

IMPLEMENTACIÓN DE UNA RED INDUSTRIAL CAN BAJO UN SISTEMA SCADA



**JULIÁN ANDRÉS VIDAL ILLERA
MILTON SERGIO ZÚÑIGA GALINDEZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
POPAYÁN
2006**

IMPLEMENTACIÓN DE UNA RED INDUSTRIAL CAN BAJO UN SISTEMA SCADA



Monografía presentada como requisito parcial para optar por el título de
Ingenieros en Automática Industrial

**JULIÁN ANDRÉS VIDAL ILLERA
MILTON SERGIO ZÚÑIGA GALINDEZ**

Director
OSCAR AMAURY ROJAS ALVARADO
Ingeniero Electrónico Especialista en Informática Industrial

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
POPAYÁN
2006**

CONTENIDO

| | |
|--|-----------|
| INTRODUCCIÓN..... | 1 |
| 1 PROTOCOLO CAN..... | 3 |
| 1.1 INTRODUCCIÓN | 3 |
| 1.2 ESPECIFICACIONES DEL PROTOCOLO..... | 5 |
| 1.3 NIVEL FÍSICO..... | 6 |
| 1.3.1 CONCEPTOS BÁSICOS | 6 |
| 1.3.2 MEDIO FÍSICO | 7 |
| 1.3.3 TOPOLOGÍA..... | 7 |
| 1.3.4 NIVELES DE SEÑALES CAN..... | 8 |
| 1.3.5 CONECTORES | 9 |
| 1.4 NIVEL DE ENLACE | 9 |
| 1.4.1 CONCEPTOS BÁSICOS | 9 |
| 1.4.2 NIVEL LLC | 10 |
| 1.4.3 NIVEL MAC..... | 10 |
| 1.5 NIVEL DE APLICACIÓN..... | 10 |
| 1.6 FORMATO DE LA TRAMA CAN | 11 |
| 1.6.1 CAN 2A – ESTÁNDAR..... | 11 |
| 1.6.2 CAN 2B – EXTENDIDA..... | 13 |
| 1.6.3 TIPOS DE TRAMA | 14 |
| 1.6.3.1 Trama de Datos | 14 |
| 1.6.3.2 Trama Remota..... | 14 |
| 1.6.3.3 Trama de Error | 14 |
| 1.6.3.4 Trama de Sobrecarga..... | 15 |
| 1.7 CÓMO FUNCIONA CAN?..... | 16 |
| 1.7.1 MÉTODO DE ARBITRAJE..... | 16 |
| 1.7.2 COMUNICACIÓN BASADA EN MENSAJE | 17 |
| 1.8 TIPOS DE ERRORES | 18 |
| 1.8.1 ERROR DE BIT..... | 18 |

| | | |
|-------|-----------------------|----|
| 1.8.2 | ERROR DE RELLENO..... | 18 |
| 1.8.3 | ERROR DE CRC | 19 |
| 1.8.4 | ERROR DE FORMA..... | 20 |
| 1.8.5 | ERROR DE ACK | 20 |
| 1.8.6 | ERROR EN EOF | 20 |

2 DESCRIPCIÓN DE LOS PROCESOS POR INTEGRAR EN LA RED21

2.1 INTRODUCCIÓN21

2.2 PLANTA DE NIVEL Y FLUJO DE AGUA.....21

2.2.1 DESCRIPCIÓN Y FUNCIONAMIENTO DE LA PLANTA DE NIVEL Y FLUJO DE AGUA21

2.2.1.1 Descripción de los Componentes Planta de Nivel y Flujo de Agua24

2.2.1.1.1 Transmisor de Presión Diferencial Yokogawa EJA110A.....24

2.2.1.1.2 Transmisor de Flujo GF + SIGNET 8550-125

2.2.1.1.3 Válvula DWYER Serie ABV25

2.2.1.1.4 Bomba Centrífuga de Agua IHM.S.A Modelo 1B-1/4 Serie 000011526

2.2.2 ENTRADAS/SALIDAS DE LA PLANTA DE NIVEL Y FLUJO DE AGUA.....27

2.3 PLANTA DE TEMPERATURA (MÓDULO PT326).....27

2.3.1 DESCRIPCIÓN Y FUNCIONAMIENTO DEL MÓDULO PT326.....27

2.3.1.1 Descripción de los Componentes del Módulo PT32629

2.3.2 ENTRADAS/SALIDAS DEL MÓDULO PT326.....30

3 DESCRIPCIÓN DE LOS DISPOSITIVOS ELECTRÓNICOS EMPLEADOS PARA EL DESARROLLO DE LA INTERFAZ CAN31

3.1 INTRODUCCIÓN31

3.2 DESCRIPCIÓN DE LAS POSIBLES ARQUITECTURAS DE LA RED INDUSTRIAL

31

3.2.1 ARQUITECTURA CON EL CONTROLADOR CAN EXTERNO32

3.2.2 ARQUITECTURA CON EL CONTROLADOR CAN INCLUIDO INTERNAMENTE EN EL PIC34

3.2.3 COMPARACIÓN DE LAS ARQUITECTURAS35

3.2.4 VENTAJAS Y DESVENTAJAS DE CADA UNA DE LAS ARQUITECTURAS37

3.3 CARACTERÍSTICAS Y MODOS DE OPERACIÓN DE LOS DISPOSITIVOS

ELECTRÓNICOS USADOS EN LA RED37

| | | |
|------------|---|-------------------|
| 3.3.1 | PIC 18F258..... | 37 |
| 3.3.2 | TRANSDUCTOR CAN MCP2551 | 46 |
| 4 | <u>PROGRAMACIÓN DE LA INTERFAZ CAN</u> | <u>47</u> |
| 4.1 | APLICACIÓN “MAESTRO” | 47 |
| 4.1.1 | GENERALIDADES APLICACIÓN “MAESTRO” | 48 |
| 4.1.2 | CONFIGURACIÓN APLICACIÓN “MAESTRO” | 51 |
| 4.1.3 | CONFIGURACIÓN DEL MÓDULO CAN | 52 |
| 4.1.4 | GENERACIÓN DE CÓDIGO | 59 |
| 4.2 | DIAGRAMAS DE FLUJO PARA LA PROGRAMACIÓN DE LOS NODOS | 60 |
| 4.2.1 | NODO MAESTRO | 60 |
| 4.2.2 | NODO ESCLAVO 1 | 62 |
| 4.2.3 | NODO ESCLAVO 2 | 64 |
| 5 | <u>DISEÑO DE LAS INTERFACES CAN</u> | <u>67</u> |
| 5.1 | INTRODUCCIÓN | 67 |
| 5.2 | NODOS ESCLAVOS 1 Y 2..... | 67 |
| 5.3 | NODO MAESTRO..... | 71 |
| 5.4 | FUENTES DE ALIMENTACIÓN Y BUS..... | 73 |
| 6 | <u>DISEÑO E IMPLEMENTACIÓN DEL SISTEMA SCADA</u> | <u>76</u> |
| 6.1 | SISTEMA SCADA..... | 76 |
| 6.2 | PROGRAMACIÓN PLC MICROLOGIX 1500 | 77 |
| 6.3 | HMI’s (HUMAN MACHINE INTERFACE) | 86 |
| 6.3.1 | MENÚ DE INGRESO | 87 |
| 6.3.2 | MENÚ DE OPCIONES DEL USUARIO | 88 |
| 6.3.3 | MENÚ PRINCIPAL..... | 91 |
| 6.3.4 | PANTALLA DEL PROCESO DE NIVEL Y FLUJO DE AGUA | 92 |
| 6.3.5 | PANTALLA DEL PROCESO DE TEMPERATURA | 96 |
| 7 | <u>CONCLUSIONES.....</u> | <u>100</u> |

8 PROYECTOS FUTUROS102

REFERENCIAS BIBLIOGRAFICAS103

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1. Niveles del modelo OSI implementados por CAN..... | 6 |
| Figura 2. Topología de la red..... | 8 |
| Figura 3. Niveles de señales CAN..... | 8 |
| Figura 4. Conector RJ45..... | 9 |
| Figura 5. Trama CAN 2A estándar..... | 11 |
| Figura 6. Trama CAN 2B Extendida..... | 13 |
| Figura 7. Ejemplo del método de arbitraje..... | 17 |
| Figura 8. Campos sometidos a “ <i>bit stuffing</i> ”..... | 19 |
| Figura 9. Constantes para el método de CRC..... | 19 |
| Figura 10. Planta de nivel y flujo de agua..... | 22 |
| Figura 11. Diagrama de instrumentos según el estándar SP5 de ISA de la planta de nivel y flujo de agua..... | 23 |
| Figura 12. Diagrama ISA de la configuración usada..... | 24 |
| Figura 13. Transmisor de presión diferencial Yokogawa EJA110A..... | 25 |
| Figura 14. Transmisor de flujo GF + SIGNET..... | 25 |
| Figura 15. Válvula DWYER serie ABV..... | 26 |
| Figura 16. Bomba centrífuga..... | 26 |
| Figura 17. Módulo PT326..... | 28 |
| Figura 18. Esquema detallado del módulo PT326..... | 28 |
| Figura 19. Sensor de temperatura LM35..... | 29 |
| Figura 20. Sensor LM35 y circuito de adecuación de señal..... | 30 |
| Figura 21. Arquitectura de la red con el controlador CAN externo MCP2515..... | 33 |
| Figura 22. Arquitectura de la red con el PIC18F258..... | 35 |
| Figura 23. Distribución de los “ <i>buffers</i> ” de transmisión..... | 40 |
| Figura 24. Diagrama de flujo de la transmisión de mensajes..... | 42 |
| Figura 25. Distribución de “ <i>buffers</i> ” de recepción..... | 43 |
| Figura 26. Diagrama de flujo de la recepción de mensajes..... | 45 |
| Figura 27. Ventana principal de la aplicación “ <i>maestro</i> ”..... | 48 |
| Figura 28. Ventana de los módulos disponibles..... | 49 |

| | |
|---|----|
| Figura 29. Ventana del módulo seleccionado..... | 50 |
| Figura 30. Ventana de las opciones de configuración..... | 50 |
| Figura 31. Menú y barra de herramientas | 51 |
| Figura 32. Selección del módulo CAN..... | 52 |
| Figura 33. Ventana de las opciones de configuración del módulo CAN..... | 53 |
| Figura 34. Ventana de configuración de los parámetros del módulo CAN..... | 53 |
| Figura 35. Ventana de error de la configuración | 54 |
| Figura 36. “Nominal bit time” | 56 |
| Figura 37. Ventana de la generación de código | 59 |
| Figura 38. Diseño circuital de los nodos esclavos 1 y 2..... | 67 |
| Figura 39. Diseño circuital del nodo maestro | 71 |
| Figura 40. Diseño circuital del bus y de la fuente de alimentación de cada nodo | 74 |
| Figura 41. Arquitectura de la red | 77 |
| Figura 42. Ventana de configuración del canal | 78 |
| Figura 43. Ventana de configuración del canal 1 | 79 |
| Figura 44. Programa en lenguaje “ladder” | 84 |
| Figura 45. Ventana del menú de ingreso..... | 87 |
| Figura 46. Ventana de confirmación del “User” y el “Password” | 88 |
| Figura 47. Ventana del “User” y/o “password” incorrecto | 88 |
| Figura 48. Ventana de opciones del usuario | 89 |
| Figura 49. Ventana de cuentas de usuario..... | 90 |
| Figura 50. Ventana de códigos de seguridad | 90 |
| Figura 51. Ventana de cambio de “Password” | 91 |
| Figura 52. Ventana menú principal..... | 92 |
| Figura 53. Mímico de la planta de nivel y flujo de agua..... | 93 |
| Figura 54. Mímico de una situación de alarma..... | 94 |
| Figura 55. Ventana de alarmas de la planta de nivel y flujo de agua | 95 |
| Figura 56. Ventana de tendencias de la planta de nivel y flujo de agua | 95 |
| Figura 57. Ventana de control manual de la planta de nivel y flujo de agua | 96 |
| Figura 58. Mímico de la planta de temperatura | 97 |
| Figura 59. Ventana de alarmas de la planta de temperatura | 98 |
| Figura 60. Ventana de tendencias de la planta de temperatura | 98 |
| Figura 61. Ventana de control manual de la planta de temperatura..... | 99 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1. Longitud del bus, velocidad y tiempo máximo de transmisión..... | 7 |
| Tabla 2. Distribución de pines del conector RJ45 | 9 |
| Tabla 3. Descripción de la trama CAN 2A estándar | 12 |
| Tabla 4. Descripción de la trama CAN 2B extendida | 14 |
| Tabla 5. Componentes y Entradas/Salidas de la planta de nivel y flujo de agua | 27 |
| Tabla 6. Componentes y Entradas/Salidas del módulo PT326 | 30 |
| Tabla 7. Comparación de las arquitecturas | 36 |
| Tabla 8. Características del PLC Micrologix 1500 | 46 |
| Tabla 9. Características del módulo analógico 1769-IF4XOF2 | 46 |

LISTA DE ANEXOS

Anexo A. Diseño y Configuración del Software de Programación para cada PIC18F258 del Nodo Maestro y los Nodos Esclavos

Anexo B. Adición de Nodos

INTRODUCCIÓN

En la actualidad la automatización es parte vital para el sostenimiento de una empresa o industria y en gran parte para el crecimiento de un país. A su vez la supervisión y la comunicación de los procesos productivos forman una parte trascendental en la automatización.

Las redes fueron introducidas en la automatización industrial en los años 70's aproximadamente y se utilizaron en primer lugar en sistemas de control digital directo (DDC por sus siglas en ingles). Posteriormente y hasta hoy en día son usadas en los sistemas de control distribuido (DCS) y sistemas de controladores lógicos programables (PLC's) para conectar todos los dispositivos de la pirámide CIM bajo uno o más buses de campo. Sin embargo, las comunicaciones digitales en dispositivos del piso de planta como transmisores no fueron vistas sino hasta los años 80's y la comunicación de los dispositivos de campo a través de un bus de campo sólo tuvo buena acogida en los 90's. Un bus de campo es un protocolo definido para conectar dispositivos inteligentes o sensores inteligentes o actuadores inteligentes a otros dispositivos como PLC's.

Cuando las comunicaciones digitales aparecieron cada vendedor creó su propio protocolo de comunicaciones independientemente de los otros. Rápidamente aparecieron en el mercado diversos protocolos propietarios y productos que sólo podían trabajar con otros productos del mismo vendedor. Esta situación tenía muchas desventajas. Una de ellas era que los vendedores no poseían una suficiente gama de productos que pudieran suplir todas las necesidades del cliente, es decir, la selección de equipos era muy limitada; por esta razón se hacia necesario la utilización de productos de diferentes vendedores dentro de la industria. Pero los dispositivos ofrecidos por los diferentes fabricantes tenían incompatibilidad de protocolos, lo cual resultaba en una imposibilidad de interconectar los dispositivos bajo un mismo protocolo teniendo así "islas automatizadas".

Para darle solución a este problema lo primero que se realizó fue desarrollar “drivers” de comunicaciones pero tenían un alto costo y requerían mucho tiempo de desarrollo. Además para cada combinación dispositivo – software era necesario un “driver” específico, lo cual llevo a una gran oferta de fabricantes de “drivers”.

Una vez se adquiría un sistema propietario se podía decir que la planta estaba “cerrada” hacia los demás fabricantes. Si se quería integrar la planta se requería adquirir hardware y software del mismo fabricante o adquirir “drivers” de comunicaciones lo cual era extremadamente costoso. Por esta razón expertos en la industria empezaron a trabajar en estándares de buses de campo lo cual daría la posibilidad de tener en la industria lo que se conoce como un “sistema abierto”.

Actualmente existen en el mercado dispositivos inteligentes de diferentes fabricantes, ya sean sensores o actuadores, que son capaces de interconectarse con diferentes buses de campo. Los buses de campo más usados en la industria son Profibus, Foundation Fieldbus, Modbus, CAN (Controller Area Network), AS-I, entre otros. Las principales ventajas de usar un bus de campo son: menos errores a la hora del cableado, simplificación del cableado, facilidad en la detección de errores en el sistema y facilidad de mantenimiento del sistema.

El interés por los avances en el campo de la automatización industrial y más específicamente por los buses de campo es la motivación del presente trabajo de grado que describe el desarrollo del proyecto *Implementación de una Red Industrial CAN Bajo un Sistema SCADA*. El cual es la integración de los procesos de nivel y flujo de agua y temperatura (Módulo PT326) disponibles en el Laboratorio de Control de Procesos del programa de Ingeniería en Automática Industrial bajo un bus de campo CAN, desarrollando la interfaz CAN para cada dispositivo que compone los procesos (sensores y actuadores) y supervisando la red desde un sistema SCADA implementado en RSVIEW®32™ de Rockwell Automation permitiendo así, la comunicación e integración de hardware de diferentes fabricantes siguiendo los lineamientos del proyecto de investigación “*Convergencia de las Redes Industriales sobre Sistemas SCADA*” registrado ante la Vicerrectoría de Investigaciones de la Universidad del Cauca.

1 PROTOCOLO CAN

1.1 INTRODUCCIÓN

Hasta la década del 60 el control de un proceso productivo se basó en la existencia de múltiples lazos analógicos, cada uno de los cuales controlaba una variable del sistema. Esta situación dio lugar a la aparición de diversos paneles de control, con un gran número de variables dinámicas. En estas condiciones, al operador del sistema le resultaba difícil supervisar con eficacia el sistema productivo en su totalidad.

En los años 60 comenzaron a introducirse computadoras en el control de procesos, realizando tareas de monitoreo del estado de las variables del sistema, generando alarmas cuando se detectaban niveles anómalos. Posteriormente se las comenzó a incluir en las labores de control, debido a que por su capacidad de cálculo y programación podían reemplazar al panel de control. El principal inconveniente que se enfrentaba era la mayor debilidad del sistema al existir un punto de fallo crítico, el propio computador y, además la complejidad de los algoritmos que éstas debían desarrollar (si se tiene en cuenta la tecnología de las computadoras en esos días) era simplemente muy alta para su implementación práctica.

Durante la década del 70, el desarrollo de los microprocesadores, microcontroladores y controladores lógicos programables, permitió la descentralización de los sistemas de control creándose los DCS (Sistemas de Control Distribuido). En este tipo de esquema un PLC o un microprocesador controla uno o más lazos del sistema, y se comunica con otros elementos de su nivel y con el nivel superior de control o supervisión. La falla de un elemento del nivel superior no compromete necesariamente el funcionamiento de los dispositivos en el nivel inferior, minimizando por ende su incidencia en el proceso.

Todos estos factores (además de la aparición de sensores inteligentes, máquinas de control numérico, robots, etc.) hicieron flexible el proceso productivo creando la necesidad de programación y control de forma remota.

En los años 80s se empezaron a implementar los buses de campo en la industria. Primero como una solución para reducir el profuso cableado, pero con el tiempo ha llegado a tener tareas de mantenimiento, diagnóstico, detección de errores, etc. En principio salieron al mercado como protocolos propietarios: las empresas que los desarrollan construyen las características del protocolo de acuerdo a sus especificaciones y conveniencia, de una manera subjetiva. Esto limitaba la capacidad de elección de las industrias a la hora de la compra, ya que, dependían de los proveedores que implementaban dicho protocolo. En la actualidad existe la tendencia a la normalización; con este fin, se han desarrollado protocolos abiertos de comunicación, entre los que se encuentra la red industrial CAN que soporta los estándares industriales.

Un bus de campo en su connotación más escueta permite la intercomunicación, la interoperabilidad, la inter-conectividad y la inter-cambiabilidad entre los diferentes dispositivos industriales. Otras características principales son:

- Su significativa reducción en el cableado necesario para el control de una instalación. Cada componente sólo requiere un cable para la conexión de los diversos nodos. En comparación con otros tipos de redes, dispone de herramientas de administración del bus que permiten la reducción del número de horas necesarias para la instalación y puesta en marcha.
- Permiten a los operadores monitorear todos los dispositivos que integran el sistema e interpretar fácilmente las interacciones entre ellos. De esta forma, la detección de las fuentes de problemas en la planta y su corrección resulta mucho más sencilla, reduciendo los costos de mantenimiento y el tiempo de parada de la planta.
- Las prestaciones del sistema mejoran con el uso de la tecnología de los buses de campo debido a la simplificación en la forma de obtener información de la planta desde los distintos sensores. Las mediciones de los distintos elementos de la red están disponibles para todos los demás dispositivos.
- La simplificación en la obtención de datos permitirá el diseño de sistemas de control más eficientes.

- La comunicación bi-direccional no sólo es entre los dispositivos de campo y los sistemas de control, sino también entre los propios dispositivos de campo.
- Sólo incluyen 3 capas (Física, Enlace y Aplicación), y un conjunto de servicios de administración. El usuario no tiene que preocuparse de las capas de enlace o de aplicación. Sólo necesita saber cuál es la funcionalidad.
- Al usuario sólo se le exige tener un conocimiento mínimo de los servicios de administración de la red, ya que parte de la información generada por dichos servicios puede ser necesaria para la reparación de averías en el sistema. De hecho, prácticamente, el usuario sólo debe preocuparse de la capa física y la capa de usuario.

1.2 ESPECIFICACIONES DEL PROTOCOLO

La red CAN, “*Controller Area Network*” fue creada inicialmente por el alemán Robert Bosch a mediados de los 80s como una solución para establecer una comunicación serial y robusta en automóviles. El objetivo principal consistía en proporcionar a los vehículos mayor confiabilidad, seguridad y mejorar la eficiencia del combustible; y por otro lado, reducir el cableado, tanto en su cantidad como en su complejidad. La red CAN ha ganado popularidad, permitiéndose la implementación de ésta en las aplicaciones de automatización industrial.

CAN es un protocolo abierto e implementa las capas de nivel físico y de nivel de enlace del modelo OSI (Open System Interconnection). El nivel físico intencionalmente se dejó abierto para proporcionar máxima flexibilidad. Se puede utilizar par trenzado, RF, etc. El nivel de aplicación se implementa en el software del desarrollador.

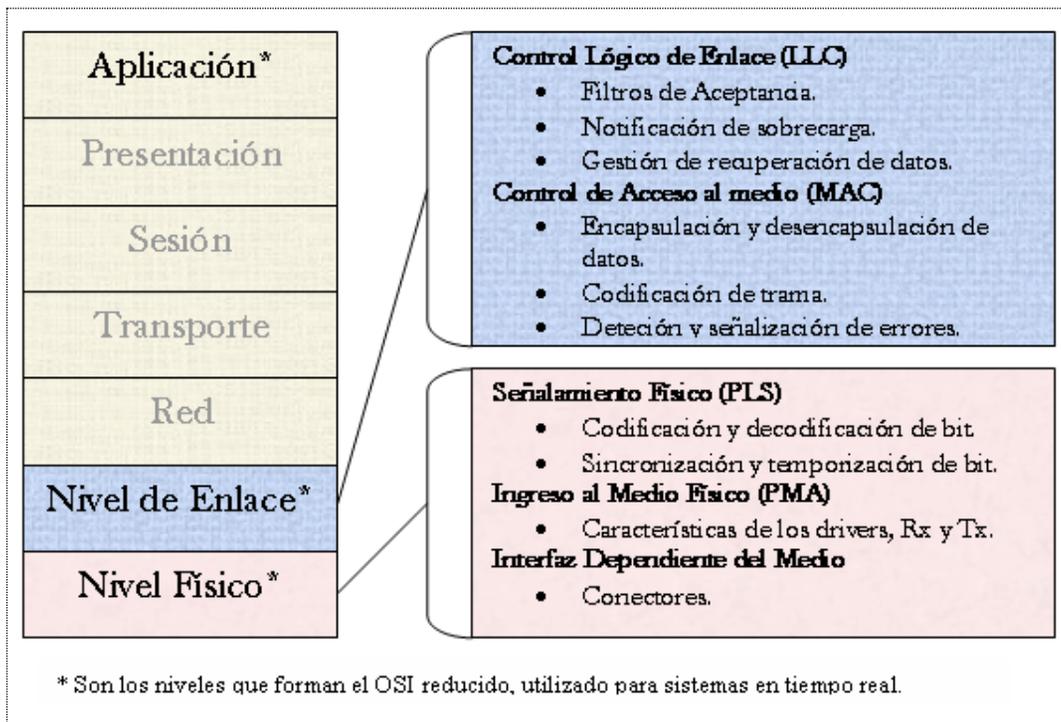


Figura 1. Niveles del modelo OSI implementados por CAN

Principales características:

- Modos de operación Multimaestro y Maestro-Esclavo.
- Modo de transmisión “*broadcast*” y “*multicast*”.
- Mecanismo de acceso al medio no destructivo (CSMA/DCR).
- El número máximo de nodos está limitado en la práctica por los “*drivers*”, pero suele ser normal un tope de 32 a 64.

1.3 NIVEL FÍSICO

1.3.1 Conceptos Básicos

Esta capa se basa en la norma ISO 11898. Es responsable de la transferencia de bits entre los distintos nodos que componen la red. Define aspectos como niveles de señal, codificación, sincronización y tiempos en que los bits se transfieren al bus.

1.3.2 Medio Físico

El medio físico es una línea de bus de dos hilos con un retorno común que es terminada en ambos extremos por resistores de 120 ohmios que representan la impedancia característica de la línea. La longitud máxima es 1 kilómetro. Se permite utilizar los dispositivos puente o los repetidores para aumentar el número de los nodos del bus que pueden ser conectados, o para aumentar la distancia permitida entre los nodos del bus (p.ej. a más de 1 kilómetro), o para proporcionar el aislamiento galvánico (un dispositivo puente contiene por lo menos dos controladores CAN y dos transductores). A continuación en la Tabla 1 se muestra las velocidades máximas de transmisión dependiendo de la longitud del bus; al igual que el tiempo máximo de transmisión del mensaje.

| LONGITUD DEL BUS | VELOCIDAD EN BIT/S | Tiempo máximo de transmisión (para una longitud máxima de mensaje de 129 bits) |
|--------------------------|---------------------------|---|
| Hasta 25 metros | 1Mbit/s | $1 \mu\text{s} \times 129 \text{ bits} = 129 \mu\text{s}$ |
| Hasta 100 metros | 500 Kbit/s | $2 \mu\text{s} \times 129 \text{ bits} = 258 \mu\text{s}$ |
| Hasta 250 metros | 250 Kbit/s | $4 \mu\text{s} \times 129 \text{ bits} = 516 \mu\text{s}$ |
| Hasta 500 metros | 125 Kbit/s | $8 \mu\text{s} \times 129 \text{ bits} = 1.032 \text{ ms}$ |
| Hasta 1000 metros | 50 Kbit/s | $20 \mu\text{s} \times 129 \text{ bits} = 2.580 \text{ ms}$ |
| Hasta 2500 metros | 20 Kbit/s | $50 \mu\text{s} \times 129 \text{ bits} = 6.450 \text{ ms}$ |

Tabla 1. Longitud del bus, velocidad y tiempo máximo de transmisión

1.3.3 Topología

La topología del cableado debe estar tan cerca como sea posible a una sola estructura de línea, para reducir al mínimo las flexiones del mismo. Los segmentos del cable para la conexión de los nodos del bus deben ser tan cortos como sea posible, especialmente en tasas altas de bit. La topología utilizada es tipo bus con derivaciones de corta longitud. El bus se cierra en los extremos con impedancias de carga. El uso de los voltajes diferenciales permite que las redes CAN funcionen cuando una de las líneas de señales es separada.

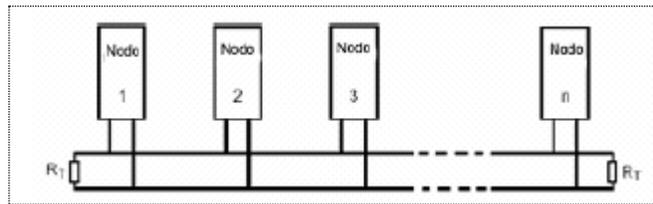


Figura 2. Topología de la red

1.3.4 Niveles de Señales CAN

Los niveles de señal son manipulados por un transductor de la empresa Microchip con referencia MCP2551. Así, este dispositivo convierte las señales provenientes del microcontrolador (que tiene implementado el módulo CAN) en señales que entiende el protocolo CAN y viceversa. Los nodos conectados al bus interpretan dos niveles lógicos denominados:

- Dominante: La tensión diferencial (CAN_H - CAN_L) es del orden de 2.0 V con CAN_H = 3.5 V y CAN_L = 1.5 V (nominales).
- Recesivo: La tensión diferencial (CAN_H - CAN_L) es del orden de 0 V con CAN_H = CAN_L = 2.5 V (nominales).

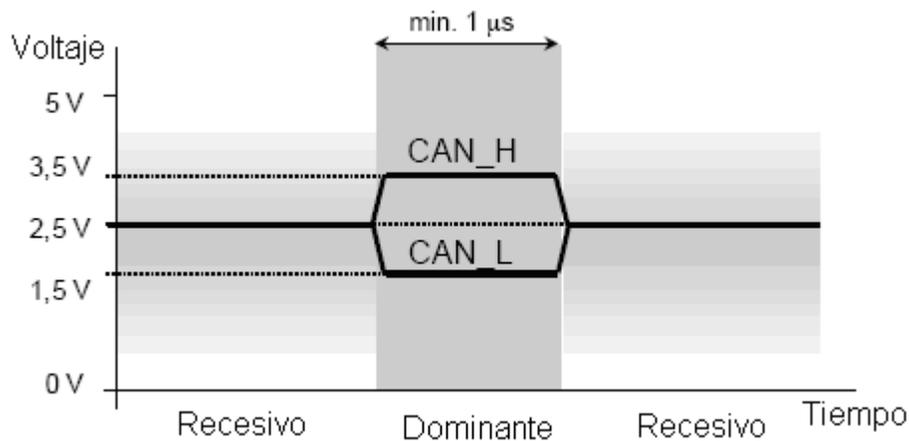


Figura 3. Niveles de señales CAN

El bus CAN funciona en ambientes extremadamente ruidosos. Gracias al simple par trenzado y las entradas diferenciales de CAN que eliminan el ruido con eficacia, dentro del rango de modo común.

1.3.5 Conectores

Para la aplicación se utiliza los conectores RJ45 haciendo uso del cable UTP. En la Figura 4 y Tabla 2 se presenta su representación gráfica y la distribución de sus pines respectivamente.

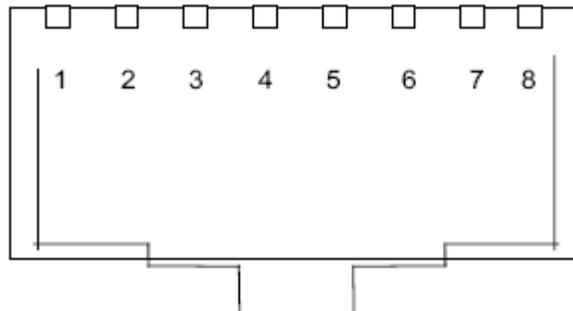


Figura 4. Conector RJ45

| Pin | Señal | Descripción |
|-----|----------|---|
| 1 | CAN_H | Línea de bus (Dominante alto) |
| 2 | CAN_L | Línea de bus (Dominante bajo) |
| 3 | CAN_GND | No usado |
| 4 | 12V | Pin reservado que es usado para alimentar a los nodos |
| 5 | -12V | Pin reservado que es usado para alimentar a los nodos |
| 6 | CAN_SHLD | No usado |
| 7 | CAN_GND | CAN GND |
| 8 | CAN_V+ | Usado para alimentar a los nodos con 5V |

Tabla 2. Distribución de pines del conector RJ45

1.4 NIVEL DE ENLACE

1.4.1 Conceptos Básicos

Implementa el nivel LLC (Control Lógico de Enlace, por sus siglas en español) y el nivel MAC (Control de acceso al medio, por sus siglas en español).

1.4.2 Nivel LLC

Este nivel proporciona facilidades para la transmisión de bloques de datos entre dos o más nodos de la red. Esto es, organiza los 1's y los 0's del nivel físico en formatos o grupos lógicos de información. Las principales funciones de éste nivel son:

- Funciones de Filtro de Aceptación.
- Notificación de Sobrecarga.
- Manejo del proceso de recuperación de datos.

1.4.3 Nivel MAC

Las principales funciones de este nivel son:

- Responsable de la encapsulación y desencapsulación de los datos.
- Manejo de acceso al medio por CSMA/CD.
- Codificación de tramas ("*stuffing*" / "*de-stuffing*").
- Detección de error.
- Señalización de error.

El protocolo de comunicación de CAN se basa en la norma de la IEEE 802.3 - CSMA/CD – Acceso múltiple con detección de portadora y detección de colisiones. Lo que el CSMA significa es que cada nodo en la red debe monitorear el bus hasta encontrar un periodo de no actividad antes de intentar enviar un mensaje (detección de portadora); una vez este periodo de no actividad ocurra, todos los nodos en la red tendrán las mismas posibilidades de transmitir un mensaje (acceso múltiple). La detección de colisiones – CD – actúa de la siguiente manera: si dos nodos empiezan a transmitir al mismo tiempo, los nodos detectarán la "colisión", y un método de arbitraje no-destructivo se encargará de determinar cuál de los nodos sigue transmitiendo y cuál cesará de transmitir.

