



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

**DISEÑO DEL CONTROL REMOTO DE UNA RED DE VARIADORES DE FRECUENCIA A TRAVÉS DEL
PROTOCOLO DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ HMI**

Jaime Iván González Donis

Asesorado por el Ing. José Antonio de León Escobar

Guatemala, septiembre de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DEL CONTROL REMOTO DE UNA RED DE VARIADORES DE FRECUENCIA A TRAVÉS DEL
PROTOCOLO DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ HMI**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

JAIME IVÁN GONZÁLEZ DONIS

ASESORADO POR EL ING. JOSÉ ANTONIO DE LEÓN ESCOBAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, SEPTIEMBRE DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADORA	Inga. Ingrid Salomé Rodríguez de Loukota
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADOR	Ing. Marvin Marino Hernández Fernández
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

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

DISEÑO DEL CONTROL REMOTO DE UNA RED DE VARIADORES DE FRECUENCIA A TRAVÉS DEL PROTOCOLO DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ HMI

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 3 de octubre del 2012.

Jaime Iván González Donis

Guatemala, 15 de junio de 2013

Ing. Calos Guzmán
Coordinador del Área de Electrotecnia
Escuela de Ingeniería Mecánica-Eléctrica
Facultad de Ingeniería
Universidad de San Carlos de Guatemala


Estimado Ingeniero Guzmán

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **"DISEÑO DEL CONTROL REMOTO DE UNA RED DE VARIADORES DE FRECUENCIA A TRAVÉS DEL PROTOCOLO DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ HMI"**, desarrollado por el estudiante **Jaime Iván González Donis**, ya que considero que cumple con los requisitos establecidos.

Por lo tanto, el autor de este trabajo y yo como asesor, nos hacemos responsables del contenido y conclusiones del mismo.

Sin otro particular aprovecho la oportunidad para saludarlo.

Atentamente,



José Antonio de San Escobar
INGENIERO ELECTRONICO
COLEGIADO No. 8,281

Asesor



Ref. EIME 56.2013
Guatemala, 25 de JUNIO 2013.

Señor Director
Ing. Guillermo Antonio Puente Romero
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

**Me permito dar aprobación al trabajo de Graduación titulado:
DISEÑO DEL CONTROL REMOTO DE UNA RED DE
VARIADORES DE FRECUENCIA A TRAVÉS DEL PROTOCOLO
DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ
HMI, del estudiante Jaime Iván González Donis que cumple con los
requisitos establecidos para tal fin.**

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

Ing. Carlos Eduardo Guzmán Salazar
Coordinador Área Electrónica

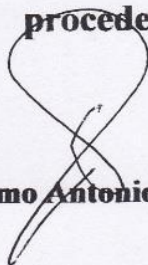


STO



REF. EIME 56. 2013.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; JAIME IVÁN GONZÁLEZ DONIS titulado: DISEÑO DEL CONTROL REMOTO DE UNA RED DE VARIADORES DE FRECUENCIA A TRAVÉS DEL PROTOCOLO DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ HMI, procede a la autorización del mismo.


Ing. Guillermo Antonio Puente Romero



GUATEMALA, 08 DE AGOSTO 2013.

Universidad de San Carlos
de Guatemala

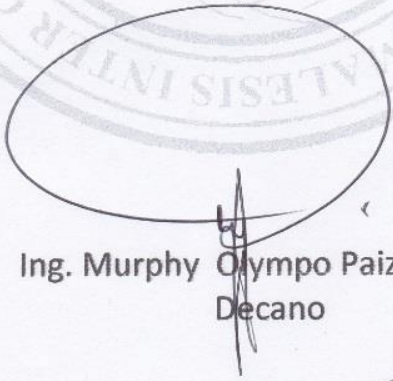


Facultad de Ingeniería
Decanato

DTG. 667.2013

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE CONTROL REMOTO DE UNA RED DE VARIADORES DE FRECUENCIA A TRAVÉS DEL PROTOCOLO DE COMUNICACIÓN MODBUS POR MEDIO DE UNA INTERFAZ HMI**, presentado por el estudiante universitario **Jaime Iván González Donis**, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. Murphy Olympo Paiz Recinos
Decano

Guatemala, 26 de septiembre de 2013

/gdech



ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. MARCO TEÓRICO.....	1
1.1. Control de procesos industriales.....	1
1.1.1. Esquemas del control industrial	6
1.1.2. Variables involucradas.....	8
1.1.3. Elementos de control	10
1.1.4. Objetivos del control	11
1.1.5. Necesidades y ventajas del control	12
1.1.6. Niveles de control industrial	13
1.1.7. Diseño de un sistema de control.....	17
1.2. Variadores de frecuencia.....	18
1.2.1. Repaso de conceptos electrotécnicos	18
1.2.2. Fundamentos tecnológicos de la regulación electrónica de velocidad de motores	21
1.2.3. Funcionamiento del motor asíncrono alimentado por convertidores de frecuencia.....	27
1.2.4. Procesos industriales con regulación de velocidad.....	30
1.3. Interfaz Hombre-Maquina	32

1.3.1.	Tipos de HMI	34
1.3.2.	Funciones de un software HMI	34
1.3.3.	Estructura general del software HMI	36
1.4.	Lenguaje de programación.....	39
1.4.1.	Descripción del lenguaje Ladder	43
1.4.2.	Principios de programación	45
2.	PROTOCOLO DE COMUNICACIÓN MODBUS	47
2.1.	Protocolo RS485	47
2.1.1.	Características y arquitectura de la conexión.....	48
2.1.2.	Sistemas de bus RS485.....	52
2.1.3.	Balanceo y desbalanceo de líneas.....	54
2.2.	Introducción al protocolo Modbus.....	55
2.2.1.	Transacciones sobre redes Modbus	56
2.2.2.	Transacciones en otro tipo de redes	57
2.2.3.	El ciclo de pregunta-respuesta	58
2.3.	Modos de transmisión serial.....	59
2.3.1.	Modo ASCII	60
2.3.2.	Modo RTU	61
2.4.	Trama de un mensaje Modbus.....	61
2.4.1.	Trama ASCII.....	62
2.4.2.	Fragmento RTU.....	63
2.4.3.	Cómo se maneja el campo de direccionamiento	64
2.4.4.	Cómo se maneja el campo de función	64
2.4.5.	Contenido del campo de datos.....	66
2.4.6.	Contenido del campo de verificación de error	67
2.4.7.	Como son transmitidos los caracteres de manera serial.....	67
2.5.	Métodos de verificación de error	69

2.5.1.	Chequeo de paridad	70
2.5.2.	Chequeo LRC	71
2.5.3.	Chequeo CRC	71
2.6.	Formatos de función Modbus	73
2.6.1.	Como son expresados los valores numéricos	73
2.6.2.	Las direcciones de datos en mensajes Modbus	73
2.6.3.	Contenido del campo en los mensajes Modbus	74
2.7.	Códigos de función admitidos por los controladores	76
2.7.1.	01 leer estado de bobina (Read Coil Status)	77
2.7.2.	02 leer estado de entrada (Read Input Status)	79
2.7.3.	03 leer registro de mantenimiento (Read Holding Registers)	81
2.7.4.	04 registro de lectura de entrada (Read Input Registers)	83
2.7.5.	05 forzar una bobina simple (Force Single Coil)	84
2.7.6.	06 registro simple de preset (Preset Single Register)	85
2.7.7.	07 leer estado de excepción (Read Exception Status)	87
3.	DISEÑO DEL SISTEMA DE MONITOREO A TRAVÉS DEL PROTOCOLO MODBUS	89
3.1.	Definición del proceso	89
3.1.1.	Descripción del diagrama de bloques del proceso	90
3.1.2.	VARIABLES controladas y no controladas	91
3.1.3.	Descripción de la interfaz	93
3.2.	Explicación de los diagramas de conexión	95
3.3.	Registros Modbus del variador	96

3.4.	Parámetros y comandos útiles para el desarrollo del proceso	97
3.5.	Medición de resultados	100
CONCLUSIONES.....		103
RECOMENDACIONES.....		105
BIBLIOGRAFÍA.....		107

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diagrama de bloques de un sistema básico.....	3
2.	Diagrama de bloques de un sistema de retroalimentación automático.....	7
3.	Tipos de respuesta de un sistema de retroalimentación automático.....	9
4.	Elementos de un sistema de retroalimentación automático	14
5.	Producción de un par en un motor de corriente continua.....	19
6.	Esquema base de un convertidor de frecuencia	22
7.	Rectificador trifásico de media onda	24
8.	Rectificador trifásico de onda completa. Puente de Graetz.....	25
9.	Curva de par/velocidad del arranque con convertidor de frecuencia....	29
10.	Curva constante de proporcionalidad entre tensión y frecuencia	29
11.	Curva de par constante.....	30
12.	Comportamiento de Cargas de Par Variable.....	31
13.	Comportamiento de Cargas de Potencia Constante	32
14.	Comportamiento de Cargas de Impacto.....	32
15.	Diagrama de un sistema HMI.	33
16.	Función de los archivos y programas para HMI.	38
17.	Estructuración de bloques de un sistema HMI.	39
18.	Ejemplo de diagrama escalera	40
19.	Programación con bloques funcionales.....	41
20.	Ejemplo de diagrama GRAFSET.....	42
21.	Programación de código de programación con texto estructurado	42
22.	Equivalencia entre lógica a relés y diagrama escalera.....	43

23.	Ejemplo de peldaños.	44
24.	Formato de programación.....	46
25.	Bus de 2 hilos RS485	53
26.	Bus de 4 hilos RS485	53
27.	Línea no balanceada	54
28.	Línea balanceada	55
29.	Ciclo maestro-esclavo consulta-respuesta.....	58
30.	Trama de mensaje ASCII.....	62
31.	Trama de mensaje RTU.....	64
32.	Orden de <i>bits</i> (ASCII).....	68
33.	Orden de <i>bits</i> (RTU).....	68
34.	Diagrama esquemático del sistema	90
35.	Diagrama de bloques del sistema	91
36.	Interfaz del variador de frecuencia.....	94
37.	Diagrama del cableado de la red.....	95
38.	Diagrama esquemático de conexión.....	96
39.	Programa Ladder para Mensaje Modbus.....	100

TABLAS

I.	Ejemplo de programación en lista de instrucciones.	40
II.	Características generales del protocolo RS-485.....	51
III.	Solicitud del Master con trama ASCII/RTU	75
IV.	Solicitud del Esclavo con trama ASCII/RTU.....	75
V.	Códigos de función soportados por controladores Modicon	76
VI.	Pregunta – Read Coil Status.....	77
VII.	Respuesta – Read Coil Status	78
VIII.	Pregunta – Read Input Status.....	79
IX.	Respuesta – Read Input Status	80

X.	Pregunta – Read Holding Registers	81
XI.	Respuesta – Read Holding Registers	82
XII.	Pregunta – Read Input Registers	83
XIII.	Respuesta – Read Holding Registers	84
XIV.	Pregunta – Force Single Coil.....	85
XV.	Respuesta – Read Holding Registers	86
XVI.	Pregunta – Preset Single Register	86
XVII.	Respuesta – Preset Single Register.....	87
XVIII.	Pregunta – ReadException Status	88
XIX.	Respuesta – Read Exception Status.....	88
XX.	Variables controladas – Registros lectura y escritura.....	92
XXI.	Variables no controladas – Registros de solo lectura.....	93
XXII.	Tiempo de transmisión para algunos datos calculados.....	101

LISTA DE SÍMBOLOS

Símbolo	Significado
-()-	Asignación de salida
- -	Contacto normalmente abierto
- / -	Contacto normalmente cerrado
AC	Corriente alterna
DC	Corriente directa
I/O	Entrada/Salida
f	Frecuencia
Fem	Fuente eléctrico-motriz
°	Grados
Hz	Hertz
OR	Operación Lógica que responde 1 si cualquiera de sus entradas es 1
XOR	Operación Lógica que responde 1 si las entradas son diferentes
AND	Operación Lógica que responde 1 si sus entradas son todas 1
RX	Recepción
TX	Transmisión
V	Voltaje

GLOSARIO

Actuador	Dispositivo mecánico o electromecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico.
ANSI	Instituto Nacional Estadounidense de Estándares.
Asíncrono	Se dice del proceso o del efecto que no ocurre en completa correspondencia temporal con otro proceso u otra causa.
CRC	Comprobación de redundancia cíclica.
DCE	Equipo terminal de circuito de datos.
Dieléctrico	Dicho de un material: Que es poco conductor y a través del cual se ejerce la inducción eléctrica.
<i>Display</i>	Palabra en inglés que significa visualizador y es el dispositivo de ciertos aparatos electrónicos, como los teléfonos y las calculadoras, destinado a la representación visual de información.
<i>Driver</i>	Palabra en inglés que significa controlador y es el dispositivo que ejerce la dirección y el dominio de un sistema.

DTE	Equipo terminal de datos.
Electrotecnia	Estudio de las aplicaciones técnicas de la electricidad.
Ethernet	Estándar de redes de área local para computadores con acceso al medio por contienda.
Fullduplex	Sistema capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.
Halfduplex	Método o protocolo de envío de información en forma bidireccional pero no simultaneo.
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos.
Impedancia	Relación entre la tensión alterna aplicada a un circuito y la intensidad de la corriente producida. Se mide en ohmios.
Ladder	También denominado lenguaje de contactos, es un lenguaje de programación grafico para autómatas.
LRC	Comprobación de redundancia longitudinal.

Microprocesador	Circuito integrado que incorpora en su interior una unidad central de proceso o CPU y todo un conjunto de elementos lógicos que permiten enlazar otros dispositivos como memorias y puertos de entrada y salida (I/O), formando un sistema completo para cumplir con una aplicación específica dentro del mundo real.
Modbus	Lenguaje de programación basado en la arquitectura maestro/esclavo.
Modulación	Técnica que se usa para transportar información sobre una onda portadora.
Parametrizar	Acción de cambiar los parámetros de un sistema en función de otro.
PID	Proporcional integral derivativo.
Protocolo	Uno o un conjunto de procedimientos destinados a estandarizar un comportamiento sistémico.
Rectificador	Aparato que transforma una corriente alterna en corriente continua.
Retroalimentación	Acción que sucede cuando un dispositivo se vuelve a alimentar dentro de un bucle.

RS-232	Estándar mundial que rige los parámetros de uno de los modos de comunicación serial para el intercambio de datos binarios entre un DTE y un DCE.
RS-485	Es un estándar de comunicaciones en bus de la capa física del Modelo OSI.
Sensor	Dispositivo que detecta una determinada acción externa, temperatura, presión, etc.
Setpoint	Es un punto fijo que se prestable.
Tag	Etiqueta que sirve para representar una variable.
Tiristor	Es un componente electrónico constituido por elementos semiconductores que utiliza realimentación interna para producir una conmutación.
Transistor	Dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador.
Trifásico	Sistema de tres corrientes eléctricas alternas iguales, procedentes del mismo generador, y desplazadas en el tiempo, cada una respecto de las otras dos, en un tercio de periodo.
VFD	Siglas en inglés (Variable Frequency Drive), Driver Variador de Frecuencia.

RESUMEN

En el presente trabajo de graduación se realiza un diseño para el control remoto de una red de variadores de frecuencia a través de uno de los protocolos de comunicación industrial más común; entre los dispositivos de mando como lo es Modbus y presentar los datos por medio de una Interfaz Hombre-Máquina.

En el capítulo I se presentan las bases teóricas sobre los conceptos que se deben de conocer previo al desarrollo del sistema, así como un análisis de las variables que interactúan en el sistema y que dan las nociones básicas para conocer su implementación. Se aborda el tema de los variadores de frecuencia para comprender su funcionamiento y su aplicación junto con los tipos de Interfaz Hombre-Máquina que permitan encontrar la mejor combinación para el sistema.

El capítulo II trata sobre el protocolo de comunicación Modbus, haciendo un estudio detallado de su operación y de la manera en que lleva a cabo la interacción entre un dispositivo configurado como maestro y otro dispositivo configurado como esclavo. Presenta los comandos necesarios para el control remoto de la red de variadores de frecuencia y su método de verificación de error.

En el capítulo III muestra el diseño del sistema de monitoreo a largas distancia de los VFD, los parámetros y comandos útiles para el desarrollo del proceso y termina con una medición de resultados.

OBJETIVOS

General

Controlar de manera remota una red de variadores de frecuencia a través del protocolo de comunicación Modbus por medio de una interfaz HMI.

Específicos

1. Examinar las variables involucradas con el control a largas distancias de una red de VFD, investigar la Interfaz Hombre-Máquina y la programación relacionada para este fin.
2. Reconocer los fundamentos y características importantes del protocolo de comunicación industrial Modbus, los parámetros y comandos que se utilizaran para el desarrollo del sistema.
3. Elaborar un sistema de monitoreo y control de un VFD usando un protocolo de comunicación que sea general, analizando el proceso y haciendo una medición de resultados.

INTRODUCCIÓN

Lo abarcado en este trabajo de graduación está relacionado con lograr el enlace por medio del protocolo Modbus entre una red de variadores que se puede diseñar para cualquier número de *drivers*, comenzado por una explicación detallada acerca del funcionamiento de cada dispositivo y se describirá cómo los mensajes se construyen y cómo se realizan las transacciones por medio del protocolo Modbus, abarcando en lo mayor posible a su explicación debido a la necesidad de aprender medios de comunicación industrial que sean soportados por el mayor número de dispositivos existentes en la industria.

Modbus es un protocolo que opera bajo el modo maestro-esclavo, es decir que se necesita un dispositivo que controle a otro y que para el sistema a desarrollar se traduce en el autómatas programable o PLC como maestro y el VFD como esclavo, debido a que el autómatas es quien contiene el procesador que permite enviar las señales al VFD. Este sistema puede ser usado para controlar una línea de producción o ya sea tener un mando seguro que permita obtener mayor flexibilidad y rendimiento para la producción en serie, debido a que por medio del PLC también puede manejar los datos que se obtienen desde la Interfaz Hombre-Máquina que puede ser una pantalla táctil o no.

Cabe destacar que además de lo anterior se necesita de la interfaz entre el PLC y el VFD que funcione como un traductor que entienda lo que el PLC le quiere decir al VFD y lo convierta en Modbus el cual sería el lenguaje del VFD y viceversa.

1. MARCO TEÓRICO

En la industria existen diversas aplicaciones en donde es importante controlar la velocidad de los motores de corriente alterna, esto se debe a las necesidades tales como las protecciones contra sobre intensidad, sobre temperatura, fallo contra desequilibrios, defectos a tierra, etc, además de necesitar de procesos de arranque y frenados suaves mediante rampas de aceleración y de frenado, lo que redundará en un aumento de la vida del motor y las instalaciones. Lo que a continuación se trata son los conceptos que permiten entender el proceso de automatización en el proyecto.

1.1. Control de procesos industriales

En un tiempo, los sistemas de procesos industriales se limitaban a un número de simples aplicaciones de control de motores y los dispositivos que se necesitan para lograr el control. Contactores magnéticos, los engranajes de interruptores eléctricos, tubos de gases, reóstatos y potenciómetros sirven generalmente como el núcleo del campo de control durante su infancia. Los avances en electrónica de estado sólido y el diseño de microcomponentes han traído importantes cambios tecnológicos en el campo.

La maquinaria automatizada, por ejemplo, ahora puede emplear pequeños equipos dedicados que están diseñados para controlar las funciones específicas del sistema y de esta manera alcanzar los niveles de control para la automatización de varios procesos industriales. Microprocesadores, la inteligencia artificial, controladores programables y dispositivos electrónicos todos se pueden combinar para controlar un único proceso. Una persona que

trabaja con este tipo de equipo debe tener una comprensión de todo el sistema para poder localizar los componentes defectuosos cuando se produce una avería.

El concepto de los sistemas servirá como la base para la comprensión de control de procesos industriales. En este enfoque, el sistema se divide en un número de bloques esenciales que son las bases para su funcionamiento. El papel desempeñado por cada bloque será significativo en el funcionamiento del sistema en general.

Después de la localización de cada bloque se ha establecido, el funcionamiento de componentes relacionados con cada bloque entonces será relevante. A través de este enfoque, una persona debe ser capaz de ver cómo las piezas de un sistema de proceso industrial encajan en un orden definido.

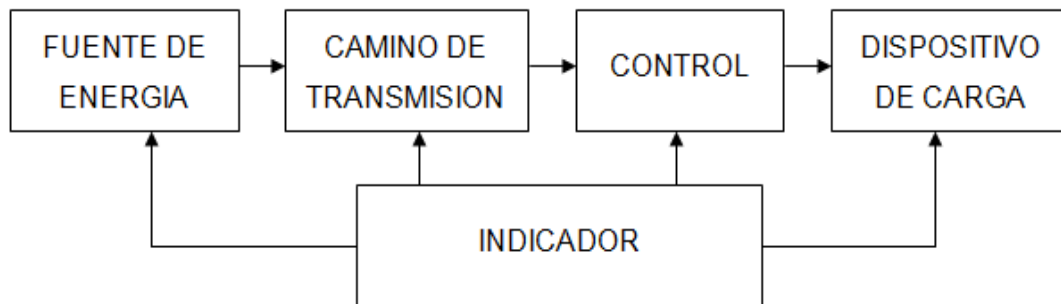
La palabra sistema se define como "una organización de partes que se conectan entre sí para formar el funcionamiento de una maquina o procedimiento operacional." Con respecto a esto, hay una amplia variedad de sistemas diferentes, utilizados en la industria de hoy. Un sistema de energía eléctrica, por ejemplo, que se necesita para producir energía eléctrica y distribuirlo a un lugar determinado.

Los sistemas hidráulicos y neumáticos se utilizan en la industria para llevar a cabo una o más operaciones, y para controlar otras funciones del sistema. Sistemas de temperatura se utilizan para controlar el calor, y para enfriar ciertas áreas durante las operaciones de fabricación. Los procesos industriales también son controlados por presión, caudal, nivel, analítica de adquisición de datos, y sistemas de transmisión. Obviamente, cada sistema industrial tiene una serie de características que lo distinguen de otros sistemas.

Sin embargo, hay un conjunto común de elementos que se encuentran en cada sistema. Estas piezas juegan el mismo papel básico en todos los sistemas. Los términos fuente de energía, camino de transmisión, control, carga, y el indicador describen la parte de un sistema. Un diagrama de bloques de estas partes se muestra en la Figura 1.

Cada elemento de un sistema básico tiene un papel específico en el funcionamiento global del sistema. Este papel se vuelve importante cuando un análisis detallado del sistema tiene lugar. Cientos y hasta miles de componentes discretos son a veces necesarias para lograr una función de bloque específico. Independientemente de la complejidad del sistema, cada bloque debe lograr su función básica para que el sistema sea operativo.

Figura 1. **Diagrama de bloques de un sistema básico**



Fuente: FARDO, Stephen. Industrial Process Control Systems p. 5.

El estar familiarizado con estas funciones y ser capaz de localizar dentro de un sistema completo es un gran paso en la comprensión de la operación general del sistema. La fuente de energía de un sistema industrial es responsable de convertir la energía en algo útil. El calor, la luz, el sonido,

química, nuclear, energía y mecánica son las fuentes primarias de energía. Una fuente de energía primaria va generalmente a través de una transformación de energía antes de que pueda ser utilizado en un sistema operativo.

