



DI-12046
/ DON
MFW 1244

TE



CINVESTAV-IPN
Biblioteca de Ingeniería Eléctrica



F8000008712

✓
CM

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

1010

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITECNICO NACIONAL
(CINVESTAV)

DEPARTAMENTO DE INGENIERIA ELECTRICA
SECCION DE COMPUTACION

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

"DESARROLLO DE UNA INTERFAZ DE USUARIO PARA UN
SISTEMA DE CONTROL DISTRIBUIDO"

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA

Tesis que presenta el Ing. Ricardo Othón Valdés Ramírez para obtener el grado de MAESTRO EN CIENCIAS en la especialidad de INGENIERIA ELECTRICA con opción en COMPUTACION.

Trabajo dirigido por el M. en C. José Oscar Olmedo Aguirre.

CLASIF. 87491.1
ADQUIS. 61-12046
FECHA 11-11-91
PROCED. 303

A mi madre, Guadalupe Ramirez Vda. de Valdés, que siempre me ha apoyado y ha dedicado su vida a mis hermanos y a mi.

A mi padre, Othón Valdés Medina, a quien recuerdo con respeto y cariño.

A mi hijo Ricardito, y a mi esposa Cristina Velasco Ramos.

A mis hermanos, Teresa, Roberto y Araceli.

A mi padrino, Bernardino Ramirez R.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

AGRADECIMIENTOS

A los M. en C. Oscar Olmedo Aguirre, Feliú Sagols Troncoso y Jorge Luis Alcántara Gómez Pineda por el interés y el tiempo dedicado a este trabajo.

A mis compañeros y amigos por la ayuda y comentarios.

A la Lic. Beatriz Breña por la ayuda en la corrección de estilo de este trabajo.

Al CINVESTAV, IIE y al Banco de México por el apoyo brindado para la obtención de grado.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. T. N.
BIBLIOTECA
INGENIERIA ELECTRICA

I N D I C E

ANTECEDENTES	1
OBJETIVO Y ALCANCE	3
1. INTRODUCCION	6
1.1 Breve historia de la computadora en el control de procesos	6
1.2 Variables de proceso	7
1.3 Tableros de control	9
1.4 Pantalla de rayos catódicos	10
1.4.1 Despliegues de información	12
2. DESCRIPCIÓN FUNCIONAL DEL SISTEMA	16
2.1 Base de datos	17
2.2 Configuración del sistema para monitorear variables de proceso	23
2.2.1 Configuración de sistemas	24
2.2.2 Configuración de subsistemas	24
2.2.3 Configuración de variables	25
2.2.4 Configuración de diagramas de proceso	25
2.2.5 Configuración de diagramas de barras	26
2.2.6 Configuración de gráficas de tendencia	27
2.3 Operación del sistema	27
2.3.1 Diagramas de proceso	29
2.3.2 Diagramas de barras	30
2.3.3 Gráficas de tendencia	31
2.4 Comunicación con los controladores programables	32
2.4.1 Protocolo de comunicaciones de los controladores SAC	35
2.4.2 Protocolo de comunicaciones de los controladores con microcontrolador 8051	38
3. DISEÑO DEL SISTEMA	41
3.1 Menus de opciones del sistema	42
3.1.1 Menu principal	43
3.1.2 Estructura de datos de los menus	44
3.2 Diagramas de proceso	45
3.2.1 La utilización de "PCX Programmer's Toolkit"	45
3.2.2 Estructura de datos para los diagramas de proceso	46
3.2.3 Biblioteca de funciones	51

3.3	Diagramas de barras	56
3.3.1	Matriz de grupos de barras	56
3.3.2	Biblioteca de funciones para diagramas de barras	59
3.4	Gráficas de tendencias	61
3.4.1	Colas circulares para gráficas de tendencias	62
3.4.2	Biblioteca de funciones	64
3.5	Variables de proceso	66
3.5.1	Información general de variables de proceso	66
3.5.2	Valor de variables de proceso	67
3.5.3	Biblioteca de funciones	68
3.6	Sistemas de proceso	69
3.6.1	Estructura de datos para diagramas de proceso	70
3.6.2	Biblioteca de funciones	71
3.7	Subsistemas de proceso	72
3.7.1	Estructura de datos de subsistemas	72
3.7.2	Biblioteca de funciones	73
4.	APLICACIÓN DEL SISTEMA	76
4.1	Simulación de variables de proceso	76
4.2	Descripción del proceso de ejemplo	77
4.3	Protocolo de comunicaciones	79
4.4	Configuración del sistema	79
5.	CONCLUSIONES Y RECOMENDACIONES	81
5.1	Conclusiones	81
5.2	Recomendaciones	82
	REFERENCIAS	83

A N T E C E D E N T E S

El impresionante desarrollo de la computación en los últimos años ha influido definitivamente en la automatización de procesos industriales.

En esta automatización se ha notado que un sistema de cómputo mejora considerablemente, entre otras cosas, la comunicación operador-máquina-proceso, ya que la proporciona más cómoda y agradable debido a su rapidez y confiabilidad.

En la actualidad, la supervisión y el control del equipo instalado en plantas industriales requiere el uso de computadoras para que la eficiencia de éstas sea óptima. Esto se debe a que la operación de las plantas es cada vez más compleja, debido a la cantidad de información que se debe analizar para su operación correcta.

Un uso práctico de las computadoras ampliamente difundido en plantas industriales son los **sistemas para monitorear las variables de proceso**. El personal de operación de estas plantas normalmente necesita monitorear continuamente variables como presión, temperatura y nivel, entre otras, a fin de detectar cualquier desviación anormal de su valor nominal (punto de ajuste). Asimismo, también necesita detectar e interpretar alarmas para la ejecución adecuada de los procedimientos de operación.

Actualmente en algunas plantas industriales este monitoreo se

realiza por computadora y la tendencia es que se incorpore en otras con el proposito de aumentar la eficiencia o debido al incremento en la complejidad de operacion de la planta.

Existe en el mercado una gran cantidad de equipos que le brindan al operador de plantas industriales estos sistemas de procedencia extranjera, lo que trae consigo un elevado costo y problemas de mantenimiento y respaldo técnico.

De lo anteriormente expuesto, nace la inquietud de desarrollar un sistema con el cual se pueda monitorear el estado de las variables de procesos industriales de una manera fácil, segura y en tiempo real. El operador puede considerar este sistema de monitoreo como una "ventana" hacia el proceso.

O B J E T I V O Y A L C A N C E

OBJETIVO

El objetivo de este trabajo es integrar un sistema que permita observar las variables de un proceso industrial en una computadora personal (PC), desarrollando los programas y enlazando la PC a controladores programables por medio de su puerto serie.

ALCANCE

Los programas presentarán la información de las variables de proceso en la forma de:

- Diagramas de proceso
- Gráficas de tendencia
- Diagramas de barras

que son las mas usadas para mostrar a los operadores de los sistemas de control el estado del proceso en las plantas industriales.

Para enlazar la PC (por su puerto serie RS-232) a los controladores programables se empleara un convertidor RS-232/RS-485 (vease la figura 1.1).

Este sistema tendra la capacidad de ser modular de tal forma que

pueda expandirse y configurarse de acuerdo con las necesidades requeridas por los usuarios. La configuración que proporciona el sistema no es la definitiva; se puede modificar de acuerdo con las necesidades o cambios que tenga la planta industrial.

El usuario podrá modificar la configuración del sistema de monitoreo, pero también será el responsable de revisar que ésta sea la que él espera.

El sistema para monitorear variables de proceso operará en el sistema operativo MS-DOS, estará escrito en el lenguaje de programación C y se utilizará el compilador Turbo C, versión 2.0.

Además se emplearán funciones de las bibliotecas: "PCX Programmer's Toolkit" y "Turbo C".

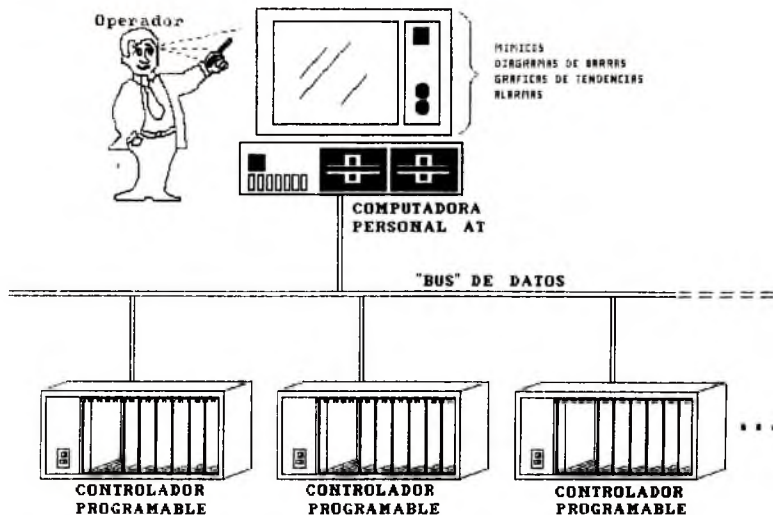


FIGURA 1.1 SISTEMA PARA MONITOREAR VARIABLES DE PROCESO.

1. INTRODUCCION

1.1 Breve historia de la computadora en el control de procesos

El uso de computadoras en el control y automatización de procesos se inició hace aproximadamente 30 años, a mediados de la década de los cincuentas.

Un hecho importante es que el desarrollo de sistemas de control por computadora fue una iniciativa más bien de los fabricantes de computadoras y equipo electrónico, y no de la industria de procesos. Esto acarreo un conjunto de problemas originados por el desconocimiento de la problemática industrial, y su posterior difusión. No fue sino hasta hace relativamente poco tiempo, al aparecer los **sistemas de control distribuido (DCS)** y los sistemas de adquisición de datos, que el uso de computadoras en el control de procesos industriales empezó a tener éxito de manera definitiva.

Antes de que aparecieran los DCS se realizaron considerables cambios en el control de procesos (Williams, 1984). Estos cambios se fueron presentando de acuerdo con el avance tecnológico, especialmente la computación.

La primera computadora digital con sistema de control la desarrolló la compañía Hughes Aircraft para la Fuerza Aérea de los Estados Unidos. Esta computadora se llamó DIGITAC y en 1954 controló con éxito un avión en vuelo.

Desde ese tiempo empezaron a surgir computadoras para el control

de procesos, pero la tecnología de la computación no ayudó lo suficiente para que la computadora entrara de lleno en el control de procesos. La creación del microprocesador hizo posible el surgimiento del DCS y con esto se notó con más frecuencia y éxito la presencia de la computadora en los cuartos de control.

Comúnmente los DCS's están formados por tres elementos principales que son:

- Controladores programables
- Red de comunicaciones
- Interfaz hombre-máquina

La interfaz hombre-máquina en un principio se implementó con computadoras ayudándose por generadores gráficos y pantallas de rayos catódicos. Posteriormente, con el avance de la tecnología, estas computadoras se sustituyeron por minicomputadoras.

Para que se instale un DCS en una planta industrial, éste tiene que ser redituable. El costo de las minicomputadoras y demás equipo sofisticado de la interfaz hombre-máquina hace que solo se pueda instalar en plantas industriales de considerable tamaño.

Este trabajo será útil en plantas industriales en donde se requiera conocer en tiempo real el estado de las variables de proceso y en donde no se justifique una minicomputadora para desplegar la información.

1.2 Variables de proceso

El control de procesos se efectúa mediante la constante

supervisión de ciertos parámetros, que indican la correcta transformación físico-química de materias primas en el tiempo adecuado. El resultado de esta transformación da un producto final.

Los parámetros referidos anteriormente son los denominados **variables de proceso** y pueden ser de origen físico y/o químico.

