

## IMPLEMENTACIÓN DEL PROTOCOLO PROFIBUS DE REDES DE COMUNICACIÓN INDUSTRIALES PARA LA AUTOMATIZACIÓN DE SISTEMAS DE MANUFACTURA

**M.Sc. Eduardo Calderón Porras**

*Universidad Nacional Autónoma de Bucaramanga  
Facultad de Ingeniería Mecatrónica.  
Email: edporras@unab.edu.co*

**Abstract:** El trabajo esta orientado al desarrollo de un sistema de comunicación maestro / esclavos, implementando el protocolo de comunicación de redes industriales PROFIBUS para el montaje de sistemas flexibles de manufactura; este sistema consta de un software SCADA para el maestro (Encargado de administrar la red de comunicación), desarrollado con el lenguaje de programación Visual Basic y un sistema para los esclavos (Encargados de realizar las actividades de automatización y control de la planta) basado en el microcontrolador Atmel AT89S8252..

**Keywords:** Redes de comunicación industrial, Fieldbus, PROFIBUS, Mecatrónica, Microcontroladores, Automatización industrial, Sistemas flexibles de manufactura.

### 1. INTRODUCCIÓN

Hacer la conexión física entre computadores usando tarjetas de interfaz y cables no asegura poder compartir información entre los computadores de la red. Para realizar este trabajo, las redes deben tener un juego de reglas o normas bien definidas, a estas normas se les conoce como protocolos, los cuales permiten el intercambio de información entre dos o más dispositivos. Los protocolos se encargan de especificar el formato del mensaje y las reglas necesarias para recibir e interpretar la información que se esta intercambiando. Un protocolo desarrolla muchas funciones, entre estas están: definir una dirección única para cada equipo conectado a la red, determinar como son transmitidos los datos entre los dispositivos y procesar la información cuando es recibida. Algunos protocolos usados para propósito general son: TCP/IP, NetBEUI, UDP, AppleTalk, SNMP y LAT. Para redes industriales se encuentran: DeviceNet, ControlNet, PROFIBUS, Modbus y FOUNDATION Fieldbus [15].

Por definición, una red industrial consiste en la distribución geográfica de instrumentos de medida, actuadores y dispositivos de control que interactúan entre si para llevar a cabo una actividad de automatización y control, [12]. Normalmente a las redes industriales se les conoce como buses de campo o Fieldbus que es como se designará en este proyecto.

Fieldbus (Bus de campo) surgió por un trabajo de estandarización iniciado por la ISA (Instrumentations Society of America), [13]. La forma de comunicación de las redes industriales es digital, transfieren bits de información de forma serial.

Al igual que en las redes de computadores, en las redes industriales también se ha trabajado en la creación de protocolos de comunicación. Se han establecido diversos protocolos muy parecidos entre si, aunque con pequeñas diferencias, [1,4,5,6], entre estos están: DeviceNet, HART, BITBUS de Intel, FIP de AFNOR, MIL-STD-1535 de ANSI, MODBUS de

GOULD INC, Interbus-S de Phoenix, DeviceNet de Allen-Bradley, Ethernet de Intel y Xerox, PROFIBUS de DIN que es en el que se centra este trabajo ya que sé esta teniendo una aceptación muy grande en todo el mundo y hay muchas empresas comprometidas con este producto como Siemens, Bosch, Klockner Moeller, ABB, AEG, Landis&Gir y algunas universidades alemanas, [9].

## 2. PROFIBUS

PROFIBUS (Process fieldbus) es un bus (red, en donde los dispositivos conectados utilizando el mismo medio físico de transmisión) de comunicación que implementa una metodología maestro / esclavo, [9], en donde los maestros son dispositivos con buena capacidad de procesamiento que controlan el acceso al medio de comunicación (hay que tener en cuenta que solo un dispositivo puede usar el medio de comunicación en un momento determinado, ya que todos se comunican por el mismo medio) y tienen a su cargo varios esclavos, que son los dispositivos que están realizando tareas de control (controladores, sensores, actuadores, etc...). El maestro viene realizando las actividades de supervisión y control de sus esclavos, comunicándose con estos periódicamente de la manera establecida por el protocolo, la figura 1 muestra esta estructura, en donde cada maestro tiene asignados unos esclavos.

