

Desarrollo de un proyecto de monitorización con WinCC

1	OBJETIVO.....	2
1.1	<i>Adquisición de datos del ordenador de proceso.....</i>	2
1.2	<i>Visualización y almacenamiento de datos.....</i>	2
1.3	<i>Avisos y alarmas.....</i>	2
1.4	<i>Finalización.....</i>	2
2	CONFIGURACIÓN DEL PROYECTO	2
2.1	<i>Computer (ordenador).....</i>	3
2.2	<i>Tag Management (Gestión de Variables).....</i>	5
2.3	<i>Structure Tag (Variable de tipo estructura).....</i>	7
2.4	<i>Graphics Designer (Editor gráfico).....</i>	7
2.5	<i>Alarm Logging (Registro de Alarmas).....</i>	9
2.6	<i>Tag Logging.....</i>	9
2.7	<i>Report Designer.....</i>	10
2.8	<i>Global Script (Guiones Globales).....</i>	10
2.9	<i>Text Library (Biblioteca de texto).....</i>	10
2.10	<i>User Administration (Gestión de usuarios).....</i>	10
2.11	<i>CrossReference (Referencias cruzadas).....</i>	10
2.12	<i>Timesynchronization (Sincronización temporal).....</i>	10
2.13	<i>Picture Tree Manager (Gestor del árbol de imágenes).....</i>	11
2.14	<i>Lifebeat Monitoring (Monitorización de signos de vida).....</i>	11
2.15	<i>Base Data.....</i>	11
2.16	<i>User Archive.....</i>	11
3	EDITORES.....	12
3.1	<i>Editor gráfico.....</i>	12
3.1.1	<i>Ret_Cast.....</i>	13
3.1.2	<i>Main.....</i>	16
3.2	<i>Editor de alarmas.....</i>	20
3.3	<i>Editor de archivos.....</i>	26
3.4	<i>Editor de guiones.....</i>	29
3.4.1	<i>IsSteelGrade().....</i>	29
3.4.2	<i>IsPowderType().....</i>	30
3.4.3	<i>ShowAlarms().....</i>	30
3.5	<i>Editor de archivos de usuario.....</i>	31
4	PRUEBAS.....	34

1 **Objetivo**

Con el desarrollo de este proyecto se pretende comprobar si el software WinCC permite llevar a cabo la monitorización descrita a continuación sobre el molde de la colada continua.

1.1 **Adquisición de datos del ordenador de proceso**

Para identificar la colada a la que pertenecen los datos, deben tomarse del ordenador de proceso la siguiente información: tipo de acero, tipo de polvos que se utilizan, caudal y temperaturas de entrada y salida del agua de refrigeración.

A la espera de conocer el método adecuado para acceder a estos datos en el ordenador de proceso, se ha implementado su adquisición mediante una entrada directa por teclado.

1.2 **Visualización y almacenamiento de datos**

Las medidas de los 18 termopares se visualizan en tiempo real (frecuencia de actualización = 2Hz) sobre un gráfico que indica su situación en el molde. Al mismo tiempo, estos valores se registran en una base de datos indicando fecha y hora de adquisición.

Adicionalmente se puede ver una gráfica que representa la evolución de la temperatura medida por el termopar, para cada uno de ellos.

1.3 **Avisos y alarmas**

Para cada termopar se define una temperatura límite, por encima de la cual se presenta en pantalla un aviso indicando cuál es el termopar en el que se origina la alarma.

1.4 **Finalización**

En planta no se necesitaría desactivar el sistema, sino sólo dejar de registrar datos durante las paradas y actualizar los datos de colada; para esta comprobación, sin embargo, se habilita un pulsador que termina la monitorización.

2 **Configuración del proyecto**

Una vez creado el nuevo proyecto, nos encontramos ante el explorador de WinCC (Fig. 1). En la parte izquierda de la pantalla aparecen los distintos componentes del proyecto que se configuran como se describirá en los próximos apartados.

Aunque aquí se describe la configuración de cada componente de manera secuencial, en el proceso de desarrollo se alterna de uno a otro en función de las necesidades del momento. Por ejemplo, una vez ya en el editor gráfico puede ser necesario volver al gestor de variables (*Tag Manager*) para crear una nueva variable necesaria para el funcionamiento de un objeto que se inserte en el gráfico.

En la definición inicial del proyecto se indica el tipo de proyecto, el dispositivo desde el que se va a ejecutar y su resolución, así como el protocolo necesario para comunicarse con el autómat. El caso que nos ocupa es un proyecto de un solo equipo, para PC con resolución 800x600 y el protocolo es, por el momento, indiferente, ya que no se va a conectar con ningún PLC.

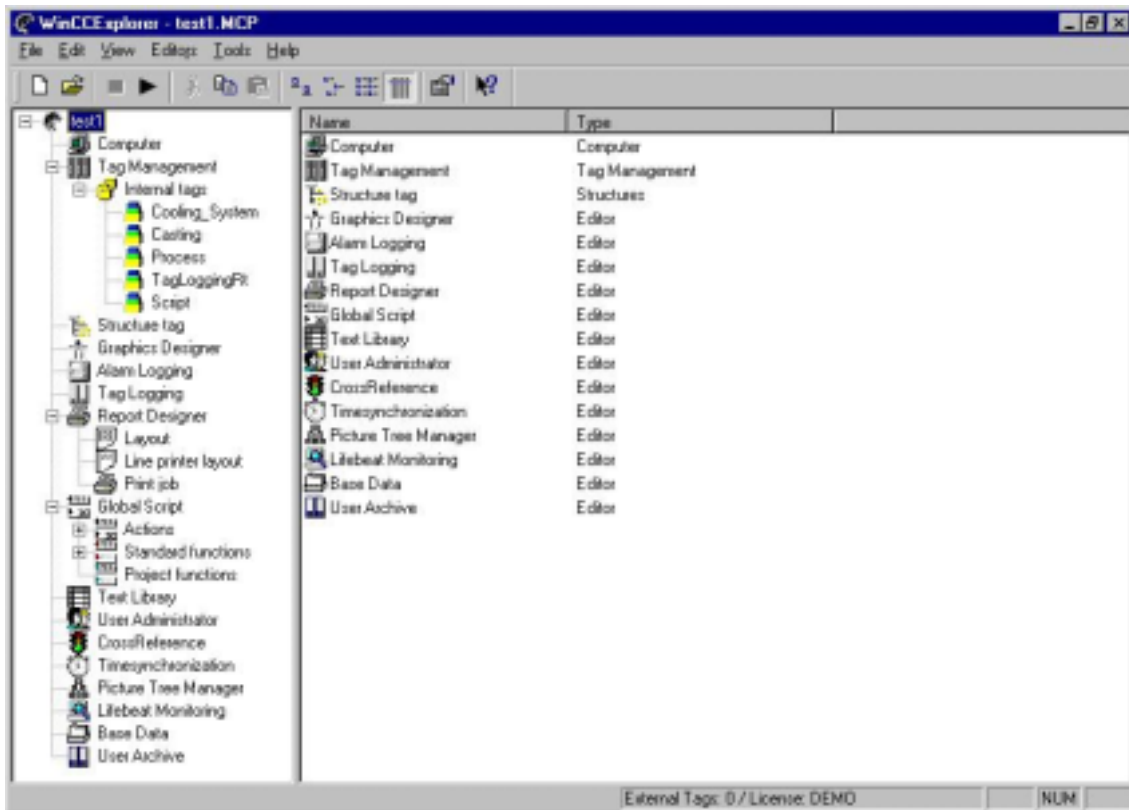


Fig. 1

2.1 Computer (ordenador)

En esta sección se listan los ordenadores que tomarán parte en el proyecto, indicando por cada uno si es servidor o cliente. En nuestro caso habrá un único ordenador (Fig. 2).

Haciendo doble clic sobre él podemos definir las propiedades en 4 pantallas:

1. **General** (general)

Aquí se define el nombre del ordenador y si funciona como servidor, cliente o multicliente. Si sólo hay un ordenador sólo se admite la opción servidor (Fig. 3).

2. **Startup** (inicio)

En este apartado se seleccionan los componentes que se iniciarán al arrancar el *runtime* (modo de ejecución). Para nuestro proyecto se han seleccionado (Fig. 4):

- *Text Library Runtime* (Biblioteca de texto), necesario para utilizar el registro de alarmas.
- *Global Script Runtime* (Guiones globales), para permitir la ejecución de acciones programadas como funciones en C (*scripts*, guiones).

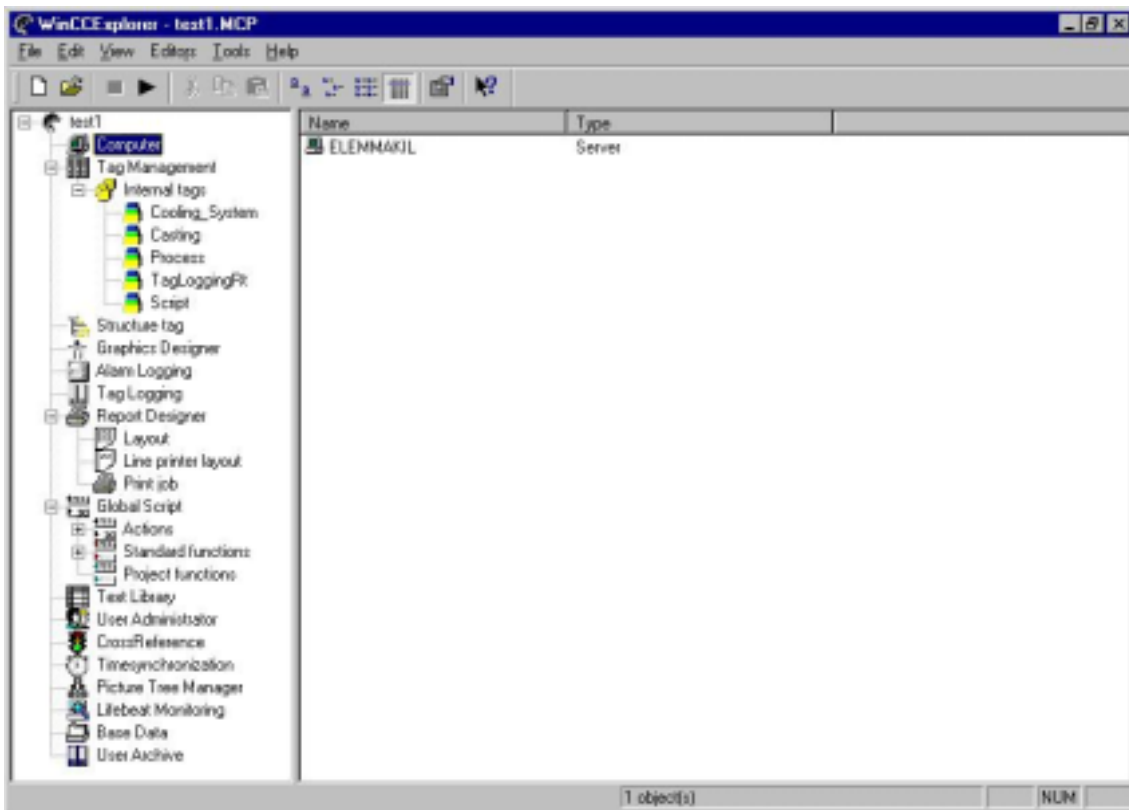


Fig. 2

- *Alarm Logging Runtime* (Registro de alarmas), para el proceso de los avisos por temperatura de los termopares.
- *Tag Logging Runtime* (Registro de variables), para almacenar los valores de las temperaturas para su representación gráfica.
- *Graphics Runtime* (Gráficos), para gestión de ventanas.

Además se ha añadido el módulo CCUsrAcv.exe (*User Archives*, archivos de usuario) para la gestión de bases de datos.

3. Parameters (parámetros)

Permite seleccionar el idioma para el modo de ejecución y desactivar mientras el proyecto está activo las combinaciones de teclas que permiten el cambio de aplicación. Para la instalación en planta deberían desactivarse todas, de modo que no se pueda abandonar la monitorización (Fig. 5).

4. Graphics Runtime (opciones para el motor gráfico)

En esta sección se definen las propiedades de las ventanas del proyecto (Fig. 6).



Fig. 3



Fig. 4

Se designa una imagen de inicio (**start picture**), que será la que se cargue al activar el proyecto. Nuestra imagen de inicio es *Ret_Cast.Pdl* (v. Graphics).

En la lista **Window Attributes** (Atributos de ventana) se seleccionan las características deseadas para las ventanas. Se han seleccionado **Full Screen** (pantalla completa) y **Adapt Picture** (adaptar imagen) para asegurarse de que la imagen ocupe exactamente la pantalla, y se han deshabilitado las demás en la lista **Turn Off** (deshabilitar) para impedir que se pueda modificar esto (i.e. minimizar, cambiar de tamaño, etc.), y que se pueda cerrar la aplicación de una manera que no sea con el pulsador de finalizar.

En el resto de campos se pueden definir acciones para teclas y la navegación entre imágenes, pero ninguna de estas opciones ha sido necesaria en el proyecto que nos ocupa.

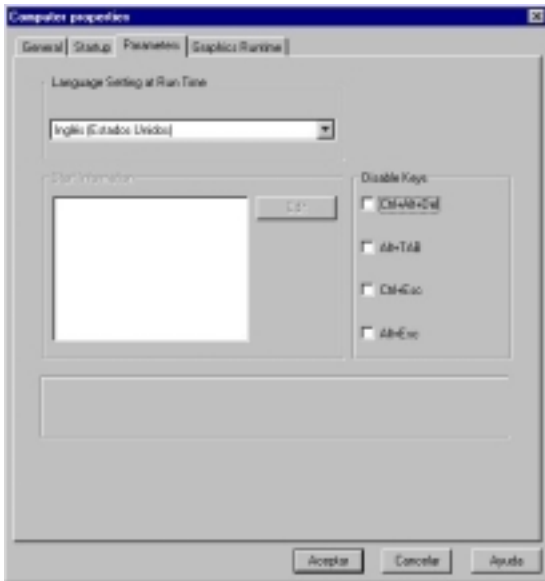


Fig. 5



Fig. 6

2.2 Tag Management (Gestión de Variables)

Este componente permite crear, estructurar y manipular las variables (*tags*) que se utilizarán. La estructura de variables es en forma de árbol de directorios de un nivel, lo que permite agruparlas por relaciones funcionales o semánticas (Fig. 7).