Los tipos de procesos que se encuentran en el mundo industrial son innumerables por lo que su clasificación resulta difícil, pero las variables de los procesos sí se pueden clasificar fácilmente por el papel e importancia que tienen en el proceso. Shinsky (1981) propone las siguientes cinco categorías de variables:

- Velocidad de producción
- Inventario
- Medio ambiente
- Calidad de producción
- Economía

Shinsky explica que las variables de velocidad de producción determinan la velocidad a la que opera el proceso. Generalmente solo hay una variable de este tipo en un proceso y por lo regular es la velocidad de alimentación de la materia prima, pero en algunos casos, es la velocidad de flujo de un producto final o intermedio.

Las variables de inventario representan la acumulación de material o energía en un punto específico del proceso. Por ejemplo: gas, líquido, sólido, composición y energía.

Las variables del medio ambiente determinan la condición en que funciona el proceso. Por ejemplo: temperatura, presión y análisis químico.

La calidad de producción debe controlarse en cualquier proceso. Quizá sea la variable más importante, porque el valor del producto depende de ella. Por ejemplo: propiedades físicas o análisis químicos.

El último grupo es de naturaleza económica y, de hecho, rara vez se controla. Este grupo representa el costo de producción, el cual no se obtiene de una sola medición. Por ejemplo: flujo, radio de flujo, posición de válvulas y análisis químico.

1.3 Tableros de control

Los operadores de las plantas industriales mantienen el proceso en forma aceptable, basándose en la información que reciben del estado en que se encuentran las variables de proceso.

En el caso de los sistemas convencionales, el operador recibe la información correspondiente al estado de las variables por medio de instrumentos montados en paneles (tableros) de control. Estos instrumentos por lo regular son indicadores, registradores, paneles de alarmas y, en algunos casos, mimicos.

La concentración de toda la información en un solo tablero facilita la operación del proceso. Sin embargo, para procesos industriales, el tamaño de los tableros del operador es

normalmente muy grande. Entre mas variables a monitorear existan, los tableros serán mayores, y será más difícil que el operador los visualice.

El operador necesita que se le proporcione la mejor herramienta posible para optimizar sus actividades, puesto que su rendimiento y sus responsabilidades son esenciales para una producción óptima.

La información, con que debe trabajar el operador, ha tenido cambios en su representación. Estos cambios han hecho que la información se interprete correctamente en beneficio de la operación del proceso. Al principio los tableros de control eran excesivamente grandes debido a que los instrumentos que contenían eran de considerable tamaño. Posteriormente con el avance de la tecnología se redujo el tamaño de estos instrumentos y con esto el de los tableros de control.

La creación del DCS hizo posible que estos tableros se sustituyeran por pantallas de rayos catódicos y la información que se presentaba con los instrumentos se mostrara con desplegados gráficos.

1.4 Pantalla de rayos catódicos

Como se menciona, el avance de la computación ha intervenido en los procesos industriales y una muestra de esto es la interfaz hombre-máquina, la cual viene precisamente a sustituir a los tableros de control convencionales.

Esta interfaz es comúnmente de rayos catódicos, aunque en el futuro posiblemente sea con otras tecnologías. Lo que se muestra en ésta son desplegados gráficos proporcionando el estado del proceso.

El principal beneficio al usar pantalla es que la información que el operador necesita se comprime en un área que los ojos pueden abarcar sin mover la cabeza, reduciendo de esta manera el tiempo necesario para detectar una falla y responder a un cambio de condiciones en situaciones de emergencia (Villavicencio, 1984).

Existen despliegues totales y parciales. En los primeros se muestra el total de la información o la mayor parte. Los segundos sólo muestran una parte del proceso.

Por supuesto que el operador en un despliegue parcial únicamente observa una parte de la instrumentación convencional del proceso, pero un despliegue total le permite observarlo con mayor amplitud.

La pantalla de rayos catódicos hace que el operador tenga mejor control de las condiciones de proceso, ya que, puede observar bastantes variables a la vez y principalmente las de operación clave, lo cual, generalmente lo hará por observaciones de las variables que no estén en condición normal. Un requisito importante de su trabajo es detectar cualquier condición anormal tan pronto como sea posible para poder realizar una acción correctiva sobre las condiciones que provocan un disturbio. Con la ayuda de una CRT es fácil detectar esta condición anormal.

1.4.1 Despliegues de información

Debido a que la información que se le debe presentar al operador por lo regular es mucha, esta debe proporcionarsele en agrupamientos significativos de despliegue.

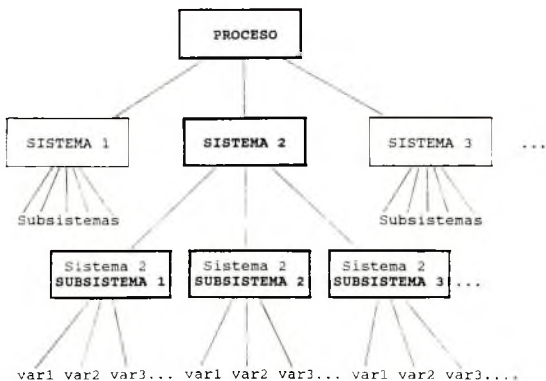


Figura 1.2 Estructura jerárquica de un proceso industrial

Para que se puedan hacer agrupamientos de despliegue es

importante conocer la estructura del proceso. Por lo regular un proceso por su naturaleza propia está dividido en sistemas y subsistemas.

Un sistema está compuesto por subsistemas y estos contendrán diversas variables. Es muy importante tomar en cuenta esta estructuración a fin de mostrar al operador la información en orden, con el objeto de que tenga acceso fácil y rápido a ella.

En la figura 1.2 se puede apreciar la estructura de un proceso industrial. Un proceso está formado por sistemas y éstos a su vez por subsistemas. Por último los subsistemas tendrán grupos de variables.

De esta forma, la información al operador se debe mostrar de lo general a lo particular. El operador debe poder visualizar el estado general del proceso y cuando él lo requiera debe poder ver la información más a detalle.

Generalmente se debe contar con un despliegue total, pero de no ser posible, se deben tener despliegues por sistemas, subsistemas y de detalle. Esto es posible aplicarlo en diagramas de proceso y diagramas de barras. En gráficas de tendencia se utiliza sólo en el caso de que las tendencias a graficar sean demasiadas.

Por otra parte, los tipos de despliegues más usados son :

- Despliegues gráficos de proceso (diagramas de proceso)
- Despliegues de diagramas de barras (diagramas de barras)
- Despliegues de tendencia (gráficas de tendencia)
- Despliegues de alarmas

Los despliegues graficos son los que reemplazan a los diagramas mimicos en los tableros de control convencionales. Una ventaja es que en estos despliegues se puede estar observando en tiempo real el valor de las variables incluidas en él, y ademas se varia el color de la variable en caso de estar en estado de alarma, para una mayor claridad para el operador. Adicionalmente se puede tener animación de una variable en el diagrama de proceso. Esta animacion consistira en la indicacion grafica del valor de la variable por medio de una barra de color que se dibuje dentro de un recipiente.

El despliegue de tendencias puede ser de dos formas, en tiempo real e histórico. Los despliegues en tiempo real muestran los cambios de la variable por periodos de tiempo cortos, 60 segundos a 24 horas. Los despliegues históricos pueden ser por periodos tan largos que van de las 8 horas hasta semanas.

El despliegue aparece como un trazo a traves de la pantalla o como una serie de horas variando la altura. En ambos casos se tiene un efecto similar a una gráfica de registrador que muestra un perfil de valores contra el tiempo. O sea que, las gráficas de tendencia sustituyen a los registradores que se encontraban montados en los tableros de control.

Las alarmas son desplegadas alfanumericas. Una condición de alarma se puede proporcionar con una anunciación audible y ademas centelleando la alarma en conflicto. Esto puede presentarse sobre un despliegue total. Ademas, se puede almacenar un registro de

alarmas para después imprimirlo.

Las alarmas generalmente tienen prioridades; de esta forma, si varias alarmas ocurren al mismo tiempo, primero se anunciarán las de mayor prioridad.

Villavicencio (1984) considera que el puesto de mando de una interfaz automatizada solo cuenta con consolas gráficas para presentar la información, por experiencia se sabe que son necesarias al menos tres superficies de pantalla (o tres pantallas, si se puede contar con ellas). En el caso de este trabajo se tendrá una sola pantalla y para que el usuario vea diferentes tipos de despliegue los debe seleccionar mediante el menú general que proporciona el sistema de monitoreo.

2. DESCRIPCIÓN FUNCIONAL DEL SISTEMA

El sistema para monitorear variables de proceso proporciona al usuario principalmente dos opciones:

- Configuración y
- Operación.

La configuración sirve para capturar la información necesaria del proceso, así como para especificar la manera en que se debe presentar la información. Se deberá configurar el sistema para poder utilizarlo por primera vez. Si ocurriera algún cambio en el proceso o en el criterio de agrupar las variables, se podrá reconfigurar lo necesario sin tener que hacerlo en todo el sistema.

La operación del sistema solo es posible cuando este se ha configurado. La opción de operación proporciona de diferentes formas al usuario la información del proceso, siendo lo más importante el valor de las variables.

Por otra parte, la PC, como se mencionó anteriormente, se podrá comunicar a controladores programables por medio de su red de comunicaciones. El propósito de esta comunicación es que los controladores le envíen el valor de las variables a la computadora personal por medio de un protocolo de comunicación. Este protocolo no se diseñó en este trabajo, debido a que los controladores ya cuentan con uno propio. Lo que se hizo fue desarrollar un módulo que pueda interpretarlo.

2.1 Base de datos

Antes de explicar la manera de configurar y cómo se pretende que opere el sistema se verá cómo se deberá estructurar la base de datos.

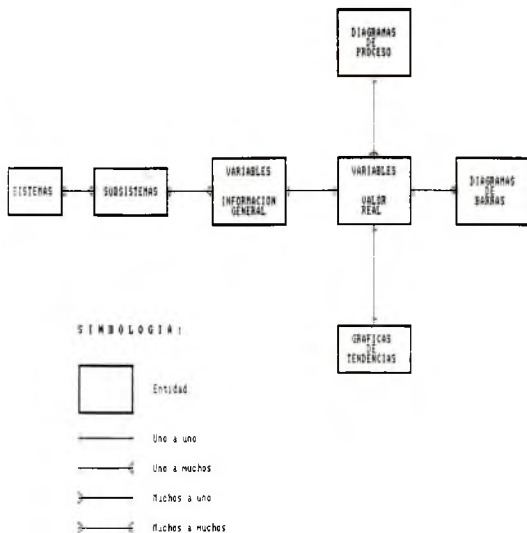


Figura 2.1 Diagrama de entidad-relación

La figura 2.1 muestra el diagrama de entidad-relación con la nomenclatura propuesta por Gane (1988). Para cada par de entidades en el diagrama que tenga una relación (entre sus elementos), la relación puede ser una a una, una a muchas, muchas a una o muchas a muchas. Este diagrama provee bastante información acerca del sistema, mostrando todas las relaciones que existen entre las entidades.

La base de datos deberá ser de tipo relacional y las entidades (ver figura 2.1) que la integran son las siguientes :

Nombre	Descripción
Sistems	Sistemas de proceso
Substms	Subsistemas de proceso
Variabl	Variables
Diagram	Diagramas de proceso
Barras	Diagramas de barras
Tenden	Graficas de tendencia

Los atributos de las entidades se muestran a continuación.

El asterisco que está a un lado del nombre del atributo indica que es un "campo llave".

Sistemas de proceso ("Sistems")

<u>Nombre</u>	<u>Descripción</u>	<u>Tamaño</u> (bytes)
* Sistema	Sistema	2
Descrip	Nombre del sistema	30

donde :

Sistema.- El sistema al que pertenece la variable. Aquí se considera que el proceso a monitorear está dividido principalmente por sistemas. Este atributo es la relación con otras entidades.

Descrip.- Nombre del sistema.

Subsistemas de proceso ("Substms")

<u>Nombre</u>	<u>Descripción</u>	<u>Tamaño</u> (bytes)
* Sistema	Sistema	2
* Substms	Subsistema	2
Descrip	Nombre del subsistema	30

donde :

Sistema.- Sistema al que pertenece el subsistema.

