

18
2 ej



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGON**

DISEÑO Y DESARROLLO DE UN SISTEMA
DE MONITOREO Y REGISTRO DE SECUENCIA DE
EVENTOS (MSE), BASADO EN MODULOS
DE CONTROL Y ADQUISICION LINEA SAC

FALLA DE ORIGEN

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

P R E S E N T A :

LAURA FABIOLA FALCON SALVADOR

DIRECTOR DE TESIS:
ING. SILVIA VEGA MUYTOY



EDO. MEX. 1995



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.

A mis padres, por su incondicional apoyo y sacrificio para la realización de mis metas trazadas hasta el momento.

A mis hermanas, por su apoyo y cariño demostrado.

**A Alejandro, por sus palabras siempre
alentadoras en los momentos difíciles y su valiosa
ayuda para la conclusión de este trabajo, que más
que mío es de ambos.**

AGRADECIMIENTOS

Al Ing. Miguel Angel Arriola Sancén, por su colaboración, apoyo y asesoría brindados para la realización del presente trabajo.

Al Instituto de Investigaciones Eléctricas (IIE), por el apoyo otorgado.

CONTENIDO

Introducción.

Capítulo

1.	ESPECIFICACIÓN DE REQUERIMIENTOS	5
1.1.	Análisis de sistemas comerciales	6
1.1.1.	Descripción de sistemas comerciales	9
1.1.2.	Tendencias detectadas	19
1.2.	Especificación de requerimientos	23
1.2.1.	Introducción	23
1.2.1.1.	Objetivo del MSE	23
1.2.1.2.	Alcance del MSE	24
1.2.1.3.	Definiciones, abreviaciones y acrónimos	25
1.2.1.4.	Referencias	26
1.2.2.	Descripción general	26
1.2.2.1.	Perspectivas del sistema MSE	26
1.2.2.2.	Funciones del sistema MSE	28
1.2.2.3.	Características del usuario	28
1.2.2.4.	Consideraciones generales	29
1.2.3.	Requerimientos específicos	30
1.2.3.1.	Especificación Funcional	30
1.2.3.1.1.	Señales de entrada	30
1.2.3.1.2.	Tiempo de resolución	30
1.2.3.1.3.	Filtrado de los eventos	31
1.2.3.1.4.	Memorización de eventos	31
1.2.3.1.5.	Presentación de eventos	32
1.2.3.1.6.	Mantenimiento	33
1.2.3.1.7.	Requerimientos funcionales	33
1.2.3.2.	Especificación eléctrica	36
1.2.3.2.1.	Voltaje en los puntos de entrada	36
1.2.3.2.2.	Inmunidad al ruido	36
1.2.3.2.3.	Alimentación	37
1.2.3.2.4.	Circuitos de comunicaciones	37

1.2.3.3.	Especificación mecánica	37
1.2.3.3.1.	Capacidad de puntos de entrada	37
1.2.3.3.2.	Dispositivos para el despliegue de eventos	38
1.2.3.3.3.	Gabinete	38
1.2.3.4.	Especificación ambiental	38
2.	DISEÑO Y DESARROLLO DEL MSE	39
2.1.	Análisis del sistema	39
2.2.	Diseño del MSE	43
2.2.1.	Introducción	43
2.2.1.1.	Propósito del diseño	43
2.2.1.2.	Alcance del MSE	43
2.2.1.3.	Definiciones y Acrónimos	46
2.2.2.	Referencia	46
2.2.3.	Descripción de la descomposición	46
2.2.3.1.	Descomposición en módulos	47
2.2.3.1.1.	Descripción del Subsistema de Adquisición	47
2.2.3.1.2.	Descripción del Subsistema de Presentación	56
2.2.4.	Descripción de las dependencias	61
2.2.4.1.	Dependencias entre módulos	61
2.2.4.2.	Dependencias entre las funciones de cada módulo	65
2.2.5.	Descripción de la interfaz	75
2.2.5.1.	Interfaces de los módulos	75.
2.2.5.2.	Diccionario de datos	77
2.2.5.3.	Catálogo de funciones	78
2.2.5.4.	Formato de ventanas	78
2.2.6.	Descripción del diseño detallado	84
2.2.6.1.	Diseño detallado de las funciones que integran el software del subsistema de adquisición (SAC)	84
2.2.6.2.	Diseño detallado de las funciones que integran el software del subsistema de presentación (PC)	89
2.2.7.	Descripción de la programación operativa	91
2.2.7.1.	Descripción del Sistema Operativo en el	

	Subsistema de Adquisición	91
2.2.7.2.	Descripción del Sistema Operativo en el Subsistema de Presentación	92
2.3.	Desarrollo del MSE	92
2.3.1.	Desarrollo de los módulos de programación del MSE	92
2.3.2.	Generación de versiones del software del subsistema de adquisición	93
2.3.3.	Generación de versiones del software del subsistema de presentación	94
3	PRUEBAS DEL MSE	97
3.1.	Programa de pruebas	97
3.1.1.	Introducción	97
3.1.2.	Objetivos	98
3.1.3.	Alcance	98
3.1.4.	Metodología de pruebas	98
3.1.5.	Características del sistema que fueron comprobadas	98
3.1.6.	Módulos del sistema que no serán probados	99
3.1.7.	Ambiente de pruebas	99
3.2.	Protocolo de pruebas del sistema	100
3.2.1.	Alcance	100
3.2.2.	Pruebas funcionales realizadas al subsistema de adquisición del MSE	101
3.2.2.1.	Prueba de adquisición de la capacidad nominal de los módulos de entrada	101
3.2.2.2.	Pruebas de la resolución de 1 mseg del sistema	103
3.2.2.3.	Pruebas de transmisión y recepción de información con el subsistema de presentación	108
3.2.2.4.	Prueba del almacenamiento de hasta los últimos 256 eventos	112
3.2.3.	Pruebas funcionales realizadas al subsistema de presentación del MSE	113
3.2.3.1.	Pruebas de recepción y transmisión de información con el subsistema de	

	adquisición	113
3.2.3.2.	Prueba de despliegue de información	116
3.2.3.3.	Prueba del almacenamiento de los últimos 1024 eventos en disco y verificación de consistencia de información	118
3.2.3.4.	Prueba del despliegue e impresión simultáneo de los eventos ocurridos	120
3.2.3.5.	Prueba del despliegue de los menús de opciones	122
3.3.	Certificación del sistema	123

Conclusiones 125

Apéndice

- A DICCIONARIO DE DATOS Y CATÁLOGO DE FUNCIONES**
- B PSEUDOCÓDIGO DE LAS FUNCIONES QUE INTEGRAN EL SOFTWARE DEL MSE.**

INTRODUCCION

La aplicación del control en los procesos industriales se ha ido incrementado considerablemente en los últimos años. Las cuatro décadas pasadas pueden ser consideradas como un testimonio firme del establecimiento de la teoría de control moderno y convencional para sistemas de control de tiempo continuo. Esta teoría, no sólo ha revolucionado los procesos industriales sino que además, ha permitido a la humanidad su inicialización en la exploración del universo.

A mediados de 1950, mientras las compañías de computadoras estuvieron experimentando con el Control Digital Directo (DDC), las de instrumentación permanecieron activas en el control analógico neumático de procesos y las de control de procesos se orientaron a las necesidades de las líneas de manufactura. Aplicando

dispositivos de control, las tres compañías fueron adquiriendo valiosas experiencias en la implantación de nuevas herramientas de control.

Posteriormente, mientras las compañías de instrumentación crecieron en la aplicación y distribución de sensores de procesos y elementos de control finales, las de control se enfocaron en la lógica discreta y dispositivos de energía. Lo anterior, es considerado como el inicio de las dos clases de sistemas de control de pisos de planta actualmente conocidos como sistemas de control distribuidos y sistemas basados en controladores programables.

En cuanto a tecnología se refiere, las compañías orientadas a lo electromecánico progresaron aplicando los nuevos dispositivos digitales existentes; paralelamente, el mundo de control de procesos tuvo un gran impacto en la industria dada la gran lista de procesos a controlar, para los que primeramente dispusieron de controladores analógicos neumáticos que posteriormente fueron reemplazados por los sistemas de control analógico electrónico. Por último, cabe mencionar que las experimentadas compañías de instrumentación aplicaron computadoras en el control de procesos de grandes plantas.

Debido a la creciente aplicación del control por varias industrias y a la gran cantidad de dispositivos de planta a mantener y controlar que esto implicaba, se originó la necesidad de crear una herramienta que permitiera el monitoreo del comportamiento de dichos dispositivos; es por esto, que a mediados de la década de los 70's se crearon los primeros **Sistemas de Registro de secuencias de Eventos** los cuales registran el estado de las señales del campo para su posterior despliegue o presentación.

Además, para evitar el incremento de costos de operación en el control de procesos, es también de gran utilidad el análisis inmediato de fallas ocurridas; esto se puede lograr con la aplicación de los sistemas de registro de secuencias de eventos, ya que proporcionan al operador en planta el registro cronológico de dichas fallas.

Los sistemas de registros de secuencias de eventos forman parte de los **sistemas supervisorios**, los cuales son una colección de equipos que proveen al operador de una localización remota de información para determinar el estado de una pieza particular del equipo o el de una entrada del proceso.

La principal aplicación de los sistemas de registro de secuencias de eventos se encuentra en subestaciones y plantas generadoras de energía eléctrica; además, se emplea en estaciones de bombeo y cualquier proceso industrial que requiere conocer con precisión y exactitud el desarrollo del mismo; pues con ellos, se registran cambios en el proceso detallando la secuencia en la que ocurrieron.

Tomando como ejemplo el funcionamiento de una unidad termoeléctrica, puede observarse la complejidad actual de todos sus equipos y sistemas, los cuales bajo condiciones normales son asimilables por el operador con relativa facilidad, situación diferente cuando se presenta una emergencia que desemboca en un disparo o salida de operación de un equipo importante o de la misma unidad. Ante esta circunstancia, el operador ha contado tradicionalmente para el análisis de anomalías, con indicadores de primera falla y anunciadores de alarmas, sin embargo, el tamaño de las centrales y el número de variables existente ocasionan que frente a una falla las ayudas del operador sean rebasadas dificultando el análisis mencionado, por ello, ahora gracias a la tecnología disponible es posible mejorar y fortalecer dichas ayudas.

OBJETIVO DE LA TESIS

El objetivo de la presente tesis es desarrollar un Sistema de registro de secuencias de eventos llamado Sistema de monitoreo y registro de secuencias de eventos (MSE), que como herramienta, proporcione al operador en planta de cualquier proceso industrial toda la información de los eventos ocurridos antes, durante y después de un disturbio del proceso, lo que le permitirá la rápida localización y corrección de los problemas que pudieran suscitarse durante la operación.

Este sistema se llevará a cabo con el apoyo de la infraestructura diseñada y construida por el Instituto de Investigaciones Eléctricas (IIE).

El presente trabajo de tesis se organiza de acuerdo a los siguientes capítulos, de los cuales se hace una breve descripción:

En el primer capítulo "Especificación del MSE" se hace una comparación de los sistemas de registro de eventos comerciales, la cual se considera como marco de referencia para el establecimiento de la especificación de requerimientos del sistema a desarrollar.

En el segundo capítulo "Diseño y desarrollo del MSE" se describe el diseño del sistema que cumplirá con los requerimientos establecidos, así como la descripción del desarrollo del sistema MSE en su totalidad.

El último capítulo "Pruebas del MSE", presenta el protocolo de pruebas aplicado a los componentes del sistema para la verificación del correcto funcionamiento de éste, así como, del cumplimiento de los requerimientos especificados en el capítulo 1.

Después de estos capítulos, se presentan las conclusiones a las que se llegaron al finalizar el desarrollo del sistema.

Por último se anexan 2 apéndices, que contienen: el diccionario de datos y un catálogo de funciones y el pseudocódigo de las funciones de software a desarrollar.

CAPITULO 1

ESPECIFICACIÓN DE REQUERIMIENTOS

Considerando que los sistemas de registro de secuencias de eventos son equipos diseñados y aplicados por diversas industrias, en este capítulo se lleva a cabo un análisis de las características y tendencias actuales de los sistemas comerciales existentes, que nos permitirán contar con un marco de referencia actual para la especificación de los requerimientos del MSE a desarrollar.

En primer lugar, se proporciona la descripción funcional general de un sistema registrador de secuencias de eventos y las principales características que distinguen a los diferentes registradores de secuencias de eventos existentes en el mercado, así como las tendencias detectadas entre ellos. Y por último, se lleva a cabo la especificación de los requerimientos del MSE.

1.1. ANÁLISIS DE SISTEMAS COMERCIALES.

Las principales funciones de un sistema registrador son tres: adquisición, procesamiento y despliegue sobre impresora.

- ⊙ **Adquisición.** Continuamente supervisan el estado de las señales provenientes de dispositivos de campo (interruptores, presostátos, botones, etc.), al detectar un cambio en algunas de dichas señales lo consideran como la ocurrencia de un evento.
- ⊙ **Procesamiento.** Consiste de la identificación y asignación del tiempo real de aparición y la descripción del evento detectado, así como su almacenamiento en una área de memoria llamada comúnmente "buffer" para su posterior despliegue.
- ⊙ **Despliegue.** Es la presentación al operador del registro de los eventos ocurridos a través de los dispositivos periféricos (impresora, CRT, etc.).

Lo anterior se puede apreciar en el diagrama funcional de la fig. 1.1.

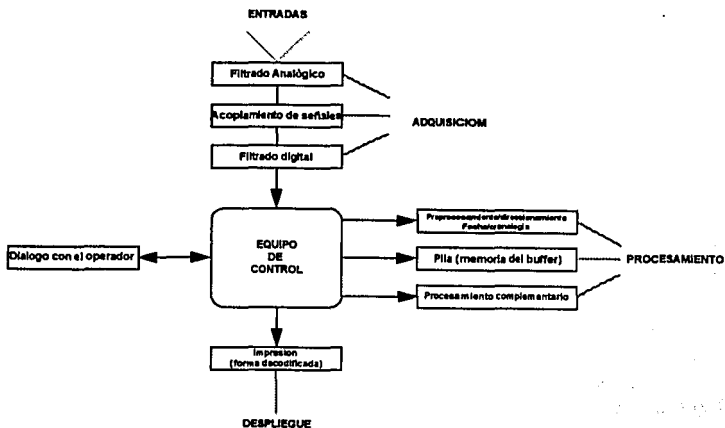


Fig. 1.1. Diagrama funcional de un sistema registrador de eventos.

Entre las principales características que distinguen a los registradores, se encuentran:

- ❑ **Puntos de entrada.** Es el número total de señales de campo que pueden ser registradas por los sistemas.
- ❑ **Tiempo de resolución.** Se refiere al tiempo mínimo entre 2 eventos sucesivos que permite distinguir su momento de aparición.
- ❑ **Tipo de señales de entrada.** Los sistemas registradores son diseñados para registrar la secuencia de operación de contactos de dos estados o voltajes.
- ❑ **Técnica de procesamiento.** Relacionada con la forma en que son identificados los cambios de estado, esta puede ser por polling (muestreo de todas las señales cada determinado tiempo) o por interrupción (en el momento que se genere un cambio de estado se produce una interrupción).
- ❑ **Configuración del sistema** (tiempo de filtrado, texto de las leyendas, etc.), la cual puede ser realizada en fábrica o bien, por el usuario en campo.
- ❑ **Tipo de aislamiento y Tipo de filtrado.** El tipo de aislamiento es una característica de los circuitos de entrada de los sistemas; pueden ser optoacopladores que proporcionan aislamiento eléctrico y de ruido granulado (ruido generado en los dispositivos activos, tales como válvulas y transistores, por la velocidad variable en forma aleatoria del movimiento de los electrones bajo la influencia de tensiones aplicadas desde el exterior mediante las adecuadas terminales de electrodos) a los registradores lógicos en el alambrado de campo; o bien, relevadores que proporcionan aislamiento eléctrico y la capacidad de manejar potencia eléctrica entre los dispositivos de campo y del sistema.
- ❑ **El filtrado de los eventos impide la impresión de eventos transitorios relacionados con rebotes de contacto con lo que se registran sólo datos reales, la confirmación de los datos se puede hacer a través de un filtro por hardware o un filtro por software.**

- ❑ Dispositivos para el despliegue de los eventos. Estos pueden variar según el sistema, en algunos el despliegue es sólo en impresora y en otros puede ser en impresora y en CRT.
- ❑ Dispositivos opcionales. Son aquellos que se pueden agregar al sistema para el despliegue de los eventos y pueden ser: impresoras remotas, CRT's remotas, etc.
- ❑ Conexión a sistemas de nivel superior, esta característica proporciona las opciones de comunicación con sistemas de control de nivel superior; estos pueden ser: sistemas de control supervisorio, sistemas de control supervisorio y adquisición de datos (por sus siglas en inglés, SCADA) y sistemas de control distribuido (por sus siglas e inglés, DCS).
- ❑ Información de los despliegues. En su forma más resumida la información que esta contenida en el despliegue de los eventos es: el tiempo y la hora de ocurrencia del evento, el número del punto en el cual se presentó el cambio de estado y el estado del evento. Es importante saber la información a desplegar por cada sistema, pues esta puede variar.
- ❑ Capacidad de memorización. Los sistemas tienen la capacidad de almacenar en memoria RAM cierta cantidad de eventos para consultas e impresiones posteriores.
- ❑ Diagnóstico. Esta característica permite diagnosticar alguna falla tanto en la operación como en el propio sistema. Pueden variar los elementos que se diagnostican dependiendo del sistema.
- ❑ Estándares aplicables. Para que los sistemas sean confiables deberán cumplir con determinados estándares, los cuales es importante mencionar.
- ❑ Temperatura de operación. Para la correcta operación del sistema se debe tomar en consideración el rango de temperatura en que funciona el sistema.
- ❑ Alimentación principal y de respaldo. Especifica el voltaje de alimentación del sistema.

1.1.1. DESCRIPCION DE SISTEMAS COMERCIALES.

A continuación se describen de manera general algunos sistemas comerciales con el objeto de distinguir y presentar las tendencias tecnológicas detectadas en las características ofrecidas por otros equipos registradores de secuencias de eventos.

1. Sistemas de Registro de Eventos SEPAC.

La compañía Sistemas Eléctricos de Potencia, Automatización y Control (SEPAC), S. A. de C. V. desarrolla equipos y/o sistemas orientados al control de centrales y subestaciones; con un 85% de ingeniería nacional y tecnología transferida por la Cía. THORN AUTOMATION de Inglaterra.

Los sistemas de registro de eventos REPEPAC son diseñados para detectar y registrar cambios de estado o aparición de fallas, desplegados sobre pantalla de rayos catódicos e impresora (local o remota) y almacenarlos en diskettes.

Entre los principales modelos de esta línea se encuentran:

- ◆ REPEPAC XAR2-256 (Reportador de eventos para 256 puntos).
- ◆ REPEPAC XAR2-512 (Reportador de eventos para 512 puntos).
- ◆ REPEPAC XAR2 (256XN) (Reportador de eventos para múltiplos de 256 juntos, máximo 2048 puntos).

Las características principales de estos sistemas son:

Capacidad de adquisición de 256 puntos empleando un solo módulo y hasta 2048 puntos empleando módulos de expansión de 256 puntos.

Tiempo de resolución de 1 milisegundo, aún para avalancha de la totalidad de puntos.

Dos niveles de aislamiento entre los dispositivos de campo y el sistema: relevadores y optoacopladores. Eliminación de rebotes de contacto a través de un filtro por software.

Es un sistema fácilmente configurable y diagnosticable; además, cuenta con la facilidad para programar la fecha y la hora.

El reporte de salida de los eventos registrados puede ser en impresora local o remota y contiene: la hora del día, minutos segundos y milisegundos; caracter del cambio de estado (APA: aparición, DES: desaparición e INE: inestable); caracter de grado de urgencia (I: información, N: normal y U: urgente); 4 caracteres para indicar la localización geográfica del punto en el sistema, 3 caracteres para la unidad generadora del evento, 4 caracteres para la clave del evento y 40 caracteres incluyendo espacios en blanco para la descripción del evento (Ver Fig. 1.2).

La función principal de estos sistemas es la de imprimir mediante impresora los diferentes eventos que aparecen en una central o subestación eléctrica.

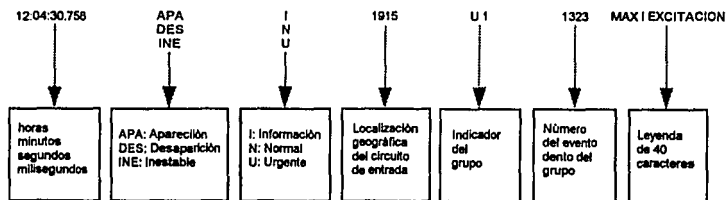


Fig. 1.2. Reporte típico en impresora de los reportadores SEPAC.

2. Sistemas de Registro de Eventos HATHAWAY.

La compañía Hathaway Instruments Inc, diseña, desarrolla y construye diversos sistemas de registro de eventos; entre los más populares se encuentran:

- ◆ Registrador de Eventos series 533 (Sistema de pequeña escala).
- ◆ Registrador de Eventos series 5330 (Sistema de mediana escala).
- ◆ Registrador de Eventos series 5000 (Sistema de gran escala).

El registrador series 533, es un sistema modular con capacidad de adquisición de 128 puntos en módulos de 4 puntos. Tiene una resolución de 1 milisegundo. Memorización para 16 eventos, con opción de expansión hasta 32 eventos; inmunidad al ruido y eliminación de rebotes de contacto a través de un filtro de Hardware. Incluye funciones de diagnóstico y resumen de eventos en alarmas.

El reporte de los eventos contiene el código del día, la hora, minutos, segundos y milisegundos, el número del punto y el estado del evento. Los eventos en estado de alarma se imprimen en color rojo y en negro los eventos en estado normal (Ver Fig. 1.3).

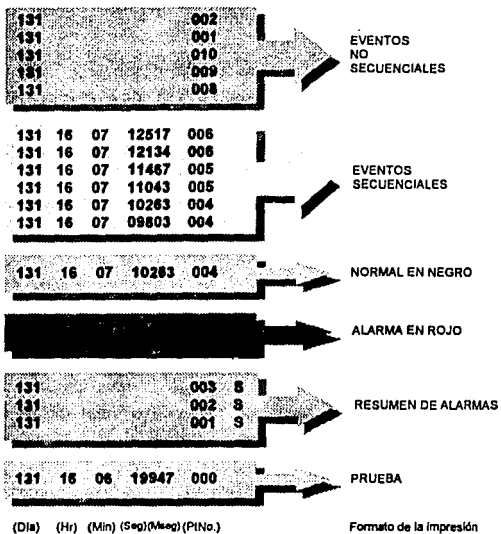


Fig. 1.3. Formato de impresión del registrador SERIES 533

Entre las principales características del modelo 5330, se tiene: capacidad de adquisición de 512 puntos de entrada, resolución de 1 milisegundo y capacidad de memorización de 80 eventos. Las entradas son aisladas por medio de optoacopladores y cuenta con un filtro por hardware para la eliminación de rebotes.

La presentación de los eventos contiene: el día y el tiempo de aparición (hora, minutos, segundos y milisegundos), el estado del punto (A: alarma y N: regreso a normal), la descripción del punto y la palabra de estado (Ver Fig. 1.4). En el caso de que se trate de un evento en alarma, la impresión es en color rojo, en otro caso la impresión es en negro.

Millisegundos	Día	Hr	Min	Seg	
					TIME TEST
			002		
			16	05:36	
	110	A	021		TURB OIL RESERVE LEVEL LOW
	182	A	126		CHARGER No 1 DC VOLTS HIGH
	561	A	042		INV No 4 AC/DC BRKR TRIPPED
	923	N	042		INV No 4 AC/DC BRKR RESET
	118		16:05:37		
	502	N	021		TURB OIL RESERVE LEVEL NORMAL
	113		16:05:38		
	316	A	316		TRANSFORMER T1 TROUBLE
	418	A	214		MAIN PWR TRANS OIL TEMP HIGH
	815	N	126		CHARGER No 1 DC VOLTS NORMAL
	920	N	615		250 DC BUS 4 GROUND YES
	118		16:06:02		
	201	A	186		TURBINE OVERSPEED TRIP
	415	N	186		TURBINE OVERSPEED RESET
	826	A	310		FUEL PROTECT CKT OFF
	980	N	428		TURB OIL PURIFIER NORMAL

Edo. del punto	Número de punto	Descripción del evento	Palabra de estado
N	428	TURB OIL PURIFIER	NORMAL

Fig. 1.4. Impresión típica del modelo 5330

El modelo 5000 al igual que el 5330 son empleados en equipos de protección de relevadores en sistemas de energía y equipos de control en plantas de energía o servicios de procesos industriales continuos.

El modelo 5000, tiene: capacidad para 2048 puntos de entrada, resolución de 1 milisegundo y capacidad de memorización de 2048 eventos. Incluye acopladores ópticos en sus circuitos de entrada para supresión de ruido, y la eliminación de rebotes en los puntos de entrada es a través de un filtro por hardware. La presentación de los eventos es la misma que la del modelo 5330.

Estos dos últimos modelos, también incluyen sumario de alarmas y funciones de diagnóstico.

3. Registradores de eventos de Productos BETA.

Productos BETA es una división de la corporación de sistemas HATHAWAY, líder en el diseño y manufactura de sistemas registradores de secuencias de eventos SER, anunciadores e instrumentos de calidad.

Entre los principales modelos de SER de Productos BETA, se encuentran:

- ◆ BETALOG 128 (Reportador de Eventos de pequeña escala).
- ◆ BETALOG 512 (Reportador de Eventos de mediana escala).
- ◆ BETALOG 4100 (Reportador de Eventos de mediana y gran escala).

El BETALOG 128 tiene una capacidad total de 128 puntos en incrementos de 32, capacidad de memorización de 256 eventos almacenados en una forma circular y una resolución de 1 milisegundo. Cuenta con un filtro por software para eliminar los rebotes en todas las entradas; las cuales, son aisladas a través de optoacopladores. Cuenta con 4 puertos de salida RS-232C.

El registro de los eventos que se presenta al operador consisten de, el tiempo, el número del punto, la descripción del evento en 60 caracteres alfanuméricos y se indica el estado del evento con un caracter (A: Alarma y N: regreso a normal). (Ver Fig. 1.5).

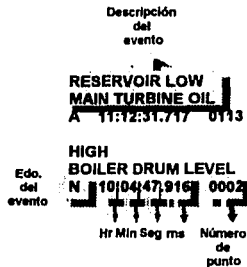


Fig. 1.5. Ejemplo de impresión del SER BETALOG 128

El modelo BETALOG 512 es un sistema modular de mediana escala. Con capacidad mínima de 64 puntos con incrementos de 64 puntos hasta un máximo de 512 puntos, resolución de 1 milisegundo y capacidad de memorizar hasta 2000 eventos en un buffer circular. Todas las entradas son aisladas a través de optoacopladores. Cuenta con un filtro por software ajustable desde teclado sobre un sólo punto o un grupo de puntos base desde 1 milisegundo a 65 segundos.

El registro típico que se muestra al operador, consiste de: el tiempo de aparición (hora, minutos, segundos y milisegundos), el número del punto, el estado del evento indicado con un caracter (A: alarma y N: regreso a normal) y la leyenda del punto en 60 caracteres alfanuméricos (Ver Fig. 1.6).

