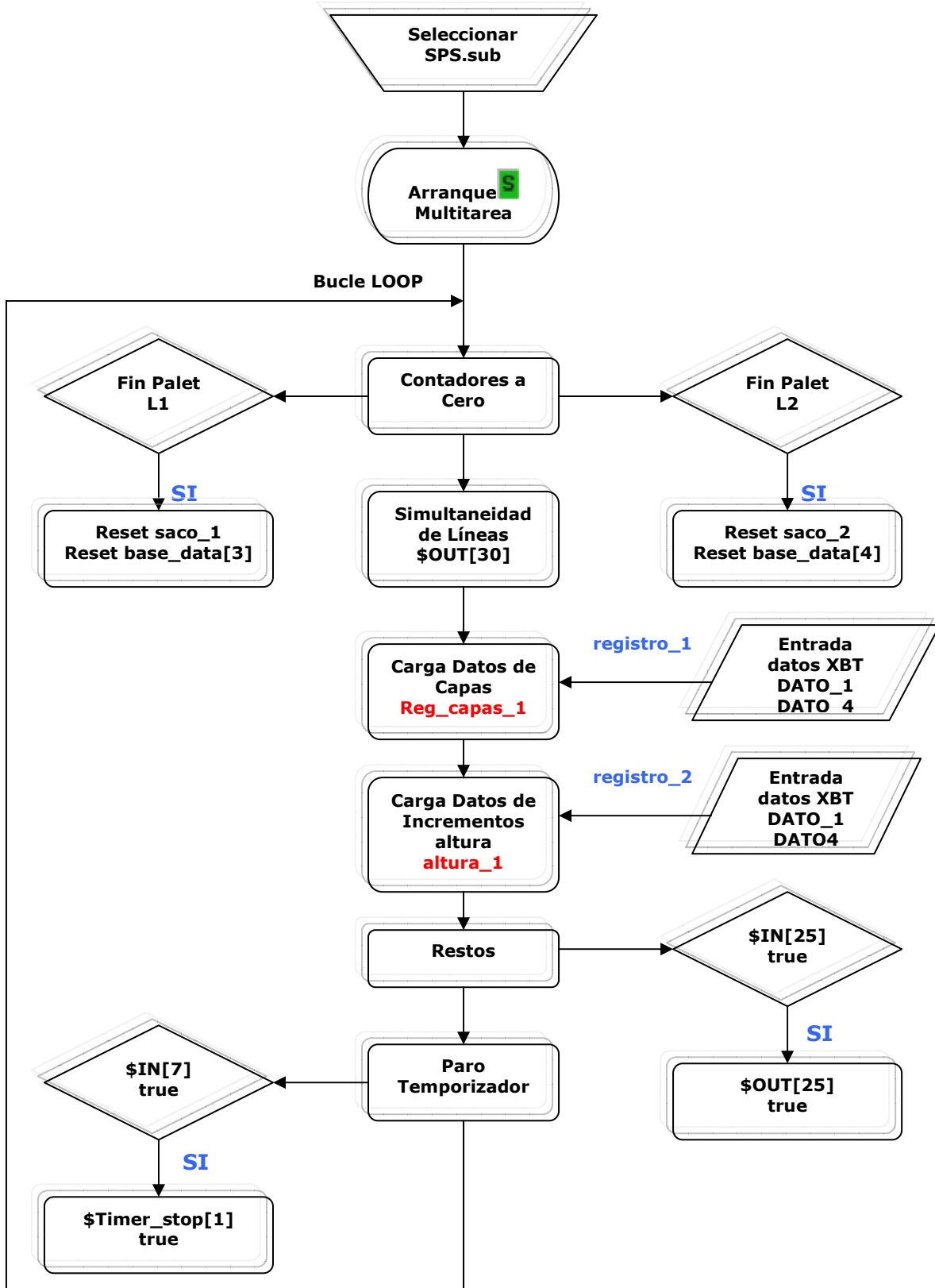




### 6.2.3 Diagramasgramas de estado software del robot.

#### 6.2.3.1 Interpretador Submit (Multitarea SUB.sps).



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

El interpretador del submit (**Ver ANEXO SPS.sub**) como hemos venido definiendo a lo largo del proyecto, es un programa multitarea de ejecución cíclica que requiere estar en **RUN** para el funcionamiento del **KRC**, este arranque se encuentra configurado de manera automática, pero es el operario final quién mediante su acción seleccionará el submit y lo arrancará, no necesitando en ningún momento del proceso pararse, ano ser que este provocado por condiciones anómalas de funcionamiento.

El programa se encuentra identificado por una serie de procesos cíclicos, integrados en un bucle infinito donde se repiten por cada período de reloj. Tal como se ha ido detallando estos procesos se ejecutan de manera cíclica. En el primero de ellos se evalúa la señal de fin de palet de cada una de las líneas, de manera que en caso de finalizado el palet, el programa multitarea reiniciará la variable de posición de saco y la coordenada Z de la base\_data.

En el segundo de los procesos, se evalúa la señal de simultaneidad de líneas, de manera que se asegura que si una de la líneas no tiene permiso de descarga por medio del PLC la señal **\$OUT[30]** permanecerá activada o desactivada consintiendo la descarga sobre la otra línea de paletizado. Este proceso solamente se evaluará si el PLC, quita permiso de descarga a alguna de las líneas de paletizado.

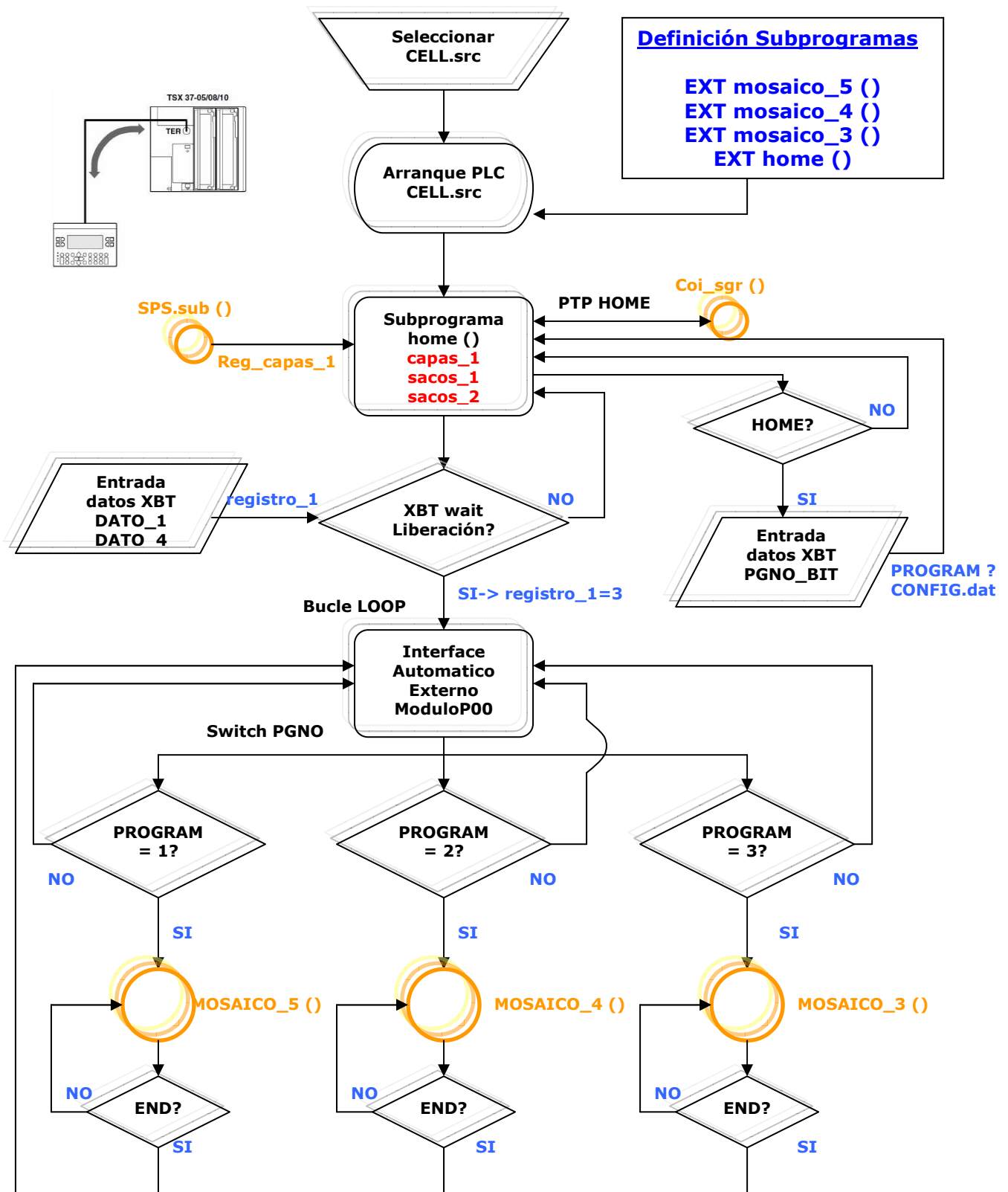
En los procesos de carga de datos, se captura los valores **registro\_1** y **registro\_2** por medio de los bit´s **DATO\_1 to DATO\_4** introducidos por pantalla en cuanto al número de alturas a paletizar y los incrementos. En el submit se generan los valores **reg\_capas\_1**, que determina el número de capas a paletizar cuando el programa se encuentre en **home()**, y el incremento de alturas **altura\_1** que se suma al incremento de Z constante por cada capa finalizada.



**Restos** simplemente captura la pulsación del armario eléctrico para tal fin, de manera que el robot finalizado los movimientos en ejecución del programa de paletizado actual, activará las señales de fin de palet en ambas líneas, y volverá posteriormente a **false** el estado de dicha señal.

El último de los procesos a llevar a cabo, es el paro de la temporización para activar el defecto de mordaza no en reposo; esto es, por que si el operario para la instalación mediante el pulsador de paro, se deshabilita la señal de habilitación de movimientos **MOVE\_ENABLE->\$IN[7]**, de esta manera si detenemos la cuenta del temporizador este no saltará la alarma con la instalación parada.

Finalizados todos los procesos, se ejecutaran todos y cada uno de ellos desde el principio, siendo este un bucle infinito mientras ningún ser humano detenga el interpretador.

**6.2.3.2 Automático externo (CELL.src).**



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Básicamente queda representada la estructura de programación del programa **Cell.src**, que al igual que el **SUB.sps**, tendrá que ser seleccionado y arrancado por primera vez mediante intervención del operador.

En el momento que el programa es seleccionado, habrá que realizar el arranque en RUN del programa mediante la comunicación con el PLC. A través de la Interface del Automático externo procederemos al rearme de los accionamientos del robot y la confirmación de mensajes, tal como se desarrolló en los procedimientos anteriores.

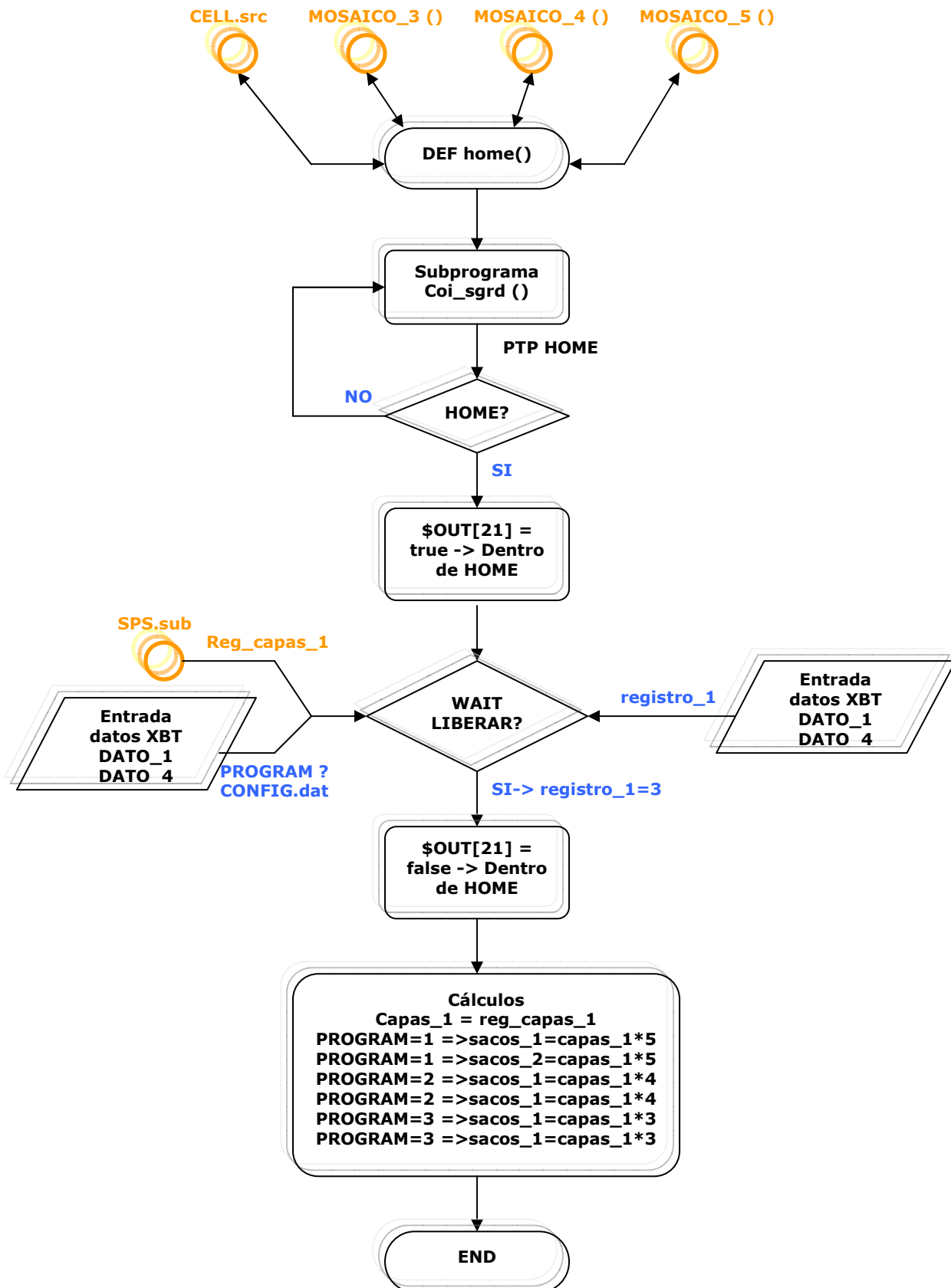
Una vez que el programa esta en RUN, se inicializa la cabecera con la declaración de los programas a usar por **CELL.src**, seguidamente se hace es una coincidencia de trayectorias COI, y la selección del programa de paletizado que queremos usar. Para ello se llama al proceso **home ()**.



Dentro de Home se llama al subproceso **coi\_sgr ()** que sitúa al robot en un punto seguro en espera de la recepción de los datos de paletización. Finalizado el proceso **coi\_sgr ()** se activará la marca correspondiente indicando la posición de HOME al PLC. En este momento podremos transmitir desde el terminal XBT el valor de programa que será transmitido por el PLC mediante las variables PGNO\_BIT y convertido por la variable de tipo signal PROGRAM en el **CONFIG.dat**. Después de transmitido el programa a seleccionar y las alturas totales que captura el proceso **home ()** a través del programa multitarea, esperaremos en dicha posición a ser liberado desde el terminal.

Cuando el robot es liberado se calculan las variables enteras **sacos\_1** y **sacos\_2** que contienen la cantidad de sacos totales que queremos paletizar en las dos líneas por igual.

Seguidamente se entra en el bucle de ejecución infinito, en donde se activa el programa de Interface **P00.src** que contiene las funciones necesarias para realizar la comunicación con el PLC, en el intercambio de datos y a través de las líneas destinadas para ello. De esta manera podremos realizar una nueva petición y envío de datos en cada cambio de programa. Dentro de este mismo bucle se ejecuta la función **CASE (Switch)**, en donde en función de la variables **PGNO\_BIT\_1**, **PGNO\_BIT\_2** y **PGNO\_BIT\_3** se ejecutará el proceso **Mosaico\_5 ()**, **Mosaico\_4 ()** o **Mosaico\_3 ()**. Estos procesos a su vez contienen un bucle de ejecución infinita con la posibilidad de salir de ellos en caso de petición de un nuevo programa como veremos más adelante.

**6.2.3.3 Home.src.**



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

El subprograma **home ()** es un proceso desde el que tienen acceso los programas principales para la petición de cambio del número de alturas o del programa a ejecutar. También tiene acceso **cell.src** en el proceso de arranque.

Lo primero que se ejecuta es el subproceso **coi\_sgrd ()** para la coincidencia de un punto seguro de trayectoria. En el momento que el robot adquiere la posición de Home, activará la señal **\$OUT[21]** para notificárselo al PLC, y pasará al modo de espera hasta el envío de la información necesaria y la posterior liberación.

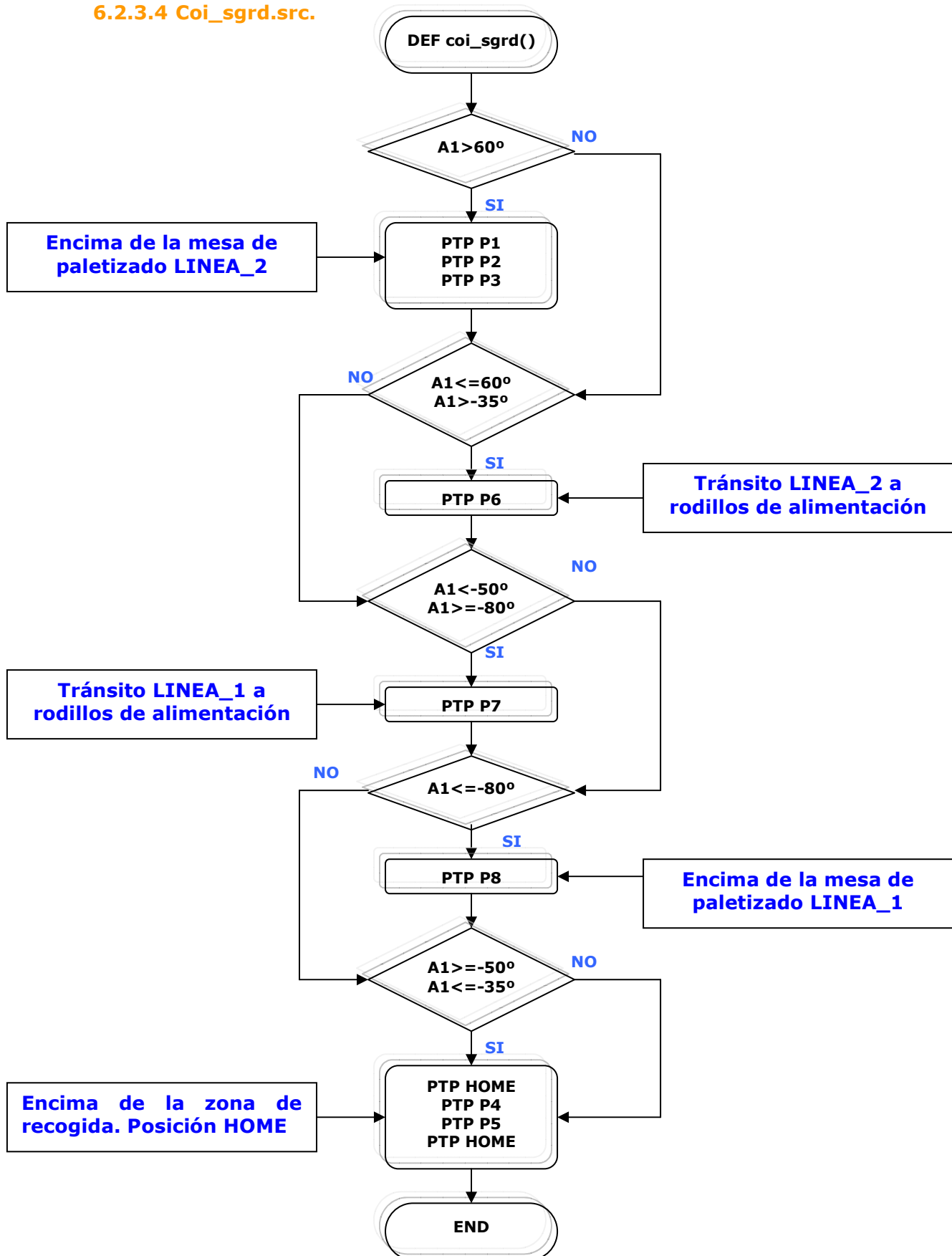
En el modo de espera, enviaremos el número de alturas desde el terminal XBT, en caso de no enviar ninguna altura, en la variable **reg\_capas\_1** capturada en el programa multitarea, estará guardado el último valor enviado. Antes de liberar, también tendremos que enviar desde el XBT el programa de mosaico que queramos realizar. Dicho programa quedará guardado en la variable **PROGRAM** del **Config.dat** que podremos usar en cualquier momento.

Una vez enviado el tipo de mosaico y el número de alturas adecuado, liberaremos al robot desde la función provista en el XBT para tal fin. La condición de liberación pone la variable **registro\_1 = 3**. Además, al ser una orden de liberación, las entradas del robot adquieren los valores necesarios para tal fin, siendo la condición de liberación la siguiente línea de código:

```
wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and (registro_1==3)).
```

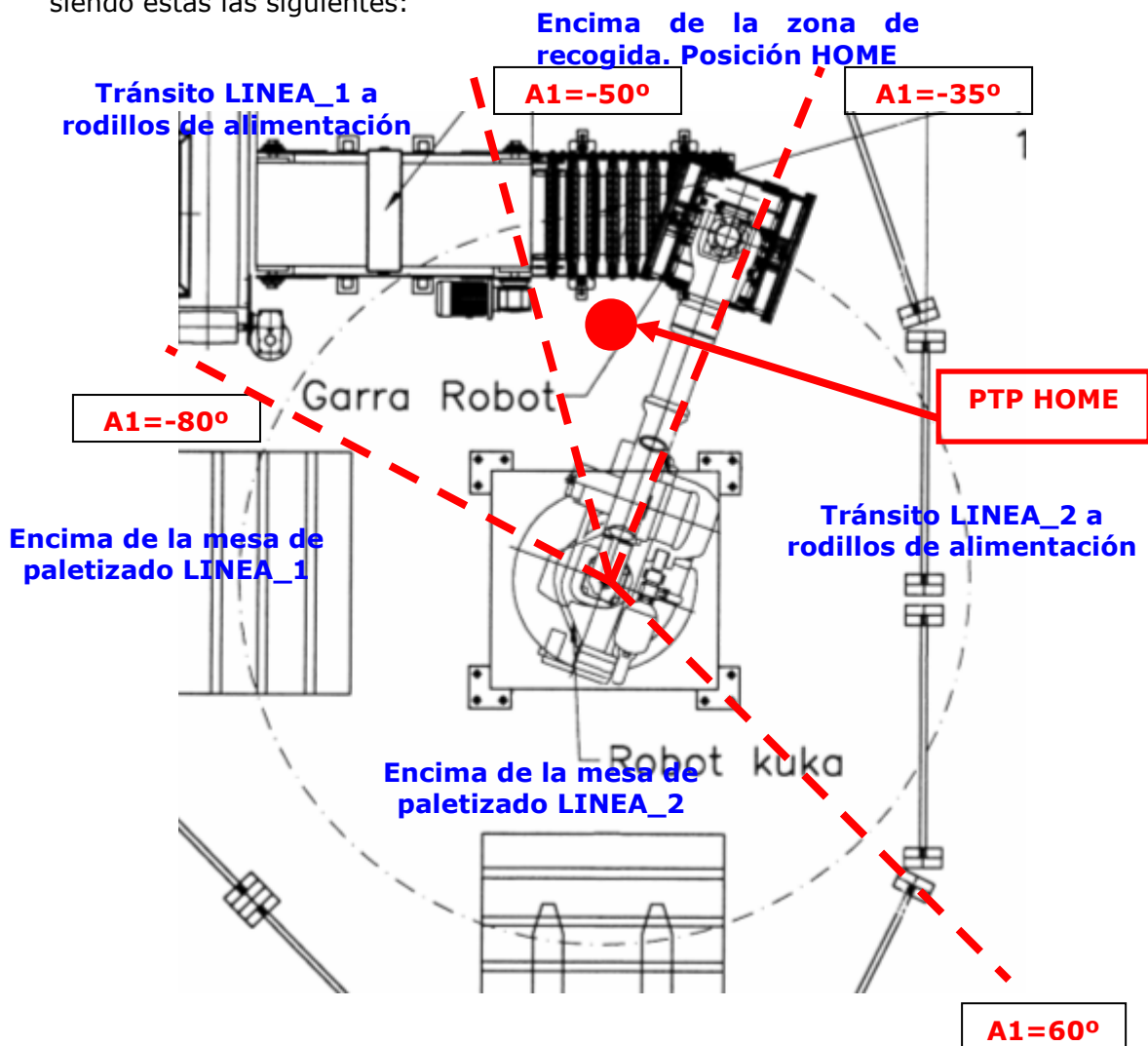
Una vez cumplida la condición de liberación pondremos a false la señal **\$OUT[21]** puesto que el valor de programa ya ha sido transmitido, y pasaremos a realizar los cálculos totales de las variables **sacos\_1** y **sacos\_2** que determinarán el número máximo de sacos a paletizar en el palet antes de su finalización. Estos valores quedarán actualizados en el fichero **Config.dat**.

6.2.3.4 Coi\_sgrd.src.



El subprograma **coi\_sgrd.src** siempre será llamado desde el proceso **home()**, aunque también será llamado en el arranque de los programas de paletizado principales **MOSAICO\_3()**, **MOSAICO\_4()** y **MOSAICO\_5()** por seguridad.

Lo que se va realizando es una comprobación del eje principal **a1** del robot, de manera que en función de su posición se le va llevando a una serie de puntos hasta alcanzar el punto HOME de destino. Para ello se recorren todas las zonas siendo estas las siguientes:

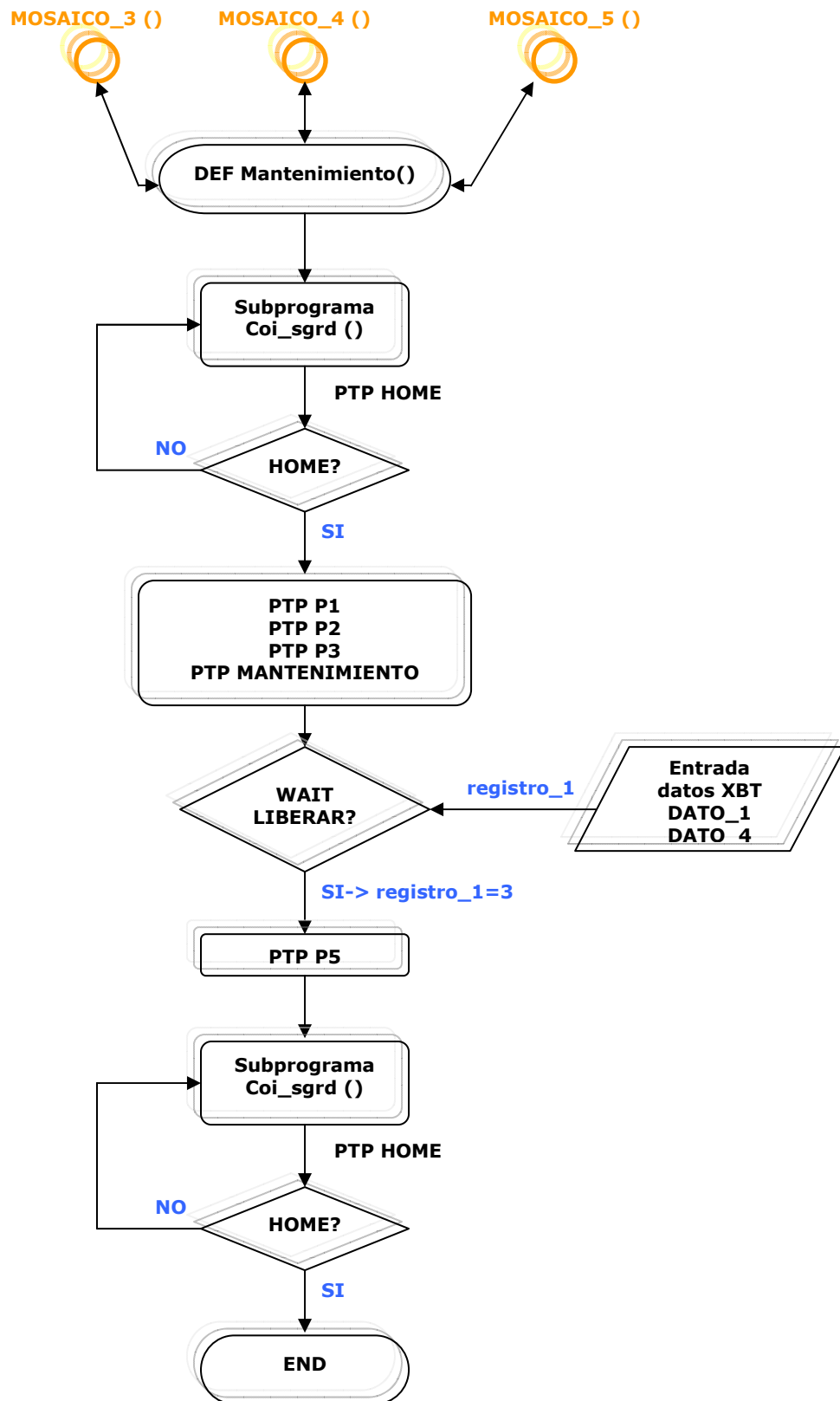




Para el caso de estar encima del palet de la línea 2, el ángulo del eje principal es **a1 > 60°** en este caso se genera la trayectoria **PTP P1, P2 y P3** para situar al robot en una zona libre de obstáculos. Si el robot se encuentra entre los ángulos **a1 ≤ 60° y a1 > -35°** corresponde a una zona de tránsito a los rodillos de alimentación, por lo tanto se ha de situar al robot en el punto **PTP P6**. De esta manera en función de los grados de desvío del robot se determina una posición auxiliar hasta alcanzar el punto exacto de reposo **PTP HOME**. Los puntos **PTP P4** y **PTP P5** son exactamente el punto home situado por encima y por debajo del mismo, de esta manera cada vez que el robot haga este gesto, le indicará visualmente al operador que se ha alcanzado la posición solicitada.

Seguidamente se continuará con el proceso desde el cual fue llamado **coi\_sgrd ()**.



6.2.3.5 Mantenimiento. src



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

El subprograma de mantenimiento es un sencillo grafo al que tienen acceso los programas principales de paletizado. El robot se sitúa en un punto de acceso donde el operador de mantenimiento tiene la posibilidad de verificar el estado de los componentes de la garra.

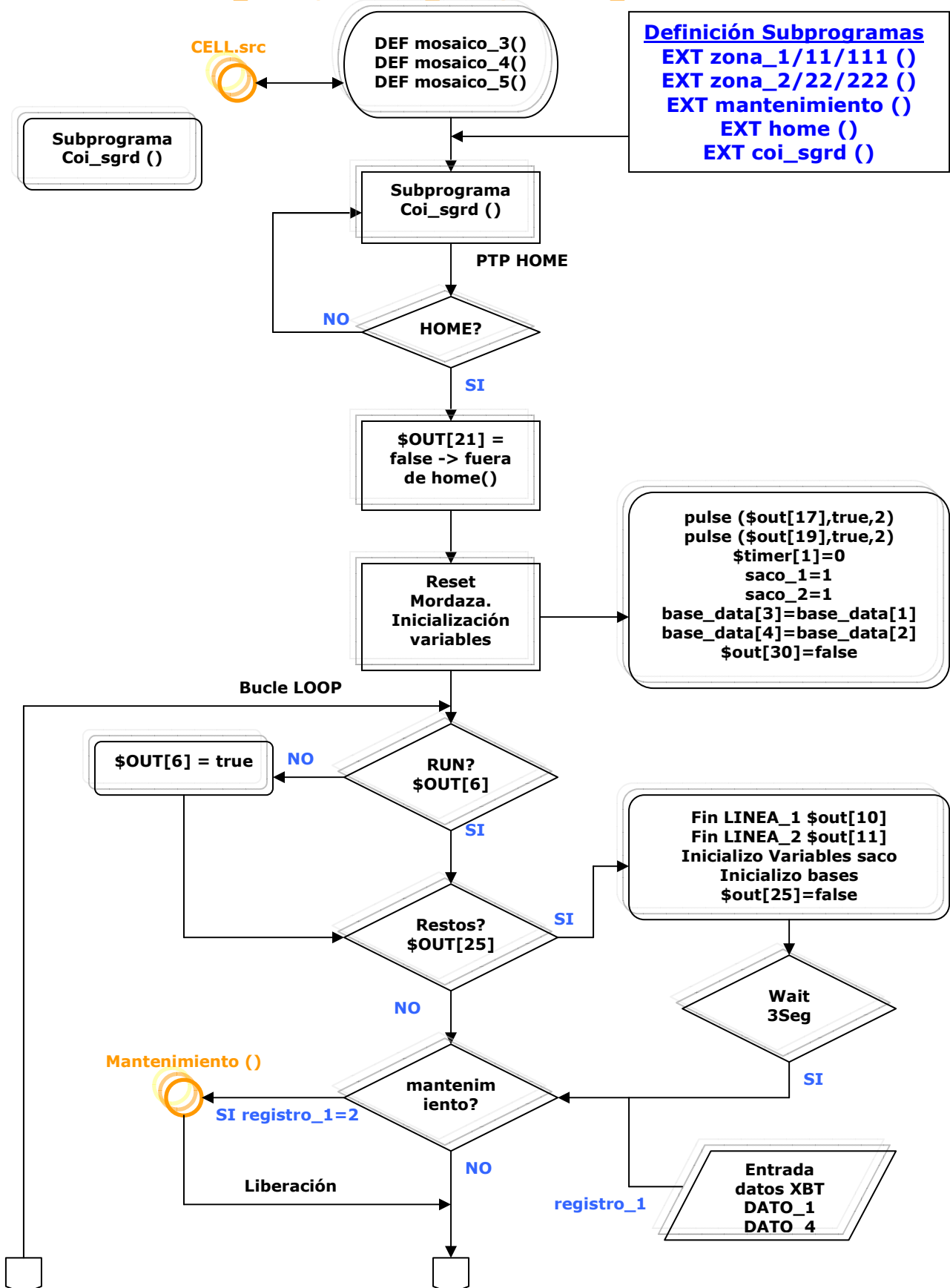
Lo primero que se ejecuta por seguridad es el subproceso **coi\_sgrd ()** para colocar al robot en un punto seguro dependiendo de la situación del eje principal. Una vez situado el robot en su posición de HOME, el robot se moverá a velocidad reducida hasta un punto de acceso dentro de la periferia.

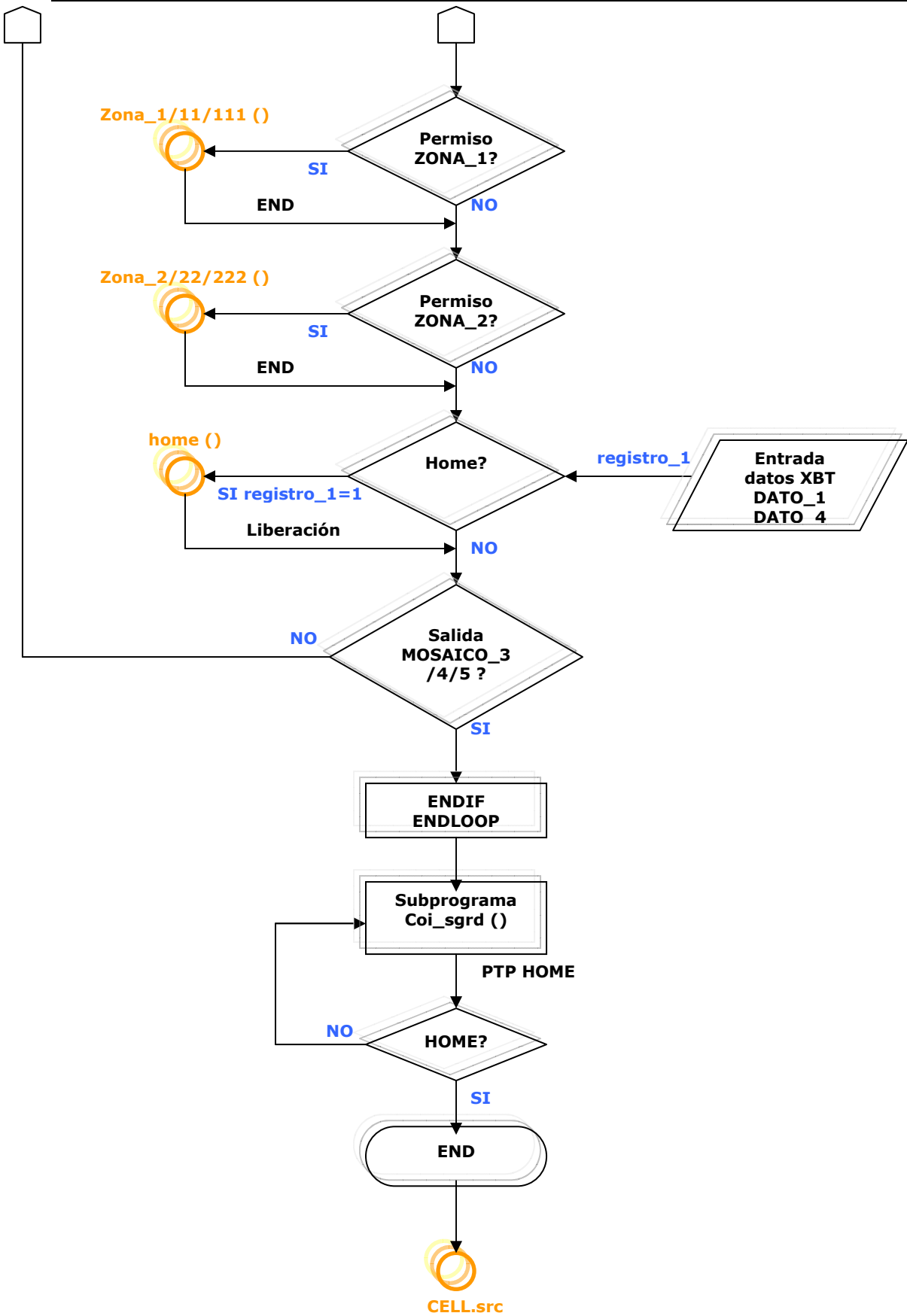
El robot se detendrá y permanecerá en dicha posición hasta que no liberemos desde el terminal XBT. Durante este tiempo el personal cualificado podrá entrar al recinto, saltando las seguridades de la máquina para poder trabajar de forma segura.



Una vez finalizados los trabajos de verificación, habrá que rearmar nuevamente las seguridades de la instalación, y una vez puesta en marcha procederemos a liberar desde el XBT.

Después de liberado, el robot se situará nuevamente en su posición HOME, y continuará con el proceso desde el que fue llamado.

6.2.3.6 Mosaico\_3.src / Mosaico\_4.src / Mosaico\_5.src





	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Los programas principales, son exactamente iguales cambiando los subprocesos de zona. Por lo tanto tendremos:

- **MOSAICO\_3()**
  - Zona\_111 ()**-> Paletizado en la línea 1.
  - Zona\_222 ()**-> Paletizado en la línea 2.
  
- **MOSAICO\_4()**
  - Zona\_11 ()**-> Paletizado en la línea 1.
  - Zona\_22 ()**-> Paletizado en la línea 2.
  
- **MOSAICO\_5()**
  - Zona\_1 ()**-> Paletizado en la línea 1.
  - Zona\_2 ()**-> Paletizado en la línea 2.

Una vez que el proceso es llamado desde el cell.src, lo primero que se ejecuta es el punto seguro de trayectoria por medio de **coi\_sgrd ()** donde si el robot ya se encuentra en el punto HOME, se le notificará al operario mediante los movimientos arriba y abajo.

Seguidamente, y antes de iniciar el bucle de ejecución cíclico, se hace un reset de las variables de programa y de la mordaza esto es:

- **\$out[21]=false**-> Al no estar dentro del subproceso HOME



Para asegurar el estado de reposo de la mordaza.

- **pulse (\$out[17],true,2)** -> Apertura de las manos de la garra.
- **pulse (\$out[19],true,2)** -> Apertura de los dedos de la garra.
  
- **\$timer[1]=0**-> Puesta a cero del temporizador para el defecto de la mordaza.
- **saco\_1=1**-> Inicialización de la variable de posición para la línea\_1.
- **saco\_2=1**-> Inicialización de la variable de posición para la línea\_2.
- **base\_data[3]=base\_data[1]**-> Inicialización de la base en la línea\_1.
- **base\_data[4]=base\_data[2]**-> Inicialización de la base en la línea\_2.
- **\$out[30]=false**-> Reset de la marca de simultaneidad de líneas.

Después de reinicializadas las señales que entraran en juego en el proceso de paletizado, se entra en un bucle de ejecución infinita, en el que permaneceremos hasta que se cumplan las condiciones necesarias para el cambio de programa de paletizado. Dentro de este bucle se evalúan por orden descendente los siguientes procesos:

### **1. Comprobación del estado RUN de programa.**

Con el programa de paletizado en marcha, el KRC pone a true la variable de sistema automático externo **APPL\_RUN** -> **\$OUT[6]** para notificárselo al PLC, y así seguir con lo establecido en el programa del PLC. En algunas circunstancias es posible que el propio sistema no mantenga esta señal a true por posibles conflictos, reinicio inesperados o que el propio operador la haya forzado a false. Para asegurarnos del correcto funcionamiento de esta señal se evalúa su estado **true** o **false**, y en caso de no encontrarse a **true** una vez habilitado el

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

movimiento; esto es **MOVE\_ENABLE->\$IN[7]=true** y con el selector en modo automático externo, se forzará a su estado **true** por programa. En caso de que su estado sea el correcto se continúa con el siguiente proceso.

## **2. Comprobación de la señal de RESTOS.**

En cada ciclo dentro del bucle y antes de comprobar los procesos de zona\_1 y zona\_2, se comprueba si la señal de restos ha sido accionada mediante el valor **\$OUT[25]** que escribe el programa multitarea submit.

Si esta señal ha sido accionada entraremos en la condición IF, donde se finaliza palet en ambas líneas y se resetean las variables de paletizado y bases de referencia de los palet's. Al cabo de un tiempo de espera establecido en tres segundos. El PLC quitará permiso de descarga en ambas líneas hasta que no se pongan unos palet's nuevos, mientras tanto la línea de programa pondrá a false la señal **\$OUT[25]** dejándola libre para una nueva evaluación de la señal de restos en caso de producirse y continuará con su ejecución cíclica dentro del bucle.

Después terminada la rama condicional IF de los restos, o si estos no se han cumplido, se continúa con el siguiente proceso.

## **3. Comprobación del estado ir a Mantenimiento.**



La condición para entrar en el subproceso de mantenimiento, depende única y exclusivamente de la orden generada desde el terminal XBT. Esta orden puede ser ejecutada en cualquier momento del programa. Lo único a tener en cuenta es que si generamos esta orden desde el XBT, en el momento que el robot está ejecutando movimientos de dejada de saco dentro de zona\_1() o de zona\_2 (), es posible que cuando el robot salga de la zona y vuelva a ver la condición IF de entrada a mantenimiento, el PLC; que sólo mantiene un tiempo las señales **\$IN[17], \$IN[18], \$IN[19]** y **\$IN[20]**, ya habrá puesto estas señales a false por lo que no entraríamos dentro de la condición. Lo mejor para ejecutar el proceso de mantenimiento, es que la línea de programa del KRC se mantenga dentro del bucle principal, para lo que podemos cancelar los permisos de entrada a las zonas mediante los selectores, o evitamos la entrada de sacos en los rodillos de alimentación para que no haya permiso de carga, de este modo el robot se mantendrá parado esperando alguna orden y en cuanto ejecutemos la orden desde el terminal se cumplirán las señales poniendo la variable **registro\_1** que adquirirá el valor "2" entrando dentro del proceso de mantenimiento.

Liberado el robot de su estado de mantenimiento desde el XBT, o si no se cumple la condición de entrada en mantenimiento, el programa continúa con la siguiente evaluación.

## **4. Entrada a zona de paletizado en LINEA-1.**

La condición para entrar en el subproceso **Zona\_1 ()**, **Zona\_11()** o **Zona\_111()** dependiendo del programa de paletizado que se este ejecutando. Es la siguiente:

- **\$out[25]==false->** La señal de restos no debe estar activada por seguridad.
- **\$in[11]==true->** El PLC debe dar Permiso de carga.
- **\$in[13]==true->** El PLC debe dar permiso de descarga en la línea 1.
- **\$out[30]==false->** La señal simultaneidad de líneas ha de estar desactivada.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Si esto se cumple, el robot entrará en la zona de paletizado. Cada vez que se entra en la zona de paletizado, se recoge un saco de los rodillos de entrada y se deposita en el palet correspondiente incrementando la variable `saco_1`, de manera que la próxima vez que entre dentro de la zona el saco lo dejará en la siguiente posición en el palet.

Finalizada las acciones de recogida del saco y dejada en el palet, o en el caso de que no se cumplieran las condiciones anteriores, se procede a continuar con el siguiente proceso en ejecución dentro del bucle.

### **5. Entrada a zona de paletizado en LINEA-1.**

La condición para entrar en el subproceso **Zona\_2 ()**, **Zona\_22()** o **Zona\_222()** dependiendo del programa de paletizado que se este ejecutando. Es la siguiente:

- **`$out[25]==false`**-> La señal de restos no debe estar activada por seguridad.
- **`$in[11]==true`**-> El PLC debe dar Permiso de carga.
- **`$in[14]==true`**-> El PLC debe dar permiso de descarga en la línea 2.
- **`$out[30]==true`**-> La señal simultaneidad de líneas ha de estar activada.

Si esto se cumple, el robot entrará en la zona de paletizado. Al igual que en el caso anterior se recoge un saco de los rodillos de entrada y se deposita en el palet correspondiente incrementando la variable `saco_2`, de manera que la próxima vez que entre dentro de la zona el saco lo dejará en la siguiente posición en el palet.



Finalizada las acciones de recogida del saco y dejada en el palet, o en el caso de que no se cumplieran las condiciones anteriores, se procede a continuar con el siguiente proceso en ejecución dentro del bucle.

### **6. Entrada a HOME para solicitud de cambio de alturas o de Programa.**

La entrada a Home, la realizaremos cada vez que queramos cambiar el mosaico de paletizado, o sin cambiar el programa de paletizado pero modificando la altura máxima a paletizar. Será por tanto como venimos comentando, que las líneas de paletizado estén vacías, esto es necesario bien para que se empiece el nuevo mosaico desde la primera altura, o bien para no descuadrar la posición de los sacos y cambiar las altura del mismo programa de paletizado. La condición será por lo tanto la siguiente:

- `$in[17]==false`**
  - `$in[18]==false`**
  - `$in[19]==true`**
  - `$in[20]==false`**
  - `registro_1==1`**
  - `saco_1<2`**
  - `saco_2<2`**
- } Orden de ir a HOME desde el terminal XBT.

