



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**LOS PROTOCOLOS DE COMUNICACIÓN EN EL
ENTORNO INDUSTRIAL, SUS FUNDAMENTOS Y SU
IMPORTANCIA EN EL SISTEMA DE AUTOMATIZACIÓN
DE UNA PLANTA DE GENERACIÓN DE ENERGÍA
GEOTÉRMICA**

Elio Alberto Orozco Vásquez
Asesorado por el Ing. Luis Felipe Bances Recinos

Guatemala, noviembre de 2006

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**LOS PROTOCOLOS DE COMUNICACIÓN EN EL
ENTORNO INDUSTRIAL, SUS FUNDAMENTOS Y SU
IMPORTANCIA EN EL SISTEMA DE AUTOMATIZACIÓN
DE UNA PLANTA DE GENERACIÓN DE ENERGÍA
GEOTÉRMICA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

ELIO ALBERTO OROZCO VASQUEZ

ASESORADO POR EL ING. LUIS FELIPE BANCES RECINOS

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO ELECTRICISTA

GUATEMALA, NOVIEMBRE DE 2006

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

LOS PROTOCOLOS DE COMUNICACIÓN EN EL ENTORNO INDUSTRIAL, SUS FUNDAMENTOS Y SU IMPORTANCIA EN EL SISTEMA AUTOMATIZACIÓN DE UNA PLANTA DE GENERACIÓN DE ENERGÍA GEOTÉRMICA,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, el 9 de agosto de 2006.

Elio Alberto Orozco Vásquez.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Julio Ismael Gonzalez Podszueck
EXAMINADOR	Inga. Marlene Lone de Hidalgo
EXAMINADOR	Ing. Edwin Solares Martínez
EXAMINADOR	Ing. Angel García Martínez
SECRETARIO	Ing. Francisco González López

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Guatemala, 29 de Septiembre de 2,006.

Ingeniero Julio Cesar Solares Peñate
Coordinador Area Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Ingeniero Solares:

Me permito dar aprobación al trabajo de tesis titulado: **“Los protocolos de comunicación en el entorno industrial, sus fundamentos y su importancia en el sistema de automatización de una planta de generación de energía geotérmica”**, desarrollado por el estudiante Elio Alberto Orozco Vásquez, ya que considero cumple con los requisitos establecidos para tal fin.

Atentamente,

A handwritten signature in blue ink, appearing to read 'Luis Felipe Bances Recinos'.

Ing. Luis Felipe Bances Recinos

Colegiado No. 5330.

ASESOR

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



Guatemala, 23 de octubre 2006.

FACULTAD DE INGENIERIA

Señor Director
Ing. Mario Renato Escobedo Martinez
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: **Los protocolos de comunicación en el entorno industrial, sus fundamentos y su importancia en el sistema de automatización de una planta de generación de energía geotérmica**, desarrollado por el estudiante, Elio Alberto Orozco Vásquez, por considerar que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,

ID Y ENSEÑAD A TODOS


Ing. Julio Cesar Solares Peñate
Coordinador Area de Electrónica

JCSP/sro



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Area, al trabajo de Graduación del estudiante, Elio Alberto Orozco Vásquez titulado: **Los protocolos de comunicación en el entorno industrial, sus fundamentos y su importancia en el sistema de automatización de una planta de generación de energía geotérmica,** procede a la autorización del mismo.

Ing. Mario Renato Escobedo Martínez

DIRECTOR



GUATEMALA, 26 DE OCTUBRE 2,006.

ACTO QUE DEDICO A:

DIOS	Por ser fuente, motivo y destino de mi vida.
A LA MEMORIA DE MI MADRE	Maria Andrea Vásquez
MI PADRE	Alberto Orozco López
MI ESPOSA	Anyelina Sierra
MIS HIJOS	Diego Andrés y Esteban
MIS HERMANOS	Héctor, Elida, Rudy, Eluvia y Silvia
MI SUEGRA	Hania David de Sierra
MIS CUÑADOS Y CUÑADAS	Evelia Avalos, Edelma Orellana, Milton Gómez, Marco Pablo Sierra, Hania Sierra
MIS SOBRINO(A)S EN GENERAL	

AGRADECIMIENTOS A:

Ormat de Proyectos

Ingeniero Miguel Galicia

Ingeniero Aaron Choresh

Ingeniero Shalom Zivelin

Ingeniero Luis Felipe Bances y Familia.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE ABREVIATURAS	XI
GLOSARIO	XIII
RESUMEN	XVII
OBJETIVOS	XIX
INTRODUCCIÓN	XXI
1. PROCESO DE TRANSMISIÓN DE DATOS	1
1.1 Técnicas de transmisión	2
1.1.1 Transmisión serie	2
1.1.2 Transmisión paralelo	2
1.2 Tipos de transmisión	3
1.2.1 Transmisión asíncrona	3
1.2.2 Transmisión síncrona	3
1.3 Modos de transmisión	4
1.3.1 Transmisión simple (SIMPLEX)	4
1.3.2 Transmisión semidoble (HALF-DUPLEX)	5
1.3.3 Transmisión doble (FULL-DUPLEX)	5
1.4 Detección de errores	6
1.4.1 Comprobación de paridad	7
1.4.2 Comprobación de redundancia cíclica (CRC)	7
1.4.3 Comprobación de suma	7
1.5 Definición de Protocolo	8
1.5.1 Elementos que definen un Protocolo	8
1.5.2 Funciones de un Protocolo	8

2. MODELOS DE REFERENCIA PARA LOS PROTOCOLOS DE COMUNICACIÓN	11
2.1 El modelo de referencia OSI	12
2.1.1 Características del modelo OSI	13
2.1.2 Capas o niveles del modelo OSI	14
2.1.2.1 Capa física	14
2.1.2.2 Capa de enlace	15
2.1.2.3 Capa de red	16
2.1.2.4 Capa de transporte	16
2.1.2.5 Capa sesión	17
2.1.2.6 Capa de presentación	17
2.1.2.7 Capa de aplicación	18
2.2 El modelo de estratificación por niveles de TCP/IP	19
2.2.1 Nivel de aplicación	20
2.2.2 Nivel de transporte	21
2.2.3 Nivel de Internet	21
2.2.4 Nivel de interfaz de red	21
3. PROTOCOLOS COMÚNMENTE EMPLEADOS EN EL ENTORNO INDUSTRIAL.	23
3.1 Protocolo HART	23
3.1.1 Comandos del protocolo HART	24
3.1.1.1 Comandos universales	24
3.1.1.2 Comandos de práctica común	24
3.1.1.3 Comandos específicos	25
3.1.2 La señal física	25
3.1.2.1 Niveles de la señal	25
3.1.3 Procedimiento de transacción	25
3.1.4 Codificación de caracteres	26

3.1.5	Formato del mensaje	26
3.1.5.1	Preámbulo	27
3.1.5.2	El inicio	27
3.1.5.3	La dirección	27
3.1.5.4	El comando	27
3.1.5.5	Cuenta de bytes	27
3.1.5.6	El estado	28
3.1.5.7	Los Datos	28
3.1.5.8	Suma de verificación	28
3.2	Protocolo Modbus	28
3.2.1	La petición	29
3.2.2	La respuesta	30
3.2.3	Trama del mensaje Modbus	30
3.2.4	Modbus ASCII	31
3.2.4.1	Trama ASCII	31
3.2.5	Modbus RTU	32
3.2.5.1	Trama RTU	33
3.2.5.1.1	La dirección	33
3.2.5.1.2	La función	34
3.2.5.1.3	El campo información	35
3.2.5.1.4	El campo comprobación de error	35
3.2.6	Transmisión de los caracteres en serie	36
3.2.7	Métodos de comprobación de error	37
3.2.8	Control de paridad	37
3.2.9	Las direcciones en Modbus	37
3.2.10	Modbus TCP/IP	38
3.2.10.1	Fundamentos del protocolo Modbus TCP	39
3.2.10.2	Prestaciones de un sistema Modbus TCP/IP	39
3.3	Protocolo Fieldbus Foundation	39

3.3.1 Nivel físico	40
3.3.2 Pila de comunicaciones	41
3.3.3 Nivel usuario	41
3.4 Protocolo Profibus	43
3.4.1 Profibus DP	43
3.4.2 Profibus PA	44
3.4.3 Profibus FMS	44
3.5 Protocolo CCM	44
3.5.1 Caracteres de control CCM	45
3.5.2 Secuencia de interrogación	46
3.5.3 Encabezado de transferencia	46
3.6 Protocolo DNP3	47
3.6.1 Operación en reposo	48
3.6.2 Operación de reportes no solicitados	48
3.6.3 Operación de reportes por votación	48
3.6.4 Operación de reportes estático	48
3.6.5 Capa de aplicación	50
3.6.6 Capa de pseudo-transporte	50
3.6.7 Capa de enlace de datos	50
3.6.8 Capa física	51
3.7 Protocolo CAN	51
3.7.1 Capa física	51
3.7.2 Capa de enlace	52
3.7.3 Capa de aplicación	52
3.8 Protocolo DeviceNet	53
3.9 Protocolo SNP	53
3.9.1 Arquitectura	53
3.9.2 Capa de configuración	54
3.9.3 Capa de sesión	54

3.9.4	Capa de aplicación	54
3.9.5	Procedimiento de transacción	55

4.	EJEMPLO PRÁCTICO: SUMINISTRO DE INFORMACIÓN DE UNA PLANTA GEOTÉRMICA A UN SISTEMA SCADA	57
4.1	Planta geotérmica Orzunil I	57
4.1.1	Automatismo de la planta Orzunil I	58
4.2	Descripción de los equipos y software a utilizar en el proyecto	60
4.2.1	El controlador lógico programable (PLC)	60
4.2.2	El módulo de comunicaciones del PLC	61
4.2.3	Software empleado	62
4.2.4	Unidad de transmisión remota (RTU)	63
4.2.5	Módulo de comunicaciones de la RTU (IoE)	63
4.2.6	Software empleado	64
4.2.7	El cable de comunicaciones	65
4.3	Ejecución del proyecto	66
4.3.1	Información requerida por el Administrador del Mercado Mayorista AMM	66
4.3.2	Configuración del PLC	67
4.3.3	Localización de las señales requeridas	70
4.3.4	Programación del PLC	71
4.3.4.1	El bloque “BIT TEST”	72
4.3.4.2	El bloque “MOVE INT”	73
4.3.5	Configuración de la unidad RTU	79
4.3.6	Monitoreo de las señales	82

CONCLUSIONES	85
RECOMENDACIONES	87
BIBLIOGRAFÍA	89
APÉNDICE	91

ÍNDICE DE ILUSTRACIONES

FIGURAS

1	Elementos de un sistema de comunicación	1
2	Transmisión “Simplex”	4
3	Transmisión “Half-Duplex”	5
4	Transmisión “Full-Duplex”	6
5	Capas o niveles del modelo OSI	14
6	Confección de los paquetes en TCP/IP	19
7	Comparación de los modelos OSI y TCP/IP	20
8	La señal HART	24
9	Cadena de bits HART	26
10	Formato del mensaje HART	26
11	Ciclo de interrogación-respuesta en Modbus	29
12	Trama del mensaje ASCII	32
13	Trama del mensaje RTU	33
14	Secuencia de bits con trama ASCII	36
15	Secuencia de bits con trama RTU	36
16	Estructura del protocolo Fieldbus Foundation	40
17	Ejemplo de lazo de control para Fieldbus	42

18	Estructura del protocolo Profibus	43
19	Encabezado del protocolo CCM	46
20	Proceso de comunicación Maestro-Esclavo en protocolo DNP3.0	49
21	Diagrama de comunicación planta geotérmica Orzunil I	59
22	Diagrama unifilar planta geotérmica Orzunil I	60
23	Ventana principal del Logicmaster 90	62
24	Cable de comunicaciones PLC-RTU	65
25	Ventana de configuración Logicmaster 90	68
26	Configuración del módulo de comunicaciones CMM	69
27	Configuración de puertos del modulo CMM	69
28	Bloque "BIT TEST"	72
29	Bloque "MOVE INT"	73
30	Configuración de la unidad RTU	79
31	Configuración de puertos	80
32	Configuración en Modbus	81
33	Configuración de señales individuales	82
34	Monitoreo de las señales en la RTU	83
35	Protocolos de comunicación en el automatismo de la planta Orzunil I.	92

TABLAS

I	Formato en modo Modbus ASCII	31
II	Formato en modo Modbus RTU	32
III	Tabla de funciones Modbus RTU	34
IV	Las direcciones en Modbus	38
V	Funciones de control y de I/O Fieldbus Foundation	42
VI	Caracteres de control ASCII	45
VII	Especificación de los campos del protocolo CMM	47
VIII	Velocidad versus distancia protocolo CAN	52
IX	Señales discretas requeridas	66
X	Señales análogas requeridas	67
XI	Ubicación de señales discretas en PLC y RTU	70
XII	Ubicación de señales análogas en PLC y RTU	71
XIII	Descripción de parámetros bloque BIT TEST	72

XIV	Descripción de parámetros bloque MOVE INT	73
XV	Señales en la unidad RTU	84

LISTA DE ABREVIATURAS

ANSI	American National Standard Institute (Instituto Nacional Estadounidense de Normas)
ASCII	American Standar Code for Information Interchange (Código Estándar Americano para el Intercambio de Información)
DOS	Disk Operating System (Sistema Operativo de Disco)
HMI	Human Machine Interfase (Interfase Hombre Maquina)
IEA	Industries Electronics Association (Asociación de industrias electronicas)
IEEE	Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)
ISA	Instrument Society of America (Sociedad de Instrumentación de América)
ISO	International Organization for Standarization (Organización Internacional para la Estandarización)
kbps	Kilo bits por segundo
Kw.	Kilovatio

Kw/hr	Kilovatio-hora
KB	Kilobyte (1024 bytes)
mA.	Miliamperio
MB	Megabyte (1,048,576 bytes)
MW	Megavatio (1,000 Kw.)
OEC	Ormat Energy Converter (Convertidor de Energía Ormat)
OSI	Open System Interconnection (Sistema de Interconexión Abierta)
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos)

GLOSARIO

Automatización	Aplicación de procesos automáticos a un aparato, proceso o sistema.
Análoga	Expresión numérica de una variable física para representar una cantidad
Baud	Unidad para representar la velocidad de transmisión de datos, igual al número de elementos de código por segundo.
Binario	Sistema de numeración que usa solamente los números 0 y 1, También llamado sistema de base 2.
Bit	Unidad de memoria mas pequeña. Guarda solamente uno de los dos estados 1 o 0. (on/off, encendido/apagado etc.)
Byte	Grupo de dígitos binarios operados como una sola unidad, regularmente consta de 8 bits.
Controlador Lógico Programable (PLC)	Dispositivo industrial de estado sólido, el cual maneja señales de entrada/salida basado en una programación establecida.
Datagrama	Mensaje de comunicación serial que combina datos desde diferentes tipos de memoria y/o localización.

Dirección	Serie de números decimales, binarios etc., para identificar un dispositivo en especial
Estándar	Es un patrón de comparación. Sinónimo de norma.
Hexadecimal	Sistema de numeración, teniendo como base 16, representado por los dígitos 0 al 9 y por las letras A a la F.
Geotermia	Palabra proveniente de los vocablos griegos Geo (tierra) y Termos (calor); en conjunto significa calor proveniente de la tierra.
Maestro-Esclavo	Forma de comunicación entre dispositivos, en la cual solo una estación puede iniciar requisiciones y la otra responder.
Multidrop	Una configuración de cableado en forma serial la cual conecta mas de dos dispositivos en cadena.
Kilovatio	Medida de energía que significa 1,000 Joules de energía por segundo.
Mnemónico	Una abreviación dada a una instrucción, usualmente un acrónimo formado por las iniciales o parte de palabras.
Palabra	Unidad de memoria usualmente de 16 bits de longitud o dos bytes.

Par a par	Forma de comunicación entre estaciones, en la cual cualquier estación puede iniciar una requisición o responder.
Paridad	Estado anticipado, ya sea par, impar o ninguno; de un grupo de dígitos binarios.
RS-232	Estándar para las características eléctricas y mecánicas de la interfase de conexión entre un (ECD) Equipo de Comunicación de Datos y un Equipo Terminal de Datos (ETD)
Rack	Espacio físico para el acomodamiento de una fuente de voltaje y de módulos de I/O en los PLC's
Reloj	Dispositivo electrónico que marca el ritmo con el que se ejecutan las instrucciones de los programas
Slot	Espacio físico en la placa base del rack de un PLC donde se insertan las tarjetas controladoras de periféricos y de expansión de un equipo.

RESUMEN

El presente trabajo surge debido a la necesidad que el Ingeniero Electricista, conozca los fundamentos y la aplicación de los protocolos de comunicación en el entorno industrial, ya que, actualmente la tendencia en la industria es ir incrementando el grado de automatización de sus procesos.

El trabajo consta de cuatro capítulos. El capítulo uno comienza definiendo los términos fundamentales relacionados con el proceso de comunicación de datos, sus características y las formas de transmisión desde uno hasta otro punto. Luego muestra también los conceptos y características fundamentales referente a los protocolos de comunicación de datos.

El capítulo dos, muestra a dos de los modelos o estándares de referencia para los protocolos de comunicación en general y sus características y.

El capítulo tres, presenta un listado de algunos de los protocolos en el entorno industrial más usados actualmente. Protocolos de baja funcionalidad pero de alta velocidad como el protocolo CAN, hasta protocolos para áreas de seguridad intrínseca como Profibus-PA o HART.

Finalmente, el capítulo cuatro presenta un ejemplo sobre la importancia de los protocolos de comunicación, en la planta geotérmica Orzunil I, localizada en el municipio de Zunil, en el departamento de Quetzaltenango. El ejemplo se refiere al envío de información de la planta a un sistema SCADA. Se abarca desde la descripción de los equipos que intervienen en este proyecto, hasta la configuración y programación en el PLC y en la unidad RTU para el envío de la información en forma exitosa.

