## **CAPITULO I**

## INTRODUCCIÓN.

En este capítulo se expone los conceptos básicos de sistemas SCADA, a continuación se describe las diferentes etapas que están implícitas en este proyecto como se muestra en la siguiente Figura Nº. 1.1



Figura Nº. 1.1 Etapas del Proyecto

De la introducción a los sistemas SCADA se procede a explicar el software que se utilizará en este proyecto, luego el protocolo y por ultimo se explicará la programación del Controlador Lógico Programable (PLC).

### **1.1 SISTEMAS SCADA**

### 1.1.1 DEFINICIÓN DE SISTEMA SCADA.

SCADA viene de las siglas de "Supervisory Control And Data Adquisition", es decir: adquisición de datos y control de supervisión.<sup>1</sup>

Un sistema SCADA está basado en un conjunto de computadoras que permite supervisar y controlar la producción a distancia en una instalación de cualquier tipo, mediante la comunicación con los módulos de campo (controladores autónomos,

<sup>&</sup>lt;sup>1</sup> www.scadas.com; Sistemas SCADA.

autómatas programables, etc.) controlando el proceso de forma automática desde la pantalla del ordenador.

Recibe toda la información que se origina en el proceso productivo.

En estos sistemas el computador efectúa tareas de supervisión y gestión de alarmas, así como tratamiento de datos y control de procesos.

La comunicación con los equipos anteriormente mencionados se realiza mediante buses especiales o redes LAN.

Ejecutándose normalmente en tiempo real, facilitando al operador de planta la posibilidad de supervisar y controlar los procesos.

#### 1.1.2 FUNCIONES DE LOS SISTEMAS SCADA.

Las principales funciones realizadas por un sistema SCADA son las siguientes:

- a) Recaudar, almacenar y presentar la información, en forma continua y confiable, correspondiente a las variables de campo, mediciones, alarmas, etc.
- b) Visualizar y ejecutar acciones de control ejecutado por el operador, tales como: abrir o cerrar válvulas, arrancar o parar bombas, etc.
- c) Indicar al operador de variaciones producidos y detectados en la planta, las que no se consideren normales (alarmas), como también los cambios que se producen en la operación diaria de la planta (eventos). Guardado estas variaciones producidas en el sistema para su posterior análisis.
- d) Aplicaciones basadas en la información recopilada por el sistema, como: reportes, gráficos de tendencia, historia de variables, cálculos, predicciones, detección de fugas, etc.

## 1.1.3 CONDICIÓN DE UN SISTEMA SCADA.

Para el manejo de la producción por medio de un sistema SCADA es necesario evaluar la instalación, el proceso a controlar debe cumplir las siguientes características:

- a) El número de variables del proceso que se necesita monitorear.
- b) La información que se obtiene del proceso se debe realizar en tiempo real, porque si se producen cambios o variaciones el mismo debe ser analizado.
- c) Facilita el manejo y las operaciones de la planta.
- d) Los datos recopilados por el sistema permitirá la toma de decisiones, tanto gerenciales como operativas.
- e) Los diferentes beneficios obtenidos en el proceso justifican la inversión en un sistema SCADA. Estos beneficios pueden reflejarse como aumento de la eficiencia de la producción, niveles de seguridad, etc.
- f) Estos sistemas permiten que la mayoría de las acciones de control sean iniciadas por un operador, como también se requiere de un Sistema de Control Automático, el cual lo puede constituir un Sistema de Control Distribuido, PLC's, Controladores a Lazo Cerrado o una combinación de ellos.

## 1.1.4 CARACTERÍSTICAS PRINCIPALES DE LOS SISTEMAS SCADA

- Supervisión en tiempo real
- Adquisición de información local y remota.
- Centralización de la información.
- Facilidad de control y de control.
- Flexibilidad de adaptarse a cambios físicos en el proceso.
- Facilidad de manejo de datos históricos.

- Manufactura flexible.
- Integración con sistemas operativos.
- Sistema abierto.

### 1.1.5 BENEFICIOS DE LOS SISTEMAS SCADA.

- Reducción de los costos de producción y operación.
- Aumento de producción.
- Diversificación de la producción.
- Reducción de los costos de mantenimiento.
- Mejoramiento de la coordinación con el área de mantenimiento.
- Se dispone de información bien precisa para efectos de estudio, análisis y estadística.

# 1.1.6 PASOS PARA EL DISEÑO Y OPTIMIZACIÓN DE LOS SISTEMAS SCADA

Para lo qué debería ser usado un SCADA

- Para presentar y hacer un seguimiento de un proceso
- Para enviar comandos al PLC
- Para gestionar alarmas, tendencias e informes
- Para interactuar con otras aplicaciones



Comandos dirigidos por eventos

Control en tiempo real

# 1.1.6.1 DISEÑO DE LA APLICACIÓN

### Paso 1

- Diseñar el esquema y jerarquía de los gráficos
- Decidir el nivel de detalle para cada pantalla



### Paso 2

- Crear una Matriz de Estados
  - Usar palabras para transmitir datos, no bits
- Descomponer los dispositivos en objetos comunes
  - Motores
  - Válvulas ON / OFF
  - Lazos de control

- ....

### Paso 3

- Definir que puntos serán usados para seguimiento en tendencias y alarmas.
  - Los puntos para tendencia y alarmas necesitan refrescarse continuamente.
  - Los puntos que son visualizados en una página gráfica, sólo necesitan refrescarse cuando estén visualizados.

### paso 4

- Definir los puntos de E / S.
  - Usar arrays para agrupar los puntos que contengan información similar.
  - Usar el tipo y frecuencia de actualización adecuada para cada punto.