Las variables de posición de saco `saco_1<2` y `saco_2<2` indica que su valor solamente puede ser cero o uno, de manera que esto indica que el palet está vacío. Bien por una finalización de palet, o bien por el accionamiento de restos, de manera

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

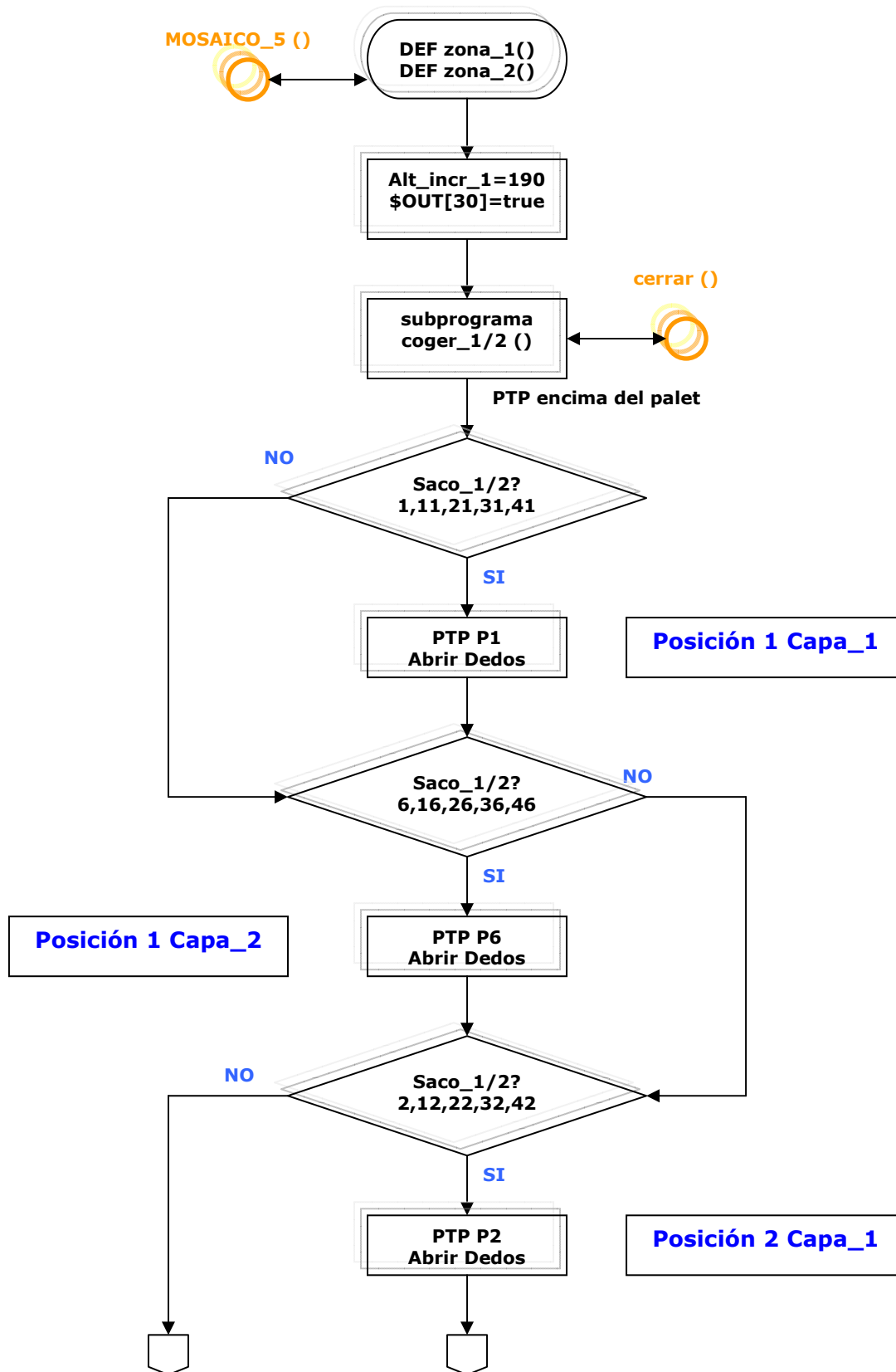
que siempre será preciso sacarlo y generar un nuevo palet mediante los procedimientos establecidos.

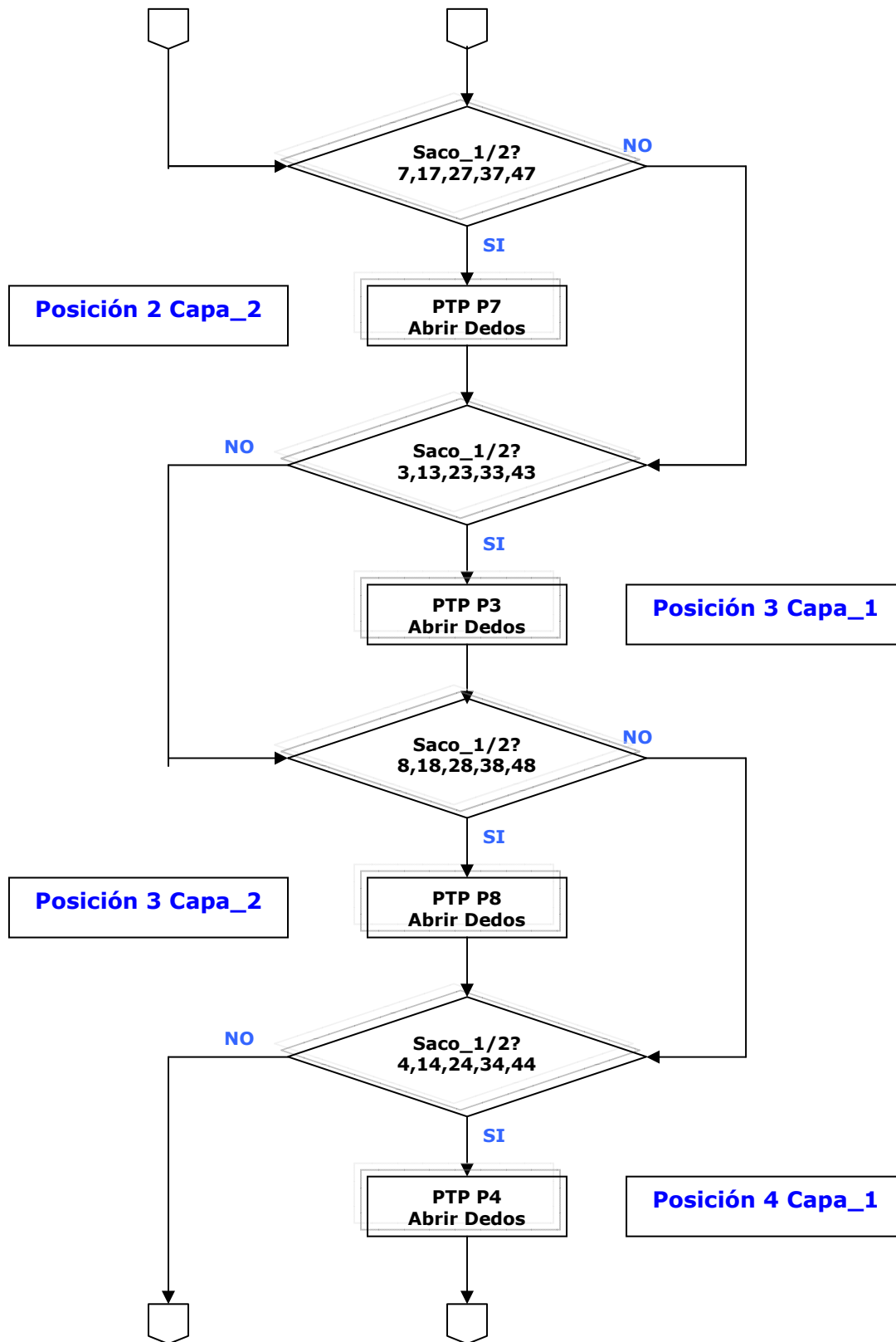
La entrada a HOME se realiza como último proceso del bucle de ejecución, de manera que si al salir de HOME hemos cambiado la variable **PROGRAM**, no coincidiendo con el programa en ejecución actual, se saldrá del bucle mediante la función **EXIT**, no volviendo a paletizar en las zonas correspondientes al mosaico actual. A parte de tener en cuenta el cambio de programa en la salida condicional del bucle, se precisará que no entren sacos en los rodillos de alimentación; es decir, no tener permiso de carga.

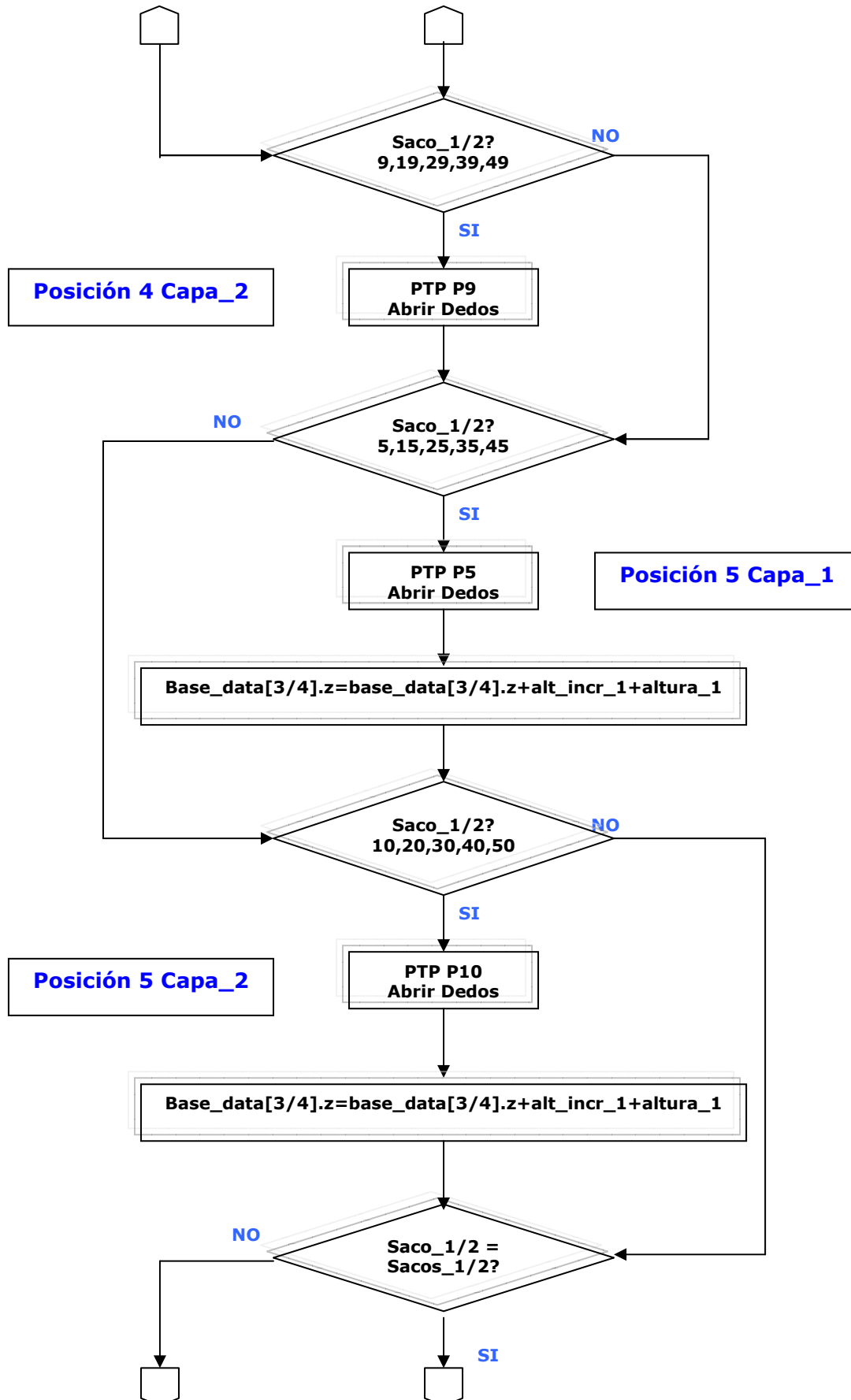
Una finalizado el bucle, se vuelve a ejecutar una coincidencia de trayectoria, de manera que cuando el robot ejecute los movimientos arriba y abajo del Home, indicará al operario que este ha salido del programa actual y a entrado en el nuevo. En este momento se admitirá nuevamente la entrada de palet's pero no tendremos permiso de descarga hasta que el operario quite los palet's finalizados con el mosaico anterior.

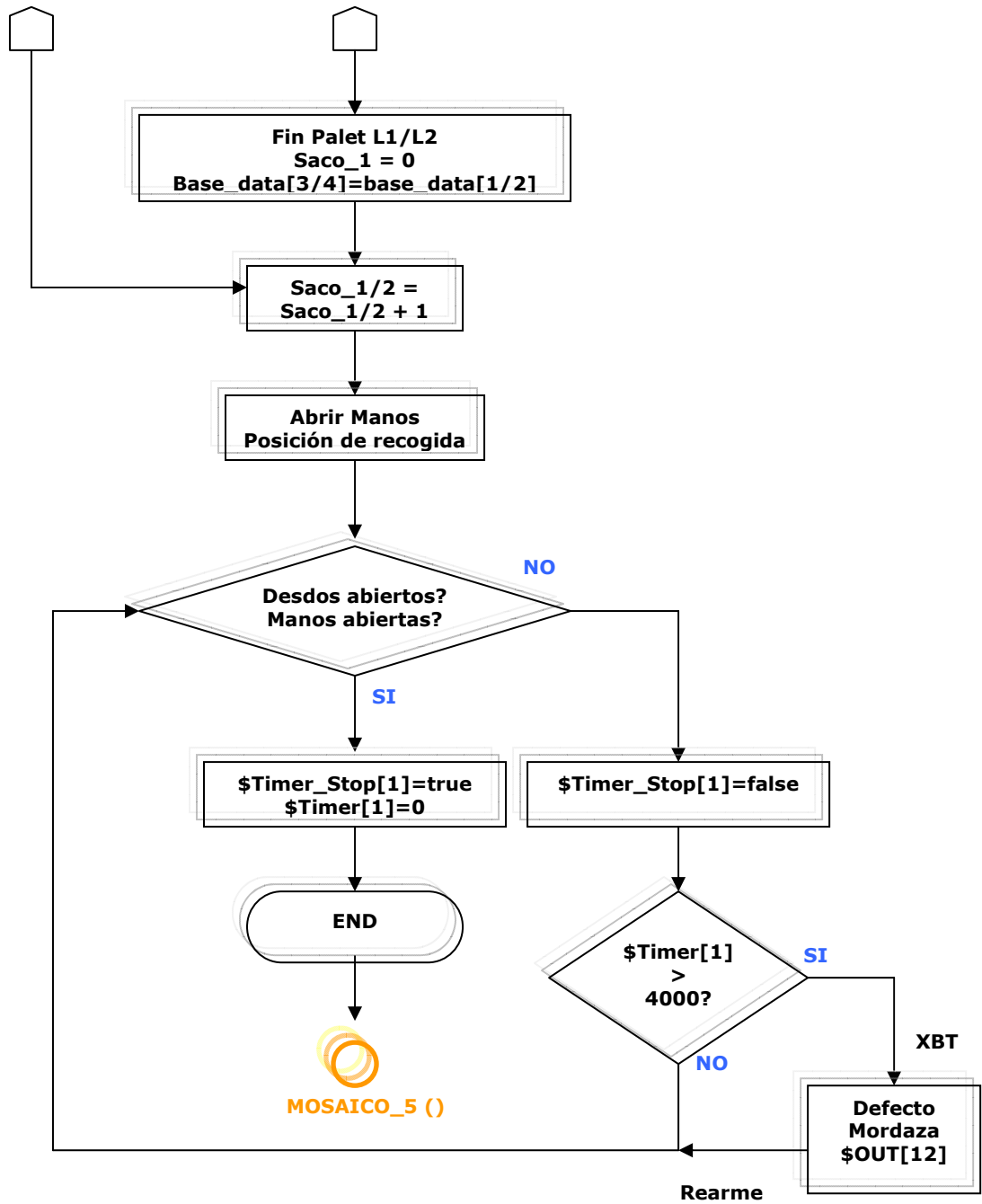




6.2.3.7 Zona\_1.src / Zona\_2.src









	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

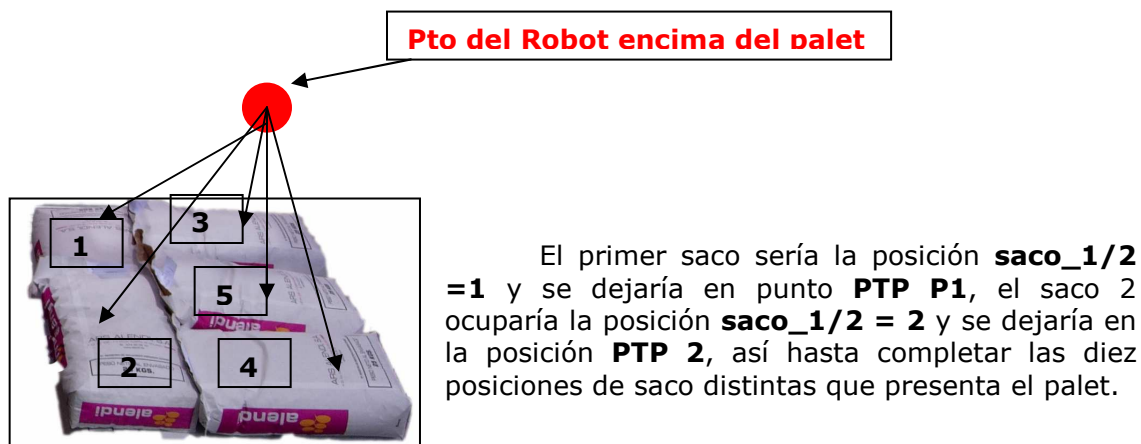
**Zona\_1 ()** es el programa de paletizado con mosaico de cinco sacos en la línea 1 y **Zona\_2 ()** el de la línea 2 que es exactamente idéntico cambiando la variable de posición de saco, fin de palet, etc. Pero funcionalmente son idénticos, por lo que se explican ambos diagramas en conjunto. Centrándonos en zona\_1/2() que será llamado desde el bucle infinito del programa principal MOSAICO\_5.

- **Zona\_1/2 () => Zona\_1() o Zona\_2 ()**.
- **Saco\_1/2 => Saco\_1 o Saco\_2**.
- **Sacos\_1/2 => Sacos\_1 o Sacos\_2**.
- **base\_data[3/4] => base\_data[3] o base\_data[4]**.
- **coger\_1/2 () => coger\_1 () o coger\_2 ()**.



Lo primero que se hace en estos procesos, es asignar un valor a la variable **alt\_incr\_1** del archivo config.dat, para el caso que nos ocupa **alt\_incr\_1=190** en milímetros será la altura base a incrementar en la coordenada Z de la base relacionada al palet; **base\_data[3]** para la línea 1 y **base\_data[4]** para la línea 2, por cada altura completada.

Seguidamente se ejecuta el subproceso **coger\_1 () o coger\_2 ()**. Mediante la ejecución de este subprograma, el robot ejecuta las trayectorias que van desde la mesa de rodillos hasta un punto situado encima del palet de la línea 1, y a partir del cual se posiciona el saco en el palet.

Como hemos visto en apartados anteriores, para el mosaico de 5 sacos por altura, existen 2 capas diferentes; esto significa, que tendremos 10 posiciones diferentes de sacos a partir de las cuales se irán repitiendo con el incremento en Z correspondiente. Si vemos un ejemplo para la capa 1 del mosaico de 5 sacos, cada saco tiene una posición que sería la siguiente:



Una vez posicionados los diez primeros sacos, el saco número once que ocupará la posición **saco\_1/2 = 11** corresponderá con el saco nº 1 de la altura 3, y por lo tanto su punto de dejada será el mismo que el punto **PTP P1** de la altura uno. Cada cinco sacos tendremos que incrementar el valor constante **alt\_incr\_1**, esto será como veremos, en los sacos múltiplos de 5; es decir **saco\_1/2= 5,10,15,20,25** y así hasta el final del palet. De esta manera vemos que los procesos de zonas se componen de condiciones IF dependientes de la variable de posición **saco\_1** o **saco\_2** la cual se irá incrementando por cada saco dejado en el palet.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Una vez que el robot esta posicionado encima del palet con el saco entre sus garras, se evalúa el valor de la variable **saco\_1/2**, de manera que solamente se ejecutará la condición **IF** correspondiente a este valor. Para posicionar un saco el robot ejecutará el punto **PTP (referenciado a la base\_data[3] en la línea 1 o base\_data[4] en la línea 2)** para posicionarse justo encima de donde debe soltarse el saco, seguidamente antes de abrir los dedos del robot se hará un movimiento relativo en Z de manera lineal a la posición PTP anterior.

Soltado el saco en la posición correspondiente, esperaremos el tiempo que asegura la retirada del robot sin peligro de llevarse el saco que se encuentra en caída. Si el saco depositado es un múltiplo de 5, se incrementará la base del palet, con la constante de altura y el valor configurado a través del XBT.

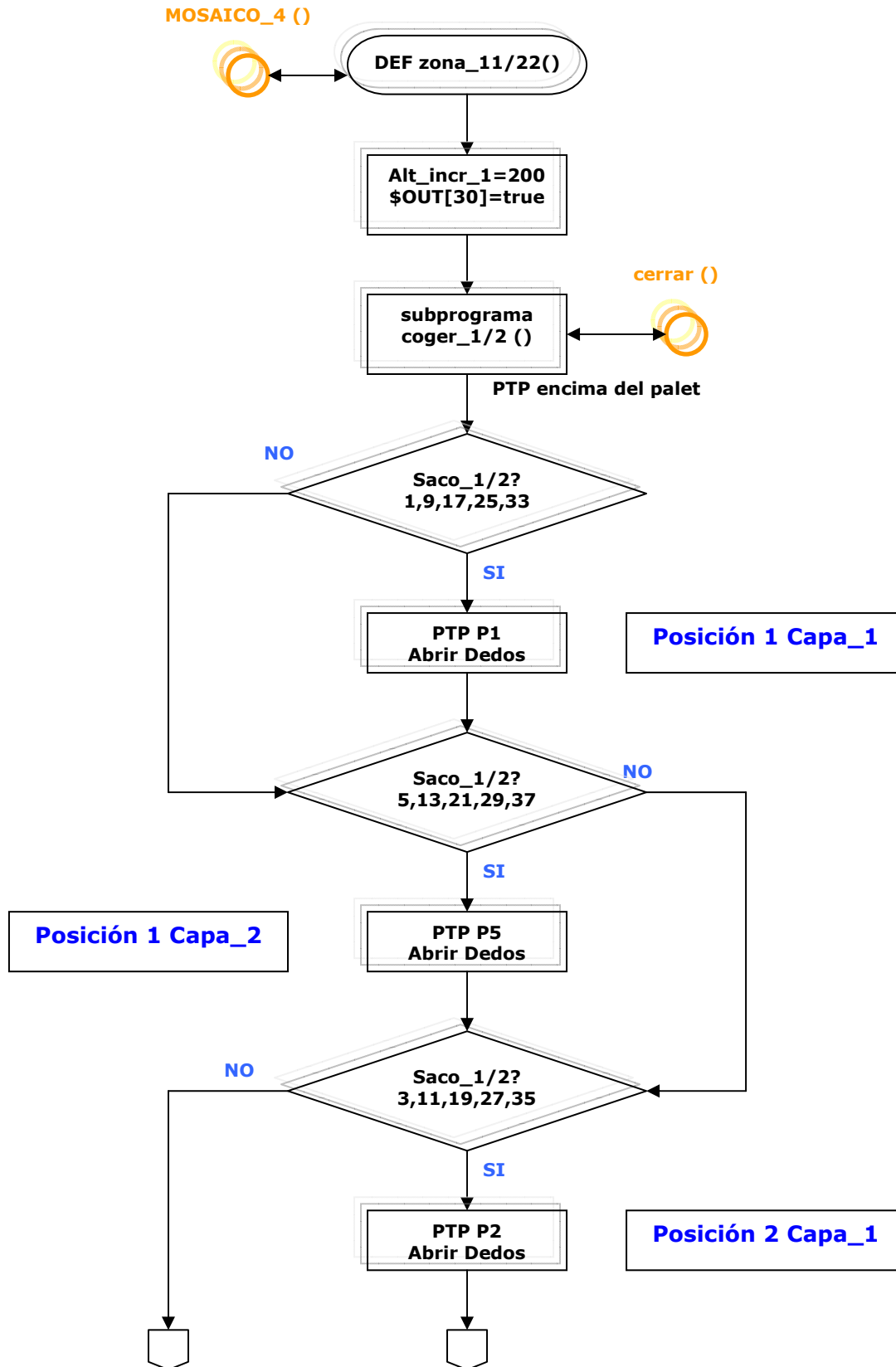
Después de evaluadas las posiciones de saco, se comprobará si el palet esta completo. Esto se hace mediante la igualdad **saco\_1/2 == sacos\_1/2**, de manera que si el valor de la variable **saco\_1/2** coincide con el valor de sacos totales a paletizar **sacos\_1/2** se generará la señal de final de palet y se iniciará la variable **saco\_1/2 = 0**. Se igualará **Base\_data[3] = base\_data[1]** o **Base\_data[4] = base\_data[2]** de manera que el valor de **Z** que se ha ido incrementado adquiere el valor de la variable **Z** de **base\_data[1]** o **base\_data[2]** , que no se modifica en ningún momento y que guarda las coordenadas iniciales de **base\_data[3]** y **base\_data[4]** respectivamente en el config.dat.

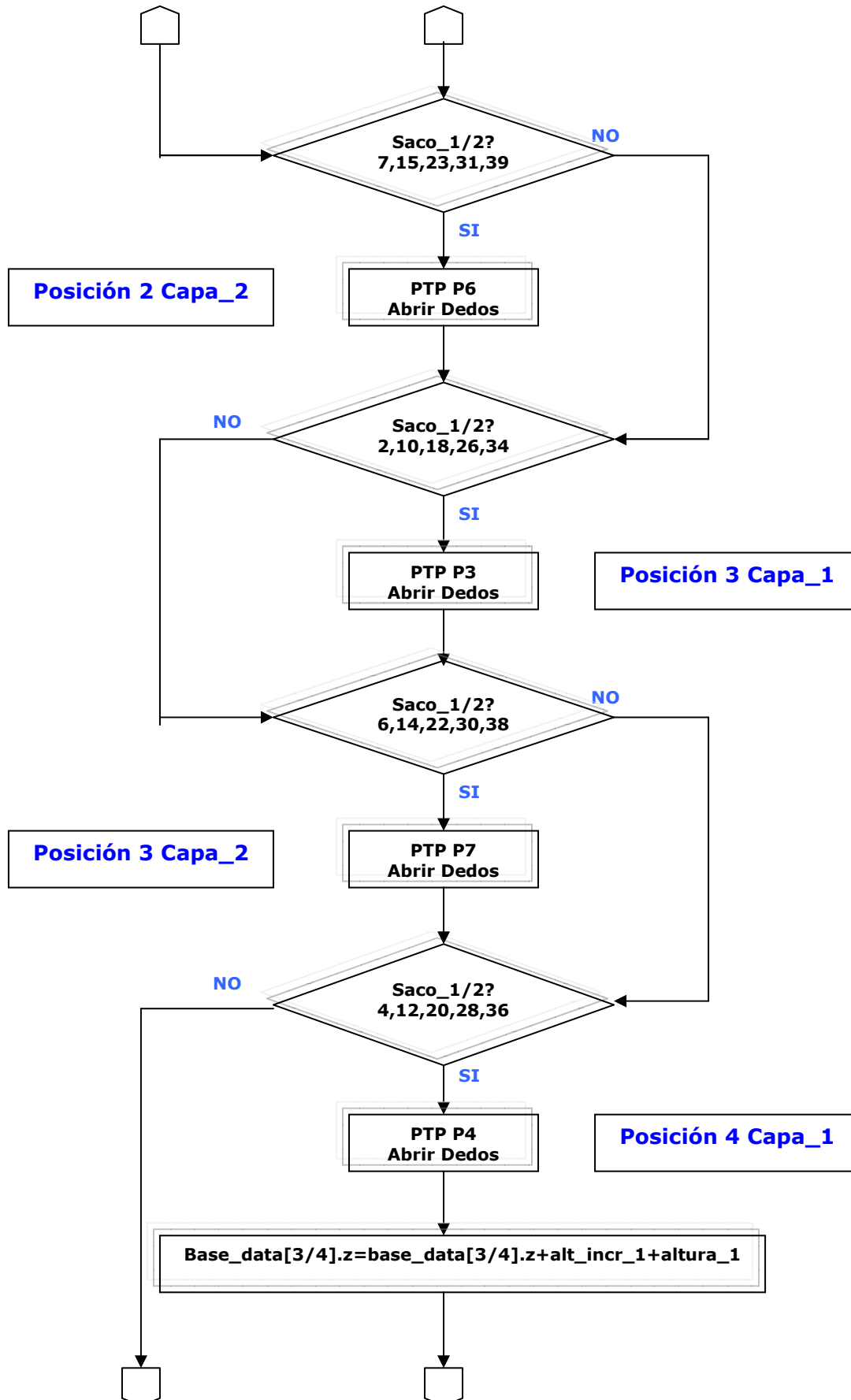
Por último se incrementa el valor de **saco\_1/2**, de esta manera en el caso de que se finalice el palet, la variable **saco\_1/2** adquirirá el valor mínimo de uno, de manera que para el siguiente palet **saco\_1/2 = 1** y entrará en la posición de saco correspondiente.

Las siguientes acciones posicionar el robot nuevamente en la zona de recogida de saco, esto se hace a la vez que se abren las manos en trayectoria mediante la función **SYN PULSE**.

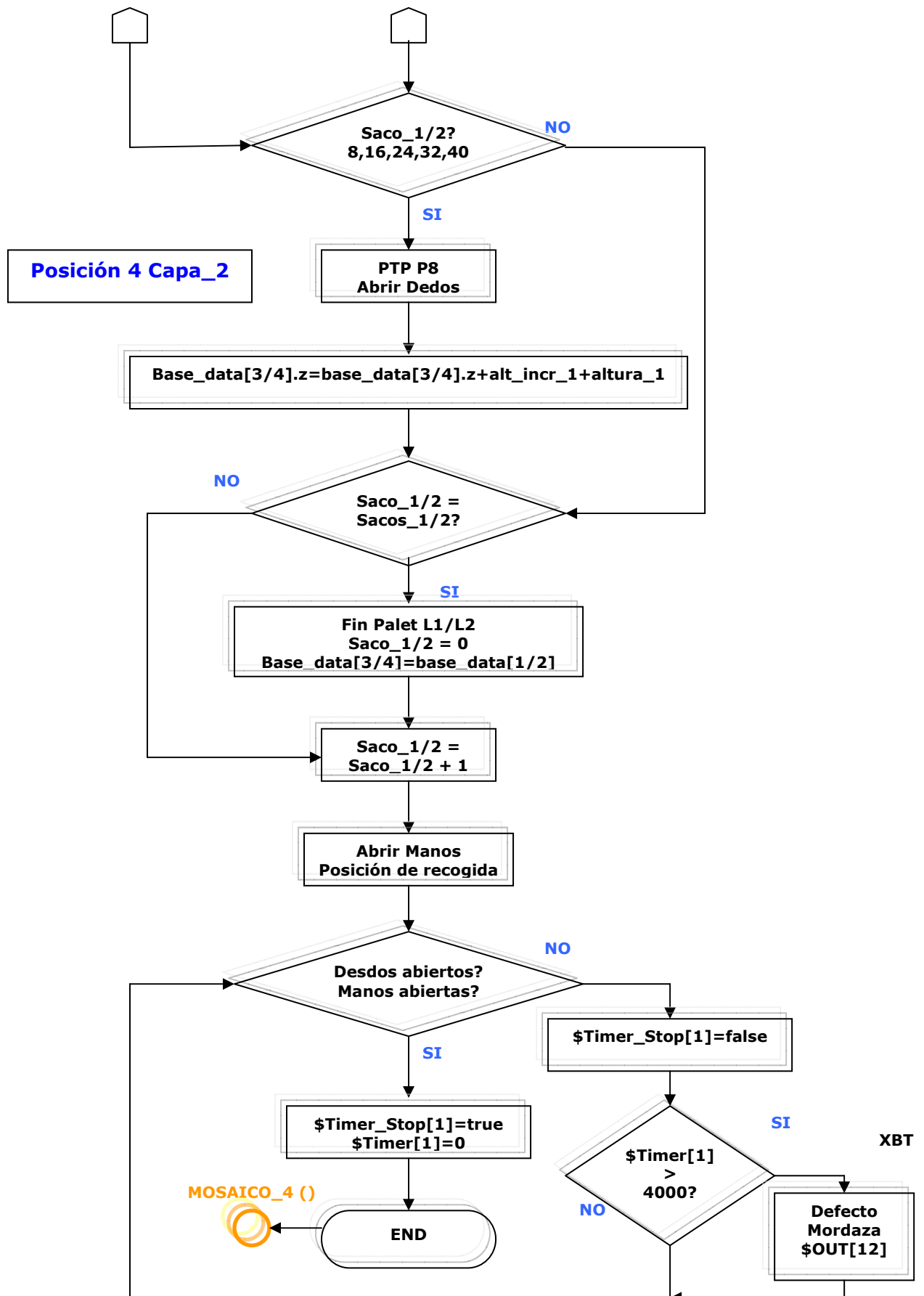
El proceso finalizará con el bucle de ejecución para comprobar el estado de reposo de la mordaza, y no se saldrá de este bucle hasta que los dos detectores magnéticos envían las señales correspondientes para el reposo.

