

## PLC Simatic S7200

Control de accionamientos MicroMaster por protocolo USS 13/01

Tanto los variadores de velocidad como la automatización basada en PLCs son en la actualidad tecnología standard no sólo en la industria sino también en las instalaciones de control de equipamientos de bombeo, ventilación y extracción de gases, etc.

Es una necesidad evidente integrar el control del drive con el PLC, donde ambos equipos se complementan tecnológicamente, realizando diferentes tareas de control pero que siempre se relacionan.

Generalmente el PLC atiende el proceso en su totalidad, mientras que el variador de velocidad se dedica al control de una parte de él, por ejemplo, a la velocidad de un accionamiento (y la variable física relativa, por ejemplo, *caudal* si es una bomba o *temperatura* si es un ventilador).

Para que el PLC pueda tener control del drive (que éste efectivamente responda a las demandas del proceso), debe transmitir órdenes al variador vía señales específicas; por ejemplo, si se trata de arrancar, parar, invertir el sentido de giro, resetear un fallo, confirmar el arranque del drive, etc., el PLC transmite o recibe señales digitales; en el caso de tener que enviar un setpoint de velocidad o saber a qué velocidad va el accionamiento, el PLC transmite o recibe señales analógicas.

Es aquí donde las soluciones convencionales demandan del PLC el consumo de canales de entradas y salidas digitales y analógicas para satisfacer este control, lo que implica



un costo no despreciable, en particular, en el caso de los módulos analógicos. Además de esto, la instalación del cableado de las señales mencionadas al drive, que necesita cables apantallados y una técnica para evitar interferencias y fallos, acrecienta aún más el costo de la instalación y también la posibilidad de fallos (todavía más cuando la distancia tendida es significativa o la cantidad de drives es variada).

Aún así, asumiendo estos costos, el nivel de automatización obtenido es limitado, dado que no se dispone de la posibilidad de acceder a parámetros del drive (si no es por su propio display).



Eso lleva a que se pierda la posibilidad de transmitir información de importancia (como la corriente del motor, el torque, las rampas de aceleración o desaceleración, etc.) al PLC, información de la que el drive dispone y que puede ser muy útil en el sistema de automatización.

La solución que Siemens propone para esto está basada en la incorporación del protocolo serial universal de automatización de drives (USS) en todos sus variadores de velocidad. Se trata de una técnica de envío y recepción de toda la información del drive a controladores de mayor jerarquía, mediante un conducto de 2 hilos en técnica RS485.

comunicación se arma sobre un par de cables trenzados y apantallados que requiere como única consideración importante incorporar un conector según RS485 en cada "puerto".

- Reducción del riesgo de fallos, como consecuencia de los dos puntos anteriores.
- Ahorro de los costos de instalación y puesta en marcha, gracias a que la instalación del cable es muy sencilla y segura y el programa sólo demanda una parametrización apropiada.
- Reducción del costo por cada drive adicional instalado, ya que sólo se requiere del tramo adicional de cable y

que se requiere de una placa especial de comunicaciones en cada caso).

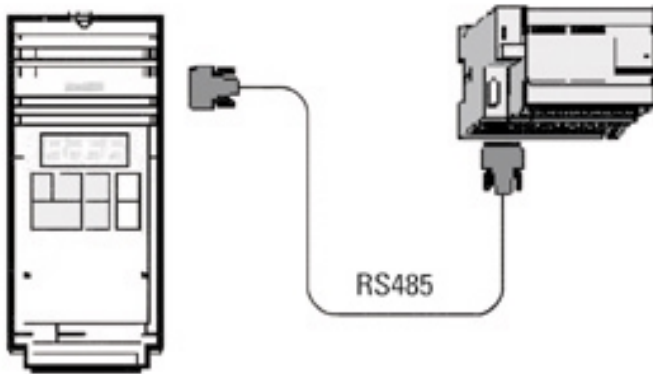
La gran versatilidad del puerto de comunicaciones del micro PLC Simatic S7200 (que permite conmutar su modo de transmitir datos del protocolo PPI básico al modo FREEMPORT, y que posibilita que el usuario transmita datos en formato de caracteres ASCII), ha hecho posible escribir el protocolo de comunicación USS en este modo, de manera que la inversión para implementarlo, más allá del PLC, requiere un costo nulo.

