

CONVERTIDOR PROFIBUS PA A 4-20mA

T.G 0981

PAVEL MAURICIO DUSSAN GUTIERREZ

ASTRID LILIANA TIRIA SILVA

Proyecto

Director

CARLOS EDUARDO COTRINO BADILLO

Ingeniero Electrónico, M.Sc.

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

BOGOTÁ D.C

2011

CONVERTIDOR PROFIBUS PA A 4-20mA

PAVEL MAURICIO DUSSAN GUTIERREZ

ASTRID LILIANA TIRIA SILVA

Trabajo de grado para optar al título de

Ingeniería Electrónica

Director

Carlos Eduardo Cotrino Badillo

Ingeniero Electrónico, M.Sc.

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

BOGOTÁ D.C

2011

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA



RECTOR MAGNÍFICO: P. JOAQUÍN EMILIO SÁNCHEZ GARCÍA, S.J

DECANO ACADÉMICO: Ing. FRANCISCO JAVIER REBOLLEDO MUÑOZ.

DECANO DEL MEDIO UNIVERSITARIO: P. SERGIO BERNAL S.J.

DIRECTOR DE DEPARTAMENTO: Ing. JORGE LUIS SÁNCHEZ TÉLLEZ.

DIRECTOR DE CARRERA: Ing. JUAN MANUEL CRUZ BOHORQUEZ, M. Ed.

DIRECTOR DEL PROYECTO: Ing. CARLOS EDUARDO COTRINO BADILLO, M.Sc.

AGRADECIMIENTOS

La culminación de este trabajo de grado fue posible gracias a la colaboración y el apoyo incondicional de muchas personas, que contribuyeron con sus conocimientos y acompañamiento para culminar este proyecto con éxito y acorde a nuestras expectativas.

Extendemos nuestros sinceros agradecimientos a nuestro director de trabajo de grado Ing. Carlos Cotrino, nuestras familias y a todas las personas que cooperaron de alguna manera en cada una de las etapas de desarrollo de este proyecto.

También, es de reconocer el apoyo de empresas como SIEMENS que nos permitieron usar en calidad de préstamo sus equipos para realizar pruebas y montajes, necesarios para la puesta en marcha del proyecto.

TABLA DE CONTENIDO

| | |
|---|----|
| 1. INTRODUCCIÓN | 12 |
| 2. MARCO CONCEPTUAL | 14 |
| 2.1 ARQUITECTURA PROFIBUS | 14 |
| 2.2 TECNOLOGÍA DE TRANSMISIÓN..... | 15 |
| 2.2.1 MBP | 15 |
| 2.2.2 CARACTERISTICAS TÉCNICAS DE PROFIBUS PA CON MBP [4]..... | 16 |
| 2.3 CODIFICACION MANCHESTER | 16 |
| 2.4 DISPOSITIVOS PROFIBUS..... | 17 |
| 2.5 FUNCIONALIDAD DE LOS DISPOSITIVOS EN EL PROTOCOLO PROFIBUS PA | 18 |
| 2.6 CARACTERISTICAS DEL PROTOCOLO PROFIBUS PA: | 18 |
| 2.7 TRANSMISION PROFIBUS PA | 20 |
| 2.8 COMUNICACIÓN CICLICA | 25 |
| 2.8.1 DIAGNOSTICO (SAP 60) | 26 |
| 2.8.2 PARAMETRIZACION (SAP 61)..... | 28 |
| 2.8.3 CONFIGURACION (SAP 62)..... | 29 |
| 2.8.4 INTERCAMBIO DE DATOS CICLICOS | 30 |
| 2.9 COMUNICACIÓN ACICLICA | 31 |
| 3. ESPECIFICACIONES | 31 |
| 4. DESCRIPCIÓN GENERAL Y DIAGRAMA EN BLOQUES | 32 |
| 5. COMPONENTES E INSTRUMENTACIÓN | 33 |
| 5.1 MAU (Medium Attachment Unit) - AMIS 49200: | 33 |
| 5.2 UFC (Unified Fieldbus Controller) | 37 |
| 5.3 FLEXIS FREESCALE: | 41 |
| 5.4 CONVERTOR DIGITAL ANALOGO | 41 |
| 5.5 DP/PA LINK..... | 42 |
| 5.5.1 ACOPLADOR DP/PA | 44 |
| 5.6 PLC SIEMENS | 45 |
| 5.7 ANALIZADOR DE PROTOCOLO SOFTING | 45 |
| 6. PRUEBAS Y ANALISIS DE RESULTADOS | 46 |
| 6.1 DESARROLLO HARDWARE: | 46 |
| 6.1.1 CIRCUITO DE IMPLEMENTACION DE LA MAU | 46 |
| 6.1.2 CIRCUITO DE CONVERSIÓN DE VOLTAJE A CORRIENTE | 48 |

| | |
|--|-----|
| 6.2 DESARROLLO SOFTWARE..... | 50 |
| 6.2.1 PROGRAMACION DEL PLC | 50 |
| 6.2.2 PROGRAMACIÓN DEL MICROCONTROLADOR..... | 52 |
| 6.2.3 ARCHIVO GSD..... | 58 |
| 6.2.3.1 CONVENCIONES | 58 |
| 6.2.4. CONEXIÓN DE LA RED PROFIBUS..... | 58 |
| 6.2.5 VERIFICACION DEL CODIGO FUENTE..... | 60 |
| 6.2.5.1. PRUEBA TRANSMISIÓN:..... | 60 |
| 6.2.5.2 ENVIO DE DATO EN CORRIENTE (ONLINE)..... | 65 |
| 6.2.5.3 ANALISIS DE TRAMAS USANDO EL ANALIZADOR DE PROTOCOLO | 67 |
| 6.2.6 DIFERENCIAS ENTRE FI303-SMAR Y CONVERTIDOR PROFIBUS PA A 4-20mA..... | 73 |
| 7. CONCLUSIONES | 74 |
| 8. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN..... | 76 |
| 9. ANEXOS..... | 78 |
| 9.1 CODIGO FUENTE..... | 78 |
| 9.2 PALABRAS CLAVE PARA REVISION DE ARCHIVOS GSD..... | 93 |
| 9.3 ARCHIVO GSD | 99 |
| 9.4 ANEXO CIRCUITO IMPRESO..... | 100 |
| 9.5 LABORATORIO DE COMUNICACIÓN INDUSTRIAL PROFIBUS PA | 103 |

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1. <i>Comunicación industrial PROFIBUS</i> | 14 |
| FIGURA 2. <i>Codificación Manchester</i> | 17 |
| FIGURA 3. <i>Diagrama del sistema de comunicación PROFIBUS</i> | 17 |
| FIGURA 4. <i>Arquitectura del modelo OSI</i> | 19 |
| FIGURA 5. <i>Estructura de la trama de datos PROFIBUS PA</i> | 20 |
| FIGURA 6. <i>Estructura del encabezado de la trama PROFIBUS y unidad de datos</i> | 23 |
| FIGURA 7. <i>Diagrama de comunicación cíclica</i> | 25 |
| FIGURA 8. <i>Trama de solicitud de diagnóstico</i> | 26 |
| FIGURA 9. <i>Trama de respuesta de diagnóstico</i> | 26 |
| FIGURA 10. <i>Trama de parametrización</i> | 28 |
| FIGURA 11. <i>Trama de configuración</i> | 29 |
| FIGURA 12. <i>Convertidor PROFIBUS PA a Corriente</i> | 32 |
| FIGURA 13. <i>Diagrama de bloques detallado-Convertidor PROFIBUS PA a 4-20mA</i> | 33 |
| FIGURA 14. <i>Configuración interna de la MAU</i> | 34 |
| FIGURA 15. <i>Filtro Pasabanda</i> | 34 |
| FIGURA 16. <i>Interfaz MDS</i> | 35 |
| FIGURA 17. <i>Lógica Interfaz MDS</i> | 35 |
| FIGURA 18. <i>Voltajes de referencia</i> | 36 |
| FIGURA 19. <i>Diagrama de Bloques y conexiones de la UFC 100</i> | 37 |
| FIGURA 20. <i>Condición de tiempo entre dos tramas</i> | 38 |
| FIGURA 21. <i>Interfaz con el microcontrolador Freescale de Motorola</i> | 39 |
| FIGURA 22. <i>Modo Nivel - Detección de portadora (RXA) y señal de recepción</i> | 40 |
| FIGURA 23. <i>Tarjeta de desarrollo FLEXIS 8-bit de Motorola</i> | 41 |

| | |
|---|----|
| FIGURA 24. <i>Diagrama De Conexión</i> | 42 |
| FIGURA 25. <i>Integración del DP/PA link al sistema</i> | 43 |
| FIGURA 26. <i>Esquema de conexión general de la red DP/PA</i> | 43 |
| FIGURA 27. <i>Acoplador DP/PA</i> | 44 |
| FIGURA 28. <i>Esquema de comunicación con el programa STEP 7 de SIEMENS.</i> | 45 |
| FIGURA 29. <i>Proceso de verificación de trama PROFIBUS PA</i> | 46 |
| FIGURA 30. <i>Circuito referencial de implementación AMIS 492x0</i> | 47 |
| FIGURA 31. <i>Circuito implementado para generar la corriente de 4-20mA</i> | 49 |
| FIGURA 32. <i>Red PROFIBUS PA implementada en STEP 7</i> | 50 |
| FIGURA 33. <i>Asignación de Parámetros a la Posición de memoria del PLC</i> | 51 |
| FIGURA 34. <i>Bloque STEP 7 para la asignación de dato en corriente</i> | 51 |
| FIGURA 35. <i>Asignacion de puertos Microcontrolador</i> | 52 |
| FIGURA 36. <i>Registros de lectura/escritura de la UFC</i> | 53 |
| FIGURA 37. <i>Diagrama de flujo del programa configurado en el microcontrolador Freescale</i> | 57 |
| FIGURA 38. <i>Arquitectura de la red PROFIBUS Implementada</i> | 59 |
| FIGURA 39. <i>Prueba de conexión de la red online con el transmisor PA</i> | 60 |
| FIGURA 40. <i>Transmisión de datos PROFIBUS PA</i> | 60 |
| FIGURA 41. <i>Secuencia de señal de los bytes de Preámbulo, Delimitador de Inicio y fin</i> | 61 |
| FIGURA 42. <i>PREAMBULO</i> | 62 |
| FIGURA 43. <i>START DELIMITER FÍSICO</i> | 62 |
| FIGURA 44. <i>END DELIMITER FÍSICO</i> | 63 |
| FIGURA 45. <i>START DELIMITER (SD1=10H)</i> | 63 |
| FIGURA 46. <i>START DELIMITER (SD2=68H)</i> | 64 |

| | |
|---|----|
| FIGURA 47. <i>SHORT ACKNOWLEDGEMENT (E5H)</i> | 64 |
| FIGURA 48. <i>SAP 60(3CH)</i> | 65 |
| FIGURA 49. <i>SAP 62(3EH)</i> | 65 |
| FIGURA 50. <i>Comunicación PLC- convertidor interfaz Step7</i> | 65 |
| FIGURA 51. <i>Comunicación PLC- convertidor interfaz Microcontrolador</i> | 66 |
| FIGURA 52. <i>Comunicación PLC- Salida Convertidor.</i> | 66 |
| FIGURA 53. <i>Voltaje vs. Corriente.</i> | 66 |
| FIGURA 54. <i>Corriente Esperada vs. Corriente Medida</i> | 67 |
| FIGURA 55. <i>Estados Comunicación cíclica- Analizador de Protocolo</i> | 68 |
| FIGURA 56. <i>Primer Diagnostico visto en el analizador de Protocolo</i> | 69 |
| FIGURA 57. <i>Trama de Parametrización visto en el Analizador de Protocolo</i> | 70 |
| FIGURA 58. <i>Tramas de Configuración y Respuesta visto en el Analizador de Protocolo</i> | 71 |
| FIGURA 59. <i>Tramas de Segundo Diagnostico y Respuesta visto en el Analizador de Protocolo</i> | 71 |
| FIGURA 60. <i>Intercambio de datos visto en el Analizador de Protocolo</i> | 72 |
| FIGURA 61. <i>Intercambio de datos estable</i> | 72 |
| FIGURA 62. <i>Dispositivos activos en la red PROFIBUS</i> | 73 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1. <i>Características cable PROFIBUS PA</i> | 16 |
| Tabla 2. <i>Tipos De Start Delimiter</i> | 21 |
| Tabla 3. <i>Estructura de FC</i> | 21 |
| Tabla 4. <i>Tipos de FC para solicitud de telegramas</i> | 23 |
| Tabla 5. <i>FC para reconocimiento de telegramas</i> | 23 |
| Tabla 6. <i>Tipos de SAP</i> | 24 |
| Tabla 7. <i>Codificación Delimitador de Inicio y Fin</i> | 24 |
| Tabla 8. <i>Octeto 1 de Diagnostico</i> | 26 |
| Tabla 9. <i>Octeto2 de Diagnostico</i> | 27 |
| Tabla 10. <i>Octeto3 de Diagnostico</i> | 27 |
| Tabla 11. <i>Octeto4 de Diagnostico</i> | 27 |
| Tabla 12. <i>Octeto5 de Diagnostico</i> | 27 |
| Tabla 13. <i>Octeto 6 de Diagnostico</i> | 27 |
| Tabla 14. <i>Octeto 1 de Parametrización</i> | 28 |
| Tabla 15. <i>Octeto 2 de Parametrización</i> | 28 |
| Tabla 16. <i>Octeto 3 de Parametrización</i> | 28 |
| Tabla 17. <i>Octeto 4 de Parametrización</i> | 29 |
| Tabla 18. <i>Octeto 5 de Parametrización</i> | 29 |
| Tabla 19. <i>Octeto 6 de Parametrización</i> | 29 |
| Tabla 20. <i>Octeto 7 de Parametrización</i> | 29 |
| Tabla 21. <i>Octeto de configuración</i> | 30 |
| Tabla 22. <i>Características y Funciones de los canales MSI y MS2</i> | 31 |

| | |
|--|----|
| Tabla 23. <i>Lógica para la transmisión de datos empleada por la MAU</i> | 39 |
| Tabla 24. <i>Lógica para la recepción de datos empleada por la MAU</i> | 40 |
| Tabla 25. <i>Relación De Datos De Entrada Con Voltaje De Salida</i> | 42 |
| Tabla 26. <i>Medición de voltajes en la MAU. Vzener 5.1v vs. Vzener 5.5V</i> | 47 |
| Tabla 27. <i>Diferencias entre FI303-SMAR y Convertidor PROFIBUS PA a 4-20mA</i> | 73 |

1. INTRODUCCIÓN

En la industria muchos componentes como válvulas, actuadores, transmisores, entre otros, operan generalmente de forma descentralizada; por ello, hoy en día se instalan sistemas abiertos e integrados que permiten mejorar los procesos de automatización y comunicación. Por medio de sistemas como estos, las estaciones remotas se comunican a través de un bus de comunicación con los equipos de control ubicados en las diferentes salas de control, para conocer el estado de funcionamiento de la planta y así supervisar y controlar los elementos que la componen. PROFIBUS es un bus de campo diseñado para realizar este tipo de interconexión en áreas de producción industriales de forma homogénea [1].

PROFIBUS es un bus de campo abierto, universal e independiente de proveedores, que tiene una amplia aplicación en procesos de fabricación y automatización. Este estándar es garantizado según las normas EN 50170 y EN 50254. [2]

Debido a la necesidad de integrar el sistema PROFIBUS a dispositivos que trabajan con señales analógicas, el uso de un convertidor de PROFIBUS a corriente es indispensable para adaptar instalaciones existentes a tecnologías modernas; por ello este proyecto tendrá como objetivo principal el diseño y construcción de un convertidor de PROFIBUS PA a 4-20mA, que sirva como interfaz entre sistemas PROFIBUS PA y dispositivos de campo para aplicaciones de automatización de procesos.

PROFIBUS PA es una de las interfaces de usuario de PROFIBUS, que permite la comunicación y alimentación simultánea de dispositivos en campo como transmisores y actuadores. En este proyecto la implementación de esta interfaz se realiza siguiendo la estructura especificada por el protocolo, ya que el convertidor cuenta con un único cable bifilar por el cual se transmiten los datos y se alimenta el sistema [1].

La transmisión de datos del dispositivo maestro al esclavo es uno de los principales enfoques que se muestran en este documento, ya que el proyecto es implementado bajo un protocolo de comunicación específico para sistemas PROFIBUS. Lo anterior permite realizar el montaje de un laboratorio de comunicación industrial que permita a los estudiantes y miembros de la academia en general, familiarizarse con el funcionamiento y las aplicaciones de redes PROFIBUS en la industria.

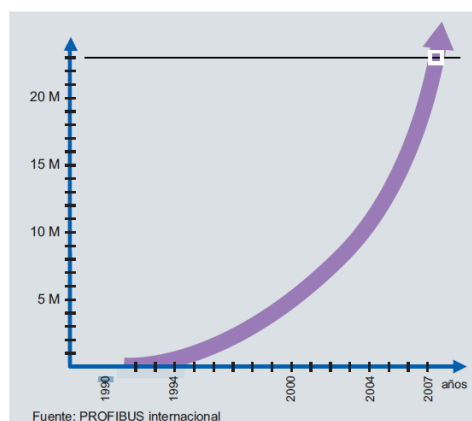
De igual manera en este documento se presenta toda la documentación teórica necesaria para comprender la estructura de la red PROFIBUS y las tramas que la componen. Posteriormente se expone detalladamente tanto el hardware como el software usados en la implementación del convertidor en su integración con la red PROFIBUS.

El objetivo principal del desarrollo de este trabajo consiste en diseñar y construir un convertidor de PROFIBUS a corriente que sirva como interfaz entre sistemas PROFIBUS PA y dispositivos de control para aplicaciones de automatización de procesos. Corroborar que los datos digitales enviados desde un PLC Siemens sean convertidos de forma satisfactoria a datos análogos los cuales podrán ser medidos usando un amperímetro, dentro del rango de conversión establecido.

Adicionalmente, para verificar el cumplimiento de los objetivos mencionados, se hará uso de un PLC SIEMENS (equipo maestro) y un analizador de protocolo SOFTING, los cuales se integrarán con el dispositivo diseñado para monitorear el flujo de tramas y el estado de las estaciones que conforman la red PROFIBUS; de esta manera se podrán mostrar los resultados y realizar un análisis detallado de las tramas obtenidas. Así mismo se realizan mediciones de la corriente obtenida a la salida del convertidor para comprobar que se está transmitiendo el valor que se envía desde el equipo maestro.

Es importante conocer que en las últimas décadas, este protocolo de comunicación ha tomado gran fuerza en aplicaciones industriales, convirtiéndose en un estándar predilecto por los fabricantes a nivel global, lo cual refleja la importancia de la implementación de este proyecto para conocer los sistemas con mayor crecimiento en la actualidad.

En el año 2007 se reportaron 23 millones de nodos PROFIBUS instalados, de los cuales más de 3,3 millones son nodos PROFIBUS PA, como se refleja en la siguiente gráfica:



Nodos PROFIBUS instalados [1]

Lo anterior, permite tener un panorama general de la fuerza que ha tomado este protocolo en la industria y deja ver un gran abanico de posibilidades de automatización de procesos e innovación en la industria [1].

2. MARCO CONCEPTUAL

PROFIBUS (Process Field Bus) es un bus de campo abierto y robusto el cual permite gran variedad de aplicaciones en control de procesos, como la comunicación y automatización de los mismos. Con base en esta premisa, muchos líderes en tecnología de automatización ofrecen interfaces PROFIBUS para sus dispositivos ya que además de estar estandarizado bajo la norma EN 50 170 [1], es un sistema que permite la interacción de equipos de diferentes proveedores. En la figura 1 se muestra el diagrama de comunicación industrial implementando sistemas PROFIBUS. [2]

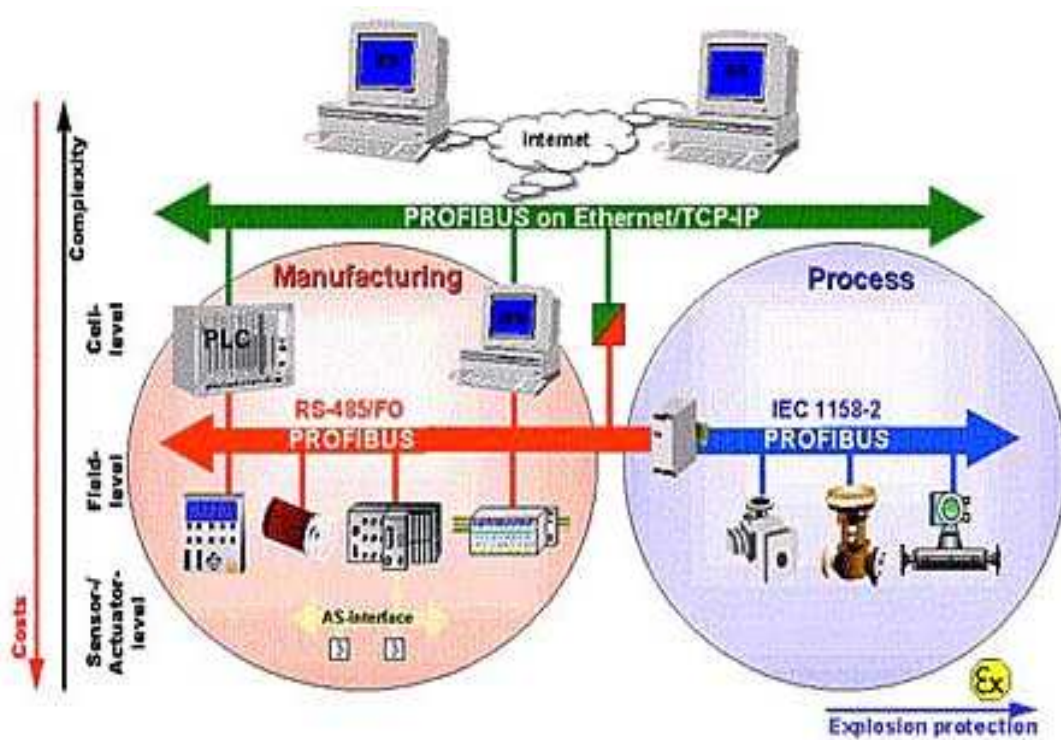


Figura 1. Comunicación industrial PROFIBUS [2]

La tecnología ha tomado un papel decisivo en la automatización de procesos, por lo que la comunicación entre instrumentos y la implementación de mecanismos estandarizados, son herramientas indispensables en el desarrollo tecnológico industrial.

2.1 ARQUITECTURA PROFIBUS

La arquitectura de la red PROFIBUS se divide en tres tipos principales: [2]

PROFIBUS DP: Esta versión de PROFIBUS, ha sido optimizada para una conexión de alta velocidad y está diseñada especialmente para aplicaciones en sistemas de control, de los cuales se destaca el acceso a dispositivos distribuidos de E/S (Entrada-/Salida) y para comunicación entre equipos maestros. PROFIBUS-DP puede emplearse para sustituir la transmisión de señales paralela con 24 V o 0 a 20 mA [3].

PROFIBUS PA: Esta diseñado especialmente para la automatización de procesos. Permite la conexión de sensores y actuadores en una línea de bus común. Además dispone de comunicación de datos y potencia en un mismo bus utilizando una tecnología de dos hilos según el estándar internacional IEC 1158-2. [3]

PROFIBUS PA permite el mantenimiento y la conexión/desconexión de equipos durante la operación, sin afectar otras estaciones en zonas de potencial explosivo. [2] Además, puede usarse para reemplazar el estándar de 4-20mA, aportando grandes ventajas a la utilización de sistemas de bus de campo. Este conjunto de características hacen que esta versión de PROFIBUS, sea ideal para realizar el diseño del convertidor de PROFIBUS PA a corriente, pues permite ver la interacción desde un dispositivo maestro a un dispositivo esclavo pasando por toda la red PROFIBUS (PROFIBUS DP y PROFIBUS PA).

PROFIBUS-FMS: Es la solución de aplicación general para tareas de comunicación de célula. Los eficaces servicios FMS aportan una amplia gama de aplicaciones y proporcionan flexibilidad. PROFIBUS FMS también se puede utilizar para tareas de comunicación complejas de gran alcance. [3]

2.2 TECNOLOGÍA DE TRANSMISIÓN

2.2.1 MBP

Debido a los requerimientos específicos de procesos en los cuales se hace uso de dispositivos de campo compactos, la tecnología de transmisión RS485 no es aplicable para PROFIBUS PA. Por lo tanto, MBP (Manchester Bus Powered) especifica un estándar de buses de campo que satisface los requerimientos de PROFIBUS PA y en el cual la capa física (**PhL-physical layer**) ofrece las siguientes funciones adicionales [4]

- Formato de transmisión síncrono
- Extensión del formato de trama que incluye la capa física (información de control específica).
- Chequeo de bloques cíclico. (cyclic redundancy check-CRC).

2.2.2 CARACTERÍSTICAS TÉCNICAS DE PROFIBUS PA CON MBP [4]

1. Estructura del Bus: Línea terminada con o sin líneas troncales y ramas (árbol).
2. Medio de transmisión: MBP
3. Tasa de transmisión: 31.25 Kbit/s
4. Redundancia: A través de un segundo medio
5. Método de Transmisión: Código Manchester, síncrono.
6. Cable Bifilar trenzado para sistemas de bus de campo según IEC 61158-2. Las principales características de este cable se muestran en la Tabla 1.

| FC PROCESS CABLE | |
|---|---|
| Atenuación • a 38,4 kHz | ≤ 3 dB/km |
| Inductancia | 0,7 mH/km |
| Resistencia natural • a 38,4 kHz | $100 \pm 20 \Omega$ |
| Valor nominal | 100 Ω |
| Capacidad de servicio a 1 kHz | aprox. 90 nF/km |
| Condiciones ambientales admitidas • Temperatura de servicio • Temperatura. Transporte y almacenamiento. • Temperatura de tendido | -40 °C a $+80$ °C -40 °C a $+80$ °C -20 °C a $+80$ °C |
| Radios de flexión • Flexión una vez • Flexión varias veces | ≥ 60 mm ≥ 120 mm |

Tabla 1. Características cable PROFIBUS PA [5]

2.3 CODIFICACION MANCHESTER

Como se mencionó anteriormente, PROFIBUS PA se basa en la norma IEC 1158-2, la cual define las reglas de transmisión según el código Manchester. La codificación Manchester define que cada período de un bit se divide en dos intervalos iguales. Un uno `1` se indica mediante una transición de nivel alto a nivel bajo en el centro del bit y un cero `0` se representa mediante una transición de nivel bajo a nivel alto. Este esquema asegura que todos los bits presentan una transición en la parte media, aún si el dato es un cero o un uno constante, proporcionando así un excelente sincronismo. Sin embargo, este tipo de transmisión necesita un ancho de banda significativamente alto. [6] [7] En la figura 2 se muestra un ejemplo de la codificación Manchester.

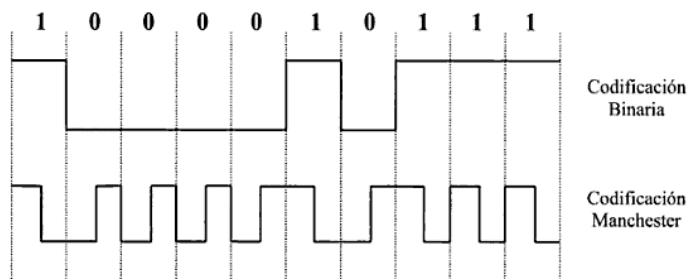


Figura 2. CODIFICACIÓN MANCHESTER [6]

2.4 DISPOSITIVOS PROFIBUS

El protocolo PROFIBUS establece las normas de comunicación desde el nivel de enlace hasta el nivel de aplicación. Esto lo hace distinguiendo entre elementos Maestro y Esclavo. Los dispositivos Maestro determinan la comunicación de datos en el bus, por lo cual requieren acceso activo al bus para obtener los valores del proceso. Los dispositivos esclavos son dispositivos periféricos (válvulas, transmisores de medida, etc.), los cuales son pasivos con respecto a la comunicación y sólo responden a una solicitud directa. En la Figura 3 se puede observar que el maestro PROFIBUS es parte del controlador mientras los sensores y actuadores son los esclavos PROFIBUS. [4]

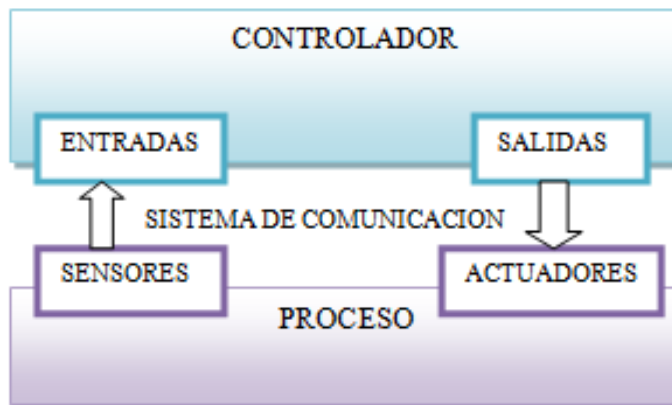


Figura 3. Diagrama del sistema de comunicación PROFIBUS. [4]

El dispositivo maestro PROFIBUS se divide en dos clases [2]:

Maestro Clase 1: encargado de las operaciones cíclicas (lecturas y escrituras) y control de circuitos abiertos, es decir, solicita periódicamente los valores medidos del proceso desde los sensores y envía la salida de control a los actuadores. [4]

Maestro Clase 2: encargado de los accesos acíclicos de parámetros y funciones de equipos PA, por lo cual no requiere estar permanentemente conectado al sistema PROFIBUS. [4]

La comunicación cíclica toma lugar entre un maestro DP clase 1 y un esclavo DP asignado a éste, mientras que la comunicación acíclica ocurre en un maestro clase 2.