En la Fig. 1.6, el regreso a normal es indicado con dos espacios.

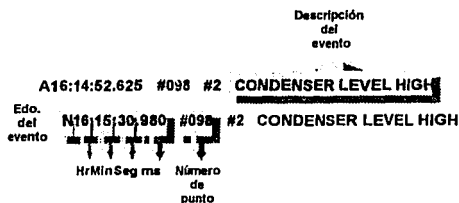


Fig. 1.6. Ejemplo de impresión del SER BETALOG 512

El modelo BETALOG 4100 es un sistema diseñado modularmente para aplicaciones de mediana y de gran escala. Con una capacidad máxima de 512 puntos en módulos de 64 para un bastidor simple y para un bastidor múltiple con un total de 4096 con 8 bastidores de 512 puntos. Capacidad de memorización de 6000 eventos. Cuenta con un filtro por software y una resolución de un milisegundo. Las entradas son aisladas con optoacopladores. La salida de los reportes presentadas al operador es la misma que para el modelo BETALOG 512.

Estos dos últimos modelos, cuentan con 7 puertos de comunicación para conectarse con numerosos dispositivos externos.

4. Registrador de eventos RONAN.

La ingeniería de la corporación RONAN ha diseñado el registrador de secuencias de eventos Ronan Series X500. Tiene una capacidad de 4096 puntos de entrada con una resolución de 1 milisegundo. Es un sistema fácilmente configurable y diagnosticable. Cuenta con un filtro por software programable por el usuario.

El despliegue de los eventos puede ser en monitor monocromático o en monitor a color y en una impresora local. Cuando se trata de monitor a color, los eventos se muestran al operador en distintos colores, asignados en planta dependiendo la descripción del evento. El registro de los eventos contiene el tiempo (hora, minutos, segundos y

milisegundos), el número del punto, la descripción del evento, y el estado del evento (*A: Alarma, **A: Alarma con prioridad alta, N: regreso a normal con una baja prioridad y -N: regreso a normal con alta prioridad).

5. Sistema de Registro de Eventos ENERTEC.

El sistema de registro de eventos EMS 1000 es el resultado de la amplia experiencia de la Societe D' Instrumentation Schlumberger de Francia y ENERTEC, en el área de adquisición y procesamiento de datos.

El sistema EMS 1000 esta constituido en forma modular con capacidad de 464 puntos de entrada en tarjetas de 16 puntos cada una, tiempo de resolución de 5 milisegundos, capacidad de memorización de 200 eventos, con un filtro por hardware con un tiempo de 15 milisegundos para eliminar los rebotes de las entradas de campo.

Las tres principales funciones del EMS 1000 son: adquisición, procesamiento y despliegue en impresora. Se puede llevar acabo comunicación local y remota vía los estándares RS-232 y RS-422 y contiene funciones que le permiten ser configurable en campo. El teclado para el diálogo con el operador y un impresor de 40 columnas están integrados al sistema. Los reportes contienen los parámetros de los eventos ocurridos, estos son, el tiempo de aparición, la clave del punto, el nivel de urgencia y la descripción de éstos.

6. Sistema de registro de eventos RIS.

Entre los sistemas de registro de eventos que la compañía Americana Rochester Instrument Systems, diseña y construye, se tiene el modelo RA-800N. Este es un sistema modular dedicado al registro de eventos, para el cual emplea técnicas de procesamiento de alta velocidad.

Entre las principales características con que cuenta este sistema, se tiene: capacidad de adquisición de hasta 500 ó 1000 puntos en grupos de 40, tiempo de

resolución de 1 milisegundo para 500 puntos de entrada y de 2 milisegundos para 1000 puntos; filtro de hardware de 4 milisegundos para eliminación de rebotes de contactos; contiene funciones de diagnóstico, resumen de eventos en alarma y actualización de tiempo, todas controladas en el panel principal.

La información de los eventos ocurridos se presenta, en código alfanumérico, a través de una impresora local, la cual contiene el tiempo de aparición a los más cercanos milisegundos, cuatro dígitos para identificación del punto, símbolos de estados que indican el tipo de evento (- (guión): evento normal, A: alarmas, P: pruebas y S: resumen de alarmas). (Ver Fig. 1.7).

Hrs.	Seg.	Min.	msg.	
112506	003	0068	-	
112507	955	0127	A	
112506	954	0172	A	
112507	892	0043	-	
112507	878	0159	A	
112507	872	0068	A	
110000	000	0000	-	
324	000	0000	P	
104513	199	0081	-	
103954	872	0068	-	
102006	242	0137	A	
100000	000	0000	-	
324	000	0000	P	

Identifi- Símbolo
 cador de
 del estado
 punto

Fig. 1.7 Impresión típica del modelo RA-800N

Cuando la memoria se encuentra vacía se genera un avance de papel automático, el cual consiste de un espacio doble entre los grupos que ocurren rápidamente o los eventos individuales. Los eventos en alarma se imprimen en color rojo y los demás eventos en negro.

1.1.2. TENDENCIAS DETECTADAS.

En la tabla 1.1 se muestran las principales características de los sistemas mencionados anteriormente. y en base a esta tabla comparativa, se deriva que las tendencias detectadas dentro del mercado son las siguiente:

- La mayoría de los productos son desarrollados con estructuras modulares lo que les permite incrementar su capacidad de punto de entrada, el número varía entre 128 y 4096 puntos de entrada que acepta el modelo BETALOG 4100.
- El tiempo de resolución en todos los modelos es de 1 mseg. a excepción del modelo EMS-1000 de ENERTEC que tiene una resolución de 5 mseg y de 2 mseg. en el RA-800N de RIS para 1000 puntos de entrada.
- Independientemente del modelo, la adquisición de los eventos es a través de tarjetas de entradas digitales para todos los sistemas.
- Los sistemas utilizan técnicas de muestreo continuo de las entradas ("Polling") o de Interrupción, ambos para indentificar la ocurrencia de los eventos; pero la técnica más empleada es la de Interrupción.
- La mayoría de los sistemas cuentan con facilidades para ser configurables de acuerdo a las necesidades del usuario, a excepción del modelo SERIES-5330 de HATHAWAY y el RA-800N de RIS que son configurados en fabrica.
- Realizan el aislamiento con el campo de aplicación a través de relevadores y optoacopladores, pero en su mayoría a través de estos últimos. Los rebotes

de contactos se eliminan por medio de filtros por "hardware" o filtros por "software". En los filtros por software se le permite al operador programar el tiempo de duración por lo que son los más empleados.

- El principal dispositivo utilizado para el despliegue de los eventos es la impresora local y algunos modelos emplean un CRT. Entre los dispositivos opcionales que se pueden conectar a los sistemas, se encuentran: impresoras remotas, CRT's remotas y anunciadores de ventana.
- Los sistemas registradores de secuencias de eventos se pueden conectar a diferentes sistemas de control: Sistemas de Control Distribuido (DCS), Sistemas de Control Supervisorio y Sistemas de Control Supervisorio y Adquisición de Datos (SCADA).
- Los reportes de salida presentan desde la más elemental hasta la más completa información sobre los eventos ocurridos dependiendo el modelo a que se haga referencia. La mínima información que contienen es la siguiente: tiempo de aparición, No. del punto y el estado del evento. Algunos modelos incluyen la descripción del evento, grado de urgencia, localización del punto y la palabra de estado.
- La capacidad de memorización varía desde 16 hasta 8000 eventos en memoria principal (RAM).
- Todos los modelos comparados disponen de una prueba de diagnóstico que permite conocer y probar la disponibilidad de sus módulos.
- En su mayoría los sistemas comerciales están regidos por las normas ANSI/IEEE 37.90/4723 "Standard Surge Withstand Capability test" la que se refiere a la prueba de capacidad que presenta el sistema contra voltajes transitorios.

MODELOS	SEFAC REPEFAC XAR2-256	SEFAC REPEFAC XAR2-256	SEFAC REPEFAC XAR2-256	HATAHA WAY SERIES-533	HATHAW AY SERIES-5330	HATHAWA Y SERIES-5000	BETA BETALOG -128	BETA BETALOG -512	BETA BETALOG-4100	RONAN SERIES-X500	ENERTEC EMS 1000	RIS RA-800N
CARACTERISTICAS												
PUNTOS DE ENTRADA	256 Puntos	512 Puntos	2048 Puntos	128 Puntos	512 Puntos	2048 Puntos	128 Puntos	512 Puntos	4096 Puntos	4096 Puntos	464 Puntos opcional 960 Puntos	500 Puntos o 1000 Puntos
T. DE RESOLUCIÓN	1 msg.	1 msg.	1 msg.	1 msg.	1 msg.	1 msg.	1 msg.	1 msg.	1 msg.	1 msg.	5 msg.	1 msg. para 500 puntos 2 msg. para 1000 puntos
TIPOS DE SEÑALES DE ENTRADA	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales	Digitales
TECNICAS DE PROCESAMIENTO	Poling	Poling	Poling	Poling	Interrupción	Interrupción	Interrupción	Interrupción	Interrupción	Interrupción	Interrupción	Poling
CONFIGURACION	En campo	En campo	En campo	En campo	En fabrica	En campo	En campo	En campo	En campo	En campo	En campo	En fabrica
T. DE AISLAMIENTO Y T. DE FILTRADO	Relevadores y optocoplador en filtro SOFTWARE	Relevadores y optocoplador en filtro SOFTWARE	Relevadores y optocoplador en filtro SOFTWARE	Relevador es Filtro por Hardware	Optocopladores Filtro por Hardware	Optocopladores Filtro por Hardware	Optocopladores Filtro por Software	Optocopladores Filtro por Software	Optocopladores Filtro por Software	Optocopladores Filtro por Software	Optocopladores Filtro por Hardware	Optocopladores o relevadores Filtro por Hardware
DISPOSITIVO PARA EL DESPLIEGUE DE LOS EVENTOS	CRT. Impresora local y diskettes	CRT. Impresora local y diskettes	CRT. Impresora local y diskettes	Impresora local	Impresora local	Impresora local	Impresora Integrada y display de cristal liquido	Impresora local	Impresora local	CRT a color, Impresora local y disco duro o floppy	Impresora local	Impresora local
DISPOSITIVOS OPCIONALES	Impresora remota	Impresora remota	Impresora remota	Impresora remota	Impresora remota, CRT's remotas y anunciador es de ventana	Impresora remota, CRT's remotas y anunciador es de ventana	CRT a color, RTU's e Impresoras remotas	CRT a color, impresoras remotas y anunciador es de ventana	CRT a color, impresoras remotas y anunciador es de ventana	Impresora remota, y monitores remotos	Impresora remota	Impresora remota
SISTEMAS DE NIVEL SUPERIOR A QUE SE PUEDEN INTEGRAR	Sistemas de control distribuido	Sistemas de control distribuido	Sistemas de control distribuido	Sistemas de transmisión de datos.	Sistemas de control supervisorio	Sistemas de control supervisorio	SCADA y sistemas de control distribuido	Sistemas de control distribuido	Sistemas de control distribuido	Sistemas de control distribuido	SCADA	Sistemas de control Supervisorio
INFORMACION DE LOS DESPLEGES	T. de aparición, No. del punto, edo. del evento, descripción, grado de urgencia, localización	T. de aparición, No. del punto, edo. del evento, descripción, grado de urgencia, localización	T. de aparición, No. del punto, edo. del evento, descripción, grado de urgencia, localización	Tiempo de aparición, No. del punto, estado del evento,	Tiempo de aparición, No. del punto, edo. del evento, descripción y palabra de estado	Tiempo de aparición, No. del punto, edo. del evento, descripción y palabra de estado	Tiempo de aparición, No. del punto, estado del evento,	Tiempo de aparición, No. del punto, estado del evento, descripción	Tiempo de aparición, No. del punto, estado del evento, descripción	Tiempo de aparición, No. del punto, estado del evento, descripción	Tiempo de aparición, clave del punto, nivel de urgencia, descripción del evento	Tiempo de aparición, No. del punto, estado del evento, descripción

CAPACIDAD DE MEMORIZACIÓN	256 eventos	256 eventos	256 eventos	16 eventos expensible a 32 eventos	90 eventos	2048 eventos	256 eventos	2000 eventos	6000 eventos	800 eventos	200 eventos opcional e 1000	64 eventos opcional hasta 320 eventos para 1000 puntos
DIAGNOSTICOS	Disponible (enlaces serie)	Disponible (enlaces serie)	Disponible (enlaces serie)	Disponible (Tarjetas de los módulos y tarjetas de entrada)	Disponible (Operación del propio sistema , puntos de entrada y dispositivos periféricos)	Disponible (operación del propio sistema , puntos de entrada y dispositivos periféricos)	Disponible (Operación del propio sistema y de los módulos que lo constituye n)	Disponible (Operación del propio sistema y de los módulos que lo constituye n)	Disponible (Operación del propio sistema y de los módulos que lo constituyen)	Disponible (Operación del propio sistema y equipo disponible)	Disponible (Operación del propio sistema , equipo y puntos de entrada)	Disponible (Operación del propio sistema y puntos de entrada)
ESTANDARES APLICABLES	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74	IEEE 472/74 ANSI 37.9/74
TEMPERATURA DE OPERACION	-5°C - 55°C	-5°C - 55°C	-5°C - 55°C	-17°C +50°C	-17°C +50°C	-17°C +50°C	0°C +50°C	0°C +50°C	0°C +50°C	0°C +60°C	0°C +50°C	0°C +50°C
ALIMENTACIÓN PRINCIPAL Y RESPALDO	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo	Voltaje alterno voltaje directo

1.2. ESPECIFICACIÓN DE REQUERIMIENTOS.

Esta sección, contiene los requerimientos del sistema de monitoreo y registro de secuencia de eventos (MSE), tomando como referencia las tendencias detectadas en los sistemas comerciales comparados anteriormente; los cuales, son el punto de partida para su posterior diseño y desarrollo.

La especificación de requerimientos, se lleva a cabo en tres partes:

- ① Introducción.
- ② Descripción General.
- ③ Requerimientos Específicos.

En la introducción se proporciona de manera general, los requerimientos del sistema (MSE) que será diseñado y desarrollado.

En la segunda parte se hace énfasis en los requerimientos que afectarán la especificación del MSE.

Por último, se describen las especificaciones de los requerimientos del MSE, con las cuales se delinearé el diseño del propio sistema.

1.2.1. INTRODUCCIÓN.

1.2.1.1 Objetivo del MSE.

El sistema de monitoreo y registro de secuencia de eventos (MSE), deberá llevar a cabo:

- ☛ La detección de los eventos (cambios de estados lógicos en las señales monitoreadas) en las entradas digitales de cualquier proceso industrial.

- ☛ La identificación y registro de los eventos detectados de manera cronológica.
- ☛ La presentación al operador de la secuencia cronológica de los eventos.
- ☛ Almacenamiento de la secuencia cronológica de los eventos para posteriores análisis del operador (ver Fig. 1.8).

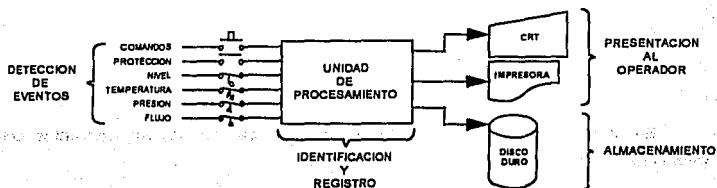


Fig. 1.8. Funciones que realizará el MSE.

1.2.1.2. Alcance del MSE.

El sistema deberá identificar los cambios de estado lógico ocurridos en las entradas del módulo SAC, etiquetarlos con la fecha y hora en que ocurrieron y desplegarlos al operador a través de un CRT y de una impresora, así como su almacenamiento en memoria primaria (RAM) y memoria secundaria (diskettes o disco duro).

La información procesada por el MSE se proporcionará al operador mediante un CRT (Tubo de rayos catódicos), reportando el historial de eventos presentados en el proceso de manera cronológica. Además, los eventos deberán imprimirse en forma continua.

El sistema deberá incluir un reloj calendario en tiempo real, para etiquetar los eventos con la fecha y hora en que ocurrieron.

Se deberá permitir la inicialización de la fecha y hora a solicitud del usuario y se podrán generar distintos reportes de los datos procesados por el MSE; así como, la configuración de la presentación de la información.

La presentación en CRT de los eventos registrados deberá contener la siguiente información: la fecha y hora en que ocurrieron, la descripción del evento y el estado del evento; en caso de que se trate de una condición en alarma el evento se presentará en un color dado y para otra condición en un color diferente.

El sistema deberá proporcionar al operador un diagnóstico funcional del sistema.

El sistema MSE podrá tener la posibilidad de ser expandible, para conectarse a un sistema de adquisición de datos (SCADA) o sistemas de control distribuido (DCS).

1.2.1.3. Definiciones, abreviaciones y acrónimos.

Para la correcta interpretación de esta especificación de requerimientos, a continuación se definen todos los términos, acrónimos y abreviaciones empleadas.

PC	Computadora personal.
CRT	Pantalla (Tubo de rayos catódicos)
MSE	Sistema de monitoreo y registro de secuencia de eventos.
EVENTO	Es una señal de adquisición que presenta un cambio de estado lógico, ya sea de 0 a 1 ó de 1 a 0.
BUFFER	Segmento de memoria RAM reservada para el almacenamiento de los eventos detectados.

1.2.1.4. Referencias.

Institución	Estándar
INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS (IEEE)	472/1974 IEEE, Guide for Surge Withstand Capability test
ELECTRONIC INDUSTRIES ASSOCIATION	RS-232-C. Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange.
ELECTRONIC INDUSTRIES ASSOCIATION	RS-485. Standar for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems
COMISIÓN FEDERAL DE ELECTRICIDAD (CFE)	GAHRO-24. Registrador de eventos secuenciales para centrales y subestaciones eléctricas.
INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS (IEEE)	ANSIEEE std. 830/1984. IEEE Guide to Software Requirement Specifications.

1.2.2. DESCRIPCIÓN GENERAL.

En esta sección se describen los factores generales que afectan al MSE y sus requerimientos. Se pretende que con esta descripción se facilite el entendimiento de dichos requerimientos.

1.2.2.1. Perspectivas del sistema MSE.

De manera global, el sistema estará conformado como se muestra en la fig. 1.9, en donde se presenta la interacción que podrá existir con el MSE, a través de una red de comunicaciones para proporcionar información a una minicomputadora de adquisición de datos, en donde la información presentada al operador y a los sistemas supervisores será

la misma.

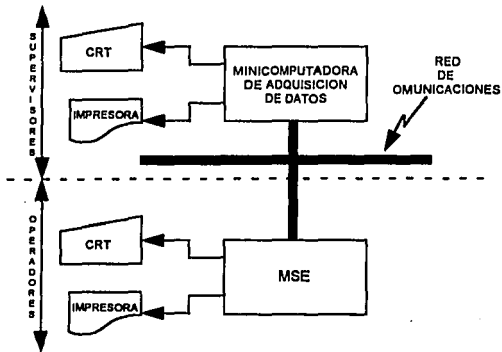


Fig. 1.9. Composición global del MSE.

La interacción con la red de comunicaciones será para transmitir información a un sistema de nivel superior (DCS, SCADA, etc.).

Cabe aclarar que el MSE podrá funcionar independientemente de que se encuentre o no conectado a la red de comunicaciones; es decir, el MSE seguirá realizando sus funciones normales.

Esta interacción del MSE con la red de comunicaciones, está fuera del alcance del presente trabajo.

1.2.2.2. Funciones del sistema MSE.

En este punto se proporcionan las funciones globales que deberá llevar a cabo el sistema MSE.

- ☛ Acoplamiento de las señales.
- ☛ Detección de los eventos.
- ☛ Identificación de los eventos con el número de señal, estado, fecha y hora en que ocurrieron.
- ☛ Almacenamiento de los eventos.
- ☛ Manejo de las secuencias de eventos, interactuando con el usuario para seleccionar el medio por el que se presentará la información.
- ☛ Presentación de las secuencias de eventos, en un formato que contendrá la información que indique al usuario la cronología de los eventos.

1.2.2.3. Características del usuario.

A continuación, se describen las características generales de los usuarios del MSE, que afectarán a los requerimientos específicos.

Los usuarios requieren que el sistema sea:

- ♦ Altamente disponible. EL sistema permanecerá conectado al proceso las 24 horas del día, los 365 días del año, es decir, el tiempo que el proceso así lo requiera.
- ♦ Altamente mantenible. El usuario requiere que el sistema le presente una forma de determinar y corregir fallas ocurridas en su operación.

- **Altamente confiable.** Se requiere que el sistema opere automáticamente para registrar cualquier disturbio de las señales conectadas al sistema.
- **Altamente preciso.** El sistema deberá registrar los eventos en el orden que se presenten hasta con una resolución de 1 mseg.

Los usuarios del sistema terminado serán, el personal de operación y el de mantenimiento de una planta de cualquier tipo de proceso industrial, los cuales no cuentan con conocimientos de programación (por ejemplo, programación estructurada) ni de la constitución interna del MSE (en cuanto a circuitería se refiere) por lo que, el usuario requiere de una interface hombre-máquina que opere a través de menús para facilitarle el manejo del propio sistema y que la información que proporcione éste (reporte en CRT e Impresora de los eventos ocurridos, diagnósticos, etc.) sea clara para su fácil y rápida interpretación.

1.2.2.4. Consideraciones generales.

Este punto proporciona una descripción general de los factores que limitarán las opciones de desarrollo del sistema MSE.

1. Limitaciones de hardware.

El equipo de hardware disponible por el IIE para la constitución de este sistema es la línea SAC IBUS-III desarrollada por el departamento de electrónica del mismo instituto, una PC, una impresora y una microcomputadora (MVAX) que realiza las funciones de adquisición de datos de nivel jerárquico superior (cuya programación queda fuera del alcance del desarrollo de este sistema).

2. Limitaciones de software.

Para el desarrollo del MSE el IIE dispone de dos ambientes de programación, un ambiente multitareas, el Sistema Operativo MTOS (Multi-tasking Operating System) para la línea SAC IBUS III y el Sistema Operativo Ms-DOS (Micro-Soft

Disk Operating System) para la PC. Además, dispone de las siguientes herramientas de programación: el lenguaje ensamblador de Intel, el lenguaje C de Intel, el lenguaje ensamblador de Microsoft y el lenguaje C de la compañía Borland conocido como Turbo C.

1.2.3. REQUERIMIENTOS ESPECÍFICOS.

En esta sección, se proporcionan los detalles de los requerimientos específicos para crear el diseño del sistema MSE.

1.2.3.1. Especificación Funcional.

En esta sección, se establecen las características funcionales bajo las cuales se desarrollará el MSE.

1.2.3.1.1. Señales de entrada.

Las señales de entrada que adquirirá el MSE serán de tipo discretas (valores lógicos 1's y 0's), provenientes de los dispositivos de campo (interruptores límites, botones, selectores, etc.).

1.2.3.1.2. Tiempo de resolución.

El tiempo de resolución del sistema será de 1 milisegundo. Este tiempo, es el tiempo mínimo para determinar la secuencia entre dos eventos distintos. (Ver fig. 1.10).

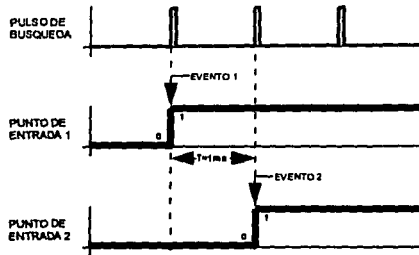


Fig. 1.10. Tiempo mínimo entre dos eventos

1.2.3.1.3. Filtrado de los eventos.

El MSE incluirá un filtro por software para cada uno de los eventos que aparezcan en los puntos de entrada. Este filtro consistirá en un retardo de tiempo para la validación del evento, lo cual posibilitará al sistema ignorar el efecto de rebote presentado en los contactos de los dispositivos de campo. La fig. 1.11 muestra como se lleva a cabo el filtrado de una señal de campo con un tiempo de retardo de 8 milisegundos.

1.2.3.1.4. Memorización de eventos.

El MSE requiere tener la capacidad de memorizar hasta 256 eventos identificados y procesados con sus respectivos parámetros para su análisis por el operador. En memoria secundaria podrá memorizar historias de eventos de hasta 1024 eventos.

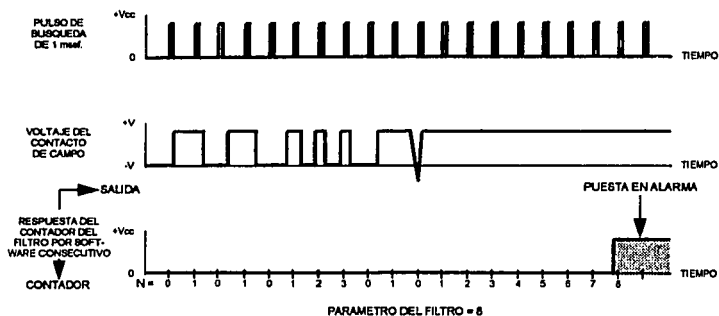


Fig. 1.11. Filtrado de una señal de campo.

1.2.3.1.5. Presentación de eventos.

La información de las secuencias de eventos ocurridas que el MSE desplegará, consistirá de: la descripción del evento, el estado del evento, tarjeta y número del punto donde se identificó el evento, la fecha actual (día, mes y año) de operación, tiempo (hora, minutos, segundos y milisegundos), tipo de contacto asociado, grado de urgencia y palabra de estado.

- + Descripción del evento. Se requiere que la descripción del evento sea de 22 caracteres alfanuméricos.
- + Estado del evento. Este podrá ser especificado como alarma, regreso a normal, deshabilitados y en prueba.
- + Tipo de contacto. Se identificarán solamente dos tipos contactos, normalmente abierto y normalmente cerrado.

- ✦ Grado de urgencia. Indicará la urgencia con la que se deberán atender los eventos registrados y estará dada por grado de atención normal, grado de atención urgente y simplemente información relacionada al evento, dependiendo el estado del evento.

- ✦ La palabra de estado estará dada en 8 caracteres.

Asimismo, se requiere que la presentación en CRT de los eventos se realice en la forma que el usuario lo requiera dependiendo el estado del evento.

De acuerdo a la aparición de los eventos se generará un reporte en impresora y CRT, en forma continua.

1.2.3.1.6. Mantenimiento.

El mantenimiento que se proporcionará al MSE se llevará a cabo a través del mantenimiento preventivo y del mantenimiento correctivo. El primero consistirá en la conservación de la operación funcional del sistema; para lo cual, el usuario requiere que el sistema ejecute una prueba de él mismo. En el caso del mantenimiento correctivo, se llevará a cabo la eliminación de las fallas generadas durante el funcionamiento del MSE.

1.2.3.1.7. Requerimientos funcionales.

En esta sección se establecen los requerimientos específicos de las funciones que deberá llevar a cabo el MSE. Estas funciones se pueden clasificar por un lado en las funciones que se podrán llevar a cabo con autonomía del operador y por otro lado las que se podrán llevar a cabo con la intervención del operador.