OBJETIVOS

General

Contribuir al enriquecimiento de los conocimientos del estudiante y del profesional de la ingeniería eléctrica, referente a los Protocolos de Comunicación en el entorno industrial, a través de la elaboración de material de apoyo relacionado al tema y la realización de un ejemplo práctico.

Específicos

1. Conocer los fundamentos de la transmisión de datos, y el papel de los protocolos de comunicación en ello.
2. Conocer el modelo OSI (sistema de interconexión abierto) y su relación con los Protocolos de Comunicación.
3. Conocer los protocolos de comunicación más comunes usados en el entorno industrial.
4. Mostrar el uso e importancia de los protocolos de comunicación, mediante un ejemplo práctico.

INTRODUCCIÓN

Los protocolos de comunicación son la plataforma sobre la cual dos o mas dispositivos se comunican en una red, sea esta de cualquier complejidad. Los instrumentos de medición, protección y control en una planta industrial se rigen bajo el mismo concepto: una serie de reglas y normas sobre las cuales se comunicaran, responderán y actuaran.

Sin embargo, en el área de las comunicaciones en entornos industriales, la estandarización de protocolos es un tema en permanente discusión, donde intervienen problemas no solamente técnicos sino tambien comerciales. Cada protocolo está optimizado para diferentes niveles de automatización y en consecuencia responden al interés de diferentes proveedores.

Es aquí donde se hace necesario para el profesional de la ingeniería eléctrica, familiarizarse en el campo de los protocolos de comunicación, no importando en que área se desempeña, ya que actualmente la tendencia en la industria de cualquier índole, es la implementación total de los llamados “dispositivos inteligentes” los cuales tienen la capacidad de comunicarse con otros elementos para formar una red industrial. Todo esto con la única finalidad de hacer mas eficientes los procesos de producción.

1. PROCESO DE TRANSMISIÓN DE DATOS

La transmisión de datos se refiere al envío de información en forma confiable desde un punto de origen hasta su objetivo, ya sea a pocos metros o a muchos kilómetros de distancia; esa confiabilidad debe ser independientemente de la aplicación, además puede ser que un estado de encendido-apagado sea suficiente o que se requiera de un rango de valores con una resolución de 1 parte en 4096 (12 bits).

Sea como sea la aplicación, se requieren tres elementos para todo proceso de comunicación de datos, estos son:

Figura 1. Elementos de un sistema de comunicación



Por "Proceso de transmisión de datos" se entiende las acciones necesarias para enviar y recibir datos. Este proceso requiere cuatro elementos básicos que son:

- Fuente o Emisor: Dispositivo que transmite los datos
- Mensaje: Son los datos a ser transmitidos
- Medio: Canal físico en donde se transmitirán los datos
- Receptor o Destino: Dispositivo destino de los datos

En el proceso de transmisión de datos, se añade información que utiliza el sistema para comprobar que la transmisión ha sido correcta. El proceso de transmisión de datos, se puede dividir básicamente en:

1. Establecer el adecuado uso de los datos para transmitirlo
2. Convertir los datos para su transmisión
3. Establecer el control en el envío de los datos
4. Comprobar la llegada de los datos a su destino
5. Detectar presencia de errores en el proceso de envío/recepción.

1.1 Técnicas de transmisión

1.1.1 Transmisión serie

Los bits se transmiten de uno en uno sobre una línea única. Por ejemplo para enviar un byte de datos (8 bits) cada bit es enviado detrás de otro a través de la línea agregando además un par más de bits para designar el inicio y final del dato. Así, a 1200 baudios, cerca de 120 bytes de datos son enviados por segundo. Esta técnica es muy útil cuando las transmisiones son a larga distancia.

1.1.2 Transmisión paralelo

Esta técnica de transmisión requiere de una línea por cada bit de valor digital transmitido, los bits se transmiten en grupo sobre varias líneas al mismo tiempo. La transmisión en paralelo es más rápida pero se encarece por necesitar de muchos conductores y también porque aumenta la complejidad de los transmisores y los receptores de la línea.

1.2 Tipos de transmisión

1.2.1 Transmisión asíncrona

La comunicación asíncrona, es un tipo de transmisión de la capa física y es más usada para proveer conectividad entre computadoras personales e impresoras, módems, faxes, PLC's etc. El aspecto más importante de la comunicación asíncrona es que los relojes del transmisor y del receptor son independientes y no sincronizados.

La forma como se consigue sincronismo es enviando caracteres de tamaño y formato determinado, sincronizándose al inicio de cada carácter o cadena de bits. Cada carácter va precedido de un bit de arranque y seguido de uno o dos bits de parada, además se agrega un bit de paridad para proveer protección contra errores en la transmisión. Este tipo de transmisión es sencillo y no tan costoso, aunque requiere muchos bits de comprobación y de control. También hay que mencionar que el hecho de usar relojes independientes en el transmisor y el receptor limita la transmisión a caracteres relativamente cortos (<8 bits) y modera la velocidad a menos de 64 kbps.

1.2.2 Transmisión síncrona

En este tipo, no hay bits de arranque ni de paro, por lo que se puede transmitir bloques que incluyan gran número de bits; y para evitar errores de delimitación, se pueden sincronizar emisor y receptor mediante canales separados de reloj, los cuales manejan la transmisión y recepción de la información.

Además de los datos propios y de la sincronización, se necesitan señales preliminares al comienzo del bloque de datos, dependiendo de que clase de protocolo se trate. Se usan bytes de sincronización si son protocolos orientados a byte y se usan banderas si se trata de protocolos orientados a bit.

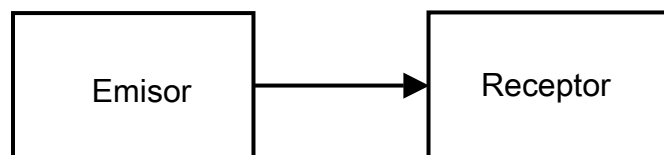
La transmisión sincrónica tiene la ventaja de que el tiempo de información es exactamente alineada con el dato recibido, permitiendo una operación con porcentaje de datos mas altos. Para bloques grandes de datos, la transmisión sincrónica es más eficiente que la asíncrona.

1.3 Modos de transmisión

1.3.1 Transmisión simple (SIMPLEX)

Se presenta cuando la línea sólo permite la transmisión en una sola dirección. Algunos ejemplos de transmisión Simple son: La radio, la televisión y los sistemas télex que sólo envían información al usuario sin efectuar ninguna transacción de respuesta al sistema.

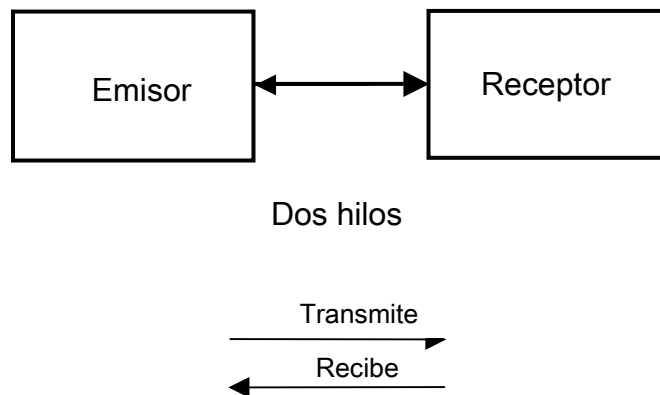
Figura 2. Transmisión "Simplex"



1.3.2 Transmisión semidoble (HALF-DUPLEX)

Este modo de transmisión se da en una sola línea, por lo que no se puede enviar y recibir información al mismo tiempo sino que se debe esperar que se termine de recibir la información para poder enviar datos. Para citar un ejemplo de este tipo de transmisión esta: los equipos de "walkie-talkie".

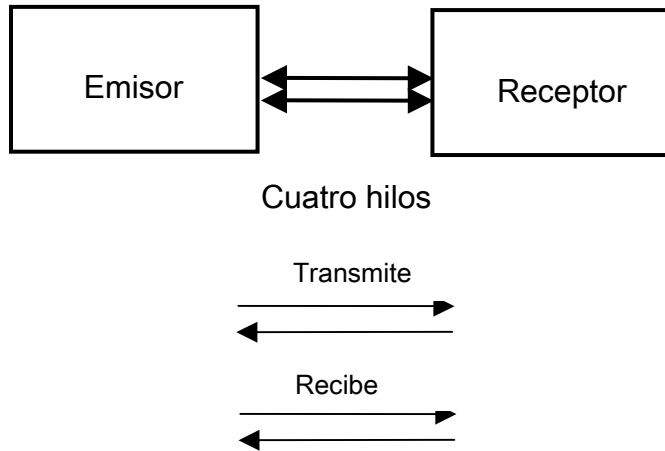
Figura 3. Transmisión "Half-Duplex"



1.3.3 Transmisión doble (FULL-DUPLEX)

En este modo de transmisión, se tienen dos líneas, una para enviar y otra para recibir datos al mismo tiempo. Un ejemplo de ello es: El teléfono.

Figura 4. Transmisión “Full-Duplex “



1.4 Detección de errores

Es necesario asegurar hasta donde sea posible que la información se transmitirá en forma segura y confiable, pero cuanto mayor es la trama que se transmite, mayor es la probabilidad de que se produzca algún error; ya que, en la práctica, un canal de comunicación está sujeto a una diversidad de perturbaciones que dan como resultado una distorsión del mensaje que se está transmitiendo.

Para detectar errores, se añade un código en función de los bits de la trama de forma que este código señale si se ha cambiado algún bit en el camino. Este código debe de ser conocido e interpretado tanto por el emisor como por el receptor.

Se pueden usar las siguientes técnicas de detección de errores:

1.4.1 Comprobación de paridad

Se añade un bit de paridad al bloque de datos por ejemplo, si hay un número par de bits 1, se añade un BIT 0 de paridad y si son impares, se le añade un BIT 1 de paridad. Algunas veces, incluso ocurre que el propio bit de paridad es alterado por el ruido, con lo que el sistema de detección fallará.

1.4.2 Comprobación de Redundancia Cíclica (CRC)

Debido a su sencillez en la implementación, esta técnica es muy difundida. Aquí, cualquier mensaje en formato binario se divide dentro de una cantidad constante, su residuo se envía de vuelta y se compara en la estación receptora con el residuo obtenido al dividir los datos recibidos entre la misma constante. Se acepta el mensaje, cuando los residuos son iguales, o se considera que hubo un error en caso contrario.

1.4.3 Comprobación de suma

A la información a transmitir se le considera como un conjunto de palabras de n bits, y la redundancia es otra palabra de n bits, que inicialmente están a "0".

Emisor: La redundancia se calcula como el complemento a 1 de la suma en complemento a 1 de las palabras de información.

Receptor: Se suman en complemento a 1 tanto las palabras de información como la de redundancia, y si el resultado es una palabra con todos los bits a "1" se da por buena.

1.5 Definición de Protocolo

Un protocolo de comunicación es el conjunto normas y reglas determinadas a cumplir por dos o mas dispositivos que desean comunicarse entre si, en otras palabras “es el idioma que los dispositivos hablan entre si”, por lo tanto estos deben aprender la gramática, la sintaxis y todas las reglas para poder entablar la comunicación en forma satisfactoria.

1.5.1 Elementos que definen un protocolo

- **Sintaxis:** Se refiere al formato, codificación y niveles de señal de datos.
- **Semántica:** Es la información de control y gestión de errores.
- **Temporización:** Se refiere a la coordinación entre la velocidad y orden secuencial de las señales.

1.5.2 Funciones de un Protocolo

Segmentación: Para que se de la comunicación de datos en forma eficiente, es recomendable dividir los bloques de datos en unidades pequeñas e iguales en tamaño, esto se refiere al proceso de segmentación. El Bloque básico de segmento en una cierta capa de un protocolo se le llama PDU (Acrónimo en ingles de: Unidad de Datos de Protocolo).

Encapsulado: Se refiere al proceso de adherir información de control al segmento de datos. Esta información de control es el direccionamiento del emisor/receptor, código de detección de errores y control de protocolo.

Control de conexión: Hay bloques de datos, sólo de control y otros de datos y control. Cuando se utilizan data gramas, todos los bloques incluyen control y datos ya que cada PDU se trata como independiente, hay protocolos más sencillos que otros, por lo que los protocolos de los emisores y receptores deben de ser compatibles al menos.

Entrega ordenada: El envío de los PDU puede acarrear el problema de que si hay varios caminos posibles lleguen al receptor desordenados o repetidos, por lo que el receptor debe tener un mecanismo para reordenarlos. Hay sistemas que tienen un mecanismo de numeración con algún número; esto hace que el módulo sean lo suficientemente alto como para que sea imposible que haya dos segmentos en la red al mismo tiempo y con el mismo número.

Control de flujo de datos: hay controles de flujo de parada y espera o de ventana deslizante. El control de flujo es necesario en varios protocolos o capas ya que el problema de saturación del receptor se puede producir en cualquier capa del protocolo.

Control de errores: generalmente se utiliza un temporizador para retransmitir una trama una vez que no se ha recibido confirmación después de expirar el tiempo del temporizador. Cada capa de protocolo debe de tener su propio control de errores.

Direccionamiento: Cada dispositivo intermedio de almacenamiento debe tener una dirección única para evitar conflictos. A su vez, en cada terminal o sistema final puede haber varios agentes o programas que utilizan la red, por lo que cada uno de ellos tiene asociado un puerto.

Multiplexación: Esto se refiere a la capacidad de multiplexar las conexiones de una capa hacia otra, es decir que de una única conexión de una capa superior, se pueden establecer varias conexiones en una capa inferior, o también se puede en forma inversa.

Servicios de transmisión: Entre los servicios que puede prestar un protocolo esta por ejemplo el servicio de "Prioridad", este se refiere como su nombre lo indica a la prioridad de algunos mensajes deben tener sobre otros, en este sentido se puede mencionar por ejemplo que los mensajes de control tienen prioridad sobre los mensajes de monitoreo.

2. MODELOS DE REFERENCIA PARA LOS PROTOCOLOS DE COMUNICACIÓN

El auge de la informática y la automatización industrial, ha dado como resultado la proliferación de fabricantes y de equipos inteligentes (dispositivos de control, medición, protección etc.), complicando la tarea de la comunicación de datos entre diferentes equipos ya que cada uno se empezó a diseñar con sus propias características tales como:

- Tipo de señal
- Memoria.
- Interfaces para Comunicaciones.
- Códigos de caracteres.

Así que, para que diferentes dispositivos puedan comunicarse utilizando medios heterogéneos, y en lugar de definir en un solo protocolo “gigante” todas las formas de comunicación, fue necesario dividir el problema de la comunicación en sub-problemas específicos y normalizar la relación entre los mismos.

En la actualidad, dos de los modelos mas conocidos para referenciar los protocolos de comunicación de datos en general son:

- Modelo OSI (Open Systems Interconnection) de ISO (Internacional Organization for Standardisation).
- Pila de protocolos TCP/IP (Transfer Control Protocol/Internet Protocol), estándar utilizado en la mayor parte de las redes actuales y en Internet.

2.1 El modelo de referencia OSI (Sistema de Interconexión Abierto)

A finales de los años setenta, la Organización Internacional de Estándares (ISO), creó un comité para desarrollar estándares de comunicación que promovieran la interoperatividad entre productos de sistemas abiertos (OSI), mismo que fue terminado y presentado en el año de 1,984

Este modelo es el más conocido y usado, pues aunque los protocolos hayan cambiado en los casi veinte años en concepto y diseño, su fortaleza está en que divide el problema general de la comunicación en problemas específicos y presenta una estructura formada de varias capas o niveles en los que se contempla desde el elemento mínimo de comunicación, hasta una aplicación completa.

Las capas se dividen en dos grupos:

1. Servicios de transporte (niveles 1, 2, 3 y 4).
2. Servicios de soporte al usuario (niveles 5, 6 y 7)

Por otro lado, aunque la mayoría de los protocolos actuales no encajan perfectamente con el modelo OSI, este se usa y ha quedado como un buen modelo de referencia y de estudio para comprender el andamiaje de los protocolos de comunicación.

2.1.1 Características del modelo OSI

El modelo OSI presenta las siguientes características:

- Tiene una estructura multinivel: Diseñado para que cada nivel se dedique a resolver tan solo una parte del problema de comunicación.
- Interacción entre niveles: Cada nivel se comunica con su similar en otros equipos, sin embargo debe hacerlo a través de los niveles inferiores en el mismo equipo. En otras palabras, el nivel N utiliza los servicios del nivel N-1 y proporciona servicios al nivel N+1.
- Dependencia: Debido a ese tipo de interacción, cada nivel depende del nivel inferior y superior de el mismo.
- Encabezados: En cada nivel, se agrega al mensaje un formato de control; este elemento permite que un nivel en el receptor se entere de que su contraparte en el emisor esta enviándole información.

2.1.2 Capas o niveles del modelo OSI

Figura 5. Capas o niveles del modelo OSI

7. Capa de Aplicación	Provee servicios al usuario final, (correo electrónico, compartición de ficheros, http, etc.)
6. Capa de Presentación	Realiza las conversiones de formato para lograr la comunicación de dispositivos
5. Capa de Sesión	Se encarga del sincronismo, manejo conexión, e intercambio de información y cierre.
4. Capa de Transporte	Proporciona control de extremo a extremo y el intercambio de información con el nivel requerido
3. Capa de Red	Da los medios para establecer y mantener las conexiones conmutadas entre los sistemas de usuario final
2. Capa de Enlace	Transferencia de datos fiable a través del medio físico mediante el envío de paquetes
1. Capa Física	Envío de bits entre cada extremo, normalizando aspectos mecánicos, eléctricos y funcionales

2.1.2.1 Capa Física

Se encarga de especificar la conexión física y el medio de comunicación entre el nodo y la red, y todas sus características físicas, mecánicas y eléctricas. Entre sus principales tareas están las siguientes:

- Especificar el tipo de conexiones físicas entre los dispositivos
- Especificar el aspecto mecánico de la interfase como por ejemplo la longitud y la calidad de los materiales.
- Especificar el aspecto eléctrico de la interfase como por ejemplo los consumos y los niveles de voltaje.
- Definir el tipo y la técnica de transmisión que se usara.
- Definir la velocidad de transmisión medida en cantidad de bits por unidad de tiempo

2.1.2.2 Capa de Enlace

Esta capa se encarga de tomar y organizar todos los dígitos 1's y 0's en formatos o grupos lógicos de información para enviarlos en forma segura a través del medio físico. Entre sus principales tareas están:

- Establecer el protocolo de comunicación que usaran los nodos de la red para acceder al medio de transmisión
- Definir la forma de acceso que el dispositivo debe seguir para transmitir y recibir mensajes.
- Llevar a cabo la transferencia de datos a través del enlace físico.
- Envío de datos con el control necesario para la sincronía.
- Detectar cualquier tipo de error en el nivel físico.