El camino de transmisión de un sistema es algo simple en comparación con otras funciones del sistema. Esta parte del sistema proporciona un camino para la transferencia de energía. Se inicia con la fuente de energía y continúa a través del sistema para el dispositivo de carga. En algunos casos, esta trayectoria puede ser una sola línea de alimentación, conductor eléctrico, haz de luz, o el tubo conectado entre la fuente y la carga.

En otros sistemas, puede haber una línea de suministro entre la fuente y la carga, y una línea de retorno de la carga a la fuente. También puede haber un número de caminos alternativos o auxiliares dentro de un sistema completo. Estas trayectorias pueden ser conectadas en serie para un número de dispositivos de carga pequeños, o pueden conectarse en paralelo para muchos dispositivos independientes. La sección de control es, con mucho, la parte más compleja de todo el sistema. En su forma más simple, el control se logra cuando un sistema está encendido o apagado. El control de este tipo puede tener lugar en cualquier lugar entre la fuente y el dispositivo de carga.

El término de control se usa comúnmente para describir esta operación. Además de este tipo de control, un sistema también puede emplear algún tipo de control parcial. Control parcial suele causar algún tipo de un cambio operativo en el sistema que no sea el estado de encendido o apagado. Los cambios en la corriente eléctrica, la presión hidráulica, la temperatura, y flujo de aire son algunas de las alteraciones del sistema alcanzados por un control parcial. La carga de un sistema se refiere a una parte específica o el número de partes diseñadas para producir alguna forma de trabajo.

Se utiliza el término de trabajo cuando la energía pasa a través de una transformación o cambio. El calor, la luz, la acción química, el sonido y el movimiento mecánico son algunas de las formas más comunes de trabajo producidos por un dispositivo de carga. Generalmente, una gran parte de la energía producida por la fuente es consumida por el dispositivo de carga durante su funcionamiento.

La carga es típicamente, la parte más evidente de todo el sistema debido a la función de trabajo. El indicador de un sistema está diseñado principalmente para mostrar ciertas condiciones de funcionamiento en varios puntos a lo largo del sistema. El indicador es una parte opcional en algunos sistemas, mientras que en otros es esencial para la operación. Cuando un indicador es esencial, el funcionamiento del sistema depende por completo de las lecturas del indicador específicas.

El término indicador operacional se utiliza para describir esta aplicación. Los indicadores de prueba también son necesarios para determinar los valores de funcionamiento diferentes. En esta función, el indicador se une temporalmente al sistema para realizar mediciones. Luces de prueba, medidores de deformación en punteros, osciloscopios, grabadoras gráfica, instrumentos de visualización digital y medidores de presión son indicadores utilizados en esta capacidad. Un indicador normalmente se añade a la carga total del sistema.

Los sistemas de proceso de control se aplican a la sección grande de la industria que se ocupa de las cosas que tienen influencia directa en la fabricación de un producto terminado. Esto implica la manipulación de una variedad de variables de proceso con el fin de lograr un control automático. El número de variables en función del producto que se está fabricando. Por lo

general, más de una variable se modifica antes de un proceso de fabricación se ha completado. Algunas variables representativas que cambian con respecto al tiempo son la presión, temperatura, flujo, nivel, conductividad, viscosidad y peso.

Cuando una variable se altera, a menudo tiene un efecto profundo en otra variable. Los sistemas de control de procesos se utilizan en las ramas de la industria que se ocupan de la manipulación de variables. Estas variables manipuladas pueden influir en la calidad de un producto fabricado.

1.1.1. Esquemas del control industrial

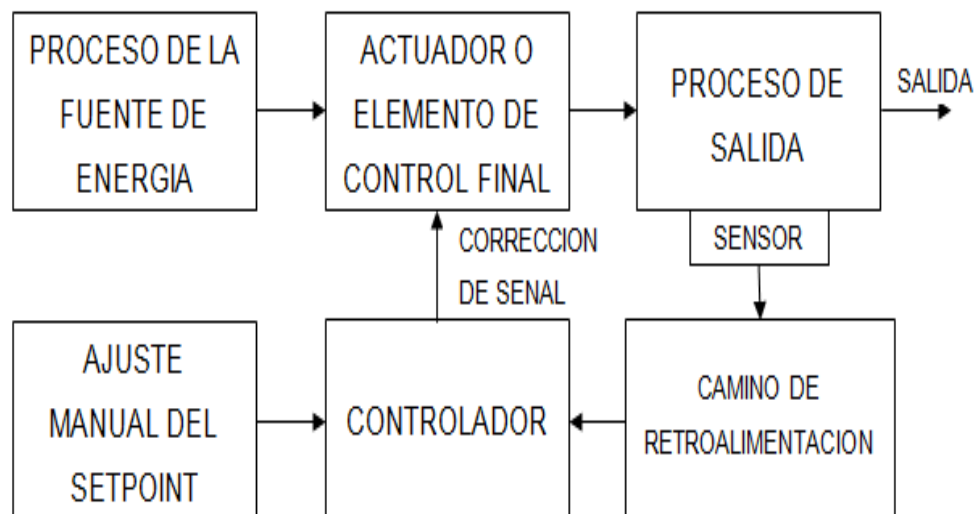
El esquema de control industrial generalmente usado es el de lazo cerrado que se refiere a un tipo de sistema que se autoregula. En este tipo de sistema, la salida real se mide y compara con un valor de salida predeterminado. Una señal de realimentación generada por el componente de detección de salida se utiliza para regular el elemento de control de modo que la salida se ajusta al valor deseado. El término retroalimentación se refiere a la dirección en la que se devuelve la señal de salida medida en el elemento de control.

En un sentido, la producción de este tipo de sistema sirve como la fuente de señal de entrada para el elemento de control de realimentación. El control de lazo cerrado se llama así porque de la vía de retorno creado por el bucle de realimentación va desde la salida a la entrada. Un diagrama de bloques de un sistema de retroalimentación de bucle cerrado simple o automático se muestra en la figura 2. La disposición de los componentes de base de este sistema es similar a la del sistema de bucle abierto. El sistema de bucle cerrado tiene un proceso de la fuente de energía, actuador o elemento final de control, la salida de procesos, el sensor, el circuito de realimentación y el controlador.

El circuito de realimentación conectado a la salida del proceso es una característica distintiva de este sistema. El origen de la señal de realimentación es un sensor conectado a la salida del proceso.

El circuito de realimentación devuelve una señal al controlador. El controlador es alimentado por un circuito sumador que compara la entrada del *setpoint* y las señales de realimentación. El proceso de la fuente de energía es responsable de establecer el valor nominal del sistema.

Figura 2. **Diagrama de bloques de un sistema de retroalimentación automático**



Fuente: FARDO, Stephen. Industrial Process Control Systems p. 29.

El valor del *setpoint* es ajustado por el operador de acuerdo con las necesidades del sistema. El valor real de la salida del proceso se determina por el sensor. Si la salida real es el mismo que el valor del *setpoint*, el controlador indica que el equilibrio del sistema y el actuador o elemento de control final se mantiene sin cambios.

Si la salida real se desvía del valor del *setpoint*, las señales aplicadas al controlador ya no están en equilibrio, y esto provoca una señal de corrección a desarrollar por el controlador. La señal de corrección es dirigida para alterar el actuador o elemento de control final. Este tipo de respuesta hace que el sistema sea auto-corrección. Estas partes se podrían aplicar a cualquiera de una amplia variedad de sistemas de procesos automáticos.

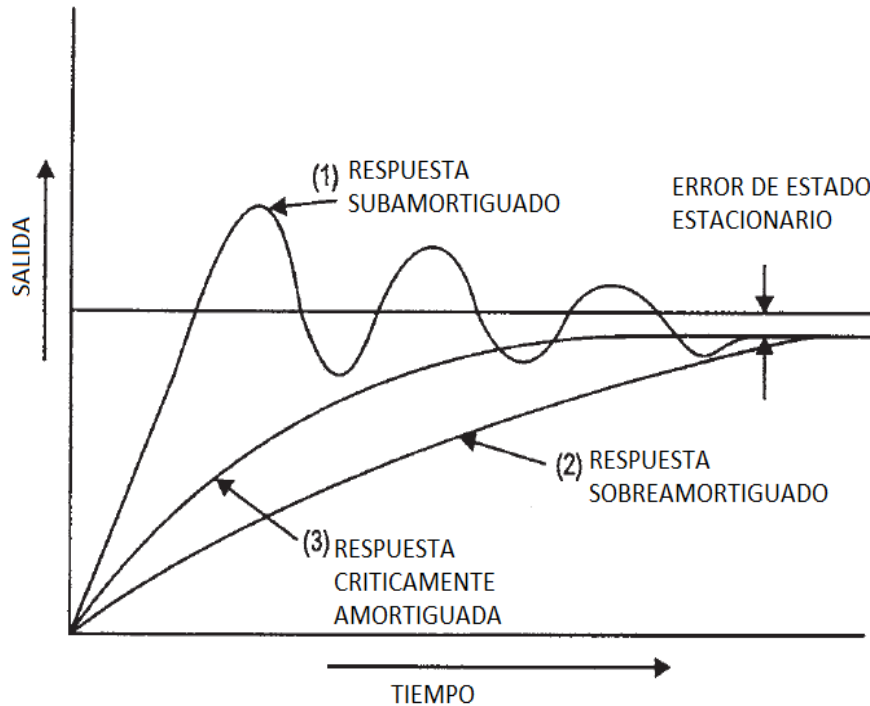
1.1.2. Variables involucradas

En control de bucle cerrado, un número de variables se utilizan para describir el funcionamiento de un sistema. Cuatro de estas variables se utilizan como criterios para evaluar el rendimiento del sistema: la respuesta transitoria, error de estado estacionario, la estabilidad y la sensibilidad. La respuesta transitoria es una condición que se utiliza para describir cómo la salida de un sistema de circuito cerrado responde. Esta condición se produce antes de que el sistema alcance un estado estable de operación. La salida de un sistema de lazo cerrado seguirá a cualquiera de los tres caminos representados por las curvas.

En la figura 3 la curva 1 muestra una respuesta subamortiguado. La salida de este tipo de sistema oscila de unos pocos ciclos por encima y por debajo del valor del *setpoint* antes de llegar a la nueva condición de estado estacionario. La curva 2 muestra una respuesta sobreamortiguado. Esto muestra el tiempo necesario para que la salida pueda llegar a una condición de estado estacionario a ser bastante largo.

La curva 3 es un ejemplo de una respuesta de forma crítica amortiguada. Esta condición se produce cuando la salida alcanza su condición de estado estable sin oscilaciones y en el menor tiempo posible.

Figura 3. **Tipos de respuesta de un sistema de retroalimentación automático**



Fuente: FARDO, Stephen. Industrial Process Control Systems p. 33.

La función de control de un sistema está diseñada, seleccionada o ajustada para un tipo particular de respuesta transitoria. Algunos procesos de fabricación no se ven afectados por la oscilación de la salida, mientras que otros no la pueden tolerar. Del mismo modo, algunos procesos no son afectados por ella tomando un largo período de alcanzar una condición de estado estacionario, mientras que otros tienen que lograr un control casi inmediatamente. La respuesta transitoria es una condición importante de la operación que debe ser tomada en cuenta cuando se selecciona un sistema particular para el control de un proceso de fabricación.

El error de estado estacionario es una variable de bucle cerrado que se refiere a un valor alcanzado por la salida después que un cambio en la entrada

se ha producido. Como puede verse en la figura 3, una ligera diferencia existente entre el valor de salida real y la salida deseada de un sistema de bucle cerrado después que la respuesta transitoria se produce. Esta diferencia de valor es el error de estado estacionario. Se muestra que el valor de salida de un sistema de bucle cerrado es compensado por una cierta cantidad, una consideración importante en el funcionamiento de un controlador.

La estabilidad es una condición de circuito cerrado que trata de la capacidad de un sistema para alcanzar el estado de equilibrio de control de una variable de proceso después de que se ha respondido a un cambio. La inestabilidad hace que el valor de salida cambie por encima y por debajo de la salida deseada. La respuesta sub-amortiguada en la figura 3 muestra un ejemplo de una mala estabilidad, lo que permite un sistema producir oscilaciones incontroladas. Si esta condición continúa durante un tiempo prolongado, puede ser perjudicial para algunos componentes del sistema.

La sensibilidad es una variable de bucle cerrado que se refiere a la relación entre el cambio porcentual de la producción para el cambio de porcentaje en la entrada del sistema. La entrada del sistema puede ser normal o afectada por alguna perturbación no deseada. Esta variable identifica la eficacia del sistema en la corrección de los cambios de valor en la entrada.

1.1.3. Elementos de control

Un elemento de control es una parte del proceso del sistema de control que ejerce influencia directa sobre la variable controlada para llevarlo a la posición del *setpoint*. El término de control final se utiliza indistintamente con el elemento de control, entre los elementos de control están los sensores, los controladores y los actuadores.

El sensor es una pieza de equipo que se utiliza para medir las variables del sistema. Los sensores son transductores que normalmente cambian la energía de una forma a otra cosa. Los sensores sirven como la fuente de señal en sistemas de control automático. Un controlador es una pieza de hardware de equipo que emplea neumática, electrónica, y / o energía mecánica para realizar una operación de control del sistema.

Estas unidades están diseñadas para mantener una variable de proceso en un valor predeterminado mediante la comparación de su valor existente a la de un valor deseado del sistema y los actuadores son mecanismos que físicamente, se mueven, o controlan el funcionamiento de un componente del sistema.

1.1.4. Objetivos del control

Refinación, combinación, manipulación, y de otro modo la manipulación de fluidos para producir rentablemente productos finales puede ser un proceso preciso, exigente, y potencialmente peligrosos. Pequeños cambios en un proceso puede tener un gran impacto en el resultado final. Variaciones en las proporciones, temperatura, flujo, turbulencia y muchos otros factores deben ser cuidadosamente controlados y consistentes para producir el producto final deseado con un mínimo de materias primas y energía.

La tecnología de control de procesos es la herramienta que permite a los fabricantes para mantener sus operaciones se ejecutan dentro de los límites especificados y establecer límites más precisos para maximizar la rentabilidad, asegurar la calidad y la seguridad que hasta este punto integra la parte del sistema. En base a todo esto se puede mencionar que los objetivos del control en los sistemas industriales son:

- Seguridad y estabilidad
- El ajuste a la cantidad y calidad demandada
- El cumplimiento de normativa medioambiental
- Funcionamiento cercano al óptimo es decir menor coste y máximo beneficio.

Todo esto se puede resumir de la siguiente manera: mantener el proceso en condiciones de funcionamiento deseadas de manera segura y eficiente al tiempo que satisface los requisitos de calidad del medio ambiente y del producto.

1.1.5. Necesidades y ventajas del control

En un corto espacio de tiempo, la industria ha hecho una transición en la industria: desde las técnicas de producción que se lograron en gran parte a través de las operaciones manuales hasta sofisticados procedimientos automáticos que requieren poco esfuerzo humano. A través de esta transición se han producido algunos cambios drásticos que se han traducido en una mejor calidad de los productos y la producción más económica. La transición a la automatización inicial fue concebida como un medio para mejorar la producción a través de los costos laborales reducidos.

En muchas partes de la industria, este objetivo en particular se ha realizado, en cierta medida. En otras fases de la industria, los ahorros de una nómina reducida han sido compensados por el aumento de los costos de adquisición de equipos. Pero con esto viene la necesidad de mejorar el personal técnico capacitado. La producción automatizada ha dado lugar a una serie de características que son de mucha más importancia que la ventaja inicial de una nómina más pequeña.

Estas características incluyen la reducción de los residuos, las tolerancias mejoradas, mejor consistencia del producto, la comodidad producción y la mejora de las técnicas de desarrollo de producto. Todas estas características han dado lugar a cambios muy decididos en toda la fabricación industrial y han marcado las necesidades de procesos industriales y las ventajas del mismo.

Con la automatización de los procesos industriales se tienen las ventajas de controlar las variables de salida a voluntad del proceso, controlar con modelos inexactos y perturbaciones, mejorar el control mediante utilización de variables auxiliares, controlar varias variables interaccionadas y controlar con objetivos más complejos los procesos de automatización industrial.

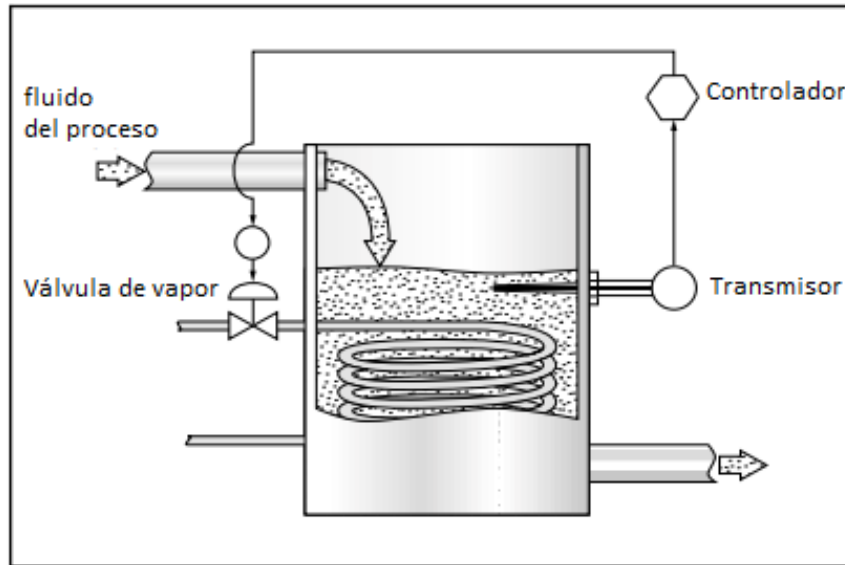
1.1.6. Niveles de control industrial

Entre los niveles de control industrial se tienen los siguientes:

- Control regulatorio básico
- Control regulatorio avanzado
- Control multivariable

En el control regulatorio básico se tiene el bucle de realimentación el cual mide una variable de proceso y envía la medición a un controlador para la comparación de punto de ajuste. Si la variable de proceso no está en el punto establecido, la acción de control se toma para devolver la variable de proceso al punto establecido. Los niveles de control industrial dan un panorama amplio sobre el control que se tienen de las variables que se involucran para el desarrollo del sistema que permitirá controlar a distancias superiores a las distancias estándar. El control regulatorio básico y el control regulatorio avanzado tendrá mayor relevancia con forme se estudian los variables

Figura 4. **Elementos de un sistema de retroalimentación automático**



Fuente: <http://www.pacontrol.com/download/Process%20Control%20Fundamentals.pdf>.

Consulta: 17 de febrero 2013.

En un control regulatorio avanzado se tiene el control anticipativo, el control anticipativo con retroalimentación, control en cascada, control por lotes, control de la relación, control selectivo y el control difuso. El control anticipativo es un sistema de control que prevé perturbaciones en la carga, y los controla antes de que puedan afectar a la variable de proceso. Para el trabajo del control anticipativo, el usuario debe tener una comprensión matemática de cómo las variables manipuladas afectará las variables del proceso.

En el control anticipativo con retroalimentación debido a la dificultad de la contabilidad para cada perturbación de carga posible en un sistema de control anticipativo, los sistemas de control anticipativos se combinan a menudo con los sistemas de retroalimentación. Controladores con funciones de suma se utilizan en estos sistemas combinados al total de la entrada de datos con el bucle de

control anticipativo y el bucle de retroalimentación, y envían una señal unificada para el elemento de control final.

El control en cascada es un sistema de control en el que se establece un secundario (esclavo) bucle de control para controlar una variable que es una fuente importante de perturbación de carga para otro bucle de control primario (maestro). El controlador del lazo primario determina el valor del *setpoint* del controlador de suma en el bucle secundario. El control por lotes son aquellos procesos que se han tomado de principio a fin en lotes. Por ejemplo, la mezcla de los ingredientes para una bebida de jugo es a menudo un proceso por lotes. Típicamente, una cantidad limitada de un solo sabor (p.ej. bebida, bebida de naranja o manzana) se mezcla a la vez.

Por estas razones, no es práctico tener un proceso de funcionamiento continuo. Procesos por lotes suelen implicar conseguir la proporción correcta de los ingredientes en el lote. Nivel, caudal, presión, temperatura, y a menudo las mediciones de masa se utilizan en las diversas etapas de los procesos por lotes. Imagine un proceso en el cual debe ser un ácido diluido con agua en la proporción de dos partes de agua a una parte de ácido. Si el tanque tiene una fuente de ácido en un lado de un recipiente de mezcla y un suministro de agua por el otro, un sistema de control podría ser desarrollado para controlar la relación de ácido a agua, a pesar de que el suministro de agua en sí misma no puede ser controlado.

Este tipo de sistema de control es el control de la relación. El control de la relación se utiliza en muchas aplicaciones, e implica un controlador de que recibe la entrada de un dispositivo de medición de flujo en la no regulada de flujo. El controlador realiza un cálculo de la relación, y señala el valor de consigna correspondiente a otro controlador que establece el flujo del segundo

fluido de manera que la proporción apropiada del segundo fluido puede ser añadida. El control selectivo se refiere a un sistema de control en el que la más importante de las dos variables se mantendrá.

Por ejemplo, en un sistema de control de la caldera, si el flujo de combustible supera el flujo de aire, a continuación, el combustible no quemado se puede acumular en la caldera y causar una explosión. El control selectivo se utiliza para permitir una mezcla de aire-rico, pero nunca una mezcla rica en combustible. El control selectivo se utiliza con mayor frecuencia cuando el equipo se debe proteger o mantener la seguridad, aún a costa de no mantener un punto de ajuste variable del proceso óptimo. El control difuso es una forma de control adaptativo en el que el controlador usa la lógica difusa para tomar decisiones sobre cómo ajustar el proceso.

El control difuso o lógica difusa es una forma de la lógica del ordenador donde si algo es o no se incluye en un conjunto se basa en una escala de clasificación en la que múltiples factores se tienen en cuenta y clasificado por el ordenador. La idea esencial del control difuso es crear una especie de inteligencia artificial que representará numerosas variables, formular una teoría de cómo hacer mejoras, ajustar el proceso y aprender de los resultados.

El control multivariable o de bucles multivariable son los bucles de control en el que un controlador primario controla una variable de proceso mediante el envío de señales a un controlador de un bucle diferente que afecta a la variable de proceso del lazo primario. Por ejemplo, la variable de proceso primario puede ser la temperatura del fluido en un tanque que se calienta con una cámara de vapor (una cámara de vapor a presión que rodea el tanque). Para controlar la variable principal (temperatura), el primario (maestro) señala el controlador secundario (esclavo) controlador que controla la presión de vapor.

El controlador primario manipulará el punto del *setpoint* del controlador secundario para mantener la temperatura de consigna de la variable de proceso primario. Cuando se sintoniza un bucle de control, es importante tener en cuenta la presencia de bucles de varias variables. El procedimiento estándar es sintonizar el circuito secundario antes de sintonizar el lazo primario debido a ajustes en el impacto del lazo secundario del lazo primario. El ajuste del lazo primario no afectará la afinación del lazo secundario.

1.1.7. Diseño de un sistema de control

En el diseño de un sistema de control se deben de llevar varias fases, empezando con la definición de los objetivos de control, por ejemplo asegurar la estabilidad o conseguir la operación óptima. Luego se deben de identificar las variables medibles y las manipulables, como mínimo se deben medir las variables objetivo de control (a veces se recurre a variables auxiliares a partir de las que se infiere su valor), a veces se miden variables de perturbación y variables auxiliares para implementar estrategias avanzadas de control. La elección de las variables manipulables es de vital importancia, ya que la calidad de control alcanzable depende en gran medida de su elección.

