

*Guía del usuario*

# UD70

Módulo de opciones grande y  
software para Unidrive

Referencia: 0460-0069  
Número de publicación: 2



## Información de seguridad

El personal encargado del montaje y la supervisión de la instalación eléctrica o del mantenimiento de un accionamiento y sus unidades opcionales externas debe contar la capacitación y la experiencia adecuadas en la realización de estas tareas.—Asimismo, se le ofrecerá la posibilidad de estudiar esta Guía del usuario y, si fuese necesario, podrá analizar su contenido antes de comenzar a trabajar.

Las tensiones presentes en el accionamiento y las unidades opcionales externas pueden causar una descarga eléctrica grave que podría resultar mortal.—La función Stop (parada) del accionamiento no elimina las tensiones peligrosas de los terminales del accionamiento ni de la unidad opcional externa, por lo que será preciso desconectar la alimentación de la red antes de realizar las tareas de mantenimiento.

Respete las instrucciones de instalación y consulte con el proveedor del equipo en caso de duda.—El propietario o usuario del equipo es responsable de garantizar una correcta instalación del accionamiento y la unidad opcional externa, así como del cumplimiento de los requisitos de operación y mantenimiento establecidos en Health and Safety at Work Act en el Reino Unido y otras leyes, normativas y códigos vigentes en el Reino Unido o el país de aplicación.

El software del accionamiento puede incorporar una función de arranque automático opcional.—A fin de impedir que el personal que trabaja en el motor o en las proximidades de éste, y de los equipos que acciona, pueda sufrir lesiones y evitar posibles daños en el equipo, deberán tomarse todas las precauciones necesarias cuando el accionamiento vaya a funcionar en este modo.

Para garantizar la seguridad del personal, no se debe confiar excesivamente en los controles Stop (parada) y Start (marcha) del accionamiento.—Si pudiera existir algún peligro derivado de la inesperada puesta en marcha del accionamiento, sería necesario instalar un enclavamiento para evitar un arranque involuntario del motor.

## Información general

El fabricante no acepta responsabilidad alguna por las consecuencias que puedan derivarse de un ajuste o una instalación inadecuados, negligentes o incorrectos de los parámetros operativos opcionales del equipo, o de la inadecuación del accionamiento de velocidad variable al motor.

El contenido de esta Guía del usuario se considera correcto en el momento de la impresión. –En aras del compromiso por una política de continuo desarrollo y mejora, el fabricante se reserva el derecho de modificar las especificaciones o prestaciones de este producto, así como el contenido de esta Guía del usuario, sin previo aviso.

Reservados todos los derechos. –Ninguna parte de esta Guía del usuario puede reproducirse o transmitirse de ninguna forma ni por ningún medio, ya sea eléctrico o mecánico, incluidos los sistemas de fotocopia, grabación, almacenamiento o recuperación de la información, sin la autorización por escrito del editor.

|                        |  |
|------------------------|--|
| Copyright              | © Noviembre 2000 Control Techniques Drives Ltd |
| Autor:                 | CT SSPD  |
| Creadores              | AH, PB   |
| Código de publicación: | 70ne2  |
| Fecha de publicación:  | Noviembre 2000                                 |
| Versión S/W::          | Archivos de sistema V2.6.0 y posteriores.      |

---

## Contenido

---

|          |   |            |
|----------|---|------------|
| <b>1</b> | <b>Introducción</b>   | <b>1-1</b> |
| 1.1      | Descripción   | 1-1        |
| 1.2      | Memoria   | 1-2        |
| 1.3      | Requisitos del PC   | 1-2        |
| 1.4      | Datos técnicos del UD70                                     | 1-3        |
| 1.5      | Conocimientos del usuario                                   | 1-3        |
| <b>2</b> | <b>Instalación</b>  | <b>2-1</b> |
| 2.1      | Procedimiento de instalación                                | 2-1        |
| 2.2      | Configuración del sistema                                   | 2-2        |
| <b>3</b> | <b>Procedimientos iniciales</b>                             | <b>3-1</b> |
| 3.1      | Introducción  | 3-1        |
| 3.2      | Programa DPL de ejemplo                                     | 3-1        |
| 3.3      | Creación de un archivo DPL con el juego de herramientas DPL | 3-5        |
| <b>4</b> | <b>Programación DPL</b>                                     | <b>4-1</b> |
| 4.1      | Encabezados de programa                                     | 4-1        |
| 4.2      | Comentarios   | 4-2        |
| 4.3      | Variables   | 4-3        |
| 4.4      | Parámetros  | 4-5        |
| 4.5      | Operadores  | 4-5        |
| 4.6      | Tareas y programación en tiempo real                        | 4-7        |
| 4.7      | Instrucciones y funciones                                   | 4-14       |
| 4.8      | Optimización de programas                                   | 4-16       |
| 4.9      | Apuntadores de parámetro                                    | 4-18       |
| 4.10     | Definición de alias (constantes)                            | 4-18       |

|          |   |            |
|----------|---|------------|
| <b>5</b> | <b>Juego de herramientas DPL</b>                | <b>5-1</b> |
| 5.1      | Descripción del juego de herramientas DPL       | 5-1        |
| 5.2      | Administración de archivos                      | 5-2        |
| 5.3      | Modificación de un programa                     | 5-6        |
| 5.4      | Aplicación de estilos                           | 5-8        |
| 5.5      | Compilación y ejecución de un programa          | 5-9        |
| 5.6      | Transferencia de un programa                    | 5-11       |
| 5.7      | Ejecución de un programa                        | 5-12       |
| 5.8      | Utilidades de control y depuración del programa | 5-12       |
| <b>6</b> | <b>Comunicaciones serie</b>                     | <b>6-1</b> |
| 6.1      | Introducción                                    | 6-1        |
| 6.2      | Conexiones de hardware                          | 6-2        |
| 6.3      | Comunicaciones ANSI                             | 6-4        |
| 6.4      | Modos de comunicaciones serie                   | 6-12       |
| 6.5      | Instrucciones ANSI                              | 6-15       |
| 6.6      | Ejemplo de instrucciones ANSI                   | 6-16       |
| <b>7</b> | <b>Referencia</b>                               | <b>7-1</b> |
| 7.1      | Tareas  | 7-1        |
| 7.2      | Instrucciones y funciones                       | 7-5        |

|           |   |             |
|-----------|---|-------------|
| <b>8</b>  | <b>Funciones</b>  | <b>8-1</b>  |
| 8.1       | Parámetros PLC  | 8-1         |
| 8.2       | Introducción  | 8-2         |
| 8.3       | Líneas del codificador  | 8-3         |
| 8.4       | Posición  | 8-4         |
| 8.5       | Activación del controlador de posición  | 8-5         |
| 8.6       | Valores predeterminados y de reinicio   | 8-6         |
| 8.7       | Descripción de los parámetros   | 8-7         |
| 8.8       | Diagramas lógicos   | 8-21        |
| 8.9       | Bloqueo digital   | 8-26        |
| 8.10      | Función de leva   | 8-28        |
| 8.11      | Cambio de referencia  | 8-31        |
| 8.12      | Unidad de temporización/contador  | 8-33        |
| 8.13      | Puertos de E/S digitales  | 8-36        |
| 8.14      | Almacenamiento en la memoria no volátil   | 8-37        |
| 8.15      | Utilización del puerto RS232 para las comunicaciones de accionamiento a accionamiento | 8-38        |
| <b>9</b>  | <b>Diagnósticos</b>   | <b>9-1</b>  |
| 9.1       | Errores en tiempo de ejecución  | 9-1         |
| 9.2       | Códigos de desconexión en tiempo de ejecución   | 9-2         |
| 9.3       | Mensajes de error del compilador  | 9-3         |
| 9.4       | Gestión avanzada de errores   | 9-5         |
| <b>10</b> | <b>Parámetros</b>   | <b>10-1</b> |
| 10.1      | Parámetros de configuración del UD70  | 10-1        |
| 10.2      | Parámetros virtuales  | 10-4        |
| 10.3      | Modos del puerto RS485  | 10-11       |
| 10.4      | Parámetros de uso general   | 10-12       |

---

# 1 Introducción

---

En un accionamiento de velocidad variable como el Unidrive, un microprocesador propio que opera en tiempo real realiza las funciones de sincronización que resultan necesarias para el buen funcionamiento de los dispositivos motorizados. Esto limita las prestaciones del microprocesador durante la realización de otras tareas y, a su vez, reduce la capacidad de adaptación del accionamiento.

El uso de un segundo procesador para ejecutar el software específico de las aplicaciones permite aumentar al máximo la capacidad de adaptación del accionamiento. Este microprocesador adicional es el UD70, que facilita la adaptación del accionamiento a las aplicaciones mediante la programación del software en el UD70.

## 1.1 Descripción

El UD70 es un microprocesador compacto que se ha integrado en un módulo de opciones grande para facilitar la instalación en accionamientos Unidrive de cualquier tamaño.

Si se utiliza junto con el juego de herramientas DPL, el UD70 ofrece al programador las opciones de escritura de software o de uso de software anteriormente escrito a fin de aumentar la capacidad de adaptación del Unidrive.

### **Juego de herramientas DPL (interfaz Windows(tm))**

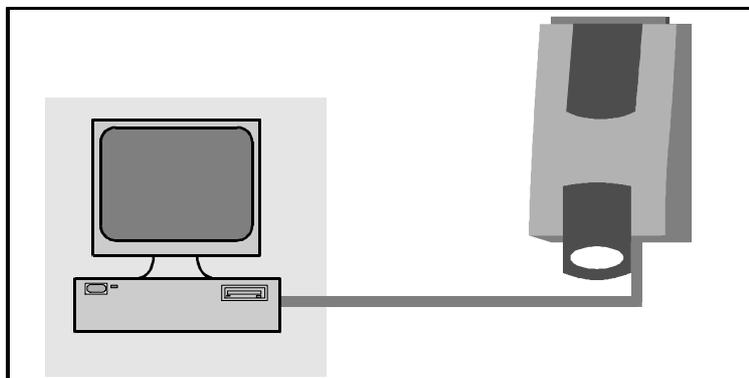
---

El juego de herramientas DPL, suministrado en dos disquetes, es un programa que se ejecuta en Microsoft(r) Windows(tm) versión 3.1x y Windows(tm) 95 y que permite escribir los programas del UD70 en un equipo host.

El UD70 emplea un lenguaje de programación de alto nivel denominado DPL (lenguaje de programación de accionamiento), que en muchos aspectos es similar al lenguaje BASIC. DPL es un programa compilado que ofrece la posibilidad de ejecución a gran velocidad.

El juego de herramientas DPL sirve para escribir, compilar y descargar un programa DPL en un UD70, e incluye una amplia serie de funciones de depuración que facilitan los procesos de desarrollo y prueba del programa DPL.

La conexión entre el UD70 y el equipo host se realiza a través de un enlace de comunicaciones serie RS232. Este enlace sólo es necesario durante los procesos de desarrollo, prueba y puesta en servicio del programa, y puede desconectarse una vez que el software se ha cargado correctamente.



***UD70 conectado a un equipo host mediante un enlace serie***

## **1.2 Memoria**

El programa UD70 compilado y el programa origen creado por el usuario se almacenan en la memoria EEPROM no volátil de la tarjeta UD70. Este tipo de memoria permite cargar los programas mediante el puerto serie.

Esto permite actualizar las versiones de los programas existentes con las más recientes sin necesidad de eliminar circuitos integrados ni utilizar equipos de programación especializados.

El sistema de archivo del UD70 impide almacenar varios programas en el UD70 simultáneamente.

El programa compilado puede almacenarse junto con el código de origen DPL para que el ingeniero del sitio pueda leer el programa almacenado en el UD70 aunque no esté ubicado en el equipo host. (Esta opción puede desactivarse cuando no sea necesaria.)

## **1.3 Requisitos del PC**

Para utilizar el PC con el juego de herramientas DPL, los requisitos mínimos son los siguientes:

PC IBM AT 386SX compatible, Windows(tm) 3.1, 4 Mb de RAM, DOS5

Se recomienda un PC 486 con 8 MB de RAM

## **1.4 Datos técnicos del UD70**

Procesador RISC Intel 960 a 32 bits

96 kb para el almacenamiento del programa por el usuario

8 kb de RAM para el usuario

Reloj de 16 MHz

Puerto RS232 para la programación (IBM AT compatible)

Puerto RS485 con aislamiento óptico para las comunicaciones serie permanentes

## **1.5 Conocimientos del usuario**

En esta Guía del usuario se asume que el usuario tiene conocimientos básicos de Microsoft(r) Windows(tm). Para obtener información específica sobre las operaciones en Windows(tm), consulte la Guía del usuario de Windows.



---

## 2 Instalación

---

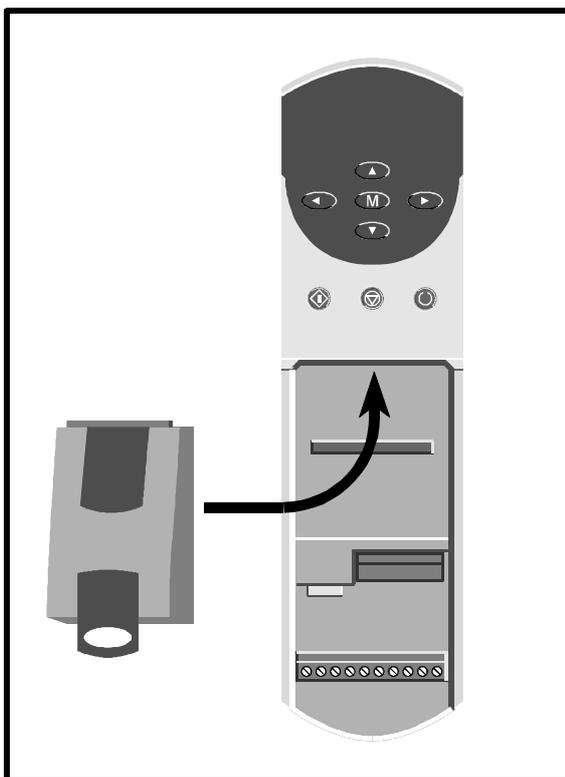


**Advertencia**

Las tensiones presentes en el accionamiento pueden causar una descarga eléctrica grave que podría resultar mortal. La función Stop (parada) del accionamiento no elimina las tensiones peligrosas del accionamiento ni de los mecanismos que acciona.

LA alimentación de CA del accionamiento tendrá que desconectarse al menos 15 minutos antes de extraer los paneles o de realizar cualquier tarea de mantenimiento.

### 2.1 Procedimiento de instalación



#### **Ubicación del UD70 en el Unidrive**

Para instalar un módulo de opciones grande en el accionamiento, consulte las instrucciones de la *Guía del usuario del Unidrive*.

## 2.2 Configuración del sistema

### Conexiones del equipo host

---

**Puerto RS232** El puerto serie RS232 es un enlace especializado que se utiliza en la conexión con el equipo host. Se trata de un conector hembra tipo D con 9 terminales. Por regla general, los cables específicos para las comunicaciones serie RS232 se encuentran disponibles en el mercado.

En la tabla siguiente se incluyen las conexiones mínimas que deben realizarse entre el UD70 y el conector de un puerto COM que disponga de clavijas con 9 y 25 terminales.

| UD70<br>n° terminal | conector de 9 terminales<br>n° terminal | conector de 25 terminales<br>n° terminal |
|---------------------|---|--|
| 2                   | 2                                       | 3  |
| 3                   | 3                                       | 2  |
| 5                   | 5                                       | 7  |

El puerto RS232 sólo debe utilizarse durante la puesta en servicio debido a que no dispone de aislamiento ni protección.

Para realizar la conexión a un equipo host, utilice las siguientes instrucciones:

- 1 Verifique que no se genera carga estática con el enchufe insertado.
- 2 Utilice un cable RS232 de longitud no superior a 3 metros para realizar la conexión al puerto serie RS232 del UD70 y al puerto de comunicaciones serie del equipo host.

### Instalación del juego de herramientas DPL en el equipo host

---

Realice uno de los procedimientos siguientes:

#### Windows 3.1

El juego de herramientas DPL requiere un mínimo de 4 Mb disponibles en la memoria del PC, que puede ser memoria RAM o virtual. La memoria virtual puede configurarse en la sección Modo extendido 386 del Panel de control de Windows.

- 1 Inicie Microsoft Windows
- 2 Inserte el disco 1 del juego de herramientas DPL en la unidad **A:** del equipo host.
- 3 En el Administrador de programas de Windows, seleccione Archivo en la barra de menús y, a continuación, Ejecutar.
- 4 Escriba **A:\SETUP**.
- 5 Haga clic en 

## Windows 95

- 1 Inserte el disco 1 del juego de herramientas DPL en la unidad A: del equipo host.
- 2 En el menú , seleccione Ejecutar....
- 3 Escriba A:\SETUP.
- 4 Haga clic en .

## Transferencia del archivo del sistema

El software del sistema no se ha cargado previamente en el UD70. Por consiguiente, la primera operación que debe realizarse consiste en programar el software del sistema mediante el juego de herramientas DPL y el procedimiento siguiente:

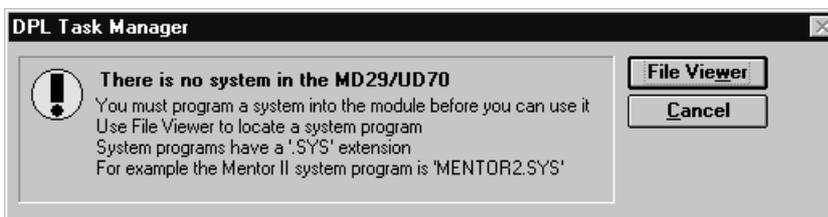
- 1 Conecte el cable de comunicaciones serie al UD70.
- 2 Conecte la alimentación de CA al accionamiento.
- 3 En el Administrador de programas de Windows 3.xx, o en el menú Inicio de Windows 95, haga clic en lo siguiente:



Se abre la ventana **DPL Toolkit**. En la parte superior de ésta se incluye una barra de menús y una barra de herramientas.

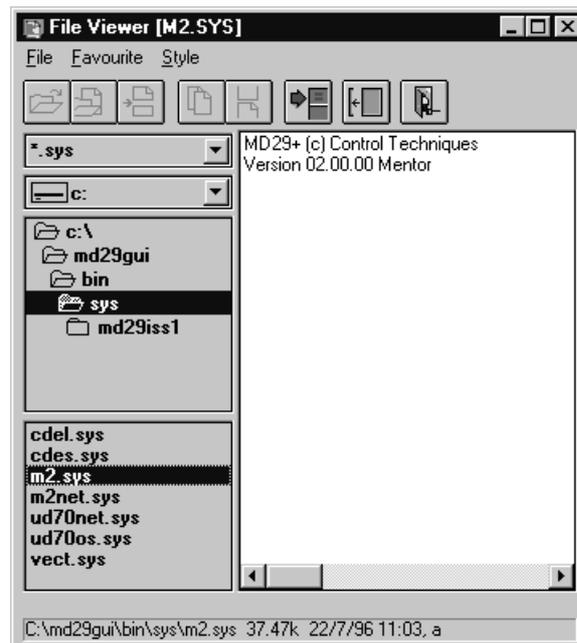


- 4 Si el puerto serie del equipo host no es un puerto COM1, abra el menú **Projects** y seleccione **Configure**. En el menú desplegable que aparece, seleccione **Comport** para abrir otro menú desplegable en el que se puede elegir el puerto de comunicaciones necesario.
- 5 Haga clic en  (**Open Task Manager**). El cuadro de diálogo **DPL Task Manager** aparece después de unos instantes.



Si el cuadro de diálogo no aparece y todos los botones de la barra de herramientas inferior de la ventana están atenuados (inactivos), la comunicación con el UD70 no ha podido establecerse. Compruebe que está utilizando el cable de conexión y el puerto COM correctos.

- Haga clic en **File Viewer**. Aparece el cuadro de diálogo File Viewer.



La ruta de acceso y los nombres de los archivos .SYS pueden seleccionarse en los paneles situados en el lado izquierdo del cuadro de diálogo.

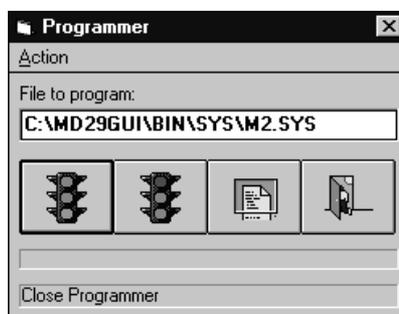
**Nota**

**Los archivos del sistema se encuentran en el directorio MD29GUI\BIN\SYS.**

- 7 Seleccione la ruta de acceso que corresponde a los archivos del sistema necesarios y haga doble clic en el archivo .SYS especificado en la tabla siguiente:

| <b>Accionamiento</b>              | <b>Archivo</b> |
|-----------------------------------|----------------|
| UD70                              | UD70OS.SYS     |
| UD70 con Interbus S o Profibus DP | IBSPROFI.SYS   |
| UD70 con ModBus Plus              | MBPLUS.SYS     |
| UD70 con CTNet                    | UD70NET.SYS    |

Aparece el cuadro de diálogo **Programmer**.



- 8 Haga clic en  (aparece una luz verde). El archivo del sistema se ha cargado en el UD70.



---

## 3 Procedimientos iniciales

---

### 3.1 Introducción

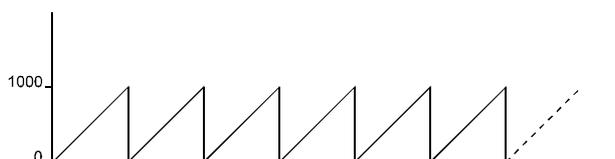
En este capítulo se describen los elementos principales de la programación DPL, además de los procedimientos para crear, compilar y ejecutar un programa de ejemplo utilizando el juego de herramientas DPL .

A continuación se ofrece un ejemplo de un programa DPL reducido, seguido de explicaciones sobre las instrucciones del programa.

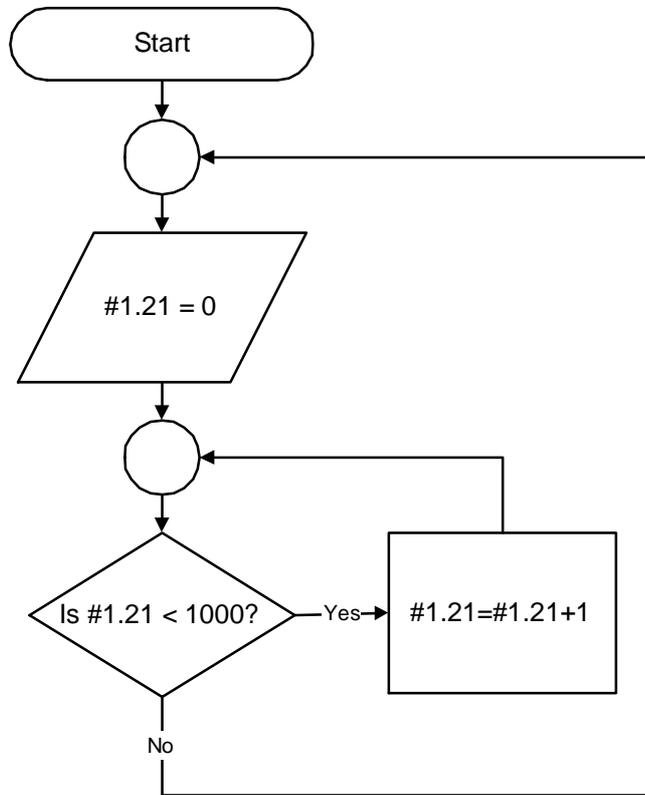
En esta *Guía del usuario* y en los programas DPL, los números de parámetro se indican mediante el símbolo # (por ejemplo, el parámetro **1.21** aparece como #1.21).

### 3.2 Programa DPL de ejemplo

El programa DPL que se describe en esta sección se denomina Sawtooth (diente de sierra), porque aplica un ciclo repetitivo con aumento lineal de la demanda de velocidad seguido de una reducción a cero instantánea, como se muestra en el diagrama.



**Ciclo repetitivo producido por el programa Sawtooth**



**Diagrama de flujo del programa Sawtooth**

**Instrucciones del programa**

```

$title Sawtooth
$version 1.1.1
$drive Unidrive
$author Nombre
$company Empresa
//Note: Esto es un comentario.
background{
  top:
    #1.21=0
    do while #1.21<1000
      #1.21=#1.21+1
    loop
  goto top:
}
  
```

## Explicación del programa de ejemplo

### **\$TITLE Sawtooth**

En la primer línea de un programa debe aparecer el nombre de éste, **\$TITLE**. El nombre debe constar de un máximo de 64 caracteres.

### **\$VERSION 1.1.1**

La segunda línea corresponde al número de versión, **\$VERSION**. Este número puede estar compuesto de ocho caracteres como máximo y el formato recomendado es **\$VERSION 1.0.0**.

Las actualizaciones pueden identificarse fácilmente, ya que el último número aumenta, por ejemplo 1.0.1. En el caso de modificaciones mayores, se indica mediante 2.0.0, 3.00, etc.

### **\$DRIVE Unidrive**

En la tercera línea se muestra el nombre del accionamiento, **\$DRIVE**, lo que permite indicar al compilador el accionamiento que está instalado. (Dado que el juego de herramientas DPL puede utilizarse con el UD70 y el MD29, el nombre del accionamiento debe indicarse.)

### **\$AUTHOR Nombre**

### **\$COMPANY Empresa**

Las líneas cuarta y quinta sirven para definir el creador del programa y el nombre de la empresa.

## **Nota**

### **El programa no se compilará si no incluye estas líneas.**

---

#### **//Note: Esto es un comentario.**

El programa ignora las líneas de comentarios, que pueden insertarse en cualquier parte de un programa. Los comentarios siempre van precedidos de una barra inversa doble [//] o de punto y coma [;].

Los comentarios resultan útiles para insertar una descripción o para proporcionar explicaciones al usuario o el programador.

#### **BACKGROUND{**

**BACKGROUND** es un tipo de tarea. (Todo código ejecutable debe estar contenido en una tarea.) Existen varios tipos de tareas distintos que, de hecho, establecen la prioridad del código y permiten ejecutar bloques de códigos en bases de tiempos diferentes.

**BACKGROUND** es una tarea que se ejecuta de forma independiente y que puede compararse con la forma en la que un PLC ejecuta un programa, por ejemplo. En la sección Programación en tiempo real del Capítulo 4, Programación DPL, se ofrece información detallada sobre las tareas.

#### **Top:**

**Top:** es una etiqueta que marca una posición absoluta en el programa. Las etiquetas siempre deben ir seguidas de dos puntos [:].

La etiqueta define el destino de un enunciado **GOTO** y puede asignársele cualquier nombre (ejemplo, mietiqueta).

### #1.21=0

El uso de una almohadilla (#) proporciona acceso a los parámetros del accionamiento, en este caso el parámetro 1.21 (menú 1, parámetro 21). Éste es un parámetro de referencia de velocidad prefijada del Unidrive y se encuentra ajustado en cero.

### DO WHILE #1.21<1000

DO WHILE es un enunciado repetitivo. En este ejemplo, da instrucciones al programa para que repita el bloque de códigos siguiente mientras el valor del parámetro 1.21 sea inferior a 1000.

### #1.21=#1.21+1

En esta línea se añade el valor 1 al parámetro 1.21. Cada vez que se ejecuta este comando, se añade 1 al valor del parámetro.

### LOOP

LOOP es la expresión final de la instrucción DO WHILE LOOP indica al programa que debe retroceder a la línea DO WHILE para comprobar que la instrucción DO WHILE continúa siendo verdadera. Cuando el valor de #1.21 = 1000, DO WHILE #1.21 < 1000 se transforma en falso. Las instrucciones entre DO WHILE y LOOP dejan de repetirse y el programa pasa a la línea siguiente después del comando LOOP.

### GOTO Top:

GOTO es una instrucción de control del flujo. En este caso indica al programa que debe retroceder a la etiqueta Top:, lo que permite que el programa se ejecute de forma continua.

#### *Nota*

**Para especificar el nombre de la etiqueta deben utilizarse dos puntos [:].**

---

### } Llave de cierre

Las instrucciones que aparecen entre llaves pertenecen a la tarea definida. Las llaves de cierre se utilizan junto con las de apertura, y en este ejemplo operan en combinación con la tarea BACKGROUND.

### 3.3 Creación de un archivo DPL con el juego de herramientas DPL

En esta sección se muestra la forma de escribir, compilar y descargar el programa DPL de ejemplo para utilizarlo con el Unidrive.

#### Apertura del juego de herramientas DPL

En el Administrador de programas de Windows 3.xx, o en el menú Inicio de Windows 95, haga clic en lo siguiente:



Aparece la ventana DPL Toolkit. En la parte superior de ésta se incluye una barra de menús y una barra de herramientas.

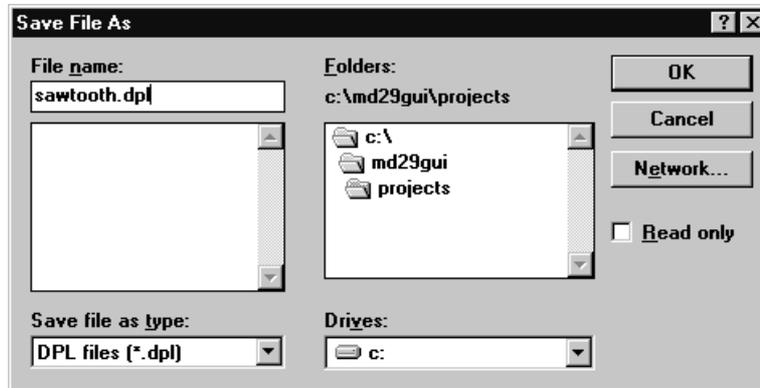


#### Creación de un archivo

- 1 Haga clic en  o abra el menú Archivo y seleccione **Nuevo**.
- 2 Introduzca el programa siguiente tal como se muestra y utilice la tecla Tab para aplicar la sangría a las líneas.

```
$TITLE Sawtooth  
$VERSION 1.1.1  
$DRIVE Unidrive  
$AUTHOR Nombre  
$COMPANY Empresa  
//Note: Esto es un comentario.  
BACKGROUND{  
    Top:  
        #1.21=0  
        DO WHILE #1.21<1000  
            #1.21=#1.21+1  
        LOOP  
    GOTOTop:  
}
```

- 3 Abra el menú Archivo y seleccione Guardar como.... A continuación aparece el cuadro de diálogo Guardar archivo como.



- 4 Seleccione el directorio **Projects** en la lista **Folders:** y escriba **SAWTOOTH.DPL** en el cuadro de texto **File name:**.
- 5 Haga clic en  para guardar el archivo.  
El programa está preparado para compilar en código máquina.

**Nota importante**

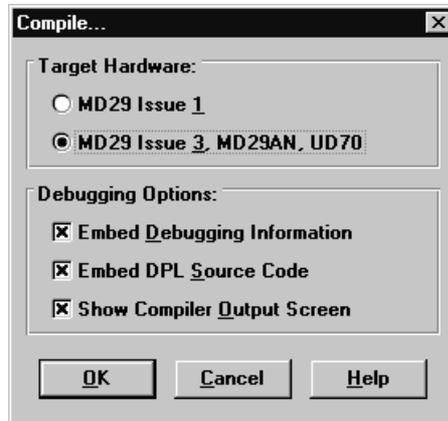
**Los programas DPL deben guardarse como archivos .DPL, de lo contrario no podrá compilarse en código máquina. Sólo se compila la versión almacenada del programa.**

### Compilación del programa

El juego de herramientas DPL contiene un compilador que convierte los programas DPL en formato de texto a código máquina binario que el UD70 puede interpretar. El compilador convierte el archivo DPL en un archivo binario con extensión .BIN.

Utilice el siguiente procedimiento.

- 1 Haga clic en , a la derecha de la barra de herramientas. Aparece el cuadro de diálogo **Compile....**



- 2 Si necesita transferir el archivo DPL origen al UD70, verifique que la casilla de verificación **Embed DPL Source Code** se encuentra activada. De esta forma, el programa DPL podrá leerse en el PC en fechas posteriores (si la copia del PC se pierde, por ejemplo).

Si no necesita transferir el archivo DPL origen, asegúrese de desactivar la casilla de verificación. Cuando el cuadro de diálogo **Compile...** aparezca a continuación, la casilla de verificación mantendrá el último ajuste.

(Las demás opciones de este cuadro de diálogo se describen en la sección **Compilación y ejecución de programas** del Capítulo 5, **Juego de herramientas DPL**.)

- 3 Haga clic en 
- 4 El cuadro **Compilation** se muestra durante unos segundos y no es necesario consultar su contenido.
- 5 El programa se ha compilado y está listo para transferirlo al UD70. Si en lugar de esto aparece una ventana **Build errors** con una serie de errores, **corrija los errores** de escritura del programa y repita el proceso de compilación. (Los mensajes de error se describen en el Capítulo 9, **Diagnósticos**.)

## Conexión con el UD70

Para descargar el archivo compilado no es necesario establecer la comunicación entre el equipo host y el UD70.

Utilice el siguiente procedimiento:

Haga clic en . La ventana de Task Manager se abre y la barra de herramientas de Task Manager aparece debajo de la barra de herramientas estándar.

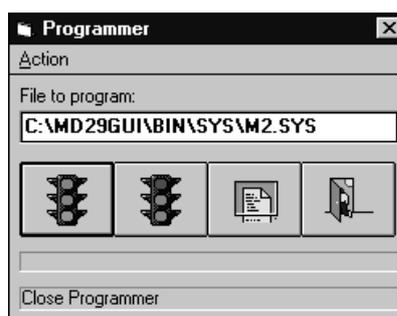


### Nota

**Si todos los botones de la barra de herramientas inferior aparecen atenuados, indica que no se ha podido establecer la comunicación con el UD70. Verifique que ha conectado al alimentación de CA al accionamiento y que el cable de comunicaciones serie está correctamente insertado.**

## Transferencia del programa

- 1 En la barra de herramientas de Task Manager, haga clic en  para abrir el cuadro de diálogo Programmer.



- 2 En el cuadro de diálogo Programmer, haga clic en  (aparece una luz verde) para transferir los archivos **SAWTOOTH.BIN** y **SAWTOOTH.DPL** al UD70. Este proceso tarda unos segundos en realizarse.

**Nota**

**La memoria del UD70 sólo tiene capacidad para un programa compilado ( como el archivo .BIN). El programa existente en el UD70 se reemplazará al descargar otro.**

---

**Ejecución del programa**

---



**Advertencia**

**El programa Sawtooth modifica inmediatamente el parámetro de referencia de velocidad del accionamiento. Por razones de seguridad, verifique que el accionamiento está desactivado antes de ejecutar el programa.**

En la barra de herramientas de Task Manager, haga clic en



El valor del parámetro de referencia de velocidad #1.21 del accionamiento cambiará. Sin embargo, el comportamiento de rampa ascendente no podrá detectarse debido a que el programa modifica el valor del parámetro a una velocidad superior a la de actualización de la pantalla.



---

## 4 Programación DPL

---

En este capítulo se describen las siguientes partes de un programa DPL:

- Encabezados de programa
- Comentarios
- Variables
- Parámetros
- Tareas
- Subrutinas definidas por el usuario
- Instrucciones

Esta explicación se completa con una sección dedicada a la optimización de los programas DPL.

### 4.1 Encabezados de programa

Un programa DPL debe comenzar con cinco encabezados de programa en el orden adecuado, indicado a continuación:

- Título de programa
- Versión de programa
- Nombre de accionamiento
- Nombre de creador
- Nombre de empresa

Cada encabezado debe estar contenido en una sola línea de instrucciones del programa.

#### Título de programa

---

**Sintaxis** `$TITLE título de programa`

El encabezado `$TITLE` está reservado al programador.

ejemplo: `$TITLE generador Sawtooth`

Longitud máxima: 64 caracteres

#### Versión de programa

---

**Sintaxis** `$VERSION número de versión`

El encabezado `$VERSION` está reservado al programador. El formato recomendado para el número de versión es el siguiente:

`$VERSION 1.0.1`

Las actualizaciones menores pueden indicarse aumentando el último número, por ejemplo `1.0.2`, mientras que las modificaciones más importantes se identificarán por el aumento del primer dígito, ejemplo `2.0.0`.

