

21
2Ej.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

**INTERFAZ HOMBRE-MAQUINA
PARA UN
SISTEMA DE CONTROL DISTRIBUIDO**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECANICO ELECTRICISTA**

P R E S E N T A

FRANCISCO JAVIER PEREZ CAMPOS

**TESIS CON
FALLA DE ORIGEN**



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

El objetivo del presente trabajo tiene la finalidad de describir, en forma general, un sistema de Control Distribuido; y en particular, un interfaz Hombre - Máquina a través del cual es posible interactuar con dicho sistema.

El sistema de control distribuido, como todos los de este tipo, cuenta con módulos autónomos que son capaces de realizar algunas acciones de control. Estos módulos controlan cierto número de variables, tanto analógicas como digitales y generalmente, se encuentran instalados cerca de los elementos de control que han de gobernar.

El sistema cuenta con un par de computadoras que realizan, entre otras cosas, la carga de información de todos y cada uno de los parámetros que intervienen en el proceso, la concentración de dicha información, el despliegue de ésta a través de diagramas mímicos, despliegue de tendencias, modificación de variables etc. Los comandos insertados a través del puesto de control son capaces de alterar cualquier parámetro del sistema independientemente del módulo en que se encuentre.

La interconexión entre computadoras y módulos se lleva a cabo mediante canal serie a través del cual, dichos módulos son configurados, es decir la computadora los carga con toda la información que requiere ese módulo para operar. También a través de este canal se realiza el monitoreo de variables y eventualmente, el cambio de parámetros.

Cada módulo está equipado con memorias RAM y ROM. En las primeras, se almacena principalmente la base de datos de dicho módulo. La memoria ROM contiene los programas de Adquisición, Secuenciación de Algoritmos de Control, Biblioteca de Algoritmos de Control, programas de Diagnóstico, Programas de Comunicación con Computador Central y Programas de Interfaz Hombre - Máquina de Campo.

Los programas de Adquisición consisten en la adquisición y muestreo de las variables de campo, tanto analógicas como digitales. Estos valores son validados, normalizados y acondicionados para que otros programas puedan hacer el tratamiento correspondiente.

Los programas de secuenciación de Algoritmos de Control tienen por objeto el ordenamiento y llamada a ejecución de los algoritmos de control para una aplicación determinada.

La biblioteca de Algoritmos de Control contiene 20 diferentes tipos de algoritmos de control que operan sobre un conjunto de variables, para modificar su comportamiento de acuerdo a un patrón preestablecido.

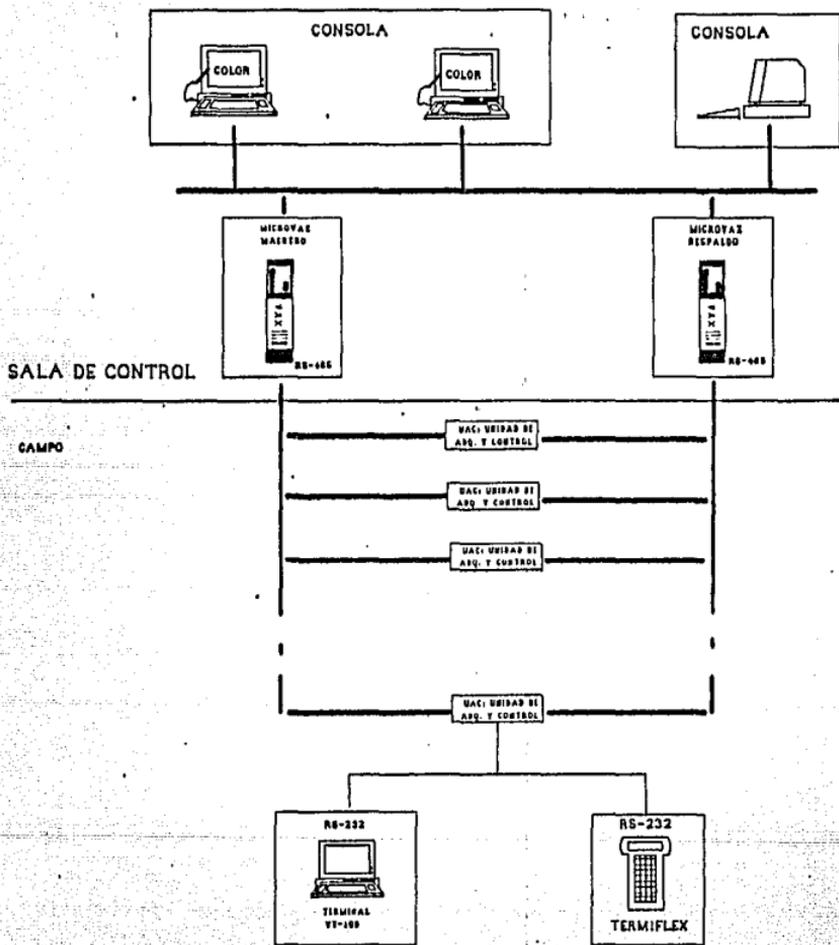
Los programas de Diagnóstico efectúan una verificación de la memoria, fuentes de alimentación, comunicación con la computadora central, y el estado funcional de algunas tarjetas.

Los programas de comunicación con computador central tienen como fin recibir la configuración para un módulo determinado, recibir órdenes de modificación de parámetros y enviar la información solicitada.

La Interfaz Hombre - Máquina de Campo tiene por objeto monitorear la base de datos local y proporcionar un medio para la modificación de parámetros. Esta comunicación Hombre - Máquina puede ser realizada por medio de una terminal tipo VT-100, una computadora Personal o una terminal portátil. Para esto, deberán conectarse del puerto serie, al canal de comunicación del módulo, empleando una interfaz tipo RS-232C.

La siguiente figura muestra un diagrama esquemático de la Arquitectura del Sistema de Control. Cabe decir, que esta arquitectura tiene un formato general ya que no depende del tipo de proceso que se desea controlar.

ARQUITECTURA DEL SISTEMA DE CONTROL



I N D I C E

1	Funciones de la Unidad de Control	1
	Funciones	1
	1.1 Adquisición de señales Analógicas	1
	1.2 Adquisición de señales Digitales	1
	1.3 Tratamiento de las señales analógicas	2
	1.4 Detección y reporte de señales Analógicas	2
	1.5 Bloques de Control	3
	1.6 Comunicación	4
	1.6.1 Introducción	4
	1.6.2 Conexión de la Unidades de Control	5
	1.6.3 Teoría de Operación	5
	1.7 Diagnóstico	6
	1.8 Modificación Remota	7
	1.9 Seguridad	7
2	Elementos Internos	8
	Elementos Internos	8
	2.1 Variables Analógicas	8
	2.1.1 Estructura General	8
	2.1.2 Estructura de las variables de Entrada	9
	2.1.3 Estructura de las variables Calculadas	10
	2.2 Variables Digitales	10
	2.2.1 Estructura General	11
	2.3 Ranuras de Control	11
	Introducción	11
	2.3.1 Definición de Ranuras	13
	2.3.1.1 Estructura General	13
	2.3.2 Secuenciamiento de Ranuras	13
	Introducción	13
	2.3.2.1 TABLAS	14
	2.4 Salidas Analógicas y Digitales	18
	2.4.1 Estructura	18
	2.5 Paro de Emergencia	19
	2.5.1 Introducción	19
	2.6 Estados Operativos	19
	Introducción	19
	2.6.1 UAC	19
	2.6.2 Variables	20
	2.6.3 Ranuras	20
3	Algoritmos de Control	21
	3.1 PID	21
	3.2 PID RAZON	23
	3.3 PID BANDA	24
	3.4 PID BATCH	24
	3.5 ESTACION MANUAL / AUTOMATICA ANALOGICA	25
	3.6 ESTACION MANUAL / AUTOMATICA DIGITAL	26
	3.8 CONTROL ON / OFF	27
	3.9 CONTROL HIDRAULICO	28
	3.10 GENERADOR DE FUNCIONES	29
	3.11 COMPENSADOR ADELANTO / ATRASO	30
	3.12 FUNCION DE TIEMPO	31
	3.14 FLUJO DE MASA (RAIZ CUADRADA)	33
	3.15 SUMADOR	33

3.16	MULTIPLICADOR	34
3.17	DIVISOR	34
3.18	CONTADOR DE EVENTOS	35
3.19	FUNCION LOGICA	35
3.20	SELECTOR DIRECTO	36
3.21	SELECTOR ALTO / BAJO	36
3.22	ACTUADOR	36
4	Interfaz Hombre-Maquina de Campo	38
	Introducción	38
	Funciones	38
	Características	38
	Pantalla. Areas de despliegue	40
4.1	Comandos generales	42
4.2	Captura de Comandos	46
4.2.1	Descripción	46
4.2.2	Tabla de Configuración de Pantalla. Descripción	46
4.2.3	Identificadores de tablas	48
4.3	Despliegue de Tablas	48
4.3.1	Descripción general	48
4.3.2	Descripción de la Tablas	49
4.3.3	Descripción de la Tabla de Ranuras	56
5	Arquitectura del Interfaz Hombre-Máquina de Campo	63
5.1	Manejador de Consola	63
5.2	Módulos Constitutivos - Recepción	63
5.3	Módulos Constitutivos - Transmisión	66

APENDICE I

Programas fuente de algunas rutinas de la Sección de Recepción (Captura).

APENDICE II

Programas fuente de algunas rutinas de la Sección de Transmisión (Despliegue).

APENDICE III

Ejemplos de despliegue de tablas.

C A P I T U L O I

Funciones de la Unidad de Control

Funciones

Las funciones principales que realiza la Unidad de Adquisición y Control (UAC) son:

- Adquisición de Señales Analógicas.
- Adquisición de Señales Digitales.
- Validación, Normalización y Acondicionamiento de Señales Analógicas.
- Detección y Reporte de cambios y niveles de alarma en Señales Analógicas.
- Detección y Reporte de cambio de nivel en Señales Digitales.
- Ejecución autónoma de diversos y sofisticados bloques de Control.
- Capacidad de Comunicación a Computadoras de Proceso y a terminales locales.
- Auto-Diagnóstico local.
- Configuración remota de todas las funciones de Control y Adquisición.
- Modificación remota de puntos de consigna y otras condiciones a los bloques de control y a las variables de adquisición.
- Generación de Señales Analógicas y Digitales, para acciones de control directas.

1.1 Adquisición de señales Analógicas

La adquisición consiste en el muestreo y conversión de las variables analógicas de entrada.

Las variables analógicas son convertidas con una resolución de 12 bits.

A cada entrada analógica se le pueden definir uno de tres periodos de muestreo para su adquisición:

- 500 milisegundos.
- 1 segundo. y
- 2 segundos.

Además del muestreo se puede definir un filtrado digital que consiste en considerar el valor promedio de las últimas cuatro muestras como el valor de la entrada.

1.2 Adquisición de señales Digitales

En el caso de las señales digitales la adquisición consiste en muestrearlas periódicamente. Este periodo de muestreo es único y de 500 milisegundos.

Cada entrada digital consiste de una palabra de 16 bits.

1.3 Tratamiento de las señales analógicas

Una vez que las señales fueron adquiridas y convertidas es preciso verificar que los valores registrados se encuentren dentro del rango especificado para el instrumento o transductor que se este utilizando, esto por ejemplo para el caso de instrumentos que producen una señal de 4 a 20 miliamperes, implicará verificar que el valor adquirido sea mayor ó igual a 4 miliamperes y menor ó igual a 20 miliamperes. Si este rango es violado, se considera que el instrumento está en falla y la entrada analógica es puesta fuera de servicio.

Si el rango fue adecuado se procede a convertir el valor analógico a un valor normalizado de 0.0 a 1.0, este valor normalizado permite que la señal pueda ser usada por los bloques de control y que sus niveles de alarma puedan ser comparados.

Una vez que las señales fueron normalizadas se procesan a fin de compensar cualquier no-linealidad que pueda tener el transductor, para esto la UAC cuenta con varias funciones de linealización que el usuario puede especificar para cada entrada analógica.

1.4 Detección y reporte de señales Analógicas

Una vez que las señales fueron adquiridas, normalizadas y linealizadas se verifican dos condiciones, sus niveles de alarma y si hubo cambio.

Se consideran seis alarmas para cada variable analógica de Entradas:

- Nivel Critico Alto.
- Nivel Precritico Alto.
- Nivel Precritico Bajo.
- Nivel Critico Bajo.
- Desviación Ascendente.
- Desviación Descendente.

Cuando la variable sobrepase el nivel Precritico se genera una Alarma Precritica, y cuando sobrepase el nivel Precritico y Critico se genera una Alarma Precritica y una Alarma Critica.

Las alarmas de desviación se especifican dando el porcentaje de cambio máximo ascendente o descendente por segundo.

Cada condición de alarma tiene un indicador independiente que permite diferenciarlas en el caso que se presenten simultáneamente.

Cada alarma de nivel tiene asociada una banda muerta para evitar que la condición de alarma cambie, cuando la variable oscile alrededor del punto de alarma.

Además se puede definir un valor mínimo de cambio, llamado sensibilidad que será el incremento ó decremento mínimo que debe sufrir la variable de entrada para que se considere que se debe reportar a la computadora de proceso su nuevo valor.

1.5 Bloques de Control

La función de Control consiste en regular el comportamiento de las variables analógicas que se desean controlar en el proceso (temperaturas, presiones, flujos, etc.).

La realización de esta función, se lleva a cabo a través de un conjunto de acciones de control simples, ejecutadas en la secuencia deseada. A ese conjunto de acciones se le conoce como " bloque de control " y comprende la actualización de entradas, salidas y la aplicación de diversos algoritmos de control. Se dispone de una biblioteca de algoritmos de control que contiene una serie de rutinas capaces de realizar todas las acciones de control que se requieren para esta aplicación.

La interconexión de cada bloque, los parámetros de los algoritmos, las entradas, las salidas y demás restricciones para cada bloque se especifican durante el proceso de configuración.

Dentro de la unidad se encuentra una biblioteca con el código para cada uno de los algoritmos de control que puede implementar la unidad, esta biblioteca se puede ampliar de acuerdo a las necesidades específicas de cada unidad de control.

Las funciones que implementan los algoritmos se listan a continuación :

- Controles PID:
 - Normal.
 - Razón.
 - Banda.
 - Batch.
- Estaciones manuales:
 - Analógica.
 - Digital.
 - Control ON / OFF.
 - Control Hidráulico.
- Generador de funciones.
- Función de Adelanto - Atraso.
- Función de tiempo.
- Operadores algebraicos:
 - Extractor de raíz cuadrada.

Funciones de Adquisición y Control

- Sumador.
- Multiplicador
- Divisor.
- Selectores de señal:
 - Selección directa.
 - Más Alto / Más Bajo.
- Puertas lógicas:
 - And.
 - Or.
 - Nand.
 - Nor.
 - Or Exclusivo.
- Contador de eventos.

1.6 Comunicación

1.6.1 Introducción

La UAC tiene la capacidad de conectarse a una computadora central por medio de una red de comunicación, lo que permite que opere dentro de un sistema de control distribuido.

Por medio de esta red la computadora central transfiere a la UAC la información que requiere esta para generar la base de datos local durante la configuración, así mismo puede monitorear el estado de las variables de proceso (señales de entrada) como de las variables controladas (señales de salida); así como informarle al operador de las condiciones de alarma que ocurran y definir los puntos de consigna de los bloques de control.

La red de comunicación está sujeta a un protocolo de comunicación que permite el intercambio de información entre los distintos elementos conectados a ella, define prioridades y detecta errores en la comunicación.

La topología soportada por la red es una configuración maestro-esclavos en multidrop. Cada unidad puede estar conectada a dos canales de comunicación, lo que permite una configuración redundante.

En la siguiente figura se muestra un diagrama de esta configuración.

1.6.2 Conexión de la Unidades de Control

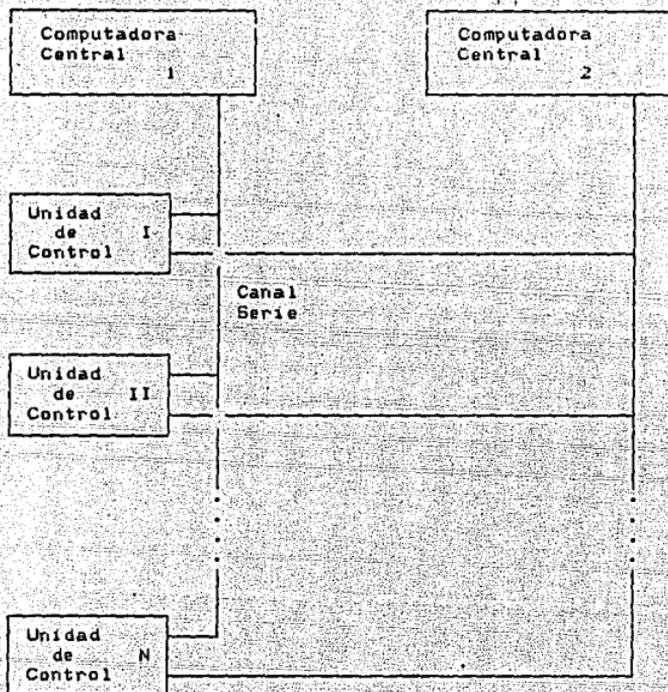


Fig.- Configuración de conexión de Unidades de Control

1.6.3 Teoría de Operación

La configuración multidrop funciona de la siguiente manera:

A cada canal serie de la computadora central se conectan determinado número de unidades de control. Cada unidad de control responderá a un código ó dirección específico. La computadora central coloca un mensaje en el canal serie, donde el primer dato del mensaje contiene la dirección de la unidad de

control. Todas las unidades de control conectadas al canal serie reciben el primer dato del mensaje y verifican que la dirección del mensaje corresponda a la dirección que tiene asignada, si la dirección corresponde, entonces se recibe el resto del mensaje, si no se ignora y el módulo de comunicación de la unidad continúa esperando otro mensaje.

El mensaje de comunicación tiene un formato general con tres secciones básicas:

- Encabezado
- Cuerpo
- Código de Redundancia

En el encabezado como primer dato está el código de dirección y en seguida la información de número de mensaje, tipo, número de datos, etc. El cuerpo contiene la información que se quiere transmitir, y finalmente el código de redundancia contiene información para poder determinar si el mensaje se recibió correctamente ó si se presentó algún error en la comunicación.

La computadora central envía dos tipos básicos de mensajes, los mensajes de comando y los de solicitud.

En los primeros envía un bloque de información y espera de la unidad de control una confirmación de su recepción.

En los de solicitud envía un bloque de solicitud de información y la unidad contesta con el bloque de datos solicitado ó con una condición de error.

Cuando el procesador de comunicación recibe un mensaje de la computadora central, verifica su integridad y de acuerdo al tipo le informa al procesador maestro que hay información disponible para él ó que se requiere determinada información.

1.7 Diagnóstico

Las funciones de diagnóstico se pueden clasificar en dos tipos, las de diagnóstico puramente interno de la unidad de control y las funciones que permiten diagnosticar las fallas que ocurren con relación a las señales de entrada y salida. En las primeras se tienen la verificación de la memoria, la comprobación de comunicación con la computadora central y la verificación de las fuentes de alimentación. En las segundas se consideran la verificación del estado funcional de las tarjetas del controlador, la verificación de los conectores a campo y los rangos de operación de los transductores.

Estas funciones son realizadas en forma periódica por la UAC y los errores detectados son reportados a la computadora central.

Como se mencionó anteriormente existen casos en los que la unidad procede a poner fuera de servicio alguna entrada analógica o digital. Las condiciones más frecuentes son la falta de conector a campo o la detección de un valor fuera de rango. Así mismo la UAC puede proceder a ponerse en fuera de servicio en forma total si detecta una falla que comprometa la operación del proceso, en el caso de que exista un mecanismo de respaldo este mecanismo asume el control al quedar la unidad fuera de servicio.

1.8 Modificación Remota

La unidad recibe a través del canal de comunicación tanto solicitudes de reporte de cambios, como ordenes de modificación de parámetros. Esto permite la interacción de la unidad en un esquema de control distribuido, donde la computadora central de proceso se encarga de mantener la supervisión sobre el proceso y de ajustar los puntos de consigna, niveles de alarma, etc. mientras que las unidades se responsabilizan del control directo de la planta.

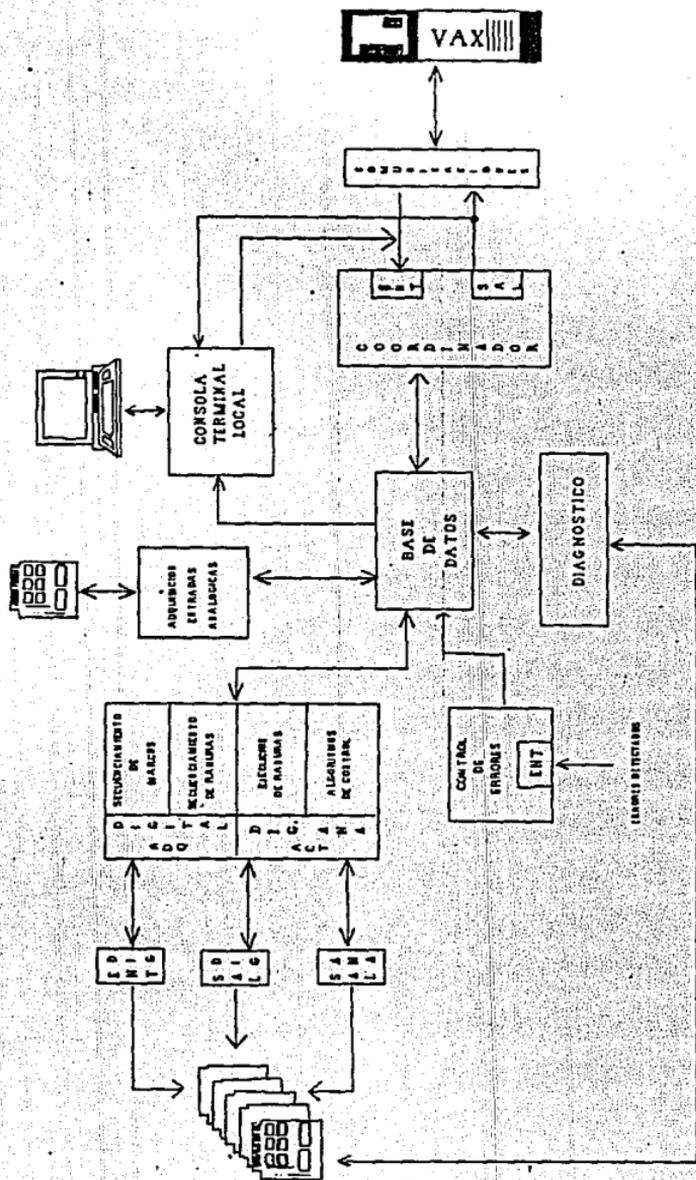
1.9 Seguridad

Además de los módulos mencionados, la unidad de control cuenta con una tarjeta de monitoreo llamada tarjeta centinela que verifica continuamente las fuentes de alimentación y las señales de Watch-Dog del proceso maestro. Si alguna de estas señales falla o si se detecta la señal de falla interna generada por el procesador maestro, se envía una señal a la planta para activar los sistema de respaldo. La tarjeta centinela le presentará al operador un código que indica el tipo de falla ocurrido en la unidad.