6.2.3.8 Zona\_11.src / Zona\_22.src

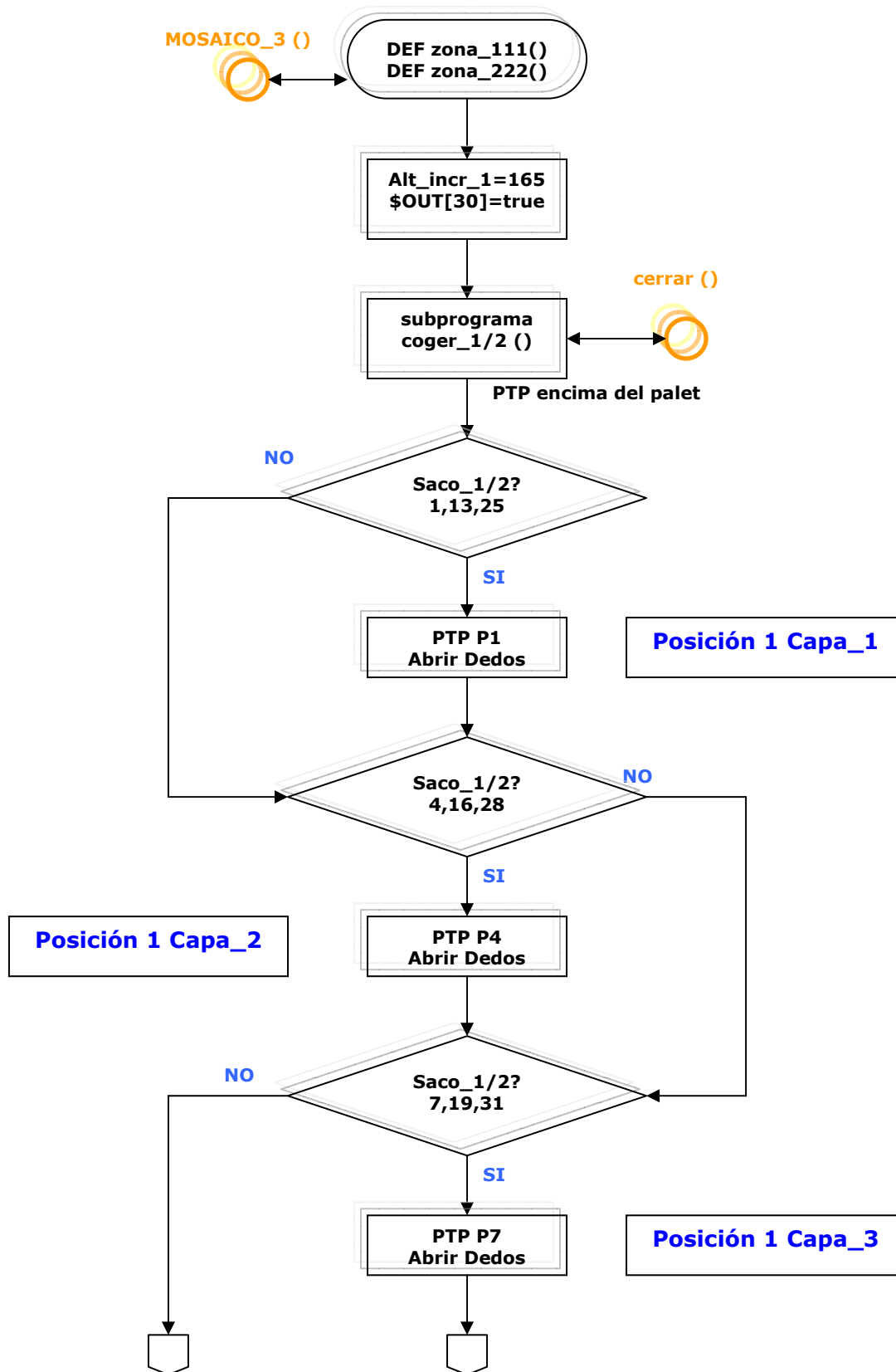


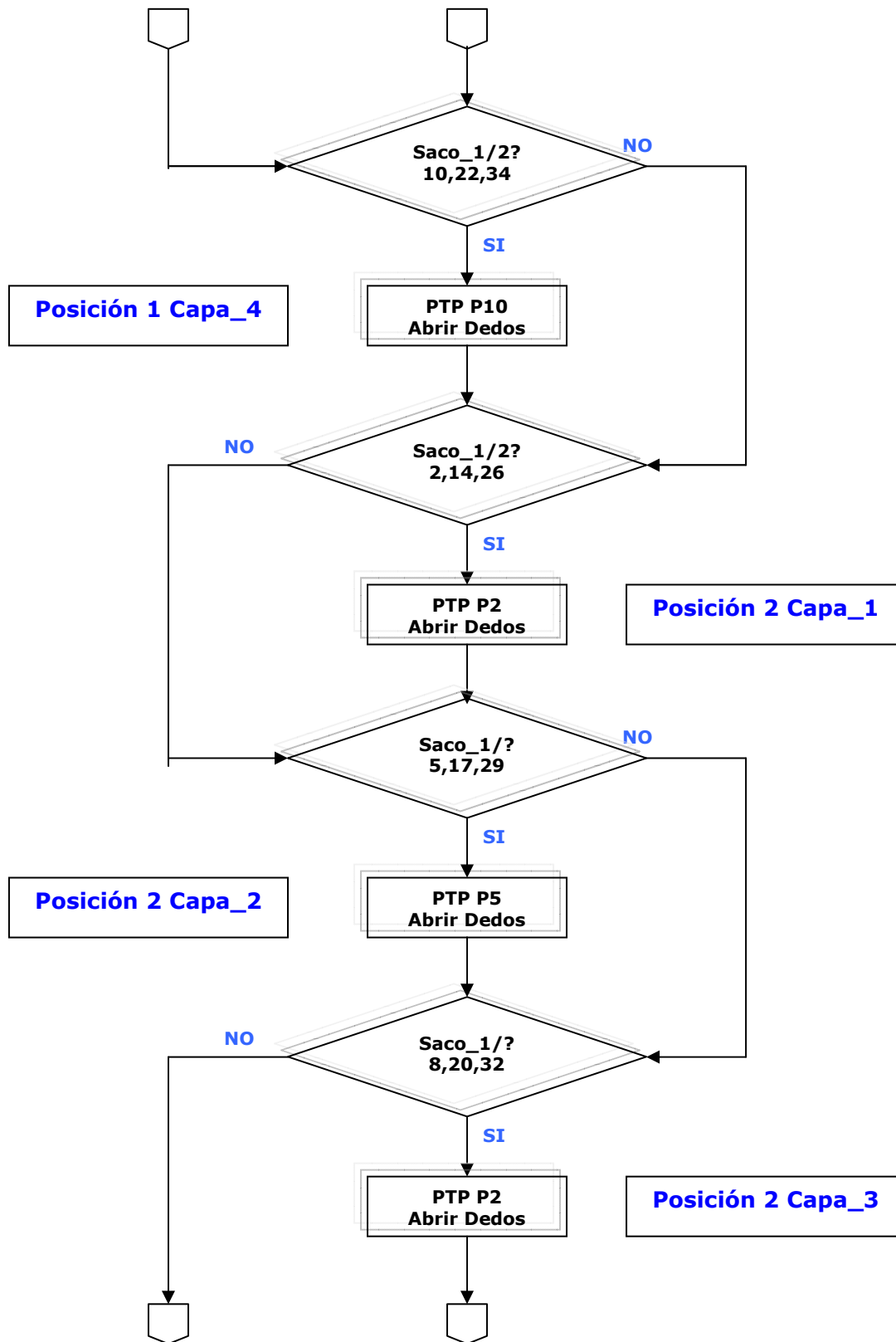


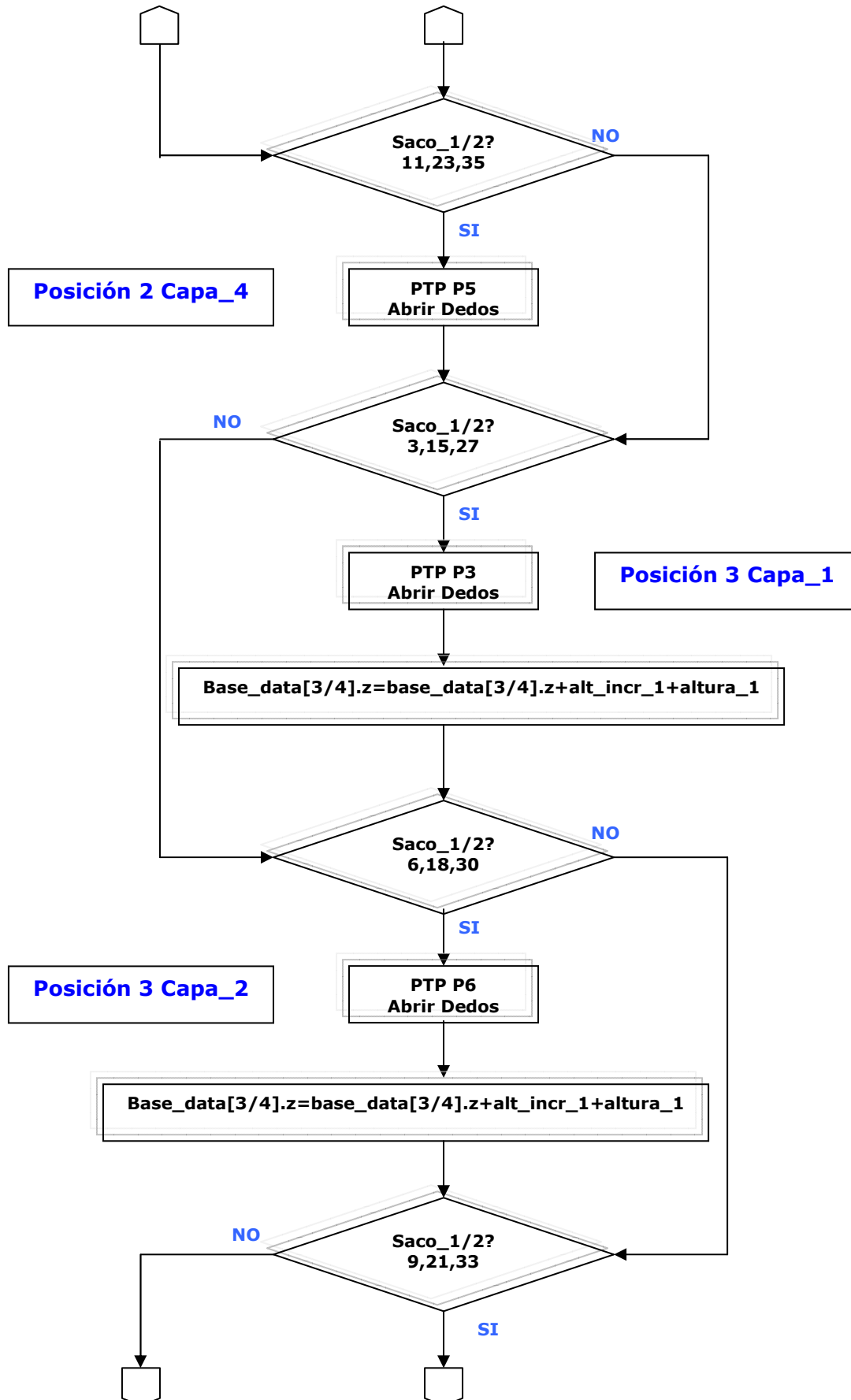


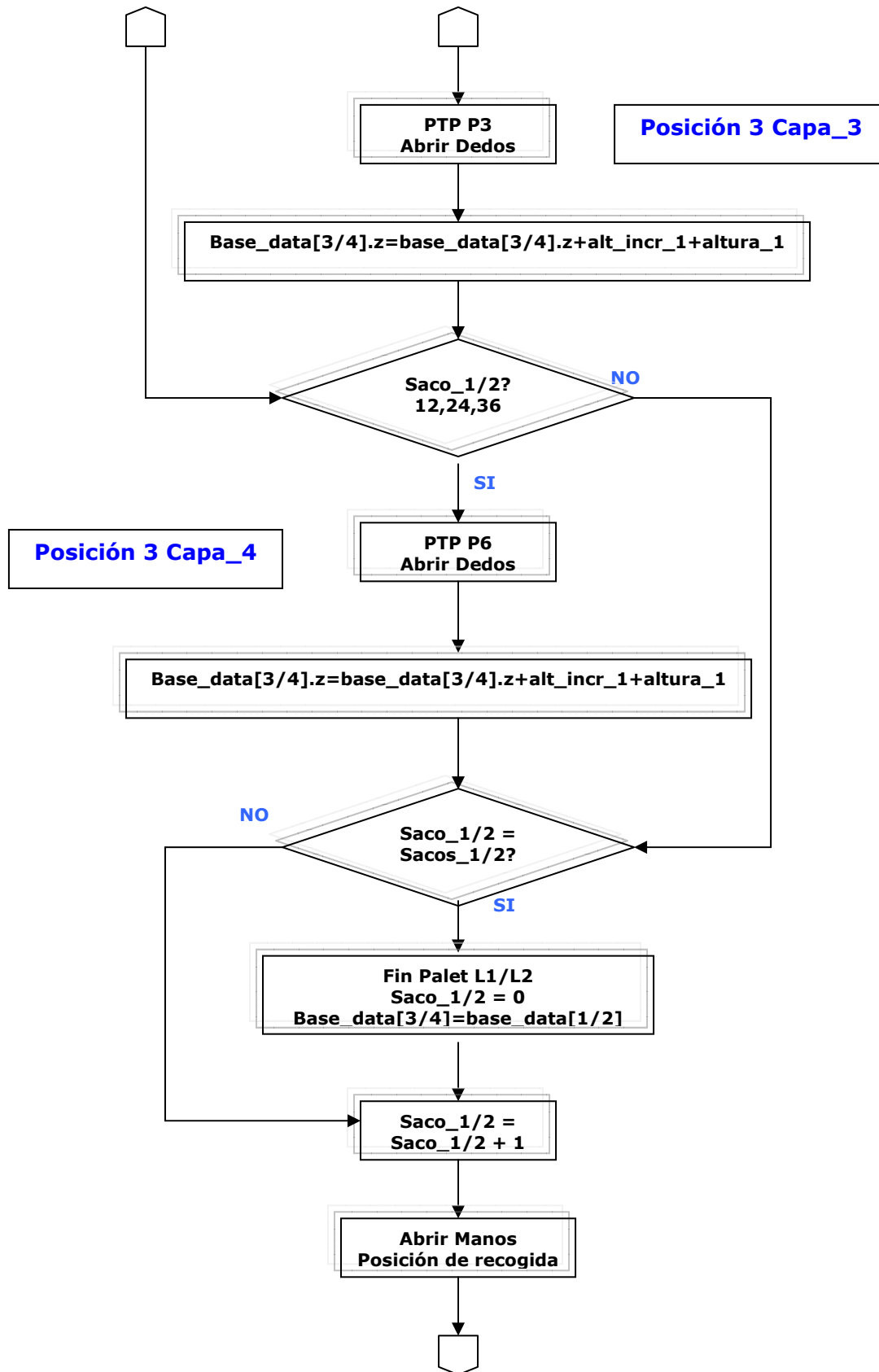


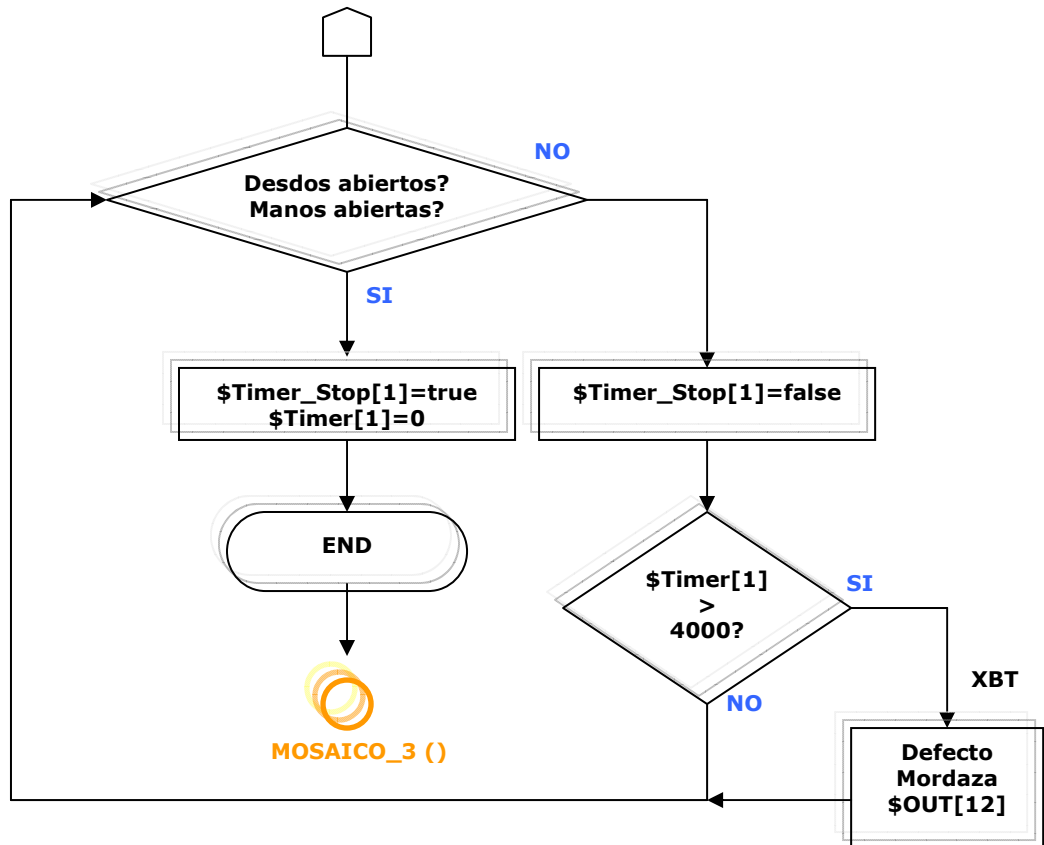
6.2.3.9 Zona\_111.src / Zona\_222.src



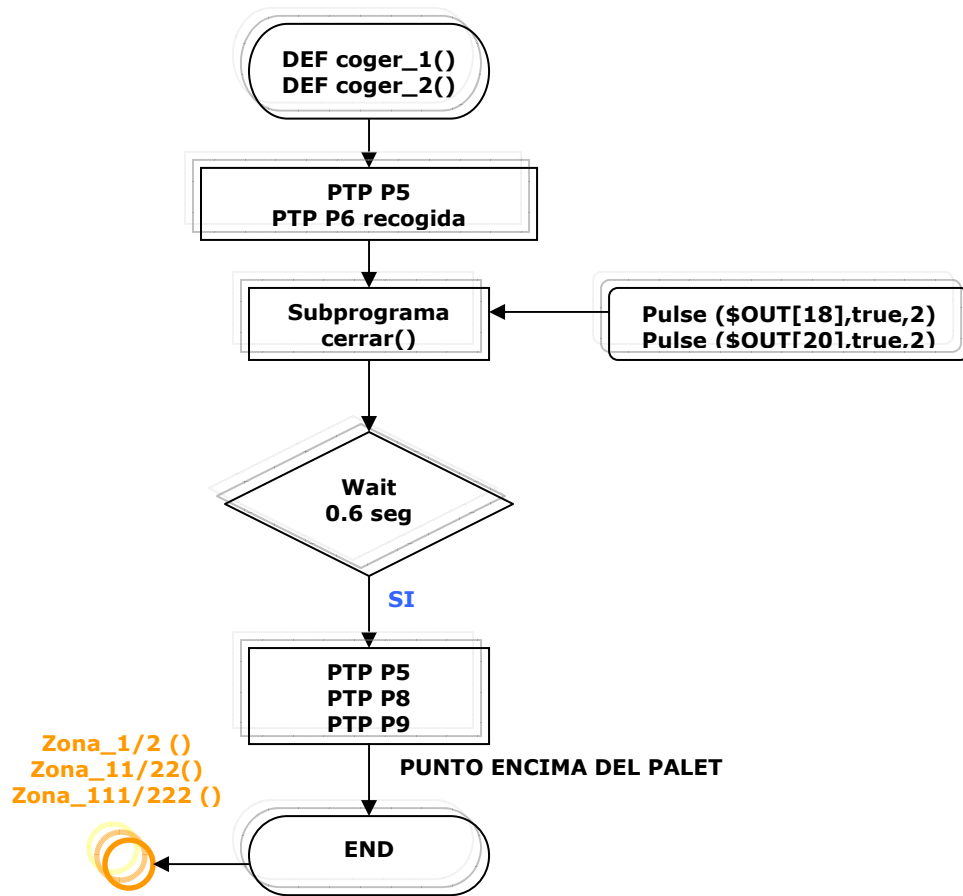










### 6.2.3.10 Cogrer\_1.src / Cogrer\_2.src



Los procesos **cogrer\_1** y **cogrer\_2**, posicionan el robot a la entrada del saco mediante movimientos de velocidad reducida. Una vez posicionado se activarán las salidas para las electroválvulas de cierre de dedos y manos mediante un pulso mantenido de dos segundos, suficiente para la activación de las bobinas de cierre.

Por último se espera 0.6 seg para asegurar el cierre y se posiciona el robot a la altura del palet de la zona correspondiente.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

## 6.3 PROGRAMA TSX 3710.

En este apartado del proyecto, detallaremos los procesos llevados a cabo en el programa del autómatas, finalizando así toda la parte software y funcional de la instalación.

En este caso basaremos el siguiente capítulo en una breve descripción de las variables empleadas, pues su entendimiento nos ayudaran conceptualmente para comprender las diferentes secciones que forman el programa.

Distinguimos en este caso las diferentes variables empleadas en el programa; siendo estas, entradas y salidas físicas, marca de memoria tipo bit, palabras de memoria, y por último los temporizadores usados, aunque estas tres últimas clases estarán referenciadas a la explicación lógica de cada una de las entradas y salidas.

### 6.3.1 Entradas digitales:

En el autómatas distinguimos tres módulos de estradas de 16 bit's conectados por medio de los módulos **10BO** integrados en el armario eléctrico. Cada una de las entradas y salidas se describe con su nomenclatura habitual, junto con nombre de la misma, que aparecerá en el programa y el comentario descriptivo.

- **%I1.0-> Di\_boton\_paro-> Botón paro instalación.**

Esta entrada corresponde con el pulsador de paro del armario eléctrico, empleado en la sección de marcha general, detectaremos mediante flanco de bajada su accionamiento para desactivar la marca de memoria **%M1** (marcha general).

- **%I1.1-> Di\_boton\_marcha -> Botón marcha instalación.**



Como en el caso anterior, este pulsador situado en el exterior del armario eléctrico activará mediante flanco positivo la marca de marcha general **%M1**, y una vez puesta en marcha la periferia, mediante detección de flanco de bajada daremos marcha al robot mediante la marca de memoria **%M4**.

- **%I1.3-> Di\_boton\_reset\_alarmas-> Botón reset alarmas.**

Este pulsador corresponde con el rearme de la instalación, y este será usado para rearmar el módulo de seguridades del armario y la marca de seguridades del robot **%M5** (mediante flanco negativo) una vez no existan mensajes por confirmar en la KCP. También se empleará para rearmar el defecto de mordaza en caso de producirse.

- **%I1.4-> Di\_seguridades -> Entrada módulo de seguridades.**
- **%I1.5-> Di\_presion\_aire-> Entrada presión de aire de la maquina.**
- **%I1.6-> Di\_serie\_termicos-> Serie de los térmicos de la instalación.**



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Estas tres señales de entrada, se utilizan conjuntamente para activar las seguridades generales a nivel de programa, declarada a través de la marca **%M0** necesaria para que todo pueda ponerse en marcha.

Individualmente se utilizan para activar el defecto en el Terminal XBT, de manera que visualizaremos el estado de los térmicos, del presostato y del módulo de seguridad con estas tres señales.

La entrada **%I1.4** ;señal llevada a través de unos de los reles de mando que indica el estado activado del módulo de seguridades, se utilizará también para desactivar la marca **%M5** de seguridad general del robot. De manera que se precisará posteriormente rearmar sus accionamientos y confirmar los mensajes de estado del KCP.

- **%I1.9-> Di\_presencia\_palet\_lleno\_l1-> Presencia de palet lleno en la línea 1.**
- **%I1.10->Di\_presencia\_palet\_lleno\_l2->Presencia de palet lleno en la línea 2.**



Las señales de presencia de palet lleno, nos indican con su estado la existencia de sacos en los palet´s. Esta señal será usada en la sección de permisos, como seguridad para que no se active el permiso de descarga en cada una de las líneas cada vez que ponemos un palet nuevo. De esta manera si hay presencia en el palet cuando se ha producido un cambio de palet, la señal o pulsador de palet nuevo no habilitará el permiso de descarga en la línea hasta que se detecte que se encuentra vacío.

- **%I1.11-> Di\_celula\_recogida\_saco-> Célula de mesa de recogida de saco.**

La señal de recogida de saco, representa la fotocélula de presencia de saco en los rodillos de alimentación. Por lo tanto mediante esta señal se determinará los siguientes estados:

1. Su estado libre, permitirá que la cinta de entrada se encuentre permanentemente en marcha en función de **%M1**.
  2. La señal filtrada de esta célula **%M8** mediante temporizador **%TM2** utilizado en la sección de recogida de saco, desactiva la marcha de los rodillos de alimentación, indicando así la llegada del saco a su posición de recogida.
  3. Su presencia junto con la cinta de entrada, activa la señal de saturación, y por lo tanto el paro de la cinta de entrada de sacos del cliente Piensos O´Couto.
  4. Mediante la detección de flancos positivos, incrementamos el contador de sacos de entrada que podemos visualizar desde el terminal XBT.
- **%I1.12-> Di\_entrada\_saco\_saturacion-> Célula de la cinta de entrada del saco para conseguir saturación.**

Esta señal representa la presencia en la cinta de saturación. Mientras este libre de obstáculo, dicha cinta se encontrará en marcha siempre que lo este la instalación **%M1**. Así mismo si pondrá en marcha la cinta de recogida de saco hasta que **%I1.11** detecte su presencia.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

- **%I1.13-> Di\_robot\_no\_actua\_I1-> El robot no paletiza en la línea de paletizado 1.**
- **%I1.14-> Di\_robot\_no\_actua\_I2 ->El robot no paletiza en la línea de paletizado 2.**

Mediante los selectores para bloquear el paletizado de las líneas, activamos estas señales de mando. Estas señales se evalúan con lógica negada bloqueando directamente los permisos de descarga, en los palet's. Así mismo y cada vez que se requiera un palet nuevo en la línea, será necesario evaluar esta señal previamente, a la vez que se desactivan las barreras de seguridad, de manera que no pueda iniciarse un palet nuevo hasta que se encuentre en su estado de permiso.

- **%I1.15-> Di\_boton\_palet\_nuvo\_I1->Palet nuevo puesto en la zona de paletizado.**
- **%I3.0-> Di\_boton\_palet\_nuevo\_I2-> palet nuevo colocado en la zona de paletizado esta vacío**

Las señales de palet nuevo, se accionan también desde el armario eléctrico de control, y será la última condición necesaria para volver a activar el permiso de descarga del palet nuevo. De esta manera el cambio e inicio del palet nuevo se producirá en las siguientes condiciones:

1. La instalación está en marcha. **%M1.**
2. El palet situado se encuentra vacío.
3. El selector de bloqueo de línea se encuentra activado.
4. Hubo notificación previa de palet terminado.
5. El pulsador de palet nuevo es accionado.

De esta manera se activará la condición previa para obtener el permiso de descarga mediante la marca de memoria **%M42.**

El permiso de descarga se producirá nuevamente en cuanto volvamos a habilitar la línea de paletizado mediante el selector.



- **%I3.1-> Di\_presencia\_rodillo.**

La presencia de rodillo filtrada mediante la memoria **%M9** y a través del temporizador **%TM1** activará la bobina electromagnética para la bajada del rodillo prensa. Este cilindro permanecerá abajo durante el paso del saco, y mientras **%M1** y dicha señal permanezcan activas.

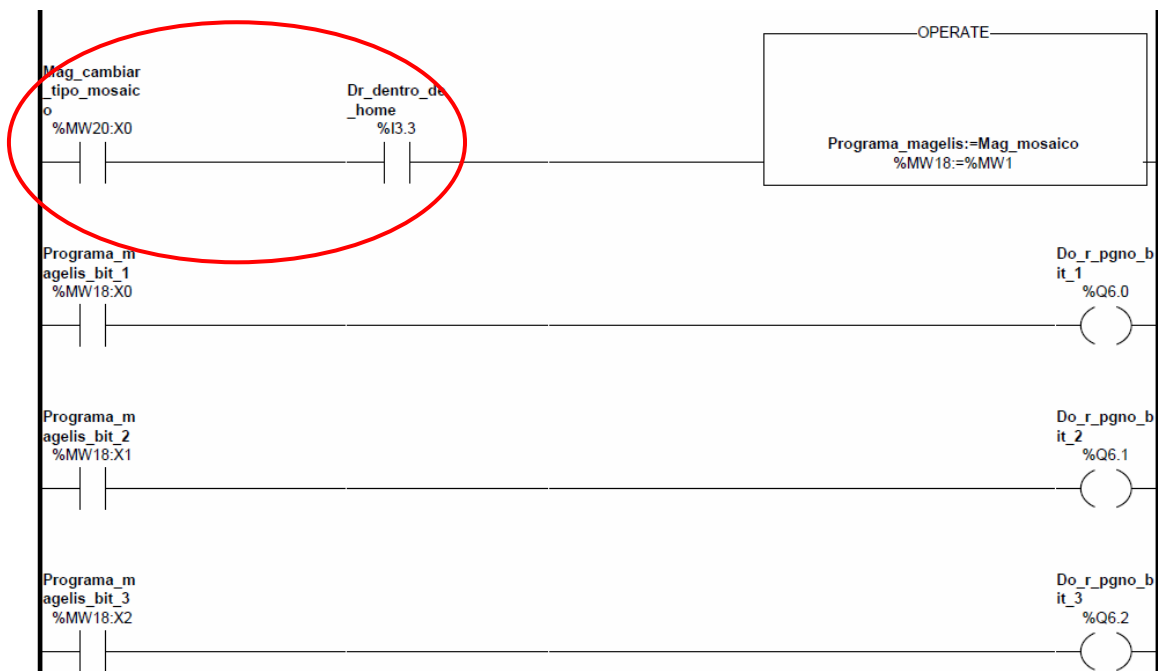
- **%I3.2-> Di\_paro\_saturacion.**

La señal para el paro de saturación, activará un relé con terminal libre de potencial, que desactivará la cinta del cliente en caso de saturación de sacos. Esta condición se producirá si hay presencia en los rodillos de recogida, y hay presencia en esta señal. De esta manera sin ser necesario evaluar el paro de la cinta de saturación, el saco se detendrá antes del vuelca sacos, no produciendo una nueva entrada a la cinta de saturación.

- **%I3.3-> Dr\_dentro\_de\_home-> Si estoy dentro del HOME se puede cambiar el mosaico**

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

El PLC es informado del estado dentro de HOME del robot, mediante **%I3.3**. Esta señal será usada por el PLC única y exclusivamente para enviar el valor de programa marcado desde el terminal XBT a través de los bits PGNO\_BIT para el cambio de mosaico. Será la condición necesaria para igualar las palabras de estado **%MW1** y **%MW18** cada vez que enviemos el dato de programa desde el XBT la cual activará la variable **%MW20:X0**.

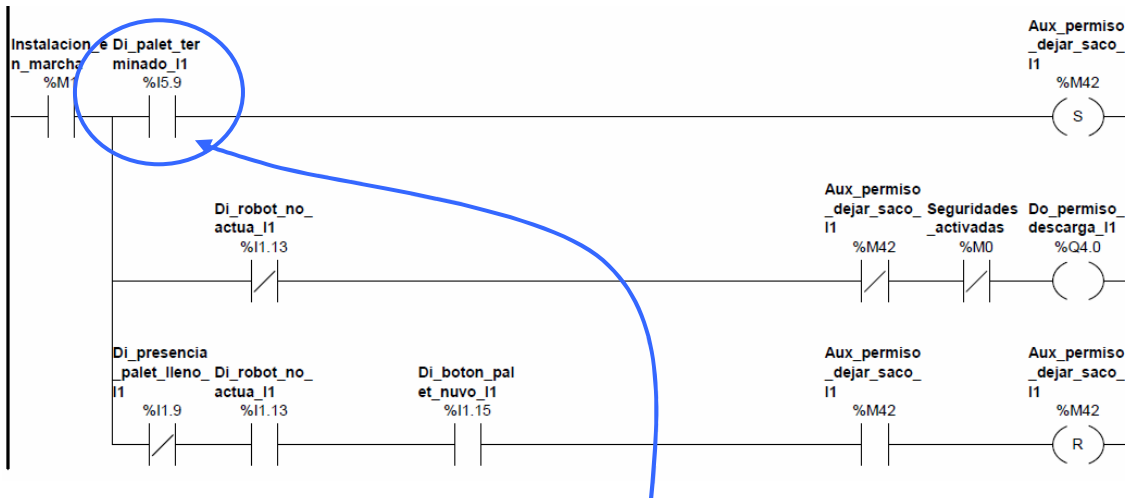


Una vez que el valor del programa es guardado en la variable **%MW118** se enviará a través de los tres primeros bits los datos al robot, mediante los bits PGNO\_BIT.

Dentro de los módulos de entrada 5 y salida 6 nos encontramos la Interface de comunicaciones con el robot, es por ello por lo que se generará un capítulo a parte para estas señales, en donde se detallará expresamente estos procesos, y analizaremos por separado el resto de señales de ambos módulos que no se encuentran integrados en la sección de programa del automático externo.

- **%I5.9-> Di\_palet\_terminado\_l1.**
- **%I5.10-> Di\_palet\_terminado\_l2.**

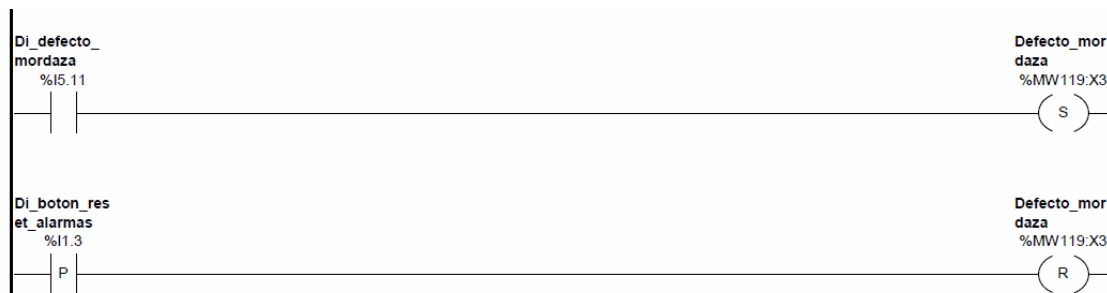
Las señales de palet terminado, se encuentran cableadas directamente desde las salidas beckhoff correspondientes. A través de (**%I5.9**) y (**%I5.10**) el robot indicará al PLC mediante pulsos mantenidos de dos segundos de esta señal, que el palet se ha finalizado en la correspondiente línea. Una vez recibida la señal, se activará en la sección de permisos una marca auxiliar (**%M42**) o (**%M43**) desactivándose así los permisos de descarga hasta que no se desarrolle el procedimiento de palet nuevo mencionado anteriormente.



Para el caso de producirse un **palet terminado** en la línea 1, se desactivará %Q4.0 de permiso de descarga.

- **%I5.11 -> Di\_defecto\_mordaza.**

Señal activada por el KRC, para indicar el defecto de reposo de la mordaza.



Dicho defecto sólo podrá desactivarse única y exclusivamente con el pulsador de reset del armario eléctrico, una vez que el operario haya visualizado el mensaje en el terminal XBT.

- **%Q6.10 -> Do\_r\_permiso\_carga.**

El permiso de descarga será también condición necesaria e indispensable para entrar en las zonas de paletizado Zona\_1() o Zona\_2(). De esta manera, estando la instalación en marcha, una vez que ha llegado el saco a la presencia de los rodillos de recogida, se activará el permiso de carga siempre y cuando haya algún permiso de descarga.

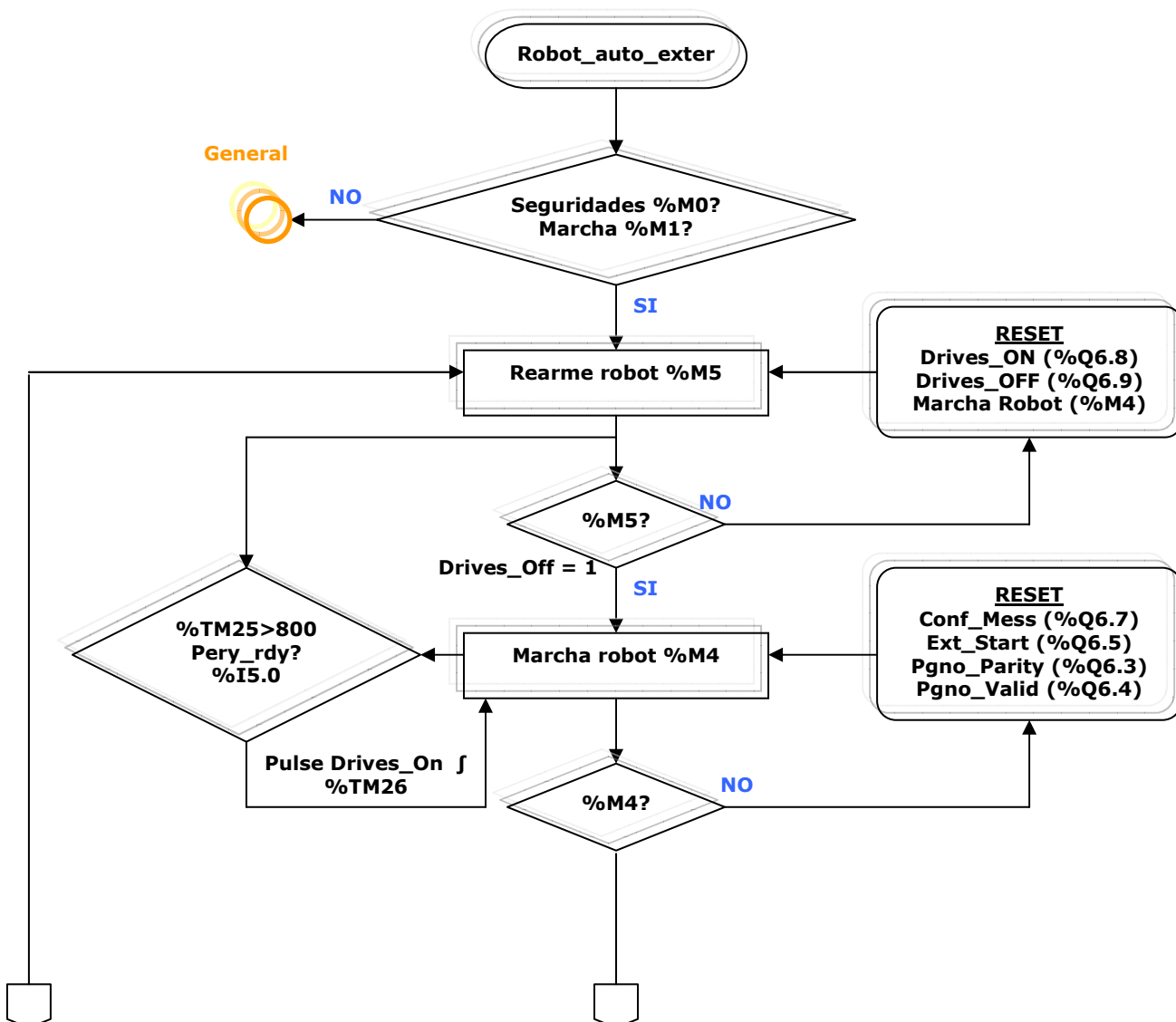


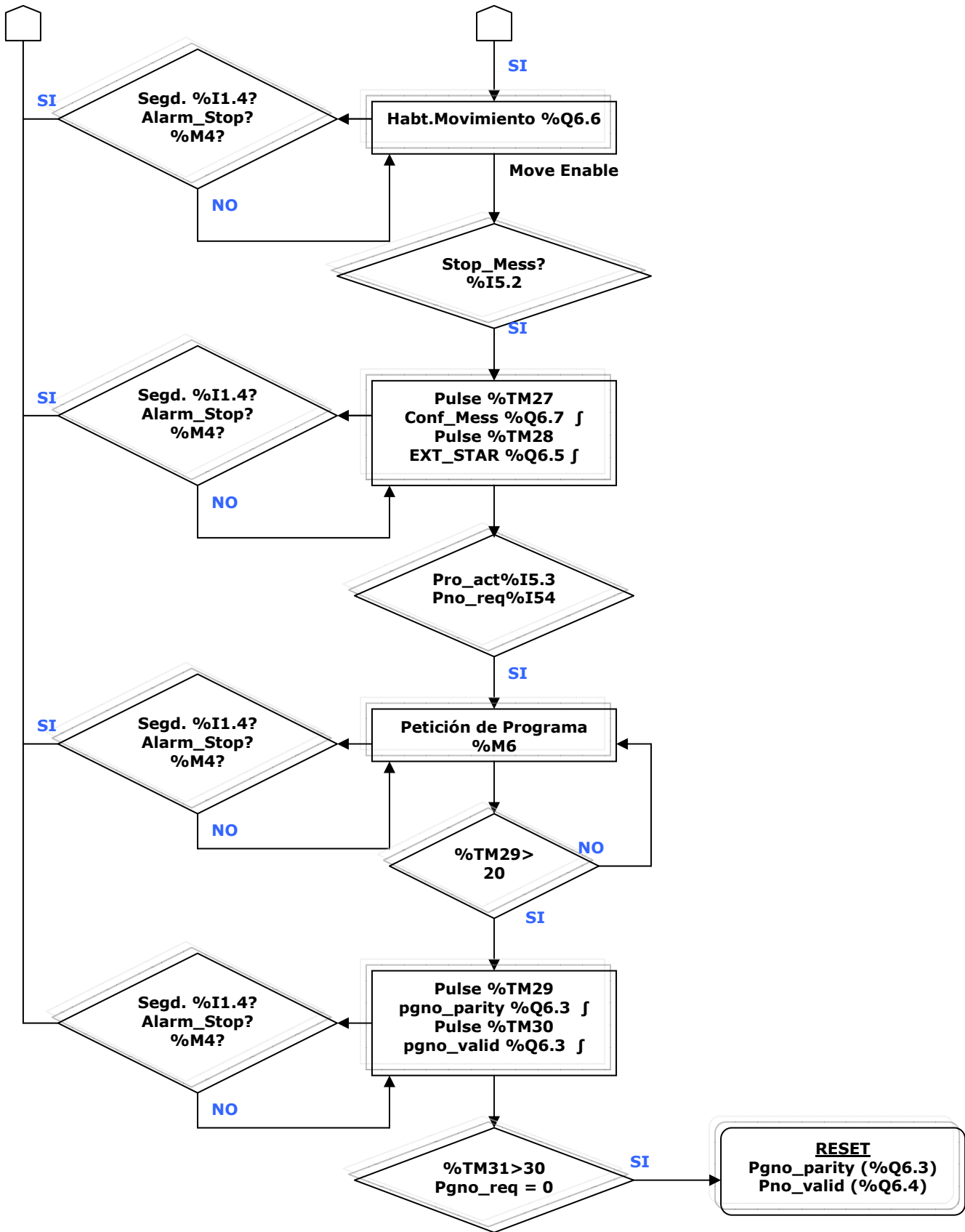
### 6.3.2 Comunicación con el robot.


Para la comunicación con el robot, se reserva el módulo de entradas número 5 y el módulo de salidas número 6, que componen las señales necesarias para la Interface con el robot mediante automático externo, ver para ello los capítulos correspondientes para esta Interface. La sección dedicada al automático externo se encuentra estructurada según el diagrama de señales adjunto en el capítulo relacionado con este apartado, empezando por la confirmación y activación de los accionamientos del robot, para pasar después a la comunicación y run del programa a ejecutar.

Se representará a continuación el diagrama de estados para esta sección del programa, detallando posteriormente cada uno de sus procesos de acuerdo con el diagrama de señales visto en la sección del automático externo.

#### 6.3.2.1 Diagrama de estados – Manejo del robot.





	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

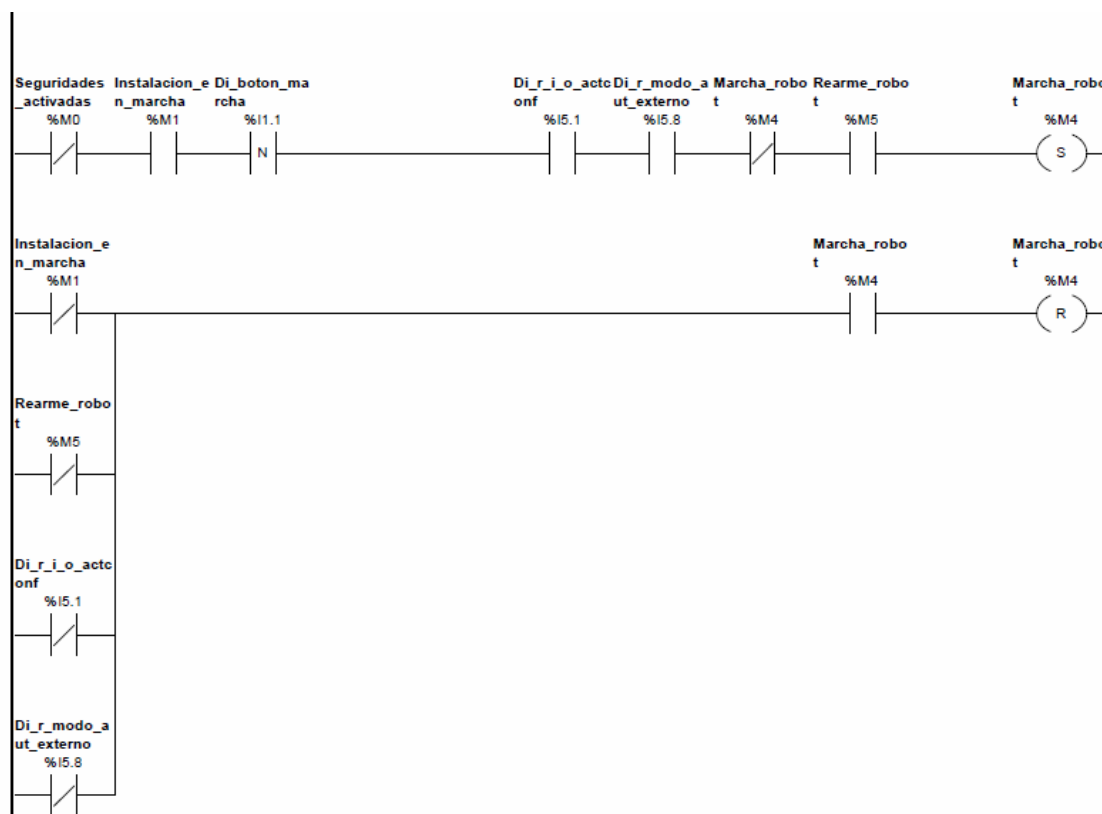
Lo primero que observamos en el diagrama, es la dependencia de las marcas de estado de marcha y seguridad general de la instalación, pues si este apartado no se encuentra OK, no podría funcionar nada.

En la primera parte del programa, se establecerá la marca de memoria que representan el rearme del robot **%M5** será necesario por lo tanto que no exista ninguna alarma en el brazo robot **alarm\_stop-> (%I5.6)** para rearmar y poder hacer a continuación la marcha general del robot, **%M4** tendrá que estar activada para que el resto de señales se procesen acuerdo al funcionamiento establecido.

Para la activación de la marcha del robot (**%M4**) se tendrá en cuenta:



1. **%M0**-> Las seguridades de la instalación tendrán que estar OK.
2. **%M1**-> La marcha general tendrá que estar activada.
3. **%I5.8**->**modo\_aut\_externo**-> El modo de funcionamiento del robot tendrá que estar establecido en automático externo.
4. **%M5**-> Las seguridades del robot tendrán que estar activadas.

Cumplidas estas condiciones, mediante el flanco de bajada del pulsador de marcha general **%I1.1**, activaremos la marcha del robot (**%M4**).



Para la activación y rearme del robot (**%M5**) tendrán que cumplirse las siguientes condiciones:

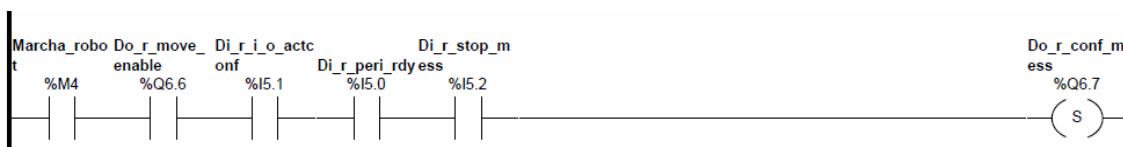
1. **%I5.6**->**alarm\_stop**-> No deben existir eventos de alarmas ni mensajes por confirmar en el KRC.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

2. **%I1.4->Modulo de seguridad->** El módulo de seguridad del armario de control debe encontrarse activado.

Si **%M5** no se encuentra activado, no se podrán, rearmar los accionamientos del robot (**%Q6.9 Drives\_off** y **%Q6.8 Drives\_On**), y en caso de no encontrarse activada la marcha del robot **%M4** la Interface de comunicación y transmisión de datos se encontrará desactivada, así como la habilitación del movimiento del robot (**%Q6.6->Move\_enable**) que sólo podrá producirse siempre y cuando esta marca se encuentre activada y seleccionado el modo automático externo del selector.

Una vez que los accionamientos del robot están reestablecidos y el movimiento habilitado, será preciso la confirmación de los mensajes de advertencia que el KRC transmite a través de la señal **Stop\_Mess**, para lo que se emplea la señal de Interface (**%Q6.7->Conf\_Mess**). Si hay mensajes por confirmar (**%I5.2**) estos serán cancelados en la KCP, siempre y cuando la marcha y accionamientos del robot se encuentren activos, para lo que no deberá de haber ninguna seguridad activada.

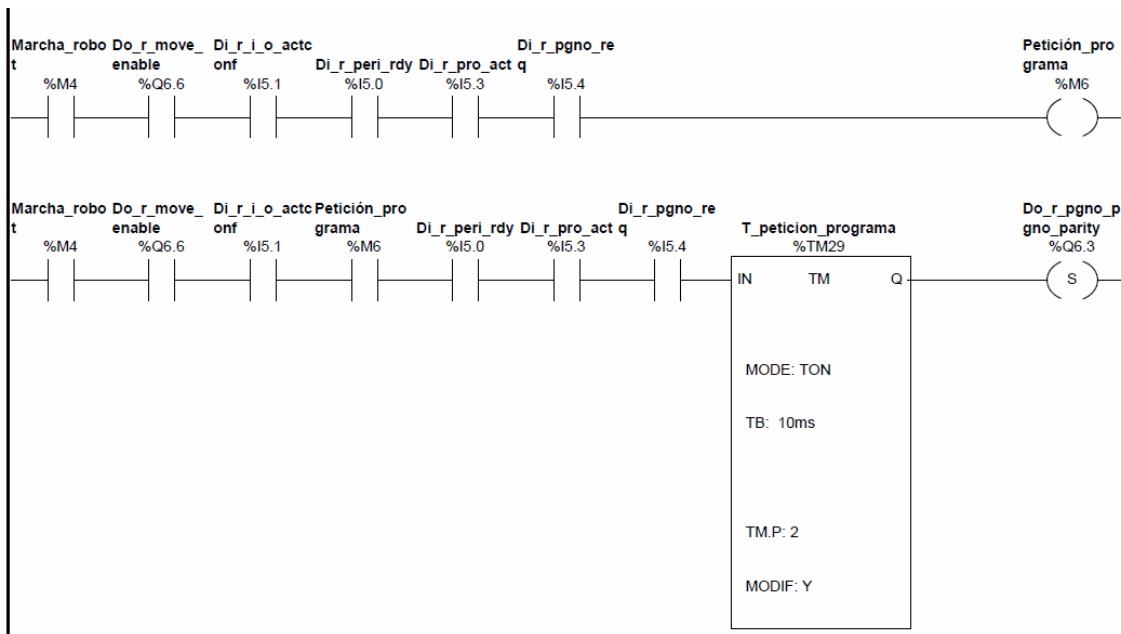


Una vez el KRC indique la no existencia de ningún mensaje a confirmar, a través del temporizador de 20ms (**%TM27**) de reloj resetearemos la señal (**Q6.7-> Conf\_Mess**) y arrancaremos el proceso activo **Cell.src** mediante la activación de la señal **EXT\_START** (**%Q6.5**) cuyo pulso de duración dependerá del timer **%TM28** y después de la confirmación del KRC mediante la señal **pro\_act->** (**%I5.3**) indicando que el proceso se encuentra activo.

La Interface de señales que da comienzo después de **EXT\_START -> Pgno\_parity->pgno\_valid** se activará después de cada paro y marcha para iniciar la ejecución de un nuevo programa según queda representado en el diagrama de señales en el caso de que nos encontráramos dentro de **cell.src**.

En caso de no existir ningún programa de paletizado en ejecución la KRC activará la señal **PGNO\_REQ->** (**%I5.4**) de manera que se procesará la señal **PGNO\_PARITY->** (**%Q6.3**) al cabo de un tiempo **%TM29** y siempre y cuando se cumplan todas las condiciones de puesta en marcha.





PGNO\_PARITY se encuentra temporizada de manera que al cabo de un tiempo de seguridad, y tras la confirmación del KRC mediante la señal **PGNO\_REQ (%I5.4)** se activará el consentimiento de lectura del PLC al KRC a través de la señal **PGNO\_VALID (%Q6.4)**. Acto seguido se resetearán ambas señales transcurrido un tiempo de seguridad (**%TM31**) para la lectura del PLC.

Cada vez que se ejecuta un paro desactivándose la marca de memoria (**%M4**) se resetean las señales **pgno\_parity-> (%Q6.3)**, **pgno\_valid-> (%Q6.4)**, **ext\_Start-> (%Q6.5)** y **Conf\_mess->(%Q6.7)**, de manera que una vez se vuelva a iniciar la marcha, se confirmarán automáticamente los mensajes de parada anteriores, se reestablecerá nuevamente **EXT\_START** y si hubiera petición de programa dentro de **cell.src** se volverían a activar las señales PGNO\_PARITY y PGNO\_VALID, para el nuevo consentimiento de lectura.

### 6.3.3 Salidas digitales.

En el autómata distinguimos tres módulos de salidas de 12 bit's conectados por medio de los módulos **10BO** integrados en el armario eléctrico.

- **%Q2.0-> Do\_cinta\_entrada\_saco\_l1-> cinta de entrada saco de la línea 1.**
- **%Q2.1-> Do\_camino\_recogida\_saco\_l1->camino de recogida saco línea 1.**

Representan las señales a 24V para activar los contactores de potencia que alimentarán los motores de cinta de saturación y rodillos de alimentación correspondientemente.

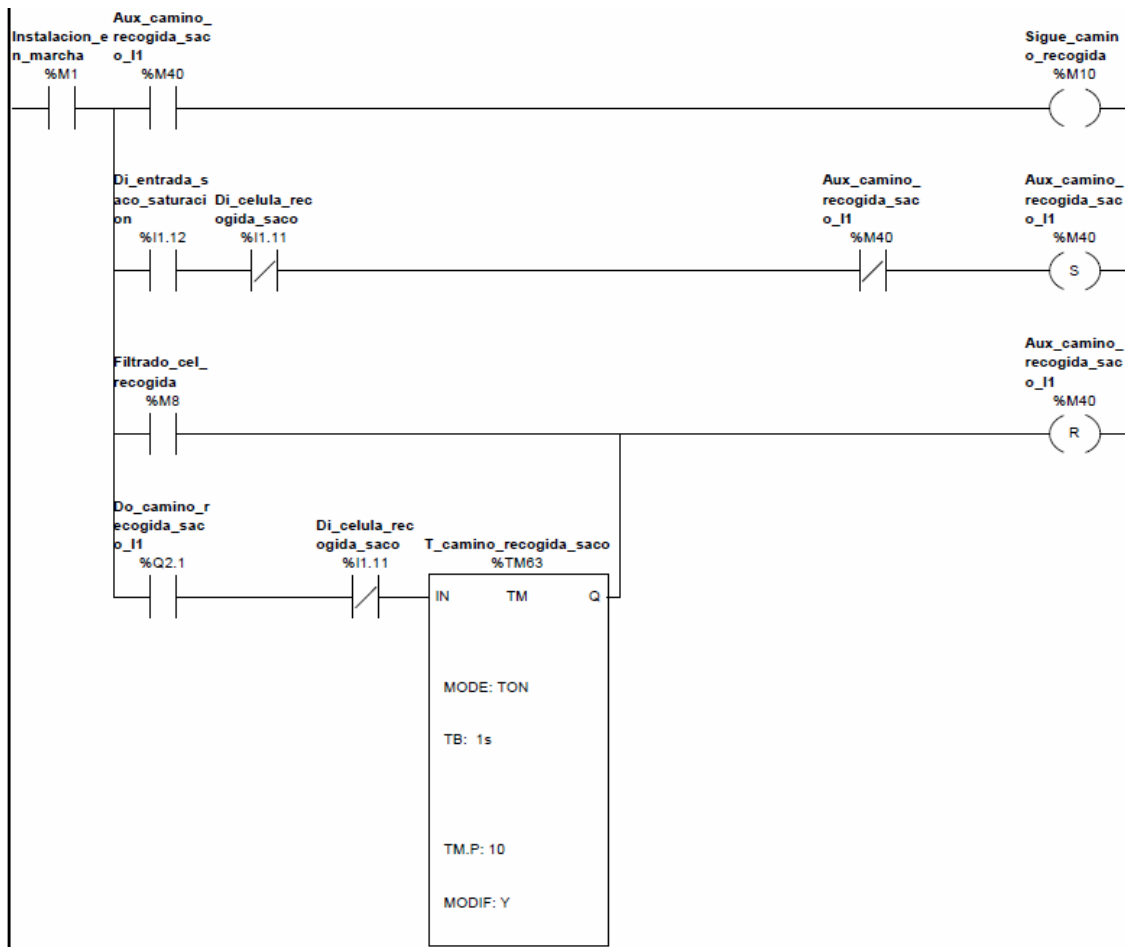


La cinta de saturación estará activa siempre que la instalación se encuentre en marcha, y haya una de las dos fotocélulas de entrada libre. También se activará cuando este seleccionado el movimiento manual desde el XBT, en donde la instalación no deberá encontrarse en marcha.

Los rodillos de alimentación (**%Q2.1**) se activarán a través de la marca auxiliar (**%M10**) que será activada bajo las siguientes condiciones:

1. La instalación tendrá que encontrarse en marcha (**%M1**).
2. Presencia previa en la cinta de saturación (**%I1.12**).
3. No haya presencia en los rodillos de alimentación.

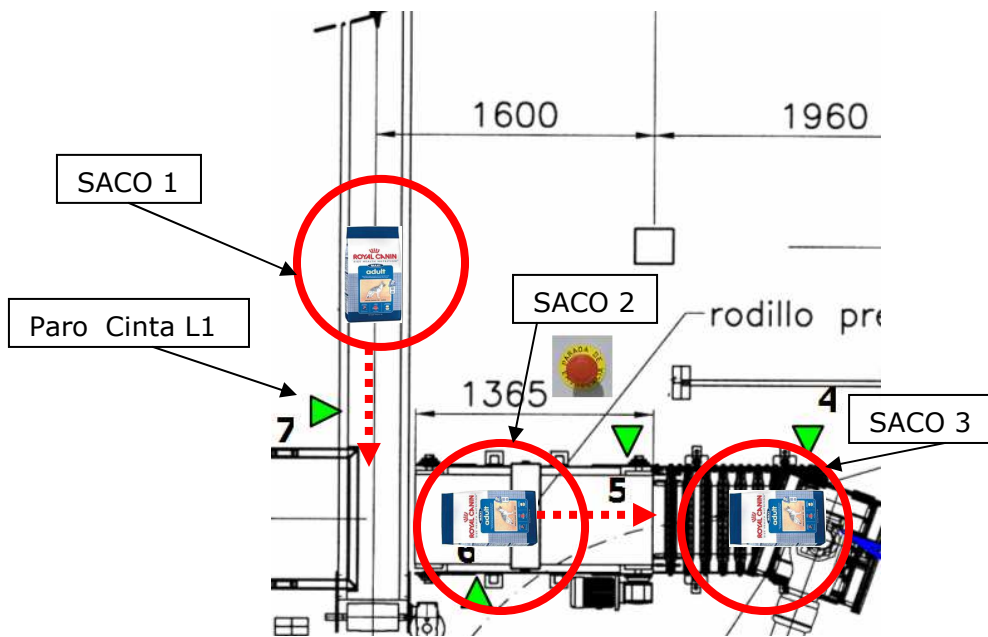
A parte de esto se contempla la posibilidad de una falsa detección en la presencia de la cinta de saturación; es decir, si al cabo de un tiempo (**%TM63**) de 10 segundos de producirse la marcha de los rodillos, no se detecta presencia de saco en la fotocélula de recogida, se desactivará la señal.