1.5 NIVEL DE APLICACIÓN

El protocolo CAN no especifica el nivel de aplicación, dando la posibilidad de tener varias opciones al momento de implementar la red. Existen varias herramientas que

implementan protocolos de alto nivel (HLP) que proporcionan la capa de aplicación. Para la aplicación actual se hace uso del PLC Micrologix 1500 de Rockwell Automation que se comunica con el nodo maestro CAN a través de la norma RS-232. Dicho sistema es supervisado por el software de Rockwell Automation. Las principales funciones de la capa de aplicación son:

- Distribución de identificadores a los mensajes.
- Mecanismos de intercambio para bloques de datos mayores de 8 bytes.
- Gestión del estado de la red.

1.6 FORMATO DE LA TRAMA CAN

El protocolo implementa dos versiones: CAN 2A estándar y CAN 2B extendida.

1.6.1 CAN 2A – Estándar

En este formato el ID (Identificador) tiene 11 bits, lo que le da la posibilidad de alcanzar 2032 identificadores diferentes.

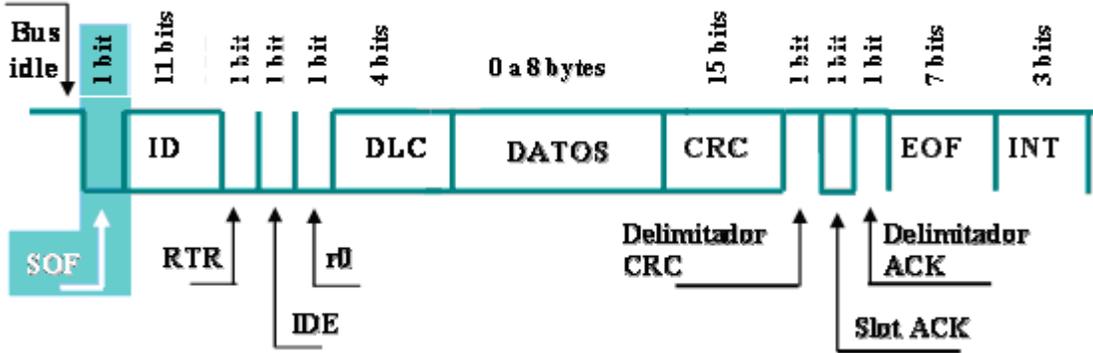


Figura 5. Trama CAN 2A estándar

| Siglas | | Descripción |
|-----------------------------|-----|--|
| SOF | | Inicio de trama: Se utiliza para la sincronización entre nodos. Siempre dominante. |
| Campo de Arbitración | ID | Identificador de mensaje: 11 bits para este tipo de trama. Establece un límite en el número de mensajes diferentes. Teóricamente $2^{11} = 2048$ mensajes posibles. |
| | RTR | Petición de transmisión remota: Identifica el tipo de trama en el bus: Dominante → Trama de datos Recesivo → Trama remota |
| Campo de Control | IDE | Identificador extendido: Identifica la versión del protocolo: Dominante → Versión CAN 2A - Estándar Recesivo → Versión CAN 2B - Extendido |
| | r0 | Bit reservado: Siempre dominante. Reservado para futuras versiones. |
| | DLC | Longitud de datos: Especifica la cantidad de bytes de datos, en el campo Datos (0 a 8 bytes) |
| Datos | | Campo de datos: Datos enviados, de acuerdo a la longitud especificada en el campo DLC |
| CRC | | Chequeo de redundancia cíclica: Secuencia de 15 bits, utilizada para detectar errores de transmisión. |
| | | Delimitador CRC: Indica el final del CRC. Siempre recesivo. |
| ACK | | Acknowledge: Compuesto por 2 bits: el slot ACK y el delimitador del ACK. Durante el bit slot ACK, el nodo transmisor envía un nivel recesivo. Un nodo que ha recibido la trama libre de errores, envía un nivel dominante (independiente de si ese nodo está configurado para aceptar ese mensaje o no). El bit <i>Delimitador ACK</i> es siempre recesivo. |
| EOF | | Fin de la trama: Secuencia de 7 bits recesivos. |
| INT | | Espacio inter-trama: Secuencia de al menos 3 bits recesivos. Se utiliza para generar espacio entre tramas consecutivas. Después de la intermisión el bus permanece en estado recesivo (bus idle) hasta el próximo SOF. |

Tabla 3. Descripción de la trama CAN 2A estándar

1.6.2 CAN 2B – Extendida

En este formato y a diferencia con CAN 2A, el ID es de 29 bits lo que hace posible 536'870.912 de identificadores de mensajes diferentes. Este formato se basa en CAN 2A y prácticamente es lo mismo; por lo tanto las diferencias pueden ser vistas comparando la Tabla 3 con la Tabla 4. El módulo CAN del PIC 18F258 soporta las dos versiones de tramas.

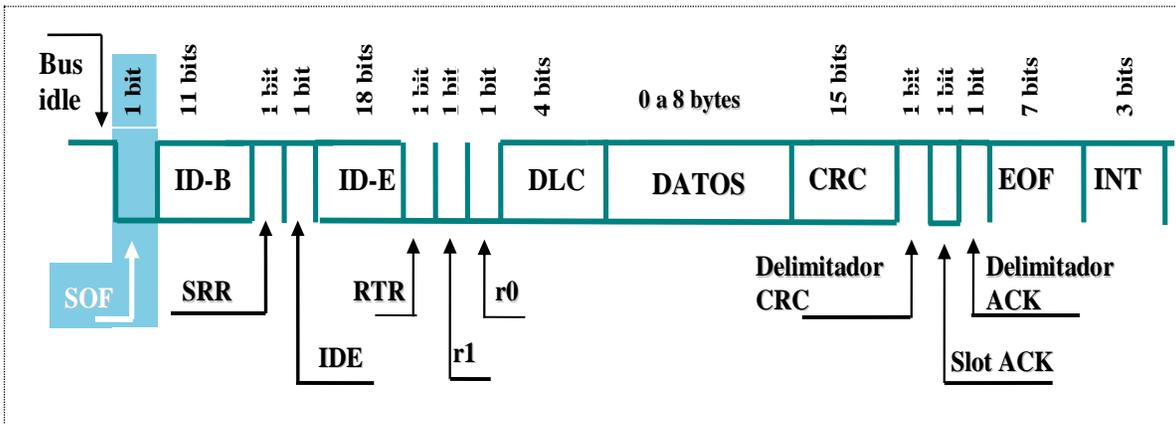


Figura 6. Trama CAN 2B Extendida

| Siglas | Descripción |
|---------------------------|---|
| SOF | Inicio de trama: Se utiliza para la sincronización entre nodos. |
| Campo de arbitraje | Identificador de mensaje: 29 bits para este tipo de trama. Establece un límite en el número de mensajes diferentes. Teóricamente $2^{29} = 536'870.912$ mensajes posibles. Está dividido en dos campos. <ul style="list-style-type: none"> ID-B → Contiene los 11 bits más significativos – MSb ID-E → Contiene los 18 bits menos significativos - LSb |
| | Solicitud remota sustituta: Bit de nivel recesivo. En caso de múltiple acceso entre tramas CAN 2A y CAN 2B, con el mismo identificador base (ID-B), tiene prioridad CAN 2A. |
| | Identificador extendido: Identifica la versión del protocolo: <p style="text-align: center;">Dominante → Versión CAN 2A -</p> |

| Siglas | | Descripción |
|----------------|--|--|
| | | Estándar Recesivo → Versión CAN 2B - Extendido |
| RTR | | Igual que en CAN 2A. |
| r0 y r1 | | Bits reservados: Siempre dominantes. Reservados para futuras versiones. |
| | | Nota: Los demás campos son iguales en CAN 2A. |

Tabla 4. Descripción de la trama CAN 2B extendida

1.6.3 Tipos de Trama

Los siguientes son los tipos de trama en el protocolo CAN.

1.6.3.1 Trama de Datos

Es la trama que el nodo con normalidad utiliza para transmitir datos a través del bus. El bit RTR debe ser dominante.

1.6.3.2 Trama Remota

Se utiliza cuando un nodo le solicita a otro que le envíe información dependiente del identificador del mensaje. El nodo que recibe la trama remota le envía los datos solicitados por medio de una trama de datos. Existen dos grandes diferencias entre una trama remota y una de datos. En la trama remota:

1. El bit RTR es recesivo.
2. No existe el campo de datos.

1.6.3.3 Trama de Error

Una trama de error es generada por cualquier nodo que detecte un error en el bus para avisarle a los restantes del suceso. Esta trama consta de dos campos: un campo de banderas de errores seguido de un campo delimitador de error. Existen dos tipos de campos de banderas de errores. El tipo que se envía depende del estado del nodo que detecta el error. Estos tipos son:

- 1. Errores Activos:** Si un “nodo activo” detecta un error en el bus, el nodo interrumpe la transmisión del mensaje y genera un “campo de bandera de error activo” que se compone de seis bits dominantes consecutivos. Esta secuencia viola la regla de “*bit stuffing*”. Los otros nodos reconocen la violación y generan tramas de error, a lo que se le llama “banderas de eco del error” que consiste de seis bits recesivos (generalmente). A esto le sigue “el campo delimitador de error” compuesto por ocho bit recesivos. Una vez se complete la trama de error, el bus retorna a su estado normal y el nodo interrumpido intenta reenviar el mensaje abortado.
- 2. Errores Pasivos:** Si un “nodo pasivo” detecta un error en el bus, el nodo transmite un “campo de banderas de error pasivo” que consta de seis bits recesivos consecutivos seguido de un “campo delimitador de error” compuesto por 8 bits recesivos. A diferencia del anterior, el mensaje seguirá siendo transmitido, porque este tipo de error no interfiere con el bus, siempre que un “nodo activo” ó el nodo que está transmitiendo el mensaje detecte el error. Si el nodo que transmite genera el “campo de banderas de error pasivo”, causará que los otros nodos generen una trama de error debido a la violación del “*bit stuffing*”.

1.6.3.4 Trama de Sobrecarga

Este tipo de trama tiene el mismo formato de la trama de error activo; la diferencia consiste en que la trama de sobrecarga sólo se genera durante el campo inter-tramas (INT). Esta trama se compone de dos campos:

1. El “campo de banderas de sobrecarga” compuesto por 6 bits dominantes consecutivos más las “banderas de sobrecarga” generadas por otros nodos para un máximo de 12 bits.
2. El “delimitador de sobrecarga” que consiste de 8 bits recesivos consecutivos.

Esta trama puede ser generada por algún nodo como resultado de dos condiciones:

1. El nodo detecta un bit dominante durante el campo INT, lo que viola la forma de la trama CAN (los bits del campo INT deben ser siempre recesivos). Existe una excepción: cuando el bit dominante es detectado durante el tercer bit del campo INT, el receptor lo interpretará como un “inicio de trama”, SOF.

2. Debido a una condición interna puede ser que el nodo no sea capaz de iniciar una recepción de un mensaje. En este caso el nodo debe generar un máximo de 2 tramas de sobrecarga secuenciales para retrasar el inicio del mensaje por leer.

1.7 CÓMO FUNCIONA CAN?

1.7.1 Método de Arbitraje

El método de arbitraje consiste en definir qué nodo sigue transmitiendo y cuál no cuando ocurra una colisión. Para esto utiliza el campo de arbitraje conformado por 12 bits: el Identificador del mensaje (ID → 11 bits) y el bit de petición de transmisión remota (RTR). Existen dos niveles lógicos CAN, uno dominante (bit 0) y el otro recesivo (bit 1). El arbitraje consiste en monitorear el bus físico cuando se está transmitiendo el campo de arbitraje. Si se presenta una colisión el bus se comporta como una compuerta AND lógica; por lo tanto un bit dominante siempre ganará sobre un bit recesivo; en otras palabras el nodo que tenga asignado el menor número en el campo ID tendrá prioridad para transmitir sobre los demás en caso de presentarse una colisión; los otros nodos pasan a modo escucha. A continuación se explicará un ejemplo:

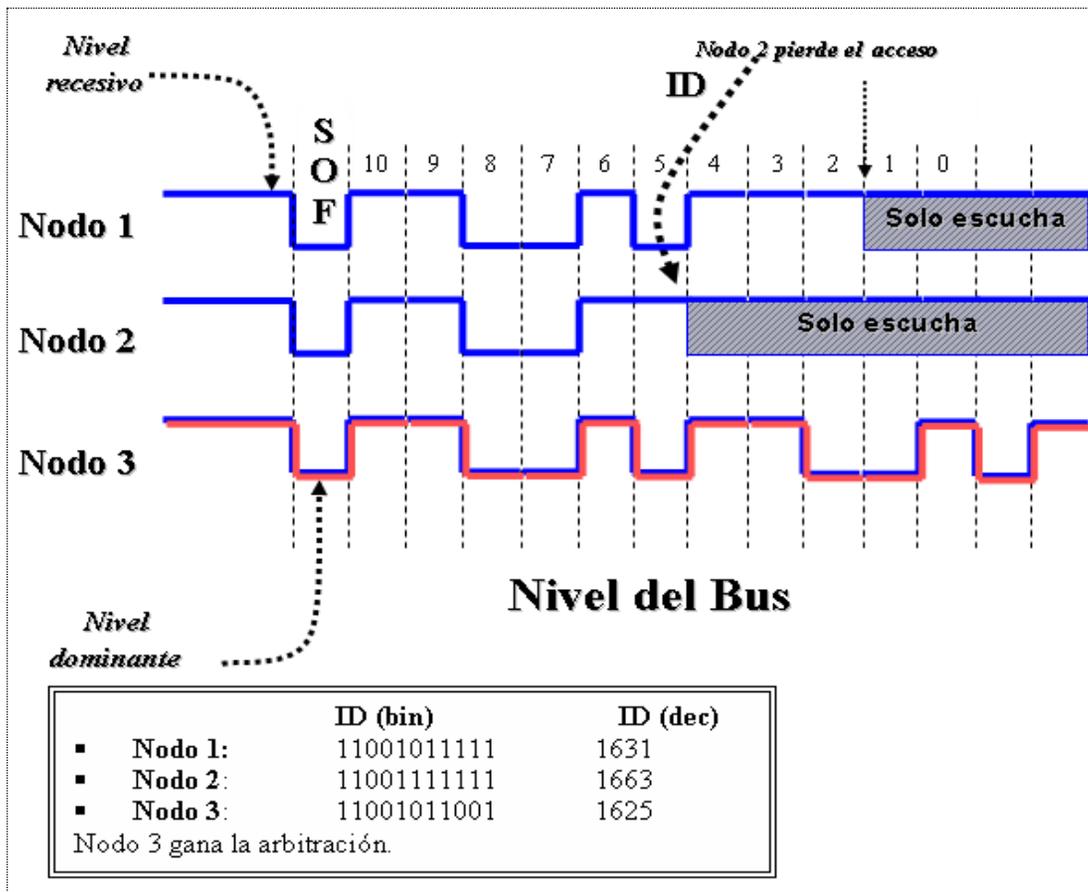


Figura 7. Ejemplo del método de arbitraje

De la figura anterior, hasta el bit 5 del ID los tres nodos envían la misma información, pero en ese instante el nodo 2 para de transmitir (pasa a sólo escucha), pues pierde el arbitraje al ser un bit recesivo mientras que el nodo 1 y 3 envían bits dominantes. Lo mismo sucede con el nodo 1 en el bit 2. Por lo tanto quien seguirá transmitiendo será el nodo 3. En el caso de presentarse una colisión entre una trama remota y una trama de datos con el mismo identificador, la trama de datos ganará el arbitraje debido al bit RTR dominante de la misma; con esto se garantiza que los nodos que solicitan datos con tramas remotas reciban los datos pedidos inmediatamente.

1.7.2 Comunicación Basada en Mensaje

Es una diferencia entre CAN y otras redes que utilizan un protocolo dirigido al mensaje y no a la dirección de los nodos destino; en otras palabras, cuando un nodo transmite un mensaje no lo hace para que lo reciba un nodo destino en especial, sino que todos los nodos reciben el mensaje; éstos por medio de unos filtros de aceptación admiten o no el

mensaje. Por ejemplo, si el nodo A tiene conectada una válvula analógica para controlar el flujo en un tanque, y el nodo maestro es quien posee la información del porcentaje de apertura de dicha válvula el procedimiento de comunicación sería el siguiente:

1. El nodo A envía una trama remota con el ID de la válvula.
2. El nodo maestro recibe la trama remota, que pasa por los filtros de aceptación.
3. El nodo maestro identifica el ID y envía un mensaje no remoto con el valor del porcentaje de apertura de la válvula en el campo de datos al nodo A.
4. El nodo A recibe la trama no remota, que pasa por los filtros de aceptación y es utilizada de manera conveniente a voluntad del diseñador de la red.

1.8 Tipos de Errores

Existen seis tipos de errores en el protocolo CAN:

- Error de Bit.
- Error de Relleno.
- Error de CRC.
- Error de Forma.
- Error de ACK.
- Error en EOF.

1.8.1 Error de Bit

Todo nodo que transmite un bit, al mismo tiempo “escucha” el nivel del bus. Un error de bit se produce cuando el nivel lógico del bit monitoreado es distinto del nivel lógico enviado.

Excepción: Una excepción es el envío de un bit recesivo durante el ID o durante el campo de ACK.

1.8.2 Error de Relleno

Un nodo detecta un error de relleno al recibir seis bits consecutivos de igual nivel. Esto es conocido como “*bit stuffing*”.

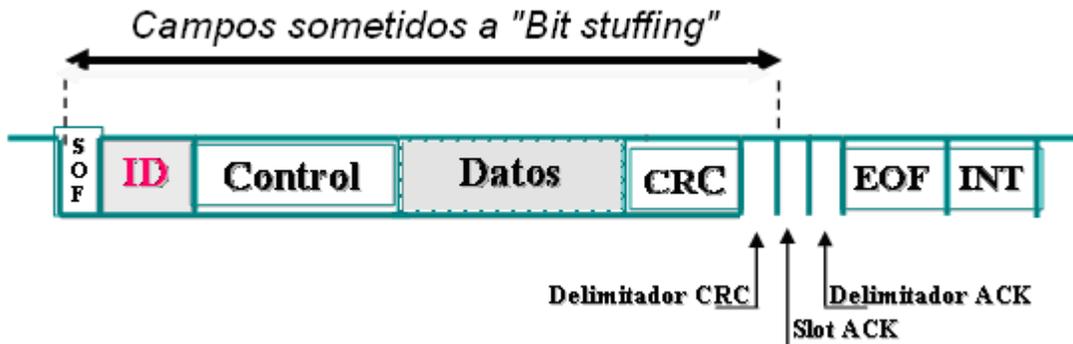


Figura 8. Campos sometidos a "bit stuffing"

1.8.3 Error de CRC

El CRC es el chequeo de redundancia cíclico, un método de chequeo de errores; para ello se definen:

- R → Secuencia de comprobación (n bits).
- M → Secuencia de datos a proteger (k bits)
- G → Polinomio generador (n+1 bits)

Cada una de esas constantes según la Figura 9 es:

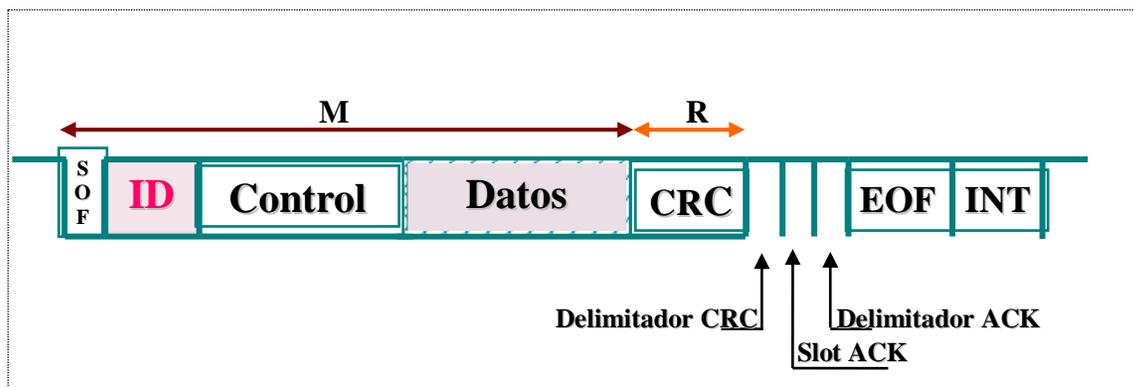


Figura 9. Constantes para el método de CRC

Así el nodo transmisor realizará los siguientes cálculos:

$$\text{Desplazamiento} \rightarrow M \times 2^n$$

$$\text{Obtención del resto} \rightarrow \frac{M \times 2^n}{G} = Q + \frac{R}{G}$$

$$\text{Composición} \rightarrow T = M \times 2^n + R$$

1.8.4 Error de Forma

Un error de forma es detectado cuando el campo de formato fijo contiene uno o más bits ilegales. Por ejemplo se presenta este tipo de error si el bit delimitador CRC, siempre recesivo, es recibido como dominante.

1.8.5 Error de ACK

Un transmisor que envía un mensaje (de datos o remoto) emite el bit ACK "slot", con un nivel recesivo. En caso de monitorear un nivel recesivo, un error de ACK se genera. En caso de monitorear un nivel dominante, el transmisor entiende que al menos un receptor recibió el mensaje correctamente.

1.8.6 Error en EOF

Si una trama ha sido recibida sin errores hasta el bit 7 del campo EOF, el mensaje se tomará como válido. Si hasta en ese bit se ha detectado algún error entonces se generará una trama de error en el primer bit del campo INT. Si son los receptores quienes detectan el error en el bit 7 se generará una trama de sobrecarga.

2 DESCRIPCIÓN DE LOS PROCESOS POR INTEGRAR EN LA RED

2.1 INTRODUCCIÓN

Para llevar a cabo la implementación de la red industrial CAN, se hace uso de dos de las plantas del laboratorio de control de procesos del programa de Ingeniería en Automática Industrial; su elección se basó en los componentes confiables, eficientes y seguros, tales como sensores y actuadores, que conforman las mismas. Los procesos por integrar son la planta de temperatura FEEDBACK PT326 y la planta de nivel y flujo de agua.

La caracterización de los procesos y la sintonización de los controladores PID correspondientes a cada planta de los proyectos *Servidor iFIX para Aplicaciones Distribuidas en Automatización Industrial*, *Servidor Wonderware Intouch para el Control de Procesos Distribuidos* e *Implementación de una Red Industrial CAN bajo un Sistema SCADA* deben ser iguales, ya que controlan los mismos procesos y sus resultados son enmarcados dentro del proyecto de investigación “*Convergencia de las Redes Industriales sobre sistemas SCADA*”

2.2 PLANTA DE NIVEL Y FLUJO DE AGUA

2.2.1 Descripción y Funcionamiento de la Planta de Nivel y Flujo de Agua

La planta de nivel y flujo de agua (Véase Figura 10) es un sistema compuesto por un circuito hidráulico y la instrumentación necesaria para efectuar el control del flujo del agua por el circuito o el control del nivel de agua en uno de los tanques del circuito.

El sistema consta de: un tanque plástico grande para el almacenamiento del agua, una bomba centrífuga, tubería PVC, tres tanques plásticos pequeños, válvulas de bola de accionamiento manual, una válvula de control de flujo, un sensor del flujo de agua con su respectivo transmisor de señal, un transmisor de nivel implementado por medio de un

transmisor de presión diferencial que entrega una señal eléctrica de 4 – 20 mA proporcional al nivel en el tanque (0 – 30 cm).

Adicionalmente se tiene un “*switch*” que aplica alimentación de 120 Vac al sistema, y otro “*switch*” para el encendido y apagado de la bomba. En la parte delantera se tiene acceso a las señales de entrada y salida de la instrumentación.

El funcionamiento de la planta hace circular el agua del tanque de almacenamiento hacia los tanques pequeños y de vuelta al tanque de almacenamiento, bajo el impulso de la bomba y la gravedad terrestre. Se puede hacer que el agua circule por uno o más tanques dependiendo de qué válvulas manuales estén abiertas o cerradas. De esta manera es posible configurar procesos de control de nivel de primero o segundo orden, según el número de tanques que intervengan en el proceso.



Figura 10. Planta de nivel y flujo de agua

El diagrama de instrumentos según el estándar SP5 de ISA de la planta de nivel y flujo de agua se muestra a continuación:

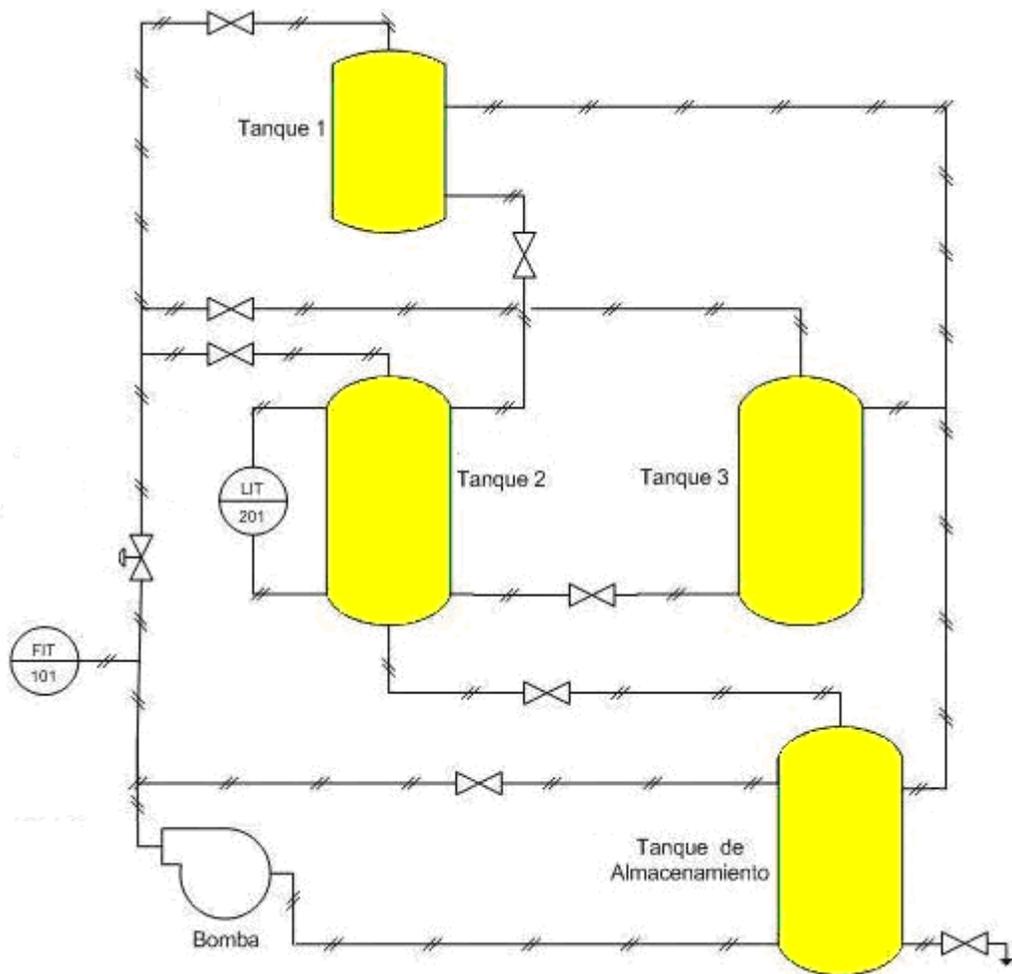


Figura 11. Diagrama de instrumentos según el estándar SP5 de ISA de la planta de nivel y flujo de agua

En la Figura 12 se puede observar la configuración que se usará en el desarrollo del proyecto.

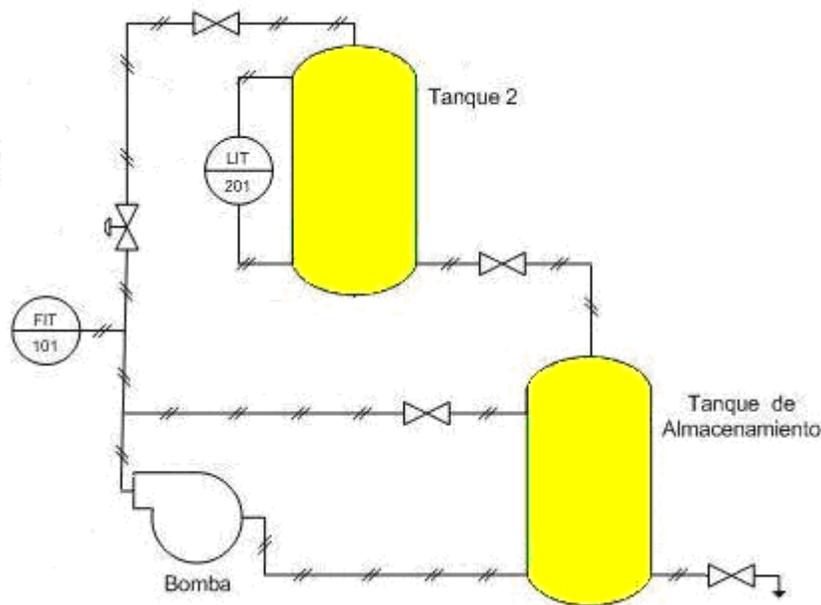


Figura 12. Diagrama ISA de la configuración usada

2.2.1.1 Descripción de los Componentes Planta de Nivel y Flujo de Agua

A continuación se describe de manera breve cada uno de los componentes que hacen parte de la planta de nivel y flujo de agua.

2.2.1.1.1 Transmisor de Presión Diferencial Yokogawa EJA110A

Haciendo uso del transmisor Yokogawa EJA110A (Ver Figura 13), es posible conocer el nivel del líquido, es decir la variable controlada, de la planta de nivel.

El transmisor puede ser usado para sensar líquidos, gases o flujo de vapor, al igual que el nivel de un líquido, densidad y presión. La salida del transmisor es una señal correspondiente a la medida de presión diferencial de 4 - 20 mA DC [1].



Figura 13. Transmisor de presión diferencial Yokogawa EJA110A

2.2.1.1.2 Transmisor de Flujo GF + SIGNET 8550-1

Permite controlar y monitorear el flujo de agua a través de la planta. Es un instrumento avanzado que convierte la variable física de flujo a una señal eléctrica de 4 - 20 mA para largas distancias de transmisión [2] (Ver Figura 14).



Figura 14. Transmisor de flujo GF + SIGNET

2.2.1.1.3 Válvula DWYER Serie ABV

Es el actuador de la planta (Ver Figura 15); realiza una acción sobre la variable manipulada (en este caso flujo de agua), dependiendo de la señal de control que reciba del controlador. La válvula de control es usada para regular continuamente el flujo de un

fluido (agua en este caso) en el sistema, cambiando la cantidad de flujo que pasa a través de la válvula para así controlar el proceso. El cambio en el flujo puede ser usado para controlar temperatura, flujo, presión, pH, y en este caso, nivel [3].

La señal de entrada de la válvula es una señal eléctrica de 4 - 20 mA, que permite así un rango de apertura de 0 a 100% correspondiente a dicha señal.



Figura 15. Válvula DWYER serie ABV

2.2.1.1.4 Bomba Centrífuga de Agua IHM.S.A Modelo 1B-1/4 Serie 0000115

Se caracteriza por su bajo consumo de energía, mayor vida útil, su bajo peso que facilita el montaje, una carcasa tipo caracol y un motor diseñado para trabajo continuo (Ver Figura 16). Su función es impulsar el agua almacenada en el tanque de suministro hacia los otros tanques que conforman la planta.



Figura 16. Bomba centrífuga

2.2.2 Entradas/Salidas de la Planta de Nivel y Flujo de Agua

| Elemento | Señal | Tipo I/O |
|---|----------------------|----------|
| Transmisor de nivel Yokogawa EJA 110 A | Corriente 4-20 mA | Entrada |
| Transmisor de flujo GF+SIGNET 8550-1 | Corriente 4-20 mA | Entrada |
| Válvula DWYER serie ABV | Corriente 4 – 20 mA. | Salida |

Tabla 5. Componentes y Entradas/Salidas de la planta de nivel y flujo de agua

2.3 PLANTA DE TEMPERATURA (MÓDULO PT326)

2.3.1 Descripción y Funcionamiento del Módulo PT326

El módulo PT326 (Véase Figura 17) consta de un tubo de conducción de aire con tres agujeros, una malla de calefacción eléctrica, un sensor de temperatura (termistor no lineal), y un ventilador de 17.000 RPM de 12V [4].

El funcionamiento del módulo succiona aire desde la atmósfera a través de una entrada ajustable; dicho aire es calentado gracias a la malla de calefacción eléctrica, posteriormente es enviado hacia la atmósfera por medio del tubo de conducción. El proceso consiste en mantener constante la temperatura del aire que fluye a través del tubo de conducción, gracias a la potencia aplicada a la malla de calefacción eléctrica, suministrada por un controlador.