Originalmente este programa se distribuyó en forma libre como un "tip" y más tarde, gracias a la innovación de MicroWin V3.1 de generar subrutinas en editores separados de la rutina principal (o OB1), como un "add-on" instalable con el CD MicroWin ToolBox V1.0, donde las rutinas de comunicación USS se invocan como bloques funcionales simples de parametrizar.

Éstos quedan instalados dentro del árbol de directorios de funciones de MicroWin, bajo la carpeta *Librerías* subdirectorio *Protocolo USS*.

**Paso 1:** Cargar comando de Inicialización de la comunicación USS

Con un doble click sobre el bloque USS\_INIT se carga esta función en el punto del programa deseado. Debe aclararse que, si bien aparecen como bloques, en realidad se trata de subrutinas protegidas que puede verse en las solapas inferiores de la pantalla de edición del programa.



Conexión de un Micromaster a un plc Simatic S7224



La conexión en bus USS

Figura 1: La conexión del MicroMaster al PLC S7200

Este protocolo está tecnológicamente orientado a las funciones imbuidas en el drive, ya sea el envío de comandos y lectura de la respuesta del estado del drive, como así también el acceso a lectura y modificación de los parámetros de él.

Las ventajas de esta solución son muy claras:

- Ahorro de canales de señales del PLC: dado que toda la comunicación se estructura entre un puerto del PLC y el puerto del drive, no es necesario usar señales físicas (ni digitales ni analógicas) para enviar los comandos apropiados al drive.
- Ahorro de cableado: toda la

del conector correspondiente. Los drivers se identifican al PLC vía el bus con un número de esclavo asignable por parámetro.

- Elevado nivel de automatización: al sistema de control ya implementado en el PLC se suma la información que el drive puede recopilar del accionamiento (motor y proceso asociado) y que le da al sistema mayor capacidad de control sin incremento adicional del costo.

**Implementación del protocolo USS en el micro PLC SIMATIC S7200**

El protocolo USS ha sido desarrollado en diversas versiones para ser implementado en toda la variedad de PLCs Simatic (sean S5 o S7, para los

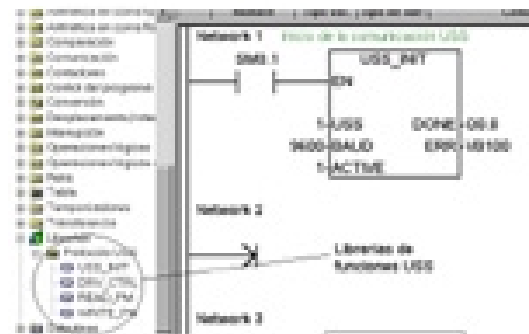


Figura 2: El bloque USS\_INIT parametrizado

Este bloque que inicializa la comunicación con los drives requiere de los siguientes parámetros:

**En** (enable): señal que lo activa. Se requiere de un flanco, en el ejemplo, sólo en el primer ciclo.

**USS:** Habilita al puerto 0 a transmitir en USS (si se pone en 1) o a bloquear este modo y quedar en PPI (ajuste en 0).

**Baud:** el baud rate de transmisión, varía entre 1200 y 19200.

**Active:** codifica las direcciones de los esclavos presentes, puestos en modo binario, pero asignándolo en modo decimal (en el ejemplo, el esclavo tiene la dirección 0).

**Done:** una señal digital que se activa luego de ejecutada esta subrutina.

**ERR:** un byte donde se deposita un código de error tras la ejecución de esta subrutina (por ejemplo, si la dirección del drive parametrizado no se encontró). El listado de códigos de errores se detalla en el manual.

