Capitulo 5

Guarda cuando diga.

Lo normal es guardar siempre, para ver los valores a lo largo del tiempo. Pero siempre sale el pijotero que quiere guardar cuando a él le dé la gana. El que quiere la gráfica de los valores, pero solo cuando pase la pieza, o durante un determinado espacio de tiempo, o si gana su equipo de futbol. Es ese mismo individuo que luego nunca mira los reports de las gráficas, y que es tan inútil con el ordenador que se excusa en que tiene mucho trabajo para que el operario jovencito realice las modificaciones en el sistema (en realidad no sabe ni meter el password).

Ejemplos de estos especimenes pueden ser:

- Paso de piezas por un detector. Cuando se active el detector tiene que almacenar el peso de la pieza. Es un ejemplo de almacenamiento acíclico.
- Prensa que tiene un ciclo de funcionamiento de 10 segundos cada minuto. Es necesario conocer la presión durante esos 10 segundos. El resto del tiempo no interesa, ya que se sabe va a estar parada. Es un ejemplo de almacenamiento de ciclo selectivo.

Vamos a realizar un ejemplo de cada uno de ellos para que cuando pase le peguemos en los morros al tipejo.

Almacenamiento acíclico.

Como dijimos tenemos que guardar el peso de una pieza, que está contenido en un valor real dentro del plc (en nuestro caso lo haré en una variable interna) cada vez que pase por delante del detector, que será otra variable de tipo bool.

El primer problema es que el WinCC no hace esto. En realidad lo que entiende por acíclico es que frente a un flaco positivo o negativo de la variable booleana guarda el valor de la real. Es decir, guardaría cada vez que pase por delante del detector, y cuando abandone el detector. La única manera de evitar esto es realizar en el plc una sentencia que haga de telerruptor: una marca auxiliar que va a ponerse cuando pase la pieza, y quitarse cuando pase la siguiente (ver mi libro Programación en Step 7).

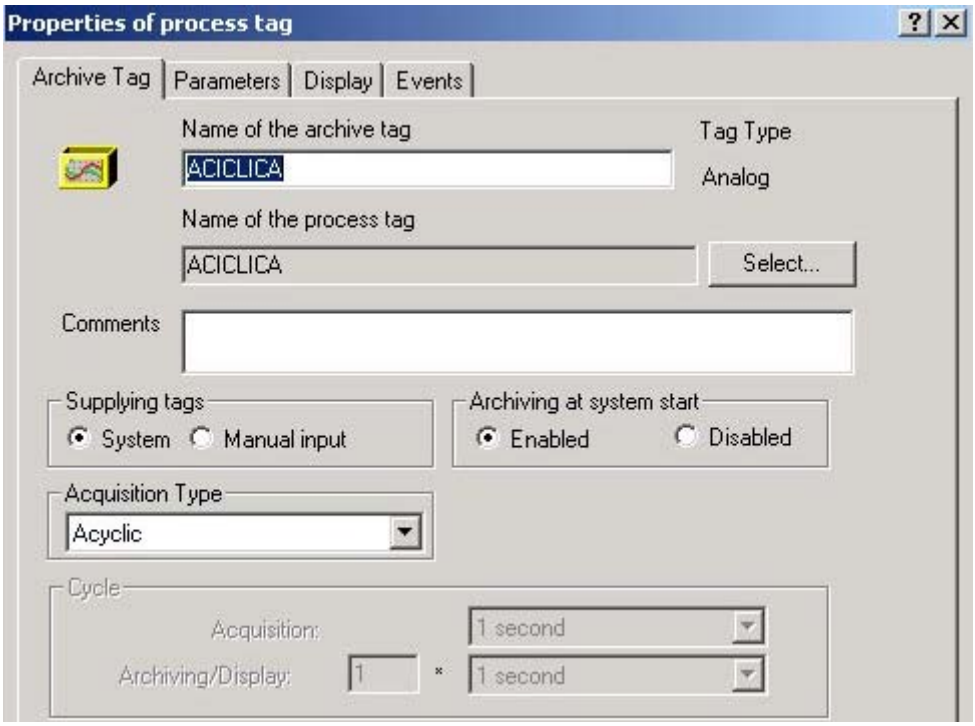
Bien, vamos a empezar haciéndonos dos variables una que llamaremos start_aciclica, de tipo bool, y otra llamada acíclica, de tipo real. La primera es el auxiliar del detector, la segunda es el peso de la pieza.