### Paso 5

- Utilizar definición de alias.
  - Convertir arrays en nombres descriptivos utilizando esta característica.
  - Utilizar nombres de variables y alias significativos.

## Paso 6

- Direccionamiento indirecto
  - Usar direccionamiento indirecto cuando sea posible
  - Se ajusta bien para los mensajes emergentes

# **1.2 INTOUCH**<sup>2</sup>

Con InTouch, un diseñador puede crear aplicaciones con características completas y potentes que utilizan las características de Windows al máximo, incluyendo el Intercambio dinámico de datos (DDE).

Enlace de objetos e incrustaciones (OLE), gráficos y más. InTouch también se puede ampliar agregando asistentes personalizados, objetos genéricos y extensiones de script.

Las aplicaciones InTouch abarcan una multitud de mercados verticales incluyendo procesamiento de alimentos, semiconductores, petróleo y gas, automotores, químicos, farmacéuticos, pulpa y papel, transporte, servicios públicos y otros.

La manera más rápida y fácil de crear aplicaciones de interfaz entre hombre y computadora (MMI) para los sistemas operativos de Microsoft Windows

El software de InTouch consta de dos componentes principales, **WindowMaker** y **WindowViewer** y de seis programas de utilidad/diagnóstico.

<sup>&</sup>lt;sup>2</sup> Guía del usuario ITMAN\_R, Enero, 1997 Wonderware Corporation

**WindowMaker** es el entorno de desarrollo donde los gráficos orientados a objetos se utilizan para crear ventanas de visualización animada y sensible al contacto. Estas ventanas de visualización se pueden conectar a sistemas industriales de E/S y otras aplicaciones Windows de Microsoft.

**WindowViewer** es el entorno de ejecución que se utiliza para mostrar las ventanas de gráficos creadas en WindowMaker.

## 1.2.1 CARACTERÍSTICAS ESPECIALES DE INTOUCH

InTouch incluye las siguientes características:

### • Sistema de alarmas distribuidas

Proporcionando a los operadores la capacidad de visualizar y reconocer simultáneamente información de alarmas desde múltiples ubicaciones remotas.

### • Historial distribuido

Permite especificar de forma dinámica una fuente de datos de archivos históricos diferente para cada pluma de un gráfico de tendencia.

## • Conversión de resolución dinámica

Ahora puede desarrollar aplicaciones con una resolución de pantalla y ejecutarlas en otra sin afectar a la aplicación original.

Las aplicaciones también se pueden ejecutar con una resolución definida por el usuario, en lugar de la resolución de visualización.

### • Direccionamiento de referencia dinámico

Las referencias de fuente de datos se pueden modificar de forma dinámica para direccionar múltiples fuentes de datos con una única etiqueta.

### • Desarrollo de aplicación en red

Las nuevas características de desarrollo remoto dan cabida a grandes instalaciones multinodo, incluyendo la actualización de todos los nodos de una red desde una única estación de desarrollo.

## • FactoryFocus

Permite a los administradores y supervisores visualizar un proceso continuo de aplicación MMI en tiempo real. La seguridad del sistema se incrementa con la capacidad de sólo visualización, ya que no se puede cambiar ningún dato. No es necesario realizar modificaciones en las aplicaciones de InTouch para utilizar InTouch FactoryFocus.

## **1.2.2 REQUISITOS DEL SISTEMA**

Para ejecutar InTouch, le recomendamos el siguiente hardware y software:

- Cualquier máquina IBM compatible con un procesador 80486 o superior.
- Por lo menos 100MB de espacio libre en disco duro.
- Por lo menos 16MB de memoria de acceso aleatorio (RAM).
- Adaptador de pantalla SVGA.
- Puerto paralelo de la impresora.
- Dispositivo de puntero (por ejemplo, mouse, trackball o joystick).
- Sistemas operativos Windows de Microsoft (3.1 o posterior), Windows para Trabajo en Grupo (3.11 o posterior), Windows 95 o Windows NT.
- Opcional: Para implementar la funcionalidad distribuida de InTouch, Wonderware NetDDE debe estar instalado y operando.

# 1.3. PROTOCOLO DE COMUNICACIONES<sup>3</sup>

La designación Modbus Modicon corresponde a una marca registrada por Gould Inc.

<sup>&</sup>lt;sup>3</sup> Modbus, modicon

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, cuyas características esenciales son las que se detallan a continuación.

## 1.3.1 MEDIO FÍSICO

El medio físico de conexión puede ser un bus semidúplex (half duplex) (RS-485 o fibra óptica) o dúplex (full duplex) (RS-422, BC 0-20mA o fibra óptica).

La comunicación es asíncrona y las velocidades de transmisión previstas van desde los 75 baudios a 19.200 baudios. La máxima distancia entre estaciones depende del nivel físico, pudiendo alcanzar hasta 1200 m sin repetidores.

### **1.3.2 PROTOCOLO**

La codificación de datos dentro de la trama puede hacerse en modo ASCII o puramente binario, según el estándar RTU (Remote Transmission Unit).

En cualquiera de los dos casos, cada mensaje obedece a una trama que contiene cuatro campos principales, según se muestra en la Figura N°.1.2. La única diferencia estriba en que la trama ASCII incluye un carácter de encabezamiento («:»=3AH) y los caracteres CR y LF al final del mensaje.

Pueden existir también diferencias en la forma de calcular el CRC, puesto que el formato RTU emplea una fórmula polinómica en vez de la simple suma en módulo 16.

Con independencia de estos pequeños detalles, a continuación se da una breve descripción de cada uno de los campos del mensaje:



Figura. Nº. 1.2 Trama genérica del mensaje según el código empleado

## 1.3.3 NÚMERO DE ESCLAVO (1 Byte):

Permite direccionar un máximo de 63 esclavos con direcciones que van del 01H hasta 3FH. El número 00H se reserva para los mensajes difundidos.

