

28
2e;

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGON



DESARROLLO DE UN CONTROLADOR
INTELIGENTE DE COMUNICACIONES EN
COMPUTADORA PERSONAL

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :

ENRIQUE SANTACRUZ SAUCEDO VILLALBA



ENEP
ARAGON

Escuela Nacional de estudios profesionales
aragón

TESIS CON
FALLA DE ORIGEN

1993



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.

INDICE

INTRODUCCION.....	i
1 ESTACION MAESTRA DE CONTROL SUPERVISORIO.....	1
1.1 Componentes de la Estación Maestra	2
1.1.1. Equipo de cómputo.....	3
1.1.2. Subsistema de Entrada/Salida.....	4
1.2 Programas de Cómputo de la Estación Maestra.....	5
1.2.1. Sistema Operativo.....	5
1.2.2. Adquisición de datos.....	6
1.2.3. Base de Datos.....	7
1.2.4. Interfaz Hombre-Máquina.....	7
1.2.5. Diagnósticos.....	8
1.3 Estación Maestra MAC-5000.....	9
1.3.1. Sistema de cómputo MAC.....	10
1.3.2. Controlador de Comunicaciones CIC.....	11
1.4 Estación Maestra Distribuida EMIX-V1.....	11

2	CONTROLADOR INTELIGENTE DE COMUNICACIONES.....	14
2.1.	Descripción General.....	15
2.2.	Arquitectura.....	16
2.2.1.	Configuración General.....	16
2.2.2.	Tarjeta de comunicaciones PC-851.....	17
2.2.3.	Modems.....	19
2.3.	Funciones.....	20
2.3.1.	Funciones del módulo maestro.....	21
2.3.2.	Funciones de los módulos esclavos.....	22
2.4.	Base de datos.....	24
2.4.1.	Módulo maestro.....	24
2.4.2.	Módulos esclavos.....	25
3	ESPECIFICACION Y DISEÑO DEL MODULO MAESTRO DEL CIC.....	27
3.1.	Descripción del modelo.....	28
3.2.	Modelo fundamental.....	29
3.2.1.	Operación del sistema.....	29
3.2.2.	Definición de Procesos.....	30
3.2.3.	Definición de Buzones.....	33
3.2.4.	Definición de Mensajes.....	37
3.2.5.	Definición de Estructuras de Datos.....	54
3.3.	Diagramas de estructuras de los procesos.....	60
3.3.1.	Proceso Raíz.....	60
3.3.2.	Proceso Distribuidor.....	63
3.3.3.	Proceso Concentrador.....	70
3.3.4.	Proceso Estadísticas.....	76
3.3.5.	Proceso Manejador de errores.....	82
3.3.6.	Proceso de enlace al sistema de cómputo (lectura).....	84
3.3.7.	Proceso de enlace al sistema de cómputo (escritura).....	86
3.3.8.	Proceso de enlace hacia las UTR (lectura).....	88
3.3.9.	Proceso de enlace hacia las UTR (escritura).....	88

4	PLAN DE PRUEBAS Y VERIFICACION FUNCIONAL DEL MODULO MAESTRO DEL CIC.....	91
4.1	Proceso de pruebas de software.....	92
4.2	Plan de pruebas para el CIC - V2.....	93
4.2.1	Objetivos y alcance del plan de pruebas.....	93
4.2.2	Elementos de prueba.....	93
4.2.3	Características probadas y excluidas.....	94
4.2.4	Técnicas	95
4.2.5	Criterios de aceptación y rechazo	99
4.2.6	Criterios de suspensión y requerimientos de continuación.....	99
4.2.7	Requerimientos.....	100
4.3	Casos de prueba.....	101
4.3.1	Proceso concentrador.....	101
4.3.2	Proceso estadísticas	104
4.3.3	Proceso distribuidor.....	105
4.3.4	Sistema integrado	105
4.4	Resultados de las pruebas.....	111
	CONCLUSIONES.....	120
	REFERENCIAS.....	123
	APENDICES	
A	CARACTERISTICAS DE LAS HERRAMIENTAS DE DESARROLLO.....	125
A.1	System Architect.....	126
A.2	TopSpeed C	128

B	METODOLOGIA DE DESARROLLO	132
B.1.	La fase de desarrollo de programas de cómputo	133
B.2.	Metodologías de diseño de programas de cómputo.....	134
B.2.1.	Modelado de procesos (Diagrama de Flujo de Datos).....	135
B.2.2.	Modelo Entidad - Relación	135
B.2.3.	Diagramas de estructura.....	136
B.2.4.	Diagramas de descomposición.....	136
B.2.5.	Sistemas de tiempo real.....	137
B.3.	Areas de aplicación del análisis orientado al flujo de datos	137
B.4.	Método de análisis orientado al flujo de datos.....	138
B.4.1.	Diagramas de flujo de datos (DFD).....	139
B.4.2.	Extensiones del modelo de flujo de datos	143
B.5.	Método propuesto para diseño de programas de cómputo	148
B.5.1.	Consideraciones en el proceso de diseño	150
C	GLOSARIO DE TERMINOS.....	151

INDICE DE FIGURAS

FIGURA 1.1 Configuración general de un sistema de Control Supervisorio	2
FIGURA 1.2 Diagrama a bloques de un sistema de Control Supervisorio	3
FIGURA 1.3 Componentes de la Estación Maestra	4
FIGURA 1.4 Arquitectura general de la Estación Maestra MAC-5000	10
FIGURA 1.5 Configuración de la Estación Maestra EMIX - V1	13
FIGURA 2.1 Arquitectura básica del CIC-V2	16
FIGURA 2.2 Configuración del CIC-V2	17
FIGURA 2.3 Arquitectura de la tarjeta de comunicaciones PC-851	18
FIGURA 2.4 Modelo de comunicaciones OSI/ISO	18
FIGURA 2.5 Esquema de direccionamiento para los módulos esclavos	19
FIGURA 2.6 Protocolos orientados a bit y a byte	20
FIGURA 3.1 Modelo fundamental del módulo maestro del CIC-V2	31
FIGURA 3.2 Manejo de buzones	34
FIGURA 3.3 Proceso Raíz	61
FIGURA 3.4 Proceso Distribuidor	64
FIGURA 3.5 Proceso Concentrador	71
FIGURA 3.6 Proceso Estadísticas	77
FIGURA 3.7 Proceso Manejador de errores	83
FIGURA 3.8 Proceso de enlace al sistema de cómputo (lectura)	85
FIGURA 3.9 Proceso de enlace al sistema de cómputo (escritura)	87
FIGURA 3.10 Proceso de enlace hacia las UTR (lectura)	89
FIGURA 3.11 Proceso de enlace hacia UTR (escritura)	90

FIGURA 4.1	Prueba de funcionamiento de la tarjeta PC-851	97
FIGURA 4.2	Ruta sugerida para prueba del sistema (caso 1).....	107
FIGURA 4.3	Ruta sugerida para prueba (caso 2)	108
FIGURA 4.4	Ruta sugerida para pruebas (caso 3).....	109
FIGURA 4.5	Ruta sugerida para prueba (caso 4)	109
FIGURA 4.6	Ruta sugerida para prueba (caso 5)	110
FIGURA B.1	Fase de desarrollo	133
FIGURA B.2	Flujo de información	138
FIGURA B.3	Diagrama de flujo de datos.....	139
FIGURA B.4	Simbología.....	140
FIGURA B.5	DFD de llamada Telefónica (a)nivel 01 (b) nivel 02	141
FIGURA B.6	Refinamientos del flujo de información.....	142
FIGURA B.7	Ejemplo de notación Ward & Mellor.....	144
FIGURA B.8	Flujo de eventos.....	145
FIGURA B.9	Flujo de eventos y transformación de control	146
FIGURA B.10	Combinando Transformaciones de Datos y Control.....	147
FIGURA B.11	Casos equivalentes.....	147
FIGURA B.12	Diagrama de transición de estados.....	148

INTRODUCCION

Con el propósito de mejorar la calidad, la continuidad del servicio a los usuarios y lograr una operación más eficiente de los sistemas eléctricos, se están integrando a nivel nacional equipos de control supervisorio y adquisición de datos. Estos sistemas pueden controlar los procesos de transmisión, generación y distribución de la energía eléctrica desde una pequeña región hasta un área geográficamente extensa; además, los datos adquiridos pueden proporcionar información a sistemas avanzados de administración de la energía en los centros de operación del CENACE (Centro Nacional de Control de Energía) de la Comisión Federal de Electricidad.

La implementación de un sistema supervisorio para el control y supervisión de la energía eléctrica requiere de la instalación de Unidades Terminales Remotas (UTR) localizadas junto a los elementos de campo tales como: relevadores, sensores, transductores, medidores y otros dispositivos electrónicos, en las distintas subestaciones, plantas, alimentadores, etc.; y de Estaciones Maestras localizadas en los centros de control que reciben los datos de las UTR mediante una amplia variedad de canales de comunicación; a su vez ejecutan acciones de control hacia los diferentes dispositivos que lo requieran.

En el Departamento de Electrónica del Instituto de Investigaciones Eléctricas (IIE) se han desarrollado equipos de control supervisorio (EM y UTR). Recientemente se desarrolló una Estación Maestra denominada MAC-5000 para el control de subáreas de distribución de la Comisión Federal de Electricidad y Compañía de Luz y Fuerza del Centro (Morelia y Toluca). Actualmente se tiene en desarrollo una Estación Maestra Distribuida denominada EMIX-VI basada en sistemas abiertos.

Un sistema supervisorio como el descrito anteriormente implica la adquisición intensiva de datos, lo cual requiere a su vez una red de comunicaciones flexible y eficiente. Los canales de comunicación limitan la velocidad a la que la adquisición de datos y el control se pueden ejecutar; por lo que se requiere de un diseño adecuado del equipo y de los programas de comunicaciones de tal manera que garanticen que la información se transfiera correctamente de la terminal remota a la Estación Maestra y viceversa.

El sistema supervisorio desarrollado en el IIE se compone básicamente de un subsistema de procesamiento central que incluye la interfaz hombre-máquina y de un subsistema de comunicaciones, denominado Controlador Inteligente de Comunicaciones (CIC). El equipo del CIC está integrado actualmente por módulos propietarios de tipo MAC desarrollados en el propio IIE; consta de una tarjeta maestra de procesamiento MAC-1186 y módulos esclavos inteligentes MAC 851, uno encargado de comunicarse al sistema de cómputo MAC y los otros como puertos de comunicación para las UTR o como enlace a una Estación Maestra de nivel superior.

La intención de este trabajo de tesis es realizar toda la programación del módulo maestro del CIC para que se ejecute en un equipo comercial, basado en computadora personal compatible con IBM, integrándole las tarjetas esclavas PC-851 (equivalentes a las MAC-851) y conectándolas a través del bus interno de la PC, con el propósito de contar con un equipo que ofrezca otras opciones de comunicación para la Estación Maestra MAC-5000 y contar además con el soporte adecuado de comunicaciones para la Estación Maestra Distribuida EMIX-VI con características de sistemas abiertos.

Otro de los objetivos es la evaluación del ambiente de desarrollo y compilador Top Speed C con manejo integrado de concurrencia, así como del paquete de CASE llamado "System Architect" utilizado para la elaboración del diseño. Dado que este diseño es parte de un sistema más grande se requiere de una herramienta en la que se sinteticen todos los requerimientos del CIC y se realice un balance contra los requerimientos del sistema entero. Para efectuar tales acciones se utilizó esta herramienta con el fin de ajustarse a una metodología de diseño y realizar un desarrollo formal para el sistema.

Esta tesis está organizada de la siguiente manera:

En el capítulo uno se proporciona una breve explicación de los sistemas supervisorios haciendo énfasis en la Estación Maestra, describiendo las partes que la constituyen (equipo y programas) y ejemplificando con las Estaciones Maestras MAC-5000 y EMIX-V1.

El capítulo dos trata los conceptos básicos acerca del Controlador Inteligente de Comunicaciones, indicando las funciones que ejecuta dentro del sistema supervisorio, tanto en lo correspondiente al módulo maestro del CIC, como en los módulos esclavos que conforman la arquitectura diseñada.

El capítulo tres contiene las especificaciones y el diseño del módulo maestro del CIC, obtenido con la herramienta de CASE llamada "System Architect" y utilizando la metodología de diseño orientado al flujo de datos con extensiones para tiempo real denominada "Data Flow Ward & Mellor".

El capítulo cuatro contiene el plan de pruebas, la verificación funcional del módulo maestro del CIC y el resultado de las pruebas efectuadas.

Se presentan las conclusiones del trabajo desarrollado y por último los apéndices y las referencias

En el apéndice A se describe las características de las herramientas empleadas para la elaboración de la especificación y diseño del módulo maestro del CIC y del ambiente de desarrollo utilizado para la realización práctica del sistema.

El apéndice B muestra las bases de la metodología utilizada para la elaboración de la especificación y el diseño del módulo maestro bajo la metodología conocida como "Data Flow Ward & Mellor", explicando mediante ejemplos sencillos su utilización en sistemas de tiempo real.

El apéndice C es un glosario de los términos y abreviaciones más frecuentemente utilizadas durante toda la tesis.

Capítulo 1

ESTACION MAESTRA DE CONTROL SUPERVISORIO

Este capítulo es una introducción a las estaciones maestras utilizadas en control supervisorio, describiendo las partes que componen el equipo y los programas de cómputo utilizados, ejemplificando con la Estación Maestra MAC-5000 desarrollada en el Instituto de Investigaciones Eléctricas y la Estación Maestra distribuida EMIX-V1 en desarrollo actualmente.

Un sistema de control supervisorio y adquisición de datos (SCADA por sus siglas en inglés) es el encargado de adquirir y concentrar la información en tiempo real de los dispositivos de campo para su supervisión y de realizar el control remoto de los dispositivos, ya sea mediante un operador o de manera automática. Se compone básicamente de una o varias Estaciones Maestras (EM) que supervisan y controlan a una o más Unidades Terminales Remotas (UTR), las cuales se enlazan directamente con los elementos de campo (sensores y actuadores). La configuración más utilizada consiste de una Estación Maestra y múltiples Unidades Terminales Remotas. La EM se localiza en un centro de control desde el cual se comunica con las UTR mediante un canal de comunicación previamente establecido (figura 1.1 y figura 1.2).

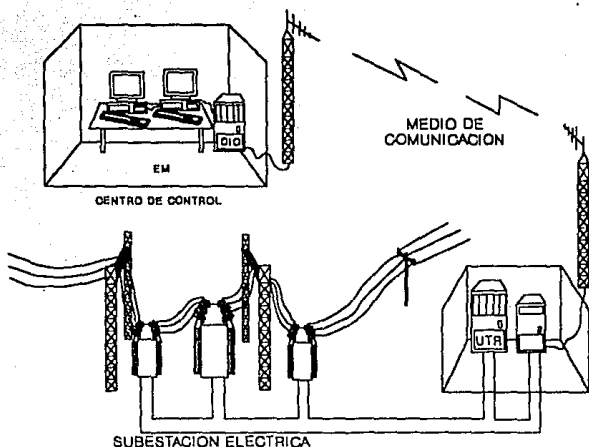


FIGURA 1.1 Configuración general de un sistema de Control Supervisorio

1.1. Componentes de la Estación Maestra

La Estación Maestra constituye un sistema de cómputo donde se centraliza y procesa, en tiempo real, la información de la red eléctrica adquirida por las UTR, y la presenta de manera eficiente al operador para tomar las decisiones adecuadas; asimismo, permite ejercer acciones de control sobre los componentes de la red.

En sistemas modernos, la EM se compone típicamente de varias computadoras, periféricos y subsistemas de entrada/salida (comunicaciones) que permiten a los operadores del centro de control supervisar el estado de los sistemas de potencia y controlarlos. El equipo de la estación maestra provee facilidades al operador para realizar la supervisión de distintas maneras: visualmente a través de pantallas o monitores (CRT), con alarmas audibles ó por medio de impresiones [IEE 91]. La figura 1.3 muestra los componentes de una Estación Maestra.

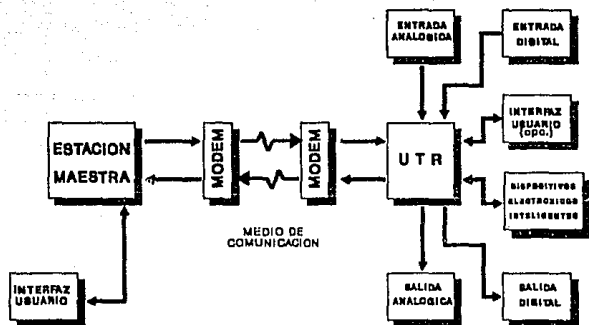


FIGURA 1.2 Diagrama a bloques de un sistema de Control Supervisorio

1.1.1. Equipo de cómputo

El sistema de cómputo de una Estación Maestra varía desde una computadora personal para sistemas pequeños hasta una red de estaciones de trabajo para sistemas medianos y grandes, pasando por sistemas dedicados basados en microprocesadores.

Existen tendencias que están marcando la pauta en el desarrollo e implementación de sistemas de control supervisorio, de las cuales las más sobresalientes son [VAL 92]:

- a) La implementación de redes de control distribuido, lo cual permite tener un sistema modular y aumentar la confiabilidad. La tendencia actual es utilizar redes de estaciones de trabajo ("workstations") basadas en arquitecturas de cómputo que utilizan procesadores RISC.
- b) Existe una predominancia definitiva en el empleo de sistemas abiertos lo que permite la portabilidad, interoperabilidad y consistencia en la interfaz con el usuario a través de diferentes plataformas. La norma a seguir es la IEEE POSIX, basada en el sistema operativo UNIX y la interfaz gráfica X-Windows,

con el consiguiente empleo de arquitecturas Cliente-Servidor y software comercial.