El primer paso es irse al global script (si, al global, no al tag logging ¿sorprendido?, jeje). Vamos a crear una project function que tiene el siguiente aspecto:

```
BOOL ACICLICA()
{

BOOL valor;

valor=GetTagBitWait("START_ACICLICA"); //Return - Type :BOOL
return(valor);
}
```

La guardamos con el nombre aciclica.fct. Aquí hemos acabado.

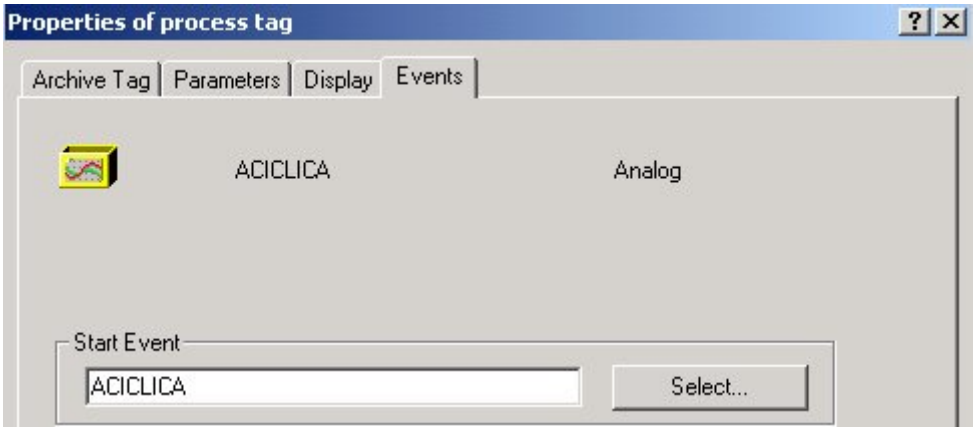
Ahora nos vamos al tag logging, y creamos un nuevo archivo llamado aciclicas, que se compone de una tendencia, llamada aciclica, que contiene una variable, que es aciclica, anteriormente definida (el peso de la pieza).



Vamos a ver que hemos hecho:

- La definimos como acquisition type: acyclic.
- Le asignamos la variable de proceso.
- Date cuenta que el tiempo de adquisición no se puede parametrizar, es fijo a 1 segundo. Esto quiere decir que si la pieza no esta mínimo un segundo delante del detector va a ver menos que steve wonder en un tunel. La úynica solución sería poner un temporizador de retardo a la desconexión activado por el detector, y eso a una marca que es la que nos genere los flancos que dispararán la adquisición. Claro está, almacenado en el momento del disparo del temporizador de retardo el valor actual del peso, ya que si no pesaremos un segundo después.

El rollito está en la última solapa, en la que debemos de asignar una función de proyecto que nos devuelva flancos para el almacenamiento de la variable. La verdad es que no se me ocurre que función podría ser la que debemos de asignar, hay tantas...¿A ti sí?¿la que hemos hecho antes?. Que listo, toma un azucarillo, te lo has ganado.



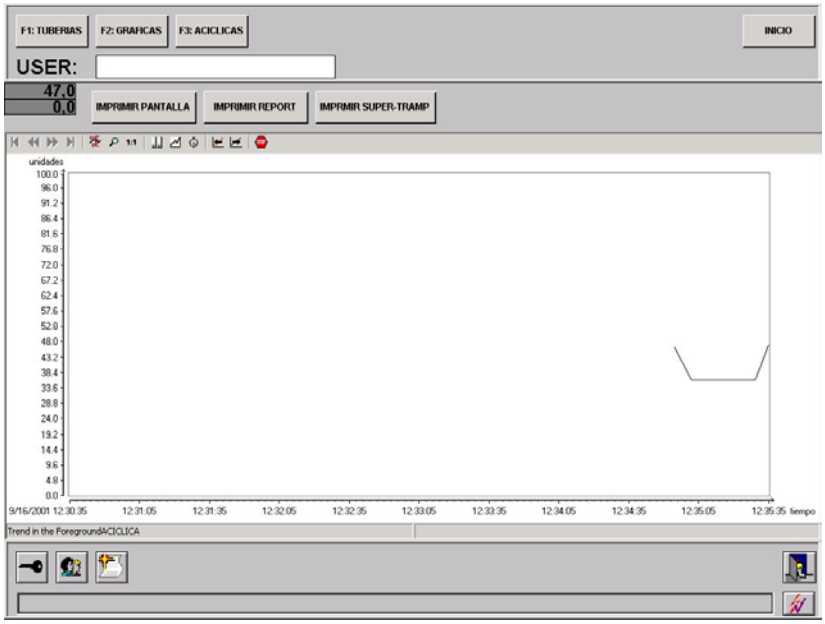
Ya hemos acabado. Acuérdate que en las propiedades de nuestro archivo aciclicas debemos definir la cantidad de puntos del tambor, que por defecto son 100 valores.