## 3. RED PROFIBUS MAESTRO / ES CLAVO

El trabajo consiste en el desarrollo de un sistema maestro / esclavo basado en PROFIBUS, ver figura 1. El maestro es un sistema SCADA hecho con el lenguaje de programación Visual Basic 6; el maestro es el administrador de la red y tendrá las siguientes características:

1. Tendrá a su cargo esclavos que son los que realizan las actividades de automatización y control.
2. Es responsable de la inicialización y configuración de la red
3. Designará quien puede utilizar el medio físico en un momento determinado.
4. Controlará la transmisión de datos (inicio, terminación, control)
5. Trabaja con dispositivos virtuales, que consiste en la representación virtual en el sistema de cada uno de los esclavos, en donde el usuarios podrán leer y modificar los parámetros de estos sin necesidad de acudir directamente al esclavo.
6. Contará con una bases de datos hecha con Access 2000 de Microsoft, esta bases de datos

servirá de apoyo para la administración y control del sistema, tendrá información de: esclavos (índice, nombre, tipo de control, variables u objetos que maneja, tipos de datos de las variables, longitud de las variables, fabricante, parámetros de funcionamiento, etc...), también tendrá información de timers, errores, servicios y configuración.

7. Capacidad para detectar la adición o eliminación de un nuevo dispositivo a la red. Los esclavos tendrán drivers que son archivos y contienen toda la información referente al esclavo, cada vez que se adiciona un esclavo, el maestro lee estos drivers y actualiza la base de datos.
8. Generación del frame (mensaje para la comunicación que utiliza el protocolo, este mensaje tiene varios campos que son: delimitador de inicio, dirección destino, dirección origen, campo de control, datos, campo de verificación y delimitador de final).
9. Interpretación de los frames recibidos.
10. Detección de fallas de transmisión.
11. Detectar la ausencia o falla de alguno de sus esclavos.
12. Generación de reportes del funcionamiento de la red, de los esclavos, parámetros, etc.

Los esclavos, que son los dispositivos que realizan las actividades de automatización y control, en este caso serán microcontroladores Atmel AT89S8252. Para el microcontrolador se desarrollara un programa hecho en lenguaje ensamblador, este programa se encargara de interpretar el frame enviado por el maestro y de desarrollar actividades propias de automatización y control. Al recibir el frame enviado por el maestro, tomara las siguientes acciones:

1. Detectar si es un frame válido
2. Detectar si el frame va dirigido a él (leyendo la dirección destino).
3. Verificar que los datos se los esta pidiendo su propio maestro (verificando el campo de dirección de origen).
4. Detección de fallas en el frame.
5. Leer los datos solicitados por el maestro, estos datos pueden ser las variables o el estado del esclavo.
6. Generar y enviar el frame de respuesta con los datos solicitados por el maestro.
7. Activar alarmas de inactividad de su maestro.
8. Almacenar los valores de las variables que maneja, para poderlas comunicar a su maestro.
9. Desarrollar la actividad de automatización y control correspondiente.

En el proyecto también se definen todos los parámetros relacionados con la implementación de la red, como son: cable, conectores, interfaz entre computador personal y el medio de comunicación usado por PROFIBUS que es RS-485. Se desarrollara de circuito electrónico para el funcionamiento del microcontrolador 8051 y la interfaz con RS-485.

#### 4. MONTAJE DE LA RED PROFIBUS

En la figura 1 se puede ver de manera global el funcionamiento del sistema. Cabe recordar que la red es un bus de datos y que solamente puede haber una estación transmitiendo en un momento determinado. El software del maestro es el que se encarga de la sincronización de la red y de evitar que una estación envíe un mensaje cuando no le corresponde. A continuación se describirá el funcionamiento del sistema maestro. En esta etapa de iniciación el maestro debe leer de la base de datos la información de cada uno de sus esclavos (índice, variables u objetos, estado actual,...) y almacenarla en una matriz para posteriormente cuando tenga el token comunicarse con sus esclavos. También presentará en pantalla todos los dispositivos conectados a la red y la información con la que cuenta de cada uno de sus esclavos, ver Figura 2.

