

# Grado en Ingeniería Electrónica Industrial y Automática

## AUTOMATIZACIÓN PRÁCTICA II



## 2. SISTEMA DE TRANSPORTE Y LLENADO DE CAJAS

### 2.1. REQUISITOS PREVIOS

Para la realización de esta segunda práctica resultará imprescindible una lectura de los siguientes capítulos del libro "*Programación de autómatas con STEP 7*":

- Capítulos VII: Operaciones de bloques de bits, en particular las operaciones de comparación ([sección 7.3](#)).
- Capítulo X: Arquitectura de bloques, en particular las funciones FC y los bloques de datos ([secciones 10.1, 10.2 y 10.3](#))

### 2.2. CONTROL MEDIANTE SEÑALES ANALÓGICAS

Para incorporar magnitudes físicas a la lógica de un control mediante PLC es necesario un montaje apropiado sensor-alimentación cableado a una de las entradas de un módulo analógico conectado bien en la periferia local, bien en la distribuida.

En el laboratorio STEP 7 no se dispone de módulos analógicos, pero la carcasa tiene varias entradas analógicas. Una de ellas, la **EW752** es regulable desde la carcasa con tensiones entre 0 y 10V mediante un potenciómetro. Usaremos esta entrada para simular la medida del sensor de peso de nuestro sistema ejemplo.

Cualquier interfaz analógico de entrada al autómata debe convertir la señal en digital para que pueda ser usada en la lógica de control. Las tarjetas conversoras A/D



más habituales tienen una resolución de 16 bits para codificar la señal, esto es, una resolución máxima de  $\pm 32768$  valores para señales bipolares y  $+32768$  para señales unipolares. En los sistemas reales se ha demostrado que no es necesario usar la resolución máxima de la tarjeta, y el límite práctico se sitúa en  $\pm 27648$  para señales bipolares (0-27648 para el caso de unipolares). Un valor entre estos rangos es el que se almacena en el área de memoria de entradas correspondiente a la entrada analógica del sensor en cada ciclo.

En la lógica del control es habitual trabajar con las magnitudes físicas reales y no las muestreadas por la tarjeta. Por ejemplo, un sensor de peso transduce medidas entre 0y 50Kg, asociando tensiones entre 0 y 10V. La tarjeta A/D muestrea estos valores con rangos 0-27648 (señal unipolar). Sin embargo, el programador desea en la mayoría de los casos razonar con los valores en Kg dentro del programa. Para ello se emplea la función SCALE (FC105) correspondiente al bloque de librerías *standard*  $\rightarrow$  *TI-57 converting blocks* que viene con la distribución.

Un bloque de librería no es más que un bloque función que complementa las funcionalidades del PLC, con la particularidad de que está protegido y no puede ser modificado por el usuario. Un símbolo de candado asociado a su icono en la ventana principal indica dicha circunstancia (ver figura 1).

Nombre del objeto	Nombre simbólico	Lenguaje
OB1		
FC1	Pesaje	AWL
FC2		AWL
<b>FC105</b>	<b>SCALE</b>	<b>AWL</b>

Figura 1. Bloque SCALE protegido (ventana principal SIMATIC Manager)

Para importar cualquier bloque de librería lo más sencillo es arrastrarlo desde la ventana de librerías (situada en el extremo izquierdo de la pantalla) a cualquier segmento de la ventana de edición. La figura 2 muestra al bloque FC105 SCALE recién incorporado al proyecto. Falta ahora rellenar los parámetros con variables reales para que el bloque pueda compilar.