Ya solo nos queda comprobar si está arrancada la parte del runtime dedicada al global script (sí hombre, en control center, en computer). Y vamos por fin a poner la gráfica. Creamos una nueva picture (recuerda, no de nuevo, sino abriendo la picture plantilla.pdl y renombrándola). La guardaremos como grafica_aciclicas.pdl. Abrimos acto seguido top.pdl y copiamos el segundo botón y lo copiamos al lado, cambiando el salto a esta nueva gráfica, es decir:

OpenPicture("GRAFICAS_ACICLICAS.PDL"); //Return-Type :void

Guardamos abrimos graifcas_aciclicas.pdl y insertamos un control online trend control, que es igual que el de continuo, salvo que ahora la tendencia a representar es ACICLICAS\ACICLICA

Nos colocamos dos I/O arriba de la gráfica, una con la variable aciclica y otra con start_aciclica. Arrancamos y damos un valor a aciclica, p. Ej 45. Cada vez que modifiquemos el valor de start_aciclica de 0 a 1 o viceversa se irá almacenando el valor de aciclica.



Almacenamiento ciclo selectivo.

Vamos a estudiar ahora como realizar un almacenamiento de ciclo selectivo. El ejemplo que vamos a tomar será el de una prensa. Cuando está prensando vamos a guardar los valores, para poder observar la evolución de la presión realizada en el tiempo. Vamos a definirnos dos variables igual que antes, una start_ciclo_selectivo de tipo bool y otra presión de tipo real.

De nuevo comenzaremos en el global script, realizando esta vez dos project functions:

```
BOOL START_CICLO_SELECTIVO()
{

BOOL valor;

valor=GetTagBitWait("start_ciclo_selectivo");  //Return - Type :BOOL
return(valor);
}
```

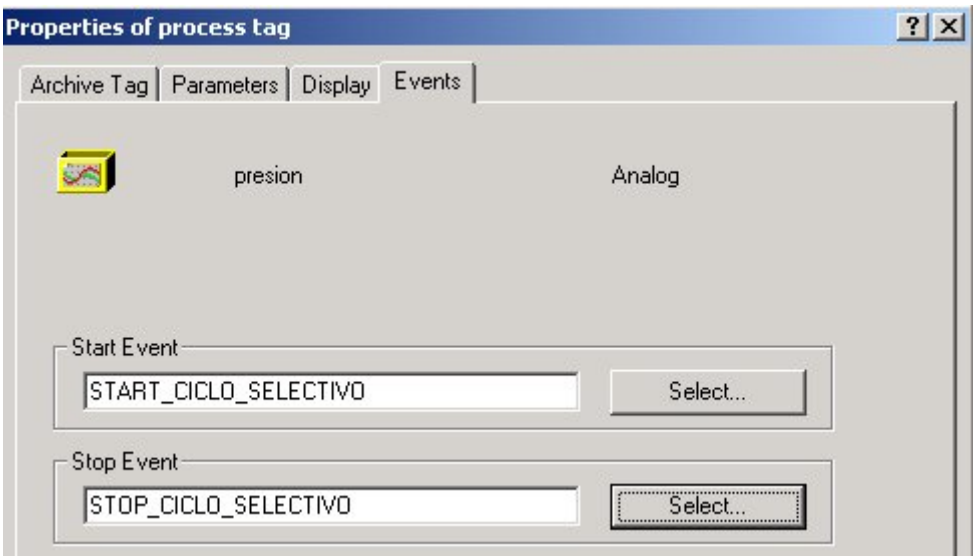
Y otra del tipo:

```
BOOL STOP_CICLO_SELECTIVO()
{

BOOL valor;

valor=GetTagBitWait("start_ciclo_selectivo");  //Return - Type :BOOL
return(!valor);
}
```