Longitud máxima: 8 caracteres

## Nombre de accionamiento

---

### *Sintaxis* **\$DRIVE nombre de accionamiento**

Dado que el juego de herramientas DPL puede utilizarse con distintos tipos de accionamientos, es preciso especificar el tipo de accionamiento en el encabezado **\$DRIVE**.

Este encabezado permite asegurarse de que el programa se compilará correctamente para el módulo de opciones y el accionamiento utilizados.

### **Nota**

**Si se especifica un accionamiento distinto, es posible que el programa no se compile o que ocurra un error en tiempo de ejecución 53 al transferir el programa al UD70.**

---

## Nombre de creador

---

### *Sintaxis* **\$AUTHOR nombre**

El encabezado **\$AUTHOR** está reservado al programador.

Longitud máxima: 64 caracteres

## Nombre de empresa

---

### *Sintaxis* **\$COMPANY empresa**

El encabezado **\$COMPANY** está reservado al programador.

Longitud máxima: 64 caracteres

## Ejemplo de encabezados de programa

---

```
$TITLE Sawtooth
$VERSION 1.0.1
$DRIVE Unidrive
$AUTHOR A.H.
$COMPANY Control Techniques
```

## 4.2 Comentarios

Los comentarios tienen un fin meramente informativo y explicativo, y funcionan igual que los comandos **REM** porque no afectan al programa.

Los comentarios comienzan con una barra invertida doble [//] o un punto y coma [;] y pueden incluirse en la línea de comentarios o al final de las líneas de instrucciones. El comentario termina al final de la línea.

*Ejemplo*      //Esta línea contiene un comentario que termina con la línea.  
                 //Si el comentario continúa en la línea siguiente, debe utilizarse  
                 //una barra invertida doble al comienzo de la línea siguiente.



## Matrices

---

Las matrices son conjuntos de variables del mismo tipo (entera o de coma flotante) que se han agrupado bajo el mismo nombre. Sólo es posible utilizar matrices unidimensionales.

Cada elemento (componente individual) de una matriz es en realidad una variable independiente. Para que el programa pueda acceder a un elemento, es preciso especificar el nombre de la matriz y colocar el número de elemento entre corchetes [ ] detrás del nombre de la matriz.

Los dos tipos básicos de matriz son los siguientes:

**Matrices dinámicas** Estas matrices pueden configurarse y modificarse mediante programas DPL. Una matriz dinámica debe contener variables enteras o de coma flotante, pero nunca ambos tipos de variables a la vez.

Para especificar una matriz dinámica por primera vez, es preciso utilizar la instrucción DIM (generalmente en la tarea Initial) y el número de elementos indicado entre corchetes detrás del nombre de variable. Las matrices dinámicas se incluyen en la memoria volátil de 8 kB del UD70, lo que limita el tamaño máximo de la matriz.

**Ejemplo**

|                   |   |
|-------------------|---|
| DIM mimatriz%(20) | ;Matriz entera con 20 elementos           |
| DIM matriz2[30]   | ;Matriz de coma flotante con 30 elementos |

Los elementos de una matriz se numeran del siguiente modo:

0 a [Número de elementos] - 1

En el ejemplo de matriz entera anterior, el primer elemento de **mimatriz%[]** es el siguiente:

**mimatriz%[0]**

El último elemento es:

**mimatriz%[19]**

**Matrices constantes** Las matrices constantes contienen valores predefinidos fijos que el programa DPL no puede modificar durante su ejecución. Los valores de estas matrices se definen en el programa DPL mediante el uso de una sección especial denominada **CONST**. (Esta sección se introduce exactamente igual que una tarea.) En una matriz constante sólo pueden definirse valores enteros.

El uso de matrices constantes ofrece una ventaja: la matriz se incluye en la memoria de 96 kB del UD70, por lo que el tamaño de la matriz puede limitarse en función del espacio disponible en el UD70 para los programas, en lugar de utilizar el tamaño de la memoria RAM de 8 kB. Este espacio sirve para almacenar el programa DPL compilado, los datos de la matriz constante y (opcionalmente) el propio archivo DPL.

**Ejemplo**

```
CONST c_matriz% {
100, 1500, 500, 0, -400, -1000
-400, -100, 0
}
```

De esta forma se define una matriz denominada `c_matriz%[]` que contiene nueve elementos. El valor de cada elemento puede separarse mediante una coma o un salto de línea.

## 4.4 Parámetros

Existen dos tipos de parámetros:

- Parámetros del accionamiento
- Parámetros virtuales

(Consulte el Capítulo 10, Parámetros.)

Los parámetros se indican mediante el símbolo # (almohadilla) y puede accederse a ellos utilizando un formato `x,y`; donde `x` representa el menú e `y` indica el parámetro del menú.

Por ejemplo, al parámetro `p7.05` se accede introduciendo `#07.05` y a `p18.01`, introduciendo `#18.01`. Es posible omitir los ceros iniciales del parámetro; `#7.5` es igual que `#07.05`, por ejemplo.

También es posible acceder a los parámetros de forma indirecta mediante una variable entera que indique el número de parámetro. Consulte la sección Apuntadores de parámetro más adelante en este capítulo.

## 4.5 Operadores

Los operadores realizan operaciones matemáticas o lógicas con los valores. En la programación DPL pueden utilizarse los siguientes operadores.

### Nota

**Algunos operadores sólo funcionan con variables o valores enteros.**

---

### Operadores para variables enteras y de coma flotante

---

- |   |                |
|---|----------------|
| + | Suma           |
| - | Resta          |
| / | División       |
| * | Multiplicación |

## Operadores para variables enteras solamente

**& AND lógico**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Ejemplo** 5 & 14 = 4

**| OR lógico**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Ejemplo** 5 | 14 = 15

**XOR lógico**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Ejemplo** 5 & 14 = 11

**Inversión de valor de bit** El operador !Value Bit invert invierte el bit menos significativo y convierte los demás bits en ceros.

**Ejemplo** 100100 (binario) se convierte en 000001 (binario)

**Inversión de campo de bits** El operador !(value, bit-field-size) Bit-field invert invierte el número especificado de bits menos significatos y convierte todos los demás en ceros. Bit-field-size especifica el número de bits menos significativos que se van a convertir.

**Ejemplo** Resultado% = !(valor%, 3)  
100100 (binario) se convierte en 000011 (binario).

**Resto (% Remainder)** Este operador permite obtener el resto de la división de un entero entre otro entero.

**Ejemplo** 5 % 2 = 1  
8 % 3 = 2

## 4.6 Tareas y programación en tiempo real

La programación en tiempo real se ejecuta utilizando un reloj como referencia para que el usuario pueda determinar el tiempo real en que se ejecutaron las instrucciones, no sólo el orden de ejecución. Para programar en tiempo real, es preciso mantener una estructura (o filosofía) de tareas.

Los programas UD70 contienen secciones denominadas tareas. Cada tarea activa una prioridad que se va a asignar a una subrutina. Estas tareas definen siete niveles de prioridad, por el siguiente orden:

- Tarea INITIAL
- Tarea BACKGROUND
- Tarea CLOCK
- Tarea ENCODER
- Tarea SPEED
- Tarea EVENT
- Tarea ERROR

En el programa, cada tarea se identifica mediante un nombre y su contenido debe incluirse entre llaves { }.

**Ejemplo**

```
CLOCK{
  instrucciones
}
```

### Tarea INITIAL

---

En el programa DPL, la tarea INITIAL suele utilizarse para inicializar las variables de programa y los parámetros del accionamiento. Esta tarea sólo se ejecuta cuando se reinicia el UD70 o se conecta la alimentación de CA.

Mientras se está ejecutando, la tarea Initial tiene prioridad absoluta sobre las demás tareas e impide que se ejecuten. Esto es significativo cuando las tareas CLOCK, EVENT o ENCODER van a manipular datos que contienen valores iniciales.

**Ejemplo**

```
INITIAL{
  // Éste es la única ubicación en la que se puede inicializar sin problemas el
  // 'temporizador'
  temporizador% = 0
}
CLOCK{
  //Esta tarea se ajusta en 5 ms
  //El valor del temporizador debe inicializarse antes de ejecutar CLOCK
  temporizador% = temporizador% + 1
  IF temporizador% > 200 THEN
    //200, intervalos 5 ms = 1 segundo
    PRINT "1 segundo transcurrido"
    temporizador% = 0
  ENDF
}
```

## Tarea BACKGROUND

---

La tarea BACKGROUND se utiliza con funciones y comandos que no precisan control basado en el tiempo ni en el codificador. Esta tarea se utiliza en las siguientes operaciones:

- Registro de datos
- Comprobación de entradas digitales
- Configuración de estado de salida

La tarea BACKGROUND se ejecuta una vez terminada la tarea Initial. Se recomienda ejecutar la mayor parte del programa en la tarea background.

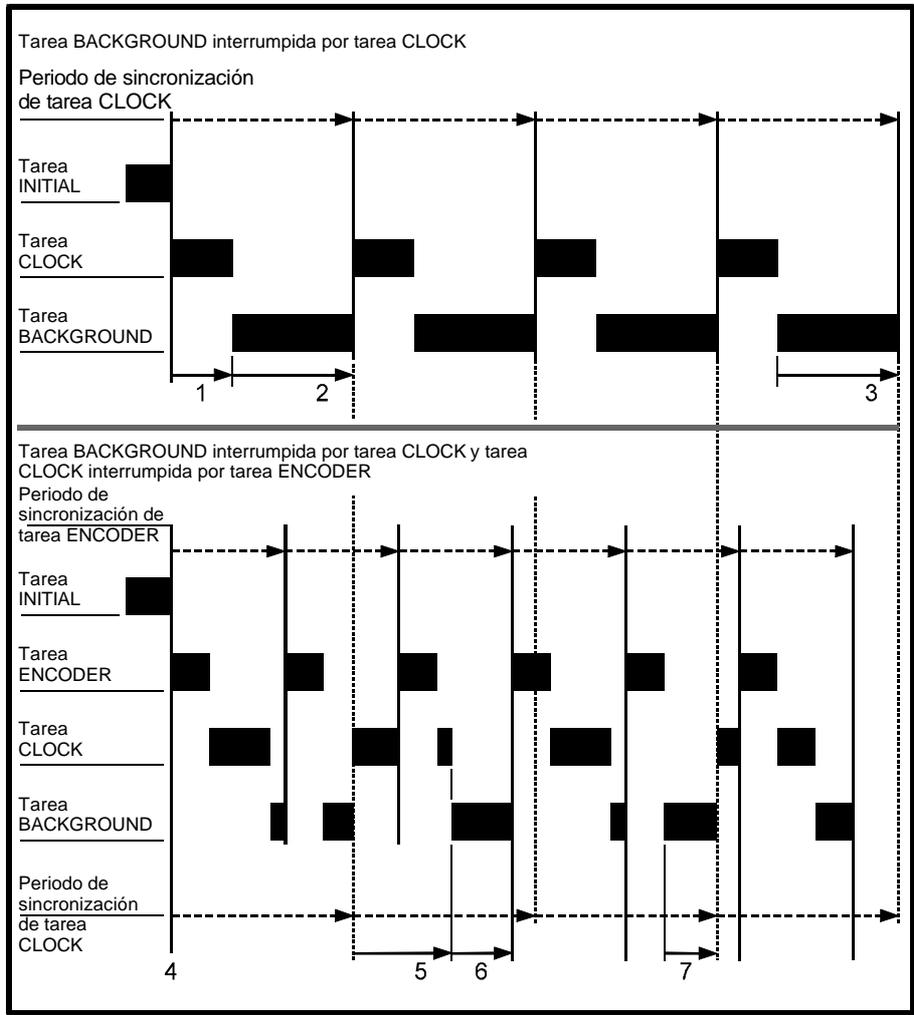
### **Nota**

**La tarea BACKGROUND no se repite automáticamente.**

---

#### *Ejemplo*

```
BACKGROUND{
  RAMP:
    #1.21 = 0
    DO WHILE #1.21<1000
      #1.21 = #1.21+1
    LOOP
  GOTO RAMP:
}
```



Ejemplos de tarea BACKGROUND interrumpida por las tareas CLOCK y ENCODER

## Interpretación del diagrama

---

### Tarea BACKGROUND interrumpida por la tarea CLOCK

- 1 La tarea BACKGROUND espera mientras la tarea CLOCK se ejecuta y se interrumpe cuando se inicia la siguiente tarea CLOCK.
- 2 La tarea BACKGROUND continúa ejecutándose hasta que la tarea CLOCK vuelve a interrumpirla.
- 3 La tarea BACKGROUND concluye.

### Tarea BACKGROUND interrumpida por las tareas ENCODER y CLOCK

- 4 ENCODER y CLOCK se inician.
- 5 La tarea CLOCK se ejecuta hasta que es interrumpida por la siguiente tarea ENCODER. La tarea CLOCK se completa una vez terminada la tarea ENCODER.
- 6 La tarea CLOCK concluye, y todavía queda tiempo suficiente para que la tarea BACKGROUND se ejecute hasta que la siguiente tarea ENCODER la interrumpa.
- 7 Dado que la tarea ENCODER concluye antes de que se inicie el siguiente periodo de CLOCK, la tarea BACKGROUND se ejecuta durante ese intervalo hasta que la interrumpe la siguiente tarea CLOCK.

## Subrutinas definidas por el usuario

---

Subrutinas que escribe el usuario y que se utilizan junto con la instrucción CALL (consulte CALL en el Capítulo 7, Referencia).

De la misma forma que no existen restricciones en cuanto a la asignación de nombre, estas subrutinas pueden insertarse en cualquier parte del programa. (Recuerde que en el nombre de las tareas se distingue entre mayúsculas y minúsculas.)

La subrutina siguiente realiza la misma función que el programa Sawtooth que se ha descrito en el Capítulo 3, Procedimientos iniciales. El nombre que se ha asignado a esta subrutina es **RAMP**:

```
BACKGROUND{
  Loop:
    CALL RAMP:
  GOTO Loop:
}
RAMP: {
  #1.21=0
  DO WHILE #1.21<1000
    #1.21=#1.21+1
  LOOP
}
```

### **Nota importante**

**Es preciso evitar que el inicio de dos tareas en tiempo real distintas inicien una subrutina definida por el usuario (esta situación se denomina reentrada).**

**Por ejemplo, las tareas BACKGROUND y Clock pueden iniciar una subrutina. Si la tarea BACKGROUND inicia la subrutina y la tarea CLOCK interrumpe la tarea BACKGROUND mientras la subrutina se está ejecutando, los valores de la variable que se está procesando podrían cambiar. Esto puede deberse a que la tarea CLOCK también ejecutará la subrutina aplicando otros valores.**

---

### **Tarea CLOCK**

---

La tarea CLOCK sirve para controlar el accionamiento, y los comandos enviados a éste, basándose en el tiempo (por ejemplo, rampa de aceleración o deceleración controlada).

La prioridad de esta tarea es secundaria y sólo puede activarse una vez que la tarea BACKGROUND se ha interrumpido.

Aunque la tarea se ejecuta en una base de tiempos constante, la selección de la base de tiempos que se va a utilizar depende del valor del parámetro de configuración del accionamiento (consulte también los parámetros de configuración del UD70 en el Capítulo 10, Parámetros), que puede ajustarse en un rango 5 ms - 200 ms.

**Ejemplo** En este ejemplo se genera una onda senoidal.

```
CLOCK{
  #1.21 = SIN (rad)*1000
  rad =rad+0.01
  IF rad > 6.283185 THEN ; 6.283185 = 2 * pi
    rad = 0
  ENDF
}
```

## Tarea ENCODER

---

La tarea ENCODER se utiliza principalmente para controlar la actividad de un codificador.

La tarea ENCODER se sincroniza con la tarea SPEED dentro del accionamiento, y se ejecuta cada cuatro ciclos de la tarea SPEED. La periodicidad con la que se ejecuta esta tarea depende de la frecuencia de conmutación seleccionada.

Accionamiento. Es posible utilizar un parámetro de configuración para duplicar el tiempo.

| Accionamiento | Frecuencia de conmutación | Base de tiempos de la tarea ENCODER |
|---------------|---------------------------|-------------------------------------|
|               | kHz                       | ms                                  |
| Unidrive      | 3, 6 o 12                 | 5,52                                |
| Unidrive      | 4,5 o 9                   | 7,36                                |

### Ejemplo

```
ENCODER{
  nuevo_pos_princip% = #90.1
  nuevo_pos_sec% = #90.3
  pos_princip_cambio% = nuevo_pos_princip% - anterior_pos_princip%
  pos_secun_cambio% = nuevo_pos_secun% - anterior_pos_secun%
  EPOS = EPOS + pos_princip_cambio% - pos_secun_cambio%
  anterior_pos_princip% = nuevo_pos_princip%
  anterior_pos_secun% = nuevo_pos_secun%
}
```

## Tarea SPEED

---

La tarea SPEED está sincronizada con el bucle de control de velocidad del Unidrive, por lo que la periodicidad con la que se ejecuta depende de la frecuencia de conmutación seleccionada.

| Accionamiento | Frecuencia de conmutación | Base de tiempos de la tarea SPEED |
|---------------|---------------------------|-----------------------------------|
|               | ms                        | kHz                               |
| Unidrive      | 3, 6 o 12                 | 1,38                              |
| Unidrive      | 4,5 o 9                   | 1,84                              |

## Tarea EVENT

---

La tarea EVENT se ejecuta cuando ocurre un evento específico. La unidad de temporización/contador determina el origen del evento.

Mientras el programa se está ejecutando la tarea EVENT tiene prioridad máxima, por lo que las demás tareas se interrumpen ante una tarea EVENT.

Para obtener más información, consulte la sección dedicada a la unidad de temporización/contador en el Capítulo 8, Funciones.

## Tarea ERROR

---

La tarea ERROR sólo se ejecuta cuando ocurre un error en tiempo de ejecución en el programa DPL. Si el juego de herramientas DPL se encuentra conectado al UD70 cuando se produce el error, el número del error aparece en pantalla.

Los errores en tiempo de ejecución pueden producirse en varias situaciones. Por ejemplo:

- Intento de escritura en un parámetro de sólo lectura
- Ejecución por exceso de una tarea en tiempo real

Aunque los errores suelen deberse a errores de programación, a veces se generan a causa de influencias externas. Por ejemplo, si los datos procedentes de la caja de E/S se pierden a causa de la ruptura del cable, puede aparecer un error que indica la interrupción de comunicaciones serie. En estos casos, el UD70 suele suspender todas las tareas, e incluso puede desconectar el accionamiento.

Para evitar que esto ocurra, puede utilizar la tarea ERROR. Ante un error, se aplica la secuencia siguiente:

- 1 Todas las tareas se suspenden.
- 2 El accionamiento se desconecta (si la desconexión está activada). Consulte los parámetros de activación de la desconexión en los parámetros de configuración del UD70 del Capítulo 10, Parámetros.
- 3 El número del error aparece en el parámetro #88.01 del UD70.
- 4 La tarea ERROR se ejecuta. Las instrucciones contenidas en la tarea ERROR pueden determinar la causa del error en tiempo de ejecución y realizar acciones adecuadas, como detener el accionamiento de forma controlada.

Para obtener más información, consulte Control de errores avanzado en el Capítulo 9, Diagnósticos.

## Tarea NOTES

---

Seudotarea que omite el compilador. El creador del programa utiliza la tarea NOTES para facilitar la comprensión del programa al usuario del accionamiento.

**Ejemplo**

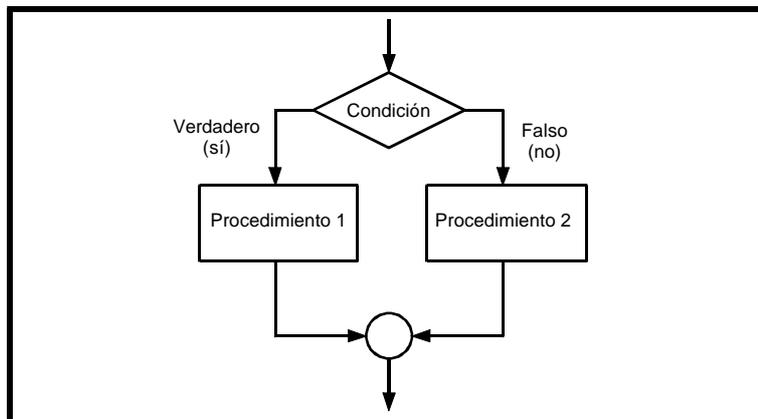
```
NOTES{
  Puede incluir su documentación aquí.
}
```

## 4.7 Instrucciones y funciones

En esta sección se describen los distintos tipos de instrucciones que se utilizan en la programación DPL.

### Instrucciones condicionales

Las instrucciones condicionales realizan una operación en función de una serie de condiciones (por ejemplo, IF).

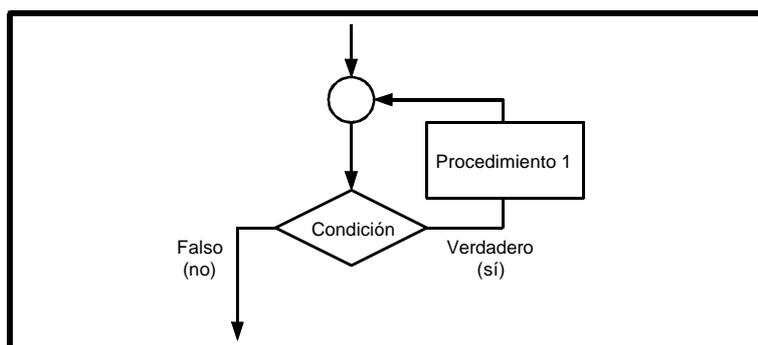


*Ejemplo de diagrama de operaciones lógicas IF, THEN*

### Instrucciones repetitivas

Las instrucciones de este tipo repiten un bloque de instrucciones hasta que se produce una condición específica.

*Ejemplo* DO WHILE  
LOOP



*Ejemplo de diagrama de operaciones lógicas DO WHILE, LOOP*

## **Instrucciones de control de flujo**

---

Estas instrucciones son responsable de que el programa pase directamente a una instrucción especificada o que está por determinarse (por ejemplo, **GOTO**).

## **Funciones matemáticas**

---

Las funciones matemáticas aplican una operación matemática a una expresión para obtener un valor (por ejemplo, **SIN**).

## **Funciones de proceso de señal**

---

Las funciones de proceso de señal permiten obtener un valor a partir de un número de muestras durante un periodo de tiempo establecido. Estas funciones sólo pueden utilizarse en las tareas **SPEED**, **CLOCK** o **ENCODER** (por ejemplo, **FILTER**).

## **Funciones de conversión de base**

---

Estas funciones actúan sobre un valor para convertir un decimal en código binario en binario, y viceversa; resultan útiles para los datos que se reciben desde una caja de E/S. Consulte el Capítulo 6, Comunicaciones serie.

## **Funciones de conversión de datos**

---

Las funciones de conversión de datos transforman una variable de coma flotante en otra entera, y viceversa.

## **Instrucciones ANSI**

---

Estas instrucciones hacen posible que el programa DPL pueda comunicarse a través de un puerto RS485 con otros accionamientos y tarjetas UD70 mediante el protocolo ANSI. Consulte el Capítulo 6, Comunicaciones serie.

## 4.8 Optimización de programas

A fin de ejecutar los programas de forma correcta, se ofrecen algunas recomendaciones.

### Variables enteras

Utilice variables enteras, en lugar de variables de coma flotante, siempre que sea posible. Las operaciones con variables de coma flotantes son 20 veces más lentas que los procesos de las variables enteras. (Consulte la instrucción *INT* en el Capítulo 7, Referencia.)

### Aritmética de coma fija

Utilice siempre la aritmética de coma fija para representar decimales. Por ejemplo, si se requiere una resolución con valor **.001, 1000** representará al valor **1**. Esto permite mantener la precisión en todas las operaciones matemáticas.

El resultado de la expresión debe corregirse mediante un factor divisor.

```
Ejemplo    a% = 1500      //    "a% = 1.5"
              b% = 2500      //    "b% = 2.5"
              c% = a% * b%    //    c% = 3750000
              // Divida entre 1000 para ajustar c%
              c% = c% / 1000  //    "c% = 3750"
              // Para obtener el valor real, es preciso volver a dividir entre 1000
              #1.21 = c% / 1000 //    "c% = 3.75"
```

### Variables enteras provisionales

Para reducir al mínimo el número de veces que se accede a un parámetro, utilice variables enteras provisionales en lugar de acceder al parámetro siempre que necesite su valor. El tiempo de acceso a un parámetro es 50 veces mayor que el que se necesita para acceder a una variable.

```
Ejemplo    IF #1.21 > 100 THEN
              rango% = 1
              ELSEIF #1.21 > 200 THEN
              rango% = 2
              ENDIF
              Se convierte en:
              temp% = #1.21
              IF temp% > 100 THEN
              rango% = 1
              ELSEIF temp% > 200 THEN
              rango% = 2
              ENDIF
```

## División por enteros

---

El uso de la división por enteros puede afectar negativamente a la exactitud del resultado, como se muestra en la expresión:

**Si #1.21 es igual que 5**

**Entonces:**

```
a = 4.5 * (#1.21 /4)
= 4.5 * (5 /4)
= 4.5 * 1
= 4.5
```

El compilador DPL emplea una división por enteros, convierte el resultado en un valor de coma flotante y utiliza un múltiplo de éste.

A fin de mantener la precisión, uno de los enunciados puede convertirse en una variable de coma flotante, como sigue:

```
a = 4.5 * (#1.21 / FLOAT(4) )
= 4.5 * (5 / FLOAT(4) )
= 4.5 * 1.25
= 5.625
```

Consulte la instrucción *FLOAT* en el Capítulo 7, Referencia.

## Instrucción PRINT

---

La instrucción PRINT no debe utilizarse en exceso. (Consulte la instrucción PRINT en el Capítulo 7, Referencia). Es preferible utilizar la ventana **Watch** del juego de herramientas DPL para controlar las variables (consulte el Capítulo 5, Juego de herramientas DPL).

Utilice la instrucción PRINT solamente en la tarea BACKGROUND. Si la instrucción PRINT se incluye en las tareas SPEED, CLOCK o ENCODER, es posible que no disponga de tiempo suficiente para ejecutarse y que no se imprima el texto enviado a la cola de impresión.

## Tarea BACKGROUND

---

Ejecute la mayor parte posible del programa en la tarea BACKGROUND, en lugar de en SPEED, CLOCK, ENCODER u otras tareas en tiempo real. Debido a que las tareas en tiempo real se ejecutan en función de una base de tiempos fija, el proceso debe realizarse durante este tiempo. Sin embargo, la tarea BACKGROUND no impone este tipo de restricciones.

## Instrucción #INT

---

La instrucción #INT permite modificar un parámetro que requiere variables de coma flotante para que acepte variables enteras. Esto permite aumentar la velocidad de la operación de forma significativa.

**Ejemplo**      #4.08 = 14.5  
                  // ajuste #4.08 en 14.5 en Unidrive  
                  //igual que  
                  #INT2.00 = 145  
                  // También se permite la lectura:  
                  valor% = #INT4.08

## 4.9 Apuntadores de parámetro

Un apuntador de parámetro es una variable entera que representa un parámetro del accionamiento.

**Ejemplo**      A% = 121 // ajuste A% para que indique #1.21  
                  #A% = 10//escriba 10 para que pr A% indique (#1.21)

### **Nota**

**Si el parámetro contiene una coma decimal, ésta se ignora. (Por ejemplo, si 0.1 es la unidad utilizada en el parámetro #4.08 del Unidrive, el valor 2.3 debe escribirse como 23.)**

---

## 4.10 Definición de alias (constantes)

A veces resulta útil asignar un nombre descriptivo a un parámetro o a un valor. Por ejemplo:

Es posible referirse al parámetro #1.21 utilizando  
SPEED\_REFERENCELas instrucciones pueden introducirse en el  
siguiente formato:  
SPEED\_REFERENCE = MAX\_SPEED

El alias puede crearse mediante la directiva \$DEFINE. La sintaxis es la siguiente: \$DEFINE nombre.

La directiva \$DEFINE puede utilizarse para asignar el valor necesario al nombre que se empleará después en el programa; el nombre se convierte así en un alias de ese valor. Cuando el programa se compile, el nombre se reemplazará por este valor.

### **Nota**

**No es posible incluir comentarios al final de una línea \$DEFINE.**

---

Un alias consta de las dos partes siguientes:

**Nombre de parámetro** Indica el nombre que se va a definir, que puede constar de cualquier combinación de letras, dígitos y caracteres de subrayados, sin espacios.

**Valor de parámetro** Permite especificar cualquier valor constante o número de parámetro.

*Ejemplo* En este ejemplo se muestra el uso de la directiva `$define` para asignar nombres a los números de parámetro (`#3.02` y `#1.21`) y a un valor (`500`).

```
$define MAX_SPEED 500
$define SPEED #3.02
$define SPEED_DEMAND #1.21
```

```
BACKGROUND{
top:
IF SPEED < MAX_SPEED THEN
    SPEED_DEMAND = SPEED_DEMAND + 1
ENDIF
GOTO top:
}
```



---

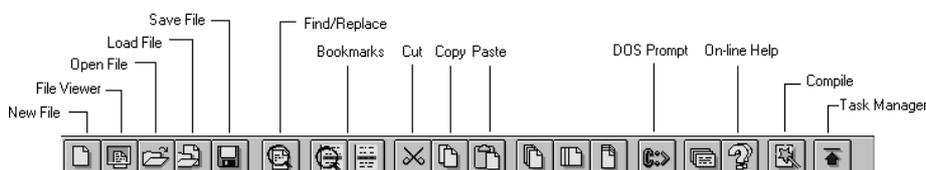
## 5 Juego de herramientas DPL

---

En este capítulo se describen el funcionamiento del juego de herramientas DPL, el proceso de compilación de programas y las utilidades de depuración.

### 5.1 Descripción del juego de herramientas DPL

El juego de herramientas DPL permite al usuario del UD70 modificar, escribir y descargar programas en el UD70 e incorpora una serie de herramientas de compilación, además de un depurador y un editor completos.



#### *Barra de herramientas principal del juego de herramientas DPL*

Las herramientas de compilación permiten al usuario realizar lo siguiente:

- Desarrollar y modificar programas en tiempo real para el UD70.
- Cortar y copiar texto de programas en el portapapeles de Windows.
- Pegar texto de programas desde el portapapeles de Windows.
- Cargar un programa existente desde el UD70.
- Compilar el programa en código máquina.

La utilidad de depuración incluye herramientas que permiten realizar lo siguiente:

- Visualizar los valores de los parámetros del accionamiento en la pantalla y modificarlos mientras el programa DPL se está ejecutando.
- Visualizar los valores de los parámetros de las variables en la pantalla y modificarlos mientras el programa DPL se está ejecutando.
- Utilizar el modo paso a paso para comprobar el programa.
- Insertar puntos de interrupción.

#### **Nota**

**En el UD70 no pueden almacenarse varios programas simultáneamente.**

---

## 5.2 Administración de archivos

Los principios utilizados en la administración de archivos del juego de herramientas DPL son similares a los de las aplicaciones Windows. Además de los procedimientos habituales, existen otros específicos de este juego de herramientas que se explican a continuación.

### Menú File

---

El menú File es el siguiente:

|                                 |        |
|---------------------------------|--------|
| <b>N</b> ew                     | Ctrl+N |
| <b>O</b> pen...                 | Ctrl+O |
| <b>L</b> oad...                 |        |
| <b>R</b> eload                  |        |
| <b>S</b> ave                    | Ctrl+S |
| <b>S</b> ave <b>A</b> s...      |        |
| <b>C</b> lose                   |        |
| <b>V</b> iew Current File       | Ctrl+I |
| <b>V</b> iew <b>L</b> ast File  |        |
| <b>P</b> rint...                |        |
| <b>P</b> rinter Setup...        |        |
| <b>P</b> rinter <b>F</b> ont... |        |
| <b>A</b> dd to                  | ▶      |
| <b>R</b> emove from             | ▶      |
| <b>U</b> se As                  | ▶      |
| <b>E</b> xit                    |        |

#### Creación de un archivo nuevo

En el menú File, seleccione New o haga clic en  para que se cree una página en blanco en la que pueda empezar a trabajar.

#### Apertura de un archivo existente

Los dos métodos que pueden utilizarse para abrir un archivo son los siguientes:

##### Apertura en una ventana nueva

En el menú File, seleccione Open...o haga clic en 

##### Apertura en la ventana existente con sustitución del contenido actual

En el menú File, seleccione Load... o haga clic en 

**Nueva apertura del último archivo guardado** Almacenamiento de un archivo

**En el menú File, seleccione Reload.** En el menú **File**, seleccione **Save As...** o haga clic en 

Para que resulte posible compilarlos, los archivos DPL deben guardarse con un nombre de archivo que tenga la extensión .DPL.

**Incorporación de un nombre de archivo en un menú** La opción **Add to** permite incorporar archivos en un menú para que facilitar el acceso. Al seleccionar esta opción, aparece la lista siguiente.

Add to Favourite  
Add to Cue Cards

En esta lista se incluyen dos de los elementos de menú principales del juego de herramientas: **Favourite** y **Cue Cards**. Al seleccionar uno de estos menús, los archivos añadidos aparecen en otro menú desplegable, desde donde pueden seleccionarse inmediatamente.

Las opciones son las siguientes:

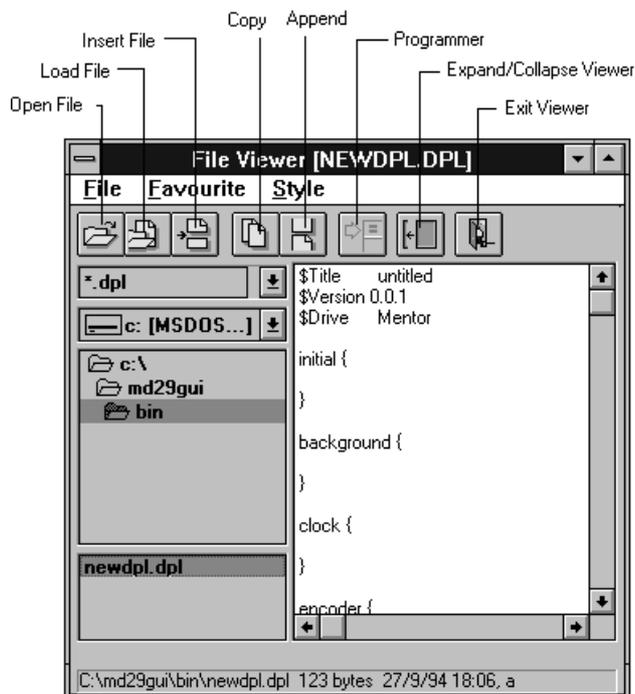
**Add to favourite**

El archivo abierto se añade al menú Favourite.

**Add to Cue Cards**

El archivo abierto se añade a Cue cards en el menú Help.

## File Viewer



File Viewer permite al usuario realizar lo siguiente:

- Visualizar un archivo sin abrirlo
- Copiar texto de un archivo cerrado y pegarlo en el archivo abierto
- Preseleccionar cada línea por separado para copiarlas en una sola operación

### Apertura de File Viewer

Realice una de las operaciones siguientes:

Haga clic en **File Viewer** (File Viewer).

En el menú **File**, seleccione **View Current File** o **View Last File**.

Al seleccionar **View Current File**, File Viewer se abre y el archivo actualmente abierto aparece en pantalla.

Si se selecciona **View last file**, File Viewer carga el último archivo guardado. Este archivo puede ser distinto del mostrado en pantalla.

**Procedimiento para copiar y pegar texto**

- 1 Seleccione el archivo y resalte el texto que desee copiar.
- 2 Haga clic en  (Copy).
- 3 Coloque el cursor en la posición en la que desee situar el texto seleccionado.
- 4 Abra el menú **Edit** y seleccione **Paste** en el menú desplegable.

**Procedimiento para copiar y pegar subrutinas**

Para seleccionar una subrutina y pegarla en programas diferentes, utilice uno de estos procedimientos.

**Utilización de File Viewer**

- 1 Guarde la subrutina como un archivo. Consulte el Capítulo 4, Programación DPL.
- 2 Coloque el cursor en el programa abierto, en la posición en la que desee insertar el texto.
- 3 Abra File Viewer. El cuadro situado en la esquina inferior izquierda de File Viewer contiene una lista de los archivos guardados.
- 4 Seleccione el nombre del archivo que contiene la subrutina elegida.
- 5 Haga clic en  (Viewer Insert).