C A P I T U L O I I

Elementos Internos

MODULOS CONSTITUTIVOS DE LAS UNIDADES DE ADQUISICION Y CONTROL



Elementos Internos

En la estructura interna de la UAC se consideran los siguientes elementos:

- Variables Digitales
- Variables Analógicas
- Ranuras de Control
- Salidas Analógicas
- Salidas Digitales
- Tabla de Paro de Emergencia

2.1 Variables Analógicas

Dentro de las variables analógicas se consideran los siguientes tipos:

Las variables de Entrada o de proceso, que son variables analógicas adquiridas del campo, convertidas y linealizadas tal como se mencionó en la sección 2.

Las variables Calculadas, que son aquellas cuyo valor es producido por las Ranuras de Control.

2.1.1 Estructura General

Todas las variables analógicas de la UAC tienen la siguiente estructura:

STATUS	Estado Operativo; Dentro o Fuera de servicio.
TIPO	Tipo de variable; Entrada, Asignada o Interna.
VALOR	Valor normalizado de la variable, rango de 0.0000 a 0.9999
IND_ALM	Indicador de Alarmas, 16 bits.
L_C_SUP	Valor normalizado a partir del cual se considera una condición de Alarma crítica superior.
L_PC_SUP	Valor normalizado a partir del cual se considera una condición de Alarma precrítica superior.
L_C_INF	Valor normalizado hasta el cual se considera una condición de Alarma crítica inferior.
L_PC_INF	Valor normalizado hasta el cual se considera una condición de Alarma precrítica inferior.

L_D_ASC	Aumento normalizado mínimo del valor de la variable en un segundo para considerar una alarma de desviación ascendente.
L_D_DSC	Disminución normalizada mínima del valor de la variable en un segundo para considerar una alarma de desviación descendente.

Indicadores de Alarma:

Bit 15	Critica superior
Bit 14	Precritica superior
Bit 13	Precritica inferior
Bit 12	Critica Inferior
Bit 11	Desviación Ascendente
Bit 10	Desviación Descendente
Bit 09	Indicador de Alarma
Bit 08	" "
Bit 07	" "
Bit 06	" "
Bit 05	" "
Bit 04	" "
Bit 03	" "
Bit 02	" "
Bit 01	" "
Bit 00	" "

2.1.2 Estructura Especifica de las variables de Entrada

En el caso de las variables de entrada de proceso se tienen además los siguientes campos:

UBICACION	Definición de la terminal física de la Unidad de Adquisición y Control de donde se toma el valor de la entrada.
R_B	Rango superior de normalización, valor del convertidor que corresponderá al valor normalizado 1.0. El valor del convertidor va de 0.000 a 0.999 que corresponde a un cambio digital de 0x0000 a 0x0FFF respectivamente
R_l	Rango inferior de normalización, valor del convertidor que corresponderá al valor normalizado 0.0.
MULTP	Factor de ganancia definido para el amplificador de entrada del convertidor A/D.

Elementos Internos

SENSB Definición de porcentaje de cambio en la variable de entrada mínimo para considerar que se deba reportar el nuevo valor a la computadora de proceso. Los valores válidos son los siguientes:

Valor definido % de cambio

1	25	%
2	13	%
3	07	%
4	03.5	%
5	01.5	%
6	00.75	%
7	00.36	%
8	00.18	%

FILTRADO Indica si el valor de la variable se promedia con el valor leído en las últimas tres muestras o si se toma directo.

ALG_ACND Código que define el tipo de algoritmo de linealización o acondicionamiento que se utilizará para la entrada.

PER_MBT Período de muestreo, define la frecuencia con la que se adquiere la variable:

0.5	Activación cada 500 milisegundos.
1	Activación cada 1 segundo.
2	Activación cada 2 segundos.

2.1.3 Estructura Específica de las variables Calculadas

No existen campos específicos para las Variables Calculadas.

2.2 Variables Digitales

Dentro de las variables digitales se consideran los siguientes tipos:

Las variables de Entrada o de proceso, que son variables digitales adquiridas del campo.

Las variables Calculadas que son las palabras de Alarma y Modo de las Ranuras.

2.2.1 Estructura General

Todas las variables digitales de la UAC tienen la siguiente estructura:

STATUS	Estado Operativo; Dentro o Fuera de servicio.
TIPO	Tipo de variable; Entrada, Asignada o Interna.
VALOR	Valor, palabra de 16 bits.
IND_CMB	Indicador de Cambio, 16 bits.

Indicadores de Cambio:

La palabra 'Indicador de Cambio' de 16 bits de longitud representa el estado lógico correspondiente a bit renenciado. De esta forma, el bit 15 de la palabra Indicador de Cambio representa el estado lógico de dicho bit.

2.3 Ranuras de Control

Introducción

Control Ejecutivo.

Control Ejecutivo es el nombre con el que se conoce al conjunto de tareas y rutinas que se encargan de supervisar la obtención de entradas, secuenciar y ejecutar las acciones de control y actualizar las salidas.

Antes de describir las funciones específicas del Control Ejecutivo, conviene conocer las consideraciones tomadas y la estructura sobre la que opera.

Antecedentes.

La selección de la frecuencia de muestreo de las variables de proceso, en una industria a la que se le aplica un sistema de control, requiere de un análisis detallado de la rapidez con que las variables a controlar cambian o pueden ser cambiadas. Para esta aplicación, se observó que frecuencias de muestreo comprendidas entre 0.5 y 2 segundos, permitían proporcionar una alta capacidad de ejecución sin incrementar en demasía el costo del sistema.

Considerando las observaciones anteriores, se estableció que las acciones de control serían realizadas con periodos de 0.5, 1, 1.5 y 2 segundos, dependiendo de las características del proceso.

A partir de esta división, se subdividió cada marco para poder realizar el resto de las funciones de la U.A.C.

- Adquisición de entradas analógicas y digitales.
- Comunicaciones.
- Atención a terminal local.
- Diagnóstico.
- Configuración.
- Salidas analógicas y digitales.

" ALGORITMO " es una rutina dentro de la biblioteca de control que realiza una función de control específica. El sistema cuenta, como ya se ha mencionado, con una serie de algoritmos cuyos tiempos de ejecución pueden variar de uno a otro.

A partir de este hecho, se consideró como unidad de duración el tiempo que consumía un algoritmo cuya ejecución fuese la más breve. Posteriormente, se dividió informalmente el tiempo de ejecución de un marco en dos acciones de control por un lado y el resto de las funciones por otro. Tomando como referencia la mitad de un marco (250 milisegundos) y considerando la duración del resto de los algoritmos, se determinó que el marco tendría capacidad para ejecutar un número de algoritmos de control cuyo tiempo de ejecución no fuera mayor de 240 unidades de duración.

" RANURA " es la definición de un elemento de control que implica al algoritmo, independientemente de su duración, y los valores de todos sus parámetros incluyendo la identidad de sus entradas y salidas.

De esta forma, la U.A.C. tiene capacidad para definir hasta 960 ranuras de control divididas en 4 marcos.

Durante el proceso de configuración, entre otras cosas, se llena la Tabla de Ranuras con las direcciones y valores asociados a cada algoritmo. Después, se procede a cargar la secuencia de las ranuras a ejecutarse en la tabla de Secuenciamiento de Ranuras. Esto último se hace con base a la frecuencia de activación establecida para cada algoritmo.

La tabla de ranuras es una estructura que tiene capacidad para almacenar la información de hasta 960 algoritmos. Esta información incluye: tipo de algoritmo, frecuencia de activación, modo de operación, constantes, puntos de ajuste, direcciones, etc.

La tabla de secuenciamiento de ranuras, solamente contiene la identificación de la ranura a ejecutarse dentro de la secuencia.

Composición el Control Ejecutivo.

El Control Ejecutivo está constituido por tres tareas:

- Tarea de Secuenciamiento de Marcos.

Elementos Internos

- Tarea de Secuenciamiento de Ranuras.
- Tarea de Ejecución de Ranuras.

La tarea de Secuenciamiento de Marcos proporciona la sincronización de la activación de los marcos. Es decir, genera un pulso cada 500 mseg.

La tarea de Secuenciamiento de ranuras obtiene un marco y verifica que la ejecución de las ranuras se haga con base a la secuencia establecida. También verifica, al momento de cargar, que la duración de las ranuras a ejecutarse no exceda 240 unidades de duración, en cuyo caso, cargará aquellas que le sean posibles y generará un mensaje de error crítico.

Tarea de Ejecución de ranuras. Cuando esta tarea es iniciada, obtiene de la tabla de secuenciación, el número de ranura a ejecutarse. Con este dato se direcciona a la tabla de ranuras en donde obtiene todo el entorno del algoritmo indicado. Estos datos son enviados como parámetros al algoritmo contenido en la biblioteca de control, para su procesamiento. Finalmente, actualiza la base de datos.

Con el fin de garantizar la continuidad de las funciones de esta tarea, se ha dividido en tres etapas de duración limitada. La primera es la de adquisición de entradas, la cual comienza inmediatamente después de iniciado el marco. La etapa siguiente realiza la ejecución de los algoritmos y por último, la tercera etapa se encarga de actualizar las salidas.

2.3.1 Definición de Ranuras

2.3.1.1 Estructura General

Todos las Ranuras de Control en la UAC tienen la siguiente estructura común:

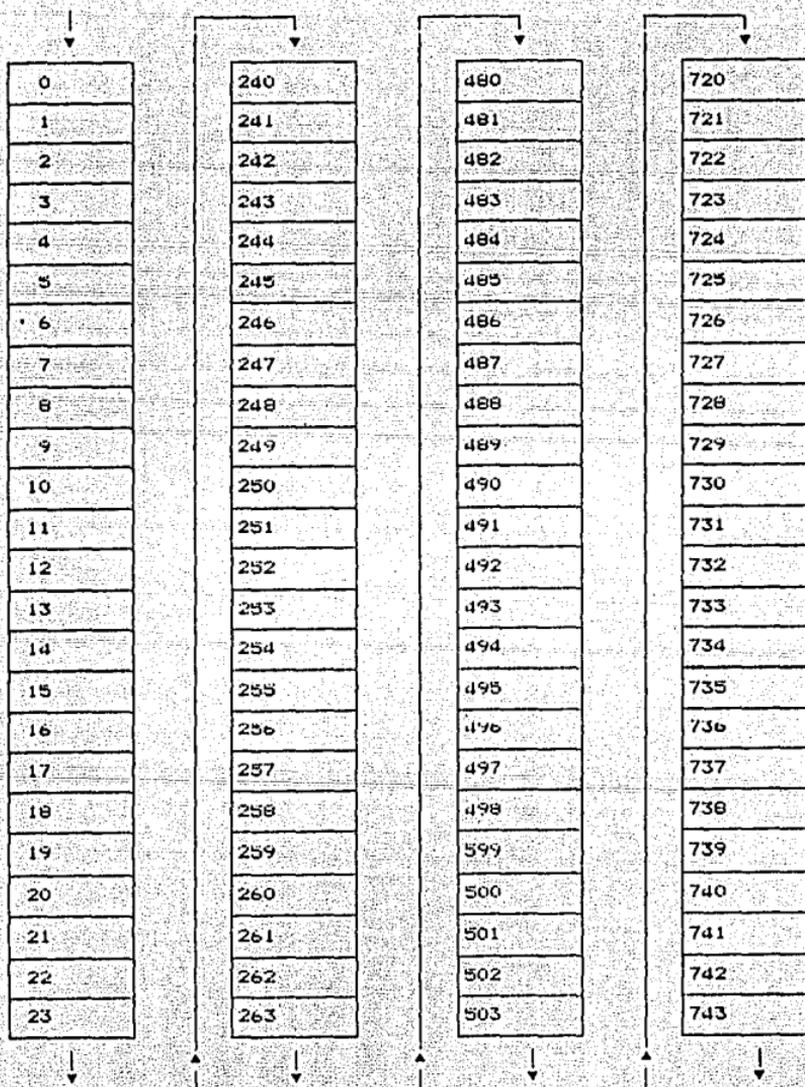
STATUS	Estado Operativo; Dentro o Fuera de servicio.
ALG	Código del algoritmo que se ejecutará en la ranura.
HIST. [4]	Valores de Historia (cuatro).

2.3.2 Secuenciamiento de Ranuras

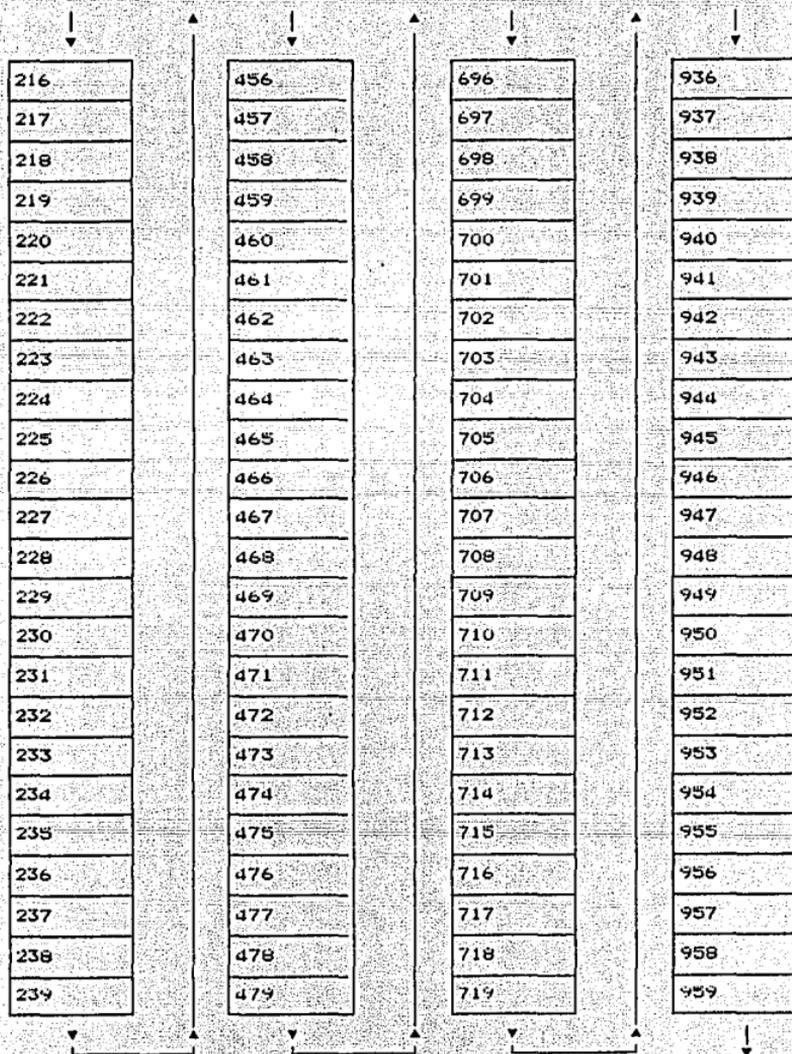
Introducción

La secuencia de ejecución de las ranuras de control se define en la siguiente forma:

Elementos Internos



Elementos Internos



Elementos Internos

La posición del número de ranura en la forma anterior define la secuencia de activación de las ranuras, en otras palabras las ranuras se activan en el orden fijado por el número de MARCO.

Por otro lado, de acuerdo a la frecuencia de activación de la ranura, ésta puede ocupar más de un marco:

4 Marcos Activación cada 500 milisegundos.
 2 Marcos Activación cada 1 segundo.
 1 Marco Activación cada 2 segundos.

Además los marcos para activación c/500 mseg. deben ser consecutivos:

N, N+240, N+480 y N+720

y para activación c/1 seg. deben ser consecutivos y alternos:

N y N+480

Por ejemplo si deseamos definir la secuencia de activación de los siguientes bloques:

El primero formado por las ranuras 1 a 5 con una frecuencia de activación de 1 seg., de modo que se activen en el siguiente orden:

4,2,3,5,1

El segundo formado por las ranuras 6 a 9 con una frecuencia de activación de 500 mseg., de modo que se activen en el siguiente orden:

9,7,6,8

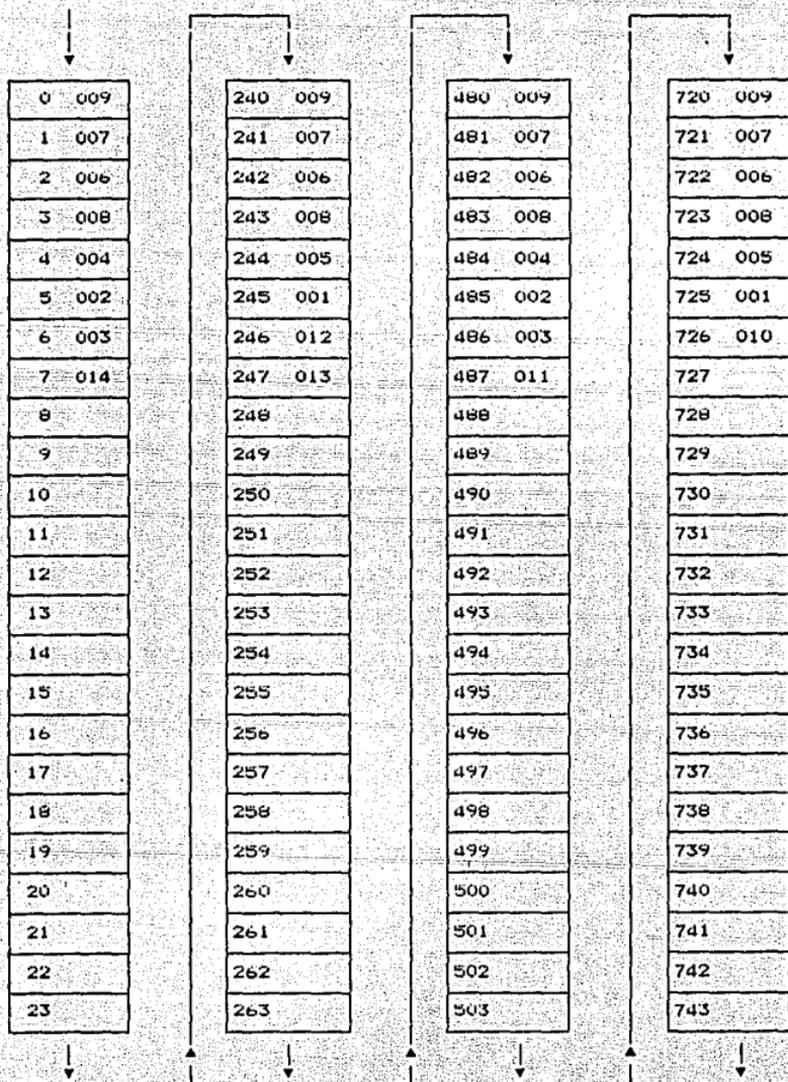
y El último formado por las ranuras 10 a 14 con una frecuencia de activación de 2 seg., de modo que se activen en el siguiente orden:

14,12,13,11,10

La forma de llenar la forma de secuencia será la siguiente:

- a) Primero se agrupan los bloques con activación de 500 mseg., luego los de 1 seg. y finalmente los de 2 seg.
- b) Segundo se cargan en la forma los identificadores de ranuras de cada bloque conforme a la secuencia deseada.

Tendremos así una forma con la siguiente información:



Elementos Internos

El número de marco definido por el procedimiento anterior deberá ser cargado en las formas que definen las ranuras. En el caso de ranuras con períodos de muestreo de 500 mseg. y 1 seg. se deberá indicar el primer marco que ocupe la ranura. Así para el caso anterior tendremos que la relación Ranura-Marco queda como sigue:

Ranura	Marco
01	245
02	005
03	006
04	004
05	244
06	002
07	001
08	003
09	000
10	726
11	487
12	246
13	247
14	007

2.4 Salidas Analógicas y Digitales

2.4.1 Estructura

Con el objeto de poder controlar el proceso es necesario enviar señales de control a campo, para lo cual se tiene una estructura que define las variables que se enviarán al exterior y la terminal física de la UAC por donde se enviarán, los campos de dicha estructura tanto para variables analógicas como para variables digitales de salida son:

STATUS	Estado Operativo; Dentro o Fuera de servicio.
UBICACION	Definición de la terminal física de la Unidad de Adquisición y Control por donde se enviará el valor de la salida.

Elementos Internos

2.5 Paro de Emergencia

2.5.1 Introducción

La Unidad tiene la capacidad de asociar a las ranuras que producen salida al exterior una condición externa, ya sea señal digital del proceso u orden de la Computadora de Proceso, a fin de que cuando esta condición se presente se obligue la salida del proceso a una condición específica.

Las condiciones específicas previstas son:

El último valor de la salida

Un valor específico predeterminado.

2.6 Estados Operativos

Introducción

La UAC y cada una de sus variables contienen el campo de STATUS el cual permite modificar el estado operativo de la UAC o de variables específicas. A continuación se detallan los distintos modos operativos y sus implicaciones:

2.6.1 UAC

Los estados operativos para la Unidad son:

En Servicio:

La unidad está operando con una configuración válida.

Bin Configurari:

La unidad acaba de ser encendida y no tiene una configuración válida.

En Configuración:

La unidad está en el proceso de recepción de una configuración.

Fuera de Servicio:

Cuando la unidad fue configurada pero no pudo armar las tablas internas de adquisición y secuenciación. O bien detectó algún error crítico en su operación.

2.6.2 Variables

Los Estados Operativos definidos para las variables son:

Libre:

Los parámetros de esta variable no han sido definidos.

En Servicio:

La variable fue definida correctamente y se encuentra en operación.

Fuera de Servicio:

La variable fue definida correctamente pero no se encuentra en operación. Esto significa que no se adquiere del exterior o no se actualiza. La variable mantiene su último valor.

Las variables de salida presentan un valor indeterminado durante el lapso que transcurre entre el encendido y la variable es configurada y puesta en servicio.

2.6.3 Ranuras

Los Estados Operativos definidos para las ranuras son:

Libre:

Los parámetros de esta ranura no han sido definidos.

En Servicio:

La ranura fue definida correctamente y se encuentra en operación.

Fuera de Servicio:

La ranura fue definida correctamente pero no se encuentra en operación. Esto significa que no se ejecuta el algoritmo y que sus salidas no se alteran.

CAPITULO III

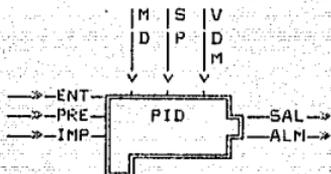
Algoritmos de Control

3.1 PID

El controlador Proporcional - Integral - Derivativo (PID) es un bloque que tiene la capacidad de controlar una entrada a un punto de referencia dado por medio de una salida de control.

