

# Apéndice B

## VISUALIZACIONES EN CoDeSys

---

La herramienta de visualización en CoDeSys permite la monitorización de datos y la asignación de valores a variables desde el teclado o el ratón, utilizando para ello un entorno gráfico. El editor de visualización está integrado en el entorno de programación de CoDeSys, resultando una herramienta potente y cómoda para monitorizar el comportamiento de un programa en la fase de depuración, o incluso como un SCADA simple en la fase de ejecución usando las herramientas más avanzadas de las que dispone. En este libro, como ha podido ver el lector, se ha usado profusamente esta herramienta en la práctica totalidad de los ejemplos y ejercicios propuestos.

Este apéndice pretende dar unas nociones elementales del uso de las visualizaciones, utilización de los elementos básicos, conexión con las variables de programa de control, etc., pero con la suficiente profundidad que permita al usuario establecer un entorno de monitorización y mando básico para los programas que esté desarrollando. Para una profundización en esta herramienta consulte [13].

### B.1. Editor de Visualizaciones

La edición de visualizaciones está basada en la colocación de elementos gráficos en una ventana de edición, cuyas características (forma, color, texto,...) pueden estar asociadas a las variables del programa del PLC que se ha realizado en CoDeSys. El editor permite sólo unos pocos tipos de elementos, pero suficientes para proporcionar una gran funcionalidad desde una notable simplicidad. Una de las características que tiene es la facilidad en la conexión de los elementos gráficos con las variables del programa del PLC, que no es habitual en otras implementaciones de este tipo. En definitiva, permite la realización de una aceptable interface de

usuario con un esfuerzo de programación relativamente pequeño.

La creación de una Visualización en un Proyecto, se realiza desde la pestaña *Visualizaciones* en el Organizador de Objetos de CoDeSys. Cada proyecto puede tener tantas visualizaciones como se desee.

Para crear una visualización se selecciona la pestaña de visualizaciones, y con el menú contextual (botón derecho del ratón) se selecciona *Add Object*. El único parámetro que es necesario introducir es el nombre de la visualización. Aparecerá una ventana gráfica donde se podrán introducir los distintos elementos. Nótese que la ventana aparecerá en la zona de edición de CoDeSys, donde están las ventanas con el código de los POU's, etc. Es decir se trata de una herramienta totalmente integrada en el desarrollo de programas para PLC y con la que se podrá trabajar de forma simultánea al desarrollo de dichos programas. Con este mismo menú contextual se pueden realizar otras operaciones con la visualizaciones, como borrarlas, copiarlas, etc.

Si se selecciona la ventana de visualización aparecerán en la barra de herramientas los botones con los que se podrán insertar los distintos elementos gráficos, que están listados en la Tabla B-1.

Para insertar cualquiera de estos elementos, se pulsa el botón correspondiente y se coloca en la ventana de edición en la posición y con el tamaño que se desee. La edición es muy similar a la de cualquier programa de edición de gráficos por lo que no se entrará en detalles. En *Extras->ElemenList* aparecerá una tabla con todos los elementos que contenga la visualización y sus coordenadas.

## B.2. Configuración básica de elementos

Haciendo *doble-click* sobre un objeto, se podrá acceder a sus parámetros de configuración, que permiten definir sus propiedades y asociarlos a variables del proceso.

Para hacer referencia a nombres de variables del programa en el PLC, hay que tener en cuenta que se puede acceder tanto a variables globales como locales, por lo que es necesario seguir la siguiente sintaxis:

- **Variables globales:** El formato será *nombre\_var* o *.nombre\_var*.

- **Variables locales:** Si la variable está definida dentro de una POU denominada *MiPOU*, se referencia como *MiPOU.nombre\_var*.

Por ejemplo, si se dispone un temporizador tipo TON denominado *Tempo* definido en una POU con nombre *Control*, la forma de usar el parámetro de entrada IN será:

`Control.Tempo.IN`

Tabla B-1. Elementos gráficos en las Visualizaciones

Elementos gráficos	Botón	Elementos gráficos	Botón
Rectángulo		Rectángulo con borde redondeado	
Elipse		Polígono	
Polilínea		Curva	
Gráfico de tarta		Gráfico bitmap	
Visualizaciones		Botones	
Gráficos WMF		Tablas	
Gráficos de tendencia		Tablas de alarmas	
Elementos Active-X		Barra de desplazamiento	
Indicador de aguja		Diagrama de barras	
Histograma			

**Nota:** Es necesario tener en cuenta que las visualizaciones no se compilan como el resto del programa, por tanto, si se escribe mal el nombre de la variable o dicha variable no existe, no se producirá ningún error de compilación ni ningún tipo de aviso, simplemente la visualización no se comportará como se desea.