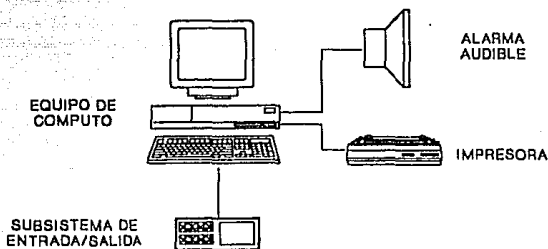


FIGURA 1.3 Componentes de la Estación Maestra

1.1.2. Subsistema de Entrada/Salida

Los subsistemas de entrada/salida generalmente soportan dispositivos con salida serie ó paralelo. Esos dispositivos son dispositivos inteligentes controlados por computadora, cuya función es liberar al sistema de cómputo de las tareas de adquisición de datos y comunicación hacia las UTR, y hacia otras estaciones maestras, por esta razón se le conoce también como controlador de comunicaciones.

El controlador de comunicaciones maneja la comunicación con los dispositivos de enlace a las UTR. Los medios de comunicación pueden ser por diferentes vías, las más comunes son: teléfono, radio, onda portadora en líneas de transmisión o distribución, fibra óptica, etc.

La comunicación se establece por medio de algún protocolo; existen muchos protocolos disponibles, la mayoría propietarios, las UTR modernas son capaces de ser programadas para trabajar con una amplia variedad de protocolos. La tendencia actual se enfoca a la realización de un protocolo estandar basado en las normas ISO/OSI para facilitar la comunicación con EM y UTR de diversos fabricantes.

1.2. Programas de Cómputo de la Estación Maestra

La programación del sistema de control supervisorio esta compuesto de un conjunto de programas de aplicación operando en conjunto con el sistema operativo multitareas y con la base de datos. Los programas requeridos pueden estar distribuidos en varios procesadores o confinados a un solo procesador (procesamiento distribuido o centralizado). Las siguientes subsecciones describen algunos programas básicos de aplicación de control supervisorio.

1.2.1. Sistema Operativo

El sistema operativo tiene como objetivo aislar a las tareas de aplicación de los detalles del hardware y permitir que los recursos (procesador, memoria, disco, etc.) del equipo de cómputo sean utilizados y compartidos en una forma eficiente.

Las arquitecturas de cómputo tradicionales no son las más eficientes para aplicaciones en tiempo real. El rendimiento de una aplicación de este tipo, depende no solo de la capacidad de procesamiento del CPU, sino también de la capacidad de atención de interrupciones externas y del manejo eficiente de los dispositivos de entrada/salida [MOL 90]

El sistema SCADA realiza la función de muestrear las UTR y procesar la información enviándola o recibéndola de los enlaces de comunicación. Los operadores interactúan simultáneamente con el sistema ya sea con diferentes pantallas, reaccionando ante múltiples alarmas, así como otras actividades. Todas estas actividades solo pueden ser manejadas por un sistema operativo en tiempo real. Cada fabricante de computadoras usualmente diseña su propio sistema operativo y lo promociona con el uso de su equipo. Existe una tendencia actual en la industria en utilizar el sistema operativo UNIX con extensiones para tiempo real como se propone en la norma POSIX IEEE P1003.1 y P1003.4. Otra de las tendencias es que los fabricantes estan personalizando el conjunto de instrucciones utilizado con su sistema operativo para ganar mayor velocidad y de esta manera ejecutar sus aplicaciones mucho más rápido. Estos son llamados sistemas RISC (Reduced Instruction Set Computing).

1.2.2. Adquisición de datos

La función de adquisición de datos ejecuta las siguientes tareas:

- Función de muestreo

Cuando los datos de la UTR se reciben, esta función se encarga de analizar cada punto. Para cada punto, determina si su estado ha cambiado, y de ser así, lo notifica a la aplicación del procesamiento de alarmas. Este programa entonces alerta al operador y ejecuta algunas otras funciones. De manera similar, los puntos analógicos se verifican contra sus límites y las funciones apropiadas son notificadas de ser necesario.

Esta función incluye la necesidad de programar un tiempo de adquisición a cada UTR muestreada. Existen varios tipos de muestreo usados en una secuencia a cada UTR y los puntos son de diferente índole (entradas digitales o de estado, entradas analógicas, contadores de eventos digitales llamados "acumuladores", puntos de control digital, etc.). Por ejemplo, un muestreo cada 2 segundos, es típico para todos los puntos digitales y 1 de cada 5 puntos analógicos de tal manera que cada 10 segundos todos los puntos analógicos son leídos. [IEE 91]

Otra técnica es la llamada exploración por excepción para puntos analógicos y digitales en la que los datos se obtienen solo cuando cambian. Una exploración de integridad se realiza periódicamente para asegurarse que los datos estén correctos.

Como funciones especiales los datos de acumuladores de pulsos son leídos y procesados.

- Secuencia de eventos

La secuencia de eventos (SOE) es una aplicación especial que es coordinada en cada exploración, si es requerida. La UTR almacena cambios de estado con una resolución de un milisegundo. Así los datos de la SOE estarán disponibles al operador para cuando los requiera.

- Estadísticas de comunicación

Esta función permite al operador contar con información estadística de las comunicaciones hacia las terminales remotas. Por cada UTR se lleva un conteo del número de mensajes enviados y contestados correcta e incorrectamente, así como de los tiempos acumulados (dentro/fuera de exploración, tipo de barrido, etc.), indicando el tipo de error/acierto encontrado; también se tiene un resumen general de todas las UTR o un resumen detallado por UTR.

1.2.3. Base de Datos

La base de datos para un sistema SCADA es una base de datos en tiempo real en la que es imperativo que sea almacenada y recuperada rápidamente. La base de datos se actualiza cada exploración de las UTR. Con cada valor para puntos digitales y analógicos se asocian varios atributos que definen todo lo referente al punto, estos atributos son predeterminados durante el diseño y la definición del sistema; asimismo, varios de ellos pueden ser cambiados por el operador. Los programas de aplicación se basan en los atributos para realizar operaciones sobre cada punto; estos atributos proporcionan información como: [IEE 91]

- Colores relacionados con el punto
- Estado del punto: activo o desactivo, abierto o cerrado, etc.
- Algo que indique que el valor de un punto tiene unos límites relacionados
- Que el dispositivo es controlable
- Que el punto tiene tendencias o no
- Que el punto es real, pseudopunto o calculado
- etc.

1.2.4. Interfaz Hombre-Máquina

La aplicación de la interfaz hombre-máquina permite al operador ver el estado del sistema, cambiar los valores de los datos, cambiar la configuración del sistema, y controlar el sistema de potencia. Estos programas son de las partes más importantes del sistema de control supervisorio y de las más complicadas. Por ejemplo, la base de datos descrita en la

subsección anterior debe tener la facilidad para habilitar al operador a cambiar el valor o estado de un punto o modificar su atributo de encendido a apagado. Similarmente, la función de control supervisorio consiste en responder a las acciones del operador de abrir y cerrar dispositivos de conmutación y modificar la base de datos para que refleje tales acciones. Algunos de los aspectos requeridos que se consideran necesarios para una interfaz hombre-máquina son: [IEE 91]

- Facilidad para mostrar pantallas por vía de números, nombres, menús o puntos.
- Facilidad para observar la información completa de un punto dado y editar los datos y sus atributos.
- Facilidad para crear o editar pantallas.
- Poder adicionar o borrar identificadores de puntos al sistema.
- Añadir características a un punto.
- Utilizar características de pantallas gráficas tales como acercamientos ("zoom"), etc.
- Control de todos los programas del sistema para inhibirlos, iniciarlos manualmente, o reprogramar sus tiempos de activación.

- Reportes y contabilidad

Cada sistema SCADA tiene la responsabilidad de efectuar periódicamente reportes y de efectuar la contabilidad del sistema. Los reportes son predefinidos y se producen automáticamente a determinados periodos del día. Las cantidades de esta contabilidad y reportes son editables por el operador. Los nuevos reportes pueden ser definidos e implementados en el sistema por ingenieros de soporte.

1.2.5. Diagnósticos

- Sistema de comunicaciones

El sistema maneja información estadística de las comunicaciones hacia las terminales remotas. Por cada UTR, el puerto de comunicación lleva un conteo de tiempos acumulados dentro/fuera de exploración y barrido lento, así como del número de mensajes

enviados y contestados correcta e incorrectamente con el tipo de error. El operador puede solicitar esta información para ser desplegada en pantalla o impresora.

- Diagnóstico remoto a UTR

El operador puede solicitar a las UTR, por medio de un comando de autodiagnóstico, que verifiquen sus módulos y le informen sobre su estado.

- Autodiagnóstico del sistema de cómputo

Indica fallas en bancos de memoria, mecanismo de protección de memoria, reloj calendario, cronómetros y manejo de interrupciones. Los periféricos, como impresoras y consolas de operación, cuentan con diagnósticos propios.

1.3. Estación Maestra MAC-5000

Como ejemplo de una Estación Maestra se describe la MAC-5000 desarrollada en el Departamento de Electrónica del Instituto de Investigaciones Eléctricas para la supervisión y telecontrol de la red eléctrica al nivel de subáreas de control de CFE y/o CLyFC.

La estación maestra está integrada por el sistema de cómputo MAC y el controlador de comunicaciones CIC, el cual maneja la comunicación hacia las unidades terminales remotas (UTR) y a otras EM. El MAC y el CIC se enlazan, unidos por un canal serie tipo RS-485 a 38400 bauds, con un disco duro y uno flexible para el manejo por el sistema de cómputo. Además se tienen dos estaciones de operador, cada una con un monitor, un teclado y una impresora, y una estación para el ingeniero de sistema, residente en una computadora personal que se enlaza con el sistema de cómputo MAC (figura 1.4).

[GUT 90]

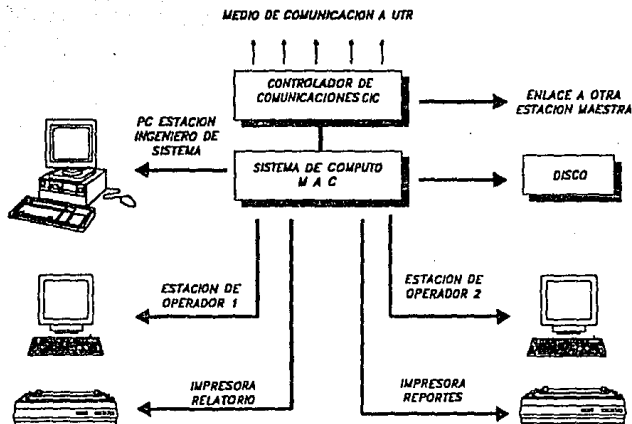


FIGURA 1.4 Arquitectura general de la Estación Maestra MAC-5000

1.3.1. Sistema de cómputo MAC

El sistema de cómputo consta de múltiples procesadores, basados en tarjetas MAC-1186, en esquema multimaestro con acceso a una base de datos común (memoria compartida). La comunicación se realiza a través de un bus o canal común (IBUS-III), y un arbitraje en paralelo y de prioridades fijas controla los accesos al bus. El sistema de cómputo MAC cuenta con cuatro módulos maestros (MAC-1186, basados en el procesador Intel 8086): uno dedicado al procesamiento de la información o módulo central; un módulo para cada estación de operador (IHM-1 e IHM-2) y el otro, para las funciones de manejo de discos (duro y flexible), la generación de informes especiales y el enlace con la estación del ingeniero de sistema basada en computadora personal. Además, el sistema de cómputo cuenta con un módulo esclavo de comunicaciones (MAC-851, basado en el procesador Intel 8031), para enlace con el controlador de comunicaciones CIC.

Cada estación de operador tiene un teclado con funciones especiales y alfanuméricas, un monitor a color de 19 pulgadas y una impresora. La impresora de la IHM-1 está dedicada al relatorio, donde se imprimen todas las alarmas, eventos o comandos del operador con la fecha y hora de ocurrencia; la impresora de la IHM-2, a informes o impresión de cualquier desplegado en pantalla de las dos consolas.

1.3.2. Controlador de Comunicaciones CIC

La arquitectura del controlador de comunicaciones CIC se compone de un módulo maestro MAC-1186 y varios módulos esclavos inteligentes MAC-851, uno encargado de comunicarse al sistema de cómputo MAC y los otros como puertos de comunicación para las UTR o como enlace a una estación maestra de nivel superior.

Las tarjetas controladoras de comunicaciones MAC-851, contienen los programas para controlar los enlaces con varios protocolos en varios medios físicos. La canasta del CIC tiene espacio para las tarjetas MX 280 las cuales contienen básicamente un modem para interconectarlo con radio VHF/UHF o línea telefónica.

1.4. Estación Maestra Distribuida EMIX-V1

Actualmente en el Departamento de Electrónica del Instituto de Investigaciones Eléctricas se está realizando el desarrollo de una plataforma abierta para aplicaciones de control y supervisión de redes eléctricas, la cual permitirá proveer al sector eléctrico nacional con equipos de nueva generación. La reducción del costo de equipos y la proliferación de redes locales de datos han originado una forma - la que se ha dado en llamar 'Sistemas de Control Distribuido' - más eficiente y económica de atacar los problemas de control.

La tendencia de los equipos de control distribuido y supervisorio es la utilización de arquitecturas basadas en los sistemas abiertos. La definición de un sistema abierto es [IEE 88]: Un sistema que permite a las aplicaciones implementadas bajo este principio a:

- Ser ejecutadas en sistemas de cualquier fabricante (portabilidad).
- Inter-operar con otras aplicaciones desarrolladas bajo el mismo principio.

- Interconectar dispositivos de diversos fabricantes (redes de comunicación)
- Presentar un estilo consistente de interacción con el usuario.

Estas capacidades se describen en términos de especificaciones extensivas para interfaces, servicios y formatos de soporte. Para los equipos que competen al control de redes eléctricas, contar con un Sistema Abierto de Control Supervisorio Distribuido requiere adaptaciones principalmente en los siguientes tópicos:

- Sistemas Operativos.
- Redes de Comunicación.

En cuanto a los Sistemas Operativos, los esfuerzos de estandarización se centran en el comité P10003 ó POSIX (Portable Operating System Interface for Computer Environments) para desarrollar un sistema operativo abierto con una interfaz de aplicaciones estandar basado en UNIX [IEE 88].

Las redes de comunicación en un sistema abierto, utilizan el modelo de referencia conocido como OSI/ISO. Como ejemplos de comunicación entre controladores y equipos de cómputo en sistemas abiertos se tiene la red Ethernet basada en el estándar IEEE 802.3 [GLO 89].

En la figura 1.5 se muestra un diagrama a bloques de la configuración del sistema. Los principales componentes son: un equipo de cómputo para procesamiento de información y manejo de una red local (LAN) con dos estaciones de operador gráficas (EO-X) y una unidad de cinta magnetica para respaldo (DDS), una terminal gráfica para el ingeniero del sistema (EI-X), dos impresoras para los informes de eventos y una impresora a color para impresiones en donde se requiera una mayor calidad de presentación, se incluye también un Controlador Inteligente de Comunicaciones realizado en una computadora personal (CIC-V2), el cual se encarga del enlace hacia las unidades terminales remotas y un controlador de equipo periferico (CEP) el cual puede enlazarse a un controlador de tablero mimico (CTM) ó a un modem para la comunicación con una estación de operador remota.

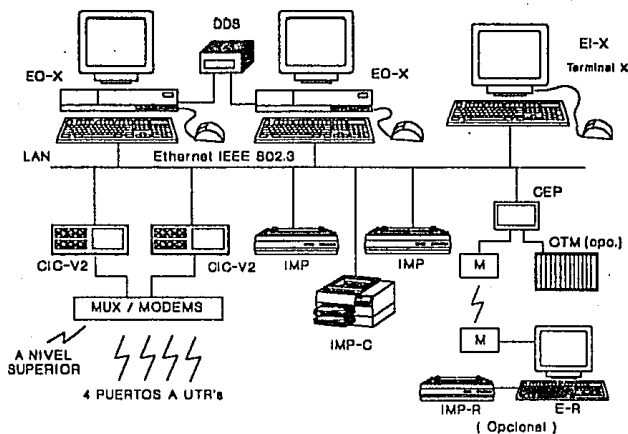


FIGURA 1.5 Configuración de la Estación Maestra EMIX - V1

El presente trabajo comprende el desarrollo del módulo maestro del Controlador Inteligente de Comunicaciones en computadora personal (CIC-V2), el cual se describe en el siguiente capítulo.

Capítulo 2

CONTROLADOR INTELIGENTE DE COMUNICACIONES

En este capítulo se describe el Controlador Inteligente de Comunicaciones en computadora personal denominado CIC-V2, el cual forma parte de la estación maestra distribuida EMIX-V1 en desarrollo en el IIE; se define su arquitectura, las funciones que ejecuta y las bases de datos utilizadas

Como parte de la EM, el Controlador Inteligente de Comunicaciones realiza y procesa todos los intercambios de información entre el sistema de cómputo y las UTR distribuidas en puntos geográficos distantes, así como los intercambios de información a otras EM; asimismo, lleva funciones estadísticas del comportamiento de las comunicaciones con las UTR y el sistema de cómputo. De esta manera, libera al sistema de cómputo de las tareas de adquisición de información y comunicación con la UTR u otra EM, manejo de modems, radios u otros medios de comunicación.

2.1. Descripción General

El Controlador Inteligente de Comunicaciones en computadora personal (CIC-V2) es un equipo que opera como el coordinador en el manejo de mensajes y comandos en las tareas de comunicaciones del sistema supervisorio.

Su esquema funcional está basado en la filosofía Maestro-Esclavo, esto es, el CIC-V2 está formado por un módulo de procesamiento maestro y de varios módulos esclavos (su número dependiendo de los espacios libres en las ranuras de expansión de la computadora en que este implementado).

El procesamiento de las comunicaciones se hace en estos módulos inteligentes esclavos, donde se ejecutan programas de recepción y transmisión de datos de/a las UTR y/o otras EM. La arquitectura con módulos esclavos permite programar en cada uno un puerto con protocolo diferente, dando interconectividad a la estación maestra con varios fabricantes de sistemas remotos.