Las funciones que deberá ejecutar el MSE con autonomía del operador, son las siguientes:

- ♦ **Inicialización del software y hardware.**
El operador requiere que al momento del encendido del sistema se lleve a cabo la inicialización del hardware y del software necesario para el funcionamiento de éste.
- ♦ **Diagnóstico.**
El operador requiere que al encendido del MSE se efectúe un diagnóstico de las partes electrónicas que lo integran.
- ♦ **Lectura del evento.**
Esta función deberá llevar a cabo la búsqueda de los canales de entrada que detectan la presencia de un evento.
- ♦ **Filtrado del evento.**
Se requiere de un filtrado de las señales de entrada para ignorar efectos de rebotes.
- ♦ **Procesamiento del evento.**
Con esta función el operador requiere que los eventos leídos sean identificados y registrados cronológicamente con toda la información que sea de su interés.
- ♦ **Presentación del evento.**
El operador requiere que los eventos ocurridos se presenten simultáneamente y en forma cronológica a través de un CRT y de una impresora.
- ♦ **Almacenamiento de la información del evento para su posterior manejo.**

Las funciones que el MSE deberá realizar con la intervención y a petición del usuario, son las siguientes:

- **Actualización de la fecha y la hora del sistema.**
Se requiere que la actualización de la fecha y la hora del sistema se pueda llevar a cabo a petición del operador.
- **Solicitud de los últimos 256 eventos registrados.**
Se requiere que el sistema almacene los últimos 256 eventos y que estos puedan ser solicitados a petición del operador para su despliegue en CRT, en impresora o almacenarse en algún dispositivo de memoria (discos duros o diskettes).
- **Estado actual de los canales con respecto a los contactos de campo.**
Se requiere que el sistema presente al usuario el estado de los canales en el momento que éste lo solicite, indicando el tiempo y la fecha en que se realizó dicha petición.
- **Configuración del MSE.**
El usuario requiere que a solicitud de él se presente la configuración del MSE.
- **Personalización del sistema.**
El operador requiere llevar a cabo la inicialización de la descripción de las señales de entrada con la información relacionada a su aplicación particular; así como, la configuración de la información de los eventos a desplegarse.
- **Actualización de la descripción de los eventos.**
El operador requiere que a petición de él se pueda actualizar la descripción de los eventos.

- **Manejo de archivos.**
Se requiere que el sistema proporcione al operador las opciones de leer, almacenar e imprimir algún archivo; así como, la permanencia residente del sistema en memoria.
- **Generación de reportes.**
El sistema deberá tener la capacidad de proporcionar al operador el sumario de alarmas, la lista de leyendas, la lista de estados actuales, la lista de eventos, el reporte del diagnóstico.

1.2.3.2. Especificación eléctrica.

Esta especificación define las características eléctricas necesarias para el MSE.

1.2.3.2.1. Voltaje en los puntos de entrada.

Se requiere que las entradas funcionen con un voltaje directo de 24 V de Corriente Directa. Este voltaje, será suministrado por contactos de campo normalmente abiertos o normalmente cerrados.

1.2.3.2.2. Inmunidad al ruido.

Todos los puntos de entrada deberán ser inmunes al ruido generado por inducción electromagnética. Sus circuitos deberán cumplir con las siguientes características:

- Protección contra inversión de polaridad.
- Rechazo a voltajes transitorios.
- Aislamiento a través de circuitos optoacopladores.

1.2.3.2.3. Alimentación.

El sistema MSE deberá tener la capacidad de operar de manera autónoma al menos durante 30 min. En caso de una falla de suministro de energía eléctrica de C. A. comercial.

1.2.3.2.4. Circuitos de comunicaciones.

La interface de comunicación entre los dispositivos de despliegue y procesamiento se requiere que cumplan con las características eléctricas de acuerdo al estándar EIA RS-232C.

La interface de comunicación con los puertos de la minicomputadora de adquisición de datos en la red de comunicaciones cumplirá las características eléctricas establecidas en el estándar EIA RS-485, el cual especifica las características eléctricas de generadores y receptores que son empleados para el intercambio de señales binarias en interconexiones multipunto de equipos digitales.

Los generadores y receptores que conforman este estándar pueden operar con un voltaje de modo común entre -7 y +7 volts.

1.2.3.3. Especificación mecánica.

Esta especificación define las características físicas con las que contará el MSE.

1.2.3.3.1. Capacidad de puntos de entrada.

El sistema MSE deberá ser suministrado con la capacidad de adquirir 240 señales de entrada. Esta capacidad representa el número máximo de puntos que el sistema es capaz de adquirir en un sólo módulo.

1.2.3.3.2. Dispositivos para el despliegue de eventos.

El MSE, deberá contar con una impresora para el reporte impreso de eventos y un monitor a color para el despliegue visual.

1.2.3.3.3. Gabinete.

El sistema deberá suministrarse con un gabinete estándar, el cual alojará en su interior y permitirá el fácil acceso a todos los elementos que conforman al MSE.

Este sistema estará de acuerdo a los normalizados por la NEMA No. 250 y será del tipo 4X, el cual proporcionará protección al sistema contra corrosión, llovizna, polvo y salpicadura o contacto directo con agua salina.

1.2.3.4. Especificación ambiental.

El sistema funcionará bajo las siguientes condiciones ambientales:

- ♦ Temperatura en el rango de 0 a 50°C
- ♦ Humedad relativa máxima de 95%

Estas especificaciones son las recomendadas por las tarjetas de la SAC y la PC industrial.

CAPITULO 2

DISEÑO Y DESARROLLO DEL MSE

Conocida la especificación del Sistema de Monitoreo y Registro de secuencia de eventos (MSE), en el presente capítulo se lleva a cabo el análisis, diseño y posterior desarrollo de dicho sistema.

2.1. ANÁLISIS DEL SISTEMA.

Tomando como referencia las características del sistema MSE a desarrollar, se establece lo siguiente:

- Las principales funciones del MSE, son:
 - Detección de los eventos en las entradas del MSE.
 - Identificación y registro de los eventos detectados de manera cronológica.
 - Presentación al operador de la secuencia cronológica de los eventos.
 - Almacenamiento de la secuencia cronológica.

- Los eventos estarán proporcionados por los contactos normalmente cerrados (NC) o normalmente abiertos (NA) de los dispositivos de campo de cualquier proceso industrial (tales como, interruptores de nivel, de presión, de velocidad, de temperatura, etc.).

- La capacidad máxima de puntos de entrada del MSE deberá ser de 240.

- Los eventos que se presentarán en los contactos de campo y que deberán ser detectados por el MSE, podrán ser, en forma secuencial (siguiendo una secuencia de ocurrencia de eventos, ocasionados por algún disturbio del proceso en planta), o bien, en forma aleatoria (con la ocurrencia de eventos por separados, que no siguen una secuencia).

- El MSE deberá realizar la eliminación de rebotes de contacto, así como, el aislamiento de las entradas con el campo.

- El tiempo de resolución entre dos eventos deberá ser de un milisegundo.

- El reconocimiento de los eventos deberá ser realizado por muestreo continuo ("polling" en inglés) es decir, se deberá llevar a cabo cada determinado tiempo la revisión continua de los puntos de entrada del MSE, para la verificación de la ocurrencia

de algún evento y por señales de interrupción que generarán los propios eventos al ocurrir.

- Se procesará y registrará la información proporcionada por el usuario para la configuración del sistema; es decir, la configuración se podrá llevar a cabo en campo.
- El medio de transmisión de información entre los módulos que forman al MSE deberán cumplir con el estándar EIA RS-232C.
- La interfaz hombre-máquina deberá realizarse en base a menús.
- Los dispositivos empleados para el despliegue de las secuencias de eventos serán, CRT e impresora.
- Se considera que el sistema se podrá conectar a sistemas de alto nivel, sistemas de adquisición de datos y control supervisorio (SCADA), sistemas de control distribuido (DCS) y controladores lógicos programables (PLC) a través de su interacción con una red de comunicaciones. Esta consideración no esta contenida en este trabajo de tesis.
- El sistema tendrá una capacidad de memorización de 1024 eventos, para la generación de reportes.
- El sistema deberá realizar un diagnóstico en las partes electrónicas que lo integran.

La fig. 2.1 muestra el modelo conceptual de los que es el sistema de Monitoreo y Registro de secuencia de evento y la fig. 2.2 muestra el enfoque usuario de dicho modelo.

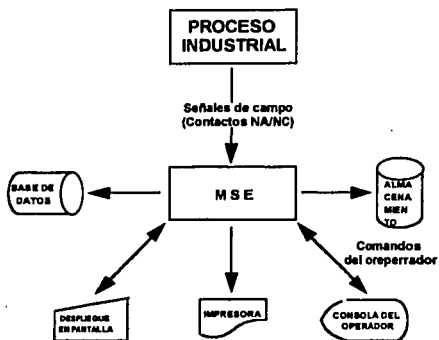


Fig. 2. 1. BOSQUEJO DEL MODELO CONCEPTUAL

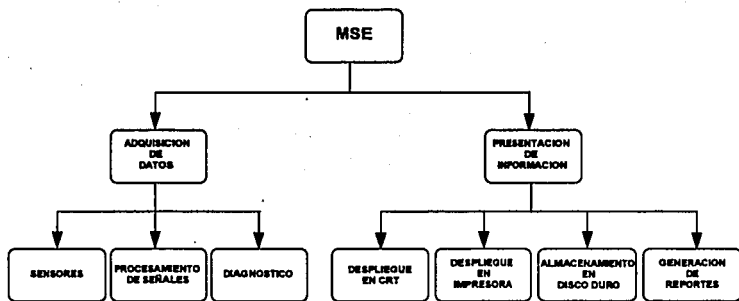


Fig. 2.2. Descomposición del modelo conceptual (enfoque usuario).

2.2. DISEÑO DEL MSE.

2.2.1. INTRODUCCIÓN.

2.2.1.1. Propósito del diseño.

A continuación, se establece la forma de estructurar el sistema MSE para satisfacer los requerimientos del sistema identificados en el capítulo anterior. Para lo cual, el diseño del MSE se describe a través de cuatro vistas:

- Descripción de la descomposición
- Descripción de las dependencias
- Descripción de interfaces
- Descripción detallada

En la primera se describe la división del sistema dentro de entidades de diseño, en la segunda, las relaciones que existen entre dichas entidades y los recursos del sistema. En la descripción de interfaces se presenta todo lo necesario para conocer el correcto funcionamiento de las entidades a diseñar. Y por último se hace una descripción detallada del diseño interno de las entidades.

2.2.1.2. Alcance del MSE.

Dadas las características generales identificadas del MSE, este será un equipo modular de multicoprocesamiento de alta resolución. Su operación comienza con la supervisión continua del estado lógico de las señales provenientes de los dispositivos de campo (como interruptores límite, relevadores, etc.), relacionados con los equipos de interés; de manera que, al detectarse un cambio de estado lógico en alguno de éstos, lo que se considera como la ocurrencia de un evento, se efectúa el registro del mismo con un identificador, fecha y tiempo en que se detectó, a su vez, se lleva a cabo su almacenamiento cronológico y presentación al operador en tiempo real. Posteriormente, de la información almacenada pueden obtenerse reportes específicos como sumario de alarmas y eventos en un lapso de tiempo definido, entre otros. De esta forma, se facilita el

análisis de algún disturbio en el proceso.

La fig. 2.3 presenta en forma general el concepto del sistema a desarrollar.

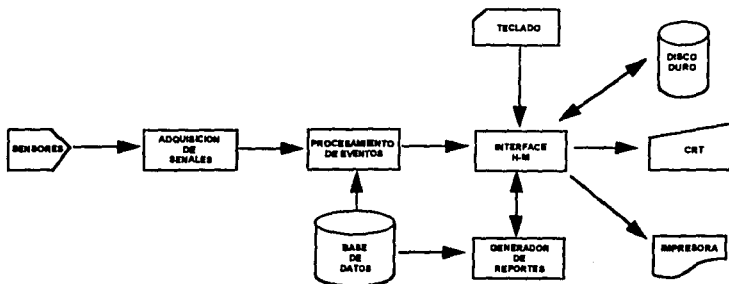


Fig. 2.3 Bosquejo Conceptual del sistema.

Desde el punto de vista físico, se considera dos opciones de configuración del prototipo del MSE:

1. En forma individual.
2. En Red.

Funcionalmente ambas configuraciones están divididas en dos subsistemas, el Subsistema de Adquisición y el Subsistema de Presentación (Fig. 2.4). A su vez cada uno de estos estará compuesto por equipos electrónicos programables ("Hardware") y la programación correspondiente ("Software"). En el primer subsistema, se adquirirá el estado lógico de contactos basado en el multicoprocesamiento. Se detectará la secuencia de eventos ocurrida, se procesará cada evento, se almacenará y será transmitido al segundo subsistema para que posteriormente, éste lo almacene, despliegue y registre. En el segundo subsistema, la información adquirida del estado de los contactos de campo se

acondiciona para su presentación impresa en forma de listados o en pantalla de video a color.

Para la configuración en red el subsistema de presentación se puede enlazar con sistemas de presentación remotos centralizados para aplicaciones de SCADA's, DCS y PLC's. El diseño y desarrollo de esta configuración se encuentra fuera del alcance del presente trabajo.

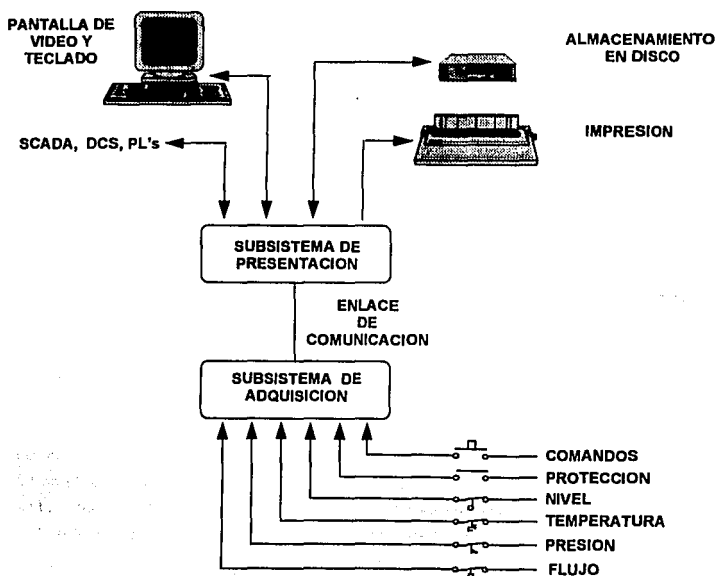


Fig. 2.4 Configuración básica del MSE.

2.2.1.3. Definiciones y Acrónimos.

MSE	Sistema de Monitoreo y Registro de Secuencia de Eventos.
SCADA	Sistema de Adquisición de Datos y Control Supervisorio.
DCS	Sistema de Control Distribuido.
PLC	Controlador Lógico Programable.

2.2.2. REFERENCIAS.

El diseño del sistema está basado en el siguiente estándar.

Institución	Estándar
INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS (IEEE)	IEEE Std 1016-1987 Recommended Practice for Software Design Descriptions.

2.2.3. DESCRIPCIÓN DE LA DESCOMPOSICIÓN.

La descripción de la descomposición proporciona la división en entidades de diseño* (módulos) del sistema en su totalidad; así como, la forma en que el sistema es estructurado, el propósito y funcionamiento de cada entidad. Además, para cada entidad se proporciona una referencia a la descripción detallada para la identificación de atributos**.

* Entidad de diseño. Es un elemento o componente de un sistema funcional y estructuralmente distinto a los otros que componen el sistema; así mismo, es nombrado y referenciado de distinta forma.
** Atributo de entidad. Es una característica o propiedad que proporciona información de una entidad de diseño.

2.2.3.1. Descomposición en módulos.

El Sistema de Monitoreo y Registro de Secuencia de Eventos MSE es un equipo modular, funcionalmente se divide en, **Subsistema de Adquisición** y **Subsistema de Presentación**, a su vez cada uno de estos módulos tiene su propia arquitectura de "Hardware" y "Software"; los cuales, se describen más adelante. La arquitectura de hardware de ambos subsistemas esta constituida por elementos electrónicos ya desarrollados, únicamente se configurarán para poder implementar el MSE. A diferencia de la arquitectura de software, la cual se desarrollará para ambos subsistemas y de las cuales se mencionan sus atributos.

2.2.3.1.1. Descripción del Subsistema de Adquisición.

En el Subsistema de adquisición se adquiere el estado lógico de los contactos del proceso a monitorear. Se detecta la secuencia de eventos ocurrida, tanto para alarma como para retorno a normal de hasta 240 puntos. Cada secuencia se procesa y transmite al Subsistema de Presentación para que, posteriormente, éste la almacene, despliegue y registre con una resolución de un milisegundo.

A continuación se describe la arquitectura de "Hardware" y "Software" del Subsistema de Adquisición.

ARQUITECTURA DE HARDWARE

Dado que el requerimiento de adquirir y reconocer la secuencia en el peor de los casos es de 240 eventos con un tiempo de resolución de un milisegundo entre cada uno de ellos, físicamente el subsistema de adquisición se divide en: **unidad de adquisición** y **unidad de procesamiento** (ambas constituidas por elementos electrónicos que integran la "línea SAC" desarrollada por el IIE).

Unidad de adquisición

La unidad de adquisición se integrará con 15 tarjetas de adquisición de señales binarias SAC-421, las cuales adquirirán y reconocerán independientemente 16 eventos cada una, comunicando por medio de señales de interrupción la presencia de algún evento válido para su posterior atención.

Las tarjetas SAC-421 contienen la inteligencia necesaria para realizar la adquisición de los eventos, determinando cuando ha ocurrido un evento real e invalidando los producidos por señales de ruido. Tienen 16 entradas digitales optoacopladas y un dispositivo microcontrolador para realizar las funciones de adquisición, reconocimiento de los eventos e intercambio de información con la unidad de procesamiento del subsistema. Estas tarjetas estarán colocadas en las ranuras dedicadas a los módulos de aplicación del controlador "SAC IBUS-III" y alambradas a una de las líneas de señales de interrupción disponibles en el IBUS-III. La figura 2.5 presenta la configuración de la unidad de adquisición con éstas.

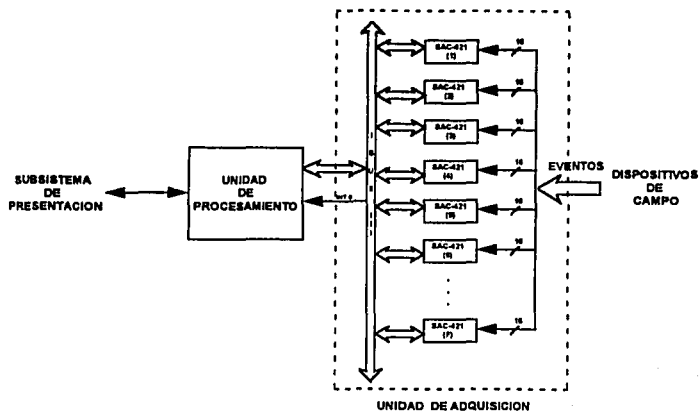


Fig. 2.5. Unidad de Adquisición del Subsistema de Adquisición

Unidad de procesamiento

Esta unidad se ocupara de procesar, registrar temporalmente y transmitir los eventos ya procesados al subsistema de presentación. Básicamente esta dividida en los submódulos de, procesamiento, almacenamiento y comunicación.

Submódulo de Procesamiento.

Este es el módulo principal de la unidad de procesamiento y cuenta con la inteligencia para realizar la identificación, asignación de tiempo de aparición y estado lógico de los eventos, reconocidos por la unidad de adquisición. Está constituido por la tarjeta procesadora "SAC-1887", la cual contiene los elementos para realizar estas tareas, además de aquellos que le permiten generar las bases de tiempo y manejo de interrupciones; éstos últimos son:

a) Reloj en tiempo real.

Dentro de este módulo se genera la base de tiempo para el reloj-calendario de tiempo real, con el cual se actualizan los registros dedicados a los años, meses, días, horas, minutos y segundos, así como para la asignación de los milisegundos de aparición de los eventos. Esto requiere configurar el Reloj/Contador Programable (PTC, Programmable Timer/Counter) 8254 de tarjeta "SAC-1887" de la siguiente forma:

Para realizar el conteo en intervalos de milisegundos se deberá suministrar a la entrada del contador 2 (CLK2) una señal de onda cuadrada con frecuencia de 1Khz. Esta frecuencia se obtendrá alambrando la entrada del contador 1 (CLK1), ubicada en el punto de conexión B de W6 al punto A de éste mismo, el cual contiene la programación del contador 1.

Como se requiere estar actualizando la base de tiempo cada segundo, el PTC 8254 realizará una interrupción cada segundo.

Para ello se alambra la salida de contador 1 (punto de conexión B en W5) a la entrada del contador 2 y su salida a la entrada IRQ7 del controlador de interrupciones (ver fig. 2.6).

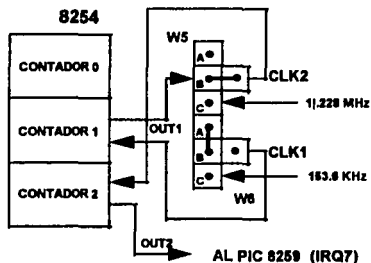


Fig. 2.6. Reloj en tiempo real del MSE.

b) Interrupciones.

El módulo de procesamiento estará configurado para el manejo de señales de interrupción, las cuales demandarán una atención inmediata. Como se ha mencionado anteriormente, los módulos de entrada generarán señales de interrupción a la ocurrencia de un evento. También, el PTC 8254 generará una señal de interrupción para la actualización del tiempo cada segundo, por lo que las señales de interrupción manejadas por el MSE estarán configuradas en orden jerárquico en el controlador de interrupciones programable (PIC) 8259 de la tarjeta SAC-1887 como se muestra en la tabla 2.1.

Esta tarjeta actúa como maestro único en el controlador "SAC IBUS-III" y

estará colocada en la ranura dedicada a los módulos maestros.

En el PIC 8259	Del dispositivo generador de la señal de interrupción
IR0	INT0 de las tarjetas de entradas digitales SAC-421
IR1	INT CLK1 del contador 2 del 8254 (Sistema Operativo)
IR2	INT RX de la tarjeta SAC-821
IR3	Disponible
IR4	INT RX de la tarjeta SX-233
IR5	Disponible
IR6	Disponible
IR7	INT CLK2 del contador 2 del 8254

Tabla 2.1. Señales de interrupción en el PIC-8259.

Submódulo de Almacenamiento.

En éste se realizará el registro interno de los eventos con los parámetros de tiempo, número de señal y estado lógico. La tarjeta de Expansión de Memoria "SAC-0929" proporciona los medios para esta función. Esta tarjeta estará configurada con memoria de acceso aleatorio ("RAM", en inglés) y se ubicará en la ranura 1 (0 en hexadecimal) de los módulos de aplicación en el controlador SAC IBUS-III.

Submódulo de comunicación.

Este módulo constituye el medio por el cual se realiza la interfaz entre las unidades de procesamiento y presentación. Está basado en la tarjeta de comunicaciones "SX-233", la cual contiene dos canales de comunicación que cumplen con el estándar eléctrico RS-232C. La figura 2.7 presenta la configuración de esta unidad con los módulos descritos.

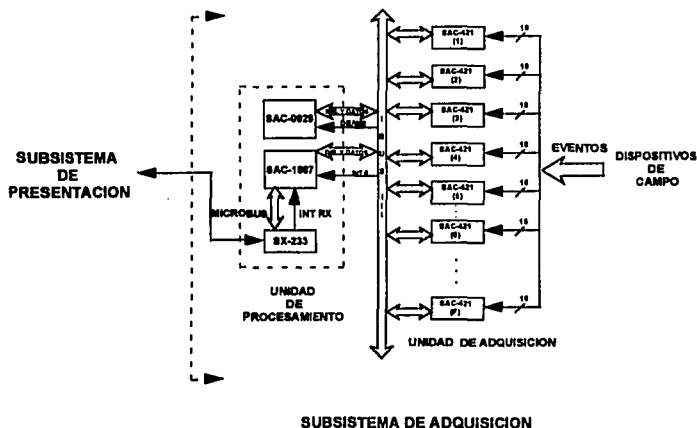


Fig. 2.7. Módulos de la unidad de procesamiento.

El esquema de interfaz que adoptará este submódulo con el subsistema de presentación y el submódulo de procesamiento será el siguiente:

Con el subsistema de presentación:

- Transmisión asíncrona dúplex (half-duplex).
- Velocidad de transmisión a 9600 "bits" por segundo.
- El formato para un carácter de transmisión estará compuesto de 1 "bit" de arranque, 8 "bits" de información, 0 "bits" de paridad y 1 "bit" de paro.

Con el submódulo de proceso:

- Transmisión de información a través del Microbus contenido en ambas tarjetas.
- Comunicación al módulo de proceso a través de señal de interrupción.

En suma, la arquitectura de hardware del subsistema de adquisición estará compuesta por los siguientes elementos de la línea SAC.

- A. Controlador programable de la línea SAC IBUS-III
- B. SAC-1887 Tarjeta de procesamiento central.
- C. SX-233 Tarjeta de comunicación serial RS-232.
- D. SAC-821 Tarjeta de comunicación de red local RS-485 (Sólo para la configuración en red del MSE).
- E. SAC-929 Tarjeta de expansión de memoria para almacenamiento.
- F. SAC-421 Tarjeta de adquisición de señales binarias.

Las figuras 2.8 y 2.9 presentan el arreglo de estos módulos en el controlador SAC IBUS-III, para la configuración individual y en red del MSE respectivamente.

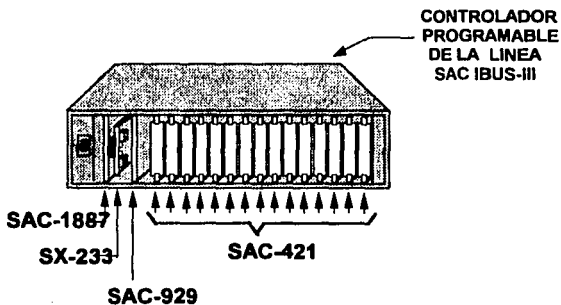


Fig. 2.8. SUBSISTEMA DE ADQUISICIÓN PARA LA CONFIGURACIÓN INDIVIDUAL DEL MSE.

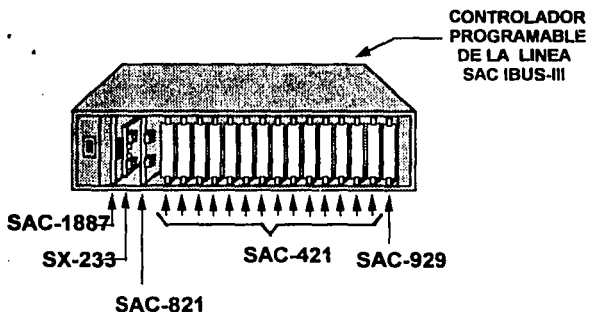


Fig. 2.9. SUBSISTEMA DE ADQUISICIÓN PARA LA CONFIGURACIÓN EN RED DEL MSE .

ARQUITECTURA DE SOFTWARE

En el subsistema de adquisición, se llevará a cabo la adquisición de las señales de campo (a través de los contactos de campo), para su posterior identificación (tarjeta y señal exacta en la que se presentó un evento), su etiquetación (la fecha y hora en que se detectó dicho evento) y por último el envío de la información completa del evento que ocurrió al subsistema de presentación.