También, en lugar de variables se pueden indicar expresiones y constantes, pero siempre respetando la identificación de variables arriba indicada.

La Figura B-1 muestra una ventana de edición de parámetros. Como se observa, a la izquierda aparecen una serie de categorías de parámetros y a la derecha los parámetros específicos de dicha categoría.

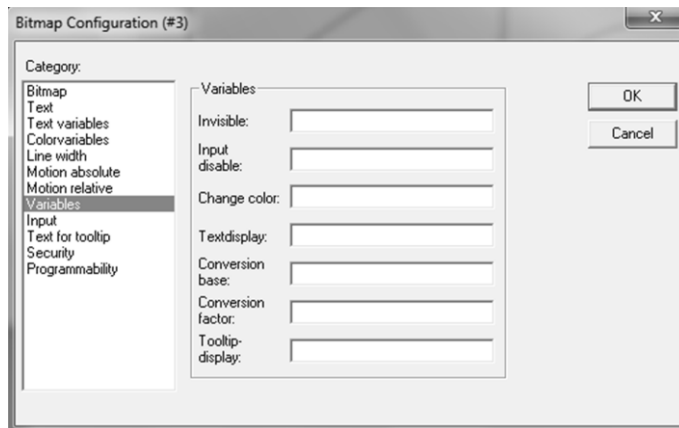


Figura B-1 – Ventana de edición de visualizaciones

Algunas de las categorías son comunes a muchos de los tipos de elementos mientras que otras son específicas de cada elemento. A continuación se describen algunas de las categorías más generales. No se van a describir todos y cada uno de los parámetros sino sólo los más útiles para un uso básico. Para más detalle, consulte el manual de usuario.

### B.2.1. Text

Permite incluir texto que se mostrará en el interior del elemento. Además, permite determinar la alineación del texto y la elección de las fuentes. También es posible

hacer que aparezca en el texto el valor de una determinada variable. Para ello se utiliza una sintaxis similar a la que usa el lenguaje C en la función *printf*. De forma general se utilizará %s en el lugar donde se quiera que aparezca el valor de la variable. La variable a mostrar se indicará en la Categoría *Variables* -> *TextDisplay*. En general se puede mostrar cualquier tipo de variable, aunque también se permite utilizar formatos específicos para tipos de datos concretos, por ejemplo %d para enteros, %f para reales, etc. (en el botón de ayuda de la ventana se encuentra una información más detallada de todos los posibles formatos).

### B.2.2. Text variables

Aunque dentro de la ventana *Text* es posible fijar las características del texto (color, fuente,...), en general éstas permanecen estáticas a lo largo del tiempo. Si se pretende que alguna de estas características cambie dinámicamente, se pueden asignar estos parámetros a nombres de variables. Algunos parámetros que se pueden modificar son:

- **Textcolor:** El color del texto.
- **Textflags:** Alineación del texto (centrado, izquierda,...).
- **Fontheight:** Tamaño de letra.
- **Fontname:** Nombre de la fuente.
- **Fontflags:** Negrita, subrayado, itálica,...

### Ejemplo

Se ha escrito una POU en lenguaje ST en el fichero *CapB\_1.pro* denominado *CambiaLetra*. En el campo *Textcolor* introducimos *CambiaLetra.ColorLetra* y en el campo *Fontheight* la variable *CambiaLetra.AlturaLetra*. El código del programa se muestra en el Ejemplo B-1.

Compruebe en el botón que aparece a la izquierda de la visualización, que dependiendo de si *M* está a TRUE o a FALSE (pulsando el botón) cambian el color y el tamaño de la letra. El color se introduce en una variable DWORD en formato RGB (dos dígitos para cada color).