2.1.2.3 Capa de red

Esta capa controla la operación de la subred, eliminando los cuellos de botella que se originan al saturarse la red, para que los paquetes de la fuente lleguen a su destino final. Cumple las siguientes tareas:

- Definir el estado de los mensajes que se envían a nodos de la red.
- Establecer los mecanismos para determinar las rutas que seguirán los paquetes dentro de la red.
- Es responsabilidad de este nivel establecer, mantener y terminar las conexiones, además proporciona el enrutamiento de mensajes, determinando si un mensaje en particular deberá enviarse al nivel de transporte o bien al nivel de enlace.
- Controlar, conmutar y dirigir la congestión de los paquetes de información en una sub-red.

2.1.2.4 Capa de transporte

La capa de transporte es una especie de puente entre las tres capas superiores orientadas a las comunicaciones y las tres inferiores orientadas al procesamiento de datos. Su principal tarea es recibir los datos de las capas superiores, dividirlos en pequeñas cantidades y pasarlos a la capa de red, además debe:

- Proporcionar mecanismos para el intercambio de datos entre dispositivos finales de una red, de forma confiable.
- Asegurar que la llegada de datos del nivel de red encuentra las características de transmisión y calidad de servicio requerido por el nivel de sesión.

- Proporcionar los medios para habilitar y deshabilitar las conexiones entre los nodos.
- Determinar el protocolo que garantice el envío del mensaje.

2.1.2.5 Capa de sesión

Es la responsable de establecer la comunicación entre los anfitriones de una red, además:

- Mantiene puntos de verificación que sirven para que, ante una interrupción de transmisión por cualquier causa, la misma se pueda reanudar desde el último punto de verificación en lugar de repetirla desde el principio.
- Establece el inicio y termino de la sesión.
- Controla el diálogo, estableciendo el orden en que los mensajes deben fluir entre los usuarios finales.
- Permite escribir programas que correrán en cualquier instalación de red.

2.1.2.6 Capa de presentación

Su finalidad es encargarse de la presentación de los datos transmitidos, así que aunque intervengan diferentes equipos con diferentes codificaciones, los datos lleguen de una forma que se pueda reconocer. Sus funciones principales son:

- Definir el formato de los datos y de los mecanismos de traducción.

- Traducir el formato original del mensaje y asignar una sintaxis a los datos para que de esa forma se envíen a través de la red.
- Determinar la forma en que se van a presentar los datos sin preocuparse de su significado o semántica.
- Establecer la independencia a los procesos de aplicación considerando las diferencias en la representación de datos.
- Proporcionar servicios para el nivel de aplicaciones al interpretar el significado de los datos intercambiados.

2.1.2.7 Capa de aplicación

Esta capa se entiende en forma directa con el usuario final a través de programas de aplicación. Contiene además funciones de administración generalmente útiles para admitir aplicaciones distribuidas.

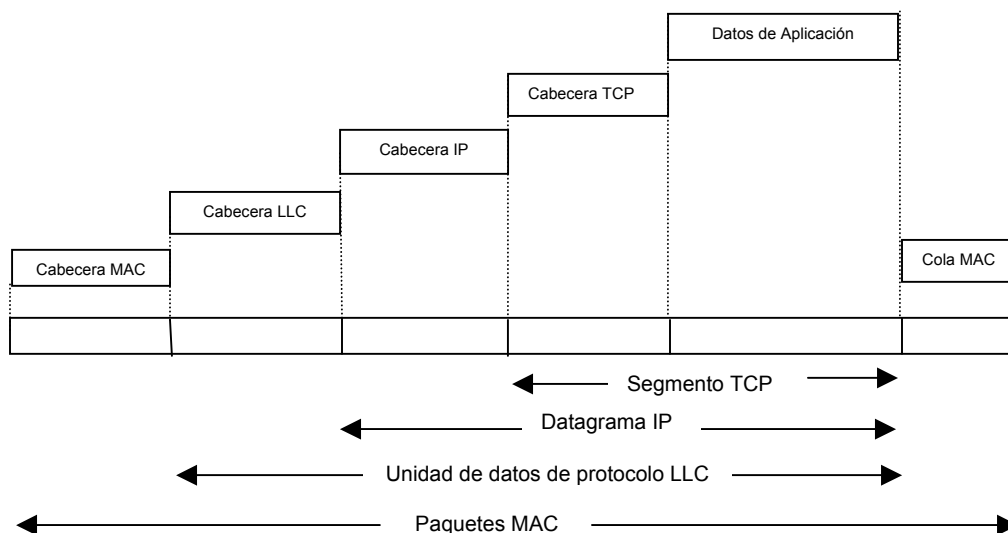
- Este es el punto de entrada que utilizan los programas para acceder al modelo OSI y utilizar los recursos de red, además contiene aplicaciones de uso general como transferencia de ficheros correo electrónico y acceso terminal a computadores remotos.
- Provee los servicios de red a las aplicaciones que serán útiles a los usuarios.
- Algunos ejemplos de aplicaciones son: Procesador de palabras, presentación grafica, base de datos etc.

2.2 El modelo de estratificación por niveles de TCP/IP

El segundo modelo nace de las investigaciones que se realizan respecto al conjunto de protocolos de TCP/IP. Con un poco de esfuerzo, el modelo OSI puede ampliarse y describir el esquema de estratificación por capas del TCP/IP, pero son lo suficientemente distintos para catalogarlos como tal.

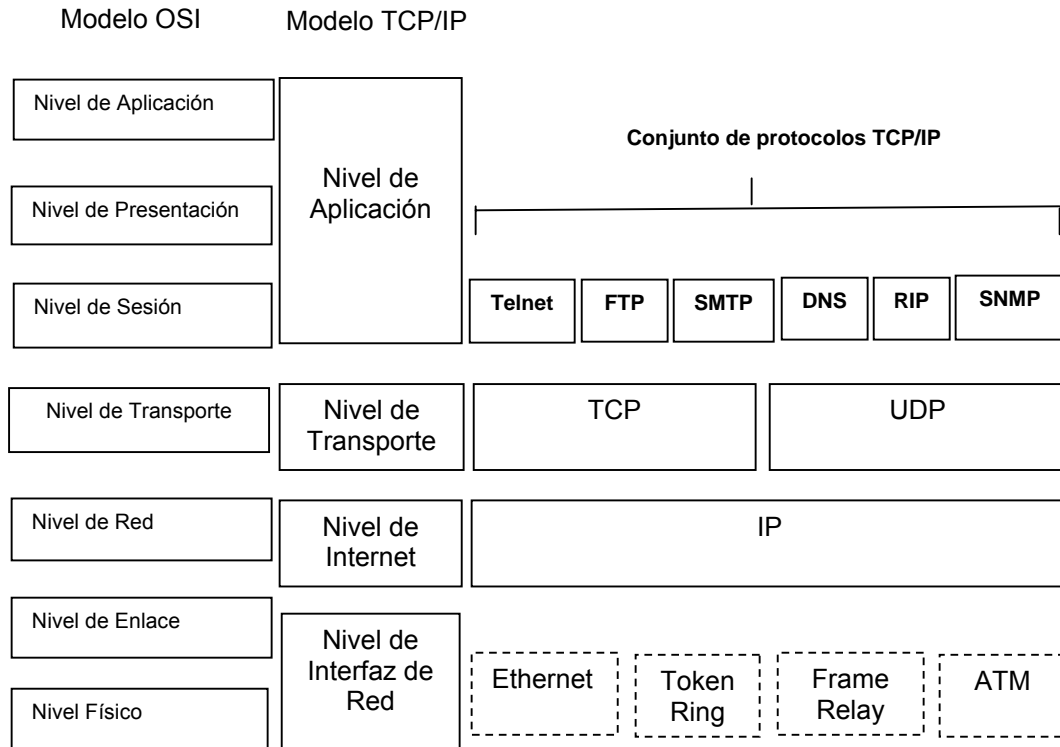
TCP/IP está basado en un modelo de referencia de cuatro niveles, y es la base de Internet que sirve para enlazar computadoras incluso utilizando diferentes sistemas operativos. En la siguiente figura se ilustra la forma en que se confeccionan las tramas en el modelo TCP/IP.

Figura 6. Confección de los paquetes en TCP/IP



Cada nivel del modelo TCP/IP corresponde a uno o más niveles del modelo de referencia OSI, esto se ve en la siguiente figura.

Figura 7. Comparación de los modelos OSI y TCP/IP



2.2.1 Nivel de aplicación

Este constituye el nivel mas alto de la pila TCP/IP y corresponde a las capas de aplicación, presentación y sesión del modelo OSI, representa las aplicaciones disponibles para el usuario.

Algunos de los protocolos pertenecientes a esta capa son por ejemplo el protocolo SMTP (correo electrónico), el protocolo FTP (transferencia de archivos) y el HTTP (transferencia de hipertexto). Básicamente, el programa de aplicación pasa los datos en la forma requerida hacia el nivel de transporte para su entrega.

2.2.2 Nivel de transporte

Este nivel es lo equivalente a lo definido para la capa de transporte del modelo OSI y tiene como finalidad, manejar los datos y proporcionar fiabilidad necesaria en el transporte de los mismos.

Aquí se divide el flujo de datos en paquetes, se añade bits para controlar el flujo de datos y el control de errores, y los pasa con una dirección de destino hacia la siguiente capa de transmisión. Esto se hace añadiendo información adicional a cada paquete, incluyendo códigos que identifican qué programa de aplicación envía y qué programa debe recibir, también una suma de verificación para asegurarse que el paquete llegue intacto.

2.2.3 Nivel de Internet

Este nivel equivale a lo definido para la capa de red del modelo OSI. Aquí se maneja la comunicación de una máquina a otra, aceptando una solicitud para enviar un paquete desde la capa de transporte, junto con una identificación de la máquina, hacia la capa destino. Eso es lo que hace el protocolo IP, pues se encarga de enviar los paquetes de información a sus destinos correspondientes.

2.2.4 Nivel de interfaz de red

El software TCP/IP de nivel inferior consta de una capa de interfaz de red responsable de aceptar los datagramas IP y transmitirlos hacia una red específica. Una interfaz de red puede consistir en un dispositivo controlador o un complejo subsistema que utiliza un protocolo de enlace de datos propios.

3. PROTOCOLOS COMUNMENTE EMPLEADOS EN EL ENTORNO INDUSTRIAL

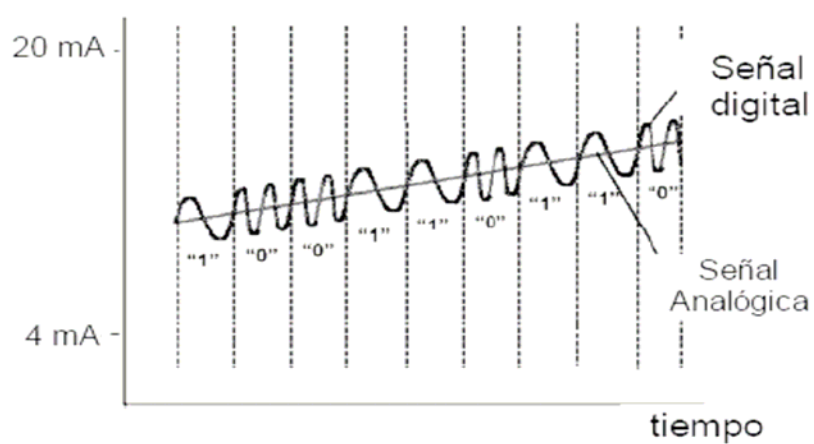
La gama de protocolos de comunicación, es amplia, todo depende de la necesidad. Los siguientes protocolos son algunos de los que tienen mayor presencia en el entorno industrial.

3.1 Protocolo HART

Es uno de los estándares líderes en la comunicación industrial con instrumentos inteligentes de campo, aquí se emite una señal a 1200 Hertz superpuesta a la señal del lazo (4-20 mA.), para enviar un "1", y una señal a 2200 Hertz para enviar un "0".

Estas dos señales coexisten en la misma línea y no interfieren una con otra porque la señal codificada tiene un promedio de corriente igual a cero y porque la señal HART es mucho mas alta que la señal de lazo (10 Hertz), esto se resuelve usando filtros pasa bajo y pasa alto.

Figura 8. La señal HART



Fuente: www.cea-ifac.com.es (Julio 2006)

3.1.1 Comandos del protocolo HART

Para llevar a cabo diferentes funciones preestablecidas, el protocolo utiliza comandos, que se dividen en tres grupos:

3.1.1.1 Comandos universales

Provee funciones que están implementadas en todos los dispositivos de campo, por ejemplo: fabricante, modelo, fecha, estado del dispositivo, etc.

3.1.1.2 Comandos de práctica común

Ejemplo: lectura de variables del dispositivo, calibración (cero y span), cambio de unidades de ingeniería, etc.

3.1.1.3 Comandos específicos

Provee funciones únicas para un dispositivo particular. Ejemplo: opciones de calibración, conversión de unidades de ingeniería, etc.

3.1.2 La señal física

Esta se refiere a la capa 1 del modelo de protocolos OSI. Se basa en el procedimiento de transacción de datos a la estructura de los mensajes, describiendo a detalle el contenido de cada byte.

3.1.2.1 Niveles de la señal

La especificación HART es que los maestros deben transmitir una señal de voltaje, sin embargo el esclavo maneja una señal de corriente. La señal de corriente se convierte a señal de voltaje mediante una resistencia de carga, de modo que todos los dispositivos utilizan receptores pasivos.

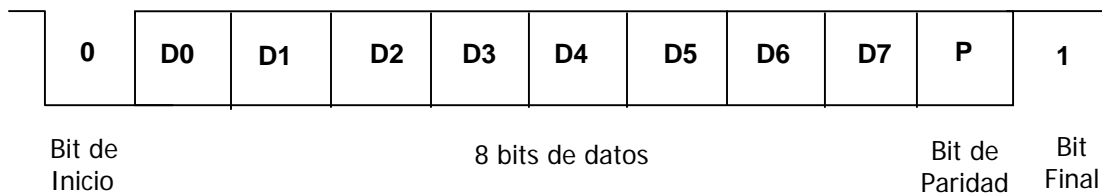
3.1.3 Procedimiento de transacción

HART es un protocolo "Half-Duplex". El maestro es el responsable de las transacciones de mensajes, por lo que, si no hay respuesta a un comando dentro de cierto tiempo, debe retransmitir el mensaje y después de unos intentos notificar el problema. La longitud y retardo típicos de los mensajes, permiten dos transacciones por segundo.

3.1.4 Codificación de caracteres

Hart codifica los mensajes como series de 8 bits. Estos se transmiten de modo serial, utilizando una "UART" (Universal Asynchronous Receiver/Transmitter) para serializar cada byte, añadiendo un bit de inicio, uno de paridad y uno de fin, esto permite que la UART receptora identifique el inicio de cada caracter, y detecte errores en la transmisión. Se muestra la cadena completa de bits en la siguiente figura.

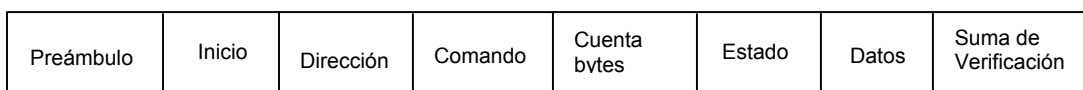
Figura 9. Cadena de bits HART



El bit menos significativo, D0 se envía primero. La mayoría de los protocolos seriales permiten pausas entre los caracteres; esto no es posible en Hart debido a las especificaciones de tiempo, de ocurrir dicho retraso se asume que la comunicación no fue establecida.

3.1.5 Formato del mensaje

Figura 10. Formato del mensaje HART



3.1.5.1 Preámbulo

Permite que el receptor sincronice la frecuencia de la señal con la cadena de caracteres que recibe, después de la detección inicial del mensaje. Contiene de 5 a 20 bytes con caracteres hexadecimales.

3.1.5.2 El inicio

Los receptores se encuentran en la búsqueda de estos caracteres cuando están en espera de un mensaje, como el primer carácter después de por lo menos dos caracteres FF, para indicar el inicio del mensaje.

3.1.5.3 La dirección

Contiene tanto la dirección del maestro como la del esclavo.

3.1.5.4 El comando

Contiene un entero del 0 al hexadecimal FD o al decimal 253, y representa la acción a realizar. El comando recibido se incluye en la respuesta del esclavo al ser enviada.

3.1.5.5 Cuenta de bytes

Indica el número de bytes que forman el resto del mensaje. El receptor lo utiliza para identificar el byte de suma de verificación y saber cuando el mensaje está completo.

3.1.5.6 El estado

Consta de dos bytes que reportan cualquier error de comunicación, el estado del comando recibido, y el estado de operación del esclavo. También es conocido como “código de respuesta”, y solo se incluye en el mensaje de respuesta de un esclavo.

3.1.5.7 Los datos

Los datos pueden estar en forma de enteros sin signo, números de punto flotante o cadenas de caracteres ASCII. El número y el formato de bytes, depende del comando recibido.

3.1.5.8 Suma de verificación

Contiene el OR exclusivo de todos los bytes que le preceden en el mensaje, comenzando con el carácter de inicio.

