

## SIMULADOR DIDÁCTICO DEL AUTÓMATA TSX MICRO DE SCHNEIDER

J. L. CALVO Y A. MASDIAS

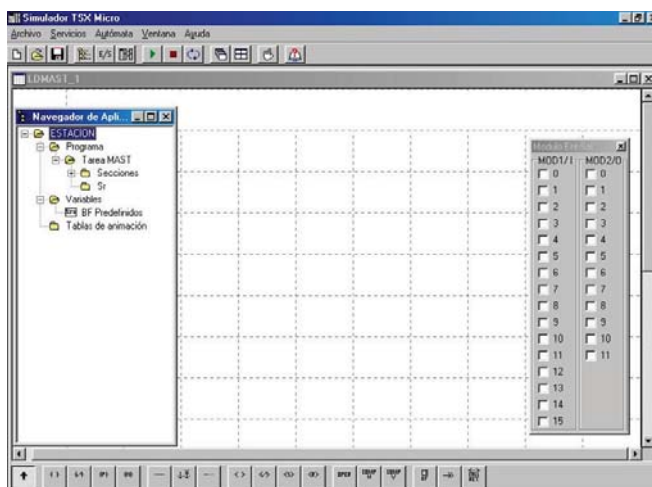
*Departamento de Ingeniería Industrial. Escuela Universitaria Politécnica de Ferrol. Universidad de La Coruña. Avda. 19 de Febrero s/n 15403-Ferrol, La Coruña. España.*

*Este documento pretende explicar el funcionamiento de un simulador del autómata programable TSX Micro de Schneider, enfocado a la programación en lenguaje de contactos. Se intenta reproducir en la medida de lo posible el entorno de trabajo del software que facilita el fabricante, para que los conocimientos adquiridos tengan validez casi en su totalidad, a la hora de enfrentarse al equipo real.*

### 1. Introducción

Los fabricantes de productos destinados a control, en ocasiones son reacios a desarrollar herramientas de simulación que sean independientes. Desde el punto de vista del aprendizaje práctico, el tiempo dedicado se va a ver limitado por la disponibilidad, tanto de las aulas, como del personal encargado de las mismas. Es por tanto necesaria la creación de simuladores para paliar estos inconvenientes. Para el caso que se expone, el software implementado desarrolla una aplicación bajo entorno windows que permite programar y simular el funcionamiento del autómata TSX MICRO de Schneider.

### 2. Programación en el simulador del TSX Micro de Schneider



**Figura 1:** Entorno del simulador

El simulador ha sido concebido para que la edición del programa que va a ser simulado se haga en lenguaje de contactos. Con este simulador en la mayoría de los casos ya se puede prescindir de un entrenador para activar las entradas, ya que está preparado para visualizar y modificar el estado de entradas y salidas, incluso forzarlas a un valor, independientemente de las combinaciones que se puedan provocar, de igual forma que si se estuviese programando en el propio autómata. Se ha intentado en la medida de lo posible, que el entorno de trabajo del

simulador, sea lo mas parecida al del software que el fabricante proporciona. Una vez instalado el programa se puede arrancar abriendo el fichero ejecutable por cualquiera de los medios que windows nos permite. Realizado esto al iniciarse la sesión, se pueden observar diversas partes (figura 1) que se detallarán posteriormente.

## 2.1. El navegador de aplicación

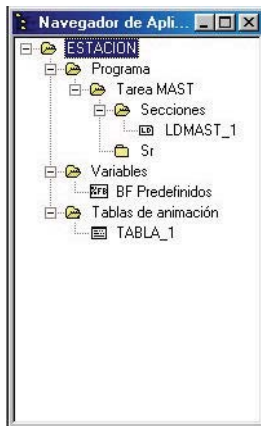


Figura 2: Navegador de aplicación

Esta parte del simulador tiene una estructura similar a la del explorador de windows (figura 2). Tenemos en primer lugar una carpeta principal, que será la correspondiente al proyecto con el que se esté trabajando. De esta se despliegan tres nuevas que detallan a continuación:

**-Carpeta PROGRAMA.-** A su vez se despliega otra denominada Tarea MAST, cuyos contenidos son creados en la ventana de edición gráfica, y está compuesta por, una carpeta en donde se encuentran las secciones de programa, y otra en donde se albergan las subrutinas empleadas en la programación, pertenecientes a la tarea actual. Para añadir en la tarea Mast, tanto nuevas secciones como nuevas subrutinas, simplemente es necesario seleccionar la carpeta correspondiente en donde necesito insertar un nuevo componente, y a continuación pulsar con el botón derecho del ratón, siguiendo las instrucciones que el programa indica.

Variable	Símbolo	Preset	Modo	TB	Comentario	%TMI.V
%TMP0		9999	TOFF	1 min		
%TMP1		9999	TP	1 s		
%TMP2		9999	TON	100 ms		
%TMP3		9999	TON	10 ms		
%TMP4		9999	TON	1 min		
%TMP5		9999	TON	1 min		

Figura 3: Variables – BF Predeterminados

**-Carpeta VARIABLES.-** De esta carpeta cuelga un único archivo cuyo nombre es 'BF Predefinidos'. Si se hace doble clic sobre el se activa una ventana (figura 3) que lleva el mismo nombre, la cual contiene los parámetros de los bloques de función, y que el usuario puede modificar a su gusto para realizar la simulación de la forma deseada.

**-Carpeta TABLAS DE ANIMACIÓN.** De esta carpeta cuelga un archivo cuyo nombre es 'TABLA\_1', que si se hace doble clic sobre ella se visualizará una ventana como la de la figura 4. Pulsando con el botón derecho del ratón sobre la carpeta 'Tablas de animación', se pueden crear nuevas tablas, y así tener grupos de variables localizados. Esta ventana es utilizada para visualizar estados de objetos bit de entradas, salidas, memoria o de salida de bloques; así como se pueden mostrar también valores de preselección o estados actuales de contaje o temporización de esos bloques función. Estas tablas permiten además modificar los valores de preselección o el estado, a 0 ó 1, de los objetos bit de entradas, salidas, memoria o de salida de bloques. Estos valores se pueden forzar a 0 ó 1, independientemente del valor que tengan que adoptar resultante de las distintas configuraciones que adquieran las variables.

Modificación	Variable	Valor
1	%TI.0	0
2	%TMI.P	9999
3	%CO.D	0
4		
5		
6		
7		
8		
9		
10		

Figura 4: Tablas de animación

## 2.2. Menús desplegables del simulador

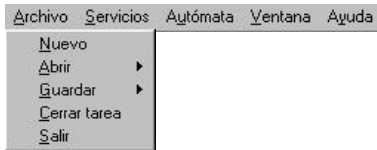


Figura 5: Menú archivo

tanto la opción de Abrir como la de Guardar tienen en su margen derecho unos triángulos que indican que esta opción posee a su vez otras.

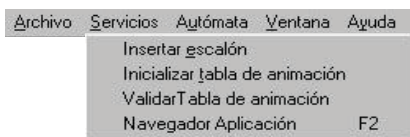


Figura 6: Menú Servicios

inicialización y validación de las tablas de animación ya comentadas, y posteriormente se pueden validar con otra de las ordenes. Por último dentro de este apartado comentar que también se puede acceder al Navegador de Aplicación.



Figura 7: Menú Autómata

-**Autómata.** Desde este menú que se muestra en la figura 7 el programa nos permite realizar tareas tales como iniciar y parar la simulación, visualizar el módulo de entradas y salidas, y además ampliar este, mostrando el módulo 3 de entradas y 4 de salidas simultáneamente al escoger esta opción.