Para efectuar lo anterior, el controlador tiene en su interior un algoritmo digital el cual toma en cuenta el error existente entre la variable de entrada y el punto de referencia así como los valores anteriores de la entrada para poder generar la señal de salida.

La dinámica de la acción de control se fija a través de las constantes de control k_p , k_i y k_d las cuales sirven para sintonizar la acción de control. Los valores de estas constantes dependen de la dinámica del proceso.



ENT Es la variable de proceso a controlar, en la mayoría de los casos se trata de una señal leída de campo, pero también puede ser una variable de salida procedente de otro algoritmo.

PRE Polariza la salida de control, se trata de una señal variable, que su uso generalmente es el anticipar el comportamiento de la salida de control; esto quiere decir que por medio del conocimiento del valor de una variable (generalmente exterior al lazo) es posible cambiar el valor de la salida para que la acción de control final sea más rápida.

IMP Es una entrada digital que es usada cuando el controlador es parte de una cascada. Cambia al controlador a modo manual imposición. Este modo consiste en retroajustar todos los valores del controlador al valor deseado en modo manual, en el cual, está depositado el valor que está tomando el otro elemento de la cascada. El controlador quedará bloqueado en el modo descrito mientras la entrada este activada.

SAL Es la salida de control generada por el bloque

ALM La palabra de Alarma y Modo es generada por el algoritmo y en esta se encuentra contenida toda la información de alarmas y el modo en que se encuentra funcionando el algoritmo.

MD Modo: El controlador puede funcionar en dos modos de operación: Automático y Manual. En el modo automático la salida de control proviene del cálculo del algoritmo de control. En el modo manual la salida está directamente determinada por el operador.

BP Es el punto de referencia al cual será controlada la entrada.

VDM Es el valor que se desea en la salida cuando se está trabajando en modo manual.

El algoritmo PID implementa las funciones básicas de un controlador PID cuya salida es función de la desviación entre la variable de proceso de entrada y el valor de referencia:

$$\text{SALIDA_DE_CONTROL} = P + I + D + FF + PL$$

donde:

P = Es la acción de la parte proporcional.
($KP * \text{DESVIACION}$)

I = Es la acción de la parte integral.
($KI * \text{INTEGRAL_DE_DESVIACION}$)

D = Es la acción de la parte derivativa.
($KD * \text{RAZON_DE_CAMBIO_DE_LA_DESVIACION}$)

FF = Es el término de anticipación de la acción de control.

PL = Salida calculada a error cero. Se usa para compensar posibles errores en estado estable que se presenten en el comportamiento de un lazo al tratar de estabilizarlo a un punto de referencia.

KP también llamada Ganancia de algoritmo de control, es la constante de proporción o proporcional que indica el porcentaje de banda proporcional del sistema que intervendrá en el cálculo del controlador.

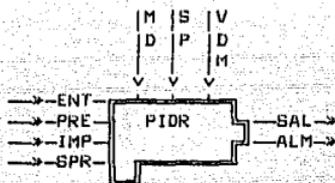
KI es la constante de integración o integral que indica el número de veces en que se duplicará el término proporcional en una unidad de tiempo, generalmente especificada en repeticiones por minuto.

KD es la constante derivativa. Indica el porcentaje de la razón de cambio de la señal de entrada que es tomado en cuenta para el cálculo de la salida de control.

El algoritmo implementado es digital, implica que el cálculo de la señal de control depende de las muestras de la señal de proceso, las cuales entran al cálculo por diferencias que toma en cuenta hasta tres historias del proceso para generar la señal de control.

3.2 PID RAZON

El controlador PID Razón es un bloque que tiene la capacidad de controlar una entrada a un punto de referencia dado por medio de una salida de control, pero además puede recibir como su punto de referencia una variable o señal para ser conectado en cascada. El controlador tiene en su interior un algoritmo digital, el cual toma en cuenta el error existente entre la variable de entrada y el punto de referencia así como los valores anteriores de la entrada para poder generar la señal de salida.



BPR. Es el punto de referencia remoto. Variable interna o externa que servirá como punto de referencia cuando el Pid Razón se encuentre conectado en cascada.

El algoritmo PID razón implementa las funciones básicas de un controlador PID cuya salida es función de la desviación entre la variable de proceso de entrada y el valor de referencia.

Su configuración normal es estar conectado en cascada con otro controlador, tomando el controlador razón la posición interna de la cascada, es decir, estará manejando directamente el elemento final. En este caso el punto de referencia de trabajo está formado por el punto de referencia remoto asociado a dos constantes:

RAZON Es la constante de escalamiento del punto de referencia remoto la cual sirve para posibles compensaciones en configuración cascada.

BRAZON Es la constante de desplazamiento del punto de referencia remoto la cual sirve para posibles compensaciones en configuración cascada.

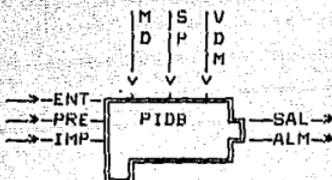
La ecuación de salida es la misma del controlador PID:

$$\text{SALIDA_DE_CONTROL} = P + I + D + \text{FF} + \text{PL}$$

Las constantes de control son las mismas descritas en el controlador PID, teniendo las mismas características.

3.3 PID BANDA

El controlador Pid Banda es un bloque que tiene la capacidad de controlar una entrada a un punto de referencia dado por medio de una salida de control, pero además se le puede especificar una segunda banda en la cual registrará una segunda constante proporcional.



El algoritmo PID Banda implementa las funciones básicas de un controlador PID cuya salida es función de la desviación entre la variable de proceso de entrada y el valor de referencia. Pero presenta una segunda constante proporcional (q_b) la cual se activa a partir de que el algoritmo entra en un determinado rango de error (pab). La ecuación de salida es la misma del controlador PID:

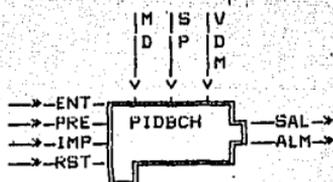
$$\text{SALIDA_DE_CONTROL} = P + I + \text{FF} + \text{PL}$$

con la diferencia que si el error es menor al rango pab se toma en cuenta la constante q_b , y en el caso contrario se toma en cuenta la constante proporcional.

3.4 PID BATCH

El controlador Pid Batch es un bloque que tiene la capacidad de controlar una entrada a un punto de referencia dado por medio de una salida de control.

La característica especial de este controlador es que pueden ser restablecidos todos sus valores de operación, por medio de las entradas de restablecimiento. Es decir el controlador puede ser inicializado en línea para poder comenzar otra acción de control.



RST Es la entrada digital de restablecimiento que se usa para restablecer los registros del controlador.

El algoritmo PID Batch implementa las funciones básicas de un controlador PID cuya salida es función de la desviación entre la variable de proceso de entrada y el valor de referencia:

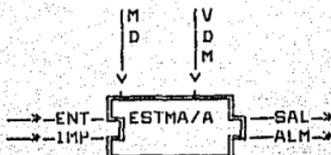
$$\text{Salida de Control} = P + I + D + \text{PRE} + \text{PL}$$

donde:

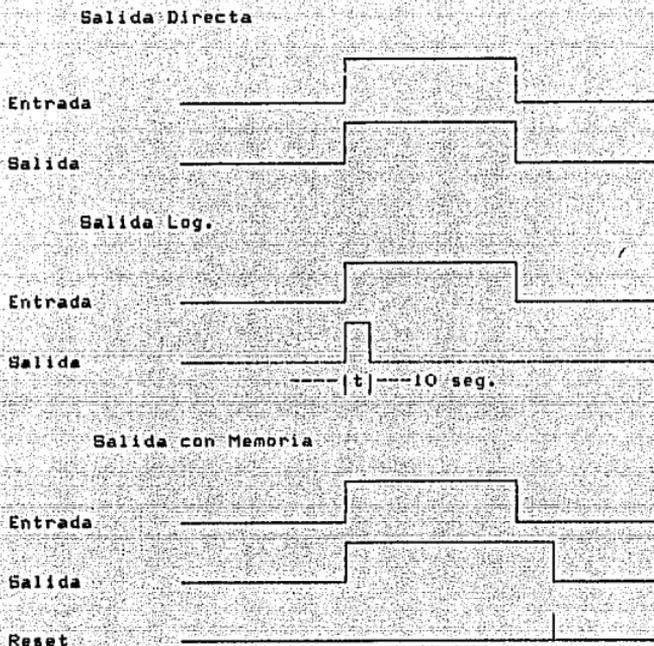
- P - Es la acción de la parte proporcional.
(K_P * Desviación)
- I - Es la acción de la parte integral.
(K_I * Integral de la Desviación)
- D - Es la acción de la parte derivativa.
(K_D * Razón de cambio de la Desviación)
- PRE - Es el término de prealimentación
- PL - Salida calculada a error cero .

3.5 ESTACION MANUAL / AUTOMATICA ANALOGICA

Este algoritmo proporciona al sistema una estación por medio de la cual el operador puede optar por la señal que se presenta a la entrada o efectuar una acción manual, en la mayoría de los casos su salida se emplea para proporcionar directamente señal al elemento final.



La ecuación de salida es la siguiente:



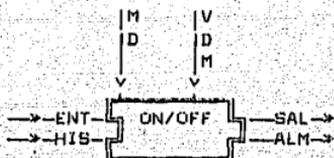
3.8 CONTROL ON / OFF

El controlador On/Off es un bloque que genera una salida digital cuyo estado depende del valor de una entrada analógica con respecto a un punto de referencia.

Se tiene una entrada analógica (ENT) la cual es comparada con un valor de referencia (SP), dependiendo del valor de comparación se genera una salida digital con las siguientes características:

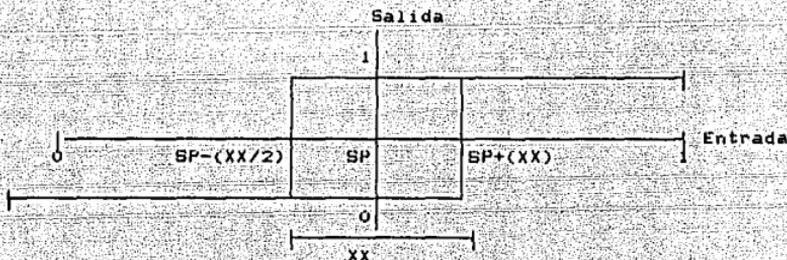
- 0 Si la entrada es menor que el valor de referencia.
- 1 Si la entrada es mayor que el valor de referencia.

Para evitar oscilaciones en la variable digital de salida, al valor de referencia se le aplica una zona de histeresis con lo cual se obtiene una zona muerta que evita que se dañe el elemento final, en el caso de que el controlador lo esté manejando directamente.



HIS Es el valor de Histeresis o zona muerta de activación de la salida, especificada en términos del punto de referencia.

El algoritmo de control ON-OFF tiene el comportamiento presentado en la siguiente figura:



Comortamiento del algoritmo de control ON/OFF

Operación Normal

- 1) Si la salida anterior es igual a cero y la variable de entrada es mayor o igual a $SP+(XX/2)$ entonces la nueva salida es igual uno ($SAL = 1$).
- 2) Si la salida anterior es igual a uno y la variable de entrada es menor o igual a $SP-(XX/2)$ entonces la nueva salida es igual a cero ($SAL = 0$).

3.9 CONTROL HIDRAULICO

Este algoritmo sirve para manejar una válvula hidráulica con una entrada de mando y dos salidas. La entrada la cual es digital, tiene los siguientes mandos: 1 -> Abrir 0 -> Cerrar. Las dos salidas son las acciones de la válvula, una salida para abrir y otra para cerrar.

La válvula hidráulica solo podrá tener dos posiciones abierta o cerrada, no puede tener posiciones intermedias.

Las salidas se activan de acuerdo a dos formas de posicionamiento: por tiempo (pulso monoestable) o por reconocimiento de posición usando interruptores límite.

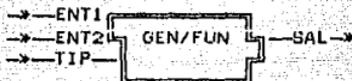
El posicionamiento por tiempo, consiste en aplicar un pulso de duración T a la salida correspondiente. T debe ser mayor o igual al tiempo de asentamiento de la válvula a manejar.

El posicionamiento por interruptor límite consiste en subir la señal digital de la salida correspondiente y bajarla cuando se reciba la señal del interruptor límite que indique que la válvula este completamente abierta o cerrada.

Se tiene otra opción que consiste en desactivar la salida de acuerdo al evento que suceda primero: por tiempo o por interruptor.

3.10 GENERADOR DE FUNCIONES

Este algoritmo puede calcular cuatro tipos de salidas dependiendo de la función seleccionada y del tipo de entrada. Las opciones de entrada son: una variable de proceso o la desviación de dos entradas.



ENT1-ENT2 Entradas analógicas del algoritmo.

TIP Es el tipo de función que calculará el algoritmo, bajo la siguiente clave:

- 1.- Curva S
- 2.- Curva segmento de línea.
- 3.- Curva polinomial.
- 4.- Curva exponencial.

K0 - K5 Constantes asociadas a cada función conforme a las ecuaciones a continuación presentadas.

MOD0 Es el tipo de entrada que se utilizará para el cálculo interno del algoritmo, bajo la siguiente clave.

- (0) E = E0
- (1) E = E0 - E1

Las cuatro funciones son:

Curva S:

La curva S está diseñada para aproximar el comportamiento de posicionadores de válvulas no-lineales.

La ecuación es:

$$\begin{aligned} \text{SALIDA} = & (K0 * E) + \\ & E**2 * (3**K0 - 3**K0 + K1 - 4) / 2 + \\ & E**4 * ((K0 + K1 - 2) / 2) \end{aligned}$$

Donde:

E es E_0 o $(E_0 - E_1)$

y $0 < K_0 < 2$
 $0 < K_1 < 4$

Curva Segmento de Línea:

La curva Segmento de Línea calcula un valor Y para una coordenada X, usando puntos de límite. La ecuación es:

$$\text{SALIDA} = k_2 + ((k_3 - k_2)/(k_1 - k_0)) * (E - k_0)$$

Donde:

k_1 = abcisa presente
 k_0 = abcisa anterior
 k_3 = ordenada presente
 k_2 = ordenada anterior

Curva Polinomial:

La curva Polinomial está diseñada para producir una curva de salida conforme a la siguiente ecuación:

$$\text{SALIDA} = k_0 * E + k_1 * (E ** 2) + k_2 * (E ** 3) + k_3 * (E ** 4) + k_4$$

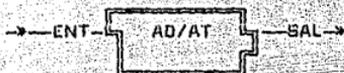
Curva Exponencial:

La curva Exponencial está diseñada para producir una curva de salida conforme a la siguiente ecuación:

$$\text{SALIDA} = k_0 * \text{EXP}(k_1 * E)$$

3.11 COMPENSADOR ADELANTO / ATRASO

Este algoritmo posee la facultad para realizar cierto tipo de acondicionamiento digital de la señal. El atraso (LAG) es esencialmente un filtro pasa bajos digital, y el adelanto (LEAD) un filtro pasa altos.



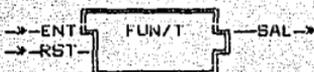
La fórmula básica de operación del algoritmo es:

$$\text{SALIDA (S)} = k (1 + \text{TLD(S)}) / (1 + \text{TLG(S)})$$

TLD = Constante de LEAD en segundos.

TLG = Constante de LAG en segundos.

3.12 FUNCION DE TIEMPO



ENT Valor digital que activa la función de tiempo.
 RST Valor que restablece la salida, la desactiva.

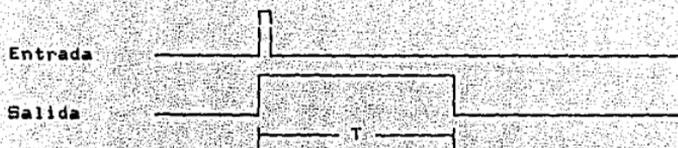
Este algoritmo permite implementar una función de tiempo que puede ser definida en cualquiera de las siguientes configuraciones.

- 1.- Pulso Unico (One-Shot)
 Se produce cuando se activa la entrada, este pulso se mantiene hasta que transcurre un lapso especificado en la variable tiempo, mientras esté transcurriendo el tiempo el algoritmo no tomará en cuenta las activaciones de la salida.
- 2.- Redisparable (Retriggerable One-Shot)
 Se genera un pulso cuando se activa la entrada y se mantiene hasta que transcurre un cierto tiempo, siempre que no se presenten cambios en la entrada ya que de existir una nueva transición, el conteo del tiempo se restablecerá redisparando el algoritmo.
- 3.- Con Retraso (Condition Delay)
 La salida sigue a la entrada con un cierto tiempo de retraso en la activación. En el momento en que la entrada es desactivada se desactiva también la salida.
- 4.- Con Persistencia (Condition Extended)
 La salida sigue a la entrada en la activación. En el momento en que la entrada es desactivada, la salida permanecerá un cierto tiempo activa y posteriormente se desactivará.

En la siguiente figura se presenta los diagramas de tiempo del algoritmo Función de Tiempo.

DIAGRAMAS DE TIEMPO

A) MONOESTABLE



B) REDISPARABLE



C) CON RETARDO

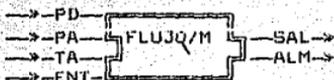


D) CON PERBISTENCIA



3.14 FLUJO DE MASA (RAIZ CUADRADA)

Este algoritmo calcula el flujo de masa de un gas con base a la presión diferencial la presión absoluta y la temperatura.



La salida es proporcional a la raíz cuadrada del cociente de la presión diferencial, multiplicada por la presión absoluta, dividida entre la temperatura, todo multiplicado por una constante, la presión absoluta y la temperatura tienen sus constantes de desplazamiento, por lo que la ecuación es de la siguiente forma:

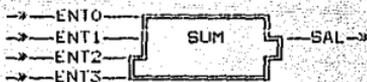
$$SALIDA = k * ((PD * (PA + KPA)) / (TA + KTA))^{1/2}$$

donde:

PD = Presión Diferencial.
 PA = Presión absoluta.
 TA = temperatura.
 KPA = Constante de desplazamiento de PA.
 KTA = Constante de desplazamiento de TA.
 k = Constante General.
 SAL = Es el valor de flujo.

3.15 SUMADOR

El algoritmo Suma produce una salida que corresponde a la suma de cuatro entradas con sus respectivos factores de desplazamiento y escalamiento.



La ecuación de salida es la siguiente:

$$SALIDA = k_4 * (ENTO + k_0) + k_5 * (ENT1 + k_1) + k_6 * (ENT2 + k_2) + k_7 * (ENT3 + k_3) + k_8$$

donde: k_8 = Constante general de desplazamiento.

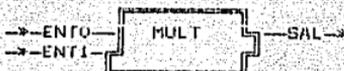
k_n = Constante de desplazamiento.

Los factores de desplazamiento sirven para realizar compensaciones a las entradas, cada una de ellas tiene una constante.

Los factores de escalamiento sirven para poder aumentar o disminuir el peso de cada uno de los componentes en la suma, el otro uso es el de convertir este algoritmo en un restador al proporcionarle constantes de escalamiento negativas.

3.16 MULTIPLICADOR

El algoritmo multiplicador produce una salida que corresponde al producto de dos valores de entrada, cada una con su respectiva constante de desplazamiento.



ENT0-ENT1 Entradas Analógicas del producto.

La ecuación contiene dos constantes generales, una de escalamiento y otra de desplazamiento, la expresión total es la siguiente:

$$\text{SALIDA} = \text{KGE} * (\text{ENTO} + \text{K0}) * (\text{ENT1} + \text{K1}) + \text{KGD}$$

donde:

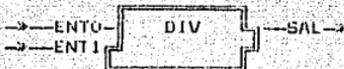
KGE = constante general de escalamiento.
 KGD = constante general de desplazamiento.
 Kn = Constante de desplazamiento.

Los factores de desplazamiento sirven para realizar compensaciones a las entradas, cada una de ellas tiene una constante.

El factor de escalamiento sirve para aumentar o disminuir el valor de la operación resultante.

3.17 DIVISOR

El algoritmo divisor produce una salida que corresponde al cociente de dos entradas, cada una con su respectiva constante de desplazamiento.



La ecuación tiene dos constantes generales, una de escalamiento y otra de desplazamiento, la expresión total es la siguiente:

$$\text{SALIDA} = \text{KGE} * ((\text{ENT1} + \text{K1}) / (\text{ENT0} + \text{K0})) + \text{KGD}$$

donde:

KGE = constante general de escalamiento
 KGD = constante general de desplazamiento
 Kn = Constante de desplazamiento.

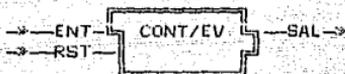
Los factores de desplazamiento sirven para realizar compensaciones a las entradas, cada una de ellas tiene una constante.

El factor de escalamiento sirve para aumentar o disminuir el valor de la operación resultante.

3.18 CONTADOR DE EVENTOS

El contador de eventos registra el número de eventos que ocurren en cada cambio de la entrada, con base a un tipo de transición especificada. La salida es incrementada de acuerdo al valor proporcionado en el parámetro "incr".

Las transiciones del nivel de la entrada que activan el conteo son las siguientes: Alto a Bajo, Bajo a Alto, o ambas.



ENT Valor digital que activa el valor de la salida.

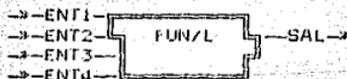
Si la constante VALOR_INI se inicializa con un valor negativo, el contador se decrementará en lugar de incrementarse.

La entrada de RESET hace que la salida se restablezca a un valor inicial especificado.

3.19 FUNCION LOGICA

Este algoritmo permite implementar una función lógica de tres niveles, cada nivel implementa una de las compuertas lógicas siguientes:

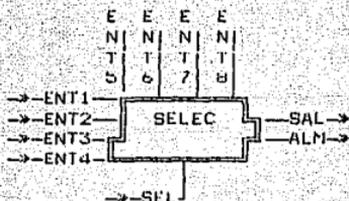
AND, OR, NAND, NOR, XOR O NOT-XOR



Las entradas a la primer compuerta son dos señales digitales. Las entradas a la segunda compuerta son la salida de la primera y una entrada digital. Las entradas a la tercer compuerta son la salida de la segunda y una cuarta entrada digital al algoritmo. La salida de la última compuerta es el resultado del algoritmo que dependerá de los valores de las cuatro entradas.

3.20 SELECTOR DIRECTO

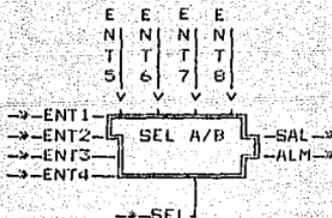
El selector es un algoritmo que permite por medio de una bandera de selección escoger una de sus ocho entradas ENT1 a ENT8 como su salida. Los valores que puede tomar dicha bandera son de 1 a 8 e indica la entrada a escoger.



