



**ESCUELA POLITÉCNICA DEL EJÉRCITO  
SEDE – LATACUNGA**

**CARRERA DE INGENIERÍA EN  
ELECTRÓNICA E INSTRUMENTACIÓN**

**PROYECTO DE GRADO PARA LA OBTENCIÓN  
DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA E INSTRUMENTACIÓN**

**“DISEÑO E IMPLEMENTACIÓN DE UN SERVIDOR  
DE DATOS OPC, Y UN PROTOCOLO DE  
COMUNICACIÓN PARA  
MICROCONTROLADORES”**

**ANTONIO MARCO SALAZAR ALTAMIRANO**

**LATACUNGA, NOVIEMBRE DEL 2005**

## **CERTIFICACIÓN**

Certifico que el presente documento fue elaborado por el señor:

Salazar Altamirano Marco Antonio

Bajo nuestra dirección, como requisito para la obtención del título de  
Ingeniero en Electrónica Especialidad Instrumentación.

---

Ing. Nancy Guerrón.  
DIRECTOR DEL PROYECTO

---

Ing. Eddie Galarza  
CODIRECTOR DEL PROYECTO

## **DEDICATORIA**

Dedico este trabajo,  
a mis tíos: Consuelo, Enrique y Adela,  
y a la memoria de mi tío Manuel Heleodoro.  
Este trabajo es fruto de sus sacrificios y esfuerzos constantes.

## **AGRADECIMIENTO**

Agradezco a Dios,  
a mi Familia,  
a mis Amigos,  
y a mis Profesores.

# CONTENIDO

<b>INTRODUCCIÓN</b>	<b>X</b>
<b>CAPÍTULO 1</b>	
<b>GENERALIDADES</b>	<b>1</b>
1.1 Comunicación serie.....	1
1.1.1 Digitalización y transmisión digital.....	1
1.1.2 Protocolo de comunicación.....	2
1.1.3 Interfaz serie RS-232.....	2
1.1.4 Interfaz serie RS-485.....	3
1.1.4.1 Protección contra interferencias electromagnéticas.....	4
1.1.4.2 Tazas altas de transmisión de datos.....	4
1.1.5 El modelo OSI.....	5
1.1.6 Control de acceso al medio.....	7
1.1.6.1 CSMA/CD.....	8
1.1.6.2 Token passing.....	8
1.1.6.3 Polling.....	8
1.1.7 Topologías físicas de una red RS485.....	9
1.2 Fundamentos de comunicaciones industriales.....	10
1.2.1 Categoría de variables.....	10
1.2.2 Selección de protocolos a nivel de instrumentación.....	12
1.2.3 Categoría de instalaciones.....	13
1.2.4 Sistemas SCADA.....	14
1.2.5 Interfaces HMI.....	14
1.2.6 Tiempo real.....	15
1.2.7 La estructura abierta.....	15
1.3 Tecnologías para interoperabilidad entre aplicaciones.....	16
1.3.1 OLE (Objetos vinculados e insertados).....	17
1.3.2 DDE (Transferencia dinámica de datos).....	17
1.3.3 OPC (OLE para el Control de Procesos).....	19

1.3.3.1	Funcionamiento de OPC.....	20
1.3.3.2	Modelo de objetos de la especificación OPC.....	21
1.4	Arquitectura de PIC16FXXX de MICROCHIP.....	23
1.4.1	El procesador.....	24
1.4.1.1	La arquitectura Harvard.....	24
1.4.1.2	Segmentación del procesador.....	25
1.4.1.3	Instrucciones reducidas.....	25
1.4.1.4	Instrucciones simétricas.....	25
1.4.2	Organización de la memoria.....	26
1.4.2.1	Memoria de programa.....	26
1.4.2.2	Memoria de datos.....	27
1.4.2.3	Memoria EEPROM.....	28
1.4.3	Líneas de entrada y salida.....	28

## **CAPÍTULO 2**

### **DISEÑO DEL PROTOCOLO DE COMUNICACIÓN 29**

2.1	Características del hardware de red.....	29
2.1.1	Método de acceso al medio y topología de red.....	30
2.1.2	Protocolo de comunicación RS-232.....	30
2.1.3	Transceivers y características de RS-485.....	31
2.1.4	Medio de comunicación.....	32
2.1.5	Requerimientos de los nodos.....	33
2.1.5.1	Nodo esclavo.....	33
2.1.5.2	Nodo maestro.....	33
2.1.6	Tipo de servidor de datos.....	34
2.1.7	Comparación con el modelo OSI.....	34
2.2	Diseño del hardware adaptador de interfase.....	35
2.3	Conectores y cables.....	36
2.4	Diseño del hardware de estaciones esclavas.....	37
2.5	Consideraciones de diseño y características de la trama.....	38
2.6	Descripción de campos de la trama.....	40
2.6.1	Inicio de trama.....	41
2.6.2	Campo de dirección.....	41
2.6.3	Campo de control.....	42
2.6.4	Campo de inicio de localidad.....	43

2.6.5	Campo de longitud.....	43
2.6.6	Campo de datos.....	43
2.6.7	Campo chequeo de error.....	43
2.7	Tipos de tramas.....	44
2.7.1	Tramas normales.....	44
2.7.2	Trama especial de consultas (polling).....	44
2.7.3	Trama especial de difusión (broadcast).....	45

### **CAPÍTULO 3**

#### **SOFTWARE Y ALGORITMOS**

**46**

3.1	Software usado para el desarrollo del proyecto.....	46
3.2	Software implementado en el nodo maestro.....	47
3.2.1	Algoritmo de creación de grupos (tópicos).....	49
3.2.2	Descripción de tablas de configuración y datos.....	50
3.2.3	Algoritmo de consulta de nodos esclavos (polling).....	52
3.2.4	Implementación del servidor OPC.....	53
3.3	Software implementado en los nodos esclavos.....	55
3.3.1	Algoritmos de transmisión y recepción de datos.....	57
3.3.2	Organización de la información en la memoria.....	59
3.4	Algoritmo de detección de errores.....	61

### **CAPÍTULO 4**

#### **RESULTADOS EXPERIMENTALES**

**64**

4.1	Requerimientos para la instalación del software desarrollado.....	64
4.2	Pruebas del protocolo de comunicación.....	65
4.2.1	Lectura de la memoria RAM.....	65
4.2.2	Escritura de la memoria RAM.....	67
4.2.3	Lectura de la memoria EEPROM.....	67
4.2.4	Escritura de la memoria EEPROM.....	68
4.2.5	Trama especial de consultas (polling), lectura.....	69
4.2.6	Trama especial de consultas (polling), Escritura.....	71
4.2.6.1	Análisis de la información obtenida del nodo esclavo.....	72
4.2.7	Trama especial de difusión (broadcast).....	75

4.3	Desconexión de un nodo esclavo.....	75
4.4	Consumo de recursos del nodo esclavo.....	76
4.5	Análisis de tiempo de la trama especial de consultas.....	79
4.6	Longitud de la red y terminadores de red.....	81
4.7	Consumo de recursos del nodo maestro.....	82
4.8	Efectos de carga del servidor OPC.....	82
4.9	Efectos de finalización del tiempo demostrativo de la biblioteca WtOPCSrv.....	84

## **CAPITULO 5**

### **CONCLUSIONES Y RECOMENDACIONES 85**

5.1	Conclusiones.....	85
5.2	Recomendaciones.....	89

### **BIBLIOGRAFÍA 91**

### **GLOSARIO 93**

### **ANEXOS 103**

A.	Diagrama eléctrico del hardware implementado.....	103
B.	Representación de la información en las localidades de memoria del MCU.....	104
C.	Tramas implementadas para la comunicación de datos.....	105
D.	Terminales de conexión de los módulos construidos.....	106
E.	Disposición de pines de los microcontroladores: PIC16F877A y PIC16F628A.....	107
F.	Memoria RAM del PIC16F877A.....	108
G.	Memoria RAM del PIC16F628A.....	109
H.	Manual de usuario.....	110
H.1.	Introducción.....	110
H.2.	Instalación del software.....	110
H.3.	Inicio del programa.....	112
H.4.	Descripción del formulario principal.....	113
H.5.	Configuración de usuarios.....	117
H.6.	Configuración del puerto serie.....	119
H.7.	Definición de dispositivos de la red.....	119

H.8.	Configuración de dispositivos.....	120
H.9.	Configuración del servidor.....	121
H.10.	Análisis de tramas.....	122
I.	Software desarrollado.....	125

## INTRODUCCIÓN

En la actualidad las empresas de manufacturación de software y equipos; que realizan el control, adquisición y monitoreo de datos, se han convertido en los proveedores de tecnología, para ello utilizan estándares industriales que permiten la integración de diversos fabricantes para realizar complejos sistemas de adquisición de datos y control por supervisión (SCADA). Los sistemas SCADA están constituidos principalmente por sistemas de comunicación de datos entre dispositivos (protocolos de comunicación, servidores de datos OPC y otros), por lo tanto estos sistemas se convierten en el pilar del control por supervisión de la información.

La adquisición de tecnología capaz de implementar un sistema SCADA de cualquier tipo de proceso obliga al consumidor a utilizar costosos productos (PLC's, controladores, software servidores de datos y clientes de datos para interfaces humano máquina "HMI"), los mismos que no permiten ser modificados por el usuario final, además de requerir las tutorías para cada uno de los fabricantes de los diferentes componentes del sistema SCADA.

El objetivo de este proyecto es diseñar e implementar el software y el hardware necesario para monitorear microcontroladores en una red, el cual cumpla con las especificaciones de un servidor de datos con enlace de objetos embebidos para control de procesos (OPC).

Para crear soluciones (aplicaciones) de bajo costo y que éstas sean integradas en los sistemas SCADA, se desarrolló una red con el estándar industrial RS485 para 32 nodos de red, un protocolo de comunicación para microcontroladores y se implementó la especificación de servidores OPC-DA (Acceso a Datos) con ayuda de la biblioteca WTOPCSrv.dll.

Este proyecto de tesis está organizado en cinco capítulos, los cuales describen toda la información necesaria para cumplir con el objetivo propuesto en este proyecto. Al final de este documento, se incluye: la bibliografía, el glosario de términos técnicos y los anexos en el cual se encuentra el manual del usuario de la aplicación desarrollada y el código fuente de los nodos esclavos.

El capítulo 1 es una introducción general a los conceptos de software y hardware utilizados en sistemas de monitoreo industrial.

El capítulo 2 expone el diseño del hardware (nodo maestro, nodo esclavo, adaptador de interfaces y conectores de red) y del protocolo de comunicación utilizado en el proyecto propuesto.

El capítulo 3 trata del software utilizado para el desarrollo del proyecto y los algoritmos implementados en el nodo maestro y esclavos.

El capítulo 4 presenta los resultados experimentales y el análisis de funcionamiento del protocolo de comunicación implementado.

El capítulo 5 aporta con las conclusiones y recomendaciones más importantes obtenidas en el desarrollo del proyecto.

En general, a lo largo de este proyecto he evitado profundizar en la descripción de funcionamiento del hardware utilizado para no cansar al lector y permitir que la lógica de funcionamiento de los principales algoritmos implementados se asimile fácilmente.

Espero que este proyecto pueda ser útil a las personas interesadas en disminuir los altos costos en la implementación de aplicaciones. De ante manos agradezco todas las sugerencias que tengan a bien hacerme llegar para mejorar este proyecto.

Marco Antonio Salazar

[marcosaam@hotmail.com](mailto:marcosaam@hotmail.com)

Latacunga, Cotopaxi.

# **CAPÍTULO 1**

## **GENERALIDADES.**

### **1.1 COMUNICACIÓN SERIE.**

#### **1.1.1 DIGITALIZACIÓN Y TRANSMISIÓN DIGITAL.<sup>1</sup>**

Consiste en digitalizar la señal análoga que se desea enviar y transmitirla utilizando técnicas digitales. Una de las ventajas que tiene la digitalización es su mayor resistencia a la contaminación por ruido durante la transmisión. Cuando un ruido afecta a una señal digital generalmente cambiará su amplitud, su fase y velocidad de transmisión.

Es posible transmitir los pulsos directamente, a esta técnica de transmisión se denomina transmisión en banda base. Para la transmisión en banda base no se emplea los niveles asociados con la lógica TTL (0 o 5 V), porque se tiene una componente continua que es difícil de manejar, además se tiene el problema de sincronización entre transmisor y receptor durante la transmisión. Problemas se resuelven empleando técnicas especiales de codificación como: NRZ y bifásicas.

En la transmisión NRZ, es difícil determinar cuando termina un bit y cuando empieza otro, si se transmite una sucesión de ceros o unos, cualquier corrimiento en el tiempo entre transmisor y receptor resultará en una pérdida de sincronización. También está el

---

<sup>1</sup> CORRALES, Luis, "Interfaces de Comunicación Industrial", EPN, Ecuador, 2004

problema de la componente de continua. Para resolver este problema se usan las técnicas de codificación bifásicas.

### **1.1.2 PROTOCOLO DE COMUNICACIÓN.**

Los protocolos son un conjunto de hábitos y procedimientos utilizados en las relaciones interpersonales. Cuando es usado bajo el contexto de redes de comunicación el término protocolo tiene un significado similar pero a un nivel más específico, esto es, un protocolo de red es un conjunto de reglas, secuencias, formatos de mensajes y procedimientos bien detallados que posibilitan la transferencia de datos entre dos o más sistemas de computación.

Si la transferencia de información del protocolo es transmitida o recibida, bit a bit, se conoce como protocolo de comunicación serie.

### **1.1.3 INTERFAZ SERIE RS-232.**

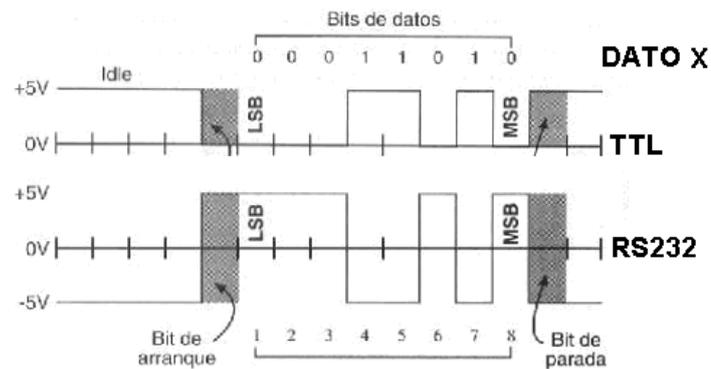
Hay dos tipos de comunicaciones digitales seriales: síncronas y asíncronas. En una transmisión síncrona los datos son enviados un bit a continuación de otro por una línea que une la salida del transmisor. El transmisor y el receptor son sincronizados con una línea extra que trasmite pulsos de reloj. El uso de esta técnica implica la existencia de un cable extra para llevar la señal de reloj, lo cual resulta en un costo extra. Esta técnica no se usa para la aplicación que aquí se analiza.

En la transmisión asíncrona no se emplea una señal de reloj, más bien se utiliza una técnica que consiste en “encapsular” los datos con un bit de inicio y uno o dos bits de parada, y así no es necesaria la línea extra de sincronismo.

Pero encapsular con un bit de inicio y otro de parada los datos no es suficiente, hay otras reglas de transmisión sobre las que se deben poner de acuerdo el transmisor y receptor. Por ejemplo, mientras el estado de la línea está en alto, el receptor deberá interpretar como que no existe transmisión y, por lo mismo, el canal está en modo de espera (idle). Cuando la línea de comunicación cambia de estado se recibe el bit de inicio, el receptor

debe interpretar ese cambio como el comienzo de la transmisión. Pero, ¿que sucede si ese cambio de estado fue producido por un transitorio? En situaciones así, el algoritmo de comunicación debe instruir al receptor esperar cierto tiempo para volver a leer la línea y así asegurarse que el cambio de estado corresponde efectivamente al bit de inicio. Por todo lo indicado, antes de iniciar cualquier comunicación con el puerto RS-232 se debe determinar el protocolo a seguir.

El Protocolo serial determina: el número de bits de datos, la paridad, el número de bits de parada, la velocidad de transmisión y el protocolo de control de flujo. Los bits de datos son enviados al receptor después del bit de inicio, el bit menos significativo es transmitido primero. Dependiendo de la configuración de la transmisión, un bit de paridad es enviado después de los bits de datos.



**Figura 1.1 Trama RS-232 típica.**

Los voltajes más usados son +12V y -12V. El estado de reposo (idle) se representa con un 1 lógico; es decir, -12V. Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros.

#### **1.1.4 INTERFAZ SERIE RS-485.**

El alcance de la RS-232 es apenas de 15 m, un valor muy reducido para aplicaciones industriales reales. Es por esto que se creó el estándar RS-485. La característica más relevante de la RS-485 es que puede trabajar en modo diferencial.

Los estándares RS-485 y RS-422 tienen mucho en común, y por tal razón a menudo se los confunde, la tabla 1.1 los compara. RS-485 se especifica como bidireccional, half-duplex y es el único estándar EIA/TIA que permite receptores múltiples y drivers en configuraciones tipo "bus". La EIA/TIA-422, por otro lado, especifica un driver único, unidireccional con receptores múltiples. Los componentes de RS-485 son compatibles hacia atrás e intercambiables con los de la RS-422, pero, los drivers RS-422 no deben ser empleados con la RS-485 porque no pueden proveer control del bus.

**Tabla 1.1 RS-232, RS-422 y RS-485 estándar.**<sup>2</sup>

<b>Descripción</b>	<b>RS-232</b>	<b>RS-422</b>	<b>RS-485</b>
Modo de operación	No Balanceado	Diferencial	Diferencial
Número máximo de transmisores	1	1	32
Número máximo de receptores	1	10	32
Longitud del cable	75 metros	1200 metros	1200 metros
Máxima velocidad de los datos	20 kbps	10 Mbps	10 Mbps
Rango de 1 lógico	-5 a -25 V	-2 a -6 V	-1.5 a -6 V
Rango de 0 lógico	+5 a +25 V	+2 a +6 V	+1.5 a +6 V
Carga del transmisor	>3 ohm	>100 ohm	>60 ohm
Resistencia de entrada del receptor	de 3 a 7 K ohm	>4 Kohm	>12 Kohm
Sensibilidad del receptor	+/- 3.0 V	+/- 200 mV	+/- 200 mV

#### **1.1.4.1 PROTECCIÓN CONTRA INTERFERENCIAS ELECTROMAGNÉTICAS.**

Los enlaces diferenciales de las interfaces RS-485 y RS-422 proporcionan una transmisión confiable ante la presencia de ruido, y las entradas diferenciales de los receptores pueden rechazar voltajes en modo-común elevados.

#### **1.1.4.2 TAZAS ALTAS DE TRANSMISIÓN DE DATOS.**

<sup>2</sup> GALARZA, Eddie, "Comunicaciones para Instrumentación", ESPE, Ecuador, 2004

Con velocidades de transmisión altas se comienza a tener problemas con las reflexiones (Jitter), para contrarrestar éstas, se instalan resistencias de terminación con un valor igual al de la impedancia del cable. Para cables RS-485 comunes (pares trenzados de 24AWG), esto significa resistencias de 120 en los dos terminales. Se puede afirmar que la RS-485 fue el primer intento por evitar un enlace físico de datos para cada sensor o transmisor. Es útil cuando no se deben monitorear o controlar muchas variables. Sin embargo, se tiene todavía un problema: no se puede conectar todos los dispositivos a un solo pórtico RS-485, para lograr una mejor flexibilidad y eficiencia que es lo que actualmente demanda la industria moderna.

### **1.1.5 EL MODELO OSI.<sup>3</sup>**

En 1984, la Organización Internacional de Estandarización (ISO) desarrolló un modelo llamado OSI (Open Systems Interconnection, Interconexión de sistemas abiertos). El cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Este modelo es el mejor conocido y el más usado para describir los entornos de red.

El modelo OSI fue el primer intento, para unificar las tecnologías de red bajo un estándar único que permitiera la compatibilidad entre todas. Por esta razón, es una arquitectura estructurada bien definida.

OSI estructura la comunicación en 7 niveles. Dentro de cada nivel se encuentra una o varias "unidades funcionales". Una unidad funcional es el elemento que se implementa en cada nivel, para cumplir con las funciones asignadas, la unidad funcional de nivel n requiere servicios de su nivel inferior, o nivel n-1. A partir de estos servicios, la unidad de nivel n ofrece unos servicios más complejos a la unidad de nivel n+1.

---

<sup>3</sup> [http://www.pchardware.org/redes/redes\\_osi.htm](http://www.pchardware.org/redes/redes_osi.htm)

Los niveles en que se divide el modelo OSI son:<sup>4</sup>

7. Aplicación
6. Presentación
5. Sesión
4. Transporte
3. Red
2. Enlace de datos
1. Física

**Figura 1.2 Niveles del modelo OSI.**

**Nivel Físico:** Es el nivel más bajo del modelo. Incluye los aspectos de interconexión física (líneas de transmisión, voltajes, circuitería,...). En este nivel se garantiza la conexión, pero no la fiabilidad de ésta.

**Nivel de Enlace de Datos:** Asegura una comunicación fiable entre dos máquinas directamente conectadas. Se encargará de la supervisión de errores en la línea, control de flujo, y acceso al medio físico. La unidad de intercambio es la trama de datos.

**Nivel de Red:** Encargada de crear decisiones de ruta y remitir paquetes para dispositivos que están más lejos que un enlace simple. Un enlace comunica un dispositivo con el otro y no a través de un tercer dispositivo.

**Nivel de transporte:** Se asegura que los paquetes sean enviados libres de error, secuencialmente y sin pérdidas o duplicaciones. También divide los mensajes largos de la capa de sesión (en remitente) y los ensambla en destino.

---

<sup>4</sup> <http://www.eui.upv.es/ineit-mucon/Applets/TutorialTCPIP/isoosi.html>

El destino envía una respuesta de recibo al remitente del mensaje, indicando si se recibió el mensaje o no (Orientado a conexión).

Nivel de Sesión: Asegura la reanudación de una conexión interrumpida por cualquier causa ajena al usuario a partir de un punto seguro, evitando problemas en sistemas transaccionales. Para ello debe implementar mecanismos fiables de recuperación.

Nivel de Presentación: Permite el intercambio de información entre computadores con distinta representación de los datos. Debe convertir los datos desde el formato local al estándar de red y viceversa. También debe realizar funciones de compresión y encriptación de los datos.

Nivel de Aplicación: Proporciona soporte a las aplicaciones de usuario, interactuando con el mismo.<sup>5</sup>

### **1.1.6 CONTROL DE ACCESO AL MEDIO.**

Una vez que se han conectado las máquinas a un medio físico compartido, a más de las direcciones lógicas y físicas que identifican individualmente a cada estación, es necesario que se establezcan reglas entre los dispositivos que están compartiendo dicho medio, para ponerse de acuerdo en como usarlo. Estas reglas constituyen los protocolos de acceso al medio, o protocolos MAC (Media Access Control).

Dicho de otra manera, el control de acceso al medio se refiere a los protocolos que determinan cómo y cuál de los computadores, PLCs o dispositivos de campo inteligentes en una red compartida puede transmitir los datos. Hay tres métodos de control de acceso al medio:

---

<sup>5</sup> SELDON, Tom, "Enciclopedia de redes", McGraw-Hill, España, 1994

#### **1.1.6.1 CSMA/CD.**

Acceso múltiple con detección de portadora y detección de colisiones (CSMA/CD). Es un método no determinístico, en donde una estación de la red espera a que el medio esté libre para realizar la transmisión, al existir una colisión de datos entre dos estaciones, tendrán que transmitir más tarde; con una penalización en tiempo, superior al que requieren otras estaciones.

#### **1.1.6.2 TOKEN PASSING.**

El paso de la Señal (Token Passing). Es un método determinístico en que cada dispositivo espera el paso de una trama especial, el cual da autorización a cada estación para transmitir. No existen colisiones y cada dispositivo tiene su turno de transmisión.

#### **1.1.6.3 POLLING.**

Este protocolo de acceso es muy popular en las redes industriales. Una estación, que en un determinado momento hace de “Maestro”, secuencialmente va interrogando (polling) al resto de dispositivos que están en la red (esclavos) y recogiendo la información si el dispositivo interrogado tiene algún dato.

En este esquema el maestro tiene el control del acceso al medio, existiendo dos posibilidades: o es el maestro el que ocupa el medio físico, o el esclavo que responde la consulta del maestro.

Los esclavos solo envían mensajes cuando el maestro así lo solicita. Hay dos tipos de transacciones:

Consulta – respuesta: Cada esclavo tiene una dirección única.

Difusión sin respuesta: Todas reciben y ninguna contesta.

El maestro envía un mensaje específico al esclavo y espera un tiempo específico hasta que este responda. El esclavo debe responder sea enviando el dato o enviando un mensaje corto para indicar que no tiene datos a transmitir. Si el dispositivo no responde dentro del tiempo fijado, el maestro asume que el esclavo tiene algún problema y continúa interrogando al resto de esclavos.

La comunicación de datos entre esclavos es ineficiente pues todo debe hacerse a través del maestro. Puesto que las configuraciones de comunicación Maestro / Esclavo utilizan este método de acceso, polling es referido a veces como método de acceso Maestro / Esclavo.

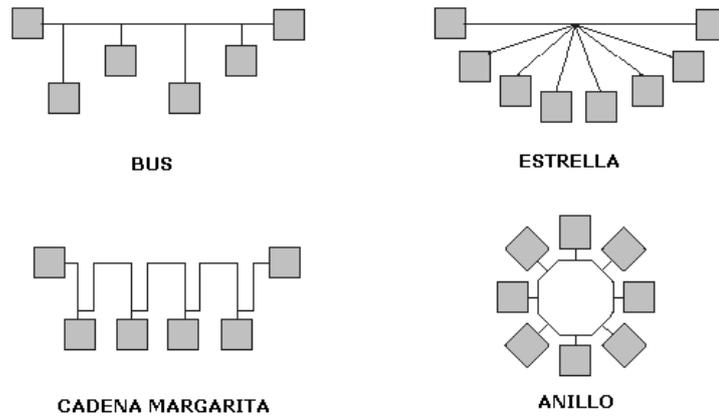
Vale aclarar que una red Maestro / Esclavo se diferencia de una tipo polling en que hay una estación maestro definida, mientras que en polling cualquier estación puede actuar de maestro.

### **1.1.7 TOPOLOGÍAS FÍSICAS DE UNA RED RS485.**

Desde el punto de vista físico, el estándar industrial RS485 recomienda la utilización de las topologías que se indican a continuación:

La topología cadena margarita (daisy chain), es utilizada por su facilidad para ser implementada y por que disminuye las reflexiones en la red.

Para conectar las estaciones existen diferentes tipos de cable físico. Cada uno tiene sus ventajas y desventajas pero en general las opciones que hoy en día están vigentes para la industria son: cable par trenzado industrial (ITP) y el cable de fibra óptica.



**Figura 1.3 Topologías físicas de una red RS485**

### **1.3. FUNDAMENTOS DE COMUNICACIONES INDUSTRIALES.**

Las comunicaciones industriales dependen de varios factores, entre ellos se analizan: la categorización de variables, el tipo de instalaciones, interfases humano maquina (HMI), entre otras.

#### **1.3.1. CATEGORÍA DE VARIABLES.<sup>6</sup>**

Dentro de un proceso se miden, supervisan y controlan las variables más relevantes del mismo. Las variables dentro de un proceso se miden con el propósito de supervisarlas, y/o con el propósito de llevar a cabo acciones de control a partir de su medición.

Es importante resaltar que existen variables que por la naturaleza del proceso, si llegasen a salir de un cierto rango considerado como máximo o mínimo permisible (ya sea en

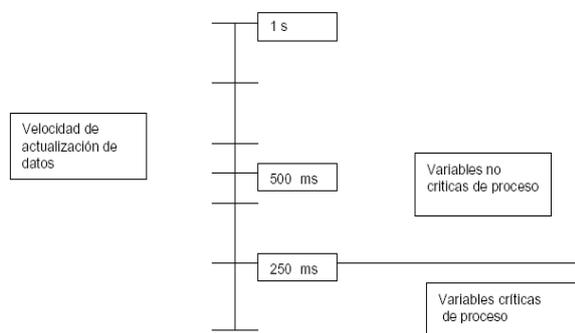
<sup>6</sup> NRF-046-PEMEX-2003, "Protocolos de Comunicación en Sistemas de Monitoreo y Control ", PEMEX, México,2003

forma repentina o paulatina), pudieran llegar a poner en riesgo al personal, al medio ambiente, a la instalación y/o a la calidad y cantidad de los productos que se producen; si no se realizan las acciones de control apropiadas en el tiempo adecuado. Se establece una clara definición de la categorización de las variables medidas en los diferentes procesos que se operan en las instalaciones, las cuales se han identificado en dos grupos:

Variables no críticas.

Variables críticas.

El criterio o factor considerado para establecer dicha categorización, es la velocidad de actualización de la variable de proceso medida. Este criterio se debe interpretar como la rapidez que se requiere para leer en tiempo real el valor de la variable medida, y realizar las acciones de control, en caso de que así sea requerido; para llevar las condiciones de proceso a un estado seguro y así no poner en riesgo al personal, al medio ambiente y/o a las instalaciones mismas.



**Figura 1.4 Categorías de variables por velocidad de actualización.**

La velocidad de actualización, es el tiempo que transcurre, desde que el instrumento deja los datos en el canal de comunicación, hasta que estos llegan al controlador a través de tarjetas de entrada/salida o tarjetas de comunicación.

En la figura 1.4 se muestran las categorías de variables en función de la velocidad de actualización del valor medido de las variables.

### **1.3.2. SELECCIÓN DE PROTOCOLOS A NIVEL DE INSTRUMENTACIÓN.**

La selección de protocolos de comunicación a nivel instrumentación se debe realizar en base a la aplicación de la siguiente forma, para:

Variables no críticas de proceso.

Variables críticas de proceso.

Sistemas de seguridad de proceso. (Sistemas de paro por emergencia y sistemas de gas y fuego)

La determinación del tipo de variable se debe realizar analizando en primer lugar la necesidad de medirla, y en segundo término, enfocándose específicamente al requerimiento de rapidez demandado por las variables de proceso; de manera que cualquier cambio en el valor medido de dicha variable sea leído por el protocolo de comunicación seleccionado a nivel instrumentación, en un tiempo apropiado y, si es requerido, ejecutar las acciones de control correspondientes; evitando conducir al proceso a un estado indeseable pudiendo poner en riesgo la integridad del personal, del medio ambiente, de las instalaciones y la calidad de los productos.

Cuando la velocidad de actualización de datos requerida por la variable de proceso, no sea soportada por las tecnologías establecidas, para el nivel de instrumentación que operan sobre un canal de comunicación en topología multiconexión, se debe realizar una conexión punto a punto, entre instrumento y tarjetas de entrada / salida del controlador.

### 1.3.3. CATEGORÍA DE INSTALACIONES.

Con la finalidad de hacer una adecuada selección de los protocolos de comunicación de los sistemas digitales de monitoreo y control en el Nivel Control y en el Nivel Supervisión, se deben identificar los requerimientos de la aplicación, de los enlaces de comunicación entre sistemas de control, y de los protocolos de comunicación. Deben considerarse las necesidades particulares de cada proceso para determinar las características y el desempeño del enlace de comunicación.

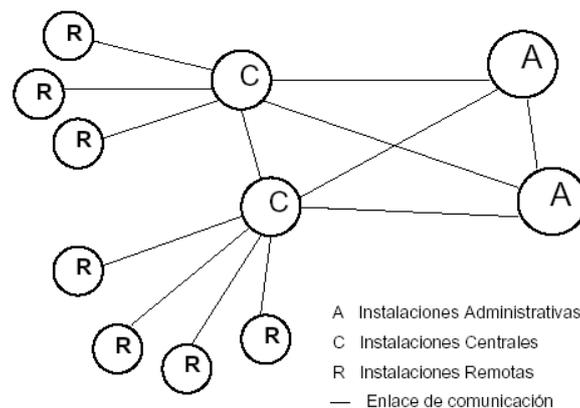
Se establece una clara definición de la categorización de las diferentes instalaciones de proceso que se operan, de acuerdo a su localización geográfica, las cuales se han identificado en tres grupos:

Instalaciones remotas.

Instalaciones centrales.

Instalaciones administrativas.

En la figura 1.5 se muestra la relación entre instalaciones remotas, centrales y administrativas.



**Figura 1.5 Relación entre instalaciones remotas, centrales y administrativas.**

### **1.2.6 SISTEMAS SCADA.<sup>7</sup>**

SCADA viene de las siglas "Supervisory Control And Data Acquisition", es decir: adquisición de datos y control de supervisión. Se trata de una aplicación de hardware software especialmente diseñada para funcionar con PC en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde la pantalla del PC. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros supervisores dentro de la empresa: control de calidad, supervisión, mantenimiento, etc.

Los programas necesarios, y en su caso el hardware adicional que se necesite, se denomina en general sistema SCADA.

### **1.2.7 INTERFACES HMI.**

En el último tiempo, el software de interfaz máquina-operador (HMI) tenía un papel claramente definido en una fábrica: simplemente servía de vínculo entre las máquinas y las personas que las utilizaban, proporcionando diversos elementos de control para visualizar datos de procesos, emitiendo alarmas, registrando datos, aportando animación a los gráficos y representando visualmente el estado de las máquinas y los procesos. Sin embargo, así como las computadoras personales y las nuevas tecnologías de software han transformado el mundo durante la última década, el sector de la fabricación industrial también se ha tambaleado sobre sus cimientos y ha evolucionado rápidamente desde finales de los años ochenta.<sup>8</sup>

Hoy, el centro de atención de los procesos de fabricación ha cambiado del hardware (máquinas) al software, y del software HMI que controla la fabricación discreta y procesos al middleware (software intermedio) capaz de ofrecer soluciones que abarcan

---

<sup>7</sup> <http://personal.redestb.es/efiguera/memoria.htm>

<sup>8</sup> [http://www.rockwellsoftware.com/corporate/softauto/sa99\\_2/hmirise\\_sp.cfm](http://www.rockwellsoftware.com/corporate/softauto/sa99_2/hmirise_sp.cfm)

toda la empresa. Con el aumento y la mayor diversidad de las necesidades, el software HMI ha evolucionado desde una herramienta de visualización en el taller de la fábrica hasta convertirse en una auténtica “interfaz de empresa” que permite capturar, controlar y transmitir datos desde el taller hasta la empresa misma del cliente.

#### **1.2.6. TIEMPO REAL.**

La capacidad en tiempo real se refiere a la capacidad del ordenador en programas de procesamiento de datos para que siempre esté listo para procesar y proporcionar los resultados dentro de un tiempo especificado. En este contexto "estrictamente en tiempo real" significa que un sistema reacciona a los eventos externos dentro de un tiempo especificado en un 100% de los casos. Además si se habla de “tiempo real” el sistema debe responder en tiempos concretos también en un 100% de los casos. Si, de otra forma, los tiempos concretos de reacción pueden superarse en ciertos casos, como en sistemas no críticos, se habla de "tiempo real suave".