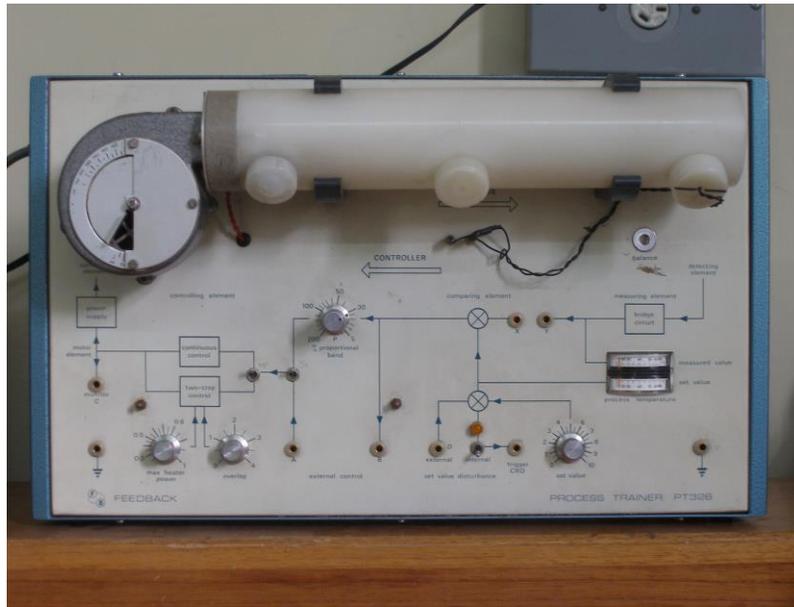


Figura 17. Módulo PT326

El esquema del módulo se puede observar más detalladamente en la Figura 18:

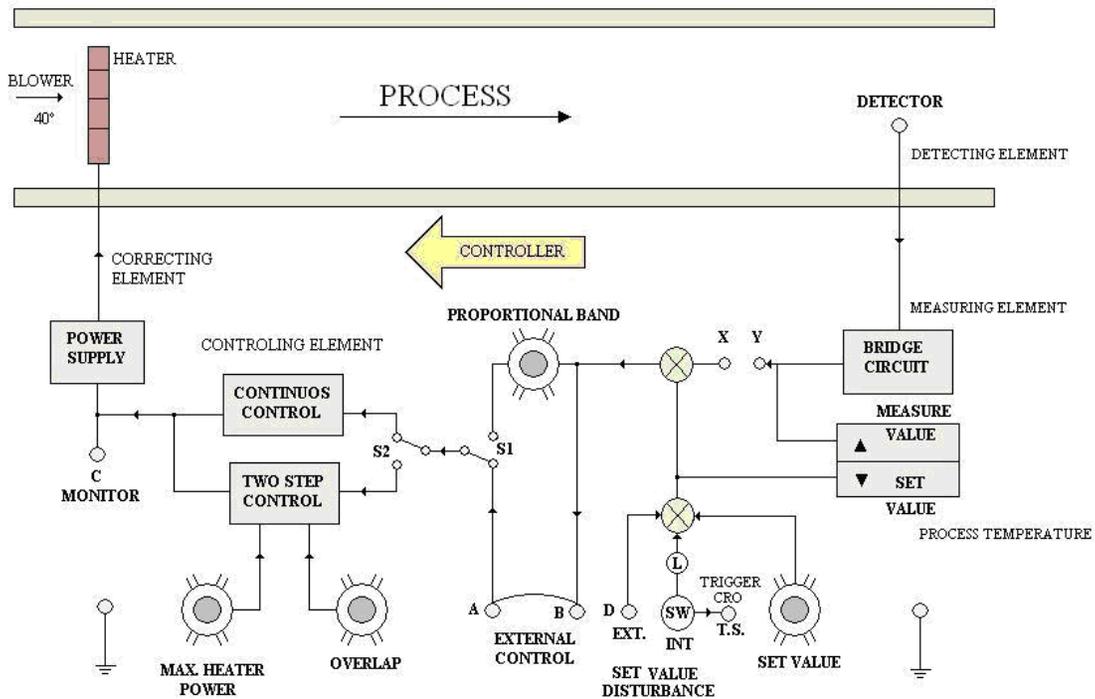


Figura 18. Esquema detallado del módulo PT326

3 DESCRIPCIÓN DE LOS DISPOSITIVOS ELECTRÓNICOS EMPLEADOS PARA EL DESARROLLO DE LA INTERFAZ CAN

3.1 INTRODUCCIÓN

Para la implementación de la red industrial CAN es necesario que los dispositivos que componen los procesos industriales por integrar, ya sean sensores, actuadores o controladores, estén en capacidad de intercambiar información a través del protocolo CAN. Para este fin es necesario construir las interfaces que permitan dicho objetivo. Actualmente en el mercado existen muchos dispositivos electrónicos, de diferentes marcas y características técnicas, capaces de realizar la conexión para lograr esta interoperabilidad.

Después de realizar el análisis y estudio detallado acerca de los elementos disponibles en el mercado y que cumplen con los requerimientos del proyecto, en cuanto al beneficio técnico y aspectos económicos, se llegó a la conclusión de que existían dos arquitecturas en dispositivos de la empresa Microchip, que satisfacen dichas características en la implementación de la red.

3.2 DESCRIPCIÓN DE LAS POSIBLES ARQUITECTURAS DE LA RED INDUSTRIAL

Cada una de las dos arquitecturas provistas por Microchip para la implementación de la red industrial tiene sus beneficios y sus limitaciones. Ellas se diferencian básicamente en el hardware que las compone y en las características operacionales y funcionales de dicho hardware.

3.2.1 Arquitectura con el Controlador CAN Externo

Esta arquitectura se compone de un PLC Micrologix 1500, un PIC 16FXXX, un controlador CAN externo (MCP2515) y el transductor CAN (MCP2551). El PIC utilizado en esta arquitectura no tiene el módulo CAN incluido internamente, es decir, no tiene implementadas las funciones CAN necesarias para el intercambio de información por medio del protocolo. Por esta razón es necesario el uso del controlador CAN externo. Este dispositivo soporta una serie de funciones, que permiten la recepción, transmisión y detección de errores de los mensajes CAN que son enviados a través de la red. El controlador MCP2515 se programa vía SPI, haciendo uso del PIC 16FXXX.

El transductor CAN es indispensable e irremplazable en las dos arquitecturas de solución que maneja Microchip. La arquitectura de la red es mostrada en la Figura 21.

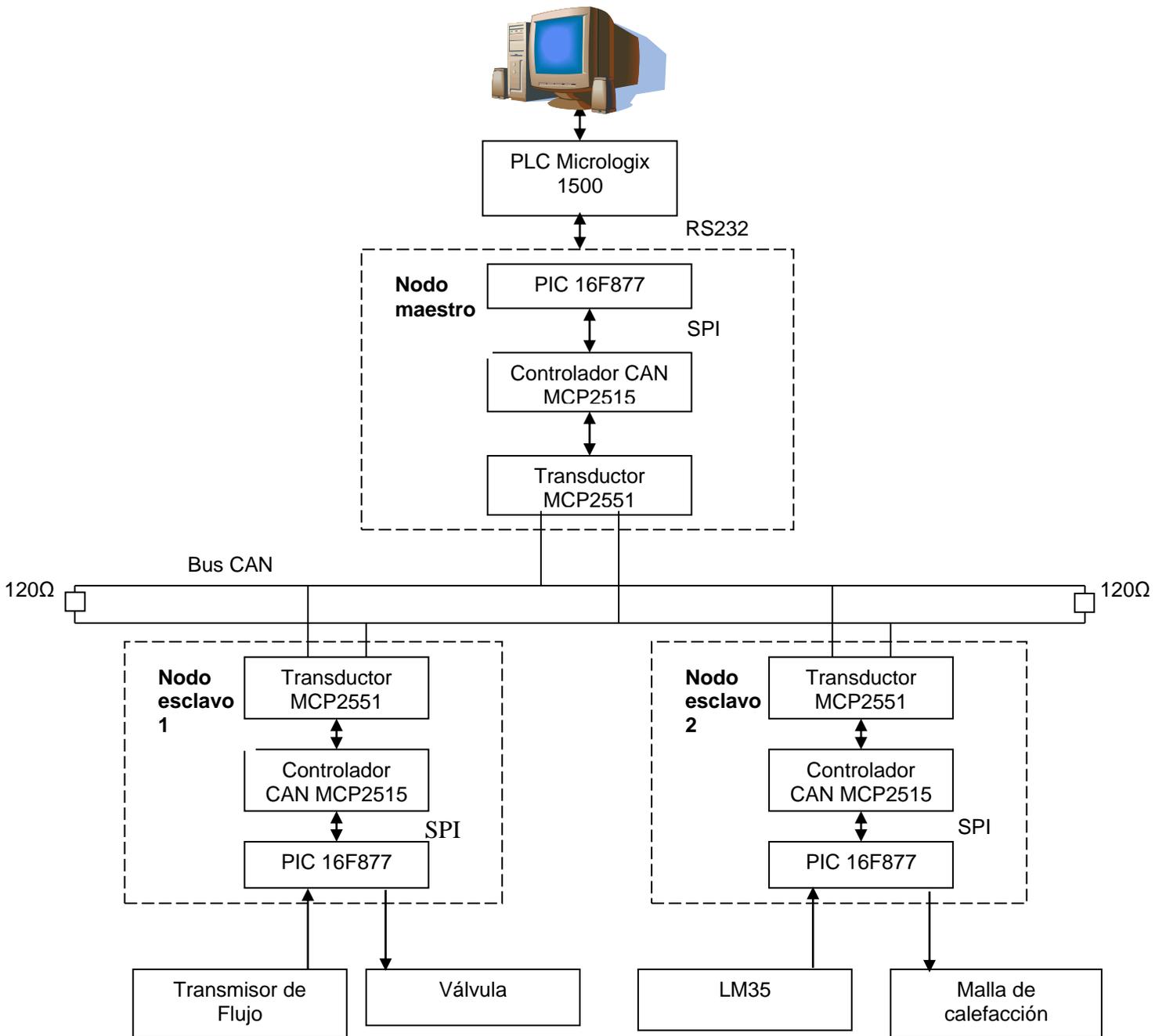


Figura 21. Arquitectura de la red con el controlador CAN externo MCP2515

3.2.2 Arquitectura con el Controlador CAN Incluido Internamente en el PIC

Esta arquitectura se compone de los siguientes elementos: un PLC Micrologix 1500, un PIC18FXX8 y el transductor MCP2551.

Como se puede observar en la Figura 22 la arquitectura no hace uso del controlador externo MCP2515, ya que el PIC tiene incorporado el módulo CAN, encargado de establecer las comunicaciones con el protocolo CAN. Este módulo soporta más funciones que las incluidas en el controlador externo MCP2515. Además tiene otra serie de características adicionales como lo son el número de máscaras y filtros que pueden ser usados.

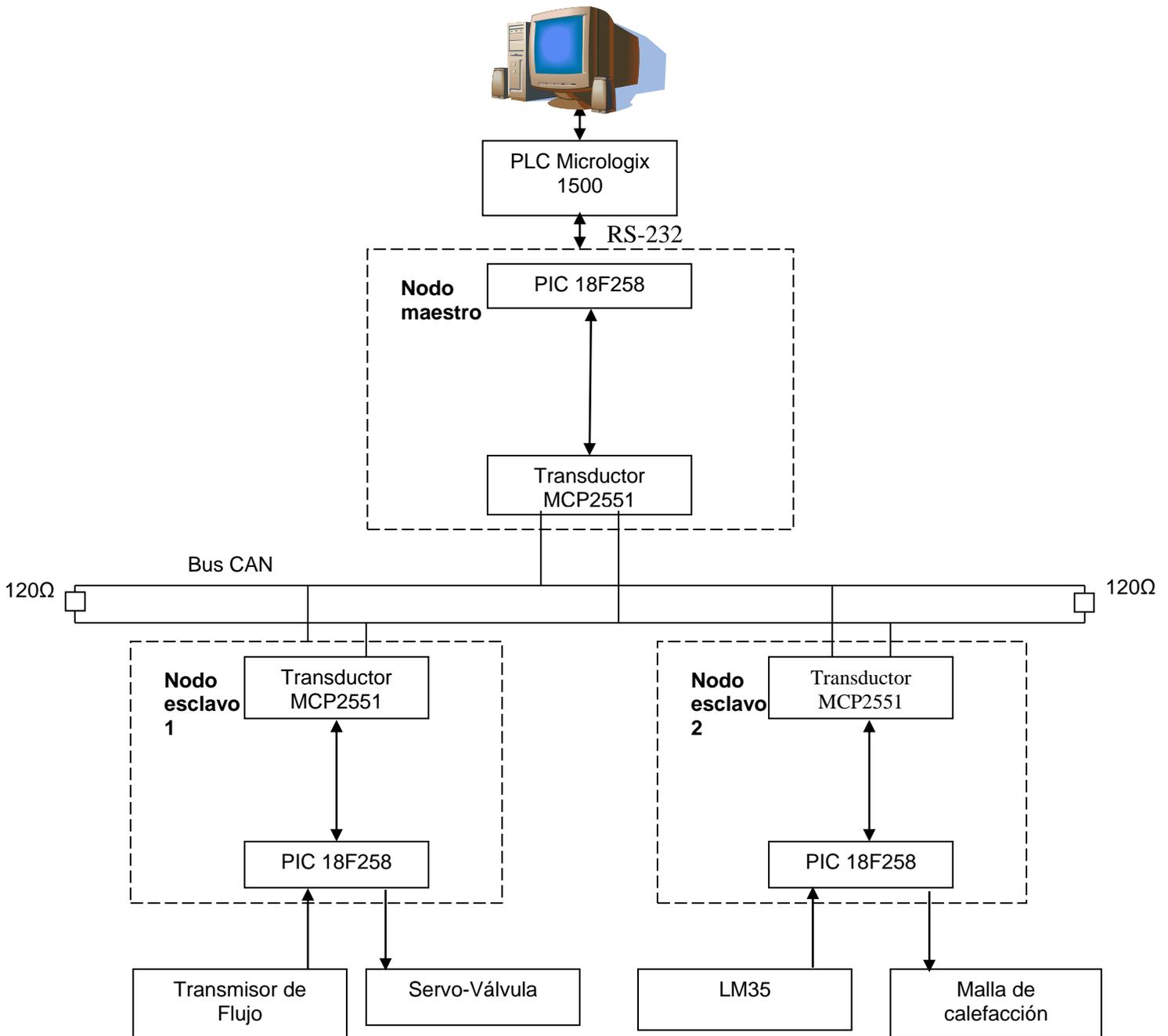


Figura 22. Arquitectura de la red con el PIC18F258

3.2.3 Comparación de las Arquitecturas

En la Tabla 7 se resumen las principales características funcionales y operacionales de cada una de las arquitecturas de la red descritas anteriormente.

| Característica por Nodo | Implementación sin CAN (PIC16FXXX) | Implementación con CAN (PIC18FXX8) |
|-------------------------------------|--|--|
| Precios | PIC 16F877 = US. 5.11 MCP2515 = US. 1.72 MCP2551 = US. 0.83 Total = US.7.66 | PIC 18F258 = US. 7.11 MCP2551 = US. 0.83 Total = US. 7.94 |
| ADC | 5 ADCs de 8/10 bits | 5 ADCs de 10/11 bits |
| Versión de CAN | CAN 2.0B Active | CAN 2.0B Active |
| "buffer" de Recepción y Transmisión | <ul style="list-style-type: none"> • 2 "buffers" de Rx. • 3 "buffers" de Tx. | <ul style="list-style-type: none"> • 2 "buffers" de Rx. • 3 "buffers" de Tx. • 6 "buffers" de Rx/Tx. |
| Máscaras y Filtros | <ul style="list-style-type: none"> • 2 Máscaras 29-bits. • 6 Filtros 29-bits. | <ul style="list-style-type: none"> • 3 Máscaras 29-bits. • 16 Filtros 29-bits. |
| PWM. | <ul style="list-style-type: none"> • CCP1 • CCP2 | <ul style="list-style-type: none"> • CCP1 • ECCP1 |
| Tipo de Tramas | <ul style="list-style-type: none"> • Identificador estándar de 11-bits • Identificador extendido de 29-bits | <ul style="list-style-type: none"> • Identificador estándar de 11-bits • Identificador extendido de 29-bits |
| Modos de Operación | <ul style="list-style-type: none"> • Modo normal. • Modo sólo escucha. • Modo "loopback". • Modo "sleep". • Modo Configuración. | <ul style="list-style-type: none"> • Modo normal. • Modo sólo escucha. • Modo "loopback". • Modo "sleep" / Deshabilitado. • Modo Configuración. |
| Comunicación | El MCP 2515 se comunica con el PIC a través de SPI, por medio de este canal se recibe, se transmite y se configura los registros. | En este caso, no existe comunicación SPI, ya que, el microcontrolador incorpora el protocolo CAN. |

Tabla 7. Comparación de las arquitecturas

3.2.4 Ventajas y Desventajas de cada una de las Arquitecturas

Las principales ventajas de la arquitectura con tecnología CAN incluida corresponden a que el PIC que sirve como controlador CAN posee 6 “*buffers*” adicionales de Rx/Tx, más máscaras y filtros de aceptación, lo cual es fundamental para la expansión del sistema, permitiendo fácilmente la adición de más nodos a la red.

Dadas las características anteriores, el proyecto se desarrolla haciendo uso de la segunda arquitectura, implementándose la red por medio del controlador CAN incluido internamente en el PIC.

3.3 CARACTERÍSTICAS Y MODOS DE OPERACIÓN DE LOS DISPOSITIVOS ELECTRÓNICOS USADOS EN LA RED

A continuación se mencionan las principales características de los dispositivos que son usados en el montaje de la red.

3.3.1 PIC 18F258

El PIC18F258 tiene memoria de programa FLASH y memoria de datos RAM (32 Kbytes y 1536 bytes respectivamente), tiene implementado 5 canales A/D, 3 puertos I/O y módulos de comunicación serial [6].

El módulo “*Controller Area Network*” (CAN) con el que cuenta el PIC es una interfaz serial, muy útil para comunicarse con otros dispositivos periféricos u otros microcontroladores. Este módulo soporta tanto las versiones CAN 2A y CAN 2B del protocolo CAN especificado por Bosch.

El módulo es el encargado de ejecutar todas las funciones de recepción y transmisión de mensajes en el bus CAN, en el cual los mensajes son transmitidos cargando los registros apropiados de datos. Los errores y el estado pueden ser revisados leyendo los registros apropiados, y cualquier mensaje detectado en el bus CAN es revisado para ver si tiene errores o no y así compararlo con los filtros de aceptación para ver si puede ser recibido y almacenado en uno de los dos registros de recepción.

El módulo CAN soporta los siguientes tipos de tramas que ya fueron explicadas en el capítulo 1:

- Trama de datos estándar.
- Trama de datos extendida.
- Trama remota.
- Trama de error.
- Trama de recepción de sobrecarga.
- Espacio entre tramas.

El módulo CAN usa los pines RB3/CANRX y RB2/CANTX/INT2 del PIC 18F258 para comunicarse con el bus CAN. Para configurar estos pines como una interfaz CAN se debe:

- Fijar el bit RB3: TRISB<3>.
- Clarear el bit RB2: TRISB<2>.

El PIC 18F258 tiene tres “*buffers*” de transmisión y dos de recepción, dos máscaras de aceptación (una por cada “*buffer*” de recepción) y un total de seis filtros de aceptación.

Existen muchos registros de control y datos asociados al módulo CAN. Para una mejor comprensión se han asociado dentro de los siguientes grupos:

- Registros de control y estado.
- Registros del “*buffer*” de transmisión.
- Registros del “*buffer*” recepción.
- Registros de control del “*baudrate*”.
- Registro de control de I/O.
- Registros de control y estado de las interrupciones.

Para más información acerca del módulo CAN dirigirse a [6] (pagina 201).

El módulo CAN tiene diferentes modos de operación. Cuando se cambian los modos de operación, el modo de operación no cambiará hasta que se completen todas las transmisiones de los mensajes pendientes. Existen seis modos de operación:

- **Modo Configuración:** Para inicializar el módulo CAN el módulo debe estar en el modo configuración. Los registros que manejan la configuración del módulo no pueden ser modificados mientras el módulo CAN esta “*online*” o corriendo. Cuando el módulo está en modo configuración pueden ser modificados los siguientes registros: los registros de configuración donde se modifican los segmentos de tiempo como el “*Baud Rate Prescaler*”, estos segmentos son abordados con más detalle en el capítulo 4; los registros de configuración de los filtros de aceptación y de las máscaras de aceptación.
- **Modo Deshabilitado:** En este modo, el módulo no transmitirá ni recibirá mensajes. Si el módulo recibe una interrupción mientras está en el modo “*Sleep*” algún mensaje podría perderse.
- **Modo Normal:** Este es el modo de operación estándar o por defecto del PIC18F258. En este modo el dispositivo monitorea todos los mensajes en el bus, genera bits de ACK, diferentes tramas, etc. La única manera para que el PIC transmita mensajes por el bus es si se encuentra en este modo.
- **Modo Sólo Escucha:** El modo sólo escucha proporciona al PIC los medios para recibir todos los mensajes, incluyendo los que tienen errores. Estando en este modo es imposible para el PIC transmitir mensajes, pero puede hacer uso de máscaras y filtros para aceptar determinados mensajes en los “*buffers*” de recepción.
- **Modo “Loopback”:** Permite una transmisión interna de mensajes desde los “*buffers*” de transmisión hasta los “*buffers*” de recepción sin transmitirlos por el bus CAN. Este modo es usado para la realización de pruebas en el sistema. El dispositivo recibe los mensajes entrantes mandados por él mismo, como si vinieran de nodos diferentes. Al igual que en el modo sólo escucha se puede hacer uso de máscaras y filtros para aceptar determinados mensajes.
- **Modo Reconocimiento de Error:** Si se fija este modo el módulo no tiene en cuenta los errores, lo cual hace que reciba todos los mensajes ya sean válidos o inválidos.

En cuanto a la transmisión de mensajes, el módulo CAN implementa tres “*buffers*” de transmisión cada uno con una capacidad de 14 registros, donde cada registro ocupa un

byte de SRAM. A continuación se describe en forma general la distribución de los registros, en la cual Microchip designa los nombres de cada registro.

- El primer byte, TXBnCTRL, es un registro de control de los “*buffers*” de transmisión. La información en este registro determina las condiciones bajo las cuales el mensaje será transmitido e indicará el estado de la transmisión.
- Los 5 bytes siguientes son utilizados para almacenar los identificadores estándares y extendidos, y también la información necesaria para el arbitraje.
- Los 8 bytes restantes son utilizados para almacenar los posibles datos del usuario. Se debe recordar que de acuerdo con la trama CAN, este espacio es flexible y depende de la cantidad de datos que requiera enviar el usuario, siendo el máximo posible 8 bytes.

El diagrama de distribución de los “*buffers*” de transmisión del módulo CAN se presenta a continuación:

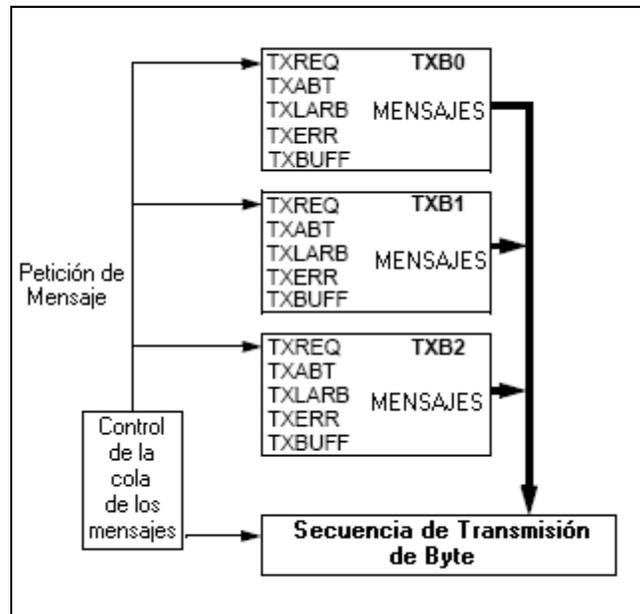


Figura 23. Distribución de los “*buffers*” de transmisión

El PIC 18F258 prioriza la transmisión de los mensajes en los “*buffers*” independiente del manejo de prioridades implícito que posee el protocolo CAN. Esta prioridad se realiza cuando se está construyendo la trama CAN, pero no hace parte de la misma, simplemente es una forma de informar al PIC cuál de los tres “*buffers*” envía primero.

Antes de enviar el inicio de trama (SOF), la prioridad de todos los “*buffers*” que están en cola es comparada, y el que tenga la mayor prioridad será quien envíe primero el mensaje para lo cual se pueden configurar 4 niveles de prioridad.

En el caso de que dos “*buffers*” contengan la misma prioridad, el “*buffer*” con el número mayor de “*buffer*” será enviado. Por ejemplo, Si el “*buffer*” 1 tiene la misma prioridad de enviar que el “*buffer*” 0, quien envíe primero será el “*buffer*” 1.

A continuación se muestra el diagrama de flujo para la transmisión del mensaje:

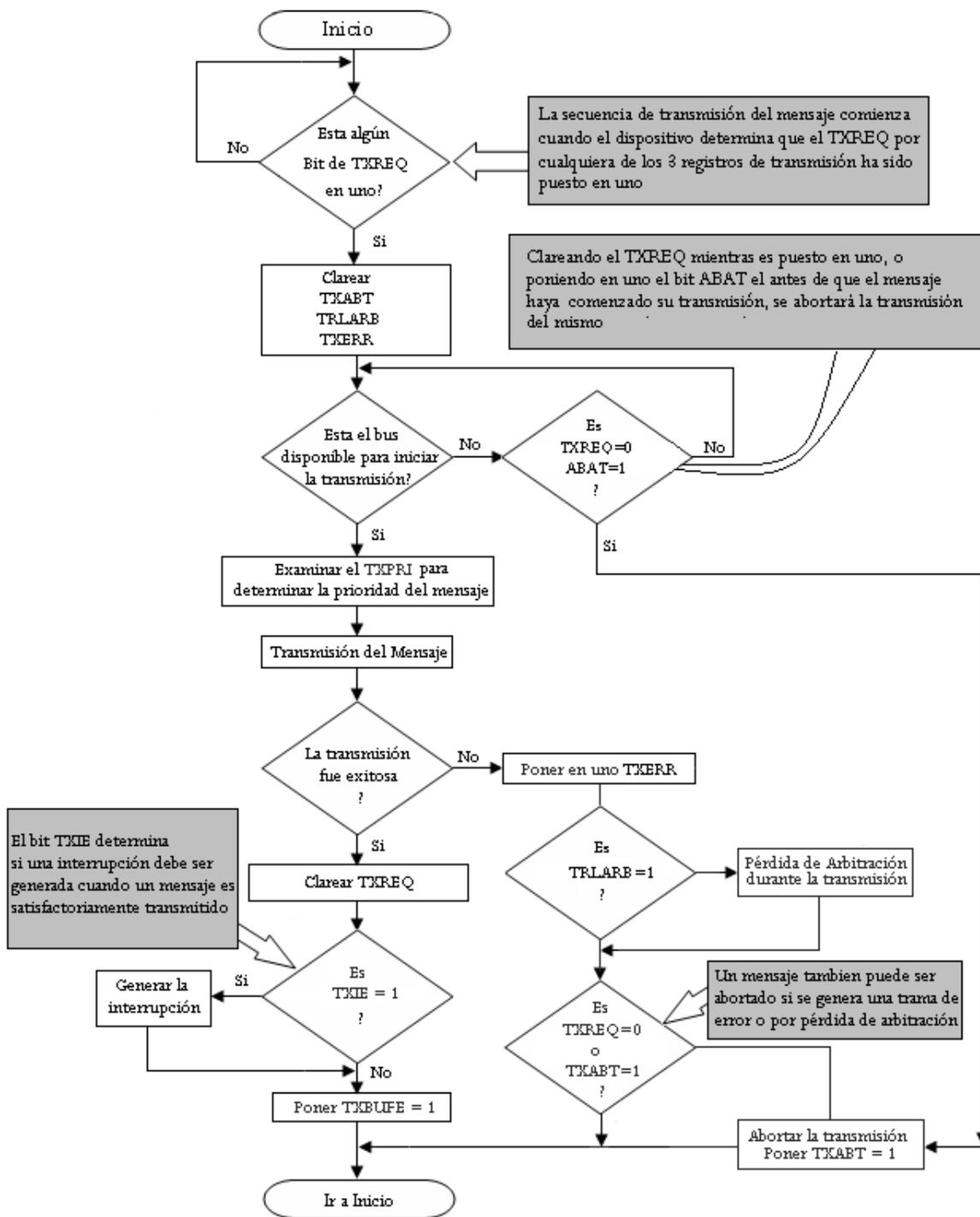


Figura 24. Diagrama de flujo de la transmisión de mensajes

Para la recepción de los mensajes el módulo CAN implementa dos “buffers” de recepción: RXB0 y RXB1, donde cada uno de ellos tiene vinculados múltiples filtros de aceptación y una máscara de aceptación.

- RXB0 tiene asociado una máscara de aceptación y dos filtros de aceptación.
- RXB1 tiene asociado una máscara de aceptación y cuatro filtros de aceptación.

Existe además el MAB (“*Message Assembly Buffer*” – “*Buffer*” de Montaje del Mensaje) que actúa como un tercer “*buffer*”. La distribución de los “*buffers*”, las máscaras y los filtros se muestra en la siguiente figura:

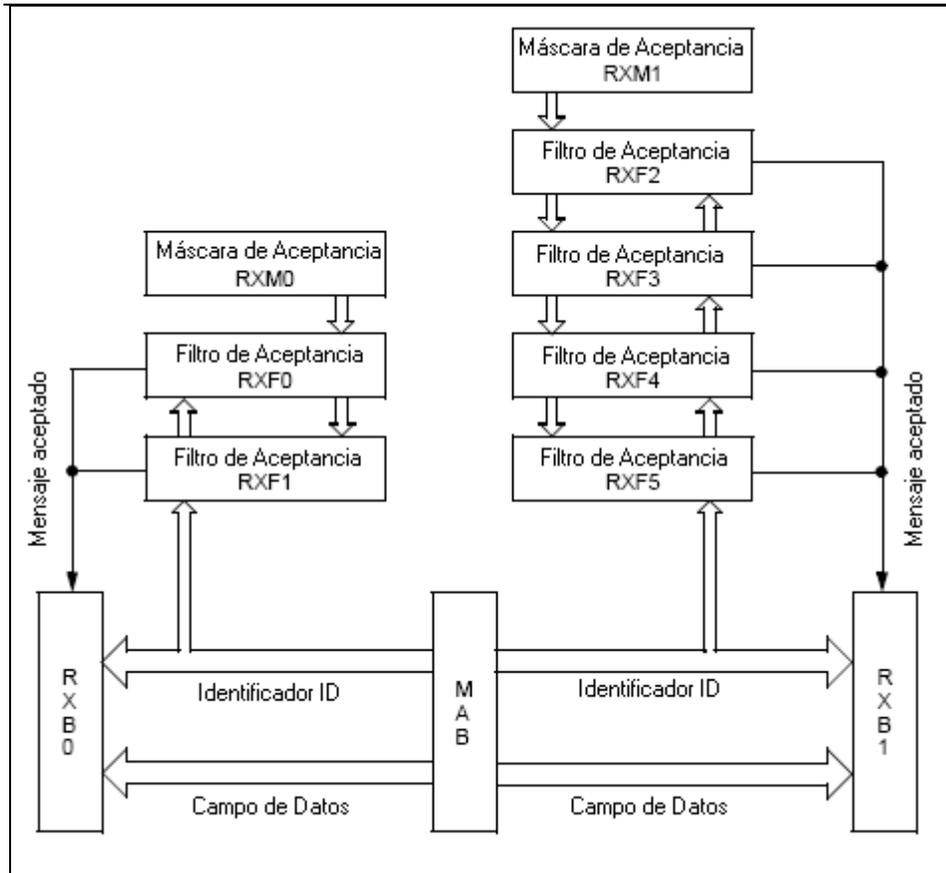


Figura 25. Distribución de “*buffers*” de recepción

El MAB es el “*buffer*” encargado de recibir todos los mensajes del bus, los cuales se transferirán a los “*buffers*” RXB0 ó RXB1 sólo si el identificador es aceptado, es decir, si coincide con alguno de los filtros de aceptación. Existen 2 banderas, RXB0IF y RXB1IF, que le indican al módulo CAN cuál de los dos “*buffers*” recibió el mensaje: RXB0 ó RXB1 respectivamente. Al finalizar la recepción del mensaje automáticamente la bandera del “*buffer*” (RXB0IF o RXB1IF) que recibió el mensaje es clareada. Esto garantiza que el PIC no intente recibir un nuevo mensaje en el “*buffer*” respectivo hasta que no se haya leído completamente el actual.

RXB0 es el “*buffer*” de alta prioridad y RXB1 es el “*buffer*” de baja prioridad. Existe la posibilidad de configurar el módulo de tal forma que si el “*buffer*” RXB0 contiene un mensaje, y un nuevo mensaje es recibido, no se genere una trama de error de sobreflujo, lo cual sería una situación típica, sino que el nuevo mensaje sea almacenado en el “*buffer*” RXB1 siempre y cuando el identificador coincida con alguno de los filtros de aceptación.

En el registro RXBnCON (n = número del “*buffer*”) existen bits para indicarle al PIC cuál de los seis filtros de aceptación fue el que aceptó el mensaje, y si éste es una trama remota o no.