**Paso 2:** reservar una zona de memoria del PLC como memoria temporal de los bloques de control.

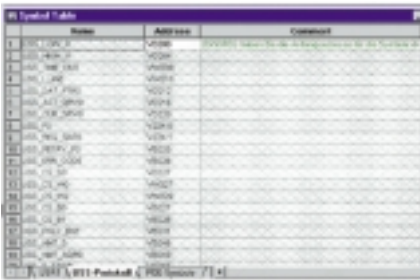


Figura 3: Tabla de símbolos y direcciones reservadas

En la Tabla de Símbolos, bajo la tabla Protocolo USS (CMP1), se debe asignar sólo la primera dirección a partir de la cual se reservará toda una zona de memoria destinada a los bloques de control USS; tipeando por ejemplo, VB100, ésta se llena automáticamente.

**Atención:** debe tenerse la precaución de no usar este espacio de memoria con datos del resto del programa, para no originar un conflicto.

**Paso 3:** Cargar y parametrizar el bloque de control del drive DRV\_CTRL

Esta subrutina es la que efectivamente permitirá el control y monitoreo básico del variador de velocidad. Se carga con un doble click sobre el block DRV\_CTRL.

**Por un lado se codifica los comandos del PLC al drive:**

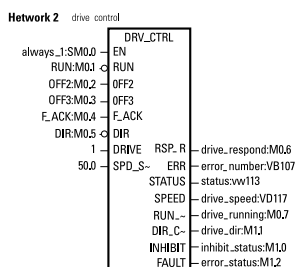


Figura 4: El bloque de control del drive

**En (enable):** siempre debe estar habilitando para que la función se pueda ejecutar.

**Run (marcha):** señal que da marcha al drive.

**OFF2:** señal que genera Paro "OFF2"

**OFF3:** señal que genera Paro "OFF3"

**F\_ACK (fault acknowledgment):** reset de fallos.

**Dir (Dirección):** señal que determina el sentido de giro según esté en estado alto o bajo.

**Drive:** dirección del drive que se quiere comandar.

**SPD\_S:** setpoint de velocidad en porcentaje de la frecuencia prefijada en el drive (en el ej.: 50%).

Por otro lado, se codifica la información que el drive devuelve al PLC:

**RSP\_R:** respuesta que indica que el drive aceptó el comando.

**ERR:** codificación de error acaecido (idem anterior).

**Status:** palabra de estado del drive (ver su manual).

**SPEED:** velocidad actual del motor en % de prefijada.

**RUN:** confirmación de que el drive está en marcha (1) o detenido (0).

**DIR\_C:** sentido de giro (1= derecha).

**INHIBIT:** indica si el drive está habilitado para arrancar (1) o no (0).

**FAULT:** codifica el error en que se puede encontrar el drive (ver manual).

Con estos 3 pasos ya alcanza para implementar el control de un drive; si se deseara controlar más drives, será necesario invocar nuevamente el comando DRV\_CTRL con la dirección Drive correspondiente al que se quiere controlar.

Nótese que cada llamada de estos bloques recarga la memoria del PLC e incrementa el tiempo de ciclo de programa, por lo que para una performance óptima se requiere no controlar más de 5 drives.

Para el reconocimiento de parte del drive que será controlado desde un bus USS por un PLC, debe parametrizarse apropiadamente según el modelo. A continuación se detalla la parametrización que corresponde a un MicroMaster Vector 3ra Generación:

P009=3 Acceso total a parámetros.  
P910=Control remoto total con acceso a parámetros.  
P092=9600 baud rate.  
P091=0 dirección de esclavo = 0

La CPU 226XM (nº de almacén 15155338) con el doble de memoria de la CPU 226 standard se convierte en el equipo ideal para armar programas pesados.



Figura 5: La CPU 226XM con doble puerto y doble memoria.