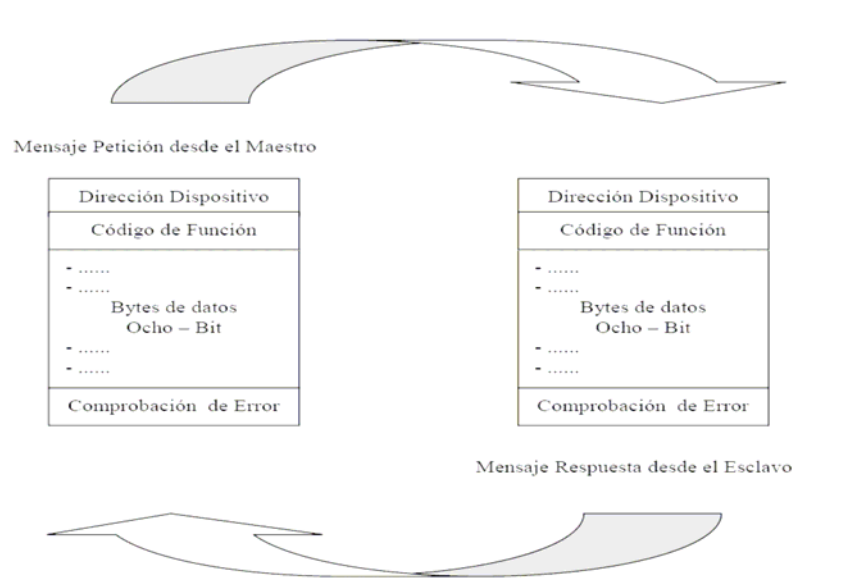
Esto provee un segundo chequeo para la integridad de la transmisión después del de paridad por byte. La combinación de estos dos garantiza la detección de hasta tres errores en un mensaje y tiene buenas probabilidades de detectar errores en más bits.

3.2 Protocolo Modbus

Es un protocolo utilizado en la automatización industrial del tipo maestro-esclavo entre dispositivos inteligentes, actualmente hay varias versiones del protocolo Modbus, estas son: ASCII, RTU y TCP.

El maestro puede direccionar esclavos individualmente o puede generar un mensaje en modo difusión a todos, estos devuelven un mensaje a las peticiones que les son direccionadas individualmente. Modbus establece el formato para la petición del maestro, colocando en ella la dirección del esclavo, un código de función que define la acción solicitada, los datos a enviar y un campo de comprobación de error.

Figura 11. Ciclo de interrogación-respuesta en MODBUS



Fuente: www.cea-ifac.es (Junio 2006)

3.2.1 La Petición

Indica al esclavo que acción realizar, en qué registro debe comenzar y cuántos ha de leer.

3.2.2 La respuesta

Si el esclavo elabora una respuesta normal, el código de función es una réplica del código de función enviado en la petición, si ocurre un error, el código de la respuesta es equivalente al original con el bit más significativo puesto en "1", además los bytes de datos contienen un código que describe el error.

3.2.3 Trama del Mensaje Modbus

En cualquier modo ASCII o RTU, un mensaje Modbus es situado por el maestro en una trama que tiene un comienzo y un final conocidos. Esto permite a los receptores comenzar en el arranque del mensaje, leer la parte de la dirección, determinar qué dispositivo es direccionado y conocer cuándo se ha completado el mensaje.

3.2.4 Modbus ASCII

Cuando se configura según el modo ASCII, cada byte en un mensaje se envía como dos caracteres ASCII. La principal ventaja de este modo es que permite intervalos de tiempo de hasta un segundo entre caracteres sin dar lugar a error.

El formato para cada byte en modo ASCII es el siguiente:

Tabla I. Formato en modo Modbus ASCII

Sistema de codificación	Bits por byte	Comprobación de error
Hexadecimal Caracteres ASCII 0-9, A-F	1 bit de arranque 7 bits de datos	Comprobación longitudinal redundante (LRC)
Un carácter hexadecimal contenido en cada carácter ASCII del mensaje	El menos significativo se envía primero	
	1 bit para paridad par o impar, ningún bit para "No" paridad, 2 bits si no se usa paridad	

3.2.4.1 Trama ASCII

En modo ASCII, los mensajes comienzan con un carácter (:) "dos puntos" y terminan con un par de caracteres (CRLF) "Retorno de Carro + Avance de Línea". Los caracteres a transmitir permitidos para todos los demás campos son 0-A, A-F hexadecimal. Los dispositivos conectados en red monitorizan el bus de red continuamente para detectar un carácter "dos puntos" cuando se recibe, cada dispositivo decodifica el próximo campo para enterarse si es el dispositivo direccionado.

Se muestra a continuación una trama típica de mensaje ASCII.

Figura 12. Trama del mensaje ASCII

ARRANQUE	DIRECCION	FUNCION	DATOS	COMPROBACION LRC	FINAL
1 Carácter :	2 Caracteres	2 Caracteres	N Caracteres	2 Caracteres	2 Caracteres CRLF

3.2.5 Modbus RTU

En modo RTU (Remote Terminal Unit), cada byte contiene dos dígitos hexadecimales de 4 bits. La principal ventaja de este modo es que su mayor densidad de carácter permite mejor rendimiento que el modo ASCII para la misma velocidad. Cada mensaje debe ser transmitido en un flujo continuo.

Tabla II. Formato en modo Modbus RTU

Sistema de codificación	Bits por byte	Comprobación de error
Binario, 8 bits Hexadecimal 0-9, A-F	1 bit de arranque 8 bits de datos El menos significativo se envía primero	Comprobación cíclica redundante (CRC)
Dos dígitos hexadecimales contenidos en cada campo de 8 bits del mensaje	1 bit para paridad par o impar, ningún bit para "No" paridad, 1 bit de paro si se usa paridad, 2 bits si no se usa paridad	

3.2.5.1 Trama RTU

Los mensajes comienzan con un intervalo silencioso de al menos 3.5 tiempos de carácter. Esto es mas fácil, implementado un múltiplo de tiempos de carácter a la velocidad configurada (T1-T2-T3-T4 figura 13). El primer campo es la dirección; por lo que, siguiendo al último carácter transmitido, un intervalo de al menos 3.5 tiempos de carácter señala el final del mensaje.

Si un intervalo silencioso de más de 1.5 tiempos de carácter tiene lugar antes de completar la trama, el receptor desecha el mensaje y asume que el próximo byte será la dirección de un nuevo mensaje. Igualmente, si un nuevo mensaje empieza antes de que transcurran 3.5 tiempos de carácter, el receptor lo considerará una continuación del mensaje previo, esto dará lugar a un error, ya que el valor en el campo CRC no será válido para el mensaje combinado.

Figura 13. Trama del mensaje RTU

ARRANQUE	DIRECCION	FUNCION	INFORMACION	COMPROB. CRC	FINAL
T1-T2-T3-T4	8 BITS	8 BITS	N x 8 BITS	16 BITS	T1-T2-T3-T4

3.2.5.1.1 La dirección

Contiene la dirección del esclavo seleccionado para la requisición. Mide un byte de tamaño y su valor está en el rango de 0–247 inclusive. Una dirección de “0” significa que el maestro enviara un mensaje a todos los esclavos.

Cuando el esclavo responde, sitúa su propia dirección para dar a conocer al maestro quien está respondiendo.

3.2.5.1.2 La función

Los códigos válidos están en el rango de 1 – 255 decimal, este indica al esclavo qué tipo de acción ha de ejecutar. Cuando el esclavo responde, utiliza el campo del código de función para indicar una respuesta normal o de error.

Tabla III. Tabla de funciones Modbus RTU

Función	Descripción
1	Lectura tabla de salidas
2	Lectura de tabla de entradas
3	Lectura de registros
4	Lectura de entradas análogas
5	Forzar salida simple
6	Preseteo registros simples
7	Lectura estado de excepción
8	Loopback maintenance
9-14	Funciones no soportadas
15	Forzar múltiples salidas
16	Preseteo de registros múltiples
17	Reporte de tipo de dispositivo
18-66	Funciones no soportadas
67	Read scratch pad memory
68-127	Funciones no soportadas
128-255	Reservadas para respuestas de excepción

3.2.5.1.3 El campo información

Contiene datos adicionales que debe usar para tomar la acción definida por el código de función; esto puede incluir partes como direcciones discretas y de registros, la cantidad de partes que han de ser manipuladas y el cómputo de bytes de datos contenidos en el campo.

Se construye utilizando conjuntos de 2 dígitos hexadecimales, en el rango de 00 a FF. Pueden formarse a partir de un par de caracteres ASCII o desde un carácter RTU, de acuerdo al modo de transmisión serie de la red.

3.2.5.1.4 El campo comprobación de error

Dos métodos son utilizados, dependiendo del método que esté siendo utilizado.

ASCII: El campo contiene dos caracteres ASCII, los caracteres de comprobación de error son el resultado de un cálculo de Comprobación Longitudinal Redundante (LRC) que es realizado sobre el contenido del mensaje, excluyendo los 'dos puntos' del comienzo y los caracteres CRLF de finalización. Los caracteres LRC son añadidos al mensaje como el último campo que precede a los caracteres CRLF.

RTU: Consta de dos bytes. El valor de comprobación de error es el resultado de un cálculo de Comprobación Cíclica Redundante (CRC) realizado sobre el contenido del mensaje. El campo CRC es añadido al mensaje como último campo del mensaje.

3.2.6 Transmisión de los caracteres en serie

Cuando los mensajes son transmitidos sobre redes Modbus, cada byte es enviado de izquierda a derecha. Primero el bit Menos Significativo (LSB) y luego el bit Mas Significativo (MSB)

Con trama de carácter ASCII, la secuencia de bits es:

Figura. 14. Secuencia de bits con trama ASCII



Con trama de carácter RTU, la secuencia de bits es:

Figura. 15. Secuencia de bits con trama RTU



3.2.7 Métodos de comprobación de error

Modbus utiliza: La comprobación de paridad y la comprobación de la trama (LRC o CRC). Ambas comprobaciones, son generadas en el maestro y aplicadas a los contenidos del mensaje antes de la transmisión, el esclavo comprueba cada carácter y trama del mensaje completo durante la recepción.

Este intervalo es suficientemente largo para que cualquier esclavo responda normalmente, si el esclavo detecta un error de transmisión, el mensaje no será tenido en cuenta. El esclavo no construirá una respuesta, así, el tiempo de espera expirará y permitirá al maestro tratar el error.

3.2.8 Control de paridad

Puede ser: Par, impar, o sin paridad; al hacer esto se contabilizará la cantidad de bits que tienen valor 1 en la porción de datos de cada carácter. Al bit de paridad habrá de darle valor 0 o 1, para que se obtenga un número par o impar respectivamente de bits con valor 1. Cuando se envía un mensaje, el bit de paridad es calculado y aplicado a la trama de cada carácter. El receptor cuenta la cantidad de bits de valor 1 y establece un error si no coincide con la paridad configurada.

3.2.9 Las direcciones en Modbus

Todas las direcciones en Modbus toman como referencia “cero”. Por ejemplo “la bobina 1” en un PLC, es direccionada como bobina 0000 en el campo de dirección de Modbus. Un poco más difícil es en el caso de los registros, por ejemplo el registro 40,001 es direccionado como registro 0000 en el campo de direcciones de un mensaje Modbus.

A continuación se presenta una tabla que muestra como se designan las direcciones en el un dispositivo por ejemplo un PLC, y en el formato del mensaje Modbus.

Tabla IV. Las direcciones en Modbus

Dirección de dispositivo	Dirección de Modbus	Descripción
1.....10,000	Dirección menos 1	Salidas
10001...20,000	Dirección menos 10,001	Entradas
40,000....50,000	Dirección menos 40,001	Registros

3.2.10 Modbus TCP/IP

Modbus TCP/IP es una variante del protocolo Modbus que permite utilizarlo sobre la capa de transporte TCP/IP; de este modo, Modbus-TCP se puede utilizar en Internet. Este se ha convertido en un estándar industrial de facto debido a su simplicidad, bajo coste, necesidades mínimas en componentes de hardware, y sobre todo es un protocolo abierto.

Se emplea para intercambiar información entre dispositivos, así como monitorizarlos y gestionarlos, también se emplea para la gestión de entradas/salidas distribuidas. La combinación de una red física versátil y escalable como Ethernet con el estándar universal de Inter redes TCP/IP y una representación de datos independiente del fabricante como Modbus, proporciona una red abierta y accesible para el intercambio de datos de proceso.

3.2.10.1 Fundamentos del protocolo Modbus TCP

Modbus/TCP simplemente encapsula una trama Modbus en un segmento TCP. TCP proporciona un servicio orientado a conexión fiable, lo que significa que toda consulta espera una respuesta.

Esta técnica de consulta/respuesta encaja perfectamente con la naturaleza maestro/esclavo de Modbus, añadido a la ventaja del determinismo que las redes Ethernet conmutadas ofrecen a los usuarios en la industria. El empleo del protocolo Modbus con TCP proporciona una solución para la gestión desde unos pocos a miles de nodos.

3.2.10.2 Prestaciones de un sistema Modbus TCP/IP

Las prestaciones dependen básicamente de la red y el hardware. Si se usa Modbus TCP/IP sobre Internet, las prestaciones serán las correspondientes a tiempos de respuesta en Internet, que no siempre serán las deseables para un sistema de control. Sin embargo pueden ser suficientes para la comunicación destinada a fines de mantenimiento, evitando así el traslado de personal hasta el lugar de la instalación.

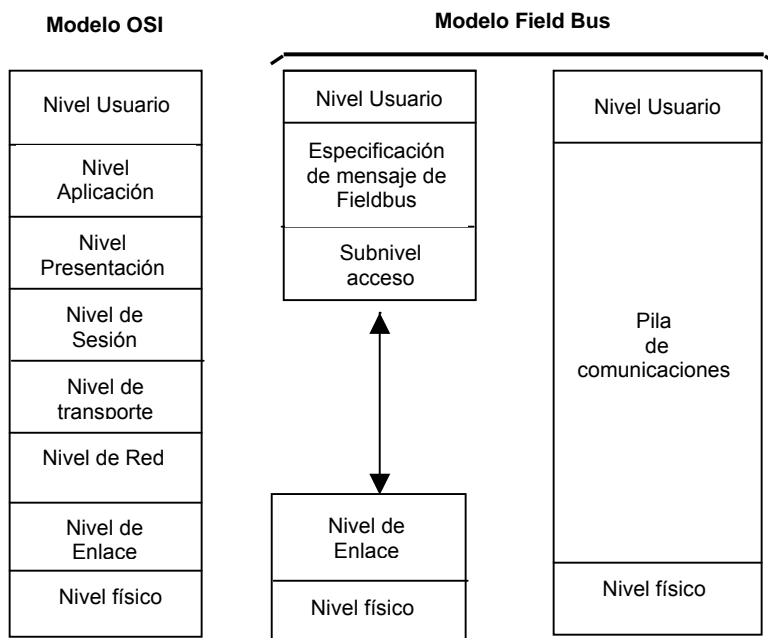
3.3 Protocolo Fieldbus Foundation

Es un protocolo para redes industriales especialmente para aplicaciones de control distribuido, esta desarrollado a partir del modelo OSI, puede comunicar grandes volúmenes de información y es ideal para aplicaciones con varios lazos complejos de control de procesos y automatización de la fabricación. Esta estructurado básicamente de:

- a) nivel físico,
- b) pila de comunicaciones y
- c) nivel de usuario.

No se implementan los niveles 3, 4, 5 y 6 del modelo OSI a causa que estos no se requieren en aplicaciones de control de proceso, aunque si se tiene en cuenta un importante nivel de usuario.

Figura 16. Estructura del Protocolo Fieldbus Foundation



3.3.1 Nivel físico

Corresponde a la capa 1 en el modelo OSI, la cual recibe mensajes codificados de las capas superiores y los convierte en señales físicas en el medio de transmisión del bus de campo.

3.3.2 Pila de comunicaciones

Provee los servicios de interfase entre los niveles físico y de usuario, tiene las siguientes características:

- Controla la transmisión de datos hacia y desde el nivel físico y el acceso a los medios de transmisión permitiendo una comunicación determinista.
- El Nivel de Aplicación comprende la transferencia de datos desde en Nivel 2 al Nivel 7 y el tratamiento de los comandos del nivel de usuario para acceder por su nombre los dispositivos remotos.

3.3.3 Nivel Usuario

Define una interfase permitiendo al usuario interactuar con los dispositivos de campo. Hay dos recursos importantes: los bloques y la descripción de dispositivos, entre los bloques los mas importantes son:

- a) Bloque de recurso: describe características del dispositivo tales como: nombre, fabricante, modelo y numero de serie.
- b) Bloque de función: son objetos que proveen acciones de control en base al comportamiento de las I/O del dispositivo.

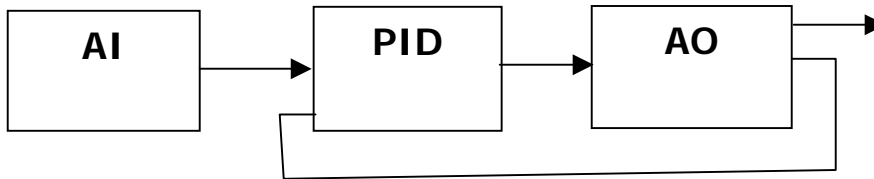
La tabla siguiente reúne algunas funciones usuales de control y de I/O.

Tabla V. Funciones de control y de I/O Fieldbus Foundation.

BLOQUE DE FUNCION	SIMBOLO
Entrada análoga	AI
Salida Análoga	AO
Entrada discreta	DI
Salida discreta	DO
Cargador manual	ML
Proporcional/Derivativo	PD
Proporcional/Integral/derivativo	PID

c) Bloque de transferencia: Manipula bloques de función de acuerdo al requerimiento de las I/O del dispositivo. El usuario crea aplicaciones sobre el bus de campo, conectando los bloques de función formando una estrategia de control distribuido, y pudiendo especificar en que tiempo y en que dispositivo se ejecutan. El esquema siguiente muestra un control de lazo cerrado utilizando 3 bloques de función, donde la entrada analógica corresponde a un transmisor de presión, mientras que el algoritmo PID y la salida analógica corresponde a un posicionador de válvula de control.