El software del subsistema de adquisición básicamente llevará a cabo las siguientes funciones:

- ① Inicialización de los dispositivos de Hardware necesarios para la transmisión y recepción de información; así como, la asignación de variables y parámetros en memoria RAM de los valores o datos necesarios para su operación.
- ② Filtrado de los eventos, validando los reales e invalidando aquellos que sean producidos por ruido.
- ③ Diagnóstico de los componentes de la arquitectura de hardware del subsistema de adquisición.
- ④ Adquisición o lectura del estado de los eventos.
- ⑤ Empaquetamiento de la información para su posterior envío al subsistema de presentación.
- ⑥ Transmisión de la información del estado de los eventos hacia el subsistema de presentación.
- ⑦ Reloj calendario en tiempo real para etiquetar a los eventos con la fecha y hora en que ocurrieron.

- ① Recepción de información del Subsistema de Presentación, para la actualización del tiempo y fecha del subsistema de adquisición y solicitudes de diagnóstico.

La fig. 2.9 muestra la arquitectura de software del subsistema de adquisición basada en las funciones de dicho subsistema.

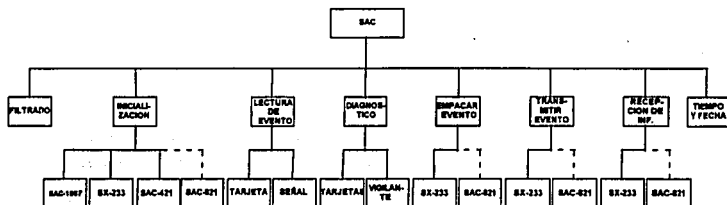


Fig. 2.9. Arquitectura del Software del Subsistema de Adquisición.

2.2.3.1.2. Descripción del Subsistema de Presentación.

En el Subsistema de Adquisición se adquiere el estado lógico de los contactos del proceso a monitorear. Se detecta la secuencia de eventos ocurrida, se procesa y transmite al Subsistema de Presentación, el cual, representará la interface principal, con la cual, el operador interactuará con el sistema. En este subsistema se realizará el registro, almacenamiento y presentación visual y escrita de los eventos transmitidos a éste por el subsistema de Adquisición. Además, realizará la transmisión y recepción de información que corresponda a las funciones de configuración del propio sistema y funciones especiales que se llevarán a cabo a petición del usuario.

A continuación se describe la arquitectura de "Hardware" y "Software" del Subsistema de Presentación.

ARQUITECTURA DE HARDWARE

Básicamente el hardware del Subsistema de Presentación estará compuesto por una Computadora Personal y una impresora.

Computadora Personal

Esta estará compuesta por una PC/AT que contenga:

- Monitor a color para el despliegue visual de los eventos.
- Teclado alfanumérico para la introducción de comandos y datos del operador.
- Unidad de disco duro para el almacenamiento masivo de información.
- 2 Puertos serie (COM1 y COM2) para la interface con el Subsistema de Adquisición.
- 1 Puerto paralelo (LPT1) para la interface con la impresora.

Impresora

Con ésta se realizará la presentación de los eventos en forma impresa, por lo que se requiere de una impresora de matriz de puntos.

La fig. 2.10 muestra el arreglo físico del MSE con los módulos de las unidades antes definidas.

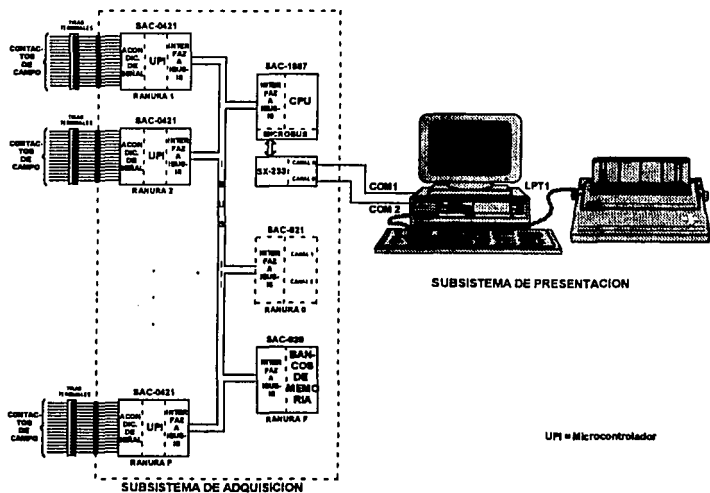


Fig. 2.10 Arreglo físico de los módulos del MSE

La figura 2.11 muestra la arquitectura completa del MSE.

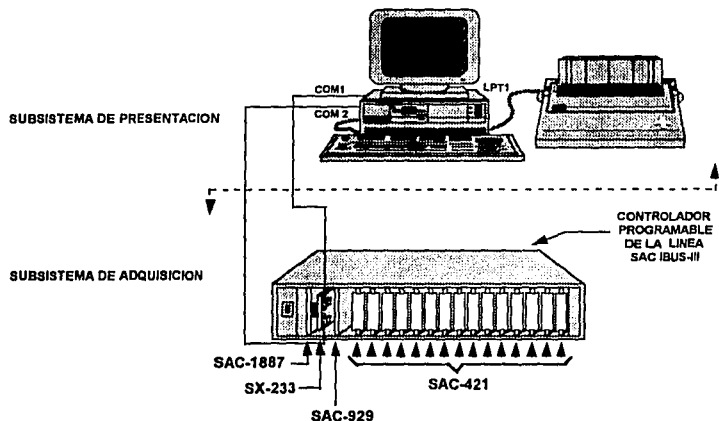


Fig. 2.11. Arquitectura del MSE .

ARQUITECTURA DE SOFTWARE

Esencialmente el software del Subsistema de Presentación llevará a cabo las siguientes funciones.

- ① Inicialización automática de los dispositivos de hardware, así como las variables y parámetros en memoria RAM necesarios para su operación.
- ② Inicialización del tiempo y la fecha del Subsistema de Adquisición.
- ③ Obtendrá la configuración del Subsistema de Adquisición a petición del usuario.

- ④ Realizará la recepción de la información enviada por el subsistema de adquisición, y la procesará, almacenará y presentará a través del monitor de la PC e impresora,
- ⑤ Obtendrá los últimos 256 eventos ocurridos y almacenados en el Subsistema de Adquisición, esto a petición del usuario.
- ⑥ Desplegará los últimos 1024 eventos almacenados en el Subsistema de Presentación, es decir en el disco duro de la PC, a petición del usuario.
- ⑦ Además realizará una serie de funciones que se ejecutarán a petición del operador del sistema, como son: manejo de archivos de usuario, configuración de la presentación de la información, actualización de la base de datos que contiene la información relativa a cada señal y generación de reportes.

La fig. 2.12 muestra la arquitectura de software del Subsistema de Presentación basada en las funciones de dicho subsistema.

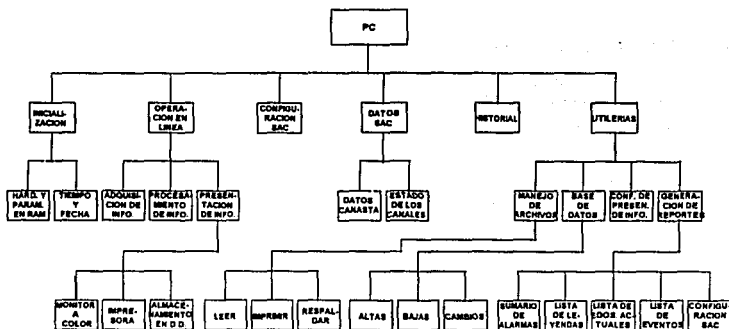


Fig. 2.12. Arquitectura de Software del Subsistema de Presentación.

2.2.4. DESCRIPCIÓN DE LAS DEPENDENCIAS.

Como ya se mencionó anteriormente, para la implementación del MSE únicamente se desarrollará el software de los subsistemas que lo constituyen, dado que los componente de hardware ya están desarrollados y solamente se configurarán de manera adecuada para el desarrollo del sistema en su totalidad.

Esta vista de diseño define la forma en que interactúan las entidades de diseño y proporciona la información necesaria para facilitar la percepción del cómo, por qué, dónde, y a que nivel del sistema se llevarán a cabo las funciones del sistema. Además, especifica el tipo de relación que existe entre las entidades tales como envío de información, descripción del orden de ejecución, o bien las interfaces de los parámetros definidos.

2.2.4.1. Dependencias entre módulos.

Funcionalmente, el sistema MSE se realizará en base a una metodología modular; constituida básicamente por:

- El módulo de adquisición.
- El módulo de procesamiento.
- El módulo de la interfaz.

En el primer módulo se llevará a cabo la inicialización del hardware del subsistema de adquisición y la adquisición del estado de las señales de entrada provenientes de los dispositivos de campo cada milisegundo, para lo que se emplearán las técnicas de muestreo continuo y de interrupción, también se llevará a cabo un filtrado de cada señal por "hardware" para eliminar los "rebotes" de los contactos.

En el segundo se recibirán los eventos reconocidos por el módulo de adquisición para su registro y almacenamiento adicionando su identificador de, tiempo y fecha de ocurrencia (el almacenamiento podrá ser de hasta 240 eventos en un buffer circular); posteriormente se efectuará el envío al módulo de la interfaz.

Por último, el módulo de la interfaz se refiere a la interfaz hombre-máquina en la cual se recibirá la información del módulo de procesamiento, se efectuará el almacenamiento masivo (de hasta 1024 eventos) y se presentará al usuario.

El módulo de adquisición y procesamiento se llevan a cabo en el subsistema de adquisición y el de la interfaz en el subsistema de presentación.

Además, el sistema también estará integrado por otros módulos de programación que se desarrollarán en el subsistema de presentación y que proporcionarán al usuario la posibilidad de interactuar de forma personalizada con el sistema:

- Módulo de inicialización.**
- Módulo de solicitud de información.**
- Módulo de configuración.**
- Módulo manejador de archivos.**
- Módulo manejador de base de datos.**
- Módulo de presentación de información.**
- Módulo de generación de reporte.**

El primer módulo estará destinado a establecer las condiciones de inicio de los parámetros de tiempo y fecha del subsistema de adquisición.

El módulo de solicitud de información, proporcionará al usuario la posibilidad de solicitar los eventos almacenados por el módulo de procesamiento, así como el estado actual de los contactos de campo acoplados al subsistema de adquisición.

El módulo de configuración efectuará una verificación de los componentes físicos (tarjetas electrónicas) presentes del subsistema de adquisición.

El módulo manejador de archivos permitirá al usuario la manipulación de sus archivos independientes al sistema MSE.

El módulo manejador de base de datos permitirá al usuario personalizar el MSE de acuerdo a sus necesidades particulares de aplicación, ofreciendo además la posibilidad de

efectuar posteriores modificaciones y adiciones al mismo. Será a través de este módulo que el operador pueda definir o modificar la base de datos, habilitar o deshabilitar señales de entrada y definir los campos de información para los reportes.

El módulo de presentación de información permitirá configurar los campos de información que serán desplegados o bien empleados para la generación de reportes.

Por último, el módulo de generación de reportes permitirá al usuario seleccionar el almacenamiento masivo que se realiza en el módulo de interfaz a determinados eventos de acuerdo a parámetros especificados por él mismo

Estos últimos se desarrollarán dentro del subsistema de presentación y se ejecutarán a petición del usuario.

En el diagrama de la fig. 2.13 muestra en forma general las dependencias entre los módulos que constituyen el MSE.

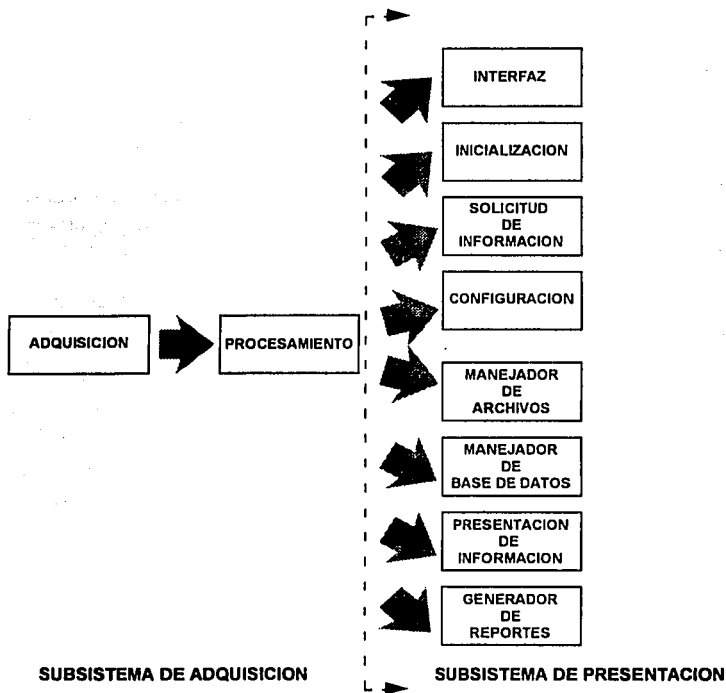


Fig. 2.13. Dependencia de los módulos de programación del sistema MSE.

Se ha mencionado anteriormente, que la programación que corresponde al subsistema de adquisición se llevará a cabo en un ambiente de programación multitareas, por lo que los módulos de programación que se llevarán a cabo en éste, se diseñarán a

través de diversas tareas y funciones. En el caso del subsistema de presentación la programación se diseñarán a través de funciones.

2.2.4.2. Dependencias entre las funciones de cada módulo.

Las funciones que básicamente llevará a cabo el módulo de adquisición se agrupan en dos etapas, inicialización y adquisición. Ahora bien, las tareas y/o funciones que se diseñarán para la etapa de inicialización son:

1) Inicialización del subsistema de adquisición.

El propósito de esta función consistirá en programar para su adecuada operación la los siguientes circuitos programables del subsistema de adquisición.

☞ La tarjeta procesadora SAC-1887.

☞ La tarjeta de comunicaciones SX-233.

☞ La tarjeta de entradas digitales con interrupción SAC-421.

Así como, la inicialización de parámetros para el correcto funcionamiento del subsistema de adquisición.

2) Diagnóstico del hardware.

El propósito de esta función será el de verificar el correcto funcionamiento del hardware del subsistema de adquisición (tarjetas, conectores, RAM, ROM y vigilante).

3) Inicialización de datos.

El propósito de esta tarea será el llevar a cabo un muestreo de datos (en el tiempo cero) obtenidos con las tarjetas SAC-421 del MSE. Así como el arranque de las tareas que se efectúan en el módulo de procesamiento, y las de refresco de vigilante e inicialización de comunicaciones con el subsistema de presentación.

En la fig. 2.14, se presenta el diagrama de flujo de datos de esta etapa del módulo de adquisición.

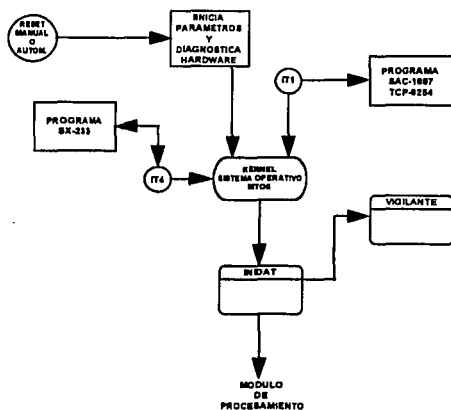


Fig. 2.14 Etapa de inicialización del Módulo de Adquisición.

Las tareas y/o funciones que se llevarán a cabo en la etapa de adquisición del módulo de adquisición son:

1) Adquisición del estado de las señales.

Al detectarse un cambio de estado en las tarjetas de entrada del MSE (SAC-421), estas generarán una interrupción a la SAC-1887 solicitando atención inmediata a través de una tarea que da servicio a dicha interrupción (SERVINT), la cual identificará el tiempo (HH:MM:SS.mm) y fecha (DD/MM/AA) de la ocurrencia del evento, así como la tarjeta en que se presentó el evento y los almacenará en una cola circular de almacenamiento (EVENTOS), para el posterior manejo y manipulación de dicha información. La fig. 2.15 presenta el diagrama de flujo de datos de la etapa de adquisición

Envío de eventos al subsistema de presentación.

Esta tarea (PC_EVEN), deberá recuperar los eventos almacenados en la cola circular POINTER y los filtrará colocando aquellos que sean válidos en un área de memoria (PC) para ser enviados a la PC del subsistema de presentación, así como el almacenamiento simultáneo de éstos en otra cola circular (BUFFER), cuya información será empleada únicamente a petición del usuario; es decir, si el valor de la variable PETICION cambia a 1, el contenido de BUFFER se pondrá en una área de memoria (PET_USR) para ser enviados a la PC. El cambio del valor de PETICION se llevará a cabo desde el subsistema de presentación a través del puerto B de la SX-233. Por último, será aquí donde se llevará a cabo el envío de la secuencia de eventos a la PC.

La Fig. 2.16 presenta el diagrama de flujo de datos del módulo de procesamiento.

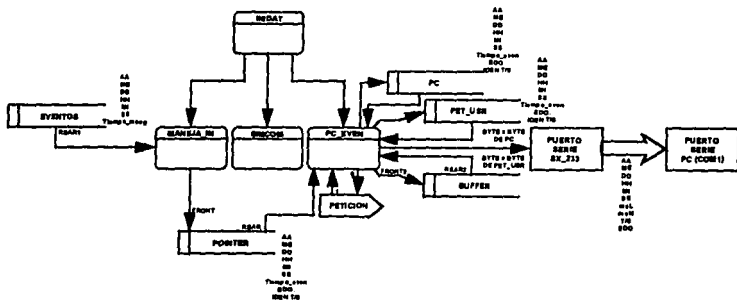


Fig. 2.16. Módulo de procesamiento.

Las funciones que se llevarán a cabo en el módulo de interfaz son:

1) Inicialización del hardware del subsistema de presentación.

Con esta función se inicializará de manera automática el hardware del subsistema de presentación (puertos de comunicación serie de la PC, COM1 y COM2); así como, la asignación de valores o datos necesarios a las variables y parámetros en memoria RAM.

2) Recepción de información.

El subsistema de adquisición generará una interrupción al microprocesador de la PC a través del puerto serie COM1 indicándole que enviará información de los eventos ocurridos, en ese momento, esta función recibirá y pondrá la información en un área de memoria (BUFFER_PC y BUFFER1_PC) para su posterior utilización.

3) Formateo de los eventos.

Esta función acondicionará la información de los eventos recibida para llevar a cabo el registro, despliegue impreso y almacenamiento de dichos eventos. La información de los eventos será proporcionada en base a la configuración de presentación de información dada por el usuario, ya que, la base de datos contiene las características de las señales de campo, que fueron dadas por el usuario con anterioridad.

4) Despliegue de los eventos.

Ya que se tenga la información de los eventos formateada de la manera que el usuario desee, esta función desplegará los eventos a través del monitor de la PC.

5) Almacenamiento de los eventos.

La función de almacenamiento realizará el almacenamiento en disco duro de toda la información relacionada con los eventos ocurridos, para emplearla posteriormente en la generación de reportes.

f) Impresión de los eventos.

Esta función llevará a cabo la impresión de la misma información en el caso de que la impresora se encuentre lista para imprimir, de lo contrario se indicará al operador.

Estas tres últimas funciones se realizarán de manera simultánea.

La fig. 2.17 muestra el diagrama de flujo de datos del módulo de interfaz, indicando las principales funciones que llevará a cabo.

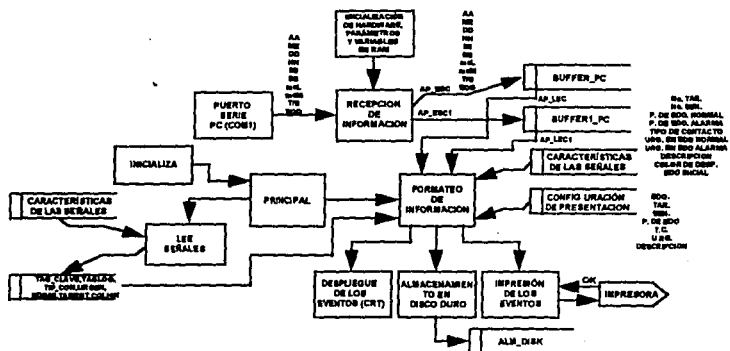


Fig. 2.17. Módulo de la interfaz.

Las funciones que permitirán al usuario interactuar con el sistema de forma personalizada se llevan a cabo en los otros módulos de programación mencionados anteriormente, y las cuales se mencionan a continuación.

Funciones del Módulo de inicialización.

1) Inicialización de tiempo y fecha.

Con esta función se podrá actualizar el tiempo y la fecha que contiene el subsistema de adquisición para la etiquetación de los eventos. Se llevará a cabo a petición del usuario.

Funciones del Módulo de solicitud de información.

1) Petición de la información contenida en el subsistema de adquisición.

Esta función se ejecutará a petición del usuario, y realiza una petición al subsistema de adquisición de la información almacenada en este (es decir de los últimos 256 eventos ocurridos), así como la petición del estado actual de los contactos de campo.

Funciones del Módulo de configuración.

1) Petición de la configuración actual del controlador SAC IBUS-III.

Esta función permitirá al usuario solicitar la manera en que esta conformada el controlador SAC del subsistema de adquisición.

Funciones del Módulo manejador de archivos.

1) Lectura de un archivo de usuario.

Esta función proporcionará al usuario la posibilidad de realizar la lectura de algún archivo especificado por él mismo.

1) Impresión de un archivo de usuario.

Esta función proporcionará al usuario la posibilidad de realizar la impresión de algún archivo especificado por él mismo.

1) Respaldo de un archivo de usuario.

Esta función proporcionará al usuario la posibilidad de realizar el respaldo de algún archivo especificado por él mismo.

Funciones del Módulo manejador de base de datos.

1) Adición de señales de campo a la base de datos.

Con esta función el usuario podrá inicializar la base de datos que contendrá toda la información relacionada con las señales de campo conectadas al subsistema de adquisición (tarjeta, señal, tipo de contacto, urgencia de atención para el estado de alarma y de regreso a normal, palabra del estado en normal y regreso a normal, tipo de contacto, identificador de la señal, descripción de la señal y el color en que se desea desplegar el evento).

2) Eliminación de señales de campo de la base de datos.

Con esta función el usuario podrá borrar la información de señales dadas de alta en la base de datos con la función anterior.

3) Actualización de señales de campo de la base de datos.

Con esta función el usuario podrá cambiar información específica de alguna señal dada de alta en la base de datos.

Funciones del Módulo de presentación de información.

1) Configuración de la presentación de información.

Esta función permitirá al usuario la posibilidad de seleccionar los diversos campos de información que desee sean desplegados para los eventos ocurridos; así mismo, se seleccionan los campos de información de los reportes a generar.

Funciones del Módulo generador de reportes.

1) Generación de reportes.

Esta función generará diferentes reportes a petición del usuario, considerando diversos parámetros seleccionados por él mismo (Sumario de eventos, lista de leyendas, lista de estados actuales de los canales, lista de eventos y configuración actual del controlador SAC IBUS-III).

1) Presentación de reportes.

Esta función, presentará al usuario el reporte generado, a través del monitor de la PC o la impresora según sea su elección, o bien realizará el respaldo del reporte generado.

El diagrama de flujo de datos de la fig. 2.18 muestra los módulos de programación que permitirán al usuario del sistema interactuar con el sistema.

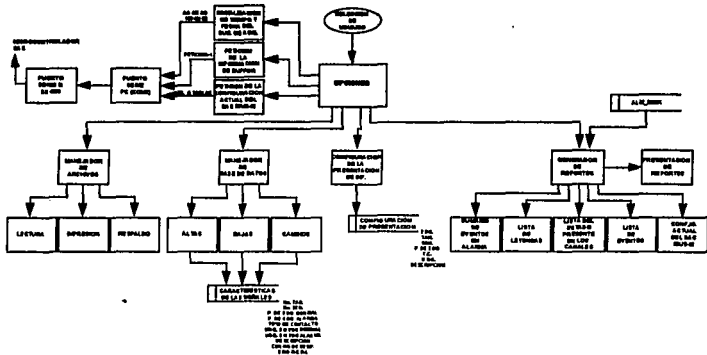


Fig. 2.18. Módulos de programación de funciones especiales.

El diagrama de la figura 2.19 muestra la dependencia entre las funciones de los módulos de programación que constituyen el sistema en su totalidad.

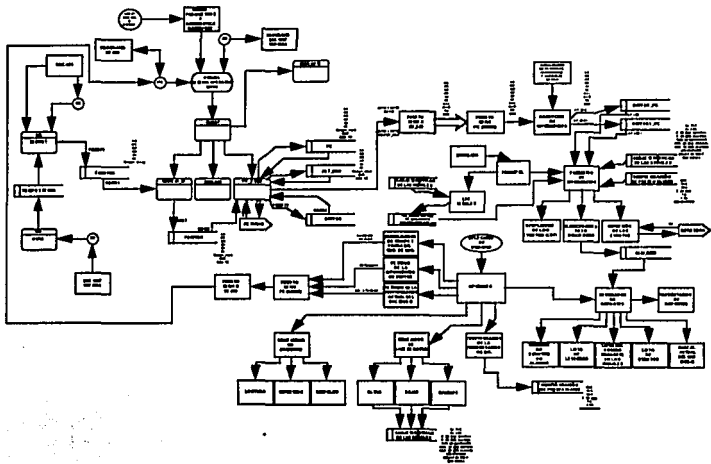


Fig. 2.19. Dependencia entre las Funciones del MSE.

2.2.5. DESCRIPCIÓN DE LA INTERFAZ.

En esta sección se proporciona lo necesario para conocer el uso correcto de las tareas y/o funciones provistas para cada módulo de programación; es decir, se describirán las entradas válidas y salidas esperadas de cada módulo, así como los formatos de las ventanas en caso de ser necesarios. Esto con el propósito de definir la base para la descripción de diseño detallado de cada función o tarea.

En base a la descripción de la dependencia de los módulos de programación, primeramente se describirán de manera general las entradas y salidas de todos los módulos de programación y posteriormente se describirá el diccionario de datos detallado de las estructuras de datos y variables utilizadas en las tareas y/o funciones de los módulos de programación, así como, el catálogo de las funciones que serán desarrolladas. Por último se establecerá el formato de las ventanas a presentar al operador.

2.2.5.1. Interfaces de los módulos.

La tabla 2.1, presenta de manera muy general las entradas y salidas que se relacionan con cada módulo de programación, esto en base a la descripción de las dependencias entre los módulos de programación dada en la sección anterior.

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Modulo de programación	Entradas	Salidas
Módulo de adquisición	Ocurrencia de eventos en las tarjetas SAC-421. Tiempo y fecha. Reset Manual o automático.	Activación a las tareas del módulo de procesamiento. Eventos con el tiempo, fecha y tarjeta en que ocurrieron.
Módulo de procesamiento	Eventos con el tiempo, fecha y tarjeta en que ocurrieron.	Secuencia de eventos ocurridos, con su identificador de T/S, tiempo, fecha y estado para su almacenamiento en BUFFER y su envío a PC.
Módulo de interfaz	Secuencia de eventos ocurrida, con su identificador de T/S, tiempo, fecha y estado. Estado de la Impresora. Características de las señales. Configuración de la presentación.	Información formateada de los eventos ocurridos para su despliegue, almacenamiento e impresión.
Módulo de inicialización	Nueva fecha y hora del Subsistema de Adquisición.	Fecha y hora en bytes.
Módulo de solicitud de información	Activación de la petición.	Hasta los últimos 256 eventos almacenados en el modulo de procesamiento.
Módulo de configuración	Activación de la petición de la configuración del subsistema de adquisición.	Configuración actual del controlador SAC IBUS-III.
Módulo manejador de archivos	Selección de operación a realizar. Nombre del archivo a manipular.	Indicador de terminación de la operación seleccionada.