Cuando se haya cumplido este procedimiento, se selecciona la configuración del sistema de control y en esto se deciden estrategias de control como bucles simples de regulación, bucles en cascada, esquemas anticipativos, emparejamiento entre variables a controlar y variables manipulables. Y finalmente la especificación de la instrumentación de monitorización y control como instrumentos de medida (sensores y transmisores), controladores y elementos finales de control. Los elementos finales de control interactúan directamente con los dispositivos exteriores que intervienen en el proceso de automatización y causan interés para todo el proceso.

1.2. Variadores de frecuencia

Sin duda alguna los accionamientos a base de motores eléctricos son los más numerosos de la mayoría de las aplicaciones, y dentro de ellos los basados en motores de corriente continua han gozado de una total hegemonía en el campo industrial durante décadas. Sin embargo los motores con menor nivel de exigencias en el mantenimiento son los motores asíncronos de jaula de ardilla, debido a que carecen de colector, tienen una relación peso-potencia mucho menor que los de continua, y por tanto un coste significativamente más bajo.

Por estas razones, dada su capacidad de soportar sobrecargas y su elevado rendimiento, es el motor más atractivo para la industria. Desde hace varios años, el elevado desarrollo de la electrónica de potencia y los microprocesadores ha permitido variar la velocidad de estos motores, de una forma rápida, robusta y fiable, mediante los reguladores electrónicos de velocidad. La elección de la instalación de un convertidor de frecuencia como método de ahorro energético supone:

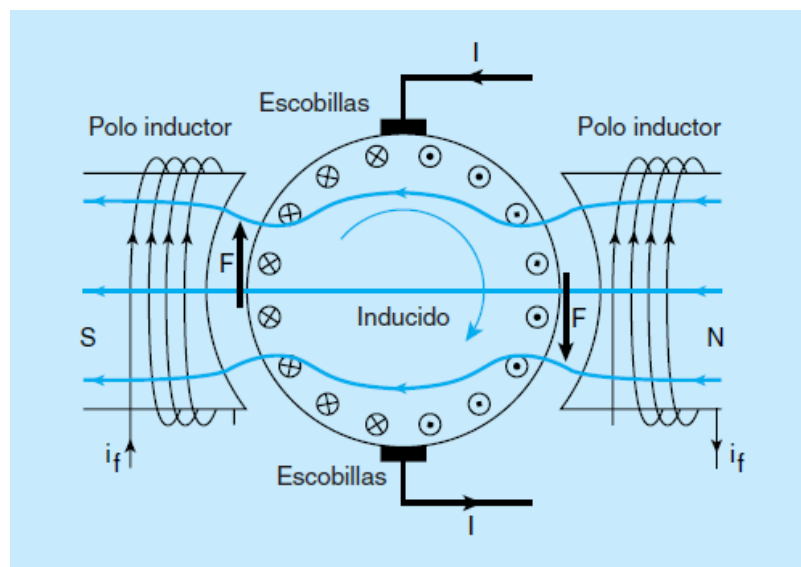
- Reducción del consumo
- Mejor control operativo, mejorando la rentabilidad y la productividad
- Minimizan las pérdidas en las instalaciones
- Ahorro en mantenimiento (el motor trabaja siempre en las condiciones óptimas de funcionamiento).

1.2.1. Repaso de conceptos electrotécnicos

Estos son conceptos básicos de los motores asíncronos de jaula de ardilla, sobre todo a los puntos de arranque, velocidad y frenado. Igualmente, a

continuación se exponen, como repaso, algunos conceptos sobre máquinas asíncronas (generadores y motores). Las máquinas eléctricas sirven para transformar la energía mecánica en eléctrica (generadores) o, inversamente, para transformar la energía eléctrica en mecánica (motores); es decir que la máquinas eléctricas son reversibles y pueden trabajar como generador o como motor. Estas máquinas asíncronas se basan en el principio de la acción de un campo magnético giratorio sobre un arrollamiento en cortocircuito.

Figura 5. **Producción de un par en un motor de corriente continúa**



Fuente: Schneider Electric, Manual Electrotécnico, p 92.

El sistema magnético de una máquina asíncrona consta de 2 núcleos: el núcleo exterior fijo que tiene la forma de un cilindro hueco y el núcleo cilíndrico interior giratorio. La parte fija de la máquina se llama estator y la parte giratoria rotor. En las ranuras ubicadas en el lado interior del estator se coloca un arrollamiento trifásico, constituido por tres arrollamientos iguales (uno por cada fase) desplazado 120° entre sí (para un motor con un solo par de polos).

Por lo visto en campos giratorios, los efectos simultáneos de las tres corrientes de una red trifásica originan un campo constante que gira a velocidad síncrona (n_s). Si en el interior del estator se coloca una espira en corto circuito (rotor) que pueda girar alrededor de un eje, se tiene que:

- Al conectar el estator a la red se origina un campo giratorio que originará una fem inducida.
- Esta fem inducida, a su vez hace circular una corriente en la espira en cortocircuito (rotor), y por la acción del campo magnético creará cuplas distribuidas a lo largo de la espira haciendo que ésta trate de seguir al campo giratorio.
- Es obvio que la espira nunca podrá alcanzar al campo giratorio debido a la cupla resistente, que está compuesta en primer término por las pérdidas por rozamiento en los cojinetes, y en segundo término por la carga que mueve el eje del motor.
- Es decir que siempre habrá una diferencia entre la velocidad " n_s " del campo giratorio y la del rotor " n " (o la del eje del motor).

Por lo tanto, se llamará velocidad relativa a la diferencia entre n_s y n , que para el caso de motores será positiva y para el caso de generadores será negativa: $n_{rel} = n_s - n$.

La magnitud característica que determina las condiciones de trabajo de la máquina asíncrona es el resbalamiento, el resbalamiento es positivo cuando la máquina trabaja como motor y negativo cuando ésta trabaja como generador y de esta forma generar el proceso.

1.2.2. Fundamentos tecnológicos de la regulación electrónica de velocidad de motores

Un regulador electrónico de velocidad está formado por circuitos que incorporan transistores de potencia como el IGBT (transistor bipolar de puerta aislada) o tiristores, siendo el principio básico de funcionamiento transformar la energía eléctrica de frecuencia industrial en energía eléctrica de frecuencia variable. Esta variación de frecuencia se consigue mediante dos etapas en serie.

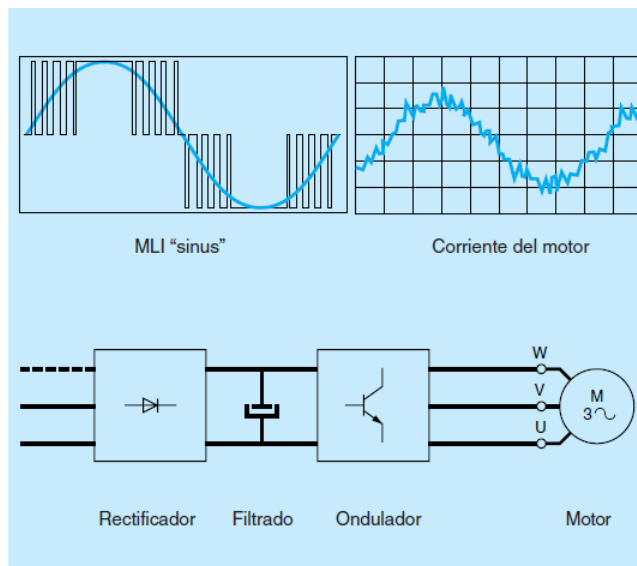
Una etapa rectificadora que transforma la corriente alterna en continua, con toda la potencia en el llamado circuito intermedio y otra inversora que transforma la corriente continua en alterna, con una frecuencia y una tensión regulables, que dependerán de los valores de consigna. A esta segunda etapa también se le suele llamar ondulator. Todo el conjunto del convertidor de frecuencia recibe el nombre de inversor.

El modo de trabajo puede ser manual o automático, según las necesidades del proceso, dada la enorme flexibilidad que ofrecen los reguladores de velocidad, permitiendo hallar soluciones para obtener puntos de trabajo óptimos en todo tipo de procesos, pudiendo ser manejados por ordenador, PLC, señales digitales o de forma manual.

La mayoría de las marcas incluyen dentro del propio convertidor protecciones para el motor, tales como protecciones contra sobre-intensidad, sobre-temperatura, fallo contra desequilibrios, defectos a tierra, etc., además de ofrecer procesos de arranque y frenados suaves mediante rampas de aceleración y de frenado, lo que redundará en un aumento de la vida del motor y las instalaciones. Como debe saberse, el uso de convertidores de frecuencia

añade un enorme potencial para el ahorro de energía disminuyendo la velocidad del motor en muchas aplicaciones.

Figura 6. **Esquema base de un convertidor de frecuencia**



Fuente: Schneider Electric, Manual Electrotécnico, p 85.

Además aportan los siguientes beneficios:

- Mejora el proceso de control y por lo tanto la calidad del producto
- Se puede programar un arranque suave, parada y freno
- Amplio rango de velocidad, par y potencia
- Bucles de velocidad
- Puede controlar varios motores
- Factor de potencia unitario
- Respuesta dinámica comparable con los drivers de DC
- Capacidad de *by-pass* ante fallos del variador

- Protección integrada del motor
- Marcha paso a paso (comando JOG)

Con respecto a la velocidad los convertidores suelen permitir dos tipos de control.

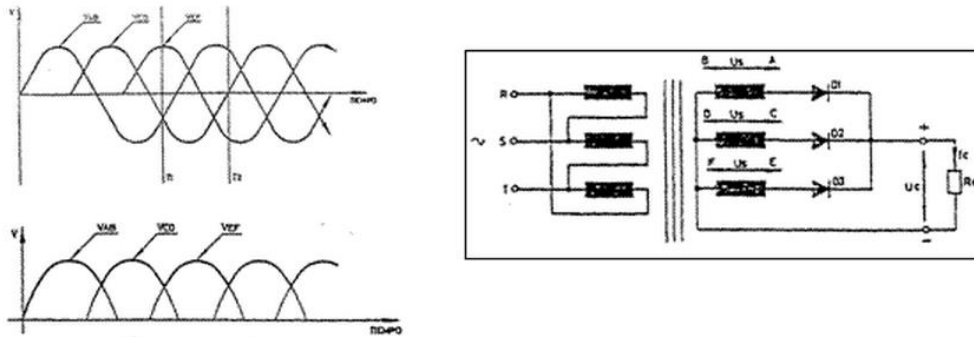
Control manual de velocidad. La velocidad puede ser establecida o modificada manualmente (display de operador). Posibilidad de variación en el sentido de giro.

Control automático de velocidad. Utilizando realimentación se puede ajustar la velocidad automáticamente. Esta solución es la ideal para su instalación en aplicaciones en las que la velocidad demandada varía de forma continua.

Entre los rectificadores para los convertidores de frecuencia, se tienen los rectificadores no controlados de tensión fija y los rectificadores controlados de tensión variable. En los rectificadores no controlados de tensión fija el fenómeno de la rectificación se da porque los diodos van conmutando cíclicamente al circuito de cc sobre las fases de ca. Es la tensión de esta red la que va forzando el paso a conducción o bloqueo de los diodos, a esta conmutación se le llama forzada. Si sólo se rectifican las semiondas positivas de la tensión alterna se tiene un montaje de media onda y si se rectifican ambas semiondas, se tiene un montaje de onda completa.

En los montajes de media onda la tensión no es continua pura, ya que exhibe cierto grado de rizado u oscilación en torno a su valor medio. Los diodos que conducen en cada momento son aquellos en los que la tensión de la fase en la que van conectados supera a la de las otras dos.

Figura 7. Rectificador trifásico de media onda



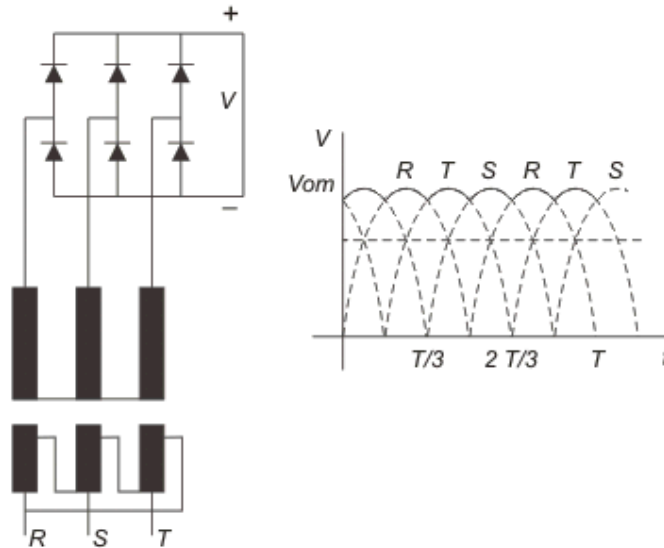
Fuente: http://www.oposinet.com/tecnologia/temas/otema58_1.htm. Consulta: 19 de febrero de 2013.

El rectificador trifásico de onda completa o puente de Graetz, está formado por seis diodos y tiene la ventaja del menor rizado. Este puente es de los más empleados en las aplicaciones industriales de potencia.

En los rectificadores controlados de tensión variable Los esquemas son similares a los anteriores, cambiando los diodos por semiconductores controlables, que normalmente son tiristores, así se logrará que la tensión de salida sea variable y con prestaciones más interesantes debido a esta circunstancia. El puente trifásico de onda completa o de Graetz es el más empleado desde el punto de vista industrial, ya que tiene las siguientes ventajas:

- Se carga simétricamente la línea trifásica
- Se absorben menos armónicos de intensidad en la línea trifásica
- La tensión continua es de rizado con menor amplitud y por tanto los filtros para alisado son menores.
- Las prestaciones dinámicas son mayores, ya que con seis pulsos se puede variar el ángulo de encendido seis veces por periodo.

Figura 8. **Rectificador trifásico de onda completa. Puente de Graetz**



Fuente: <http://www.textoscientificos.com/fisica/alta-tension/generacion-tensiones-continuas>

Consulta: 19 de febrero de 2013.

En un circuito intermedio la etapa central es el denominado circuito intermedio de continua y que puede funcionar como fuente de tensión o intensidad para la etapa final del ondulador, según la disposición que se adopte. A veces al ondulador se le llama inversor tal como aparece en la figura, aunque es más correcto llamar inversor a todo el conjunto (rectificador, circuito intermedio y ondulador). La función del circuito intermedio es alimentar la tercera etapa, es decir al ondulador, y esto puede hacerlo funcionando como fuente de tensión, en cuyo caso se colocaría un condensador electrostático entre los terminales (+) y (-) para mantener constante la tensión y daría lugar a un inversor con circuito intermedio de tensión.

Cuando el circuito intermedio funciona como fuente de intensidad para el ondulador, se pone una inductancia en serie con una de sus ramas, su función es mantener constante la intensidad, y se estaría hablando de un inversor con circuito intermedio de intensidad. Según la configuración que se adopte las

características del inversor son distintas y condiciona cuestiones tales como: armónicos, resistencia de frenado, gama de potencias, accionamiento para un solo motor o varios a la vez, etc.

El ondulator es un conmutador electrónico que comunica alternativamente la tensión o intensidad continua del circuito intermedio sobre las fases del motor de c.a. conectado a su salida. La disposición más común es el puente trifásico de Graetz y está formado por semiconductores controlables que pueden ser tiristores, tiristores desconectables por puerta (GTO), transistores de potencia, IGBT (transistor bipolar de puerta aislada o MOSFET (transistor de efecto campo de óxido metálico). De los anteriores el que más se está utilizando para motores industriales de BT es el IGBT.

En función de la mayor o menor perfección del sistema de conmutación se logrará que las ondas de tensión a la salida hagan que las corrientes absorbidas se acerquen más o menos al sistema trifásico senoidal. Hay distintas formas de regular la tensión de salida del inversor como son:

- Variar el valor de la tensión en el circuito intermedio
- Variar el ancho de la zona de conducción de cada semionda de salida
- Variar la tensión de salida en función de la proporción entre los tiempos de conexión y desconexión de los semiconductores de potencia mediante la técnica de regulación PWM (iniciales de Modulación del Ancho de Pulso, en inglés). Además de regular la salida, este método tiene la ventaja de generar una onda de tensión de salida que mejora notablemente la onda de intensidad absorbida por el motor, lo cual hace que el motor funcione de forma semejante a si estuviera alimentado por tensiones senoidal de la red. Con ello se logra la grandísima ventaja de emplear motores normalizados de fabricación en serie sin la necesidad

de fabricar motores específicos para poder ser regulados por convertidores.

1.2.3. Funcionamiento del motor asíncrono alimentado por convertidores de frecuencia

Los inversores con circuito intermedio de tensión son los más usados en aplicaciones prácticas, siendo su campo predominante el de las pequeñas y medianas potencias. Un inversor se elige en función de parámetros tales como:

- Accionar a un solo motor o varios
- Banda necesaria de regulación y su precisión
- Consecuencias sobre la red eléctrica del convertidor adoptado
- ¿Tiene sentido económico prever un retorno de energía?
- Velocidad de respuesta para adaptarse a los cambios de consigna

Para aprovechar al máximo el motor hay que controlarlo de modo que el flujo se aproxime lo más posible al nominal para el cual ha sido diseñado. Cuando el motor está regulado con flujo constante e igual al nominal presenta unas curvas características como la que se muestra en la figura 9.

Una forma de lograr que el flujo sea constante de manera aproximada, es hacer que la tensión y la frecuencia varíen de forma proporcional. Sin embargo esto es sólo aproximado, y a medida que las frecuencias van bajando los flujos disminuyen también por lo que el par para bajas frecuencias disminuye de forma importante.

Para lograr el funcionamiento con flujo constante es preciso que a bajas frecuencias la tensión sea más elevada que lo que dicta la ley sencilla de la

proporcionalidad. Cuando la regulación necesaria para modificar la velocidad supera la frecuencia nominal (50 Hz), el flujo ha de disminuir, ya que la tensión no debe ser elevada para no sobrepasar las posibilidades dieléctricas del bobinado del motor.

En este caso las curvas de par para frecuencias elevadas decrecen, por lo que habrá que verificar que los menores pares disponibles cumplen los requisitos de la máquina accionada a alta velocidad.

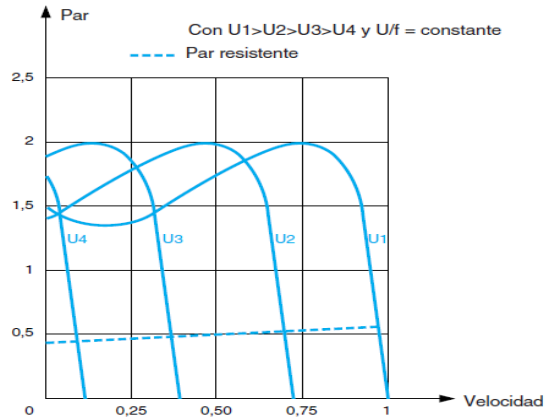
En general en aquellos inversores con circuito intermedio de tensión, para el control del par electromagnético del accionamiento se emplean los siguientes métodos:

- Regular la tensión del estator en función de la frecuencia. (Control V/f)
- Regulación mediante la descomposición vectorial de la intensidad del estator sobre unos ejes orientados con el flujo magnético. (Control vectorial).

Por medio del método de control de V/f la tensión de alimentación evoluciona proporcionalmente a la frecuencia. Cuando V/f es constante el motor funciona de forma aproximada con flujo constante en los regímenes permanentes.

Este tipo de control es más fácil de llevar a la práctica en un convertidor y se suele emplear cuando los requisitos de regulación son de baja velocidad. Como se ha mencionado anteriormente la proporcionalidad V/f desaparece en las bajas frecuencias, además la característica de la curva de par depende también de la frecuencia del rotor y de su temperatura, por lo que el dispositivo de control del convertidor ha de incluir las correspondientes correcciones.

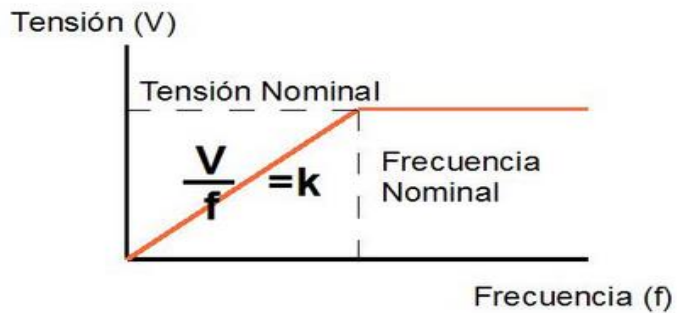
Figura 9. **Curva de par/velocidad del arranque con convertidor de frecuencia**



Fuente: Schneider Electric, Manual Electrotécnico, p 85.

En los convertidores con este tipo de control, una de las parametrizaciones más importante es la selección o ajuste de la curva V/f . Algunos convertidores traen varias curvas ya ajustadas en su programación.

Figura 10. **Curva constante de proporcionalidad entre tensión y frecuencia**



Fuente: <http://www.infopl.net/blog4/2010/02/27/el-convertidor-de-frecuencia-ii-control-escalar/>

Consulta: 19 de febrero de 2013.

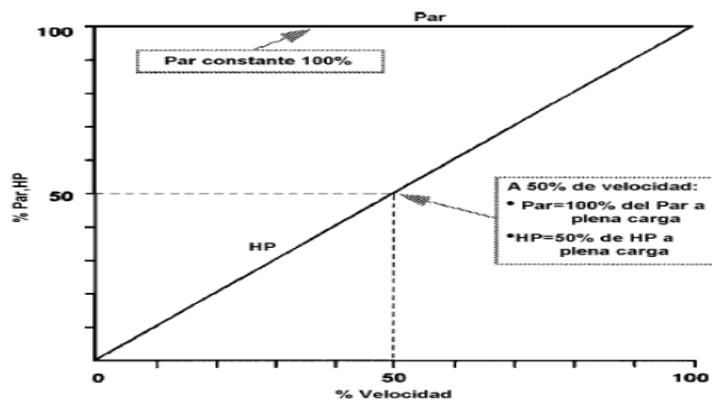
1.2.4. Procesos industriales con regulación de velocidad

Para estimar el ahorro es necesario conocer el proceso industrial en que se pretende instalar el regulador de velocidad. No todos los procesos ahorran energía, incluso hay procesos en que la energía no aumenta con la velocidad. Con el fin de identificar puntos de ahorro en la industria, se verá a continuación las cargas típicas que se pueden encontrar. En la mayoría de los procesos, se tendrán combinaciones de varios de estos tipos.

