Prácticas Presenciales

GRUPO SAN VALERO

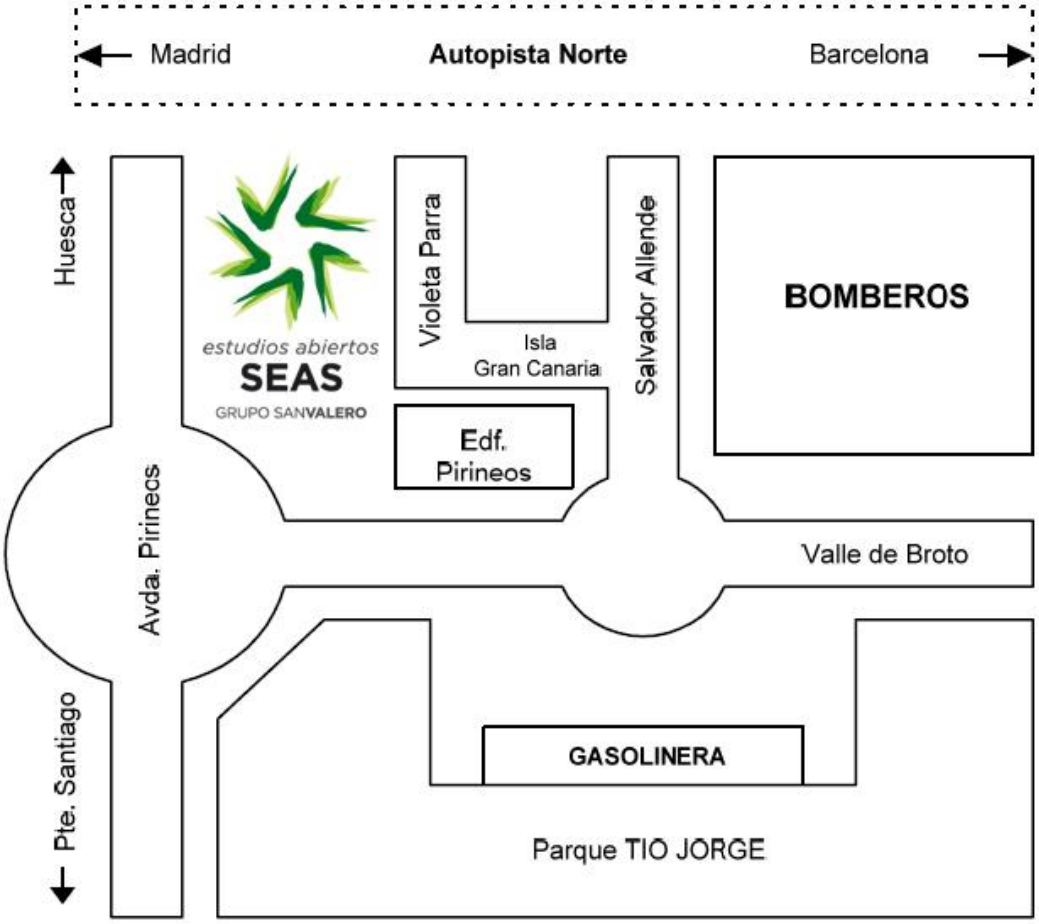
Estudios abiertos

SEAS

"Autómatas Programables Nivel Avanzado"

Área: (M005) Autómatas Programables

LUGAR DE CELEBRACIÓN
Instalaciones de Fundación San Valero, en c/ Violeta Parra 9
50015 Zaragoza
Planta E, de 10:00 a 14:00 h.



Las líneas de autobús que tienen parada en las proximidades de Fundación San Valero son: 29, 36, 35, 45, 42 y Ci1.
Para más información visitar la página Web de TUZSA. http://www.tuzsa.es

PRÁCTICAS PRESENCIALES
Autómatas Programables Nivel Avanzado

DESCRIPCIÓN:

Durante la jornada presencial se pondrán en práctica los conocimientos adquiridos durante el estudio del módulo correspondiente a la asignatura de **Autómatas Programables Nivel Avanzado.**

REQUISITOS:

Es requisito para la realización de la práctica, haber cursado la asignatura de Autómatas Programables y haber trabajado las 4 primeras unidades didácticas de la asignatura de Autómatas Programables Nivel Avanzado.