Una vez que la estación maestra obtenga el token, empieza la comunicación con cada uno de sus esclavos, esta comunicación la hace de una manera ordenada de acuerdo al índice de cada esclavo (Hay que recordar que todos los dispositivos conectados a la red deben tener un índice o identificador único, preferiblemente un número consecutivo sin importar si es un maestro o un esclavo). La comunicación con el esclavo puede ser de solicitud de información y de envío de información para cambiar el valor de algunas de las variables del esclavo, cuando le envía el mensaje (el frame, es decir el formato del mensaje es especificado por el protocolo) al esclavo, el maestro pasa a un estado de espera de respuesta por parte del esclavo. En este estado el maestro espera un tiempo determinado, si pasado el tiempo el esclavo no responde, el maestro vuelve a intentar el número de veces especificado en la configuración, si no responde debe mostrar en pantalla un mensaje de alarma, notificar la falla, actualizar la base de datos y pasar a enviar información al siguiente esclavo. En la Figura 3 se muestra un diagrama del funcionamiento del sistema maestro. Una de las rutinas más importantes del sistema maestro es la construir e interpretar el frame (trama de información enviada o recibida por la red).

En protocolo PROFIBUS el frame se estructura así:

SD	FC	D	SA	INFO...	FCS	ED
----	----	---	----	---------	-----	----

Donde:

SD = Delimitador de inicio = 11011000

FC = Control del frame = Tipo de frame enviado, los tipos de frames (Pasar\_Token, rame\_sin\_respuesta, Frame\_con\_respuesta, Frame\_de\_respuesta, Token\_recibido)

DA = Dirección destino = 1 byte con la dirección destino.

SA = Dirección fuente = 1 byte con la dirección del maestro.

INFO = Información = En este campo se pone el índice de cada variable del esclavo y el valor a enviar o una "p" si es solicitud de datos.

FCS = Secuencia de control

ED = Delimitador final = 11111100

Por ejemplo, si el maestro actual que tiene el índice 1, le va a enviar un frame a la estación con índice 3, solicitándole el valor actual de la variable 1, el frame quedara así:

SD=11011000

FC = Frame\_con\_respuesta = 79

DA = Dirección destino = 3

SA = Dirección fuente = 1

INFO = 1p

FCS = 11111111

ED = 11111100

Como se ha dicho anteriormente, PROFIBUS opera con la norma RS-485, por lo tanto es necesario desarrollar un dispositivo electrónico que convierta las señales RS-232 a RS-485 y viceversa. En la figura 4 se muestra el esquema de una alternativa para este convertidor. Este convertidor esta formado de un circuito integrado MAX232 que toma las señales del puerto serial RS-232 del computador (Estas señales son de -12V a 12V) y las convierte a voltajes TTL (5V), una vez convertidas estas señales, se las envía al circuito integrado 75176 que es el que las convierte a las señales diferenciales manejadas por la norma RS-485. También se usa un inversor 7404 que es el que se encarga de conmutar el 75176 para que reciba o envíe datos según sea necesario. Los conectores finales se modificaron para que fueran ambos DB9, de los cuales un extremo va al computador y el otro extremo a la red PROFIBUS. También se desarrollo una caja para esta interfaz, la cual ayuda a proteger el prototipo del ambiente exterior. Este prototipo opera a un voltaje de 5V. El RS-485 es un medio físico de transmisión y no un protocolo como la norma RS-232, por lo tanto en el

RS-485 pueden conectarse buses de distintas configuraciones pero con características físicas semejantes como niveles de tensión y tipo de conexión. Los niveles de tensión empleados van entre  $\pm 5V$  y  $\pm 6V$  para el nivel lógico 1 y 0V para el nivel lógico 0, y las distancias y frecuencias admitidas en el bus son (1200m a 1500m y 2400 baudios a 19200 baudios). Estas líneas deben cargarse a la salida con resistencias de terminación de línea entre 100  $\Omega$  y 250  $\Omega$ , dependiendo del tipo de cable empleado.

Es un enlace XON, XOFF (que existen solo líneas de datos y a lo sumo una línea tierra), en este caso solo usa una misma línea para transmitir y recibir ya que es semidúplex, esto requiere un software de control de enlace (capa OSI 2) que haga conmutar la línea para que la terminal transmita o reciba datos, la figura 5 muestra el principio de enlace. Este tipo de enlace se suele utilizar en topología de bus como lo muestra la figura 6. En la conexión en red, el número máximo de terminales conectadas suele estar limitado a 32 por razones de carga, sin embargo pueden utilizarse repetidores o amplificadores de bus para tener más terminales.