**Ejemplo B-1. Código para cambio de tipo de letra en Visualizaciones**

---

```
AlturaLetra:INT;
ColorLetra: DWORD;
-----
IF M THEN
    AlturaLetra:=16;
    ColorLetra:=16#FF00FF;
ELSE
    AlturaLetra:=22;
    ColorLetra:=16#121212;
END_IF
```

---

**B.2.3. Line width**

En esta categoría se puede seleccionar el grosor de la línea de la figura indicando el número de píxeles o bien indicar un nombre de variable si se quiere que cambie a lo largo de la simulación.

**B.2.4. Color**

Permite seleccionar el color del interior del objeto (rectángulos, círculos...) y de la línea del marco. Marcando las casillas correspondientes se puede eliminar el color del interior o del marco. A cada objeto se le pueden asignar dos colores, tanto al interior como al marco, denominados *Color* y *Color Alarma*.

El objeto cambia el color entre estos dos dependiendo del valor de la variable booleana que se introduce en la categoría *Variables -> Change Color*.

En *CapB\_1.pro* se muestra como usando *Color* y *Color Alarma* puede cambiar el color de la figura. La variable de cambio es *Alarma*, cuyo valor se puede cambiar pulsando el interruptor de alarma.

**B.2.5. Colorvariables**

*Color* y *Color Alarma* son valores fijos si lo definimos en la categoría anterior. En esta categoría se pueden asignar a variables, por lo que todos esos colores pueden

cambiar a lo largo de la ejecución del programa sin más que cambiar el valor de las variables. Como se indicó anteriormente, las variables de color se definen del tipo `DWORD` y se expresan en formato RGB.

### B.2.6. Motion absolute

Permite el movimiento del objeto durante la ejecución del programa. Se permite desplazar el objeto en sentido vertical, horizontal, girarlo o escalarlo. En *CapB\_1.pro* está implementado un pequeño programa en ST que permite mover todos estos parámetros. Nótese que en el giro, la rotación se realiza alrededor del punto negro que aparece cuando se selecciona el objeto. El objeto es un rectángulo, pero nótese que los lados no giran sino el objeto completo. Si se quiere un giro del objeto se deberá seleccionar necesariamente un objeto de tipo *polígono*.

### B.2.7. Motion relative

Permite mover los límites laterales del objeto. En *CapB\_2.pro* se puede ver un ejemplo de movimiento de los laterales de una elipse. En la categoría *Shape* de los parámetros de la elipse se puede cambiar el tipo de objeto a un rectángulo, línea, etc., para ver el efecto sobre otras formas geométricas.

### B.2.8. Variables

En las características de un objeto se introducen variables que realizan algunas acciones sobre él, algunas de las cuales ya se han visto en la descripción de categorías anteriores.

- **Change Color:** Cambia entre *Color* y *Color Alarma*.
- **TextDisplay:** Valor de la variable que se representa en el texto cuando incluye %s.
- **Invisible:** El objeto desaparece cuando la variable está a TRUE. Compruebe su efecto en *CapB\_3.pro*.
- **Disable Input:** Deshabilita todo lo que esté en la Categoría *Input*. Por ejemplo,

en *CapB\_3.pro* el interruptor *Deshabilita botones* activa la variable *Deshabilita* que está puesta en *Variables->Disable Input* de los dos botones que se encuentran a su izquierda. Se comprueba que los botones dejan de funcionar cuando se activa *Deshabilita botones*.

### B.2.9. Input

Es la forma más simple de dar valores a una variable booleana del programa del PLC. Cuando pulsamos con el ratón sobre el objeto, cambia el valor de la variable que tiene asignada. Este es el uso más típico del objeto *botón*, pero se puede hacer con cualquier otro objeto.

- **Toggle Variable:** Funcionamiento como interruptor. Cada vez que pulsemos el objeto cambia el estado de la variable asociada. Por ejemplo, compruebe los parámetros y el funcionamiento del objeto *Desactiva Botones* en *CapB\_3.pro*.
- **Tap Variable:** Funcionamiento como pulsador. La variable asociada solo cambia mientras esté pulsado el elemento. Por ejemplo, analice el objeto *Invisible* en *CapB\_3.pro*.
- **Tap False:** Determina el estado inicial del pulsador.
- **Text Input of Variable TextDisplay:** Opción muy útil para introducir valores a una variable no booleana. Se introduce el valor a la variable definida en la categoría *Variables->Textdisplay*. Hay varias opciones que permiten introducir valores por teclado, con un teclado en pantalla, con los valores ocultos, etc. En *CapB\_3.pro* se muestra un ejemplo, en el que al pulsar el círculo se pueden introducir valores, mediante un teclado numérico en pantalla, que se asignan a la variable *Numero*, cuyo valor se muestra en una ventana.