#### **1.2.8 LA ESTRUCTURA ABIERTA.**

Aún no se ha establecido un estándar para poseer extensiones compatibles en tiempo real de sistemas operativos. De una forma estrictamente determinante, los sistemas estándar actuales deben ser modificados de forma general, así que la principal ventaja de un sistema basado en PC - su estructura abierta – puede llegar a ser un inconveniente. No obstante, la estructura abierta, permite a la empresa o el desarrollador más libertad en la elección de la herramienta adecuada para el análisis, diseño y programación del sistema SCADA. La solución comienza a ser propietaria nuevamente (cada empresa ofrece su solución) y la conversión a futuras generaciones de sistemas operativos lo hace más difícil.<sup>9</sup>

---

<sup>9</sup> <http://personal.redestb.es/efiguera/memoria.htm>

### 1.3 TECNOLOGÍAS PARA INTEROPERABILIDAD ENTRE APLICACIONES.

Las instalaciones administrativas disponen de procesos para el manejo de información, todos ellos vinculados para lograr un objetivo en común que es lograr que la empresa sea más competitiva. Estos procesos de manejo de información se efectúan en forma local en servidores específicos, tal como servidores de correo, servidores de información para la administración del conocimiento, servidores para la administración del mantenimiento, servidores de información de proceso, servidores para la planificación de recursos de la corporación (ERP), etc., como se muestra en la figura 1.6

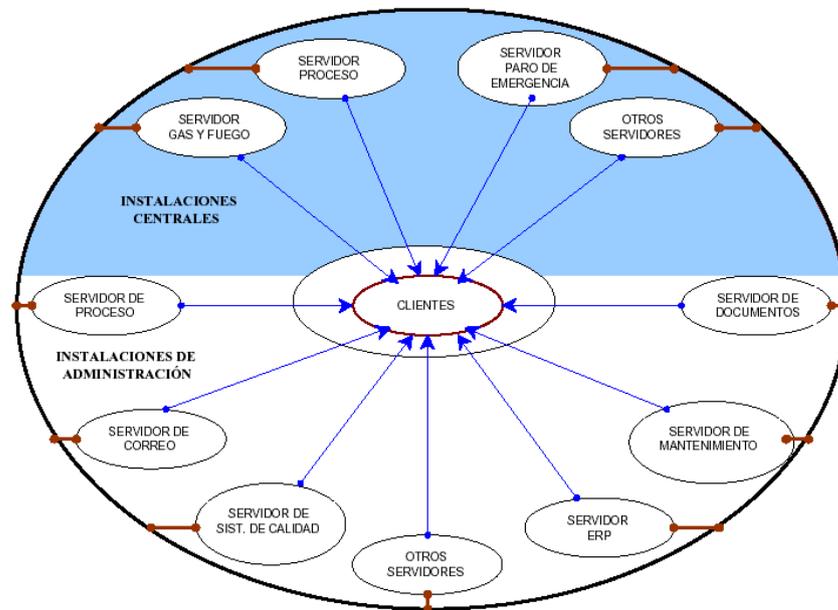


Figura 1.6 Modelo funcional para la intercomunicación.

Las aplicaciones residentes en los servidores de información para las actividades del negocio o las actividades de control del proceso, deben estar vinculadas por medio de tecnologías, algoritmos o protocolos que permitan el intercambio satisfactorio de la información. Para múltiples aplicaciones corriendo bajo un ambiente operativo Windows, en una misma estación de supervisión, se debe utilizar OPC (OLE para Control de Procesos). Cuando una aplicación no soporte OPC, debe emplearse el Intercambio Dinámico de Datos (DDE) para lograr la transferencia de datos de una

aplicación a otra. Dichas aplicaciones deben de soportar Interconectividad de Bases de Datos Abierta (ODBC).

Independientemente del sistema operativo, cuando se requiera transferir datos entre aplicaciones residentes en distintos servidores o estaciones de supervisión, debe emplearse el estándar OPC (cliente, servidor o ambos, según sea requerido).

### **1.3.1 OLE (OBJETOS VINCULADOS E INSERTADOS).<sup>10</sup>**

En el mundo de los objetos vinculados e insertados, un objeto es cualquier cosa que un usuario pueda manipular con una aplicación Windows. Los programas que acompañan a los sistemas operativos Windows son un buen ejemplo de aplicaciones que manipulan objetos OLE; por ejemplo, el programa PaintBrush maneja objetos que son mapas de bits y la Grabadora de Sonidos objetos de sonido. Aunque distintos, las aplicaciones OLE utilizan métodos similares para ambos tipos de objeto.

Vincular e insertar son los dos métodos con los que los usuarios manejan los objetos OLE. Los objetos pueden estar vinculados o insertados en el documento de una aplicación cliente OLE. Un objeto insertado es una parte física del documento. Uno vinculado está separado de aquel. Una aplicación cliente puede editar uno de estos objetos llamando a la aplicación servidora OLE como si fuera un editor.

### **1.3.3 DDE (TRANSFERENCIA DINÁMICA DE DATOS).**

En Windows cualquier aplicación cliente puede comunicarse con cualquier servidor, usando el protocolo DDE. La transferencia dinámica de datos (DDE ) es la técnica de comunicación que utilizan los programas. La comunicación entre procedimientos es el

---

<sup>10</sup> [http://diea-gw.diea.ulpgc.es/users/aurelio/int\\_equipos/trab9899/ole\\_dde\\_opc/index.html](http://diea-gw.diea.ulpgc.es/users/aurelio/int_equipos/trab9899/ole_dde_opc/index.html)

paso de información a través de una memoria común siguiendo las reglas de un protocolo que lo sincroniza. También intervienen en el proceso el portapapeles (clipboard), las bibliotecas para vínculos dinámicos (DLL) y los objetos vinculados e insertados (OLE).

El portapapeles es el conjunto de funciones que se apoyan en la memoria global para transferir datos entre aplicaciones. No hay que confundir esta memoria global con la utilizada por el programa Portapapeles (CLIPBRD.EXE), aplicación que visualiza los datos depositados en la memoria global. La tarea básica del portapapeles es la transferencia de datos entre aplicaciones.

Una DLL es un módulo que contiene código, datos y recursos de Windows al que pueden acceder concurrentemente varios programas. De forma que aunque varios pueden compartir la misma instancia de una DLL, solamente puede existir simultáneamente una instancia de la misma DLL. Una aplicación y una DLL acceden a un bloque de memoria global para compartir los datos.

El proceso comienza cuando la aplicación llama a una de las funciones de la DLL, momento en que ésta reserva el bloque de memoria global que ambas utilizarán durante la transferencia. Cuando la aplicación que llamó termina, comunica el hecho a la DLL para que lo desasigne. La reserva de memoria global dura pues, lo que la aplicación que llamó a la DLL. Es por esto por lo que dos aplicaciones no pueden acceder al mismo bloque de memoria global que reservó una DLL.

Las aplicaciones DDE utilizan un sistema de identificación que tiene una estructura de árbol. El nombre de aplicación es el nivel más alto de la jerarquía y define a la aplicación servidor. Cada nombre de aplicación controla uno o varios temas, identificados por sus nombres de tema (topic name), nombres que también se refieren a la aplicación servidor que, a su vez, tienen uno o más nombres de identificador (item name), con los detalles de cada tema.

### 1.4.3 OPC (OLE PARA EL CONTROL DE PROCESOS).

En cooperación con Microsoft, una fuerza constituida por cinco empresas, Intellution, Opto-22, Fisher-Rosemount, Rockwell Software y Intuitiv Software, nace OPC Foundation en Mayo de 1995.

El OPC, (OLE para el Control de Procesos) “es una especificación técnica no propietaria definida por la entidad OPC Foundation<sup>11</sup> y consiste en un sistema de interfaces estándar basado en OLE/COM de Microsoft”; con OPC es posible interoperar dispositivos industriales con sistemas de información o aplicativos de escritorio. En otras palabras, el OPC permite desarrollar de una manera muy práctica y eficiente aplicaciones que pretendan comunicarse con equipos industriales controlados por PLCs.

OPC ofrece varias ventajas las cuales fueron citadas por OPC Foundation en su OPC Overview<sup>12</sup>; se destacan las siguientes:

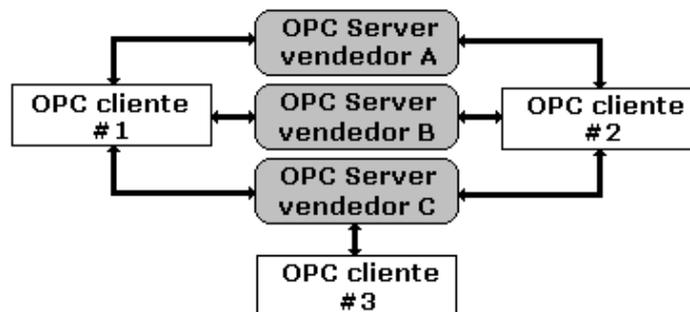


Figura 1.7 Relación entre Cliente/Servidor OPC.

Los fabricantes de hardware tienen que hacer solamente un conjunto de componentes de software para que los clientes los utilicen en sus aplicaciones.

<sup>11</sup> <http://www.opcfoundation.org>

<sup>12</sup> OPC Foundation. OPC Overview: OPC DataAccess Overview. p. 11

Los desarrolladores de software no tienen que reescribir drivers debido a cambios en características o adiciones en un hardware.

Los clientes tendrán más opciones con las cuales puedan desarrollar diversos sistemas de aplicación a nivel industrial.

Requerimientos de funcionalidad de Acceso a datos de una interface estándar de OPC 2.01:

OPC es soportado completamente por VC++, Visual Basic y Delphi.

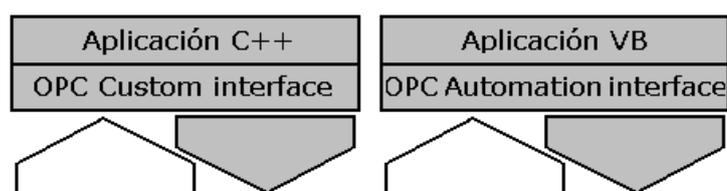
Cualquier cliente con interfaz OLE con ciertas limitaciones.

No soporta el uso con VBScript o JavaScript.

La especificación OPC requiere como Sistema Operativo Windows 95/98 (con DCOM), Windows NT 4.0 o Superior. En todos los casos es recomendable instalar la última versión de Services Pack correspondiente.

#### 1.4.3.1 FUNCIONAMIENTO DE OPC.

Un cliente OPC puede conectarse a servidores OPC, de uno o varios vendedores. Se puede construir un cliente con una interfase personalizada, para lo cual se puede usar un lenguaje de alto nivel como Visual C++, pero los clientes más comunes se construyen bajo una interfase automatizada que puede ser desarrollada en lenguajes como Visual Basic 6.0, Delphi y recientemente .NET gracias a COM-Interop. La Figura 1.8 representa el funcionamiento del OPC con las interfaces personalizada y automatizada.



**Figura 1.8 Funcionamiento e Interfaces de OPC.**

#### **1.4.3.2 MODELO DE OBJETOS DE LA ESPECIFICACIÓN OPC.**

El modelo jerárquico de objetos definido por OPC Foundation se representa en la figura.1.9.

OPCServer. Es una instancia de un servidor OPC. Se debe crear un objeto OPCServer antes de poder referenciar los otros objetos. Este contiene la colección OPCGroups y el objeto OPCBrowser.

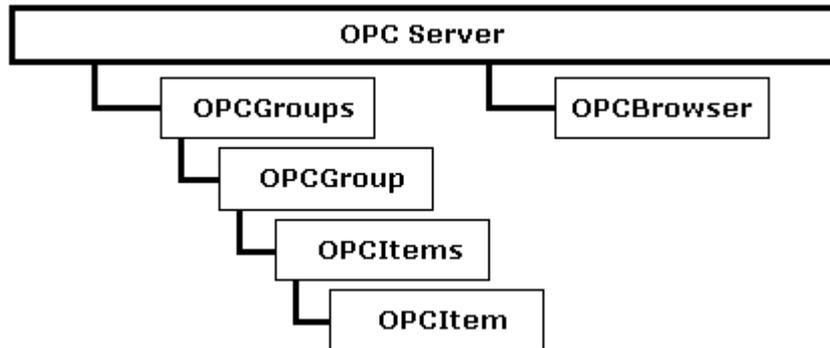
OPCGroups. Es una colección de los objetos OPCGroup que el cliente ha creado.

OPCGroup. El propósito de este objeto es mantener la información de estado y proveer el mecanismo para ofrecer los servicios de adquisición de datos por la colección de objetos OPCItem.

OPCItems. Es una colección que contiene todos los objetos OPCItem que el cliente ha creado.

OPCItem. Es un objeto que mantiene la definición de los ítems, sus valores, estados y datos de la última actualización.

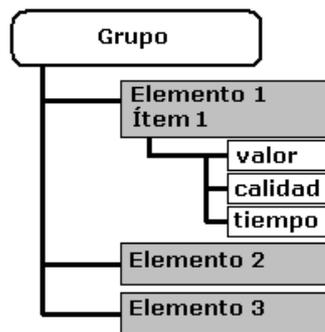
OPCBrowser. Es un objeto que permite buscar nombres de ítems en un servidor configurado.



**Figura 1.9 Modelo de Objetos del Servidor de Automatización OPC.**

Un servidor de acceso a datos OPC está formado por varios objetos: el servidor, el grupo y el elemento. “El servidor de objetos OPC ofrece información sobre el servidor y sirve como un contenedor de grupos de objetos OPC. El grupo de objetos OPC mantiene información acerca de sí mismo y proporciona los mecanismos para contener y organizar lógicamente los elementos OPC”; los grupos OPC proporcionan una forma para organizar los datos de los clientes, por ejemplo, el grupo podría representar los elementos en un pantalla particular del operador o a través de un informe; los datos pueden ser leídos y escritos, y las conexiones basadas en excepciones, pueden ser creadas entre el cliente y los elementos en el grupo y pueden ser activadas y desactivadas según sea necesario; un cliente OPC puede configurar que porcentaje de los datos deben ser cambiados antes de la actualización.

Hay dos tipos de grupos, públicos y locales (o privados); los públicos se realizan para ser compartidos entre varios clientes, mientras que los locales son privados para el cliente en cuestión. Existen interfaces específicas opcionales para los grupos públicos; dentro de cada grupo, el cliente puede definir uno o más elementos OPC, la figura 1.10 ilustra esta relación:



**Figura 1.10 Relaciones entre Grupos e Ítems.**

Los elementos OPC representan conexiones a fuentes de datos dentro del servidor; un elemento OPC, no es accesible por el cliente como un objeto. De esta forma, no hay una interfaz externa definida para un elemento OPC; todos los accesos al elemento OPC se realizan a través del objeto grupo OPC que contiene el elemento OPC, o simplemente el grupo en el que el elemento ha sido definido.

Asociado a cada elemento (ítem), existe: el valor (tag), calidad de la transacción y el valor de tiempo en que se actualizo el elemento. “Los elementos no son las fuentes de datos, sólo son conexiones a ellas; el elemento OPC debe ser entendido como la dirección de los datos, no como la fuente física actual de los datos a los que la dirección referencia”, puesto que la fuente real de los datos es el dispositivo controlador, regularmente un PLC.

## **1.5 ARQUITECTURA DE PIC16FXXX DE MICROCHIP<sup>13</sup>**

Microchip ha agrupado sus modelos de microcontroladores (MCU) de acuerdo a la longitud de la palabra del procesador. La línea 16XXXX constituye la línea media de los MCU de microchip, con una longitud de palabra del procesador de 14-bits, 8-bits

<sup>13</sup> CATANIA, Carlos, TCP/IP sobre sistemas embebidos y su implantación sobre la arquitectura PIC, Universidad Marcelino Champagnat , 2005

para la memoria de datos, velocidades de hasta 20Mhz ofrecen una excelente relación precio/rendimiento la cual los ha hecho una elección común a la hora de trabajar con MCU de 8bits, las características más relevantes se mencionan a continuación.

## 1.5.1 EL PROCESADOR.

### 1.5.1.1 LA ARQUITECTURA HARVARD.

Cuenta con la memoria de programa y la memoria de datos separadas, y solo accesibles a través de buses distintos. El PIC16FXXX cuenta con un bus de datos de 14-bits para acceder a la memoria de programas y un bus independiente de 8-bits para acceder a la memoria de datos.

Esto mejora el ancho de banda en relación a la arquitectura tradicional de Von-Neumann, en la cual memoria de programa y la memoria de datos son accedidas a través del mismo bus.

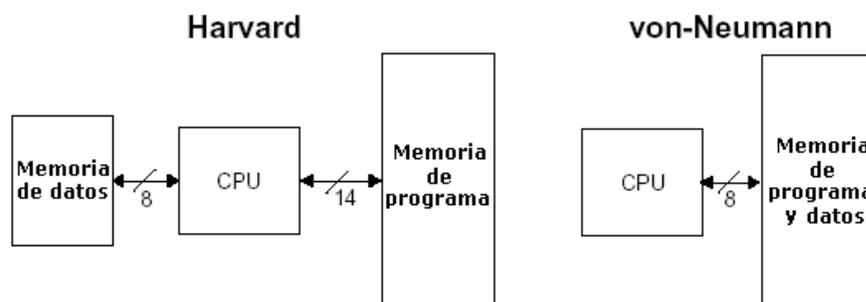


Figura 1.11 Arquitectura de Microcontroladores.

Esta arquitectura ofrece la posibilidad de poder acceder a una sola instrucción en un ciclo de reloj. Mientras la memoria de programa es accedida, la memoria de datos está

en un bus independiente y puede ser leída y escrita. Esta separación de buses permite que una instrucción sea ejecutada mientras la siguiente es extraída.

#### **1.5.1.2 SEGMENTACIÓN DEL PROCESADOR.**

Consiste en descomponer en etapas el procesamiento de cada instrucción solapando cada una de éstas y así lograr un paralelismo implícito. En el caso de la arquitectura 16XXXX es de 2 etapas en la cual se solapan la extracción y ejecución de cada instrucción. La extracción de una instrucción demora tres ciclos (TCy3) y la ejecución demora cuatro ciclos (TCy4).

Si embargo debido a que este solapado en la extracción y ejecución de la instrucción previa, una instrucción es extraída y otra es ejecutada cada ciclo de máquina (TCy).

#### **1.5.1.3 INSTRUCCIONES REDUCIDAS.**

Como su arquitectura RISC lo indica, cuenta con un conjunto reducido de 35 palabras que permiten realizar todas las tareas. Esto también permite un rápido aprendizaje.

#### **1.5.1.4 INSTRUCCIONES SIMÉTRICAS.**

Es posible realizar cualquier operación sobre cualquier registro utilizando cualquier método de direccionamiento. Esta simetría junto con la ausencia de instrucciones especiales permite realizar una programación eficiente y reduce la curva de aprendizaje.

## 1.5.2 ORGANIZACIÓN DE LA MEMORIA.

### 1.5.2.1 MEMORIA DE PROGRAMA.

La arquitectura de los 16XXX cuenta con contador de programa de 13bits el cual es capaz de direccionar hasta 8K x 14 palabras. Dependiendo del modelo de MCU esos 8Kx14 palabras estarán físicamente implementados o no. Normalmente la memoria de programa puede ser de tipo EEPROM o FLASH.

Cuentan con una pila que permite hasta 8 niveles de profundidad. Sin embargo ésta no se encuentra en la memoria de programa, ni en la memoria de datos por lo que no es accesible ni para lectura ni para escritura.

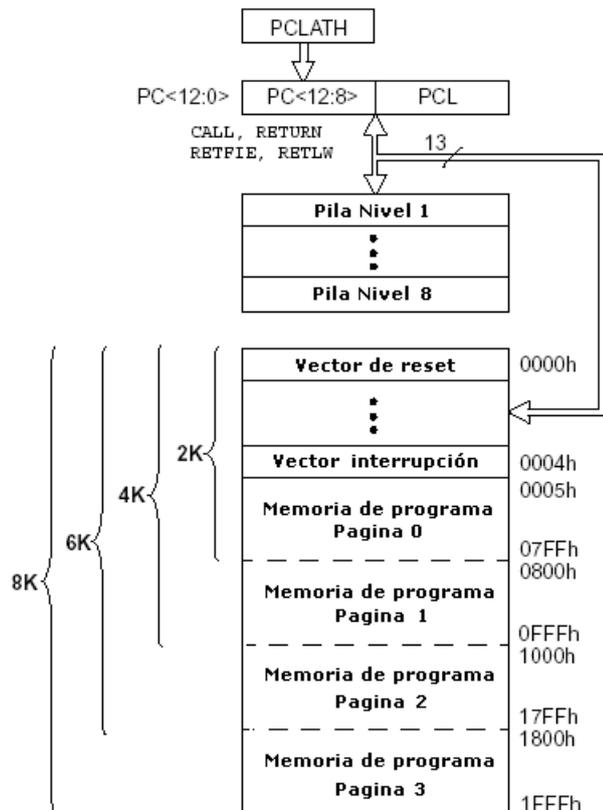
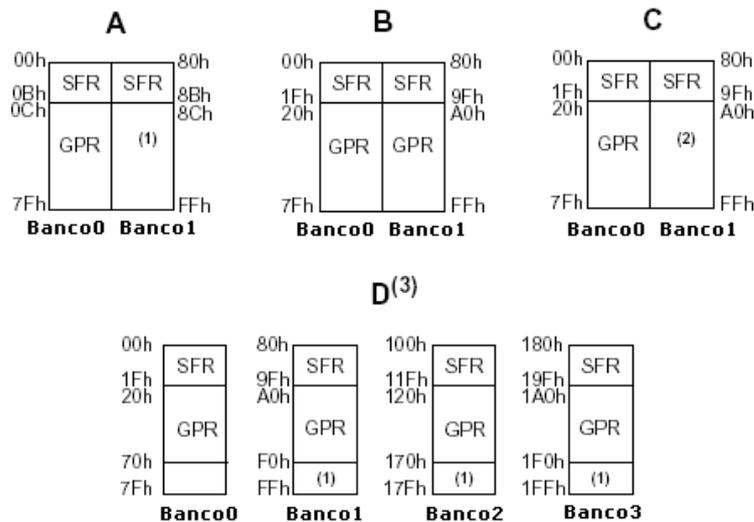


Figura 1.12 Memoria de programa de PIC16FXXX.

### 1.5.2.2 MEMORIA DE DATOS.

La memoria de datos está dividida en múltiples bancos, los cuales contienen registros especiales (SFR) y los registros de propósito general (GPR) que pueden ser usados para el almacenamiento de los datos del programa. Estos registros son de 8 bits. Cada banco está compuesto por 128 bytes.

En la parte inferior de éstos, se encuentran los SFR y arriba de éstos se encuentran los GPR. Los SFR son utilizados por el CPU y los módulos periféricos del MCU para controlar los distintos tipos de operación del MCU.



**Nota 1: Mapeado (Copia) en el Banco0**  
**2: No implementado, se lee como 0**  
**3: Algunos dispositivos tienen el GPR ubicado en el FSR**

Figura 1.13 Memoria de datos de PIC16FXXX.

A veces por cuestión de comodidad, algunos registros SFR son duplicados en todos los bancos, para facilitar la realización de algunas operaciones.

### 1.5.2.3 MEMORIA EEPROM.

Los MCU de la familia 16XXXX cuentan con una cierta cantidad de memoria EEPROM dependiendo del modelo. La cual permite guardar datos de manera no volátil. Dependiendo del modelo del MCU pueden llegar a ser de hasta 256 bytes.

**Tabla 1.2 Memoria de EEPROM de PIC16FXXX.**

<b>Memoria EEPROM Tamaño</b>	<b>Rango de Dirección</b>
64	0h - 3Fh
128	0h - 7Fh
256	0h - FFh

### 1.5.3 LÍNEAS DE ENTRADA Y SALIDA.

Las líneas de entrada y salida generales pueden considerarse los periféricos más simples con los que cuentan los MCU PIC16XXXX. Estas líneas de E/S le permiten monitorizar y controlar otros dispositivos. Cada una de las patitas del MCU se agrupa para formar lo que se denominan puertos. Para la mayoría de estos puertos, las patitas pueden configurarse para funcionar como líneas de E/S de acuerdo a las necesidades de cada caso a través de un SFR llamado TRIS<X> el cual se encargaría de establecer la dirección del PUERTO<X>.

Para agregar funcionalidad y flexibilidad muchas de las patitas del los MCU están multiplexadas con funciones alternativas. Mediante diversos SFR se pueden activar o desactivar dichas funciones. En esos casos la funcionalidad de los pines de E/S cambiará para ajustarse al nuevo módulo.

## **CAPÍTULO 2**

### **DISEÑO DEL PROTOCOLO DE COMUNICACIÓN.**

#### **2.8 CARACTERÍSTICAS DEL HARDWARE DE RED.**

Para seleccionar el hardware adecuado, que se adapte al medio, al mercado nacional y a las principales características de los sistemas de comunicación basados en PLC's de clase media y baja, se analizó las principales características de funcionamiento de hardware y software de los protocolos: Siemens (protocolo PPI), Omron (protocolo Host Link), y Modicom (protocolo Modbus).

Las características analizadas en los protocolos antes mencionados fueron: método de acceso al medio, tipo de transceivers utilizados para transmisión y recepción, topología de red, características de las interfaces del hardware, medio de comunicación, tipo de servidor de datos para aplicaciones clientes HMI, características de las tramas, complejidad de implementación, información existente, chequeo de error, requerimientos de hardware esclavo y requerimientos del hardware maestro.

Del análisis de las características antes mencionadas, se propone el hardware y el software capaces de monitorear a un microcontrolador. A continuación se indican las características más importantes.

### 2.8.1 MÉTODO DE ACCESO AL MEDIO Y TOPOLOGÍA DE RED.

El método de acceso al medio es una derivación de Polling, conocido como maestro esclavo. En el cual se define como estación maestra a un sistema capaz de monitorear a la red utilizando un servidor de datos OPC. Los esclavos están definidos por las estaciones que responderán solo cuando el maestro realice una consulta a dicha estación.

El hardware capaz de permitir el monitoreo de las estaciones será implementado en una topología física de red tipo bus como se indica en la figura 2.1. De ello se muestra una instalación multipunto (conexión de varias estaciones en una red con topología bus).

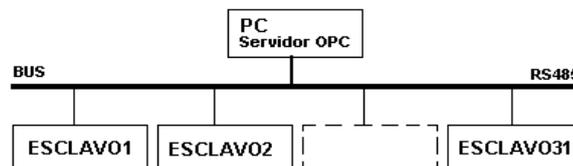


Figura 2.1 Topología física tipo Bus.

### 2.8.2 PROTOCOLO DE COMUNICACIÓN RS-232.

El protocolo de comunicación está basado en datos binarios, lo cual implica la utilización de 8 bits de datos para el protocolo RS-232, operando como Half-Duplex.

El protocolo está formado por un bit de inicio, 8 bits de datos y uno de parada, no se utiliza ningún tipo de control de flujo y la transferencia de información es 115200 bits por segundo.

### 2.8.3 TRANSCEIVERS Y CARACTERÍSTICAS DE RS-485.

En el mercado nacional es muy sencillo obtener circuitos transceivers TTL-RS232, TTL-RS422, TTL-RS485, TTL-RF y USB. El bajo costo y las características de los circuitos integrados basados en RS-485, permite realizar la implementación de la topología bus con un máximo de 32 estaciones incluida la estación maestra, esto en aplicaciones industriales. Con el objetivo de minimizar el cableado se utilizará redes tipo Half-Duplex. A continuación se mencionan las características generales del circuito MAX485.

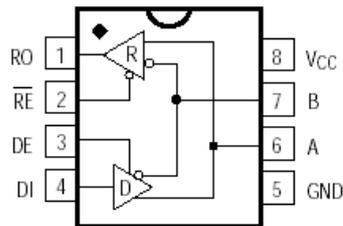


Figura 2.2 Disposición de pines de MAX485.

- Voltaje de modo común extendido (+12v a -7v).
- Transmisión multipunto.
- Compatible con RS422.
- Contención (Apagado) en situaciones de alto consumo de energía.
- Rango de voltaje 4.25 a 5.25V de la fuente de energía simple.
- Retardo de transmisión 22ns.
- Apagado automático como protección térmica.

- Disposición de pines (figura 2.2) compatible con MAX485, DS3695 y DS75176B.
- Combinación de impedancias de transmisión y recepción para 32 transceivers.
- Histéresis típica de recepción 70mV.
- Velocidad de transmisión 2.5Mbps a 1200mts.
- Diseño para aplicaciones Half Duplex.
- Bajo consumo en transceivers RS485.
- Transceiver para aplicaciones sensibles a EMI (Interferencias Electromagnéticas).
- Aplicaciones de control industrial en redes de área local.
- Tiempo de apagado automático 200ns.
- Resistencia de entrada del Transceiver 12K Ohmios.

#### **2.8.4 MEDIO DE COMUNICACIÓN.**

La selección del cable está determinada por la velocidad de comunicación, la codificación de la información, la distancia de las estaciones, el grosor del cable, el blindaje y la calidad del cable (atenuación del cable).

Es necesario incrementar un par de cables en cada corrida con el objetivo de asegurar la señal de masa (ground).<sup>14</sup>

El tipo de cable debe ser par trenzado con blindaje STP Categoría 5 como por ejemplo Par Trenzado Industrial (ITP), Belden 9841-9844 o Alpha 6222-6230.

---

<sup>14</sup> BALCELLS, Joseph, Interferencias Electromagnéticas En Sistemas Electrónicos, Marcombo, México,1992

## **2.8.5 REQUERIMIENTOS DE LOS NODOS.**

### **2.8.5.1 NODO ESCLAVO.**

Cada nodo esclavo requiere:

- Memoria RAM: 135 localidades de 8 bits.
- Memoria EEPROM: 3 localidades de 8 bits.
- Memoria de programa (593 palabras de 14bits “PIC16F628A”).
- Periférico UART configurable para 115200bps.
- Un puerto de salida 1 bit (control de transceiver RS485).
- Varios software de desarrollo de aplicaciones de microcontroladores.
- Adaptador de interfase (RS232/RS485).

### **2.8.5.2 NODO MAESTRO.**

El nodo maestro requiere:

- Software servidor de datos OPC del protocolo desarrollado.
- Sistema Operativo: Windows XP.
- Procesador: Pentium 4 o superior.
- Memoria RAM: 8 megabytes.
- Monitor: SVGA.

- Puerto serie (COM).

### 2.8.6 TIPO DE SERVIDOR DE DATOS.

El servidor de datos seleccionado es: OPC (OLE para Control de Procesos), de “Foundation OPC” el cual se describe en el capítulo 1. El intercambio de datos entre las aplicaciones está basado en la biblioteca: WTOPCSrv.dll para desarrollo rápido de servidores de datos OPC, de la empresa: WinTECH. WTOPCSrv.dll es una biblioteca que tiene costo.

### 2.8.7 COMPARACIÓN CON EL MODELO OSI.

La anatomía del protocolo de comunicación comparado con el modelo OSI (figura 2.3) emplea las capas de aplicación, enlace de datos y física.

Capa de Aplicación: En el caso del maestro, sirve datos a las aplicaciones clientes (Microsoft Visual Basic, iFIX, LabView, RSView, WinCC, LokOut, InTouch 9, entre otros), los datos son provenientes del controlador. En el caso del esclavo los datos recibidos del maestro son utilizados en los “periféricos” del controlador (puertos, temporizadores, memoria y otros).

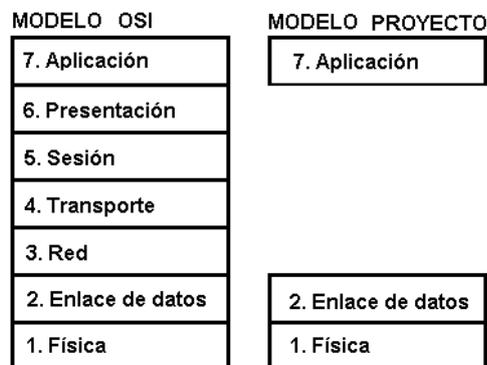
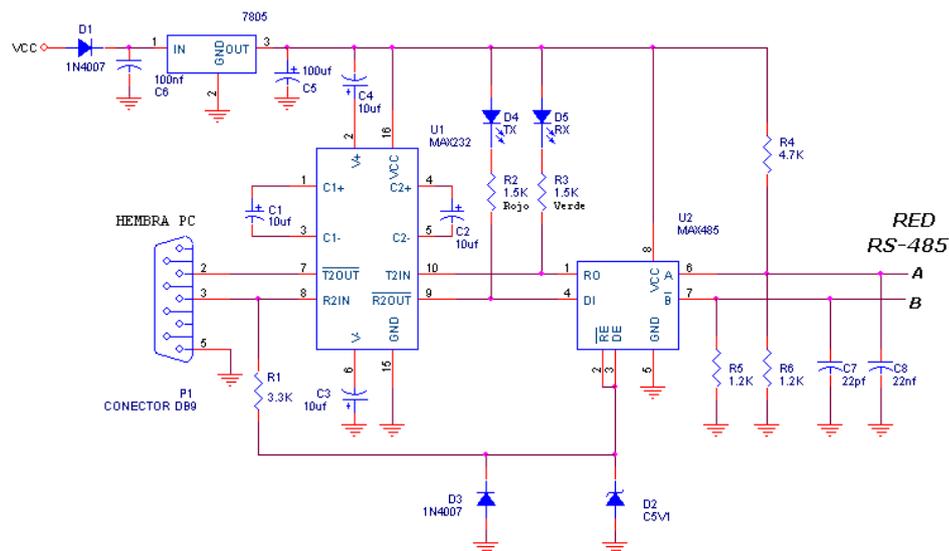


Figura 2.3 Comparación del modelo OSI con el modelo del proyecto.

La capa de Enlace de Datos, puede recibir y transmitir las tramas, para ello el software es capaz de encapsular la información en tramas, añadir la cabeza (header) y la cola (trailer), y de realizar el proceso contrario, en su determinado momento.

La capa Física convierte la información lógica en impulsos eléctricos. Por ejemplo, las especificaciones eléctricas: TTL, RS-232, RS-485, la topología física de la red, tipo de conectores, repetidores, cables y otros elementos físicos utilizados en el proceso de comunicación de los datos.