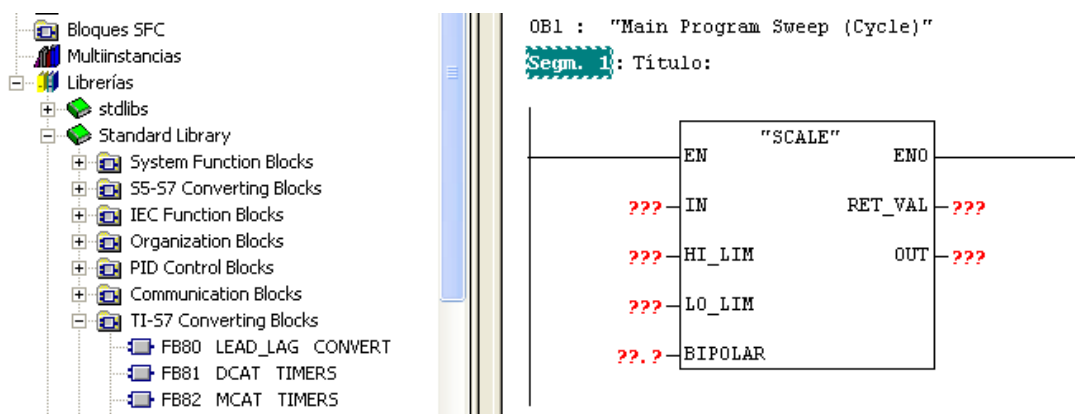


Figura 2. Bloque SCALE recién incorporado al bloque OB1 del proyecto

El bloque SCALE se emplea para convertir la señal leída del módulo analógico (con rangos  $\pm 27648$ ) a su magnitud física. El significado de los parámetros es el siguiente:

- IN: Señal transducida y muestreada de 16 bits (el rango dependerá del parámetro BIPOLAR). Tipo INT.
- HI\_LIM: Umbral superior de la magnitud física transducida. Tipo REAL.
- LOW\_LIM: Umbral inferior de la magnitud física transducida. Tipo REAL.
- BIPOLAR: TRUE para señales bipolares (rangos  $\pm 27648$ ). FALSE para señales unipolares (0-27648). El sensor de peso del ejemplo es unipolar.
- OUT: Valor convertido a la magnitud física. Tipo REAL.
- RET\_VAL: Valor de retorno que puede tratarse para comprobar el estado. un valor W#16#0000 indica lecturas correctas, mientras que cualquier otro valor (en concreto W#16#0008) indica que se ha producido un desbordamiento (por arriba de HI\_LIM o por debajo de LOW\_LIM).

Para una información más detallada consulte el alumno los manuales que acompañan la distribución de STEP 7. Para ello, basta con importar la función al proyecto, seleccionarla con el ratón en la ventana de edición y pulsar F1.

## 2.3. SIMULACIÓN DE VARIABLES ANALÓGICAS

Para la simulación de variables analógicas en PLCSIM se configura el visor típico de una variable de tamaño palabra (e.g. MW10) como regulador, lo que permite modificar sus valores de manera continua mediante el típico recurso gráfico de aguja. Existen dos tipos:

- Decimal: Señal unipolar. Valores entre 0 y 32768
- Entero: Señales bipolares, con rangos  $\pm 32768$ .

Para modificar en simulación valores de coma flotante, el visor del área de memoria debe apuntar a un dato tamaño doble palabra (e.g. MD10) y estar configurado como regulador *real*. La [figura 3](#) muestra una sesión con 3 visores en modo de regulación.

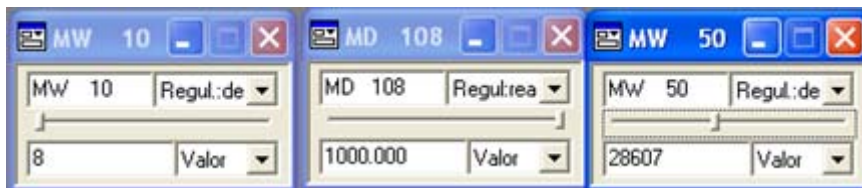


Figura 3. Visores configurados como reguladores en PLCSIM

## 2.4. OBJETIVO DE LA PRÁCTICA

Uno de los objetivos de esta práctica es familiarizar al alumno con el uso de bloques función y funciones parametrizadas. El paso de parámetros es el mecanismo de paso de información entre el bloque invocante e invocado, semejante al mecanismo de paso por valor de argumentos a una función en un lenguaje de alto nivel. El otro

objetivo es programar un control real que tenga como entrada una señal analógica transducida. En el sistema ejemplo es necesario leer una báscula con rangos de entrada al PLC [0V-0Kg, 10V-50Kg].

Para ello el alumno tendrá que programar un control sencillo de una parte de un sistema de transporte y llenado de cajas; sus especificaciones se describen en detalle en las secciones [2.6](#), [2.7](#) y [2.8](#).