El protocolo ofrece dos tipos de mensajes básicos: cíclicos y acíclicos. Los mensajes cíclicos permiten el intercambio de datos de baja prioridad y por tanto no son críticos en cuanto al tiempo de respuesta, y los mensajes acíclicos permiten acortar el tiempo de respuesta de los datos críticos.[8]

2.5 FUNCIONALIDAD DE LOS DISPOSITIVOS EN EL PROTOCOLO PROFIBUS PA

PROFIBUS incluye algunas extensiones compatibles las cuales son características de PROFIBUS PA como son [4]:

DPVO Protocol Funcionality: Los dispositivos PROFIBUS PA con comunicación cíclica para todos los esclavos PROFIBUS como el implementado en este proyecto, cuentan con otras características tales como: formato de datos para tráfico de información cíclica especificado en 5 bytes de entrada por módulo (4 bytes por valor de medida más 1 byte de estado), identificación de bytes que son reflejados en la configuración de trama (CHK_CFG), elaboración específica de parámetros de asignación de datos (USER_PRM_DATA), los cuales son especificados por el parámetro de asignación de trama (SET_PRM).

DPV1 Protocol Funcionality: El perfil PROFIBUS PA puede soportar al menos un canal MS2 (canal con capacidad de comunicación acíclica); esto es necesario para el uso de servicios de lectura y escritura acíclica de los parámetros definidos en PROFIBUS PA.

DPV2 Protocol Funcionality: Las funcionalidades añadidas bajo DPV2 no están incorporadas a dispositivos PROFIBUS PA.

I&M Funtions: Todos los dispositivos PROFIBUS que soportan comunicación acíclica están equipados con identificación y funciones de mantenimiento.

2.6 CARACTERISTICAS DEL PROTOCOLO PROFIBUS PA:

PROFIBUS PA usa las funciones básicas de PROFIBUS DP para la transmisión de valores de medida y estado, y las funciones extendidas del mismo para la parametrización y operación de los dispositivos de campo. [8]

El protocolo PROFIBUS PA está formado por: [2][9]

- La **Capa 1 o Capa física** define las características de la transmisión, es decir, se ocupa de la transmisión de bits a lo largo de un canal de comunicación. En esta capa se encuentran los medios materiales para la comunicación como lo son conectores, cables, etc.

- La **Capa 2 o Capa de Enlace (FDL – Fieldbus Data Link)** define el protocolo de acceso al bus. Por medio de esta capa es posible llevar a cabo una comunicación sin errores, pues se encarga de solucionar los problemas de reenvío, o mensajes duplicados cuando hay destrucción de tramas.

- La **Capa 7 o Capa de aplicación** define las funciones de aplicación y los protocolos implementados para el intercambio de datos.

En la figura 4 se presenta la arquitectura del modelo OSI, resaltando las capas empleadas en el protocolo PROFIBUS PA, descritas anteriormente.



Figura 4. Arquitectura del modelo OSI

Las demás capas de este modelo no hacen parte de la red PROFIBUS PA, ya que se refieren a dispositivos enrutadores de datos y de control de otros equipos, e información que no hace parte del tratamiento de datos que se emplea en el desarrollo de este proyecto.

2.7 TRANSMISION PROFIBUS PA

En la transmisión de datos con PROFIBUS PA, cada mensaje se compone de una señal previa (Preámbulo), una señal delimitadora de arranque (start delimiter) y una señal delimitadora de fin (end delimiter), e intercala entre éstas dos últimas una señal de datos de longitud máxima de 256 bytes que contiene la dirección del instrumento, su identificación, los valores de medida, etc.[4]. En la figura 5 se muestra la estructura de la trama PROFIBUS PA. [8]

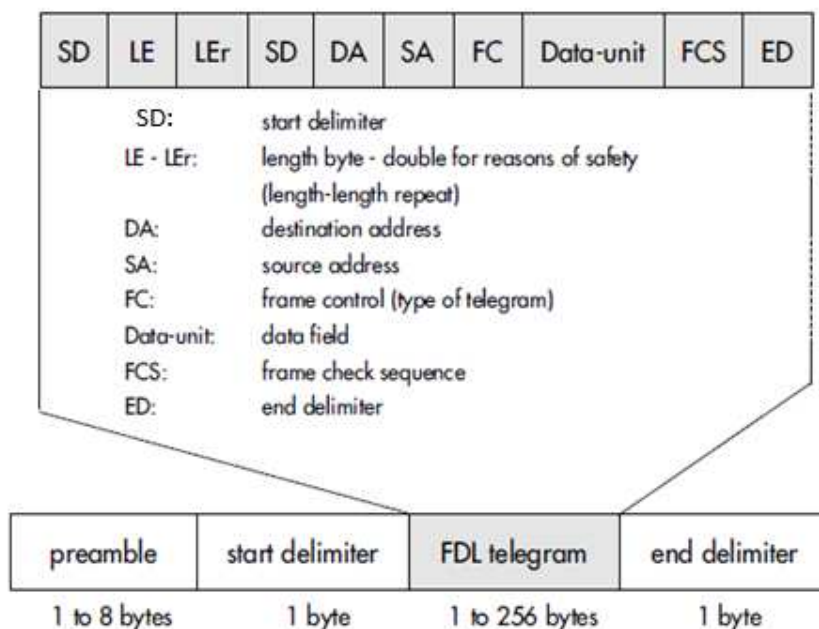


Figura 5. Estructura de la trama de datos PROFIBUS PA [10]

PREÁMBULO: Para recibir o transmitir el mensaje síncrono, primero se debe enviar un preámbulo, el cual consiste de secuencias de bits con valores de 0/1 (1 a 8 Bytes).[4]

CAPA FÍSICA- DELIMITADOR DE INICIO (PhL-SD): La siguiente secuencia de símbolos muestra el PhL-SD: 1, N+,N-,1,0,N-,N+,0. Esta secuencia precede el PhSDU (Physical Layer Service Data Unit=FDL frame+CRC) [4].

DELIMITADOR DE INICIO (SD): Identifica el inicio de una trama y su formato general. PROFIBUS emplea cuatro tipos de SD para tramas de solicitud y respuesta, además de una quinta para un reconocimiento corto (short acknowledgement), como se muestra en la tabla 2.

| FORMATO | VALOR | LONGITUD DE DATOS |
|---------|-------|----------------------------|
| SD1 | 0x10 | 0 bytes |
| SD2 | 0x68 | 1-32 bytes |
| SD3 | 0xA2 | 8 bytes |
| SD4 | 0xDC | 0 bytes |
| SC | 0xE5 | 0 bytes, Short Acknowledge |

Tabla 2. Tipos De Start Delimiter [11]

LE/LEr (Length/ Length Repeated): Indica la longitud de un telegrama incluyendo los bytes correspondientes a los campos DA, SA, FC y DU (Data Unit). LEr es igual a LE y corresponde a un campo de protección de datos redundantes. Transmite un máximo de 246 bytes en el campo DU, por lo cual este campo puede variar entre 4 y 249 bytes.

DA/SA (Destination Address/Source Address): El dispositivo maestro se direcciona a un esclavo específico mediante 8 bits que pueden tener valores válidos desde 0-127 (00H-7FH), donde las direcciones 126 y 127 están reservadas para propósitos de puesta en marcha y reconocimiento de la red. [11]

FC (Function Code or Frame Control): Especifica el tipo de telegrama (petición, respuesta, reconocimiento), tipo de estación (pasiva/activa, maestro/esclavo), prioridad y reconocimiento de telegrama (exitoso o no exitoso). En la tabla 3, se muestra la estructura de este campo [11]:

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3-Bit 0 |
|-------|-------|-------|-------|--------------------|
| RES | FRAME | FCB | FCV | FunctionCodeNumber |

Tabla 3. Estructura de FC [11].

Bit 7-RES: Puede tomar cualquier valor entre 0 y 1.

Bit 6-FRAME: Indica el tipo de telegrama; así si este bit es 0 el tipo de mensaje es Acknowledge/Response y si es 1 es Send/Request.

Bit 5- FCB (Frame Count bit): Toma valores entre 0 y 1 de forma alterna, para luego comparar el valor de la trama anterior y detectar si se repitió el mensaje.

Bit 4-FCV (Frame Count bit Valid): Cuando este campo toma el valor de 0 la función variante de FCB es inválida ya que indica el primer mensaje enviado, mientras que si este valor es 1 la función variante de FCB es válida.

Bit 3 a Bit 0-Function code Number:

PROFIBUS utiliza los siguientes servicios de transmisión en el envío de tramas [11]:

- SRD (Send and Request Data with acknowledge): Con este servicio el maestro envía los datos de salida a los esclavos y recibe los datos de entrada desde los esclavos en un solo ciclo de trama, lo cual hace eficiente el intercambio de datos entre dispositivos de Entrada/ Salida.
- SDN (Send Data with No acknowledge): Se utiliza cuando un mensaje debe ser enviado a un grupo de esclavos (multi-cast) o a todos los esclavos (broadcast) al mismo tiempo. El usuario recibe la confirmación de la transmisión pero no reconoce si el dato que recibió es válido.
- SRD (Send and Request Data with Reply): Permite al usuario transferir y solicitar datos hacia o desde una sola estación remota. Si algún error ocurre en la transferencia, ésta se repite.

Por otro lado el usuario está en capacidad de solicitar datos al usuario remoto sin enviar datos (mensaje sin información). En la tabla 4 y la tabla 5 se muestran los tipos de FC para solicitud y reconocimiento de telegramas, respectivamente.

| Bit 3 a Bit 0- FC CODE | FUNCIÓN(FC=1-MSB) |
|------------------------|--|
| 0x04 | SND low (Envia datos sin reconocimiento) |
| 0x06 | SND high (Envia datos sin reconocimiento) |
| 0x07 | Reservado/(Solicitud de datos de diagnóstico) |
| 0x09 | Solicitud de estado FDL con respuesta |
| 0xC | SRD low (Enviar y solicitar datos con reconocimiento) |
| 0xD | SRD high (Enviar y solicitar datos con reconocimiento) |
| 0xE | Solicitud de ID con respuesta |
| 0xF | Solicitud de estado LSAP con respuesta |

Tabla 4. Tipos de FC para solicitud de telegramas [11]

| Bit 3- Bit 0- FC CODE | FUNCIÓN(FC=0-MSB) |
|-----------------------|---|
| 0x00 | ACK positivo |
| 0x01 | ACK negativo (User Error, Interface Error) |
| 0x02 | ACK negativo (No hay recursos/ espacio en memoria para enviar datos (RR)) |
| 0x03 | ACK Negativo (Servicio no activado, SSAP no activado) |
| 0x08 | Respuesta FDL de baja prioridad y envío de datos OK |
| 0x09 | ACK negativo (No respuesta FDL y envío de datos OK) |
| 0xA | Respuesta FDL de alta prioridad y envío de datos OK |
| 0xC | Respuesta FDL de baja prioridad, No hay recursos para el envío de datos |
| 0xD | Respuesta FDL de alta prioridad (recursos para envío de datos) |

Tabla 5. FC para reconocimiento de telegramas [11]

SERVICE ACCESS POINTS (SSAP & DSAP): En la Figura 6, se muestra la estructura de la trama PROFIBUS, incluyendo los bytes de SSAP y DSAP.

| | | | | | | | | | | | |
|----|----|-----|----|----|----|----|------|------|-----|-----|----|
| SD | LE | LEr | SD | DA | SA | FC | DSAP | SSAP | DU | FCS | ED |
| 1b | 1b | 1b | 1b | 1b | 1b | 1b | 1b | 1b | var | 1b | 1b |

Figura 6. Estructura del encabezado de la trama PROFIBUS y unidad de datos [11]

Los intercambios de datos se manejan en el encabezado del telegrama mediante los SAP's (Service Access Points), que indican cuales datos van a ser transmitidos o cual función se debe llevar a cabo. Sólo los telegramas que incluyen campos de datos usan los bytes DSAP y SSAP (Destination Service Access Point & Source Service Access Point). En la tabla 6 se muestran los tipos de SAP empleados en PROFIBUS y los servicios:

| SAP | SERVICE |
|---------------|---|
| Default SAP=0 | Cyclical Data Exchange (Write Read Data) |
| SAP54 | Master-to-Master SAP (M-M Communication) |
| SAP55 | Change Station Address (Set Slave Add) |
| SAP56 | Read Inputs (Rd Inp) |
| SAP57 | Read Outputs (Rd Outp) |
| SAP58 | Control Commands to a DP Slave (Global Control) |
| SAP59 | Read Configuration Data (Get Cfg) |
| SAP60 | Read Diagnostic Data (Slave Diagnosis) |
| SAP61 | Send Parameterization Data (Set Prm) |
| SAP62 | Check Configuration Data (Chk Cfg) |

Tabla 6. Tipos de SAP [11]

DU (Data Unit): Este campo contiene los datos de la estación de destino DA (datos de solicitud) o de fuente SA (datos de respuesta). Generalmente este campo es limitado a 32 bytes, pero el estándar permite hasta 245 bytes.

En telegramas con longitud de dato variable, el tamaño puede variar entre 1 y 246 bytes y para telegramas con longitud de dato fija, el tamaño es de 8 bytes.

FCS (Frame Check Sequence): Contiene la secuencia de verificación de tramas desde la dirección de destino DA hasta la unidad de datos DU.

CAPA FÍSICA- DELIMITADOR DE FIN (PhL-ED): La secuencia de símbolos que describen el PhL-ED es: 1, N+, N-, N+, N-, 1, 0, 1. Esta secuencia es enviada inmediatamente después del PhSDU. (Physical Service Data Unit). [4]

En la tabla 7 se muestran las reglas de codificación para el delimitador de inicio y de fin descritos anteriormente [4].

| Symbols | Encoding |
|---------------------|-----------------------------|
| 1 (ONE) | Hi- Lo transition (mid-bit) |
| 0 (ZERO) | Lo-Hi transition (mid-bit) |
| N+ (NON- DATA PLUS) | Hi (No tansition) |
| N- (NON-DATA MINUS) | Lo (No transition) |

Tabla 7. Codificación Delimitador de Inicio y Fin. [4]

2.8 COMUNICACIÓN CICLICA

La comunicación cíclica se efectúa con un único maestro para evitar datos erróneos; de esta manera, para sincronizar el ciclo de transmisión y recepción, y conocer el estado inicial del maestro, es necesaria la FSPMM1 (Fieldbus Service Protocol Mapping Machine of Master Class 1). El estado de esta máquina se establece por cada esclavo y maneja la secuencia de instrucciones, como se explica a continuación [4].

En el primer ciclo se especifica el SAP (Service Access Point) para realizar el diagnóstico del esclavo. Luego, basado en la respuesta de diagnóstico, el maestro detecta el estado de esclavo, ya que este puede estar siendo controlado por otro maestro y en tal caso debe esperar para el envío de parámetros. En el momento que se puede realizar el envío de parámetros se verifica que estos sean compatibles y que el esclavo pueda aceptarlos [4].

Continuando con el ciclo, el maestro envía los datos de configuración al esclavo para verificar la conexión, la cual se responde por el esclavo en el siguiente ciclo con un segundo diagnóstico. En el caso que esta respuesta sea positiva el esclavo está listo para el intercambio de datos cíclico con el maestro. En caso contrario, se envía de nuevo el primer diagnóstico para iniciar un nuevo ciclo. La figura 7 muestra el esquema de comunicación cíclico explicado [4].

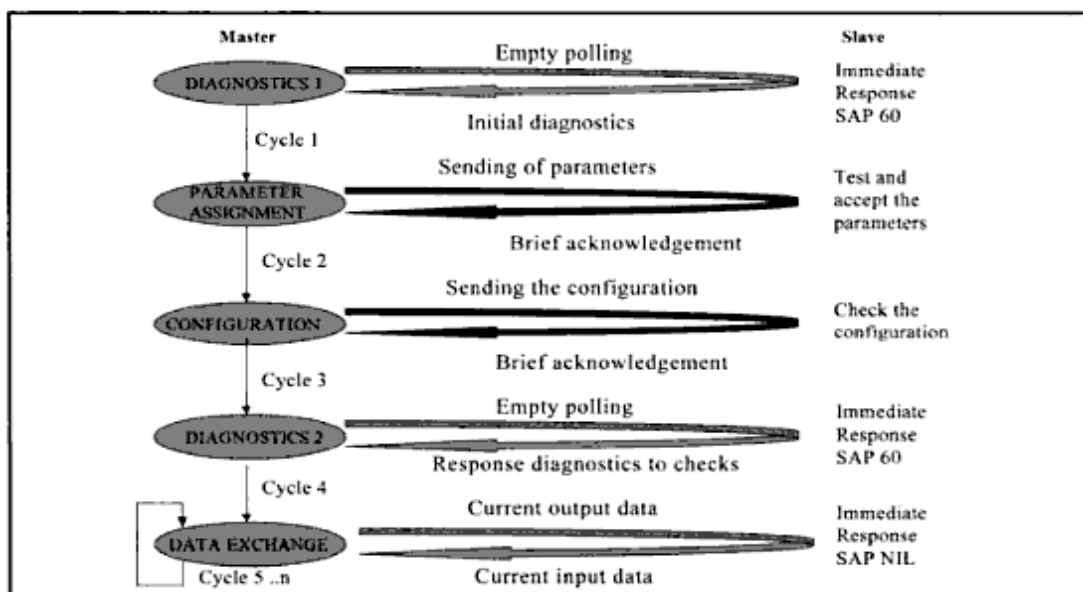


Figura 7. Diagrama de comunicación cíclica [4]

2.8.1 DIAGNOSTICO (SAP 60)

Un maestro solicita datos de diagnóstico antes de enviar la trama de parametrización y después de la configuración, antes de asumir el modo de intercambio de datos cíclico con el esclavo. El maestro evalúa la información de diagnóstico para verificar si la parametrización y configuración es correcta. La figura 8 muestra la trama de solicitud de diagnóstico enviada por el maestro, mientras que la figura 9 muestra la trama de respuesta de diagnóstico enviada por el esclavo.

| SD2 | LE | LEr | SD | DA | SA | FC | DSAP | SSAP | FCS | ED |
|-----|----|-----|-----|----|----|----|----------|----------|-----|-----|
| 68H | X | X | 68H | 8x | 8x | X | 3CH (60) | 3EH (62) | X | 16H |

Figura 8. Trama de solicitud de diagnóstico[11]

| SD | LE | LEr | SD | DA | SA | FC | DSAP | SSAP | DU | FCS | ED |
|-----|----|-----|-----|----|----|----|----------|----------|-----|-----|-----|
| 68H | X | X | 68H | 8x | 8x | X | 3EH (62) | 3CH (60) | X.. | X | 16H |

Figura 9. Trama de respuesta de diagnóstico[11]

El diagnóstico tiene varias funciones y es subdividido en seis octetos de APDU, los cuales brindan información sobre el estado de la conexión cíclica y el diagnóstico específico del usuario, el cual es utilizado para mostrar estados especiales durante la transferencia de datos [4], como se muestra en las tablas 8-13.

| OCTETO 1 | |
|----------|---|
| BIT | DIAGNOSTICO |
| 0 | Station_Non_Existent: Activado por el maestro si el esclavo no es detectado en la red |
| 1 | Station_Not_Ready: Activado por el esclavo si no está listo para transferencia de datos |
| 2 | Cfg_Fault: Activado por el esclavo si los datos de configuración enviados por el maestro no concuerdan con su configuración interna |
| 3 | Ext_Diag: Activado por el esclavo para indicar que existen datos de diagnóstico extendidos |
| 4 | Not_Supported: Activado por el esclavo si el servicio no es soportado |
| 5 | Invalid_Slave_Response: Activado por el maestro si no recibe una respuesta correcta desde el esclavo |
| 6 | Prm_Fault: Activado por el esclavo si los datos de parametrización enviados por el maestro no son correctos |
| 7 | Master_Lock: Activado por el maestro para indicar que el esclavo ha sido parametrizado por otro maestro |

Tabla 8. Octeto 1 de Diagnostico [11]

| OCTETO 2 | |
|----------|---|
| BIT | DIAGNOSTICO |
| 0 | Prm_Req: Activado por el esclavo si necesita ser parametrizado |
| 1 | Stat_Diag: Activado por el esclavo para hacer que el maestro solicite información de diagnóstico hasta que este bit sea desactivado |
| 2 | El esclavo activa este bit en 1 |
| 3 | WD_ON: Activado por el esclavo para indicar que el watchdog está activo |
| 4 | Freeze_Mode: Activado por el esclavo después de haber recibido un comando de Freeze desde el maestro |
| 5 | Sync_Mode: Activado por el esclavo después de haber recibido un comando de Sync desde el maestro |
| 6 | Reservado |
| 7 | Deactivated: Activado por el maestro si el esclavo se encuentra inactivo y es retirado del intercambio de datos cíclico |

Tabla 9. Octeto 2 de Diagnostico [11]

| OCTETO 3 | |
|----------|---|
| BIT | DIAGNOSTICO |
| 0-6 | Reservados |
| 7 | Ext_Diag_Overflow: Activado si existe más información de diagnóstico que la especificada en Ext_Diag_Data |

Tabla 10. Octeto 3 de Diagnostico [11]

| OCTETO 4 | |
|----------|--|
| BIT | DIAGNOSTICO |
| 0-7 | Master_Add: Dirección del maestro que parametrizó el esclavo. Si ningún maestro ha parametrizado el esclavo, se inserta el valor 255 (FF si parametrización) |

Tabla 11. Octeto 4 de Diagnostico [11]

| OCTETO 5 | |
|----------|--|
| BIT | DIAGNOSTICO |
| 0-7 | Byte alto del número de identificación del fabricante para identificación y verificación |

Tabla 12. Octeto 5 de Diagnostico [11]

| OCTETO 6 | |
|----------|--|
| BIT | DIAGNOSTICO |
| 0-7 | Byte bajo del número de identificación del fabricante para identificación y verificación |

Tabla 13. Octeto 6 de Diagnostico [11]

2.8.2 PARAMETRIZACION (SAP 61)

Como se explicó anteriormente, luego de realizar el diagnóstico, se procede a asignar los parámetros, los cuales son definidos mediante el archivo GSD. La figura 10 muestra la trama de parametrización enviada por el maestro.

| | | | | | | | | | | | |
|-----|----|-----|----|----|----|----|-------------|-------------|-----|-----|-----|
| SD | LE | LEr | SD | DA | SA | FC | DSAP | SSAP | DU | FCS | ED |
| 68H | X | X | X | 8x | 8x | X | 3DH (61) | 3EH (62) | X.. | X | 16H |

Figura 10. Trama de parametrización [11]

Esta función se utiliza para establecer los parámetros de un esclavo en cualquier momento excepto en el modo de intercambio de datos. La trama de parametrización contiene 7 octetos de información requerida por el estándar PROFIBUS, como se muestra en las Tablas 14-20.

| OCTETO 1 | |
|----------|--|
| BIT | PARAMETRIZACIÓN |
| 0 | Reservado. En cero |
| 1 | Reservado. En cero |
| 2 | Reservado. En cero |
| 3 | WD_On: En 1 activa el control de watchdog |
| 4 | Freeze_Req: En 1, el esclavo operará en modo Freeze. |
| 5 | Sync_Req: En 1, el esclavo operará en modo Sync |
| 6 | Unlock_Req: En 1, el esclavo es liberado para otros maestros |
| 7 | Lock_Req: En 1, el esclavo es bloqueado para otros maestros |

Tabla 14. Octeto 1 de Parametrización [11]

| OCTETO 2 | |
|----------|--------------------|
| BIT | PARAMETRIZACIÓN |
| 0-7 | Valor de WD_Fact_1 |

Tabla 15. Octeto 2 de Parametrización [11]

| OCTETO 3 | |
|----------|--------------------|
| BIT | PARAMETRIZACIÓN |
| 0-7 | Valor de WD_Fact_2 |

Tabla 16. Octeto 3 de Parametrización [11]

| OCTETO 4 | |
|----------|---|
| BIT | PARAMETRIZACIÓN |
| 0-7 | Valor del mínimo tiempo de retardo Tsdr min |

Tabla 17. Octeto 4 de Parametrización [11]

| OCTETO 5 | |
|----------|--|
| BIT | PARAMETRIZACIÓN |
| 0-7 | Byte alto del número de identificación del esclavo |

Tabla 18. Octeto 5 de Parametrización [11]

| OCTETO 6 | |
|----------|--|
| BIT | PARAMETRIZACIÓN |
| 0-7 | Byte bajo del número de identificación del esclavo |

Tabla 19. Octeto 6 de Parametrización [11]

| OCTETO 7 | |
|----------|--|
| BIT | PARAMETRIZACIÓN |
| 0-7 | Valor de identificación de grupo del esclavo |

Tabla 20. Octeto 7 de Parametrización [11]

La trama de parametrización es respondida con un reconocimiento corto (short acknowledgement).

2.8.3 CONFIGURACION (SAP 62)

Después de la parametrización, el esclavo espera la trama de configuración que especifica el número de bytes de entrada y salida que serán intercambiados en cada ciclo de trama. Además, hace que el esclavo compare la configuración que le fue enviada con la configuración almacenada. La figura 11 muestra la trama de configuración enviada por el maestro.

| | | | | | | | | | | | |
|-----|----|-----|----|----|----|----|-------------|------------|-----|-----|-----|
| SD | LE | LEr | SD | DA | SA | FC | DSAP | SSAP | DU | FCS | ED |
| 68H | X | X | X | 8x | 8x | X | 62 (3EH) | 62 (3E) | X.. | X | 16H |

Figura 11. Trama de configuración [11]

Después que los parámetros son asignados, el esclavo es configurado por el maestro. En esta configuración se especifica la cantidad y la estructura de las entradas y salidas cíclicas correspondientes a cada uno de los módulos [4].

En la Tabla 21 se muestra el dato de configuración enviado por el maestro.

| BIT | CONFIGURACION |
|------------|---|
| 0-3 | Longitud del dato 0000=1 byte 1111=16 bytes |
| 4-5 | Entrada/Salida 00=Formato especial 01=Entrada 10=Salida 11=Entrada y salida |
| 6 | 0=Byte 1=Word |
| 7 | 0=Consistencia del Byte/Word 1=Consistencia de toda la longitud |

Tabla 21. Octeto de configuración [11]

La trama de configuración es respondida con un reconocimiento corto (short acknowledgement).

2.8.4 INTERCAMBIO DE DATOS CICLICOS

Al llevar a cabo la asignación de parámetros, la configuración y el segundo diagnóstico, se ejecuta la conexión cíclica entre el maestro y el esclavo. El maestro envía cíclicamente datos de salida al esclavo y recibe datos de entrada en respuesta (si se requiere). Si el esclavo es un dispositivo de salida (sin datos de entrada a enviar), debe responder con un reconocimiento corto (short acknowledgement).

A diferencia de otros tipos de trama las cuales contienen 11 bytes de información de encabezado, la trama de intercambio de datos tiene sólo 9, siendo los bytes de DSAP y SSAP retirados. La unidad de datos (DU) contiene desde 1 hasta 244 bytes a ser transferidos (de salida y/o de entrada).

2.9 COMUNICACIÓN ACICLICA

Tiene dos canales de comunicación, MS1 y MS2. En la tabla 22 se muestran las características principales de cada canal y su función.[4]

| CANAL | FUNCIÓN | CONDICIONES |
|-------|--|--|
| MS1 | Realiza intercambio de datos cíclico | El esclavo puede intercambiar datos de forma cíclica solo con un maestro al mismo tiempo |
| MS2 | Funciones de lectura y escritura (Ajustes del dispositivo) | Un esclavo puede establecer múltiples conexiones con uno o más maestros al mismo tiempo |

Tabla 22. Características y Funciones de los canales MS1 y MS2

El SAP (Service Acces Point) especifica cual canal es usado para leer o escribir datos, sin embargo hay ciertas condiciones para inicializar y finalizar la transmisión en cada canal, en caso del MS1 el estado del dispositivo esclavo debe ser DATA_EXCH, con lo cual el dispositivo es capaz de intercambiar la información de forma cíclica con el maestro. La conexión del MS2 es establecida a través de otro parámetro de requerimiento, el cual es enviado al DSAP (Destination Service Access Point) [4]

3. ESPECIFICACIONES

- Comunicación digital, PROFIBUS PA, según la norma IEC 1158-2 (H1): 31.25 Kbit/s y alimentación a través del mismo bus.
- Salida de corriente 4-20mA
- Alimentación eléctrica a través de PROFIBUS PA, 24Vdc±10%
- El convertidor no va a incluir seguridad intrínseca, según las normas FISCO.
- El convertidor no tendrá encerramiento industrial.

4. DESCRIPCIÓN GENERAL Y DIAGRAMA EN BLOQUES

El proyecto consiste en la realización de un convertidor PROFIBUS PA a 4-20mA que sirva como interfaz entre el protocolo y dispositivos de campo como actuadores; para esto se tendrá en cuenta la norma IEC 61158/EN 50 170, la cual establece las reglas de transmisión del sistema PROFIBUS y permite que los dispositivos de campo se puedan conectar.