OBJETIVOS:

1. Conocer los pasos a seguir a la hora de realizar la automatización de máquinas.
2. Conocer los elementos empleados en detección y su tratamiento por parte del PLC.
3. Conocer los actuadores y su tratamiento por parte del autómata.
4. Trabajar con módulos de función y módulos de datos.
5. Conocer el entorno de desarrollo de aplicaciones para autómatas SIEMENS.
6. Conocer los fundamentos de programación en lenguaje AWL.
7. Conocer los modos de estructuración y ordenación a la hora de programar
8. (Funciones).

PROPUESTA DE LA PRÁCTICA:

1. Automatización de una máquina. (Control de cintas transportadoras).

2. Direccionamiento de los distintos dispositivos a conectar al PLC.

3. Estructuración del programa. Uso de funciones y módulos de datos.

4. Desarrollo del programa.

5. Transferencia y modos de funcionamiento del autómata.

6. Pruebas de funcionamiento del sistema.

DESARROLLO DE LA PRÁCTICA:

Planteamiento

Basándonos en el diagrama técnico de la máquina y la secuencia de funcionamiento, deberemos realizar un estudio preliminar de cara a determinar las necesidades a nivel de equipamiento. En función del número de detectores y actuadores realizaremos el dimensionado de entradas y salidas necesarias en nuestro PLC.

Estructuración y programación

Una vez planteadas las pautas básicas se procederá a estructurar la aplicación mediante funciones, generando aquellas que sean necesarias mediante el entorno de desarrollo.

Transferencia y pruebas de funcionamiento

Una vez finalizada la programación de los distintos bloques de programa, se procederá a realizar la transferencia de los mismos al PLC, prestando atención a los distintos modos de funcionamiento de estos equipos.

Con el programa transferido a la memoria del autómata, simularemos mediante el equipo la funcionalidad del sistema.

BIBLIOGRAFÍA:

Manual de las asignaturas de Autómatas Programables y Autómatas Programables Nivel Avanzado de SEAS

MATERIALES NECESARIOS:

Para la realización de las prácticas, es necesaria la disponibilidad de:

Autómata programable S7-300 CPU 314-IFM.

Ordenador personal.

Entorno de desarrollo STEP-7.

Adaptador de comunicaciones PC-ADAPTER.

DURACIÓN ESTIMADA:

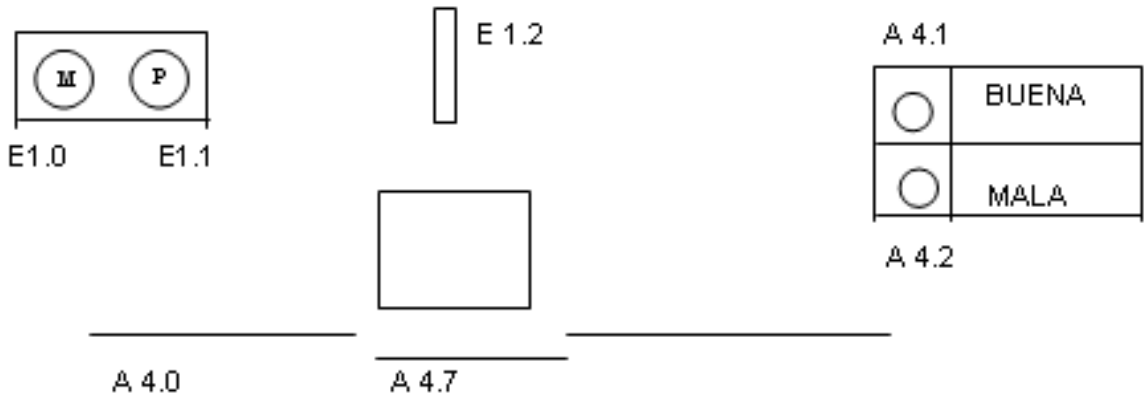
4 horas.

SOLUCIÓN:

Tenemos una cinta transportadora que lleva los productos. Lo que queremos es que los productos que se salgan del peso estipulado tanto por arriba como por abajo, se desechen.

Además tenemos unas luces indicadoras de producto bueno o de producto malo. Queremos que mientras está la pieza debajo de la célula detectora, se encienda la luz correspondiente.

Diagrama técnico:



Para ello vamos a tener en un módulo de datos los límites de peso tanto superior como inferior.

Con el byte 0 de entradas vamos a simular los pesos de las piezas que circulan por la cinta. Queremos que cuando la pieza tenga un peso comprendido entre los límites correctos se encienda una luz que nos indique que la pieza es correcta. Cuando pase una pieza que su peso se salga de los límites queremos que nos indique una luz que la pieza es mala y a la vez que se abra una trampilla para que la pieza salga de la cinta.