SEL Es la constante de selección, por medio de la cual se indica la entrada que se ha de reflejar a la salida.

3.21 SELECTOR ALTO / BAJO

Este algoritmo, contiene un selector de mayor o de menor valor entre 8 entradas. El selector de mayor elige como su salida la entrada con el valor más alto, mientras que el de menor la entrada con valor más bajo. Por medio de una bandera se puede acceder al tipo de selector deseado.

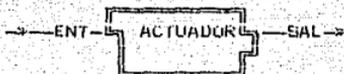


ENT1-ENT8 Entradas analógicas.

SEL Es la entrada de selección por medio de la cual se indica el tipo de salida que se desea: la más alta o la más baja.

3.22 ACTUADOR

Este algoritmo contiene la ecuación para la simulación de un actuador de primer orden que corresponde a la mayoría de los elementos finales de la industria.



Por medio de la especificación de una constante de tiempo, se puede suavizar el comportamiento de una señal, donde esta constante corresponde al tiempo de asentamiento del elemento o válvula a simular.

C A P I T U L O I V

Interfaz Hombre - Máquina de Campo

Introducción

Este capítulo describe la estructura del manejador de Consola, que es una interfaz Hombre - Máquina que permite al usuario interactuar con un sistema de procesamiento en tiempo real.

Los puntos que serán descritos son:

La operación general del sistema así como los comandos que espera y las respuestas que produce. O sea, describe el formato y contenido de la información que entra y sale del manejador.

La organización general de este subsistema, sus módulos constitutivos, la interrelación de estos, así como los flujos de información y de control del manejador.

Funciones

La función principal de la consola, es la de desplegar en la pantalla la información contenida en la base de datos del sistema de Control Distribuido. Estos despliegues serán actualizados cada cierto periodo de tiempo establecido por el usuario.

El Manejador de Consola proporciona un mecanismo de fácil acceso para realizar las siguientes acciones:

- Permitir la entrada de caracteres del teclado.
- Dar facilidades de edición para comandos del teclado.
- Enviar los comandos así generados a las tareas a las que correspondan.
- Ejecutar los siguientes comandos locales :
 - Establecimiento de Fecha y hora.
 - Limpieza de Pantalla.
 - Configuración de la Pantalla.
- Desplegar mensajes de ayuda y mensajes de error.
- Desplegar las tablas que monitorean la información de la base de datos.
- Actualizar la pantalla de acuerdo al ciclo establecido.
- Mantener actualizada la hora.
- Invocar al depurador del sistema.

Características

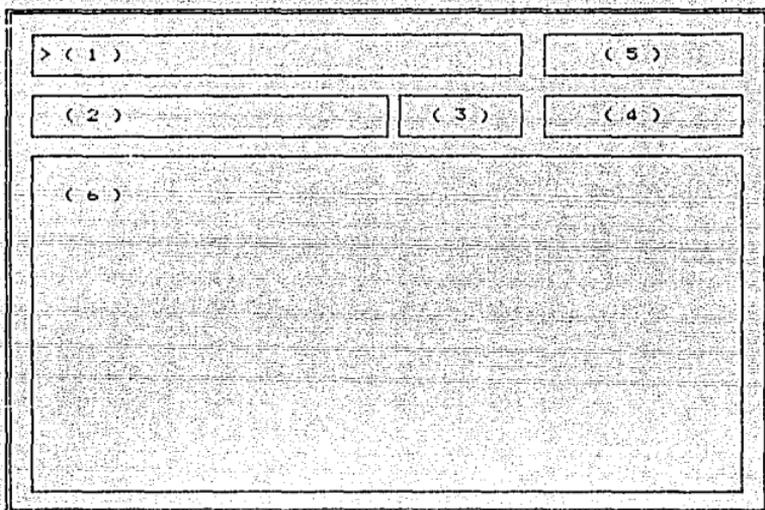
Funcionamiento de la Interfaz H-M.

La consola se encontrará conectada al sistema a través de un canal serie tipo RS-232C. Contará con 6 secciones de pantalla independientes que serán:

- 1 - Sección de Captura de Comandos
- 2 - Sección de Mensajes de Error.
- 3 - Sección de status del CPU.
- 4 - Sección de Alarma.
- 5 - Sección de Fecha - Hora.
- 6 - Sección de Desplegado.

Pantalla. Areas de despliegue

PANTALLA



La Sección de Captura de Comandos consta de 1 línea de 60 caracteres. Su ubicación es renglón 1 columna 1. Este campo puede reconocerse fácilmente puesto que en la columna 1 se despliega un paréntesis triangular (>) que indica la posición a partir de la cual se realizará el eco de lo que se está tecleando en la terminal.

La Sección de Mensajes de Error es un campo de 53 caracteres. Su ubicación es renglón 2 columna 1. En este campo aparecerán los mensajes de ayuda y los mensajes de error que se generen tanto por las rutinas que hacen uso de ésta, como por las de la consola misma.

La Sección de Status de CPU es una sección que se localiza en el renglón 2 columna 55. En este campo, como su nombre lo indica, aparecerá las características de operación de la Unidad de Procesamiento.

La Sección de Alarma es un campo de 19 caracteres ubicado en el renglón 2 columna 61. En esta área se indicará la existencia de errores detectados en la unidad.

La Sección de Fecha - Hora es un campo de 19 caracteres que se localiza en el renglón 1 columna 61. En este campo siempre se mantendrá actualizada la fecha y la hora del sistema.

La Sección de Despliegado, es un Área que comienza en el renglón 3 y tiene una longitud de 80 caracteres. Esta zona puede ser dividida a su vez en varias ventanas dependiendo de las necesidades que se tengan. Es aquí donde se desplegarán las tablas, o la información requerida.

La información que se encuentra desplegada en las ventanas, podrá ser actualizada periódicamente, de acuerdo a lo especificado por el usuario. El ciclo de actualización podrá variar entre 10 y 5000 ms. En caso de que no se desee llevar a cabo la actualización de la información de alguna de las ventanas, esta podrá ser detenida a través de la Tabla de Configuración de Pantalla.

4.1 Comandos generales

En la Sección de Captura se hace el eco de los datos tecleados por el usuario. Se dispone de las facilidades de edición señaladas en la sección de Captura de Comandos.

La sección de Captura tiene por objeto recibir datos del teclado, almacenarlos en un buffer y discriminar si se trata de comando o no.

Un comando esta constituido por un identificador de entrada y por un cuerpo. El identificador de entrada es una clave de tres caracteres que indica la tarea a la que corresponde dicho comando. El cuerpo puede tener una longitud de hasta 124 caracteres y contiene la información que se desea transferir a una tarea.

Se contemplan 2 tipos de comandos: los que son atendidos por el manejador de consola, llamados internos, y los que son atendidos por otras tareas; éstos últimos son llamados externos.

El Manejador de Consola atiende los siguientes comandos:

Limpieza de Pantalla:

Cuando este comando es ejecutado, se limpia la pantalla y se envía un mensaje al campo de mensajes de error con la leyenda:

" CONTROL DISTRIBUIDO U.A.C Sidetec "

L

Establecimiento de Fecha y Hora:

Mediante este comando se establece la fecha y la hora del sistema, misma que es desplegada y actualizada en la pantalla.

T dd-mm-aa hh:mm

Donde:

d	es el día.
m	es el mes.
a	es el año.
h	son las horas.
m	son los minutos.

Llama a la Tabla de Configuración de Pantalla:

Este comando detiene cualquier despliegue realizándose e invoca a la Tabla de Configuración de Pantalla.

V n

Donde:

n	es la ventana donde aparecerá la Tabla de Configuración de Pantalla (TCP).
---	--

Mensajes:

Mensaje que aparece cuando se envía el comando T solo, o bien, cuando existe algún error en el establecimiento de la fecha-hora.

- Marca de Tiempo T dd-mm-aa hh:mm

Causa probable: Los parámetros no fueron introducidos en el orden y formato correcto o fueron mal especificados.

Los mensajes que aparecerán cuando se detecte algún error al configurarse la pantalla son:

- Ciclo de Actualización Fuera de Rango (I - 5500).

El valor proporcionado para el ciclo de actualización está fuera del rango permitido.

- Identificador Incorrecto.

El identificador de la tabla no ha sido reconocido por el sistema. Se recomienda consultar la lista de identificadores de tablas en este mismo documento.

- Indicador Incorrecto (SI - NO).

En este campo, únicamente se espera un caracter: S o N para indicar si se realizará la actualización de la ventana referenciada.

- Limite Fuera de Rango.

El valor proporcionado como limite es mayor o igual a 960. En este campo pueden ser entrados valores hexadecimales los cuales deberán ser precedidos por indentificador hexadecimal "0x..".

El manejador de consola podrá enviar al Coordinador los siguientes comandos:

Modifica Parámetro:

Este comando es utilizado para modificar el valor de un parámetro de una variable de tabla y tipo particular.

C MOP,TTtvvv,nn,VVVVVV

Donde:

- t es la tabla que se desea modificar.
- T es el tipo de tabla:
 - 0: entrada
 - 1: salida
 - 2: calculada.
- v es el número de variable

n es el número de parámetro.
V es el valor (número real).

Consulta Variable:

Este comando es utilizado para consultar todos los parámetros de una variable de una tabla y tipo particular.

C COV,TTtVVV,0,0

Donde:

t es la tabla que se desea consultar
t es el tipo de tabla:
0: entrada
1: salida
2: calculada.
V es el número de la variable.

Configura Variable:

Este comando es utilizado para configurar todos los valores necesarios para una variable. La cantidad de valores y sus rangos dependerán del tipo de tabla al que pertenece la variable.

C CNF,XXXXXXXX...XXXXX

Donde:

X es la cadena de caracteres que le indicará al Coordinador. La tabla a configurar, el tipo, la variable y los valores de los parámetros correspondientes a este tipo de variable.

Consulta Cambios:

Este comando es utilizado para consultar todos los parámetros de las variables que cambiaron de una tabla y tipo particular.

C COC,TTt000,0,0

Donde:

t Es la tabla de la que se desean consultar los cambios.
t es el tipo de tabla:
0: entrada
1: salida
2: calculada.

Consulta Tabla:

Este comando es utilizado para consultar todos los parámetros de todas las variables de una tabla y tipo particular.

C COT,TTt000,0,0

Donde:

t es la tabla que se desea consultar.

t es el tipo de tabla:
 0: entrada
 1: salida
 2: calculada.

Consulta Estado de la Unidad:

Este comando es utilizado para conocer en que estado se encuentra la Unidad de Control.

C EST,0,0,0

Arranque en Frío:

Este comando le indica al Coordinador que se debe hacer una inicialización de toda la base de datos para realizar un arranque en frío.

C ARR,0,0,0

Fin de Configuración:

Este comando se encarga de indicarle al Coordinador que la configuración que se estaba realizando, ha terminado.

C FIN,0,0,0

Los mensajes que aparecerán como resultado de estas operaciones son:

- Comando Procesado Satisfactoriamente.
- Comando Inválido.
 El error fue detectado por las rutinas de Validación (elemento inválido, operación inválida error en congruencia, etc)..
- Error en Ejecución.
 El error fue detectado por las rutinas de Ejecución (Comando incompleto, variable no existente, unidad fuera de servicio etc).

4.2 Captura de Comandos

4.2.1 Descripción

La captura de comandos cuenta con un buffer de 128 caracteres para la edición de comandos. Realiza el eco en pantalla de los caracteres tecleados por el usuario y proporciona algunas facilidades de edición. Cuenta además con una memoria de los últimos 10 comandos introducidos, los cuales son almacenados en una tabla circular. De esta forma, podrán recordarse comandos anteriores o posteriores a la posición actual.

Las facilidades de edición proporcionadas son:

```

cntl A  Mueve el cursor a la izquierda.
cntl S  Mueve el cursor a la derecha.
cntl Q  Posiciona el cursor al inicio.
cntl E  Posiciona el cursor al final.
rubout  Borra el caracter anterior.
cntl D  Borra el caracter actual.
cntl W  Pasa al buffer de edición anterior.
cntl Z  Pasa al buffer de edición siguiente.

```

4.2.2 Tabla de Configuración de Pantalla. Descripción

TABLA DE CONFIGURACION DE PANTALLA

CICLO DE ACTUALIZACION => 10

VENTANA 1:

```

IDENTIFICADOR DE TABLA =>
ACTUALIZACION ACTIVADA => NO
LIMITE INF A DESPLEGAR => 0
LIMITE SUP A DESPLEGAR => 0

```

VENTANA 2:

```

IDENTIFICADOR DE TABLA =>
ACTUALIZACION ACTIVADA => NO
LIMITE INF A DESPLEGAR => 0
LIMITE SUP A DESPLEGAR => 0

```

Fig. 4.2.1 Formato de la Tabla de Configuración de Pantalla

Ciclo de actualización: En este campo se especifica la frecuencia del ciclo de actualización y se configura con datos dentro del rango de 1 y 5500 ms. En caso de introducir un valor fuera de

este rango se desplegará el mensaje " Ciclo de Actualización Fuera de Rango (1 - 5500)" y se restablecerá el campo con el valor por omisión. Para pasar al campo siguiente se deberá oprimir la tecla 'Enter'.

Identificador de la Tabla: En este campo se indica la tabla que se desea desplegar, para lo cual deberá emplearse el mnemónico asociado a dicha tabla.

Limite inf a desplegar: En este campo normalmente se selecciona el limite inferior de la tabla que se desea desplegar. Cuando se elige el desplegado de la Tabla de Ranuras, este campo indicará el número de ranura que se desea desplegar.

Limite sup a desplegar: En este campo se selecciona el limite superior de la tabla que se desea desplegar. El número máximo de renglones a desplegar son 16.

4.2.3 Identificadores de tablas.

Apuntadores Adquisición.....	ADQ_AP
Adquisición. Procesador 0.....	ADQ_PA
Adquisición. Procesador 1.....	ADQ_PB
Adquisición. Procesador 2.....	ADQ_PC
Adquisición. Procesador 3.....	ADQ_PD
Registros de Adquisición.....	ADQ_RA
Variables a Muestrear.....	ADQ_VA
Tabla de Definición.....	DEF_MR
Despliega Memoria.....	DES_ME
Despliega Stack.....	DES_ST
Definición de Variables Analógicas - Entrada.....	DVA_SE
Definición de Variables Analógicas - Salida.....	DVA_SS
Definición de Variables Digitales - Entrada.....	DVD_SE
Definición de Variables Digitales - Salida.....	DVD_SS
Apuntadores a Ranuras.....	RAN_AP
Relfislog - Tabla Base.....	REL_FL
Relfislog - Entradas Analógicas.....	RFL_EA
Relfislog - Salidas Analógicas.....	RFL_SA
Relfislog - Entradas Digitales.....	RFL_ED
Relfislog - Salidas Digitales.....	RFL_SD
Respuesta del Coordinador.....	RES_CO
Secuenciación Marco 0.....	SEC_MA
Secuenciación Marco 1.....	SEC_MB
Secuenciación Marco 2.....	SEC_MC
Secuenciación Marco 3.....	SEC_MD
Secuenciación de Ranuras.....	SEC_RN
Tabla de Errores.....	TAB_ER
Mensajes del Usuario.....	TAB_MU
Plan de Emergencia.....	TAB_PL
Tabla de Ranuras.....	TAB_RN
Tareas Activas.....	TAR_AC
Tipo de Tarjeta.....	TIP_TA
Apuntadores a la Base de Datos.....	TSC_AP
Control Base de Datos.....	TSC_BD
Variables Analógicas - Sección Calculadas.....	TVA_SC
Variables Analógicas - Sección Entrada.....	TVA_SE
Variables Analógicas - Sección Salida.....	TVA_SS
Variables Digitales - Sección Calculadas.....	TVD_SC
Variables Digitales - Sección Entrada.....	TVD_SE
Variables Digitales - Sección Salida.....	TVD_SS
Apuntadores a Variables.....	VAR_AP

4.3 Despliegue de Tablas

4.3.1 Descripción general

Funcionamiento de la Interfaz H-M.

El despliegue de tablas tiene como función principal desplegar la información contenida en la base de datos del sistema. Esta acción es realizada por una biblioteca de rutinas, en la que cada una accesa una determinada sección de la base de datos y la despliega en forma de tabla.

A continuación se describe brevemente el contenido de cada una de las tablas:

4.3.2 Descripción de la Tablas

Tabla Apuntadores Adquisición:

Esta tabla despliega los apuntadores de Adquisición: dir_cont_aña[] y ad_fito.

Adquisición Procesador X:

Donde X es el número del procesador (0 - 3).

Despliega el contenido de la estructura ADQ-PR: identificador y variable para el procesador indicado.

Registros de Adquisición:

Despliega los registros de adquisición. Para realizar esta acción lee y despliega el contenido del registro "reg_adq".

Variabes a Muestrear:

Despliega el contenido de la tabla "variables a muestrear". Para realizar esta acción lee y despliega la identificación de la variable contenida en el arreglo de enteros "var_mues".

Tabla de Definición:

Despliega el número de ranura configurada en el arreglo de estructuras ce_Defn_Rnr para cada marco.

Despliega Memoria:

Despliega una zona de memoria elegida y la presenta en formato hexadecimal y su equivalente en ASCII. La selección de esta tabla debe hacerse en la ventana 1 puesto que ocupa la pantalla completa para su despliegue.

Despliega Stack:

Despliega el area de stack de una tarea y la presenta en formato hexadecimal y su equivalente en ASCII. La selección de esta tabla debe hacerse en la ventana 1 puesto que ocupa la pantalla para su despliegue.

Definición de entradas Analógicas:

Despliega la tabla de definición de variables analógicas sección de entradas:

Status	Estado de la variable:
	"Libre " Libre
	"En Ser" En servicio
	"Fuera " Fuera de Servicio

T	"Paro E" Paro de Emergencia
	"En Trn" En transición
	"Config" Configurada
T	Tipo de variable. (La entrada).
UFis	Ubicación física.
R SN	Rango superior de normalización.
K IN	Rango inferior de normalización.
GCu	Ganancia del convertidor.
Sn	Sensibilidad.
Aa	Algoritmo de acondicionamiento.
LCsup	Límite crítico superior.
LPCsu	Límite precritico superior.
LCinf	Límite crítico inferior.
LPCin	Límite precritico inferior.
LDasc	Límite de desviación ascendente.
LDdes	Límite de desviación descendente.

Definición de Salidas Analógicas:Definición de Entradas Digitales:Definición de Salidas Digitales:

Despliega la tabla de definición de variables correspondiente :

##	Número de variable.
Status	Estado de la variable: "Libre " Libre "En Ser" En servicio "Fuera " Fuera de Servicio "Paro E" Paro de Emergencia "En Trn" En transición "Config" Configurada
Tipo	Tipo de variable "E" Entrada. "S" Salida.
Ubicación	Ubicación física.
Ind_S	Indicador de servicio.

Apuntadores a Ranuras:

Despliega la tabla de apuntadores a ranuras. Visualiza el contenido del arreglo ce_Tab_Var.rn_tabla.

Relatividad:

Despliega las tablas: Base, Entradas Analógicas, Salidas Analógicas, Entradas Digitales y Salidas Digitales. Todas las tablas tienen los mismos campos. La única que difiere en formato es la tabla base, que comprende los datos relacionados con las tarjetas: Comunicaciones, Centinela y Controladoras.

Csta	Canasta en la que se encuentra ubicada la variable.
Slot	Slot (ranura) en la que se encuentra ubicada la variable.
Ttarj	Tipo de tarjeta. "Com " Comunicaciones. "C_Ean" Controlador de entradas analógicas. "EntAn" Entradas Analógicas "SAn_V" Salidas Analógicas de voltaje.

Funcionamiento de la Interfaz H-M.

	"SAn_C"	Salidas Analógicas de Corriente.
	"EntDg"	Entradas Digitales.
	"SalDg"	Salidas Digitales
	"Cent "	Centinela
Tvar		Tipo de variable.
	"SDig"	Salidas Digitales.
	"EDig"	Entradas Digitales.
	"SA_V"	Salidas Analógicas de Voltaje.
	"SA_C"	Salidas Analógicas de Corriente.
	"EA_V"	Entradas Analógicas de Voltaje.
	"EA_C"	Entradas Analógicas de Corriente.
Polar		Polaridad de la variable.
Cnal		Canal en que se encuentra ubicada la variable.
Status		Estado operacional.
	"No_Err"	No error
	"Incorr"	Tarjeta incorrecta.
	"No_Res"	La tarjeta no responde como debe.
	"Conect"	Tarjeta sin conector a campo.
	"Polard"	Polaridad incorrecta en SAC-511.
	"CntMUp"	Error en control de modo de operación (SAC-611).
	"NoNorm"	La tarjeta no estaba en modo normal al accederla (SAC-410, SAC-611) .
	"Er_SAC"	Indicador general de error en el SAC.

Respuesta del Coordinador:

Despliega el contenido del buffer en el que el Coordinador envía su mensaje de respuesta a la solicitud de un comando de consulta.

Secuenciación de Ranuras por Marco:

Despliega, para el marco solicitado, la tabla de secuenciación de ranuras. Además proporciona la información de número total de ranuras por marco y duración acumulada por marco.

#		Número de línea.
Status		Estado de la ranura:
	"Libre "	Libre
	"En Ser"	En servicio
	"Fuera "	Fuera de Servicio
	"Paro E"	Paro de Emergencia
	"En Trn"	En transición
	"Config"	Configurada
Ran		Número de Ranura configurada.
Dur		Duración en unidades de procesamiento.
Algoritmo		Algoritmo configurado.
(##)		Número de algoritmo.
	0	Dummy Idem.
	1	Pid Idem.
	2	PidRazon Pid.Razón.
	3	PidBanda Pid.Banda.
	4	EstAnIMA Estación Analógica Manual / Automatico.
	5	PidBatch Pid.Batch.

Funcionamiento de la Interfaz H-M.

6	FlujoMas	Flujo de Masa
7	Selector	Idem.
8	Suma	Idem.
9	Producto	Idem.
10	Divisor	Idem.
11	EstDigMA	Estación Digital Manual / Automático.
12	ContEven	Contador de Eventos.
13	Fcn_Tmpo	Función de Tiempo.
14	FcnLogic	Función Lógica.
15	Gen_func	Generador de Funciones.
16	CompAdAt	Comparador Adelanto/ Atraso.
17	SelecA/B	Selector Alto / Bajo.
18	Actuador	Idem.
19	On_Off	Idem.
20	VIV_Hdr	Valvula Hidráulica.

Secuenciación de Ranuras:

Despliega en forma abreviada y completa la tabla de secuenciación de ranuras. Es decir, incluye la información correspondiente a los marcos 0 a 3. Además proporciona la información de número total de ranuras y duración total.