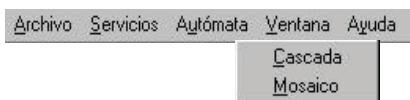


Figura 8: Menú ventana

-**Ventana.** Este menú presentado en la figura 8 muestra dos opciones que sirven para organizar las distintas ventanas que estén activas en el simulador. Dicha organización se puede hacer en forma de Mosaico, o bien en forma de Cascada. Simplemente pincharemos sobre una de ellas con el ratón y las colocará.

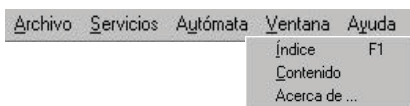


Figura 9: Menú Ayuda

-**Ayuda.** Tiene una forma similar a los programas bajo entorno Windows. Para este caso posee tres opciones tal y como se muestra en la figura 9. Las dos primeras sirven para conseguir ayuda, y la última da información acerca de la creación del programa.

## 2.3. Barra de herramientas

La mayoría de las ordenes que se pueden llevar a cabo con los menús desplegables, y en especial aquellas que tienen un uso más frecuente, se ven reflejadas en las barras de herramientas, para así poder ejecutar esas funciones de una forma más rápida y sencilla. Se

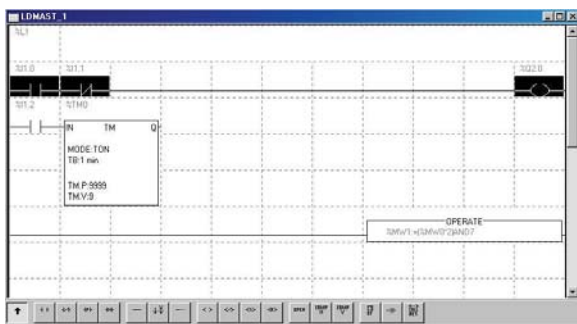


**Figura 10:** Barra de herramientas

muestran en la figura 10 las barras de herramientas que posee este simulador, junto con una leyenda que indica la función que tiene cada uno de los botones.

## 2.4. Ventana de edición gráfica

La edición del circuito se lleva a cabo en esta ventana acondicionada a tal fin (figura 11). En primer lugar, una vez creada, lo que se hace es pinchar sobre el elemento que necesitamos insertar en el circuito (cada elemento tiene asociado en la parte baja de la ventana un botón), y a continuación lo posicionamos pinchando con el puntero del ratón sobre la celda en donde se quiera insertar. Así se van seleccionando los distintos elementos y conformando el circuito.



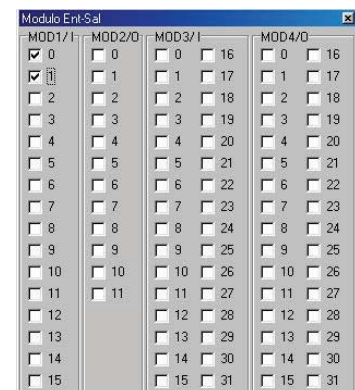
**Figura 11:** Ventana de edición gráfica

## 2.5. Simulación de una tarea

Para la ejecución de una tarea, previa edición del circuito, es necesario dirigirse al menú Automata y seleccionar la opción Run para empezar a simular. Si estamos simulando y lo que queremos es parar, hay que dirigirse al mismo menú y seleccionar la opción Stop. El estado de las entradas y salidas se puede observar, bien en el Módulo de Entradas Salidas explicado en el siguiente punto, bien en la propia ventana de sección o subrutina en la que se marcarán las celdas correspondientes a los contactos activos.

## 2.6. Modulo de entradas salidas

Una vez editado el programa en lenguaje de contactos, se puede proceder a la simulación. Además de las tablas de animación ya comentadas tenemos un módulo de entradas y salidas, que presenta entradas y salidas de forma independiente, es decir bit a bit (figura 12)



**Figura 12:** Modulo de Entradas Salidas