## 2.9 DISEÑO DEL HARDWARE ADAPTADOR DE INTERFASE.



Para poder conectar los nodos esclavos en la red, es necesario poseer la red RS485, para esto se convierte la señal de la interfase RS232 de la PC a TTL y de TTL a RS485, como indica el diagrama eléctrico de la figura 2.4. El control de consultas es realizado por el maestro (PC) por ello se acondiciona el voltaje de transmisión para el control del transceiver RS485 con R1, D2 y D3. C1, C2, C3, C4<sup>15</sup> son encargados de posibilitar los voltajes de la interfase RS232. R4, R5 y R6 son resistencias de polarización de red, que permiten mantener a la red en un voltaje conocido. C7 y C8 deben ser capacitores de 100pf<sup>16</sup> los cuales son recomendados por los fabricantes del circuito integrado.

<sup>15</sup> Maxim, MAX220 – MAX249, www.maxim-ic.com, 1996

<sup>16</sup> Maxim, RS422/RS485 Transceivers, www.maxim-ic.com, 1996

La señal de la interfase RS485 tiene una lógica inversa a la información representada en el software del nodo maestro, la misma es invertida en el UART y posteriormente se invierte dos veces en el adaptador de interfase RS232/RS485, con relación al voltaje VA referido a masa, figura 2.5.

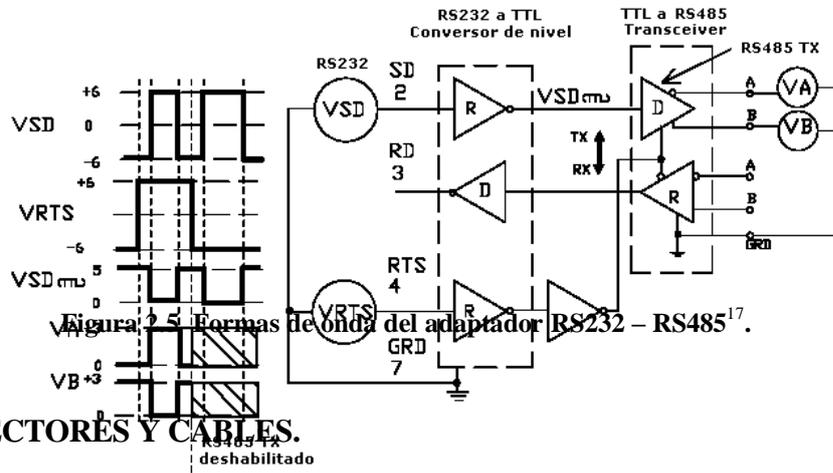


Figura 2.5 Formas de onda del adaptador RS232 – RS485<sup>17</sup>.

## 2.10 CONECTORES Y CABLES.

El conector al PC y el conector al adaptador de interfase son de tipo DB9 hembra, el cable es 26AWG para transmisión de datos, las señales son de tipo RS232 y la longitud máxima del cable es 15 metros, como lo indica la figura 2.6.

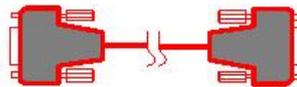


Figura 2.6

DB9 HEMBRA		
Colores	Señal	Pines
VERDE	RD	2
ROJO	TD	3
NEGRO	GND	5

or RS232.

El cable de red RS485 es UTP CAT5 con conectores RJ11, la conexión que se indica en la figura 2.7, también permite energizar los nodos esclavos. La longitud máxima del cable es 1200 metros.



RJ11 MACHO

Señal	Pines
	VCC
2	B
3	A
4	GND

<sup>17</sup> B&B Electronics, RS422 and RS485 Application Note, www.bb-elec.com, 1997

Figura 2.7 Cable y conector de red RS485.

## 2.11 DISEÑO DEL HARDWARE DE ESTACIONES ESCLAVAS.

Aplicando los requerimientos del nodo esclavo a los microcontroladores PIC16F877, PIC16F877A y PIC16F628A y descargando el software desarrollado para el protocolo propuesto, se demuestra el funcionamiento del hardware implementado en la figura 2.8, para 7 nodos esclavos.

El control del transceiver MAX485 es realizado solo cuando el controlador va a transmitir y por ello necesita un bit de salida de control de un puerto, la lógica de control es directa (nivel alto permite la transmisión del esclavo).

La lógica TTL en la comunicación serie es inversa y realizada por el periférico UART. El sistema tiene una conexión directa con la red, lo cual no ofrece aislamiento para descargas eléctricas.

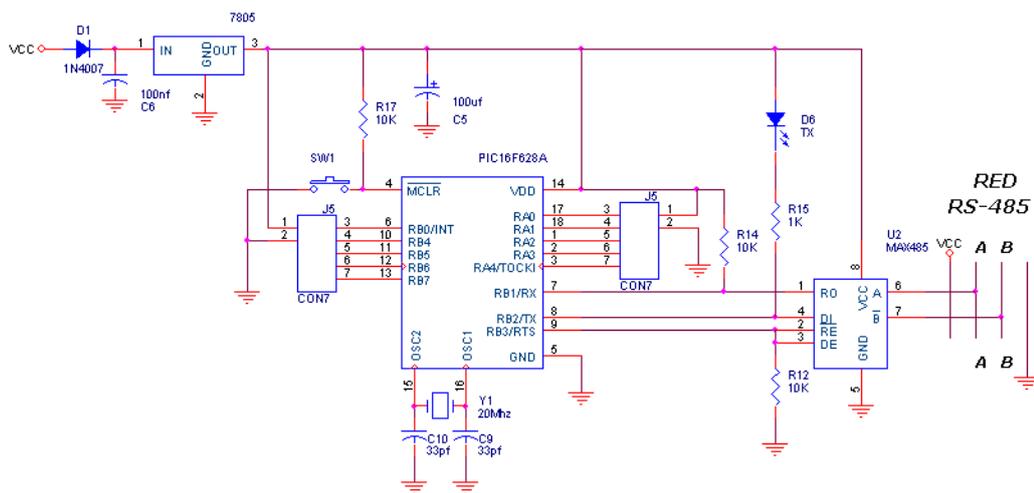


Figura 2.8 Hardware implementado para el nodo esclavo.

## 2.12 CONSIDERACIONES DE DISEÑO Y CARACTERÍSTICAS DE LA TRAMA.

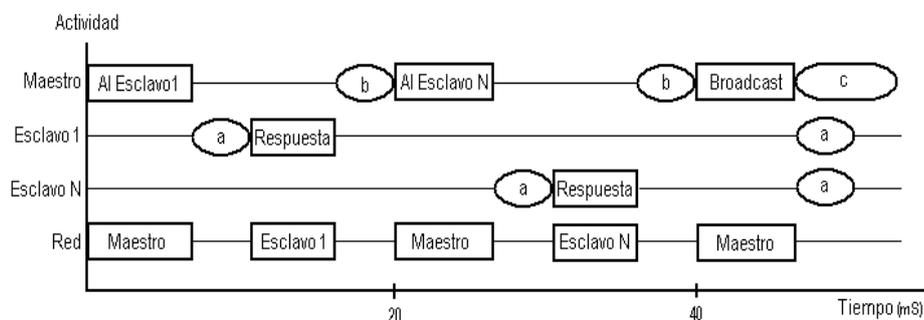
Un protocolo de comunicación debe poseer las características fundamentales de comunicación, las mismas que deben conocer cada uno de los integrantes de las conversaciones (nodos).

Lógicamente se ha definido la integración de ráfagas de bytes para la formación de la trama, toda la información de la trama se transmitirá por bytes encapsulados, manteniendo el protocolo de comunicación RS232 con: un bit de inicio, ocho bits de datos y un bit de parada. El hardware (UART) debe tener la capacidad de reconocer los bytes encapsulados de la trama que se encuentran en el bus.

Cada nodo de la red debe ser capaz de saber cuando se inicia cualquier transmisión de una trama, reconocer si la trama le ha sido enviado a él, el objetivo de la trama, la información que contiene la trama, verificar la veracidad de la información de la trama y la capacidad de responder si éste lo tuviera que hacer.

Las ráfagas de bytes que tienen características similares, permiten el reconocimiento de la trama por los diferentes nodos, estas ráfagas son conocidas como campos de la trama, los campos desarrollados para poder realizar las comunicaciones en el proyecto propuesto son: inicio de trama (3 bytes), dirección de destino de la trama (1 byte), control de trama (1 byte), localidad de inicio de consulta o escritura en memoria del controlador (1 byte), longitud de los datos (1 byte), datos a transmitir (n bytes) y codificación de la información para el análisis de errores (1 byte).

Las características físicas de los diferentes MCU existentes en el mercado, dan la posibilidad de definir las características lógicas de cada campo de la trama, las mismas que permiten que el protocolo propuesto como proyecto se adapte fácilmente al hardware de los microcontroladores existentes.



### **Figura 2.9 Diagrama de actividad y tiempo de nodos y de red.**

El método de acceso al medio maestro esclavo, define que el nodo maestro consulta al esclavo y éste solo responde (transmite) al maestro dependiendo del objetivo de la consulta realizada por el maestro. De ello, la trama transmitida por el maestro es más compleja que la trama transmitida por el esclavo, ya que solo el maestro analizará la información transmitida por un esclavo.

En la figura 2.9 se puede observar el método de acceso a la red maestro esclavo con la actividad de la red, el maestro y dos esclavos.

### **2.13 DESCRIPCIÓN DE CAMPOS DE LA TRAMA.**

El tipo de codificación de la información es binaria, con el objeto de aumentar su velocidad y representación numérica.

El protocolo define dos tipos de tramas de transmisión, una para el nodo maestro y otra para el nodo esclavo, las mismas que permitirán transmitir información de datos tipo booleanos, entero de 8 bits, entero de 16 bits y mensaje.

Los campos de la trama de transmisión del maestro son: inicio de trama, dirección, control, localidad de inicio, longitud, datos y control de error, como lo muestra la figura 2.10.



**Figura 2.10 Trama de transmisión del nodo maestro.**

Los campos de la trama de transmisión del nodo esclavo son: inicio de trama, dirección, longitud, datos y control de error, como lo indica la figura 2.11



**Figura 2.11 Trama de transmisión de nodo esclavo.**

### 2.13.1 INICIO DE TRAMA.

Permite identificar el inicio de una trama y está formado por tres bytes y cada byte está representado por el número decimal 170 o en código hexadecimal AA (figura 2.12).



**Figura 2.12 Bits que representan un byte del campo de inicio de trama.**

### 2.13.2 CAMPO DE DIRECCIÓN.

Es el campo que permite identificar a que nodo corresponde la trama o de cual procede la trama en caso de ser una respuesta de un cliente hacia el maestro. Está formado por un byte, el cuál permite un direccionamiento lógico de hasta 253 estaciones, de 1 a 169 y de 171 a 255 para consultas (Polling). Las direcciones 0 y 170 se encuentran

reservadas. La dirección 0 permite realizar consultas de difusión sin respuesta (Broadcast) y la dirección 170 no se recomienda su utilización con el objeto de evitar conflictos con el inicio de trama.

El número máximo de nodos a utilizar en la red dependerá del tipo de hardware de red a utilizar, en el proyecto propuesto permitirá un direccionamiento físico de hasta 32 estaciones.

### 2.13.3 CAMPO DE CONTROL.

El proyecto se fundamenta en el monitoreo del hardware basado en microcontroladores, el campo de control permite establecer el objetivo de la transmisión. La figura 2.13 describe los 8 bits del campo de control.



**Figura 2.13 Campo de control.**

- FSREE. Si esta bandera es un 1 lógico, el tipo de transacción de la trama será realizado en la EEPROM y caso contrario en la memoria RAM.
- RWf. Si es un 1 lógico es una transacción de lectura (lee en el MCU), si es un 0 lógico será una transacción de escritura.
- RB0f, RB1f, RB2f y RB3f. Configuran el banco de la memoria RAM en donde se realizará la transacción.
- Los Bit5 y Bit4. Son banderas reservadas para futuras implementaciones.

El campo de control de trama está representado por un byte y solo es transmitido por el nodo maestro, el esclavo analiza el contenido de este campo para determinar el objetivo de la trama.

#### **2.13.4 CAMPO DE INICIO DE LOCALIDAD.**

Es un campo formado por un byte el cual permite ingresar de la localidad 0 hasta la 255 para indicar el inicio de la transacción en un nodo esclavo y se ayuda de los bits RB0, RB1, RB2 y RB3 del campo de control, para barrer hasta 4Kbytes de memoria. El inicio de la transacción depende del tamaño físico disponible en el banco de memoria del controlador. Este campo solo se implementa en las tramas de transmisión del maestro.

#### **2.13.5 CAMPO DE LONGITUD.**

Es un campo de un byte, el cual permite identificar la longitud de los datos binarios a transmitir en cada estación, esta longitud depende del tamaño del banco de memoria del modelo del microcontrolador y de la localidad de inicio de trama.

#### **2.13.6 CAMPO DE DATOS.**

El tamaño de este campo está determinado por el número especificado en el campo de longitud, su tamaño varía de 0 a 255 bytes.

#### **2.13.7 CAMPO DE CHEQUEO DE ERROR.**

El tamaño del campo de chequeo de error es de un byte y utiliza el chequeo CRC8. Si el CRC calculado coincide con el número 170 en decimal (inicio de trama), se suma uno al dato calculado y este dato es el nuevo CRC.

## **2.14 TIPOS DE TRAMAS.**

A continuación se explica los tipos de tramas utilizados por el protocolo propuesto.

### **2.14.1 TRAMAS NORMALES.**

Este tipo de trama esta orientado al mantenimiento y configuración del microcontrolador. Son tramas de lectura o escritura en la memoria RAM o EEPROM. Las transacciones recibidas en el nodo esclavo se analizan en un bloque de memoria temporal, si la transacción es correcta se copia la información en las localidades de memoria del trabajo del microcontrolador. Este tipo de tramas son distintas a las tramas especiales de: consulta y difusión, las que se explican más adelante.

Cada transacción que realice el maestro al esclavo, el esclavo responde con: el campo de inicio, dirección del dispositivo, longitud de datos y el campo de chequeo de errores. En el caso de una trama de escritura, el esclavo responde un eco de los datos escritos y si es una transacción de lectura, el esclavo responde con la información solicitada por el maestro.

### **2.14.2 TRAMA ESPECIAL DE CONSULTAS (POLLING).**

Es una trama de escritura o lectura, ubicada en el banco 3 de la memoria RAM que inicia en la posición 50 y que tiene una longitud de 60 bytes de datos. El campo de datos está compuesto por: 10 bytes (80 bits) para datos booleanos, 20 bytes para datos tipo entero de 8 bits, 20 bytes para datos enteros de 16bits y 10 bytes para un dato tipo mensaje de 10 caracteres. Cuando el maestro realiza la primera consulta, es una trama de lectura.

Este tipo de trama primeramente es verificada en localidades de memoria temporal del nodo esclavo, luego dependiendo de la necesidad del nodo se copia la información en el banco 3. El nodo esclavo responde a la trama POLLING transmitiendo su dirección, la longitud igual a 60, 60 bytes de datos los cuales pertenecen al banco 3 o del banco que se desee que funcione como buffer de datos del controlador, los datos se transmitirán desde la localidad de inicio del buffer igual a 50, (esto dependerá del modelo y marca del controlador).

La información que se obtiene de cada estación es transformada de bytes (60) a variables (111) para actualizar los datos del servidor OPC. El arreglo de la información es: 10 bytes que representan a 80 variables discretas, 20 bytes que representa a 20 variables enteras positivas de 0 a 255, 20 bytes que forman 10 variables enteras positivas de 0 a 65535 (cada variable es formada por 2 bytes) y 10 bytes que forman una variable tipo mensaje de 10 caracteres (anexo B).

### **2.14.3 TRAMA ESPECIAL DE DIFUSIÓN (BROADCAST).**

Es conocida como trama de difusión en la que solo el maestro transmite y todas las estaciones la reciben, la analizan y ejecutan algún algoritmo pero ninguna responde. Este tipo de trama solo se presenta cuando la dirección de destino es 0 y permite realizar todas las funciones existentes por el campo de control.

## **CAPÍTULO 3**

### **SOFTWARE Y ALGORITMOS.**

#### **3.5 SOFTWARE USADO PARA EL DESARROLLO DEL PROYECTO.**

En el mercado de software y hardware existen muchas herramientas que permiten al desarrollador de proyectos, buscar la herramienta que se adapte a sus necesidades y habilidades.

Los microcontroladores seleccionados para la implementación del protocolo propuesto son: el PIC16F877A y PIC16F628A de Microchip. Por su bajo costo y su alto rendimiento en aplicaciones de desarrollo industrial. El software de desarrollo que estos microcontroladores utilizan es el MPLAB IDE 7.2 y es un programa de desarrollo integrado que permite: escribir, depurar y optimizar los programas de los microcontroladores PIC. El software tiene un editor de texto, un simulador, un organizador de proyectos, un ensamblador y un programador de microcontroladores PIC.

El ensamblador MPASMWIN es parte de MPLAB y soporta todas las instrucciones de los microcontroladores PIC. El lenguaje ensamblador es una representación alfa numérica del lenguaje de máquina del microcontrolador. Cada instrucción en el lenguaje ensamblador (ASM) corresponde a una instrucción en lenguaje de máquina (HEX), sin tener en cuenta macros y directivas.

Un programa en lenguaje ensamblador es rápido y corto, debido a que genera el código más óptimo posible, por ello es necesario conocer la estructura y arquitectura del microcontrolador lo cual depende del fabricante y el modelo del microcontrolador.

El software utilizado en el proyecto para grabar el código de lenguaje de máquina en el microcontrolador PIC es: melabs Programmer de la empresa microEngineering.

El software utilizado en la estación maestra, para el control del protocolo y el control del servidor OPC, es: Microsoft Visual Basic 6.0.

El control del servidor OPC está basado en Microsoft Visual Basic 6.0 y la biblioteca WTOPCSrv.dll para desarrollo rápido de servidores de datos OPC, de la empresa: WinTECH. WTOPCSrv.dll es una biblioteca que tiene costo.

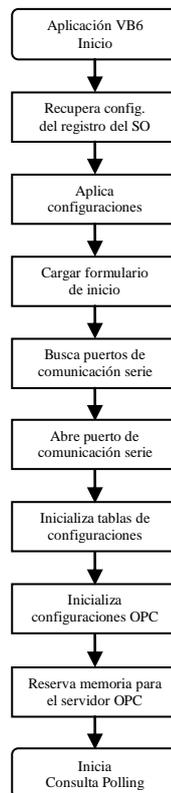
Los algoritmos y diagramas de bloques que se describen en este capítulo, están basados en la programación por eventos (interrupciones) del software principal. En el caso del nodo esclavo el software principal realiza el control del dispositivo y sus periféricos, en el caso del nodo maestro el software principal esta basado en el algoritmo de consultas polling. El funcionamiento del software principal en el nodo maestro y el nodo esclavo está basado en un algoritmo que se repite infinitamente, pero es controlado por diferentes eventos.

En el momento que se ejecute cualquier evento (interrupción), ya sea en el nodo maestro o en el nodo esclavo, se debe detener el funcionamiento de los algoritmos del programa principal y al terminar el evento que origino dicha interrupción se debe reanudar el programa principal.

### **3.6 SOFTWARE IMPLEMENTADO EN EL NODO MAESTRO.**

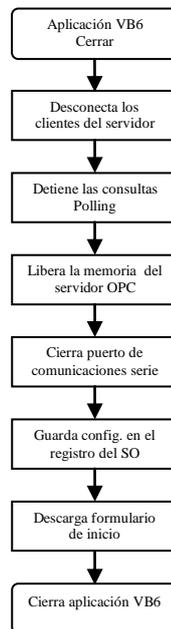
La aplicación propuesta en el presente proyecto, está basada en el desarrollo de interfaces gráficas similares a las utilizadas por los servidores OPC comerciales para PLC's y en la "lógica de funcionamiento" de diversos sistemas de monitoreo.

El software implementado tiene la capacidad de administrar el control lógico de la red RS485, de actualizar la información proveniente de las estaciones esclavas en la memoria reservada por el servidor OPC, recibir información de los clientes del servidor OPC. Para ello, se ha implementado algoritmos que permiten realizar las configuraciones de cada una de las etapas involucradas en el sistema. Los algoritmos implementados se han dividido en: algoritmos de visualización de la interfase gráfica, de protocolo, de servidor OPC y de respaldo de configuraciones.



**Figura 3.1 Pasos necesarios para iniciar la aplicación en el nodo maestro.**

En la figura 3.1 se puede observar el diagrama de bloques para iniciar el algoritmo de consultas polling. La figura 3.2 indica el diagrama en bloques que se ejecuta cuando se desea finalizar el funcionamiento del algoritmo de consultas polling y finalizar la aplicación desarrollada.



**Figura 3.2 Pasos necesarios para cerrar la aplicación en el nodo maestro.**

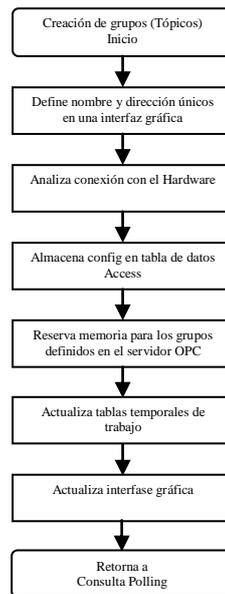
De los algoritmos y variables que interactúan en el funcionamiento del software se describirán los más importantes, como son: creación de grupos (tópicos), descripción de tablas de configuración y datos, algoritmo de consultas (polling) y configuraciones del servidor OPC.

### **3.6.1 ALGORITMO DE CREACIÓN DE GRUPOS (TÓPICOS).**

Para iniciar el monitoreo de las estaciones, éstas deben ser definidas previamente, para ello se debe seleccionar la estación o estaciones que van a ser monitoreadas. Utilizando el protocolo propuesto, con tramas especiales de consulta (polling) y configurando el servidor de datos, la aplicación es capaz de transferir los datos solicitados por las aplicaciones clientes. Los datos se transferirán de las estaciones que previamente han sido definidas en la creación de grupos (tópicos).

Para mantener la configuración de los grupos (tópicos) o nodos esclavos creados, se almacena la información en una tabla de datos de la aplicación Microsoft Access, a la cual se accede cada vez que inicie la aplicación o se modifiquen los grupos (tópicos) existentes.

La figura 3.3 indica el diagrama de bloques para la creación de grupos (tópicos) de las estaciones esclavas en la aplicación propuesta.



**Figura 3.3 Creación de grupos de estaciones esclavas.**

### **3.6.2 DESCRIPCIÓN DE TABLAS DE CONFIGURACIÓN Y DATOS.**

Para el manejo y administración a nivel interno, la aplicación utiliza *tablas* temporales. Estas tablas permiten organizar la información, y son declaradas de uso público, para permitir su acceso desde cualquier algoritmo de la aplicación.

El espacio de memoria OPC es reservado con la tabla TagHandle, esta tabla permite tener acceso a los datos de todas las estaciones (en la aplicación desarrollada máximo puede definirse 32 nodos esclavos, con 112 variables).

Las tablas que almacenan las configuraciones de los dispositivos definidos como grupos (tópicos) son: TablaDireccion(33) almacena la dirección física de cada estación, TablaNodo(33) almacena el nombre de cada estación, TablaTimeOut(33) almacena el periodo de tiempo mínimo que debe transcurrir antes de que la estación pueda ser consultada, TablaIndexOPC(33) almacena la posición que le corresponde a cada estación en la tabla TagHandle (memoria reservada para el servidor OPC) . El contenido de estas tablas se actualiza si existe alguna modificación en la definición de los grupos (tópicos). Para obtener la configuración de algún dispositivo previamente definido como grupo (tópico), se debe leer las tablas con el mismo índice.

Tablas de estado de las estaciones: TablaTimeOutContador(33) registra el tiempo que transcurre desde la última consulta que se realizó a una estación, TablaReintentos(33) cuenta el número de intentos por comunicarse el maestro con el esclavo (sin recibir respuesta, acepta máximo 3 intentos), TablaPrimeraConsulta(33) indica si se realizó la primera consulta a una estación y especifica el tipo de trama que se utilizará en la consulta (falso = escritura, verdad = lectura), TablaContadorFallas(33) almacena el número de transacciones sin respuesta en un segundo de cada nodo esclavo, TablaContadorExitos(33) almacena el número de transacciones exitosas en un segundo de cada nodo esclavo.

Tablas de análisis de la información: TablaDatosRealesTx(256) esta tabla almacena todas las variables que se lee del servidor de datos OPC y que están referidas a un solo grupo o estación (los datos no se encuentran encapsulados), TablaDatosRealesRx(256) contiene la información real que se actualizará en el servidor de datos OPC (Los datos no son encapsulados y provienen de algún nodo esclavo), TablaDatosTx(256) almacena datos encapsulados provenientes de la tabla TablaDatosRealesTx. (Son datos que se transmitirán al microcontrolador), TablaDatosRx(256) almacena datos encapsulados provenientes del microcontrolador (esta información se deberá acondicionar para copiarla en la TablaDatosRealesRx).

Las constantes y variables usualmente empleadas en conjunto con las tablas mencionadas anteriormente son: NumeroReintentos es una constante que indica el número máximo de reintentos de un esclavo, TiempoDeConsulta es una constante que

indica el tiempo mínimo que se demora el servidor en cada consulta, IndicePolling es un índice variable que indica la estación en que se encuentra el algoritmo de consultas (polling), MaxIndicePolling indica el número máximo de nodos esclavos definidos por el algoritmo de creación de grupos (tópicos).

### 3.6.3 ALGORITMO DE CONSULTA DE NODOS ESCLAVOS (POLLING).

El algoritmo de consultas (polling), se encuentra basado en el funcionamiento de las tablas antes descritas. El diagrama de flujo de la figura 3.4, describe su funcionamiento.

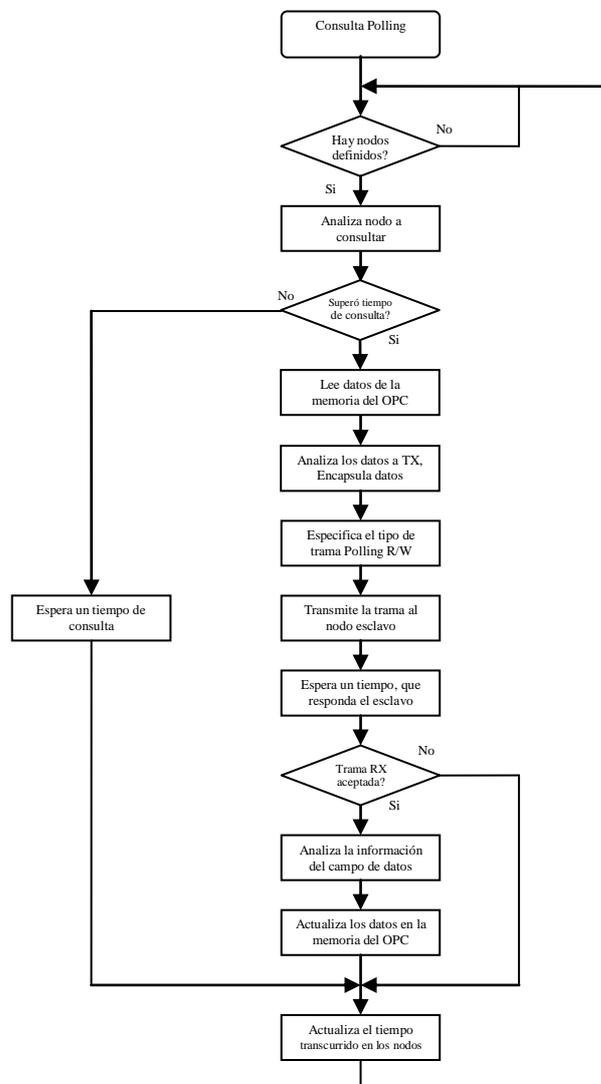
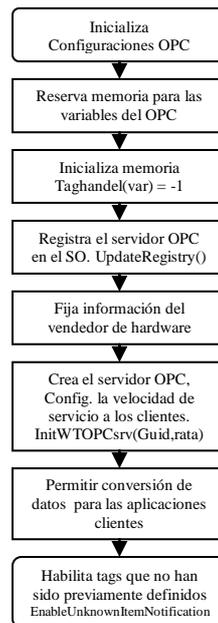


Figura 3.4 Algoritmo de consulta de nodos esclavos (polling).

### 3.6.4 IMPLEMENTACIÓN DEL SERVIDOR OPC.

El servidor de datos OPC está basado en la biblioteca WtOPCSrv.dll, su desarrollador es WinTECH Software, esta biblioteca permite un fácil uso de las API para la integración y personalización de datos con OPC. Todos los detalles del control de objetos del servidor OPC, son manipulados por la biblioteca, dando como resultado un alto nivel en la presentación de datos, esta biblioteca permite una fácil integración en aplicaciones que requieran de interfaces OPC 2.0 con los estándares de acceso a datos y exploración de los mismos.

La figura 3.5 describe el proceso para iniciar la biblioteca WtOPCSrv.dll.



**Figura 3.5 Pasos necesarios para iniciar el servidor de datos OPC.**

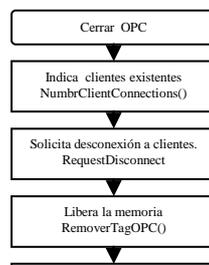
Antes de utilizar las funciones para manipular la biblioteca WtOPCSrv.dll, ésta debe ser inicializada y debe definirse el tamaño del buffer de memoria para almacenar la información que administrará el servidor OPC.

Descripción de las principales funciones usadas para la manipulación de la biblioteca WtOPCSrv.dll.

- Configura la velocidad con que el servidor puede actualizar una variable.  
***ResetServerRate***(ServerRate As Long) Long
- Desactiva el modo demostrativo de la biblioteca WtOPCSrv.dll.  
***Deactivate30MinTimer***(Authorization As Byte) Long
- Remueve el registro de Windows del servidor OPC.  
***UnregisterServer***(CLSID\_Svr As Byte, Name As Byte) Long
- Define en un espacio reservado de memoria, la creación de un variable.  
***CreateTag***(Name As Byte, Value As Variant, InitialQuality As Integer, IsWritable As Long) Long
- Remueve variables definidas por CreateTag.  
***RemoveTag***(TagHandle As Long) Long
- Actualiza el espacio reservado de memoria, asignado a una variable.  
***UpdateTag***(TagHandle As Long, Value As Variant, Quality As Integer) Long
- Lee el espacio reservado de memoria, asignado a una variable.  
***ReadTag***(TagHandle As Long, Value As Variant) Long

El espacio de memoria reservado es manipulado por la tabla TagHandle(elemento), el elemento representa la posición de la memoria, o el lugar en que se encontrará la variable.

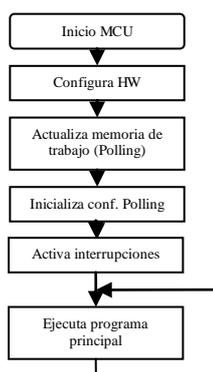
La figura 3.6 describe el proceso para detener el servicio de la biblioteca WtOPCSrv.dll (servidor OPC).



**Figura 3.6 Pasos necesarios para detener al servidor de datos OPC.**

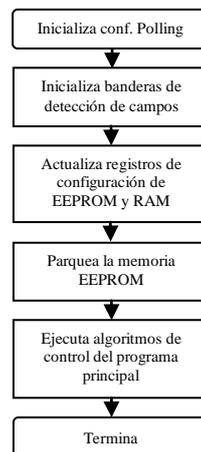
### **3.7 SOFTWARE IMPLEMENTADO EN LOS NODOS ESCLAVOS.**

En la figura 3.7, se describe el algoritmo de inicialización del microcontrolador. El microcontrolador primero configura el hardware para el programa principal de control, luego inicializa las configuraciones del protocolo de comunicación y posteriormente inicia la ejecución del programa principal. El diagrama en bloques de la figura 3.7, solo se ejecuta cuando el microcontrolador inicia, por ejemplo: conexión de la polarización o cuando se produce un reset en el microcontrolador.



**Figura 3.7 Algoritmo de inicio del microcontrolador.**

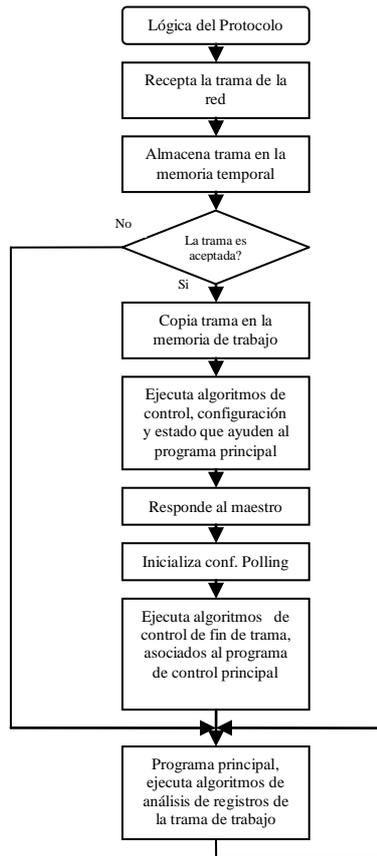
El diagrama de bloques de la figura 3.8, se utiliza para inicializar las variables que tienen relación con el protocolo de comunicación. El diagrama de bloques de la figura 3.8 se ejecuta: al iniciar el microcontrolador, cuando una trama recibida tiene errores de CRC, después de ejecutar los algoritmos de una trama broadcast y después de realizar la respuesta de cualquier trama.



**Figura 3.8 Algoritmo de inicialización del protocolo de comunicación.**

El algoritmo de recepción del protocolo de comunicación, recibe la trama de la red y sólo cuando la trama está completa y sin errores, el protocolo de comunicación interactúa con los algoritmos del programa principal o las localidades de memoria asignadas para el intercambio de información entre el protocolo de comunicación y el programa principal.

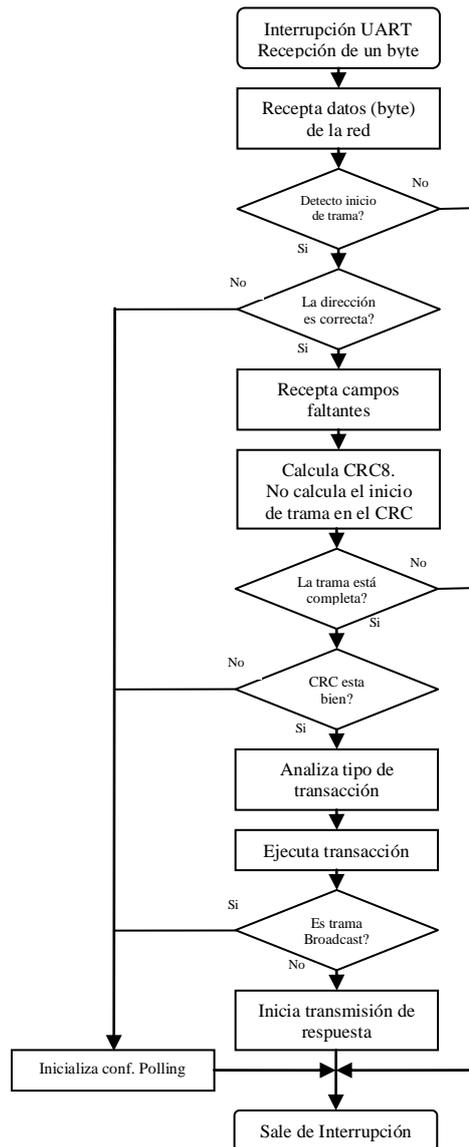
El diagrama de flujo de la figura 3.9, describe el proceso en que interactúa el protocolo de comunicación con el programa principal de control.