Aparte de las variables generadas por el propio programa, se han definido las siguientes:

Nombre	Uso	Tipo	Valor inicial	Límites	Ubicación
MsgTxt	Cadena de texto para la pantalla de salida	Text Tag 8-bit character set	"_"	N/A	Internal tags

Nombre	Uso	Tipo	Valor inicial	Límites	Ubicación				
NTermopar	Indica para qué termopar se ha hecho la llamada a la representación gráfica en el formato archivovvariable	Text Tag 8-bit character set	N/A	N/A	Internal tags				
AlarmRcvd	Variable en la que se indica la existencia de un mensaje de alarma	Unsigned 8-bit Value	0	N/A	Internal tags				
Temp_x(x)	Temperatura en cada uno de los termopares, donde x(x) va de 1 a 18	Unsigned 16-bit Value	N/A	N/A	Internal tags\Process				
SteelGrade	Guarda el tipo de acero de la colada	Text Tag 8-bit character set	"_"	N/A	Internal tags\Casting				
PowderType	Guarda el tipo de polvos de lubricación utilizados en la colada	Text Tag 8-bit character set	"_"	N/A	Internal tags\Casting				
Water_Temp_In	Temperatura del agua de refrigeración a la entrada de la camisa	Unsigned 16-bit Value	N/A	N/A	Internal tags\Cooling System				
Water_Temp_Out	Temperatura del agua de refrigeración a la salida de la camisa	Unsigned 16-bit Value	N/A </tr <tr> <td>Water_Rate</td> <td>Caudal de agua de refrigeración</td> <td>Unsigned 16-bit Value</td> <td>N/A</td> <td>N/A</td> <td>Internal tags\Cooling System</td> </tr>	Water_Rate	Caudal de agua de refrigeración	Unsigned 16-bit Value	N/A	N/A	Internal tags\Cooling System
Water_Rate	Caudal de agua de refrigeración	Unsigned 16-bit Value	N/A	N/A	Internal tags\Cooling System				

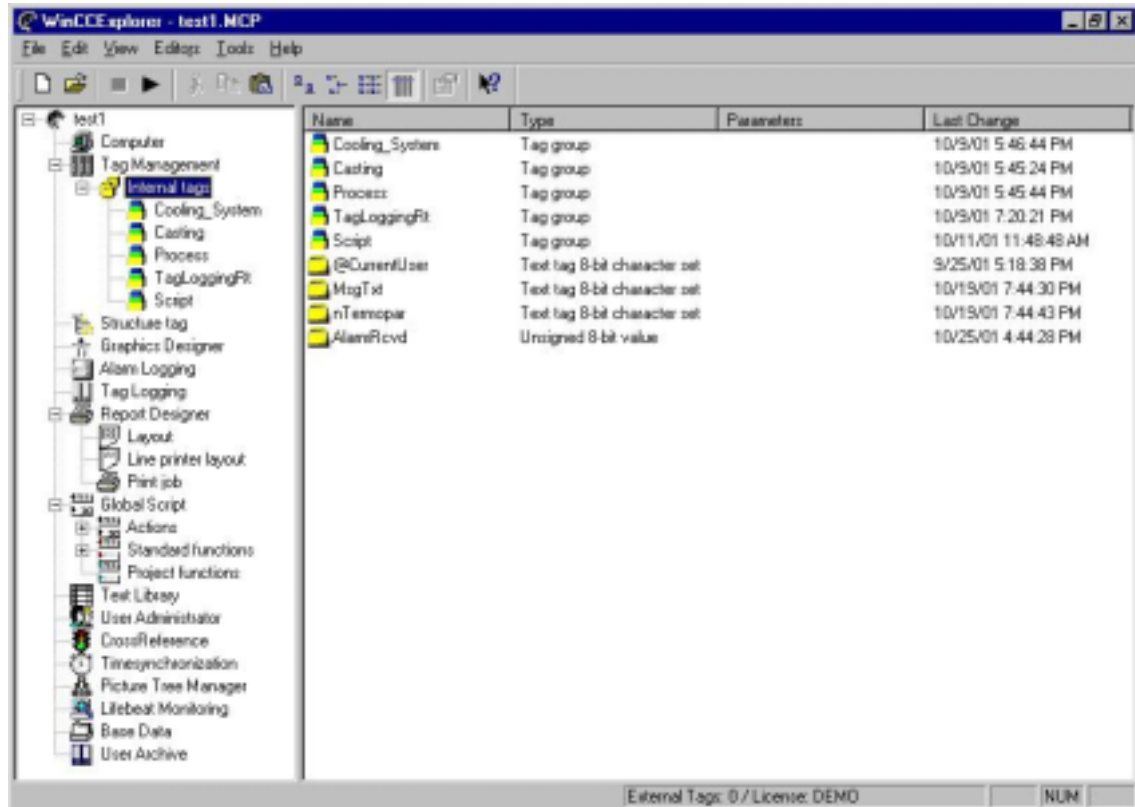


Fig. 7

Las variables se crean pulsando el botón derecho sobre el área de trabajo del gestor de variables (parte derecha de la pantalla) y seleccionando **New Tag** en el menú contextual.

Las carpetas de variables se crean pulsando con el botón derecho sobre **Internal Tags** y seleccionando **New Group** en el menú contextual (Fig. 8).



Fig. 8

En ambos casos se pueden editar las propiedades haciendo doble clic sobre el icono de la variable o el grupo (Fig 9).

De las variables anteriormente descritas, las que se encuentran en *Internal Tags* son propias del programa de control, y no están directamente relacionadas con el proceso. Las que se encuentran en las carpetas son medidas del proceso; estas variables se actualizarían en funcionamiento en planta con los valores contenidos en las variables correspondientes del PLC. Para determinar esta correspondencia, el programa es integrable con STEP7 de Siemens, identificando las variables del proyecto de WinCC con las homónimas del de STEP7.

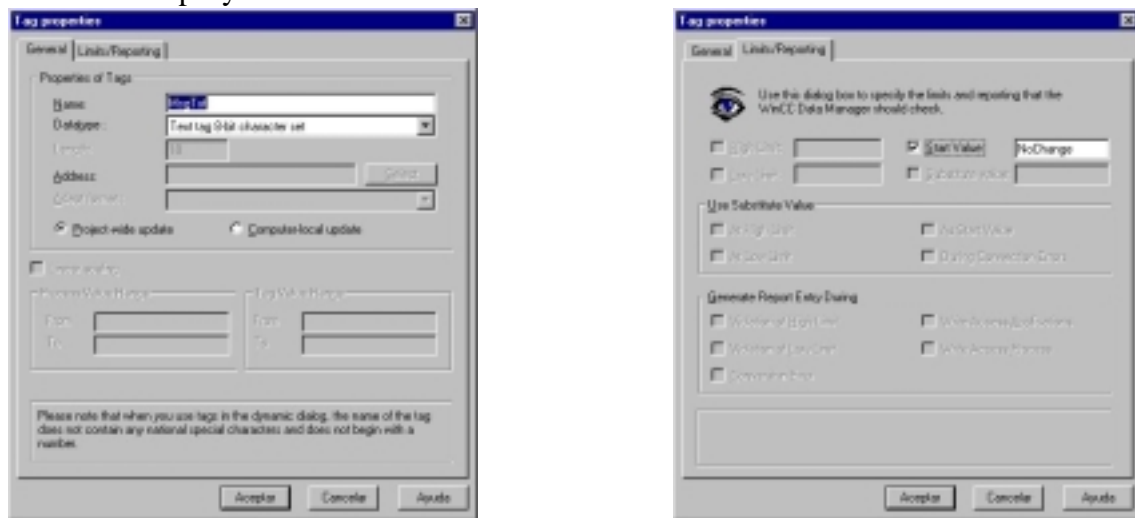


Fig. 9

Otra opción consiste en asignar el *driver* adecuado (pulsar el botón derecho sobre **Tag Management** y seleccionar **New Driver**) y a través de este imponer directamente las direcciones de memoria correspondientes del PLC.

2.3 Structure Tag (Variable de tipo estructura)

Este componente permite diseñar variables compuestas de varias básicas. En el presente proyecto no ha sido necesaria la creación de este tipo de variable.

2.4 Graphics Designer (Editor gráfico)

Con este componente se crean, renombran y eliminan las imágenes. También es posible ver un resumen de las propiedades de cada una de ellas y fijar la imagen de inicio. Todo ello desde el menú contextual que aparece pulsando el botón derecho sobre la imagen considerada (Fig. 10).

Haciendo doble clic sobre el icono de una imagen se abre ésta en el editor gráfico, cuyo funcionamiento se explica más adelante.

Las imágenes que se utilizan en el proyecto se describen a continuación (para información más detallada, véase más abajo el apartado sobre el editor gráfico):

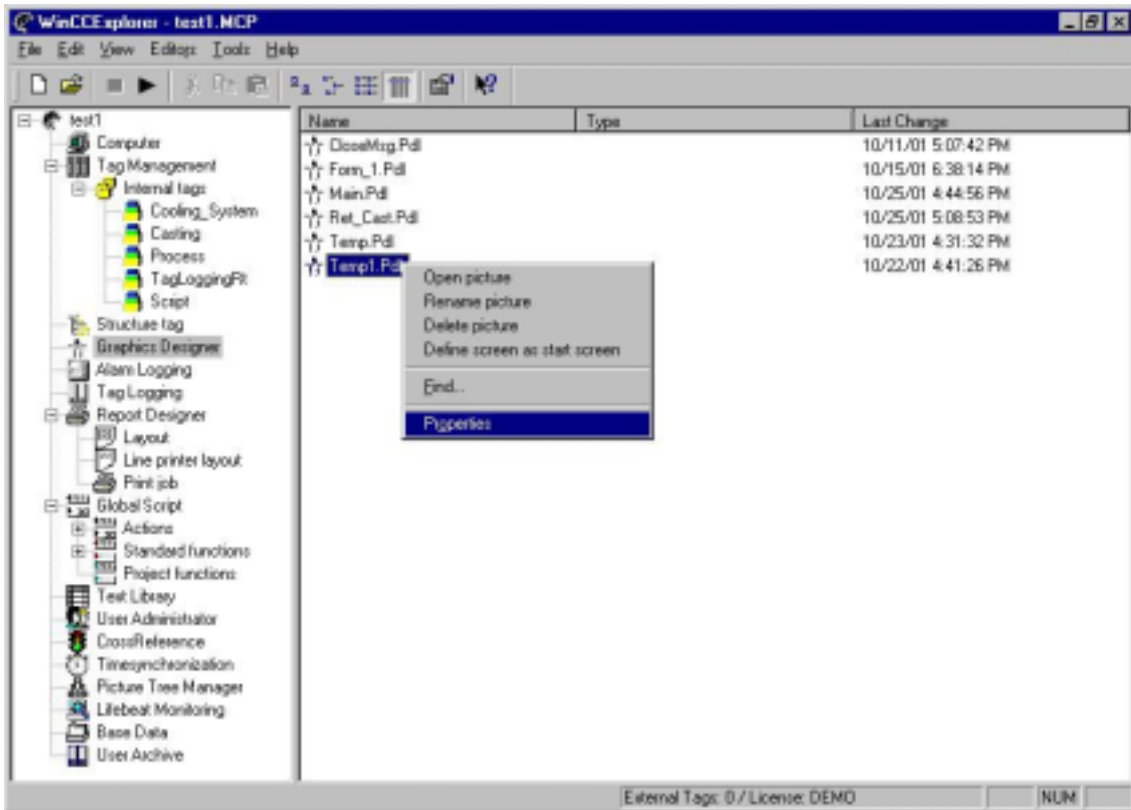


Fig. 10

- *Ret_Cast*: Recoge la información relativa a la colada; tipo de acero y polvos lubricantes, caudal de agua de refrigeración y su temperatura a la entrada y a la salida de la camisa de refrigeración (Fig. 11).

Retrieving Casting Data

Materials

Steel Grade:

Powder Type:

Cooling System

Water Rate:

Water Temperature (Inlet):

Water Temperature (outlet):

Fig. 11

- *Main*: Representa el molde y la posición de los termopares; para cada uno de ellos muestra la temperatura actual y dispone un botón para obtener la gráfica que representa la evolución temporal de la temperatura. En caso de alarma, la tabla de mensajes se superpone hasta que sea cerrada; esta tabla indica en qué termopar se ha sobrepasado la temperatura límite establecida (v. *Alarm Logging*), y la fecha y hora en la que ocurrió. Por último, la ventana *Main* permite desactivar el proyecto mediante la pulsación del botón de finalizar (Figs. 12 y 13).

- *Temp*: Esta ventana contiene una gráfica de evolución temporal de temperatura que muestra los datos correspondientes al termopar desde el que se realiza la llamada, en la ventana *Main*, y un botón para el retorno a dicha ventana (Fig. 14).
- *CloseMsg*: Muestra el mensaje “Closing Runtime. Please wait...” mientras se desactiva el proyecto (Fig. 15).

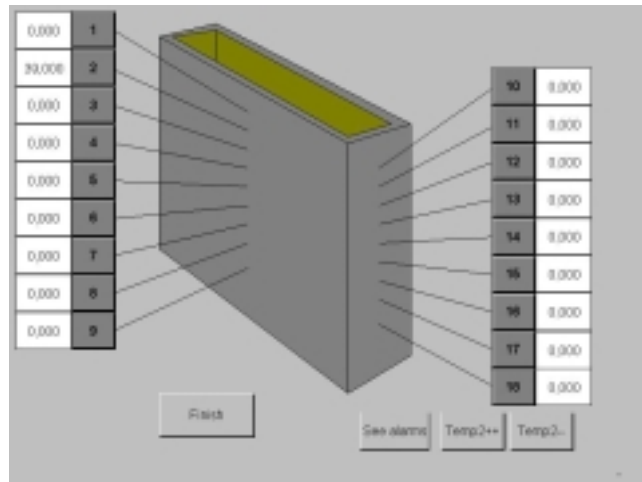


Fig. 12

2.5 Alarm Logging (Registro de Alarmas)

Este componente permite definir mensajes activados por eventos, tales como la superación de determinado valor en una variable, o el estado de un bit en otra.

Pulsando el botón derecho sobre el icono o el área de trabajo se obtiene un menú contextual desde el que se puede abrir el editor de alarmas con el comando *Open* (Fig. 15). La descripción de este editor se realiza más adelante, junto con los demás editores.