El dispositivo diseñado consta de 3 bloques principales: un bloque de desacoplamiento físico, un bloque de enlace y manipulación de datos y un bloque de conversión digital/analógica. La figura 12, muestra el diagrama de bloques general.



Figura 12. Convertidor PROFIBUS PA a Corriente

En el bloque de desacoplamiento físico se tiene como entrada un cable bifilar el cual consta del bus de datos y la alimentación, proveniente de la red PROFIBUS PA. La función de este bloque es permitir el manejo de datos y alimentación de forma independiente, de tal forma que se facilite la manipulación de las tramas y dispositivos de acuerdo con el protocolo PROFIBUS PA. A la salida de este bloque se tienen las señales de transmisión y recepción de datos y la alimentación del sistema por separado.

En el bloque de enlace y manipulación de datos se tratan las señales de transmisión y recepción del sistema de acuerdo con el protocolo PROFIBUS PA. Su función principal es gestionar la operación del sistema a partir de instrucciones programadas que controlan el proceso de adquisición de datos, tiempos de espera y ajuste de tramas para trabajar con el conversor digital/analógico.

El conversor Digital-Analógico recibe los datos procesados por el bloque de enlace y manipulación de datos y la fuente de alimentación. Su función es tomar los paquetes de bits del mensaje y realizar la conversión a una señal de corriente que permita controlar cualquier elemento esclavo que trabaje con señales de 4-20mA.

La descripción detallada de los bloques descritos anteriormente y los dispositivos que componen el sistema se presenta en la figura 13. Cada recuadro punteado hace referencia a un bloque del diagrama de bloques general en el mismo orden que fue presentado.

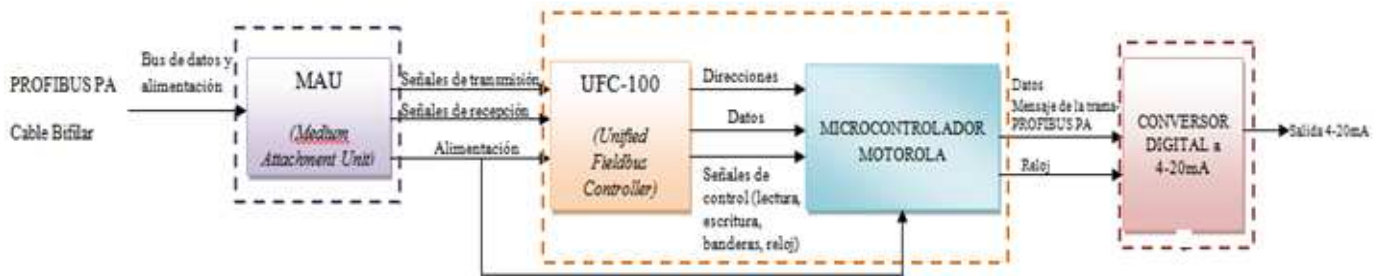


Figura 13. Diagrama de bloques detallado-Convertidor PROFIBUS PA a 4-20mA

5. COMPONENTES E INSTRUMENTACIÓN:

5.1 MAU (Medium Attachment Unit) - AMIS 49200:

Este dispositivo fue diseñado especialmente para aplicaciones en buses de campo como PROFIBUS PA, para cumplir con las normas establecidas según la IEC 61158-2, norma que rige la comunicación de buses de campo a nivel físico.

La MAU (Medium Attachment Unit), es un dispositivo transmisor- receptor de baja velocidad que permite manipular la fuente de alimentación y los datos provenientes del bus de forma independiente. Además de proporcionar voltajes de 5V y 3V por medio de la configuración interna de los reguladores shunt y serie, también maneja los datos del bus empleando el código Manchester establecido para las redes de comunicación PROFIBUS PA. Así, para efectuar la transmisión, la MAU envía una señal Manchester codificada dada por la interfaz MDS-MAU, y para la recepción el bloque receptor opera sobre una señal Manchester codificada, la cual decodifica y verifica la amplitud apropiada por medio de un circuito con detección de portadora y cruce por cero. De esta forma, la señal es luego transmitida a un controlador con interfaz MDS-MAU. [12]

La configuración interna de la MAU posee tres bloques principales que son: bloque de recepción, bloque de transmisión y bloque de alimentación, como se puede observar en la figura 14. [12]

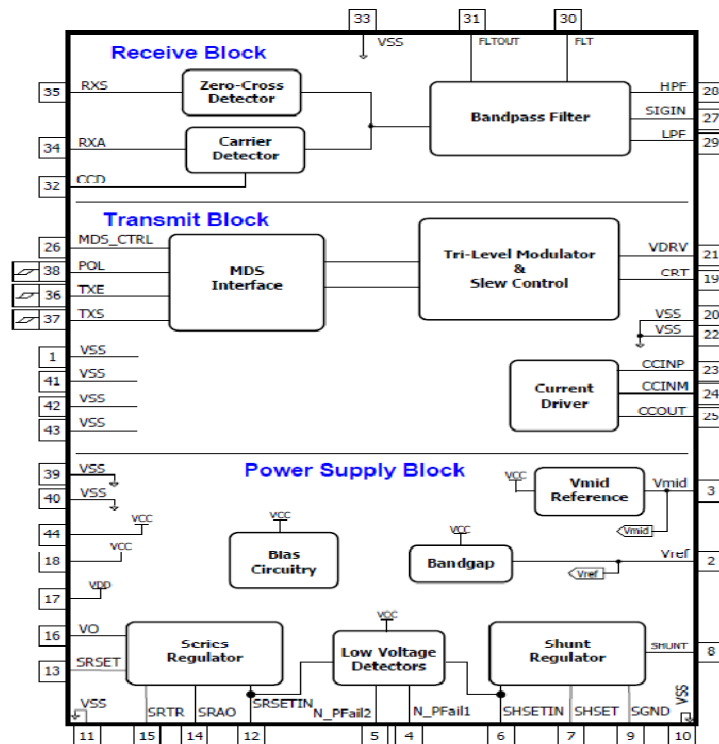


Figura 14. Configuración interna de la MAU [12]

BLOQUE DE RECEPCIÓN [12]: Este bloque se compone de un filtro pasabanda, un detector de potadora (encargado de generar la señal RxA) y un detector de cruce por cero (encargado de generar RxS por detección de transición de estados del código Manchester).

En la figura 15, se visualiza el filtro pasabanda usado por la MAU, el cual filtra la señal de comunicación entrante y su arquitectura se basa en una conexión en serie de un filtro pasa-altas y uno pasa-bajos, cada uno compuesto por un seguidor de voltaje y resistencias integradas, teniendo dos polos, por lo que sólo cuatro capacitores externos son necesarios.

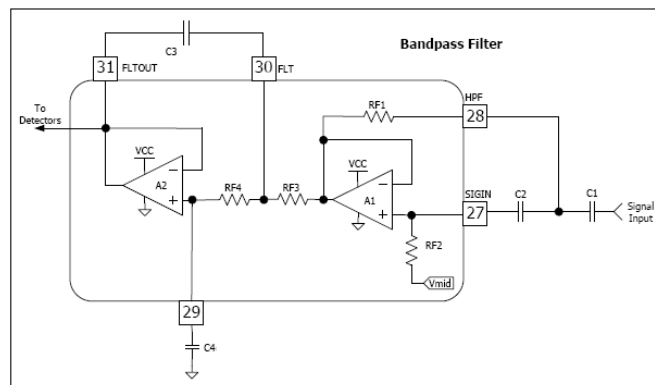


Figura 15. Filtro Pasabanda. [12]

BLOQUE DE TRANSMISIÓN: Está compuesto por una interfaz MDS que se encarga de decodificar las señales de entrada para generar las señales TxE y TxS, las cuales contienen la información que envía el dispositivo maestro. El circuito que compone esta interfaz se puede ver en la figura 16. [12]

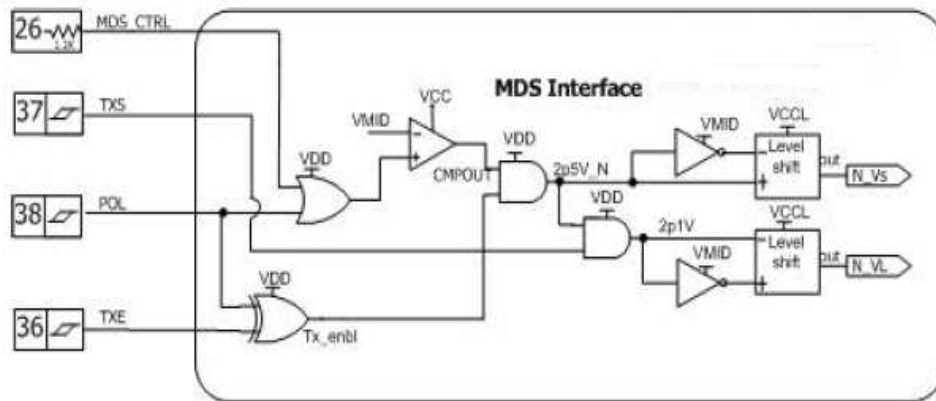


Figura 16. Interfaz MDS [12]

El pin POL es usado para seleccionar la polaridad de la señal habilitadora de la transmisión TxE (Transmit enable). La lógica de esta interfaz se da gracias a un modulador de tres niveles que genera señales de corriente usadas como entradas del controlador de slewrate, el cual a su vez genera 3 niveles distintos de voltaje (VS, VH, VL), los cuales establecen la lógica de operación de la interfaz MDS, como se hace explícito en la Figura 17.[12]

| POL | TxE | TxS | VDRV |
|------|------|------|------|
| Low | Low | Low | Vs |
| | | High | |
| | High | Low | VH |
| | | High | VL |
| High | Low | Low | VH |
| | | High | Vl |
| | High | Low | Vs |
| | | High | |

Figura 17. Lógica Interfaz MDS [12]

El bloque de transmisión de la MAU también cuenta con un amplificador operacional que permite manejar corrientes y voltajes acordes con la velocidad de transmisión del bus (31,25kbps). [12]

BLOQUE DE ALIMENTACIÓN: Este bloque provee los voltajes típicos (3V, 5V) necesarios para la alimentación de los circuitos de supervisión de voltaje, circuitos de polarización y otros circuitos dependientes. De esta forma el bloque de alimentación contiene cuatro subbloques: un regulador shunt, un regulador serie, dos detectores de voltaje bajo y voltaje de referencia. [12]

El regulador shunt establece el voltaje de alimentación Vcc usado para el circuito analógico, el cual es dividido por una red interna para proporcionar un voltaje igual a Vref en el pin SHSET de la MAU. [12]

El regulador serie establece el voltaje de alimentación Vdd usado para el circuito digital. Este bloque produce un voltaje regulado de 3V en el pin VO a partir de VCC. [12]

Los detectores de bajo voltaje son usados para monitorear los voltajes de alimentación y generar señales de falla. Al sensar el voltaje de los pines SHSETIN y SRSETIN, se detectan las alarmas de bajo voltaje. Típicamente SHSETIN y SRSETIN monitorean los voltajes VCC y VDD respectivamente. [12]

El voltaje de referencia se usa internamente para generar un nivel BIAS para voltajes AC. Este circuito genera dos señales de voltaje: VREF, usado como voltaje de referencia para todos los circuitos en la MAU y VMID, que regula a partir de un amplificador operacional para proporcionar un nivel común (BIAS) para las señales AC, como se observa en la figura 18. [12]

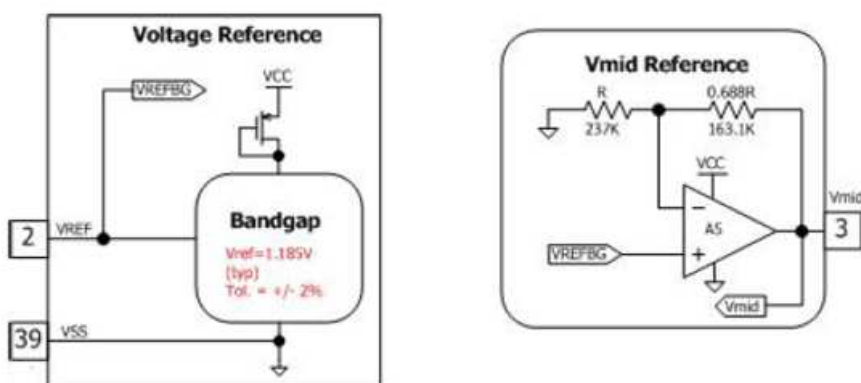


Figura 18. Voltajes de referencia [12]

Como se puede observar, la MAU es un elemento que incorpora las principales funcionalidades para el manejo de datos y distribución de fuentes de alimentación, lo cual se tomó como punto de partida para la elaboración y diseño del convertidor PROFIBUS PA a 4-20mA. Además de manejar una velocidad de datos de 31,25Kbps (parámetro establecido en las especificaciones de este proyecto), la MAU se integra fácilmente con el bus de campo PROFIBUS PA y otros componentes usados en la realización del proyecto, como la UFC (Unified Fieldbus Controller). AMIS 49200 es una de las referencias comerciales de MAU's disponibles y bastante accesibles económicamente. [12]

5.2 UFC (Unified Fieldbus Controller)

Es un dispositivo periférico que permite manejar los datos de acuerdo con los protocolos establecidos para buses de campo PROFIBUS PA. La UFC 100 incluye todas las funciones de tiempo crítico en el hardware e integra el manejo de la capa física y la capa de datos, funciones que permiten que la adquisición de datos se realice de forma eficiente y acorde con las normas establecidas para comunicaciones industriales. La UFC además de ser el complemento de la MAU, también se conecta a esta última con gran facilidad, como se indica en la figura 19. [13]

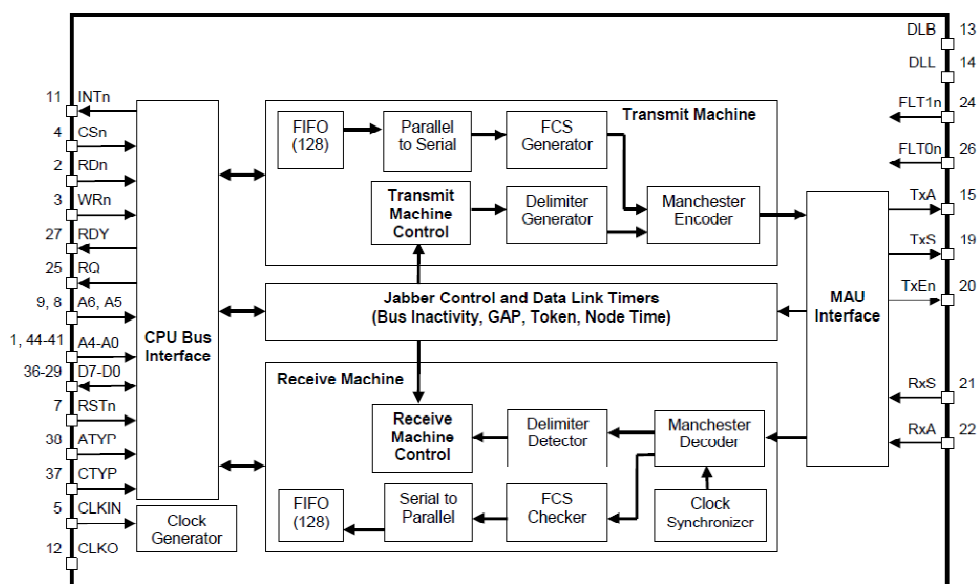


Figura 19. Diagrama de Bloques y conexiones de la UFC 100[13]

Se compone de una máquina de transmisión, una máquina de recepción y temporizadores de enlaces de datos, los cuales se encargan de manipular los datos de acuerdo con la velocidad del bus.

La máquina de transmisión genera una interrupción cuando todos los bytes de la trama hayan sido transmitidos desde el registro de la FIFO de transmisión (registro de la UFC de 128 bytes), dando continuación al ciclo para la recepción de una nueva trama. Cabe aclarar que la UFC toma en cuenta el tiempo establecido según el protocolo para el envío de una trama y la recepción de la siguiente. En la figura 20 se muestra el tiempo que debe haber entre 2 tramas según el protocolo de PROFIBUS PA. [13]

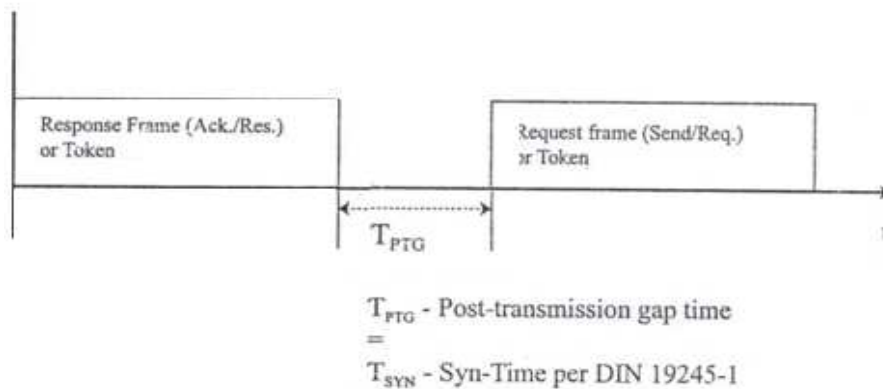


Figura 20. Condición de tiempo entre dos tramas [4]

La transmisión inicia con el envío de un número determinado de bytes correspondientes al preámbulo seguido por el delimitador de inicio y termina con el envío de los bytes restantes de la trama, seguido de dos bytes de FCS y el delimitador final. [13]

Es importante aclarar que los temporizadores de enlace de datos de la UFC tienen un contador de 32 bits con una resolución de 1/32ms. [13]

La UFC puede ser conectada a cualquier bus síncrono o asíncrono y es posible operarla con una CPU INTEL o Freescale. Para este proyecto se hizo uso de la tarjeta de desarrollo flexis de Freescale, ya que maneja un lenguaje de programación sencillo como lo es ANSI C, además de disponer de este recurso en la universidad. [13]

Por otro lado, también es importante tener en cuenta la configuración de otros bloques como son [13]:

CLOCK: El reloj de entrada debe configurarse en un rango de 1MHz- 40MHz, mientras que los 2 relojes de salida (pines 12 y 13 de la UFC) pueden ser configurados para diferentes frecuencias. Este bloque es fundamental ya que con él se generan las señales de reloj internas para el manejo de registros, tiempos de transmisión y recepción, etc.

INTERRUPT ENCODER: Este bloque contiene todas las interrupciones codificadas bajo un valor binario. Desde el software se puede acceder a este bloque para generar las tablas de saltos necesarias durante la transmisión y recepción de la trama.

Es fundamental conocer la conexión de este dispositivo con las otras interfaces, por ello a continuación se presentan los diagramas de conexión y referencias para la correcta operación de la UFC con otros dispositivos que hacen parte del proyecto.

INTERFAZ CON CPU FREESCALE:

En la figura 21, se muestra la conexión de la UFC 100 con la CPU freescale. Debido a la compatibilidad y facilidad de conexión propuesta por el manual de usuario de la UFC 100, esta configuración fue la escogida para la conexión entre UFC, MAU y CPU.

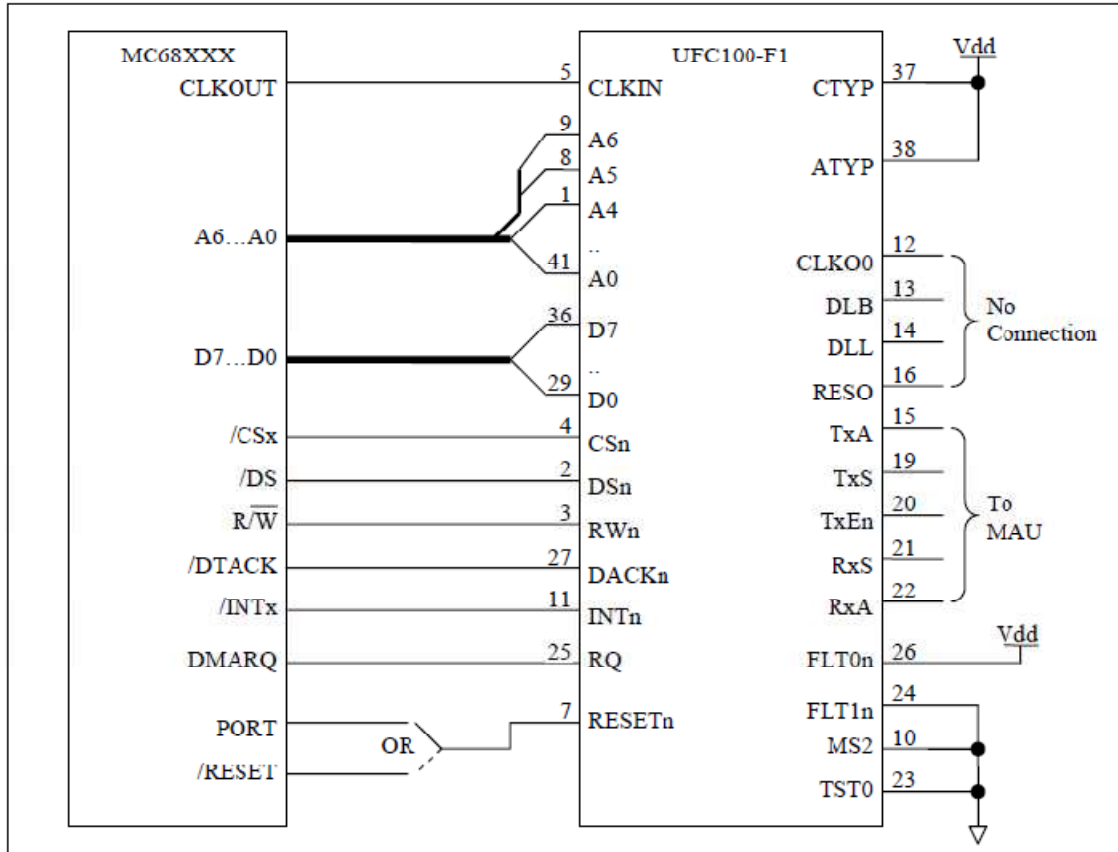


Figura 21. Interfaz con el microcontrolador Freescale [13]

INTERFAZ CON LA MAU[13]:

Señales llamadas TxE, TxS y TxA para la transmisión y RxA, RxS para recepción.

En la tabla 23, se muestra la lógica usada para la transmisión de datos.

| Transmit Mode | TxE | TxS | Operation |
|---------------|-----|-----|----------------------|
| Enable Low | 1 | 0 | Transmitter Inactive |
| | 1 | 1 | Transmitter Inactive |
| | 0 | 0 | Transmit '0' |
| | 0 | 1 | Transmit '1' |

Tabla 23. Lógica para la transmisión de datos empleada por la MAU [13]

La señal TxA es una salida de la UFC que indica el estado de la transmisión (Activa=1, Inactiva=0). Para el desarrollo del proyecto esta señal no se conecta, se deja como pin de prueba de transmisión.

TxEn es la señal de control, la cual marca el inicio y el final de la transmisión de la trama, y TxS es la señal de transmisión.

Por otro lado, la interfaz de recepción de la MAU emplea la lógica mostrada en la tabla 24.

| Receive Mode | RxA | RxS | Indication |
|--------------|------------|-----|------------------|
| Level high | 0 | 1 | Receive Inactive |
| | 0 | 0 | Receive Inactive |
| | 1 | 0 | Receive '0' |
| | 1 | 1 | Receive '1' |
| Level low | 1 | 0 | Receive Inactive |
| | 1 | 1 | Receive Inactive |
| | 0 | 0 | Receive '0' |
| | 0 | 1 | Receive '1' |
| Pulse high | 0 | 1 | Receive Inactive |
| | 0 | 0 | Receive Inactive |
| | High pulse | 0 | Receive '0' |
| | High pulse | 1 | Receive '1' |
| Pulse low | 1 | 0 | Receive Inactive |
| | 1 | 1 | Receive Inactive |
| | Low pulse | 0 | Receive '0' |
| | Low pulse | 1 | Receive '1' |

Tabla 24. Lógica para la recepción de datos empleada por la MAU [13]

La señal RxS es la señal de recepción y la señal RxA detecta actividad de recepción por detección de portadora. Es necesario contar esta última señal, pues el ruido también podría ser detectado como un dato como se muestra en la figura 22.

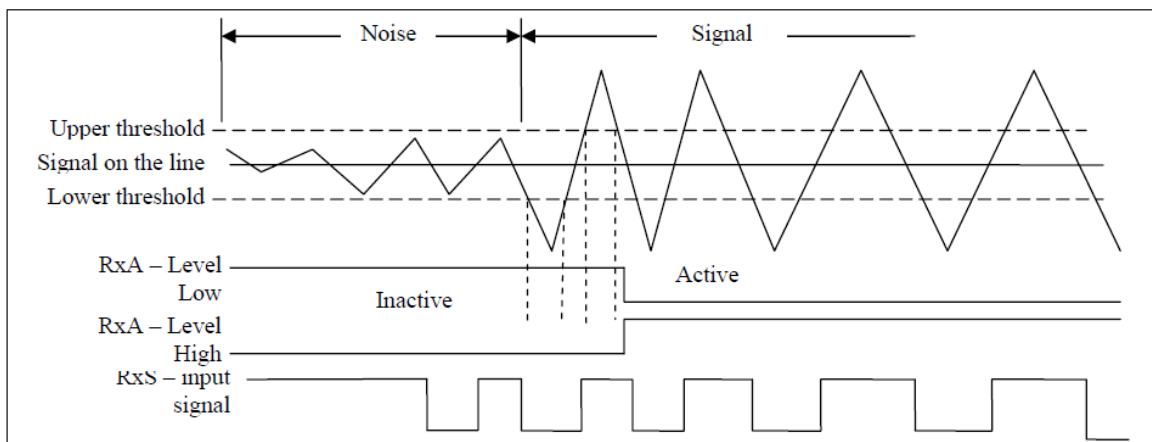


Figura 22. Modo Nivel - Detección de portadora (RxA) y señal de recepción (RXS)[13]

RxA es generada por la MAU sólo si se detecta actividad suficiente en las terminales de recepción por encima del nivel de ruido y por un intervalo de tiempo aceptable.

5.3 FLEXIS FREESCALE:

La tarjeta de desarrollo que se muestra en la figura 23, fue diseñada por Freescale con el fin de tener una interfaz entre el programador (PC) y el microcontrolador. De esta manera el código fuente utilizado en el desarrollo del proyecto se pudo quemar en el microcontrolador a través de un puerto Mini-USB- USB, el cual se conecta con el computador. El código fue compilado con el software CODEWARRIOR y es posible detallarlo en el anexo 10.1.

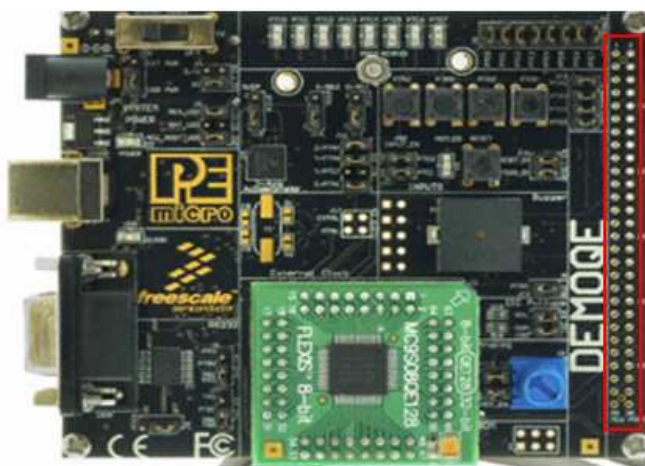


Figura 23. Tarjeta de desarrollo FLEXIS 8-bit de Motorola

Los pines encerrados en el cuadro rojo corresponden a las entradas y salidas asignadas en la programación y asignación de pines de la tarjeta. Las entradas provienen del circuito de la UFC que se vio en la sección 5.2, del cual se reciben los datos ya tratados según el protocolo PROFIBUS PA. El programa se encarga de manejar los datos recibidos, tiempos de espera, interrupciones y configurar tramas de repuesta al maestro PROFIBUS, es decir, es el encargado de establecer la comunicación maestro-esclavo.

5.4 CONVERTOR DIGITAL ANALÓGICO

Para diseñar el conversor digital analógico se parte de un dispositivo que se encargue de convertir la información digital cargada a la entrada en un rango de voltaje, para luego, por medio de una red de amplificadores (los cuales cumplen las funciones de circuito de referencia y amplificador de corriente) ajustar la escala de dicho voltaje de entrada a una corriente de salida de 4-20mA.