Existen diferentes protocolos implementados en la versión anterior del CIC en la EM MAC-5000 los cuales pueden utilizarse en la versión actual del CIC-V2 debido a que se mantiene una completa compatibilidad en el formato de los mensajes de comunicación. Los protocolos disponibles actualmente son: [JIM 90]

Fabricante	Nombre del protocolo
I.I.E.	PCS
CDC	CDC-44/500
CDC	CDC-44/400
HARRIS	Harris 5000
L & N	Conitel 2020
DTE	COM 832

2.2. Arquitectura

El desarrollo del CIC-V2 en una computadora personal es una alternativa eficiente para los sistemas de control supervisorio, ya que al integrar todos los elementos del control de comunicaciones en un equipo comercial el costo disminuye, además, su futura integración a una red local se facilita (solo requiere de cambios en los programas de enlace), lo cual le da una gran flexibilidad y facilidad de adaptarse a otros equipos; por otro lado se mantiene una arquitectura maestro-esclavo como en la versión anterior lo que permite aprovechar los protocolos ya implementados así como la filosofía general de operación.

2.2.1. Configuración General

Las figuras 2.1 y 2.2 muestran la arquitectura básica del CIC-V2, formada por una computadora personal, tarjetas controladoras de comunicaciones y acometidas al medio físico. Las tarjetas controladoras de comunicaciones PC-851, a las que también llamaremos puertos, contienen los programas para controlar los enlaces a través de varios protocolos.

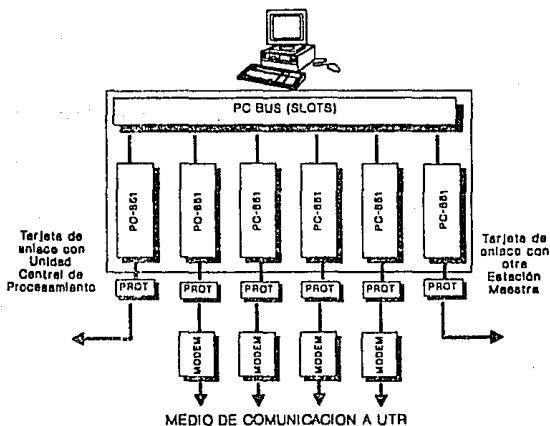


FIGURA 2.1 Arquitectura básica del CIC-V2

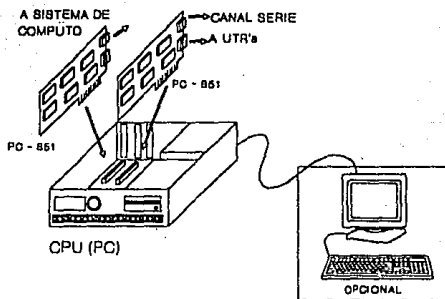


FIGURA 2.2 Configuración del CIC-V2

La computadora en la que está implementado el prototipo cuenta con un procesador 80286 donde se ejecutan los programas del módulo maestro del CIC-V2; se requiere además de un mínimo de RAM de 640K y un disco duro de 20 MBytes; la comunicación con las tarjetas PC-851 se realiza a través del PC-BUS. La intención de este trabajo es tener un sistema que pueda ejecutarse en un computadora industrial con PC-BUS, así como en cualquier computador AT o compatible.

2.2.2. Tarjeta de comunicaciones PC-851

Las tarjetas PC-851 tienen una arquitectura basada en el microcontrolador 80C51 de Intel. Este esquema tiene como periféricos: memorias (64 KBytes de ROM, 32 KBytes de RAM), dos canales serie RS485/RS232 programables desde el microcontrolador por medio de un DUART y con velocidades de comunicación de 50 a 38400 baud; maneja las líneas TX, RX, RTS, CTS, DSR, DTR, CD, con la interfaz RS-232C, la red de comunicación RS-485 cuenta con detección de colisiones. Existe también otro canal serie RS-232C que se maneja directamente del microcontrolador (canal bit-bit), utilizado para el protocolo de control supervisorio y con velocidad programable de hasta 2400 bauds [ORT 91]. La figura 2.3 muestra el diagrama a bloques de la tarjeta.

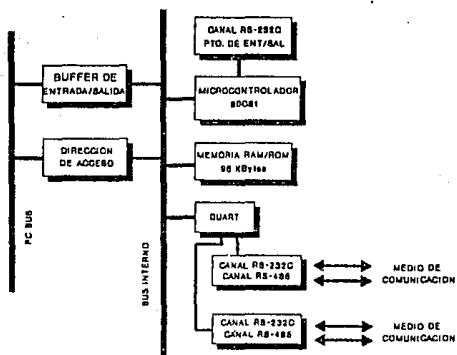


FIGURA 2.3 Arquitectura de la tarjeta de comunicaciones PC-851

De acuerdo a los niveles del modelo OSI (interconexión de sistemas abiertos), de la Organización Estándar Internacional (ISO) y de comité de la IEEE 802 (ver figura 2.4) en la tarjeta controladora de comunicaciones PC-851 se manejan los tres niveles inferiores del modelo [IBA 89]:

- Nivel físico
- Nivel de acceso
- Nivel de control de enlace lógico

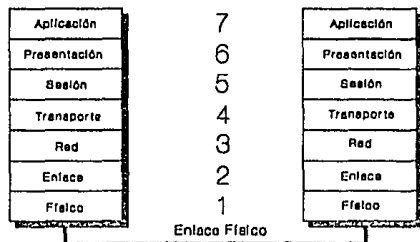


FIGURA 2.4 Modelo de comunicaciones OSI/ISO

2.2.3. Modems

El CIC-V2 generalmente requiere de un modem para poder comunicarse con las UTR u otra EM, el cual debe conectarse a las líneas de campo por el canal bit-bit. El modem debe programarse para poderse conectar a línea telefónica, radio o hilo piloto.

La arquitectura de los modems puede variar de acuerdo a la norma que se utilice (BELL 103, 211A, CCITT V.21, V.22, V.23, V.26, V.27, etc.), y de acuerdo al tipo de modulación empleado: modulación en amplitud (AM), por corrimiento de frecuencia (FSK), por corrimiento de fase (PSK) ó por cuadratura de amplitud modulada (QAM) [PIC 92].

Otro aspecto de esta arquitectura, es la disponibilidad de multicanalización en vías de comunicación, siendo de esta manera importante de canalizar el canal RS-232C manejado por el puerto para control supervisorio. La figura 2.5 muestra un esquema de direccionamiento para las vías de comunicación desde los módulos esclavos.

Las consideraciones anteriores hacen flexible esta arquitectura para conectar modems de manera externa al sistema de comunicación.

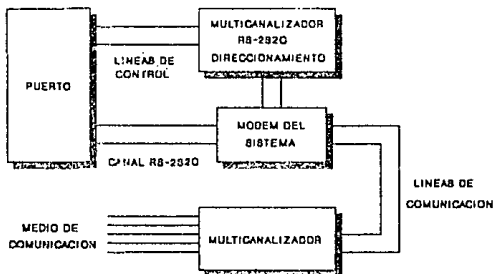


FIGURA 2.5 Esquema de direccionamiento para los módulos esclavos.

2.3. Funciones

Las tareas del módulo maestro juegan un papel importante dentro del sistema de control supervisorio para la transferencia bidireccional de información entre los puertos y el sistema de cómputo.

Las funciones del módulo maestro son manejadas por medio de una base de datos donde se describe la configuración del sistema, cuantos puertos tiene, cuantos modems y el tipo de protocolo que es manejado por cada uno de los puertos, las características de comunicación del sistema remoto y que tipo de UTR se utiliza.

Los puertos desarrollan su función de manera independiente para comunicarse con las UTR. El modo de comunicación es utilizando un protocolo propio para control supervisorio, este protocolo define la estructura y el formato de los mensajes enviados de la EM a las UTR. Algunos protocolos envían un bit de datos a la vez con un tiempo entre cada uno (protocolos orientados a bit); otros, para aprovechar las características de los microprocesadores, transmiten 8 bits (protocolos orientados a byte), tal como se muestra en la figura 2.6. Esos protocolos utilizan una longitud variable del mensaje, especificando la longitud al principio de éste.

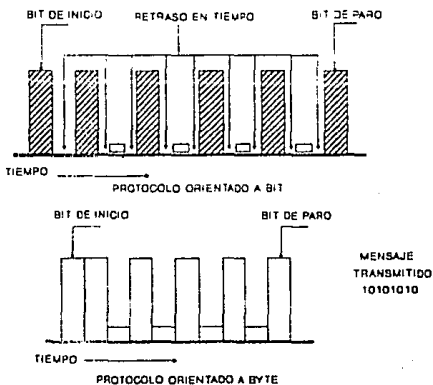


FIGURA 2.6 Protocolos orientados a bit y a byte

Se definieron las siguientes operaciones para el CIC-V2:

1. Realizar la adquisición de información periódica de la UTR (transmisión y recepción).
2. Manejar el medio de comunicación con la UTR.
3. Detectar los cambios digitales etiquetándolos (nivel puerto).
4. Registrar fecha y hora de eventos (analógicos y digitales).
5. Realizar estadísticas de comunicaciones.
6. Enviar telecontroles (enrutar, enviar, codificar) solicitados por el sistema de cómputo.
7. Actuar como UTR (emulación).
8. Tener interfaz hombre-máquina local.
9. Realizar diagnóstico y manejo de errores.

Debido a que se cuenta con inteligencia en cada puerto, se separaron las funciones en: (a) funciones ejecutadas en módulo maestro del CIC y (b) funciones ejecutadas en puertos; ambos tipos de funciones se describen a continuación.

2.3.1. Funciones del módulo maestro

El módulo maestro del CIC-V2 se encarga, al iniciar su operación, de la inicialización y configuración del equipo; posteriormente realiza la coordinación del intercambio de información en el sistema, la actualización de la base de datos y lleva además funciones estadísticas.

Desde el punto de vista operacional, el módulo maestro del CIC-V2 es un sistema que trabaja en un ambiente de multitareas que se ejecutan concurrentemente en un solo procesador. Cada tarea, que en adelante llamaremos PROCESO, tiene asignados sus recursos propios y comparte otros que son administrados por el sistema operativo; la sincronización y comunicación entre los diferentes procesos se realiza mediante semáforos y buzones.

Una vez que se establece la comunicación entre el módulo maestro y el módulo esclavo, el primero coordina la tarea de distribuir mensajes enviados por el sistema de cómputo y asignarlos a un puerto de exploración; por otra parte, se encarga de recibir mensajes (datos analógicos o digitales) con algún cambio de estado a través del puerto, el cual registra y procesa la información en el formato del protocolo de adquisición de datos.

Estos datos se depositan en una base de datos, la cual será enviada al sistema de cómputo por el puerto de comunicación correspondiente. Otra característica del módulo maestro del CIC-V2 es llevar un registro estadístico de mensajes enviados y recibidos por los puertos. Las estadísticas se dividen en dos grupos: a) estadísticas de exploraciones y b) estadísticas de comunicaciones, de las que se enuncian sus funciones principales:

a) Estadísticas de exploraciones

- Número de exploraciones intentadas
- Número de exploraciones exitosas
- Número de exploraciones fallidas
- Número de incumplimientos en ciclos de exploración

b) Estadísticas de Comunicaciones

- Número de intentos de comunicación con la UTR
- Número de intentos exitosos de comunicación con la UTR
- Número de intentos fallidos en la comunicación con la UTR
- Número de intentos de comunicación con CTS de modem activo en la Rx
- Número de intentos de comunicación en bit de sentido
- Número de intentos de comunicación con error en bit de dirección de la UTR
- etc.

2.3.2. Funciones de los módulos esclavos

Cada puerto se personaliza con un protocolo de comunicación independiente del sistema de cómputo para manejar la comunicación con las UTR, de manera que el sistema puede tener varios protocolos diferentes trabajando en forma simultánea, si es que el sistema de control supervisorio cuenta con UTR de diferentes fabricantes y protocolos. Cada puerto lleva una imagen del estado de los puntos digitales en el campo y efectúa las detecciones de cambio de estado; además efectúa los barridos cíclicos a las agrupaciones de puntos en las UTR, de manera autónoma. Para el caso de entradas digitales, únicamente reporta si el punto sufrió un cambio de estado o si se detectó un cambio momentáneo (recierre), para el caso de entradas analógicas reporta el valor de la medición para la verificación de límites por el procesador central. [VAR 92]

En el puerto se localizan funciones de aplicación que no necesitan comunicarse con las UTR para ser realizadas y otras funciones que si lo requieren (funciones internas y externas). Las funciones locales o internas realizan accesos a la base de datos, ya sean de lectura o escritura, para mantener el correcto funcionamiento del puerto. A continuación se enlistan las funciones internas:

1. Cargado de la base de datos de una UTR.
2. Cargado de las imágenes digitales de una UTR.
3. Cargado de las líneas, modems y canales del puerto.
4. Cargado de la tabla de adquisiciones periódicas del puerto.
5. Cargado del mapa de los grupos de una UTR.
6. Poner en barrido lento o barrido normal a una UTR.
7. Cambiar el barrido a un grupo.
8. Meter o sacar a una UTR en exploración.
9. Meter o sacar un grupo de exploración.
10. Poner al puerto en modo paro o modo normal.
11. Dar de alta o baja a un modem.
12. Dar de alta o baja a un canal.
13. Cambiar los tiempos de operación de un modem.
14. Cambiar los tiempos de operación de un canal.

Las funciones externas solicitan a las UTR que realicen alguna operación en particular, ya sea de control o adquisición, y regresen el resultado de la operación. A continuación se enlistan las funciones correspondientes:

1. Exploración de estados.
2. Exploración de grupo
3. Exploración de acumuladores.
4. Exploración de analógicas.
5. Control de puntos de salida digital.
6. Diagnóstico de modem y canal.

2.4. Base de datos

El modelo conceptual de cualquier base de datos (BDD) consiste en definir la información que llevará dentro de sí misma, en forma independiente de la manera física de como se almacenan los datos [DAT 81].

Para el caso específico de una red eléctrica que está siendo supervisada y controlada por una estación maestra, se define y se describe cada entidad o elemento físico que necesita ser considerado dentro de la base de datos (subestación, estación remota, puertos, canales de comunicación, etc.) . La base de datos con la que opera el módulo maestro del CIC-V2 obedece al modelo relacional, el cual concibe la base de datos como un conjunto de relaciones que se representan en forma de tablas [JIM 90]. Estas relaciones son creadas en el proceso de inicialización de la estación maestra y enviadas al CIC-V2 desde el sistema de cómputo; el CIC-V2, a su vez, construye una base de datos basada en la anterior y se la envía a los puertos para darles a conocer el equipo que está siendo utilizado.

2.4.1. Módulo maestro

La BDD del maestro del CIC-V2 está formada por un total de cinco estructuras de datos, a las que llamaremos RELACIONES, que contienen la información que es manipulada durante toda la operación del maestro en el CIC-V2. Esta BDD es accesada y modificada por los diferentes procesos que forman al módulo maestro, es decir, la base de datos puede considerarse como varias estructuras de datos globales que son compartidas por todos los procesos.

Todas las operaciones realizadas a la BDD por cada uno de los procesos se protegen mediante llamadas a funciones que bloquean a los demás procesos mientras que el proceso poseedor del CPU termina su tarea de lectura o escritura, garantizando de esta manera la integridad de los datos.

Las cinco relaciones que forman la base de datos en el maestro del CIC son: [JIM 90]

Relación	Nombre corto
Unidades terminales remotas	(UTR)
Puertos de comunicación	(PUERTO)
Grupos de UTR	(ADQ_GPOS)
Canales de digitalizadores	(CAN_DPG)
Estadísticas	(ESTADIS)

2.4.2. Módulos esclavos

La composición y situación del sistema se informa a los módulos esclavos (puertos) mediante una base de datos consistente en una serie de arreglos que describen las características de las UTR y los equipos de comunicaciones del mismo. Esta información se transmite desde el sistema de cómputo hacia el módulo maestro del CIC-V2 quien a su vez lo reenvía al puerto correspondiente al inicio de sus operaciones. La base de datos debe representar correctamente el estado de la UTR, modems, canales, etc. u ocurriran errores en el intercambio de información.

La descripción de los puntos a explorar se encuentra en la base de datos, y estos se agrupan por funciones dando lugar a los "grupos de exploración". Además existe una tabla que permite realizar la transformación de los números de punto como los conoce el maestro del CIC-V2 (dentro de cada grupo) con los números de punto correspondientes en la UTR.

Los tipos de punto son:

- Puntos de entrada digital ó de estado.
- Puntos de entrada analógica.
- Puntos contadores de eventos digitales.
- Puntos de control digital.

Asimismo, la base de datos específica para cada UTR indica cuales son las líneas, modems y canales para comunicación que tiene disponibles, así como las velocidades de comunicación de información apropiadas.

Finalmente, la base de datos especifica la frecuencia (período de tiempo entre exploraciones) y los tipos de exploraciones que deben realizarse mediante un conjunto de funciones de ejecución periódica.

Capítulo 3

ESPECIFICACION Y DISEÑO DEL MODULO MAESTRO DEL CIC

Este capítulo comprende la especificación y el diseño del módulo maestro del CIC-V2; se explica el modelo fundamental del sistema, detallando los elementos que forman su estructura (procesos, flujo de datos, buzones y la base de datos) junto con los diagramas de estructura realizados para cada uno de los procesos.

En esta especificación se encuentran además los requerimientos para la implementación del CIC-V2, el cual se codificó con el compilador Top Speed C [TOP 91] con manejo de concurrencia.

3.1. Descripción del modelo

El diseño se describe como una serie de pasos en el cual la representación del flujo de datos, los diagramas de estructuras y los procedimientos se sintetizan de la definición de requisitos del CIC. Este diseño se expresa de manera que se puedan implementar estos componentes en un lenguaje de programación. La especificación es parte de este proceso, en el cual el diseño se expresa en una forma abstracta de alto nivel.