Substms.- El subsistema al que pertenece la variable. Aquí se considera que el proceso a monitorear está dividido principalmente por sistemas y éstos tienen subsistemas. Este atributo es la relación con otras entidades.

Descrip.- Descripción o nombre del subsistema.

Variables ("Variabl")

<u>Nombre</u>	<u>Descripción</u>	<u>Tamaño</u> (bytes)
* Identificad	Identificador	6
Descripción	Descripción completa	30
Descripcort	Descripción corta	16
* Subsistema	Subsistema	2
Tipo_senál	Tipo de señal (presión,temp)	9
Tipo_var	Tipo de variable	1
Rango	Rango de señal (4-20 mA,1-5V)	9
Prioridad	Prioridad	1
Unidades	Unidades de ingeniería(kg/cm)	1
Dispositivo	Dispositivo de origen	8
Tendencia	Tendencia de la variable	1
SP	Punto de ajuste	2
Alarma_B	Alarma crítica baja	2
Alarma_A	Alarma crítica alta	2
Color_NOR	Color de despliegue normal	2
Color_B	Color de despliegue bajo	2

donde :

- Identificad.- Identificador de variable.
- Descripción.- Descripción completa de la variable.
- Descripcort.- Descripción corta de la variable. Se puede utilizar en funciones en donde no se tenga suficiente espacio para la descripción completa.
- Subsistema.- Subsistema al que pertenece la variable. Este atributo es la relación con la entidad "sistemas".
- Tipo_señal.- Indica si la variable proviene de una señal de temperatura, presión, etc.
- Tipo_var.- Tipo de variable. Puede ser digital o analógica.
- Rango.- Rango de la señal que transmite el sensor hacia el controlador o acondicionador de señales. Por ejemplo 0 - 10 V CD.
- Prioridad.- Prioridad de la variable. Define qué tan importante es la variable.
- Unidades.- Representa las unidades de ingeniería de las variables analógicas.
- Dispositivo.- Es el dispositivo en campo que transmite la señal. Este atributo puede ser opcional.
- Tendencia.- Este atributo indica si la variable tiende a aumentar o disminuir.
- SP.- Punto de ajuste o valor normal al que se debe de encontrar de la variable.
- Alarma_B.- Estado de alarma crítica baja.
- Alarma_A.- Estado de alarma crítica alta.
- Color_nor.- Color de la variable cuando se encuentre en estado normal.
- Color_B.- Color de la variable en estado de alarma crítico bajo.
- Color_A.- Color de la variable en estado de alarma crítico alto.

Diagramas de proceso ("Diagram")

La estructura de los diagramas de proceso en disco se muestra en la figura 2.2.

Información general del diagrama	Número de variables en diag.	Primera variable en diag.	Segunda variable en diag.	n variable en diag.
----------------------------------	------------------------------	---------------------------	---------------------------	-------	---------------------

Figura 2.2 Formato del archivo de diagramas de proceso.

<u>Nombre</u>	<u>Descripción</u>	<u>Tamaño</u> (bytes)
* DiagNr	Número de diagrama	2
ScrName	Nombre del archivo PCX	13
Titulo	Titulo del diagrama	30
NVars	Número de variables en diagrama	2
* Identificad	Identificador de variable	6
Vlx	Coordenada X para desplegar	2
Vly	Coordenada Y para desplegar	2
Animacion	Animación de variable (Si/No)	1
LX	Coordenada X izquierda	2
RX	Coordenada X derecha	2
TY	Coordenada Y arriba	2
BY	Coordenada Y abajo	2
LastVal	Valor anterior	2

donde :

DiagNr.- Indica el numero de diagrama.

- ScrName.- Nombre del archivo con formato PCX que contiene el dibujo del diagrama de proceso (pantalla estática).
- Titulo.- Titulo del diagrama. Se escribirá en la parte superior del diagrama de proceso.
- NVars.- Indica el número de variables que contiene cada diagrama de proceso.
- Identificad.- Identificador de las variables que se deseen desplegar en el diagrama. Este atributo y los siguientes son para cada variable que se incluya en el diagrama.
- Vlx.- Coordenada X en donde se desplegará el valor de la variable.
- Vly.- Coordenada Y en donde se desplegará el valor de la variable.
- Animación.- Indica si se desea a la variable con animación.
- LX,RX,TY,BY.- Coordenadas izquierda, derecha, arriba y abajo de la animación.
- LastVal.- Valor anterior del actual que tiene la variable.

Diagramas de barras ("Barras")

El diagrama de barras va a tener 15 grupos de barras, donde cada grupo tiene 8 variables. Los atributos que contienen cada grupo son:

<u>Nombre</u>	<u>Descripción</u>	<u>Tamaño</u> (bytes)
* Identificad	Identificador de variable	6
LastVal	Valor anterior al actual	2

donde :

- Identificad.- Identificador de variable.
- LastVal.- Valor anterior al actual de la variable.

En este caso la entidad es de un valor fijo, a diferencia de las anteriores; Esto se debe a que, de antemano se sabe cuántos grupos se van a tener en total y cuántas variables tiene cada grupo.

Gráficas de tendencia ("Tenden")

Se contemplan 3 grupos de graficas de tendencia y cada grupo tiene tres variables. Los atributos que contiene cada grupo son:

<u>Nombre</u>	<u>Descripción</u>	<u>Tamaño</u> (bytes)
* Identificad	Identificador de variable	6
Color	El color de la tendencia	2
Elementos	Valores acumulados	75

donde :

Identificad.- Identificador de variable.

Color.- Color con el que se debe de desplegar la tendencia.

Elementos.- Valores acumulados en el transcurso del tiempo.

2.2 Configuración del sistema para monitorear variables de proceso

En la configuración del sistema de monitoreo se incorporará la información de las variables y del proceso, así como también, se diseñará la forma en que se deba presentar esta información al usuario.

Para la configuración del sistema se realizará un menú, el cual contemplará las opciones siguientes:

- Sistemas
- Subsistemas
- Variables
- Diagramas de proceso
- Gráficas de tendencia
- Diagrama de barras

En cada una de las opciones de configuración se podrá dar alta, y modificar la información previamente capturada.

2.2.1 Configuración de sistemas

Como se mencionó anteriormente, un proceso está dividido en sistemas (ver figura 2.1). Esta división estructura el proceso de una manera adecuada. Por ejemplo, en una central termoeléctrica se tienen sistemas como: turbina, generador, caldera, domo de agua, agua de alimentación y otros sistemas más.

De acuerdo con el menú indicado en el punto 2.2, la opción de configuración de sistemas tendrá las opciones de alta y modificación. En "alta" se adicionarán los sistemas que no estén registrados y en modificación se podrá cambiar la información de los sistemas previamente introducidos.

Al dar de alta o modificar un sistema, se verificará que no se repita. Esta validación ayudará al usuario a no repetir los sistemas y eliminar conflictos en el funcionamiento del programa. Lo que se tendrá que configurar es la clave del sistema y su descripción.

2.2.2 Configuración de subsistemas

Los sistemas están formados por subsistemas. Las opciones de alta y modificación se tendrán similarmente a los sistemas.

Al dar de alta o modificar un subsistema primero se deberá mencionar a qué sistema pertenece, o sea que, todos los subsistemas deberán pertenecer a un sistema previamente registrado. En seguida se introducirán los datos del subsistema, como su clave y descripción. También se validará que no se repitan los subsistemas.

2.2.3 Configuración de variables

Las variables pertenecerán a un sólo subsistema (véase la figura 1.2), por lo que al darlas de alta se deberá introducir la clave del subsistema al que pertenecen. Al hacerlo se verificará que este exista. Cuando se da de alta la variable se revisará que esta no se haya dado de alta con anterioridad. Además de lo mencionado, se dará de alta todo lo que contiene la entidad de variables ("Variabl").

2.2.4 Configuración de diagramas de proceso

La configuración de los diagramas de proceso hace que el sistema de monitoreo sea atractivo para los usuarios ya que para diseñar los diagramas no se debe realizar ninguna programación adicional.

Antes de configurar un diagrama de proceso se deberá realizar el dibujo del diagrama de proceso con cualquier editor gráfico que utilice formato gráfico PCX. De acuerdo con sus necesidades los usuarios realizarán este dibujo con las facilidades que ofrecen editores comerciales (por ejemplo "Paintbrush"). Una vez que se tiene este dibujo se procederá a configurar el diagrama. Para configurarlo se deberá introducir el nombre del archivo (PCX) que contiene el dibujo del proceso y después los identificadores de las variables que se desee observar en el diagrama. En la siguiente sección se mostrará la operación de los diagramas de proceso.

Al dar de alta una variable en un diagrama se revisará que se haya dado de alta previamente. Si el usuario lo considera conveniente y necesario será posible que una variable se repita en uno o más diagramas. Para decir donde se desea desplegar una variable bastará con posicionar el cursor con las teclas de flechas y posteriormente dar "enter". Al hacer esto, se grabarán las coordenadas donde se desplegará la variable en la estructura de datos correspondiente a diagramas de proceso.

También se tendrá la posibilidad de darle animación a la variable por medio de una barra que esté asociada al valor de la variable. Para esto se indican las coordenadas también con las teclas de flechas, dando "enter" al localizar las ubicaciones deseadas.

2.2.5 Configuración de diagramas de barras

Los diagramas de barras se mostraran en 15 grupos de 8 variables

cada grupo. En esta parte, las variables que no se den de alta no aparecerán en el diagrama. Las variables se podrán repetir en los grupos de barras. Esto podrá servir para pruebas de funcionalidad del sistema de monitoreo.

Adicionalmente se podrá dar de alta el nombre de cada grupo de barras.

2.2.6 Configuración de gráficas de tendencia

Las gráficas de tendencias estarán compuestas por tres grupos de tres variables cada uno. Cualquier variable se podrá graficar en cada grupo e incluso se podrá repetir en todos ellos. También a opción del usuario se dará de alta el título de la tendencia. Al igual que en los casos anteriores, la existencia de la variable se validará al darla de alta. Si la variable no existe, no se podrá continuar con la configuración de la tendencia.

2.3 Operación del sistema

Al principio del capítulo se mencionó que el usuario podrá seleccionar cualquiera de dos opciones principales. La primera es para configurar el sistema de monitoreo (inciso 2.2) y la segunda es la operación de éste.

La operación del sistema de monitoreo consiste en presentar la información (valor de las variables) del proceso por medio de diagramas de proceso, diagramas de barras y gráficas de

tendencia. Por supuesto que para poder presentar esta información es necesario contar con un módulo (programa) que reciba de los controladores programables el valor de las variables y lo deposite en la estructura de datos correspondiente. Al presentar esta información se toma de esta estructura de datos el valor de las variables de proceso. A continuación se definirán los diferentes tipos de despliegue. La parte de comunicación se describirá en el inciso 2.4.

Para la operación del sistema se realizará un menú principal en el cual el usuario podrá seleccionar de manera fácil y rápida la función de despliegue del sistema que se requiere.

Existen diferentes técnicas con las cuales un usuario podrá seleccionar una opción. La mayoría de éstas se conocen como menús. En un sistema de "software" debe de existir un proceso que despliegue menús al usuario y que le permita seleccionar una opción.

El menú principal que se realizara para el sistema de monitoreo es horizontal y estará ubicado en la parte superior de la pantalla. Dicho menú contendrá menús verticales "pop-down", los cuales irán apareciendo según la selección del menú horizontal. El procedimiento de uso será como sigue: El usuario moverá una barra sobresaltada que se desplaza de opción a opción en el menú horizontal. Los menús verticales aparecerán y desaparecerán bajo la opción horizontal y el usuario entonces podrá mover otra barra sobresaltada en el menú vertical que se desplazará hacia arriba y hacia abajo.

La ventaja de esta técnica es que ocupa poco espacio en pantalla proporcionando más espacio a la demás información que se pueda mostrar. Los menús verticales son ventanas "pop-down", por lo que aparecerán y desaparecerán según el desplazamiento de la opción en el menú horizontal.