# 1.3.4 CÓDIGO DE OPERACIÓN O FUNCIÓN (1 Byte):

Cada función permite transmitir datos u órdenes al esclavo. Existen dos tipos básicos de órdenes:

• Ordenes de lectura/escritura de datos en los registros o en la memoria del esclavo.

• Ordenes de control del esclavo y el propio sistema de comunicaciones (RUN/STOP, carga y descarga de programas, verificación de contadores de intercambio, etc.)

La Tabla 1 muestra la lista de funciones disponibles en el protocolo MODBUS con sus correspondientes códigos de operación.

## 1.3.5 CAMPO DE SUBFUNCIONES/DATOS (N Bytes):

Este campo suele contener, en primer lugar, los parámetros necesarios para ejecutar la función indicada por el byte anterior. Estos parámetros podrán ser códigos de subfunciones en el caso de órdenes de control (función 00H) o direcciones del primer

bit o byte, número de bits o palabras a leer o escribir, valor del bit o palabra en caso de escritura, etc.

### **1.3.6 PALABRA DE CONTROL DE ERRORES (2 Bytes):**

En código ASCII, esta palabra es simplemente la suma de comprobación ('checksum') del mensaje en módulo 16 expresado en ASCII. En el caso de codificación RTU el CRC se calcula con una fórmula polinómica según el algoritmo mostrado en la Figura Nº 1.3.

| lunción | Código          | Tarea                                       |
|---------|-----------------|---|
| 0       | 00 <sub>H</sub> | Control de estaciones esclavas              |
| 1       | 01 <sub>H</sub> | Lectura de n bits de salida o internos      |
| 2       | 02 <sub>H</sub> | Lectura de <i>n</i> bits de entradas        |
| 3       | 03 <sub>H</sub> | Lectura de n palabras de salidas o internos |
| 4       | 04 <sub>H</sub> | Lectura de n palabras de entradas           |
| 5       | 05 <sub>H</sub> | Escritura de un bit                         |
| 6       | 06 <sub>H</sub> | Escritura de una palabra                    |
| 7       | 07 <sub>H</sub> | Lectura rápida de 8 bits                    |
| 8       | 08 <sub>H</sub> | Control de contadores de diagnósticos       |
|         |                 | número 1 a 8                                |
| 9       | 09 <sub>H</sub> | No utilizado                                |
| 10      | 0A <sub>H</sub> | No utilizado                                |
| 11      | 0B H            | Control del contador de diagnósticos        |
|         |                 | número 9                                    |
| 12      | 0C <sub>H</sub> | No utilizado                                |
| 13      | 0D <sub>H</sub> | No utilizado                                |
| 14      | 0E <sub>H</sub> | No utilizado                                |
| 15      | 0F <sub>H</sub> | Escritura de n bits                         |
| 16      | 10 <sub>H</sub> | Escritura de n palabras                     |



Tabla 1. Funciones básicas y códigos de operación

Figura Nº. 1.3. Cálculo del CRC codificación RTU

### 1.3.7 DESCRIPCIÓN DE LAS FUNCIONES DEL PROTOCOLO

## 1.3.7.1 FUNCIÓN 0:

Esta función permite ejecutar órdenes de control, tales como marcha, paro, carga y lectura de programas de usuario del autómata. Para codificar cada una de las citadas órdenes se emplean los cuatro primeros bytes del campo de datos. La trama resultante es la representada en la Figura Nº. 1. 3 y la interpretación de los códigos de subfunción se especifica en la tabla 2.

En caso de las órdenes de marcha y paro, el campo de «información» de la trama representada en la Figura Nº. 1.4 está vacío y, por tanto, el mensaje se compone simplemente de 6 bytes de función más 2 bytes de CRC. La respuesta del esclavo a estas órdenes es un mensaje idéntico al enviado por el maestro. Cabe señalar, además, que después de un paro el autómata sólo acepta ejecutar subfunciones de la función 00H.

| N°<br>Esclavo         | 004 | Código<br>Subfunción | Datos<br>Subfunción | información | CRC(16) |
|-----------------------|-----|----------------------|---------------------|-------------|---------|
| (00-3F <sub>H</sub> ) |     | SF0 SF1              | D0 D1               |             | HL      |

Figura Nº. 1.4. Trama genérica de las subfunciones de control de esclavos (cód. función 00H)

| Có<br>subfu<br>SF0 | tigo<br>nción<br>SF1 | Da<br>subfu<br>D0 | tos<br>nción<br>D1 | Tarea  |
|--------------------|----------------------|-------------------|--------------------|--|
| 00 <sub>H</sub>    | 00 <sub>H</sub>      | 00 <sub>H</sub>   | 00 <sub>H</sub>    | Paro del esclavo sin inicializar   |
| 00 <sub>H</sub>    | 01 <sub>H</sub>      | 00 <sub>H</sub>   | 00 <sub>H</sub>    | Marcha del esclavo sin inicializar   |
| 00 <sub>H</sub>    | 02 <sub>H</sub>      | 00 <sub>H</sub>   | 00 <sub>H</sub>    | Marcha e inicialización del esclavo  |
| 00 <sub>H</sub>    | 03 <sub>H</sub>      | 00 <sub>H</sub>   | XXH                | Lectura de la secuencia XX de programa de<br>usuario en el esclavo   |
| 00 <sub>H</sub>    | 04 <sub>H</sub>      | YYH               | XX <sub>H</sub>    | Carga de una secuencia de programa de usuario<br>en el esclavo<br>Petición: YY = secuencia a cargar, XX=<br>próxima secuencia<br>Respuesta: XX= código error, YY= 00 |

Tabla 2. Subfunciones correspondientes a la función =00H

#### 1.3.7.2 FUNCIONES 1 Y 2:

Lectura de bits del autómata. La trama es la indicada en la Figura N°. 1.5 La forma de direccionamiento de los bits es a base de dar la dirección de la palabra que los contiene y luego la posición del bit. Obsérvese también que la respuesta es dada siempre en octetos completos.