Por otro lado, los esclavos, que son los dispositivos que realizan las actividades de automatización y control, en este caso serán microcontroladores Atmel AT89S8252. Las características de este microcontrolador son [21]: Compatible con el Intel 8051, microcontrolador de 8 bits, 8K bytes de memoria flash, 2K bytes de memoria EEPROM, Voltaje de operación de 4V a 6V, 256 bytes de memoria RAM interna, 32 líneas I/O programables, 3 Timers/Counters de 16 bits, Nueve fuentes de interrupción, Puerto serial programable. La figura 7 muestra un esquema general del funcionamiento del sistema.

El circuito electrónico del sistema esclavo se montará en una tableta de baquelita con pistas de cobre para prototipos. El microcontrolador esta conectado directamente al circuito integrado 75176 que como ya se ha dicho es el que convierte las señales de transmisión de microcontrolador a las señales diferenciales manejadas por RS-485. El puerto p1.7 es el encargado de conmutar el 75176 para que envíe o reciba datos según sea necesario, para realizar esta actividad el puerto p1.7 se debe conmutar por medio de software. La conexión para la red se hace con conectores DB9, estos van directamente conectados al 75176. Este circuito opera a un voltaje de 5V. En la figura 8 se muestra un esquema detallado de las conexiones del sistema esclavo. Como puede verse en

el esquema, el sistema queda prácticamente con los cuatros puertos de I/O libres para realizar las actividades de control Automatización y control asignadas. Uno de los algoritmos más importantes del sistema esclavo es la transmisión y recepción de información, este algoritmo desarrollado en lenguaje ensamblador se presenta a continuación:

;Rutina para los esclavos, el sistema recibe un frame del PC (maestro),

```

Indice equ 43      ;identificador del esclavo
NumVar equ 2       ;número de variables u objetos
                    que maneja el esclavo
TRANSM equ P1.7    ;activa transmisión y
                    recepción del max485
                    org 00H
                    ljmp Inicio
                    org 23H      ;Interrupción puerto serial
                    jb ri,recibe
                    clr ti
                    setb TRANSM  ;Activa transmisión del esclavo
                    cjne @r1,#0h,siga
                    mov r1,#20h
                    clr TRANSM    ;VUELVE A ESCUCHAR
SOLAMENTE
                    reti
siga: mov a,@r1
                    mov sbuf,a
                    inc r1
                    reti
recibe:
                    clr ri        ;reseteo recepción
                    mov a,sbuf
                    cjne a,#252,final ;carácter final del frame (ED)
                    mov @r1,a
                    inc r1
                    mov @r1,#0h   ;adiciono un carácter nulo para
                    saber cuando dejar de transmitir
                    mov r1,#20h   ; cargo en a el valor de SD
                    (Delimitador de inicio)
                    cjne @r1,#216,sal ; Si SD no es el delimitador de
                    inicio correcto, salgo
                    mov r1,#21h   ; cargo en a el valor de FC (frame
                    de control)
                    cjne @r1,#79,sal ; Verifico el valor de FC, si es
                    79, es un frame con respuesta
                    mov r1,#23h   ; cargo en r1 el valor de DA
                    (Dirección destino)
                    cjne @r1,#Indice,sal ;Comparo el DA con el
                    índice del esclavo, Si es igual puedo responder
                    ljmp verifi    ;procedimiento para verificar el
                    frame
                    reti