El convertidor digital a voltaje usado en este proyecto corresponde a la referencia AD558. Este dispositivo maneja una resolución de 8 bits y genera 2 posibles rangos de salida de 0 V a 2.56 V o de 0 V a 10 V; dependiendo del tipo de interconexión y la fuente de alimentación usada (De 0V-2.56V la fuente de alimentación debe estar en el rango de +4.5 V a +16.5 V.; para el intervalo de 0 V a 10 V el rango de voltaje de alimentación requerido va desde +11.4 V a +16.5 V.) como se muestra en la Figura 24.[15]

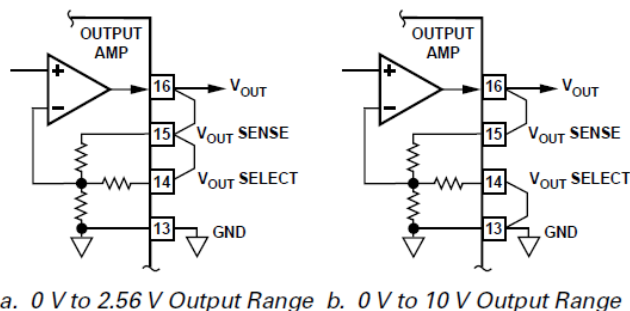


Figura 24. Diagrama de Conexión[15]

El rango de salida seleccionado es el de 0V a 2.56V, pues maneja una fuente de alimentación que se encuentra dentro del rango de voltajes que maneja la MAU. En la tabla 25 se observa la relación de conversión entre los datos y el voltaje.

| BINARIO | SALIDA EN VOLTAJE |
|----------|-------------------|
| 00000000 | 0V |
| 00000001 | 0.010V |
| 00000010 | 0.020V |
| 00001111 | 0.150V |
| 00010000 | 0.160V |
| 01111111 | 1.270V |
| 10000000 | 1.280V |
| 11000000 | 1.920V |
| 11111111 | 2.55V |

Tabla 25. Relación de Datos de Entrada Con Voltaje De Salida

5.5 DP/PA LINK

El DP/PA Link está conformado por el módulo IM157 (esclavo DP) y un máximo de 5 acopladores DP/PA, como se puede ver en la figura 25. [15]

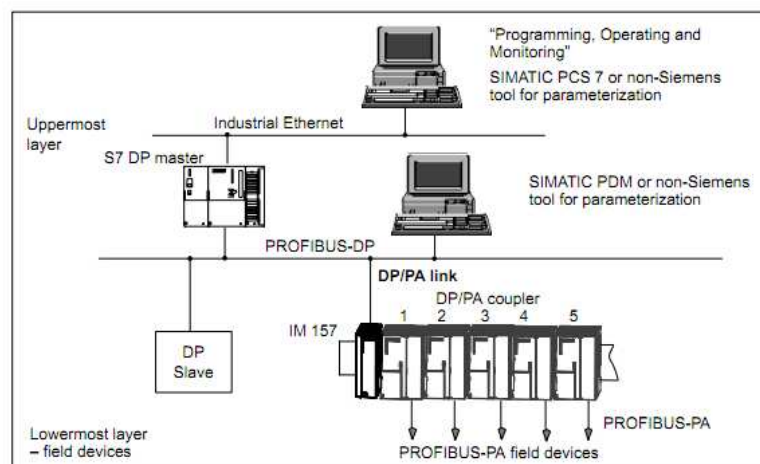


Figura 25. Integración del DP/PA link al sistema [15]

El DP/PA link es un esclavo DP visto desde PROFIBUS DP y un maestro PA visto desde PROFIBUS PA. Este dispositivo provee la conexión de PROFIBUS DP a PROFIBUS PA. El principio de operación es el que se presenta a continuación [15]:

- EL DP/PA link mapea la red subyacente PROFIBUS PA a un esclavo DP
- PROFIBUS DP es desacoplado de PROFIBUS PA a través del DP /PA link a medida que la transmisión transcurre.
- El maestro y el esclavo PA forman un sistema independiente (bus en una red separada).
- Los acopladores DP/PA se usan para hacer más eficiente la energía utilizada. El acoplador DP/PA y los dispositivos de campo PA forman un bus común PROFIBUS PA.

El proceso descrito anteriormente se puede observar en la figura 26.

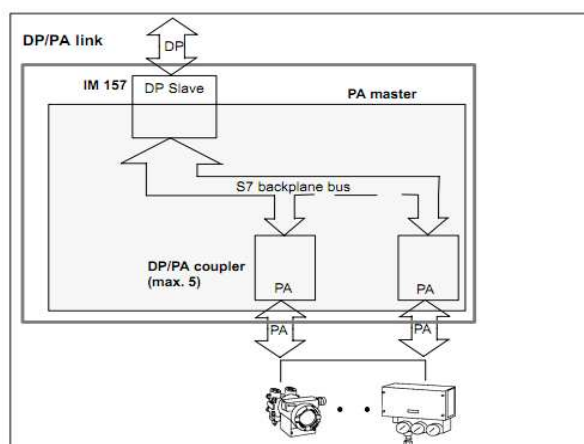


Figura 26. Esquema de conexión general de la red DP/PA [15]

5.5.1 ACOPLADOR DP/PA

El acoplador DP/PA es utilizado para la conexión del esclavo DP y la red PA de entrada al convertidor diseñado en este proyecto. Algunas de las características de este instrumento son las siguientes [15]:

- Aislamiento entre PROFIBUS DP y PROFIBUS PA.
- Conversión entre los medios de transmisión RS-485 e IEC 1158-2
- Diagnóstico por medio de LEDs
- Velocidad de transmisión de PROFIBUS DP de 45,45Kbps.
- Velocidad de transmisión de PROFIBUS PA de 31,25Kbps
- Alimentación integrada para PROFIBU PA.

La figura 27 muestra las conexiones y partes del acoplador DP/PA [15].

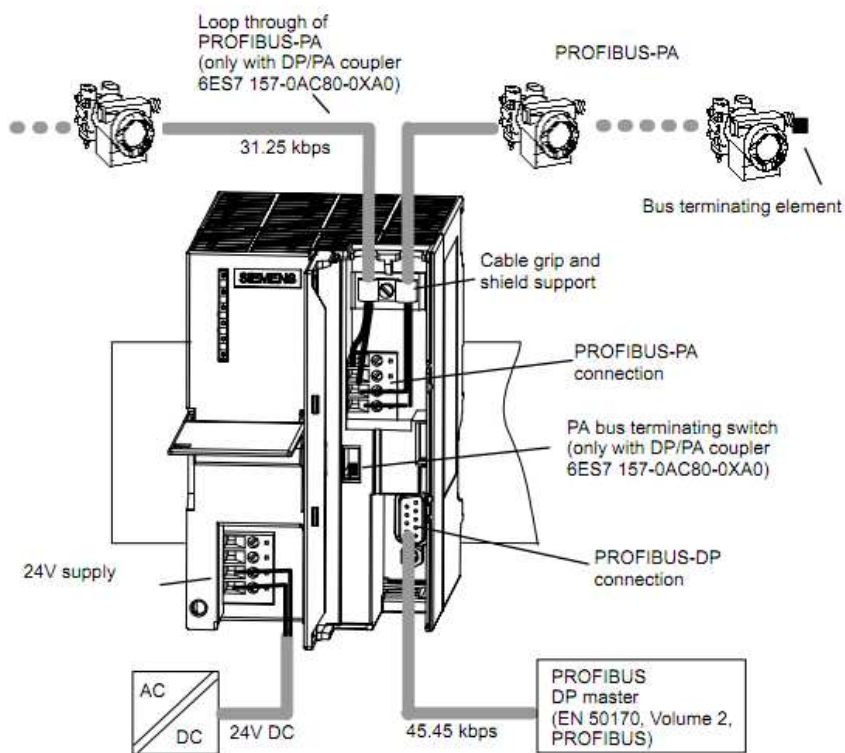


Figura 27. Acoplador DP/PA [15]

5.6 PLC SIEMENS

EL PLC está conformado por una fuente de alimentación, la CPU y módulos de entrada-salida. Usando el software STEP 7 de SIEMENS se programa y monitorea el sistema que compone el convertidor PROFIBUS PA a 4-20mA. La figura 28 muestra un ejemplo de la interconexión de los elementos que hacen parte del PLC.

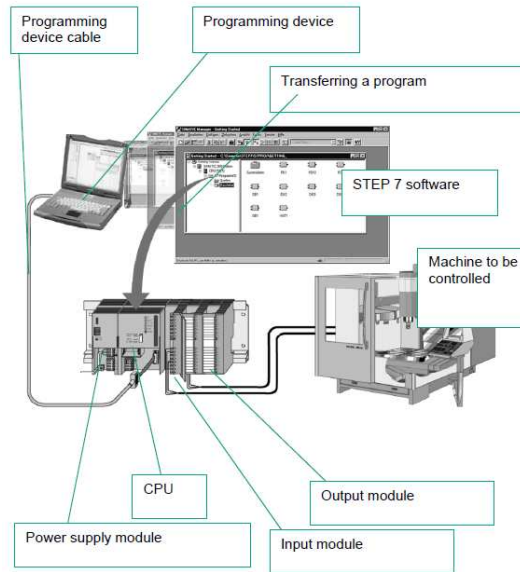


Figura 28. Esquema de comunicación con el programa STEP 7 de SIEMENS[16]

5.7 ANALIZADOR DE PROTOCOLO SOFTING

Existen algunas herramientas de diagnóstico para la comunicación y automatización industrial. El Analizador de Protocolo Softing es un instrumento ideal para realizar análisis de procesos de comunicación en redes PROFIBUS, ya que ayuda a prevenir fallos en la planta debido a problemas de comunicación, examina y corrige los errores latentes y proporciona una descripción clara de todas las estaciones del sistema. [18]

El analizador de Protocolo exhibe el estado operacional por medio de un indicador según el siguiente criterio [19]:

VERDE: Indica operación correcta

AMARILLO: Indica situaciones excepcionales tales como diagnósticos o reparametrizaciones.

ROJO: La estación ha fallado.

Además de ofrecer una información legible y fácil de interpretar, permite optimizar la red de acuerdo con los parámetros actuales. Para la conexión directa a las redes PROFIBUS PA está disponible una segunda interfaz para la parte física del bus MBP (Manchester Bus Powered), es decir, el tráfico del bus se puede registrar y comparar en ambos lados de un acoplador del segmento PA, es decir, se puede analizar tanto la parte de PA como de DP.[19]

El análisis y corrección de las tramas PROFIBUS PA se realizó a partir del estudio de los datos entregados por el analizador de protocolo mencionado anteriormente, asegurando que la entrada al convertidor fuera la deseada. En la figura 29 se muestra el proceso de conexión del sistema para la verificación de tramas usando el analizador de protocolo.



Figura 29. Proceso de verificación de trama PROFIBUS PA

6. PRUEBAS Y ANALISIS DE RESULTADOS

6.1 DESARROLLO HARDWARE:

6.1.1 CIRCUITO DE IMPLEMENTACION DE LA MAU

En la figura 30 se muestra el circuito de implementación de la MAU según el esquema dado por el fabricante.

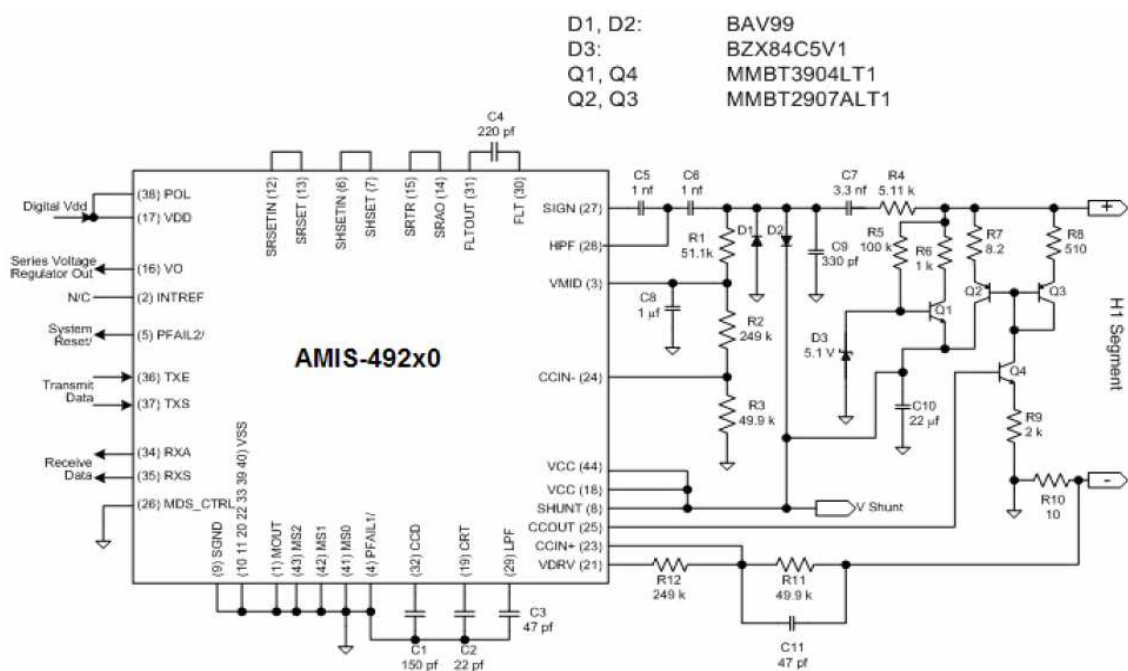


Figura 30. Circuito referencial de implementación AMIS 492x0[12]

A este circuito se le realizaron algunas mediciones de verificación, en las cuales se revisó que los voltajes de polarización de los componentes que constituyen la MAU permitieran que los bloques que la componen funcionen de manera adecuada, al igual que los circuitos externos que dependen de ella. Al realizar esta medición se pudo evidenciar que el diodo zener (D3) de 5.1V propuesto por el fabricante no manejaba un voltaje suficiente para polarizar el circuito de la MAU, ya que con una caída de 0,6V en la juntura base-emisor del transistor Q1, el voltaje shunt es aproximadamente 4.5V, lo cual no permite la operación de los reguladores de voltaje internos de la MAU, por lo que fue necesario hacer ciertas modificaciones en el diodo D3 con respecto al valor recomendado por el fabricante en la hoja de especificaciones. En la tabla 26 se pueden ver los voltajes de polarización obtenidos con el diodo zener de 5.1V y 5.5V.

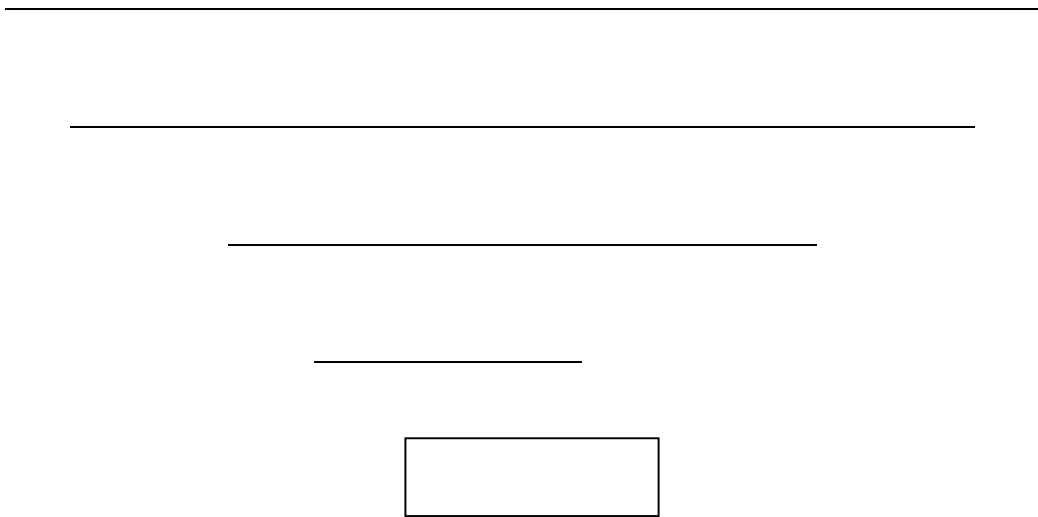
| PARAMETRO | DIODO DE 5.5V | DIODO DE 5.1V |
|-----------|---------------|---------------|
| VSHUNT | 5.08V | 4.6V |
| VD3 | 5.84V | 5,31V |
| VO | 3.03V | 1,57V |
| VREF | 1.19V | 1,175V |
| SRSETIN | 1.19V | 1,175V |

Tabla 26. Medición de voltajes en la MAU. Vzener 5.1V vs. Vzener 5.5V

Al comparar los resultados obtenidos con las especificaciones dadas por el fabricante, las cuales indican $V_O=3V$ y $V_{SHUNT}=5V$ y al observar los voltajes de alimentación de los circuitos externos conectados a la MAU, se concluye que el diodo de 5.5V permite trabajar con el voltaje suficiente para alimentar los circuitos.

Del mismo modo se realizaron las mediciones correspondientes para evaluar los rangos de corrientes arrojados por el dispositivo. Realizando el cálculo de la corriente que atraviesa el transistor Q4 (ecuación[a]), se tiene que la corriente del bus debe ser 10mA para que se efectuó la comunicación correctamente, como se muestra en el siguiente desarrollo:

$$I_{bus} = \frac{[R_3 V_{mid} (R_{12} + R_{11})] - [V_{drv} (R_2 R_{11} + R_3 R_{11})]}{- [R_F (R_2 R_{12} + R_3 R_{12})]} \quad [a]$$



6.1.2 CIRCUITO DE CONVERSIÓN DE VOLTAJE A CORRIENTE

En la figura 31 se muestra el circuito implementado para generar la señal en corriente de 4-20mA a partir del voltaje entregado por el Convertidor Digital Análogo AD558JN de 8 bits.

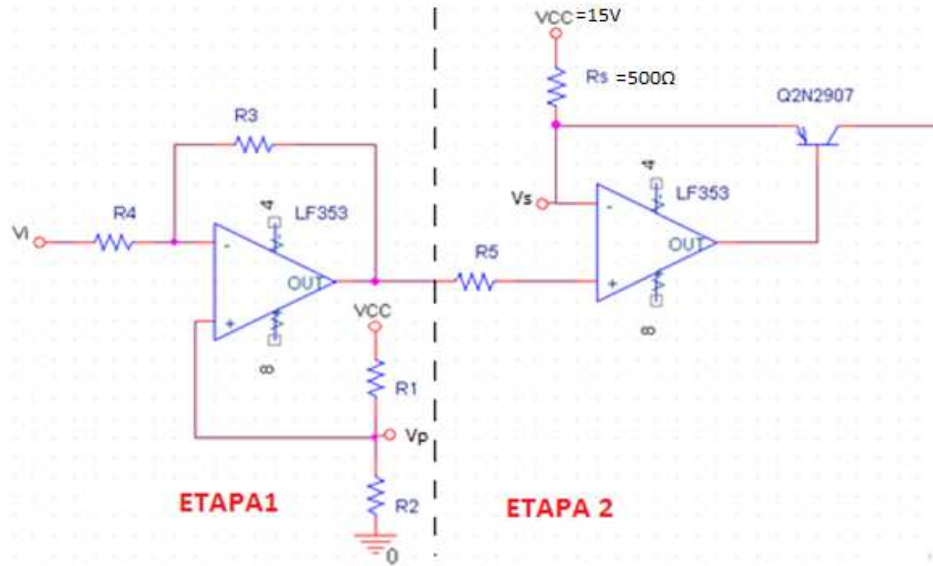


Figura 31. Circuito implementado para generar la corriente de 4-20mA

Este circuito consta de 2 etapas: La etapa 1 es un amplificador de voltaje empleado para adaptar el margen de 0-2.56 voltios entregado por el DAC. La etapa 2 es un generador de corriente de 4-20mA controlado por el voltaje entregado en la etapa 1.

Los cálculos realizados para hallar los valores de relación de las resistencias se muestran a continuación:

$$V_p - \frac{(V_i - V_p)}{R_4} (R_3) = V_s \quad \Rightarrow \quad R = \frac{R_3}{R_4}$$

Asumiendo \$R_s = 500\Omega\$ con \$V_{cc} = 15V\$, para obtener una corriente de 4mA a través del transistor Q2N2907 se tiene:

$$\frac{(V_{cc} - V_s)}{R_s} = 4mA \quad \Rightarrow \quad V_s = 13V$$

Para obtener una corriente de 20mA a través del transistor Q2N2907 se tiene:

$$\frac{(V_{cc} - V_s)}{R_s} = 20mA \quad \Rightarrow \quad V_s = 13V$$

Por lo tanto,

$$V_p - \frac{(V_i - V_p)}{R_4} (R_3) = 13V \quad \text{con} \quad V_i = 0V$$

$$V_p - \frac{(V_i - V_p)}{R_4} (R_3) = 5V \quad \text{con} \quad V_i = 2.56V$$

Se obtiene:

$$V_p = 3.1V$$

$$R = 3.125$$

Asumiendo $R_3 = 150K\Omega$

$$R = \frac{R_3}{R_4} V_s \quad \Rightarrow R_4 \cong 47K\Omega$$

Ahora bien,

$$V_{CC} \left(\frac{R_2}{R_1 + R_2} \right) = V_p$$

Asumiendo $R_2 = 12K\Omega \quad \Rightarrow R_1 \cong 3,3K\Omega$

6.2 DESARROLLO SOFTWARE

6.2.1 PROGRAMACION DEL PLC

Para realizar la programación del PLC fue empleado el software de configuración STEP 7 de SIEMENS.

Inicialmente, fue configurada la arquitectura de la red PROFIBUS implementada donde se incluyó la CPU 414-3 del PLC con dirección 1 en la red PROFIBUS DP, el DP/PA Link IM 153-2 con dirección 3 en la red PROFIBUS DP y dirección 1 en la red PROFIBUS PA, el DP/PA Coupler FDC 152 y el convertidor PROFIBUS PA a 4-20mA TG_0981 con direcciones 4 y 9 en la red PROFIBUS PA, respectivamente, como se observa en la figura 32.

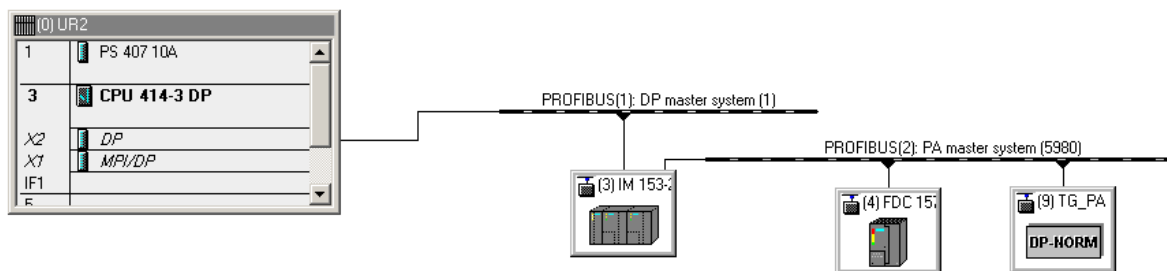


Figura 32. Red PROFIBUS PA implementada en STEP 7

En la tabla de símbolos, fueron creadas las variables “valor corriente” y “dato análogo”. “valor corriente” fue direccionada en la posición de memoria MD10 (registro interno de trabajo del PLC) y “dato análogo” fue direccionada como salida análoga por comunicaciones desde el PLC con formato PQD527, como se observa en la Figura 33.

| Status | Symbol | Address | Data type | Comment |
|--------|-----------------|---------|-----------|---------------------------|
| 1 | dato analogo | PQD 527 | REAL | salida al esclavo |
| 2 | valor corriente | MD 10 | REAL | valor constante corriente |
| 3 | I/O_FLT2 | OB 83 | OB 83 | I/O Point Fault 2 |
| 4 | | | | |

Figura 33. Asignación de Parámetros a la Posición de memoria del PLC.

Para enviar los datos de corriente desde el PLC al convertidor PROFIBUS PA a 4-20mA, fue desarrollada una rutina en lenguaje de programación ladder donde, a partir de un bloque “MOVE”, se asigna el valor de la variable “valor corriente” a la variable “dato análogo” a ser transmitido. Para la realización de pruebas, el dato en corriente dado en la variable “valor corriente” es modificado de forma manual dentro del rango de 4-20mA, como se observa en la figura 34.

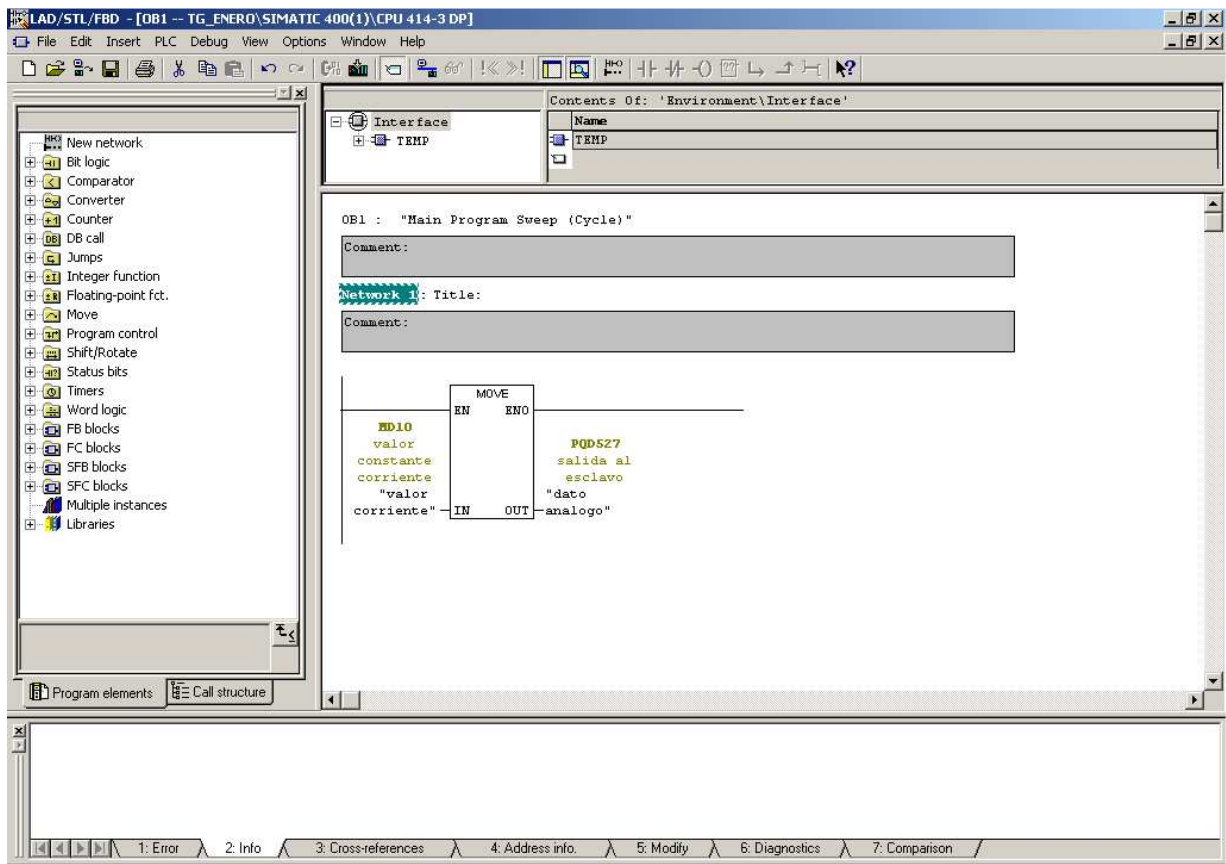


Figura 34. Bloque STEP 7 para la asignación de dato en corriente.

| Address HEX | Name | Access Read/ Write | Reset value | Description |
|----------------|---------------------------------|--------------------------|-------------------|--|
| 00 | Reset, Version | R/W | 0x10 | Software reset of YTZ420, UFC100-F1 Version |
| 01 | Mode | R/W | 0x00 | Selection of operating modes |
| 02 | Control | R/W | 0x00 | Control functions |
| 03 | Status | R/- | 0x82 | Shows status |
| 04 | Interrupt status | R/- | 0x00 | Shows reason of interrupts |
| 05 | Error status | R/- | 0x00 | Shows reason of communication errors |
| 06 | Interrupt mask | R/W | 0xFE | Mask for interrupts |
| 07 | Error mask | R/W | 0xFB | Mask for error interrupts |
| 08 | Tx length (LOW) | R/W | 0x00 | Length of transmitted frame (Lower byte) |
| 09 | Tx length (HIGH) | R/W | 0x00 | Length of transmitted frame (Higher 2 bits) |
| 0A | Bit time counter ⁽¹⁾ | -/W | .. ⁽³⁾ | Not used |
| 0B | FIFO control | R/W | 0x00 | Control register of FIFO |
| 0C | FIFO status | R/- | 0x11 | Shows status of FIFO |
| 0D | DATA | R/W | 0x00 | Transmit/Receive data to/from FIFO |
| 0E | Reserved | -/- | .. ⁽³⁾ | Not used |
| 0F | Switch ⁽¹⁾ | R/W | .. ⁽³⁾ | Not used |
| 10 | Clock mode ⁽²⁾ | R/W | 0x00 | DL mode, Timer enables |
| 11 | Timer status | R | 0x00 | Node-timer status |
| 12 | Node time (LOW) | R/W | 0x00 | Timer to hold DL NODE time (Lower byte) |
| 13 | Node time (HIGH) | R/W | 0x00 | Timer to hold DL NODE time (Higher byte) |
| 14 | Gap time | -/W | 0xFF | Value to generate minimum inter-PDU delay |
| 15 | Reserved | -/- | .. ⁽³⁾ | Not used |
| 16 | Watch time (LOW) | -/W | 0xFF | Value to detect no-activity of bus (Lower byte) |
| 17 | Watch time (HIGH) | -/W | 0xFF | Value to detect no-activity of bus (Higher byte) |
| 18 | Token counter (LOW) | R/W | 0x00 | Remaining token holding time (Lower byte) |
| 19 | Token counter (HIGH) | R/W | 0x00 | Remaining token holding time (Higher byte) |
| 1A | Timer control | R/W | 0x00 | Control DL timers |
| 1B | Node address ⁽¹⁾ | R/W | 0x00 | Node address |
| 1C | Reserved | - | .. ⁽³⁾ | Not used |
| 1D | Reserved | - | .. ⁽³⁾ | Not used |
| 1E | FC window ⁽¹⁾ | R/W | .. ⁽³⁾ | Not used |
| 1F | FC data ⁽¹⁾ | R/W | .. ⁽³⁾ | Not used |

(1): These are FRONTIER-1+ registers.
(2): FRONTIER-1+ has additional bits in this register, which are used only in FRONTIER-1+ compatible mode.
(3): Unused registers read as 0x00.