Cargas de par constante: son cargas en las que el Par no es función de la velocidad sino que permanece constante, mientras que la potencia varía linealmente con la velocidad. Las aplicaciones típicas son:

- Accionamientos de tracción
- Bandas transportadoras
- Bombas de desplazamiento positivo
- Grúas y malacates

Figura 11. Curva de par constante



Fuente: [http://3.bp.blogspot.com/-](http://3.bp.blogspot.com/-M8q8HI0weLo/Ti6Ym3RzpnI/AAAAAAAAABs/kLUfQOvh5I8/s1600/Grafica%2B4.png)

M8q8HI0weLo/Ti6Ym3RzpnI/AAAAAAAAABs/kLUfQOvh5I8/s1600/Grafica%2B4.png Consulta:

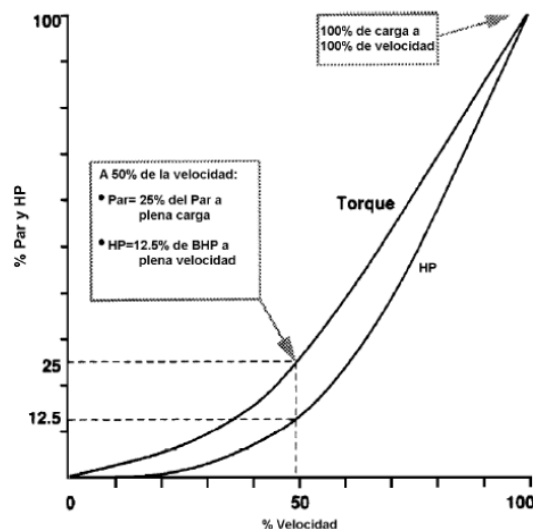
19 de febrero de 2013.

Cargas de par variable: las cargas en las que el par se reduce cuando operan por debajo de la velocidad base y se incrementa al operar por encima de dicha velocidad base se clasifican como cargas de par variable. En muchas de estas cargas el par se reduce con el cuadrado de la velocidad, siendo las bombas centrífugas y cierto tipo de ventiladores y sopladores las cargas más representativas.

Cargas de potencia constante: en las cargas de potencia constante, cuando se opera por encima de la velocidad base, el Par requerido decrece, mientras que la potencia se mantiene casi constante. Las aplicaciones típicas son:

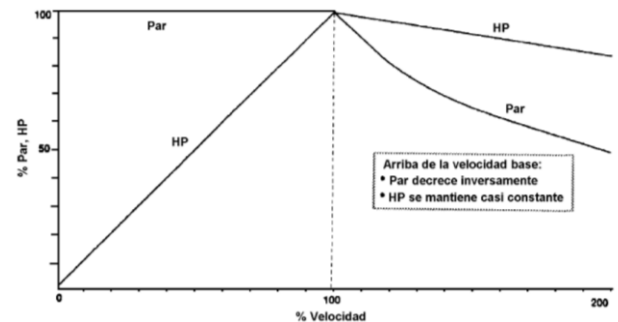
- Molinos
- Embobinadoras

Figura 12. **Comportamiento de cargas de par variable**



Fuente: http://ernestrincon.blogspot.com/2011_07_01_archive.html Consulta: 19 de febrero de 2013.

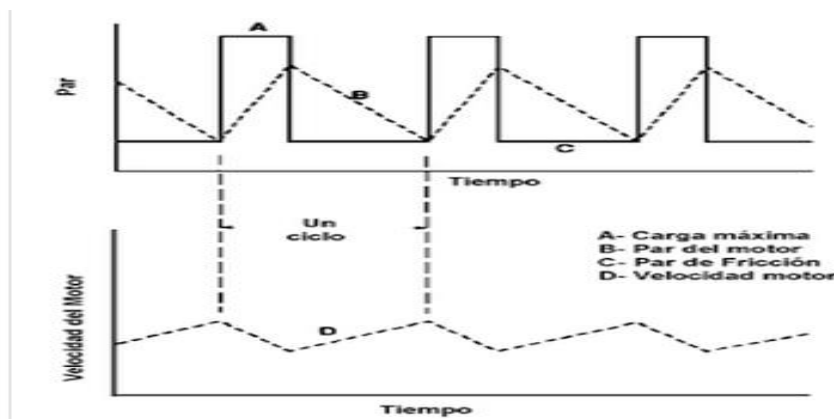
Figura 13. **Comportamiento de cargas de potencia constante**



Fuente: http://ernestrincon.blogspot.com/2011_07_01_archive.html Consulta: 19 de febrero de 2013.

En el caso de las prensas se requiere que la combinación del motor y el VFD produzcan un par suficiente de aceleración para regresar el volante de inercia a la velocidad requerida antes de que inicie el siguiente ciclo.

Figura 14. **Comportamiento de Cargas de Impacto**



Fuente: http://ernestrincon.blogspot.com/2011_07_01_archive.html Consulta: 19 de febrero de 2013.

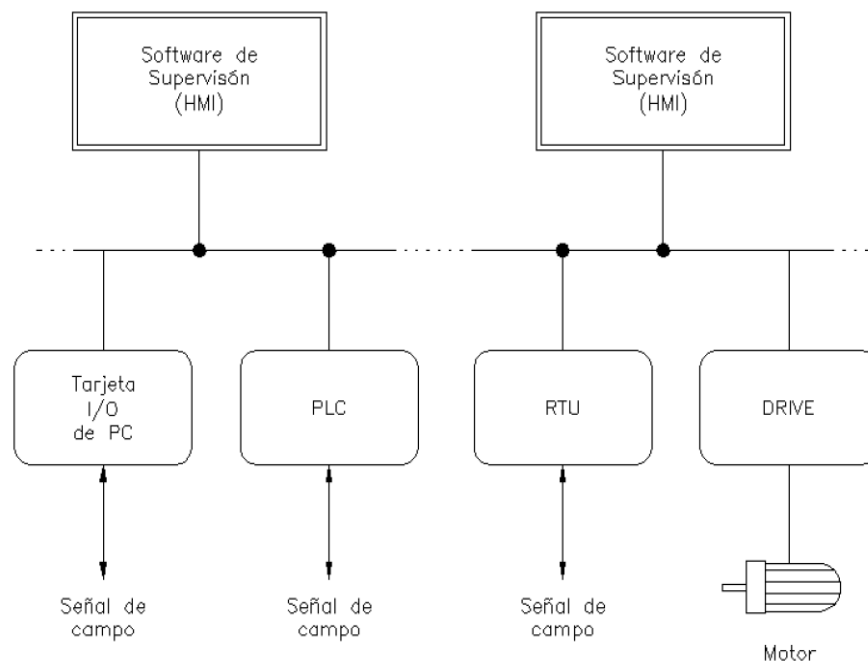
1.3. Interfaz Hombre-Máquina

La sigla HMI es la abreviación en ingles de Interfaz Hombre-Máquina. Los sistemas HMI pueden pensarse como una “ventana” de un proceso. Esta

ventana puede estar en dispositivos especiales como paneles de operador o en una computadora. Los sistemas HMI en computadoras se los conoce también como software HMI (en adelante HMI) o de monitoreo y control de supervisión.

Las señales del procesos son conducidas al HMI por medio de dispositivos como tarjetas de entrada/salida en la computadora, PLC's (Controladores lógicos programables), RTU (Que son Unidades Remotas de I/O) o DRIVE's (Variadores de velocidad de motores). Todos estos dispositivos deben tener una comunicación que entienda el HMI.

Figura 15. **Diagrama de un sistema HMI**



Fuente: <http://iaci.unq.edu.ar/materias/laboratorio2/HMI/Introduccion%20HMI.pdf> Consulta: 19 de febrero de 2013.

La Unidad Terminal Remota o RTU por sus siglas en ingles es la que envía señales a un sitio remoto donde se procesan las cuales son señales

independientes que se obtienen de los procesos. Los *Drive's* que en este caso se tratan de variadores de frecuencia.

1.3.1. Tipos de HMI

- Desarrollos a medida. Se desarrollan en un entorno de programación gráfica como VC++, Visual Basic, Delphi, etc.
- Paquetes enlatados HMI. Son paquetes de software que contemplan la mayoría de las funciones estándares de los sistemas SCADA. Ejemplos son FIX, WinCC, Wonderware, etc.

1.3.2. Funciones de un software HMI

Estas son las funciones de un software HMI:

- Monitoreo: es la habilidad de obtener y mostrar datos de la planta en tiempo real. Estos datos se pueden mostrar como números, texto o gráficos que permitan una lectura más fácil de interpretar.
- Supervisión: esta función permite junto con el monitoreo la posibilidad de ajustar las condiciones de trabajo del proceso directamente desde la computadora.
- Alarmas: es la capacidad de reconocer eventos excepcionales dentro del proceso y reportarlo estos eventos. Las alarmas son reportadas basadas en límites de control pre-establecidos.
- Control: es la capacidad de aplicar algoritmos que ajustan los valores del proceso y así mantener estos valores dentro de ciertos límites. Control va

más allá del control de supervisión removiendo la necesidad de la interacción humana. Sin embargo la aplicación de esta función desde un software corriendo en una PC puede quedar limitada por la confiabilidad que quiera obtenerse del sistema.

- Históricos: es la capacidad de muestrear y almacenar en archivos, datos del proceso a una determinada frecuencia. Este almacenamiento de datos es una poderosa herramienta para la optimización y corrección de procesos.

Y estas son las principales tareas de un software de supervisión y control que conforman la parte vital para el monitoreo y el control que se tiene a largas distancias:

- Permitir una comunicación con dispositivos de campo
- Actualizar una base de datos “dinámica” con las variables del proceso
- Visualizar las variables mediante pantallas con objetos animados (mímicos).
- Permitir que el operador pueda enviar señales al proceso, mediante botones, controles ON/OFF, ajustes continuos con el mouse o teclado.
- Supervisar niveles de alarma y alertar/actuar en caso de que las variables excedan los límites normales.
- Almacenar los valores de las variables para análisis estadístico y/o control.
- Controlar en forma limitada ciertas variables de proceso

Entre los tipos de software de supervisión y control para PC se tienen los lenguajes de programación visual como Visual C++ o Visual Basic que se utilizan para desarrollar software HMI a medida del usuario. Una vez generado el software el usuario no tiene posibilidad de re-programarlo. Y los Paquetes de desarrollo que están orientados a tareas HMI que pueden ser utilizados para desarrollar un HMI a medida del usuario y/o para ejecutar un HMI desarrollado para el usuario. El usuario podrá reprogramarlo si tiene la llave (software) como parahacerlo. Ejemplos son FIX Dynamics, Wonderware, PCIM, Factory Link, WinCC.

De esta manera facilitan las tareas de diseño los paquetes orientados HMI/SCADA:

- Incorporan protocolos para comunicarse con los dispositivos de campo más conocidos. Drivers, OPC.
- Tienen herramientas para crear bases de datos dinámicas que encaminan al proceso.
- Incluyen gran cantidad de librería de objetos para representar dispositivos de uso en la industria como: motores, tanques, indicadores, interruptores, etc.

1.3.3. Estructura general del software HMI

El software HMI está compuesto por un conjunto de programas y archivos. Hay programas para diseño y configuración del sistema y otros que son el motor mismo del sistema. En la figura 16 se muestra cómo funcionan algunos de los programas y archivos más importantes. Los rectángulos de la figura

representan programas y las elipses representan archivos. Los programas que están con recuadro simple representan programas de diseño o configuración del sistema y los que tienen doble recuadro representan programas que son el motor del HMI.

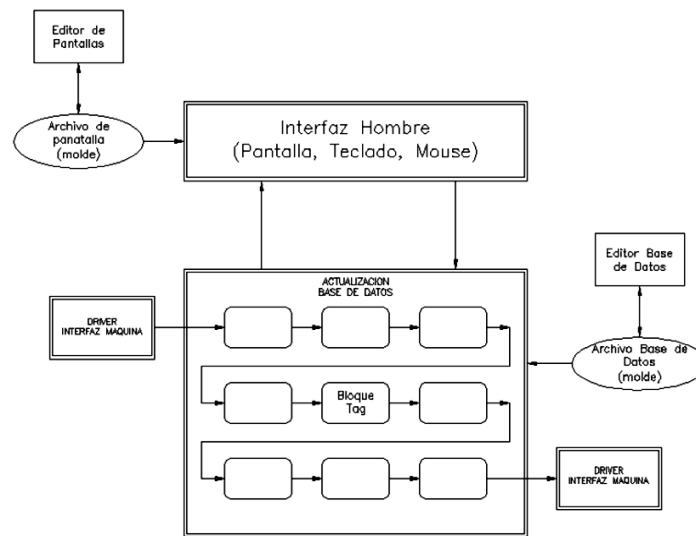
Con los programas de diseño, como el “editor de pantallas” se crea moldes de pantallas para visualización de datos del proceso. Estos moldes son guardados en archivos “Archivo de pantalla” y almacenan la forma como serán visualizados los datos en las pantallas.

Interfaz Hombre: es un programa que se encarga de refrescar las variables de la base de datos en la pantalla, y actualizarla, si corresponde, por entradas del teclado o el mouse. Este programa realiza la interfaz entre la base de datos y el hombre. El diseño de esta interfaz está establecido en el archivo molde “Archivo de pantalla” que debe estar previamente creado.

Base de datos: es un lugar de la memoria de la computadora donde se almacenan los datos requeridos del proceso. Estos datos varían en el tiempo según cambien los datos del procesos, por esta razón se denomina “base de datos dinámica”. La base de datos está formada por bloques que pueden estar interconectados. La creación de la base de datos, sus bloques y la relación entre ellos se realiza a través de “editor de base de datos”.

Driver: la conexión entre los bloques de la base de datos y las señales de los procesos se realiza por medio de drivers. Estos drivers manejan los protocolos de comunicación entre el HMI y los distintos dispositivos de campo. Los drivers son entonces la interfaz hacia la máquina que llevan a cabo la tarea de control con el dispositivo de acción final para tomar la correcta validación del proceso y que manipula muchas de las variables que ayudan a la interacción.

Figura 16. **Función de los archivos y programas para HMI**

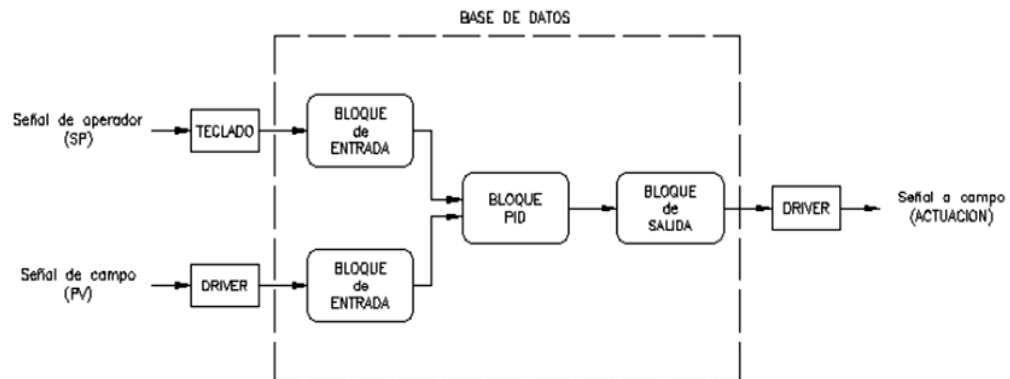


Fuente: <http://iaci.unq.edu.ar/materias/laboratorio2/HMI/Introduccion%20HMI.pdf> Consulta: 19 de febrero de 2013.

Bloques (tags): como se mencionó, la base de datos está compuesta por bloques. Para agregar o modificar las características de cada bloque se utiliza el editor de la base de datos. Los bloques pueden recibir información de los drivers u otros bloques y enviar información hacia los drivers u otros bloques. Las funciones principales de los bloques son:

- Recibir datos de otros bloques o al driver
- Enviar datos a otros bloques o al driver
- Establecer enlaces (links) a la pantalla (visualización, teclado o mouse)
- Realizar cálculos de acuerdo a instrucciones del bloque
- Comparar los valores con umbrales de alarmas
- Escalar los datos del driver a unidades de ingeniería. Los bloques pueden estructurarse en cadenas para procesar una señal.

Figura 17. Estructuración de bloques de un sistema HMI



Fuente: <http://iaci.unq.edu.ar/materias/laboratorio2/HMI/Introduccion%20HMI.pdf> Consulta: 19 de febrero de 2013.

1.4. Lenguaje de programación

El estándar IEC 1131-3 define 5 lenguajes de programación que pueden ser usados para definir los procedimientos de control y automatización. Estos lenguajes son los Nemónicos o lista de instrucciones, diagrama escalera, gráficos secuenciales, diagrama de bloques funcionales y el texto estructurado que forman el conjunto de lenguajes útiles para programar los autómatas que llevarán a cabo la tarea principal de monitorio y control a largas distancias, pero que fuera de estos lenguajes dentro del campo de automatización industrial se limitan a ser los únicos que permiten realizar la tarea básica del proyecto. Los lenguajes son los que traducen una instrucción a una máquina que realiza el proceso automatizado.

- Nemónicos o lista de instrucciones: es un lenguaje de bajo nivel basado en operaciones Booleanas y cuya apariencia es similar al código del lenguaje ensamblador.

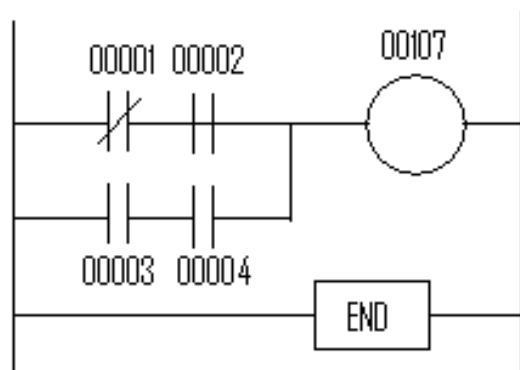
Tabla I. **Ejemplo de programación en lista de instrucciones**

ETIQUETA	INSTRUCCIÓN	OPERANDO
00001	LD	00001
00002	LDN	00002
00003	AND	
00004	LD	00003
00005	LD	
00006	AND	
00007	OR	
00008	ST	00107
00009	AND	

Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 48.

- Diagrama escalera: es un lenguaje de programación gráfica que conserva la estructura de los diagramas eléctricos de control.

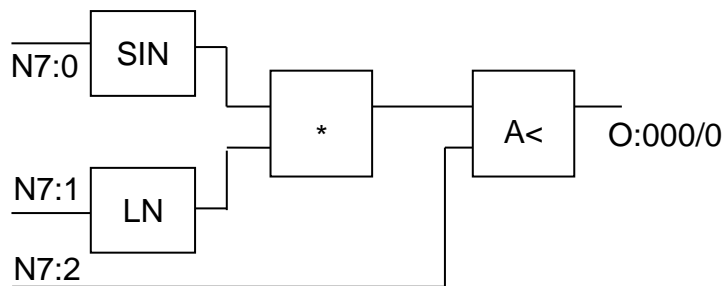
Figura 18. **Ejemplo de diagrama escalera**



Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 49.

- Gráficos secuenciales (GRAFCET): este lenguaje divide el ciclo de proceso en un cierto número de pasos bien definidos, y en transiciones que los separan. Este lenguaje es el núcleo del estándar IEC 1131-3. Los otros lenguajes se usan para describir las acciones realizadas en cada uno de los pasos, y para describir las condiciones lógicas para pasar de una etapa a otra (Transiciones).
- Diagrama de bloques funcionales: es un lenguaje gráfico que permite al usuario construir procedimientos complejos mediante la unión de bloques funcionales prediseñados.

Figura 19. **Programación con bloques funcionales**

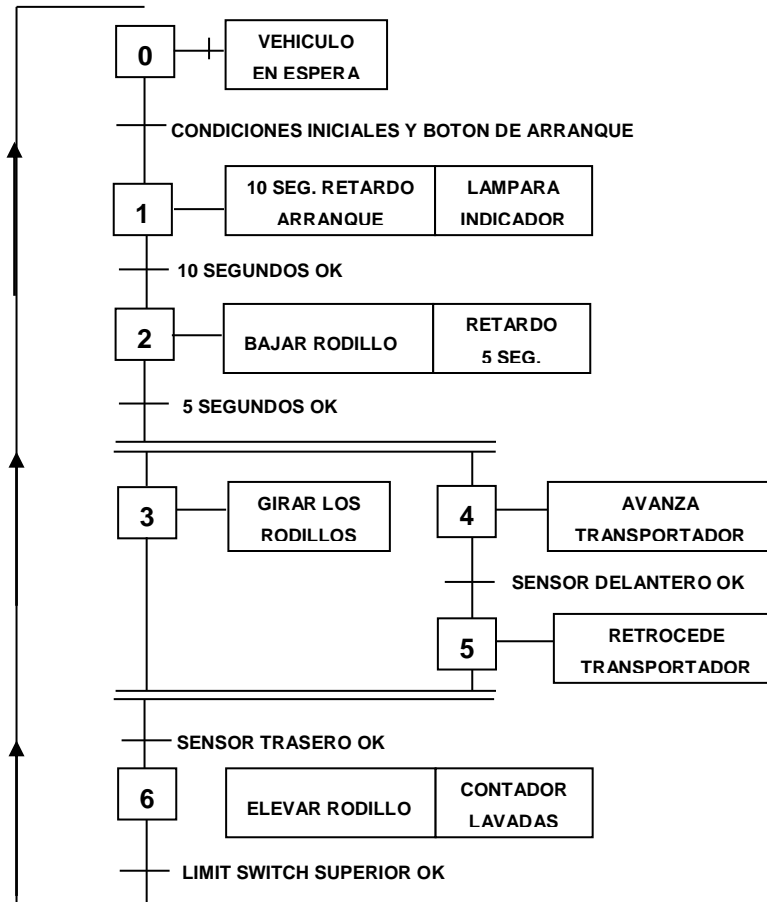


Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 49.

- Texto estructurado: este es un lenguaje estructurado de alto nivel parecido al PASCAL, pero más intuitivo para el ingeniero de control. Este lenguaje es usado principalmente para implementar procedimientos complejos que no pueden ser expresados mediante lenguajes gráficos.

Estos lenguajes son los que conforman el conjunto de instrucciones básicas que llevan a cabo la tarea de manipular el autómata. Una instrucción es una sentencia que le indica a la máquina la operación que se debe de realizar.

Figura 20. **Ejemplo de diagrama GRAFSET**



Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 50.

Figura 21. **Programación de código de programación con texto estructurado**

```

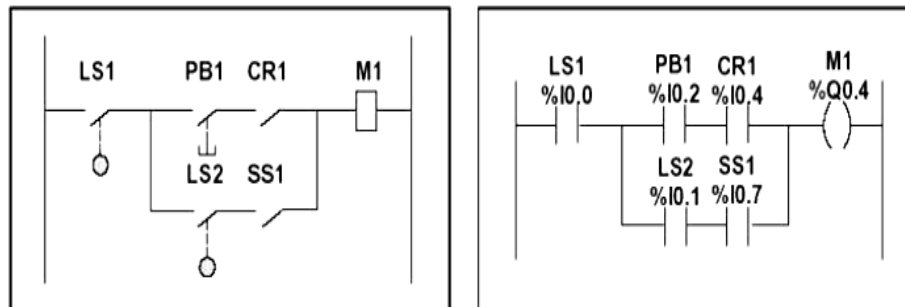
    REPEAT
    UNTIL N7:0>=10;
    END_REPEAT;
  
```

Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 50.

1.4.1. Descripción del lenguaje Ladder

Son similares a los diagramas de lógica cableada usados para representar los circuitos de control a relés. La principal diferencia entre las dos es que en la programación en escalera, todas las entradas son representadas mediante símbolos de contacto (-| |-), todas las salidas son representadas mediante bobinas de salida (-()-), y las operaciones numéricas están incluidas en el conjunto de instrucciones del cual dispone el lenguaje.