```

```

final: mov @r1,a      ;Si no es el delimitador final,
almacena el valor enviado
      inc r1
      reti
sal:   mov r1,#20h   ;reinicia el valor de r1 y sale
      reti
Inicio
      mov 7fh,#'I'   ;dirección donde están
almacenadas la variable 1
      mov 7eh,#'J'   ;dirección donde están
almacenadas la variable 2
      mov 7dh,#'L'   ;dirección donde están
almacenadas la variable 3
      mov TH1,#$FD   ;transmission a 9600 b/s
      mov TL1,#$FD
      mov TMOD,#$20 ;Timer1 modo 2
      mov TCON,#$00
      setb TR1       ;Activo timer 1
      mov SCON,#$50 ;01010000, modo 1 de
transmisión
      mov IE,#$90   ;10010000, habilita interrupción
serial
      clr TRANSM     ;EL MICRO ESCUCHA
SOLAMENTE
      clr ri
      mov r1,#20h
PROGRAM NOP ;ciclo repetitivo de espera
      NOP
      NOP
      NOP
      ljmp PROGRAM
verifi: ;verificación y frame de respuesta
      ;verifico si el frame es para pedir datos o para
actualizar.
      ;si es para pedir datos, los campos donde van los
datos lleva
      ;el valor "p", si no llevan el valor que se actualiza
      mov r1,#26h   ;dirección en la que se almacena
el valor de la variable 1
      cjne @r1,#'p',ver1
      sjmp ver11
ver1:  mov 7fh,@r1
ver11: mov r1,#28h   ;dirección en la que se
almacena el valor de la variable 2
      cjne @r1,#'p',ver2
      sjmp ver22
ver2:  mov 7eh,@r1
ver22: mov r1,#2Ah   ;dirección en la que se
almacena el valor de la variable 1
      cjne @r1,#'p',ver3
      sjmp frame
ver3:  mov 7dh,@r1
      ;Procedimiento que armar el frame a enviar
frame: mov 21h,#71   ;Cambio el FC, a frame sin

```

```

respuesta
      mov a,23h
      mov 23h,24h   ;Cambio dirección destino
      mov 24h,a     ;Cambio dirección fuente
      mov r1,#26h   ;dirección donde se encuentra la
primera variable del objeto que llego
      mov @r1,7fh
      mov r1,#28h   ;dirección donde se encuentra la
segunda variable del objeto que llego
      mov @r1,7eh
      mov r1,#2Ah   ;dirección donde se encuentra la
tercera variable del objeto que llego
      mov @r1,7dh
      setb TRANSM   ;Activa trans. del esclavo
      mov r1,#20h   ;empiezo a responder el frame
      mov a,@r1
      inc r1
      mov sbuf,a
      reti

```

## 6. CONCLUSIONES

Realizada la prueba se vio que los maestros se comunican con cada uno de sus esclavos y pasan el frame al maestro siguiente en el tiempo asignado, muestran la información de las variables que manejan los esclavos, si ocurre un error lo notifican al instante, respetan los tiempos asignados a cada timer. Los esclavos interpretan adecuadamente el frame, responden cuando se les solicita información, leen adecuadamente los sensores y realizan todas las actividades esperadas. En la prueba se hizo una muestra estadística y se vio que con el cambio de cable mejoró sustancialmente el funcionamiento de la red, por lo tanto es muy importante seleccionar un cable adecuado para este tipo de redes.

## REFERENCIAS

- [1]. FRED HALSALL. Data communications, computer networks and open systems. Addison-Wesley. 1992.
- [2]. HOOPER, et al. Diseño de redes locales. Addison-Wesley Iberoamericana. 1989
- [3]. J MARTIN, et al. Local area networks, Architectures and implementations. Prentice Hall. 1994.
- [4]. U. REMBOLD, et al. Compute integrated manufacturing and engineering. Addison Wesley. 1993. Pag. 258-292.
- [5]. VINCENT C. JONES. MAP/TOP Networking. Mc Graw Hill. 1987. Pag. 1-23
- [6]. ANSI/IEEE ESTANDAR 802.4. Local area Networks, Token passing bus access method. IEEE. 1985.

- [7]. ANSI/IEEE ESTANDAR 802.5. Local area Networks, Token ring access method. IEEE. 1985.
- [8]. JOSEP BALCELLS, JOSE LUIS ROMERAL. Autómatas programables. Alfaomega Maracombó. 1997. Pag 271-384.
- [9]. PROFIBUS, Technical description. Order No. 4.002
- [10]. N. I. AutomationView. Vol. 4. No 2, 1999.
- [11]. SMAR, Fieldbus Tutorial.
- [12]. N.I. AutomationView. Vol. 5. No 3. Otoño 2000.
- [13]. N.I. AutomationView. Vol. 4. No 3. Otoño 1999.
- [14]. ROGER S. PRESSMAN. Ingeniería del Software. Tercera edición. McGraw-Hill. 1993.
- [15]. EDWARD YOURDON. Análisis Estructurado Moderno. Prentice hall. 1993.
- [16]. RICHARD BARKER, CLIFF LONGMAN. CASE METHOD, Function and Process Modelling, ORACLE. ADDISON WESLEY. 1992