**Utilización de la barra de herramientas principal**

- 1 Guarde la subrutina como un archivo. Consulte el Capítulo 4, Programación DPL.
- 2 Coloque el cursor en el programa abierto, en la posición en la que desee insertar el texto.
- 3 Abra el menú **Edit** y seleccione **Insert File** en el menú desplegable.

## 5.3 Modificación de un programa

### Menú Edit

---

El menú Edit contiene lo siguiente:

| <u>U</u> ndo                             | <u>A</u> lt+Bksp |
|--|------------------|
| C <u>u</u> t                             | Ctrl+X           |
| C <u>u</u> t <u>L</u> ine                | Ctrl+D           |
| <u>C</u> opy                             | Ctrl+C           |
| <u>P</u> aste                            | Ctrl+V           |
| <u>A</u> ppend                           | Ctrl+K           |
| <u>C</u> lear                            |                  |
| <u>F</u> ind / <u>R</u> eplace           | Ctrl+F           |
| <u>G</u> o To Line...                    |                  |
| <u>G</u> o To Next <u>B</u> ookmark      | Ctrl+G           |
| <u>T</u> oggle <u>B</u> ook <u>M</u> ark | Ctrl+M           |
| <u>C</u> lear All <u>B</u> ookmarks      |                  |
| <u>T</u> oggle <u>C</u> ase              |                  |
| <u>I</u> nsert File...                   |                  |
| <u>I</u> nsert Time/Date                 |                  |
| <u>S</u> elect All                       |                  |

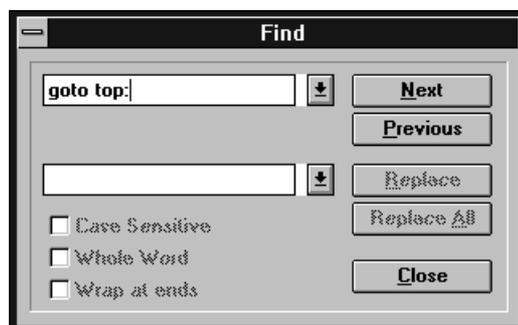
Las herramientas de edición básicas son similares a las contenidas en las aplicaciones Windows, y permiten cortar, copiar, pegar y deshacer operaciones.

#### Corte de una línea

Seleccione Cut Line para eliminar las líneas de instrucciones resaltadas.

#### Búsqueda y sustitución de texto

Para reemplazar caracteres y palabras, seleccione Find/Replace. Se abrirá el cuadro de diálogo **Find**.



La opción Find/Replace se utiliza de la misma manera que en los procesadores de texto de Windows.

#### **Anexo de líneas de instruccione**

Para copiar líneas de File Viewer en el programa que se está creando y mantener un orden específico, realice lo siguiente:

- 1 Seleccione por orden las líneas mostradas en File Viewer que desee copiar en el programa nuevo.
- 2 Después de realizar cada selección, haga clic en  (**Append**).
- 3 Coloque el cursor en la posición en la que desee situar las líneas dentro del programa nuevo.
- 4 Abra el menú **Edit** y seleccione **Paste** en el menú desplegable.

#### **Marcadores**

---

Los marcadores resultan útiles durante la negociación de programas largos. El creador del programa puede insertar marcadores en las posiciones de referencia que se utilicen con frecuencia. Un solo programa puede incluir una serie de marcadores.

#### **Inserción de un marcador**

- 1 Coloque el cursor en la posición del programa en la que desee insertar el marcador.
- 2 Haga clic en  (**BookMark**).

#### **Búsqueda de un marcador**

- 1 Haga clic en  (**Next BookMark**).
- 2 El cursor se desplaza hasta el marcador que se ha insertado. Al hacer clic en **Next BookMark** otra vez, el cursor resalta el siguiente marcador. Los marcadores se resaltan en el orden en el que se han insertado.

#### **Borrado de marcadores**

Abra el menú **Edit** y seleccione **Clear All BookMarks**.

## 5.4 Aplicación de estilos

El menú Style contiene lo siguiente:

|                      |          |
|----------------------|----------|
| √ Style 1 (Normal)   | Shift+F1 |
| Style 2              | Shift+F2 |
| Style 3              | Shift+F3 |
| Font...              |          |
| Text Colour...       |          |
| Background Colour... |          |
| √ Autoindent         |          |
| Tab stops            | ▶        |
| Save Settings As ... | ▶        |

### Styles

---

Permite modificar la presentación de la pantalla del juego de herramientas DPL. Con los 48 colores de fondo y de texto que existen pueden realizarse más de 2000 combinaciones de colores.

Font incluye una amplia lista de fuentes de texto, incluidas las fuentes TrueType.

### Auto-indent

---

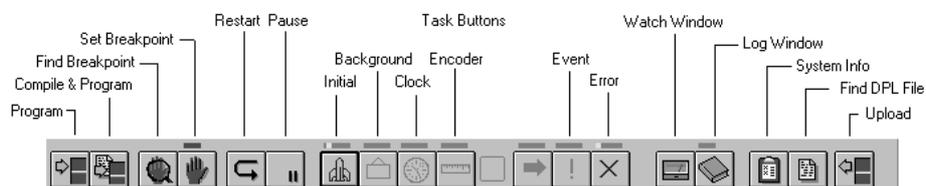
Permite introducir un tabulador para los comandos DO WHILE...LOOP y IF...ENDIF. Al presionar la tecla Intro al final de la línea, el sangrado se aplica automáticamente a la línea siguiente. Para eliminar el sangrado, presione la tecla de retroceso.

Con este método de sangrado, las faltas de concordancia en la programación pueden detectarse fácilmente, ya que permite asegurarse de que un enunciado IF finaliza con otro ENDIF y DO termina con LOOP (consulte el Capítulo 7, Referencia).

## 5.5 Compilación y ejecución de un programa

### Barra de herramientas de Task Manager

El administrador de tareas (Task Manager) del juego de herramientas DPL contiene potentes herramientas de compilación y depuración que permiten al programador comprobar el programa de forma exhaustiva. Mientras unas herramientas de depuración son automáticas y detectan errores de programación, otras facilitan al programador la comprobación de la lógica del programa línea a línea.

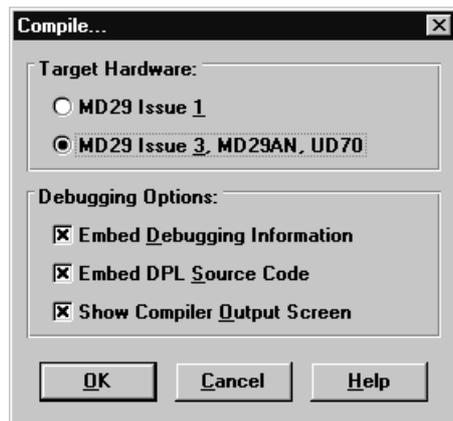


### Compilación de un programa

- 1 Guarde el programa que ha escrito como un archivo .DPL
- 2 Haga clic en  (Compile). Si desea compilar y cargar el programa automáticamente, haga clic en:



Se abre el cuadro de diálogo Compile...



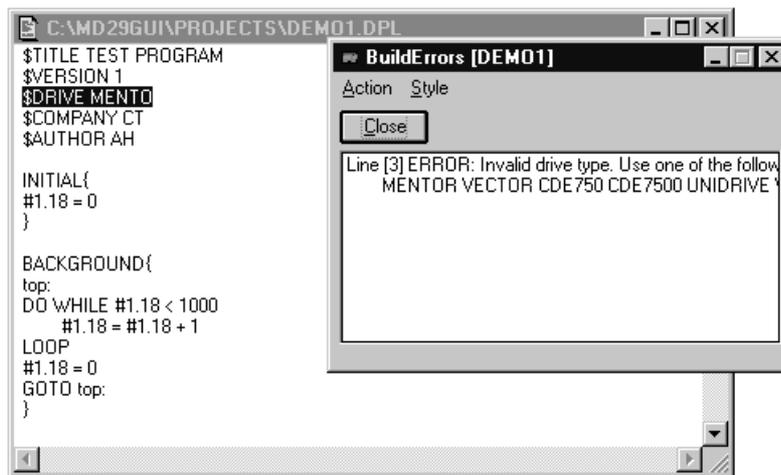
Normalmente, las opciones pueden permanecer tal como aparecen. Para continuar compilando, haga clic en:



## Opciones de depuración

En las opciones de depuración se incluyen varias funciones que permiten cancelar la selección del programa DPL. Consulte la descripción de estas opciones a continuación en Control y depuración del programa.

Si se detectan errores al compilar el archivo, se abre una ventana **Build Errors** con los errores del programa. Conforme se resalta cada uno de los errores en **Build Errors**, también se indica la línea correspondiente del programa.



- Utilización de Build Errors**
- 1 Haga clic en la primera línea que aparece como un error.
  - 2 Corrija el error (el tipo de error se indica en la ventana **Build Errors**).
  - 3 Haga clic en .
  - 4 Haga clic  en el cuadro de diálogo **Save Changes** que aparece.
  - 5 Para actualizar el programa compilado, repita el proceso de compilación.

### **Nota**

**Las herramientas de depuración del compilador resaltan los fallos del programa que se atribuyen a errores de programación, en lugar de indicar problemas debidos a la lógica del programa. Si un programa parece que no funciona después de compilarlo y transferirlo al UD70, es probable que presente problemas de lógica.**

## Errores y advertencias

Un error indica que el compilador no ha podido interpretar una línea o un comando del programa DPL. Esto suele ocurrir si un comando presenta errores ortográficos, si se utiliza incorrectamente, etc.

Una advertencia indica que el compilador ha interpretado los comandos, pero el código no funciona de la forma prevista. La advertencia más frecuente es Possible loss of accuracy in assignment (posible pérdida de precisión en la asignación) y suele aparecer cuando se asigna un valor de coma flotante a variables o parámetros enteros.

El Capítulo 9, Diagnósticos, incluye una lista detallada de los errores y advertencias.

## 5.6 Transferencia de un programa

Los programas sólo pueden transferirse cuando no contienen errores. El programa DPL puede ejecutarse después de transferir el archivo de programa al UD70.

Para transferir el archivo, haga clic en  (Down-load) en la barra de herramientas de Task Manager.



**Advertencia**

**Si selecciona Auto-Run, el programa se ejecutará automáticamente después de la transferencia y el UD70 se inicializará (lectura de los parámetros de configuración).**

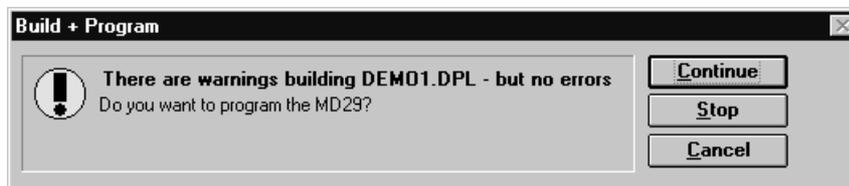
## Compilación y transferencia de un programa

Para compilar y transferir un programa en una sola operación, haga clic en



De forma alternativa, puede seleccionar **Quick Start** en el menú **Project**.

El proceso de transferencia no se realiza cuando se detectan errores. Sin embargo, si el programa sólo contiene advertencias, aparece el cuadro de diálogo **Build + Program**.



Para ignorar las advertencias y continuar con la transferencia, haga clic en el botón **Continue**; de lo contrario, haga clic en el botón **Stop**.

## 5.7 Ejecución de un programa

Haga clic en  (Restart) para ejecutar el programa DPL desde el principio (es decir, a partir de la tarea INITIAL). Los botones del administrador de tareas que corresponden a la tarea actual del programa se activan.

### Interrupción del programa

---

Haga clic en  (Pause).

### Reanudación del programa

---

Vuelva a hacer clic en  (Pause).

## 5.8 Utilidades de control y depuración del programa

### Single-stepping

---

Las líneas del programa DPL se ejecutan de una en una. Mediante el uso de esta utilidad, es posible controlar el funcionamiento del programa instrucción a instrucción. Las demás tareas pueden ejecutarse a toda velocidad durante este proceso.

#### Aplicación de Single-stepping a una tarea

1 Haga clic en el botón correspondiente de Task Manager, como se indica:



Tarea INITIAL



Tarea BACKGROUND



Tarea CLOCK



Tarea ENCODER



Tarea SPEED

Esto permite detener la tarea seleccionada. (No es posible aplicar este proceso a la tarea EVENT.)

2 Al hacer clic en el botón correspondiente de Task Manager varias veces, el punto de ejecución avanzará hasta la línea siguiente.

#### Borrado de Single-stepping

Para eliminar la función Single-stepping, utilice uno de los métodos siguientes:

- Mantenga presionada la tecla **Ctrl** y haga clic en el botón correspondiente de Task Manager.
- Abra el menú **Run** y seleccione la opción **Run Task** adecuada.

## Puntos de interrupción

---

Un punto de interrupción es una línea de una tarea en la que la tarea interrumpe la ejecución del programa y pasa al modo paso a paso (single-stepping).

Los puntos de interrupción son útiles para comprobar el momento en el que un programa se encuentra en una parte concreta de un código, o para verificar el estado de las variables DPL en un punto determinado.

- Inserción de un punto de interrupción**
- 1 Coloque el cursor en la línea en la que se desee insertar el punto de interrupción.
  - 2 Haga clic en  (Set breakpoint) para activar el punto de interrupción. Cuando el programa llega a la línea definida durante la ejecución, la tarea se detiene y se inicia el modo paso a paso.

### **Nota**

**Un punto de interrupción sólo puede corresponder a un momento determinado de una tarea. En las subrutinas no se permite el uso de puntos de interrupción ni de operaciones paso a paso.**

---

### **Búsqueda de puntos de interrupción en tareas independientes**

- 1 Haga clic en  (Next breakpoint). El cursor se desplaza hasta el punto de interrupción siguiente del programa DPL.

### **Eliminación de puntos de interrupción**

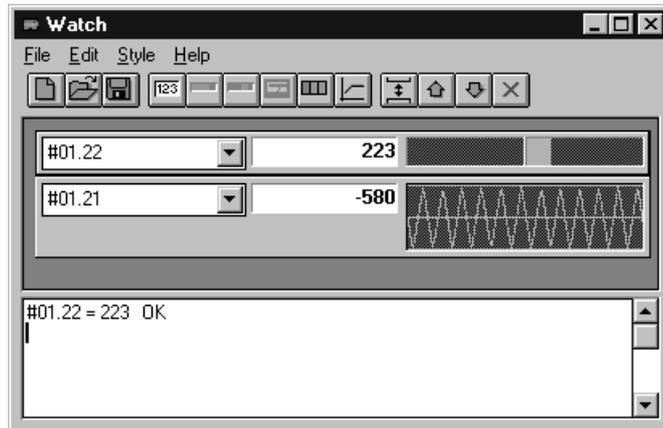
- 1 Coloque el cursor en la línea en la que se encuentra el punto de interrupción que desee eliminar, o utilice el botón **Next breakpoint** para buscar la línea que incluye dicho punto.
- 2 Haga clic en el botón **Set breakpoint**.

## Utilización de la ventana Watch

---

La ventana **Watch** permite al programador comprobar la lógica del programa mientras se ejecuta en el UD70 mediante la lectura y escritura de parámetros y variables.

Para mostrar la ventana **Watch**, haga clic en  en la barra de herramientas de Task Manager.



**Ejemplo de presentación en la ventana Watch**

La ventana **Watch** está dividida en dos secciones. La parte superior muestra los valores de los parámetros y las variables, que se actualizan **continuamente** mientras se ejecuta el programa. Los valores pueden mostrarse de forma numérica y gráfica. La parte inferior permite visualizar e introducir parámetros y variables de programa de forma instantánea.

**Control continuo de una variable o un parámetro**

Un elemento puede visualizarse en la parte superior de la ventana **Watch** de las cinco formas siguientes:

| Método de visualización                     | Botón |
|---|-------|
| Sólo valor                                  |       |
| Valor con gráfico de barras unipolar        |       |
| Valor con gráfico de barras bipolar         |       |
| Valor con gráfico de líneas bipolar         |       |
| Valor con presentación de bits individuales |       |

Para controlar un parámetro o una variable, utilice el siguiente procedimiento:

- 1 Haga clic en el botón de la barra de herramientas que corresponda al método de visualización deseado.
- 2 Seleccione el parámetro o la variable que desee visualizar mediante uno de los métodos siguientes:
  - En el cuadro de texto de parámetro de la izquierda, escriba el nombre del parámetro o la variable que desee visualizar.
  - Haga clic en  y seleccione el parámetro o la variable deseado en la lista que aparece.
  - En la ventana principal del editor del juego de herramientas DPL, haga doble clic en el nombre del parámetro o la variable.

### **Cambio del valor a plena escala**

El valor de representación gráfica a plena escala es 1000 de forma predeterminada. Para modificar este valor, haga doble clic en el cuadro de presentación de valores. Introduzca el valor a plena escala deseado en el cuadro de diálogo **Max Value** que se abre y, a continuación, haga clic en el botón **OK**.

### **Alternancia de valores bipolares**

Para cambiar la representación gráfica de un elemento visualizado a otro tipo (por ejemplo, de barras unipolar a bipolar), desplace el cursor del ratón sobre el área de representación gráfica y presione el botón derecho del ratón. A continuación, seleccione la opción deseada en el menú emergente que aparece.

**Utilización de la sección inferior de la ventana Watch** La parte inferior de la ventana **Watch** permite al usuario realizar una lectura instantánea de un parámetro o una variable, así como escribir en cualquier parámetro o variable.

### **Lectura de un parámetro o una variable**

- 1 Escriba el nombre del parámetro o la variable (ejemplo, #1.21).
- 2 Presione la tecla Intro. El valor se muestra a la derecha del nombre del parámetro o la variable.

### **Configuración de un valor para un parámetro o una variable**

- 1 Escriba el nombre del parámetro o la variable seguido de un signo de igualdad y del valor que quiera introducir (ejemplo, #1.21 = 1000).
- 2 Presione la tecla Intro. Si el valor se ha introducido correctamente, **OK** aparece a la derecha.

Para insertar automáticamente un parámetro o una variable de la parte inferior de la ventana en la superior, presione las teclas Mayús e Intro después de escribir el nombre.

**Nota**

**Si el programa se modifica, será preciso guardarlo, compilarlo y transferirlo de nuevo al UD70.**

**Almacenamiento de la configuración de la ventana Watch**

La configuración de la ventana **Watch** puede almacenarse en disco para utilizarla posteriormente. Para esto, seleccione **Save** en el menú **File** de la ventana **Watch**.

Si desea obtener información detallada sobre la ventana **Watch**, consulte la ayuda en línea. Para visualizar esta ayuda, presione la tecla **F1**.

**Transferencia del archivo DPL origen desde el UD70 al equipo host**

El archivo .DPL origen puede transferirse al equipo host cuando el archivo de programa original no se encuentra disponible o el programa que se está ejecutando no se identifica de forma precisa.

Para recuperar el archivo .DPL, utilice el procedimiento siguiente:

- 1 Haga clic en  (**Upload**) en la barra de herramientas DPL del administrador de tareas. Si se ha instalado una versión del programa DPL en el equipo host, aparece un cuadro de diálogo en el que se solicita confirmación para sobrescribirlo.
- 2 Para averiguar cuál es el programa residente en el UD70 e intentar cargarlo, haga clic en:



(Source code)

**Nota importante**

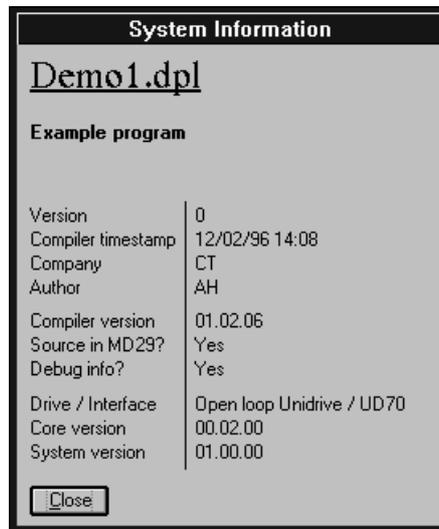
**Es más seguro transferir el programa \*.DPL desde el UD70 que recuperar el código de origen del equipo host, a menos que haya verificado que el programa instalado en el host es el que necesita.**

## Información del sistema

---

Proporciona información sobre el programa del UD70 al usuario, como el nombre de programa, la fecha de compilación, el número de versión del sistema operativo UD70, etc.

Para ver esta información, haga clic en  (System information). Se abre el cuadro de mensajes System Information:



## La ventana Log

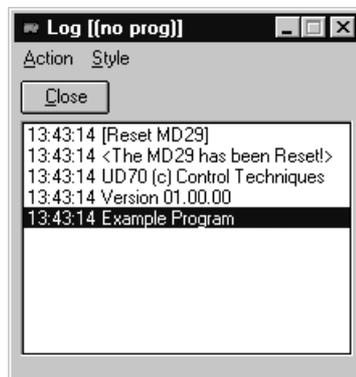
---

En la ventana Log puede mostrarse la siguiente información:

- Mensajes del sistema (ejemplo, el inicio y parada de un programa)
- Valores de la ventana Watch (útil para el registro de datos)
- Resultado de la instrucción DPL PRINT (consulte la instrucción PRINT en el Capítulo 7, Referencia).

Estas funciones se activan y desactivan en el menú **Action**.

Para acceder a la ventana Log, haga clic en  (**Log window**). La ventana Log se abre:



---

## 6 Comunicaciones serie

---

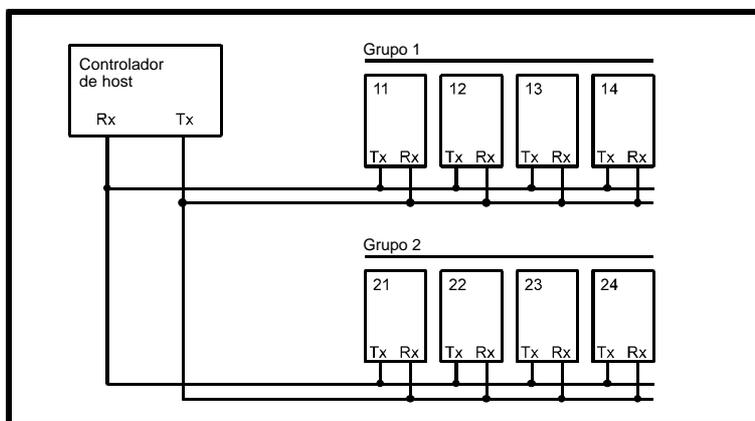
### 6.1 Introducción

El enlace de comunicaciones serie permite utilizar una o varias tarjetas UD70 en sistemas controlados por una unidad host, como un PLC o un PC. En las comunicaciones serie se utiliza el enlace RS485 estándar.

El UD70 también puede funcionar como host en un sistema y controlar los accionamiento, los módulos UD70, las tarjetas MD29 u otros dispositivos instalados que tengan una interfaz adecuada.

El controlador de host puede funcionar con un máximo de treinta y dos dispositivos EIA RS485 mediante el uso de repetidores de línea. Cada transmisor y receptor de los dispositivos de Control Techniques carga dos unidades en la línea. Por consiguiente, estos dispositivos cargan cuatro unidades en la línea en el modo de dos hilos. Esto significa que no es posible conectar más de siete de estos dispositivos en un solo grupo, lo que supone una carga máxima de cuatro unidades en el repetidor de línea. Cuando se utiliza el modo de cuatro hilos, pueden conectarse un máximo de 15 dispositivos.

Sin embargo, el uso de repetidores de línea hace posible el funcionamiento con un máximo de 81 dispositivos de Control Techniques. En este caso, los dispositivos se organizan en 9 grupos de 9 dispositivos, como máximo. Además, es posible aplicar comandos a uno o varios grupos específicos sin que esto afecte a otros dispositivos o grupos de dispositivos.



*Enlace multiterminal RS485 con dos grupos de cuatro unidades*

El puerto de comunicaciones del UD70 es el conector macho tipo D que está situado en el lado derecho de la placa. El UD70 puede utilizarse en el modo de 2 o 4 hilos. El puerto RS485 dispone de aislamiento óptico completo. También es posible utilizar el puerto RS422.

**Precaución**

**Aunque es posible realizar una conexión RS232 con el puerto RS485, no es recomendable debido a que sus características son inferiores (eliminación de ruidos, longitud máxima del cable limitada, etc). RS232 no equivale a RS485 de dos hilos.**

## 6.2 Conexiones de hardware

En la tabla siguiente se especifican las conexiones de hardware del puerto de comunicaciones RS485.

| Clavija | RS485 4 hilos          | RS485 2 hilos             |
|---------|------------------------|---------------------------|
| 1       | 0V                     | 0V                        |
| 2       | $\overline{\text{Tx}}$ | $\overline{\text{Tx/Rx}}$ |
| 3       | $\overline{\text{Rx}}$ | $\overline{\text{Tx/Rx}}$ |
| 4       | DIO *                  | DIO *                     |
| 5       | DII *                  | DII *                     |
| 6       | Tx                     | Tx/Rx                     |
| 7       | Rx                     | Tx/Rx                     |
| 8       | DO *                   | DO *                      |
| 9       | OVD *                  | OVD *                     |

\* Terminales 4, 5, 8 y 9 de las conexiones de E/S digitales del UD70. Al no formar parte de las conexiones de las comunicaciones serie, no precisan conectarse a líneas de este tipo ni a la clavija de 0 V de las comunicaciones serie (clavija 1).

### Conexión a tierra

Se recomienda conectar el cable de comunicaciones de datos a una toma de tierra sin interferencias a través de un circuito de baja inductancia.

### Tendido del cable de comunicaciones serie

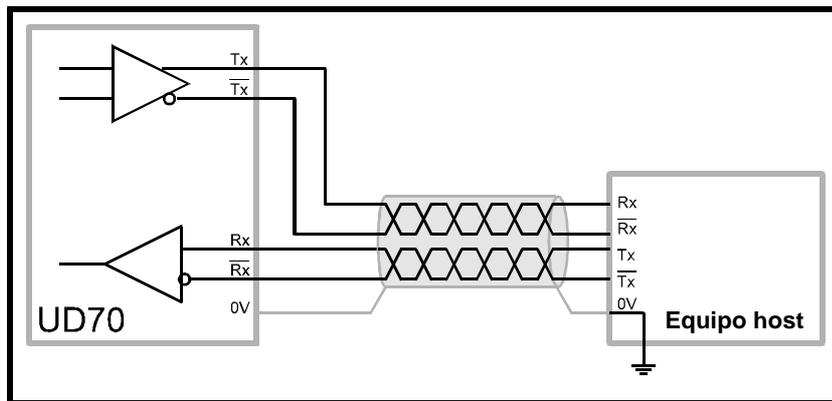
El cable de comunicaciones de datos no debe tener una trayectoria paralela a la de otros cables de alimentación, especialmente los que conectan los accionamientos a motores. Si el tendido en paralelo es inevitable, asegúrese de que quede un espacio mínimo de 300 mm entre el cable de comunicaciones y el de alimentación.

Los cables cruzados en ángulo recto suelen causar problemas.

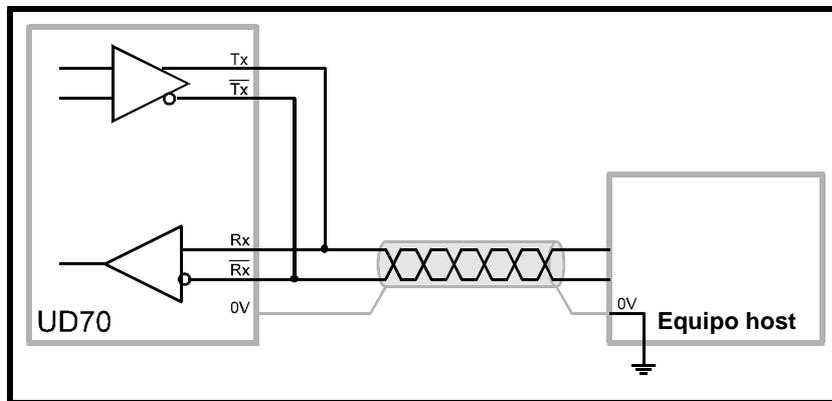
El cable de un enlace RS485 debe tener una longitud máxima de 1200 metros.

## Terminación del cable

Si utiliza un sistema RS485 multiterminal, conecte una resistencia de 120 entre las dos líneas de recepción de la última unidad de la cadena (es decir, la unidad más alejada del host). Verifique que la resistencia no se encuentra instalada en las demás unidades del sistema. Si se conectan resistencias terminales en unidades distintas de la última, puede producirse una pérdida excesiva de señal.



*Conexiones en el modo de 4 hilos*



*Conexiones en el modo de 2 hilos*

## 6.3 Comunicaciones ANSI

### Utilización del protocolo secundario ANSI estándar

Protocolo estándar integrado que define la estructura de los mensajes utilizada para leer y escribir parámetros en el UD70 es el protocolo ANSI x3.28-2.5-A4. Este protocolo se describe en esta sección.

El usuario también puede crear un protocolo propio e introducirlo en un programa DPL mediante el uso de comandos de puerto de nivel inferior, como GETCHAR y PUTCHAR (consulte el Capítulo 7, Referencia).

El protocolo secundario ANSI se activa cuando el parámetro de configuración del modo RS485 se configura en 1 (4 hilos) (ajuste predeterminado) o 5 (2 hilos). Para obtener información detallada sobre otros modos de comunicación, consulte los modos de comunicaciones serie más adelante en este capítulo.

### Conceptos básicos de la transmisión de datos

Los datos se transmiten en forma de carácter a una velocidad fija o velocidad en baudios. Un carácter está formado por 7 u 8 bits, normalmente.

Para que el receptor identifique los datos como válidos, cada carácter se enmarca en una trama. Esta trama contiene un bit de inicio, un bit de parada y un bit de paridad opcional. Sin la trama, el receptor no puede sincronizarse con los datos transmitidos.

A continuación se muestra una trama:

| Byte de carácter ASCII de bajo nivel |                     |                             |  |  |                              |  |     |                |               |
|--------------------------------------|---------------------|-----------------------------|--|--|------------------------------|--|-----|----------------|---------------|
|                                      |                     | Primer carácter hexadecimal |  |  | Segundo carácter hexadecimal |  |     |                |               |
| Bit de inicio                        | Siete bits de datos |                             |  |  |                              |  |     | Bit de paridad | Bit de parada |
| 0                                    | LSB                 |                             |  |  |                              |  | MSB | 1              |               |

Esta trama se conoce como trama de 10 bits, ya que son 10 los bits que se transmiten en total. El formato es el siguiente:

**1 bit de inicio, 7 bits de datos, par/impar/sin paridad, 1 bit de parada.**

lsb se refiere al bit menos significativo (el bit 0)

msb se refiere al bit más significativo (bit 6)

El receptor utiliza el bit de paridad para comprobar la integridad de los datos que ha recibido

El conjunto de caracteres que se utiliza recibe el nombre de conjunto ASCII de bajo nivel. Este conjunto está formado por 128 caracteres con numeración decimal del 0 al 127. Los primeros 32 caracteres del conjunto ASCII (hex. 00 a 1F) sirven para representar códigos especiales; es decir, códigos de control que tienen un significado especial (por ejemplo, el inicio de texto se denomina STX y su código ASCII es 02.)

En un PC o terminal, el carácter **STX** puede transmitirse presionando **Ctrl+B**. Si el UD70 se encuentra en el modo ANSI estándar, interpreta que el carácter STX va seguido de un comando.

Mediante el código de control **EOT** (fin de transmisión) se envía un aviso de recepción de un mensaje nuevo a todas las tarjetas UD70 y accionamientos del bus RS485. Este código suele enviarse al inicio de un mensaje para que todos los dispositivos se ajusten en **Ready to receive message** (listo para recibir el mensaje).

### Caracteres de control

Los comandos y peticiones se envían en paquetes de mensajes. Cada mensaje se inicia con un carácter de control especial, y puede contener más caracteres de este tipo. A continuación se proporciona una lista de todos los caracteres de control que pueden utilizarse al enviar y recibir un mensaje:

| Carácter | Significado  | Código ASCII (decimal) | Teclas... |
|----------|--|------------------------|-----------|
| EOT      | Reinicio<br>Avisa al UD70 para que se prepare para un nuevo mensaje.<br>También indica que un parámetro no existe. | 04                     | Ctrl D    |
| ENQ      | Consulta<br>Se utiliza para interrogar al accionamiento.   | 05                     | Ctrl E    |
| STX      | Inicio de texto<br>Se utiliza para iniciar un comando.   | 02                     | Ctrl B    |
| ETX      | Fin de texto<br>Se utiliza al final de un comando.   | 03                     | Ctrl C    |
| ACK      | Confirmación de recepción (mensaje aceptado)   | 06                     | Ctrl F    |
| NAK      | Confirmación negativa (mensaje no descifrado)  | 21                     | Ctrl U    |
| BS       | Retroceso (ir al parámetro anterior)   | 08                     | Ctrl H    |

### Velocidad en baudios

El usuario puede seleccionar una velocidad en baudios en el rango 300 - 38400. El valor predeterminado es 4800. Consulte los parámetros de configuración.

## Dirección

---

Es preciso asignar una identificación o dirección única a cada unidad de un bus de comunicaciones ANSI para que sólo la unidad UD70 de destino responda. La dirección está compuesta de dos partes:

- La dirección de grupo, que es el primer dígito.
- La dirección de la unidad, que es el segundo dígito.

Ambas direcciones tienen un rango de 1 a 9 y no se permite una dirección de grupo o de unidad con valor 0 (las direcciones 01, 10, 20, etc. no son válidas). Esto se debe a la posibilidad de agrupar tarjetas UD70 y algunos accionamientos (máximo de 9 unidades por grupo), y a que un mensaje puede enviarse a través del bus de comunicaciones ANSI a todas las unidades del grupo. Para remitir un mensaje a un grupo en particular, se utiliza la dirección cero (0) como dirección de la unidad. Por ejemplo, para dirigirse a todas las unidades del grupo 6, la dirección completa que se utilizará será 60.

Además, se incluye una función adicional que permite enviar un mensaje simultáneamente a todas las unidades de todos los grupos utilizando la dirección 00. Esta dirección se puede utilizar para enviar un comando de inicio (Start) a un grupo de accionamientos que se han acoplado mecánicamente para accionar una línea transportadora. Así, todos los accionamientos comenzarán a funcionar a la vez.

### **Nota**

**Es importante tener en cuenta que las tarjetas UD70 no reconocen el comando cuando se utiliza la dirección de grupo. (Si varias tarjetas intentan responder al mismo tiempo, el bus de comunicaciones serie puede presentar datos no significativos.)**

---

A fin de proteger la integridad de los datos, el formato de la dirección transmitida requiere la repetición de cada uno de los dos dígitos de la dirección; por ejemplo, la dirección del UD70 número 23 se envía en forma de cuatro caracteres:

2 2 3 3

La dirección serie se envía inmediatamente después del primer carácter de control del mensaje (EOT).

## Identificación de parámetros

---

Todos los parámetros están identificados por cuatro dígitos que representan el número del menú y del parámetro, sin la coma decimal.

- Ejemplo**
- Para enviar un mensaje al menú 4, parámetro 8, escriba 0408 (es preciso incluir el cero inicial)
  - Para enviar un mensaje al menú 18, parámetro 11, escriba 1811.

## Campo de datos

Los datos que se van a enviar o solicitar ocupan los caracteres situados inmediatamente detrás del número de parámetro. En la estructura de un mensaje, la longitud mínima del campo de datos es de dos caracteres.

Los datos se expresan normalmente en forma de valor numérico decimal. También es posible utilizar el formato hexadecimal, para lo que es preciso especificar el primer carácter del campo de datos como una X.

El primer carácter del campo de datos (D1) sólo puede ser uno de los siguientes:

Espacio (32 dec.)

+

-

X (para hex.) - hex. se utiliza normalmente para acceder a los datos de la caja de E/S.

## Suma de comprobación de bloque (BCC)

Para evitar el deterioro de los mensajes que envía o recibe el UD70 durante la transmisión, todos los mensajes de escritura y respuestas de datos terminan con el carácter de suma de comprobación de bloque (BCC). Consulte Cálculo de la suma de comprobación de bloque (BCC) más adelante en esta sección.

