



## **PROTOCOLO AS-INTERFACE, ANALIZADOR DE RED**

**Autores:**  
**DIEGO ANDRÉS ORDUZ TIBADUIZA**  
**RIGOBERTO ORTIZ BALLESTEROS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**  
**FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS**  
**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y**  
**TELECOMUNICACIONES**  
**GRUPO DE INVESTIGACION CEMOS**



**BUCARAMANGA, OCTUBRE DE 2006**



## **PROTOCOLO AS-INTERFACE, ANALIZADOR DE RED**

**Autores:**

**DIEGO ANDRÉS ORDUZ TIBADUIZA  
RIGOBERTO ORTIZ BALLESTEROS**

**Tesis de grado para optar el título de  
Ingeniero Electrónico**

**Director:**

**MPE. Jaime Guillermo Barrero Pérez**

**Codirector:**

**MIE. Omar Leonardo Núñez Gualdrón**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES  
GRUPO DE INVESTIGACIÓN CEMOS**



**BUCARAMANGA, OCTUBRE DE 2006**

## TÍTULO: PROTOCOLO *AS-INTERFACE*, ANALIZADOR DE RED\*

DIEGO ANDRÉS ORDUZ TIBADUIZA

RIGOBERTO ORTÍZ BALLESTEROS\*\*

**Palabras claves:** *As-Interface*, Bus de Campo, Protocolo, Analizador, Microcontrolador, *Sniffer*, *Labview*.

En este informe se presenta el diseño e implementación de un analizador de red en un dispositivo programable con su respectiva *HMI* en el computador, basado en el protocolo de buses de campo *AS-Interface*; se describe la implementación de la capa física y la interfaz lógica necesaria para su funcionamiento dentro de una red *AS-I*, especificados en la norma EN50295, en la cual se establecen cada uno de los parámetros asociados con el protocolo.

Como resultado de este trabajo de grado se logró implementar un analizador de red para supervisar el estado operativo de una red *As-Interface* comercial; este analizador tiene la capacidad de detectar esclavos digitales con cualquier perfil de configuración y esclavos analógicos con perfil S.7.3, el dispositivo cuenta con una interfaz física capaz de extraer potencia del bus de datos, una interfaz de visualización y presentación de datos por medio de una aplicación gráfica desarrollada en *labview* con indicadores del estado de los esclavos y *strings* para el análisis de las transacciones maestro-esclavo, una sistema de comunicación de datos por puerto *USB* que actúa como interfaz directa entre el PC y la red industrial, un reporte de tramas limitado por el número de datos adquiridos (alrededor de 320), y un tiempo de actualización de 100ms.

---

\* Trabajo de grado

\*\* Facultad de Ingenierías Fisicomecánicas, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez.CoDirector: Omar Leonardo Núñez G.

## TITLE: AS-INTERFACE PROTOCOL, A NETWORK ANALYZER\*

DIEGO ANDRÉS ORDUZ TIBADUIZA  
RIGOBERTO ORTÍZ BALLESTEROS\*\*

**Key words:** *As-Interface*, Bus de Campo, Protocolo, Analizador, Microcontrolador, *Sniffer*, *Labview*.

In this report appears the design and implementation of a network analyzer in a programmable device with its respective HMI in the computer, based on the protocol of field bus AS-Interface; describes the implementation of the physical layer and the necessary logical interface for its operation within a AS-I network, specified in the norm EN50295, in which they settle down each one of the parameters associated with the protocol.

As result of this work of degree was managed to implement a network analyzer to supervise the operative state of a AS-Interface commercial network; this analyzer has the capacity to detect digital slaves with any profile of configuration and analogical slaves with S.7.3 profile, the device counts on a physical interface able to extract power of the fieldbus, an interface of visualization and presentation of data by means of a graphical application developed in labview with indicators of the state of the slaves and strings for the analysis of the transactions master-slave, a system of data communication by port USB that acts like interface direct between the PC and the industrial network, a report of plots limited by the number of data collected (around 320), and a time of update of 100ms.

---

\* Work of degree

\*\* Faculty of Fisic-mecanics Engineerings, Electrical, Electronic School of Engineerings and Telecommunications. Director: Jaime Guillermo Pérez.CoDirector : Omar Leonardo Núñez G.

*A Dios, por darme vida y por permitirme ser todo lo que soy.*

*A mis Padres Ana Aydeé y Jorge por acompañarme en todo momento y ser la fuente de fortaleza para triunfar sobre todas las dificultades. Gracias por tu esfuerzo y dedicación por hacerme una mejor persona..*

*A mi Hermana Silvia, por ser tan especial.*

*A mi hermosa Gueyler Andrea por darme su apoyo incondicional, comprensión y por su cariño.*

*A toda mi familia y amigos que de alguna u otra manera me ayudaron para lograr esta meta, por haber permitido que sacrificara tiempo valioso de su compañía para poder terminar mi carrera.*

*Diego Andrés Ordúz Tibaduiza*

*A mi madre Marlene Ballesteros y a mi padre Crisanto Ortiz Corzo quienes incansablemente brindaron su apoyo en mis momentos de flaqueza, llevándome siempre a un punto de fortaleza.*

*A mis hermanos que de una u otra manera fueron generosos con migo, dándome fuerzas para continuar en el camino.*

*A Mónica Yurley Ramírez, quien estando a mí lado siempre fue comprensiva con mi causa, sin olvidar ese amor fraternal que junto con sus padres siempre me brindaron.*

*A toda mi familia que vio en mí un potencial vivo, capaz de cumplir metas inalcanzables ante sus ojos. Igualmente a todos aquellos que alguna vez tendieron su mano a favor de brindar lo único que vale en este mundo “La Amistad”*

*Rigoberto Ortiz Ballesteros*

## AGRADECIMIENTOS

Quienes desarrollamos este proyecto, agradecemos a todas aquellas personas que fueron apoyo y gran ayuda en el desarrollo de nuestro trabajo, a nuestros directores Jaime Guillermo Barrero Pérez y Omar Leonardo Núñez Gualdrón por confiar en nosotros, por su colaboración, exigencia, paciencia y apoyo durante horas incansables en busca de nuestra meta; a todos los que de alguna u otra forma hicieron posible llegar a feliz término esta labor.

Al ingeniero José Alejandro Amaya por su grandiosa cooperación.

A nuestros familiares y amigos, en quienes encontramos un soporte constante durante la ejecución de este proyecto.

También deseamos agradecer especialmente a *Analog Devices* por su generosidad.

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN</b>	16
<b>CAPITULO 1</b>	
<b>Protocolo <i>AS-Interface</i></b>	21
1.1 Componente e Interfases de una red <i>AS-I</i>	21
1.2 Características	23
1.2.1 Transmisión de una red <i>AS-I</i>	23
1.2.2 Codificación de la Señal	24
1.2.3 Requerimientos de Recepción	26
1.3 Comunicación	27
1.3.1 Principios	27
1.3.2 Control de Transmisión	27
1.3.3 Requerimientos de Tiempo	28
1.4 Estructura de Mensajes	29
1.5 Transacciones <i>AS-I</i>	30
<b>CAPITULO 2</b>	
<b>Hardware</b>	34
2.1 Etapa de Recepción de Señal Analógica	36
2.2 Etapa de Codificación de Pulsos	38
2.3 Etapa de Comunicaciones	39
2.4 Fuente de Alimentación	41

2.5 Printed Circuit Board	42
---------------------------	----

## **CAPITULO 3**

<b>Software</b>	44
-----------------	----

3.1 Diagrama de Flujo de la Interfaz Gráfica	44
--	----

3.2 Interfaz Gráfica	51
----------------------	----

3.2.1 Panel de Controles	51
--------------------------	----

3.2.2 Estado de los Esclavos	53
------------------------------	----

3.2.3 Intercambio de Datos	54
----------------------------	----

3.2.4 Administrador	56
---------------------	----

3.2.5 Híbrido	57
---------------	----

3.2.6 Créditos y Ayuda	58
------------------------	----

3.3 Diagrama de Bloques de la Interfaz Gráfica	60
--	----

3.4 Programación del Microcontrolador	60
---------------------------------------	----

3.4.1 Microcontrolador 68HC12E128	60
-----------------------------------	----

▪ Periféricos Utilizados	61
--------------------------	----

▪ Estructura	62
--------------	----

3.4.2 Diagrama de Flujo del programa en el Microcontrolador	65
---	----

3.5 Analizador Comercial	68
--------------------------	----

## **CAPITULO 4**

<b>Pruebas y Resultados</b>	69
-----------------------------	----

4.1 Señal de Recepción	69
------------------------	----

4.1.1 Señal de Maestro	70
------------------------	----

4.1.2 Señal de Esclavo	71
------------------------	----

4.2 Especificaciones de la Fuente de Alimentación	72
4.3 Emulador de Red Híbrida	73
4.4 Detección del Módulo Esclavo diseñado por el grupo de investigación en el protocolo AS-I de la Universidad Industrial de Santander	74
4.5 Interfaz Gráfica del Analizador AS-I en una Red Comercial	75

## **CAPITULO 5**

<b>Observaciones</b>	79
----------------------	----

## **CAPITULO 6**

<b>Recomendaciones y Futuros Desarrollos</b>	80
--	----

## **CAPITULO 7**

<b>Conclusiones</b>	81
---------------------	----

## **FIGURAS**

Figura 1 Pirámide de Automatización	18
Figura 1.1 Esquema Interfases de una red AS	22
Figura 1.2 Formas de Onda de la Señal AS-I	25
Figura 1.4 Validación de Pulsos	26
Figura 1.5 Control de Transmisión	28
Figura 1.6 Tiempos de Transacciones	28
Figura 1.7 Estructura de los mensajes	30

Figura 1.8 Tramas del maestro para cada transacción en modo estandar	32
Figura 1.9 Tramas del esclavo para cada transacción en modo estandar	33
Figura 2.1 Diagrama de Bloques Analizador <i>AS-I</i>	35
Figura 2.2 Diagrama Comparador de Niveles Analógico	36
Figura 2.3 Tarjeta de Desarrollo del Microcontrolado	39
Figura 2.4 Diagrama esquemático acondicionador del controlador <i>USB – TTL</i>	40
Figura 2.5 Fuente de Alimentación Nivel Positivo	41
Figura 2.6 Fuente Generador de Nivel Negativo	42
Figura 2.7 Capa Inferior Circuito Impreso	43
Figura 2.8 Capa Superior Circuito Impreso	43
Figura 2.9 Hardware	43
Figura 3.1 Diagrama Inicio Programa <i>Labview</i>	45
Figura 3.2 Diagrama de Análisis Programa <i>Labview</i>	47
Figura 3.3 Diagrama de Respuesta Programa <i>Labview</i>	48
Figura 3.4 Diagrama de Pregunta Programa <i>Labview</i>	49
Figura 3.5 Diagrama de Dirección y Error Programa <i>Labview</i>	50
Figura 3.6 Diagrama de Analógico Programa <i>Labview</i>	50
Figura 3.7 Panel de Control Analizado <i>AS-I</i>	52
Figura 3.8 Panel de Estado de Esclavos en la Red	53
Figura 3.9 Panel de Intercambio de Datos de Una Red <i>AS-I</i>	55
Figura 3.10 Pantalla Analizador Para una Red Híbrida <i>AS-I</i>	57
Figura 3.11 Pantalla Analizador Para una Red Híbrida <i>AS-I</i>	58

Figura 3.12 Pantalla de Créditos y Ayuda Analizador AS-I	59
Figura 3.13 Diagrama Principal Programa Microcontrolador	65
Figura 3.14 Diagrama Principal segunda parte Programa Microcontrolador	66
Figura 3.15 Diagrama Principal segunda parte Programa Microcontrolador	67
Figura 3.16 Diagrama Transmit()Programa Microcontrolador	67
Figura 4.1 Señal de la red AS-I y SCI del microcontrolador	70
Figura 4.2 Señal de la SCI del microcontrolador y red AS-I con mensajes de varios esclavos	70
Figura 4.3 Petición de maestro	71
Figura 4.4.a Detección de pulsos negativos de maestro	71
Figura 4.4.b Detección de pulsos positivos de maestro	71
Figura 4.5 Respuesta de esclavo	72
Figura 4.6.a Detección Pulsos Negativos de esclavo	72
Figura 4.6.b Detección pulsos positivos de esclavo	72
Figura 4.7 Pantalla Del Analizador con Red Híbrida	74
Figura 4.8 Pantalla Del Analizador Detectando un Esclavo Diseñado por el Grupo AS-I/UIS y Su Respectiva Transferencia de Datos	75
Figura 4.9 Pantalla Del Analizador Estado de Esclavos con red Comercial	76
Figura 4.10 Pantalla Del Analizador en Intercambio de Datos	76
Figura 4.11 Pantalla Del Analizador en Estado de Esclavos con un Esclavo Fuera de Línea	77
Figura 4.12 Pantalla Del Analizador en Estado de Esclavos con dos Esclavos Fuera de Línea	77

## TABLAS

Tabla 1.1 Características principales de la red <i>AS-I</i>	24
Tabla 1.2 Explicación de bits de la trama del mensaje	30
Tabla 3.1 Datos por defecto del analizador <i>AS-I</i> para adquisición USB	51
Tabla 3.2 Tabala de Comparación Analizador Comercial vs Analizador Implementado	68
Tabla 4.1 Mediciones de la fuente de alimentación	73

## INTRODUCCIÓN

Hoy en día, la evolución de los procesos automáticos industriales, van ligados a la generación de nuevas técnicas, que facilitan el control y la supervisión de procesos de forma remota.

Cada vez más, se hace indispensable contar con dispositivos capaces de analizar, monitorear y diagnosticar estados operativos en campos de acción industrial. En este sentido se hace necesario la apropiación de tecnologías comerciales actuales y el estudio de protocolos estándar de comunicación industrial, con el fin de generar e implementar soluciones de mayor rendimiento y menor costo para uso de la industria.

En este trabajo se presenta el diseño e implementación de un analizador de red en un dispositivo programable con su respectiva *HMI* en el computador, basado en el protocolo de buses de campo *AS-Interface*; se describe la implementación de la capa física y la interfaz lógica necesaria para su funcionamiento dentro de una red *AS-I* comercial, especificados en la norma EN50295 [1], en la cual se establecen cada uno de los parámetros asociados con el protocolo.

Dentro de los protocolos de comunicación orientados a buses de campo, *AS-I* es uno de los de mayor proyección en el nivel de sensor / actuador, debido a que se perfila como una de las soluciones de fácil implementación y más eficaces para el control distribuido en el sector productivo. Algunas de sus características sobresalientes son: su gran versatilidad, bajos costos de mantenimiento, alta fiabilidad, fácil integración de nuevos miembros a la red y la transmisión de datos y

alimentación a través de una misma línea . Así mismo, cuenta con el respaldo de la *AS-INTERNATIONAL ASSOCIATION* y de un número considerado de fabricantes de instrumentos de control industrial, tales como: *Siemens, Schneider, Allan Bradley*, entre otros), los cuales garantizan su permanencia en el mercado. No obstante, *AS-i* es un concepto independiente del fabricante, y especifica su funcionamiento a través de la norma EN50295 [1] (o IEC 62026-5).

El dispositivo fue diseñado para leer cualquier clase de perfil esclavo *AS-i* y su correcto funcionamiento fue evaluado con la puesta en marcha dentro de un red comercial perteneciente al Grupo de Investigación en Mecatrónica de la Universidad Industrial de Santander.

El dispositivo se destaca por tener una interfaz de comunicación física genérica y una interfaz hombre-máquina implementada en *LABVIEW* con adquisición por puerto *USB*. Adicionalmente, tiene la capacidad de extraer la potencia para su funcionamiento a través de la línea, con requerimientos de consumo dentro de los rangos establecidos por la norma [1].

### **Pirámide de automatización**

La automatización de procesos en la actualidad, ha traído consigo el manejo de una gran cantidad de información, que obligan a la división de tareas entre grupos de tecnologías organizadas de manera jerárquica; formando lo que se conoce en el campo industrial como Pirámide de Automatización. Así, dependiendo de la funcionalidad y el tipo de conexiones, se suelen distinguir cuatro niveles en un proceso industrial, la figura 1 ilustra cada uno de estos niveles, los cuales son brevemente descritos a continuación.

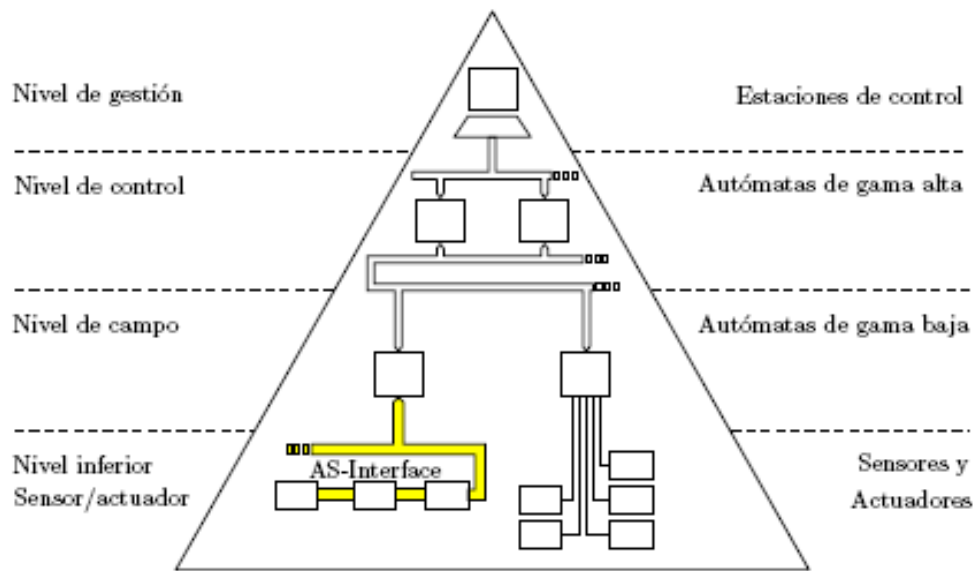


Figura 1 Pirámide de Automatización

- **Nivel de sensor/actuador:** Este nivel también es llamado nivel de instrumentación, esta formado por los instrumentos de medida (Sensores) y de mando (Actuadores) distribuidos en una línea de producción.
- **Nivel de campo:** Integra pequeños automatismos (*PLCs*<sup>1</sup>, *PIDs*, multiplexores de entradas/salidas, entre otros) en subredes o islas. En este nivel se puede encontrar uno o varios autómatas modulares actuando como maestros de la red. Los dispositivos de este nivel junto con los del nivel sensor/actuador poseen capacidad suficiente como para realizar procesos productivos por si mismos.
- **Nivel de control:** Enlaza las células de fabricación o zonas de trabajo. En este nivel se sitúan los autómatas de gama alta y los computadores dedicados al diseño, control de calidad, programación, monitoreo y supervisión de la planta.
- **Nivel de gestión:** Se utilizan para redes de oficinas, en las cuales se intercambia un alto volumen de información y los tiempos de respuesta no son críticos puesto que las comunicaciones en este nivel de la pirámide industrial ya

<sup>1</sup> Programmable logic Controller

no necesitan ser de tipo industrial, es decir muy robustas o de corto tiempo de acceso.

Así mismo, la necesidad de mejorar la eficiencia en el área de producción por medio del monitoreo y supervisión de procesos, ha permitido un desarrollo en las comunicaciones industriales, impulsando la generación de diferentes técnicas de control descentralizado y de fácil integración de nuevos elementos .