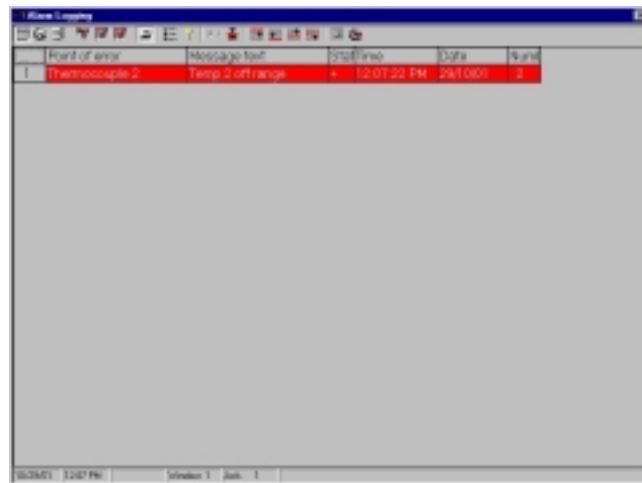


Fig. 13

Con el registro de alarmas se configura la tabla de la fig. 13, que presenta los mensajes de alarma cuando se producen, superponiéndose a la ventana *Main*.

2.6 Tag Logging

Este componente permite crear archivos para el almacenamiento temporal o definitivo de los valores de las variables.

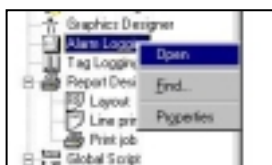


Fig. 15



Fig. 14

En el presente proyecto se ha utilizado un archivo para almacenar las temperaturas de los termopares, muestreadas con una frecuencia de 2Hz.

Pulsando el botón derecho sobre el icono o el área de trabajo se obtiene un menú contextual desde el que se puede abrir el editor de archivos con el comando *Open* (Fig. 16). La descripción de este editor se realiza más adelante, junto con los demás editores.



Fig. 16

2.7 Report Designer

Este componente permite generar modelos de informes, que tomarían los datos de las variables y los archivos, para ser impresos. En el presente proyecto no ha sido necesaria la realización de ningún informe.

2.8 Global Script (Guiones Globales)

Este componente incluye las funciones (guiones, *scripts*) que son globales al proyecto; i.e. pueden ser llamadas desde cualquier script que se realice.

Estas funciones se agrupan en tres conjuntos (Fig. 17):

- *Actions*: scripts para su uso como acciones activadas por eventos. Se distinguen acciones globales (*Global Actions*), visibles desde todos los ordenadores incluidos en el proyecto y locales, accesibles sólo desde el ordenador para el que fueron definidas.
- *Standard Functions*: Funciones estándar provistas por el propio programa relacionadas con el tratamiento de imágenes y objetos, entrada/salida, etc.
- *Project Functions*: Funciones definidas específicamente para el proyecto. En el presente proyecto se han definido tres (**IsSteelGrade()**, **IsPowderType()**, **ShowAlarms()**), que se describirán junto con el editor de funciones.

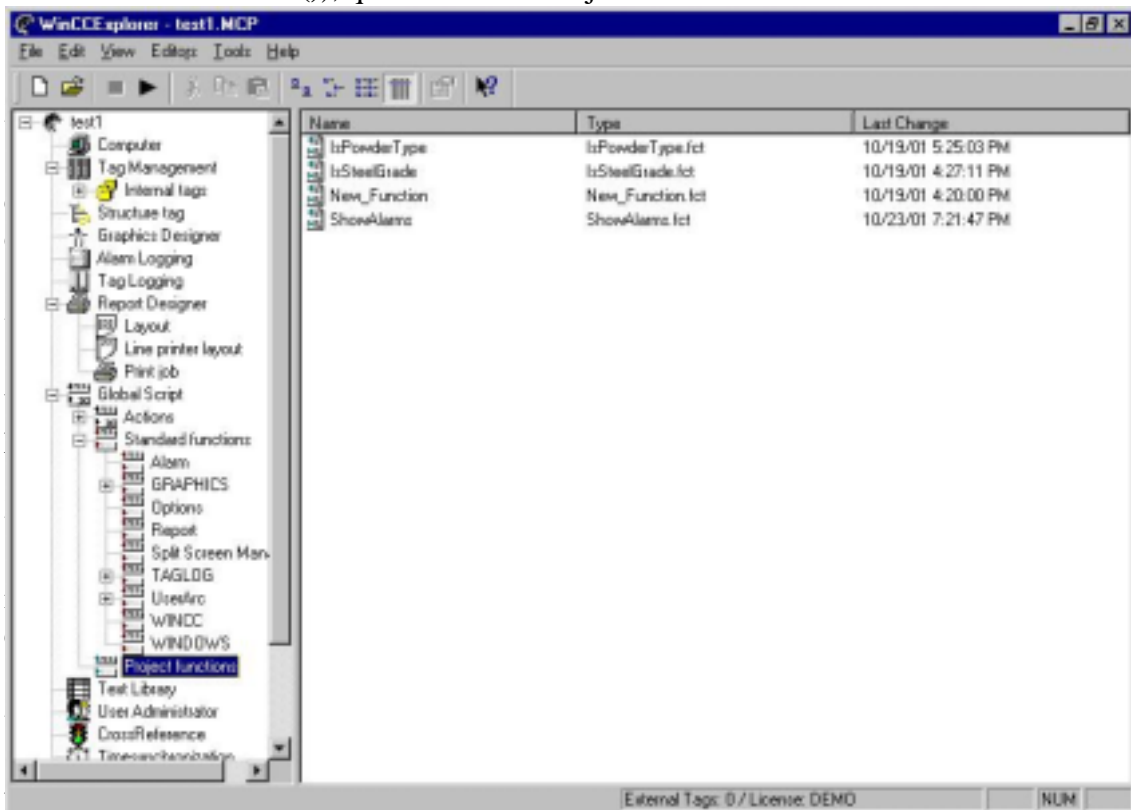


Fig. 17

2.13 Picture Tree Manager (Gestor del árbol de imágenes)

Este componente permite establecer una jerarquía de tipo árbol para las imágenes del proyecto. Dada la reducida complejidad del presente proyecto, no se ha considerado necesario la elaboración de esta estructura.

2.14 Lifebeat Monitoring (Monitorización de signos de vida)

Este componente permite comprobar durante la ejecución el estado del PLC. Al no tener conexión con ningún autómeta, no se ha estudiado el funcionamiento de este módulo.

2.15 Base Data

Este componente contiene dos asistentes: *Split Screen Wizard* y *Alarm Logging Wizard*.

El primero de ellos no está incluido en la licencia, por lo que no se ha podido estudiar su comportamiento.

El segundo es un asistente para la configuración del registro de alarmas. Sin embargo, se ha optado por configurar éste directamente debido a la mayor flexibilidad de ese método.

2.16 User Archive

Este componente permite el almacenamiento de valores de variables en formato de base de datos SQL. En el presente proyecto se ha creado un archivo de base de datos con campos correspondientes a las temperaturas de cada uno de los termopares, así

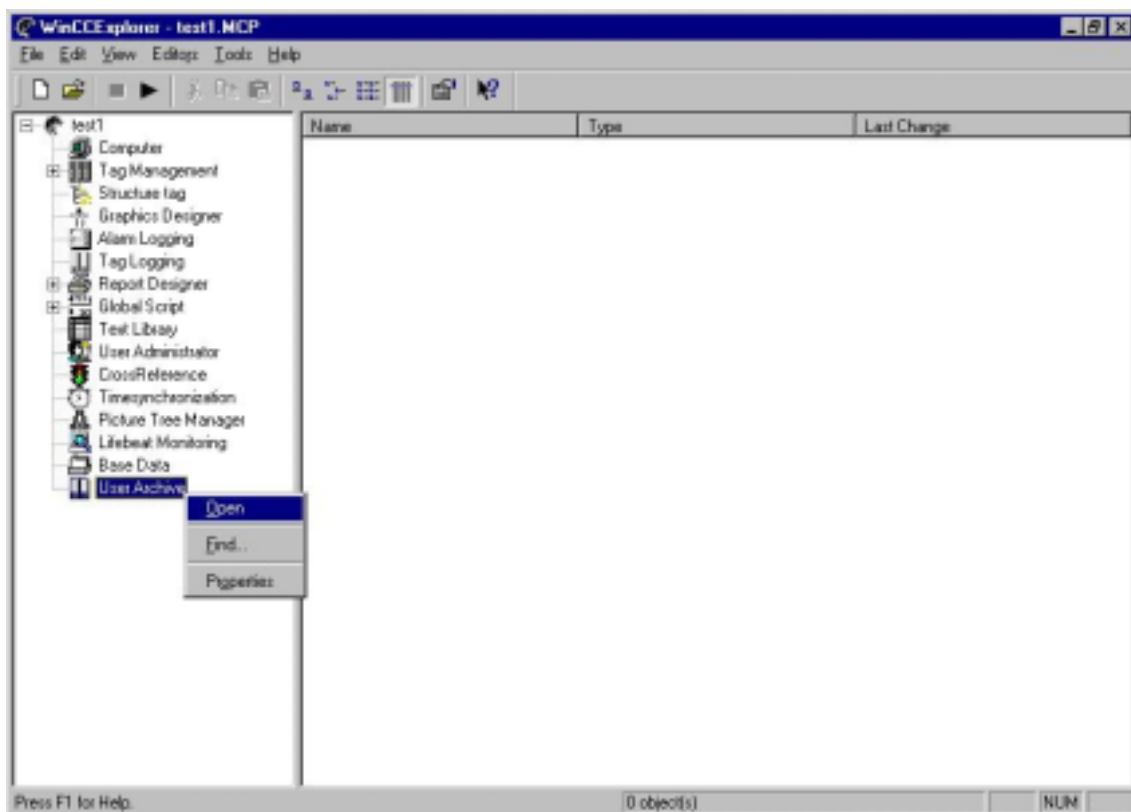


Fig. 18

como de la fecha y hora en que se realiza la medida para sincronización.

Pulsando el botón derecho sobre el icono o el área de trabajo se obtiene un menú contextual desde el que se puede abrir el editor de archivos de usuario con el comando *Open* (Fig. 18). La descripción de este editor se realiza más adelante, junto con los demás editores.

3 Editores

En esta sección se detalla la utilización de cada uno de los editores, así como los elementos del proyecto desarrollados con ellos.

3.1 Editor gráfico

En el editor gráfico se diseña la apariencia y el comportamiento de las pantallas que aparecerán durante la ejecución.

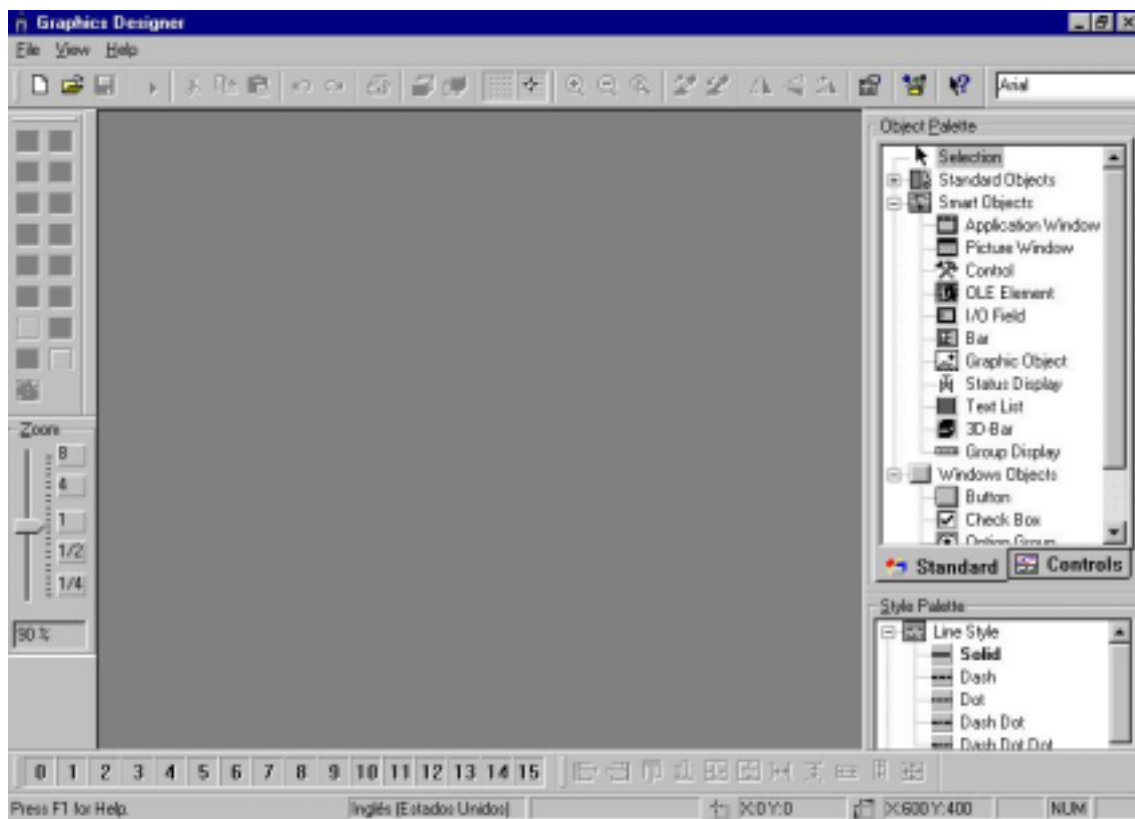


Fig. 19

El aspecto del editor es el que aparece en la fig. 19; alrededor del área de trabajo se sitúan las barras de herramientas, con el control del zoom, paleta de colores e iconos para las tareas más comunes, y las paletas de objetos y estilos de línea.

Inicialmente, al abrir una nueva imagen, aparecerá solamente la base que contendrá a la imagen. Ésta se realiza colocando objetos (líneas, círculos, botones, campos de entrada/salida, etc.) de la paleta de objetos sobre la base, y configurando las propiedades de cada uno de ellos. A todos los efectos la base es un objeto y también pueden ajustarse sus propiedades.

Algunos de los objetos disponen además de un diálogo de configuración que permite configurar las propiedades más importantes de una forma sencilla y rápida.

A continuación se presentan las imágenes que se utilizan en el proyecto con sus correspondientes elementos desglosados y explicados en función de sus propiedades.

3.1.1 Ret_Cast

En primer lugar se ajustan las propiedades de la base para la resolución de pantalla elegida (Fig. 20): indicar el tamaño de 800x600. El resto de propiedades mantienen su valor por defecto. No es necesario adentrarse en la pestaña *Events* (eventos) ya que esta imagen no necesita responder a ningún evento.

