

Apéndice A

A1 Programas en CoDeSys2.3 VARIABLES GLOBALES

VAR_GLOBAL

(* Binary inputs *)

sen_T1_full AT %IX0.0: BOOL;

sen_T1_halfull AT %IX0.1: BOOL;

sen_T1_empty AT %IX0.2: BOOL;

sen_T2_full AT %IX0.3: BOOL;

sen_T2_halfull AT %IX0.4: BOOL;

sen_T2_empty AT %IX0.5: BOOL;

sen_mix_full AT %IX0.6: BOOL;

sen_mix_empty AT %IX0.7: BOOL;

(* Analog inputs *)

level_TANK1_INT AT %IW2: UINT;

level_TANK2_INT AT %IW4: UINT;

level_Mix_INT AT %IW6: UINT;

(* Binay outputs to System - Aktuators *)

V_T1_to_T2a AT %QX0.0: BOOL;

V_T1_to_T2b AT %QX0.3: BOOL;

V_T1_to_Mix AT %QX0.1: BOOL;

V_fill_T1 AT %QX1.1: BOOL;

V_T2_to_T1a AT %QX0.2: BOOL;

V_T2_to_T1b AT %QX1.2: BOOL;

V_T2_to_Mix AT %QX0.4: BOOL;

V_Evac_Mix AT %QX0.5: BOOL;

Pumpe1 AT %QX0.6: BOOL;

Pumpe2 AT %QX0.7: BOOL;

V_Mix_T1 AT %QX1.0: BOOL;

V_Mix_T2 AT %QX1.3: BOOL;

(* dates to Control Panel *)

level_TANK1_REAL: REAL;

level_TANK2_REAL: REAL;

level_Mix_REAL: REAL;

pumpe1_level AT %QW2: INT;

pumpe2_level AT %QW4: INT;

air1 AT %QX1.4: BOOL;

air2 AT %QX1.5: BOOL;

air3 AT %QX1.6: BOOL;

vo_warnung: BOOL;

vo_strWarnung: STRING;

vi_resetWarnung: BOOL;

END_VAR

/* Sensor que informa Tanque 1 lleno

/* Sensor que informa Tanque 1 semi-lleño

/* Sensor que informa Tanque 1 vacío

/* Sensor que informa Tanque 2 lleno

/* Sensor que informa Tanque 2 semi-lleño

/* Sensor que informa Tanque 2 vacío

/* Sensor que informa Tanque mezcla lleno

/* Sensor que informa Tanque mezcla vacío

/* Nivel del tanque1

/* Nivel del tanque2

/* Nivel del tanque de mezcla

/* Apertura/Cierre, válvula 1 de T1 a T2

/* Apertura/Cierre, válvula mix de T1 a T2

/* Apertura/Cierre, válvula de T1 a mezcla

/* Apertura/Cierre, válvula para llenar T1

/* Apertura/Cierre, válvula de T2 a T1

/* Apertura/Cierre, válvula mix T2 a T1

/* Apertura/Cierre, válvula de T2 a mezcla

/* Apertura/Cierre, válvula vaciado mezcla

/* Encendido/Apagado, bomba 1

/* Encendido/Apagado, bomba 2

/* Apertura/Cierre, válvula de mezcla a T1

/* Apertura/Cierre, válvula de mezcla a T2

/* nivel de T1 en unidades [0..1]

/* nivel de T2 en unidades [0..1]

/* nivel de mezcla en unidades [0..1]

/* nivel de bombeo B1 en unidades [0..1]

/* nivel de bombeo B2 en unidades [0..1]

/* aire1

/* aire2

/* aire3

/* variable de alarma

/* mensaje de alarma

/* variable de reset

MAIN FUNCTION (PLC_PRG)

PROGRAM PLC_PRG (PRG-ST)

VAR

hand: BOOL;

auto: BOOL;

ak_T1_T2a AT %QX0.0: BOOL;

ak_T1_T2b AT %QX0.3: BOOL;

ak_T1_Mix AT %QX0.1: BOOL;

ak_Fill_T1 AT %QX1.1: BOOL;

ak_T2_T1a AT %QX0.2: BOOL;

ak_T2_T1b AT %QX1.2: BOOL;

ak_T2_mix AT %QX0.4: BOOL;

ak_Mix_evac AT %QX0.5: BOOL;

ak_Mix_T1 AT %QX1.0:BOOL;

ak_Pumpe1 AT %QX0.6: BOOL;

ak_Pumpe2 AT %QX0.7: BOOL;

pumpe1_level AT %QW2: INT;

pumpe2_level AT %QW4: INT;

FLAG: BOOL:=TRUE;

END_VAR

ak_Mix_T1:=FALSE;

IF (auto)=TRUE THEN /*Modo automático

auto:=TRUE;

FLAG:=TRUE;

Practica();

ELSE IF (hand)=TRUE THEN /*Modo manual

hand:=TRUE;

IF (FLAG) = TRUE THEN

ak_Fill_T1:=FALSE; /*Inicialización de variables

ak_T1_T2a:=FALSE;

ak_T2_T1a:=FALSE;

ak_T1_T2b:=FALSE;

ak_T2_T1b:=FALSE;

ak_T2_Mix:=FALSE;

ak_Mix_evac:=FALSE;

ak_Pumpe1:=FALSE;

ak_Pumpe2:=FALSE;

FLAG:=FALSE;

ELSE;END_IF

handcontrol();

ELSE

ak_Fill_T1:=FALSE;

ak_T1_T2a:=FALSE;

ak_T2_T1a:=FALSE;

ak_T1_T2b:=FALSE;

ak_T2_T1b:=FALSE;

ak_T2_Mix:=FALSE;

ak_Mix_evac:=FALSE;

```

ak_Pumpe1:=FALSE;
ak_Pumpe2:=FALSE;;
END_IF
END_IF
conversion();
35

```

4.1.3 AUTOCONTROL (Practica)

PROGRAM Practica

VAR

```

sen_T1_full AT %IX0.0: BOOL;
sen_T1_half AT %IX0.1: BOOL;
sen_T1_empty AT %IX0.2: BOOL;
sen_T2_full AT %IX0.3: BOOL;
sen_T2_half AT %IX0.4: BOOL;
sen_T2_empty AT %IX0.5: BOOL;
sen_Mix_full AT %IX0.6: BOOL;
ak_T1_T2a AT %QX0.0: BOOL;
ak_T1_T2b AT %QX0.3: BOOL;
ak_T1_Mix AT %QX0.1: BOOL;
ak_Fill_T1 AT %QX1.1: BOOL;
ak_T2_T1a AT %QX0.2: BOOL;
ak_T2_T1b AT %QX1.2: BOOL;
ak_T2_mix AT %QX0.4: BOOL;
ak_Mix_evac AT %QX0.5: BOOL;
ak_Mix_T1 AT %QX1.0:BOOL;
ak_Pumpe1 AT %QX0.6: BOOL;
ak_Pumpe2 AT %QX0.7: BOOL;
reset: BOOL:=FALSE;
FULLEN: BOOL:=TRUE;
LEEREN: BOOL:=FALSE;
FLAG: BOOL:=FALSE;
ALLE: BOOL:=FALSE;
MIX1: BOOL :=FALSE;

```

36

```

TRANS: BOOL:=FALSE;
CAR: BOOL;
END_VAR

```

```

pumpe1_level:= pumpe1_level+30000;
WHILE (pumpe1_level) < 0 DO
pumpe1_level:=pumpe1_level-100;
END_WHILE
pumpe2_level:= pumpe2_level+30000;
WHILE (pumpe2_level) < 0 DO
pumpe2_level:=pumpe2_level-100;
END_WHILE

```

IF (FULLEN)=TRUE THEN

/*Llenado Tanque 1

IF (sen_T1_full) =FALSE THEN

FULLEN:=TRUE;

LEEREN:=FALSE;

```

        ak_Fill_T1:=TRUE;
        ak_T1_T2a:=FALSE;
        ak_T2_T1a:=FALSE;
        ak_T1_T2b:=FALSE;
        ak_T2_T1b:=FALSE;
        ak_T2_Mix:=FALSE;
        ak_T1_Mix:=FALSE;
        ak_Mix_evac:=FALSE;
        ak_Pumpe1:=FALSE;
        ak_Pumpe2:=FALSE;
ELSE
    LEEREN:=TRUE;
    FULLEN:=FALSE;
37
    ak_Fill_T1:=FALSE;
    ak_T1_T2a:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_T2_Mix:=FALSE;
    ak_T1_Mix:=FALSE;
    ak_Mix_evac:=FALSE;
    ak_Pumpe1:=FALSE;
    ak_Pumpe2:=FALSE;
END_IF
ELSE;
END_IF
IF (LEEREN)=TRUE THEN
    IF (sen_T1_empty) =FALSE THEN
        FULLEN:=FALSE;
        LEEREN:=TRUE;
        IF (sen_T2_full)=FALSE THEN    /*Activar Bomba1
            ak_T1_T2a:=TRUE;
            ak_T1_T2b:=TRUE;
            ak_T2_T1a:=FALSE;
            ak_T2_T1b:=FALSE;
            ak_Fill_T1:=FALSE;
            ak_T2_Mix:=FALSE;
            ak_T1_Mix:=FALSE;
            ak_Mix_evac:=FALSE;
            ak_Pumpe1:=TRUE;
            ak_Pumpe2:=FALSE;
        ELSE
            FLAG:=1;
        END_IF
    ELSE
38        FLAG:=1;
    END_IF
END_IF

```

```

IF (FLAG)=TRUE THEN
    ak_Pumpe1:=FALSE;
    ak_Pumpe2:=TRUE;           /*Activar bomba 2
    ak_T1_T2a:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_Fill_T1:=FALSE;
    ak_T2_Mix:=TRUE;
    ak_T1_Mix:=FALSE;
    ak_Mix_evac:=FALSE;
    IF (sen_Mix_full)=TRUE THEN
        CAR:=TRUE;
        ALLE:=FALSE;
        FLAG:=FALSE;
    ELSE;END_IF
ELSE;END_IF

```

```

IF(CAR)=TRUE THEN           /*Evacuacion de mezcla
    ak_T1_T2a:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_T2_Mix:=FALSE;
    ak_Mix_evac:=TRUE;
    ak_Pumpe1:=FALSE;
    ak_Pumpe2:=FALSE;

```

39

```

    IF (sen_Mix_empty)=TRUE THEN
        ak_Mix_evac:=FALSE;
        CAR:=FALSE;
        MIX1:=TRUE;
        ALLE:=FALSE;
        LEEREN:=FALSE;
        FLAG:=FALSE;
        TRANS:=TRUE;
    ELSE;END_IF
END_IF

```

```

IF (MIX1) = TRUE THEN       /*Trasbase de líquido de T1 a mezcla
    ak_T1_T2a:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_T2_Mix:=FALSE;
    ak_T1_Mix:=TRUE;
    ak_Mix_evac:=FALSE;
    ak_Pumpe1:=TRUE;
    ak_Pumpe2:=FALSE;
    IF (sen_Mix_full)=TRUE THEN

```

```

        ALLE:=TRUE;
        MIX1:=FALSE;
    ELSE IF(sen_T2_empty)=TRUE THEN
        ALLE:=TRUE;
        MIX1:=FALSE;
    ELSE;END_IF
END_IF END_IF

```

```

IF (ALLE)=TRUE THEN

```

```

40

```

```

    ak_T1_T2a:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_T2_Mix:=FALSE;
    ak_T1_Mix:=FALSE;
    ak_Mix_evac:=TRUE;
    ak_Pumpe1:=FALSE;
    ak_Pumpe2:=FALSE;
    IF (sen_Mix_empty)=TRUE THEN
        ak_Mix_evac:=FALSE;
        FULLEN:=TRUE;
        ALLE:=FALSE;
        MIX1:=FALSE;
        LEEREN:=FALSE;
        FLAG:=FALSE;
        TRANS:=TRUE;
    ELSE;END_IF
ELSE;END_IF

```

```

IF (reset)=TRUE THEN

```

```

/*Resetear sistema

```

```

    ak_T1_T2a:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_T2_Mix:=FALSE;
    ak_T1_Mix:=FALSE;
    ak_Mix_evac:=FALSE;
    ak_Pumpe1:=FALSE;
    ak_Pumpe2:=FALSE;
    FULLEN:=TRUE;

```

```

ELSE;END_IF

```

```

41

```

HANDCONTROL

```

PROGRAM handcontrol

```

```

/*Programa para en control manual

```

```

VAR

```

```

ak_T1_T2a AT %QX0.0: BOOL;