**Figura 3.9 Interacción del protocolo de comunicación con el programa principal de control.**

### **3.7.1 ALGORITMOS DE TRANSMISIÓN Y RECEPCIÓN DE DATOS.**

Los algoritmos de transmisión y recepción (figura 3.10 y figura 3.11) están basados en la interrupción del periférico de comunicación UART, El uso de interrupciones del programa principal aumenta el rendimiento del algoritmo de control existente en el programa principal.

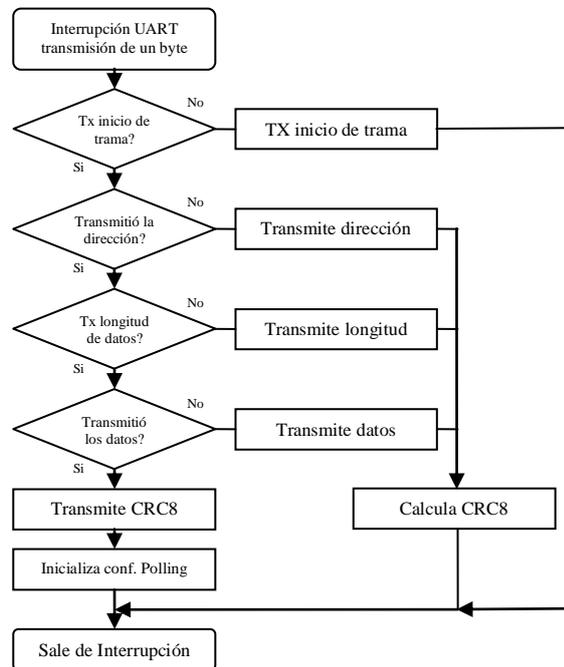


**Figura 3.10 Algoritmo de recepción de datos.**

Para que se produzca una interrupción de recepción del periférico UART, éste tiene que recibir 10 bits de la red, los 10 bits corresponden a: un bit de inicio, 8 bits de datos (un byte) y un bit de parada. Cuando el UART haya completado la recepción de los 10 bits, se genera la interrupción (evento) del programa principal y se ejecuta el algoritmo de análisis del byte recibido que se ilustra en la figura 3.10.

El proceso de transmisión de la trama de respuesta del nodo esclavo, está basado en la interrupción del programa principal. El periférico UART genera una interrupción

cuando éste finaliza la transmisión de un byte, la interrupción generada suspende la ejecución del programa principal y ejecuta el algoritmo de la figura 3.11.



**Figura 3.11 Diagramas de flujo de transmisión de datos.**

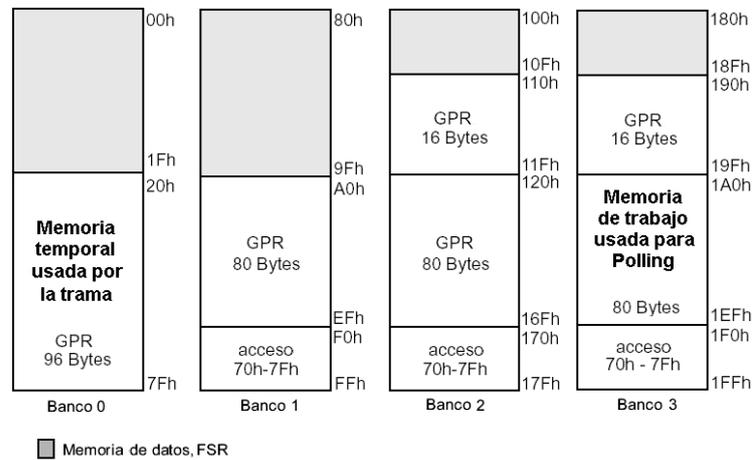
La transmisión y la recepción de la trama de comunicación, genera una interrupción por cada byte que sea receptado o transmitido, de cualquier trama de comunicación.

### **3.7.2 ORGANIZACIÓN DE LA INFORMACIÓN EN LA MEMORIA.**

La organización de información en la memoria del microcontrolador está condicionada por el número de bancos y la capacidad de memoria de cada banco.

Si coincide la dirección de la trama con la dirección del nodo esclavo, toda la información de la trama obtenida de la red, se almacena en una memoria temporal. Se analiza la información de la memoria temporal y dependiendo del tipo de transacción se copia en la memoria de trabajo.

En la figura 3.12 se ilustra la organización de la memoria para los microcontroladores PIC16F877A.



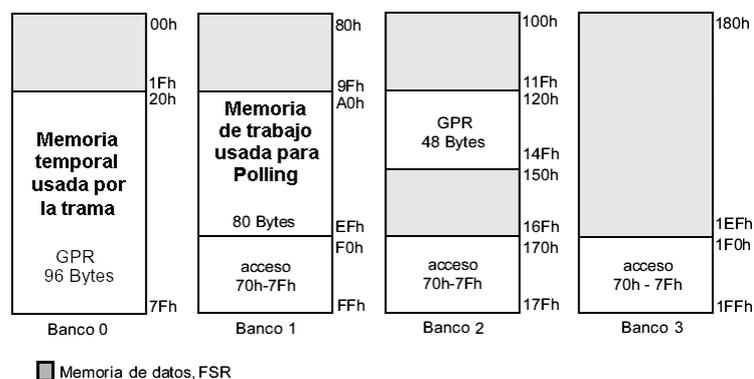
**Figura 3.12** Uso de la memoria para consultas en el PIC16F877A.

La memoria de trabajo permite almacenar la información que se utilizará para una trama especial de consultas (polling).

Para el control de tramas de tipo no especial, la información se analiza en la memoria temporal. Si es una trama de escritura, la información se mueve al bloque de memoria especificado y retransmite la información desde la memoria temporal. Si es una trama, de lectura se copia el bloque de memoria especificado por la transacción a la memoria temporal y luego se transmite al maestro.

Para el control de trama especial de consultas (polling), la información receptada es analizada (CRC) en la memoria temporal y luego solo la información que es requerida es copiada a la memoria de trabajo. Antes de responder al maestro usando la información de la memoria de trabajo, la estación actualiza los registros asociados al programa principal o al de control. En el anexo B, se puede observar las localidades de memoria asociadas a la memoria temporal y a la memoria de trabajo, estas localidades

almacenan los diferentes tipos de datos que pueden manejar los algoritmos implementados. En la figura 3.13, se puede observar una organización de memoria diferente a la que se adaptó el protocolo de comunicación para tramas especiales de consultas (polling).



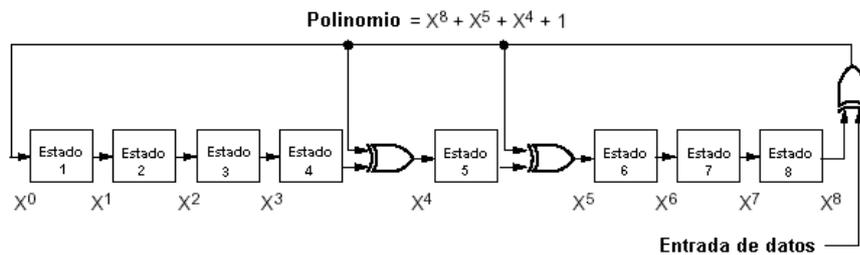
**Figura 3.13** Uso de la memoria para consultas en el PIC16F628A.

### 3.8 ALGORITMO DE DETECCIÓN DE ERRORES.

La técnica usada para la detección de errores es chequeo por redundancia cíclica de ocho bits, el algoritmo de detección de errores CRC-8, esta basado en el polinomio:  $x^8 + x^5 + x^4 + 1$ , como se indica en la figura 3.14.

El algoritmo usado en la implementación es conocido con el nombre “DOW Dallas CRC-8”, éste algoritmo se encuentra detallado por la empresa Maxim/Dallas AP Note 27<sup>18</sup>, y se utiliza en los productos iButton.

<sup>18</sup> AP Note27 "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor iButton Products", Maxim/Dallas.



**Figura 3.14 Representación del polinomio CRC8.**

La nota de aplicación AP Note27, describe ampliamente el algoritmo CRC-8. Su implementación puede estar basada en: un tabla 256 bytes de CRC, arreglos de medio byte (nibbles array) o aplicando aritmética booleana. En el proyecto se usó el algoritmo basado en aritmética booleana, éste algoritmo se indica en el diagrama de flujo de la figura 3.15.

En el proyecto propuesto, no toma en consideración el campo de inicio de trama para calcular el dato de error CRC. Si el dato de error CRC calculado, coincide con el dato de inicio de trama (170) se suma una unidad al código de error CRC calculado (171), esto se realiza para evitar conflicto de inicio de trama.

Utilizando el algoritmo de la figura 3.15 se calcula el dato de error CRC de los bytes de la trama a transmitir, al final se adjunta el dato calculado del error CRC (campo de detección de error) y se transmite en la trama. El nodo receptor, recalcula el dato de error CRC de la información recibida de la trama y compara con el campo de detección de error recibido, si los datos son iguales, el nodo acepta la trama, caso contrario la rechaza.

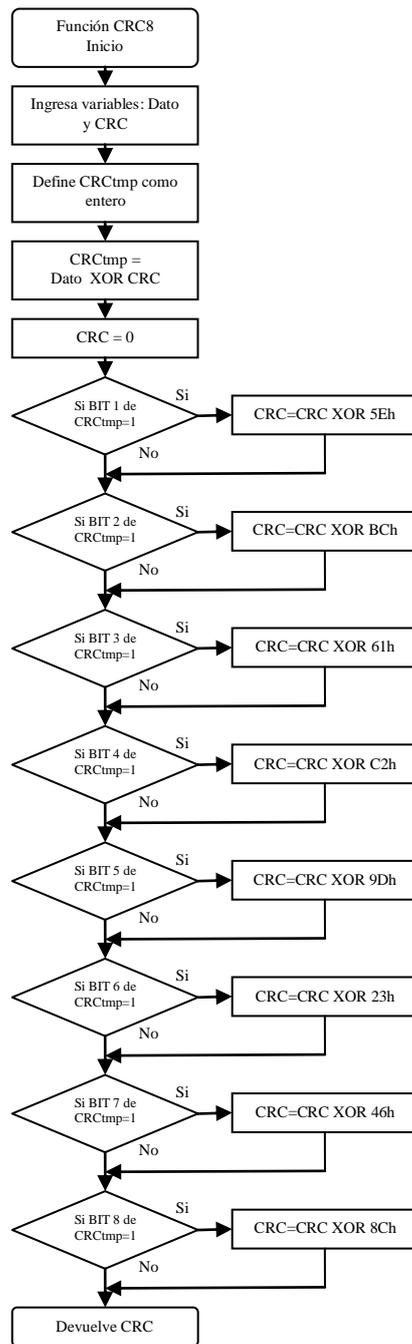


Figura 3.15 Algoritmo de chequeo de error CRC8.

## **CAPÍTULO 4**

### **RESULTADOS EXPERIMENTALES.**

#### **4.1 REQUERIMIENTOS PARA LA INSTALACIÓN DEL SOFTWARE DESARROLLADO.**

Para instalar la aplicación del proyecto: el usuario debe tener permisos de administrador del sistema, instalar los componentes del servidor OPC: “OPC Core Components 2.00 Redistributable” el cual se obtiene de la pagina Web de OPC Foundation, verificar que el sistema tenga la biblioteca actxprxy.dll; si no existiera se debe obtener el “ActiveX Redistributable Installation Kit” de Microsoft que se encuentra en la página Web de OPC Foundation.

El software diseñado se instala directamente en el sistema operativo (SO) Microsoft Windows XP o Microsoft Windows 2000, el arreglo de la interfase gráfica se adapta a los estilos del SO.

Si el SO está basado en Windows 95/98/ME, se deben incluir los componentes de objetos genéricos de distribución (DCOM), además debe existir la biblioteca “MSVBVM60.dll” o “VBRun60.exe”.

El nombre del software desarrollado es “MicroBUS” y fue implementado para monitorear (utilizando la especificación OPC) y analizar una red RS485 de

microcontroladores basados en el protocolo de comunicación propuesto en este proyecto.

## **4.2 PRUEBAS DEL PROTOCOLO DE COMUNICACIÓN**

Las pruebas de funcionamiento de los diferentes tipos de tramas, se realizó con un nodo esclavo conectado a una distancia de 25 metros del nodo maestro, la velocidad de comunicación se configuró en 115200 bps y se instaló el software “MicroBUS” en el nodo maestro.

El formulario “Análisis de tramas” del software “MicroBUS” permite transmitir los tipos de tramas analizados en este capítulo, para esto el usuario debe ingresar la información de los campos de: dirección, localidad de inicio, longitud del campo de datos (solo en tramas de lectura) y el campo de datos (solo en tramas de escritura). Los datos ingresados en los campos antes descritos deben ser tipo decimal.

Los tipos de transacciones (lectura, escritura y tipo de memoria) se diferencian por los bits del campo de control y éstos se modifican desde la interfase gráfica del formulario “Análisis de tramas”, el campo de inicio, control y chequeo de errores se añaden automáticamente al transmitir la trama.

En este capítulo se utilizan tablas para describir y analizar la información de los campos de las tramas de: transmisión del nodo maestro y transmisión del nodo esclavo (respuesta del nodo esclavo), la información de los campos de las tramas está basado en ejemplos que demuestran el funcionamiento del protocolo de comunicación.

### **4.2.1 LECTURA DE LA MEMORIA RAM.**

La tabla 4.1 indica un ejemplo de la estructura de las tramas de: transmisión (transmite el nodo maestro al nodo esclavo) y recepción (transmite el nodo esclavo al nodo maestro), de una transacción de lectura de la memoria RAM del microcontrolador (nodo esclavo).

En esta prueba (tabla 4.1) se realiza la lectura de la memoria RAM (campo de control = 65d) desde la localidad 85h (banco 2, localidad de inicio 5d), hasta la localidad 99h (banco 2, longitud 21d). El dispositivo esclavo es un microcontrolador PIC16F628A y la dirección del dispositivo es 5d (campo de dirección).

**Tabla 4.1 Datos para la lectura de la memoria RAM del nodo esclavo.**

MAESTRO		ESCLAVO	
INICIO	170 170 170	INICIO	170 170 170
DIRECCIÓN	5d	DIRECCIÓN	5d
CONTROL	65d	LONGITUD	21d
LOCALIDAD	5d	DATOS	32,243,0,0,0,0,85,32,0,8,0, 0,0,255,0,0,0,0,0,38,10
LONGITUD	21d	CRC	19 d
DATOS	--		
CRC	70d		

Analizando los datos más importantes obtenidos (byte1 = 32d, byte2 = 243d, byte20 = 38d y byte21 = 10d del campo de datos) del nodo esclavo con ayuda del anexo G se demuestra la configuración del hardware. Por ejemplo:

- TRISA=32d (100000b), indica que el puerto A está configurado como salidas.
- TRISB=243d (11110011b), el bit3 y bit4 están configurados como salidas.
- TXSTA=38d (100110b), el bit5 habilita la transmisión, el bit2 permite alta velocidad de comunicación.
- SPBRG=10d (1010b), el generador de baudios esta configurado para 115200bps.

#### 4.2.2 ESCRITURA DE LA MEMORIA RAM.

La tabla 4.2 indica la escritura de un byte con dato 1d (campo de longitud = 1d y campo de datos = 1d) en el registro PORTA (localidad de inicio 5d) del banco 0 de la memoria RAM (campo de control 0d) del microcontrolador PIC16F628A con dirección 5d. Previamente el puerto A (PORTA) debe estar configurado como salidas digitales.

**Tabla 4.2 Datos para escritura en la memoria RAM en el nodo esclavo.**

MAESTRO		ESCLAVO	
INICIO	170 170 170	INICIO	170 170 170
DIRECCIÓN	5d	DIRECCIÓN	5d
CONTROL	0d	LONGITUD	1d
LOCALIDAD	5d	DATOS	1d
LONGITUD	1d	CRC	175d
DATOS	1d		
CRC	125d		

La tabla 4.2 (esclavo) indica la trama de respuesta del nodo esclavo, dicha trama realiza un eco del campo de datos para indicar que se ha realizado la transacción.

#### 4.2.3 LECTURA DE LA MEMORIA EEPROM.

Las configuraciones: dirección del dispositivo (Localidad 0d), velocidad de comunicación (Localidad 1d) y parqueo de la memoria EEPROM (Localidad 2d), se encuentran almacenadas en la memoria EEPROM, esto permite la modificación de las configuraciones por software.

La tabla 4.3 indica un ejemplo de lectura de la memoria EEPROM (campo de control 195d) desde la localidad 0d hasta la localidad 2d (campo de longitud 3d), del microcontrolador PIC16F628A con dirección 5.

**Tabla 4.3 Datos para lectura de la memoria EEPROM del nodo esclavo.**

MAESTRO		ESCLAVO	
INICIO	170 170 170	INICIO	170 170 170
DIRECCIÓN	5d	DIRECCIÓN	5d
CONTROL	195d	LONGITUD	3d
LOCALIDAD	0d	DATOS	5d, 5d, 255d
LONGITUD	3d	CRC	165d
DATOS	--		
CRC	212d		

La información obtenida corresponde a: la localidad 0d almacena el dato 5d para la dirección del dispositivo, la localidad 1d almacena la configuración de velocidad del microcontrolador, la localidad 2d almacena el valor 255d.

#### 4.2.4 ESCRITURA DE LA MEMORIA EEPROM.

Como ejemplo para probar la trama de escritura de la memoria EEPROM se propone modificar por software la dirección del dispositivo esclavo con dirección 5d, por la dirección 2d.

La tabla 4.4 indica los campos de la trama de transmisión para modificar la localidad 0d (campo de localidad de inicio) de la memoria EEPROM (campo de control 131d) con el dato 2d (campo de datos).

**Tabla 4.4 Datos para escritura de la memoria EEPROM en el nodo esclavo.**

MAESTRO		ESCLAVO	
INICIO	170 170 170	INICIO	170 170 170
DIRECCIÓN	5d	DIRECCIÓN	5d
CONTROL	131d	LONGITUD	1d
LOCALIDAD	0d	DATOS	2d
LONGITUD	1d	CRC	77d
DATOS	2d		
CRC	251d		

Cualquier transacción de escritura que realice el nodo maestro en la memoria EEPROM y que involucre las tres primeras localidades de dicha memoria, puede desconfigurar el funcionamiento del protocolo en el microcontrolador.

#### 4.2.5 TRAMA ESPECIAL DE CONSULTAS (POLLING), LECTURA.

Cuando el nodo maestro transmite una trama, el microcontrolador analiza los campos de la trama recibida, si los campos son: localidad = 50d, longitud = 60d y el campo de control hace referencia a la memoria RAM (FSREE = 0) en el banco 3, el software del microcontrolador acepta la transacción como trama especial de consultas “Polling”.

Si el microcontrolador detecta una trama especial de consultas (Polling), cualquier tipo de transacción, sea ésta de lectura o escritura, será referida al buffer de memoria asignado para este tipo de transacciones. El buffer utilizado para tramas especiales de consultas, puede estar localizado en cualquier posición de la memoria RAM, lo cual dependerá de la organización de memoria del microcontrolador. El buffer de memoria reservada para este tipo de tramas se llama “memoria de trabajo”.

Antes de iniciar un proceso de monitoreo, el servidor de datos, debe conocer el estado de las variables de cada estación. Por ello se inicia con una trama de lectura (figura 4.5) de la memoria de trabajo del microcontrolador, con una trama especial de consultas (polling).

Los datos de la lectura de la memoria de trabajo del microcontrolador con dirección 2h, utilizando una trama especial de consultas, se presenta en la tabla 4.6.

**Tabla 4.5 Datos para lectura de la memoria de trabajo del microcontrolador.**

MAESTRO	
INICIO	170 170 170
DIRECCIÓN	2
CONTROL	67d
LOCALIDAD	50d
LONGITUD	60d
DATOS	--
CRC	115d

**Tabla 4.6 Respuesta del esclavo.**

ESCLAVO	
INICIO	170, 170, 170
DIRECCIÓN	2
LONGITUD	60d
DATOS	0, 247, 247, 255, 247, 127, 255, 127, 40, 100, 255, 0, 0, 0, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 0, 0, 32, 77, 105, 99, 114, 111, 66, 85, 83, 32
CRC	219d

El maestro chequea que no existan errores en la trama recibida, luego adecua la información y la copia en el buffer de memoria del servidor de datos OPC. Este proceso se realiza cada vez que se lee por primera vez el controlador, o cuando éste permaneció sin responder un determinado periodo de tiempo (tres consultas sucesivas).

El proceso de análisis de la información, varía después de la primera trama recibida de un nodo esclavo. Antes de copiar la información en el buffer del servidor OPC, se verifica si algún cliente del servidor OPC varió algún tag, si un tag fue variado no se actualizará dicho tag con el dato recibido del nodo esclavo. (Un tag está representado por la información de la aplicación cliente y se encuentra almacenado en el elemento del servidor de datos OPC).

#### 4.2.6 TRAMA ESPECIAL DE CONSULTAS (POLLING), ESCRITURA.

En este caso se desea escribir la memoria de trabajo del microcontrolador con dirección 2, utilizando una trama especial de consultas, como lo indica la tabla 4.7, en la que se propone la escritura de los datos en código ASCII: “0123456789 ABCDEFGHIJKLMN OPQRSTUVWXYZabdefghijklmnopqrstuv”.

**Tabla 4.7 Datos para la escritura de la memoria de trabajo del microcontrolador.**

MAESTRO	
INICIO	170 170 170
DIRECCIÓN	2
CONTROL	3d
LOCALIDAD	50d
LONGITUD	60d
DATOS	48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 32, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 32, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118,
CRC	137d

**Tabla 4.8 Respuesta del esclavo.**

ESCLAVO	
INICIO	170, 170, 170
DIRECCIÓN	2
LONGITUD	60d
DATOS	48, 247, 247, 255, 247, 127, 255, 127, 40, 100, 32, 0, 0, 0, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 0, 0, 32, 77, 105, 99, 114, 111, 66, 85, 83, 32
CRC	186d

La trama de respuesta del nodo esclavo se indica en la tabla 4.8. El maestro recibe la información, chequea que no existan errores en la trama recibida, luego adecua la información y la copia en el buffer de memoria del servidor de datos OPC.

El nodo esclavo copia la localidad 50d y la 60d de la memoria temporal (trama recibida del maestro byte 1 y 11, obsérvese en la tabla 4.7 y tabla 4.8) a la memoria de trabajo (trama que se enviará al maestro) ésta información corresponde al dato 48d (localidad 50d) y al dato 32d (localidad 60d) en la memoria del microcontrolador. Antes de copiar la información de la memoria temporal a la memoria de trabajo, la información debe ser analizada, ésto depende de la información que requiera el programa principal del microcontrolador y la información que se desee compartir con el maestro.

#### 4.2.6.1 ANÁLISIS DE LA INFORMACIÓN OBTENIDA DEL NODO ESCLAVO.

La información obtenida del campo de datos de la trama que se indica en la tabla 4.8, es analizada por la aplicación desarrollada en el nodo maestro. Los primeros 10 bytes del campo de datos, constituyen las variables booleanas, los bytes se desencapsulan y se copia la información en el espacio de memoria del servidor OPC, dichas localidades de memoria son representadas por los elementos (ítems) de D00 a D97, que corresponden a 80 elementos del servidor de datos, como lo indica la tabla 4.9.

**Tabla 4.9 Representación de datos booleanos en el buffer de memoria del servidor OPC.**

Dato recibido		Elementos (ítems) del servidor OPC
Decimal	Binario	
48	00110000b	D00=0 D01=0 D02=0 D03=0 D04=1 D05=1 D06=0 D07=0
247	11110111b	D10=1 D11=1 D12=1 D13=0 D14=1 D15=1 D16=1 D17=1
247	11110111b	D20=1 D21=1 D22=1 D23=0 D24=1 D25=1 D26=1 D27=1
255	11111111b	D30=1 D31=1 D32=1 D33=1 D34=1 D35=1 D36=1 D37=1
247	11110111b	D40=1 D41=1 D42=1 D43=0 D44=1 D45=1 D46=1 D47=1
127	01111111b	D50=1 D51=1 D52=1 D53=1 D54=1 D55=1 D56=1 D57=0
L 255	11111111b	D60=1 D61=1 D62=1 D63=1 D64=1 D65=1 D66=1 D67=1
127	01111111b	D70=1 D71=1 D72=1 D73=1 D74=1 D75=1 D76=1 D77=0
40	00101000b	D80=0 D81=0 D82=0 D83=1 D84=0 D85=1 D86=0 D87=0
s 100	01100100b	D90=0 D91=0 D92=1 D93=0 D94=0 D95=1 D96=1 D97=0

bytes 11 al 30 del campo de datos de la tabla 4.8 corresponden a las variables enteras positivas de 8 bits, estos bytes se copian directamente al espacio de memoria del servidor OPC, estas localidades de memoria son representadas por los elementos (ítems) del I00 a I19 (tabla 4.10) y corresponden a 20 elementos del servidor de datos OPC.

**Tabla 4.10 Representación de datos enteros positivos de 8 bits.**

<b>Dato recibido</b>	<b>Elementos (ítems)</b>
Decimal	del servidor OPC
32	I00= 32
0	I01= 0
0	I02= 0
0	I03= 0
4	I04= 4
5	I05= 5
6	I06= 6
7	I07= 7
8	I08= 8
9	I09= 9
10	I10= 10
11	I11= 11
12	I12= 12
13	I13= 13
14	I14= 14
15	I15= 15
16	I16= 16
17	I17= 17
18	I18= 18
19	I19= 19

Los bytes 31 al 50 del campo de datos de la tabla 4.8 corresponden a las variables enteras positivas de 16 bits. Los datos de 16 bits se encapsulan uniendo dos bytes, el primer byte recibido corresponde a los bits más significativos, el siguiente byte recibido corresponde a los bits menos significativos. Las variables enteras positivas de 16 bits son representadas por los elementos (ítems) del A00 a A09 (tabla 4.11) y corresponden a 10 elementos del servidor de datos OPC.

**Tabla 4.11 Representación de datos enteros positivos de 16 bits.**

<b>Dato recibido</b>	<b>Elementos (ítems)</b>
Decimal	del servidor OPC
20, 21	A00= 5141
22, 23	A01= 5655
24, 25	A02= 6469
26, 27	A03= 6683
28, 29	A04= 7197
30, 31	A05= 7711
32, 33	A06= 8225

34, 35	A07=	8739
36, 37	A08=	9253
0, 0	A09=	0

Los bytes 51 al 60 del campo de datos de la tabla 4.8 corresponden a la variable tipo mensaje. El encapsulado de la variable tipo mensaje está formado máximo por 10 caracteres codificados en ASCII, el primer byte recibido corresponde al primer carácter, el segundo byte recibido corresponde al segundo carácter y así sucesivamente (tabla 4.12), cualquier carácter nulo (0d) representa el final del mensaje. Si el mensaje tiene 10 caracteres diferentes del nulo (0d), el software de la estación maestra incluye el carácter nulo (0d) a la cadena de caracteres. Cuando el mensaje se encuentra encapsulado se copia al espacio de memoria del servidor OPC, esta localidad de memoria es representada por el elemento (ítem) M00.

**Tabla 4.12 Representación del dato tipo cadena de caracteres.**

Datos recibidos Decimal	Elemento (ítem) del servidor OPC
32, 77, 105, 99, 114, 111, 66, 85, 83, 32	M00= MicroBUS

#### 4.2.7 TRAMA ESPECIAL DE DIFUSIÓN (BROADCAST).

Cuando el nodo maestro transmite una trama y el campo de dirección es cero, todos los nodos esclavos analizan y ejecutan la transacción, pero ninguno responde. A dicha trama se le conoce como trama especial de difusión (broadcast).

**Tabla 4.13 Escritura de la memoria RAM de todos los nodos esclavos.**

MAESTRO	
INICIO	170 170 170
DIRECCIÓN	0d
CONTROL	0d
LOCALIDAD	50d
LONGITUD	1d
DATOS	1d
CRC	11d

La tabla 4.13 indica la estructura de una trama de difusión, que escribe el dato 1d en la localidad 50d del banco 0 (memoria RAM). Todos los esclavos conectados a la red ejecutan dicha transacción y ninguno responde. El maestro espera un determinado periodo de tiempo para que los nodos esclavos ejecuten los algoritmos de control.

#### **4.3 DESCONEXIÓN DE UN NODO ESCLAVO.**

La desconexión de un nodo esclavo se produce por los siguientes factores: avería del hardware, corrupción del software existente en la memoria de programas, avería del adaptador de interfaces, errores en la trama de comunicación y desconexión del cable de red RS485. Estos factores impiden que exista comunicación entre el maestro y el esclavo.

Si el nodo esclavo no responde a las tramas especiales de consulta (Polling) o la transacción tiene errores de CRC, la actualización del buffer del servidor OPC se suspende.

Cuando no responde un nodo esclavo, se dice que el nodo esclavo no fue reconocido (NACK) por el nodo maestro y el elemento (ítem) error = 1. En el caso contrario se dice, que la estación fue reconocida (ACK) por el nodo maestro y el elemento (ítem) error = 0.

Para que una estación pase del estado reconocido (ACK) al estado no reconocido (NACK), el maestro analiza tres tramas consecutivas de consulta (polling) a las que el esclavo no haya respondido correctamente y posteriormente actualiza el elemento (ítem) error = 1. En cada transacción que el esclavo no fuere reconocido por el maestro, los datos no se actualizan en el buffer de memoria del servidor de datos OPC.

Para que una estación pase del estado no reconocido (NACK) al estado reconocido (ACK), el maestro transmite tramas especiales de consulta (polling) de lectura y si el esclavo responde; el maestro actualiza la información del esclavo en el buffer del servidor de datos OPC, además actualiza el elemento (ítem) error = 0.

#### 4.4 CONSUMO DE RECURSOS DEL NODO ESCLAVO.

El microcontrolador consume 13 localidades (RAM) para el control del protocolo, 60 localidades (RAM) para la recepción de información (memoria temporal), 60 localidades (RAM) para la memoria de trabajo. El consumo de recursos de la memoria de propósito general es 133 localidades, que representa el 33% de localidades del PIC16F877A y el 60% de localidades del PIC16F628A.

El consumo del recurso de la memoria de programas para los algoritmos del protocolo, es mínimo 480 palabras. El consumo del recurso de la memoria de programas para el PIC16F877A es 6% y para el PIC16F628A es el 24%

El protocolo consume 3 localidades de la memoria EEPROM para almacenar las configuraciones del protocolo de comunicación, esto representa menos del 3% de las localidades de la memoria EEPROM.

$$\begin{aligned} \text{Ciclos de máquina} &= \frac{4}{\text{Frecuencia del oscilador}} \\ \text{Ciclos de máquina} &= \frac{4}{20 \text{ Mhz}} \\ \text{Ciclos de máquina} &= 0.2 \text{ microsegundos} \end{aligned}$$

**Figura 4.1 Cálculo del ciclo de máquina del microcontrolador.**

Cada nodo esclavo funciona con un oscilador de 20Mhz, por ello el microcontrolador ejecuta un ciclo de máquina cada 0.2 microsegundos (figura 4.1). La respuesta de funcionamiento de los microcontroladores: PIC16F877A y el PIC16F628A, es similar a dicha frecuencia de trabajo. Para calcular los retardos de tiempo de la tabla 4.14 (figura 4.2), se utilizó como referencia el oscilador de 20Mhz.

**Tabla 4.14 Retardos aproximados en la ejecución de algoritmos en el nodo esclavo.**

Descripción del procesamiento:	Ciclos de máquina:	Tiempo en microsegundos:
Recepción de cada byte del campo de inicio	51	10,2
Recepción del campo de dirección	79	15,8
Recepción del campo de control	80	16
Recepción del campo de localidad	85	17
Recepción del campo de longitud de datos	94	18,8
Recepción de cada byte del campo de datos	103	20,6
Recepción del campo de CRC y análisis de errores	66	13,2
Análisis de la trama especial de consultas (Escritura)	97	19,4
Inicialización de configuraciones del protocolo	65	13
Transmisión de cada byte	60	12

El algoritmo de control ubicado en el programa principal es interrumpido cuando el UART recibe o transmite cualquier byte de información. El tiempo de procesamiento de cualquier byte recibido de la red y que no hacen referencia a dicho nodo esclavo, es de 10.2 microsegundos aproximadamente (cálculo realizado para un oscilador de 20Mhz).

$\text{Recepción de cada byte del campo de inicio} = \text{Tiempo de un ciclo de máquina} \cdot \text{Número de ciclos a ejecutar}$ $\text{Recepción de cada byte del campo de inicio} = 0.2 \text{ microsegundos} \cdot 51 \text{ ciclos de máquina}$ $\text{Recepción de cada byte del campo de inicio} = 10.2 \text{ microsegundos}$
---

#### **Figura 4.2 Cálculo del tiempo de ejecución del algoritmo de recepción de un byte del campo de inicio.**

Cuando el programa principal es interrumpido por la recepción de cada byte de la trama especial de consulta y la trama hace referencia a dicho nodo esclavo, cada interrupción dura un promedio de tiempo de 20.2 microsegundos (promedio calculado en base a la tabla 4.14 y referido a una transacción tipo polling de escritura).

El tiempo máximo que es interrumpido el programa principal, se produce cuando el microcontrolador recepta el último byte de una trama especial de consultas. El trabajo que realiza el microcontrolador es: recepta el último byte (CRC), analiza que no existan errores en la trama, analiza el tipo de trama, transmite el primer byte del campo de inicio y reinicia las configuraciones de recepción del protocolo. El microcontrolador ejecuta un mínimo de 222 ciclos de máquina que corresponden a 44.4 microsegundos de interrupción del programa principal, en este análisis no está tomado en cuenta la ejecución de algoritmos de copia de la memoria temporal a la memoria de trabajo. Un nodo esclavo procesa la información durante un tiempo mínimo aproximado de 44.4 microsegundos antes de iniciar la respuesta de la trama especial de consultas (polling).

#### **4.5 ANÁLISIS DE TIEMPO DE LA TRAMA ESPECIAL DE CONSULTAS.**

Cada byte que se transmite por la red RS485 produce una interrupción de recepción del periférico UART en todos los nodos esclavos que se encuentren conectados en la red, menos en el nodo que transmita la trama.