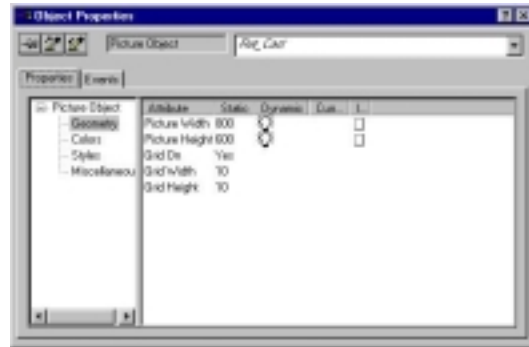


Fig. 20

A continuación se sitúan los recuadros de texto estático (*Static Text*), y se escriben en ellos las cadenas que queremos que presenten (**Captions**). Esto puede hacerse al colocarlos con el ratón, o bien modificando las propiedades en las secciones de geometría (**Geometry**) y fuente (**Font**) (Fig. 21).

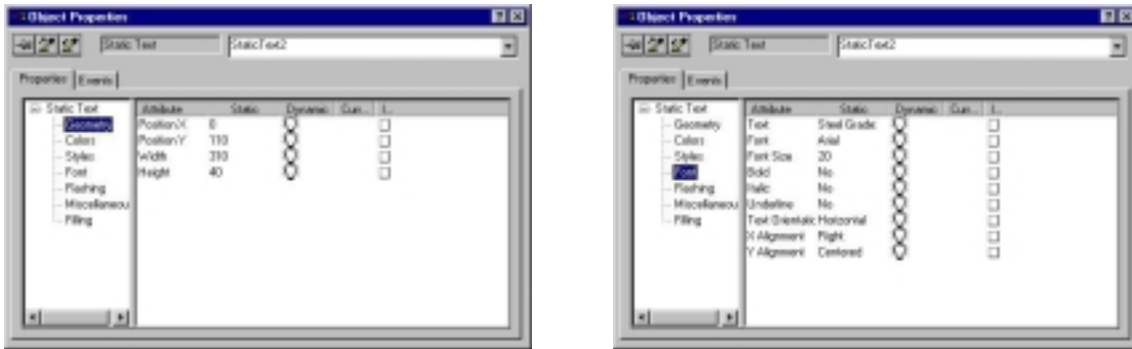


Fig. 21

Una vez situados todos los textos estáticos pondremos los campos de entrada/salida (*I/O Fields*) en los que se introducirán los datos referidos por los textos estáticos. Estos campos de entrada/salida deben configurarse como de entrada (**Field Type: Input**) en la sección **Input/Output** (entrada/salida) de las propiedades, además de la posición y tamaño en la sección **Geometry**, como se muestra en la fig. 22.

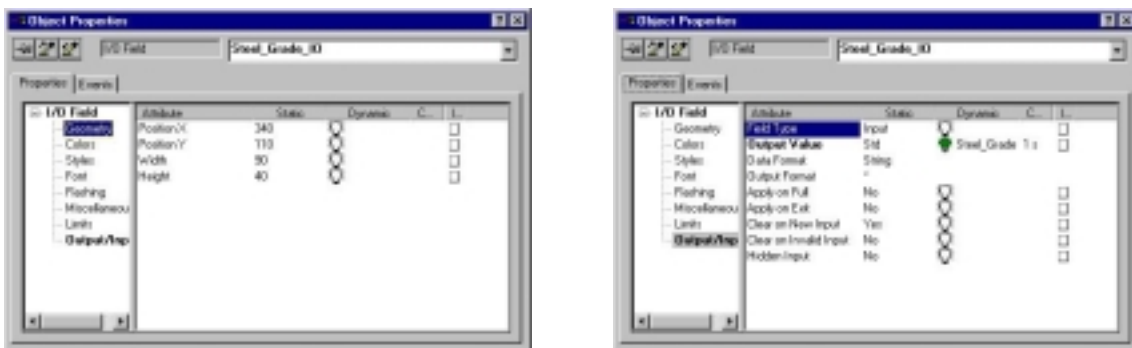


Fig. 22

Los campos para el tipo de acero y polvos se configuran como se ve en la anterior figura para salida de tipo cadena de caracteres (**Data Format: String**), mientras que los

correspondientes al caudal y las temperaturas del agua de refrigeración son de tipo numérico y se configura el formato de salida indicando el número de dígitos y de posiciones decimales, como se ve en la fig. 23. La cadena 999,9 indica que se representarán 4 dígitos, uno de los cuales será decimal.

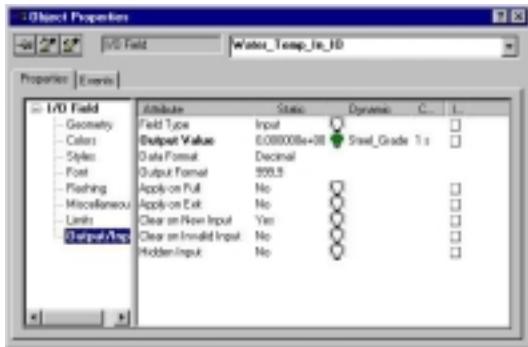


Fig. 23

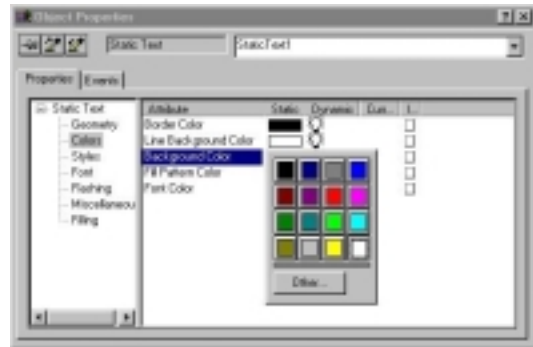


Fig. 24

El título de la ventana es un caso especial de recuadro de texto estático, en el que se han cambiado las propiedades de la sección **Colors** (colores) para variar el color del fondo, como se ve en la Fig. 24. Haciendo doble clic sobre el color correspondiente se presenta la paleta de colores para seleccionar el color deseado.

Por último situamos el botón que valida los datos introducidos y da paso a la ventana *Main*. Se modifican adecuadamente las propiedades de **Geometry** y **Font** como se ve en la fig. 25.

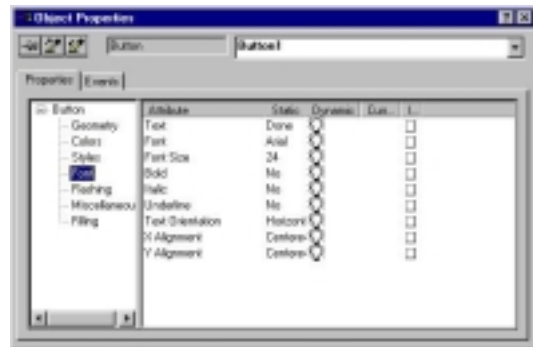
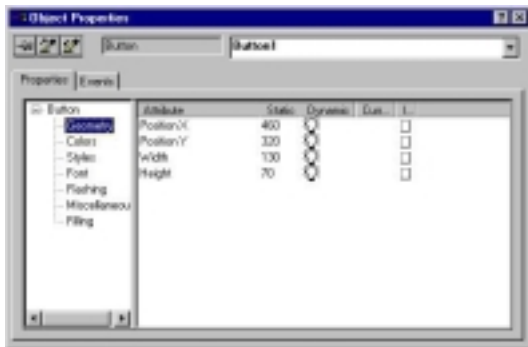


Fig. 25

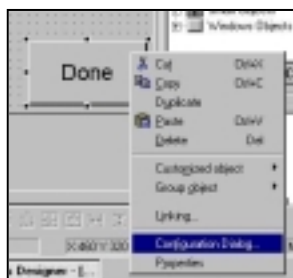


Fig. 26



Fig. 27

Para que el botón abra la ventana *Main* se le indica en el diálogo de configuración (**Configuration Dialog** en el menú contextual que se obtiene pulsando el botón derecho sobre el botón, fig. 26), como se ve en la fig. 27.

Sin embargo no utilizaremos este método, ya que queremos comprobar la validez de los datos introducidos. Para ello debemos recurrir a la activación de acciones mediante eventos. En la sección **Events** del diálogo de propiedades del botón hacemos doble clic sobre la flecha blanca en **Mouse: Mouse Action** y aparecerá la ventana de


```

// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

char inputfield_c[25];
double inputfield_n1, inputfield_n2;

strcpy( inputfield_c, GetInputValueChar( "Ret_Cast", "Steel_Grade_IO" ) );
if ( ! IsSteelGrade( inputfield_c ) ) return;
strcpy( inputfield_c, GetInputValueChar( "Ret_Cast", "Powder_Type_IO" ) );
if ( ! IsPowderType( inputfield_c ) ) return;
inputfield_n1 = GetInputValueDouble( "Ret_Cast", "Water_Rate_IO" ); //Return-Type
:double
if ( inputfield_n1 <= 0 ) return;
inputfield_n1 = GetInputValueDouble( "Ret_Cast", "Water_Temp_In_IO" ); //Return-Type
:double
if ( inputfield_n1 <= 0 ) return;
inputfield_n2 = GetInputValueDouble( "Ret_Cast", "Water_Temp_Out_IO" ); //Return-Type
:double
if ( inputfield_n2 <= inputfield_n1 ) return;
// If this point is reached, all input is correct
SetPropBOOL("AlarmLog.Pdl","Controll","Visible",0); //Return-Type :BOOL
OpenPicture( "Main.Pdl" ); //Return-Type :void
}

```

La última línea indica que se de paso a la ventana *Main*.

3.1.2 Main

Como en el caso anterior, lo primero es definir las propiedades geométricas de la imagen (Fig. 32) para una resolución de 800x600.

Posteriormente se dibuja el molde con la herramienta **Polygon** (polígono) de la sección **Standard Objects** de la paleta de objetos (Fig. 33). Para definir los colores de las líneas y el relleno modificaremos las propiedades de los polígonos. Si van a dibujarse varios polígonos con las mismas propiedades resulta más cómodo cambiar las propiedades en la paleta, lo que afecta a todos los polígonos dibujados a continuación (Fig. 34). Para dibujar el polígono, una vez seleccionada la herramienta se definen los vértices haciendo clic en el lugar correspondiente, y con un doble clic se indica el último, para que el programa cierre el polígono. Los vértices pueden cambiarse

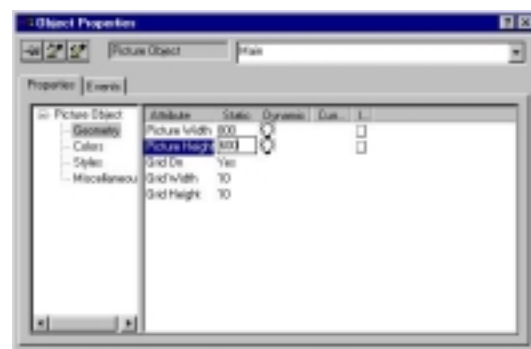


Fig. 32

posteriormente seleccionando el polígono y arrastrando los cuadrados que aparecen en los vértices, para definir la nueva posición de estos.

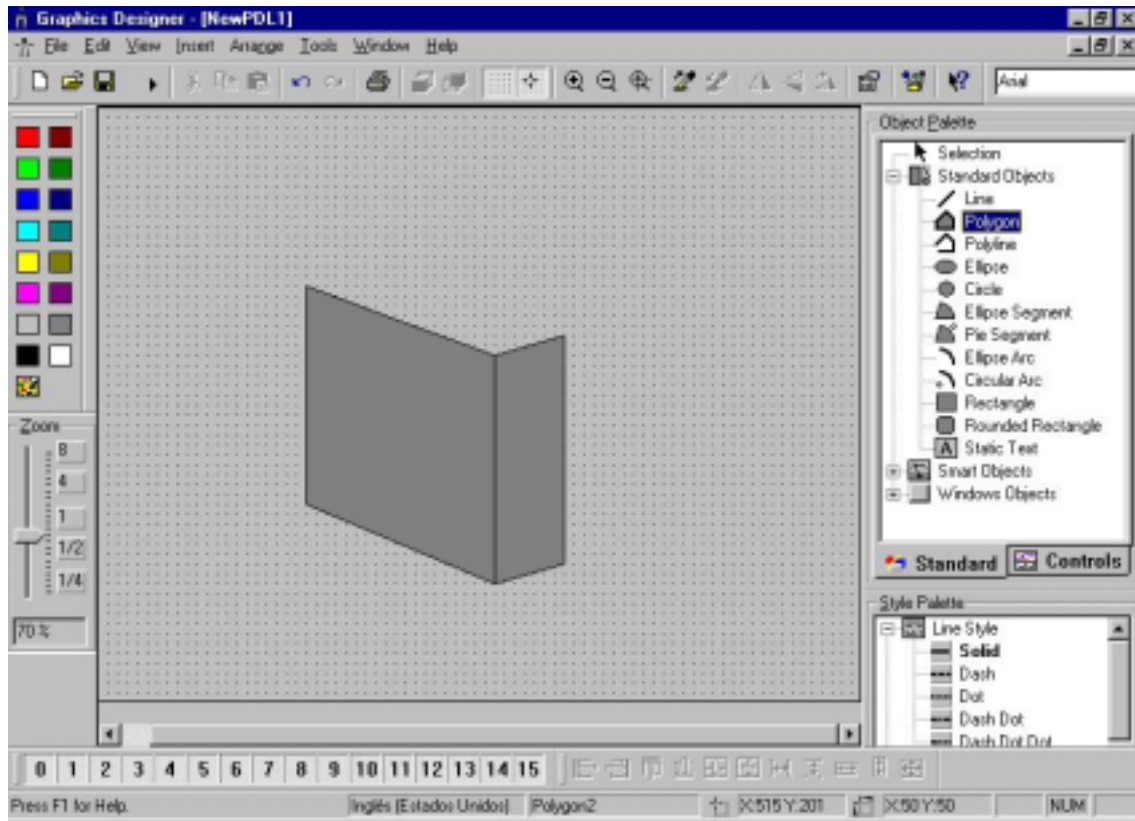


Fig. 33

Después se agregan los botones correspondientes a cada uno de los termopares, así como un campo de entrada/salida por cada uno de ellos y las líneas que los unen con los emplazamientos correspondientes en el molde. También se incluye el botón de finalización (*Finish*).