## Lectura de parámetros

Para leer un parámetro, se envía el siguiente mensaje:

| Control    | Dirección |           |           |           | Parámetro |           |           |           | Control    |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| <b>EOT</b> | <b>GA</b> | <b>GA</b> | <b>UA</b> | <b>UA</b> | <b>M1</b> | <b>M2</b> | <b>P1</b> | <b>P2</b> | <b>ENQ</b> |

donde:

GA = Dirección de grupo

UA = Dirección de unidad

M1 M2 = Número de menú

P1 P2 = Número de parámetro

### **Nota**

**En este mensaje no se envía ningún carácter BCC.**

Si el UD70 descifra el mensaje, utiliza la siguiente estructura en la respuesta:

| Control    | Parámetro |           |           |           | Datos     |     |           |            | Control    | BCC |
|------------|-----------|-----------|-----------|-----------|-----------|-----|-----------|------------|------------|-----|
| <b>STX</b> | <b>M1</b> | <b>M2</b> | <b>P1</b> | <b>P2</b> | <b>D1</b> | ... | <b>Dn</b> | <b>ETX</b> | <b>BCC</b> |     |

donde:

- M1 M2 = Número de menú
- P1 P2 = Número de parámetro
- D1..Dn = Datos
  - Primer carácter:
    - + o espacio para valores positivos
    - para valores negativos
    - X para valores hex.
- BCC = Suma de comprobación de bloque

Si uno de los parámetros solicitados no existe, el UD70 enviará un carácter EOT (ASCII 04).

**Ejemplo** Para visualizar el punto de referencia de velocidad de un Unidrive que es la unidad 2 del grupo 1, se envía lo siguiente:

| Control    | Dirección |          |          |          | Parámetro |          |          |          | Control    |
|------------|-----------|----------|----------|----------|-----------|----------|----------|----------|------------|
| <b>STX</b> | <b>1</b>  | <b>1</b> | <b>2</b> | <b>2</b> | <b>0</b>  | <b>1</b> | <b>2</b> | <b>1</b> | <b>ENQ</b> |

La unidad contesta lo siguiente:

| Control    | Parámetro |          |          |          | Datos |          |          |   | Control  | BCC        |          |
|------------|-----------|----------|----------|----------|-------|----------|----------|---|----------|------------|----------|
| <b>STX</b> | <b>0</b>  | <b>1</b> | <b>2</b> | <b>1</b> | -     | <b>4</b> | <b>7</b> | . | <b>6</b> | <b>ETX</b> | <b>7</b> |

**Nota**

**Cuando el UD70 responde a un comando, la longitud del campo de datos de la respuesta puede variar en función del tipo de accionamiento y del parámetro que se visualiza.**

## Nueva lectura de datos

Una vez que un mensaje de lectura se recibe y se descifra (es decir, los datos devueltos son válidos), puede enviarse un solo carácter de código de control para interrogar de nuevo al parámetro, al parámetro siguiente o al anterior. Los códigos de control son los siguientes:

| Código de control | Función                               | Teclas... |
|-------------------|---------------------------------------|-----------|
| NAK               | Devolver el valor del mismo parámetro | Ctrl U    |
| ACK               | Leer el parámetro siguiente           | Ctrl F    |
| BS                | Leer el parámetro anterior            | Ctrl H    |

Esta función permite ahorrar tiempo si un parámetro se controla durante un periodo de tiempo.

## Escritura en parámetros

Para escribir datos en un parámetro (accionamiento o virtual), debe utilizarse la siguiente estructura de mensaje:

| Control    | Dirección |           |           | Control   | Parámetro  |           |           |           | Datos     |           |            | Control   | BCC        |  |
|------------|-----------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|------------|-----------|------------|--|
| <b>EOT</b> | <b>GA</b> | <b>GA</b> | <b>UA</b> | <b>UA</b> | <b>STX</b> | <b>M1</b> | <b>M2</b> | <b>P1</b> | <b>P2</b> | <b>D1</b> | <b>...</b> | <b>Dn</b> | <b>ETX</b> |  |

donde:

- GA = Dirección de grupo
- GU = Dirección de unidad
- M1 M2 = Número de menú
- P1 P2 = Número de parámetro
- D1...DN = Datos

Primer carácter:

- + o espacio para valores positivos
- para valores negativos
- X para valores hex.

- BCC = Suma de comprobación de bloque

El campo de datos puede tener una longitud variable y su longitud máxima depende del parámetro que se está editando.

El UD70 utilizará un solo carácter de control en la respuesta, como se indica:

| Código de control | Significado  |
|-------------------|--|
| ACK               | Confirmación de recepción: el mensaje se ha descifrado y aplicado.   |
| NAK               | Mensaje no válido: tamaño de datos demasiado grande o fuera de rango, parámetro no válido, parámetro de sólo lectura o BCC incorrecto. |

**Ejemplo** Al ajustar el parámetro **Frecuencia prefijada 1** en +76.4 para el Unidrive (unidad 6, grupo 2), se envía:

| Control    | Dirección |          |          |          | Control    | Parámetro |          |          |          | Datos    |          |          |          | Control  | BCC      |            |          |
|------------|-----------|----------|----------|----------|------------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|------------|----------|
| <b>EOT</b> | <b>2</b>  | <b>2</b> | <b>6</b> | <b>6</b> | <b>STX</b> | <b>0</b>  | <b>1</b> | <b>2</b> | <b>1</b> | <b>+</b> | <b>0</b> | <b>7</b> | <b>6</b> | <b>.</b> | <b>4</b> | <b>ETX</b> | <b>1</b> |

### Reescritura de datos

Después de que se haya enviado un mensaje de escritura que incluya el campo de dirección y de que la recepción se haya confirmado mediante <ACK> o <NAK>, los siguientes mensajes de escritura que se envíen a ese UD70 podrán utilizar una estructura de mensaje de reescritura en la que no será necesario volver a transmitir la dirección. La estructura de reescritura es la siguiente:

|            |           |           |           |           |           |            |           |            |            |
|------------|-----------|-----------|-----------|-----------|-----------|------------|-----------|------------|------------|
| <b>STX</b> | <b>M1</b> | <b>M2</b> | <b>P1</b> | <b>P2</b> | <b>D1</b> | <b>...</b> | <b>Dn</b> | <b>ETX</b> | <b>BCC</b> |
|------------|-----------|-----------|-----------|-----------|-----------|------------|-----------|------------|------------|

Cuando el mensaje se envíe a otro UD70 o se reciba un carácter no válido, la función de reescritura dejará de funcionar y sólo será posible enviar otro mensaje al primer UD70 utilizando el mensaje de escritura completo con la dirección.

### Cálculo de la suma de comprobación de bloque (BCC)

La suma de comprobación de bloque se calcula aplicando una función OR exclusiva a todos los caracteres de un mensaje que siguen al carácter de control STX.

#### Tabla de definición de XOR

| A | B | Resultado |
|---|---|-----------|
| 0 | 0 | 0         |
| 0 | 1 | 1         |
| 1 | 0 | 1         |
| 1 | 1 | 0         |

Por ejemplo, el comando serie que se utiliza para ajustar **Velocidad prefijada 1** en -34,5 Hz en un Unidrive:

El mensaje será el siguiente:

|                           |           |          |          |          |                        |           |          |          |          |          |          |          |          |          |            |          |
|---------------------------|-----------|----------|----------|----------|------------------------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|------------|----------|
| Control                   | Dirección |          |          |          | Control                | Parámetro |          |          |          | Datos    |          |          |          | Control  | BCC        |          |
| <b>EOT</b>                | <b>1</b>  | <b>1</b> | <b>2</b> | <b>2</b> | <b>STX</b>             | <b>0</b>  | <b>1</b> | <b>2</b> | <b>1</b> | <b>-</b> | <b>3</b> | <b>4</b> | <b>.</b> | <b>5</b> | <b>ETX</b> | <b>0</b> |
| No incluido en el cálculo |           |          |          |          | Incluido en el cálculo |           |          |          |          |          |          |          |          |          | Resultado  |          |

El primer carácter del cálculo de BCC es 0 (00110000 en binario), cuyo valor se toma como valor inicial o resultante. El carácter siguiente es 1 (00110001 en binario), al que ahora se aplica el operador OR exclusivo (XOR). Con el resultado anterior, se obtiene un nuevo valor binario de 00000001.

En la siguiente tabla se muestra el cálculo completo:

| Carácter | Valor binario | Resultado XOR |
|----------|---------------|---------------|
| 0        | 0011 0000     | -             |
| 1        | 0011 0001     | 0000 0001     |
| 2        | 0011 0010     | 0011 0011     |
| 1        | 0011 0001     | 0000 0010     |
| -        | 0010 1101     | 0010 1111     |
| 3        | 0011 0011     | 0001 1100     |
| 4        | 0011 0100     | 0010 1000     |
| .        | 0010 1110     | 0000 0110     |
| 5        | 0011 0101     | 0011 0011     |
| ETX      | 0000 0011     | 0011 0000     |

El valor final será el BCC, si su valor decimal equivalente es mayor que 31 (los caracteres ASCII entre 00 y 31 se utilizan como códigos de control).

Cuando el resultado final de XOR es un valor decimal menor que 32, se suma 32. En este ejemplo, 0011 0000 es el valor decimal 48. Al ser mayor que 31, se acepta como valor final de BCC. El valor decimal 48 es el carácter 0 y el mensaje completo es el siguiente:

|            |          |          |          |          |            |          |          |          |          |          |          |          |          |          |            |          |
|------------|----------|----------|----------|----------|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|------------|----------|
| <b>EOT</b> | <b>1</b> | <b>1</b> | <b>2</b> | <b>2</b> | <b>STX</b> | <b>0</b> | <b>1</b> | <b>2</b> | <b>1</b> | <b>-</b> | <b>3</b> | <b>4</b> | <b>.</b> | <b>5</b> | <b>ETX</b> | <b>0</b> |
|------------|----------|----------|----------|----------|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|------------|----------|

**Ejemplo** Programa QuickBasic para calcular el BCC

```

mess$ = CHR$(4)+"1122"+CHR$(2)+"0125"+"-34.5"+CHR$(3)
bcc%= 0
FOR n%= 7 a LEN(mess$) empieza en el carácter siguiente a 'chr$(2).
bcc%= bcc% XOR ASC(MID$(mess$, n%, 1))
NEXT
IF bcc% < 32 THEN bcc%= bcc% + 32
mess$ = mess$ + CHR$(bcc%)
PRINT mess$

```

En DPL, las funciones ANSIREAD y ANSIWRITE calculan automáticamente el BCC.

## 6.4 Modos de comunicaciones serie

El UD70 dispone de los once modos de comunicación siguientes:

- ANSI secundario estándar de 4 y 2 hilos
- Transferencia de parámetros a gran velocidad
- Modos principal/secundario programables
- Modos de usuario
- Modo de caja de E/S

Las opciones disponibles son las siguientes (para obtener información detallada sobre los parámetros actuales, consulte los parámetros de configuración en el Capítulo 10, Parámetros):

### **Modo 1 4 hilos estándar**

RS485 con protocolo ANSI secundario (predeterminado).

### **Modo 2 Modo principal**

En este modo se utilizan los datos del parámetro definido, se convierten a una escala de 16000 y se envían continuamente a un accionamiento secundario. La unidad secundaria puede ser un MD29, un Unidrive con módulo de opciones de comunicación estándar, un accionamiento Mentor II, un accionamiento CDE u otro UD70.

### **Modo 3 Modo secundario**

En este modo los datos se envían desde el puerto serie y se convierten a escala utilizando el valor del parámetro de configuración a escala antes de insertarlos en el parámetro de destino.

### **Modo 4 Modo de cascada**

En este modo se envía un parámetro definido a un accionamiento remoto (sin conversión a escala) y los datos recibidos se insertan en el parámetro de destino. Los datos no se convierten a escala.

La unidad transmisora o receptora puede ser un UD70, un MD29, un accionamiento Unidrive con módulo de opciones de comunicación estándar o un accionamiento Mentor II.

Este modo suele utilizarse en aplicaciones del tipo trefilado, en las que se asigna una referencia de velocidad de una fuente externa (como un potenciómetro) al primer accionamiento. El modo 4 permite pasar esta referencia al accionamiento siguiente de la cadena, que, a su vez, emplea una opción de aplicaciones para recibir los datos y aplicar el trefilado. A continuación, ese accionamiento pasa la nueva referencia al accionamiento siguiente, y así sucesivamente.

### **Modo 5 RS485 de 2 hilos**

Emplea el protocolo ANSI secundario. Recuerde que el RS232 no es un RS485 de 2 hilos.

#### **Modo 6 Modo de usuario**

Este modo desactiva todos los protocolos internos y permite que el usuario utilice el puerto RS485 directamente desde un programa DPL. Por lo general, este modo se utiliza con los comandos DPL del protocolo ANSI principal (ANSIREAD, ANSIWRITE, etc.). Los protocolos definidos por el usuario también pueden aplicarse en DPL con comandos PUTCHAR y GETCHAR de nivel inferior. La trama de datos de las comunicaciones se organiza de la siguiente manera:

1 bit de inicio, 7 bits de datos, paridad EVEN (par), 1 bit de parada, 10 bits en total

#### **Modo 7 Modo de usuario**

Este modo desactiva todos los protocolos internos y permite la lectura y escritura directamente desde el puerto ANSI mediante el uso de las funciones PUTCHAR y GETCHAR. La trama de datos de las comunicaciones se organiza de la manera siguiente:

1 bit de inicio, 8 bits de datos, paridad EVEN (par), 1 bit de parada, 11 bits en total

#### **Modo 8 Modo de usuario**

Este modo desactiva todos los protocolos internos y permite la lectura y escritura directamente desde el puerto ANSI RS485 mediante el uso de las funciones PUTCHAR y GETCHAR. La trama de datos de las comunicaciones se organiza de la siguiente manera:

1 bit de inicio, 8 bits de parada, paridad NO (sin paridad), 1 bit de parada, 10 bits en total

#### **Modo 9 Modo de usuario**

La trama de datos de las comunicaciones se organiza de la siguiente manera:

1 bit de inicio, 9 bits de datos, sin parida, 1 bit de parada, 11 bits en total

#### **Modo 10 Modo de caja de E/S**

Este modo permite conectar directamente una *sola* caja de E/S al UD70 a través del puerto EIA RS485.

En el caso de que se necesiten varias cajas de E/S, habrá que configurarlas en el modo ANSI estándar. Para transferir los datos es preciso seleccionar el modo 6 y utilizar los comandos DPL RS485.

#### **Modo 11 Modo de usuario**

Este modo permite evitar la memoria intermedia del software y establecer una interconexión directa con el hardware. De esta forma se reduce el tiempo de transferencia de la información a través del puerto RS485 del UD70. La velocidad en baudios es programable, al igual que en los otros modos, y la trama de datos de las comunicaciones se organiza de la siguiente manera:

1 bit de inicio, 9 bits de datos, sin parida, 1 bit de parada, 11 bits en total

**Modo 12 Reservado**

**Modo 13 Modbus - RTU (sólo modo secundario)**

**Modo 14 Modbus - ASCII (sólo modo secundario)**

Los modos secundarios Modbus RTU y ASCII permiten utilizar las funciones Read Multiple Registers, Preset Single Registers y Preset Multiple Registers.

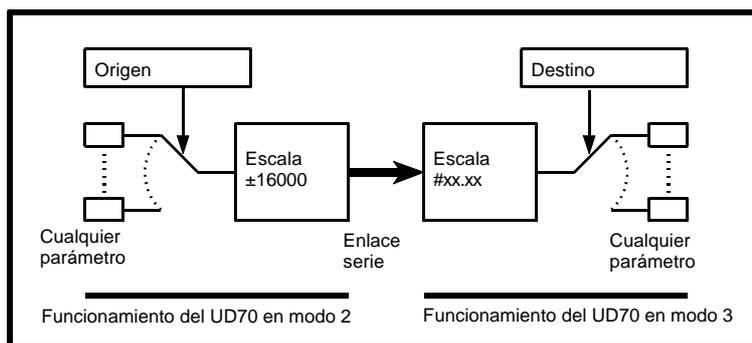
Este modo limita a 20 el número de registros consecutivos, y el rango de dirección de nodo entre 11 y 99.

Para obtener información sobre el protocolo Modbus, póngase en contacto con el Centro de accionamientos local.

**Nota**

**En los modos 2, 3 y 4, la comprobación de la integridad de los datos es mínima.**

**Utilización de los modos 2 y 3**



*Funciones de los modos 2 y 3*

**Utilización de los modos 6, 7, 8 y 9**

En los modos 6, 7 y 8 es posible utilizar el UD70 como un dispositivo ANSI principal que controle otros módulos UD70, tarjetas MD29, accionamientos y demás dispositivos ANSI compatibles. De esta forma se evita tener que utilizar un PLC o un PC personalizado para controlar un sistema o proceso. Si se utiliza una caja de E/S, es posible que el sistema no necesite un PLC.

Los modos 6 a 9 también permiten crear protocolos serie personalizados que servirán para comunicarse con dispositivos ANSI no compatibles, como módulos de visualización inteligentes, etc.

El puerto ANSI RS485 se configura en la memoria intermedia de la siguiente manera:

- Recepción: 60 bytes
- Transmisión: 25 bytes

## **Utilización del modo 11**

---

Cuando se utilizan otros modos, el inicio de la transmisión y recepción de datos puede demorarse 8 ms como máximo. En el modo 11, los datos se desvían a la memoria FIFO del software, que permite reducir las demoras del sistema. El parámetro #17.07 sirve para programar la velocidad en baudios.

## **Utilización de los modos 13 y 14**

---

El protocolo Modbus sólo puede ser secundario, ya que el UD70 no puede utilizarse como protocolo Modbus principal en los modos 13 y 14. La velocidad en baudios se selecciona mediante el parámetro #17.07.

## **6.5 Instrucciones ANSI**

Las instrucciones ANSI RS485 que se encuentran disponibles son las siguientes:

|           |   |
|-----------|---|
| ANSIREAD  | Envío de un comando Read (lectura) a una unidad remota    |
| ANSIREPLY | Lectura de una respuesta enviada por una unidad remota    |
| ANSIWRITE | Envío de un comando Write (escritura) a una unidad remota |
| GETCHAR   | Recepción directa de un solo carácter                     |
| PUTCHAR   | Transmisión directa de un solo carácter                   |

Para obtener información sobre estos comandos, consulte el Capítulo 7, Referencia.

## 6.6 Ejemplo de instrucciones ANSI

En el fragmento de programa siguiente se ofrece un ejemplo de lectura de una referencia de velocidad desde un Unidrive, y de escritura de ésta referencia en un accionamiento CD.

Ambos accionamientos están conectados al bus RS485. La dirección del Unidrive es 11 y la del accionamiento CD es 12.

Para que este programa funcione, el modo RS485 debe configurarse en 6.

```
BACKGROUND{
top:
// enviar comando de lectura para leer parámetro #1.21
resultado% = ANSIREAD(11, "0121")
IF resultado% = 0 THEN
    PRINT "No se pudo enviar el mensaje"
    GOTO top:
ENDIF
CALL obtener_respuesta: // obtener respuesta de comando de lectura
IF respuesta% <= -65536 THEN // ¿respuesta válida?
    PRINT "Error de lectura de datos desde Unidrive"
    GOTO top:
ENDIF
// enviar comando de escritura a CD utilizando 1 espacio decimal
resultado% = ANSIWRITE(12, "SP", respuesta%, 1)
IF resultado% = 0 THEN
    PRINT "No se pudo enviar el mensaje"
    GOTO top:
ENDIF
CALL obtener_respuesta: // ¿admitir escritura?
IF respuesta% >= -65540 THEN // -65540 es ACK
    PRINT "Error al enviar el mensaje a CD"
ENDIF
GOTO top:
}
obtener_respuesta: { // subrutina para espera de respuesta
    expirado% = 0 // reiniciar temporizador
    DO
        // -65536 si no se ha recibido respuesta
        respuesta% = ANSIREPLY
        LOOP WHILE respuesta% = -65536 AND expirado% < 50
        // bucle sólo si se ha recibido respuesta o
        // ha expirado el tiempo.
        // si la tarea reloj se ajusta en 5 ms, el tiempo expirará en
        250ms
    }
}
CLOCK {
expirado% = expirado% + 1 // incrementar variable de tiempo de
expiración
}
}
```

**Nota**

**Consulte otros ejemplos de comandos ANSI en el archivo de ayuda del juego de herramientas DPL.**

---



---

## 7 Referencia

---

### 7.1 Tareas

Consulte también la sección Tareas y programación en tiempo real en el Capítulo 4, Programación DPL.

#### Tarea INITIAL

---

Normalmente, la tarea INITIAL sirve para inicializar las variables del programa y los parámetros del accionamiento.

La tarea INITIAL permite arrancar el programa DPL y sólo se ejecuta cuando el UD70 se pone a cero o al conectar la alimentación de CA al accionamiento.

La tarea INITIAL se considera especial porque es imposible ejecutar otras tareas mientras se está utilizando. Esto es importante en el caso de que otras tareas en tiempo real manipulen datos con valores iniciales.

#### Tarea BACKGROUND

---

La tarea BACKGROUND se utiliza con funciones y comandos que no precisan de control basado en el tiempo ni en el codificador, y resulta útil en los casos siguientes:

- Registro de datos
- Comprobación de entradas digitales
- Configuración de estado de salida

La tarea BACKGROUND se ejecuta una vez terminada la tarea Initial. Se recomienda ejecutar la mayor parte del programa en la tarea BACKGROUND.

#### **Nota**

**La tarea BACKGROUND no se repite automáticamente.**

---

#### *Ejemplo*

```
BACKGROUND{
  RAMP:
  #1.21 = 0
  DO WHILE #1.21 < 1000
    #1.21 = #1.21+1
  LOOP
  GOTO RAMP:
}
```

## Tarea CLOCK

---

La tarea CLOCK permite ejercer un control de tiempo crítico sobre el accionamiento y los comandos enviados al accionamiento (por ejemplo, rampa de aceleración y deceleración controlada).

La tarea CLOCK se ejecuta en una base de tiempos fija y no mantiene la sincronía con el accionamiento. La base de tiempos que se utiliza depende del parámetro de configuración (consulte los parámetros de configuración en el Capítulo 10, Parámetros), que puede ajustarse en un rango de entre 5 y 200ms.

**Ejemplo** En este ejemplo se genera una onda senoidal.

```
CLOCK{  
  #1.21 = SIN (rad)*1000  
  rad = rad+0.01  
  IF rad>6.283185 THEN  
    rad = 0  
  ENDIF  
}
```

## Tarea SPEED

---

En el Unidrive, la tarea SPEED está sincronizada con el bucle de control de velocidad, por lo que la periodicidad con que se ejecuta esta tarea depende de la frecuencia de conmutación PWM.

| Accionamiento | Frecuencia de conmutación | Base de tiempos de la tarea SPEED |
|---------------|---------------------------|-----------------------------------|
|               | kHz                       | ms                                |
| Unidrive      | 3, 6 o 12                 | 1,38                              |
| Unidrive      | 4,5 o 9                   | 1,84                              |

## Tarea ENCODER

---

La función típica de la tarea ENCODER consiste en controlar la actividad de un codificador.

En el Unidrive, la tarea ENCODER está sincronizada con el bucle de control de velocidad y se ejecuta cada cuatro ciclos de la tarea SPEED. La periodicidad con que se ejecuta esta tarea depende de la frecuencia de conmutación PWM.

| <b>Frecuencia de conmutación PWM</b> | <b>Base de tiempos de la tarea ENCODER</b> |
|--------------------------------------|--|
| <b>kHz</b>                           | <b>ms</b>                                  |
| 3, 6 o 12                            | 5,52                                       |
| 4,5 o 9                              | 7,36                                       |

*Ejemplo*

```
ENCODER{
    principal% = #90.2
    secundario% = #90.4
    EPOS = EPOS + principal% - secundario%
}
```

## Tarea EVENT

---

La tarea EVENT es una tarea especial que se ejecuta cuando ocurre un evento específico. La unidad de temporización/contador determina el origen del evento.

Para obtener más información, consulte la sección Unidad de temporización/contador en el Capítulo 8, Funciones.

## Tarea ERROR

---

La tarea ERROR se ejecuta cuando ocurre un error en tiempo de ejecución en un programa DPL.

Para obtener más información, consulte la sección Gestión avanzada de errores en el Capítulo 9, Diagnósticos.

## Tarea definida por el usuario

---

Este tipo de tareas son subrutinas que escribe el usuario y que se utilizan junto con la instrucción CALL (consulte la instrucción *CALL* más adelante en esta sección).

No existen restricciones en cuanto a la asignación de nombre a las tareas definidas por el usuario, ni a su inserción en cualquier parte del programa.

**Ejemplo** El nombre de esta tarea es **RAMP**:

```
BACKGROUND{
  Loop:
  CALL RAMP:
  GOTO Loop:
}
RAMP: {
  #1.21=0
  DO WHILE #1.21<1500
    #1.12=#1.21+1
  LOOP
}
```

**Distinción entre mayúscula y minúscula** En el nombre de las subrutinas se distingue entre mayúscula y minúscula. Si la instrucción CALL del ejemplo anterior se escribe como **CALL ramp**, el programa no se compilará debido a que la palabra "ramp" debería introducirse en mayúscula. El error de compilación que aparece será el siguiente:

Undefined reference to \_0103ramp. (Referencia a rampa sin definir)

## Tarea NOTES

---

Seudotarea que ignora el compilador. El creador del programa puede utilizar la tarea NOTES para proporcionar información al usuario.

**Ejemplo**

```
NOTES{
  Puede incluir su documentación aquí.
}
```

## Sección CONST

---

La sección CONST sirve para definir datos de matrices constantes y no es una tarea. Consulte la sección Variables en el Capítulo 4, Programación DPL.

## 7.2 Instrucciones y funciones

Las instrucciones y funciones siguientes pueden contener varias formas de sintaxis. En estos casos, la primera forma de sintaxis se aplicará a variables enteras y la segunda a variables de coma flotante. (Consulte el Capítulo 4, Programación DPL).

### ABS

---

**Sintaxis**      resultado% = ABS (expresión%)  
                 resultado = ABS (expresión)

Esta función matemática permite obtener el valor absoluto de una expresión sin considerar el signo positivo o negativo de la expresión (los números negativos se convierten en positivos).

**Ejemplo**      ABS(45.5 - 100) ;    el resultado es 54.5

### ANSIREAD

---

**Sintaxis**      resultado% = ANSIREAD (dirección accionamiento%, "nemónico")

Función del puerto RS485 de 4 hilos con la que siempre se obtiene un valor entero.

Esta función transmite una petición de lectura de parámetro a una unidad o un accionamiento remoto a través del puerto ANSI RS485.

La dirección del accionamiento es una expresión entera, cuyo valor suele estar comprendido entre 01 y 99.

El nemónico es una cadena que contiene el número de parámetro. El formato y la longitud de la cadena dependen del accionamiento remoto.

Si la petición de lectura se ha enviado correctamente, se obtiene el valor 1. Si el mensaje no ha podido enviarse, se obtiene el valor 0 (por ejemplo, si la transmisión ya está realizándose).

#### **Nota**

**La instrucción ANSIREAD no espera una respuesta del accionamiento remoto.**

---

**Ejemplo**      resultado% = ANSIREAD(12,"0122") \\ leer #1.22 desde accionamiento 12

Consulte también Ejemplo de comandos ANSI en el Capítulo 6, Comunicaciones serie.

## ANSIREPLY

---

**Sintaxis** resultado% = ANSIREPLY

Función del puerto RS485 de 4 hilos que se utiliza junto con las funciones ANSIREAD y ANSIWRITE. Esta función suele utilizarse para obtener los datos que devuelve un accionamiento inmediatamente después de emitir los comandos ANSIREAD o ANSIWRITE.

Se obtiene la siguiente información:

- 65536 Todavía no se ha recibido ninguna respuesta
- 65537 Se ha recibido respuesta, pero contiene una suma de comprobación de bloque errónea
- 65538 EOT recibido (por ejemplo, el parámetro no existe)
- 65539 NAK recibido
- 65540 ACK recibido

Cualquier otro valor corresponderá al nemónico introducido en ANSIREAD.

Consulte también Ejemplo de comandos ANSI en el Capítulo 6, Comunicaciones serie.

## ANSIWRITE

---

**Sintaxis** resultado% = ANSIWRITE(*dirección accionamiento%*, "*nemónico*", *valor%*, *atributo%*)

Función del puerto RS485 de cuatro hilos que sirve para transmitir una petición de escritura de parámetro a un accionamiento remoto a través del puerto ANSI RS485.

La dirección del accionamiento es una expresión entera, cuyo valor suele estar comprendido entre 01 y 99. El nemónico es una cadena que contiene el número de parámetro. El formato y la longitud de la cadena dependen del accionamiento remoto. El valor debe ser una expresión numérica entera.

A continuación se proporcionan los argumentos de atributo que se han especificado:

- |     |  |
|-----|--|
| 0   | Sin decimales  |
| 1   | Un decimal   |
| 2   | Dos decimales  |
| 3   | Tres decimales   |
| 128 | Escritura hex en CT para la caja de E/S<br>(seis caracteres prefijados mediante X) |
| 129 | Escritura hex en accionamiento CD<br>(cuatro caracteres prefijados mediante >)     |
| 130 | Escritura hex en unidad de CD<br>(dos caracteres prefijados mediante >)            |

Si la petición de escritura se ha enviado correctamente, se obtiene el valor 1.

Si el mensaje no ha podido enviarse, se obtiene el valor 0 (por ejemplo, la transmisión ya está realizándose).

### Nota

**Esta instrucción no espera una respuesta de un accionamiento remoto.**

**Ejemplo** resultado% = ANSIWRITE(13,"0211", 150, 1) \\ ajustar #2.11 en 15.0 en accionamiento remoto 13  
Consulte también Ejemplo de comandos ANSI en el Capítulo 6, Comunicaciones serie.

## ARCTAN

---

**Sintaxis** resultado = ARCTAN (expresión)

Esta función matemática permite obtener el arco tangente de la expresión en radianes.

**Ejemplo** x = ARCTAN(0.8)  
Se obtiene el valor 0,674740942 radianes.  
Consulte también la instrucción *TAN*.

## AVERAGE

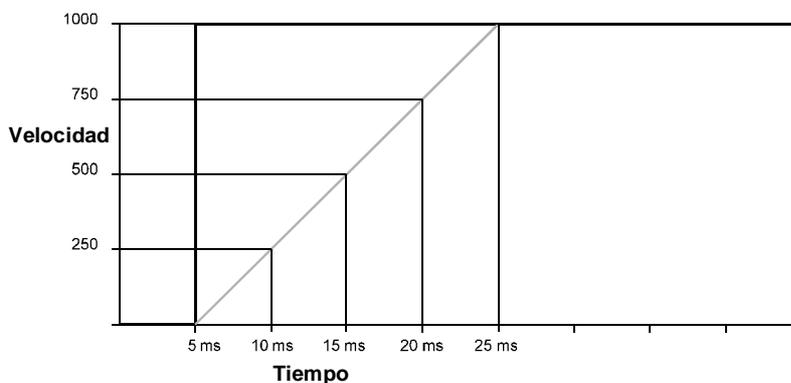
---

**Sintaxis** resultado% = AVERAGE (expresión entrada%, número de muestras%)

Esta función de proceso de señal permite obtener la media dinámica del valor introducido para el número deseado de muestras. El número de muestras debe ser un valor entero constante y la expresión introducida, un valor entero.

Esta función sólo puede utilizarse en las tareas SPEED, CLOCK o ENCODER. AVERAGE debe ejecutarse en una base de tiempos regular.

Si se aplica un cambio gradual de 0 a 1000 al parámetro #1.21, el resultado de la media será el siguiente (con una base de tiempos de reloj de 5 ms).



**Ejemplo** CLOCK{  
  entrada% = #1.21  
  #1.22 = AVERAGE (entrada%, 4)  
}

Consulte también la instrucción *FILTER*.

## BCD2BIN

**Sintaxis** ent% = BCD2BIN (expresión entera%)

Esta función convierte un número decimal con codificación binaria en un entero binario normal y permite operar con datos digitales enviados por la caja de E/S (como selectores rotativos). (El decimal con codificación binaria es un método de escritura de números decimales en formato binario.)

954 en formato decimal se representa como 1001 0101 0100 en BCD.

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 |   |   |   | 5 |   |   |   | 4 |   |   |   |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

**Ejemplo** r% = BCD2BIN(#82.46)

## BIN2BCD

**Sintaxis** ent% = BIN2BCD (expresión entera%)

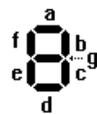
Esta función convierte un entero binario normal en un número decimal con codificación binaria y permite operar con datos digitales enviados a la caja de E/S.

Consulte el proceso de conversión de decimales en decimales con codificación binaria en la instrucción *BCD2BIN*.

## BCD2SEG

**Sintaxis** ent% = BCD2SEG (expresión entera%)

Esta función convierte los decimales con codificación binaria en códigos LED de 7 segmentos y permite operar con datos digitales enviados a la caja de E/S. Los datos enviados consisten en señales de impulso para un máximo de tres dígitos visualizados en 7 segmentos, como sigue:



|               |   |   |   |   |   |   |   |   |
|---------------|---|---|---|---|---|---|---|---|
| Número de bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Segmento      | - | g | f | e | d | c | b | a |

El byte menos significativo representa el primero de los tres dígitos. El valor máximo que puede convertirse es 999. Los ceros iniciales también se envían.

**Ejemplo** a% = BIN2BCD(123)  
seg\_datos% = BCD2SEG(a%)  
#83.46 = seg\_datos% ;salida 5200646 (0x4F5B06 = 321)

## CALL

---

**Sintaxis**      CALL programa-etiqueta:

Instrucción de control de flujo que sirve para ejecutar un programa de aplicación estándar o una tarea definida por el usuario. Una vez terminada la subrutina, el programa regresa a la línea que sigue a la instrucción [CALL].

**Ejemplo 1**      ENCODER{  
                  CALL diglk\_codificador: // Tarea de aplicación estándar  
                  //Bloqueo digital  
                  ...Resto de tarea ENCODER  
                  }

**Ejemplo 2**      INITIAL{  
                  CALL SETUP:  
                  }  
                  SETUP:{ //Éste es el nombre de la subrutina  
                  #1.21 = 0  
                  }

### **Nota**

**No es posible aplicar el proceso paso a paso en una tarea definida por el usuario.**

**En el nombre de la subrutina se distingue entre mayúscula y minúscula.**

---

## COS

---

**Sintaxis**      resultado = COS(expresión)

Esta función matemática permite obtener el coseno de un ángulo expresado en radianes. Con esta función siempre se obtiene una variable de coma flotante.

Consulte las instrucciones *SIN* y *TAN*.

**Ejemplo**      valor = COS (3.1416)

## CRC16

---

**Sintaxis**      CRC% = CRC16 (carácter%, CRC%)

Esta función sirve para calcular un valor CRC (prueba de redundancia cíclica, normalmente utilizada en los protocolos de comunicaciones serie) de 16 bits.

## DELAY

---

**Sintaxis**      DELAY (expresión entera%)

La instrucción DELAY hace que el programa realice una pausa durante el tiempo especificado por la expresión numérica entera, y aplica incrementos de 0,1 segundo. Sin embargo, esta instrucción no es muy fiable si se desea obtener un valor de retardo exacto. En el peor de los casos, el retardo puede durar 100 ms menos que el especificado (nunca tendrá una duración superior).

DELAY sólo puede utilizarse en las tareas INITIAL o BACKGROUND.

**Ejemplo**      #18.12 = 10 //ajustar parámetro 18.12 en 10  
                 DELAY(10) //retardar programa 1 segundo.  
                 #18.12 = 0

## DIM

---

**Sintaxis**      DIM variable% [número\_de\_elementos%]  
                 DIM variable [número\_de\_elementos%]

La instrucción DIM se utiliza para especificar una matriz formada por una serie de variables del mismo tipo (enteros o coma flotante). Aunque la instrucción no genera ningún código, indica al compilador que debe reservar espacio para una matriz dinámica.

Antes de utilizar una matriz, es preciso especificarla. Aunque se recomienda especificar la matriz en la tarea INITIAL, también puede definirse en la tarea en la que se va a utilizar.