#	Número de línea.	
St	Estado de la ranura:	
	"Lb "	Libre
	"ES"	En servicio
	"FS"	Fuera de Servicio
	"PE"	Paro de Emergencia
	"ET"	En transición
	"Cf"	Configurada
Rn	Número de Ranura configurada.	
Al	Algoritmo configurado.	
	0	Dummy Idem.
	1	Pid Idem.
	2	PidFazon Pid Fazon.
	3	PidBanda Pid Banda.
	4	EstAnIMA Estación Analógica Manual / Automático.
	5	PidBatch Pid Batch.
	6	FlujoMas Flujo de Masa.
	7	Selector Idem.
	8	Suma Idem.
	9	Producto Idem.
	10	Divisor Idem.
	11	EstDigMA Estación Digital Manual / Automático.
	12	ContEven Contador de Eventos.
	13	Fcn_Tmpo Función de Tiempo.
	14	FcnLogic Función Lógica.
	15	Gen_func Generador de Funciones.
	16	CompAdAt Compensador Adelanto/Atraso.
	17	SelecA/B Selector Alto / Bajo.
	18	Actuador Idem.
	19	On_Off Idem.

Funcionamiento de la Interfaz H-M.

20 Vlv_Hdr VÁLVULA Hidráulica.

Tabla de Errores:

Despliega en forma de columnas los errores registrados en la tabla de errores, sus identificaciones y estadísticas.

Clave	Es la clave del error ocurrido.
Descrip	Es el descriptor del error.
Tarea	Tarea que reportó el error.
Repor	Indica si el error ya fue reportado.
#Cons	Número de error consecutivo.
Ocur	Ocurrencia de este tipo de error.

Mensajes del Usuario:

Despliega el contenido del buffer 'ct_dtbuff', mismo que es utilizado para almacenar mensajes provenientes de rutinas que se sirvieron de la utilería "wprintf".

Paro de Emergencia:

Despliega los indicadores de emergencia 1 y 2.

EDO n	Estado operacional del indicador n (1 y 2).
PARO DE EMERGENCIA n	

Bajo este rótulo se despliega la dirección, número de bit y valor del indicador n (1 ó 2) en ese orden:

Dirección : bit = valor

Tareas Activas:

Despliega la información relacionada con las tareas que se encuentran registradas en el sistema.

Nombre	Es el nombre de la tarea.
Ident	Es el identificador de la tarea. Este es expresado en ambos formatos hexadecimal y decimal.
St	Es el estado operacional de la tarea.
Inic	(stack) dirección de inicio del stack para esa tarea en particular.
Fin	(stack) dirección del fin del stack para una tarea en particular.
Ocp	Porcentaje de ocupación del stack.

Tipo de Tarjetas:

Despliega la ubicación, el estado y el tipo de tarjeta contenido en la tabla TDF.

Ubicación	Ubicación de la tarjeta.
Status	Estado de la tarjeta:
	"Unipolar-sen" Unipolar sencillo.
	"Bipolar -sen" Bipolar sencillo.
	"Unipolar-dif" Unipolar diferencial.
	"Bipolar -dif" Bipolar diferencial.
	"Sin Conector" Sin conector a campo.
	"Falla Física" Idem.
	"No cont modo" No controla modo.
	"Sin tarjeta " Idem.
Ident	Identificación de la tarjeta:

"Com "	Comunicaciones.
"C_Ean"	Controlador de entradas analógicas.
"EntAn"	Entradas Analógicas
"SAn_V"	Salidas Analógicas de voltaje.
"SAn_C"	Salidas Analógicas de corriente.
"EntDg"	Entradas Digitales.
"SalDg"	Salidas Digitales.
"Cent "	Centinela

Apuntadores a la base de datos:

Accesa la estructura de control de la base y despliega los apuntadores a cada una de las tablas de la base de datos. La información correspondiente a las ranuras, la despliega en función del parámetro "límite inferior" que recibe de la Tabla de Configuración de Pantalla.

Status	Estado de la tabla correspondiente.
"Libre "	Idem.
"Tab Op"	Tabla ocupada.
"Smf Bq"	Semáforo bloqueado.
"Tab Iv"	Tabla Inválida.
Tbl	Tabla de la cual se despliega la información.
"ANL"	Variabes analógicas.
"DIG"	Variabes digitales.
"DAN"	Definición de variables analógicas.
"DDG"	Definición de variables digitales.
"KAN"	Tabla de ranuras.
"RFL"	Relfisiog.
"AP0"	Adquisición procesador 0.
"AP1"	Adquisición procesador 1.
"AP2"	Adquisición procesador 2.
"AP3"	Adquisición procesador 3.
"TPE"	Tabla de paro de emergencia.
Entrada	Apuntador al inicio de la sección de variables de entrada.
Salida	Apuntador al inicio de la sección de variables de entrada.
Calculada	Apuntadores al inicio de la sección de variables calculadas.
Otros	"RD" Ranura desplegada.
	"FC" Ranuras configuradas.
	"RT" Ranuras totales.

Control base de datos:

Accesa la estructura de control de la base de datos y despliega semáforos, usuarios y número de campos / ranuras de las tablas de la base de datos.

Status	Gral.
"Libre "	Idem.
"Tab Op"	Tabla ocupada.
"Smf Bq"	Semáforo bloqueado.
"Tab Iv"	Tabla Inválida.
Status	Estado operacional de la tabla
"Libre "	Idem.
"Tab Op"	Tabla ocupada.

Funcionamiento de la Interfaz H-M.

	"Smf Bq"	Semáforo bloqueado.
	"Tab Iv"	Tabla Inválida.
Semf		Número de semáforo.
Usur		Número de usuario.
Tabla		Tabla cuya información es desplegada.
	"VAR_ANL"	Tabla de variables analógicas.
	"VAR_DIG"	Tabla de variables digitales.
	"DEF_ANL"	Definición de variables analógicas.
	"DEF_DIG"	Definición de variables digitales.
	"RANURAS"	Tabla de ranuras.
	"REL_FLG"	Tabla relación física - lógica.
	"ADQ_PKO"	Tabla Adquisición Procesador 0.
	"ADQ_PRI"	Tabla Adquisición Procesador 1.
	"ADQ_PK2"	Tabla Adquisición Procesador 2.
	"ADQ_PR3"	Tabla Adquisición Procesador 3.
	"PAR_EMG"	Tabla Paro de Emergencia.
Ent		Número de variables configuradas para la sección de entrada.
Sal		Número de variables configuradas para la sección de salida.
Cal		Número de variables configuradas para la sección de calculadas.
RT		Número de ranuras totales.

Tabla de variables analógicas:

Despliega la información relacionada a las variables analógicas:

##		Número de variable.
Status		Estado de la variable:
	"Libre "	Libre
	"En Ser"	En servicio
	"Fuera "	Fuera de Servicio
	"Paro E"	Paro de Emergencia
	"En Trn"	En transición
	"Config"	Configurada
Tipo		Tipo de variable
	"E"	Entrada.
	"S"	Salida.
	"C"	Calculada
V_NOR		Valor en rango normalizado.
IA&C		Indicador de alarmas.
V_UI		Valor en unidades de ingeniería.
IND_S		Indicador de servicio.

Tabla de variables digitales:

Despliega la información relacionada con las variables digitales:

##		Número de variable.
Status		Estado de la variable:
	"Libre "	Libre
	"En Ser"	En servicio
	"Fuera "	Fuera de Servicio
	"Paro E"	Paro de Emergencia
	"En Trn"	En transición

Funcionamiento de la Interfaz H-M.

	"Config" Configurada
Tipo	Tipo de variable
	"E" Entrada.
	"S" Salida.
	"C" Calculada
Valor	Valor digital.
Ind.C	Indicador de bits de cambio.
IND_S	Indicador de servicio.

Apuntadores a variables:

Despliega la información de la tabla de variables. Es decir, presenta el contenido de la estructura de Tab_Var.

Apuntador Apuntador a la tabla referenciada.

NcC Número de Campos configurados.

4.3.3 Descripción de la Tabla de Ranuras

Tabla de Ranuras:

Despliega el algoritmo configurado para una ranura determinada. Se dispone de una librería de 20 tablas para desplegar un número igual de algoritmos:

Definiciones comunes:

Status Estado operacional de la ranura:

"Libre " Libre

"En Ser" En servicio

"Fucha " Fuera de Servicio

"Paro L" Paro de Emergencia

"En Trn" En transición

"Config" Configurada

INDICADOR DE SERVICIO

Representación binaria del indicador de servicio en donde el bit más significativo es el que está más a la izquierda.

A1 Código del Algoritmo configurado en esa ranura.

SCal Valor de salida calculada.

HISA Valores de historia en modo automático.

HISM Valores de historia en modo manual.

ALARMA Y MUDO

Representación binaria de la palabra de alarma y modo en donde el bit más significativo es el que está más a la izquierda.

PID:

Modo Modo de operación:

Funcionamiento de la Interfaz H-M.

	"Man_Loc"	Manual Local.
	"Man_Rem"	Manual Remoto.
	"Ret_Man"	Retroajuste Manual.
	"Imp_Man"	Imposición Manual.
	"Automat"	Automático.
	"Superv "	Supervisorio.
	"Comput "	Computadora.
KP		Constante Proporcional.
KI		Constante de Integración.
KD		Constante de derivación
SP		Punto de Ajuste Local.
PL		Compensación para entrada cero.
KPA		Constante de Prealimentación.
A-C		Abrir / Cerrar.
KLAE		Límite alto en variable de entrada.
KLBI		Límite bajo en variable de entrada.
KLAS		Límite alto en variable de salida.
KLBS		Límite bajo en variable de salida.
VDM		Valor deseado.
VAR DE PROCESO		Dirección y valor de la variable de entrada. Esta información está expresada con el siguiente formato: Dirección := valor
SALIDA		Salida. Esta información está expresada con el siguiente formato: Dirección := valor
IMPOSICION		Dirección, bit y valor del indicador de imposición. Esta información está expresada con el siguiente formato: Dirección : bit = valor
CONFIGURACION		Representación binaria de la palabra de configuración en donde el bit más significativo es el que está más a la izquierda.
PREALIMENTACION		Dirección y valor de la variable de prealimentación. Esta información está expresada con el siguiente formato: Dirección := valor
PID RAZON:		
Modo		Modo de operación: (Ver descripción PID).
KP		Constante Proporcional.
KI		Constante de Integración.
KD		Constante de derivación
SP		Punto de Ajuste Local.
PL		Compensación para entrada cero.
KPA		Constante de Prealimentación.
A-C		Abrir / Cerrar.

KLAE Limite alto en variable de entrada.
 KLBE Limite bajo en variable de entrada.
 KLAS Limite alto en variable de salida.
 KLBS Limite bajo en variable de salida.
 VDM Valor deseado.
 VAR DE PROCESO (Ver descripción PID).
 SALIDA (Ver descripción PID).
 IMPOSICION (Ver descripción PID).
 CONFIGURACION (Ver descripción PID).
 PREALIMENTACION (Ver descripción PID).
 RZN Factor de proporción para la referencia remota.
 PLC Termino de compensación para la referencia remota.
 RSUP Número de la ranura superior de la cascada.
 T-R Tipo de relación para la referencia remota.
 AJUSTE RMTD. Dirección y valor del punto de ajuste remoto.
 Esta información está expresada con el siguiente formato:
 Dirección := valor
 RAJT Valor de retroajuste producido.

PID BANDA:

Modo Modo de operación: (Ver descripción PID).
 KP Constante Proporcional.
 KI Constante de Integración.
 KD Constante de derivación.
 SP Punto de Ajuste Local.
 PL Compensación para entrada cero.
 KPA Constante de Prealimentación.
 A-C Abrir / Cerrar.
 KLAE Limite alto en variable de entrada.
 KLBE Limite bajo en variable de entrada.
 KLAS Limite alto en variable de salida.
 KLBS Limite bajo en variable de salida.
 VDM Valor deseado.
 VAR DE PROCESO (Ver descripción PID).
 SALIDA (Ver descripción PID).
 IMPOSICION (Ver descripción PID).
 CONFIGURACION (Ver descripción PID).
 PREALIMENTACION (Ver descripción PID).
 GB Constante de proporción de la banda.
 PAB Punto de Activación de la banda.

ESTACION ANALOGICA M/A:

Modo Modo de operación: (Ver descripción PID).
 KE Constante de proporción de la entrada.
 VKE Constante de polarización de la entrada.
 A-C Abrir / Cerrar.
 KLAE Limite alto en variable de entrada.
 KLBE Limite bajo en variable de entrada.

Funcionamiento de la Interfaz H-M.

KLAS Limite alto en variable de salida.
 KLBS Limite bajo en variable de salida.
 VDM Valor deseado.
 VAR DE PROCESO (Ver descripción PID).
 SALIDA (Ver descripción PID).
 IMPOSICION (Ver descripción PID).
 CONFIGURACION (Ver descripción PID).

PID BATCH:

Modo Modo de operación: (Ver descripción PID).
 KP Constante Proporcional.
 KI Constante de Integración.
 KD Constante de derivación.
 UP Punto de Ajuste Local.
 PL Compensación para entrada cero.
 KPA Constante de Prealimentación.
 A-C Abrir / Cerrar.
 KLAE Limite alto en variable de entrada.
 KLBE Limite bajo en variable de entrada.
 KLAS Limite alto en variable de salida.
 KLBS Limite bajo en variable de salida.
 VDM Valor deseado.
 RSTREM Reset remoto.
 VAL_IN Valor inicial.
 VAR DE PROCESO (Ver descripción PID).
 SALIDA (Ver descripción PID).
 IMPOSICION (Ver descripción PID).
 CONFIGURACION (Ver descripción PID).
 PREALIMENTACION (Ver descripción PID).
 RESTAURACION Dirección, bit y valor de la variable de restauración. Esta información está expresada con el siguiente formato:
 Dirección : bit = valor

FLUJO DE MASA:

KTA Constante de polarización de la temperatura.
 KG Constante de proporción general de la ecuación.
 KPL Limite inferior de presión diferencial.
 KPA Constante de polarización de la presión absoluta.
 PRESION DIF Dirección y valor de la variable de presión diferencial. Esta información está expresada con el siguiente formato:
 Dirección := valor
 PRESION ABS Dirección y valor de la variable de presión absoluta. Esta información está expresada con el siguiente formato:
 Dirección := valor

TEMPERATURA

Dirección y valor de la variable de temperatura. Esta información está expresada con el siguiente formato:

Dirección := valor

SELECTOR:**VAR DE ENT**

Dirección y valor de la variable de entrada. Esta información está expresada con el siguiente formato:

Dirección := valor

SE

Selector de entrada.

SUMA:**CESC**

Constante de escalamiento general de la ecuación.

ENTRADA

Dirección y valor de la variable de entrada. Esta información está expresada con el siguiente formato:

Dirección := valor

CPRO

Constante de proporción entrada 1 a 8.

PRODUCTO:**ENTRADA**

(Ver descripción SUMA).

CTE

Constante de entrada 1 y 2.

ESCL

Constante de escalamiento general de la ecuación.

DESP

Constante de desplazamiento de la multiplicación.

DIVISOR:

(Ver descripción PRODUCTO).

ESTACION DIGITAL M/A:**Modo**

Modo de operación: (Ver descripción PID).

VAR DE ENTRADA

Dirección, bit y valor de la variable de entrada. Esta información está expresada con el siguiente formato:

Dirección : bit = valor

VAR DE SALIDA

Dirección, bit y valor de la variable de salida. Esta información está expresada con el siguiente formato:

Dirección : bit = valor

TIPO

Tipo de Alarma.

VDM

Valor digital deseado.

VE

Valor de Emergencia.

RSTREM Reset remoto.
 RESTAURACION
 (Ver descripción PID BATCH)

CONTADOR DE EVENTOS:

VAR DE ENTRADA
 (ver descripción ESTACION DIGITAL)
 TIPO Tipo de activación (Cambio Asc / Dsc).
 INCR Incremento.
 V_INI Valor inicial del contador.
 RESTAURACION
 (Ver descripción PID BATCH)

FUNCION TIEMPO:

VAR DE ENTRADA
 (ver descripción ESTACION DIGITAL)
 TIPO Tipo de activación.
 TMPO Tiempo (x 500 msec).
 RESTAURACION
 (Ver descripción PID BATCH)

FUNCION LOGICA:

ENTRADA (ver descripción de SUMA).
 CPRU Constante de función lógica 0 - 2.

GENERADOR DE FUNCIONES:

VAR DL ENI
 (ver descripción de SELECTOR).
 MD Modo.
 KG Constantes generales 0 - 5.

COMPARADOR ADELANTO/ATRASO:

BLA Constante de atraso.
 BLL Constante de adelanto.
 KEA Constante de escalamiento para atraso.
 KEE Constante de escalamiento para adelanto.
 COMP Constante de corrección de compensación.
 VAR DE ENTRADA
 (ver descripción ESTACION DIGITAL)

SELECTOR A/B:

VAR DE ENT Dirección y valor de la variable de entrada.
 Esta información está expresada con el
 siguiente formato:
 Dirección := valor
 SE Selector de entrada.

ACTUADOR:

VAR DE ENTRADA
 (ver descripción ESTACION DIGITAL).
 CTMPO Constante de tiempo.
 SALIDA Dirección de la variable de salida.

ON / OFF:

VAR DE ENTRADA
 (ver descripción ESTACION DIGITAL).
 VAR DE SALIDA
 (ver descripción ESTACION DIGITAL).
 SP Punto de Ajuste Local.
 VDM Valor digital deseado.
 VE Valor de Emergencia.
 RSTREM Reset remoto.
 IMPOSICION
 (ver descripción PID)
 RESTAURACION
 (ver descripción PID BATCH)

VALVULA HIDRAULICA:

VAR DE ENTRADA
 (ver descripción ESTACION DIGITAL).
 TPO_INT
 VE Valor de Emergencia.
 Limite (abierto) Limite variable de salida abrir.
 Salida (abierto) Salida variable de salida abrir.
 Limite (cerrado) Limite variable de salida cerrar.
 Salida (cerrado) Salida variable de salida cerrar.
 TMPO tiempo.
 TPO_RST tipo de restablecimiento.

CAPITULO V

Arquitectura del Interfaz Hombre - Máquina de Campo

5.1 Manejador de Consola

El manejador de consola propiamente dicho está compuesto por dos grupos de rutinas que realizan las funciones de entrada, salida, registro y coordinación.

El núcleo de la consola está dividido en secciones:

- Sección de Recepción.
- Sección de Transmisión.

La Sección de Recepción comprende la captura de caracteres, las facilidades de edición, la realización del eco en pantalla, la interpretación de los comandos internos y el despliegue y establecimiento de la Tabla de Configuración de Pantalla.

La Sección de Transmisión cuenta con una biblioteca de rutinas que monitorean la base de datos. Se encarga de desplegar periódicamente cualquiera de las tablas contenidas en la biblioteca, siempre que el usuario así lo especifique. También monitorea la banderas de error del sistema y mantiene actualizada la fecha y la hora desplegadas en la pantalla.

5.2 Módulos Constitutivos - Recepción

En la Sección de Recepción se hace el eco de los datos tecleados por el usuario. Se cuenta con las siguientes facilidades de edición. Esta sección, como ya se dijo, tiene por objeto recibir datos del teclado, almacenarlos en un buffer y discriminar si se trata de comando o no.

aactivi:

A través de esta rutina se establece la activación de la actualización en la Tabla de Configuración de la Pantalla. El programa fija los valores establecidos con anterioridad y llama a la rutina "eco" que los desplegará. A continuación espera los caracteres válidos para continuar: S, N, CR, o "W", en caso contrario desplegará un mensaje de error y mantendrá el valor por omisión.

act:

Fija el dato establecido con anterioridad o el valor default, acepta los caracteres numéricos que el usuario teclee y verifica que el dato esté comprendido dentro de cierto rango, en caso contrario se desplegará un mensaje de error y se mantendrá el valor default. Durante la edición de este dato puede emplearse la tecla espacio hacia atrás.

carátulas:

Con el offset de desplegado recibido como parámetro, esta rutina posiciona el cursor en la ventana correspondiente y despliega la carátula de la Tabla de Configuración de Pantalla.

cons_rx:

Luego de haber sido iniciada por el sistema operativo, esta tarea llama a la rutina de inicialización "inic_rx" y entra en un ciclo permanente en donde llama a las rutinas: editor e inter. Mismas que se encargan de entrar e interpretar la información respectivamente. A través de esta rutina se llama al depurador del sistema al encontrarse activa la bandera correspondiente.

eco:

Formatea la cadena de caracteres recibida e introduce una secuencia de caracteres de control para activar el video inverso de la posición del cursor. Los parámetros recibidos "linea" y "columna" permiten desplegar la cadena en cuestión en cualquier posición dentro de la pantalla.

editor:

Editor es la rutina que permite la entrada de datos a través del teclado y que además proporciona algunas facilidades de edición.

Acepta los siguientes caracteres:

0 - 9, A - Z, a - z, "A", "C", "D", "E", BS, LF, "K",
CR, "Q", "S", "W", "Z", ESC, "!", "+", "-", ".", ":",
"/", ";", "=", "\\", barra, espacio.

Proporciona los comandos de edición:

Espacio hacia atrás,
Aumento de línea - retorno de carro,
supresión de carácter,
recuerda buffer anterior,
recuerda buffer siguiente,
cursor a la izquierda,
cursor a la derecha,
origen,
final,
cancelación - llama al depurador,
almacena carácter (crea buffer).

errores:

recibe como parámetro el código correspondiente a un mensaje de error y despliega dicho mensaje en la Sección de Mensajes de Error o en la Sección de Alarma.