## ANEXOS

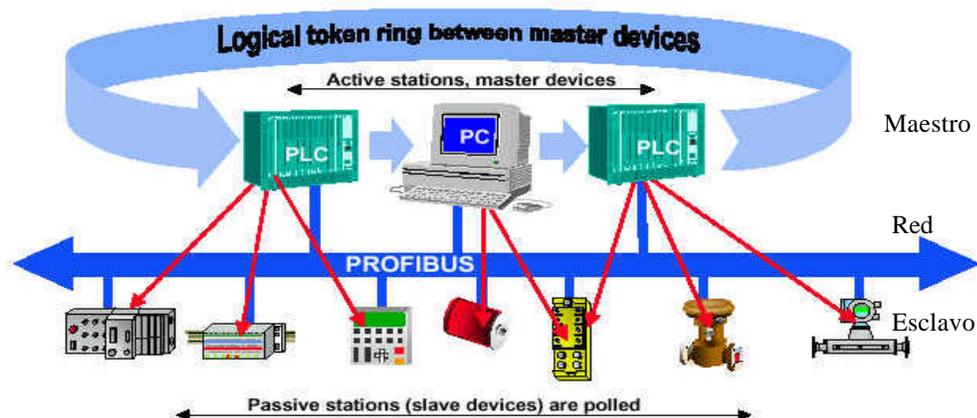


Fig. 1. Esquema de arquitectura maestro / esclavo [9]

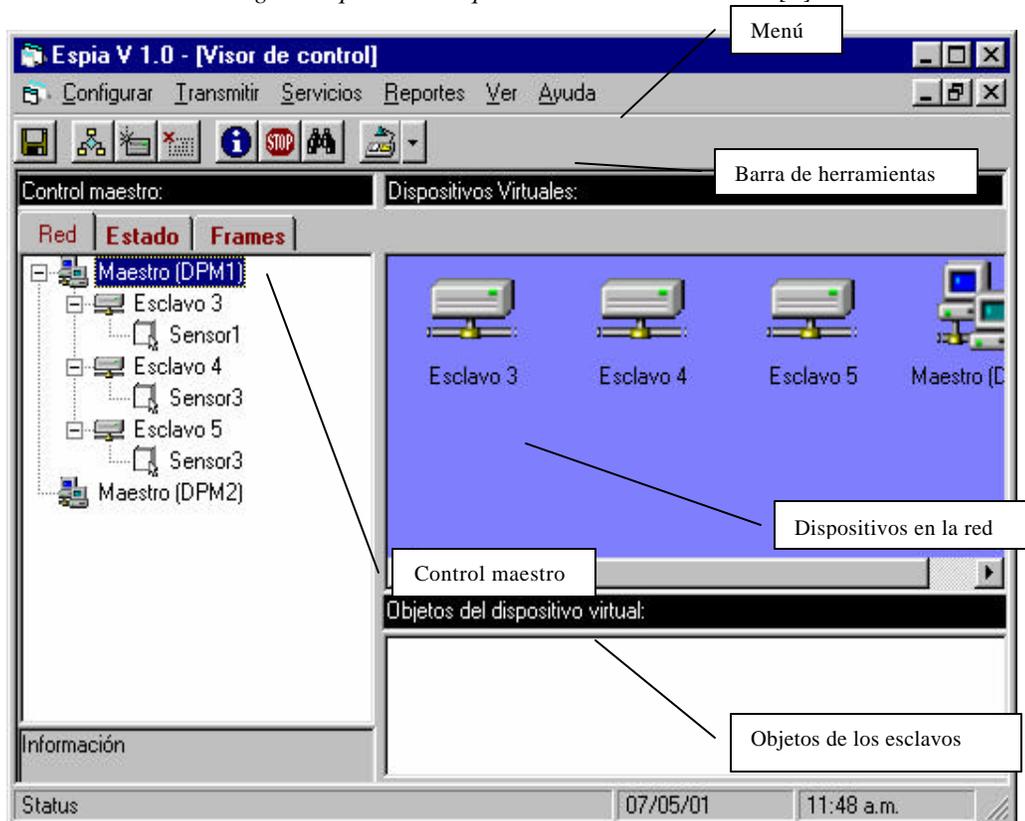


Fig. 2. Pantalla principal del sistema maestro (Software SCADA desarrollado en Visual Basic)