- **%Q2.2-> Do\_paro\_I1-> Paro línea 1.**

Existe una seguridad de saturación que para la línea de entrada del cliente antes del vuelca sacos, para evitar el atascamiento del mismo. La situación se produce cuando habiendo un saco en los rodillos de alimentación se produce una presencia de saco antes del vuelca sacos, contando con que puede existir un saco en la cinta de saturación, no se debería de accionar el vuelca sacos ya que la cinta de saturación estaría parada y el saco se podría quedar atascado. De esta manera a través de esta señal se activará un relé con contacto libre de potencial por el cual se lleva la alimentación al contactor de la cinta de entrada, provocando así su paro hasta que el robot recoja el saco correspondiente dejando libre la presencia.

En la imagen adjunta se observa que si detectamos presencia con el **saco 1** habiendo un **saco 3** en los rodillos de alimentación, la cinta de entrada tendría que pararse puesto que el **saco 2** hará detenerse la cinta de saturación y el **saco 1** podría quedarse atascado si actuase el vuelca sacos.

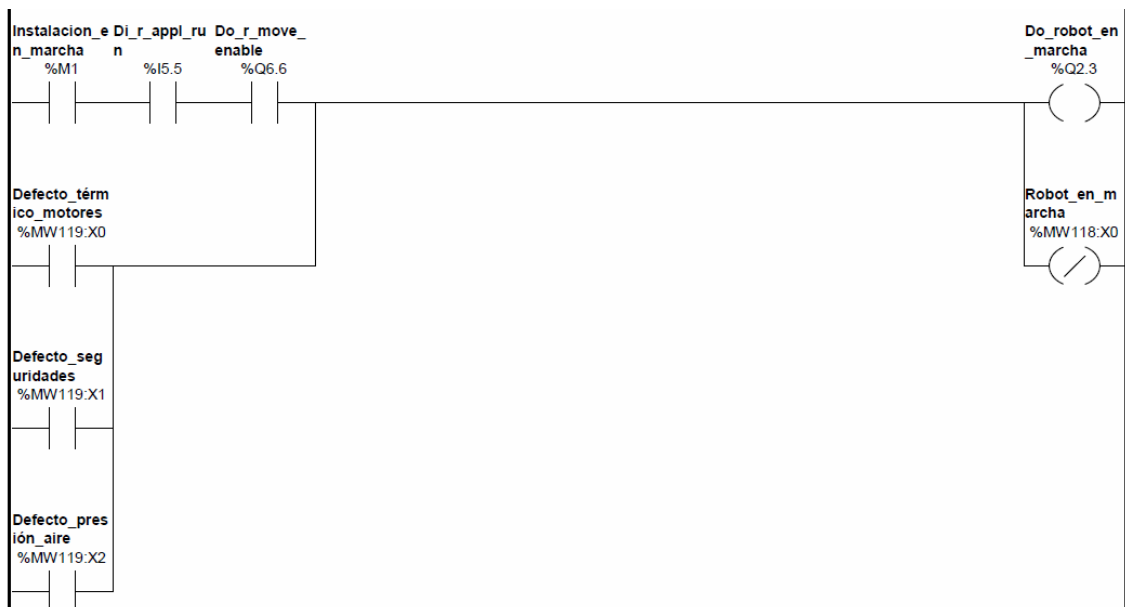




- **%Q2.3-> Do\_robot\_en\_marcha-> Robot en marcha.**

La salida de marcha robot, se emplea para su visualización en el terminal XBT. Así pues dentro de la sección de alarmas programaremos las condiciones que son necesarias para señalar la marcha del robot, que son las siguientes:

1. Marcha general **%M1**.
2. Existencia de programa de paletizado en ejecución **appl\_run-> (%I5.5)**.
3. Movimiento habilitado **%Q6.6**

Dicho mensaje se visualizara siempre y cuando no se produzca ningún defecto de seguridad con mayor prioridad.



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

- **%Q2.4-> Do\_bajar\_rodillo-> Rodillo para prensar el saco.**

Los rodillos actuarán siempre y cuando la instalación se encuentre en marcha y mientras se produzca presencia en **%I3.1->Presencia de rodillo**. La salida **%Q2.4->Bajar\_rodillo** representa la activación de la bobina electromagnética del cilindro prensa.

- **%Q4.0-> Do\_permiso\_descarga\_I1.**
- **%Q4.1-> Do\_permiso\_descarga\_I2.**

Los permisos de descarga de la línea de paletizado, son condiciones directas del robot para entrar en las zonas de paletizado. Así mismo si alguna de estas señales no se encuentra activa, no se enviará al KRC el permiso de descarga comentado anteriormente. Las condiciones necesarias para que el PLC active dicha señal son las siguientes:

1. Marcha y seguridades generales activas. (**%M0**) y (**%M1**).
2. Confirmación por parte del KRC de la finalización del palet (**%I5.9/10**).
3. Selector de quitar permiso de paletizado desactivado (**%I.13/14**).
4. Desactivación de la marca auxiliar de permisos **%M42**.

### 6.3.4 Comunicación con terminal XBT.

Las siguientes señales se encuentran en la sección de comunicaciones, donde se establece el traspaso de datos desde el terminal XBT hacia el robot. Ya se comentó anteriormente cuales eran el significado de estas señales de entrada directa al robot.

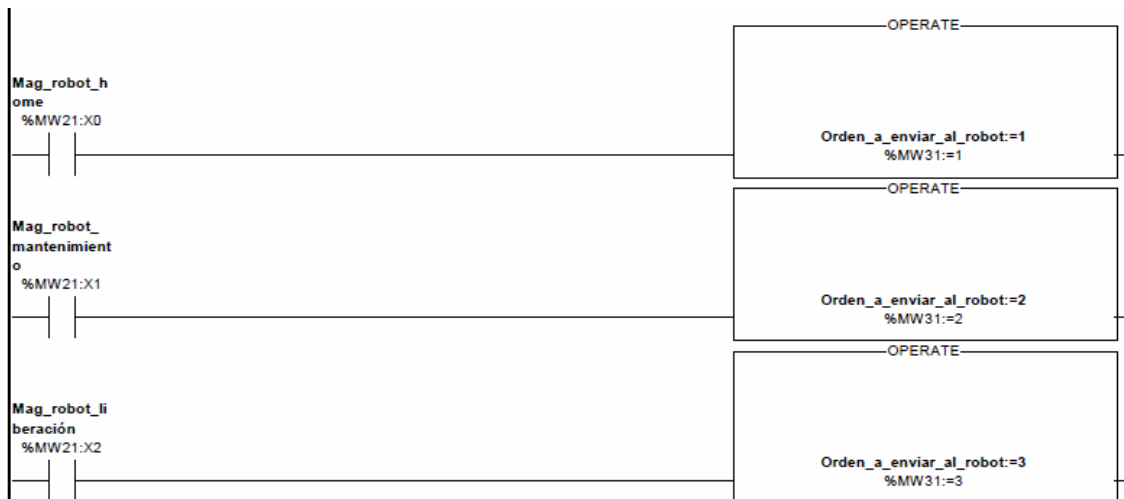
- **%Q4.2-> Do\_robot\_carga\_datos-> ROBOT:Carga datos línea 1.**
- **%Q4.4-> Do\_robot\_carga\_datos\_mosaico-> ROBOT: Carga datos de mosaico**
- **%Q4.5-> Do\_robot\_carga\_datos\_capas-> ROBOT: Carga datos de numero de capas**
- **%Q4.6-> Do\_robot\_carga\_datos\_ordenes-> ROBOT: Carga datos de ordenes**
- **%Q4.7-> Do\_robot\_carga\_datos\_incremento->ROBOT: Carga datos de incremento de alturas.**

**%Q4.2->** será enviada por el PLC hacia el robot para indicar que se quiere enviar algún dato procesado desde el terminal XBT nº de programa, alturas, etc.

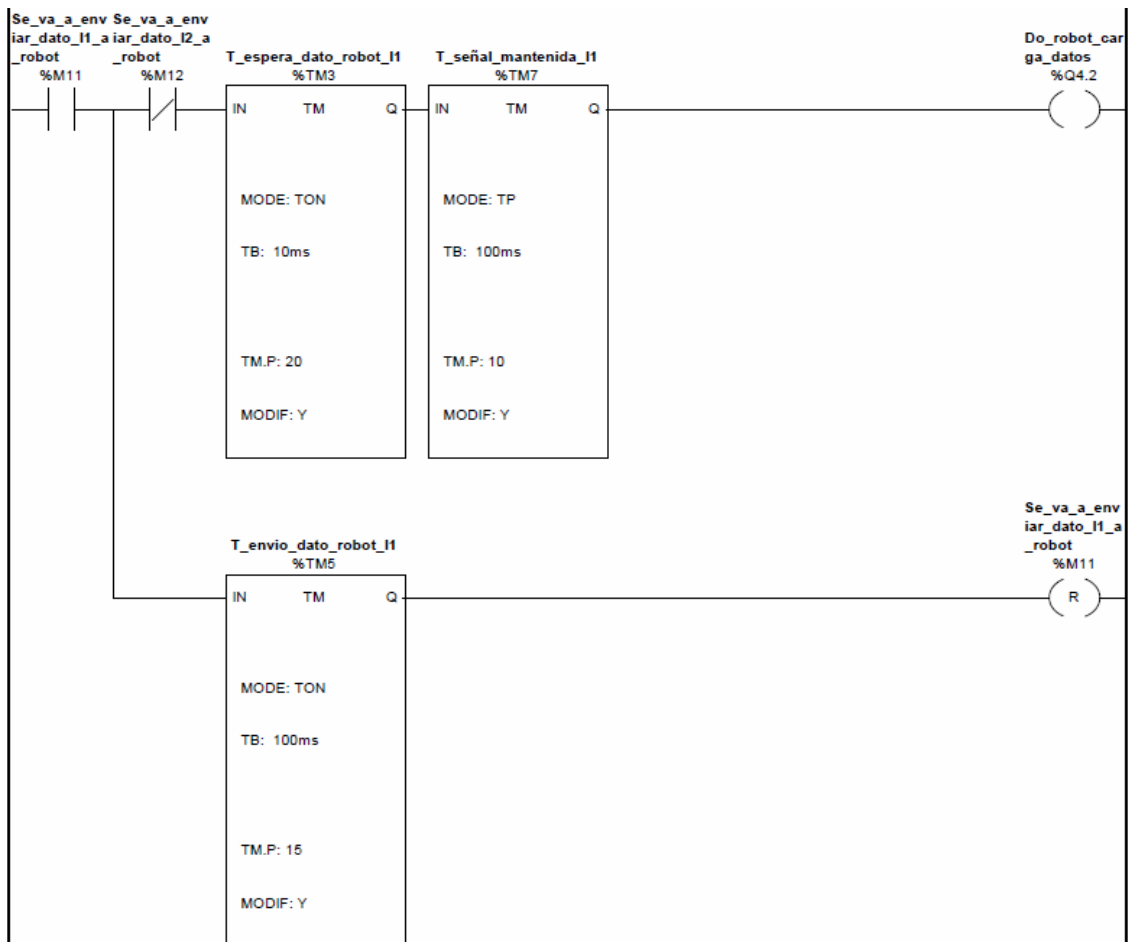


Cada vez que se procesa una orden desde el Terminal XBT se activa la marca de memoria (**%M11**), esto será si ejecutamos algún movimiento manual a través de los bit's de la palabra **%MW20** o si la variable **%MW31** que captura el valor de la orden a través de los bit's de la palabra **%MW21** es distinto de cero. En nuestro caso las únicas órdenes que procesamos desde el XBT son las siguientes:

- **%MW20:X0 -> Confirmación cambio de Mosaico (NO USADO).**
  - **%MW20:X1 -> Confirmación para cambiar Capas.**
  - **%MW20:X2 -> Confirmación para cambiar incrementos.**
  - **%MW20:X6 -> Reset de contadores.**
- 
- **%MW21:X0-> Orden de ir a HOME.**
  - **%MW21:X1-> Orden de ir a Mantenimiento.**
  - **%MW21:X2-> Orden de liberación.**
- } **%MW31**



Una vez que la palabra %MW31 es cargado con algún valor, y a la vez que se activa %M11, se activa mediante un pulso de señal la salida %Q4.2, este pulso será de una duración de 1 segundo a través de %TM7 suficiente para que el KRC realice la lectura. Mientras se envía el dato %M11 volverá a desactivarse mientras que %M12 permanecerá desactivada puesto que en esta instalación las órdenes procesadas desde el XBT en la línea 1 serán las mismas que en la línea2.




En el momento que %M11 es resetada a través de %TM5 en un tiempo de 1,5 segundos %Q4.2 será desactivada. De esta manera se le indica al robot que se van a cargar los números de alturas o los números de capas, por lo tanto tendremos lo siguiente:

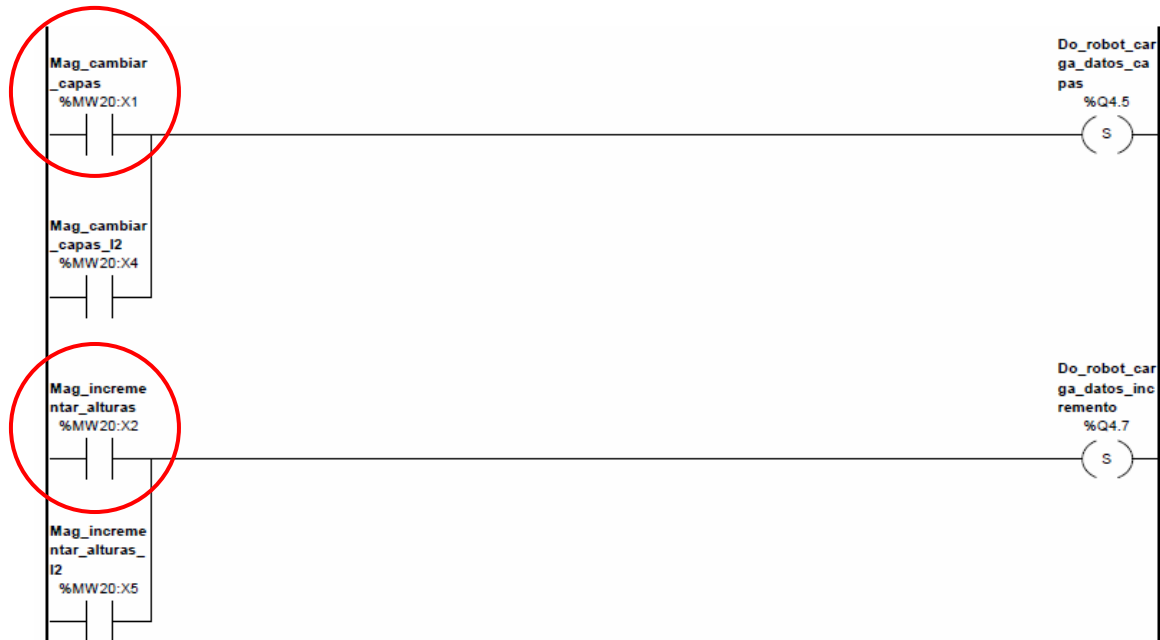
- **%Q4.2->salida PLC->Entrada Directa al robot \$IN[15].**

Se pondrá a nivel '1' o 'true' desde el terminal XBT, para indicarle al robot que se quiere cargar el número de alturas o el número de incrementos. Recordemos que estos valores son recogidos por el programa del robot desde el programa multitarea submit.

Seguidamente será necesario especificar si los datos que se quieren cargar son el número de alturas del palet o el valor del incremento de alturas, puesto que

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

el programa a ejecutar se transmite a través de los bits PGNO\_BIT. Para ello si desde el terminal se activa la variable **%MW20:X1->Los datos a cargar serán las capas ->%Q4.5**, y con **%MW20:X2-> Los datos a cargar serán los incrementos-> %Q4.7**.



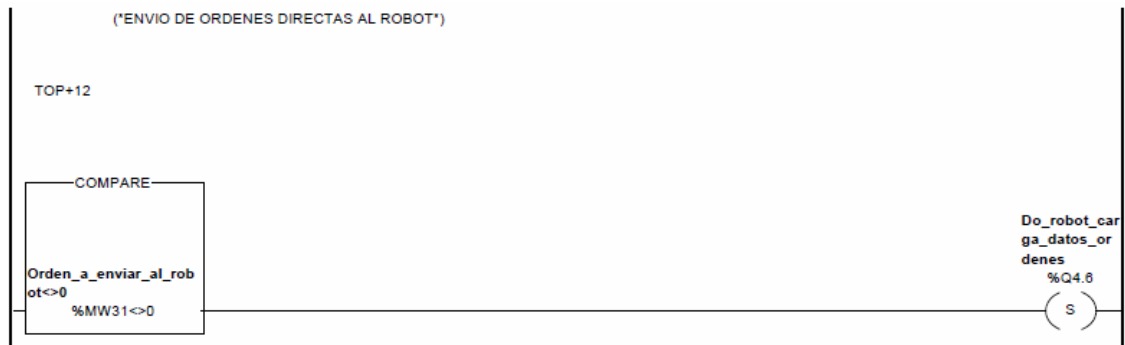
Las variables **%MW20:X4** y **%MW20:X5** no se usan en esta instalación siendo bit's reservados para cuando haya dos líneas a los que se les puede pasar datos diferentes.

- **%Q4.5->salida PLC->Entrada Directa al robot \$IN[18].**
- **%Q4.7->salida PLC->Entrada Directa al robot \$IN[20].**

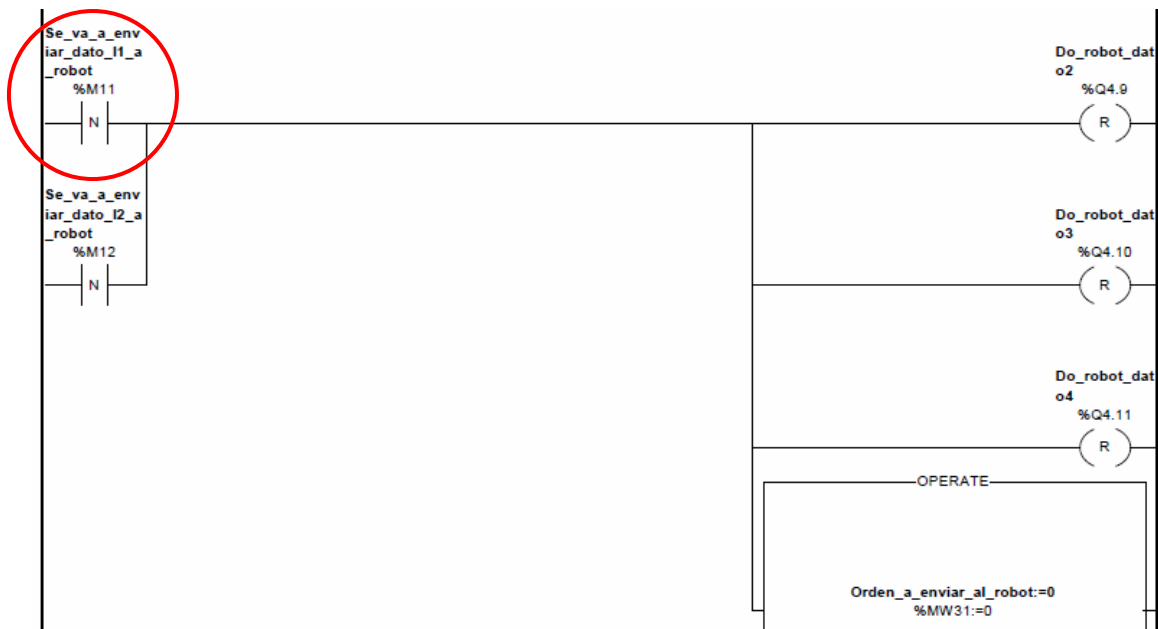
Para el caso en el que enviemos el número de mosaico, la salida **%Q4.4-> Salida PLC->No será la entrada Directa al robot \$IN[17]** por lo tanto no será usada aunque su valor cambie de estado desde el terminal XBT, lo que realmente será usado es la variable **%MW1** del terminal que guarda el valor del programa seleccionado. Y que será transmitido una vez el KRC confirme que se encuentra en estado HOME.

Seguidamente será necesario indicar si lo que se quiere hacer es cargar una orden y que tipo de orden, así por ejemplo en el momento que la variable **%MW31** es distinto de cero, también se activará la señal **%Q4.6-> Carga\_Datos\_de\_Ordenes-> Entrada Directa al robot \$IN[19]**. Indicando así al KRC que se va a transmitir una orden mediante la variable **%MW21**.





Por último, las salidas **%Q4.2**, **%Q4.3**, **%Q4.4**, **%Q4.5**, **%Q4.6**, **%Q4.7** y **%Q4.8 (DATO\_1)**, **%Q4.9 (DATO\_1)**, **%Q4.10 (DATO\_1)**, **%Q4.11 (DATO\_1)**, y **%MW31 = 0** serán reseteadas en el momento que **%M11** caiga a nivel bajo nuevamente.

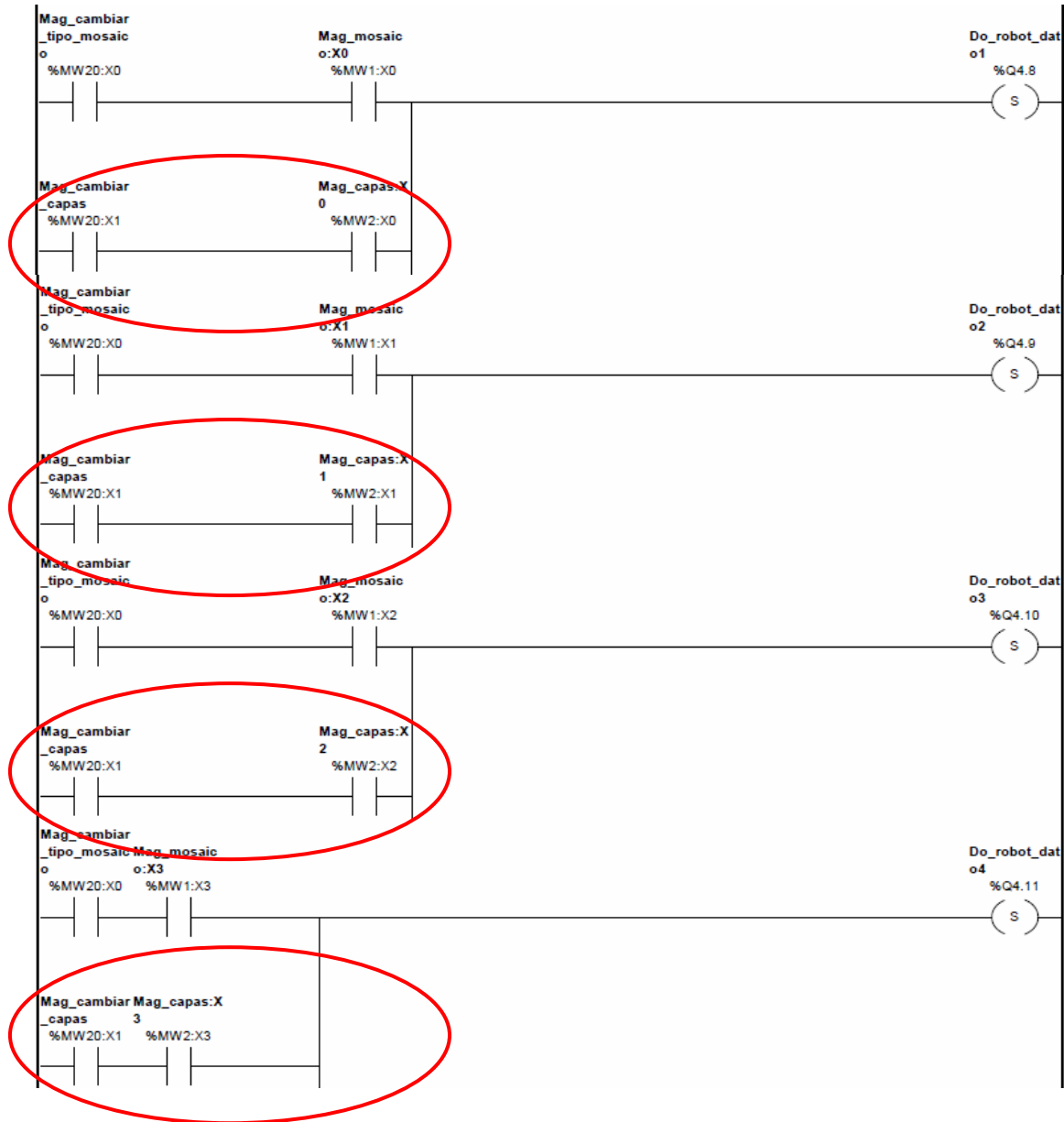


La transmisión de datos se efectúa a través de las siguientes salidas:

- **%Q4.8-> Do\_robot\_dato1-> ROBOT: Dato 1-> \$IN[21]**
- **%Q4.9-> Do\_robot\_dato2-> ROBOT: Dato 2 -> \$IN[22]**
- **%Q4.10-> Do\_robot\_dato3-> ROBOT: Dato 3 -> \$IN[23]**
- **%Q4.11-> Do\_robot\_dato4-> ROBOT: Dato 4 ->\$IN[24]**

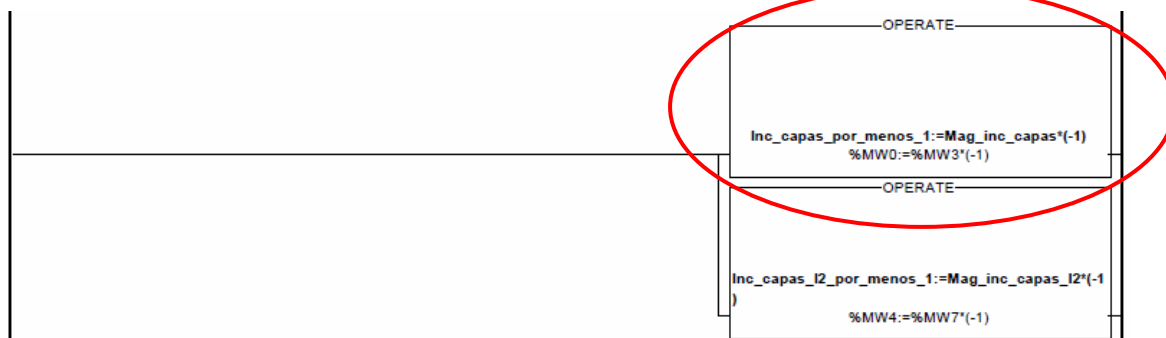
Estos bit's que en el programa del robot forman una señal a través de variables declaradas en el archivo \$Config.dat, se pondrán en estado '1' o '0' dependiendo de las variables anteriores; es decir, si se van a enviar datos se enviará el valor en binario del número de capas o el incremento de alturas, y si se van a ejecutar ordenes se enviarán en binario los valores de la variable **registro\_1** declarada en el robot y la cual usamos para entrar en **home()**, **Mantenimiento ()** o provocar la **liberación**.

En el caso que queramos enviar el número de capas o de incrementos, se emplearán como ya se vio en apartado del XBT las variable **%MW2** y **%MW3** respectivamente que guardarán dichos valores, mientras que para activar el envío se emplean los bit's **%MW20:X1** y **%MW20:X2**. que serán las que activen la marca **%M11** en este caso.

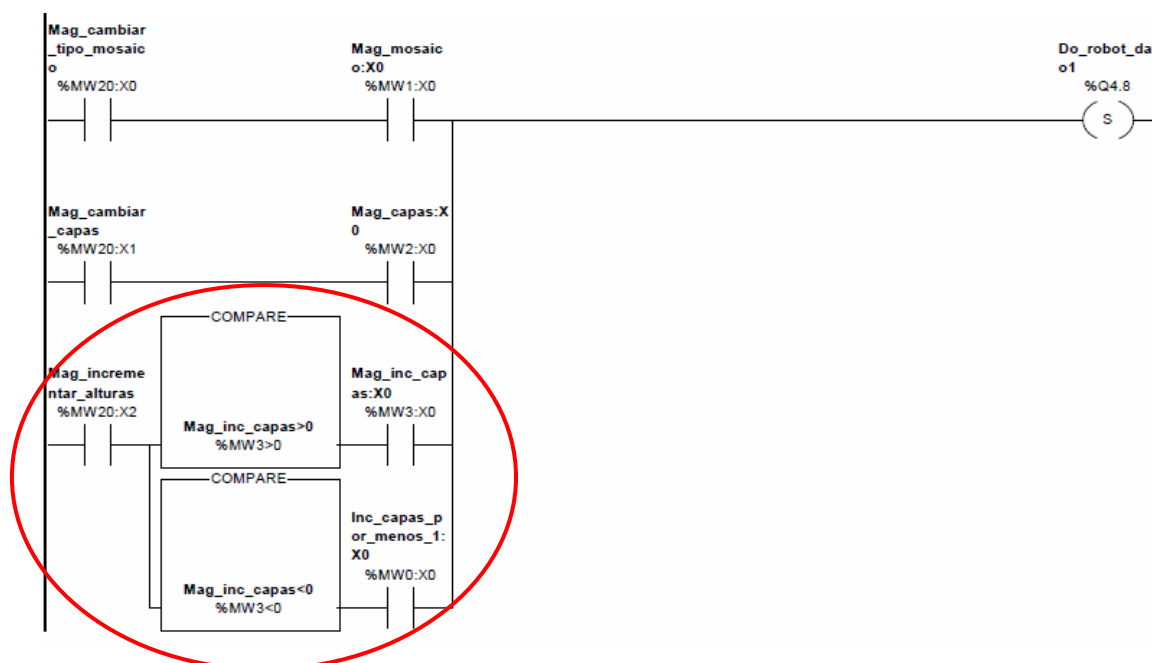


Los bit's de datos transmitirán el dato de capas, si activamos desde el XBT la variable **%MW20:X1** y los bit's de la variable **%MW2** que contienen el valor en binario que seleccionamos desde el Terminal.

Para el caso de los incrementos de alturas, en el caso de que la variable **%MW3** tenga un valor negativo se activará con el mismo valor y signo contrario la variable de signo **%MW0**.



De esta manera **%Q4.8**, **%Q4.9** y **%Q4.10** enviarán el dato a transmitir mientras que **%Q4.11** se activará si el valor a transmitir es negativo.



Si activamos el envío de los incrementos mediante **%MW20:X2** tenemos el caso en el que el valor sea positivo o negativo.

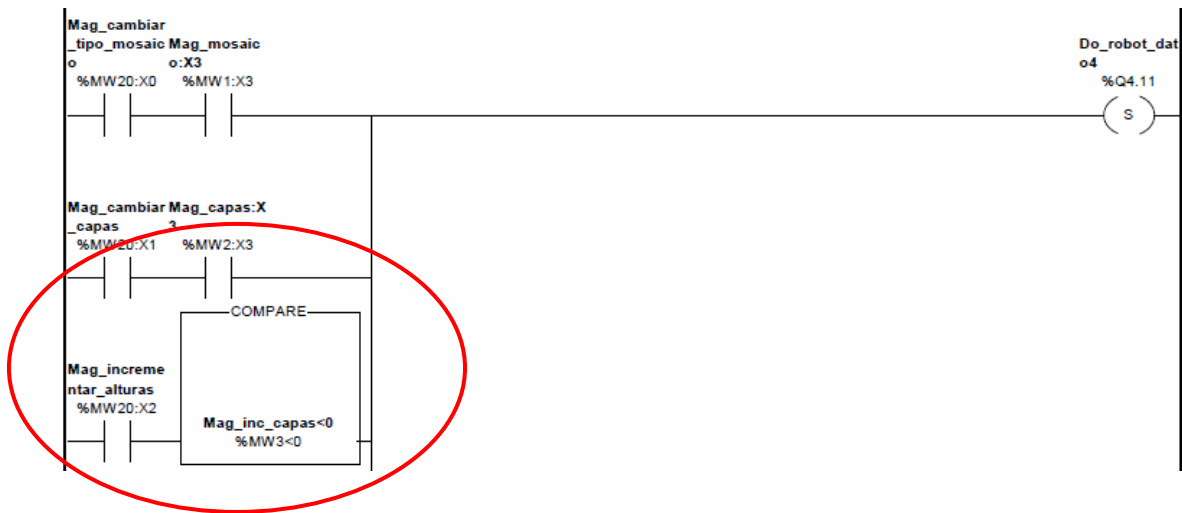
- **%MW3 > 0.**

{	<b>%Q4.8 = %MW3:X0</b> <b>%Q4.9 = %MW3:X1</b> <b>%Q4.10 = %MW3:X3</b>
---	---
  
- **%MW3 < 0.**

{	<b>%Q4.8 = %MW0:X0</b> <b>%Q4.9 = %MW0:X1</b> <b>%Q4.10 = %MW0:X3</b>
---	---

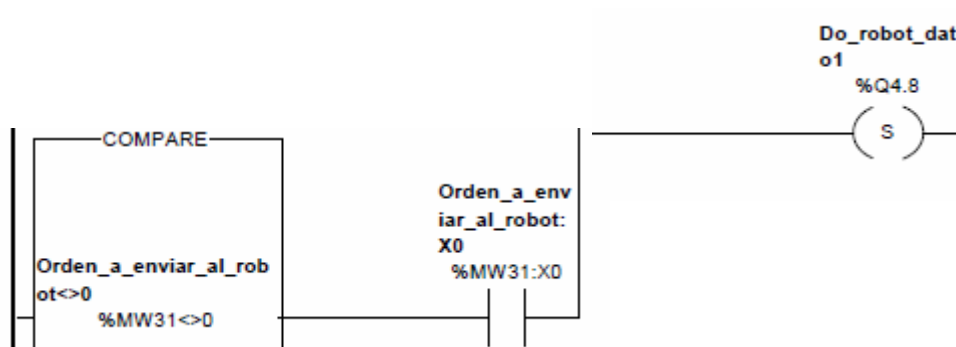
De esta manera siempre se transmitirá el bit positivo tanto si el valor del XBT es positivo como si es negativo.

El bit de signo **%Q4.11** se transmitirá si **%MW3 < 0** lógicamente.



Como último caso nos queda el cambio de la variable **registro\_1** del robot cuando lo que transmitimos es un tipo de orden. Las señales de salidas adquirirán los siguientes valores:



- **%MW21:X0 -> Ir a Home ->                    %MW31 = 1**
- **%MW21:X1 -> Ir a Mantenimiento -> %MW31 = 2**
- **%MW21:X2 -> Liberación ->                %MW31 = 3**



Si la orden es distinta de cero, se transmitirá el bit de la orden correspondiente.

	%Q4.11->DATO_4	%Q4.10->DATO_3	%Q4.9->DATO_2	%Q4.8->DATO_1
<b>Registro_1 = 1 HOME</b>	false	false	false	true
<b>Registro_1 = 2 MANTENIMIENTO</b>	false	false	true	false
<b>Registro_1 = 3 LIBERACION</b>	false	false	true	true

**Tabla: ORDENES del Robot.**

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

## 7 CONCLUSIONES.

### **"OPTIMIZACION DE UNA INDUSTRIA ALIMENTARIA"**

La paletización robotizada como herramienta, ha sustituido la mano de obra aumentando la productividad y la eficiencia. Disminuyó los procedimientos administrativos de recursos humanos y legales que conllevan la rotación de personal; muy necesario siempre para la realización de estos tipos de trabajo.

El principal problema al que se enfrentaba "**Piensos O´Couto**" radicaba en la necesidad de generar un mayor volumen de suministro para cumplir con sus compromisos de abastecimiento. Esto sin el personal suficiente y las contingencias que se ocasionaban al tratarse de trabajos forzados, limitaba las exigencias productivas de la compañía.

La integración de la instalación de robot paletizado con control automático, garantizó, una mayor producción; sin ser por ello, una herramienta de sustitución del personal que pasarían a realizar trabajos de retirada de palet´s y suministro de material para la realización del ensacado del producto y su transporte hacia la entrada del robot. Todo esto teniendo en cuenta, que el espacio ocupado resultó mínimo, ya que el brazo robotizado se adaptó perfectamente a las limitaciones del recinto, siendo así el modo más adecuado para automatizar la producción del cliente.

Como resultado final, se obtiene un palet totalmente geométrico en las esquinas del palet, aunque esto depende de las características del producto y la homogeneidad con la que el saco llega a la línea de entrada del robot

Se creó en cierto modo, una mayor actividad y más dinamismo a la hora de ejecutar las diferentes operaciones implicadas en los objetivos de la empresa. Pues esto significaba, que disponiendo de los mismos recursos humanos, se llevaban a cabo trabajos de paletizado (automáticos para el robot), a la vez que se transportaba en camiones los palet´s que iban siendo retirados.



Mediante este sistema, se consiguió en caso de necesidad, una disponibilidad del 100%; esto es, 24horas de trabajo al día durante siete días a la semana con un personal mínimo, siendo capaz de paletizar más 16 toneladas de producto terminado a la hora, lo que hace un total de unas 400 o 500 toneladas al día.

### **"INSTALACIÓN"**

Este tipo de instalaciones, resultan muy versátiles en su ejecución y puesta en marcha. Se requiere poner especial cuidado en la instalación del robot, pues deben medirse correctamente la situación de los elementos y el vallado necesario antes de anclar la base sobre el que se apoyará. Siendo el robot el elemento sobre el que más hincapié hay que hacer en su instalación.

Una vez ubicados todos los elementos, el cableado y las conexiones son sencillas, mediante borneros y conectores.

El cliente ha de proveer de las tomas de alimentación correspondientes para la llegada de potencia a los armarios eléctricos.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

En cuanto a inconvenientes, nos encontramos el que una vez situado el robot y ancladas las escuadras de sujeción de los palet's, estos no eran paralelos al sentido positivo en **X+** de la base del robot, por lo que fue necesario crear una base (**base\_data[1]** y **base\_data[2]**) para cada escuadra que fuera paralelos al eje **X+** de desplazamiento, facilitando así las modificaciones posteriores de las cotas de dejada de saco.

### HARDWARE.

El **KR 180-2 PA** por sus características y materiales empleados, resulta ser el robot más idóneo para la paletización. El brazo principal y el brazo paralelo son de plástico reforzado con fibra de carbono especialmente ligero y robusto, si le añadimos que en su extremo el peso máximo es de 65Kgr (garra-25Kgr + Peso Saco máximo 40Kgr), se obtienen unas velocidades y aceleraciones óptimas en cada uno de los movimientos estando muy por debajo de los 180Kgr que es capaz de mover con una regulación óptima.

El sistema de agarre de sacos es la única limitación del robot en cuanto a la velocidad de paletizado. Esta mordaza permite la sujeción sin ejercer una presión sobre el mismo para no dañar el saco, pero a velocidades altas la inercia adquirida en el momento de arranque y parada puede desplazar el producto en su interior no quedando la carga homogénea en el momento de su depósito, y llegando incluso a correr el riesgo de que el saco salga disparado del sistema de agarre. Es por ello que para "**Piensos O' Couto**" no se recomienda superar una velocidad de paletizado en automático del 65%.



Como segundo elemento básico de control, el autómatas **TSX 37-10** permite una configuración libre de entradas y salidas mediante módulos y extensiones, siendo muy versátiles y adecuados para muy diferentes tipos de instalaciones. Disponen de un conector terminal serigrafiado **TER** que funciona de forma predeterminada en modalidad UNI-TELWAY maestro y permite conectar un equipo de diálogo de operador tipo XBT como el analizado en este proyecto.

Como desventaja podríamos decir que el espacio ocupado por el sistema de control es bastante voluminoso entre la CPU, los rack de extensión y módulos de entradas y salidas tipo telefax empleados, precisan de incorporar un armario eléctrico solamente para el control por autómatas. A parte nos encontramos con la necesidad de desconectar el terminal XBT para usar el PC de programación.

### DESARROLLO SOFTWARE.

El software de KUKA (**Vx-Works**); a pesar se poseer una sintaxis propia, es muy familiar, pues la lógica de programación es parecida a otros lenguajes existentes, y son pocos los conocimientos necesarios para su entendimiento. Básicamente existen dos partes bien diferenciadas a la hora de desarrollar un programa en (**Vx-Works**).

1. **Programación lógica.** Se programan todas las condiciones, bucles y activación de las salidas físicas, etc.
2. **Programación de trayectorias.** Intercaladas en la parte lógica del programa, se detallan todos los puntos con sus velocidades y

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

aceleraciones por donde ha de situarse el robot durante el programa para ejecutar cada una de las trayectorias.

El software desarrollado, permite el control automático mediante el PLC, esto dota a la instalación de un automatismo elevado, donde el operario requerirá una formación mínima ya que la instalación se manejará mediante los pulsadores de mando provistos en el armario eléctrico y el terminal de diálogo XBT, no siendo necesaria intervención alguna (salvo contadas ocasiones) en la KCP o consola del robot.

Los programas de paletización **MOSAICO\_3()**, **MOSAICO\_4()** y **MOSAICO\_5()** así como los programas de **Zona\_1 ()** y **Zona\_2()** para el posicionamiento de los sacos, son fácilmente configurables, y en caso de precisar alguna configuración más de paletizado; por ejemplo **MOSIACO\_6 ()** o **ZONA\_3 ()**, se pueden duplicar fácilmente y editar las modificaciones posteriormente, lo que dota a este sistema de rapidez en la creación de nuevos dibujos de paletizado.

Para el autómatas, se emplea el software de programación **PL7** que ofrece 4 lenguajes IEC: lista (IL), contactos (LD), estructurado (ST) y Grafcet (SFC). Para nuestro caso se ha desarrollado un sencillo programa en lenguaje (LD) de contactos, y esto es debido a la sencillez de la periferia que integra la instalación.

Como puntos destacados, mencionamos las dos secciones con más desarrollo que integran este software, siendo estas la parte para el manejo del robot mediante automático externo y la parte de comunicaciones con el terminal XBT. Esta última parte; y como hemos venido comentando a lo largo de este proyecto, esta pensado y desarrollado para abordar y configurar un mayor número de opciones desde el terminal XBT, como pueden ser diferentes configuraciones para varias líneas, un mayor número de ordenes de robot, etc. No habiendo sido necesarias para esta instalación.

Los problemas en este sentido a nivel software, es el especial cuidado que hay que poner en la configuración de los drivers de entrada y salida del robot para la comunicación con el PLC, y el desarrollo detallado del diagrama de señales adjunto, que es valido para la puesta en marcha de los diferentes tipos de KRC que nos podemos encontrar en el mercado. Queda un poco limitado el software del PLC, pues este solamente podrá ser usado con terminales como el desarrollado en este proyecto, ya que el envío de órdenes y datos a través del terminal es muy característico para este tipo de instalaciones.