Figura Nº. 1.5 Petición y respuesta de la función: Lectura de bits (01H, 02H)

Funciones 3 y 4:

Lectura de palabras del autómata. La trama es la indicada en la Figura Nº. 1.6 Obsérvese que la petición indica el número de palabras a leer, mientras que en la respuesta se indica el número de octetos leídos.

| Petición              | del maes        | tro              |          |             |     |     |
|-----------------------|-----------------|------------------|----------|-------------|-----|-----|
| Nº                    | 03 <sub>M</sub> | Dirección        | Nº de    |             |     |     |
| Esclavo               | 0               | 1º Palabra       | Palabras | CRC         |     |     |
| (00-3F <sub>H</sub> ) | 04 <sub>H</sub> | PP PP            | NN NN    | ΗL          |     |     |
| PPPP = Di             | rección de      | la palabra (hex) |          |             |     |     |
|                       |                 |                  |          |             |     |     |
| Respuest              | ta del esc      | lavo             |          |             |     |     |
| N°                    | 03н             | Nº Octetos       | 1*       | Otras Palab | ras |     |
| Esclavo               | 0               | leidos           | Palabra  | Hasta máx.  | 128 | CRC |
| (00-3F <sub>H</sub> ) | 04 <sub>M</sub> | NN NN            | ΗL       | H L,H L,HL, |     | HL  |

Figura Nº. 1.6 Petición y respuesta de la función: Lectura de palabras (03H,04H)

## 1.3.7.3 FUNCIÓN 5:

Escritura de un bit. La trama es la indiada en la Figura N°. 1.7 El direccionamiento del bit se efectúa tal como se ha indicado para las funciones 1 y 2.

| Petición                               | del maes        | tro                                     |                 |             |                  |                           |
|--|-----------------|---|-----------------|-------------|------------------|---------------------------|
| Nº<br>Esclavo<br>(00-3F <sub>H</sub> ) | 05 <sub>H</sub> | Dirección<br>Bit<br>PP PB               | XX <sub>H</sub> | 00,4        | CRC<br>H L       |                           |
| PPP = Dire                             | ección de l     | a palabra (hex),                        | B= Direcc       | ión del bit | dentro de la pal | abra 0 a F <sub>H</sub> . |
|  |                 |   |                 |             |                  |                           |
| Respuest                               | a del esc       | lavo                                    |                 |             |                  |                           |
| Nº<br>Esclavo<br>(00-3F <sub>H</sub> ) | 05 <sub>H</sub> | Dirección<br>Bit<br>PP PB               | XX <sub>H</sub> | 00,4        | CRC<br>H L       |                           |
| $XX_H = 00H$                           | I para bit =    | = 0 y XX <sub>H</sub> = FF <sub>H</sub> | para bit =      | 1           |                  |                           |

Figura Nº. 1.7 Petición y respuesta de la función: Escritura de un bit (05H)

# 1.3.7.4 FUNCIÓN 6:

Escritura de una palabra. La trama es la indicada en la Figura Nº. 1.8

| Petición del maestro                   |                       |                               |                            |            |  |  |  |  |  |  |
|--|-----------------------|-------------------------------|----------------------------|------------|--|--|--|--|--|--|
| Nº<br>Esclavo<br>(00-3F <sub>H</sub> ) | 06 <sub>H</sub>       | Dirección<br>Palabra<br>PP PP | Valor<br>Palabra<br>DD DD  | CRC<br>H L |  |  |  |  |  |  |
| Respuest                               | Respuesta del esclavo |                               |                            |            |  |  |  |  |  |  |
| N°<br>Esclavo<br>(00-3F <sub>H</sub> ) | 06н                   | Dirección<br>Palabra<br>PP PP | Nº de<br>Palabras<br>DD DD | CRC<br>H L |  |  |  |  |  |  |

Figura Nº. 1.8. Petición y respuesta de la función: Escritura de una palabra (06H)

# 1.3.7.5 FUNCIÓN 7:

Petición de lectura rápida de un octeto. La trama es la mostrada en la Figura N°. 1.9 Obsérvese que la petición no tiene campo de dirección, esto es debido a que el octeto legible por esta función es fijo en cada esclavo y viene fijado en su configuración.



Figura Nº. 1.9 Petición y respuesta de la función: Lectura rápida de un octeto (07H)

## 1.3.7.6 FUNCIÓN 8:

Petición del contenido y control de los 8 primeros contadores de diagnóstico de un esclavo (véase tabla 3). Las tramas de petición y respuesta pueden verse en la Figura N°. 1.10

| Petición del maestro                   |                 |                                 |                             |            |  |  |  |  |  |  |  |
|--|-----------------|---------------------------------|-----------------------------|------------|--|--|--|--|--|--|--|
| Nº<br>Esclavo<br>(00-3F <sub>i</sub> ) | 08 <sub>H</sub> | Código<br>Subfunción<br>SF0 SF1 | Dato<br>Subfunción<br>D0 D1 | CRC<br>H L |  |  |  |  |  |  |  |
| Respuesta del esclavo                  |                 |                                 |                             |            |  |  |  |  |  |  |  |
| Nº<br>Esclavo<br>(00-3F <sub>4</sub> ) | 08 <sub>H</sub> | Código<br>Subfunción<br>SF0 SF1 | Valor<br>Contador<br>H L    | CRC<br>H L |  |  |  |  |  |  |  |