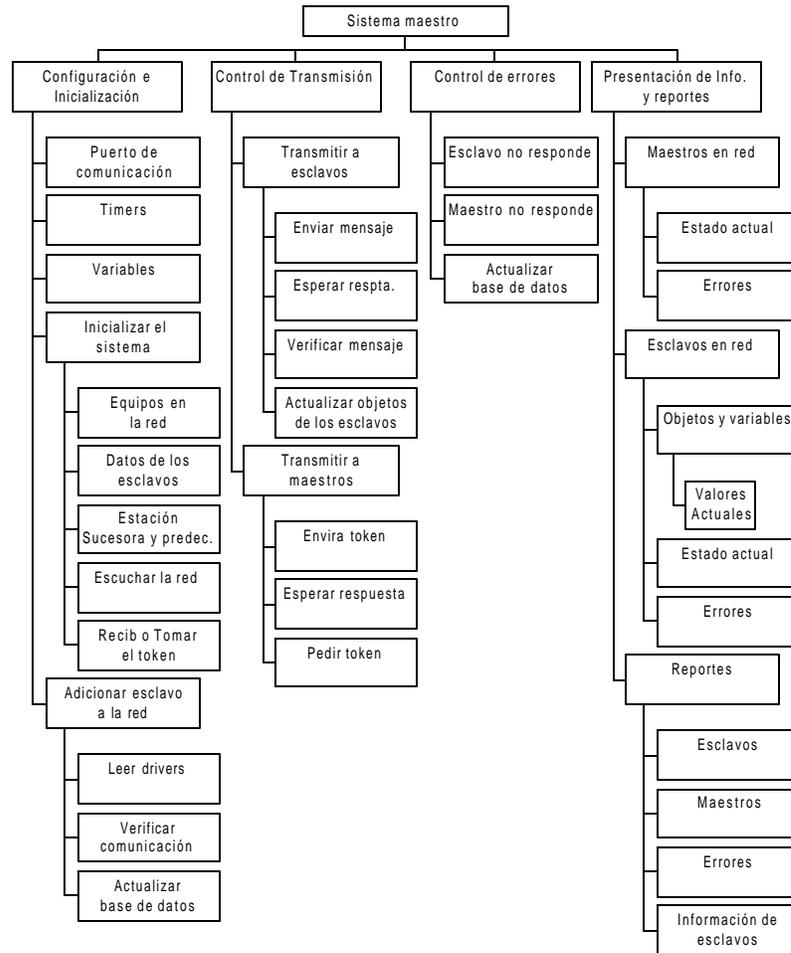


Fig. 3. Diagrama jerárquico del sistema maestro

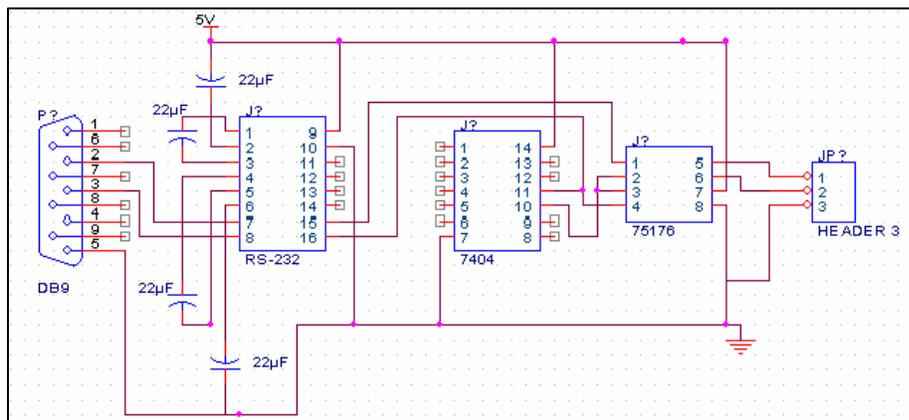


Fig. 4. Esquema del convertidor de señales RS-232 – RS-485 y viceversa

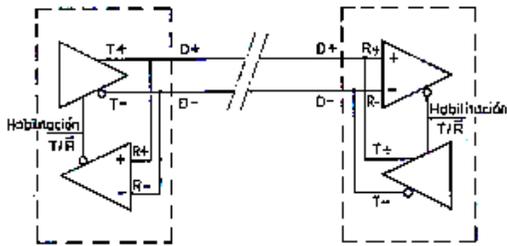


Fig. 5. Enlace punto a punto RS-485