Además se puede realizar la configuración para recibir todos los mensajes válidos; sólo los mensajes con identificador estándar, los mensajes con identificador extendido o todos los mensajes aunque tengan errores, situación que se utilizaría para efectos de depuración de los errores, por medio de la fijación de los bits RXM0 y RXM1 del registro RXBnCON.

A continuación se presenta el diagrama de flujo de la recepción de mensajes:

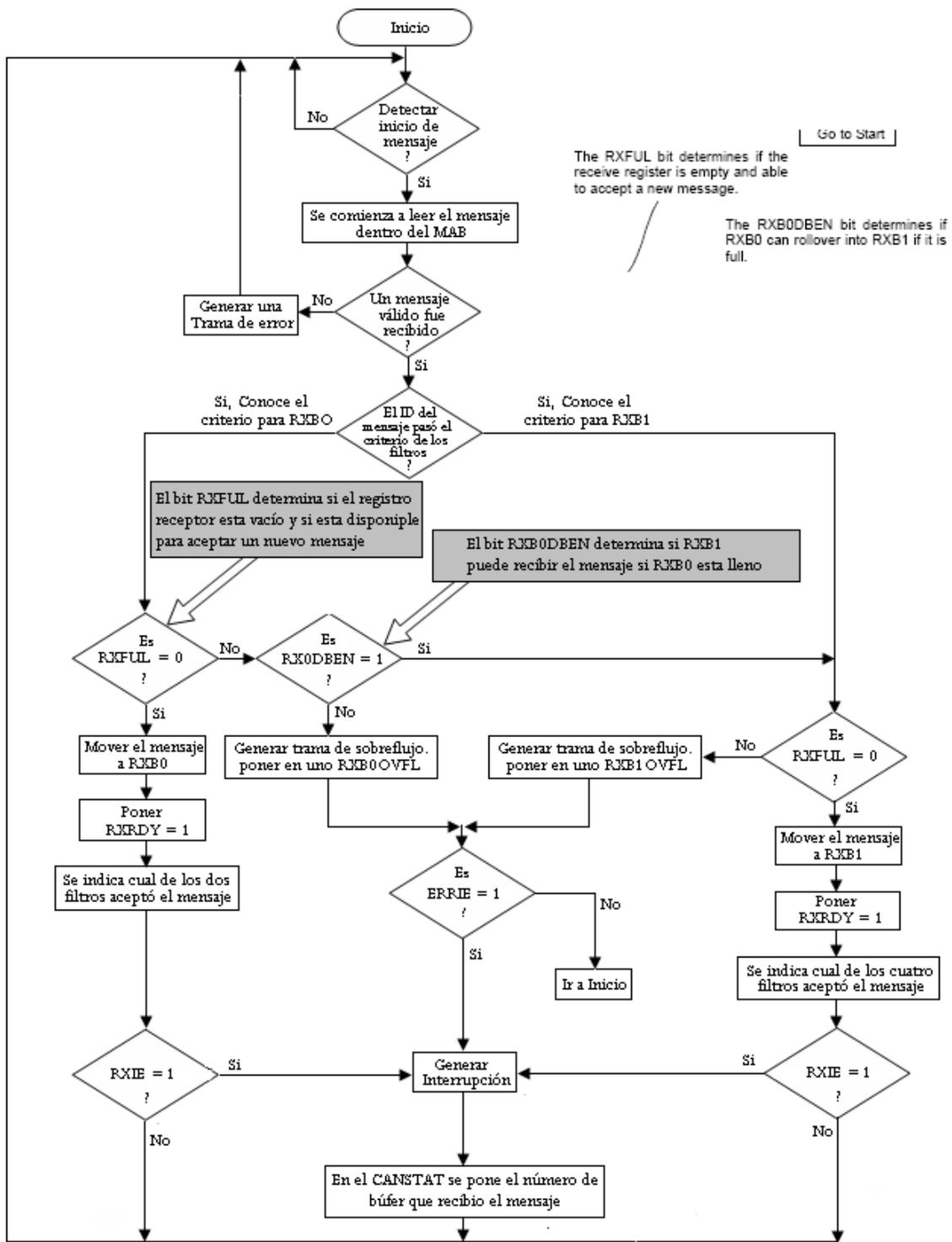


Figura 26. Diagrama de flujo de la recepción de mensajes

3.3.2 Transductor CAN MCP2551

El MCP2551 es un dispositivo CAN de alta velocidad que sirve como la interfaz entre el controlador CAN y el bus físico, con velocidades hasta 1 Mb/s [7].

Cada nodo en la red CAN debe tener un dispositivo para convertir las señales digitales generadas por el controlador CAN (en este caso el PIC18F258) a señales apropiadas para la transmisión a través del bus CAN.

Los procesos de nivel y de temperatura son controlados por un PLC Micrologix1500 [8], el cual tiene las siguientes características generales:

| ELEMENTO | DESCRIPCIÓN |
|------------------------------|---|
| Unidad Base | 1764 – 24AWA Serie C |
| Procesador | 1764 LRP |
| Número de Entradas Discretas | 11 entradas discretas |
| Número de Salidas Discretas | 11 salidas discretas |
| Conexión de Comunicación | Conexión al Puerto RS232 Conexión a una RED DH485 Conexión a DviceNet |

Tabla 8. Características del PLC Micrologix 1500

El PLC cuenta con un módulo analógico 1769-IF4XOF2 [9] cuyas características principales son:

| CARACTERÍSTICA | MÓDULO 1769-IF4XOF2 |
|--------------------------------------|---|
| Número de Entradas | 4 entradas analógicas (4 de voltaje y 4 de corriente) |
| Número de Salidas | 2 salidas analógicas (2 de voltaje y 2 de corriente) |
| Rango de Señal | Voltaje: 0 – 10 Vdc Corriente: 0 – 20 mA |
| Resolución Conversor Análogo-Digital | 8 Bits |

Tabla 9. Características del módulo analógico 1769-IF4XOF2

4 PROGRAMACIÓN DE LA INTERFAZ CAN

En este capítulo se describe la forma de programación de la red. Básicamente se define el software de programación que se encarga de la gestión de la misma.

Como se observó en el capítulo anterior, el reto más grande es la programación del controlador CAN (PIC18F258) que hará parte de cada uno de los tres nodos que integran la red.

La red está compuesta por un nodo maestro y dos nodos esclavos. El nodo maestro es el encargado de la parte de gestión en la red, y como puede observarse en el capítulo anterior está compuesto, además del PIC18F258, por el PLC Micrologix1500, lo cual lo hace responsable de la gestión de los datos en la red. Cada cierto tiempo establecido por el usuario de una manera conveniente el nodo maestro obtiene los datos necesarios para mantener el control de la red. Dichos datos son los valores enviados por los nodos esclavos hacia el nodo maestro, más específicamente las señales eléctricas presentes en los sensores, que corresponden al valor de la variable física sensada. De igual manera el nodo maestro envía datos hacia los nodos esclavos, correspondientes a las señales de control que influirán sobre el actuador de cada proceso. Más adelante se verá en detalle la forma de programación de cada uno de los nodos, dentro de lo cual se detalla los criterios para la escogencia del tiempo de envío de datos por parte del nodo maestro y de los nodos esclavos.

4.1 APLICACIÓN “MAESTRO”

Para la programación de cada módulo CAN de los PIC's es necesario la instalación de un software de Microchip denominado **Aplicación “Maestro”**.

4.1.1 Generalidades Aplicación “Maestro”

La aplicación “maestro” es una herramienta software “stand-alone” que permite a los usuarios configurar e incorporar módulos pre-escritos dentro de sus proyectos [10]. El usuario a través de una interfaz gráfica puede seleccionar uno o más módulos disponibles en la aplicación y configurar los parámetros de cada uno de acuerdo con los requerimientos necesarios. Una vez completado este procedimiento la aplicación “maestro” genera librerías que pueden ser incorporadas al proyecto de cada usuario haciendo uso de MPLAB IDE.

En el momento en que la aplicación “maestro” se ejecuta, la siguiente ventana principal se visualiza (Véase Figura 27):

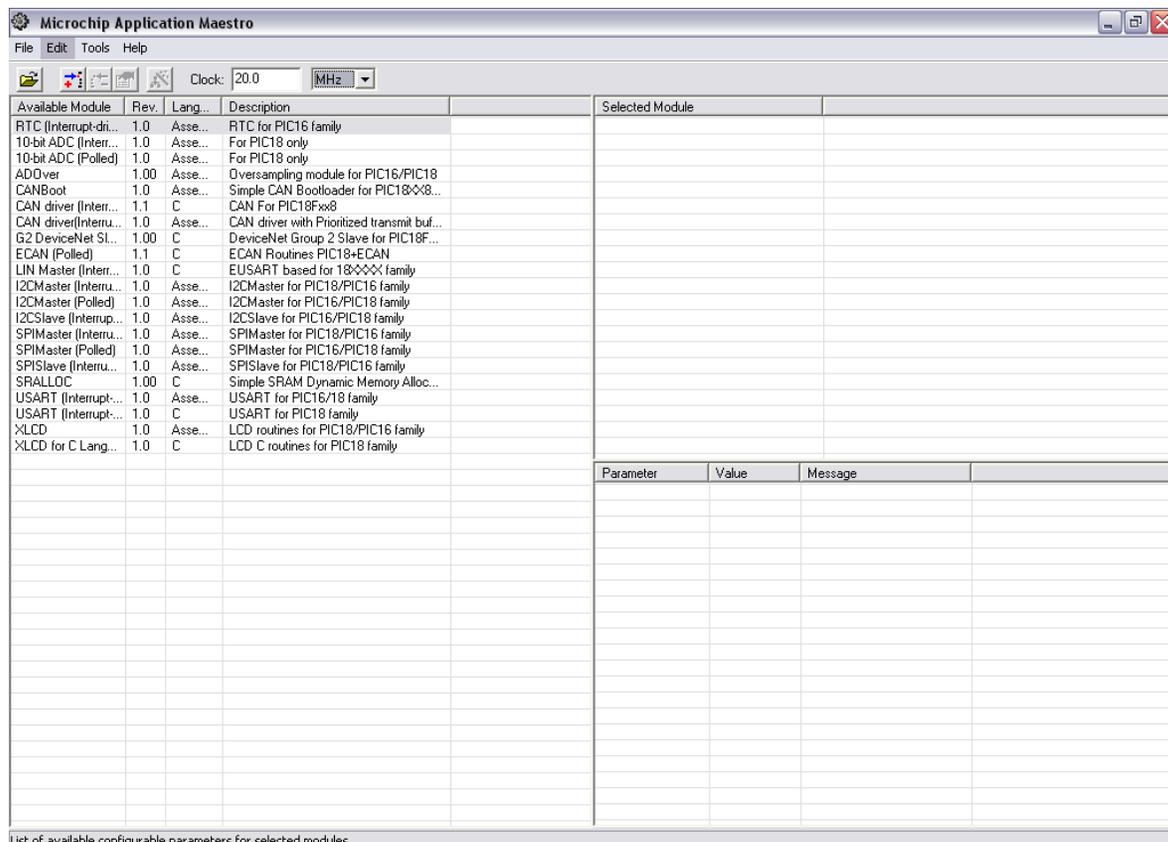


Figura 27. Ventana principal de la aplicación “maestro”

La ventana principal se divide en tres sub-ventanas, las cuales definen los módulos disponibles y sus opciones de configuración.

La sub-ventana ubicada en la parte izquierda (Figura 28) de la ventana principal muestra todos los módulos utilizables por la aplicación “maestro”.

La sub-ventana superior derecha (Figura 29) despliega el módulo que esta siendo configurado en el proyecto actual. La sub-ventana inferior derecha (Figura 30) despliega las opciones de configuración disponibles para el módulo que ha sido adicionado en la sub-ventana superior derecha.

Todas las funcionalidades del software de la aplicación “maestro” están disponibles tanto en la barra de menú ubicada en la parte superior, como en la barra de herramientas (Figura 31).

The screenshot shows the 'Microchip Application Maestro' window. At the top, there is a menu bar with 'File', 'Edit', 'Tools', and 'Help'. Below the menu bar is a toolbar with several icons. To the right of the toolbar, there is a 'Clock:' label, a text input field containing '20.0', and a dropdown menu set to 'Hz'. Below these elements is a table with the following columns: 'Available Module', 'Rev.', 'Lang...', and 'Description'. The table lists various modules such as RTC, ADC, CAN, I2C, SPI, and USART, along with their respective revision numbers and languages.

| Available Module | Rev. | Lang... | Description |
|-----------------------|------|---------|---|
| RTC (Interrupt-dri... | 1.0 | Asse... | RTC for PIC16 family |
| 10-bit ADC (Interr... | 1.0 | Asse... | For PIC18 only |
| 10-bit ADC (Polled) | 1.0 | Asse... | For PIC18 only |
| ADOver | 1.00 | Asse... | Oversampling module for PIC16/PIC18 |
| CANBoot | 1.0 | Asse... | Simple CAN Bootloader for PIC18xx8... |
| CAN driver (Interr... | 1.1 | C | CAN For PIC18Fxx8 |
| CAN driver(Interru... | 1.0 | Asse... | CAN driver with Prioritized transmit buf... |
| G2 DeviceNet SI... | 1.00 | C | DeviceNet Group 2 Slave for PIC18F... |
| ECAN (Polled) | 1.1 | C | ECAN Routines PIC18+ECAN |
| LIN Master (Interr... | 1.0 | C | EUSART based for 18xxx family |
| I2CMaster (Interru... | 1.0 | Asse... | I2CMaster for PIC18/PIC16 family |
| I2CMaster (Polled) | 1.0 | Asse... | I2CMaster for PIC16/PIC18 family |
| I2CSlave (Interrup... | 1.0 | Asse... | I2CSlave for PIC16/PIC18 family |
| SPIMaster (Interru... | 1.0 | Asse... | SPIMaster for PIC18/PIC16 family |
| SPIMaster (Polled) | 1.0 | Asse... | SPIMaster for PIC16/PIC18 family |
| SPISlave (Interru... | 1.0 | Asse... | SPISlave for PIC18/PIC16 family |
| SRALLOC | 1.00 | C | Simple SRAM Dynamic Memory Alloc... |
| USART (Interrupt... | 1.0 | Asse... | USART for PIC16/18 family |
| USART (Interrupt... | 1.0 | C | USART for PIC18 family |
| XLCD | 1.0 | Asse... | LCD routines for PIC18/PIC16 family |
| XLCD for C Lang... | 1.0 | C | LCD C routines for PIC18 family |

Figura 28. Ventana de los módulos disponibles

The image shows a software window with a standard Windows-style title bar containing minimize, maximize, and close buttons. Below the title bar is a header row with the text "Selected Module". The main area of the window is a table with two columns and approximately 15 rows, all of which are currently empty.

| Selected Module | |
|-----------------|--|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Figura 29. Ventana del módulo seleccionado

The image shows a software window displaying a configuration table. The table has four columns: "Parameter", "Value", "Message", and an empty fourth column. The table contains approximately 15 rows, all of which are empty.

| Parameter | Value | Message | |
|-----------|-------|---------|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Figura 30. Ventana de las opciones de configuración



Figura 31. Menú y barra de herramientas

4.1.2 Configuración Aplicación “Maestro”

Cuando se ejecuta por primera vez la aplicación “maestro” todos los módulos disponibles son desplegados en la sub-ventana izquierda. Se selecciona el módulo necesario para la aplicación; en este caso se adiciona el módulo CAN con el lenguaje fuente escrito en lenguaje C (Figura 32). La adición del módulo se realiza dando un “click” sobre su nombre y arrastrándolo hasta la sub-ventana superior derecha, o seleccionando el módulo dando un “click” sobre su nombre y haciendo un “click” sobre el botón “Add Module” ubicado en la barra de herramientas.

De igual manera un módulo puede ser removido de la aplicación en desarrollo. Se puede remover un módulo ya sea haciendo un “click” en su nombre en la sub-ventana superior derecha y arrastrándolo de vuelta hasta la sub-ventana izquierda, o seleccionando el módulo dando un “click” sobre su nombre y haciendo un “click” sobre el botón “Remove Module” ubicado en la barra de herramientas, o escogiendo la opción “Remove” del menú de edición, o presionando la tecla . Además se debe configurar la velocidad del reloj a la cual la aplicación trabajará. Para este caso se configura a 20MHz.

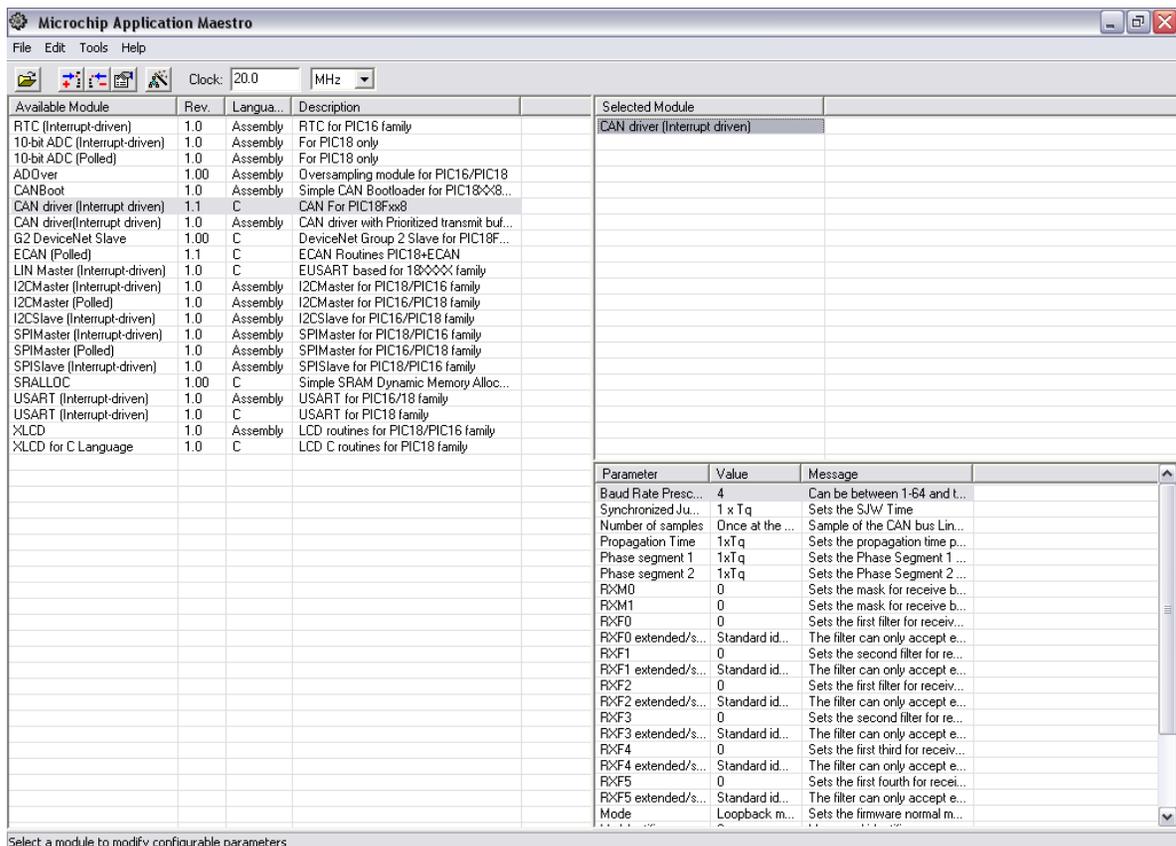


Figura 32. Selección del módulo CAN

4.1.3 Configuración del Módulo CAN

Una vez el módulo CAN ha sido adicionado a la aplicación “maestro”, debe ser configurado de acuerdo con la aplicación por implementar. El módulo CAN al igual que los otros módulos que hacen parte de la aplicación “maestro” tienen una lista predefinida de opciones o parámetros de configuración. Cuando el módulo CAN es seleccionado, aparece la lista de opciones de configuración disponibles en la sub-ventana inferior derecha (Figura 33). Para cambiar una opción determinada se realiza un doble “click” sobre la opción, lo cual despliega una ventana como la mostrada en la Figura 34 (cada ventana de configuración es diferente, depende del módulo y del parámetro seleccionado).

| Parameter | Value | Message |
|--------------------|-----------------|-------------------------------------|
| Baud Rate Presc... | 4 | Can be between 1-64 and t... |
| Synchronized Ju... | 1 x Tq | Sets the SJW Time |
| Number of samples | Once at the ... | Sample of the CAN bus Lin... |
| Propagation Time | 1xTq | Sets the propagation time p... |
| Phase segment 1 | 1xTq | Sets the Phase Segment 1 ... |
| Phase segment 2 | 1xTq | Sets the Phase Segment 2 ... |
| RXM0 | 0 | Sets the mask for receive b... |
| RXM1 | 0 | Sets the mask for receive b... |
| RXF0 | 0 | Sets the first filter for receiv... |
| RXF0 extended/s... | Standard id... | The filter can only accept e... |
| RXF1 | 0 | Sets the second filter for re... |
| RXF1 extended/s... | Standard id... | The filter can only accept e... |
| RXF2 | 0 | Sets the first filter for receiv... |
| RXF2 extended/s... | Standard id... | The filter can only accept e... |
| RXF3 | 0 | Sets the second filter for re... |
| RXF3 extended/s... | Standard id... | The filter can only accept e... |
| RXF4 | 0 | Sets the first third for receiv... |
| RXF4 extended/s... | Standard id... | The filter can only accept e... |
| RXF5 | 0 | Sets the first fourth for recei... |
| RXF5 extended/s... | Standard id... | The filter can only accept e... |
| Mode | Loopback m... | Sets the firmware normal m... |

Figura 33. Ventana de las opciones de configuración del módulo CAN

Microchip Application Maestro

File Edit Tools Help

Clock: 20.0 MHz

| Available Module | Rev. | Langua... | Description | Selected Module |
|-------------------------------|------|-----------|---|-------------------------------|
| RTC (Interrupt-driven) | 1.0 | Assembly | RTC for PIC16 family | CAN driver (Interrupt driven) |
| 10-bit ADC (Interrupt-driven) | 1.0 | Assembly | For PIC18 only | |
| 10-bit ADC (Polled) | 1.0 | Assembly | For PIC18 only | |
| ADOver | 1.00 | Assembly | Oversampling module for PIC16/PIC18 | |
| CANBoot | 1.0 | Assembly | Simple CAN Bootloader for PIC18xx8... | |
| CAN driver (Interrupt driven) | 1.1 | C | CAN For PIC18Fxx8 | |
| CAN driver (Interrupt driven) | 1.0 | Assembly | CAN driver with Prioritized transmit buf... | |
| G2 DeviceNet Slave | 1.00 | C | DeviceNet Group 2 Slave for PIC18F... | |
| ECAN (Polled) | 1.1 | C | ECAN Routines PIC18+ECAN | |
| LIN Master (Interrupt-driven) | 1.0 | C | EUSART based for 18xxx family | |
| I2CMaster (Interrupt-driven) | 1.0 | Assembly | I2CMaster for PIC18/PIC16 family | |
| I2CMaster (Polled) | 1.0 | Assembly | I2CMaster for PIC16/PIC18 family | |
| I2CSlave (Interrupt-driven) | 1.0 | Assembly | I2CSlave for PIC16/PIC18 family | |
| SPIMaster (Interrupt-driven) | 1.0 | Assembly | SPIMaster for PIC18/PIC16 family | |
| SPIMaster (Polled) | 1.0 | Assembly | SPIMaster for PIC16/PIC18 family | |
| SPISlave (Interrupt-driven) | 1.0 | Assembly | SPISlave for PIC18/PIC16 family | |
| SRALLOC | 1.00 | C | Simple SRAM Dynamic Memory Alloc... | |
| USART (Interrupt-driven) | 1.0 | Assembly | USART for PIC16/18 family | |
| USART (Interrupt-driven) | 1.0 | C | USART for PIC18 family | |
| XLCD | 1.0 | Assembly | LCD routines for PIC18/PIC16 family | |
| XLCD for C Language | 1.0 | C | LCD C routines for PIC18 family | |

Modify Parameter

Baud Rate Prescaler value

Parameter Message

Can be between 1-64 and the prescaler is set by Tq = (2xprescaler value)/Fosc

OK Cancel

| Parameter | Value | Message |
|--------------------|-----------------|-------------------------------------|
| Baud Rate Presc... | 4 | Can be between 1-64 and t... |
| Synchronized Ju... | 1 x Tq | Sets the SJW Time |
| Number of samples | Once at the ... | Sample of the CAN bus Lin... |
| Propagation Time | 1xTq | Sets the propagation time p... |
| Phase segment 1 | 1xTq | Sets the Phase Segment 1 ... |
| Phase segment 2 | 1xTq | Sets the Phase Segment 2 ... |
| RXM0 | 0 | Sets the mask for receive b... |
| RXM1 | 0 | Sets the mask for receive b... |
| RXF0 | 0 | Sets the first filter for receiv... |
| RXF0 extended/s... | Standard id... | The filter can only accept e... |
| RXF1 | 0 | Sets the second filter for re... |
| RXF1 extended/s... | Standard id... | The filter can only accept e... |
| RXF2 | 0 | Sets the first filter for receiv... |
| RXF2 extended/s... | Standard id... | The filter can only accept e... |
| RXF3 | 0 | Sets the second filter for re... |
| RXF3 extended/s... | Standard id... | The filter can only accept e... |
| RXF4 | 0 | Sets the first third for receiv... |
| RXF4 extended/s... | Standard id... | The filter can only accept e... |
| RXF5 | 0 | Sets the first fourth for recei... |
| RXF5 extended/s... | Standard id... | The filter can only accept e... |
| Mode | Loopback m... | Sets the firmware normal m... |

List of available configurable parameters for selected modules

Figura 34. Ventana de configuración de los parámetros del módulo CAN

La ventana aparece con el valor actual de la opción; se puede cambiar el valor según se necesite; si el valor cambiado no se encuentra dentro de los valores predeterminados de la opción un mensaje de error se despliega (Véase Figura 35).

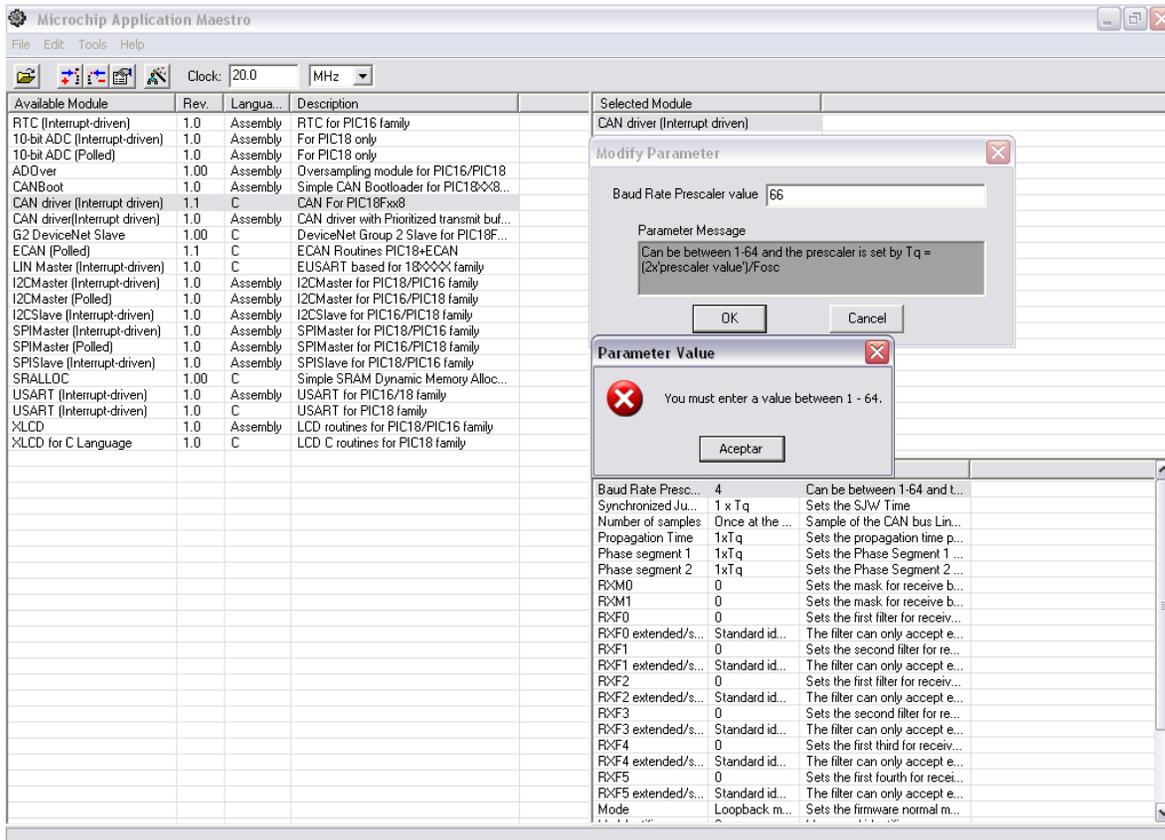


Figura 35. Ventana de error de la configuración

La configuración de los parámetros del módulo CAN y el criterio de escogencia de los mismos es explicada a continuación:

- “*Baud Rate Prescaler*” (BRP): El valor de este parámetro es su valor por defecto (4), debido a que se configura únicamente cuando todos los dispositivos que hacen parte de la red no tienen o no trabajan a la misma frecuencia de oscilación, es decir, cada dispositivo tiene un oscilador de valores diferentes. Si cada dispositivo tiene diferentes frecuencias de reloj, el “*bit rate*” tiene que ser ajustado fijando el “*baud rate prescaler*” y el número del “*time quanta*” en cada segmento. Cada nodo que hace parte de la red está trabajando con un oscilador de 20MHz, lo que significa que todos los dispositivos están funcionando con la misma frecuencia de oscilación, por tal razón este parámetro es indiferente en la

aplicación. El valor del “*baud rate prescaler*” puede variar entre los valores enteros de 1-64.

- “*Synchronization Segment*” (Sync-Seg): Hace parte del “*nominal bit time*”. El “*nominal bit time*” puede ser visto como un conjunto de segmentos de tiempo (Figura 36), estos segmentos son el “*synchronization segment*” (Sync-Seg), “*propagation time segment*” (Prop-Seg), “*phase buffer segment 1*” (Phase-Seg1), “*phase buffer segment 2*” (Phase-Seg2), que son explicados mas adelante. Estos tiempos son unidades enteras llamados “*time quanta*” o Tq. El “*nominal bit time*” esta definido así:

$$Nominal\ Bit\ Time = 1 / Nominal\ Bit\ Rate \quad (1)$$

O

$$Nominal\ Bit\ Time = Tq \times (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) \quad (2)$$

Donde el “*nominal bit rate*” es el número de bits transmitidos por segundo y puede soportar hasta un valor máximo de 1 Mb/s. Por defecto el “*nominal bit time*” es programable desde un mínimo de 8 Tq hasta un máximo de 25 Tq. El mínimo “*nominal bit time*” que se puede tener es 1 μ s correspondiente a un valor máximo del “*nominal bit rate*” 1 Mb/s.