**Ejemplo**      INITIAL{  
                 DIM mimatriz%[100] //identifica un entero denominado  
                 // [mimatriz] que incluye 100 elementos.  
                 mimatriz%[0] = 10 //ajustar el primer elemento en 10 al inicializar  
                 mimatriz%[1] = 20 //ajustar el segundo elemento en 20 al inicializar  
                 mimatriz%[99] = 50 //inicializar el último elemento  
                 índice% = 1  
                 }  
                 CLOCK{  
                 mimatriz%[índice%] = #3.02 //obtener realimentación de velocidad  
                 índice% = índice + 1 //incrementar contador de índice  
                 IF índice% = 100 THEN  
                      índice% = 0  
                 ENDIF  
                 }  
                 }

Consulte también Matrices en la sección Variables del Capítulo 4, Programación DPL.

## DO WHILE

---

**Sintaxis 1** DO WHILE Expresión condicional  
Instrucción  
LOOP

**Sintaxis 2** DO  
Instrucción  
LOOP WHILE Expresión condicional

**Sintaxis 3** DO WHILE Expresión condicional LOOP

Instrucción en bucle o iterativa que hace que un bloque de instrucciones se repita hasta que una expresión determinada se convierta en falsa.

La **sintaxis 1** permite evaluar la expresión condicional en primer lugar. Si el resultado es verdadero, las instrucciones se ejecutan y el programa continúa funcionando en bucle hasta que la expresión condicional se convierte en falsa.

La **sintaxis 2** permite ejecutar las instrucciones en primer lugar y evaluar la expresión condicional. Esto permite asegurarse de que las instrucciones del bucle se van a ejecutar al menos una vez.

La **sintaxis 3** es idéntica a la sintaxis 1, con la diferencia de que el bucle no incluye instrucciones de ejecución.

**Ejemplo 1** DO WHILE #1.21 < 1000  
#1.21 = #1.21 + 1  
LOOP

**Ejemplo 2** DO  
a = a + 0.001  
LOOP WHILE a < 6

**Ejemplo 3** DO WHILE #3.02 > 10 LOOP

(Si el valor de #3.02 es mayor que 10, el programa continuará repitiéndose.)

## EXIT

---

**Sintaxis** EXIT  
Instrucción de control de flujo que proporciona un método rápido de terminación de la tarea actual.

**Ejemplo** CLOCK{  
IF #18.22 = 1 THEN EXIT  
...  
(Si el valor de #18.22 es 1, el resto de la tarea CLOCK se cancela.)

## EXP

---

**Sintaxis** resultado = EXP (expresión)  
Mediante esta función matemática se obtiene la función exponencial ( $e^{\text{expresión}}$ ).

**Ejemplo** X = EXP(4.5)  
Se obtiene el valor 90,0171313.  
Consulte también la instrucción LN.

## FILTER

---

**Sintaxis**      `result% = FILTER (input_expression%, time_constant %)`  
                  `result = FILTER (input_expression, time_constant%)`

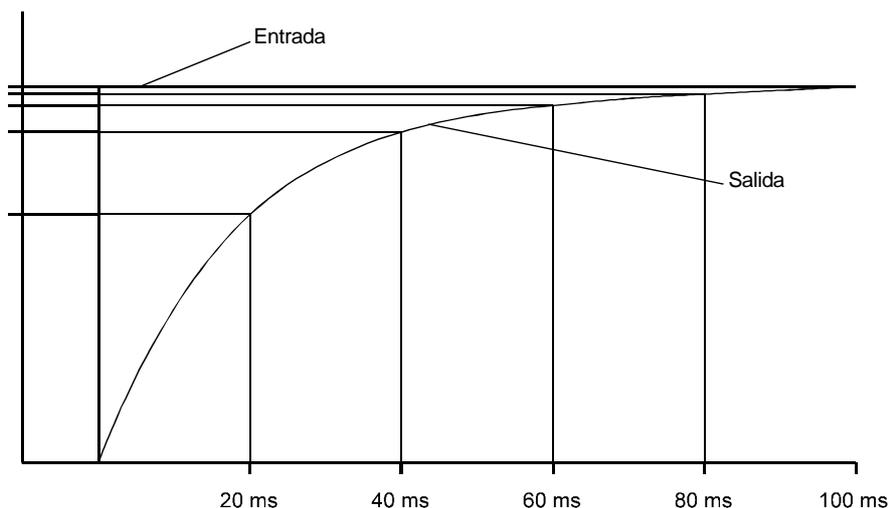
Función de proceso de señal que permite obtener el resultado de una función de filtro de primer orden utilizando la expresión introducida como entrada.

La constante de tiempo del filtro depende del valor especificado y de la frecuencia con la que se ejecuta la instrucción FILTER.

**Ejemplo**      `CLOCK{`  
                  `#1.21 = FILTER (#7.01, 4)`  
                  `}`

Con una base de tiempos de reloj de 5 ms, la constante de tiempo del filtro real será  $4 \times 5 \text{ ms} = 20 \text{ ms}$ .

Si se aplica un cambio gradual de 0 a 1000 al parámetro #1.21, la salida del filtro será la mostrada en el diagrama siguiente.



Después de aplicar cinco constantes de tiempo suele obtenerse el valor final en el resultado.

Esta función siempre debe utilizarse en la tarea CLOCK, ENCODER o SPEED.

## FLOAT

---

**Sintaxis**      resultado = FLOAT (expresión entera%)

Esta función convierte una variable entera en otra de coma flotante.

**Ejemplo**      error% = 123  
                  resultado = error% / 1000

Dado que  $123 / 1000 = 0$  en la aritmética de números enteros, esto genera un resultado de [resultado = 0]. Para obtener el resultado correcto, uno de los operandos debe convertirse en un valor de coma flotante.

error% = 123  
resultado = FLOAT(error%) / 1000

Esto permite hallar el valor correcto: **resultado = 0.123**

Consulte también la instrucción *INT*.

## GETCHAR

---

**Sintaxis**      resultado% = GETCHAR

Función del puerto RS485 que permite visualizar un carácter desde el puerto de comunicaciones RS485.

Si no se reciben caracteres en la memoria intermedia, el valor que se obtiene es -1; de lo contrario, el valor que se obtiene es el código ASCII del carácter leído.

Consulte también la instrucción *PUTCHAR*.

## GETKEY

---

**Sintaxis**      resultado% = GETKEY

Función del puerto RS232 que permite visualizar un carácter desde el puerto de comunicaciones RS232 PC. Esta función sólo puede utilizarse cuando las comunicaciones del juego de herramientas están desactivadas (Consulte Desactivación de las comunicaciones del juego de herramientas en el Capítulo 10, Parámetros).

Esta función sólo puede utilizarse si el modo de terminal elemental está activado (consulte Activación del modo de terminal elemental en el Capítulo 10, Parámetros).

El valor que se obtiene es el código ASCII del carácter leído. Si el carácter no existe, el valor resultante será -1.

El formato de los datos del puerto RS232 se ha establecido de la siguiente manera:

19200 baudios, 8 bits de datos, sin paridad, 1 bit de inicio, 1 bit de parada.

Consulte también la instrucción *PUTKEY*.

## GOTO

---

**Sintaxis** GOTO etiqueta:

Instrucción de bifurcación incondicional que hace que el programa pase a la línea indicada mediante [etiqueta] durante la ejecución, y que continúe ejecutándose a partir de esa línea. Al definir una etiqueta e indicar la etiqueta a la que debe dirigirse, es preciso incluir dos puntos [:] detrás de ésta. Los dos puntos [:] permiten identificar el nombre como una etiqueta. En otros lenguajes de programación se emplea un procedimiento distinto.

Si la tarea BACKGROUND se va a ejecutar continuamente, es preciso incluir una instrucción GOTO.

**Ejemplo**

```
BACKGROUND{
top: //éste es el nombre de la etiqueta
IF #18.32 = 1 THEN
    #1.21 = 100
ELSE
    #1.21 = 0
ENDIF
GOTO top: //ir a la línea con la etiqueta [top:]
}
```

## IF

---

**Sintaxis 1** IF expresión condicional THEN  
Instrucción  
ENDIF

**Sintaxis 2** IF expresión condicional THEN  
Instrucción si la condición es verdadera  
ELSE  
Instrucción si la condición es falsa  
ENDIF

**Sintaxis 3** IF expresión condicional 1 THEN  
Instrucciones si verdadero, ir a ENDIF  
ELSEIF expresión condicional 2 THEN  
Instrucciones si verdadero, ir a ENDIF  
ELSEIF expresión condicional 3 THEN...  
Instrucciones si verdadero, ir a ENDIF  
ELSE  
Instrucción  
ENDIF

**Sintaxis 4** IF *expresión condicional* THEN *Instrucción*

Las instrucciones condicionales IF, THEN, ELSE, ELSEIF y ENDIF realizan una operación hasta que se cumple la condición especificada.

**Ejemplo 1**     IF #1.21 = 25 THEN  
                   PRINT "25Hz"  
                   ENDIF

**Ejemplo 2**     IF a >= b AND (c%>- 1 OR z%>1) THEN  
                   a = b  
                   ELSE  
                   z = z+3  
                   ENDIF

**Ejemplo 3**     IF A% = 1 THEN  
                   #1.11 = 0  
                   ELSEIF a% = 2 OR a% = 3 THEN  
                   #1.11 = 1  
                   ELSEIF a% > 4 THEN  
                   #18.21 = 1  
                   ELSE  
                   #18.21 = 0  
                   ENDIF

En el ejemplo 3, el operador **OR** se emplea en la expresión condicional. Para combinar expresiones condicionales pueden utilizarse los operadores siguientes:

|     |           |
|-----|-----------|
| AND | Y lógico  |
| OR  | O lógico  |
| NOT | NO lógico |

En las expresiones condicionales pueden utilizarse los siguientes operadores condicionales:

|    |                   |
|----|-------------------|
| >  | Mayor que         |
| <  | Menor que         |
| >= | Mayor o igual que |
| <= | Menor o igual que |
| <> | Distinto de       |

**Nota**

**En el caso de las sintaxis 1 a 3, la instrucción IF siempre debe incluir ENDIF al final de la serie de instrucciones condicionales.**

---

## INT

**Sintaxis 1** result % = INT (expresión de coma flotante)

Esta función convierte una variable de coma flotante en una variable entera (consulte la instrucción *FLOAT*) y redondea el resultado al siguiente número entero.

**Ejemplo** a = 14.234  
res% = INT(a) / 2 //convierte 'a' en un entero y lo divide  
//entre 2  
Resultado = 7

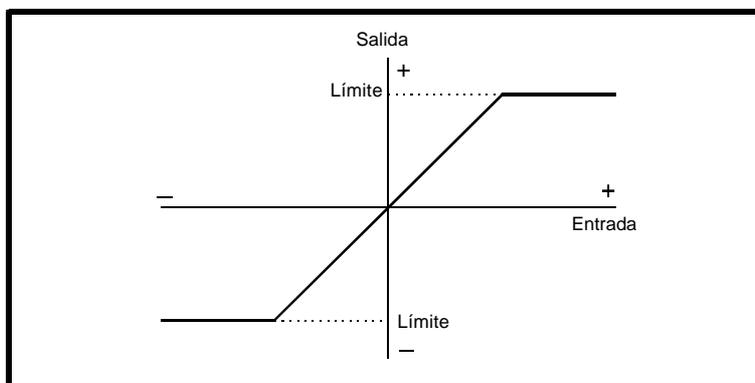
### Nota

**Si el valor entero de la expresión de coma flotante contiene más de 31 bits (214783647), el UD70 se desconectará:Error 50: Maths Error (error matemático).**

## LIMIT

**Sintaxis 1** resultado% = LIMIT (expresión%, límite\_valor\_expresión%)  
resultado = LIMIT (expresión, límite\_valor\_expresión)

Esta función matemática limita el valor que se devuelve a los parámetros del accionamiento y evita que se escriban valores fuera de rango en los parámetros. Esta función se aplica a los valores positivos y negativos de la expresión.



**Ejemplo** #1.21 = LIMIT (inc%, 1000)  
;if inc% > 1000 entonces se limita a 1000  
;if inc% < -1000 entonces se limita a -1000

De esta forma el resultado se limita a 1000.  
Consulte también las instrucciones *MIN* y *MAX*.

## LN

---

**Sintaxis** resultado = LN (expresión)

Esta función permite hallar el logaritmo neperiano de una expresión.

**Ejemplo** y = LN(1.5)

Se obtiene el valor 0,405465108

Consulte también la instrucción *EXP*.

## MAX

---

**Sintaxis** resultado% = MAX (expresión\_A%, expresión\_B%)  
resultado = MAX (expresión\_A, expresión\_B)

Esta función matemática permite hallar la mayor de las dos expresiones.

**Ejemplo** a% = 12  
b% = MAX (a%, 100)

El valor que se obtiene en **b%** es 100 debido a que **a%** es menor que 100. Si **a%** fuese 105, el valor resultante en **b%** sería el valor de **a%** (105).

## MIN

---

**Sintaxis** resultado% = MIN (expresión\_A%, expresión\_B%)  
resultado = MIN (expresión\_A, expresión\_B)

Esta función matemática permite hallar la menor de las dos funciones.

**Ejemplo** a% = 12  
b% = MIN(a%, 100)

El valor obtenido en **b%** es 12 debido a que **a%** es menor que 100. Si **a%** fuese 105, el valor resultante en **b%** sería 100.

## PRINT

---

**Sintaxis** PRINT *lista de expresiones separadas mediante ",."*

La instrucción PRINT permite enviar cadenas o valores al puerto serie PC RS232 del UD70. Todos los elementos incluidos en la lista de expresiones deben separarse mediante comas. Las cadenas aparecen entre comillas dobles (" ").

Entre cada carácter separado por una coma se imprime automáticamente un carácter de tabulación (ASCII 09). Los valores negativos van precedidos del signo -, mientras que los positivos no llevan prefijo.

**Operación en varias tareas** La forma en que funciona la instrucción depende de la tarea, según se indica: En la tarea BACKGROUND, la instrucción PRINT detiene la ejecución del programa hasta que se transmiten todos los caracteres. Se recomienda utilizar la instrucción PRINT en la tarea BACKGROUND solamente.

En las tareas CLOCK y ENCODER, el programa continúa ejecutándose sin esperar. Con una velocidad de 19200 baudios, un carácter tardará 0,5 ms en transmitirse. La instrucción PRINT sobrecargará inmediatamente la tarea CLOCK o ENCODER. Si se ejecuta otra instrucción PRINT antes de que la instrucción PRINT anterior termine, los caracteres restantes no se imprimirán.

La cadena puede contener caracteres no imprimibles si se utiliza una secuencia de escape. Esta secuencia comienza con un carácter de barra invertida [\], seguido del carácter que no se quiere imprimir.

Los caracteres disponibles son los siguientes:

| Carácter | Función         | Carácter ASCII |
|----------|-----------------|----------------|
| \b       | Retroceso       | 8              |
| \p       | Avance de línea | 9              |
| \t       | Tabulador       | 10             |

Continuación de la tabla...

| Carácter | Función                | Carácter ASCII |
|----------|------------------------|----------------|
| \f       | Avance de página       | 12             |
| \r       | Retorno de carro       | 13             |
| \v       | Tabulación vertical    | 11             |
| \\       | Barra invertida simple | 92             |
| \NNN     | (N es un número octal) |                |

**Ejemplo** PRINT "Hola\r\n Adiós\r\n"

Las palabras que se imprimen son las siguientes:

**Hola**  
**Adiós**

### Nota

**No utilice en exceso la instrucción PRINT. Para controlar las variables es preferible utilizar la ventana Watch del juego de herramientas.**

**La instrucción PRINT funciona en los modos de terminal normal y de terminal elemental. En el modo de terminal normal, es preciso utilizar el juego de herramientas DPL para controlar el resultado de la impresión.**

**El puerto RS485 se ha configurado en 19200 baudios, 8 bits de datos, 1 bit de paridad, sin paridad.**

---

## PUTCHAR

---

**Sintaxis** resultado% = PUTCHAR (carácter%)

Función del puerto RS485 que permite introducir un carácter en el puerto de comunicaciones RS485.

Si no resulta posible escribir el carácter (por ejemplo cuando la memoria intermedia del puerto RS485 esté llena), la función devolverá el valor 0; de lo contrario, se obtendrá el valor 1.

## PUTKEY

---

**Sintaxis** resultado% = PUTKEY (carácter%)

Función del puerto RS232 que permite introducir un carácter en el puerto de comunicaciones RS232 sólo durante el funcionamiento en el modo de terminal elemental. Si la función no se realiza correctamente, se obtiene el valor 0; de lo contrario, se obtiene el valor 1.

**Ejemplo** r% = PUTKEY(65) // enviar 'A' a puerto RS232

Consulte también la instrucción *PRINT*.

## REINIT

---

**Sintaxis** REINIT

El UD70 lee los parámetros de configuración en el momento en el que se conecta la alimentación de CA al accionamiento o se reinicia el UD70 solamente. Si un programa DPL modifica estos parámetros, puede utilizar la instrucción REINIT para obligar al UD70 a que lea de nuevo los parámetros a fin de aplicar los cambios.

Esta instrucción no causa el reinicio del sistema, sino que el programa continúa ejecutándose con normalidad a partir de la siguiente instrucción.

**Ejemplo** #17.11 = 25 //Cambiar base de tiempos de la tarea CLOCK a 25 ms  
reinit //reinicializa el UD70 para aplicar el cambio

**NOTE**

**REINIT no lee el parámetro #17.12, Activación de bucle de posición. Para activar o desactivar el bucle de posición mientras el accionamiento está funcionando, utilice Q20%.31 (consulte el Capítulo 8).**

---

**SIN**

---

**Sintaxis** resultado = SIN (expresión)

Esta función matemática permite hallar el seno de un ángulo expresado en radianes.

Consulte las instrucciones *COS* y *TAN*.

**Ejemplo** #1.21 = SIN(3.1416)

**SGN**

---

**Sintaxis** resultado% = SGN (expresión%)  
resultado = SGN (expresión)

Esta función matemática permite obtener un valor en el que se indique el signo, positivo o negativo, de la expresión introducida. Si el valor introducido es positivo o cero, se obtiene el valor 1. Cuando el valor es negativo, se obtiene -1.

**Ejemplo** PRINT SGN (45), SGN (-16), SGN(0)  
Los valores que se imprimen son: 1, -1, 1

**SQR**

---

**Sintaxis** resultado = SQR (expresión)

Esta función matemática permite hallar la raíz cuadrada de una expresión.

**Ejemplo** PRINT SQR (25), SQR (16)

Los valores que se imprimen son: 5 y 4

**TAN**

---

**Sintaxis** resultado = TAN (expresión)

Esta función matemática permite hallar la tangente de un ángulo expresada en radianes.

Consulte las instrucciones *COS* y *SIN*.

**Ejemplo** resultado = TAN (3.1416)

## TIME

---

**Sintaxis** resultado% = TIME

La instrucción TIME permite saber el número de milisegundos que han transcurrido desde la última vez que se conectó la alimentación de CA o se reinició el UD70.

**Ejemplo** t% = TIME

## WDOG

---

**Sintaxis** WDOG

La instrucción WDOG permite actualizar el controlador de secuencia del programa DPL.

El controlador de secuencia se activa mediante el ajuste del parámetro de configuración adecuado en el valor 1, seguido del envío de una instrucción WDOG. Cuando está activada, la instrucción WDOG debe ejecutarse cada 200 ms. Si una instrucción WDOG no se ejecuta en 200 ms, el accionamiento sufre una desconexión del tipo Prc2.

La instrucción WDOG sólo puede utilizarse en las tareas INITIAL y BACKGROUND. El sistema operativo del UD70 actualiza automáticamente cada controlador de secuencia de las tareas en tiempo real.

Consulte un ejemplo en la ayuda en línea.

### Notas

**Si cualquiera de estas tareas se procesa paso a paso, el controlador de secuencia se desconectará.**

**Cuando esto ocurre, el accionamiento presenta una desconexión en Prc2. Como la desconexión del controlador es una función del accionamiento, el programa DPL continúa ejecutándose.**

**Esta función no se encuentra disponible en accionamientos vectoriales.**

---



---

## 8 Funciones

---

En este capítulo se describen las siguientes funciones del UD70:

- Controlador de posición de un solo eje
- Unidad de temporización/contador
- Puertos de E/S digitales
- Almacenamiento en memoria no volátil

### 8.1 Parámetros PLC

El UD70 contiene una serie especial de variables conocidas como parámetros PLC. Estos parámetros se identifican mediante el carácter de subrayado ( \_ ) que precede a las letras P, Q, R y S, a las que sigue un número, como se indica:

| Registro | Rango y tipo   | Menú virtual utilizado para el acceso |
|----------|--|---------------------------------------|
| _Px%     | x = 0 a 99<br>(entero de 32 bits con signo de polaridad) | #70.xx                                |
| _Qx%     | x = 0 a 99<br>(entero de 32 bits con signo de polaridad) | #71.xx                                |
| _Rx%     | x = 0 a 99<br>(entero de 32 bits con signo de polaridad) | #72.xx                                |
| _Sx%     | x = 0 a 99<br>(entero de 32 bits con signo de polaridad) | #73.xx                                |

También es posible acceder a los registros mediante los menús virtuales 70 a 73. Esto ofrece la posibilidad de modificar los registros P, Q, R y S empleando el puerto RS485 o las opciones de comunicación a alta velocidad que haya configuradas (CTNet, Interbus Profibus etc.).

Si se configura #17.19=1, los registros P y Q pueden almacenarse en la memoria no volátil del UD70 (consulte la sección Almacenamiento en la memoria no volátil más adelante en este capítulo). Sin embargo, los registros R y S no pueden almacenarse.

Cuando el bucle de posición interno está activado, los registros Q se reservan para utilizarlos con este bucle.

No existen restricciones de uso de los registros P, R y S, por lo que es posible utilizarlos como registros generales. Si se instala el módulo de opciones CT Net, los registros R y S funcionan como registros de transmisión y recepción durante la transferencia cíclica de datos. Cuando el bucle de posición está desactivado (#17.12 = 0), los registros Q también sirven como registros generales.

**Ejemplos**

```
_P1% = 15      // ajustar _P1% en 15
_Q20%.5 = 1    // ajustar bit 5 de _Q20% en 1
```

## 8.2 Introducción

El sistema operativo del UD70 incorpora un controlador de posición avanzado y un generador de perfil. Los bloques de función se han diseñado para proporcionar control de posición y generación de perfil a partir de lo siguiente:

- Referencia de posición
- Referencia de velocidad
- Cuadro de leva diferencial
- Bloqueo digital (si se utiliza un codificador auxiliar)

Las funciones principales del software de control de posición son las siguientes:

- Control de posición secundaria mediante el uso de rampas lineales o S para el perfil de velocidad.
- Control de velocidad secundaria mediante el uso de rampas lineales o S para el perfil de velocidad.
- Cambio lento entre control de posición y velocidad.
- Bloqueo digital fijo o condicionado con coeficiente secundario de 8 y precisión de 8 decimales.
- Cuadro de leva diferencial que proporciona un control automático de la posición secundaria, respecto de la posición principal.
- Cambio lento al control de posición o velocidad a partir del control de leva o del bloqueo digital.
- Fuente de realimentación de bucle de posición a elegir entre las entradas del codificador de realimentación (principal) y auxiliar (referencia).
- Bucle de control PID de tres designaciones (D puede configurarse como designación predictiva o derivada) con introducción automática del resultado en la referencia de velocidad de acceso rápido (#91.02).

Para utilizar el bucle de control de posición es imprescindible que el accionamiento también esté configurado correctamente. El accionamiento debe utilizar #1.21 como origen de la referencia de velocidad (#1.14 = 3, #1.15 = 1) y la referencia bipolar debe estar activada (#1.10 = 1). Sin embargo, las rampas de velocidad deben estar desactivadas (#2.02 = 0), ya que los bloques de generación de perfil incluyen generación de rampas. Para obtener un rendimiento óptimo, las ganancias del bucle de velocidad del Unidrive (#3.10, #3.11, #3.12) deben ajustarse con precisión; de lo contrario, puede producirse cierta inestabilidad.

Para que el Unidrive aplique velocidades de deceleración rápidas, el accionamiento necesitará una resistencia de frenado. El modo de rampa del Unidrive (#2.04) debe configurarse en rampas rápidas, en lugar de en las rampas controladas convencionales.

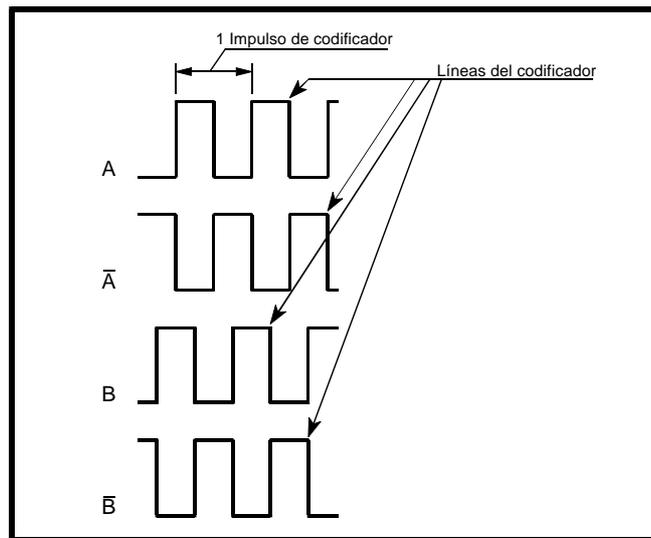
En los registros `_Q20%` y `_Q32%`, la selección entre los distintos bloques se controla mediante el uso de bits individuales. Es posible remitirse a estos bits por separado o acceder a ellos como si se tratase de una palabra completa.

ejemplo

- `_Q20% = 2` ; ajustar bit 1 de `_Q20%` en 1 y los demás bits en 0
- `_Q20%.1 = 1` ; ajustar bit 1 de `_Q20%` en 1 sin modificar los demás bits

### 8.3 Líneas del codificador

Todos los parámetros de posición, velocidad y aceleración del controlador de posición emplean las unidades de las "líneas del codificador". En el diagrama siguiente se muestra la forma de determinar el número de líneas de codificador para un codificador concreto.



**Relación entre las líneas del codificador y los impulsos**

## 8.4 Posición

Los valores de posición son posiciones absolutas obtenidas a partir de la posición de encendido. La posición se mide en líneas del codificador, donde el número de líneas por revolución = 4 \* impulsos del codificador por revolución (ppr). Con un resólver, los impulsos del codificador simulados por revolución dependen de la velocidad máxima del accionamiento (#1.06).

| Velocidad máxima (#1.06) | Resolución del resólver | ppr simulados por el codificador |
|--------------------------|-------------------------|----------------------------------|
| 0 - 3000 rpm             | 14 bits                 | 4096                             |
| 3001 - 12000 rpm         | 12 bits                 | 1024                             |
| 12001 - 30000 rpm        | 10 bits                 | 256                              |

### Velocidad

La cadencia o velocidad máxima se especifica para todo el sistema y puede convertirse a velocidad del motor equivalente, N, en rpm.

Para realizar la conversión entre la velocidad especificada (N) y las líneas del codificador por seg, utilice las ecuaciones siguientes:

$$\text{lines per second} = \frac{N * \text{encoder ppr} * 4}{60}$$

$$N = \frac{\text{lines per second} * 60}{\text{encoder ppr} * 4}$$

### Aceleración

Siempre se especifica la velocidad máxima de aceleración o deceleración de los ejes de una máquina determinada. El tiempo mínimo ( $t_{\min}$ ) necesario para pasar de cero a N rpm puede calcularse, y este valor puede convertirse a variaciones de rpm del motor equivalentes.

Para realizar la conversión entre los valores especificados ( $t_{\min}$  y N) y las líneas del codificador por  $\text{seg}^2$ , utilice las siguientes ecuaciones:

$$\text{lines per second}^2 = \frac{N * \text{encoder ppr} * 4}{t_{\min} * 60}$$

$$t_{\min} = \frac{N * \text{encoder ppr} * 4}{\text{lines per second}^2 * 60}$$

Cuando utilice rampas S, calcule la velocidad de las rampas como si se tratara de una rampa lineal normal. Aunque la rampa S para ese ajuste se demore el mismo tiempo que otra lineal, la velocidad de aceleración máxima aumentará. (Consulte\_Q13%).

NOTA: las unidades asignadas a las rampas del Unidrive no se aplican en estos cálculos, ya que las rampas internas deberían estar desactivadas. El controlador de posición genera y controla todas las rampas del perfil.

## 8-4 Funciones

## 8.5 Activación del controlador de posición

Para activar el controlador de posición es preciso configurar #17.12.

| #17.12 | Bucle de posición   |
|--------|---|
| 0      | Desactivado   |
| 1      | Sincronización con tarea ENCODER<br>5,52 ms (frecuencia de conmutación de 3,6 o 12 kHz)<br>7,36 ms (frecuencia de conmutación de 4,5 o 9 kHz) |
| 2      | Sincronización con tarea SPEED<br>1,38 ms (frecuencia de conmutación de 3,6 o 12 kHz)<br>1,84 ms (frecuencia de conmutación de 4,5 o 9 kHz)   |

El comando REINIT no lee el parámetro #17.12. Para desactivar el controlador de posición mientras el accionamiento está funcionando, ajuste \_Q20%.31 en 1.

### Ejemplo de cálculos

El bucle de posición hará que el motor funcione a una velocidad máxima de 2700 rpm, con un margen de maniobra de 300 rpm para la recuperación de posición. El dispositivo de realimentación del motor será un codificador que funciona a 1024 impulsos por revolución, y el tiempo de aceleración lineal mínimo que necesitará el eje de la máquina para pasar de cero a 2700 rpm será 0,6 segundos.

Velocidad máxima, \_Q14%

$$\text{Lines per second} = \frac{2700 * 1024 * 4}{60} = 184320$$

Velocidad de bloqueo digital, \_Q15%

$$\text{Lines per second} = \frac{3000 * 1024 * 4}{60} = 204800$$

Velocidad de aceleración \_Q12%

$$\text{Lines per second}^2 = \frac{2700 * 1024 * 4}{0.6 * 60} = 307200$$

El ajuste de los parámetros de velocidad y aceleración del bucle de posición debe ser el siguiente:

|                |  |
|----------------|--|
| _Q14% = 184320 | velocidad máxima con perfil de posición.   |
| _Q15% = 204800 | velocidad máxima de bloqueo digital.<br>Proporciona un margen de maniobra de 300 rpm para recuperar la pérdida de posición durante la aceleración. |
| _Q12% = 307200 | velocidad de aceleración.  |
| _Q13% = _Q12%  | ajusta las velocidades de aceleración y deceleración en el mismo valor.  |

## 8.6 Valores predeterminados y de reinicio

Los valores predeterminados que se incluyen en la lista son los que se asignarán a ciertos parámetros cuando el valor obtenido al reiniciar el controlador de posición no sea válido. "-----" indica que el parámetro no dispone de un valor predeterminado.

El controlador de posición puede reiniciarse de las 3 formas siguientes:

UD70 HARD RESET (reinicialización por hardware)

UD70 SOFT RESET (reinicialización por software)

POSITION LOOP RESET (reinicialización por bucle de posición)

Al realizar un **HARD RESET**, el UD70 se pone completamente a cero. Esto ocurre en los siguientes casos:

- #17.19 se ajusta en 1.
- Se conecta la alimentación del Unidrive.

El sistema recuperará los valores del registro Q que se han almacenado en la memoria EEPROM, y todas las rutinas iniciales del archivo del sistema y el programa DPL se ejecutarán. (NOTA: #17.19 almacenará los valores Q actuales y los volverá a leer inmediatamente sin que se produzcan cambios.)

Aunque los valores del registro Q almacenados no se leen al realizar un **SOFT RESET**, la tarea INITIAL de DPL se ejecuta. Esto ocurre en los casos siguientes:

- #MM.00 se ajusta en 1070 y se presiona el botón RESET.
- El accionamiento vuelve a funcionar después de una desconexión.

Esto también causa el reinicio del controlador de posición.

El ajuste del bit de reinicio del bucle de posición (Q32%17) en 1 activa **POSITION LOOP RESET**. Esto no afecta a las tareas DPL del UD70, pero causa el reinicio del controlador de posición.

En las descripciones de los parámetros, el campo RESET indica si el valor que se ha asignado a un parámetro al reinicializar por hardware (H), por software (S) y por bucle de posición (P).

**S/V** indica la asignación del valor almacenado en Qxx%. Si #17.20 = 1, se almacena el último valor anterior a la desconexión de la alimentación.

**N/A** indica que el valor no se ha modificado.

**CAL** indica que el valor se vuelve a calcular en cada ciclo del controlador de posición; por consiguiente, no existe un valor predeterminado o de reinicio.

## 8.7 Descripción de los parámetros

| <b>_Q0%</b>          | <b>Demanda de velocidad final</b>  |
|----------------------|------------------------------------|
| Unidades             | Líneas del codificador por segundo |
| Rango                | $-2^{31}$ a $2^{31}$               |
| Valor predeterminado | -----                              |
| Reinicio             | H = CAL    S = CAL    P = CAL      |

\_Q0% permite obtener la salida de referencia de velocidad del controlador PID en líneas del codificador por segundo. El valor se restablece a 0 al conectar la alimentación.

| <b>_Q1%</b>          | <b>Posición de codificador auxiliar</b> |
|----------------------|---|
| Unidades             | Líneas del codificador                  |
| Rango                | $-2^{31}$ a $2^{31}$                    |
| Valor predeterminado | -----                                   |
| Reinicio             | H = S/V    S = N/A    P = 0             |

\_Q1% muestra la posición del codificador auxiliar (o de referencia) respecto de la posición de encendido. El programa DPL puede modificar el valor de este parámetro en cualquier momento.

| <b>_Q2%</b>          | <b>Referencia de posición de destino secundaria</b> |
|----------------------|---|
| Unidades             | Líneas del codificador                              |
| Rango                | $-2^{31}$ a $2^{31}$                                |
| Valor predeterminado | -----   |
| Reinicio             | H = S/V    S = N/A    P = 0                         |

\_Q2% es la posición de destino del eje secundario y se introduce en el bloque de generación de perfil. Cuando se ha seleccionado, el bucle de control intenta hacer coincidir la posición de realimentación (\_Q8% o \_Q1%) con \_Q2%. Las velocidades de rampa (\_Q12% y \_Q13%), el tipo de rampa (lineal o S) y la velocidad máxima (\_Q14%) hacen variar el perfil de velocidad real.