Tabla 2.2. Entradas y salidas de los módulos de programación.

Módulo de programación	Entradas	Salidas
Módulo manejador de base de datos	Selección de operación a realizar. Información de las señales de campo.	Indicador de terminación de almacenamiento de la información de las señales de campo actualizada.
Módulo de presentación de información	Campos seleccionados para el despliegue de información.	Registro de los campos seleccionados.
Módulo generador de reportes	Selección del reporte a generar. Parámetros a considerar en la generación de reportes. Selección de la manera de presentar los reportes.	Indicador de reporte generado.

Continuación Tabla 2.2. Entradas y salidas de los módulos de programación.

2.2.5.2. Diccionario de datos.

El diccionario de datos, contiene la documentación de las estructuras de datos y variables que se emplearán en la descripción del diseño detallado de las tareas y/o funciones de los módulos de programación que integran el sistema MSE, en base a la sección 2.2.4.2.

El contenido del diccionario de datos se divide en dos grupos, el primero consta de las variables utilizadas en las funciones y tareas de los módulos de programación que se desarrollarán para el subsistema de adquisición (SAC) del MSE - módulo de adquisición y procesamiento - y el segundo grupo estará integrado por las variables utilizadas en las funciones de los módulos de programación que se desarrollarán para el subsistema de presentación (PC) - interfaz, inicialización, solicitud de información, configuración, manejador de archivos, manejador de base de datos, presentación de información y generador de reportes -.

En el apéndice A se encuentra el diccionario de datos completo de las estructuras de datos y variables empleadas en la descripción del diseño detallado del software del MSE, conteniendo el nombre, tipo, descripción y la función o tarea que emplea a cada una de éstas.

2.2.5.3. Catálogo de funciones.

Este catálogo contiene la descripción de las funciones que se desarrollarán para la implementación de los módulos de programación que constituye el sistema, basado en la dependencia que existe entre las funciones de cada módulo (sección 2.2.4.2)

El catálogo de funciones se encuentra documentado en el apéndice A y contiene el nombre, el tipo y una breve descripción de las funciones, así como las funciones y/o tareas que las emplean.

Las funciones se agrupan de la siguiente manera: primeramente, se describen las funciones que se desarrollarán para la implementación de los módulos de programación que constituyen el subsistema de adquisición (SAC) - adquisición y procesamiento - y a continuación se describen las funciones utilizadas por los módulos del subsistema de presentación (PC) - interfaz, inicialización, solicitud de información, configuración, manejador de archivos, manejador de base de datos, presentación de información y generador de reportes -.

2.2.5.4. Formato de ventanas.

En esta última sección de la descripción de las interfaces, se muestra el formato de las ventanas que se presentarán al usuario. Estos formatos se originaron en base a la norma establecida por la Comisión Federal de Electricidad (CFE) para monitores de secuencia de eventos y por requerimiento de usuarios del MSE en la planta de ciclo combinado de Gómez Palacio, Durango.

La figura 20 muestra la ventana principal del MSE, en donde se efectuará el despliegue de los eventos y se encontrará activo el menú de opciones principal. Las figuras de la 21 a la 28 muestran las ventanas que se presentarán al usuario, conforme éste haga la petición de ejecutar las funciones del MSE.



Fig. 20 Ventana Inicial con el menú principal.

La figura 21 muestra el menú que se activará al ejecutar la petición de los datos del módulo de la SAC, la figura 22 presenta el menú que se activará al seleccionar la opción de utilerias y en la figura 23 se encuentra la ventana que se presentará al ejecutar la función que permite al usuario configurar la información de los eventos que requiere sea desplegada.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

DATOS DE LA CANASTA

1. **DATOS CANASTA**
2. **ESTADO LOGICO DE LOS CANALES**

SELECCIONE UNA OPCION [ESC] - SALIR

Fig. 21. Ventana con el Menú de Petición de datos del módulo SAC

MONITOR DE SECUENCIA DE EVENTOS

ARCHIVO	BASE DATOS	PRES. DE INF.	REPORTES
---------	------------	---------------	----------

SELECCIONE UNA OPCION [ESC] - SALIR

Fig.22. Ventana con el Menú de Utilerías.

ESTADO EN 2025 2024
AUTORIZADO AL 30 JUNIO

El menú de la figura 27 contiene los distintos reportes a generar, dependiendo la selección del usuario. Y por último, la figura 28 muestra los parámetros que el usuario podrá seleccionar para realizar el reporte de Lista de Eventos.

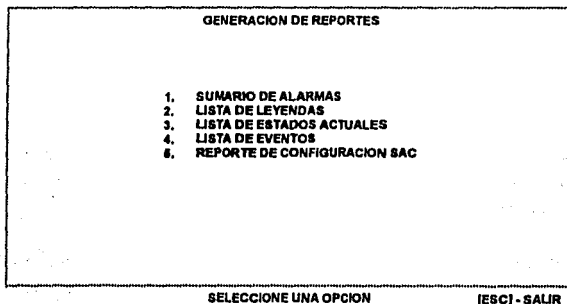


Fig.27. Ventana con el menú de Generación de Reportes.

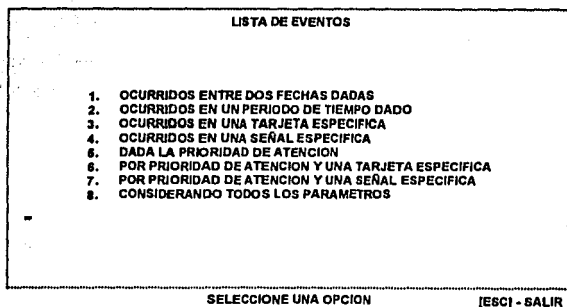


Fig.28. Parámetros a considerar para la generación del reporte de Lista de eventos

2.2.6. DESCRIPCIÓN DEL DISEÑO DETALLADO.

La descripción de diseño detallado, contiene los detalles internos de cada entidad de diseño (tareas y funciones) que constituyen el software del sistema, para su posterior desarrollo.

En esta sección, se describe el diseño detallado de las tareas y/o funciones que componen el software de los subsistemas - adquisición (SAC) y presentación (PC) - que integran el sistema; ésto se lleva a cabo mencionando: el nombre o identificador, el propósito y las entradas y salidas esperadas para cada función. El funcionamiento interno de cada una de éstas, se aprecia mejor en el pseudocódigo correspondiente a cada función, incluido en el apéndice B.

El nombre de las funciones hace referencia a las descritas en la sección 2.2.4.2, en donde se describieron las dependencias existentes entre las funciones de cada módulo de programación.

2.2.6.1. Diseño detallado de las funciones que integran el software del subsistema de adquisición (SAC)

En seguida, se describe el diseño de tallado de las funciones básicas que constituyen el software del subsistema de adquisición (SAC) y cuyo pseudocódigo se incluye en el apéndice B.

Nombre: SAC_MSE()
Descripción: Es el programa principal que ejecuta la inicialización de parámetros y diagnóstico de hardware del subsistema de adquisición al encender el MSE.
Entradas: Encendido del sistema.
Salidas: Arranca la tarea INIDAT()

Nombre: Principia()
Descripción: Esta función inicializa el diagnóstico de las tarjetas SAC-421 y activa la adquisición continua de señales digitales, habilitadas por una máscara de muestreo (con la que se seleccionan los canales de campo que se requieren muestrear).
Entradas: La dirección de la tarjeta
Salidas: Resultado del diagnóstico y presencia del conector inicial. Activación de la adquisición continua.

Nombre: diag_mse()
Descripción: Esta rutina activa el diagnóstico de las tarjetas de entradas digitales del MSE.
Se diagnostica:
- Sección de procesamiento (UPI)
- Interfaz a IBUS-III
- Las 16 entradas digitales de las tarjetas
- El conector de campo
Entradas: Dirección de la tarjeta a diagnosticar.
Salidas: Resultado del diagnóstico de las SAC-421 y presencia o ausencia del conector de campo.

Nombre: INIDAT()
Descripción: Esta tarea sube el primer muestreo de datos obtenidos para las tarjetas del monitor de secuencias de eventos. Estos datos se consideran ocurren en el tiempo cero del arranque del sistema.
Entradas: Datos de inicialización dat_ini[]
Dirección hardware dir_dw[]
Tabla de identificadores tab_iden
Salidas: Datos de inicialización de las tarjetas de monitor de secuencia de eventos que se enviarán a la PC.

Nombre: SERVINT()

Descripción: Al detectarse un evento en las entradas de las tarjetas SAC 421, estas generarán una interrupción a la tarjeta SAC 1887 indicando que requieren atención inmediata, esta tarea atenderá a estas interrupciones llamando a la función que realizará la lectura de las tarjetas de entrada para localizar a aquella que haya generado la interrupción.

Entradas: Mascara de muestreo para el PIC

Salidas: Activación de la función ReadCard421()

Nombre: ReadCard_421()

Descripción: Esta función es llamada directamente por la tarea SERVINT(), realiza la identificación de la tarjeta SAC-421 que generó la interrupción a la SAC-1887. Etiqueta con tiempo (hasta los ms) y fecha el evento ocurrido y lo almacena en EVENTOS.

Entradas: hora, minuto, segund, dia, mes, year, front1, rear1, dir_hw,

Salidas: Coloca la información de los eventos ocurridos en la cola de almacenamiento EVENTOS (tiemp, HORA, MINUTO, SEGUND, DIA, MES, YEAR, ranura, bytelow y bytehight).

Nombre: lee_8254.

Descripción: Esta función lee el tiempo del evento en mseg. La cuenta del tiempo se lleva en el contador 2 del temporizador 8254 de la SAC-1887.

Entradas: Dirección del puerto de control del 8254.

Salidas: tiempo_seg.

Nombre: hora()

Descripción: Esta tarea actualiza las variables globales de tiempo y hora para etiquetar algún evento ocurrido antes de ser almacenado en EVENTOS.

Entradas: HORA, MINUTO, SEGUND, DIA, MES y YEAR.

Salidas: HORA, MINUTO, SEGUND, DIA, MES y YEAR actualizadas.

- Nombre:** Maneja_int()
- Descripción:** Esta tarea se encarga de recuperar los eventos almacenados en la cola de EVENTOS dejados por la rutina de servicio de interrupción, los procesa obteniendo el tiempo del evento, el estado de la señal y su identificador (Tarjeta/Señal). Todos los datos son dejados en una cola que será procesada por la tarea de comunicaciones y enviada a la PC. Todas las señales que cambien con sus respectivos rebotes estarán almacenadas en esta cola.
- Entradas:** FRONT1, REAR1, EVENTOS.
- Salidas:** Coloca la información de los eventos en la cola de almacenamiento POINTER (HORA, MINUTO, SEGUND, DIA, MES, YEAR, tiempo_evento, estado e identificador).
-
- Nombre:** PC_EVEN()
- Descripción:** Esta tarea se encarga de recuperar los eventos almacenados en la cola POINTER y colocarlos en una área de memoria para ser enviados a la PC, así como la programación del puerto del DUART A de la SX-233, para el envío de información al subsistema de presentación.
- Entradas:** POINTER.
- Salidas:** PC.
-
- Nombre:** almacena_buffer()
- Descripción:** Esta función se encarga de almacenar cada segundo los eventos ocurridos en ese lapso de tiempo en la cola BUFFER. La capacidad máxima de BUFFER es de 256 eventos.
- Entradas:** POINTER
- Salidas:** BUFFER

Nombre: lee_buffer()
Descripción: Esta función recupera los eventos almacenados en la cola BUFFER y los coloca en un área de memoria para ser enviado al subsistema de presentación a petición del usuario.
Entradas: BUFFER.
Salidas: PET_USR.

Nombre: envia()
Descripción: Esta función realiza el envío de información contenida en el área de memoria PC byte por byte al subsistema de presentación, indicando a éste el inicio y fin de la información con un encabezado y fin de la trama enviada.
Entradas: Dirección inicial del área de memoria de PC en DataPtr y el tamaño de la trama a enviar en NumEvents.
Salidas: Envío byte por byte de la información por el DUART a de la tarjeta SX-233.

Nombre: envia1()
Descripción: Esta función realiza el envío de información contenida en el área de memoria PET_USR byte por byte al subsistema de presentación, indicando a éste el inicio y fin de la información con un encabezado y fin de la trama enviada.
Entradas: Dirección inicial del área de memoria de PET_USR en DataPtr y el tamaño de la trama a enviar en NumEvents.
Salidas: Envío de la información byte por byte por el DUART A de la tarjeta SX-233.

Nombre: SIMCOM()
Descripción: Esta tarea enciende y apaga el led A de la tarjeta SX-233 para indicar que el MSE esta funcionando.

2.2.6.2. Diseño detallado de las funciones que integran el software del subsistema de presentación (PC)

El diseño detallado de las funciones básicas que constituyen el software del subsistema de presentación (PC), es el siguiente:

Nombre: Principal()
Descripción: Función principal del subsistema de presentación, el cual da inicio al procesamiento en línea del sistema.
Entradas: ap_lec, ap_es.
Salidas: Despliegue de eventos y activación del menú principal

Nombre: Menú_principal()
Descripción: Activa las funciones del menú principal a petición del usuario.
Entradas: opción.
Salidas: Llamado a las funciones que ejecutan las opciones del menú principal.

Nombre: inicialización()
Descripción: Esta función lleva a cabo la programación del puerto serie COM1 de la PC para activar la recepción de información del subsistema de adquisición a través de interrupciones al microprocesador del CPU de la PC.
Entradas: Dirección de hardware del puerto serie COM1.
Salidas: Ninguna.

Nombre: Inl_fecha.
Descripción: Solicita al usuario la nueva fecha y hora del sistema y lo transmite por el puerto serie COM" al subsistema de adquisición.
Entradas: Hora_u, minuto_u, segundo_u, dia_u, mes_u y año_u.
Salidas: La nueva hora y fecha en hexadecimal.

Nombre: pet_buffer().
Descripción: Ejecuta la petición de hasta los últimos 256 eventos almacenados en el BUFFER del subsistema de adquisición , o bien, del estado actual de los contactos de campo y los presenta en pantalla; además, de almacenarlos en un archivo de disco dado por el usuario.
Entradas: selección, name_arch.
Salidas: El contenido de BUFFER o el estado actual de los contactos de campo.

Nombre: util().
Descripción: Esta función activa el menú de utilerías y ejecuta las funciones correspondiente a petición del usuario.
Entradas: opción.
Salidas: Llamado a las funciones que ejecutan las funciones del menú de utilerías.

Nombre: archivo().
Descripción: Esta función activa el menú manejador de archivo y ejecutas las funciones correspondientes dependiendo la opción seleccionada.
Entradas: opción.
Salidas: Llamado a las funciones que ejecutan las funciones del menú de utilerías.

Nombre: se_base().
Descripción: Esta función activa el menú manejador de base de datos y ejecuta las funciones correspondientes dependiendo la opción seleccionada.
Entradas: opción.
Salidas: Llamado a las funciones que ejecutan las funciones del menú de utilerías.

Nombre: reportes()
Descripción: Esta función habilita el menú de generación de reportes, en el que se hace el llamado a las funciones que generan cada reporte.
Entradas: opción.
Salidas: Reportes generados.

Nombre: opc_rep4().
Descripción: Esta función habilita el menú de selección de los parámetros a considerar para la generación del reporte de lista de eventos y hace el llamado a las funciones que generan dicho reporte.
Entradas: opción.
Salidas: Reporte generado.

2.2.7. DESCRIPCIÓN DE LA PROGRAMACIÓN OPERATIVA.

La función de un sistema operativo es tomar el control de la ejecución de programas del usuario y realizar la interfaz entre el "hardware" del sistema y dichos programas de aplicación. Por lo que, a continuación se describen los Sistemas Operativos bajo los cuales se desarrollará el software del MSE.

2.2.7.1. Descripción del Sistema Operativo en el Subsistema de Adquisición.

El sistema operativo que permite la interfaz con el hardware de los módulos SAC es el desarrollado por IPI, Co. denominado "MTOS" (Multi-tasking Operating System), el cual, es un sistema operativo multitareas, con capacidad de manejar multiprocesamiento y operación en tiempo real. Entre sus principales características, se tienen:

- ♦ Manejo de hasta 2056 tareas.
- ♦ Soporte para la programación en los lenguajes de programación PLM, C y ensamblador.
- ♦ Esquema de sincronización de tareas basados en banderas de

eventos, buzones, semáforos, creación y eliminación de tareas dinámicas, manejo de memoria, servicios discretos, etc., y otros servicios relacionados con temporizadores y soporte de coprocesador.

2.2.7.2. Descripción del Sistema Operativo en el Subsistema de Presentación.

El desarrollo de los programas de aplicación residentes en la PC se llevará a cabo bajo el ambiente del sistema operativo MS-DOS ("Micro-soft Disk Operating System", en inglés), el cual permite la comunicación con la unidad de procesamiento de datos (CPU), CRT, unidades de disco, puertos serie y paralelo, teclado y otros periféricos en la computadora personal, además de proporcionar el soporte para la programación en lenguajes de alto nivel como el "C", Pascal, Ensamblador y otros.

2.3. DESARROLLO DEL MSE.

En esta sección se describe la forma en que se desarrolló el software del MSE, considerando los recursos de programación con que se contaba y el diseño de éste, descrito en la sección anterior.

2.3.1. DESARROLLO DE LOS MÓDULOS DE PROGRAMACIÓN DEL MSE.

Primeramente se definió el ambiente de desarrollo, los lenguajes de programación y el sistema operativo de cada subsistema para la construcción de los módulos de programación (ver fig. 2.13) tal como se muestra en la tabla 2.3.

Subsistema de adquisición	Subsistema de presentación
Ambiente de desarrollo Intel	Ambiente de desarrollo Turbo C
Lenguajes Ensamblador y C	Lenguaje C
Sistema Operativo MTOS	Sistema operativo MS-DOS
(Multi Task Operating System)	(Micro-soft Disk Operating System)

Tabla 2.3 Ambientes de programación de los Subsistemas del MSE

Una vez definidos los ambientes de desarrollo para cada subsistema, y considerando el pseudocódigo de las tareas y/o funciones particulares de cada uno de éstos, dados en el diseño detallado de la sección anterior, se procedió a codificar cada función o tarea y por último, se establecieron pruebas funcionales particulares a cada uno de ellos.

Posteriormente, se realizaron verificaciones preliminares de funcionamiento de cada componente de forma independiente; después, se integraron todos los módulos para cada subsistema separadamente y se verificó su funcionamiento respecto a lo esperado. Por último, se integró totalmente el sistema MSE y se verificó cada módulo funcional respecto a los especificado en el capítulo 1.

2.3.2. GENERACIÓN DE VERSIONES DEL SOFTWARE DEL SUBSISTEMA DE ADQUISICIÓN.

En el diagrama de flujo de la figura 2.29, se muestra el procedimiento que se llevó a cabo para la generación de versiones del MSE del subsistema de adquisición, el cual, se describe a continuación.

El pseudocódigo, se refiere al descrito en la sección anterior para cada función y/o tarea que integra los módulos de programación, a partir de los cuales se llevó a cabo la codificación en lenguaje C y en ensamblador. Si al ejecutar la compilación de dichas funciones o tareas por separado se detectaban errores se hacían las correcciones necesarias al código.

Cuando no existían errores al compilar las funciones o tareas, se ejecutaba LINK86 para enlazar todos las funciones y tareas codificadas, y el código generado se depositaba en el programa MTOS86.LNK, el cual no tiene una dirección de memoria específica. Posteriormente se ejecutaba LOC86 para localizar el código y generar el programa MSE.SYS y se reserva un área de memoria para dicho código.

Posteriormente, mediante OH86.EXE el contenido del archivo MSE.SYS se convierte a código hexadecimal y se almacena en MSE.HEX (archivo con un máximo de 1MB), el cual mediante CONV02.EXE creaba los segmentos de memoria de 64 Kb para almacenar los archivos EXXXX.HEX y FXXXX.HEX, de código ejecutable del subsistema de adquisición.

Por último, se ejecutaba HEX.OBJ para convertir el contenido de cada segmento de memoria (E y F) a código binario y colocarlo en los archivos MSE_E.ROM y MSE_F.ROM. Estos dos últimos archivos contienen el código a grabar en las memorias de la tarjeta SAC-1887 destinadas a la aplicación del MSE.

El grabado de las memorias EEPROM de la SAC-1887 se llevó a cabo empleando el software para programación de memorias EMP.

2.3.3. GENERACIÓN DE VERSIONES DEL SOFTWARE DEL SUBSISTEMA DE PRESENTACIÓN.

Al igual que para el subsistema de adquisición, los módulos de programación del de presentación se codificaron basándose en el pseudocódigo descrito para cada función y se compilaron por separado para corregir los errores. Posteriormente se empleó un proyecto para enlazar todas las funciones y generar la versión ejecutable del software del Subsistema de presentación.

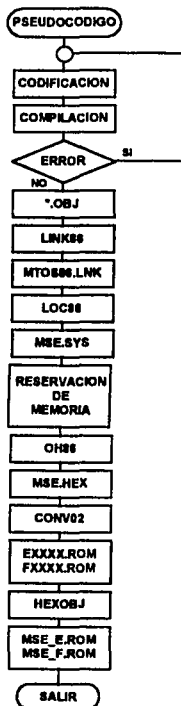


Fig. 2.29. Generación de versiones de software para la SAC.

CAPITULO 3

PRUEBAS DEL MSE

Este capítulo contempla un programa de pruebas a diferentes niveles del MSE, para su aplicación en el prototipo del sistema.

3.1. PROGRAMA DE PRUEBAS.

3.1.1. INTRODUCCIÓN.

Este programa toma como referencia las normas de ingeniería descritas en la especificación de requerimientos del sistema. Para la parte funcional y de integridad se establece un protocolo personalizado de acuerdo a las necesidades del proyecto.

3.1.2. OBJETIVOS.

Los objetivos de este programa de pruebas son:

- ❑ Identificar las características del sistema MSE que serán probados para cada subsistema.
- ❑ Identificar los procedimientos y actividades requeridas para realizar las pruebas.
- ❑ Definir las herramientas de prueba y el ambiente para realizarlas.

3.1.3. ALCANCE.

Este programa establece las pruebas básicas asociadas con los módulos principales que integran los subsistema del MSE.

3.1.4. METODOLOGÍA DE PRUEBAS.

La metodología empleada para realizar las pruebas del MSE es la conocida como "de abajo hacia arriba" (down-top, en Inglés); la cual, propone la prueba e integración desde los módulos jerárquicamente inferiores hasta la integración y comprobación de todo el sistema.

3.1.5. CARACTERÍSTICAS DEL SISTEMA QUE FUERON COMPROBADAS.

↪ En el subsistema de adquisición se comprobará:

- ① La adquisición de su capacidad nominal (240 señales binarias).
- ② Respuesta en tiempo; es decir, la resolución de 1ms entre dos o más eventos.
- ③ La transmisión y recepción de información con el subsistema de presentación.

- ④ Almacenamiento de hasta los últimos 256 eventos

⇔ En el subsistema de presentación:

- ① La recepción y transmisión de información con el subsistema de adquisición.
- ② El correcto despliegue de la información de los eventos ocurridos.
- ③ El almacenamiento de los últimos 1024 eventos en disco y verificación de consistencia de información.
- ④ El despliegue e impresión simultánea de los eventos ocurridos.
- ⑤ Los despliegues de los menús de opciones y operación de las funciones que se realizarán a petición del usuario.

3.1.6. MÓDULOS DEL SISTEMA QUE NO SERÁN PROBADOS.

Las pruebas respecto a las especificaciones eléctricas del subsistema de adquisición (línea SAC) no se llevarán a cabo en esta sección; dado que, ya fueron realizadas por el Departamento de Electrónica del Instituto de Investigaciones Eléctricas IIE.

Además, la computadora personal y la impresora que integran el subsistema de presentación no serán probados por tratarse de equipos comerciales previamente probados y garantizados por el fabricante.

3.1.7. AMBIENTE DE PRUEBAS.

••• Equipo. Las pruebas serán realizadas considerando los siguientes equipos:

- ✓ Una canasta SAC conteniendo los módulos de adquisición, procesamiento y comunicaciones del MSE.
- ✓ La PC e impresora que forman parte del subsistema de presentación del MSE.
- ✓ Una PC auxiliar para la verificación de las comunicaciones.

- ✓ Una canasta SAC con los módulos que conforman el probador, o generador de rebotes, que simula el comportamiento de los contactos de campo (24 volts).
 - ✓ Una PC que forma parte del probador.
 - ✓ Un analizador de estados lógicos para la verificación de señales de interrupción (reloj, interrupción de los módulos de adquisición).
- **Herramientas de programación. Se requieren los siguientes programas herramienta y de aplicación.**
- ✓ En los módulos SAC y PC la programación que conforman las tareas y funciones del MSE.
 - ✓ En la PC auxiliar, el programa XTALK para ejercitar el protocolo de comunicación SAC-PC, el programa DEBUGGER de MTOS para la verificación del procesamiento de los datos de prueba en el sistema operativo MTOS.
 - ✓ En la PC del probador, el software del generador de rebotes con el cual se podrán comprobar características del MSE, como son el procesamiento de una avalancha de eventos, resolución entre eventos y eliminación de rebotes consecutivos de eventos.

3.2. PROTOCÓLO DE PRUEBAS DEL SISTEMA.

La aplicación de un protocolo de pruebas tiene como objetivo la validación de la funcionalidad de un sistema, de acuerdo a su especificación y diseño.

3.2.1. ALCANCE.

El protocolo de pruebas para el MSE, contempla los procedimientos para la validación de las pruebas propuestas y a su vez para la funcionalidad del sistema.

3.2.2. PRUEBAS FUNCIONALES REALIZADAS AL SUBSISTEMA DE ADQUISICIÓN DEL MSE.

3.2.2.1. Prueba de adquisición de la capacidad nominal de los módulos de entrada.

+ Alcance.

Esta prueba tiene como objetivo verificar que los módulos de entrada SAC-421 adquieren los estados de eventos de prueba, los filtran y generan la señal de interrupción correspondiente vía IBUS-III a la unidad de procesamiento. Además, verificar que la capacidad de los módulos de entrada es de 240 señales binarias.

+ Equipo.

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - 15 tarjeta de entradas digitales con interrupción SAC-421.
- Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.
- Un analizador de estados lógicos.

+ Procedimiento.

- En la figura 3.1 se muestra la manera en que se conectó el equipo para la realización de esta prueba. Para conectar las señales de entrada de la tarjeta SAC-421 al analizador de estados lógicos, y dado que éstas tienen un voltaje de 20 - 24 volts, fue necesario conectar divisores de voltaje a dichas entrada. La señal del reloj del MTOS y la de interrupción INT0 de la SAC-1887 se conectan directamente, dado que son señales de 5 volts.