Figura Nº. 1.10 Petición y respuesta de la función: Control de contadores (08H)

| S                | ubfunc          | ión              | Da              | tos             | Terres   |
|------------------|-----------------|------------------|-----------------|-----------------|--|
| $\mathbb{N}^{o}$ | Ce              | ódigo            | D0              | D1              | Tarea  |
| 0                | 00 11           | 008              | $XY_{H}$        | $ZT_{H}$        | El esclavo envía el eco XYZT de petición como test.                        |
| 3                | 00 <sub>H</sub> | 03 <sub>11</sub> | ZZ <sub>H</sub> | 00 <sub>H</sub> | Modifica el carácter de fin de trama en modo ASCII por<br>ZZ <sub>et</sub> |
| 10               | 00,             | 0A <sub>st</sub> | 00,             | 00,             | Puesta a cero de los contadores  |
| 11               | 00 <sub>H</sub> | $0B_{H}$         | 00 1            | 00 <sub>H</sub> | Lectura del contador l   |
| 12               | 00 <sub>H</sub> | 0C <sub>H</sub>  | 00 1            | 00 <sub>H</sub> | Lectura del contador l   |
| -13              | 00,4            | 0D <sub>H</sub>  | 00,4            | 00,             | Lectura del contador l   |
| 14               | 00 <sub>H</sub> | 0E <sub>H</sub>  | 00 1            | 00 <sub>H</sub> | Lectura del contador l   |
| 15               | 00 1            | 0F <sub>H</sub>  | 00 1            | 00 <sub>H</sub> | Lectura del contador l   |
| 18               | 00,4            | 12 <sub>10</sub> | 00,4            | 00,4            | Lectura del contador l   |

Tabla 3 Petición del contenido y control de los 8 primeros contadores de diagnóstico de un esclavo

# 1.3.7.7 FUNCIÓN 11:

La petición del contenido del contador de diagnóstico número 9, no se realiza por la función 8, sino por la función 11. Las tramas de petición y respuestas son las indicadas por la Figura N°. 1.11



Figura Nº. 1.11 Petición y respuesta de la función: Contenido contador 9 (0BH)

## 1.3.7.8 FUNCIÓN 15:

Escritura de bits del autómata. La trama es la indicada en la Figura Nº. 1.12 La forma de direccionamiento es análoga a la indicada para las funciones 1 y 2.



Figura Nº. 1.12 Petición y respuesta: Escritura de bits (0FH )

## 1.3.7.9 FUNCIÓN 16:

Escritura de palabras del autómata. La trama es la indicada en la Figura Nº. 1.13.

| Petición                               | maestro          |                                  |                            |                                   |                                     |            |
|--|------------------|----------------------------------|----------------------------|-----------------------------------|-------------------------------------|------------|
| Nº<br>Esclavo<br>(00-3F <sub>a</sub> ) | 10F <sub>H</sub> | Dirección<br>1º Palabra<br>PP PP | Nº de<br>Palabras<br>NN NN | N <sup>e</sup> de<br>Octetos<br>M | Valor de<br>las pelebras<br>HL, HL, | CRC<br>H L |
| Respuest                               | a del esc        | lavo                             |                            |                                   |                                     |            |
| Nº<br>Esclavo<br>(00-3F <sub>a</sub> ) | 10 <sub>H</sub>  | Dirección<br>1º Palabra<br>PP PP | Nº de<br>Palabras<br>NN NN | CRC L                             |                                     |            |

Figura Nº. 1.13 Petición y respuesta: Escritura de palabras (10H)

## **1.3.8 MENSAJES DE ERROR:**

Puede ocurrir que un mensaje se interrumpa antes de terminar. Cada esclavo interpreta que el menaje ha terminado si transcurre un tiempo de silencio equivalente a 3,5 caracteres. Después de este tiempo el esclavo considera que el carácter siguiente es el campo de dirección de esclavo de un nuevo mensaje.

Cuando un esclavo recibe una trama incompleta o errónea desde el punto de vista lógico, envía un mensaje de error como respuesta, excepto en el caso de mensajes de difusión. La trama del mensaje de error es la indicada en al Figura N°. 1.14.



03

06

Datos incorrectos

Autómata ocupado

Si la estación maestra no recibe respuesta de un esclavo durante un tiempo superior a un límite establecido, declara el esclavo fuera de servicio, a pesar de que al cabo de un cierto número de ciclos hace nuevos intentos de conexión.

## 1.4. PROGRAMACIÓN DEL PLC

### **1.4.1 INTRODUCCIÓN:**

Un PLC o "autómata" es un dispositivo electrónico programable por el usuario que se utiliza para controlar, dentro de un entorno industrial, máquinas o procesos lógicos y/o secuénciales.

Normalmente se requiere un PLC para:

- Reemplazar la lógica de relés para el comando de motores, máquinas, cilindros, neumáticos e hidráulicos, etc.
- Reemplazar temporizadores y contadores electromecánicos.
- Actuar como interfase entre una PC y el proceso de fabricación.
- Efectuar diagnósticos de fallas y alarmas.
- Controlar y comandar tareas repetitivas y peligrosas.
- Regulación de aparatos remotos desde un punto de la fábrica.

El lenguaje empleado es sencillo y al alcance de todas las personas. Está basado en los esquemas eléctricos funcionales de control.

Este lenguaje es conocido como Diagrama Escalera (Ladder).