- 0 Comando Procesado Satisfactoriamente
- 1 Comando Inválido
- 2 Error en Ejecución
- 3 Número de Parámetros Inválido
- 4 Sobran Parámetros, se Ignoran los Restantes.

```

5   Limite Fuera de Rango.
6   Ciclo de Actualización Fuera de Rango(1-5500)
7   Identificador de Tabla Incorrecto.
8   Identificador Incorrecto ( SI - NO )
9   " " " " " "
10  CRIT
11  SOFT
12  HARD
13  Tabla Ocupada
14  Semáforo Bloqueado
15  Tabla Inválida
16  Tabla No Existente
17  Doble Acceso
18  Campo Especificado Incorrecto o No Existente
19  Marca de Tiempo: F dd-mm-aa hh:mm
20  CONTROL DISTRIBUIDO U.A.C Sidetec
21  " " " " " "
22  Algoritmo No definido
23  Consulta: C <comando>, <ubicación>, <parámetro>,
    <valor>
24  Modifica : M tabla tipo #ranura/variable
    parámetro dato
25  Tarea No Existente
26  (disponible)
27  (disponible)
28  (disponible)
29  (disponible)
30  " " (borra error crítico)
31  " " (borra error hardware)
32  " " (borra error software)
33  Límites Incorrectos o Tabla Vacía
34  )
35  Código de Error Incorrecto o No Existente

```

gen:

Rutina general que fija el offset de despliegado y coordina la presentación y establecimiento de la Tabla de Configuración de Pantalla. También verifica si la tabla a desplegar requiere una sola ventana o no.

ident:

Al ser editado un dato en el "campo de identificación de la tabla" de la Tabla de Configuración de Pantalla, esta rutina investiga si se trata de una tabla del sistema. Si el dato no es reconocido, desplegará un mensaje de error y fijará el último dato establecido o el valor default. Esta rutina es común para el establecimiento de los identificadores de las dos ventanas.

inic_rx:

Esta rutina se ejecuta una sola vez, luego de haber sido arrancada la tarea de Excepción. Su función es la inicialización de las estructuras globales.

inter:

Recibe el buffer editado por el usuario y lee el primer caracter del mismo. Si se trata de:

- C - Llama a la rutina "llacons" para validar la sintaxis del buffer recibido.
- L - Llama a la rutina de limpieza de pantalla, despliega mensaje de Control Distribuido y prompt.
- T - Llama a la rutina "tiempo" para fija la fecha y la hora del sistema.
- V - Llama a la rutina "yen" para desplegar la Tabla de Configuración de Pantalla.

límite:

Rutina que permite la especificación de los límites de desplegado para cualquier tabla excepto ranuras.

llacons:

El comando que es introducido a través de la consola debe tener el formato adecuado para que esta rutina realice una validación de sintaxis. Posteriormente envía el buffer validado al Coordinador a través de la rutina "Camb_for" para su procesamiento.

tiempo:

El buffer recibido por esta rutina es verificado en cada uno de sus caracteres y posteriormente enviado al sistema operativo para que este realice el establecimiento de la fecha y la hora.

valini:

Accesa al arreglo de estructuras de la Tabla de Configuración de Pantalla y despliega la información que ahí se encuentra.

wprintf:

Despliega un buffer de 320 caracteres que puede ser accedido por cualquier usuario.

5.3 Módulos Constitutivos - Transmisión

Su función principal es desplegar la información contenida en la base de datos del sistema .

La tarea de Transmisión es activada una sola vez y estará activa permanentemente. Durante su ejecución, como ya se dijo, se inicializan las variables, se leen los registros de la Tabla de Identificadores y se efectúa el despliegue de la tabla solicitada en Ventana 1 y/o Ventana 2, dependiendo de los registros actuales.

- adquis:** Despliega la Tabla de Adquisición.
Despliega el contenido de las tablas de adquisición después de haber obtenido las direcciones y tamaños de las tablas de cada uno de los procesadores.
- adq_ptr:** Despliega los apuntadores de Adquisición.
Despliega los apuntadores de adquisición en formato hexadecimal luego de haber conseguido los apuntadores correspondientes.
- clrscr:** Limpia la pantalla.
A través de esta rutina se invoca una utilería con las que cuentan las terminales tipo VT-100 o las secuencias ANSI de las computadoras personales cuando se utilizan como terminales.
- cons_tx:** Tarea de Transmisión.
Tarea arrancada por el sistema operativo que permanece activa indefinidamente y que realiza la llamada a las rutinas que despliegan las tablas de la base de datos cada vez que se cumple el ciclo de actualización fijado por el usuario. También mantiene actualizada la fecha y la hora en la pantalla.
- cursor:** Posiciona el cursor.
Rutina de uso general que posiciona el cursor en la línea y columna recibidos como parámetros.
- desp_alm:** Despliega la Palabra de Alarma y Modo.
Rutina utilizada por el programa "ranuras" y su función es la de abreviar el procedimiento para desplegar la Palabra de Alarma y Modo en formato binario.
- desp_conf:** Despliega la palabra de Configuración.
Rutina utilizada por el programa "ranuras" y su función es la de abreviar el procedimiento para desplegar la Palabra de Configuración en formato binario.
- desp_ind:** Despliega Indicador de Servicio.
Rutina utilizada por el programa "ranuras" y su función es la de abreviar el procedimiento para desplegar el Indicador de Servicio en formato binario.
- desp_mem:** Despliega Memoria.
Despliega la zona de memoria especificada por el usuario y despliega el contenido de ésta en formato hexadecimal y en ASCII.
- desp_stack:** Despliega Stack.
Despliega el Área de stack de una tarea especificada por el usuario y la despliega en formato hexadecimal y ASCII.

- dsp_apbd:** Despliega la tabla Apuntadores a la Base de Datos. Esta rutina presenta en la pantalla los apuntadores de la estructura de control de la base de datos.
- dsp_cbds:** Despliega la Tabla Control de Base de Datos. Presenta en la pantalla la estructura de control de la base de datos.
- dtrprint:** Despliega Tabla de Mensajes de Usuario. Despliega el buffer que emplea la rutina "wprintf" para almacenar los mensajes que otras rutinas le han enviado.
- encab:** Despliega el encabezado de la Tabla de Ranuras. Despliega el estado, el número de algoritmo el indicador de servicio y la salida calculada de la Tabla de ranuras.
- inic_tx:** Inicializa registros y variables de la tarea o tx. Cuando esta rutina es llamada por "cons_tx" inicializa los registros y arreglos globales de la sección de Transmisión.
- ini_tab:** Inicializa la Tabla de Rótulos. Inicializa la tabla que contiene los rótulos de:
Respuesta a petición de tabla,
Status del algoritmo,
Status del algoritmo (abreviado),
Tipo de variable,
Modo de Operación,
Nombre del algoritmo,
Estado de Relfislog,
Tipo de Tarjeta relfislog,
Tipo de variable relfislog,
Estado de la Tabla tipo de tarjetas.
- paro:** Despliega la Tabla de Paro de Emergencia. Despliega los indicadores de Emergencia 1 y 2.
- pdef_san1:** Despliega la Tabla de Definición de Entradas Analógicas. Obtiene la dirección y el tamaño de la tabla, verifica los límites de desplegado y despliega la sección solicitada.
- pdef_edig:** Despliega la Tabla de Definición de Entradas Digitales. Obtiene la dirección y el tamaño de la tabla, verifica los límites de desplegado y despliega la sección solicitada.
- pdef_san2:** Despliega la tabla de Definición de Salidas Analógicas.

Obtiene la dirección y el tamaño de la tabla, verifica los límites de desplegado y despliega la sección solicitada.

pdef_sdigs: Despliega la Tabla de Definición de Salidas Digitales.

Obtiene la dirección y el tamaño de la tabla, verifica los límites de desplegado y despliega la sección solicitada.

pprintf: Despliega de depuración.

Despliega una cadena de caracteres en la ventana del usuario (renglones 22 y 23). Esta rutina espera tres parámetros: línea, columna y buffer de los cuales el primero indica que se utilizará el renglón 22 si es igual a 0 ó 23 si es distinto de 0. La columna puede tener valores entre 0 y 79, es decir lo largo de la pantalla. Por último el bufer contiene lo que se desea desplegar.

ranuras: Despliega la Tabla de Ranuras.

Despliega en la pantalla la ranura seleccionada de la Tabla de Ranuras. Contiene las estructuras de los siguientes algoritmos:

- Dummy,
- Pid,
- Pid Razón,
- Pid Banda,
- Estacion Analógica Manual / Automático,
- Pid Batch,
- Flujo de Masa,
- Selector,
- Sumador,
- Multiplicador,
- divisor,
- Estación Digital Manual / Automático,
- Contador de Eventos,
- Función de Tiempo,
- Función Lógica,
- Generador de Funciones,
- Compensador Adelanto - Atraso,
- Selector Alto / Bajo,
- Actuador,
- On - Off,
- Válvula Hidráulica.

ran_act: Despliega la información asociada a un algoritmo actuador.

ran_adat: Despliega la información asociada a un algoritmo Compensador Adelanto - Atraso.

ran_ap: despliega la Tabla de Apuntadores a la Tabla de Ranuras.

Valida el rango de los límites de despliegue y posteriormente accesa un arreglo público donde se encuentra esta información.

- ran_bandas:** Despliega la información asociada a un algoritmo Pid Banda.
- ran_batch:** Despliega la información asociada a un algoritmo Pid Batch.
- ran_cntev:** Despliega la información asociada a un algoritmo Contador de Eventos.
- ran_divisi:** Despliega la información asociada a un algoritmo Divisor.
- ran_eama:** Despliega la información asociada a un algoritmo Estación Analógica Manual/Automático.
- ran_edma:** Despliega la información asociada a un algoritmo Estación Digital Manual/Automático.
- ran_flim:** Despliega la información asociada a un algoritmo Flujo de Masa.
- ran_flogi:** Despliega la información asociada a un algoritmo Función Lógica.
- ran_ftmpo:** Despliega la información asociada a un algoritmo Función Tiempo.
- ran_genf:** Despliega la información asociada a un algoritmo Generador de Funciones.
- ran_onoff:** Despliega la información asociada a un algoritmo On - Off.
- ran_pid:** Despliega la información asociada a un algoritmo Pid.
- ran_prodi:** Despliega la información asociada a un algoritmo Multiplicador.
- ran_razón:** Despliega la información asociada a un algoritmo Pid-Razón.
- ran_selab:** Despliega la información asociada a un algoritmo Selector Alto - Bajo.
- ran_selec:** Despliega la información asociada a un algoritmo Selector.
- ran_suma:** Despliega la información asociada a un algoritmo Suma.

- ran_vhds:** Despliega la información asociada a un algoritmo Válvula Hidráulica.
- Refilog:** Despliega la Tabla Relfislog.
Permite seleccionar cualquier sección de la Tabla Relación física - Lógica. Como parámetro recibe, entre otros, un indicador de la tabla a desplegar:
Comunicaciones - Centinela y Controladoras,
Salidas Digitales,
Entradas Digitales,
Salidas Analógicas,
Entradas Analógicas.
- regadq:** Despliega la Tabla de Registros de Adquisición.
Despliega los registros de Adquisición empleados por las rutinas que llevan el mismo nombre.
- reloj:** Despliega la Fecha y la Hora.
Obtiene la fecha y la hora proporcionada por MTOS y la despliega en el campo de fecha - hora de la pantalla.
- res_coor:** Despliega la Respuesta del Coordinador.
Despliega y monitorea continuamente el buffer con la respuesta del coordinador cuando este responde a un comando de consulta.
- sec_mra:** Despliega la Tabla Secuenciación de Ranuras.
Despliega para cada Marco la información contenida en la Tabla de Secuenciación de Ranuras.
- sec_rna:** Despliega la Tabla de Secuenciación de Ranuras.
Despliega para cuatro marcos el contenido de la tabla de Secuenciación de Ranuras.
- tabdef:** Despliega la Tabla de Definición de Ranuras.
Despliega los números de ranuras configuradas para cada marco.
- tari:** Despliega la tabla de Tareas Activas.
Despliega la información de la estructura de identificación de tareas.
- terror:** Despliega la Tabla de Errores.
Despliega el contenido de la estructura tabla_errores donde se registran los errores detectados por otras tareas.
- tidetari:** Despliega la Tabla tipo de Tarjetas.
Lee el contenido de la variable pública "tdt" y despliega el contenido de esta empleando rotulos para el estado operacional y tipo de tarjeta.
- var_anl:** Despliega la Tabla de Variables Analógicas.

Rutina común para desplegar cualquiera de las tres secciones de la tabla. Al inicio de su ejecución obtiene las direcciones y tamaños para las tres secciones, posteriormente almacena en un arreglo temporal la información solicitada y finalmente despliega el contenido de la tabla.

var_ap: Despliega los apuntadores a la Tabla de Variables. Despliega los apuntadores a las tablas de Variables Analógicas, Variables Digitales y Paro de Emergencia.

var_a_mi: Despliega la tabla de Variables a mostrar. Despliega el contenido de la variable pública "var_mues" utilizada por las rutinas de Adquisición.

var_digi: Despliega la Tabla de Variables Digitales. Rutina común para desplegar cualquiera de las tres secciones de la tabla. Al inicio de su ejecución obtiene las direcciones y tamaños para las tres secciones, posteriormente almacena en un arreglo temporal la información solicitada y finalmente despliega el contenido de la tabla.

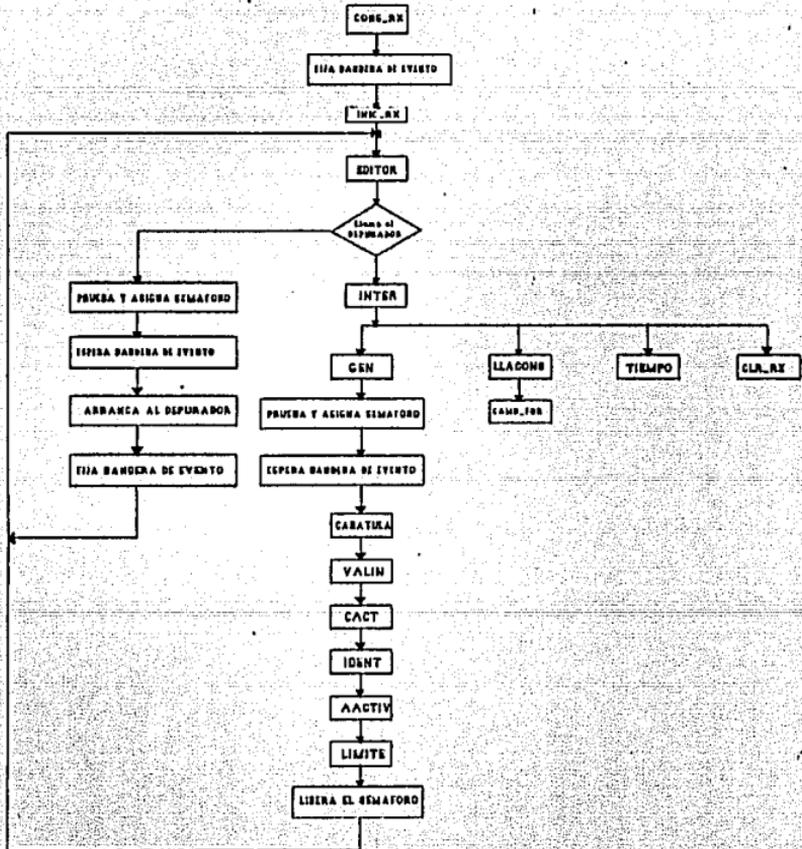


Diagrama de flujo de la Beccion de Captura

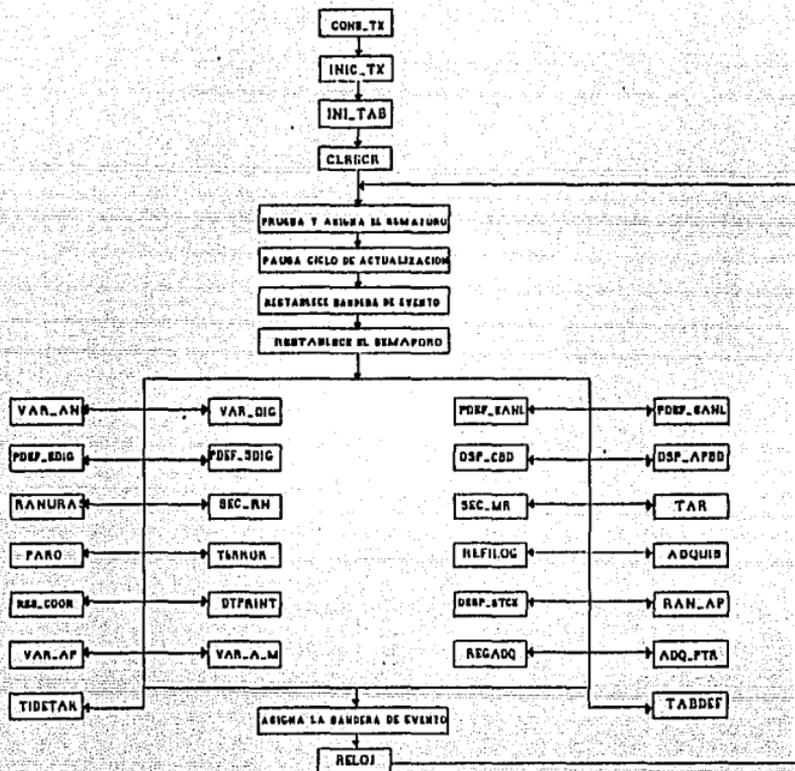


Diagrama de Flujo de la Sección de Despliegue

Conclusiones

El desarrollo y crecimiento tanto en materia tecnológica como de población, ha obligado a tomar ciertas medidas tendientes a optimizar e incrementar la producción de bienes de consumo por un lado y garantizar la producción de satisfactores por el otro.

Este compromiso ha venido a revolucionar las técnicas de producción en una forma tal, que actualmente es necesario valerse de equipos que puedan cumplir con los requerimientos de producción, de una forma segura y económica.

De esta forma, es evidente que los sistemas de control representan una buena alternativa para la solución de estos problemas, en la medida en que proporcionan un alto grado de confiabilidad y además de que se adaptan a cualquier condición de proceso.

El sistema de control distribuido fue concebido para cubrir las necesidades de control en la industria nacional. El diseño y realización fueron efectuados en México por ingenieros mexicanos. Este sistema presenta varias ventajas con respecto a otros sistemas similares. Entre otras la estabilidad, tiempo de muestreo, adaptabilidad, mantenimiento e instrucciones en español.

Si bien es cierto que se tiene una dependencia tecnológica muy fuerte, también lo es el hecho de que se tienen recursos para emprender y concluir exitosamente un proyecto de esta naturaleza. Actualmente, este sistema se encuentra en una fase de optimización, en la que se le darán otras facilidades y se mejorarán las ya existentes, con el fin de mantenerlo competitivo en el mercado internacional.

Bibliografía

**Distributed Micro/Minicomputer systems.
Structure, Implementation and Application.**
Cay Veitzman.

Process Control Instrumentation Technology
Curtis D. Johnson

**The Logical Design of Multiple Microprocessor
Systems.**
B.A. Bowen and R.J.A. Ruhr

Real Time Computer Control
S. Bennett and D.A. Linkens

The C Programming Language
Brian W. Kernighan - Dennis N. Ritchie

**MTOS-86. Real Time Multitasking Operating System
(Guía de Usuario).** Industrial Programming Inc.

MCS-86. Assembly Language Reference Manual
Intel Corporation.

**C Asynch Manager. Asynchronous Support Tools for
C Programmers.**
Blaise Computing Inc.

Microsoft Codeview Window Oriented Debugger
Microsoft Corporation.