Figura 22. **Equivalencia entre lógica a relés y diagrama escalera**



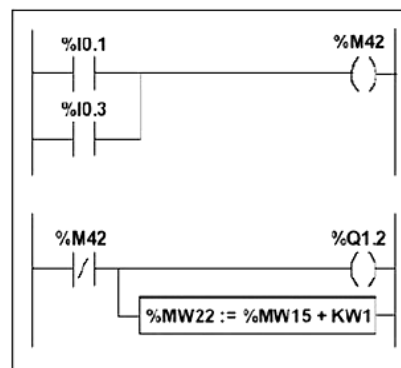
Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 50.

La figura 22 ilustra un diagrama simplificado del control con lógica a relés y su equivalente en diagrama de escalera. Nótese que en el diagrama de escalera, todos los contactos de entrada están asociados con uno de los elementos de conmutación del diagrama de lógica a relés. La bobina de salida M1 en el diagrama de relés está representada con un símbolo de bobina de salida en el diagrama escalera. La dirección numérica que aparece en la parte superior de cada símbolo de contacto o de bobina en el diagrama escalera, hace referencia a la localización de la conexión de Entrada / Salida física en el bus de direcciones del PLC.

Un programa escrito en escalera (LADDER) está compuesto por peldaños (Conjunto de instrucciones gráficas específicas que se encuentra entre dos barras verticales que representan las líneas de potencial) que son ejecutados secuencialmente por el PLC. El conjunto de instrucciones gráficas disponibles representan:

- Las entradas / Salidas físicas del PLC (Pulsadores, sensores, relé auxiliares, luces pilotos, etc.).
- Las funciones internas del PLC (Temporizadores, contadores, programadores cíclicos, etc.).
- Las operaciones matemáticas (Suma, división, AND, XOR, etc.)
- Las operaciones de comparación y manipulación de datos (A <B, A = B, Desplazar, rotar, etc.).
- Las variables internas al PLC (*bits*, palabras, etc.)

Figura 23. **Ejemplo de peldaños**



Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 50.

Estas instrucciones gráficas se relacionan mediante líneas de conexión horizontales y verticales, que al final conducen a una o varias salidas y/o acciones. Un peldaño no puede soportar más que un grupo de instrucciones enlazadas. Por ejemplo el siguiente diagrama está compuesto por dos peldaños.

1.4.2. Principios de programación

En un PLC típico, cada peldaño está compuesto por ejemplo por un conjunto máximo de 7 líneas y 11 columnas organizadas en dos zonas perfectamente definidas:

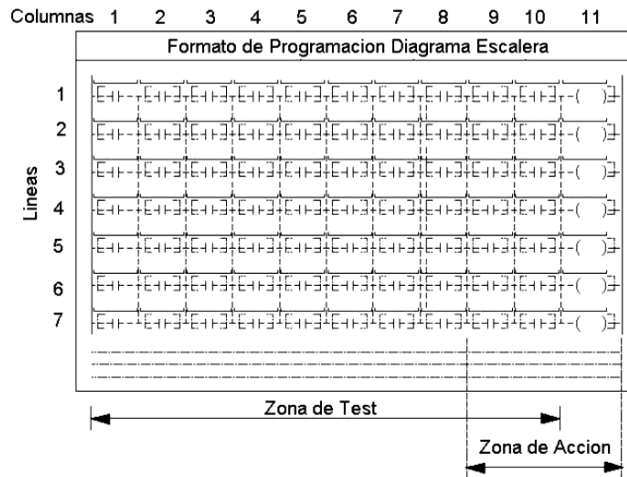
- Una zona de TEST que contiene las condiciones que se deben cumplirse o ser ciertas (TRUE) para que una acción tome lugar.
- Una zona de acciones la cual contiene las salidas u operaciones que se ejecutan a partir de los TEST conectados a ellas.

De esta forma el peldaño se visualiza como un formato de programación de 7 por 11, comenzando la programación a partir del tope de la izquierda. Aquí, se deben programar las instrucciones de TEST, comparaciones, y funciones, justificándolas por la izquierda. Estas pruebas lógicas entonces conducen a la zona de acción donde se programan las bobinas de salida, operaciones numéricas e instrucciones de control del flujo del programa; justificándola por la derecha.

El peldaño es resuelto o ejecutado por el PLC (Se hacen los TEST y se asignan las salidas u operaciones) desde arriba hacia abajo y desde la izquierda a la derecha. Generalmente, además de la zonas TEST y ACCIONES mencionadas, también existe espacio para comentarla finalidad lógica el

peldaño. Normalmente este espacio aparece como cabecera o también puede ir como comentario lateral derecho.

Figura 24. **Formato de programación**



Fuente: NAVARRO, Danilo. Controlador Lógico Programable (PLC). p. 103.

Entonces como resumen cabe mencionar que el lenguaje básico es el lenguaje de máquina, una colección de instrucciones muy detallada que controla la circuitería interna de la máquina. Este es el dialecto natural de la máquina. Muy pocos programas se escriben actualmente en lenguaje de maquina por dos razones importantes: primero, porque el lenguaje de maquina es muy incómodo para trabajar y segundo porque la mayoría de las maquinas se pide programar en diversos tipos de lenguajes, que son lenguajes de alto nivel, cuyas instrucciones son más compatibles con los lenguajes y la forma de pensar humanos como lo es el lenguaje c que además es de propósito general.

Debido a que los programas diseñados en este lenguaje se pueden ejecutar en cualquier máquina, casi sin modificaciones. Por tanto el uso del lenguaje de alto nivel ofrece tres ventajas importantes, sencillez, uniformidad y portabilidad.

2. PROTOCOLO DE COMUNICACIÓN MODBUS

En principio un protocolo de comunicación es un conjunto de reglas que permiten la transferencia e intercambio de datos binarios entre los distintos dispositivos que conforman una red. Estos han tenido un proceso de evolución gradual a medida que la tecnología electrónica ha avanzado y muy en especial en lo que se refiere a los microprocesadores.

2.1. Protocolo RS485

La Electronics Industries Association (EIA), en 1983 autorizó un nuevo estándar de transmisión diferencial llamado RS-485. Este estándar es similar en muchos aspectos al popular estándar EIA RS-422, de hecho RS-485 se puede considerar como el resultado de la expansión del RS-422, para permitir "drivers" y receptores múltiples multiterminal, compartiendo la misma línea de datos de transmisión. El protocolo RS485 tiene una gran demanda de uso industrial debido a que es capaz de manejar corrientes elevadas y permite que los dispositivos electrónicos sean de fácil acceso.

El estándar RS-485, como el estándar RS-422, especifica solamente las características eléctricas del "driver" y del receptor para ser utilizado en la línea de transmisión, pero no especifica o recomienda ningún protocolo de software, es decir, cuando se decide utilizar el protocolo RS-485 para interconectar dispositivos se deben de seguir las especificaciones de hardware que marca y que más adelante se describen, pero si se necesita otro tipo de software se deja a criterio del usuario o de la aplicación a desarrollar. El estándar EIA RS-485 ha tenido mucha aceptación.

Los usuarios son ahora capaces de hacer redes de área local económicas y enlaces en comunicaciones multiterminales utilizando cables de par trenzado y el protocolo de su opción. Dicha aceptación del estándar RS-485 está también reflejado por el hecho de que otras normas la refieren cuando se especifica un enlace de datos multiterminal, ANSI (American National Standard Institute), normas IPI (Intelligent Peripheral Interface) y SCSI (Small Computer Systems Interface), han utilizado el estándar RS-485 como la base para la interconexión en modo diferencial.

El estándar IPI especifica la interconexión entre controladores de disco y adaptadores de *host* a velocidades de transmisión de 2.5 mega baudios sobre un enlace de datos hasta 50 metros NRZ (Non Return to Zero). El estándar SCSI especifica la interconexión entre computadores personales, "drives" de disco, impresoras, escáner, y otros periféricos a velocidades de transmisión de 4 mega baudios sobre un enlace de 25metros. Hasta la introducción del estándar RS-485, el estándar RS-422 fue la interconexión estándar más ampliamente aceptada para la transmisión de datos en modo diferencial.

El estándar RS-485 fue creado para permitir la conexión de varios dispositivos a través de una línea común. Cada dispositivo representa una Unidad de Carga (UL = Unit Load) y se denomina nodo. Una red RS-485 puede contener hasta 32 UL, donde cada UL representa una carga de 12 K. Utilizando receptores de alta impedancia, la cantidad de nodos puede aumentarse hasta 256.

2.1.1. Características y arquitectura de la conexión

El protocolo RS-485 para poder ser usado necesita un adaptador RS-232 que puede usarse en cualquier puerto serial de cualquier computadora o un

dispositivo entrada/salida. Su alta velocidad y su poco consumo de poder, permite al convertidor poder usarlo con el puerto serie RS-232 o cualquier alimentación externa de 3.3 hasta 40 volts. Las características con que cuenta este protocolo son:

- Puerto de energía para conexión externa
- Conector DE-9F (DB9) con tres tipos de cable para el conector RJ11
- Entrada / salida de un unbuffered
- Rango de datos de entrada es igual al rango de los datos de salida

Opciones eléctricas:

- Solo transmisión
- Solo recepción
- RTS-Hardware Control de flujo TX/RX en convertidor DCE
- CTS-Hardware Control de flujo TX/RX en convertidor DTE
- Revertidor de señal TX y RX

Opciones mecánicas:

- Convertidor Hembra con Thumbscrews(Tornillo Manual)
- Convertidor Macho con Thumbscrews
- Convertidor Hembra con Jackscrews(Tornillo no Manual)
- Convertidor Macho con Jackscrews

El cableado puede hacerse por medio de cualquier RJ11 4P/4C (4 pines / 4 conectores), 6P/4C (6 pines / 4 conectores), o el estándar 4P/AC (4pines / 4 conectores) con un acoplador de teléfono o una placa de pared jack.

También se puede acoplar un RJ-45 Ethernet o jacks. El voltaje de entrada debe de ser de 3.3 volts y hasta 40 volts. RS 485 es un dispositivo que usa una señal de par trenzado (TP), dos cables trenzados en ellos mismos. Se está hablando de “Transmisión de datos Balanceado”, o Diferencia en transmisión de voltaje”. Si podría etiquetar los cables TP ‘A’ y el otro ‘B’, la señal se encuentra inactiva cuando el voltaje A es negativo y el voltaje en B son positivo.

Sin embargo, sí el cable A es positivo y el voltaje B es en negativo la señal se encuentra activa. Por supuesto, la diferencia entre el cable A y B es importante. Este dispositivo es usado para comunicaciones multipunto. Mas dispositivos pueden ser conectados a un cable de señal simple, similar a la red Ethernet, cual usa cable coaxial. La mayoría de los sistemas RS-485 usan arquitectura Maestro / esclavo, en donde cada esclavo cuenta con una dirección única y responde solamente a paquetes con la dirección de esa unidad. Estos paquetes están generados por el maestro, que periódicamente jala todos los dispositivos esclavos conectados.

Existen dos versiones del RS-485, el Par Trenzado Simple ó Par Trenzado Doble que a continuación se especificaran.

Par trenzado Simple: en esta versión, todos los dispositivos son conectados a un par trenzado simple. Así, estos deben de tener los driver con sus salidas de estado triple (incluyendo al maestro). La comunicación pasa a través de la línea simple en ambas direcciones. Es importante evitar más dispositivos se transmitan a la vez (problemas de software).

Doble par trenzado: aquí, el maestro no necesita la salida de estado triple, desde los dispositivos esclavos transmite arriba del segundo par trenzado, que

es la intención para mandar datos desde el esclavo hacia el maestro. Esta solución a menudo permite implementar comunicación multipunto en el sistema, donde originalmente se diseñó (HW así como SW) para el RS-232. Por supuesto, el software del maestro necesita de ser modificado para mandar paquetes de preguntas a los dispositivos esclavos.

El rendimiento de procesamiento de datos creciente es evidente en grandes volúmenes. Algunas veces puedes ver el sistema RS485 en un sistema multipunto. Es virtualmente idéntico al RS-422; no es usada la salida de los drivers del alto estado de impedancia. La única diferencia en hardware del circuito RS-485 y RS-422 es la habilidad de mandar la salida en estado de alta impedancia.

Tabla II. **Características generales del protocolo RS-485**

ESPECIFICACIONES		RS-485
Modo de operación		DIFERENCIAL
Total de numero de drivers y receptor de una Línea		1 DRIVER 32 RECEPTORES
Máximo Longitud del Cable		4000 ft.
Máximo Rango de Datos		10Mb/s
Máximo voltaje de salida del driver		-7V hasta +12V
Nivel de la señal de salida del Driver (Cargado Min.)	Cargado	+/-1.5V
Nivel de la señal de salida del Driver (Descargado Min.)	Descargado	+/-6V
Impedancia de carga del driver (Ohm)		54
Max. corriente del driver en alto estado Z	Power On	+/-100uA

Fuente: Liz Y. Magallanes D. Análisis de protocolo para la implementación de redes entre micro controladores p. 41.

2.1.2. Sistemas de bus RS485

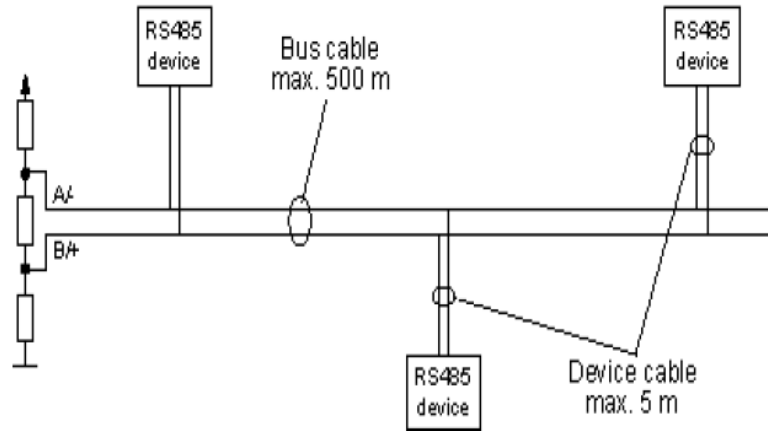
La interfaz RS485 ha sido desarrollada analógicamente a la interfaz RS422 alta velocidad a grandes distancias y encuentra creciente aplicación en el sector industrial. Pero mientras que la RS422 sólo permite la conexión unidireccional de hasta 10 receptores en un transmisor, la RS485 establece un Bus bidireccional con hasta 32 participantes. Físicamente las dos interfaces RS422 y RS485 pueden instalarse tanto como sistema de 2 hilos o de 4 hilos.

Dado que varios transmisores trabajan en una línea común, tiene que garantizarse con un protocolo que en todo momento esté activo como máximo un transmisor de datos. Los otros transmisores tienen que encontrarse en ese momento en estado ultra ohmio. La norma RS485 define solamente las especificaciones eléctricas para receptores y transmisores de diferencia en sistemas de bus digitales. La norma ISO 8482 estandariza además adicionalmente la topología de longitud máx. de 500 metros.

Bus de 2 hilos RS485: se puede montar una red con dos hilos simplificando de esta forma considerablemente las conexiones en el bus de comunicación. Las redes de 2 hilos RS2485 solo pueden comunicarse de forma dual-no simultáneo (*half-duplex*), lo que conlleva a un tipo de comunicación de pregunta-respuesta.

La ventaja de estas redes es que se puede montar una estructura multi-maestro (*multi-master*), es decir cada nodo tiene la facultad de comunicarse con cualquier otro. Evidentemente todo tiene un precio, las reglas de arbitraje y la detección de colisiones se hace imprescindible en este tipo de redes. La elección de hilos que puede manejar el protocolo RS485 es completamente dependiente del uso que tenga en el proceso y del papel que llevará a cabo.

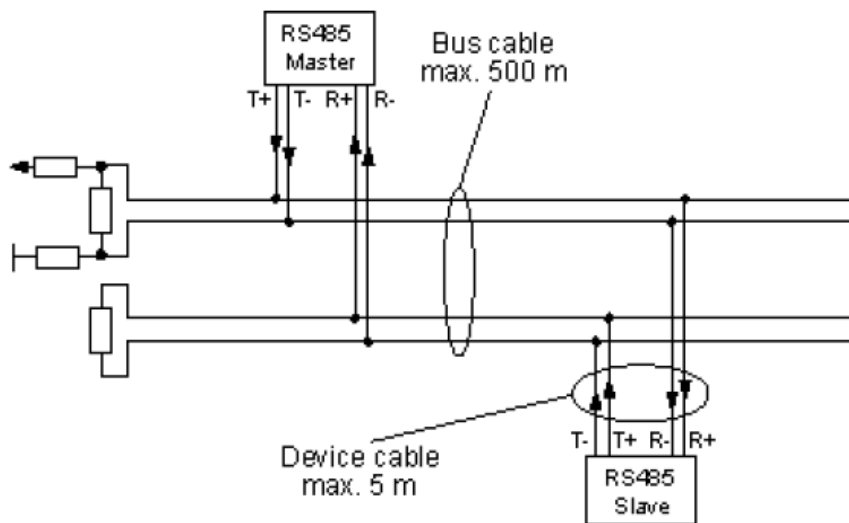
Figura 25. **Bus de 2 hilos RS485**



Fuente: <http://www.etitudela.com/entrenadorcomunicaciones/downloads/introduccionrs485.pdf>

Consulta: 13 de marzo de 2013.

Figura 26. **Bus de 4 hilos RS485**



Fuente: <http://www.etitudela.com/entrenadorcomunicaciones/downloads/introduccionrs485.pdf>

Consulta: 13 de marzo de 2013.

Bus de 4 hilos RS485: se pueden montar redes de cuatro hilos con una comunicación dual-simultánea (*full-duplex*) con una conexión semejante a la conexión punto a punto. Aunque la comunicación entre dos nodos puede ser

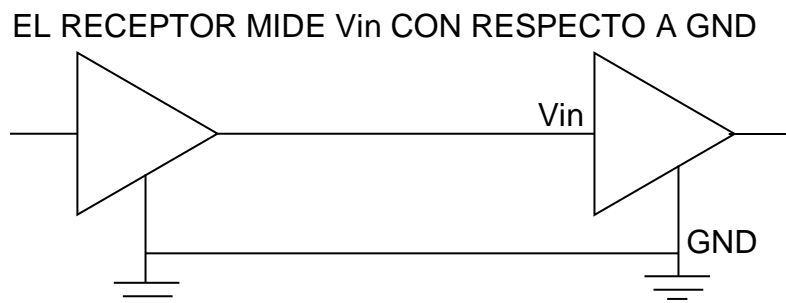
(*full-duplex*) solo un nodo de la red tiene la facultad de comunicarse con todos los demás, a este nodo se le denomina maestro (*Master*), al resto de nodos se les denomina esclavos (*Slave*). Los modos esclavos no pueden establecer una comunicación entre sí, estos se han de comunicar siempre con el master.

2.1.3. Balanceo y desbalanceo de líneas

La razón por la que RS-485 puede transmitir a largas distancias, es porque utiliza el balanceo de líneas. Cada señal tiene dedicados un par de cables, sobre uno de ellos se encontrará un voltaje y en el otro se estará su complemento, de esta forma, el receptor responde a la diferencia entre voltajes. La ventaja de las líneas no balanceadas es su inmunidad al ruido.

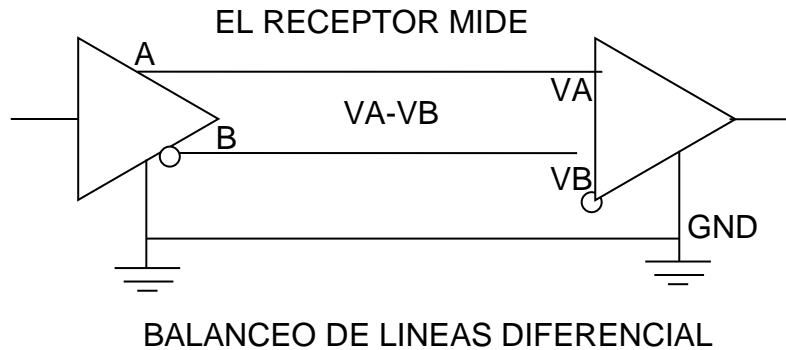
En cuanto a las líneas balanceadas la TIA/EIA-485 designa a estas dos líneas como A y B. En el controlador TX, una entrada alta TTL causa que la línea A sea más positiva (+) que la línea B, mientras que un bajo en lógica TTL causa que la línea B sea más positiva (+) que la línea A. Por otra parte en el controlador de recepción RX, si la entrada A es más positiva que la entrada B, la salida lógica TTL será "1" y si la entrada B es más (+) que la entrada A, la salida lógica TTL será un "0".

Figura 27. Línea no balanceada



Fuente: <http://www.i-micro.com/pdf/articulos/rs-485.pdf> Consulta: 11 de marzo de 2013.

Figura 28. **Línea balanceada**



Fuente: <http://www.i-micro.com/pdf/articulos/rs-485.pdf> Consulta: 11 de marzo de 2013.

2.2. **Introducción al protocolo Modbus**

La designación Modbus Modicon corresponde a una marca registrada por Gould Inc. Como en tantos otros casos, la designación no corresponde propiamente al estándar de red, incluyendo todos los aspectos desde el nivel físico hasta el de aplicación, sino a un protocolo de enlace (nivel OSI 2). Puede, por tanto, implementarse con diversos tipos de conexión física y cada fabricante suele suministrar un software de aplicación propio, que permite parametrizar sus productos.

No obstante, se suele hablar de MODBUS como un estándar de bus de campo. Este protocolo define una estructura de mensaje en los controladores de reconocer y utilizar, independientemente del tipo de redes por las que se comunican. En él se describe el proceso que un controlador utiliza para solicitar acceso a otro dispositivo, cómo responderá a las peticiones de los otros dispositivos, y cómo los errores serán detectados y reportados. Se establece un modelo común para la disposición y el contenido de los campos del mensaje.

El protocolo Modbus proporciona el estándar interno que los controladores Modicon utilizaran para los mensajes de análisis. Durante las comunicaciones en una red Modbus, el protocolo determina cómo cada controlador sabrá su dirección de dispositivo, reconocerá un mensaje dirigido a él, determinar el tipo de medidas que deben tomarse, y extraer cualquier dato u otra información contenida en el mensaje. Si la respuesta es requerida, el controlador construirá el mensaje de respuesta y lo enviara utilizando el protocolo Modbus. En otras redes, los mensajes con el protocolo Modbus están incrustados en el marco o estructura de los paquetes que se utilizan en la red.

2.2.1. Transacciones sobre redes Modbus

Los puertos Estándar Modbus en los controladores Modicon usan una interface serial compatible RS232 que define el pin-out de los conectores, el cableado, los niveles de señal, las tasas de transmisión en baudios y comprobación de la paridad. Los controladores pueden ser conectados en red directamente o a través de módems. Los controladores se comunican usando una técnica maestro-esclavo, en el que sólo un dispositivo (el maestro) puede iniciar transacciones (llamadas "consultas").

Los otros dispositivos (los esclavos) responden al suministrar los datos solicitados para el maestro, o realizando la acción solicitada en la consulta. Los accesorios típicos maestros incluyen los procesadores del host y los paneles de programación. Los esclavos típicos incluyen controladores programables.

El maestro puede dirigirse a esclavos individuales o iniciar una difusión de mensajes para todos los esclavos. Los esclavos devuelven un mensaje (llamado una «respuesta») a las preguntas que se les envían individualmente. No se devuelven respuestas a las consultas de difusión del maestro. El

protocolo Modbus establece el formato para la consulta del maestro colocando en ella el dispositivo (o emisión) dirección, un código de función que define la acción solicitada, cualquier dato a ser enviado, y un campo de comprobación de errores. El mensaje de respuesta del esclavo también se construyó utilizando el protocolo Modbus.