Otro lenguaje que se puede utilizar para la programación de PLCs, es el Diagrama de Flujo Secuencial o SFC (anteriormente denominado Grafcet), reconocido como el lenguaje gráfico mejor adaptado a la expresión de la parte secuencial de la automatización de la producción.

El SFC representa la sucesión de las etapas en el ciclo de producción. La evolución del ciclo, Etapa por Etapa se controla por una "Transición" ubicada entre cada etapa.

A cada una de las etapas le puede corresponder una o varias acciones. A cada transición le corresponde una "receptividad", condición que debe cumplirse para poder superar la transición, lo que permite la evolución de una etapa a la siguiente.

Para asegurar la estandarización de los lenguajes de programación de los PLCs, y asegurarle al usuario una única forma de programar, sin importar la marca comercial del PI-C, ha sido establecida la norma IEC 1131-3 que fija criterios en tal sentido.

Así, la norma define los lenguajes de programación: Escalera (ladder). Lista de instrucciones (Assembler), Estructurado (Similar el Pascal), Bloques de Función y

Diagrama Flujo de Secuencial (SFC o Grafcet). Según el tipo de PLC que se escoja, podrá tener uno o más de estos lenguajes.

Cuando la aplicación crece en complejidad dado el tipo de señales a manejar, es posible incrementar la capacidad de Entradas/Salidas. Además permite el control de señales, tanto digitales como analógicas.

Un concepto que cada día es más necesario aplicar, es la comunicación entre PLCs o con un sistema de supervisión (SCADA).

Cuando es el momento de realizarlo, el PLC dispone de la capacidad de resolverlo agregando los módulos de comunicación necesarios.

### 1.4.2. PLC TD100

Este PLC se caracteriza por tener: I/Os análogas, puertos LCD, PWM, controladores de motor de paso, interrupciones, entradas para medición de frecuencia de pulso, Reloj/Calendario de tiempo real, puerto RS232, puerto de redes RS485, PID. Adicionalmente, los comandos en BASIC simplifican enormemente el uso de estas opciones.

### 1.4.3 DESARROLLO Y PROGRAMACIÓN

El lenguaje de programación es fácil manejo para realizar tareas complejas usando el lenguaje Ladder+BASIC. Con los altos costos de ingeniería, disminuyendo los tiempos de desarrollo representando grandes.

La mayoría de los PLCs de la industria se programan en un lenguaje altamente especializado llamado "Lenguaje de Contactos", "Lógica de Escalera" o "Ladder Logic". Esta metodología fue de las primeras y su fortaleza radica en el manejo de entradas y salidas digitales, pero cuando se necesitan procesar datos en forma numérica, hacer matemática, almacenamiento de datos o comunicaciones se convierte en algo bastante complejo.

Para programar un modulo análogo en un PLC se inicializa, se adquiere el valor almacenado, se hace las conversiones y escalamiento correspondientes.

Por otro lado, los computadores de escritorio son programados típicamente en leguajes de alto nivel tales como BASIC, C, PASCAL, C++, Java, etc. Estos poderosos lenguajes están orientados al procesamiento de datos pero no para el manejo de entradas y salidas digitales, como en el caso de PLCs.

Combinando estas dos importantes tecnologías de programación en uno solo, obteniendo el lenguaje industrial "Ladder+ Basic" de PLCs, que puede manejar sofisticadas tareas con solo unas pocas líneas.

### 1.4.3.1 CREANDO UN PROGRAMA EN LENGUAJE ESCALERA

Para entender de mejor manera la programación tipo escalera realizaremos un pequeño programa como se indica a continuación en la Figura Nº. 1.15:



Figura Nº. 1.15 Programación tipo escalera (ejemplo)

#### Siga los siguientes pasos:

- 1. Abra el menú del Archivo y seleccione "Nuevo".
- 2. Definir el I/Os para ser usado para el programa.

Los I/Os siguientes se requieren:

- Las entradas: La salida, la Parada, el Manual, el Paso,
- Los rendimientos: Out1, Out2,.... Out8
- Las paradas: La RUN
- Los cronómetros: La duración
- Sequencer: Seq1



Abra la I/O etiqueta corrección Ventana apretando <F2>. (Aunque usted también puede hacer clic adelante "Revise" el menú y seleccione el artículo la "Mesa de I/O" a logre el mismo, nosotros recomendamos aprender la tecla directa F2 como fuertemente es a menudo mucho más conveniente usar).

- Desplace a la ventana de las Entradas usando el cursor del left/right codifica o
  por haciendo clic en el left/right colorido rojo flechan los botones o
  simplemente lo seleccionan de la caja selecta entre los botones de flecha de
  left/right.
- Mueva la barra de momento culminante de color azul profunda para Entrar #1 posición haciendo clic en él. Haga clic para abrir un campo del texto por entrar en el nombre para la Entrada de nuevo #1.
- Entre en la Salida del nombre para Entrada #1. la Prensa <Entre> importante para aceptar el nombre. El campo del texto será cerrado y la Salida del nombre se asigna para Entrar #1 ahora. Si usted cometiera un error, simplemente apriete el "spacebar" o hace clic de nuevo en la situación de la entrada para revisarlo.
- 3. Definir contadores o cronómetros.

Después de que usted ha entrado en la "Duración" de nombre de etiqueta por el Cronómetro #1, una caja de entrada de texto se abre al Valor" Fijo la situación de

Cronómetro #1 para usted para entrar en el SV para el cronómetro. El rango de SV es entre 0 y 9999. Entre en el valor 1000 a esta situación.

| <u>8</u> 1/     | U Labels   | ×         |
|-----------------|------------|-----------|
| •               | Tiners 7   |           |
| 0               | Label Name | Set Value |
| I               | Duration   | 1000      |
| 3               |            |           |
| 4               |            |           |
| 6               |            |           |
| - É             |            |           |
| 9<br>10         |            |           |
| 11              |            |           |
| $\frac{12}{13}$ |            |           |
| 14              |            |           |
| 16              |            |           |
| 17              |            | -         |

Para un cronómetro normal con 0.1s cronometre base, el valor 1000, representa 100.0 segundos que los medios que la "Duración" el cronómetro quiere tiempo-fuera después de 100.0 segundos.