## 2.5. NORMATIVA GENERAL DE PROGRAMACIÓN

Tanto en esta como en el resto de prácticas de programación será imprescindible cumplir con el conjunto de requisitos generales siguientes, con independencia de las instrucciones concretas que se detallen en cada caso. Éstos son:

- Proyecto de programación STEP 7 debidamente configurado
- Tabla de símbolos adecuada
- Sesión de simulación PLCSIM configurada adecuadamente para mostrar las variables fundamentales del control y los símbolos. Dicha sesión deberá incluir un archivo de organización (extensión \*.lay) dentro del proyecto STEP 7 que guarde la configuración predefinida por el alumno. De esta manera se asegura la portabilidad del proyecto para su simulación en otro equipo.
- Caso de requerirse la entrega del proyecto STEP 7 completo se generará una versión comprimida (\*.zip) desde la ventana principal (VP) del Administrador SIMATIC mediante los comandos **VP:Archivo**→**Archivar** y la selección del proyecto del alumno. Este archivo comprimido, y solo éste, es el que se entregará al profesor.
- Abundantes comentarios: línea, segmento y bloque
- Estructuración adecuada de los segmentos en cada bloque

## 2.6. SISTEMA DE TRANSPORTE Y LLENADO DE CAJAS

Se pretende controlar una zona de llenado de cajas junto con su transporte posterior. El sistema consta de 3 cintas que se mueven a velocidad de 10cm/s en régimen permanente. La cinta 1 lleva el material de relleno, por ejemplo arena blanca fina con fines decorativos. La cinta 2 transporta las cajas vacías donde se vuelca la arena. Las cajas tienen un largo de 40cm y un ancho que no afecta al control. El sensor óptico A determina la posición donde se produce la carga. En ese punto una báscula controla la fase de llenado hasta los 40Kg de peso (incluyendo la caja).



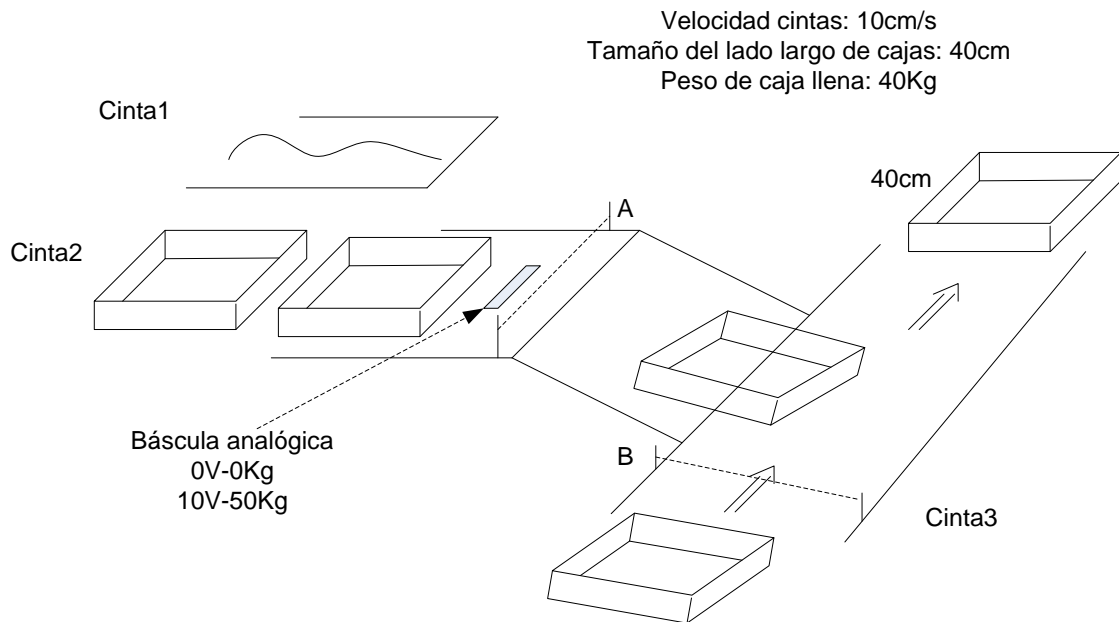


Figura 4. Transporte y llenado de cajas

Una vez llenada una caja, la cinta 2 se vuelve a poner en marcha, lo que hace que la caja se deslice por una rampa hasta una cinta 3 que se encarga de evacuarla. Esa cinta 3 puede también transportar otras cajas del mismo tamaño pero provenientes de otra zona de la planta. La [figura 4](#) ilustra la trayectoria de cajas descrita.

Las cajas por la cinta 2 pueden estar muy próximas entre sí, incluso rozándose. Para resolver el problema en la desembocadura, un sensor B situado en las proximidades permite conocer si hay una caja con peligro de colisión. El control debe conseguir que entre las cajas de la cinta 3 haya un espacio de al menos el doble de la longitud de una caja (80cm).

## ESPECIFICACIONES GENERALES

Se considera una botonera simplificada con un único pulsador de marcha *Pon*. El sistema para la producción después de haber llenado 100 cajas. La última caja debe evacuarse como las anteriores (se considera un tiempo límite de evacuación de 50s). No se considera la parada de emergencia, pero si se considera defecto valores por desbordamiento en la lectura del sensor de peso.

