Buses de Campo
Optimización del tráfico de información

Francisco J. Orcajo Campillo

La respuesta de un sistema automatizado mediante bus de campo depende de las características intrínsecas del bus implantado, pero también, de la gestión que realicemos en ese bus. La instalación de buses lentos es válida cuando no requerimos tiempos de respuesta reducidos. Sin embargo, si realizamos una gestión adecuada, podemos implementar este tipo de buses –abiertos y de coste reducido- en aplicaciones más exigentes como las del comando de equipos.

Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción.
El objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional bucle de corriente 4-20 mA.
Típicamente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLCs, transductores, actuadotes y sensores. Cada dispositivo de campo incorpora cierta capacidad de proceso, que lo convierte en un dispositivo inteligente, manteniendo siempre un costo bajo. Cada uno de estos elementos será capaz de ejecutar funciones simples de diagnóstico, control o mantenimiento, así como de comunicarse bidireccionalmente a través del bus [1]

Esta definición nos resulta útil como punto de partida para el problema que pretendemos abordar ¿Cómo podemos optimizar los intercambios de información entre el gestor del bus y los diferentes participantes en el mismo? Nuestro problema se agudiza cuando pretendemos obtener respuestas rápidas –p.e. el arranque y parada de diferentes dispositivos- sobre un bus de baja velocidad. Este trabajo muestra diferentes técnicas para garantizar un tiempo de respuesta reducido e independiente del número de participantes en el bus.

La arquitectura Maestro-Esclavo

Una gran parte de los buses de campo industriales responden a la arquitectura Maestro-Esclavo (del inglés Master-Slave). En este caso, el Maestro es el equipo que realiza las peticiones (lectura y escritura de variables) y los diferentes Esclavos integrados en el bus atienden las peticiones que reciben del maestro. Ver Fig.1.

Buses lentos

El problema de la optimización del tráfico de la información es especialmente importante cuando nuestra instalación esté dotada de un "bus lento". Este tipo de buses disponen de velocidades de transmisión menores a los 100 Kbits/s. El más extendido de todos ellos es MODBUS desarrollado por Modicon y que se ha convertido en un estándar de facto puesto que es abierto a quien quiera utilizarlo. En una instalación típica, nos encontraremos con un MODBUS configurado a una velocidad de 19,2 Kbits/s. En este caso, la gestión de una trama de 252 bytes ocupará un tiempo mayor de 100 milisegundos. Este dato, es de una magnitud tal, que de no realizarse una gestión adecuada, se provocará el colapso del bus por la acumulación de peticiones por parte del Maestro que no pueden ser atendidas o como mínimo, obtendremos tiempos de respuesta inaceptables para el comando de equipos.