El “*time quanta*” (Tq) es un valor derivado del periodo del oscilador y del “*baud rate prescaler*”, por lo tanto:

$$Tq(\mu s) = (2 \times (BRP + 1)) / F_{osc}(MHz) \quad (3)$$

Haciendo uso de la ecuación anterior (3) y de acuerdo al valor dado a BRP y del oscilador usado en la aplicación se tiene:

$$Tq(\mu s) = (2 \times (4 + 1)) / 20(MHz) \quad (4)$$

$$Tq(\mu s) = 0.5 \quad (5)$$

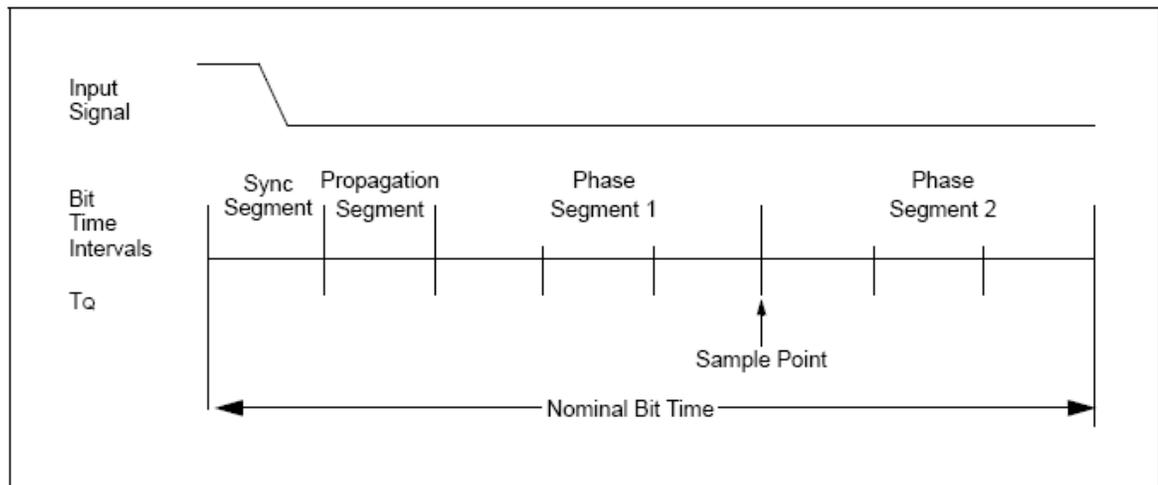


Figura 36. “Nominal bit time”

- “*Number of Samples*”: Su valor puede ser escogido entre sólo dos opciones, un muestreo o múltiples muestreos en el “*sample point*”. El “*sample point*” es el instante de tiempo en el cual el nivel del bus es leído y el valor del bit recibido es determinado. Es posible especificar múltiples muestreos del bus en el “*sample point*”, si el “*nominal bit time*” es lento y contiene múltiples T_q . El valor de este parámetro se fija en múltiples muestreos ya que se cumplen las condiciones mencionadas anteriormente.
- “*Propagation Time Segment*” (Prop_Seg): Esta parte del “*nominal bit time*” es usada para compensar los tiempos de retardos físicos dentro de la red. Los tiempos de retardos están compuestos del tiempo de la señal de propagación en el bus y el tiempo interno de retardo de los nodos. El tamaño del “*propagation time segment*” puede ser programado desde 1 T_q hasta 8 T_q . El valor del “*propagation time segment*” esta definido en 2 T_q , a causa de que en la red no se presentan retardos muy significativos. El retardo debido a la transmisión de la señal por el bus es del orden de los μs (Ver Tabla 1) al igual que el retardo que adicionan los nodos por el tiempo de ejecución del programa.
- “*Phase Buffer Segment 1*” (Phase_Seg1) y “*Phase Buffer Segment 2*” (Phase_Seg2): Estos segmentos son usados para ubicar de manera óptima el punto de muestreo dentro del “*nominal bit time*”. Como se observa en la Figura 36 el punto de muestreo ocurre entre el “*phase buffer segment 1*” y el “*phase buffer segment 2*”. Estos segmentos pueden ser acortados o alargados con la re-sincronización del proceso, y ambos pueden tomar valores desde 1 T_q a 8 T_q . La

terminación del “*phase buffer segment 1*” determina el punto de muestreo dentro del “*nominal bit time*” y el “*phase buffer segment 2*” proporciona un retardo antes de la próxima transmisión de un dato. Los valores definidos para estos dos parámetros son 7 Tq y 6 Tq respectivamente, esto gracias a que se desea tener un valor del nominal bit rate de 125 Kb/s que es suficiente para la aplicación. Así, basándose en la ecuación 2 y 5 se llega a:

$$No\ min\ al\ Bit\ Time = Tq \times (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2)$$

$$No\ min\ al\ Bit\ Time = 0.5\mu s \times (1 + 2 + 7 + 6)$$

$$No\ min\ al\ Bit\ Time = 8\mu s$$

Ahora para hallar el nominal bit rate se aplica la ecuación 1:

$$No\ min\ al\ Bit\ Time = 1 / No\ min\ al\ Bit\ Rate$$

$$No\ min\ al\ Bit\ Rate = 1 / No\ min\ al\ Bit\ Time$$

$$No\ min\ al\ Bit\ Rate = 1 / 8\mu s$$

$$No\ min\ al\ Bit\ Rate = 125\ Kb / s$$

- RXM0 y RXM1: Son las máscaras de aceptación asociadas a cada “*buffer*” de recepción, RXB0 y RXB1 respectivamente, se usan para determinar si un mensaje es admitido por alguno de los “*buffers*” de recepción, así mismo determinan cuáles bits del identificador van a ser comparados con los filtros de aceptación. Cualquier bit de la máscara de aceptación que sea fijado en cero automáticamente será aceptado sin importar el bit del filtro de aceptación. Los registros de las máscaras solo pueden ser configurados si el PIC esta en el modo de operación de configuración.
- RXF0, RXF1, RXF2, RXF3, RXF4, RXF5: RXF0 y RXF1 son los filtros de aceptación asociados al “*buffer*” de recepción RXB0. Mientras que RXF2, RXF3, RXF4, RXF5 son los filtros asociados al “*buffer*” de recepción RXB1. En el momento en que un mensaje válido es recibido en el MAB los campos del

identificador son comparados con los valores de los filtros; si los valores concuerdan el mensaje es cargado en el “*buffer*” de recepción apropiado. Los registros de los filtros de aceptación al igual que los registros de las máscaras de aceptación solo pueden ser configurados si el PIC esta en el modo de operación de configuración. Para el nodo maestro los filtros de aceptación han sido configurados con los números del 1 al 6.

- “*RXF0 extended/standar, RXF1 extended/standar, RXF2 extended/standar, RXF3 extended/standar, RXF4 extended/standar y RXF5 extended/standar*”: Estos parámetros definen si se aceptan mensajes extendidos o estándares; en la aplicación sólo son usados mensajes estándares por los requerimientos de la red y su número pequeño de nodos.
- “*Mode*”: Define el modo de operación del módulo CAN. Cuando se cambian los modos de operación, el modo de operación no cambiará hasta que se completen todas las transmisiones de los mensajes pendientes. Existen seis modos de operación como se detalló en el capítulo 3. Haciendo uso de la aplicación “*maestro*” solo se puede definir el modo normal o “*loopback*”. El modo configuración no puede ser escogido, pues definiendo los parámetros mencionados anteriormente como el BRP, “*Syng_Seg*”, etc, ya se está definiendo los registros de configuración y la inicialización del módulo. Por esta razón el modo de operación es fijado a modo normal.
- “*My identifier*”: Es el identificador que va a ser asignado a cada nodo o mensaje y que lo hará único. Ya que en la aplicación se cuenta con tres nodos y cada uno de ellos es responsable de una variable o mensaje, los identificadores que han sido asignados para cada mensaje son 1, 2 y 3 respectivamente, donde el identificador 1 pertenece a la variable nivel, el 2 a la variable temperatura y el 3 es el mensaje donde se almacenan las señales de control enviadas por el nodo maestro a los nodos esclavos.
- “*My identifier is Std/Ext*”: Define si el identificador asignado es estándar o extendido; como se mencionó anteriormente sólo se hace uso de identificadores estándares en la aplicación; por tal motivo este parámetro se define como identificador estándar.
- “*RX buffer*” y “*TX buffer*”: Define el tamaño del “*buffer*”, cada “*buffer*” ocupa 14 bytes de SRAM. Se ha dejado los valores por defecto de 4 para cada uno de ellos.

- **“Interrupt Priority”**: Define la prioridad de la interrupción CAN, es decir, la interrupción que se presenta en el módulo cuando un mensaje es transmitido o recibido. Puede ser tanto alta como baja. Para la aplicación se ha definido como interrupción alta.
- **“Enable CAN Error Handler”**: Habilita al usuario para desarrollar una rutina de error que le permite saber cuándo se ha presentado alguna falla en el sistema. La rutina implementada para la aplicación se describe posteriormente.

4.1.4 Generación de Código

Una vez el módulo CAN es configurado se procede a crear los archivos haciendo uso del botón **“Generate Code”** ubicado en la barra de herramientas o en el menú de herramientas. Al generar el código es necesario conocer en qué directorio se encuentra el proyecto al cual se le adicionarán los archivos generados por la aplicación **“maestro”**; de esta manera se guardarán en dicho directorio todos los archivos generados (Véase Figura 37).

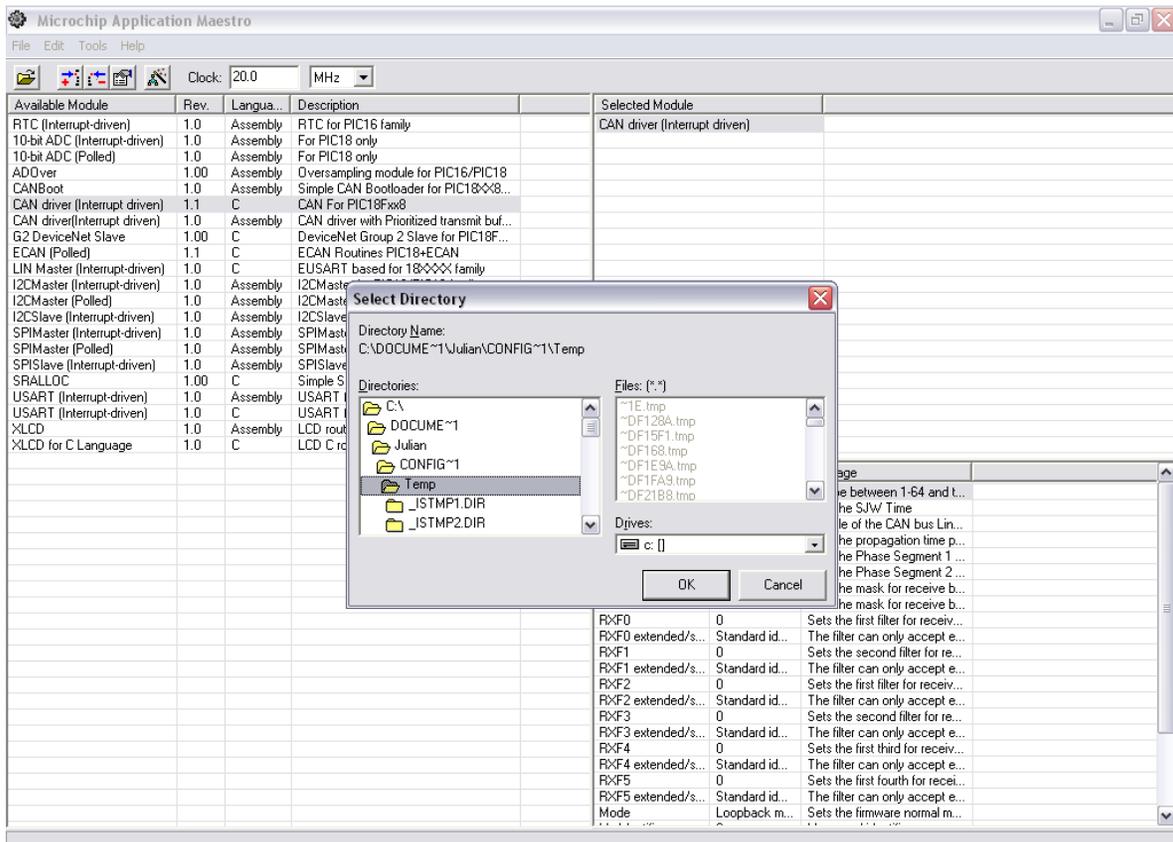


Figura 37. Ventana de la generación de código

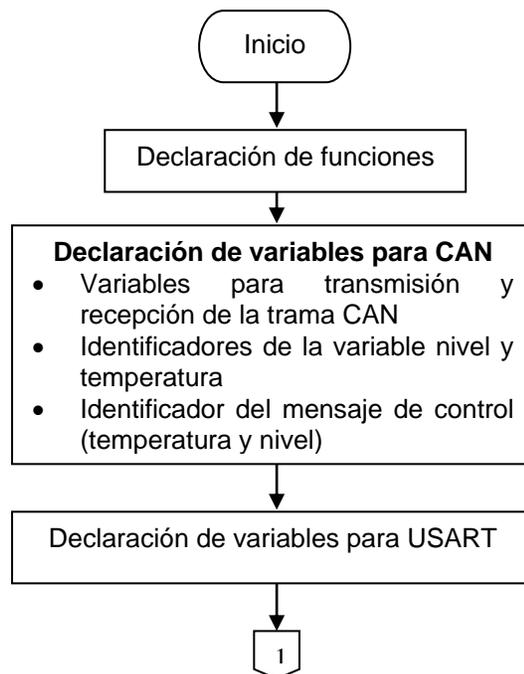
Los archivos generados por la aplicación “*maestro*” son los siguientes:

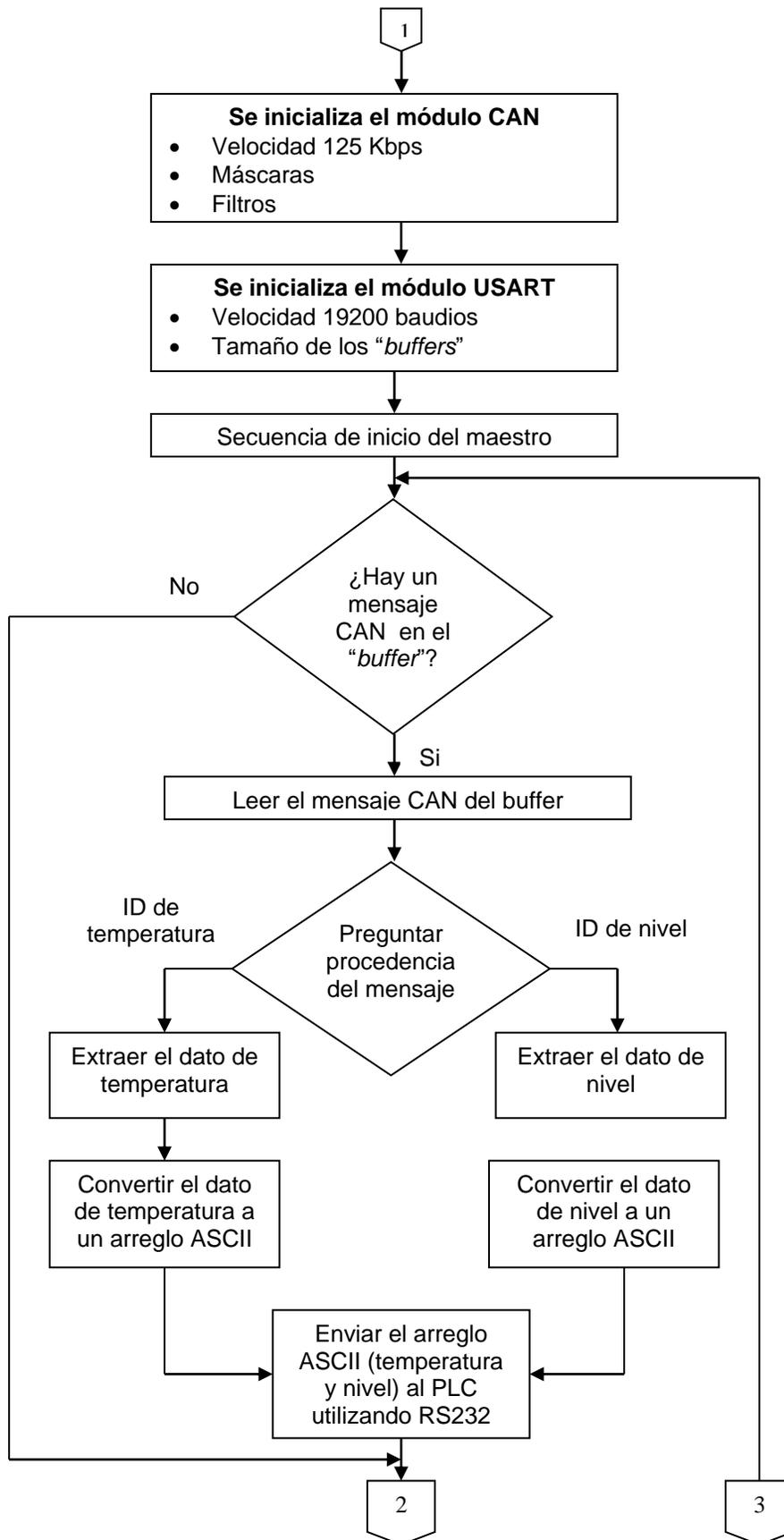
- Can.c que contiene los “*drivers*” para el PIC, es decir, se definen las estructuras de Tx y Rx en lenguaje C, al igual que las funciones CAN usadas por el módulo CAN.
- CanDef.h que contiene el valor de cada parámetro del módulo CAN una vez fue generado el código por la aplicación “*maestro*”.
- Can.h que contiene las diferentes funciones CAN; es el encabezado de Can.c.
- CanIntC18.txt que contiene un ejemplo sencillo del modo de operación de algunas funciones del módulo CAN.
- CanIntCReadMe.pdf que contiene la definición del módulo y explicaciones acerca del modo de operación de las funciones del módulo CAN.

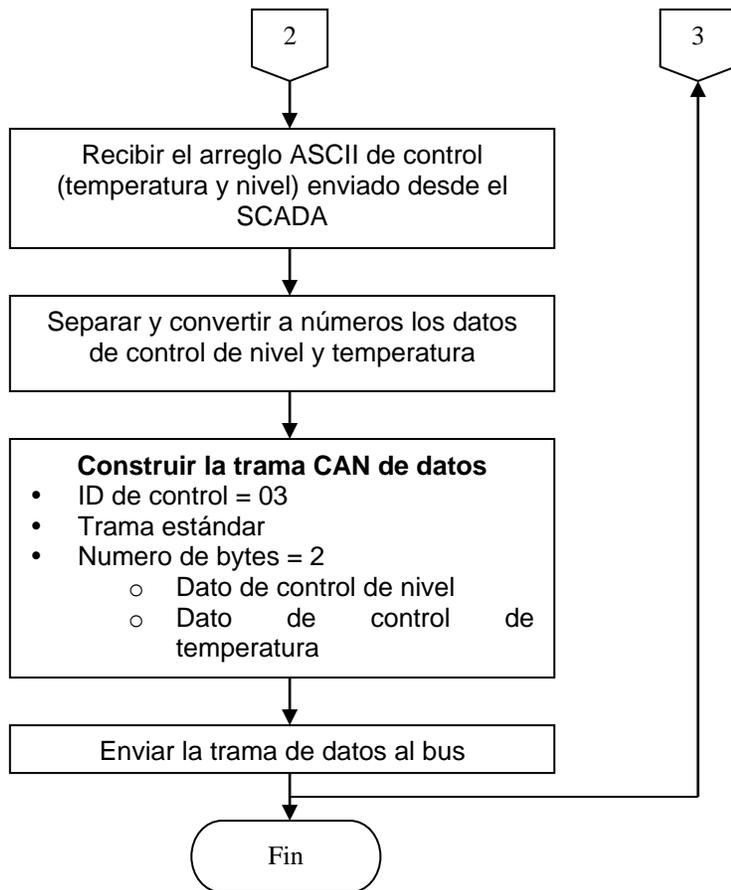
4.2 DIAGRAMAS DE FLUJO PARA LA PROGRAMACIÓN DE LOS NODOS

A continuación se muestran los diagramas de flujo que se desarrollaron y se implementaron en la programación de cada uno de los nodos.

4.2.1 Nodo Maestro

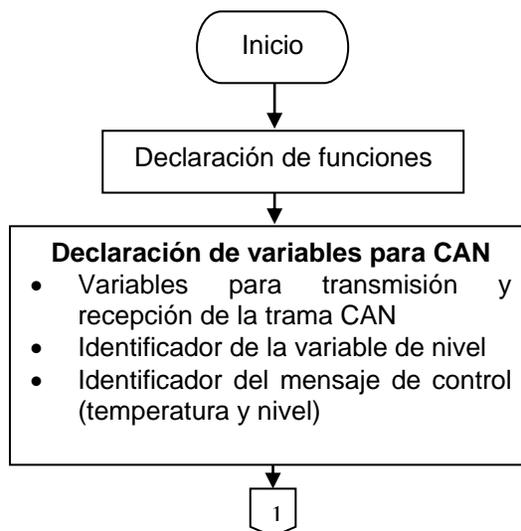


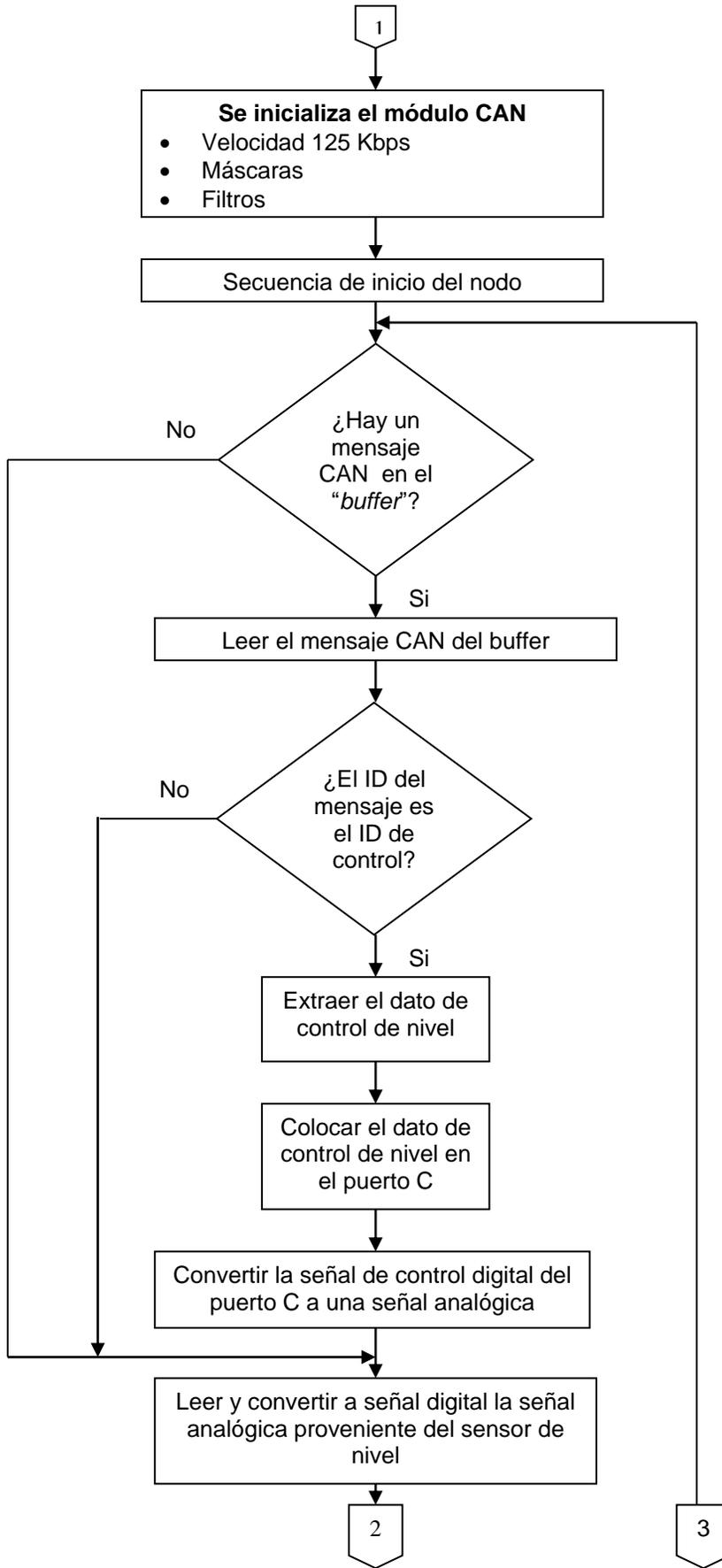


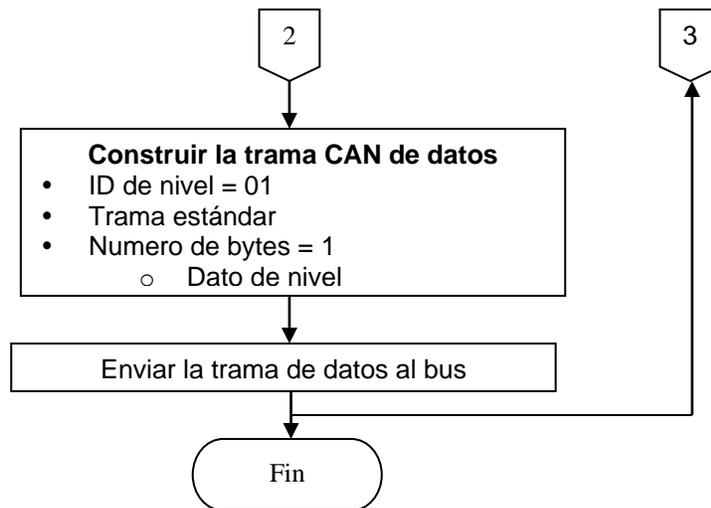


4.2.2 Nodo Esclavo 1

El proceso de nivel que tiene como componentes principales un sensor de nivel y una válvula esta conectado al nodo esclavo 1.

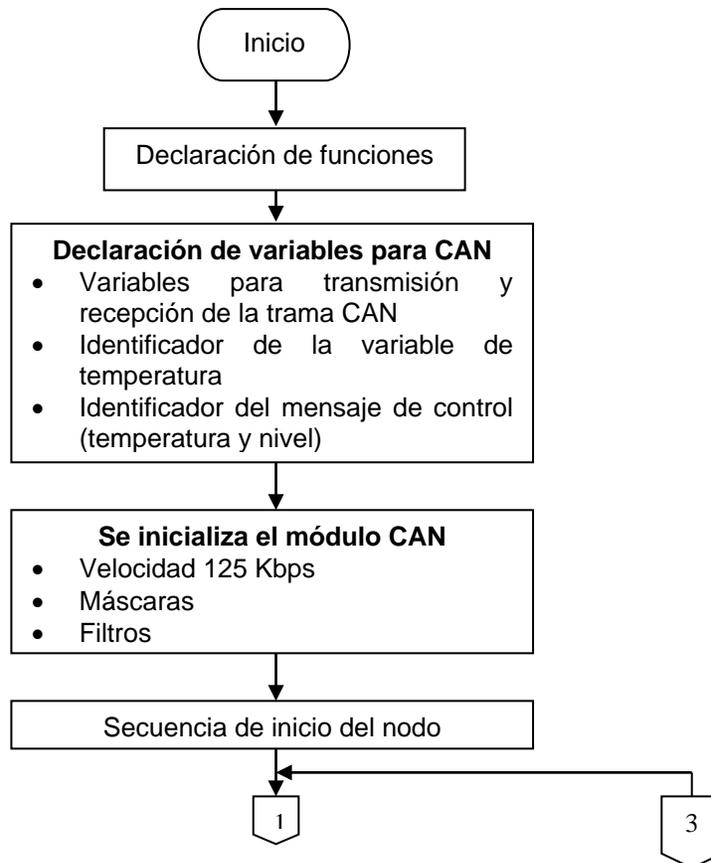


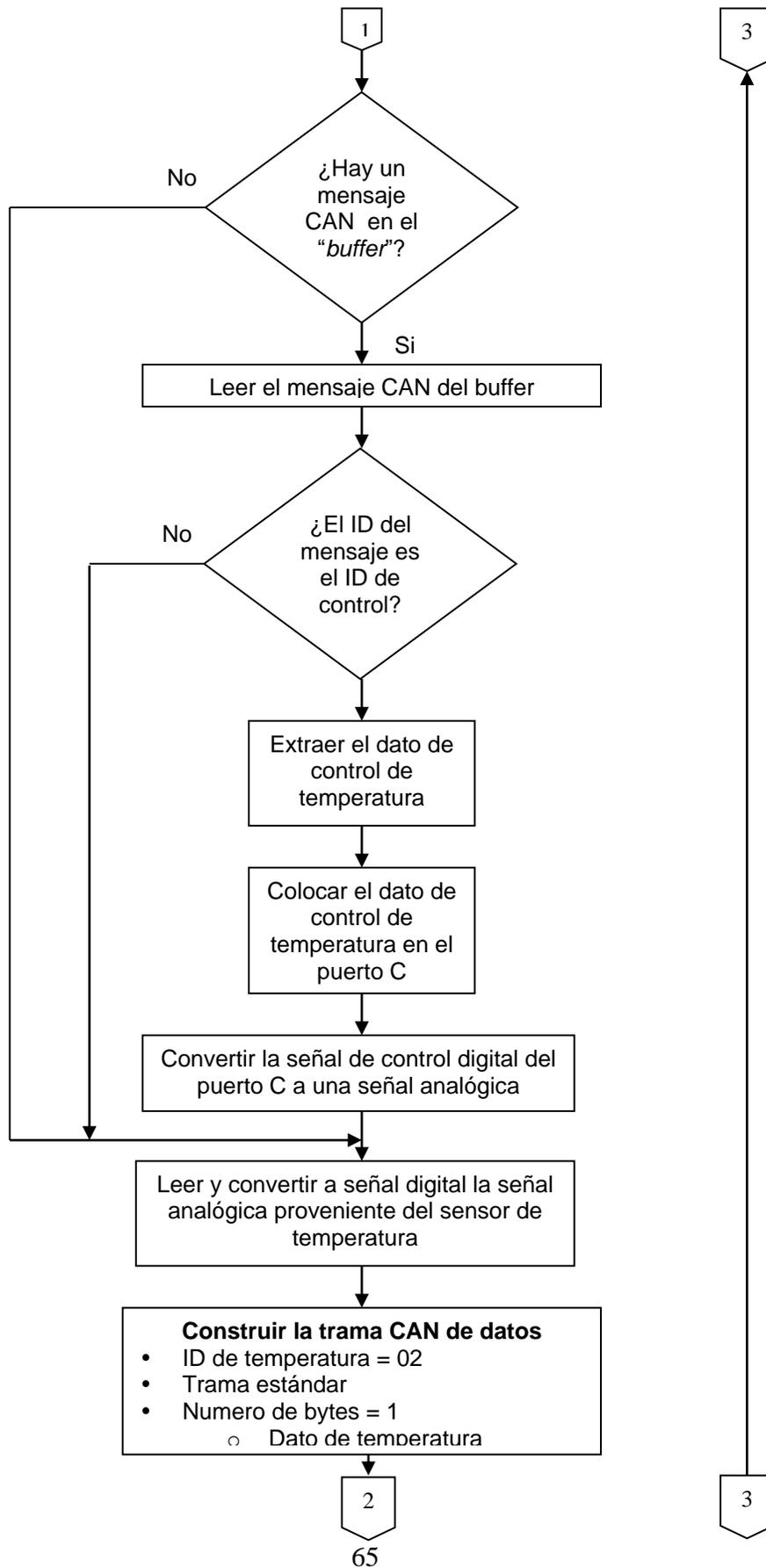


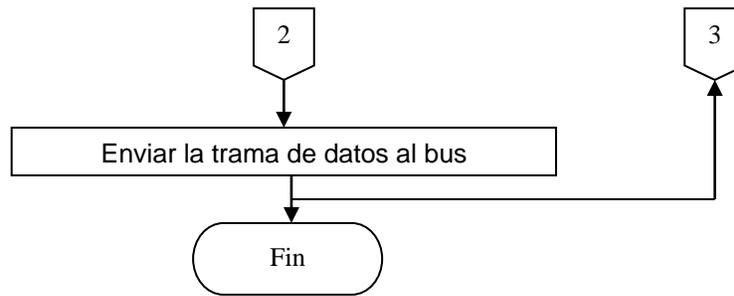


4.2.3 Nodo Esclavo 2

El proceso de temperatura que tiene como componentes principales un sensor de temperatura (LM35) y una malla de calefacción esta conectado al nodo esclavo 2.







Para observar detalladamente el código usado para la programación de cada uno de los nodos dirigirse al Anexo A y si se desea conocer la forma de adicionar un nodo a la red observar el Anexo B.

5 DISEÑO DE LAS INTERFACES CAN

5.1 INTRODUCCIÓN

En este capítulo se explica el diseño circuital de cada interfaz CAN (nodo) de la red, haciendo una descripción del funcionamiento de los elementos que la componen.

5.2 NODOS ESCLAVOS 1 Y 2

A continuación se muestra y se describen los componentes de las tarjetas impresas desarrolladas para cada nodo esclavo.

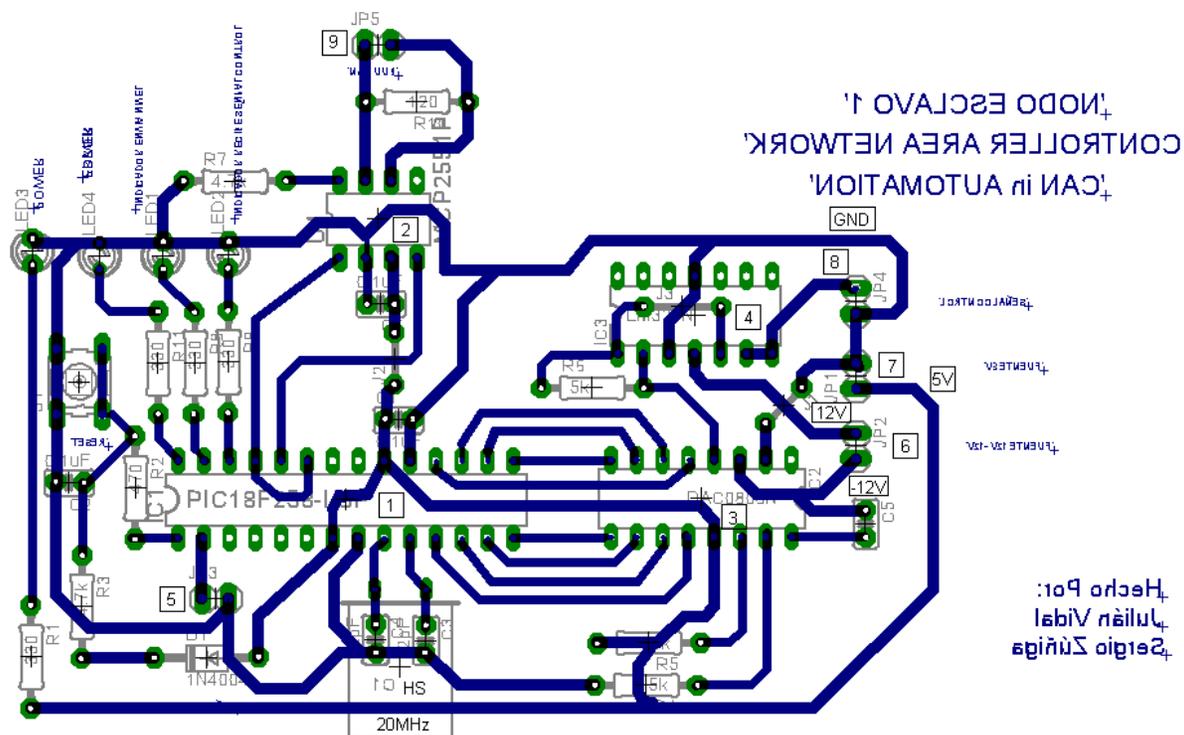


Figura 38. Diseño circuital de los nodos esclavos 1 y 2

Siguiendo la numeración utilizada en la gráfica anterior se detalla cada uno de sus elementos.