Las opciones en el menú principal son las funciones de despliegue que se le proporcionarán al usuario y son:

- Diagramas de proceso
- Diagramas de barras
- Gráficas de tendencias
- Alarmas

2.3.1 Diagramas de proceso

Los diagramas de proceso son de gran ayuda para el operador de una planta industrial, ya que ellos se muestra el proceso esquemáticamente con el valor actual de las variables asociadas a éste.

Un diagrama de proceso estará compuesto por una pantalla estática y una dinámica. La pantalla estática será fija y se diseñará con la ayuda de un editor gráfico que permita escribir los archivos en formato gráfico PCX, como por ejemplo, "Paintbrush". La pantalla dinámica será la impresión de los valores de las variables de proceso y la animación de estas variables, si es que se configuró con animación (ver sección anterior). Se decidió que el usuario diseñe la pantalla estática de acuerdo con sus

necesidades específicas porque hace más flexible al sistema de monitoreo. Esto se debe a que no se tiene que realizar ninguna programación para diseñar estos diagramas.

Una vez que se haya elaborado el diagrama de proceso con la ayuda del editor gráfico se deberá almacenar con un nombre adecuado de acuerdo con su función. Posteriormente, en la configuración del sistema en la parte de diagramas de proceso se preguntará por este nombre.

El sistema siempre contará con un diagrama general de la planta y diagramas de los sistemas que la contemplan. Si el usuario lo requiere, el diagrama general y los sistemas se podrán llamar desde el menú principal.

Los sistemas también se podrán llamar desde el diagrama general por medio del número de sistema, el cual el usuario deberá escribir al estar dibujando el diagrama de proceso.

2.3.2 Diagramas de barras

Los diagramas de barras son una forma esquemática de mostrar dinámicamente con barras los valores de las variables. Esto sólo es para variables analógicas ya que las digitales sólo tienen dos valores posibles (abierto/cerrado).

En los diagramas de barras se contará con un diagrama general y diagramas de grupos. En el diagrama general se mostrará hasta un total de 15 grupos donde cada grupo es de 8 variables teniendo así hasta un total de 120 variables que se pueden mostrar en el

diagrama general y 8 por diagrama de grupo. Con esto se observa que la cantidad de información que se puede mostrar es bastante amplia.

Desde el menú principal se podrá llamar al diagrama general y a los diferentes grupos que se tienen.

Como se mencionó anteriormente, en el menú de configuración correspondiente se dará el nombre de cada diagrama de grupo y de sus variables. Si una variable no se dió de alta, la localidad que le corresponde aparecerá vacía. Es posible que no se haya dado de alta ninguna variable en diagrama de barras y no pase nada, únicamente no se mostrará barra alguna.

En el diagrama general sólo se mostrarán los valores reales de las variables con barras y el cambio de color en las barras indicará las variables en estado de alarma. Cuando se desee ver algo más descriptivo de una variable se deberá llamar al grupo en donde esta se encuentre. En las pantallas de grupos para cada variable se indicará gráficamente el valor normal o punto de ajuste, el real y sus alarmas críticas y precriticas proporcionando al usuario la información que usualmente requiere tener.

2.3.3 Gráficas de tendencia

Estas gráficas muestran el valor de las variables y su tendencia. Se mostrarán 3 gráficas por pantalla debido a que es lo común que se le muestra a los operadores. La gráfica de tendencia se

representará como una sucesión de puntos en unos ejes de coordenadas variable-tiempo.

Además de las gráficas se contará con una barra que vaya indicando el valor más reciente en la gráfica de tendencia. Esto es con el objeto de darle más claridad al usuario acerca del estado actual de la variable con respecto a sus estados anteriores en un cierto lapso de tiempo. También se mostrará información más a detalle de las variables que se estén graficando.

Cualquiera de estas graficas se podran obtener seleccionando la opción de tendencias del menú principal.

En el menú de configuración y en la opción de tendencias, se dará el nombre de cada grupo de tendencias y las variables incluidas.

2.4 Comunicación con los controladores programables

Como se menciona anteriormente, el sistema para monitorear las variables de proceso se comunicará con los controladores programables por medio de su red de comunicaciones.

El propósito principal de esta comunicación es que los controladores envíen el valor de las variables a la computadora personal. Esta comunicación se realiza por medio de un protocolo de comunicación que los controladores ya tienen implementado.

Existe una gran variedad de controladores programables en el

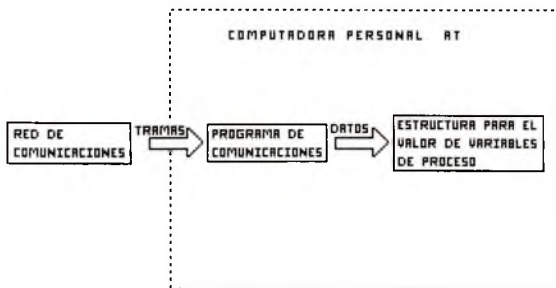


Figura 2.3 Manejador del protocolo de comunicaciones.

mercado internacional, pero en este trabajo sólo se tomaron en cuenta dos tipos de estos, que son: Controladores SAC (Sistema de adquisición y control) y los controladores de un sistema distribuido mínimo. El primer tipo se desarrolló en el IIE (Instituto de Investigaciones Eléctricas) por el Departamento de Electrónica y el segundo se desarrolló en el CINVESTAV (Centro de Investigación y de Estudios Avanzados del IPN).

Los dos tipos de controladores tienen protocolos de comunicaciones distintos y para enlazar cualquiera de ellos a la PC se deberá realizar un programa que interprete su protocolo.

El programa de comunicaciones debe interpretar los comandos que se le envían por la red de comunicaciones y contestar a ellos. Al recibir los datos (valor de las variables de proceso) se deben escribir en el vector donde se encuentran los valores de las variables (véase la figura 2.3).

El programa de comunicaciones debe manejar el UART 8250 de la PC para poder transmitir y recibir datos por su puerto serie RS-232.

Existen tres formas de realizar comunicación mediante el UART 8250 de la PC.

La primera es utilizando la rutina de comunicaciones que proporciona el BIOS. La desventaja de utilizar esta rutina es que es muy lenta, debido a que la velocidad de comunicación es de hasta 9600 bits/segundo. Otra desventaja es que los protocolos que se mencionan utilizan el bit de paridad con otro propósito y esto no lo permite la rutina del BIOS. La ventaja es que ya existen las primitivas para transmitir y recibir un dato del puerto de comunicaciones y el UART no se tiene que programar.

La segunda forma es no utilizar la rutina de comunicaciones del BIOS y programar el UART sin la rutina de interrupción de comunicaciones de acuerdo con las necesidades que se requieran. La desventaja es que se necesita tener al procesador de la PC atendiendo todo el tiempo el programa de comunicaciones y esto no es posible porque también se debe atender al usuario (desplegando información). Las ventajas son que el UART se puede programar

para que opere a velocidades superiores a los 9600 bits/segundo y se puede utilizar el bit de paridad para otro propósito.

La tercer forma y la más adecuada es no utilizar la rutina de comunicaciones del BIOS y programar el UART 8250 con la rutina de interrupción de comunicaciones. De esta manera, el procesador sólo se interrumpirá para atender las peticiones del UART 8250.

2.4.1 Protocolo de comunicaciones de los controladores SAC

El protocolo de comunicaciones de los controladores SAC es maestro-esclavo. La topología es tipo "bus" con maestro. Como se aprecia en la figura 1.1, el maestro de la red será la PC.

El protocolo utiliza la técnica de control centralizado (Valdés, Coronel y Hernández, 1990). En esta técnica, el controlador maestro ejecuta un "polling" en forma continua para permitir a los controladores esclavos la transmisión de información por la red. Los esclavos pueden disponer del canal sólo cuando el maestro lo autorice.

Cuando el maestro (PC) requiere enviar un mensaje a algún esclavo, lo hace de manera inmediata sin considerar el turno del esclavo en el "polling", aunque en este trabajo no se considera la transmisión de mensajes hacia los controladores. Los controladores son los únicos que transmitirán mensajes hacia el maestro. En estos mensajes se empaquetará el valor de las variables de proceso.

Debido a que no hay jerarquías entre los controladores esclavos,

éstos no se pueden comunicar entre si y únicamente se les permite enviar un mensaje cuando el maestro les indique que pueden disponer del canal. El número máximo de controladores esclavos es de 31 con sólo un maestro (Ibarguengoitia, 1988).

Los controladores esclavos quedan identificados en la red con una dirección lógica única, la cual se asigna por medio de puentes alambrados en sus tarjetas de comunicaciones.

Las características principales del protocolo de comunicaciones son las siguientes:

- Uso del reconocimiento positivo para la información recibida.
- Aplicación de tres retransmisiones para el caso en el que se cumpla el tiempo de espera.
- Cambio automático de canal en caso de falla, realizado por el control maestro.
- Facilidad de sincronización para actualizar el número de esclavos que operan en la red.

El protocolo maneja dos tipos de paquetes que son: paquetes de control o comandos y paquetes de datos, los cuales se muestran en la figura 2.4.

El formato de datos se divide en tres partes:

- Encabezado
- Información
- Código de detección de errores

DIRECCION DESTINO	CONTROL	DIRECCION FUENTE	CONTROL 1	CONTROL 2	CONTADOR	INFORMACION	CRC-16
1 BYTE	1 BYTE	1 BYTE	1 BYTE	1 BYTE	2 BYTES	500 BYTES MAX	2 BYTES

Formato para mensajes de información.

DIRECCION	COMANDO	DIRECCION
DESTINO (DE CONTROL)		FUENTE
1 BYTE	1 BYTE	1 BYTE

Formato para mensajes de control.

Figura 2.4 Formatos de los mensajes de los controladores SAC.

El encabezado contiene la dirección destino, un carácter de información que indica el tipo de información contenida en el

paquete, la dirección fuente que identifica al controlador que envía el mensaje, dos controles secundarios de propósito general, y un contador formado por dos bytes que indica la longitud del mensaje. De acuerdo con las características de "hardware" de la tarjeta de comunicaciones de los controladores los mensajes pueden contener hasta un máximo de 500 bytes en el campo de datos.

El código de detección de errores que se aplica es el código cíclico redundante CRC-16 (Campbell, 1987). Este se calcula para todos los bytes que forman el paquete.

El formato de control contiene la dirección destino, un comando de control y la dirección fuente. Para este caso, el comando de control se utiliza para sincronizar el flujo de datos dentro de la red.

2.4.2 Protocolo de comunicaciones de controladores con microcontrolador 8051

El desarrollo de este sistema se basa en la arquitectura "BITBUS" de Intel.

El protocolo de comunicaciones de este sistema distribuido también es maestro-esclavo debido a que se basa en la arquitectura BITBUS. En el caso de estos controladores el maestro no será la PC. La PC será otro nodo esclavo en la red de comunicaciones.

El protocolo utiliza la técnica de control centralizado. En esta técnica, el controlador maestro ejecuta un "polling" en forma continua para permitir a los controladores esclavos la transmisión de información por la red. Los esclavos pueden disponer del canal sólo cuando el maestro lo autorice.

A continuación se describe el protocolo de comunicaciones de estos controladores.

En el caso de este sistema distribuido si se permite la comunicación entre esclavos. Los mensajes podrán enviarse sólo cuando se reciba la señal de permiso dada por el maestro (PC).

Al final de cada mensaje transmitido se envia la señal de fin de mensaje hacia el nodo maestro para que este continúe el "polling".

Cada mensaje enviado por algún nodo esclavo hacia otro nodo esclavo, requiere un acuse de recibo o reconocimiento por parte del nodo destinatario, el cual se envia sólo cuando se recibe la señal de permiso.

Si el mensaje recibido está incorrecto no se envia reconocimiento y se espera volver a recibir respuesta por parte del destinatario.

Según se observa, el protocolo de comunicaciones de estos controladores es sencillo, pero suficiente para realizar la comunicación entre controladores y PC.

Dirección destino (1 byte)	Dirección fuente (1 byte)	Comando (1 byte)	Longitud (1 byte)	Datos (N bytes)	"Check-Sum" (1 byte)
-------------------------------	------------------------------	---------------------	----------------------	--------------------	-------------------------

Figura 2.4 Formato de los mensajes de los módulos controladores.