## 8 ANEXO.A: (Programa multitarea Sps.sub).

```
DEF SPS ( )

  LOOP
    ;Make your modifications here

;=====CONTADORES A CERO=====

if $out[10]==true then
saco_1=1
base_data[3]=base_data[1]
endif
if $out[11]==true then
saco_2=1
base_data[4]=base_data[2]
endif

;=====SIMULTANEIDAD DE LINEA=====

if (($in[13]==true) and ($in[14]==false) and ($out[30]==true)) then
$out[30]=false
endif

if (($in[13]==false) and ($in[14]==true) and ($out[30]==false)) then
$out[30]=true
endif

;=====CAPTURA DE DATOS BINARIOS=====

;;;;;;;;;;;;;;Capas
if (($in[15]==true) and ($in[17]==false) and ($in[18]==true) and ($in[19]==false) and
($in[20]==false)) then
reg_capas_1=registro_1
endif

;;;;;;;;;;;;;;Incrementos

if (($in[15]==true) and ($in[17]==false) and ($in[18]==false) and ($in[19]==false) and
($in[20]==true)) then
if $in[24]==false then
reg_alt_incr_1=registro_2
altura_1=reg_alt_incr_1*5
endif
if $in[24]==true then
reg_alt_incr_1=0-registro_2
altura_1=reg_alt_incr_1*5
endif
endif

;=====RESTOS=====

if $in[25]==true then
$out[25]=true
endif

;=====Paro temporizador=====

if $in[7]==false then
$timer_stop[1]=true
endif

  ;ENDFOLD (USER PLC)
  ENDLOOP
  ;FOLD ;%{H}
  ;FOLD
END
```



## 9 ANEXO.B: (Programa Automático externo).

### 9.1 "CELL.SRC"

```
DEF CELL ( )
EXT mosaico_5 ( )
EXT mosaico_4 ( )
EXT mosaico_3 ( )
EXT home ( )

;FOLD INIT
DECL CHAR DMY[3]
DMY[]="---"
;ENDFOLD (INIT)
;FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
home ( )
;FOLD CHECK HOME
$H_POS=XHOME
IF CHECK_HOME==TRUE THEN
  P00 (#CHK_HOME,#PGNO_GET,DMY[],0 ) ;Testing Home-Position
ENDIF
;ENDFOLD (CHECK HOME)
;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%MKUKATPBASIS,%CMOVE,%VPTP,%P
1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
;FOLD AUTOEXT INI
P00 (#INIT_EXT,#PGNO_GET,DMY[],0 ) ; Initialize extern mode
;ENDFOLD (AUTOEXT INI)
LOOP
P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
SWITCH PGNO ; Select with Programnumber

CASE 1
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_5 ( ) ; Call User-Program

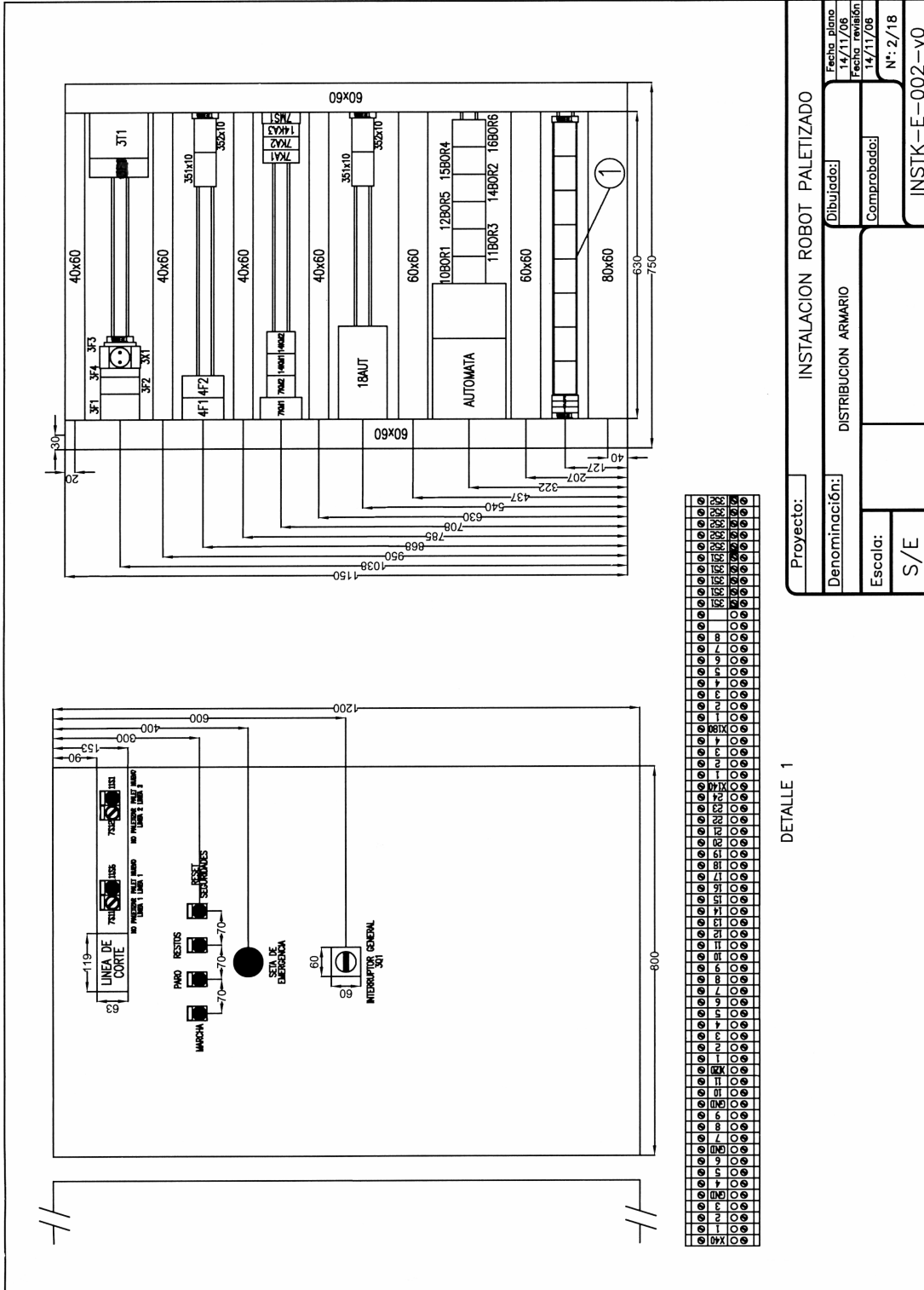
CASE 2
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_4 ( ) ; Call User-Program

CASE 3
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_3 ( ) ; Call User-Program

;DEFAULT
;P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
ENDSWITCH
ENDLOOP
END
```



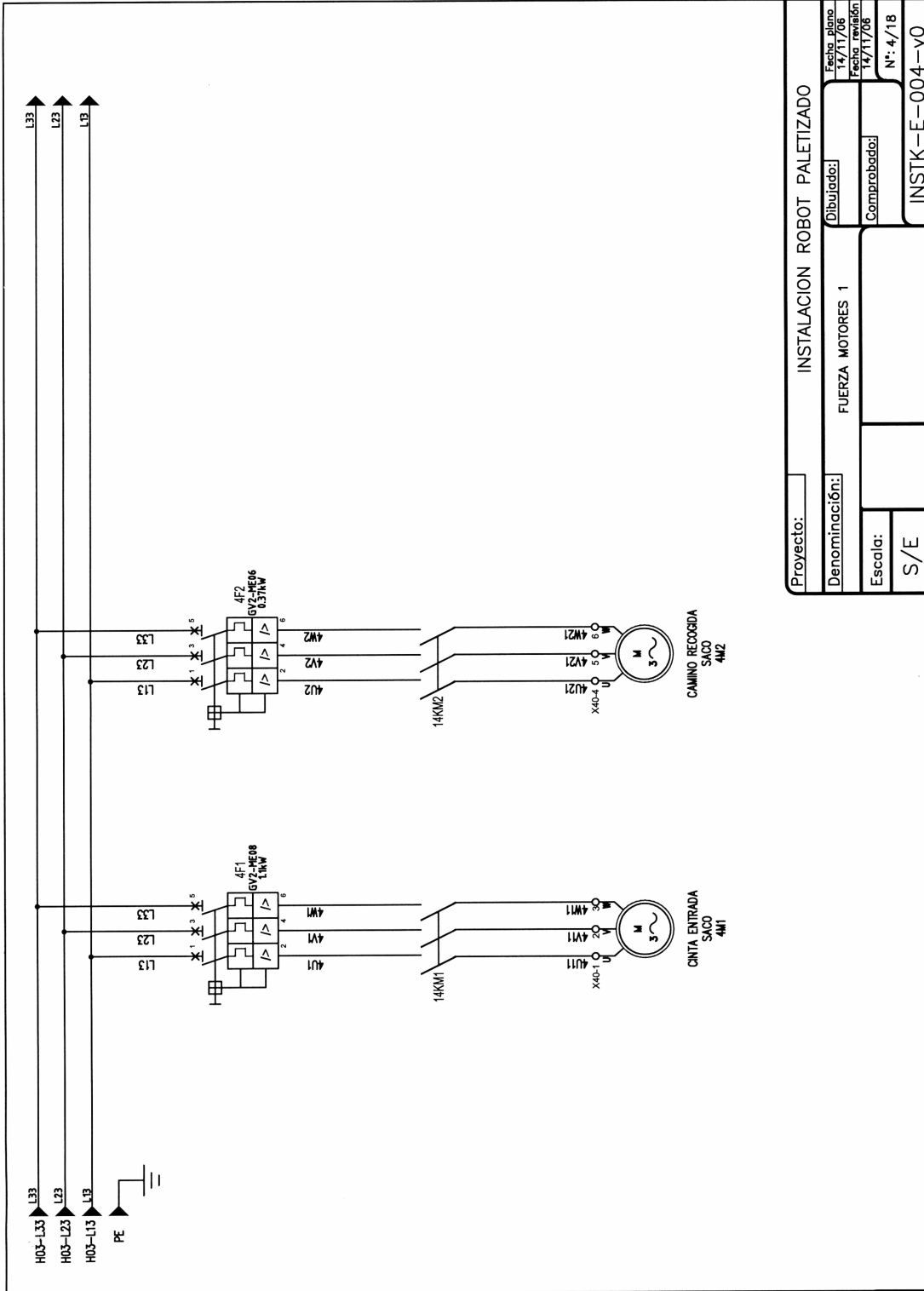




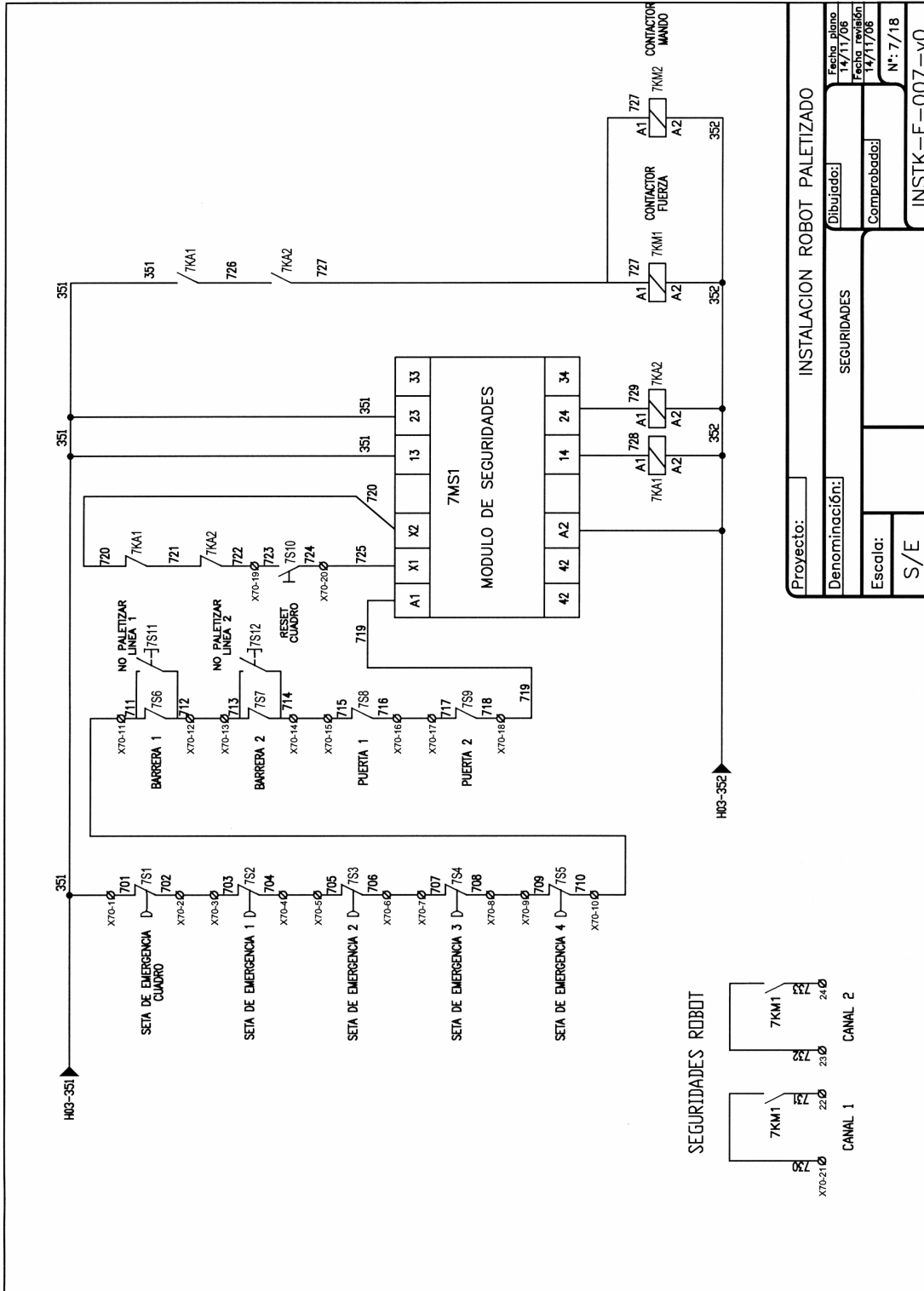
Proyecto:	INSTALACION ROBOT PALETIZADO		
Denominación:	DISTRIBUCION ARMARIO		
Fecha plano:	14/11/06	Dibujado:	
Fecha revisión:	14/11/06	Comprobado:	
Nº:	2/18	INSTK-E-002-v0	
Escala:	S/E		

DETALLE 1

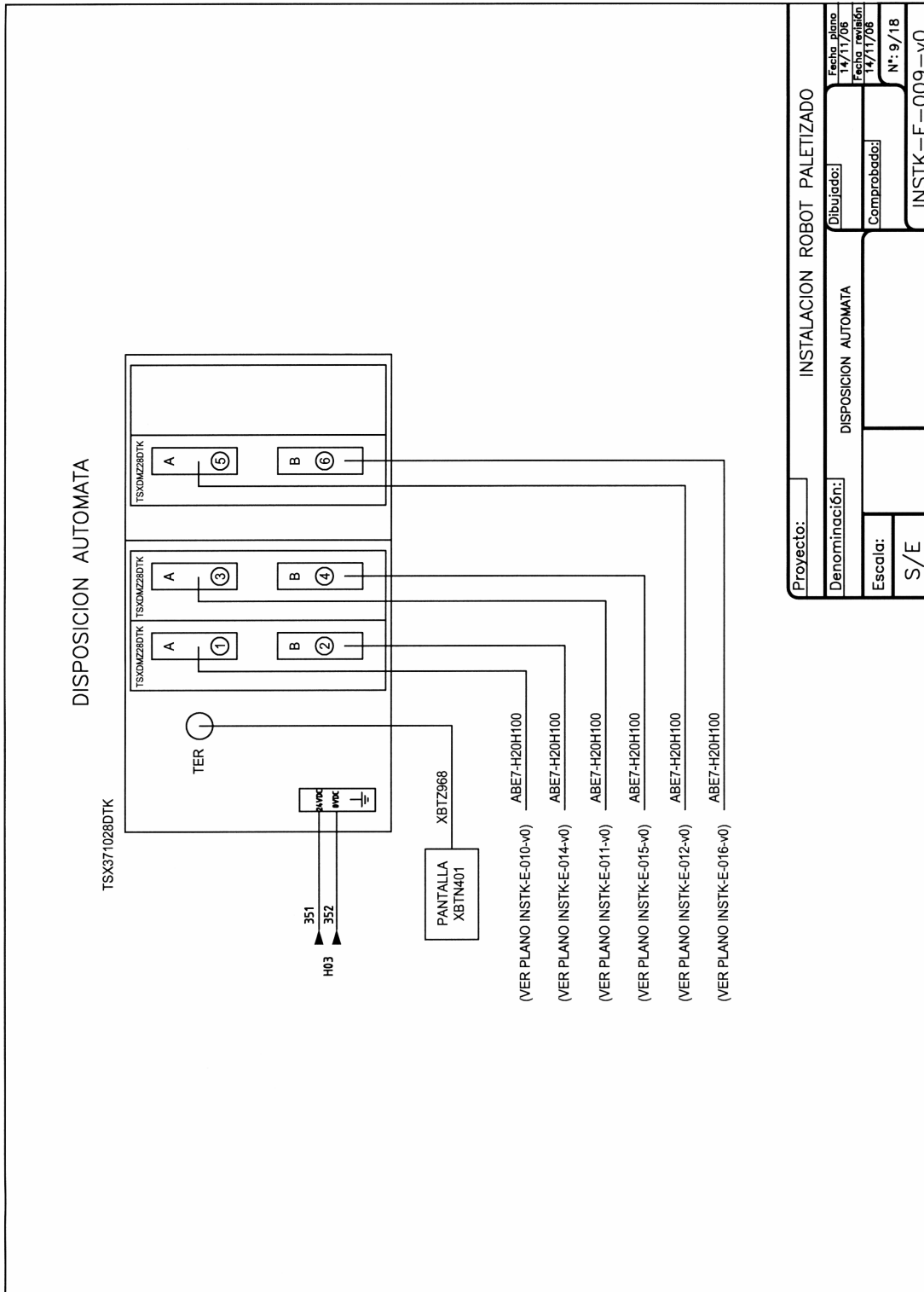




Proyecto:	INSTALACION ROBOT PALETIZADO		
Denominación:	FUERZA MOTORES 1		
Escala:	S/E	Dibujado:	Fecha plano: 14/11/06
		Comprobado:	Fecha revisión: 14/11/06
			Nº: 4/18
			INSTK-E-004-v0



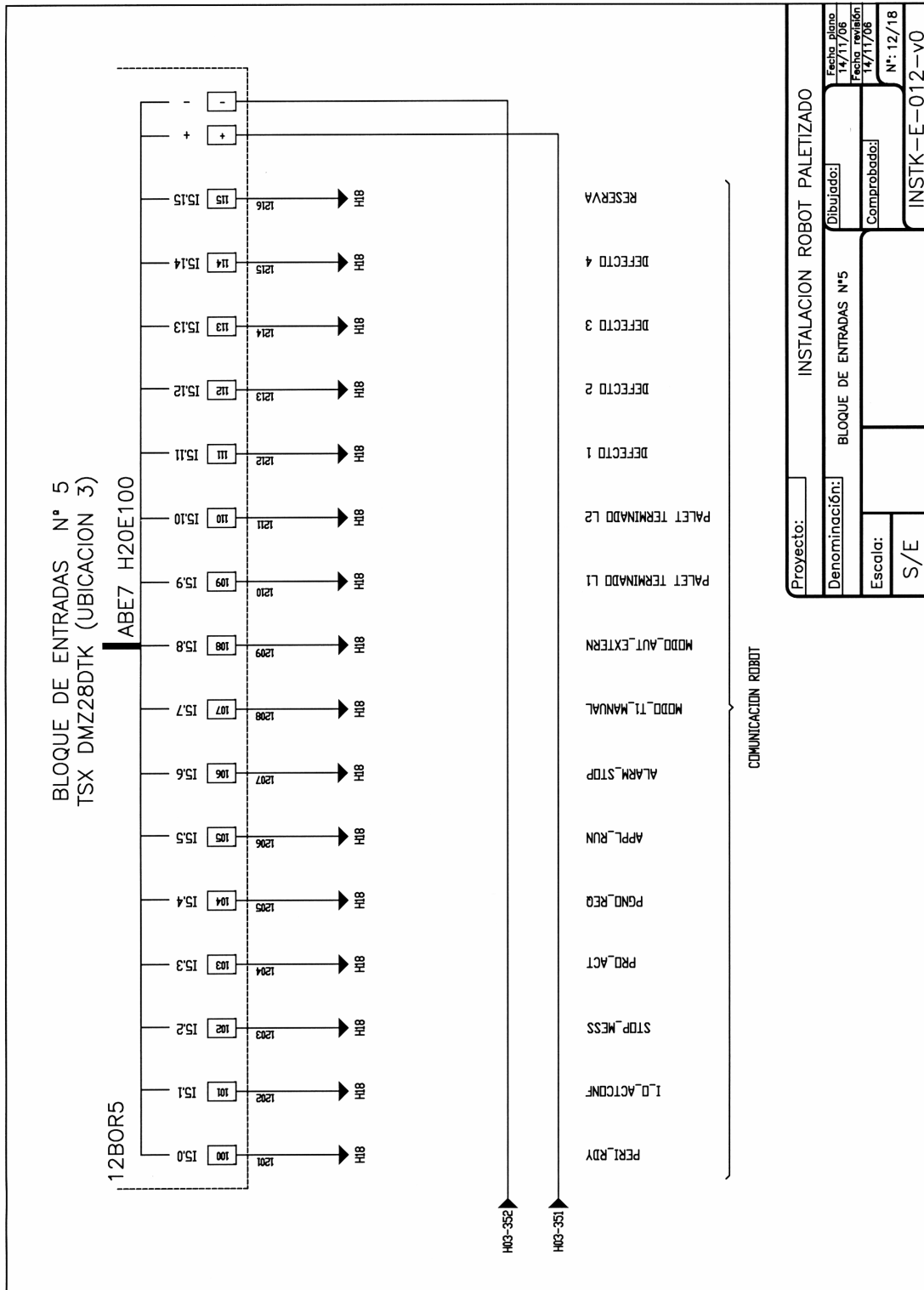
Proyecto:	INSTALACION ROBOT PALETIZADO		
Denominación:	SEGURIDADES		
Escala:	S/E	Dibujado:	14/11/08
		Comprobado:	14/11/08
			Nº: 7/18
			INSTK-E-007-V0



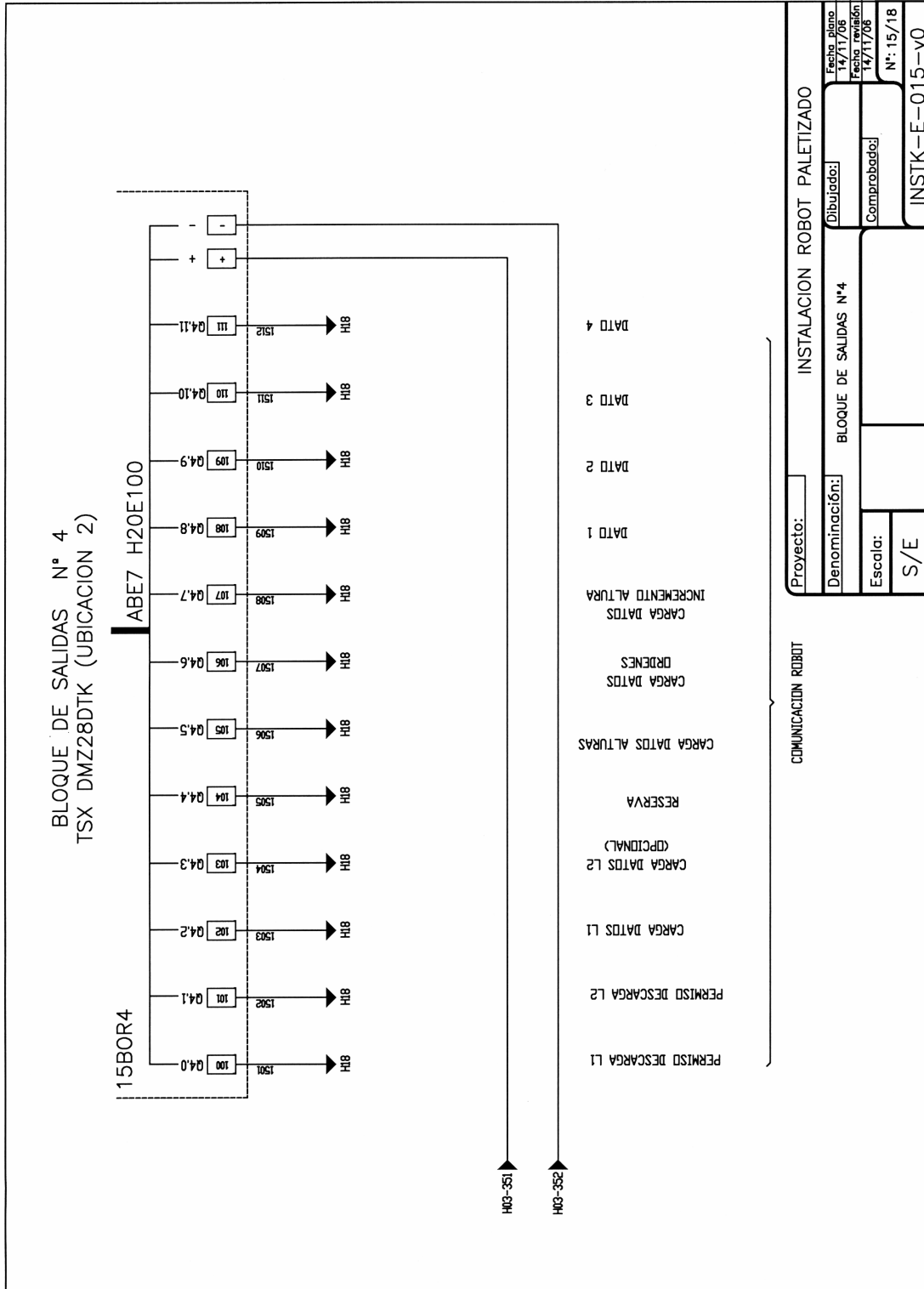




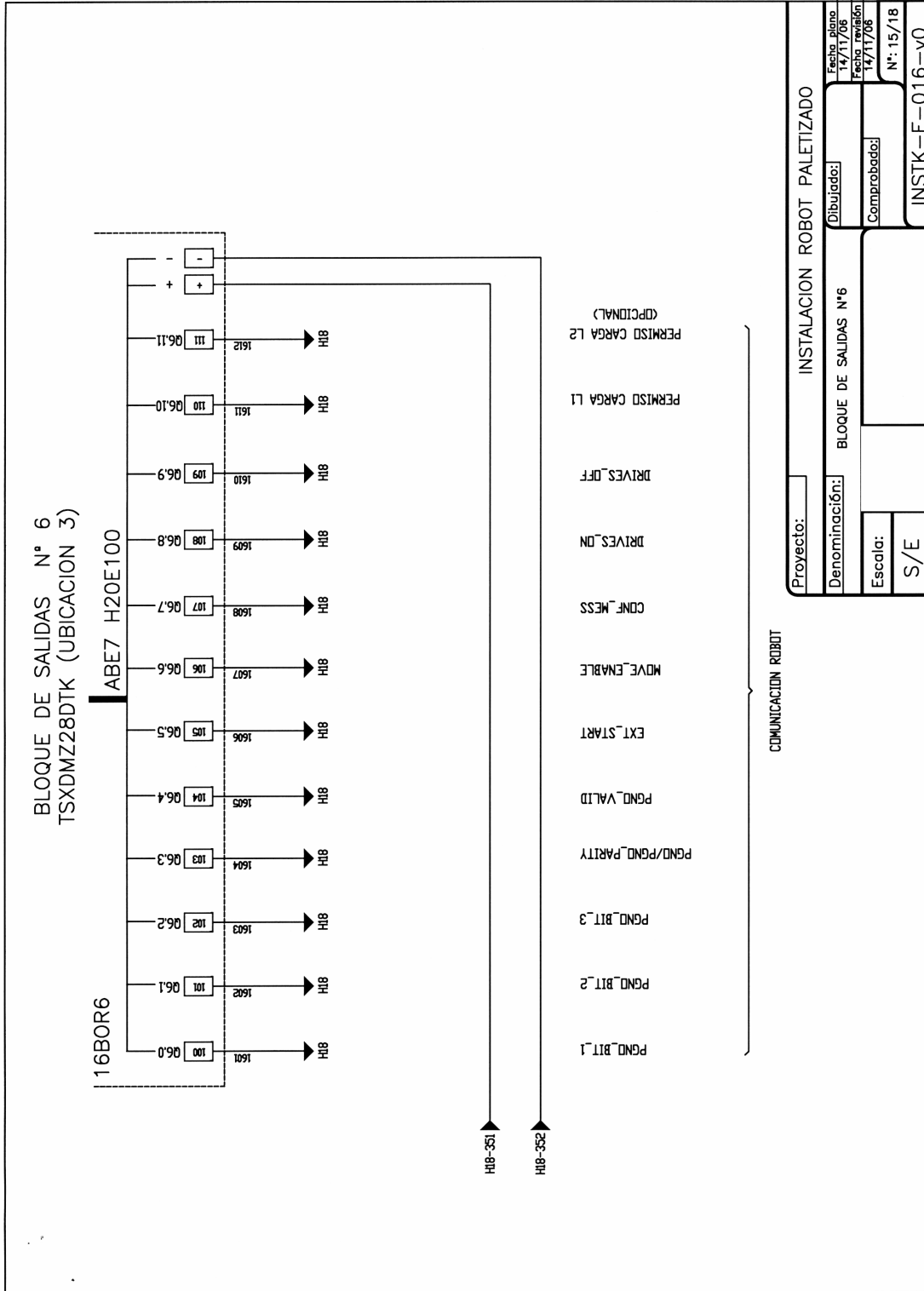


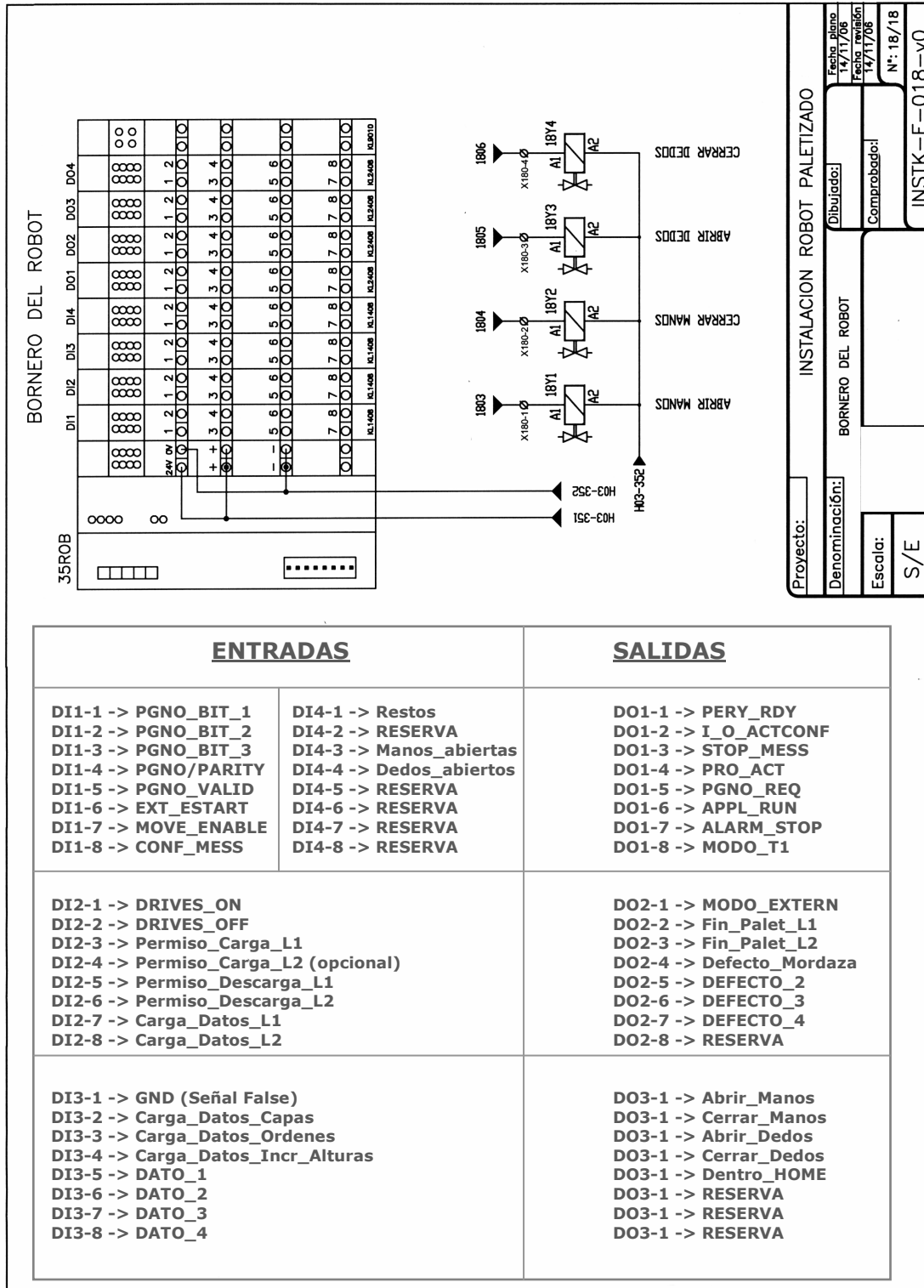








Proyecto:		INSTALACION ROBOT PALETIZADO	
Denominación:	BLOQUE DE SALIDAS N°4		
Escala:	S/E	Dibujado:	Comprobado:
		Fecha plano:	14/11/06
		Fecha revisión:	14/11/06
		N°:	15/18
		INSTK-E-015-v0	





	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

## 11 ANEXO.D: (Programa del PLC).

### 11.1 VARIABLES.

#### 11.1.1 Bit(s) Internos(s)



VARIABLE	SIMBOLO	COMENTARIO
%M0	Seguridades_activadas	
%M1	Instalacion_en_marcha	
%M4	Marcha_robot	Acciona la marcha del Robot paletizado, con el general actuando
%M5	Rearme_robot	Me proporciona los accionamientos y el reset de la interface de comunicaciones en caso de producirse una parada de emergencia
%M6	Petición_programa	Se activa con la petición del programa del KRC Pgno_req del automático externo
%M7	Sigue_peticion	
%M8	Filtrado_cel_recogida	
%M9	Filtrado_cel_rodillo	
%M10	Sigue_camino_recogida	
%M11	Se_va_a_enviar_dato_I1_a_robot	
%M12	Se_va_a_enviar_dato_I2_a_robot	
%M40	Aux_camino_recogida_saco_I1	
%M42	Aux_permiso_dejar_saco_I1	
%M43	Aux_permiso_dejar_saco_I2	
%M50	Marca_reset_contadores	

#### 11.1.2 Palabra(s) Internas(s).

VARIABLE	SIMBOLO	COMENTARIO
%MW0	Inc_capas_por_menos_1	Incremento de capas linea multiplicado por menos 1 para los datos negativos
%MW1	Mag_mosaico	Mosaico lineas
%MW2	Mag_capas	Capas lineas
%MW3	Mag_inc_capas	Incremento de altura lineas
%MW4	Inc_capas_I2_por_menos_1	Incremento de capas linea 1 multiplicado por menos 1 para los datos negativos (RESERVA LINEA 2)
%MW5	Mag_mosaico_I2	Mosaico linea 2 (RESERVA LINEA 2)
%MW6	Mag_capas_I2	Capas linea 2 (RESERVA LINEA 2)
%MW7	Mag_inc_capas_I2	Incremento de altura linea 2 (RESERVA LINEA 2)
%MW8	Sacos_totales	Sacos totales linea
%MW9	Palet_totales	Palets totales de linea
%MW15	Mag_automatiko_man	
%MW15:X0	Robot_automático_externo	
%MW18	Programa_magelis	Numero de mosaico en ejecución



%MW18:X0	Programa_magelis_bit_1	
%MW18:X1	Programa_magelis_bit_2	
%MW18:X2	Programa_magelis_bit_3	
%MW20		
%MW20:X0	Mag_cambiar_tipo_mosaico	Magelis: Cambiar tipo de mosaico linea
%MW20:X1	Mag_cambiar_capas	Magelis: Cambiar capas linea
%MW20:X2	Mag_incrementar_alturas	Magelis: Cambiar incrementar alturas linea
%MW20:X3	Mag_cambiar_tipo_mosaico_I2	Magelis: Cambiar tipo de mosaico linea 2 (RESERVA LINEA 2)
%MW20:X4	Mag_cambiar_capas_I2	Magelis: Cambiar capas linea 2 (RESERVA LINEA 2)
%MW20:X5	Mag_incrementar_alturas_I2	Magelis: Cambiar incrementar alturas linea 2 (RESERVA LINEA 2)
%MW20:X6	Mag_reset_contadores	Magelis: Ponemos a cero los contadores de producción.
%MW21		
%MW21:X0	Mag_robot_home	Magelis: Orden de robot 1
%MW21:X1	Mag_robot_mantenimiento	Magelis: Orden de robot 2
%MW21:X2	Mag_robot_liberación	Magelis: Orden de robot 3
%MW21:X3	Mag_robot_orden4	Magelis: Orden de robot 4 (RESERVA)
%MW21:X4	Mag_robot_orden5	Magelis: Orden de robot 5 (RESERVA)
%MW21:X5	Mag_robot_orden6	Magelis: Orden de robot 6 (RESERVA)
%MW21:X6	Mag_robot_orden7	Magelis: Orden de robot 7 (RESERVA)
%MW21:X7	Mag_robot_orden8	Magelis: Orden de robot 8 (RESERVA)
%MW21:X8	Mag_robot_orden9	Magelis: Orden de robot 9 (RESERVA)
%MW21:X9	Mag_robot_orden10	Magelis: Orden de robot 10 (RESERVA)
%MW21:X10	Mag_robot_orden11	Magelis: Orden de robot 11 (RESERVA)
%MW21:X11	Mag_robot_orden12	Magelis: Orden de robot 12 (RESERVA)
%MW21:X12	Mag_robot_orden13	Magelis: Orden de robot 13 (RESERVA)
%MW21:X13	Mag_robot_orden14	Magelis: Orden de robot 14 (RESERVA)
%MW22		
%MW22:X0	Mag_manual_cinta_entrada	
%MW22:X2	Mag_manual_mesa_recogida	
%MW22:X3	Mag_manual_bajar_rodillo	
%MW31	Orden_a_enviar_al_robot	
%MW118	Robot_defectos	
%MW118:X0	Robot_en_marcha	
%MW118:X1	Interface_preparada	
%MW118:X2	Accionamientos_preparados	
%MW119	Periferia_defectos	
%MW119:X0	Defecto_térmico_motores	
%MW119:X1	Defecto_seguridades	
%MW119:X2	Defecto_presión_aire	
%MW119:X3	Defecto_mordaza	



	<i>Proyecto</i> <b>PROYECTO FIN DE CARRERA</b>	
	<i>Documento</i> <b>Instalación de Robot Paletizado.</b>	

### 11.1.3 Temporizador(es).

VARIABLE	SIMBOLO	PRESET	MODO	TB	REG
%TM0		15	TON	10 ms	SI
%TM1	T_filtrado_bajar_rodillo	15	TON	10 ms	SI
%TM2	T_filtrado_cel_recogida	15	TON	10 ms	SI
%TM3	T_espera_dato_robot_l1	20	TON	10 ms	SI
%TM4	T_espera_dato_robot_l2	20	TON	10 ms	SI
%TM5	T_envio_dato_robot_l1	15	TON	100 ms	SI
%TM6	T_envio_dato_robot_l2	15	TON	100 ms	SI
%TM7	T_señal_mantenida_l1	10	TP	100 ms	SI
%TM8	T_señal_mantenida_l2	10	TP	100 ms	SI
%TM25	T_activar_drives_on	8	TON	100 ms	SI
%TM26	T_pulso_drives_on	2	TON	10 ms	SI
%TM27	T_conf_mess	2	TON	10 ms	SI
%TM28	T_pro_act	2	TON	10 ms	SI
%TM29	T_peticion_programa	2	TON	10 ms	SI
%TM30	T_pgno_valid	3	TON	10 ms	SI
%TM31	T_desactivacion	3	TON	10 ms	SI
%TM32	T_alarma_ioact	10	TON	1 s	SI
%TM34	T_paro_l1	15	TON	100 ms	SI
%TM63	T_camino_recogida_saco	10	TON	1 s	SI

### 11.1.4 Módulo de entradas\_1.

VARIABLE	SIMBOLO	COMENTARIO
%I1.0	Di_boton_paro	Botón paro instalación
%I1.1	Di_boton_marcha	Botón marcha instalación
%I1.2	Reserva	
%I1.3	Di_boton_reset_alarmas	Botón reset alarmas
%I1.4	Di_seguridades	Entrada de seguridades del modulo de seguridades
%I1.5	Di_presion_aire	Entrada presion de aire de la maquina
%I1.6	Di_serie_termicos	Serie de los termicos de la instalacion
%I1.9	Di_presencia_palet_lleno_l1	Presencia palet lleno en la linea 1
%I1.10	Di_presencia_palet_lleno_l2	Presencia de palet lleno en la linea 2
%I1.11	Di_celula_recogida_saco	Celula de mesa de recogida de saco
%I1.12	Di_entrada_saco_saturacion	Celula de la cinta de entrada del saco para conseguir saturacion
%I1.13	Di_robot_no_actua_l1	El robot no paletiza en la linea de paletizado 1
%I1.14	Di_robot_no_actua_l2	El robot no paletiza en la linea de paletizado 2
%I1.15	Di_boton_palet_nuvo_l1	Palet nuevo puesto en la zona de paletizado

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

### 11.1.5 Módulo de entradas\_3.



VARIABLE	SIMBOLO	COMENTARIO
%I3.0	Di_boton_palet_nuevo_I2	palet nuevo colocado en la zona de paletizado esta vacio
%I3.1	Di_presencia_rodillo	
%I3.2	Di_paro_saturacion	
%I3.3	Dr_dentro_de_home	Si estoy dentro del HOME se puede cambiar el mosaico

### 11.1.6 Módulo de entradas\_5.

VARIABLE	SIMBOLO	COMENTARIO
%I5.0	Di_r_peri_rdy	La KRC me dice que los accionamientos estan preparados
%I5.1	Di_r_i_o_actconf	La KRC me confirma que la interface de E/S para comunicar mediante Ato.EXterno está activada.
%I5.2	Di_r_stop_mess	La KRC me informa que hay mensajes que confirmar, por paradas externas o inicio de sistema, o cualquier otro evento.
%I5.3	Di_r_pro_act	La KRC me comunica que hay un programa acxtivo o seleccionado, ya sea programa base cell o de paletizado.
%I5.4	Di_r_pgno_req	La KRC me solicita un numero de programa
%I5.5	Di_r_appl_run	Se está ejecutando un programa de paletizado en RUN en la KRC
%I5.6	Di_r_alarm_sto	Estando a nivel alto,el KRC me informa de haberse producido un evento de parada de emergencia (nivel bajo).
%I5.7	Di_r_modo_t1_manual	El KRC me indica mediante esta señal, que la llave se encuentra en el modo de operación MANUAL.
%I5.8	Di_r_modo_aut_externo	El KRC me indica mediante esta señal, que la llave se encuentra en el modo de operación Automático_Externo.
%I5.9	Di_palet_terminado_I1	el paletizado se ha terminado en la linea 1
%I5.10	Di_palet_terminado_I2	el paletizado se ha terminado en la linea 2
%I5.11	Di_defecto_mordaza	

### 11.1.7 Módulo de salidas\_2.

VARIABLE	SIMBOLO	COMENTARIO
%Q2.0	Do_cinta_entrada_saco_I1	cinta de entrada de saco de la linea 1
%Q2.1	Do_camino_recogida_saco_I1	camino de recogida de saco de linea 1
%Q2.2	Do_paro_I1	Paro línea 1
%Q2.3	Do_robot_en_marcha	Robot en marcha
%Q2.4	Do_bajar_rodillo	Rodillo para prensar el saco

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

### 11.1.8 Módulo de salidas\_4.

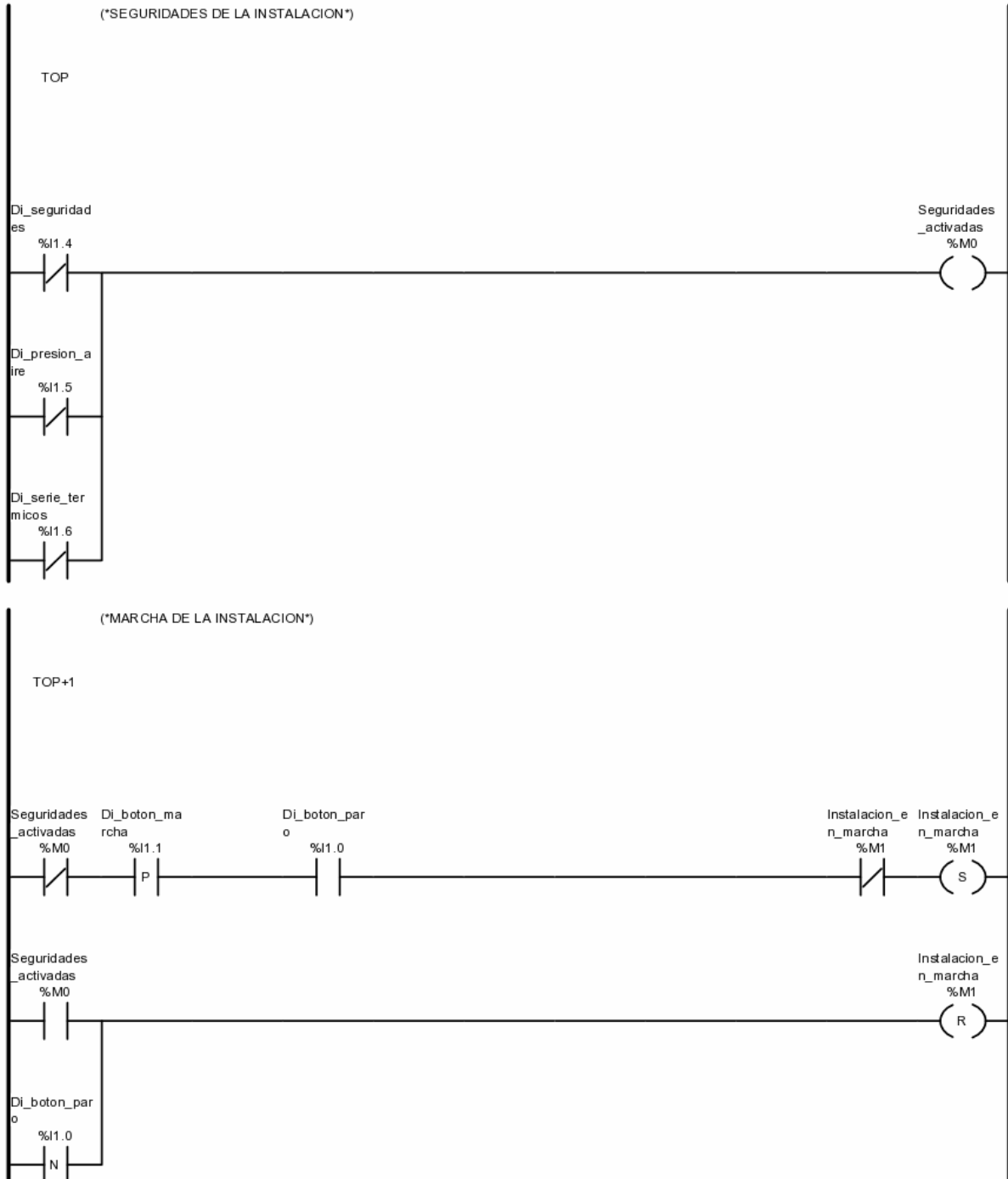
VARIABLE	SIMBOLO	COMENTARIO
%Q4.0	Do_permiso_descarga_l1	
%Q4.1	Do_permiso_descarga_l2	
%Q4.2	Do_robot_carga_datos	ROBOT: Carga datos linea
%Q4.3	Do_robot_carga_datos_l2	ROBOT: Carga datos linea 2 (OPCIONAL)
%Q4.4	Do_robot_carga_datos_mosaico	ROBOT: Carga datos de mosaico
%Q4.5	Do_robot_carga_datos_capas	ROBOT: Carga datos de numero de capas
%Q4.6	Do_robot_carga_datos_ordenes	ROBOT: Carga datos de ordenes
%Q4.7	Do_robot_carga_datos_incremento	ROBOT: Carga datos de incremento de alturas
%Q4.8	Do_robot_dato1	ROBOT: Dato 1
%Q4.9	Do_robot_dato2	ROBOT: Dato 2
%Q4.10	Do_robot_dato3	ROBOT: Dato 3
%Q4.11	Do_robot_dato4	ROBOT: Dato 4

### 11.1.9 Módulo de salidas\_6.

VARIABLE	SIMBOLO	COMENTARIO
%Q6.0	Do_r_pgno_bit_1	Bit 1 del código binario para determinar el programa
%Q6.1	Do_r_pgno_bit_2	Bit 2 del código binario para determinar el programa
%Q6.2	Do_r_pgno_bit_3	Bit 3 del código binario para determinar el programa
%Q6.3	Do_r_pgno_pgno_parity	Bit que representa la preasignación al envío de programa y la paridad en caso de estar habilitada
%Q6.4	Do_r_pgno_valid	Decimos al KRC que hemos transmitido el nº de programa, validando así el dato
%Q6.5	Do_r_ext_start	Señal para iniciar o continuar un programa en la interface E/S del KRC
%Q6.6	Do_r_move_enable	Habilitamos el movimiento del KRC, pudiendo realizarse desplazamientos manuales, ejecutar o parar un programa.
%Q6.7	Do_r_conf_mess	Confirmamos al KRC todos los tipos de mensajes confirmables una vez subsanada la causa que los produjo.
%Q6.8	Do_r_drives_on	Activamos accionamientos del KRC
%Q6.9	Do_r_drives_off	Desactivamos accionamientos del KRC
%Q6.10	Do_r_permiso_carga	

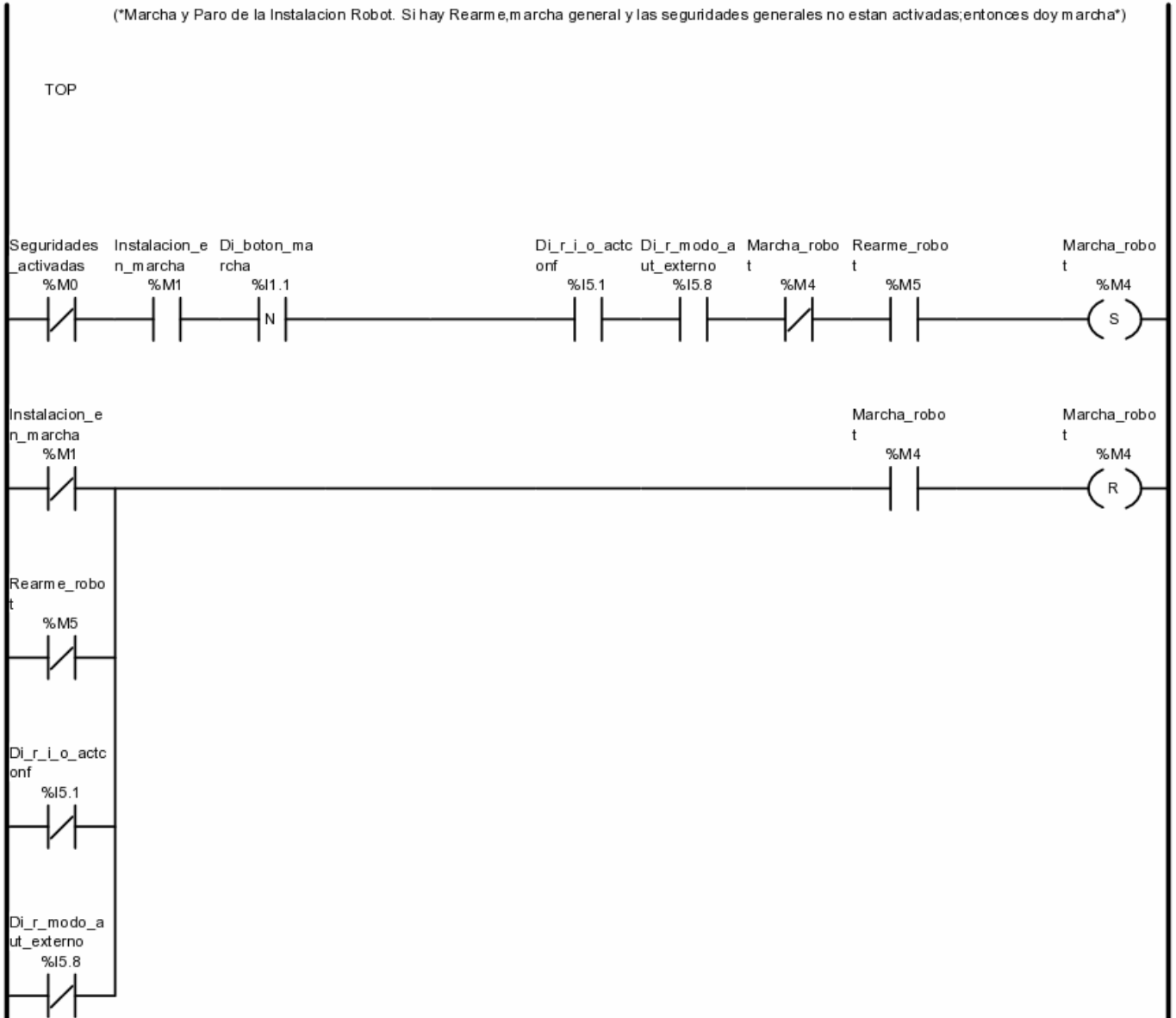
## 11.2 PROGRAMACIÓN EN CONTACTOS.