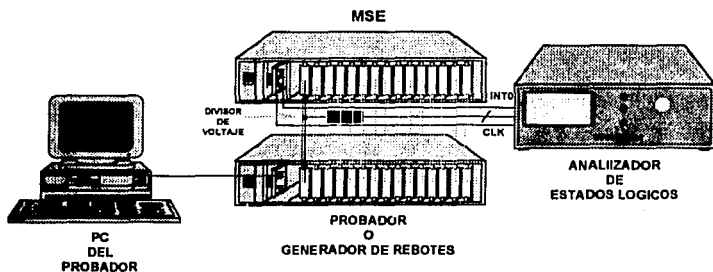


Fig. 3.1. Conexión de equipo para prueba de los módulos de entrada.

- En el canal 1 del analizador de estados lógicos se concentró el reloj del MTOS, en el canal 2 la línea de INTO y en otro se conectaron uno a uno los 240 canales de entrada del MSE, para verificar que al ocurrir un evento en cada uno de los canales, las tarjetas SAC-421 efectúan el filtrado y activan la interrupción
- Se aplicaron señales en las entradas de los módulos a través del probador con tiempo de activación de 1 mseg.

✦ **Resultados.**

- Al generarse un cambio de estado en el canal 0 de la tarjeta 0 del MSE, en el analizador de estados lógicos se presentó la gráfica que se muestra en la figura 3.2, en la que se observa que después de 416 μ s de que es generado un cambio de estado en el canal 0, se presenta la señal de interrupción INTO. Este procedimiento se realizó para todos los canales de entradas.

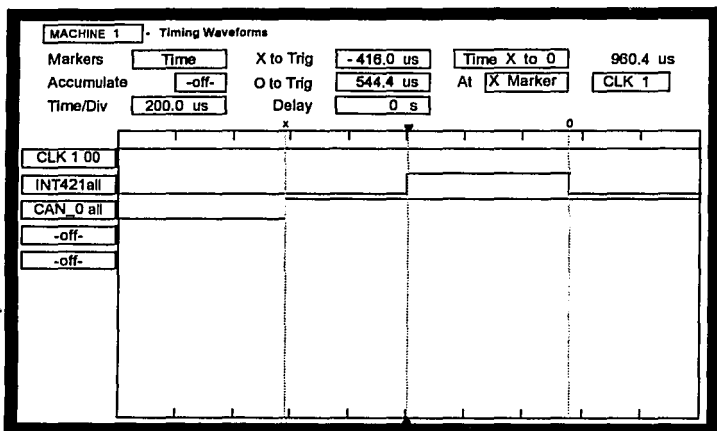


Fig. 3.2. Gráfica del analizador de estados lógicos.

- Una vez hecho lo anterior, para cada canal de entrada, se determina que el MSE puede tener hasta 240 puntos de entrada, pues al haber un cambio de estado en algún canal de las 15 tarjetas SAC-421, estas efectúan el filtrado correspondiente de la señal y activan la interrupción al microprocesador de la tarjeta SAC-1887.

3.2.2.2. Pruebas de la resolución de 1 mseg. del sistema.

- ✦ Alcance.

Estas prueba tienen como objetivo verificar que el sistema tiene una resolución de 1 mseg., es decir el tiempo mínimo para determinar la secuencia entre dos eventos distintos es 1 mseg.

✦ **Equipo.**

- El equipo empleado para la realización de esta prueba es el mismo que se utilizó en la prueba anterior (Ver sección 3.2.2.1), a diferencia de que para la realización de estas pruebas únicamente se empleará una tarjeta SAC-421.

✦ **Procedimiento.**

- El equipo se conectó de igual forma que para la prueba 3.2.2.1 (Ver fig. 3.1).
- En el canal 1 del analizador de estados lógicos se conectó el reloj MTOS, en el canal 2 la línea de INTO del la tarjeta SAC-1887 y en los siguientes, se fueron conectando los canales de la tarjeta SAC-421, incrementando de uno en uno para efectuar las pruebas.
- Se aplicaron señales en las entradas del módulo (SAC-421) a través del probador con tiempo de activación de 1 mseg.
- La primera prueba consiste en la medición del tiempo que transcurre entre la aparición de un evento (canal 0) y la generación de la interrupción de atención a dicha señal por la tarjeta SAC- 421. Posteriormente, se efectuará la medición del tiempo de procesamiento del evento, es decir, a partir del momento en que es identificada la interrupción de la SAC-421 hasta que es finalizado el procesamiento de dicho evento (etiquetación con tiempo y fecha, ubicación y almacenamiento en BUFFER de dicho evento).
Por último se comprobará que el tiempo desde que se presenta el evento (la señal del canal 0) hasta que se obtiene el estado de éste es de 1 mseg. o menor.
- La segunda prueba consiste de la introducción al analizador de estados lógicos de los eventos ocurridos en 3 canales de la tarjeta SAC-421, los cuales son generados con el probador y verificar que el tiempo entre la ocurrencia de un evento y otro es de un mseg. o menor; para lo cual, deberán aparecer en el analizador 3 escalones (correspondientes a los distintos canales) y 3 pulsos de interrupción (donde cada pulso corresponde a la interrupción generada por la tarjeta SAC-421 solicitando atención inmediata para el procesamiento de dicho evento).

+ Resultados.

- Los resultados obtenidos en la primera prueba se describen a continuación y se pueden confirmar en la gráfica de la figura 3.4 .
- Al introducir una señal al canal 0, el tiempo que transcurre antes de que sea identificado por la tarjeta SAC-421 es de $416 \mu\text{s}$ (X to Trig), a partir de este momento, el tiempo de procesamiento de dicho evento es de $544 \mu\text{s}$ (O to Trig) y el tiempo total desde que se presenta el evento hasta que finaliza su procesamiento es de $960.4 \mu\text{s}$. Con lo anterior se demuestra que la resolución del MSE es de 1 mseg. Esto se puede ver en la gráfica del analizador de estados lógicos que se muestra en la figura 3.3.

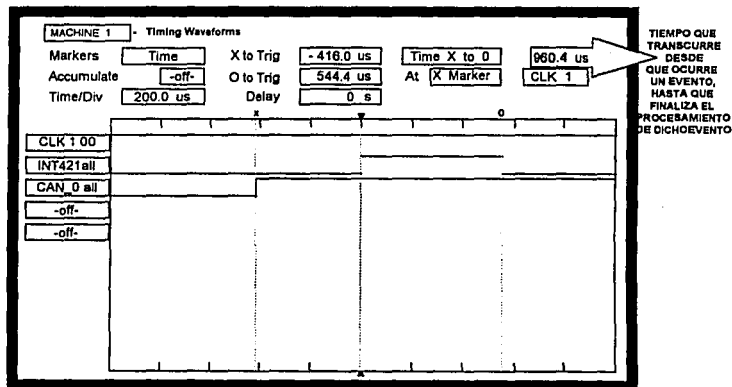


Fig. 3.3. Gráfica del analizador de estados lógicos.

- Los resultados obtenidos en la segunda prueba se muestran en las siguientes tres figuras, en donde se observan los tres escalones y los tres pulsos de interrupción.
- El tiempo que transcurre desde que se presenta cada evento hasta que finaliza su procesamiento en cada uno de los canales fue el siguiente:
 Tiempo del canal 0: 980.4 μ s (ver fig. 3.4).
 Tiempo del canal 1: 550.4 μ s (ver fig. 3.5).
 Tiempo del canal 2: 960.4 μ s (ver fig. 3.6).
- Con los resultados obtenidos se comprueba que el tiempo para determinar la secuencia entre dos eventos distintos es menor a 1 mseg.

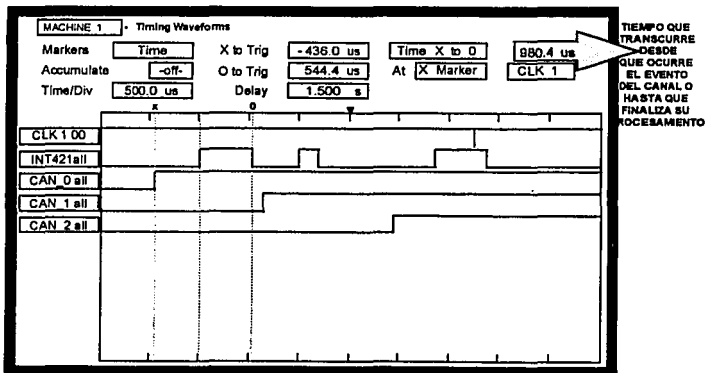


Fig. 3.4. Gráfica del analizador de estados lógicos.

3.2.2.3. Pruebas de transmisión y recepción de información con el subsistema de presentación.

+ Alcance.

Estas pruebas tienen como objetivo verificar el correcto procesamiento y registro de eventos simulados en las entradas de los módulos SAC-421, tomando en consideración los datos que son transmitidos a la PC del subsistema de presentación; también se verificará que los datos que se envían de la PC al subsistema de adquisición son recibidos correctamente y procesados de manera adecuada.

+ Equipo.

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - Una tarjeta de entradas digitales con interrupción SAC-421.
- Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.
- Una PC auxiliar con el programa DEBUGER.
- Una PC auxiliar con el programa XTALK.

+ Procedimiento.

- La figura 3.7 muestra la manera en que se conectó el equipo para la realización de estas pruebas.

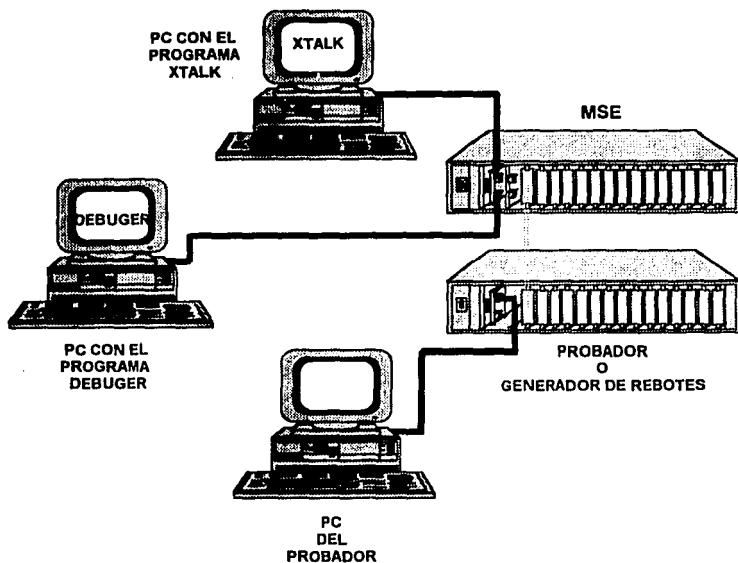


Fig.3.7. Conexión de equipo para prueba de resolución del MSE.

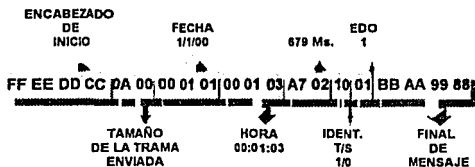
- Activadas las rutinas para el procesamiento de datos, se llevará a cabo la primera prueba, en la que a través del programa DEBUGER y por el puerto B de la tarjeta SX-233, se actualizarán los valores de las variables de tiempo y fecha para que al generar un evento a través del probador, se verifique que la trama que generará el módulo de procesamiento contiene esta fecha actualizada, y se encuentre completa con la información del evento (tarjeta, señal, estado, tiempo y fecha); esto último, se llevará a cabo haciendo uso del programa XTALK, con lo que se verificará la información que será enviada a la

PC del subsistema de presentación por el puerto A de la tarjeta SX-233.

- La segunda prueba consistirá de la verificación de la información que es almacenada en el módulo de procesamiento (en BUFFER) y que al recibir del subsistema de presentación la petición de dicha información, esta será enviada por el puerto A de la tarjeta SX-233. Para esta prueba, bastará con cambiar el valor de la variable PETICION a 1, a través del puerto B de la SX-233 y utilizando el programa DEBUGER. Al efectuar esto, la canasta enviará por el puerto A la información almacenada en BUFFER, la cual será verificada empleando el programa XTALK.
- La información que se recibirá por el puerto A de la tarjeta SX-233, del evento generado deberá ser la misma que se recibirá al cambiar el valor de la variable petición.

Resultados.

- Al simular un evento por el canal 0 de la tarjeta SAC-421, a través del programa XTALK se recibió lo siguiente:



En donde los valores de la fecha y el tiempo, tienen el valor inicial que se asignó en la programación del módulo de inicialización de parámetros.

- Los valores que se pusieron en las variables de fecha fueron:

Valor de la variable AÑO: 5E (94 dec).

Valor de la variable MES: 03 (3 dec).

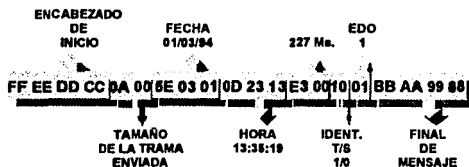
Valor de la variable DIA: 01 (1 dec).

Valor de la variable HORA: 0D (13 dec).

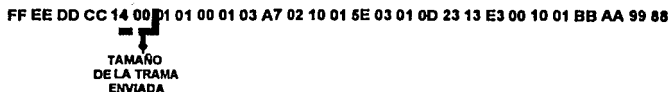
Valor de la variable MINUTO: 23 (35 dec).

Valor de la variable SEGUND: 00 (0 dec).

- Al generar otro evento, la información que se recibió a través del programa XTALK fue la siguiente:



- Como se puede ver en la información recibida, el tiempo y la fecha con la que está etiquetado el evento, es igual a los valores que se pusieron a dichas variables anteriormente, además se puede ver que la información es completa, con lo cual se deduce que el procesamiento de los eventos se lleva a cabo de manera adecuada.
- En la realización de la segunda prueba, primeramente se comprobó a través del programa DEBUGER que el contenido de la variable PETICION era 0 y posteriormente se cambió a 1, al efectuar esto último el módulo de procesamiento envió la trama que se muestra a continuación.



- En la información recibida se observa que los 2 eventos que ya se habían enviado por el canal A de la tarjeta SX-233 se enviaron una vez más pero en una sola trama, la cual tiene como longitud 20 bytes de información. Como, la trama contiene la información de los eventos completa, se concluye que el almacenamiento de los eventos en el subsistema de adquisición (en BUFFER) es el adecuado y que esta información será enviada al subsistema de presentación a petición del usuario.

3.2.2.4. Prueba del almacenamiento de hasta los últimos 256 eventos.

+ Alcance.

Estas pruebas tienen como objetivo verificar que el subsistema de adquisición es capaz de almacenar en memoria hasta los últimos 256 eventos ocurridos, aún sin encontrarse conectado el subsistema de presentación.

+ Equipo.

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - Una tarjeta de entradas digitales con interrupción SAC-421.
- Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.
- Una PC auxiliar con el programa DEBUGER.
- Una PC auxiliar con el programa XTALK.

+ Procedimiento.

- El equipo se conectó de la misma forma que para las pruebas anteriores (fig. 3.7)

- Se realizó el mismo procedimiento que para la verificación del correcto almacenamiento de la información en el subsistema de adquisición. Primeramente se generaron varios eventos con el probador y posteriormente se cambió el valor de la variable PETICION a 1 para que se enviara a través del puerto A de la tarjeta SX-233 el contenido de BUFFER. Como el tamaño de la trama enviada por evento es igual a 10 bytes, el número de bytes correspondientes a 256 eventos no deberá sobrepasar los 2560 (256 eventos X 10 bytes = 2560 bytes).

+ **Resultados.**

- Después de haber generado varios eventos en el módulo de entrada (SAC-421), se cambió el valor de la variable PETICION a 1 y se recibió la información enviada por el canal A de la tarjeta SX-233 haciendo uso del programa XTALK, con lo que se verificó que el tamaño de la trama no sobrepasaba los 2560 bytes. Esto se realizó varias veces incrementando el número de eventos generados y el tamaño máximo de la trama en todos los casos no sobrepasó los 2560 bytes.
- Con lo anterior se deduce que el subsistema de adquisición es capaz de almacenar hasta los últimos 256 eventos ocurridos.

3.2.3. PRUEBAS FUNCIONALES REALIZADAS AL SUBSISTEMA DE PRESENTACIÓN DEL MSE.

3.2.3.1. Pruebas de recepción y transmisión de información con el subsistema de adquisición.

+ **Alcance.**

Estas pruebas tienen como objetivo verificar que la información que es enviada por el subsistema de adquisición es recibida y procesada correctamente por el subsistema de presentación. Posteriormente, se comprobará que el envío de información que realiza el subsistema de presentación al subsistema de adquisición es correcta.

✦ **Equipo.**

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - Una tarjeta de entradas digitales con interrupción SAC-421.
- Una PC con el software del subsistema de presentación del MSE.
- Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.
- Una PC auxiliar con el programa XTALK.

✦ **Procedimiento.**

- El equipo se conectó como se muestra en la figura 3.8
- En este caso, el puerto serie A de la tarjeta SX-233 se conecta tanto a la PC que contiene el programa XTALK (por el COM1) como al puerto serie COM1 de la PC que contiene el software del subsistema de presentación; esta conexión se realiza con la finalidad de obtener la información de los eventos antes de ser procesada por el subsistema de presentación, para poder compararla con la información que será desplegada por éste.
- El puerto B de la tarjeta SX-233 se conecta al puerto serie 2 de la PC del subsistema de presentación para realizar las pruebas de envío de información al subsistema de adquisición.

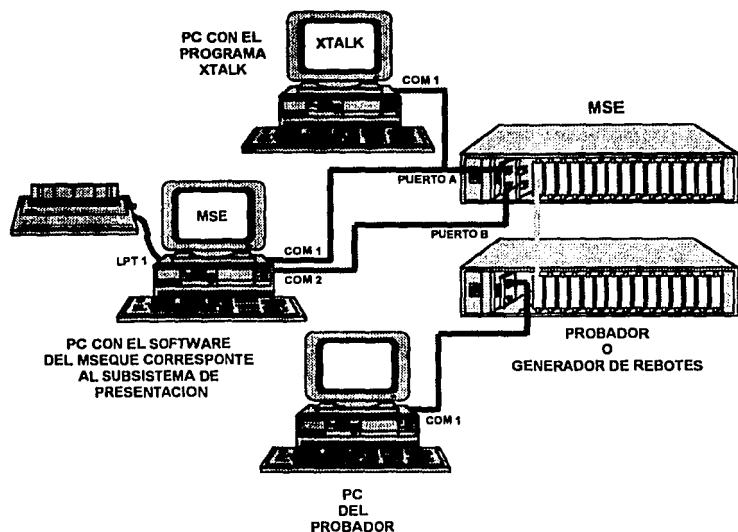


Fig. 3.8. Conexión de equipo para pruebas de transmisión y recepción de información del subsistema de presentación.

- Para comprobar que la información adquirida por el subsistema de presentación es recibida adecuadamente, se generarán una serie de eventos utilizando el generador de rebotes, y la información que será recibida con el programa XTALK se comparará con la información que deberá desplegar el subsistema de presentación. En la función RECIBE que lleva a cabo la recepción de la información del subsistema de adquisición y la coloca en un área de memoria para su posterior procesamiento, se anexará una función que permite desplegar en hexadecimal la información que se va recibiendo por el COM1 de la

PC, con lo que será más fácil realizar esta prueba.

- Para comprobar que el subsistema de presentación envía de manera adecuada la información de la fecha al subsistema de adquisición; primeramente, se reinicializa el módulo de adquisición y se genera una serie de eventos, posteriormente se lleva a cabo el envío de la fecha actualizada y la generación de más eventos. Por último se verificará que la información enviada por el subsistema de adquisición si fue actualizada, comparando la fecha desplegada antes y después del envío de la fecha.

✦ **Resultados.**

- Al efectuar la primera prueba, se comprobó que efectivamente la información recibida por el subsistema de presentación (la presentada por el programa XTALK) es la misma que se recibe por el COM1 de la PC, con lo que se comprueba que el subsistema de presentación recibe de manera adecuada la información enviada por el subsistema de adquisición, pues tanto la trama enviada por el subsistema de adquisición como la presentada en el monitor de la PC del subsistema de presentación es la misma.
- El envío de la fecha por el puerto serie COM2 de la PC es correcta dado que la etiqueta de los eventos generados posteriores al envío de la fecha es realizada con la fecha actualizada.

3.2.3.2. Prueba de despliegue de información.

✦ **Alcance.**

Esta prueba tiene como finalidad, verificar que el despliegue de los eventos se realiza de acuerdo a la información descrita en la especificación de requerimientos.

✦ **Equipo.**

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.

- Una tarjeta de expansión de memoria SAC-929.
- Una tarjeta de comunicaciones SX-233.
- Una tarjeta de entradas digitales con interrupción SAC-421.
- ◆ Una PC con el software del subsistema de presentación del MSE.
- ◆ Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.

✦ Procedimiento.

- ◆ En esta prueba el equipo se conectó de la misma forma que para las pruebas 3.2.3.1, a excepción de la PC que contiene el programa XTALK que en esta prueba no se empleará.
- ◆ Esta prueba consiste de la generación de eventos y verificación de la información que es registrada y desplegada por el subsistema de presentación.

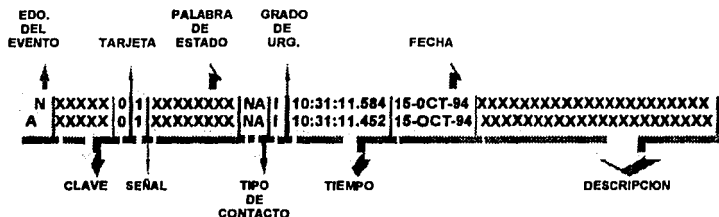
Se verificará que la información desplegada de cada evento consista de los siguientes datos: clave del evento ocurrido, tarjeta y señal en donde se detectó dicho evento, la palabra del estado actual del evento, tipo de contacto, la urgencia de atención, el tiempo y la fecha en que ocurrió el evento, y la leyenda que describe dicho punto.

✦ Resultados.

- ◆ Al generar un rebote el canal 0 de la tarjeta 0, la información que apareció en la pantalla de la PC del subsistema de presentación, fue la siguiente:

```
N XXXXX 0 1 XXXXXXXX NA | 10:31:11.584 15-OCT-94 XXXXXXXXXXXXXXXXXXXX
A XXXXX 0 1 XXXXXXXX NA | 10:31:11.452 15-OCT-94 XXXXXXXXXXXXXXXXXXXX
```

- ◆ A continuación se identificó cada campo de información desplegado, como se muestra a continuación:



- Es importante señalar que los campos de información que se presentaron en pantalla conteniendo equis, se debió a que la base de datos que contiene la información de los canales de campo no se encuentra actualizada, y se despliegan valores por defecto.
- De lo anterior se concluye que la información que es recibida por el subsistema de información es procesada y desplegada como se especificó en los requerimientos.

3.2.3.3. Prueba del almacenamiento de los últimos 1024 eventos en disco y verificación de consistencia de información.

✦ Alcance.

Con esta prueba se verificará que en disco duro se lleva a cabo el almacenamiento de los últimos 1024 eventos ocurridos y que dicha información consiste de los mismos campos de información que se obtuvieron en la prueba 3.2.3.2.

✦ Equipo.

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - Una tarjeta de entradas digitales con interrupción SAC-

421.

- ♦ Una PC con el software del subsistema de presentación del MSE.
 - ♦ Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.
- ✦ Procedimiento.
- ♦ En esta prueba el equipo se conectó de la misma forma que para la prueba 3.2.3.2.
 - ♦ Aquí se comprobará que la información de los eventos ocurridos es almacenada en forma continua en el archivo de disco ALM_DISK.DAT y máximo número de registros almacenados en este disco es de 1024 eventos. Para lo que, al ejecutar el software del MSE del subsistema de adquisición se realizará una generación continua de eventos los cuales deberán ser almacenados en dicho archivo.
 - ♦ El tamaño y el contenido de ese archivo se podrá ver para esta prueba auxiliándonos de un editor de textos (por ejemplo, el EDIT del DOS).
- ✦ Resultados.
- ♦ Al generar una serie de eventos en los canales de la tarjeta SAC-421, estos fueron adquiridos y procesados para su posterior despliegue y almacenamiento en el subsistema de presentación. La información almacenada en el archivo de disco ALM_DISK.DAT se revisó de vez en cuando para ir analizando que su tamaño nunca fue mayor a 1024 eventos. Al llegar por primera vez a este tamaño el almacenamiento continuó utilizándose este archivo como una cola circular.
 - ♦ De la prueba realizada se concluye que en el subsistema de adquisición se realiza el despliegue y almacenamiento simultáneo de los eventos ocurridos; y que el almacenamiento se realiza en una cola circular con tamaño no mayor a 1024 eventos (es decir, se almacenan únicamente los últimos 1024 eventos ocurridos).

3.2.3.4. Prueba del despliegue e impresión simultáneo de los eventos ocurridos.

+ Alcance.

El objetivo de esta prueba es comprobar que el MSE lleva a cabo el despliegue e impresión de los eventos ocurridos de manera simultánea.

+ Equipo.

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - Una tarjeta de entradas digitales con interrupción SAC-421.
- Una PC con el software del subsistema de presentación del MSE.
- Una impresora conectada al puerto paralelo LPT1 de la PC del subsistema de presentación.
- Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.

+ Procedimiento.

- En esta prueba el equipo se conectó como se muestra en la figura 3.8., omitiendo únicamente la PC con el programa XTALK.
- Para la realización de esta prueba se encenderá la impresora y se generarán varios eventos; los cuales, deberán ser desplegados e impresos de manera simultánea por el subsistema de presentación.

+ Resultados.

- Al generar una serie de eventos en los canales de la tarjeta SAC-421, estos fueron adquiridos, procesados y posteriormente desplegados en la pantalla e impresora del subsistema de presentación de manera simultánea, como se muestra en la figura 3.9 y en la 3.10.

N	L5303	0	11	ARRANCAR	NA	I	10:31:24.138	15-OCT-94	ARRANCAR_BCBP
A	L5303	0	11	PARAR	NA	I	10:31:24.120	15-OCT-94	ARRANCAR_BCBP
N	L5303	0	11	ARRANCAR	NA	I	10:31:24.102	15-OCT-94	ARRANCAR_BCBP
A	L5303	0	11	PARAR	NA	I	10:31:24.100	15-OCT-94	ARRANCAR_BCBP
N	L5211	0	3	NO	NA	I	10:31:17.800	15-OCT-94	BAJO_NIVEL_DOMO
A	L5211	0	3	SI	NA	I	10:31:17.607	15-OCT-94	BAJO_NIVEL_DOMO
N	L3143	0	0	NO	NA	I	10:31:15.290	15-OCT-94	F/S_BBAS_CIR_O_ALT_P_C
A	L3143	0	0	SI	NA	I	10:31:14.909	15-OCT-94	F/S_BBAS_CIR_O_ALT_P_C
N	L5210	0	1	NO	NA	I	10:31:11.584	15-OCT-94	ALTO_NIVEL_DOMO
A	L5210	0	1	SI	NA	I	10:31:11.452	15-OCT-94	ALTO_NIVEL_DOMO
N	L5210	0	1	NO	NA	I	10:31:11.450	15-OCT-94	ALTO_NIVEL_DOMO
A	L5210	0	1	SI	NA	I	10:31:11.431	15-OCT-94	ALTO_NIVEL_DOMO
N	L5210	0	1	NO	NA	I	10:31:11.427	15-OCT-94	ALTO_NIVEL_DOMO
A	L5210	0	1	SI	NA	I	10:31:11.426	15-OCT-94	ALTO_NIVEL_DOMO

I	NICIAIZA	D	ATOS SAC	U	TILERIAS	CONF.	SAC	H	ISTORICO
---	----------	---	----------	---	----------	-------	-----	---	----------

IMPRESORA EN LINEA SELECCIONE UNA OPCION [F1]-CLS [ESC]-SALIR

Fig. 3.9. Despliegue en pantalla de los eventos ocurridos.