1. PIC18F258

Está compuesto por 28 pines, así:

- Pin 1: Recibe la señal del “*RESET*”.
- Pin 2: Entrada analógica, que para los nodos 1 y 2 son señales provenientes de los sensores de nivel y de temperatura respectivamente. Adicionalmente el nodo 2 tiene a la entrada de este pin una resistencia de 250Ω encargada de convertir la señal de $4\text{mA} - 20\text{mA}$ proveniente del sensor de temperatura (LM35) a una señal de voltaje de $1\text{V} - 5\text{V}$.
- Pin 3 – 7: No implementados.
- Pin 8 y 19: Conectados a tierra (GND).
- Pin 9 y 10: Entradas de reloj. Se hace uso de un reloj de 20MHz.
- Pin 11 – 18: Componen el puerto C del PIC. Sirven como salida digital de 8 bits de los valores de las señales de control enviadas por el nodo maestro hacia los nodos esclavos. Dichas señales son recibidas por un conversor Digital - Análogo (DAC0808).
- Pin 20: Entrada de voltaje de alimentación del PIC (5V).
- Pin 21 – 28: Conforman el puerto B del PIC. Los pines 23 y 24 son los encargados de la transmisión y recepción de las señales del módulo CAN y están conectados a los pines de transmisión y recepción del transductor MCP2551 respectivamente. Los pines 26 a 28 son utilizados para desplegar información visual sobre el funcionamiento del mismo. El pin 26 esta conectado a un LED que sirve como indicador para visualizar en que instante de tiempo los nodos esclavos están enviando la señal proveniente del sensor hacia el nodo maestro a través de la red. Cuando se esta ejecutando esta acción el LED titila de forma intermitente. El pin 27, al igual que el pin 26, esta conectado a un LED y tiene la misma función que el pin 26 la diferencia es que titila una vez los nodos esclavos han recibido la señal de control proveniente del nodo maestro a través de la red. El pin 28 esta conectado a un LED al igual que los pines 26 y 27, y es un indicador de error. Si se presenta un error definido por CAN (Ver apartado 1.8) dicho LED se encenderá.

2. Transductor CAN MCP2551

Consta de 8 pines que consisten en:

- Pin 1 y 4: Reciben las señales de transmisión y recepción provenientes del módulo CAN del PIC.
- Pin 2: Conectado a tierra (GND).
- Pin 3: Entrada de voltaje de alimentación (5V).
- Pin 5: No implementado.
- Pin 6 y 7: Señales adecuadas a los niveles CAN que van hacia el bus físico, CANL y CANH respectivamente.
- Pin 8: Conectado a una resistencia para trabajar en el modo “*Slope-Control*”, que permite reducir las interferencias electromagnéticas [7] (pagina 3)

3. Conversor Digital – Análogo DAC0808

Recibe las señales digitales provenientes del puerto C del PIC que corresponden a la señal de control enviada por el nodo maestro. Consta de 16 pines descritos a continuación:

- Pin 1: No implementado.
- Pin 2: Conectado a GND.
- Pin 3: Conectado a -12V.
- Pin 4: Salida del conversor que es recibida por un amplificador operacional para su posterior adecuación.
- Pin 5 – 12: Entradas digitales provenientes del PIC.
- Pin 13: Conectado a 5V.
- Pin 14: Voltaje de referencia. El valor de la salida analógica, en este caso la señal de control, depende del voltaje con que se alimente este pin. Para el nodo 1 (nivel) se ha polarizado con 5V, ya que se necesita que la señal de control se encuentre dentro del rango de 1V – 5V, señal que es recibida por la válvula de control. Para el caso del nodo 2 (temperatura) se ha polarizado con 12V y así obtener una señal de control dentro del rango de 0V - 10V que es recibida por el actuador (malla de calefacción).

- Pin 15: Conectado a una resistencia.
- Pin 16: Conectado a un condensador que a su vez esta polarizado con -12V.

Para más información acerca de este dispositivo y de su circuito típico de aplicación ver [11].

4. LM324

El LM324 esta compuesto por 4 amplificadores operacionales, de los cuales solo se hará uso de dos de ellos. El primero de los mismos se usa para adecuar la señal entregada por el DAC a los niveles adecuados de voltaje y hace parte fundamental del circuito típico de aplicación del DAC [11]. El segundo amplificador se usa como impulsor de corriente de manera preventiva, por lo cual la señal proveniente del primer amplificador es recibida por la entrada no inversora del segundo. La señal de salida del impulsor de corriente es la señal de control enviada hacia los respectivos actuadores de las plantas. Para más información acerca de este dispositivo ver [12]. Los pines 4 y 11 están conectados a 12V y GND respectivamente.

5. Conector de la señal del sensor o entrada analógica

Conector de dos pines a donde llega la señal de los sensores de nivel y temperatura. Como ya se menciona anteriormente corresponde al conversor A/D (pin 2) del PIC.

6. Fuente de alimentación de 12V /-12V

Conector de dos pines donde llegan las señales de 12V y -12V por medio de los pines 4 y 5 del conector RJ45 respectivamente. Estas dos señales son necesarias para la alimentación de ciertos dispositivos de la interfaz CAN de los nodos.

7. Fuente de alimentación de 5V

Conector de dos pines donde llegan las señales de 5V y GND por medio de los pines 7 y 8 del conector RJ45 respectivamente. Además se cuenta con un LED indicador de POWER conectado a esta fuente para conocer si los nodos se encuentran encendidos o si se ha presentado alguna falla en la fuente de alimentación de 5V.

8. Señal de control

Señal enviada hacia los actuadores de las plantas de nivel y temperatura a través de un conector de dos pines. Como se menciona anteriormente esta señal es entregada por el impulsor de corriente (ver apartado 5.2, numeral 4).

9. Salida CAN

Salidas CAN_H y CAN_L que llegan al medio de transmisión físico (cable UTP).

5.3 NODO MAESTRO

A continuación se muestra y se describen los componentes de la tarjeta impresa desarrollada para el nodo maestro.

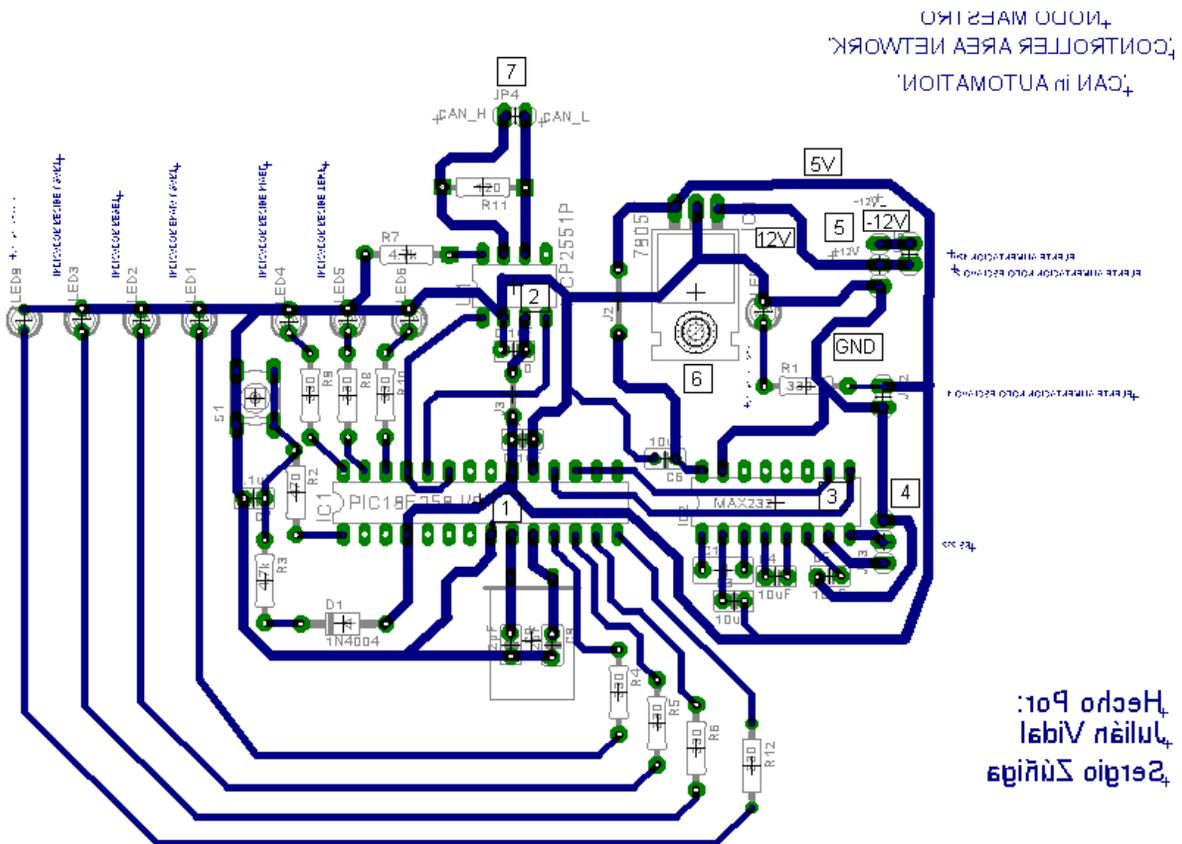


Figura 39. Diseño circuital del nodo maestro

Siguiendo la numeración utilizada en la gráfica anterior se detalla cada uno de sus elementos.

1. PIC18F258

A diferencia de los nodos esclavos el nodo maestro no usa el pin 2, ya que esta obteniendo las señales de los sensores de temperatura y de nivel por medio de la red CAN y las señales de control del PLC a través del USART.

El puerto C se usa para desplegar información visual sobre el funcionamiento del mismo, mas específicamente los pines 11, 12, 13 y 14, donde cada pin esta conectado a un LED. El pin 11 sirve como indicador para visualizar en que instante de tiempo el nodo maestro está enviando las señales provenientes de los sensores hacia el PLC a través de la comunicación USART o RS-232. El pin 12 sirve como indicador para visualizar cuando se ha puesto en marcha al nodo maestro (secuencia de inicio). El pin 13 sirve como indicador para visualizar en que instante de tiempo el nodo maestro está recibiendo las señales de control provenientes del PLC a través de la comunicación USART o RS-232 y el pin 14 es un indicador de error que opera de la misma manera que el indicador de error de los nodos esclavos (Ver apartado 5.2, Numeral 1). Los pines 17 y 18 son empleados para la comunicación serial o RS-232 y se conectan al MAX232, dispositivo descrito posteriormente.

Los pines 26, 27 y 28 al igual que los pines 11, 12, 13 y 14 sirven como indicadores. El pin 26 indica en que instante de tiempo se están enviando las señales de control provenientes del PLC hacia los nodos esclavos a través de la red. El pin 27 indica cuando el nodo maestro ha recibido la señal proveniente del sensor de temperatura a través de la red, y el pin 28 indica en que instante el nodo maestro ha recibido la señal proveniente del sensor de nivel a través de la red.

Los pines no mencionados anteriormente cumplen la misma función descrita en el apartado 5.2, Numeral 1.

2. Transductor CAN MCP2551

El diseño y la función de sus pines es similar a la descrita en el apartado 5.2, Numeral 2.

3. MAX232

Su función es adecuar las señales provenientes del PIC para ser transmitidas a través de la comunicación serial hacia el PLC y viceversa.

4. Salida RS-232

Conector de tres pines y medio físico a través del cual se realiza la comunicación bidireccional con el PLC.

5. Fuente de alimentación de 12V/-12V/GND

Conector de tres pines a donde llegan las señales de 12V, -12V y GND que corresponden a los pines 4,5 y 7 respectivamente del conector RJ45. Dichas señales provienen de la tarjeta de alimentación y son necesarias para la alimentación de los dispositivos de la interfaz CAN del nodo.

6. Regulador de voltaje

Regula a 5V la señal de 12V proveniente del pin 4 del conector RJ45, esto con el fin de proteger y alimentar ciertos dispositivos de la interfaz CAN.

7. Salida CAN

Salidas CAN_H y CAN_L que llegan al medio de transmisión físico (cable UTP).

5.4 FUENTES DE ALIMENTACIÓN Y BUS

A continuación se muestra la tarjeta impresa diseñada y desarrollada para el bus y para la fuente de alimentación de cada nodo.

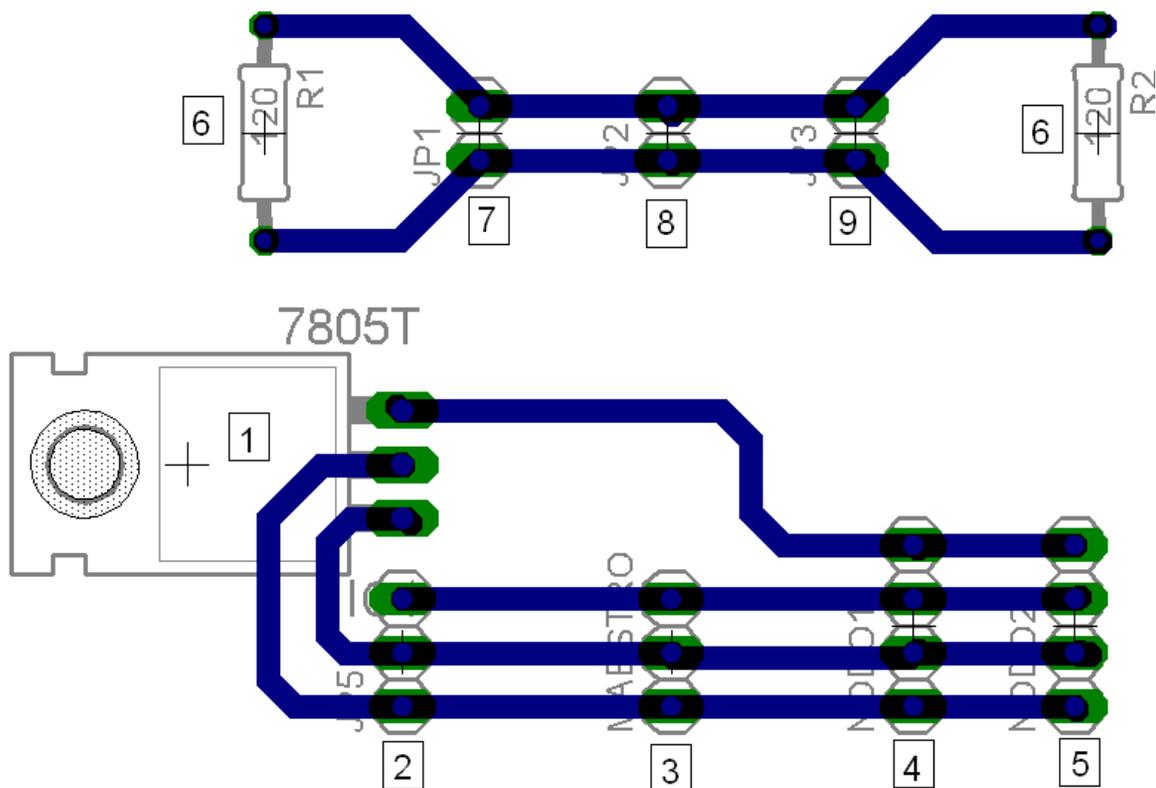


Figura 40. Diseño circuitual del bus y de la fuente de alimentación de cada nodo

Siguiendo la numeración utilizada en la gráfica anterior se detalla cada uno de sus elementos.

1. Regulador LM7805

Regula a 5V la señal de voltaje de 12V proveniente de la fuente externa conectada a 2 según la gráfica anterior.

2. Entrada fuente externa

Conector de 3 pines de la fuente externa encargada se suplir los voltajes necesarios para los diferentes nodos. Según la distribución de pines del conector de abajo hacia arriba se tiene: GND, 12V y -12V. Como se puede observar en la gráfica anterior la señal de 12V y GND se conectan al regulador.

3. Fuente de alimentación del nodo maestro

Conector de tres pines correspondientes a GND, 12V y -12V. Estas señales viajan hacia el nodo maestro a través del cable UTP y por medio de los pines 4, 5 y 7 del mismo.

4. Fuente de alimentación del nodo esclavo 1

Conector de cuatro pines correspondientes a GND, 12V, -12V y 5V. La señal de 5V es proporcionada por el regulador mencionado anteriormente. Estas señales viajan hacia el nodo 1 a través del cable UTP y por medio de los pines 4, 5, 7 y 8 del mismo.

5. Fuente de alimentación del nodo esclavo

Conector de cuatro pines correspondientes a GND, 12V, -12V y 5V. Al igual que para el nodo 1 la señal de 5V es proporcionada por el regulador. Todas las señales viajan hacia el nodo 2 a través del cable UTP y por medio de los pines 4, 5, 7 y 8 del mismo.

6. Terminadores de red

Como se puede observar en la Figura 40 los terminadores de red son dos resistencias de 120Ω .

7. Conexión nodo maestro

A este punto llegan las señales CAN_H y CAN_L provenientes del nodo maestro a través del cable UTP y por medio de los pines 1 y 2 respectivamente.

8. Conexión nodo esclavo 1

A este punto llegan las señales CAN_H y CAN_L provenientes del nodo esclavo 1 a través del cable UTP y por medio de los pines 1 y 2 respectivamente.

9. Conexión nodo esclavo

A este punto llegan las señales CAN_H y CAN_L provenientes del nodo esclavo 2 a través del cable UTP y por medio de los pines 1 y 2 respectivamente.

6 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA SCADA

En este capítulo se describe la configuración del sistema SCADA, programación en lenguaje “*ladder*” del PLC Micrologix1500 usando el software RSLogix 500 y desarrollo de los HMI’s (Human –Machine Interface) en RSView32 7.0.

6.1 SISTEMA SCADA

El sistema SCADA permite gestionar los datos de la red; las variables sensadas son adquiridas por el nodo maestro y son llevadas hacia el PLC vía RS-232 y a su vez hacia el sistema SCADA. Este mismo procedimiento es llevado a cabo cuando las señales de control van a ser enviadas hacia los actuadores. La arquitectura del sistema integrado es mostrada en la Figura 41.

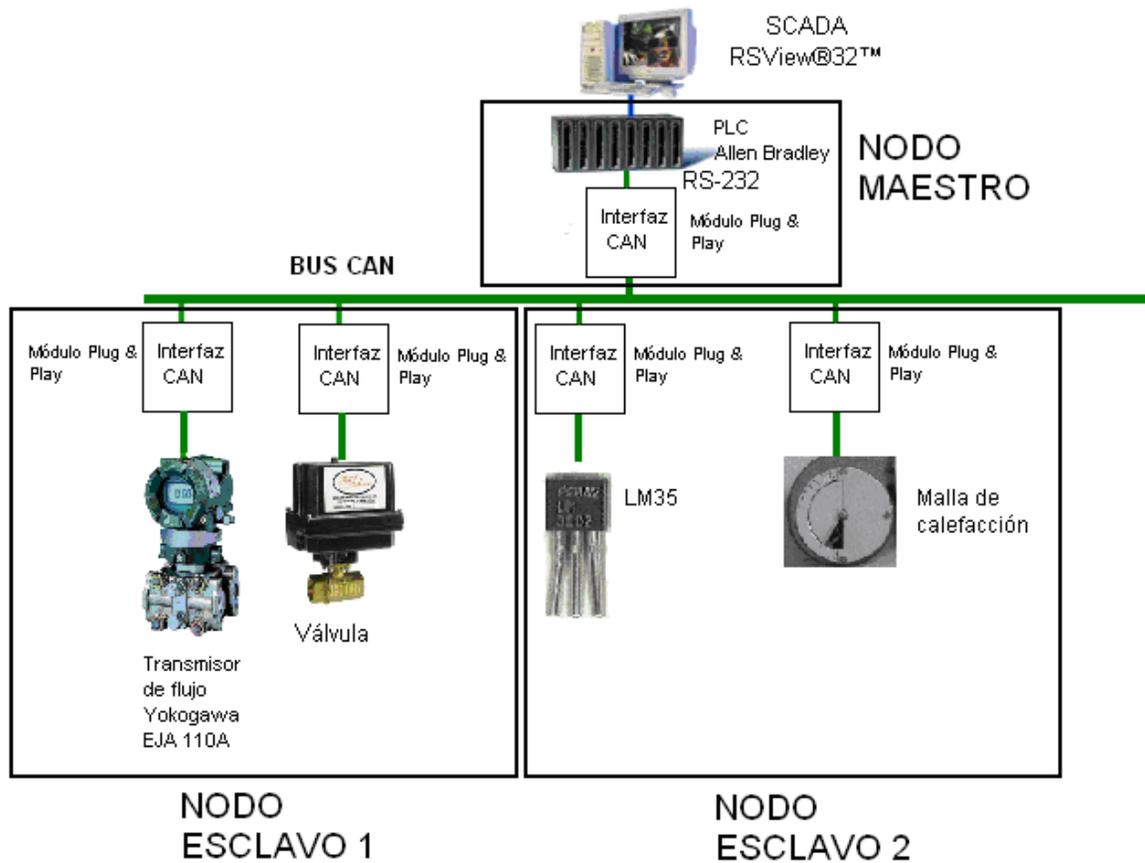


Figura 41. Arquitectura de la red

6.2 PROGRAMACIÓN PLC MICROLOGIX 1500

La programación del PLC se realiza mediante el software RSLogix 500 en lenguaje “*ladder*”. A continuación se describe detalladamente la forma de programación del mismo, describiendo escalón por escalón las funciones usadas, iniciando con la configuración del canal de comunicación por el cual se realizará el intercambio bidireccional de información entre el nodo maestro y el PLC, mediante el canal 1 de este último.

Una vez se ha abierto la ventana de configuración del canal en el software RSLogix500 la siguiente ventana es desplegada (Figura 42):

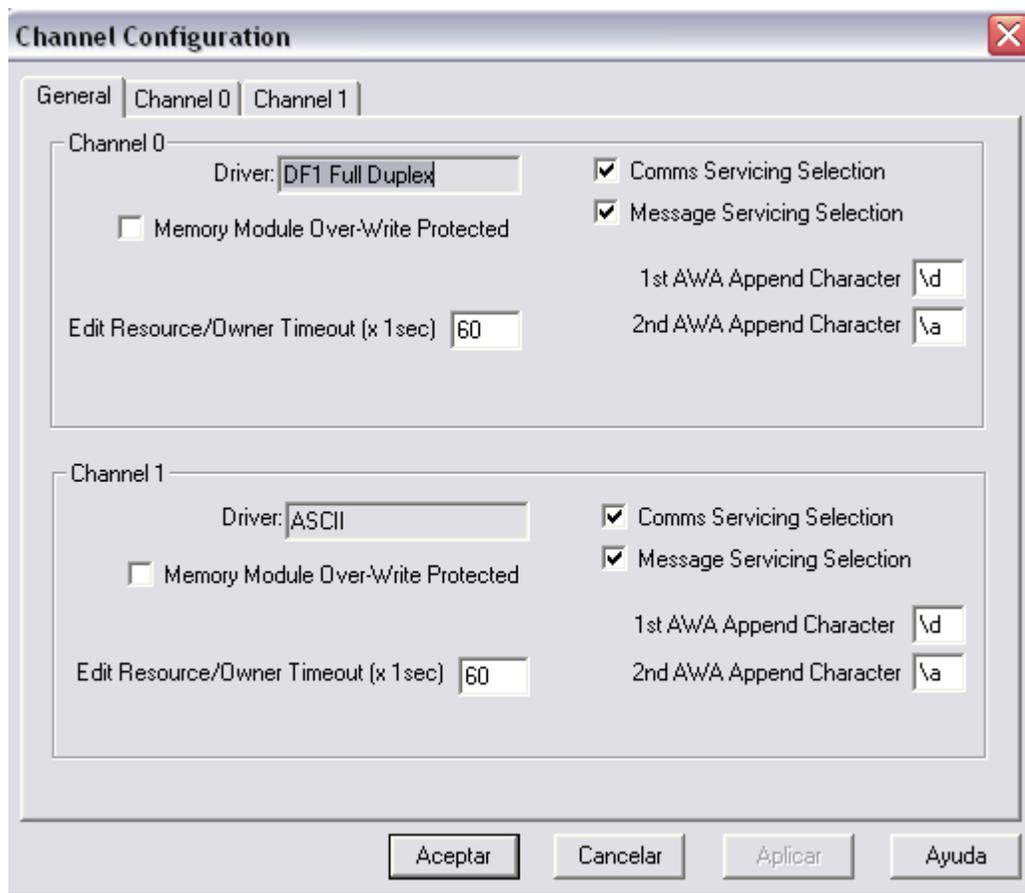


Figura 42. Ventana de configuración del canal

Se procede a configurar el canal 1 teniendo como referencia la Figura 43, donde se selecciona, en el campo “*Driver*”, la opción “*ASCII*”, en el campo “*Baud*”, la velocidad de comunicación en bits/seg, y en “*Parity*”, el bit de paridad [13].

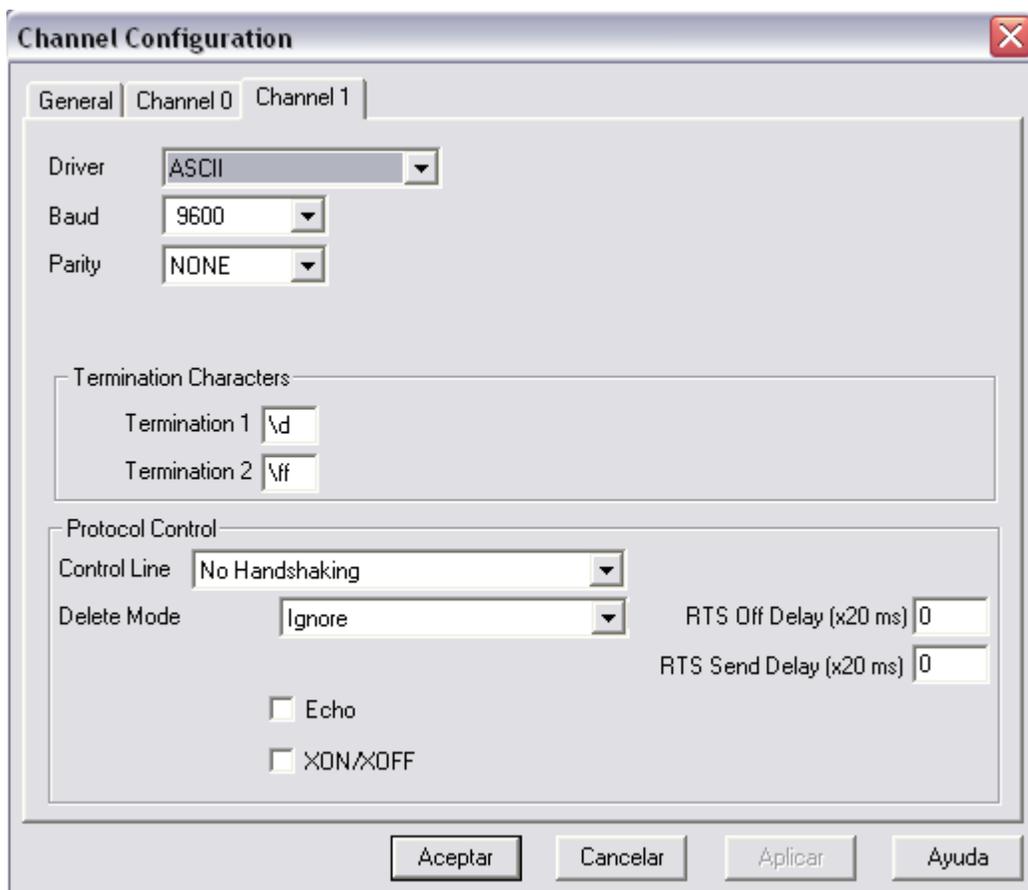
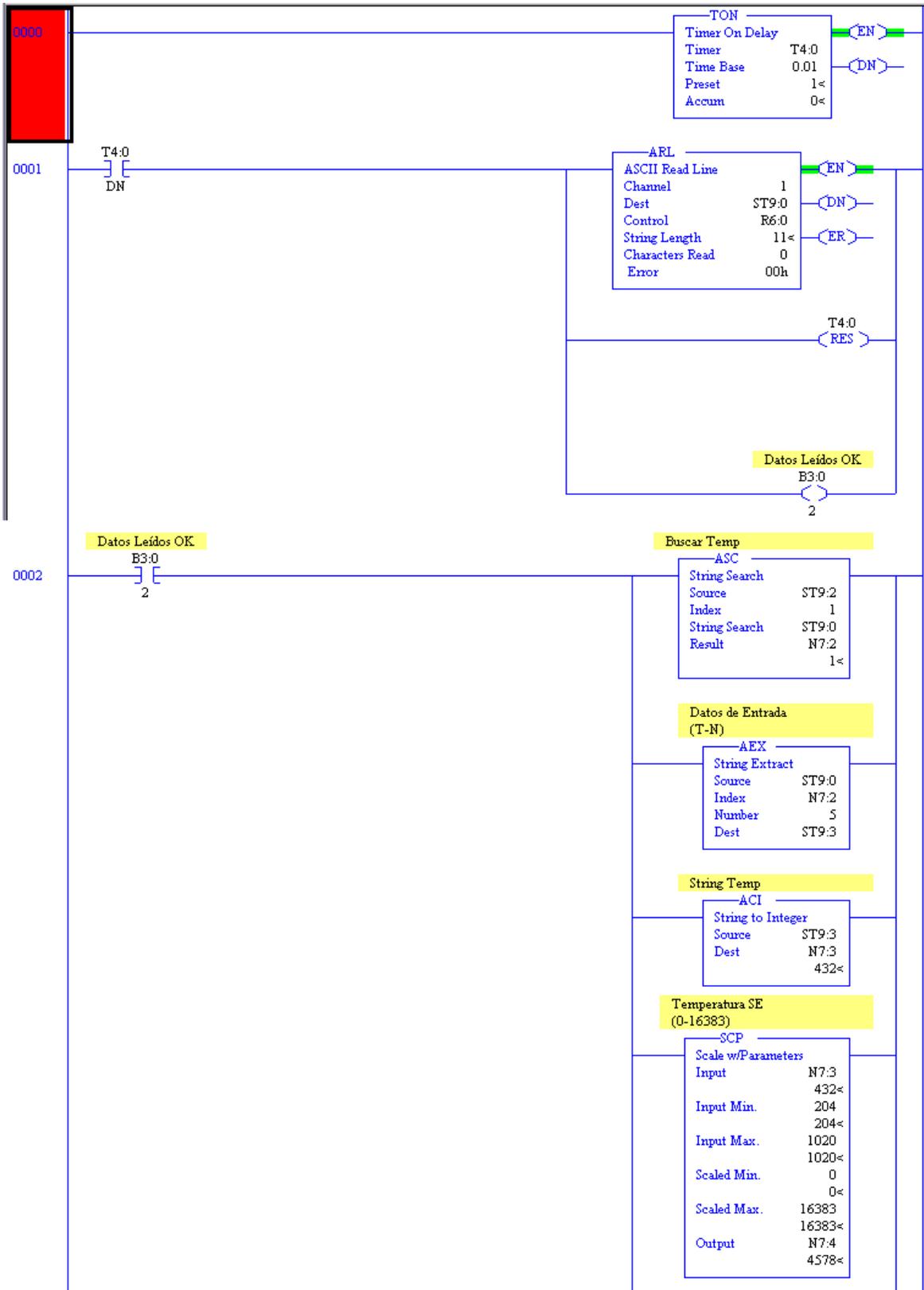


Figura 43. Ventana de configuración del canal 1

Configurando el canal se comienza con la programación “*ladder*”. Cada escalón puede ser visualizado en la Figura 44



Medición de
Temperatura en °C

SCP

| | |
|--------------------|--------|
| Scale w/Parameters | N7:4 |
| Input | 4578< |
| Input Min. | 0 |
| | 0< |
| Input Max. | 16383 |
| | 16383< |
| Scaled Min. | 0 |
| | 0< |
| Scaled Max. | 100 |
| | 100< |
| Output | N7:13 |
| | 28< |

Buscar Nivel

ASC

| | |
|---------------|-------|
| String Search | |
| Source | ST9:4 |
| Index | 1 |
| String Search | ST9:0 |
| Result | N7:0 |
| | 6< |

Datos de Entrada
(T-N)