4. crear el programa (circuito)

| atari | Step |  |  |   | NUN<br>(RLV)      |
|-------|------|--|--|---|-------------------|
| RUN   | *    |  |  | l | Dusation<br>(TIM) |

Con el indicador del circuito (el triángulo colorido rojo) a Circuito #1, apriete el <Spacebar> entrar en la "Escalera de mano Revisan" el modo. Usted también puede entrar en el circuito revise el modo el haciendo clic doble a Circuito #1.

| E  | TRiL06    | I Version          | 5.0 - Uni00   | ed.pc5    |      |           |                       |            |     |              |
|----|-----------|--------------------|---------------|-----------|------|-----------|-----------------------|------------|-----|--------------|
| Fi | le Edit   | Controlle          | r Simulat     | e Circuit | Help |           |                       |            |     |              |
|    | Circuit ≭ | । - <mark>-</mark> | - -, <b>-</b> | -, 4+     | , –( | $2_T = ($ | $)_{a} \rightarrow f$ | n), Lifn), | 「花」 | DEL          |
| Π  |           |                    |               |           |      |           |                       |            |     | -            |
| h  |           |                    |               |           |      |           |                       |            |     | <u>н</u>     |
| Ш  |           |                    |               |           |      |           |                       |            |     |              |
| Ш  |           |                    |               |           |      |           |                       |            |     |              |
| Ш  |           |                    |               |           |      |           |                       |            |     |              |
| Ш  |           |                    |               |           |      |           |                       |            |     |              |
| Ш  |           |                    |               |           |      |           |                       |            |     |              |
|    |           |                    |               |           |      |           |                       |            |     | <u>لئے ،</u> |
| Ľ۵ |           |                    |               |           |      |           |                       |            |     | · · //       |

Una vez usted entra en la "Escalera de mano Revise" el modo, una fila de iconos de la escalera de mano aparece a lo largo de la cima del TRiLOGI principal la ventana sólo debajo del tirón abajo el menú. Lo siguiente es una descripción de cada artículo, como se muestra en la figura Nº 1.16.

|                     | <1> - el clic izquierdo para insertar un contacto de la serie normalmente-abierto.                     |
|---------------------|--|
|                     | <2> - el clic correcto para insertar un contacto de la serie normalmente-cerrado.                      |
| ЧHэ                 | <3> - el clic izquierdo para insertar un N.O. el contacto paralelo al elemento resaltado               |
|                     | <4> - el clic correcto para insertar un N.C. el contacto paralelo al elemento resaltado                |
| Τ°                  | <5> - el clic izquierdo para insertar un N.O. el contacto paralelo para adjuntar uno o más elementos.  |
|                     | <6> - el clic correcto para insertar un N.C. el contacto paralelo para adjuntar uno o más elementos.   |
| -( ),               | <7> - la inserción un rollo normal que puede ser un rendimiento, parada, cronómetro o contador.        |
| └─(_) <sub>8</sub>  | <8> - la inserción un rollo del rendimiento paralelo (no una rama entera) al rollo actual.             |
| [Fn]g               | <9> - la inserción un rollo de la función especial que incluye ejecución de CusFn                      |
| └─[Fn] <sub>d</sub> | <0> - la inserción un rollo de la función especial paralelo al rollo actual.                           |
| μ <del>ι</del>      | - El invertido el elemento de N.O. a N.C. o de N.C. a N.O.   |
| Ŧ                   | Haga clic para mover la barra del momento culminante al derecho (el mismo efecto como apretar la llave |
| _                   | de la flecha correcta).  |
|                     | Esto puede usarse para mover el cursor a una unión que no puede seleccionarse por el clic del ratón.   |
| DEL                 | Doble-haga clic para anular un elemento resaltado.   |

Figura Nº. 1.16 Menú de la programación Lader.

Ahora inserte el primer elemento izquierdo-haciendo clic en el icono. El icono cambiará a un color amarillo luminoso a 2.

Muéstrele el tipo del elemento que usted está creando. Al mismo tiempo, una mesa de I/O debe aparecer en la pantalla con un el fondo de crema-color ligero en lugar de la luz normal el fondo azul. La mesa de I/O ahora los actos como un menú automático para usted escoger cualquiera de la etiqueta predefinida nombra para este contacto.

Normalmente no se significan los volúmenes en la mesa ser revisado en este momento. Desplace a la mesa de la Entrada y clic en la Salida de nombre de etiqueta y un contacto normalmente-abierto se creará a Circuito #1.

| 📸 TRiLOGI Version 5.0 - [C:\TRiLOGI\TL5\usr\samples\Demo.PC5] - [Source: Local Disk] | - O X |
|--|-------|
| File Edit Controller Simulate Circuit Help   |       |
| Circuit#1 ⊣⊢₁⊣⊣₂⊣⊢₂ −-()7 └-()8 −(En)8 └-(En)8 → CEL                                 |       |
| statt<br>▶ <mark>──┴│├───</mark>   | -     |

Luego, cree la RUN del contacto que es paralelo al contacto de la Salida izquierdohaciendo clic en el icono. La mesa de I/O aparecerá de nuevo. Desplace a la mesa de la Parada y seleccione la parada de la RUN.

| 😤 TRiLOGI Version 5.0 - [C:\TRiLOGI\TL5\usr\samples\Demo.PC5] - [Source | : Lo 💶 🗵    |
|---|-------------|
| File Edit Controller Simulate Circuit Hela                              |             |
|   |             |
| stat<br>RUN<br>III  | II<br>I     |
|   | <u>)</u> // |

De igual manera seguimos con el resto del programa.