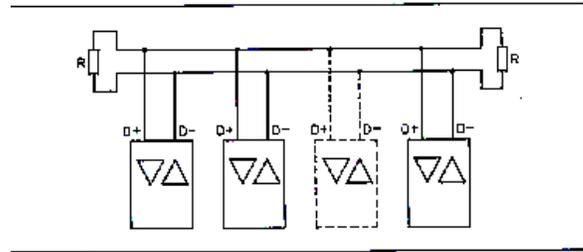


Fig. 6. Enlace red mediante bus RS-485

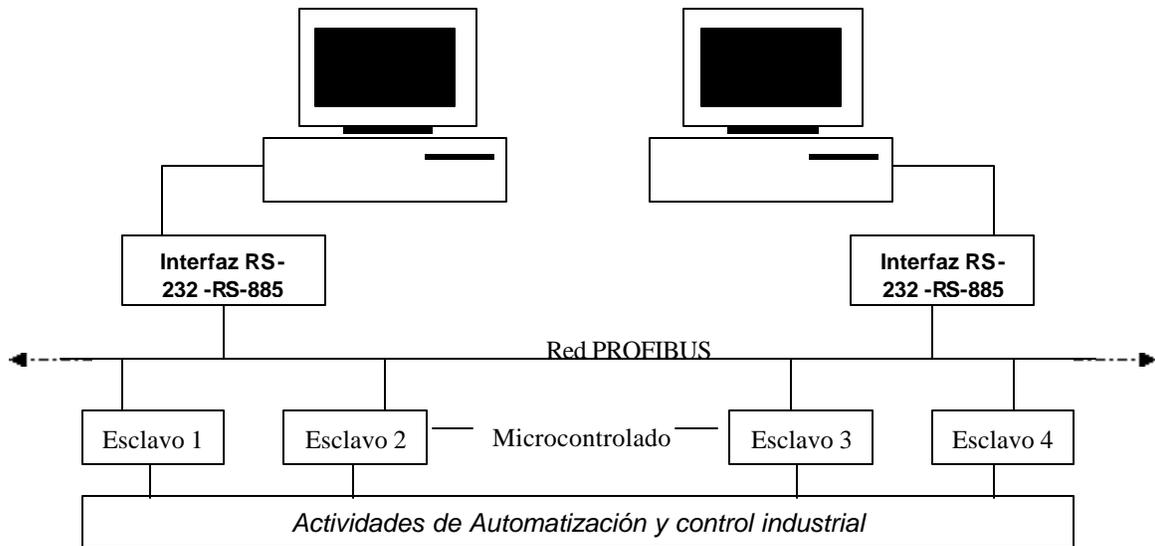


Fig. 7. Esquema general de la interfaz del microcontrolador

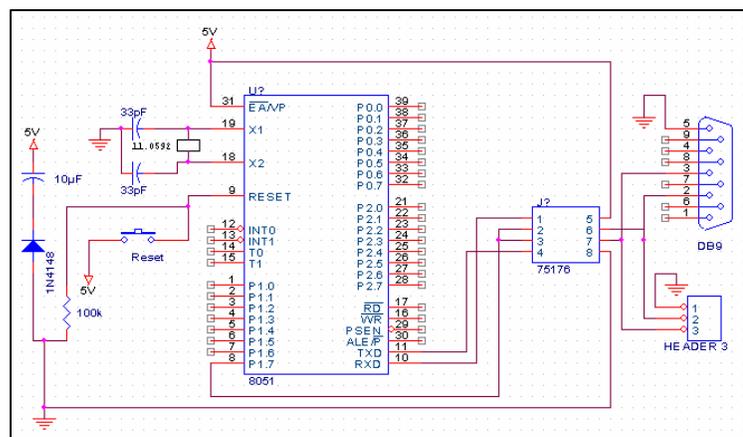


Fig. 8. Esquema del sistema esclavo.