### B.2.10. Text for tooltip

Texto que aparece cuando pasa el ratón por encima del objeto cuando está en modo ejecución. En *CapB\_3.pro* cuando se pasa el ratón por encima del círculo amarillo, aparece un texto.



## B.3. Configuración avanzada de elementos

En la sección anterior se describieron algunos de los atributos más comunes compartidos por los distintos elementos. Los elementos más comunes (rectángulos, círculos, líneas, botones, etc.) prácticamente se definen con los parámetros arriba indicados. En esta sección se verán algunos elementos gráficos más específicos y que tienen atributos distintos, aunque también pueden compartir alguno de los atributos anteriores.

### B.3.1. Table

Se utiliza para la representación de matrices. El ejemplo *CapB\_4.pro* muestra un ejemplo de representación de una tabla declarada y usada en el programa PLG\_PLC escrito en código ST. Cuando se pulsa el botón *Duplica* los valores de la tabla se duplican.

Las cuatro primeras categorías son específicas para tablas:

- **Table:** Define la tabla a representar, que obligatoriamente tiene que estar declarada en algún programa.
- **Columns:** Se indica las columnas que se quieren representar. En la zona izquierda están todas las columnas y a la izquierda las seleccionadas para ser representadas. Si se hace *doble-click* en una columna de las seleccionadas a la derecha, se puede cambiar el nombre del encabezamiento de la columna, el ancho de cada columna, etc.
- **Rows:** Permite configurar el alto de cada fila.
- **Selection:** Sirve para determinar cómo se van a marcar las celdas seleccionadas con el ratón: color, recuadro, celdas, columnas o filas. También se pueden marcar las celdas determinadas por variables de los programas.

### B.3.2. Scrollbar

La configuración de las barras de desplazamiento no tiene ningún problema especial. Estudie el comportamiento de la barra de desplazamiento en *CapB\_4.pro*, donde la selección de una celda de la tabla se realiza desde dos barras de

desplazamiento.

### **B.3.3. Indicador de aguja**

El indicador de aguja (*Meter*) presenta muchas opciones de diseño. En cualquier caso, en la propia ventana de configuración tiene una pre-visualización, por lo que el diseño se realiza de una forma muy simple. En *CapB\_4.pro* la variable de selección *Y* está indicada con uno de estos elementos gráficos. Modifique los parámetros de diseño y observe el resultado.

### **B.3.4. Diagrama de barras**

Al igual que en el caso anterior, tiene varias opciones de configuración, pero al disponer de una pre-visualización es fácil de usar. En *CapB\_4.pro* el valor de la variable *SeleccionX* está representada por un diagrama de este tipo.

### **B.3.5. Inclusión de archivos gráficos**

Se pueden introducir en una visualización dibujos o fotografías en distintos formatos. Para esto hay dos posibilidades: el elemento *bitmap* permite introducir archivos en formato *bmp*, *tif* o *jpg*, mientras que el elemento *WMF* admite el formato *WMF* (*Windows MetaFile*). Los parámetros de configuración son los típicos que se vieron en la sección anterior.

### **B.3.6. Graficas de variables frente al tiempo**

Permite representar una o más variables frente al tiempo. De nuevo presenta muchas opciones, pero lo más recomendable es experimentar con él. Para esto, se proporciona en *CapB\_5.pro* un simulador de un depósito escrito en ST. Pulsando los botones de *Entrada* y *Salida* se abren las válvulas de entrada y salida respectivamente. En la gráfica está representado el nivel del depósito. De nuevo, practique con las distintas opciones que presenta este elemento para entender su funcionamiento.

Como última actividad de este libro, preste atención al simulador escrito en ST. Realiza una integración de la ecuación diferencial simple del depósito utilizando el método de Euler. En cada ciclo del autómata realiza un paso de integración de 0.1 segundos. Como el programa está incluido en una tarea cíclica que se ejecuta cada 100 ms., el comportamiento del simulador lo podemos considerar en tiempo real.