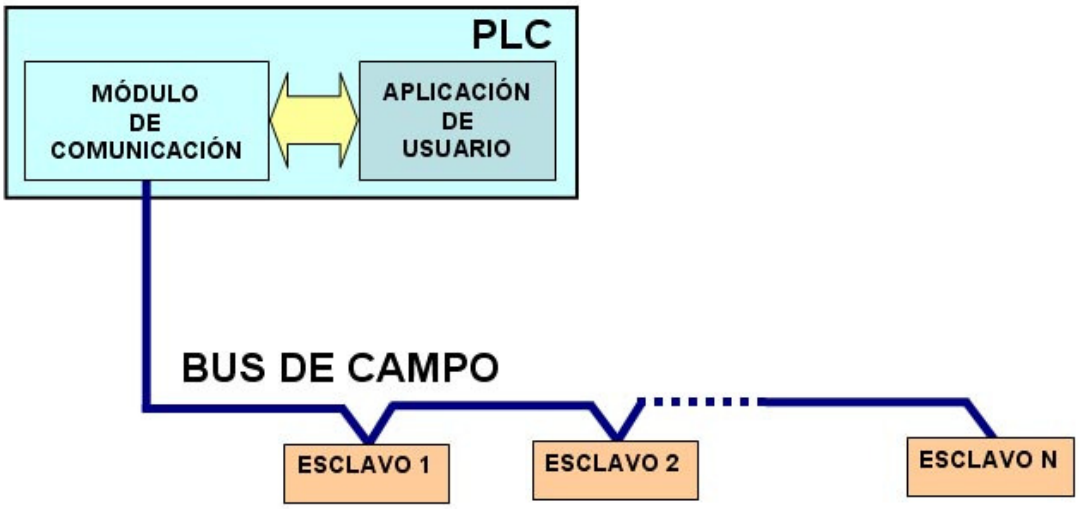

Fig.1. Arquitectura de un Bus de Campo

Como vemos en la figura 1, existe un intercambio de datos entre la aplicación de usuario –el programa del PLC- y el módulo de comunicación que es quien sirve las peticiones de la primera. Cuando se recibe una petición –ya sea de lectura o de escritura- el módulo de comunicación debe gestionarla: lanzarla al bus y esperar una respuesta. En este tipo de buses, no se pueden simultanear diferentes peticiones sobre el medio físico, por lo que debemos garantizar que siempre que nuestro programa realice una petición, el comunicador está libre para recibirla. Ciertamente, el propio Módulo de Comunicación suele disponer de una memoria en la que es capaz de almacenar diferentes peticiones y "gestionarlas". Pero, ¿qué ocurrirá si de forma permanente enviamos más peticiones de las que es capaz de atender? La respuesta es obvia: colapsaremos el módulo.

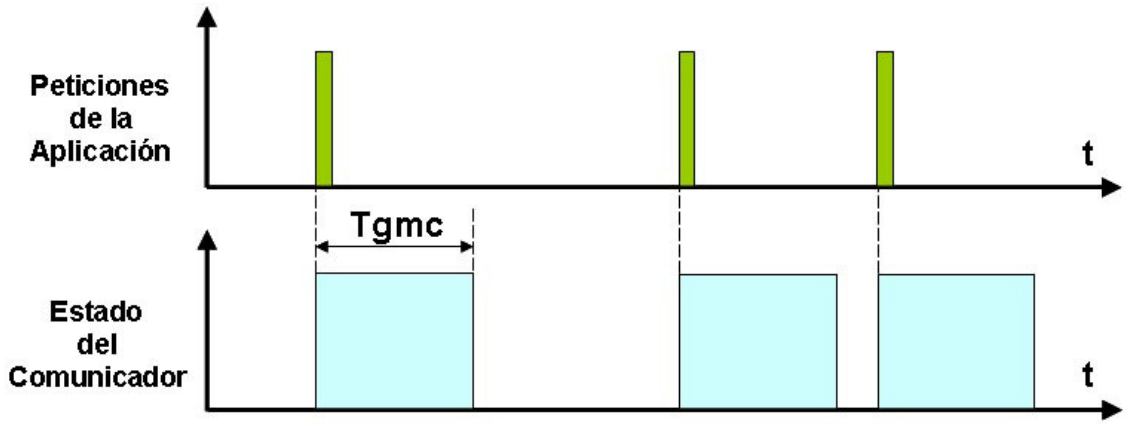

Fig. 2 Cronograma peticiones-respuesta del módulo de comunicaciones

Existe un dato fundamental para que podamos diseñar una aplicación que no sobrecargue la capacidad de nuestro sistema de comunicaciones: es el tiempo de gestión del módulo de comunicaciones, **Tgmc**, en segundos, para dar respuesta a cada una de las peticiones, ver Fig.2.

De lo anterior, la frecuencia máxima de las peticiones, Fmp en Hz, que realicemos al módulo será:

$$Fmp \leq \frac{1}{Tgmc}$$

Si no conocemos el dato del tiempo de gestión de comunicaciones, podemos realizar la aproximación siguiente:

$$Fmp \leq \frac{Ct}{Ks * 8(Ltrama + 12)}$$

En donde

Ct es el caudal de transmisión en bits/s; 19.200 bits/s p.e
Ltrama es la longitud de los registros que queremos leer o escribir, en bytes. Le sumamos el valor **12** que es el tamaño mínimo de la trama de petición de comunicaciones en el caso más desfavorable.
Ks es un factor de seguridad que nos evite el solapamiento de peticiones, Fs>1; se aconseja tomar Fs=1,5

Como vemos, una vez definido el caudal de transmisión, Ct –que será el máximo que nos acepten el conjunto de participantes en el bus- la Frecuencia máxima de las peticiones, Fmp es inversamente proporcional a la longitud de la trama, por lo que aconsejamos realizar peticiones para tramas menores o iguales a los 40 octetos.

