :=#LO_LIM
6           HI_LIM   :=#HI_LIM
7           h_min    :=#hmin
8           h_max    :=#hmax
9           reset    :=FALSE
10          REAL_VAL :=#h_real
11          h>h_min  :=#"h>hmin"
12          h>h_max  :=#"h>hmax"
13          h<h_min  :=#"h<hmin"
14          error    :=#error_medida
```

RESTO DE SEGMENTOS

El resto de segmentos son las transiciones de estados y la activación de los actuadores (llamadas a los AGVs). Cabe destacar el uso de conectores de comparación en las transiciones entre etapas:





TEST UNITARIOS

UT_SFC_PROGRAMA

Test unitario de SFC_Programa.

Es necesario probarla dentro de un grafcet para asegurarse que los bits de maniobra y de error síncrono funcionan bien.

Los parámetros que no pertenecen a la maniobra son los siguientes:

- x0: estado inicial.
- m1: bit de maniobra (determina cuándo está activa la maniobra).
- x2: estado final (tras terminar la maniobra).
- x10: estados de error síncrono.
- r1: receptividad entre x0 y m1.
- r2: receptividad entre x2 y x0.
- rearme: receptividad entre x10 y x0.

¿Cómo ejecutar el test unitario? Se definen los parámetros indicados anteriormente y los parámetros de la maniobra. Luego se llama al FC.

Importante: es necesario definir x0 como el estado inicial en el OB100 para poder realizar la prueba correctamente.

TABLA DE SÍMBOLOS

Se define la siguiente tabla de símbolos:

1	x0	Default tag table	Bool	%M0.0
2	m1	Default tag table	Bool	%M0.1
3	x2	Default tag table	Bool	%M0.2
4	x10	Default tag table	Bool	%M0.3
5	bit_fin	Default tag table	Bool	%M0.4
6	bit_error_tiempo	Default tag table	Bool	%M0.5
7	bit_error_config	Default tag table	Bool	%M0.6
8	bit_ocupacion	Default tag table	Bool	%M0.7
9	sensor1	Default tag table	Bool	%E1.0
10	sensor2	Default tag table	Bool	%E1.1
11	reset	Default tag table	Bool	%E1.2
12	r1	Default tag table	Bool	%E1.3
13	r2	Default tag table	Bool	%E1.4
14	rearme	Default tag table	Bool	%E1.5
15	actuador1	Default tag table	Bool	%A1.6
16	temp	Default tag table	Timer	%T1

LLAMADA AL BLOQUE

Se llama al test unitario desde el OB1 (x0 debe estar definido como el estado inicial en el OB100 para que funcione):

Network 1: .....

Comment

```
1    CALL "UT_SFC_Programa"
```

Network 1: Inicialización

Comment

```
1    SET
2    S    "x0"
3    R    "x2"
4    R    "x10"
5    R    "m1"
```

LLAMADA A SFC_PROGRAMA

Cuando se active el bit de maniobra se llama al FB. Cuando finaliza la maniobra, ya sea por tiempo, configuración incorrecta, o éxito, se resetea el bit de maniobra y se va cambia de estado:

```
1    U    "m1"
2
3    SPBN _man
4
5    CALL "SFC_Programa", "UT_SFC_Programa_DB"
6       sensor1          :="sensor1"
7       sensor2          :="sensor2"
8       reset            :="reset"
9       tiempo_error     :=s5t#10s
10      temp_error       :="temp"
11      bit_fin          :="bit_fin"
12      bit_error_tiempo :="bit_error_tiempo"
13      bit_error_config :="bit_error_config"
14      bit_ocupacion    :="bit_ocupacion"
15      actuador1        :="actuador1"
```