Las actividades de especificación y diseño están conectadas de una forma tan estrecha que es imposible separarlas. De ahí que se considere a la especificación como parte del proceso de diseño, siendo una especificación una representación abstracta de un diseño. [SOM 88]

El proceso de desarrollo de programas involucra la elección de los métodos adecuados en combinación con metodologías bien definidas. La idea de una metodología que guíe al programador en su trabajo durante las fases de diseño y desarrollo es contar con una manera sistemática de solucionar problemas y tener una técnica estándar y uniforme durante todo el ciclo de desarrollo. Las herramientas que automatizan las actividades que involucra el seguir alguna metodología de las existentes en ingeniería de software tienen el término genérico de CASE "Computer Aided Software Engineering". [GHE 91]

Para la especificación y el diseño del módulo maestro del CIC-V2 se utilizó una herramienta gráfica de CASE llamada **SYSTEM ARCHITECT** [SYS 90] para la elaboración de los diagramas, ajustándose a la metodología "*Data Flow Ward & Mellor*" para diseño de sistemas en tiempo real y tomando en cuenta también el *Modelo Orientado al Flujo de Datos* que define las diferentes etapas que transforman al flujo de información en estructura de programas. En el Apéndice B se describe la metodología utilizada.

El medio ambiente de esta herramienta es completamente gráfico y permite construir el sistema a cualquier nivel de abstracción, pudiéndose definir a cada uno de sus elementos (símbolos, flujo de datos, etc) en una base de datos interna; cuenta también con un sistema de reportes en el que se le indica mediante un programa, con base en SQL, el tipo de datos deseados en el reporte, lo cual es útil para la documentación del sistema desarrollado.

3.2. Modelo fundamental

El módulo maestro del CIC-V2 está estructurado en dos niveles de abstracción, el primero, llamado el modelo fundamental, está basado en la metodología de Ward & Mellor y muestra los procesos juntos con los eventos y mensajes asociados. El segundo nivel de abstracción desglosa cada proceso en un diagrama de estructura, describiendo mediante pseudocódigo cada uno de los módulos que forman al diagrama.

Se presenta a continuación la documentación obtenida para el *Modelo Fundamental del Sistema*. En el diagrama en la figura 3.1 que muestra el primer nivel de abstracción, se aprecian nueve procesos independientes manejados por el módulo maestro, así como sus interacciones a través de buzones, los mensajes transmitidos entre ellos y la forma en que accesan la Base de Datos. La herramienta SYSTEM ARCHITECT tiene la flexibilidad de permitir la definición de cada elemento del diagrama según el tipo al que pertenecen.

3.2.1. Operación del sistema

El sistema consiste de nueve procesos que se ejecutan concurrentemente en el procesador de la computadora personal, además de un proceso secuencial, que se ejecuta al iniciar por primera vez el sistema, encargado de la inicialización del equipo así como de la asignación de recursos (memoria, buzones, semáforos, etc.).

La comunicación entre procesos se realiza mediante buzones; un buzón es un área de memoria en donde se depositan los mensajes de varios procesos para ser leídos por algún otro. La comunicación con las tarjetas esclavas se efectúa a través de memorias FIFO y por medio de interrupciones para notificar cuando existan datos presentes. La estructura de los mensajes se mantiene igual que la utilizada en la comunicación entre procesos.

Todos los procesos tienen la misma prioridad de activación, por lo que su funcionamiento sigue una política de asignación en rueda "Round Robin" debido a que cada determinado tiempo (1/8 seg aprox.) se realiza un cambio de contexto (guardar el estado de un proceso para que al ser reactivado continúe la tarea asignada), pasando el

proceso poseedor del CPU al final de la lista de espera de procesos para activación. Existen excepciones en el modo de activar un proceso, como en el caso de los procesos de enlace hacia las tarjetas PC-851, estos procesos se activan cuando ocurre una interrupción de las tarjetas (módulos esclavos) al módulo maestro, notificándolo mediante un semáforo de manera que la interrupción sea atendida lo más pronto posible.

3.2.2. Definición de Procesos

En el diagrama de la figura 3.1 los procesos se representan por medio de un círculo en el cual se especifica el número de proceso y el nombre del mismo, tres círculos más pequeños en la parte superior indican que el proceso se expande a otro diagrama; asimismo, cada proceso tiene internamente una narrativa que describe su función. La narrativa de cada proceso se encuentra en el reporte llamado "Definición de Procesos" elaborado con el sistema de reportes del SYSTEM ARCHITECT y del cual se presenta su contenido.

- *Proceso de asignación de recursos (RAIZ)*

En este módulo se establece la inicialización de la operación del sistema en el Maestro del CIC, se obtienen los recursos necesarios para los procesos: semáforos, buzones, memoria, se instalan los vectores de interrupción para el manejo de los Puertos de Comunicación, se obtiene la imagen de la tabla de estadísticas y el estado inicial de los Puertos, se inicializa el reloj de tiempo real, se activan los procesos del Maestro del CIC y se avisa a la Computadora Central que el CIC está listo para operar.

- *Proceso concentrador (CONCENT)*

Este proceso recibe los mensajes provenientes de los puertos del CIC, ejecuta procesamientos dependiendo del número de función recibida, valida resultados, lleva control de tiempos acumulados, actualiza la Base de Datos y envía el mensaje al proceso de estadísticas.

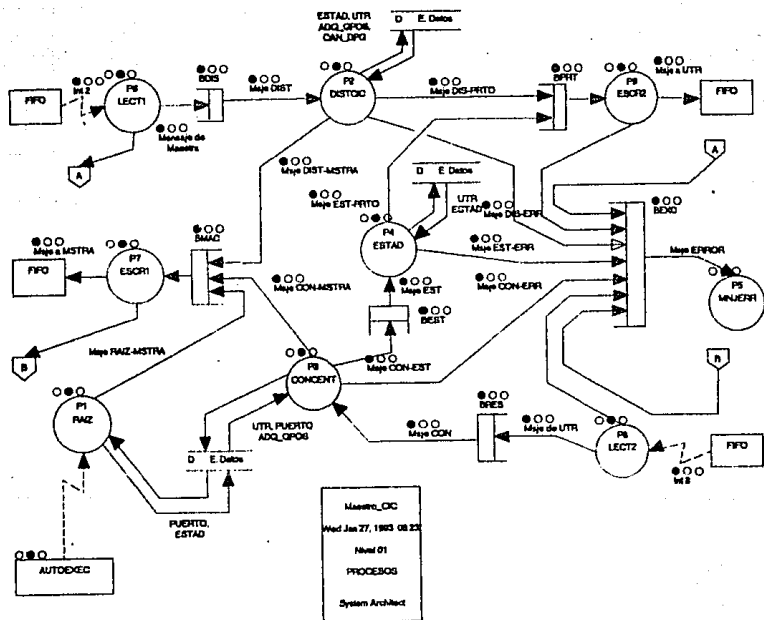


FIGURA 3.1 Modelo fundamental del módulo maestro del CIC-V2

- Proceso distribuidor (DISTCIC)

Este proceso recibe los mensajes provenientes de la Computadora Central por medio del Puerto de enlace; ejecuta los procesamientos dependiendo del número de función; realocaliza la Base de Datos; envía la imagen de la Base de Datos anterior; inicializa el reloj de tiempo real, la cuenta de tiempos acumulados y los puertos del CIC.

- Proceso de escritura al sistema de cómputo (ESCR1)

Este proceso está dedicado a la recepción de mensajes que provienen del DISTRIBUIDOR, del CONCENTRADOR o del módulo de ESTADÍSTICAS, los arregla y los escribe directamente al Puerto de enlace con la Computadora Central, para su envío a través de la tarjeta PC-851.

- Proceso de escritura a UTR (ESCR2)

Recibe los mensajes enviados por el DISTRIBUIDOR del CIC y del proceso de ESTADÍSTICAS, los arregla, selecciona el puerto de enlace con las UTR adecuado y los escribe para su posterior procesamiento.

- Proceso de estadísticas de comunicaciones (ESTAD)

Recibe mensajes con los resultados de la ejecución de las funciones por los Puertos y realiza estadísticas actualizando la Base de Datos en los campos de resultados exitosos, resultados con falla y tiempos de exploración.

- Proceso de lectura al sistema de cómputo (LECT1)

Este es el proceso encargado de la recepción de los mensajes directamente del Puerto de Comunicación que sirve de enlace con la Computadora Central, arregla los mensajes y los envía al DISTRIBUIDOR del CIC para que este ejecute el procesamiento adecuado.

- Proceso de lectura a las UTR (LECT2)

Proceso encargado de la recepción de los mensajes directamente de los Puertos de Comunicación que se enlazan con las diferentes UTR, arregla los mensajes y los envía al módulo CONCENTRADOR del CIC para su posterior procesamiento.

- Proceso manejador de errores (MNJERR)

El proceso manejador de errores recibe los errores detectados por el DISTRIBUIDOR, por el CONCENTRADOR o por el módulo de ESTADÍSTICAS, clasifica el error, realiza las operaciones pertinentes y avisa a la Computadora Central del Sistema. Lleva además un diagnóstico de los procesos del maestro del CIC, en el que se puede supervisar el estado de los procesos (activo, en espera, bloqueado, etc.).

3.2.3. Definición de Buzones

Los buzones permiten el manejo de la comunicación/sincronización de procesos a través de mensajes; dado que el compilador utilizado en el desarrollo no posee esta función, se codificaron rutinas utilizando semáforos.

Los buzones se implementaron tomando como referencia el modelo utilizado en el sistema operativo SOPCO-51 [DAZ 90]. Los mensajes se insertan en un extremo del buzón y se leen en el otro, según la norma "primero en entrar-primero en salir", la figura 3.2(a) ejemplifica esta acción. Un proceso que lee un buzón puede obtener el mensaje más antiguo que contiene o en su defecto una indicación de que el buzón está vacío, (ver figura 3.2(b)). Los buzones se encuentran en un área de memoria global, cada uno de ellos asociado a colas de procesos bloqueados ordenados por prioridad; estos procesos bloqueados esperan recibir mensaje si el buzón está vacío, o esperan enviar un mensaje si el buzón está lleno (figura 3.2(c)), estas dos condiciones no se pueden presentar al mismo tiempo ya que la comunicación es en un solo sentido, por lo que no ocasionan ningún conflicto.

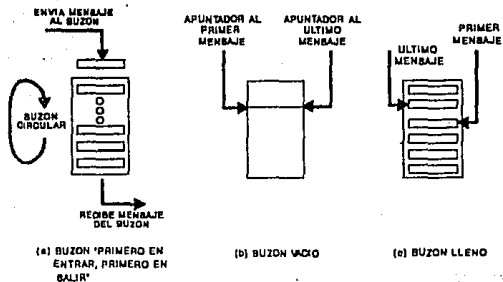


FIGURA 3.2 Manejo de buzones

Los descriptores de los buzones tienen la siguiente estructura:

Nombre	Tipo	Descripción
sem_envia	SIGNAL (dependiente del compilador)	Semaforo asociado con el buzón para la cola de procesos bloqueados en envia.
sem_rec	SIGNAL	Semaforo asociado con el buzón para la cola de procesos bloqueados esperando recibir.
nom[6]	char	Nombre del buzón, 6 caracteres
creado	int	indica si el buzón esta creado o no
dir	char *	dirección de inicio del buzón
longit	int	longitud del buzón, max 64k
p_mje	int	dirección inicial del primer mensaje
u_mje	int	dirección final último mensaje

Existen tantas estructuras como buzones en el sistema. Se puede realizar el intercambio de mensajes por copia o por referencia, tal como se explica a continuación:

- **Por copia:** el mensaje y su longitud implícita se duplican en el buzón y posteriormente en el área del proceso que lo recibe.
- **Por referencia:** se copia sólo la dirección del mensaje en el buzón para posteriormente copiarla en una variable del proceso que recibe.

El buzón es circular y se maneja por medio de los apuntadores del descriptor, definiéndose si está lleno por un espacio que se deja entre el inicio del primer mensaje y el final del último, es decir, primero se almacena o saca un dato del buzón y después se incrementa el apuntador.

Para el uso de los buzones se tienen las siguientes funciones:

- Creación de buzón:** Inicializa el directorio del buzón, se marca el tipo de comunicación que se realizará, se inicializa las variables *dir*, *p_mje* y *u_mje* con la dirección donde principia el buzón. Verifica que el buzón no haya sido creado anteriormente.
- Envía:** Obtiene la longitud del mensaje y llama al procedimiento que lo copia al buzón; si hay procesos bloqueados envía una señal para alistar al primer proceso que se bloqueó. Si no hay espacio en el buzón para guardar el mensaje y la selección es bloqueo, el proceso es insertado en la cola del semáforo *sem_envia* que tiene a los procesos que están esperando que halla espacio en el buzón para enviar su mensaje.
- Recibe:** Copia el primer mensaje almacenado en el buzón al área de recepción del proceso. Si no hay mensajes en el buzón y la opción es bloqueo, el proceso es ubicado en la cola del semáforo *sem_rec* en la que se encuentran los procesos bloqueados esperando recibir. Después de recibir un mensaje verifica si existen procesos bloqueados esperando enviar, de ser así, envía una señal para alistar al primer proceso que se bloqueó.

Los buzones se representan en el diagrama como un rectángulo con la línea de uno de sus lados dentro de la figura, el nombre se presenta con tres pequeños círculos en su parte superior con el círculo de la izquierda remarcado, que indica que tiene un comentario a manera de descripción ya que este símbolo no tiene una definición formal en la biblioteca del SYSTEM ARCHITECT. La definición de buzones se encuentra en el reporte "Definición de Buzones" y se muestra a continuación su contenido.

- Buzón del proceso distribuidor (BDIS)

Buzón mediante el cual el proceso manejador del Puerto de Enlace con la Computadora Central transfiere los mensajes recibidos al DISTRIBUIDOR para su procesamiento.

- Buzón del proceso de estadísticas (BEST)

Mediante este buzón el proceso CONCENTRADOR transfiere los resultados de las operaciones al módulo de ESTADÍSTICAS para su actualización en la Base de Datos.

- Buzón del manejador de errores (BEXC)

A través de este buzón se envían los mensajes con avisos de error para su manipulación en el proceso manejador de errores y excepciones. Se envían también los mensajes para realizar el diagnóstico de los procesos.

- Buzón del enlace al sistema de cómputo (BMAC)

El proceso manejador de la escritura de mensajes al Puerto de Enlace con la Computadora Central los recibe mediante este buzón. Los mensajes pueden ser enviados desde el DISTRIBUIDOR, el CONCENTRADOR, el proceso RAIZ o el módulo de ESTADÍSTICAS.

- Buzón del enlace a las UTR (BPRT)

El proceso DISTRIBUIDOR y el de ESTADÍSTICAS envían los mensajes a través de este buzón al proceso manejador de la escritura a los Puertos que se enlazan con las UTR.

- Buzón del concentrador (BRES)

Mediante este buzón el proceso manejador de lectura de los puertos enlazados con las UTR envía los mensajes al proceso CONCENTRADOR del módulo maestro del CIC.

3.2.4. Definición de Mensajes

En el diagrama 3.1 del modelo fundamental, las líneas continuas indican flujo de datos, interpretados en nuestro diseño como mensajes de un proceso a otro; al igual que los buzones, estos símbolos están definidos con un comentario en donde se explica ampliamente la gama de valores y la estructura manejada en ellos. Para todos los mensajes se maneja un mismo tipo de encabezado con la siguiente estructura:

ENCABEZADO:

long	Entero	(2 bytes)
func	Entero	(2 bytes)
resu	Entero	(2 bytes)
n_msje	caracter	(1 byte)
destino	caracter	(1 byte)

La siguiente tabla (3.1) muestra las funciones definidas en el CIC para el módulo maestro; así como los procedimientos en los que se realiza alguna acción con ellas siendo que una función se puede utilizar en más de un procedimiento. Las abreviaturas de los procedimientos que se manejan se describen en la tabla 3.2.

TABLA 3.1 Funciones del módulo maestro del CIC

Número de función	Proceso Distribuidor	Proceso Concentrador	Proceso Estadísticas	Significado del mensaje
1	----	RPTEXP	ESTEXP	REPORTE DE EXPLORACION
3	UTRS	ECOEST	ESTCTR	CONTROL DIGITAL
7	DPGS	ECOEST	ESTCOM	MEDICION DEL DPG
8	ACUM	----	----	ACUMULADORES
10	BDCIC	----	----	CARGADO BDD CIC
11	BDDPRT	ECOMAC	----	CARGADO DE UTR
12	BDDPRT	ECOMAC	----	CARGADO ADQ. PERIODICAS
13	BDDPRT	ECOMAC	----	CARGADO LINEA, MODEM, CANAL
14	BDDPRT	ECOMAC	----	CARGADO DE IMAGENES DIGITALES
15	BDDPRT	ECOMAC	----	CARG. MAP. GPOS EN UTR CDÇ o DTE
16	BDDPRT	ECOMAC	----	CARG. MAP. GPOS EN UTR HARR
20	---	BARUTR	----	CAMBIO DE BARRIDO A LAS UTR
23	PRTOS	ECOMAC	----	CAMBIO DE TIEMPO DE CANAL
24	UTRS	UTREXP	----	METER UTR A EXPLORACION
25	UTRS	UTREXP	----	SACAR UTR DE EXPLORACION

26	UTRS	GPOEXP	----	METER GRUPO A EXPLORACION
27	UTRS	GPOEXP	----	SACAR GRUPO DE EXPLORACION
28	UTRS	ECOMAC	----	CAMBIO DE BARRIDO A GRUPO
29	UTRS	ECOEST	ESTCOM	INICIALIZA DPG
30	PRTOS	ECOMAC	----	CAMBIO DE TIEMPO A MODEM
31	UTRS	ECOEST	ESTCOM	PROGRAMA DPG
33	UTRS	ECOEST	ESTCOM	DIAGNOSTICO UTR
34	PRTOS	PARNOR	----	PUERTO MODO NORMAL
35	PRTOS	PARNOR	----	PUERTO MODO PARO
36	PRTOS	ECOMAC	----	ALTA MODEM
37	PRTOS	ECOMAC	----	BAJA MODEM
38	PRTOS	ECOMAC	----	ALTA CANAL
39	PRTOS	ECOMAC	----	BAJA CANAL
40	RELOJ	----	----	INICIALIZA RELOJ DE TIEMPO REAL
41	PRTOS	ECOMAC	----	DIAGNOSTICO DE MODEM Y CANAL
42	UTRS	ECOEST	ESTCOM	AJUSTE DE PUNTO ANALOGICO
48	UTRS	ECOEST	ESTCOM	COMANDO DE SUBIR BAJAR
50	ESTAD1	----	----	SOLICITUD ESTADISTICAS
51	ESTAD2	----	----	RESET ESTADISTICAS
52	ESTAD3	----	----	CARG DE IMAGEN Y RESET ESTAD
53	ESTAD4	----	----	ENVIA TABLA DE IMAGEN
54	DIAGCI	----	----	DIAGNOSTICO DEL CIC
62	BDDPRT	ECOMAC	----	FUNCIONES PROTOCOLO HARRIS
63	BDDPRT	ECOMAC	----	FUNCIONES PROTOCOLO HARRIS
64	BDDPRT	ECOMAC	----	FUNCIONES PROTOCOLO HARRIS
65	BDDPRT	ECOMAC	----	FUNCIONES PROTOCOLO HARRIS
66	BDDPRT	ECOMAC	----	FUNCIONES PROTOCOLO HARRIS
67	----	ECOES2	ESCOM3	TRANSMISION ESTACION MAESTRA
68	----	ECOES2	ESCOM2	RECEPCION ESTACION MAESTRA
69	BDDPRT	ECOMAC	----	PROTOCOLO L&N
70	BDDPRT	ECOMAC	----	PROTOCOLO L&N
71	PRTOS	ECOMAC	----	PROTOCOLO L&N
0xFFE0	X REFE	X REF	----	MENSAJE POR REFERENCIA

Existe además una función que se genera en el proceso raíz y que sirve para indicar al sistema de cómputo que el CIC-V2 esta listo para entrar en operación.

Número de función	Proceso en donde se genera	Significado del mensaje
43	RAIZ	CIC LISTO

TABLA 3.2 Procedimientos por proceso

PROCEDIMIENTOS PARA EL PROCESO DISTRIBUIDOR	
UTRS	Funciones de UTR
PRTOS	Funciones de Puertos
BDCIC	Cargado de base de datos del CIC
BDDPRT	Cargado BDD del puerto e inicialización
X REFE	Mensaje por referencia
DPGS	Adquisición canales DPGS
ADCUM	Adquisición de acumuladores
ESTAD1	Solicitud de estadísticas
ESTAD2	Reset de estadísticas
ESTAD3	Cargado de imagen y reset a estadísticas
ESTAD4	Envía imagen de estadísticas
RELOJ	Inicializa el reloj de tiempo real
DIAGCI	Diagnostico del CIC
PROCEDIMIENTOS PARA EL PROCESO CONCENTRADOR	
ECOMAC	Envía mensaje tal cual a sistema de cómputo
PARNOR	Modo Paro/Modo Normal
GPOEXP	Meter/Sacar Grupo a exploración
UTREXP	Meter/Sacar UTR a exploración
BARUTR	Cambio de barrido a la UTR
ECOEST	ECOMAC más estadísticas
RPTEXP	Reporte de exploración
X REF	Mensaje por referencia
ECOES2	ECOMAC más estadísticas
PROCEDIMIENTOS PARA EL PROCESO ESTADÍSTICAS	
ESTEXP	Estadísticas de exploración
ESTCOM	Estadísticas de comunicaciones
ESTCTR	Estadísticas de controles
ESCOM2	Estadísticas de comunicaciones 2
ESCOM3	Estadísticas de comunicaciones 3

A continuación se enlistan los mensajes manejados por el módulo maestro y se muestra el contenido del reporte "Definición de Mensajes" el cual describe la función que ejecuta.

- Señal de interrupción (Int 2)

Evento que le indica al proceso manejador del Puerto de Enlace con la Computadora Central que existen datos presentes en la memoria FIFO de la tarjeta PC-851 listos para ser leídos.

- Señal de interrupción (Int 3)

Evento que le indica al proceso encargado de las comunicaciones provenientes de las UTR que hay datos disponibles en el puerto para su lectura.

- Mensaje del sistema de cómputo

El proceso LECT1 se encarga de recibir los mensajes del sistema de cómputo y les da un formato para enviarlos al DISTRIBUIDOR para un procesamiento posterior. La estructura de los mensajes que envía es la siguiente:

encabezado	
tipo	(1 byte)
ptr mem	(apuntador)
datos	-----

El mensaje puede enviarse por copia o por referencia, según se especifique en el parámetro función.

- Mensaje del puerto hacia el proceso concentrador (Msje CON)

El concentrador procesa los mensajes provenientes de los puertos de manera distinta, de acuerdo a la función que se está requiriendo, para enrutarlos al procedimiento adecuado. Se tienen agrupadas las funciones en varios procedimientos los cuales tienen estructuras bien definidas; cada uno de estos procedimientos tiene una tarea específica, que son las que se muestran a continuación.

(1) PROCEDIMIENTO : ECO A MAC (ECOMAC)

Este procedimiento se encarga de hacer llegar el mensaje proveniente de los puertos al sistema de cómputo. Generalmente los mensajes consisten únicamente del encabezado y son la respuesta proveniente del puerto a alguna acción solicitada.

estructura del mensaje:

encabezado	
puerto	(1 byte)
datos	-----

(2) PROCEDIMIENTO : PUERTO PARO/NORMAL (*PARNOR*)

Este procedimiento se encarga de enviar el comando para que la tarjeta entre en operación normal (exploración) o para deshabilitar la operación (modo paro). En el mensaje se debe especificar el número de la tarjeta (puerto) sobre la que se efectuará la acción.

estructura :

encabezado	
puerto	(1 byte)

(3) PROCEDIMIENTO : GPO EXPLORACION (*GPOEXP*)

Este procedimiento envía los comandos para meter un grupo a exploración, así como para quitarlo de exploración. En el mensaje se define cual es la UTR en donde se encuentra el grupo junto con el grupo sobre el que se desea efectuar la operación.

estructura :

encabezado	
utr	(1 byte)
n gpo	(1 byte)

(4) PROCEDIMIENTO : UTR EXPLORACION (*UTREXP*)

Este procedimiento envía el comando necesario para que una UTR sea explorada, o sea sacada de exploración. El número de UTR en que se efectuará la acción se define en el mensaje.

estructura :

encabezado	
utr	(1 byte)

(5) PROCEDIMIENTO : CAMBIO DE BARRIDO A UTR (BARUTR)

Este procedimiento se encarga del cambio de barrido a las UTR, necesitando especificar la UTR y el tipo de barrido deseado. Este mensaje se genera automáticamente cuando la UTR no responde a los comandos de exploración. Después de cierto número de intentos se decide colocarla en barrido lento.

estructura :

encabezado	
utr	(1 byte)
t barr	(1 byte)

(6) PROCEDIMIENTO : ECO A CENTRAL Y ESTADÍSTICAS (ECOEST)

Este procedimiento se encarga de redirigir los mensajes hacia el proceso que los envía al sistema de cómputo y hacia el proceso de estadísticas en donde se realizan las estadísticas de comunicación.

estructura :

encabezado	
datos	-----

(7) PROCEDIMIENTO : REPORTE DE EXPLORACION (RPTEXP)

El procedimiento maneja los mensajes en donde se incluyen los datos de los grupos explorados para la UTR a la que se le solicitó que enviara un reporte del estado del grupo.

estructura :

encabezado	
utr	(1 byte)
grupo	(1 byte)
datos	-----

(8) PROCEDIMIENTO : ECO A CENTRAL Y ESTADÍSTICAS 2 (ECOES2)

Este procedimiento es similar al descrito en el punto (6) de esta sección, solo que se realizó para ser utilizado con otras funciones (67 y 68).

estructura :

encabezado	
datos	-----

- Mensajes del proceso concentrador al manejador de errores (Msje CON-ERR)

Cuando el proceso CONCENTRADOR detecta algún error en la recepción del mensaje a través del buzón BRES coloca un código de error en el elemento RESU del mensaje y envía el encabezado al manejador de errores para que éste se encargue de realizar las acciones pertinentes.

- Mensajes del proceso concentrador al de estadísticas (Msje CON-EST)

El proceso CONCENTRADOR recibe mensajes provenientes del puerto de enlace con las UTR y decide cuando se envían a la Computadora Central y cuando debe de realizarse alguna estadística con estos mensajes. Se muestra a continuación una lista de los procedimientos y la estructura del mensaje que requieren ser enviadas al proceso ESTADÍSTICAS.

(1) PROCEDIMIENTO : ECO A CENTRAL Y ESTADÍSTICAS (ECOEST)

Este procedimiento envía los mensajes al sistema de cómputo así como al proceso de estadísticas para que se contabilicen los resultados obtenidos.

estructura del mensaje :

encabezado	
datos	-----

(2) PROCEDIMIENTO : REPORTE DE EXPLORACION (RPTEXP)

Este procedimiento envía el mensaje al proceso de estadísticas para que se realicen las actualizaciones a la base de datos para los datos indicados en el mensaje.

estructura :

encabezado	
utr	(1 byte)
grupo	(1 byte)
datos	-----

(3) PROCEDIMIENTO : ECO A CENTRAL Y ESTADISTICAS 2 (ECOES2)

Dentro de los mensajes enviados del proceso concentrador al proceso de estadísticas se encuentran aquellos que pasan por este procedimiento, el cual es básicamente el mismo que el descrito anteriormente en (1) pero se utiliza para las funciones de transmisión y recepción

estructura :

encabezado	
datos	-----

- Mensaje del proceso concentrador al sistema de cómputo (Msje CON-MSTRA)

El proceso CONCENTRADOR envía a la Computadora Central todos los mensajes que recibe después de que ha realizado las actualizaciones en la Base de Datos (cuando sea necesario) y después de mandar el mensaje al proceso de ESTADISTICAS (solo para algunos procedimientos).

Se muestra a continuación una relación de los procedimientos para los cuales los mensajes se envían a la Computadora Central por medio del buzón *BMAC*, la estructura de los mensajes se puede observar en la definición de los mensajes que recibe el concentrador en *Msje CON*.

La descripción de los procedimientos es la misma que la elaborada para los mensajes en *Msje CON*.

(1) ECO A MAC	(<i>ECOMAC</i>)
(2) PUERTO PARO/NORMAL	(<i>PARNOR</i>)
(3) GPO EXPLORACION	(<i>GPOEXP</i>)
(4) UTR EXPLORACION	(<i>UTREXP</i>)
(5) CAMBIO DE BARRIDO A UTR	(<i>BARUTR</i>)
(6) ECO A CENTRAL Y ESTADISTICAS	(<i>ECOEST</i>)
(7) REPORTE DE EXPLORACION	(<i>RPTEXP</i>)
(8) ECO A CENTRAL Y ESTADISTICAS 2	(<i>ECOES2</i>)

- Mensaje del proceso distribuidor al manejador de error (*Msje DIS-ERR*)

El proceso DISTRIBUIDOR recibe mensajes provenientes de la Computadora Central y los procesa, cuando alguno de estos mensajes presenta un error en la recepción o cuando se ejecuta la función, se manda el encabezado del mensaje con un código de error al proceso manejador de errores *MNJERR* por medio del buzón *BEXC*.

- Mensaje del proceso distribuidor a los puertos (*Msje DIS-PRTO*)

La siguiente relación muestra los procedimientos para las cuales el mensaje se envía del proceso distribuidor al Puerto (Buzón *BPRT*).

(1) PROCEDIMIENTO : FUNCIONES PARA UTR (*UTRS*)

En este procedimiento se envían a la tarjeta PC-851 los mensajes en los que se indique una función relacionada con alguna acción tomada para la UTR especificada en el mismo mensaje.

(2) PROCEDIMIENTO : FUNCIONES PARA PUERTOS (*PRTO*)

A través de este procedimiento se le envían al puerto los comandos internos de operación de la tarjeta PC-851, indicando cada comando por el número de función contenido en el mensaje.

(3) PROCEDIMIENTO : CARGADO BDD PUERTO E INICIALIZACION
(BDCIC)

Este procedimiento se encarga de las funciones de inicialización de cada tarjeta de comunicaciones, siguiendo un orden específico para enviar los datos al puerto de la configuración del equipo; los datos enviados al puerto deben de corresponder con el equipo instalado, pues la tarjeta no hace una validación y esto podría ocasionar el mal funcionamiento del sistema

(4) PROCEDIMIENTO : FUNCIONES PARA DPGS (DPGS)

(5) PROCEDIMIENTO : FUNCIONES PARA ACUMULADORES (ADCUM)

- Mensajes hacia el proceso distribuidor (Msje DIST)

De acuerdo a la función requerida por el sistema de cómputo, el mensaje se interpreta de diferentes maneras. A continuación se muestra una lista de las estructuras de los mensajes que corresponden a ciertos procedimientos junto con las funciones que se pide ejecutar desde la Computadora Central. Las descripciones son básicamente las mismas que la descritas en *Msje DIS-PRTO*.

(1) PROCEDIMIENTO : FUNCIONES PARA UTR (UTRS)

estructura :

encabezado	
utr	(1 byte)
datos	-----

(2) PROCEDIMIENTO : FUNCIONES PARA PUERTOS (PRTOS)

estructura :

encabezado	
puerto	(1 byte)
datos	-----

(3) PROCEDIMIENTO : BASE DE DATOS DEL CIC (*BDCIC*)

estructura :

encabezado	
datos	-----

(4) PROCEDIMIENTO : CARGADO BDD PUERTO E INICIALIZACION
(*BDDPRT*)

estructura :

encabezado	
puerto	(1 byte)
datos	-----

(5) PROCEDIMIENTO : FUNCIONES PARA DPGS (*DPGS*)

estructura :

encabezado	
utr	(1 byte)
dpg	(1 byte)
n can dpg	(1 byte)

(6) PROCEDIMIENTO : FUNCIONES PARA ACUMULADORES (*ADCUM*)

estructura :

encabezado	
tipo	(1 byte)

(7) PROCEDIMIENTO : SOLICITUD DE ESTADÍSTICAS (ESTAD1)

estructura :

encabezado	
utr	(1 byte)

(8) PROCEDIMIENTO : RESET A ESTADÍSTICAS (ESTAD2)

estructura :

encabezado	
utr	(1 byte)

(9) PROCEDIMIENTO : CARGADO DE IMAGEN Y RESET A ESTADÍSTICAS (ESTAD3)

estructura :

encabezado	
imagen de estadísticas	"n" bytes de la BDD

(10) PROCEDIMIENTO : ENVIO DE IMAGEN (ESTAD4)

estructura :

encabezado	
imagen de estadísticas	

(11) PROCEDIMIENTO : INICIALIZA RELOJ DE TIEMPO REAL (RELOJ)

estructura :

encabezado	
udia	(1 byte)
ddia	(1 byte)
umes	(1 byte)
dmes	(1 byte)
uanio	(1 byte)
danio	(1 byte)

dsem	(1 byte)
useg	(1 byte)
dseg	(1 byte)
umin	(1 byte)
dmin	(1 byte)
uhor	(1 byte)
dhor	(1 byte)

(12) PROCEDIMIENTO : DIAGNOSTICO DEL CIC (*DIAGCI*)

En este procedimiento se recibe el encabezado del mensaje proveniente del sistema de cómputo y lo regresa como reconocimiento, de esta manera el sistema de cómputo diagnostica que el CIC-V2 esté funcionando.

- Mensaje del proceso distribuidor al sistema de cómputo (Msje DIST-MSTRA)

El proceso DISTRIBUIDOR recibe los mensajes del puerto de enlace con la Computadora Central y se encarga de ejecutar los procedimientos requeridos, para ciertas funciones el proceso debe informarle a la Computadora Central si la operación fue exitosa o si hubo algún error, lo cual lo realiza enviando el mensaje al buzón *BMAC*. A continuación se describen brevemente los procedimientos que envían mensajes a este buzón.

(1) PROCEDIMIENTO : BASE DE DATOS DEL CIC (*BDDCIC*)

Actualiza la tabla de puertos y copia la BDD del mensaje a la localidad correspondiente. Si no hubo error informa a la Computadora Central de operación exitosa, si hubo error informa también al manejador de errores.

(2) PROCEDIMIENTO : CARGADO BDD PUERTO E INICIALIZACION (*BDDPRT*)

Obtiene status del puerto, así como su protocolo y le envía los datos necesarios en la base de datos para su inicialización. Si hubo error lo informa a la Computadora Central y al manejador de errores.

(3) PROCEDIMIENTO : SOLICITUD DE ESTADISTICAS (ESTAD1)

Obtiene de tres relaciones los datos necesarios para llamar a la función *acum_fpos*, la cual se encarga de actualizar el tiempo que ha permanecido la UTR en determinado barrido. Crea un nuevo mensaje y lo pasa por referencia al buzón de la Computadora Central.

estructura del mensaje enviado a *BMAC* :

encabezado	
utr	(1 byte)
datos	(1 byte)

(4) PROCEDIMIENTO : RESET A ESTADISTICAS (ESTAD2)

Limpia los valores de *ESTAD*, si no hubo error manda el mensaje original a la Computadora Central.

(5) PROCEDIMIENTO : CARGADO DE IMAGEN Y RESET A ESTADISTICAS (ESTAD3)

Construye un nuevo mensaje con la imagen de estadísticas colocada al final del mensaje y lo envía por referencia a la Computadora Central; limpia los valores de *ESTAD*.

(6) PROCEDIMIENTO : ENVIO DE IMAGEN (ESTAD4)

Crea un nuevo mensaje con la imagen al final de éste y lo envía por referencia al buzón de la Computadora Central.

(7) PROCEDIMIENTO : INICIALIZA RELOJ DE TIEMPO REAL (RELOJ)

Informa a la Computadora Central de operación exitosa o si hubo algún error colocándole al elemento *RESU* un código de error y enviando únicamente el encabezado del mensaje.