Una vez que tenemos el valor de la Frecuencia máxima de las peticiones, podemos generar en nuestra aplicación PLC un reloj interno que sirva para gestionar las peticiones lanzadas por el Maestro. Aquí tendremos, obviamente, que:

$$Freloj \leq Fmp$$

Este reloj interno gestionará nuestras peticiones, pero ¿cuál son estas peticiones? Distinguiremos, como mínimo, tres tipos.

Lectura del estado de los componentes del bus. Como ya hemos dicho antes, diseñaremos estas lecturas con tramas cortas, aunque sobre algún equipo fuese preciso realizar diferentes demandas de lectura.
Escrituras de parámetros. Que en la mayoría de los casos consideraremos “no prioritarias” en el sentido de que no precisan de un tiempo de respuesta crítico.
Comandos. Son órdenes prioritarias sobre los Esclavos que precisan ser ejecutadas en el mínimo tiempo posible, independientemente del número de componentes del sistema y con un tiempo máximo de respuesta garantizado.

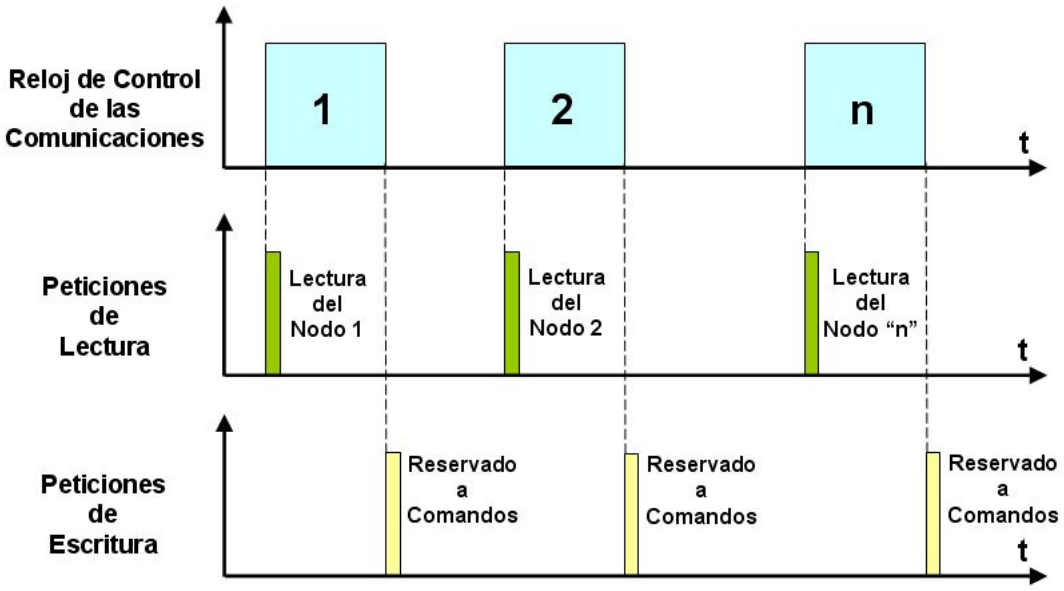

Fig. 3 Cronograma de Peticiones gestionadas por un Reloj de control

Una vez que disponemos de un Reloj para el control de las comunicaciones ($F_{reloj} < F_{mp}$) disponemos de un patrón que nos permite gestionar el lanzamiento de peticiones con la certeza de que no saturaremos nuestro módulo de comunicaciones.