```
17    U    "bit_error_tiempo"
18    S    "x10"
19    R    "m1"
20
21    U    "bit_error_config"
22    S    "x10"
23    R    "m1"
24
25    U    "bit_fin"
26    S    "x2"
27    R    "m1"
28
29 _man: NOP 0
```

RESTO DE SEGMENTOS

El resto de segmentos definen la sucesión de las demás etapas para poder probar el FB:

Network 1:	x0 -> m1
Network 2:	Llamada a la maniobra
Network 3:	x2 -> x0
Network 4:	x10 -> x0

UT_SFC_CILINDRO

Se realiza el mismo test que en el caso de SFC_Programa, pero llamando a SFC_Cilindro. La tabla de símbolos esta vez es la siguiente:

x0	Default tag table	Bool	%M0.0
m1	Default tag table	Bool	%M0.1
x2	Default tag table	Bool	%M0.2
x10	Default tag table	Bool	%M0.3
bit_fin	Default tag table	Bool	%M0.4
bit_error_tiempo	Default tag table	Bool	%M0.5
bit_error_config	Default tag table	Bool	%M0.6
bit_ocupacion	Default tag table	Bool	%M0.7
sini	Default tag table	Bool	%E1.0
sfin	Default tag table	Bool	%E1.1
reset	Default tag table	Bool	%E1.2
r1	Default tag table	Bool	%E1.3
r2	Default tag table	Bool	%E1.4
rearme	Default tag table	Bool	%E1.5
actuador1	Default tag table	Bool	%A1.6
temp	Default tag table	Timer	%T1

UT_SIMPLE_EFECTO

El test unitario consiste en llamar a la unidad productiva. Para ello se definen los parámetros en la tabla de símbolos y se llama al test unitario desde el OB1:

bit_standby	Default tag table	Bool	%M0.7
bit_ocupacion	Default tag table	Bool	%M1.0
bit_error_tiempo_expansion	Default tag table	Bool	%M1.1
bit_error_tiempo_compresion	Default tag table	Bool	%M1.2
bit_error_config_expansion	Default tag table	Bool	%M1.3
bit_error_config_compresion	Default tag table	Bool	%M1.4
bit_emer	Default tag table	Bool	%M1.5
start	Default tag table	Bool	%E1.0
emer	Default tag table	Bool	%E1.1
cong	Default tag table	Bool	%E1.2
rearme	Default tag table	Bool	%E1.3
scomp	Default tag table	Bool	%E1.4
sexp	Default tag table	Bool	%E1.5
push	Default tag table	Bool	%E1.6
expandir	Default tag table	Bool	%A1.6
temp	Default tag table	Timer	%T1

```
1    CALL "UT_Simple_Efecto"
```

```
CALL "UP_Simple_Efecto", "UP_Simple_Efecto_DB"
   start                    :="start"
   emer                     :="emer"
   cong                     :="cong"
   rearme                   :="rearme"
   scomp                    :="scomp"
   sexp                     :="sexp"
   push                     :="push"
   tiempo_max               :=s5t#10s
   temp_max                 :="temp"
   expandir                 :="expandir"
   bit_standby              :="bit_standby"
   bit_ocupacion            :="bit_ocupacion"
   bit_error_tiempo_expansion  :="bit_error_tiempo_expansion"
   bit_error_tiempo_compresion :="bit_error_tiempo_compresion"
   bit_error_config_expansion  :="bit_error_config_expansion"
   bit_error_config_compresion :="bit_error_config_compresion"
   bit_emer                 :="bit_emer"
```

UT_ROBOT

El test unitario consiste en llamar a la unidad productiva. Para ello se definen los parámetros en la tabla de símbolos y se llama al test unitario desde el OB1:

start	Default tag table	Bool	%E1.0
emer	Default tag table	Bool	%E1.1
cong	Default tag table	Bool	%E1.2
rearme	Default tag table	Bool	%E1.3
dc	Default tag table	Bool	%E1.4
fin_p	Default tag table	Bool	%E1.5
home	Default tag table	Bool	%E1.7
cargar	Default tag table	Bool	%A1.6
descargar	Default tag table	Bool	%A1.7
temp	Default tag table	Timer	%T1
bit_ocupacion	Default tag table	Bool	%M1.0
bit_error_tiempo_carga	Default tag table	Bool	%M1.1
bit_error_tiempo_descarga	Default tag table	Bool	%M1.2
bit_error_config_carga	Default tag table	Bool	%M1.3
bit_error_config_descarga	Default tag table	Bool	%M1.4
bit_emer	Default tag table	Bool	%M1.5
bit_standby	Default tag table	Bool	%M0.7