(8) PROCEDIMIENTO : DIAGNOSTICO DEL CIC (DIAGCI)

Recibe el mensaje y lo regresa a la Estación Maestra a manera de reconocimiento.

- Mensajes hacia el proceso estadísticas (Msje EST)

Se envían hacia el proceso ESTADÍSTICAS las funciones que requieran una actualización de la Base de Datos o aquellas que deban incrementar un acumulador para realizar estadísticas. Se muestran a continuación las estructuras de los mensajes considerados por este proceso.

(1) PROCEDIMIENTO : ESTADÍSTICAS DE EXPLORACION (ESTEXP)

En este procedimiento se llevan registros de mensajes enviados y recibidos por los puertos tales como:

- Número de exploraciones intentadas
- Número de exploraciones exitosas
- Número de exploraciones fallidas
- Número de incumplimientos en ciclos de exploración

estructura del mensaje:

encabezado	
utr	(1 byte)
datos	(1 byte)

(2) PROCEDIMIENTO : ESTADÍSTICAS DE COMUNICACIONES

(ESTCOM)

En este procedimiento se lleva un registro del comportamiento de las comunicaciones con las UTR (intentos exitosos, fallidos, etc). Después de cierto número de intentos de comunicación con la UTR fallidos, la coloca en barrido lento.

estructura del mensaje:

encabezado	
utr	(1 byte)
t barr	(1 byte)

(3) PROCEDIMIENTO : ESTADISTICAS DE CONTROLES (ESTCTR)

En este procedimiento se registran cantidades como: número de controles intentados, número de controles exitosos, número de controles fallidos, etc. Estos valores se colocan en la relación ESTADISTICAS.

estructura del mensaje:

encabezado	
utr	(1 byte)
datos	-----

(4) PROCEDIMIENTO : ESTADISTICAS DE COMUNICACIONES 2 (ESCOM2)

estructura :

encabezado	
utr	(1 byte)
puerto	(1 byte)

(5) PROCEDIMIENTO : ESTADISTICAS DE COMUNICACIONES 3 (ESCOM3)

estructura :

encabezado	
utr	(1 byte)
puerto	(1 byte)

- Mensaje del proceso estadísticas al manejador de error (Msje EST-ERR)

Cuando el proceso ESTADISTICAS detecta algún error en la recepción del mensaje coloca un código de error en el elemento RESU del mensaje y manda únicamente el encabezado al proceso encargado de manejar los errores (BEXC).

- Mensaje del proceso estadísticas al puerto (Msje EST-PRTO)

Existe un procedimiento para el cual el proceso ESTADÍSTICAS necesita leer los datos de la relación UTR y de ESTAD para sacar estadísticas, actualizar las relaciones y formar un nuevo mensaje para enviarlo al buzón BPRT. La estructura y las funciones para las cuales el procedimiento forma un nuevo mensaje se definen a continuación.

**(1) PROCEDIMIENTO : ESTADÍSTICAS DE COMUNICACIONES
(ESTCOM)**

estructura :

encabezado	
utr	(1 byte)
t barr	(1 byte)

- Mensajes hacia el sistema de cómputo (Msje a MSTR)

El proceso ESCR1 envía para su procesamiento un mensaje al sistema de cómputo con la siguiente estructura.

encabezado	
tipo	(1 byte)
ptr mem	(apuntador)
datos	-----

- Mensaje hacia las Unidades Terminales Remotas (Msje a UTR)

El proceso encargado de enviar los mensajes a las distintas UTR tiene un formato de mensajes definido de la siguiente manera:

encabezado	
ptr mem	(apuntador)
datos	-----

- Mensajes provenientes de las Unidades Terminales Remotas (Msje de UTR)

El proceso LECT2 se encarga de recibir los mensajes del Puerto de Enlace con las UTR y les da un formato para enviarlos al proceso CONCENTRADOR para su procesamiento, la estructura utilizada en este caso es:

encabezado	
tipo	(1 byte)
ptr mem	(apuntador)
datos	-----

3.2.5. Definición de Estructuras de Datos

Las estructuras de datos forman lo que llamamos la Base de Datos del sistema y están definidas formalmente en un almacén de datos como estructuras de datos que contienen elementos. cada elemento a su vez se describe por medio de una narrativa, el tipo de dato y su longitud; esta Base de Datos se define externamente y el reporte obtenido muestra la agrupación de los elementos en estructuras y los posibles valores para cada elemento. El reporte se denominó "Definición de Estructuras de Datos" y contiene la siguiente información:

- Estructura ADQ_GPOS

En esta sección se especifican cada uno de los campos de datos que forman la estructura de ADQ_GPOS que contiene los grupos de UTR, existe un registro idéntico para cada UTR que se encuentre enlazada con el CIC-V2. Se presenta a continuación una tabla con los campos que componen a esta estructura y posteriormente se da una descripción de cada campo.

Nombre	Tipo	Descripción
FREC_H	caracter	Especifica la frecuencia de Adquisición, campo de horas (0..24)
FREC_M	caracter	Especifica la frecuencia de Adquisición, campo de minutos (0..60)
FREC_S	caracter	Especifica la frecuencia de Adquisición, campo de segundos (0..60)

GPO_ACT	caracter	Especifica el estado de actualización del GRUPO SI = 0FFH NO = 00H
N_GPO	caracter	Especifica el número de GRUPO dentro de la UTR GRUPOS MIXTOS --> 0..14 GRUPOS DCM --> 15
OP_ADQ	caracter	Campo para uso general
STS_ADQ	caracter	Especifica el estado de la adquisición DENTRO = 0 FUERA = 1
TIPO_ADQ	caracter	Especifica el tipo de adquisición PERIODICA = 0 POR SOLICITUD = 1
TIPO_GRU	caracter	Este campo especifica el tipo de GRUPO 1 COMBINADO 2 SOLO ACUMULADORES 3 SOLO ESTADO 4 SOLO ANALÓGICAS 5 DIGITAL CON DETECCIÓN DE CAMBIO
UTR_ADQ	caracter	Número identificador de cada UTR

- Estructura CAN_DPG

En esta sección se especifican cada uno de los campos de datos que forman la estructura que contiene a los canales de digitalizadores, existe en registro idéntico para cada digitalizador que se encuentre enlazado con el CIC-V2.

DEC	caracter	Especifica el número de bytes a considerar
FC	caracter (2)	Especifica el factor de conversión o escala
NIV_INF	caracter	Especifica el nivel de disparo inferior (n veces el valor nominal), cuatro bits para dígito entero y cuatro bits para dígito decimal 0.0 - 3.1 0.0 = DISPARO PERMANENTE 3.1 = NO DISPARO
NIV_SUP	caracter	Especifica el nivel de disparo superior (n veces el valor nominal), cuatro bits para dígito entero y cuatro bits para dígito decimal. 0.0 - 9.9 0.0 = DISPARO PERMANENTE 9.9 = NO DISPARO
NOM	caracter (10)	Especifica el nombre del canal o de la variable
N_CAN_DPG	caracter	Número que identifica al canal del DIGITALIZADOR

N DPG	caracter	Número de identificación para cada DIGITALIZADOR
OFF	caracter	Especifica el 'offset'
OP_CAN	caracter	Campo para uso general
STS_CAN	caracter	Especifica el estado del canal ALAMBRADO = 0 NO ALAMBRADO = 1
UI	caracter	Especifica las unidades de ingeniería asociadas

- Estructura ESTADIST

En esta sección se especifican cada uno de los campos de datos que forman la estructura de las estadísticas; existe un registro idéntico para cada UTR enlazada.

Nombre	Tipo	Descripción
EST	caracter (50)	Este campo está formado por una secuencia de cincuenta bytes, asociados de dos en dos para cada concepto de los que se llevan ESTADÍSTICAS. En la tabla 3.3 se describe el significado de cada valor en este campo.
T_FUERA	entero largo	Este campo contiene el tiempo acumulado en que la UTR ha estado en FUERA DE EXPLORACION.
T_LENTO	entero largo	Este campo contiene el tiempo acumulado en que la UTR ha estado en BARRIDO LENTO.
T_NORMAL	entero largo	Este campo contiene el tiempo acumulado en que la UTR ha estado en BARRIDO NORMAL.
UTR_EST	caracter	Número identificador de cada UTR.

TABLA 3.3 Descripción de los valores en la tabla de estadísticas

ESTADÍSTICAS DE EXPLORACIONES	
Byte	Descripción
1 y 2	Número de exploraciones intentadas
3 y 4	Número de exploraciones exitosas
5 y 6	Número de exploraciones fallidas
7 y 8	Número de incumplimientos en ciclos de exploración
ESTADÍSTICAS DE COMUNICACIONES	
Byte	Descripción
9 y 10	Número de intentos de comunicación con la UTR
11 y 12	Número de intentos exitosos en la comunicación con la UTR
13 y 14	Número de intentos fallidos en la comunicación con la UTR
15 y 16	Número de intentos de comunicación con CTS de modem activo en la Rx
17 y 18	Número de intentos de comunicación con error en bit de sentido
19 y 20	Número de intentos de comunicación con error en bit de dirección UTR
21 y 22	Número de intentos de comunicación con error en función corta

23 y 24	Número de intentos de comunicación con error en función extendida
25 y 26	Número de intentos de comunicación con ausencia de portadora de radio frecuencia
27 y 28	Número de intentos de comunicación con ausencia de portadora de modem
29 y 30	Número de intentos de comunicación con ausencia de bit de inicio
31 y 32	Número de intentos de comunicación con error en CRC-16
33 y 34	Número de intentos de comunicación con error en bit de paro
35 y 36	Número de intentos de comunicación con CTS del modem inactivo en la Tx
37 y 38	Número de intentos de comunicación con DCD del modem activo en la Tx
39 y 40	Número de controles intentados
41 y 42	Número de controles exitosos
43 y 44	Número de controles fallidos
45 y 46	Número de controles con NO OPERACION, solicitud previa
47 y 48	Número de controles con error en comunicación
49 y 50	Número de controles sin modem y/o canal de salida

- Estructura PUERTO

Se especifica cada uno de los campos que forman la estructura de los puertos de comunicación; existe un registro idéntico para cada puerto contenido en el CIC-V2

Nombre	Tipo	Descripción
N_PRTO	caracter	Se refiere al número identificador de cada PUERTO en el CIC. El número de PUERTO corresponde al número configurado en el dip-switch de la tarjeta PC-851, los valores permitidos están entre 1 y 16.
OP_PRTO	caracter	Campo de propósito general

PROT	caracter	Protocolo de comunicaciones utilizado por el PUERTO en su enlace con las UTR's MODO MAESTRA - PCS = 0 - CDC-44500 = 1 - CDC-44400 = 2 - HARRIS = 3 - L & N = 4 - DTE = 5 MODO REMOTA - PCS = 8 - CDC-44500 = 9 - CDC-44400 = 10 - HARRIS = 11 - L & N = 12 RED DE COMPUTO - MAC = 16
STS_PRTO	caracter	Especifica el estado operativo del PUERTO BIT 0 --> 0/1 EXPLORACION/FUERA DE EXPLORACION BIT 1 --> 0/1 INICIALIZADO/SIN INICIALIZAR BIT 2 --> 0/1 PRESENTE/NO PRESENTE

- Estructura UTR

Esta estructura contiene los campos necesario para describir las características de cada una de las UTR que están enlazadas con el CIC-V2.

Nombre	Tipo	Descripción
DGN	caracter	Resultado del último diagnóstico (formato PCS)
DIR	caracter	Dirección de la UTR en línea (1..15), en función del protocolo
D_CAMBIO	caracter	Especificación de cambio de día en la cuenta del tiempo del último cambio de estado de la UTR
FALLAS	caracter	Número de intentos fallidos consecutivos al establecer comunicación con la UTR
H_CAMBIO	entero largo	Fecha y hora del último cambio de estado de la UTR (exploración/fuera de exploración)
NOM_UTR	caracter (6)	Campo utilizado para especificar el nombre de la UTR, en caracteres ASCII

N_INT	caracter	Número de intentos requeridos antes de cambiar de tipo de barrido
N_LINEA	caracter	Número de línea en la que se encuentra la UTR (0..3)
N_SE	caracter	Número de identificación para la subestación en que está ubicada la UTR
N_UTR	caracter	Número identificador de cada UTR
OP_UTR	caracter	Campo para uso general
PROT_UTR	caracter	Protocolo de comunicaciones utilizado por la UTR en su enlace con el CIC MODO MAESTRA - PCS = 0 - CDC-44500 = 1 - CDC-44400 = 2 - HARRIS = 3 - L & N = 4 - DTE = 5 MODO REMOTA - PCS = 8 - CDC-44500 = 9 - CDC-44400 = 10 - HARRIS = 11 - L & N = 12 RED DE COMPUTO - MAC = 16
PRTO_UTR	caracter	Identificador del número del puerto en el CIC con el que la UTR establece el enlace de comunicación (1..8)
STS_UTR	caracter	Estado de operación de la UTR DENTRO = 0 FUERA = 1
T_BARR	caracter	Tipo de barrido en las exploraciones de la UTR NORMAL = 0 LENTO = 1

VEL	caracter	Velocidad de transmisión/recepción en la comunicación de la UTR con el CIC
		0 → 75 bps
		1 → 150 bps
		2 → 300 bps
		3 → 600 bps
		4 → 1200 bps
		5 → 2400 bps
		6 → 4800 bps
		7 → 9600 bps

3.3. Diagramas de estructuras de los procesos

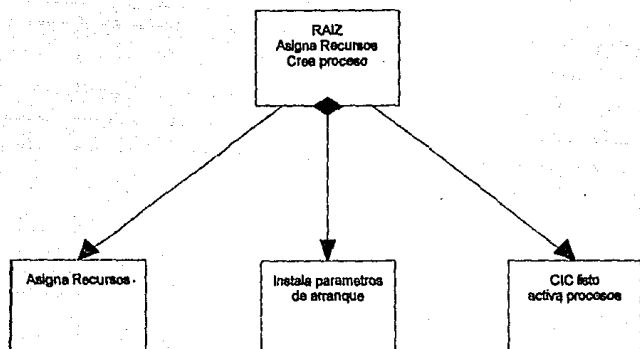
Se presentan a continuación los diagramas de estructuras correspondientes a cada uno de los procesos del módulo maestro del CIC-V2; cada diagrama tiene un recuadro en el que se indica el nivel en que se encuentra y el nombre del proceso que se expandió. El rectángulo con un diamante en la parte inferior se conoce como centro de transacción e indica la función a llevarse a cabo mediante toda la estructura jerárquica. Se utilizó la técnica TOP-DOWN en el desarrollo del diagrama, las flechas indican llamadas a otros módulos y la estructura total debe interpretarse de izquierda a derecha, existen varios diagramas que además indican un flujo de datos interno al programa que se simboliza como una pequeña flecha saliendo de un círculo cuya dirección indica el sentido del flujo.

El pseudocódigo se deriva de una refinación del diagrama de estructura del cual se muestra para cada uno de ellos un reporte realizado en SYSTEM ARCHITECT llamado "Definición de los módulos en el diagrama"¹.

3.3.1. Proceso Raíz

En la figura 3.3 se muestra el diagrama de estructura correspondiente al proceso raíz. Cada uno de los módulos que lo integran cuenta con un pseudocódigo para facilitar su implementación en algún lenguaje de programación. Se muestra su contenido ordenado de arriba hacia abajo y de izquierda a derecha.

¹ Nota: Se numeraron los procedimientos empezando con una "P" para evitar confusión con la secuencia del documento



1. RAIZ
 Wed Jan 27, 1993 06:34
 Nivel 02
 Algoritmo de Inicialización
 y arranque de Operación
 del Maestro del CIC
 System Architect

FIGURA 3.3 Proceso Raiz

P1.0 Rafz

- Ejecuta al principio del programa

1. Crea los buzones a utilizar
2. Crea los menús de ihm
3. Obtiene la base de datos a memoria
4. Obtiene área para imagen de estadísticas
5. Obtiene el estado de los puertos
6. Reset a tarjetas y a interrupciones pendientes
7. Inicializa señales de interrupción y terminación
8. Instala los manejadores de interrupción
9. crea y activa los procesos
10. envía mensaje de CIC listo a central

P1.1 Asigna recursos

Este módulo se encarga de la asignación de memoria a las diferentes estructuras de datos utilizadas durante la ejecución del maestro CIC, inicializa las variables tipo semáforo, coloca la presentación junto con los menús de la IHM local, hace requerimientos de memoria para los buzones y modifica la tabla del vector de interrupciones para instalar los manejadores de interrupción necesarios para controlar el enlace a las tarjetas PC-851.

P1.2 Instala parámetros de arranque

1. En este módulo se inicializa la base de datos del maestro del CIC, esperando su envío a través de la tarjeta de enlace al sistema de cómputo.
2. Una vez recibida la BDD se realiza su conversión de formato MAC a el nuevo formato de estructuras utilizado en el CIC-V2.
3. Obtiene memoria para el área de imagen de la estructura estadísticas
4. Lee el estado general de todos los puertos y los coloca en la tabla *tab_sfs*.

P1.3 CIC listo

1. Crea los nueve procesos del sistema asignandoles la misma prioridad.
2. Construye el encabezado del mensaje en el que se indica que el CIC ha terminado las tareas de inicialización y se encuentra listo para el intercambio de información.
3. Activa la señal de terminación del programa, con lo que todos los procesos empiezan a ejecutarse concurrentemente hasta que un comando de la IHM indique el término del programa.

3.3.2. Proceso Distribuidor

El diagrama de estructura correspondiente al proceso DISTRIBUIDOR se muestra en la figura 3.4. Cada uno de los módulos que integran al diagrama posee un pseudocódigo del que se muestra su contenido a continuación. Los módulos se describen de arriba hacia abajo y de izquierda a derecha identificándolos por el número asignado por el SYSTEM ARCHITECT.