- Tanto la información desplegada en la pantalla (ver fig. 3.10.), como la impresa (ver fig. 3.11) fue la misma.

MONITOR DE SECUENCIA DE EVENTOS

IDENT	T	S	ESTADO	C	U	TIEMPO	FECHA	EVENTO	
A	L5210	0	1	SI	NA	I	10:31:11.426	15-OCT-94	ALTO_NIVEL_DOMO
N	L5210	0	1	NO	NA	I	10:31:11.427	15-OCT-94	ALTO_NIVEL_DOMO
A	L5210	0	1	SI	NA	I	10:31:11.431	15-OCT-94	ALTO_NIVEL_DOMO
N	L5210	0	1	NO	NA	I	10:31:11.450	15-OCT-94	ALTO_NIVEL_DOMO
A	L5210	0	1	SI	NA	I	10:31:11.452	15-OCT-94	ALTO_NIVEL_DOMO
N	L5210	0	1	NO	NA	I	10:31:11.584	15-OCT-94	ALTO_NIVEL_DOMO
A	L3143	0	0	SI	NA	I	10:31:14.909	15-OCT-94	F/S_BBAS_CIR_O_ALT_P_C
N	L3143	0	0	NO	NA	I	10:31:15.290	15-OCT-94	F/S_BBAS_CIR_O_ALT_P_C
A	L5211	0	3	SI	NA	I	10:31:17.607	15-OCT-94	BAJO_NIVEL_DOMO
N	L5211	0	3	NO	NA	I	10:31:17.800	15-OCT-94	BAJO_NIVEL_DOMO
A	L5303	0	11	PARAR	NA	I	10:31:24.100	15-OCT-94	ARRANCAR_BCBP
N	L5303	0	11	ARRANCAR	NA	I	10:31:24.102	15-OCT-94	ARRANCAR_BCBP
A	L5303	0	11	PARAR	NA	I	10:31:24.120	15-OCT-94	ARRANCAR_BCBP
N	L5303	0	11	ARRANCAR	NA	I	10:31:24.138	15-OCT-94	ARRANCAR_BCBP

Fig. 3.10. Impresión de los eventos ocurridos.

- Con lo anterior se comprueba que el sistema MSE realiza el despliegue de la información de los eventos ocurridos simultáneamente en pantalla y en impresora como se describió en la especificación de los requerimientos.

3.2.3.5. Prueba del despliegue de los menús de opciones.

+ Alcance.

Esta prueba verificará la correcta realización de las funciones que se llevan a cabo a petición del usuario.

+ Equipo.

- Una canasta SAC conteniendo las siguientes tarjetas.
 - Una tarjeta procesadora SAC-1887 con la programación correspondiente al MSE.
 - Una tarjeta de expansión de memoria SAC-929.
 - Una tarjeta de comunicaciones SX-233.
 - Una tarjeta de entradas digitales con interrupción SAC-421.
- Una PC con el software del subsistema de presentación del MSE.
- Una impresora conectada al puerto paralelo LPT1 de la PC del subsistema de presentación.
- Una canasta SAC con las tarjetas que conforman el probador, o bien, generador de rebotes, así como la PC con el software correspondiente para la ejecución de dicho probador.

+ Procedimiento.

- Para esta prueba, el equipo se conectó igual que para la realización de la prueba 3.2.3.4.
- A partir de la presentación del menú principal en el monitor de la PC del MSE, se llevará a cabo la ejecución de las diversas funciones individualmente con datos de prueba propios, así como la selección de los menús subsiguiente.

+ Resultados.

- La ejecución de todas las funciones se realizaron de acuerdo a la especificación de requerimientos. Y las pantallas que aparecen son de acuerdo a las que se presentaron en el diseño del software.
- De lo anterior se deduce que las funciones que permitirán al usuario interactuar de forma personalizada con el MSE operan de acuerdo a lo establecido en el diseño del capítulo anterior y cumple con los requerimientos especificados en el capítulo 1.

3.3. CERTIFICACIÓN DEL SISTEMA.

Esta etapa representa la fase final en el desarrollo del sistema y la cual consiste en la verificación de las características funcionales comparadas con la especificación del diseño y los resultados de pruebas realizadas.

La certificación de cada una de las funciones y características del sistema fueron realizadas por el laboratorio de Pruebas Eléctricas y Mecánicas de CFE (LAPEM/CFE), en la termoeléctrica de Gómez Palacio Durango y en la termoeléctrica de Dos Bocas Veracruz, en donde actualmente se encuentra funcionando el sistema MSE.

CONCLUSIONES

Realizando un balance entre los requerimientos especificados en el capítulo 1 y lo logrado al finalizar este trabajo, se puede decir que se cumplió con el objetivo primordial establecido al inicio de dicho trabajo; pues, el Sistema de Monitoreo y Registro de Secuencia de Eventos (MSE) desarrollado proporciona al operador en planta de cualquier proceso industrial toda la información de los eventos ocurridos antes, durante y después de un disturbio del proceso, satisfaciendo así su necesidad de conocer y atender de manera rápida y eficaz dichos disturbios.

Las principales características del MSE desarrollado y con el cual se logra lo anterior son las siguientes:

- Tiempo de resolución de 1 milisegundo.
- Capacidad de hasta 240 puntos de entrada.
- Las señales a monitorear serán contactos de dos estados o voltajes (es decir, señales digitales).
- La técnicas de procesamiento utilizadas para la identificación de los cambios de estado de los contactos de campo son: Interrupción y Polling.
- El aislamiento con los contactos de campo se realiza a través de optoacopladores.
- El filtrado de los eventos se efectúa a través de un filtro por software.
- Los dispositivos de despliegue son, CRT a color e impresora.
- La información que contendrán los despliegues es: estado del evento, la clave del contacto de campo, el número de la tarjeta y señal, el tipo de contacto de campo (NA, NC), la urgencia de atención (N, U o I), el tiempo y la fecha de aparición del evento y la descripción de las señales de campo.
- Tiene una capacidad de memorización en RAM de hasta los últimos 256 eventos ocurridos.
- Efectúa el autodiagnóstico del hardware del sistema al inicializarse el sistema, así como la ejecución a petición del usuario del diagnóstico de la configuración del sistema.
- Es totalmente configurable en campo.
- Efectúa diversas funciones que permiten al usuario interactuar en forma personalizada con el sistema.
- Genera diversos reportes de entre los últimos 1024 eventos ocurridos y almacenados simultáneamente en un archivo de disco considerando parámetros dados por el operador en planta.

En el diseño del MSE se tomó en consideración la posibilidad de expansión del sistema con su interacción con una red de comunicaciones, que permita proporcionar información a sistemas de nivel superior (DCS, SCADA's, etc.) y cuyo desarrollo no se ha realizado, por encontrarse fuera del alcance del trabajo de tesis que se está concluyendo.

Funcionalmente, el sistema se probó en la planta de ciclo combinado de Gómez Palacio Durango, y en la termoeléctrica de Dos Bocas Veracruz; gracias a esto, se realiza en su momento la corrección y mejoramiento del sistema. Y es en estas dos plantas de CFE, donde actualmente se encuentra instalado y en funcionamiento el MSE.

El desarrollo e implantación del MSE se efectuó en un periodo de tiempo aproximado de un año y medio, en las instalaciones y con el apoyo del Instituto de Investigaciones Eléctricas (IIE).

APENDICE A

DICCIONARIO DE DATOS Y CATALOGO DE FUNCIONES

Este apéndice contiene las variables y estructuras de datos utilizadas en las funciones que integran los módulos de programación del software del MSE. Así como, el catálogo de las funciones que integran dicho software.

DICCIONARIO DE DATOS

A continuación se describen las variables empleadas en los módulos de programación que se desarrollarán para el subsistema de adquisición del MSE (módulo de adquisición y procesamiento).

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
config_con[]	char	Arreglo que indicará la presencia de conectores que podrá tener la canasta SAC	DIAGNOSTICO0 INICIALIZA0
config_iden[]	char	Arreglo que contendrá los identificadores de las diversas tarjetas que aceptará el MSE	DIAGNOSTICO0 INICIALIZA0
tab_conec	word	Tabla de conectores	DIAGNOSTICO0 INICIALIZA0
config_diag[]	char	Arreglo que contendrá el resultado del diagnóstico de la configuración del MSE	DIAGNOSTICO0 INICIALIZA0
tab_diag	word	Tabla de diagnóstico	DIAGNOSTICO0
diag_mvax	word	Los bytes que contendrán el diagnóstico de las tarjetas	DIAGNOSTICO0
conec_mvax	word	Los bytes que contendrán el diagnóstico de los conectores	DIAGNOSTICO0
ap_iden	word	Dirección a la tabla de identificadores	Diagnostico0
ap_conec	word	Dirección a la tabla de conectores	DIAGNOSTICO0
ap_diag	word	Contendrá la dirección de la tabla de diagnóstico	DIAGNOSTICO0
conector	byte	Indicará la presencia o ausencia de conector	DIAGNOSTICO0

Estructura/Variable	Tipo	Descripción	Utilizada por:
datini	word	Será la tabla de inicialización de la canasta	INICIALIZA0
DIR_RAM0	word	Dirección de la RAM0	DIAGNOSTICO0 INICIALIZA0
DIR_RAM1	word	Dirección de la RAM1	DIAGNOSTICO0 INICIALIZA0
DIR_RAM2	word	Dirección de la RAM2	DIAGNOSTICO0 INICIALIZA0
DIR_RAM3	word	Dirección de la RAM3	DIAGNOSTICO0 INICIALIZA0
DIR_ROM0	word	Dirección del banco de memoria 0	DIAGNOSTICO0 INICIALIZA0
DIR_ROM1	word	Dirección del banco de memoria 1	DIAGNOSTICO0 INICIALIZA0
DIR_ROM2	word	Dirección del banco de memoria 2	DIAGNOSTICO0 INICIALIZA0
DIR_ROM3	word	Dirección del banco de memoria 3	DIAGNOSTICO0 INICIALIZA0
res_diag	byte	Resultado del diagnóstico del watchdog	DIAGNOSTICO0 INICIALIZA0
diag_mem	byte	Diagnóstico final de las RAM	DIAGNOSTICO0 INICIALIZA0
RAM_OFF	byte	RAM en mal estado (Número)	DIAGNOSTICO0 INICIALIZA0
diag_rom	byte	Diagnóstico final de las ROM	DIAGNOSTICO0 INICIALIZA0
chek_sum1	word	Checksum calculado	INICIALIZA0
romala	byte	ROM en mas estado (Número)	DIAGNOSTICO0 INICIALIZA0

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
tiempo	word	Reloj grande en watchdog	INICIALIZA0
timer	word	Reloj pequeño en watchdog	INICIALIZA0
contad01	word	Contador en watchdog	INICIALIZA0
contad02	word	Contador en watchdog	INICIALIZA0
front	word	Índice apuntador de escritura a la cola circular de almacenamiento POINTER	Tarea: MANEJA_IN0
rear	word	Índice apuntador de lectura a la cola circular de almacenamiento POINTER	Tarea: PC_EVENT0
front1	word	Índice apuntador de escritura a la cola circular de almacenamiento EVENTOS	Tarea: SERVINT0
rear1	word	Índice apuntador de escritura a la cola circular de almacenamiento EVENTOS	Tarea: MENEJA_IN0
rear2	word	Índice apuntador de lectura a la cola circular de almacenamiento BUFFER	Tarea: PC_EVENT0

Estructura/Variable	Tipo	Descripción	Utilizada por:
front2	word	Índice apuntador de escritura a la cola circular de almacenamiento BUFFER	Tarea: PC_EVEN0
HORA	byte	Variable que contiene la hora tiempo real del MSE	Tarea: HORA0 INTER0 READCARD_4210 INICIALIZA0
MINUTO	byte	Variable que contiene los minutos tiempo real del MSE	Tarea: HORA0 INTER0 READCARD_4210 INICIALIZA0
SEGUND	byte	Variable que contiene los segundos tiempo real del MSE	Tarea: HORA INTER0 READCARD_4210 INICIALIZA0
DIA	byte	Variable que contiene el día tiempo real del MSE	Tarea: HORA INTER0 READCARD_4210 INICIALIZA0
MES	byte	Variable que contiene el mes tiempo real del MSE	Tarea: HORA INTER0 READCARD_4210 INICIALIZA0
YEAR	byte	Variable que contiene el año tiempo real del MSE	Tarea: HORA INTER0 READCARD_4210 INICIALIZA0
ref_tarj[]	unsigned char	Arreglo que contendrá el número de los puntos como referencia para cada tarjeta SAC-421	Tarea: MANEJA_IN0
oeye	int	Bandera que indicará la ocurrencia de un evento	Tarea: PC_EVEN0

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
tiempo_mseg	unsigned int	Contiene el tiempo de ocurrencia de un evento, es decir, los milisegundos	LEE_82540
datos	struct	Estructura de datos que contendrá el tiempo de ocurrencia de un evento, el estado de la señal y su identificador (nibble bajo canal y alto slot), se anexan las variables de tiempo y fecha del evento	Tarea: MANEJA_IN0
POINTER[]	struct datos	Es una cola circular de almacenamiento con la estructura de datos	Tarea: MANEJA_IN0
BUFFER[]	struct datos	Es una cola circular de almacenamiento con la estructura de datos	Tarea: PC_EVENT0
*fuera	struct datos	Apuntador de memoria a POINTER	Tarea: PC_EVENT0 AMACENA_BUFFER0
*fuera2	struct datos	Apuntador de memoria a BUFFER	Tarea: PC_EVENT0
monitor	struct	Estructura de datos que contendrá la información básica obtenida de una tarjeta SAC-421 al ocurrir un evento. Se incluyen las variables de tiempo y fecha.	Tarea: SERVINT0

Estructura/Variable	Tipo	Descripción	Utilizada por:
EVENTOS	struct monitor	Es una cola circular de almacenamiento con la estructura de datos	Tarea: SERVINT0 Tarea: MANEJA_INT0
*fuera1	struct monitor	Apuntador de memoria a EVENTOS	Tarea: MANEJA_INT0
petición	char	Tomará el valor de 1 cuando el usuario solicite el contenido de la cola de almacenamiento circular de buffer	PC_EVENT
hmvax	struct	Estructura de datos que contendrá la información completa de los eventos que será enviados a la PC (incluye el tiempo y fecha de la canasta, tiempo de ocurrencia, identificador y estado de los eventos)	Tarea: PC_EVENT0
PC[]	struct hmvax	Arreglo con la estructura de hmvax, que contendrá la información que se enviará a la PC	Tarea: PC_EVENT0
tab_iden[]	word	Arreglo que contendrá los identificadores de las tarjetas SAC	Tarea: SERVIN0 Tarea: INIDAT0 DIAGNOSTICO0

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
dir_hw[]	int	Arreglo que contendrá las direcciones de hardware de las tarjetas de la canasta SAC	Tarea: SERVIN() Tarea: INIDAT() DIAGNOSTICO()
dat_ini[]	char	Arreglo que contiene el valor de los contactos de campo al inicializar el MSE	Tarea: SERVIN() Tarea: INIDAT()
estado_actual[]	unsigned char	Arreglo que contendrá el estado actual de los canales	Tarea: PC_EVENTO()
estado_anterior[]	unsigned char	Arreglo que contendrá el estado anterior de los canales	Tarea: PC_EVENTO()
PET_USR	struct hmvax	Arreglo con la estructura de hmvax, que contendrá la información que se enviará a la PC a petición del usuario	Tarea: PC_EVENTO()

Las variables utilizadas en los módulos de programación que se desarrollarán para el subsistema de presentación (interfaz, inicialización, solicitud de información, configuración, manejador de archivos, manejador de base de datos, presentación de información y generador de reportes) son las siguientes;

Estructura/Variable	Tipo	Descripción	Utilizada por:
ap_esc	int	apuntador de escritura a la lista circular buffer	PRINCIPAL()
ap_lect	int	apuntador de lectura a la lista circular buffer	PRINCIPAL()
ap_esc1	int	apuntador de escritura a la lista circular buffer1	PET_BUFFER()
ap_lect1	int	apuntador de lectura a la lista circular buffer1	PET_BUFFER()
estado	int	Indicador del byte de información que se estará recibiendo para almacenarlo en variables de memoria	RECIBE()
contador	unsigned int	Número de bytes que conforman la trama que se recibirá	RECIBE()
db	int	Toma el valor de 1 para indicar que la trama de información se ha recibido totalmente	RECIBE() PET_BUFFER()
banaux	int	Bandera que indica si el archivo ALM_DISK alcanzó o no su máximo tamaño	PRINCIPAL() ALMACENA()
res	int	Toma el valor de 1 para indicar que existe el archivo de almacenamiento ALM_DISK	PRINCIPAL() ALMACENA()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
res1	int	Toma el valor de 1 para indica que el apuntador al archivo ALM_DISK esta al inicio de este, al momento de reescribir dicho archivo	PRINCIPAL() ALMACENA()
buff[80]	char	Área de memoria utilizada como almacenamiento temporal de la información a desplegar del evento ocurrido	IMPRIME() FORMATEA()
buff_alm[80]	char	Área de memoria utilizada como almacenamiento temporal de la información a almacenar del evento ocurrido	FORMATEA()
colhw[256*2]	int	Área de memoria que contendrá el color de presentación de las señales de campo para eventos en alarma y regreso a normal	LEE_SEÑALES() DESPLIEGA()
tagest[256*2][24]	char	Área de memoria que contendrá la palabra del estado en alarma y regreso a normal de las señales de campo	LEE_SEÑALES() FORMATEA() PET_BUFFER()

Estructura/Variable	Tipo	Descripción	Utilizada por:
buff_dal[80]	char	Área de memoria utilizada como almacenamiento temporal de la información de los eventos que envía el subsistema de adquisición, al hacer la petición de los últimos 256 eventos almacenados en BUFFER	FORMATEA()
tagcol	int	Variable contendrá el color de despliegue de las señales, el cual relaciona los dos colores de despliegue de cada señal	LEE_SEÑALES()
taglog[256][24]	char	Área de memoria que contendrá la leyenda de las señales de campo	LEE_SEÑALES() FORMATEA() PET_BUFFER()
tag_clave[256][7]	char	Área de memoria que contendrá la clave de las señales de campo	LEE_SEÑALES() FORMATEA() PET_BUFFER()
urgen[256*2][2]	char	Área de memoria que contendrá la urgencia de atención de las señales de campo para estado en alarma y regreso a normal (N, U,I)	LEE_SEÑALES() FORMATEA() PET_BUFFER()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
edoini[256][2]	char	Área de memoria que contendrá el estado inicial de las señales de campo (N, A, I o D)	LEE_SEÑALES() FORMATEA() DESPLGA() IMPRIME() PET_BUFFER()
tip_con[256][3];	char	Área de memoria que contendrá el tipo de contacto de las señales de campo (NA, NC)	LEE_SEÑALES() FORMATEA() PET_BUFFER()
name_arch[]	unsigned char	Contendrá el nombre del archivo a manipular	LEER(), IMPRIMIR() ALMACENAR()
tx_datos[]	unsigned char	Área de memoria que contendrá la información del tiempo y la fecha que será enviada al subsistema de adquisición	INI_FECHA()
form1, form2, form3, form4, form5, form6, form7, form8, form9, form10	int	Contendrán la habilitación o deshabilitación de los campos de información a presentar	PRINCIPAL() CONFIG()
encabezado[]	char	Contendrá el nombre de los campos de información desplegados al inicializar el sistema	PRINCIPAL()
b_p[]	char	Contendrá el nombre de los campos de información desplegados según el usuario	CONFIG()

Estructura/Variable	Tipo	Descripción	Utilizada por:
eventos	struct	Estructura de datos que contendrá la información de las señales de campo: Numero de la tarjeta Numero de la señal Clave de la señal Descripción de la señal Tipo de contacto NA o NC Grado de urgencia en normal Grado de urgencia en alarma Palabra de estado normal Palabra de estado alarma Opción de colores Estado inicial del evento char numtar[3] char numsen[3] char clave[7] char descrip[33] char tipo_con[3] char urg_nor[2] char urg_alar[2] char edo_norm[9] char edo_alar[9] char color[2] char edo_ini[2]	BASE_DATOS()
rec_even2	struct eventos	Estructura de datos de tipo eventos para recuperar la información contenida en la base de datos	ALTAS() CAMBIOS() BAJAS()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Estructura/Variable	Tipo	Descripción	Utilizada por:
info_even	struct eventos	Estructura de datos de tipo eventos para solicitar información al usuario	ALTAS() CAMBIOS()
archi	struct	Estructura de datos que contendrá la información del archivo ALM_DISK: Numero de la tarjeta Numero de la señal Clave de la señal Descripción de la señal Tipo de contacto NA o NC Grado de urgencia Palabra de estado Estado inicial del evento Tiempo del evento Fecha del evento char tar[3] char sen[3] char clave[7] char descrip[33] char tipo_con[3] char urg[2] char estado[9] char edo_ini[2] char hora[14] char fecha[11]	REPORTES()
men	struct archi	Estructura de datos de tipo archi, para recuperar los datos del archivo ALM_DISK	REPORTES()

Estructura/Variable	Tipo	Descripción	Utilizada por:
buffer[] []	unsigned char	Arreglo de memoria para almacenar temporalmente los datos recibidos del subsistema de adquisición	RECIBE()
buffer1[][]	unsigned char	Arreglo de memoria para almacenar temporalmente los datos recibidos del subsistema de adquisición	RECIBE()

CATALOGO DE FUNCIONES

A continuación, se describen las funciones que se desarrollarán en las tareas y funciones de los módulos de programación que constituyen al subsistema de adquisición (SAC).

Función	Tipo	Descripción	Utilizada en:
sac_mse()	void	Función principal de la inicialización de parámetros del subsistema de adquisición	INICIALIZA()
init_pic()	void	Inicializa el Controlador de Interrupciones Programable de la tarjeta procesadora SAC-1887	INICIALIZA()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
carga()	void	Inicia los parámetros necesarios para el funcionamiento del subsistema de adquisición	INICIALIZA()
reset_so() set_so()	void	Rutinas de manejo del led B de la tarjeta de comunicaciones SAC SX-233	INICIALIZA()
inicia()	void	Rutina de manejo de ambos leds de la tarjeta de comunicaciones SAC-SX-233	INICIALIZA()
ram_diag()	Int	Realiza el diagnóstico de los bancos de memoria RAM de la tarjeta procesadora (2 bancos de 64 Kb)	INICIALIZA()
leeide()	Int	Esta función lee del registro de identificación de las tarjetas su identificador y verifica la presencia de conector, dejando la configuración de tarjetas existentes en la tabla dir_hw	INICIALIZA()
sac_821diag();	Int	Realiza el diagnóstico de la tarjeta de comunicaciones SAC-821, en el caso de que se encuentre presenta en el módulo de la SAC	INICIALIZA()

Función	Tipo	Descripción	Utilizada en:
ldetar()	int	Esta función lee el identificador de la tarjeta de una tabla, lo compara con los identificadores validos y dependiendo del resultado ejecuta la rutina de diagnóstico para cada tarjeta	INICIALIZA()
config();	int	Coloca en diag_mvax dos byte indicando la presencia o ausencia de las 18 tarjetas que conforman al subsistema de presentación (entre SAC-421 y tarjeta de comunicación) y en conec_mvax coloca 2 bytes indicando la presencia o ausencia de conector	INICIALIZA()
compara();	int	Compara los identificadores de la configuración del módulo SAC con la tabla de identificadores correcto	INICIALIZA()
mtos();	void	Función que indica al usuario que la inicialización a finalizado	INICIALIZA()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
principia()	void	Esta función Inicializa el diagnóstico de las tarjetas SAC-421 y activa la adquisición continua de señales digitales	DIAGNOSTICO()
rset();	int	Activa la inicialización de la tarjeta del MSE SAC-421.	DIAGNOSTICO()
diag_mse();	int	Esta rutina activa el diagnóstico de las tarjetas de entradas digitales del MSE.	DIAGNOSTICO()
prog_cont2();	void	Programa el contador 2 del temporizador 8254 de la tarjeta SAC-1887	Tarea: INIDAT()
filtro();	int	Programa el tiempo de retardo para el proceso de muestreo, de las señales que presenten rebote (debouncing).	DIAGNOSTICO()
mascara();	int	Selecciona los canales de campo que se requieren muestrear, mediante máscaras de muestreo (dos dígitos hexadecimales por tarjeta significando un uno canal habilitado y cero deshabilitado	DIAGNOSTICO()

Función	Tipo	Descripción	Utilizada en:
actadq();	int()	Activa la adquisición continua de señales digitales, habilitadas por la mascara de muestreo.	DIAGNOSTICO()
Ini_global();	void	Ejecuta la inicialización de la inicialización de las variables globales	Tarea: INIDAT()
Inicio()	void	Inicializa el estado anterior de los eventos con el contenido de dat_ini	Tarea: INIDAT()
sal_conec();	void	Coloca en un buffer el resultado de la presencia de conector	Tarea: INIDAT()
sal_diag();	void	Coloca en un buffer el resultado del diagnóstico de tarjetas	Tarea: INIDAT()
hora()	int	Actualiza las variables de tiempo para etiquetar algún evento	Tarea: HORA()
lee_8254()	void	Lee el tiempo del evento en mseg.	LEE_8254()
transmite();	void	Formatea los datos a ser enviados a la PC, agregando el número de bytes y dejándolo en una tabla para ser manejados por la tarea que llama al manejador de comunicaciones	Tarea: INIDAT()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
for_diag();	void	Coloca en un buffer el resultado del diagnóstico de RAM, ROM, vigilante y de configuración	Tarea: INIDAT()
Read_Card_421()	void	Lee directamente las tarjetas SAC 421 e identifica cual generó una interrupción a la SAC-1887.	Tarea: SERVINT()
*recuo_evento()	struct monitor	Obtiene la dirección en memoria de los eventos almacenados en EVENTOS	Tarea: MANEJA_IN()
ranu()	int	Localiza el número de la tarjeta en donde se detectó el evento	Tarea: MANEJA_IN()
numero_canal()	int	Localiza el número del canal en donde se detectó el evento	Tarea: MANEJA_IN()
*recupera()	struct datos	Localiza la dirección en memoria de POINTER	Tarea: PCEVEN()
compara()	int	Identifica si hay ocurrencia de algún evento	Tarea: PCEVEN()
*rec_buffer()	struct datos	Recupera los eventos almacenados en BUFFER a petición del usuario	Tarea: PCEVEN()

Función	Tipo	Descripción	Utilizada en:
almacena_buffer()	int	Almacena cada segundo los eventos ocurridos en ese lapso de tiempo	Tarea: PCEVEN()
lee_buffer()	int	Lee todos los eventos presentes en BUFFER	Tarea: PCEVEN()
envia()	int	Enviará los eventos de BUFFER al subsistema de presentación	Tarea: PCEVEN()
envia1()	int	Enviará los eventos de BUFFER al subsistema de presentación, cuando el usuario haga la petición	Tarea: PCEVEN()

A continuación, se describen las funciones que se desarrollarán para los módulos de programación que constituyen al subsistema de presentación (PC).