AEX

| | |
|----------------|-------|
| String Extract | |
| Source | ST9:0 |
| Index | N7:0 |
| Number | 5 |
| Dest | ST9:5 |

String Nivel

ACI

| | |
|-------------------|-------|
| String to Integer | |
| Source | ST9:5 |
| Dest | N7:1 |
| | 558< |

Nivel SE
(0-16383)

SCP

| | |
|--------------------|--------|
| Scale w/Parameters | N7:1 |
| Input | 558< |
| Input Min. | 180 |
| | 180< |
| Input Max. | 1020 |
| | 1020< |
| Scaled Min. | 0 |
| | 0< |
| Scaled Max. | 16383 |
| | 16383< |
| Output | N7:5 |
| | 7372< |

Medicion Nivel en cm

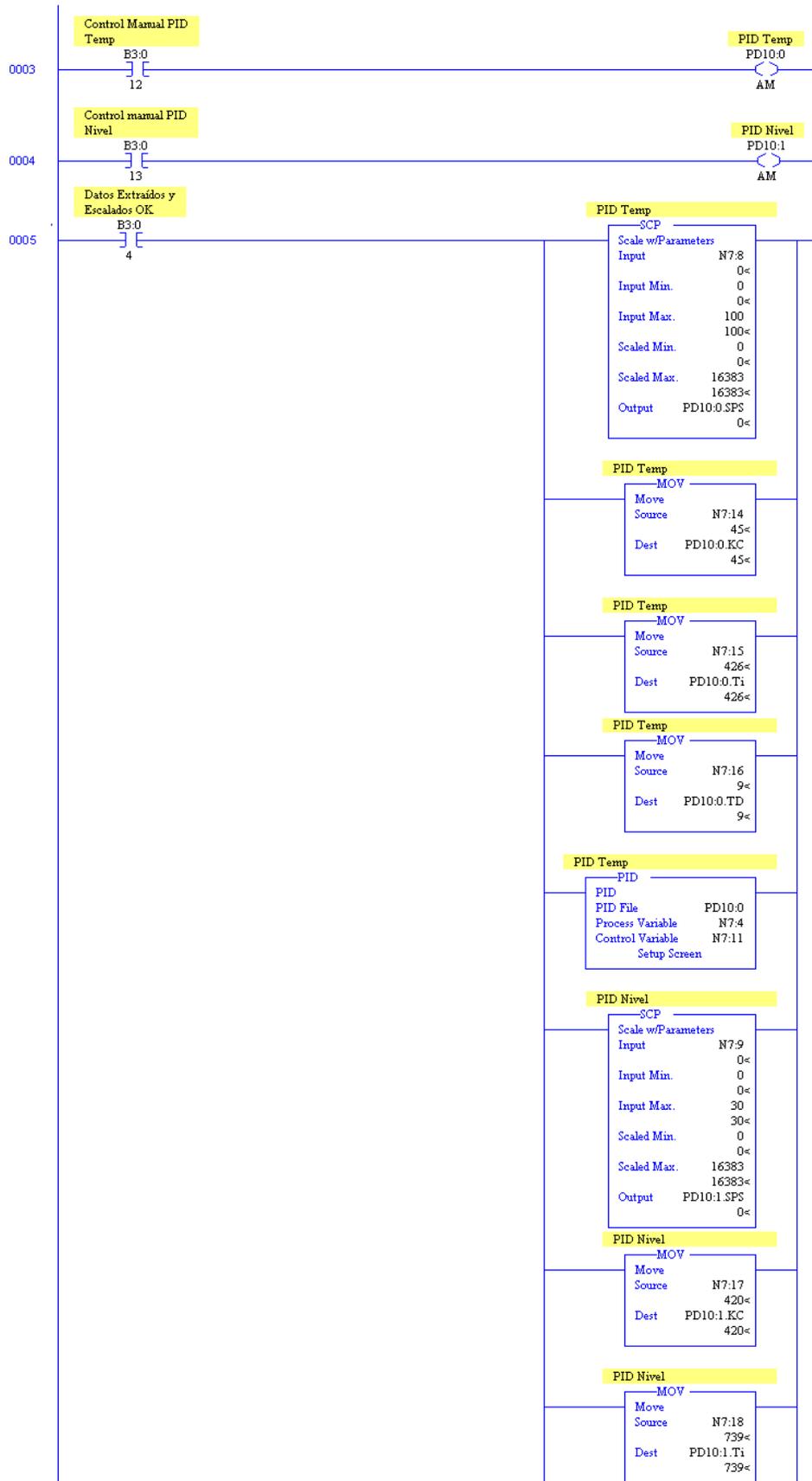
SCP

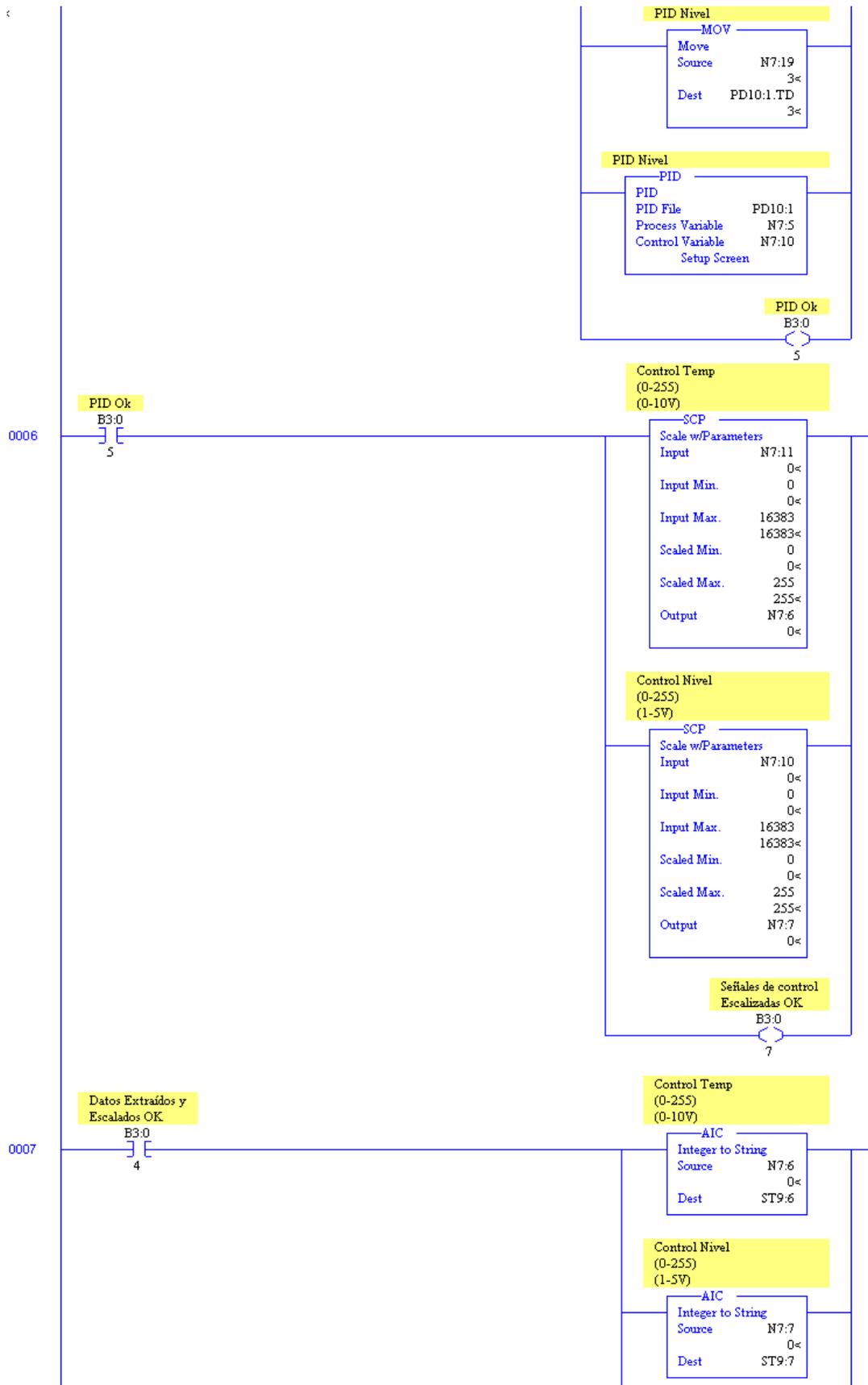
| | |
|--------------------|--------|
| Scale w/Parameters | N7:5 |
| Input | 7372< |
| Input Min. | 0 |
| | 0< |
| Input Max. | 16383 |
| | 16383< |
| Scaled Min. | 0 |
| | 0< |
| Scaled Max. | 30 |
| | 30< |
| Output | N7:12 |
| | 13< |

Datos Extraídos y
Escalados OK

B3:0

4





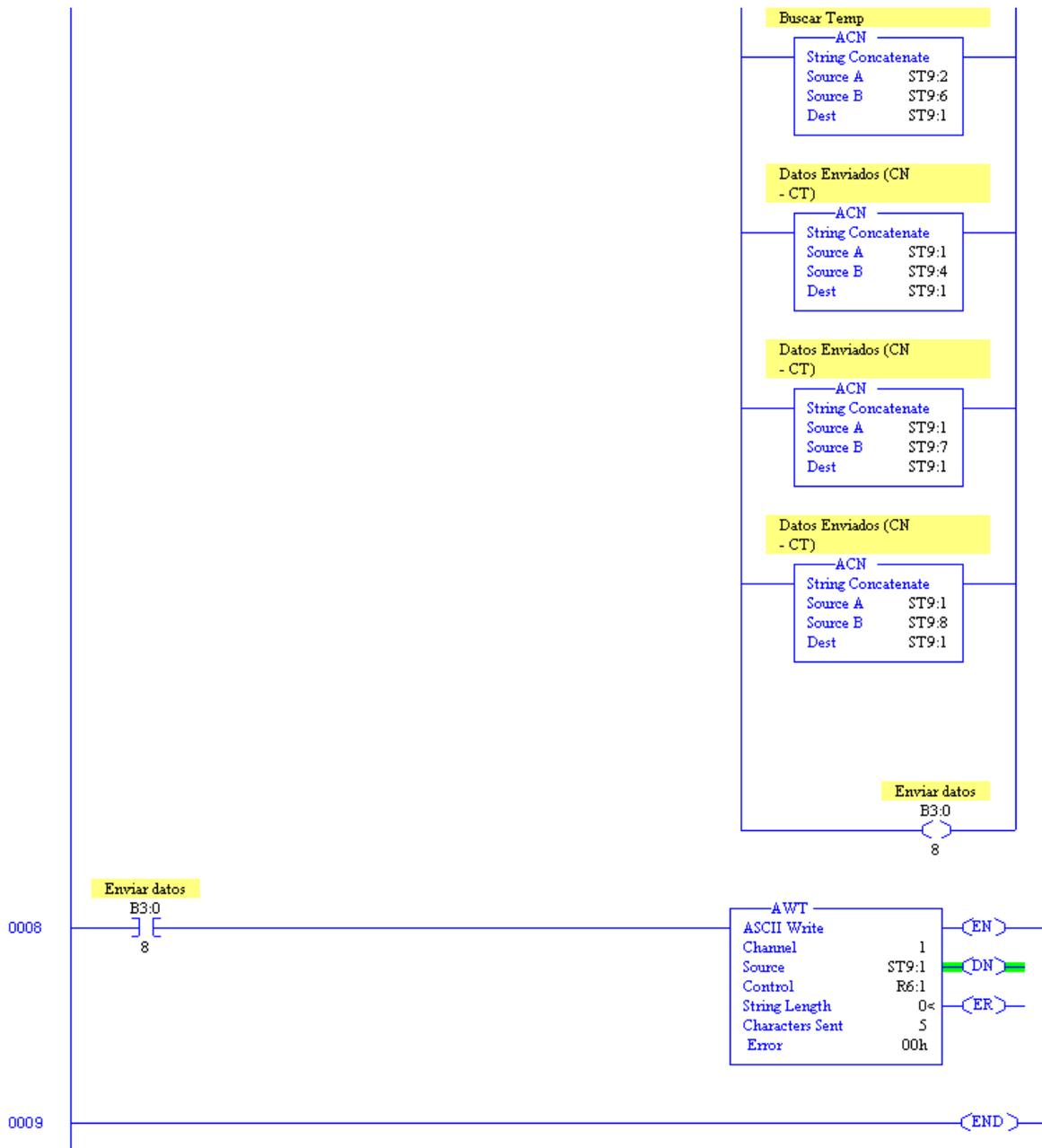


Figura 44. Programa en lenguaje “*ladder*”

Escalón 0000: Un temporizador se activa y empieza su conteo hasta 1 segundo. El sistema adquiere los datos de las variables del proceso, se considera que este tiempo de adquisición es suficiente debido a los requerimientos de los procesos por controlar.

Escalón 0001: En el momento en que el temporizador acaba su conteo el bit *DN* es activado, y se ejecuta el bloque *ARL*, cuya función es adquirir los datos de las variables de los procesos vía RS-232 que provienen del PIC. Esta función lee lo que se encuentra

almacenado en el buffer de recepción RS-232 del PLC [14]. Así mismo se resetea el temporizador para que arranque otra vez su conteo y se activa una bandera llamada *Datos Leídos OK*.

Escalón 0002: Con la bandera *Datos Leídos OK* se procede a separar los datos de las variables de los procesos (nivel y temperatura), ya que ambas variables son enviadas por el PIC como un solo dato ("*string*"); por lo tanto el PLC adquiere un solo dato donde vienen concatenadas las dos señales. Los datos son enviados por el PIC hacia el PLC como un solo dato con el siguiente formato TXXXXNXXXX donde el primer dígito T significa que los siguientes cuatro dígitos XXXX pertenecen a la variable temperatura y el quinto dígito N que los siguientes cuatro dígitos XXXX pertenecen a la variable nivel, por tal razón es necesario la extracción de cada dato para así procesar cada señal por separado. La razón de los cuatro dígitos es debido a que el conversor A/D que posee el PIC es de 10 bits por lo tanto el mayor valor que se tiene es 1024. El bloque *ASC (Buscar Temp)* busca el dato temperatura; una vez la cadena de datos es encontrada se procede a extraer el dato de temperatura (TXXXX), es decir, los cuatro dígitos siguientes al dígito T; esto se realiza con el bloque *AEX*. Luego convertimos esa cadena de 4 dígitos de la señal de temperatura a un número entero debido a que se recibe en formato "*string*". Posteriormente se realiza la escalación del dato de temperatura obtenido. Este mismo procedimiento se realiza para encontrar y extraer el dato de nivel. Al final del escalón se activa una bandera denominada *Datos Extraídos y Escalados OK*, que es usada en el siguiente escalón.

Escalón 0003: En este escalón se fija una bandera con el fin de pasar del modo de control automático al modo de control manual del PID encargado del proceso de temperatura.

Escalón 0004: En este escalón se fija una bandera con el fin de pasar del modo de control automático al modo de control manual del PID encargado del proceso de nivel.

Escalón 0005: Cuando la bandera *Datos Extraídos y Escalados OK* se activa quiere decir que cada variable del proceso se tiene en forma independiente, y ya pueden ser usadas una independiente de la otra. Esto con el fin de que cada variable sea usada por un bloque PID. En este escalón se realiza la implementación del controlador PID, se tienen los valores del Set Point y de las constantes (K_c , T_i y T_d) del controlador que son

insertadas desde el sistema supervisorio, y los bloques PID's que realizan el control de cada proceso. Se activa una bandera llamada *PID OK*.

Escalón 0006: Las señales de control que van a ser enviadas hacia los actuadores son escaladas. Una bandera de *Señales de Control Escaladas OK* es activada.

Escalón 0007: Se procede a realizar el procedimiento contrario al que se llevó a cabo en el escalón 0002. Se concatenarán en un solo dato y usando el mismo formato mencionado en el escalón 0002 (TXXXXNXXXX) las señales de control que van a ser recibidas por el nodo maestro y que posteriormente serán enviadas a los nodos esclavos. En primer lugar se convierte la señal de control que va hacia la planta de temperatura en una cadena, al igual que la señal de control que va hacia la planta de nivel. Posteriormente se concatena la señal de control de temperatura con la letra T para ir formando el formato de envío del dato, con lo cual se tendría TXXXX. Después se concatena la letra N con la cadena formada anteriormente, por lo tanto se tiene lo siguiente TXXXXN. Luego concatenamos la señal de control de nivel a la cadena formada anteriormente con lo cual se llega a TXXXXNXXXX que es el formato que se desea enviar hacia el nodo maestro. Adicionalmente al formato se le adiciona un terminador de línea denotado con la letra F para que el nodo maestro esté en condiciones de determinar cuándo ha terminado el envío de datos desde el PLC. Se activa una bandera denominada *Señales de Control OK*.

Escalón 0008: Se envían los datos hacia el nodo maestro por medio del protocolo RS-232 usando el bloque AWT.

6.3 HMI's (Human Machine Interface)

La aplicación está compuesta por 5 ventanas principales, las cuales son descritas a continuación:

- Menú de ingreso.
- Menú de opciones del usuario.
- Procesos.
- Pantalla del proceso de temperatura.

- Pantalla del proceso de nivel.

6.3.1 Menú de Ingreso

La pantalla del menú de ingreso es mostrada en la Figura 45 y es la primera pantalla que se despliega cuando se pone en marcha el sistema. El menú ingreso se diseña para que la aplicación tenga cierto grado de seguridad. Se definen varios usuarios para la aplicación; dependiendo del tipo de usuario cada uno tendrá determinados privilegios sobre la aplicación. En la aplicación se ha definido dos usuarios, un supervisor y un operario. El supervisor tiene más privilegios sobre el sistema que el operario; más adelante se detalla sobre los privilegios de cada uno. Para ingresar en la aplicación es necesario hacer un “click” en el botón “Login”; al presionarlo la ventana de ingreso de usuario es desplegada (Figura 46). En el ítem “User” se escribe el “login” que ha sido asignado al usuario que ingresará en la aplicación y en el ítem “Password”, la clave que ha sido asignada al respectivo usuario. Si el “login” y/o el “Password” son incorrectos un mensaje de error se despliega (Figura 47).



Figura 45. Ventana del menú de ingreso

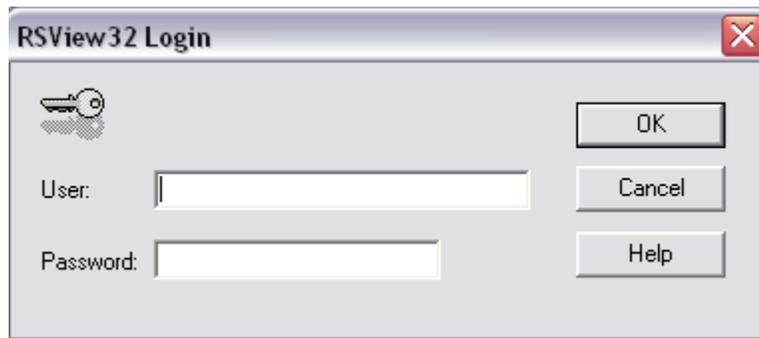


Figura 46. Ventana de confirmación del “User” y el “Password”



Figura 47. Ventana del “User” y/o “Password” incorrecto

Si se desea salir del sistema se oprime el botón “Salir” (Figura 45).

6.3.2 Menú de Opciones del Usuario

Una vez el usuario se ha registrado correctamente en la aplicación la ventana de opciones de usuario se despliega (Figura 48).

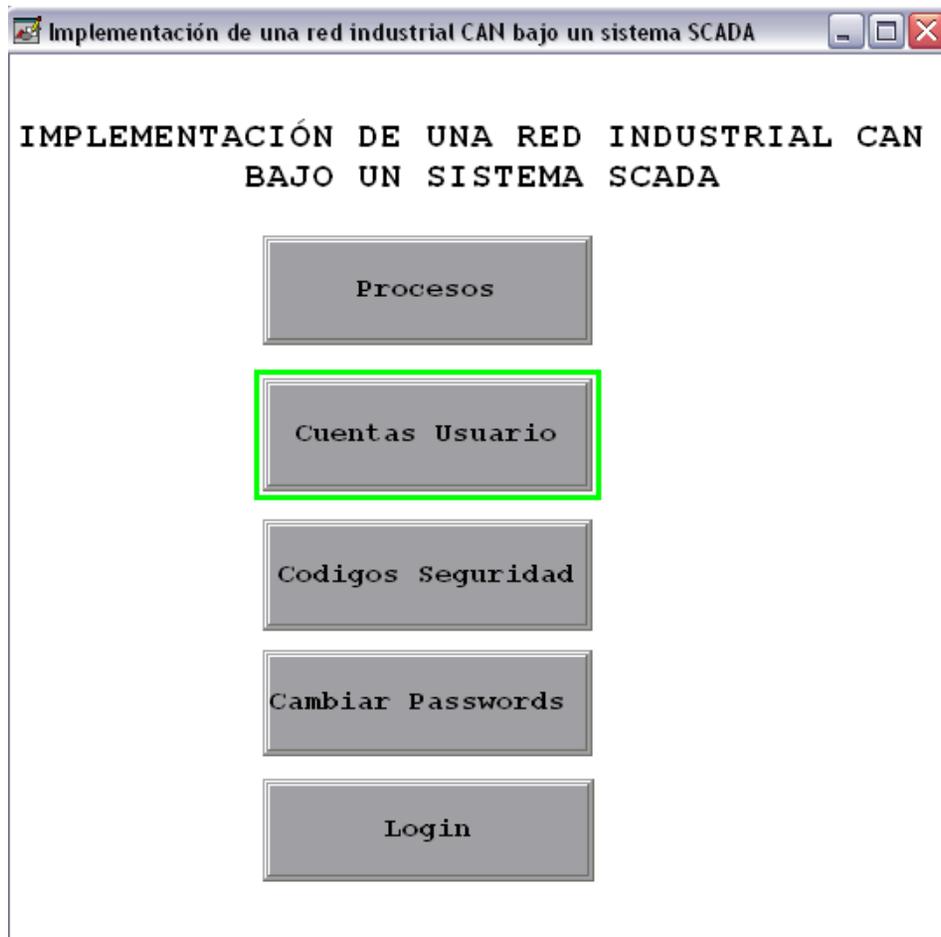


Figura 48. Ventana de opciones del usuario

El primer botón "*Procesos*" permite ingresar a la pantalla de los procesos o menú principal que se explica en el siguiente apartado.

El botón "*Cuentas Usuario*" solo puede ser accedido por el supervisor del sistema permitiéndole agregar usuarios a la aplicación o eliminarlos de la misma. La Figura 49 muestra la ventana de cuentas de usuario.

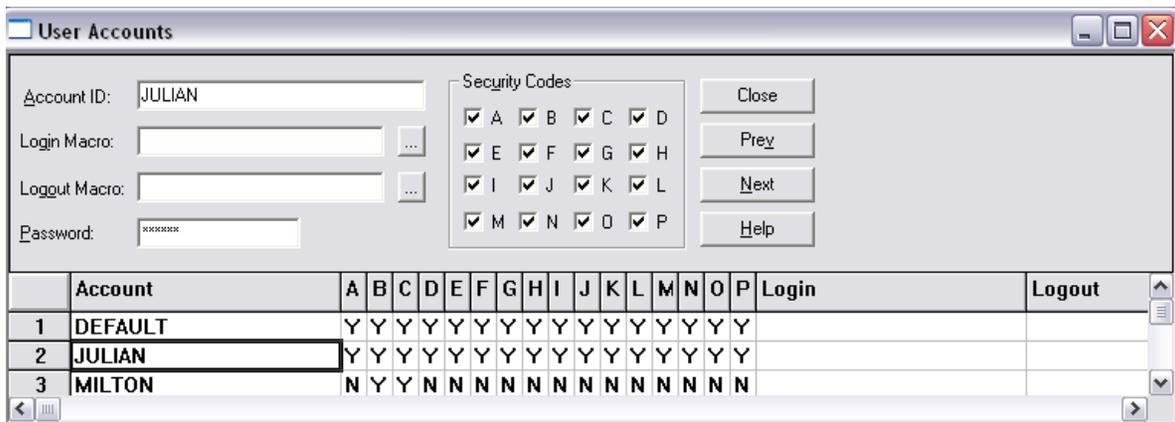


Figura 49. Ventana de cuentas de usuario

En el ítem “*Account ID*” se define el “*login*” del usuario y en el ítem “*Password*” se define la clave de acceso del mismo. Los “*Security Codes*” permiten definir los privilegios de cada usuario.

El botón “*Códigos Seguridad*” permite definir qué funciones o comandos van a estar restringidos en la aplicación. Al igual que el botón “*Cuentas Usuario*”, este botón sólo puede ser accedido por el supervisor.

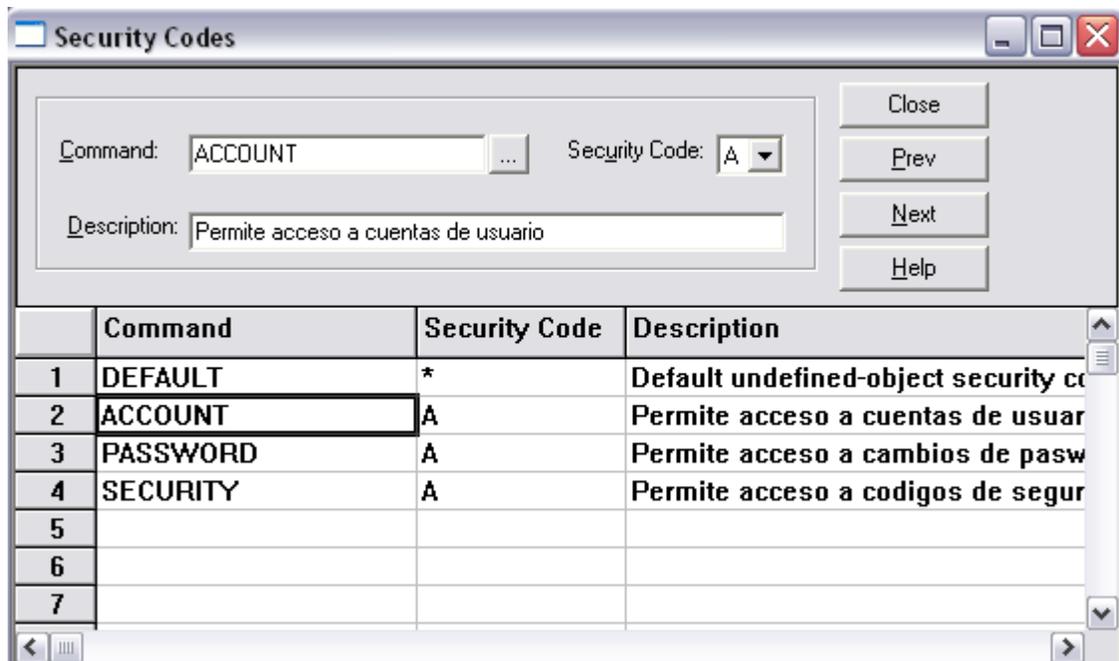


Figura 50. Ventana de códigos de seguridad

El botón “*Cambiar Passwords*” puede ser usado tanto por el supervisor como por el operario. Permite realizar un cambio del “*Password*” del usuario que ha ingresado en la aplicación.

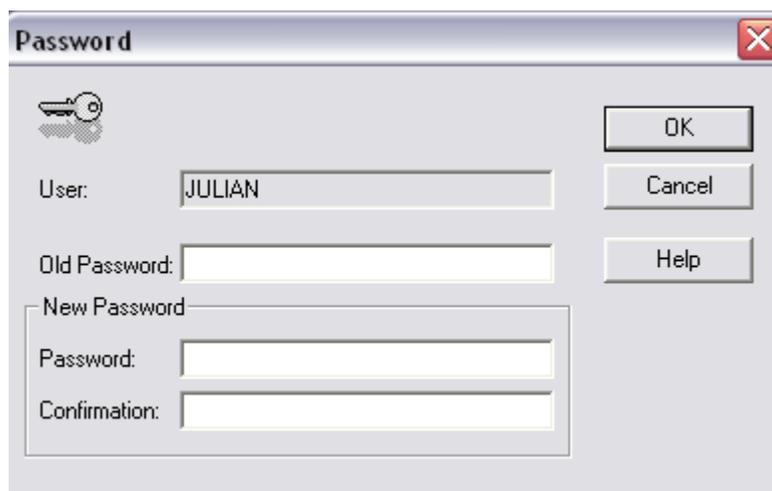
The image shows a standard Windows-style dialog box titled "Password". In the top-left corner, there is a small icon of a key. Below the icon, the text "User:" is followed by a text input field containing the name "JULIAN". Underneath that, the text "Old Password:" is followed by an empty text input field. A larger rectangular box labeled "New Password" contains two sub-fields: "Password:" and "Confirmation:", each followed by an empty text input field. On the right side of the dialog, there are three buttons stacked vertically: "OK", "Cancel", and "Help". The dialog has a standard Windows window border with a close button (X) in the top-right corner.

Figura 51. Ventana de cambio de “*Password*”

El botón “*Login*” permite regresar al menú de ingreso.

Para información más detallada acerca de la configuración del sistema de seguridad de RSVIEW32™ dirigirse a [15] (pagina 277).

6.3.3 Menú Principal

Es desplegado cuando el usuario oprime el botón “*Procesos*” ubicado en el menú opciones usuario. En la pantalla de menú principal, que puede ser vista en la Figura 52, se tiene 4 botones que tienen la siguiente función:

- Planta de nivel: Permite ingresar a la ventana del proceso de nivel.
- Planta de temperatura: Permite ingresar a la ventana del proceso de temperatura.
- Opciones usuarios: Permite devolverse al menú de opciones del usuario.

RED INDUSTRIAL CAN

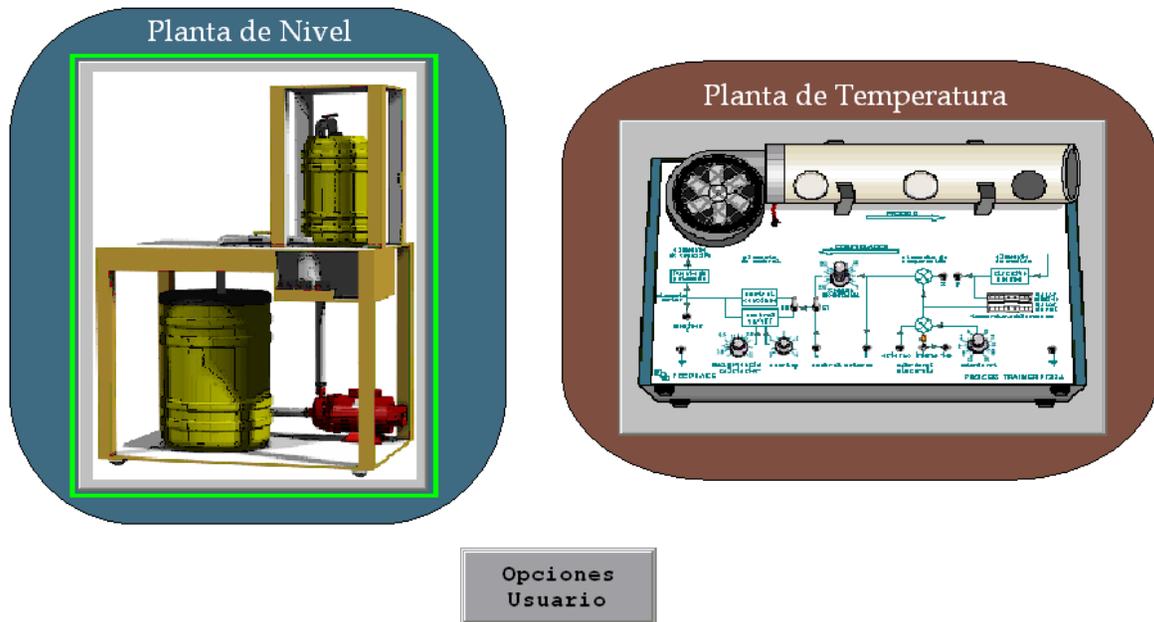


Figura 52. Ventana menú principal

6.3.4 Pantalla del Proceso de Nivel y Flujo de Agua

Esta pantalla muestra al usuario un mímico del proceso de nivel y flujo de agua (Figura 53), donde se visualiza en un indicador el valor de la variable controlada (nivel), y en otro el valor de la señal de control. Así mismo se tiene controles para configurar el Set Point, las constantes del controlador y el modo de control (Automático/Manual).