### **Antecedentes**

A nivel local, en la Universidad Industrial de Santander, se ha iniciado el desarrollo de trabajos orientados a fortalecer el área de las comunicaciones industriales, teniendo como resultado trabajos tales como:

- Nuñez Gualdrón Omar Leonardo. Modulo de integración de sensores y actuadores lógicos basado en protocolo *AS-Interface* de comunicaciones industriales. Tesis de Maestría, 2006.
- Marriaga Arnol, Abreu Oscar. Implementación de nodo esclavo en un microcontrolador. Tesis de pregrado, 2006.
- Carreño Sinle, Ardila Pedro. MODBUS, Monitoreo de la red empleando LABVIEW. Tesis de pregrado, 2005.
- Duque P. Jorge E. Procesador de comunicación MODBUS. Implementación con microcontrolador. Tesis de Maestría, 2005.
- López C. Duván A. Redes de Comunicación AS-i. Funcionamiento, Ventajas y Estado del Arte. Monografía Especialización en Telecomunicaciones, 2004.
- Niño Varon, Becerra Alvaro. Redes industriales. MODBUS y ETHERNET implementados en automatas programables TRILOGI, KOYO y MODICON-TELEMECANIQUE. Tesis de pregrado, 2002.
- Gelvez F. Julio. Redes de Comunicaciones Industriales. Monografía Especialización en Telecomunicaciones, 2002.

Los resultados obtenidos con estos proyecto, son una guía para futuros trabajos basados en redes de comunicaciones industriales, que involucren la implementación de protocolos en dispositivos programables, habilitando funciones de monitoreo y supervisión por computador, con capacidad de integración dentro de una red comercial. De igual forma, se desarrolló un aporte académico, en áreas de desarrollo profesional como instrumentación electrónica, automatización de procesos y comunicaciones industriales, dentro de la Universidad Industrial de Santander.

## 1. Protocolo *AS-INTERFACE*

AS-Interface es un bus de campo serial, que ha sido utilizado desde 1994, para la conexión de sensores/actuadores digitales (posteriormente para sensores analógicos) a una red industrial. Este protocolo fue diseñado para trabajar con el nivel de instrumentación de la pirámide de automatización, en donde permite adquirir información del proceso y enviar ordenes de control interpretadas por un maestro *AS-I* a los elementos conectados a la red; dicho protocolo puede manejar redes de corta envergadura sin que sea necesario el uso de repetidores de señal (Distancias menores a 100m); pero su principal característica y ventaja, es la transmisión de datos y alimentación a través de la misma línea. Finalmente, sus especificaciones están abiertas a dominio público, bajo los estándares IEC 62026-2 y EN50295 [1].

### 1.1. COMPONENTES E INTERFASES DE UNA RED *AS-I*

*AS-Interface* corresponde a un sistema de comunicación tipo maestro-esclavo, compuesto por un solo maestro y máximo 31 esclavos (62 para direccionamiento extendido). El maestro envía datos y parámetros a un determinado esclavo, éste toma la información, la procesa, ejecuta la orden recibida y notifica al maestro la orden cumplida, esto procedimiento se conoce como *cyclic polling* ó *request - answer*.

Un esquema general que muestra cada una de las interfaces de *AS-Interface* se presenta en la figura 1.1, las cuales van a ser brevemente descritas a continuación, junto con los principales agentes activos de la red.

## Interfaces de AS-Interface

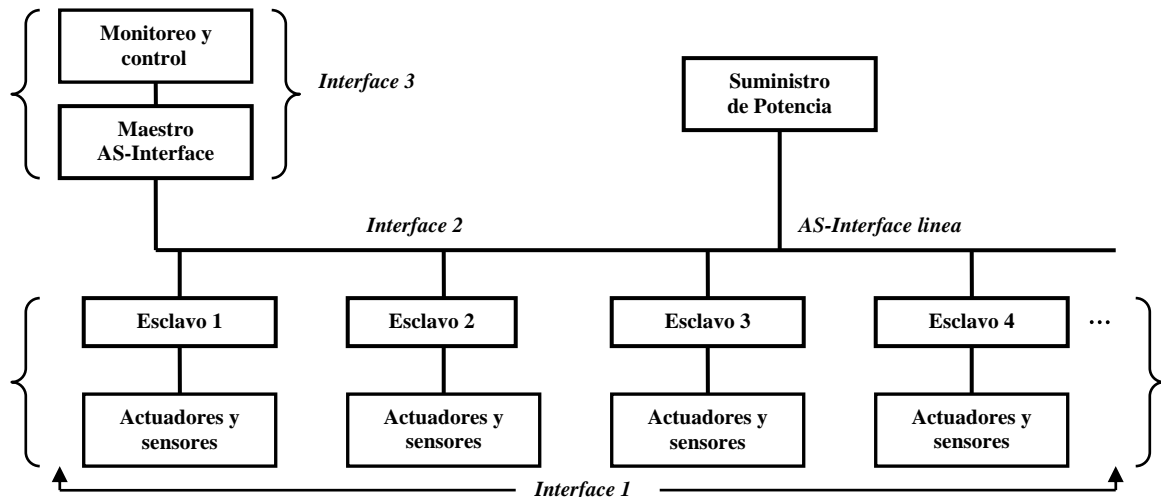


Figura 1.1 Esquema de Interfaces de una red AS-I

**Interface 1:** Es la interfaz entre los actuadores y *sensores* con el controlador inmediato, configurado como esclavo en la red AS-I.

**Esclavo AS-Interface:** Es el elemento que puede ser modificado por el maestro de forma remota a través de la línea AS-I.

**Interface 2:** Contiene la capa física de comunicación y alimentación de todos los dispositivos de la red, encargada de codificar la información y distribuirla, al igual que la fuente de potencia.

**Fuente AS-Interface:** Suministra la potencia DC para la red AS-I, energizando cada uno de sus componentes a través de la línea AS-Interface.

**Maestro AS-Interface:** Unidad encargada de ordenar y monitorear la red, se encarga de mantener la comunicación constante entre los esclavos. Es el elemento más importante de la red, la cual direcciona cada uno de los dispositivos

interconectados, además de darles ordenes y pedir registros de funcionamiento para el monitoreo y supervisión.

**Interface 3:** Es la interfase hombre maquina, que permite, el monitoreo de la red y la supervisión por parte del operario. Adicionalmente,proporciona todas las funciones utilizadas por el maestro para el control de la red AS-I.

**Repetidor AS-Interface:** Unidad encargada de reenviar la señal presente en la red AS-*i*, y de aislar galvánicamente la red para dispositivos encontrados a más de 100m de longitud.

## 1.2. CARACTERÍSTICAS

### 1.2.1. TRANSMISIÓN DE LA LÍNEA AS-I

El sistema de transmisión AS-*i* ejecuta una comunicación digital, serial y multipunto, de un maestro con otros dispositivos configurados como esclavos, sobre una línea bifilar sin apantallamiento. A través de la cual se transmiten simultáneamente los datos del proceso y la alimentación de los dispositivos conectados a la red.

El maestro genera una pregunta (*Request*) y el esclavo responde a dicha petición (*Response*), generando una señal binaria de 14*Bits* y 7*Bits* respectivamente, según lo estipulado en [1].

Algunas de las características principales del protocolo AS-interface se consignan en la Tabla 1.1

Tabla1.1. Características principales de la res AS-I

Característica	Descripción
Velocidad de Transmisión	166.67 kbps
Tiempo de <i>bit</i> [ $T_B$ ]	6 $\mu$ s
Número de esclavos	31 en modo estandar, 62 en modo extendido
Tiempo máximo de ciclo	5 ms
Detección de error	Bit de paridad
Corrección de error	Reenvío de datos por parte del maestro
Acceso al medio	Sondeo cíclico por parte del maestro
Tipo de bus	Maestro – Esclavo, Monomaestro

### 1.2.2. CODIFICACIÓN DE LA SEÑAL

En el bus de comunicación, las tramas de señal transmitidas son codificadas en formato Manchester II, seguido de una modulación APM (*Alternate Pulse Modulation*) esto quiere decir que durante la transmisión , cada pulso negativo es seguido por un pulso positivo (Redundancia Implícita). Producto de este proceso, un “0” binario, es representado por un alto y un bajo (Flanco descendente), mientras que un uno “1”, se representa con un bajo y un alto, (Flanco ascendente) ( Figura 1.2 ), cada uno de ellos en medio tiempo de *bit*, en este caso es de 3  $\mu$ s. Cuando no hay transmisión, el nivel permanecerá en un estado lógico alto.

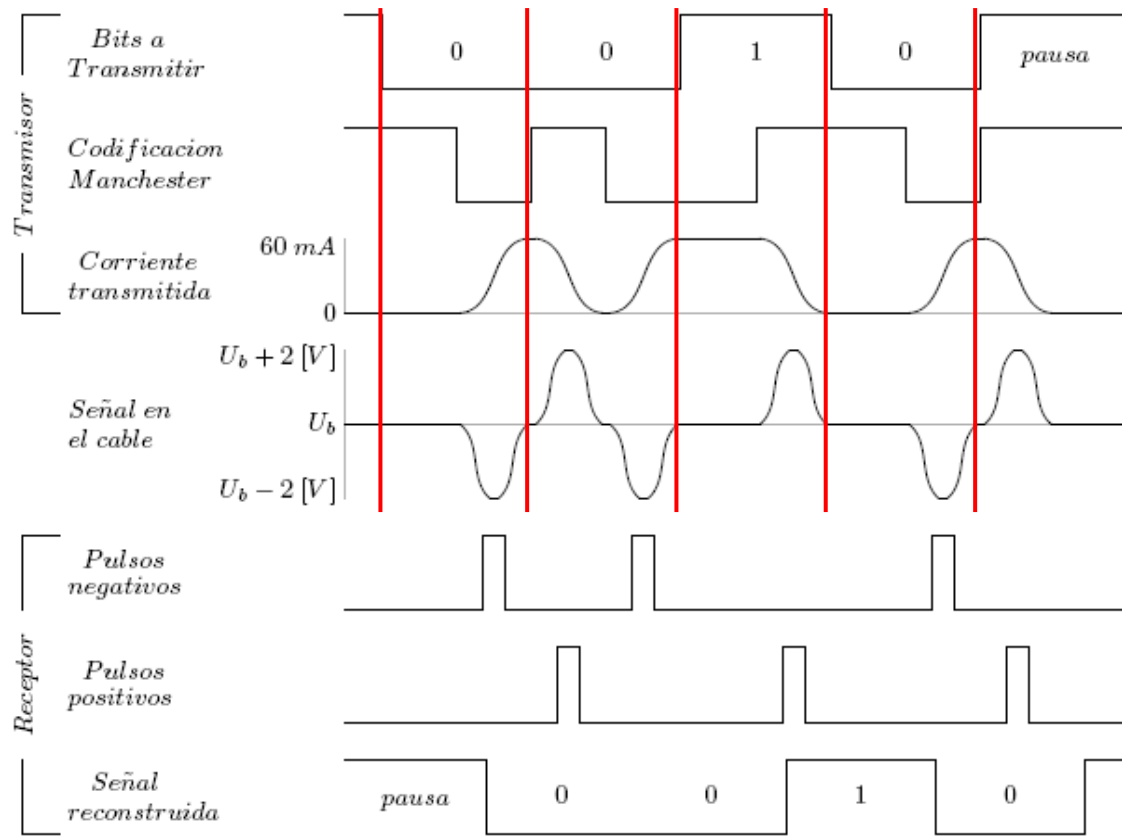


Figura 1.2. Formas de onda de la señal AS-I

Después de dicho proceso, los pulsos de voltaje resultantes pueden ser modelados idealmente, tal como se menciona en [1], por la ecuación.

$$U(t) = \pm U_{SEND} * \text{Sin}^2\left(\frac{2\pi t}{6 * 10^{-6}}\right) \text{ Con } U_{SEND} \approx 2 \text{ [V]} \quad (I)$$

La señal resultante, transmitida por la red a una velocidad aproximada de 166.67kbps, es mostrada la figura 1.2.

### 1.2.3. REQUERIMIENTOS DE RECEPCIÓN

Tomando como referencia la figura 1.3, la máxima amplitud del pulso ( $V_{max}$ ) en una red AS-I puede variar entre  $1.5V_{pico}$  y  $4V_{pico}$ , siendo consecuencia directa de estas variaciones la ubicación o la configuración de los esclavos, debido a que poseen distintas configuraciones.

Para la validación de un pulso y entenderlo como parte del mensaje, su amplitud debe estar entre el 65% y 100% de  $V_{max}$ . Además, los picos inferiores al 30% de  $V_{max}$ , se consideran ruido que no debe afectar el mensaje. Todo esto se realiza en base a la amplitud del pulso inicial de la transmisión. Los tiempos de duración para cada uno de los pulsos debe ser de  $3 \mu s$ , con unas tolerancias descritas en [1 página 19].

Algo que se debe tener en cuenta, es que al utilizar la misma línea para transmitir señal y a la vez energía, la fuente debe tener requerimientos de simetría muy bien definidos, con el fin de no alterar la señal analógica que se quiere transmitir. De igual forma, cuando se quiere tomar la señal presente en la red, la adquisición debe ser totalmente simétrica, evitando afectar la configuración de la línea de transmisión.



Figura 1.3. Validación de pulsos

## **1.3. COMUNICACIÓN**

### **1.3.1. PRINCIPIOS**

El protocolo *AS-i*, es un sistema de comunicación monomaestro con 31 esclavos (62 en modo extendido), cada uno con única dirección en el rango de 1 a 31. Aquellos esclavos que posean una dirección de operación, responden a los datos y peticiones realizadas por el maestro.

Una transacción simple esta compuesta de una petición de maestro y se complementa con una respuesta de esclavo. Una transacción es combinada cuando existen varias transacciones simples.

### **1.3.2. CONTROL DE TRANSMISIÓN.**

El intercambio de datos entre el maestro y los esclavos está implementada por el proceso de transacciones. Una transacción inicia con una petición de maestro, el cual espera por una respuesta del esclavo dentro de determinado tiempo; si el esclavo no responde en ese tiempo, el maestro repetirá el mensaje una vez más. Después de recibir una respuesta válida, el maestro deberá iniciar la siguiente transacción, una vez el tiempo de espera haya terminado. Los esclavos no responden al maestro, si detectan una falla en la petición o si el maestro emite una petición no definida. Revisar la figura 1.4.

### 1.3.3. REQUERIMIENTOS DE TIEMPO

En el protocolo *AS-i* una transacción se divide en dos acciones, *master request* – *slave response*; dos intervalos de tiempo, *master pause* – *send pause*; y en ausencia de la respuesta de esclavo, esta se monitorea por el *slave response time-out*. Revisar la figura 1.4 y 1.5.

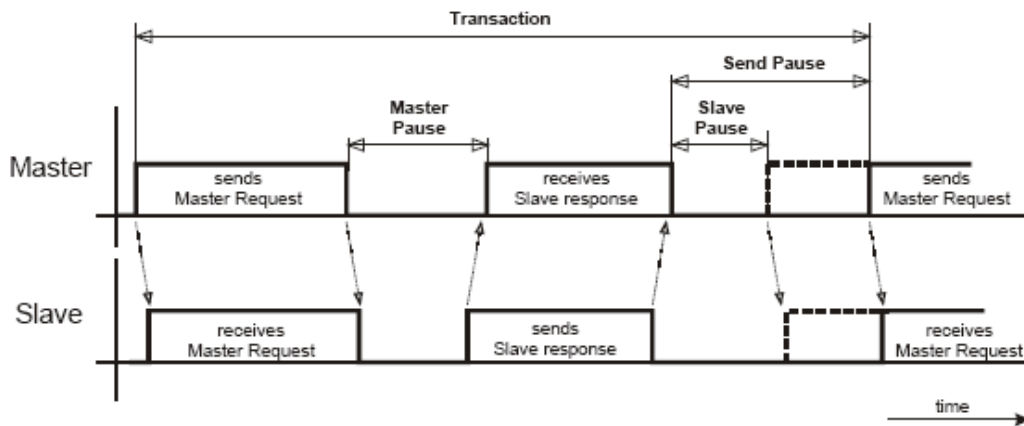


Figura 1.4 . Control de transmisión

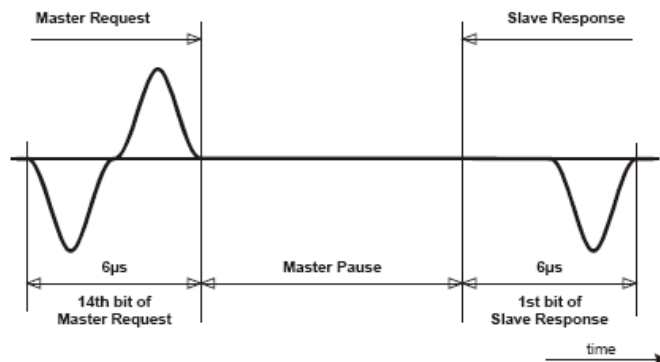


Figura 1.5. Tiempos en transacciones

A continuación se definen los tiempos de espera y las acciones respectivas según la norma [1].

- Petición de maestro (*Master Request*): Mensaje enviado por el maestro hacia un esclavo en operación.

- Pausa de maestro (*Master pause*): Durante este tiempo el esclavo procesa la pregunta del maestro y produce su respuesta.
- Respuesta de esclavo (*Slave Response*): Envío de datos del esclavo hacia el maestro, o el comando de respuesta al maestro después de cierta petición.
- Pausa de esclavo (*Slave pause*): Después de recibir una respuesta de esclavo, debe existir un tiempo mínimo durante el cual no debe realizarse ninguna transmisión. Esta pausa corresponde a un tiempo entre  $1T_B$  y  $2T_B$ .
- Pausa de envío (*Send pause*): Después de recibir una respuesta de esclavo, debe existir un tiempo mínimo durante el cual no debe realizarse ninguna transmisión. En operación normal con más de 30 transacciones por ciclo, esta pausa debe ser igual a una pausa de esclavo. Si existen 30 o menos transacciones por ciclos, esta pausa puede ser prolongada hasta  $500 \mu s$ .
- *Slave response time-out*: Si el maestro no recibe ninguna respuesta del esclavo que consulta, dentro de un tiempo definido (*Slave response time-out*), el maestro debe finalizar la transacción o repetir la petición. Esta pausa debe ser máximo de  $10T_B$ .

#### **1.4. ESTRUCTURA DE LOS MENSAJES AS-*i***

La estructura del datagrama para el maestro y el esclavo, se muestra en la figura 1.6 a y b respectivamente, y las especificaciones de estas se muestran en la tabla 1.2.

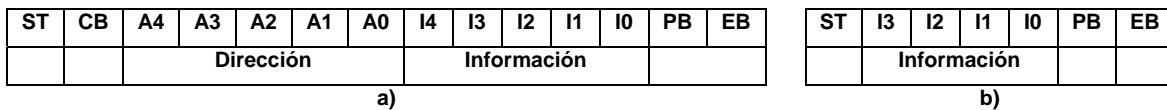


Figura 1.6. Estructura de los mensajes : a) Petición de maestro b) Respuesta de esclavo

BIT	Significado	Comentario
ST	<i>Bit de Inicio</i>	Identifica el inicio del mensaje. Siempre ST=0
CB	<i>Bit de Control</i>	Identifica el tipo de pregunta del maestro
A4...A0	Dirección (5 Bits)	Dirección de los esclavos
I4...I0	Información	Bits que contienen la petición del maestro
I3...I0	Información	Bits que contienen la respuesta del esclavo
PB	<i>Bit de Paridad</i>	<i>Bit de detección de errores</i>
EB	<i>Bit de Fin</i>	Identifica el final del mensaje. Siempre EB=1

Tabla 1.2. Explicación de bits de la trama del mensaje

Como se puede observar en las figuras anteriores, tanto las peticiones de los maestros como las respuestas de los esclavos poseen un *bit* de inicio fijo en “0” y un *bit* de fin fijo en “1”. Adicionalmente, las estructuras utilizan un *bit* de paridad para la detección de errores, lo cual permite mayor seguridad en el intercambio de datos; donde la paridad calculada es par, es decir, el mensaje siempre tiene un número par de “1”, sin incluir los *bits* de inicio y fin de trama.

## 1.5. TRANSACCIONES AS-I

El protocolo AS-I define cuatro tipos de transacciones, las cuales son definidas a continuación. Así mismo se presentarán las estructuras básicas de cada uno de los respectivos datagramas de maestro (Figura 1.7) y esclavo (Figura 1.8).

**Data\_Exchange:** Sirve para entregar y/o recibir el patrón de datos de las entradas/ salidas del esclavo.

**Write\_Parameter:** Sirve para entregar y/o recibir el patrón de *bits* de los parámetros de los puertos del esclavo.

**Address\_Assignment:** Asignar dirección al esclavo.

**Comands:** Toma datos y aplica las siguientes funciones:

- *Reset AS-I Slave:* Reinicia la máquina de estados del esclavo (enviándolo al estado *INIT*) en un tiempo inferior a 2ms, tiempo durante el cual el esclavo no debe responder ninguna petición hecha por el maestro. Además, este comando permite recuperar la dirección del esclavo, si esta ha sido borrada con el comando *Delete Address*.
- *Delete Address:* Borra la dirección actual del esclavo, para permitir la asignación de una nueva dirección con la transacción *Address Assignment*.
- *Read I/O Configuration:* Permite al maestro leer el registro *I/O Configuration* del esclavo.
- *Read Identification Code:* Permite al maestro leer el registro *ID Code* del esclavo.
- *Read Status:* Permite al maestro leer el registro *Status* del esclavo.

<b>Data_Exchange</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=0	A4	A3	A2	A1	A0	=0	D3	D2	D1	D0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Write_Parameter</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=0	A4	A3	A2	A1	A0	=1	P3	P2	P1	P0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Address_Assignment</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=0	=0	=0	=0	=0	=0	A4	A3	A2	A1	A0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Write_Extended_ID-Code_1</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	=0	=0	=0	=0	=0	=0	ID3	ID2	ID1	ID0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Delete_Address</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=0	=0	=0	=0	=0	PB	=1
			5 Bits Dirección					4 Bits Información						
<b>Reset_Slave</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=1	=1	=0	=0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Read_I/O_Configuration</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=0	=0	=0	=0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Read_ID-Code</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=0	=0	=0	=1	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Read_Extended_ID-Code_1</b>	<b>ST</b>	<b>CB</b>							<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=0	=0	=1	=0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Read_Extended_ID-Code_2</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=0	=0	=1	=1	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Read_Status</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=1	=1	=1	=0	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>R1</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	A4	A3	A2	A1	A0	=1	=1	=1	=1	=1	PB	=1
			5 Bits Dirección					5 Bits Información						
<b>Broadcast</b>	<b>ST</b>	<b>CB</b>						<b>I4</b>	<b>I3</b>	<b>I2</b>	<b>I1</b>	<b>I0</b>		<b>EB</b>
	=0	=1	=1	=1	=1	=1	=1	=1	=0	=1	=0	=1	PB	=1
			5 Bits Dirección					5 Bits Información						

Figura 1.7. Tramas del maestro para cada transacción en modo estandar

Data_Exchange	ST	I3	I2	I1	I0		EB
	=0	D3	D2	D1	D0	PB	=1
	4 Bits información						
Write_Parameter	ST	I3	I2	I1	I0		EB
	=0	P3	P2	P1	P0	PB	=1
	4 Bits información						
Address_Assignment	ST	I3	I2	I1	I0		EB
	=0	=0	=1	=1	=0	PB	=1
	4 Bits información						
Write_Extended_ID-Code_1	ST	I3	I2	I1	I0		EB
	=0	=0	=0	=0	=0	PB	=1
	4 Bits información						
Delete_Address	ST	I3	I2	I1	I0		EB
	=0	=0	=0	=0	=0	PB	=1
	4 Bits información						
Reset_Slave	ST	I3	I2	I1	I0		EB
	=0	=0	=1	=1	=0	PB	=1
	4 Bits información						
Read_I/O-Configuration	ST	I3	I2	I1	I0		EB
	=0	=I3	=I2	=I1	=I0	PB	=1
	4 Bits información						
Read_Extended_I/D-Code_1	ST	I3	I2	I1	I0		EB
	=0	=I3	=I2	=I1	=I0	PB	=1
	4 Bits información						
Read_Extended_I/D-Code_2	ST	I3	I2	I1	I0		EB
	=0	=I3	=I2	=I1	=I0	PB	=1
	4 Bits información						
Read_Status	ST	I3	I2	I1	I0		EB
	=0	=S3	=S2	=S1	=S0	PB	=1
	4 Bits información						
R1	ST	I3	I2	I1	I0		EB
	=0	=S3	=S2	=S1	=S0	PB	=1
	4 Bits información						
Broadcast	ST	I3	I2	I1	I0		EB
	=0	=S3	=S2	=S1	=S0	PB	=1
	4 Bits información						

Figura 1.8. Tramas del esclavo para cada transacción en modo estandar

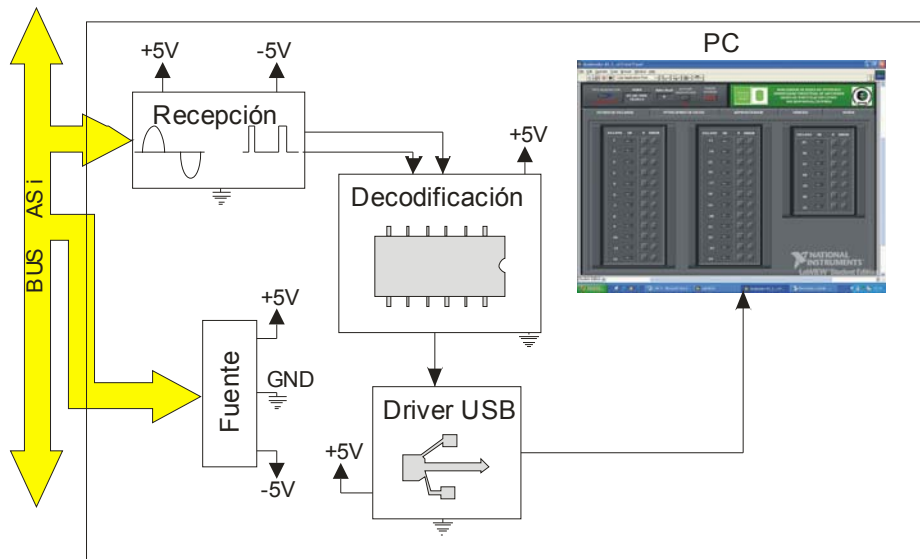
## 2. HARDWARE

El diseño e implementación de un analizador de red *AS-interface*, implica la existencia de una capa física, la cual se encarga de hacer el puente entre los datos obtenidos de la red y su posterior procesamiento, cumpliendo con los requerimientos mínimos para no causar efectos nocivos a la red. Además el dispositivo debe estar en capacidad de extraer niveles de potencia de la línea suficientes para su funcionamiento.

El dispositivo físico esta conformado por 4 etapas independientes, energizadas directamente de la línea *AS-i*, las cuales son:

- Una primera etapa de recepción de datos provenientes de la línea *AS-I*, es la encargada de tomar la señal analógica y entregarla en forma de pulsos para su posterior decodificación.
- Una segunda etapa de Decodificación y transmisión, es la encargada de tomar los pulsos entregados por la primera etapa y decodificarlos, de tal forma que se pueda entender en lenguaje de máquina (Lenguaje Binario), lo que esta sucediendo en la red. Adicionalmente esta etapa envía datos a una etapa posterior.
- Una etapa de transmisión *USB*, la cual entrega los datos al PC, para que estos sean procesados como información de la red por parte de la interfaz de usuario.
- Una ultima etapa de alimentación, capaz de extraer la potencia necesaria de la red, para que el dispositivo Hardware del analizador *AS-I*, permanezca en correcto funcionamiento.

En la figura 2.1 se muestra el diagrama de bloques para el analizador de red *AS-Interface*, resaltando las etapas de hardware mencionadas anteriormente y el flujo de señal para el funcionamiento a nivel global del dispositivo, tomando todo lo necesario de la red, desde la energía hasta las transacciones.



**Figura 2.1 Diagrama de Bloques Analizador AS-I**

El primero de los bloques representa la etapa de recepción, que toma la señal analógica presente en el bus de datos de la red AS-i, y la transforma en pulsos positivos de 5V de amplitud. Seguidamente esta una etapa de decodificación y transmisión de datos al PC, la cual se hace por medio del microcontrolador y un controlador USB, respectivamente. Además se encuentra la fuente de alimentación dual, extrayendo la potencia del bus de datos AS-i, con una referencia interna, que solo tiene significado para el circuito de recepción.

Una de las fases de mayor importancia en el funcionamiento global del sistema es la decodificación, a través de la cual se logra proporcionar exactitud en el manejo de los requerimientos de tiempo y descifrar la señal mensaje presente en el bus de datos.

A continuación se realizará una explicación detallada de cada uno de los bloques funcionales, presentando características técnicas y diagramas esquemáticos de los circuitos diseñados.

## 2.1 ETAPA DE RECEPCIÓN ANALÓGICA

Esta etapa es la encargada de la interfaz directa con el bus de datos *AS-I*, ya que esta dispuesta de forma tal que realice un desacople de los niveles de continua presentes en la red, tomando solo la señal analógica que representa el mensaje; posteriormente realiza un proceso de comparación para detectar niveles aceptables, con especificaciones detalladas en la norma [1], aceptando para el sistema niveles superiores a 1.2 V, ya que cumple con la condición requerida en [1], para la cual se aceptan niveles que superen el 60% de la amplitud del pulso máximo (la norma toma como referencia un pulso de 2V). Finalmente esta etapa convierte estas formas de onda analógicas en pulsos de 0-5V. En la figura 2.2, se muestra el circuito esquemático utilizado para este fin.

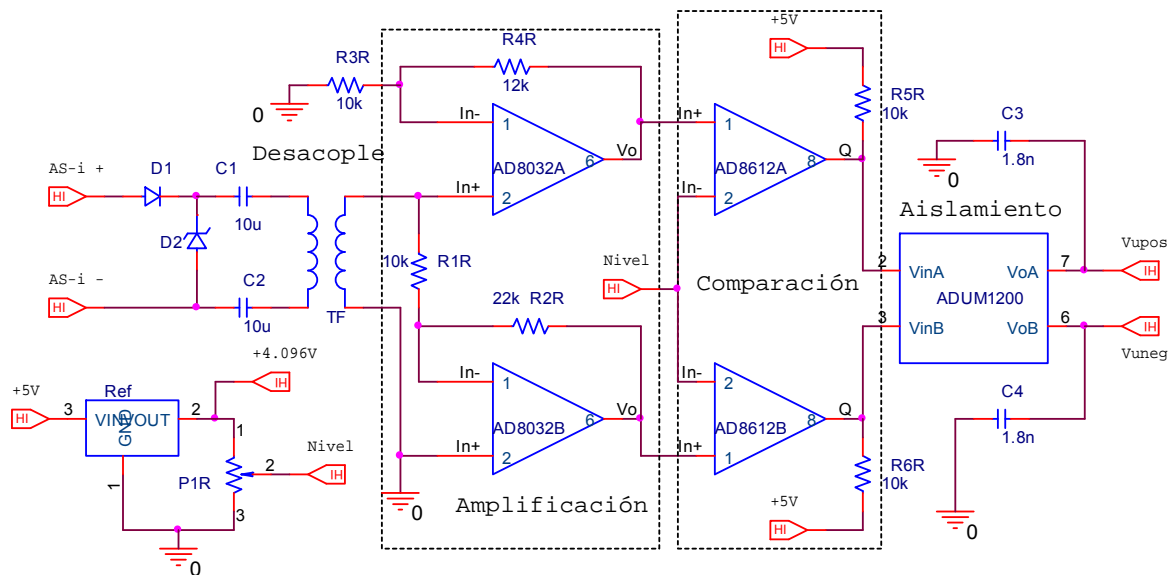


Figura 2.2 Diagrama Comparador de Niveles Analógicos

Inicialmente la señal presente en el bus *AS-I*, es filtrada mediante condensadores no polares, con el fin de eliminar la componente de continua; seguidamente un transformador (WB2010-1) con relación 1:1 realiza el desacople total de señal [19].

Superando el desacople de continua, la señal es llevada a través de dos canales independientes, donde el problema radica en identificar la polaridad adecuada. De esta forma y teniendo en cuenta que el microcontrolador solo maneja referencias positivas, se realizó un acondicionamiento de señal para que estos picos sean pulsos digitales y decodificables para el procesador.

Esto se desarrolló, utilizando configuraciones de amplificador no inversor (1) y amplificador inversor(2), con ganancias de 2.2 V/V y -2.2 V/V, para la extracción de los pulsos analógicos positivos y negativos, respectivamente.

$$G_{POS} = 1 + \frac{R_{4R}}{R_{3R}} = 2.2 \text{ V/V} \quad (1) \quad G_{NEG} = -\frac{R_{2R}}{R_{1R}} = -2.2 \text{ V/V} \quad (2)$$

Para este acondicionamiento de señal, es utilizado el amplificador operacional AD8032 de *Analog Devices*, justificado en características importantes como las relacionadas a continuación [14].

- Polarización especificada a +2.7 hasta +5 V y ±5 V.
- Corriente de polarización máxima de 800 uA
- Ancho de banda de 80 MHz para G = 1 V/V
- Slew Rate de 30 V/uS
- Rail to Rail Input and Output
- Corriente de salida de 15 mA
- Bajo voltaje de Offset de 1.5 mV.

Continuando con el seguimiento de la señal, esta es comparada con un nivel de referencia constante definido de tal manera que se cumpla con las especificaciones de la norma [1]. Anteriormente se habló de un nivel aceptable de pulsos que supere el 60% del pulso máximo, por esta razón, se define como criterio de diseño el peor de los casos, cuando un nivel AS-i mínimo aceptable es de 1.5V, siendo el 60% de este 0.9V, al ser amplificado por la etapa de recepción,

este se convierte en 1.8V, el cual es el valor de referenciada justificado para el sistema. Para esta función se seleccionó el comparador AD8612 de *Analog Devices*, con muy buenas especificaciones como las relacionadas a continuación [15].

- Alta velocidad de comparación
- Tiempo de respuesta de 4 ns a +5V
- Entradas superiores a 100 MHz
- Funciones de *Latch*

Su carácter digital, da como nivel alto cinco *volts* (5 V), amplitud necesaria para ser tomada como entrada en los puertos de propósito general del microcontrolador, utilizados para el proceso de detección y decodificación.

Estos pulsos entregados por el comparador son finalmente acondicionados por un aislador digital ADUM1200 de *Analog Devices* [17], el cual cumple las funciones de desacoplar digitalmente y acondicionar los pulsos para la correcta lectura en los puertos del microcontrolador, además protege los puertos del mismo. Ver funcionamiento en el capítulo 4.

## **2.2 ETAPA DE DECODIFICACION DE PULSOS**

Esta etapa es la encargada de decodificar la señal recibida y entregarla a la etapa de comunicaciones con el PC, estableciendo un código de comunicaciones específico para el correcto funcionamiento de las partes [20]. En la sección 3.7 de la hoja de datos del microcontrolador se habla con mayor profundidad del funcionamiento de este bloque, ya que es netamente un proceso de software.

Para este fin se implementó la tarjeta de desarrollo simplificada para la aplicación, como la mostrada en la figura 2.3, habilitando periféricos como el módulo *Timer*, la interfaz *SCI* y los puertos de propósito general.

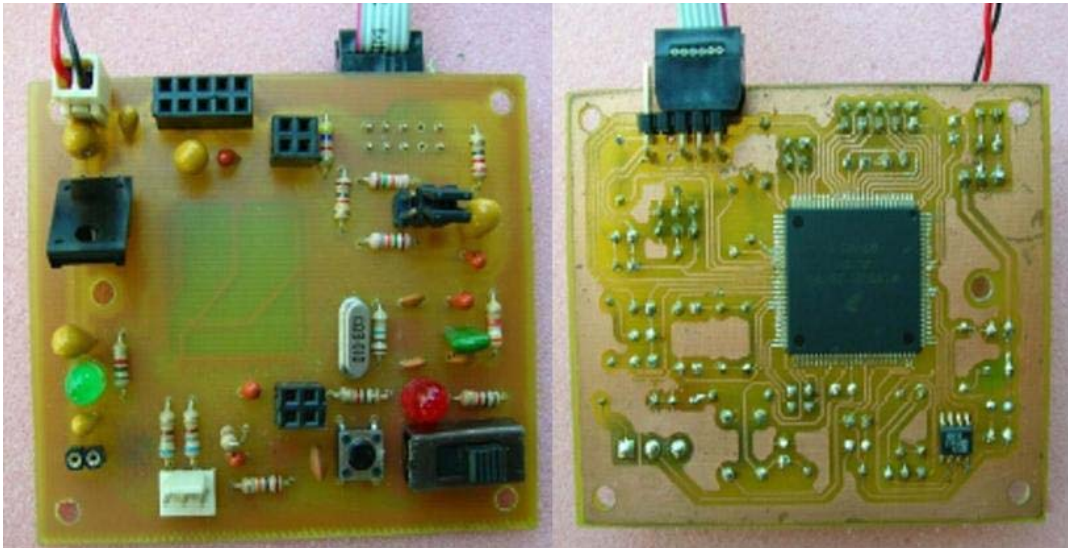


Figura 2.3 Tarjeta de Desarrollo del Microcontrolador.

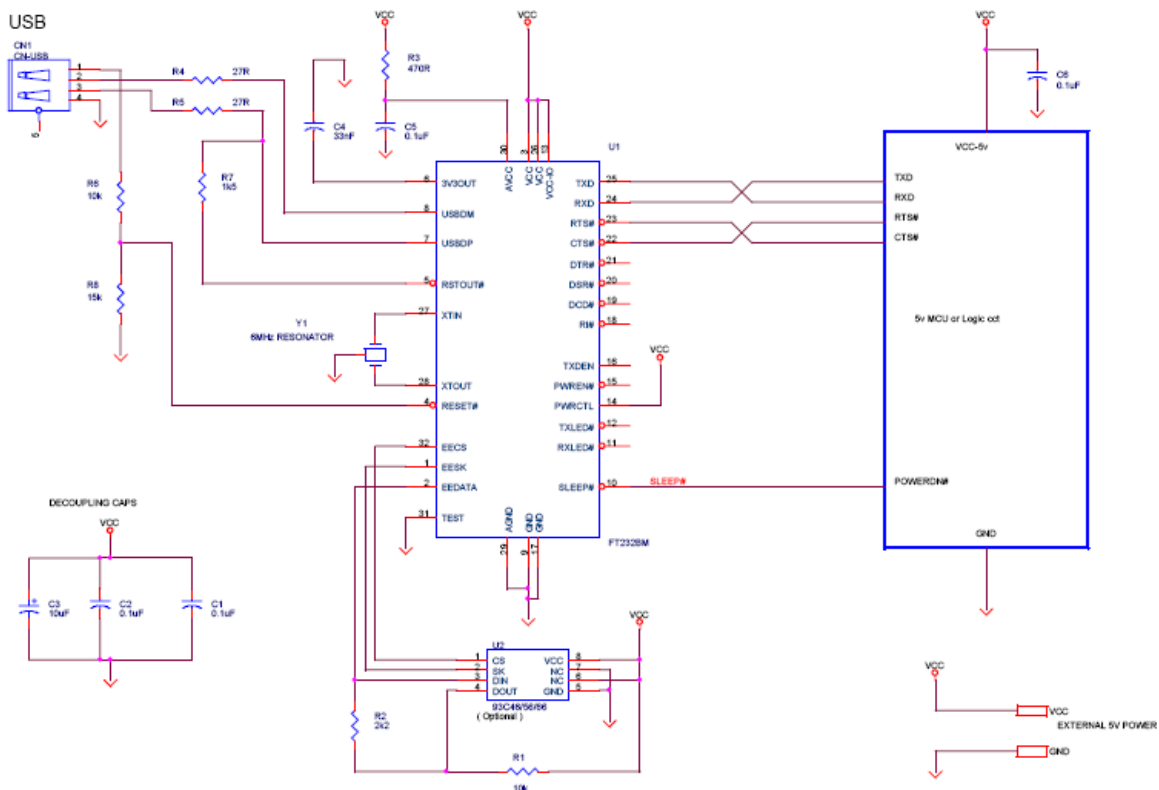
## 2.3 ETAPA DE COMUNICACIONES

Este bloque se encarga de recibir los datos que decodifica el microcontrolador y transmitirlos, utilizando el controlador USB 2.0 referencia FT232BM de *Future Technology Devices*, el cual posee las siguientes características.

- Transferencia de datos seriales asíncronos mediante el estándar USB 2.0 en un solo circuito integrado.
- Soporta protocolo de comunicación serial, esto es transferencia de 7 u 8 *bits*, además de permitir la adición del *Bit* de paridad, *bit* de inicio y *bit* de parada.
- Descarga datos a 3Mbps mediante interfaz USB – TTL.
- Posee un Driver software de fabricante, llamado VirtualCOM, compatible con variedad de sistemas operativos, entre ellos Windows XP.

Para más información acerca de las características de este dispositivo, referirse a la hoja de datos [23].

En la figura 2.4 se muestra el diagrama esquemático utilizado para el acondicionamiento del controlador *USB – TTL*



FT232B APPLICATION SCHEMATIC  
 INTERFACING TO 5 V LOGIC - SELF POWERED APPLICATION  
 UPDATED 21st January 2003

**Figura 2.4 Diagrama esquemático acondicionador del controlador *USB – TTL***

Este controlador da la posibilidad de implementar una memoria EEPROM, la cual no se tiene en cuenta, debido a que la aplicación implica envío de datos continuos.

## 2.4 FUENTE DE ALIMENTACIÓN

Basados en las tesis de maestría y pregrado referenciadas en [21] y [22], respectivamente, que sustentan los primeros desarrollos en el estudio e implementación del protocolo AS-I, se tomó una topología utilizada para obtener la potencia necesaria del mismo bus de datos.

Se adquiere la tensión diferencial de 30 V presente en el bus, la cual es filtrada por medio de una configuración inductiva para evitar efectos degradantes por parte de la señal analógica; dicho filtro cuenta con inductores de 1.8 mH. A continuación se procede a generar una referencia o tierra y mediante el uso de reguladores de voltaje, se acondiciona el nivel de tensión necesario para el correcto funcionamiento de la tarjeta de adquisición.

En la figura 2.5, se muestra el diagrama esquemático de la primera parte de esta fuente, es decir la que entrega el nivel de tensión positivo de 5V.

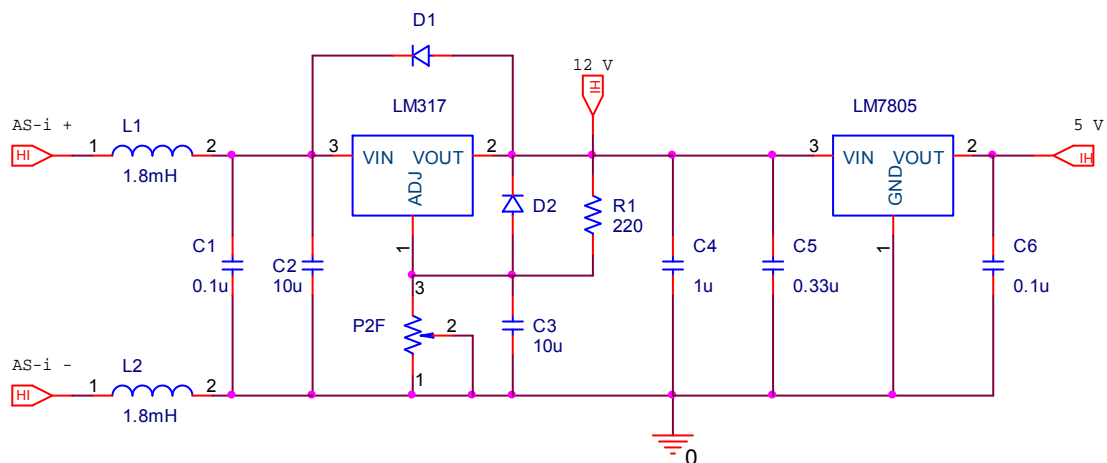
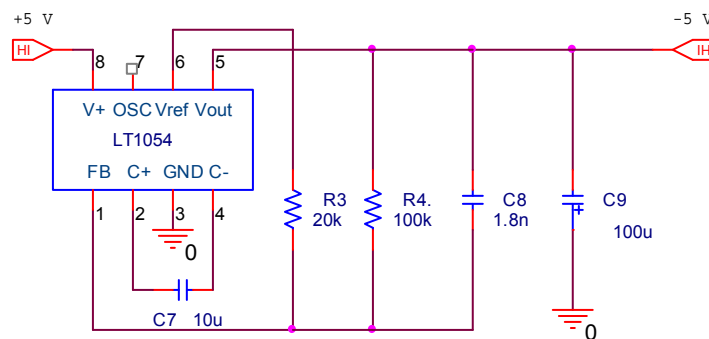


Figura 2.5 Fuente de Alimentación Nivel Positivo

Para efectos de regulación de tensión, se utilizó un circuito integrado LM317, el cual reduce la tensión del BUS de 30 V a un valor de 12 V, luego se conecta en cascada un circuito integrado LM7805, el cual permite obtener el nivel de referencia positivo de 5 V. Adicionalmente en cada punto estratégico en el que la

señal de tensión esta presente, son ubicados condensadores electrolíticos de 10  $\mu\text{F}$ , condensadores de tantalio de 0.1 y 1  $\mu\text{F}$  para disminuir los niveles de ruido e interferencia ocasionados por fluctuaciones en la fuente primaria de tensión.

La tarjeta cuenta con una alimentación dual de  $\pm 5\text{V}$ .. En la figura 2.6 se muestra el esquema de la fuente generadora del nivel negativo.



**Figura 2.6 Fuente Generador de Nivel Negativo**

Debido a que no existe una referencia negativa como tal, es necesario recurrir a un circuito integrado con capacidad de invertir la tensión en su entrada y con capacidad de carga. Para este fin se utilizó un circuito integrado LT1054, el cual a partir de la conmutación de un juego de condensadores, logra invertir el voltaje presente en su entrada, entregándolo con valor negativo y una capacidad de carga de 100mA.

## **2.5 PRINTED CIRCUIT BOARD**

Al integrar todos los módulos anteriormente descritos en una tarjeta de circuito impreso se deben tener algunas recomendaciones [24].

En las figura 2.7 ,2.8, y 2.9 se muestra el diseño implementado para el analizador AS-I. En esta tarjeta de circuito impreso se encuentran inmersas cada una de las etapas de Recepción, fuente y transmisión.

La sección de decodificación es una tarjeta independiente, ya que se trata de un sistema de desarrollo implementado con lo necesario para la aplicación.

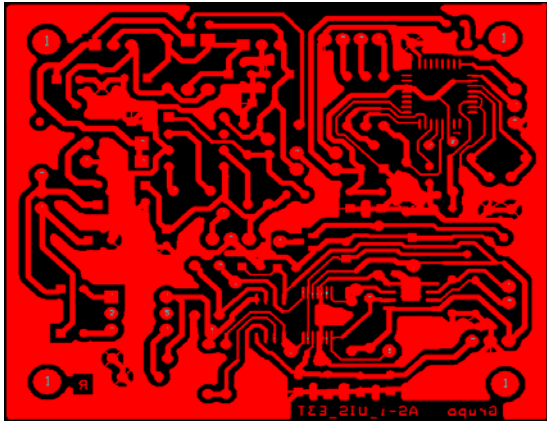


Figura 2.7 Capa Inferior Circuito Impreso

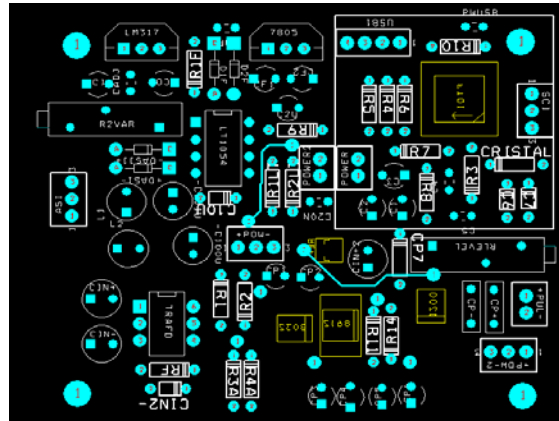


Figura 2.8 Capa Superior Circuito Impreso

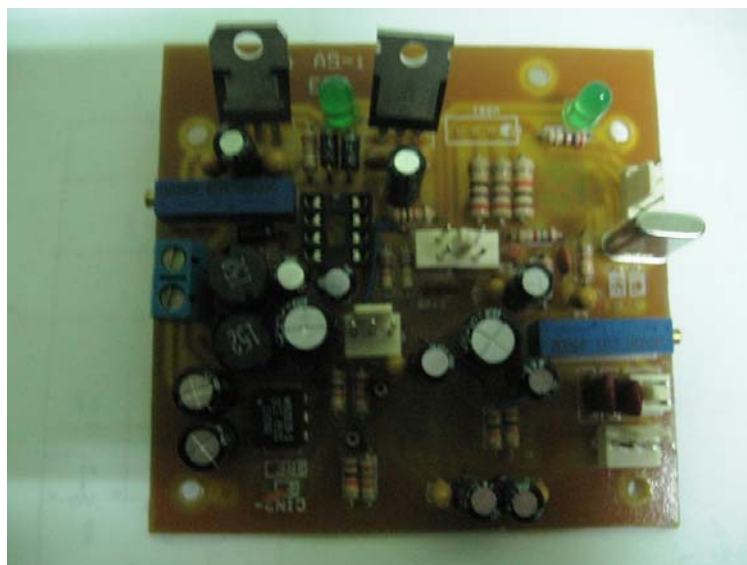


Figura 2.9 Hardware

## 3. SOFTWARE

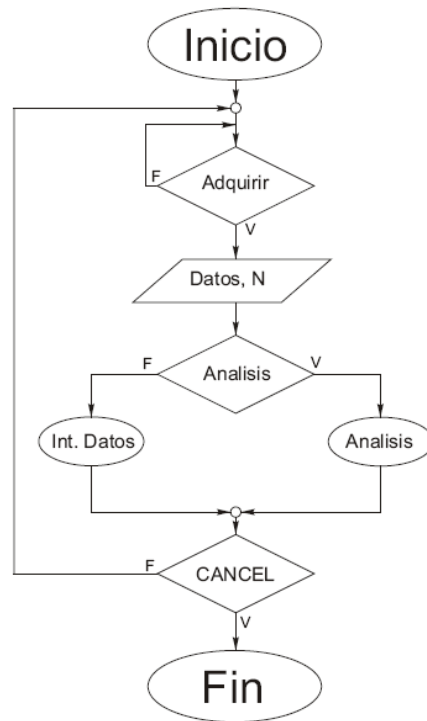
### 3.1 DIAGRAMA DE FLUJO DE LA INTERFAZ GRÁFICA

#### Variables Utilizadas en el diagrama de flujo

- **N** → Numero de elementos del vector de datos
- **Datos** → Vector de elementos leído del puerto
- **Adquirir** → Pregunta si esta habilitada la adquisición de datos
- **Análisis** → Pregunta si esta activada la lectura y visualización de datos o en su defecto se realiza el análisis de los datos leídos para la parte de Híbridos (Frió Caliente). (Intercambio de Datos o Completo)
- **END** → Pregunta por el estado del botón CANCEL de la interfaz y termina la ejecución del programa.
- **ERR** → Cuando es un dato de error enviado por el micro, esto es terminado en 00.
- **DIR** → la primera trama de maestro, corresponde a la dirección al cual se dirige la siguiente pregunta. Corresponda a D4,D3,D2,D1,D0,CB,0,1
- **PRE** → Corresponde a la segunda parte de la trama del maestro, aquí encontramos la pregunta que este realiza. I4,I3,I2,I1,I0,PB,1,0
- **RES** → Es la palabra que representa la respuesta dada por el esclavo I3,I2,I1,I0,PB,1,1.
- **i** → variable de control para la lectura del Buffer de entrada
- **ERROR** → Solo se presenta cuando la palabra en el buffer corresponde a ERR.
- **Retorna** → Siempre al final de una rutina, retorna y continúa la ejecución del programa principal desde donde es llamada.

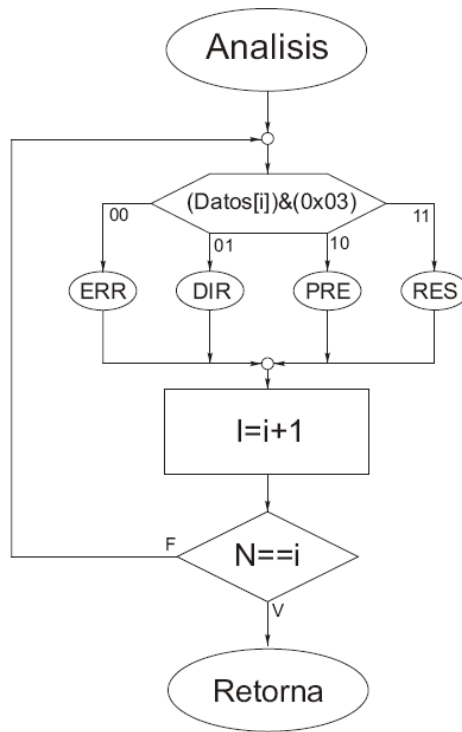
- **CB** → Control Bit.
- **REPITE** → Led que indica repetición de trama, esto es cuando el maestro hace una doble petición.
- **DETECTADO** → Coloca el esclavo en la lista de detectados y lo muestra.
- **DIRant** → Dirección anterior.
- **PREant** → Pregunta anterior. Estas dos últimas se guardan para verificar repetición de trama por parte de maestro.
- **Digital** → Indica que es un esclavo digital y su respuesta se publicara digitalmente.
- **Analógico** → Esclavo analógico, su respuesta se publica en valor analógico.
- **REPORTE** → Indica si es necesario generar reporte.
- **AC 0...18** → se usan para llevar el control de los bits enviados por el esclavo, para al final ordenarlos y darles su peso.
- **Val** → Bit de validación, se genera si el dato enviado por el esclavo es correcto, entonces se puede publicar el valor en los acumuladores.
- **AC[DIR]** → Acumulador, tiene el valor analógico del AC0.18.
- **Ov** → Indica overflow, se realizo una conversión analógico digital fuera del rango del conversor.
- **XXXX[DIR]** → Todos indican que para cada esclavo existe un indicador y una variable, es la misma lógica de un vector de datos.

En la figura 3.1, se puede encontrar, que el programa realiza una pregunta al usuario, acerca de si desea adquirir o prefiere estar en un estado inactivo. De resultar correcta esta afirmación, se procede a leer los datos existentes en el bus y la cantidad adquirida. Posteriormente, se pregunta si se realizará un análisis con todas las funciones programadas en el áreas internas (Incluidas las funciones para esclavos híbridos) o si se prefiere solo revisar el intercambio de datos, dependiendo de la respuesta, se selecciona una acción a ejecutar (Ver figura 3.1). Finalmente se habilita la opción de detener o no el proceso.



**Figura 3.1 Diagrama de inicio**

Si la decisión es realizar un análisis exhaustivo de la información en la red, estarán a disposición todas las funciones programadas. La figura 3.2 muestra una pregunta en forma de filtro, realizando una mascara de 03h a cada uno de los datos capturados, para extraer solo los dos primeros bits, los cuáles contienen la información de cómo reconstruir la palabra encontrada en la red, hace referencia al protocolo implementado para la comunicación entre el dispositivo y el PC, con explicación en la sección 3.4.3.



**Figura 3.2 Diagrama de Análisis**

La Figura 3.3 muestra el caso donde el resultado del filtrado sea 11b, correspondiente a una respuesta enviada por el esclavo encuestado, con base en esto se realiza la separación de datos, tomando la información necesaria para mostrarla al usuario de forma inmediata, si se trata de una respuesta de esclavo digital, de lo contrario se procede a acumular información (Proceso que se explicará con la figura 3.6).

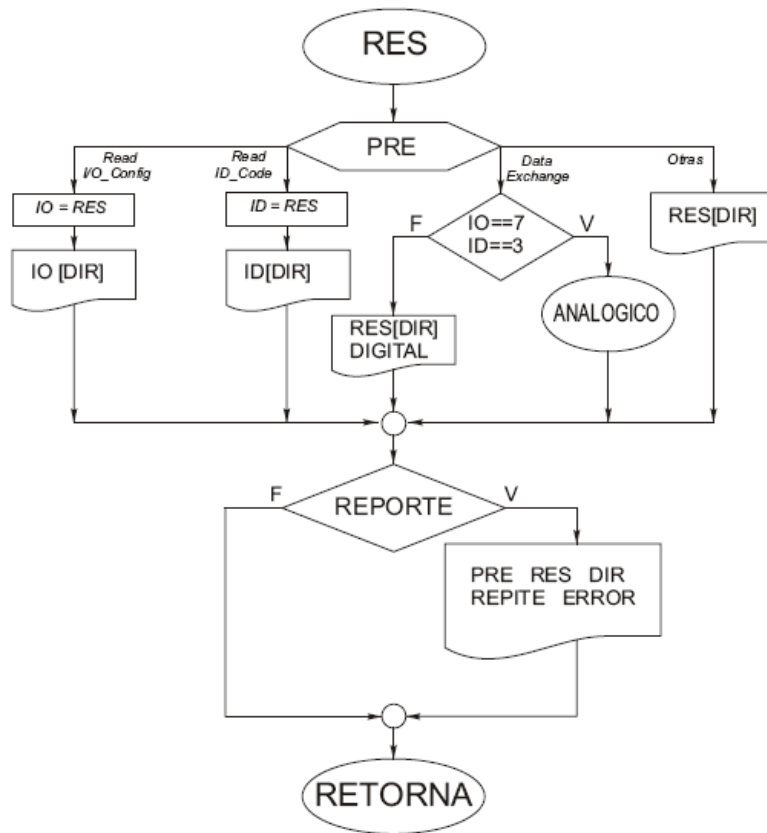


Figura 3.3 Diagrama de Respuesta

La Figura 3.4 muestra el caso en que los bits filtrados son 10b, correspondientes a una pregunta enviada por el maestro al esclavo cuya dirección es indagada en la figura 3.5. En esta instancia se almacena la pregunta y se imprime o se muestra en la casilla correspondiente a la dirección del esclavo que se realiza dicha petición.

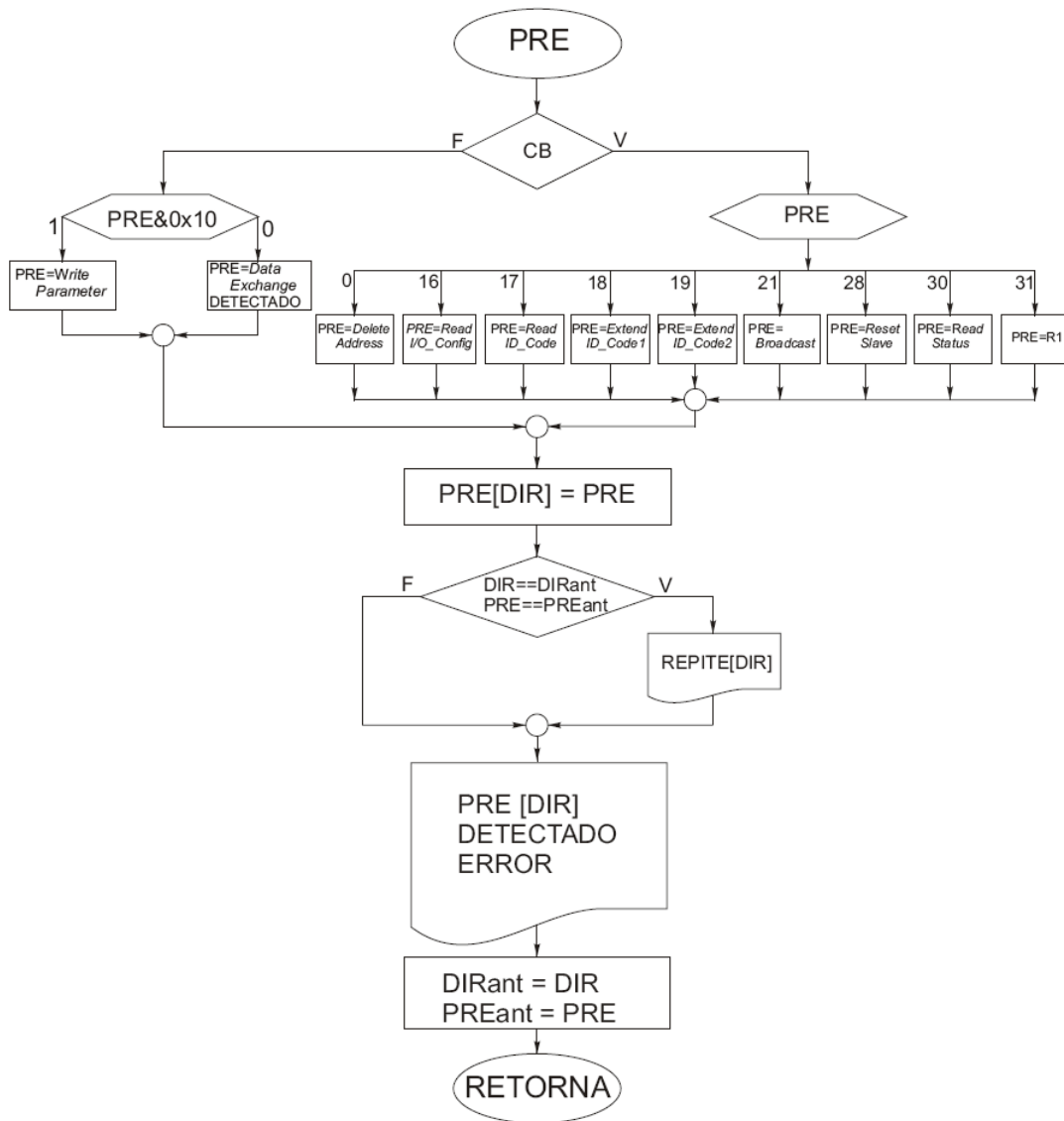


Figura 3.4 Diagrama de Pregunta

La Figura 3.5 muestra el caso en que los bits filtrados son 00b, correspondiendo a un error en la recepción de datos por parte del microcontrolador, obligando al programa a eliminar la dirección y la pregunta guardadas hasta que haya una nueva dirección en los datos. De tratarse de 01b, el procedimiento es guardar en un campo de memoria la dirección que contiene el dato recibido y esperar la pregunta, para de esta manera publicarla y esperar una respuesta de esclavo, para continuar con el proceso.

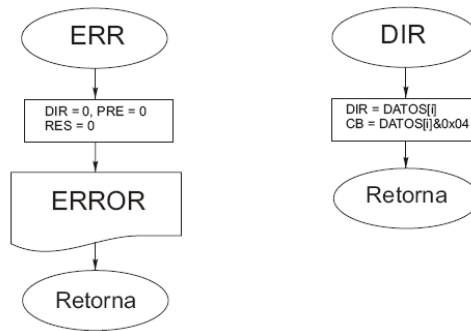


Figura 3.5 Diagrama de Dirección y Error

El analizador AS-i, esta capacitado para entender perfiles analógicos y descifrar la información que estos entregan. La Figura 3.6, muestra como se sincroniza el instrumento para capturar y ordenar los bits enviados en cada transacción hecha por el maestro, ya que cada esclavo analógico responde con 3 de los “n” bits necesarios para una respuesta completa, para más información acerca de configuración y perfiles de redes analógicas, ver [21].

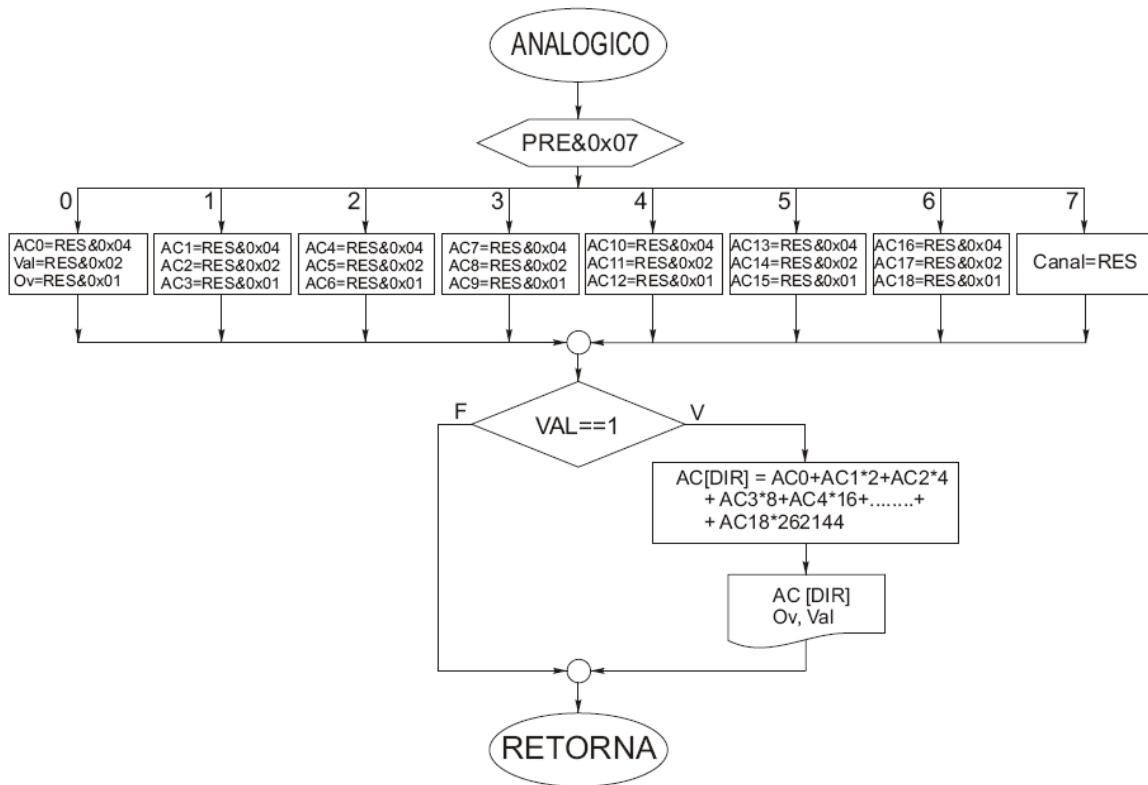


Figura 3.6 Diagrama de Analógico

## 3.2 INTERFAZ GRÁFICA

La interfaz en *Labview* del analizador implementado es una herramienta de fácil acceso y análisis preciso de una red *AS-Interface*, permite observar el intercambio de datos en el bus, decodificando todas las funciones implementadas por el maestro, analiza y provee un reporte del estado de la red, puede fácilmente identificar los esclavos de la red y el mal funcionamiento de cada uno de ellos por medio de *leds*. La interfaz cuenta con 6 partes que serán descritas a continuación, donde cada cual juega un papel importante en los procesos de decodificación, análisis, supervisión, almacenamiento e interpretación de los mensajes transmitidos a través de la red *AS-i* en modo estándar.

Los datos por defecto de la interfaz para la adquisición por puerto *USB* son listados en la tabla 3.1

<b>Puerto Serie</b>	COMx. Donde "x" es el número de puerto donde fue conectado el analizador, lo cual depende de cómo el computador tenga configurado sus puertos, p.e: COM3.
<b>Baud Rate</b>	725 200 bps
<b>Paridad</b>	None "ninguna"
<b>Bit de parada</b>	1 bit
<b>Bits de datos</b>	8 bits

Tabla 3.1 Datos por Defecto del analizador AS-I para adquisición por puerto USB

### 3.2.1 PANEL DE CONTROLES

En esta sección se encuentran los botones de mando para inicializar el programa, dicho panel se muestra en la figura 3.7 y se compone de los siguientes controles.



Figura 3.7 Panel de Control Analizador AS-I

- **Tipo de Adquisición:** Este botón es usado para activar o desactivar la visualización de la etapa híbrida, es decir permite a la interfaz mostrar los datos cuando se trabajan esclavos tanto analógicos como digitales.
- **Activar adquisición:** Este es el botón que activa o desactiva la adquisición de datos para la interfaz; el programa comienza a recibir datos por el puerto *USB* cuando dicho pulsador se encuentra activo.
- **Parar sistema:** Este es el botón que desactiva o cancela la adquisición de datos, al pulsarlo el programa deja de obtener datos del puerto *USB* y queda de nuevo listo para continuar recibiendo datos.
- **Fecha y Hora:** En esta etapa del panel de control se visualiza la fecha y hora del sistema; es un bloque de tipo informativo que puede ser de ayuda en muchos casos para el operario al momento de comparar alarmas y/o datos.
- **Bytes Read:** Finalmente quedó en la interfaz final, por que este era mas informativo para ustedes.

### 3.2.2 ESTADO DE LOS ESCLAVOS

En esta etapa del analizador se encuentran alojados los indicadores correspondientes al listado de esclavos activos de la red, ver figura 3.8; la pantalla tiene listado del esclavo #1, hasta el esclavo #31, según requerimientos de la norma [1], cada uno con tres *leds* que funcionan de la siguiente manera:

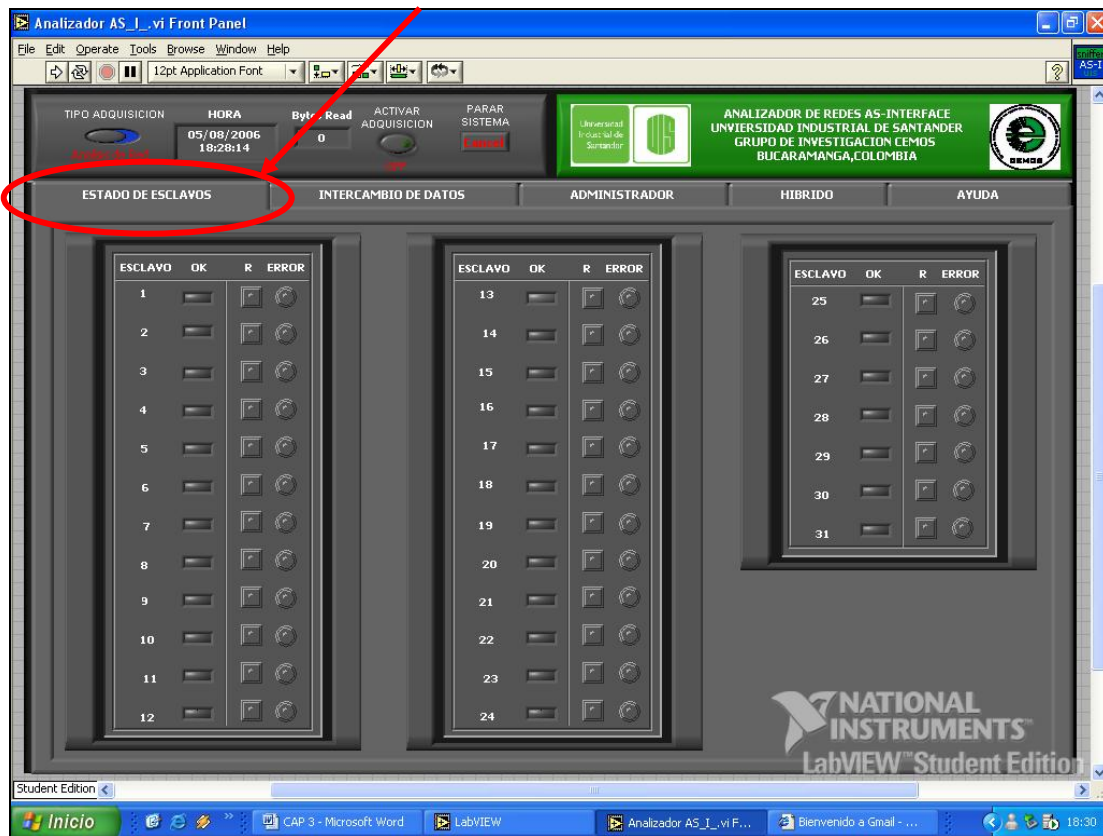


Figura 3.8 Panel de Estado de Esclavos en la Red

- **OK:** Este indicador posee dos estados, el primero es el estado por defecto “OFF” (Color Gris), el cual se muestra cuando el analizador no ha detectado el esclavo como componente activo de la red; el segundo estado se presenta cuando el esclavo es detectado por el analizador e inmediatamente la interfaz despliega un cambio de color gris a verde.

- **R:** Este indicador permite al usuario identificar una retransmisión por parte del maestro. En su estado por defecto refleja un color gris, en caso de activarse refleja un color naranja.
- **ERROR:** Es un *led* indicador que permite identificar cuando un esclavo que se encuentra en la lista de activos ha dejado de comunicarse con el maestro, en ese momento el *led* deja de ser un color rojo desactivando el *led* de esclavo activo llevándolo a gris, en caso contrario, el indicador manifiesta el color por defecto de la pantalla (gris).

La activación de estos *leds* no se realiza al mismo tiempo, es decir, sólo un indicador se visualiza activo mientras los otros dos están apagados, excepto cuando se hace retransmisión por una o dos veces, la interfaz conmuta la activación de *leds* para esclavo activo y retransmisión.

Esta herramienta es de utilidad para el usuario, permitiendo conocer los esclavos activos en la red, los esclavos con retransmisión constante de las peticiones del maestro y aquellos que por diferentes motivos han sido sacados de la lista de esclavos activos de la red.

### 3.2.3 INTERCAMBIO DE DATOS

En esta pantalla el usuario encuentra los esclavos listados igual que en la etapa anterior, pero adicionalmente visualiza 5 *bits* que hacen parte de la trama del maestro de la red, ver figura 3.9, los cuales corresponden a la petición enviada por el maestro. De igual forma también se visualizan 4 *bits* que representan la respuesta del esclavo a la respectiva petición. A manera de información adicional el analizador por medio de esta etapa muestra al usuario un cuadro donde se despliega el tipo de transacción que se ha ejecutado. Estos datos son actualizados

constantemente con una tasa de muestreo programable en la interfaz de usuario, permitiendo un valor mínimo de visualización de 1ms, ya que este es el tiempo especificado como mínimo para el driver USB utilizado [23].

En el intercambio de datos, solamente se encuentra información actualizada de aquellos esclavos que han sido detectados como activos por la etapa anterior (Estado de los esclavos), por tal razón para aquellos que no están en funcionamiento, estos *bits* se visualizarán inactivos, resaltados con un color gris opaco.

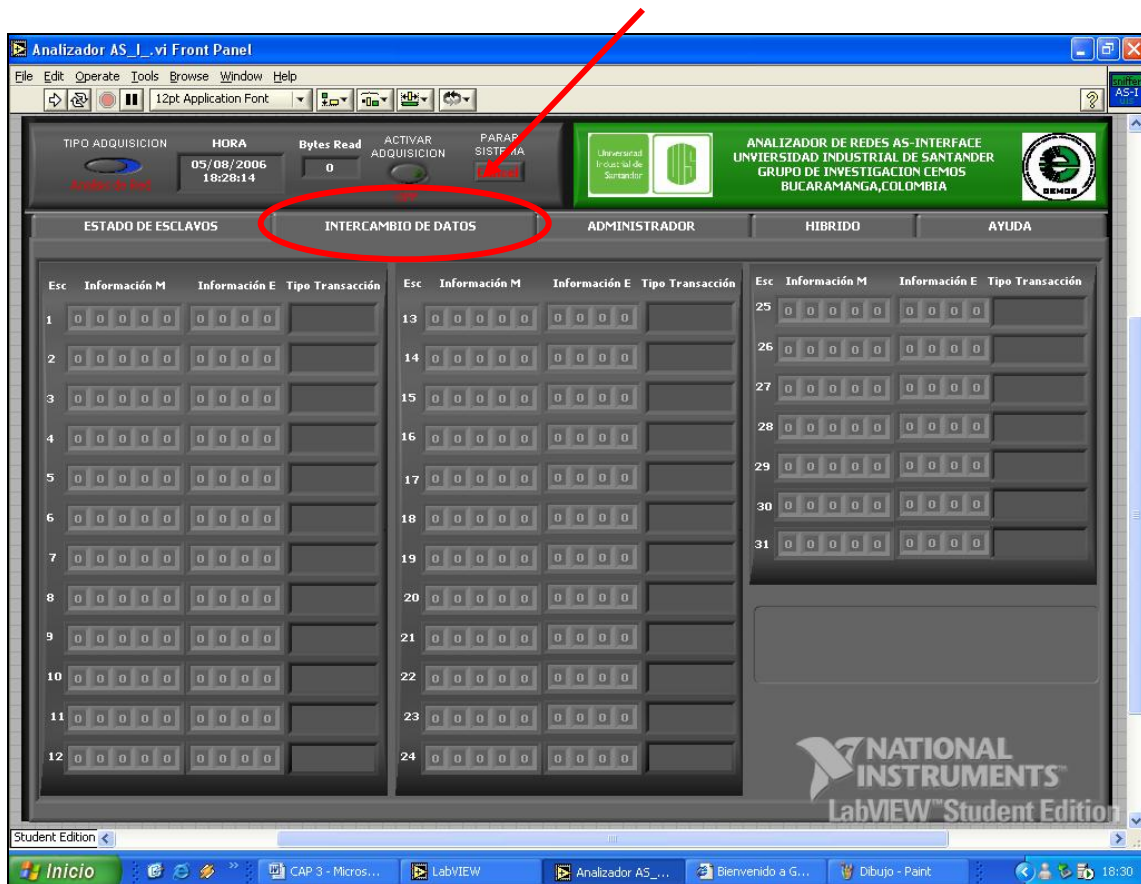


Figura 3.9 Panel de Intercambio de Datos de Una Red AS-I

### 3.2.4 ADMINISTRADOR

Es la sección del analizador que cumple la función de administración de datos, en ella se encuentran alojados dos paneles para reproducir gráficamente el arreglo binario de maestro y un esclavo elegido con anterioridad con la opción de la interfaz. Adicionalmente, se encuentra la opción de visualizar los reportes de la red hasta el momento almacenados en una hoja de cálculo; este archivo cuenta con la hora, la fecha, el tipo de transacción, los *bits* información de maestro y esclavo, el número de esclavo, el número de repeticiones hasta el momento realizadas a ese esclavo. Finalmente, cuenta con un fragmento de la interfaz donde se visualizan en formato hexagesimal los datos adquiridos por el analizador, teniendo en cuenta que este se eligió este formato debido a que esta herramienta implementada además de ser una aplicación industrial, es un desarrollo de tipo académico en el cual puede dar un gran aporte al proceso de enseñanza de los protocolos de comunicación; Además se da a conocer el dato del tiempo promedio entre fallas (*TMBF*) para cada esclavo activo en cierta cantidad de datos adquiridos (Figura 3.10).

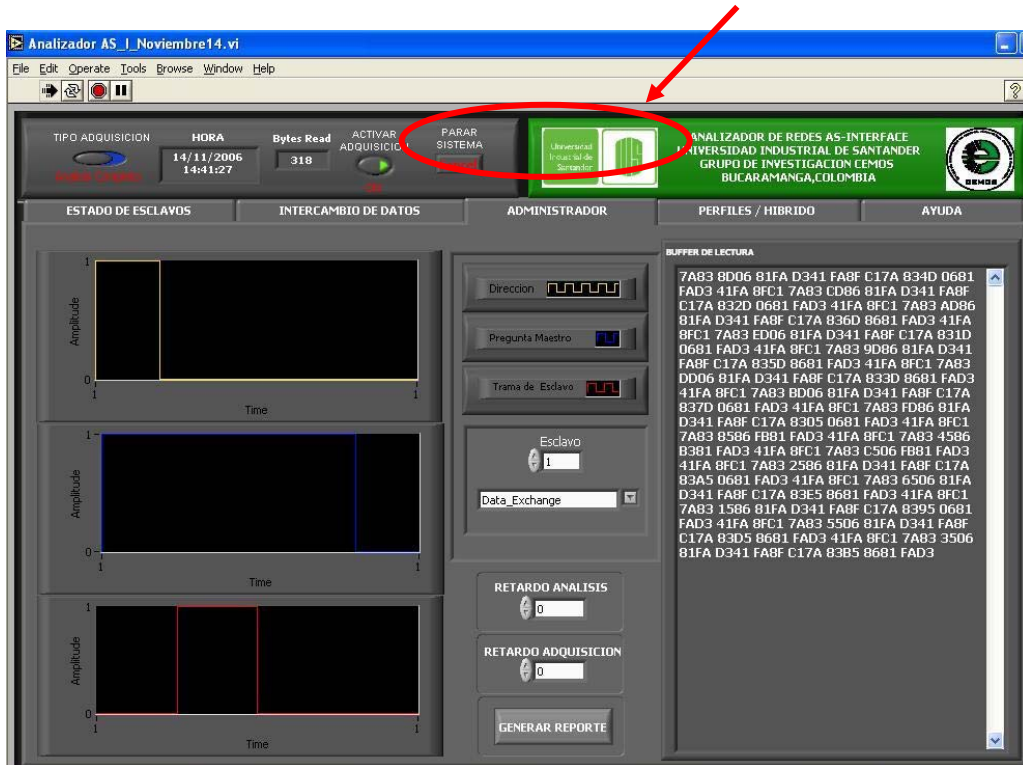


Figura 3.10 Pantalla Analizador Para una Red Híbrida AS-I

### 3.2.5 HÍBRIDO

En esta etapa de la interfaz se despliega información importante que definen el perfil de los esclavos AS-i conectados, como son la configuración de entradas salidas (*I/O Code*) y el identificador de fabricante (*ID Code*). Adicionalmente se implementó la opción de desplegar información para redes híbridas, donde se presenta la capacidad del analizador para identificar esclavos analógicos, visualizando indicadores de desbordamiento (*OVER*) y validación (V) (Figura 3.11). Hay que tener en cuenta que esta configuración híbrida no existe hasta el momento como solución de automatización en redes AS-i comerciales, fue un montaje de ámbito académico en una tesis de maestría de la Universidad Industrial de Santander [21].

La etapa híbrido visualizada por el analizador, está en capacidad de detectar cualquier tipo de perfil I/O, ID de esclavo digital y esclavos con perfiles analógicos I/O=7 e ID=3, perfil de estudio en el trabajo de maestría citado en [21]; sin

embargo se deja clara la dificultad de hacer una prueba en caliente, ante la ausencia de un implementación completa de un dispositivo maestro con capacidad de manejar este tipo de dispositivos, lo cual se alejaba de los alcances de dicho proyecto.

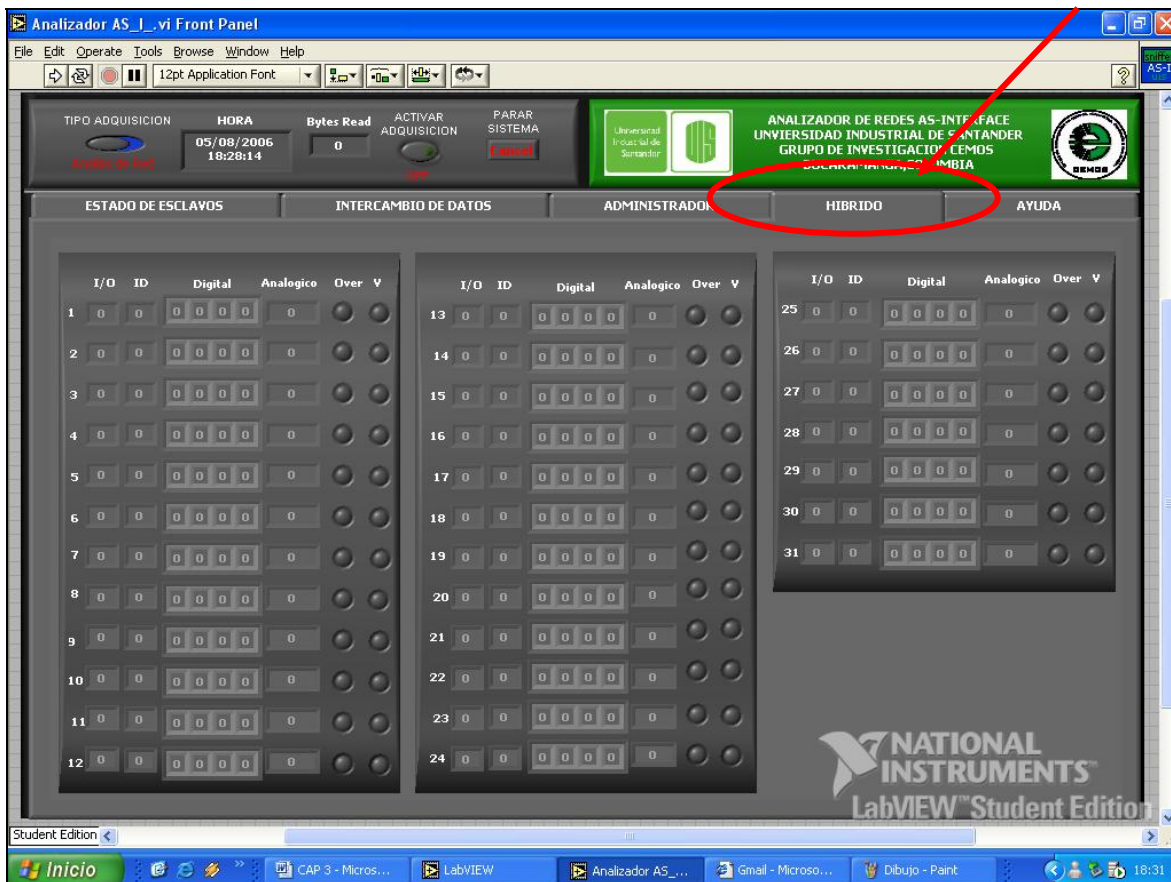


Figura 3.11 Pantalla Analizador Para una Red Híbrida AS-I

### 3.2.6 CRÉDITOS / AYUDA

En esta última etapa de la interfaz del analizador, se visualizan los nombres de los diseñadores del dispositivo, el director y codirector del proyecto, además de las

entidades patrocinadoras del proyecto, en la parte inferior izquierda se encuentra un enlace hacia un archivo donde se encuentra un manual de funcionamiento del analizador, en el cual se explica tanto las conexiones de *hardware* como el manejo del *software* (Figura 3.12).

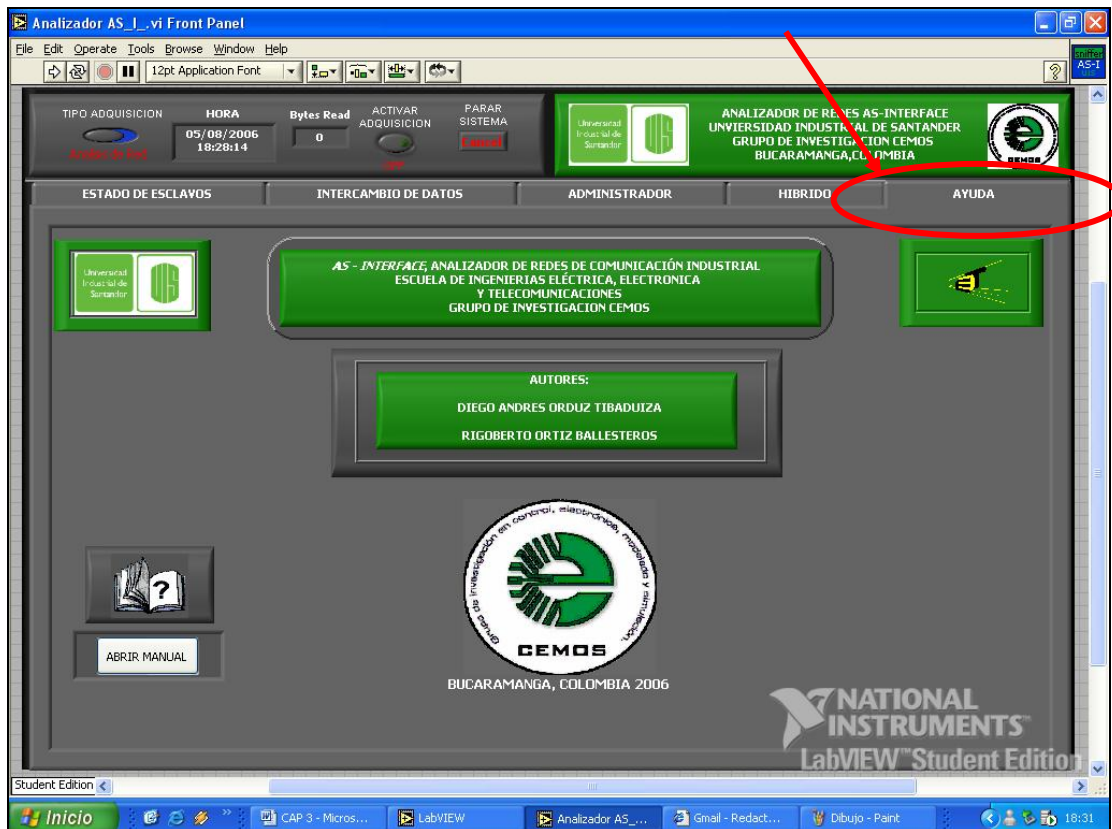


Figura 3.12 Pantalla de Créditos y Ayuda Analizador AS-I

### **3.3 DIAGRAMA DE BLOQUES DE INTERFAZ GRÁFICA**

El diagrama de bloques del desarrollo de la interfaz en *labview*, es anexado en el CD, junto a la entrega de los documentos del proyecto, esto debido a su complejidad y tamaño.

### **3.4 PROGRAMACIÓN DEL MICROCONTROLADOR**

Finalizando la capa física, se encuentra un dispositivo controlador el cual es el encargado de la decodificación de los pulsos digitales entregados por la etapa de recepción, además tiene la tarea de transmitir esta información al PC mediante la interfaz SCI.

#### **3.4.1 MICROCONTROLADOR 68HC12E128**

Controlador de gama alta de motorola “freescale”, con grandes prestaciones de velocidad, puertos de propósito general, periféricos de múltiples aplicaciones y alta capacidad de memoria *flash*. Optimo para aplicaciones que impliquen programas extensos con numerosas líneas de código [20]. En la aplicación presentada para la decodificación de los pulsos digitales y la posterior transmisión de datos al computador, aparentemente no se requiere tantas capacidades del dispositivo, sin embargo existen dos que sobresalen y toman gran importancia a la hora de diseñar.

Una de ellas es la frecuencia del bus interno del microcontrolador fijada a 25 MHz, ya que los pulsos digitales entregados por la etapa de recepción son de una rata de baudios que duplica a la rata de *bits* en la comunicación de la red, debido al proceso de codificación, es decir, el maestro en la red *AS-i* se comunica con cada uno de los esclavos a 166.67kbps, (un *bit* cada 6 $\mu$ s), y los pulsos se deben recibir a 333.33kbps, (un pulso cada 3 $\mu$ s). Teniendo en cuenta que se diseñó un sistema en linea, estos datos deben ser analizados y decodificados cada vez que se

reciben, de esta manera se va armando la palabra digital, que a su vez debe ser enviada al computador.

La segunda razón es la existencia de periféricos con capacidades superiores, que ayudan a mejorar las prestaciones ofrecidas por este controlador, tales como la existencia de un PLL, quien es el encargado de proporcionar la frecuencia interna del bus de datos. Una interfaz serial de comunicaciones asincrónica *SCI*, con velocidad de transmisión configurable. Temporizadores ajustables para obtener intervalos de tiempo muy exactos con un margen de error de la mitad del tiempo de un ciclo en el bus ver [20].

A continuación se hablará en detalle de cada uno de los periféricos utilizados.

#### ▪ PERIFÉRICOS UTILIZADOS

**PLL interno:** Da la capacidad al microcontrolador para elevar la frecuencia del bus de datos al valor seleccionado no superior a 25Mhz, debido a que es la máxima frecuencia configurable, utilizando un cristal de 8Mhz y activando el PLL mediante la herramienta de programación .

**TIMER:** Temporizador configurable, 7 canales. Da la capacidad de tener control muy exacto sobre los tiempos de ejecución en la opción *output capture* al igual que permite medir tiempos de ejecución de eventos externos con la opción *input capture*. El registro *TIM0\_TSCR2*, es un preescalador que ajusta la frecuencia de operación del contador. *TIM0\_TC7* es el registro que indica el valor hasta el cual se desea contar, esto para una configuración *output capture* (canal 7 del *Timer*). Este canal es configurado para generar un tiempo de espera de 3 $\mu$ s, equivalente a 75 ciclos del *TIMER*.

**SCI:** Interfaz de comunicación serial, con una tasa de baudios configurable, y opciones de comunicación habilitadas, como *bit* de paridad, *bit* de inicio, *bit* de parada y longitud de palabra. Cuenta con un registro llamado *SCIOBD* cuya función es la de un preescalador de frecuencia, encargado de dar la tasa de transmisión de datos. Un registro de transmisión y recepción, el cual permite escribir la palabra que será enviada o recibida (*SCIODRL*). Para transmitir datos al PC, se utiliza una tasa de 725200 *bps*.

## ▪ ESTRUCTURA

EL programa para la decodificación se escribió en código C, con la ayuda de la herramienta *CodeWarrior* sugerida por *freescale*. Es una estructura continua, ya que el microcontrolador siempre esta dispuesto para recibir datos y retransmitirlos al PC.

La etapa física entrega pulsos positivos de 0-5V, tomando como entradas los canales PTB2 y PTB3, los cuales siempre están dispuestos para la recepción de pulsos. En PTB3 esta la entrada de pulsos cada vez que se detecta un flanco positivo, es decir, cada vez que halla un cambio de 0V a 2V en la red. Mientras que en PTB2, se encuentran los pulsos que reflejan un flanco negativo, es decir, un cambio de 0V a -2V en la red. Este proceso se realiza después de un filtrado previo de la componente de continua de la red.

Hasta el momento se ha hablado bastante de la forma cómo se codifica la señal en la red AS-i. De esta manera se pueden extraer una serie de parámetros que permiten decodificar los datos encontrados, los cuales son:

- Si encontramos un pulso positivo seguido de uno negativo, se le llama **PNeg**, el cual indica un cero lógico.
- Si encontramos un pulso negativo seguido de uno positivo, se le llama **NPos**, el cual indica un Uno lógico.

- También se pueden encontrar descansos, o lugares en los que no existe ni un pulso positivo ni uno negativo. De aquí salen nuevas convenciones, un descanso seguido de un pulso negativo, indica un Cero lógico, mientras que un descanso seguido de un pulso positivo indica un Uno lógico. Denominados **DNeg** y **DPos** respectivamente.
- En forma general, la segunda mitad en la codificación Manchester, siempre esta reflejado el bit de información, esto es **X1** Representa **1**, mientras **X0** Representa **0**.

El programa esta estructurado de acuerdo a los lineamientos anteriores, entendienddo por un **PNeg** como una entrada valida en PTB3 y tres microsegundos después una entrada valida en PTB2, es decir, PTB3 tiene como entrada un uno lógico y tres microsegundos después PTB2 tiene una entrada uno lógico.

Existe un **NPos** si en determinado tiempo PTB2 presenta un uno lógico y tres microsegundos después PTB3 presenta un uno lógico.

**DNeg**, indica que en determinado tiempo PTB3 presenta un cero lógico y tres microsegundos después PTB2 se coloca en uno lógico.

**DPos**, indica que en determinado tiempo PTB2 presenta un cero lógico y tres microsegundos después PTB3 se coloca en uno lógico.

El programa inicia con la configuración de puertos, e inicialización de variables utilizadas y periféricos configurados.

**E** → Palabra de 8BITS en la que se almacena la información o respuesta del esclavo en caso de que solo se reciban 7 bits, o puede contener la información de la dirección del esclavo al cual va dirigida la pregunta del maestro.

**M** → Palabra de 8BITS en la que se almacena la información o pregunta realizada por el maestro.

**K** → Variable utilizada para indicar si el tiempo de espera es por la primera parte de un *bit* o es la segunda parte de un *bit*, ya que los datos están codificados en **MANCHESTER II**.

**L** → Variable utilizada para almacenar el número de *bits* que han sido leídos de la red mientras se acumula una palabra.

Siempre se esta recibiendo la lectura de PTB2 y PTB3, de esta manera se acumula la palabra **E**, hasta que el numero de *bits* es igual a siete, esto es hasta que la longitud **L** obtenga el valor de siete. Se continua con la lectura de los puertos y si después de 6 microsegundos no hay un resultado positivo en ninguno de los puertos de entrada se procede a trasmitir la palabra **E**, teniendo en cuenta una pequeña convención para el envió de datos.

*EB PB I0 I1 I2 I3 1 1*

Las letras utilizadas corresponden a la terminología manejada en [1].

En el caso que **L** supere los siete *bits*, se procede a acumular los demás *bits* en la palabra **M**, correspondiendo para la palabra **E** la siguiente convención.

*A0 A1 A2 A3 A4 CB 0 1*

De la misma manera la palabra **M** se envía al final de recibir los 14 *bits* del maestro con la siguiente convención.

*PB I0 I1 I2 I3 I4 1 0*

En los tres casos se respetan los valores fijos de los dos *bits* menos significativos, ya que estos siempre indican que clase de información es la que se recibe en *LabView*.

- 00** Mensaje de error, solo indica que hay un posible error en los datos transmitidos así que no se deben tener en cuenta los datos de dirección y pregunta almacenados.
- 01** Primera parte de la información en la trama de maestro, representa la dirección y el CB.
- 10** Segunda parte de la información obtenida en la trama del maestro, representa la pregunta realizada.
- 11** Respuesta del esclavo.

Finalmente el programa empieza de nuevo el proceso, habilitando la recepción de PTB2 como *bit* de inicio. Esto se puede observar más detalladamente en el diagrama de flujo presentado a continuación.

**3.4.2 DIAGRAMA DE FLUJO DEL MICROCONTROLADOR**

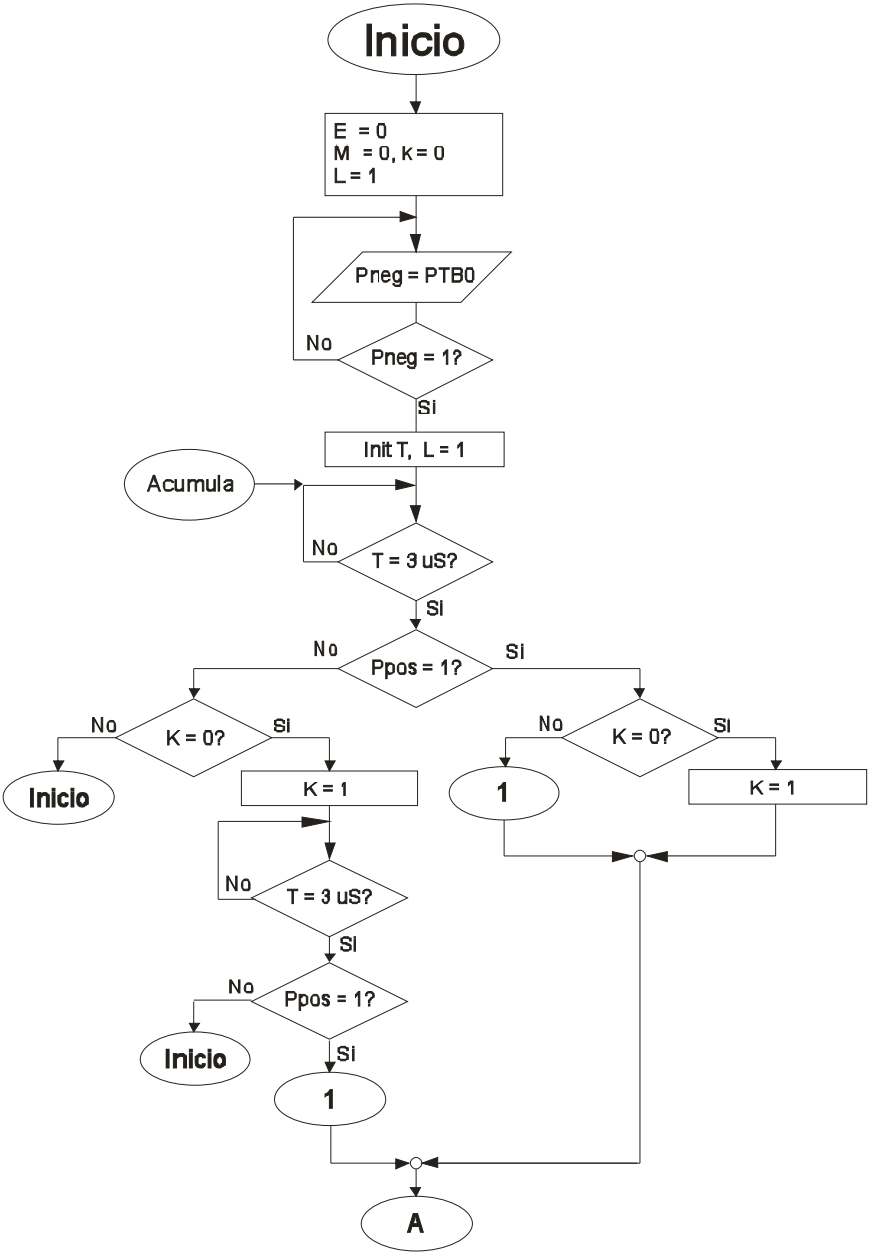


Figura 3.13 Diagrama Principal

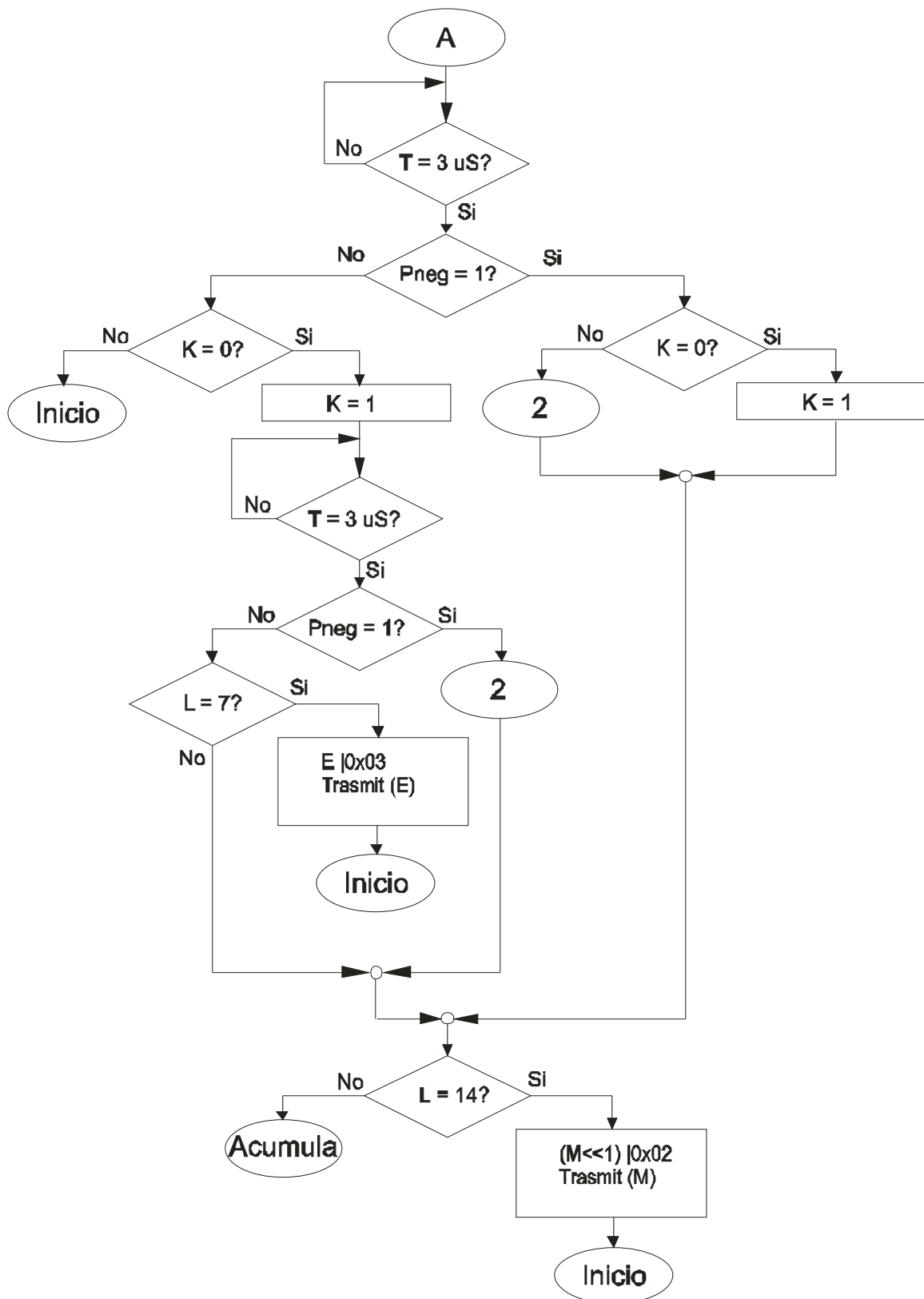


Figura 3.14 Diagrama Principal Primera parte

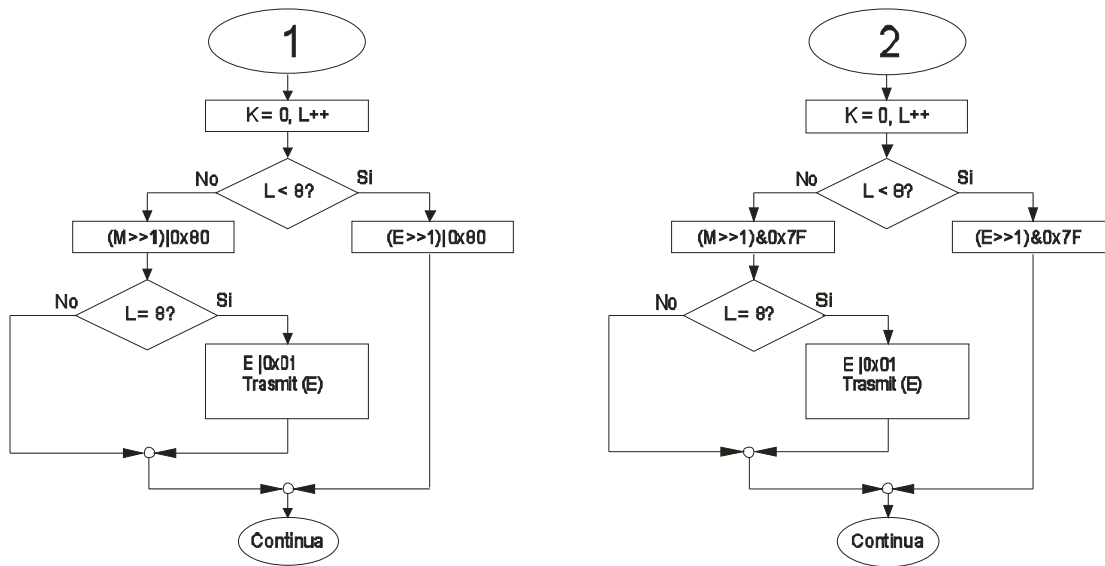


Figura 3.15 Diagrama Principal segunda parte

Las Figuras 3.13, 3.14 y 3.15 muestran en forma general todo el proceso de decodificación con descripción precisa anteriormente.

En este diagrama se señala la función **Trasmit()**, de la cual no se hizo alusión anteriormente, sin embargo, solo corresponde a la parte de envió de datos al PC, ver figura 3.16. Ya después de realizar la correspondiente configuración, esta función toma el valor encontrado en su campo de datos “Los paréntesis”, para escribirlo en el registro determinado por el fabricante del microcontrolador para el envió de datos por *SCI*. Ver Programa Anexo del microcontrolador [CD].

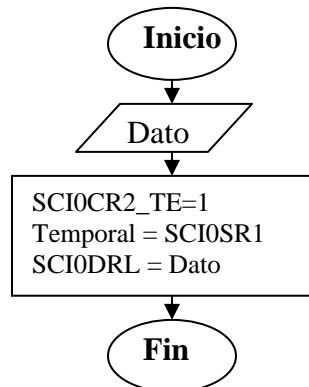


Figura 3.16 Diagrama Trasmit()

### 3.5 ANALIZADOR COMERCIAL

Dentro del campo de las comunicaciones industriales siempre se ha buscado integrar dispositivos que ayuden a mejorar la interacción del usuario con los elementos de automatización y control, es por ello que varias marcas de desarrollo industrial han encaminado sus esfuerzos en traer al mercado elementos con características de monitoreo y visualización de fácil manejo y buen desempeño.

Teniendo en cuenta las características y los requerimientos encontrados en el mercado, el dispositivo implementado en este proyecto cuenta con soluciones a las necesidades industriales; Aquí se muestra un elemento totalmente funcional, el cual posee características mejoradas a las de otros analizadores y cuenta con ideas innovadoras que suplen mas campos de trabajo (Ver Tabla 3.2).

<b>Características</b>	<b>Analizador Comercial</b>	<b>Analizador AS-I Implementado</b>
<i>Indicadores del Estado de Esclavos</i>	✓	✓
<i>Indicadores de Intercambio de Datos</i>	✓	✓
<i>Lector de perfiles de Esclavo</i>	✓	✓
<i>Estadísticas o Resumen de funcionamiento de la red</i>	✓	✓
<i>Informe de los datos de la red</i>	✓	✓
<i>Detección de perfiles de esclavos digitales</i>	✓	✓
<i>Opciones de filtro de información</i>	✓	✓
<i>Informe de datos de la red por eventos</i>	✓	✗
<i>Tiempo de adquisición y actualización inferior a 1s</i>	✗	✓
<i>Detección de perfiles de esclavos analógicos</i>	✗	✓
<i>Transmisión de Datos USB</i>	✗	✓
<i>Manejo de datos de una red híbrida</i>	✗	✓

**Tabla 3.2** Tabla de comparación Analizador Comercial vs Analizador Implementado.

## 4. PRUEBAS Y RESULTADOS

Una vez se ha implementado el analizador *AS-I* a nivel de *Hardware* y *Software*, es preciso validar el dispositivo, por medio de pruebas de funcionamiento sobre una red comercial. En este capítulo se presentan los resultados obtenidos a través de las medidas y registros realizados con el osciloscopio a través de las diferentes etapas del dispositivo *AS-I*, haciendo una comparación de estos resultados con los requerimientos establecidos por la norma EN50295 [1].

### 4.1 Señal de Recepción

La norma establece como válidos aquellos pulsos analógicos que sean superiores a  $1.5 V_{pp}$ . Utilizando este valor de tensión para determinar las amplitudes válidas, la etapa de recepción fue ajustada para generar pulsos positivos una vez sea superada el nivel de comparación.

La figura 4.1, en la parte superior muestra la señal del bus *AS-I* después de la etapa de desacople; además en la parte inferior de esta figura se visualiza la señal transmitida al computador desde la *SCI* del microcontrolador para una transacción de maestro y esclavo, observando claramente que para una transacción de maestro se envían dos *bytes* hacia el microcontrolador y para una respuesta de esclavo se trabaja un *byte*.

En la figura 4.2 podemos captar una señal mas completa de las transacciones en el bus, donde se pueden detallar tres transacciones de maestro al mismo número

de esclavos, haciendo la aclaración que en esta ocasión solo se presentaron dos respuestas de esclavos.

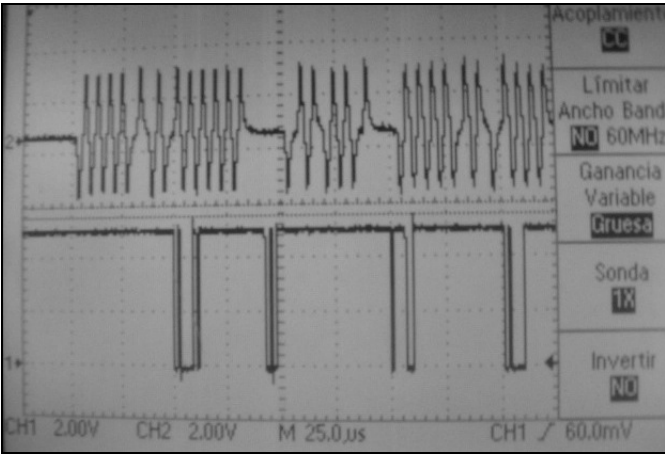


Figura 4.1. Señal de la red AS-I y SCI del microcontrolador

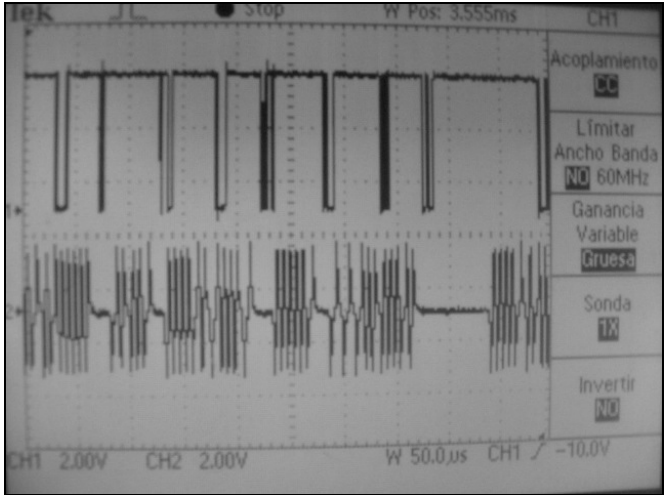


Figura 4.2. Señal de la SCI del microcontrolador y red AS-I con mensajes de varios esclavos

### 4.1.1 Señal de Maestro

En la figura 4.3 se captura una petición de maestro tomada de la señal después de desacoplar la componente de continua del bus. Aquí se observa la amplitud de la señal (aproximadamente  $2.5 V_p$ ) y también se puede verificar que el ancho total de la señal adquirida es de  $84\mu s$ , teniendo en cuenta que existe un descanso de

3 $\mu$ s antes del primer flanco de bajada, cumpliendo con las características de la norma.

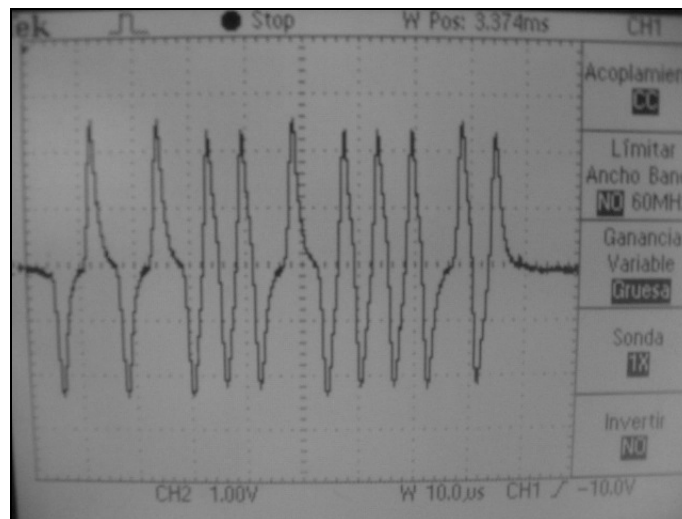


Figura 4.3 Petición de maestro

Haciendo un seguimiento de la señal de maestro dentro del circuito de recepción y tomando muestras a la salida del comparador, se obtiene la señal característica de detección para pulsos negativos y positivos en el microcontrolador como se muestra en la figura 4.4.

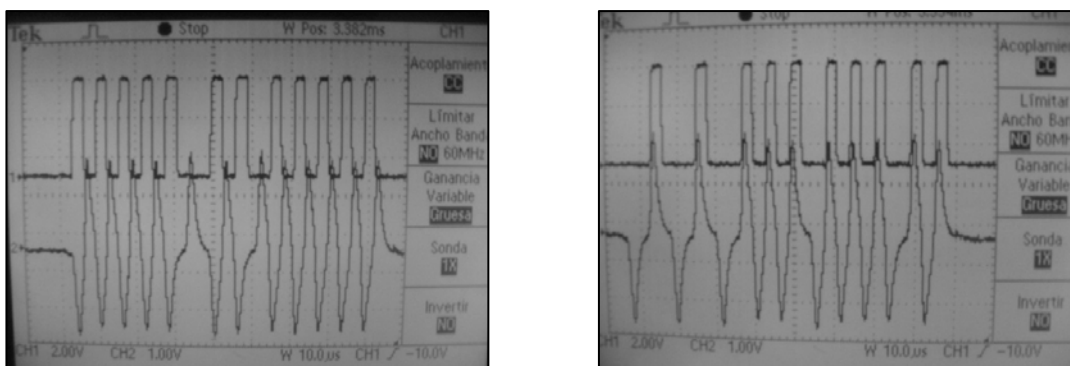


Figura 4.4. (a) Detección de pulsos negativos de maestro (b) Detección de pulsos positivos de maestro

#### 4.1.2 Señal de Esclavo

En la figura 4.5 se captura una respuesta de esclavo tomada de la señal después de desacoplar la señal de continua del bus. Aquí se observa que el ancho total de

la señal adquirida es de  $42\mu\text{s}$ , incluyendo el descanso de  $3\mu\text{s}$  antes del primer flanco de bajada, cumpliendo con los parámetros de la norma [1].



Figura 4.5. Respuesta de esclavo

Haciendo un seguimiento de la señal de esclavo dentro del circuito de recepción y tomando muestras a la salida del comparador, se obtiene la señal característica de detección para pulsos negativos y positivos en el microcontrolador como se muestra en la figura 4.6.

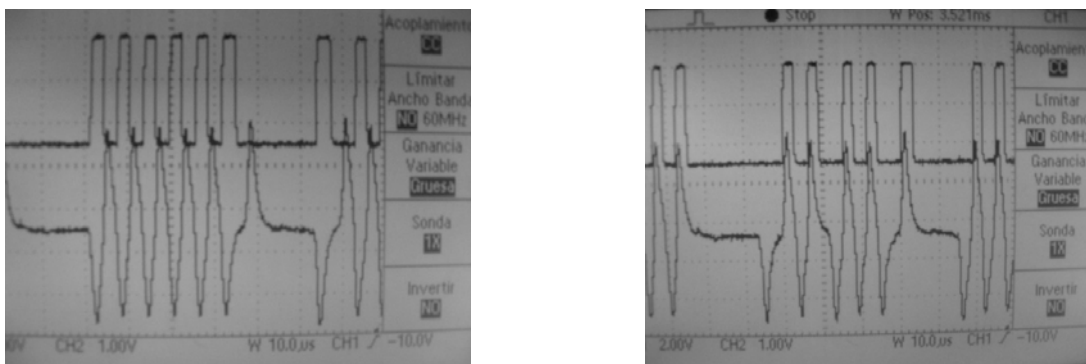


Figura 4.6. (a) Detección Pulsos Negativos de esclavo (b) Detección pulsos positivos de esclavo

## 4.2 Especificaciones de la fuente de alimentación

La fuente de alimentación diseñada, genera dos referencias internas de voltaje,  $+5\text{V}$  y  $-5\text{V}$ , que son utilizadas para alimentar los dispositivos internos del

analizador, junto con la tierra del mismo. La tabla 4.1 muestra los valores medidos en esta etapa, en los cuales se incluyen las tensiones intermedias previas. Allí también se incluye el valor del rizado máximo que se puede presentar en condiciones de plena carga del dispositivo durante su operación.

Tensión [V]	Rizo Máx. [mV <sub>pp</sub> ]	%Rizo
30	17	0.057
10	17	0.17
5	17	0.34
-4.7	35	0.744

**Tabla 4.1 Mediciones Fuente de Alimentación**

Los valores de rizado máximo medidos en la fuente de alimentación, son lo suficientemente pequeños comparados con los niveles de tensión generados. En ninguna de las pruebas el porcentaje de rizado supera el 1 %, de manera que no afectan el funcionamiento del dispositivo analizador, garantizando una tensión regulada para todos sus elementos internos. Otro factor importante es el consumo de corriente del analizador, ya que para la implementación realizada éste no supera los 92mA, siendo un valor menor a los 150mA establecidos como consumo máximo en la norma EN50295 para dispositivos conectados a la red [1].

### 4.3 Emulador de Red Híbrida

Se simularon dos esclavos de tipo analógico en las direcciones 2 y 3, teniendo estos valores analógicos una entrada de 3V y 4V respectivamente, representados en un valor digital de 16 *bits*; Además se emuló un esclavo digital con dirección 5 y características de entradas y salidas en “1” lógico. Obteniendo así una respuesta satisfactoria de detección de datos como se observa en la figura 4.7.



Figura 4.7 Pantalla Del Analizador con Red Híbrida

#### 4.4 Detección del Módulo Esclavo diseñado por el grupo AS-i de la Universidad Industrial de Santander

Configurando el esclavo con dirección 25, se obtuvieron los resultados que se presentan a continuación en la figura 4.8, validando la funcionalidad del analizador para direcciones superiores a las existentes en la red de pruebas (Red comercial).





Figura 4.8 Pantalla Del Analizador Detectando un Esclavo Diseñado por el Grupo AS-/UIS y Su Respectiva Transferencia de Datos

#### 4.5 Interfaz Gráfica del Analizador AS-/ en una red comercial.

Durante las pruebas, se trabajó con un par de esclavos de la red comercial suministrada por el Grupo de Investigación en Mecatrónica de la Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander, los cuales poseían las direcciones fijas #2 y #3; a partir de ahí, se anexaron a la red dichos esclavos y posteriormente intencionalmente se sacaron de línea uno a uno para poder detectar la pérdida de comunicación entre el maestro y cada uno de ellos, los resultados de estas acciones se ven reflejadas en las figuras que se muestran a continuación .

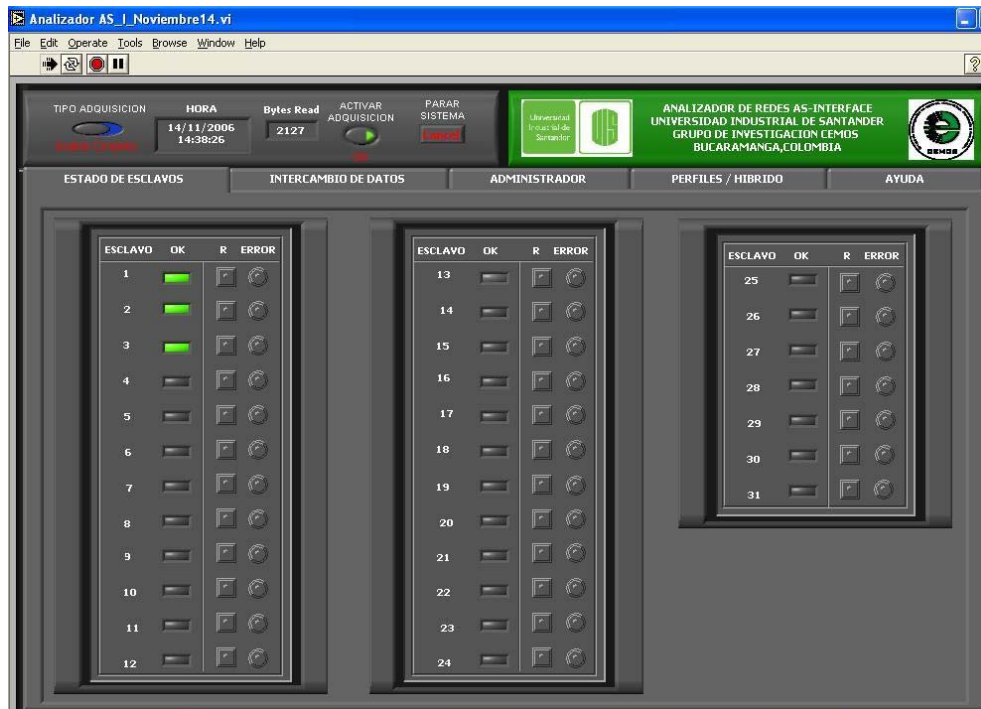


Figura 4.9 Pantalla Del Analizador Estado de Esclavos con red Comercial

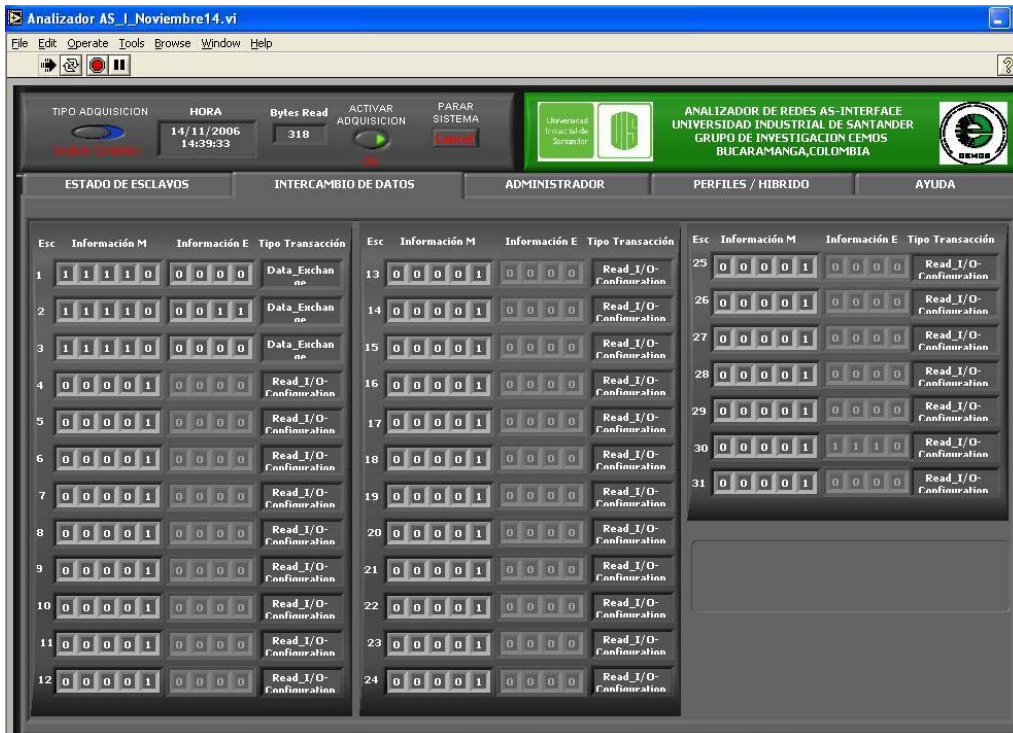


Figura 4.10 Pantalla Del Analizador en Intercambio de Datos



Figura 4.11 Pantalla Del Analizador en Estado de Esclavos con un Esclavo Fuera de Línea

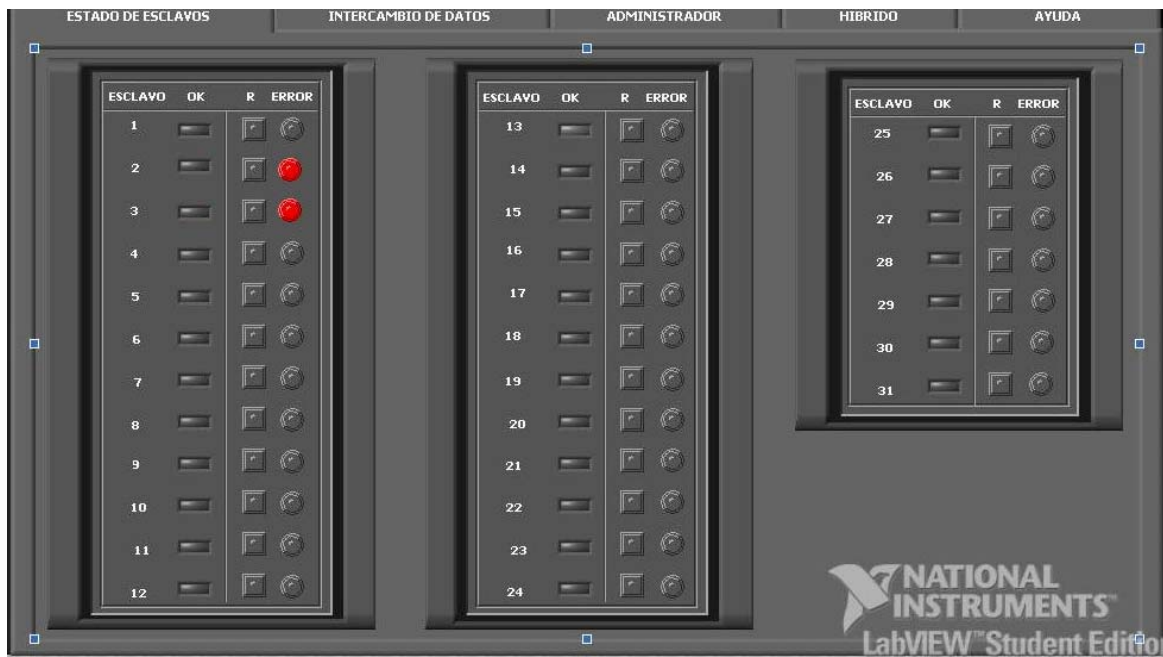


Figura 4.12 Pantalla Del Analizador en Estado de Esclavos con dos Esclavos Fuera de Línea

En resumen el analizador cumple con los rangos y características establecidas en la norma EN50295 en cuanto a consumo de corriente para dispositivos alimentados por la red; el analizador tiene capacidad para detectar cualquier perfil de esclavo de una red AS-*i* en modo normal con un máximo de 31 esclavos [1]. Los procesos de desacople para la extracción de potencia y la recepción de señal son realizados de manera exitosa por parte del dispositivo implementado.

Finalmente es necesario destacar que todas las medidas realizadas fueron tomadas con la puesta en marcha del analizador (*Sniffer*) AS-*I* sobre una red comercial, validando su funcionamiento como componente pasivo de la red que no altera la integridad física de la señal intercambiada por sus componentes activos, demostrando así que se mantiene el equilibrio de impedancias del canal de comunicación, mediante el reconocimiento de todas las transacciones requeridas por el maestro y las respuestas de los esclavos; además se verificó el correcto funcionamiento del módulo híbrido del *Sniffer* emulando la presencia de esclavos analógicos en la red.

## 5. OBSERVACIONES

- Se implementó un analizador *AS-Interface* que es compatible con los requerimientos expuestos de la norma EN50295.
- El desarrollo de *hardware* y *software* debe ser un proceso de realimentación conjunta, que permita la depuración adecuada de la implementación final que se desee realizar. Con esto se garantiza avanzar en una misma dirección, teniendo en cuenta los factores de incidencia.
- Se generó una interfaz física genérica capaz de extraer potencia e información del bus *AS-I* para la alimentación del circuito y análisis de datos respectivamente.
- El dispositivo es capaz de analizar cualquier tipo de transacción emitida en una red *AS-I* comercial en modo estándar, además tiene la capacidad de detectar esclavos digitales con cualquier perfil y esclavos analógicos con perfil I/O 7 ID 3.

## 6. RECOMENDACIONES Y FUTUROS DESARROLLOS

- Integrar todas las etapas en una tarjeta única, con el fin de brindar una solución mas compacta desde el punto de vista de *hardware*.
- Mejorar la aplicación en *Labview* para ampliar la cobertura de lectura a transacciones en modo extendido.
- Mejorar la aplicación en *Labview* para posibilitar la captura de datos para un tiempo especificado ( Filtro o ventana de tiempo ) .
- Ampliar la aplicación en *Labview* para nutrir la información desplegada en la etapa de administrador a partir de los datos de la red.

## 7. CONCLUSIONES

- Con el propósito de tener un analizador plenamente funcional mediante su conexión al bus, se implementó un circuito que permite la extracción de potencia del bus de comunicación, para la alimentación de los circuitos internos del dispositivo. Este circuito genera voltajes de referencia de 5V, -5V y tierra.
- En este trabajo se realizó la implementación de un analizador *AS-interface* en el microcontrolador MC9S12E128, cumpliendo con los requerimientos mínimos expuestos en la norma EN50295[1]. Este microcontrolador fue seleccionado debido a los límites de velocidad máxima de su interfaz *SCI* para poder alcanzar las exigencias de transmisión y los requerimientos de tiempo.
- La programación del microcontrolador fue realizada en lenguaje C, por ser un lenguaje de programación de alto nivel y hace parte del estado del arte.
- Se diseñó e implementó una interfaz gráfica en *Labview* capaz de analizar y supervisar la comunicación de una red industrial *AS-I* en modo estándar.
- El analizador *AS-I* está diseñado para detectar cualquier perfil de esclavo digital, incluyendo además la detección de las transacciones: *Data\_exchange*, *Write\_parameter*, *Reset\_slave*, *Read\_I/O\_configuration*, *Read\_identification\_code*, *Read\_Status*, entre otras.

- Las medidas realizadas fueron tomadas con la puesta en marcha del analizador (*Sniffer*) AS-I sobre una red comercial, validando su funcionamiento como componente pasivo de la red que no altera la integridad física de la señal intercambiada por sus componentes activos, demostrando así que se mantiene el equilibrio de impedancias del canal de comunicación, mediante el reconocimiento de todas las transacciones requeridas por el maestro y las respuestas de los esclavos.
- El dispositivo implementado cuenta con una etapa adicional la cual permite trabajar con sistemas híbridos (Analógico y Digital), detectando datos para esclavos analógicos con perfil  $I/O=7$ ,  $ID=3$  y mostrando el valor analógico que dicho esclavo posee.

## ANEXOS

- *Datasheets*
- Emulador Híbrido – Micro
- Programación Microcontrolador
- Programación *Labview*
- *Drivers USB*
- Simulaciones Matlab

## REFERENCIAS

- [1] **ACTUATOR SENSOR INTERFACE COMPLETE SPECIFICATION VERSION 3.0**, Revision 0 , AS-Interface® . September 17, 2004.
- [2] **REDES DE COMUNICACIONES INDUSTRIALES**, Monografía. Julio Augusto Gelvez Figueredo; Universidad Industrial de Santander, 2002.
- [3] **REDES DE COMUNICACIÓN INDUSTRIAL BASADAS EN PROFIBUS Y AS-INTERFACE**. Tesis de Grado. José Norberto Díaz Gutiérrez - Edison Martínez Oviedo; Universidad Industrial de Santander 2002.
- [4] **LABVIEW, PROGRAMACIÓN GRÁFICA PARA EL CONTROL DE LA INSTRUMENTACIÓN**. Lázaro Antonio Manuel, 1997.
- [5] **MODBUS. MONITOREO DE LA RED EMPLEANDO LABVIEW**. Tesis de Grado. Pedro Ardila Albarracín – Sinle Marcela Carreño Martínez; Universidad Industrial de Santander, 2005.
- [8] **LABVIEW 7 EXPRESS, USER MANUAL**. National Instruments. & **LABVIEW BASICS I & II COURSE MANUAL**. National Instruments September 2002.
- [9] **LABVIEW ADVANCED PROGRAMMING TECHNIQUES**, Rick Bitter - Taqi Mohiuddin - Matt Nawrocki; CRC Press LLC 2001.
- [10] **UNIVERSAL SERIAL BUS IN INDUSTRIAL MEASUREMENT AND CONTROL APPLICATIONS**, Geral Kupris; Motorola Semiconductors Productos Sector.

[11] **USB DESING BY EXAMPLE, A PRACTICAL GUIDE TO BUILDING I/O DEVICES**, Jhon Hyde; Intel.

[12] **REPOTENCIACIÓN Y ACTUALIZACIÓN DE UN POTENCIOSTATO GALVANOSTATO PRINCETON MODELO 363 PARA EL LABORATORIO DE CORROSIÓN DE LA ESCUELA DE INGENIERÍA METALÚRGICA Y CIENCIA DE MATERIALES**, Tesis de Grado. Diego Fernando Monroy Jácome - Sergio Andrés Sabala Vargas; Universidad Industrial de Santander 2005.

[13] **CMOS DATA CONVERTERS FOR COMMUNICATIONS**, Mikael Gustavsson - J. Jacob Wikner – Nianxiong Nick Tan; *Kluwer academic publishers*.

[14] **ANALOG DEVICES. AD8031/AD8032 DATASHEET, RAIL-TO-RAIL I/O AMPLIFIERS**, 1999.

[15] **ANALOG DEVICES. AD8612 DATASHEET, SINGLE SUPPLY COMPARATORS**, 2000.

[16] **ANALOG DEVICES. AD8012 DATASHEET, LOW POWER AMPLIFIER**, 2003.

[17] **ANALOG DEVICES. ADUM1200 DATASHEET, ISOLATOR**, 2003.

[18] **C/C++ CURSO DE PROGRAMACIÓN, FRANCISCO J. CEBALLOS.. ALFAOMEGA, 2ND Ed., 2002.**

[19] **COILCRAFT. WB2010-PC DATASHEET, WIDEBAND RF TRANSFORMERS**, 2004.

[20] **MOTOROLA. HC12 CPU12 REFERENCE MANUAL.**