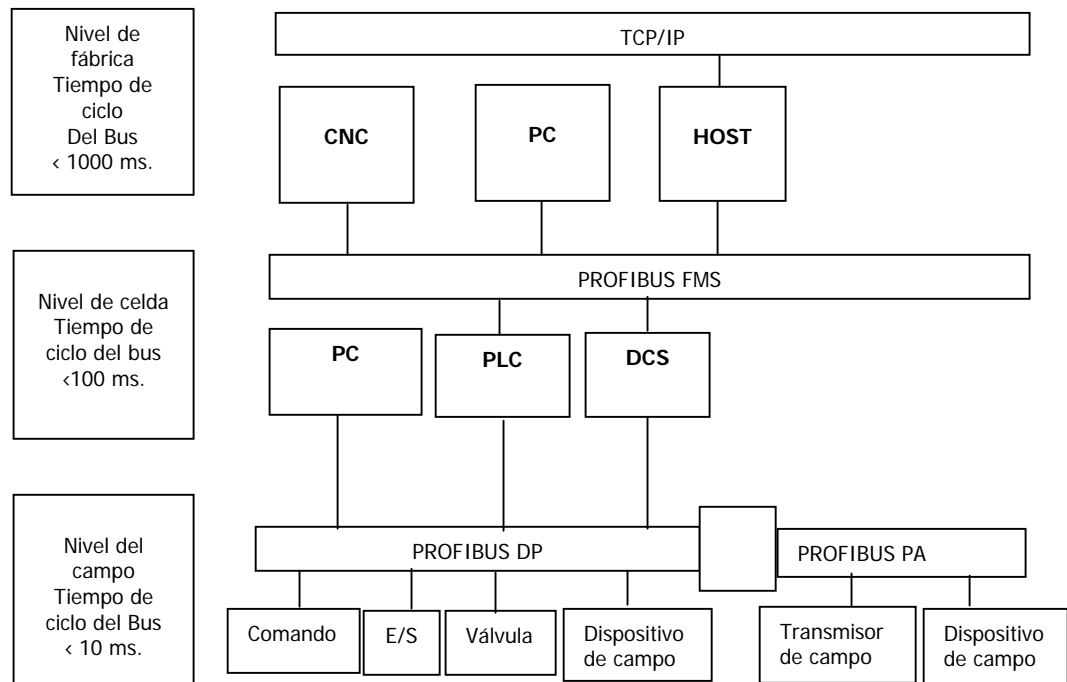
Figura 17. Ejemplo de lazo de control para Fieldbus



3.4 Protocolo Profibus

Es un estándar originado en normas europeas desarrollado a finales de la década de los 80, que cumple también con el modelo OSI. Es utilizado en aplicaciones de transmisión de datos de alta velocidad, y en la mayoría de aplicaciones industriales, gracias a las 3 implementaciones: FMS, DP y PA.

Figura 18. Estructura del protocolo Profibus.



3.4.1 Profibus DP

Esta diseñado para la comunicación entre los sistemas de control de la automatización y la entrada/salida distribuida en el nivel del dispositivo, donde importa más la velocidad que la cantidad de datos transmitidos.

En una red DP se tiene definido los niveles 1 y 2 del modelo OSI, no así los niveles 3 al 7. Se tiene definido el nivel de usuario y dispone de un servicio de intercomunicación con el nivel 2.

3.4.2 Profibus PA

Es la solución de Profibus para la automatización de procesos, ya que permite la comunicación de datos y alimentación de dispositivos sobre el mismo bus de dos hilos. Utiliza el mismo protocolo de transmisión que el DP, y ambos pueden ser integrados en la red con el uso de un segmento acoplador.

3.4.3 Profibus FMS

Es la solución de uso general para las tareas de la comunicación en el nivel de célula, es la más completa y esta diseñada para proveer facilidades de comunicación entre varios controladores programables como PLC's y PC's acceder a dispositivos de campo, permitir la gestión distribuida de procesos, ejecución remota de procesos, etc.

El nivel físico más frecuente usado por Profibus FMS es RS-485, manejando velocidades de transmisión desde 9.6 Kb/s hasta 12 Mb/s.

3.5 Protocolo CCM

Es un protocolo de Ge Fanuc, este transfiere datos usando el formato de datos serial asíncrono, y esta basado en el estándar ANSI X3.28, implementando carácter asíncrono usando un formato ASCII o 8-bit binario. CCM tiene dos modos de operación: par-a-par y maestro-esclavo.

Par-a-par es una configuración punto-a-punto donde solo dos dispositivos comparten una línea simple de comunicación; aquí, cualquier dispositivo sobre la línea puede iniciar la comunicación. El modo maestro-esclavo es usado en una configuración “multidrop” con un maestro y uno o mas esclavos pero solamente el maestro puede iniciar la comunicación.

3.5.1 Caracteres de control CCM

El protocolo CCM usa caracteres especiales de control ASCII ya sea para par-a-par o para maestro-esclavo. La tabla de abajo muestra los caracteres de control ASCII usados por el protocolo CCM.

Tabla VI. Caracteres de control ASCII

CARACTER	VALOR HEXADECIMAL	SIGNIFICADO
ENQ	05	Interrogar
ACK	06	Reconocimiento
NAK	15	Desconocimiento
SOH	01	Arrancar encabezado
ETB	17	Fin de bloque
STX	02	Arranque de texto
ETX	03	Fin de texto
EOT	04	Fin de transmisión

3.5.2 Secuencia de interrogación

Es una petición de iniciar comunicación enviada del maestro al dispositivo “par”. Si el dispositivo objetivo no esta ocupado, este envía una señal positiva de reconocimiento (ACK), de lo contrario, envía una señal negativa de reconocimiento (NAK).

3.5.3 Encabezado de transferencia

Después de una señal positiva, el dispositivo de inicialización envía 17 mensajes de carácter llamados: el encabezado. El bloque de encabezado especifica la dirección, tipo de memoria y el tamaño de los datos a transferir. Si el encabezado es correcto, la estación objetivo responderá con una señal positiva de reconocimiento (ACK), si es incorrecto enviara una señal negativa (NAK).

Se muestra a continuación el encabezado del protocolo CMM y su descripción.

Figura 19. Encabezado del protocolo CCM

S O H	ID Objetivo		Dirección Flujo de datos y memoria objetivo	Tipo de memoria Objetivo	Dirección De la memoria objetivo MSB		Dirección de la memoria objetivo LSB		No. De bloque de datos completos		No. De bytes en el ultimo bloque		ID Fuente		E T B	L R C
	2	3			6	7	8	9	10	11	12	13	14	15		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Tabla VII. Especificación de los campos del protocolo CCM

CAMPO	BYTE	DESCRIPCION	RANGO VALIDO
SOH	1	Arranque de carácter de control del encabezado	01h
ID objetivo	2,3	No. ID del dispositivo objetivo. Valor representado en código hex. ASCII	1-255 decimal (par-a-par) 1-90 decimal (maestro-esclavo)
Dirección flujo de datos/tipo de memoria objetivo	4,5	Bytes 4 and 5 supply target memory type. Byte 4 también especifica la dirección de los datos.	Variara dependiendo el tipo de dispositivo objetivo.
Dirección de memoria objetivo	6-9	Este campo especifica la dirección donde empezara la transferencia de datos.	Variara dependiendo el tipo de dispositivo objetivo
Contador de bloque de datos completos	10,11	No. De bloque de datos a ser transferidos (256 byte)	0-255 decimal
Contador de bytes en el ultimo bloque	12,13	El numero de bytes (menos de 256) en el final o solamente bloque de datos	0-255 decimal
ID Fuente	14,15	ID del dispositivo fuente	1-255 decimal (par-a-par) 1-90 decimal (maestro-esclavo)
ETB	16	Carácter de control Fin de bloque	17h
LRC	17	Chequeo de redundancia longitudinal. Este es un OR exclusivo de los bytes 2 al 15.	n/a

3.6 Protocolo DNP3

El protocolo de red distribuida versión 3.0 (DNP3) es un protocolo publico, abierto y optimizado, desarrollado para sistemas SCADA. Esto incluye comunicaciones de RTU a IED, comunicaciones de maestro a esclavo remoto, e incluso instancias de par-a-par y aplicaciones de red. Siendo un protocolo de capa de aplicación basado en objetos, DNP3 tiene la flexibilidad para soportar múltiples modos de operación.

Los mensajes pueden ser enviados entre el maestro DNP3 y el esclavo RTU mediante uno de los siguientes modos de operación.

3.6.1 Operación en reposo

El maestro no invoca al esclavo. El esclavo puede enviar reportes no solicitados y el maestro puede enviar confirmaciones de la capa de aplicación al esclavo.

3.6.2 Operación de reportes no solicitados

La comunicación no es solicitada, pero el maestro ocasionalmente envía votos de integridad para datos clase 0, para verificar que su base de datos esta actualizada.

3.6.3 Operación de reportes por votación

El maestro regularmente vota por un dato de evento y ocasionalmente por datos clase 0.

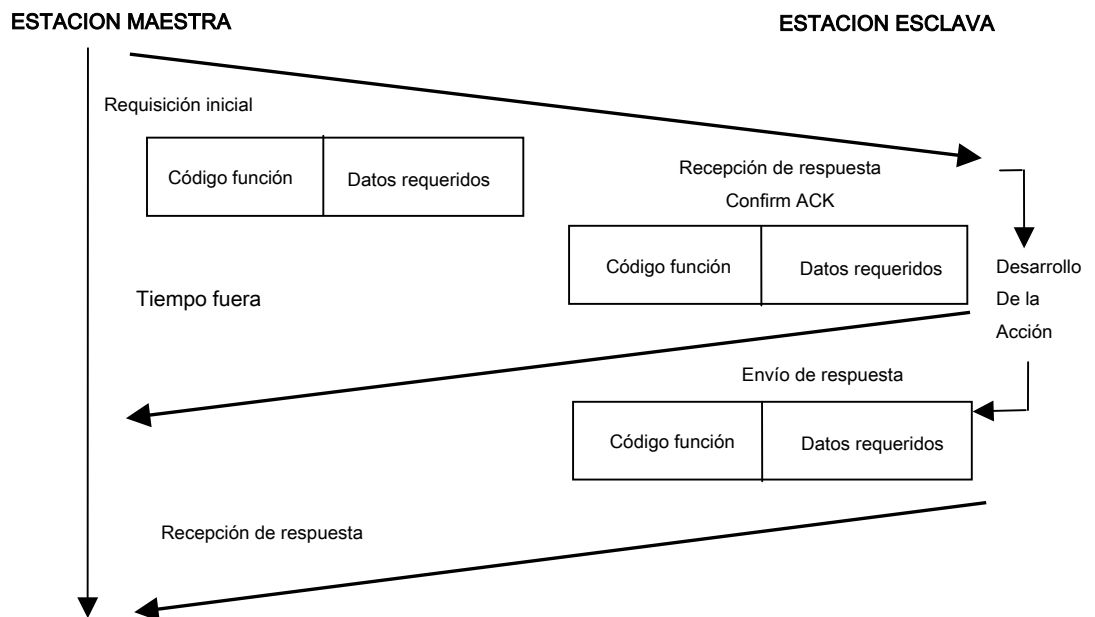
3.6.4 Operación de reportes estático

El maestro vota solo por datos clase "0" o datos específicos que requiera, puede direccionar esclavos individuales o iniciar un mensaje de difusión. DNP3 establece el formato para la requisición del mensaje del maestro poniéndolo dentro de la dirección del esclavo. La respuesta del esclavo contiene campos confirmando la acción tomada, cualquier dato para ser retornado y un campo de chequeo de error.

La figura 20 muestra al maestro iniciando una requisición, y la secuencia de comunicación mostrando la dirección de mensaje entre ellos.

El mensaje de requisición esta contenido en la capa de aplicación dentro del mensaje, una respuesta de confirmación (ACK) es requerida para este mensaje. El esclavo envía un mensaje ACK al maestro.

Figura 20. Proceso de comunicación Maestro-Esclavo DNP3.0



La pila DNP3 tiene tres capas: Capa de aplicación, Capa de enlace de datos, y la capa física. Para soportar funciones avanzadas de las unidades remotas (RTU) y largos mensajes, DNP3 agrega una pseudo-capa de transporte para usarse con la capa de enlace de datos.

3.6.5 Capa de aplicación

Describe el formato del mensaje, atributos, servicios y procedimientos para la capa de aplicación. La capa de aplicación construye mensajes basados en la necesidad del usuario de disponer datos de aplicación.

3.6.6 Capa de pseudo-transporte

Describe el formato del segmento, atributos, servicios y funciones de la capa de pseudo-transporte. La función principal de la capa de transporte es romper los fragmentos de datos de la capa de aplicación dentro de unidades de datos dimensionadas en la capa de enlace, segmentos llamados unidades de datos del protocolo de transporte (TPUD) para su transmisión y posterior armado en el fragmento original en la recepción.

3.6.7 Capa de enlace de datos

Maneja el enlace lógico entre maestro-esclavo y mejora las características físicas de error de canal. Agrega un bloque de encabezado seguido de bloques de datos opcionales para constituir la unidad de dato del protocolo de enlace (LPDU) o formato hasta 292 bytes de tamaño.

Los campos de bloque de encabezado, consisten de dos bytes “sync” que determinan donde empieza el formato; un byte “length” que especifica el numero de bytes pendientes en el formato, un byte “link control” usado entre capas de envío y recepción; dos bytes “destination address”; dos bytes “source address”; y dos bytes para chequeo de redundancia cíclica. La sección de datos contiene los datos bajados de la capa de pseudo-transporte.

El código de función de enlace de datos identifica el propósito del mensaje e indica que función se quiere desarrollar, este contiene la dirección, tipo del formato, e información del control de flujo.

3.6.8 Capa física

Convierte cada estructura a una tira de bits y los envía sobre el medio físico. En otra configuración, los protocolos TCP/IP son usados para proveer transporte a los mensajes DNP3 sobre la red, tal configuración es llamada capa de adaptación.

3.7 Protocolo CAN

CAN es un protocolo de comunicaciones serie orientado a mensajes, es decir la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Esta basado en arquitectura de bus para transferencia de mensajes en ambientes distribuidos. Originalmente fue diseñado para la industria automotriz pero ha alcanzado bastante aceptación en el área de control y automatización industrial. El protocolo CAN fue concebido como un protocolo de alta seguridad, cubriendo los niveles 1,2, y 7 del modelo OSI.

3.7.1 Capa física

La capa física en CAN es responsable de la transferencia de bits entre los distintos nodos que componen la red. Define aspectos como niveles de señal, codificación, sincronización y tiempos en los que los bits se transfieren al bus.

Tabla VIII. Velocidad versus distancia en protocolo CAN

Velocidad	Tiempo de Bit	Máxima longitud
1 Mbps.	1 μ S.	30 metros
800 Kbps.	1,25 μ S.	50 metros
500 Kbps.	2 μ S.	100 metros
250 Kbps.	4 μ S.	250 metros
125 Kbps.	8 μ S.	500 metros
50 Kbps.	20 μ S.	1000 metros
20 kbps.	50 μ S.	2500 metros
10 Kbps.	100 μ S.	5000 metros

3.7.2 Capa de enlace

Define las tareas independientes del método de acceso al medio. En redes multimaestro, la técnica de acceso al medio es muy importante ya que todo nodo activo tiene los derechos para controlar la red y acaparar los recursos. Por lo tanto la capa de enlace define el método de acceso al medio así como los tipos de tramas para el envío de mensajes.

3.7.3 Capa de aplicación

Existen diferentes estándares que definen la capa de aplicación algunos son muy específicos y están relacionados con sus campos de aplicación. Entre las capas de aplicación más utilizadas se puede mencionar: CAL, CANopen, DeviceNet etc.

3.8 Protocolo DeviceNet

Fue originalmente desarrollado por Allen Bradley pero luego paso a ser una especificación abierta, este es un bus basado en CAN. Su capa física y capa de enlace se basan en las normas ISO 11898. DeviceNet define una de las mas sofisticadas capas de aplicaciones industriales sobre el protocolo CAN.

Este protocolo utiliza una definición basada en orientación a objetos para modelar los servicios de comunicación y el comportamiento externo de los nodos. Define mensajes y conexiones para trabajar como maestro-esclavo, interrogación cíclica, mensajes espontáneos de cambio de estado, comunicación uno-uno, carga y descarga de bloques de datos etc. Se pueden conectar hasta 64 nodos con velocidades de 125 Kbps a 500 Kbps en distancias de 100 a 500 metros.

3.9 Protocolo SNP

SNP (protocolo de la serie noventa) es otro de los protocolos usados por Ge Fanuc, en especial por los PLC's de la serie 90. Es un protocolo Maestro/Esclavo, half duplex, a través del estándar recomendado RS-422.

3.9.1 Arquitectura

El protocolo SNP se divide en tres partes: Una capa de configuración, una capa de Sesión y una capa de Aplicación.

3.9.2 Capa de configuración

Consiste en los datos de configuración serial y temporizadores del protocolo, de manera de soportar un amplio número de aplicaciones obteniendo el óptimo desempeño.

Ejemplo de bloque de configuración:

Baud Rate:	300	600	2400	4800	9600	19.2K
Data Length:	8 bits					
Parity:	impar		par		ninguno	
Stop Bits:	1	2				
Max Link Idle Time:	1 a 60 segundos					
Modem Turn Around Time:	0 a 2550 milisegundos, en incrementos de 10ms.					

3.9.3 Capa de sesión

Esta capa trata con el formato de mensajes SNP, los tipos de mensajes SNP, el establecimiento y mantenimiento de una sesión de comunicación entre el dispositivo maestro y un PLC Ge Fanuc de la serie 90.

3.9.4 Capa de aplicación

Esta consiste en la requisición, que el maestro envía al PLC para leer o escribir datos en el, cargar lógica de programa al PLC o bajarla de el, o simplemente enviar comandos de arranque y paro a un PLC.

