Variables remanentes - RETAIN, PERSISTENT

Las variables remanentes pueden mantener su valor sobre el tiempo de ejecución habitual del programa. Estas se declaran como "Variables Retain" o aún más exigentes como "Variables persistentes". En cada caso se emplea un intervalo de memoria propio para la administración.

El tipo de declaración elegido determina el grado de "resistencia" de una variable remanente en caso de resets, descargas o un reboot (reinicio) del control. En la práctica ante todo estará solicitada la combinación de ambos tipos (véase bajo ↘ Variables persistentes).

	**¡INDICACIÓN!** Cuando se abre un **proyecto V2.3**, se mantienen inalteradamente efectivas las declaraciones de variables Retain, las declaraciones de variables persistentes sin embargo deben ser revisadas o bien creadas nuevamente, véase ↘ abajo: ¡se debe crear una lista propia de variables globales!

	**¡INDICACIÓN!** : No está permitido utilizar una " ↘ declaración AT" en combinación con "VAR RETAIN" o "VAR PERSISTENT"!

Vea a continuación:

↘ Variables Retain

↘ Variables persistentes

↘ Tabla resumen sobre el comportamiento

Variables Retain

Las variables declaradas como "Variables retain" se administran dependiendo del sistema de destino, pero típicamente en un **intervalo de memoria propio**. Estas se identifican en el proyecto CoDeSys en su declaración en un módulo o bien en una lista de variables globales con la palabra clave "**RETAIN**".

Ejemplo:

```
VAR RETAIN
 iRem1 : INT; (* 1ª variable Retain*)
END_VAR
```

Las variables Retain mantienen su valor tras una finalización incontrolada o tras un comando en línea ' ↘ Reset en caliente <Aplicación>', así como tras una desconexión y conexión (Reboot) normal del control. Ante un reinicio del programa se continúa trabajando con los valores memorizados. Todas las otras variables en este caso se reinicializan, bien sea con sus propios valores inicializados, o con las inicializaciones predeterminadas.

Ejemplo de aplicación: Un contador de unidades en la instalación de fabricación, que tras la interrupción de corriente debe continuar contando.

Las variables Retain sin embargo ante un ' ↘ Reset (origen)' y; en contrapartida a las variables persistentes; ante ' ↘ Reset (en frío)' y una nueva descarga del programa, se inicializan nuevamente.

La propiedad Retain puede ser combinada con propiedad Persistente. Vea para ello abajo la ↘ Tabla resumen.

¡INDICACIÓN!

Cuando una variable local está declarada en un **programa** como RETAIN, se memoriza **exactamente esta variable** en el rango Retain (como una variable Retain global).

Cuando una variable local está declarada en un **bloque de funciones** como RETAIN, la **completa instancia** de este bloque de funciones se memoriza en el rango Retain (todos los datos del módulo), en donde sin embargo, sólo la variable Retain declarada se trata como tal.

Cuando una variable local está declarada en una **función** como RETAIN, esto no tiene ningún efecto. ¡La variable no se memoriza en el rango Retain! ¡Cuando una variable local se declara en una función como PERSISTENT, esto permanece asimismo **sin ningún efecto**!

Variables persistentes

Las variables persistentes se identifican con la palabra clave "PERSISTENT" (**VAR_GLOBAL PERSISTENT**). Estas se reinicializan sólo ante ↘ Reset (origen) del control. A diferencia de las variables Retain estas mantienen su valor tras una descarga. Un ejemplo de aplicación para "variables Retain persistentes" sería un contador de horas de servicio, que tras una interrupción de corriente o descarga debe continuar contando. Véase abajo la ↘ Tabla resumen.

Variables persistentes se tratan de la siguiente manera; y con ello **diferente que en CoDeSys V2.3**:

Las variables persistentes SÓLO pueden ser declaradas en una **lista de variables globales especial** del tipo de objeto ' ↘ Variables persistentes', que pertenezca a una ↘ aplicación. Sólo hay UNA lista de estas por aplicación.

A partir de V3.3.0.1 una declaración con "VAR_GLOBAL PERSISTENT" provoca lo mismo que una declaración con "VAR_GLOBAL PERSISTENT RETAIN" o "VAR_GLOBAL RETAIN PERSISTENT".

Como las variables Retain, las variables persistentes se administran en un intervalo de memoria propio.

Ejemplo:

```
VAR GLOBAL PERSISTENT RETAIN
 iVarPers1 : DINT; (* 1ª variable Persistent+Retain App1 *)
bVarPers : BOOL;  (* 2ª variable Persistent+Retain App1 *)
END_VAR
```

	**¡INDICACIÓN!** Actualmente son sólo **posibles variables persistentes globales**.

Del sistema de destino se debe preparar por aplicación un **intervalo de memoria separado** para la lista de variables persistentes.

Ante cada nueva **carga de la aplicación** se compara la lista de variables persistentes sobre el control con la del proyecto. La lista de variables en el control se identifica en este caso entre otras también a través del nombre de la aplicación. En caso de **inconsistencias** se requiere al usuario antes de la descarga, una **reinicialización** de todas las variables persistentes. La inconsistencia se genera por ejemplo mediante renombrado o borrado u otra modificación de las declaraciones de variables persistentes existentes.

	**¡INDICACIÓN!** ¡Entonces, **cada modificación en la parte de declaración de la lista de variables persistentes y el efecto de la reinicialización, que se consulta a raíz de ello, debería ser analizada cuidadosamente**!

Nuevas declaraciones pueden ser agregadas sólo al final de la lista, sin embargo al cargar se reconocen como nuevas y no requieren una reinicialización de la lista completa.

Tabla resumen sobre el comportamiento de variables remanentes

x = se mantiene el valor - = el valor se inicializa nuevamente

tras comando en línea	VAR	VAR RETAIN	VAR PERSISTENT VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Reset en caliente <aplicación>	-	x	x
Reset en frío <aplicación>	-	-	x
Reset origen <aplicación>	-	-	-
Cargar(=Descargar) <aplicación>	-	-	x
Cambio en línea <aplicación>	x	x	x
Reboot control	-	x	x