Los demás botones que se ven en la ventana sirven para propósitos de testeo y pruebas y serán descritos en la sección correspondiente; el resto de los elementos se describe detalladamente a continuación.

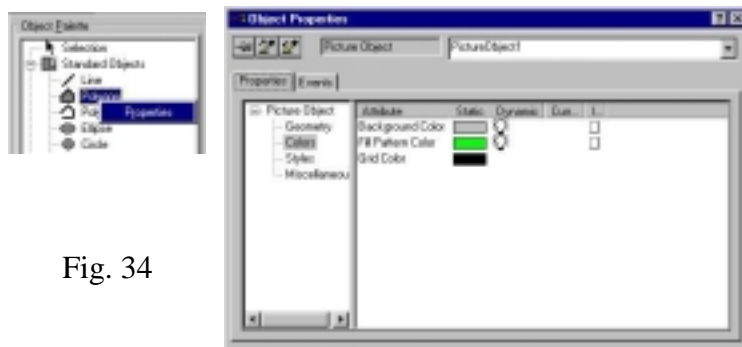


Fig. 34

Los botones de los termopares llaman a la imagen que representa gráficamente la evolución temporal de la temperatura correspondiente al termopar. Para ello asociaremos una acción a la pulsación del botón. Se podría hacer más sencillo, indicando en el diálogo de configuración una ventana para abrir, pero en ese caso se necesitaría una ventana independiente por cada termopar, con el consiguiente incremento en la utilización de recursos.

En su lugar se ha optado por incluir una única ventana actualizada dinámicamente.

La función correspondiente a la acción es:

```
#include "apdefap.h"
void OnClic(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

SetTagChar( "nTermopar", "Temp_arch\\Temp_1" ); //Return-Type :BOOL
OpenPicture( "Temp.Pdl" );
}
```

La función coloca en la variable **nTermopar** la fuente de actualización que deberá utilizar la gráfica: en el archivo Temp_arch, el campo correspondiente al termopar seleccionado. Después abre la imagen **Temp.Pdl**, que se encargará de dibujar la gráfica.

Las propiedades geométricas de los botones se definen como en los casos anteriores (Fig. 35).



Fig. 35

Las líneas se dibujan con la herramienta **Line**, haciendo clic sobre los puntos donde se quieran situar los extremos. Estas posiciones pueden ser modificadas posteriormente seleccionando la línea y arrastrando los cuadrados que aparecen en los extremos.

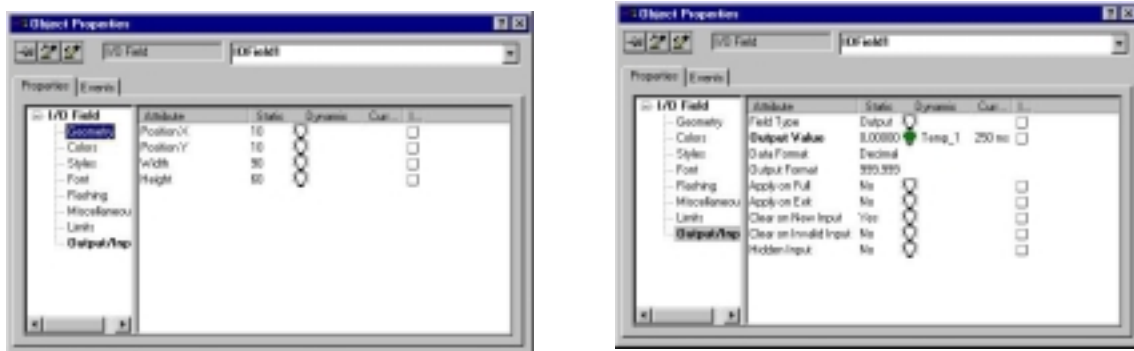


Fig. 36

Los campos de entrada/salida se configuran como salida (Fig. 36) y se conectan, mediante el diálogo de configuración (Fig. 37), a las variables que registran las temperaturas de los termopares, con un periodo de muestreo de 250ms.

Por último, se añade un control de tipo **WinCCAlarmControl**, que se presenta como una tabla para la monitorización y gestión de los mensajes de alarma y error generados por el registro de alarmas.

Para incluir este control hay que hacer doble clic sobre el elemento **Control** de **Smart Objects** en la paleta de objetos (Fig. 38) y seleccionar **WinCCAlarmControl** de la lista (Fig. 39). Después se sitúa como cualquier otro objeto sobre la imagen.

Configuraremos las propiedades geométricas del control de alarmas para que ocupe la pantalla completa (800x600) de manera que oculte toda la imagen. Para que aparezca solamente cuando hay un mensaje desactivaremos la propiedad **Visible** en **Miscellaneous: Display** (Fig. 40), que en su momento se activará desde otro lugar (cf. *infra*).



Fig. 37

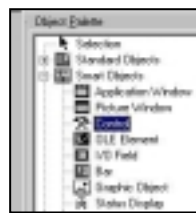


Fig. 38



Fig. 39

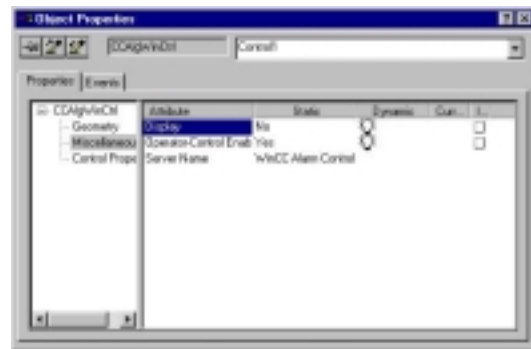
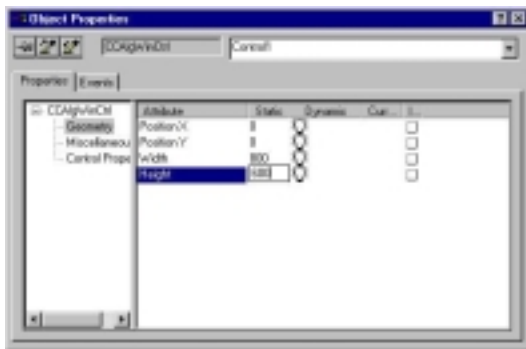


Fig. 40

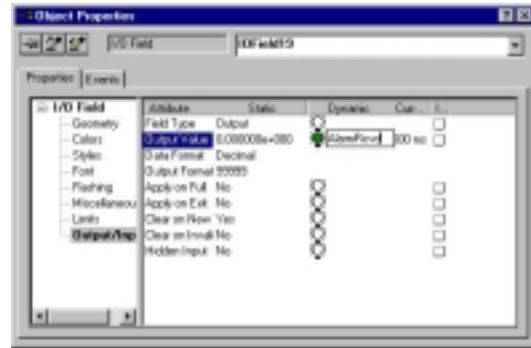
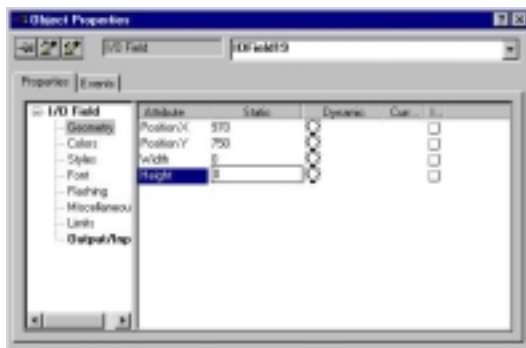


Fig. 41

Para activar la visualización del control de alarmas utilizaremos un campo de entrada/salida vinculado a una variable, **AlarmRcvd**, que es modificada por el registro de alarmas cuando se produce alguna.

Primeramente configuramos el campo de entrada/salida como salida y con tamaño nulo para que no se muestre y vinculamos la variable en la ventana de propiedades bajo **Output/Input: Output Value** (Fig. 41).

El siguiente paso consiste en asignar una acción al evento de cambio en la variable **AlarmRcvd**, bajo **Property Topics: Input/Output: Output Value: Change** en la ventana de eventos (Fig. 42). La acción comprueba que el cambio se deba a un mensaje del registro de alarmas comprobando el nuevo valor de **AlarmRcvd**, y en caso pertinente abre el control de alarmas.

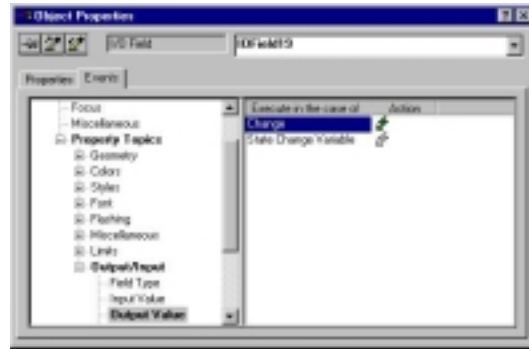


Fig. 42

Para continuar con la ventana **Main** basta con cerrar el control de alarmas, que volverá a aparecer si se genera otro mensaje.

3.2 Editor de alarmas

En el editor de alarmas se definen las circunstancias que producirán mensajes y se diseñan estos mensajes; qué dicen y qué información contienen.

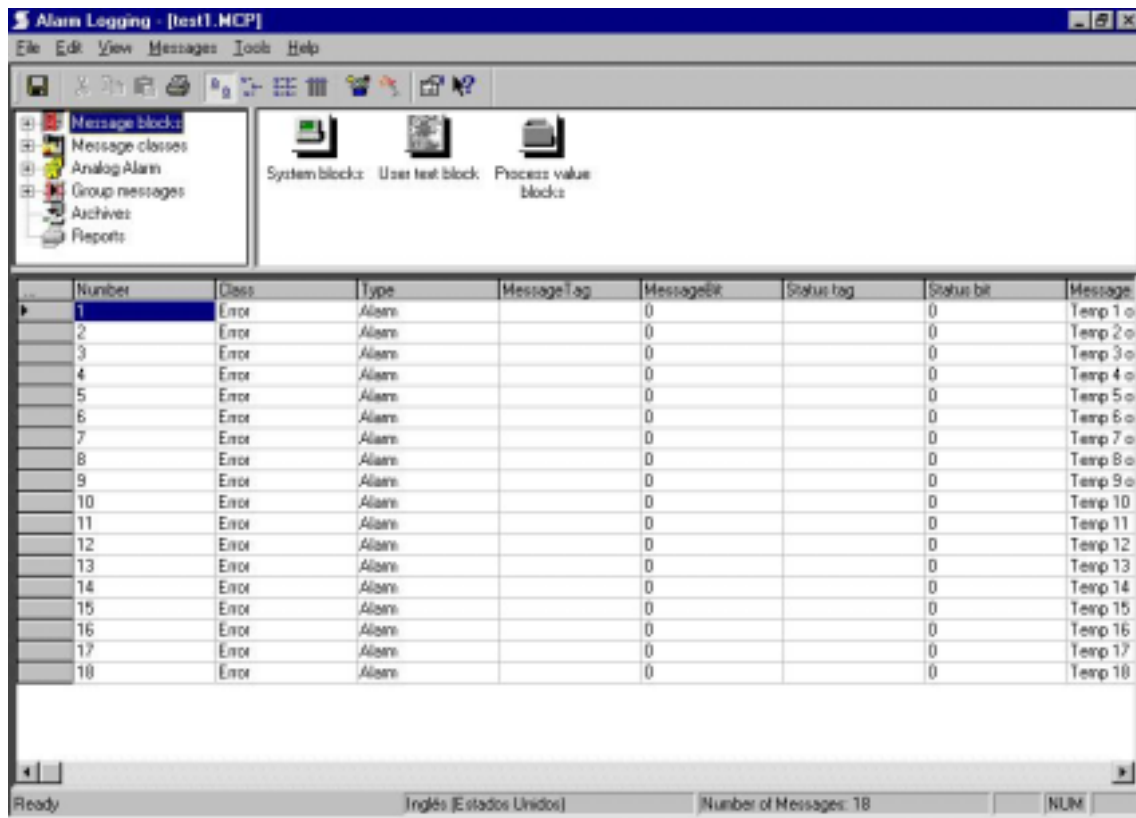


Fig. 43

El aspecto del editor es como se muestra en la fig. 43. El área de trabajo se divide en 3 partes: en la superior izquierda se encuentra la lista de elementos configurables. En la superior derecha se visualizan los componentes del elemento seleccionado para su edición. En la parte inferior se ve la tabla de mensajes.

El primer elemento que aparece para ser configurado es **Message Blocks**, que permite asignar las propiedades de los elementos que conformarán el mensaje. Estos elementos se dividen en tres categorías: **System** (sistema), que incluye fecha y hora en la que



Fig. 44



Fig. 45

se produce el mensaje, duración del mismo, estado actual y número del mensaje; **User Text Block** (bloques de texto definidos por el usuario), en el que creamos dos bloques: **message text** (texto del mensaje) y **Point of Error** (Localización del error); y **Process Value Blocks** (bloque de valores de proceso), que ha quedado vacío.

Los componentes de **System** están predefinidos, pero se puede configurar su representación. Para ello pulsamos sobre el componente con el botón derecho y seleccionamos **Properties** en el menú contextual (Fig. 44). En el cuadro de diálogo seleccionamos las opciones correspondientes (Fig. 45). En la figura se muestra el diálogo de configuración del campo **Date** (fecha), para una representación tipo día.mes.año, con el año representado con dos dígitos y justificado a la izquierda.

Para crear los bloques de texto definidos por el usuario, una vez seleccionado en el árbol de la parte superior izquierda de la ventana el elemento **Message Blocks: User Text Block**, pulsamos con el botón derecho sobre el área superior derecha y seleccionamos **Add/Remove** en el menú contextual (Fig. 46).

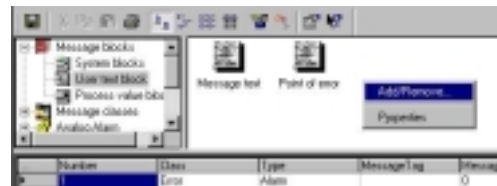


Fig. 46

En el cuadro de diálogo añadimos los bloques de texto necesarios seleccionándolos y pulsando en el botón -> y luego **OK** (Fig. 47). Después se configura cada uno de los bloques pulsando sobre su icono con el botón derecho y seleccionando **Properties** en el menú contextual (Fig. 48). En el cuadro de diálogo se definen el nombre del bloque, su longitud máxima, parpadeo y alineación (Fig. 49).



