

# Control FPWIN Pro

## Software Según el Estándar IEC 61131-3

|                |                          |
|----------------|--------------------------|
| <b>Título</b>  | <b>Definición de DUT</b> |
| <b>Versión</b> | <b>1.000</b>             |

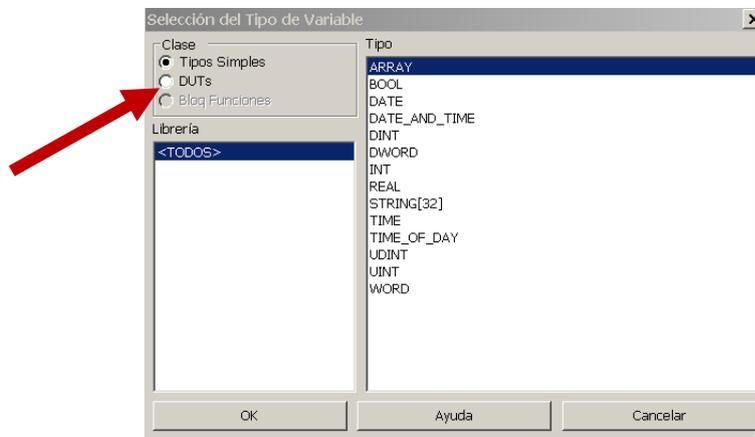
### Definición de DUT

El nombre DUT viene del inglés “Data Unit Type”, lo que se podría definir en castellano como “Estructura de Datos”.

Muchas funciones de alto nivel de Panasonic (PID, Control de servomotores, PWM, ...) requieren el uso de variables del tipo DUT. Así mismo el conocerlas simplifica en muchas ocasiones la programación de una aplicación por lo que es muy recomendable estudiarlas a fondo.

Una estructura de datos no es más que la definición de diferentes variables (del mismo o diferente tipo) dentro de un mismo nombre genérico.

Posteriormente se podrán definir variables globales o locales del tipo DUT



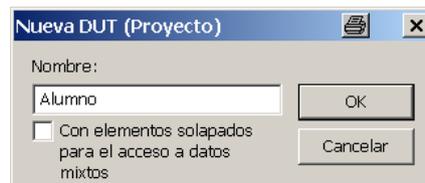
Las variables del tipo DUT normalmente se utilizan a lo largo de un programa con la siguiente notación:

`nombre_variable_externa.nombre_variable_interna`

Solamente se utiliza la notación `nombre_variable_externa` cuando la función así lo requiera (como por ejemplo a la entrada de la función F173)

En un aula de 30 alumnos, para cada alumno el profesor desea saber 5 datos (nombre, apellido, edad, curso y nota media).

Para crear una nueva DUT seleccionar el apartado DUT del navegador del proyecto y con el botón derecho del ratón seleccionar “Nueva DUT”. Asignarle el nombre “Alumno”.



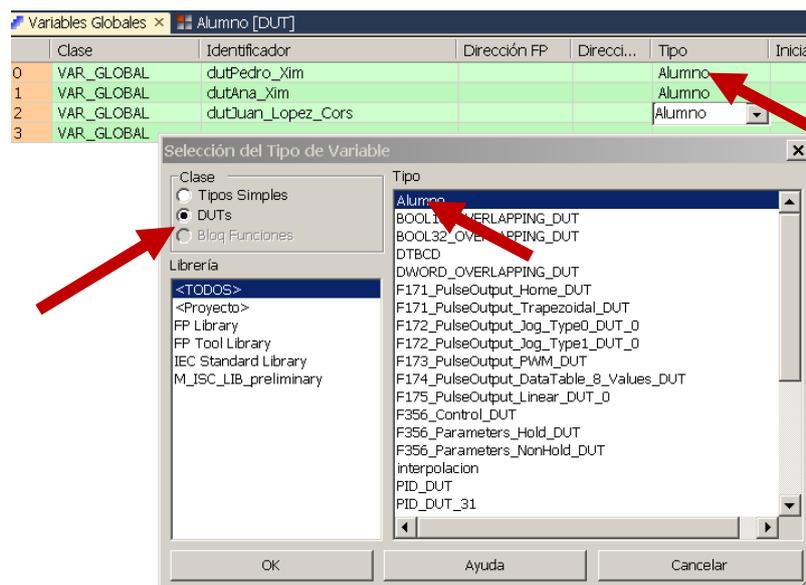
En la zona de edición se abrirá la ventana de definición de variables internas de la DUT que tiene la misma apariencia que la cabecera de las POUs.

Definir los 5 datos indicados anteriormente:

|   | Identificador | Tipo       | Inicial | Comentario              |
|---|---------------|------------|---------|-------------------------|
| 0 | sNombre       | STRING[32] | "       | Nombre del Alumno       |
| 1 | sApellidos    | STRING[32] | "       | 1er apellido            |
| 2 | iEdad         | INT        | 0       | Años                    |
| 3 | iCurso        | INT        | 0       | Curso actual del alumno |
| 4 | rNota         | Real       | 0.0     | Nota de la asignatura   |
| 5 |               |            |         |                         |

A continuación cerrar y guardar la DUT.

Definir en variables globales los 30 alumnos del aula. Cada uno de ellos será del tipo DUT “Alumno”:



Programar con la notación anteriormente citada

|   |  |
|---|--|
| 1 | · · · 'Ana' — MOVE — dutAna_Xim.sNombre · · ·    |
| 2 | · · · 'Xim' — MOVE — dutAna_Xim.sApellidos · · · |
| 3 | · · · 13 — MOVE — dutAna_Xim.iEdad · · ·         |
| 4 | · · · 9.98 — MOVE — dutAna_Xim.rNota · · ·       |

**Beneficios aportados:** En este ejemplo, el beneficio aportado es clarificar la ejecución del programa y definir muchas menos variables  $5 + 30 = 35$  (en lugar de las  $5 \cdot 30 = 150$  que tendría que definir si no existiesen las DUTs).

## DUTs de Interés

Existen muchas DUTs predefinidas que facilitan enormemente la programación. Tal es el caso de todas instrucciones para el control de salida de pulsos o PID. Se recomienda revisar la lista de DUTs predefinidas prestando especial atención a todas las DUTs del tipo Overlapping (solapamiento)

Estas DUTs son muy utilizadas cuando se requiere acceder a una misma área de memoria (ej DDT10) y visualizar el contenido en diferentes formatos (ej. Real, INT parte baja, INT parte alta, ...). Existen infinidad de aplicaciones en la que codificar la información es vital para ahorrar recursos del sistema y sin duda que las DUTs Overlapping son de gran ayuda para este fin.

## Sobre Este Documento

Este documento no tiene carácter oficial ni se podrá responsabilizar a Panasonic Electric Works España por las erratas o información errónea contenida en el mismo, declinando toda responsabilidad por su utilización.