 Luego, mueva la barra del momento culminante a la unión del fin correcta de los circuitos paralelos como siga:

| <u>ڻ</u> |              |                                |            | TRiLO     | GI Version 5.31 - Untitled | (             | 0 0 0      |
|----------|--------------|--------------------------------|------------|-----------|----------------------------|---------------|------------|
| File     | Edit         | Controller                     | Simulate   | Circuit   | Help                       |               |            |
| C        | ircuit #     | 2                              |            |           |                            |               |            |
|          | salida<br>⊣¦ | parada<br>i  <br>t1<br>t1<br>t | manual<br> | paso<br>H |                            | out2<br>(OUT) | •          |
| 4        |              |                                |            |           |                            | <b>)</b>      | ) <b>•</b> |

6. Ahora, haga clic en el icono para insertar un rollo de la función especial. Un menú aparecerá para usted seleccionar la función especial

deseada. Haga clic en el [AVseq]-adelanto del artículo Sequencer" para insertar el :

| Select a Fu | nction                     | × |
|-------------|----------------------------|---|
| 1. [DNCtr]  | - Decrement Rev. Counter   |   |
| 2.[RSCur]   | - Reset Counter            |   |
| 3.[UpCtr]   | - Increment Rev. Counter   |   |
| 4.[AVseq]   | - Advance Sequencer        |   |
| 5.[RSseq]   | - Reset Sequencer          |   |
| 6.[StepN]   | - Set Sequencer to Step #N |   |
| 7.[Latch]   | - Latching Relay/Output    |   |
| 8.[Clear]   | - Clear Latched Relay      |   |
| 9.[ILock]   | - Interlock Begin          |   |
| A. [ILoff]  | - Interlock End            |   |
| B. [dDIFU]  | - Differentiate Up         |   |
| C.[dDIFD]   | - Differentiate Down       |   |
| D. [CusFn]  | - Custom Function          |   |
| E.[dCusF]   | - Diff. Up Custon Funcs    |   |
| F. [MaRST]  | - Master Reset             |   |
|             |                            |   |

### 1.4.3.2 CREANDO UN PROGRAMA EN LENGUAJE BASIC

### a. Las Variables del entero

El compilador de TBASIC maneja palabras de 32-bits. Maneja 26 variables (UN a Z) ése es 32 bits en la longitud. Estas variables pueden representar los números entre - 23' a +23'. Las variables del sistema restantes y memoria de los datos DM[n] es todas las variables del 16-bit que los medios que ellos pueden guardar sólo números entre - 32768 a +32767. Estas variables del 16-bits son:

DM[1].. DM[4000] (el Total = 4000), input[n], output[n], relay[n]. timerbit[n]. ctrbit[n]. timerPV[n],ctrPV[n], HSCPV[n]. (la Velocidad Alta Lo opuesto). TIME[n]. DATE[n].

#### b. Los Datos del string

Un string es una sucesión de caracteres alfanuméricos (el 8-bit ASCII codifica) qué colectivamente forman una entidad. Una constante del string puede contener de O a 70 caracteres adjuntados en doble comilla. Por ejemplo

"TBASIC hizo el procesando numérico a PLC un bit de pastel! " "\$102.345.00"

TBASIC apoya un máximo de 26 variables del string UN \$, B \$0'0 Z \$. Cada variable del string puede contener de 0 (el string nulo) o a un máximo de 70 carácteres.

#### c. Las Declaraciones de TBASIC y Funciones

La mayoría de las declaraciones y funciones que el idioma de TBASIC se clasifica más abajo.

ABS, ADC, ASC, CALL, CHR\$, CLRBITCLRIO, DELAY, FOR..NEXT, GetCtrSV, GetTimerSV, GetHigh16, GOTO, HEX\$, HEXVAL, HSCDEF, HSCOFF, HSTIMER, If..Then..Else, INCOMM, INPUT\$, INTRDEF, INTROFF, LEN, LET, LOAD\_EEP, LSHIFT, MID\$, NETCMD\$, OUTCOMM, PAUSE, PIDcompute, PIDdef, PMON, PMOFF, PRINT #.

PulseFrequency, PulsePeriod, PulseWidth, ReadModbus, REFRESH, REM ('), RESET,RETURN, RSHIFT, SAVE,\_EEP, SetBAUD, SetBIT, SetCtrSV, SetTimerSV, SetDAC, SetHIGH16, SetLCD, SetLED, SetIO, SetPassword, SetProtoco,l SetPWM, SetSystem, STATUS.

StepCount, StepCountAbs, StepHome, StepMove, StepMoveAbs, StepSpeed, StepStop, STR\$, STRCMP, STRLWR\$, STRUPR\$, TESTBIT, TestIO, ToggleIO, VAL, WHILE, WriteModbus,

### 1.4.3.3 CARRERA DE LA SIMULACIÓN DE CUSFN.

Después de que ha completado codificando un CusFn, ejecute el simulador apretando <F9> o <Ctrl-F9> la llave. Ejecute el CusFn encendiendo su entrada del mando. Si su CusFn ejecuta un orden que afecta el estado de la lógica de cualquier I/O, el efecto puede verse inmediatamente en la pantalla del simulador.

Sin embargo, si el cómputo afecta sólo las variables, que usted puede necesitar examinar las variables interiores.