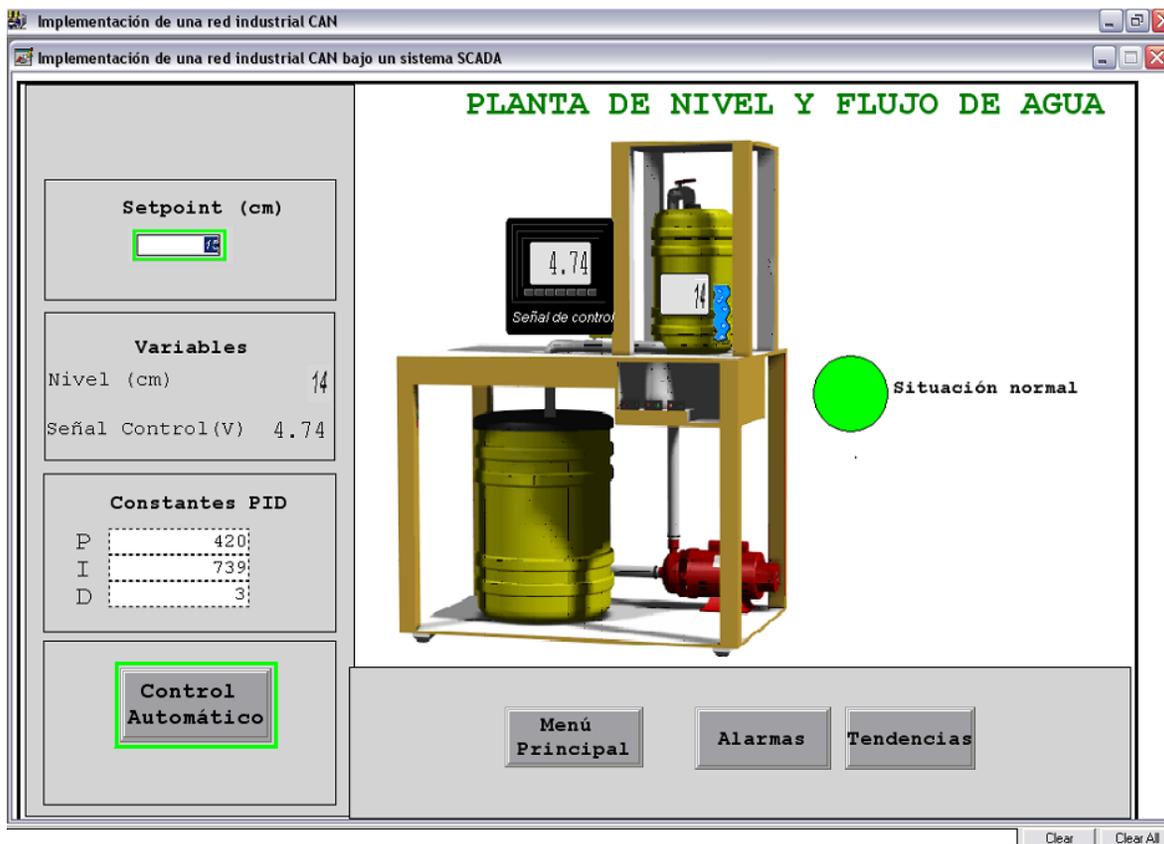


Figura 53. Mímico de la planta de nivel y flujo de agua

El mímico cuenta con 4 botones adicionales, descritos a continuación:

- Menú principal: Permite al usuario volver a la pantalla de menú principal.
- Alarmas: Permite al usuario visualizar las alarmas que se han presentado en el proceso de nivel y flujo de agua. Adicionalmente en el mímico se tiene un indicador encargado de informar al usuario una situación de alarma. Como se ve en la figura anterior el indicador está de color verde junto a un mensaje de “*Situación normal*”, lo que indica que no se ha presentado ninguna alarma en el sistema. Si se presenta una situación de alarma dicho indicador pasa a ser de color rojo y junto a él aparece un mensaje de “*Situación de alarma*” (Ver Figura 54). Al presionar el botón “*Alarmas*” la ventana mostrada en la Figura 55 se despliega. La ventana de alarmas muestra la “*tag*” o variable que entra en condición de alarma, su valor, la hora y la fecha en que se presenta la condición de alarma, la gravedad y una etiqueta de la misma. Tanto en la Figura 54 como en la Figura 55 se puede observar que se presenta una situación de alarma ya que el valor de la variable del proceso (nivel) se encuentra por debajo del límite inferior

que ha sido definido en 10cm. La ventana de alarmas cuenta con el botón *Cerrar* que permite al usuario salir de la misma.

- Tendencias: Gráfica de tiempo real donde se puede visualizar el comportamiento de la variable controlada (nivel) y el Set Point (Figura 56). Posee un botón denominado *Cerrar* que permite al usuario salir de la ventana.
- Control Automático: Permite al usuario cambiar a modo de control manual. Una vez se presione este botón el usuario podrá manipular la variable de control desde un rango de 0 – 100% correspondiendo 0% a un valor de 0.9V y 100% a un valor de 4.86V (Ver Figura 57).

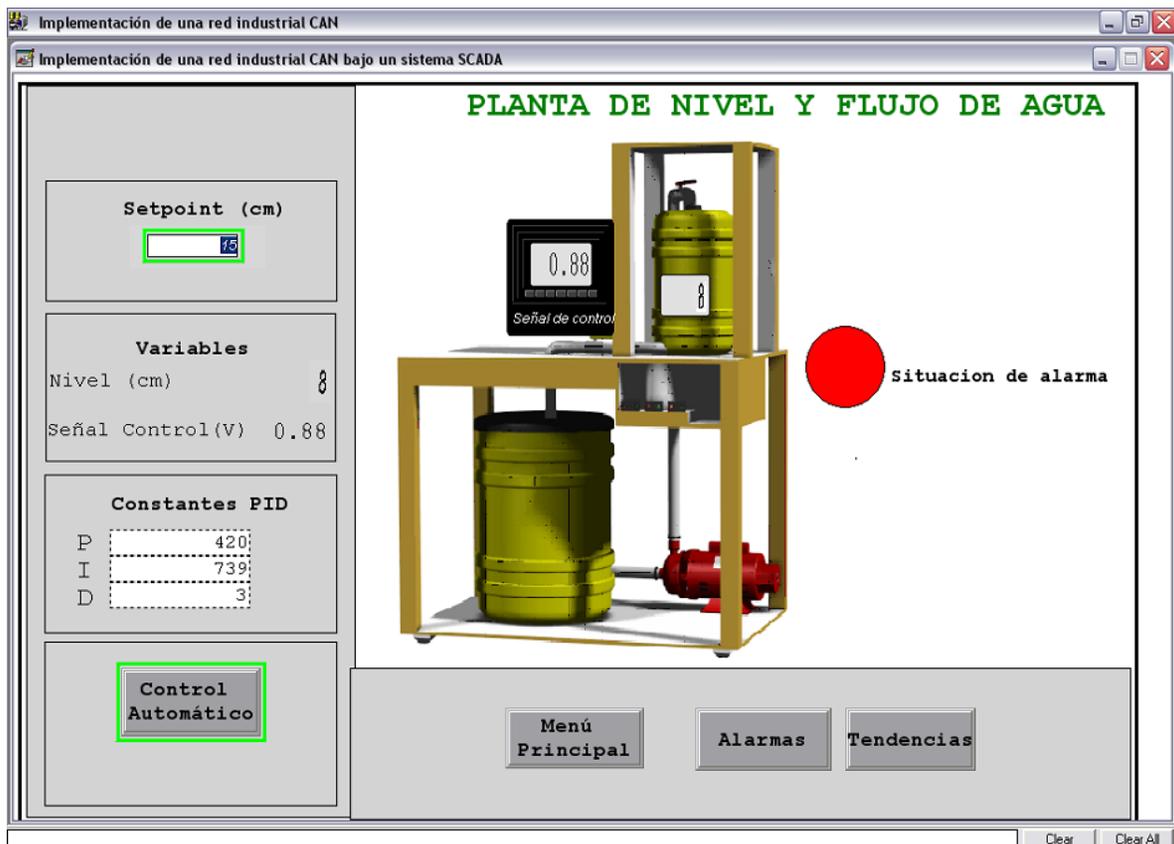


Figura 54. Mímico de una situación de alarma

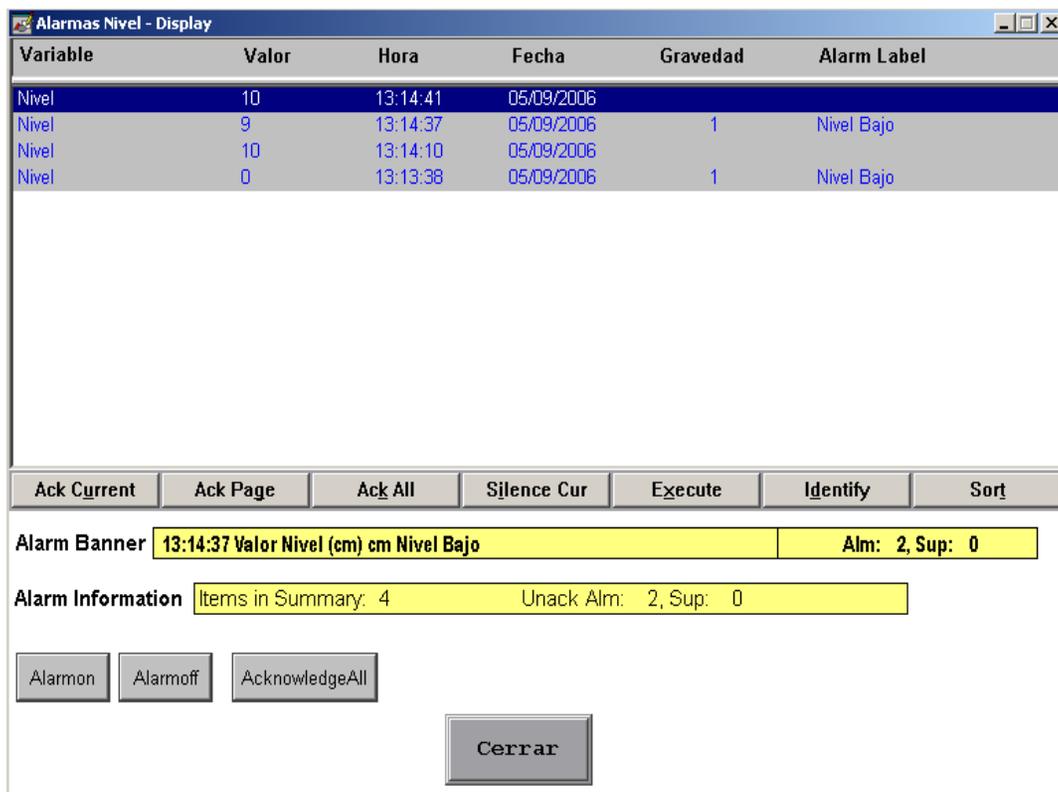


Figura 55. Ventana de alarmas de la planta de nivel y flujo de agua

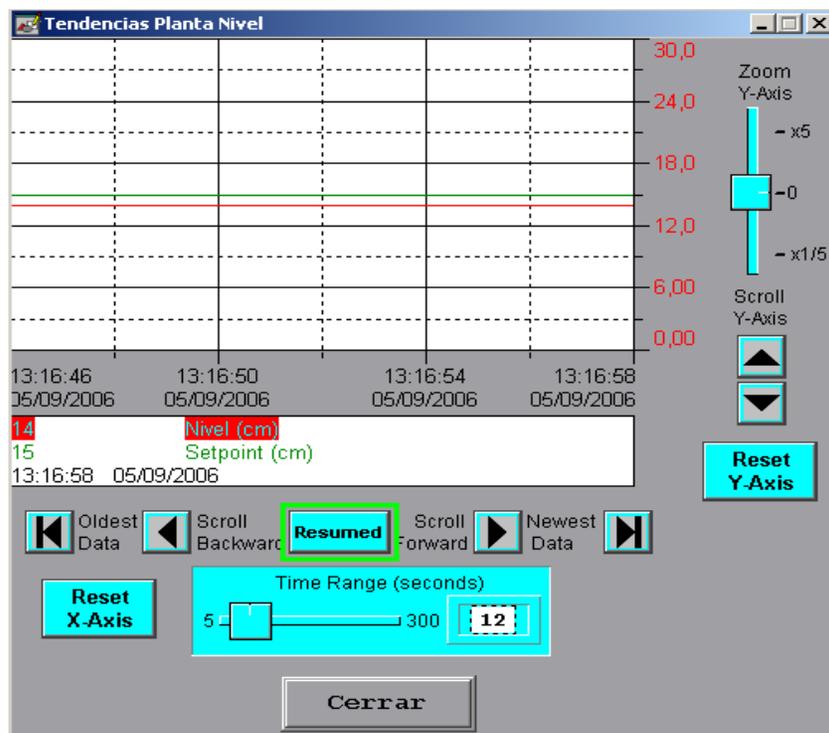


Figura 56. Ventana de tendencias de la planta de nivel y flujo de agua

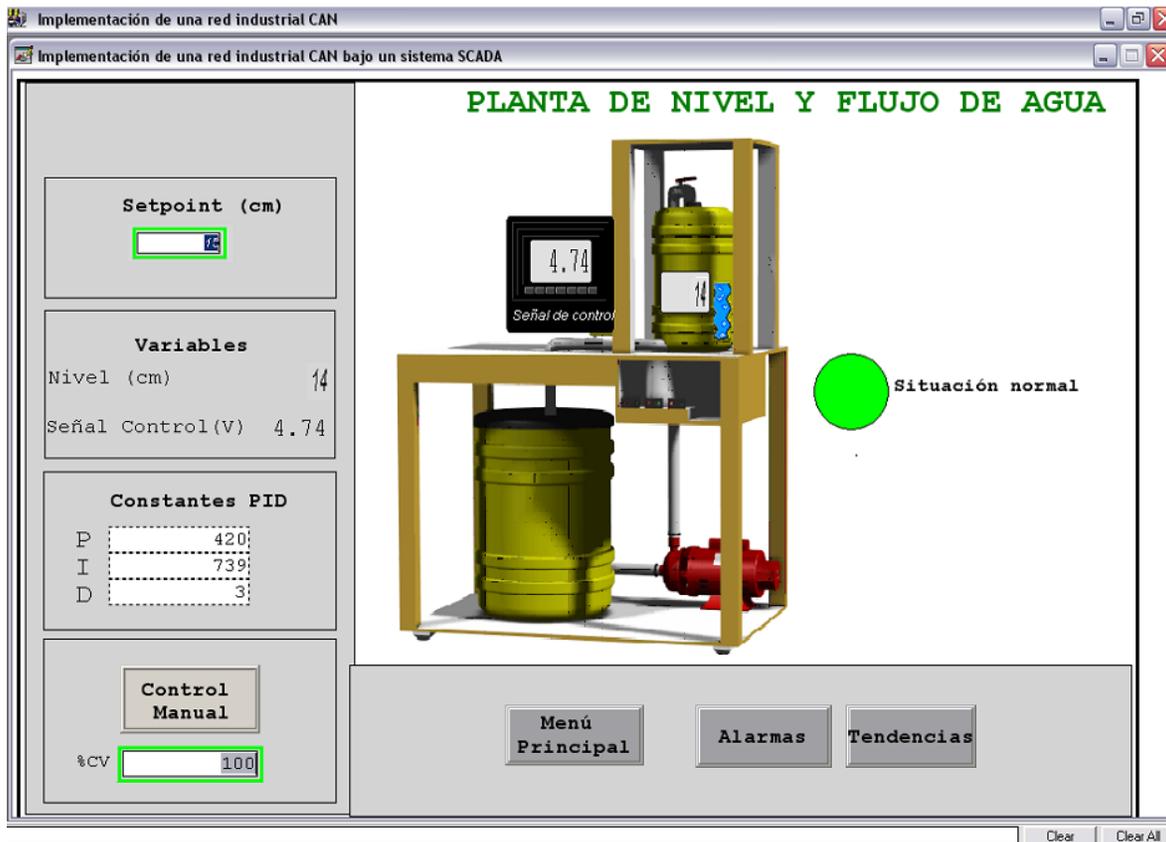


Figura 57. Ventana de control manual de la planta de nivel y flujo de agua

6.3.5 Pantalla del Proceso de Temperatura

Esta pantalla al igual que la pantalla del proceso de nivel muestra al usuario un mímico del proceso de temperatura (Figura 58); se visualiza en un indicador el valor de la variable controlada (temperatura), y en otro el valor de la señal de control. Se tiene controles para configurar el Set Point, las constantes del controlador y el modo de control (Automático/Manual).

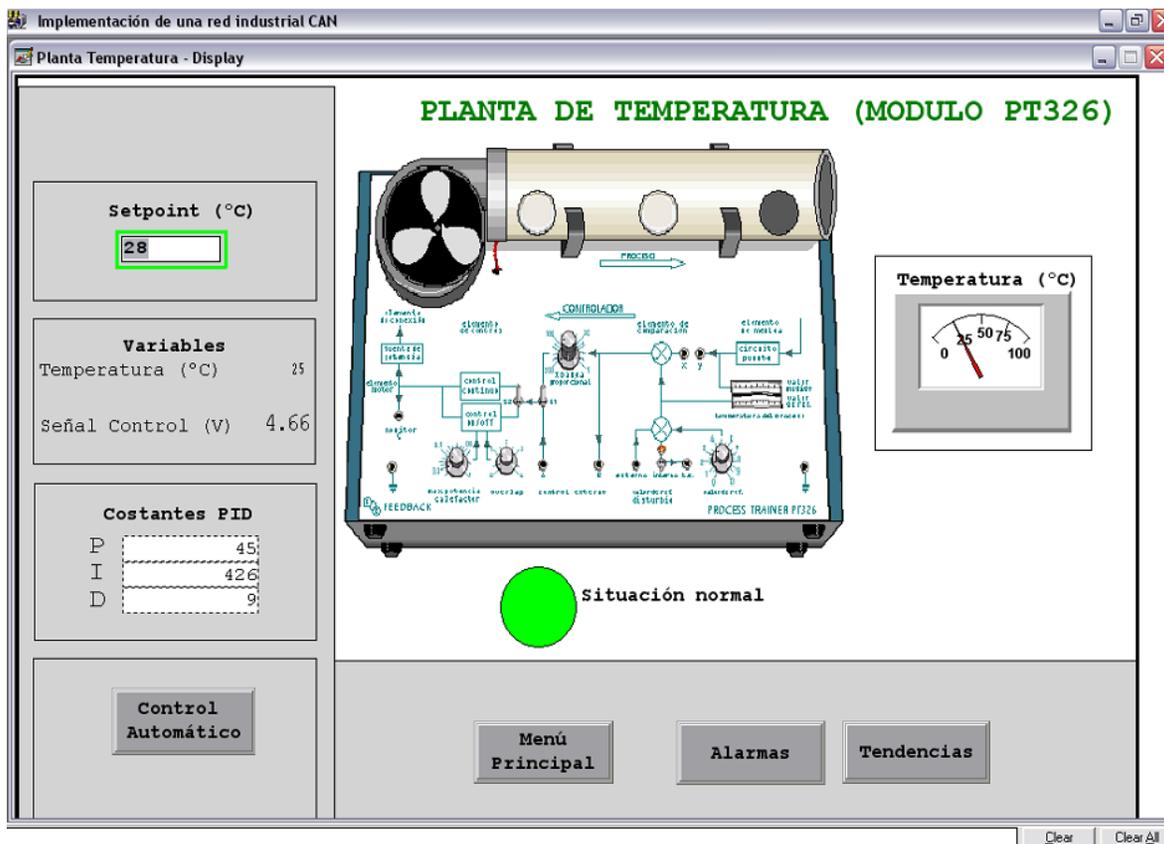


Figura 58. Mímico de la planta de temperatura

Este mímico posee 4 botones iguales a los de la pantalla del proceso de nivel.

- Menú principal: Permite al usuario volver a la pantalla de menú principal.
- Alarmas: Permite al usuario visualizar las alarmas que se han presentado en el proceso de temperatura (Figura 59). Tanto el indicador de alarma como la ventana de alarmas son iguales a los descritos en el proceso de nivel.
- Tendencias: Gráfica de tiempo real donde se puede visualizar la variable controlada (temperatura) y el Set Point (Figura 60). Posee un botón denominado *Cerrar* que permite al usuario salir de la ventana.
- Control Automático: Permite al usuario cambiar a modo de control manual. Una vez se presione este botón el usuario podrá manipular la variable de control desde un rango de 0 – 100% correspondiendo 0% a un valor de 0V y 100% a un valor de 10.06V (Ver Figura 61).

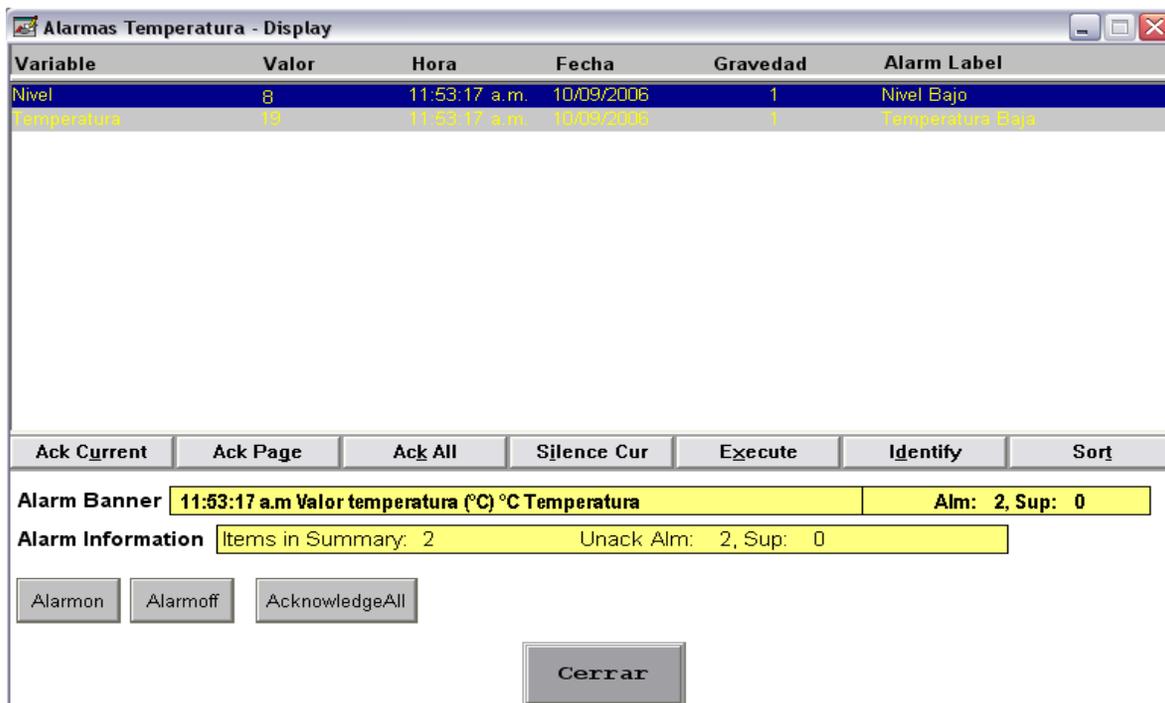


Figura 59. Ventana de alarmas de la planta de temperatura

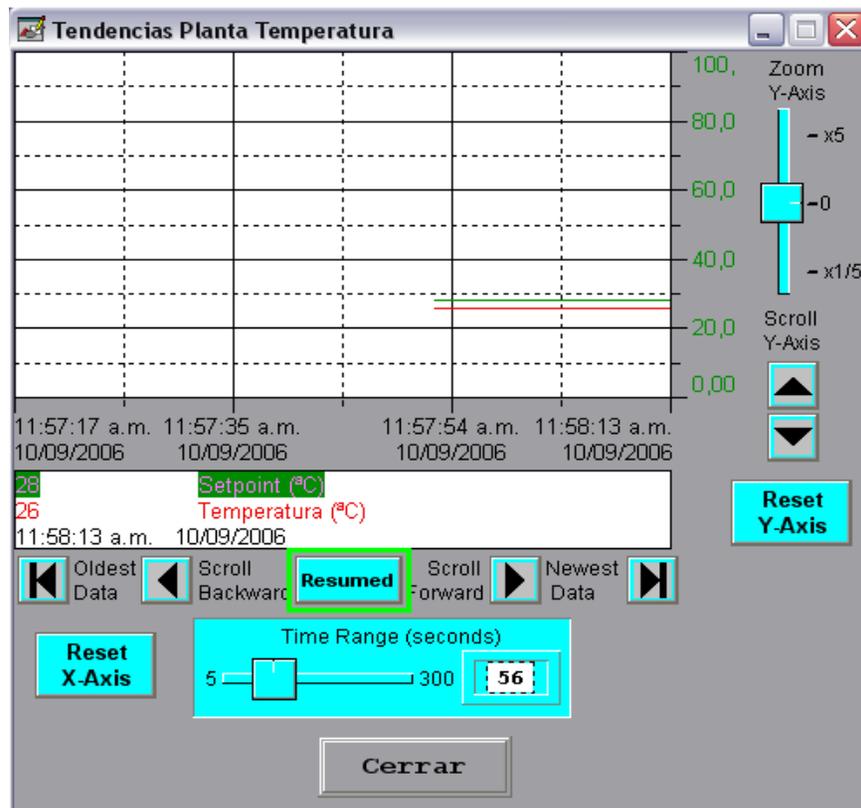


Figura 60. Ventana de tendencias de la planta de temperatura

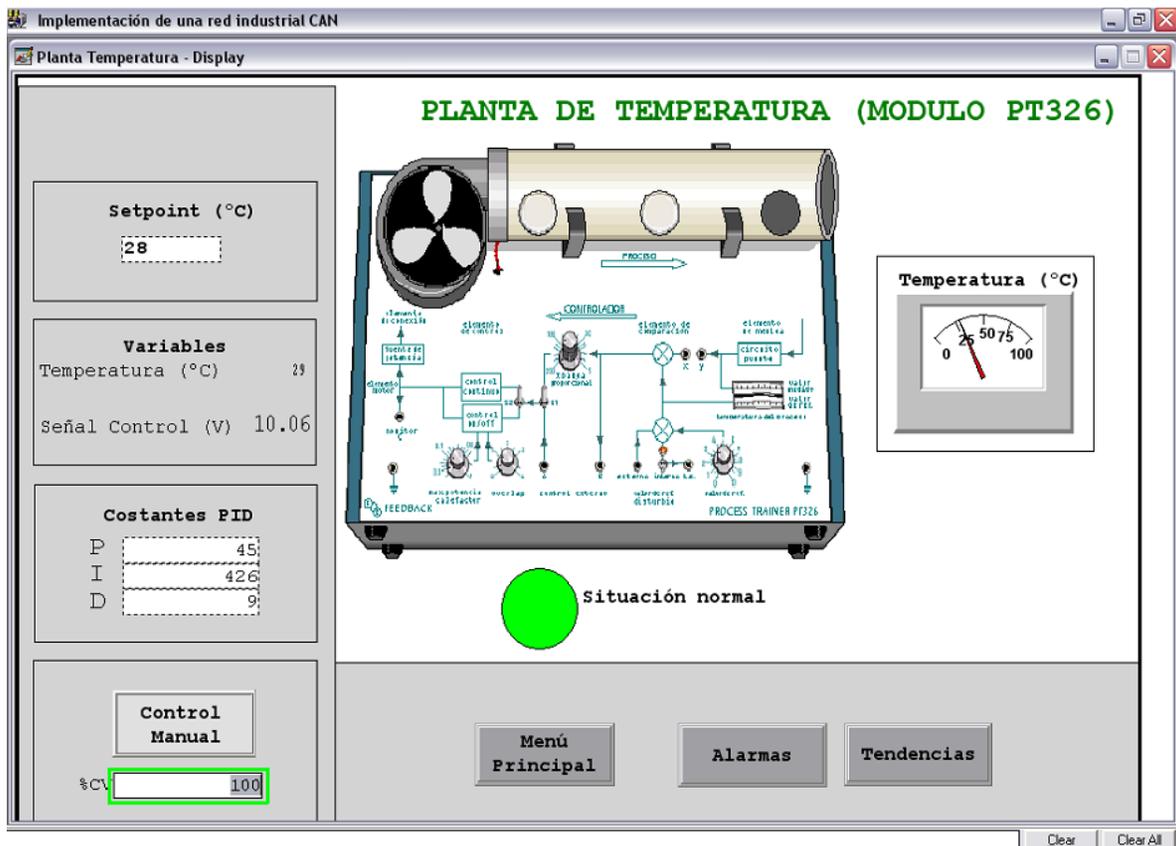


Figura 61. Ventana de control manual de la planta de temperatura

7 CONCLUSIONES

- La realización del proyecto *Implementación de una Red Industrial CAN Bajo un Sistema SCADA* permite darle solución al problema de interconexión de los dispositivos de campo de diferentes fabricantes mediante el desarrollo de las interfaces CAN, logrando de esta manera el intercambio de información a través del protocolo CAN (Controller Area Network), dando fin a las conocidas “islas automatizadas” y obteniendo un “sistema abierto”, donde pueden ser adicionados diferentes dispositivos de manera sencilla.
- La red implementada permite la integración del nivel de campo con el nivel de control dentro del proyecto de investigación “*Convergencia de las Redes Industriales sobre Sistemas SCADA.*”.
- Se diseño y valido el modelo OSI reducido utilizado en aplicaciones industriales, cumpliendo con los requerimientos de tiempo real necesarios en los procesos industriales.
- Se valido e implemento las características del protocolo CAN, corroborando todas las ventajas y beneficios que ofrece el mismo en aplicaciones industriales.
- Dentro de las ventajas de la implementación de la red industrial CAN para el control y monitoreo de procesos se encuentran la conexión de dispositivos al bus “*Plug & Play*” que permite la fácil adición o retiro de los nodos sin afectar el funcionamiento del sistema, la simplificación del cableado, el sencillo reconocimiento de errores y fallas del sistema, el fácil mantenimiento del mismo, la alta confiabilidad y eficacia en el envío de datos, la alta velocidad de transmisión y la optima operación en ambientes ruidosos. Estas, a su vez, contribuyen a la optimización de procesos productivos, reduciendo así los costos de operación y mantenimiento, al igual que los tiempos de diagnostico de fallas del sistema.
- La aplicación aportara a la infraestructura del laboratorio de Control de Procesos del programa de Ingeniería en Automática Industrial, permitiendo así a alumnos y docentes de la Universidad del Cauca, y en general a personas interesadas en el área de investigaciones de los sistemas de supervisión y comunicación de

procesos industriales, explorar y experimentar las temáticas acerca del bus de campo CAN para el desarrollo de futuros proyectos.

8 PROYECTOS FUTUROS

- Integrar a la red industrial otras plantas o dispositivos de campo presentes en el laboratorio de Automática Industrial o en otros laboratorios, como la planta de flujo de aire, con el fin de contar con todo un “sistema abierto” conectado bajo un mismo bus que proporcione el fácil monitoreo y control de procesos, al igual que el intercambio de información entre todos los dispositivos que componen el mismo.
- Mejorar el sistema implementado, adicionando más funcionalidades en el software de supervisión de la red, que permitan a los usuarios tener un mayor número de opciones, como el almacenamiento de la información generada por los procesos en bases de datos y el análisis de la misma, para la toma de decisiones sobre las herramientas de trabajo de la red.
- Agregar funciones al sistema supervisorio que permitan realizar un sistema de generación de reportes con el fin de llevar un control mas estricto de la información.

REFERENCIAS BIBLIOGRAFICAS

- [1] YOKOGAWA ELECTRIC CORPORATION. "Model EJA110A Differential Pressure Transmitter, General Specifications".
<http://www.rmcontrols.com/docs/GS01C21B01-00E.pdf>
- [2] GF + SIGNET 8550 Flow Transmitters.
<http://www.gfsignet.com/products/pdf/8550spec.pdf>
- [3] DWYER INSTRUMENTS, INC. "Series ABV Automated Ball Valve, Specifications - Installation and Operating Instructions".
http://www.dwyer-inst.com/htdocs/pdffiles/iom/valves/ABV_iom.pdf
- [4] FEEDBACK INSTRUMENTS LIMITED. "Process Trainer PT326 Manual". England.
- [5] NATIONAL SEMICONDUCTOR. "LM35 Precision Centigrade Temperature Sensors".
<http://www.national.com/ds.cgi/LM/LM35.pdf>
- [6] MICROCHIP TECHNOLOGY, INC. "PIC18FXX8 Data Sheet".
<http://ww1.microchip.com/downloads/en/DeviceDoc/41159d.pdf>
- [7] MICROCHIP TECHNOLOGY, INC. "MCP2551 High Speed CAN Transceiver"
<http://ww1.microchip.com/downloads/en/DeviceDoc/21667d.pdf>
- [8] ROCKWELL AUTOMATION. "Controladores MicroLogix 1500". Boletín 1764. Publicación 1764-TD001A-ES-P. Marzo 2002.
- [9] ROCKWELL AUTOMATION. "Compact 8-Bit Low Resolution Analog I/O Combination Module". Manual de Usuario. Número de Catálogo 1769-IF4XOF2. Publicación 1769-UM008A-EN-P. Noviembre 2001.
- [10] MICROCHIP TECHNOLOGY, INC. "Microchip Application Maestro™ Software User's Guide". 2003.
- [11] NATIONAL SEMICONDUCTOR. "DAC0808/DAC0807/DAC0806 8 - Bit D/A Converters". Enero, 1995.
- [12] ST, "LM124 - LM224 - LM324 Low Power Quad Operational Amplifiers". Diciembre, 2001.
- [13] ROCKWELL AUTOMATION. "RSLogix500 Help. Channel Configuration".
- [14] ROCKWELL AUTOMATION. "MicroLogix™ 1200 y MicroLogix 1500 Programmable Controllers". Boletines 1762 y 1764. Publicación 1762-RM001E-EN-P. Octubre, 2003.

[15] ROCKWELL AUTOMATION. "RSView32 User's Guide".