Durante el funcionamiento en el modo de velocidad (\_Q32%.3 = 1), la posición de parada calculada sirve para actualizar \_Q2%.

| <b>_Q3%</b>          | <b>Referencia de velocidad interna</b> |
|----------------------|--|
| Unidades             | Líneas del codificador por segundo     |
| Rango                | $-2^{31}$ a $2^{31}$                   |
| Valor predeterminado | 0                                      |
| Reinicio             | H = 0    S = 0    P = 0                |

\_Q3% indica la velocidad que debe alcanzar el eje secundario y se introduce en el bloque de generación de perfil. Las velocidades de rampa (\_Q12% y \_Q13%) y el tipo de rampa (lineal o S) hacen variar el perfil de velocidad real. El perfil de velocidad proporciona una referencia de posición actualizada de forma continua al bucle de control de posición.

| <b>_Q4%</b>          | <b>Referencia de eje secundario</b> |
|----------------------|-------------------------------------|
| Unidades             | Líneas del codificador              |
| Rango                | $-2^{31}$ a $2^{31}$                |
| Valor predeterminado | -----                               |
| Reinicio             | H = S/V    S = N/A    P = 0         |

\_Q4% representa la referencia de posición final del eje secundario incluida en el perfil. Este valor puede tener su origen en el perfil de posición secundaria, el perfil de velocidad secundaria, el controlador de bloqueo digital o el perfil del cuadro de leva, en función del ajuste de \_Q32%.2, \_Q32%.3 y \_Q32%.4.

| <b>_Q5%</b>          | <b>Ganancia proporcional</b>  |
|----------------------|-------------------------------|
| Unidades             | 0,1%                          |
| Rango                | 0 a $2^{31}$                  |
| Valor predeterminado | 1000                          |
| Reinicio             | H = S/V    S = N/A    P = N/A |

\_Q5% define la ganancia proporcional del bucle PID. Con un error de 1 cálculo y una ganancia de 1000, el resultado de la designación P será 1.

| <b>_Q6%</b>          | <b>Ganancia integral</b>      |
|----------------------|-------------------------------|
| Unidades             | 0,1% por segundo              |
| Rango                | 0 a $2^{31}$                  |
| Valor predeterminado | 0                             |
| Reinicio             | H = S/V    S = N/A    P = N/A |

\_Q6% define la ganancia integral del bucle PID. Con un error constante de 1 cálculo y una ganancia de 1000, el resultado de la designación I será 1 después de 1 segundo.

| <b>_Q7%</b>          | <b>Ganancia derivada/predictiva</b> |
|----------------------|-------------------------------------|
| Unidades             | 0,1% por segundo                    |
| Rango                | 0 a $2^{31}$                        |
| Valor predeterminado | 1000                                |
| Reinicio             | H = S/V    S = N/A    P = N/A       |

\_Q7% define la ganancia derivada o predictiva (dependiendo de \_Q20%.0) del bucle PID. Con una velocidad constante de variación de error de 1 cálculo por segundo y una ganancia de 1000, el resultado de D será 1.

| <b>_Q8%</b>          | <b>Posición absoluta de eje secundario</b> |
|----------------------|--|
| Unidades             | Líneas del codificador                     |
| Rango                | -231 a $2^{31}$                            |
| Valor predeterminado | -----                                      |
| Reinicio             | H = S/V    S = N/A    P = 0                |

\_Q8% muestra la posición del codificar secundario (realimentación) respecto de la posición de encendido. El programa DPL puede modificar el valor de este parámetro en cualquier momento.

| <b>_Q9%</b>          | <b>Referencia de bucle PID</b> |
|----------------------|--------------------------------|
| Unidades             | Líneas del codificador         |
| Rango                | -231 a 2 <sup>31</sup>         |
| Valor predeterminado | -----                          |
| Reinicio             | H = S/V    S = N/A    P = 0    |

\_Q9% es la referencia de posición respecto del bucle de control PID. Mediante el ajuste de \_Q20%.1 en 1, la referencia del eje secundario incluida en el perfil (\_Q4%) continuará aplicándose en \_Q9%. Si \_Q20%.1 se ajusta en 0, el programa DPL podrá introducir una referencia de posición directamente en \_Q9%; en este caso no se aplicarán rampas a la referencia.

| <b>_Q10%</b>         | <b>Error de seguimiento de PID</b> |
|----------------------|------------------------------------|
| Unidades             | Líneas del codificador             |
| Rango                | -2 <sup>31</sup> a 2 <sup>31</sup> |
| Valor predeterminado | CAL                                |
| Reinicio             | H = CAL    S = CAL    P = CAL      |

\_Q10% muestra el error de seguimiento ocurrido entre la referencia de posición del perfil (\_Q9%) y la posición de realimentación (\_Q8% o \_Q1%). El valor se calcula en líneas del codificador.

| <b>_Q11%</b>         | <b>Coefficiente para bloqueo digital</b> |
|----------------------|--|
| Unidades             | Coefficiente * 10 <sup>8</sup>           |
| Rango                | 1 a 800000000                            |
| Valor predeterminado | 100000000                                |
| Reinicio             | H = S/V    S = N/A    P = N/A            |

\_Q11% especifica el coeficiente que se aplica a la función de bloqueo digital, multiplicado por 108. Esto permite obtener un coeficiente máximo de 8,00000000 (8 decimales). Los valores de \_Q11% se fijan en función del rango máximo o mínimo.

| <b>_Q12%</b>         | <b>Velocidad máxima de aceleración</b>          |
|----------------------|---|
| Unidades             | Líneas del codificador por segundo <sup>2</sup> |
| Rango                | 1 a 2 <sup>31</sup>                             |
| Valor predeterminado | 20480   |
| Reinicio             | H = S/V    S = N/A    P = N/A                   |

\_Q12% define la velocidad máxima de variación de la velocidad (aceleración) que se utiliza en los generadores de perfil de velocidad y posición. Se aplica a la aceleración en los dos sentidos: marcha adelante o marcha atrás.

Si se seleccionan las rampas S (\_Q32%.8 = 1), \_Q12% no se utiliza.

| <b>_Q13%</b>         | <b>Velocidad máxima de deceleración/rampa S</b> |
|----------------------|---|
| Unidades             | Líneas del codificador por segundo <sup>2</sup> |
| Rango                | 1 a 2 <sup>31</sup>                             |
| Valor predeterminado | _Q12%   |
| Reinicio             | H = S/V    S = N/A    P = N/A                   |

\_Q13% define la velocidad máxima de variación de la velocidad (aceleración) que se utiliza en los generadores de perfil de velocidad y posición. Se aplica a la deceleración en los dos sentidos: marcha adelante o marcha atrás.

Si \_Q13% se ajusta en 0, el valor utilizado en la rampa de aceleración (\_Q12%) se introduce en \_Q13%.

Al seleccionar las rampas S (\_Q32%.8 = 1), \_Q13% define la velocidad de rampa. Este valor se configura de la misma forma que en el caso de una rampa lineal. La rampa S durará el mismo periodo de tiempo que la rampa lineal equivalente, lo que generará una aceleración máxima superior (2 \* aceleración máxima).

| <b>_Q14%</b>         | <b>Velocidad máxima</b>            |
|----------------------|------------------------------------|
| Unidades             | Líneas del codificador por segundo |
| Rango                | 0 a 2 <sup>31</sup>                |
| Valor predeterminado | 20480                              |
| Reinicio             | H = S/V    S = N/A    P = N/A      |

\_Q14% define la velocidad máxima de variación de la velocidad (aceleración) que se utiliza en los generadores de perfil de velocidad y posición. El valor definido en \_Q14% no debe ser mayor que el ajuste de velocidad máxima del accionamiento, #1.06. (Consulte \_Q15%.)

| <b>_Q15%</b>         | <b>Velocidad máxima de bloqueo digital</b> |
|----------------------|--|
| Unidades             | Líneas del codificador por segundo         |
| Rango                | $-2^{31}$ a $2^{31}$                       |
| Valor predeterminado | 20480                                      |
| Reinicio             | H = S/V    S = N/A    P = N/A              |

\_Q15% define la velocidad máxima que el controlador de bloqueo digital puede utilizar para recuperar la posición perdida durante la aceleración del eje secundario. \_Q15% debe ajustarse en un valor superior al de \_Q14% para que exista un margen de maniobra que permita la recuperación. Normalmente debería ajustarse en el mismo valor que el dispositivo de bloqueo de velocidad máxima del accionamiento, #1.06.

| <b>_Q16%</b>         | <b>Salida PI máxima</b>            |
|----------------------|------------------------------------|
| Unidades             | Líneas del codificador por segundo |
| Rango                | $-2^{31}$ a $2^{31}$               |
| Valor predeterminado | 32000                              |
| Reinicio             | H = S/V    S = N/A    P = N/A      |

\_Q16% define la salida máxima del bucle PI. Si la salida PI alcanza el límite establecido, el integrador se mantiene en el valor máximo. Esto impide que el integrador se "enrolle" durante los periodos en los que se detecta un error constante o importante. \_Q16% sólo limita la salida conjunta de P y I; la designación D se añade a la salida DESPUÉS de aplicar el límite.

| <b>_Q17%</b>         | <b>Valor absoluto registrado de codificador de realimentación</b> |
|----------------------|---|
| Unidades             | Líneas del codificador  |
| Rango                | $-2^{31}$ a $2^{31}$  |
| Valor predeterminado | -----   |
| Reinicio             | H = S/V    S = N/A    P = N/A                                     |

Si \_Q20%.3 se ajusta en 1, \_Q17% almacenará la posición absoluta del impulso de marcado del codificador de realimentación. \_Q20%.3 se restablecerá a cero después de registrar el valor.

|                      |  |
|----------------------|--|
| <b>_Q18%</b>         | <b>Valor absoluto registrado de codificador auxiliar</b> |
| Unidades             | Líneas del codificador                                   |
| Rango                | $-2^{31}$ a $2^{31}$                                     |
| Valor predeterminado | -----  |
| Reinicio             | H = S/V    S = N/A    P = N/A                            |

Si **\_Q20%.4** se ajusta en 1, **\_Q18%** almacenará la posición absoluta del impulso de marcado del codificador auxiliar (referencia). **\_Q20%.4** se restablecerá a cero después de registrar el valor.

|                |   |
|----------------|---|
| <b>_Q20%</b>   | <b>Palabra de control en mapa de bits</b><br>Todos los bits de <b>_Q20%</b> son bits de lectura y escritura y funcionan como selectores que permiten elegir entre las distintas funciones disponibles. (Consulte los diagramas lógicos.) Los bits que permanecen sin descripción se reservan para utilizarlos posteriormente. |
| <b>_Q20%.0</b> | Selecciona el modo de operación de la designación D.<br>0    Predictiva<br>1    Derivada  |
| <b>_Q20%.1</b> | Se ajusta en 1 para que resulte posible introducir la referencia de posición automática en el bucle PID ( <b>_Q9%</b> = <b>_Q4%</b> ).  |
| <b>_Q20%.2</b> | Se ajusta en 1 para aplicar un filtro de primer orden a D.  |
| <b>_Q20%.3</b> | Se ajusta en 1 para registrar la posición absoluta del impulso de marcado de realimentación en el siguiente impulso. <b>_Q20%.3</b> se restablecerá a 0 cuando se detecte el impulso de marcado.  |
| <b>_Q20%.4</b> | Se ajusta en 1 para registrar la posición absoluta del impulso de marcado auxiliar (referencia) en el siguiente impulso. <b>_Q20%.4</b> se restablecerá a 0 cuando se detecte el impulso de marcado.  |
| <b>_Q20%.5</b> | Se ajusta en 1 para registrar las posiciones actuales absolutas de los codificadores de realimentación y auxiliar mientras la entrada fija está activada. <b>_Q20%.5</b> se restablecerá a cero cuando se detecte el impulso de entrada fija.   |
| <b>_Q20%.6</b> | Se ajusta en 1 para activar la conversión automática de <b>_Q0%</b> a rpm y permitir la escritura en la referencia de velocidad de acceso rápido #91.02.  |
| <b>_Q20%.7</b> | Selecciona la fuente de realimentación del bucle de posición.<br>0    Codificador de realimentación<br>1    Codificador auxiliar (referencia)   |

|                 |   |
|-----------------|---|
| <b>_Q20%.31</b> | <p>_Q20%.31 ofrece una forma de activar y desactivar el bucle de posición mientras el accionamiento está funcionando. #17.12 debe programarse y almacenarse para configurar el bucle de posición; sin embargo, el comando REINIT no vuelve a leer el parámetro #17.12.</p> <p>NOTA: al ajustar _Q20%.31 en 1, todos los parámetros de posición se restablecen a cero, excepto _Q8%, _Q0% y #91.02. Esto significa que #91.02 mantendrá el último valor de referencia de velocidad que se ha calculado y que el accionamiento continuará funcionando a la misma velocidad. El programa DPL debería verificar que _Q0% y #91.02 se restablecen a cero después de ajustar _Q20%.31.</p> <p>0      Activar bucle de posición</p> <p>1      Desactivar bucle de posición</p> |
|-----------------|---|

|                      |   |
|----------------------|---|
| <b>_Q21%</b>         | <b>Valor de inmovilización de codificador de realimentación</b> |
| Unidades             | Líneas del codificador  |
| Rango                | $-2^{31}$ a $2^{31}$  |
| Valor predeterminado | -----   |
| Reinicio             | H = S/V    S = N/A    P = N/A                                   |

Si \_Q20%.5 se ajusta en 1, \_Q21% almacena la posición absoluta del codificador de realimentación cuando la entrada fija del módulo de segundo codificador está activada. \_Q20%.5 se restablece a cero después de registrar el valor.

|                      |  |
|----------------------|--|
| <b>_Q22%</b>         | <b>Valor de inmovilización de codificador auxiliar</b> |
| Unidades             | Líneas del codificador                                 |
| Rango                | $-2^{31}$ a $2^{31}$                                   |
| Valor predeterminado | -----  |
| Reinicio             | H = S/V    S = N/A    P = N/A                          |

Si \_Q20%.5 se ajusta en 1, \_Q22% almacena la posición absoluta del codificador auxiliar (o de referencia) cuando la entrada fija del módulo de segundo codificador está activada. \_Q20%.5 se restablece a cero después de registrar el valor.

| <b>_Q24%</b>         | <b>Realimentación de velocidad de codificador auxiliar</b> |
|----------------------|--|
| Unidades             | Líneas del codificador por segundo                         |
| Rango                | $-2^{31}$ a $2^{31}$                                       |
| Valor predeterminado | CAL  |
| Reinicio             | H = CAL   S = CAL   P = CAL                                |

\_Q24% muestra la velocidad del codificador auxiliar (o de referencia). Este valor se calcula directamente a partir de las líneas del codificador y no se filtra.

| <b>_Q25%</b>         | <b>Realimentación de velocidad de codificador principal</b> |
|----------------------|---|
| Unidades             | Líneas del codificador por segundo                          |
| Rango                | $-2^{31}$ a $2^{31}$  |
| Valor predeterminado | CAL   |
| Reinicio             | H = CAL   S = CAL   P = CAL                                 |

\_Q24% muestra la velocidad del codificador de realimentación. Este valor se calcula directamente a partir de las líneas del codificador y no se filtra.

| <b>_Q26%</b>         | <b>Desfase de posición secundaria</b> |
|----------------------|---------------------------------------|
| Unidades             | Líneas del codificador                |
| Rango                | $-2^{31}$ a $2^{31}$                  |
| Valor predeterminado | 0                                     |
| Reinicio             | H = 0   S = 0   P = 0                 |

\_Q26% define el desfase de posición que se va a añadir a \_Q4%. El perfil de posición de desfase sólo puede utilizarse si no se están empleando otros perfiles de posición. \_Q26% puede aplicarse si el eje secundario se adapta a un perfil de velocidad.

Mientras se utilice el control de posición (\_Q32%.3 = 0) y haya rampas S seleccionadas (\_Q32%.8 = 1), \_Q26% no podrá utilizarse.

| <b>_Q27%</b>         | <b>Desfase de velocidad secundaria</b> |
|----------------------|--|
| Unidades             | Líneas del codificador por segundo     |
| Rango                | $-2^{31}$ a $2^{31}$                   |
| Valor predeterminado | 0                                      |
| Reinicio             | H = 0   S = 0   P = 0                  |

\_Q27% define el desfase de la velocidad secundaria. El desfase puede aplicarse mientras se utilizan otros perfiles de velocidad y posición, y se añade a \_Q4% para obtener la referencia de posición del bucle final, \_Q9%.

|                 |   |
|-----------------|---|
| <b>_Q31%</b>    | <b>Parámetro de indicación en mapa de bits</b><br>Todos los bits de _Q31% son bits de sólo lectura que controla el sistema operativo. Los bits que permanecen sin descripción se reservan para un uso futuro. Un valor 0 en uno de los bits indica que el enunciado es falso. _Q31% se restablece a 0 al ajustar _Q32%.17 en 1. |
| <b>_Q31%.1</b>  | Error de seguimiento del bucle de posición con un valor superior al establecido en _Q37%.   |
| <b>_Q31%.9</b>  | El eje secundario está recuperando la posición perdida durante la aceleración.  |
| <b>_Q31%.10</b> | Perfil de posición en proceso. El eje no ha alcanzado el punto de referencia de destino.  |
| <b>_Q31%.11</b> | Perfil de posición en proceso. No se ha alcanzado la posición de desfase del eje.   |
| <b>_Q31%.12</b> | El eje se ha bloqueado por completo en la referencia principal, obtenida a partir de la entrada del codificador auxiliar.   |
| <b>_Q32%</b>    | <b>Parámetro de control en mapa de bits</b><br>Todos los bits de _Q32% son bits de lectura y escritura, y funcionan como selectores que permiten elegir entre las distintas funciones disponibles. (Consulte los diagramas lógicos.) Los bits que permanecen sin descripción se reservan para utilizarlos posteriormente.       |
| <b>_Q32%.1</b>  | Selecciona la dirección del bloqueo digital. Esto permite bloquear el eje en la referencia de posición o velocidad principal, así como funcionar en cualquier dirección.<br>0 Misma dirección<br>1 Dirección contraria  |
| <b>_Q32%.2</b>  | Selecciona la salida de bloqueo digital como referencia de posición. Este parámetro se utiliza en combinación con _Q32%.3 y _Q32%.4.<br>0 Consulte _Q32%.3.<br>1 Selección de referencia de posición de bloqueo digital   |
| <b>_Q32%.3</b>  | Permite elegir entre la referencia de posición de destino _Q2% y la referencia de posición derivada de la referencia de velocidad secundaria _Q3%. Este parámetro se utiliza junto con _Q32%.2 y _Q32%.4.<br>0 Referencia de posición de destino _Q2%.<br>1 Referencia de posición secundaria, _Q3%.                            |
| <b>_Q32%.4</b>  | Selecciona la salida del cuadro de leva como referencia de posición. La función de leva sólo puede utilizarse si el cuadro de leva se ha inicializado correctamente. Este parámetro se utiliza junto con _Q32%.2 y _Q32%.3.<br>0 Consulte _Q32%.2 y _Q32%.3.<br>1 Selección de referencia de posición de cuadro de leva         |

|                 |   |
|-----------------|---|
| <b>_Q32%.8</b>  | <p>Selecciona rampas S para el perfil de referencia de posición. El perfil de aceleración será senoidal. Cuando se seleccionan rampas S, las velocidades de aceleración y deceleración se definen en _Q13%. Al seleccionar las referencias de bloqueo digital o de cuadro de leva, _Q32%.8 se restablecerá a fin de desactivar las rampas S.</p> <p>NOTA: la referencia de desfase de posición (_Q26%) no podrá utilizarse durante el funcionamiento en el modo de control de posición si hay rampas S seleccionadas.</p> <p>0 Rampas lineales<br/>1 Rampas S</p> |
| <b>_Q32%.12</b> | <p>Selecciona la referencia de posición del cuadro de leva cuando se visualiza el impulso de marcado auxiliar mediante el ajuste de _Q32%.4 en 1. _Q32%.12 se restablece después de ajustar _Q32%.4. El cuadro de leva sólo puede utilizarse si la función CAM se ha inicializado correctamente.</p> <p>0 Nada<br/>1 Activar cuadro de leva con impulso de marcado auxiliar</p>   |
| <b>_Q32%.14</b> | <p>Selecciona la posición de salida de bloqueo digital cuando se visualiza el impulso de marcado auxiliar mediante el ajuste de _Q32%.2 en 1 para visualizar el impulso de marcado auxiliar. _Q32%.14 se restablece después de ajustar _Q32%.2. La entrada del codificar auxiliar también se registra para aplicar un bloqueo fijo (consulte _Q32%.27).</p> <p>0 Nada<br/>1 Activar bloqueo digital con impulso de marcado auxiliar</p>   |
| <b>_Q32%.17</b> | <p>Proporciona un método rápido y sencillo para restablecer todos los parámetros de posición de sólo lectura. Esto no afecta a los parámetros de configuración ni a los valores de posición del codificador registrados. _Q32%.17 se restablece después de restablecer todos los valores. Este parámetro no debe ajustarse mientras el accionamiento está funcionando, ya que puede provocar una parada brusca sin rampas y la consiguiente pérdida de la información de posición.</p> <p>0 Nada<br/>1 Restablecer todos los parámetros de posición</p>           |
| <b>_Q32%.19</b> | <p>Selecciona las rampas S para la referencia de velocidad _Q3%. Aunque la rampa S se demore el mismo tiempo que otra lineal, la velocidad de aceleración máxima aumentará. (Consulte _Q13%). _Q12% no se utiliza si hay rampas S seleccionadas.</p> <p>NOTA: las rampas S no se encuentran disponibles durante la aceleración a velocidad de línea en el modo de control de bloqueo digital.</p> <p>0 Rampas lineales<br/>1 Rampas S</p>   |

|                 |  |
|-----------------|--|
| <b>_Q32%.20</b> | <p>Selecciona rampas S para el desfase de referencia de velocidad, _Q27%. Aunque la rampa S se demore el mismo tiempo que otra lineal, la velocidad de aceleración máxima aumentará. (Consulte _Q13%).<br/>_Q12% no se utiliza si hay rampas S seleccionadas.</p> <p>0 Rampas lineales<br/>1 Rampas S</p>  |
| <b>_Q32%.27</b> | <p>Permite elegir entre los modos de bloqueo digital fijo y condicionado. En el modo de bloqueo fijo, la pérdida de posición debida a la aceleración se recupera mediante la ejecución a una velocidad mayor (_Q15%) que la de referencia. Una vez recuperada la posición, el eje mantiene el bloqueo de posición y velocidad. En el modo de bloqueo condicionado, la pérdida de posición debida a la aceleración no se recupera y el eje secundario funciona con el bloqueo de velocidad solamente.</p> <p>0 Bloqueo digital fijo<br/>1 Bloqueo digital condicionado</p>          |
| <b>_Q32%.28</b> | <p>Las rampas pueden aplicarse a la referencia secundaria para impedir el uso de velocidades de aceleración y deceleración demasiado altas, especialmente cuando el bloqueo digital está activado. Si no se seleccionan rampas, la referencia del codificador auxiliar simplemente se multiplica por el coeficiente (_Q11%).</p> <p>0 Activar rampas<br/>1 Desactivar rampas</p>   |
| <b>_Q32%.29</b> | <p>Desactiva la escritura automática de la posición de retorno (_Q40%) en _Q2% al cambiar del control de bloqueo digital al control de posición (_Q32%.2 de 1 a 0). Cuando está desactivada, _Q2% se actualiza después de cancelar la selección del bloqueo digital, lo que permite parar lentamente sin sobreimpulso.</p> <p>0 Activar posición inicial automática<br/>1 Desactivar posición inicial automática</p>   |
| <b>_Q32%.30</b> | <p>Ofrece una función de disyuntor de seguridad que impide que el eje rebase un punto determinado mientras se utiliza la referencia de posición _Q2%. Si _Q32%.30 se ha ajustado y la velocidad es positiva (dirección de avance), _Q2%, _Q4% y _Q9% se verán obligados a adaptarse a la posición de realimentación actual (_Q8%), lo que provoca una parada inmediata sin rampas.</p> <p>NOTA: esta función sólo funciona durante la ejecución.</p> <p>0 Nada<br/>1 Disyuntor de seguridad activo</p>   |
| <b>_Q32%.31</b> | <p>Ofrece una función de disyuntor de seguridad que impide que el eje rebase un punto determinado mientras se utiliza la referencia de posición _Q2%. Si _Q32%.31 se ha ajustado y la velocidad es negativa (dirección de retroceso), _Q2%, _Q4% y _Q9% se verán obligados a adaptarse a la posición de realimentación actual (_Q8%), lo que provoca una parada inmediata sin rampas.</p> <p>NOTA: esta función sólo funciona durante la ejecución en control de posición y siempre que la fuente de realimentación es _Q8%.</p> <p>0 Nada<br/>1 Disyuntor de seguridad activo</p> |

| <b>_Q34%</b>         | <b>Índice de leva</b>        |
|----------------------|------------------------------|
| Unidades             |                              |
| Rango                | 0 a número de puntos de leva |
| Valor predeterminado | -----                        |
| Reinicio             | H = S/V    S = N/A    P = 0  |

\_Q34% muestra el índice del cuadro de leva actual. Éste indica la sección del perfil de leva que está procesándose en ese momento.

| <b>_Q35%</b>         | <b>Posición de partida principal de cuadro de leva</b> |
|----------------------|--|
| Unidades             | Líneas del codificador                                 |
| Rango                | $2^{31}$ a $2^{31}$                                    |
| Valor predeterminado | -----  |
| Reinicio             | H = S/V    S = N/A    P = 0                            |

\_Q35% registra la posición principal \_Q1% en el momento de la activación del cuadro de leva. Este valor se sobrescribe cada vez que el cuadro de leva se reinicia. El valor se utiliza continuamente mientras el cuadro de leva está activado y no debe sobrescribirlo el usuario.

| <b>_Q36%</b>         | <b>Posición de partida principal de cuadro de leva</b> |
|----------------------|--|
| Unidades             | Líneas del codificador                                 |
| Rango                | $-2^{31}$ a $2^{31}$                                   |
| Valor predeterminado | -----  |
| Reinicio             | H = S/V    S = N/A    P = 0                            |

\_Q36% registra la referencia de posición secundaria \_Q4% en el momento de la activación del cuadro de leva. Este valor se sobrescribe cada vez que el cuadro de leva se reinicia. El valor se utiliza continuamente mientras el cuadro de leva está activado y no debe sobrescribirlo el usuario.

| <b>_Q37%</b>         | <b>Error de seguimiento máximo</b> |
|----------------------|------------------------------------|
| Unidades             | Líneas del codificador             |
| Rango                | $-2^{31}$ a $2^{31}$               |
| Valor predeterminado | -----                              |
| Reinicio             | H = S/V    S = N/A    P = N/A      |

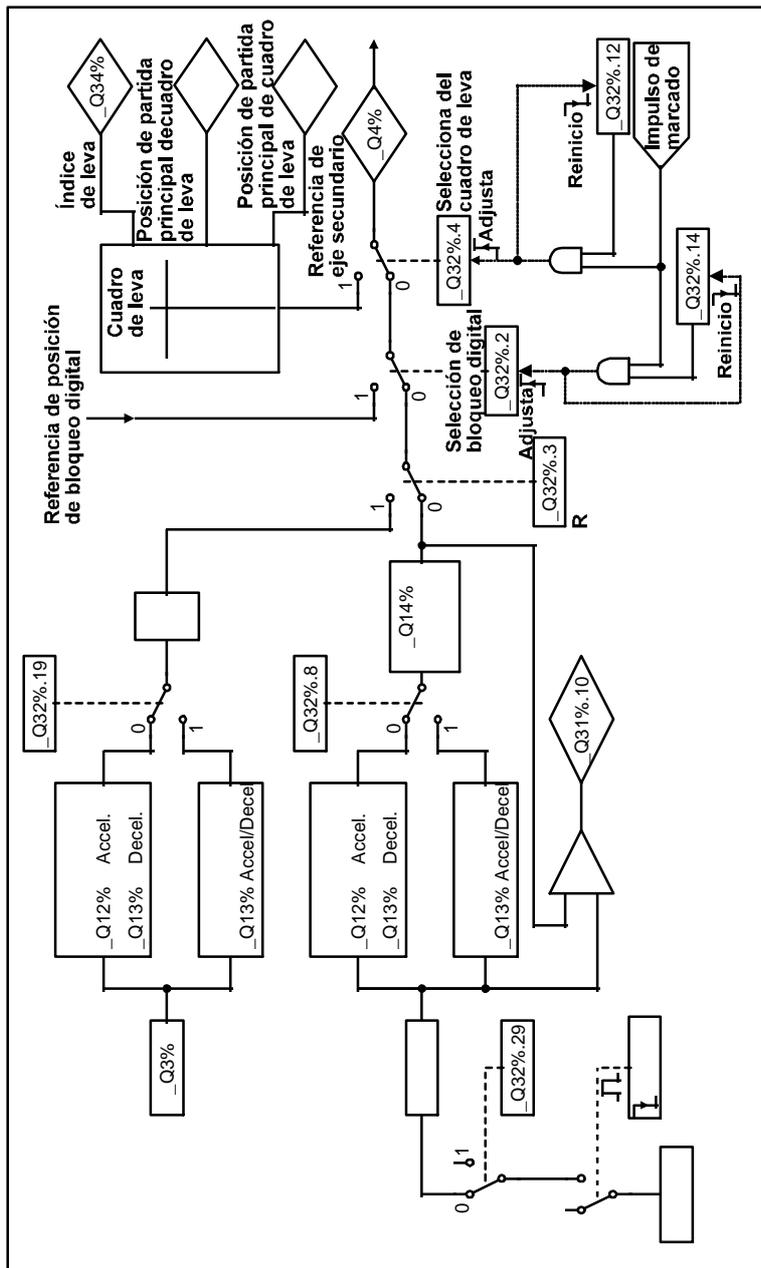
\_Q37% define el valor máximo que puede tener un error de posición o seguimiento (\_Q10%) en el bucle PID. Si el valor máximo se supera, \_Q31%.1 se ajusta.

| <b>_Q40%</b>         | <b>Posición de retorno</b>    |
|----------------------|-------------------------------|
| Unidades             | Líneas del codificador        |
| Rango                | $-2^{31}$ a $2^{31}$          |
| Valor predeterminado | -----                         |
| Reinicio             | H = S/V    S = N/A    P = N/A |

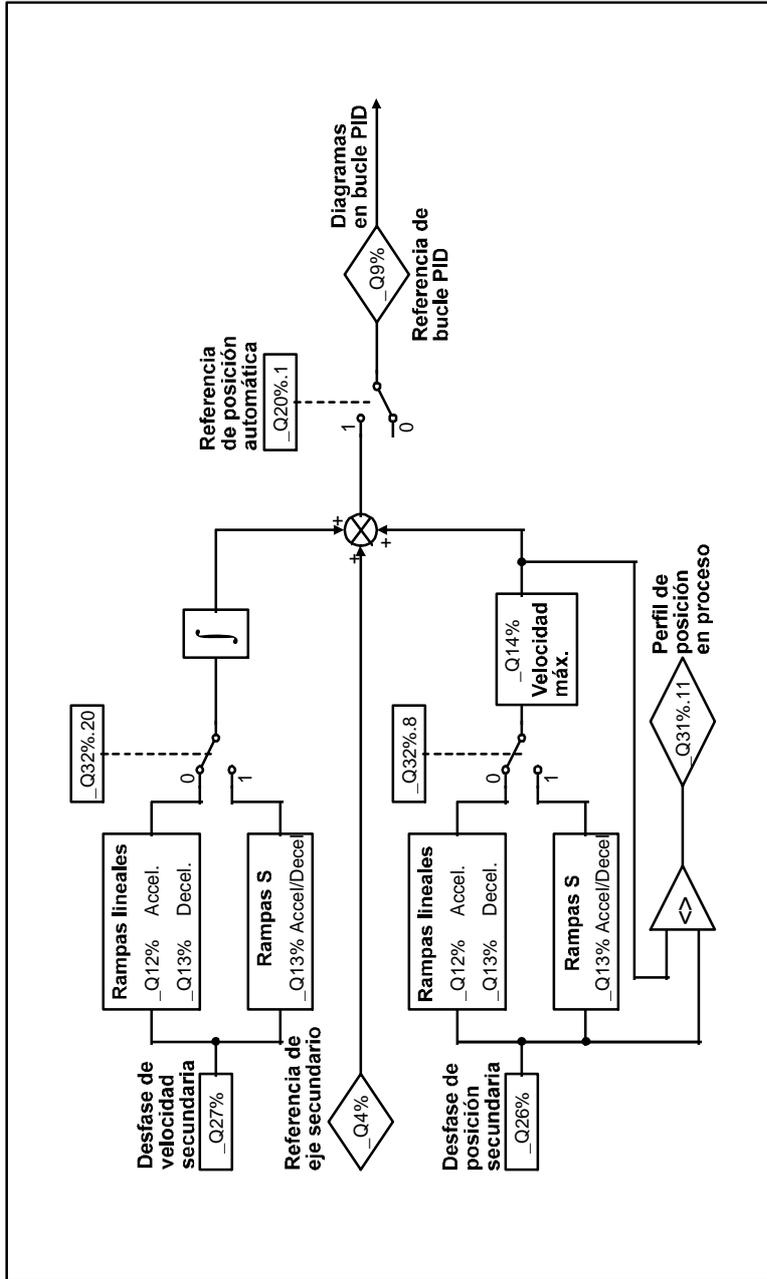
Al cambiar del modo de bloqueo digital al modo de control de posición ( $\_Q32\%.2$  de 1 a 0),  $\_Q40\%$  se introduce en la referencia de posición principal  $\_Q2\%$  y, de este modo, el eje vuelve a la posición definida en  $\_Q40\%$ . La escritura automática puede desactivarse ajustando  $\_Q32\%.29 = 1$ .