```

```

ak_T1_T2b AT %QX0.3: BOOL;

```

```

ak_T1_Mix AT %QX0.1: BOOL;
ak_Fill_T1 AT %QX1.1: BOOL;
ak_T2_T1a AT %QX0.2: BOOL;
ak_T2_T1b AT %QX1.2: BOOL;
ak_T2_mix AT %QX0.4: BOOL;
ak_Mix_evac AT %QX0.5: BOOL;
sen_T1_full AT %IX0.0: BOOL;
sen_T1_empty AT %IX0.2: BOOL;
sen_T2_full AT %IX0.3: BOOL;
sen_T2_empty AT %IX0.5: BOOL;
sen_Mix_full AT %IX0.6: BOOL;
sen_Mix_empty AT %IX0.7: BOOL;
ak_Mix_T1 AT %QX1.0:BOOL;
ak_Pumpe1 AT %QX0.6: BOOL;
ak_Pumpe2 AT %QX0.7: BOOL;
hand: BOOL;
p1_level AT %QW2: INT;
p2_level AT %QW4: INT;

```

```

pumpe1mas: BOOL :=FALSE;
pumpe1menos: BOOL :=FALSE;
pumpe2mas: BOOL :=FALSE;

```

42

```

pumpe2menos: BOOL :=FALSE;
FLAG: BOOL :=FALSE;
RESET: BOOL;
akt_T1_T2: BOOL;
akt_T2_T1: BOOL;
END_VAR
hand:=TRUE;

```

```

IF (FLAG)=TRUE THEN                                /*Iniciación de variables

```

```

    ak_T1_T2a:=FALSE;
    ak_T1_T2b:=FALSE;
    ak_pumpe1:=FALSE;
    ak_Fill_T1:=FALSE;
    ak_T1_Mix:=FALSE;
    ak_T2_T1a:=FALSE;
    ak_T2_T1b:=FALSE;
    ak_T2_Mix:=FALSE;
    ak_Mix_evac:=FALSE;
    ak_pumpe2:=FALSE;
    FLAG:=FALSE;

```

```

ELSE;END_IF

```

```

IF (akt_T1_T2)=TRUE THEN                            /*trasbase t1 a t2
    IF (sen_T1_empty) = FALSE AND (sen_T2_full) =FALSE THEN
        ak_T1_T2a:=TRUE;
        ak_T1_T2b:=TRUE;
        ak_pumpe1:=TRUE;

```

```

        ak_Fill_T1:=FALSE;
        ak_T1_Mix:=FALSE;
        ak_T2_T1a:=FALSE;
        ak_T2_T1b:=FALSE;
        ak_T2_Mix:=FALSE;
43
        ak_Mix_evac:=FALSE;
        ak_pumpe2:=FALSE;
        FLAG:=FALSE;
    ELSE
        FLAG:=TRUE;
    END_IF
ELSE;END_IF

IF (ak_T1_Mix)=TRUE THEN          /*Trasbase de T1 a mezcla
    IF (sen_T1_empty) = FALSE AND (sen_Mix_full) =FALSE THEN
        ak_T1_T2a:=FALSE;
        ak_T1_T2b:=FALSE;
        ak_pumpe1:=TRUE;
        ak_Fill_T1:=FALSE;
        ak_T1_Mix:=TRUE;
        ak_T2_T1a:=FALSE;
        ak_T2_T1b:=FALSE;
        ak_T2_Mix:=FALSE;
        ak_Mix_evac:=FALSE;
        ak_pumpe2:=FALSE;
        FLAG:=FALSE;
    ELSE
        FLAG:=TRUE;
    END_IF
ELSE;END_IF

IF (akt_T2_T1)=TRUE THEN          /*Trasbase de T2 a T1
    IF (sen_T2_empty) = FALSE AND (sen_T1_full) =FALSE THEN
        ak_T1_T2a:=FALSE;
        ak_T1_T2b:=FALSE;
        ak_pumpe1:=FALSE;
        ak_Fill_T1:=FALSE;
44
        ak_T1_Mix:=FALSE;
        ak_T2_T1a:=TRUE;
        ak_T2_T1b:=TRUE;
        ak_T2_Mix:=FALSE;
        ak_Mix_evac:=FALSE;
        ak_pumpe2:=TRUE;
        FLAG:=FALSE;
    ELSE
        FLAG:=TRUE;
    END_IF
ELSE;END_IF

```

```

IF (ak_T2_mix)=TRUE THEN                                /*Trasbase de T2 a mezcla
    IF (sen_T2_empty) =FALSE AND (sen_Mix_full) =FALSE THEN
        ak_T1_T2a:=FALSE;
        ak_T1_T2b:=FALSE;
        ak_pumpe1:=FALSE;
        ak_Fill_T1:=FALSE;
        ak_T1_Mix:=FALSE;
        ak_T2_T1a:=FALSE;
        ak_T2_T1b:=FALSE;
        ak_T2_Mix:=TRUE;
        ak_Mix_evac:=FALSE;
        ak_pumpe2:=TRUE;
        FLAG:=FALSE;
    ELSE
        FLAG:=TRUE;
    END_IF
ELSE;END_IF

```

```

IF (ak_Mix_evac)=TRUE THEN                                /*Evacuación de mezcla
    IF (sen_Mix_empty)=FALSE THEN
        ak_T1_T2a:=FALSE;
        ak_T1_T2b:=FALSE;
        ak_pumpe1:=FALSE;
        ak_Fill_T1:=FALSE;
        ak_T1_Mix:=FALSE;
        ak_T2_T1a:=FALSE;
        ak_T2_T1b:=FALSE;
        ak_T2_Mix:=FALSE;
        ak_Mix_evac:=TRUE;
        ak_pumpe2:=FALSE;
        FLAG:=FALSE;
    ELSE
        FLAG:=TRUE;
    END_IF
ELSE;END_IF

```

```

IF (pumpe1mas) = TRUE THEN                                /*Manejo de niveles del bombeo
    p1_level:=p1_level+5000;
    WHILE (p1_level) < 0 DO
        p1_level:=p1_level-100;
    END_WHILE
    pumpe1mas:=FALSE;
ELSE;END_IF

```

```

IF (pumpe1menos) = TRUE THEN
    p1_level:=p1_level-5000;
    WHILE (p1_level) < 0 DO
        p1_level:=p1_level+100;
    END_WHILE

```

```

        pumpe1menos:=FALSE;
ELSE;END_IF

IF (pumpe2mas) = TRUE THEN
    p2_level:=p2_level+5000;
    WHILE (p2_level) < 0 DO
        p2_level:=p2_level-100;
    END_WHILE
    pumpe2mas:=FALSE;
ELSE;END_IF

IF (pumpe2menos) = TRUE THEN
    p2_level:=p2_level-5000;
    WHILE (p2_level) < 0 DO
        p2_level:=p2_level+100;
    END_WHILE
    pumpe2menos:=FALSE;
ELSE;END_IF

IF (RESET) = TRUE THEN
    FLAG:=TRUE;
    RESET:=FALSE;
ELSE;END_IF

```

CONVERSION

```

PROGRAM conversion                                /*Programa de conversión de unidades
VAR
level_T1_INT AT %IW2: UINT;
level_T2_INT AT %IW4: UINT;
level_M_INT AT %IW6: UINT;
END_VAR
*****
Level_TANK1_REAL := (level_T1_INT- 0.2*27648) / (0.8*27648);
Level_TANK2_REAL := (level_T2_INT- 0.2*27648) / (0.8*27648);
Level_Mix_REAL := (level_M_INT- 0.2*27648) / (0.8*27648);

```

PERIFERIA DESCENTRALIZADA

INTRODUCCIÓN

En el mundo industrial, ocurre con frecuencia que el PLC de control está colocado a una distancia tal que, el llevar todos los cables de las entradas y salidas hasta los órganos de control y accionamiento de la máquina, exigirían un mazo de cables de unas proporciones elevadas, con los consiguientes problemas de tendido de los mismos, espacio, pérdidas de señal, costo económico de material y colocación, etc. Para evitar estos problemas, los fabricantes de PLCs y otros componentes industriales, proporcionan dentro de su gama de productos, la posibilidad de enlazar el PLC y los elementos de la máquina mediante un bus de datos, el cual con sólo dos o cuatro hilos (según el tipo de bus), transmite la información de forma rápida y eficaz. Lógicamente, es necesario colocar a pie de máquina un módulo que centralice la conexión de todos los controles y accionamientos, que enlace (mediante el bus) con el PLC y que procese la información recibida para transmitirla al PLC. Estos módulos pueden ser de entradas/salidas descentralizadas, válvulas inteligentes o elementos similares. Su característica principal es la alta velocidad de comunicación, de tal forma que, prácticamente no influye en el tiempo total de Scan del PLC (aumenta con valores inferiores a un 5 %)

MÓDULOS DE ENTRADAS – SALIDAS DESCENTRALIZADAS.

Actualmente existen productos adaptados a todas las necesidades: entradas salidas digitales y analógicas, temporizadores, contadores, etc. Incluso existen productos dotados de CPU propia, mediante la cual pueden ser maestros de otros módulos de E/S descentralizada. Dependiendo del fabricante, se pueden incluir en todo tipo de redes (Profibus, DeviceNet, etc.)

Los módulos utilizados en la Célula del centro son del tipo Siemens ET200M. Se han colocado dos de estos módulos integrados directamente en la línea principal de Profibus. Tienen su propia dirección profibus y están montados en dos armarios colocados a ambos lados de la cinta, junto a los puestos mas alejados del PLC de control de la propia cinta. A estos módulos se les han conectado módulos de entradas/salidas digitales estándar, y el tratamiento que les da el PLC es exactamente igual que si estuvieran conectadas en el mismo rack de la CPU. Sus principales características son:

- Admite la conexión de todos los módulos de E/S del S7-300
- Configuración (en profibus) de los módulos definible por el usuario, direccionándolos como E/S del propio PLC
- Velocidad de transmisión de hasta 12 Mbit/s.
- Conexiones a un solo hilo
- Módulos con bus activo (se pueden conectar y desconectar con el sistema funcionando)

Otros ejemplos de módulos de E/S descentralizadas son:



Módulos de E/S normales conectados al ET200M

ET 200 S de Siemens.

- Máximo de 63 módulos conectables
- Módulos de entradas/salidas analógicas
- Módulos de contador
- Módulos arrancadores de motores
- Puerto comunicaciones RS232

Dentro de las posibilidades de éste módulo, existe la IM151 que funciona como CPU de sus propias entradas y salidas descentralizadas. De sus características, son de resaltar:

- Módulos de programación propios, FBs, FCs, DBs (128 de cada)
- 2048 marcas internas
- 128 temporizadores
- 64 contadores de impulsos
- 1536 Bytes E/A de direccionamiento lógico
- Memori card de 2MB
- Memoria integrada de 40 Kb (para carga) y 24 KB (para trabajo).
- Módulos de E/S digitales
- Módulos de E/S analógicos
- Módulo contador para encoder incremental a 100 KHz
- Módulo contador para encoder absoluto
- Interface serie RS232 – RS422/485



Módulos de E/S ET200S de Siemens



Módulo CPU -- IM151

El módulo ET200X está especialmente pensado para colocación en máquina sin necesidad de armario, lo que reduce de forma importante costos, dentro de sus características principales destacan:

- 7 módulos de expansión.
- Módulo de enlace ASI
- Arrancadores de motor mecánicos y electrónicos de hasta 5,5 Kw
- Módulos neumáticos (preparados para material FESTO)
- Programable vía Profibus DP



Módulo neumático

ET200B de tamaño compacto con interfase Profibus DP integrado. Características principales:

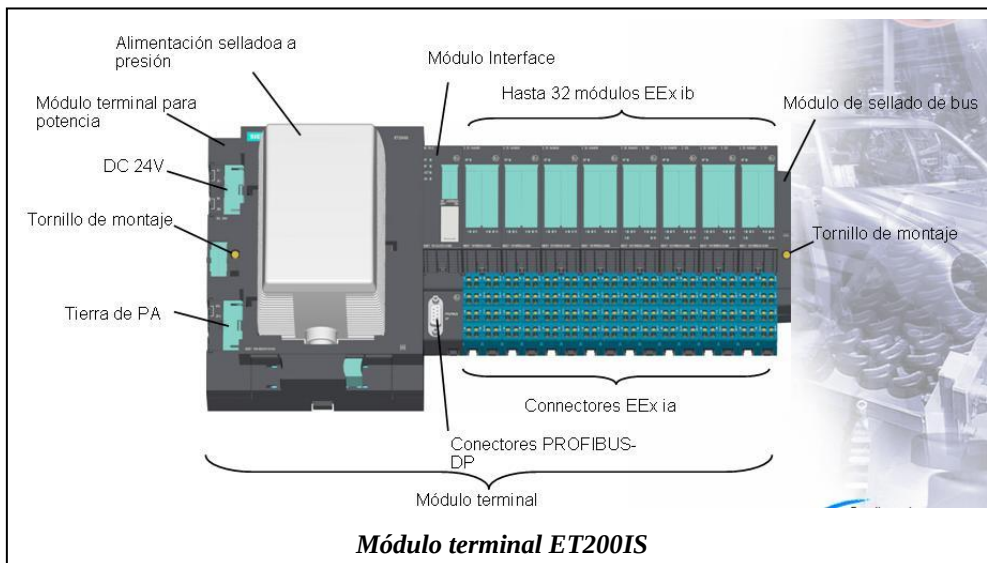
- Diferentes versiones: Entradas digitales, Salidas digitales, E/S digitales, en CC y AC, varias tensiones, etc.
- Velocidad de transmisión de hasta 12 Mbit/s
- Cableado a tres o cuatro hilos.
- Fijación de los cables por tornillo o resorte.