La dirección destino es la que indica el número de controlador al que va dirigido el mensaje.

La dirección fuente es el número de controlador que envía el mensaje.

El comando es el código de control.

Longitud es el número de bytes de datos que contiene el mensaje.

Los datos son los que van a continuación de la longitud.

"Check-Sum" es el código de detección de errores.

3. DISEÑO DEL SISTEMA

En este capítulo se presenta el diseño del sistema para monitorear variables de proceso. Se describirán las estructuras de datos que se utilizaron para tener la información del sistema en memoria principal y las funciones que se realizaron.

Cuando el usuario opere el sistema de monitoreo desde MS-DOS, se realizarán cuatro pasos para que el usuario pueda hacer uso de este. Los cuatro pasos están elaborados en la función "main" del programa de monitoreo y son:

- Primero, los archivos que contienen la información de configuración se cargan del disco a la memoria principal. Estos archivos son las entidades que se explican en el capítulo anterior. Al leer los archivos del disco se crean e inicializan las estructuras de datos (que son dinámicas). Las variables que se reserva memoria desde la compilación sólo se inicializan.

- Segundo, se inicializa el "driver" gráfico de Turbo C. Al hacer esto, se cambia automáticamente de modo texto a modo gráfico. Posteriormente se regresa a modo texto, a fin de mostrar el menú principal. El modo gráfico que se selecciona es el adecuado para la tarjeta gráfica que se incluye en la PC donde operará el sistema de monitoreo. El usuario debe asegurarse que los dibujos que realizó para los diagramas de proceso estén en la misma resolución.

- Tercero, se inicializa el módulo de comunicaciones. Se inicializan las variables que correspondan al programa que maneja el protocolo de comunicación de los controladores. También se configura el UART ("Universal Asynchronous Receiver Transmitter") 8250 de la PC y se instala la rutina de servicio de interrupción de comunicaciones.
- Cuarto, se crea el menú principal y los submenús, para lo cual se asigna la memoria dinámica (que se utiliza para el manejo de éstos). Después, se presenta el menú principal.

Después de realizar estos cuatro pasos el usuario estará en posibilidad de usar el sistema de monitoreo.

3.1 Menús de opciones del sistema

Es importante que a los operadores de las plantas industriales se les proporcione una herramienta (para monitorear el proceso) que sea fácil de usar, a fin de que no los distraiga de sus actividades de operación. Si se le dá al operador una herramienta de uso complejo, ésta va a distraer parte de sus actividades, aumentando así, la posibilidad de que se presente un disturbio en el proceso.

En el análisis que se realizó al sistema de monitoreo en el capítulo anterior se hizo evidente que los menús son prácticos y sencillos de usar. Este trabajo utiliza menús para presentar las diferentes opciones al usuario.

3.1.1 Menú principal

Al entrar al sistema por primera vez se tendrá un menú horizontal con las opciones que se muestran en la figura 3.1. Este menú es el principal de los que se tienen en el sistema de monitoreo. Se realizó utilizando la técnica de ventanas "pop-up" explicadas por Stevens (1987) y opera en modo texto. Este menú horizontal contiene menús verticales "pop-down" que van apareciendo según la selección del menú horizontal.

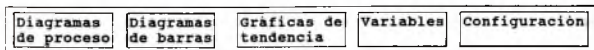


Figura 3.1 Menú principal del sistema para monitorear variables

Para la selección de las opciones que se muestran en la figura 3.1, se mueve una barra sobresaltada que se cambia de opción a opción, según la presión de cualquiera de las teclas de "flecha" (izquierda o derecha). A fin de seleccionar cualquiera de los submenús (menús verticales), basta con presionar la tecla de "enter" o la de flecha "abajo" cuando la barra sobresaltada se encuentre sobre el menú que se desea.

3.1.2 Estructura de datos de los menús

Los menús se construyen como ventanas y se controlan por una serie de tablas. Estas tablas describen la selección en la barra deslizante del menú principal (horizontal) y las que pertenecen a los menús verticales. Estas tablas se encuentran en el archivo `menú.c`.

La tabla principal es un arreglo de estructuras de tipo `MENU`, el cual se muestra a continuación:

```
typedef struct w_menu {
    char *mname;
    char *mselcs;
    void (**func) (int, int);
}MENU;
```

Este arreglo tiene una entrada por cada una de las opciones en el menú de barra deslizante. Cada opción contiene el nombre que se despliega en la opción y algunos apuntadores para describir el contenido de los menús verticales "pop-up" que están asociados a la selección horizontal. En estos apuntadores está el que apunta a los nombres de selección de los menús verticales y un apuntador a un arreglo de apuntadores de las funciones.

Al seleccionarse una opción en el menú principal se llamará a la función correspondiente apuntada por el elemento `(**func)` que es el apuntador al arreglo de apuntadores de las funciones.

3.2 Diagramas de proceso

Para invocar la función de diagramas de proceso el usuario debe posicionar la barra sobresaltada en la opción "diagramas" en el menú principal, mencionado anteriormente, y presionar la tecla "enter". Al hacerlo se presentará un submenú vertical con los nombres de los diagramas que fueron dados de alta. Si el usuario lo desea se podrá seleccionar cualquiera de estos diagramas o se podrá salir al menú principal.

A fin de desplegar los dibujos de los diagramas de proceso (pantalla estática) se utilizó el paquete de programación PCX Programmer's Toolkit, versión 4.0. Para desplegar los valores o la animación de las variables (pantalla dinámica) que se encuentran dadas de alta en el diagrama, se utiliza el "driver" gráfico del compilador Turbo C, versión 2.0.

3.2.1 La utilización de "PCX Programmer's Toolkit"

Para realizar los dibujos de diagramas de proceso se utilizó el paquete "PCX Programmer's Toolkit" debido a que contiene una biblioteca de funciones que se pueden ligar con programas objeto que estén en el lenguaje de programación C y facilitan el manejo de dibujos que estén en formato gráfico PCX (Tyler, 1989). Estas funciones están declaradas en el archivo "pcxlib.h" y las que se utilizan en este trabajo son:

- `pcxSetBuffer()`.- Con esta función se define un bloque de memoria (asignada dinámicamente con anterioridad) para

almacenar un dibujo PCX en memoria principal. En este trabajo se asignó un "buffer" de 20 kbytes para cualquier dibujo que se requiera desplegar. No se asignaron más, porque al cargarlos de disco a memoria es rápido y no se va a desplegar más de un sólo dibujo a la vez. En el caso de que fuera tardado leer el archivo de disco y después desplegarlo, se hubiera tenido que asignar más bloques de memoria. Se pasa como parámetro el apuntador al bloque de memoria asignado previamente.

- `pcxSetDisplay()`.- Se selecciona el tipo de tarjeta gráfica que se tiene. Es recomendable utilizar el sistema de monitoreo en PC's que tengan tarjetas gráficas EGA ("Enhance Graphics Adapter") o VGA ("Video Graphics Adapter") para conseguir mayor resolución en los dibujos, aunque también se puede trabajar con tarjetas CGA. A esta función se le pasa como parámetro el número que le corresponda a la tarjeta gráfica (véase `pcxlib.h`)
- `pcxSetMode()`.- Se selecciona el modo de video que se requiere. Puede ser modo texto o grafico.
- `pcxFileDisplay()`.- Despliega el dibujo con formato PCX. Se pasa como parámetro el nombre del archivo que contiene el dibujo a desplegar y las coordenadas de la pantalla donde se desplegará.

3.2.2 Estructura de datos para los diagramas de proceso

Debido a que no se tiene una cantidad de diagramas de proceso

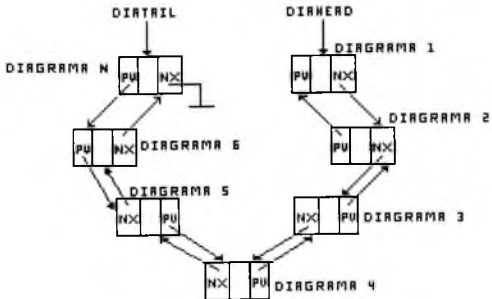


Figura 3.2 Lista doblemente ligada para diagramas de proceso

fija y su tamaño tampoco es fijo se utilizó asignación dinámica de memoria para su estructura de datos.

La estructura de datos que se usó para tener la información de los diagramas de proceso en memoria principal es una lista doblemente ligada (consúltese la figura 3.2) en donde cada nodo de la estructura es un diagrama de proceso.

La estructura es de tipo **DIAG** y los elementos que la forman son:

```
typedef struct _diag{
    char *ScrName;
    char *title;
    int NVars;
    int diagnr;
    struct _diag *_nx;
```



```

    struct _diag_ *_pv;
    DVAR      *vh;
    DVAR      *vt;
} DIAG;

```

El elemento ScrName es un apuntador a una cadena de caracteres que contiene el nombre del archivo con formato PCX en donde se encuentra la pantalla estática.

El elemento "title" es una cadena de caracteres, donde se encuentra almacenado el nombre del diagrama de proceso. Este nombre se mostrará en el submenú de diagramas de proceso y en la parte superior del diagrama cuando este aparezca en la pantalla.

El entero NVars tiene el número de variables que se van a desplegar en el diagrama.

Los elementos _nx y _pv son apuntadores que tienen la dirección del siguiente nodo y el anterior (respectivamente), de la lista doblemente ligada para los diagramas de proceso.

La información correspondiente a las variables que se incluyan en el diagrama de procesc (véase la figura 3.3) están en otra lista doblemente ligada (para información de variables). Los elementos vh y vt son apuntadores que señalan el final y el principio de esta lista de variables.

Los elementos de los nodos para la lista ligada de variables se muestran a continuación:

```

typedef struct dvar {
    char    tag[7];
    DANA    *VarPtr;

```

```

int    vlx, vly;
char  Animation[2];
int    LX, RX, TY, BY;
int    LastVal;
struct dvar *vnxt;
struct dvar *vprv;
) DVAR;

```

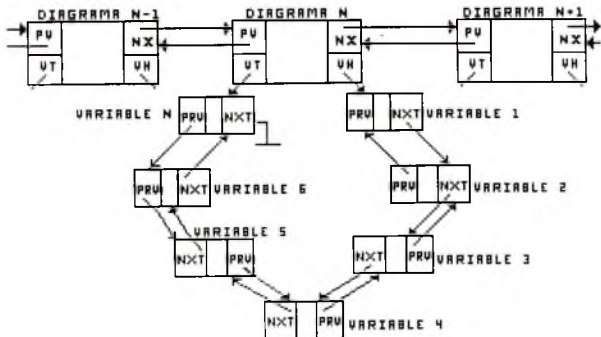


Figura 3.3 Lista doblemente ligada para variables en diagramas de proceso.

El elemento tag es el identificador de la variable y es el campo llave para poder relacionar la entidad de diagramas con la de variables.

Una vez que se haya hecho la relación entre las entidades de diagramas con la de variables es conveniente conservar el apuntador a la tupla de la variable (que en realidad será un nodo de otra lista doblemente ligada) a fin de que cuando se tenga que solicitar información de esta variable se accese rápido. El apuntador a la tupla de la variable será el elemento VarPtr.

En los enteros vlx y vly se almacenan las coordenadas en donde se desplegará el valor numérico de la variable.

El elemento "Animation" es el que indica si la variable se requiere con animación. Como se mencionó en el capítulo anterior, la animación de la variable será la representación del valor de la variable con una barra, donde el tamaño de ésta será proporcional al valor de la variable. Si la variable aumenta su valor, la barra será más grande y si disminuye, el tamaño de la barra también disminuirá. Esta barra puede representar por ejemplo el líquido que se encuentre dentro de un recipiente.

Los enteros LX, RX, TY, BY conservan las coordenadas de la barra de animación que se mencionó anteriormente. Estas coordenadas son: izquierda, derecha, arriba y abajo respectivamente, y solo tendrán un valor en el caso de que el elemento "Animation" esté en verdadero.