Para que un nodo esclavo recepte un byte de una trama, el periférico UART de dicho nodo tiene que recibir 10 bits (un bit de inicio, 8 bits de datos y un bit de parada), si la

velocidad de comunicación de datos está configurada a 115200 bps un bit se transmitirá cada 8,6 microsegundos y el periférico UART receptorá un byte cada 86 microsegundos.

$$\text{Ciclos de máquina antes de interrupción} = \frac{\# \text{ de bits del protocolo RS232}}{\text{Ciclo de máquina} \cdot \text{Velocidad de comunicación (bps)}}$$

$$\text{Ciclos de máquina antes de interrupción} = \frac{20\text{Mhz} \cdot 10}{4 \cdot 115200\text{bps}}$$

$$\text{Ciclos de máquina antes de interrupción} = 434 \text{ Ciclos}$$

**Figura 4.3 Ciclos de máquina a ejecutar antes de una interrupción UART.**

Un nodo esclavo podría ejecutar máximo 434 ciclos de máquina (figura 4.3) para atender al programa principal de control y al programa del protocolo de comunicación, antes de que se produzca una interrupción de recepción del periférico UART.

El tiempo que se tarda en transmitir el nodo maestro una trama especial de consultas (polling) de escritura a los nodos esclavos, es de 5,903 milisegundos y está formado por 68 bytes de los campos de la trama (figura 4.4).

$$\text{Transmisión del maestro} = \frac{\# \text{ de bits del protocolo RS232} \cdot \text{bytes de la trama}}{\text{Velocidad de comunicación (bps)}}$$

$$\text{Transmisión del maestro} = \frac{10 \cdot 68}{115200}$$

$$\text{Transmisión del maestro} = 5.902 \text{ milisegundos}$$

**Figura 4.4 Tiempo de transmisión de una trama especial de consultas por el maestro.**

El tiempo que se tarda en transmitir una respuesta el nodo esclavo a una trama especial de consultas (polling) de escritura, es de 5,729 milisegundos y está formado por 66 bytes de los campos de la trama (figura 4.5).

$$\text{Respuesta del esclavo} = \frac{\# \text{ de bits del protocolo RS232} \bullet \text{ bytes de la trama}}{\text{Velocidad de comunicaci3n (bps)}}$$

$$\text{Respuesta del esclavo} = \frac{10 \bullet 66}{115200}$$

$$\text{Respuesta del esclavo} = 5.729 \text{ milisegundos}$$

**Figura 4.5 Tiempo de respuesta de una trama especial de consultas por el esclavo.**

El tiempo m3nimo requerido para realizar una consulta por parte de la estaci3n maestra es de 11.67 milisegundos y est3 formado por: el tiempo de transmisi3n (maestro) 5,9 milisegundos, tiempo de an3lisis de la trama (esclavo) 44.4 microsegundos, tiempo de respuesta del esclavo 5,7 milisegundos y el tiempo de procesamiento de la informaci3n por el nodo maestro (3ste tiempo depende de las caracter3sticas f3sicas del computador y de las aplicaciones que se est3 ejecutando en dicho nodo).

El n3mero m3ximo de consultas (figura 4.6) que puede realizar el nodo maestro en un segundo es 85. El software y el hardware implementado en la pr3ctica permiten un m3ximo de 50 consultas en un segundo y 3sto depende de las caracter3sticas del hardware del nodo maestro (computador) y las tareas (aplicaciones) que el nodo maestro est3 ejecutando en ese instante.

$$\text{N3mero m3ximo de consultas} = \frac{\text{Tiempo de comparaci3n}}{\text{Transmisi3n del maestro} + \text{Procesamiento} + \text{Respuesta del esclavo}}$$

$$\text{N3mero m3ximo de consultas} = \frac{1000 \text{ milisegundos}}{(5.903 + 0,0444 + 5.729) \text{ milisegundos}}$$

$$\text{N3mero m3ximo de consultas} = 85 \text{ Transacciones}$$

**Figura 4.6 Número máximo de transacciones en un segundo.**

#### **4.6 LONGITUD DE LA RED Y TERMINADORES DE RED**

La interfase RS485, necesita terminadores de red para eliminar las reflexiones producidas por la impedancia característica que presenta el cable par trenzado de la red, la impedancia característica varía de 100 a 120 ohmios, dependiendo del tipo de cable que se utilice y por ello se debe incrementar un terminador en cada extremo de la red.

Dependiendo de la relación: velocidad de datos y velocidad de propagación, se puede determinar la distancia máxima en que la red puede funcionar sin necesidad de terminadores de red.<sup>19</sup> La figura 4.7 indica el cálculo de la distancia máxima posible sin utilizar terminadores de red.

Para velocidades de comunicación de datos de 115200 bps y distancias mayores de 78 metros (cálculo figura 4.7), se recomienda la utilización de terminadores de red, a los dos extremos de la misma.

$$\begin{aligned} \text{longitud del cable} &= 0.03 \cdot \frac{\text{Velocidad de la luz}}{\text{Velocidad de transmisión de datos}} \\ \text{longitud del cable} &= 0.03 \cdot \frac{300000000 \text{ m/s}}{115200 \text{ bps}} \\ \text{longitud del cable} &= 78 \text{ metros} \end{aligned}$$

**Figura 4.7 Cálculo de la distancia máxima de la red, sin terminadores.**

---

<sup>19</sup> TEXAS INSTRUMENTS, Interface circuit for TIA/EIA-485 (Design notes), Texas Instruments, 1998

#### **4.7 CONSUMO DE RECURSOS DEL NODO MAESTRO.**

El espacio necesario requerido en el disco duro para almacenar el software es: 9,60 MB, el tamaño de la aplicación MicroBUS versión 2.0.0.507 es 268KB, el tamaño de la librería WtOPCSvr.dll versión 9, 0, 0, 6 es: 356KB.

El consumo de recursos de memoria RAM y uso del procesador, depende del número de nodos a consultar y el número de clientes OPC. El consumo de recursos de memoria y procesador para ciertas condiciones, se indica en la tabla 4.15.

#### **4.8 EFECTOS DE CARGA DEL SERVIDOR OPC.**

El buffer de memoria del servidor de datos OPC se actualiza por parte de los nodos esclavos cada vez que estos son consultados por el protocolo de comunicación (mínimo cada 12 milisegundos).

El buffer de memoria del servidor OPC, se actualiza por parte de cada cliente del servidor OPC. Cada cliente tiene un tiempo de actualización de datos al que responde el servidor OPC.

El software implementado manipula 3584 elementos (tags), lo que corresponde a la definición de 32 grupos (nodos esclavos) de 112 elementos (tags). La aplicación desarrollada, puede manipular 11200 elementos (tags) en un segundo (actualiza 112 tags cada 10 milisegundos).

Un servidor de datos OPC basado en la biblioteca WtOPCSrv.dll puede manipular hasta 20000 elementos (tags) y responder a siete aplicaciones clientes en el mismo instante de tiempo.

**Tabla 4.15 Efectos de la conexión de clientes OPC al servidor OPC.**

Nodos esclavos	Consultas por segundo	Sin clientes OPC		OPC Quick Client		OPC Quick Client, LookOUT	
		Memoria (KB)	Procesador (%)	Memoria (KB)	Procesador (%)	Memoria (KB)	Procesador (%)
0	0	6736	0	7196	0	7216	0
1	10	7068	1	7544	2	7588	2
2	20	7208	2	7716	3	7744	3
3	30	7364	3	7852	4	7892	4
4	40	7484	6	7976	7	8008	8
5	49	7600	7	8068	9	8120	12

**Características del nodo maestro:**

Procesador: Intel 2806Mhz

Memoria RAM:512MB

Tarjeta principal: VIA Technologies P4M266A-8235

Sistema Operativo: Microsoft Windows XP professional. Versión 5.1.2600

Aplicación servidor OPC: MicroBUS. Versión 2.0.0.507

Tiempo de transmisión de cada trama (consulta) de la aplicación MicroBUS V 2 es 20 milisegundos

Cada nodo esclavo está configurado para ser monitoreado cada 100 milisegundos

La tabla 4.15, indica el consumo de recursos de la aplicación desarrollada, para el incremento de nodos y el incremento de aplicaciones clientes conectadas al servidor de datos OPC.

#### **4.9 EFECTOS DE FINALIZACIÓN DEL TIEMPO DEMOSTRATIVO DE LA BIBLIOTECA WTOPCSRV.**

El servidor de datos OPC basado en la biblioteca WtOPCSrv.dll funciona durante 30 minutos como periodo de tiempo demostrativo. Durante el periodo de demostración es permitida la utilización de todas las funciones de la biblioteca, también permite la manipulación de datos del buffer de memoria del servidor OPC por parte de las aplicaciones clientes del servidor OPC.

Al finalizar el periodo demostrativo, la biblioteca WtOPCSrv.dll permite que las aplicaciones clientes del servidor OPC puedan modificar el contenido de los elementos (ítems) del buffer de memoria asignada para el servidor OPC, pero no permite la lectura y la actualización del buffer de memoria del servidor de datos OPC por parte de las funciones de la biblioteca (la capa de aplicación suspende la comunicación con la capa de enlace de datos).

## **CAPÍTULO 5**

### **CONCLUSIONES Y RECOMENDACIONES.**

#### **5.1 CONCLUSIONES.**

- Se ha implementado un servidor de datos OPC, que permite monitorear a una red de microcontroladores comerciales utilizando una red RS485, esto permite desarrollar tecnología de bajo costo y la integración de tecnología de altas prestaciones en la pequeña industria existente en el país.
- El proyecto desarrollado responde, satisfactoriamente en la utilización de la especificación para servidores de datos OPC y cumple con el estándar RS485.
- El proyecto desarrollado funciona en cualquier sistema SCADA que soporte la especificación propuesta por la Fundación OPC (Enlace de Objetos Incrustados para Control de Procesos).
- Se demostró el correcto funcionamiento del protocolo de comunicación en la red RS485 con 7 nodos esclavos conectados a una distancia de 27 metros, funcionando a una velocidad de 115200 bps.
- Las interfaces gráficas desarrolladas en el software de la estación maestra son: configuración del servidor, creación de tópicos, configuración del puerto de comunicaciones, administración de usuarios, análisis de tramas, estado del servidor, estado de la red, búsqueda de dispositivos, modificación de la dirección física de un dispositivo de la red y lista de los dispositivos creados.

- El software de control de protocolo y servicio de datos OPC se elaboró utilizando el lenguaje de programación Visual Basic 6.0. Este software de programación permite crear aplicaciones ejecutables y de bajo costo.
- Se implementó el servidor de datos OPC utilizando la biblioteca WtOPCSrv.dll ya que permite el desarrollo rápido de servidores OPC.
- El software desarrollado responde satisfactoriamente en el mismo instante de tiempo a cuatro clientes OPC.
- El servidor de datos OPC desarrollado en base a la biblioteca WtOPCSrv.dll permite administrar el control de los clientes, cuando éstos manipulan la información contenida en un elemento.
- No es posible ejecutar dos servidores de datos OPC para un mismo sistema, ya que crea conflictos de hardware.
- Los elementos definidos en el servidor de datos OPC, permite a los clientes OPC acceder a la información de los nodos esclavos de la red.
- El registro en el sistema operativo del servidor de datos OPC, permite que un cliente OPC pueda iniciar la aplicación del servidor OPC.
- La velocidad de actualización de datos de los clientes del servidor OPC, depende de cada uno de los clientes OPC y no del servidor de datos OPC.
- Para manipular las variables en un cliente OPC (software HMI), se debe conocer como operan los algoritmos de control, debido a que su modificación afecta al proceso, la maquinaria y al personal.
- No pueden existir dos estaciones con la misma dirección. Si existen dos estaciones con la misma dirección, se generará colisión de datos por la transmisión simultánea de los

nodos esclavos. Dicha información es rechazada en la detección de errores del nodo maestro.

- Se ha construido un hardware para monitoreo de datos de bajo costo para ello se ha empleado elementos existentes en el mercado nacional como son: el microcontrolador PIC16F628 y el transceiver MAX485.
- El protocolo de comunicación desarrollado se adapta a las características de cualquier microcontrolador de la familia Microchip, para ello requiere mínimo de 133 localidades de memoria RAM, 3 localidades de memoria EEPROM y de 480 palabras en la memoria de programas.
- El programa de los nodos esclavos se desarrolló con el software MPLAB ya que permite controlar de manera avanzada el hardware del microcontrolador PIC16F628A.
- Cada byte que se transmite por la red RS485 produce una interrupción de recepción en los nodos esclavos conectados en la red, menos en el nodo que transmite la trama. Cada nodo esclavo analiza si el byte pertenece o no a dicho nodo esclavo.
- El estándar RS485 no define ninguna topología pero sugiere el uso de las topologías: cadena margarita (daisy chain) y bus.
- El diseño de hardware y el desarrollo de software para el estándar RS485, es más sencillo y más barato en comparación con otras redes industriales.
- La longitud del cable de red puede sobrepasar los 1200 metros con la utilización de repetidores o segmentación de la red.
- La utilización de terminadores incrementa el consumo de corriente en los transceivers RS485, por lo cual debe considerarse en la operación del sistema.

- La utilización de resistencias de polarización (bias resistor) en la red RS485, permite mantener en un estado definido de voltaje en los receptores cuando éstos están funcionando en alta impedancia.
- El monitoreo de los nodos esclavos (microcontroladores comerciales) responden correctamente al utilizar el método de acceso al medio “maestro/esclavo”.
- El algoritmo de consultas (polling) selecciona el nodo a consultar dependiendo del orden del dispositivo en la lista de dispositivos definidos y del tiempo mínimo de consulta que éste tenga.
- Para la comunicación de datos se utilizó la codificación binaria, ya que ésta permite transmitir mayor cantidad de información que la transmisión basada en codificación ASCII.
- El algoritmo de consultas (polling) desarrollado en el nodo maestro, trabaja con un bloque de memoria definido en los nodos esclavos, esto limita el monitoreo de la memoria de los nodos esclavos.
- Se puede incrementar el número de nodos de 32 a 128 si la impedancia de entrada de todos los transceivers RS485 aumenta de 12 k $\Omega$  a 48 k $\Omega$ .
- Para transmitir a velocidades mayores de 115200 bps con una longitud de la red igual a 78 metros o para una longitud de la red mayor de 78 metros con una velocidad de transmisión igual a 115200 bps, se debe utilizar terminadores en los extremos de la red.
- El proyecto desarrollado es capaz de adaptarse a diferentes tipos de aplicaciones (industria, oficina, vivienda, wireless y otras) que requieran ser monitoreadas manteniendo el estándar industrial RS485 y la especificación OPC.

## **5.2 RECOMENDACIONES.**

- Para aumentar el rendimiento del servidor de datos OPC, la aplicación desarrollada debe funcionar en sistemas operativos basados en tecnología NT de Microsoft.
- Para que no disminuya el rendimiento en la velocidad de acceso a la medio de la aplicación desarrollada en el nodo maestro, el sistema operativo debe ejecutar la mínima cantidad de tareas en el mismo instante de tiempo.
- Antes de desinstalar la aplicación desarrollada en el nodo maestro, se recomienda remover el registro del servidor OPC.
- Para aislar la red RS485 del hardware de control (microcontrolador), se puede utilizar el circuito integrado MAX1480.
- Para mantener la especificación RS485 todos los nodos deben mostrar una conexión común a masa y debe ser aislada de tierra.
- No se debe utilizarse la dirección 0 en las estaciones esclavas, ya que es una dirección reservada.
- No se debe transmitir a los nodos esclavos la trama de difusión que afecte localidades de memoria que contengan configuraciones del dispositivo.
- Para mejorar el rendimiento en la detección de tramas, la detección del campo de inicio debería ser realizada por hardware.
- Para que exista mayor seguridad de la información recibida, se recomienda la utilización del CRC16 de CCITT.
- El campo de datos debe ser limitado, esto depende del tamaño del banco de la memoria del microcontrolador.

- Para mejorar el rendimiento del software existente en el microcontrolador debe implementarse la interrupción del programa principal en la recepción y transmitir de cada byte de la trama utilizada en el protocolo de comunicación.
- Para implementar el proyecto propuesto en cualquier instalación, se debe usar conectores y protecciones apropiados al medio de trabajo.
- Para que una trama sea detectada como inicio, se debe analizar el campo de inicio de trama y de dirección.
- Para evitar perturbaciones en el programa principal cuando la temperatura supera los 65 grados centígrados, se debe parquear la memoria EEPROM en una localidad con el dato 255d.
- Para disminuir la complejidad en la programación, debe disponerse de herramientas de programación y aumentar el nivel de aprendizaje. Se debe utilizar programas de desarrollo de software de alto nivel (Basic, C, etc.), para la generación del software de microcontroladores.
- Para adquirir nuevos conocimientos en redes industriales, se recomienda realizar prácticas de laboratorio con microcontroladores IPC@CHIP de la empresa BECK, que incluyen en su arquitectura el periférico de interfase de red Ethernet.

## BIBLIOGRAFÍA

- ANSI/ISA-S51.1 , Standart Process Instrumentation Terminology, Instrument Society of America, California, 1979
- B&B ELECTRONICS, RS422 and RS485 Application Note, [www.bb-elec.com](http://www.bb-elec.com), 1997
- BALCELLS, Joseph, Interferencias Electromagnéticas En Sistemas Electrónicos, Marcombo, México, 1992
- CATANIA, Carlos, TCP/IP sobre sistemas embebidos y su implantación sobre la arquitectura PIC, Universidad Marcelino Champagnat , 2005
- CORRALES, Luis, “Interfaces de Comunicación Industrial”, EPN, Ecuador, 2004
- DALLAS, "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor iButton Products", Maxim/Dallas.
- GALARZA, Eddie, “Comunicaciones para Instrumentación”, ESPE, Ecuador,2004
- KEPWARE, "Omronhl32, Siemens S7-200 Device Driver Help", USA, 1998.
- MAXIM, MAX220 – MAX249, [www.maxim-ic.com](http://www.maxim-ic.com), 1996
- MAXIM, RS422/RS485 Transceivers, [www.maxim-ic.com](http://www.maxim-ic.com), 1996

- NRF-046-PEMEX-2003, “Protocolos de Comunicación en Sistemas de Monitoreo y Control”, PEMEX, México, 2003
- OPC FOUNDATION. OPC Overview: OPC DataAccess Overview. p. 11
- SELDON, Tom, “Enciclopedia de redes”, McGraw-Hill, España, 1994
- TEXAS INSTRUMENTS, Interface circuit for TIA/EIA-485 (Design notes), Texas Instruments, Literature number SLLA036, 1998
- [http://diea-gw.diea.ulpgc.es/users/aurelio/int\\_equipos/trab9899/ole\\_dde\\_opc/index.html](http://diea-gw.diea.ulpgc.es/users/aurelio/int_equipos/trab9899/ole_dde_opc/index.html)
- <http://personal.redestb.es/efiguera/memoria.htm>
- <http://www.angelfire.com/mb/chang/OSIIEEE.html>
- <http://www.eui.upv.es/ineit-mucon/Applets/TutorialTCPIP/isoosi.html>
- <http://www.opcfoundation.org>
- [http://www.pchardware.org/redes/redes\\_osi.htm](http://www.pchardware.org/redes/redes_osi.htm)
- [http://www.rockwellsoftware.com/corporate/softauto/sa99\\_2/hmirise\\_sp.cfm](http://www.rockwellsoftware.com/corporate/softauto/sa99_2/hmirise_sp.cfm)

## GLOSARIO

**.NET.-** Software de desarrollo de aplicaciones y componentes. Desarrollado a partir de COM+ 2.0.

**Adaptador de interfase.-** Interfase de comunicación. Convierte de RS232 a RS485 y viceversa.

**Ancho de Banda.** Es usado para indicar la rapidez de flujo de los datos en un canal de comunicación. Generalmente hablando, el ancho de banda es proporcional a la cantidad de datos transmitidos o recibidos por unidad de tiempo. En sistemas digitales, ancho de banda es expresado como velocidad de datos (en bits) por segundo (bps). En sistemas analógicos, el ancho de banda es expresado en términos de la diferencia entre la componente de señal de frecuencia más alto y la componente de señal de frecuencia más bajo.

**API.-** (Application programming interface). Mensajería entre aplicaciones, es un conjunto de funciones, tipos y mensajes pre-definidos para poder programar sobre los sistemas operativos de Microsoft.

**ASM.-** (Assembler). Código ensamblador.

**AWG.-** (American Wire Gauge). Área seccional del conductor.

**Banco de memoria.-** Grupo de localidades en un espacio especificado de memoria.

**BIT.-** Es la unidad más pequeña de información y la unidad base en comunicaciones.

**BPS.-** Es una medida para la velocidad de transmisión de datos, la cantidad de Bits que se transmiten por unidad de tiempo. Este término sustituye al término Baud debido a que un cambio de estado electrónico puede implicar varios bits.

**Byte.-** Conjunto de bits continuos mínimos que hacen posible un direccionamiento de información en un sistema computarizado. Está formado por 8 bits.

**Cache.-** Bloque de memoria que contiene información utilizada frecuentemente.

**Cliente.-** Sistema capaz de acceder a un recurso compartido.

**CCITT.-** Comité Consultivo Internacional Telefónico y telegráfico.

**CMOS.-** (Complementary Metal Oxide Silicon). Silicio de óxido metálico.

**Codificación.-** Códigos metódicos y sistemáticos.

**Códigos.-** Acuerdos previos sobre un conjunto de significados que definen una serie de símbolos y caracteres.

**COM.-** (Common Object Model). Modelo de objetos genéricos. Es un conjunto de redes distribuidas diseñadas para permitir a las aplicaciones Windows acceder a las aplicaciones y objetos sobre otras plataformas, COM integra un sistema de vinculación de objetos OLE.

**COM-Interop.-** Véase .NET.

**Comunicación asíncrona.-** Envío de datos no acoplado a una señal de reloj.

**Comunicación sincrónica.-** Envío de datos acoplado a una señal de reloj.

**Controlador(es).-** Dispositivo que opera automáticamente para regular una variable controlada.

**CPU.-** (Center Process Unit). Unidad de procesamiento central.

**CRC.-** (Cyclic Redundance Check). Comprobación de redundancia cíclica, permite una verificación fiable de la consistencia de la trama.

**CSMA/CD.-** Detección de portadora de acceso múltiple / detección de colisiones.

**DCE.-** (Data Communications Equipment). Equipos de conversión entre el DTE y el canal de transmisión.

**DCOM.-** (Distributed Common Object Model), Es una versión de COM para redes.

**DDE.-** Intercambio dinámico de datos, usado entre aplicaciones con DLL.

**DLL.-** Biblioteca para vínculos dinámicos.

**DTE.-** (Data Terminal Equipment), equipos que son la fuente y destino de los datos. Comprenden equipos de computación (Host, Microcomputadores y Terminales).

**E/S.-** Entradas y salidas.

**EEPROM.-** Memoria de solo lectura, programable eléctricamente.

**EIA/TIA.-** Asociación de industrias electrónicas/Asociación de industrias de telecomunicaciones.

**EMI.-** Interferencias electromagnéticas.

**Emisor.-** Dispositivo que recepta la transmisión de datos.

**Encapsular.-** Agrupar la información.

**Esclavo.-** Dispositivo que responde a un maestro, no puede iniciar una comunicación. Los dispositivos deben tener direcciones únicas.

**Fibra óptica.-** Cable utilizado para la comunicación digital, utiliza fotones para transmitir la información.

**Flash memory.-** Es una memoria que puede ser grabada y borrada en el circuito embebido.

**GPR.-** Registros de propósito general de microcontroladores Microchip.

**Grupo objeto.-** Objeto que almacena ítems, sean de un dispositivo físico o alguna clasificación de ítems.

**GUI.-** (Graphical User Interface). Interfase gráfica de usuario.

**Handshaking.-** Protocolo de acuerdo mutuo, utilizado en las comunicaciones.

**Hardware.-** Conjunto de componentes electrónicos, eléctricos y mecánicos que soportan la información y realizan operaciones para las cuales fueron diseñados.

**Header.-** Cabeza de la trama de comunicación.

**HMI.-** (Human Machine Interface). Interfase entre el humano y la máquina.

**Host Link.-** Protocolo de comunicación utilizado por el PLC Omron de modelo CQM1.

**I/O Server.-** Software de comunicación dispositivo-PC.

**Interfaces.-** Conexión que permite la comunicación entre dos o más entidades.

**Interoperabilidad.-** Característica que permite interconectar dispositivos de diversos fabricantes para que operen en forma integrada en una aplicación determinada.

**ISO.-** Organización internacional para la estandarización.

**Ítem.-** Nombre de una localidad de memoria, la cual almacena una variable.

**ITP Cable.-** Cable trenzado para uso industrial.

**JavaScript.-** Lenguaje de secuencia de comandos, que utilizan los clientes.

**LAN.-** (Local Area Network). Red de área local.

**MAC.-** “Media access control” Control de acceso al medio.

**Maestro.-** Estación que administra las comunicaciones en un método de acceso al medio, Maestro/Esclavo.

**Masa.-** terminal común de un circuito, conductor de referencia de potencial cero con respecto a su fuente de alimentación.

**MCU.-** Microcontrolador.

**Medio.-** Consiste en el recorrido de los datos desde el origen hasta su destino.

**Mpsasmwin.-** Ensamblador incorporado en el Mplab.

**Mplab IDE.-** Software de desarrollo integrado que permite: escribir, depurar y optimizar los programas para microcontroladores. De la empresa Microchip.

**Multiconexión (multidrop).-** Es un sistema de comunicación en el cual más de dos dispositivos son conectados en un medio de transmisión único. En dicho sistema cada dispositivo debe tener una dirección única.

**Nodo.-** Lugar donde se conectan uno o más dispositivos en un canal de comunicación.

**NR.-** Codificación retorno a cero.

**NRZ.-** Codificación no retorno a cero.

**Objeto Servidor.-** Provee una vía de acceso de lectura o escritura de la información utilizando comunicación con las fuentes de datos. La comunicación del objeto servidor y el objeto cliente, se realiza por medio de interfaces.

**ODBC.-** Inter conectividad de bases de datos abierta.

**OLE.-** Objetos vinculados e insertados.

**OPC.-** (Ole for Process Control). OLE para Control de Procesos.

**OSI.-** Interconexión de sistemas abiertos.

**PC.-** Computador personal, unión de hardware y software.

**PLC.-** Controlador Lógico Programable, sistema lógico que combina hardware y software, de modo que sus instrucciones codificadas puedan modificarse (reprogramarse); proporciona automatización “flexible” en vez de “dedicada”.

**Polling.-** En una red de dispositivos, es el método por el cual un dispositivo maestro solicita información en forma secuencial a cada uno de los dispositivos esclavos conectados a la red. Cuando el dispositivo esclavo es interrogado, tiene la oportunidad de transmitir información hacia el maestro.

**PPI.-** (Point to point interface). Protocolo de comunicación utilizado por el PLC Siemens S7-200 CPU224.

**Protocolo.-** Conjunto de reglas que posibilitan la transferencia de datos entre dos o más computadores.

**RAM.-** (Random access memories). Memoria de escritura y lectura de tipo volátil.

**Receptor.-** Dispositivo de destino de los datos.

**Red.-** Conjunto de computadoras, terminales, periféricos, equipos de control, etc., a través de un medio físico o inalámbrico.

**Redundancia.-** Uso de elementos o sistemas múltiples, de igual o diferente tecnología, para desempeñar la misma función.

**Repetidor.-** Es el dispositivo más sencillo para la interconexión de redes, que proporciona una simple regeneración de la señal.

**RFI.-** Interferencia de radio frecuencia.

**RISC.-** (Reduced Instrucción Set Computer). Conjunto de instrucciones reducidas para el procesamiento.

**RS-232C.-** Norma internacional de transmisión serial entre dos dispositivos, que utiliza señales de voltaje para comunicarse. Debido a que se emplea una señal de voltaje, el RS-232C no puede transmitir señales a largas distancias (máximo 15 metros).

**RS-485.-** Norma internacional de comunicación serial que permite entablar la comunicación entre 32 dispositivos sobre un mismo canal. Esta norma tiene la característica de transmisión diferencial y la norma permite la comunicación entre dos dispositivos separados a una distancia máxima de 1200 metros.

**S.O.-** Sistema operativo.

**SCADA.-** (Supervisory Control And Data Acquisition). Control por Supervisión y Adquisición de Datos.

**Servidor.-** Realiza la distribución de la información centralizada. Es un sistema capaz de compartir sus recursos.

**SFR.-** Registros de funciones especiales, permiten la configuración de microcontroladores de la empresa Microchip.

**Sistema de Control Distribuido.-** Es una red de procesadores digitales de información, con sistema operativo distribuido y procesamiento en tiempo real operando bajo los conceptos de la teoría de control automático.

**Sistema de comunicación.** Conjunto de equipos y dispositivos que permiten transmitir información a través de un medio de comunicación.

**Sistema digital de monitoreo y control.-** El término hace referencia a cualquier dispositivo basado en instrumentación y sistemas de computadoras o bien basados en microprocesadores, para funciones de control o de adquisición de datos.

**Software.-** Conjunto de instrucciones (programas) y datos que, almacenados en la memoria de la máquina, describen el trabajo a realizar.

**Tag.-** Se define como una variable y es el valor físico.

**TagName.-** Es el nombre de la variable o del Tag.

**TCY.-** Tiempo de instrucción, necesario para completar una instrucción.  $TCY = F_{osc} / 4$ .

**Tierra.-** (Earth) Potencial de la tierra física.

**Topología.-** Define la estructura mediante la cual los diferentes dispositivos usados en el sistema digital de monitoreo y control se interconectan.

**Trama de Difusión (Broadcast).-** En una red Maestro/Esclavo, se produce cuando un mensaje (trama) es transmitido por el maestro y es aceptado por todas las estaciones esclavas. Ninguna estación esclava responde a dicho mensaje.

**Trama.-** Tira de bits con un formato predefinido usado en protocolos orientados a bit.

**Transceiver.-** Convertidor de señales.

**Transmisión Half-Duplex.-** La transmisión de los datos se produce en ambos sentidos pero alternativamente, en un solo sentido a la vez. Si se está recibiendo datos no se puede transmitir.

**TTL.-** Lógica transistor - transistor, circuitos integrados de la serie 74XXX

**UART.-** (Universal Asynchronous Receiver Transmitter). Transmisor receptor asíncrono universal, permite controlar la interfase RS-232.

**USB.-** Bus serie universal.

**UTP.-** Cable par trenzado sin apantallamiento.

**Variables críticas de proceso.-** Son aquellas variables de proceso en donde la velocidad de actualización del valor medido, es requerida en un número mayor a 4 veces por segundo. Esto significa que la actualización de datos se debe realizar en un lapso de tiempo menor a 250 ms ya sea para propósitos de monitoreo y/o de control.

**Variables no críticas de proceso.-** Son aquellas variables de proceso en donde la velocidad de actualización del valor medido, es requerida en un número menor a 4 veces en el lapso de un segundo. Esto significa que la actualización de datos se debe realizar en un lapso de tiempo igual o mayor a 250 ms, (por ejemplo 270 ó 300 ms), ya sea para propósitos de monitoreo y/o control.

**VBScript.-** Lenguaje de secuencia de comandos usados al lado del cliente por VB6.0.

**VB6.0.-** Lenguaje de programación Microsoft Visual Basic versión 6.0

**VC++.-** Lenguaje de programación Microsoft Visual C ++

**WTOPCSrv.dll.-** Biblioteca para el desarrollo de servidores OPC de la empresa WinTech.

# ANEXO A

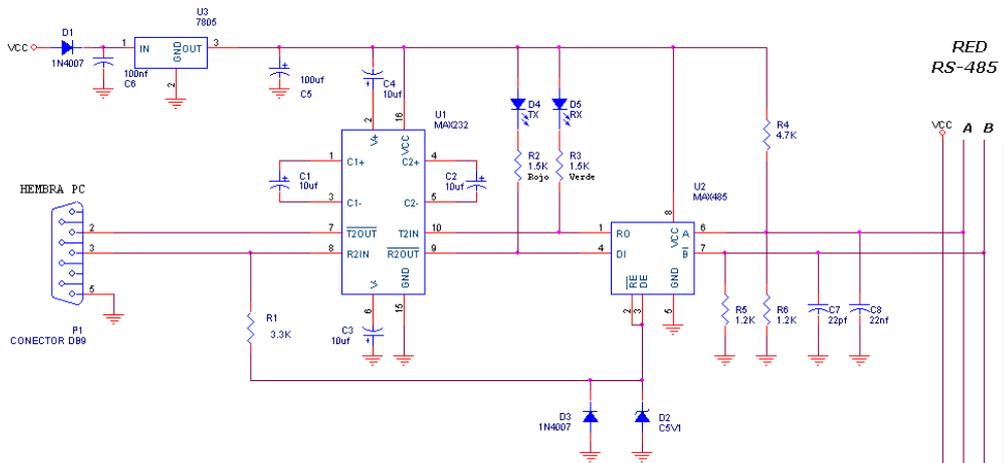
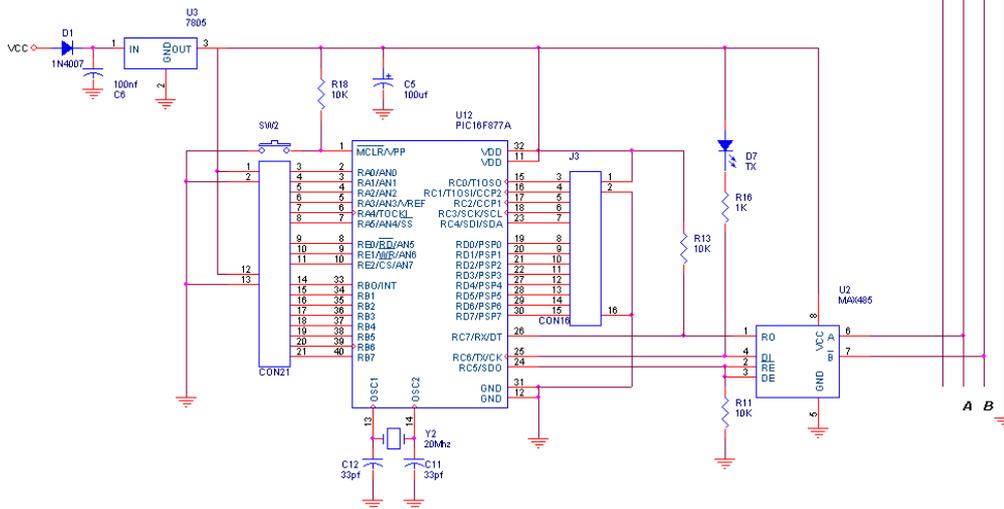


Figura A.1 Diagrama eléctrico del hardware implementado.