P2.0 Distribuidor

- Repite en un ciclo infinito
- 1. Recibe un mensaje del buzón BDIS
- 2. Si no hay error
despliega el mensaje recibido
procesa el mensaje
- 3. Si hay error
envía el encabezado del mensaje al manejador de errores por el buzón BEXC

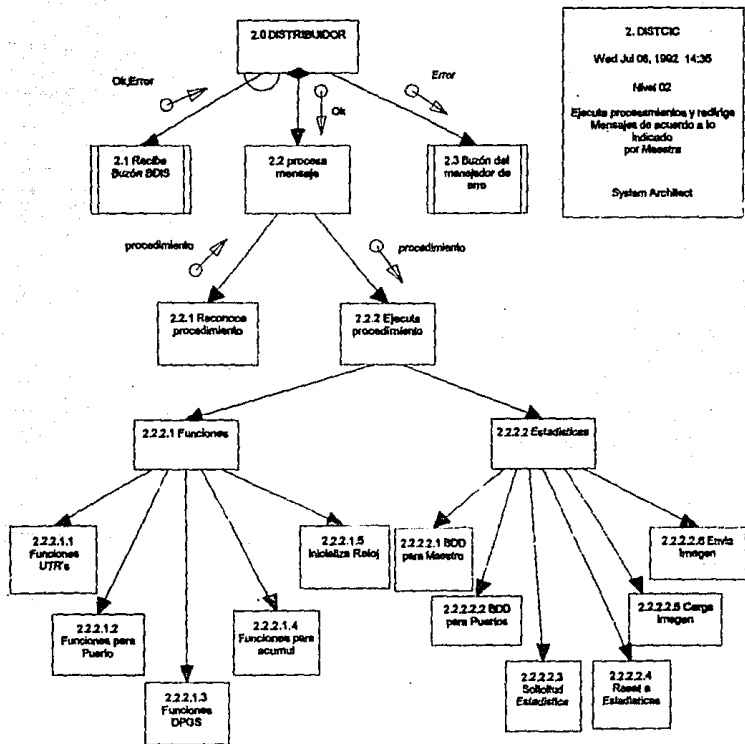
Cuando se ejecuta un procesamiento se le pasa un apuntador a este mensaje e internamente lo interpreta de acuerdo a la función requerida.

P2.1 Recibe Buzón BDIS

Llamada a la rutina general para recepción de mensajes por un buzón, se pide leer el buzón BDIS, con tamaño máximo de mensaje estándar, y con bloqueo.

P2.2 Procesa mensaje

- 1. Busca en la tabla de funciones la que corresponda a la indicada en el mensaje
- 2. Dirige a procesamiento relativo
- 3. Si no encuentra procesamiento
llama a excepción por FUERA_DE_RUTA
- 4. Si encuentra procesamiento
Despliega un mensaje indicando el tipo de procesamiento que se realiza,
ejecuta el procesamiento pasándole un apuntador al mensaje.



2. DISTCIC
 Wed Jul 06, 1992 14:35
 Nivel 02
 Ejecuta procesamientos y redirige Mensajes de acuerdo a lo indicado por Maestra
 System Architect

FIGURA 3.4 Proceso Distribuidor

P2.3 Buzón del manejador de error

Llama a la rutina general de envío por buzones. Después de colocar en el elemento RESU un código de error, ajusta el elemento DESTINO y coloca en LONG la longitud del encabezado, envía el mensaje por BEXC con bloqueo.

P2.2.1 Reconoce procedimiento

Hace una búsqueda lineal en la tabla de procesamiento para encontrar la función indicada en el mensaje y posteriormente direcciona a procesamiento relativo; la búsqueda se facilita conociendo la estructura de la tabla "TABLA DE PROCESAMIENTOS", arreglo que cuenta con tantos elementos como funciones existan (ver tabla 1 en sección 3.2.4).

P2.2.2 Ejecuta procedimiento

De acuerdo al procesamiento solicitado hace la llamada adecuada para su ejecución después de haber desplegado un mensaje indicando el tipo de procesamiento de que se trata; se pasa como parámetro un apuntador al mensaje.

P2.2.2.1.1 Funciones UTR

En este módulo se reciben mensajes que tienen como destino alguna UTR. Para asegurar el correcto funcionamiento del sistema se verifica que la UTR esté declarada y el equipo de comunicación esté presente.

1. Recibe un apuntador al mensaje conteniendo la Base de Datos.
2. con el valor de utr del mensaje obtiene el PRTO_UTR de la relación UTR
3. con el valor obtenido encuentra el estado del puerto de la relación PUERTO
4. verifica si la tarjeta esta presente
5. obtiene el protocolo del puerto
6. Inicialización solo para puertos PCS ($prot \geq PCS$ y $prot \leq MSTR$)
7. envía el mensaje al buzón BPRT con bloqueo
8. Si hubo PROTOCOLO_INVALIDO hace una llamada a una excepción
9. Si hubo error de PUERTO_AUSENTE llama a ERROR_FUN pasándole como parámetro el apuntador al mensaje

P2.2.2.1.2 Funciones para Puerto

En este procedimiento, al igual que en el anterior, se realiza una verificación de equipo para los mensajes que tienen como destino alguna tarjeta PC-851 (puerto).

1. Recibe un apuntador al mensaje conteniendo la Base de Datos.
2. Con el valor de puerto del mensaje obtiene el estado del puerto de la relación PUERTO
3. Verifica si la tarjeta está presente
4. Obtiene el protocolo del puerto
5. Realiza inicialización para puertos PCS ($prot \geq PCS$ y $prot \leq MSTR$)
envía el mensaje al buzón BPRT con bloqueo
6. Si hubo PROTOCOLO_INVALIDO hace una llamada a una excepción
7. Si hubo error de PUERTO_AUSENTE llama a ERROR_FUN pasándole como parámetro el apuntador al mensaje.

P2.2.2.1.3 Funciones DPGS

En este procedimiento se realiza la actualización de la BDD en los parámetros relativos a los canales de digitalizadores.

1. Recibe un apuntador al mensaje conteniendo la Base de Datos.
2. Con el valor de utr del mensaje obtiene el PRTO_UTR de la relación UTR
3. Obtiene el estado del puerto con el valor determinado anteriormente
4. Verifica si la tarjeta está presente
5. Obtiene el protocolo del puerto
6. Realiza inicialización solo para puertos PCS ($pro \geq PCS$ y $pro \leq MSTR$)
7. Obtiene los tuplos (N_CAN_DPG, STS_CAN) que cumplan que N_DPG sea igual al elemento *dpg* del mensaje
Para todos los tuplos
Si STS_CAN es ALAMBRADO
envía al buzón BPRT el mensaje después de actualizar *n_can_dpg*,
long y destino
8. Llama a excepción si hubo PROTOCOLO_INVALIDO
9. Llama a ERROR_FUNC si hubo DPGS_NOALAM o PUERTO_AUSENTE

P2.2.2.1.4 Funciones para acumuladores

Este procedimiento se encarga del manejo de los mensajes relacionados con los puntos llamados acumuladores que fluyen entre el sistema de cómputo y las UTR.

1. Recibe un apuntador al mensaje conteniendo la Base de Datos.
2. Obtiene los tuplos (UTR_ADQ, N_GPO, STS_ADQ, TIPO_GRU, TIPO_ADQ) que cumplan que TIPO_GRU sea del mismo valor que el tipo contenido en el mensaje
 Para todos los tuplos
 De acuerdo al tipo obtiene de la relación UTR los valores de PRTO_UTR, STS_UTR, N_UTR
 Con el valor de PRTO_UTR obtiene el estado del puerto de la relación PUERTO
 Verifica si la tarjeta esta presente
 Obtiene el protocolo del puerto
 Realiza una inicialización solo para puertos PCS (*pro* >= PCS y *pro* <= MSTR)
3. Envía el mensaje al buzón BPRT.
 Si hubo PROTOCOLO_INVALIDO llama al manejador de excepciones
3. Si hubo error de PUERTO_AUSENTE_O_FUERA o UTR_FUERA llama a ERR_FUNC

P2.2.2.2.1 BDD para Maestro

Este módulo se encarga de transformar la BDD con formato MAC al formato de estructuras utilizado en el CIC y de colocar el resultado en el área de memoria asignada.

1. Recibe un apuntador al mensaje, la vía, y el tipo.
2. Copia la Base de Datos a su localidad
3. Si no hubo error
 Obtiene el estado de los puertos y los coloca en la tabla *tab_sts*
 Verifica que los puertos estén presentes
 Envía el encabezado al buzón BMAC
4. Si hubo error
 Coloca un código de error en el elemento *resu* del mensaje
 Envía el encabezado del mensaje al buzón BMAC
 Envía el encabezado del mensaje al buzón BEXC

P2.2.2.1.4 Funciones para acumuladores

Este procedimiento se encarga del manejo de los mensajes relacionados con los puntos llamados acumuladores que fluyen entre el sistema de cómputo y las UTR.

1. Recibe un apuntador al mensaje conteniendo la Base de Datos.
2. Obtiene los tuplos (UTR_ADQ, N_GPO, STS_ADQ, TIPO_GRU, TIPO_ADQ) que cumplan que TIPO_GRU sea del mismo valor que el tipo contenido en el mensaje
 Para todos los tuplos
 - De acuerdo al tipo obtiene de la relación UTR los valores de PRTO_UTR, STS_UTR, N_UTR
 - Con el valor de PRTO_UTR obtiene el estado del puerto de la relación PUERTO
 - Verifica si la tarjeta esta presente
 - Obtiene el protocolo del puerto
 - Realiza una inicialización solo para puertos PCS (*pro* >= PCS y *pro* <= MSTR)
 - Envía el mensaje al buzón BPRT.
 - Si hubo PROTOCOLO_INVALIDO llama al manejador de excepciones
3. Si hubo error de PUERTO_AUSENTE_O_FUERA o UTR_FUERA llama a ERR_FUNC

P2.2.2.2.1 BDD para Maestro

Este módulo se encarga de transformar la BDD con formato MAC al formato de estructuras utilizado en el CIC y de colocar el resultado en el área de memoria asignada.

1. Recibe un apuntador al mensaje, la via, y el tipo.
2. Copia la Base de Datos a su localidad
3. Si no hubo error
 - Obtiene el estado de los puertos y los coloca en la tabla *tab_sts*
 - Verifica que los puertos estén presentes
 - Envía el encabezado al buzón BMAC
4. Si hubo error
 - Coloca un código de error en el elemento *resu* del mensaje
 - Envía el encabezado del mensaje al buzón BMAC
 - Envía el encabezado del mensaje al buzón BEXC

P2.2.2.2.2 BDD para Puertos

Este módulo se encarga del manejo de los mensajes de inicialización y cargado de la BDD de todos los puertos presentes en la computadora del CIC-V2.

1. Recibe un apuntador al mensaje, la vía y el tipo.
2. Con el valor de puerto del mensaje obtiene el estado de puerto
3. Verifica si la tarjeta esta presente
4. Obtiene el protocolo del puerto
5. Realiza la inicialización sólo para puertos PCS
6. Si la función es CARG_LIN_MOD
 - Realiza un reset al puerto
 - Genera pulso de subida
 - Genera pulso de bajada
7. Si la vía es memoria
 - Envía el mensaje por referencia al buzón BPRT
 - si no
 - Envía el mensaje por copia al buzón BPRT
8. Si hubo algún error (PUERTO_AUSENTE_O_FUERA)
9. Envía el encabezado del mensaje al buzón BMAC y al buzón BEXC

P2.2.2.2.3 Solicitud Estadísticas

Este módulo se encarga de formar el mensaje con los datos a los que se les ha realizado estadísticas y lo envía al sistema de cómputo para su procesamiento.

1. Recibe un mensaje conteniendo la Base de Datos.
2. Obtiene todos los tuplos (N_UTR, STS_UTR, T_BARR) de la relación UTR que cumplan con el valor de *utr* del mensaje
3. Para todos los tuples
 - Con el valor de *utr* se obtiene PRTO_UTR
 - Con el valor anterior se obtiene de la relación PUERTO el elemento STS_PRTO
 - De acuerdo a los valores anteriores de encuentra si está en
 - t_normal*
 - t_lento*
 - t_fuera*
 - Se hace una llamada a la función ACUM_TPOS con el valor adecuado
4. Encuentra todos los tuples (UTR_EST, T_NORMAL, T_LENTO, T_FUERA Y EST) de la relación ESTAD
5. Crea un nuevo mensaje el cual va a pasar por referencia y le coloca los tuples al final del mismo
6. Envía el mensaje por referencia al buzón BMAC.

P2.2.2.2.4 Reset a Estadísticas

Este módulo registra el día y la hora en que el sistema de cómputo solicita que se limpien los valores de la estructura estadísticas y posteriormente realiza la acción.

1. Recibe un apuntador al mensaje.
2. Lee el reloj
3. Despliega la fecha y la hora
4. De la relación UTR encuentra los tuples (N_UTR, H_CAMBIO, D_CAMBIO) que cumplan con el valor de *utr* del mensaje
5. Para todos los tuples
actualiza H_CAMBIO y D_CAMBIO
6. De la relación ESTAD obtiene los tuples (UTR_EST, T_NORMAL, T_LENTO, T_FUERA, EST) que cumplan con el valor de *utr* del mensaje
7. Para todos los tuples
coloca ceros en todos los elementos del tuple
8. Si no hubo error envía al buzón BMAC el mensaje

P2.2.2.2.5 Carga Imagen

Este módulo se encarga de actualizar el área de memoria reservada para contener una imagen de la estructura estadísticas y de enviarla al sistema de cómputo.

1. Recibe un apuntador al mensaje
2. Obtiene de la relación UTR los tuples (N_UTR, STS_UTR, T_BARR) para N_UTR mayor o igual a 1
3. Para todos los tuples
Obtiene de la relación UTR el PRTO_UTR
Obtiene de la relación PUERTO el estado del Puerto de acuerdo al dato que se generó en un punto anterior
Encuentra si la UTR se encuentra en T_NORMAL, T_LENTO o T_FUERA
Llama a ACUM_TPOS con el estado obtenido
4. Obtiene todos los tuplos de la relación ESTAD y los coloca en el área de memoria de imagen
5. Construye un nuevo mensaje colocándole la imagen al final
6. Manda este nuevo mensaje por referencia al buzón del sistema de cómputo BMAC

P2.2.2.2.6 Envía Imagen

Este módulo se encarga de enviar los datos de la imagen de estadísticas a solicitud del sistema de cómputo.

1. Recibe un apuntador al mensaje
2. Construye un nuevo mensaje colocándolo al final los datos de la imagen de estadísticas
3. Envía el mensaje por referencia al buzón de la Estación Maestra BMAC

3.3.3. Proceso Concentrador

El diagrama de estructura perteneciente al proceso concentrador se muestra en la figura 3.5; como en el diagrama anterior se realiza la descripción de los módulos de arriba hacia abajo y de izquierda a derecha.

P3.0 Concentrador

En este módulo se ejecuta el ciclo principal correspondiente al proceso Concentrador.

- Repite en un ciclo infinito
 1. Recibe el un mensaje del buzón BRES
 2. Si no hay error
 - despliega el mensaje recibido
 - procesa el mensaje
 3. Si hay error
 - envía el encabezado del mensaje al manejador de errores por el buzón BEXC

P3.1 Recibe buzón BRES

Llamada a la rutina general para recepción de mensajes por un buzón, se pide leer el buzón BRES, con tamaño máximo de mensaje estándar, y con bloqueo.

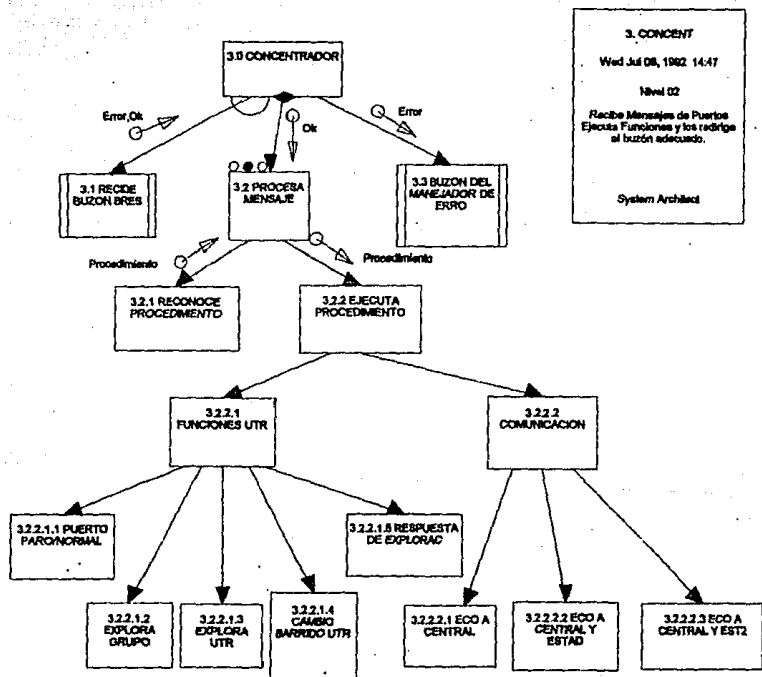


FIGURA 3.5 Proceso Concentrador

P3.2 Procesa mensaje

En este módulo se realiza el direccionamiento de los mensajes a el módulo adecuado para su procesamiento.

1. Busca en la tabla de funciones la que corresponda a la indicada en el mensaje
2. Direcciona a procesamiento relativo
3. Si no encuentra procesamiento
llama a excepción por FUERA_DE_RUTA
4. Si encuentra procesamiento
despliega un mensaje indicando el tipo de procesamiento que se realiza
ejecuta el procesamiento pasándole un apuntador al mensaje

P3.3 Buzón del manejo de error

Llama a la rutina general de envío por buzones después de colocar en el elemento *resu* un código de error, ajusta el elemento destino y coloca en *long* la longitud del encabezado, envía el mensaje por BEXC con bloqueo.

P3.2.1 Reconoce procedimiento

Hace una búsqueda lineal en la tabla de procesamiento para encontrar la función indicada en el mensaje y posteriormente direcciona a procesamiento relativo; la búsqueda se facilita conociendo la estructura de la tabla (ver tabla 1 en sección 3.2.4).