Fig. 47

Tanto para **Message Text** como para **Point of Error** se han definido las siguientes propiedades: longitud máxima de 20 caracteres, alineación a la izquierda y sin parpadeo.

No ha sido necesario definir ningún elemento en la sección de valores de proceso; en caso de que fuera preciso se procedería de manera análoga a la definición de bloques de texto definidos por el usuario.

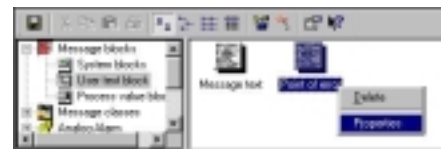


Fig. 48

En **Message Classes** (tipos de mensaje) se pueden definir distintas categorías en las que incluir los mensajes, cada una con diferentes características en la presentación: colores de las celdas en función del estado del mensaje, texto del estado, etc.



Fig. 49

En el presente proyecto no ha sido necesario crear nuevas categorías, aunque se ha modificado la de mensajes de error para hacer las alarmas más visibles.

De ser precisa la creación de nuevos tipos de mensaje se procedería análogamente al proceso de creación de nuevos bloques de texto definidos por el usuario: pulsar sobre el área de trabajo de **Message Classes** con el botón derecho y seleccionar **Add/Remove** en el menú contextual.

En el diálogo de configuración incluir los tipos de mensaje necesarios seleccionándolos y pulsando en **->** y luego **OK**. Los tipos de mensajes así creados se configuran del mismo modo que los existentes por defecto, como se indica a continuación.

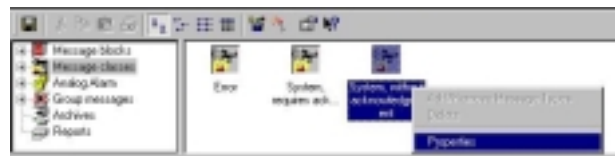


Fig. 50

Para configurar un tipo de mensaje, pulsamos sobre el icono con el botón derecho y seleccionamos **Properties** en el menú contextual (Fig. 50). El diálogo de configuración que aparece tiene tres apartados que se seleccionan con las lengüetas de la parte superior.

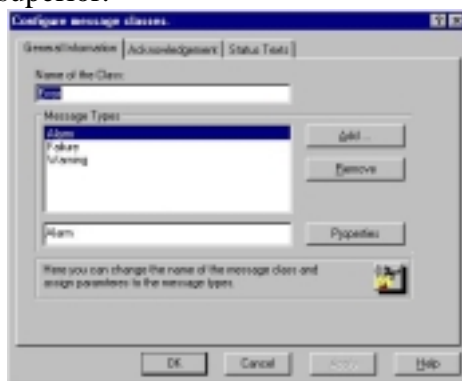


Fig. 51

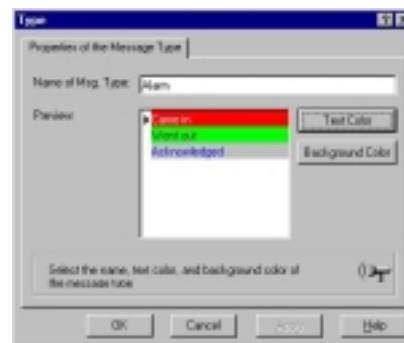


Fig. 52

La primera, **General Information** (información general, fig. 51), permite definir el nombre de la clase y los tipos de mensaje que engloba. Se pueden añadir o quitar tipos con el botón **Add/Remove**. Para configurar uno de los tipos lo seleccionamos y pulsamos sobre el botón **Properties**. Con esto se abre un nuevo diálogo de configuración (Fig. 52) donde podemos determinar el color del fondo y del texto del mensaje para los distintos estados: **came in** (recibido), **went out** (se recibió, pero la situación que lo provocó ha cesado) y **acknowledged** (aceptada por el operario).

La segunda, **Acknowledgement** (aceptación de la alarma, fig. 53), permite configurar los modos en los que se puede aceptar el mensaje: *Acknowledgement came in* indica si el mensaje precisa reconocimiento cuando se produce; *Acknowledgement went*

out indica si es necesario el reconocimiento del mensaje cuando cesa la causa; *Flashing On* indica si el mensaje parpadea o no; *Central Signaling Devices* indica si se utiliza un reconocimiento independiente (*Separate Ack Key*) para los dispositivos centrales de señalización (tales como sirenas), o bien se controlan mediante el reconocimiento del mensaje (*By Means Of Single Ack.*), i.e. al aceptar el mensaje se detiene el dispositivo.

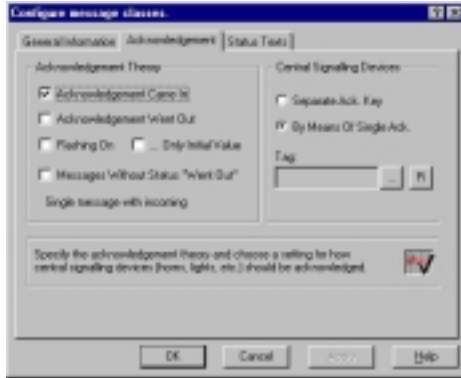


Fig. 53

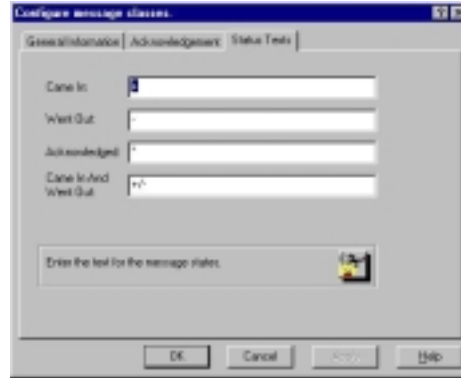


Fig. 54

La tercera, **Status Texts** (Textos de estado, fig. 54), permite definir las cadenas que definirán en el mensaje el estado. Los estados posibles son: *Came in*, recibido; *Went out*, recibido y reconocido pero después la causa cesó; *Acknowledged*, reconocido; *Came in and Went out*, recibido y la causa cesó, sin que se reconociera el mensaje en ese periodo.



Fig. 55

El siguiente elemento de la configuración es **Analog Alarms**, que permite generar avisos que se activan cuando el valor de una variable excede unos límites impuestos. En el presente proyecto se utilizan estas alarmas para indicar una salida de rango de las medidas de los termopares.

Para definir una alarma analógica pulsamos con el botón derecho sobre el área de trabajo de **Analog Alarms** y seleccionamos **New** del menú contextual (Fig. 55). A continuación obtendremos un diálogo de configuración (Fig. 57) igual al que obtendríamos para editar las propiedades de una alarma analógica existente si pulsásemos sobre ella con el botón derecho y seleccionásemos **Properties** en el menú contextual (Fig. 56).



Fig. 56



Fig. 57

El campo **Tag to be monitored:** indica el nombre de la variable cuyos valores serán comprobados por la alarma analógica; en este caso, la variable que guarda la temperatura leída por el termopar correspondiente.

La casilla de verificación **a message for all limit values; message number:** indica cuando está marcada que tanto si se supera el límite superior como si se rebasa el inferior se generará el mismo mensaje, que será el indicado en el campo que aparece a continuación. Para el contenido de los mensajes ver la descripción más abajo.

Delay permite establecer un retraso entre la detección del rebasamiento del límite y la generación del mensaje.

Haciendo doble clic sobre una alarma analógica se pueden definir los límites para la variable que comprueba (Fig. 58). Se crean nuevos límites pulsando sobre el área de trabajo con el botón derecho y seleccionando **New** del menú contextual. Haciendo doble clic sobre los iconos correspondientes a los límites se accede a su diálogo de configuración (Fig. 59).

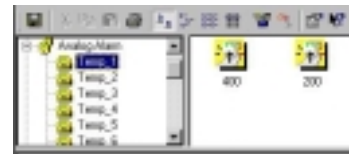


Fig. 58

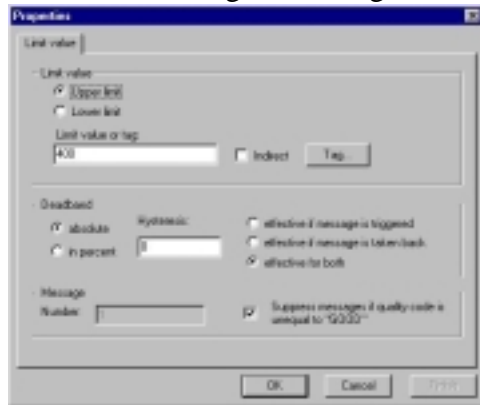


Fig. 59

En este diálogo se puede indicar si se trata de un límite superior (**Upper Limit**) o inferior (**Lower Limit**); se puede definir el valor límite (**Limit Value or Tag**) bien indicando directamente el valor o marcando la casilla **Indirect** y asignando una variable de la que se leerá dinámicamente el valor límite.

Deadband permite definir un ciclo de histéresis, al dar un margen sobre el valor límite. Este margen puede ser expresado en términos absolutos (*absolute*) o relativos en porcentaje (**in percent**). Este efecto puede ser aplicado bien sólo a la generación del mensaje (**effective if message is triggered**), sólo a su desactivación (**effective if message is taken back**) o a ambos (**effective for both**).

Se pueden definir varios límites tanto superiores como inferiores. Si no se ha activado la opción de utilizar el mismo mensaje para todos los límites esto permite, por ejemplo, gestionar mensajes indicativos de distintos niveles de severidad en la alarma.

En el presente proyecto no ha sido necesaria la utilización de mensajes en grupo, archivos de mensajes o informes de mensajes; debido a esto no se ha realizado ninguna modificación en los tres últimos elementos: **Group Messages**, **Archives** y **Reports**.



Fig. 60



Fig. 61

Para crear un nuevo mensaje pulsamos con el botón derecho sobre el área inferior de la ventana, donde se muestra la tabla de mensajes y seleccionamos **Append New Line** del menú contextual (Fig. 60). Para configurar el mensaje pulsamos con el botón derecho sobre la línea del mensaje y seleccionamos **Properties** en el menú contextual (Fig. 61).

El diálogo de configuración se divide en tres secciones: **Parameters** (parámetros, fig. 62), **Text** (texto, fig. 63) y **Tag/Action** (variable/acción, fig. 64).

En la sección **Parameters** seleccionamos en las listas desplegadas la clase (**Class**), tipo (**Type**) y grupo (**group**) a los que pertenecerá el mensaje de entre los creados en las etapas anteriores descritas arriba. El grupo de opciones englobado en **This message** permite activar o desactivar las siguientes propiedades:

- **is single acknowledgement only**: indica que sólo se puede reconocer este mensaje de forma individual.

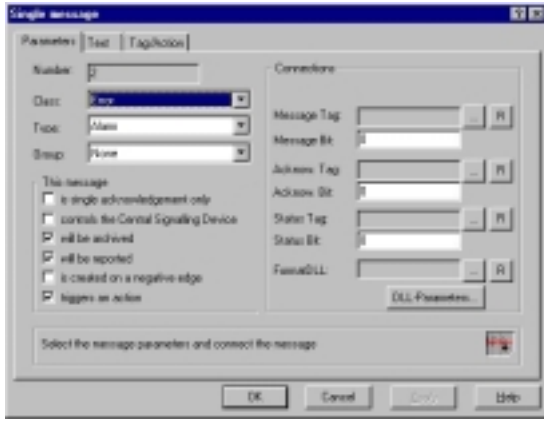


Fig. 62

indica que el mensaje se genera en un flanco descendente del reloj.

- **Triggers an action:** indica que el mensaje lanza una acción.

El grupo **Connections** permite monitorizar el mensaje a través de variables. Message Tag será la variable utilizada para indicar la emisión de un mensaje, modificando el bit indicado en **Message Bit**. Análogamente el reconocimiento del mensaje se realizará mediante la variable **Acknow. Tag** y el bit **Acknow. Bit** y el estado mediante la variable **Status Tag** y el bit **Status Bit**.

En la sección **Text** se definen los textos de los bloques de texto definidos por el usuario. Para cada uno de los bloques se introduce la cadena de texto que queremos que aparezca en el mensaje.

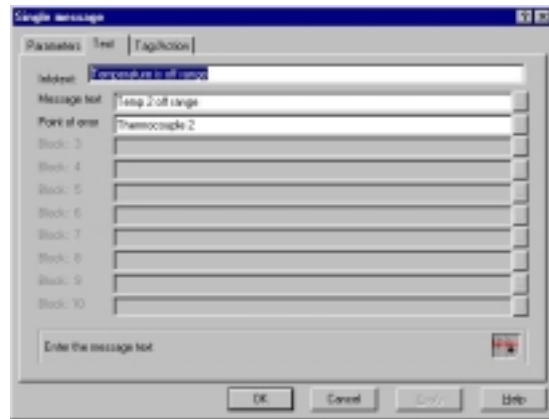


Fig. 63

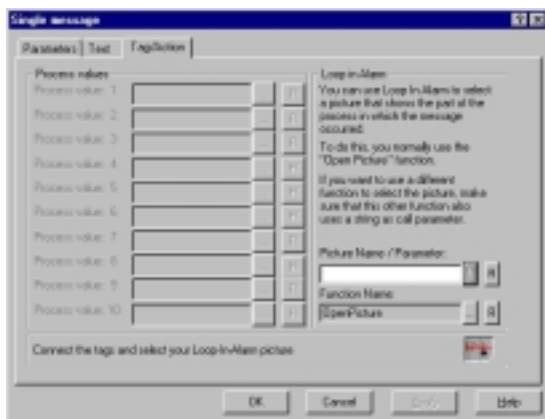


Fig. 61

Dado que el presente proyecto tan sólo presenta una pantalla y basta con cerrar el control de alarmas para volver a ella no se ha configurado esta función.

- **Controls the Central Signaling Device:** indica que el dispositivo central de señalización responderá al estado de este mensaje.
- **Will be archived:** indica que el mensaje podrá ser guardado en un archivo, registrando sus ocurrencias.
- **Will be reported:** indica que el mensaje podrá ser accedido desde un informe para su presentación.
- **Is created on a negative edge:**

En la sección **Tag/Action** se asignan las variables que se monitorizan a través de los valores de proceso, i.e. el mensaje mostrará el valor de la variable asignada al valor de proceso en el momento en el que se generó el mensaje.

También permite configurar la función **Loop in Alarm**, por la que pulsando sobre el mensaje se abre la ventana indicada, por ejemplo para ver la parte del proceso donde ocurrió la alarma.