Módulo compacto ET200B

ET200IS con módulos de E/S remotas para zonas con riesgo de explosión.

- Modular (máximo 32 módulos)
- Montaje en bastidor estándar
- Certificaciones de todo tipo (Cenelec, FM, UL, CSA, Instrumentación ANSI/ISA, montaje en barcos, etc.)



Para configurar una unidad de E/S descentralizadas en Profibús, desde configurar Hardware, buscar en la librería la unidad de E/S correspondiente (en la imagen una ET200M), y colocarla como esclava del maestro DP (X2), a continuación, direccional las entradas y salidas (se pueden cambiar las que son colocadas por defecto)

Configuración de hardware en TIA Portal. El rack principal contiene:

- 1 PS 307 5A
- 2 CPU 315-2 DP (Maestro DP)
- 3
- 4 DI16xDC24V
- 5 DI16xDC24V
- 6 DO16xDC24V/0.5A
- 7 DO16xDC24V/0.5A
- 8
- 9
- 10
- 11

Se muestra una conexión a un dispositivo (3) PER-01.

Configuración de los módulos de E/S (3) PER-01:

Slot	Módulo	Referencia	Dirección E	Dirección S	Comentario
4	DI16xDC24V	6ES7 321-1BH01-0AA0	16...17		
5	DI16xDC24V	6ES7 321-1BH01-0AA0	20...21		
6	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0		24...25	
7					

ET200M como E/S descentralizadas de un PLC con CPU 315-2 DP conectada vía Profibús

**Manual de formación
para soluciones generales en automatización
Totally Integrated Automation (T I A)**

MÓDULO A4

Programación de la CPU 315-2 DP

Este documento fue suministrado por SIEMENS Siemens A&D SCE (Tecnología en Automatización y Accionamientos, Siemens A&D, coopera con la Educación) para formación. Siemens no hace ningún tipo de garantía con respecto a su contenido.

El préstamo o copia de este documento, incluyendo el uso e informe de su contenido, sólo se permite dentro de los centros de formación.

En caso de excepciones se requiere el permiso por escrito de Siemens A&D SCE (Mr. Knust: E-Mail: michael.knust@hvr.siemens.de). Cualquier incumplimiento de estas normas estará sujeto al pago de los posibles perjuicios causados. Todos los derechos quedan reservados para la traducción y posibilidad de patente.

Agradecemos al Ingeniero Michael Dziallas, a los tutores de las escuelas de formación profesional, así como a todas aquellas personas que nos han prestado su colaboración para la elaboración de este documento.

		PAGE:
1.	Inicio.....	4
2.	Notas para la elaboración de una aplicación en una CPU 315-2DP.....	6
3.	Generación de la configuración Hardware de una CPU 315-2DP	7
4.	Escribiendo un Programa STEP 7.....	19
5.	Depurando un Programa STEP 7.....	22

Los símbolos siguientes acceden a los módulos especificados:



Información



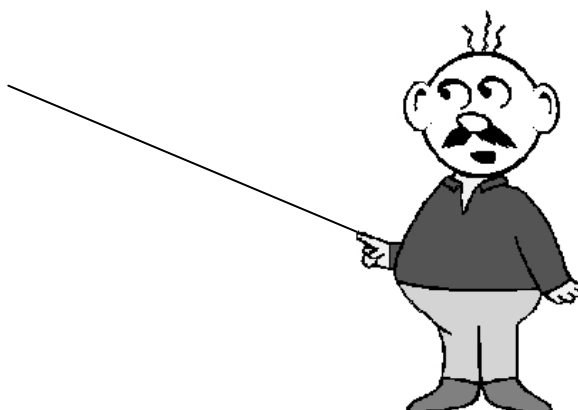
Programación



Ejercicio ejemplo

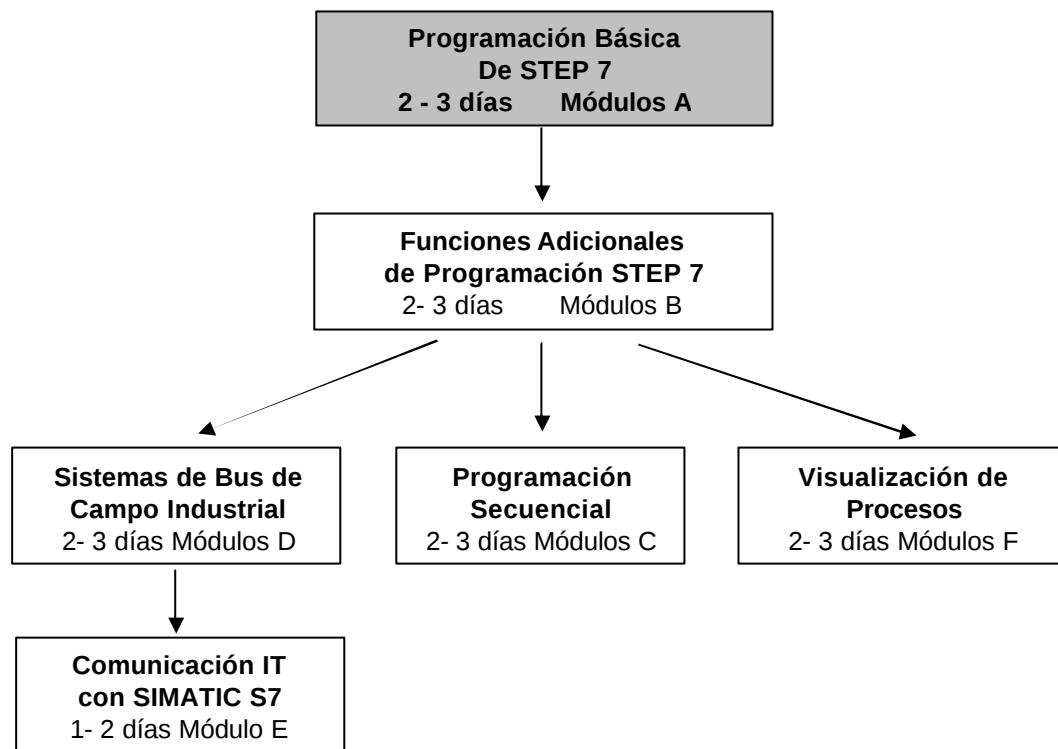


Notas



1. INICIO

El módulo A4 pertenece al contenido de la **Programación Básica de STEP 7**



Finalidad del Aprendizaje:

En este módulo, el lector aprenderá sobre como configurar el hardware de una CPU 315-2DP, así como la elaboración y depuración de un programa STEP 7. El módulo agrupa el procedimiento inicial a través de los siguientes pasos.

- Aplicación con un proyecto STEP 7
- Generación de la configuración hardware en una CPU 315-2DP
- Creación de un programa STEP 7
- Depuración de un programa STEP 7

Requisitos:

Para el correcto aprovechamiento de este módulo, se requieren los siguientes conocimientos:

- Conocimientos de uso de Windows 95/98/2000/ME/NT4.0
- Programación Básica de PLC con STEP 7 (Módulo A3 - 'Puesta en Marcha' programando PLC con STEP 7)

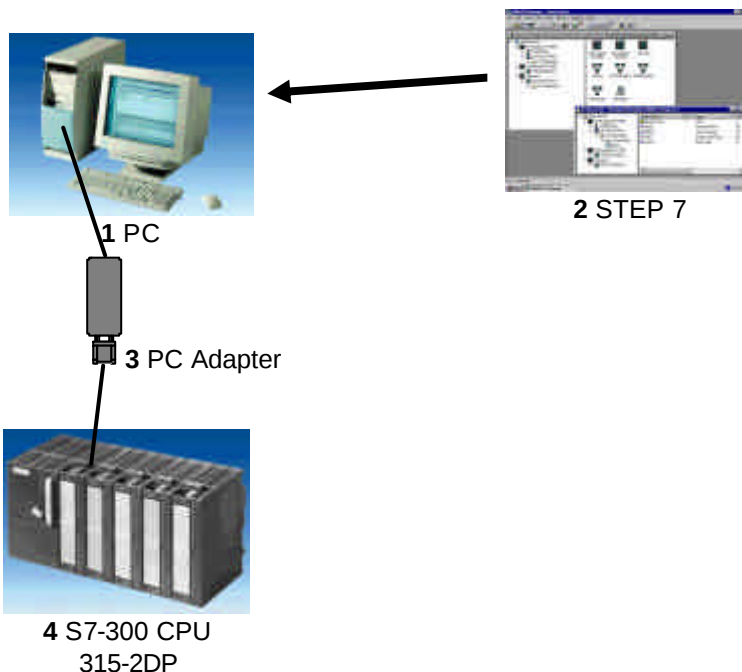
Inicio	Notas	Configuración Hardware	Programa STEP 7	Depuración
---------------	-------	------------------------	-----------------	------------

Hardware y software Necesarios

- 1 PC, Sistema Operativo Windows 95/98/2000/ME/NT4.0 con
 - Mínimo: 133MHz y 64MB RAM, aprox. 65 MB de espacio libre en disco duro
 - Óptimo: 500MHz y 128MB RAM, aprox. 65 MB de espacio libre en disco duro
- 2 Software STEP 7 V 5.x
- 3 Interfase MPI para PC (p.e. PC- Adapter)
- 4 PLC SIMATIC S7-300 con al menos un módulo de entradas/salidas

Ejemplo de configuración:

- Fuente de Alimentación: PS 307 2A
- CPU: CPU 314
- Entradas Digitales: DI 16x DC24V
- Salidas Digitales: DO 16x DC24V / 0.5 A



2. NOTAS PARA LA ELABORACIÓN DE UNA APLICACIÓN EN UNA CPU 315-2DP



La CPU 315-2DP es un autómata que tiene integrada una interfase PROFIBUS DP.

Para dicha CPU, se dispone de los siguientes protocolos PROFIBUS:

- Interfase DP como Maestro, según EN 50170.
- Interfase DP como Esclavo, según EN 50170.

PROFIBUS-DP (Periferia Distribuida) es el protocolo de conexión de elementos de campo o periféricos a una CPU con un rápido tiempo de respuesta.

Otra característica de este protocolo es que la CPU es capaz de direccionar estos módulos de periferia como si fueran módulos propios de entrada/salida.

Esta CPU es por tanto suficiente como elemento de formación, disponiendo de las siguientes características:

- 16K de instrucciones. 48K de memoria de trabajo 80K de espacio disponible.
- 1024 Bytes de Entradas/Salidas Digitales
- 128 Bytes de Entradas/Salidas Analógicas
- 0,3 ms / 1K instrucciones
- 64 contadores
- 128 temporizadores
- 2048 bits de Marcas

3. GENERACIÓN DE LA CONFIGURACIÓN HARDWARE DE UNA CPU 315-2DP



La gestión de archivos se realiza en STEP 7 con el **Administrador SIMATIC**. Se puede, por ejemplo, copiar bloques de programa o ser llamados en procesos posteriores con otras herramientas, simplemente haciendo click con el ratón. Las operaciones disponibles son las mismas que pueden llevarse a cabo con cualquier otra aplicación bajo WINDOWS 95/98/2000/ME/NT4.0. (p.e. con un click del botón derecho del ratón se llega al menú de opciones en un módulo determinado). En la carpeta **SIMATIC S7-300 y CPU**, se muestra la estructura del hardware del PLC. Esto implica que cada proyecto pertenece a un hardware en particular.