## 8.8 Diagramas lógicos

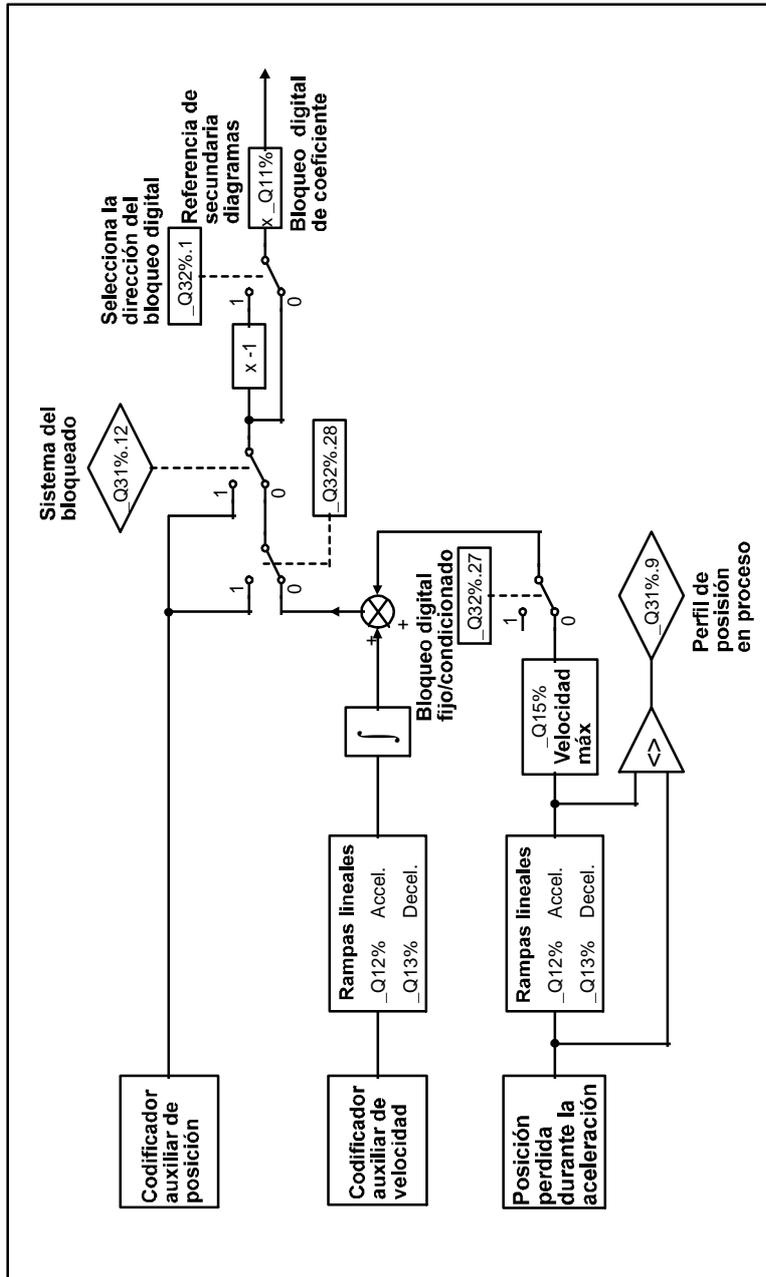
### Referencia de posición principal/secundaria



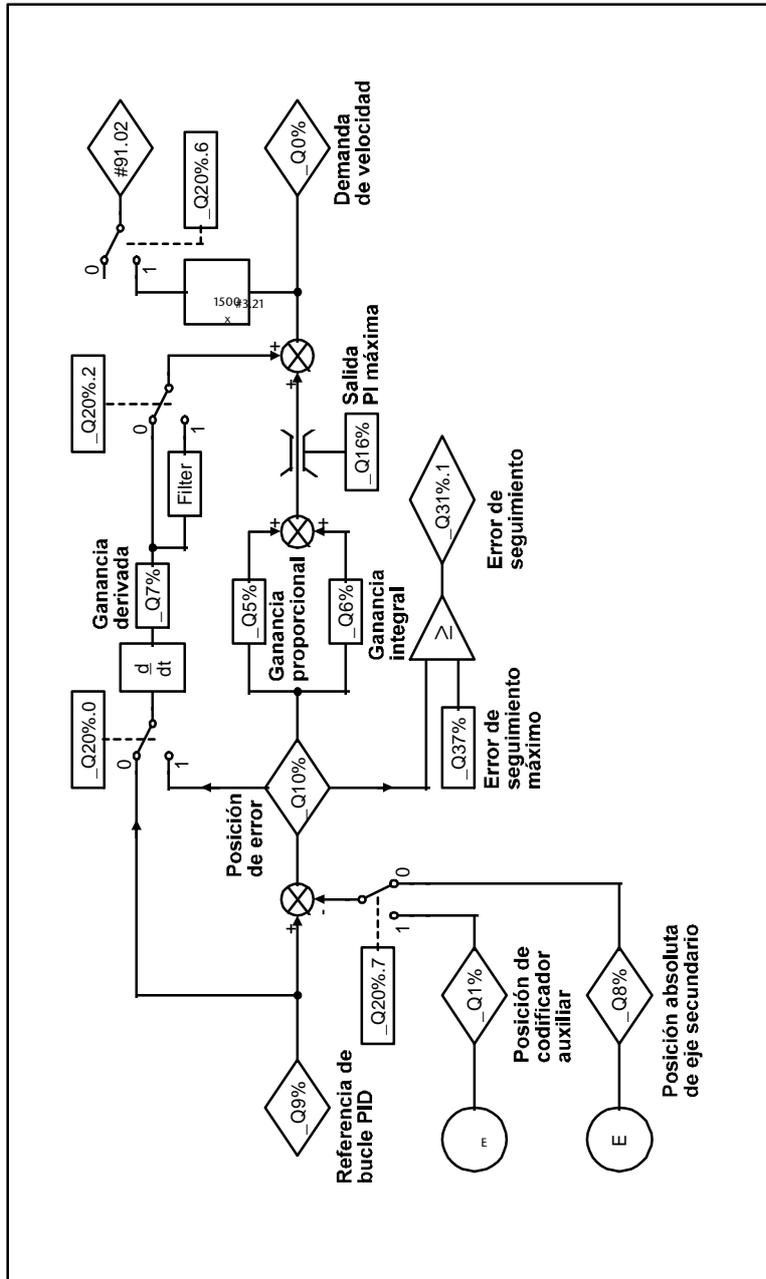
## Desfase de posición secundaria



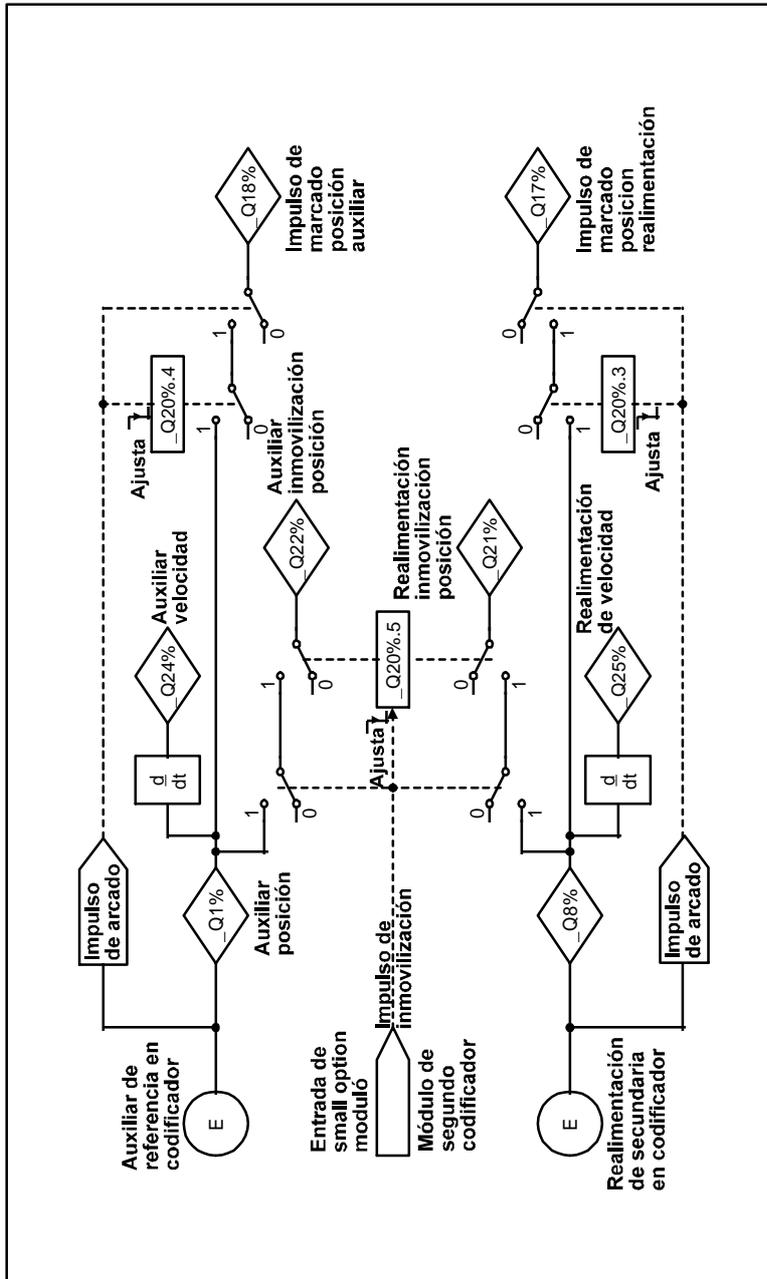
## Control de bloqueo digital



## Fuente de realimentación y bucle PID



Entradas de codificador e impulsos de marcado



El impulso de inmovilización sólo se encuentra disponible en la segunda generación de los módulos de segundo codificador que se han instalado en los accionamientos Unidrive con software V3.00.00, y versiones posteriores. Para activar este impulso, es preciso realizar el ajuste #16.10 = 1. (Consulte la guía del usuario de Unidrive.)

## 8.9 Bloqueo digital

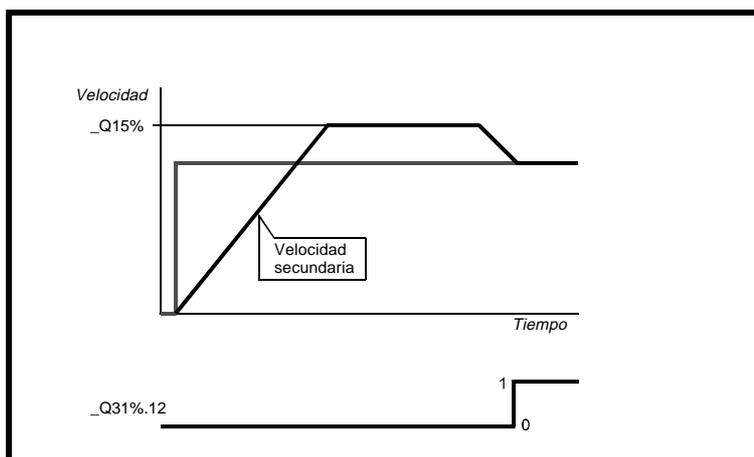
La función de bloqueo digital permite fijar la posición y la velocidad del eje secundario en el codificador principal de forma permanente o condicionada. También se puede introducir un coeficiente de bloqueo digital (\_Q11%) de 8 decimales entre la referencia y la realimentación.

### Bloqueo digital fijo

Bloqueo de posición que se aplica entre la referencia principal y el eje secundario y que se selecciona mediante el ajuste \_Q32%.27 = 0. Cuando el control de bloqueo digital está activado (\_Q32%.2 = 1), el eje secundario acelera a la velocidad de bloqueo digital máxima (\_Q15%) utilizando rampas lineales o rampas S. (NOTA: Si se seleccionan rampas S mientras el bloqueo digital está activado, la velocidad del eje secundario aumentará en rampa hasta la velocidad principal y la posición se recuperará mediante rampas lineales.) El eje continuará funcionando a la velocidad máxima hasta que se detecte un error de posición generado durante la aceleración.

Después de recuperar la posición, la velocidad del eje secundario se bloqueará en la velocidad principal, multiplicada por el coeficiente, \_Q11%. \_Q31%.12 se ajustará después de aplicar el bloqueo de posición absoluta.

NOTA: la velocidad de bloqueo digital máxima (\_Q15%) siempre debe ajustarse en un valor mayor que el de la velocidad de línea máxima, multiplicado por el coeficiente (\_Q11%). De no cumplirse esta condición, el eje podría no alcanzar la posición absoluta de bloqueo.

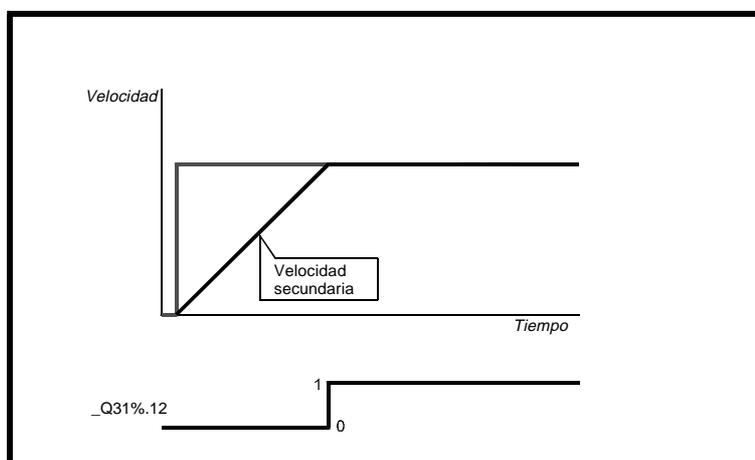


**Efecto del bloqueo digital fijo en la velocidad del motor secundario**

## Bloqueo digital condicionado

El bloqueo condicionado es un bloqueo de velocidad entre la referencia principal y el eje secundario y se selecciona mediante el ajuste  $\_Q32\%.27 = 1$ . Cuando el bloque digital está activado ( $\_Q32\%.2 = 1$ ), el eje secundario acelera a la velocidad principal, multiplicada por el coeficiente,  $\_Q11\%$ , utilizando rampas lineales. (NOTA: si se seleccionan rampas S mientras el bloqueo digital está activado, el eje acelerará a la velocidad principal utilizando rampas S.) Los errores de posición generados durante la aceleración se ignoran.

$\_Q31\%.12$  se ajusta después de aplicar el bloqueo de velocidad. El bloqueo de posición permite mantener fija la velocidad del eje secundario en la referencia principal.



**Efecto del bloqueo digital condicionado en la velocidad del motor secundario**

## Utilización de rampas S con bloqueo digital

Al seleccionar rampas S, la velocidad principal deberá mantenerse constante, en la medida de lo posible, mientras el eje secundario acelera hasta alcanzar esta velocidad. Si no se mantiene constante, pueden surgir problemas al completar el perfil de rampa S, en particular cuando la velocidad de la rampa sea bastante lenta.

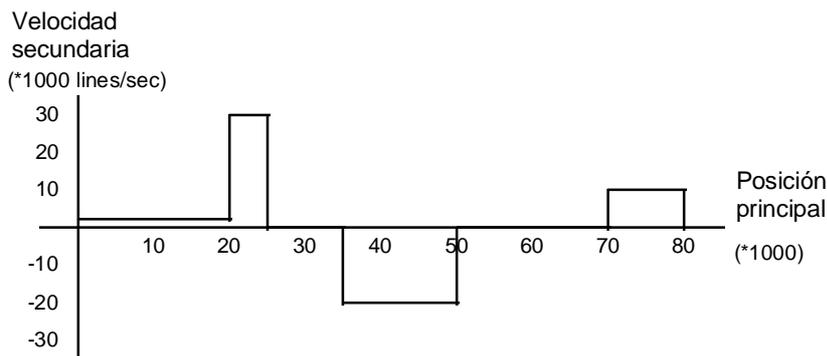
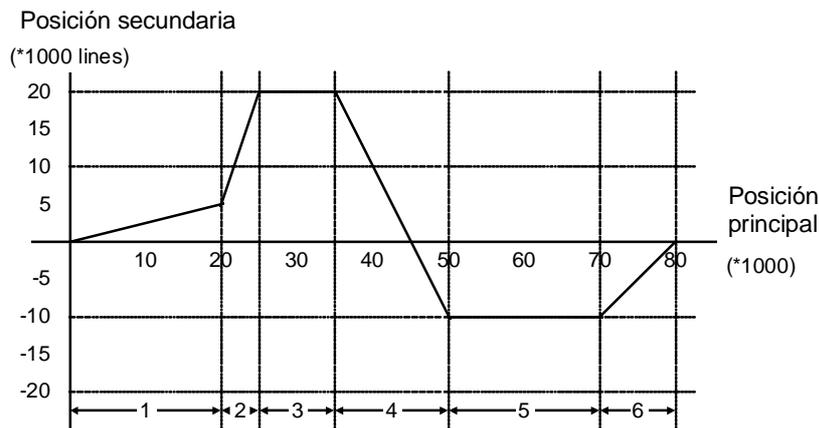
## Utilización del impulso de marcado para activar el bloqueo digital

El bloqueo digital puede activarse utilizando la entrada de impulso de marcado del codificador auxiliar, mediante el ajuste  $\_Q32\%.14 = 1$ . Al producirse el impulso,  $\_Q32\%.2$  se ajusta en 1 y  $\_Q32\%.14$  se restablece a 0. Si se selecciona el bloqueo fijo ( $\_Q32\%.27 =$ ), el eje secundario alcanza la posición perdida respecto del eje principal a partir del momento en el que se produce el impulso de marcado.

## 8.10 Función de leva

La función de leva ofrece un método para conseguir que un eje secundario siga el movimiento continuo de un codificador principal. El perfil de leva puede dividirse en dos secciones, en las que se conocen las posiciones de partida y destino de los ejes principal y secundario. Para crear el cuadro de leva, defina el cambio de posición de los ejes, respecto de la posición de partida, en cada sección para los codificadores principal y secundario. El "punto de referencia" es la posición que ocupa el eje secundario en el momento en el que se activa la función de leva. El software utiliza la interpolación lineal para desplazar la leva por cada sección conforme cambia la posición principal.

En las secciones siguientes se describe detalladamente la forma de crear un cuadro de leva utilizando el ejemplo de perfil de leva siguiente. El perfil de velocidad necesario para generar el perfil de posición también se indica a continuación.



## Definición del cuadro de leva

---

Para poder utilizar la función de leva, es necesario inicializar el cuadro adecuado. El mejor momento para realizar esta operación es el inicio del programa, en la tarea INITIAL.

setup% = CAMINIT (master%, slave%, tablesize%, reserved1%, reserved2%)

donde:

setup% = 0 inicialización incorrecta

= 1 inicialización correcta

master% = nombre de matriz principal

slave% = nombre de matriz secundaria

tablesize% = número de puntos de leva definidos. Aunque una matriz pueden contener más puntos, ambas deben tener este tamaño, como mínimo, para que se inicialicen correctamente.

reserved1% = 0 este argumento se reserva para un uso posterior, y debe ajustarse en 0.

reserved2% = 0 este argumento se reserva para un uso posterior, y debe ajustarse en 0.

Las matrices master% y slave% pueden ser matrices constantes o variables y pertenecer a tipos diferentes. El número máximo de elementos que puede contener una sola matriz es 500. Las matrices constantes se definen al compilar el programa y no pueden modificarse posteriormente a menos que se vuelva a compilar el programa. Los programas DPL se almacenan en la memoria de programas de 96 K, por lo que la única restricción consiste en que el programa compilado no debe ocupar más de 96 K.

Las matrices dinámicas utilizan la memoria RAM de 8 K, por lo que el número total de elementos que puede contener una matriz es bastante menor. La ventaja que ofrecen estas matrices es que el programa DPL puede modificar los valores mientras se utiliza la función de leva. Sin embargo, la modificación de los elementos debe concluir antes de \_Q34% se aplique en 3 elementos.

En el perfil de leva del ejemplo incluido a continuación, los nombres de matriz serán "master%" y "slave%". El perfil de leva puede dividirse en 6 secciones independientes, por lo que se utilizarán 6 elementos de matriz para volver a crear el perfil.

## Creación del cuadro de leva

Las posiciones de leva al principio y al final de cada sección pueden determinarse a partir del perfil.

| Sección | Principal |       | Secundario |        | Cambio de posición |        |
|---------|-----------|-------|------------|--------|--------------------|--------|
|         | Principio | Fin   | Principio  | Fin    | master%            | slave% |
| 1       | 0         | 20000 | 0          | 5000   | 20000              | 5000   |
| 2       | 20000     | 25000 | 5000       | 20000  | 5000               | 15000  |
| 3       | 25000     | 35000 | 20000      | 20000  | 10000              | 0      |
| 4       | 35000     | 50000 | 20000      | -10000 | 15000              | -30000 |
| 5       | 50000     | 70000 | -10000     | -10000 | 20000              | 0      |
| 6       | 70000     | 80000 | -10000     | 0      | 10000              | 10000  |

En las dos últimas columnas se obtienen los valores necesarios para cada matriz. Las matrices pueden definirse de la siguiente forma:

```
CONST master%{  
20000,5000,10000,15000,20000,10000  
}  
CONST slave%{  
5000,15000,0,-30000,0,10000  
}
```

Si el eje secundario debe volver siempre a la misma posición al final de cada ciclo de leva, la suma de todos los elementos de la matriz secundaria debería ser igual a 0. Para facilitar este proceso, puede crear una matriz utilizando una hoja de cálculo y, a continuación, copiar y pegar las series de números en el editor DPL.

Para llamar a la leva en el programa DPL, se utiliza la línea de comandos siguiente.

```
setup% = CAMINIT(master%, slave%, 6, 0, 0)
```

## Nueva definición del cuadro de leva

El cuadro de leva puede volver definirse mediante un programa DPL siempre que se desee. Para esto, es preciso cancelar la selección del control de leva (`_Q32%4` restablecido a 0) antes de llamar de nuevo a la función CAMINIT. Es posible utilizar cualquier combinación de matrices definidas previamente, y ambas matrices deben contener el número de puntos de leva establecido, como mínimo.

## **Utilización del impulso de marcado para activar el cuadro de leva**

---

El cuadro de leva puede activarse utilizando la entrada del impulso de marcado del codificador auxiliar, mediante el ajuste  $\_Q32\%.12 = 1$ . Al producirse el impulso,  $\_Q32\%.4$  se ajusta en 1,  $\_Q32\%.12$  se restablece a 0 y los puntos definidos por la función CAM se utilizan para calcular la referencia de posición necesaria.

## **8.11 Cambio de referencia**

El controlador de posición ofrece la posibilidad de alternar varios métodos de control sin causar vacíos en el perfil del eje secundario.

### **Cambio a control de posición**

---

Durante la ejecución en los modos de control de velocidad, de bloqueo digital o de cuadro de leva, la posición de parada del eje se calcula constantemente y se introduce en  $\_Q2\%$ . La configuración de rampa actual definida en  $\_Q12\%$  y  $\_Q13\%$  permite calcular la posición de parada del eje. Las rampas S no pueden utilizarse al cambiar al control de posición.

NOTA: cuando se produce un cambio de control de bloqueo digital a control de posición, se introduce una posición de retorno ( $\_Q40\%$ ) en  $\_Q2\%$  una vez, con lo que la posición de parada calculada se sobrescribe. Para que el eje pueda desplazarse hasta la posición de parada calculada, puede desactivar esta función mediante el ajuste  $\_Q32\%.29 = 1$ .

### **Cambio a control de velocidad**

---

Cuando el origen de la referencia cambia de control de posición, bloqueo digital o cuadro de leva a control de velocidad, la referencia de velocidad aumenta o desciende en rampa hasta alcanzar el valor de referencia de velocidad establecido,  $\_Q3\%$ . Las rampas definidas en  $\_Q12\%$  y  $\_Q13\%$  permiten asegurarse de que el cambio se realizará lentamente. Al cambiar al control de referencia de velocidad, es posible seleccionar rampas S ( $\_Q32\%.19 = 1$ ).

### **Cambio a control de bloqueo digital**

---

Desde el momento en que se activa el bloqueo digital ( $\_Q32\%.2 = 1$ ), el controlador de posición fija la posición del eje secundario en la referencia principal. El eje secundario se acelera hasta alcanzar la velocidad del eje principal, multiplicada por el coeficiente ajustado en  $\_Q11\%$ .

Si se selecciona el bloqueo fijo ( $\_Q32\%.27 = 0$ ), el eje secundario se acelera a la velocidad de bloqueo digital máxima ( $\_Q15\%$ ) hasta que se recupera la posición. A partir de ese momento, la velocidad se reduce a la velocidad de línea principal, multiplicada por el coeficiente ajustado en  $\_Q11\%$ .

## **Alternancia del control de bloqueo digital**

---

Desde el momento en que se activa el bloqueo digital ( $\_Q32\%.2 = 1$ ), el controlador de posición fija el eje secundario en la referencia principal. Si el bloqueo es fijo ( $\_Q32\%.27 = 0$ ), el eje secundario recupera la posición perdida durante la aceleración. Mientras el bloqueo digital está activado, los valores internos de referencia de velocidad postrampa, referencia de posición y posición de parada se actualizan continuamente.

Al pasar del modo de bloqueo digital al de control de velocidad, se dispone de una referencia de velocidad postrampa. La referencia interna aplicará en rampa hasta alcanzar el valor de referencia de velocidad  $\_Q3\%$ , lo que garantiza un funcionamiento uniforme durante el cambio de referencia. En este caso, las rampas S deben estar desactivadas ( $\_Q32\%.19 = 0$ ).

Si el bloqueo digital cambia al modo de control de posición, la posición de retorno ( $\_Q40\%$ ) se introduce en  $\_Q2\%$ , lo que proporciona una referencia de posición inicial al eje. Mediante el uso de  $\_Q12\%$  y  $\_Q13\%$ , la referencia de posición postrampa empieza a aplicarse hasta que se alcanza la referencia de posición principal. Si es preciso invertir la dirección del eje, tendrá que utilizar rampas. No obstante, las rampas S tendrán que estar desactivadas ( $\_Q32\%.8 = 0$ ).

## **Cambio a control de cuadro de leva**

---

Cuando el cuadro de leva está activado ( $\_Q32\%.4 = 1$ ), la referencia de posición responde a un perfil predefinido, relacionado con el eje principal. Las posiciones de partida de los ejes principal y secundario se introducen en  $\_Q35\%$  y  $\_Q36\%$  respectivamente.

Al cambiar del control de cuadro de leva al de posición, la posición de parada del eje se calcula constantemente y se introduce en  $\_Q2\%$ . La configuración de rampa actual definida en  $\_Q12\%$  y  $\_Q13\%$  sirve para calcular la posición de parada del eje. Las rampas S no pueden utilizarse al cambiar al control de posición.

Consulte los ejemplos de aplicaciones y programas que utilizan bloqueo digital, cuadro de leva y cambio de referencia en el archivo de ayuda.

## **Control interno de cambios**

---

Algunas configuraciones de cambio del generador de perfil no son válidas, por lo que el controlador de posición las restablecerá automáticamente.

## **Bloqueo digital**

---

Con el bloqueo digital no es posible utilizar las rampas S para controlar la posición; por lo tanto,  $\_Q32\%.8$  se restablece siempre a cero. Al iniciar el bloqueo digital mediante el uso del impulso de marcado auxiliar,  $\_Q32\%.2$  se ajusta en 1 y  $\_Q32\%.14$  se restablece a 0 después de detectar la entrada del impulso.

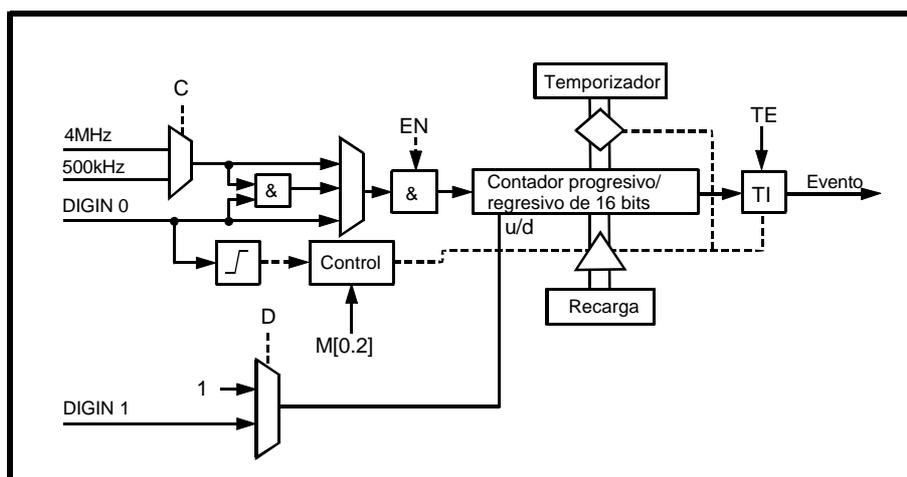
## Cuadro de leva

Con el cuadro de leva no es posible utilizar las rampas S para controlar la posición; por lo tanto,  $Q32\%.8$  siempre se restablece a 0. Al iniciar el cuadro de leva mediante el uso del impulso de marcado auxiliar,  $Q32\%.4$  se ajusta en 1 y  $Q32\%.12$  se restablece a 0 después de detectar el impulso. Si el cuadro de leva no se inicializa correctamente, no será posible ajustar  $Q32\%.4$ .

## Reinicio

Cuando se solicita el reinicio del generador de perfil mediante el uso de  $Q32\%.17$ , los registros se ponen a cero. Después de borrar los registros  $Q32\%.17$  se restablece a 0.

## 8.12 Unidad de temporización/contador



**Diagrama lógico de la unidad de temporización/contador**

El equipo de temporización/contador que incorpora el UD70 ofrece las características siguientes:

- La posibilidad de seleccionar la velocidad de conteo entre 500kHz y 4MHz.
- La posibilidad de seleccionar la dirección de conteo.
- El contador puede circuitarse mediante una fuente de señal externa.
- La tarea EVENT puede iniciarse cuando el temporizador/contador funciona por exceso o por defecto, o mediante la aplicación de la lógica 1 a la entrada 0 *DIGIN* (TTL digital).
- El valor de la unidad de temporización/contador puede fijarse mediante la aplicación de la lógica 0 a la entrada 0 *DIGIN*.

- La unidad de temporización/contador puede cronometrarse mediante una transición 1 a 0 en la entrada 0 *DIGIN*. La frecuencia de entrada máxima es de 1MHz.

La unidad de temporización/contador se controla utilizando los parámetros virtuales del menú 85, según se indica:

| Parámetro | Función                                    |
|-----------|--|
| #85.01    | Palabra de control                         |
| #85.02    | Palabra de estado                          |
| #85.03    | Valor de temporizador/contador (0 ~ 65535) |
| #85.04    | Valor de recarga de temporizador/contador  |
| #85.05    | Valor de temporizador/contador en modo 2   |

#### Palabra de control (#85.01)

| Bit |    | Notas  |
|-----|----|--|
| 0   | TE | Activar EVENT de temporizador<br>0 = desactivar origen de EVENT<br>1 = activar origen de EVENT<br>Si TE y TI <i>Activar evento</i> = 1, la tarea EVENT se ejecuta. Para borrar el evento se precisa la lectura del registro de estado.   |
| 1   | EN | Activar temporizador<br>0 = desactivar todo cálculo<br>1 = activar cálculo   |
| 2   | R  | Modo de recarga en ajuste automático de temporizador<br>0 = sin recarga. El temporizador se ajusta automáticamente y continúa con el cálculo<br>1 = recarga automática. El valor del registro de recarga se inserta en el contador<br><br>Nota 1: este bit no produce ningún efecto en el modo 2 (modo de captura)<br>Nota 2: el ajuste automático se define como funcionamiento por exceso o por defecto. |
| 3   | C  | Selección de reloj interno previo a escala<br>0 = reloj de 500 kHz<br>1 = reloj de 4 MHz   |
| 4   | D  | Control de dirección<br>0 =Conteo ascendente (u/d = 1)<br>1 en DIGIN1: controla la dirección de conteo<br><br>Si DIGIN1 = 0, el temporizador cuenta hacia delante<br>Si DIGIN1 = 1, el temporizador cuenta hacia atrás   |

| Bit   |   | Notas  |
|-------|---|--|
| 5 ~ 7 | M | <p>Modo de temporizador</p> <p><b>Modo 0 (000): modo normal</b></p> <p>La fuente de reloj interna (4 MHz o 500 KHz) cronometra el contador. El indicador del evento TI se configura cuando el contador se ajusta automáticamente.</p> <p><b>Modo 1 (001): modo externo</b></p> <p>El contador se cronometra desde una fuente externa DIGINO (en el polo -). El indicador del evento TI se configura cuando el contador se ajusta automáticamente.</p> <p><b>Modo 2 (010): modo de captura</b></p> <p>La fuente de reloj interna (4 MHz o 500 KHz) cronometra el contador y la transición de 1 a 0 en DIGINO hace que el valor actual del contador quede fijo en el registro TIMER. A continuación, el registro RELOAD se vuelve a cargar en el contador y el indicador TI se configura. (Utilice #85.05, en lugar de #85.03, para leer el valor del temporizador en el modo 2.)</p> <p><b>Modo 3 (011): modo de activación</b></p> <p>La fuente de reloj interna (4 MHz o 500 KHz) cronometra el contador. Sin embargo, el indicador del evento TI se configura al detectar una transición de 1 a 0 en DIGINO, EN LUGAR de configurarse en el momento en el que el contador se ajusta automáticamente. El modo de recarga funciona de la misma forma que en el modo 0 o 1.</p> <p><b>Modo 4 (100): modo circuitado</b></p> <p>La fuente de reloj interna (4 MHz o 500 KHz) cronometra el contador y la entrada DIGINO (activa de bajo nivel) circuita la entrada del contador. El indicador del evento TI se configura cuando el contador se ajusta automáticamente.</p> |

#### Palabra de estado (#85.02)

| Bit |    | Notas  |
|-----|----|--|
| 0   | TI | <p>Indicador de evento de temporizador</p> <p>0 indica que no ha ocurrido ningún evento<br/>1 indica que ha ocurrido un evento</p> <p>Los dos eventos definidos son los siguientes:<br/>Modos 0, 1 y 4: ajuste automático del contador (funcionamiento por exceso o defecto)<br/>Modos 2 y 3: transición de 1 a 0 en DIGINO</p> <p>Si el indicador TE se ajusta en #85.01, el bit TI se borra automáticamente; de lo contrario, se elimina al leer la palabra de estado.</p> |
| 1   | OV | <p>Indicador de ajuste automático</p> <p>0 = Sin ajuste automático<br/>1 = Ajuste automático del contador/temporizador</p> <p>Este indicador es válido para <b>todos</b> los modos de temporizador.</p> <p>Al leer el registro de estado, el bit OV se borra automáticamente.</p>  |

**Temporizador #85.03** El valor actual del contador/temporizador puede visualizarse e introducirse siempre que se desee. Sin embargo, este parámetro no debe leerse con el modo 2 seleccionado (ni siquiera en la ventana **Watch**); en su lugar, será preciso utilizar #85.05 para leer el valor del contador/temporizador.

Éste es un temporizador de 16 bits con un rango de entre 0 y 65535.

**Valor de recarga #85.04** El indicador OV se configura cuando el contador/temporizador funciona por exceso o por defecto. En los modos 0, 1 y 4, el indicador del evento TI también se configura. Si el modo de recarga está activado (R está ajustado en 1), el temporizador/contador incluye el contenido del registro RELOAD al inicializarse. (En el modo 2, el proceso de recarga se realiza cuando DIGINO pasa de 1 a 0).

Si R (bit 2 de #85.01) se ajusta en 0, el temporizador/contador se ajusta automáticamente y continúa realizando el cálculo. El valor del registro de recarga puede visualizarse e introducirse siempre que se desee.

**Valor de temporizador en modo 2 (#85.03)** Utilice este parámetro para visualizar el valor del contador/temporizador que se ha registrado con el modo 2 seleccionado. No lea el parámetro #85.03.

Consulte el ejemplo de uso de la unidad de temporización/contador en el tema Unidad de temporización de la ayuda en línea del juego de herramientas DPL.

## 8.13 Puertos de E/S digitales

Normalmente, el UD70 dispone de dos entradas digitales y una salida digital. Estas entradas y la salida presentan una lógica TTL (lógica de 5 V), y pueden utilizarse con el conector del puerto RS485 de 9 terminales, según se indica:

| Clavija | Descripción                    |
|---------|--------------------------------|
| 1       | RS485 aislado 0 V              |
| 2       | RS485 /TX                      |
| 3       | RS485 /RX                      |
| 4       | DIGIN0 (entrada digital TTL 0) |
| 5       | DIGIN1 (entrada digital TTL 1) |
| 6       | RS485 TX                       |
| 7       | RS485 RX                       |
| 8       | Salida digital TTL             |
| 9       | Accionamiento de 0 V           |

Las entradas digitalesTTL se utilizan junto con la unidad de temporización/contador. Un programa DPL puede leer directamente estas entradas.

La salida digital TTL también puede controlarse directamente desde un programa DPL. Los parámetros virtuales incluidos a continuación proporcionan acceso a las entradas y salidas de TTL:

| Parámetro | Función                    |
|-----------|----------------------------|
| #86.01    | Entrada digital 0 (DIGIN0) |
| #86.02    | Entrada digital 1 (DIGIN1) |
| #86.03    | Salida digital             |

Las entradas presentan el valor 0 con una lógica elevada (5 V o sin conectar), y 1 con una lógica inferior (0 V).

Cuando se escribe el valor 0, la salida presenta una lógica elevada (5 V).



**Advertencia**

**La salida digital nominal es de 15 miliamperios (receptor/fuente) como máximo.**

**Reduzca la extensión de las conexiones a las entradas y salidas digitales cuanto sea posible (valor recomendado: 0,5 metros máximo). Si se utilizan cables de longitud superior, será preciso el almacenamiento en la memoria intermedia o la interconexión a niveles de lógica diferentes.**

## 8.14 Almacenamiento en la memoria no volátil

El UD70 incluye una función que permite almacenar las regiones P y Q del equipo de registro PLC en la memoria no volátil del UD70. Para iniciar el proceso de almacenamiento, puede ajustar el parámetro #17.19 en 1.

Este parámetro recupera inmediatamente el valor cero después del ajuste. Asimismo, el UD70 se restablece cuando el parámetro está configurado, lo que causa el reinicio del programa de usuario.

Esta función sirve para almacenar información, como el valor del diámetro de una enrolladora, los contadores en tiempo de ejecución o las variables de un programa. Además, al utilizar el controlador de posición incorporado, es posible almacenar todos los parámetros de configuración (prefijo \_Q) y los registros de posición. El UD70 pone automáticamente a cero todos los registros cuando se restablece la alimentación de CA.

## 8.15 Utilización del puerto RS232 para las comunicaciones de accionamiento a accionamiento

La tarjeta MD29 y el módulo UD70 disponen de un modo en el que es posible establecer un enlace entre el puerto RS232 y otra tarjeta MD29 u otro módulo UD70, así como transmitir variables de 32 bits entre éstos.

El puerto RS232 podrá emplearse en las comunicaciones de accionamiento a accionamiento siempre que otro dispositivo esté utilizando el puerto RS485 (por ejemplo, una interfaz de usuario (MMI)). El protocolo empleado se sirve de la prueba de redundancia cíclica (CRC) para garantizar la integridad de los datos.



### Advertencia

**Debido a que el puerto RS232 presenta características inferiores, el cable de comunicaciones serie que conecta los accionamientos debe tener la menor longitud posible (inferior a 10 metros). En caso necesario, puede utilizar los convertidores RS232-RS485 para prolongar el cable.**

**No se admite el uso de comunicaciones serie multiterminal. El enlace RS232 accionamiento a accionamiento se ha diseñado de forma exclusiva para la conexión a una tarjeta MD29 o un módulo UD70 externo.**

**En este enlace no se utiliza protocolo de intercambio (es decir, no se enviará ningún mensaje error si la conexión se interrumpe, por ejemplo).**

## Conexiones

Las conexiones con los conectores RS232 que se requieren son las siguientes:

| Unidad de origen | Unidad de destino |
|------------------|-------------------|
| 2                | 3                 |
| 3                | 2                 |
| 5                | 5                 |

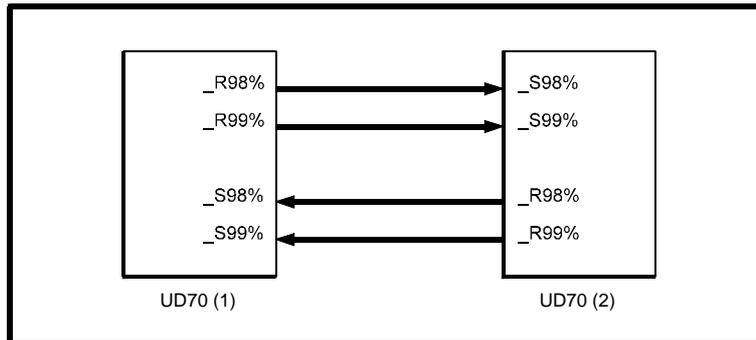
Para activar el modo RS232, ajuste los siguientes parámetros de configuración en 1:

- Desactivar comunicaciones de juego de herramientas
- Comunicaciones accionamiento a accionamiento

Consulte los parámetros de configuración del UD70 en el Capítulo 10, Parámetros.