### 11.2.1 LD\_Mast\_General.

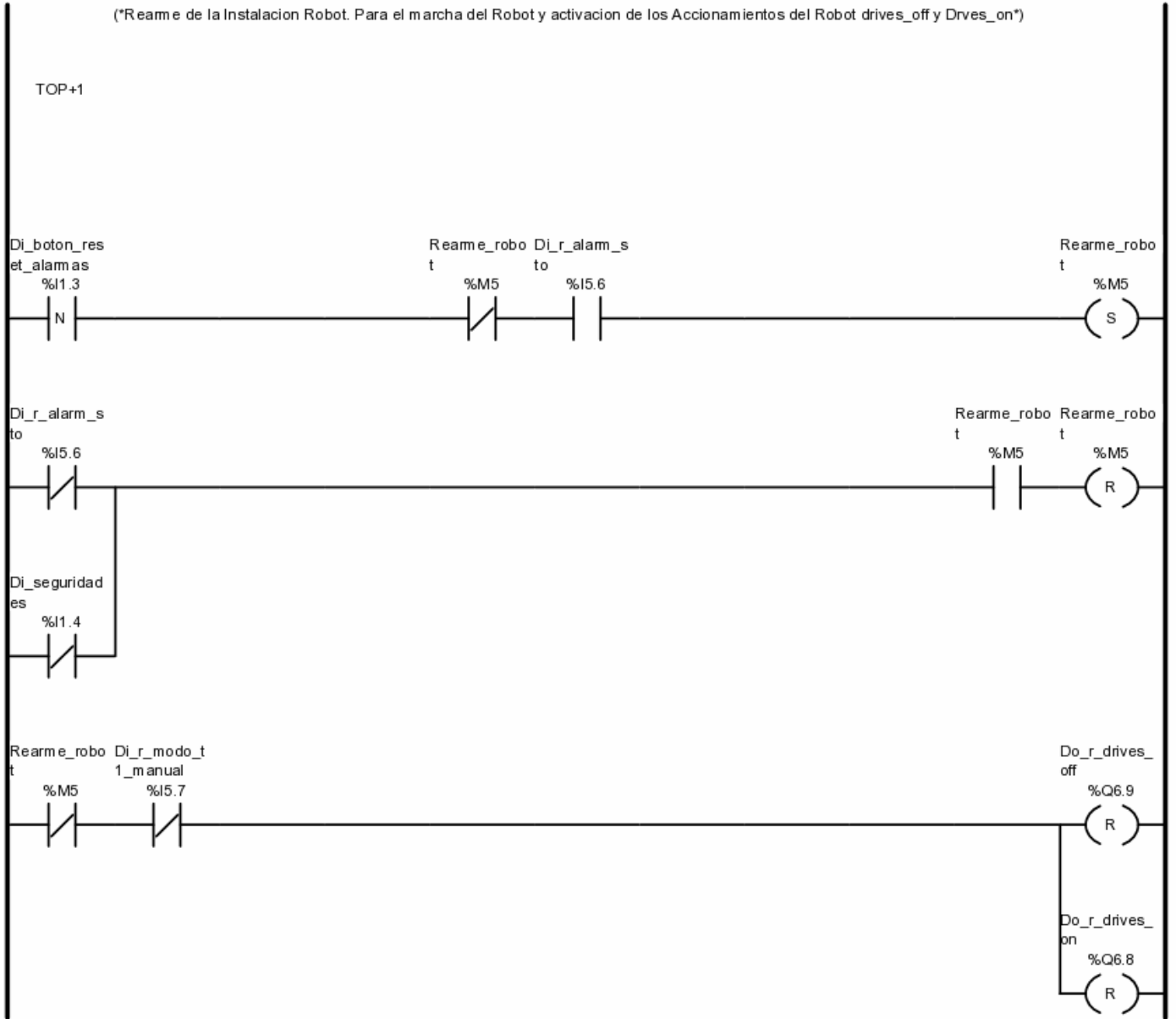


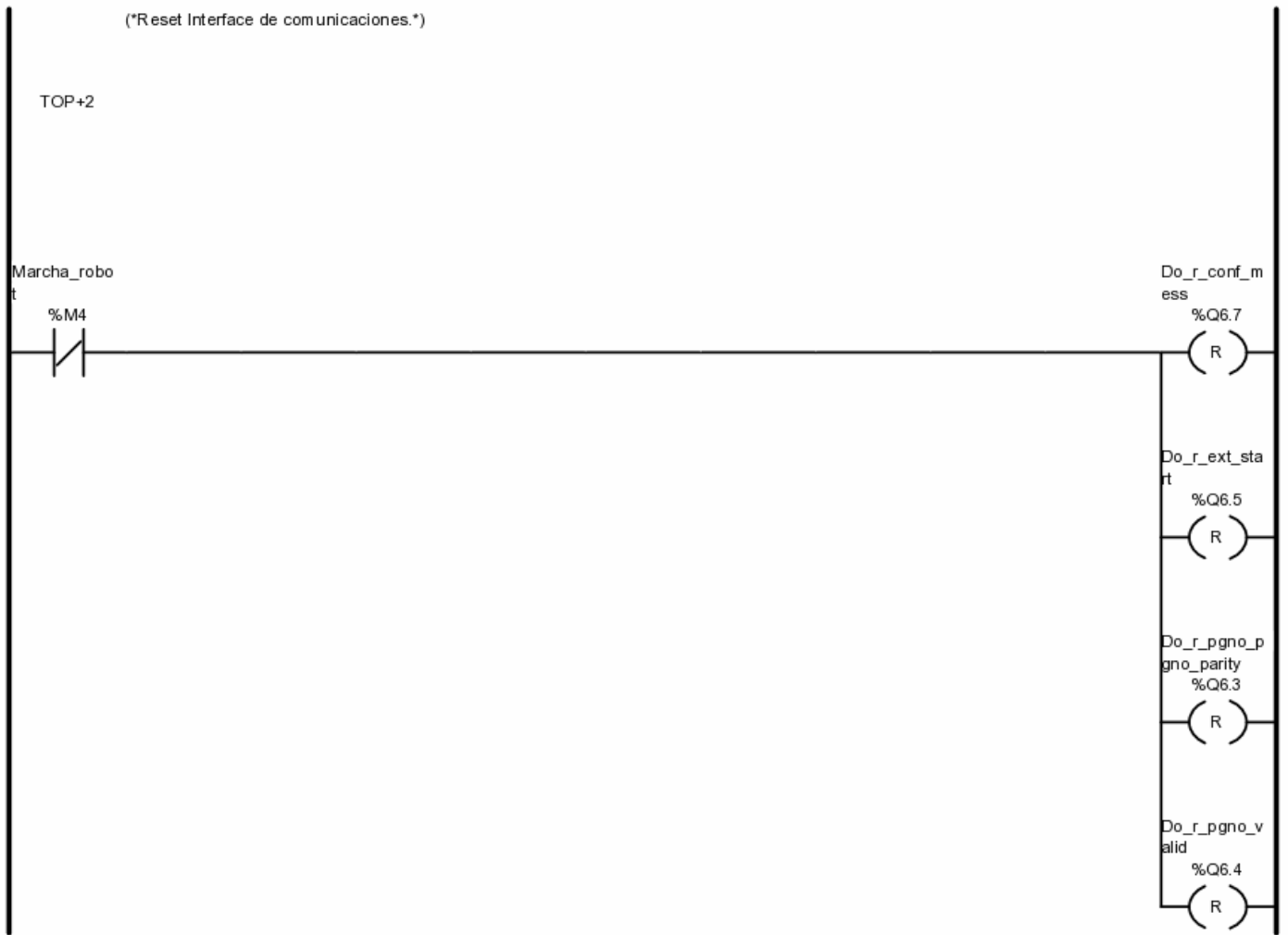
## 11.2.2 LD\_Mast\_Robot\_Auto\_Extern.

(\*Marcha y Paro de la Instalacion Robot. Si hay Rearme, marcha general y las seguridades generales no estan activadas; entonces doy marcha\*)



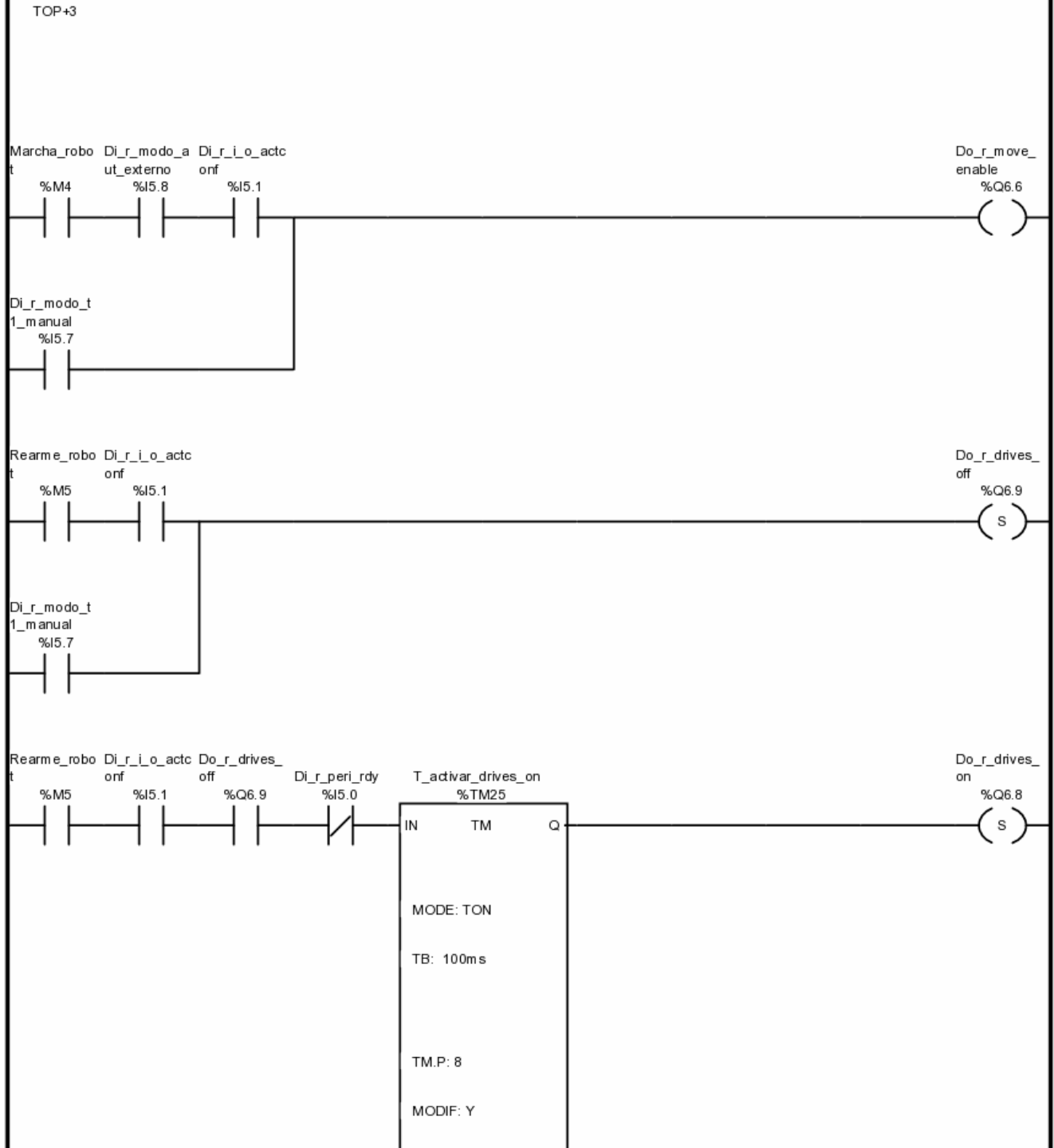
(\*Rearme de la Instalacion Robot. Para el marcha del Robot y activacion de los Accionamientos del Robot drives\_off y Drves\_on\*)

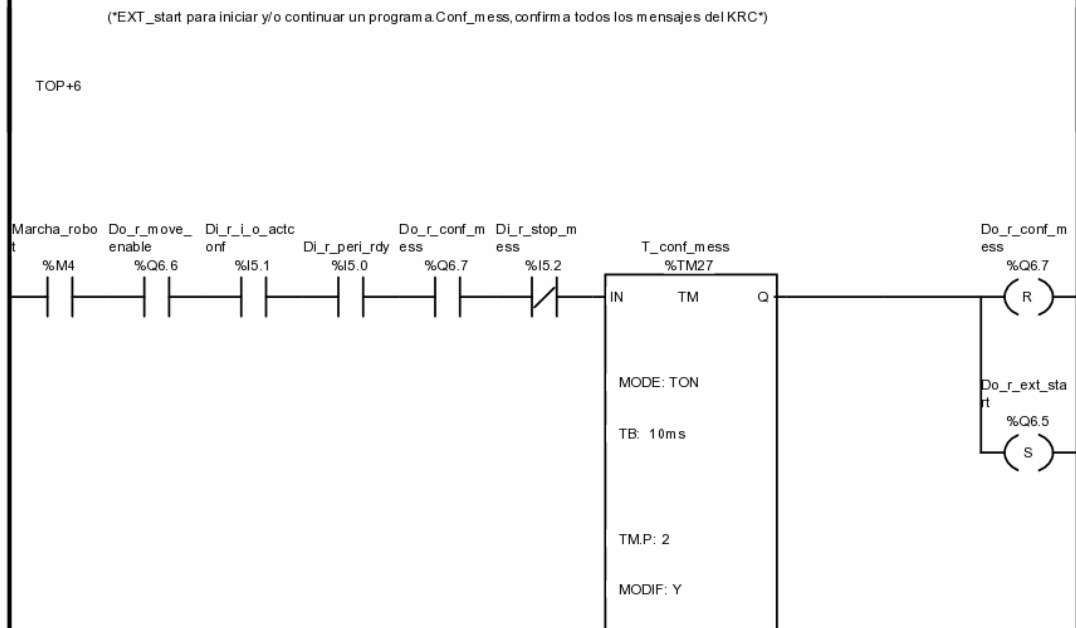
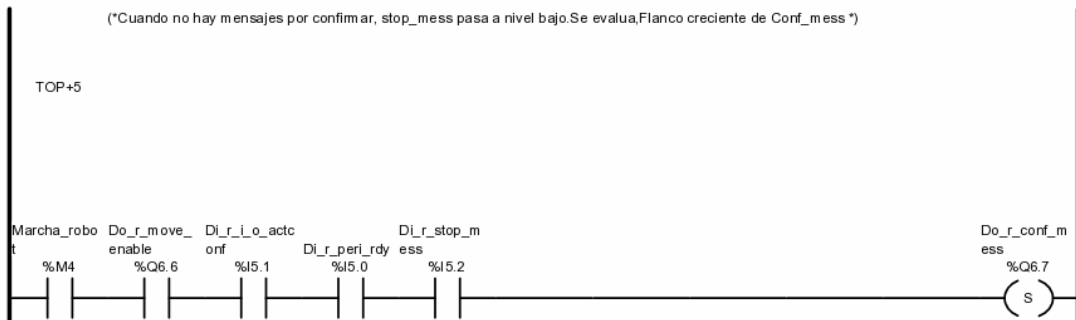
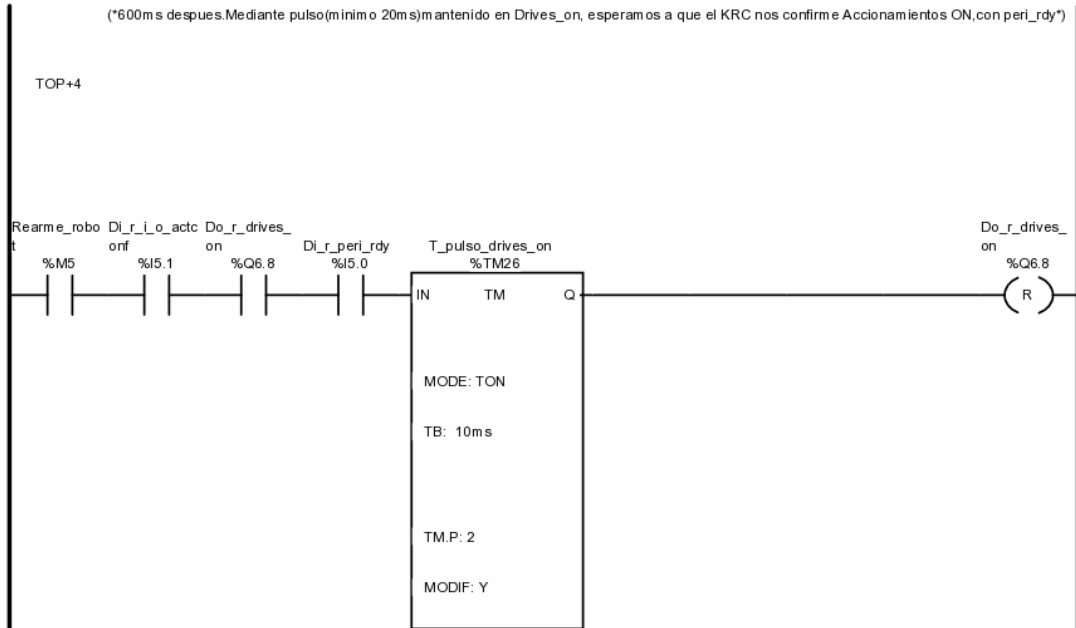






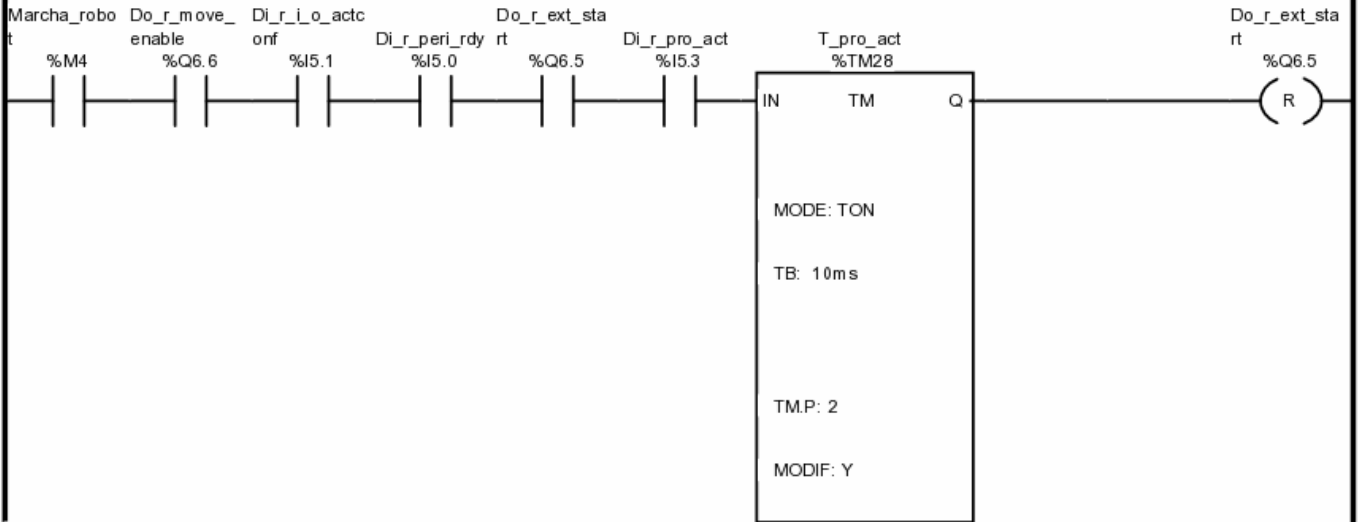
(\*Con la interface i\_o\_act activada y en modo Aut.Externo.Habilitamos señal Move\_enable para el movimiento del KRC, y damos Accionamientos\*)





(\*Mediante flanco creciente en ext\_start, ponemos en Run program a base cell del KRC, que me indica a través de Pro\_act que el proceso está a»(1)

TOP+7

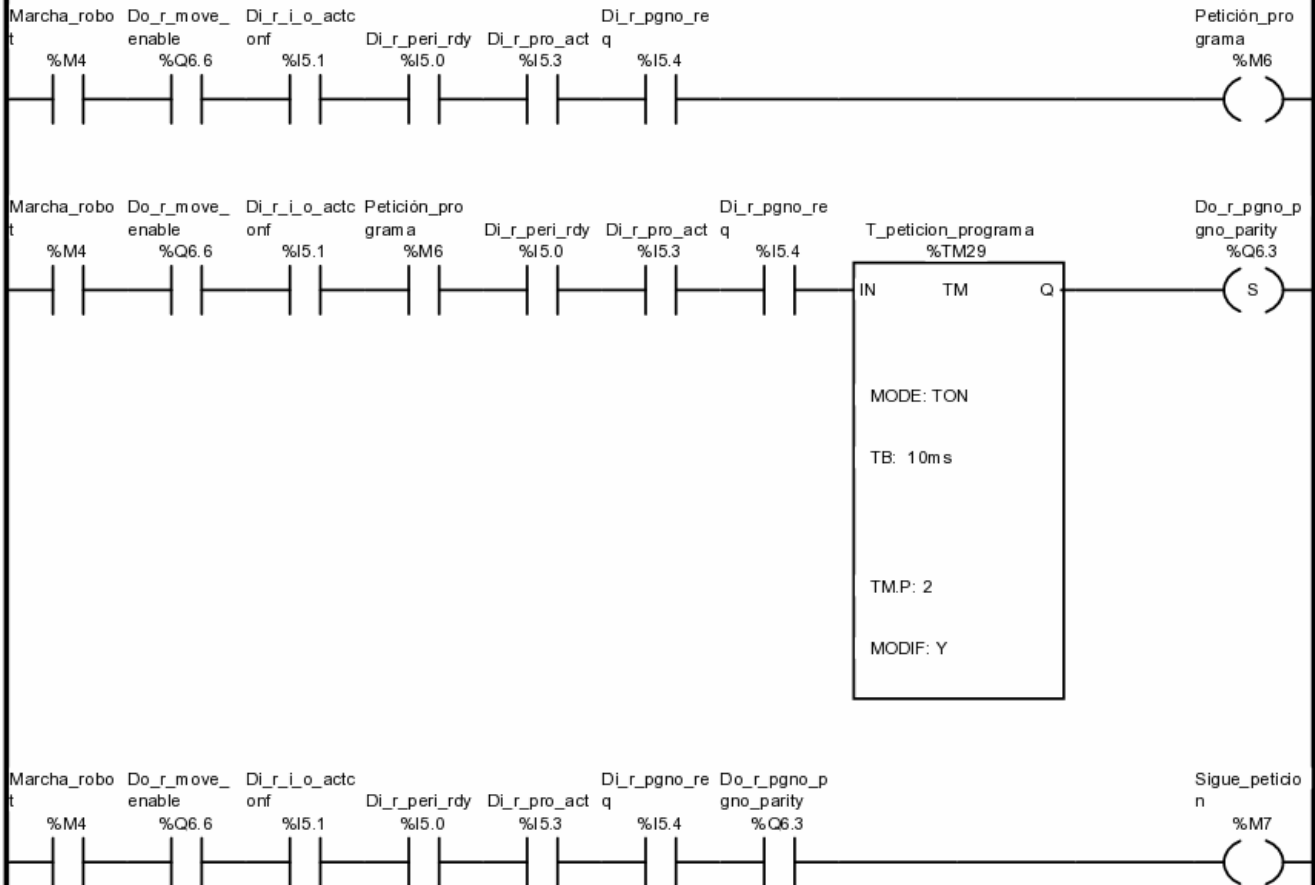


Lista de los reenvios del comentario:

(1):(\*Mediante flanco creciente en ext\_start, ponemos en Run program a base cell del KRC, que me indica a través de Pro\_act que el proceso está activo\*)

(\*Con el programa base Cell.Src activo, se nos genera la petición Pgno\_req para la emisión del numero de programa, y solicitud de Pgno\_Parity\*)

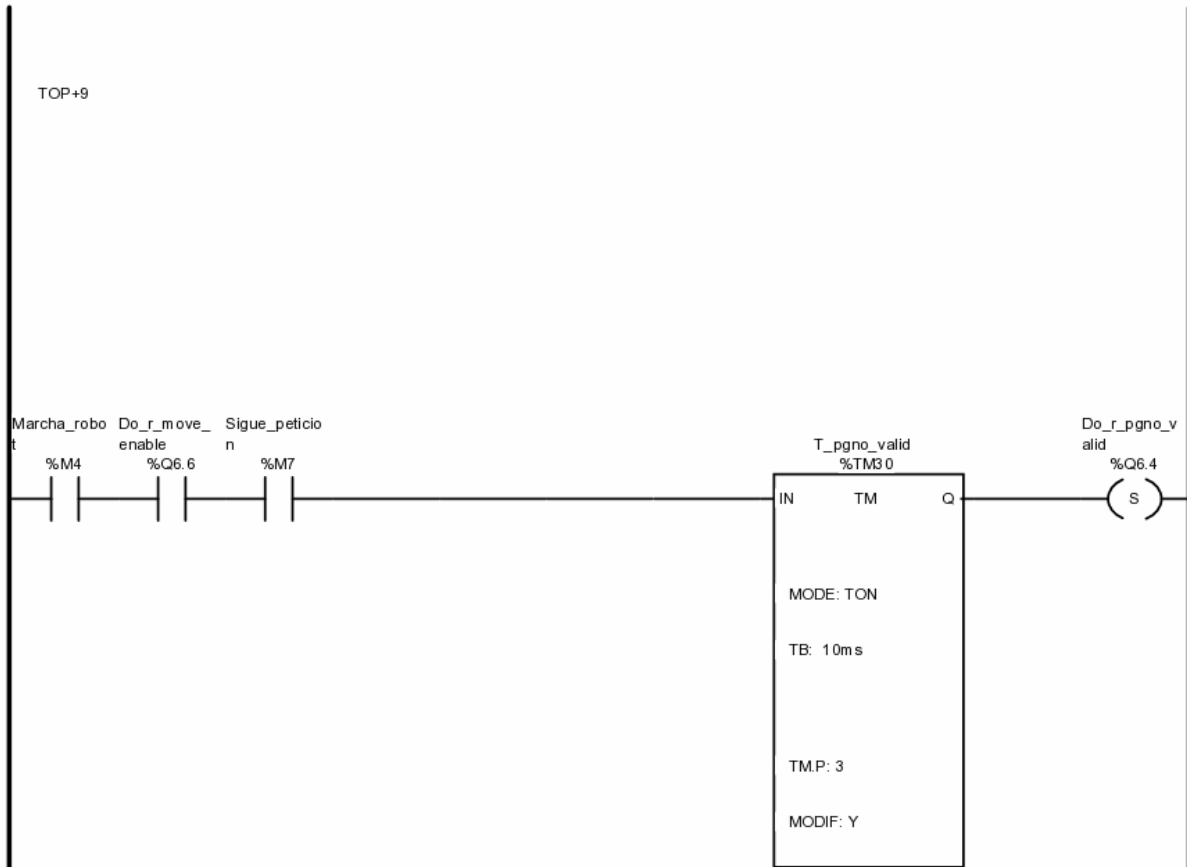
TOP+8



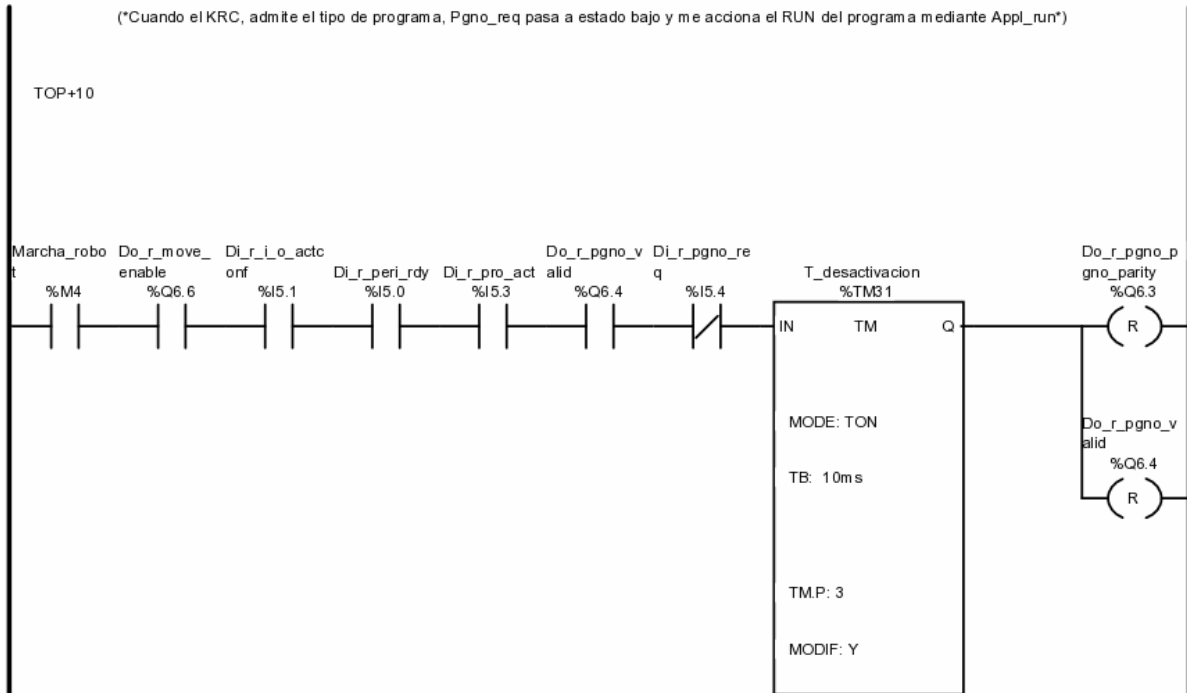
Carpeta: 7

MAST - Robot\_auto\_exter

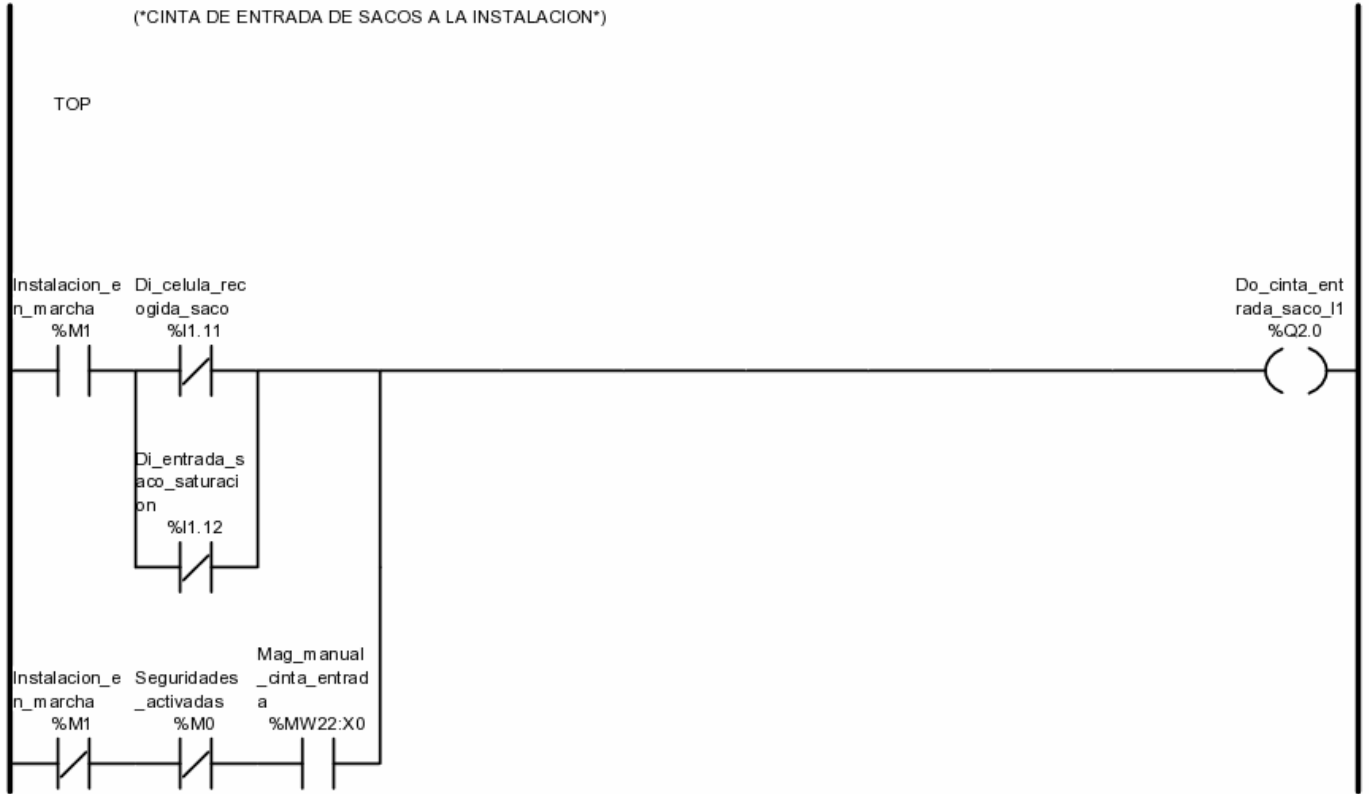
Fecha: 28/8/09



(\*Cuando el KRC, admite el tipo de programa, Pgno\_req pasa a estado bajo y me acciona el RUN del programa mediante Appl\_run\*)



### 11.2.3 LD\_Mast\_Recogida\_De\_Saco.



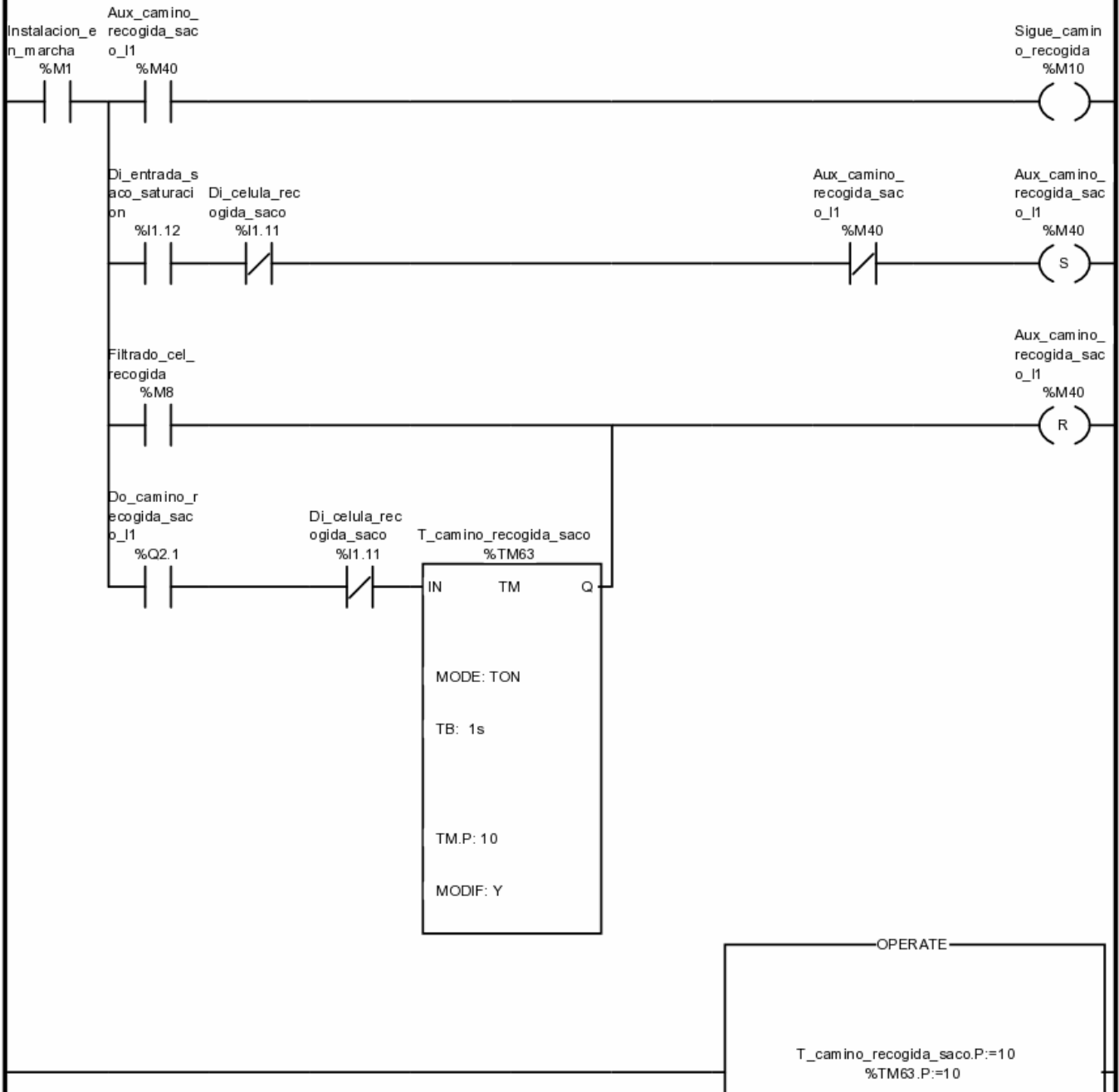
Carpeta: 1

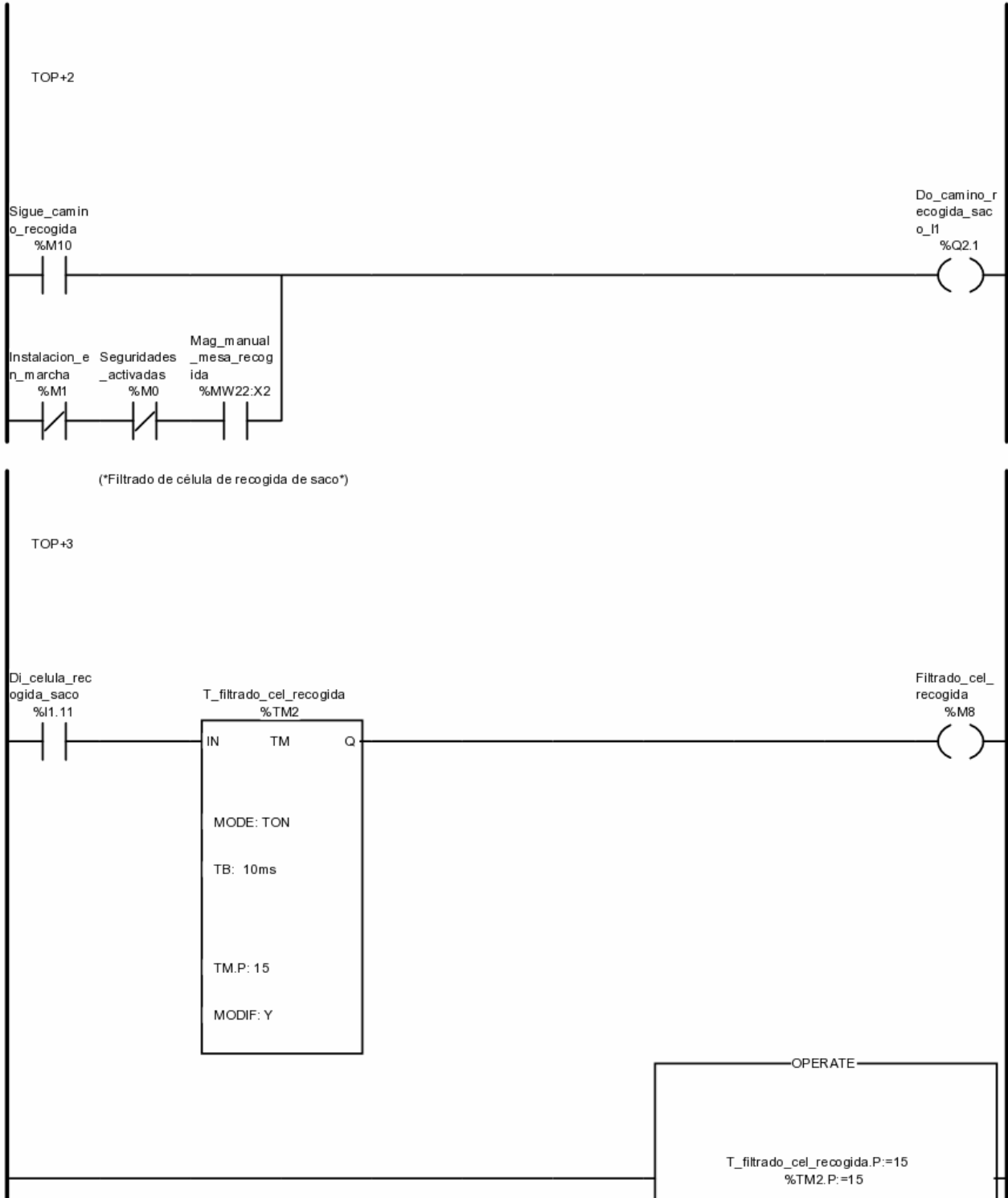
MAST - Recogida\_de\_saco

Fecha: 28/8/09

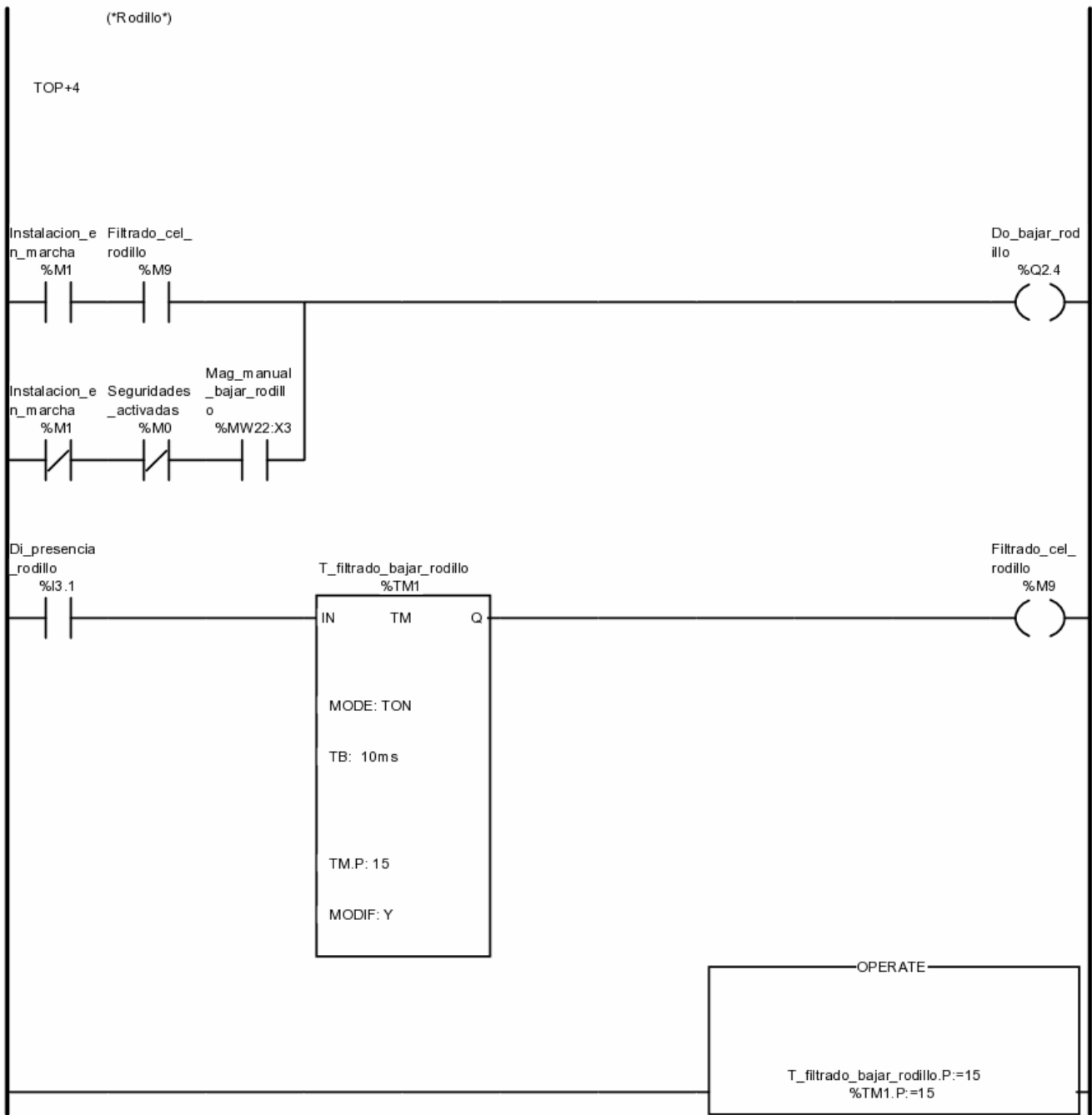
(\*CAMINO DE RECOGIDA DE SACO DE LA INSTALACION\*)