## ANEXO B

DATOS BOLEANOS ON/OFF	BANCO				DATOS ENTEROS 16 BITS	BANCO			
	0	1	2	3		0	1	2	3
<b>D00-D07</b>	50	178	306	434	<b>A00 (Alto)</b>	80	208	336	464
<b>D10-D17</b>	51	179	307	435	<b>A00 (Bajo)</b>	81	209	337	465
<b>D20-D27</b>	52	180	308	436	<b>A01 (Alto)</b>	82	210	338	466
<b>D30-D37</b>	53	181	309	437	<b>A01 (Bajo)</b>	83	211	339	467
<b>D40-D47</b>	54	182	310	438	<b>A02 (Alto)</b>	84	212	340	468
<b>D50-D57</b>	55	183	311	439	<b>A02 (Bajo)</b>	85	213	341	469
<b>D60-D67</b>	56	184	312	440	<b>A03 (Alto)</b>	86	214	342	470
<b>D70-D77</b>	57	185	313	441	<b>A03 (Bajo)</b>	87	215	343	471
<b>D80-D87</b>	58	186	314	442	<b>A04 (Alto)</b>	88	216	344	472
<b>D90-D97</b>	59	187	315	443	<b>A04 (Bajo)</b>	89	217	345	473
<b>DATOS ENTEROS 8 BITS</b>					<b>A05 (Alto)</b>	90	218	346	474
<b>I00</b>	60	188	316	444	<b>A05 (Bajo)</b>	91	219	347	475
<b>I01</b>	61	189	317	445	<b>A06 (Alto)</b>	92	220	348	476
<b>I02</b>	62	190	318	446	<b>A06 (Bajo)</b>	93	221	349	477
<b>I03</b>	63	191	319	447	<b>A07 (Alto)</b>	94	222	350	478
<b>I04</b>	64	192	320	448	<b>A07 (Bajo)</b>	95	223	351	479
<b>I05</b>	65	193	321	449	<b>A08 (Alto)</b>	96	224	352	480
<b>I06</b>	66	194	322	450	<b>A08 (Bajo)</b>	97	225	353	481
<b>I07</b>	67	195	323	451	<b>A09 (Alto)</b>	98	226	354	482
<b>I08</b>	68	196	324	452	<b>A09 (Bajo)</b>	99	227	355	483
<b>I09</b>	69	197	325	453	<b>MENSAJE 10 CARACTERES</b>				
<b>I10</b>	70	198	326	454	<b>M00 (0)</b>	100	228	356	484
<b>I11</b>	71	199	327	455	<b>M00 (1)</b>	101	229	357	485
<b>I12</b>	72	200	328	456	<b>M00 (2)</b>	102	230	358	486
<b>I13</b>	73	201	329	457	<b>M00 (3)</b>	103	231	359	487
<b>I14</b>	74	202	330	458	<b>M00 (4)</b>	104	232	360	488
<b>I15</b>	75	203	331	459	<b>M00 (5)</b>	105	233	361	489
<b>I16</b>	76	204	332	460	<b>M00 (6)</b>	106	234	362	490
<b>I17</b>	77	205	333	461	<b>M00 (7)</b>	107	235	363	491
<b>I18</b>	78	206	334	462	<b>M00 (8)</b>	108	236	364	492
<b>I19</b>	79	207	335	463	<b>M00 (9)</b>	109	237	365	493

**Figura A.2 Representación de la información (usada por la trama de consultas), en las localidades de memoria del microcontrolador.**

## ANEXO C

### MAESTRO

TRAMA DE TRANSMISIÓN



### ESCLAVO

TRAMA DE TRANSMISIÓN



#### CAMPO DE INICIO x3



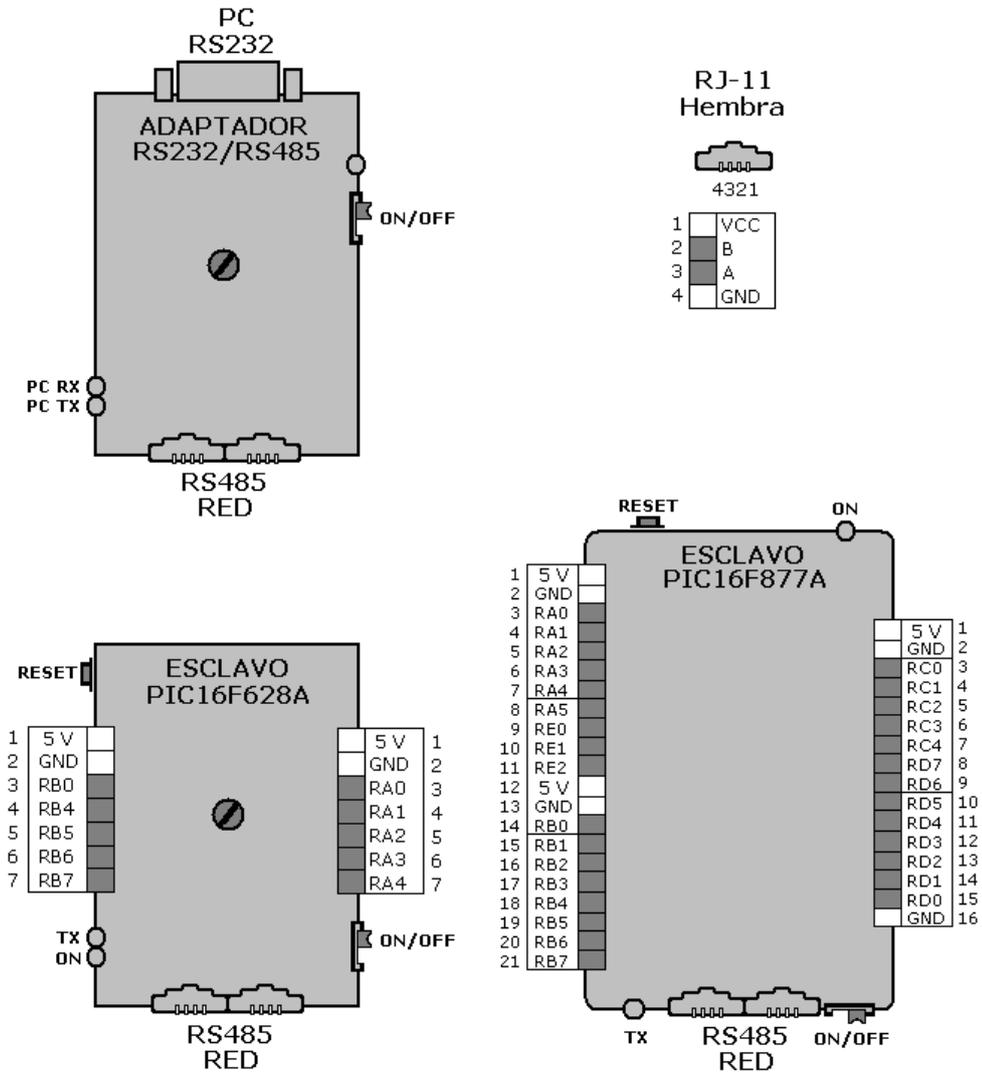
#### CAMPO DE CONTROL



- FSREE** 1: Busca la memoria de datos "EEPROM" no volátil. No utiliza los bits "RBXf".  
 0: Busca la memoria de datos "static RAM" ó los registros de propósito general y los registros de funciones especiales para ello utiliza los bancos "RBXf".
- Rwf** 1: Trama de lectura de datos.  
 0: Trama de escritura de datos.
- RBXf** 1/0: Configura el banco a usar en el microcontrolador, cuando FSREE=0

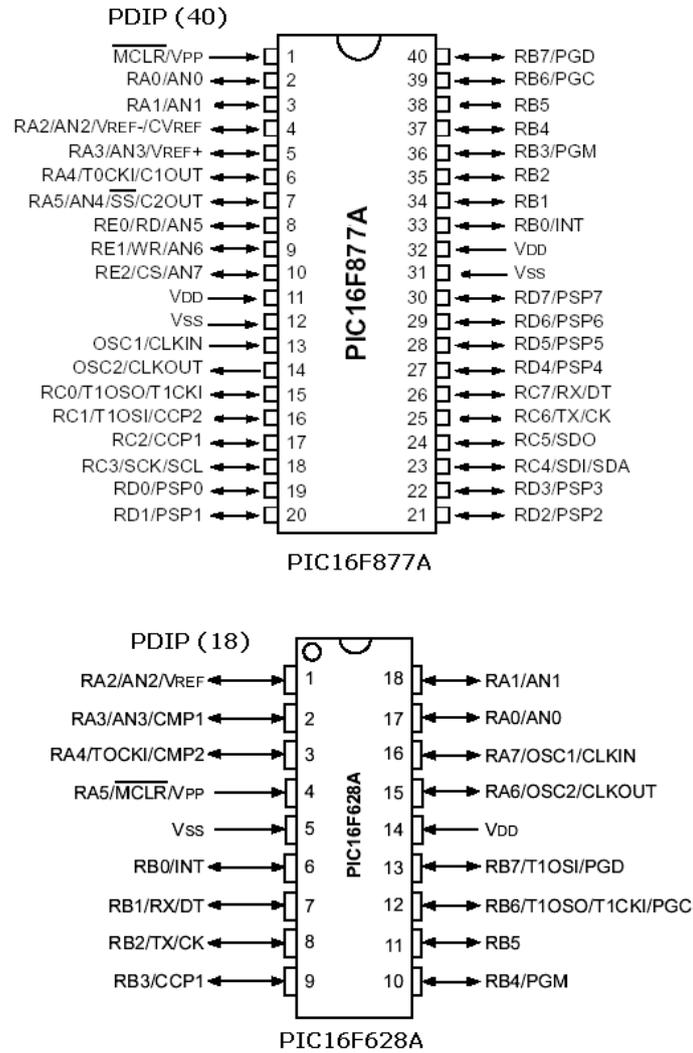
**Figura A.3** Tramas implementadas para la comunicación de datos.

## ANEXO D



**Figura A.4** Terminales de conexión de los módulos construidos.

## ANEXO E



**Figura A.5 Disposición de pines de los microcontroladores: PIC16F877A y PIC16F628A.**



## ANEXO G

	Localidad		Localidad		Localidad		Localidad
Indirect addr. <sup>(1)</sup>	00h	Indirect addr. <sup>(1)</sup>	80h	Indirect addr. <sup>(1)</sup>	100h	Indirect addr. <sup>(1)</sup>	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
	07h		87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
	0Dh		8Dh		10Dh		18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh		8Fh		10Fh		18Fh
T1CON	10h		90h				
TMR2	11h		91h				
T2CON	12h	PR2	92h				
	13h		93h				
	14h		94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah	EEDATA	9Ah				
	1Bh	EEADR	9Bh				
	1Ch	EECON1	9Ch				
	1Dh	EECON2 <sup>(1)</sup>	9Dh				
	1Eh		9Eh				
CMCON	1Fh	VRCON	9Fh		11Fh		1EFh
	20h		A0h		120h		
GPR 80 Bytes	20h - 6Fh	GPR 80 Bytes	A0h - EFh	GPR 48 Bytes	11Fh - 14Fh		
16 Bytes	6Fh - 70h	acceso 70h-7Fh	EFh - F0h		14Fh - 150h		
	70h - 7Fh		F0h - FFh	acceso 70h-7Fh	150h - 16Fh		
	7Fh		FFh	acceso 70h-7Fh	16Fh - 170h		1EFh - 1F0h
Banco 0		Banco 1		Banco 2		Banco 3	

Localidad de memoria no implementada, se lee como 0 lógico.  
 Nota 1: No es un registro físico.

**Figura A.7 Memoria RAM del PIC16F628A.**

## ANEXO H MANUAL DE USUARIO

### H.1. INTRODUCCIÓN.

La aplicación MicroBUS permite monitorear una red de microcontroladores con el protocolo MicroBUS y servir los datos por medio de la especificación de la Fundación OPC a las aplicaciones clientes de OPC.

### H.2. INSTALACIÓN DEL SOFTWARE.

El CD es auto ejecutable y corre directamente el documento basado en HTTP. Caso contrario dicha aplicación puede ser iniciada explorando el CD y ejecutando el archivo MicroBUS.html.

Cuando inicia el CD, automáticamente se puede verificar el contenido de éste en forma visual, como lo indica la figura A.8.

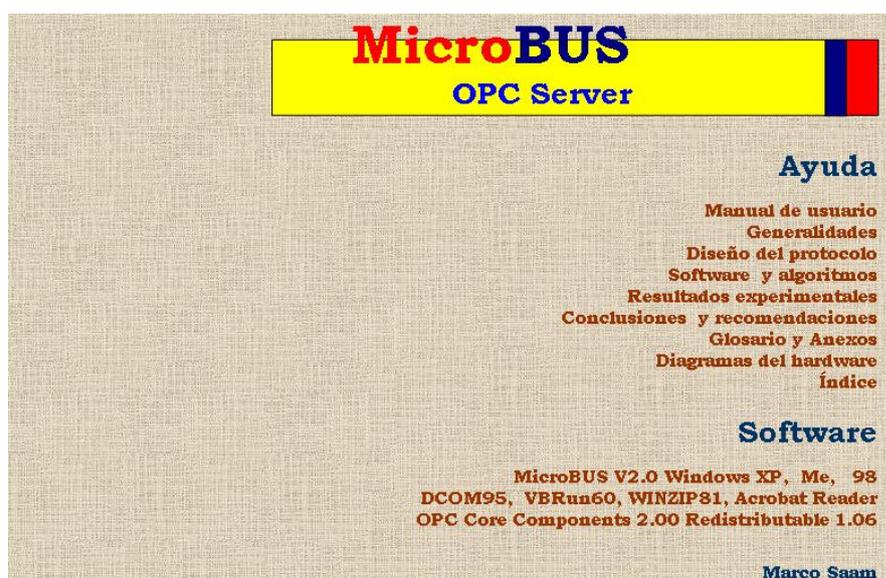
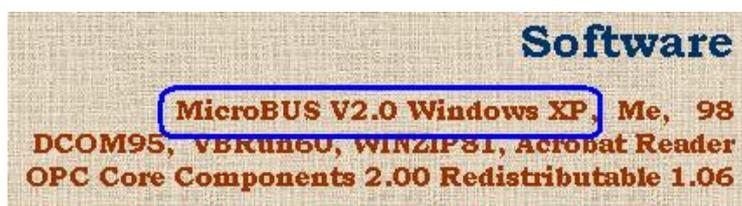


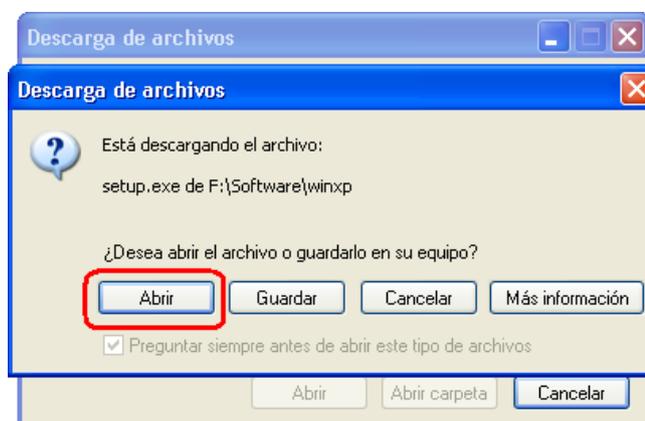
Figura A.8. Ejecución automática del CD.

Para instalar el software MicroBUS, de un clic izquierdo sobre el software que desee instalar esto depende del sistema operativo en que se desee trabajar. En el caso de instalar el software en el sistema operativo Microsoft Windows XP observe la figura A.9.



**Figura A.9. Selección del software a instalar.**

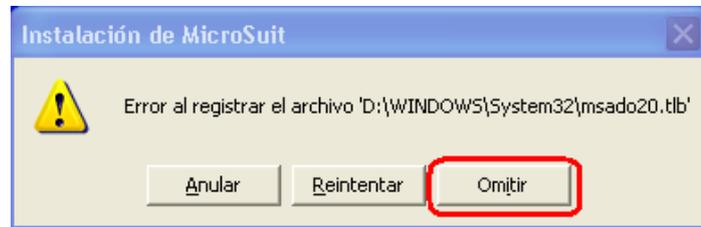
Después de dar clic sobre el software que se desee instalar, la página MicroBUS accede a la descarga del archivo. Esta interfase permite instalar la aplicación MicroBUS, refiérase a la figura A.10. Si usted desea copiar el archivo, explore el CD y copie toda la carpeta que contenga la versión requerida para su sistema operativo.



**Figura A.10. Ejecución de la instalación del software MicroBUS.**

Prosiga la instalación del software hasta su finalización, si existen componentes previamente instalados en el sistema operativo, el software le preguntará si desea reinstalarlos, se recomienda no reinstalar ningún componente. En caso de que existan componentes ejecutándose en ese instante de tiempo, omita la copia de dichos

componente o el registro de dicho componente, esto se puede visualizar en la figura A.11. Si usted anula la instalación de algún componente que haya presentado error la aplicación de instalación procede a desinstalar los componentes instalados.



**Figura A.11. Omisión en errores en la instalación del software.**

Al instalarse el software, la aplicación crea el directorio:

\Archivos de programa\MicroSUIT.

En dicho directorio se almacenan los archivos:

MicroBUS.exe, aplicación ejecutable.

MicroBUS.EXE.manifest, archivo de manejo de estilos visuales para XP

MicroBUS.mdb, almacena las configuraciones de los nodos.

Agradecimientos.txt, agradecimientos técnicos.

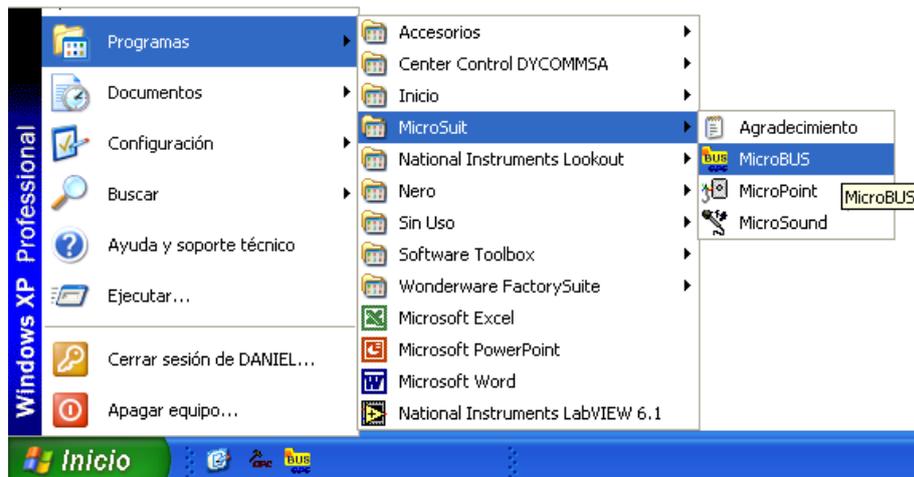
ST6UNST, archivo de registro que permite la desinstalación de la aplicación.

MicroBUSv628A.asm, código fuente del protocolo MicroBUS.

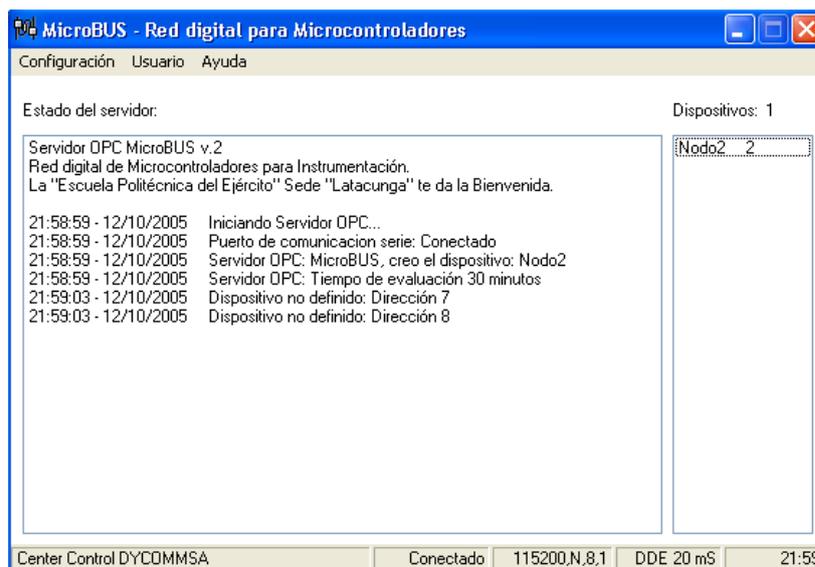
### **H.3. INICIO DEL PROGRAMA.**

Después de una instalación exitosa del software, la figura A.12 indica la manera de ejecutar el software.

Al iniciar la aplicación ésta se visualiza como lo indica la figura A.13. La aplicación debe ser configurada para que pueda operar según las conveniencias requeridas. Su interfase gráfica fue desarrollada usando como patrón las interfases graficas de varios software de monitoreo industrial.



**Figura A.12. Inicio de la aplicación MicroBUS.**



**Figura A.13. Inicio del software MicroBUS.**

#### **H.4. DESCRIPCIÓN DEL FORMULARIO PRINCIPAL.**

Antes de iniciar la configuración del software, se describe la manera de acceder a las herramientas que el software usa para este fin. Para acceder a las configuraciones del software puede hacerlo desde varios lugares, esto es con el fin de permitir una rápida configuración, se puede acceder desde: la barra de menús, la barra de estado y por los

controles de las listas ubicadas en el centro del formulario principal de la aplicación MicroBUS.

Los menús del software son: configuración, usuario y ayuda. El menú de configuración se indica en la figura A.14.



**Figura A.14. Herramientas del menú de configuración.**

La figura A.14, indica las herramientas que permiten configurar el sistema de monitoreo y el método de intercambio de datos del servidor.

Las herramientas de configuración del puerto de comunicación serie, definición de dispositivos y configuración del servidor, se describen posteriormente. Antes de acceder a dichas herramientas, el usuario debe ser reconocido como usuario de la aplicación.



**Figura A.15. Herramientas del menú de usuario.**

La figura A.15 indica las herramientas de administración de usuarios y de análisis de tramas, los cuales se describen posteriormente.

El menú Servidor OPC/Iniciar registro OPC reconfigura el registro de Windows para permitir que el servidor de datos OPC pueda ser iniciado automáticamente por cualquier

cliente OPC. El menú Servidor OPC/Eliminar registro OPC elimina las configuraciones del registro de Windows. Si la aplicación no se encuentra registrada en el sistema operativo esta no puede ser localizada por los clientes OPC. Antes de desinstalar la aplicación definitivamente se recomienda eliminar las configuraciones del registro de Windows. (Eliminar registro OPC).

La figura A.16, muestra el contenido del menú de Ayuda. Este menú sólo permite la visualización del formulario “Acerca de MicroBUS”, como lo indica la figura A.17.



**Figura A.16. Descripción del menú de ayuda.**



**Figura A.17. Acerca de la aplicación.**

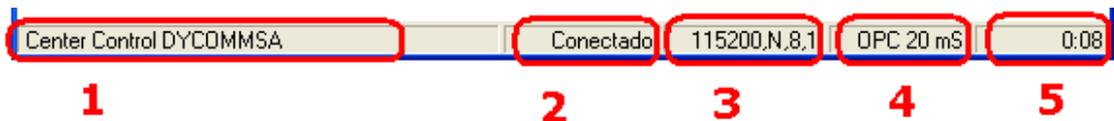
La figura A.18 corresponde a la barra de estado de la aplicación. Al pulsar, con doble clic sobre el área indicada como 1 borra la lista de “Estado del servidor” (dicha lista ubicada en el centro de la aplicación), para ello debe ser un usuario registrado por la aplicación.

Al pulsar con doble clic del ratón sobre el área indicada como 2 en la barra de estado, el software accede a la herramienta de definición de grupos (tópicos) o “Definición de Dispositivos” esclavos a ser monitoreados.

Al pulsar con doble clic sobre el área indicada como 3 de la figura A.18 se accede a la herramienta de “Configuración del Puerto Serie”.

Al pulsar con doble clic sobre el área indicada como 4 de la figura A.18 se accede a la herramienta de “Configuración del Servidor” de datos.

Al pulsar con doble clic sobre el área indicada como 5 de la figura A.18 el software detiene el algoritmo de consulta de dispositivos y busca los nodos de la red que se encuentran conectados y no están definidos por la herramienta “Definición de Dispositivos”. Al culminar la búsqueda el software indica los resultados obtenidos en la lista de “Estado del servidor”.



**Figura A.18. Descripción de la barra de estado.**

El software desarrollado permite visualizar en la lista central del formulario principal el estado del servidor de datos OPC, mire la figura A.13, al pulsar sobre el título “Estado del servidor” la aplicación modifica el contenido de la lista y presenta el estado de los nodos de red (figura A.19), dicho título cambia por “Estado de la red”. Para cambiar el contenido de la lista se debe dar clic sobre el título “Estado de la red”.

Al lado derecho del formulario principal se puede observar la lista de nodos definidos por la herramienta “Definición de Dispositivos” (figura A.13 o figura A.19).

Si se da doble clic sobre la lista del “Estado de la red” se inicia la herramienta de configuración “Definición de Dispositivos”. Si se da doble clic sobre cualquier elemento de la lista de dispositivos del lado izquierdo se ejecuta la herramienta “Análisis de Tramas.”

La lista “Estado del servidor” (figura A.13) indica: el registro de los usuarios, si el servidor de datos tiene errores en la actualización de datos, si la lectura de elementos del servidor de datos no es correcta, si algún cliente OPC define algún elemento que no esté registrado en la memoria del servidor, cuando no existen nodos esclavos definidos y el estado de la comunicación serie cuando ésta cambia.

La lista “Estado de la red” (figura A.19) indica: los nombres de los nodos (tópicos) definidos que son consultados por el protocolo de comunicación, la dirección de cada nodo de red, indica el tiempo mínimo de consulta de cada nodo, el número de tramas exitosas realizadas a dicho nodo, el número de tramas que tienen errores o sin respuesta y el estado actual de cada nodo (ACK significa que el nodo esta respondiendo correctamente y NACK que el nodo no está respondiendo).

MicroBUS - Red digital para Microcontroladores

Configuración Usuario Ayuda

Estado de la red: Dispositivos: 5

Tópico	Dirección	Timeout	Bien	Errores	Estado
23:02:48 Nodo2	2	100	10	0	ACK
23:02:48 Nodo4	4	100	10	0	ACK
23:02:48 Nodo5	5	100	10	0	ACK
23:02:48 Nodo7	7	100	10	0	ACK
23:02:48 Nodo8	8	100	10	0	ACK

Nodo2 2  
Nodo4 4  
Nodo5 5  
Nodo7 7  
Nodo8 8

Center Control DYCOMMSA      Conectado      115200.N,8,1      OPC 20 mS      23:02

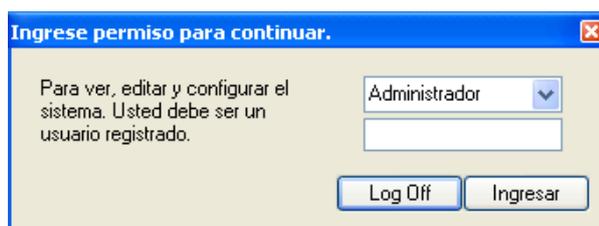
**Figura A.19. Descripción del formulario principal de la aplicación.**

## H.5. CONFIGURACIÓN DE USUARIOS.

Para configurar el sistema, borrar la lista de estado del servidor e ingresar en la herramienta de “Análisis de Tramas”, el software MicroBUS previamente debe reconocer al usuario y para ello inicia el formulario de validación de usuarios.

La figura A.20, indica el formulario de permisos de usuarios. El usuario debe escoger su nombre de usuario e ingresar su contraseña.

Cuando se inicia por primera vez el software el usuario por defecto es el “*Administrador*” y su contraseña es “*1*”. El administrador debe ingresar en la herramienta de configuración de usuarios y cambiar la contraseña de dicho usuario administrador (figura A.21).



Ingrese permiso para continuar.

Para ver, editar y configurar el sistema. Usted debe ser un usuario registrado.

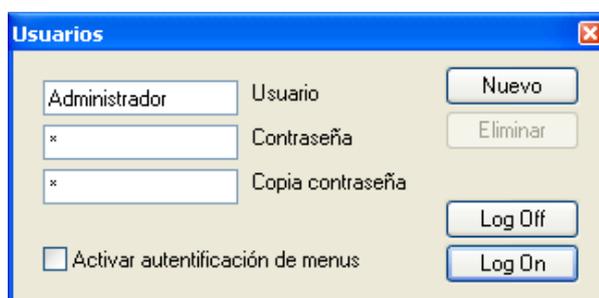
Administrador

Log Off Ingresar

**Figura A.20. Descripción del formulario principal de la aplicación.**

En la herramienta de configuración de usuarios (figura A.21) se puede administrar los usuarios que configuren el sistema. La herramienta permite crear nuevos usuarios, eliminar usuarios y modificar la clave de usuario.

Si un usuario activa la autenticación de menús éste tendrá que digitar su contraseña para ingresar a cualquier menú. Si el usuario no activa la autenticación de menús éste no necesita digitar la contraseña para ingresar en cualquier herramienta de configuración.



Usuarios

Administrador Usuario Nuevo

\* Contraseña Eliminar

\* Copia contraseña Log Off

Activar autenticación de menus Log On

**Figura A.21. Administración de usuarios.**

## H.6. CONFIGURACIÓN DEL PUERTO SERIE.

Cuando se accede a la herramienta de “Configuración del Puerto Serie” (figura A.22) el software detiene la consulta de esclavos y detecta los puertos de comunicación serie que tiene el computador. Esta herramienta configura el número de puerto de comunicación serie y la velocidad de comunicación de dicho puerto. Para cambiar la configuración del puerto serie se debe pulsar la tecla “Aplicar”

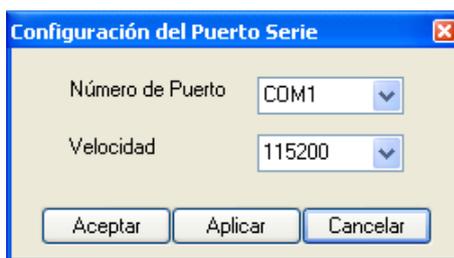


Figura A.22. Configuración del puerto de comunicaciones.

## H.7. DEFINICIÓN DE DISPOSITIVOS DE LA RED.

Cuando se accede a la herramienta “Definición de Dispositivos” se detiene la consulta de esclavos. Este formulario permite configurar los nodos esclavos que podrán ser consultados por el software maestro.

La figura A.23 indica la herramienta de configuración “Definición de Dispositivos”, ésta herramienta permite “Añadir” nuevas estaciones, “Modificar” las configuraciones de las estaciones, “Eliminar” estaciones definidas y también permite buscar estaciones de la red, figura A.24. Para buscar estaciones de la red no debe estar definida ninguna estación en el formulario “Definición de Dispositivos”.

Para modificar o eliminar una estación se debe marcar primero la estación (lista del formulario) que se desee modificar o eliminar. Cuando se añade o modifica un dispositivo el software inicia el formulario de configuración de dispositivos (figura A.25). El software permite definir máximo 31 dispositivos de red.



Figura A.23. Formulario de administración de dispositivos de la red.

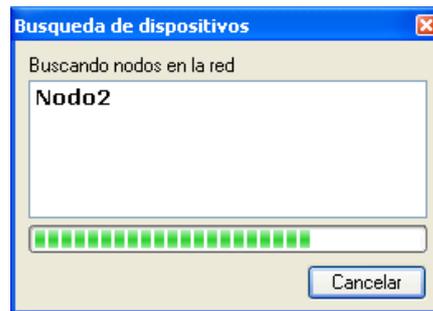


Figura A.24. Búsqueda de dispositivos.

## H.8. CONFIGURACIÓN DE DISPOSITIVOS.

La figura A.25 indica las configuraciones de un nodo esclavo. A este formulario se accede cuando se desea añadir o modificar un nodo. Para añadir o modificar un nodo el usuario debe estar en el formulario “Definición de Dispositivos“.

Para añadir un nodo o modificar un nodo de la red se pone un nombre único (máximo 15 caracteres), este nombre se usa como identificador del dispositivo. La dirección debe ser única puede ir de 1 a 169 y de 171 a 255. Se debe ingresar el tiempo mínimo en que el servidor de datos realizará cada consulta a dicho dispositivo (TimeOut) el tiempo es ingresado en milisegundos y permite un máximo tiempo de 1000 segundos. Se puede verificar si dicho dispositivo se encuentra conectado en el puerto de comunicación serie. Para verificar que un dispositivo está conectado debe pulsar sobre el mensaje “Analizar”, si el nodo responde el mensaje analizar cambiará por “OK”, si el nodo no responde el mensaje “Analizar” cambiará por “No responde”.

Configuración del Dispositivo

Nombre:  
Nodo2

Dirección:  
2

TimeOut: (mseg)  
100

Crear

Cancelar

**Analizar**

Figura A.25. Configuración de nodos esclavos.

## H.9. CONFIGURACIÓN DEL SERVIDOR.

La figura A.26 indica el formulario de configuración del servidor. Cuando se accede a la “Configuración del Servidor” se detiene el proceso de consulta a los dispositivos de red.

En el formulario “Configuración del servidor” se puede manipular el “Tiempo Polling”, ésta configuración hace referencia al tiempo mínimo en que la aplicación MicroBUS consultará a cada dispositivo de la red. El tiempo de consulta “Tiempo Polling” está definido en milisegundos, mínimo acepta 10 milisegundos y máximo 1000 milisegundos.

Configuración del Servidor

**Servidor**

DDE

OPC

**Tiempo Polling**

20 mSeg

Figura A.26. Configuración del servidor de datos.

## H.10. ANÁLISIS DE TRAMAS.

El formulario “Análisis de Tramas” (figura A.27) es utilizado para conocer el estado de los registros y configurar los registros de cualquier nodo esclavo conectado a la red. En este formulario se puede realizar cualquier consulta o escritura permitida por las características de la trama. Para transmitir una trama se debe ingresar la dirección del nodo esclavo, configurar el campo de control del dispositivo, especificar la localidad de inicio de transacción, indicar la longitud de datos y dependiendo del tipo de trama se debe incluir los datos (trama de escritura de datos).

Para transmitir una trama se deben llenar todos los campos requeridos para dicha transmisión (los datos deben ser ingresados en decimal) y posteriormente se debe pulsar sobre el mensaje “TX” o sobre la tecla “Polling”, el software espera un instante de tiempo y analiza los datos recibidos.

El software calcula el código de error para la trama de transmisión cuando se transmite la trama. El código de error se visualiza en código decimal.

Cuando se realiza una transmisión en que responde el nodo receptor, la trama es verificada y si no tiene errores la información de la trama receptada se escribe en los campos que corresponden a la recepción de datos.

Cuando un nodo esclavo responde a una trama se puede visualizar (codificados en ASCII) en el control RichTextBox. El control RichTextBox está localizado en la ficha “Buffer” del control TabStrip ubicado en el lado derecho del formulario “Análisis de Tramas”.