Contiene campos que confirman la acción tomada, cualquier dato que sea devuelto, y un campo de comprobación de errores. Si se produce un error en la recepción del mensaje, o si el esclavo no puede realizar la acción solicitada, el esclavo construirá un mensaje de error y lo enviara como su respuesta.

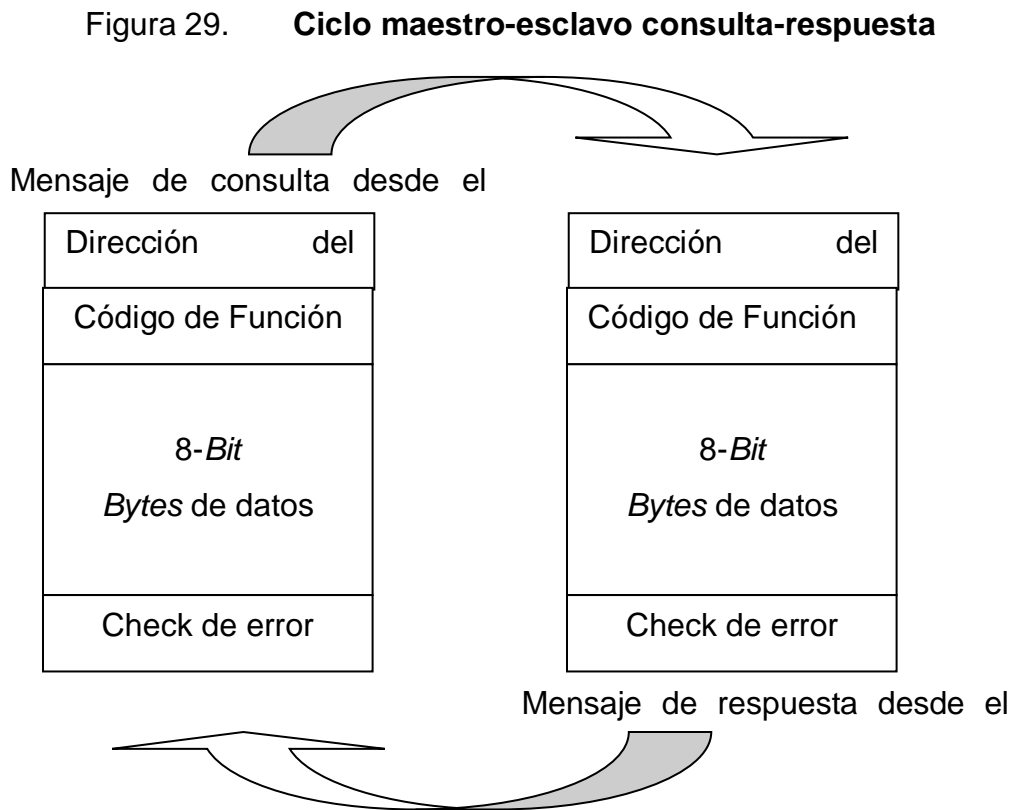
2.2.2. Transacciones en otro tipo de redes

Además de sus capacidades estándar de Modbus, algunos modelos de controladores Modicon pueden comunicarse a través de Modbus Plus utilizando una función de los puertos o adaptadores de red, y más de MAP, utilizando adaptadores de red. En estas redes, los controladores se comunican usando una técnica peer-to-peer, en el que cualquier controlador puede iniciar transacciones con los otros controladores. Así, un controlador puede operar ya sea como esclavo o como maestro en transacciones individuales.

Se tienen múltiples caminos internos con frecuencia para permitir el procesamiento simultáneo de las transacciones maestro y esclavo. A nivel del mensaje, el protocolo Modbus se sigue aplicando el principio maestro-esclavo a pesar de que el método de comunicación de la red es peer-to-peer. Si un controlador de origen de un mensaje, lo hace como un dispositivo maestro, y espera una respuesta de un dispositivo esclavo. Del mismo modo, cuando un controlador recibe un mensaje que construye una respuesta del esclavo y la devuelve al controlador de origen.

2.2.3. El ciclo de pregunta-respuesta

La Consulta: el código de función en la consulta le dice al dispositivo esclavo direccionado qué tipo de acción a realizar. Los *bytes* de datos contienen la información adicional que el esclavo necesita para realizar la función. Por ejemplo, el código de función 03 consultará al esclavo para leer registros de las explotaciones y responder con su contenido. El campo de datos debe contener la información que indica que el esclavo debe registrarse para empezar en el número de registros y de leer. El campo de comprobación de error proporciona un método para el esclavo para validar la integridad de los contenidos del mensaje.



Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 14.

La Respuesta: si el esclavo hace una respuesta normal, el código de función en la respuesta es un eco de código de función de la consulta. Los *bytes* de datos contienen los datos recogidos por el esclavo, como los valores de registro o de su estatuto. Si se produce un error, el código de función se ha modificado para indicar que la respuesta es una respuesta de error, y los *bytes* de datos contienen un código que describe el error. El campo de comprobación de error permite que el maestro para confirmar que el contenido del mensaje son válidos.

2.3. Modos de transmisión serial

Los controladores pueden ser configurados para comunicarse en redes Modbus estándar utilizando uno de los dos modos de transmisión: ASCII o RTU. Los usuarios seleccionan el modo deseado, junto con los parámetros de comunicación del puerto serie (velocidad de transmisión, modo de paridad, etc.), durante la configuración de cada controlador. El modo y parámetros de serie debe ser la misma para todos los dispositivos en una red Modbus. La selección del modo ASCII o RTU se refiere sólo a las redes Modbus estándar. En él se definen los contenidos de *bits* de los campos de mensajes transmitidos en serie en esas redes. Determina cómo la información se empaqueta en los campos del mensaje y descifrado.

En otras redes como MAP y Modbus Plus, los mensajes Modbus se colocan en marcos que no están relacionados con transmisión serie. Por ejemplo, una petición de lectura de registros de retención se puede manejar entre dos controladores en Modbus Plus sin tener en cuenta la configuración actual del puerto serie o bien controlador Modbus. Las redes MAP son una pila de protocolos basadas en el modelo de referencia de interconexión de sistemas abiertos OSI de ISO. Propone uno o varios protocolos estándar para cada nivel.

Una de sus características es la elección a nivel físico de paso de testigo en bus (Token Bus, normalizado como IEEE 802.4), debido a la exigencia en la industria de tiempos de respuesta acotados. Modbus Plus es un sistema de red de área local para aplicaciones de control industrial, dispositivos de red pueden intercambiar mensajes para el control y seguimiento de procesos en lugares remotos en la planta industrial.

2.3.1. Modo ASCII

Cuando los controladores están configurados para comunicarse en una red Modbus utilizando el modo ASCII (Código Estándar Americano para Intercambio de Información), cada *byte* de 8-*bits* 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 para producirse entre caracteres sin causar un error.

Es importante mencionar que el formato para cada *byte* en modo ASCII es diferente dependiendo de cada palabra. El formato para cada *byte* en modo ASCII es:

- Codificación de sistema: los caracteres hexadecimales, ASCII 0-9, A-F
Un carácter hexadecimal contenido en cada carácter ASCII del mensaje.
- *Bits* por *byte*: 1 *bit* de inicio, 7 *bits* de datos, *bit* menos significativo enviado, 1 *bit* de paridad par /impar; sin *bit* de no paridad, 1 *bit* deparada si se utiliza paridad, 2 *bits* si no hay paridad.
- Campo de comprobación de error: comprobación de redundancia longitudinal (LRC)

2.3.2. Modo RTU

Cuando los controladores están configurados para comunicarse en una red Modbus mediante el modo RTU (Remote Terminal Unit), cada *byte* de 8-*bits* en un mensaje contiene dos caracteres hexadecimales de 4-*bits*. La principal ventaja de este modo es que su mayor densidad de caracteres permite un mejor rendimiento de datos que ASCII para la misma velocidad de transmisión. Cada mensaje debe ser transmitido en un flujo continuo. El formato para cada *byte* en modo RTU es:

Codificación de sistema: 8-*bit* binario, hexadecimal 0-9, A-F, dos caracteres hexadecimales contenidos en cada campo del mensaje de 8-*bit*.

Bits por Byte: 1 *bit* de inicio, 8 *bits* de datos, *bit* menos significativo enviado, 1 *bit* para la paridad par /impar; sin *bit* de no paridad, 1 *bit* de parada si se utiliza paridad, 2 *bits* si no hay paridad.

Campo de comprobación de error: comprobación de redundancia cíclica (CRC).

2.4. Trama de un mensaje Modbus

En cualquiera de los dos modos de transmisión en serie (ASCII o RTU), un mensaje Modbus es colocado por el dispositivo de transmisión en un marco que tiene un principio conocido y el punto final. Esto permite a los dispositivos receptores comenzar al principio del mensaje, leerla parte de dirección y determinar qué dispositivo se dirige (o todos los dispositivos, si el mensaje se retransmite), y para saber cuándo el mensaje se ha completado.

Mensajes parciales pueden detectar errores y se puede establecer como un resultado. En redes como MAP o Modbus Plus, el protocolo de red se ocupa de la elaboración de mensajes con delimitadores de comienzo y final que son

específicos de la red. Estos protocolos también se encargan de la entrega al dispositivo de destino, por lo que el campo de dirección Modbus incrustada en el mensaje es innecesario para la transmisión real (La dirección Modbus es convertida a una dirección de nodo de red y enrutamiento de ruta por el controlador de origen o su adaptador de red.).

2.4.1. Trama ASCII

En el modo ASCII, los mensajes comienzan con un carácter 'dos puntos' (:) (HexadecimalASCII3A), y terminan con un "retorno de carro avance de línea" (CRLF) par (ASCII0Dy hex0A). Los caracteres permitidos de transmisión para todos los demás campos son hexadecimal0-9, A-F. Los dispositivos de red monitorizan el bus de red de forma continua durante el carácter 'dos puntos' carácter.

Cuando se recibe uno, cada dispositivo descodifica el campo siguiente (el campo de dirección) para averiguar si es el dispositivo direccionado. Intervalos de hasta un segundo puede transcurrir entre caracteres dentro del mensaje. Si se produce un intervalo mayor, el dispositivo receptor asume que un error ha ocurrido. Una trama de mensaje típica se muestra a continuación.

Figura 30. Trama de mensaje ASCII

COMIENZO	DIRECCION	FUNCION	DATOS	LRC CHECK	FIN
1 CHAR	2 CHARS	2 CHARS	CHARS	2 CHARS	2 CHARS CRLF

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 17.

2.4.2. Fragmento RTU

En el modo RTU, los mensajes comienzan con un intervalo de silencio de al menos 3,5 veces el carácter. Esto es más fácil de implementar como un múltiplo de caracteres a la velocidad de transmisión que se utiliza en la red (mostrado como T1-T2-T3-T4 en la figura siguiente). El primer campo es entonces transmitido la dirección del dispositivo. Los caracteres permitidos de transmisión para todos los campos son hexadecimales 0-9, A-F.

Los dispositivos de red monitorizan el bus de red continuamente, incluso durante los intervalos "silenciosos". Cuando el primer campo (campo de dirección) es recibido, cada dispositivo lo descodifica para averiguar si es el dispositivo direccionado.

Tras el último carácter transmitido, un intervalo similar de al menos 3,5 veces el carácter marca el final del mensaje. Un nuevo mensaje puede comenzar después de este intervalo.

El telegrama completo debe ser transmitido como un flujo continuo. Si un intervalo de silencio de más de 1,5 veces el carácter se produce antes de la finalización de la trama, el receptor asume que el dispositivo desplaza el mensaje incompleto y que el siguiente *byte* será el campo de dirección de un nuevo mensaje.

De manera similar, si un nuevo mensaje empieza antes que 3,5 veces el carácter después de un mensaje anterior, el dispositivo receptor lo considerará una continuación del mensaje anterior. Esto establecerá un error, como el valor en el campo final CRC no será válida para los mensajes combinados. Una trama de mensaje típica se muestra a continuación.

Figura 31. **Trama de mensaje RTU**

COMIENZO	DIRECCION	FUNCION	DATOS	LRC CHECK	FIN
T1-T2-T3-T4	8 BITS	8 BITS	BITS	16 BITS	T1-T2-T3-T4

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 18.

2.4.3. **Cómo se maneja el campo de direccionamiento**

El campo de dirección de un mensaje contiene dos caracteres (ASCII) u ocho *bits* (RTU). Las direcciones válidas de dispositivos esclavos están en el rango de 0 a 247 decimales. Los dispositivos esclavos individuales se asignan direcciones en el intervalo de 1 a 247. Un maestro se dirige a un esclavo mediante la colocación de la dirección del esclavo en el campo de dirección del mensaje.

Cuando el esclavo envía su respuesta, coloca su propia dirección en este campo de la dirección de la respuesta para permitir al maestro saber que el esclavo responde. La dirección 0 se utiliza para la dirección de difusión, que todos los dispositivos esclavos reconocen. Cuando se utiliza el protocolo Modbus en redes de mayor nivel, las emisiones no puede ser permitidas o pueden ser remplazadas por otros métodos.

2.4.4. **Cómo se maneja el campo de función**

El campo de código de función de un mensaje contiene dos caracteres (ASCII) u ocho *bits* (RTU). Los códigos válidos están en el rango de 1 a 255

decimales. De éstos, algunos códigos son aplicables a todos los controladores de Modicon, mientras que algunos códigos se aplican solo a ciertos modelos, y otros están reservados para uso futuro.

Cuando se envía un mensaje desde un maestro a un dispositivo esclavo el campo de código de función le indica al esclavo qué tipo de acción a realizar. Los ejemplos son para leer los estados ON/OFF de un grupo de bobinas o entradas discretas; para leer el contenido de datos de un grupo de registros; para leer el estado de diagnóstico del esclavo; escribir en bobinas o de los registros designados, o para permitir la carga, grabar, o verificar el programa en el esclavo.

Cuando el esclavo responde al maestro, se utiliza el campo de código de función para indicar una respuesta normal (sin errores) o algún tipo de error (llamado una respuesta de excepción). Para una respuesta normal, el esclavo simplemente se hace eco al código de función original. Para una respuesta de excepción, el esclavo devuelve un código que es equivalente al código de función original con su *bit* más significativo a un 1 lógico. Por ejemplo, un mensaje de maestro a esclavo para leer un grupo de registros de retención tendría el código siguiente función: 0000 0011 (Hexadecimal 03).

Si el dispositivo esclavo realiza la acción solicitada sin error, devuelve el mismo código en su respuesta. Si se produce una excepción, devuelve: 1000 0011 (Hexadecimal 83).

Además de su modificación del código de función para una respuesta de excepción, el esclavo coloca un código único en el campo de datos del mensaje de respuesta. Esto le dice al maestro el tipo de error, o la razón de la excepción. El programa de aplicación del dispositivo maestro tiene la responsabilidad de

manejar respuestas de excepción. Los procesos típicos son para publicar reintentos posteriores del mensaje, para tratar los mensajes de diagnóstico para el esclavo, y para notificar a los operadores.

2.4.5. Contenido del campo de datos

El campo de datos se construye utilizando dos dígitos hexadecimales, en el intervalo de 00 a FF en hexadecimal. Estos se pueden hacer a partir de un par de caracteres ASCII, o de un carácter RTU, de acuerdo con el modo de la red de transmisión en serie. El campo de datos de mensajes enviados desde un maestro a los dispositivos esclavos contiene información adicional que el esclavo debe utilizar para tomar la acción definida por el código de función. Esto puede incluir elementos como direcciones discretas y registro, la cantidad de temas que se manejan, y el recuento de *bytes* de datos reales sobre el terreno.

Por ejemplo, si el maestro solicita un esclavo a leer un grupo de registros de retención (código de función 03), el campo de datos especifica el registro de partida y cómo muchos registros se van a leer. Si el maestro escribe a un grupo de registros en el esclavo (código de función 10 hexadecimal), el campo de datos especifica el registro de inicio, el número de registros de escribir, el recuento de *bytes* de datos a seguir en el campo de datos, y los datos que se han escrito en los registros. Si no ocurre ningún error, el campo de datos de una respuesta de un esclavo a un maestro contiene los datos solicitados.

Si se produce un error, el campo contiene un código de excepción que la aplicación maestra puede utilizar para determinar la siguiente acción a tomar. El campo de datos puede ser inexistente (de longitud cero) en ciertos tipos de mensajes. Por ejemplo, en una solicitud de un dispositivo maestro para un esclavo a responder con su registro de sucesos de comunicaciones

(hexadecimal 0B el código de función), el esclavo no requiere ninguna información adicional. El código de función sólo especifica la acción.

2.4.6. Contenido del campo de verificación de error

Existen dos tipos de métodos de comprobación de errores que se utilizan para redes Modbus estándar. El contenido del campo de comprobación de errores dependerá del método que se está utilizando.

ASCII: cuando el modo ASCII se utiliza para enmarcar carácter, el campo de comprobación de errores contiene dos caracteres ASCII. Los caracteres de control de error son el resultado de una verificación de redundancia longitudinal (LRC) cálculo que se realiza en el contenido del mensaje, exclusivo de "dos puntos" al comienzo y terminación de caracteres CRLF. Los caracteres LRC se adjuntan al mensaje en el último campo que precede a los caracteres CRLF.

RTU: cuando el modo RTU se utiliza para la elaboración de caracteres, el campo de comprobación de error contiene un valor de 16-bit implementado como dos *bytes* de 8-bits. El valor de comprobación de error es el resultado de un cálculo de comprobación de redundancia cíclica a cabo en el contenido del mensaje. El campo CRC se anexa al mensaje como el último campo en el mensaje. Cuando se hace esto, el *byte* de orden inferior del campo se añade primero, seguido por el *byte* de orden superior. El *byte* CRC de orden superior es el último *byte* a ser enviado en el mensaje.

2.4.7. Como son transmitidos los caracteres de manera serial

Cuando los mensajes se transmiten en las redes serie Modbus estándar, cada carácter o *byte* se envía en este orden (de izquierda a derecha):

Bit menos significativo (LSB). . . *Bit* más significativo (MSB)

Con trama de caracteres ASCII, la secuencia de *bits* es la siguiente:

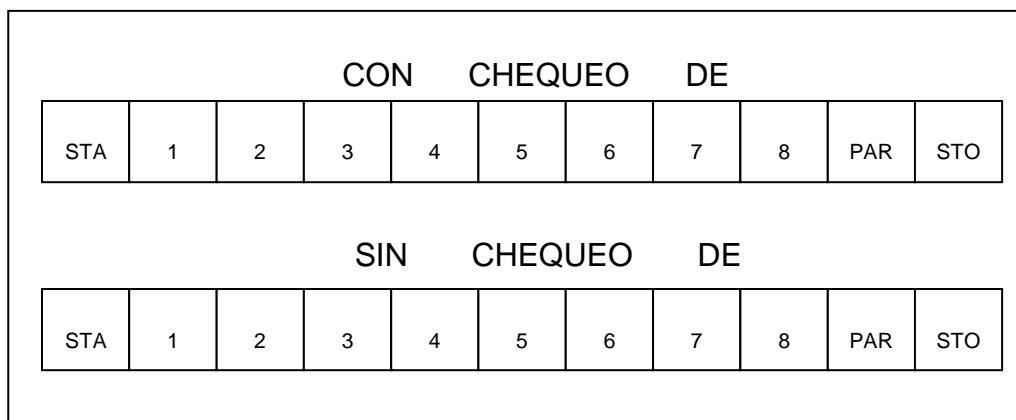
Figura 32. **Orden de *bits* (ASCII)**



Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 22.

Con trama de caracteres RTU, la secuencia de *bits* es la siguiente:

Figura 33. **Orden de *bits* (RTU)**



Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 22.

2.5. Métodos de verificación de error

Las redes serial Estándar Modbus utilizan dos tipos de comprobación de errores. La comprobación de paridad (par o impar) puede aplicarse opcionalmente a cada carácter. La trama de control (LRC o CRC) se aplica a todo el mensaje. Tanto el carácter de comprobación de verificación de trama y mensaje son generados en el dispositivo maestro y aplican el contenido del mensaje antes de la transmisión.

El dispositivo esclavo comprueba cada carácter y la trama de mensaje durante toda la recepción. El maestro está configurado por el usuario para esperar durante un intervalo de tiempo de espera predeterminado antes de abortar la transacción. Este intervalo se fija con el tiempo suficiente para que cualquier esclavo pueda responder con normalidad. Si el esclavo detecta un error de transmisión, el mensaje no actúa sobre ellos.

El esclavo no construirá una respuesta al maestro. Así, el tiempo de espera expira y permite que el programa del maestro maneje el error. Tenga en cuenta que si un mensaje dirigido a un dispositivo esclavo no existente también causará un time out. En virtud de las redes, el campo del mensaje Modbus LRC o CRC de verificación no se aplica. En el caso de un error de transmisión, los protocolos de comunicación específicos a esas redes notifican al dispositivo de origen que ha ocurrido un error, y permiten que se vuelva a intentar o abortar de acuerdo con la forma en que ha sido configurada.

Si el mensaje es entregado, pero el dispositivo esclavo no puede responder, un error de tiempo de espera puede ocurrir y puede ser detectado por el programa del maestro.

2.5.1. Chequeo de paridad

Los usuarios pueden configurar los controladores para la comprobación de paridad par o impar, o sin comprobación de paridad. Esto determinará cómo el *bit* de paridad se establecerá en cada personaje. Si cualquiera de paridad Par o Impar especifica la cantidad de *bits* 1 se contará en la parte de datos de cada carácter (siete *bits* de datos para el modo ASCII, y ocho para RTU). El *bit* de paridad a continuación, se establece en un 0 o 1 para dar como resultado un total de par o impar de *bits* 1. Por ejemplo, estos ocho *bits* de datos están contenidos en una trama de caracteres RTU: 1100 0101.

La cantidad total de 1 *bits* en la trama es de cuatro. Si se utiliza paridad par, el *bit* de la trama paridad será un 0, por lo que la cantidad total de *bits* 1 todavía es un número par (cuatro). Si se utiliza paridad impar, el *bit* de paridad será un 1, por lo que una cantidad impar (cinco). Cuando se transmite el mensaje, el *bit* de paridad se calcula y se aplicará a la estructura de cada carácter. El dispositivo de recepción cuenta la cantidad de *bits* 1 y establece un error si no son el mismo que el configurado para ese dispositivo (todos los dispositivos en la red Modbus deben estar configurados para utilizar el método de verificación de paridad mismo).

Tenga en cuenta que la comprobación de paridad sólo puede detectar un error si un número impar de *bits* son recogidos o dejados en un marco de carácter durante la transmisión. Por ejemplo, si la comprobación de paridad impar se emplea, y dos *bits* 1 se reducen de un carácter que contiene tres *bits* de 1, el resultado es todavía un conteo impar de *bits* 1. Si no se especifica la comprobación de paridad, el *bit* de paridad no se transmite y sin verificación de paridad se pueden hacer. Un *bit* de parada adicional se transmite al llenar la trama de caracteres.

2.5.2. Chequeo LRC

En el modo ASCII, los mensajes incluyen un campo de comprobación de errores que se basan en un método de comprobación de redundancia longitudinal (LRC). El campo LRC comprueba el contenido del mensaje, sin contar los "dos puntos" al comienzo y finalización por CRLF. Se aplica independientemente de cualquier método de comprobación de paridad utilizado para los caracteres individuales del mensaje.