TOP+1

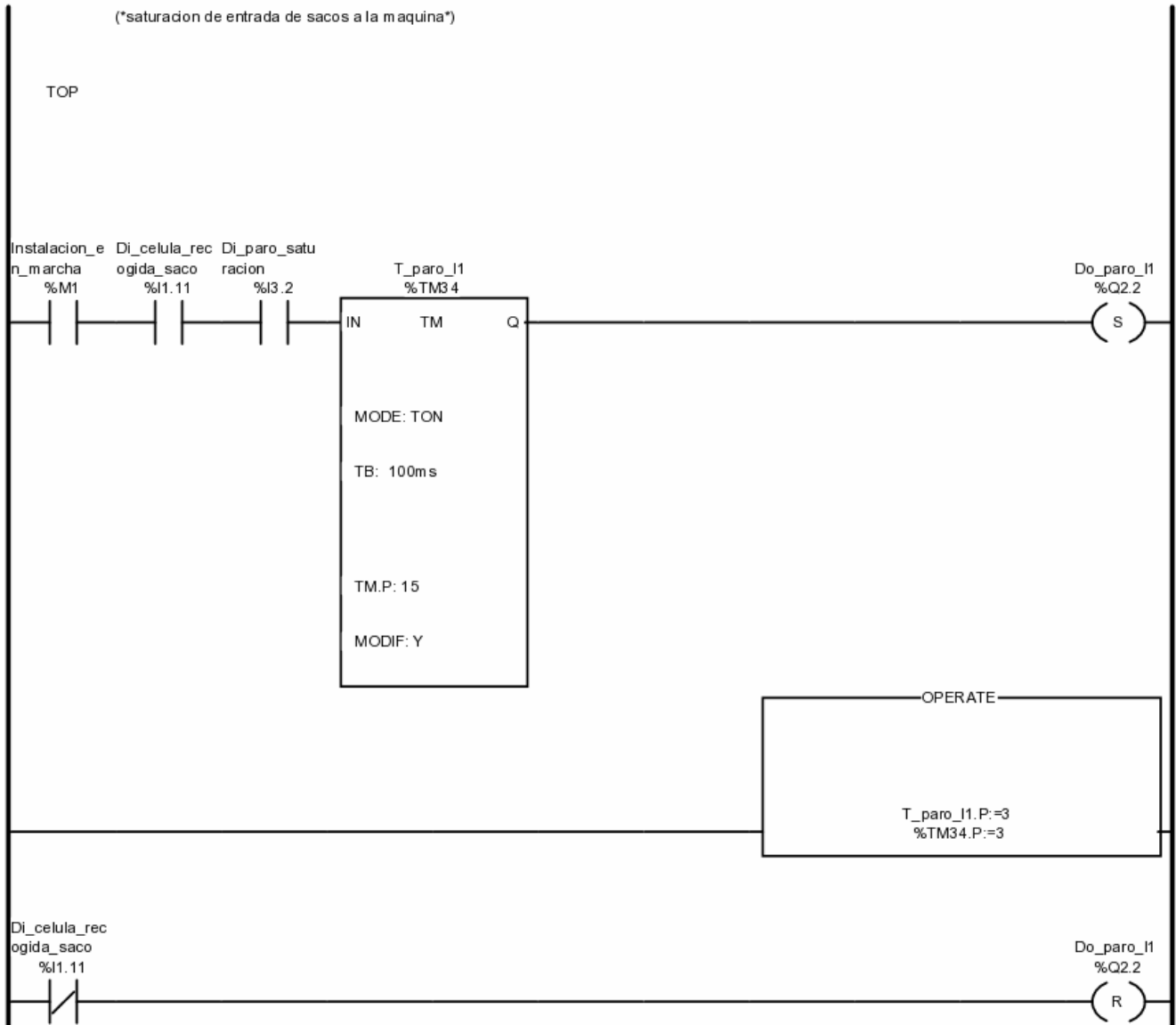








### 11.2.4 LD\_Mast\_Saturación.

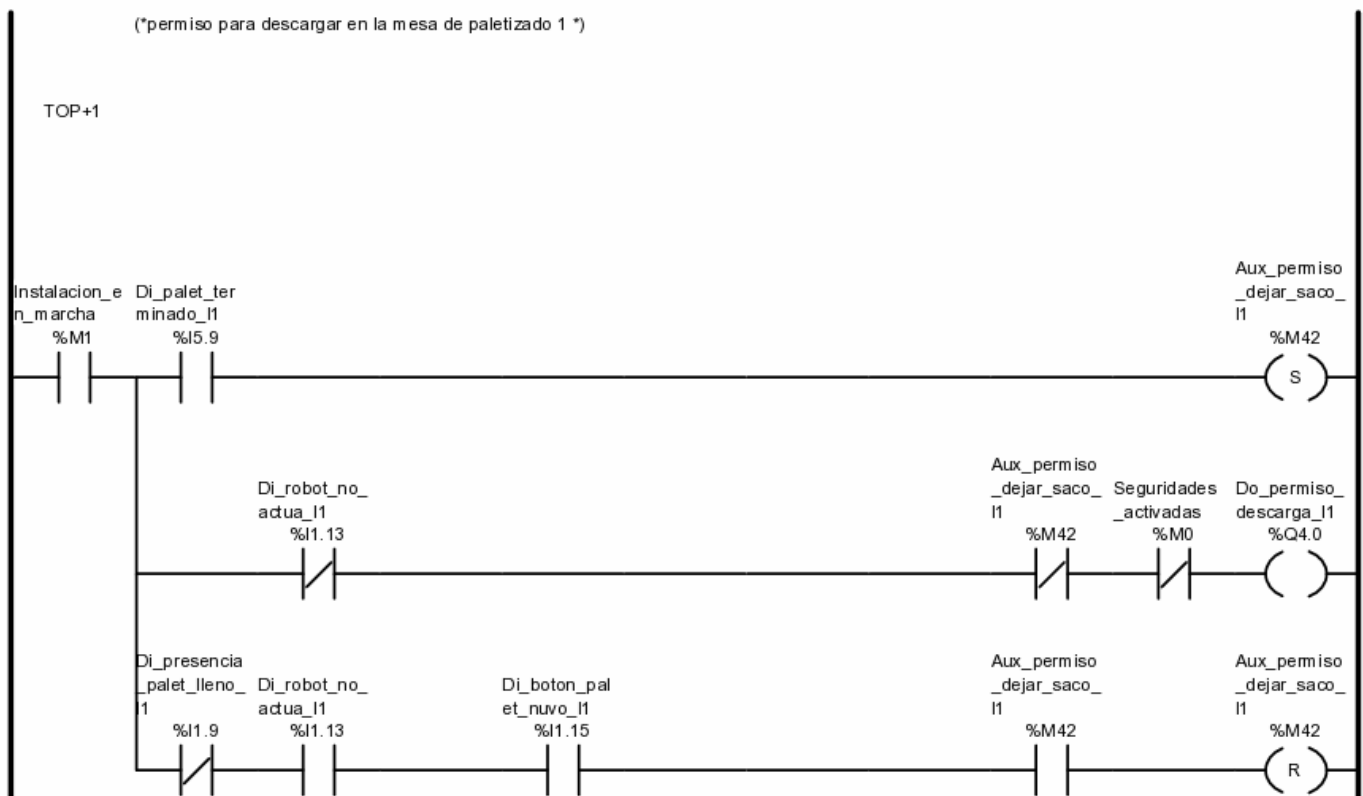
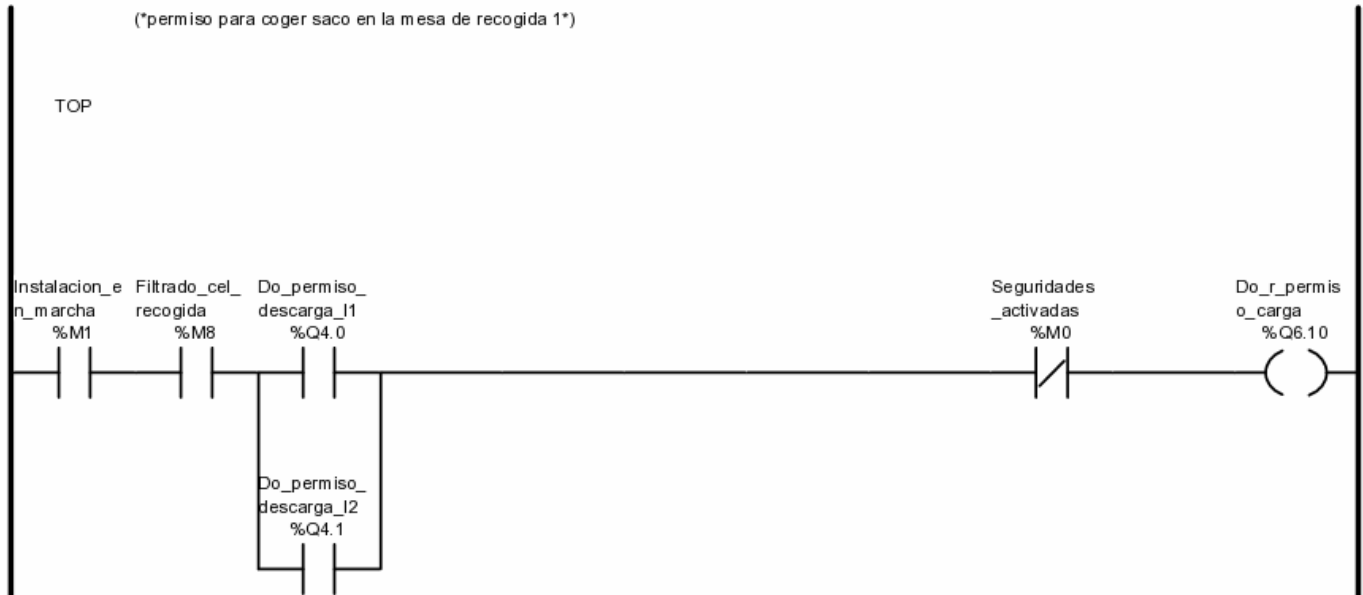


Carpeta: 1

MAST - Saturacion

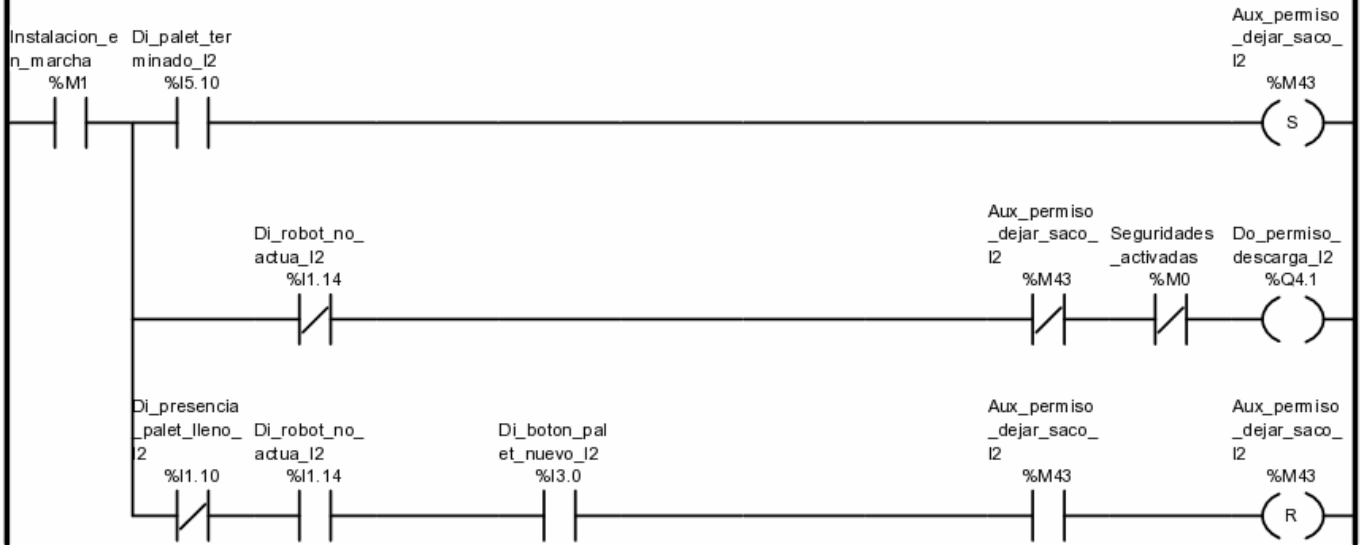
Fecha: 28/8/09

## 11.2.5 LD\_Mast\_Perminos.



(\*permiso para descargar en la mesa de paletizado 2 \*)

TOP+2

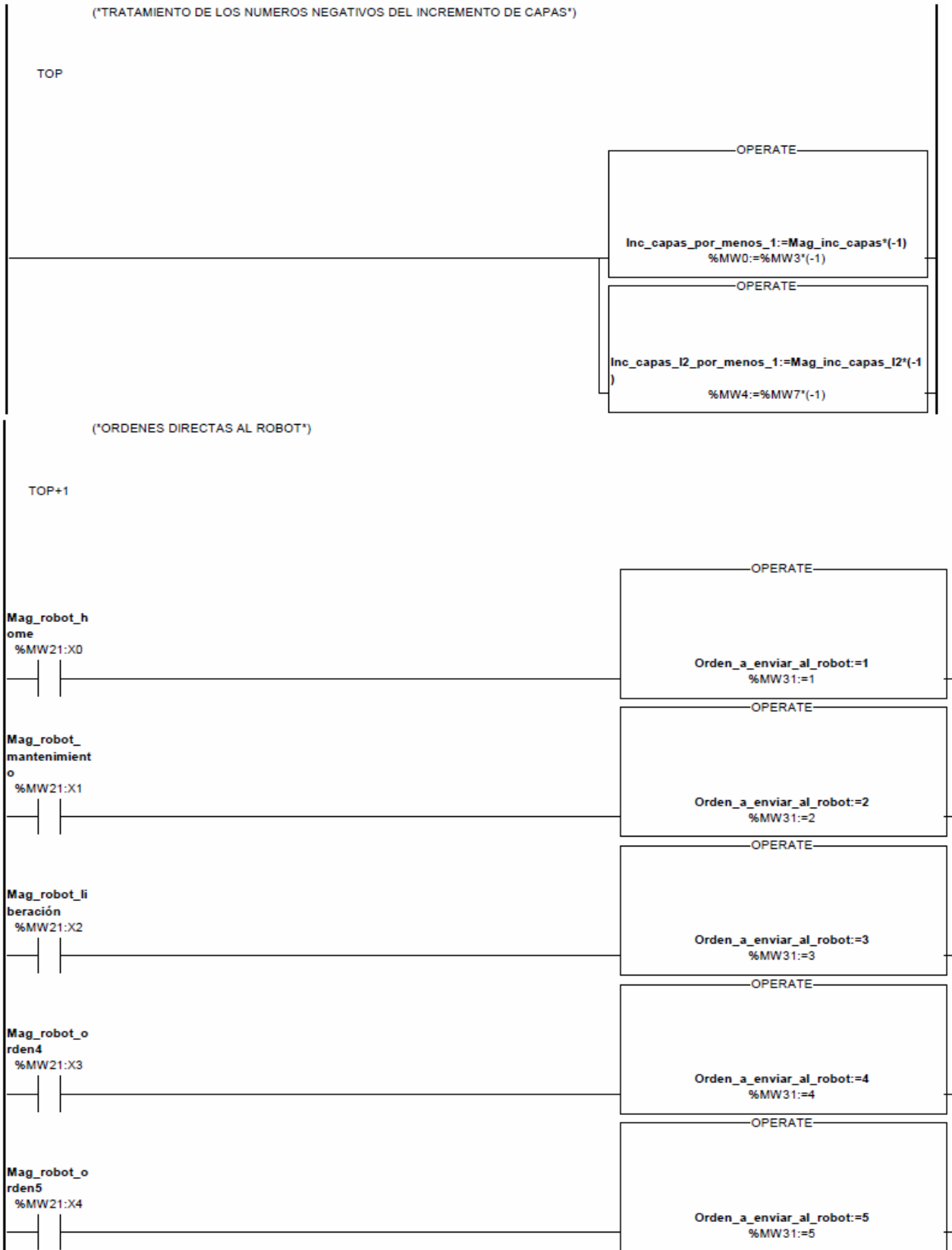


Carpeta: 2

MAST - Permisos

Fecha: 28/8/09

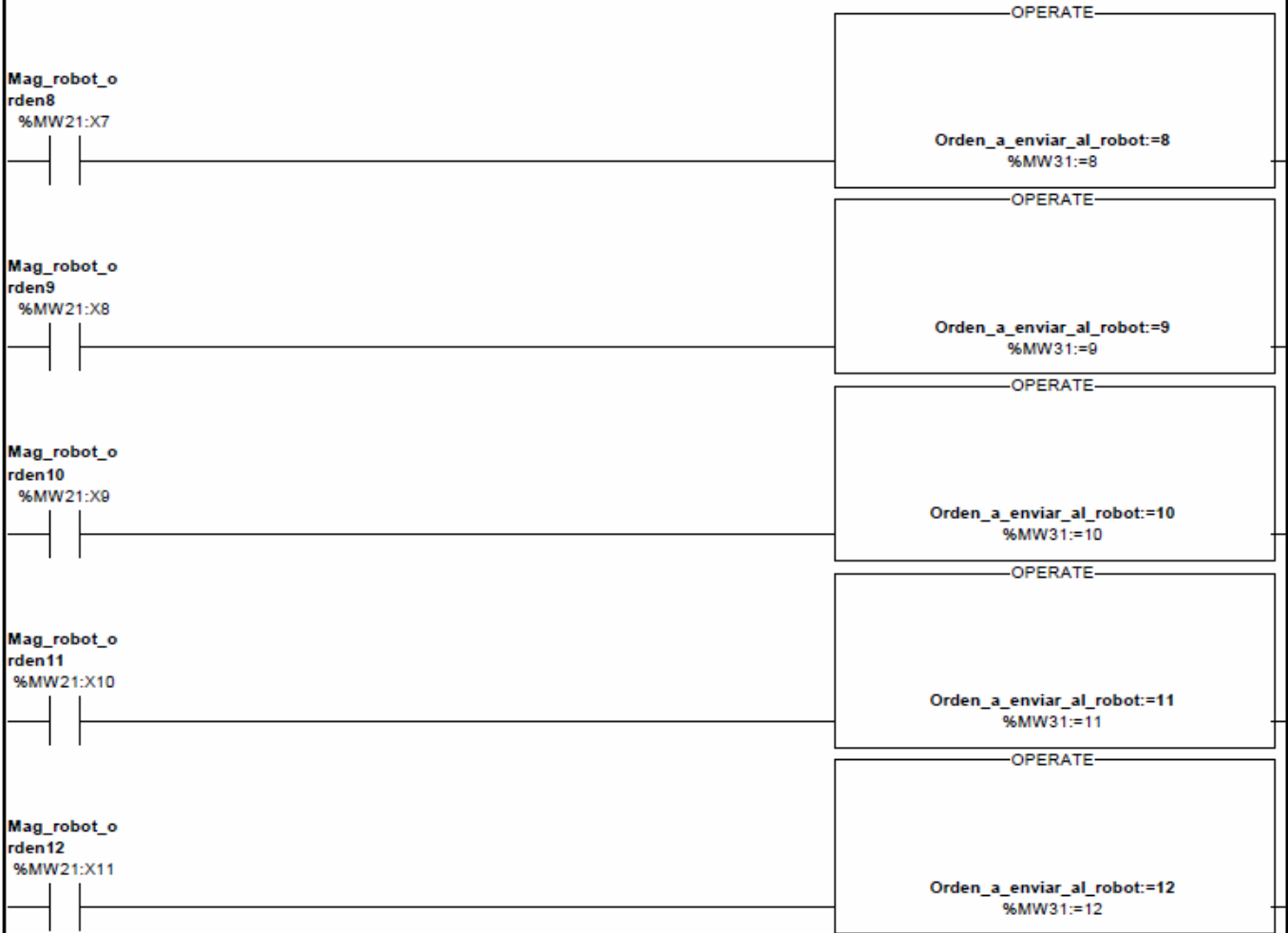
## 11.2.6 LD\_Mast\_Com\_Robot.

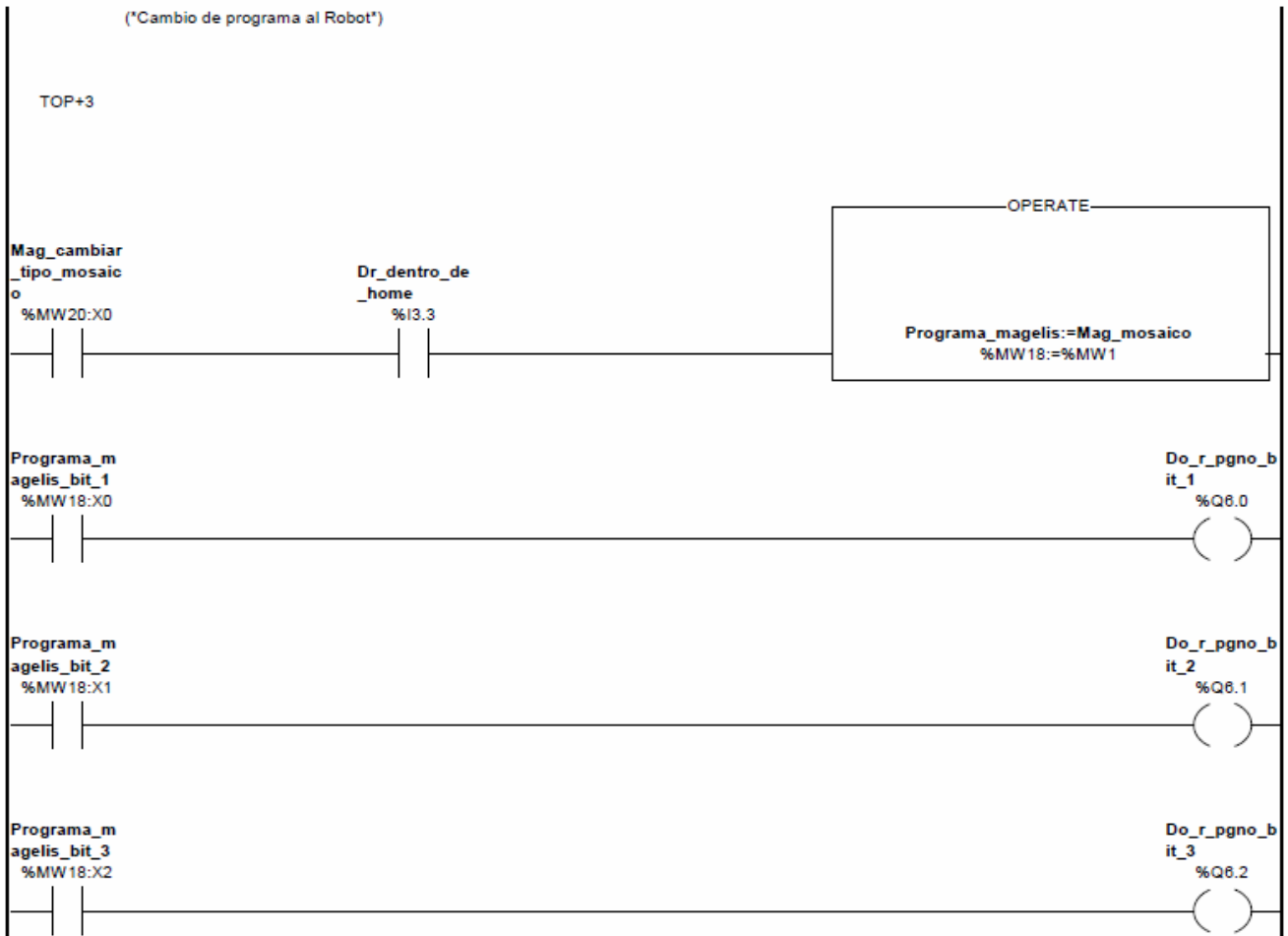
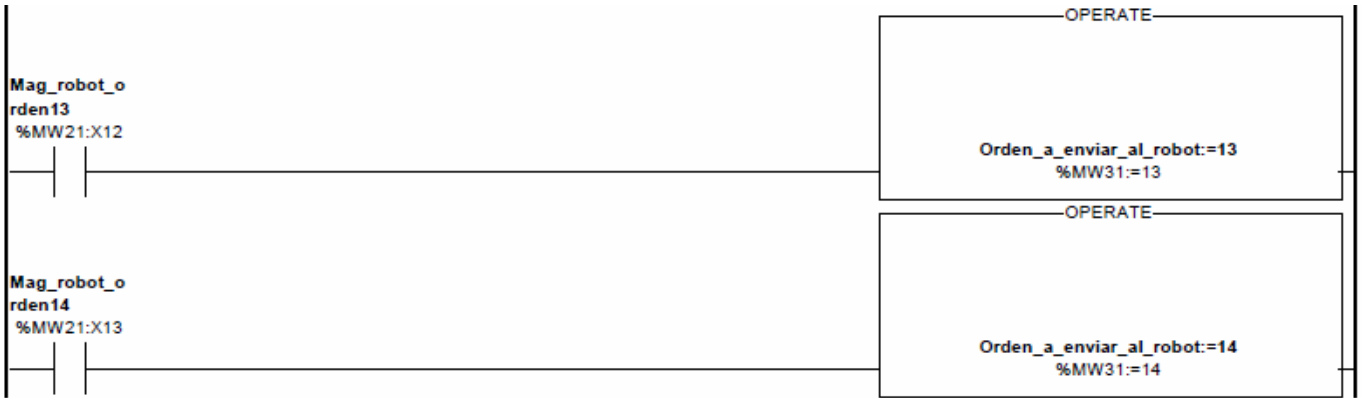




(\*ORDENES DIRECTAS AL ROBOT\*)

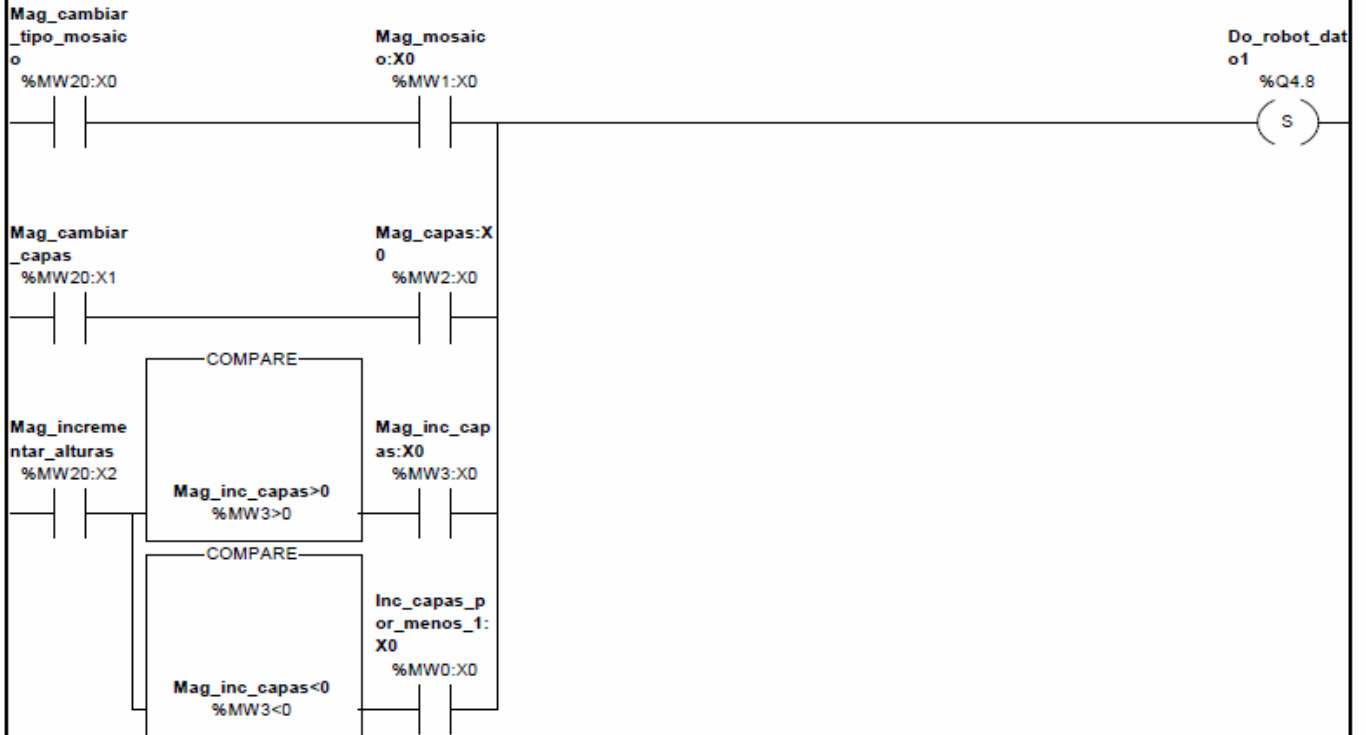
TOP+2





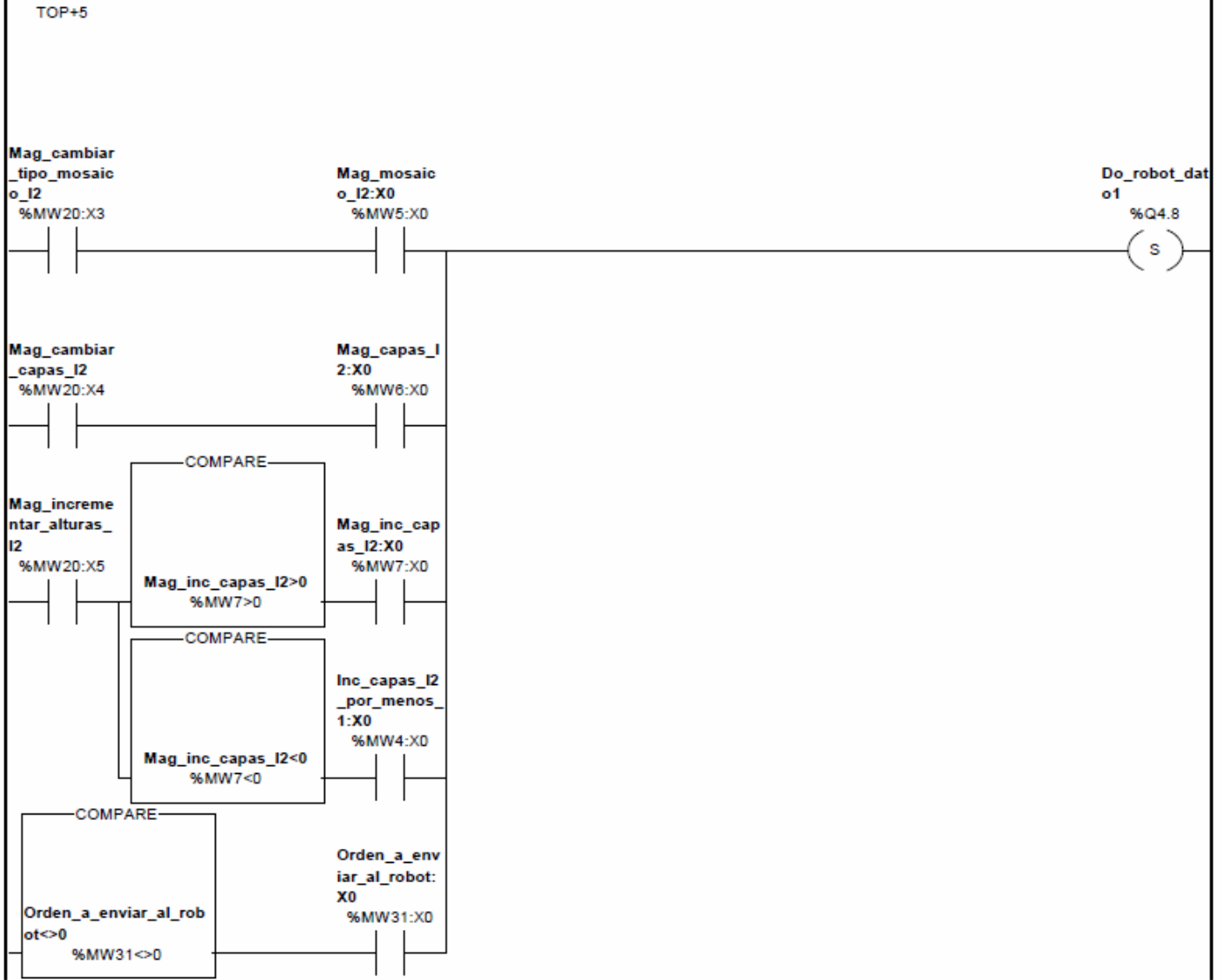
(\*SALIDAS DE DATO 1 DE COMUNICACION CON EL ROBOT\*)

TOP+4

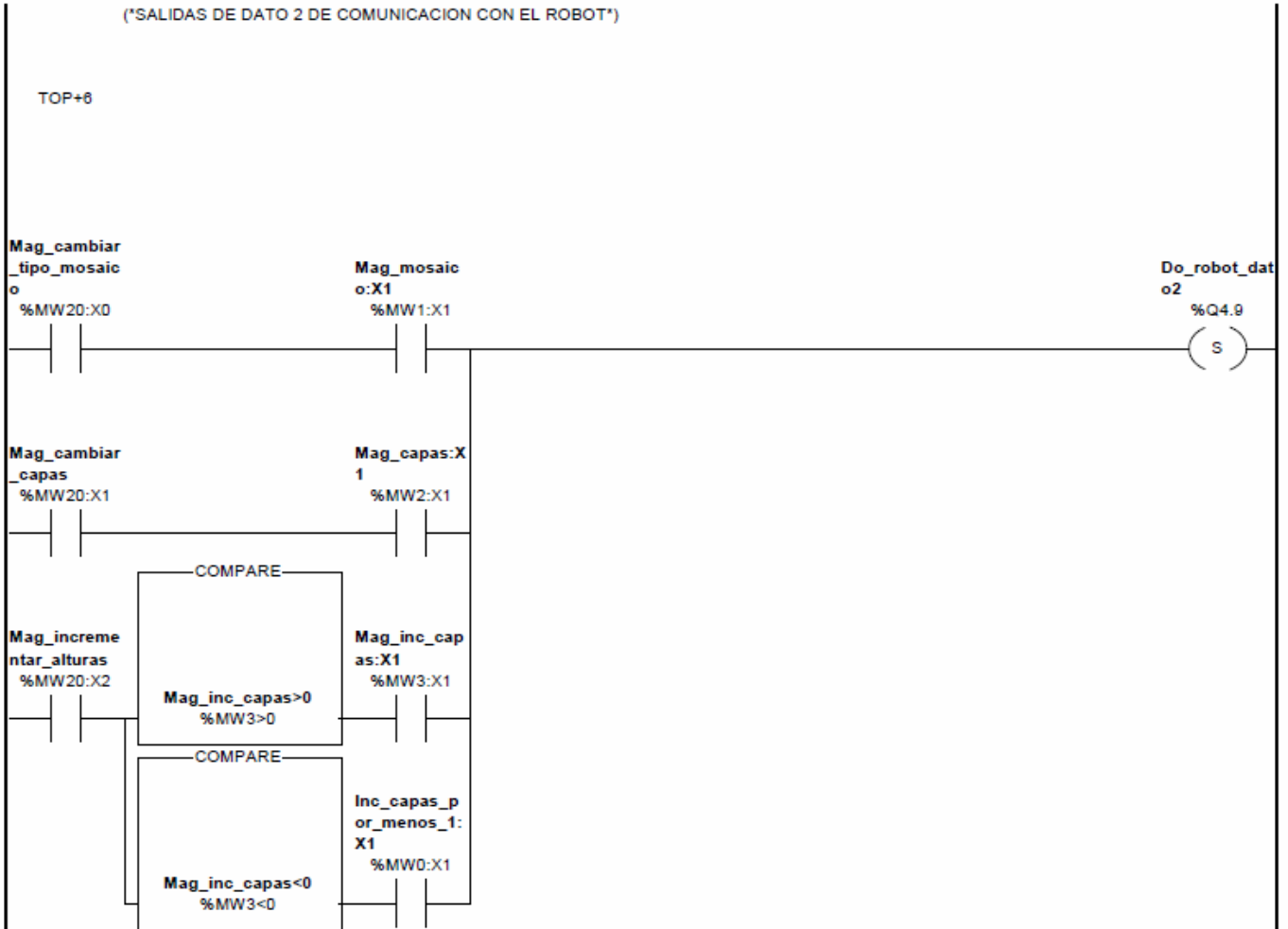




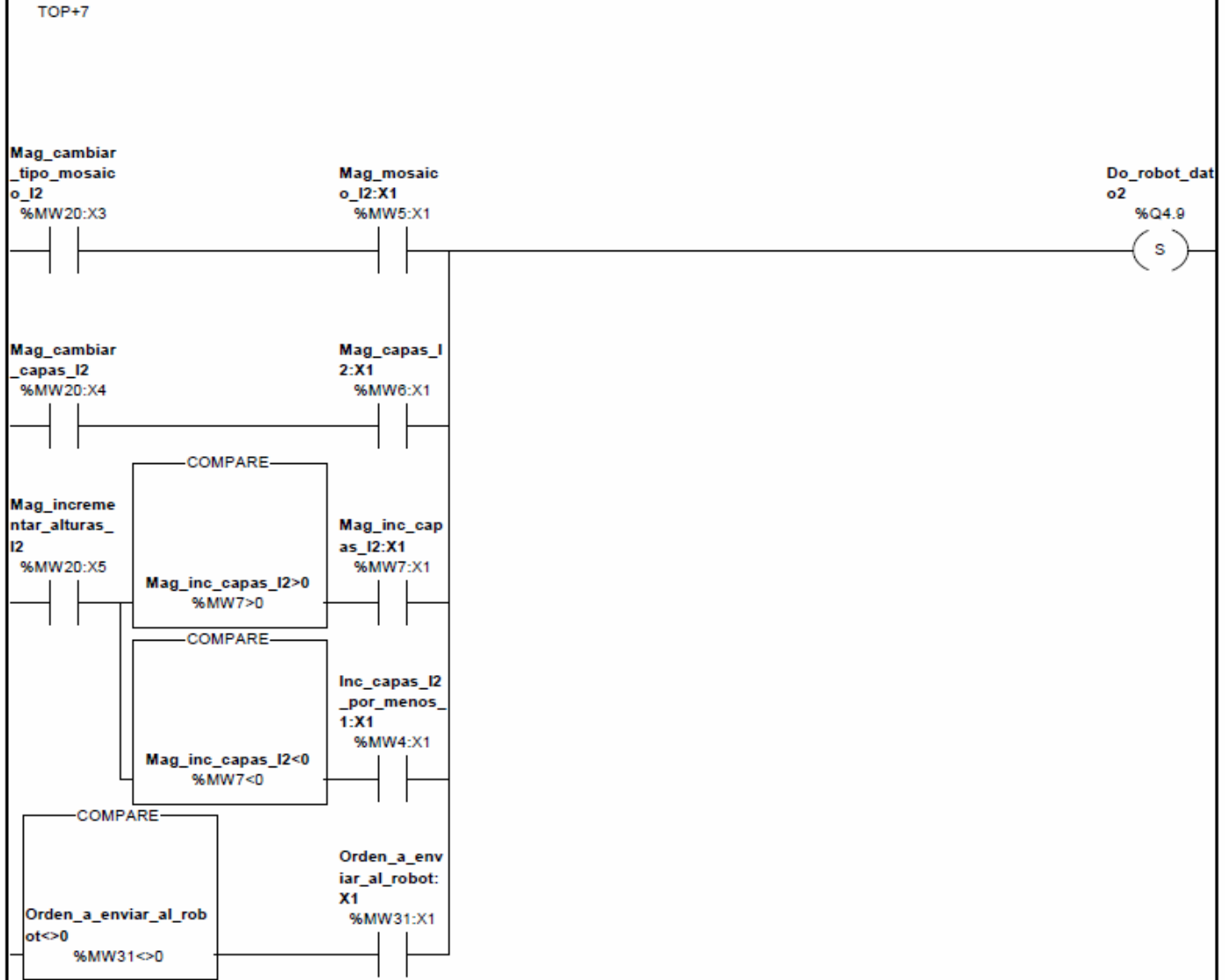
(\*SALIDAS DE DATO 1 DE COMUNICACION CON EL ROBOT\*)



(\*SALIDAS DE DATO 2 DE COMUNICACION CON EL ROBOT\*)

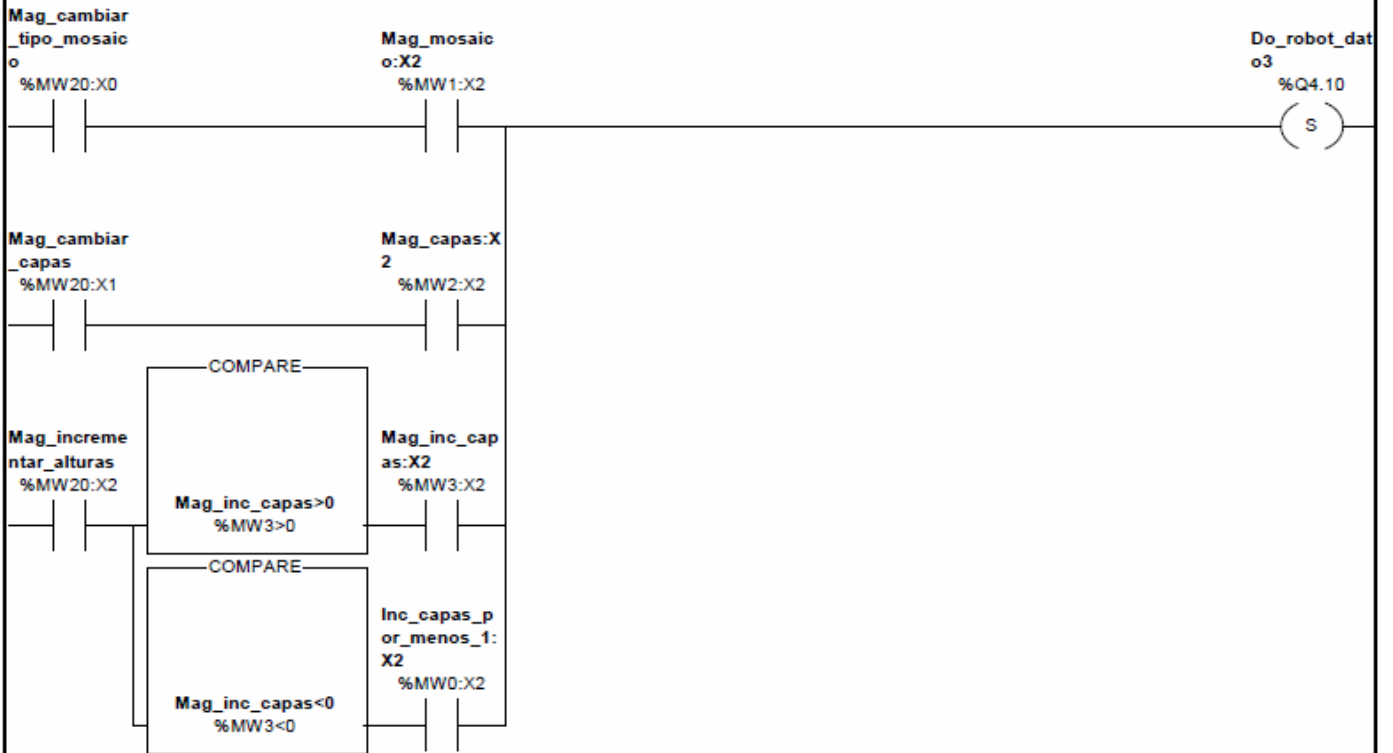


(\*SALIDAS DE DATO 2 DE COMUNICACION CON EL ROBOT\*)

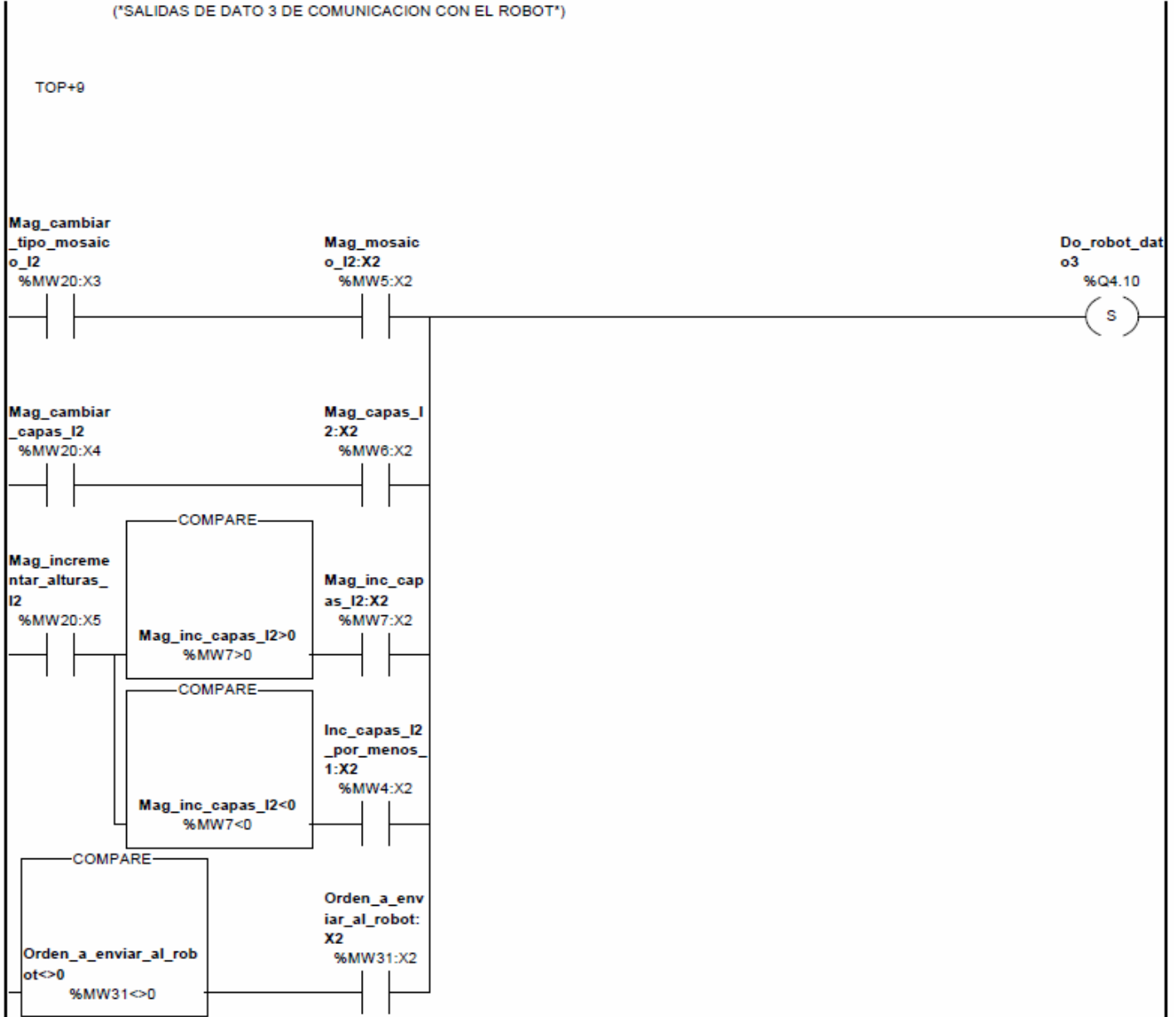


(\*SALIDAS DE DATO 3 DE COMUNICACION CON EL ROBOT\*)