En STEP 7, cada proyecto lleva asociado una estructura fija de directorios. Los programas son almacenados de la siguiente manera:

Proyecto:
El directorio contiene el hardware (p.e. Equipo SIMATIC 300) y la subestructuras (p.e. MPI y PROFIBUS).

Equipo SIMATIC 300:
Aqui se almacena la correspondiente configuración hardware (Hardware/SC*1) y datos de la CPU.

FUENTES/SO*1:
Las fuentes se depositan aqui (Ficheros AWL). Estas fuentes podrán ser compiladas en bloques posteriormente

CPU:
El programa S7 y los enlaces (Enlaces/CO*1) son registrados aqui.

Bloques/AP-off*1:
Aqui se depositan los bloques de programa (OB, FB, FC, SFB, SFC, DB etc.).

Símbolos/SY*1:
Aqui se almacenan la lista de símbolos asociados a sus direcciones

Programa S7: Los programas de usuario (Bloques/AP-off*1), tablas de símbolos (Símbolos/SY*1), y Ficheros Fuente (Fuentes/SO*1) se administran aqui.

*1 Términos desde STEP 7 Versión 2.x

Inicio	Notas	Configuración Hardware	Programa STEP 7	Depuración
--------	-------	-------------------------------	-----------------	------------



En las carpetas **Equipo SIMATIC 300** y **CPU**, se muestra la estructura del hardware de la CPU. Aquí se muestra un ejemplo para el caso de una CPU 315-2DP. Deben de ajustarse los temporizadores, contadores, marcas y direcciones de los módulos de entrada/salida.



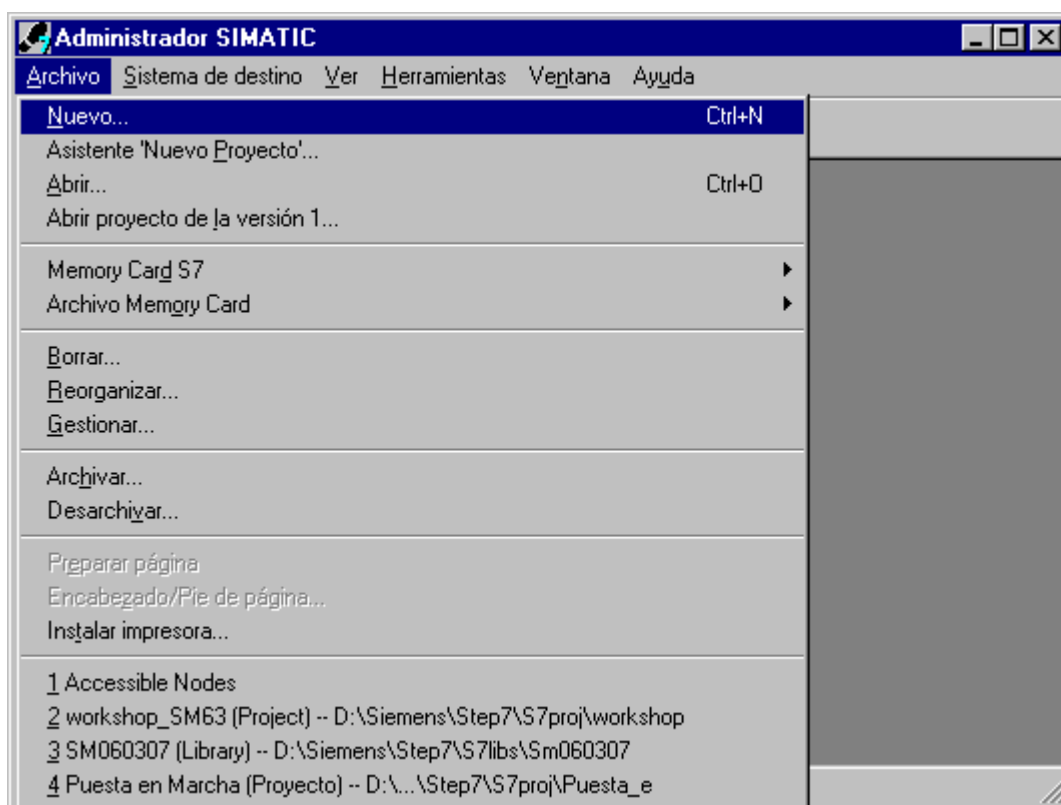
Para poder obtener un proyecto donde elaborar el programa de usuario, se deben de seguir los siguientes pasos:

1. Abrimos la herramienta principal **Administrador SIMATIC**, con un doble click en el icono siguiente (→ Administrador SIMATIC).



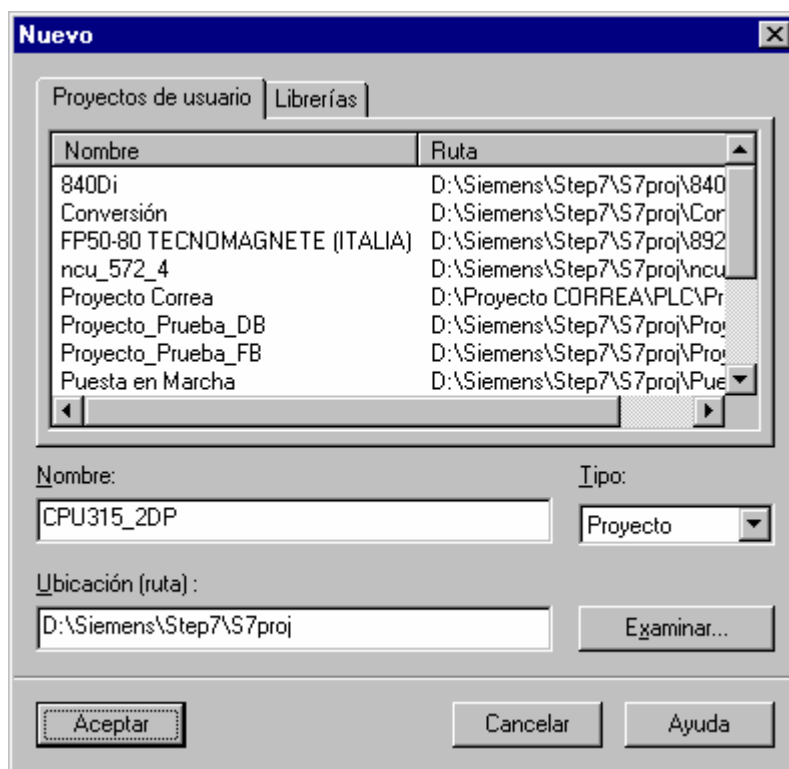
SIMATIC Manager

2. Los programas STEP 7 son gestionados por proyectos. Cada proyecto debe ser creado como nuevo archivo (→ Archivo → Nuevo).