Figura 36. Registros de lectura/escritura de la UFC

El código fuente programado en el Microcontrolador se realizó siguiendo la siguiente secuencia:

Por medio de la rutina de inicialización, se configuran condiciones iniciales de la UFC a partir de la escritura de datos a ciertos registros, siguiendo las recomendaciones dadas por el fabricante de la UFC100, donde se habilita el estado de recepción en espera de tramas enviadas desde el maestro[13].

Cada vez que se presenta una interrupción, la UFC activa la señal de hardware INTn configurada en un puerto de la tarjeta de desarrollo Freescale donde, a partir de la lectura del registro de estado de interrupción (04 en la figura 36), se conoce el motivo de dicha interrupción que puede ser de transmisión, de recepción o que hubo una condición de error

Si se presenta una interrupción por recepción, se lee desde la UFC el primer byte de la trama enviada desde el maestro que corresponde al delimitador de inicio SD. Este byte es comparado (si es de tipo SD1, SD2, etc) para conocer el formato de la trama que fue enviada y así almacenar los datos en registros previamente configurados. Si el delimitador de inicio es de tipo SD2, se revisan los bits más significativos de los bytes dirección de fuente SA y dirección de destino DA que de ser '1', indican que hay puntos de servicio de acceso SAP's a partir de los cuales se ejecuta la máquina de estados del protocolo PROFIBUS PA, explicada en la sección 2.8. En caso que no existan SAPs en la trama enviada desde el maestro, indica que la unidad de datos DU corresponde a la corriente a ser convertida a 4-20mA, es decir el estado de intercambio de datos cíclico.

Después, se almacena la dirección de destino DA (dirección del esclavo) y se revisa si corresponde con la dirección configurada para el Conversor PROFIBUS PA a 4-20mA (0x09). De ser así, se continúa con la comparación de los datos almacenados y si no, se habilita la UFC para recibir una nueva trama

Posteriormente, se compara el byte código de función FC para conocer el estado de la comunicación, por ejemplo si la trama enviada desde el maestro es un primer diagnóstico, si es de solicitud de estado, etc.

A continuación, si se encuentra activa la bandera que indica la existencia de puntos de servicios de acceso SAPs, estos son comparados para saber si la trama enviada es de diagnóstico, parametrización, configuración o intercambio de datos cíclico.

Si la trama es de diagnóstico y se encuentra activa la bandera que indica primer diagnóstico, se prepara la trama de respuesta previamente establecida con el primer diagnóstico del Convertidor PROFIBUS PA a 4-20mA.

Si la trama es de parametrización o configuración, se prepara una trama en respuesta con un reconocimiento corto (0xE5) y se comparan los datos provenientes desde el maestro para generar la trama de respuesta cuando se solicite el segundo diagnóstico.

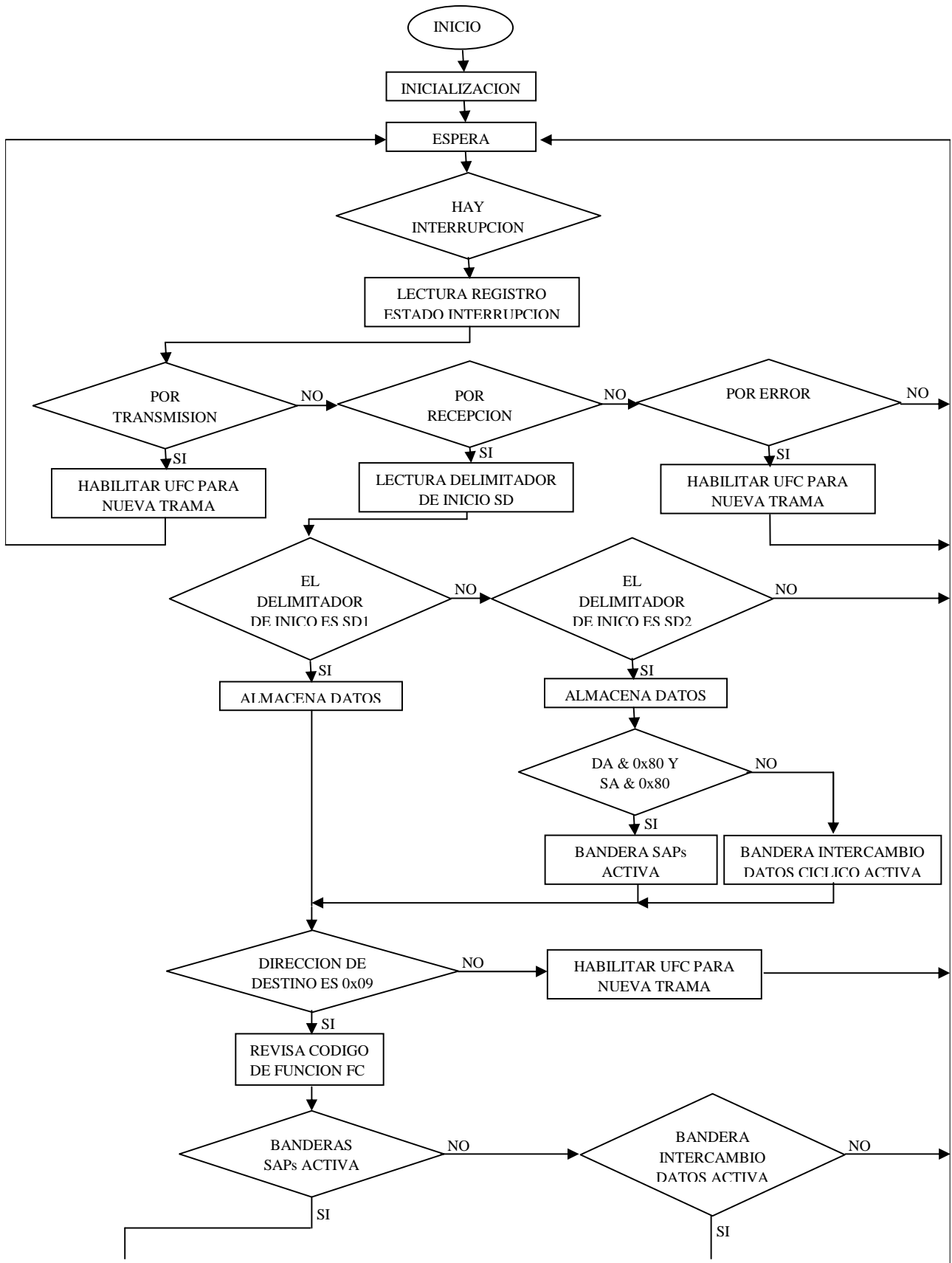
Cuando se presenta la trama de intercambio de datos cíclico, el dato en corriente es convertido y escalizado para luego ser enviado al conversor digital analógico. La respuesta a este tipo de trama se realiza con un reconocimiento corto, al igual que las tramas de parametrización y configuración.

Finalmente, para cada ciclo, a partir de una rutina de transmisión que activa las condiciones de transmisión de la UFC, se envían las tramas previamente preparadas en respuesta a las tramas recibidas desde el maestro.

Ahora bien, si se presenta una interrupción por transmisión, se habilita la UFC para recibir una nueva trama

Si se presenta una interrupción por condición de error, se limpian los registros de almacenamiento de datos de la UFC provenientes desde el maestro y se habilita para recibir una nueva trama

En la figura 37 se muestra el diagrama de flujo de la programación configurada en el Microcontrolador Freescale.



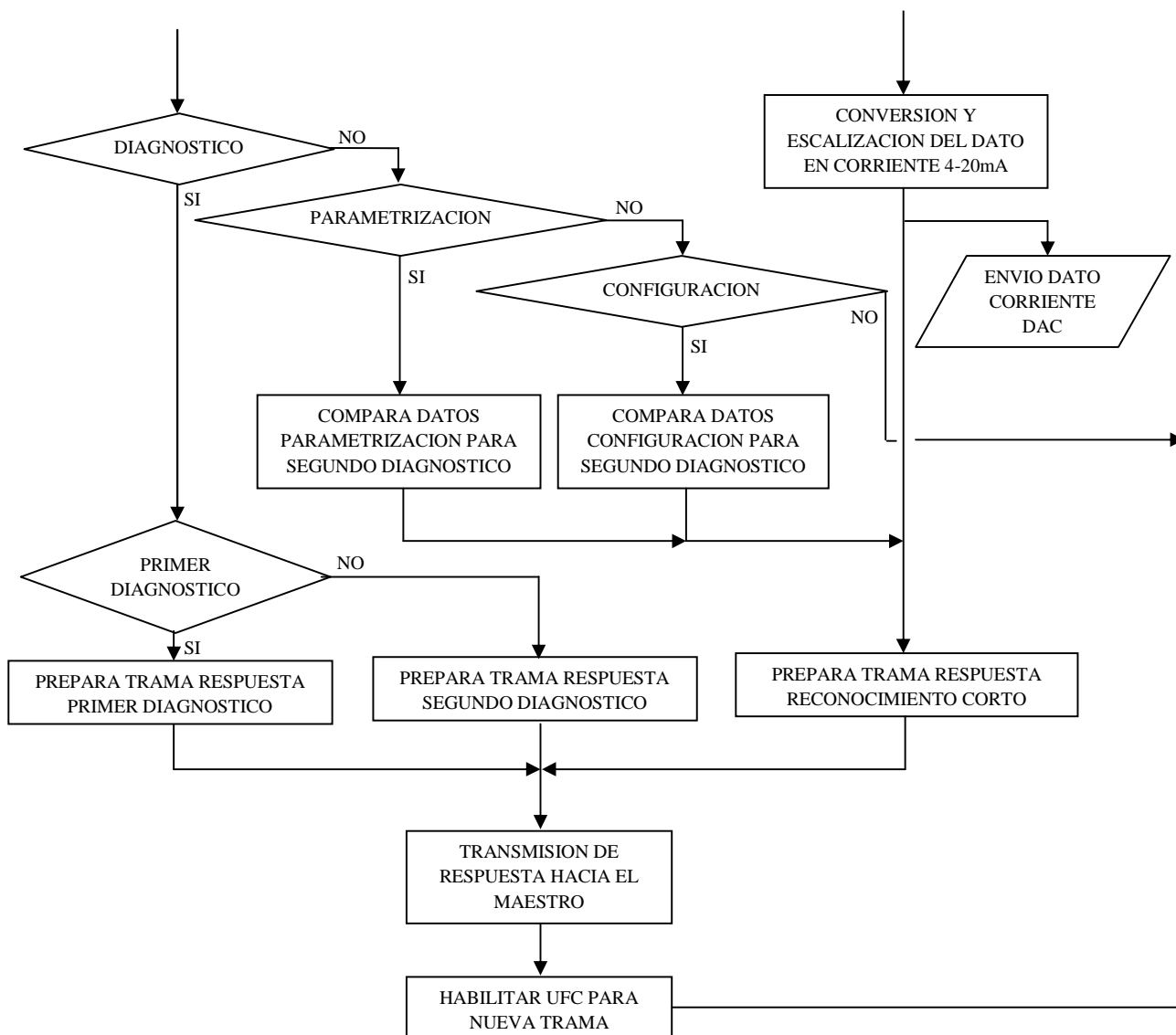


Figura 37. Diagrama de flujo del programa configurado en el microcontrolador Freescale.

En el ANEXO 10.1 se muestra el código desarrollado.

6.2.3 ARCHIVO GSD

Las herramientas disponibles para dispositivos PROFIBUS utilizan un formato especial de archivo ASCII, referido en este caso a un archivo de descripción general de la estación (GSD), el cual proporciona: información necesaria para identificar un dispositivo esclavo que está conectado al maestro, descripción de los datos del dispositivo esclavo que se pueden acceder a través de la red PROFIBUS (parámetros de configuración) y descripción de las capacidades de comunicación con el esclavo (velocidad de transmisión).[17]

Todos los dispositivos con una interfaz de comunicación IEC 61784-1, CP3 /1y CP3 / 2 compatibles con la interfaz empleada en la realización de este proyecto, deben disponer de un archivo GSD [17].

El GSD debe contener la información del distribuidor y los dispositivos, estados de hardware y software de liberación, velocidades de transferencia, intervalos de tiempo de vigilancia y la asignación de la señal en el conector del bus. [17]

6.2.3.1 CONVENCIONES

El tipo de ID especificado para las palabras clave (ver ANEXO 10.2) se refiere a los parámetros con el mismo nombre. En el caso de los parámetros, se puede establecer diferencias entre [17]:

- Obligatorio (M): absolutamente necesario
- Opcional (O): es posible añadir más.
- Por defecto (D): Opcional con defecto (= 0 si no existe)
- Agrupados (G): Al menos una palabra clave del grupo es necesaria

6.2.4. CONEXIÓN DE LA RED PROFIBUS

Para la conexión de la red PROFIBUS se retoma la documentación mostrada en capítulos anteriores y se realizan los procesos de configuración del PLC y la red de comunicación mostrada en el capítulo 5.6, luego se procede a realizar la conexión de la red PROFIBUS que se muestra en la figura 38.

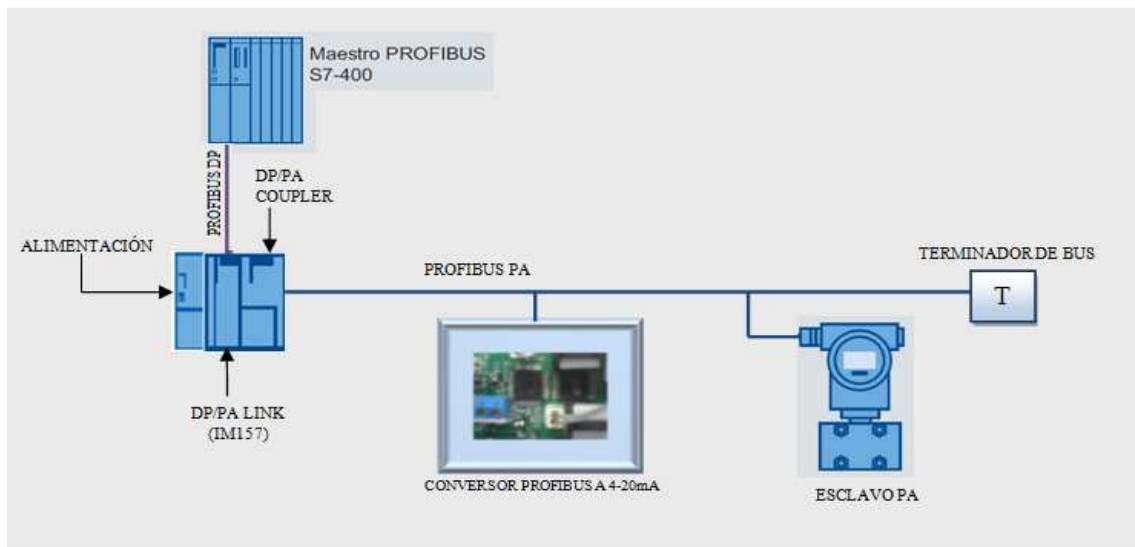


Figura 38. Arquitectura de la red PROFIBUS Implementada

El terminador de bus que se observa en la arquitectura de la figura 38, es una resistencia de cierre de bus automática en los acopladores DP/PA, la cual permite tener un aislamiento automático de segmentos parciales en caso de cortocircuito o ruptura de cable, y modificación de la configuración y de los instrumentos durante el funcionamiento (añadir o retirar segmentos). [1]

Es importante corroborar que el hardware conectado a la red se encuentre con la configuración cargada; para ello se realizó el siguiente procedimiento:

1. Al encender el PLC y la alimentación todas las alarmas del LINK y el COUPLER se activan. Esto ocurre cuando hay fallas en el hardware conectado a la red o no se ha cargado la configuración del sistema.
2. Al cargar el hardware y el software programado en el programa STEP 7 de SIEMENS (capítulo 5.6), estas alarmas desaparecen y la red está lista para ser implementada. Para comprobar esto se hace una prueba sencilla con el transmisor PA conectado a la red. Con el PLC en posición RUN se accede al bloque programado en el software y se verifica que el valor mostrado en el simulador corresponda al valor visualizado en el display del transmisor PA conectado a la red. La figura 39, muestra el valor en milibares mostrado en el transmisor y el valor mostrado en el simulador.

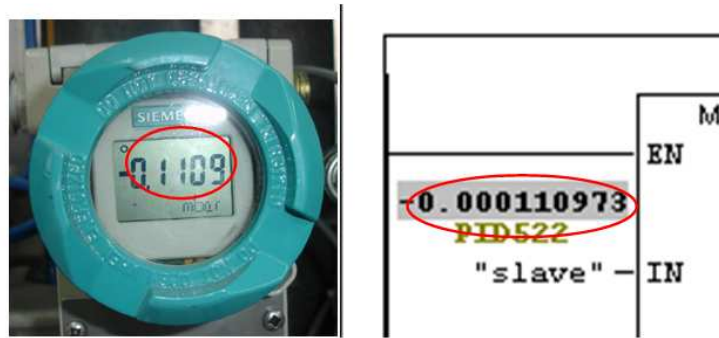


Figura 39. Prueba de conexión de la red online con el transmisor PA.

6.2.5 VERIFICACION DEL CODIGO FUENTE

6.2.5.1. PRUEBA TRANSMISIÓN:

Para verificar que la transmisión se está realizando, se efectuaron las mediciones usando un osciloscopio para observar las tramas enviadas desde el Convertidor PROFIBUS PA a 4-20mA hacia el elemento maestro. A continuación se muestran cada uno de los bytes que conformaban las tramas de transmisión, dependiendo del estado:

En la figura 40 se observa cómo a partir del pulso negativo de la señal RxA enviada desde la MAU hacia la UFC (visualizada en azul-CH2), inicia la transmisión de datos hacia el elemento maestro.

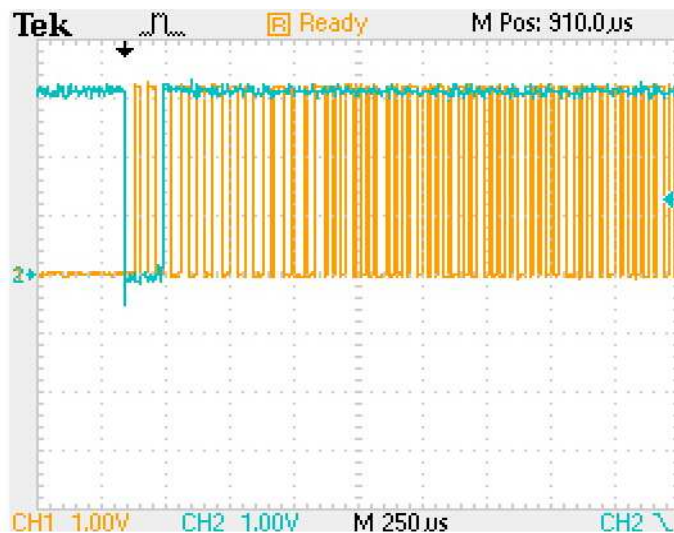


Figura 40. Transmisión de datos PROFIBUS PA

Las secuencias de señal teóricas de los bytes de Preámbulo, Delimitador de Inicio y Delimitador de Fin físicos que hacen parte de la trama en el protocolo PROFIBUS PA se muestran en la figura 41.

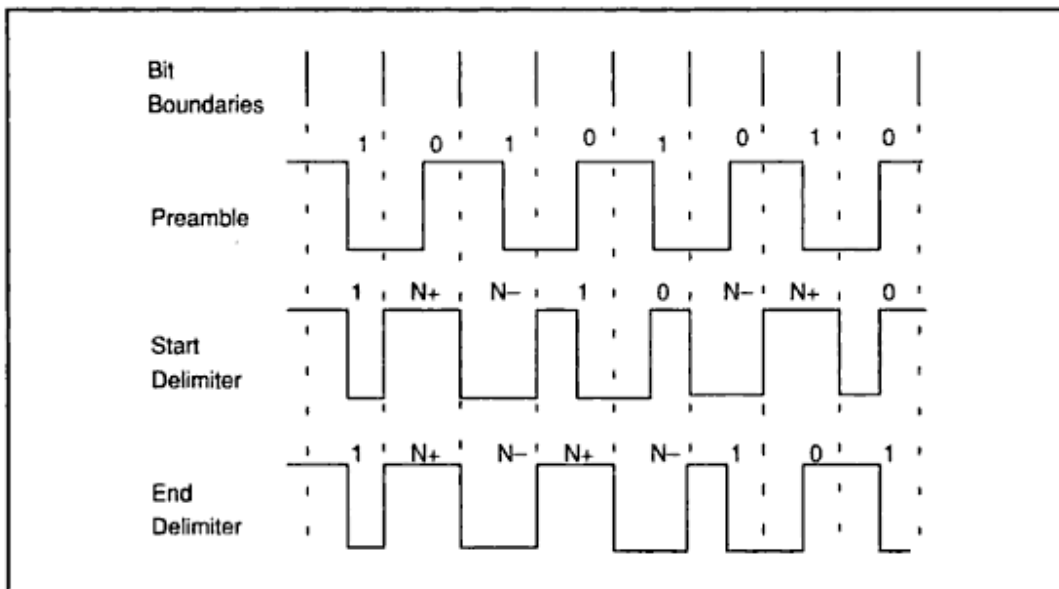


Figura 41. Secuencia de señal de los bytes de Preámbulo, Delimitador de Inicio y fin

En las figuras 42, 43 y 44 se puede observar las secuencias de señal obtenidas por medio de un osciloscopio del Preámbulo, Delimitador de Inicio y Delimitador de Fin físicos en las pruebas experimentales de recepción y transmisión realizadas, corroborando la comunicación correcta del dispositivo desarrollado bajo el protocolo PROFIBUS PA

La figura 42 muestra el preámbulo de la transmisión de datos.

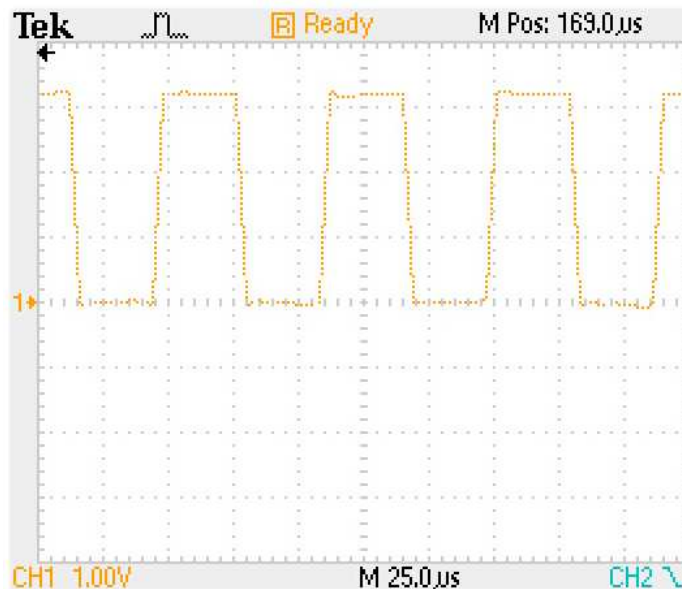


Figura 42. PREAMBULO

La figura 43 muestra el Start delimiter físico, byte que precede al preámbulo.

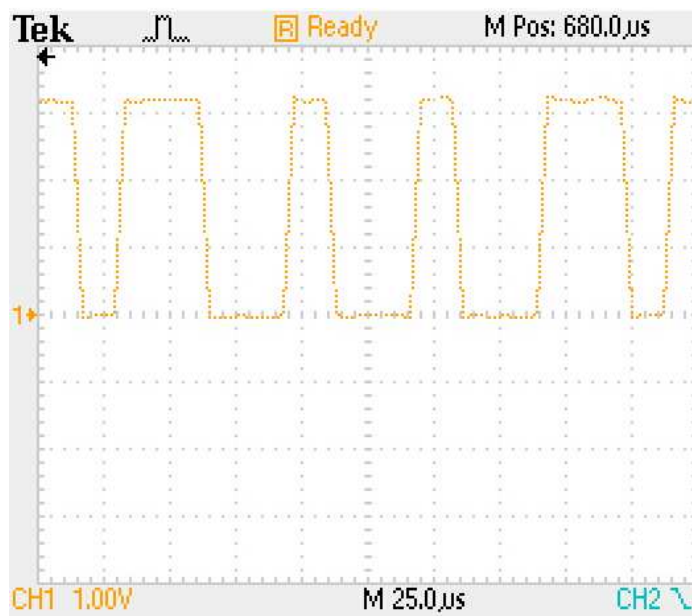


Figura 43. START DELIMITER FÍSICO

La figura 44 muestra el end delimiter físico, byte final de toda transmisión de datos.

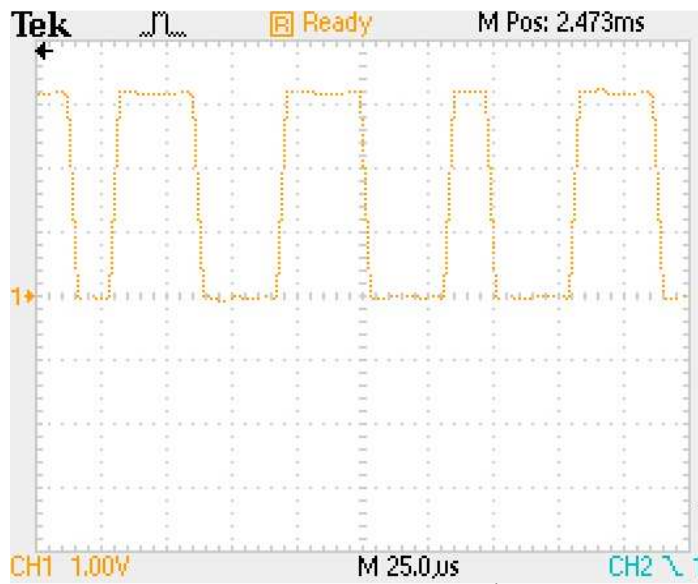


Figura 44. END DELIMITER FÍSICO

La figura 45 muestra el start delimiter SD1, empleado para respuestas a tramas provenientes del maestro con errores.

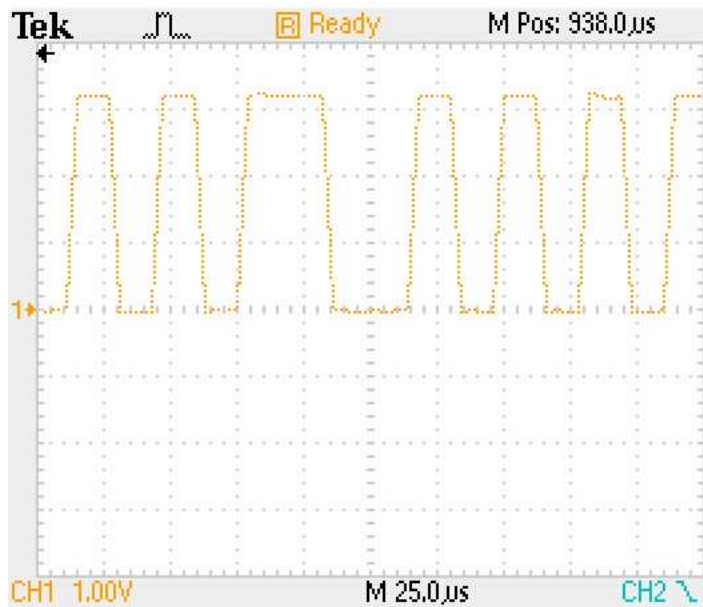


Figura 45. START DELIMITER (SD1=10H)

La figura 46 muestra el start delimiter SD2, empleado en tramas de respuesta a peticiones de diagnóstico

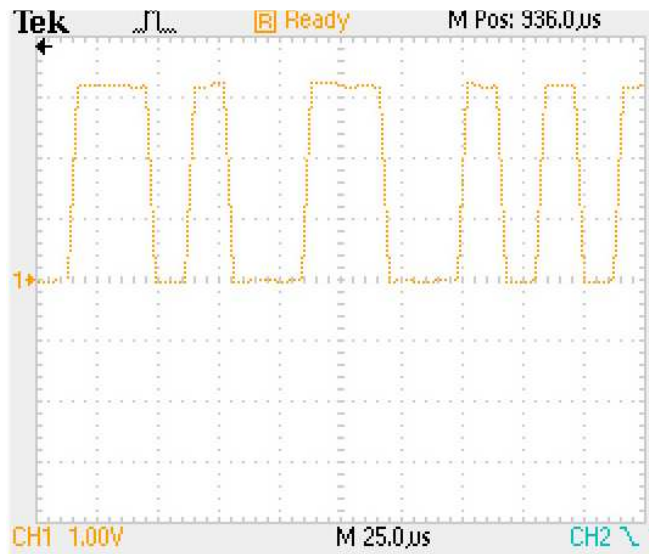


Figura 46. START DELIMITER (SD2=68H)

La figura 47 muestra el byte de short acknowledgement, empleado como respuesta desde el elemento esclavo (Convertidor de Profibus PA a 4-20mA) cuando se envían desde el maestro tramas de parametrización, configuración e intercambio de datos cíclico.