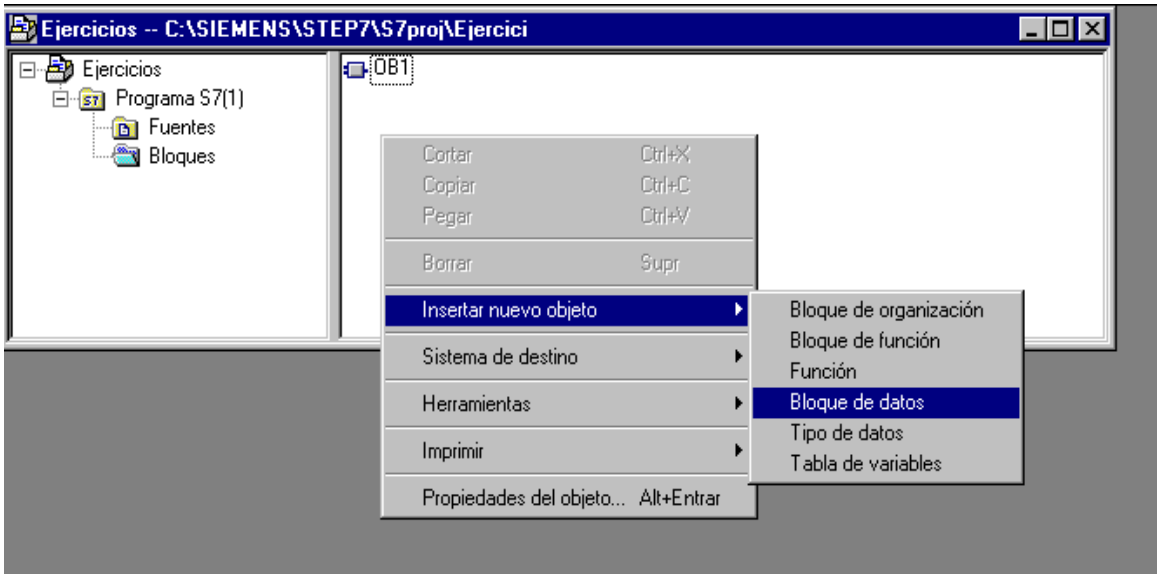
Teoría y ejemplo previo:

Vamos a crear un bloque nuevo. Vamos a hacer un bloque de datos. En estos bloques lo único que vamos a introducir son datos. No vamos a programar nada. Va a ser una tabla con datos los cuales podemos leer y podemos escribir nuevos valores en los datos.

Para crear un DB vamos al administrador de SIMATIC. Nos situamos en la parte izquierda encima de donde pone bloques. Lo podemos hacer tanto en ONLINE como en OFFLINE. En la parte derecha tenemos todos los bloques que tenemos creados hasta ahora.

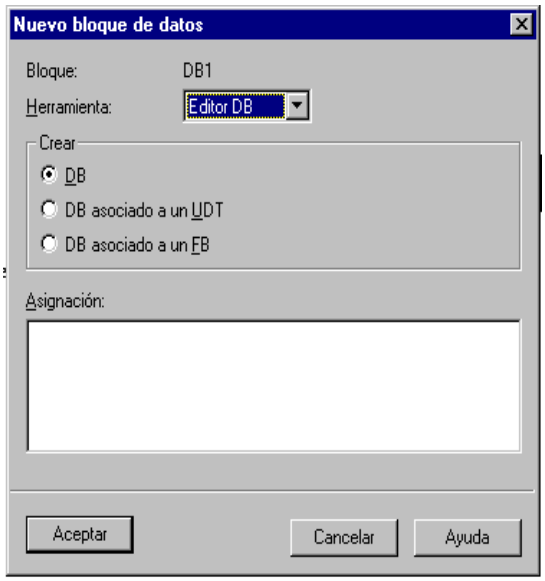
Tenemos que insertar un bloque nuevo. Pinchamos en la parte derecha con el botón derecho del ratón y aparece una ventana donde le decimos que queremos insertar un nuevo objeto.

También lo podemos hacer en el menú de “Insertar” que tenemos en la barra de herramientas.



Insertamos un bloque de datos.

Cuando entramos en el nuevo DB vemos que nos ofrece tres posibilidades. Podemos hacer un DB asociado a una FB, podemos hacer un DB asociado a un UDT, y podemos hacer un DB simple.



En principio lo que queremos hacer es un DB simple.

Una vez lo tenemos creado veremos que sale el icono del DB junto con los demás bloques.

Le podemos dar el número de DB que nosotros queramos. Por ejemplo podemos hacer el DB 1.

Entramos en el DB y vamos a rellenarlo.