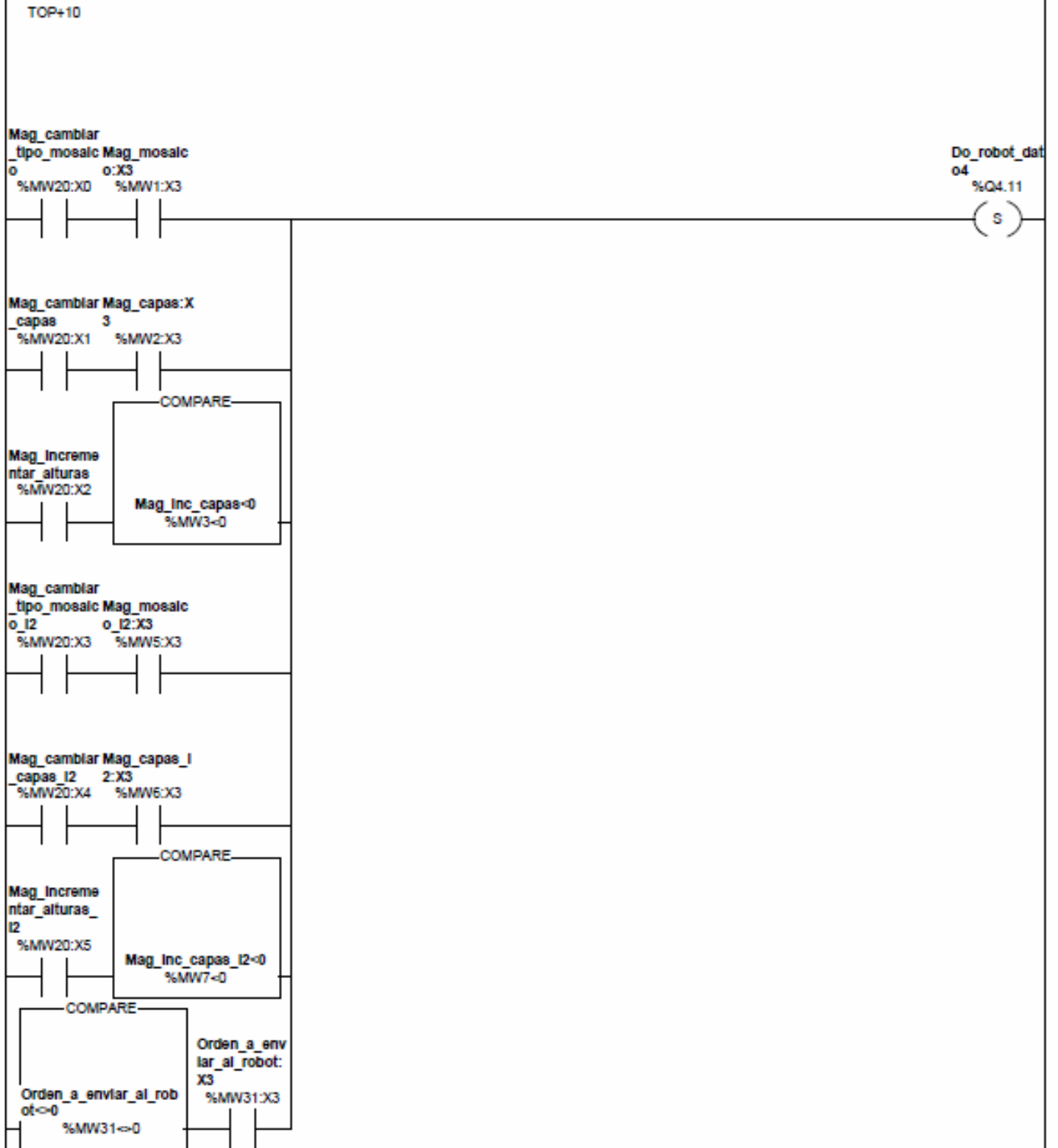
TOP+8



(\*SALIDAS DE DATO 3 DE COMUNICACION CON EL ROBOT\*)

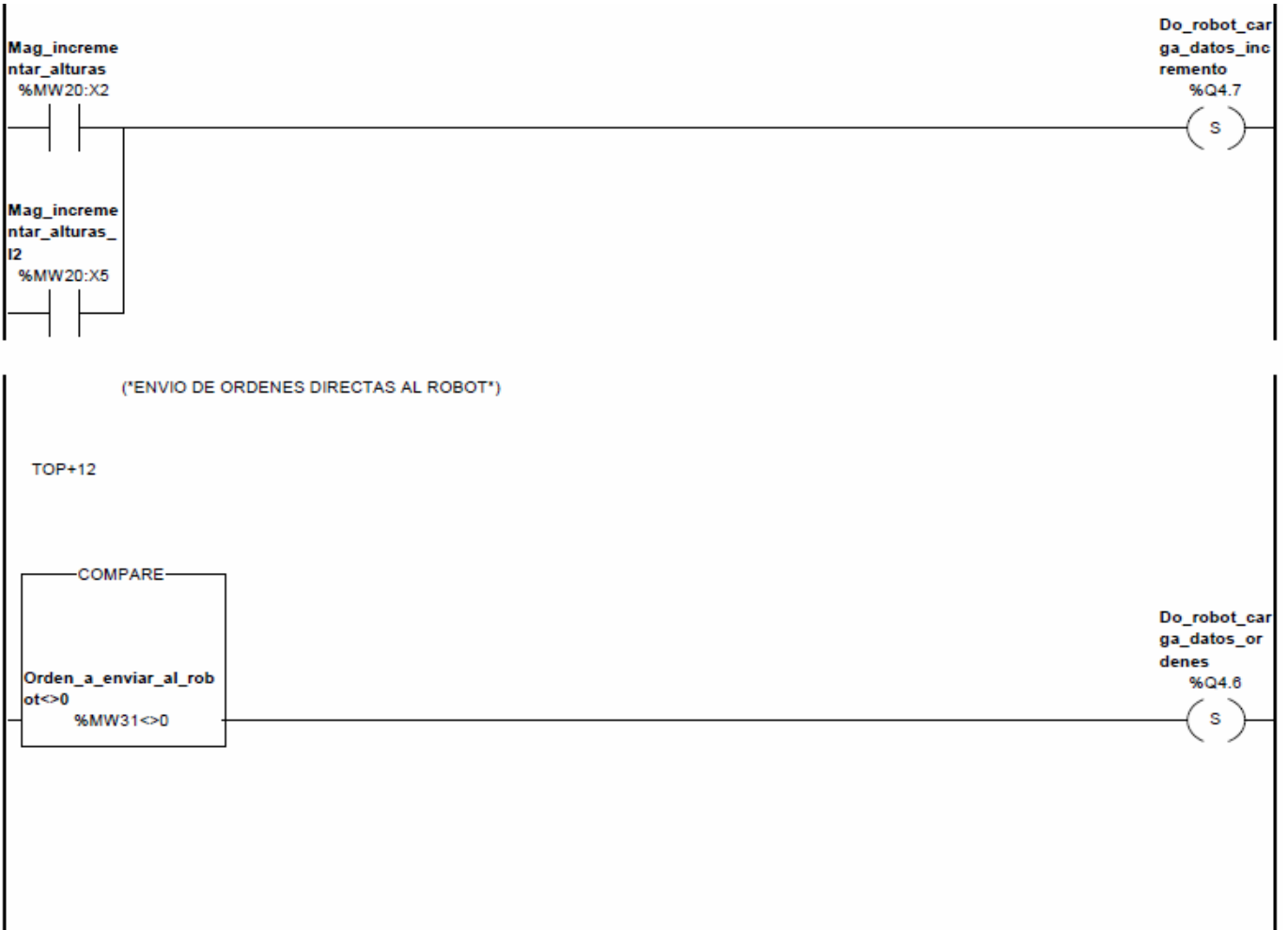


("SALIDAS DE DATO 4 DE COMUNICACION CON EL ROBOT")



(\*CLASIFICACION DE LOS DATOS QUE SE VAN A ENVIAR\*)





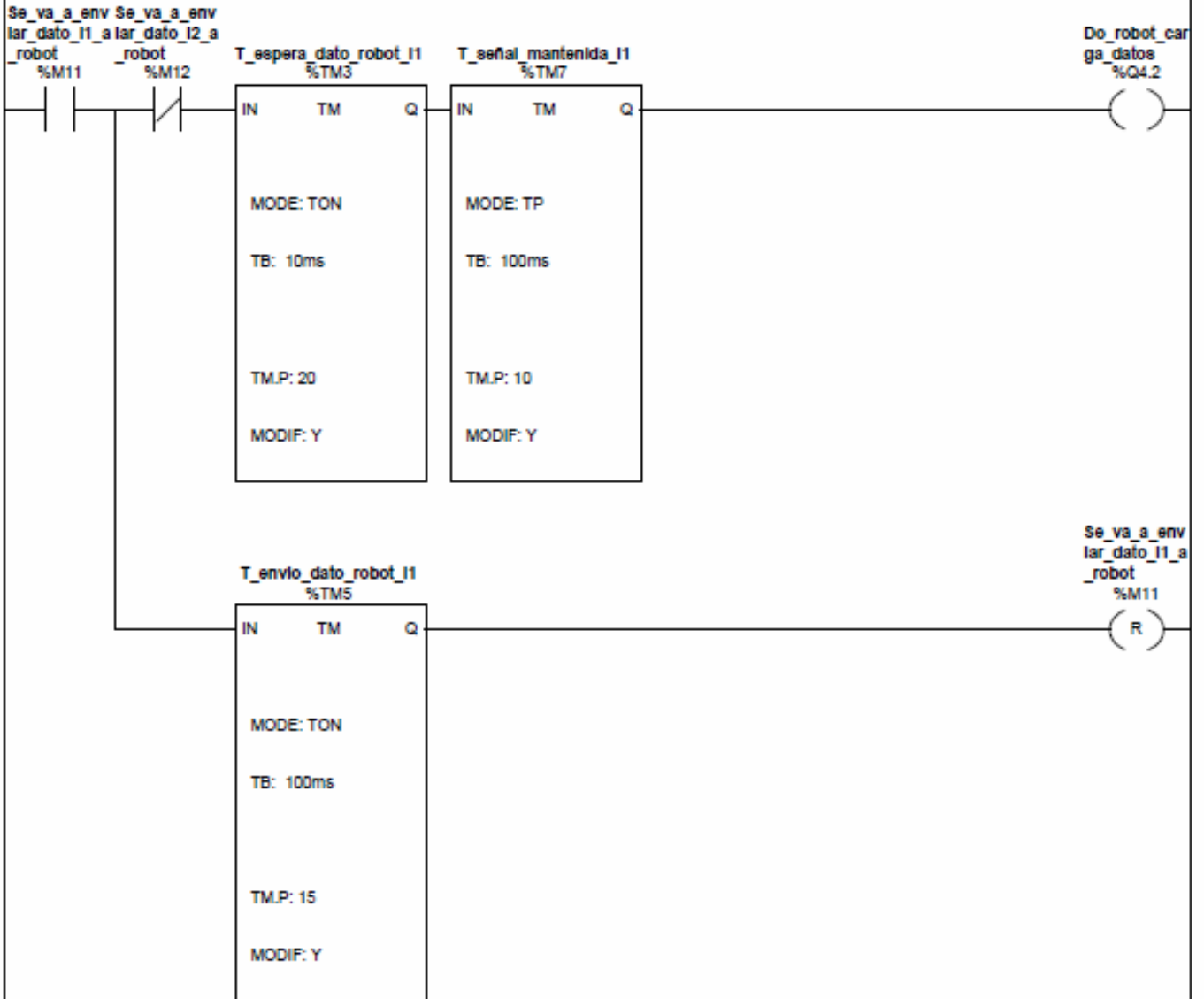


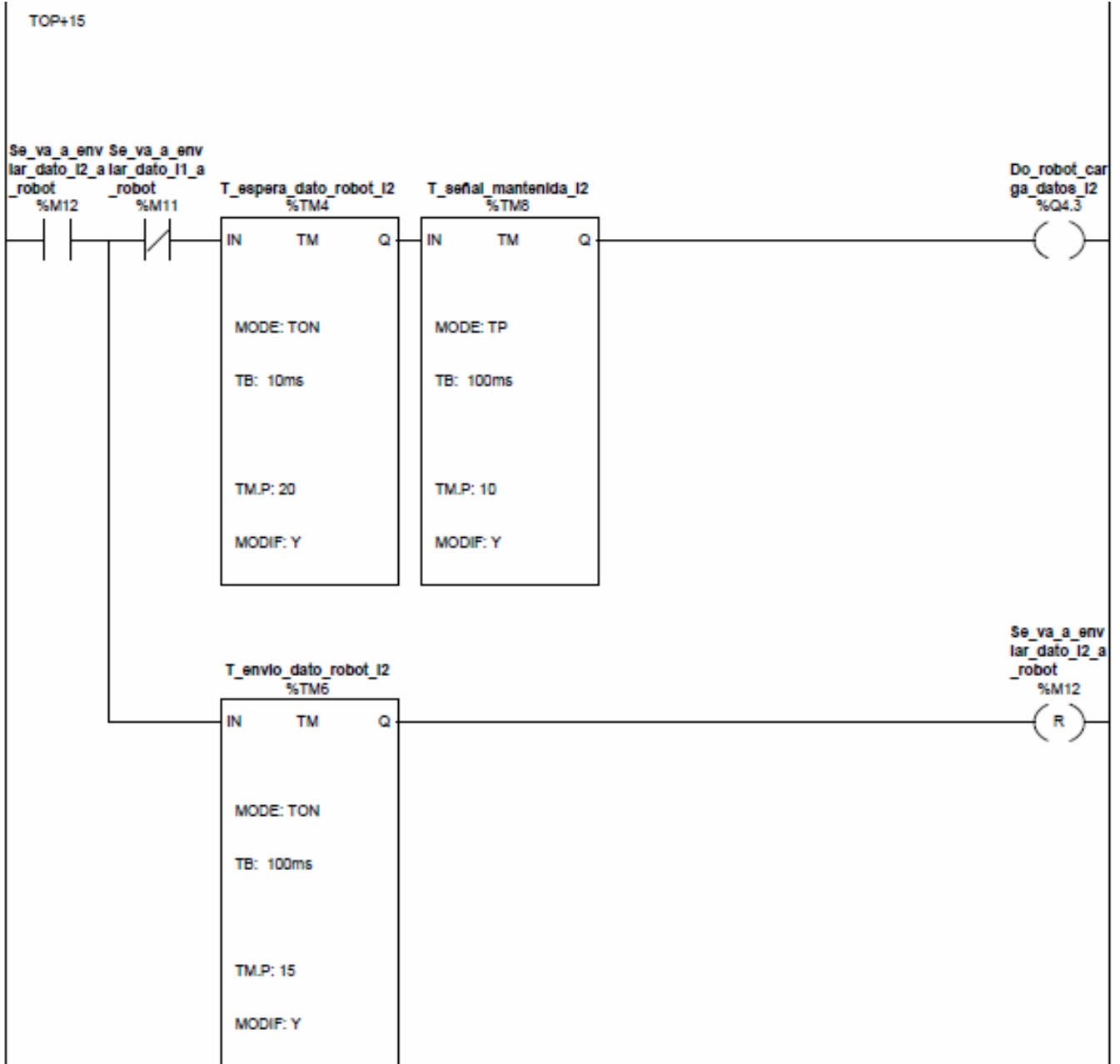
(\*EXISTE ALGUN DATO QUE SE QUIERE ENVIAR AL ROBOT\*)

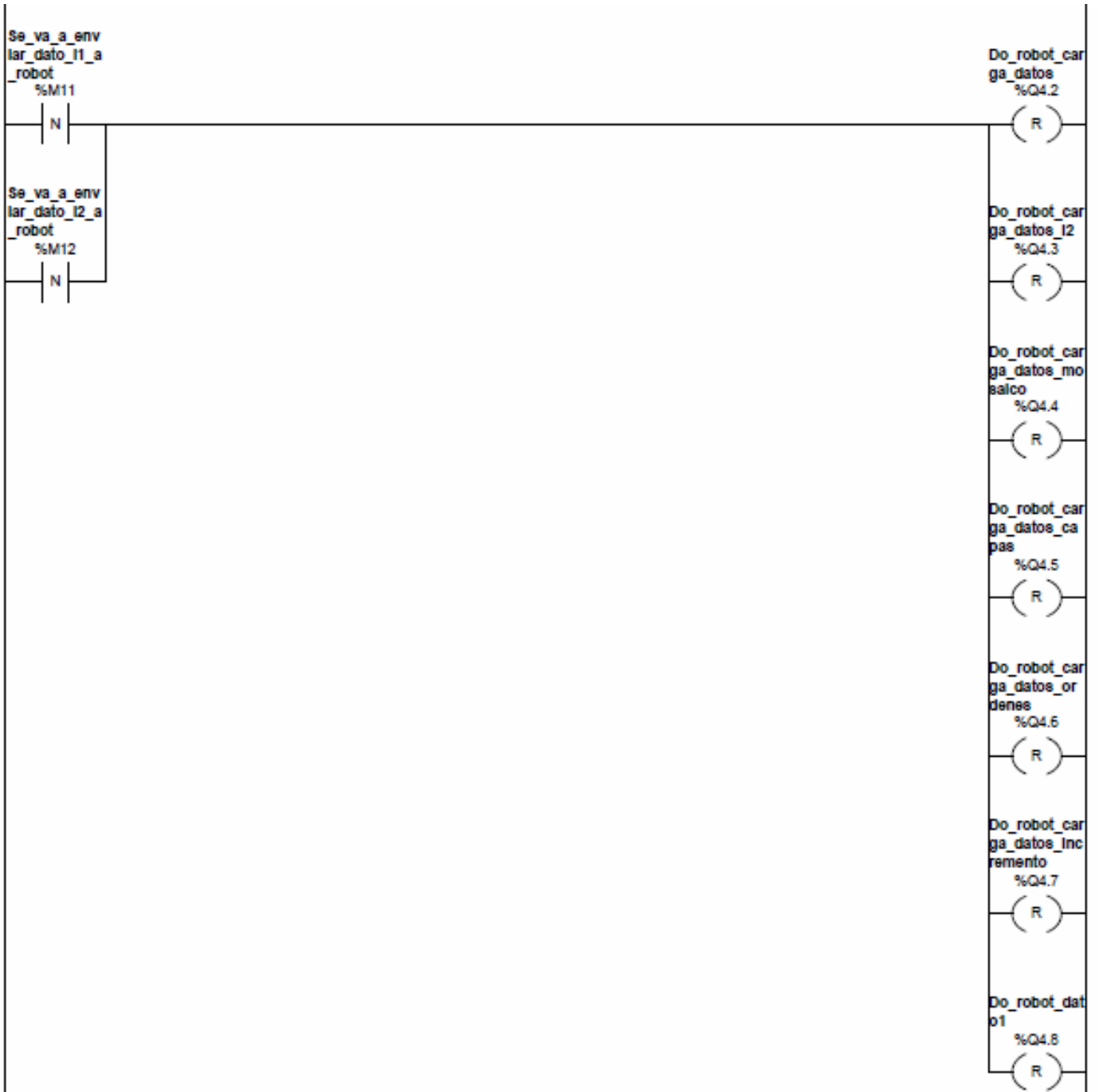


(\*ESPERA A QUE EL ROBOT PROCESA LAS SALIDAS QUE SE ENVIAN\*)

TOP+14







(\*RESETEO DE TODAS LAS SALIDAS DE COMUNICACION\*)

TOP+17

Se\_va\_a\_enviar\_dato\_l1\_a\_robot

%M11

N

Se\_va\_a\_enviar\_dato\_l2\_a\_robot

%M12

N

Do\_robot\_data2

%Q4.9

(R)

Do\_robot\_data3

%Q4.10

(R)

Do\_robot\_data4

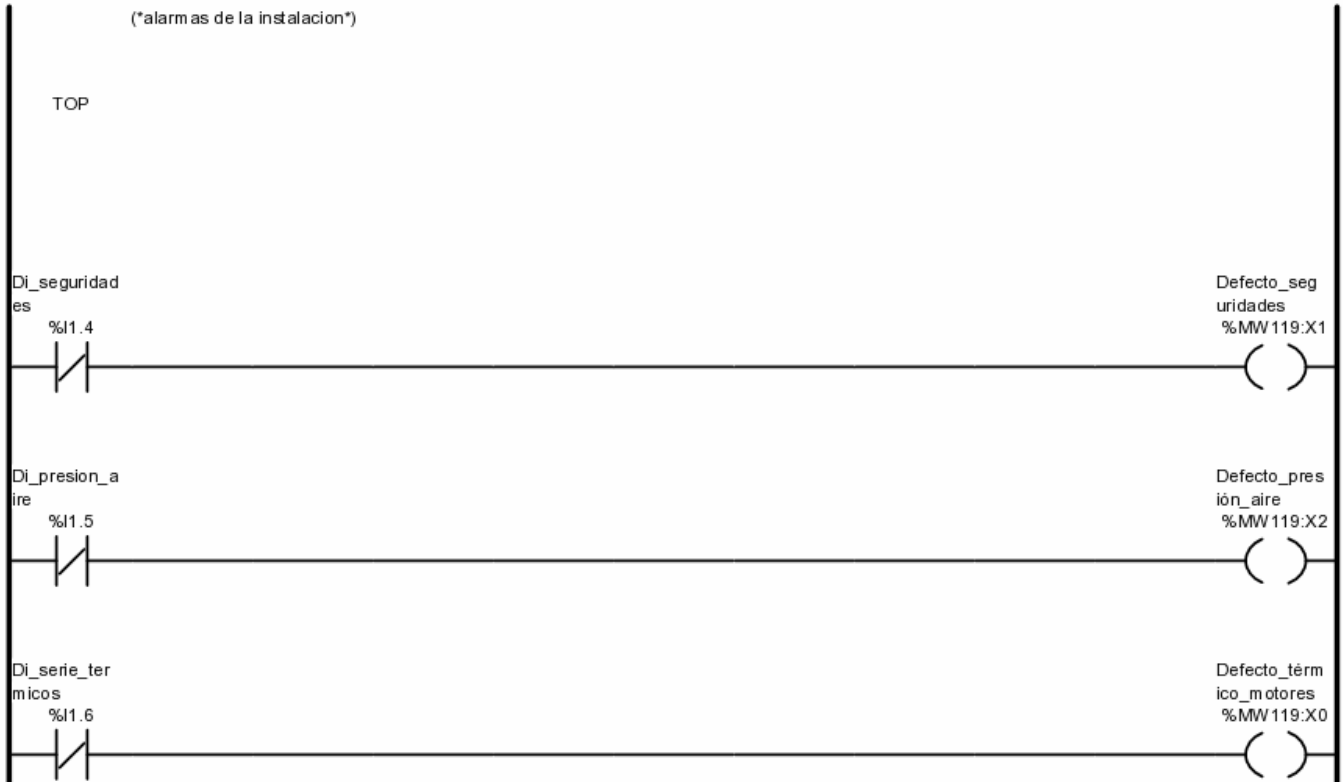
%Q4.11

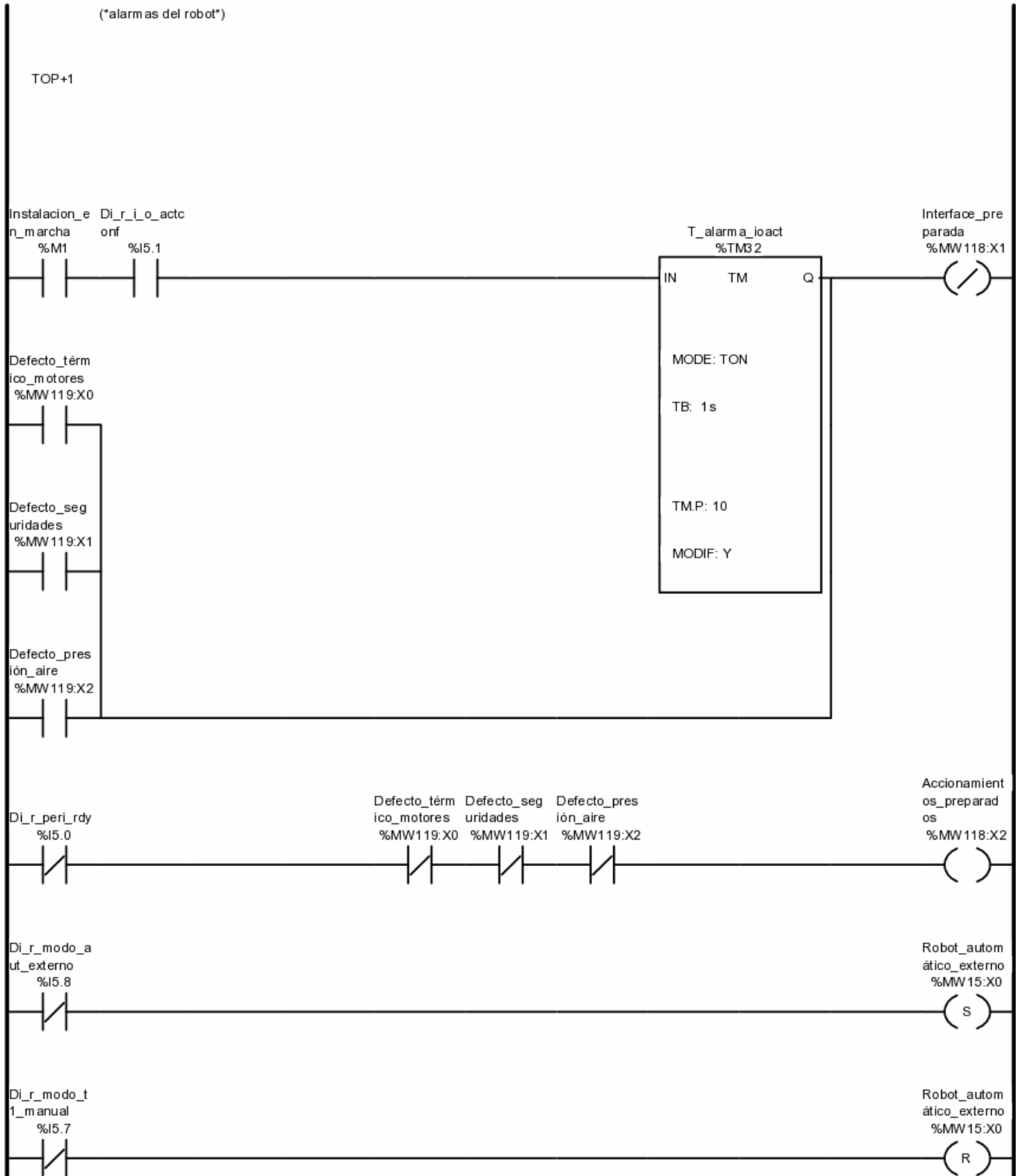
(R)

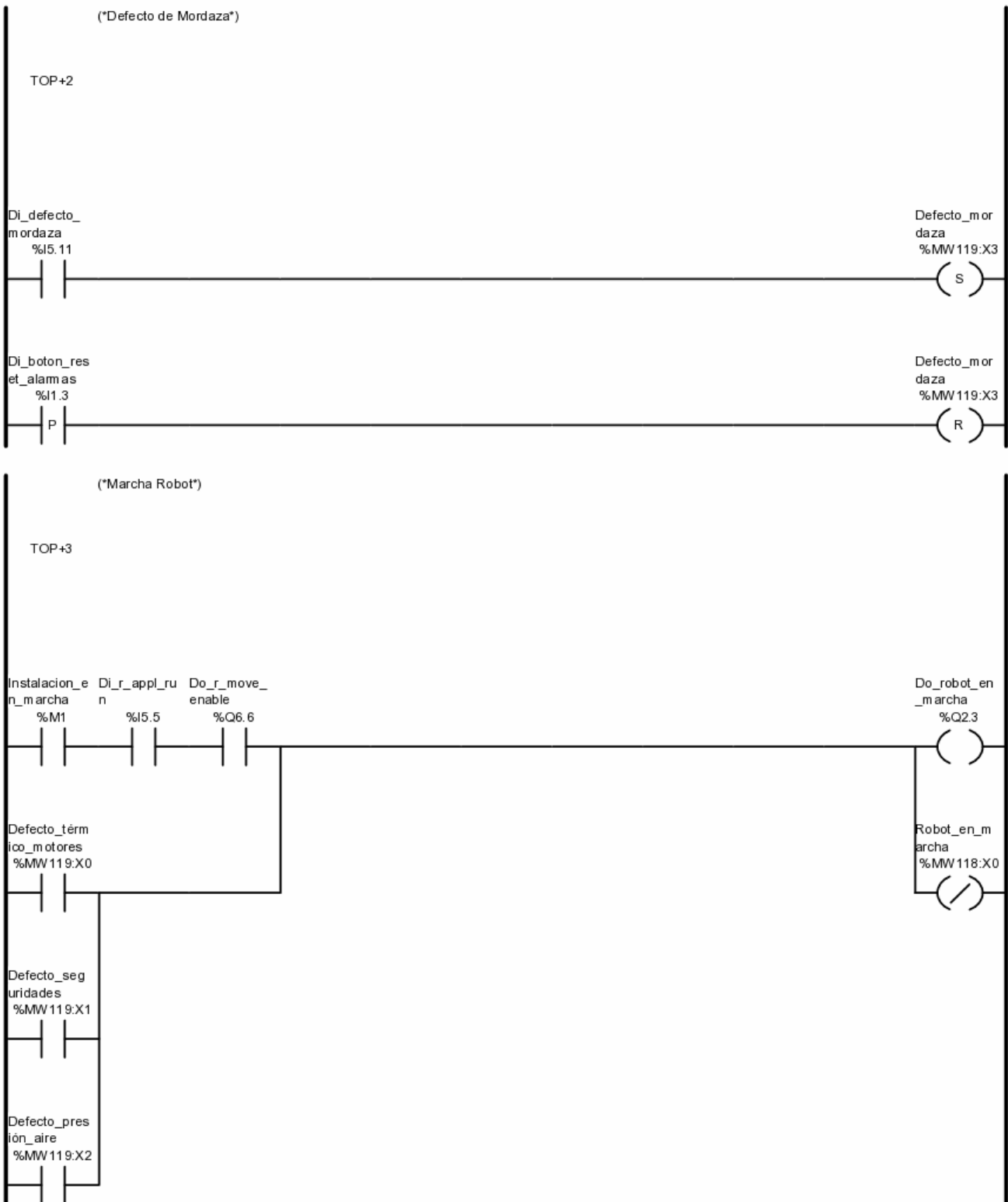
OPERATE

Orden\_a\_enviar\_al\_robot:=0  
%MW31:=0

## 11.2.7 LD\_Mast\_Alarmas.











## 12 ANEXO.E: (Programa de Robot Paletizado).

### 12.1 MOSAICO\_3.src / MOSAICO\_4.src. / MOSAICO\_5.src.

```

DEF mosaico_3 ( )
ext zona_222 ( )
ext zona_111 ( )
ext coi_sgrd ( )
ext mantenimiento ( )
ext home ( )

INI
;Make your modifications here

;=====PUNTOS DE SEGURIDAD=====
coi_sgrd ( )
$out[21]=false

;=====Reset de mordaza=====
pulse ($out[17],true,2)
pulse ($out[19],true,2)
$timer[1]=0
saco_1=1
saco_2=1
base_data[3]=base_data[1]
base_data[4]=base_data[2]
$out[30]=false

;=====PROGRAMA CICLICO DEPENDIENDO DE SELECCION=====
loop

continue
if (($out[6]==false) and ($in[7]==true) and ($out[9]==true)) then
$out[6]=true
endif

;=====RESTOS=====

continue
if $out[25]==true then
wait sec 1
pulse ($out[10],true,2)
pulse ($out[11],true,2)
saco_1=1
saco_2=1
base_data[3]=base_data[1]
base_data[4]=base_data[2]
wait sec 3
$out[25]=false
endif

;=====MANTENIMIENTO=====
continue
if (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and
(registro_1==2)) then
mantenimiento()
endif

;=====ZONA PALETIZADO 1=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[13]==true) and ($out[30]==false)) then
zona_111 ( )

```

```

DEF mosaico_3() = DEF mosaico_4 ( ) = DEF mosaico_5 ( )
zona_222() = zona_22 ( ) = zona_2 ( )
zona_111() = zona_11 ( ) = zona_1 ( )

```

Mosaico\_4() y Mosaico\_5() sería igual pero cambiando las variables.

```

endif
endif

;=====ZONA PALETIZADO 2=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[14]==true) and ($out[30]==true)) then
zona_222 ()
endif
endif

;=====HOME=====
=
continue
if (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and
(registro_1==1) and (saco_1<2) and (saco_2<2)) then
home ()
endif

;=====SALIDA CONDICIONAL=====
continue
if (((program==1) or (program==2)) and ($in[11]==false) and (saco_1<2) and (saco_2<2))
then
wait sec 1
exit
endif

endloop
wait sec 0
coi_sgrd ()

END

```

MOSAICO\_5 () -> if (((program==2) or (program==3))  
MOSAICO\_4 () -> if (((program==1) or (program==3))

## 12.2 MANTENIMIENTO.src

```

DEF mantenimiento( )
INI
;Make your modifications here

coi_sgrd ()

PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P2 CONT Vel= 30 % PDAT2 Tool[1] Base[0]
PTP P3 CONT Vel= 30 % PDAT3 Tool[1] Base[0]
;punto de mantenimiento
PTP P4 CONT Vel= 30 % PDAT4 Tool[1] Base[0]

;=====salida de mantenimiento=====

wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))

PTP P5 CONT Vel= 30 % PDAT5 Tool[1] Base[0]

coi_sgrd ()

END

```

### 12.3 HOME.src

```
DEF home( )  
  
coi_sgrd ( )  
  
;marca para saber que estamos dentro del Home y cambiar de programa  
$out[21]=true  
  
wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)  
and (registro_1==3))  
  
;=====CAMBIO DE MOSAICOS=====  
;Mosaico seleccionado  
$out[21]=false  
  
;=====CAMBIO DE CAPAS=====  
capas_1=reg_capas_1  
if program==1 then  
sacos_1=capas_1*5  
sacos_2=capas_1*5  
endif  
if program==2 then  
sacos_1=capas_1*4  
sacos_2=capas_1*4  
endif  
if program==3 then  
sacos_1=capas_1*3  
sacos_2=capas_1*3  
endif  
  
END
```

### 12.4 COI\_SGRD.src

```
DEF coi_sgrd( )  
INI  
;Make your modifications here  
  
;=====ENCIMA MESA DE PALETIZADO LINEA 2=====if ($axis_act.a1>60) then  
PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[0  
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[0  
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[0  
endif  
;=====TRANSITO PALET_2 ZONA RECOGIDA=====if (($axis_act.a1<=60) and ($axis_act.a1>-35)) then  
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[0  
endif  
;=====TRANSITO LINEA_1 y ZONA RECOGIDA=====if (($axis_act.a1<-50) and ($axis_act.a1>=-80)) then  
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[0  
endif  
;=====ENCIMA MESA PALETIZADO LINEA 1 =====if ($axis_act.a1<=-80) then  
PTP P8 CONT Vel= 100 % PDAT8 Tool[1] Base[0  
endif  
;=====ENCIMA MESA RECOGIDA=====if (($axis_act.a1>=-50) and ($axis_act.a1<=-35)) then  
PTP HOME CONT Vel= 100 % PDAT17
```

```

endif

PTP HOME CONT Vel= 100 % PDAT19

;punto por encima de Home
PTP P4 CONT Vel= 100 % PDAT4 Tool[1] Base[0]
;punto por debajo de Home
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]

PTP HOME CONT Vel= 100 % PDAT18

END

```

## 12.5 ZONA\_1.src / ZONA\_2.src

DEF zona\_1( )

```

alt_incr_1=190
$out[30]=true
coger_1 ( )

;;;saco 1 capa 1;;;;;;

```

```

if (saco_1==1) or (saco_1==11) or (saco_1==21) or (saco_1==31) or (saco_1==41) or
(saco_1==51) or (saco_1==61) or (saco_1==71) then
PTP P1 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

```

;;;;;;;;;saco 1 capa 2;;;;;;;;;

```

```

if (saco_1==6) or (saco_1==16) or (saco_1==26) or (saco_1==36) or (saco_1==46) or
(saco_1==56) or (saco_1==66) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

```

;;;;;;;;;saco 2 capa 1;;;;;;;;;

```

```

if (saco_1==2) or (saco_1==12) or (saco_1==22) or (saco_1==32) or (saco_1==42) or
(saco_1==52) or (saco_1==62) or (saco_1==72) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

```

;;;;;;;;;saco 2 capa 2;;;;;;;;;

```

```

if (saco_1==7) or (saco_1==17) or (saco_1==27) or (saco_1==37) or (saco_1==47) or
(saco_1==57) or (saco_1==67) then
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

```

;;;;;;;;;saco 3 capa 1;;;;;;;;;

```

```

Zona_1.src      = Zona_2.src
DEF zona_1()   = DEF zona_2 ().
Coger_1 ( )    = Coger_2 ().
Saco_1         = saco_2.
Base_data[3]   = base_data[1].
Base_data[4]   = base_data[2].
Sacos_1        = Sacos_2.
pulse_out[10]  = pulse_out[11].
PTP->Tool[1] Base[3] = PTP->Tol[1] Base[4]

```

Zona\_2.src sería igual que Zona\_1.src pero cambiando las variables.



```
if (saco_1==3) or (saco_1==13) or (saco_1==23) or (saco_1==33) or (saco_1==43) or
(saco_1==53) or (saco_1==63) or (saco_1==73) then
PTP P3 CONT Vel= 100 % PDAT4 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa2;;;;;;;;;;;;

if (saco_1==8) or (saco_1==18) or (saco_1==28) or (saco_1==38) or (saco_1==48) or
(saco_1==58) or (saco_1==68) then
PTP P8 CONT Vel= 100 % PDAT8 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 4 capa 1;;;;;;;;;;;;

if (saco_1==4) or (saco_1==14) or (saco_1==24) or (saco_1==34) or (saco_1==44) or
(saco_1==54) or (saco_1==64) or (saco_1==74) then
PTP P4 CONT Vel= 100 % PDAT5 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 4 capa 2;;;;;;;;;;;;

if (saco_1==9) or (saco_1==19) or (saco_1==29) or (saco_1==39) or (saco_1==49) or
(saco_1==59) or (saco_1==69) then
PTP P9 CONT Vel= 100 % PDAT9 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 5 capa 1;;;;;;;;;;;;

if (saco_1==5) or (saco_1==15) or (saco_1==25) or (saco_1==35) or (saco_1==45) or
(saco_1==55) or (saco_1==65) or (saco_1==75) then
PTP P5 CONT Vel= 100 % PDAT10 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 5 capa 2;;;;;;;;;;;;

if (saco_1==10) or (saco_1==20) or (saco_1==30) or (saco_1==40) or (saco_1==50) or
(saco_1==60) or (saco_1==70) then
PTP P10 CONT Vel= 100 % PDAT11 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;permiso para sacar palet;;;;;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then
pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[1]
wait sec 0.5
endif
```

;;;;;;;;;;incremento saco;;;;;;;;;;

saco\_1=saco\_1+1

ptp\_rel{z 110}c\_ptp

;;;;;;;;;;punto elevado encima de zona 2

SYN PULSE 17 'Abrir\_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]

PTP P24 CONT Vel= 100 % PDAT24 Tool[1] Base[0]

PTP P25 CONT Vel= 100 % PDAT25 Tool[1] Base[0]

;verificacion manos y dedos abiertos

continue

loop

continue

if ((\$in[27]==false) or (\$in[28]==false)) then

\$timer\_stop[1]=false

endif

continue

if \$timer[1]>=4000 then

wait sec 0

\$timer\_stop[1]=true

\$timer[1]=0

pulse (\$out[12],true,2)

endif

continue

if ((\$in[27]==true) and (\$in[28]==true)) then

\$timer\_stop[1]=true

\$timer[1]=0

exit

endif

endloop

END

## 12.6 ZONA\_11.src / ZONA\_22-src.

DEF zona\_11( )

alt\_incr\_1=200

\$out[30]=true

coger\_1( )

;;;;;;;;saco 1 capa 1;;;;;;;;;

if (saco\_1==1) or (saco\_1==9) or (saco\_1==17) or (saco\_1==25) or (saco\_1==33) or (saco\_1==41) or (saco\_1==49) or (saco\_1==57) then

PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[3]

ptp\_rel {z -140} c\_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 1 capa 2;;;;;;;;;

if (saco\_1==5) or (saco\_1==13) or (saco\_1==21) or (saco\_1==29) or (saco\_1==37) or (saco\_1==45) or (saco\_1==53) then

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[3]

ptp\_rel {z -120} c\_ptp

Zona\_11.src = Zona\_22.src  
DEF zona\_11() = DEF zona\_22 ().  
Coger\_1 () = Coger\_2 ().  
Saco\_1 = saco\_2.  
Base\_data[3] = base\_data[1].  
Base\_data[4] = base\_data[2].  
Sacos\_1 = Sacos\_2.  
pulse\_out[10] = pulse\_out[11].

Zona\_22.src sería igual que Zona\_11.src pero cambiando las variables.

```
pulse ($out[19],true,2)
wait sec 0.3

endif

;;;;;;;;;;;;saco 2 capa 1;;;;;;;;;;;;

if (saco_1==3) or (saco_1==11) or (saco_1==19) or (saco_1==27) or (saco_1==35) or
(saco_1==43) or (saco_1==51) or (saco_1==59) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 2 capa 2;;;;;;;;;;;;

if (saco_1==7) or (saco_1==15) or (saco_1==23) or (saco_1==31) or (saco_1==39) or
(saco_1==47) or (saco_1==55) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa 1;;;;;;;;;;;;

if (saco_1==2) or (saco_1==10) or (saco_1==18) or (saco_1==26) or (saco_1==34) or
(saco_1==42) or (saco_1==50) or (saco_1==58) then
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa2;;;;;;;;;;;;

if (saco_1==6) or (saco_1==14) or (saco_1==22) or (saco_1==30) or (saco_1==38) or
(saco_1==46) or (saco_1==54) then
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 4 capa 1;;;;;;;;;;;;

if (saco_1==4) or (saco_1==12) or (saco_1==20) or (saco_1==28) or (saco_1==36) or
(saco_1==44) or (saco_1==52) or (saco_1==60) then
PTP P4 CONT Vel= 100 % PDAT4 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 4 capa 2;;;;;;;;;;;;

if (saco_1==8) or (saco_1==16) or (saco_1==24) or (saco_1==32) or (saco_1==40) or
(saco_1==48) or (saco_1==56) then
PTP P8 CONT Vel= 100 % PDAT8 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;permiso para sacar palet;;;;;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then
```



```

pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[3]
wait sec 0.5
endif

;;;;;;;;;;;;;incremento sacco;;;;;;;;;;;;;

saco_1=saco_1+1

ptp_rel{z 110}c_ptp

;;;;;;;;;;;;;punto  elevado encima de zona 2

SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P24 CONT Vel= 100 % PDAT24 Tool[1] Base[0]
PTP P25 CONT Vel= 100 % PDAT25 Tool[1] Base[0]

;verificacion manos y dedos abiertos
continue
loop
continue
if (($in[27]==false) or ($in[28]==false)) then
$timer_stop[1]=false
endif
continue
if $timer[1]>=4000 then
wait sec 0
$timer_stop[1]=true
$timer[1]=0
pulse ($out[12],true,2)
endif
continue
if (($in[27]==true) and ($in[28]==true)) then
$timer_stop[1]=true
$timer[1]=0
exit
endif
endloop
END

```

## 12.7 ZONA\_111.src /ZONA\_222.src.

```
DEF zona_111( )
```

```
alt_incr_1=165
$out[30]=true
coger_1( )
```

```
;;;;;;;;saco 1 capa 1;;;;;;;;;;;;;
```

```
if (saco_1==1) or (saco_1==13) or (saco_1==25) then
PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

```
;;;;;;;;;;;;;saco 1 capa 2;;;;;;;;;;;;;
```

```

Zona_112.src      = Zona_222.src
DEF zona_111()    = DEF zona_222 ().
Coger_1 ( )       = Coger_2 ().
Saco_1            = sacco_2.
Base_data[3]      = base_data[1].
Base_data[4]      = base_data[2].
Sacos_1           = Sacos_2.
pulse_out[10]     = pulse_out[11].

```

Zona\_222.src sería igual que Zona\_111.src pero cambiando las variables.

```
if (saco_1==4) or (saco_1==16) or (saco_1==28) then
PTP P4 CONT Vel= 100 % PDAT4 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 1 capa 3;;;;;;;;;;;;

if (saco_1==7) or (saco_1==19) or (saco_1==31) then
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 1 capa 4;;;;;;;;;;;;

if (saco_1==10) or (saco_1==22) or (saco_1==34) then
PTP P10 CONT Vel= 100 % PDAT10 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 2 capa 1;;;;;;;;;;;;

if (saco_1==2) or (saco_1==14) or (saco_1==26) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 2 capa 2;;;;;;;;;;;;

if (saco_1==5) or (saco_1==17) or (saco_1==29) then
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 2 capa 3;;;;;;;;;;;;

if (saco_1==8) or (saco_1==20) or (saco_1==32) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 2 capa 4;;;;;;;;;;;;

if (saco_1==11) or (saco_1==23) or (saco_1==35) then
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa 1;;;;;;;;;;;;

if (saco_1==3) or (saco_1==15) or (saco_1==27) then
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
```

```
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 3 capa2;;;;;;;;;;;;

if (saco_1==6) or (saco_1==18) or (saco_1==30) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 3 capa3;;;;;;;;;;;;

if (saco_1==9) or (saco_1==21) or (saco_1==33) then
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 3 capa4;;;;;;;;;;;;

if (saco_1==12) or (saco_1==24) or (saco_1==36) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;permiso para sacar palet;;;;;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then
pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[1]
wait sec 0.5
endif

;;;;;;;;;;;;incremento sacco;;;;;;;;;;;;

saco_1=saco_1+1

ptp_rel{z 110}c_ptp

;;;;;;;;;;;;punto elevado encima de zona 2

SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P30 CONT Vel= 100 % PDAT30 Tool[1] Base[0]
PTP P31 CONT Vel= 100 % PDAT31 Tool[1] Base[0]

;verificacion manos y dedos abiertos
continue
loop
continue
if (($in[27]==false) or ($in[28]==false)) then
$timer_stop[1]=false
endif
continue
if $timer[1]>=4000 then
```

```
wait sec 0
$timer_stop[1]=true
$timer[1]=0
pulse ($out[12],true,2)
endif
continue
if (($in[27]==true) and ($in[28]==true)) then
$timer_stop[1]=true
$timer[1]=0
exit
endif
endloop
END
```

## 12.8 COGER\_1.src

```
DEF coger_1( )

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]
;punto de recogida
PTP P6 Vel= 80 % PDAT6 Tool[1] Base[0]
cerrar ()
wait sec 0.6
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]

PTP P8 CONT Vel= 100 % PDAT15 Tool[1] Base[0]
PTP P9 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

END
```

## 12.9 COGER\_2.src

```
DEF coger_2( )

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]
;punto de recogida
PTP P6 Vel= 80 % PDAT6 Tool[1] Base[0]
cerrar ()
wait sec 0.6
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]

PTP P8 CONT Vel= 100 % PDAT15 Tool[1] Base[0]
PTP P11 CONT Vel= 100 % PDAT11 Tool[1] Base[0]
PTP P9 CONT Vel= 100 % PDAT16 Tool[1] Base[0]
PTP P10 CONT Vel= 100 % PDAT10 Tool[1] Base[0]

END
```

## 12.10 CERRAR.src

```
DEF cerrar( )

pulse ($out[18],true,2)
pulse ($out[20],true,2)

END
```