3.9.5 Procedimiento de transacción

Primero, la unidad maestra transmite una señal de pausa de unos tres pulsos de reloj, esta señal permite a los esclavos prepararse para aceptar el mensaje. Después, la unidad maestra espera un tiempo aproximado de 50 milisegundos para asegurarse que los esclavos estén listos.

Luego de la espera, el maestro enviara un mensaje “adherido” al esclavo, este mensaje contiene su dirección, así como el valor de un temporizador que será necesario para sincronización. El esclavo responde con un mensaje de “respuesta adherida”, que confirma el requerimiento y el reconocimiento del temporizador para futura sincronización.

Los esclavos envían un comando ACK para decirle al principal que han entendido el comando, este lo procesa y termina el comando. Finalmente la unidad principal envía un comando ACK el cual significa que ha recibido la confirmación de los esclavos.

4. EJEMPLO PRÁCTICO: SUMINISTRO DE INFORMACIÓN DE UNA PLANTA GEOTÉRMICA A UN SISTEMA SCADA

4.1 Planta geotérmica Orzunil I

La aplicación de los protocolos de comunicación en este ejemplo, se desarrolla en la planta geotérmica de generación Orzunil I, ubicada en el municipio de Zunil, Quetzaltenango; ya que, por requerimientos del Administrador del Mercado Mayorista AMM, es necesario enviar información del estado de la planta al sistema SCADA. Esta planta tiene una potencia firme instalada de 24 MW, generados mediante flujo geotérmico (agua caliente y vapor saturado) a través de siete unidades de generación llamadas OEC's (Ormat energy converter) de 4.5 MW nominales.

La planta, es del tipo binario lo cual quiere decir que el flujo geotérmico proveniente de los pozos de producción, no entran en contacto directo con las turbinas, sino que, su energía calorífica se transfiere por medio de intercambiadores de calor a un fluido motriz de bajo punto de ebullición, el cual es vaporizado por esta acción.

El ciclo del fluido motriz se basa en el ciclo de Rankine; en la etapa de precalentamiento, el agua caliente eleva la temperatura del fluido motriz; luego, en la segunda etapa, el vapor geotérmico lo eleva aun mas en temperatura y presión a un punto optimo para generar energía mecánica. El vapor del fluido motriz se expande en una turbina de alta presión y luego entra a una turbina de baja presión generando el movimiento del generador donde se genera la energía eléctrica a 13,200 voltios.

4.1.1 Automatismo de la planta geotérmica Orzunil I

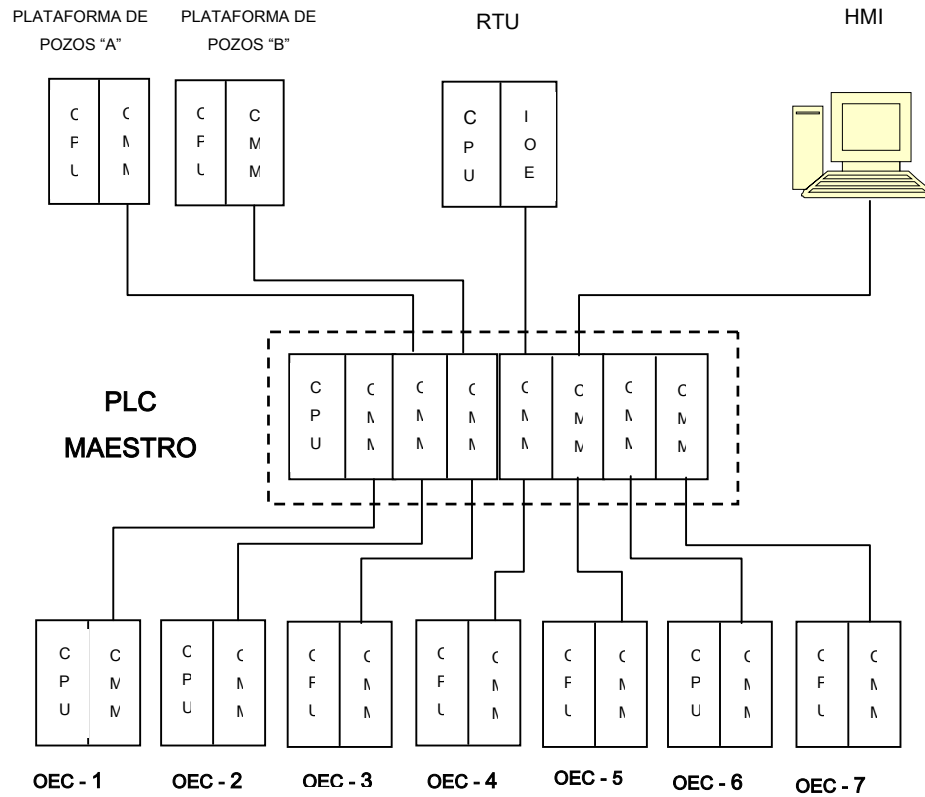
El control automático de la planta, es un Sistema de Control Distribuido SCD, un control central maestro y nueve esclavos remotos (siete para las unidades de generación y dos para los pozos de producción), conectado todo en una red tipo estrella (ver figura 21).

Este control se hace por medio de PLC's de la marca Ge Fanuc, conectados al PLC maestro mediante fibra óptica debido a las distancias considerables especialmente de los pozos de producción.

Los PLC's esclavos controlan de manera independiente la operación de las unidades de generación y de los pozos de producción, recibiendo únicamente vía remota los "settings" de generación y de presión respectivamente. Por otro lado, el PLC Maestro ubicado en el cuarto de control realiza tareas importantes como:

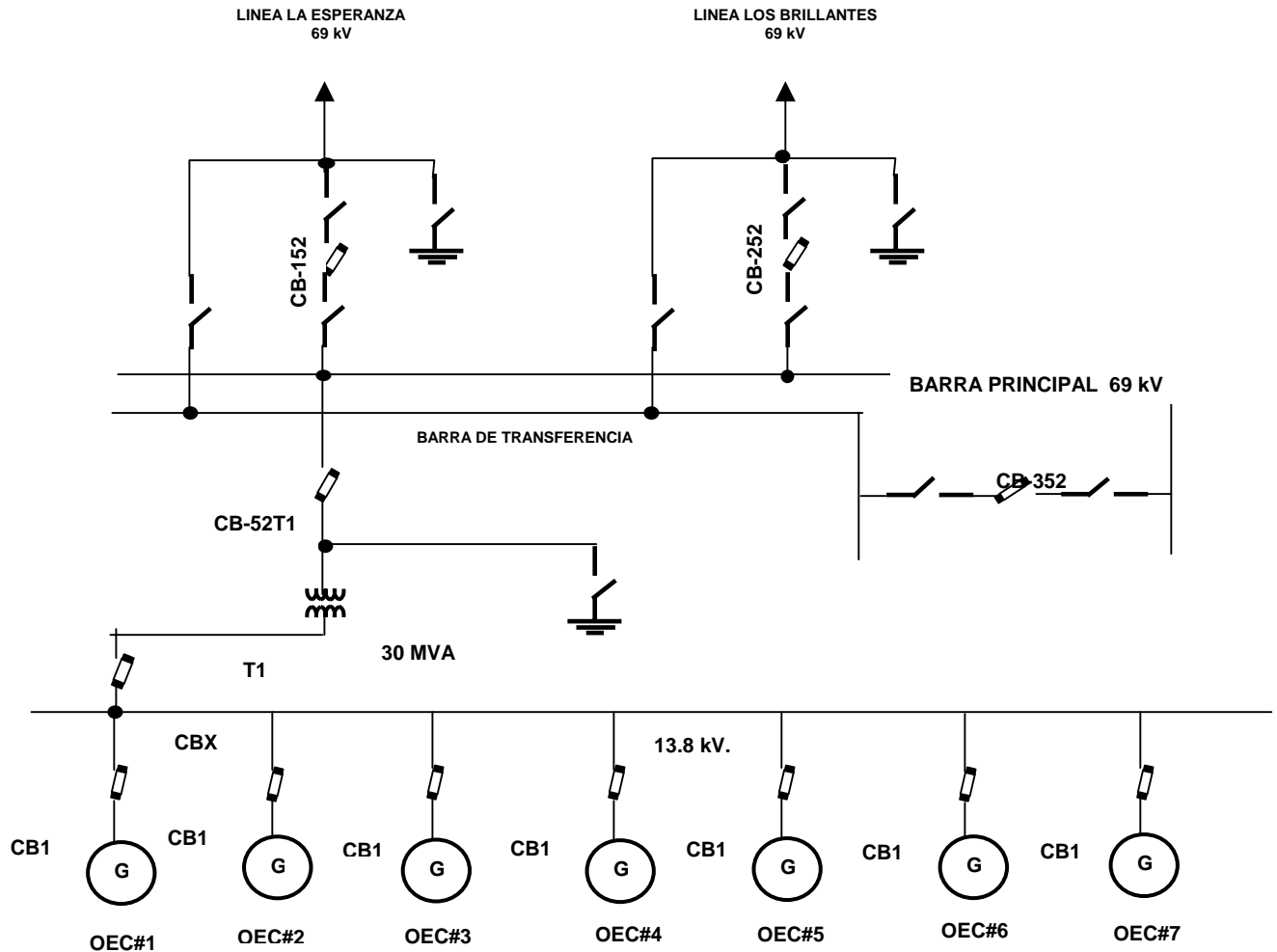
1. El control de los equipos auxiliares de la planta (planta de emergencia, compresores de aire, sistema contra incendios etc.).
2. Contiene algoritmos fundamentales para el funcionamiento de la planta, como por ejemplo "el lazo MSP" (main steam pressure).
3. Comunicarse con los PLC's esclavos, y a las PC's de la consola de mando para el recurso HMI (human machine interfase).
4. Comunicarse a la unidad RTU para el envío de datos al sistema SCADA.

Figura 21. Diagrama de comunicación planta geotérmica Orzunil 1



Como parte del procedimiento para definir las señales que deben ser transmitidas al sistema SCADA, es necesario elaborar un diagrama unifilar de la planta de generación Orzunil I, el cual se presenta en la siguiente figura.

Figura 22. Diagrama unifilar Planta Geotérmica Orzunil I



4.2 Descripción de los equipos y software a utilizar en el proyecto

4.2.1 El Controlador Lógico Programable (PLC)

Los PLC's usados en este proyecto son de la marca Ge Fanuc 90-70, estos son miembros de la familia de la serie 90 diseñados por "GE Fanuc Automation North America, Inc."

La arquitectura de la CPU esta basada en el microprocesador Intel 80X86, entre sus características principales están:

- Reloj entre 12MHz y 96MHz
- Alta densidad I/O en una simple tarjeta (32 puntos)
- Capacidad de programación en lenguaje C
- CPU de redundancia en línea para aplicaciones criticas
- Base de datos desde 32 KB hasta 6MB
- Diagnósticos de respaldo y subrutinas parametrizadas.

4.2.2 El Módulo de Comunicaciones del PLC

El “módulo coprocesador de comunicaciones” (CMM) es un miembro de la familia de los módulos de comunicación Ge Fanuc. Los protocolos CCM, RTU y SNP están disponibles en sus dos puertos en cualquiera de las nueve configuraciones siguientes:

- Solo CCM = Protocolo CCM en ambos puertos
- CCM/RTU = Protocolo CCM en puerto 1, protocolo RTU en el 2
- RTU/CCM = Protocolo RTU en puerto 1, protocolo CCM en el 2
- Solo RTU = Protocolo RTU en ambos puertos
- Solo SNP = Protocolo SNP en ambos puertos
- SNP/CCM = Protocolo SNP en puerto 1, protocolo CCM en el 2
- CCM/SNP = Protocolo CCM en puerto 1, protocolo SNP en el 2
- SNP/RTU = Protocolo SNP en puerto 1, protocolo RTU en el 2
- RTU/SNP = Protocolo RTU en puerto 1, protocolo SNP en el 2

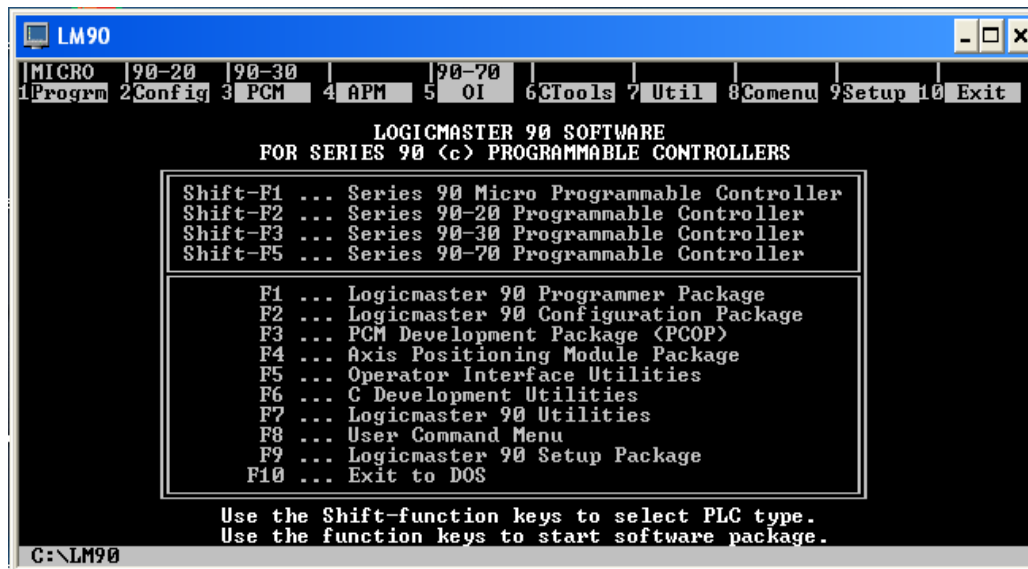
Características:

- Protocolo SNP (Maestro, Esclavo, Par)
- Protocolo Modbus RTU (Solamente esclavo)
- Microprocesador 80C186 de 12 Megahertz,
- Dos puertos seriales RS-422/RS-485 o RS-232

4.2.3 Software empleado

El software usado en los PLC's Ge Fanuc es: Logicmaster 90, versión 7.05, basado sobre la plataforma DOS. Logicmaster 90 es un software abierto y no se necesita licencia alguna para correr en computadores que tengan desde Windows 95 hasta Windows XP. La ventana principal del Logicmaster 90 es la que se muestra a continuación.

Figura 23. Ventana principal del Logicmaster 90



4.2.4 Unidad de transmisión remota RTU

La RTU que se usa en el ejemplo, es marca Callisto, de Siemens. Esta ha sido diseñada específicamente para propósitos en la industria eléctrica e impulsada por los requerimientos de las aplicaciones avanzadas en sistemas SCADA, automatización de la distribución, y monitoreo de carga.

Es de mencionar también que la unidad RTU cumple con las especificaciones que en su momento emitió el Administrador del Mercado Mayorista a través de la “Norma de Coordinación Operativa No. 2”. Las principales características de esta unidad remota son:

- Multiprocesador con arquitectura basado en Red de Área Local,
- Soporte para una colección de protocolos industriales estándar,
- Tiempo de secuencia de eventos con una resolución de 1 milisegundo
- Interfase de Sistema de Posicionamiento Global GPS, para tiempo de sincronización en sistemas amplios
- Facilidades de diagnóstico remoto
- Opción Interfase hombre-maquina HMI en Windows

4.2.5 Modulo de comunicaciones de la unidad RTU (IoE)

El modulo de comunicaciones (IoE) de la unidad Callisto, tiene cuatro puertos seriales de datos, los cuales pueden ser físicamente RS-485 o RS-232. Las opciones de comunicación son: comunicación síncrona/asíncrona, y protocolos orientados a byte o a bit.

El modulo loE puede usarse para intercambiar datos con centrales de control remotas, instrumentos de medición inteligentes, o relés distribuidos a través del sistema Callisto.

loE en conjunto con todos los módulos Callisto, utilizan tecnología ArcNET y protocolo Voyager DAQ's para enlazarse a redes LAN de alta velocidad.

4.2.6 Software empleado

El software Callisto permite a los usuarios definir todas las variables requeridas, a fin de operar totalmente una unidad remota. Este se divide

básicamente en tres partes:

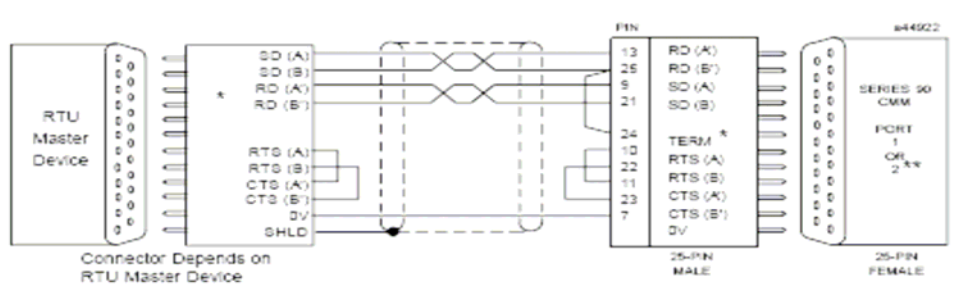
1. Pilot Logic Software (Programmable Integrated Logic Tool): es una herramienta útil para crear aplicaciones lógicas. Este utiliza diagrama de bloques con funciones pictoriales para permitir al usuario dibujar los algoritmos en un ambiente amigable, permitiendo funciones aritméticas, booleanas y funciones de comparación.
2. Callisto View Software: Ofrece facilidades para monitorear y configurar completamente sistemas remotos, desde parámetros de comunicación e interfases IED, hasta factores de escala análogos y calculos en los sistemas de potencia.

3. CalTerm software: CalTerm provee la facilidad de realizar diagnósticos, examinar datos en tiempo real, ejercitar la RTU tanto entradas como salidas y recuperar archivos de información; es decir, provee al usuario un interfase dentro de la RTU Callisto.

4.2.7 El cable de comunicaciones

Se ilustra en la siguiente figura, el diagrama del cable elaborado para las comunicaciones entre el PLC y la RTU, según el estándar recomendado RS-232. Se trata de un cable con hilos de cobre calibre 24 AWG con arreglo “Twisted-pair”.