## Parámetros de intercambio de datos

Los datos que se quieran transmitir tendrán que incluirse en los parámetros internos especiales `_R98%` y `_R99%`. Los datos procedentes de la unidad remota también se incluirán en los parámetros `_S98%` y `_S99%`.



*Parámetros de origen y destino para la transferencia de datos*



---

## 9 Diagnósticos

---

En este capítulo se describe lo siguiente:

- Errores en tiempo de ejecución y códigos de desconexión
- Errores y advertencias del compilador
- Gestión avanzada de errores

### 9.1 Errores en tiempo de ejecución

Error que ocurre en una operación específica del UD70. Puede deberse a un error en el programa DPL (como un intento de escribir en un parámetro que no existe o de dividir un valor entre cero) o a un error en una operación automática, como la pérdida de comunicaciones con una caja de E/S.

Algunas desconexiones en tiempo de ejecución pueden evitarse (por ejemplo, cuando el valor introducido en el parámetro es superior al rango admitido). Cuando se genera un error de este tipo, se realizan las acciones siguientes:

- Se interrumpe la ejecución de todas las tareas DPL.
- Si el parámetro de activación de la desconexión general en tiempo de ejecución (#17.14) está ajustado en 1 y el error no afecta a las comunicaciones serie, el accionamiento se desconecta. La desconexión hace que el accionamiento desactive inmediatamente las salidas de potencia. Si el error está relacionado con la caja de E/S, el accionamiento se desconecta cuando el parámetro de activación de desconexión de la caja de E/S está ajustado en 1.
- Si el programa DPL presenta una tarea ERROR, comienza a ejecutarse (para obtener información detallada, consulte la sección Gestión avanzada de errores de este capítulo).

## 9.2 Códigos de desconexión en tiempo de ejecución

Cuando el accionamiento se desconecta, la pantalla muestra el código **tr** seguido de un número de dos dígitos (por ejemplo, **tr 41**). El número del error puede determinarse en una tarea ERROR mediante la lectura del parámetro virtual #88.01.

Los códigos de desconexión que existen son los siguientes:

| Número de error | Descripción   | Acción                                  |
|-----------------|---|---|
| 40              | Error no identificado   | Desconexión siempre                     |
| 41              | El parámetro no existe  | Desconexión si #17.14 = 1               |
| 42              | Fallo de escritura en parámetro: el parámetro es de sólo lectura                          | Desconexión si #17.14 = 1               |
| 43              | Fallo de lectura de parámetro: el parámetro es de sólo escritura                          | Desconexión si #17.14 = 1               |
| 44              | Fallo de escritura en parámetro: valor del parámetro por encima del rango                 | Desconexión si #17.14 = 1 y #17.17 = 1  |
| 45              | Fallo de acceso a parámetro virtual: IOLINK no se está ejecutando                         | Desconexión si #17.14 = 1 y #17.15 = 1  |
| 46 ~ 48         | Error interno   | Desconexión siempre                     |
| 49              | Cargado sistema incorrecto  | Desconexión siempre                     |
| 50              | Error matemático en el programa; por ejemplo, división entre cero, valor por exceso, etc. | Desconexión si #17.14 = 1               |
| 51              | Índice de matriz DPL fuera de rango   | Desconexión si #17.14 = 1               |
| 52              | Desconexión generada por usuario mediante palabra de control                              | Desconexión siempre                     |
| 53              | Programa DPL incompatible   | Desconexión siempre                     |
| 54              | Sobrecarga DPL: expirado tiempo de ejecución de una tarea                                 | Desconexión si #17.14 = 1               |
| 55              | Desconexión RS485 (modo 3, modo 4, etc)   | Desconexión si #17.14 = 1 y #17.15 = 1* |
| 56              | Módulo de opciones y archivo de sistema incompatibles                                     | Desconexión siempre                     |
| 57              | Llamada a sistema operativo no permitida  | Desconexión siempre                     |
| 58 - 59         | Error interno   | Desconexión si #17.14 = 1               |
| 60 - 69         | Desconexiones generadas por opción de comunicaciones a alta velocidad                     | Desconexión si #17.14 = 1               |
| Prc2            | Desconexión de controlador de secuencia (consulte el comando WDOG, Capítulo 7)            | Desconexión si #17.18 = 1               |

(\* La desconexión 55 sólo ocurre cuando la caja de E/S está conectada y funcionando y se produce un fallo de comunicaciones serie en el modo 3 o 4.)

## 9.3 Mensajes de error del compilador

En estas secciones se incluye una lista con todos los errores que pueden generarse al compilar un programa DPL.

ERROR: Argument x must be an integer  
(El argumento x debe ser un entero)

El argumento se ha introducido como un número de coma flotante en lugar de un entero.

**Ejemplo** ANSW RITE (11,"SP",50,31)

donde:

11 = Argumento 1  
SP = Argumento 2  
50 = Argumento 3  
31 = Argumento 4

El argumento 4 debe ser un entero.

ERROR: Array must be dimensioned  
(Es preciso ajustar el tamaño de la matriz)

Antes de que sea posible hacer referencia a una matriz en un enunciado DIM, es preciso definirla. Esto permite asignar un tamaño a la matriz antes de que sea necesario (consulte la instrucción DIM en el Capítulo 7, Referencia).

ERROR: **CALL** can call only in-built functions or user tasks  
(CALL sólo funciona en funciones incorporadas o en tareas de usuario)

La instrucción CALL sólo sirve para llamar a una aplicación estándar o una tarea definida por el usuario (la instrucción CALL no permite llamar a una etiqueta, por ejemplo). (Consulte la instrucción CALL en el Capítulo 7, Referencia).

ERROR: **DELAY** can be used only in the **INITIAL** and **BACKGROUND** (DELAY sólo puede utilizarse en las tareas INITIAL y BACKGROUND)

Por su condición, las tareas ENCODER y CLOCK no admiten la instrucción DELAY. Un programa DPL sólo puede interumpirse mientras se ejecuta en las tareas INITIAL o BACKGROUND (consulte la instrucción DELAY en el Capítulo 7, Referencia).

ERROR: **DIM** must have an integer number of elements  
(DIM debe contener un número de elementos entero)

Para definir el tamaño de una matriz debe utilizarse un número entero.

ERROR: Empty Tasks are not permitted - remove the Task and recompile (No se admiten tareas sin contenido. Elimine la tarea y vuelva a compilar)

Se ha definido una tarea que no incluye instrucciones entre llaves. Elimine la tarea y las llaves.

ERROR: Expression is already a float - remove **FLOAT** instruction (la expresión ya es de coma flotante. Borre la instrucción FLOAT)

Se ha asignado una instrucción FLOAT a una expresión definida con o variable de coma flotante. Borre la instrucción FLOAT.

ERROR: Expression is already an Integer variable - remove **INT** instruction (La expresión ya es una variable entera. Borre la instrucción INT)

Se ha asignado una instrucción INT a una expresión definida con o variable entera. Borre la instrucción INT.

ERROR: Expression too complex - break into parts (Expresión demasiado compleja. Descompóngala en segmentos)

ERROR: Invalid Drive type (Tipo de accionamiento no válido)

ERROR: Label duplicated (Etiqueta duplicada)

Se ha introducido una etiqueta con el mismo nombre varias veces. Compruebe si está duplicada en otras tareas.

ERROR: Label is in another task (La etiqueta está incluida en otra tarea)

La instrucción GOTO está incluida en una tarea distinta de la tarea en la que se ha definido la etiqueta. La etiqueta y la instrucción GOTO deben estar incluidas en la misma tarea.

ERROR: Label not found (Etiqueta no encontrada)

No se ha definido una etiqueta. Defínala.

ERROR: Operators only allowed on integer arguments (Los operadores sólo pueden utilizarse en argumentos de número entero)

ERROR: Maximum bit-field size is 32 (El tamaño máximo del campo de bits es 32)

Se ha utilizado el operador de inversión del campo de bits y el campo especificado contiene más de 32 bits. Vuelva a especificar el tamaño del campo.

ERROR: Syntax error (Error de sintaxis)

ERROR: Variable has not been initialized (La variable no se ha inicializado)

Especifique asignar un valor inicial a las variables antes de que sea posible utilizarlas como argumento. Esta operación suele realizarse en la tarea INITIAL. Recuerde que en los nombres de variable se distingue entre mayúscula y minúscula.

ERROR: Variable is not an array (La variable no es una matriz)

## Mensajes de advertencia

---

ADVERTENCIA: Title will be truncated to 64 characters  
(El título se truncará en el carácter 64)

ADVERTENCIA: Version will be truncated to 8 characters  
(La versión se truncará en el carácter 8)

ADVERTENCIA: Possible loss of accuracy in assignment  
(Posible pérdida de precisión en la asignación)

Esta advertencia indica que se ha asignado un número de coma flotante a un parámetro entero. Por ejemplo:

**#1.14 = 3.142**

Dado que sólo es posible utilizar variables enteras para configurar #1.14, #1.14 será = 3.0

## 9.4 Gestión avanzada de errores

Aunque los errores que ocurren durante la ejecución del programa suelen deberse a errores de programación, a veces pueden tener su origen en factores externos. Por ejemplo, si los datos procedentes de una caja de E/S se pierden a causa de la ruptura del cable, puede generarse un error que indique la interrupción de las comunicaciones serie. Normalmente, el UD70 detiene todas las tareas y puede desconectar el accionamiento de forma opcional.

Para evitar que esto ocurra, puede utilizar la tarea ERROR. La secuencia de eventos es la siguiente:

- 1 Todas las tareas se suspenden.
- 2 El accionamiento se desconecta (si la desconexión está activada). Consulte los parámetros de activación de la desconexión en los parámetros de configuración del UD70 (Capítulo 10, Parámetros).
- 3 El número de error aparece en el parámetro #88.01 del UD70.
- 4 La tarea ERROR se ejecuta. Las instrucciones contenidas en la tarea ERROR pueden determinar la causa del error en tiempo de ejecución y tomar las medidas necesarias, como detener el accionamiento de forma controlada.

Es posible utilizar todas las instrucciones DPL en la tarea ERROR.

La causa del error puede determinarse mediante la lectura del parámetro virtual #88.01. Esto permite obtener el código de error adecuado que se ha definido en los códigos de desconexión en tiempo de ejecución al principio de este capítulo.

Si el parámetro de activación de la desconexión general en tiempo de ejecución (#1714) no se ajusta en 1, el accionamiento no se desconecta automáticamente. Cuando se requiere una desconexión del accionamiento, el código de error debe escribirse en el parámetro de código de desconexión adecuado (según se indica en los códigos de desconexión del principio de este capítulo).

Para restablecer el UD70 y reiniciar el programa DPL, ajuste el parámetro #8801 en 1070.

Cuando ocurre un error en tiempo de ejecución, todas las tareas DPL dejarán de ejecutarse.

Consulte un ejemplo de la tarea ERROR en la ayuda en línea.

## 10 Parámetros

### 10.1 Parámetros de configuración del UD70

Los parámetros de configuración sólo se aplican después de reiniciar el UD70. El dispositivo se reinicia en los casos siguiente:

- Conexión de la alimentación de CA al accionamiento
- Ejecución de una instrucción REINIT en un programa DPL
- Selección de **Reset Target** en el menú Run del juego de herramientas DPL
- Escritura del valor 1070 en el parámetro #88.01

La tabla siguiente incluye todos los parámetros de configuración del UD70.

| Función  | Parámetro | Valor mín | Valor máx | Predeterminado |
|--|-----------|-----------|-----------|----------------|
| Código de Módulo de opciones grande  | #17.01    | Ninguno   | Ninguno   | 1 = UD70       |
| Versión de software del sistema operativo del Módulo de opciones grande  | #17.02    | Ninguno   | Ninguno   | Ninguno        |
| Número de línea DPL en la que ocurre la desconexión  | #17.03    |           |           |                |
| Sólo modo de depuración  |           |           |           |                |
| Recursos disponibles (%)   | #17.04    | 0         | 100       | Ninguno        |
| #17.04 indica la cantidad de recursos del procesador disponibles mediante la presentación del porcentaje de tiempo que se ejecutará la tarea BACKGROUND en un intervalo de 128 ms.   |           |           |           |                |
| Dirección serie ANSI   | #17.05    | 11        | 99        | 11             |
| Este parámetro define la dirección asignada a las comunicaciones serie. Rango: 11 a 99.  |           |           |           |                |
| Modo RS485   | #17.06    | 1         | 12        | 1              |
| Este parámetro establece el modo que se va a utilizar en las comunicaciones serie. (Consulte abajo).   |           |           |           |                |
| Velocidad en baudios RS485 (modos 1, 5, 6 y 7)   | #17.07    | 300       | 38400     | 4800           |
| Este parámetro define la velocidad en baudios: 24 representa 2400... 192 representa 19200, etc. El valor máximo es 38400 y se ajusta mediante un código de 38.   |           |           |           |                |
| Apuntador de parámetro RS485 1   | #17.08    | 01.00     | 19.50     | 0.00           |
| Define el parámetro de origen de los datos que se van a transmitir a un accionamiento remoto en las comunicaciones de los modos 2 y 4, así como el parámetro de destino de los datos que se reciban mientras se utilice el modo 3. |           |           |           |                |
| Apuntador de parámetro RS485 2   | #17.09    | 01.00     | 19.50     | 0.00           |
| Define el parámetro al que se van a enviar los datos procedentes de un accionamiento remoto cuando se utilicen las comunicaciones serie de modo 4.   |           |           |           |                |

| <b>Función</b>  | <b>Parámetro</b> | <b>Valor<br/>mín</b> | <b>Valor máx</b> | <b>Predeter<br/>minado</b> |
|---|------------------|----------------------|------------------|----------------------------|
| Factor de escala RS485  | #17.10           | 0.000                | 4.000            | 1.000                      |
| Define el factor de escala utilizado en las comunicaciones serie de modo 3.   |                  |                      |                  |                            |
| Base de tiempos de la sección de reloj (ms)   | #17.11           | 5                    | 200              | 10                         |
| Define el periodo de tiempo de reloj, expresado en milisegundos, durante el cual se va a ejecutar la tarea CLOCK de un programa DPL. Rango: 5 a 200 ms  |                  |                      |                  |                            |
| Configuración de controlador de posición  | #17.12           | 0                    | 2                | 0                          |
| Define el periodo de tiempo de reloj, expresado en milisegundos, durante el cual se va a ejecutar la tarea CLOCK de un programa DPL. El comando REINIT no lee este parámetro.   |                  |                      |                  |                            |
| Modo de autoejecución   | #17.13           | 0                    | 1                | 1                          |
| Si se ajusta en 1, #17.13 permite iniciar automáticamente un programa DPL al reiniciar el UD70 o al conectar la alimentación de CA. Si #17.13 se ajusta en 0, es preciso enviar un comando desde el software del juego de herramientas del UD70 para iniciar el programa DPL.<br>Nota: el valor predeterminado es 0 en accionamientos Unidrive con software V2.xx.xx. |                  |                      |                  |                            |
| Activación de desconexión general en tiempo de ejecución  | #17.14           | 0                    | 1                | 0                          |
| Si #17.14 se ajusta en 1, cualquier desconexión en tiempo de ejecución causará la desconexión del accionamiento, a excepción de las desconexiones del puerto RS485 (consulte #17.15). Si #17.14 se ajusta en 0, el accionamiento sólo se desconectará a causa de una desconexión en tiempo de ejecución grave.  |                  |                      |                  |                            |
| Modo de desconexión RS485   | #17.15           | 0                    | 1                | 0                          |
| Si ocurre una desconexión del puerto RS485 (a causa de la interrupción de las comunicaciones en el modo 3, 4 o 10, por ejemplo) y los parámetros #17.14 y #17.15 se han ajustado en 1, el accionamiento se desconecta. Aunque el accionamiento no se desconecte cuando #17.14 o #17.15 se ajuste en 0, todas las tareas se detendrán.                                 |                  |                      |                  |                            |

| <b>Función</b>   | <b>Parámetro</b> | <b>Valor mín</b> | <b>Valor máx</b> | <b>Predeterminado</b> |
|--|------------------|------------------|------------------|-----------------------|
| Desconexión por fallo del enlace de la caja de E/S (modo 10)   | #17.16           | 0                | 1                | 0                     |
| Si #17.16 se ajusta en 1 y el parámetro #17.14 <i>Desconexión general</i> está activado, el accionamiento se desconecta al producirse un fallo del enlace de comunicación entre el UD70 y la caja de E/S en las comunicaciones de modo 10.   |                  |                  |                  |                       |
| Desconexión por introducción de un valor superior al rango el parámetro  | #17.17           | 0                | 1                | 0                     |
| En cada parámetro del accionamiento puede utilizarse un rango de valores finito; por consiguiente, los valores de parámetro que queden fuera del rango podrán causar un fallo en el programa. Si este parámetro se ajusta en 1, se generará un error en tiempo de ejecución cada vez que el valor introducido en el parámetro supere el límite. Mediante el ajuste 0, el UD70 aplica automáticamente un límite para la introducción de valores, y no se genera ningún error. |                  |                  |                  |                       |
| Activar controlador de secuencia   | #17.18           | 0                | 1                | 0                     |
| Si #17.18 se ajusta en 1, el programa DPL necesita ejecutar un comando WDOG cada 200 ms como mínimo; de lo contrario, el accionamiento se desconecta. Para obtener información detallada, consulte el comando WDOG (Capítulo 7).   |                  |                  |                  |                       |
| Almacenamiento de menú 20 en no volátil  | #17.19           | 0                | 1                | 0                     |
| El menú 20 y los parámetros PLC se almacenan en la memoria no volátil del UD70, en lugar de en el accionamiento. Si este parámetro se ajusta en 1, el UD70 almacena los valores actuales. Los parámetros se restablecen a 0 cuando concluye la secuencia de almacenamiento.  |                  |                  |                  |                       |
| Activar almacenamiento de menú 20 al apagar  | #17.20           | 0                | 1                | 0                     |
| Si este parámetro se ajusta en 1, el UD70 almacena los valores del menú 20 y de los parámetros PLC en la memoria no volátil cuando se desconecta la alimentación del accionamiento.  |                  |                  |                  |                       |
| Desactivar comunicaciones de juego de herramientas   | #17.21           | 0                | 1                | 0                     |
| Mediante el ajuste de #17.21 en 1 se desactivan los protocolos de comunicación. Esto se utiliza durante el proceso de depuración y mientras se emplea el juego de herramientas DPL. Con el ajuste de #17.21 en 1 también se configura el puerto RS232 en el modo ASCII normal.   |                  |                  |                  |                       |

## 10.2 Parámetros virtuales

Parámetros especiales que no forman parte del conjunto de parámetros estándar del accionamiento. Sólo existen en el UD70 y se utilizan para lo siguiente:

- Acceder a valores que no aparecen como parámetros estándar del accionamiento (ejemplo, los contadores del codificador)
- Acceder a los parámetros del accionamiento a mayor velocidad
- Acceder a los parámetros del accionamiento aplicando una resolución mayor
- Acceder a los parámetros del UD70, como los parámetros de control de la unidad de temporización/contador, los datos de la caja de E/S, etc

Para acceder a los parámetros virtuales puede utilizarse un programa DPL o las comunicaciones serie. Los menús virtuales se describen en el orden siguiente:

90, 91, 70 ~ 73, 85, 86, 88, 80

### Menú 90

#### Parámetros generales

Los códigos de frecuencia de actualización L2 y L3 se describen al final de esta tabla. La frecuencia de actualización no se indica en los casos en los que no es aplicable.

| Parámetro | Descripción   |
|-----------|---|
| #90.01    | Posición de resolver/codificador principal (RO)   |
| #90.03    | Posición de codificador auxiliar (RO)   |
| #90.05    | Sólo se utiliza si se ha instalado un módulo de opciones de codificador SIN-COS en el accionamiento y el software del accionamiento pertenece a la versión 3 u otra posterior<br>Número de revoluciones de un codificador SIN-COS (RO)<br>Frecuencia de actualización: L2   |
| #90.06    | Sólo se utiliza si se ha instalado un módulo de opciones de codificador SIN-COS en el accionamiento y el software del accionamiento pertenece a la versión 3 u otra posterior<br>Posición de un codificador SIN-COS (RO)<br>#90.6 indica los ocho bits menos significativos solamente.<br>Frecuencia de actualización: L2 |
| #90.10    | Identificador de modo de accionamiento (RO)<br>Los códigos son los siguientes:<br>16 Bucle abierto<br>17 Vectorial de bucle cerrado<br>18 Servo   |
| #90.11    | Estado de accionamiento y palabra de control<br>#90.11 indica el valor del parámetro de estado del accionamiento (#10.40). La palabra de control puede introducirse en #90.11 como se indica en la tabla siguiente.   |

### Palabra de control

| Bit | Controles...   |
|-----|--|
| b15 | Ajuste b15 en 0 para que b6 no modifique #1.46<br>Ajuste b15 en 1 para que b6 controle #1.46 |
| b14 | Ajuste b14 en 0 para que b5 no modifique #1.45<br>Ajuste b14 en 1 para que b5 controle #1.45 |
| b13 | #18.33   |
| b12 | Ajuste b12 en 0 para que b3 no modifique #6.32<br>Ajuste b12 en 1 para que b3 controle #6.32 |
| b11 | Ajuste b11 en 0 para que b2 no modifique #6.31<br>Ajuste b11 en 1 para que b2 controle #6.31 |
| b10 | Ajuste b10 en 0 para que b1 no modifique #6.30<br>Ajuste b10 en 1 para que b1 controle #6.30 |
| b9  | Ajuste b9 en 0 para que b0 no modifique #6.15<br>Ajuste b9 en 1 para que b0 controle #6.15   |
| b8  | #18.32   |
| b7  | #18.31   |
| b6  | #1.46  |
| b5  | #1.45  |
| b4  | Desconexión  |
| b3  | #6.32  |
| b2  | #6.31  |
| b1  | #6.30  |
| b0  | #6.15  |

### Palabra de estado

| Bit | Parámetro     |
|-----|---------------|
| b15 | No se utiliza |
| b14 | #10.15        |
| b13 | #10.14        |
| b12 | #10.13        |
| b11 | #10.12        |
| b10 | #10.11        |
| b9  | #10.10        |
| b8  | #10.09        |
| b7  | #10.08        |
| b6  | #10.07        |
| b5  | #10.06        |
| b4  | #10.05        |
| b3  | #10.04        |
| b2  | #10.03        |
| b1  | #10.02        |
| b0  | #10.01        |

## Menú 91 Parámetros virtuales específicos del accionamiento

### Parámetros de acceso rápido

Los códigos de frecuencia de actualización L2 y L3 se describen al final de esta tabla. La frecuencia de actualización no se indica en los casos en los que no es aplicable.

| Parámetro | Descripción   |     |         |   |   |   |   |   |  |   |   |
|-----------|---|-----|---------|---|---|---|---|---|--|---|---|
| #91.01    | <p>Activación de actualización rápida<br/>Parámetro con mapa de bits en el que los bits controlan lo siguiente:</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Función</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Ajuste en 1 para activar todos los parámetros rápidos</td> </tr> <tr> <td>1</td> <td>Ajuste en 1 para activar el punto de referencia de frecuencia rápida (#91.02)</td> </tr> <tr> <td>2</td> <td>Ajuste en 1 para activar la referencia de velocidad fija rápida (#91.03)</td> </tr> <tr> <td>3</td> <td>Ajuste en 1 para activar el punto de referencia de par de torsión rápido (#91.04)</td> </tr> </tbody> </table> | Bit | Función | 0 | Ajuste en 1 para activar todos los parámetros rápidos | 1 | Ajuste en 1 para activar el punto de referencia de frecuencia rápida (#91.02) | 2 | Ajuste en 1 para activar la referencia de velocidad fija rápida (#91.03) | 3 | Ajuste en 1 para activar el punto de referencia de par de torsión rápido (#91.04) |
| Bit       | Función   |     |         |   |   |   |   |   |  |   |   |
| 0         | Ajuste en 1 para activar todos los parámetros rápidos   |     |         |   |   |   |   |   |  |   |   |
| 1         | Ajuste en 1 para activar el punto de referencia de frecuencia rápida (#91.02)   |     |         |   |   |   |   |   |  |   |   |
| 2         | Ajuste en 1 para activar la referencia de velocidad fija rápida (#91.03)  |     |         |   |   |   |   |   |  |   |   |
| 3         | Ajuste en 1 para activar el punto de referencia de par de torsión rápido (#91.04)   |     |         |   |   |   |   |   |  |   |   |
| #91.02    | <p>Punto de referencia de velocidad<br/>El valor de #91.02 es un número de 16 bits formado por 15 bits más un bit de signo. El valor a plena escala permite obtener la velocidad proporcionada por el parámetro #91.05 <i>Velocidad a plena escala</i>. Si el accionamiento se utiliza en bucle abierto, la resolución se obtiene a partir de los 15 bits. Si el accionamiento se utiliza en bucle cerrado, la resolución depende del codificador.<br/>Frecuencias de actualización...<br/>bucle abierto: L3<br/>bucle cerrado: L2</p>  |     |         |   |   |   |   |   |  |   |   |
| #91.03    | <p>Referencia de velocidad fija<br/>Utilice #91.03 solamente cuando el accionamiento funcione en bucle cerrado. En el Unidrive, #91.03 se asigna como referencia de velocidad fija (#3.19).<br/>Unidades: 0,01 RPM<br/>Frecuencia de actualización: L2</p>  |     |         |   |   |   |   |   |  |   |   |
| #91.04    | <p>Punto de referencia de par de torsión<br/>#91.04 se asigna a la referencia de par de torsión (#4.08).<br/>Unidades: 0,1%<br/>Frecuencia de actualización: L2</p>   |     |         |   |   |   |   |   |  |   |   |
| #91.05    | <p>Velocidad a plena escala<br/>Utilice #91.05 solamente cuando el accionamiento funcione en bucle cerrado a fin de ajustar la velocidad a plena escala. Esto permite utilizar la resolución de 16 bits del parámetro #91.02 <i>Punto de referencia de velocidad</i> siempre que se desee.<br/>Unidades: 1 RPM</p>  |     |         |   |   |   |   |   |  |   |   |

| <b>Parámetro</b> | <b>Descripción</b>  |
|------------------|---|
| #91.06           | Realimentación de velocidad (#3.2)<br>Utilice #91.06 solamente cuando el accionamiento funcione en bucle cerrado.<br>Unidades: 0,01 RPM (ejemplo, 123 representa 1,23 RPM)<br>Frecuencia de actualización: L3 |
| #91.07           | Realimentación de corriente (#4.02)<br>Unidades: 0,01 A (ejemplo, 100 representa 1,00 A)<br>Frecuencia de actualización: L3   |
| #91.08           | Valor de entrada analógica 1<br>Se aplica a escala para que 4000 represente la señal [plena escala] en la entrada.<br>Frecuencia de actualización: L3   |
| #91.09           | Valor de entrada analógica 2<br>Se aplica a escala para que 1000 represente la señal [plena escala] en la entrada.<br>Frecuencia de actualización: L3   |
| #91.10           | Valor de entrada analógica 3<br>Se aplica a escala para que 1000 represente la señal [plena escala] en la entrada.<br>Frecuencia de actualización: L3   |

### Frecuencias de actualización

| <b>Código</b> | <b>Frecuencia de conmutación PWM</b> |                   |
|---------------|--------------------------------------|-------------------|
|               | <b>3, 6, 12 kHz</b>                  | <b>4,5, 9 kHz</b> |
| L2            | 345 s                                | 460 s             |
| L3            | 1,38 ms                              | 1,84 ms           |

### Menús 70 a 73

Estos menús están relacionados con el equipo de registro PLC, según se indica:

|        |      |
|--------|------|
| #70.xx | _Px% |
| #71.xx | _Qx% |
| #72.xx | _Rx% |
| #73.xx | _Sx% |

**x** es un valor entre 0 y 99.

## Menú 85

### Parámetros del temporizador/contador

---

Consulte la sección Unidad de temporización/contador en el Capítulo 8, Funciones.

| Parámetro | Función                                   |
|-----------|---|
| #85.01    | Palabra de control                        |
| #85.02    | Palabra de estado                         |
| #85.03    | Valor de temporizador/contador            |
| #85.04    | Valor de recarga de temporizador/contador |
| #85.05    | Valor de temporizador/contador en modo 2  |

## Menú 86

### Parámetros de E/S digitales

---

Para obtener información sobre la conexión, consulte la sección dedica a los puertos de E/S digitales en el Capítulo 8, Funciones.

| Parámetro | Función                    |
|-----------|----------------------------|
| #86.01    | Entrada digital 0 (DIGIN0) |
| #86.02    | Entrada digital 1 (DIGIN1) |
| #86.03    | Salida digital             |

La entrada incluirá el valor 1 si no está conectada o presenta una lógica elevada (+5 V), Frecuencia de entrada máxima = 1MHz. El ajuste de #86.03 en 0 permite cambiar la lógica de la salida a una lógica inferior (0 V).



**Advertencia**

**La salida digital nominal es de 15 miliamperios (receptor/fuente) como máximo.**

**Reduzca la extensión de las conexiones a las entradas y salidas digitales cuanto sea posible (valor recomendado: 0,5 metros máximo). Si se utilizan cables de longitud superior, será preciso el almacenamiento en la memoria intermedia o la interconexión a niveles de lógica diferentes.**

**Menú 88**  
**Parámetros de estado**

| Parámetro | Función  |
|-----------|--|
| #88.01    | <p>READ (lectura) Código de error en tiempo de ejecución. Sólo es válido en la tarea ERROR.</p> <p>WRITE (escritura) La introducción de un valor 1070 restablece el UD70.</p> <p>Consulte la sección Gestión avanzada de errores en el Capítulo 9, Diagnósticos.</p> |

**Menú 80**  
**Parámetros de la caja de E/S**

| N° parámetro | Menú 80 ENT analógica | Menú 81 SAL analógica | Menú 82 ENT digital | Menú 83 SAL digital | Menú 84 Control         |
|--------------|-----------------------|-----------------------|---------------------|---------------------|-------------------------|
| 0            | Ninguno               | Ninguno               | Ninguno             | Ninguno             | Ninguno                 |
| 1            | ADC1 (4000)           | DAC1 (1000)           | bit 1               | bit 1               | 0 = ADC1<br>1 = 4-20 mA |
| 2            | ADC2 (1000)           | DAC2 (1000)           | bit 2               | bit 2               |                         |
| 3            | ADC3 (1000)           | DAC3 (1000)           | bit 3               | bit 3               |                         |
| 4            | ADC4 (1000)           |                       | bit 4               | bit 4               |                         |
| 5            | ADC5 (1000)           |                       | bit 5               | bit 5               |                         |
| ...          |                       |                       |                     |                     |                         |
| 32           |                       |                       | bit 32              | bit 32              |                         |
| 40           |                       |                       | bits 1 - 8          | bits 1 - 8          |                         |
| 41           |                       |                       | bits 9 - 16         | bits 9 - 16         |                         |
| 42           |                       |                       | bits 17 - 24        | bits 17 - 24        |                         |
| 43           |                       |                       | bits 25 - 32        | bits 25 - 32        |                         |
| 44           |                       |                       | bits 1 - 16         | bits 1 - 16         |                         |
| 45           |                       |                       | bits 17 - 32        | bits 17 - 32        |                         |
| 46           |                       |                       | bits 9 - 32         | bits 9 - 32         |                         |

**Nota**

**Los parámetros incluidos en los menús 81 y 83 son sólo de escritura. No es posible visualizarlos mediante el enlace de comunicaciones serie.**

### 10.3 Modos del puerto RS485

Estos modos se configuran mediante el uso del parámetro Modo RS485. Para obtener información detallada sobre los parámetros de origen, destino y escala, consulte los parámetros de configuración del UD70 al principio de este capítulo.

| Modo | Descripción  |
|------|--|
| 1    | RS485 de 4 hilos estándar, protocolo ANSI.   |
| 2    | Modo principal. Protocolo binario que permite transferir continuamente el valor del parámetro definido en #17.08 a otro UD70, MD29 u accionamiento. La escala que se aplica a los datos es 16000.  |
| 3    | Modo secundario. El UD70 recibe la información, aplica una escala e incluye el valor final en el parámetro especificado por #17.08.  |
| 4    | Modo de cascada. El valor del parámetro definido en #17.08 se transmite y los datos recibidos se incluyen en el parámetro definido en #17.09. No se aplica ninguna escala.   |
| 5    | RS485 de 2 hilos, protocolo ANSI.  |
| 6    | Modo de usuario. Este modo desactiva todos los protocolos internos y permite que el usuario utilice el puerto RS485 directamente desde el programa DPL. Normalmente, este modo se utiliza junto con los comandos principales ANSI (ANSIREAD, ANSIWRITE, etc) de DPL. Los protocolos definidos por el usuario también puede utilizarse en DPL con los comandos PUTCHAR y GETCHAR de nivel inferior. La trama de datos de las comunicaciones se organiza como sigue: 1 bit de inicio, 7 bits de datos, paridad par y 1 bit de parada (10 bits en total). |
| 7    | Modo de usuario. Igual que el modo 6, con la diferencia de que la trama de datos de las comunicaciones se organiza como sigue: 1 bit de inicio, 8 bits de datos, paridad par y 1 bit de parada (11 bits en total).   |
| 8    | Modo de usuario. Igual que el modo 6, con la diferencia de que la trama de datos de las comunicaciones se organiza como sigue: 1 bit de inicio, 8 bits de datos, sin paridad y 1 bit de parada (10 bits en total).   |
| 9    | Modo de usuario. Igual que el modo 6, con la diferencia de que la trama de datos de las comunicaciones se organiza como sigue: 1 bit de inicio, 9 bits de datos, sin paridad y 1 bit de parada (11 bits en total).   |
| 10   | Modo de caja de E/S. Este modo permite conectar directamente una sola caja de E/S al UD70 mediante el puerto EIA RS485.  |
| 11   | Modo de usuario. Los datos se desvían a la memoria FIFO del software, lo que permite reducir el retardo de los datos que se envían o reciben. La velocidad en baudios es la seleccionada en el parámetro #17.07. La trama de datos de las comunicaciones se organiza como sigue: 1 bit de inicio, 9 bits de datos, sin paridad y 1 bit de parada (11 bits en total).   |
| 12   | Reservado  |
| 13   | Modbus RTU   |
| 14   | Modbus ASCII   |

## 10.4 Parámetros de uso general

El Unidrive incluye parámetros de uso general que se utilizan con los programas del UD70.

Aunque los parámetros del tipo RO pueden modificarse mediante el UD70, no es posible utilizar el teclado del accionamiento para realizar los cambios. RW indica que el parámetro puede modificarse mediante el teclado.

| N° parámetro de Unidrive | Descripción  | Tipo | Rango |
|--------------------------|--|------|-------|
| #18.01                   | Parámetro de sólo lectura. Se almacena en el accionamiento al desconectar la alimentación. | RO   | 32000 |
| #18.02 ~ #18.10          | Parámetros de sólo lectura.  | RO   | 32000 |
| #18.11 ~ #18.30          | Parámetros de lectura y escritura.   | RW   | 32000 |
| #18.31 ~ #18.50          | Parámetros de bits.  | RW   | 0 o 1 |
| #19.01                   | Parámetro de sólo lectura. Se almacena en el accionamiento al desconectar la alimentación. | RO   | 32000 |
| #19.02 ~ #19.10          | Parámetros de sólo lectura.  | RO   | 32000 |
| #19.11 ~ #19.30          | Parámetros de lectura y escritura.   | RW   | 32000 |
| #19.31 ~ #19.50          | Parámetros de bits.  | RW   | 0 o 1 |
| #20.01 ~ #20.50          | Parámetros almacenados en el UD70.   | RW   | 32000 |

### Nota

**Los parámetros del menú 20 se almacenan en la memoria no volátil del UD70, en lugar del accionamiento. Para almacenar estos parámetros, ajuste #17.19 en 1. Si #17.20 está ajustado en 1, los parámetros también se almacenan al desconectar la alimentación.**