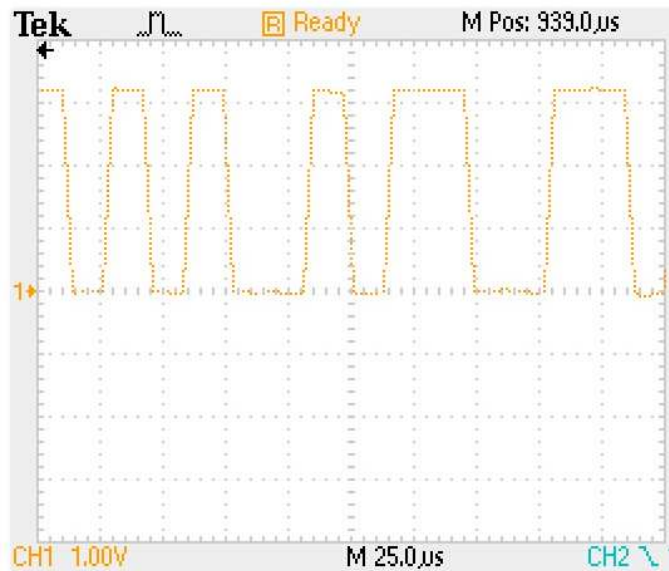


Figura 47. SHORT ACKNOWLEDGEMENT (E5H)

Las figuras 48 y 49 muestran los SAPs 60 y 62 incluidos en la trama de respuesta a una petición de diagnóstico desde el elemento maestro.

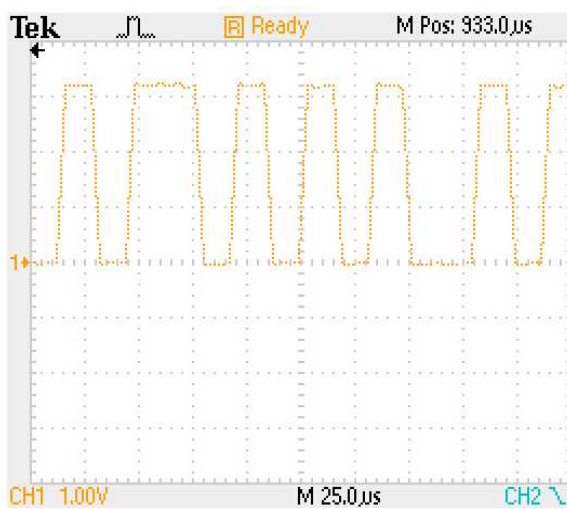


Figura 48. SAP 60(3CH)

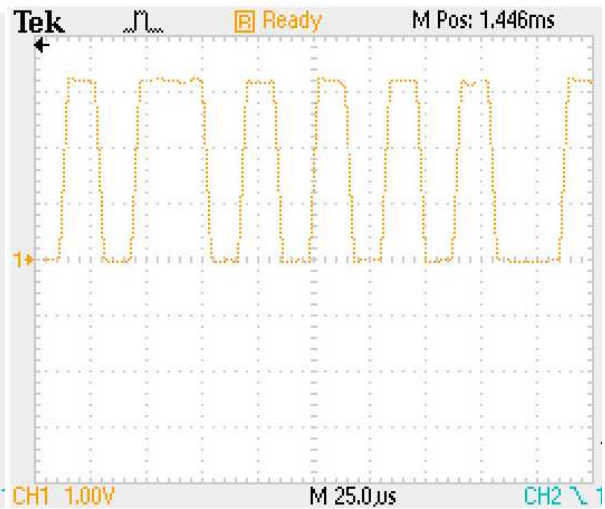


Figura 49. SAP 62(3EH)

6.2.5.2 ENVIO DE DATO EN CORRIENTE (ONLINE)

En el software de programación STEP 7, fue modificado el dato en corriente a ser enviado al Convertidor de PROFIBUS PA a 4-20mA estando en modo online, como se observa en la figura 50, un valor de corriente de 0.0065 (6.5mA).

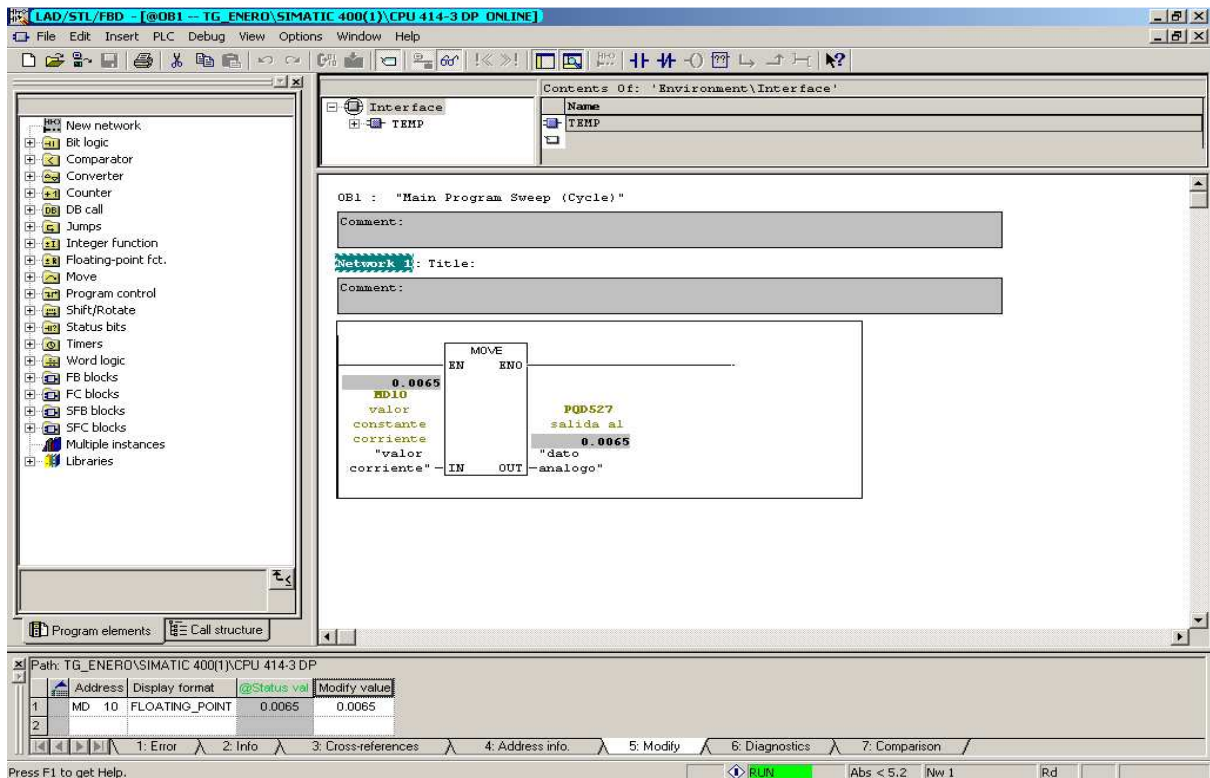


Figura 50. Comunicación PLC- convertidor interfaz Step7.

En la interfaz de programación del microcontrolador se observa el dato proveniente desde el PLC Siemens, corroborando así la correcta comunicación del Convertor en la red PROFIBUS implementada, como se puede evidenciar en la Figura 51.

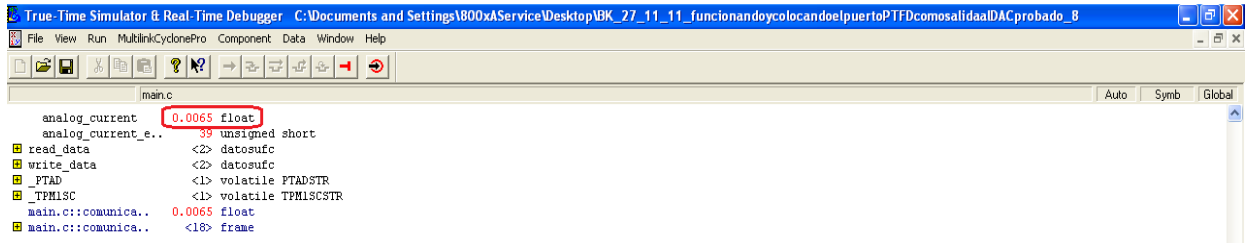


Figura 51. Comunicación PLC- convertidor interfaz Microcontrolador.

En la figura 52, se observa el valor análogo de corriente obtenido que concuerda con el dato enviado desde el PLC Siemens



Figura 52. Comunicación PLC- Salida Convertidor.

La figura 53, muestra la relación entre el voltaje a la salida del Convertor Digital Analógico AD558JN y la corriente obtenida

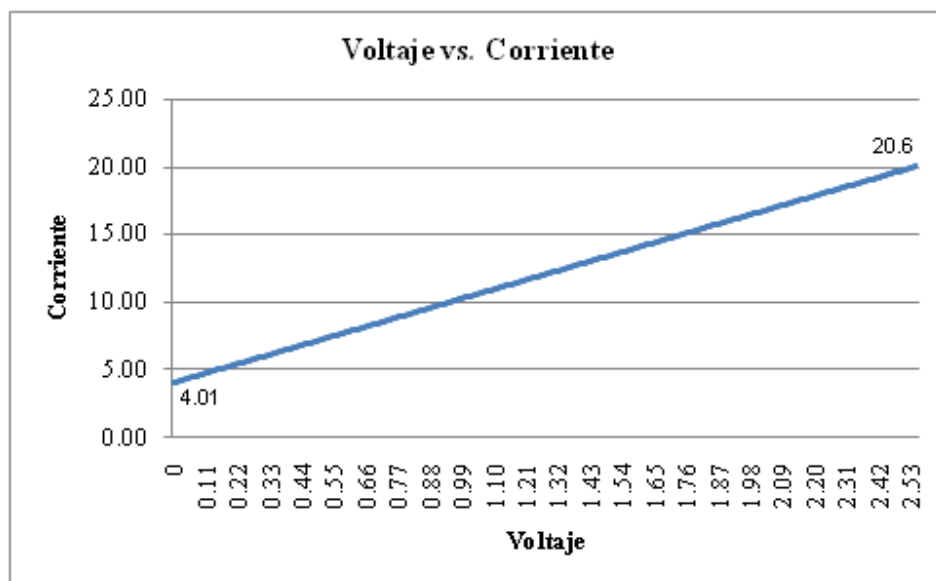


Figura 53. Voltaje vs. Corriente.

La figura 54, muestra la relación entre el valor de corriente medido y el valor de corriente esperado

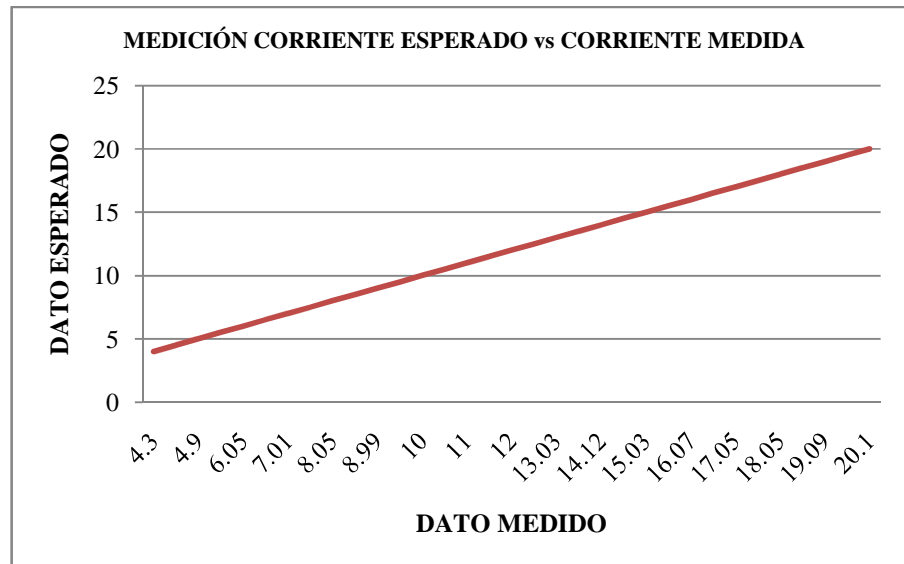


Figura 54. Corriente Esperada vs. Corriente Medida.

Al observar las gráficas 53 y 54 se comprueba la linealidad del Convertidor PROFIBUS PA a 4-20mA diseñado.

En la industria, por estándar la resolución mínima y comúnmente usada para dispositivos conversores de señales digitales a analógicas es de 12 bits. Teniendo en cuenta que el rango a ser convertido por el dispositivo es de 4-20mA se tiene que la resolución en corriente sería:

$$\# \text{ de bits} = 12$$

$$\text{Resolucion} = \frac{\text{Rango}}{2^n - 1} = \frac{20mA - 4mA}{2^{12} - 1}$$

$$\text{Resolucion} = 0,003907mA$$

Debido a que el Convertidor Digital Analógico empleado para el desarrollo de este proyecto posee una resolución de 8 bit, se tiene que la resolución en corriente es:

$$\# \text{ de bits} = 8$$

$$\text{Resolucion} = \frac{\text{Rango}}{2^n - 1} = \frac{20mA - 4mA}{2^8 - 1}$$

$$\text{Resolucion} = 0,0627mA$$

6.2.5.3 ANALISIS DE TRAMAS USANDO EL ANALIZADOR DE PROTOCOLO

Por medio del analizador de protocolo fueron corroboradas las tramas de recepción provenientes desde el elemento maestro (PLC SIEMENS) y las tramas de respuesta enviadas desde el elemento esclavo (Convertidor PROFIBUS PA a 4-20mA)

En la figura 55 se observan los estados presentes en la comunicación de acuerdo con la máquina de estados mostrada en el capítulo 2.8 conforme al protocolo PROFIBUS PA.

| No. | Time Stamp | Address | Protocol | Primitive | Service | Data |
|-----|-----------------|----------------|----------|-----------|----------------|-------------------|
| 563 | 12:41:41.560352 | 1 -> 0 | FDL | Request | FDL STATUS | |
| 564 | 12:41:41.575936 | 1 -> 1 | FDL | Request | TOKEN | |
| 565 | 12:41:41.581648 | 1 -> 4 | DP | Request | DATA EXCHANGE | |
| 566 | 12:41:41.591730 | 1.62 -> 9.60 | DP | Request | DIAGNOSIS | |
| 567 | 12:41:41.599878 | 1.62 <- 9.60 | DP | Response | DIAGNOSIS | 02 05 00 FF 97 00 |
| 568 | 12:41:41.611476 | 1 -> 1 | FDL | Request | TOKEN | |
| 569 | 12:41:41.617184 | 1 -> 4 | DP | Request | DATA EXCHANGE | |
| 570 | 12:41:41.627170 | 1.62 -> 9.61 | DP | Request | SET PARAMETERS | 88 0D 0F 0B 97 00 |
| 571 | 12:41:41.638600 | 1.62 <- 9.61 | FDL | Response | SC | |
| 572 | 12:41:41.645986 | 1 -> 2 | FDL | Request | FDL STATUS | |
| 573 | 12:41:41.661554 | 1 -> 1 | FDL | Request | TOKEN | |
| 574 | 12:41:41.667282 | 1 -> 4 | DP | Request | DATA EXCHANGE | |
| 575 | 12:41:41.677246 | 1.62 -> 9.62 | DP | Request | CHECK CONFIG | A4 |
| 576 | 12:41:41.685480 | 1.62 <- 9.62 | FDL | Response | SC | |
| 577 | 12:41:41.692850 | 1 -> 0 | FDL | Request | FDL STATUS | |
| 578 | 12:41:41.708434 | 1 -> 1 | FDL | Request | TOKEN | |
| 579 | 12:41:41.714146 | 1 -> 4 | DP | Request | DATA EXCHANGE | |
| 580 | 12:41:41.724226 | 1.62 -> 9.60 | DP | Request | DIAGNOSIS | |
| 581 | 12:41:41.732362 | 1.62 <- 9.60 | DP | Response | DIAGNOSIS | 80 0C 00 01 97 00 |
| 582 | 12:41:41.743858 | 1 -> 1 | FDL | Request | TOKEN | |
| 583 | 12:41:41.749570 | 1 -> 4 | DP | Request | DATA EXCHANGE | |
| 584 | 12:41:41.759650 | 1 -> 9 | DP | Request | DATA EXCHANGE | 00 00 00 00 00 |
| 585 | 12:41:41.769322 | 1 <- 9 | FDL | Response | SC | |
| 586 | 12:41:41.776706 | 1 -> 2 | FDL | Request | FDL STATUS | |
| 587 | 12:41:41.792274 | 1 -> 1 | FDL | Request | TOKEN | |
| 588 | 12:41:41.797986 | 1 -> 4 | DP | Request | DATA EXCHANGE | |
| 589 | 12:41:41.808066 | 1 -> 9 | DP | Request | DATA EXCHANGE | 00 00 00 00 00 |
| 590 | 12:41:41.817738 | 1 <- 9 | FDL | Response | SC | |
| 591 | 12:41:41.825090 | 1.62 -> 127.58 | DP | Request | GLOBAL CONTROL | 00 00 |
| 592 | 12:41:41.834090 | 1 <- 127 | FDL | Response | ACK NEG RS | |

Figura 55. Estados Comunicación cíclica- Analizador de Protocolo

En las líneas 566 y 567 detalladas en la Figura 56 se muestra la trama de primer diagnóstico enviada por el maestro y la trama de respuesta desde el convertidor

| | | | | | | |
|-----|-----------------|--------------|----|----------|-----------|-------------------|
| 566 | 12:41:41.591730 | 1.62 -> 9.60 | DP | Request | DIAGNOSIS | |
| 567 | 12:41:41.599878 | 1.62 <- 9.60 | DP | Response | DIAGNOSIS | 02 05 00 FF 97 00 |

| | | | | | | |
|---|--|--|--|--|--|--|
| <ul style="list-style-type: none"> PROFIBUS Frame <ul style="list-style-type: none"> Time:00:00:03.582278 Frame Type <ul style="list-style-type: none"> Type:SD2 Source address:1.62 Destination address:9.60 FDL Service <ul style="list-style-type: none"> Service:SRD_HIGH Primitive:Request DP Service | | | | | | |
|---|--|--|--|--|--|--|

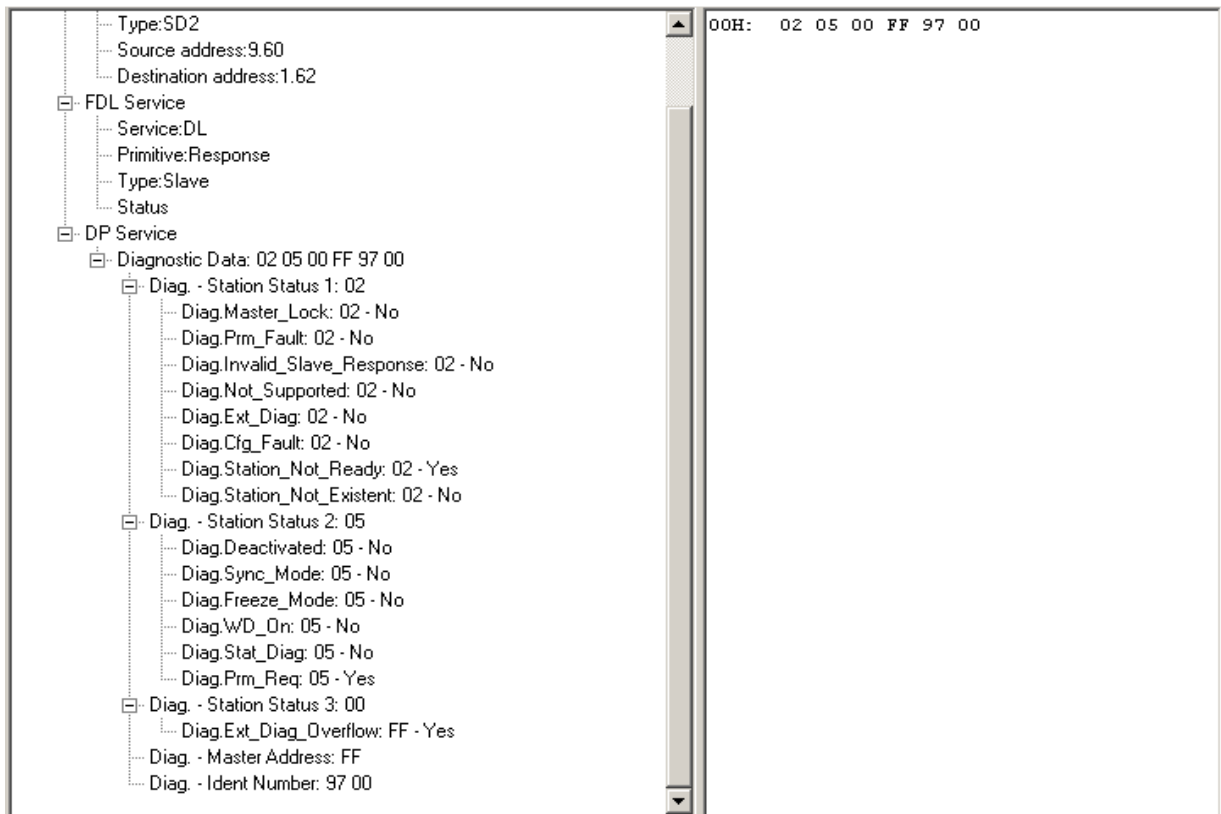


Figura 56. Primer Diagnostico visto en el analizador de Protocolo

En la figura 57 (líneas 570 y 571) se detalla la trama de parametrización enviada por el maestro y la trama de respuesta desde el convertidor (Short Acknowledgement).

| | | | | | | |
|-----|-----------------|--------------|-----|----------|----------------|-------------------|
| 570 | 12:41:41.627170 | 1.62 -> 9.61 | DP | Request | SET PARAMETERS | 88 0D 0F 0B 97 00 |
| 571 | 12:41:41.638600 | 1.62 <- 9.61 | FDL | Response | SC | |

The top screenshot shows a PROFIBUS Frame with the following details:

- Time: 00:00:39.804921
- Frame Type: SD2, Source address: 1.62, Destination address: 9.61
- FDL Service: SRD_HIGH, Primitive: Request
- DP Service: Set Param - Data: C8 0D 0F 0B 97 00 00
 - Status (Byte 1): 2
 - Lock & Unlock: 11 - Slave is unlocked for other masters
 - Sync_Req: 0 - The DP slave was indicated not to operate in Sync mode
 - Freeze_Request: 0 - The DP slave is indicated not to operate in Freeze mode
 - WD_On: 1 - Watchdog control is activated
 - Reserved: 0 - Reserved
 - Reserved: 0 - Reserved
 - Reserved: 0 - Reserved
 - Watchdog (Bytes 2 & 3): 0d 0f - T_WD = 10ms * WD_Fact_1 * WD_Fact_2
 - Min T_SDR: 0B - Minimum response time
 - Ident Number: 97 00 - Expected Ident Number
 - Group: 00 - Groups for Global_Control. Each bit represent a group. It is accepted only if the Lock_Req bit is set

The bottom screenshot shows a PROFIBUS Frame with the following details:

- Time: 00:00:52.447797
- Frame Type: SC, Source address: 9.61, Destination address: 1.62
- DP Service

Figura 57. Trama de Parametrización visto en el Analizador de Protocolo

En la figura 58 (líneas 575 y 576) se detalla la trama de configuración enviada por el maestro y la trama de respuesta desde el convertidor (Short Acknowledgement).

| | | | | | | |
|-----|-----------------|--------------|-----|----------|--------------|----|
| 575 | 12:41:41.677346 | 1.62 -> 9.62 | DP | Request | CHECK CONFIG | A4 |
| 576 | 12:41:41.685480 | 1.62 <- 9.62 | FDL | Response | SC | |

The screenshot shows a PROFIBUS Frame with the following details:

- Time: 00:00:03.850904
- Frame Type: SD2, Repeated frame: Yes, Source address: 1.62, Destination address: 9.62
- FDL Service: SRD_HIGH, Primitive: Request
- DP Service: Check Config - Data: A4
 - Module
 - Simple Identifier: A4
 - Consistency Over: 1 - Whole Length
 - Length Format: 0 - Byte
 - Input/Output: 02 - Output
 - Data Length: 04
 - Data



Figura 58. Tramas de Configuración y Respuesta visto en el Analizador de Protocolo

En la figura 59 (líneas 580 y 581) se detalla la trama de segundo diagnóstico enviada por el maestro y la trama de respuesta desde el convertidor, dado que la parametrización y configuración realizada por el maestro fue correcta.

| | | | | | | |
|-----|-----------------|--------------|----|----------|-----------|-------------------|
| 580 | 12:41:41.724226 | 1.62 -> 9.60 | DP | Request | DIAGNOSIS | |
| 581 | 12:41:41.732362 | 1.62 <- 9.60 | DP | Response | DIAGNOSIS | 80 0C 00 01 97 00 |

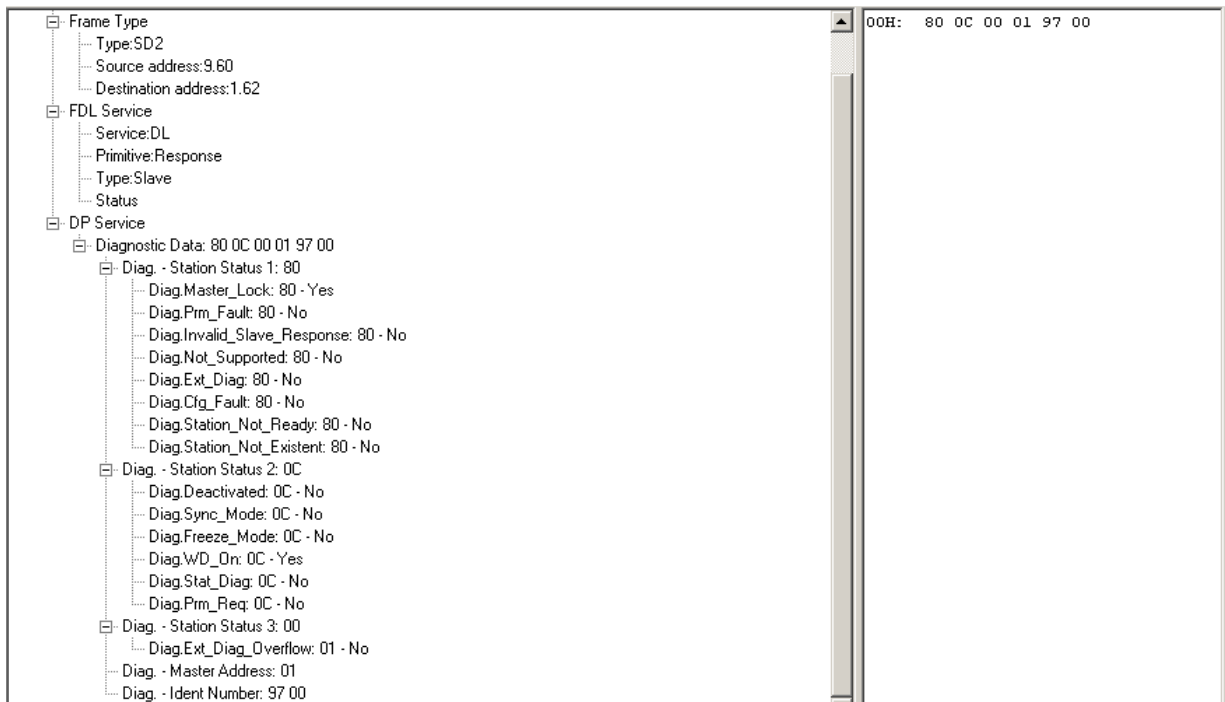


Figura 59. Tramas de Segundo Diagnostico y Respuesta visto en el Analizador de Protocolo

En la figura 60 (líneas 584, 585, 589 y 590) se detalla el inicio del intercambio de datos cíclico entre el maestro y el esclavo siendo la respuesta de éste último un Short acknowledgement (SC) en cada ciclo.

| | | | | | | |
|-----|-----------------|--------|-----|----------|---------------|----------------|
| 584 | 12:41:41.759650 | 1 -> 9 | DP | Request | DATA EXCHANGE | 00 00 00 00 00 |
| 585 | 12:41:41.769322 | 1 <- 9 | FDL | Response | SC | |
| 589 | 12:41:41.808066 | 1 -> 9 | DP | Request | DATA EXCHANGE | 00 00 00 00 00 |
| 590 | 12:41:41.817738 | 1 <- 9 | FDL | Response | SC | |

Figura 60. Intercambio de datos visto en el Analizador de Protocolo

En la figura 61 se muestra un intercambio de datos constante entre el maestro y el Convertidor PROFIBUS PA a 4-20mA.