LastVal es el entero que contiene el valor de la variable que se encuentra dibujado en la pantalla. Cuando se actualiza el tamaño de la barra de animación no se dibuja otra vez toda la barra, lo que se hace es: En caso de que el valor de la variable sea menor,

sólo se borra lo necesario y en caso de que el valor haya aumentado, se dibuja lo que falte de la barra .

Los elementos vnxt y vprv son apuntadores que señalan el elemento previo y el siguiente, respectivamente, de la lista doblemente ligada de las variables.

3.2.3 Biblioteca de funciones para diagramas de proceso

void diags(int hsel, int vsel)

Esta es la función principal para desplegar un diagrama de proceso en operación. Primero revisa si existe el diagrama, y si se trata del diagrama general o cualquier otro. Si el diagrama no existe muestra un mensaje de error. Puede llamar a DrwMDiag() o DrwDiag().

DrwMDiag()

Esta función despliega el diagrama general (o primero de la lista ligada).

DrwDiag(DIAG *diag)

Esta función despliega el diagrama de proceso diag. El diagrama no debe de ser el diagrama general.

void DispVals (DIAG *diag)

Esta función despliega los valores de las variables en el diagrama de proceso.

void InitDVars (DIAG *diag)

Esta función inicializa un diagrama de proceso, para que pueda

tener variables asociadas. Establece una lista ligada de estructuras de tipo DVAR, primero limpiando cualquier descripción existente. Una estructura de tipo DVAR describe las características de una variable incluida en un diagrama de proceso. Si se encuentra una variable establecida en el diagrama, se borra; por lo tanto, se utiliza cuando ya no se desea utilizar el diagrama porque se libera la memoria que pertenece a las estructuras de tipo DVAR.

DVAR *EstablishDVar (void)

Esta función establece una variable en un diagrama de proceso que se ha inicializado por InitDVars.

void near AddeVar (DIAG *diag, DVAR *varbl)

Añade una variable al final de la lista doblemente ligada de las variables que contiene el diagrama.

void near AddBVar (DIAG *diag, DVAR *varbl)

Agrega una variable al inicio de la lista doblemente ligada de las variables.

void near DeleteVar (DIAG *diag, DVAR *varbl)

Esta función elimina la variable varbl del diagrama diag.

void InitDiag (void)

Esta función inicializa los apuntadores de la lista doblemente ligada de diagramas de proceso. Si se encuentra algún diagrama establecido en la lista, se borra; por lo tanto, esta función se utiliza al principio y al fin del programa de monitoreo.

DIAG *EstablishDiag (void)

Esta función asigna memoria para una estructura de tipo DIAG e inicializa los elementos que contienen esta estructura.

void AddEDIag (DIAG *diag).

Añade un nodo al final de la lista de diagramas de proceso.

void AddBDiag (DIAG *diag)

Añade un nodo al principio de la lista de diagramas de proceso.

void DeleteDiag (DIAG *diag)

Esta función elimina el nodo diag de la lista doblemente ligada de diagramas de proceso.

int VerDiag (DIAG **dl)

Verifica la existencia del nodo dl en la lista doblemente ligada.

WINDOW *ConfDiaWnd (int op)

Esta función es para dibujar las ventanas que se utilizan para dar de alta o modificar diagramas de proceso. También dibuja las ventanas para dar de alta o modificar las variables en el diagrama de proceso.

EstDiaFlds (WINDOW *wnd, DIAG *diag)

Establece los campos en la ventana para introducir los datos correspondientes al diagrama de proceso.

void SavDiag (void)

Esta función salva a disco la lista doblemente ligada de diagramas de proceso en el archivo DIAGRAM.DAT. Se salva el

primer diagrama (primer nodo de la lista) y posteriormente las variables asociadas a éste (vease la figura 3.3). Después se salva el segundo y así sucesivamente hasta el último diagrama de la lista.

void LoadDiag (void)

Lee en el disco los diagramas de proceso con sus variables y crea las estructuras de datos para tenerlos en memoria principal.

DiaCopy (DIAG *dest, DIAG *src)

Copia la estructura src a dest. Todos los elementos que se encuentren en la estructura fuente son copiados a la otra.

void AddDiag ()

Esta función es la principal para adicionar un diagrama de proceso. Es llamada desde el menú principal cuando el usuario selecciona la opción de adicionar un diagrama.

getvarbls(DIAG *diag)

El propósito de esta función es conocer cuántas variables va a tener un diagrama y adicionarlas a su lista ligada de variables. Le presenta al usuario una ventana donde pueda ir dando de alta las variables que se requiera desplegar en el diagrama. También revisa si la variable va a tener animación.

getxy(DIAG *diag, DVAR *varbl)

Esta función cambia del modo texto al modo gráfico y despliega el diagrama de proceso para que el usuario lo vea e indique las coordenadas en donde desea que se despliegue la variable y las coordenadas de la animación si es que se tienen.

CoordXY (int *GetX, int *GetY, void *cursor)

Esta función mueve el cursor en el diagrama que despliega la función getxy de acuerdo con las teclas de flechas que presione el usuario. Regresa cuando se presiona la tecla Esc o Enter.

VarCopy (DVAR *dest, DVAR *src)

Esta función copia los elementos de la estructura src a los elementos de la estructura dest. El tipo de las estructuras es DVAR y son para describir a las variables que estarán en diagramas de proceso.

ShowDiag(DIAG *diag)

Esta función despliega el diagrama de proceso diag. Salva la pantalla de modo texto, cambia a modo gráfico, despliega el diagrama diag, espera a que se presione cualquier tecla y por último cambia a modo texto y recupera la pantalla salvada.

void ChnDiag(void)

Esta función es la principal para modificar un diagrama de proceso. Lo primero que hace es revisar si existe el diagrama; si es así, muestra su información en una ventana. Esta ventana tiene opciones para cambiar el diagrama que se está mostrando, las cuales son: mostrar el diagrama previo de la lista ligada, el siguiente, el primero, el último o cualquier diagrama. Además de las opciones para cambiar el diagrama se tienen las opciones de salvar la información y para dar de alta otro diagrama.

ChnVars (DIAG *diag)

Esta función sirve para poder dar de alta o modificar variables

en diagramas de proceso. Es llamada cuando se esté modificando un diagrama por la función ChnDiag().

3.3 Diagramas de barras

Cuando el usuario requiera ver diagramas de barras, se debe posicionar la barra sobresaltada en la opción de "Barras" del menú principal (véase la figura 3.1) y presionar la tecla de "enter". Al hacerlo se presentará un menú vertical en donde se podrá seleccionar el diagrama de barras general o cualquiera de los 15 que puede tener el sistema. Si el usuario lo desea, es posible que no se seleccione ninguno y se regrese al menú principal.

Los diagramas de barras están en modo gráfico, por lo que al seleccionar cualquiera de éstos, se cambia de modo texto a modo gráfico.

3.3.1 Matriz de grupos de barras

Para tener en memoria principal la información correspondiente a la configuración de los diagramas de barras, se utiliza una matriz de estructuras (véase la figura 3.4). Cada estructura que está en la matriz es un grupo de variables y su tamaño es de 3 por 5 estructuras. La matriz corresponde a la presentación del diagrama general de barras, es decir, se van a presentar 3 líneas de 5 columnas cada una. Cada elemento de la matriz será un grupo de variables.

La matriz (declarada BGRP bars[3][5]) tiene memoria asignada desde la compilación, ya que su tamaño es fijo y pequeño.

Elemento número 1	Elemento número 2	Elemento número 3	Elemento número 4	Elemento número 5
Elemento número 6	Elemento número 7	Elemento número 8	Elemento número 9	Elemento número 10
Elemento número 11	Elemento número 12	Elemento número 13	Elemento número 14	Elemento número 15

Figura 3.4 Matriz para diagramas de barras

La estructura que representa un elemento de la matriz y es un grupo de variables es:

```
typedef struct bgrp {
```

```
        char   Title[15];
        BVar   Vars[8];
} BGRP;
```

El elemento Title es una cadena de caracteres que contiene el título del grupo de variables de barras.

El elemento Vars es un arreglo de ocho estructuras de tipo BVAR, que abarca la información de configuración de cada variable incluida en cada grupo de barras. Los elementos de esta estructura son los siguientes:

```
typedef struct bvar {
    char   tag[7];
    DANA   *VarPtr;
    int    LastVal;
} BVAR;
```

El elemento tag es el identificador de la variable y es el campo llave para poder relacionar la entidad de barras con la de variables.

Una vez que se haya hecho la relación entre las entidades de barras con la de variables se apunta a la tupla de la variable, para que, cuando sea necesario solicitar información de esta variable se accese rápido. El apuntador a la tupla de la variable será el elemento VarPtr.

LastVal es el elemento que sirve para salvar el valor con el cual se dibujó la barra. Si el valor siguiente aumenta, sólo se dibuja lo faltante, y si disminuye, se borra lo que sobra de la barra.

3.3.2 Biblioteca de funciones para diagramas de barras

void InitBars(void)

Esta función inicializa la matriz que se ocupa para diagrama de barras.

void barras(int hsel, int vsel)

Esta es la función principal para desplegar los diagramas de barras. Es llamada desde el menú principal y puede desplegar el diagrama general de barras o a un grupo de barras, dependiendo de la opción que se haya seleccionado.

void DrwGScr (BGRP *grp)

Esta función dibuja en pantalla el marco de un grupo de barras. Además escribe los títulos necesarios en la pantalla.

void DrwMScr(void)

Dibuja el marco del diagrama de barras general, y escribe los títulos y la fecha en la pantalla.

void DrwAllErs(void)

Esta función dibuja todas las barras del diagrama general.

void DrwBars(BGRP *grp, int x, int y)

Dibuja un grupo de barras del diagrama general. Si el valor de la variable aumenta, sólo se dibuja lo que falte de la barra, y si el valor disminuye, se borra lo que sobre de la barra.

void DrwGrp(BGRP *grp)

Dibuja las barras de un grupo de barras.

int ReadNum (void)

Es llamada desde la función DrwAllBars() y lee de teclado el número del grupo de barras que se desea ver.

WINDOW *ConfBarWnd (int op)

Esta función despliega las ventanas para dar de alta o modificar la información de los diagramas de barras.

void AddBars ()

Es la función principal para modificar o dar de alta una variable en los diagramas de barras. Se muestra una ventana con las variables que contiene un grupo de barras y opciones para cambiar el grupo que se despliega y salvar la información del grupo.

BarCopy (BGRP *dest, BGRP *src)

Esta función copia la información de un grupo de barras en otro grupo.

void SavBars (void)

Salva de la memoria principal a disco toda la matriz que se utiliza para los diagramas de barras.

void LoadBars (void)

Esta función sirve para leer de disco el archivo "barras.dat" que contiene la información de los diagramas de barras. Después de leerlos escribe la información en la matriz de diagramas de barras.

int ValidVar (char *bf)

Esta función es para validar si existe la variable que se quiere incluir en un grupo de barras. La validación consiste en

recorrer la lista doblemente ligada de variables y revisar su existencia ahí.

3.4 Gráficas de tendencia

De forma similar a los casos anteriores, para seleccionar gráficas de tendencias el usuario debe posicionar la barra sobresaltada del menú principal en la opción de "Tendencias" (véase la figura 3.1). Al hacerlo se mostrará un menú vertical en donde se presentarán tres submenús equivalentes a tres grupos de tendencias.

Cada grupo de tendencias es de tres variables y por cada una se tiene una gráfica de tendencia y una barra. La barra indica el valor más actual de la variable. La gráfica de tendencia es una sucesión de puntos en la pantalla que muestra el comportamiento de una variable a través del tiempo. La coordenada Y representa el valor de la variable y la coordenada X el tiempo. En el lado derecho de la tendencia se encuentra el valor más actualizado.

La impresión de estos puntos son cada cierto periodo de tiempo y en el caso de este trabajo se estableció que fuera cada segundo.