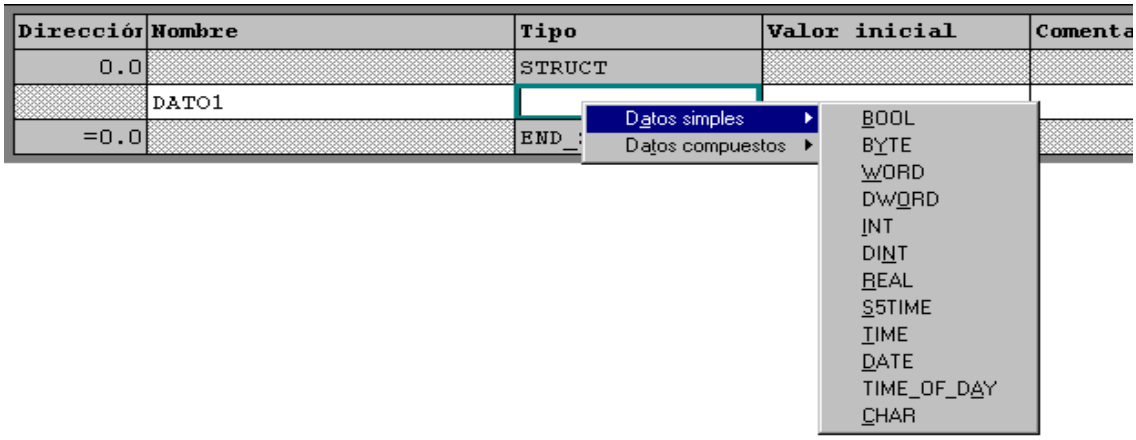
El DB es una tabla en la que tenemos que rellenar los datos. Tenemos varios campos para rellenar.

Direcció	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
=0.0		END_STRUCT		

El primer campo que tenemos es la dirección. De esto no nos tenemos que preocupar. Es automático. Conforme vayamos creando datos, se le van asignando las direcciones que corresponden.

El siguiente campo es el nombre. Es obligatorio dar un nombre a cada dato que vayamos a introducir.

A continuación tenemos que definir el tipo de datos que va a ser. Si no sabemos como definir un tipo de datos pinchamos en la casilla correspondiente con el botón derecho del ratón y podemos elegir el tipo de datos que queremos.



A continuación tenemos que definir el valor inicial. Tenemos que poner el valor en el formato que corresponde. Si no sabemos el formato o no queremos ningún valor inicial, por defecto nos pone un 0, pero escrito en el formato correcto.

Ya tenemos el primer dato definido.

Vamos a crear la siguiente tabla:

VALOR_1	INT	0
VALOR_2	REAL	3.0
VALOR_3	WORD	W#16#0

Esto en un DB quedaría de la siguiente manera:

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	VALOR_1	INT	0	
+2.0	VALOR_2	REAL	0.000000e+000	
+6.0	VALOR_3	WORD	W#16#0	
=8.0		END_STRUCT		

Ya tenemos un DB. Vamos a ver ahora como trabajamos con él.

Vamos a leer el valor 3.0, y vamos a escribir un valor en VALOR_3.

Para acceder a los datos de un DB tenemos dos posibilidades. Podemos abrir primero un DB y luego acceder directamente a la palabra de datos, o podemos decir cada vez a que DB y a que palabra de datos queremos acceder.

```
AUF   DB    1          L     DB1.DBW   0
L     DBW   0          T     MW        10
T     MW    10         BE
BE
```

Para acceder a un dato, le llamamos DB.... Puede ser DBB si es un byte, DBW si es una palabra, DBD si es una doble palabra o DBX 0.0 si es un bit.

También podemos dar un nombre al DB y acceder a los datos por su nombre. Por ejemplo en el DB que hemos creado antes podríamos acceder a los datos del siguiente modo:

```
L     DATOS.VALOR_1
```

DATOS es el nombre del DB y VALOR_1 es el nombre del dato.

Vamos a ver como leemos un valor y lo metemos en una marca, y como escribimos un valor y luego lo vamos.

Vamos a leer el valor 3.0 que tenemos en VALOR_2 y vamos a ponerlo en la doble palabra de marcas 100. Tiene que ser en una doble palabra de marcas porque es un número real.

```
AUF   DB    1
L     DBD   2
T     MD    100
```

Ahora vamos a escribir un valor en VALOR_3.

```
L     8
T     DB1.DBW6
```

Hemos metido un 8 en este valor.

Vamos a entrar en el DB y vamos a ver como vemos este nuevo dato que hemos introducido.