Figura 24. Cable de comunicaciones PLC-RTU*



*Fuente: Manual del Usuario GFK-0582D

4.3 Ejecución del proyecto

4.3.1 Información requerida por el Administrador del Mercado Mayorista (AMM)

El AMM como ente operador del Sistema Nacional Interconectado (SNI) y para el eficiente monitoreo y control del mismo, requiere el envío de información de la Planta Orzunil I, hacia el sistema SCADA, la información contiene tanto señales discretas como continuas.

Tabla IX. Señales discretas requeridas

No.	SEÑALES DISCRETAS
1	Posición Abierto/Cerrado interruptor generador 1
2	Posición Abierto/Cerrado interruptor generador 2
3	Posición Abierto/Cerrado interruptor generador 3
4	Posición Abierto/Cerrado interruptor generador 4
5	Posición Abierto/Cerrado interruptor generador 5
6	Posición Abierto/Cerrado interruptor generador 6
7	Posición Abierto/Cerrado interruptor generador 7
8	Alarma generador 1
9	Alarma generador 2
10	Alarma generador 3
11	Alarma generador 4
12	Alarma generador 5
13	Alarma generador 6
14	Alarma generador 7

Tabla X. Señales análogas requeridas

No.	SEÑALES ANALOGAS
1	MW generador No. 1
2	MVAR generador No. 1
3	MW generador No. 2
4	MVAR generador No. 2
5	MW generador No. 3
6	MVAR generador No. 3
7	MW generador No. 4
8	MVAR generador No. 4
9	MW generador No. 5
10	MVAR generador No. 5
11	MW generador No. 6
12	MVAR generador No. 6
13	MW generador No. 7
14	MVAR generador No. 7

4.3.2 Configuración del PLC

En primer lugar, después de apagar la fuente de alimentación de voltaje al PLC Ge Fanuc 90-70, se inserta el modulo de comunicaciones "CMM" (modulo coprocesador de comunicaciones) en el "slot" numero 3 del "rack" 2 del PLC. Luego, después de energizar el PLC; desde la pantalla principal del "Logicmaster" (figura 23), se selecciona "F2" para acceder al modo de configuración. Se obtiene la siguiente pantalla:

Figura 25. Ventana de configuración Logicmaster 90.



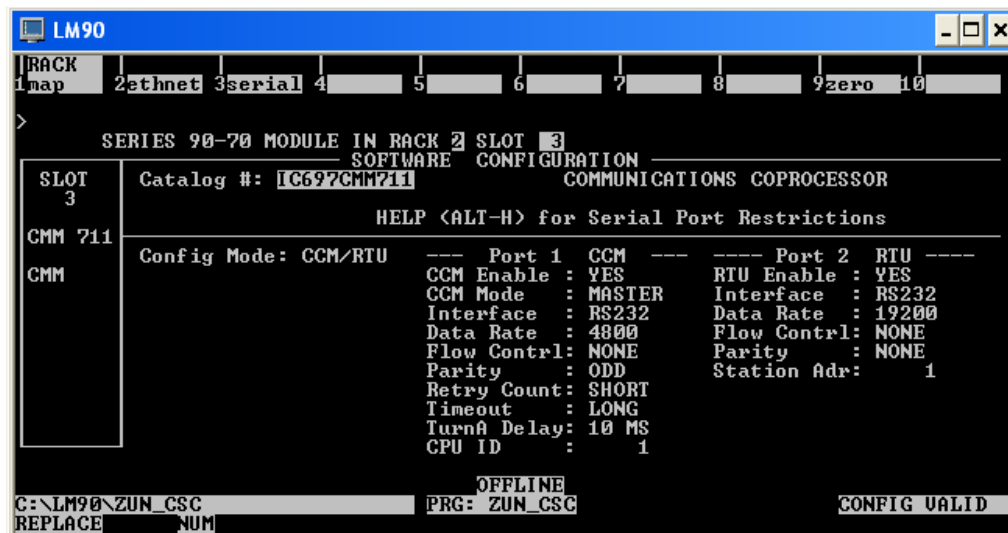
A continuación se presiona “F1” (I/O configuration) para configurar el módulo de comunicaciones y entablar la comunicación entre el PLC y la unidad RTU, en este ejemplo el módulo se agrega al “slot” 3 del “rack” 2, (ver figura siguiente).

Figura 26. Configuración del módulo de comunicaciones CMM.



Por ultimo seleccionar "F10" (zoom), para las opciones de modulo y de configuración. En este caso se elige protocolo CCM para el primer puerto y protocolo Modbus RTU para el segundo, este último es el que se utiliza para este proyecto. De esta manera y después de configurar los parámetros de puerto, queda listo el nuevo modulo de comunicaciones IC697CMM711. Se obtiene la siguiente pantalla:

Figura 27. Configuración de puertos modulo CMM



4.3.3 Localización de las señales requeridas

Se muestra a continuación las tablas con las señales requeridas, con su respectiva localización en el programa del PLC, es decir su registro (y bit en el caso de las señales discretas) y también su respectivo registro destino en la unidad RTU, según normas de protocolo Modbus descritas en la tabla numero IV.

Tabla XI. Ubicación de señales discretas en PLC y RTU

No.	Descripción	Dirección PLC	Dirección RTU
1	Posición interruptor generador 1 Abierto/Cerrado	R4271, bit 8	R6300, bit 1
2	Posición interruptor generador 2 Abierto/Cerrado	R4272, bit 8	R6300, bit 2
3	Posición interruptor generador 3 Abierto/Cerrado	R4273, bit 8	R6300, bit 3
4	Posición interruptor generador 4 Abierto/Cerrado	R4274, bit 8	R6300, bit 4
5	Posición interruptor generador 5 Abierto/Cerrado	R4275, bit 8	R6300, bit 5
6	Posición interruptor generador 6 Abierto/Cerrado	R4276, bit 8	R6300, bit 6
7	Posición interruptor generador 7 Abierto/Cerrado	R4277, bit 8	R6300, bit 7
8	Alarma Generador No. 1	R4278, bit 10	R6300, bit 8
9	Alarma Generador No. 2	R4279, bit 10	R6300, bit 9
10	Alarma Generador No. 3	R4571, bit 10	R6300, bit 10
11	Alarma Generador No. 4	R4572, bit 10	R6300, bit 11
12	Alarma Generador No. 5	R4573, bit 10	R6300, bit 12
13	Alarma Generador No. 6	R4574, bit 10	R6300, bit 13
14	Alarma Generador No. 7	R4575, bit 10	R6300, bit 14

Tabla XII. Ubicación señales análogas en PLC y RTU

No.	Descripción	Dirección PLC	Dirección RTU
1	MW Generador No. 1	R4576	R6301
2	MVAR Generador No. 1	R4577	R6302
3	MW Generador No. 2	R4578	R6303
4	MVAR Generador No. 2	R4579	R6304
5	MW Generador No. 3	R4771	R6305
6	MVAR Generador No. 3	R4772	R6306
7	MW Generador No. 4	R4773	R6307
8	MVAR Generador No. 4	R4774	R6308
9	MW Generador No. 5	R4775	R6309
10	MVAR Generador No. 5	R4776	R6310
11	MW Generador No. 6	R4777	R6311
12	MVAR Generador No. 6	R4778	R6312
13	MW Generador No. 7	R4779	R6313
14	MVAR Generador No. 7	R5171	R6314

4.3.4 Programación del PLC

A continuación, se programan y cargan las nuevas rutinas en el programa del PLC. En Logicmaster 90, el programa se divide en bloques por lo que las nuevas rutinas se cargan en el bloque llamado: “RTU-communications”. El objetivo de la lógica agregada al programa en el PLC es el siguiente: Trasladar el “estado actual” de las variables requeridas a los “registros a leer” por la unidad RTU. Esto se hace mediante las funciones de bloque “BIT TEST” para las señales discretas y “MOVE INT” para las señales análogas. Se presentan las definiciones concernientes:

4.3.4.1 El bloque “BIT TEST”

Esta función es usada para chequear si un bit dentro de una cadena de bits tiene el valor actual de “1” o “0”. El resultado es colocado en la salida “Q”

Figura 28. Bloque: “Bit test”

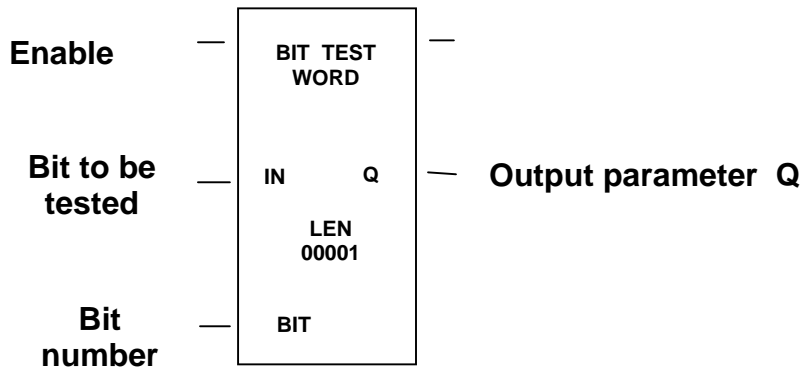


Tabla XIII. Descripción de parámetros bloque BIT TEST

Parámetro	Descripción
Enable	Cuando la función es habilitada, el chequeo de bit es desarrollado
IN	Contiene la primera palabra de datos a ser operada
BIT	Contiene el número de bit de IN que tiene que ser chequeado. El rango válido es $(1 \leq \text{BIT} \leq (16 * \text{LEN}))$.
Q	Q es energizada si el bit chequeado fue “1”.
LEN	Es el número de palabras en la cadena a ser chequeado

4.3.4.2 El bloque “MOVE INT”

Esta función es usada para copiar datos como bits individuales desde una localidad a otra. Consta de dos parámetros de entrada y dos de salida.

Figura 29. Bloque “Move integer”

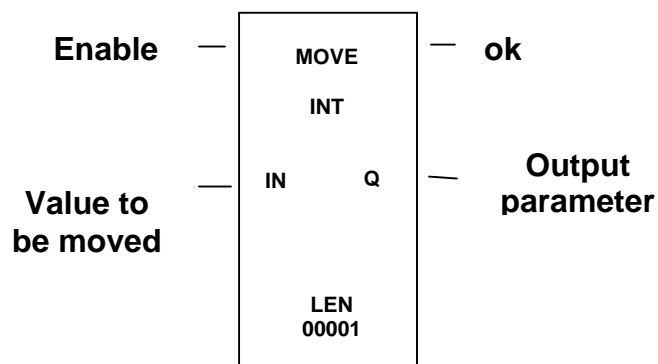


Tabla XIV. Descripción de parámetros bloque MOVE INT

Parámetro	Descripción
Enable	Cuando la función es habilitada, el traslado es desarrollado
IN	Contiene el valor a ser movido.
Ok	Q es energizada siempre que la función sea habilitada
Q	Cuando el movimiento es desarrollado, el valor en IN es escrito en Q.
LEN	Especifica el número de palabras o bits a ser movidos.

A continuación se presenta la programación agregada al PLC maestro, que realiza la acción descrita en el párrafo denominado: Programación del PLC.

```

|
| << RUNG 101 >>
|
|         +-----+
+-----+ BIT_+-
|         |TEST_|
|         | WORD|                                     %T00001
| %R04271-+|N Q+------( )--
|         | LEN |
|         |00001|
| CONST -+BI T |
| 00008 +-----+
|
| << RUNG 102 >>
|
|         +-----+
+-----+ BIT_+-
|         |TEST_|
|         | WORD|                                     %T00002
| %R04272-+|N Q+------( )--
|         | LEN |
|         |00001|
| CONST -+BI T |
| 00008 +-----+
|
| << RUNG 103 >>
|
|         +-----+
+-----+ BIT_+-
|         |TEST_|
|         | WORD|                                     %T00003
| %R04273-+|N Q+------( )--
|         | LEN |
|         |00001|
| CONST -+BI T |
| 00008 +-----+
|
| << RUNG 104 >>
|
|         +-----+
+-----+ BIT_+-
|         |TEST_|
|         | WORD|                                     %T00004
| %R04274-+|N Q+------( )--
|         | LEN |
|         |00001|
| CONST -+BI T |
| 00008 +-----+
|

```

```

| << RUNG 105 >>
|
|      +-----+
+-----+ BIT_+
|      |TEST_|
|      | WORD|                                %T00005
| %R04275-IN Q+------( )--
|      | LEN |
|      |00001|
| CONST -+BI T |
| 00008 +-----+
|
| << RUNG 106 >>
|
|      +-----+
+-----+ BIT_+
|      |TEST_|
|      | WORD|                                %T00006
| %R04276-IN Q+------( )--
|      | LEN |
|      |00001|
| CONST -+BI T |
| 00008 +-----+
|
| << RUNG 107 >>
|
|      +-----+
+-----+ BIT_+
|      |TEST_|
|      | WORD|                                %T00007
| %R04277-IN Q+------( )--
|      | LEN |
|      |00001|
| CONST -+BI T |
| 00008 +-----+
|
| << RUNG 108 >>
|
|      +-----+
+-----+ BIT_+
|      |TEST_|
|      | WORD|                                %T00008
| %R04278-IN Q+------( )--
|      | LEN |
|      |00001|
| CONST -+BI T |
| 00010 +-----+
|

```

```

| << RUNG 109 >>
|
|           +-----+
+-----+ BIT_+
|           |TEST_|
|           | WORD|                                     %T00009
| %R04279-+|N  Q+------( )--
|           | LEN |
|           |00001|
| CONST -+BI T |
| 00010 +-----+
|
| << RUNG 110 >>
|
|           +-----+
+-----+ BIT_+
|           |TEST_|
|           | WORD|                                     %T00010
| %R04571-+|N  Q+------( )--
|           | LEN |
|           |00001|
| CONST -+BI T |
| 00010 +-----+
|
| << RUNG 111 >>
|
|           +-----+
+-----+ BIT_+
|           |TEST_|
|           | WORD|                                     %T00011
| %R04572-+|N  Q+------( )--
|           | LEN |
|           |00001|
| CONST -+BI T |
| 00010 +-----+
|
| << RUNG 112 >>
|
|           +-----+
+-----+ BIT_+
|           |TEST_|
|           | WORD|                                     %T00012
| %R04573-+|N  Q+------( )--
|           | LEN |
|           |00001|
| CONST -+BI T |
| 00010 +-----+
|

```

```

| << RUNG 113 >>
|
|          +-----+
+-----+ BIT_+
|          |TEST_|
|          | WORD|                                %T00013
| %R04574--+IN Q+------( )--
|          | LEN |
|          |00001|
| CONST --+BI T |
| 00010 +-----+
|
| << RUNG 114 >>
|
|          +-----+
+-----+ BIT_+
|          |TEST_|
|          | WORD|                                %T00014
| %R04575--+IN Q+------( )--
|          | LEN |
|          |00001|
| CONST --+BI T |
| 00010 +-----+
|
| << RUNG 115 >>
|
|          +-----+
+-----+MOVE_+
|          | WORD|
|          |   |
| %T00001--+IN Q+-%R06300
|          | LEN |
|          |00001|
|          +-----+
|
| << RUNG 116 >>
|
|          +-----+          +-----+          +-----+
+-----+MOVE_+-----+MOVE_+-----+MOVE_+
|          | INT |          | INT |          | INT |
|          |   |          |   |          |   |
| %R04576--+IN Q+-%R06301 %R04577--+IN Q+-%R06302 %R04578--+IN Q+-%R06303
|          | LEN |          | LEN |          | LEN |
|          |00001|          |00001|          |00001|
|          +-----+          +-----+          +-----+
|

```

```

| << RUNG 117 >>
|
|          +-----+          +-----+          +-----+
+-----+MOVE_+-----+MOVE_+-----+MOVE_+
|          | INT |          | INT |          | INT |
|          |     |          |     |          |     |
| %R04579-+I N  Q+-%R06304 %R044771-+I N  Q+-%R06305 %R04772-+I N  Q+-%R06306
|          | LEN |          | LEN |          | LEN |
|          |00001|          |00001|          |00001|
|          +-----+          +-----+          +-----+
|
| << RUNG 118 >>
|
|          +-----+          +-----+          +-----+
+-----+MOVE_+-----+MOVE_+-----+MOVE_+
|          | INT |          | INT |          | INT |
|          |     |          |     |          |     |
| %R04773-+I N  Q+-%R06307 %R04774-+I N  Q+-%R06308 %R04775-+I N  Q+-%R06309
|          | LEN |          | LEN |          | LEN |
|          |00001|          |00001|          |00001|
|          +-----+          +-----+          +-----+
|
| << RUNG 119 >>
|
|          +-----+          +-----+          +-----+
+-----+MOVE_+-----+MOVE_+-----+MOVE_+
|          | INT |          | INT |          | INT |
|          |     |          |     |          |     |
| %R04776-+I N  Q+-%R06310 %R054777-+I N  Q+-%R06311 %R04778-+I N  Q+-%R06312
|          | LEN |          | LEN |          | LEN |
|          |00001|          |00001|          |00001|
|          +-----+          +-----+          +-----+
|
| << RUNG 120 >>
|
|          +-----+          +-----+
+-----+MOVE_+-----+MOVE_+
|          | INT |          | INT |
|          |     |          |     |
| %R04779-+I N  Q+-%R06313 %R05171-+I N  Q+-%R06314
|          | LEN |          | LEN |
|          |00001|          |00001|
|          +-----+          +-----+
+

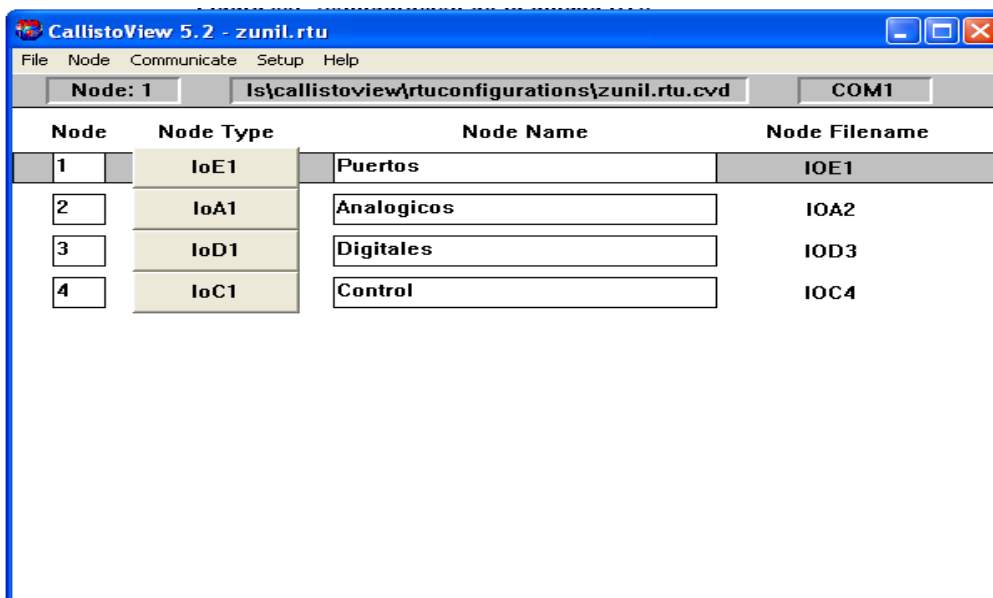
```

[END OF BLOCK LOGIC]

4.3.5 Configuración de la unidad RTU

Esto se hace usando la herramienta “CallistoView” para agregar a la RTU cada módulo (Nodo) según las especificaciones físicas. A continuación una figura que muestra la ventana de configuración.