El dar doble clic sobre el control RichTextBox permite desglosar la información contenida en dicho control, la información que el control indica es: el dato receptado en código ASCII, el valor en decimal, el valor en hexadecimal y el valor en código binario (obsérvese la figura A.27).

**Análisis de Tramas**

	TX	RX
DIRECCION	2	2
CONTROL	195	
LOCALIDAD	50	
LONGITUD	60	60
DATOS		= MicroBU
CRC	11	BE

TimeOut: 500         Escribir     Leer

Errores recepción: 0

Datos recibidos: MicroBUS Protocol. Red digital de Microcontroladores 1234560

Localidad Inicial: 32  
Localidad Final: 6E

Longitud del buffer: 1186

Analisis de la trama recibida:

170	AA	10101010
170	AA	10101010
170	AA	10101010
2	2	00000010
60	3C	00111100
77	4D	01001101

**Figura A.27. Formulario análisis de tramas.**

Para variar el campo de control utilizado en la transmisión se debe dar clic sobre el área de la ficha “Configuración” del control TabStrip ubicado en el lado derecho del formulario “Análisis de Tramas”. También se debe indicar si es una trama de escritura o lectura, dicha selección está dada por las opciones “Escribir” y “Leer” ubicadas en la parte inferior del formulario “Análisis de Tramas”. La figura A.28 indica las configuraciones de la trama: tipo de memoria (EEPROM o RAM) a la que se refiere la trama, número de banco de memoria RAM a la que se refiere la trama de transmisión y tipo de datos que se pueden transmitir.

Cuando se ha seleccionado el tipo de datos byte y la opción de tramas de escritura, el software permite que se modifique el campo de datos con números enteros tipo decimal de 0 a 255. El campo de longitud se mantiene en uno y no puede ser modificado. Cuando se ha seleccionado el tipo de datos entero y la opción de tramas de escritura, el software permite que se modifique el campo de datos con números enteros tipo decimal de 0 a 65535. El campo de longitud se mantiene en dos y no puede ser modificado, el software divide el campo de datos en dos bytes. Cuando se ha seleccionado el tipo de datos “char” y la opción de tramas de escritura, el software permite que se modifique el campo de datos con un dato tipo caracter ASCII. El campo de longitud se mantiene en uno y no puede ser modificado. Cuando se ha seleccionado el tipo de datos “string” y la opción de tramas de escritura, el software permite que se modifique el campo de datos

con una cadena de caracteres. El campo de longitud es calculado dependiendo de la longitud de la cadena de caracteres. El software no toma en cuenta el carácter nulo al final de la cadena de caracteres tanto para la transmisión y recepción de tramas.

Modificar el número de banco sólo es posible cuando la opción RAM está seleccionada.



**Figura A.28. Configuración del campo de control.**

# ANEXO I

## SOFTWARE DESARROLLADO.

```
***** MicroBUS - Red digital para Microcontroladores
***** Software de desarrollo: MPLAB 7.20
***** Archivo fuente: MicroBUSv628A.asm
***** Versión: 2.0.0.580
```

```
***** Mensajes de compilación & advertencia *****
```

```
ERRORLEVEL -205
ERRORLEVEL -215
ERRORLEVEL -301
ERRORLEVEL -207
ERRORLEVEL -302
```

```
***** Declaración del procesador & configuración *****
```

```
PROCESSOR 16F628A
#include "p16F628A.inc"
#include "C:\modulos\m_bank.asm"
__CONFIG _CP_OFF & _WDT_ON & _PWRTE_ON & _HS_OSC &
         _BODEN_ON & _MCLRE_ON & _LVP_OFF
```

```
***** Memoria de datos. EEPROM - configuraciones del protocolo - ****
```

```
ORG 0x2100
DE d'5',d'5',d'255'

ORG 0x2132
DE "MicroBUS. Red digital de Microcontroladores. DYCOMMSA SAAM",0
```

```
***** Estructura de la memoria de programa. FLASH *****
```

```
ORG 0x00
GOTO INICIO
ORG 0x04
GOTO ISR
```

```
***** Declaración de registro del FSR *****
```

```
INDF EQU 00H
PLC EQU 02H
FSR EQU 04H
```

```
***** Declaración de constantes simbólicas *****
```

```
CONSTANT BASE = 0x20
CONSTANT MAPEADO = 0x70
```

**\*\*\*\*\* Control de circuitos RS-485 \*\*\*\*\***

```
#DEFINE      RTS_485      PORTB,3
```

**\*\*\*\*\* Declaración del puntero de tablas \*\*\*\*\***

```
LTABLA      EQU          BASE
```

**\*\*\*\*\* Definición de banderas de la trama de recepción y transmisión \*\*\*\*\***

```
BANrx      EQU          BASE+d'6'  
#DEFINE     INIt        BANrx,0x00  
#DEFINE     DIRt        BANrx,0x01  
#DEFINE     CONt        BANrx,0x02  
#DEFINE     LOCt        BANrx,0x03  
#DEFINE     LONt        BANrx,0x04  
#DEFINE     DATt        BANrx,0x05  
#DEFINE     BROADt      BANrx,0x06  
#DEFINE     BIENt       BANrx,0x07
```

**\*\*\*\*\* Declaración de registros para la trama \*\*\*\*\***

```
CBLOCK      BASE+d'7'  
            INI_R  
            DIR_R  
            CON_R  
            LOC_R  
            LON_R  
            CRC_R  
            CRC  
            RXi  
            RXI  
            RXt  
            TXt  
ENDC
```

**\*\*\*\*\* Definición de bits para el control de trama \*\*\*\*\***

```
#DEFINE     RP0f        CON_R,0x00  
#DEFINE     RP1f        CON_R,0x01  
#DEFINE     RP2f        CON_R,0x02  
#DEFINE     RP3f        CON_R,0x03  
#DEFINE     TESTf       CON_R,0x04  
#DEFINE     RWf         CON_R,0x06  
#DEFINE     FSREE       CON_R,0x07
```

**\*\*\*\*\* Declaración de memoria temporal de recepción de trama \*\*\*\*\***

```
CBLOCK      BASE+d'18'  
            DATOS  
ENDC
```

**\*\*\*\*\* Declaración de constantes simbólicas de variables de la trama \*\*\*\*\***

```

CBLOCK      BASE+d'18'
            D0X, D1X, D2X, D3X, D4X, D5X, D6X, D7X, D8X, D9X
            I00, I01, I02, I03, I04, I05, I06, I07, I08, I09
            I10, I11, I12, I13, I14, I15, I16, I17, I18, I19
            A00:0, A00H, A00L,      A01:0, A01H, A01L
            A02:0, A02H, A02L,      A03:0, A03H, A03L
            A04:0, A04H, A04L,      A05:0, A05H, A05L
            A06:0, A06H, A06L,      A07:0, A07H, A07L
            A08:0, A08H, A08L,      A09:0, A09H, A09L
            M00
ENDC

```

**\*\*\*\*\* Declaración de constantes simbólicas de variables discretas de D0X \*\*\*\*\***

```

#DEFINE     D00     D0X,0x00
#DEFINE     D01     D0X,0x01
#DEFINE     D02     D0X,0x02
#DEFINE     D03     D0X,0x03
#DEFINE     D04     D0X,0x04
#DEFINE     D05     D0X,0x05
#DEFINE     D06     D0X,0x06
#DEFINE     D07     D0X,0x07

```

**\*\*\*\*\* Declaración de registros para del puntero de programa \*\*\*\*\***

```

W_TEMP      EQU     MAPEADO+d'12'
STATUS_TEMP EQU     MAPEADO+d'13'
PCLATH_TEMP EQU     MAPEADO+d'14'
FSR_TEMP    EQU     MAPEADO+d'15'

```

**\*\*\*\*\* Tablas de trabajo \*\*\*\*\***

```

TABLA
MOVWF      PLC
MARCO      DT     "MicroBUS Protocolo",0
MARCOCL    DT     "          ",0

SPEEDcom
ADDWF      PLC,F
DT         d'255',d'64',d'31',d'21',d'18',d'10',d'4',d'2',d'1',d'0'

```

**\*\*\* Configuración del protocolo \*\*\***

```

PROTOCOLO_ini
BANK1
MOVLW     b'00100100'
MOVWF     TXSTA
MOVLW     d'1'
MOVWF     EEADR
BSF       EECON1,RD
MOVFW    EEDATA
MOVWF     RXt

```

	SUBLW	d'9'
	BNC	SEEDe
	MOVFW	RXt
	CALL	SPEEDcom
SEEDe	MOVWF	SPBRG
	BSF	PIE1,RCIE
	BANK0	
	BCF	RTS_485
	MOVLW	b'10010000'
	MOVWF	RCSTA
	BSF	INTCON,PEIE
	MOVLW	d'0'
	BANK1	
	MOVWF	EEADR
	BSF	EECON1,RD
	MOVFW	EEDATA
	BANK0	
	MOVWF	DIR_R
	MOVLW	d'170'
	MOVWF	INI_R
	MOVLW	d'3'
	MOVWF	RXi
	MOVLW	d'0'
	MOVWF	BANrx
	MOVWF	LON_R
	MOVWF	RXI
	MOVWF	CRC
	MOVLW	d'2'
	BANK1	
	MOVWF	EEADR
	BANK0	

```

;*****
;*****
;***** Escriba el código de inicialización de variables
;***** usadas por el programa principal (Esto depende de la lógica del programa)
;***** (Código de inicialización del algoritmo de control principal)
;*****

```

RETURN

```

;***** Direccionamiento indirecto *****

```

IBANK	BTFSC	RP0f
	GOTO	RP01
	BCF	FSR,7
	BTFSC	RP1f
	GOTO	RP11
	GOTO	RP10
RP01	BSF	FSR,7
	BTFSC	RP1f
	GOTO	RP11
RP10	BCF	STATUS,IRP
	GOTO	SIBAN
RP11	BSF	STATUS,IRP
SIBAN	NOP	
	RETURN	

**;\*\*\*\*\* Transmisión de datos \*\*\*\*\***

TXw

```
BANK1
BTFSS      TXSTA,TRMT
GOTO      TXw
BANK0
MOVWF     TXt
BCF       PIR1,TXIF
MOVWF     TXREG
BANK0
XORWF     CRC,F
CLRWF
BTFSC     CRC,0
XORLW     0x5E
BTFSC     CRC,1
XORLW     0xBC
BTFSC     CRC,2
XORLW     0x61
BTFSC     CRC,3
XORLW     0xC2
BTFSC     CRC,4
XORLW     0x9D
BTFSC     CRC,5
XORLW     0x23
BTFSC     CRC,6
XORLW     0x46
BTFSC     CRC,7
XORLW     0x8C
MOVWF     CRC
MOVFW     TXt
RETURN
```

**;\*\*\*\*\* Rutina de servicio de interrupción "ISR" \*\*\*\*\***

ISR

```
BCF       INTCON,GIE
BTFSC     INTCON,GIE
GOTO      ISR
MOVWF     W_TEMP
SWAPF     STATUS,W
MOVWF     STATUS_TEMP
CLRF      STATUS
MOVFW     PCLATH
MOVWF     PCLATH_TEMP
CLRF      PCLATH
BCF       STATUS,IRP
MOVFW     FSR
MOVWF     FSR_TEMP
```

**;\*\*\*\* Determina el origen de la interrupción \*\*\*\***

```
BTFSC     PIR1,RCIF
GOTO      PROTOCOLOrx
GOTO      ISREnd
```

**;\*\*\*\*\* Fin de la rutina de servicio de interrupción "ISR" \*\*\*\*\***

```
ISREnd
    MOVFW    FSR_TEMP
    MOVWF    FSR
    MOVFW    PCLATH_TEMP
    MOVWF    PCLATH
    SWAPF    STATUS_TEMP,W
    MOVWF    STATUS
    SWAPF    W_TEMP,F
    SWAPF    W_TEMP,W
    RETFIE
```

**;\*\*\*\*\* Protocolo de comunicación - recepción \*\*\*\*\***

```
PROTOCOLOrx
    BANK0
    BCF      PIR1,RCIF
    BTFSC    RCSTA,FERR
    GOTO     FINrxtx
    BTFSC    RCSTA,OERR
    GOTO     FINrxtx
    MOVF     RCREG,W
    MOVWF    RXt
    XORLW    d'170'
    BTFSC    STATUS,Z
    GOTO     CABEZArx
    MOVLW    d'3'
    MOVWF    RXi
    MOVFW    RXt

CABEZAi
    BTFSS    INIt
    GOTO     ISREnd
    BTFSS    DIRt
    GOTO     DIRECCIONrx
    BTFSS    CONt
    GOTO     CONTROLrx
    BTFSS    LOCt
    GOTO     LOCALIDADrx
    BTFSS    LONt
    GOTO     LONGITUDrx
    BTFSS    DATt
    GOTO     DATOSrx
    MOVWF    CRC_R
    MOVLW    d'170'
    XORWF    CRC,W
    BTFSC    STATUS,Z
    INCF     CRC,F
    MOVFW    CRC_R
    XORWF    CRC,W
    BTFSC    STATUS,Z
    GOTO     ANALISISrx
    GOTO     CRCe

CRC8rx
    XORWF    CRC,F
    CLRW
    BTFSC    CRC,0
    XORLW    0x5E
```

	BTFSC	CRC,1
	XORLW	0xBC
	BTFSC	CRC,2
	XORLW	0x61
	BTFSC	CRC,3
	XORLW	0xC2
	BTFSC	CRC,4
	XORLW	0x9D
	BTFSC	CRC,5
	XORLW	0x23
	BTFSC	CRC,6
	XORLW	0x46
	BTFSC	CRC,7
	XORLW	0x8C
	MOVWF	CRC
VOLVERrx	GOTO	ISREnd

**;\*\*\*\*\* Validación de los campos de la trama recibida \*\*\*\*\***

CABEZArx	DECFSZ	RXi,F
	GOTO	CABEZAi
	MOVLW	d'3'
	MOVWF	RXi
	BTFSC	INIt
	GOTO	FINrxtx
	BSF	INIt
	GOTO	ISREnd
DIRECCIONrx	CLRF	CRC
	BSF	DIRt
	MOVFW	RXt
	BTFSC	STATUS,Z
	GOTO	BROADCASTrx
	XORWF	DIR_R,W
	BTFSS	STATUS,Z
	GOTO	FINrxtx
	MOVFW	RXt
	GOTO	CRC8rx
CONTROLrx	MOVWF	CON_R
	BSF	CONt
	BTFSS	TESTf
	GOTO	CRC8rx
	CLRF	LON_R
	BSF	LOCt
	BSF	LONt
	BSF	DATt
	GOTO	CRC8rx
LOCALIDADrx	MOVWF	LOC_R
	BSF	LOCt
	GOTO	CRC8rx

LONGITUDrx

MOVWF	LON_R
CLRF	RXI
BTFSC	RWf
BSF	DATt
MOVFW	LON_R
BTFSC	STATUS,Z
BSF	DATt
BSF	LONt
MOVFW	LON_R
GOTO	CRC8rx

DATOSrx

BCF	STATUS,IRP
MOVLW	DATOS
ADDWF	RXI,W
MOVWF	FSR
XORLW	d'128'
BTFSC	STATUS,Z
GOTO	FINrxtx
MOVFW	RXt
MOVWF	INDF
INCF	RXI,F
BSF	DATt
MOVFW	RXI
XORWF	LON_R,W
BTFSS	STATUS,Z
BCF	DATt
MOVFW	RXt
GOTO	CRC8rx

BROADCASTrx

BSF	BROADt
GOTO	CRC8rx

**;\*\*\*\*\* Análisis de la información de la transacción del maestro \*\*\*\*\***

ANALISISrx

BANK0	
MOVFW	LOC_R
XORLW	d'50'
BTFSS	STATUS,Z
GOTO	ANALISISrama
MOVFW	LON_R
XORLW	d'60'
BTFSS	STATUS,Z
GOTO	ANALISISrama
MOVFW	CON_R
ANDLW	d'207'
XORLW	d'3'
BCF	RP1f
BTFSS	STATUS,Z
GOTO	ANALISISrama
BSF	RWf

;\*\*\*\*\*

```

;***** Escriba el código que interactúe con el protocolo y el
;***** el programa principal (Esto depende de la lógica de control)
;***** Copie las localidades que deben ser actualizar en el servidor de datos OPC
;***** para ello copie las localidades del banco 0 en el banco 1.
;***** Mire los ejemplos siguientes:
;*****

```

**;Ejemplo 1. Escritura en OPC**

```

BANK0
MOVFW    PORTB
ANDLW    d'241'
BANK1
MOVWF    D0X

```

**;Ejemplo 2. Lectura de OPC**

```

BANK0
MOVFW    D1X
ANDLW    d'31'
MOVWF    PORTA
BANK1
MOVWF    D1X
BANK0
MOVFW    A00
BANK1
MOVWF    A00
BANK0
MOVFW    A01
BANK1
MOVWF    A01

```

**;\*\*\*\*\* Análisis del tipo de trama \*\*\*\*\***

**ANALISIStrama**

```

BANK0
CLRF     RXI
BTFSC   TESTf
GOTO    TESTtx
MOVFW   LON_R
BTFSC   STATUS,Z
GOTO    LON0e
BTFSC   FSREE
GOTO    EEPROMtx
BTFSC   RWf
GOTO    LEER_BYTErx

```

**ESCRIBE\_FSR**

```

BANKISEL  DATOS
MOVLW    DATOS
ADDWF    RXI,W
MOVWF    FSR
MOVFW    INDF
MOVWF    RXt
MOVFW    LOC_R
MOVWF    FSR
MOVFW    RXt

```

CALL	IBANK
MOVWF	INDF
INCF	LOC_R,F
INCF	RXl,F
MOVFW	RXl
XORWF	LON_R,W
BTFSS	STATUS,Z
GOTO	ESCRIBE_FSR
GOTO	PROTOCOLOtx

LEER\_BYTErx

MOVFW	LOC_R
MOVWF	FSR
CALL	IBANK
MOVFW	INDF
MOVWF	RXt
BANKISEL	DATOS
MOVLW	DATOS
ADDWF	RXl,W
MOVWF	FSR
BCF	FSR,7
MOVFW	RXt
MOVWF	INDF
INCF	LOC_R,F
INCF	RXl,F
MOVFW	RXl
XORWF	LON_R,W
BTFSS	STATUS,Z
GOTO	LEER_BYTErx
GOTO	PROTOCOLOtx

EEPROMtx

BCF	STATUS,IRP
CLRF	RXl
BTFSC	RWf
GOTO	LEER_EEBYTErx

ESCRIBIR\_EE

MOVFW	LOC_R
BANK1	
BTFSC	EECON1,WR
GOTO	\$_1
MOVWF	EEADR
BANK0	
MOVLW	DATOS
ADDWF	RXl,W
MOVWF	FSR
BTFSC	FSR,7
GOTO	FINrxtx
MOVFW	INDF
BANK1	
MOVWF	EEDATA
BSF	EECON1,WREN
MOVLW	55H
MOVWF	EECON2
MOVLW	0xAA
MOVWF	EECON2
BSF	EECON1,WR
BCF	EECON1,WREN
BANK0	
INCF	LOC_R,F

	BTFSC	STATUS,Z
	GOTO	FINrxtx
	INCF	RXl,F
	MOVFW	RXl
	XORWF	LON_R,W
	BTFSS	STATUS,Z
	GOTO	ESCRIBIR_EE
	GOTO	PROTOCOLOtx
LEER_EE	BCF	STATUS,IRP
	CLRF	RXl
LEER_EE	MOVFW	LOC_R
	BANK1	
	MOVWF	EEADR
	BSF	EECON1,RD
	MOVFW	EEDATA
	BANK0	
	MOVWF	RXt
	MOVLW	DATOS
	ADDWF	RXl,W
	MOVWF	FSR
	BTFSC	FSR,7
	GOTO	FINrxtx
	MOVFW	RXt
	MOVWF	INDF
	INCF	LOC_R,F
	BTFSC	STATUS,Z
	GOTO	FINrxtx
	INCF	RXl,F
	MOVFW	RXl
	XORWF	LON_R,W
	BTFSS	STATUS,Z
	GOTO	LEER_EE
	GOTO	PROTOCOLOtx

**;\*\*\*\*\* Protocolo de comunicación - tramas especiales de transmisión \*\*\*\*\***

LON0e	BSF	TESTf
	GOTO	PROTOCOLOtx
CRCe	BSF	TESTf
	GOTO	FINrxtx
BROADtx	GOTO	FINrxtx
TESTtx	BSF	TESTf
	GOTO	PROTOCOLOtx

**;\*\*\*\*\* Protocolo de comunicación - transmisión \*\*\*\*\***

```

PROTOCOLOtx
    BANK0
    BSF          RTS_485
    BTFSC       BROADt
    GOTO        FINrxtx
    MOVLW      d'255'
    CALL       TXw
    MOVFW      INI_R
    CALL       TXw
    MOVFW      INI_R
    CALL       TXw
    MOVFW      INI_R
    CALL       TXw
    CLRF       CRC
    MOVFW      DIR_R
    CALL       TXw
    MOVFW      LON_R
    CALL       TXw
    BTFSC      TESTf
    GOTO        PASO2tx
    CLRF       RXI

PASO1tx
    BCF        STATUS,IRP
    MOVFW      RXI
    XORWF      LON_R,W
    BTFSC      STATUS,Z
    GOTO        PASO2tx
    MOVLW      DATOS
    ADDWF      RXI,W
    MOVWF      FSR
    BTFSC      FSR,7
    GOTO        FINrxtx
    MOVFW      INDF
    CALL       TXw
    INCF      RXI,F
    GOTO        PASO1tx

PASO2tx
    MOVFW      CRC
    XORLW      d'170'
    BTFSC      STATUS,Z
    INCF      CRC,F
    MOVFW      CRC
    CALL       TXw
    CALL       TXw
    CALL       PROTOCOLO_ini
    BCF        RTS_485
    BANK0
    GOTO        ISRend

FINrxtx

```

**;\*\*\*\*\* Inicialización polling \*\*\*\*\***

```

POLLING_ini
    BANK1
    MOVLW      ''
    MOVWF      d'100'
    MOVLW      'M'
    MOVWF      d'101'

```

```

        MOVLW      'i'
        MOVWF     d'102'
        MOVLW      'c'
        MOVWF     d'103'
        MOVLW      'r'
        MOVWF     d'104'
        MOVLW      'o'
        MOVWF     d'105'
        MOVLW      'B'
        MOVWF     d'106'
        MOVLW      'U'
        MOVWF     d'107'
        MOVLW      'S'
        MOVWF     d'108'
        MOVLW      ''
        MOVWF     d'109'
        MOVLW      d'0'
        MOVWF     d'58'
        MOVLW      d'60'
        MOVWF     d'59'
        BCF       STATUS,IRP
CONTEO
        MOVFW     d'59'
        MOVWF     FSR
        BSF       FSR,7
        MOVFW     d'58'
        MOVWF     INDF
        INCF     d'58',F
        INCF     d'59',F
        MOVFW     d'59'
        XORLW    d'100'
        BTFSS    STATUS,Z
        GOTO     CONTEO
        BANK0
        RETURN

```

**;\*\*\*\*\* Programa principal \*\*\*\*\***

```

INICIO
        CLRWDT
        BANK0
        CLRF     PORTA
        CLRF     PORTB
        MOVLW    7
        MOVWF    CMCON
        BANK1
        MOVLW    b'00000000'
        MOVWF    TRISA
        MOVLW    b'11110011'
        MOVWF    TRISB
        MOVLW    b'11101111'
        MOVWF    OPTION_REG
        BANK0
        BCF      RTS_485
        CLRF     INTCON
        CALL     PROTOCOLO_ini
        CALL     POLLING_ini
        BSF      INTCON,PEIE

```

```
BUCLE      MOVLW      b'11000000'  
           MOVWF      INTCON  
           CLRWDT  
  
           BANK0  
           MOVFW      PORTB  
           BANK1  
           MOVWF      D0X  
           BANK0  
           BTFSC      PORTB,0  
           BSF        PORTA,0  
           BTFSC      PORTB,4  
           BSF        PORTA,1  
           BTFSC      PORTB,5  
           BSF        PORTA,2  
           BTFSC      PORTB,6  
           BSF        PORTA,3  
           BTFSC      PORTB,7  
           BSF        PORTA,4  
           BANK0  
           MOVFW      PORTA  
           BANK1  
           MOVWF      D1X  
  
           GOTO      BUCLE  
           END
```

# **MicroBUS**

## **Manual de usuario**

**Revisión 1**  
**Versión 2.0.0.507**  
**Octubre 2005**

**Center Control Dycommsa**

## **PROYECTO**

Diseño e implementación de un servidor de datos OPC y un protocolo de comunicación para microcontroladores.

## **INTRODUCCIÓN**

La aplicación MicroBUS permite monitorear una red de microcontroladores con el protocolo MicroBUS y servir los datos por medio de la especificación de OPC Foundation a las aplicaciones clientes de OPC.

## **REQUERIMIENTOS PARA LA INSTALACIÓN**

Para instalar la aplicación MicroBUS se requiere: tener permisos de administrador del sistema, instalar los componentes del servidor OPC "OPC Core Components 2.00 Redistributable" el cual se obtiene de la página Web de OPC Foundation, verificar que el sistema tenga la biblioteca actxprxy.dll, si no existiera se debe obtener el "ActiveX Redistributable Installation Kit" de Microsoft que se encuentra en la página Web de OPC Foundation.

El software diseñado se instala directamente en el sistema operativo (SO) Microsoft Windows XP o Microsoft Windows 2000, el arreglo de la interfase grafica se apta a los estilos del SO.

Si el SO esta basado en Windows 95/98/ME, se deben incluir los componentes de objetos genéricos de distribución (DCOM), además debe existir la biblioteca "MSVBVM60.dll" o "VBRun60.exe".

## **INSTALACIÓN DEL SOFTWARE**

El CD es auto ejecutable y corre directamente el documento basado en HTTP. Caso contrario dicha aplicación puede ser iniciada explorando el CD y ejecutando el archivo MicroBUS.html.

Cuando inicia el CD, automáticamente se puede verificar el contenido de éste en forma visual, como lo indica la figura 1.

Para instalar el software MicroBUS, de un clic izquierdo sobre el software que desee instalar esto depende del sistema operativo en que se desee trabajar. En el caso de instalar el software en el sistema operativo Microsoft Windows XP observe la figura 2.

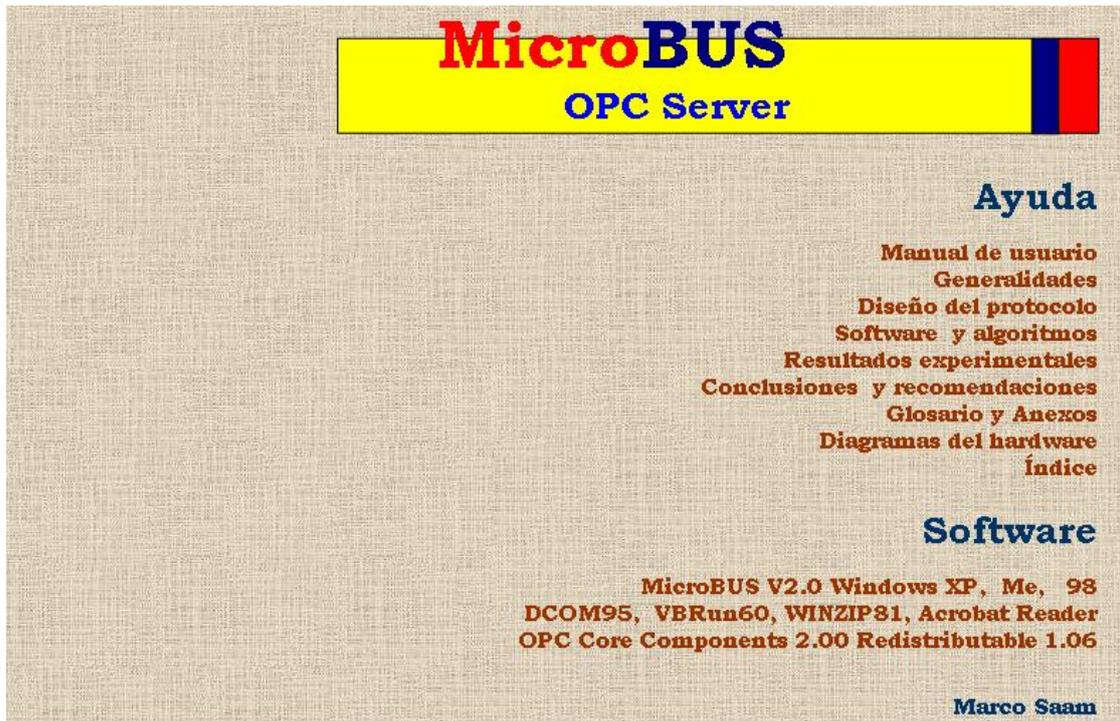


Figura 1. Ejecución automática del CD.

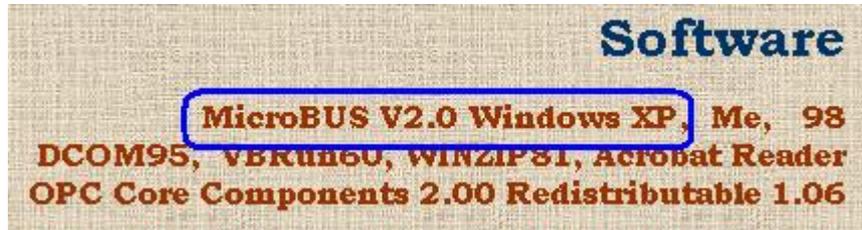


Figura 2. Selección del software a instalar.

Después de dar clic sobre el software que se desee instalar, la página MicroBUS accede a la descarga del archivo. Esta interfase permite instalar la aplicación MicroBUS, refiérase a la figura 3. Si usted desea copiar el archivo, explore el CD y copie toda la carpeta que contenga la versión requerida para su sistema operativo.



Figura 3. Ejecución de la instalación del software MicroBUS.

Prosiga la instalación del software hasta su finalización, si existen componentes previamente instalados en el sistema operativo, el software le preguntará si desea reinstalarlos, se recomienda no reinstalar ningún componente. En caso de que existan componentes ejecutándose en ese instante de tiempo, omita la copia de dichos componente o el registro de dicho componente, esto se puede visualizar en la figura 4. Si usted anula la instalación de algún componente que haya presentado error la aplicación de instalación procede a desinstalar los componentes instalados.

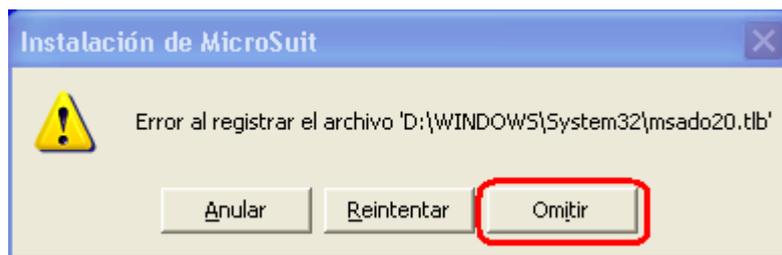


Figura 4. Omisión en errores en la instalación del software.

Al instalarse el software, la aplicación crea el directorio:

C:\Archivos de programa\MicroSUIT.

En dicho directorio se almacenan los archivos:

MicroBUS.exe, aplicación ejecutable.

MicroBUS.EXE.manifest, archivo de manejo de estilos visuales para XP

MicroBUS.mdb, almacena las configuraciones de los nodos.

Agradecimientos.txt, agradecimientos técnicos.

ST6UNST, archivo de registro que permite la desinstalación de la aplicación.

MicroBUSv628A.asm, código fuente del protocolo MicroBUS, desarrollado en MPLAB

MicroBUSv877A.asm, código fuente del protocolo MicroBUS, desarrollado en MPLAB

## INICIO DEL PROGRAMA

Después de una instalación exitosa del software, la figura 5 indica la manera de ejecutar el software.

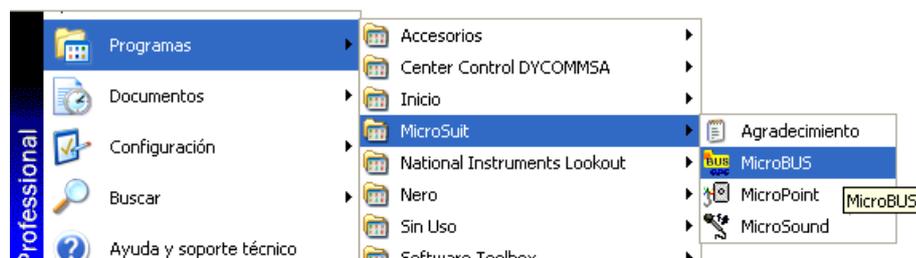


Figura 5. Inicio de la aplicación MicroBUS.

Al iniciar la aplicación esta se visualiza como se indica en la figura 6. La aplicación debe ser configurada para que pueda operar según las conveniencias requeridas. Su interfase gráfica fue desarrollada usando como patrón las interfaces graficas de varios software de monitoreo industrial.

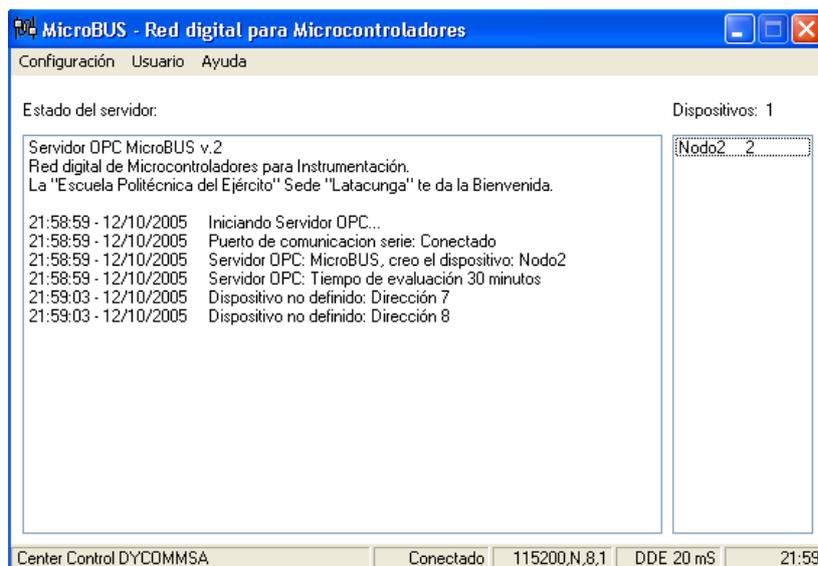


Figura 6. Inicio del software MicroBUS.

## DESCRIPCIÓN DEL FORMULARIO PRINCIPAL

Antes de iniciar la configuración del software, se describe la manera de acceder a las herramientas que el software usa para este fin.