Entramos en el DB y lo único que vemos es la columna de valores iniciales. Aquí vamos que sigue habiendo un 0.

Si vamos al menú de Ver tenemos la opción de Ver > datos.

Vemos que el DB tiene una nueva columna que es la de datos actuales. Aquí veremos que tenemos el nuevo valor. Esta pantalla no refresca valores. Hace una lectura cuando abrimos la ventana. Si hacemos algún cambio y queremos visualizarlo, tendremos que cerrar y abrir de nuevo la pantalla.

Para ver el dato que acabamos de introducir, tenemos que hacer la operación en ONLINE. Si vamos a ver datos en OFFLINE, veremos los datos actuales de la CPU del PC. El PC no ha ejecutado programa. Veremos que los valores iniciales son los mismos que los actuales.

Tenemos que tener en cuenta que estos valores se guardan en disco duro. Cada vez que transfiramos el programa bien de OFFLINE a ONLINE o viceversa, también se transfieren los valores actuales.

Si queremos dejar el DB como estaba al principio con sus valores iniciales para comenzar el proceso de nuevo, vamos al menú de edición y tenemos la posibilidad de reinicializar bloque de datos. Los valores actuales se cambian por los valores iniciales. A partir de ahí hace lo que se le diga por programa.

Si el DB ha adquirido unos valores distintos de los iniciales los guarda en esta columna de valores actuales. Cada vez que transferimos al autómata el DB también estamos transfiriendo esta columna de valores actuales aunque no la veamos.

Si queremos volver a los valores iniciales no tenemos más remedio que reinicializar el DB y transferir estos cambios al autómata.

PROGRAMA Y TABLAS DE DATOS:

```
      U     E     1.0          //Cuando le demos al botón de marcha
      S     A     4.0          //Pon en marcha la cinta transportadora
      U     E     1.1          //Cuando le demos al botón de paro
      R     A     4.0          //Para la cinta
      U     E     1.2          //Cuando pase la pieza por la balanza
      SPB   M001
      R     A     4.1          // Si no hay pieza manten todo apagado.
      R     A     4.2
      R     A     4.7          //Salta a la meta 1
      BEA
                                //Si no hay pieza  termina el programa
M001: L     EB    0            //Carga el peso de la pieza
      L     DB1.DBW0           //Carga el límite superior de peso
      <I                       //Si la pieza pesa menos que el limite sup.
      =     M     0.0          //Activa la marca 0.0
      L     EB    0            //Carga el peso de la pieza
      L     DB1.DBW2           //Carga el límite inferior
      >I                       //Si la pieza pesa más que el limite inf.
      =     M     0.1          //Activa la marca 0.1
      U     M     0.0          //Si está activa la marca 0.0
      U     M     0.1          //Y está activa la marca 0.1
```

```
S     A     4.1          //Enciende la luz de pieza buena
R     A     4.2          //Apaga la luz de pieza mala
NOT                      //En caso contrario
S     A     4.2          //Enciende la luz de mala
R     A     4.1          //Apaga la luz de buena
U     A     4.2          //Si la pieza es mala
=     A     4.7          //Abre la trampilla de desecho
BE
```

Hemos visto como hacer una carga y una transferencia condicionada por una entrada. Para hacerlo condicional lo podemos hacer de dos maneras. Bien lo hacemos como en este ejemplo utilizando metas, o bien lo hacemos con llamadas a otros módulos de modo condicional.

Si se cumple la condición salta al módulo que le digamos y allí tendremos la carga y la transferencia. Si no se cumple la condición no salta al módulo en cuestión y por tanto no lee las instrucciones de carga y transferencia.

Para completar el ejercicio, nos falta hacer el módulo de datos donde vamos a guardar el límite superior y el límite inferior de pesado para las piezas.

Vamos a crear el DB1.

En el DB1 vamos a crear dos palabras de datos. Tenemos que definir un tipo para cada uno de los datos. En este caso van a ser dos números enteros. Diremos que son de tipo (INT).

También podemos suponer que son valores reales. Para ello lo deberíamos haber tenido en cuenta a la hora de programar.

Para comparar números reales tenemos otra instrucción, y además los números reales se guardan en dobles palabras, no en palabras simples.

Para guardar los datos deberíamos haber utilizado la doble palabra de datos 0 y la doble palabra de datos 4.

Veamos como quedaría el DB.

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	PESO_MAXIMO	INT	25	
+2.0	PESO_MINIMO	INT	20	
=4.0		END_STRUCT		

PRÁCTICAS PRESENCIALES
Autómatas Programables Nivel Avanzado