Aplicación avanzada: lectura y escritura de parámetros.

A diferencia de las funciones ya explicadas, donde la comunicación se mantiene ciclicamente (los datos se transmiten permanentemente entre el drive y el PLC), cada vez que éste lo solicita o está habilitado, la solicitud de lectura y escritura de parámetros se realiza por exclusiva invocación, o sea, acíclicamente.

**Paso 4:** lectura de parámetros.

Con el bloque de inicialización activado tal como se describió, es posible cargar con igual procedimiento el bloque READ\_PM.

La parametrización es muy sencilla, tal como se muestra aquí:

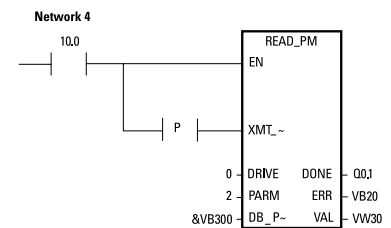


Figura 6: La función de leer parámetros.

**EN:** señal que habilita la función.

**XMT:** señal que permite que el comando se ejecute.

**DRIVE:** dirección del drive que se desea interrogar.

**PARM:** número de parámetro que se desea leer.

**DB\_P:** buffer reservado para codificar los resultados de la función.

**El drive responde con los siguientes comandos:**

**DONE:** señal que marca la ejecución del comando.

**ERR:** byte donde se deposita el código de error (idem anterior).

**VAL:** word donde se deposita el valor del parámetro leído.

**Paso 5:** escritura de parámetros.

Del mismo modo se puede cargar el bloque WRITE\_PM para realizar la asignación o cambio del valor de un parámetro.

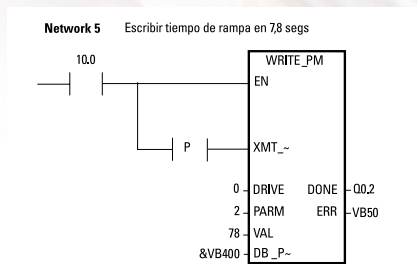


Figura 7: La función de escribir parámetros.

**EN:** señal que habilita la función.

**XMT:** señal que permite que el comando se ejecute.

**DRIVE:** dirección del drive que se desea interrogar.

**PARM:** Nº de parámetro que se desea escribir.

**VAL:** word donde se deposita el valor a escribir en el parámetro elegido; puede ajustarse un valor constante o una zona de memoria donde se carga el valor.

**DB\_P:** buffer reservado para codificar los resultados de la función.

**El drive responde con los siguientes comandos:**

**DONE:** señal que marca la ejecución del comando.

**ERR :** byte donde se deposita el código de error (idem anterior).

**Atención:** Cuando la operación WRITE\_PM se utiliza para actualizar el juego de parámetros depositado en la EEPROM del accionamiento, se deberá tener en cuenta que no se exceda el número máximo de ciclos de escritura en la EEPROM (aprox. 50.000).

En caso de excederse el número máximo de ciclos de escritura, se corromperán los datos almacenados y se perderán los datos subsiguientes.

El número de ciclos de lectura es ilimitado.

Si es necesario escribir con frecuencia en los parámetros del accionamiento, es recomendable ajustar P971 (control de almacenamiento en la EEPROM) a cero.

**Conclusiones:**

La implementación del protocolo USS en el PLC Simatic S7200 reduce sustancialmente las inversiones requeridas para lograr un sistema completo de automatización de uno o varios drives de velocidad variable; la reducción de costos no sólo pasa por el equipamiento no requerido (como prescindir de módulos de señales) sino también por los tiempos de instalación y puesta en marcha y por la disminución de potenciales fallos. A la vez, el resultado final permite lograr un mayor grado de automatización y aprovechamiento del equipamiento instalado, que de otro modo no se aprovecharía.

La programación del control se basa en bloques dedicados, que sólo requieren de una parametrización sencilla sin complejidad alguna.