**Manual de usuario del compilador Lattice para
MS-DOS.**
Lattice Inc.

Manual de usuario del compilador Aztec C 86
Manx Software Inc. Systems

A P E N D I C E I

Fuentes de algunas rutinas de la Sección de Captura

```
1 /#####
2 #
3 # 1. Nombre de Procedimiento.
4 #     cons_rx.c - Coordinador de la captura de datos del Manejador
5 #     de Consola.
6 #
7 # 2. forma de Llamada.
8 #     cons_rx()
9 #
10 # 3. Descripcion Funcional.
11 #     Tarea arrancada por el sistema operativo que
12 #     permanece activa indefinidamente y que controla
13 #     el ciclo de la aceptacion de caracteres en el campo
14 #     de captura de comandos.
15 #
16 # 4. Teoria de Operacion.
17 #     Luego de haber sido iniciada, esta tarea llama a la
18 #     rutina inic_rx y entra en un ciclo permanente en
19 #     donde llama a las rutinas: editor e inter. Mismas
20 #     que se encargaran de entrar e interpretar la infor-
21 #     macion respectivamente. A traves de esta rutina se
22 #     llama al depurador del sistema al encontrarse activa
23 #     la clave correspondiente.
24 #
25 # 5. Respuesta.
26 #
27 # 6. Limitaciones.
28 #
29 # 7. Historia de modificaciones.
30 #     F. Perez C. 31/03/88 Creacion
31 #
32 # 8. MDTA: Copyright (C) 1988 SIBTEC S.A.
33 #
34 #####
35 //
36 //
37
38 #include "atos.h"
39 #include "semafors.h"
40 #include "rutcons.h"
41
42 #define TIPDI 1
43 #define ACTIVO 1
44
45 extern int cr_cnl_d;
46 extern TASK tarea_consrx;
47
48 TASK tarea_consrx;
49
50 void
51 cons_rx()
52 {
53     char buffer[128+1];
54     int langbuf;
55     int stty;
```

```
56 1
57 1      /* Asigna el Grupo de Banderas de Eventos dado */
58 1      /* SEF(grp,flags,&sts) */;
59 1      SEF(EFLG_CONS,EFMK_CONS,&sts);
60 1
61 1      inic_rx();
62 1
63 1      for ( ; )
64 1      {
65 2          mont_bs( 0x04 );          /* fija bit 4H */
66 2
67 2          longbuf = sizeof(buffer);
68 2          editor( buffer,&longbuf );
69 2          if ( cr_cntl_d != ACTIVO )
70 2              inter( buffer,longbuf );
71 2          #1 == if ( cr_cntl_d == ACTIVO )          /* debugger activo? */
72 3
73 3
74 3
75 3          mont_br( 0x04 );          /* restablece bit 4H */
76 3          /* Prueba y Asigna el Semaforo */
77 3          /* TSF(sflog,&sts) */;
78 3          TSF(SMF_PTYL ,&sts );
79 3          mont_bs( 0x04 );          /* fija bit 4H */
80 3
81 3          cr_cntl_d = 0;
82 3
83 3
84 3          mont_br( 0x04 );          /* restablece bit 4H */
85 3          /* Bloquea la Tarea hasta que se cumplan las Banderas de Evento dadas */
86 3          /* WEF(typ,grp,flags,&sts) */;
87 3          WEF(TIPOI,EFLG_CONS,EFMK_CONS,&sts);
88 3          mont_bs( 0x04 );          /* fija bit 4H */
89 3
90 3          clr_rx();
91 3
92 3          mont_br( 0x04 );          /* restablece bit 4H */
93 3          /* Reinicia la Tarea especifica si esta Durmiente */
94 3          /* debugger */
95 3          /* SIRT(tcord,task,prts,priv,p1,p2,&sts) */;
96 3          SIRT( WTEWD, 1, 3, 0, 0, 0, &sts );
97 3          mont_bs( 0x04 );          /* fija bit 4H */
98 3
99 3          clr_rx();
100 3          errores(20);          /* mje de entrada. */
101 3          errores(34);          /* limpia campo de mjes */
102 3
103 3          /* Suella el Semaforo especificado */
104 3          /* RSF(sflog,&sts) */;
105 3          RSF(SMF_PTYL ,&sts );
106 2      }
107 1
108 1      wprintf("cons_rx");
109 1      errores(4);          /* Se abandono la ejecucion de una tarea. */
110 1
```

```
111 | | /* Termina esta Tarea incondicionalmente */  
112 | | /* ENDT() */  
113 | | ENDT();  
114 | | )  
115 |  
116 |  
117 |
```

```
1 //#####
2 |
3 | 1. Nombre de Procedimiento.
4 | editor.c - Controla la aceptacion de los caracteres teclados.
5 |
6 | 2. Forma de Llamada:
7 | char tbuffer;
8 | int tlongbuf;
9 | editor( buffer,longbuf )
10 |
11 | 3. Descripcion funcional.
12 | "Editor" es la rutina que se encarga de la aceptacion
13 | de los caracteres teclados por el usuario que propor-
14 | ciona varias facilidades de edicion. Los buffers gene-
15 | rados mediante esta rutina son analizados posterior-
16 | mente por una rutina interprete.
17 |
18 | 4. Teoria de Operacion.
19 | Editor acepta los siguientes caracteres:
20 | O - 9, A - Z, a - z, "A", "C", "D", "E",
21 | BS, LF, "K", CR, "Q", "S", "M", "Z", ESC,
22 | "!", "!", "!", "!", "!", "!", "!", "!", "!", "!",
23 | "!", "!", "!", "!", "!", "!", "!", "!", "!", "!",
24 | "!", "!", "!", "!", "!", "!", "!", "!", "!", "!",
25 |
26 | Proporciona los comandos de edicion:
27 | Espacio hacia atras
28 | Aumento de linea - Retorno de carro
29 | Supresion de caracter
30 | Recuerda buffer anterior
31 | Recuerda buffer siguiente
32 | Cursor a la izquierda
33 | Cursor a la derecha
34 | Origen
35 | Final
36 | Cancelacion
37 | Almacena caracter ( crea buffer ).
38 |
39 | 5. Respuesta.
40 |
41 | 6. Limitaciones.
42 |
43 | 7. Historia de modificaciones.
44 | F. Perez C. 31/03/88 Creacion
45 |
46 | 8. MDTA: Copyright (C) 1988 SIBETEC S.A.
47 |
48 | //#####
49 | //
50 |
51 | #include "rutcons.h"
52 |
53 | #define ACTIVO 1
54 | #define A2T 1
55 | #define COL 1
```

```
56 #define TAMA 59
57
58 static int nivel_ent; /* ind. entrada la lista de comandos */
59 static int nivel_sal; /* ind. salida la lista de comandos */
60
61
62 | struct prof(
63 |     char  bx[128+1];
64 |     int   cuenta;
65 | );
66
67 static struct prof guarda[10];
68
69 void
70 inic_editor()
71 | {
72 |     int i;
73 |     for (i = 0; i < 10; i++)
74 |     | {
75 |         guarda[i].bx[0] = '\0';
76 |         guarda[i].cuenta = 0;
77 |     }
78 |     nivel_ent = 0;
79 |     nivel_sal = 0;
80 | }
81
82 int cr_ap_actual; /* Apuntador de posicion actual */
83 int ap1; /* Apuntador auxiliar para copias */
84 int ap_final; /* Apuntador al final del buffer */
85 char cr_buf[128+1];
86
87
88 int cond;
89 char c[4,4];
90 char lecla;
91 extern int cr_cntl_d;
92 extern short id_compil;
93
94
95
96 void
97 editor(buffer, longbuf)
98 char *buffer;
99 int *longbuf;
100 | {
101 |     short Ren0;
102 |
103 |     /*----- Obtiene los valores de los renglones -----*/
104 |
105 |     if ( id_compil == AZT )
106 |         Ren0 = 1;
107 |     else
108 |         Ren0 = 0;
109 |
110 |     /*----- Inicializacion -----*/
```

```
111 1
112 1      cr_buf[ 0 ] = '\0';
113 1      cr_ap_actual = 0;
114 1      ap_final = 0;
115 1      apt = 0;
116 1
117 1      do
118 2          |
119 2          |   cond = 0;           /* bandera para permitir caracteres */
120 2          |   do
121 3          |   |
122 3          |   |   ----- Entrada de Caracteres -----
123 3          |   |   -----
124 3          |   |
125 3          |   |       mont_bs( 0x10 );           /* fija bit 4H */
126 3          |   |       mont_br( 0x04 );           /* restablece bit 4H */
127 3          |   |       c = getch();
128 3          |   |       mont_bs( 0x04 );           /* fija bit 4H */
129 3          |   |       mont_br( 0x10 );           /* restablece bit 4H */
130 3          |   |
131 3          |   |       if ( c )= '0' && c != '9' )
132 3          |   |           cond = 1;
133 3          |   |       else if ( c )= 'A' && c != 'Z' )
134 3          |   |           cond = 1;
135 3          |   |       else if ( c )= 'a' && c != 'z' )
136 3          |   |           cond = 1;
137 3          |   |       else if ( c == 1 || c == 3 || c == 4 || c == 5 )
138 3          |   |           cond = 1; /* At(izquierda), C(escape), D(delete), E(fin) */
139 3          |   |       else if ( c == 8 || c == 10 || c == 11 || c == 13 )
140 3          |   |           cond = 1; /* BS(esp.atras), LF(linea), K(cancela), CR(retorno) */
141 3          |   |       else if ( c == 17 || c == 19 || c == 23 || c == 26 || c == 27 )
142 3          |   |           cond = 1; /* O(inicio), S(der.), W(buf.ant), Z(buf.sigto), EBC */
143 3          |   |       else if ( c == '?' || c == '+' || c == '-' || c == '=' )
144 3          |   |           cond = 1;
145 3          |   |       else if ( c == '\0' || c == 127 || c == 32 || c == 59 )
146 3          |   |           cond = 1; /* null, delete, space, '*' */
147 3          |   |
148 3          |   |   }
149 3          |   |   while ( cond == 0 );
150 3          |   |
151 2          |   |   if ( c == 27 )
152 2          |   |       |
153 2          |   |       |   d = getch();
154 2          |   |       |   if ( d == '?' )
155 2          |   |       |       |
156 2          |   |       |       |   e = getch();
157 2          |   |       |       |   switch ( e )
158 2          |   |       |       |   |
159 2          |   |       |       |   |   case 'A':
160 2          |   |       |       |   |       c = 26;
161 2          |   |       |       |   |       break;
162 2          |   |       |       |   |   case 'B':
163 2          |   |       |       |   |
164 2          |   |       |       |   |
165 2          |   |       |       |   |
```



```
1  /*****
2  *                               Ultima Modificacion es 11-4-88 1:15 pm
3  *
4  * 1. Nombre de Procedimiento.
5  *   inter.c - Interpreta el comando recibido.
6  *
7  * 2. Forma de Llamada.
8  *       char  b[];
9  *       int   len;
10 *       inter(b,len)
11 *
12 * 3. Descripcion Funcional.
13 *   Esta rutina es llamada por la rutina "cons_rx"
14 *   despues de haber sido editado algun buffer.
15 *   El buffer recibido por esta rutina es verificado
16 *   y posteriormente enviado a las rutinas correspon-
17 *   dientes dependiendo del comando especificado.
18 *
19 * 4. Teoria de Operacion.
20 *   "inter" recibe el comando tecleado por el usuario,
21 *   enseguida lee cada uno de los caracteres. Si se tra-
22 *   ta del caracter:
23 *
24 *       C - llama a la rutina llacons para validar
25 *           la sintaxis del buffer recibido.
26 *       L - llama a la rutina de limpieza de pantalla
27 *           y despliega mensaje de Sidetec y prompt.
28 *       M - llama a la rutina mod_bd para modificar
29 *           la Base de Datos.
30 *       T - llama a la rutina para fijar la fecha
31 *           y la hora del sistema.
32 *       V - llama a la rutina gen para desplegar la
33 *           Tabla de Configuracion de Pantalla.
34 *
35 * 5. Respuesta.
36 *
37 * 6. Limitaciones.
38 *
39 * 7. Historia de modificaciones.
40 *       F. Perez C. 31/03/88 Creacion
41 *
42 * 8. NOTA: Copyright (C) 1988 SIDETEC S.A.
43 *
44 *
45 */
46
47 #include "ntos.h"
48 #include "incl.h"
49 #include "rutcons.h"
50 /* #include "rutntos.h"
51 #include "rutbased.h"
52 #include "rutsopntos.h"
53 #include "rutntecjc.h" */
54 #include "dec_t_er.h"
55
```

```
56 extern ERR_FLAGS err_flags;
57 static char bftesi(60);
58
59 void
60 inter(b,lea)
61 char b[];
62 int len;
63
64 {
65     int ventana;
66     char dr(7);
67     char rd(3);
68     int ii;
69     long bf;
70     long atol();
71     int bi;
72     int atoi();
73     long dirc,lq;
74
75     /*----- Inicializacion -----*/
76
77     for (i = 0; i < 6; i++)
78         dr[i] = ' ';
79     dr[6] = '\0';
80
81     for (i = 0; i < 2; i++)
82         rd[i] = ' ';
83     rd[2] = '\0';
84
85     switch ( b[0] )
86     {
87
88     /*----- Rutina de Consulta -----*/
89
90         case 'c':
91         case 'C':
92             if (len > 1)
93             {
94                 strcat(b,"10");
95                 llacoss(b);
96             }
97             break;
98
99         /*----- Limpia la Pantalla -----*/
100
101         case '!':
102         case 'L':
103             clr_mx();
104             errores(20); /* Mensaje Sidetec */
105             errores(34); /* Prompt */
106             break;
107
108         /*----- Fija Fecha y Hora -----*/
109
110     }
```

```
111 2 | case '1':
112 2 |     case '1':
113 2 |         if (len > 1)
114 2 |             tiempo(b);
115 2 |         else if (len == 1)
116 2 |             errores(19);
117 2 |         break;
118 2 |
119 2 | //----- Configura Pantalla -----//
120 2 |
121 2 |     case 'y':
122 2 |     case 'Y':
123 2 |         errores(20);           /* Mensaje Sidetec */
124 2 |         errores(34);          /* Prompt */
125 2 |         ventana = (b[2] == '2') ? 2 : 1;
126 2 |         gen(ventana);
127 2 |         break;
128 2 |
129 2 | //----- Banderas de Errores -----//
130 2 | //----- Provisional -----//
131 2 |
132 2 |     case 'i':
133 2 |     case 'I':
134 2 |         errores(20);           /* Mensaje Sidetec */
135 2 |         errores(34);          /* Prompt */
136 3 |     switch ( b[1] )
137 3 |     {
138 3 |         case 'H':
139 3 |             err_flags.EH = 1;
140 3 |             break;
141 3 |         case 'h':
142 3 |             err_flags.EH = 0;
143 3 |             break;
144 3 |         case 'S':
145 3 |             err_flags.ES = 1;
146 3 |             break;
147 3 |         case 's':
148 3 |             err_flags.ES = 0;
149 3 |             break;
150 3 |         case 'C':
151 3 |             err_flags.IC = 1;
152 3 |             break;
153 3 |         case 'c':
154 3 |             err_flags.IC = 0;
155 3 |             break;
156 2 |     }
157 2 |     break;
158 2 | //-----//
159 2 |
160 2 |     default:
161 2 |         errores(20);           /* Mensaje Sidetec */
162 2 |         errores(34);          /* Prompt */
163 2 |         break;
164 2 |
165 1 | }
```

A P E N D I C E II

Fuentes de algunas rutinas de la Sección de Despliegue

```
56 #define TAMN 59
57
58 static int nivel_ent; /* ind. entrada la lista de comandos */
59 static int nivel_sal; /* ind. salida la lista de comandos */
60
61
62 | struct prof{
63 |     char _bx[128*1];
64 |     int cuenta;
65 | };
66
67 static struct prof guarda[10];
68
69 void
70 inic_editor()
71 | {
72 |     int i;
73 |     for (i = 0; i < 10; i++)
74 |     {
75 |         guarda[i].bx[0] = '\0';
76 |         guarda[i].cuenta = 0;
77 |     }
78 |     nivel_ent = 0;
79 |     nivel_sal = 0;
80 | }
81
82 int cr_ap_actual; /* Apuntador de posicion actual */
83 int apl; /* Apuntador auxiliar para inputs */
84 int ap_final; /* Apuntador al final del buffer */
85 char cr_buf[128*1];
86
87
88 int cond;
89 char c,d,e;
90 char tecla;
91 extern int cr_ent,d;
92 extern short id_comp;
93
94
95
96 void
97 editor(buffer, longbuf)
98 char *buffer;
99 int *longbuf;
100 | {
101 |     short Renov;
102 |
103 |     /*----- Obtiene los valores de los renques -----*/
104 |
105 |     if ( id_comp == AZI )
106 |         Renov = 1;
```

```
1  /*****
2  |
3  | 1. Nombre de Procedimiento.
4  |     cons_tx.c - Coordina las acciones del Manejador de Consola,
5  |     Seccion Transmission.
6  |
7  | 2. Forma de Llamada.
8  |     cons_tx()
9  |
10 | 3. Descripcion Funcional.
11 |     Tarea arrancada por el sistema operativo que
12 |     permanece activa indefinidamente y que realiza
13 |     la llamada a las rutinas que despliegan las
14 |     tablas de la base de datos cada vez que se cumple
15 |     el ciclo de actualizacion fijado por el usuario.
16 |
17 | 4. Teoria de Operacion.
18 |     Al iniciar la ejecucion de esta rutina, se iniciali-
19 |     zan los registros de la Seccion de Transmission y
20 |     enseguida se entra en un ciclo indefinido en el que
21 |     como primera accion se solicita el acceso exclusivo
22 |     a la Tabla de Configuracion de Pantalla, se registran
23 |     los datos fijados por el usuario, se libera la Tabla
24 |     de Configuracion y finalmente se llama a la rutina de
25 |     despliegue elegida.
26 |
27 | 5. Respuesta.
28 |
29 | 6. Limitaciones.
30 |
31 | 7. Historia de modificaciones.
32 |     F. Perez C. 01/04/88 Creacion
33 |
34 | 8. NOTAs Copyright (C) 1988 SIDETEC S.A.
35 |
36 | *****/
37 |
38 | /**/
39 |
40 | #include "mtos.h"
41 | #include "incl.h"
42 | #include "paco.h"
43 | #include "semaforo.h"
44 | #include "def_caps.h"
45 | #include "def_bdat.h"
46 | #include "base_dat.h"
47 | #include "base_cnt.h"
48 | #include "dec_t_er.h"
49 | #include "rutcotz.h"
50 |
51 | #define TIPO1 1
52 |
53 | extern CNT_PILL ci_cntpill;
54 | extern char ci_mensy[];
55 | extern ERR_FLAGS err_flags;
```

```
56
57 TASK tarea_consrx; // Manejador de Consola. Tarea de Salida //
58 TCP ct_rv[2]; // Registros de Ventana //
59 #define int ciclo; // ciclo de actualizacion //
60
61
62 void
63 cons_tx()
64 {
65     int i;
66     int s;
67
68
69     CNT_BASE %cnt_base;
70     obt_orgh(&cnt_base);
71
72
73     // ----- Inicializacion de Variables Globales ----- //
74
75     inic_tx(); // Rutina de Inicializacion. //
76     ini_tab(); // Inicializacion de mensajes. //
77     clrscr(); // Rutina de Limpieza de pantalla. //
78     errores(20); // Despliega Mensaje de Entrada. //
79     errores(34); // Limpia el campo de entrada //
80     ciclo = 10;
81
82     for ( ; ; )
83     {
84
85         // ----- fija el Ciclo de Actualizacion ----- //
86
87
88
89         if ( ciclo > 250 )
90         {
91             ciclo /= 100; // divide el ciclo de actualizacion //
92
93             mont_bs( 0x08 ); // fija bit 0H //
94             // Suspénde la Tarea por el Periodo de Tiempo dado //
95             // PAUSE(time,len,&sts) //
96             PAUSE( NSEC100, ciclo, &sts );
97             mont_br( 0x08 ); // fija bit 0H //
98
99         }
100         else
101         {
102
103             mont_bs( 0x08 ); // fija bit 0H //
104             PAUSE( NSEC1, ciclo, &sts );
105             mont_br( 0x08 ); // restablece el bit 0H //
106
107         }
108
109         mont_bs( 0x08 ); // fija bit 0H //
110         // Prueba y Asigna el Senaforo //
```

```
111 2      /* TSF(sflag,&sts) */  
112 2      TSF(SMF_PILL, &sts);  
113 2      snt_br( 0x0B ); /* restablece el bit BH */  
114 2  
115 2      ciclo = ct_cnpttl.actual;  
116 2  
117 2      if ( ct_cnpttl.in_can == 1 ) /* indicador de cambio activo? */  
118 3      {  
119 3  
120 3      /* ----- Obtiene los registros de la TCP ----- */  
121 3  
122 3      /* solicita acceso exclusivo a la T.C.P. */  
123 3  
124 3      ct_rv[0].cod = ct_cnpttl.vent[0].cod;  
125 3      ct_rv[0].aa  = ct_cnpttl.vent[0].aa;  
126 3      ct_rv[0].sec = ct_cnpttl.vent[0].sec;  
127 3      ct_rv[0].linf = ct_cnpttl.vent[0].linf;  
128 3      ct_rv[0].lsup = ct_cnpttl.vent[0].lsup;  
129 3      ct_rv[0].offset = ct_cnpttl.vent[0].offset;  
130 3  
131 3      ct_rv[1].cod = ct_cnpttl.vent[1].cod;  
132 3      ct_rv[1].aa  = ct_cnpttl.vent[1].aa;  
133 3      ct_rv[1].sec = ct_cnpttl.vent[1].sec;  
134 3      ct_rv[1].linf = ct_cnpttl.vent[1].linf;  
135 3      ct_rv[1].lsup = ct_cnpttl.vent[1].lsup;  
136 3      ct_rv[1].offset = ct_cnpttl.vent[1].offset;  
137 3  
138 3      ct_cnpttl.in_can = 0; /* restablece el indicador de cambio */  
139 3  
140 2  
141 2  
142 2  
143 2      /* Restaura un Grupo de Banderas de Evento */  
144 2      /* REF(grp,flags,&sts) */;  
145 2      REF(EFLG_CONS,EFM_CONS,&sts);  
146 2  
147 2  
148 2      /* Suelta el Semaforo especificado */  
149 2      /* RSF(sflag,&sts) */;  
150 2      RSF(SMF_PILL, &sts);  
151 2  
152 2  
153 2      /* ----- Despliega la tabla solicitada ----- */  
154 2  
155 2  
156 2      for ( i = 0; i < 2; i++ )  
157 3      {  
158 3          if ( ct_rv[i].aa != 0 )  
159 4          {  
160 4              switch ( ct_rv[i].cod )  
161 5              {  
162 5                  case ITAG_VA:  
163 5                      var_anl(ct_rv[i].cod,ct_rv[i].sec,ct_rv[i].linf,ct_rv[i].lsup,  
164 5                          ct_rv[i].offset);  
164 5                      break;
```

```
165 5 | | | | case iTAB_VU:
166 5 | | | |     var_dig(ct_rv[i].cod,ct_rv[i].sec,ct_rv[i].linf,ct_rv[i].lsup,
      | | | |     ct_rv[i].offset);
167 5 | | | |     break;
168 5 | | | | case iTAB_DA:
169 5 | | | |     if (ct_rv[i].sec == SEC_ENT)
170 5 | | | |         pdef_ganl(ct_rv[i].cod,ct_rv[i].sec,ct_rv[i].linf,ct_rv[i]
      | | | |         .lsup,ct_rv[i].offset);
171 5 | | | |     else if (ct_rv[i].sec == SEC_SAL)
172 5 | | | |         pdef_ganl(ct_rv[i].cod,ct_rv[i].sec,ct_rv[i].linf,ct_rv[i]
      | | | |         .lsup,ct_rv[i].offset);
173 5 | | | |     break;
174 5 | | | | case iTAB_DD:
175 5 | | | |     if (ct_rv[i].sec == SEC_ENT)
176 5 | | | |         pdef_rdig(ct_rv[i].cod,ct_rv[i].sec,ct_rv[i].linf,ct_rv[i]
      | | | |         .lsup,ct_rv[i].offset);
177 5 | | | |     else if (ct_rv[i].sec == SEC_SAL)
178 5 | | | |         pdef_rdig(ct_rv[i].cod,ct_rv[i].sec,ct_rv[i].linf,ct_rv[i]
      | | | |         .lsup,ct_rv[i].offset);
179 5 | | | |     break;
180 5 | | | | case iTAB_CBD:
181 5 | | | |     dsp_cbd ( cnt_base,ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].
      | | | |     offset);
182 5 | | | |     break;
183 5 | | | | case iTAB_APD:
184 5 | | | |     dsp_apbd ( cnt_base,ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].
      | | | |     offset);
185 5 | | | |     break;
186 5 | | | | case iTAB_RN:
187 5 | | | |     ranuras(ct_rv[i].linf,ct_rv[i].offset);
188 5 | | | |     break;
189 5 | | | | case iSEC_RAN:
190 5 | | | |     sec_rn ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset );
191 5 | | | |     break;
192 5 | | | | case iSEC_MR:
193 5 | | | |     sec_ar ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset,ct_rv[i].
      | | | |     sec );
194 5 | | | |     break;
195 5 | | | | case iTAR_ACT:
196 5 | | | |     tar ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset);
197 5 | | | |     break;
198 5 | | | | case iTAB_PE:
199 5 | | | |     paro (ct_rv[i].offset);
200 5 | | | |     break;
201 5 | | | | case iTAB_ERR:
202 5 | | | |     terror ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset);
203 5 | | | |     break;
204 5 | | | | case iTAB_RFL:
205 5 | | | |     retilog ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset,ct_rv[i]
      | | | |     .sec );
206 5 | | | |     break;
207 5 | | | | case iTAB_ADD:
208 5 | | | |     adquis ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset,ct_rv[i].
      | | | |     sec );
209 5 | | | |     break;
```

```
210 5 | | | | case iRES_COO:
211 5 | | | |     res_coor ( ct_rv[i].offset );
212 5 | | | |     break;
213 5 | | | | case iIAB_MUS:
214 5 | | | |     dtpri ( ct_rv[i].offset );
215 5 | | | |     break;
216 5 | | | | case iDES_MEN:
217 5 | | | |     desp_men ( ); //
218 5 | | | |     break;
219 5 | | | | case iDES_STK:
220 5 | | | |     desp_stck ( ct_rv[i].linf,lo,ct_rv[i].lsup );
221 5 | | | |     break;
222 5 | | | | case iRAN_API:
223 5 | | | |     ran_ap ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset);
224 5 | | | |     break;
225 5 | | | | case iVAR_API:
226 5 | | | |     var_ap ( ct_rv[i].offset );
227 5 | | | |     break;
228 5 | | | | case iADD_VAN:
229 5 | | | |     var_a ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset);
230 5 | | | |     break;
231 5 | | | | case iADD_RDA:
232 5 | | | |     reqdq ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset);
233 5 | | | |     break;
234 5 | | | | case iADD_PTR:
235 5 | | | |     adq_ptr ( ct_rv[i].offset );
236 5 | | | |     break;
237 5 | | | | case iTIP_TAR:
238 5 | | | |     tldetar ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset );
239 5 | | | |     break;
240 5 | | | | case iDEF_MAR:
241 5 | | | |     labdel ( ct_rv[i].linf,ct_rv[i].lsup,ct_rv[i].offset );
242 5 | | | |     break;
243 5 | | | | default:
244 5 | | | |     errores(7); // id. de tabla incorrecto. //
245 5 | | | |     vprintf( ct_mensj, " i %02.2d valor i %03.3d%0",i,ct_rv[i].
246 5 | | | |     cod);
247 5 | | | |     wprintf( ct_mensj );
248 4 | | | |     break;
249 3 | | | |
250 2 | | | |
251 2 | | | |
252 2 | | | |
253 2 | | | | // Asigna el Grupo de Banderas de Evento dado //
254 2 | | | | // SIF(grp,flags,dsts) //;
255 2 | | | | SIF(EFLG_CONS,EFMK_CONS,dsts);
256 2 | | | |
257 2 | | | | rvdj();
258 2 | | | |
259 2 | | | | if (err_flags.ES == 1)
260 2 | | | |     errores(11);
261 2 | | | | else
262 2 | | | |     errores(31);
263 2 | | | | if (err_flags.EH == 1)
```

```
264 2 |         errores(12);  
265 2 |         *l =  
266 2 |         errores(32);  
267 2 |         if (err_flags.EC == 1)  
268 2 |             errores(10);  
269 2 |         *l =  
270 2 |         errores(30);  
271 1 |     }  
272 1 |     wprintf("cons_tx");  
273 1 |     errores(4); // Se Abandono la ejecucion de la tarea //  
274 1 |  
275 1 |     // Termina esta Tarea Incondicionalmente //  
276 1 |     // ENDT() //;  
277 1 |     ENDT();  
278 1 | }  
279
```

```
1 /*****
2 *
3 * 1. Nombre de Procedimiento.
4 *     ranuras.c - Despliega la Tabla de Ranuras.
5 *
6 * 2. Forma de Llamada.
7 *
8 *     int ran;
9 *     int off;
10 *     ranuras(ran,off)
11 *
12 * 3. Descripcion Funcional.
13 *     Esta rutina despliega en la pantalla la ranura
14 *     seleccionada de la Tabla de Ranuras. Contiene las
15 *     estructuras de los siguientes algoritmos:
16 *
17 *     00 - dummy          11 - estd_ma
18 *     01 - pid           12 - cut_evn
19 *     02 - pid_razon     13 - fcn_tiempo
20 *     03 - pid_banda    14 - fcn_log
21 *     04 - esta_ma      15 - gen_func
22 *     05 - pid_batch    16 - cap_adj_str
23 *     06 - flj_ms       17 - selec_ab
24 *     07 - selector     18 - act
25 *     08 - suma         19 - on_off
26 *     09 - producto     20 - viv_hdr
27 *     10 - divisor
28 *
29 * 4. Teoria de Operacion.
30 *     Obtiene la direccion y el tamaño de la tabla de
31 *     ranuras, verifica que la tabla no este vacía,
32 *     despliega el indicador de servicio y la frecuencia
33 *     de activacion, discrimina el tipo de algoritmo
34 *     configurado y despliega la tabla correspondiente.
35 *
36 * 5. Respuesta.
37 *
38 * 6. Limitaciones.
39 *
40 * 7. Historia de modificaciones.
41 *     F. Perez C. 01/04/88 Creacion
42 *
43 * 8. NOTAs: Copyright (C) 1988 SIDETEC S.A.
44 *
45 */
46 //
47
48 #include "ntos.h"
49 #include "incl.h"
50 #include "def_caps.h"
51 #include "def_idfa.h"
52 #include "def_bdat.h"
53 #include "base_dal.h"
54 #include "base_cnt.h"
55 #include "rutcols.h"
```

```
56
57 extern codg ct_as[];
58 extern codg ct_of[];
59 extern char ct_mensj[];
60
61
62 void
63 ranuras(ran,off)
64 int ran;
65 int off;
66 {
67     int dia[3];
68     char *dirtab[3];
69     unsigned bits;
70     STR_ALG *str_alg;
71     CHI_BASE *ap_in_base;
72     obt_ergtb(4*ap_in_base);
73
74
75     dirtab[SEC_ENT] = (char *) ap_in_base -> rn_tabla[ran];
76
77     dia[SEC_ENT] = ap_in_base -> rn_num_ranuras;
78
79     if (ran < dia[SEC_ENT])
80     {
81         if (dia[0] != 0)
82         {
83             vprintf(ct_mensj, "%c(%02.2d;%02.2dH TABLA DE RANURAS . Rnr i %3.3d",
84                 ESC,3,off,ran);
85             aprntf( ct_mensj );
86
87             str_alg = (STR_ALG *)dirtab[0];
88
89             bits = str_alg -> ind_sq;
90             desp_ind(6,off,bits);
91
92             vprintf(ct_mensj, "%c(%02.2d;%02.2dH F%04.2f",
93                 ESC,6,off+28,
94                 str_alg -> frc_act);
95             aprntf( ct_mensj );
96
97             switch ( str_alg -> alg )
98             {
99
100              /*-----*/
101              case ALG_DUMMY:
102                  vprintf(ct_mensj, "%c(%02.2d;%02.2dH Algoritmo DUMMY10",
103                      ESC,4,off);
104                  aprntf( ct_mensj );
105                  break;
106              /*-----*/
107              case ALG_PID:
108                  ran_pid(str_alg,off);
109                  break;
110
```

```
111 4 | | | |----- Algoritmo Pid Razon -----|/
112 4 | | | |
113 4 | | | | case ALG_PIDRAZON:
114 4 | | | |   ran_razon(str_alg,off);
115 4 | | | |   break;
116 4 | | | |
117 4 | | | |----- Algoritmo Pid Banda -----|/
118 4 | | | | case ALG_PIDBANDA:
119 4 | | | |   ran_banda(str_alg,off);
120 4 | | | |   break;
121 4 | | | |
122 4 | | | |----- Algoritmo Estacion Analogica N/A -----|/
123 4 | | | | case ALG_EAMA:
124 4 | | | |   ran_eama(str_alg,off);
125 4 | | | |   break;
126 4 | | | |
127 4 | | | |----- Algoritmo Pid Batch -----|/
128 4 | | | | case ALG_PIDBATCH:
129 4 | | | |   ran_batch(str_alg,off);
130 4 | | | |   break;
131 4 | | | |
132 4 | | | |----- Algoritmo Flujo de Masa -----|/
133 4 | | | | case ALG_FLJM:
134 4 | | | |   ran_flj(str_alg,off);
135 4 | | | |   break;
136 4 | | | |
137 4 | | | |----- Algoritmo Selector -----|/
138 4 | | | | case ALG_SELECTOR:
139 4 | | | |   ran_selec(str_alg,off);
140 4 | | | |   break;
141 4 | | | |
142 4 | | | |----- Algoritmo suma -----|/
143 4 | | | | case ALG_SUMA:
144 4 | | | |   ran_suma(str_alg,off);
145 4 | | | |   break;
146 4 | | | |
147 4 | | | |----- Algoritmo Producto -----|/
148 4 | | | | case ALG_PRODUCTO:
149 4 | | | |   ran_prod(str_alg,off);
150 4 | | | |   break;
151 4 | | | |
152 4 | | | |----- Algoritmo Divisor -----|/
153 4 | | | | case ALG_DIVISOR:
154 4 | | | |   ran_divis(str_alg,off);
155 4 | | | |   break;
156 4 | | | |
157 4 | | | |----- Algoritmo Estacion Digital N/A -----|/
158 4 | | | | case ALG_EDMA:
159 4 | | | |   ran_edma(str_alg,off);
160 4 | | | |   break;
161 4 | | | |
162 4 | | | |----- Algoritmo Contador de Eventos -----|/
163 4 | | | | case ALG_CNTEVN:
164 4 | | | |   ran_cnlev(str_alg,off);
165 4 | | | |   break;
```

```

166 4 | | | |
167 4 | | | |----- Algoritmo Funcion de Tiempo -----|/
168 4 | | | |     case ALG_FUNCIEMPO :
169 4 | | | |         ran_fiempo(str_alg,off);
170 4 | | | |         break;
171 4 | | | |
172 4 | | | |----- Algoritmo Funcion Logica -----|/
173 4 | | | |     case ALG_FUNCLOG :
174 4 | | | |         ran_flog(str_alg,off);
175 4 | | | |         break;
176 4 | | | |
177 4 | | | |----- Algoritmo Generador de Funciones -----|/
178 4 | | | |     case ALG_GENF :
179 4 | | | |         ran_genf(str_alg,off);
180 4 | | | |         break;
181 4 | | | |
182 4 | | | |----- Algoritmo Comparador Adelanto / Atraso -----|/
183 4 | | | |     case ALG_CMP_AA :
184 4 | | | |         ran_adat(str_alg,off);
185 4 | | | |         break;
186 4 | | | |
187 4 | | | |----- Algoritmo Selector A/B -----|/
188 4 | | | |     case ALG_SELAB :
189 4 | | | |         ran_selab(str_alg,off);
190 4 | | | |         break;
191 4 | | | |
192 4 | | | |----- Algoritmo Actuador -----|/
193 4 | | | |     case ALG_ACT :
194 4 | | | |         ran_actuador(str_alg,off);
195 4 | | | |         break;
196 4 | | | |
197 4 | | | |----- Algoritmo ON/OFF -----|/
198 4 | | | |     case ALG_ON_OFF :
199 4 | | | |         ran_onoff(str_alg,off);
200 4 | | | |         break;
201 4 | | | |
202 4 | | | |----- Algoritmo Valvula Hidraulica -----|/
203 4 | | | |     case ALG_VLV_HDR :
204 4 | | | |         ran_vhdr(str_alg,off);
205 4 | | | |         break;
206 4 | | | |
207 4 | | | |----- Algoritmo No definido -----|/
208 4 | | | |     default :
209 4 | | | |         errores(22);
210 4 | | | |         break;
211 3 | | | |
212 2 | | | |
213 1 | | | |
214 | | | |
    
```