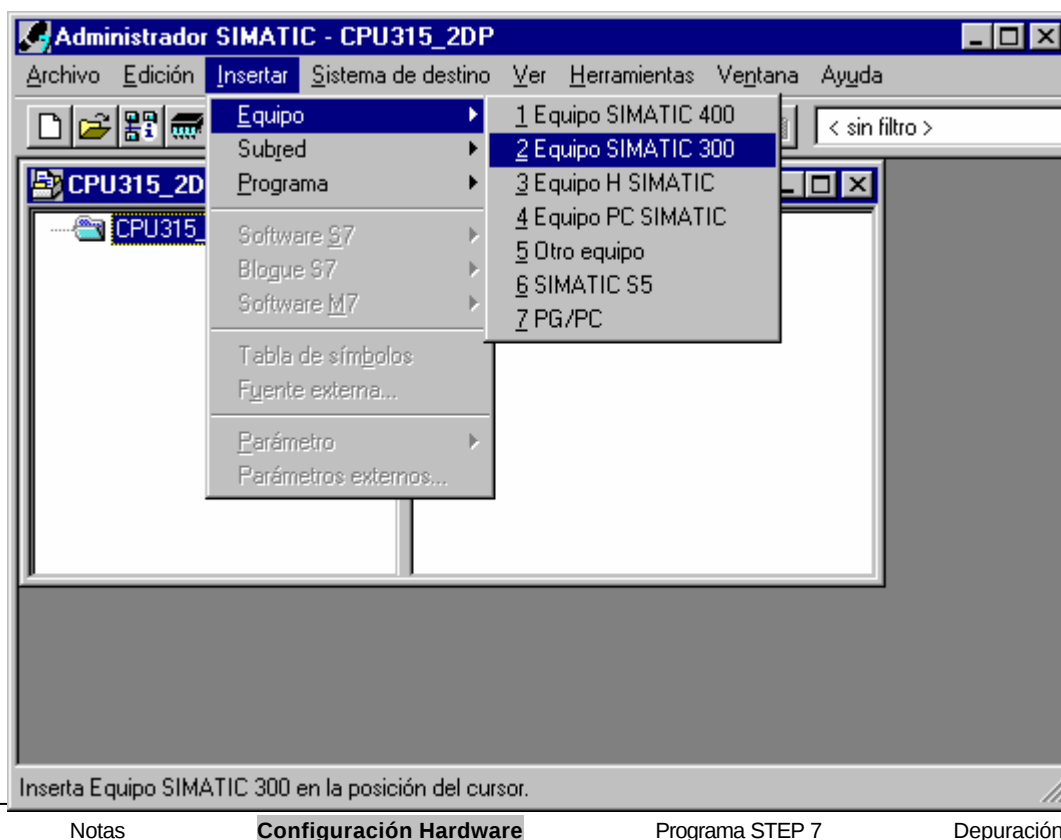




3. Damos al proyecto el nombre **CPU315_2DP** (→ CPU315_2DP → Aceptar).

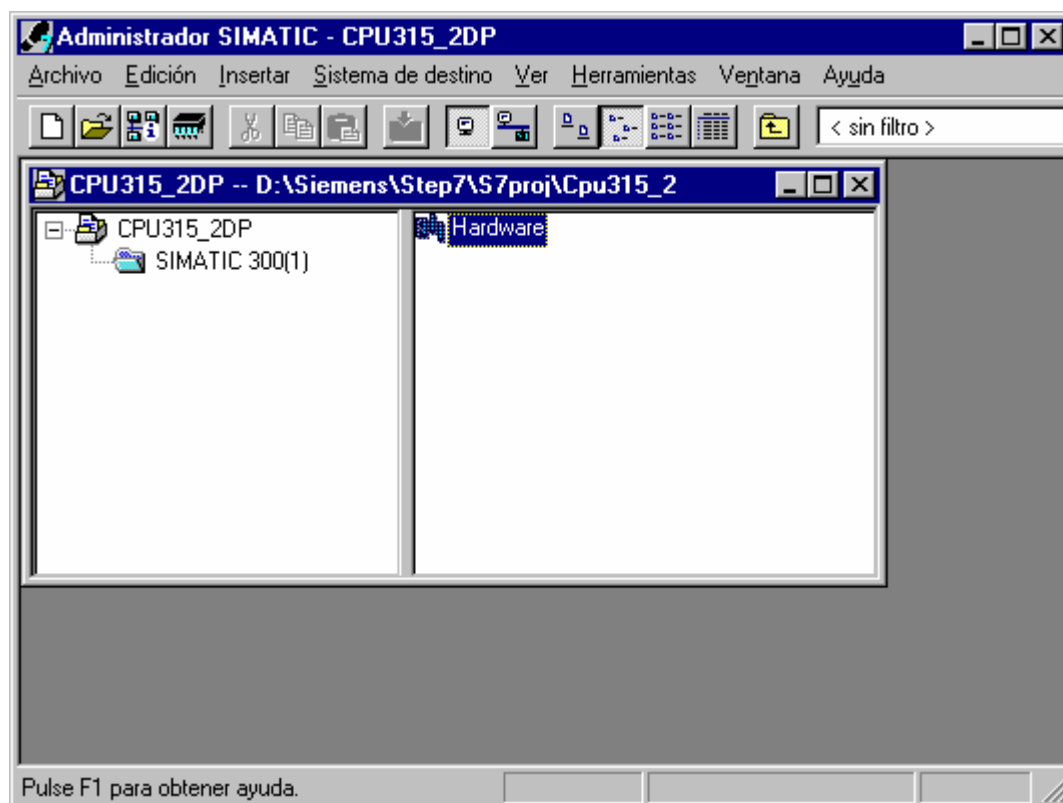


4. Insertamos un **Equipo SIMATIC 300** (→ Insertar → Equipo → Equipo SIMATIC 300).

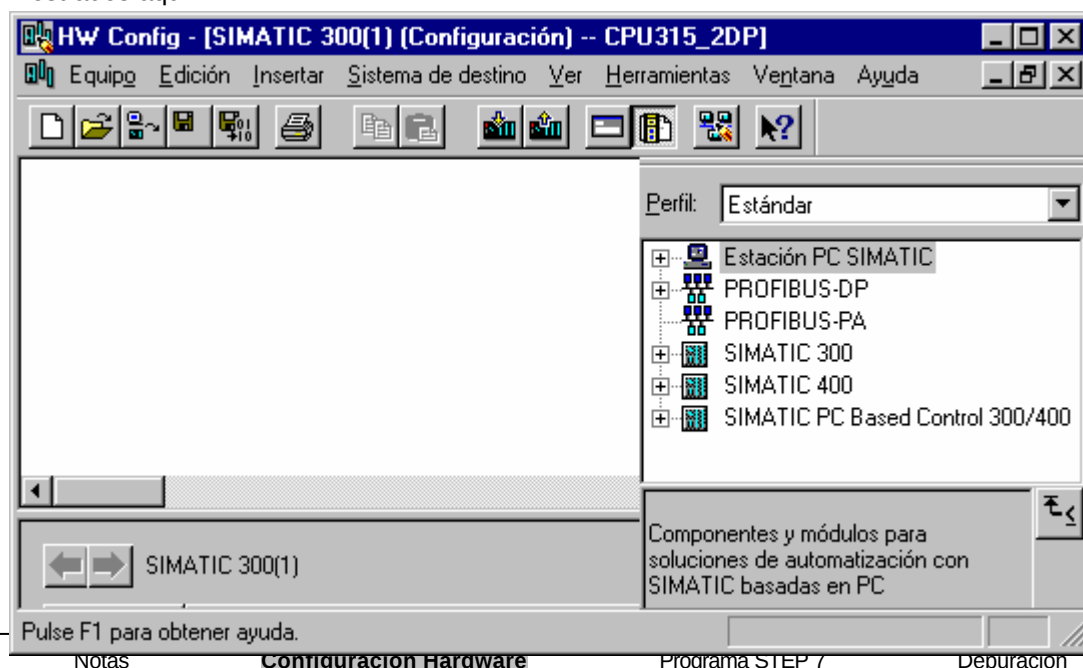




5. Abrimos la aplicación 'Hardware' con un doble click del ratón (→ Hardware).



6. Abrir el Catálogo hardware haciendo click con el botón izquierdo en el símbolo  (→ ).
El contenido se divide en los siguientes módulos:
- PROFIBUS-DP, SIMATIC 300, SIMATIC 400 y Controles SIMATIC Basados en PC.
Todos los componentes, bloques y módulos de interfase necesarios para el proyecto, son mostrados aquí.



Inicio

Notas

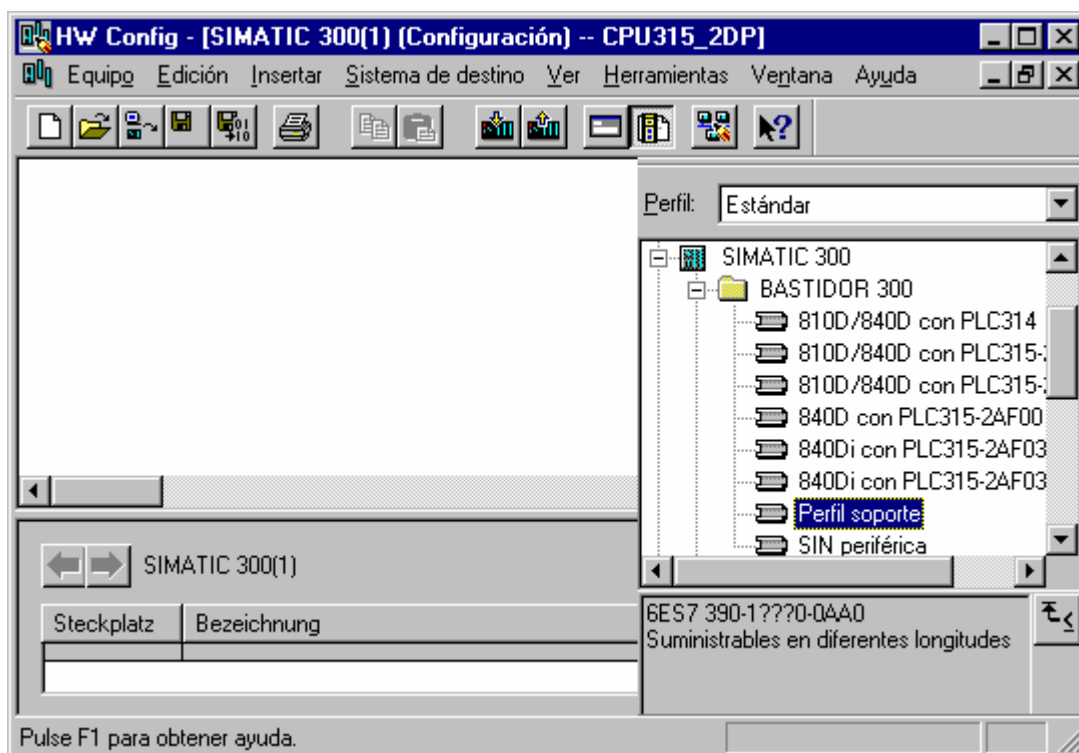
Configuración Hardware

Programa STEP 7

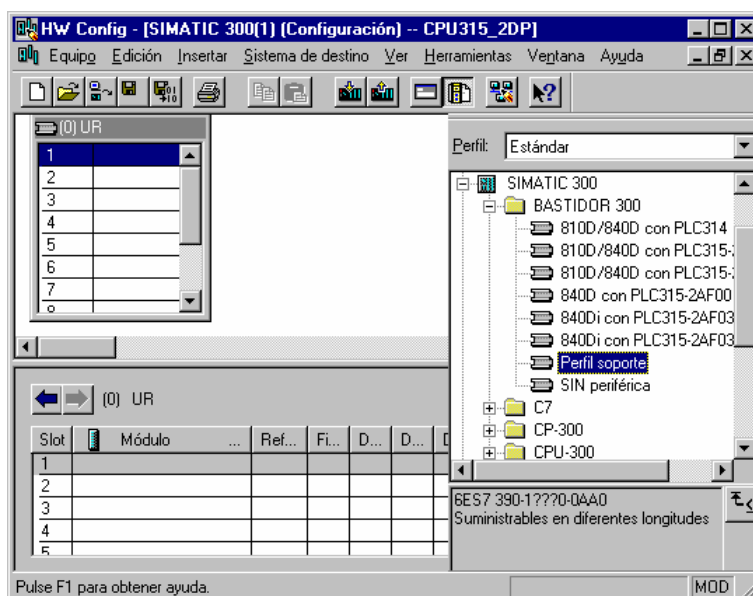
Depuración



7. Hacer doble click en **Perfil Soporte** (→ SIMATIC 300 → Bastidor 300 → Perfil Soporte).



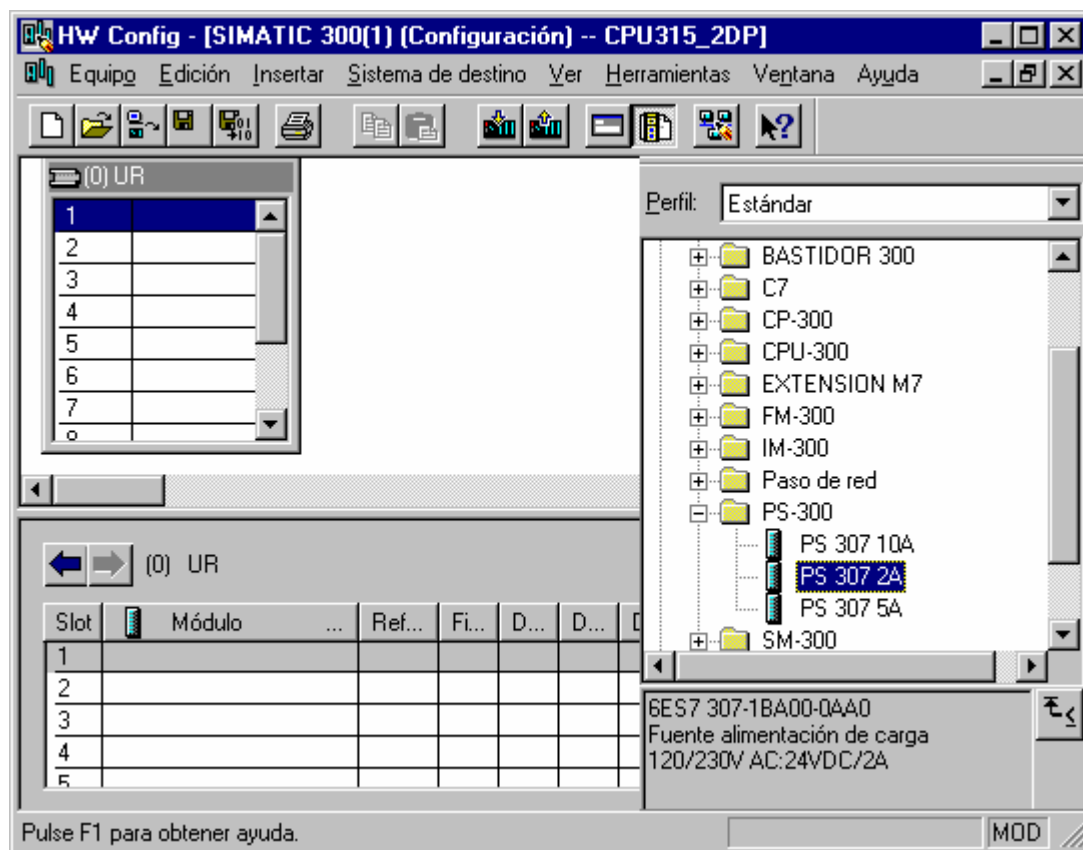
Se mostrará de manera inmediata una tabla de configuración correspondiente a la estructura del soporte RACK 0.





8. Ahora se permite seleccionar todos los módulos que se encuentren en el bastidor e insertarlos en la tabla de configuración desde el catálogo hardware. Para llevar a cabo esta operación, primero se deberá de hacer click en el elemento del catálogo a insertar, mantener el botón del ratón pulsado y arrastrarlo a la primera posición en el bastidor.

Vamos a comenzar con la Fuente de Alimentación **PS 307 2A** (→ SIMATIC 300 → PS-300 → PS 307 2A).