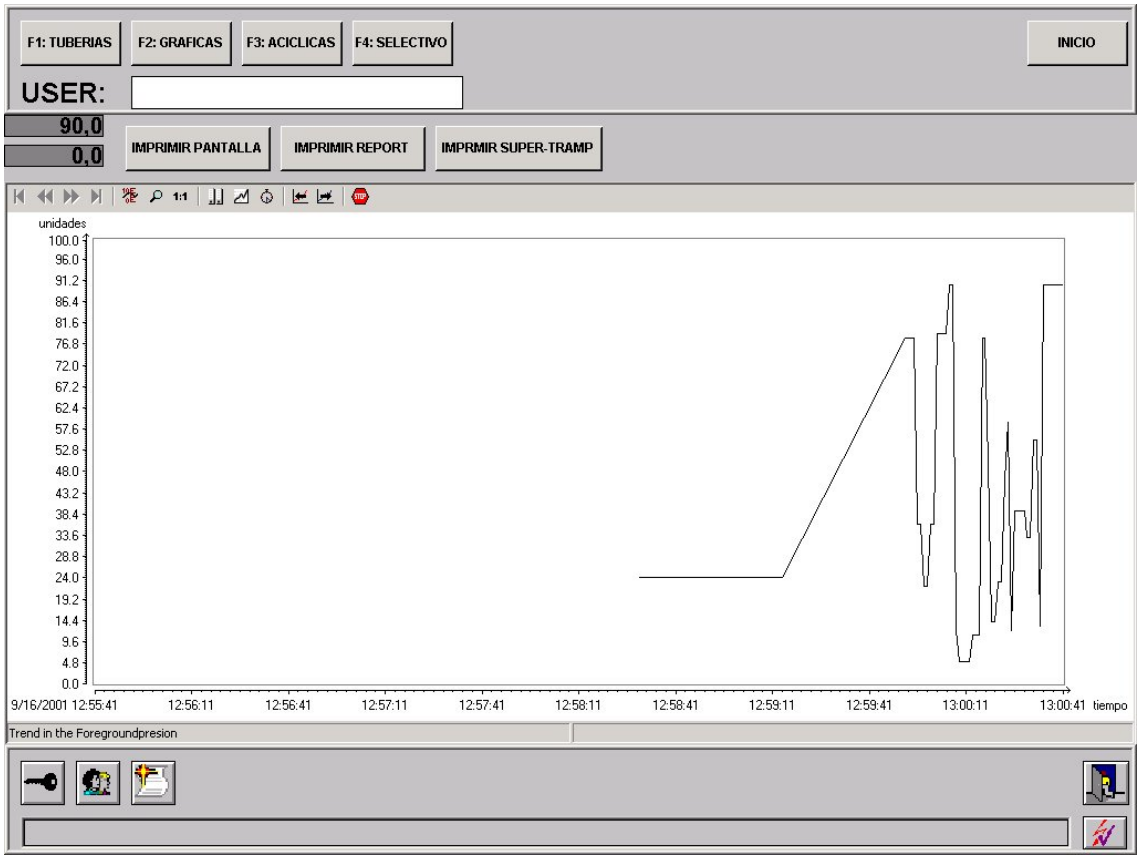
Una dará un true cuando esté el valor start_ciclo_selectivo, y la otra cuando no esté dicha variable.

Ahora vamos a tag logging y creamos un nuevo archivo, de nombre ciclo_selectivo, que posee una tendencia llamada presion, que almacena la presión cada segundo. Lo más importante es no olvidar que es de ciclo selectivo.

El tema está de nuevo en la última solapa, en la que asignamos las funciones anteriormente descritas al arranque de la gráfica y la parada de la misma. De esta manera, cuando esté la variable start_ciclo_selectivo arrancará la grabación del valor presión cada segundo, hasta que start_ciclo_selectivo valga false, en cuyo momento parará la grabación de los valores de presión.



Como antes, creamos una nueva picture, que llamaremos grasficas_ciclo_selectivo.pdl, y un botón en top, y blablabla.. el tema es que al final nos queda esto:



Almacenamiento upon change.

La última manera que nos queda por ver en el WinCC para almacenar los valores se le llama upon change. Básicamente es igual que la primera que vimos en el anterior capítulo , la de ciclo contínuo, salvo porque el tiemp ode adquisición de la variable está definido a 1 segundo. Solo se alamecena variable si cambia el valor de la misma. Por lo tanto, para la mayoría de los procesos este tipo de adquisición ahorrará espacio en el disco duro. No realizamos ejemplo, ya que es igual al del capítulo anterior, salvo por lo de upon change en lugar de cyclic-continuous.