3.3 Editor de archivos

En el editor de archivos se crean y configuran los archivos en los que se almacenarán valores de variables. En el presente proyecto se ha creado un único archivo que incluye los valores registrados por los 18 termopares.

El aspecto del editor se muestra en la fig. 62.

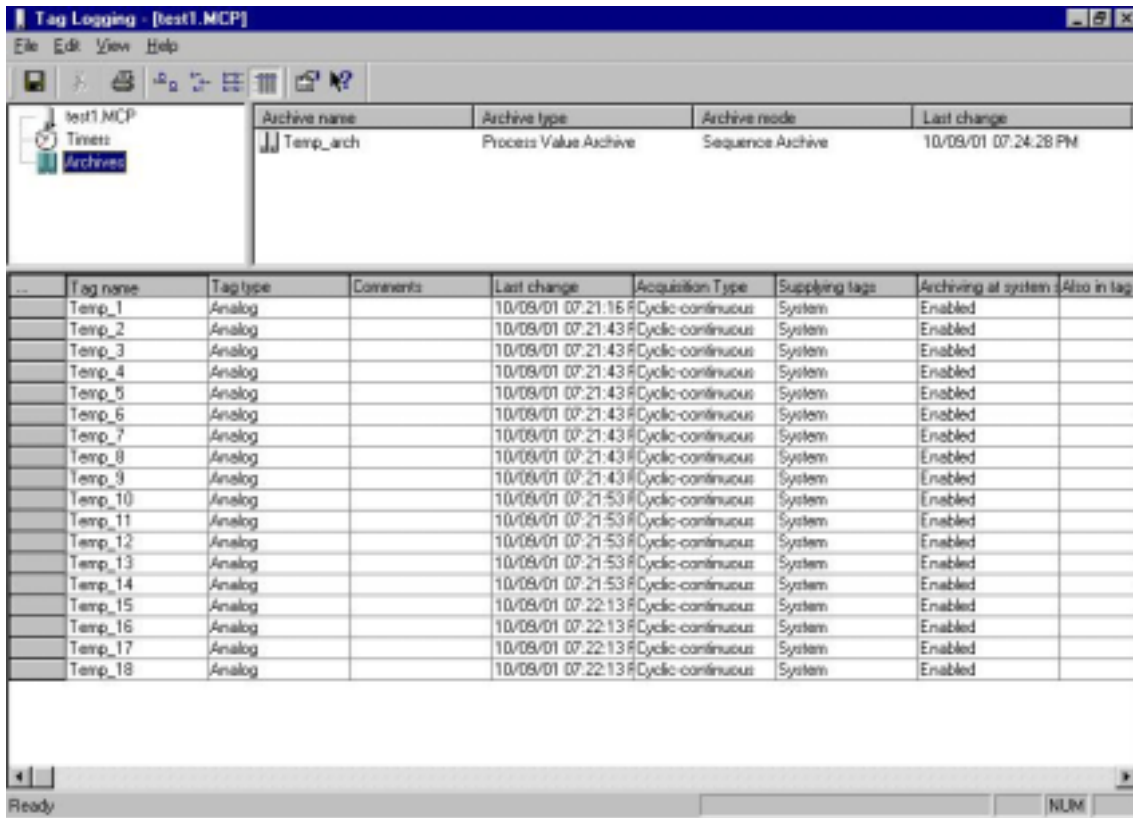


Fig. 62

La ventana del editor se divide en tres áreas: en la superior izquierda está la lista de elementos que se pueden configurar (temporizadores y archivos) y en la superior derecha aparecen los componentes del elemento seleccionado. En el área inferior se visualiza la estructura del archivo seleccionado.



Fig. 63

menú contextual.

El primer elemento configurable es el temporizador. Para crear un nuevo temporizador pulsamos con el botón derecho sobre **Timers** y seleccionamos **New...** (Fig. 63) en el menú contextual. Para la configuración del nuevo temporizador aparece un cuadro de diálogo (Fig. 64) igual al que se obtiene para la configuración de un temporizador existente pulsando sobre él con el botón derecho y seleccionando **Properties** en el

menú contextual. En el cuadro de diálogo escribimos el nombre que queremos darle al temporizador en el campo **Name**. Para definir el periodo hay que combinar los campos **Base** y **Factor**; El periodo resulta de multiplicar la base seleccionada en la lista desplegable de **Base** por el número indicado en **Factor** (e.g en la fig. 64 el periodo es de 500ms = 1*500ms).



Fig. 64

continúa el momento en el que debe iniciarse el temporizador.

El otro elemento de configuración es el de los archivos. Para definir un nuevo archivo se llama al asistente de creación de archivos pulsando sobre el icono **Archives** con el botón derecho y seleccionando **Archives Wizard** del menú contextual (Fig. 65).

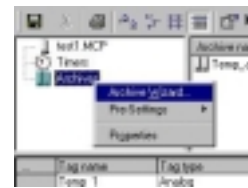


Fig. 65

El primer cuadro de diálogo del asistente (Fig. 66) pide al usuario que introduzca el nombre en el campo **Archive** y que seleccione el tipo de archivo: *Process Value Archive*, que almacena los valores y los mantiene accesibles al sistema, o *Compressed Archive*, que guarda los datos en formato comprimido lo que disminuye el espacio ocupado en disco pero no permite el acceso en tiempo real.



Fig. 66



Fig. 67

En el segundo cuadro de diálogo (Fig. 67) se seleccionan las variables que se almacenarán en el archivo. Para ello pulsamos el botón **Select...** y en la ventana de selección (Fig. 68) hacemos doble clic sobre la variable que queremos incluir. Para el presente proyecto se seleccionaron las variables **Temp_1** a **Temp_18**, para registrar la evolución de las temperaturas medidas por cada termopar.

Para editar las propiedades de un archivo pulsamos sobre él con el botón derecho y seleccionamos **Properties** en el menú contextual (Fig. 69). En el diálogo de configuración hay dos partes: **General Information** (Información General, fig. 70) y **Archive Parameters** (Parámetros del Archivo, fig. 71).

En **General Information** podemos modificar el nombre del archivo (**Archive Name**), añadir comentarios para una mejor descripción del archivo y su función o seleccionar el servidor si hay varios.



Fig. 68

En **Archiving at System Start** se define si se inicia el proceso de registro de datos al iniciar el sistema (**Enabled**) o si hay que iniciarlo posteriormente, e.g. desde una acción (**Locked**). Para el presente proyecto se eligió la opción de iniciar el archivo con el sistema, ya que queremos guardar los datos de toda la colada. Sin embargo, si la obtención de datos de colada fuese anterior al comienzo de ésta, debería cambiarse esta opción e implementarse el inicio del registro mediante alguna señal que indique el comienzo de la colada, tal como el cambio de alguna de las variables.

En **Archive Type** se define el tipo de archivo; al haber definido este aspecto en el asistente, se encuentra bloqueado y sólo admite la opción **Process Value Archive**.



Fig. 69

En **Authorization Level** se pueden incluir contraseñas par proteger el acceso a los datos. Pueden especificarse contraseñas diferentes para lectura y escritura.

En **Action during Archive Start/Enable** se puede indicar una acción para que sea ejecutada al iniciarse el registro, y cada vez que éste se reanuda.

En **Archive Parameters** Se definen las características del archivo. En primer lugar, bajo **archive Mode**, se indica si el archivo es de corto plazo (**Short-term archive**) o de secuencia (**Sequence archive**). En el presente proyecto se ha optado por un archivo de secuencia ya que permite una longitud indefinida (el archivo de corto plazo tiene una longitud limitada, insuficiente teniendo en cuenta la frecuencia de las mediciones) y la permanencia de los datos (el archivo de corto plazo se pierde al cerrar el sistema).

En el caso de seleccionar un archivo de secuencia no se pueden determinar más propiedades. Para un archivo de corto plazo se pueden definir los siguientes parámetros:

- **Number of Entries** (Número de entradas), máximo número de registros que puede almacenar el archivo; este parámetro no se puede definir directamente, sino que depende del tamaño máximo del archivo (*cf. infra*).
- **Size in Kbytes/Tags** (Tamaño en Kbytes/Tags), Tamaño máximo del archivo expresado en Kbytes por cada variable registrada.

- **Memory Location** (Localización en memoria), indica si el archivo se almacenará en la memoria principal (**In the main memory**) o en el disco duro (**On the hard disk**).
- **Action for exporting the short-term archive** (Acción para exportar el archivo de corto plazo), permite seleccionar una acción para guardar el archivo de corto plazo en un formato que dependerá de la acción seleccionada.
- **Fill Level Message** (Mensaje de nivel de llenado), indica en que momento se presenta un mensaje de advertencia por el llenado del archivo en función del porcentaje utilizado del tamaño máximo; **At 100% full** indica que el mensaje se presente cuando el archivo se llene por completo. En la lista desplegable se puede seleccionar otro porcentaje de llenado para la presentación del mensaje.



Fig. 70



Fig. 71

3.4 Editor de guiones

El editor de guiones permite editar funciones para su utilización como acciones en las imágenes, botones, controles, etc., así como modificar las funciones existentes para su adaptación a los requerimientos del proyecto. Su aspecto se muestra en la fig. 72.

En el árbol de funciones que aparece a la izquierda se selecciona la función que se quiere editar, y se abre haciendo doble clic sobre ella. Si en lugar de hacer doble clic se pulsa el botón derecho sobre la función y se selecciona la opción **Supply Parameters**, aparecerá un cuadro de diálogo para introducir los parámetros (este cuadro de diálogo es idéntico al descrito en la sección del editor gráfico, definición de acciones para el botón de **Ret_Cast**, Fig. 30); el editor insertará la función con los parámetros indicados en la posición actual del cursor.

Las funciones definidas en el proyecto son **IsSteelGrade()**, **IsPowderType()** y **ShowAlarms()**.

3.4.1 IsSteelGrade()

Esta función determina si la cadena que se le pasa como parámetro es un tipo válido de acero. Por el momento sólo comprueba que el valor no sea el inicial ('-'); para su aplicación real debería comparar la cadena con los elementos de una lista que contuviera todos los tipos de acero admisibles.

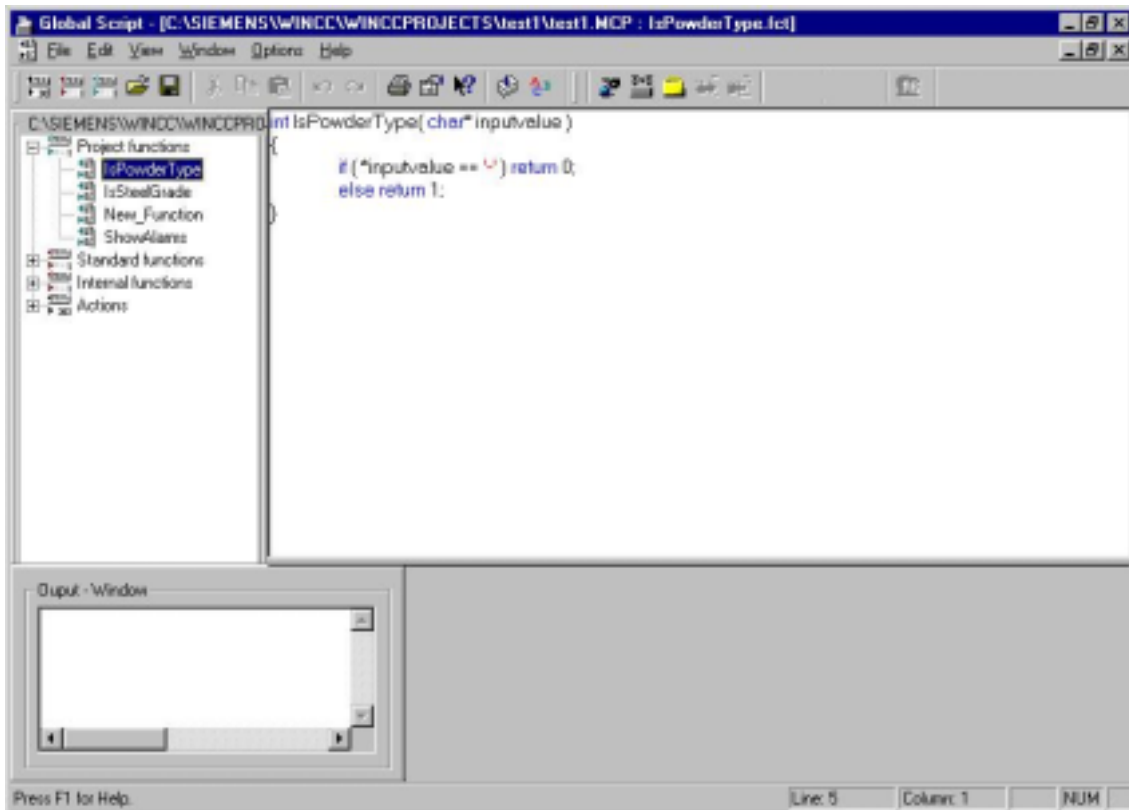


Fig. 72

```
int IsSteelGrade( char* inputvalue )
{
    if ( *inputvalue == '-' ) return 0;
    else return 1;
}

```

3.4.2 IsPowderType()

Esta función determina si la cadena que se le pasa como parámetro es un tipo válido de polvo de lubricación. Por el momento sólo comprueba que el valor no sea el inicial ('-'); para su aplicación real debería comparar la cadena con los elementos de una lista que contuviera todos los tipos de polvo admisibles.

```
int IsPowderType( char* inputvalue )
{
    if ( *inputvalue == '-' ) return 0;
    else return 1;
}

```

3.4.3 ShowAlarms()

Esta función hace visible la tabla de mensajes de alarma y error.

```
void ShowAlarms( char* param )
{
    SetPropBOOL( "Main.Pdl", "Controll", "Visible", -1 );
}

```


3.5 Editor de archivos de usuario

El editor de archivos de usuario (Fig. 73) permite configurar archivos de datos y vistas de éstos para su uso con la base de datos interna (SyBase SQL). Para el presente proyecto se ha creado un único archivo con un campo por cada termopar para almacenar las medidas de temperatura, y un campo de fecha y hora para tener una referencia temporal del momento de la medida.