Para almacenar en memoria principal la información relacionada con las gráficas de tendencias se seleccionó una "cola circular" (véase la figura 3.5). Esta "cola circular" puede tener hasta 75 elementos, donde cada uno es un valor de la variable a la que corresponda la tendencia. Se dispuso que fueran 75 elementos

porque se van a dibujar hasta 75 líneas por tendencia, debido a la manera en que se dibujan en la pantalla.

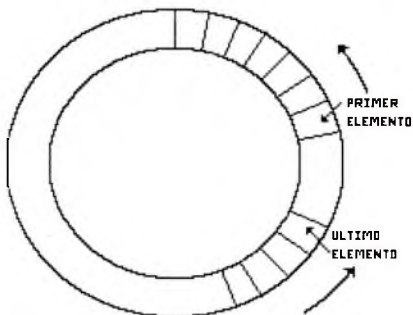


Figura 3.5 Cola circular para tendencia de variable.

3.4.1 Colas circulares para gráficas de tendencia

Para dibujar una tendencia, se toma el elemento final de la cola y el anterior a éste. Estos dos elementos corresponden a las coordenadas en el eje "Y" de la pantalla para dibujar una línea. Las coordenadas "X" se irán desplazando 5 píxeles por cada elemento. Para dibujar la línea siguiente se tiene que tomar de la cola circular el elemento anterior al último que se leyó y entre estos dos se dibujara otra línea, tomando en cuenta el

corrimento de 5 pixeles por línea. Esto se repite hasta haber dibujado toda la tendencia. Debido a que cada segundo se introducirá un valor a la cola circular, un segundo será la distancia de 5 pixeles en el eje X.

La memoria para las "colas circulares" se asigna en la compilación, debido a que el tamaño de éstas es fijo y no ocupa gran espacio.

La definición de la cola circular es la siguiente :

```
typedef struct {
    char tag[7];
    DANA *VarPtr;
    int first,last;
    int color;
    int elements[MAXELEMENTS];
} TAIL;
```

El elemento tag es una cadena de caracteres que contiene el identificador de la variable. Este elemento establece la relación entre la cola circular y la entidad de variables.

Como en los casos anteriores, el elemento VarPtr es el apuntador al nodo de la lista doblemente ligada de las variables.

Los enteros "first" y "last" son para controlar el principio y el fin de la cola.

En el elemento "elements" se almacenan los valores de la variable a la que está asociada la tendencia.

3.4.2 Biblioteca de funciones para gráficas de tendencia

void TailInit(TAIL *t, int color)

Esta función inicializa los elementos de la cola circular "t".

EmptyTail (TAIL *t)

Esta función regresa verdadero, si la cola circular t está vacía; en caso contrario, regresa falso.

FullTail (TAIL *t)

Regresa verdadero si la cola circular está llena; en caso contrario, regresa falso.

void TailIns (TAIL *t, int v)

Esta función introduce el valor v al final de la cola circular.

VAL TailGive (TAIL *t)

Saca el primer elemento que se introdujo a la cola circular y lo regresa. Actualiza los enteros de control de la cola circular.

void DrwBdrT(int group)

Esta función dibuja el marco de la pantalla para gráficas de tendencia, así como los títulos correspondientes a la gráfica.

void DrwTndncs(void)

Dibuja tres gráficas de tendencias y tres barras que indican el último valor de la variable.

void DrwBarT(int x, int y, int lastval, int val, int color)

Dibuja la barra que indica el valor actual de la variable.

void DrwTndnc(TAIL *ten, int yy)

Dibuja una gráfica de tendencia que representa el valor de una variable a través del tiempo.

void ActTndnc(TAIL *t, int val)

Inserta el elemento val a la cola circular t. En el caso de que la cola circular esté llena, no se realiza nada.

void Inittndn (void)

Esta función inicializa las tendencias y las variables que se emplean.

void Tندن(int hsel, int vsel)

Es la función esencial para desplegar gráficas de tendencia; es llamada desde el menú principal.

void AddTندn ()

Esta es la función principal para adicionar una variable a gráficas de tendencia, es decir, sirve para que el sistema realice la tendencia de una variable.

TندnCopy (TAIL *dest, TAIL *src)

Copy los valores de la tendencia src a los elementos de dest.

void SavTندn (void)

Escribe a disco la información que contengan las colas circulares de las tendencias; se escribe al archivo Tندن.dat.

void LoadTندn (void)

Esta función lee del disco la información de las gráficas de tendencia y la escribe a las colas circulares correspondientes.

3.5 Variables de proceso

La información de configuración perteneciente a las variables de proceso se almacena en disco en la entidad VARIABL y en memoria principal en una lista doblemente ligada. Para almacenar los valores de las variables que se adquieran de los controladores por el puerto serie de la PC se tendrá un vector de estructuras que se explicarán más adelante.

3.5.1 Información general de variables de proceso

A continuación se muestran los elementos que contiene cada nodo de la lista doblemente ligada para la información de las variables.

```
typedef struct _dana_ {
    char tag [ 7];
    char descripción[31];
    char descripcort[16];
    char subsistema [ 3];
    char tipo senal [ 9];
    char rango [ 9];
    char dispositivo[ 9];
    char estadonorml[ 7];
    char prioridad [ 3];
    char estado_alar[ 7];
    char color_nor [ 3];
    char color_a [ 3];
    int ColNormal, ColAlarma;
    int dananr;
    struct _dana_ *_nx;
    struct _dana_ *_pv;
} DANA;
```

El elemento tag contiene el identificador de la variable. Con este elemento se realizan las relaciones con las demás entidades

y para poder consultar la información de las variables.

Los demás elementos, a excepción de los siguientes, corresponden a los campos que se explicaron en el capítulo anterior.

El entero `dananr` sirve para indicar el número de la variable. Este número se va asignando conforme se vayan adicionando a la lista.

Los apuntadores `_nx` y `_pv` contienen la dirección del siguiente nodo de la lista y del anterior.

3.5.2 Valor de las variables de proceso

El valor de las variables de proceso se almacena en un vector de estructuras. Se decidió que este vector fuera así, para que cuando se requiera leer un valor de alguna variable se haga con el índice que tenga en el vector. El índice lo tiene la estructura anterior.

Cada elemento del vector incluye lo siguiente:

```
typedef struct _dana_ {
    char tag [ 7];
    int value;
} DANA;
```

El elemento `tag` contiene el identificador de la variable. "Value" es el elemento que contiene el valor actual de la variable. Para desplegar el valor de la variable se lee este entero.

3.5.3 Biblioteca de funciones para variables

void InitDana (void)

Esta función establece una lista ligada de estructuras de tipo DANA, primero limpiando cualquier descripción existente. Una estructura de tipo DANA describe las características de una variable. La función revisa si se encuentra una variable establecida en la lista ligada, se borra; por lo tanto, se utiliza cuando ya no se desea tener las variables en memoria principal porque se libera la memoria que pertenece a las estructuras de tipo DANA.

DANA *EstablishVana (void)

Esta función asigna memoria para una estructura de tipo DANA e inicializa los elementos que contienen esta estructura.

void AddEDana (DANA *dana)

Adiciona una variable al final de la lista doblemente ligada de variables.

void AddBDana(DANA *dana)

Agrega una variable al final de la lista doblemente ligada de variables.

void RemDana(DANA *dana)

Elimina una variable de la lista doblemente ligada de variables.

void InsDana (DANA *v1, DANA *v2)

Esta función inserta la estructura v1 después de la estructura v2 en la lista doblemente ligada de variables.

int VerifyDana (DANA **v1)

Verifica la existencia de una variable en la lista doblemente ligada de variables.

void LoadVarAna(void)

Esta función lee de disco la información de las variables y las va escribiendo conforme las lea en la lista doblemente ligada de variables. Se lee del archivo con nombre variabl.dat.

void AddVarAna(void)

Esta función es la principal para dar de alta una variable. Se presenta una ventana con los campos que debe llenar el usuario.

void ChnVarAna(void)

Esta función es para modificar la información de una variable. Se presenta una ventana con diferentes opciones para que el usuario pueda modificar la información de una manera rápida y sencilla.

void SavVarAna(void)

Salva la lista doblemente ligada de variables a disco.

DanCopy (DANA *dest, DANA *src)

Copia la estructura src de tipo DANA a la estructura dest.

int ValidDana (char *bf)

Esta función checa que al dar de alta o al modificar una variable no se repita.

3.6 Sistemas de proceso

3.6.1 Estructura de datos de sistemas de proceso

Los sistemas de proceso se almacenan en disco en el archivo llamado `systems.dat`. En memoria principal se almacenan en una lista doblemente ligada, donde cada nodo es un sistema de proceso. La estructura de datos es similar a las que se han explicado anteriormente. Los elementos de la estructura son:

```
typedef struct _sist_ {
    char sistema [ 3];
    char descrip [31];
    int sistnr;
    struct _sist_ *nx;
    struct _sist_ *pv;
} SIST;
```

El elemento `sistema` es la clave del sistema. Con este elemento se relaciona a los subsistemas.

El elemento `descrip` contiene el nombre del sistema de proceso.

El entero `sistnr` indica el número de sistema que le corresponde en la lista doblemente ligada. Este número se asigna cuando se vayan dando de alta.

Los elementos `nx` y `pv` son apuntadores que contienen la dirección del nodo siguiente y previo de la lista ligada.

3.6.2 Biblioteca de funciones para sistemas

```
void InitSist (void)
```

Esta función inicializa la lista doblemente ligada para sistemas

de proceso. Si se encuentran sistemas en la lista, se borran. También se inicializan los apuntadores a la lista.

SIST *EstablishSist (void)

Se reserva espacio en memoria principal para una estructura de tipo SIST. Esta estructura es para los sistemas de proceso.

void AddESist (SIST *sist)

Adiciona la estructura sist de tipo SIST al final de la lista de sistemas de proceso.

void AddBSist (SIST *sist)

Agrega la estructura sist al inicio de la lista de sistemas de proceso.

void RemSist(SIST *sist)

Esta función elimina una estructura de la lista doblemente ligada de sistemas de proceso.

void InsSist (SIST *v1, SIST *v2)

Inserta la estructura v1 después de la estructura v2 en la lista doblemente ligada.

int VerSist (SIST **v1)

Verifica que no se repita el sistema cuando se esté dando de alta o se esté modificando.

void LoadSist(void)

Esta función lee de disco la información que pertenece a los sistemas de proceso y la escribe en la lista ligada de sistemas. Cada sistema es una estructura en la lista. El archivo que se lee

tiene por nombre Systems.dat.

void AddSist (void)

Esta función sirve para dar de alta un sistema de proceso. Muestra al usuario una ventana con varias opciones para facilitar el alta de sistemas.

void ChnSist(void)

La utilidad de esta función es modificar la información que pertenece a un sistema de proceso.

void SavSist(void)

Esta función salva a disco los sistemas de proceso que se encuentren en la lista doblemente ligada de sistemas.

int ValidSist (char *bf)

Valida que no se repita el sistema cuando se dé de alta o se modifique su información.

SisCopy (SIST *dest, SIST *src)

Copia la estructura src de tipo SIST a la estructura dest del mismo tipo.

3.7 Subsistemas de proceso

3.7.1 Estructura de datos para subsistemas de proceso

Los subsistemas de proceso se almacenan en disco en el archivo llamado substms.dat. En memoria principal se almacenan en una

lista doblemente ligada similar a la de sistemas. Cada nodo es un subsistema de proceso. Los elementos de la estructura son:

```
typedef struct _subs_ {
    char sistema [ 3];
    char subsist [ 3];
    char descrip [31];
    int subsnr;
    struct _subs_ *nx;
    struct _subs_ *pv;
} SUBS;
```

El elemento sistema es la clave del sistema al que pertenece el subsistema.

El elemento descrip contiene el nombre del subsistema.

El entero subsnr indica el número de subsistema que le corresponde en la lista doblemente ligada. Este número se asigna cuando el subsistema se vaya dando de alta.

Los elementos nx y pv son apuntadores que contienen la dirección del nodo siguiente y previo de la lista ligada.