```
1    CALL  "UT_Robot"
```

```
1    CALL  "UP_Robot", "UP_Robot_DB"
2        start                    :="start"
3        emer                     :="emer"
4        rearme                   :="rearme"
5        cong                     :="cong"
6        dc                       :="dc"
7        fin_p                    :="fin_p"
8        home                     :="home"
9        tiempo_max               :=s5t#10s
10       temp_max                 :="temp"
11       cargar                   :="cargar"
12       descargar                :="descargar"
13       bit_standby              :="bit_standby"
14       bit_ocupacion            :="bit_ocupacion"
15       bit_error_tiempo_carga   :="bit_error_tiempo_carga"
16       bit_error_tiempo_descarga :="bit_error_tiempo_descarga"
17       bit_error_config_carga   :="bit_error_config_carga"
18       bit_error_config_descarga :="bit_error_config_descarga"
19       bit_emer                 :="bit_emer"
```

UT_CNC

El test unitario consiste en llamar a la unidad productiva. Para ello se definen los parámetros en la tabla de símbolos y se llama al test unitario desde el OB1:

start	Default tag table	Bool	%E1.0
emer	Default tag table	Bool	%E1.1
cong	Default tag table	Bool	%E1.2
rearme	Default tag table	Bool	%E1.3
lapiz	Default tag table	Bool	%E1.4
sacapuntas	Default tag table	Bool	%E1.5
escuadra	Default tag table	Bool	%E1.6
cartabon	Default tag table	Bool	%E2.0
cnc_lapiz	Default tag table	Bool	%A1.6
cnc_sacapuntas	Default tag table	Bool	%A2.0
cnc_escuadra	Default tag table	Bool	%A2.1
cnc_cartabon	Default tag table	Bool	%A2.2
temp	Default tag table	Timer	%T1
bit_ocupacion	Default tag table	Bool	%M1.0
bit_error_tiempo_lapiz	Default tag table	Bool	%M1.1
bit_error_tiempo_sacapuntas	Default tag table	Bool	%M1.2
bit_error_tiempo_escuadra	Default tag table	Bool	%M1.6
bit_error_tiempo_cartabon	Default tag table	Bool	%M1.7
bit_error_config_lapiz	Default tag table	Bool	%M1.3
bit_error_config_sacapuntas	Default tag table	Bool	%M1.4
bit_error_config_escuadra	Default tag table	Bool	%M2.0
bit_error_config_cartabon	Default tag table	Bool	%M2.1
bit_emer	Default tag table	Bool	%M1.5
bit_standby	Default tag table	Bool	%M0.7
home	Default tag table	Bool	%E1.7
descargar	Default tag table	Bool	%A1.7

```
1    CALL  "UT_CNC"
```

```
1    CALL  "UP_CNC", "UP_CNC_DB"                                          %FB1, %DB1
2       start                      :="start"                                 %E1.0
3       emer                       :="emer"                                  %E1.1
4       rearme                     :="rearme"                                %E1.3
5       cong                       :="cong"                                  %E1.2
6       lapiz                      :="lapiz"                                 %E1.4
7       sacapuntas                 :="sacapuntas"                            %E1.5
8       escuadra                   :="escuadra"                              %E1.6
9       cartabon                   :="cartabon"                              %E2.0
10      home                       :="home"                                  %E1.7
11      tiempo_max                 :=s5t#10s                                s5t#10s
12      temp_max                   :="temp"                                   %T1
13      cnc_lapiz                  :="cnc_lapiz"                             %A1.6
14      cnc_sacapuntas             :="cnc_sacapuntas"                        %A2.0
15      cnc_escuadra               :="cnc_escuadra"                          %A2.1
16      cnc_cartabon               :="cnc_cartabon"                          %A2.2
17      bit_standby                :="bit_standby"                           %M0.7
18      bit_ocupacion              :="bit_ocupacion"                         %M1.0
19      bit_error_tiempo_lapiz     :="bit_error_tiempo_lapiz"                %M1.1
20      bit_error_tiempo_sacapuntas :="bit_error_tiempo_sacapuntas"          %M1.2
21      bit_error_tiempo_escuadra  :="bit_error_tiempo_escuadra"             %M1.6
22      bit_error_tiempo_cartabon  :="bit_error_tiempo_cartabon"             %M1.7
23      bit_error_config_lapiz     :="bit_error_config_lapiz"                %M1.3
24      bit_error_config_sacapuntas :="bit_error_config_sacapuntas"          %M1.4
25      bit_error_config_escuadra  :="bit_error_config_escuadra"             %M2.0
26      bit_error_config_cartabon  :="bit_error_config_cartabon"             %M2.1
27      bit_emer                   :="bit_emer"                              %M1.5
```

UT_PALETS_VACIOS

Test unitario de la unidad productiva UP_Palets_Vacios. La llamada y la tabla de símbolos requerida es como sigue:

```
CALL "UP_Palets_Vacios", "UT_UP_Palets_Vacios_DB"
   start          :="start"
   emer           :="ut_emer"
   rearme         :="ut_rearme"
   cong           :="ut_cong"
   hraw           :="pa_hraw"
   LO_LIM         :="datosProduccion".LO_LIM
   HI_LIM         :="datosProduccion".HI_LIM
   hmin           :="datosProduccion".hmin
   hmax           :="datosProduccion".hmax
   recoger_pila   :="pa_recoger_pila"
   pila_saturada  :="pa_pila_saturada"
   error_medida   :="pa_error_medida"
   h_real         :="pa_hreal"
   h>hmin         :="pa_h>hmin"
   h>hmax         :="pa_h>hmax"
   h<hmin         :="pa_h<hmin"
```

```
1  call "UT_Palets_Vacios"
2
```

pa_emer	Bool	%M9.3
pa_rearme	Bool	%M9.4
pa_recoger_pila	Bool	%A9.5
pa_pila_saturada	Bool	%A9.6
pa_error_medida	Bool	%A9.7
pa_h<hmin	Bool	%A10.0
pa_h>hmin	Bool	%A10.1
pa_h>hmax	Bool	%A10.2
pa_hreal	Real	%AD10
pa_hraw	Int	%EW14