En nuestro caso, proponemos ejecutar las peticiones de lectura de forma cíclica –ver figura 3- de manera que el tiempo de peticiones sobre un mismo nodo (en el caso más simple que será cuando sólo lancemos una sola trama por cada esclavo) tendremos que el tiempo de refresco del estado de cada esclavo, Trslv, será:

$$Trslv = \frac{1}{n(Freloj)}$$

En donde:

n es el número de esclavos en el bus

Freloj es la frecuencia del reloj de control de las peticiones que habíamos determinado previamente.

Con esta solución garantizamos el tiempo de lectura –tiempo de refresco del estado de los esclavos- pero debemos gestionar nuestras peticiones de escritura de manera que no se produzcan de manera simultánea. Si ello no fuese posible, deberíamos realizar la siguiente variante:

Lecturas cíclicas gestionadas por el Reloj de control de las comunicaciones.

Escrituras por excepción. En este caso, ante la petición de una solicitud de escritura se interrumpirá el funcionamiento del Reloj.

Cualquiera de estas soluciones, nos garantiza que las peticiones de escritura se ejecutan con el mínimo retardo dependiendo únicamente de las características intrínsecas del hardware utilizado. Por otro lado, el refresco del estado de los equipos estará en función del hardware y del número de componentes del bus; como no puede ser de otro modo.

Prioridad en las peticiones

Otro de los elementos que debemos jerarquizar es la prioridad en las peticiones. Lo entenderemos más fácilmente con un ejemplo. Si queremos comandar un equipo en bus, necesitaremos saber su estado (marcha, paro o defecto) que estará almacenado en la tabla "Estado". Los posibles defectos de ese equipo estarán en la tabla "Defectos", la intensidad consumida por este equipo, en "Intensidad", etc. En nuestro caso la tabla "Estado" es la información prioritaria y deberemos leerla periódicamente. Sin embargo la tabla "Defectos" la leeremos sólo si "Estado" nos indica que es preciso hacerlo e "Intensidad" únicamente cuando el equipo esté en marcha. De esta manera, reduciremos la longitud de las tramas y realizaremos las peticiones, únicamente, cuando nos aporten información relevante.

De lo anterior, obtenemos una serie de conclusiones que nos pueden resultar muy útiles en el diseño de nuestras arquitecturas de automatización.

Conclusiones

Cuando precisemos tiempos de refresco muy reducidos deberemos reducir el número de esclavos que componen el bus y reducir el tamaño de las tramas de lectura porque el tiempo de refresco depende de estos dos parámetros. Como el número de esclavos de la instalación no es una "variable" habrá que estudiar, en determinadas ocasiones, la necesidad de instalar varios buses, por debajo del número máximo que esté determinado en su topología. Sólo así, obtendremos tiempos de respuesta reducidos en los llamados buses lentos.

Otro de los elementos sobre los que podemos actuar, es la longitud de la trama. En aplicaciones críticas, tendremos que lanzar tramas de lectura reducidas –con los valores fundamentales únicamente- realizando una gestión específica para el resto de lecturas no prioritarias de manera que no sobrecarguemos la capacidad de nuestro bus.

Como vimos al inicio, el tiempo de gestión de una trama de 252 octetos en bus de 19,2 Kbits/s es de unos 100 ms. Si somos capaces de reducir nuestras tramas a unos 20 octetos –siguiendo la estrategia de priorizarlas- obtendremos tiempos de respuesta de 10 ms. Si trabajamos con tramas "priorizadas" gestionadas por un reloj, en el ejemplo que estamos analizando, tendremos que, en un bus con 20 esclavos, realizaremos una lectura cada 200 ms. Este orden de magnitud nos permitirá comandar equipos en bus siempre y cuando el número de participantes sea menor de unos 30 equipos.

[1] Definición tomada de la Wikipedia en http://es.wikipedia.org/wiki/Bus_de_campo

Francisco J. Orcajo Campillo

francisco@orcajocampillo.com

Ingeniero Técnico Industrial por la Escuela Politécnica de Burgos. Actualmente desempeña el ejercicio libre de la profesión diseñando e implementando soluciones programadas con PLCs y SCADAs. Anteriormente trabajó en el Grupo Schneider, en T.J. Bocanegra y en Telettra.

Ha colaborado con la revista Técnica Industrial con artículos técnicos sobre variadores de velocidad y SCADAs.