Figura 61. Intercambio de datos estable

En la figura 62 se observan los dispositivos activos en la red PROFIBUS implementada

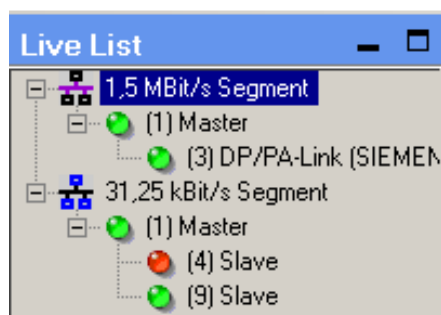


Figura 62. Dispositivos activos en la red PROFIBUS

6.2.6 DIFERENCIAS ENTRE FI303-SMAR Y CONVERTIDOR PROFIBUS PA A 4-20mA

Algunos fabricantes han desarrollado interfaces para PROFIBUS PA con base a la norma descrita, y sus especificaciones. Un ejemplo es el convertidor de PROFIBUS PA a corriente FI303 creado por SMAR, que permite la comunicación entre los dispositivos en campo y la sala de control, incluyendo algunas características que reducen considerablemente los costos de instalación, operación y mantenimiento. [20] En este proyecto se tomaron como referencia algunas especificaciones de este dispositivo, teniendo en cuenta que tendrán algunas diferencias como se muestra en la tabla 27:

| | FI303-SMAR | Convertidor PROFIBUS PA a 4-20mA |
|--------------------------------|---|------------------------------------|
| Salida | Tres salidas de corriente 4-20mA | Una salida de corriente 4-20mA |
| Alimentación Eléctrica. | A través del Bus PROFIBUS PA, 9 - 32 Vdc. | A través de PROFIBUS PA, 24Vdc±10% |

Tabla 27. Diferencias entre FI303-SMAR y Convertidor PROFIBUS PA a 4-20mA

Por ser un prototipo, especificaciones como configuración, material, montaje, peso, certificación de área clasificada, límites de temperatura y humedad, no fueron tomadas en cuenta para este proyecto.

7. CONCLUSIONES

El adecuado funcionamiento del medio físico es de gran importancia para asegurar la comunicación del convertidor con la red PROFIBUS, ya que si existen fallas en el circuito de interfaz con el medio (MAU) en cuanto a voltajes de referencia y corriente del bus, la comunicación con el dispositivo maestro es errada; por lo tanto es importante verificar las ecuaciones y topología del circuito implementado.

La construcción del convertidor PROFIBUS PA a 4-20mA no solo permitió adquirir gran conocimiento en cuanto a la arquitectura de las redes PROFIBUS, sino en cuanto la construcción de las tramas específicas de acuerdo a las normas establecidas para comunicación con este tipo de protocolo. Además, se puede notar el gran campo de aplicación que tienen dispositivos como el convertidor PROFIBUS PA a 4-20mA, para la integración y centralización de redes industriales.

De igual manera, es de resaltar el arduo trabajo investigativo que se llevó a cabo junto con otros proyectos de grado (*CONVERTIDOR DE 4-20mA A PROFIBUS PA- TG1003*) para reunir la información teórica suficiente para la comprensión de las redes PROFIBUS y las tramas que la componen, ya que por lo menos a nivel Colombia este tipo de redes no son muy comunes y por lo tanto la documentación en este país es escasa. Así mismo fue necesario realizar un análisis minucioso de los circuitos de aplicación propuestos por los fabricantes, pues su montaje y funcionamiento no fue inmediato por las variables que se expusieron en este documento.

El alcance de este proyecto llegó hasta el establecimiento de un ciclo de comunicación entre un maestro DP y un esclavo PA, usando una tecnología muy común en la industria como lo es el lazo de 4-20mA. Este proyecto no contempla el intercambio de parámetros acíclico para funciones de diagnóstico y configuración, así como funciones de alarmas y eventos, por lo que se abre la ventana para que futuros proyectos profundicen en el tema.

Fue necesario emplear una fuente externa para la alimentación de la etapa de conversión de voltaje a corriente, ya que el regulador de 5V presente en la MAU, no soporta el nivel de carga entregado, reduciendo el voltaje en un 30%.

Debido al gran crecimiento que ha tenido el sistema PROFIBUS en los procesos de control y automatización industrial a nivel mundial, empresas que deseen implementar este sistema para la modernización de sus plantas, podrán adaptar los dispositivos de campo existentes que manejen señales de 4-20mA empleando el Convertidor PROFIBUS PA a 4-20mA desarrollado en este proyecto, sin necesidad de cambiarlos, reduciendo costos en gran medida y obteniendo a su vez muchos beneficios bajo la implementación de tecnologías modernas.

Teniendo en cuenta la gran gama de aplicaciones y compatibilidad del protocolo con otros sistemas de comunicación implementados en la industria (como ETHERNET- PROFINET), este proyecto podría ser tomado como base teórico-práctica para ampliar el alcance de las prácticas de laboratorio de comunicación que se podrían realizar con el equipo disponible en la Universidad, verificando la interacción de PROFIBUS con otras redes disponibles en la industria.

8. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

- [1] http://www.automation.siemens.com/w2/efiles/pcs7/pdf/78/prdbrief/kb_profibus_es.pdf
[online] PROFIBUS-El bus polivalente para la comunicación en la industria de procesos.
Folleto · Abril 2008.Siemens. Consultada: 13/11/2010
- [2] <http://www.smar.com/espanol/profibus.asp>[online].PROFIBUS. Consultada: 17/12/2009
- [3] GARCÍA, Andrés. El Control Automático En La Industria. Ediciones de la Universidad de Castilla-la mancha. España. 2005
- [4] Diedrich/ Bangemann. PROFIBUS PA. Instrumentation Technology for the Process Industry.Oldenbourg Industrieverlag.2007
- [5] <http://upcommons.upc.edu/pfc/bitstream/2099.1/4298/10/Profibus.pdf>
- [6] LAPORTA, Jorge, MIRALLES, Marcel. Fundamentos de Telemática. Editorial Universidad Politécnica de Valencia.2005
- [7] DERICKSON, Dennis, MULLER, Marcus. Digital communications test and measurement high-speed physical layer characterization.Ed.Prentice Hall.2007
- [8] http://www.dte.upct.es/personal/manuel.jimenez/docencia/GD6_Comunic_Ind/pdfs/Tema%208.pdf
[online]. PROFIBUS (DP/PA/FMS). Universidad Politécnica de Cartagena. Dpto. de Tecnología Electrónica, Comunicaciones Industriales. Consultada: 28/10/2009
- [9] http://www.frm.utn.edu.ar/comunicaciones/modelo_osi.html. Modelo de referencia OSI. Interconexión de sistemas abiertos
- [10] http://www.samson.de/pdf_en/l453en.pdf. SAMSON AG PROFIBUS PA Thecnical Information Article 1999.
- [11] AGROMAG, INTRODUCTION TO PROFIBUS DP. Referencia Técnica Año 2002
- [12] www.onsemi.com AMIS-492x0 Field Bus MAU. June 2008-Rev.6
- [13] <http://www.softing.com> [online] Unified Fieldbus Controller. UFC - Datasheet.Aniotek Inc./Softing AG. Rev 1.0. Junio 5 del 2007. Consultada: 18/06/2010.

- [14] http://www.datasheetcatalog.org/datasheet/analogdevices/263100547AD558_a.pdf. Consultada: 26/06/2011
- [15] http://www.diselectric.com/descarregues/SIEMENS/PROFIBUS/MANUAL_PA.pdf [online].Manual SIMATIC, DP/PA bus coupler.Consultada: 15/01/2011
- [16] www.siemens.com.co. Manual SIMATIC- Working with STEP 7 (A5E02904800-0105/2010). Consultada: 31/10/2010.
- [17] www.profibus.com.Specification for PROFIBUS- Device Description and Device Integration, Volume 1: GSD(**GSD-Spec_2122_V51_Jul08**). Consultada: 17/12/2010
- [18] www.softing.com. [online]Technical Article.Protocol analysis for fieldbuses An ear to PROFIBUS.Dr.-Ing. Hans Endl is the product manager for diagnostic products at Softing AG. Consultada: 16/11/2009
- [19] <http://www.softing.com>. [online] PROFIBUS Protocol Analyzer Product Information. Consultada: 16/11/2009
- [20] <http://www.smar.com/espanol/products/detail.asp?prod=35> [online] FI303 Convertidor de Corriente PROFIBUS PA. Consultada 03/10/2009

9. ANEXOS

9.1 CODIGO FUENTE

```
/******  
*      MAIN.C      *  
*****/  
#include <hides.h> /* for EnableInterrupts macro */  
#include "derivative.h" /* include peripheral declarations */  
#include "read_write_ufc.h"  
#include "rx_tx_control.h"  
#include "structures.h"  
  
void MCU_init(void); /* Device initialization function declaration */  
  
    frame_control communication_frame;  
datosufc read_data,write_data;  
  
void main(void) {  
    MCU_init(); /* call Device Initialization */  
  
    PTAD_PTAD2=0; /*Activacion de senal de RESET para la inicializacion de la UFC*/  
  
    PTAD_PTAD2=1;  
    initialize_ufc(&comunication_frame); /*Rutina de inicializacion de la UFC*/  
  
    /* include your code here */  
  
    for(;;) {  
  
        if (TPM1SC_TOF)  
            {  
                TPM1SC_TOF=~TPM1SC_TOF;  
            }  
  
        if (!PTAD_PTAD3) /*Condicion de senal de interrupcion INTn en bajo*/  
            {  
                interrupt_ufc(&comunication_frame);  
            }  
  
        /* __RESET_WATCHDOG(); by default, COP is disabled with device init. When enabling, also reset  
the watchdog. */  
        } /* loop forever */  
        /* please make sure that you never leave main */  
    }  
  
/******  
*      MAIN.H      *  
*****/  
  
#if !defined(MAIN_H)
```

```

#define MAIN_H

#endif

/*****
*   READ_WRITE_UFC.C   *
*****/

#include "derivative.h"
#include "main.h"
#include "read_write_ufc.h"

void read_ufc (datosufc *registers)/*Rutina de lectura de datos desde la UFC*/
{
    PTAD_PTAD1=((registers->address)&(0x01)) && (0x01);
    PTBD_PTBD1=((registers->address)&(0x02)) && (0x01);
    PTBD_PTBD2=((registers->address)&(0x04)) && (0x01);
    PTBD_PTBD3=((registers->address)&(0x08)) && (0x01);
    PTBD_PTBD4=((registers->address)&(0x10)) && (0x01);

    PTDD_PTDD6=1;/*Senal de seleccion de lectura/escritura (RWn)*/
    PTDD_PTDD4=0;/*Senal de habilitacion de acceso de lectura o escritura (CSn)*/
    PTDD_PTDD5=0;/*Senal de flash de lectura (DSn)*/

    registers->data=PTCD;

    PTDD_PTDD5=1;/*Senal de flash de lectura (DSn)*/
    PTDD_PTDD4=1;/*Senal de habilitacion de acceso de lectura o escritura (CSn)*/
    PTDD_PTDD6=0;/*Senal de seleccion de lectura/escritura (RWn)*/
}

void write_ufc (datosufc *registers) /*Rutina de escritura de datos hacia la UFC*/
{
    PTAD_PTAD1=((registers->address)&(0x01)) && (0x01);
    PTBD_PTBD1=((registers->address)&(0x02)) && (0x01);
    PTBD_PTBD2=((registers->address)&(0x04)) && (0x01);
    PTBD_PTBD3=((registers->address)&(0x08)) && (0x01);
    PTBD_PTBD4=((registers->address)&(0x10)) && (0x01);

    PTDD_PTDD6=0;/*Senal de seleccion de lectura/escritura (RWn)*/
    PTDD_PTDD4=0;/*Senal de habilitacion de acceso de lectura o escritura (CSn)*/
    PTDD_PTDD5=0;/*Senal de flash de lectura (DSn)*/

    PTCD=registers->data;

    PTDD_PTDD5=1;/*Senal de flash de lectura (DSn)*/
    PTDD_PTDD4=1;/*Senal de habilitacion de acceso de lectura o escritura (CSn)*/
    PTDD_PTDD6=1;/*Senal de seleccion de lectura/escritura (RWn)*/
}

/*****
*   READ_WRITE_UFC.H   *
*****/

```

```

#if !defined(READ_WRITE_UFC_H);
#define READ_WRITE_UFC_H
#include <stddef.h>

typedef struct datosufc datosufc; /*Estructura de datos para lectura/escritura con la UFC*/
struct datosufc
{
unsigned char address,
data;
};

void read_ufc (datosufc *registro);
void write_ufc (datosufc *registro);

# endif

/*****
*      STRUCTURES.H      *
*****/

#if !defined(STRUCTURES_H)
#define STRUCTURES_H
#include <stddef.h>

typedef struct frame frame; /*estructura de la trama PROFIBUS PA*/
struct frame
{
unsigned char SD,LE,LEr,DA,SA,FC,SSAP,DSAP,DATAUNIT[9];
};

typedef struct frame_control frame_control;
struct frame_control
{
frame frame_receive, frame_transmit;
datosufc read_data, write_data;
unsigned char error_stat, /*Variable de estado de error*/
interrupt_stat, /*Variable de estado de interrupciones*/
stat, /*Variable de estado de la UFC*/
FIFO_stat, /*Variable de estado de las FIFO de recepción y transmisión*/
transmit_length_ufc, /*Longitud de bytes a ser escritos a la UFC*/
first_diagnostic, /*bandera de primer diagnóstico*/
cyclic_data[4], /*variable de almacenamiento de datos en data exchange enviados por el PLC*/
address_check; /*bandera de chequeo de dirección del esclavo enviada por el PLC*/
char diag_data_frame[9]; /*Bytes de diagnóstico*/
unsigned char DAC_analog_current[2]; /*variable donde se separa el dato de corriente después de haber
sido convertido a entero*/
float analog_current; /*dato de corriente análogo en formato flotante*/
signed short analog_current_escal; /*dato en corriente escalizado*/
};
# endif

/*****

```



```

*      RX_TX_CONTROL.C      *
*****/

#include "main.h"
#include "derivative.h"
#include "read_write_ufc.h"
#include "structures.h"
#include "rx_tx_control.h"
#include <stddef.h>
#include <math.h>

void initialize_ufc(frame_control *handle_frame)/*Rutina de inicialización de la UFC*/
{
    PTAD_PTAD2=0;/*Activación de señal de RESET para la inicialización de la UFC*/
    while (!TPM1SC_TOF){}
    PTAD_PTAD2=1;

    PTDD_PTDD4=1;/*Señal de habilitación de acceso de lectura o escritura (CSn)*/
    PTDD_PTDD5=1;/*Señal de flash de lectura (DSn)*/
    PTCDD=0xFF;/*Habilitación de los puertos de datos con la UFC como salida*/

/*RESET REGISTER*/
    handle_frame->write_data.address=0x00;
    handle_frame->write_data.data=0x80;
    write_ufc(&(handle_frame->write_data));

/*MODE REGISTER*/
    handle_frame->write_data.address=0x01;
    handle_frame->write_data.data=0x04;
    write_ufc(&(handle_frame->write_data));

/*INTERRUPT MASK*/
    handle_frame->write_data.address=0x06;
    handle_frame->write_data.data=0x02;

/*ERROR MASK*/
    handle_frame->write_data.address=0x07;
    handle_frame->write_data.data=0x00;
    write_ufc(&(handle_frame->write_data));

/*FIFO CONTROL REGISTER*/
    handle_frame->write_data.address=0x0B;
    handle_frame->write_data.data=0x43;
    write_ufc(&(handle_frame->write_data));

/*CLOCK MODE REGISTER*/
    handle_frame->write_data.address=0x10;
    handle_frame->write_data.data=0x00;
    write_ufc(&(handle_frame->write_data));

/*GAP TIME REGISTER*/

```

```

handle_frame->write_data.address=0x14;
handle_frame->write_data.data=0x10;
write_ufc(&(handle_frame->write_data));

/*WATCH TIMER REGISTER (LOW BYTE)*/
handle_frame->write_data.address=0x16;
handle_frame->write_data.data=0x00;
write_ufc(&(handle_frame->write_data));

/*WATCH TIMER REGISTER (HIGHT BYTE)*/
handle_frame->write_data.address=0x17;
handle_frame->write_data.data=0x20;
write_ufc(&(handle_frame->write_data));

/*TIMER CONTROL REGISTER*/
handle_frame->write_data.address=0x1A;
handle_frame->write_data.data=0x02;
write_ufc(&(handle_frame->write_data));

/*CONTROL REGISTER*/
handle_frame->write_data.address=0x02;
handle_frame->write_data.data=0x41;/*activa IE bit para que todas las interrupciones no enmascaradas
activen INTn*/
write_ufc(&(handle_frame->write_data));

handle_frame->first_diagnostic=0x00;/*Enceramiento de bandera de primer diagnóstico*/
handle_frame->interrupt_stat=0x00;/*Enceramiento de variable de estado de interrupciones*/
handle_frame->error_stat=0x00;/*Enceramiento de variable de estado de errores*/
handle_frame->FIFO_stat=0x00;/*Enceramiento de variable de estado de las FIFO de recepción y
transmisión*/
handle_frame->stat=0x00;/*Enceramiento de variable de estado de la UFC*/
}

void interrupt_ufc(frame_control *handle_frame)
{
char i=0, transmit=0,diagnostic=0,parameterization=0,configuration=0, data_exchange=0,
data_exchange_state=0, state_machine=0, new_cycle=0, frame_DSAP=0, address_check=0,
address_right=0, SD1=0;
float xmax=255,ymax=0.016,b=0.004;/*Inicializacion de variables para escalizacion del dato de corriente
enviado por el PLC*/
unsigned short current;
handle_frame->frame_receive.DA=0x00;/*Encerramiento de Destination Address*/

PTCDD=0x00;/*Habilitacion de los puertos de datos con la UFC como entrada*/

handle_frame->read_data.address=0x04;/*Lectura del registro de interrupt status*/
read_ufc(&(handle_frame->read_data));
handle_frame->interrupt_stat=handle_frame->read_data.data;

handle_frame->read_data.address=0x05;/*Lectura del registro de error status*/
read_ufc(&(handle_frame->read_data));
handle_frame->error_stat=handle_frame->read_data.data;

```

```

handle_frame->read_data.address=0x03; /*Lectura del registro de status*/
read_ufc(&(handle_frame->read_data));
handle_frame->stat=handle_frame->read_data.data;

handle_frame->read_data.address=0x0C; /*Lectura del registro de FIFO status*/
read_ufc(&(handle_frame->read_data));
handle_frame->FIFO_stat=handle_frame->read_data.data;

if ((handle_frame->interrupt_stat)&0x20) /*Si se presenta interrupción por transmisión*/
{
    PTCDD=0xFF; /*Habilitacion de los puertos de datos con la UFC como salida*/
    handle_frame->write_data.address=0x0B; /*Escritura al registro FIFO control para resetear la FIFO de
recepcion*/
    handle_frame->write_data.data=0x73;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x02; /*Escritura al registro de control para habilitar recepción y
activar IE bit para que todas las interrupciones no enmascaradas activen INTn*/
    handle_frame->write_data.data=0x41;
    write_ufc(&(handle_frame->write_data));
}

if ((handle_frame->interrupt_stat)&0x10) /*Si se presenta interrupción por recepción*/
{
    PTCDD=0x00; /*Habilitación de los puertos de datos con la UFC como entrada*/
    handle_frame->read_data.address=0x0D; /*Lectura de dato almacenado en la FIFO de recepción*/
    read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.SD=handle_frame->read_data.data; /*Almacenamiento del Start
Delimiter*/

    if ((handle_frame->frame_receive.SD)==0x10) /*Si el Start Delimiter es SD1*/
    {
        handle_frame->read_data.address=0x0D; /*Lectura de dato almacenado en la FIFO de recepción*/
        read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.DA=handle_frame->read_data.data; /*Almacenamiento Del Destination
Address*/

        handle_frame->read_data.address=0x0D; /*Lectura de dato almacenado en la FIFO de recepción*/
        read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.SA=handle_frame->read_data.data; /*Almacenamiento del Source
Address*/

        handle_frame->read_data.address=0x0D; /*Lectura de dato almacenado en la FIFO de recepción*/
        read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.FC=handle_frame->read_data.data; /*Almacenamiento del Function
Code*/
    }

    if ((handle_frame->frame_receive.SD)==0xA2) /*Si el Start Delimiter es SD3*/
    {
        handle_frame->read_data.address=0x0D; /*Lectura de dato almacenado en la FIFO de recepción*/

```

```

read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.DA=handle_frame->read_data.data;/*Almacenamiento del Destination
Address*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.SA=handle_frame->read_data.data;/*Almacenamiento del Source
Address*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.FC=handle_frame->read_data.data;/*Almacenamiento del Function
Code*/

if (((handle_frame->frame_receive.DA)&0x80) && ((handle_frame->frame_receive.SA)&0x80))/*Si el
bit más significativo de DA y SA es 1 indica que hay DSAP y SSAP*/
    {
    handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
    read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.DSAP=handle_frame->read_data.data;/*Almacenamiento del
Destination Service Access Point*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.SSAP=handle_frame->read_data.data;/*Almacenamiento del Source
Service Access Point*/

for (i=0;i<6;i++)
    {
    handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
    read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.DATAUNIT[i]=handle_frame->read_data.data;/*Almacenamiento
del Data Unit*/
    }
        i=0;
        frame_DSAP=1;/*Activa bandera que indica que en la trama fue recibido un DSAP*/
    }
else /*Si el bit más significativo de DA y SA no es 1 indica que no hay DSAP y SSAP, es decir, se entra
en modo de data exchange*/
    {
    for (i=0;i<8;i++)
        {
        handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
        read_ufc(&(handle_frame->read_data));
            handle_frame->frame_receive.DATAUNIT[i]=handle_frame->read_data.data;/*Almacenamiento
del Data Unit*/
        }
            i=0;
            frame_DSAP=0;/*Desactiva bandera que indica que en la trama fue recibido un DSAP*/
        }
    }

if ((handle_frame->frame_receive.SD)==0x68)/*Si el Start Delimiter es SD2*/

```

```

{
handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.LE=handle_frame->read_data.data;/*Almacenamiento del Lenght data
net*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.LEr=handle_frame->read_data.data;/*Almacenamiento del Lenght data
net repetido*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.DA=handle_frame->read_data.data;/*Almacenamiento del Destination
Address*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.SA=handle_frame->read_data.data;/*Almacenamiento del Source
Address*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.FC=handle_frame->read_data.data;/*Almacenamiento del Function
Code*/

if (((handle_frame->frame_receive.DA)&0x80) && ((handle_frame->frame_receive.SA)&0x80))
{
    handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
    read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.DSAP=handle_frame->read_data.data;/*Almacenamiento del
Destination Service Access Point*/

handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.SSAP=handle_frame->read_data.data;/*Almacenamiento del Source
Service Access Point*/

for (i=0;i<(handle_frame->frame_receive.LE)-5;i++)
{
    handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
    read_ufc(&(handle_frame->read_data));
        handle_frame->frame_receive.DATAUNIT[i]=handle_frame->read_data.data;/*Almacenamiento
del Data Unit*/
    }
    i=0;
    frame_DSAP=1;/*Activa bandera que indica que en la trama fue recibido un DSAP*/
}
else
{

```

```

for (i=0;i<(handle_frame->frame_receive.LE)-3;i++)
{
handle_frame->read_data.address=0x0D;/*Lectura de dato almacenado en la FIFO de recepción*/
read_ufc(&(handle_frame->read_data));
    handle_frame->frame_receive.DATAUNIT[i]=handle_frame->read_data.data;/*Almacenamiento
del Data Unit*/

}
    frame_DSAP=0;/*Desactiva bandera que indica que en la trama fue recibido un DSAP*/
    data_exchange=1;/*Activa bandera que indica paso al modo de intercambio de datos cíclico*/
}
}
if ((handle_frame->frame_receive.SD)==0xDC)/*Si el Start Delimiter es 0xDC*/
    {}/*Start delimiter no activado*/

if ((handle_frame->frame_receive.SD)==0x00)/*Si el Start Delimiter es 0x00*/
    {}/*Start delimiter no activado*/

if ((handle_frame->frame_receive.SD)==0xE5)/*Si el Start Delimiter es Short Acknowledgement*/
    {}/*Start delimiter no activado*/

    PTCDD=0xFF;/*Habilitacion de los puertos de datos con la UFC como salida*/

handle_frame->write_data.address=0x0B;/*Escritura al registro FIFO control para resetear la FIFO de
recepcion*/
handle_frame->write_data.data=0x73;
    write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x02;/*Escritura al registro de control para habilitar recepción y
activar IE bit para que todas las interrupciones no enmascaradas activen INTn*/
handle_frame->write_data.data=0x41;
    write_ufc(&(handle_frame->write_data));
} else
{
    PTCDD=0xFF;/*Habilitacion de los puertos de datos con la UFC como salida*/

handle_frame->write_data.address=0x0B;/*Escritura al registro FIFO control para resetear la FIFO de
recepcion*/
handle_frame->write_data.data=0x73;
    write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x02;/*Escritura al registro de control para habilitar recepción y
activar IE bit para que todas las interrupciones no enmascaradas activen INTn*/
handle_frame->write_data.data=0x41;
    write_ufc(&(handle_frame->write_data));
}

    address_check=handle_frame->frame_receive.DA;/*Asignacion del byte de Destination Address*/
address_check &=0x7F;/*Coloca en cero el bit mas significativo de Destination Address para
comparacion*/

if ((address_check)==0x09 || (address_check)==0x7F) /*Revisa si Destination Address recibida concuerda
con la direccion asignada al esclavo (0x09)*/

```

```

{
if (handle_frame->frame_receive.FC==0x6D)/*Si el Function Code es 0x6D*/
{
handle_frame->first_diagnostic=0x00;/*Activa bandera de primero diagnostico*/
}
}
if ((handle_frame->frame_receive.FC & 0x0F)==0x09) /*Si los cuatro primeros bits del Function Code
son 1*/
{
SD1=1;
handle_frame->frame_transmit.SD=0x10;
handle_frame->frame_transmit.DA=handle_frame->frame_receive.SA;
handle_frame->frame_transmit.SA=handle_frame->frame_receive.DA;
handle_frame->frame_transmit.FC=0x08;
handle_frame->transmit_length_ufc=1;/*Numero de bytes a ser escritos a la UFC a ser transmitidos*/
transmit=1; /*Activa bandera de transmision*/
}
address_right=1;/*Activa bandera de direccion correcta*/
} /*end if direccion del esclavo es 0x09*/
else/*si Destination Address recibida no concuerda con la direccion asignada al esclavo (0x09)*/
{
address_right=0;/*Desactiva bandera de direccion correcta*/
transmit=0;/*Desactiva bandera de transmision*/

PTCDD=0xFF;/*Habilitacion de los puertos de datos con la UFC como salida*/

handle_frame->write_data.address=0x0B;/*Escritura al registro FIFO control para resetear la FIFO de
recepcion*/
handle_frame->write_data.data=0x73;
write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x02;/*Escritura al registro de control para habilitar recepción y
activar IE bit para que todas las interrupciones no enmascaradas activen INTn*/
handle_frame->write_data.data=0x41;
write_ufc(&(handle_frame->write_data));
}

if (address_right==1)/*Si la direccion es correcta*/
{
if (frame_DSAP==1)/*Si fue recibido un DSAP*/
{
if ((handle_frame->frame_receive.DSAP)!=0x3C && (handle_frame->frame_receive.DSAP)!=0x3D &&
(handle_frame->frame_receive.DSAP)!=0x3E)/*Si el DSAP recibido no es de diagnóstico, configuración
o parametrización*/
{
handle_frame->frame_transmit.SD=0x10;
handle_frame->frame_receive.DA &=0x7F;
handle_frame->frame_receive.SA &=0x7F;
handle_frame->frame_transmit.DA=handle_frame->frame_receive.SA;
handle_frame->frame_transmit.SA=handle_frame->frame_receive.DA;
handle_frame->frame_transmit.FC=0x03;/*Indica que el servicio no es manejado por el esclavo*/
handle_frame->transmit_length_ufc=0x04;
SD1=1;
transmit=1;/*Activa bandera de transmisión*/
}
}
}
}

```

```

    }

if (((handle_frame->frame_receive.DSAP)==0x3C) && ((handle_frame-
>frame_receive.SSAP)==0x3E))/*Si la trama recibida es de diagnóstico*/
{
    handle_frame->frame_transmit.SD=0x68;
    handle_frame->frame_transmit.LE=11;
    handle_frame->frame_transmit.LEr=11;
    handle_frame->frame_transmit.SD=handle_frame->frame_receive.SD;
    handle_frame->frame_transmit.DA=handle_frame->frame_receive.SA;
    handle_frame->frame_transmit.SA=handle_frame->frame_receive.DA;
    handle_frame->frame_transmit.FC=0x08;
    handle_frame->frame_transmit.DSAP=handle_frame->frame_receive.SSAP;
    handle_frame->frame_transmit.SSAP=handle_frame->frame_receive.DSAP;

if (((handle_frame->first_diagnostic)==0x00))/*Si la bandera de primer diagnóstico esta activa*/
    {
    handle_frame->first_diagnostic=0x01;/*Desactiva bandera de primer diagnóstico*/
    handle_frame->frame_transmit.DATAUNIT[0]=0x02;/*Byte 1 de primer diagnóstico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[1]=0x05;/*Byte 2 de primer diagnóstico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[2]=0x00;/*Byte 3 de primer diagnóstico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[3]=0xFF;/*Byte 4 de primer diagnóstico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[4]=0x97;/*Byte 5 de primer diagnóstico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[6]=0x00;/*Byte 6 de primer diagnóstico a ser transmitido*/
    }/*end if primer diagnóstico*/
else /*Si no es primer diagnóstico*/
    {
    handle_frame->frame_transmit.DATAUNIT[0]=0x80;/*Byte 1 de segundo diagnostico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[1]=0x0C;/*Byte 2 de segundo diagnostico a ser
transmitido*/
    handle_frame->frame_transmit.DATAUNIT[2]=0x00;/*Byte 3 de segundo diagnostico a ser transmitido*/
    handle_frame->frame_transmit.DATAUNIT[3]=0x01;/*Byte 4 de segundo diagnostico a ser transmitido*/
    }

    handle_frame->transmit_length_ufc=15;/*Numero de bytes a ser escritos a la UFC a ser transmitidos*/
    diagnostic=1;/*Activa bandera de transmisión de diagnóstico*/
    transmit=1;/*Activa bandera de transmisión*/
    }/*end si la trama recibida es de diagnóstico*/

if (((handle_frame->frame_receive.DSAP)==0x3D) && ((handle_frame-
>frame_receive.SSAP)==0x3E))/*Si la trama recibida es de parametrización*/
    {
    handle_frame->transmit_length_ufc=1;/*Numero de bytes a ser escritos a la UFC a ser trasmitidos*/
    transmit=1;/*Activa bandera de transmision*/
    parameterization=1;/*Activa bandera de transmisión de parametrización*/

    }/*end si la trama recibida es de parametrización*/

if (((handle_frame->frame_receive.DSAP)==0x3E) && ((handle_frame-
>frame_receive.SSAP)==0x3E))/*Si la trama recibida es de configuración*/
    {
    handle_frame->transmit_length_ufc=1;/*Numero de bytes a ser escritos a la UFC a ser trasmitidos*/