A P E N D I C E III

Ejemplos de despliegues

22-JUL-1987 08:01

> <

TABLA DE REGISTROS DE ADQUISICION					
Num	Iac	VdC	VdN	Num	Iac
000	12	000	000	000	00
001	12	000	000	000	00
002	12	000	000	000	00
003	12	000	000	000	00
004	12	000	000	000	00
005	12	000	000	000	00
006	12	000	000	000	00
007	12	000	000	000	00
008	12	000	000	000	00
009	12	000	000	000	00
010	12	000	000	000	00
011	12	000	000	000	00
012	12	000	000	000	00
013	12	000	000	000	00
014	12	000	000	000	00
-1	ff	fff	000	000	00
000	00	000	000	000	00

APUNTADORES DE ADQUISICION

AD_FIFO : d000b000

DIR_CONT_ANA[0] : d000b000

DIR_CONT_ANA[1] : 0

DIR_CONT_ANA[2] : 0

DIR_CONT_ANA[3] : 0

Ejemplo de Despliegue de Tabla de Registros de Adquisicion
y apuntadores de Adquisicion.

22-JUL-1987 07:57

APUNTADORES A LA TABLA DE VARIABLES

Apuntador Ncc

VARIABLES ANALOGICAS

Entrada 547e98c0 016

Salida 547e9540 016

Calculada 547e9e40 000

VARIABLES DIGITALES

Entrada 547e85b4 004

Salida 547e8528 004

Calculada 547e7bce 000

PARO DE EMERGENCIA

Entrada 50d410b6

TABLA DE TAREAS ACTIVAS.Linea : 00-12

Nombre	Ident	Inic	STACK	Fin	Ucp
INT_COMS	012	a	6324ffdc	6324f80c	00
SECN_MRC	014	e	6324f80c	6324f30c	00
SECN_RAN	016	10	6324f30c	6324eccc	00
EJEC_RAN	018	12	6324eccc	6324dc8c	00
ADQ	020	14	6324dc8c	6324d4bc	00
COMUNICA	022	16	6324d4bc	6324ccec	00
COORDINA	024	18	6324ccec	6324c51c	00
INT_LUCI	026	1a	6324c51c	6324b96c	00
CREA_ADO	028	1c	6324b96c	6324b5bc	00
CONT_LFP	029	1e	6324b5bc	6324b57c	00
CONS_PIL	032	20	6324b57c	6324adac	00
CONS_La	034	22	6324adac	6324a6ad	00
DIAGNOST	036	24	6324a6ad	6324a26c	00

Ejemplo de Despliegue de Tabla de Apuntadores a la Tabla de Variables
y Tabla de Tareas Activas.

> <

22-JUL-1987 07:55

CONTROL DE BASE DE DATOS.

APUNTADORES A LA BASE DE DATOS.

Status	Gral	Libre				Status	Gral	Libre	Dir: 50d43632		
Status	Semf	Usur	Tabla	Ent	Sal	Cal	Status	Tbl	Entrada	Salida	Calculada
Libre	1	18	VAR_ANL	16	16	0	Tab Op	ANL	547e98c0	547e9540	547e8640
Libre	2	18	VAR_DIG	4	4	0	Libre	DIG	547e85b4	547e8528	547e7bc8
Libre	3	91	DEF_ANL	16	16		Libre	DAN	547e6e38	547e6c78	
Libre	4	24	DEF_DIG	4	4		Libre	DDG	547e6c08	547e6b7c	
Tab Op	5	18	RANURAS	0	RT:	240	Libre	RAN	2000c296	#Ran:	0
Libre	11	95	REL_FLG	41			Libre	RFL	50d42a7a		
Libre	12	93	ADQ_PRO	40			Libre	AP0	50d42412		
Libre	13	24	ADQ_PR1	40			Libre	AP1	50d41daa		
Libre	14	24	ADQ_PR2	40			Libre	AP2	50d41742		
Libre	15	24	ADQ_PR3	40			Libre	AP3	50d410da		
Libre	16	18	PAR_EMG	1			Libre	IPE	50d410b6		

RD 0 / RC 0 / RT 240

Ejemplo de despliegue de Tabla Control de Base de Datos y tabla de Apuntadores a la Base de Datos.

> <

22-JUL-1987 07:52

DEF. ENTRADAS ANALOGICAS.Linea: 00-07							DEF. SALIDAS ANALOGICAS.Linea: 00-16				
Status	T	UFis	R_SN	R_IN	GCo	Sn	Aa	Status	Tipo	Ubicacion	Ind_S
Libre	E	00000	00.00	00.00	00	00	0	Libre	S	00000	0000
En Ser	E	00702	00.90	00.10	00	01	1	En Ser	S	00500	0000
En Ser	E	00703	00.90	00.10	00	01	1	En Ser	S	00500	0000
En Ser	E	00700	00.90	00.10	00	01	1	En Ser	S	00500	0000
En Ser	E	00701	00.90	00.10	00	01	1	En Ser	S	00500	0000
En Ser	E	00702	00.90	00.10	00	01	1	En Ser	S	00500	0000
En Ser	E	00703	00.90	00.10	00	01	1	En Ser	S	00500	0000
En Ser	E	00700	00.90	00.10	00	01	1	En Ser	S	00501	0000
LCsup	LPCsu	LCinf	LPCin	LDasc	LDdes			En Ser	S	00502	0000
0.000	0.000	0.000	0.000	0.000	0.000			En Ser	S	00503	0000
0.700	0.600	0.100	0.310	0.200	0.200			En Ser	S	00503	0000
0.700	0.600	0.100	0.310	0.200	0.200			En Ser	S	00501	0000
0.700	0.600	0.100	0.310	0.200	0.200			En Ser	S	00502	0000
0.700	0.600	0.100	0.310	0.200	0.200			En Ser	S	00503	0000
0.700	0.600	0.100	0.310	0.200	0.200			En Ser	S	00503	0000
0.700	0.600	0.100	0.310	0.200	0.200			En Ser	S	00501	0000
0.700	0.600	0.100	0.310	0.200	0.200			Libre	S	00000	0000

Ejemplo de despliegue de Tabla Definicion de Entradas Analogicas y Definicion de Salidas Analogicas.

> <

22-JUL-1987 07:53

DEF. ENTRADAS DIGITALES.Linea :00-04

Status	Tipo	Ubicacion	Ind_S
Libre	E	00000	0000
En Ser	E	01215	0000
En Ser	E	01215	0000
En Ser	E	01215	0000
Libre	E	00000	0000

DEF. SALIDAS DIGITALES.Linea : 00-04

Status	Tipo	Ubicacion	Ind_S
Libre	S	00000	0000
En Ser	S	00415	0000
En Ser	S	00415	0000
En Ser	S	00415	0000
Libre	S	00000	0000

Ejemplo de despliegue de Tabla de Definicion de Entradas Digitales y Definicion de Salidas Digitales.

> <

22-JUL-1987 07:47

TABLA RELFISLOG.Linea : 00-16

Csta	Slot	Ttarj	Tvar	Polar	Chal	Status
0	0	Com		0	0	No_Err
0	f	Cent2		0	0	No_Err
0	6	C_Ean		0	0	Conect
0	0			0	0	No_Err
0	0			0	0	No_Err
0	0			0	0	No_Err
0	4	SalDg	SDig	0	f	No_Err
0	4	SalDg	SDig	0	f	No_Err
0	4	SalDg	SDig	0	f	No_Err
0	c	EntDg		0	f	No_Err
0	c	EntDg		0	f	No_Err
0	5	SAn_C	SA_V	1	0	No_Err
0	5	SAn_C	SA_V	1	0	No_Err
0	5	SAn_C	SA_V	1	0	No_Err
0	5	SAn_C	SA_V	1	0	No_Err
0	5	SAn_C	SA_V	1	0	No_Err

TABLA DE ERRORES.Linea : 00-16

Clave	Descrip	Tarea	Repor	#Cons	Ocr
011	025	24	00	01	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00
000	000	00	00	00	00

Ejemplo de despliegue de Tabla Relacion Fisica - Logica y Tabla de Errores.

> <

01-NOV-1987 12:20

TABLA DE RANURAS . Rnr : 00
 Status AI Modo IC SCAl HisA HisM
 En Ser 01 Automat 22 0.5 0.5 0.5
 INDS: 0:000000001110:0:0:1:1 F:0.50
 KP KI KD SP PL KPA KLAE
 50.0 3.3 0.9 0.5 0.0 0.0 0.9

TABLA DE RANURAS . Rnr : 1
 Status AI IC SCAl
 En Ser 18 35 0.5
 INDS: 0:00000000010:0:0:1:1 F:0.50
 CTMPO VAR DE ENTRADA SALIDA
 25. 012000 : = 0.5 011000

KLBE KLAS KLBS A-C VDM SALIDA
 0.1 0.8 0.2 0 0.5 011000

VAR DE PROCESO IMPOSICION
 012001 : = 0.5 020001 : 00 = 0

ALARMA Y MODO PREALIMENTACION
 0000:1000:0000:1000 010001 : = 0.6

CONFIGURACION
 0110:0000:1000:0000

Ejemplo de Despliegue de la Tabla de Ranuras.

> <

01-NOV-1987 12:25

SECUENCIACION DE RANURAS. Linea: 00-10 SECUENCIACION DE RANURAS. Linea: 00-10
 Ranuras Totales: 40 Duracion Total: 112 Marco: 0 Ran/Marco: 10 Duracion: 28
 # St Rn AI St Rn AI St Rn AI St Rn AI # Status Ran Dur Algoritmo(##)
 0 ES 0 1 ES 0 1 ES 0 1 ES 0 1 0 En Ser 0 4 Pid (1)
 1 ES 1 13 ES 1 13 ES 1 13 ES 1 13 1 En Ser 1 1 Actuador (18)
 2 ES 2 8 ES 2 8 ES 2 8 ES 2 8 2 Config 2 4 Pid (1)
 3 ES 3 4 ES 3 4 ES 3 4 ES 3 4 3 En Ser 3 1 Actuador (18)
 4 ES 4 15 ES 4 15 ES 4 15 ES 4 15 4 Config 4 4 Pid (1)
 5 ES 5 20 ES 5 20 ES 5 20 ES 5 20 5 En Ser 5 4 PidRazon (2)
 6 ES 6 14 ES 6 14 ES 6 14 ES 6 14 6 En Ser 6 1 Actuador (18)
 7 ES 7 3 ES 7 3 ES 7 3 ES 7 3 7 En Ser 7 4 PidBanda (3)
 8 ES 8 2 ES 8 2 ES 8 8 ES 8 8 8 En Ser 8 1 Actuador (18)
 9 ES 9 6 ES 9 6 ES 9 6 ES 9 6 9 En Ser 9 4 PidBatch (5)
 10 Libre 0

Ejemplo de despliegue de la Tabla de Secuenciacion de Ranuras global y Tabla de Secuenciacion de Ranuras por Marco.

> <

22-JUL-1987 07:58

STACK TAREA 10 Cons_Rx																	
Direccion	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
7d0:	48	13	2d	12	04	00	02	1a	ff	ff	ff	03	09	12	03	ff	H.-.....
7e0:	02	1a	00	ff	ff	03	62	19	0a	02	0b	00	40	19	03	00b.....g..
7f0:	02	1a	ff	ff	ff	11	0a	00	ff	05	2b	0e	35	02	ff	00+..5..
800:	ff	28	ff	ff	39	ff	5e	ff	ff	ff	ff	ff	01	08	5c	ff9.....\
810:	ff	00														
820:	ff	7f	ff	ff	00	00										
830:	ff	00	00													
840:	ff	00	00													
850:	ff	00	00													
860:	ff	00	00													
870:	ff	7f	ff	ff	00	00										
880:	ff	00	00													
890:	ff	00	00													
8a0:	ff	ff	7f	ff	00	00										
8b0:	ff	00	00													

Ejemplo de despliegue del Stack para una Tarea.

> <

22-JUL-1987 07:49

VARIABLES ANALOGICAS.Linea : 00-16					VARIABLES ANALOGICAS.Linea : 00-16				
Status	Tipo	V_NOR	IA&C	V_UI IND_S	Status	Tipo	V_NOR	IA&C	V_UI IND_S
En Ser	S	-5.0000	0000	0.00 0003	En Ser	E	-5.0000	0000	0.00 8003
En Ser	S	0.0000	0000	0.00 0003	Fuera	I	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000
En Ser	S	0.0000	0000	0.00 0003	Fuera	E	0.0000	0012	0.00 0000

Ejemplo de despliegue de Tabla de Variables Analogicas de Salida y Tabla de Variables Analogicas de Entrada.

> <

22-JUL-1987 07:50

VARIABLES DIGITALES. Linea : 0-20
Status Tipo Valor Ind.C Ind.S
En Ser E fffb 0000 8003
En Ser E f300 0000 0000
En Ser E f300 0000 0000
En Ser E f300 0000 0000

VARIABLES DIGITALES. Linea : 00-040
Status Tipo Valor Ind.C Ind.S
En Ser S fffb 0000 0003
En Ser S 0000 0000 0003
En Ser S 0000 0000 0003
En Ser S 0000 0000 0003

Ejemplo de Despliegue de Tabla de Variables Digitales Entrada y Tabla de variables Digitales Salida.