## 2.7. MODELADO

El modelado de control graficet se detalla en la [figura 5](#). Consta de dos ramas de control, una para la cinta 3 y otra para las cintas 1 y 2. La evacuación de la última caja (la número 100) se realiza mediante divergencia OR en la etapa 8, con actuadores de evacuación asociados a la etapa 9. Las etapas 10 y 11 permiten sincronizar la vuelta al reposo. Existen otras soluciones no ortodoxas, como emplear una transición pozo en la etapa 9 sin vuelta al reposo pero es conveniente en la medida de lo posible hacer graficet claros y con ciclos de producción estables.

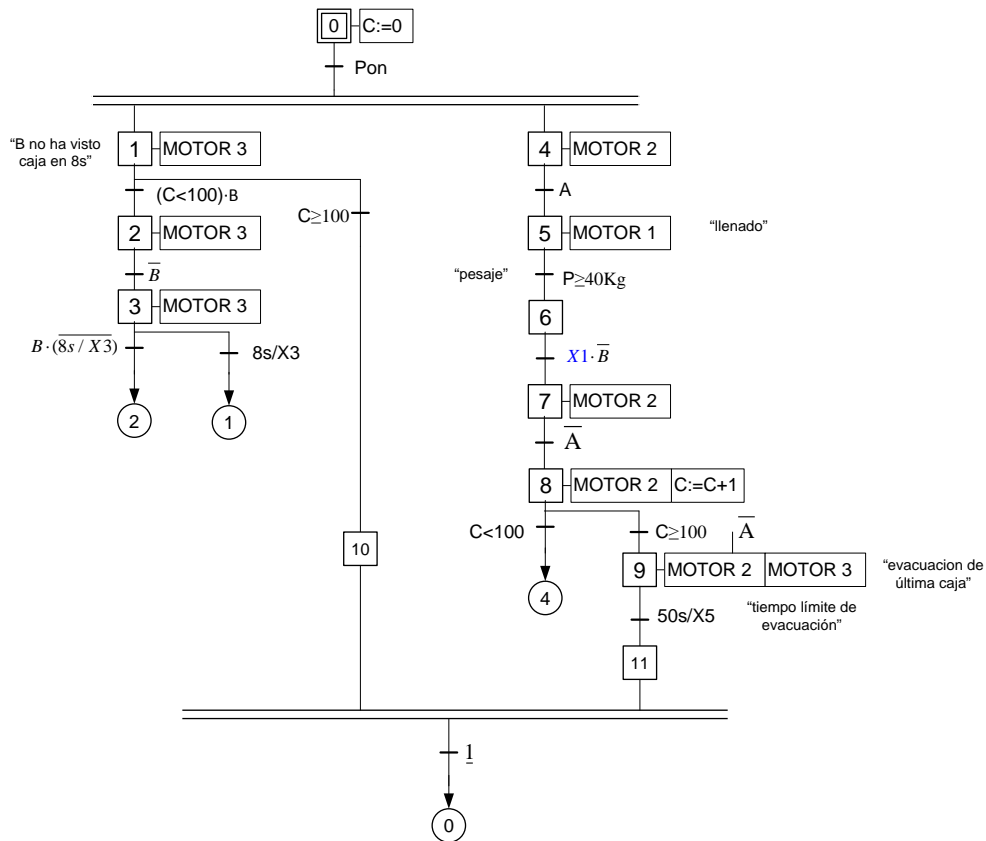


Figura 5. Modelado grafcet (en azul la coordinación horizontal)

**Nota:** el alumno debe ceñirse en la programación del control a los estados del grafcet de la figura 5 en la medida de lo posible.

## 2.8. ESTRUCTURACIÓN DEL PROGRAMA

Tabla 1. Interfaz de una función que controla la lectura de peso de una báscula

<pre> <b>VAR_INPUT</b> in : INT ; //señal muestreada (tarjeta AD) event : BOOL ; //evento que valida el pesaje <b>END_VAR</b> <b>VAR_OUTPUT</b> weight : REAL ; //salida en Kg overflow : BOOL ; //desbordamiento full : BOOL ; //TRUE si peso ≥40 kg <b>END_VAR</b> <b>VAR_TEMP</b> retFC105 : WORD ; //RET_VAL de FC105 <b>END_VAR</b>                 </pre>	<pre> <b>VAR_INPUT</b> reset : BOOL units : WORD ; add : BOOL ; //TRUE→suma <b>END_VAR</b> <b>VAR_OUTPUT</b> end : BOOL ; //TRUE counter&lt;100 <b>END_VAR</b>                 </pre>
<b>FC: "Pesaje"</b>	<b>FC: "Contador"</b>