```



```

transmit=1;/*Activa bandera de transmisión*/
configuration=1;/*Activa bandera de transmisión de configuración*/
    }/*End si la trama recibida es de configuración*/
}
else
{
if (data_exchange==1)/*Si la bandera de modo data exchange esta activa*/
{
handle_frame->cyclic_data[0]=handle_frame->frame_receive.DATAUNIT[0];/*Almacenamiento de Byte
1 del dato de corriente enviado desde el PLC*/
handle_frame->cyclic_data[1]=handle_frame->frame_receive.DATAUNIT[1];/*Almacenamiento de Byte
2 del dato de corriente enviado desde el PLC*/
handle_frame->cyclic_data[2]=handle_frame->frame_receive.DATAUNIT[2];/*Almacenamiento de Byte
3 del dato de corriente enviado desde el PLC*/
handle_frame->cyclic_data[3]=handle_frame->frame_receive.DATAUNIT[3];/*Almacenamiento de Byte
4 del dato de corriente enviado desde el PLC*/

handle_frame->analog_current=*(float*)&(handle_frame->cyclic_data);/*Almacenamiento del dato de
corriente enviado por el PLC en variable de tipo flotante*/
handle_frame->analog_current_escal=(xmax/ymax)*((handle_frame->analog_current)-b);/*Escalizacion
del dato de corriente enviado desde el PLC*/

handle_frame->transmit_length_ufc=1;/*Numero de bytes a ser escritos a la UFC a ser trasmitidos*/
transmit=1;/*Activa bandera de transmision*/
    data_exchange_state=1;/*Activa bandera de transmisión de data exchange*/

current=(signed short)&(handle_frame->analog_current_escal);/*Almacenamiento de dato de corriente
escalizado en variable de tipo short*/
handle_frame->DAC_analog_current[0]=*(char*)current;/*Separación del dato de corriente escalizado
(Byte 1)*/
handle_frame->DAC_analog_current[1]=*(char*)(current+1);/*Separacion del dato de corriente
escalizado (Byte 2)*/

    PTFDD=0xFF;/*Habilitacion de los puertos de datos con la UFC como salida*/
if ((handle_frame->analog_current_escal)>0)/*Si el dato en corriente es mayor a 0*/
{
if ((handle_frame->analog_current_escal)>=255)/*Si el dato en corriente es mayor a 255*/
{
PTFD=0xFF;
}
else
{
PTFD=handle_frame->DAC_analog_current[1];
}
}
else
{
PTFD=0x00;
}
}
}
if (transmit==1)/*Si la bandera de transmisión esta activa*/

```

```

{
if (SD1==1)/*Si el Start Delimiter a ser transmitido es SD1 (0x10)*/
{

    PTCDD=0xFF;/*Habilitacion de los puertos de datos con la UFC como salida*/

handle_frame->write_data.address=0x01;/*Escritura al registro Mode*/
    handle_frame->write_data.data=0x04;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0B;/*Escritura al registro FIFO Control*/
    handle_frame->write_data.data=0x47;/*reset de FIFO de transmision*/
    write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x08;/*Escritura del número de bytes a ser escritos a la FIFO de
transmision (Low Byte)*/
handle_frame->write_data.data=handle_frame->transmit_length_ufc;
    write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x09;/*Escritura del número de bytes a ser escritos a la FIFO de
transmision (High Byte)*/
handle_frame->write_data.data=0x00;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0D;/*Escritura del Start Delimiter a ser transmitido*/
    handle_frame->write_data.data=handle_frame->frame_transmit.SD;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0D;/*Escritura del Destination Address a ser transmitido*/
    handle_frame->write_data.data=handle_frame->frame_transmit.DA;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0D;/*Escritura del Source Address a ser transmitido*/
    handle_frame->write_data.data=handle_frame->frame_transmit.SA;
    write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x02;/*Escritura al registro de Control para habilitar transmision*/
handle_frame->write_data.data=0x05;
    write_ufc(&(handle_frame->write_data));

PTCDD=0x00; /*Habilitacion de los puertos de datos con la UFC como entrada*/

handle_frame->read_data.address=0x03;/*Lectura del registro de status*/
    read_ufc(&(handle_frame->read_data));

if ((handle_frame->read_data.data)&0x02)/*Revisa si la bandera que indica si hay espacio en la FIFO de
transmision se encuentra activada*/
{
    PTCDD=0xFF;/*Habilitacion de los puertos de datos con la UFC como salida*/
handle_frame->write_data.address=0x0D;
    handle_frame->write_data.data=handle_frame->frame_transmit.FC;/*Escritura del Function Code a
ser transmitido*/
    write_ufc(&(handle_frame->write_data));
}
}

```

```

}
}

if (diagnostic==1) /*Si la bandera de transmisión de diagnóstico esta activa*/
{
    PTCDD=0xFF; /*Habilitacion de los puertos de datos con la UFC como salida*/
    handle_frame->write_data.address=0x01; /*Escritura al registro Mode*/
    handle_frame->write_data.data=0x04;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0B; /*Escritura al registro FIFO Control*/
    handle_frame->write_data.data=0x47; /*reset de FIFO de transmision*/
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x08; /*Escritura del número de bytes a ser escritos a la FIFO de
    transmision (Low Byte)*/
    handle_frame->write_data.data=handle_frame->transmit_length_ufc;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x09; /*Escritura del número de bytes a ser escritos a la FIFO de
    transmision (High Byte)*/
    handle_frame->write_data.data=0x00;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0D;
    handle_frame->write_data.data=handle_frame->frame_transmit.SD; /*Escritura del Start Delimiter a
    ser transmitido*/

    handle_frame->write_data.address=0x0D;
    handle_frame->write_data.data=handle_frame->frame_transmit.LE; /*Escritura del Net Data Lenght
    a ser transmitido*/
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0D;
    handle_frame->write_data.data=handle_frame->frame_transmit.LEr; /*Escritura del Net Data Lenght
    repetido a ser transmitido*/
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0D;
    handle_frame->write_data.data=handle_frame->frame_transmit.SD; /*Escritura del Start Delimiter a
    ser transmitido*/
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x02; /*Escritura al registro de Control para habilitar transmision*/
    handle_frame->write_data.data=0x05;
    write_ufc(&(handle_frame->write_data));

    PTCDD=0x00; /*Habilitacion de los puertos de datos con la UFC como entrada*/
    handle_frame->read_data.address=0x03; /*Lectura del registro de status*/
    read_ufc(&(handle_frame->read_data));

    if ((handle_frame->read_data.data)&0x02) /*Revisa si la bandera que indica si hay espacio en la FIFO de
    transmision se encuentra activada*/

```

```

    {
        PTCDD=0xFF; /*Habilitacion de los puertos de datos con la UFC como salida*/
        handle_frame->write_data.address=0x0D;
        handle_frame->write_data.data=handle_frame->frame_transmit.DA; /*Escritura del Destination
Address a ser transmitido*/
        write_ufc(&(handle_frame->write_data));

        handle_frame->write_data.address=0x0D;
        handle_frame->write_data.data=handle_frame->frame_transmit.SA; /*Escritura del Source Address
a ser transmitido*/
        write_ufc(&(handle_frame->write_data));

        handle_frame->write_data.address=0x0D; /*Escritura del Function Code a ser transmitido*/
        handle_frame->write_data.data=0x08;
        write_ufc(&(handle_frame->write_data));

        handle_frame->write_data.address=0x0D;
        handle_frame->write_data.data=handle_frame->frame_transmit.DSAP; /*Escritura del Destination
service Access Point a ser transmitido*/
        write_ufc(&(handle_frame->write_data));

        handle_frame->write_data.address=0x0D;
        handle_frame->write_data.data=handle_frame->frame_transmit.SSAP; /*Escritura del Source
service Access Point a ser transmitido*/
        write_ufc(&(handle_frame->write_data));

        for (i=0; i<6; i++)
        {
            handle_frame->write_data.address=0x0D;
            handle_frame->write_data.data=handle_frame->frame_transmit.DATAUNIT[i]; /*Escritura del
Data Unit a ser transmitido*/
            write_ufc(&(handle_frame->write_data));
        }
    }
} /*end if transmitir diagnostico*/

if ((configuration==1) || (parameterization==1) || (data_exchange_state==1)) /*Si las banderas de
transmision de parametrizacion, configuracion o data exchange estan activas*/

    PTCDD=0xFF; /*Habilitacion de los puertos de datos con la UFC como salida*/
    handle_frame->write_data.address=0x01; /*Escritura al registro Mode*/
    handle_frame->write_data.data=0x04;
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x0B; /*Escritura al registro FIFO Control*/
    handle_frame->write_data.data=0x47; /*reset de FIFO de transmision*/
    write_ufc(&(handle_frame->write_data));

    handle_frame->write_data.address=0x08; /*Escritura del numero de bytes a ser escritos a la FIFO de
transmision (Low Byte)*/
    handle_frame->write_data.data=0x01;
    write_ufc(&(handle_frame->write_data));

```

```

handle_frame->write_data.address=0x09; /*Escritura del número de bytes a ser escritos a la FIFO de
transmision (High Byte)*/
handle_frame->write_data.data=0x00;
write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x0D; /*Transmision de Short Acknowledgement*/
handle_frame->write_data.data=0xE5;
write_ufc(&(handle_frame->write_data));

handle_frame->write_data.address=0x02;
handle_frame->write_data.data=0x05; /*Escritura al registro de Control para habilitar transmisión*/
write_ufc(&(handle_frame->write_data));

PTCDD=0x00;
handle_frame->read_data.address=0x03;
read_ufc(&(handle_frame->read_data));
} /*end if transmitir configuración o parametrizacion*/
} /*end if hay bandera para transmitir*/
} /*end interrupt*/

/*****
*      RX_TX_CONTROL.H      *
*****/

#if !defined(RX_TX_CONTROL_H);

#define RX_TX_CONTROL_H
#include <stddef.h>
#include <math.h>
#include "read_write_ufc.h"
#include "structures.h"

void initialize_ufc(frame_control *handle_frame);
void interrupt_ufc(frame_control *handle_frame);
#endif

```

9.2 PALABRAS CLAVE PARA REVISION DE ARCHIVOS GSD

GSD_Revision: (M startingwithGSD_Revision 1)

ID del formato del archivo GSD.

Vendor_Name: (M)

Nombre del fabricante.

Model_Name: (M)

Asignación del fabricante del dispositivo

Revision: (M)

Versión de revisión del dispositivo.

Revision_Number: (O starting with GSD_Revision 1)

ID de la versión del dispositivo. Este valor de revisión debe ser acorde al valor de revisión de diagnóstico del esclavo.

Tipo: Unsigned8 (1 - 63)

Ident_Number: (M)

Tipo del dispositivo.

Tipo: Unsigned16

Protocol_Ident: (M)

Identificación del protocolo del dispositivo

0: PROFIBUS DP

16 - 255: Fabricante específico

Station_Type: (M)

Tipo de dispositivo DP.

0: DP Slave,

1: DP Master (Class 1)

FMS_supp: (D)

Este dispositivo es una mezcla entre FMS/DP.

Tipo: Booleano (1: True)

Hardware_Release: (M)

Versión final del Hardware del dispositivo.

Software_Release (M)

Versión final del Software del dispositivo.

9.6_supp: (G)

Velocidad que soporta el dispositivo 9.6 kbit/s.

Tipo: Booleano (1: True)

19.2_supp: (G)

Velocidad que soporta el dispositivo 19.2 kbit/s.

Tipo: Booleano (1: True)

31.25_supp: (G starting with GSD_Revision 2)

Velocidad que soporta el dispositivo 31.25 kbit/s.

Tipo: Booleano (1: True)

45.45_supp: (G starting with GSD_Revision 2)

Velocidad que soporta el dispositivo 45.45 kbit/s.

Tipo: Booleano (1: True)

93.75_supp: (G)

Velocidad que soporta el dispositivo 93.75 kbit/s.

Tipo: Booleano (1: True)

187.5_supp: (G)

Velocidad que soporta el dispositivo 187.5 kbit/s.

Tipo: Booleano (1: True)

500_supp: (G)

Velocidad que soporta el dispositivo 500 kbit/s.

Tipo: Boolean0 (1: True)

1.5M_supp: (G)

Velocidad que soporta el dispositivo 1.5 Mbit/s.

Tipo: Booleano (1: True)

3M_supp: (G starting with GSD_Revision 1)

Velocidad que soporta el dispositivo 3 Mbit/s.

Tipo: Booleano (1: True)

6M_supp: (G starting with GSD_Revision 1)

Velocidad que soporta el dispositivo 6 Mbit/s.

Tipo: Booleano (1: True)

12M_supp: (G starting with GSD_Revision 1)

Velocidad que soporta el dispositivo 12 Mbit/s.

Tipo: Booleano (1: True).

MaxTsd_r_9.6: (G)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 9.6kbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsd_r_19.2: (G)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 19.2kbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsd_r_31.25: (G starting with GSD_Revision 2)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 31.25kbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsd_r_45.45: (G starting with GSD_Revision 2)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 45.45kbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsd_r_93.75: (G)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 93.75kbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsd_r_187.5: (G)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 187.5kbit/s.

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsdrr_500: (G)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 500kbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsdrr_1.5M: (G)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 1.5 Mbit/s.

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsdrr_3M: (G starting with GSD_Revision 1)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 3 Mbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsdrr_6M: (G starting with GSD_Revision 1)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 6 Mbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

MaxTsdrr_12M: (G starting with GSD_Revision 1)

Tiempo de respuesta necesario para responder un requerimiento de mensaje a una velocidad de 12 Mbit/s

Tipo: Unsigned16

Base de tiempo: Bit Time

Redundancy: (D)

Este valor especifica si el dispositivo soporta ingeniería de transmisión redundante.

Tipo: Booleano

0: No, 1: La redundancia es soportada.

Repeater_Ctrl_Sig: (D)

El tipo de conector utilizado.

Tipo: Unsigned8

0: No conectada, 1: RS485, 2: TTL

24V_Pins: (D)

El sentido del conector del bus

Tipo: Unsigned8

0: No conectado, 1: Entrada, 2: Salida

Implementation_Type: (O starting with GSD_Revision 1)

Estándar de implementación usada en el esclavo

Bitmap_: (O starting with GSD_Revision 1)

Nombre del archivo del mapa de bits

Freeze_Mode_supp: (D)

Un dispositivo esclavo que soporta el modo de bloqueo garantiza que en el próximo ciclo de datos (después del comando “Freeze control”), el valor de los datos que fueron bloqueados por última vez se transfieren al bus.

Tipo: Booleano (1: True)

Sync_Mode_supp: (D)

El dispositivo DP soporta el modo de sincronismo.

Tipo: Booleano (1: True)

Auto_Baud_supp: (D)

El dispositivo DP reconoce automáticamente la velocidad de transmisión.

Tipo: Booleano (1: True)

Set_Slave_Add_supp: (D)

El dispositivo DP soporta la función “Set_Slave_Add”.

Tipo: Booleano (1: True)

Min_Slave_Intervall: (M)

This time specifies the minimum interval between two slave list cycles for the DP device.

Type: Unsigned16

Time base: 100 μ s

Modular_Station: (D)

En este campo se especifica si el dispositivo DP es una estación modular

Tipo: Booleano

0: Dispositivo compacto

1: Dispositivo modular

Max_Module: (M if Modular_Station)

El máximo número de módulos de la estación

Tipo: Unsigned8

Max_Input_Len: (M if Modular_Station)

El máximo largo del dato de entrada de la estación modular en bytes.

Tipo: Unsigned8

Max_Output_Len: (M if Modular_Station)

El máximo largo del dato de salida de la estación modular en bytes.

Tipo: Unsigned8

Max_Data_Len: (O only if Modular_Station)

Este parámetro corresponde a la suma de los 2 datos anteriores (Max_Input_Len y Max_Output_Len).

Tipo: Unsigned16

9.3 ARCHIVO GSD

Con base en las convenciones dadas en el capítulo 5.6 y apoyados en el software GSD Editor V.5.0 tomado de la página de PROFIBUS (www.profibus.com), se ejecuta y compila el programa para verificar que su instalación se lleve sin ningún inconveniente de sintaxis.

```

;*****
;
;**   GSD file ( Profibus PA to current converter )   **
;**                                                    **
;**   Manufacture : TG0981                            **
;**                                                    **
;**   Stand: 01.13.2000      Revised by: MG.          **
;**   smar0981.GSD                                           **
;*****
;
;
```

```

#Profibus_DP
GSD_Revision=3
Vendor_Name= "TG0981"
Model_Name= "TG_PA"
Revision = "3"
Ident_Number= 0x9700
Protocol_Ident= 0
Station_Type= 0
FMS_supp= 0
Hardware_Release="A01"
Software_Release="Z01"
31.25_supp= 1
45.45_supp= 1
93.75_supp= 1
MaxTsdr_31.25= 100
MaxTsdr_45.45= 250
MaxTsdr_93.75= 1000
Redundancy= 0
Repeater_Ctrl_Sig=0
24V_Pins= 0
Freeze_Mode_supp=0
Sync_Mode_supp=0
Auto_Baud_supp=0
Set_Slave_Add_supp=0
Min_Slave_Intervall= 200
Modular_Station= 1
Max_Module= 2
Max_Input_Len=45
Max_Output_Len=30
Max_Data_Len=75
Max_Diag_Data_Len=14
Slave_Family=0
User_Prm_Data_Len=0

```

```

;Modules for Analog Output

```

```

Module = "Analog Ouput " 0xA4
1
EndModule
SlotDefinition

```

```

Slot(1) = "Analog Output (AO)" 1 1

```

```

EndSlotDefinition

```

9.4 ANEXO CIRCUITO IMPRESO

El desarrollo de este montaje se concibió, con el fin de realizar una conexión segura entre la capa física y la capa de datos.

Para realizar el circuito impreso se tuvo en cuenta la separación de tierras analógicas y digitales en el sistema, por medio de planos. Por otro lado no fue necesario tener en cuenta impedancias reflejadas pues se está operando a una frecuencia baja.

Del mismo modo es importante manejar de forma separada las líneas digitales de las líneas analógicas sin cruzar ninguna de ellas para evitar que se presente acople electromagnético. Las líneas de datos que maneja la UFC están en el costado izquierdo del circuito mientras que las señales analógicas que son controladas por la MAU se manejan en el costado derecho del circuito impreso. Al hacer este ejercicio se evita que señales variantes como son los datos, interfieran directamente con los voltajes de alimentación del circuito.

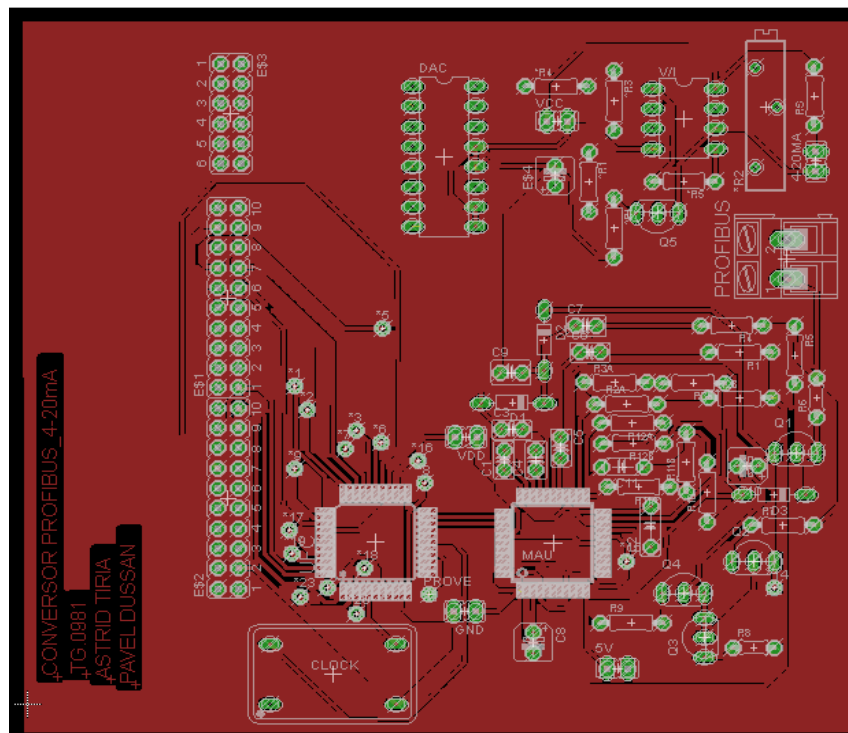


Imagen 1. Capa Top – Interfaz micro controlador- PROFIBUS

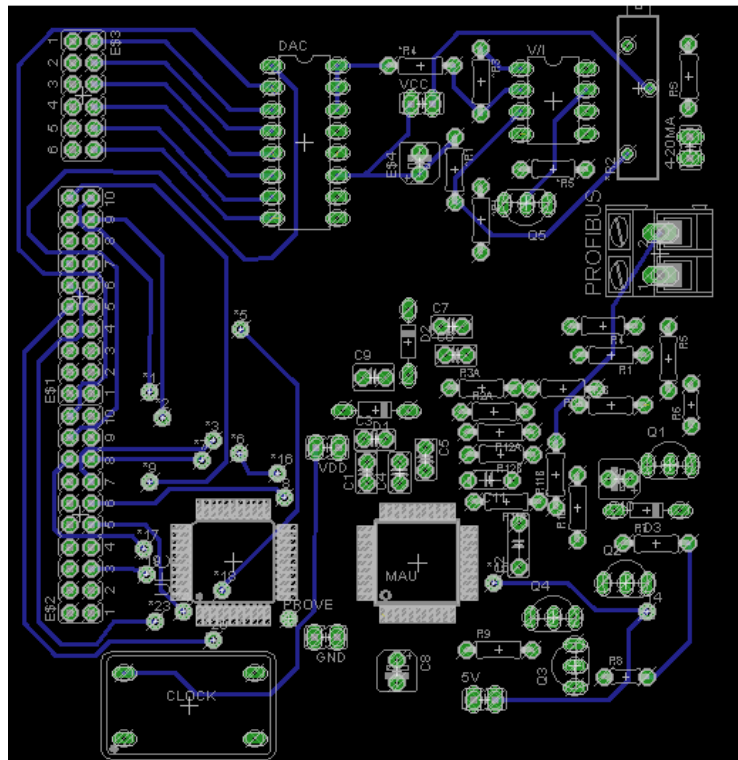


Imagen 2. Capa Top - Interfaz micro controlador- PROFIBUS

9.5 LABORATORIO DE COMUNICACIÓN INDUSTRIAL PROFIBUS PA

DEPARTAMENTO DE ELECTRÓNICA
FACULTAD DE INGENIERIA
PONTIFICIA UNIVERSIDAD JAVERIANA

OBJETIVOS

- Comprender el protocolo de comunicación PROFIBUS PA
- Realizar un montaje de una red PROFIBUS entendiendo el funcionamiento de cada parte de la red.
- Usar herramientas de análisis para verificar que la comunicación se efectuó de acuerdo al estándar PROFIBUS PA.
- Evaluar la aplicación y posibles ventajas de sistemas PROFIBUS en la industria

EQUIPO DISPONIBLE

- PLC Siemens
- Acoplador DP/PA y Fuente de alimentación
- Analizador de Protocolo Softing bus check BC-450
- Convertidor PROFIBUS PA a 4-20mA

TRABAJO PREVIO

PARA PODER DESARROLLAR LA PRÁCTICA SE DEBE RESOLVER LAS SIGUIENTES PREGUNTAS Y REALIZAR UN ESTUDIO PREVIO DE LOS MANUALES DE LOS EQUIPOS REQUERIDOS PARA EL DESARROLLO DE ESTA PRACTICA.

3.1. Investigue la estructura de la trama PROFIBUS PA y la función que cumple cada grupo de bytes.

3.2. Describa brevemente la secuencia de mensajes que se deben seguir para realizar la comunicación a nivel PROFIBUS PA.

3.3. De acuerdo a la arquitectura PROFIBUS complete el diagrama de interconexión de la Figura 1

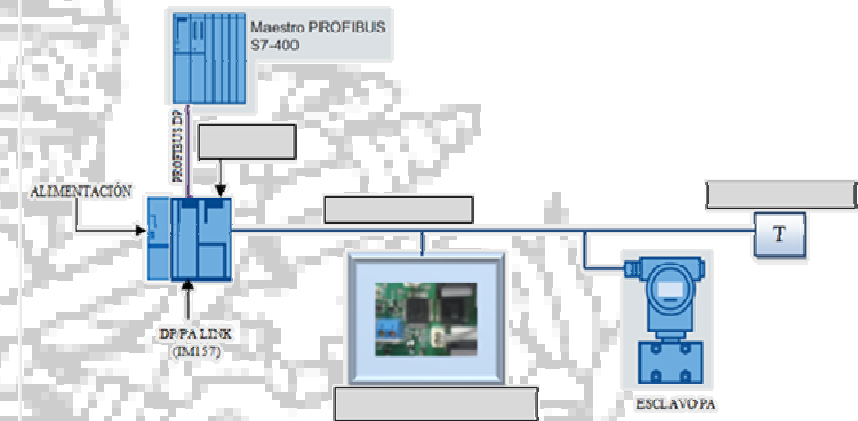


Figura1. Red PROFIBUS PA

PROCEDIMIENTO

4.1. Configure y realice la conexión de la arquitectura de la RED PROFIBUS que realizó en el trabajo previo.

4.2. Conecte al PLC el analizador de protocolo Softing de acuerdo a las indicaciones de su instructor. Este dispositivo le permitirá monitorear la actividad de las redes (DP/PA) y ver en tiempo real el estado de comunicación y transmisión de tramas.

4.3. Abra el software STEP 7 de Siemens y replique la red que ha instalado, teniendo en cuenta el número de parte de los dispositivos. En

la opción “hardware” podrá realizar la búsqueda e interconexión de los dispositivos que conforman la red.

NOTA: Tenga en cuenta que es indispensable contar con los archivos GSD de todos los dispositivos instalados. Estos archivos deben ser anexados tanto en el analizador de protocolo como en el software Simatic Manager Step 7. Si tiene alguna inquietud recurra a la opción de ayuda de cada programa y digite GSD file.

4.4. En la opción de software cree un bloque en el cual pueda insertar un valor flotante de 4-20mA el cual se almacene en una posición de memoria del PLC para ser enviada a través de la red PROFIBUS.

NOTA: Para conocer el bloque y tipo de dato requerido consulte el manual del software Step 7 (Programming with Sep 7)

4.5. Después de crear la red y el bloque compile en el PLC la información y proceda a observar las tramas en el software del analizador de protocolo dando clic en la opción RUN. Tome una impresión de pantalla de la secuencia que sigue la transmisión para completar el ciclo de comunicación.

4.6. Usando el DVM mida la corriente a la salida del convertidor la cual debe coincidir con el valor que ha ingresado en el bloque que creo en el software del S7. Para diferentes valores entre 4 y 20mA realice la medición correspondiente según la tabla 1.

| VALOR EN CORRIENTE S7 | VALOR EN CORRIENTE DVM |
|-----------------------|------------------------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Tabla 1. Mediciones de corriente

ANÁLISIS DE RESULTADOS.

- 5.1. Compare el valor que ha insertado en el bloque del S7 y la medición de la corriente a la salida del convertidor. Dibuje la gráfica de linealidad ¿Qué puede concluir sobre el grado de exactitud del dato transmitido? ¿Considera que este protocolo es fiable tanto en comunicación como en velocidad?
- 5.2 ¿Qué ventajas cree que sistemas de comunicaciones como estos pueden aportar a la industria?
- 5.3 Que mejoras incorporaría al sistema para hacerlo más eficiente.

INFORME

Realice un informe en formato de artículo IEEE, en el que incluya una breve explicación de los experimentos realizados, la tabla de parámetros obtenidos y el análisis de los resultados obtenidos, así como el trabajo previo y la solución a las preguntas realizadas en la práctica y problemas obtenidos.