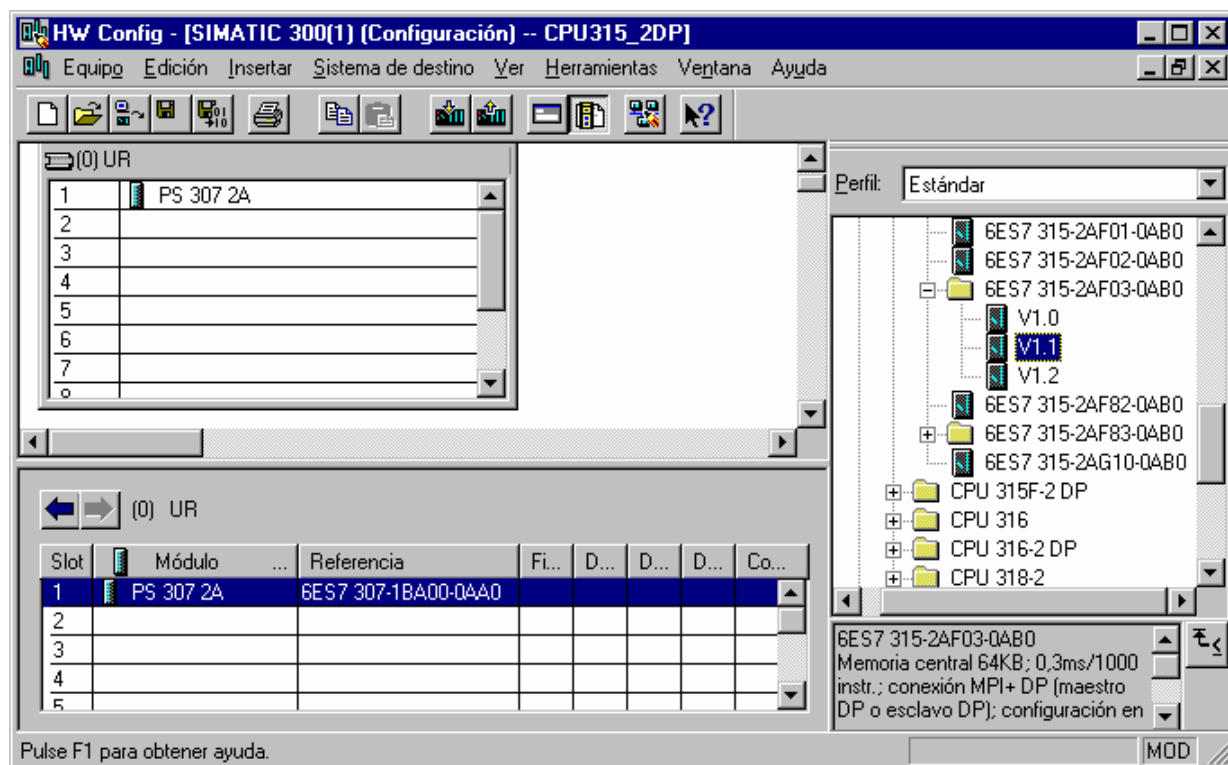
Nota:

Si su hardware difiere de lo mostrado en el ejercicio, se deberá de seleccionar los módulos apropiados en el catálogo e insertarlos en el bastidor.

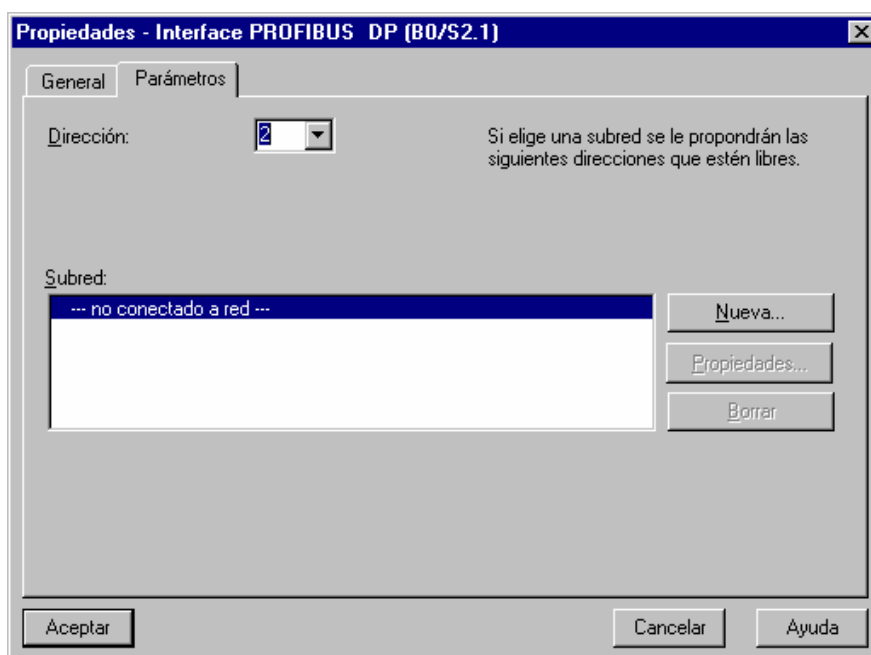
Las referencias de los módulos, las cuales se encuentran serigrafiadas en los componentes físicos, se muestran al pie del catálogo. Asegurese de que son iguales!.



9. En el siguiente paso, colocamos la CPU 315-2DP en la segunda posición del bastidor. Esto permite que se pueda visualizar la referencia y versión de la CPU. (→ SIMATIC 300 → CPU-300 → CPU 315-2DP → 6ES7 315-2AF03-0AB0 → V1.1).

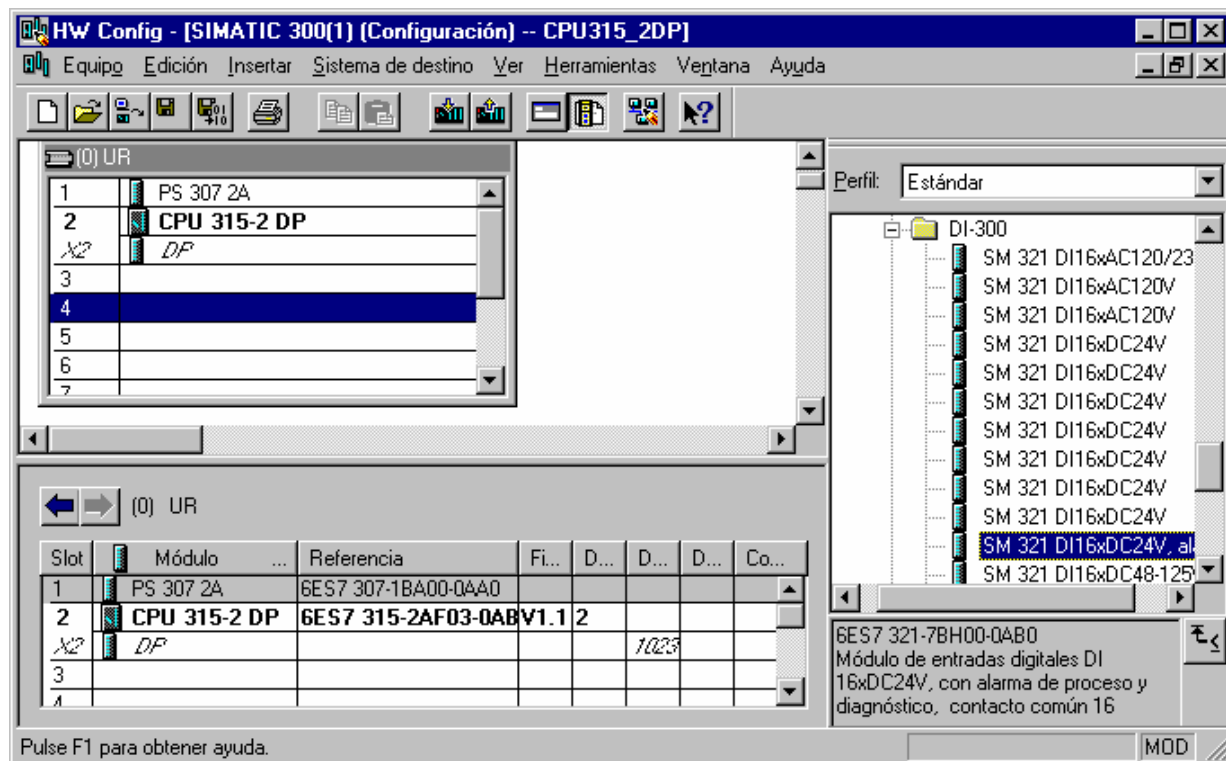


10. En la pantalla siguiente se ajustará el interfase PROFIBUS integrado. En nuestro caso no vamos a ajustar nada por ahora. Hacemos click en **Aceptar** (→ Aceptar).





11. En el siguiente paso arrastramos el módulo de 16 entradas digitales al cuarto puesto. La referencia del módulo se puede leer en la tapa del módulo (→ SIMATIC 300 → SM300 → DI-300 → SM 321 DI16xDC24V).

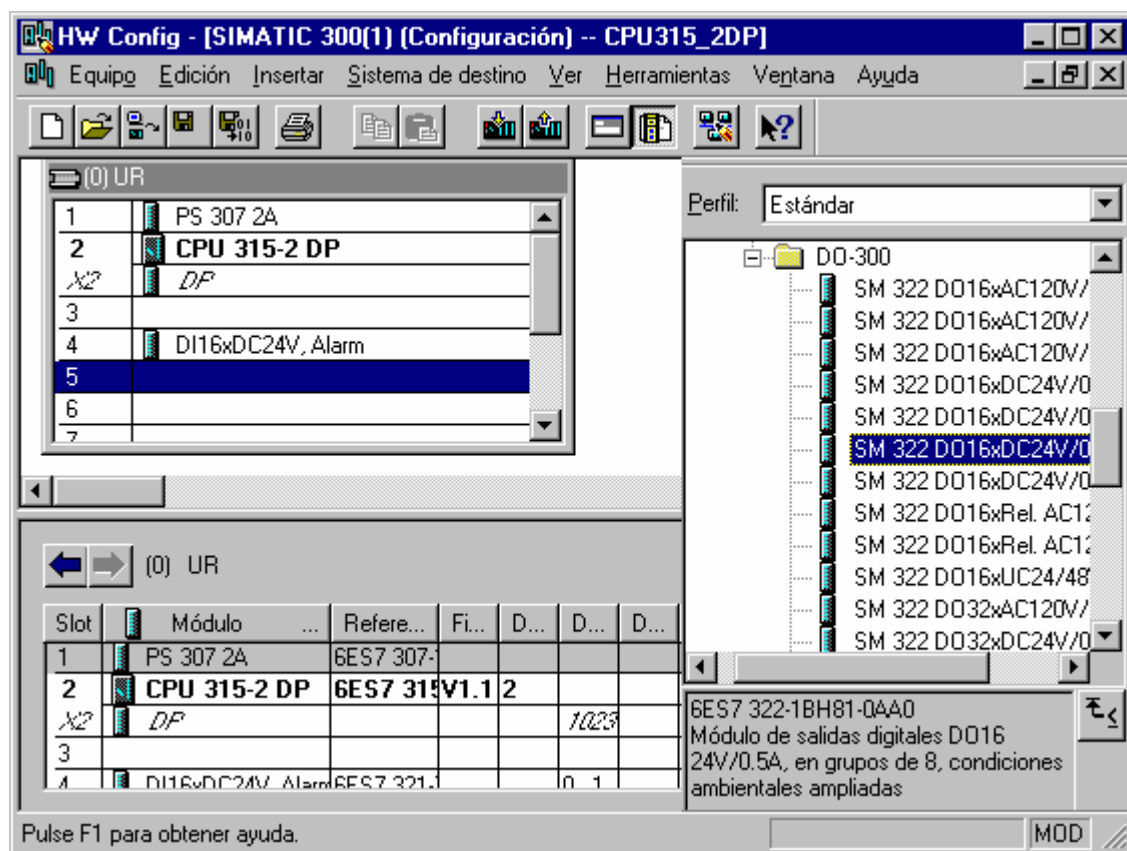


Nota:

El puesto 3 se reserva para los módulos de interfase. En caso de no tener ninguno físicamente en el bastidor, se dejará vacío este puesto. Si existiera, se buscará su correspondiente referencia en el catálogo en **CPU- 300 -> IM - 300**.



12. En el paso siguiente arrastramos el módulo de 16 salidas digitales en el quinto puesto. La referencia del módulo se puede leer en la tapa del módulo (→ SIMATIC 300 → SM300 → DO-300 → SM 322 DO16xDC24V/0,5A).



Nota: La referencia del módulo se muestra al pie del catálogo.



- 13 Se pueden cambiar las características de algunos módulos.
 (→ hacer click con el botón derecho del ratón en la CPU 315-2DP → insertar→Propiedades del objeto → Aceptar).
 p.e. Podemos generar una marca ciclo en todas las CPUs (→ Ciclo/Marca de Ciclo → Ú **Marca Ciclo** → Byte de Marcas 100).