Sin parámetros.  Gestiona todas las salidas del sistema	Sin parámetros.  Gestiona la secuencia de escape debido a valores fuera de rango obtenidos en el sensor de peso	Sin parámetros.  Gestiona la configuración de los temporizadores exclusivamente, (no su amortización)
<b>FC: "Actuadores"</b>	<b>FC: "Escape"</b>	<b>FC: "Config Temp"</b>

Para superar correctamente esta práctica se pide la estructuración en bloques del programa STEP 7 con los interfaces que aparecen en la [tabla 1](#).

La función "Pesaje" implementa la lectura y la lógica del sensor, por lo que debe invocar al bloque protegido "SCALE" (FC105).

**IMPORTANTE:** Para poder simular desde el PLC la señal de entrada de la báscula con el potenciómetro situado en la carcasa es necesario usar la entrada **EW752**. Para una lectura directa de los valores de la tarjeta A/D sin pasar por el área de memoria utilícese el direccionamiento **PEW752**, que accede directamente a la periferia.

El programa debe contener también un bloque de datos **DB1** que almacena los valores nominales para el límite de unidades de producción (100) y el peso de las cajas llenas (40kg) ([figura 6](#)). Esto facilita modificar estos parámetros en el futuro sin tener que recompilar el código.

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	unidades	WORD	W#16#64	numero de unidades a producir
+2.0	pesolim	REAL	4.000000e+001	peso de las cajas llenas
=6.0		END_STRUCT		

Figura 6. Bloque de datos DB1

A título aclarativo, una posible vista de la ventana principal del proyecto aparece en la [figura 7](#).

Nombre del objeto	Nombre simbólico	Lenguaje
OB1	main	AWL
OB100	arranque	AWL
FC1	Pesaje	AWL
FC2	Contador	AWL
FC3	Temporizadores Config	AWL
FC4	Actuadores	AWL
FC5	escape	AWL
FC105	SCALE	AWL
DB1	datos	DB

Figura 7. Arquitectura de bloques



**Nota importante:** Es imprescindible que el alumno se atenga estrictamente a la arquitectura de bloques descrita y respete la nomenclatura de las variables que componen el interfaz.

## 2.9. TAREAS A REALIZAR PARA EL INICIO DE LA PRÁCTICA

Al inicio de la práctica el alumno (además de cumplir con los requisitos previos recogidos en la [sección 2.1](#)) debe traer un proyecto software S7 con una función FC1 que gestiona la lectura de peso unipolar que llega desde una báscula con rangos [0(0Kg)-27648(50Kg)]. Dicha función debe responder al interfaz de la [tabla 2](#) y su implementación responder a la semántica indicada.

Tabla 2. Interfaz de una función que controla la lectura de peso de una báscula

<b>VAR_INPUT</b>	
in : INT ;	//señal muestreada (tarjeta AD)
<b>END_VAR</b>	
<b>VAR_OUTPUT</b>	
weight : REAL ;	//salida en Kg
overflow : BOOL ;	//desbordamiento
<b>END_VAR</b>	
<b>VAR_TEMP</b>	
retFC105 : WORD ;	//RET_VAL de FC105
<b>END_VAR</b>	
<b>FC1: "tratamiento de una señal analógica"</b>	

### TAREA 1

Al comienzo de la práctica el alumno deberá simular correctamente la función FC1 propuesta desde PLCSIM; para ello debe existir al menos un visor de regulación para la señal de entrada (in) muestreada y un visor para ver la magnitud física (weight) configurado como REAL.

El desbordamiento se producirá cuando se introduzcan valores superiores a 27648 en la entrada.

### TAREA 2

Además el alumno debe analizar detenidamente el modelo grafcet del sistema descrito y responder a las siguientes preguntas:

1. ¿Por qué no existe un retorno al reposo de las dos ramas del grafcet?
2. ¿Existe el riesgo de una implementación con etapas fugaces? En caso afirmativo indique dichas etapas
3. Indique para cada estado del grafcet la interpretación semántica del control



## 2.10. ENTREGA AL PROFESOR

Se pide:

*Programa de control STEP 7 sujeto a la normativa general de la sección 2.4 para el control del sistema de transporte y llenado de cajas descrito.*

**El código del programa de control se entregará al profesor en papel al comienzo de la práctica 3.**