Figura 30. Configuración de la unidad RTU

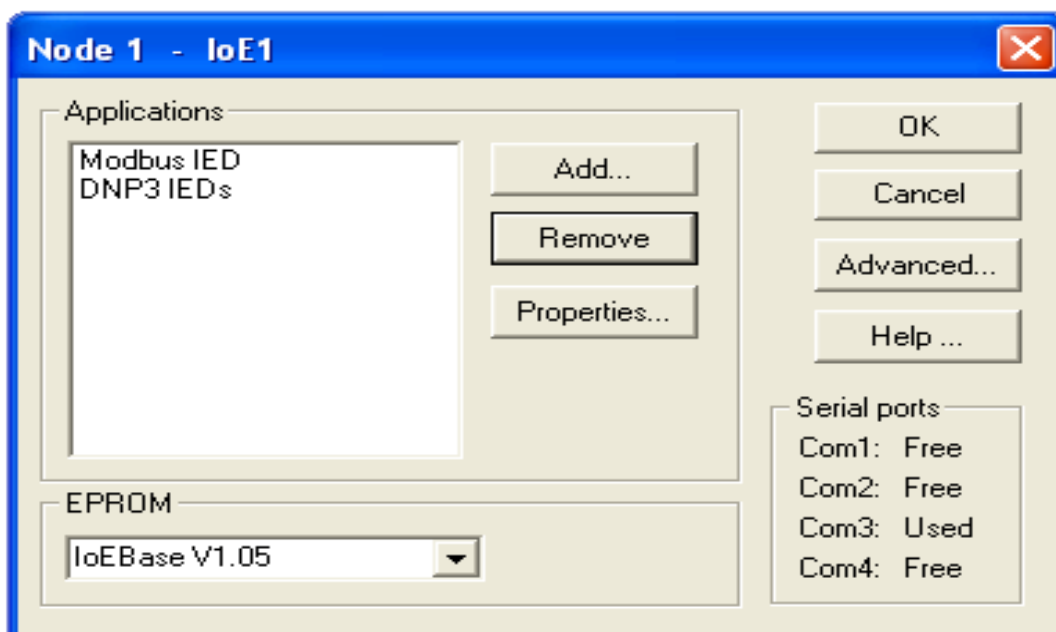


Luego de agregar los módulos se procede a configurar cada uno, en este ejemplo solo se considera el modulo de comunicaciones IoE que es el que servirá para recibir información del PLC y enviar la misma hacia el sistema SCADA. Esto se hace, modulando primeramente la señal eléctrica en el puerto de salida RS-232 a través de un modulador y convirtiéndola en señal de microondas.

Se configuran los parámetros en el modulo de comunicaciones partiendo de la elección del protocolo a utilizar para la comunicación con el PLC y el SCADA. En este ejemplo se elige el protocolo Modbus para el primer caso; ya que, se eligió anteriormente este mismo protocolo para el modulo de comunicaciones del PLC, pues si algo debe cumplirse estrictamente es que para comunicar dos dispositivos, estos deben configurarse con el mismo protocolo de comunicación.

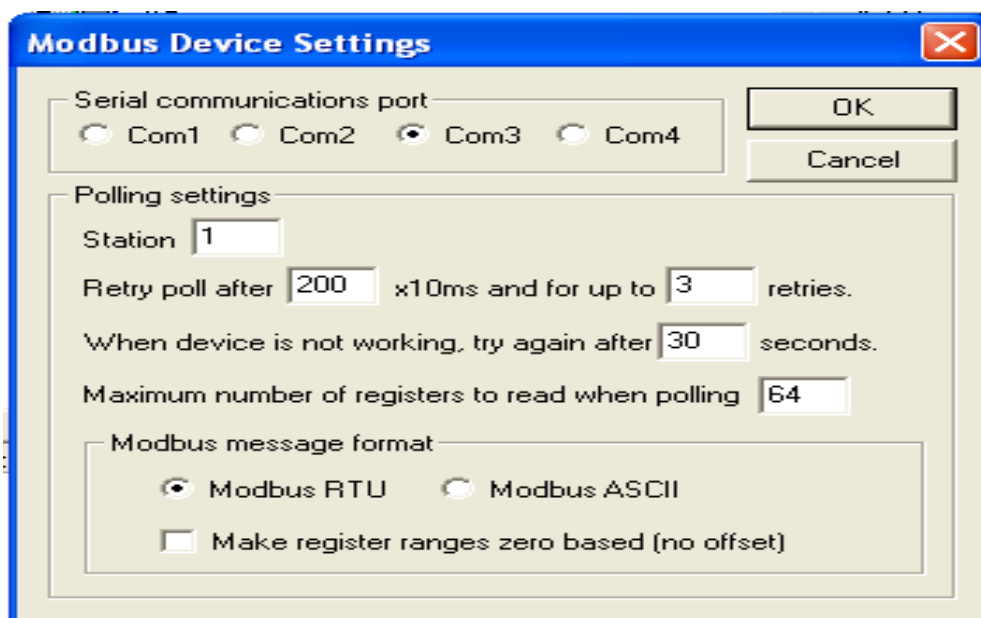
Para el segundo puerto, para el envío de la información al sistema SCADA se elige protocolo DNP3.0. La ventana de configuración se muestra en seguida.

Figura 31. Configuración de puertos



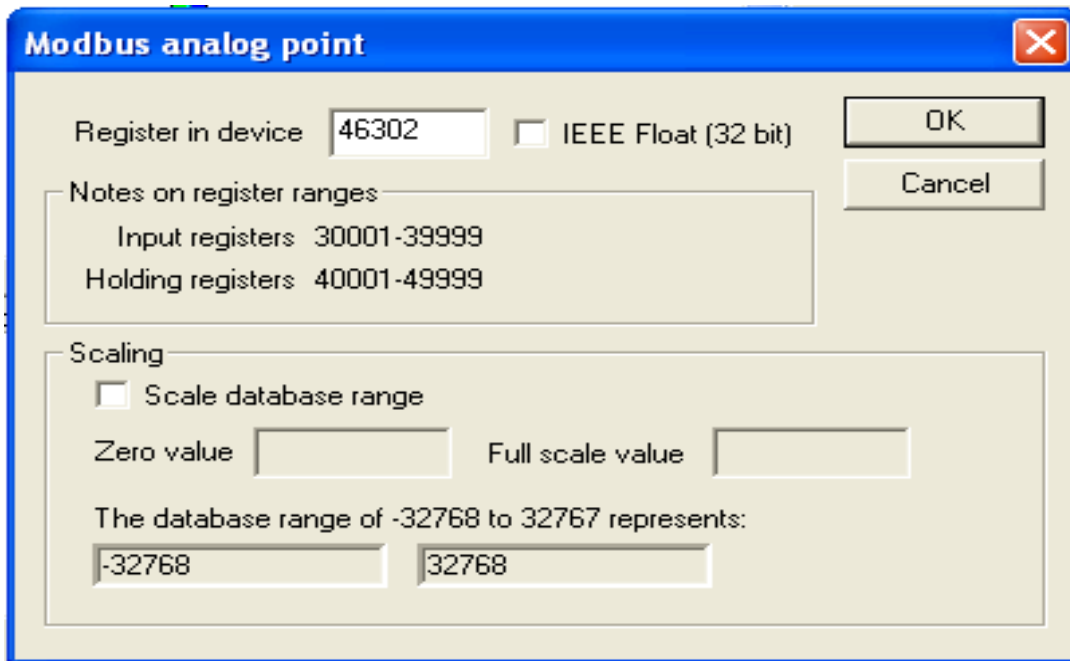
Para la configuración del formato de mensaje y otros parámetros de comunicación en el primer puerto, se selecciona el botón “Properties” teniendo seleccionada la opción de “Modbus IED”. Se obtiene la siguiente pantalla:

Figura 32. Configuración en Modbus



A continuación se direccionan individualmente los registros a leer desde el PLC tomando en cuenta las normas de Modbus. Por ejemplo, para direccionar el registro 6,301 que contiene la señal de “MW Generador No. 1” (ver tabla numero XII) debe sumarse 40,001, dando como resultado 46,302. Esto se ve en la figura siguiente.

Figura 33. Configuración de señales individuales

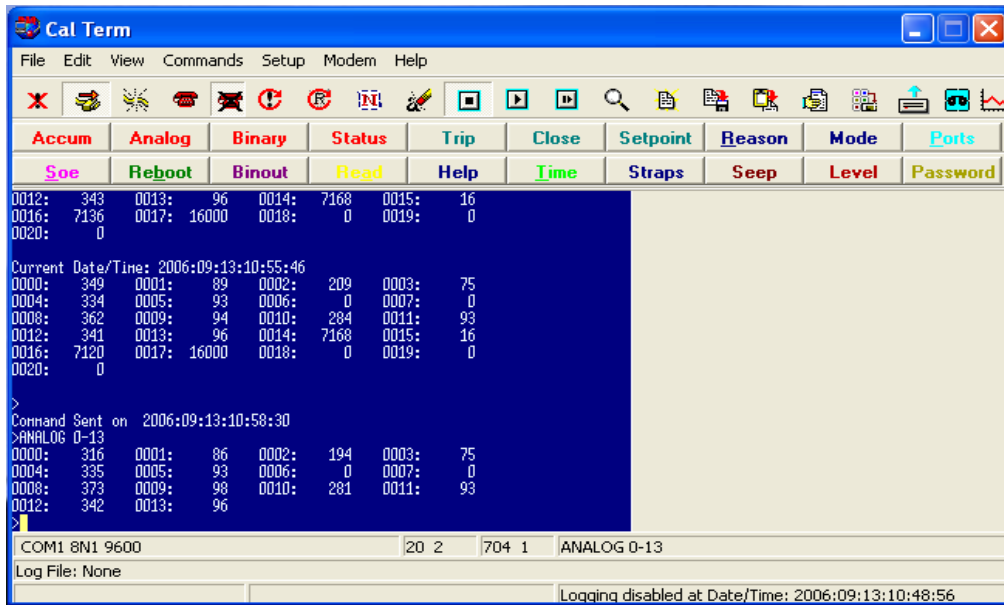


4.3.6 Monitoreo de las señales

Para el monitoreo de las señales en la unidad RTU, se hace uso de la herramienta CalTerm, el cual, como software de visualización, las señales las presenta empezando desde cero, por lo que no debe confundirse con las reglas de Modbus descritas en la tabla No. IV.

Para este ejemplo, después de seleccionar el botón "Analog" se le pide al programa visualizar las 14 variables análogas correspondientes a las señales de potencia activa y reactiva de las siete unidades de generación, es decir se le pide visualizar de la "0" a la "13". Esto se muestra en la figura siguiente.

Figura 34. Monitoreo de las señales en la RTU



Para una mejor visualización de las señales obtenidas, se elabora la siguiente tabla indicando la potencia real (MW), y reactiva (Mvar.) de cada unidad de generación. Se puede ver entonces, que la transmisión de señales desde el PLC a la unidad RTU es exitosa, siendo la transmisión de la RTU al sistema SCADA un procedimiento con la misma lógica, es decir configurar parámetros de protocolo, puertos, velocidad de transmisión etc.

Tabla XV. Señales en la unidad RTU

OEC No.	Dirección	Variable
1	0000	3.16 MW
	0001	0.86 MVAR
2	0002	1.94 MW
	0003	0.75 MVAR
3	0004	3.35 MW
	0005	0.93 MVAR
4	0006	f.l*
	0007	f.l*
5	0008	3.73 MW
	0009	0.98 MVAR
6	0010	2.81 MW
	0011	0.93 MVAR
7	0012	3.42 MW
	0013	0.96 MVAR

*Fuera de línea.

CONCLUSIONES

1. La confiabilidad en una planta industrial, la versatilidad, la reducción de costos y la integración de la información gerencial han sido potenciados usando dispositivos y equipos inteligentes que forman redes con distintos niveles funcionales desde los más próximos al proceso de control hasta los dedicados a las gestiones de planificación de recursos empresariales.
2. En el campo de las comunicaciones industriales, aún no está completamente definido un estándar que rija universalmente el proceso. Cada fabricante se ha empeñado en desarrollar su propia tecnología y la misma es aceptada por un grupo de fabricantes de equipo pero no así por la totalidad de los mismos. Por el momento, la pugna sigue y únicamente se vislumbra algunos puntos que indican estándares universales a futuro cercano.

RECOMENDACIONES

1. Como consecuencia de la constante evolución de los sistemas automatizados y temas relacionados, se aconseja al estudiante y al profesional de la ingeniería eléctrica mantenerse informado, ya que este es un área en constante cambio, para ello se sugiere consultar fuentes como: libros, boletines, paginas electrónicas etc.
2. Considerar la implementación del tema de “Transmisión de Datos” en uno de los cursos profesionales del pensa de estudios de la carrera de Ingeniería Eléctrica, puesto que las exigencias modernas van encaminadas en ir aumentando el grado de automatización.

BIBLIOGRAFÍA

1. Bolton W. **Programmable Logic Controllers**. 3ª. Edición.
Gran Bretaña: Editorial Newnes.
2. Ge Fanuc Automation. **Serial communications user's manual
GFK-0582D**
3. Fraser Roy E. **Process measurement and control**
Estados Unidos de América: Editorial Prentice may
4. Thompson Lawrence. **Industrial Data Communications: Fundamentals
and Applications**.
Estados Unidos de America: Instrument Society of America ISA.

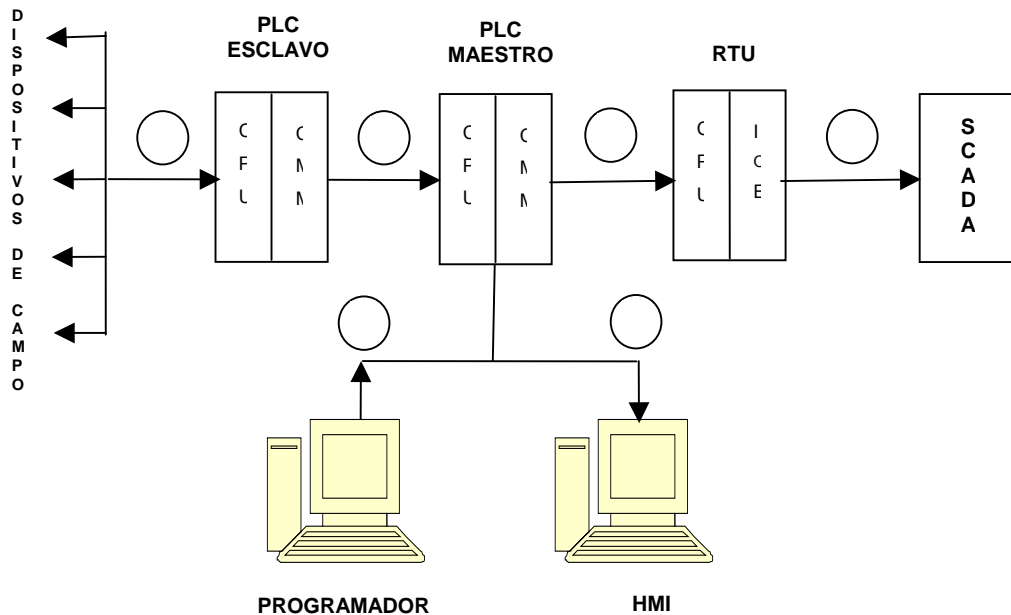
Referencia electrónica

4. www.industrialethernet.com junio de 2006
5. www.plc-tips.com mayo de 2006
6. www.mrplc.net mayo de 2006
7. www.profibus.org septiembre de 2006
8. www.fieldbusfoundation.orgT septiembre de 2006

APÉNDICE

En la planta Geotérmica de Zunil, como en toda industria moderna guatemalteca, la importancia del eficiente proceso de transmisión de datos es determinante en su operación y rendimiento ya que desde el nivel de campo, se dispone de instrumentos con capacidad y funciones de diagnóstico, control y mantenimiento. La presencia y variedad de los protocolos de comunicación en esta planta, se puede resumir mediante la siguiente grafica ilustrativa, desde el nivel de campo hasta el envío de información a un sistema SCADA:

Figura 35. Protocolos de comunicación en el automatismo de la planta Orzunil I.



1. Protocolo Hart
2. Protocolo CCM
3. Protocolo Modbus RTU
4. Protocolo DNP3.0
5. Protocolo SNP
6. Protocolo TCP/IP