

Uso de punteros para variables

Somachine 4.1 (SP1)

Autor: César Rufo Domínguez

Versión	Autor	Fecha	Comentarios
V 1.0	César Rufo	12/5/15	

Para acceder a determinadas variables (elemento concreto dentro de un array, direccionamientos indirectos y bit concreto dentro de una palabra de sistema, entre otros) necesitamos manejar instrucciones de tipo **POINTER TO** y **ADR**.

Veamos un ejemplo y comentaremos su uso: (en verde: declaraciones de variables)

```
PROGRAM CONTROL_LED
VAR
    hideETHError: BOOL;
    pLw: POINTER TO LWORD;
END_VAR
```

```
IF hideETHError THEN
    pLw := ADR(PLC_R.i_lwSystemFault_1);
    pLw^ := PLC_R.i_lwSystemFault_1 OR 16#8; // bit 4 (corresponde a ETH2) se fuerza a TRUE
END_IF
```

La instrucción ADR devuelve la dirección del objeto, no su valor. Puesto que la variable entre paréntesis es de tipo LWORD en este caso, la declaración inicial del puntero es POINTER TO LWORD también pero el puntero pLw es por el contrario un valor DWORD.

Para trabajar con los valores es necesario “desreferenciar” el puntero. Eso se realiza añadiendo al nombre de la variable el símbolo circunflejo “^”. Gracias a esta desreferenciación es posible escribir en un objeto de lectura: el valor resultado del OR lógico se escribe en una variable a la que se apunta por la dirección obtenida antes.

Otro ejemplo con enteros:

```
pt:POINTER TO INT;
var_int1:INT;
var_int2:INT;
```

```
pt := ADR(var_int1);
var_int2:=pt^;
```

En este caso parece más sencillo: se apunta a la variable operada por la instrucción ADR y el valor que tenga en ese momento la variable var_int1 en este caso se copia al desreferenciar sobre var_int2. Aquí el puntero es POINTER TO INT porque la variable entre paréntesis de ADR también lo es.

Un ejemplo aplicado a un array o matriz:

```
MADRIZ: ARRAY [0..89] OF DINT;  
pl: POINTER TO INT;  
PUNTERO: INT :=3;
```

```
pl:= ADR(PUNTERO);  
MADRIZ[pl^]:=MADRIZ[pl^-1];
```

Por el valor inicial en 3 del puntero, se copiará el valor de MADRIZ[2] sobre el valor de MADRIZ[3]. Si en el programa se genera una evolución del valor del puntero se provocará el desplazamiento de los elementos dentro del array.

Para comprobar el acceso al puntero durante el tiempo de ejecución, se puede utilizar la función de comprobación disponible de forma implícita CheckPointer. Se llama antes de cada acceso a la dirección de un puntero. Para conseguirlo, hay que añadir el objeto POU para comprobaciones implícitas a la aplicación. Para ello, marque la casilla de verificación relacionada con el tipo CheckPointer, seleccione un lenguaje de implementación y confirme su configuración haciendo clic en Abrir:

