

**MÓDULO DE INTEGRACIÓN  
DE SENSORES Y ACTUADORES LÓGICOS  
BASADO EN EL PROTOCOLO AS-I  
DE COMUNICACIONES INDUSTRIALES**

OMAR LEONARDO NÚÑEZ GUALDRÓN

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**  
Facultad de Ingenierías Físico Mecánicas  
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones  
Bucaramanga

2006

# MÓDULO DE INTEGRACIÓN DE SENSORES Y ACTUADORES LÓGICOS BASADO EN EL PROTOCOLO AS-I DE COMUNICACIONES INDUSTRIALES

OMAR LEONARDO NÚÑEZ GUALDRÓN

Trabajo de investigación para optar al título de  
Magíster en Ingeniería Electrónica

Director: Prof. MPE. Jaime Guillermo Barrero Pérez

Codirector: Prof. MSc. Jorge Enrique Meneses Flórez



**UNIVERSIDAD INDUSTRIAL DE SANTANDER**  
Facultad de Ingenierías Físico Mecánicas  
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones  
Bucaramanga

2006

*A DIOS Todopoderoso*  
*A mis padres Reynaldo y Maria Luisa*  
*A mi hermana Nahir*  
*A mi tío Samuel*  
*A mi novia Luisa Fernanda*

*“No basta saber, se debe también aplicar.  
No es suficiente querer, se debe también hacer.”*  
Johann Wolfgang Goethe

# Dedicatoria

A DIOS, por darme la sabiduría y fortaleza para seguir adelante.

A mis padres, Reynaldo y Maria Luisa, quienes son y seguirán siendo gestores de mis más grandes éxitos, arquitectos de esta nueva obra.

A mi hermana Nahir, por su apoyo y aliento incondicional en todo momento.

A mi novia Luisa Fernanda, por su amor, entrega, paciencia y gran fuerza de motivación.

A mi tío, Samuel, por su confianza y apoyo firme en mis deseos de superación.

A todos mis familiares, quienes directa o indirectamente aportaron su grano de arena para la culminación de este nuevo proyecto.

*Omar Leonardo Núñez Gualdrón*

# Agradecimientos

Muchas personas contribuyeron al desarrollo de este trabajo de investigación, por tal motivo el Autor expresa sus agradecimientos a:

Profesor Jaime Guillermo Barrero Pérez, Director del proyecto, por sus aportes, orientación y total confianza.

Profesor Jorge Enrique Meneses Flórez, Codirector del proyecto, por su apoyo constante e importante gestión.

Los docentes de la Maestría, de quienes tuve la oportunidad de recibir y compartir valiosos conocimientos e innumerables experiencias.

Profesor Jose Alejandro Amaya Palacio, quien por su gran disposición y colaboración fue parte vital del desarrollo de este proyecto.

A los estudiantes de ingeniería electrónica Arnol Marriaga, Oscar Abreu, Danny Guiza, Jose Contreras, Rigoberto Ortiz y Diego Orduz, quienes con el desarrollo de su proyecto de grado hicieron importantes y valiosos aportes al desarrollo de este trabajo de investigación.

A todos mis amigos y compañeros, que hicieron parte de este gran proceso, por sus aportes y experiencias, ellos perduraran en toda mi existencia.

Al Dr. Andreas Schiff, quien como asesor del foro de expertos de la Asociación Internacional AS-Interface, hizo valiosos aportes respecto al conocimiento y asimilación del protocolo.

Al Sr. Michael Bryant, por su valiosa gestión, al proporcionar la versión actualizada de la norma IEC 62026-2 que rige el protocolo.

Al Dr. Klaus Pffizer, por su impotante aporte, al proporcionar muestras de los ASIC comerciales que manejan el protocolo AS-Interface.

Gracias a todos por su compañía durante esta nueva oportunidad, por su motivación y palabras de aliento, por su apoyo firme e incondicional en cada adversidad. Hace un tiempo se inició una nueva etapa y hoy nuevamente se recoge el fruto de esta gran cosecha, donde germinaran nuevas oportunidades.

# Índice General

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.2. Justificación . . . . .	3
1.3. Objetivos del trabajo de investigación . . . . .	4
1.4. Alcances del proyecto de investigación . . . . .	4
1.5. Estructura del documento . . . . .	5
<b>2. Marco Contextual</b>	<b>7</b>
2.1. Revisión del estado del arte . . . . .	7
2.2. Modelo de referencia OSI . . . . .	11
2.3. Sistemas de Comunicación Digital . . . . .	13
2.4. Comunicaciones Industriales . . . . .	16
2.4.1. Bus de Campo . . . . .	18
<b>3. Protocolo de Comunicación AS-Interface</b>	<b>19</b>
3.1. Generalidades . . . . .	19
3.2. AS-Interface y el Modelo OSI . . . . .	20
3.3. Tipos de Interfases . . . . .	21
3.4. Señal Física . . . . .	21
3.5. Proceso de Comunicación . . . . .	23
3.6. Tipos de Transacciones . . . . .	24
3.7. Manejo de Variables Analógicas . . . . .	27
3.7.1. Transacción Combinada Tipo I . . . . .	27
3.7.2. Perfil Esclavo S.7.3 . . . . .	29
3.8. Requerimientos del Receptor . . . . .	30
3.9. Manejo de Errores . . . . .	31

<b>4. Diseño de Hardware</b>	<b>33</b>
4.1. Sistema de desarrollo AdaptHC9S12E128 . . . . .	33
4.2. Diagrama de Bloques . . . . .	35
4.3. Transmisor AS-Interface . . . . .	36
4.3.1. Interfaz de Acople . . . . .	37
4.3.2. Driver de línea . . . . .	38
4.4. Receptor AS-Interface . . . . .	40
4.5. Circuitos Complementarios . . . . .	42
4.5.1. Circuito de Conmutación Rx/Tx . . . . .	42
4.5.2. Fuente de Alimentación . . . . .	43
<b>5. Implementación del Software</b>	<b>45</b>
5.1. Implementación de protocolos en microcontrolador . . . . .	45
5.2. Maestro AS-Interface . . . . .	46
5.3. Esclavo AS-Interface . . . . .	46
5.4. Implementación . . . . .	49
5.5. Generación de Patrones Analógicos . . . . .	50
5.5.1. Representación de 24 muestras . . . . .	52
5.5.2. Representación de 20 muestras . . . . .	53
5.5.3. Representación de 30 muestras . . . . .	53
5.5.4. Tiempo de respuesta del modulo DAC . . . . .	54
5.5.5. Evaluación de las aproximaciones . . . . .	56
5.6. Funciones Implementadas . . . . .	57
5.6.1. Reset Slave . . . . .	57
5.6.2. Read I/O Configuration . . . . .	58
5.6.3. Read ID Code . . . . .	58
5.6.4. Read ID2 Code . . . . .	58
5.6.5. Write Parameter . . . . .	59
5.6.6. Data Exchange . . . . .	60
<b>6. Pruebas y Resultados</b>	<b>61</b>
6.1. Pruebas sobre la arquitectura de red comercial . . . . .	61
6.1.1. Pruebas de corriente . . . . .	61
6.1.2. Pruebas de voltage . . . . .	61
6.2. Pruebas del Procesador AS-i . . . . .	63
6.2.1. Pruebas de Transmisión . . . . .	63

---

6.2.2. Pruebas de Recepción . . . . .	64
6.3. Pruebas Varias . . . . .	65
6.4. Aplicación de Validación . . . . .	68
<b>7. Conclusiones y Recomendaciones</b>	<b>71</b>
7.1. Conclusiones . . . . .	71
7.2. Recomendaciones . . . . .	73
7.3. Futuros trabajos en esta área . . . . .	74
<b>A. Análisis de la Fuente Howland</b>	<b>75</b>
A.1. Función de transferencia . . . . .	75
A.2. Análisis de Impedancias . . . . .	76
A.2.1. Impedancia de Entrada . . . . .	76
A.3. Impedancia de Salida . . . . .	76
<b>B. Análisis de la Fuente Howland Diferencial</b>	<b>79</b>
B.1. Función de transferencia . . . . .	79
B.2. Impedancia de Salida . . . . .	81
B.3. Corriente en Modo Común . . . . .	83
<b>C. Análisis de la línea de transmisión AS-Interface</b>	<b>85</b>
C.1. Modelo Circuitual . . . . .	85
<b>D. Circuitos de Acople</b>	<b>89</b>
D.1. Acople de Referencias . . . . .	89
D.1.1. Capacitor <i>Bypass</i> . . . . .	89
D.1.2. Desacople . . . . .	90
D.1.3. Ferritas <i>Bead</i> . . . . .	90
D.2. Aplicaciones PLC . . . . .	91
D.2.1. Coupling Capacitors . . . . .	91
D.2.2. Coupling Transformers . . . . .	91
D.2.3. Blocking Inductors . . . . .	91
D.2.4. Resistores . . . . .	92
<b>E. Pruebas sobre el <i>Driver</i> de línea</b>	<b>93</b>
E.1. Pruebas de Simulación . . . . .	93
E.2. Prueba Experimentales . . . . .	94

<b>F. Esquemático de la tarjeta diseñada para el Microcontrolador</b>	<b>99</b>
<b>G. Estudio de Impedancias</b>	<b>101</b>
G.1. Impedancia característica de la línea . . . . .	101
G.2. Impedancia equivalente del esclavo . . . . .	102
<b>Referencia Bibliográfica</b>	<b>107</b>

# Índice de Figuras

1.1. Gráfica comparativa de los buses de campo. . . . .	2
2.1. Modelo OSI extrapolado a buses de campo. . . . .	13
2.2. Diagrama de bloques de un sistema de comunicación digital. . . . .	14
2.3. Pirámide de automatización. . . . .	17
3.1. Interfases del protocolo AS-i. Lógica (Izquierda) y Física(Derecha). . . . .	21
3.2. Señal AS-i. . . . .	22
3.3. Proceso de encuestamiento cíclico. . . . .	23
3.4. Transacción AS-i. . . . .	24
3.5. Estructura del datagrama AS-i. . . . .	25
3.6. Requerimientos del Receptor AS-i. . . . .	31
4.1. Arquitectura del HCS12. . . . .	34
4.2. Tarjeta de aplicación diseñada. (a) Cara superior. (b) Cara posterior. . . . .	35
4.3. Diagrama de bloques del procesador AS-i. . . . .	36
4.4. Seguidor de emisor. . . . .	37
4.5. Circuito de acople discreto. . . . .	38
4.6. Circuito de acople definitivo del sistema digital. . . . .	38
4.7. Diagrama circuital del driver de línea. . . . .	39
4.8. Etapas de amplificación y comparación del circuito de recepción. . . . .	41
4.9. Etapas de aislamiento digital y nivel de comparacion del circuito de recepción. . . . .	41
4.10. Esquema de aplicación del multiplexor analógico usado como <i>switch</i> . . . . .	42
4.11. Diagrama circuital de la fuente de alimentación positiva. . . . .	43
4.12. Diagrama circuital de la fuente de alimentación negativa. . . . .	44
5.1. Máquina de estados del maestro AS-i. . . . .	47
5.2. Diagrama de estados del esclavo AS-i. . . . .	48
5.3. Diagrama de flujo de la aplicación maestra. . . . .	52

5.4.	Representación de los patrones generados con 24 muestras. . . . .	53
5.5.	Representación de los patrones generados con 20 muestras. . . . .	54
5.6.	Representación de los patrones generados con 30 muestras. . . . .	55
5.7.	Representación analógica final para un $T_{BIT}$ . . . . .	57
6.1.	Formas de onda de corriente. (a) Puerto ASi+. (b) Puerto ASi-. . . . .	62
6.2.	Forma de onda de voltage ASi. . . . .	62
6.3.	(a) Forma de onda de corriente generada. (b) Forma de onda de voltaje sobre el bus. . . . .	63
6.4.	Patrones ideales del arreglo binario 0110001b. . . . .	64
6.5.	(a) Pulsos detectados para una respuesta de esclavo antes de la etapa de aislamiento. (b) Pulsos detectados para una petición de maestro después de la etapa de aislamiento. . . . .	65
6.6.	Petición <i>Data_Exchange</i> del maestro al procesador AS-i configurado con la dirección dos. . . . .	66
6.7.	Patrones ideales para la petición <i>Data_Exchange</i> , al esclavo identificado con dirección dos. . . . .	67
6.8.	Transacción AS-i con el procesador configurado con la dirección cinco. . . . .	67
6.9.	Patrones ideales para una transacción AS-i al procesador configurado con la direc- ción cinco. (a) Petición de maestro. (b) Respuesta del esclavo. . . . .	68
6.10.	Arquitectura de validación propuesta. . . . .	69
A.1.	Fuente Howland sencilla. . . . .	75
A.2.	Cálculo de la impedancia de salida. . . . .	77
B.1.	Fuente Howland Diferencial. . . . .	79
B.2.	Cálculo de la impedancia de salida. . . . .	82
B.3.	Cálculo de la corriente en modo común. . . . .	83
C.1.	Modelo circuital de la línea AS-i. . . . .	86
C.2.	Equivalente circuital para evaluar el aporte de la fuente de AC. . . . .	86
C.3.	Equivalente circuital para evaluar el aporte de la fuente de DC. . . . .	86
E.1.	Resultados de simulación en Orcad. . . . .	93
E.2.	Curva de %Error para $R_{Feedback} = 1k\Omega$ . . . . .	95
E.3.	Curva de %Error para $R_{Feedback} = 5 \cdot 1k\Omega$ . . . . .	96
E.4.	Curva de %Error para $R_{Feedback} = 11 \cdot 7k\Omega$ . . . . .	96
F.1.	Esquemático de la tarjeta de aplicación para el microcontrolador. . . . .	100
G.1.	Modelo equivalente de la línea AS-i. . . . .	101

---

G.2. Resultados de la variación de la resistencia en el modelo de la línea AS-i. (a) Opción 1. (b) Opción 2. . . . .	102
G.3. Resultados de la variación de la capacitancia en el modelo de la línea AS-i. (a) Opción 3. (b) Opción 4. . . . .	103
G.4. Resultados de la variación de la inductancia en el modelo de la línea AS-i. (a) Opción 5. (b) Opción 6. . . . .	104
G.5. Resultados de la variación de la conductancia en el modelo de la línea AS-i. (a) Opción 7. (b) Opción 8. . . . .	104
G.6. Resultados de la variación de la longitud en el modelo de la línea AS-i. (a) Opción 9. (b) Opción 10. . . . .	105
G.7. Resultados de la variación del tiempo de propagación sobre la línea AS-i. . . . .	105



# Índice de Tablas

2.1. Capas del modelo OSI. . . . .	12
3.1. Descripción de los campos del datagrama AS-i. . . . .	25
3.2. Tipos de transacciones AS-i simples. . . . .	26
3.3. Tipos de transacciones AS-i combinadas. . . . .	26
3.4. Definición de los <i>bits</i> de datos I/O en transacciones tipo I. . . . .	27
3.5. Transferencia de datos del maestro al esclavo en transacciones tipo I. . . . .	28
3.6. Transferencia de datos del esclavo al maestro en transacciones tipo I. . . . .	28
3.7. Definición de los <i>bits</i> de parametros en transacciones tipo I. . . . .	29
3.8. Arreglo de datos para la transferencia de valores analógicos de 16-Bits. . . . .	30
3.9. Representación de datos de 12-Bits y 8-Bits. . . . .	30
5.1. Monitoreo del sistema AS-i. . . . .	51
5.2. Prueba de tiempos de respuesta del módulo DAC. . . . .	55
5.3. Resultados comparativos en el proceso de generación. . . . .	56
5.4. Estructura de una petición de <i>Reset</i> . . . . .	58
5.5. Estructura de la respuesta del esclavo. . . . .	58
5.6. Estructura de una petición de lectura de configuración I/O . . . . .	58
5.7. Opciones de configuración I/O para dispositivos esclavos . . . . .	59
5.8. Estructura de una petición de lectura del código de identificación . . . . .	59
5.9. Estructura de una petición de lectura del ID2-Code . . . . .	59
5.10. Estructura de una petición de escritura de parámetros . . . . .	60
5.11. Estructura de una petición de intercambio de datos . . . . .	60
C.1. Referencias del modelo circuital. . . . .	85
E.1. Resultados de simulación para $R_{Feedback} = 10k\Omega$ . . . . .	94
E.2. Resultados de simulación para $R_{Feedback} = 1k\Omega$ . . . . .	94
E.3. Resultados experimentales para $R_{Feedback} = 1k\Omega$ . . . . .	95

---

E.4. Resultados de experimentales para $R_{Feedback} = 5 \cdot 1k\Omega$ . . . . .	95
E.5. Resultados de experimentales para $R_{Feedback} = 11 \cdot 7k\Omega$ . . . . .	96
G.1. Parámetros de la línea AS-i . . . . .	102
G.2. Variación de la resistencia del modelo de la línea . . . . .	103
G.3. Variación de la capacitancia del modelo de la línea . . . . .	103
G.4. Variación de la inductancia del modelo de la línea . . . . .	103
G.5. Variación de la conductancia del modelo de la línea . . . . .	103
G.6. Variación de la longitud del modelo de la línea . . . . .	104
G.7. Variación del tiempo de propagación del modelo de la línea . . . . .	104

# Resumen

Titulo: MÓDULO DE INTEGRACIÓN DE SENSORES Y ACTUADORES LÓGICOS BASADO EN EL PROTOCOLO AS-I DE COMUNICACIONES INDUSTRIALES<sup>1</sup>

Autor : OMAR LEONARDO NÚÑEZ GUALDRÓN<sup>2</sup>

Palabras Claves : AS-Interface, Automatización Industrial, Bus de Campo, Driver de Línea, Microcontrolador, Power Line Communications, Pirámide de Automatización, Redes Industriales.

La comunicación de datos en entornos industriales facilita el conocimiento del estado operativo de un proceso en cualquier momento, razón por la cual se hace necesaria la interconexión de sensores, actuadores, y dispositivos de control y supervisión, a través de los diferentes niveles de la jerarquía de automatización. Debido a que la adopción de los buses de campo requiere de una inversión elevada, surge la necesidad de desarrollar procesadores de comunicación que soporten especificaciones adoptadas internacionalmente y garanticen la compatibilidad con soluciones comerciales. En este trabajo de maestría se presenta el diseño e implementación de un procesador de comunicación *AS-Interface* que actúa como dispositivo esclavo, para habilitar el intercambio de datos sobre líneas de DC e interfazar las variables de proceso con un controlador maestro. La implementación realizada en microcontrolador, resultó de la integración de componentes *Software* y *Hardware*, siguiendo los requerimientos lógicos y físicos del estándar IEC 62026-2, para establecer un proceso de comunicación maestro-esclavo por encuestamiento cíclico y acoplar patrones analógicos modulados en corriente a un bus DC. La aplicación de este desarrollo tiene por objeto proporcionar una solución de comunicación a las empresas del sector industrial con mediano y bajo nivel de automatización, para el manejo de sensores y actuadores.

---

<sup>1</sup> Trabajo de Investigación

<sup>2</sup> Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Magister en Ingeniería Electrónica. Jaime Guillermo Barrero Pérez

# Abstract

Title : INTEGRATION MODULE OF LOGICAL SENSORS AND ACTUATORS BASED ON AS-I PROTOCOL OF INDUSTRIAL COMMUNICATIONS<sup>3</sup>

Author : OMAR LEONARDO NÚÑEZ GUALDRÓN<sup>4</sup>

Keywords : AS-Interface, Industrial Automation, Fieldbus, Line Driver, Microcontroller, Power Line Communications, Automation Pyramid, Industrial Network

The Power Line Communications technology represents a powerful alternative of communication with a wide application in domotics networks and automation solutions, to facilitate the information interchange. At present, the industrial automation recover an important role in any manufacturing proces, where the principal components are sensors and actuators. This situation is produced due the actual necessity of the companies to increase their productivity and to facilitate the remote engineering development. It has promoted the application of news information technologies between differents levels of an automated proces. Data communication in industrial environments allows the knowledge of the process operative state at any time, reason for which becomes necessary the interconnection of sensors, actuators, control and supervision systems, through the different levels from the automation hierarchy. Just the adoption of the fieldbuses requires of an high investment, arises the necessity to develop communication processors that support specifications internationally adopted and guarantee the compatibility with commercial solutions. This document presents the design and implementation of an AS-Interface communication processor that acts like slave device, to enable the data interchange over DC power lines and to integrate the process variables with the master controller. The implementation made in microcontroller, resulted from the integration of Software and Hardware components, following the logical and physical requirements of standard IEC 62026-2, to establish a master-slave process communication by cyclic polling and to connect analogical patterns modulated in current. The application of this development intends to provide a communication solution to the companies with medium and low automation level, for the handling of logical and analogical, sensors and actuators.

---

<sup>3</sup> Trabajo de Investigación

<sup>4</sup> Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Magister en Ingeniería Electrónica. Jaime Guillermo Barrero Pérez

# Capítulo 1

## Introducción

El uso de líneas de continua para propósitos de comunicación de datos no es una idea nueva. La tecnología PLC<sup>1</sup> representa una poderosa alternativa de comunicación con amplia aplicación en redes domóticas y soluciones de automatización, para facilitar el intercambio de información sobre líneas de energía [1], [2]. En el mundo de hoy, la *Automatización* y la *Informática Industrial* desempeñan un papel importante en cualquier proceso de manufactura, donde los principales componentes son los sensores y los actuadores; situación que se genera debido a la necesidad actual de las empresas por elevar su productividad y facilitar el desarrollo de la ingeniería a distancia [3], [4]. De esta forma se ha promovido la aplicación de las nuevas tecnologías de información a nivel industrial, para satisfacer las actuales exigencias de comunicación entre los diferentes niveles de un proceso automatizado.

La comunicación en entornos industriales entre los dispositivos de campo y sus sistemas de control se ha efectuado tradicionalmente a través de la interfaz analógica de corriente de 4 – 20mA, la cual es una tecnología muy difundida por su capacidad para transmitir con precisión las mediciones de variables de proceso, pero posee inconvenientes en aplicaciones de control distribuido. Es así como los buses de campo se han convertido en la opción más adecuada de intercambio de información en entornos industriales, proporcionando características como: manejo de un mayor volumen de datos, mayor diversidad de datos (diagnóstico, información, configuración y proceso), simplicidad de conexión, mejoramiento de la confiabilidad y reducción de costos (puesta en marcha y mantenimiento).

Actualmente, existen en el mercado un gran número de alternativas totalmente diferenciadas, con características propias, tanto en el nivel físico como en el enlace de comunicación, entre las cuales aparecen protocolos de gran difusión como: *Ethernet*, *Foundation Fieldbus*, *Interbus*,

---

<sup>1</sup> *Power Line Communications*.

*Profibus*, *CAN*<sup>2</sup>, *Devicenet*, *Controlnet* y *AS-Interface*; los cuales se posicionan dentro de los diferentes niveles de la pirámide de automatización<sup>3</sup>. En la figura 1.1 se muestra una gráfica comparativa de los buses de campo, cualificando características como la complejidad de su estructura, manejo de volumen de datos, funcionalidad y costo.

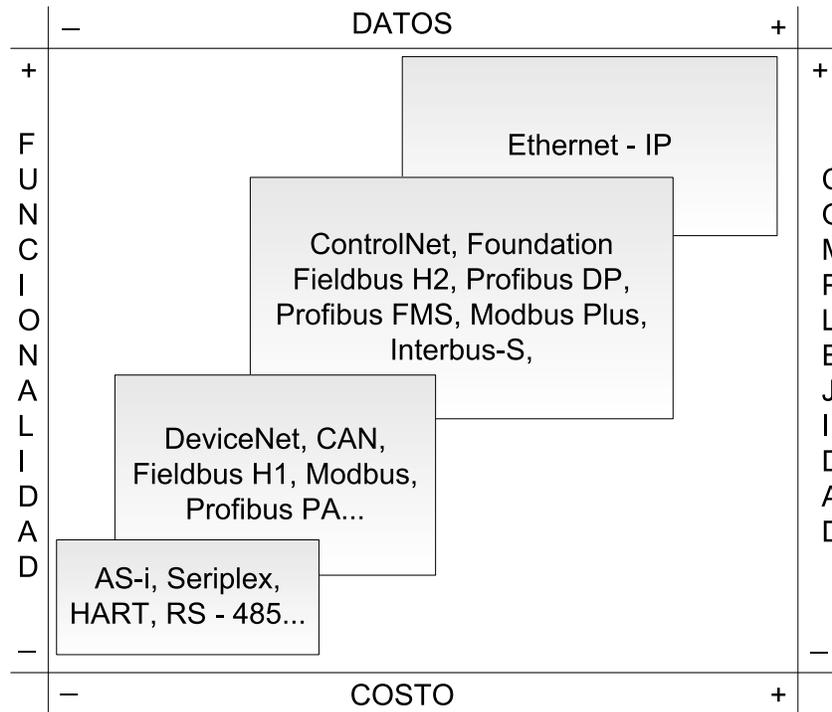


Figura 1.1: Gráfica comparativa de los buses de campo.

Actualmente, algunos fabricantes buscando asegurar el mercado han desarrollado protocolos de comunicación que no pueden ser considerados estándares, debido a que no permiten la compatibilidad entre diferentes arquitecturas. Por esta razón se demanda la adopción de sistemas abiertos, normalizados y regulados por organismos internacionales, para facilitar el desarrollo de dispositivos que aseguren una mayor interoperabilidad entre diferentes alternativas comerciales y una mayor rapidez de integración de nuevas tecnologías. La *Universidad Industrial de Santander* a través del *Grupo de Investigación en Control, Electrónica y Simulación*, y el *Grupo de Investigación en Mecatrónica*, está interesada en desarrollar trabajos de investigación en el tema de las comunicaciones industriales, orientados al diseño e implementación de procesadores de comunicación, el cual es un tópico considerado de actualidad científica en el área de la *Informática Industrial*.

<sup>2</sup> *Control Area Network*.

<sup>3</sup> Modelo de distribución jerárquica aceptada para la clasificación de los dispositivos industriales dentro de un proceso automatizado, en cuatro niveles de proceso: gestión, supervisión, control y sensor/actuador.

## 1.1. Planteamiento del problema

Debido a la gran importancia que tienen actualmente las comunicaciones digitales en los entornos industriales, factores como la falta de apropiación de las tecnologías existentes y la escasez de desarrollos en el manejo de los estándares comerciales de comunicación, reflejan la necesidad de generar avances en el campo de la *Informática Industrial*, orientados al diseño e implementación de procesadores de comunicación en *Hardware Embebido* y el desarrollo de módulos inteligentes en el primer nivel de proceso, de tal forma que se especifique la interconexión lógica y física de sensores y/o actuadores analógicos con un controlador a través de una estación maestra, para habilitar funciones de adaptabilidad (fácil implementación), conectividad (compatibilidad de arquitecturas), diagnóstico (mantenimiento y detección de fallos) e intercambio de datos sobre una línea de distribución de potencia en DC.

## 1.2. Justificación

El estudio y solución de problemas en el área de la *Automatización de Procesos* es mundialmente uno de los temas de actualidad científica con mayor proyección en el entorno industrial, debido a la creciente necesidad de implementar interfases de comunicación y desarrollar estrategias de optimización de procesos.

El desarrollo de este trabajo de investigación se justifica en el interés común existente entre las Escuelas de *Ingeniería Eléctrica, Electrónica y de Telecomunicaciones* e *Ingeniería Mecánica*, por fortalecer la línea de investigación en *Automatización e Informática Industrial*, mitigar la dependencia tecnológica y proponer soluciones de conectividad con base en estándares comerciales. La revisión del estado del arte permite mostrar que el desarrollo propuesto en este trabajo de investigación, correspondiente al estudio de estándares de comunicación industrial e implementación en *Hardware Embebido*, es una temática en creciente avance a nivel internacional. Así mismo, dicha revisión permite evidenciar la carencia de desarrollos en esta línea a nivel nacional y regional, enmarcando la necesidad de proponer soluciones alternativas de comunicación que cubran problemas específicos en el sector industrial, como la interconexión de dispositivos de campo analógicos y estaciones de control.

*AS-Interface* representa un estándar de comunicación en entornos industriales, que ofrece soluciones de automatización con facilidades de implementación y mantenimiento a bajo costo, capacidad de detección de fallos y manejo de funciones de diagnóstico y configuración, directamente sobre la interfaz sensor/actuador. Este estándar presenta un creciente auge a nivel internacional, pero poca difusión en el panorama industrial nacional, posee especificaciones abiertas<sup>4</sup> y cuenta con el

---

<sup>4</sup> Estándar que permite la inclusión de especificaciones mantenidas y actualizadas por organizaciones interna-

respaldo de la *AS-Interface International Association*<sup>5</sup>, lo cual refleja su gran solidez y proyección de permanencia en el mercado, al contar con la vinculación de fabricantes reconocidos como *ABB*, *Siemens*, *IFM*, *Schneider* y *Allan Bradley*.

### 1.3. Objetivos del trabajo de investigación

El objetivo general de este trabajo de investigación fue el diseño e implementación en microcontrolador de un procesador de comunicaciones que permita la integración de sensores analógicos, bajo los requerimientos del protocolo *AS-Interface*, según el estándar *EN50295*.

Este procesador esta basado en la integración de componentes *Hardware* y *Software*, que garanticen los requerimientos físicos y lógicos del estandar, implementado como dispositivo esclavo para la aplicación de soluciones de comunicación industrial en el nivel sensor/actuador de la jerarquía de automatización, a través de líneas de DC.

Para el cumplimiento del objetivo general expuesto anteriormente fueron trazados los siguientes objetivos específicos:

- Diseñar una interfaz *Hardware* que garantice la conexión física del dispositivo esclavo a la red *AS-Interface*, para efectuar la transmisión y recepción de datos sobre el bus de alimentación de los componentes de red.
- Implementar en un microcontrolador las funciones lógicas requeridas para el intercambio de datos según el perfil *S – 7 . 3* de dispositivos esclavos, el cual habilita el manejo de sensores analógicos mediante el uso de transacciones combinadas tipo I.
- Validar el funcionamiento del procesador de comunicaciones mediante el desarrollo de una aplicación *Software* en un PC que implemente las funciones del perfil *M . 3* de dispositivos maestros, para habilitar la comunicación con el dispositivo esclavo propuesto.

### 1.4. Alcances del proyecto de investigación

Con el desarrollo de este trabajo de investigación se estudió el protocolo *AS-Interface* utilizado en el nivel sensor/actuador de un proceso automatizado, para llevar a cabo el diseño e implementación de un procesador de comunicaciones en un microcontrolador, cuya función principal es

---

cionales como la *IEC* y la *CENELEC*, permitiendo la incorporación de nuevas tecnologías y requerimientos.

<sup>5</sup> [www.as-interface.com](http://www.as-interface.com) o [www.as-interface.net](http://www.as-interface.net).

la integración de dispositivos de instrumentación convencionales (sensor analógico no inteligente) a un bus de campo habilitado para la transmisión de datos y potencia por el mismo cable. El módulo cumple con los requerimientos del perfil  $S - 7.3$  de dispositivos esclavos expuesto en el estándar  $EN50295^6$  para garantizar la interconexión de un sensor analógico con salida de voltaje a una estación maestra soportada por el perfil  $M.3$  de dispositivos maestros, mediante el uso de transacciones combinadas tipo I. [5] [6].

Los resultados de esta investigación se extendieron a los grupos comprometidos con el desarrollo de la línea de investigación en *Informática Industrial*, con el objeto de proponer dispositivos de medida y control con funciones específicas de comunicación, diagnóstico e integración a arquitecturas de red comerciales. En segunda instancia se realizó una contribución en el estado del arte y en la estructuración temática de una materia electiva orientada al estudio, caracterización, diseño e implementación de buses de campo en *Hardware Embebido*, con su respectivo proceso de validación y aplicación como propuesta dentro de la reforma curricular que se lleva a cabo en la *Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones*.

## 1.5. Estructura del documento

Este documento se encuentra organizado en 7 capítulos y 6 apéndices. El capítulo 2 presenta los conceptos básicos relacionados con el conocimiento de los sistemas de comunicación digital y su aplicación en ambientes industriales, enmarca los buses de campo en los niveles de un sistema automatizado dentro del contexto del trabajo de investigación y revisa el estado del arte actual del tema.

El capítulo 3 presenta los conceptos que definen el comportamiento físico y lógico del protocolo *AS-Interface* de acuerdo a los requerimientos exigidos por el estándar  $IEC62026 - 2$ , con el objeto de establecer los requerimientos del procesador de comunicación.

El capítulo 4 presenta la descripción de los esquemas diseñados e implementados a nivel *hardware*, para cumplir con los requerimientos físicos del protocolo *AS-Interface*. Se destacan los avances en las interfases de transmisión (*driver* de línea) y recepción (detección de pulsos).

El capítulo 5 presenta la descripción de los algoritmos implementados a nivel *software* en el microcontrolador, para cumplir con los requerimientos lógicos del protocolo *AS-Interface*. Se destacan los avances en la implementación completa de la máquina de estados del dispositivo

---

<sup>6</sup> *Low-voltage switchgear and controlgear - Controller device interface systems - Actuator Sensor Interface (AS-I)*

esclavo, el desarrollo parcial del dispositivo maestro, el proceso de generación de los patrones analógicos de corriente y la decodificación de los pulsos detectados de la red.

El capítulo 6 presenta los resultados y análisis obtenidos al evaluar el desempeño de procesador de comunicaciones *AS-Interface* y validar su funcionamiento para el manejo de sensores analógicos.

Finalmente, en el capítulo 7 se exponen las conclusiones y aportes realizados con este trabajo, y se sugieren líneas específicas de estudio para el desarrollo de futuras investigaciones.

# Capítulo 2

## Marco Contextual

Este capítulo presenta los conceptos básicos relacionados con los sistemas de comunicación digital y su aplicación en ambientes industriales, enmarca los buses de campo en los niveles de un sistema automatizado y revisa el estado del arte del tema, con el objetivo de contextualizar el trabajo de investigación

La evolución de la gran mayoría de aplicaciones de comunicación, permite ilustrar una dependencia de nuestra sociedad del flujo de información oportuno y exacto, lo cual implica un gran número de consideraciones relacionadas con el diseño, implementación y operación de sistemas de transmisión de datos. Durante un período aproximado de un siglo, el telégrafo ha sido reemplazado por una amplia variedad de redes que han marcado el desarrollo de la sociedad moderna, por ejemplo, cuando realizamos nuestras operaciones diarias, es difícil evitar encontrarse una aplicación que no sea dependiente de un teleproceso, un acceso remoto o un intercambio de información, y aunque podamos afrontar una aplicación en el entorno industrial sin la capacidad de comunicación de datos, sería imposible proporcionar un servicio confiable y eficiente.

### 2.1. Revisión del estado del arte

En la actualidad, la investigación en el campo de la *Informática Industrial* ha tenido gran auge en el entorno académico, orientando su desarrollo hacia tres grandes áreas de trabajo:

- Implementación de protocolos de comunicación en *Hardware Embebido*.
- Desarrollo y adaptación de nuevas interfaces de comunicación basadas en protocolos comerciales.
- Diseño de nuevos sistemas y protocolos de comunicación con propósitos específicos.

Hasta el momento en la *Universidad Industrial de Santander* se había abordado esta temática implementando alternativas comerciales proporcionadas por fabricantes como *Siemens*, *Schneider* y *Allan Bradley*, resaltando el manejo de sus características técnicas, configuración y programación; sin embargo es poco el trabajo desarrollado respecto al estudio e implementación de protocolos e interfases de comunicación. Específicamente en la *Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones*, se han realizado aportes a través de proyectos y monografías, donde recientemente se pueden destacar importantes avances en el conocimiento del protocolo Modbus. A continuación se relacionan estos trabajos:

- Redes industriales. Modbus y Ethernet implementados en autómatas programables TRILOGI, KOYO Y MODICON-TELEMECANIQUE. Niño B. Carlos E. y Álvaro B. Becerra A. Tesis de Pregrado, 2002.
- Redes de Comunicaciones Industriales. Gélvez F. Julio A. Monografía Especialización en Telecomunicaciones, 2002.
- Redes de Comunicación AS-i. Funcionamiento, Ventajas y Estado del Arte. López C. Duván A. Monografía Especialización en Telecomunicaciones, 2004.
- Modbus. Monitoreo de la Red empleando LabView. Carreño Sinle M. y Albarracin Pedro. Tesis de Pregrado, 2005.
- Modbus. Implementación de Procesador de Comunicaciones. Duque P. Jorge E. Tesis de Maestría, 2005.
- Modbus RTU. Implementación del Protocolo en Microcontrolador. Torres Roger. Tesis de Pregrado, 2006.
- Integración de Modbus aun sistema SCADA. Fajardo Diana. Tesis de Pregrado, 2006.

De igual forma en la Escuela de Ingeniería Mecánica se han desarrollado proyectos de adaptación tecnológica orientados a la implementación y configuración de redes comerciales como son:

- Redes industriales de comunicación basadas en Profibus. Puerto Jhon C. y Gómez José D. Tesis de Pregrado, 1999.
- Redes industriales de comunicación basadas en Profibus y AS-Interface. Díaz G. José N. y Martínez O. Edisón. Tesis de Pregrado, 2002.
- BACnet. Protocolo de Comunicación. Archila D. Jhon F. y Barragan R. Humberto. Monografía de Especialización, 2003.

A nivel nacional, es en el grupo de investigación en *Percepción y Sistemas Inteligentes* de la *Universidad del Valle*, donde se han desarrollado un gran número de trabajos relacionados con las redes de comunicación industrial, entre los cuales se destacan:

- Plataforma de Experimentación en Informática Industrial. Proyecto cuyo objetivo era desarrollar una plataforma de experimentación en el ámbito de la informática industrial para la investigación y desarrollo de aplicaciones con características de portabilidad, fácil replicabilidad y accesabilidad remota.
- Laboratorio Virtual de Informática Industrial. Proyecto cuyo objetivo era dotar a la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle de un Laboratorio Virtual de Informática Industrial, con capacidad de monitorear y/o controlar una planta piloto que permita la experimentación remota sobre procesos reales y la asimilación de conceptos prácticos difíciles de explicar en forma verbal o escrita.
- Implementación de un esclavo con el protocolo *DeviceNet*. Arévalo B. William A. y Jaramillo T. Freddy E., 2003.
- Estudio e Implementación del Protocolo *HART* de Comunicaciones Industriales. Castillo Javier F. y Tenorio Jorge A., 2002.
- Adaptación de un canal *Modbus-MDLC* para la operación remota de un sistema de reconexión desde un punto de mando centralizado. Buitrago A. Jose A., 2001.
- Implementación de una red *Modbus/TCP*. Ruiz O. Andrés F., 2001. Desarrollo de una red industrial con conectividad TCP/IP a través del protocolo Modbus/TCP usando el sistema embebido TINI de Dallas Semiconductor.
- Implementación de una red Inalámbrica usando el protocolo *Modbus*. Escobar Angélica M., Sánchez Hugo A., 2000.
- Interface *CAN-TII*, según el estándar *IEEE P 1451*. Agudelo Meyer A. y Fula O. Marco A., 1999.
- Driver de bus de campo *CAN* para una red de transductores inteligentes. Navarro S. Erika, 1999.
- Diseño e implementación de un sensor inteligente de presión bajo el estándar *IEEE P 1451*. Berreneche B. Liliana y Rojas E. Engelbert, 1998.

- Diseño de un procesador de aplicación con capacidad para manejo de una red *HART* basado en el estándar *IEEE P 1451*. Medina E. Carlos A. y Palacios R. Víctor J., 1998.
- Desarrollo y Montaje de una Plataforma de Sistemas Distribuidos usando Redes *FIP* (Protocolo de Instrumentación de Procesos). Garzón, Motta, 1997.

Finalmente, a nivel internacional se han encontrado trabajos enfocados hacia el desarrollo e implementación en *Hardware Embebido* de protocolos como Modbus, Profibus y CAN, y la aplicación de la tecnología PLC a arquitecturas de comunicación. A continuación se relacionan los siguientes aportes:

- Power Line Communications in Automotive Industry. Trnka M. Doctoral Thesis. Czech Technical University, Praga, 2005. Trabajo que aplica la tecnología PLC al bus CAN.
- Entorno multidisciplinario para el desarrollo de sistemas de control distribuido con requisitos de tiempo real. Portillo B. Javier. Tesis Doctoral. Universidad del País Vasco - Escuela Superior de Ingenieros de Bilbao, España, 2004.
- Applicability of Power-Line Communications to Data Transfer of On-line Condition Monitoring of Electrical Drives. Ahola J. Doctoral Thesis. Lappeenranta University of Technology, Finlandia, 2003.
- Red industrial de adquisición de datos y control para la gestión empresarial. Universidad Nacional de Cuyo, Argentina, 2003. Investigación acerca de las técnicas de comunicación de *Hardware* y *Software* en el desarrollo de un sistema integrado de producción para el estudio, análisis, simulación y verificación de sistemas de control de producción en tiempo real, realizando la supervisión de redes AS-i y Modbus a través de un software SCADA.
- ASI Master Bus Controller. Poh S. Serena. Engineering Thesis. University of Queensland, Australia, 2003. Trabajo que llevo a cabo el diseño parcial de un controlador maestro para el bus AS-i utilizando una FPGA Xilinx Spartan II.
- DC Distributed Power Systems Analysis, Design and Control for a Renewable Energy System. Karlsson P. Doctoral Thesis. Lund University, Suecia, 2002.
- Real-Time Communications over Hybrid Wired/Wireless PROFIBUS-Based Networks. Andrade F. A. Mário J. Doctoral Thesis. Universidade do Porto, Portugal, 2002.
- Protótipo de rede industrial utilizando o padrão serial RS-485 e protocolo Modbus. Cuhna J. Universidad Regional de Blumenau, Brasil, 2002. En este trabajo se utilizaron los puertos

RS232 y RS485 como medio de transmisión serial y el protocolo Modbus para el intercambio de mensajes entre dos controladores de peso y un computador.

- Development of Modbus communication protocol for the BX-24 microcontroller. Cady F. and Stringam B. University of Montana, USA, 2001. Trabajo de investigación que consistió en el desarrollo de las rutinas de comunicación Modbus para el microcontrolador BX-24 y la interfaz Modbus para el programa de monitoreo en un PC.
- Implementación de *HART*. Febre U. Erika. Tesis de Ingeniería. Universidad simón Bolívar, Venezuela, 2001.
- Power Electronics System Communications. Milosavljevic I. Master Thesis. Virginia Polytechnic Institute and State University, USA, 1999.
- Power-Line Communications. Channel Properties and Communication Strategies. Selander Lars. Doctoral Thesis. Lund University, Suecia, 1999.

## 2.2. Modelo de referencia OSI

Ante un intercambio digital de datos entre equipos, es de vital importancia definir el sistema de transmisión, el método de acceso y las informaciones relativas al establecimiento de los enlaces. Por tal motivo, en 1983 la ISO (*International Standards Organization*) reglamentó el modelo de referencia OSI (*Open System Interconnection*), el cual define el comportamiento de un sistema de comunicación a través de la descripción de siete capas independientes (ver tabla 2.1), con una separación estructurada que permite la modificación interna de cualquiera de estas, sin afectar el funcionamiento de las demás. Cada capa se fundamenta en la anterior, para extraer la información de control necesaria para su funcionamiento y enviar los datos a la siguiente, cumpliendo una serie de convenciones predefinidas que en conjunto conforman el protocolo [7], [8].

Este modelo es válido tanto para grandes flujos de información, como para aplicaciones más sencillas. Su implementación estricta depende del grado de complejidad de la aplicación y no se considera necesario para construir un sistema de comunicación [9]. Por tal motivo en el campo de las comunicaciones industriales, la arquitectura OSI se simplifica al uso específico de tres capas [4], [10], como se muestra en la figura 2.1:

- Nivel 1: Capa Física. Este nivel procura la transmisión transparente de *bits* a través del soporte físico, en el orden definido por el nivel de enlace. Define las características eléctricas y mecánicas (conectores o medios de enlace tipo *hardware*) de la línea de transmisión, las señales de control que determinan la temporización y el orden de transmisión, y realiza

Tabla 2.1: Capas del modelo OSI.

Capa	Identificación	Funciones
7	Aplicación	Especifica las funciones de usuario para el intercambio de variables, a través de servicios de comunicación específicos.
6	Presentación	Define la representación de los datos, la conversión del tipo de representación en un formato adecuado para el equipo y ejecuta acciones de diagnóstico.
5	Sesión	Realiza tareas de sincronización. Administra el establecimiento, disolución y vigilancia de una sesión.
4	Transporte	Realiza el establecimiento y/o disolución del enlace, y la formación, repetición y clasificación de paquetes. Revisa la integridad de los datos.
3	Red	Define el direccionamiento de paquetes y las rutas de comunicación. Realiza procesos de control de flujo.
2	Enlace de Datos	Define el método de acceso al medio, realiza la gestión de colisiones, limita los bloques de datos, y ejecuta algoritmos de detección y corrección de errores.
1	Física	Describe los niveles de voltaje y requerimientos de tiempo para la transmisión de <i>bits</i> . Realiza <i>Test</i> de errores a nivel de <i>bit</i> .

un diagnóstico de errores a nivel de *bit*. Entre los estándares más usados en este nivel, se encuentran las interfases RS-232 y RS-485.

- Nivel 2: Capa de Enlace de Datos. Este nivel tiene como función asegurar la transmisión de la cadena de *bits* entre los dos sistemas involucrados en el proceso de comunicación, apoyándose en un medio físico de conexión. Este nivel es el encargado de formar las tramas de envío añadiendo datos de control y ejecutar los métodos de direccionamiento, detección y recuperación de errores, reenvío de tramas perdidas y regulación de tráfico. En redes locales procura el acceso exclusivo al medio definiendo los subniveles MAC (*Medium Access Control*) y LLC (*Logic Link Control*). Este último aplica generalmente a la norma IEEE 802.2, aunque en sistemas de bus de campo se utilizan métodos de acceso considerablemente modificados debido a las características de tiempo real exigidas para dicha aplicación.
- Nivel 7: Capa de Aplicación. Este nivel se encarga de proporcionar un entorno que facilite el

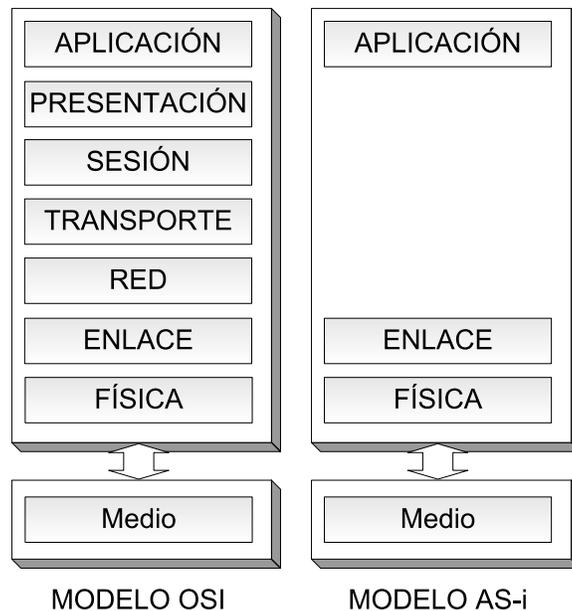


Figura 2.1: Modelo OSI extrapolado a buses de campo.

entendimiento entre usuarios de distintas máquinas digitales a nivel temático, sin importar medios, ni protocolos de comunicación. Comprende los servicios específicos de enlace con las diferentes aplicaciones de comunicación.

## 2.3. Sistemas de Comunicación Digital

La transmisión digital de información introduce suficientes ventajas como para dominar el creciente desarrollo de los sistemas de comunicación, de tal forma que se está convirtiendo en la única alternativa práctica de transporte de información. Aunque estas tan sólo representan una pequeña fracción dentro de este gran contexto, una fracción mayor está orientada a la transmisión de señales analógicas como voz e imagen. Hasta hace poco la transmisión digital era considerada ineficiente en términos de ancho de banda y costo de la conversión AD<sup>1</sup> y DA<sup>2</sup>; sin embargo aquí se enuncian argumentos que han transformado dicha posición [11]:

- La codificación de señales analógicas en forma digital se ha beneficiado de los avances en los algoritmos de compresión, los cuales reducen considerablemente la velocidad de transmisión requerida para representar una señal con alta calidad.
- Las técnicas de procesamiento de señal y codificación han incrementado la velocidad de transmisión que es posible alcanzar con un canal físico determinado.

<sup>1</sup> Análogo a Digital.

<sup>2</sup> Digital a Análogo.

- La evolución de la microelectrónica ha permitido el desarrollo de circuitos integrados que generan una reducción considerable de tiempos y costos en la ejecución de funciones complejas de procesamiento de señal (codificación, compresión y modulación).
- La aparición de nuevos medios de transmisión han permitido intercambiar información a altas velocidades sobre largas distancias.

Estos desarrollos han permitido cambiar la forma de concebir los sistemas de comunicación. Hoy en día la mayor problemática se centra en características como el ancho de banda y la capacidad total del sistema, las cuales son fácilmente alcanzables con la transmisión digital.

El propósito de un sistema de comunicación es transmitir los mensajes (secuencias de símbolos) provenientes de la fuente a un punto de destino con la mayor velocidad y precisión posible. La fuente y el destino están separados físicamente en el espacio y un canal de comunicaciones los conecta de alguna forma, estos canales aceptan señales eléctricas o electromagnéticas y su salida es usualmente una versión distorsionada de la entrada, debido a la naturaleza no ideal del medio físico que lo constituye. Además de la dispersión, la señal que transporta información también está afectada por señales eléctricas impredecibles (ruido) debido a causas naturales o artificiales, las cuales introducen errores y limitan la velocidad a la cual puede transportarse. A continuación se presenta la descripción funcional de cada uno de los bloques de un sistema de comunicación digital como el mostrado en la figura 2.2 [11], [7]:

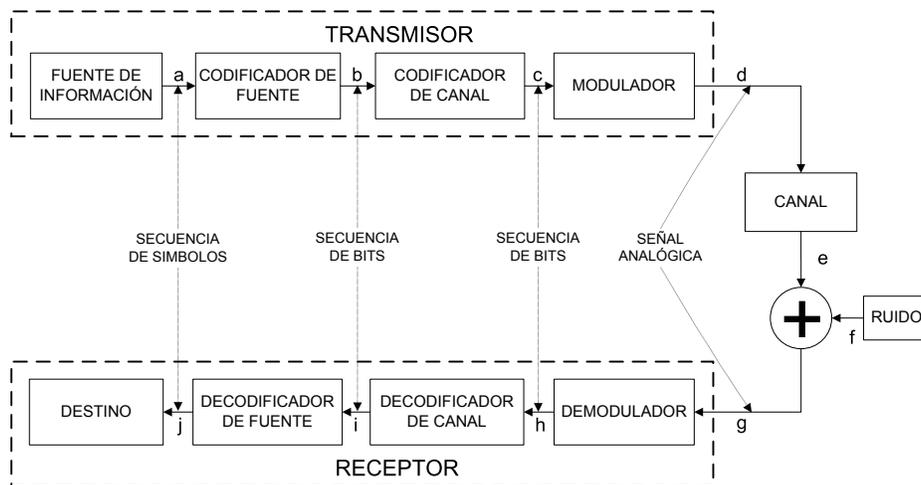


Figura 2.2: Diagrama de bloques de un sistema de comunicación digital.

1. *Fuente de Información*: Las fuentes de información según la naturaleza de su salida se clasifican en: analógicas o discretas. La salida de una fuente de información discreta consiste

en una secuencia de símbolos discretos (caracteres) y las fuentes de información analógica emiten una o más señales de amplitud continua en el tiempo, las cuales pueden convertirse en una fuente discreta a través de un adecuado proceso de muestreo y cuantización. Las fuentes de información se caracterizan mediante parámetros básicos como: el alfabeto de la fuente (símbolos), la velocidad de transmisión, la probabilidad de ocurrencia de cada símbolo y la dependencia probabilística de los símbolos. A partir de estas características es posible construir un modelo probabilístico de la fuente que permita estimar su Entropía ( $H$  en [*Bits/Símbolo*]) y su velocidad de transmisión ( $R$  en [*Bits/Seg*]). La Entropía se define como la información promedio contenida por símbolo en una secuencia de mensaje<sup>3</sup> y la velocidad de información representa el número mínimo de *Bits/Seg* necesarios en promedio para representar la información proveniente de la fuente.

2. *Codificación de Fuente*: El codificador de fuente convierte la secuencia de símbolos de entrada en una secuencia binaria, asignando palabras de código. Existen dos formas de realizar este proceso, la *Codificación de Longitud Fija* donde se asigna una palabra binaria de longitud fija a cada símbolo individual de la fuente y la *Codificación de Longitud Variable* donde se toma una subsecuencia de dos o más símbolos en un bloque, y se asigna palabras de código de longitud variable. Los parámetros más importantes de un codificador de fuente son el tamaño del bloque, la longitud de las palabras de código, la velocidad promedio de salida y la eficiencia del decodificador (definida como la relación entre la velocidad de salida y la velocidad mínima,  $R$ ). En el receptor, el decodificador debe realizar el proceso inverso convirtiendo la salida binaria en una secuencia de símbolos, y superar obstáculos como los crecientes requerimientos de memoria y la pérdida de sincronismo debido a errores de transmisión.
3. *Codificación de Canal*: Consiste en una metodología práctica para obtener alta confiabilidad y eficiencia en la transmisión. La codificación de canal con el propósito de control de errores, se logra adicionando *bits* extras a la salida del codificador de fuente, los cuales no llevan información, pero permiten la detección y corrección de errores asociados a los *bits* de información. Existen dos métodos para llevar a cabo este proceso, la *Codificación en Bloques*, donde se adicionan  $r$  *bits* de control a un bloque de  $k$  *bits* de información proporcionado por el codificador de fuente. El segundo método es la *Codificación Convolutiva*, donde la secuencia del mensaje se codifica en forma continua, entrelazando *bits* de información y de control. Ambos métodos requieren capacidad de almacenamiento y procesamiento de datos,

---

<sup>3</sup> Propiedad directamente relacionada con el número de símbolos del sistema a través de la expresión,  $H = \log_2[\#\text{Símbolos}]$

tanto en el trasmisor como en el receptor. Los parámetros más importantes de este codificador son el método de codificación, la eficiencia (relación entre la velocidad de salida del codificador de fuente y la capacidad del canal), la capacidad de control de errores y la complejidad. En el receptor, el decodificador de canal recupera los *bits* que transporta a partir de la secuencia binaria codificada, y realiza la detección y corrección de errores.

4. *Modulación*: La modulación es una de las más poderosas herramientas de diseño, que se utiliza para minimizar los efectos del ruido en el canal, adecuar el espectro de frecuencias de la señal transmitida a las características del canal, proveer la capacidad de multiplexado de varias señales y superar las limitaciones del enlace. El modulador acepta como entrada un flujo de *bits* y lo convierte en una señal eléctrica adecuada para la transmisión sobre el canal. Sus parámetros más importantes son el tipo de forma de onda utilizado, la duración de la misma, el nivel de potencia y el ancho de banda. La modulación debe ser un proceso reversible, razón por la cual la extracción del mensaje se consigue con técnicas de demodulación.
5. *Canal*: Provee la conexión eléctrica entre la fuente y el destino, a través de alternativas como el par trenzado, cable coaxial, el espacio libre o la fibra óptica. Debido a sus limitaciones físicas posee un ancho de banda ( $B$  en [*Hertz*]) finito, genera distorsión, produce atenuación y permite el efecto de señales eléctricas impredecibles e indeseadas, sintetizadas como ruido. El medio puede ser caracterizado con factores como la relación señal a ruido ( $SNR$  en [*dB*]), el ancho de banda utilizable ( $B$  en [*Hz*]), la respuesta de amplitud y fase, y las características estadísticas del ruido. Si estos parámetros son conocidos, es posible calcular la capacidad del canal ( $C$  en [*Bits/Seg*]), definida como la máxima velocidad a la cual es teóricamente posible transmitir sin errores.

## 2.4. Comunicaciones Industriales

El montaje de un sistema de automatización es difícil de concebir a simple vista, debido a la gran cantidad de dispositivos que trabajan conectados entre sí, utilizando diferentes arquitecturas y protocolos. Actualmente en los entornos industriales se encuentra una distribución jerárquica de la interconexión de dispositivos a través de los diferentes niveles de control, divididos según una determinada escala funcional. Dicha distribución mostrada en figura 2.3, la cual se conoce con el nombre de *Pirámide de Automatización*, define una serie de niveles basados en características como la velocidad de transferencia, el grado de protección, el tipo y el uso de los datos transmitidos. Este modelo no es una estructura rígida y esta sujeta a modificaciones, que en la mayoría de casos dependen del tipo de aplicación que se desee implementar, del proceso a automatizar, del punto

de vista de fabricantes y de la subjetividad de los conocedores del tema.

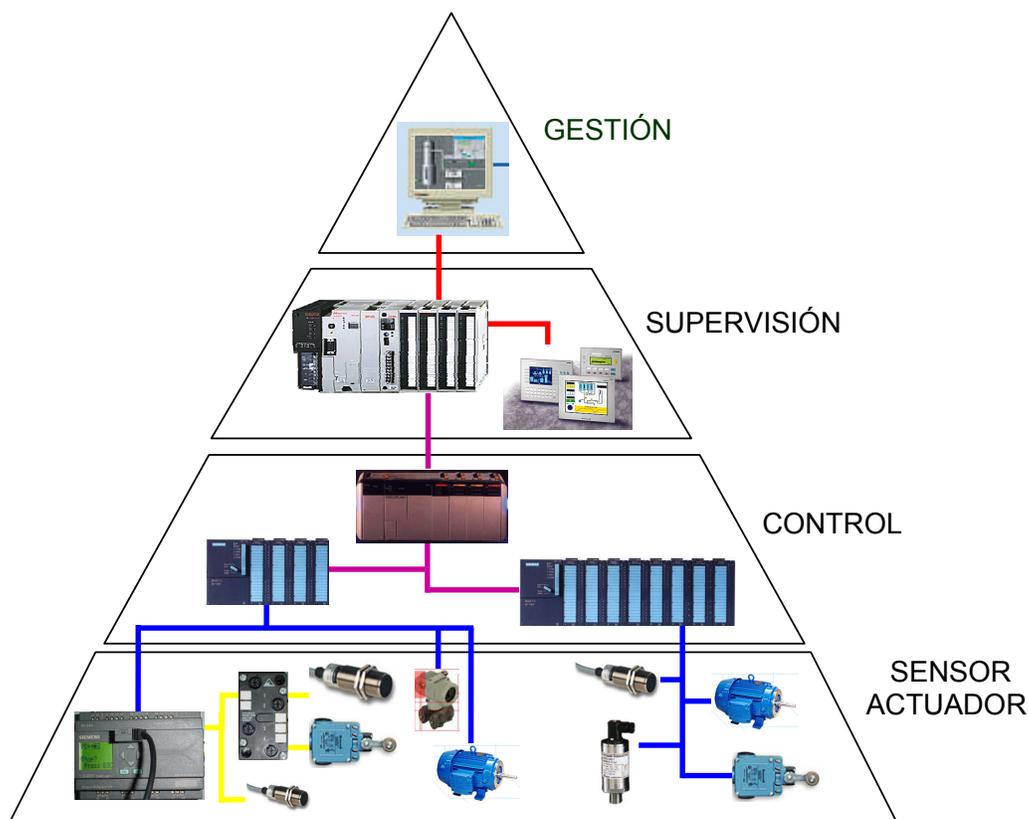


Figura 2.3: Pirámide de automatización.

Para comprender mejor la función de la red *AS-Interface* y su posición dentro de esta organización jerárquica, a continuación se definen las propiedades fundamentales de cada uno de estos niveles.

- Nivel de Gestión: En este nivel se conectan los *software SCADA*, *PC's* y las estaciones de trabajo para monitorear, gestionar y establecer la conexión con el proceso productivo a fin de obtener información de carácter administrativo y comercial (ventas e inventarios). Este entorno se caracteriza por manejar un alto volumen de datos (*MBytes*) y alta velocidad de transmisión (*Gbps*). Las redes empleadas son de tipo LAN (*Local Area Network*) y WAN (*Wide Area Network*) bajo protocolo *Ethernet*.
- Nivel de Supervisión: Enlaza las células de fabricación o zonas de trabajo. En este nivel se sitúan los PLC (*Programmable Logic Controller*) de gama alta, los ordenadores dedicados al diseño y control de calidad, y los terminales de programación, interconectados a través de redes tipo LAN (*MAP<sup>4</sup>* o *Ethernet*)

<sup>4</sup> *Manufacturing Automation Protocol*.. Es una red de gestión especialmente diseñada para el entorno industrial

- Nivel de Control: Integra pequeños automatismos tales como: PLC, RTU (*Remote Terminal Unit*) y controladores de proceso, en subredes o islas. En el nivel más alto de estas redes se encuentran uno o varios autómatas modulares actuando como maestros de la red o maestros flotantes. En este nivel se emplean buses de campo como *Modbus* y *Profibus*.
- Nivel de Sensor/Actuador: Nivel más bajo en la automatización de procesos. En este se conectan sensores y actuadores binarios y/o analógicos, los cuales capturan o actúan directamente sobre las variables de proceso. Se caracteriza por el manejo de un volumen de datos no muy elevado y tasas de transmisión grandes. Allí se pueden encontrar dispositivos como sensores inductivos, sensores capacitivos, contactos, arrancadores de motor, válvulas magnéticas y válvulas neumáticas. Este es el campo de aplicación del protocolo AS-i<sup>5</sup>, presentándose como la solución más adecuada en el primer nivel de proceso; donde también se encuentran protocolos como *DeviceNet* y *Seriplex*

### 2.4.1. Bus de Campo

Típicamente un bus de campo es un protocolo de comunicación digital, de alta velocidad, bidireccional, multipunto, montado sobre un bus serie que actúa como sistema de transmisión e intercambio de información entre dispositivos de campo y dispositivos de control. El objetivo de un bus de campo es sustituir las conexiones punto a punto entre dispositivos de red a través del tradicional bucle analógico de  $4 - 20mA$ , dentro de un DCS (*Distributed Control System*). La principal ventaja que ofrecen los buses de campo es la reducción de costos de instalación, mantenimiento y operación; asegurando la interoperabilidad y compatibilidad entre instrumentos de diferentes fabricantes.

---

y ampliamente empleada en LANs industriales. Fue creada por *General Motors* y está normalizada por la IEEE. No es una red que actúe al nivel de bus de campo, pero existen pasarelas para la adaptación de protocolo necesaria para su conexión

<sup>5</sup> Acrónimo de *Actuator Sensor Interface*.

# Capítulo 3

## Protocolo de Comunicación AS-Interface

Este capítulo presenta los fundamentos relacionados con el conocimiento físico y lógico del protocolo *AS-Interface* de acuerdo a los requerimientos exigidos por el estándar *EN50295*, con el objeto de definir las principales características técnicas del proceso de comunicación.

*AS-Interface* representa un estándar de comunicación en entornos industriales, que surgió hace aproximadamente 15 años para ofrecer soluciones de automatización con facilidades de implementación y mantenimiento a bajo costo, capacidad de detección de fallos, y manejo de funciones de diagnóstico y configuración, directamente sobre la interfaz sensor/actuador. Este estandar presenta un creciente auge a nivel internacional, reflejando una gran solidez y proyección de permanencia en el mercado.

### 3.1. Generalidades

El protocolo AS-i es un bus de campo desarrollado inicialmente por *Siemens*, para la interconexión de actuadores y sensores binarios, pero la arquitectura se extendió al manejo de entradas/salidas analógicas. A nivel físico, la red puede adoptar cualquier tipo de topología, estructuras en bus, árbol o estrella. Las especificaciones del protocolo AS-i están abiertas a dominio público bajo la regulación de la IEC<sup>1</sup> en el estándar 62026-2 , con la designación *Low-voltage switchgear and controlgear - Controller-device interfaces (CDIs) - Part 2: Actuator Sensor Interface (AS-i)* y actualmente es soportada por la organización independiente *AS-International Association*<sup>2</sup>.

---

<sup>1</sup> *International Electrotechnical Commission.*

<sup>2</sup> [www-as-interface.net](http://www-as-interface.net).

Entre las principales características de este sistema se encuentran:

- Velocidad de transmisión fija de  $166.67\text{kbps}$ .
- Como medio físico de transmisión, emplea un cable bifilar sin apantallamiento, que permite tanto la alimentación del bus, como el transporte de datos. Presenta un sistema de conexión tipo vampiro (autoregeneradora) que evita errores de polaridad ante la incorporación o eliminación de dispositivos.
- Para la transmisión de datos realiza procesos de codificación *Manchester II* y modulación *APM*.
- La longitud máxima de cada segmento es de 100 metros. Sin embargo se pueden instalar repetidores que permiten la unión de hasta tres segmentos (300 metros máximo de longitud).
- Permite la interconexión de un máximo de 31 esclavos, donde cada dispositivo habilita el manejo de 4 I/O digitales.
- En operación normal se establece un tiempo de ciclo máximo de  $5\text{ms}$ , para la consulta de los esclavos y añade dos ciclos extras para operaciones de administración del bus, como detección de fallos y control de pausas.

## 3.2. AS-Interface y el Modelo OSI

El modelo OSI aplica al protocolo AS-i en la implementación de tres capas, como se muestra en la figura 2.1 [10]. Esta reducción en el número de capas genera simplicidad en la estructura del datagrama y disminuye el tráfico sobre la red.

La capa de aplicación es la responsable de realizar la interfaz con el usuario (sensor o actuador) e interpretar información desde los puertos I/O que interactúan con el mundo real, a través de las diferentes funciones. La capa de enlace es la encargada de establecer, mantener y terminar las transacciones para la transferencia de datos entre nodos, implementando esquemas de detección de errores (*bit* de paridad), corrección de errores (retransmisión) y manipulación del frame (*bit* de inicio y parada) [11].

Cada nodo es físicamente conectado al canal de comunicación a través de la capa física, la cual depende de las características del medio. Esta interfaz se divide en dos subcapas: la SE (*Symbol Encoding*), la cual convierte los mensajes AS-i en señales eléctricas aptas para la transmisión (codificación *Manchester II*) y la MDP (*Médium Dependant Physical*), la cual ejecuta la respectiva modulación de la señal de tal forma que esta pueda ser ubicada sobre la línea (modulación *APM*<sup>3</sup>).

---

<sup>3</sup> *Alternate Pulse Modulation*.

### 3.3. Tipos de Interfases

El estándar IEC 62026-2 especifica dos tipos de interfaces, las cuales se muestran en la figura 3.1 y se definen a continuación [6]:

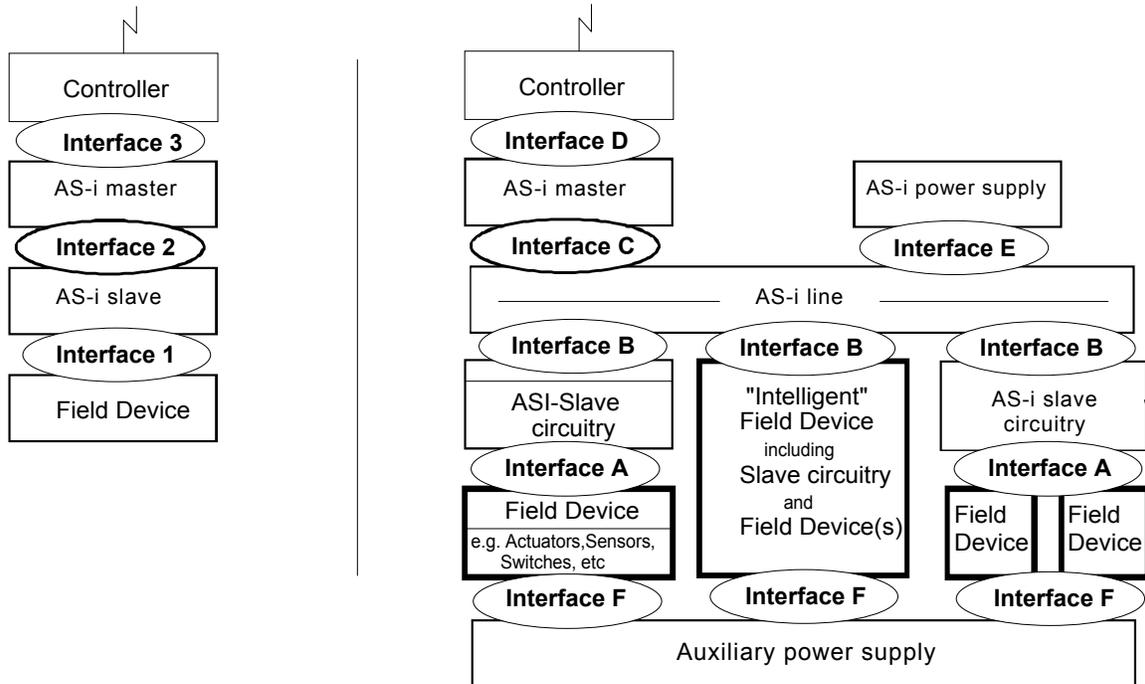


Figura 3.1: Interfases del protocolo AS-i. Lógica (Izquierda) y Física(Derecha).

1. *Interfaz Lógica.* Esta se divide en tres etapas: la primera define los puertos I/O, la parametrización del esclavo y la sincronía de las señales; la segunda, proporciona las definiciones lógicas para el intercambio de datos, incluyendo los procesos de codificación y administración de la red; y finalmente la tercera, proporciona las funciones usadas por el controlador, para acceder al maestro e intercambiar información con el esclavo, manipular banderas, y validar las listas del maestro.
2. *Interfaz Física.* Esta se divide en seis etapas que buscan mantener los requerimientos de señal y potencia. Entre estas se destacan las intefases C y B, las cuales definen la conexión física al bus AS-i, de los dispositivos maestro y esclavo, respectivamente.

### 3.4. Señal Física

La tasa de transmisión de  $166.67\text{kbps}$  del bus AS-i define un tiempo de bit,  $T_{BIT} = 6\mu\text{s}$ , el cual no debe tener una desviación superior a  $\pm 0.2\%$ . Cada transacción se caracteriza por una petición

de maestro y su respectiva respuesta de esclavo, con *bits* fijos de inicio y fin. Estos mensajes son codificados en formato *Manchester II*, donde un "0" lógico genera un flanco negativo de señal, un "1" lógico genera un flanco positivo de señal y el periodo de inactividad se toma como un nivel lógico alto. Esta señal codificada, es modulada con un esquema APM, representado por pulsos de corriente con amplitudes entre  $55mA$  y  $68mA$ , los cuales generan pulsos de voltaje tipo  $Sin^2$  con amplitudes de  $\pm 2V$  sobre el nivel de DC, dado el comportamiento inductivo de la línea y el circuito de desacople que ofrece la fuente AS-i. En la figura 3.2 se muestran las señales típicas del sistema AS-i [6].

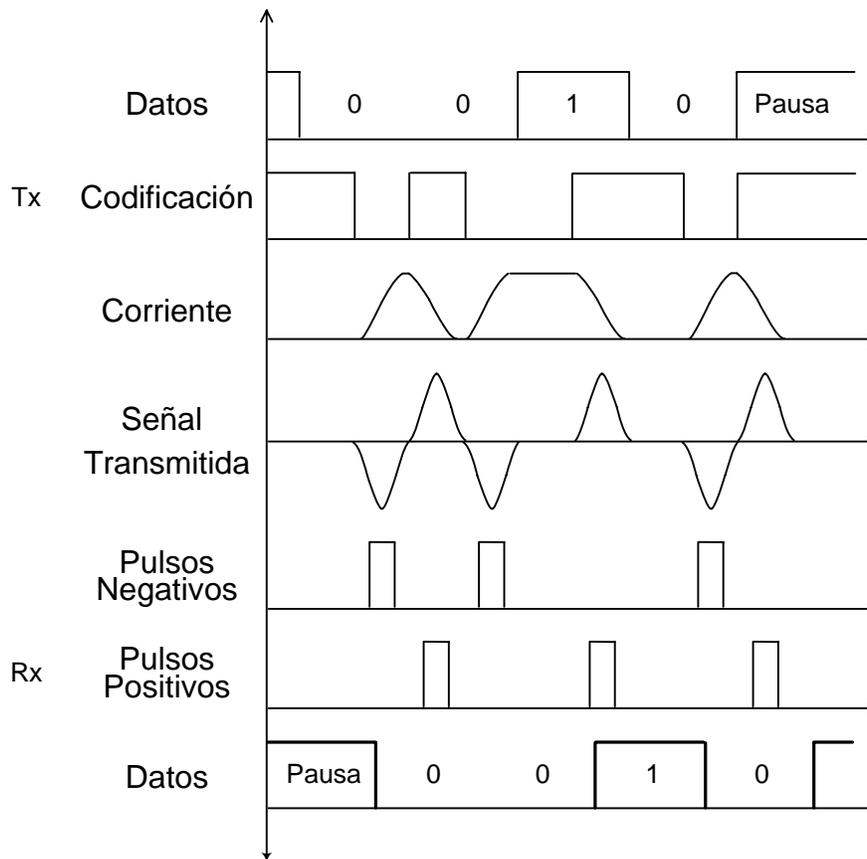


Figura 3.2: Señal AS-i.

La forma de onda de la corriente puede ser representada determinísticamente de la siguiente forma:

Durante un flanco negativo,

$$I_{AS-i}(t) = I_{send} \left[ \frac{t}{3\mu s} - \frac{1}{2\pi} \text{Sin} \left( \frac{2\pi}{3\mu s} t \right) \right] \quad (3.1)$$

y durante un flanco positivo,

$$I_{AS-i}(t) = I_{send} \left[ 1 - \frac{t}{3\mu s} + \frac{1}{2\pi} \text{Sin} \left( \frac{2\pi}{3\mu s} t \right) \right] \quad (3.2)$$

Debido a las inductancias de desacople la forma de onda de la corriente genera un pulso de voltage negativo en cada flanco positivo y un pulso de voltage positivo en cada flanco negativo. Una expresión para describir el comportamiento ideal de esta señal de voltage en el bus AS-i es:

$$V_{AS-i}(t) = \pm V_{send} \left[ \text{Sin}^2 \left( \frac{2\pi}{6\mu s} t \right) \right] \quad (3.3)$$

### 3.5. Proceso de Comunicación

AS-i es un sistema de comunicación tipo maestro-esclavo compuesto de un maestro único y un máximo de 31 esclavos con dirección única, donde el maestro de red intercambia datos y parámetros a la estación esclava, la cual envía información a sus puertos de salida o procesa una petición para retornar el estado de sus puertos de entrada. El estándar emplea un esquema *Cyclic Polling* como el mostrado en la figura 3.3, el cual consiste en el encuestamiento cíclico del estado de cada uno de los esclavos activos, a través de transacciones como la mostrada en la figura 3.4.

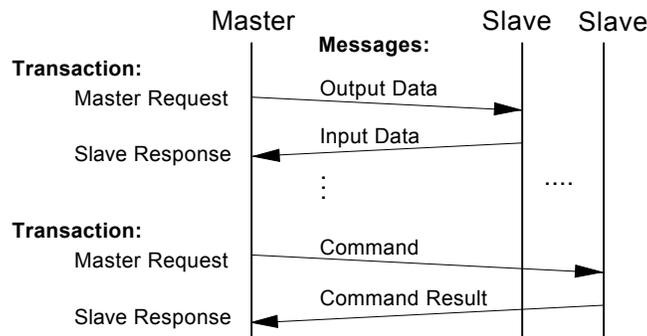


Figura 3.3: Proceso de encuestamiento cíclico.

En esta figura se resaltan intervalos de tiempo de gran importancia durante la implementación del procesador como son:

- *Master Request:* Envío de un mensaje desde el maestro hacia un esclavo.
- *Master Pause:* Durante este tiempo el esclavo procesa la función pedida, genera la respuesta respectiva e inicia este tiempo para enviarsela al maestro. Un esclavo comenzará su respuesta dentro de un periodo de tiempo entre  $2T_{BIT}$  y  $5T_{BIT}$ , después del fin de la petición de maestro. De igual forma el maestro será capaz de aceptar el comienzo de una respuesta de esclavo dentro de un periodo de tiempo entre  $12\mu s$  y  $63\mu s$  después del fin de su petición.

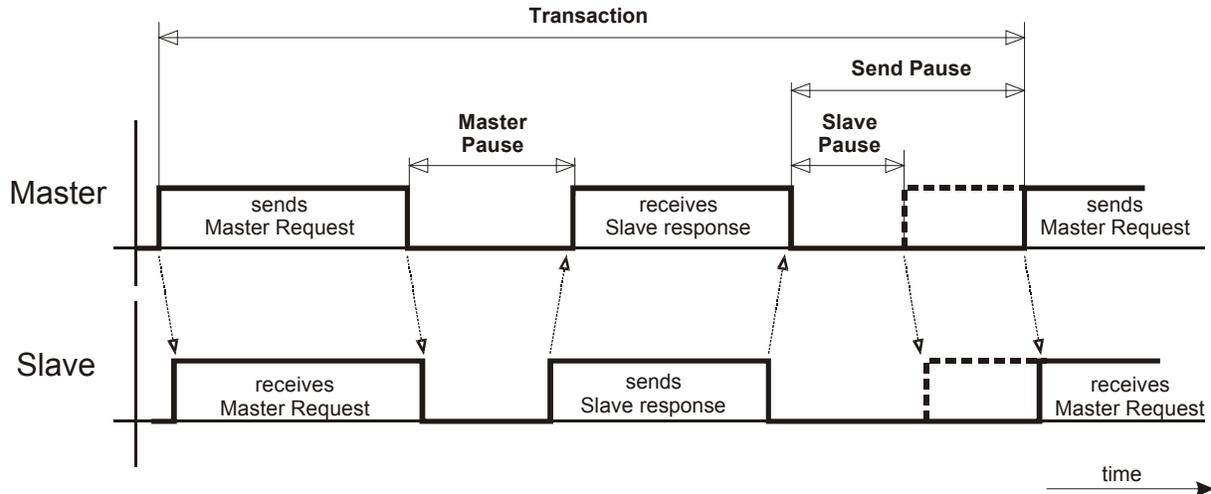


Figura 3.4: Transacción AS-i.

- *Slave Response*: Envío de los datos del esclavo al maestro.
- *Slave Pause*: Después de recibida la respuesta del esclavo, habrá un periodo mínimo durante el cual no ocurrirá la siguiente transmisión. Esta pausa es equivalente a un periodo de tiempo entre  $1.5T_{BIT}$  y  $2T_{BIT}$ . El esclavo será capaz de aceptar el comienzo de una nueva petición de maestro después de una pausa de  $6\mu s$ .
- *Send Pause*: Después de recibida la respuesta del esclavo habrá un periodo mínimo durante el cual no ocurrirá la siguiente transmisión. En operación normal el tiempo de esta pausa será de un *Slave Pause* en caso de tener más de 30 transacciones por ciclo. En el caso de tener 30 o menos transacciones, el *Send Pause* puede ser prolongado a un tiempo máximo de  $500\mu s$ .
- *Slave Response Time-out*: En caso de no recibir respuesta del esclavo en este intervalo de tiempo, el maestro finalizará la transmisión o la reenviará. Este periodo corresponde a la ventana de tiempo de  $11T_{BIT}^{+0\mu s}_{-3\mu s}$ .

Para llevar a cabo este proceso de comunicación el maestro elabora una petición de 14-Bits y el esclavo una respuesta de 7-Bits, de longitud corta y tamaño constante, tal como se muestra en la figura 3.5. Estos datagramas poseen los campos definidos en la tabla 3.1.

### 3.6. Tipos de Transacciones

El sistema AS-i esta en capacidad de procesar dos tipos fundamentales de transacciones; las transacciones simples o sencillas y las transacciones combinadas [6].

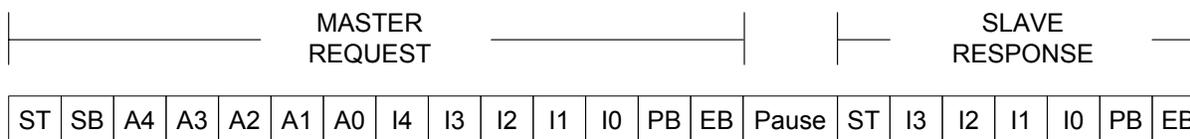


Figura 3.5: Estructura del datagrama AS-i.

Tabla 3.1: Descripción de los campos del datagrama AS-i.

Bit	Descripción
ST	<i>Bit</i> de inicio de datagrama. Fijo a "0".
CB	<i>Bit</i> de Control. 0 = Transmisión de Datos/Parámetros. 1 = Transmisión de Comandos.
A4 - A0	Dirección del Esclavo.
I4	<i>Bit</i> de Información. 0 = Petición de Datos. 1 = Petición de Parámetros.
I3 - I0	<i>Bits</i> de Información.
PB	<i>Bit</i> de Paridad. Paridad Par.
EB	<i>Bit</i> de fin de datagrama. Fijo a "1".

1. *Transacción Simple*. En la tabla 3.2 se muestran los cuatro tipos de transacciones simples, las cuales definen las funciones de intercambio de datos, parametrización, administración de la red y diagnóstico.
2. *Transacción Combinada*. Transacciones definidas para la transferencia de más de 4-*Bits* netos de información, estableciendo un conjunto adicional de reglas. Para identificar esclavos que son capaces de realizar transacciones combinadas se utilizan los *ID-Codes* o *Extended ID-Codes*. En la tabla 3.3 se presentan las diferentes alternativas disponibles en el estándar para la transferencia de valores y funciones especiales.
  - Tipo I: Este tipo de transacción será detallada en la sección 3.7, donde se presentan los conceptos relacionados con la transmisión de variables analógicas.
  - Tipo II: Este tipo de transacción usa el mecanismo de transferencia de datos de AS-i para construir un canal de comunicación serial *full duplex* para la transferencia de *bits* entre el maestro y el esclavo. Esta puede ser usada para en el manejo de sensores y actuadores

Tabla 3.2: Tipos de transacciones AS-i simples.

Tipo	Descripción
Data Exchange	Entrega y/o recibe los patrones de <i>bits</i> hacia/desde los puertos de salida/entrada del dispositivo esclavo.
Write Parameter	Entrega los patrones de <i>bits</i> a los puertos de parámetros del dispositivo esclavo.
Address Assignment	Asignación de dirección no-volátil (1...31) a un dispositivo esclavo para reemplazar la dirección zero.
Commands	<i>Read_identification_code</i> <i>Reserve_R1</i> <i>Read_I/O_configuration</i> <i>Read_status</i> <i>Reset_slave Delete_address</i>

Tabla 3.3: Tipos de transacciones AS-i combinadas.

Tipo	Esclavo	Maestro	Descripción
I	S-7.3	M.3	Entradas - Salidas de 16-Bits.
I	S-7.4	M.3	Dispositivos de campo complejos.
II	S-7.5.5	M.4	Combinación de dispositivos de campo.
II	S-B.A.5	M.4	Comunicación serial en dispositivo de campo.
III	S-7.A.7	M.4	4I/4O en modo de direccionamiento extendido.
III	S-7.A.A	M.4	8I/8O en modo de direccionamiento extendido.
IV	S-7.A.8	M.4	Entradas de 16-Bits en direccionamiento extendido.
IV	S-7.A.9	M.4	Entradas duales de 16-Bits en modo extendido.
V	S-6.0	M.4	Entradas - Salidas de 16-Bits de alta velocidad.
<i>Safety</i>	S-7.B	–	Cualquier esclavo de entrada para trabajo seguro.

analógicos, dispositivos de campo con parámetros variables y el reemplazo de interfases de 4 – 20mA.

- Tipo III: Este tipo de transacción usa el mecanismo de transferencia de datos de AS-i para construir canales de comunicación *full duplex* de 4-Bits u 8-Bits entre el maestro y el esclavo, en el modo de direccionamiento extendido. Esta transacción es usada específicamente en el manejo de teclados, torres visualizadoras de señal, válvulas terminales y sensores/actuadores de 8-Bits.
- Tipo IV: Este tipo de transacción usa el mecanismo de transferencia de datos de AS-i para construir un canal de comunicación sencillo o dual de 16-Bits para transferencia de datos

desde el esclavo al maestro en el modo de direccionamiento extendido. Este tipo es montado para el uso de sensores de *16-Bits* de canal sencillo o dual y el reemplazo de interfases de  $4 - 20mA$ .

- Tipo V: Este tipo de transacción usa el mecanismo de transferencia de datos de AS-i para construir canales de comunicación *full duplex* rápida, de *8-Bits*, *12-Bits* o *16-Bits* para transferencia de datos desde el maestro al esclavo usando 2, 3 o 4 direcciones estándar. Este tipo es montado para el uso de sensores/actuadores de *16-Bits* y alta velocidad, y el reemplazo de interfases de  $4 - 20mA$  por lazos de control.

## 3.7. Manejo de Variables Analógicas

En esta sección se introducen dos conceptos fundamentales para el entendimiento del proceso de comunicación llevado a cabo por el sistema AS-i, para el manejo de variables analógicas. Estos dos conceptos hacen referencia al uso de transacciones combinadas tipo I para la implementación del perfil *S - 7.3* de dispositivos esclavos.

### 3.7.1. Transacción Combinada Tipo I

Esta transacción usa el mecanismo de transferencia de datos de AS-i para construir un canal de comunicación *half duplex* para transferencia de *bytes* entre el maestro y el esclavo. Esta puede ser usada para el manejo de sensores y actuadores analógicos, *Scanners*, visualizadores de 8 caracteres máximo, dispositivos de campo con parámetros variables y reemplazo de interfases de  $4 - 20mA$ . El manejo de este tipo de transacciones define un arreglo de tres *bits* de información para la transferencia de datos a través de la transacción sencilla *Data\_exchange*, como se muestra en la tabla 3.4:

Tabla 3.4: Definición de los *bits* de datos I/O en transacciones tipo I.

<b>Bit</b>	<b>Descripción</b>
D3	<i>Toggle Bit</i> : Bit de control para la transferencia de datos.
D2	Bit más significativo de la tripleta de datos.
D1	Bit de la tripleta de datos.
D0	Bit menos significativo de la tripleta de datos.

El intercambio de datos se puede generar siguiendo dos esquemas. El primero se presenta cuando el maestro funciona como fuente de datos, en este caso la transferencia es controlada por el esclavo según el esquema mostrado en la tabla 3.5. El esclavo realiza un comando de petición de *3-Bits* y

el maestro responde en el siguiente telegrama *Data\_Exchange*. La transferencia obliga el uso de la petición de inicio  $K111$  y la de fin  $K000$ , de tal forma que si se envían menos de 23-Bits, se puede omitir el arreglo  $K110$ . El último *bit* corresponde al estado de validación ( $V = 1$ ), si la secuencia ha sido recibida correctamente. Este esquema aplica al manejo de actuadores analógicos.

Tabla 3.5: Transferencia de datos del maestro al esclavo en transacciones tipo I.

<i>Master Request</i>				<i>Slave Response</i>			
$K$	X	X	X	$\bar{K}$	1	1	1
$K$	D2	D1	D0	$\bar{K}$	1	1	0
$K$	D2	D1	D0	$\bar{K}$	1	0	1
$K$	D2	D1	D0	$\bar{K}$	1	0	0
$K$	D2	D1	D0	$\bar{K}$	0	1	1
$K$	D2	D1	D0	$\bar{K}$	0	1	0
$K$	D2	D1	D0	$\bar{K}$	0	0	1
$K$	D2	D1	D0	$\bar{K}$	0	0	0
$K$	D2	D1	V	$\bar{K}$	1	1	1

El segundo esquema, el cual es de interes directo en este trabajo por aplicar al manejo de sensores analógicos, se presenta cuando el esclavo funciona como fuente de datos, en este caso la transferencia es controlada por el maestro según el esquema mostrado en la tabla 3.6. El maestro realiza un comando de petición de 3-Bits y el esclavo responde en el siguiente telegrama *Data\_Exchange*. La petición  $K111$  sirve como comando *latch*, ya que después de recibirlo, los *bits* de datos no serán actualizados por la fuente hasta recibir la secuencia completa para asegurar la consistencia del mensaje.

Tabla 3.6: Transferencia de datos del esclavo al maestro en transacciones tipo I.

<i>Master Request</i>				<i>Slave Response</i>			
$K$	1	1	1	$\bar{K}$	D2	D1	D0
$K$	1	1	0	$\bar{K}$	D2	D1	D0
$K$	1	0	1	$\bar{K}$	D2	D1	D0
$K$	1	0	0	$\bar{K}$	D2	D1	D0
$K$	0	1	1	$\bar{K}$	D2	D1	D0
$K$	0	1	0	$\bar{K}$	D2	D1	D0
$K$	0	0	1	$\bar{K}$	D2	D1	D0
$K$	0	0	0	$\bar{K}$	D2	D1	V

Independiente de la dirección de la transferencia de datos, el *bit* de control (*Toggle Bit*) es invertido por el maestro, cada vez que el contenido de los *bits* de datos o comandos cambian.

Respecto de los tiempos de respuesta, si el esclavo funciona como entrada, cuando el maestro direcciona una nueva tripleta de datos, este responderá dentro de  $250\mu s$  con un nuevo valor. De forma contaria, si el esclavo funciona como salida, despues de recibido el comando *K111*, el maestro cambia el *Toggle Bit* y envía un nuevo dato. El esclavo responderá con una nueva petición dentro de  $250\mu s$ , pero puede esperar un tiempo máximo de 3s para comenzar una nueva secuencia. Este proceso continua hasta que el esclavo direcciona los últimos datos con el *bit* de validación.

### 3.7.2. Perfil Esclavo S.7.3

Este perfil esclavo fue reglamentado para la transferencia de valores analógicos, a través del intercambio de dos *bytes* de información con sensores o actuadores, mediante el uso de las transacciones combinadas tipo I referenciadas en la sección 3.7.1

Para definir la dirección de la transferencia de datos y el número de canales del esclavo se utiliza el código extendido ID2 a través de la transacción sencilla *Read\_ID2-Code*, de acuerdo con la información presentada en la tabla 3.7:

Tabla 3.7: Definición de los *bits* de parametros en transacciones tipo I.

I3	I2	I1	I0	Descripción
X	X	0	0	Esclavo posee un canal.
X	X	0	1	Esclavo posee dos canales.
X	X	1	0	Esclavo posee cuatro canales.
1	1	1	1	Esclavo posee cuatro canales. Solo para esclavos que no soportan códigos extendidos ID1/2.
X	0	X	X	Modo Transparente.
X	1	X	X	Transferencia de valores analógicos.
0	X	X	X	Esclavo configurado como salida.
1	X	X	X	Esclavo configurado como entrada.

El manejo de la transmisión se realiza en paquetes de tres *bits* netos de información más un *bit* de control (*Toggle Bit*), tal como se especificó en los esquemas de transferencia mostradas en las tablas 3.5 y 3.6, lo cual implica el uso de aproximadamente 7 ciclos AS-i. En la tabla 3.8 se muestra una tranferencia completa de una palabra de 16-*Bits*.

Esta representación corresponde a información concreta acerca del dispositivo esclavo, los tres

Tabla 3.8: Arreglo de datos para la transferencia de valores analógicos de 16-Bits.

Extensión	Información						Adicional
E3-E1	D16-D14	D13-D11	D10-D8	D7-D5	D4-D2	D1	O V
1 1 1	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0	0 0

primeros *bits* (E3-E1) indican el canal que esta proporcionando el mensaje, los siguientes 16 *bits* (D16-D1) muestran el valor del dato transferido, el *bit* bandera en caso de *overflow* y el *bit* de validación. Los datos son representados en formato complemento a dos<sup>4</sup>, con una longitud fija de 16-Bits. Para sensores o actuadores que poseen menor resolución de conversión, se llevan a cero los *bits* menos significativos. En la tabla 3.9 se muestra el formato de representación para la transmisión de datos con 12-Bits y 8-Bits de resolución.

Tabla 3.9: Representación de datos de 12-Bits y 8-Bits.

#Bits	D16	D15	D14	D14	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
16	0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1
12	0	1	0	0	0	1	1	0	0	1	1	1	0	0	0	0
8	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0

### 3.8. Requerimientos del Receptor

En un sistema AS-i real tanto las formas de onda como las amplitudes descritas en la sección 3.4, estarán influenciadas por las propiedades físicas de la línea. Por tal motivo los receptores deben ser capaces de detectar pulsos dispersos como los mostrados en la figura 3.6, los cuales deben cumplir con las siguientes características mínimas:

- La máxima amplitud del pulso  $V_{max}$  del mensaje puede variar entre  $1.5V_{pico}$  y  $4V_{pico}$ . Las diferencias de amplitud entre peticiones consecutivas del maestro no variarán para una misma configuración, es decir, los valores mostrados en la figura 3.6 representan el peor caso en el cual se realicen diferentes configuraciones y ubicaciones de las estaciones esclavas sobre la línea. Finalmente, para una configuración uniforme, la máxima variación de  $V_{max}$  entre dos respuestas de esclavo ubicados en diferentes posiciones de la línea será en una escala de  $1 : 1.5$ .

<sup>4</sup> En este formato el *bit* D16 representa el *bit* de signo.

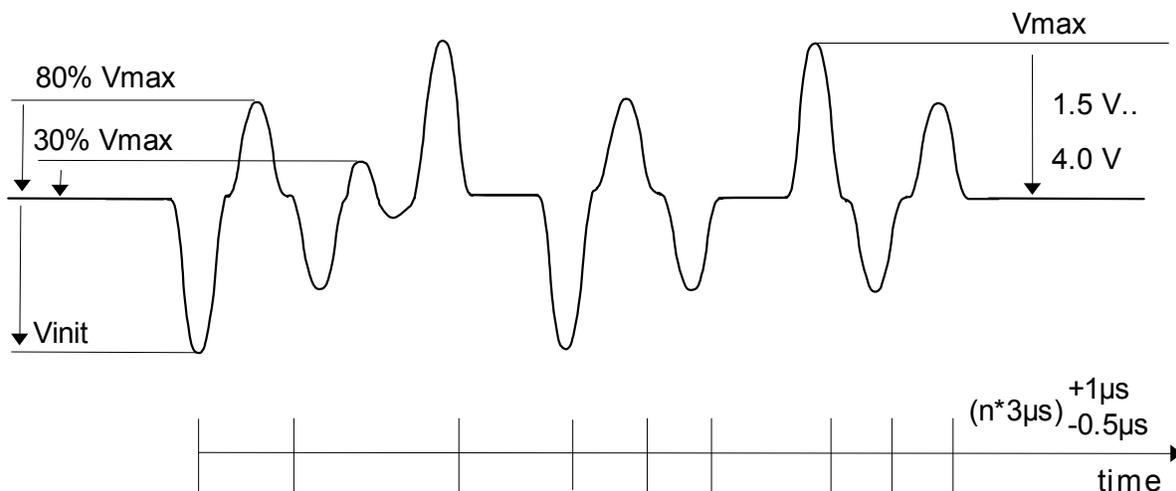


Figura 3.6: Requerimientos del Receptor AS-i.

- La amplitud de un pulso válido dentro de un mensaje puede variar desde el 65 % al 100 % de la amplitud máxima  $V_{max}$ . De acuerdo a versiones previas de la norma el rango de variación puede ser desde el 80 % al 100 %.
- Los pulsos validos que deben ser aceptados por el receptor comienzan en una ventana de tiempo de  $(3 \mu s \cdot n)_{-0.5 \mu s}^{+1.0 \mu s}$  en relación al pulso inicial  $V_{init}$ .
- Los pulsos fuera de una ventana de tiempo de  $(3 \mu s \cdot n)_{-0.8 \mu s}^{+1.6 \mu s}$  deben ser rechazados por el receptor. Estas desviaciones pueden ocurrir debido a la combinación de diferentes efectos como cargas capacitivas sobre la línea, o desviaciones en la frecuencia del oscilador del transmisor o del receptor.
- Los pulsos de ruido (noise) o rizado (ringing) de hasta un 30 % de  $V_{max}$  no deben perturbar la recepción del mensaje.
- La máxima desviación del tiempo nominal de *bit* ( $3 \mu s$ ) debe ser menor o igual a  $\pm 0.1 \%$  para el maestro y menor o igual a  $\pm 0.2 \%$  para el esclavo.

### 3.9. Manejo de Errores

La detección de errores es un proceso necesario para ignorar cualquier datagrama que haya sido deteriorado por efectos del ruido sobre la línea. Cualquier petición del maestro<sup>5</sup> o respuesta

<sup>5</sup> *master request*.

de esclavo<sup>6</sup> sobre la línea AS-i será sometido a un chequeo por parte del receptor, ante la posible ocurrencia de algunos de los siguientes errores de transmisión [6]:

- *Start\_bit\_error*: se produce cuando el pulso inicial que sigue a una pausa no es de polaridad negativa. Este *bit* es la referencia del proceso de decodificación.
- *Alternating\_error*: se genera cuando dos pulsos consecutivos no son de polaridad diferente.
- *No\_information\_error*: Los pulsos de cualquier petición o respuesta deben ser detectados dentro de una ventana de tiempo de  $(3\mu s \cdot n)_{-0.5\mu s}^{+1.0\mu s}$  después del pulso inicial, donde  $n=26$  para el maestro y  $n=12$  para el esclavo.
- *Parity\_error*: se produce cuando la suma de todos los *bits* de información de un datagrama no cumple con el tipo de paridad par estipulado por la norma.
- *End\_bit\_error*: se genera cuando el pulso final del datagrama no es de polaridad positiva. Este *bit* corresponde al tiempo  $(6\mu s \cdot n)$ , donde  $n=13$  para el maestro y  $n=6$  para el esclavo.
- *Length\_error*: error producido cuando se detecta algún tipo de señal durante los periodos de pausa.

---

<sup>6</sup> *slave response*.

# Capítulo 4

## Diseño de Hardware

En este capítulo se describirá la interfaz *Hardware* que se implementó cumpliendo los requerimientos físicos del protocolo, para transmitir datos sobre la línea de alimentación en DC. Cabe resaltar que este desarrollo se puede considerar una interfaz universal en la cual es posible implementar cualquiera de los perfiles funcionales definidos por el estándar [12], con tan solo cambiar la programación del microcontrolador.

### 4.1. Sistema de desarrollo AdaptHC9S12E128

En esta aplicación, para el manejo de sensores y actuadores analógicos, se resalta el proceso de generación de patrones de transmisión y detección de pulsos, mediante el uso de dispositivos con módulos de conversión ADC y DAC, que poseen buena velocidad, resolución y tiempo de respuesta [13]. Para lograr un resultado exitoso se requiere un equilibrio entre requerimientos de desempeño y la complejidad de la implementación, razón por la cual se adoptó el microcontrolador MC68HC9S12E128 de la familia HCS12 de *Motorola*. En la figura 4.1 se muestra un diagrama funcional de la arquitectura de este *core*.

Este microcontrolador de 16-Bits esta equipado con 128Kbytes de memoria EEPROM<sup>1</sup>, permitiendo su reprogramación. La MCU<sup>2</sup> puede funcionar con ayuda del módulo PLL<sup>3</sup> a una velocidad máxima de 25MHz, usando un oscilador externo de 8MHz. Existen ventajas como la disponibilidad de herramientas de programación por puerto USB (Interfaz DBM), depuración y aplicación del Procesador Experto a través del *Software Codewarrior*. Adicionalmente, cuenta con módulos periféricos de gran utilidad para esta aplicación [14], entre los cuales se destacan:

---

<sup>1</sup> *Electrically Erasable Programmable Read Only Memory.*

<sup>2</sup> *Microcontroller Unit.*

<sup>3</sup> *Phase Locked Loop.*

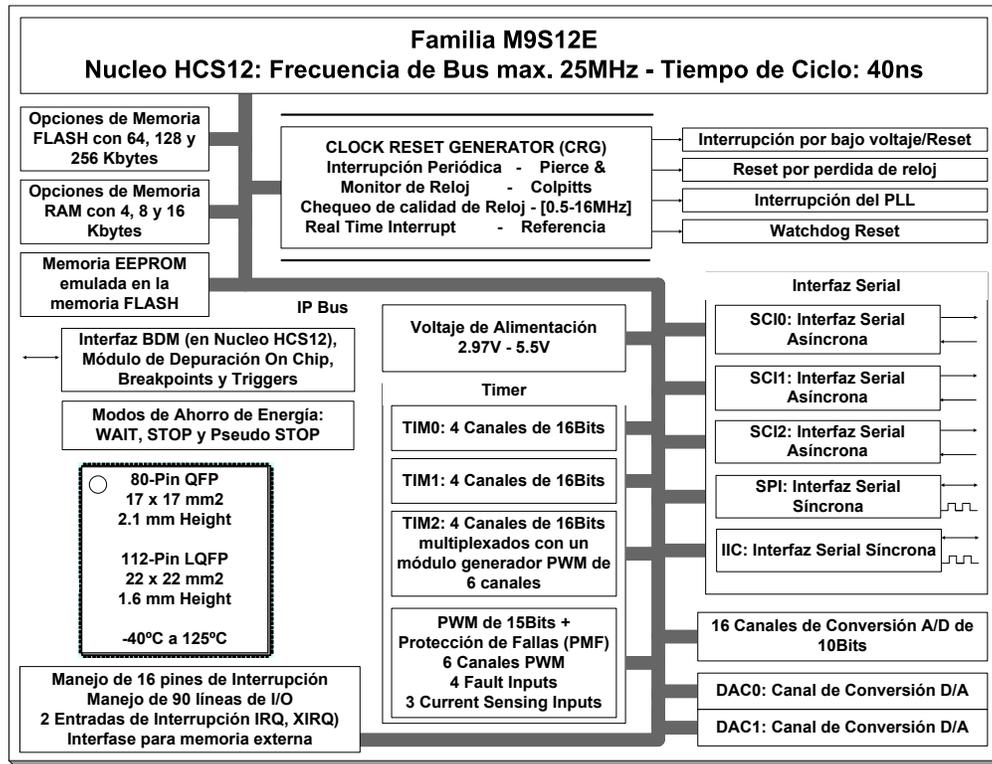


Figura 4.1: Arquitectura del HCS12.

- *GPIO: Pines de Propósito General.* En esta aplicación fueron usados como interfaz I/O para los sensores y actuadores binarios que maneja el perfil funcional  $S - 3,0$  de dispositivos esclavos. Adicionalmente actúa como interfaz de control para el proceso de conmutación entre las etapas de recepción y transmisión.
- *Timer & Counter.* Los *timers* fueron utilizados para generar las ventanas de tiempo necesarias en la manipulación de las transacciones AS-i, incluyendo el control de las pausas y longitudes de trama, según los requerimientos expuestos en la sección 3.5. Adicionalmente se hizo uso de la función especial *input\_capture* que ofrecen los *timer* IOC6 e IOC7 para actuar como interfaz de entrada de los canales de salida de la etapa de comparación, que detectan la secuencia de pulsos positivos y negativos capturados del bus AS-i.
- *SCI: Interfaz Serial.* Este periférico se usó para habilitar la comunicación USB entre la aplicación maestra y el PC a través del *driver* integrado *FTDI232BM*, para la visualización del estado operativo del esclavo analógico.
- *ADC: Convertidor Análogo Digital.* El módulo ADC de 10-Bits que posee el HCS12 se utilizó para capturar la señal de un potenciómetro que simula el comportamiento de una entrada

analógica.

- *DAC: Convertidor Digital Análogo.* El módulo DAC de 8-Bits que posee el HCS12 se utilizó para generar los patrones analógicos de corriente de la señal AS-i, actuando como interfaz de salida del Codificador *Manchester II* y el Modulador *APM* implementados por *software* en el microcontrolador. Este módulo se convierte en uno de los principales argumentos de selección del HCS12, sumado a la gran exigencia de operación a  $25\text{MHz}$  y, existencia de poderosas herramientas de depuración y compilación, ya que el manejo de conveyores externos demanda una alta complejidad de configuración y adecuación.

Ademas de la tarjeta de desarrollo comercial utilizada se implementó la tarjeta de aplicación mostrada en la figura 4.2, para adaptar el *core* a las necesidades del diseño. En esta tarjeta solo se habilitaron los pines y periféricos necesarios para cumplir con los requerimientos del procesador e implementar varios nodos esclavos que permitieran validar el funcionamiento en red de los dispositivos. La tarjeta diseñada demostró tener un buen desempeño al presentar las mismas prestaciones que el sistema comercial, con respecto a la aplicación. En el anexo ?? se presenta el respectivo diagrama esquemático de esta tarjeta.

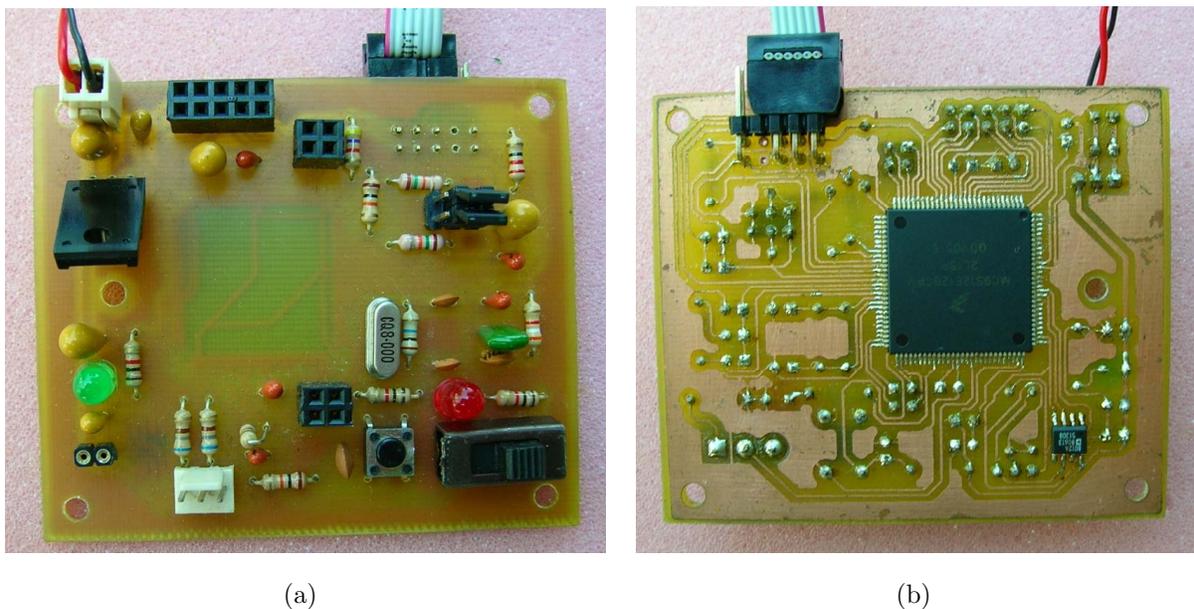


Figura 4.2: Tarjeta de aplicación diseñada. (a) Cara superior. (b) Cara posterior.

## 4.2. Diagrama de Bloques

En la figura 4.3 se presenta un diagrama de bloques, con cada uno de los subsistemas individuales que componen el procesador de comunicaciones AS-i. Todas las capas superiores a la capa

MDP mencionada en la sección 3.2, serán presentadas en la capítulo 5, ya que fueron desarrolladas como parte de la programación del microcontrolador. A continuación, se realizará una descripción de cada uno de estos módulos funcionales.

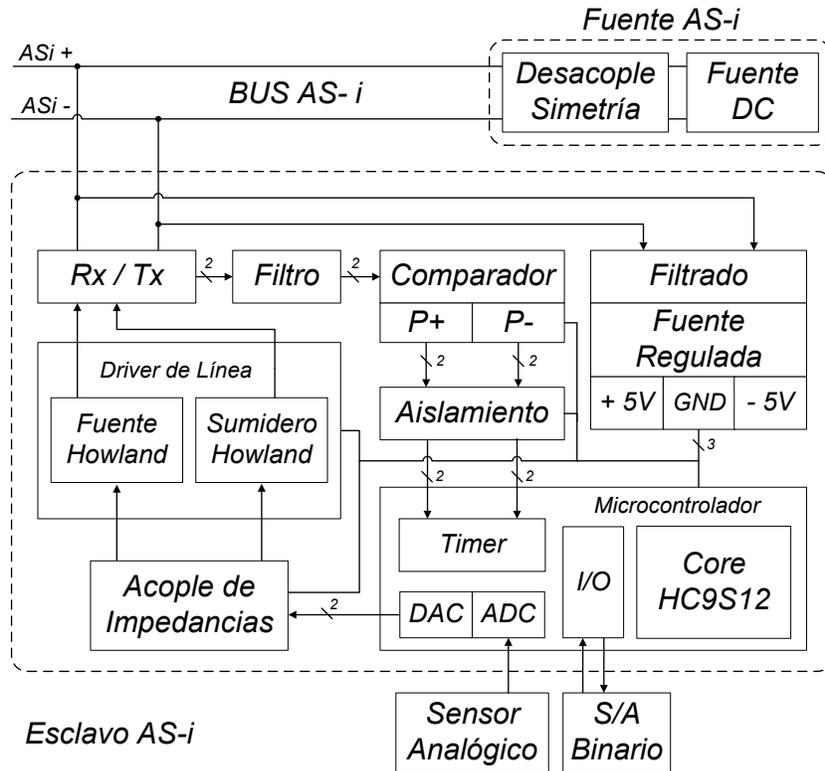


Figura 4.3: Diagrama de bloques del procesador AS-i.

### 4.3. Transmisor AS-Interface

Los patrones analógicos de tensión correspondientes a la señal de corriente AS-i, fueron generados usando el módulo de conversión DAC del microcontrolador, para lo cual se determinó una representación de 30 muestras por  $T_{BIT}$ , maximizando la velocidad de muestreo de acuerdo al tiempo de establecimiento característico de este periférico. Este proceso de generación es argumentado en la sección 5.5. Uno de los componentes más críticos en el diseño de cualquier sistema PLC (*Power Line Communication*) es el circuito interfaz o circuito de acople (*coupling circuit*) con la red de distribución de potencia, ya que esta puede llegar a ser compleja considerando las características del canal. Debido al manejo de altos voltajes, variación de impedancias, grandes amplitudes y las perturbaciones dependientes del tiempo, los circuitos de acople necesitan ser cuidadosamente diseñados para garantizar la transmisión de la señal con el ancho de banda y los

niveles de potencia adecuados [15]. En el anexo D se presentan varias consideraciones prácticas sobre el diseño de este tipo de circuitos, donde se pueden presentar acoples inductivos, capacitivos y algunas configuraciones híbridas.

### 4.3.1. Interfaz de Acople

Para garantizar el acople del módulo DAC al sistema analógico se diseñó una interfaz de aislamiento. En primera instancia se implementó un transformador integrado con relación 1:1<sup>4</sup> [16], para proteger el sistema digital y aislar las referencias de alimentación, pero pruebas iniciales no permitieron un buen desempeño, razón por la cual se acondicionó un seguidor de emisor con entrada y salida desacopladas en DC como el mostrado en la figura 4.4, para evitar la saturación del transformador y garantizar la corriente de funcionamiento [17], [18].

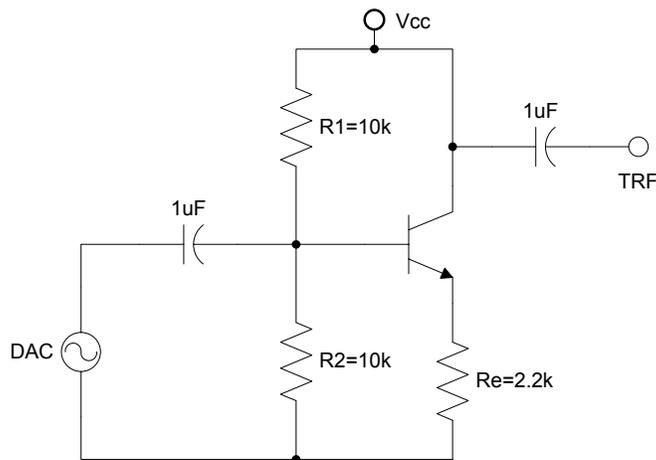


Figura 4.4: Seguidor de emisor.

Esta solución no resultó ser la más adecuada, ya que generaba un fenómeno de distorsión particular en los pulsos de inicio y fin de los patrones analógicos que representan los datagramas, impidiendo el reconocimiento de los mismos y causando pérdida de sincronía en el proceso de decodificación, al ser detectados errores como el *bit* de inicio y la alternancia (revisar la sección 3.9). Dicho fenómeno el cual fue simulado previamente, como resultado de comprobación, se generaba en el momento de acoplar la carga de tipo inductivo (transformador) al condensador de desacople, lo cual no garantizaba la región de operación del transistor debido a la rata de cambio de la señal de voltage; por tal razón no se recomienda el uso de la configuración discreta mostrada en la figura 4.5.

Para evitar este tipo de inconvenientes se optó por reemplazar esta topología por un acople de alta impedancia usando un *buffer* integrado, con el opeacional de proposito general AD8032

<sup>4</sup> Referencia *WB2010 – PCL* de *Coilcraf*.

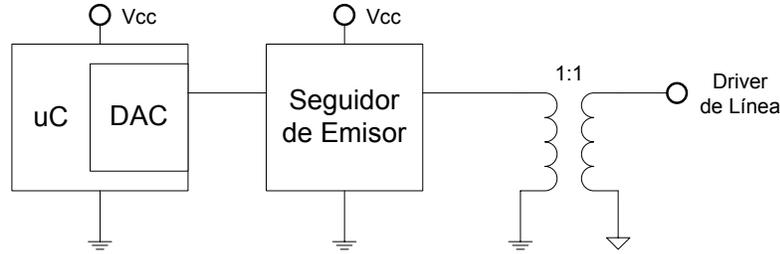


Figura 4.5: Circuito de acople discreto.

configurado como seguidor, el cual actúa como interfaz directa entre el sistema digital y el *driver* de línea, de acuerdo con el esquema mostrado en la figura 4.6

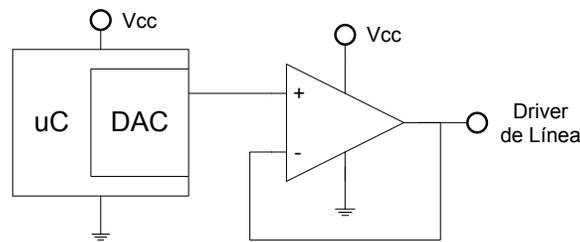


Figura 4.6: Circuito de acople definitivo del sistema digital.

### 4.3.2. Driver de línea

La salida del seguidor es directamente acoplada al *driver* de línea, el cual fue diseñado con base en el concepto de conversión tensión-corriente del generador *Howland* [19], [20], [21]. Debido a que la interfaz física del estándar define como medio de transmisión una línea diferencial, se aisló la referencia negativa del sistema a través del diseño de un generador de corriente diferencial como el mostrado en la figura 4.7, el cual actúa como fuente y sumidero para una carga flotante, y garantiza el acople de la señal modulada al bus. Para esta conexión fue necesario hacer un desacople capacitivo a la salida del generador de corriente de tal forma que no se reflejara un nivel de continua que afectara la integridad del *driver*.

Si  $R_2$  ( $R_8$ ) es mucho más grande que  $R_5$  ( $R_{11}$ ), entonces se puede asumir que  $I_L$  solo fluye por la carga, de tal forma que puede ser aplicado el análisis matemático presentado en los anexos A y B. Típicamente  $R_5$  está entre un valor de  $1\Omega$  y  $100\Omega$ , mientras  $R_2$  ( $R_8$ ) está entre un valor de  $10k\Omega$  y  $100k\Omega$ , con el fin de ajustar las impedancias de entrada y salida de la configuración. En algunas aplicaciones es necesario colocar un capacitor en paralelo con  $R_2$ , para proporcionar una red de realimentación de un solo polo dominante que evite oscilaciones. Así mismo un capacitor en paralelo con  $R_3$  asegura que la realimentación positiva sea menor que la negativa cuando la

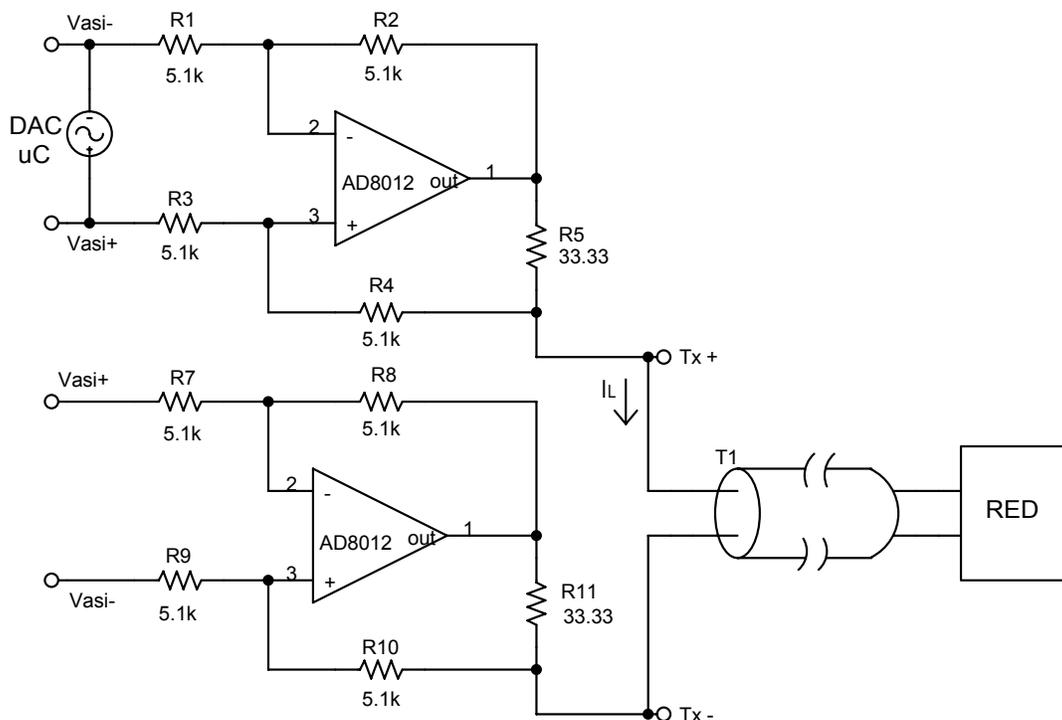


Figura 4.7: Diagrama circuital del driver de línea.

fuentes de alimentación sea encendida, sin embargo en la implementación final no fueron tenidos en cuenta. Un aspecto importante en la caracterización de la fuente de corriente era definir el rango de operación, para garantizar su funcionamiento lineal, para lo cual se realizaron simulaciones con ayuda de Matlab, cuyos resultados son mostrados en el anexo G y permiten evidenciar que bajo condiciones de operación normal la impedancia característica de la línea no es superior a  $220\Omega$ .

El operacional AD8012 seleccionado para implementar este circuito, presenta excelente desempeño en DC, especialmente bajo voltage de *Offset*, bajas derivas y alto CMRR [22], los cuales son parámetros importantes en el diseño de una fuente de corriente de precisión controlada por voltage, usando el esquema básico del generador *Howland*, para proporcionar una corriente de carga bidireccional generada por una entrada de voltage diferencial. El desempeño y precisión de este circuito depende de las tolerancias de los resistores seleccionados. El resistor de ajuste,  $R_{Scale}$  ( $R_5 = R_{11}$ ) y los cuatro resistores de realimentación de la topología  $R_{Feedback}$  ( $R_n = R$ ), afectan directamente el nivel de corriente en la carga y la impedancia de salida de la fuente, tal como se muestra en los apéndices A y B.

Según la aplicación presentada, el voltage de entrada al *Driver* es de  $2V_{P-P}$  (revisar sección 5.5) y la corriente deseada de  $60mA$ , exigen un valor de  $33.33\Omega$  en  $R_{Scale}$ . De esta forma el error introducido por las desviaciones del operacional es el siguiente [23]:

$$I_{Error} = 2 \left[ \frac{V_{OS}}{R_{Scale}} \right] + I_{OS} \left[ \frac{R_{Feedback}}{R_{Scale}} \right] \quad (4.1)$$

Tomando los valores para  $V_{OS}$  e  $I_{OS}$  del *datasheet* proporcionado por el fabricante se obtienen los siguientes resultados a una temperatura base de  $25^{\circ}C$ :

Asumiendo la desviación típica de la corriente,  $I_{OS} = 3\mu A$

$$I_{Error} = 2 \left[ \frac{1.5mV}{33.33\Omega} \right] + 3\mu A \left[ \frac{5.1k\Omega}{33.33\Omega} \right] = 549\mu A \quad (4.2)$$

Para lo cual se obtiene una desviación porcentual típica de:

$$Error_{Típico} = \frac{I_{Error}}{I_{Source}} = \frac{549\mu A}{60mA} = 0.915\% \quad (4.3)$$

Asimismo, asumiendo la desviación máxima de la corriente,  $I_{OS} = 12\mu A$

$$I_{Error} = 2 \left[ \frac{1.5mV}{33.33\Omega} \right] + 12\mu A \left[ \frac{5.1k\Omega}{33.33\Omega} \right] = 1.92mA \quad (4.4)$$

Se obtiene una desviación porcentual máxima de:

$$Error_{Máximo} = \frac{I_{Error}}{I_{Source}} = \frac{1.92mA}{60mA} = 3.2\% \quad (4.5)$$

En conclusión, bajas derivas en los operacionales y resistores con baja tolerancia son requeridos para lograr una fuente de corriente de precisión. Los resultados presentados anteriormente dejan evidenciar para el peor caso ( $3.2\%$ ) una máxima variación de corriente entre  $58.08mA$  y  $61.92mA$ , valores que se encuentran entre los límites exigidos por la norma,  $55mA$  y  $68mA$ , respectivamente. Para argumentar la variabilidad de los niveles de corriente en la topología del *driver* diferencial se llevaron a cabo algunas pruebas, las cuales son presentadas en el anexo E

## 4.4. Receptor AS-Interface

La señal de voltaje sobre el bus AS-i equivale a un arreglo de pulsos tipo  $Sin^2$ , los cuales se generan si se garantiza la forma de onda de corriente en el circuito de desacople y simetría que ofrece la fuente AS-i. Para realizar el proceso de recepción se diseñó el esquema de detección de pulsos mostrado en las figuras 4.8 y 4.9. En primera instancia se implementó un desacople en continua de tipo capacitivo ( $C_{f1} - C_{f2}$ ) y una etapa de aislamiento con transformador ( $TX1$ ) [16], para extraer la señal mensaje del bus. Este patrón de voltaje es llevado a una etapa de amplificación usando la referencia AD8031, en configuración no inversora para los pulsos positivos e inversora para

los negativos, con el objetivo de mejorar la resolución de la etapa de comparación implementada con la referencia *AD8561*, como se muestra en la figura 4.8. De esta forma se construyeron dos canales de salida independientes para los pulsos detectados, los cuales fueron acoplados a las entradas del módulo *timer* del microcontrolador, configurados como detector de flanco (modo *input\_capture*), a través del aislador digital *ADuM1200*, como se muestra en la figura 4.9. Este sistema habilita completamente el proceso de reconstrucción de la información con el respectivo algoritmo de decodificación diseñado. En la sección 6.2.2 se presentan pruebas de desempeño de esta etapa.

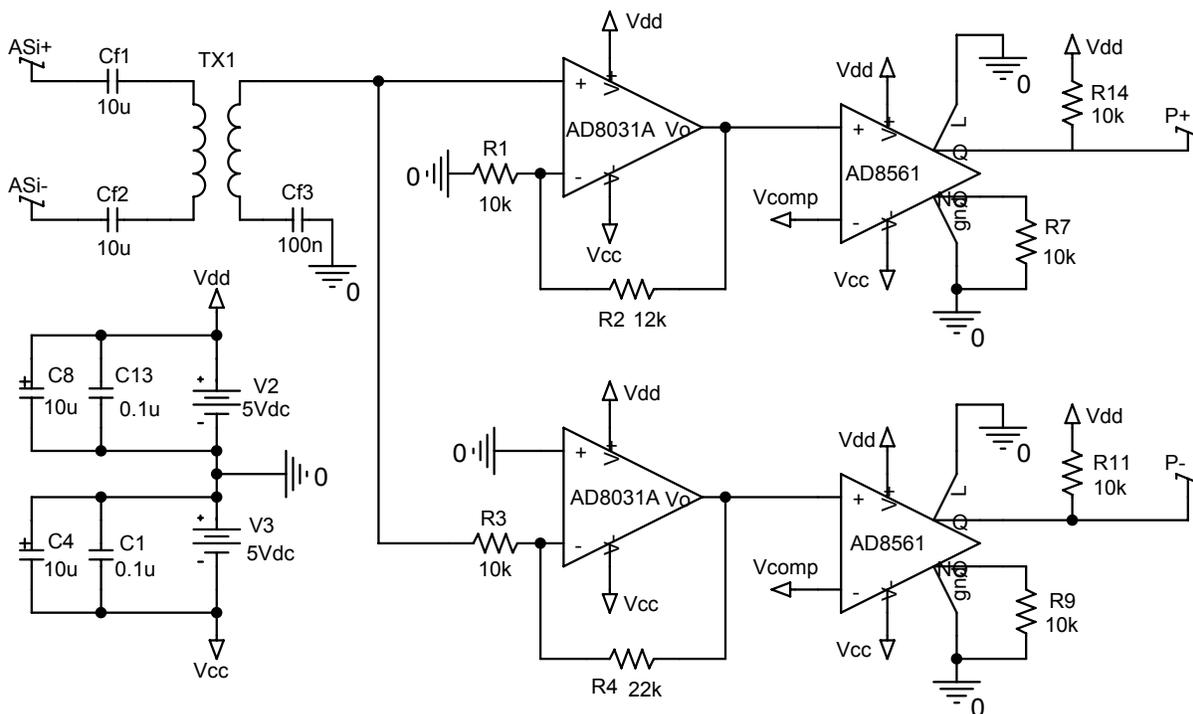


Figura 4.8: Etapas de amplificación y comparación del circuito de recepción.

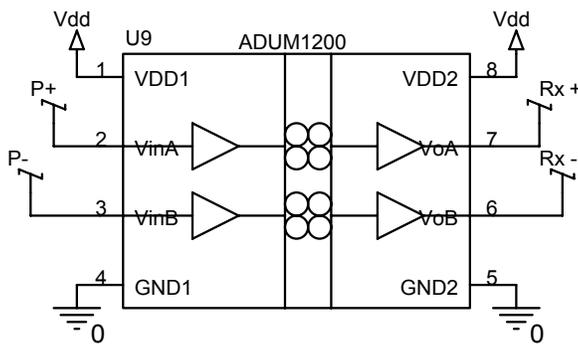


Figura 4.9: Etapas de aislamiento digital y nivel de comparación del circuito de recepción.

## 4.5. Circuitos Complementarios

Adicional a los bloques principales anteriormente descritos se diseñaron circuitos complementarios para el procesador de comunicaciones AS-i, como son el circuito de conmutación y el esquema de alimentación.

### 4.5.1. Circuito de Conmutación Rx/Tx

Debido a que el modo de transmisión definido por AS-i es *Half Duplex*, donde los componentes de red están en capacidad de transmitir y recibir en ambas direcciones, pero no de forma simultánea, se consideró un circuito de conmutación que habilita las funciones del transmisor y el receptor del procesador, en las ventanas de tiempo presentadas en la sección 3.5 para ejecutar la transacción AS-i. El receptor se habilita cuando el maestro desarrolla el encuestamiento cíclico de los esclavos activos, una vez se detecta la petición y se decodifica, se conmuta el *switch* para habilitar el transmisor que se encarga del envío de la respectiva respuesta. Para el correcto funcionamiento de este circuito se utilizó un multiplexor analógico *ADG884* controlado por el microcontrolador [24], el cual se caracteriza por su configuración doble canal, alta velocidad de conmutación y tiempos de respuesta. Para la selección del *switch* se tuvieron en cuenta factores de diseño como la alimentación sencilla a +5V y el ancho de banda. El esquema utilizado para la aplicación del multiplexor analógico es mostrado en la figura 4.10:

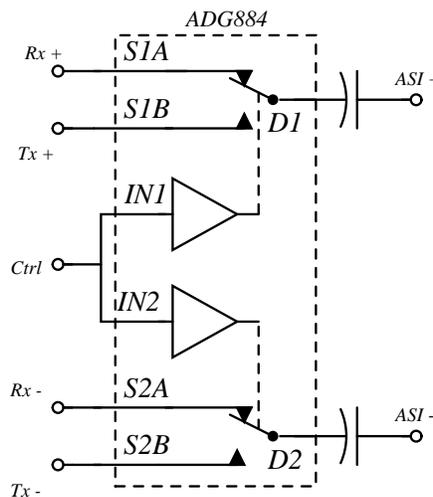


Figura 4.10: Esquema de aplicación del multiplexor analógico usado como *switch*.

### 4.5.2. Fuente de Alimentación

Una de las principales ventajas que ofrece el estándar AS-i, es el uso de un canal compartido para la transmisión de datos y potencia, simultáneamente. Por tal razón se debe extraer del bus la energía necesaria para el funcionamiento del procesador de comunicación. Para este fin se realizó un adecuado filtrado de señal, tanto en corriente como en voltaje con circuitos LC [25], de tal forma que se lograra obtener una señal regulada para la alimentación de los componentes internos. Esta señal de continua previamente filtrada, es llevada a una etapa de regulación variable que reduce el nivel de voltaje a 5V, valor definido para la alimentación de cada uno de los subsistemas y el cual se consideró como requerimiento de selección para los integrados utilizados. El diagrama circuital mostrado en la figura 4.11 representa el esquema de alimentación positivo del procesador.

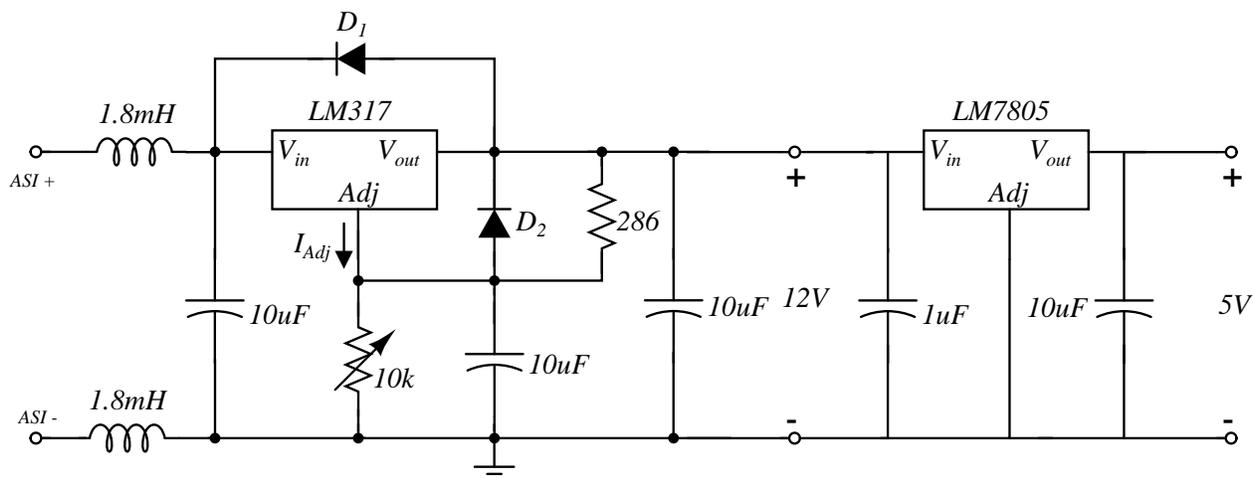


Figura 4.11: Diagrama circuital de la fuente de alimentación positiva.

La referencia negativa se implementó con el circuito integrado *LT1054*, el cual genera una fuente con capacidad de manejo de  $100mA$  a partir de la referencia positiva establecida con el *LM7805* [26]. El diagrama circuital mostrado en la figura 4.12 representa el esquema de alimentación negativa.

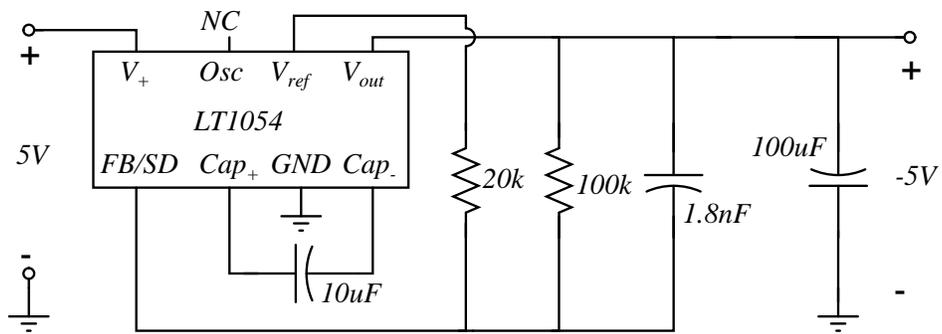


Figura 4.12: Diagrama circuital de la fuente de alimentación negativa.

# Capítulo 5

## Implementación del Software

En este capítulo se describe la implementación de las capas superiores del protocolo AS-i, incluyendo la subcapa SE descrita en la sección 3.2, donde fueron implementados los procesos de codificación *Manchester II* y modulación en corriente *APM*, con el objetivo de definir las señales analógicas expuestas en la sección 3.4, para garantizar el proceso de comunicación a través de las funciones definidas por la capa de aplicación. El código diseñado fue desarrollado en lenguaje de alto nivel, aprovechando las ventajas de programación ofrecidas por el compilador *Codewarrior*, sin embargo se resalta el hecho de implementar líneas de código en *Assembler*, con el propósito de optimizar los tiempos de ejecución de los algoritmos [27], [28].

### 5.1. Implementación de protocolos en microcontrolador

Las comunicaciones digitales justifican su evolución dentro del avance de la *Teoría de la Información*, la cual cuantifica los límites teóricos de un sistema de comunicación, y la *Microelectrónica*, la cual ha permitido el desarrollo de dispositivos con capacidad de implementar algoritmos complejos de forma económica. Tradicionalmente, la ingeniería de desarrollo en el área de la *Informática Industrial*, se orienta al espacio de diseño comprendido por las dimensiones de potencia y ancho de banda; sin embargo el desarrollo *microelectrónico* ha permitido la inclusión de la complejidad de procesamiento como una tercera dimensión de diseño, debido a la necesidad de llevar los sistemas a límites cercanos a los teóricos e implementar de forma eficiente, complejos algoritmos de procesamiento digital en *chips* que poseen características óptimas de programación, depuración, consumo y capacidad de cálculo.

En términos generales, existen metodologías de implementación de procesadores de comunicación, entre las cuales se destacan la forma *Compacta*, donde el protocolo y las tareas de aplicación se implementan en el mismo microcontrolador o la forma *Independiente*, donde se usa un

microcontrolador dedicado a la ejecución de las tareas de comunicación, sin ningún tipo de restricción y separando las funciones de aplicación [29]. Teniendo en cuenta estas alternativas, este documento describe la implementación independiente del procesador de comunicaciones AS-i, para su funcionamiento como dispositivo esclavo dentro de una red convencional, utilizando el microcontrolador descrito en la sección 4.1 [30], [31].

## 5.2. Maestro AS-Interface

La unidad maestra, es un sistema con un comportamiento lógico y funcional bien definido, que organiza y monitorea la red, programando el intercambio de datos, parámetros y/o comandos con un dispositivo esclavo a través de la línea AS-i. Esta unidad resultó de la integración de componentes *Hardware* y *Software*, a través de los cuales se maneja la fase de intercambio de datos. El *Hardware* implica los avances en el diseño de la interfaz universal presentados en el capítulo 4, y respecto al *Software* se presenta la funcionalidad del microcontrolador, para implementar las funciones lógicas que cumplan parcialmente con la máquina de estados mostrada en la figura 5.1.

La funcionalidad del maestro se resume en seis fases individuales, las cuales son: *Offline*, *Detección*, *Activación*, *Intercambio de Datos*, *Administración* e *Inclusión*, las cuales son ejecutadas a través de dos funciones específicas:

- *Execution Control Function*: Responsable de la inicialización del sistema AS-i, del intercambio cíclico de datos y acíclico de comandos, y de la ejecución de funciones de control.
- *Transmisión Control Function*: Encargada de manejar la transmisión física de las peticiones y la retransmisión automática en caso de falla [6].

## 5.3. Esclavo AS-Interface

El esclavo AS-i es un sistema que proporciona todos los medios físicos y lógicos para conectar la aplicación (actuador y/o sensor) a la línea AS-i, de tal forma que se habilite la recepción y envío de mensajes sobre una línea de distribución de potencia. Cada esclavo debe tener una descripción de la configuración I/O, el código de identificación y opcionalmente los códigos de identificación extendida (ID1 e ID2), los cuales pueden ser leídos por el maestro. La combinación de la configuración I/O y los códigos ID e ID2 definen el *Perfil del Esclavo*, de acuerdo a la siguiente expresión:

$$S - [I/O - code].[ID - code].[ID2 - code] \quad (5.1)$$

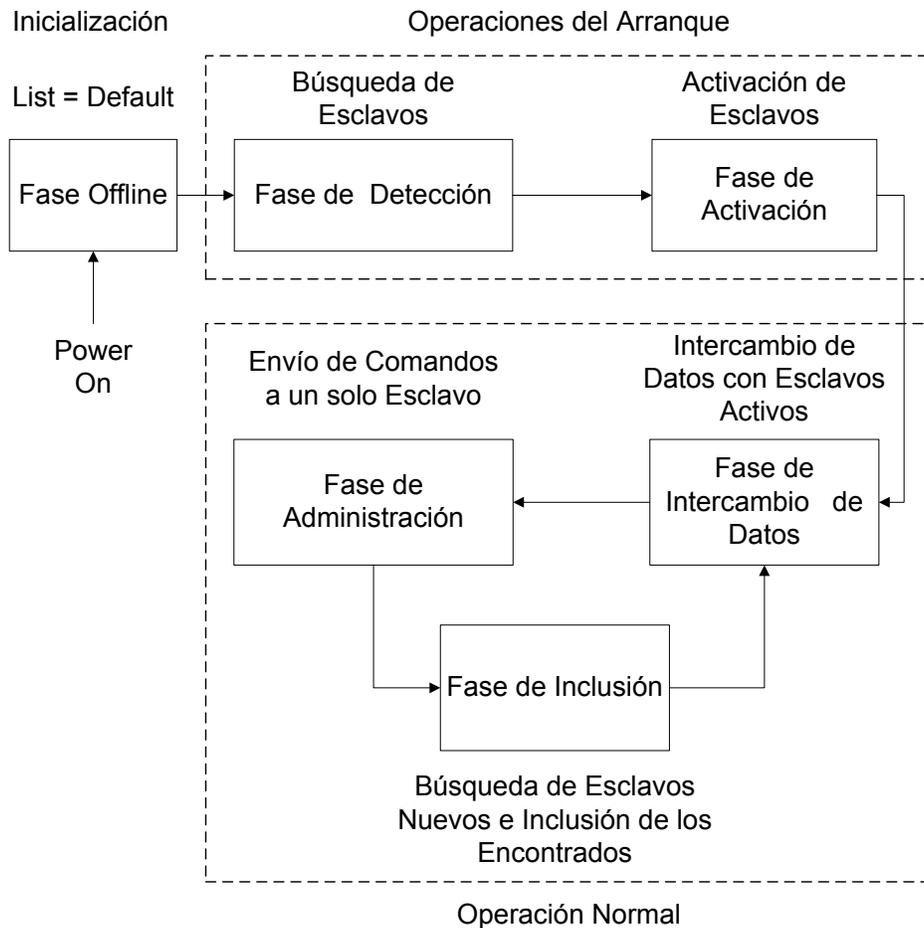


Figura 5.1: Máquina de estados del maestro AS-i.

Siguiendo la referencia expuesta en la ecuación 5.1, el esclavo objeto de esta investigación corresponde al perfil  $S - 7.3.C$ . Esta asignación define valores fijos para la configuración  $I/O$  y el código de identificación  $ID$ ; el código  $ID2$  permite manejar información adicional importante para la caracterización del dispositivo esclavo de acuerdo con la tabla 3.7. De esta tabla se puede concluir que el  $ID2$  que describe el perfil del procesador propuesto es  $1100b$  o  $0Ch$ , el cual corresponde a la configuración del módulo como un esclavo que posee un solo canal de entrada, habilitado para la transferencia de valores analógicos.

El esclavo maneja una serie de registros y banderas para su funcionamiento, entre los cuales se pueden destacar:

- *Address*: contiene la dirección de operación.
- *Data\_Output*: contiene la última información de salida transmitida a través de la petición *Data\_exchange*.
- *Data\_Input*: Registro Opcional, que contiene información de entrada en el modo síncrono.

- *Receive*: contiene la última petición de maestro recibida, omitiendo los *bits* de inicio y parada.
- *Transmit*: contiene la última respuesta de esclavo.
- *Status*: contiene 4-*Bits* independientes de información acerca del estado del esclavo.
- *Data\_Exchange\_Disable\_flag*: bandera seteada por un *RESET* y reseteada por la primera petición *Write\_Parameter* recibida, indicando la inhabilidad para intercambiar datos.

En la figura 5.2 se muestra la máquina de estados que debe ejecutar el esclavo AS-i, realizando una breve descripción de sus actividades particulares [6].

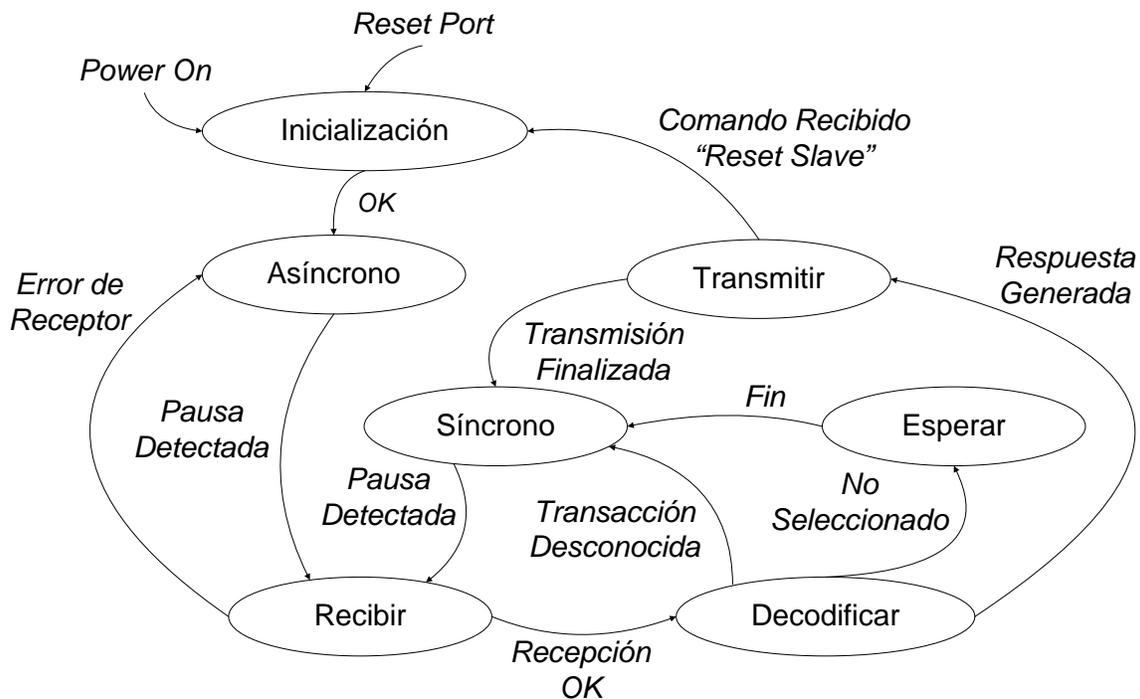


Figura 5.2: Diagrama de estados del esclavo AS-i.

- *Inicialización*: Estado inicial del esclavo después del encendido de la fuente AS-i o de un *reset* del mismo. Esta etapa carga los registros *Data\_Output* y *Parameter\_Output* con el valor por defecto *0Fh*, mueve su contenido a los respectivos puertos, *resetea* el registro *Status*, carga la dirección, configuración I/O y códigos ID a sus respectivos registros, y *setea* la bandera *Data\_Exchange\_Disable*.
- *Asíncrono*: Esta fase *resetea* la bandera *Sync*, registra el arreglo de datos de entrada y detecta la pausa. Si la pausa es detectada, se ejecuta la fase de *Recepción*.

- *Recepción*: Esta función espera por el *bit* de inicio, carga la petición de maestro en el registro *Receive*, ejecuta las rutinas de chequeo de errores y verifica la pausa del maestro. Si un error es detectado retorna a la fase anterior, de lo contrario ejecuta la etapa de *Decodificación*.
- *Decodificación*: Esta etapa compara la dirección recibida con el registro *address* (si no es igual cambia a la fase *Espera*), analiza la información (si es una petición desconocida cambia a la fase *Síncrona*), si la bandera *data\_exchange\_disable* es *seteada* y el maestro realiza un intercambio de datos cambia al estado *Síncrono*, ejecuta la tarea pedida, carga la respuesta en el registro *transmit* y ejecuta la fase *Transmisión*.
- *Espera*: Este estado espera por una señal detectable; si no existe, espera por el fin del tiempo de respuesta (ventana entre  $7T_{BIT}$  y  $9T_{BIT}$ ); en caso contrario, espera el fin de la respuesta del esclavo y cambia a la función *Síncrono*.
- *Transmisión*: Esta etapa espera por el final de la pausa del maestro (mínimo  $2T_{BIT}$ , si *Sync\_flag=set*; máximo  $5T_{BIT}$ , si *Sync\_flag=reset*) y envía la respuesta del esclavo. Si se recibe un *Reset\_slave*, se cambia a la fase *Inicialización*.
- *Síncrono*: Esta fase *setea* la bandera *Sync\_flag*, encuesta los datos de entrada y detecta la pausa. Si la pausa es detectada, cambia al estado de *Recepción*.

## 5.4. Implementación

La evolución de las redes industriales y, el desarrollo de microcontroladores y microprocesadores con mayor nivel de integración, permiten implementar unidades con mayor capacidad de manipulación de datos a menor costo. Estos microprocesadores incorporan usualmente protocolos de comunicación que reducen el programa de aplicación, para un rápido establecimiento de la red, y cuentan con alta capacidad de procesamiento (almacenamiento) para implementar dispositivos inteligentes con funciones de medida, aplicación o control, y capacidad para cerrar lazos de realimentación o comunicación a través de la red industrial. Con este trabajo se identificó una metodología para alcanzar esta funcionalidad y una arquitectura básica para desarrollar un dispositivo genérico microcontrolado, que permita implementar cualquier perfil funcional con un alto grado de flexibilidad, a través de un sencillo cambio de programa.

Los alcances que se exponen en este capítulo incluyen la implementación lógica completa de un esclavo *AS-Interface* con su respectiva aplicación maestra simplificada, de tal forma que se habilite la verificación y validación del proceso de comunicación definido por el estándar IEC 62026-2, a través del uso de la interfaz *hardware* desarrollada. Para esta implementación se tuvo en cuenta

una organización modular, garantizando la descripción funcional de las fases implicadas en cada perfil funcional, con base en las máquinas de estados mostradas en las figuras 5.1 y 5.2.

Los algoritmos desarrollados corresponden al perfil maestro  $M \cdot 3$  y el perfil esclavo  $S - 7 \cdot 3$  [12], los cuales fueron diseñados para conectar sensores analógicos al bus AS-i a través del manejo de transacciones combinadas tipo I. La implementación parcial de la aplicación maestra se orienta hacia la ejecución de las pruebas de validación del procesador, donde se sintetiza la fase de intercambio de datos, complementada con un proceso de reconocimiento previo a través de transacciones sencillas *I/O Configuration*, *ID Code*, *ID2 Code* y *Write Parameter*. Este esquema se extrajo de observar el tráfico de la red durante el proceso de reconocimiento del procesador AS-i por parte del maestro comercial  $CP - 342$  de *Siemens*, ya que por problemas de detección que impedian una decodificación 100% fiable, se abortaba el proceso de activación del esclavo que habilita la fase de intercambio de datos [6].

Los resultados obtenidos de esta prueba se resumen en la tabla 5.1, donde se resalta que la arquitectura de red utilizada incluye el procesador AS-i desarrollado, identificado con la dirección 4 y un módulo esclavo comercial con dirección 2.

Con base en estos resultados y los esquemas que propone la norma para describir el comportamiento del maestro, se elaboró e implementó el diagrama de flujo mostrado en la figura 5.3, el cual representa la implementación parcial propuesta inicialmente para la validación del procesador, donde se sintetizan las fases *offline*, detección, activación e intercambio de datos.

Respecto a la filosofía de programación, inicialmente se evaluó un desarrollo con base en la programación de alto nivel<sup>1</sup> que ofrece la herramienta *Codewarrior* [32]; sin embargo pruebas preliminares permitieron concluir que aún exigiendo la operación del microcontrolador a su máxima frecuencia de bus ( $25MHz$ ), no era posible alcanzar algunos requerimientos de tiempo que exige el protocolo, como por ejemplo el control de las pausas y el manejo de las transacciones, expuestos en la sección 3.5. Por tal razón fue necesario recurrir a la programación en código *assembler* [33], [34], de tal forma que se explotara de forma eficiente la arquitectura de *16-Bits* del *core* y así optimizar los tiempos de ejecución de algunas rutinas críticas como son la generación de los patrones analógicos y la reconstrucción del mensaje, con su respectiva identificación de los diferentes tipos de errores, presentados en la sección 3.9.

## 5.5. Generación de Patrones Analógicos

En esta sección se abordaran los resultados de las pruebas realizadas para caracterizar el proceso de generación de los patrones analógicos de corriente, que representan el mensaje AS-i. Para evaluar

---

<sup>1</sup> Lenguaje C.

Tabla 5.1: Monitoreo del sistema AS-i.

Mensaje	Maestro	Esclavo	Comentario
1	Data_Exchange	2	Operación Normal
2	I/O Configuration	4	Inicia el proceso de reconocimiento del esclavo 4
3	Data_Exchange	2	Operación Normal
4	ID Code	4	Continúa el proceso de reconocimiento sin problemas
5	Data_Exchange	2	Operación Normal
6	Write_Parameter	4	Intenta realizar el Write_Parameter, pero por problemas de detección el esclavo no la reconoce
7	Write_Parameter	4	Intenta una retransmisión del comando después de $75\mu s$ , hasta lograr la respuesta del esclavo
8	Data_Exchange	2	Operación Normal
9	Data_Exchange	4	Termina el proceso de reconocimiento del esclavo, donde el maestro lo declara como activo e intenta establecer el intercambio de datos, pero por problemas en la recepción el esclavo no la reconoce
10	Data_Exchange	4	Nuevamente intenta una retransmisión del comando de intercambio después de $75\mu s$ , pero definitivamente el esclavo no la reconoce, lo que conlleva a que el maestro aborte el proceso y reintente una vez más
11	Data_Exchange	2	Operación Normal

el desempeño de la solución propuesta se implementó una aplicación en *Matlab* que permitiera visualizar las diferentes aproximaciones logradas y calcular el error cuadrático medio como criterio de validación de su precisión [35]. De esta forma se buscaba un equilibrio entre la precisión requerida y la exigencia del *driver* de línea.

Vale la pena resaltar que este proceso de generación de la onda analógica fue desarrollado completamente en *assembler*, debido a que con la programación en alto nivel no se alcanzaron los tiempos de actualización de los registros, que requería la aplicación para generar una onda lo más cercana posible al comportamiento ideal expuesto en la sección 3.4 [6].

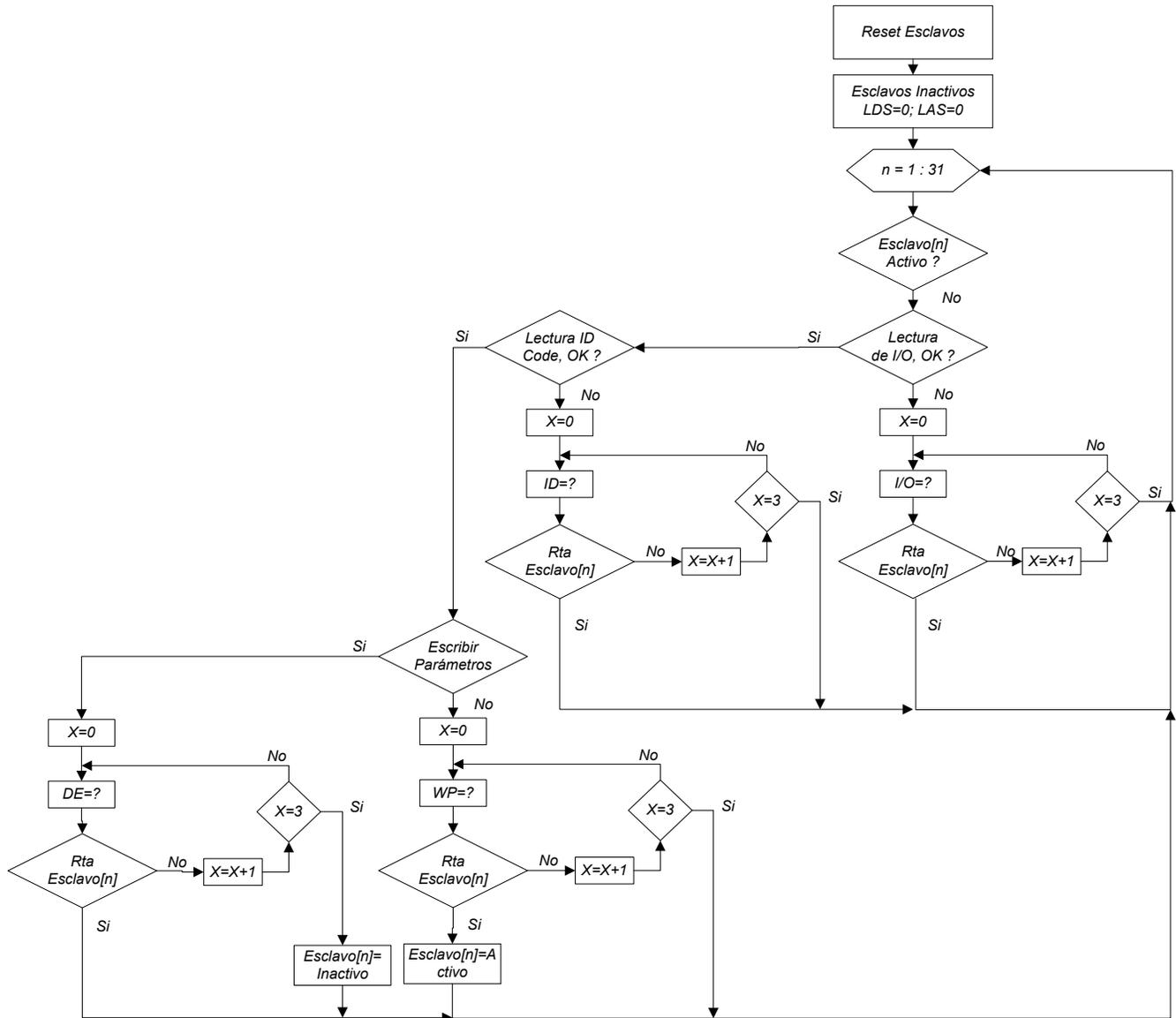


Figura 5.3: Diagrama de flujo de la aplicación maestra.

### 5.5.1. Representación de 24 muestras

Inicialmente se realizaron pruebas de generación con el propósito de evaluar el verdadero desempeño del módulo DAC del micro, ya que según el fabricante este tenía un tiempo crítico de establecimiento de  $5\mu s$ . En la figura 5.4 se muestra la primera aproximación con 24 muestras, aprovechando el *full range* del convertor. Utilizar el rango de operación completo del módulo, implicaba incrementar la exigencia del *driver* desde el punto de vista de *slew rate* y niveles de saturación, afectando considerablemente el rango de operación de la fuente *Howland*. Por otra parte reducir el número de niveles de cuantización para representar la señal implicaba perder la resolución y por ende la precisión de la señal. Esta primera aproximación representa la máxima ve-

locidad de actualización alcanzable aprovechando teóricamente los 255 de cuantización que maneja el modulo DAC de 8-Bits, sin embargo esta situación no se pudo alcanzar experimentalmente.

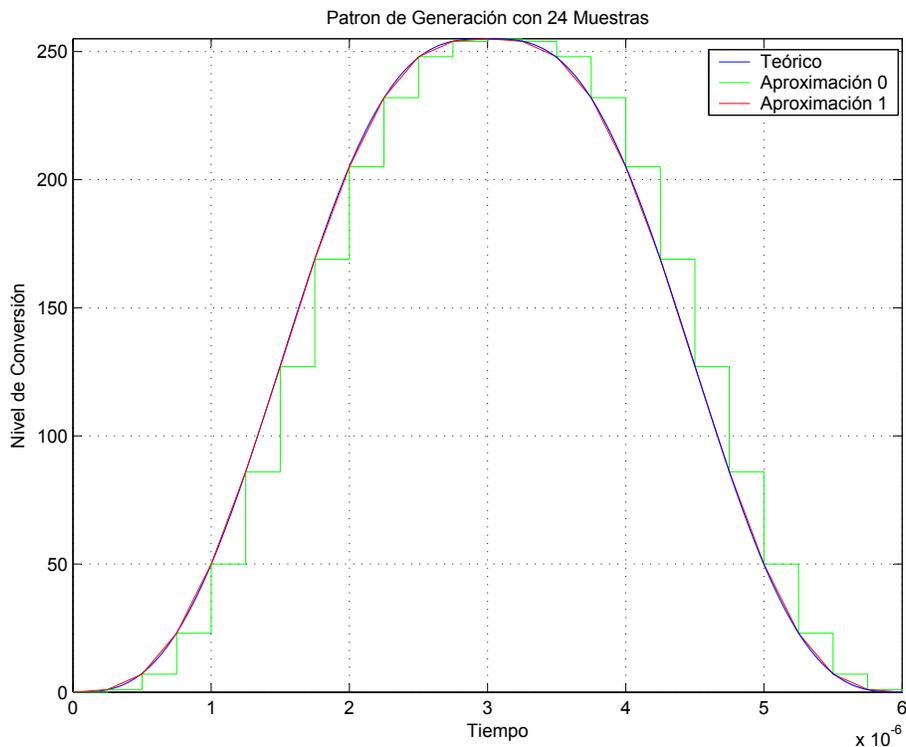


Figura 5.4: Representación de los patrones generados con 24 muestras.

### 5.5.2. Representación de 20 muestras

Posteriormente, se optó por sacrificar los niveles de cuantización con el objetivo de maximizar la frecuencia de actualización del convertor y demandar la máxima frecuencia de operación del bus interno. En otras palabras se consideró prioritario la velocidad de procesamiento por encima de la precisión, sin embargo como se verá más adelante el resultado final expone un error cuadrático medio bastante bajo para la aproximación final, comparado con la ecuación determinística que modela la señal de corriente en el bus. En la figura 5.5 se muestra la aproximación para 20 muestras, teniendo en cuenta la velocidad de respuesta mostrada por el módulo periférico. Este resultado utiliza 102 niveles de cuantización con un tiempo de actualización de  $300ns$  por muestra.

### 5.5.3. Representación de 30 muestras

Finalmente se realizaron pruebas ajustando la velocidad de respuesta mostrada por el módulo DAC, lo cual converge a la solución definitiva de 30 muestras, donde se concluye en la optimización

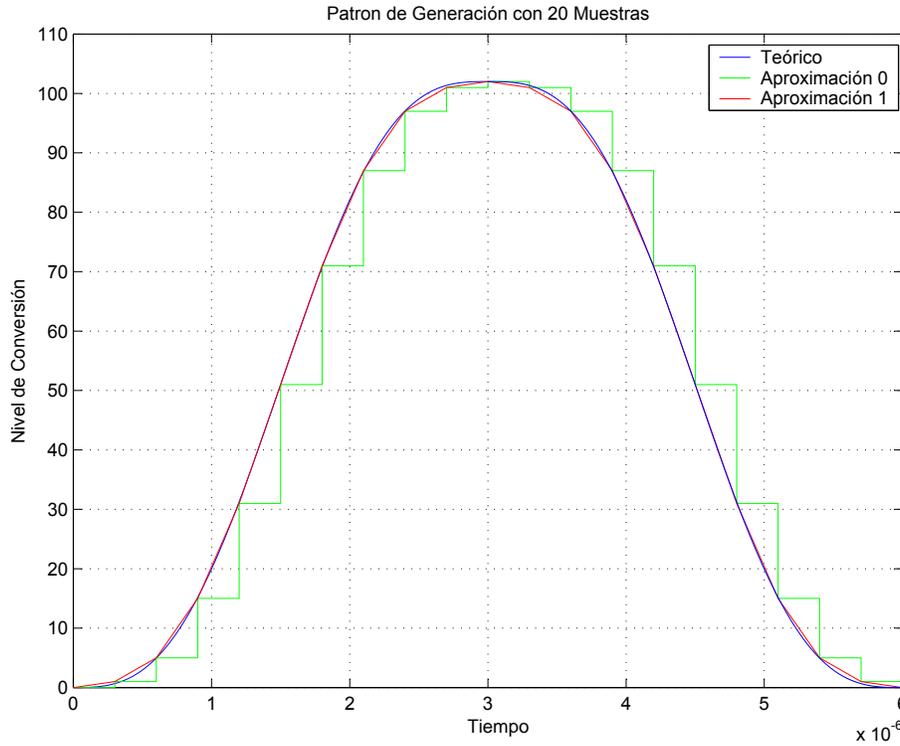


Figura 5.5: Representación de los patrones generados con 20 muestras.

del algoritmo de generación para extraer la velocidad de actualización, la cual corresponde a  $200ns$  por muestra. El resultado de la aproximación que hace uso de 102 niveles de cuantización se muestra en la figura 5.6. En conclusión el proceso termina con la generación de una onda de voltage de  $2V_{P-P}$  que no eleva los requerimientos del operacional utilizado en la implementación del *driver*.

#### 5.5.4. Tiempo de respuesta del modulo DAC

El módulo *DAC* está configurado para trabajar en un escala de valores *full range* de 0 a 255 para una resolución de 8-Bits, basado en un referencia teórica de  $5V$ . Sin embargo se pudo verificar experimentalmente que dicha referencia de voltaje obedece a  $4.5V$ , lo cual conllevó a realizar un recalcu aproximado de los valores de las muestras para ubicar la señal analógica en este rango. Realizando este ajuste se obtuvieron los siguientes valores: 0, 0, 2, 5, 11, 20, 31, 44, 58, 71, 82, 91, 97, 100, 102 y 102; los cuales corresponden a la reconstrucción de la mitad de la onda simétrica generada para la representación final de 30 muestras por  $T_{BIT}$ . La justificación del nivel de amplitud de salida, se centra basicamente en el equilibrio de minimizar los requerimientos del *driver*, maximizando la capacidad de respuesta del DAC. Para evaluar dicha capacidad se generó una prueba en la cual se tomaron mediciones para diferentes cambios de nivel entre muestras, las cuales se presentan en la tabla 5.2.

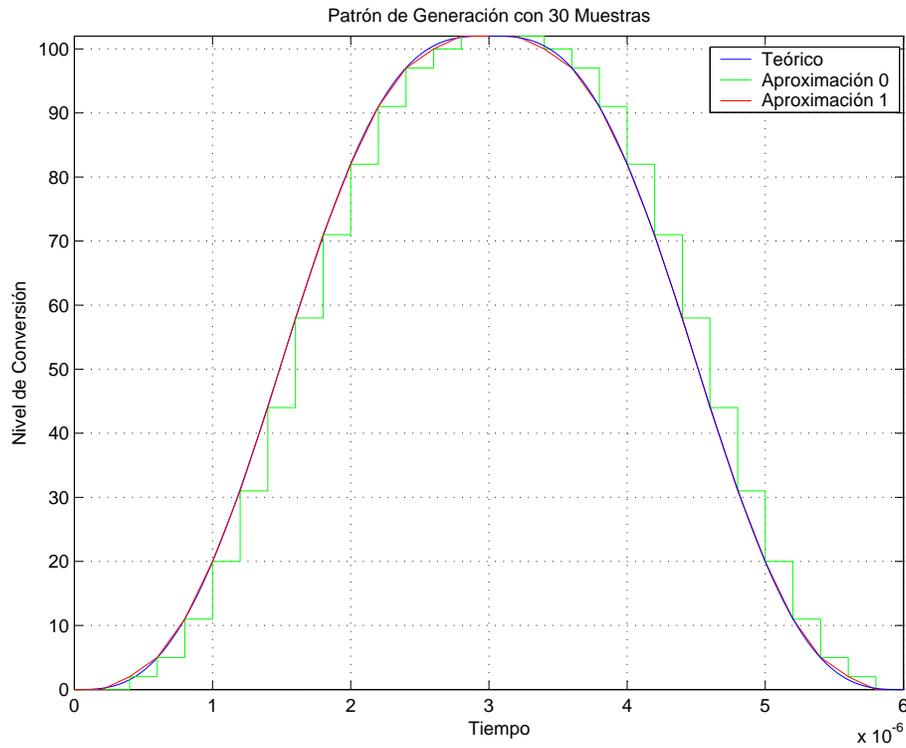


Figura 5.6: Representación de los patrones generados con 30 muestras.

Tabla 5.2: Prueba de tiempos de respuesta del módulo DAC.

$\Delta Voltage$ [V]	$\Delta Tiempo$ [ns]
0.56	280
0.32	160
1	520
0.36	200
0.88	440
0.24	120
0.88	480
1	520
0.36	160

Según estas pruebas, se puede observar que el módulo DAC está en capacidad de hacer un cambio máximo de 1V en 520nSeg. Por tal razón para la frecuencia de operación del procesador a 25Mhz, se permite el siguiente cambio máximo:

$$\Delta V_{Máximo} = \left( \frac{1}{520ns} \right) \left( \frac{T_{BIT}}{\#Muestras} \right) = 0.3846 \quad (5.2)$$

Este resultado indica que la máxima variación de niveles es:

$$\Delta_{Máximo} = \Delta V_{Máximo} \left( \frac{255}{5} \right) = 19.61 \quad (5.3)$$

Con la solución propuesta se da un margen de error de cinco niveles puesto que la máxima diferencia cada  $200ns$  es de 14 niveles.

### 5.5.5. Evaluación de las aproximaciones

Con el objeto de evaluar las aproximaciones anteriormente realizadas de una forma cuantitativa, se puede recurrir al cálculo de indicadores estadísticos simples como el *Error Cuadrático Medio*, el *Error Absoluto Medio* o el *Coefficiente de Sesgo*. En esta sección se optó por utilizar el EMC (*Error Cuadrático Medio*) [36], para estimar la precisión de la onda generada, la cual se define como: "El grado de correspondencia entre pares individuales de valores teóricos y valores aproximados" [35].

El EMC está definido por la siguiente expresión:

$$EMC = \sqrt{\sum_{i=1}^N \frac{[V_{I(Teórico)} - V_{I(Aprox)}]^2}{N}} \quad (5.4)$$

Con base en estas definiciones y teniendo en cuenta los resultados mostrados en las figuras 5.5, 5.4 y 5.6, se presentan en la tabla 5.3 los errores estimados para cada una de las opciones de generación:

Tabla 5.3: Resultados comparativos en el proceso de generación.

Muestras	Niveles	EMC
20	102	0.4213
24	255	0.6638
30	102	0.2661

Estos resultados reflejan la validez del proceso de generación detallado en esta sección. Inicialmente se planteó una solución de 20 muestras con un error de referencia de 0.4213, la cual no utilizaba el rango completo del DAC. Como el objetivo era maximizar la frecuencia de actualización de los registros del módulo, aprovechando el rango de operación *full scale* del mismo, se mejoró el algoritmo hasta 24 muestras con 255 niveles, donde se generó un aumento del 57.56% en el error comparado con la solución de 20 muestras, debido a la lenta capacidad de respuesta del DAC.

De esta forma, manteniendo los mismos criterios de optimización, finalmente se maximizó el número de muestras por  $T_{BIT}$  a 30, para lograr una representación más fiable de la onda de corriente, aplicando rutinas de programación en *Assembler*. Al llegar a este punto, solo faltaba por

definir el número máximo de niveles permitido por los tiempos de establecimiento del convertor para generar cambios válidos de nivel, el cual correspondió a 102. Esta solución final sustentada en la metodología presentada, muestra una reducción considerable del EMC, en un 36.83% y 59.91%, comparado con las soluciones de 20 y 24 muestras, respectivamente.

De acuerdo a los resultados anteriormente presentados, se logró obtener un patrón base equivalente a un tiempo de bit, como el mostrado en la figura 5.7

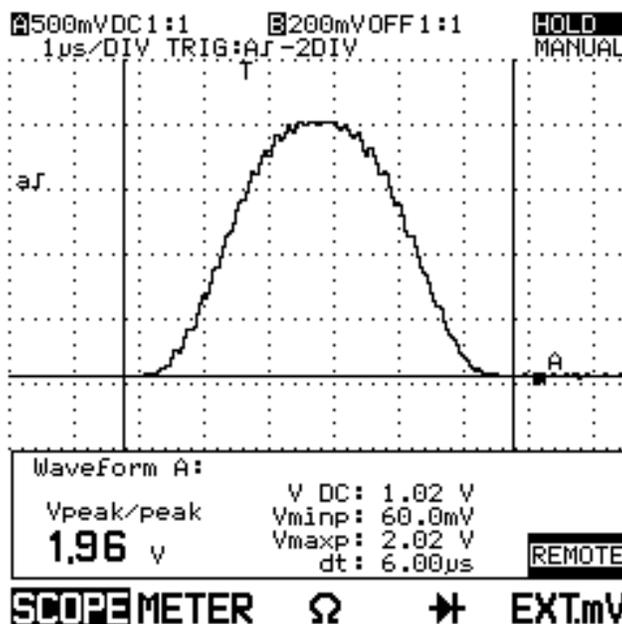


Figura 5.7: Representación analógica final para un  $T_{BIT}$ .

## 5.6. Funciones Implementadas

Como parte del desarrollo de la capa de aplicación del protocolo AS-i, en esta sección se presentan las funciones implementadas en *software* para el control de la comunicación, orientadas principalmente a los procesos de reconocimiento, activación e intercambio de datos. Por tal motivo se presentan las respectivas estructuras de datos (datagramas), tanto del maestro como del esclavo, que deben identificar los receptores de los componentes de red, para decodificar dichas funciones.

### 5.6.1. Reset Slave

Esta función es una petición tipo comando ( $I4 = 1$ ), implementada durante la fase *Offline* para realizar un *Reset* aun esclavo específico. Para tal efecto se elabora una estructura de datos como

la mostrada en la tabla 5.4, de tal forma que el esclavo responda con un arreglo de *7-bits* como el mostrado en la tabla 5.5.

Tabla 5.4: Estructura de una petición de *Reset*.

ST	CB	Dirección					I4	Información	PB	EB
0	1	A4	A3	A2	A1	A0	1	1 1 0 0	PB	1

Tabla 5.5: Estructura de la respuesta del esclavo.

ST	Información	PB	EB
0	0 1 1 0	PB	1

### 5.6.2. Read I/O Configuration

Esta función es una petición tipo comando ( $I4 = 1$ ), usada por el maestro para obtener la configuración de entrada y salida del dispositivo esclavo. Para tal efecto se elabora una estructura de datos como la mostrada en la tabla 5.6, de tal forma que el esclavo responda con un arreglo de *4 bits*, de acuerdo a la información mostrada en la tabla 5.7.

Tabla 5.6: Estructura de una petición de lectura de configuración I/O

ST	CB	Dirección					I4	Información	PB	EB
0	1	A4	A3	A2	A1	A0	1	0 0 0 0	PB	1

### 5.6.3. Read ID Code

Esta función es una petición tipo comando ( $I4 = 1$ ), usada por el maestro para obtener el código de identificación de un esclavo específico. Para tal efecto se elabora una estructura de datos como la mostrada en la tabla 5.7. Si este código es *Ah*, el esclavo usa direccionamiento extendido y si es *Bh*, maneja señales de seguridad (*safety signals*).

### 5.6.4. Read ID2 Code

Esta función tipo comando ( $I4 = 1$ ), es usada para obtener información adicional de un esclavo específico, de acuerdo a la tabla 3.7.

Tabla 5.7: Opciones de configuración I/O para dispositivos esclavos

Código I/O <i>4-Bits</i>	Configuración I/O			
	D0	D1	D2	D3
0h	IN	IN	IN	IN
1h	IN	IN	IN	OUT
2h	IN	IN	IN	I/O
3h	IN	IN	OUT	OUT
4h	IN	IN	I/O	I/O
5h	IN	OUT	OUT	OUT
6h	I/O	I/O	I/O	I/O
7h	I/O	I/O	I/O	I/O
8h	OUT	OUT	OUT	OUT
9h	OUT	OUT	OUT	IN
Ah	OUT	OUT	OUT	I/O
Bh	OUT	OUT	IN	IN
Ch	OUT	OUT	I/O	I/O
Dh	OUT	IN	IN	IN
Eh	OUT	I/O	I/O	I/O
Fh	TRI	TRI	TRI	TRI

Tabla 5.8: Estructura de una petición de lectura del código de identificación

ST	CB	Dirección					I4	Información	PB	EB
0	1	A4	A3	A2	A1	A0	1	0 0 0 1	PB	1

Tabla 5.9: Estructura de una petición de lectura del ID2-Code

ST	CB	Dirección					I4	Información	PB	EB
0	1	A4	A3	A2	A1	A0	1	0 0 1 1	PB	1

### 5.6.5. Write Parameter

Esta función es usada por el maestro para escribir el registro de parámetros de salida y obtener información de los parámetros de entrada. Para tal efecto se elabora una estructura de datos como la mostrada en la tabla 5.10.

Tabla 5.10: Estructura de una petición de escritura de parámetros

<b>ST</b>	<b>CB</b>	<b>Dirección</b>					<b>I4</b>	<b>Información</b>				<b>PB</b>	<b>EB</b>
0	0	A4	A3	A2	A1	A0	1	P3	P2	P1	P0	PB	1

### 5.6.6. Data Exchange

Esta función es usada por el maestro para escribir el registro de datos de salida del esclavo y obtener información de sus entradas. Para tal fin se elabora una estructura de datos como la mostrada en la tabla ??.

Tabla 5.11: Estructura de una petición de intercambio de datos

<b>ST</b>	<b>CB</b>	<b>Dirección</b>					<b>I4</b>	<b>Información</b>				<b>PB</b>	<b>EB</b>
0	0	A4	A3	A2	A1	A0	0	D3	D2	D1	D0	PB	1

# Capítulo 6

## Pruebas y Resultados

En este capítulo se presentan los resultados finales de las pruebas realizadas a la arquitectura de red comercial y al procesador AS-i desarrollado, para evaluar su desempeño y facilitar el proceso de validación.

### 6.1. Pruebas sobre la arquitectura de red comercial

En esta sección se presentan algunos registros de las formas de onda obtenidas de la arquitectura de red comercial con que cuenta el Laboratorio de Mecatrónica, los cuales permiten evidenciar el cumplimiento de los requerimientos expuestos en el estandar IEC 62026-2 ???. Esta arquitectura de red proporcionada por la compañía *Siemens*, permitió asimilar y depurar el comportamiento del protocolo *AS-Interface*.

#### 6.1.1. Pruebas de corriente

Las figuras 6.1(a) y 6.1(b) muestran las formas de onda de la señal de corriente en el bus, sobre los puertos ASi+ y ASi-, capturadas como la caída de tensión en una resistencia de valor nominal  $10.2\Omega$ . De estas gráficas se resaltan los valores mostrados por el osciloscopio, donde se comprueba la frecuencia fundamental de trabajo de  $166kHz$  y el valor pico a pico de la señal de corriente de  $60.39mA$  ( $616mV/10.2\Omega$ ).

#### 6.1.2. Pruebas de voltage

En la figura 6.2 se muestra la forma de onda de la señal de voltaje en el bus AS-i, capturada diferencialmente a la salida del maestro de red. En está se observa el patrón analógico de voltaje de una petición de maestro, donde se resaltan valores como la frecuencia fundamental de

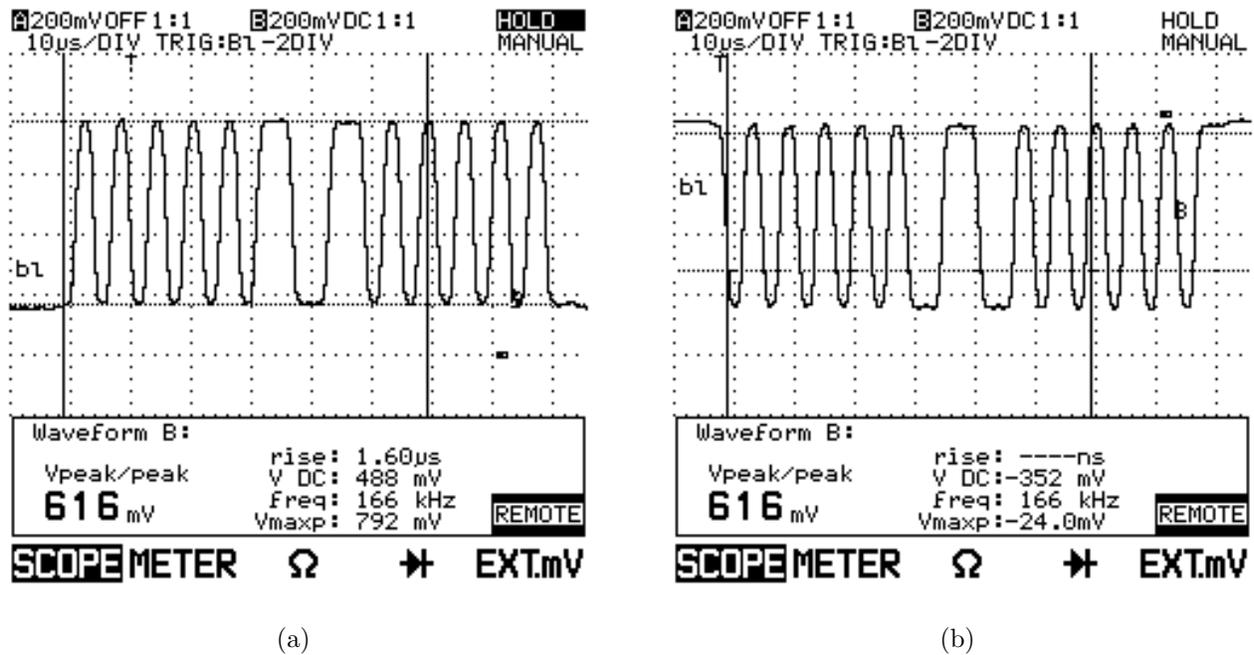


Figura 6.1: Formas de onda de corriente. (a) Puerto ASi+. (b) Puerto ASi-.

operación de  $166\text{kHz}$ , el valor máximo de  $5V_{P-P}$  y una ventana de tiempo de aproximadamente  $88\mu\text{s}$ , características que cumplen con las especificaciones definidas por la norma [6].

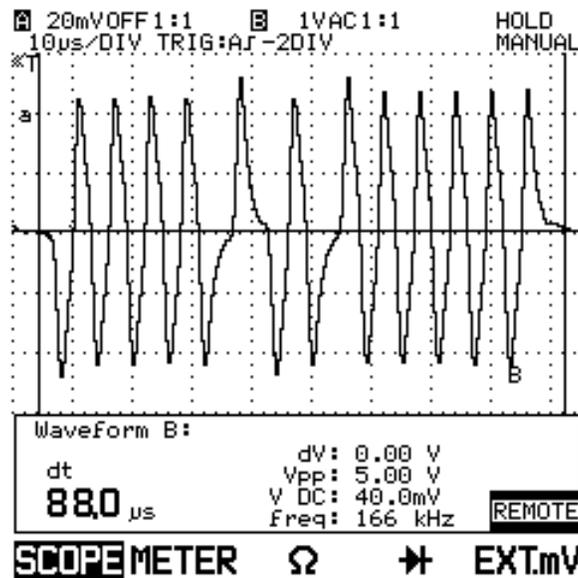


Figura 6.2: Forma de onda de voltage ASi.

## 6.2. Pruebas del Procesador AS-i

### 6.2.1. Pruebas de Transmisión

Las formas de onda de la figura 6.3(a) muestran los patrones de corriente generados como señal de voltage con el microcontrolador. El Canal *A* corresponde a la señal de salida del modulo DAC y el Canal *B* a la señal de entrada al *driver* de línea, después del circuito de acople (*buffer*). De esta gráfica se resaltan los valores mostrados por el osciloscopio, donde se comprueba el  $T_{BIT}$  efectivo generado de  $6\mu s$  y la tensión de  $2V_{P-P}$ , seleccionada para minimizar los requerimientos del *driver* de línea y maximizar los requerimientos de muestreo, teniendo en cuenta los tiempos de respuesta del modulo DAC. Durante estas pruebas se evaluaron diferentes algoritmos de generación, sin embargo con la solución final se optimizaron los recursos del microcontrolador, exigiendo su operación a una frecuencia de bus de  $25MHz$  para garantizar 30 muestras por  $T_{BIT}$ . El patrón analógico mostrado corresponde a la generación del arreglo binario  $0110001b$  modulado en corriente.

Esta onda de voltage es convertida en una señal de corriente de aproximadamente  $60mA$  a través del generador *Howland*, con el ajuste de un resistor externo (resistores  $R_5$  y  $R_{11}$  de la figura 4.7). Esta señal de corriente inyectada en el bus generó los patrones analógicos de voltaje mostrados en la figura 6.3(b), donde se sigue garantizando el  $T_{BIT}$  y los valores de amplitud exigidos por la norma [6].

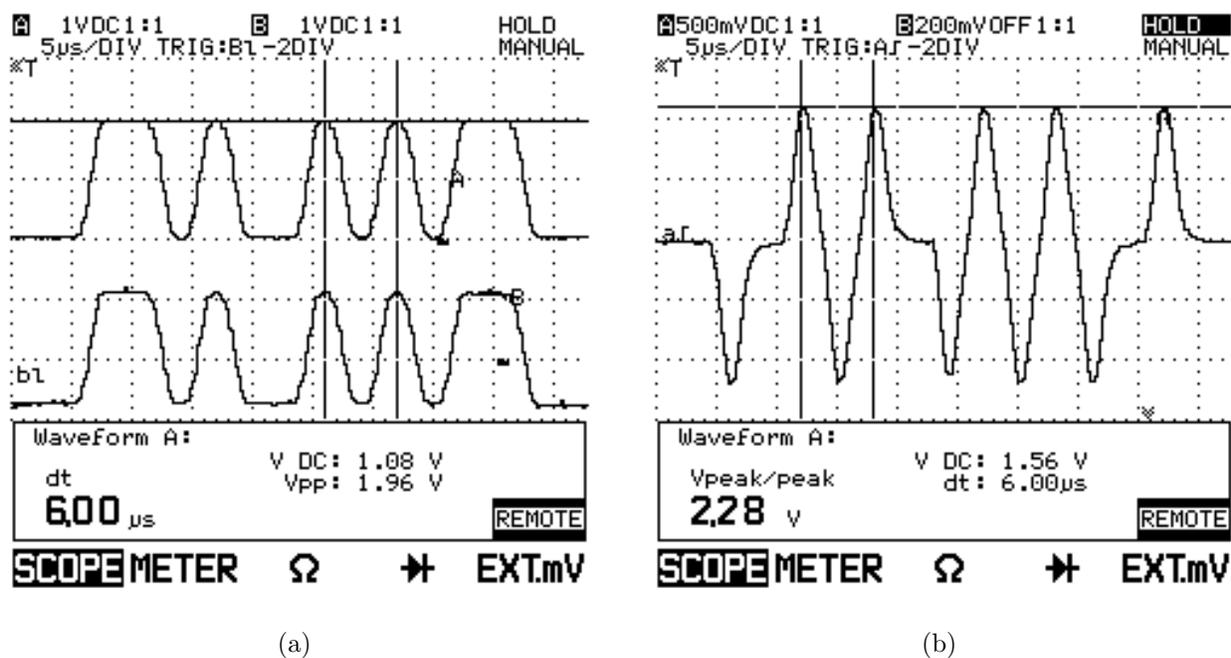


Figura 6.3: (a) Forma de onda de corriente generada. (b) Forma de onda de voltaje sobre el bus.

En la figura 6.4, se presentan los patrones de voltage y corriente ideales generados mediante

una aplicación en Matlab, de tal forma que se pueda establecer un marco de comparación con respecto a las formas de onda presentadas en la figura 6.3

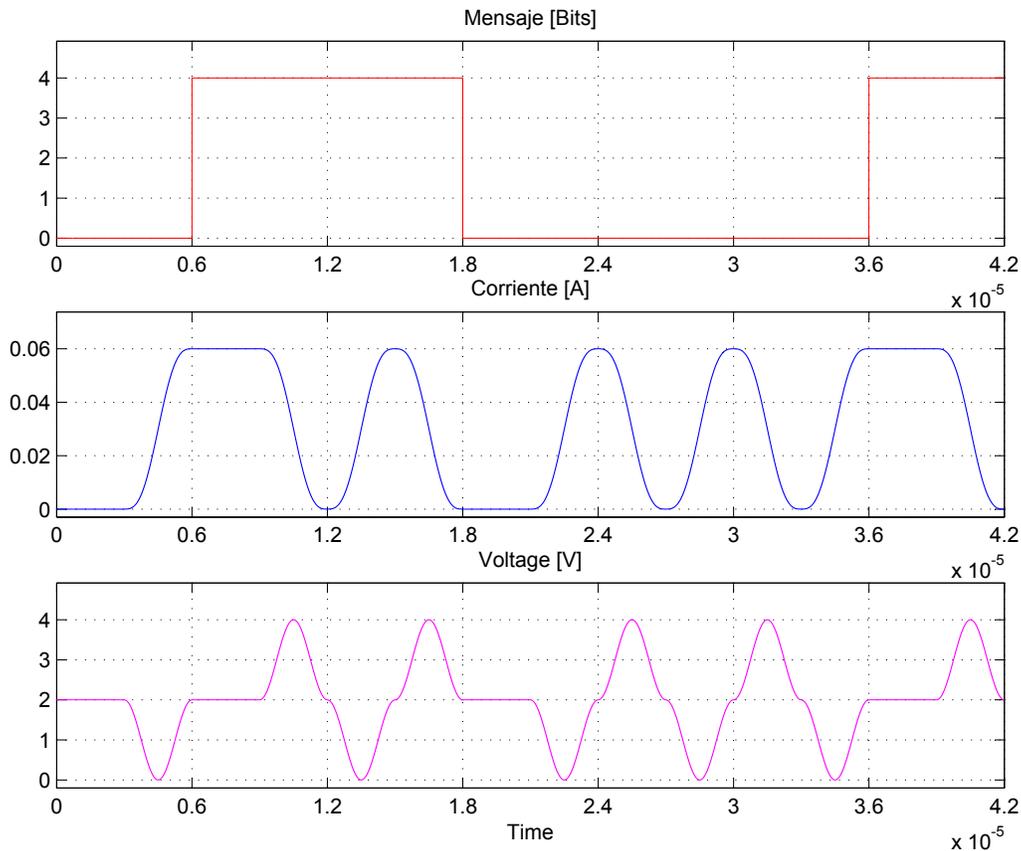


Figura 6.4: Patrones ideales del arreglo binario 0110001b.

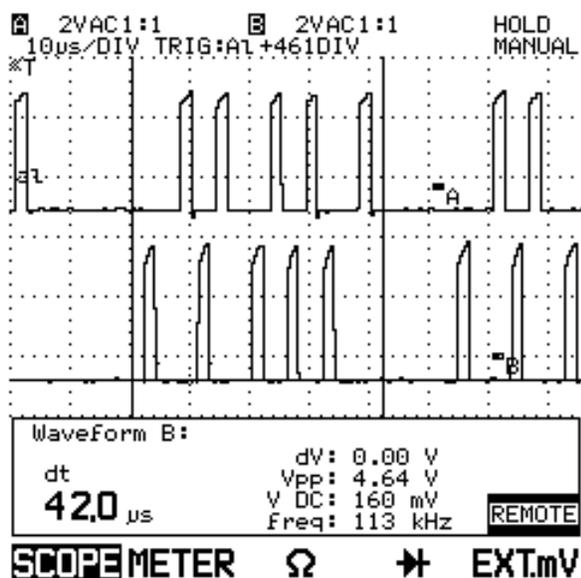
De esta figura es importante resaltar que los periodos de descanso reales no son tan cercanos al ideal, ya que la constante de tiempo que impone el circuito  $RL$  de simetría y desacople, impide un cambio fuerte en las condiciones de operación, presentando una curva de descarga lenta que no permite forzar un nivel cero de señal de forma instantánea. Sin embargo, se puede apreciar en los patrones extraídos de la red para un dispositivo comercial, mostrados en la figura 6.2, que dicho comportamiento es normal y aceptable, ya que no perturba el proceso de recepción, donde lo importante son los lapsos de tiempo existentes entre cada pulso detectado y no la forma de onda.

### 6.2.2. Pruebas de Recepción

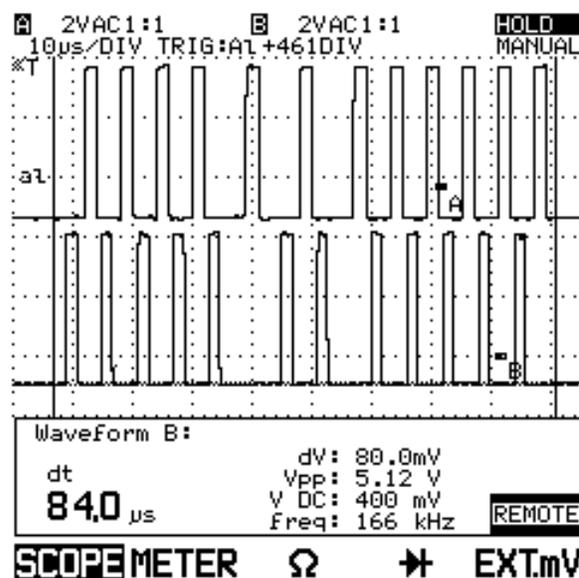
En la figura 6.5 se muestran las formas de onda de salida de los dos canales de recepción implementados por *hardware*, para la detección independiente de los pulsos positivos (Canal A) y negativos (Canal B), generados por el proceso de comparación de la señal de voltaje en el bus. La

figura 6.5(a) corresponde a los pulsos detectados para una respuesta de esclavo antes de la etapa de aislamiento digital, donde se resalta la adecuada alternancia de los pulsos, la ventana de tiempo de  $42\mu\text{s}$  para los 7-Bits de la trama, y el inicio y fin del mensaje con un pulso negativo y positivo, respectivamente.

Asimismo, en la figura 6.5(b) se muestran los pulsos positivos (Canal A) y negativos (Canal B) detectados para una petición de maestro después de la etapa de aislamiento, donde de igual forma se resalta la adecuada alternancia y polaridad de los pulsos de inicio y fin, y la ventana de tiempo de  $84\mu\text{s}$  para los 14-Bits de la trama. Adicionalmente, se destaca que el proceso de aislamiento digital de señal permite mejorar la forma de los pulsos.



(a)



(b)

Figura 6.5: (a) Pulsos detectados para una respuesta de esclavo antes de la etapa de aislamiento. (b) Pulsos detectados para una petición de maestro después de la etapa de aislamiento.

### 6.3. Pruebas Varias

Se hicieron pruebas adicionando y desconectando esclavos comerciales de la red para evaluar si había algún evento que generara la desconfiguración del procesador AS-i, sobre todo que se presentara una posible pérdida de sincronía.

De igual forma se realizó una desconexión intencional de la fuente AS-i con el objetivo de evaluar si el transitorio en el voltage de suministro podría llegar a afectar el nivel de alimentación

del procesador, el cual podría ocasionar un estado de *Reset* del microcontrolador, que llevara a perder su secuencia lógica.

Se mantuvo operando el procesador AS-i de forma autonoma durante un periodo aproximado de 4 horas sin presentar ningún tipo de funcionamiento anormal.

Se evaluó el procesador generando cambios aleatorios del registro de dirección en un rango entre 1 - 31, y este respondia de forma satisfactoria, logrando el reconocimiento por parte del maestro *CP342 – 2* de *Siemens* para el manejo de dispositivos lógicos, a través de las funciones de activación expuestas en la sección 5.5.

Realizando una prueba específica con el esclavo se cargo el registro *address* con la dirección dos. Para tal efecto en la figura 6.6 se muestran los patrones analógicos en tensión y corriente de la petición de maestro que decodifica el procesador AS-i para establecer el intercambio de datos a través de la función *Data\_Exchange*.

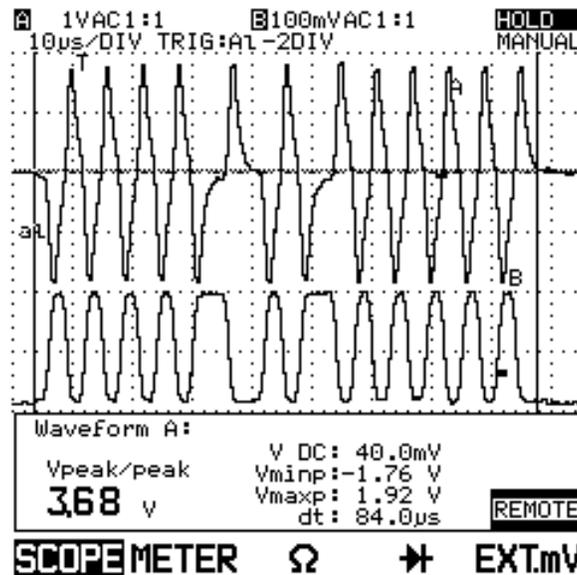


Figura 6.6: Petición *Data\_Exchange* del maestro al procesador AS-i configurado con la dirección dos.

Para efectos de comprobación en la figura 6.7, se reprodujo con ayuda de la aplicación en Matlab, el arreglo binario `00000100111111b` que debe decodificar el procesador AS-i, el cual corresponde efectivamente a una petición de intercambio de datos por parte del maestro, dirigido al esclavo identificado con la dirección dos.

Finalmente en la figura 6.8 se muestran los datagramas capturados en el momento que el procesador fue configurado con la dirección cinco, en la cual se puede observar la petición por parte del maestro para solicitar un *Data\_Exchange* y la respectiva respuesta generada por el procesador AS-i.

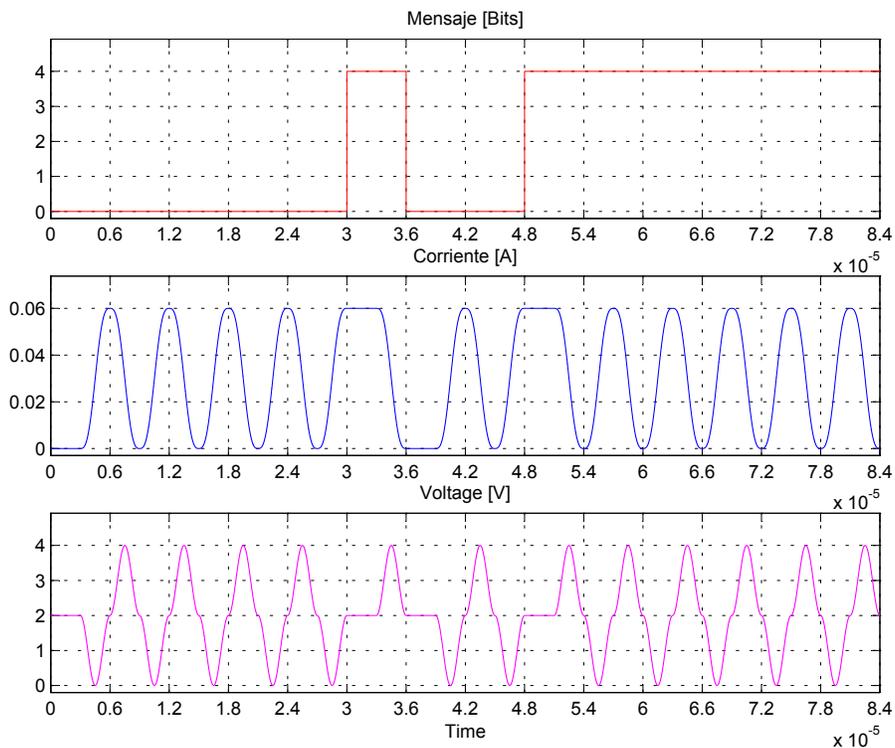


Figura 6.7: Patrones ideales para la petición *Data\_Exchange*, al esclavo identificado con dirección dos.

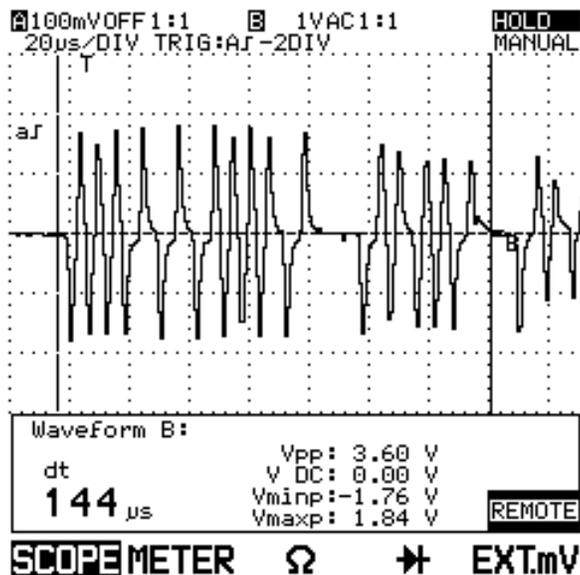
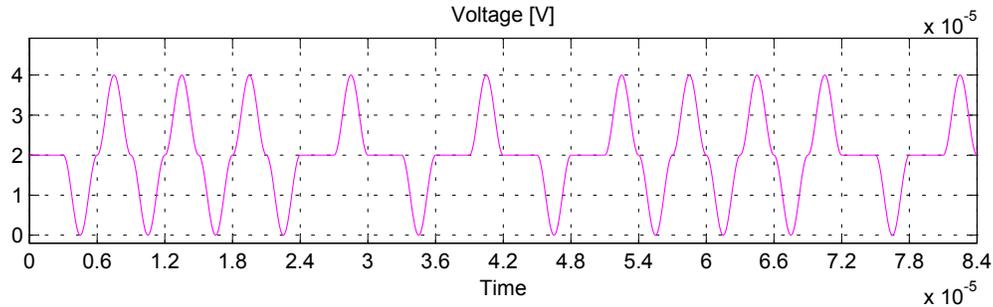


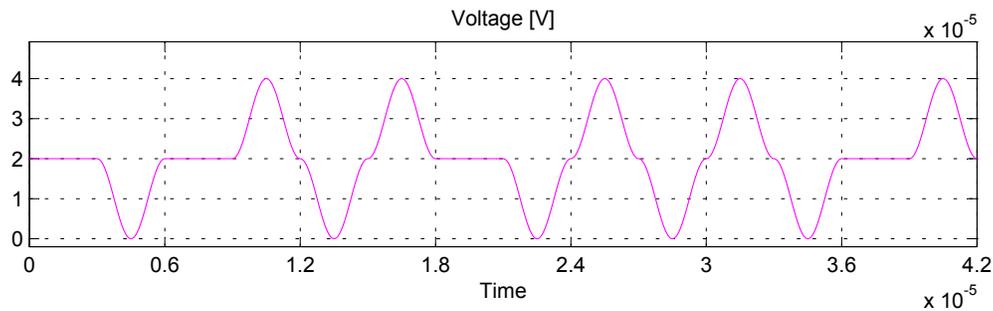
Figura 6.8: Transacción AS-i con el procesador configurado con la dirección cinco.

De igual forma, en la figura 6.9(a), se reprodujo con ayuda de Matlab, el arreglo binario  $00001010111101b$  que debe decodificar el procesador AS-i, el cual corresponde efectivamente a una

petición de intercambio de datos por parte del maestro, dirigido al esclavo identificado con la dirección cinco. También en la figura 6.9(b) se presenta el arreglo correspondiente a la respectiva respuesta de esclavo por parte del procesador AS-i.



(a)



(b)

Figura 6.9: Patrones ideales para una transacción AS-i al procesador configurado con la dirección cinco. (a) Petición de maestro. (b) Respuesta del esclavo.

## 6.4. Aplicación de Validación

Con el objetivo de validar el procesador desarrollado, recreando una situación típica, se propone implementar una arquitectura de red como la mostrada en la figura 6.10, en la cual se evalúa la interconectividad de una arquitectura híbrida compuesta por los perfiles funcionales  $S - 7.3$  de esclavos analógicos y  $S - 3.0$  para dispositivos lógicos, los cuales son manejados por la aplicación maestra diseñada. Es importante resaltar que comercialmente no se consigue una arquitectura de red híbrida debido al perfil cerrado y específico que manejan los dispositivos maestros comerciales, por tal razón se está presentando como resultado una alternativa de interconectividad nueva al mezclar variables binarias y analógicas bajo la misma estación de control. El esquema presentado es sencillo, pero permite evaluar el comportamiento en red de los dispositivos diseñados, de acuerdo a los requerimientos de comunicación exigidos por el estándar. En este se programó una aplicación

Host Controller elemental, donde se maneja un puerto de salida de un esclavo binario dependiendo del nivel sentido por el esclavo analógico.

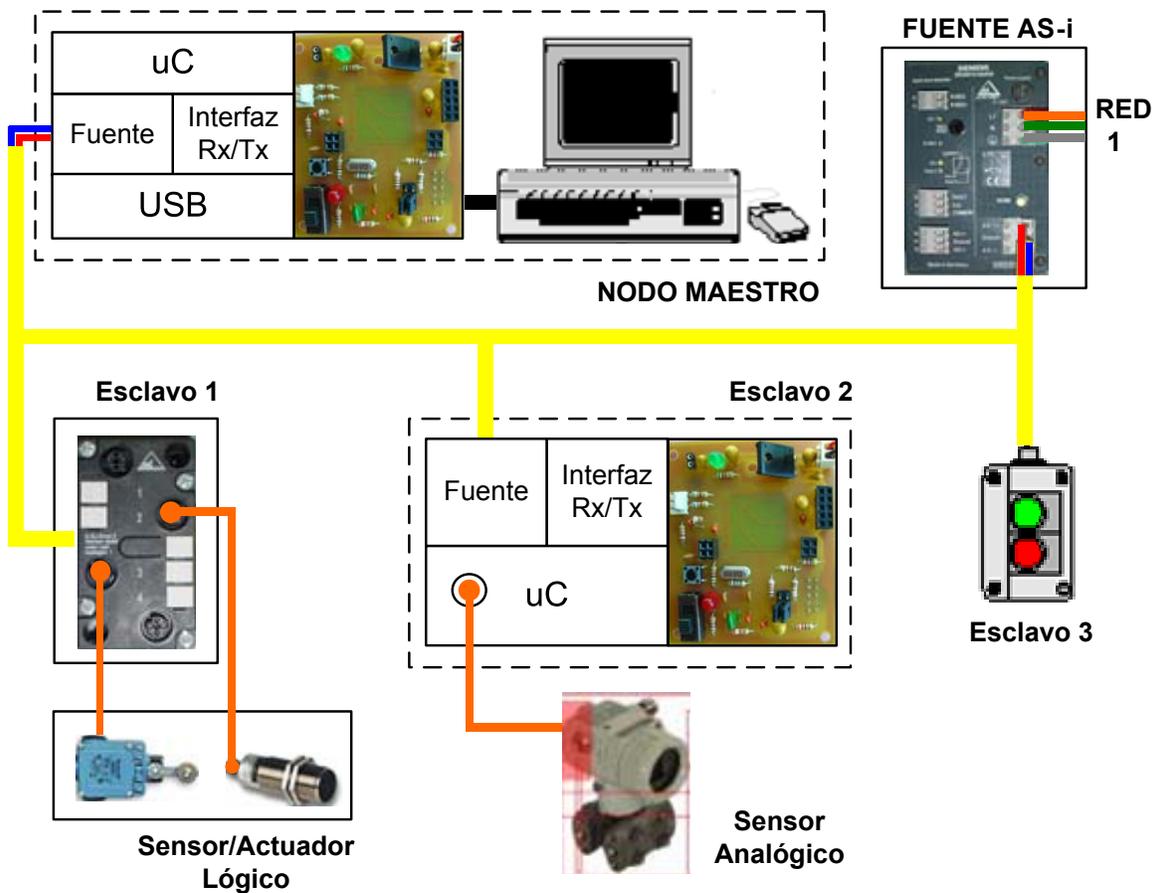


Figura 6.10: Arquitectura de validación propuesta.



# Capítulo 7

## Conclusiones y Recomendaciones

### 7.1. Conclusiones

Se realizó la revisión bibliográfica necesaria para estudiar los conceptos y requerimientos básicos relacionados con la implementación de sistemas de comunicación digital, con el objetivo de encontrar un punto de convergencia entre el modelo OSI extrapolado a buses de campo y el diseño de un procesador de comunicaciones *AS-Interface*. De esta forma se definieron las etapas de desarrollo de un dispositivo esclavo con su respectivo driver de comunicación para habilitar la transmisión de datos sobre líneas de DC.

Dentro del proceso de identificación realizado para un sistema de comunicación industrial, *AS-Interface* fue concebido como un protocolo abierto de alta seguridad en el nivel sensor/actuador, con capacidad para manejar fuentes de información de tipo digital y analógico, para lo cual se adoptaron poderosos esquemas de codificación de fuente de longitud fija (*Manchester II*) y modulación en corriente (*Alternate Pulse Modulation*), que permitan combatir los efectos degradantes del canal sobre la señal, maximizar la velocidad de transmisión y reducir la sensibilidad a interferencias electromagnéticas. Adicionalmente, se ejecuta una codificación de canal en bloques, adicionando *bits* de inicio y parada, para delimitar el datagrama con el propósito de control de errores y sincronía en la recepción, de tal forma que se garantice la comunicación confiable y eficiente de la información sobre el bus.

Para la implementación del procesador *AS-i*, se optó por seleccionar al microcontrolador y evitar la evaluación de soluciones embebidas como el DSP, FPGA o CPLD, ya que debido a los requerimientos de la aplicación se requiere un alto grado de portabilidad y manejo en condiciones adversas, sin exigir una alta capacidad y velocidad de cálculo. Estos dispositivos

programables fueron desarrollados para ejecutar aplicaciones específicas de procesamiento lógico o digital, lo cual no justifica su manejo al subutilizar su potencialidad, por tal razón el microcontrolador MC68HC9S12E128 representó una alternativa de alta flexibilidad, versatilidad y velocidad de ejecución, bajo consumo, bajo costo, facilidad de adaptación al mundo físico y excelente desempeño; con una arquitectura optima que maneja un gran número de módulos periféricos.

El *driver* de línea desarrollado se basa en la implementación de dos generadores *Howland* en configuración diferencial, que actúan como fuente y sumidero, acoplados directamente a los puertos ASi+ y ASi-, respectivamente. De esta forma se presentó un diseño simétrico que no generó desequilibrios críticos en las impedancias características de la línea diferencial usada como canal de comunicación.

El código desarrollado en este proyecto, el cual representa la implementación parcial del perfil lógico  $M \cdot 3$  de dispositivos maestros y el perfil completo  $S - 7 \cdot 3$  de módulos esclavos, fue desarrollado en Lenguaje C con ayuda del compilador *Codewarrior*, permitiendo la programación del microcontrolador a través del puerto USB y habilitando importantes herramientas de depuración que facilitan el proceso de validación de los programas. Sin embargo *Assembler* representó una alternativa de solución para aprovechar la arquitectura de 16-*bits* del micro y reducir los tiempos de ejecución de algunas rutinas críticas como el control de pausas y la generación de patrones analógicos.

Como resultado final de este trabajo se entrega la implementación lógica y física necesaria para establecer un enlace de comunicación basado en los requerimientos del estándar IEC 62026-2, que define el bus de campo *AS-Interface*. Dicha implementación fue completamente depurada y verificada bajo el desarrollo de un perfil esclavo que habilita el manejo de entradas analógicas en el nivel sensor/actuador de la jeraquía de automatización.

Los resultados mostrados en la sección 6, acerca de las pruebas realizadas al procesador de comunicaciones *AS-Interface* sobre una arquitectura de red convencional, fueron totalmente satisfactorios, permitiendo evidenciar la capacidad para desarrollar sistemas que cumplan los requerimientos de compatibilidad e interoperabilidad con soluciones comerciales proporcionadas por fabricantes como *Siemens*.

El desarrollo de este trabajo de investigación permitió que el grupo de investigación *CEMOS* y el grupo de investigación en *Mecatrónica*, adelantaran un trabajo interdisciplinario, orientado a

la implementación del protocolo *AS-Interface* en un microcontrolador, cuyo objetivo converge en la apropiación de la tecnología de bus de campo.

## 7.2. Recomendaciones

Optimizar el desempeño de la fuente de alimentación diseñada, en función del mejoramiento de parámetros como el rizado, el *PSRR* y la eficiencia.

Evaluar el desempeño de otras topologías de conversión tensión-corriente.

Realizar la implementación completa del dispositivo maestro en microcontrolador.

Diseñar un equipo multifuncional, configurable por *Hardware* que habilite el manejo de diferentes perfiles lógicos, tanto de dispositivos maestros, como de dispositivos esclavos.

Consolidar la implementación de un perfil maestro híbrido que permita el manejo de variables binarias y analógicas, bajo una misma arquitectura de red.

Realizar la programación lógica de otros perfiles funcionales definidos por la norma.

Implementar el manejo del modo de direccionamiento extendido, que permita ampliar el número de esclavos con posibilidad de conexión a la red.

Mejorar el consumo de procesador de comunicaciones AS-i, para cumplir con el requerimiento de  $100mA$  que establece la norma.

Realizar un estudio más detallado de la interfaz física del protocolo, de tal forma que se puedan plantear mejoras efectivas, fundamentalmente en lo relacionado con el desequilibrio de impedancias e impedancias características que deben manejar los diferentes dispositivos de red para evitar efectos degradantes en la señal mensaje.

### 7.3. Futuros trabajos en esta área

Para fortalecer el desarrollo de la línea de investigación en *Informática Industrial* se propone continuar el proceso de investigación iniciado con protocolos como *Modbus* [29] y *AS-Interface*, a través de la implementación de pasarelas de comunicación (*Gateways*) entre estas dos arquitecturas, la implementación completa del maestro *AS-i* en *Hardware Embebido*, el diseño de fuentes de alimentación robustas con características especiales de simetría y desacople, y el desarrollo de componentes de red que garanticen la compatibilidad con arquitecturas comerciales. De igual forma, se propone incentivar el trabajo de asimilación de otros buses de campo reconocidos comercialmente como *DeviceNet*, *HART*, *Profibus* y *CAN* [37], con el objetivo de recorrer todos los niveles de la pirámide de automatización, de tal forma que se adquiera la experiencia y habilidad necesaria, para proponer aplicaciones con esquemas de comunicación industrial más complejos, como *Ethernet*, cuyo trabajo se pretende orientar hacia la implementación de *Modbus TCP-IP*. Asimismo, se resalta el importante avance en la aplicación de la tecnología PLC a la automatización de subestaciones eléctricas, donde se está avanzando en el estudio de estándares como el *IEC 60870-5-101*, y el conocimiento de las tecnologías *Wireless*, con el estudio de protocolos como *Bluetooth*. Finalmente, todo este proceso de asimilación, conocimiento e implementación de las arquitecturas de comunicación industrial, debe converger en el planteamiento, evaluación y validación de una estándar de comunicación de aplicación específica.

# Apéndice A

## Análisis de la Fuente Howland

### A.1. Función de transferencia

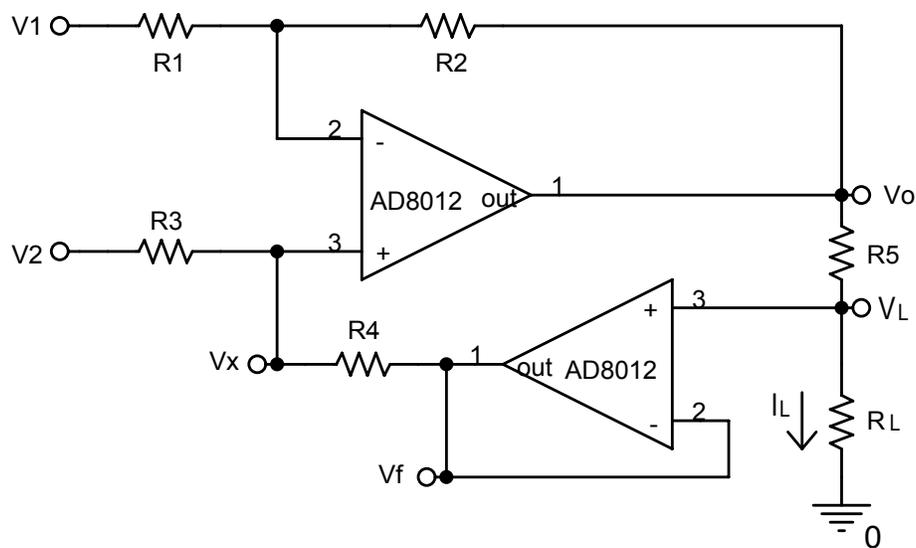


Figura A.1: Fuente Howland sencilla.

$$\frac{(V_1 - V_X)}{R_1} = \frac{(V_X - V_O)}{R_2} \quad (\text{A.1})$$

$$\frac{(V_L - V_X)}{R_4} = \frac{(V_X - V_2)}{R_3} \quad (\text{A.2})$$

$$I_L = \frac{(V_O - V_L)}{R_5} \quad (\text{A.3})$$

de A.1

$$V_O = V_X \left[ \frac{R_2}{R_1} + 1 \right] - V_1 \left[ \frac{R_2}{R_1} \right] \quad (\text{A.4})$$

de A.2

$$V_L = V_X \left[ 1 + \frac{R_4}{R_3} \right] - V_2 \left[ \frac{R_4}{R_3} \right] \quad (\text{A.5})$$

Reemplazando A.4 y A.5 en A.3

$$I_L = \frac{1}{R_5} \left[ V_X \left( \frac{R_2}{R_1} - \frac{R_4}{R_3} \right) + V_2 \left( \frac{R_4}{R_3} \right) - V_1 \left( \frac{R_2}{R_1} \right) \right] \quad (\text{A.6})$$

Para eliminar la dependencia de  $V_X$  en la función de transferencia se ajusta la siguiente relación:

$$\frac{R_2}{R_1} = \frac{R_4}{R_3} \quad (\text{A.7})$$

$$R_2 = R_1 = R_4 = R_3$$

De esta forma la función de transferencia del convertidor voltage-corriente *Howland* es:

$$I_L = \frac{V_I}{R_5} \quad (\text{A.8})$$

donde;

$$V_I = (V_2 - V_1) \quad (\text{A.9})$$

## A.2. Análisis de Impedancias

### A.2.1. Impedancia de Entrada

Por simple inspección y considerando la alta resistencia de entrada del operacional, la impedancia de entrada de la topología es:

$$R_I = R_1 + R_3 = 2R \quad (\text{A.10})$$

### A.3. Impedancia de Salida

$$R_O = \frac{\partial V_L}{\partial I_L} \quad (\text{A.11})$$

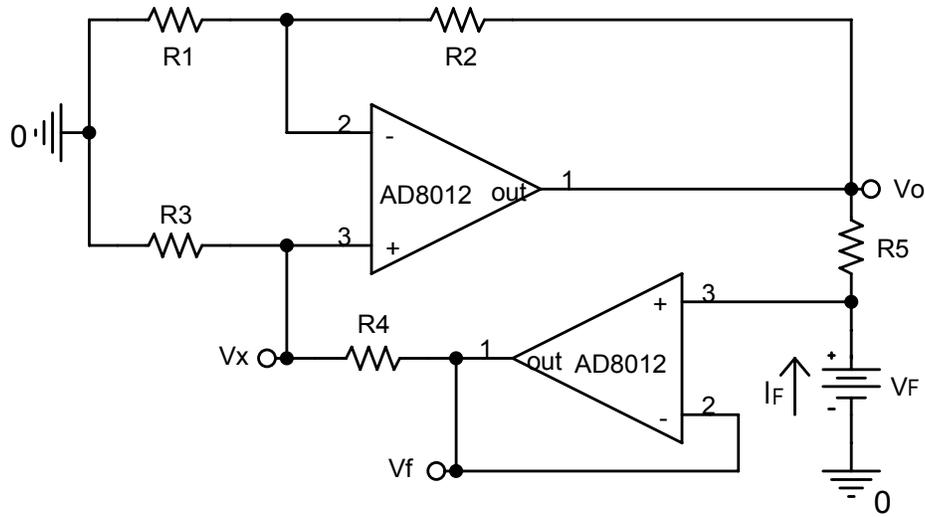


Figura A.2: Cálculo de la impedancia de salida.

$$I_f = \frac{V_f - V_O}{R_5} \quad (\text{A.12})$$

$$V_X = V_f \left( \frac{R_3}{R_3 + R_4} \right) \quad (\text{A.13})$$

$$V_X = V_O \left( \frac{R_1}{R_1 + R_2} \right) \quad (\text{A.14})$$

Igualando A.13 y A.14

$$V_O = V_f \left( \frac{R_3}{R_3 + R_4} \right) \left( \frac{R_1 + R_2}{R_1} \right) \quad (\text{A.15})$$

Reemplazando A.15 en A.12

$$R_O = \frac{V_f}{I_f} = \frac{R_5}{\left[ 1 - \left( \frac{R_3(R_1 + R_2)}{R_1(R_3 + R_4)} \right) \right]} \quad (\text{A.16})$$

De esta forma se concluye que si se siguen cumpliendo la condiciones dadas en A.7 se obtiene una impedancia de salida idealmente infinita, por tal razón se resalta la importancia de lograr un equilibrio alto con resistencias de precisión de 1%.

Para efectos del diseño se ajustan todas las resistencias a un valor grande para lograr una resistencia de entrada alta, sin embargo esto conllevaría una mayor imprecisión y desequilibrio, ya que generaría una mayor diferencia entre los valores de las resistencias, dadas las tolerancias, que afectarían los parámetros anteriormente expuestos.



# Apéndice B

## Análisis de la Fuente Howland Diferencial

### B.1. Función de transferencia

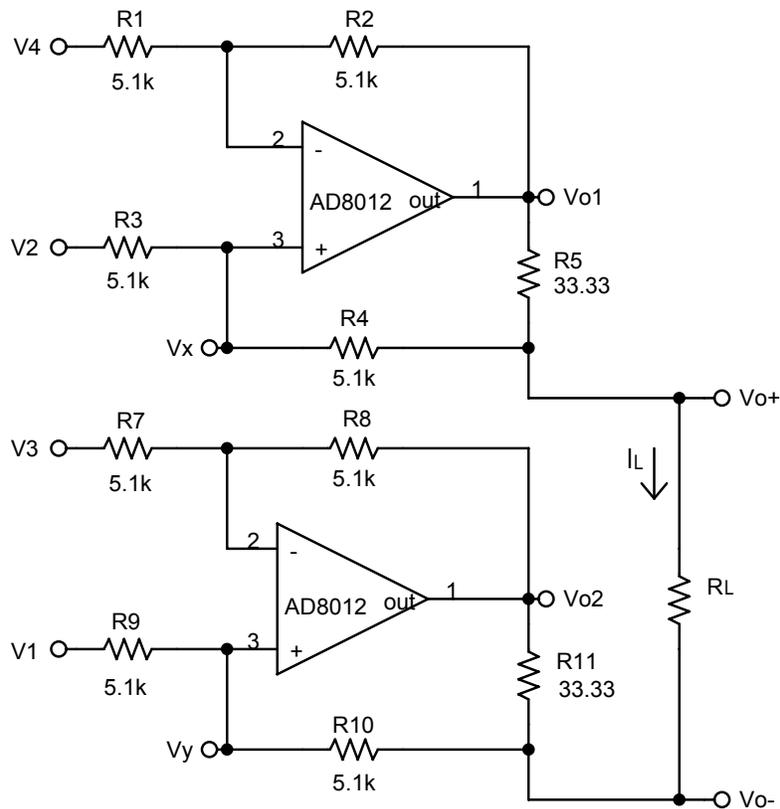


Figura B.1: Fuente Howland Diferencial.

$$\frac{(V_2 - V_X)}{R_3} = \frac{(V_X - V_{O+})}{R_4} \quad (\text{B.1})$$

$$\frac{(V_3 - V_Y)}{R_7} = \frac{(V_Y - V_{O2})}{R_8} \quad (\text{B.2})$$

$$\frac{(V_{O1} - V_X)}{R_2} = \frac{(V_X - V_4)}{R_1} \quad (\text{B.3})$$

$$\frac{(V_{O-} - V_Y)}{R_{10}} = \frac{(V_Y - V_1)}{R_9} \quad (\text{B.4})$$

$$I_L = \frac{(V_{O+} - V_{O-})}{R_L} \quad (\text{B.5})$$

$$I_L = \frac{(V_{O1} - V_{O+})}{R_5} \quad (\text{B.6})$$

$$I_L = \frac{(V_{O-} - V_{O2})}{R_{11}} \quad (\text{B.7})$$

De B.1

$$V_{O+} = \left[ V_X \left[ \frac{1}{R_4} + \frac{1}{R_3} \right] - \frac{V_2}{R_3} \right] R_4 \quad (\text{B.8})$$

De B.3

$$V_{O1} = \left[ V_X \left[ \frac{1}{R_1} + \frac{1}{R_2} \right] - \frac{V_4}{R_1} \right] R_2 \quad (\text{B.9})$$

De B.2

$$V_{O2} = \left[ V_Y \left[ \frac{1}{R_8} + \frac{1}{R_7} \right] - \frac{V_3}{R_7} \right] R_8 \quad (\text{B.10})$$

De B.4

$$V_{O-} = \left[ V_Y \left[ \frac{1}{R_9} + \frac{1}{R_{10}} \right] - \frac{V_1}{R_9} \right] R_{10} \quad (\text{B.11})$$

Reemplazando B.8 y B.9 en B.6

$$I_L = \frac{1}{R_5} \left[ V_X \left[ \frac{R_2}{R_1} - \frac{R_4}{R_3} \right] - \frac{V_4 R_2}{R_1} + \frac{V_2 R_4}{R_3} \right] \quad (\text{B.12})$$

Para eliminar la dependencia de  $V_X$

$$\frac{R_2}{R_1} = \frac{R_4}{R_3} = \frac{R_8}{R_7} = \frac{R_9}{R_{10}} \quad (\text{B.13})$$

Lo cual genera la siguiente expresión,

$$I_L = \left[ \frac{(V_2 - V_4)}{R_5} \right] \frac{R_2}{R_1} \quad (\text{B.14})$$

Por otra parte, si se reemplaza B.10 y B.11 en B.7

$$I_L = \frac{1}{R_{11}} \left[ V_Y \left[ \frac{R_{10}}{R_9} - \frac{R_8}{R_7} \right] - \frac{V_1 R_9}{R_{10}} + \frac{V_3 R_8}{R_7} \right] \quad (\text{B.15})$$

Para eliminar la dependencia de  $V_Y$  se debe cumplir la relación B.13, lo cual genera la siguiente expresión,

$$I_L = \left[ \frac{(V_3 - V_1)}{R_{11}} \right] \frac{R_8}{R_7} \quad (\text{B.16})$$

De estos resultados se concluye que se deben cumplir las siguientes relaciones para garantizar una operación de forma diferencial:

$$V_2 = V_3, V_1 = V_4, R_1 = R_7, R_2 = R_8, R_3 = R_9, R_4 = R_{10} \text{ y } R_5 = R_{11}$$

## B.2. Impedancia de Salida

$$R_O = \frac{V_F}{I_F} \quad (\text{B.17})$$

$$V_F = (V_{O+} - V_{O-}) \quad (\text{B.18})$$

$$I_F = \frac{(V_{O+} - V_{O1})}{R_5} \quad (\text{B.19})$$

$$I_F = \frac{(V_{O2} - V_{O-})}{R_{11}} \quad (\text{B.20})$$

$$V_X = \frac{(V_{O+})R_3}{R_4 + R_3} \quad (\text{B.21})$$

$$V_X = \frac{(V_{O1})R_1}{R_2 + R_1} \quad (\text{B.22})$$

$$V_Y = \frac{(V_{O-})R_9}{R_9 + R_{10}} \quad (\text{B.23})$$

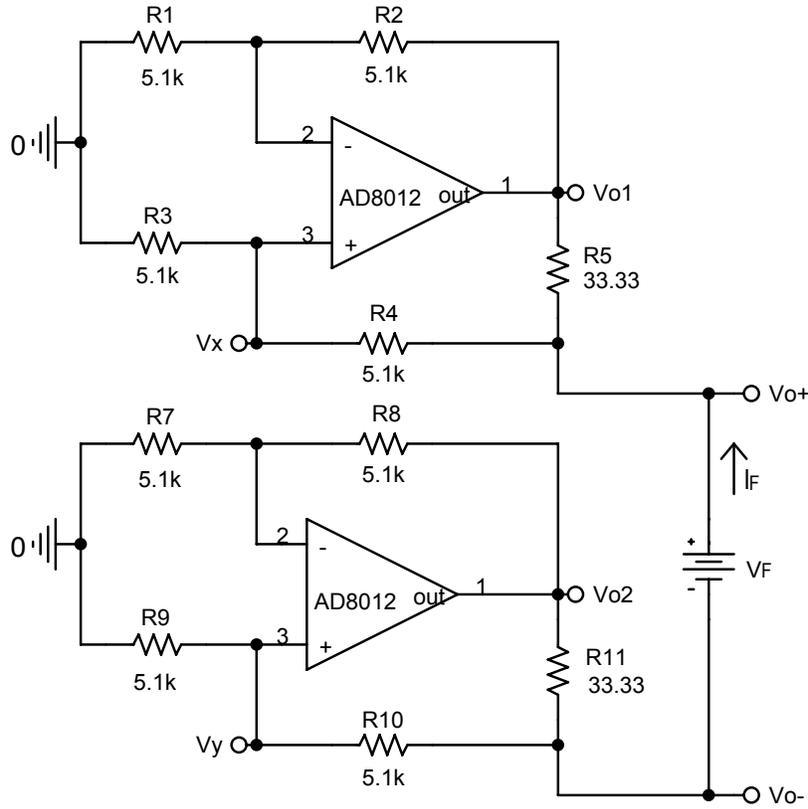


Figura B.2: Cálculo de la impedancia de salida.

$$V_Y = \frac{(V_{O2})R_7}{R_7 + R_8} \quad (\text{B.24})$$

Despejando  $V_{O+}$  de B.19,  $V_{O-}$  de B.20 y reemplazando en B.18

$$V_F = I_F(R_5 + R_{11}) + V_{O1} - V_{O2} \quad (\text{B.25})$$

Igualando B.21 y B.22,

$$V_{O1} = \left[ \frac{R_3(R_2 + R_1)}{R_1(R_4 + R_3)} \right] V_{O+} \quad (\text{B.26})$$

Igualando B.23 y B.24,

$$V_{O2} = \left[ \frac{R_9(R_7 + R_8)}{R_7(R_9 + R_{10})} \right] V_{O-} \quad (\text{B.27})$$

Si se cumple la relacion

$$\left[ \frac{R_3(R_2 + R_1)}{R_1(R_4 + R_3)} \right] = \left[ \frac{R_9(R_7 + R_8)}{R_7(R_9 + R_{10})} \right] \quad (\text{B.28})$$

y se reemplaza B.26 y B.27 en B.25, se obtiene,

$$V_F = I_F(R_5 + R_{11}) + V_F \left[ \frac{R_3(R_2 + R_1)}{R_1(R_4 + R_3)} \right] \quad (B.29)$$

Expresión que permite obtener una relación determinística para la impedancia de salida de la configuración *Howland* diferencial:

$$R_O = \frac{V_F}{I_F} = \frac{(R_5 + R_{11})}{1 - \left[ \frac{R_3(R_2 + R_1)}{R_1(R_4 + R_3)} \right]} \quad (B.30)$$

Esta expresión permite concluir que si se garantiza la condición B.13, se puede obtener una impedancia de salida idealmente  $\infty$ .

### B.3. Corriente en Modo Común

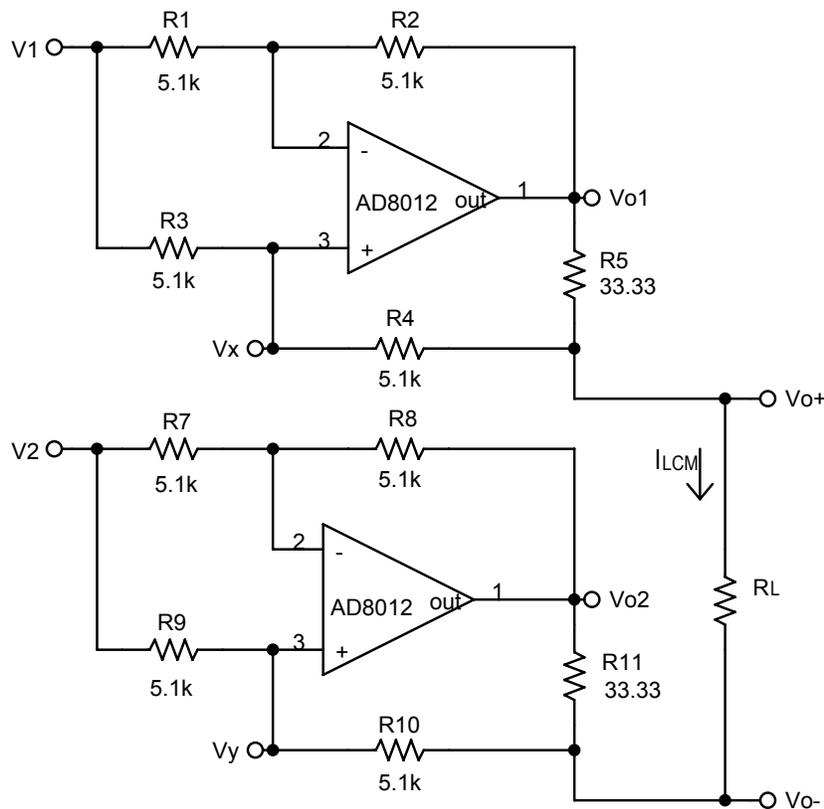


Figura B.3: Cálculo de la corriente en modo común.

$$I_L = \frac{(V_{O+} - V_{O-})}{R_L} \quad (B.31)$$

$$I_L = \frac{(V_{O1} - V_{O+})}{R_5} \quad (\text{B.32})$$

$$I_L = \frac{(V_{O-} - V_{O2})}{R_{11}} \quad (\text{B.33})$$

$$\frac{(V_1 - V_X)}{R_1} = \frac{(V_X - V_{O1})}{R_2} \quad (\text{B.34})$$

$$\frac{(V_1 - V_X)}{R_3} = \frac{(V_X - V_{O+})}{R_4} \quad (\text{B.35})$$

$$\frac{(V_2 - V_Y)}{R_7} = \frac{(V_Y - V_{O2})}{R_8} \quad (\text{B.36})$$

$$\frac{(V_2 - V_Y)}{R_9} = \frac{(V_Y - V_{O-})}{R_{10}} \quad (\text{B.37})$$

De B.43

$$V_{O1} = \left[ V_X \left[ \frac{1}{R_7} + \frac{1}{R_1} \right] - \frac{V_1}{R_1} \right] R_2 \quad (\text{B.38})$$

De B.35

$$V_{O+} = \left[ V_X \left[ \frac{1}{R_4} + \frac{1}{R_3} \right] - \frac{V_1}{R_3} \right] R_4 \quad (\text{B.39})$$

Reemplazando B.38 y B.39 en B.32

$$I_L = \frac{1}{R_5} \left[ V_X \left[ \frac{R_2}{R_1} - \frac{R_4}{R_3} \right] \right] + V_1 \left[ \frac{R_4}{R_3} - \frac{R_2}{R_1} \right] \quad (\text{B.40})$$

De B.36

$$V_{O2} = \left[ V_Y \left[ \frac{1}{R_8} + \frac{1}{R_7} \right] - \frac{V_2}{R_7} \right] R_8 \quad (\text{B.41})$$

De B.37

$$V_{O-} = \left[ V_Y \left[ \frac{1}{R_{10}} + \frac{1}{R_9} \right] - \frac{V_2}{R_9} \right] R_{10} \quad (\text{B.42})$$

Reemplazando B.38 y B.39 en B.32

$$I_L = \frac{1}{R_{11}} \left[ V_Y \left[ \frac{R_{10}}{R_9} - \frac{R_8}{R_7} \right] \right] + V_2 \left[ \frac{R_8}{R_7} - \frac{R_{10}}{R_9} \right] \quad (\text{B.43})$$

Para dar ganancia al circuito y cumplir con la condición en modod común se debe cumplir la condición B.13

# Apéndice C

## Análisis de la línea de transmisión AS-Interface

### C.1. Modelo Circuitual

Para el análisis circuitual presentado a continuación se debe tener en cuenta las referencias de la tabla C.1.

Tabla C.1: Referencias del modelo circuitual.

Referencia	Descripción
$Z_L$	Impedancia de Línea
$Z_E$	Impedancia del Esclavo
$R_A, R_B$	Impedancia de la fuente de corriente
$V_{DC}$	Bus de DC

Del equivalente circuitual mostrado en la figura C.1 y aplicando el concepto de superposición, se pueden obtener los siguientes resultados,

Eliminado el efecto de la fuente de DC,

$$V_{X(AC)} = I_A [(R_A + R_B) // Z_L // Z_E // Z_S] \quad (C.1)$$

Eliminado el efecto de la fuente de corriente,

$$V_{X(DC)} = V_{DC} \left[ \frac{Z_E // Z_L // (R_A + R_B)}{Z_S + (Z_E // Z_L // (R_A + R_B))} \right] \quad (C.2)$$

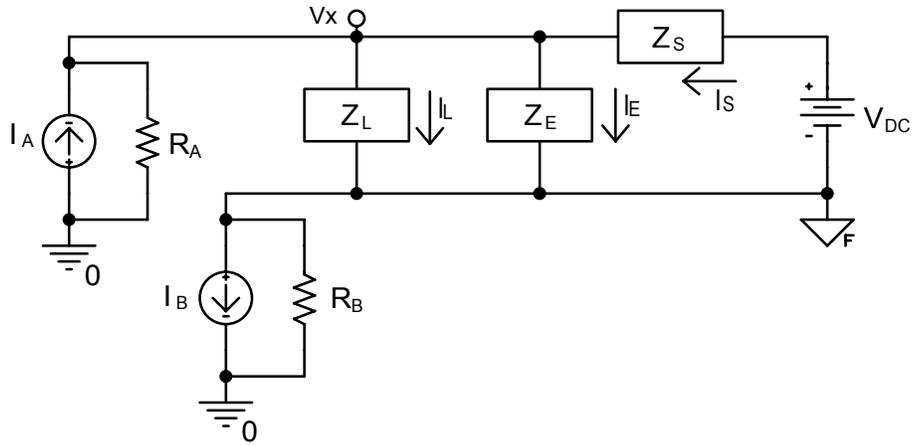


Figura C.1: Modelo circuital de la línea AS-i.

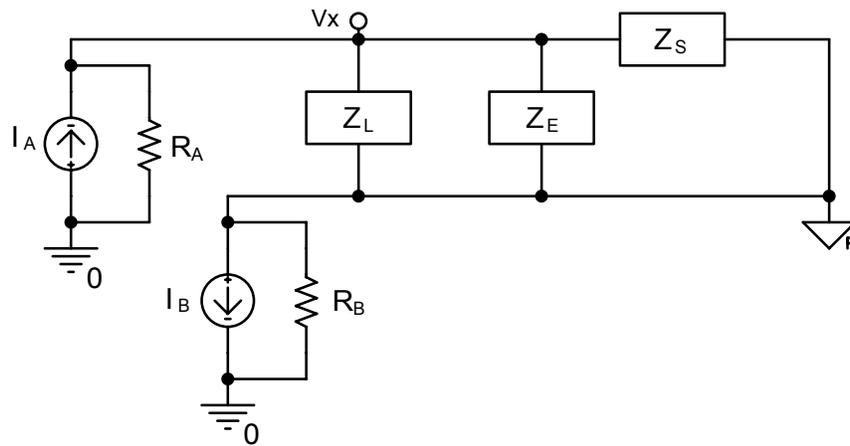


Figura C.2: Equivalente circuital para evaluar el aporte de la fuente de AC.

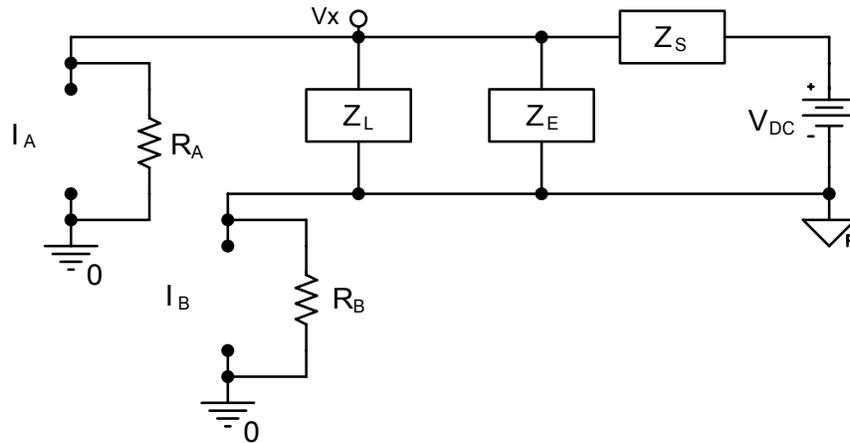


Figura C.3: Equivalente circuital para evaluar el aporte de la fuente de DC.

---

En conclusión se puede observar claramente que la señal sobre el bus AS-i esta conformada por una componente de continua  $V_{X(DC)}$ , aproximadamente igual al  $V_{DC}$  y una componente de señal  $V_{X(AC)}$ , proporcional a la señal de corriente generada con la fuente *Howland*, la cual posee un carga equivalente de tipo inductivo, debido al comportamiento dominante del circuito de simetria y desacople ofrecido por  $Z_S$ .



# Apéndice D

## Circuitos de Acople

### D.1. Acople de Referencias

Los planos de referencia para los circuitos analógicos y digitales deben estar separados, y las tierras deben conectarse cerca de las alimentaciones, teniendo el cuidado de no generar lazos. Si la aplicación exige la conexión de estas dos referencias, se debe hacer uso de *ferritas bead* para compartir la misma referencia, de no ser así se estaría generando un lazo que actuaría como una antena de *RF*. En la referencia [38] se presentan algunas técnicas de reducción de ruido en las fuentes de alimentación y acople de planos de referencia, entre las cuales se destacan:

#### D.1.1. Capacitor *Bypass*

La mayoría de circuitos operan en DC. Sin embargo este voltaje generalmente posee fluctuaciones que generan rizado y componentes de ruido, afectando la funcionalidad de los dispositivos. La principal función del  $C_{Bypass}$  es mitigar las componentes de rizado y disminuir las componentes de ruido, de esta forma se crea un camino de baja impedancia a tierra. De igual forma también ayuda a compensar la caída de voltaje causada por transitorios de corriente generados por conmutación. El valor exacto de este condensador debe ser calculado en función de parámetros como la capacitancia de salida, la velocidad de conmutación, la inductancia parasita y el margen de ruido deseado.

La diferencia entre un capacitor *Bypass* y uno de filtro depende de donde se este usando. Cuando es usado para eliminar ruido de baja frecuencia en la línea de alimentación, este se refiere como un condensador de filtro. Y su aplicación más frecuente es un condensador de  $22\mu F$  entre  $V_{DD}$  y  $G_{ND}$ . Por otra parte el *Bypass* es utilizado en aplicaciones de alta frecuencia para proporcionar un camino de baja impedancia ante cambios instantáneos de corriente y de esta forma proteger el sistema de alimentación contra fluctuaciones inducidas.

### D.1.2. Desacople

Los condensadores de desacople deben ser ubicados lo más cerca posible a cada pin de alimentación. Típicamente se usan condensadores de  $0.1\mu F$  conectados entre  $V_{DD}$  y  $G_{ND}$ . Lo recomendado es usar capacitores de montaje superficial caracterizados por su baja  $ESL$  y  $ESR$ , sin embargo los condensadores monolíticos o cerámicos ofrecen un óptimo desempeño. El valor de este capacitor depende fuertemente de la frecuencia del ruido generado a altas frecuencias (componentes armónicas).

### D.1.3. Ferritas *Bead*

Los circuitos de conmutación típicamente generan ruido, por tal razón es necesario aislar estas componentes para evitar que se reflejen en el PCB (*Printed Circuit Board*). La mejor forma de solucionar este problema es insertar una ferrita entre la alimentación del circuito digital y la alimentación de la parte analógica. Esta ferrita no incrementa ni degrada el desempeño del circuito, está solo es usada para proporcionar aislamiento de las componentes de ruido. Las ferritas *Beads* estan compuestas por materiales ferromagnéticos que no son susceptibles a campos magnéticos externos radiados

Para la selección de la ferrita es importante conocer la frecuencia del ruido no deseado, con el objetivo de establecer la impedancia de la ferrita en función de la frecuencia, tamaño, material y numero de vueltas. La impedancia en DC de la ferrita debe ser cercana a cero, mientras que a la frecuencia de operación deber ser superior a  $50\Omega$  bajo condiciones de carga. De esta forma se previene el efecto de las componentes armónicas sobre el sistema. La ferrita debe ser capaz de proporcionar la máxima corriente proporcionada por las fuentes de alimentación.

La ferritas *Bead* son usadas frecuentemente en proceso de filtrado en aplicaciones de alta frecuencia. Estas se pueden dividir en dos clases: Ferritas Resonantes *High-Q* y Ferritas no Resonantes (*Low-Q*). Las primeras no se pueden ubicar en un circuito digital, estas son usadas para construir circuitos osciladores y filtros para RF, o en aplicaciones que necesiten elementos altamente resonantes. En aplicaciones de filtrado de la alimentación de circuitos digitales lo último que se quiere es la resonancia, por tal razón el segundo tipo es la mejor opción, colocandolas en serie con la alimentación. Estos elementos se forman atravesando un nucleo de ferrita con unas cuantas vueltas de cable metálico para formar un camino de alta impedancia en el rango de operación, de esta forma circulará una corriente que genera un campo magnético funcionando como un simple *RF choke*.

## D.2. Aplicaciones PLC

La superposición de una señal mensaje sobre un nivel de *DC* implica que los circuitos de acople (*Driver* de Línea) y la línea de alimentación (Bus AS-i) deben ser cuidadosamente diseñados e interconectados para lograr una compatibilidad óptima, entre potencia y comunicación. Acoplar una señal de comunicación sobre el canal PLC se puede lograr usando alternativas como el acople de *Modo Diferencial* o el acople en *Modo Común*, así mismo se debe proporcionar el adecuado aislamiento galvánico del sistema PLC con la línea de potencia, lo cual se logra a través de acoples capacitivos o inductivos. El acoplamiento inductivo puede generar pérdidas de varios decibeles, pero evita la conexión física a la red, lo cual lo hace más seguro y fácil de implementar que el capacitivo, mientras que el acoplamiento capacitivo, realiza el filtrado pasa alto requerido para ejecutar un diseño fácil y compacto. Desde el punto de vista práctico, los circuitos de acople se aplican como una combinación de ambas técnicas. [2]

### D.2.1. Coupling Capacitors

Los capacitores de acople son ampliamente usados en sistemas PLC para acoplar la señal a la línea de alimentación, pero también actúan como parte de configuraciones más sofisticadas como filtros de orden superior. Los requerimientos y características esenciales de estos componentes han sido estandarizadas por la norma *ANSI C93.1 – 1972*. Estos condensadores se encargan de llevar la corriente de comunicación, por lo tanto deben ser caracterizados para aplicaciones de alta frecuencia (la frecuencia auto-resonante debe ser mayor que la frecuencia de modulación) y filtrar el voltaje de potencia (caída a través del capacitor), por lo tanto también deben soportar altos niveles de voltaje.

### D.2.2. Coupling Transformers

Los transformadores de acople proporcionan aislamiento galvánico, actúan como adaptador de impedancias y deben ser capaces de manejar la señal de comunicación de alta frecuencia. La señal de potencia maneja frecuencias bajas y niveles altos, generando un efecto de saturación, razón por la cual debe ser previamente filtrada antes de entrar al transformador.

### D.2.3. Blocking Inductors

Los inductores de bloqueo son diseñados para la frecuencia de potencia (prevenir saturación) y para los niveles de corriente de manejo (prevenir grandes caídas de voltaje). Estos inductores

deben bloquear la frecuencia de modulación, razón por la cual el punto de resonancia debe estar por encima de esta. Los inductores con núcleo de aire satisfacen de forma adecuada esta aplicación.

#### **D.2.4. Resistores**

En los circuitos de acoplamiento se debe evitar el uso de resistores, ya que representan una pérdida de potencia, tanto de señal, como de alimentación.

# Apéndice E

## Pruebas sobre el *Driver* de línea

A continuación se presentan los resultados obtenidos para validar el diseño de la fuente de corriente diferencial diseñada, la cual actúa como *Driver* de línea para acoplar el procesador al bus AS-i. Estas pruebas se realizaron para los valores definidos previamente, los cuales son  $2V_{P-P}$  para generar una corriente  $60mA$  a través de una resistencia de  $33.33\Omega$  y utilizando los circuitos integrados seleccionados para la aplicación.

### E.1. Pruebas de Simulación

Estas pruebas cuyos resultados gráficos son mostrados en la figura E.1 se desarrollaron usando una fuente senoidal configurada a la frecuencia de operación de  $166.67kHz$ , para realizar un barrido sobre la resistencia de carga, sin modelo de la línea.

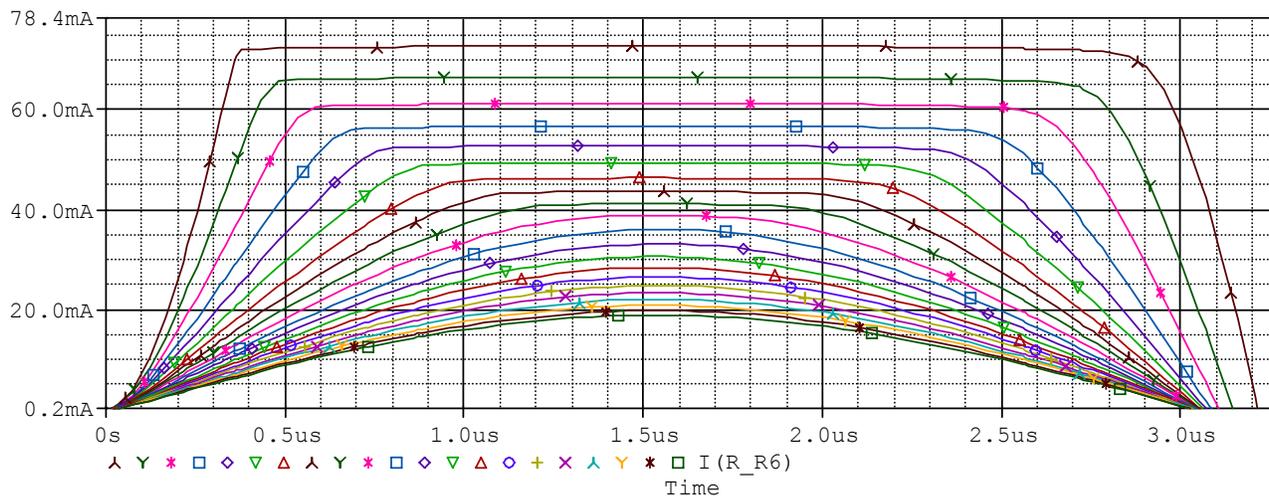


Figura E.1: Resultados de simulación en Orcad.

En la tabla E.1 se muestra una comparación entre los valores teóricos y los resultados obtenidos por simulación para  $R_{Feedback} = 10k\Omega$ .

Tabla E.1: Resultados de simulación para  $R_{Feedback} = 10k\Omega$ .

$R_{Scale}$ [ $\Omega$ ]	I Teórico [mA]	I Simulación [mA]	% Error
53	37.73	36.02	4.53%
58	34.48	33.10	4.01%
63	31.74	30.56	3.72%
73	27.39	26.44	3.49%
88	22.72	21.98	3.27%
103	19.41	18.81	3.13%

De igual forma, en la tabla E.2 se muestra una comparación entre los valores teóricos y los resultados obtenidos por simulación para  $R_{Feedback} = 1k\Omega$ .

Tabla E.2: Resultados de simulación para  $R_{Feedback} = 1k\Omega$ .

$R_{Scale}$ [ $\Omega$ ]	I Teórico [mA]	I Simulación [mA]	% Error
53	37.73	37.16	1.52%
73	27.39	27.22	0.62%
103	19.41	19.31	0.53%

## E.2. Prueba Experimentales

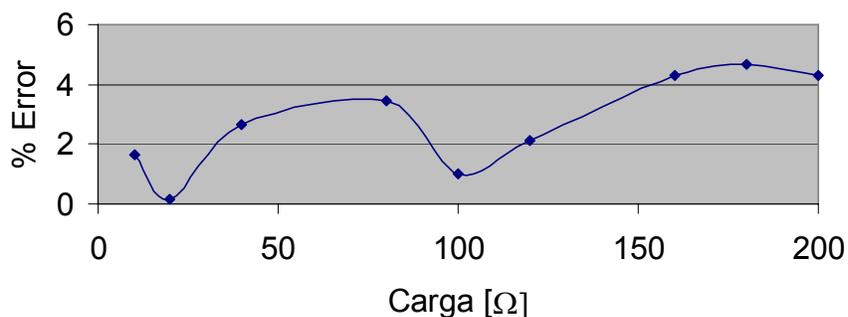
Estas pruebas se desarrollaron usando un generador de señales físico a una frecuencia de  $166.67kHz$ . La intención era evaluar el comportamiento de la fuente *Howland* ante variaciones de la  $R_{Feedback}$ .

A continuación se presentan los resultados para una  $R_{Feedback} = 1k\Omega$ , con respecto a un valor teórico nominal de  $60,606mA$ . En la tabla E.3 se muestran los valores medidos para la corriente y el voltage en la carga, que fueron tomados como referencia para el cálculo del porcentaje de error visualizados en la figura E.2.

En la tabla E.4 se muestran los valores medidos para la corriente y el voltage en la carga con una  $R_{Feedback} = 5.1k\Omega$ , los cuales fueron tomados como referencia para calcular los porcentajes

Tabla E.3: Resultados experimentales para  $R_{Feedback} = 1k\Omega$ .

$R_L$ [ $\Omega$ ]	$V_L$ [V]	$I_L$ [mA]	% Error
10	0.61	61.10	1.64%
20	1.21	60.05	0.17%
40	2.36	59.00	2.65%
80	4.68	58.50	3.47%
100	6.01	60.00	1.00%
120	7.12	59.33	2.10%
160	9.28	58.00	4.30%
180	10.40	57.78	4.66%
200	11.60	58.00	4.30%

Figura E.2: Curva de %Error para  $R_{Feedback} = 1k\Omega$ .

de error presentados en la figura E.3.

Tabla E.4: Resultados de experimentales para  $R_{Feedback} = 5.1k\Omega$ .

$R_L$ [ $\Omega$ ]	$V_L$ [V]	$I_L$ [mA]	% Error
10	0.61	61.60	1.64%
20	1.16	58.00	4.30%
40	2.36	59.00	2.65%
80	4.64	58.00	4.30%
100	5.68	56.80	6.28%
120	6.96	57.50	4.30%
160	9.20	56.66	5.12%
180	10.20	56.00	6.50%
200	11.20	58.00	7.60%

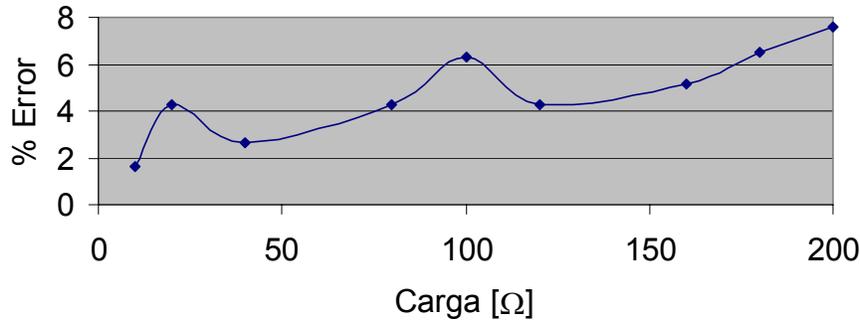


Figura E.3: Curva de %Error para  $R_{Feedback} = 5.1k\Omega$ .

Finalmente en la tabla E.5 se muestran los valores obtenidos de la corriente y el voltage en la carga para una  $R_{Feedback} = 11.7k\Omega$ , los cuales sirvieron de referencia para estimar los porcentajes de error mostrados en la figura E.4.

Tabla E.5: Resultados de experimentales para  $R_{Feedback} = 11.7k\Omega$ .

$R_L$ [Ω]	$V_L$ [V]	$I_L$ [mA]	% Error
10	0.60	60.40	0.34%
20	1.19	59.50	1.82%
40	2.32	58.00	4.30%
80	4.36	54.50	10.07%
100	5.32	53.20	12.22%
120	6.40	53.33	12.00%
160	8.16	51.00	15.85%
180	8.88	49.33	18.60%
200	9.20	46.00	24.10%

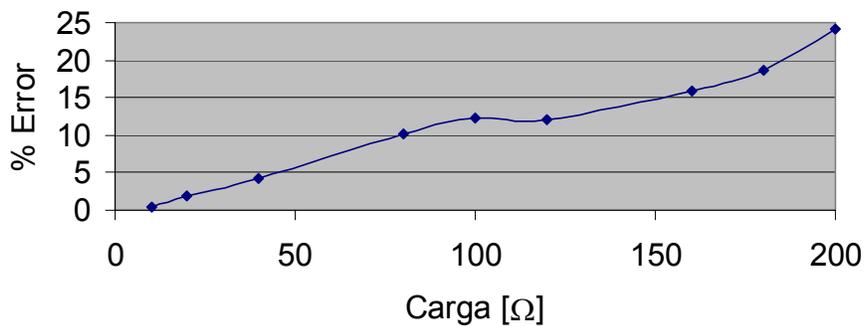


Figura E.4: Curva de %Error para  $R_{Feedback} = 11.7k\Omega$ .

De este proceso se puede evidenciar claramente las consideraciones expuestas en el proceso de diseño del *driver* de línea, donde con mayor resistencia se presentan mayores errores, debido a que las tolerancias generarían un desequilibrio crítico en la impedancia de salida y la relación de transconductancia. En esto caso es importante considerar un equilibrio entre la impedancia de entrada y la imprecisión del nivel de corriente. Con la solución final ( $R_{Feedback} = 5.1k\Omega$ ) no se permite una impedancia de entrada elevada, según los requerimientos ideales del generador de corriente, pero si se logra estabilidad y exactitud en el valor de corriente, al reducir los errores provocados por las tolerancias.



## Apéndice F

# Esquemático de la tarjeta diseñada para el Microcontrolador

Esta tarjeta fue diseñada con la finalidad de llevar a cabo la implementación de otros dispositivos esclavos que permitieran facilitar el proceso de validación del procesador propuesto a través de una sencilla aplicación de comunicación en red.

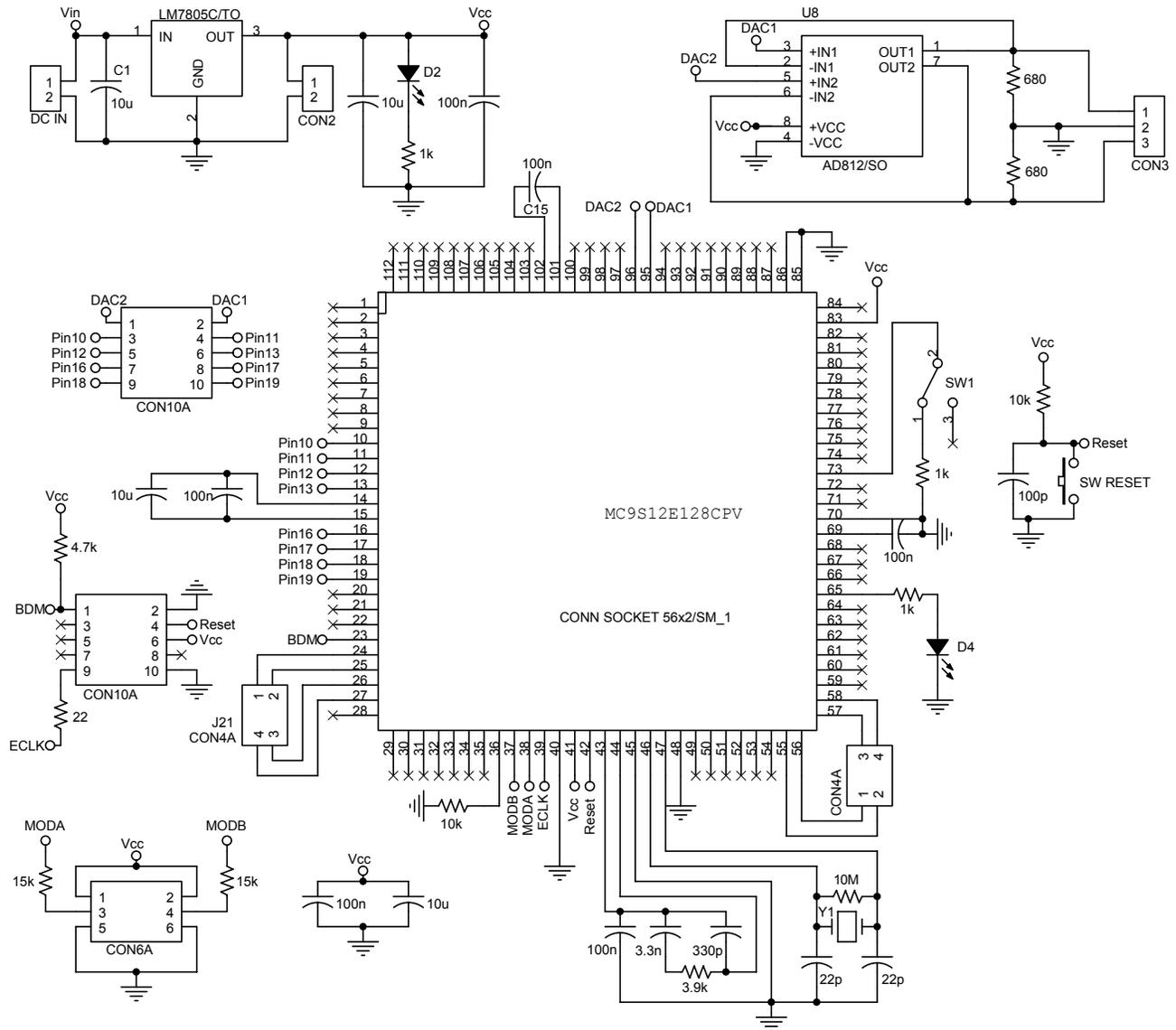


Figura F.1: Esquemático de la tarjeta de aplicación para el microcontrolador.

# Apéndice G

## Estudio de Impedancias

En este apéndice se presentan los resultados obtenidos con Matlab para obtener el rango de operación del *driver* de línea de acuerdo a la impedancia característica de la línea y la impedancia equivalente del dispositivo esclavo.

tomando como referencia la expresión de impedancia característica de la línea AS-i y asumiendo que el sistema esta diseñado para una máxima transferencia de señal, es decir considerando una resistencia de carga similar al  $Z_0$  de la línea

### G.1. Impedancia característica de la línea

En esta sección se presentan los resultados obtenidos para evaluar el comportamiento de la impedancia característica de la línea y las demoras de propagación de la misma, con el objetivo de definir el rango de operación o variabilidad de la carga. Para este estudio se tomó como referencia el modelo de la línea AS-i presentado en la figura G.1.

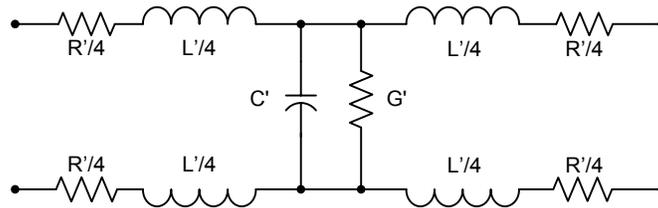


Figura G.1: Modelo equivalente de la línea AS-i.

Donde el rango de valores permitido para la frecuencia de operación es mostrado en la tabla G.1.

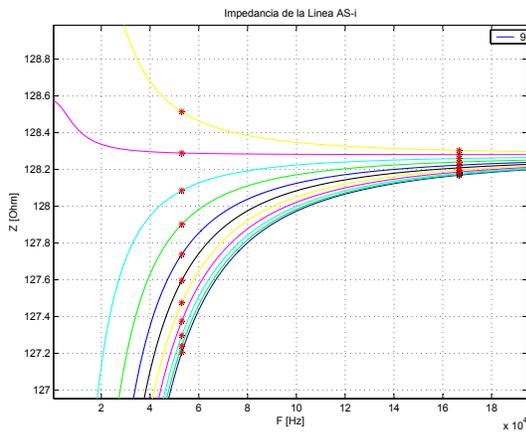
Tabla G.1: Parámetros de la línea AS-i

Parámetro	Rango
$R'$	$< 90m\Omega/m$
$C'$	$< 80pF/m$
$G'$	$\leq 5\mu S/m$
$L'$	$400nH/m - 1300nH/m$
$Z$	$70\Omega - 140\Omega$
$t'$	$\leq 8.3nS/m$

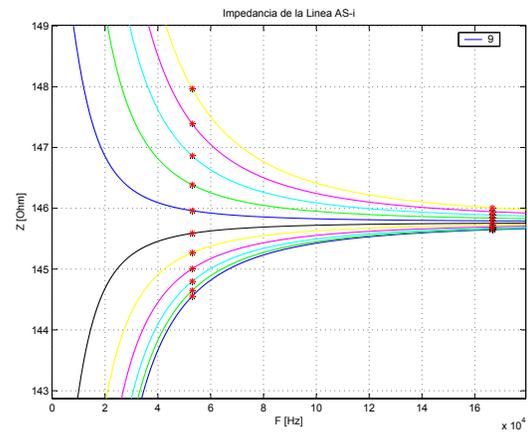
Con base en esta información se definen expresiones para la impedancia característica de la línea y el tiempo de propagación:

$$Z = \left| \sqrt{\frac{R' + j\omega L'}{G' + j\omega C'}} \right| \quad (G.1)$$

$$t' = \sqrt{L'C' - \frac{R'G'}{\omega^2}} \quad (G.2)$$



(a)



(b)

Figura G.2: Resultados de la variación de la resistencia en el modelo de la línea AS-i. (a) Opción 1. (b) Opción 2.

## G.2. Impedancia equivalente del esclavo

Tabla G.2: Variación de la resistencia del modelo de la línea

Opción	Rmin [ $\Omega/m$ ]	Rmax [ $\Omega/m$ ]	C [pF/m]	L [nH/m]	G [ $\mu S/m$ ]	Long. [m]
1	9	89	79	1300	4.9	100
2	9	89	40	850	2.5	50

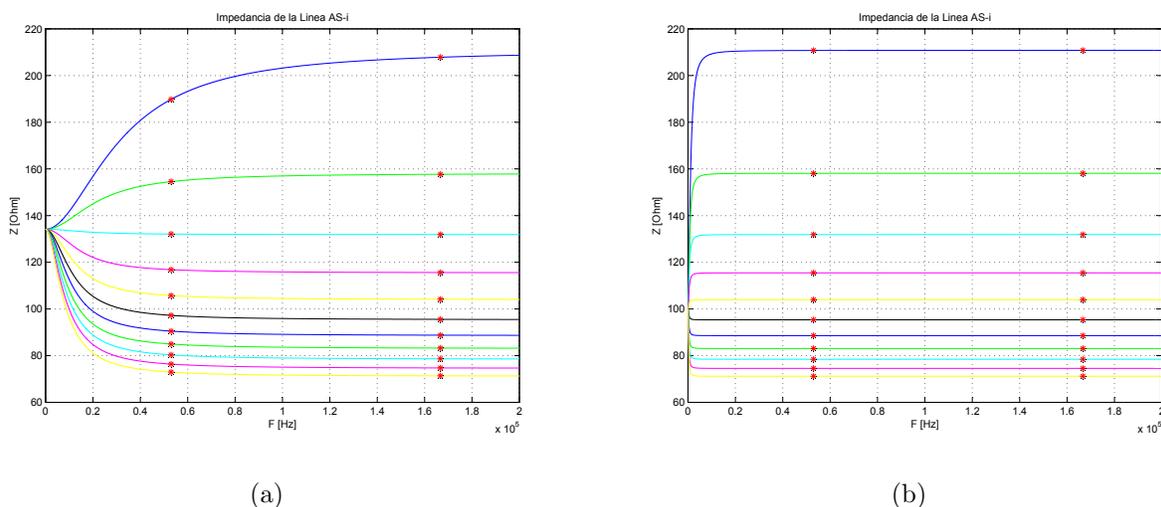


Figura G.3: Resultados de la variación de la capacitancia en el modelo de la línea AS-i. (a) Opción 3. (b) Opción 4.

Tabla G.3: Variación de la capacitancia del modelo de la línea

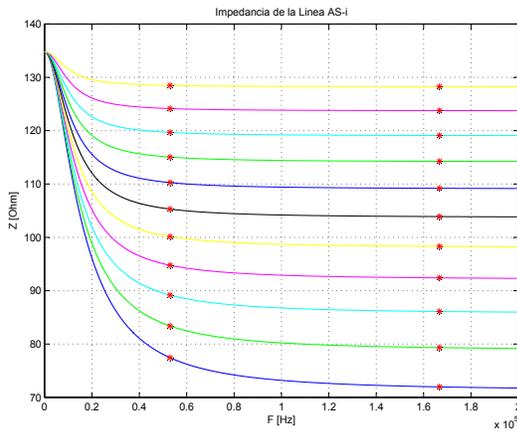
Opción	Cmin [pF/m]	Cmax [pF/m]	R [ $\Omega/m$ ]	L [nH/m]	G [ $\mu S/m$ ]	Long. [m]
3	9	79	45	400	2.5	50
4	9	79	1	400	0.1	1

Tabla G.4: Variación de la inductancia del modelo de la línea

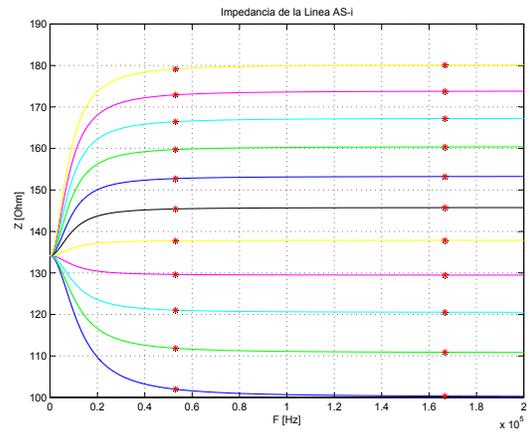
Opción	Lmin [nH/m]	Lmax [nH/m]	R [ $\Omega/m$ ]	C [pF/m]	G [ $\mu S/m$ ]	Long. [m]
5	401	1299	89	79	4.9	100
6	401	1299	45	40	2.5	50

Tabla G.5: Variación de la conductancia del modelo de la línea

Opción	Gmin [ $\mu S/m$ ]	Gmax [ $\mu S/m$ ]	R [ $\Omega/m$ ]	C [pF/m]	L [nH/m]	Long. [m]
7	0.9	4.9	89	79	1300	100
8	0.9	4.9	45	40	850	50

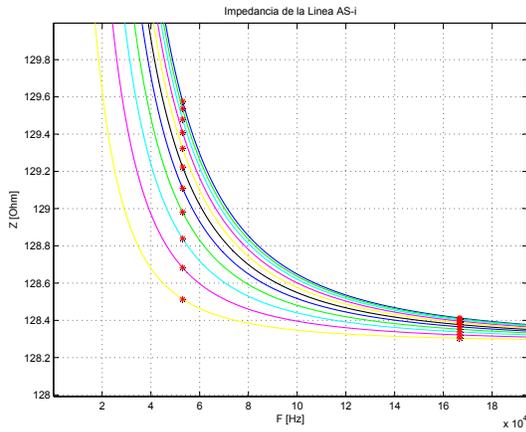


(a)

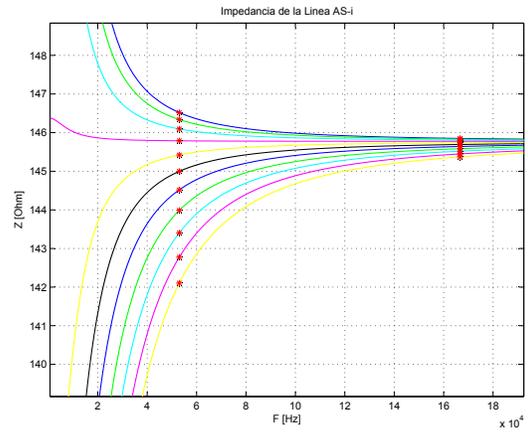


(b)

Figura G.4: Resultados de la variación de la inductancia en el modelo de la línea AS-i. (a) Opción 5. (b) Opción 6.



(a)



(b)

Figura G.5: Resultados de la variación de la conductancia en el modelo de la línea AS-i. (a) Opción 7. (b) Opción 8.

Tabla G.6: Variación de la longitud del modelo de la línea

Opción	Long.min [m]	Long.max [m]	R [ $\Omega/m$ ]	C [pF/m]	L [nH/m]	G [ $\mu S/m$ ]
9	0.1	100	89	79	1299	4.9
10	0.1	100	45	40	850	2.5

Tabla G.7: Variación del tiempo de propagación del modelo de la línea

Opción	Long.min [m]	Long.max [m]	R [ $\Omega/m$ ]	C [pF/m]	L [nH/m]	G [ $\mu S/m$ ]
11	1	100	1	1	400	0.1

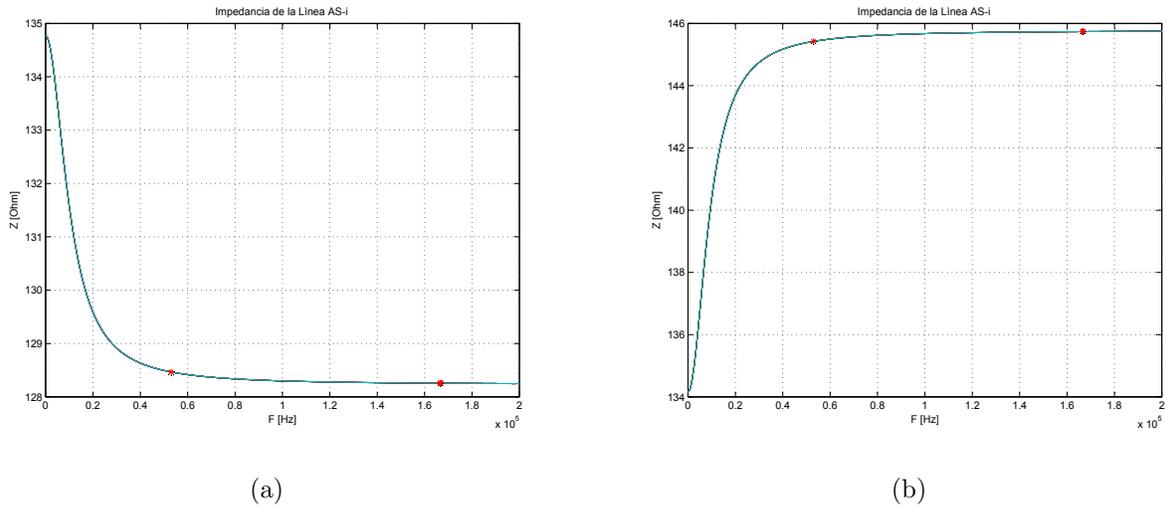


Figura G.6: Resultados de la variación de la longitud en el modelo de la línea AS-i. (a) Opción 9. (b) Opción 10.

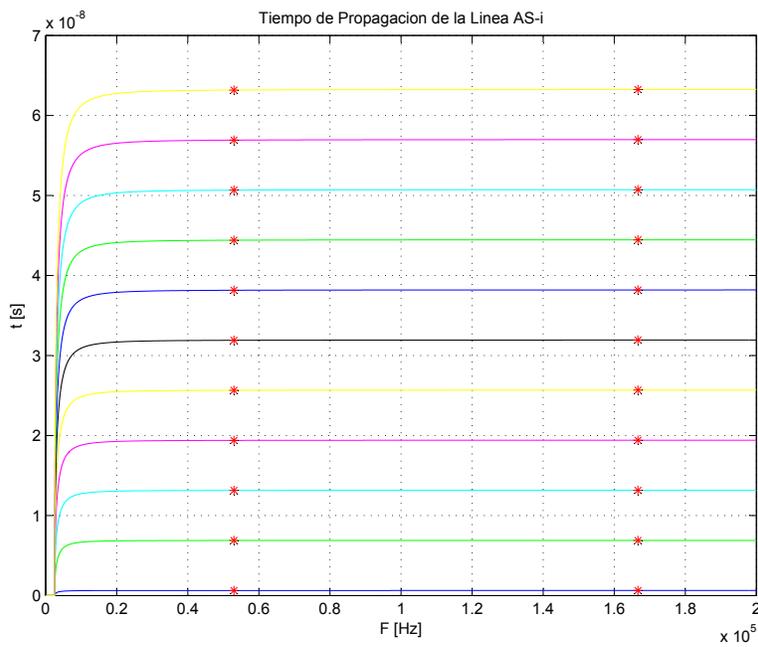


Figura G.7: Resultados de la variación del tiempo de propagación sobre la línea AS-i.



# Bibliografía

- [1] M. Lee, R. Newman, S. Katar, and L. Yonge, “HomePlug 1.0 Powerline Communication LANs - Protocol Description and Performance Results. Version 5.4,” *International Journal of Communication Systems*, vol. 1, no. 6, pp. 1–25, 2000.
- [2] Y. G. O. Bilal, E. Liu and T. Korhonen, “Design of broadband coupling circuits for Power Line Communications,” Helsinki.
- [3] J. Balcells and J. L. Romeral, *Automatas Programables*, 2nd ed. Barcelona: Marcombo Editores, 1997.
- [4] S. Mackay, E. Wright, and J. Park, *Industrial Data Communications*, 3rd ed. Australia: Elsevier Newnes, 2004.
- [5] (2005) Asi international association. [Internet]. Visite: <http://www.as-interface.com>
- [6] *AS-Interface Complete Specification, V3.0, Rev. 0*, I.E.C Std. IEC 62 026-2, September 2004. [Internet]. Visite: <http://www.iec.org>
- [7] G. Held, *Understanding Data Communications*, 3rd ed. Georgia, USA: Jhon Wiley and Sons, 2001.
- [8] B. P. Upender and P. J. Koopman, “Communication Protocols for Embedded Systems,” *Embedded Systems Programming*, vol. 7, no. 11, pp. 46–58, 1994.
- [9] P. Peyton, *Digital Communication Systems*. New Jersey, USA: Prentice Hall, 1987.
- [10] M. Svéda and F. Zezulka, “Interconnecting Low-Level Fieldbusses,” in *Proceedings 23rd Euro-micro’97 Conference IEEE Computer Society*, vol. 35, no. 1, Budapest, 1997, pp. 614–620.
- [11] G. Held, *High Speed Digital Transmission Networking*, 2nd ed. Georgia, USA: Jhon Wiley and Sons, 2000.

- [12] *Profiles Annex A and B to the Complete AS-Interface Specification, V3.0, Rev. 0*, I.E.C Std. IEC 62026-2, September 2004. [Internet]. Visite: <http://www.iec.org>
- [13] M. Gustavsson, J. Wikner, and N.Ñ. Tan, *CMOS Data Converters for Communications*. New York, USA: Kluwer Academics, 2002.
- [14] F. Semiconductor, “MC9S12E-Family Device User Guide,” Motorola, Tech. Rep., 2003. [Internet]. Visite: [www.freescale.com](http://www.freescale.com)
- [15] M. Trnka, “Power Line Communications in Automotive Industry,” Ph.D. dissertation, Czech Technical University in Prague, Republica Checa, 2005.
- [16] C. Inc., “Document 116-1: Wideband RF Transformers. TTWB Series,” Coilcraft Inc, Tech. Rep., 2005. [Internet]. Visite: [www.coilcraft.com](http://www.coilcraft.com)
- [17] A. Corporation, “Rev 5359A-BDC/04. Application Note: Single-to-differential Conversion in High-frequency Applications,” ATMEL Corporation, Tech. Rep., 2004. [Internet]. Visite: [www.atmel.com/literature](http://www.atmel.com/literature)
- [18] R. Reeder, “Transformer-Coupled Front-End for Wideband A/D Converters,” Analog Devices, Tech. Rep., April 2005. [Internet]. Visite: [www.analog.com](http://www.analog.com)
- [19] D. Tsunami, J. McNames, A. Colbert, S. Pearson, and R. Hammerschlag, “Variable Frequency Bioimpedance Instrumentation,” in *26th Annual International Conference of the IEEE EMBS*, San Francisco, CA, 2004, pp. 2386–2389.
- [20] C. J. Poletto and C. L. VanDoren, “A High Voltage, Constant Current Stimulator for Electrocutaneous Stimulation Through Small Electrodes,” *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 8, pp. 929–936, 1999.
- [21] P. Bertemes, B. H. Brown, and A. J. Wilson, “A comparison of modified Howland circuits as current generators with current mirror type circuits,” Department of Medical Physics and Clinical Engineering, 1999.
- [22] A. D. Designers, “AD8012: Dual 350MHz Low Power Amplifier,” Analog Devices Inc., Tech. Rep., 2003. [Internet]. Visite: [www.analog.com](http://www.analog.com)
- [23] —, “AD707:,” Analog Devices Inc., Tech. Rep., 2001. [Internet]. Visite: [www.analog.com](http://www.analog.com)
- [24] —, “ADG884: Analog Multiplexer,” Analog Devices, Tech. Rep., 2005. [Internet]. Visite: [www.analog.com](http://www.analog.com)

- [25] A. Microsystems, “AS2702 AS-Interface Slave IC. Application Note,” Austria Microsystems, Tech. Rep., 2003. [Internet]. Visite: [www.austriamicrosystems.com](http://www.austriamicrosystems.com)
- [26] L. T. Designers, “LT1054: Capacitor Switched Regulator,” Linear Technologies Inc., Tech. Rep., 2005. [Internet]. Visite: [www.linear.com](http://www.linear.com)
- [27] F. J. Ceballos, *C/C++ Curso de Programación*, 2nd ed. Alfaomega, 2002.
- [28] G. Doughman, *Programming the M68HC12 Family*. San Diego, USA: Annabooks, 2000.
- [29] J. Duque, “Implementación de un Procesador de Comunicación Modbus,” Master’s thesis, Universidad Industrial de Santander, Bucaramanga, Colombia, 2005.
- [30] M. Svéda, R. Vrba, F. Zezulka, and P. Legát, “ASI Instrumentation,” *Microprocessing and Microprogramming*, vol. 40, no. 1, pp. 879–882, 1994.
- [31] M. Svéda, F. Zezulka, and R. Vrba, “Coupling Architectures for Low-Level Fieldbusses,” in *Proceedings of the 7th IEEE ECBS’2000 Conference IEEE Computer Society*, vol. 30, no. 7, Edinburgh, Scotland, 2000, pp. 148–155.
- [32] Metrowerks, “CodeWarrior Development Studio IDE 5.5 User’s Guide,” Metrowerks Corporation, Tech. Rep., 2003. [Internet]. Visite: [www.metrowerks.com](http://www.metrowerks.com)
- [33] Motorola, “CPU12RM/AD: HC12 CPU12 Reference Manual Rev. 1.0,” Motorola Inc., Tech. Rep., 2003. [Internet]. Visite: [www.freescale.com](http://www.freescale.com)
- [34] M. Corporation, “Motorola HC12 Assembler,” Metrowerks Corporation, Tech. Rep., 2003. [Internet]. Visite: [www.metrowerks.com](http://www.metrowerks.com)
- [35] O. J. Casellas, “Simulación de los campos de viento de la Península Ibérica y el área geográfica de Catalunya con alta resolución espacial para distintas situaciones meteorológicas típicas.” Ph.D. dissertation, Universitat Politècnica de Catalunya, España, 2005.
- [36] J. Azpiroz and S. Charleston, “Estudio comparativo de cuatro métodos de compresión para imágenes de resonancia magnética nuclear,” Mexico, 1999.
- [37] I. T. Company, “I/O Bus Networks including DeviceNet,” Industrial Text and Video Company, Tech. Rep., 1999. [Internet]. Visite: [www.industrialtext.com](http://www.industrialtext.com)
- [38] K. Mustafa, “Filtering Techniques: Isolating Analog and Digital Power Supplies in TI PLL-Based CDC Devices,” TEXAS Instruments, Tech. Rep., 2001. [Internet]. Visite: [www.ti.com](http://www.ti.com)