3.7.2 Biblioteca de funciones para subsistemas

void Initsubs (void)

Esta función inicializa los apuntadores de la lista doblemente ligada de subsistemas de proceso. Si se encuentra algún subsistema establecido en la lista, se borra; por lo tanto, esta función se utiliza al principio y al fin del programa de monitoreo.

SUBS *EstablishSubs (void)

Esta función asigna espacio en memoria principal para una

estructura de tipo SUBS. Además inicializa los elementos de la estructura.

void AddSubs (SUBS *subs)

Adiciona una estructura al final de la lista de subsistemas.

void AddBSubs (SUBS *subs)

Agrega un elemento al principio de la lista doblemente ligada de subsistemas.

void RemSubs(SUBS *subs)

Elimina una estructura de la lista de subsistemas de proceso.

void InsSubs (SUBS *v1, SUBS *v2)

Inserta el subsistema v1 después del v2 en la lista doblemente ligada de subsistemas.

int VerSubs (SUBS **v1)

Verifica la existencia del subsistema v1 en la lista ligada de subsistemas.

void LoadSubs(void)

Esta función carga de disco a memoria principal los subsistemas de proceso. Cada vez que se lee un subsistema del archivo Subsist.dat, se escribe en un nodo de la lista ligada de subsistemas.

void AddSubs (void)

Esta función se llama desde el menú principal para dar de alta un subsistema.

void ChnSubs (void)

Sirve para modificar la información de un subsistema. Se presenta una ventana al usuario con la información del subsistema y con opciones que hacen fácil y rápida esta tarea.

void SavSubs (void)

Salva la información de subsistemas que está en la lista ligada de subsistemas a disco.

int ValidSubs (char *bf)

Esta función se encarga de revisar que no se repita el subsistema cuando se esté dando de alta o modificando.

SubCopy (SUBS *dest, SUBS *src)

Copia todos los elementos de la estructura src a los elementos correspondientes de la estructura dest.

4. APLICACIÓN DEL SISTEMA

La funcionalidad del sistema de monitoreo se efectuó de dos maneras:

- Simulación de variables de proceso
- Enlace con controladores programables con microcontrolador 8051

El estudio que se realizó a los controladores programables SAC (sección 2.4.1) sirvió para determinar que si es posible el enlace de la PC con estos controladores.

Se describió su protocolo de comunicación pero no se realizó el programa para manejarlo debido a que no fue posible contar con este equipo.

4.1 Simulación de las variables de proceso

Aunque fue de manera simple, la simulación de las variables de proceso sirvió para probar la funcionalidad del sistema de monitoreo.

En lugar de enlazar la PC a los controladores programables (para que estos transmitieran el valor de las variables) con la rutina de servicio de interrupción de reloj de la PC se generan números aleatorios que simulan el valor de las variables (véase la figura 4.1).

Para la generación de números aleatorios se utilizó la función random () de la biblioteca de funciones de Turbo C. Se simularon

hasta 120 variables.

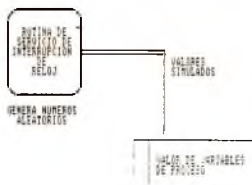


Figura 4.1 Generador de valores simulados de variables de proceso

4.2 Descripción del proceso de ejemplo

Otra prueba que se realizó para verificar la funcionalidad del sistema de monitoreo fue enlazar la PC a un sistema distribuido desarrollado por Tinoco (1990) en el CINVESTAV. La finalidad de enlazar el sistema de monitoreo con los controladores es para poder ver mediante diferentes despliegues de información la operación del proceso.

Este sistema distribuido está formado por tres módulos controladores que se comunican por medio de una red de comunicaciones.

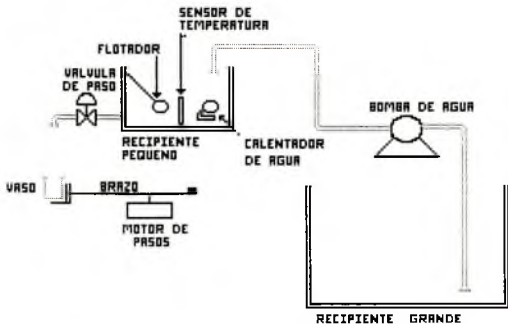


Figura 4.2 Proceso de calentamiento de agua

A fin de probar este trabajo, los módulos controladores controlan un sistema que proporcione un vaso de agua a una cierta temperatura. El proceso de calentamiento de agua es el que el usuario requiere ver en la pantalla de la PC.

Este sistema opera de la siguiente manera:

El agua se translada de un recipiente donde está almacenada hacia uno donde se calienta a la temperatura deseada (véase la figura 4.2).

Ya caliente se vacía en un vaso en la cantidad adecuada.

El sistema contiene dos recipientes de agua, uno mayor que el otro.

Como se aprecia en la figura 4.1 en el recipiente mayor se encuentra una bomba de agua, que la bombea hacia el depósito menor.

En el recipiente donde se calienta el agua, que es el de menor tamaño, se encuentra un flotador, un calentador, un sensor de temperatura y una válvula de paso.

Además de lo que hay en los dos recipientes se tiene un motor de pasos.

4.1 Protocolo de comunicaciones

El protocolo de comunicaciones que se utilizó fue el que tiene los controladores del sistema distribuido descrito en la sección 2.4.2.

La transmisión y recepción de mensajes se programó con la rutina de servicio de interrupción (número 12) de comunicaciones.

4.4 Configuración del sistema

Se configuró el sistema de monitoreo y se notó que para el ejemplo de aplicación fueron suficientes los desplegados con que cuenta este sistema.

Debido a la simplicidad del ejemplo de aplicación solo se dió de

alta un sistema y un subsistema.

Se dieron de alta las variables de proceso que transmitieron los controladores.

Se realizó el dibujo del diagrama de proceso en el editor gráfico "Paintbrush", el cual es similar al que se muestra en la figura 4.2.

Se dió de alta un diagrama de proceso que funciona como el diagrama general de proceso. Además se dieron de alta las variables que contiene este diagrama.

En los diagramas de barras se dieron de alta las variables.

En las gráficas de tendencia se dieron de alta las variables.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Fue posible el monitoreo de señales de procesos industriales mediante una computadora personal AT en ambiente MS-DOS.

Resultó útil que el sistema de monitoreo sea configurable ya que se puede utilizar fácilmente en distintas aplicaciones.

La comunicación del sistema de monitoreo se hizo posible con los controladores programables mediante la rutina de servicio de interrupción de comunicaciones.

Para este tipo de trabajos es más eficiente no utilizar la rutina de comunicaciones que proporciona el BIOS y programar el UART 8250 de la PC con la rutina de interrupción de comunicaciones.

La velocidad de desplegar los diagramas satisface los requisitos de operación en plantas industriales con 120 variables.

Se pueden ir almacenando los valores de las variables para las gráficas de tendencia con la rutina de interrupción de reloj.

La PC se puede comunicar con otros controladores programables desarrollando únicamente el protocolo de comunicaciones.

La utilización del formato gráfico PCX fue de gran utilidad porque se ahorra memoria y el despliegue de dibujos es muy rápido.

El modelo relacional es eficiente y fácil de implementar.

Las listas doblemente ligadas son de gran utilidad cuando no se tiene un número fijo de estructuras de datos y se requiere un crecimiento de estructuras sin recompilar el programa fuente.

5.2 Recomendaciones

Se puede realizar un editor gráfico con iconos para ampliar el trabajo. El editor debe ser con formato gráfico PCX a fin de que no se alteren las rutinas donde se despliegan los dibujos.

Se pueden desarrollar otros protocolos de comunicación para poder enlazar el sistema de monitoreo con otros controladores programables.

R E F E R E N C I A S

Campbell Joe, 1987

"C Programmer's Guide to Serial Communications"

Editor: Marie Butler-Knight

Howard W. Sams & Company

4300 West 62nd Street, Indianapolis, Indiana 46268 USA

Gane Chris, 1988

"Rapid System Development"

Diciembre de 1988, primer capitulo,

Prentice-Hall, Inc.,

Englewood Cliffs, NJ.

Ibarquengoytia, G. P., Jiménez, H. D., 1988,

"Sistema de comunicaciones para el sistema de control distribuido de la planta de Valle de México",

Artículo Numero I-RVP-88-GE-24,

Primera Reunión de Verano del Capítulo de Potencia (IEEE).

Acapulco, Guerrero, julio 31-agosto 4, 1988.

Shinsky F.G., 1981

"Controlling Multivariable Process"

Editor: Paul W. Murrill, Ph. D.

Instrument Society of America,

67 Alexander Drive, North Carolina 27709.

Stevens Al, 1987

"C Data Base Development"

Primera impresión,

Management Information Source, Inc.

1107 N.W. 14th Avenue, Portland, Oregon 97209.

Stevens Al, 1988

"Memory-Resident Utilities Screen I/O and Programming Techniques"

Primera impresión

Management Information Source, Inc.

1107 N.W. 14th Avenue, Portland, Oregon 97209.

Tinoco Alvarado Alejandro, 1990

"Sistema distribuido con microcontroladores de la línea 8051"

Centro de Investigación y de Estudios Avanzados del IPN,

México, D.F.

Tyler Bert, 1989

"Mastering the PCX Format"

BYTE

Editor Fred Langa,
Publicación mensual,

McGraw-Hill, Inc.,

Septiembre de 1989, páginas 183 a 187.

Ullman Jeffrey D., 1982

"Principles of Database Systems"

Segunda edición

Computer Science Press, Inc.

11 Taft Court, Rockville, Maryland 20850

Valdés R. Ricardo, Hernández P. Carlos, Coronel P. Martín, 1990,
"Communications Network in a Distributed Control System for the
Fossil Power Plant Dos Bocas, Ver. Mexico."

Proceedings of ISA.

1990 POWID Symposium - Toronto, Canada - mayo 21 a 23.

67 Alexander Drive, North Carolina 27709

Villavicencio Alejandro, 1984,

"La interfaz hombre-máquina del SADRE"

Boletín IIE

Editor: Gabriel Nagore,
Cuernavaca, Mor.

Publicación bimestral,

Julio/agosto de 1984, páginas 186 a 197.

Villavicencio Alejandro, 1984,

"Sistema de adquisición de datos y registro de eventos (SADRE)"

Boletín IIE

Editor: Gabriel Nagore,
Cuernavaca, Mor.

Publicación bimestral,

Marzo/abril de 1984, páginas 69 a 80.

Williams Theodore J., 1984

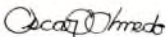
"The Use of Digital Computers in Process Control"

Editor: Paul W. Murrill, Ph. D.

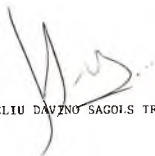
Instrument Society of America

67 Alexander Drive, North Carolina 27709

EL JURADO DESIGNADO POR LA SECCION DE COMPUTACION DEL DEPARTAMENTO DE INGENIERIA ELECTIRICA APROBO LA TESIS "DESARROLLO DE UNA INTERFAZ DE USUARIO PARA UN SISTEMA DE CONTROL DISTRIBUIDO" PRESENTADA POR EL ING. RICARDO OTHON VALDES RAMIREZ, EL DIA 24 DE ENERO DE 1991



M. EN C. JOSE OSCAR OLMEDO AGUIRRE



M. EN C. FELIU DAVINO SAGOIS TRONCOSO



M. EN C. JORGE LUIS ALCANTARA GOMEZ PINEDA

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO NACIONAL

BIBLIOTECA DE INGENIERIA ELECTRICA
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro
antes del vencimiento de préstamo señalado
por el último sello.

07 FEB. 2002

20 Feb 02

DEVOLUCION

AUTOR VALDES RAMIREZ, R.O.

TITULO DESARROLLO DE UNA INTERFAZ
DE USUARIO PARA UN SISTEMA...

CLASIF. XM REGTRO. BI
91.1 12046

NOMBRE DEL LECTOR	FECHA PREST.	FECHA DEVOL.
Dr. Guillermo Morales	7-12-95	7/16/96
Rafael Escamilla	7/16/92	5/17/92

Unidad de procesamiento

Almacenamiento

Red

Dispositivos

Perif.