El campo LRC es de un *byte*, contiene un valor binario de 8-*bits*. El valor de LRC es calculado por el dispositivo de transmisión, que agrega el LRC al mensaje. El dispositivo de recepción calcula una LRC durante la recepción del mensaje, y compara el valor calculado con el valor real que recibió en el campo LRC. Si los dos valores no son iguales, se producirá un error. El LRC se calcula mediante la suma de los sucesivos *bytes* de 8 *bits* del mensaje, descartando cualquier acarreo, y luego los dos complementan el resultado.

Se realiza en el contenido del campo de mensaje ASCII excluyendo el carácter 'colon' con que comienza el mensaje, y excluyendo el par CRLF al final del mensaje. En la lógica de escalera, la función CKSM calcula un LRC de los contenidos del mensaje.

2.5.3. Chequeo CRC

En el modo RTU, los mensajes incluyen un campo de comprobación de errores que se basa en un método comprobación de redundancia cíclica (CRC). El campo CRC comprueba el contenido de todo el mensaje. Se aplica independientemente de cualquier método de comprobación de paridad utilizado para los caracteres individuales del mensaje. El campo CRC es de dos *bytes*,

contiene un valor binario de 16-*bit*. El valor CRC se calcula por el dispositivo de transmisión, que agrega el CRC para el mensaje.

El dispositivo receptor vuelve a calcular un CRC durante la recepción del mensaje, y compara el valor calculado con el valor real recibido en el campo CRC. Si los dos valores no son iguales, se producirá un error. El CRC es iniciado por primera precargado de un registro de 16-*bit* con todos 1. Entonces se inicia un proceso de aplicación de sucesivas de 8-*bits* del mensaje con el contenido actual del registro. Sólo los ocho *bits* de datos en cada carácter se utilizan para la generación de la CRC. Iniciar y detener *bits* y el *bit* de paridad, no se aplican a la CRC.

Durante la generación del CRC, cada carácter de 8-*bits* es OR exclusivo con el contenido del registro. A continuación, el resultado se desplaza en la dirección del *bit* menos significativo (LSB), con un cero llena en la posición del *bit* más significativo (MSB). El LSB se extrae y se examina. Si el LSB es un 1, el registro es entonces OR exclusivo con un valor predeterminado, fijo. Si el LSB es un 0, no se produce un OR exclusivo. Este proceso se repite hasta que ocho cambios se han realizado.

Después del último cambio (octava), el siguiente *byte* de 8-*bits* es OR exclusivo con el valor actual del registro, y el proceso se repite para ocho desplazamientos más como se describe anteriormente. Los contenidos finales del registro, después de que todos los *bytes* del mensaje han sido aplicados, es el valor de CRC.

Cuando el CRC se anexa al mensaje, el *byte* de orden inferior se añade primero, seguido por el *byte* de orden superior. En la lógica de escalera, la función CKSM calcula un CRC de los contenidos del mensaje.

2.6. Formatos de función Modbus

Los formatos de función Modbus permiten indicar cómo son expresados los valores numéricos entre otros.

2.6.1. Cómo son expresados los valores numéricos

A menos que se especifique lo contrario, los valores numéricos (por ejemplo, direcciones, códigos o datos) se expresan como valores decimales en el texto de esta sección. Se expresan como valores hexadecimales en los campos de mensaje de las cifras.

2.6.2. Las direcciones de datos en mensajes Modbus

En todas las direcciones de datos en mensajes Modbus se hace referencia a cero. La primera aparición de un elemento de datos es direccionada con un cero, que permite tener el patrón de inicio de *byte* entre cada ciclo de repuesta y de pregunta que forma el dato contenido dentro de cada *bit* de información y que es relevante para los datos.

Se ha indicado además que estas combinaciones son homogéneas. Por ejemplo:

- La bobina conocida como "bobina 1" en un controlador programable se aborda como bobina de 0000 en el campo de dirección de un mensaje Modbus.
- Bobina 127 decimal se direcciona como hex bobina 007E (126 decimales).

- El registro Holding 40001 se denomina registro 0000 en el campo de dirección del mensaje. El campo de código de función ya especifica una operación de "registro de mantenimiento". Por lo tanto, la referencia "4XXXX" es implícita.
- El registro de retención 40108 tiene la dirección hexadecimal de registro 006B (107 decimal).

2.6.3. Contenido del campo en los mensajes Modbus

La figura 7 muestra un ejemplo de un mensaje de consulta Modbus. La Figura 8 es un ejemplo de una respuesta normal. Ambos ejemplos muestran el contenido del campo en hexadecimal, y también muestran cómo un mensaje puede ser enmarcado en ASCII o en modo RTU.

La consulta principal es un Registro Holding que lee la petición a la dirección del dispositivo esclavo 06. Las solicitudes de datos del mensaje de los tres registros de retención, 40108 a través de 40110. Tenga en cuenta que el mensaje especifica la dirección del registro inicial como 0107 (006B hex).

La respuesta del esclavo se hace eco del código de la función, lo que indica esto es una respuesta normal. El campo "Cuenta de *bytes*" especifica el número de elementos de *8-bits* de datos que están siendo devueltos.

Se muestra el número de *bytes* de *8-bits* a seguir en los datos, ya sea ASCII o RTU. Con ASCII, este valor es la mitad el número real de caracteres ASCII en los datos. En ASCII, cada valor hexadecimal de *4-bit* requiere un carácter ASCII, por lo tanto, dos caracteres ASCII debe seguir en el mensaje para contener cada elemento *8-bit* de datos.

Tabla III. **Solicitud del Master con trama ASCII/RTU**

SOLICITUD			
Campo nombre	Ejemplo (Hex)	Carácter ASCII	Campo RTU 8-Bit
Cabecera		: (colon)	Nada
Direccionamiento esclavo	06	0 6	0000 0110
Función	03	0 3	0000 0011
Direccionamiento inicial Hi	00	0 0	0000 0000
Direccionamiento inicial Lo	6B	6 B	0110 1011
No. de Registro Hi	00	0 0	0000 0000
No. de Registro Lo	03	0 3	0000 0011
Comprobación de error		LRC (2 Caracteres)	CRC (16 bits)
Trailer		CR LF	Nada

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 28.

Tabla IV. **Solicitud del Esclavo con trama ASCII/RTU**

SOLICITUD			
Campo nombre	Ejemplo	Carácter ASCII	Campo RTU 8-Bit
Cabecera		: (colon)	Nada
Direccionamiento esclavo	06	0 6	0000 0110
Función	03	0 3	0000 0011
Conteo de Byte	06	0 6	0000 0110
Dato Hi	02	0 2	0000 0010
Dato Lo	2B	2 B	0010 1011
Dato Hi	00	0 0	0000 0000
Dato Lo	00	0 0	0000 0000
Comprobación de error		LRC (2 Caracteres)	CRC (16 bits)
Trailer		CR LF	Nada
	Total de Bytes	23	11

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 28.

2.7. Códigos de función admitidos por los controladores

La lista siguiente muestra los códigos de función soportados por los controladores Modicon. Los códigos se enumeran en decimal.

'Y' indica que la función es compatible. "N" indica que no está soportado.

Tabla V. Códigos de función soportados por controladores Modicon

Code	Name	384	484	584	884	M84	984
01	Read Coil Status	Y	Y	Y	Y	Y	Y
02	Read Input Status	Y	Y	Y	Y	Y	Y
03	Read Holding Registers	Y	Y	Y	Y	Y	Y
04	Read Input Registers	Y	Y	Y	Y	Y	Y
05	Force Single Coil	Y	Y	Y	Y	Y	Y
06	Preset Single Register	Y	Y	Y	Y	Y	Y
07	Read Exception Status	Y	Y	Y	Y	Y	Y
08	Diagnostics						
09	Program 484	N	Y	N	N	N	N
10	Poll 484	N	Y	N	N	N	N
11	Fetch Comm. Event Ctr.	Y	N	Y	N	N	Y
12	Fetch Comm. Event Log	Y	N	Y	N	N	Y
13	Program Controller	Y	N	Y	N	N	Y
14	Poll Controller	Y	N	Y	N	N	Y
15	Force Multiple Coils	Y	Y	Y	Y	Y	Y
16	Preset Multiple Registers	Y	Y	Y	Y	Y	Y
17	Report Slave ID	Y	Y	Y	Y	Y	Y
18	Program 884/M84	N	N	N	Y	Y	N
19	Reset Comm. Link	N	N	N	Y	Y	N
20	Read General Reference	N	N	Y	N	N	Y
21	Write General Reference	N	N	Y	N	N	Y
22	Mask Write 4X Register	N	N	N	N	N	(1)
23	Read/Write 4X Registers	N	N	N	N	N	(1)
24	Read FIFO Queue	N	N	N	N	N	(1)

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 31.

El símbolo (1) indica las funciones soportadas solo en 984-785. Es importante mencionar que en este trabajo solo se explicaran los códigos de función útiles para el propósito del tema central.

2.7.1. 01 leer estado de bobina (Read Coil Status)

Descripción: lee el estado ON / OFF de las salidas discretas (0X referencias 0X, “coils” o bobinas) en el esclavo. El Broadcast no es compatible.

Pregunta: el mensaje de petición especifica la bobina de arranque y la cantidad de bobinas para ser leído. Las bobinas son direccionadas a partir de cero: Las bobinas o coils 1-16 son direccionadas como 0-15. Aquí está un ejemplo de una petición de lectura de bobinas 20-56 de dispositivo esclavo 17:

Tabla VI. Pregunta – Read Coil Status

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	01
Direccionamiento Inicial Hi	00
Direccionamiento Inicial Lo	13
No. de puntos Hi	00
No. de puntos Lo	25
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 31.

Respuesta: el estado de la bobina en el mensaje de respuesta se empaqueta como una bobina por *bit* del campo de datos. El estado se indica como: 1 = ON, 0 = OFF. El LSB del primer *byte* de datos contiene la bobina dirigida en la consulta. Las otras bobinas siguen hacia el extremo alto en el

orden de este *byte*, y de "orden inferior al orden superior" en los *bits* subsecuentes. Si la cantidad de vuelta por la bobina no es un múltiplo de ocho, los restantes *bits* del *byte* de datos final se rellenan con ceros (hacia el final del orden superior del *byte*). El campo Cuenta de *bytes* especifica la cantidad de *bytes* de datos completos. He aquí un ejemplo de una respuesta a la pregunta sobre la otra página:

Tabla VII. **Respuesta – Read Coil Status**

RESPUESTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	01
Conteo de <i>Byte</i>	05
Dato (Coil 27-20)	CD
Dato (Coil 35-28)	6B
Dato (Coil 43-36)	B2
Dato (Coil 51-44)	0E
Dato (Coil 56-52)	1B
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 34.

El estado de las bobinas 27-20 se muestra como el valor hexadecimal de *byte* CD, o binario 1100 1101. La bobina 27 es el MSB de este *byte*, y la bobina 20 es el LSB. De izquierda a derecha, el estado de las bobinas de 27 a 20 es: ON-ON-OFF-ON-OFF-ON-OFF-ON. Por convención, los *bits* dentro de un *byte* se muestran con el MSB a la izquierda, y el LSB a la derecha. Así, las bobinas en el primer *byte* son '27 hasta 20', de izquierda a derecha.

El siguiente *byte* tiene bobinas '35 hasta 28', de izquierda a derecha. Como los *bits* se transmiten en serie, que el flujo de LSB a MSB: 20. . . 27, 28. .

. 35, y así sucesivamente. En el último *byte* de datos, el estado de las bobinas 56-52 se muestra como el valor hex 1B *byte*, o binario 0001 1011. La bobina 56 está en la posición de *bit* MSB desde la izquierda, y la bobina 52 es el LSB de este *byte*. El estado de las bobinas de 56 a 52 es: ON-ON-OFF-ON-ON. Observe cómo los tres *bits* restantes (hacia el final del orden superior) se rellenan con ceros.

2.7.2. 02 leer estado de entrada (Read Input Status)

Descripción: lee el estado ON / OFF de las entradas de referencia discretas (1x) en el esclavo. El Broadcast no es compatible.

Pregunta: el mensaje de petición especifica la entrada inicial y la cantidad de insumos para ser leído. Las entradas se direccionan a partir de cero: entradas 1-16 son direccionadas como 0-15. Aquí está un ejemplo de una petición de lectura de entradas 10197-10218 desde el dispositivo esclavo 17:

Tabla VIII. Pregunta – Read Input Status

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	02
Direccionamiento Inicial Hi	00
Direccionamiento Inicial Lo	C4
No. de puntos Hi	00
No. de puntos Lo	16
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 34.

Respuesta: el estado de entrada en el mensaje de respuesta se empaqueta como una entrada por *bit* del campo de datos. El estado se indica como: 1 = ON, 0 = OFF. El LSB del primer *byte* de datos contiene la entrada dirigida en la consulta. Las otras entradas siguen hacia el final del orden superior de este *byte*, y de "orden inferior al orden superior" en los *bytes* siguientes. Si la cantidad de entrada devuelta no es un múltiplo de ocho, los *bits* restantes del *byte* de datos final se rellenan con ceros (hacia el final en el orden superior del *byte*). El campo Cuenta de *Bytes* especifica la cantidad de *bytes* de datos completos. He aquí un ejemplo de una respuesta a la pregunta sobre la otra página:

Tabla IX. **Respuesta – Read Input Status**

RESPUESTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	02
Conteo de <i>Byte</i>	03
Dato (Coil 10204-10197)	AC
Dato (Coil 10212-10205)	DB
Dato (Coil 10218-10213)	35
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 36.

El estado de las entradas 10204-10197 se muestra como el valor de *byte* hex AC, o binario 1010 1100. La entrada de 10204 es el MSB de este *byte*, y la entrada de 10197 es el LSB. De izquierda a derecha, el estado de las entradas 10204 a través de 10197 es: ON-OFF-ON-OFF-ON-OFF-ON-OFF. El estado de las entradas 10218-10213 se muestra como el valor de *byte* 35 hex, binario o

0011 0101. De entrada 10218 se encuentra en la tercera posición de *bit* de la izquierda, y de entrada 10213 es el LSB.

El estado de las entradas 10218 hasta 10213 es: ON-ON-OFF-ON-OFF-ON. Observe cómo los dos *bits* restantes (hacia el extremo del orden superior) se rellenan con ceros.

2.7.3. 03 leer registro de mantenimiento (Read Holding Registers)

Lee el contenido binario del *holding registers* (referencia 4X) en el esclavo. Pregunta: el mensaje de petición especifica el registro inicial y la cantidad de registros que se deben leer. Los registros se direccionan empezando desde cero: los registros 1-16 se direccionan como 0-15. A continuación un ejemplo de una petición de lectura de los registros 40108-40110 del dispositivo esclavo 17:

Tabla X. Pregunta – Read Holding Registers

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	03
Direccionamiento Inicial Hi	00
Direccionamiento Inicial Lo	6B
No. de puntos Hi	00
No. de puntos Lo	03
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 37.

Respuesta: los datos de registro en el mensaje de respuesta se empaquetan como dos *bytes* por registro, con el contenido binario justificado a

la derecha dentro de cada *byte*. Para cada registro, el primer *byte* contiene los *bits* de orden superior y el segundo contiene los *bits* de orden inferior.

Los datos se escanean en el esclavo, a razón de 125 registros por escaneo por los controladores 984-X8X (984 a 685, etc.), y a razón de 32 registros por escaneo para los otros controladores. La respuesta se devuelve cuando los datos son completamente ensamblados.

A continuación un ejemplo de una respuesta a la pregunta sobre la otra página:

Tabla XI. **Respuesta – Read Holding Registers**

RESPUESTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	03
Conteo de <i>Byte</i>	06
Dato Hi (Registro 40108)	02
Dato Lo (Registro 40108)	2B
Dato Hi (Registro 40109)	00
Dato Lo (Registro 40109)	00
Dato Hi (Registro 40110)	00
Dato Lo (Registro 40110)	64
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 36.

El contenido del registro 40108 se muestra como los dos valores de *byte* de 02 2B hex, o decimal 555. Los contenidos de los registros 40109-40110 son 00 00 y 00 64hex, o 0 y 100 decimal. En total el registro proporciona una guía completa en todo el manejo remoto de la red.

2.7.4. 04 registro de lectura de entrada (Read Input Registers)

Lee el contenido binario del registro de entrada (referencia 3X) en el esclavo.

Pregunta: el mensaje de petición especifica el registro inicial y la cantidad de registros que se debe leer. Los registros se direccionan empezando desde cero: el registra 1-16 se direcciona como 0-15.

Tabla XII. Pregunta – Read Input Registers

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	04
Direccionamiento Inicial Hi	00
Direccionamiento Inicial Lo	08
No. de puntos Hi	00
No. de puntos Lo	01
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 39.

Respuesta: los datos de registro en el mensaje de respuesta se empaquetan como dos *bytes* por registro, con el contenido binario justificado a la derecha dentro de cada *byte*. Para cada registro, el primer *byte* contiene los *bits* de orden superior y el segundo contiene los *bits* de orden inferior. Los datos se escanean en el esclavo, a razón de 125 registros por escaneo en los controladores 984-X8X (984 a 685, etc.), y a razón de 32 registros por escaneo para todos los otros controladores. En total cada registro cumple la función básica de mitigación de *byte*.

La respuesta se devuelve cuando los datos son completamente ensamblados. He aquí un ejemplo de una respuesta a la pregunta sobre la otra página:

Tabla XIII. **Respuesta – Read Holding Registers**

RESPUESTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	04
Conteo de <i>Byte</i>	02
Dato Hi (Registro 30009)	00
Dato Lo (Registro 30009)	0A
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 36.

El contenido del registro 30009 se muestran como los dos valores de *byte* de 00 0Ahex, o 10 en decimal.

2.7.5. 05 forzar una bobina simple (Force Single Coil)

Fuerza una bobina simple (Referencia 0x) a ON u OFF. Al emitirse, la función fuerza la misma bobina de referencia en todos los esclavos conectados. La función anulará el controlador de memoria de proteger el estado y el estado desactivar de la bobina. El estado forzado seguirá siendo válido hasta que la lógica del controlador resuelva la siguiente bobina. La bobina permanecerá forzado si no se programa en la lógica del controlador.

Pregunta: el mensaje de petición especifica la referencia de la bobina para ser forzado. Las bobinas se abordan a partir de cero: bobina 1 tiene la dirección

0. El estado ON / OFF se especifica por una constante en el campo de datos de consulta. Un valor de solicitudes FF 00 hex de la bobina a estar en ON. Un valor de 00 00 le pide que sea OFF. Todos los demás valores son ilegales y no afectará a la bobina. Aquí está un ejemplo de una solicitud para forzar bobina 173 en ON en el dispositivo esclavo 17:

Tabla XIV. **Pregunta – Force Single Coil**

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	05
Direccionamiento Inicial Hi	00
Direccionamiento Inicial Lo	AC
No. de puntos Hi	FF
No. de puntos Lo	00
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 41.

Respuesta: la respuesta normal es un eco de la consulta, volvió después de que el estado de la bobina ha sido forzado. He aquí un ejemplo de una respuesta a la pregunta sobre la otra página.

2.7.6. 06 registro simple de preset (Preset Single Register)

Fija un valor en un Single Holding Register (referencia 4X). Al emitirse, la función pre-ajusta la misma referencia de registro en todos los esclavos conectados. La función anulará la memoria del controlador de proteger Estado. Los valores de preselección serán válidos en el registro hasta que la lógica del controlador siguiente resuelve el contenido del registro. El valor del registro se mantendrá si no se ha programado en la lógica del controlador.

Tabla XV. **Respuesta – Read Holding Registers**

RESPUESTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	05
Direccionamiento de la bobina Hi	00
Direccionamiento de la bobina Lo	AC
Forzar Dato Hi	FF
Forzar Dato Lo	00

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 42.

Pregunta: el mensaje de petición especifica la referencia del registro que se desea preseleccionar. Los registros se direccionan empezando desde cero: registro 1 se direcciona como 0. El valor de preselección se especifica en el campo de datos de consulta. Los controladores M84 y 484 usan un valor binario de 10-*bits*, con los seis *bits* de orden superior establecidos con ceros. Todos los otros controladores utilizan valores de 16 *bits*. He aquí un ejemplo de una petición de registro predeterminado 40002-00 03 hex en el dispositivo esclavo 17:

Tabla XVI. **Pregunta – Preset Single Register**

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	06
Direccionamiento Inicial Hi	00
Direccionamiento Inicial Lo	01
Dato de preset Hi	00
Dato de preset Lo	03
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 43.

Respuesta: la respuesta normal es un eco de la consulta, devuelve el contenido del registro después de que se haya ajustado. He aquí un ejemplo de una respuesta a la pregunta sobre la otra página:

Tabla XVII. **Respuesta – Preset Single Register**

RESPUESTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	06
Direccionamiento del Registro Hi	00
Direccionamiento del Registro Lo	01
Dato Preset Hi	00
Dato Preset Lo	03
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 44.

2.7.7. 07 leer estado de excepción (Read Exception Status)

Lee el contenido de ocho bobinas de estado de excepción dentro del controlador esclavo. Algunas bobinas tienen asignaciones predefinidas en los diversos controladores. Otras bobinas puede ser programado por el usuario para almacenar la información sobre el estado del controlador de, por ejemplo, la "máquina de ON / OFF", "encabezado retraído", "seguridad satisfecha", "condiciones de error existe", u otras banderas definidas por el usuario.

Broadcast no es compatible. La función proporciona un método sencillo para acceder a esta información, porque las referencias de bobina de excepción se conoce (no hay referencia la bobina se necesita en la función).Aquí está un

ejemplo de una petición para leer el estado de excepción en el dispositivo esclavo 17:

Tabla XVIII. **Pregunta – ReadException Status**

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	07
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 45.

La respuesta normal contiene el estado de las bobinas de estado de excepción ocho. Las bobinas se empaquetan en un *byte* de datos, con un *bit* por cada bobina. El estado de la bobina de referencia más bajo está contenido en el *bit* menos significativo del *byte*. He aquí un ejemplo de una respuesta a la pregunta sobre la otra página:

Tabla XIX. **Respuesta – Read Exception Status**

PREGUNTA	
Campo Nombre	Ejemplo (Hex)
Direccionamiento del esclavo	11
Función	07
Dato de la bobina	6D
Comprobación de error (LRC o CRC)	-

Fuente: PI-MBUS-300 Rev. E. Modicon Modbus Protocol Reference Guide. p. 46.

3. DISEÑO DEL SISTEMA DE MONITOREO A TRAVÉS DEL PROTOCOLO MODBUS

El monitoreo consiste en el seguimiento, vigilancia y control permanente a las actividades, que dentro de un proceso industrial ayuda a conocer el estado del dispositivo que maneja una maquina o un proceso productivo. El monitoreo de una red de variadores de frecuencia se puede utilizar para manejar una línea de producción o para conocer el comportamiento que ha tenido en los últimos meses. El protocolo Modbus es un protocolo maestro-esclavo en el que un maestro manda los comandos hacia un esclavo y cada esclavo tiene que tener un numero de ID que es el número de nodo que ocupa en la red.