Función	Tipo	Descripción	Utilizada en:
Presentación()	void	Realiza la presentación de la pantalla principal del MSE	PRINCIPAL()
Inicialización()	void	Lleva a cabo la inicialización del hardware de la PC	PRINCIPAL()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
lee_señales()	void	Lee la Información de las señales de campo de la base de datos y la coloca en área de memoria, para su utilización en otras funciones	PRINCIPAL()
formatea()	char	Da a la información almacenada en BUFFER y BUFFER1 un formato de cadena considerando las características de las señales y la configuración de presentación	PRINCIPAL()
desplega()	void	Despliega en pantalla los eventos recibidos considerando el formato dado por formatea() y el color de despliegue dependiendo del estado de la señal	PRINCIPAL()
imprime()	void	Imprime los eventos recibidos	PRINCIPAL()
almacena()	void	Almacena los eventos recibidos en el archivo de disco ALM_DISK.	PRINCIPAL()
menú_principal()	void	Ejecuta diversas funciones a petición del usuario	PRINCIPAL()

Función	Tipo	Descripción	Utilizada en:
set_carac_line()	void	Define las características del puerto serie COM1	INICIALIZACION()
set_vecis()	int	Lee los vectores de interrupción y activa la función RECIBE()	INICIALIZACION()
set_baud_rate()	void	Define las características de velocidad de transmisión y recepción de puerto serie COM1	INICIALIZACION
recibe()	void interrup	Recibe la información transmitida por el subsistema de adquisición y la almacena en BUFFER o BUFFER1 dependiendo el encabezado y final de la trama recibida	INICIALIZACION()
desable_IRQ4()	int	Deshabilita la interrupción de requerimientos de entrada al CPU	INICIALIZACION()
set_carac_line()	void	Define las características del puerto serie COM1	INICIALIZACION()
enable_IRQ4()	int	Habilita la interrupción de requerimiento de entrada al CPU	INICIALIZACION()
set_vecis()	int	Lee los vectores de interrupción y activa la función RECIBE()	INICIALIZACION()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
set_baud_rate()	void	Define las características de velocidad de transmisión y recepción de puerto serie COM1	INICIALIZACION
recibe()	void interrup	Recibe la Información transmitida por el subsistema de adquisición y la almacena en BUFFER o BUFFER1 dependiendo el encabezado y final de la trama recibida	INICIALIZACION()
desable_IRQ4()	int	Deshabilita la interrupción de requerimientos de entrada al CPU	INICIALIZACION()
enable_IRQ4()	int	Habilita la interrupción de requerimiento de entrada al CPU	INICIALIZACION()
final()	void	Finaliza la ejecución del MSE almacenando variables y cerrando todos los archivos en disco utilizados	MENU_PRINCIPAL()
slfptr()	int	Obtiene el estado de operación de la impresora con llamada directa al BIOS	PRINCIPAL() IMPRIMIR()
printer	int	Imprime una cadena de caracteres por el puerto paralelo LPT1	IMPRIME() IMPRIMIR()

Función	Tipo	Descripción	Utilizada en:
programa_8250()	Int	Programa la velocidad del puerto serie COM2 a 9600 baudios	INI_FECHA()
func_interrupcion()	Int	Transmite la nueva fecha y hora por el puerto serie COM2 de la PC al módulo de la SAC	INI_FECHA()
pet_can()	Int	Solicita al módulo de la SAC hasta los últimos 256 eventos almacenados en BUFFER del subsistema de adquisición	PET_BUFFER()
edo_can()	Int	Solicita al módulo de la SAC el estado actual de los contactos de campo	PET_BUFFER()
util()	Int	Activa el menú de utilerías	MENU_PRINCIPAL()
archivo()	Int	Activa el menú de opciones del manejador de archivos	UTIL()
sel_base()	Int	Activa el menú de opciones del manejador de base de datos	UTIL()
editar()	Int	Permite editar la base de datos que contiene la información de las señales de campo en su totalidad	UTIL()
leer()	Int	Ejecuta la lectura de un archivo dado por el usuario	ARCHIVO()

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
respaldar()	Int	Ejecuta el respaldo de un archivo dado por el usuario	ARCHIVO()
imprimir()	Int	Ejecuta la impresión de un archivo dado por el usuario	ARCHIVO()
dos()	Int	Deja residente en memoria de la PC la ejecución del MSE y permite la ejecución de los comandos del Sistema Operativo DOS desde la línea de comandos	ARCHIVO()
altas()	Int	Añade la información de una señal a la base de datos que contiene las características de las señales de campo	SEL_BASE()
cambios()	Int	Cambia la información de una señal específica de la base de datos que contiene las características de las señales de campo	SEL_BASE()
bajas()	Int	Elimina el registro de una señal específica de la base de datos que contiene las características de las señales de campo	SEL_BASE()

Función	Tipo	Descripción	Utilizada en:
config0	int	Permite al usuario configurar la presentación de la información	UTIL0
diag0	int	Presenta la configuración actual del módulo SAC	MENU_PRINCIPAL0
lim_fecha0	int	Obtiene las fechas límites de los eventos almacenados en ALM_DISK y las presenta	REPORTES0
rep10	int	Genera el reporte de sumario de alarmas en un lapso de tiempo dado por el usuario, el cual, contendrá los eventos en estado de alarma almacenados en ALM_DISK que hayan ocurrido dentro de ese lapso de tiempo	REPORTES0
rep20	int	Genera el reporte lista de leyendas, el cual, contendrá las características de las señales de campo	REPORTES0
rep30	int	Genera el reporte lista de estados actuales, cuyo contenido será el estado actual de los contactos de campo	REPORTES0

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

Función	Tipo	Descripción	Utilizada en:
rep40	Int	Genera los reportes de lista de eventos considerando diversos parámetros dados por el usuario	REPORTES
rep50	Int	Genera el reporte de la configuración de la canasta SAC, el cual indicará la presencia o ausencia de tarjetas en la SAC	REPORTES0
opc_rep0	Int	Activa el menú de opciones para la presentación de los reportes, ya sea en pantalla, impresora o en un archivo de disco	REPORTES0
despliega	Int	Despliega en pantalla un archivo de disco	OPC_REP0
imp_rep0	Int	Imprime el último reporte generado	OPC_REP0
gra_rep0	Int	Almacena en un archivo dado por el usuario el último reporte generado	OPC_REP0

APENDICE B

PSEUDOCÓDIGO DE LAS FUNCIONES QUE INTEGRAN EL SOFTWARE DEL MSE

Este apéndice abarca el pseudocódigo de las funciones básicas que integran el software del MSE.

FUNCIONES DEL SUBSISTEMA DE ADQUISICION (SAC)

El pseudocódigo de las funciones básicas que integran el software del subsistema de adquisición (SAC) se lista a continuación:

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

```
sac_mse()
{
    Inicializa el PIC de la tarjeta SAC 1887
    Init_pic();
    Carga los parámetros iniciales necesarios para el funcionamiento del MSE
    carga();
    Enciende y apaga el led B de la tarjeta de comunicaciones SAC SX-233
    para indicar que el sistema se esta inicializando;
    reset_so();
    set_so();
    Indica al usuario que se inicializa el diagnóstico de hardware
    inicia();
    Diagnóstico de los bancos de memoria RAM de la tarjeta procesadora
    ram_diag();
    Lectura del identificador de la tarjeta y verificación de la presencia de
    conector
    leeide();
    Lectura del identificador de la tarjeta y lo compara con los identificadores validos
    Si el resultado de la comparación es valida ejecuta el diagnóstico para cada tarjeta
    idetar();
    Envía en dos byte la presencia o ausencia de tarjetas y en
    2 bytes la presencia o ausencia de conector
    config();
    Realiza el diagnóstico de la tarjeta de comunicaciones SAC-821
    cuando este presente
    sac_821diag();
    Compara la configuración de identificadores con la tabla de identificadores
    compara();
    Existe o no alguna tarjeta dañada, salta al sistema operativo ( mtos)
    mtos();
}
```

```
principia()
{
    Inicializa la tarjeta
    rset();
    Diagnostica la tarjeta
    diag_mse();
    Actualiza el resultado del diagnóstico y el resultado de la presencia del conector
    en las tablas correspondientes ap_diag, ap_conec
    Carga los datos de datini en la tabla de inicialización de la tarjeta del MSE
    Coloca la máscara de adquisición
    mascara();
    Coloca el tiempo de rebote
    filtro();
    Activa el modo de adquisición continua
    actadq();
}
```

```

diag_mse(tar_dir)
{
    If (SAC-421 en buen estado) {
        if (el UPI esta en buen estado) {
            if (la interfaz al bus esta bien) {
                if (el IBUS-III es correcto) {
                    if ( los puertos de datos del UPI están bien) {
                        res_diag = 1;
                        Obtiene la ausencia o presencia de conector
                        conector = ausencia o presencia de conector;
                    }
                    else
                        res_diag = 0;
                }
            }
            else
                res_diag = 0;
        }
        else
            res_diag = 0;
    }
    else
        res_diag = 0;
}

```

```

inidat()
{
    Programa el contador 2 del temporizador 8254 de la SAC 1887
    prog_cont2();
    arranca el vigilante
    Ejecuta la inicialización de las variables globales
    Inl_global();
    Inicializa el manejador de comunicaciones
    Inicializa el Monitor de Secuencia de Eventos
    inicio()
    arranca las tareas MANEJA_IN(), SIMCOM() y PCEVEN()
    while(1) {
        Envío del mensaje del estado inicial del monitor de secuencia de eventos
        Envío del mensaje con los identificadores detectados en la canasta del MSE
        Envío del mensaje con los diagnósticos de la canasta del MSE
        sal_conec();
        sal_diag();
        for_diag();
        transmite();
        Envío de todos los estados lógicos de los canales de la canasta del MSE
        Envío del mensaje de fin de inicialización.
    }
    fin de la tarea
}

```

servint()

```
{
    Enmascaramiento la interrupción del MTOS
    Se envía la máscara al PIC
    ReadCard421()
    Desenmascara la interrupción al MTOS
    Se envía máscara al PIC
}
```

void ReadCard421()

```
{
    While (existan_interrupción) {
        if (hay espacio en cola EVENTOS) {
            existen_interrupción=0
            /* Lee las 15 tarjetas de entradas digitales*/
            for (contador = 1; contador <=15; contador++) {
                lectura = inp(dir_hw[contador] + 1 & 0x1
                if (lectura == 0x1)
                    /*Almacena en EVENTOS[front1] la información del evento
                    ocurrido*/
                    eventos[front1].tiemp = lee_82540;
                    eventos[front1].hora = HORA;
                    eventos[front1].minuto = MINUTO;
                    eventos[front1].segund = SEGUND;
                    eventos[front1].dia= DIA;
                    eventos[front1].mes = MES;
                    eventos[front1].year = YEAR,
                    directarjeta = dir_hw[contador];
                    eventos[front1].ranura = contador;
                    eventos[front1].bytelow = inp(directarjeta)
                    eventos[front1].bytehigh = inp(directarjetas);
                    front1++;
                }
            }
        }
    }
}
```

lee_8254()

```
{
    Direccionamiento del puerto de control del 8254
    Se congela la cuenta de milisegundos
    Direccionamiento del puerto de datos del 8254
    Lectura del puerto de datos del 8254
    Almacenamiento del byte en low_time
    Lectura del puerto de datos del 8254
    Almacenamiento del byte en high_time
    tiempo_mseg = high_time;
    tiempo_mseg << 8;
    tiempo_mseg = tiempo_mseg | low_time;
    tiempo_mseg = 1000 - tiempo_mseg;
    return (tiempo_mseg);
}
```

hora()

```
{
    SEGUND++,
    if (SEGUND == 60) {
        SEGUND = 0;
        MINUTO++;
        if (MINUTO == 60) {
            MINUTO = 0,
            HORA++;
            if (HORA == 24) {
                HORA = 0;
                DIA++;
                if (DIA == 32) {
                    DIA = 1;
                    MES++;
                    if (MES == 13) {
                        MES = 1;
                        YEAR++;
                        if (YEAR == 100) {
                            YEAR = 0;
                        }
                    }
                }
            }
        }
    }
}
```

```

void maneja_int()
{
    for (;;) {
        while (recuo_even() != 0) /* Mientras haya eventos en EVENTOS*/
            time_even = eventos.tiemp;
            slot = eventos.ranura;
            partebaja = eventos.bytelow;
            partealta = eventos.bytehigh;
            verifica que exista un cambio de estado en la parte baja de la tarjeta
            if (detecta_cambios != 0) {
                while (detecta_cambios > 0) {
                    Obtiene el número de canal en que se detectó
                    el cambio
                    canal = numero_canal();
                    En iden coloca (Tarjeta/Señal)
                    iden = slot | canal;
                    obtiene el estado edo de la parte baja de la tarjeta
                    Actualiza la tabla dat_inl con el estado de la parte baja
                    Almacena en POINTER la información de los eventos
                    if (hay espacio en cola POINTER) {
                        pointer[front].hora = eventos.hora;
                        pointer[front].minuto = eventos.minuto;
                        pointer[front].segund = eventos.segund;
                        pointer[front].dia = eventos.dia;
                        pointer[front].mes = eventos.mes;
                        pointer[front].year = eventos.year;
                        pointer[front].tiempo_evento = time_even;
                        pointer[front].estado = edo;
                        pointer[front].identificador = iden;
                        front++;
                    }
                    if (POINTER llegó al máximo)
                        front = 0;
                }
            }
        }
        Verifica que exista un cambio de estado en la parte alta de la tarjeta
        if (detecta_cambios != 0) {
            while (detecta_cambios > 0) {
                Obtiene el número de canal en que se detectó
                el cambio
                canal = numero_canal();
                En iden coloca (Tarjeta/Señal)
                iden = slot | canal;
                obtiene el estado edo de la parte baja de la tarjeta
                Almacena en POINTER la información de los eventos
                if (hay espacio en cola POINTER) {
                    pointer[front].hora = eventos.hora;
                    pointer[front].minuto = eventos.minuto;
                    pointer[front].segund = eventos.segund;
                    pointer[front].dia = eventos.dia;
                    pointer[front].mes = eventos.mes;
                }
            }
        }
    }
}

```


SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

```
        lee_buffer();
        peticion = '0';
    }
}
Fin de la tarea
}

almacena_buffer()
{
    if (hay espacio en BUFFER) {
        buffer[front2].hora = pointer.hora;
        buffer[front2].minuto = pointer.minuto;
        buffer[front2].segund = pointer.segund;
        buffer[front2].dia = pointer.dia;
        buffer[front2].mes = pointer.mes;
        buffer[front2].year = pointer.year;
        buffer[front2].tiempo_evento = pointer.tiempo_evento;
        buffer[front2].estado = pointer.estado;
        buffer[front2].Identificador = pointer.Identificador;
        if (buffer llegó al máximo)
            front2 = 0;
    }
}

lee_buffer()
{
    secuencia = 0;
    Recupera los eventos almacenados en BUFFER
    y los coloca en el área de memoria PET_USR
    while (haya eventos en BUFFER) {
        &pet_usr.hora = buffer.hora;
        &pet_usr.minuto = buffer.minuto;
        &pet_usr.segund = buffer.segund;
        &pet_usr.dia = buffer.dia;
        &pet_usr.mes = buffer.mes;
        &pet_usr.year = buffer.year;
        &pet_usr.tiempo_evento = buffer.tiempo_evento;
        &pet_usr.estado = buffer.estado;
        &pet_usr.Identificador = buffer.Identificador;
        &pet_usr++;
        secuencia++;
    }
    Pregunta si existe información en pet_usr
    if (secuencia > 0) {
        Calcula el tamaño de la trama a enviar almacenada en pet_usr
        Se envia la trama
        envia1();
    }
}
```

envia(DataPtr,NumEvents)

```
{
    Envío del encabezado de la trama
    FF EE DD CC
    Envío del tamaño de la trama
    NumEvents
    Envío de los eventos almacenados en PC, cuyo formato es:
        AA ME DD HH MI SS msL msM T/S EDO = 10 bytes
    for (ii = 0; ii == NumEvents; ii++, DataPtr++) {
        Envío de DataPtr
    }
    Envío del fin de trama
    BB AA 99 88
}
```

envia1(DataPtr,NumEvents)

```
{
    Envío del encabezado de la trama
    77 66 55 44
    Envío del tamaño de la trama
    NumEvents
    Envío de los eventos almacenados en PET_USR, cuyo formato es:
        AA ME DD HH MI SS msL msM T/S EDO = 10 bytes
    for (ii = 0; ii == NumEvents; ii++, DataPtr++) {
        Envío de DataPtr
    }
    Envío del final de la trama
    33 22 11 00
}
```

SIMCOM()

```
{
    for (;;) {
        enciende led
        realiza una pausa de un segundo
        apaga led
    }
    fin de la tarea
}
```

FUNCIONES DEL SUBSISTEMA DE PRESENTACION (PC)

El pseudocódigo de las funciones básicas que integran el software del subsistema de presentación (PC) se lista a continuación:

```
principal()
{
    Almacena la información de las señales en un área de memoria
    lee_señales();
    Realiza la presentación de la pantalla principal
    Presentación();
    Presenta el menú de opciones principal
    let_menu_principal();
    Realiza la inicialización de hardware();
    Inicializa();
    El proceso principal es:
    while (b) {
        if (ap_lec != ap_esc) {
            ap_lec++;
            if (ap_lec == MAX_BUF_SIZE)
                ap_lec = 0;
            pregunta el estado de la impresora
            edo_imp = stlptr();
            if (impresora en línea) {
                formatea la información recibida
                formatea();
                presenta la información al usuario en pantalla e impresora
                despliega();
                almacena();
                imprime();
            }
            else {
                formatea la información recibida
                formatea();
                Presenta la información al usuario únicamente en pantalla
                despliega();
                almacena();
            }
        }
        if (ap_lec == ap_esc) {
            cierra todos los archivos de almacenamiento en uso
            fclose_all();
            guarda_apuntadores();
        }
        if (se pulsa cualquier tecla) {
            se activa el menú principal
            menu_principal();
        }
    }
}
```

```

    }
}

```

menu_principal()

```

{
    if (opcion == ESC)
        finaliza la ejecución del MSE
    if (opcion == 'I') {
        ejecuta la inicialización de la fecha del subsistema de adquisición
        ini_fecha();
    }
    if (opcion == 'd') {
        Ejecuta la petición de la información almacenada en el buffer del
        subsistema de adquisición o el estado actual de los canales
        pres_dat();
        solicita el nombre del archivo en donde se almacenará la información
        pet_arch();
        ejecuta la selección de la información a pedir y la lleva a cabo
        sel_pet_can();
    }
    if (opcion == 'u') {
        Ejecuta el menú de utilerías
        pres_util();
        util();
    }
    if (opcion == 'c')
        Presenta la configuración actual del módulo SAC
        pres_conf();
        conf();
    }
    if (opcion == 'h') {
        despliega el archivo ALM_DISK
        despliega(ALM_DISK);
    }
    if (opcion es diferente a las anteriores)
        manda mensaje de error
}

```

void Inicialización()

```
{
    Deshabilita la interrupción de requerimientos de entrada al CPU
    disable_IRQ40;
    Inicializa la velocidad y características del puerto serie COM1
    set_baud_rate();
    set_carac_line();
    Lee los vectores de interrupción y activa la función RECIBE()
    set_vects();
    Habilita la interrupción de requerimientos de entrada al CPU
    Enable IRQ40;
}
```

Ini_fecha()

```
{
    limpia_vent();
    Petición al usuario de la nueva hora y fecha
    lee(hora_u, minuto_u, segundo_u);
    lee(hora_u, minuto_u, segundo_u);
    Convierte a hexadecimal la nueva fecha y hora
    conv_hex(hora_u, minuto_u, segundo_u);
    conv_hex(hora_u, minuto_u, segundo_u);
    Inicializa la velocidad del puerto serie COM2
    programa_8250();
    Transmite la hora y fecha al subsistema de adquisición
    func_interrupción();
}
```

pet_buffer() *

```
{
    limpia_pantalla();
    let_menu_pet_buffer();
    Solicita al usuario el nombre del archivo
    pet_arch();
    Selección de información a solicitar y activación de las funciones correspondientes
    lee(opcion);
    if (opcion == 1) {
        Solicita los eventos almacenados en BUFFER
        pet_can();
    }
    if (opcion == 2) {
        Solicita el estado actual de los contactos de campo
        edo_can();
    }
}
```

util()

```
{
    limpia_pantalla();
    let_menu_util();
    Selección de opción a ejecutar
    lee(opcion);
    switch (opcion) {
        case 'a':    Activa el menú manejador de archivos
                    archivo();
                    break;
        case 'b':    Activa el menú del manejador de base de datos
                    sel_base();
                    break;
        case 'p':    Ejecuta la función que permite al usuario configurar la
                    información a presentar
                    config();
                    break;
        case 'r':    Activa la función de generación de reportes();
                    reportes();
                    break;
    }
}
```

archivo()

```
{
    limpia_pantalla();
    let_menu_archivo();
    Selección de opción a ejecutar
    lee(opcion);
    switch (opcion) {
        case 'l':    Llama la función que realiza la lectura de un archivo
                    leer();
                    break;
        case 'i':    Llama la función que realiza la impresión de un archivo
                    imprimir();
                    break;
        case 'r':    Llama la función que realiza el respaldo de un archivo
                    respaldar();
                    break;
        case 'd':    Permite ejecutar los comandos del Sistema Operativo desde
                    la línea de comandos
                    dos();
                    break;
    }
}
```

sel_base()

```
{
    limpia_pantalla();
    let_menu_base_datos();
    Selección de opción a ejecutar
    lee(opcion);
    switch (opcion) {
        case 'a' :    Llama la función que añade información a la base de datos
                    altas();
                    break;
        case 'c' :    Llama la función que cambia la información de una señal
                    específica
                    cambios();
                    break;
        case 'b' :    Llama la función que elimina información de la base de datos
                    bajas();
                    break;
        case 'e' :    Llama la función que permite editar la base de datos completa
                    editar();
    }
}
```

reportes()

```
{
    limpia_pantalla();
    let_reportes_a_generar();
    Selección de reporte a generar
    lee(opcion);
    switch(opcion) {
        case '1' :    Presenta las fechas límites permitidas para la generación del
                    reporte de sumario de alarmas
                    lim_fecha();
                    solicita al usuario las fechas para su reporte
                    fech_rep();
                    genera el reporte
                    rep_1();
                    Presentación del reporte generado considerando la forma en
                    que el usuario desea visualizarlo
                    opc_rep();
                    break;
        case '2' :    Generación del reporte de la Lista de leyendas
                    rep2();
                    Presentación del reporte generado considerando la forma en
                    que el usuario desea visualizarlo
                    opc_rep();
                    break;
        case '3' :    Solicita el estado actual de los contactos de campo al módulo
                    de la SAC
                    edo_can();
                    Generación del reporte lista del estado actual de los canales
    }
}
```



```

        rep3();
        Presentación del reporte generado considerando la forma en
        que el usuario desea visualizarlo
        opc_rep0;
        break;
    case'4' :   Generación de reportes considerando diversos parámetros
                dados por el usuario.
                Presenta las fechas límites permitidas para la generación del
                reporte
                lim_fecha0;
                solicita la opción del reporte a generar
                opc_rep40;
                Presentación del reporte generado considerando la forma en
                que el usuario desea visualizarlo
                opc_rep0;
                break;
                break;
    case '5' :   Genera el reporte de la configuración actual de las canasta
                rep50;
                Presentación del reporte generado considerando la forma en
                que el usuario desea visualizarlo
                opc_rep0;
                break;
}
}

```

opc_rep4()

```

{
    limpia_pantalla0;
    let_reportes_lista_eventos0;
    Selección de reporte a generar
    lee(opcion);
    switch(opcion) {
        case '1' :   Presenta las fechas límites permitidas para la generación del
                    reporte
                    lim_fecha0;
                    solicita al usuario las fechas para la generación de la lista de
                    eventos ocurridos entre dos fecha
                    fech_usu0;
                    generación del reporte
                    rep_410;
                    Presentación del reporte generado considerando la forma en
                    que el usuario desea visualizarlo
                    opc_rep0;
                    break;
        case '2' :   Presenta las fechas límites permitidas para la generación del
                    reporte
                    lim_fecha0;
                    solicita al usuario los datos para la generación de la lista de
                    eventos ocurridos en una fecha y un lapso de tiempo

```

SISTEMA DE MONITOREO Y REGISTRO DE SECUENCIA DE EVENTOS (MSE)

```
    fech_usu();
    tiempo_usu();
    generación del reporte
    rep_42();
    Presentación del reporte generado considerando la forma en
    que el usuario desea visualizarlo
    opc_rep();
    break;
case '3' : Presenta las fechas límites permitidas para la generación del
    reporte
    lim_fecha();
    solicita al usuario los datos para la generación de la lista de
    eventos ocurridos en una fecha, un lapso de tiempo y una
    tarjeta determinada
    fech_usu();
    tiempo_usu();
    tarjeta_usu()
    generación del reporte
    rep_43();
    Presentación del reporte generado considerando la forma en
    que el usuario desea visualizarlo
    opc_rep();
    break;
case '4' : Presenta las fechas límites permitidas para la generación del
    reporte
    lim_fecha();
    solicita al usuario los datos para la generación de la lista de
    eventos ocurridos en una fecha, un lapso de tiempo y una
    señal específica
    fech_usu();
    tiempo_usu();
    tarjeta_usu();
    señal_usu();
    generación del reporte
    rep_44();
    Presentación del reporte generado considerando la forma en
    que el usuario desea visualizarlo
    opc_rep();
    break;
case '5' : Presenta las fechas límites permitidas para la generación del
    reporte
    lim_fecha();
    solicita al usuario los datos para la generación de la lista de
    eventos ocurridos en una fecha, un lapso de tiempo y una
    prioridad de atención
    fech_usu();
    tiempo_usu();
    prioridad_usu();
    generación del reporte
    rep_45();
    Presentación del reporte generado considerando la forma en
```

```

que el usuario desea visualizarlo
opc_rep();
break;
case '6' : Presenta las fechas límites permitidas para la generación del
reporte
lim_fecha();
solicita al usuario los datos para la generación de la lista de
eventos ocurridos en una fecha, un lapso de tiempo, la
prioridad de atención y una tarjeta determinada
fech_usu();
tiempo_usu();
prioridad_usu();
tarjeta_usu();
generación del reporte
rep_46();
Presentación del reporte generado considerando la forma en
que el usuario desea visualizarlo
opc_rep();
break;
case '7' : Presenta las fechas límites permitidas para la generación del
reporte
lim_fecha();
solicita al usuario los datos para la generación de la lista de
eventos ocurridos entre dos fecha, un lapso de tiempo, una
prioridad de atención y una señal específica
fech_usu();
tiempo_usu();
prioridad_usu();
tarjeta_usu();
señal_usu();
generación del reporte
rep_47();
Presentación del reporte generado considerando la forma en
que el usuario desea visualizarlo
opc_rep();
break;
case '8' : Generación de la lista de eventos considerando todos los
parámetros con valores predeterminados y que pueden ser
cambiados a selección del usuario aquellos que él
seleccione
rep_48();
Presentación del reporte generado considerando la forma en
que el usuario desea visualizarlo
opc_rep();
break;

```