Para acceder a las configuraciones del software puede hacerlo desde varios lugares, esto es con el fin de permitir una rápida configuración.

Se puede acceder desde: la barra de menús, la barra de estado y por los controles de las listas ubicadas en el centro del formulario principal de la aplicación MicroBUS.

Los menús del software son: configuración, usuario y ayuda. El menú de configuración se indica en la figura 7.



Figura 7. Herramientas del menú de configuración.

La figura 7, indica las herramientas que permiten configurar el sistema de monitoreo y el método de intercambio de datos del servidor.

Las herramientas de configuración del puerto de comunicación serie, definición de dispositivos y configuración del servidor, se describen posteriormente. Antes de acceder a dichas herramientas, el usuario debe ser reconocido como usuario de la aplicación.



Figura 8. Herramientas del menú de usuario.

La figura 8 indica las herramientas de administración de usuarios y de análisis de tramas, los cuales se describen posteriormente.

El menú Servidor OPC/Iniciar registro OPC reconfigura el registro de Windows para permitir que el servidor de datos OPC pueda ser iniciado automáticamente por cualquier cliente OPC.

El menú Servidor OPC/Eliminar registro OPC elimina las configuraciones del registro de Windows. Si la aplicación no se encuentra registrada en el sistema operativo esta no puede ser localizada por los clientes OPC. Antes de desinstalar la aplicación definitivamente se recomienda eliminar las configuraciones del registro de Windows. (Eliminar registro OPC).

La figura 9, muestra el contenido del menú de Ayuda. Este menú sólo permite la visualización del formulario "Acerca de MicroBUS", como lo indica la figura 10.



Figura 9. Descripción del menú de ayuda.



Figura 10. Acerca de la aplicación.

La figura 11 corresponde a la barra de estado de la aplicación. Al pulsar, con doble clic sobre el área indicada como 1 borra la lista de “Estado del servidor” (dicha lista ubicada en el centro de la aplicación), para ello debe ser un usuario registrado por la aplicación.

Al pulsar con doble clic del ratón sobre el área indicada como 2 en la barra de estado, el software accede a la herramienta de definición de grupos (tópicos) o “Definición de Dispositivos” esclavos a ser monitoreados.

Al pulsar con doble clic sobre el área indicada como 3 de la figura 11 se accede a la herramienta de “Configuración del Puerto Serie”.

Al pulsar con doble clic sobre el área indicada como 4 de la figura 11 se accede a la herramienta de “Configuración del Servidor” de datos.

Al pulsar con doble clic sobre el área indicada como 5 de la figura 11 el software detiene el algoritmo de consulta de dispositivos y busca los nodos de la red que se encuentran conectados y no están definidos por la herramienta “Definición de Dispositivos”. Al culminar la búsqueda el software indica los resultados obtenidos en la lista de “Estado del servidor”.

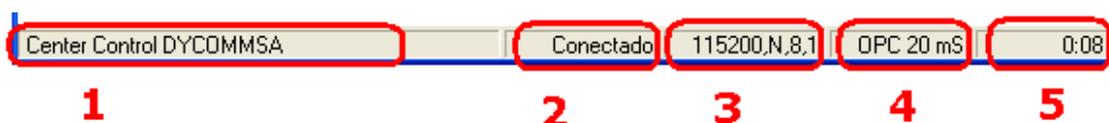


Figura 11. Descripción de la barra de estado.

El software desarrollado permite visualizar en la lista central del formulario principal el estado del servidor de datos OPC, mire la figura 6, al pulsar sobre el título “Estado del servidor” la aplicación modifica el contenido de la lista y presenta el estado de los nodos de red (figura12), dicho titulo cambia por “Estado de la red”. Para cambiar el contenido de la lista se debe dar clic sobre el titulo “Estado de la red”.

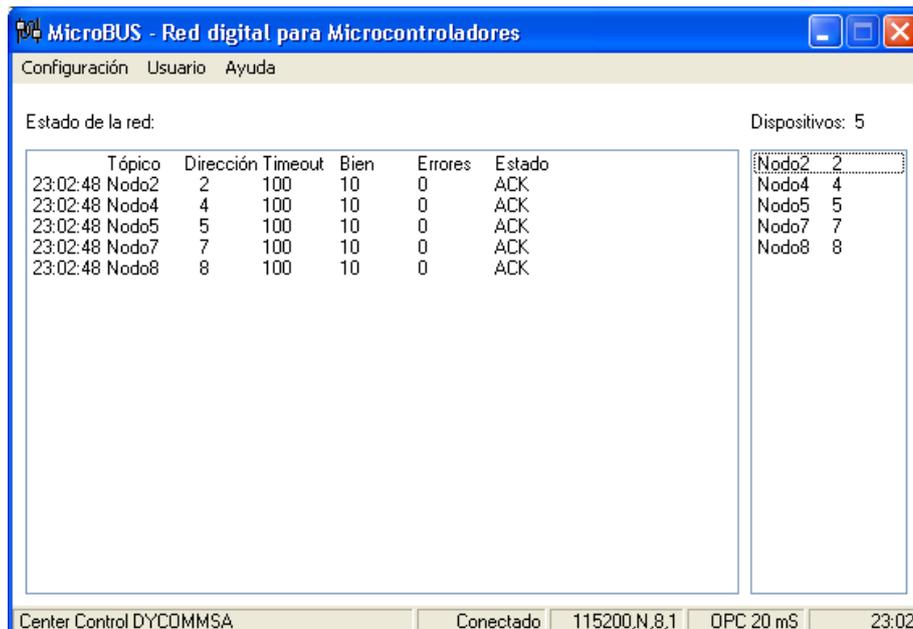
Al lado derecho del formulario principal se puede observar la lista de nodos definidos por la herramienta “Definición de Dispositivos” (figura 6 o figura 12).

Si se da doble clic sobre la lista del “Estado de la red” se inicia la herramienta de configuración “Definición de Dispositivos”.

Si se da doble clic sobre cualquier elemento de la lista de dispositivos del lado izquierdo se ejecuta la herramienta “Análisis de Tramas.”

La lista “Estado del servidor” (figura 6) indica: el registro de los usuarios, si el servidor de datos tiene errores en la actualización de datos, si la lectura de elementos del servidor de datos no es correcta, si algún cliente OPC define algún elemento que no esté registrado en la memoria del servidor, cuando no existen nodos esclavos definidos y el estado de la comunicación serie cuando ésta cambia.

La lista “Estado de la red” (figura 12) indica: los nodos (tópicos) definidos que son consultados por el protocolo de comunicación, la dirección de cada nodo de red, indica el tiempo mínimo de consulta de cada nodo, el número de tramas exitosas realizadas a dicho nodo, el número de tramas que tienen errores o sin respuesta y el estado actual de cada nodo (ACK significa que el nodo esta respondiendo correctamente y NACK que el nodo no está respondiendo).



The screenshot shows a software window titled "MicroBUS - Red digital para Microcontroladores". It has a menu bar with "Configuración", "Usuario", and "Ayuda". The main area is divided into two sections. The left section, titled "Estado de la red:", contains a table with the following data:

	Tópico	Dirección	Timeout	Bien	Errores	Estado
23:02:48	Nodo2	2	100	10	0	ACK
23:02:48	Nodo4	4	100	10	0	ACK
23:02:48	Nodo5	5	100	10	0	ACK
23:02:48	Nodo7	7	100	10	0	ACK
23:02:48	Nodo8	8	100	10	0	ACK

The right section, titled "Dispositivos: 5", contains a list of nodes:

- Nodo2 2
- Nodo4 4
- Nodo5 5
- Nodo7 7
- Nodo8 8

At the bottom of the window, there is a status bar with the following information: "Center Control DYCOMMSA", "Conectado", "115200,N,8,1", "OPC 20 mS", and "23:02".

Figura 12. Descripción del formulario principal de la aplicación.

## CONFIGURACIÓN DE USUARIOS

Para configurar el sistema borrar la lista de estado del servidor e ingresar en la herramienta de “Análisis de Tramas”, el software MicroBUS previamente debe reconocer al usuario y para ello inicia el formulario de validación de usuarios.

La figura 13, indica el formulario de permisos de usuarios. El usuario debe escoger su nombre de usuario e ingresar su contraseña.

Cuando se inicia por primera vez el software el usuario por defecto es el **“Administrador” y su contraseña es “1”**. El administrador debe ingresar en la herramienta de configuración de usuarios y cambiar la contraseña de dicho usuario administrador (figura14).

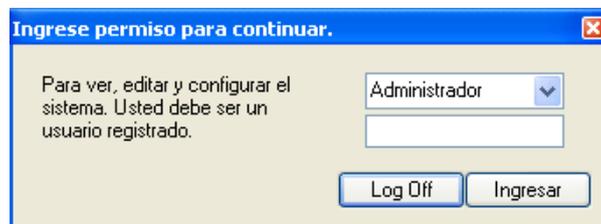


Figura 13. Descripción del formulario principal de la aplicación.

En la herramienta de configuración de usuarios (figura14) se puede administrar los usuarios que configuren el sistema. La herramienta permite crear nuevos usuarios, eliminar usuarios y modificar la clave de usuario.

Si un usuario activa la autenticación de menús éste tendrá que digitar su contraseña para ingresar a cualquier menú. Si el usuario no activa la autenticación de menús éste no necesita digitar la contraseña para ingresar en cualquier herramienta de configuración.

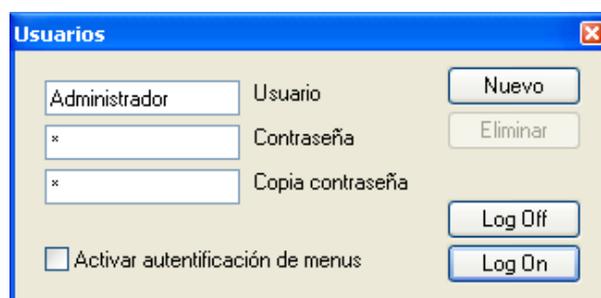


Figura 14. Administración de usuarios.

## CONFIGURACIÓN DEL PUERTO SERIE

Cuando se accede a la herramienta de “Configuración del Puerto Serie” (figura 15) el software detiene la consulta de esclavos y detecta los puertos de comunicación serie que tiene el computador. Esta herramienta configura el número de puerto de comunicación serie y la velocidad de comunicación de dicho puerto. Para cambiar la configuración del puerto serie se debe pulsar la tecla “Aplicar”

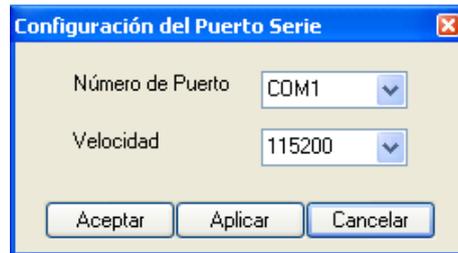


Figura 15. Configuración del puerto de comunicaciones.

## DEFINICIÓN DE DISPOSITIVOS DE LA RED

Cuando se accede a la herramienta “Definición de Dispositivos” se detiene la consulta de esclavos. Este formulario permite configurar los nodos esclavos que podrán ser consultados por el software maestro.

La figura 16 indica la herramienta de configuración “Definición de Dispositivos”, ésta herramienta permite “Añadir” nuevas estaciones, “Modificar” las configuraciones de las estaciones, “Eliminar” estaciones definidas y también permite buscar estaciones de la red, figura 17. Para buscar estaciones de la red no debe estar definida ninguna estación en el formulario “Definición de Dispositivos”.

Para modificar o eliminar una estación se debe marcar primero la estación (lista del formulario) que se desee modificar o eliminar.

Cuando se añade o modifica un dispositivo el software inicia el formulario de configuración de dispositivos (figura 18). El software permite definir máximo 31 dispositivos de red.

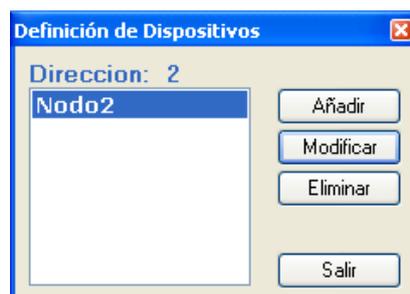


Figura 16. Formulario de administración de dispositivos de la red.

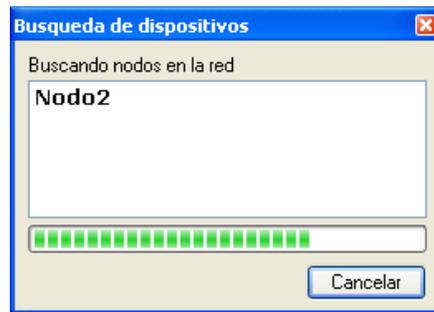


Figura 17. Búsqueda de dispositivos.

## CONFIGURACIÓN DE DISPOSITIVOS

La figura 18 indica las configuraciones de un nodo esclavo. A este formulario se accede cuando se desea añadir o modificar un nodo. Para añadir o modificar un nodo el usuario debe estar en el formulario "Definición de Dispositivos".

Para añadir un nodo o modificar un nodo de la red se pone un nombre único (máximo de 15 caracteres), este nombre se usa como identificador del dispositivo. La dirección debe ser única puede ir de 1 a 169 y de 171 a 255. Se debe ingresar el tiempo mínimo en que el servidor de datos realizará cada consulta a dicho dispositivo (TimeOut), el tiempo es ingresado en milisegundos y permite un máximo tiempo de 1000 segundos. Se puede verificar si dicho dispositivo se encuentra conectado en el puerto de comunicación serie. Para verificar que un dispositivo está conectado debe pulsar sobre el mensaje "Analizar", si el nodo responde el mensaje analizar cambiará por "OK", si el nodo no responde el mensaje "Analizar" cambiará por "No responde"

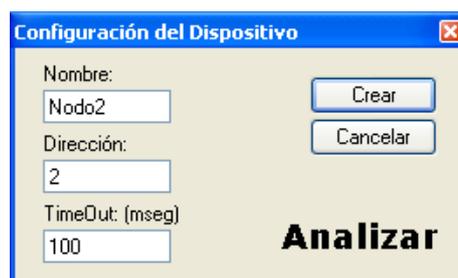


Figura 18. Configuración de nodos esclavos.

## CONFIGURACIÓN DEL SERVIDOR

La figura 19 indica el formulario de configuración del servidor. Cuando se accede a la "Configuración del Servidor" se detiene el proceso de consulta a los dispositivos de red.

En el formulario “Configuración del servidor” se puede manipular el “Tiempo Polling”, ésta configuración hace referencia al tiempo mínimo en que la aplicación MicroBUS consultará a cada dispositivo de la red. El tiempo de consulta “Tiempo Polling” está definido en milisegundos, mínimo acepta 20 milisegundos y máximo 1000 milisegundos.



Figura 19. Configuración del servidor de datos.

## ANÁLISIS DE TRAMAS

El formulario “Análisis de Tramas” (figura 20) es utilizado para conocer el estado de los registros y configurar los registros de cualquier nodo esclavo conectado a la red. En este formulario se puede realizar cualquier consulta o escritura permitida por las características de la trama. Para transmitir una trama se debe ingresar la dirección del nodo esclavo, configurar el campo de control del dispositivo, especificar la localidad de inicio de transacción, indicar la longitud de datos y dependiendo del tipo de trama se debe incluir los datos (trama de escritura de datos).

Para transmitir una trama se deben llenar todos los campos requeridos para dicha transmisión (los datos deben ser ingresados en decimal) y posteriormente se debe pulsar sobre el mensaje “TX” o sobre la tecla “Polling”, el software espera un instante de tiempo y ejecuta la recepción de datos.

El software calcula el código de error para la trama de transmisión cuando se transmite la trama. El código de error se visualiza en código hexadecimal.

Cuando se realiza una transmisión en que responde el nodo receptor, la trama es verificada para que no tenga errores y si no tiene errores la información de la trama receptada se escribe en los campos que corresponden a la recepción de datos.

Cuando un nodo esclavo responde a una trama se puede visualizar (codificados en ASCII) en el control RichTextBox. El control RichTextBox está localizado en la ficha “Buffer” del control TabStrip ubicado en el lado derecho del formulario “Análisis de Tramas”.

El dar doble clic sobre el control RichTextBox permite desglosar la información contenida en dicho control, la información que el control indica es: el dato receptado en código ASCII, el valor en decimal, el valor en hexadecimal y el valor en código binario (obsérvese la figura 20).

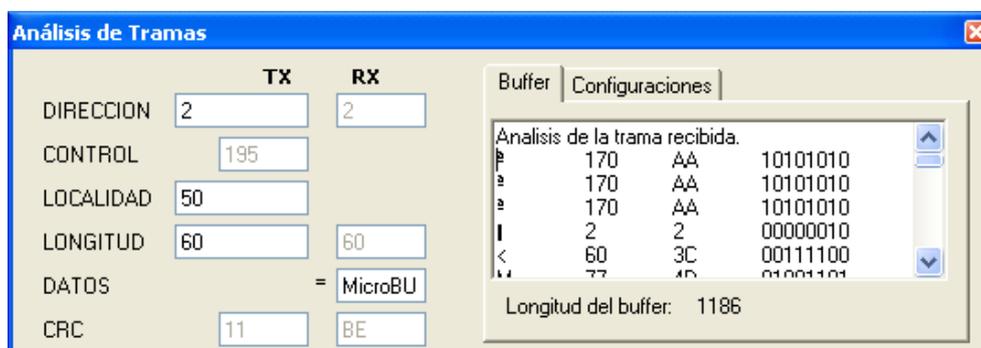


Figura 21. Descripción del formulario principal de la aplicación.

Figura 20. Formulario análisis de tramas.

Para variar el campo de control utilizado en la transmisión se debe dar clic sobre el área de la ficha “Configuración” del control TabStrip ubicado en el lado derecho del formulario “Análisis de Tramas”. También se debe indicar si es una trama de escritura o lectura, dicha selección está dada por las opciones “Escribir” y “Leer” ubicadas en la parte inferior del formulario “Análisis de Tramas”

La figura 21 indica las configuraciones de la trama como son: el tipo de memoria (EEPROM o RAM) a la que se refiere la trama, el número de banco de memoria RAM a la que se refiere la trama de transmisión y el tipo de datos que se pueden transmitir.

Cuando se ha seleccionado el tipo de datos byte y la opción de tramas de escritura, el software permite que se modifique el campo de datos con números enteros tipo decimal de 0 a 255. El campo de longitud se mantiene en uno y no puede ser modificado.

Cuando se ha seleccionado el tipo de datos entero y la opción de tramas de escritura, el software permite que se modifique el campo de datos con números enteros tipo decimal de 0 a 65535. El campo de longitud se mantiene en dos y no puede ser modificado, el software divide el campo de datos en dos bytes.

Cuando se ha seleccionado el tipo de datos char y la opción de tramas de escritura, el software permite que se modifique el campo de datos con un dato tipo carácter ASCII. El campo de longitud se mantiene en uno y no puede ser modificado.

Cuando se ha seleccionado el tipo de datos string y la opción de tramas de escritura, el software permite que se modifique el campo de datos con una cadena de caracteres. El campo de longitud es calculado dependiendo de la longitud de la cadena de caracteres. El software no toma en cuenta el carácter nulo al final de la cadena de caracteres tanto para la transmisión y recepción de tramas.



Figura 21. Configuración del campo de control.

Modificar el número de banco sólo es posible cuando la opción RAM está seleccionada.

## DESCRIPCIÓN DEL HARDWARE DE RED

El diagrama de la figura 22 indica la forma en que se debe conectar el hardware. El número máximo de estaciones esclavas es 31 y la longitud máxima de la red desde el adaptador de interfase debe ser 1200 metros. Para distancias mayores de 78 metros, se debe poner terminadores resistivos de 120 ohmios a los dos extremos de la red RS485.

La figura 23 indica el nombre de los terminales de los módulos utilizados. Cada módulo soporta voltajes de 7 a 15 voltios de entrada por el conector RJ11. Internamente cada módulo tiene un regulador de voltaje de 5 Voltios.

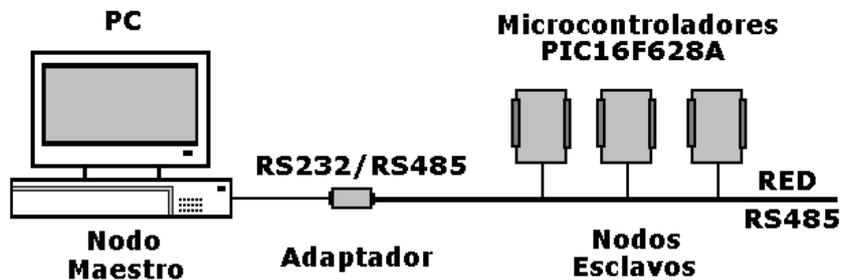


Figura 22. Descripción de la red RS485.

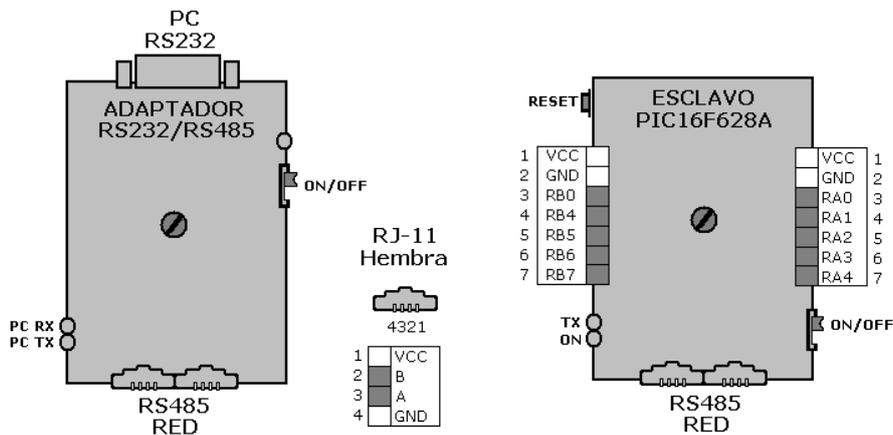


Figura 23. Descripción de terminales del hardware de red.

## EJEMPLO DE FUNCIONAMIENTO DEL SERVIDOR OPC

Como ejemplo para demostrar el funcionamiento del software configure el puerto de comunicación serie, defina los nodos de red Nodo2, Nodo4, Nodo5, Nodo7 y Nodo8 con las direcciones 2, 4, 5, 7 y 8 respectivamente, en el software MicroBUS como lo indica la figura 12. Al finalizar la configuración cierre la aplicación MicoBUS.

Instale el software “Software Toolbox OPC Quick Client” de la empresa “Software Toolbox”. Ejecute el software “OPC Quick Client” y añada un nuevo servidor como lo indica la figura 24.

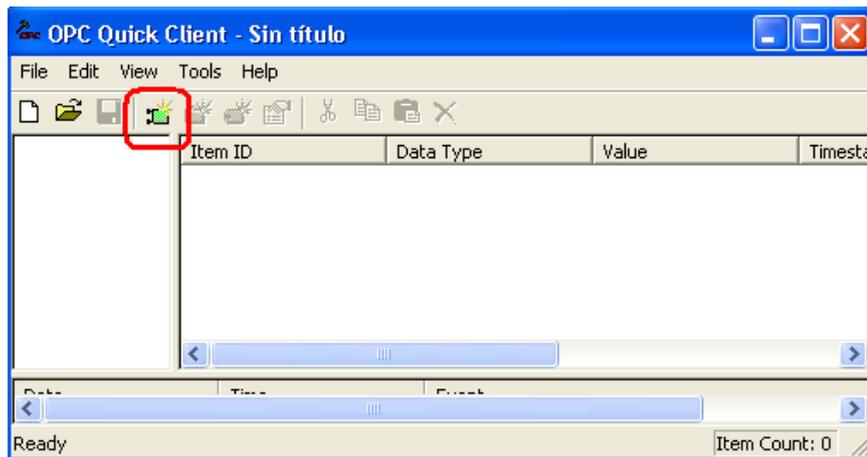


Figura 24. Software “OPC Quick Client”.

Para añadir y monitorear un servidor de datos OPC, el cliente “OPC Quick Client” debe explorar los servidores OPC existentes en el computador (nodo local) o en la red de área local (máquina remota).

Para monitorear la red de microcontroladores con el ejemplo propuesto, seleccione el servidor de datos OPC “MicroBUS\_v2” y pulse la tecla aceptar como lo indica la figura 25.

Para que cualquier aplicación cliente OPC acceda a los datos de un servidor OPC de manera local se necesita que el servidor OPC este desarrollándose en el mismo sistema operativo en el mismo computador.

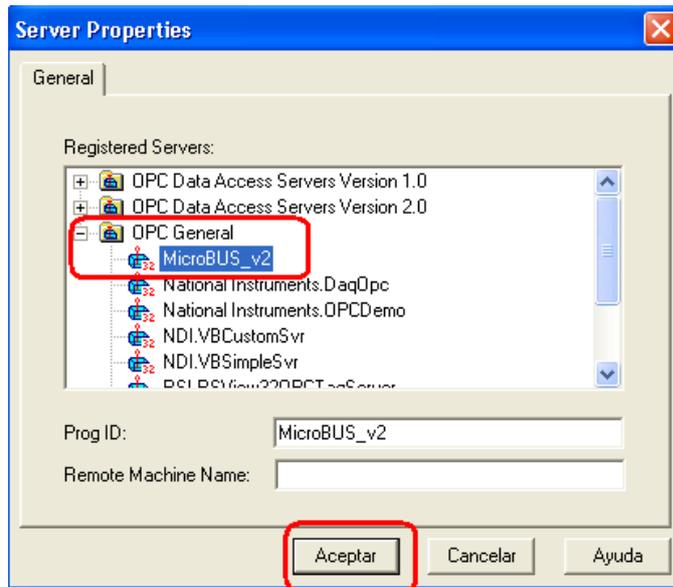


Figura 25. Selección del servidor de datos OPC a monitorear.

La figura 26 indica cómo debe crearse un grupo en el cliente OPC, dicho grupo se usará para contener los elementos a monitorear del servidor de datos OPC.

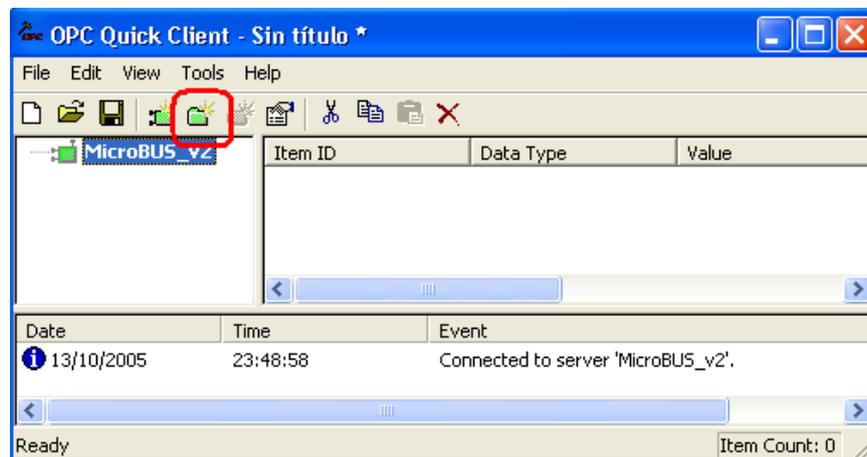


Figura 26. Creación de un grupo referido al servidor MicroBUS\_v2.

Siga los pasos indicados en la figura 27. El nombre hará referencia al grupo definido en la aplicación cliente OPC, la velocidad de actualización "Update Rate" actualizará los datos del cliente OPC (el cliente OPC actualizará los datos desde el servidor OPC cada 100 milisegundos).

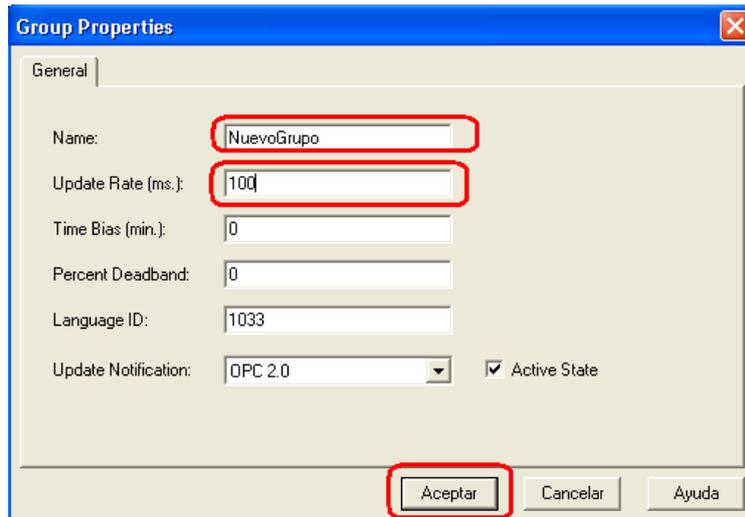


Figura 27. Configuración del grupo creado en el cliente OPC.

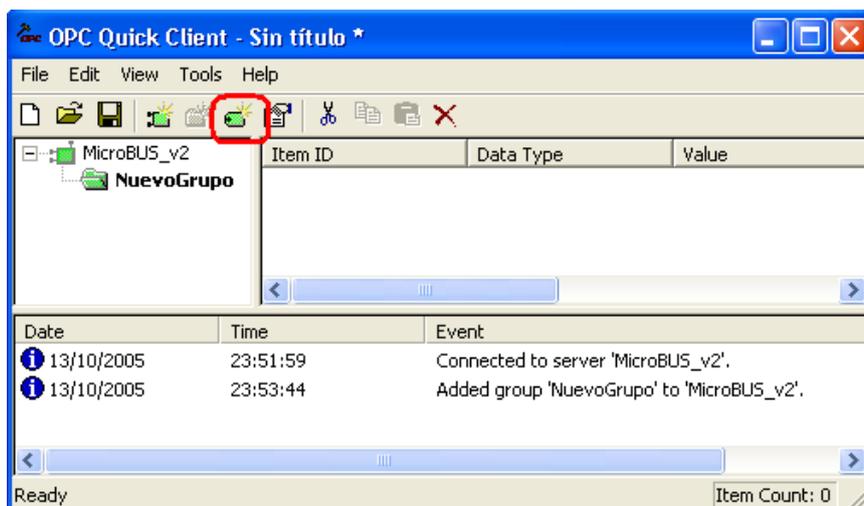


Figura 28. Creación de un elemento referido al servidor MicroBUS\_v2.

La figura 28 indica cómo añadir elementos al grupo creado en la figura 27. El formulario indicado en la figura 29 muestra la secuencia necesaria para añadir un elemento en un grupo de la aplicación cliente OPC.

El cliente OPC primero debe explorar los grupos del servidor OPC, como lo indica la marca 1 de la figura 29, después se debe seleccionar el elemento del servidor OPC, luego validar el elemento como lo indica la marca 3 de la figura 29 y por último seleccionar el tipo de dato con que trabajará el cliente OPC.

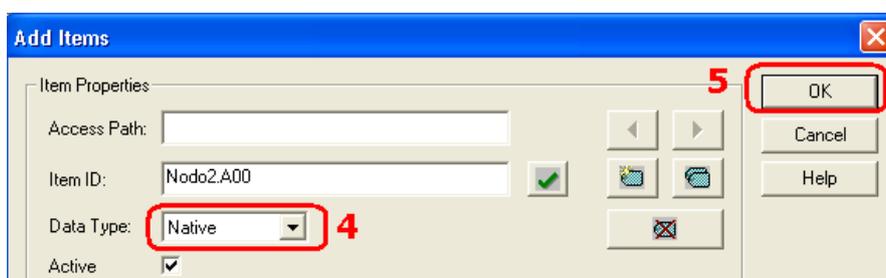


Figura 29. Exploración de elementos del servidor OPC MicroBUS\_v2.

En la figura 30 se indica el monitoreo de la variable “A00” (tag) del “Nodo2”, el tipo de dato usado para el monitoreo, el tiempo en que fue actualizado dicho elemento (tag), el número de actualizaciones hechas por la estación cliente OPC y el estado de la conexión con el servidor OPC.

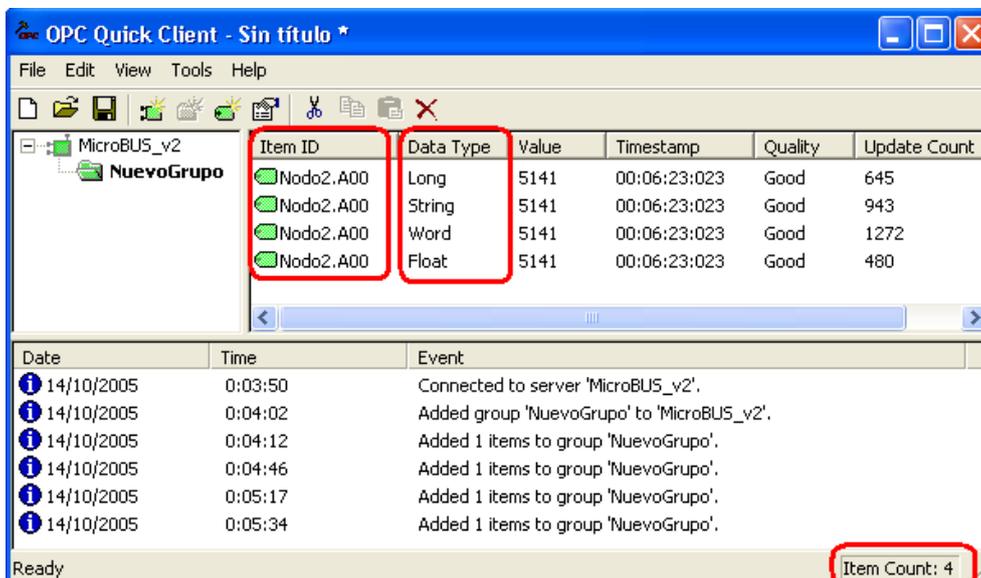


Figura 30. Monitoreo de la variable A00 del Nodo2.

**INDICE**

Proyecto.....2  
 Introducción.....2  
 Requerimientos para la instalación.....2  
 Instalación del software.....2

Inicio del programa.....	5
Descripción del formulario principal.....	6
Configuración de usuarios.....	9
Configuración del puerto serie.....	10
Definición de dispositivos de la red.....	11
Configuración de dispositivos.....	11
Configuración del servidor.....	12
Análisis de tramas.....	13
Descripción del hardware de red.....	15
Ejemplo de funcionamiento del servidor OPC.....	16

**LATACUNGA, NOVIEMBRE DEL 2005**

**ELABORADO POR:**

---

**ING. MARCO SALAZAR**

---

**ING. NANCY GUERRÓN  
DIRECTOR DE CARRERA**

---

**DR. EDUARDO VÁSQUEZ ALCÁZAR  
SECRETARIO ACADÉMICO**