Propiedades - CPU 315-2 DP - (B0/S2)

Alarmas horarias | Alarma cíclica | Diagnóstico / Reloj | Protección | Comunicación

General | Arranque | **Ciclo / Marca de ciclo** | Remanencia | Alarmas

Ciclo

☒ Actualizar cíclicamente la imagen de proceso del OB1

Tiempo de vigilancia del ciclo [ms]: 150

Tiempo de ciclo mín. [ms]: 0

Carga del ciclo por comunicaciones [%]: 20

Tamaño de la imagen del proceso: [dropdown]

Llamada del OB85 en caso de error de acceso a la periferia: No llamar al OB85

Marca de ciclo

☒ Marca de ciclo

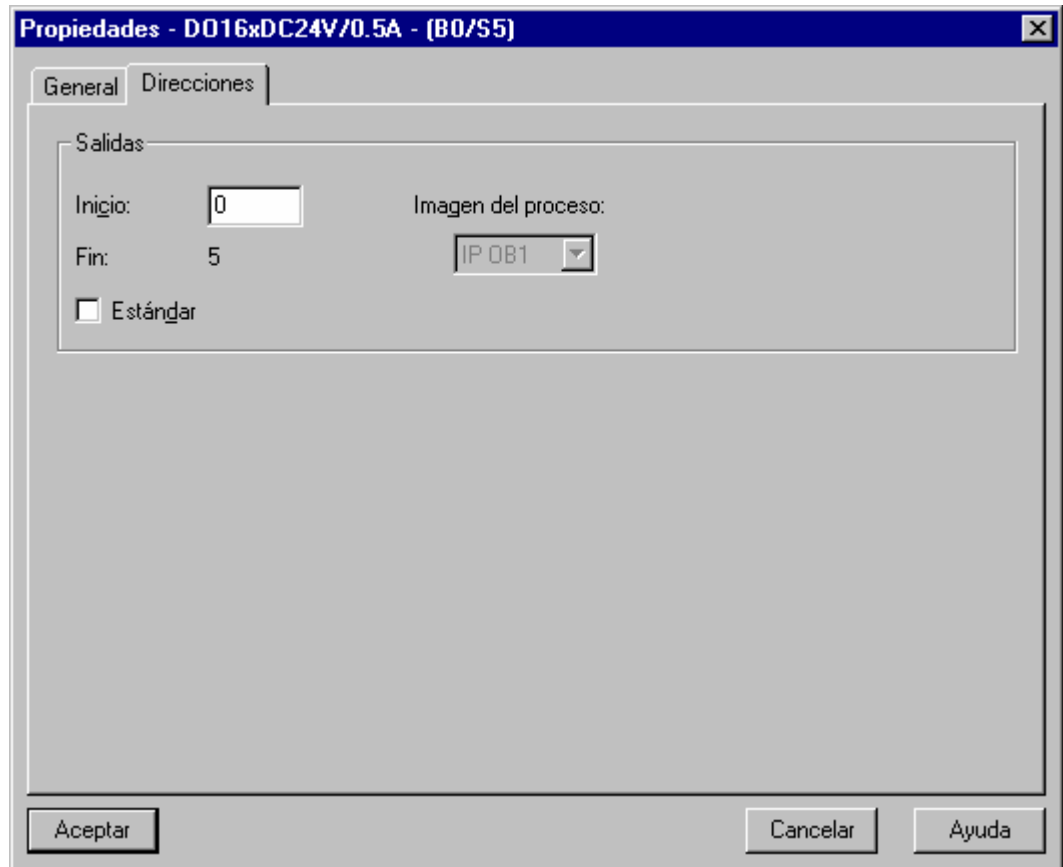
Byte de marcas: 100

Aceptar Cancelar Ayuda



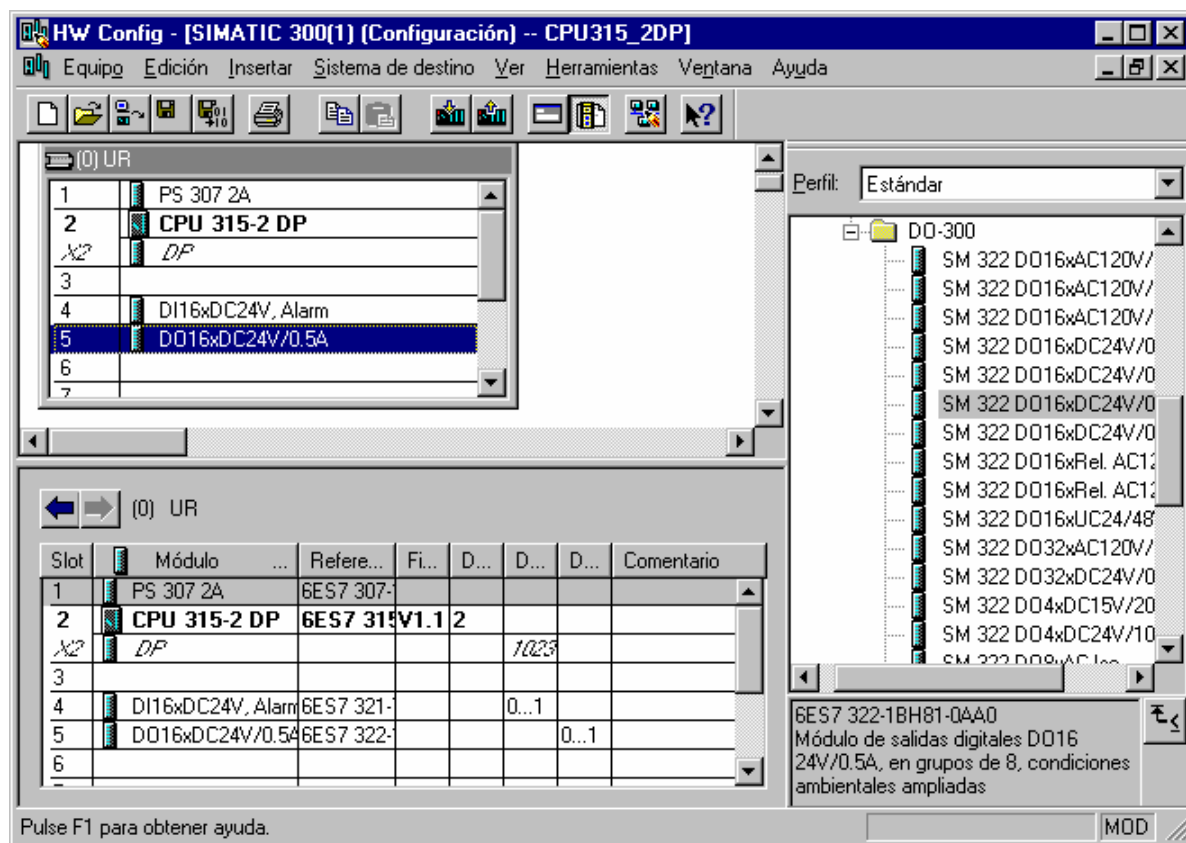
14. Las direcciones de los módulos de Entrada/Salida sólo pueden ser cambiadas en CPUs S7-300 con interfase PROFIBUS integrado.

Para lograr esto, hay que hacer doble click en el módulo apropiado, quitar la opción estándar y ajustar la dirección en el campo 'Inicio'. (→ DO 16xDC24V/0.5A → Direcciones→ quitar opción estándar → 0 → Aceptar).





15. La tabla de configuración será guardada, compilada y cargada en la CPU, haciendo click en  y . El selector de modos de la CPU debería estar en STOP! (→  → )



4. ESCRIBIENDO UN PROGRAMA STEP 7



El programa que va a ser depurado va a ser elaborado en Lista de Instrucciones (AWL) y contiene tan solo dos líneas.

La frecuencia de los bits del byte de marcas MB100, activado en la configuración hardware, son trasferidas a un byte de salidas.

Tabla de Símbolos:

MB100	Reloj	Byte de marca ciclo
QB0	QB	Visualizador de salida

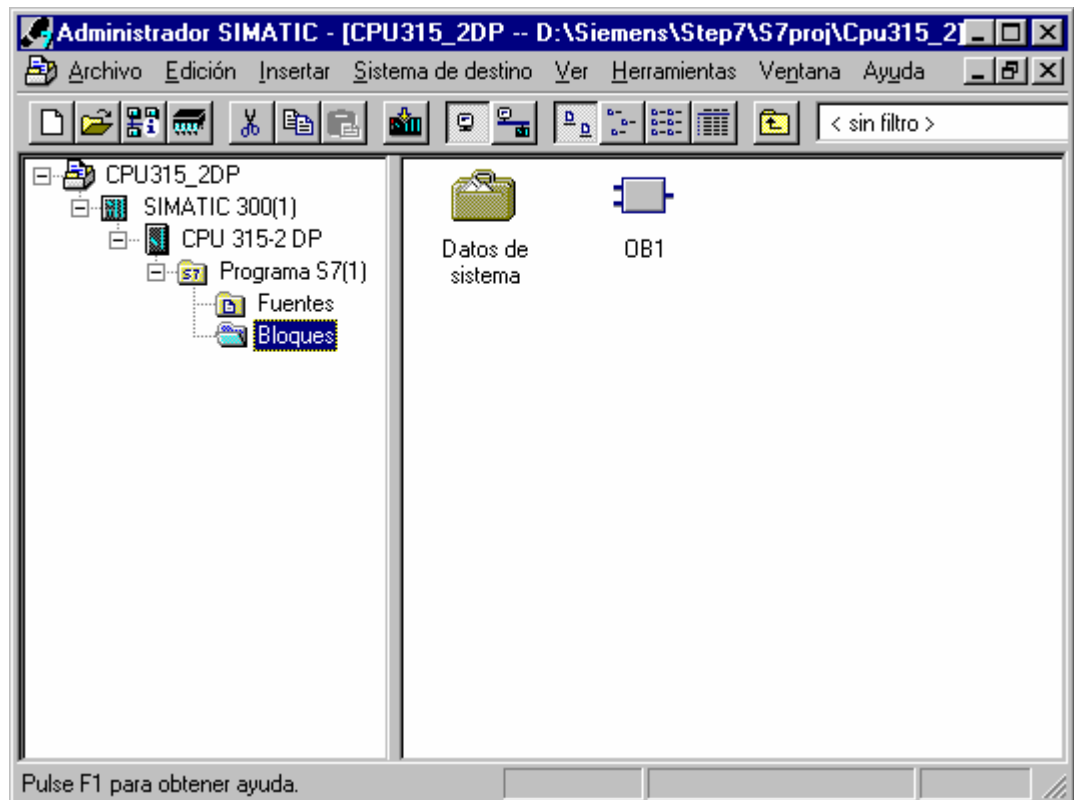


Cada bit de la marca ciclo tiene pre-asignado un período o frecuencia. A continuación se muestra el valor de cada bit:

Bit:	7	6	5	4	3	2	1	0
Período (s):	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Frecuencia (Hz):	0.5	0.625	1	1.25	2	2.5	5	10

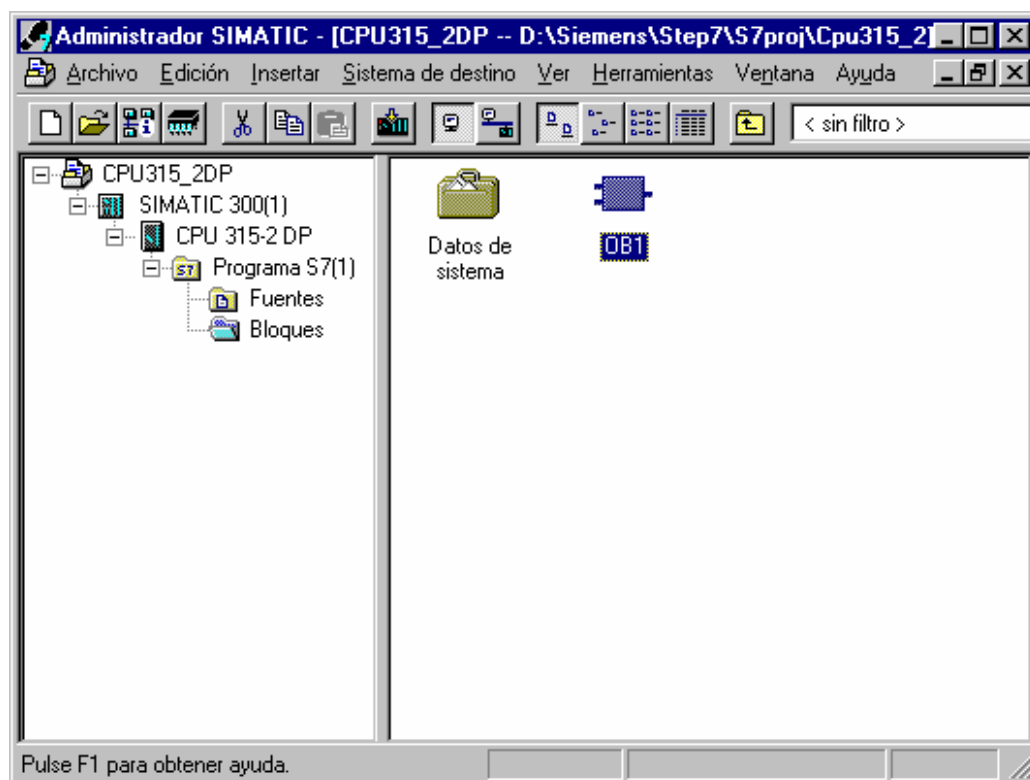


- Seleccionamos la carpeta **Bloques** en el **Administrador SIMATIC** (→ Administrador SIMATIC →Blocks)