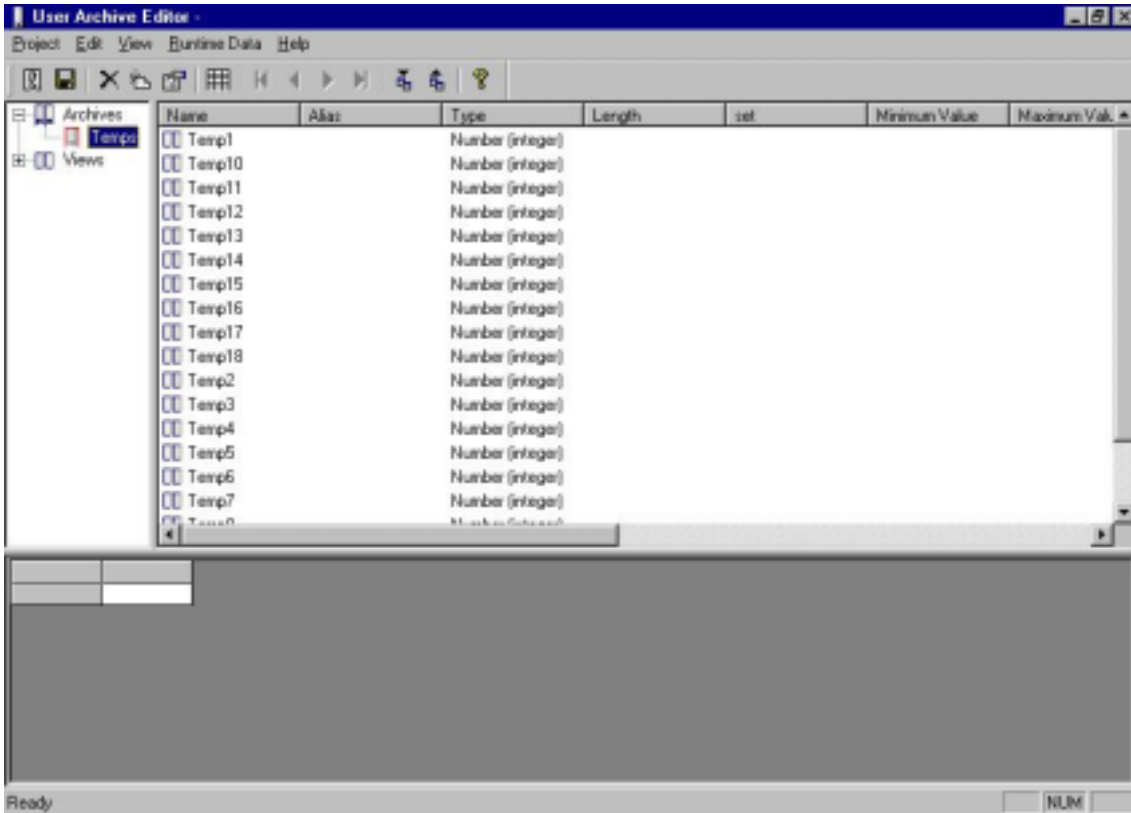


Fig. 73

Para crear un archivo pulsamos sobre **Archives** con el botón derecho y seleccionamos **New Archive** en el menú contextual (Fig. 74). En el diálogo de configuración (Fig. 75) indicaremos el nombre que queremos dar al archivo (**Archivename**), el **alias**, si se quiere definir uno, y si el archivo será de tamaño limitado (**Limited**) o no (**Unlimited**); en caso afirmativo deberá asignarse en **Records** el número máximo de registros que puede contener.



Fig. 74

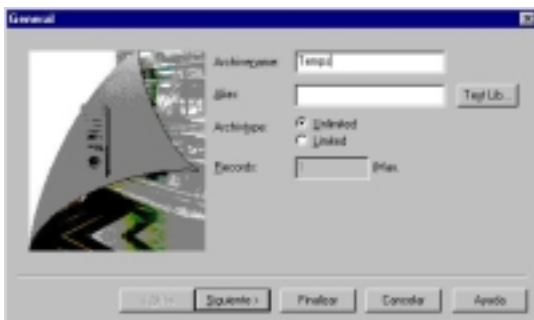


Fig. 72



Fig. 72

Pulsando en **Siguiente** se suceden una serie de diálogos de configuración para otros aspectos del archivo. Estos diálogos son los mismos que los correspondientes a la modificación de propiedades de archivos existentes, que se explican a continuación. Para acceder a ellos, pulsamos sobre el archivo con el botón derecho y seleccionamos **Properties** en el menú contextual.

En la sección **Communication** (Fig. 76) se especifica si el archivo utilizará algún método de comunicación, y en su caso cuál. Las opciones posibles son: sin comunicación (**No communication**), Comunicación a través de una variable de WinCC sin formato (**Communication via a WinCC Raw Data Tag¹**) y comunicación a través de una variable de WinCC (**Communication via a WinCC Tag**).



Fig. 76

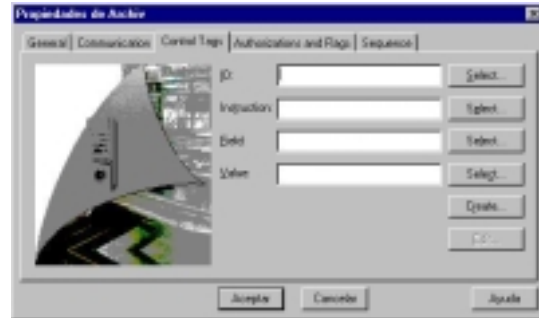


Fig. 77

En la sección **Control Tags** (Fig. 77) se definen las variables de control. Estas variables permiten la manipulación del archivo. Las variables de control son: **ID**, que indica el identificador del registro que se quiere seleccionar; **Instruction**, que puede ser leer (*Read*), escribir (*Write*) o borrar (*Delete*); **Field**, que indica el campo del archivo y **Value**, el valor que se quiere escribir (o recibe el valor en caso de lectura o borrado).

En la sección **Authorization and Flags** (Fig. 78) se asignan las contraseñas para restringir el acceso de lectura (*Reading*) y escritura (*Writing*) al archivo. También se pueden activar las banderas (*Flags*) que almacenan para cada campo el momento de la última modificación (**Field – Last modification**) y el usuario que la realizó (**Field – Last user**).

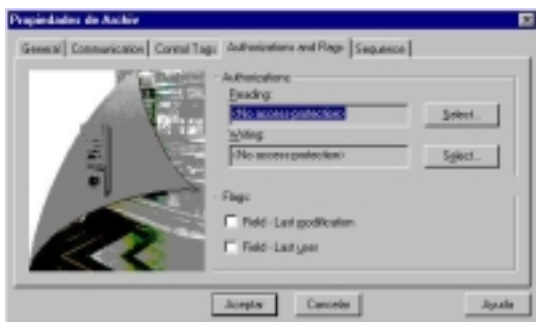


Fig. 78



Fig. 79

En la sección **Sequence** (Fig. 79) se establece la secuencia de archivos seleccionándolos y desplazándolos hacia arriba (mayor prioridad) con el botón **Up** y hacia abajo (menor prioridad) con el botón **Down**.

¹ Las variables sin formato (Raw Data Tag) permiten direccionar memoria del PLC y copiarla tal cual, sin interpretarla como un número, un carácter, etc.

Las vistas (**Views**) son presentaciones de los datos de los archivos de acuerdo a ciertos criterios establecidos en la definición de la vista. Una vista puede presentar todas o parte de las columnas (campos) del archivo, y todos, parte o ninguno de los registros, en función del resultado de las condiciones.

Para crear una nueva vista pulsamos con el botón derecho sobre **Views** y seleccionamos **New View** en el menú contextual (Fig. 80).



Fig. 80

Los diálogos de configuración son los mismos que los de edición de propiedades, que se obtienen pulsando con el botón derecho sobre el icono de la vista y seleccionando **Properties** en el menú contextual.

En el diálogo **General** (Fig. 81) indicaremos el nombre de la vista y, en su caso, el alias.



Fig. 81



Fig. 82

En el diálogo **Relation** (Fig. 82) se establece la condición que deben cumplir los registros que aparezcan en la vista. Esta condición se escribe en formato SQL.

El diálogo **Sequence** (Fig. 83) es igual que el homónimo de archivos, pero referido a las vistas.



Fig. 83

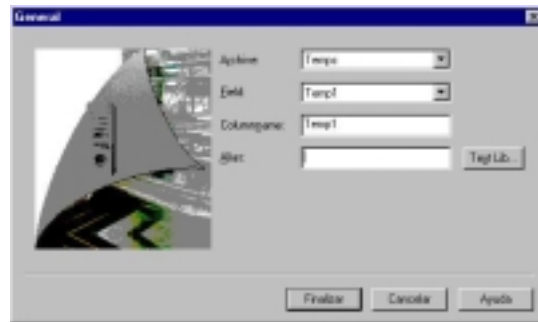


Fig. 84

Para seleccionar cuáles son las columnas que aparecerán en la vista pulsamos con el botón derecho sobre el icono de la vista y seleccionamos **New Column** en el menú contextual. En el diálogo que se presenta (Fig. 84) seleccionamos el archivo del que queremos seleccionar la columna (**Archive**), la columna que queremos incluir en la vista (**Field**) y escribimos el nombre con el que queremos que aparezca (**Columnname**) y, en su caso el **Alias**.

Repetiremos este proceso para cada una de las columnas que queramos incluir.

4 Pruebas

Para comprobar los resultados es necesario que los valores de las variables cambien, en particular las temperaturas de los termopares. Con este fin, se han añadido a la ventana *Main* tres botones: **See Alarms**, que hace visible la tabla de mensajes; **Temp2++**, que incrementa en una unidad el valor de la variable *Temp_2*; y **Temp1--**, que disminuye en una unidad el valor de dicha variable.

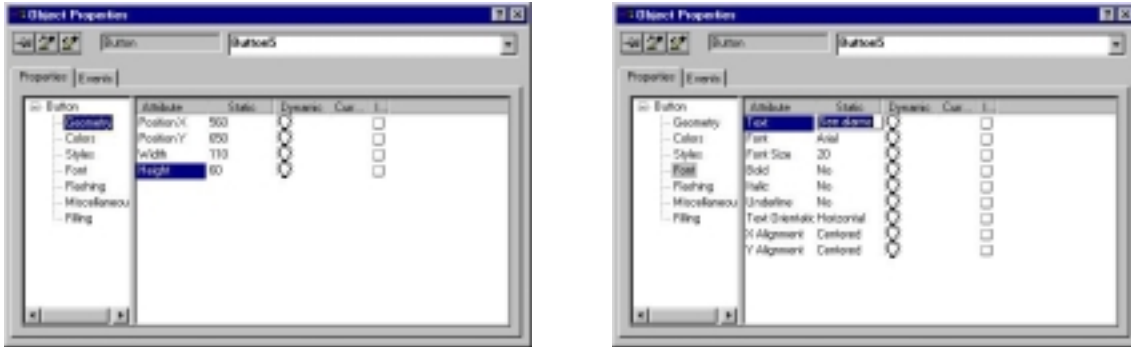


Fig. 85

Los botones se crean y sitúan como se explica en la sección dedicada al editor gráfico. Las propiedades se configuran como se muestra en las figs. 85 (**See Alarms**), 86 (**Temp++**) y 87 (**Temp--**).

Por último se definen las acciones asociadas a la pulsación de cada uno de los

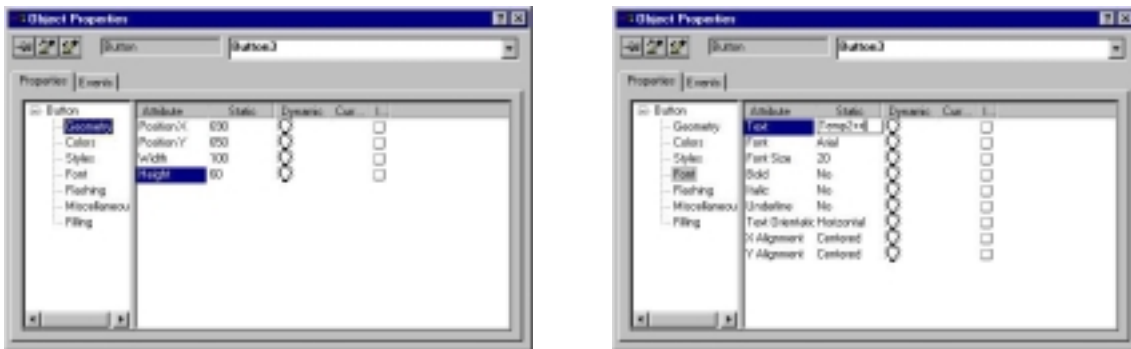


Fig. 86

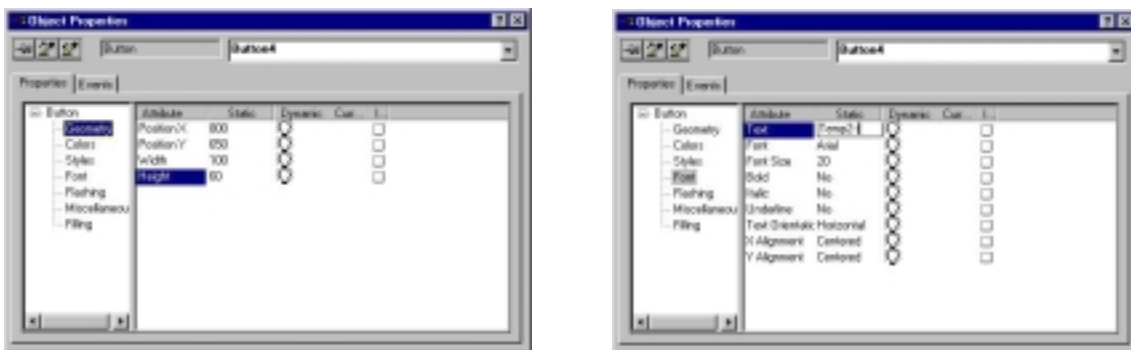


Fig. 87

botones.

Para **See Alarms** la función es:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

SetPropBOOL( "Main.Pdl", "Controll", "Visible", -1 );

}

```

Para Temp2++ es:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

double aux;

aux = GetTagDouble( "Temp_2" ); //Return-Type :double
aux++;
SetTagDouble( "Temp_2", aux ); //Return-Type :BOOL

}

```

Y para Temp2-- es:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

```

```
// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

double aux;

aux = GetTagDouble( "Temp_2" );           //Return-Type :double
aux--;
SetTagDouble( "Temp_2", aux );           //Return-Type :BOOL

}
```

Con ayuda de estos botones se puede comprobar la respuesta a las variaciones de temperatura, la aparición o no de la tabla de mensajes, etc.