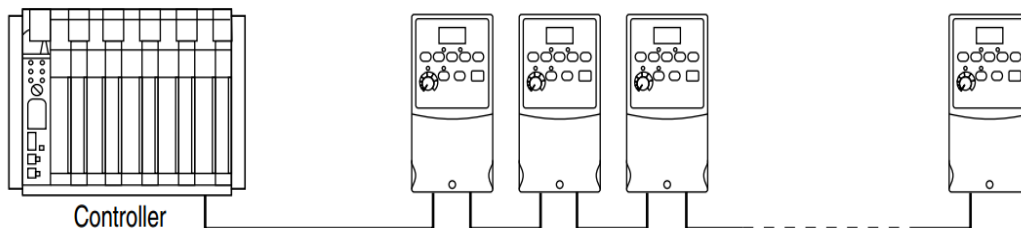
La comunicación entre la red de variadores de frecuencia y el PLC es half-duplex lo que quiere decir que solamente se puede realizar una comunicación a la vez y esta puede ser de transmisión o de recepción. En la figura 34 se muestra un diagrama esquemático del proceso en el que se muestra que el tipo de protocolo que se utiliza para conexión es el protocolo RS-485, que permite conectar hasta 247 nodos a velocidades de 19200 baudios lo cual permite procesar los datos a velocidades aceptables.

3.1. Definición del proceso

El proceso consiste en monitorear una red de variadores de frecuencia usando el protocolo de comunicación industrial Modbus, lo cual se puede hacer fácilmente debido a la comodidad y manejo del protocolo, para luego mostrar los datos a través de una interfaz HMI la cual puede ser una pantalla táctil. Lo que hace popular este protocolo es que trabaja por medio de registros que

pueden ser configurados dependiendo de la utilidad o del dispositivo a controlar. Consta de registros de solo lectura o de escritura, en los que se guardan los datos que se obtienen del variador para ser mostrados en la interfaz.

Figura 34. Diagrama esquemático del sistema



Fuente: BRADLEY, Allen, Adjustable Frequency AC Drive, C-3.

3.1.1. Descripción del diagrama de bloques del proceso

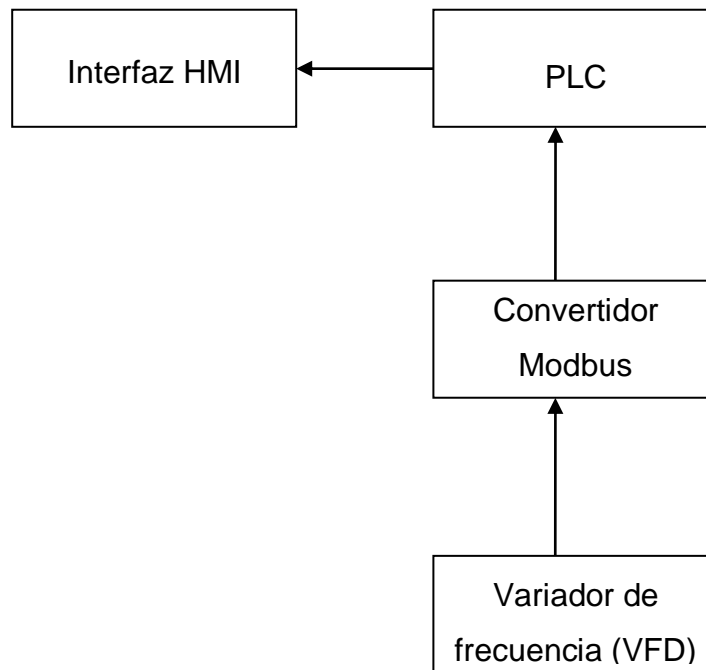
El proceso se centra entre el variador de frecuencia y la interfaz HMI, siendo este controlado por el PLC. En el diagrama de bloques de muestra que el variador de frecuencia se comunica con el PLC por medio de un convertidor Modbus, el cual hace el trabajo de conversión entre el puerto de comunicación que trae el PLC al puerto de comunicación Modbus.

Los variadores de frecuencia constan de registros en donde almacenan la información de todos sus parámetros como el cambio de velocidad, la razón de cambio, el periodo, la frecuencia, el estado del variador (encendido o apagado), etc.

El PLC utiliza el protocolo Modbus para ingresar estos datos en sus propios registros de almacenamiento que pueden ser de lectura o escritura. Los

datos almacenados en los registros del PLC son posteriormente mostrados por la Interfaz HMI que puede ser una pantalla táctil, debido a que puede ser uno o varios variadores de frecuencia, los datos en la pantalla deben de ser parametrizados para tener la opción de cambio a las diferentes pantallas correspondientes a cada variador.

Figura 35. **Diagrama de bloques del sistema**



Fuente: elaboración propia.

3.1.2. **Variables controladas y no controladas**

El variador de frecuencia consta de variables que pueden ser monitoreadas y controladas por el PLC por medio de Modbus, las cuales generalmente se encuentran en un registro de 16 *bits* en el que se puede apagar o encender, cambiarle la dirección de giro, la razón de aceleración, la

fuelle de frecuencia, entre otros. La tabla XX muestra una descripción de las variables controladas que son los registros de lectura o escritura que tiene el variador.

Tabla XX. **Variables controladas – Registros lectura y escritura**

Bits	Descripción
0	1 = Alto, 0 = Continuar
1	1 = Comienzo, 0 = Parar
2	No usado
3	1 = Limpiar error, 0 = No limpiar error
5,4	00 = no usado
	01 = Comando avanzar
	10 = Comando retroceder
	11 = no usado
6	No usado
7	No usado
9,8	00 = no usado
	01 = Habilitar Aceleración a razón de 1
	10 = Habilitar Aceleración a razón de 2
	11 = no usado
11,10	00 = no usado
	01 = Habilitar Desaceleración a razón de 1
	10 = Habilitar Desaceleración a razón de 2
	11 = no usado
14,13,12	000 = no usado
	001 = Fuente de frecuencia, perilla principal
	010 = Fuente de frecuencia, frecuencia interna
	011 = Fuente de frecuencia, registro de lectura y escritura
	100 = Fuente de frecuencia, registro de solo lectura predefinido.
	101 = Fuente de frecuencia, registro de solo lectura predefinido.
	110 = Fuente de frecuencia, registro de solo lectura predefinido.
	111 = Fuente de frecuencia, registro de solo lectura predefinido.
15	No usado

Fuente: BRADLEY, Allen. Adjustable Frequency AC Drive, C-3.

También se tienen variables que son de solo lectura, los cuales se encuentran en los registros de lectura del variador de frecuencia, de los que se puede mencionar si el variador está activo o no activo, ocupado o disponible, comando avanzar o comando retroceder, acelerando o no acelerando, desacelerando o no desacelerando, entre otros. La tabla XXI muestra el registro de 16 *bits*, que tiene las variables no controladas en el variador.

Tabla XXI. **Variables no controladas – Registros de solo lectura**

Bits	Descripción
0	1 = Listo, 0 = No listo
1	1 = Activo, 0 = No Activo
2	1 = Comando Avanzar, 0 = Comando Retro.
3	1 = Rotando hacia adelante, 0 = Rotando de retroceso
4	1 = Acelerando, 0 = No Acelerando
5	1 = Desacelerando, 0 = No Desacelerando
6	1 = Alarma, 0 = No Alarmado
7	1 = En Falla, 0 = Normal
8	1 = En referencia, 0 = No en referencia
9	1 = Referencia controlada por el puerto Comm.
10	1 = Operación cmd. Controlada por el puerto Comm
11	1 = Los parámetros han sido bloqueados
12	Estado de la entrada digital 1
13	Estado de la entrada digital 2
14	No usado
15	No usado

Fuente: BRADLEY, Allen. Adjustable Frequency AC Drive, C-3.

3.1.3. Descripción de la interfaz

La figura 35 muestra una interfaz que se realizó para fines ilustrativos, de los parámetros que se pueden manejar y visualizar desde la HMI. Se muestra

que las variables anteriormente mencionadas pueden ya sea ser cambiadas desde la pantalla o ser mostradas únicamente.

Figura 36. Interfaz del variador de frecuencia



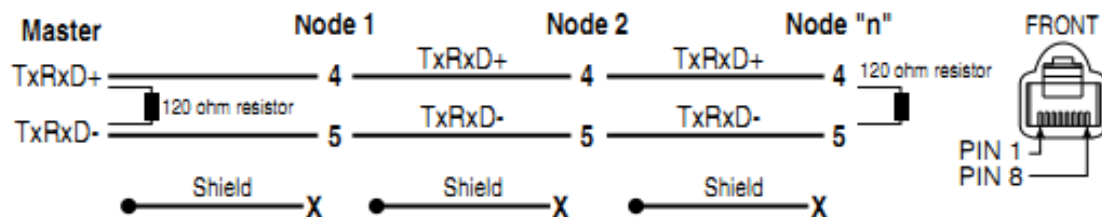
Fuente: elaboración propia.

Se puede cambiar directamente la dirección de giro del variador de frecuencia o cambiar directamente la frecuencia, ver el estado de su temperatura o directamente apagar o encender. En las pantallas HMI se tiene la opción de parametrización en lo que se permite usar una pantalla para todos los variadores que se tienen en la red.

3.2. Explicación de los diagramas de conexión

El cableado de red consiste en un cable blindado de 2 conductores de cable que se conectan en cadena de nodo a nodo, por medio de conectores RJ-45 que son los que se conectan la entrada del variador de frecuencia y un terminal de 4 borneros que se utiliza en el convertidor de Modbus al puerto que trae el PLC que comúnmente es un puerto PS/2.

Figura 37. Diagrama del cableado de la red



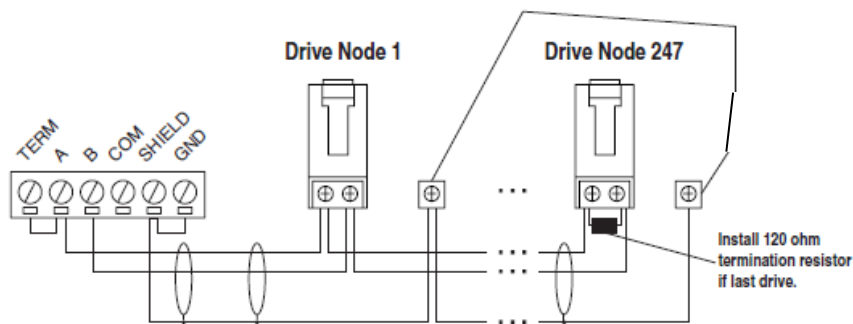
Fuente: Allen-Bradley, Adjustable Frequency AC Drive, C-3.

Sólo los pines 4 y 5 del conector RJ45 deben estar conectados. Las terminaciones de cableado en el controlador maestro puede variar dependiendo de la configuración que el controlador maestro utiliza y "TxRxD +" y "TxRxD-" se muestran a modo de ilustración.

Tenga en cuenta que no existe un estándar para la simbología "+" y "-" de los alambres, y, consecuentemente, los fabricantes de dispositivos Modbus pueden interpretar de manera diferente. Si se tienen problemas con la configuración inicial y en establecer la comunicación, intente intercambiar los dos cables de red en el maestro controlador. En la figura 38 se muestra un diagrama esquemático de conexión del protocolo Modbus con los nodos que

representan los variadores de frecuencia. Las conexiones en el convertidor Modbus solamente involucran la terminal A y B, con su respectivo shield que va conectado con el shield de cada variador de frecuencia.

Figura 38. Diagrama esquemático de conexión



Fuente: Allen-Bradley, Modbus RTU Contrl, p. 4.

3.3. Registros Modbus del variador

De los registros anteriormente mencionados en el capítulo 2, se referirá a los que se utilizan para la realización de este proyecto, los cuales son los siguientes:

- 03 *Read Holding Registers*
- 06 *Preset Single Register*
- 16 (10 Hexadecimal) *Preset Multiple Registers*

- 03 *Read Holding Registers*: este registro representa el código de función Modbus que se utilizara para leer remotamente al variador y representa una palabra de 16 *bits*, que puede posteriormente ser almacenada en los registros del PLC.

- *06 Preset Single Register*: este es el código de función Modbus que permitirá escribir en un simple registro de almacenamiento de manera remota al variador y es representada por una palabra de 16 *bits*, al ingresar este código de función el PLC procederá a “escribir” en el variador los datos que se encuentran en el registro especificado previamente por la instrucción enviada.
- *16 Preset Multiple Registers*: este código de función Modbus permitirá “escribir” remotamente al variador varios registros a la vez, en la instrucción que se envía al variador se debe especificar los registros del PLC que contienen la información que se va a “escribir” en el variador y permitirá escribir desde 1 hasta 120 registros contiguos en el variador de frecuencia.

3.4. Parámetros y comandos útiles para el desarrollo del proceso

En el PLC se utiliza el lenguaje de programación Ladder (escalera), para la configuración de Modbus, los comandos que se utilizan para la configuración son los siguientes:

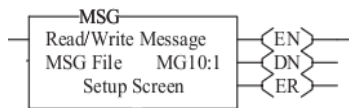


Utiliza variables lógicas binarias del PLC para evaluar si activar o no las salidas en determinado “escalón”. En la programación Ladder, estas variables se representan por contactos, que justamente pueden estar en solo dos estados: abierto o cerrado. Los contactos se representan con la letra “E” y dos números que indican el módulo al cual pertenecen y la bornera a la cual están asociados.



ASIGNACION DE SALIDA

Las salidas de un programa Ladder son equivalentes a las cargas (bobinas de relés, lámparas, etc.) en un circuito eléctrico, se identifica con la letra “S”, “A” u otra letra, dependiendo de los fabricantes, y dos números que indican el modulo al cual pertenece y la bornera a la cual están asociados.



INSTRUCCIONES DE ENTRADA Y SALIDA

Las instrucciones de entrada y salida se utilizaran para enviar y recibir mensajes de Modbus desde el PLC hacia el variador de frecuencia, en los PLC estos instrucciones de mensaje se representan con las letras “MSG” y permite configurar el comando de función que va a utilizar por ejemplo si es lectura o escritura y el registro en donde se va a leer y escribir en el PLC y en el variador de frecuencia.

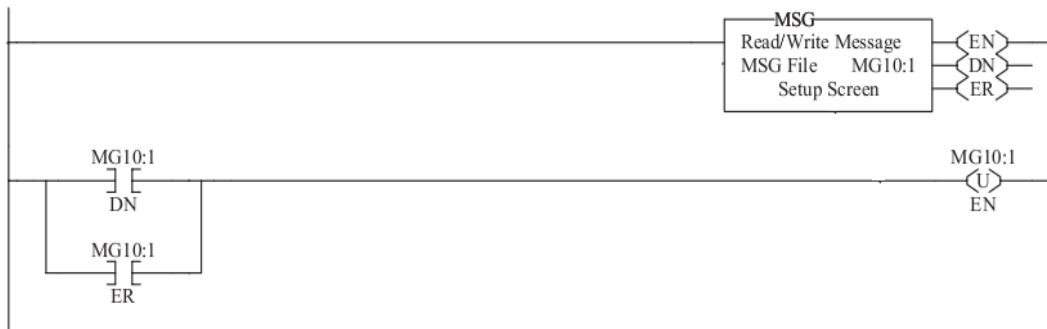
Entre otros mensajes también indica si el mensaje se envió con éxito o si se produjo algún error en él envió. Este tipo de configuración también es dependiente del tipo y modelo de PLC ya que algunos PLC recientes crean directamente el registro de Modbus haciendo un mapeo de todas las bobinas y directamente se ingresa el valor que se ingresara en el variador de frecuencia. En este trabajo se hace referencia a un modelo general de PLC y de variador de frecuencia obteniendo una base para poder manejar cualquier modelo y marca de equipo.

Entre los principales parámetros que se deben programar en Ladder y que representa los conjuntos básicos para el proceso que se llevarán a cabo se tienen los siguientes:

- CHANNEL: para la comunicación entre PLC y PLC elige la opción 1(Integral), para este caso se elegirá la opción 0(Integral) para comunicación entre PLC y el variador de frecuencia.
- MODBUS COMMAND: se especifica una de las opciones posibles, acá se indica si se desea 03 lectura, 06 escritura o 16 escritura de registro múltiple.
- DATA TABLE ADDRESS: acá se coloca el tag donde se ingresara el dato de escritura del PLC al variador o donde se leerá el dato de escritura del variador al PLC.
- SIZE EN ELEMENTS: en esta opción se deberá de colocar un 1 que indica que se va a leer o escribir 1 palabra es decir los 16 *bits* de la variable que se indica.
- MESSAGE TIMEOUT: es el tiempo que se le quiere indicar para que el PLC espere una respuesta del variador.
- MB DATA ADDRESS: se coloca el número de registro al que se quiere encontrar en el variador.
- SLAVE NODE ADDRESS: se ingresa el número del nodo del variador al que se quiere ingresar la función.

Cada una de estos parámetros se deben de tomar en cuenta tanto para las variables que permiten el control interno como para las variables que permiten el control externo, debido a que son parte de las exigencias que complementa el autómeta.

Figura 39. Programa Ladder para Mensaje Modbus



Fuente: BRADLEY, Allen. Modbus RTU Contrl, p. 9.

3.5. Medición de resultados

Los mensajes Modbus se procesan al final del escaneo del controlador. En el peor de los casos el retraso es un tiempo de escaneo, que se produce si el controlador ha empezado una nueva búsqueda. El retraso medio es de 0,5 veces el tiempo de escaneo. El tiempo asignado para el servicio de los puertos Modbus al final del escaneo del controlador (antes de comenzar un nuevo ciclo) depende del modelo de controlador.

En Modbus se puede estimar el tiempo de transmisión de los *bits*. Las funciones Modbus 1 a 4 y 15 a 16 permiten que el maestro solicite más datos que pueden ser procesados durante el tiempo asignado para el servicio del Modbus esclavo.

Si el esclavo no puede procesar todos los datos, se va a un buffer y los datos son procesados en el final de los análisis subsiguientes. El cálculo del tiempo LRC es inferior a 1 ms y el tiempo calculado de CRC es de aproximadamente 0,3 ms por cada 8 *bits* de datos que se devuelven en la respuesta. Modbus tiene una ecuación para estimar el tiempo de transmisión:

$$Tiempo (ms) = \frac{1000 * (Conteo de Carácter) * (bits por carácter)}{Baud Rate} \quad (3.1)$$

En donde el conteo de carácter es un campo de la trama de *bits* enviada y el Baud Rate es la cantidad de *bits* transmitidas por segundo que en Modbus no es especificada, las velocidades típicas son 9600 o 19200. La tabla XXII muestra los datos que se tomaron al hacer él envió para lectura y escritura tomando en cuenta una velocidad de *bit* (Baud Rate) igual a 9600 que se transmite sobre una conexión serial.

Tabla XXII. **Tiempo de transmisión para algunos datos calculados**

Tipo de función	Conteo de Carácter	Bit por carácter	Tiempo de transmisión (ms)
Read Coil Status	5	8	4.17
Read Input Status	3	8	2.5
Read Holding Registers	6	8	5
Read Input Registers	2	8	1.67

Fuente: elaboración propia.

El tiempo de transmisión para los datos se calcularon en base a las medidas que el variador mostro, tanto para el tiempo de aceleración de subida como para el de desaceleración secuencial en tiempo mixto y en tiempo compartido. Por lo que se puede decir de todo este análisis que son correctos cada uno de los tiempos, porque al tomar el tiempo de cada registro individual y combinar junto con el tiempo que el variador toma para ascender dentro de la gráfica de estabilidad, vemos que el tiempo no es parte de un estado crítico dentro de todo el proceso y que su evolución dentro del mismo es lo equivalente que se esperaría que operara tomando en cuenta que las distancias son largas y que en ningún momento exista interferencia.

CONCLUSIONES

1. Las principales variables para el control a largas distancias del dispositivo son: la frecuencia de salida, el estado, encendido/apagado, acelerado/decelerado del VFD y las pantallas táctiles permiten visualizar dichas variables.
2. El protocolo Modbus funciona con registros de lectura y escritura que proporcionan un mapeo para obtener de manera remota las variables del VFD.
3. El sistema de monitoreo y control de un VFD consiste en la conexión entre el PLC que actúa como el controlador junto con el VFD que actúa como el dispositivo controlado, comunicándose ambos por medio de Modbus y posteriormente enviando los datos hacia una pantalla táctil que los despliega de manera remota.

RECOMENDACIONES

1. Dentro de las variables involucradas en este proceso de control existen algunas que no son completamente necesarias como: la transmisión, la recepción, comunicación de arranque y entrada de paro del VFD, por lo que es podrían obviarse para evitar el uso innecesario del ancho de banda.
2. Es preferente usar Modbus como protocolo de comunicación para el sistema de control de VFD debido a que la mayoría de dispositivos en el mercado industrial pueden ser manejados a través de este protocolo, debido a su versatilidad, fácil acoplamiento y facilidad de uso.
3. Como una buena práctica se debe de verificar con base en una medición de resultados que el sistema de monitoreo y control del VFD despliegue los datos correctos, partiendo desde la comunicación Modbus entre el PLC y el VFD y terminando con los datos desplegados en la pantalla táctil.

BIBLIOGRAFÍA

1. FARDO, Stephen W.; PATRICK, Dale R. *Industrial process control system*. 2a. ed. Estados Unidos de America: Collage Park, 1997. 460 p.
2. *Generación de altas tensiones continuas* [en línea]. [ref. 25 octubre de 2006]. Disponible en web: <http://www.textoscientificos.com/fisica/alta-tension/generacion-tensiones-continuas>.
3. *Industrial automation training* [en línea]. Disponible en web: <http://www.pacontrol.com/download/Process%20Control%20Fundamentals.pdf>. [Consulta: 17 de febrero 2013].
4. *Introducción a HMI* [en línea]. [ref. 2 de julio de 2008]. Disponible en web: <http://iaci.unq.edu.ar/materias/laboratorio2/HMI/Introduccion%20HMI.pdf>.
5. Modicon. *Modbus Protocol Reference Guide*. Estados Unidos de América: Modicon, 1996. 121 p.
6. *Motion Control. El convertidor de frecuencia (II): Control Escalar* [en línea]. [ref. 27 de febrero de 2010. Disponible en web: <http://www.infopl.net/blog4/2010/02/27/el-convertidor-de-frecuencia-ii-control-escalar/>.

7. NAVARRO, Danilo A. *Control lógico programable. Cursos tutorial* [en línea]. [ref. julio 2001]. Disponible en web: <http://es.scribd.com/doc/69292441/Controlador-Logico-Programable-PLC-Curso-Tutorial>.
8. *Oposinet Temario Tecnología temas oposiciones enseñanza* [en línea]. [ref. 5 de julio de 2012]. Disponible en web: http://www.oposinet.com/tecnologia/temas/otema58_1.htm.
9. PÉREZ LÓPEZ, Eric. *Protocolo RS485* [en línea]. Disponible en web: <http://www.i-micro.com/pdf/articulos/rs-485.pdf>. [Consulta: 11 de marzo de 2013].
10. RINCÓN, Ernesto. *Motores y variadores de frecuencia* [en línea]. [ref. 26 de julio de 2012]. Disponible en web: http://ernestrincon.blogspot.com/2011_07_01_archive.html.
11. Schneider Electric. *Manual electrotécnico*. España: Schneider Electric, 1999. 285 p.
12. *Sistemas de bus RS485* [en línea]. [ref. 14 de noviembre de 2006]. Disponible en web: <http://www.etitudela.com/entrenador-comunicaciones/downloads/introduccionrs485.pdf>.
13. WISNIEWSKY, David M. *Modbus RTU Control*. Estados Unidos de América: Allen-Bradley, 1997.