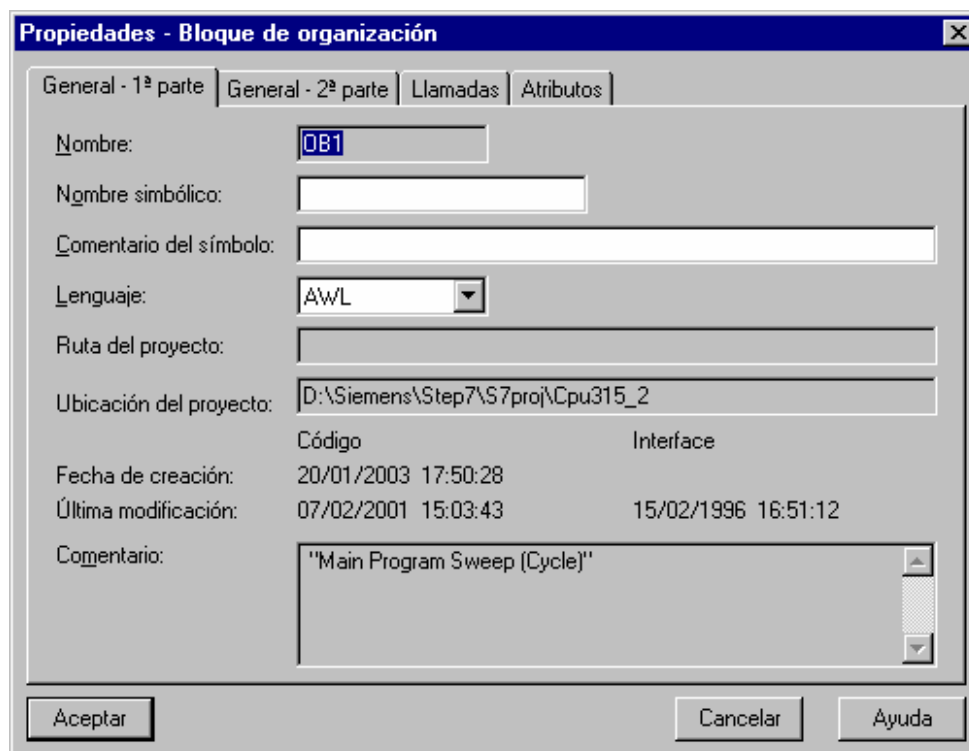




17. En el Administrador SIMATIC hacemos doble click en el bloque **OB1** (→ OB1).



18. Aceptamos las opciones del OB1 con **Aceptar** (→Aceptar).

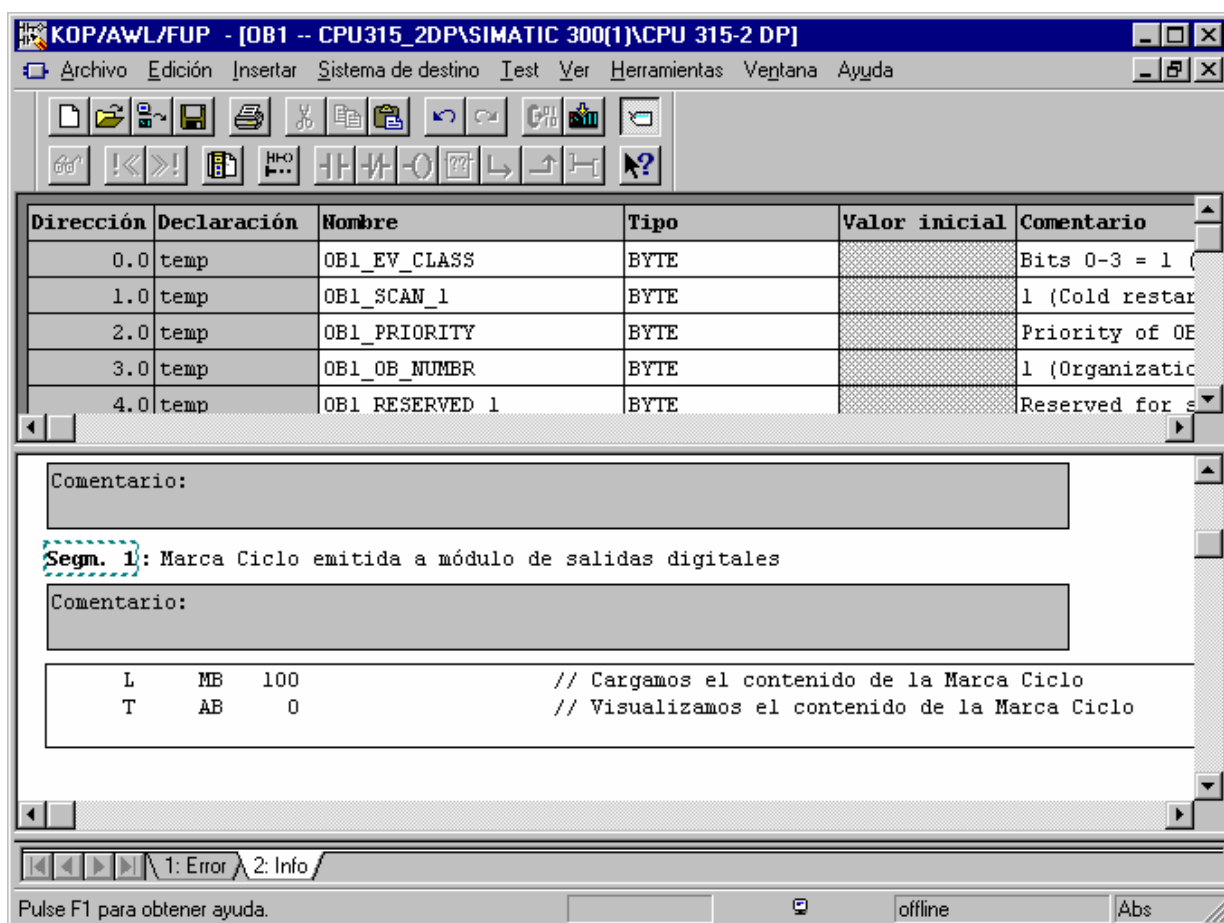




19. Existe un **Editor de Programas KOP, AWL**, que da la posibilidad de editar el Programa STEP 7. Para poder hacer esto, el bloque de organización OB1 debería de haberse abierto ya, mostrando el primer segmento. Se debe de seleccionar el primer segmento, para poder crear la primera operación. Ahora es cuando se puede escribir el primer Programa STEP 7. Los programas STEP 7 normalmente se dividen en segmentos. Se puede insertar un nuevo segmento haciendo click en el símbolo de segmento .



Nota: Los comentarios del programa se han de separar de las instrucciones de programa a través de los caracteres '//'.
 Los comentarios se escriben en la línea de la instrucción a la que se refieren.



En el segmento 1	L	MB 100	//Línea 1
		T	QB 0 //Línea 2

La línea 1 carga el contenido de la Marca Ciclo y la línea 2 lo transfiere al byte de salidas 0. Los 8 bits del byte de salidas parpadean a diferentes frecuencias.



Nota: La dirección del byte de salida puede ser diferente, según la configuración hardware.

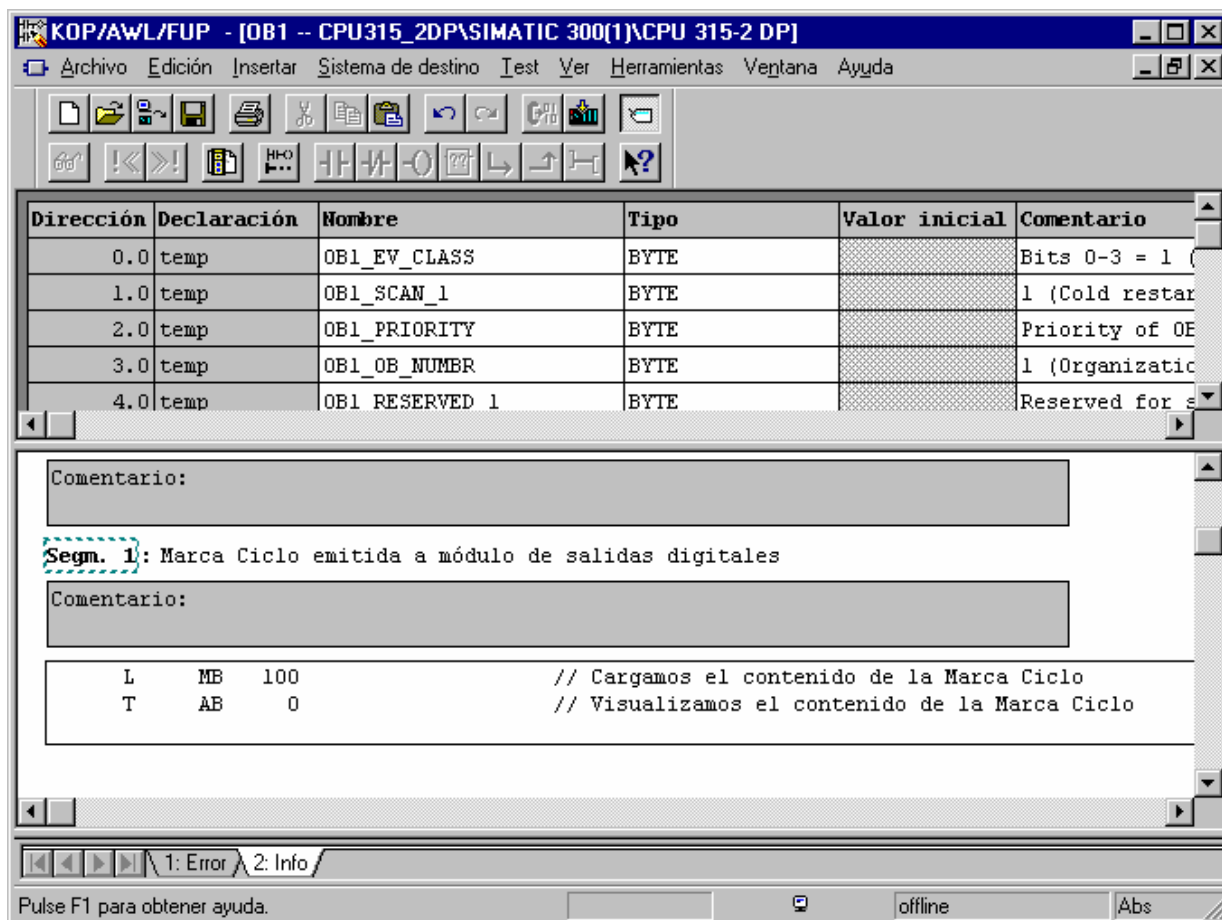
5. DEPURANDO UN PROGRAMA STEP 7



El Programa STEP 7 a depurar será cargado ahora en la CPU. En este ejemplo sólo se va a depurar el bloque de organización OB1.

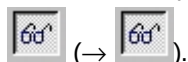
19. Guardar el bloque haciendo click en  y cargar el programa en la CPU, haciendo click en .

El selector de modos de la CPU debería estar en STOP! (→  → )





20. Al pasar el selector de modos a 'RUN', la ejecución del programa es comenzada. Tras haber hecho esto, el programa contenido en el 'OB1' puede ser observado haciendo click en el símbolo



KOP/AWL/FUP - [@OB1 -- CPU315_2DP\SIMATIC 300(1)\CPU 315-2 DP] ONLINE

Archivo Edición Insertar Sistema de destino Test Ver Herramientas Ventana Ayuda

Dirección Declaración Nombre Tipo Valor inicial Comentario

0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = 1
1.0	temp	OB1_SCAN_1	BYTE		1 (Cold restart)
2.0	temp	OB1_PRIORITY	BYTE		Priority of OB
3.0	temp	OB1_OB_NUMBR	BYTE		1 (Organizational)
4.0	temp	OB1_RESERVED 1	BYTE		Reserved for s

iclo emitida a módulo de salidas digi

100 // Cargamos e
0 // Visualizam

RLO	STA	ESTANDAR
0	1	36
0	1	36

1: Error 2: Info

Pulse F1 para obtener ayuda.

RUN Abs