P3.2.2 Ejecuta procedimiento

De acuerdo al procesamiento solicitado hace la llamada adecuada para su ejecución después de haber desplegado un mensaje indicando el tipo de procesamiento de que se trata; le pasa como parámetro un apuntador al mensaje.

P3.2.2.1 Funciones UTR

Estos procedimientos se realizan cuando el puerto manda un mensaje de una acción que se efectuó en la UTR.

P3.2.2.2 Comunicación

En este módulo se agrupan las funciones que piden que el mensaje se pase a la estación maestra sin realizarle ningún procesamiento y también aquellas que piden además ser enviadas al proceso de estadísticas.

P3.2.2.1.1 Puerto Paro/Normal

Este módulo se encarga de enviar el comando de activación (modo normal) o de desactivación (modo paro) del puerto especificado en el mensaje, además de llevar un conteo de los tiempos acumulados del estado del puerto.

1. Recibe un apuntador a un mensaje
2. Con el valor del puerto obtiene cuantos tuples existen en la relación UTR que cumplan con igualar el valor en *prto_utr* y de que tamaño son
3. Si no hay error
 - a) Asigna suficiente memoria para los tuples
 - b) Obtiene los valores de tres atributos para cada tuple (*utr*, *sts*, *t_barr*)
 - c) De todos los tuples, de acuerdo a su *sts* y su *t_barr* se asignará un identificador que puede ser *t_normal*, *t_lento* y *t_fuera*
 - d) Con el identificador llama a ACUM_TPOS
 - e) libera la memoria asignada para los tuples
 - f) Si la función es FUN_M_NORMAL
 $sts_prto = 1$
 si no $sts_prto = 0$
 - g) llama a una actualización de la relación PUERTO con los nuevos valores
 - h) llama a ECO A CENTRAL y pasa como parámetro un apuntador al mensaje.

P3.2.2.1.2 Explora grupo

Este módulo recibe un mensaje con el estado de la adquisición de determinado grupo dentro de la UTR y actualiza la BDD en los campos pertinentes.

1. Recibe un apuntador al mensaje
2. Si la función del mensaje es FUN_GPO_DEN
 sts_adq igual a cero
 si no sts_adq igual a uno
3. Con los datos de *utr*, *n_gpo* y *sts_adq* se hace una llamada para actualizar la relación ADQ_GPOS
4. Llama a ECO A CENTRAL pasándole el mensaje

P3.2.2.1.3 Explora UTR

Este módulo recibe un mensaje conteniendo información del estado de la exploración de la UTR; con el tipo de barrido reportado se actualiza la BDD en los tiempos acumulados.

1. Recibe como parámetro un apuntador al mensaje
2. Con el valor de *utr* del mensaje obtiene de la relación UTR el número de puerto
3. Con el número de puerto obtiene el estado del puerto de la relación PUERTO
4. Con el estado del puerto y de la *utr* se encuentra el valor que defina el tipo de barrido
5. Modifica la relación UTR para actualizar el estado
6. Llama a acumula tipos con los valores de UTR y el identificador obtenido
7. Llama a ECO A CENTRAL con el mensaje

P3.2.2.1.4 Cambio de barrido a UTR

En este módulo se recibe un mensaje notificando un cambio de tipo de barrido para alguna UTR; con esta información se actualiza la estructura UTR en lo referente a tiempos acumulados y tipo de barrido.

1. Recibe un apuntador al mensaje
2. Con los datos *utr* y *t_barr* del mensaje se pide modificar la Estructura de Datos en su relación UTR
3. Con el dato *utr* del mensaje obtiene el número de puerto
4. Con el número de puerto obtiene el estado del puerto
5. Verifica que la UTR este en EXPLORACION
6. Si el barrido es BARRIDO NORMAL
 - se identifica ahora como *t_lento*
 - si no
 - se identifica como *t_normal*
7. Acumula los tiempos con el nuevo identificador
8. Llama a ECO A CENTRAL con el mensaje

P3.2.2.1.5 Respuesta de exploración

En este módulo se recibe el mensaje con la respuesta de exploración proveniente de alguna UTR, actualiza la BDD y envía el mensaje al sistema de cómputo.

1. Recibe un apuntador al mensaje, la vía y el tipo, la estructura están definidas en él
2. Envía por el buzón BEST el mensaje
3. Con el valor de *utr* obtiene el *gpo_act* de la relación ADQ_GPOS
4. Si no hay error
 - si *resu* = NO o *rosu* = RES_EXP
 - si *gpo_act* = NO
 - gpo_act* = 0ffH
 - actualiza la relación ADQ_GPOS
 - si *resu* diferente de NO o RES_EXP
 - si *gpo_act* diferente de NO
 - gpo_act* = 00
 - actualiza la relación ADQ_GPOS
 - si no hay error
 - si *resu* diferente de RES_EXP
 - si la vía es memoria
 - envía el mensaje por referencia al buzón BMAC
 - si no
 - envía el mensaje por copia al buzón BMAC
 - si no
 - libera la memoria del mensaje por referencia
 - construye un nuevo mensaje con los valores obtenidos durante el procesamiento y manda por copia el mensaje al buzón BMAC
 - 5. Si hubo error
 - llama a ERROR_FUNC
 - si la vía fue memoria
 - libera la memoria utilizada

P3.2.2.2.1 Eco a Central

Este módulo se encarga de redirigir todos los mensajes hacia el sistema de cómputo.

1. Recibe un apuntador a un mensaje
2. Si la función es CARG_LIN_MOD ó BDD_HAR_62
 - obtiene con el valor puerto del mensaje el *sts_prto* de la relación PUERTO
 - modifica el estado con la operación STS_PRTO AND INIC_PRTO
 - actualiza la Estructura de Datos con los nuevos valores
3. Si no hay errores
 - envía por el buzón BMAC el mensaje
4. Si hay errores
 - manda el mensaje al manejador de errores

P3.2.2.2.2 Eco a Central y Estadísticas

Este procedimiento se encarga de redirigir el mensaje proveniente del puerto de enlace con las UTR hacia el proceso de ESTADÍSTICAS y hacia la Computadora Central

1. Recibe un apuntador al mensaje, la vía y el tipo
2. El mensaje apuntado se envía al buzón BEST
3. Si la vía es memoria
se envía el mensaje por referencia al buzón BMAC
4. Si la vía no es memoria
se envía por copia al buzón BMAC

P3.2.2.2.3 Eco a Central y Estadísticas 2

Este módulo es funcionalmente el mismo que el anterior; pero es utilizado para números de funciones diferentes.

1. Recibe un apuntador al mensaje, la vía y el tipo
2. El mensaje apuntado se envía al buzón BEST
3. Si la vía es MEMORIA
envía por referencia al buzón BMAC
4. Si la vía no es MEMORIA
envía por copia el mensaje al buzón BMAC

3.3.4. Proceso Estadísticas

La figura 3.6 muestra el diagrama correspondiente a este proceso. El pseudocódigo de los módulos que lo forman también se describe a continuación.

P4.0 Estadísticas

Este módulo contiene el ciclo principal del proceso de estadísticas.

Repite en un ciclo infinito

1. Recibe el un mensaje del buzón BEST
2. Si no hay error
despliega el mensaje recibido
procesa el mensaje
3. Si hay error
envía el encabezado del mensaje al manejador de errores por el buzón BEXC

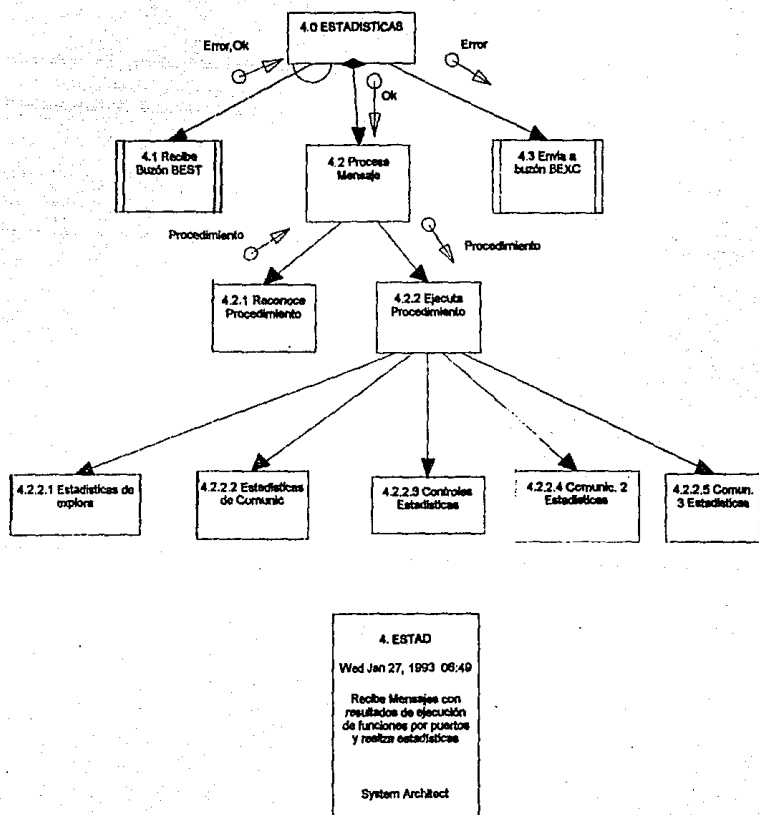


FIGURA 3.6 Proceso Estadísticas

P4.1 Recibe Buzón BEST

Llamada a la rutina general para recepción de mensajes por un buzón, se pide leer el buzón BEST, con tamaño máximo de mensaje estándar, y con bloqueo.

P4.2 Procesa Mensaje

Este módulo se encarga de direccionar el mensaje al procedimiento adecuado, tomando en cuenta el número de la función del mensaje y el proceso de que se trata.

1. Busca en la tabla de funciones la que corresponda a la indicada en el mensaje
2. Direcciona a procesamiento relativo
3. Si no encuentra procesamiento
llama a excepción por FUERA_DE_RUTA
4. Si encuentra procesamiento
despliega un mensaje indicando el tipo de procesamiento que se realiza
ejecuta el procesamiento pasándole un apuntador al mensaje

P4.3 Envía a buzón BEXC

Llama a la rutina general de envío por buzones después de colocar en el elemento RESU un código de error, ajusta el elemento DESTINO y coloca en LONG la longitud del encabezado, envía el mensaje por BEXC con bloqueo.

P4.2.1 Reconoce Procedimiento

Hace una búsqueda lineal en la tabla de procesamiento para encontrar la función indicada en el mensaje y posteriormente direcciona a procesamiento relativo (ver tabla 1 en sección 3.2.4). Regresa el número identificador del procesamiento requerido

P4.2.2 Ejecuta Procedimiento

De acuerdo al procesamiento solicitado hace una llamada a la rutina adecuada para su ejecución después de haber desplegado un mensaje indicando el tipo de procesamiento de que se trata; le pasa como parámetro un apuntador al mensaje.

P4.2.2.1 Estadísticas de exploración

Este módulo lleva las estadísticas de las exploraciones realizadas a las UTR por los módulos esclavos.

1. Recibe un apuntador al mensaje
2. De acuerdo al valor de *utr* en el mensaje obtiene de la relación ESTAD el elemento EST
3. Si no hubo error
 - a) Incrementa estadísticas de INT_EXP
 - b) Si el elemento *resu* del mensaje tiene valor de NO o RES_EXP incrementa estadísticas de EXI_EXP
Si no incrementa estadísticas de FAL_EXP
 - c) Si el elemento *resu* es mayor que INC_CICLO incrementa estadísticas de INC_CIC_EXP
 - d) Actualiza los nuevos valores en la estructura de datos
 - e) Hace una llamada a ESTADISTICAS DE COMUNICACIONES pasando como parámetro un apuntador al mensaje
4. Si hubo error llama a ERROR_FUNC pasando un apuntador al mensaje

P4.2.2.2 Estadísticas de Comunicaciones

Este módulo se encarga de llevar las estadísticas de las comunicaciones entre los puertos y el módulo maestro del CIC-V2 (intentos de comunicación a la UTR, intentos exitosos, intentos con falla, etc.).

1. Recibe un apuntador al mensaje
2. Con el valor de *utr* del mensaje obtiene
 - a) De la relación ESTAD el elemento *est*
 - b) De la relación UTR el elemento *prto_utr*
3. Si no hubo error

Si el valor de *resu* del mensaje es menor que ERR_COM_UTR coloca los siguientes parámetros al mensaje

long = LON_BAR_UTR
func = FUN_BAR_UTR
n_msj = 0
 actualiza el destino
 incrementa las estadísticas para INT_COM_UTR
 Si se cumple que (*resu* = 0) o (*resu* >= ERR_INF_EXI_UTR y *resu* <= ERR_SUP_EXI_UTR)
 incrementa las estadísticas para EXI_COM_UTR
 actualiza el valor de *est* a la relación ESTAD
 obtiene *t_barr* de la relación UTR con el valor de *utr*
 Si no hay error
 coloca un cero en fallas
 si el *t_barr* es BAR_LENTO
t_barr = BAR_NORMAL

- resu* = NO
 envía por el buzón BPRT el mensaje con bloqueo
 actualiza en la relación UTR los nuevos datos
- Si no se cumple la condición anterior de *resu*
 incrementa las estadísticas para FAL_COM_UTR
 incrementa las estadísticas para *resu*+6
 actualiza el valor de *est* a la relación ESTAD. De la relación UTR y
 con el valor de *utr* del mensaje obtiene
- t_barr*
 - fallas*
 - n_int*
- si no hay error
 incrementa el número de fallas
 si fallas es mayor que *n_int*
 si el *t_barr* es BAR_NORMAL
t_barr = BAR_LENTO
resu = NO
 envía por el buzón BPRT el mensaje con bloqueo
 actualiza en la relación UTR los nuevos datos
5. Si hubo error
 llama a ERROR_FUN pasándole un apuntador al mensaje

P4.2.2.3 Estadísticas de Controles

Este módulo se encarga de llevar la cuenta de estadísticas de controles (controles intentados, controles exitosos, controles con falla, etc.).

- Recibe un apuntador al mensaje
- Con el valor de *utr* del mensaje obtiene de la relación ESTAD el elemento *EST*
- Si no hay error
 incrementa las estadísticas de INT_CTR
 Si el valor de *resu* del mensaje es NO
 incrementa las estadísticas de EXI_CTR
 Si el valor de *resu* es diferente de NO
 incrementa las estadísticas de FAL_CTR
 Si *resu* es ERR_NOP
 incrementa las estadísticas de CON_NO_OPER
 Si *resu* NO es ERR_NOP
 Si *resu* >= ERR_INF_COM y *resu* <= ERR_SUP_COM
 incrementa las estadísticas de CON_ERR_COM
 si no
 si *resu* >= ERR_MOD_CAN
 incrementa las estadísticas de
 CON_MOD_CAN
 actualiza la relación ESTAD con los nuevos datos
 llama a ESTADISTICAS DE COMUNICACIONES pasándole un
 apuntador al mensaje
- Si hay error
 llama a ERROR_FUNC pasándole un apuntador al mensaje

P4.2.2.4 Estadísticas de Comunicaciones 2

En este módulo se realizan estadísticas de comunicaciones para otras de las funciones. (intentos de comunicación con la UTR, intentos exitosos, fallidos, etc.).

1. Recibe un apuntador al mensaje
2. Con el valor de *utr* del mensaje obtiene el elemento *est* de la relación ESTAD
3. Si no hay error
 - Incrementa las estadísticas de INT_COM_UTR
 - si *resu* = NO o (*resu* >= ERR_INF_EXI_UTR y *resu* <= ERR_SUP_EXI_UTR)
 - incrementa las estadísticas de EXI_COM_UTR
 - si no
 - Incrementa las estadísticas de FAL_COM_UTR
 - incrementa las estadísticas de *resu*+6
 - actualiza la relación ESTAD con los nuevos datos
4. Si hay error
 - llama a ERROR_FUNC pasándole como parametro un apuntador al mensaje

P4.2.2.5 Estadísticas de Comunicaciones 3

Este es el último módulo en el cual se llevan estadísticas del comportamiento de comunicación con la UTR. (intentos de exploración, exploraciones exitosas, con falla, etc.).

1. Recibe un apuntador al mensaje
2. Con el valor de *utr* del mensaje obtiene el elemento EST de la relación ESTADIST
3. Si no hay error
 - incrementa las estadísticas de INT_EXP
 - si *resu* = NO o (*resu* >= ERR_INF_EXI_UTR y *resu* <= ERR_SUP_EXI_UTR)
 - incrementa las estadísticas de EXI_EXP
 - si no
 - incrementa las estadísticas de FAL_EXP
 - incrementa las estadísticas de *resu*+6
 - actualiza la relación ESTADIST con los nuevos datos
4. Si hay error
 - llama a ERROR_FUNC pasándole como parámetro el apuntador al mensaje

3.3.5. Proceso Manejador de errores

En la figura 3.7 se muestra el diagrama de estructura correspondiente al proceso manejador de errores, indicando un flujo interno de datos. Se describen en las siguientes subsecciones las funciones ejecutadas por cada uno de sus módulos.

P5.0 Manejador de Errores

Este proceso recibe los errores detectados, clasifica el error, realiza las operaciones pertinentes y avisa a la computadora central del sistema. Lleva además un diagnóstico de los procesos del maestro del CIC informando el estado actual de cada uno de ellos (bloqueado, activo, en espera de cambio de contexto, etc.).

P5.1 Recibe mensaje de buzón BEXC

En este módulo se realiza la recepción del mensaje y la clasificación del mismo, de donde se deduce si es error fatal, no fatal o diagnóstico.

1. Recibe mensaje por buzón BEXC
2. Busca en la tabla de errores el tipo de error correspondiente
3. Si es error fatal
 llama al procedimiento que reinicializa el CIC
4. Si es error no fatal
 informa al sistema de cómputo del error ocurrido
5. Si es diagnóstico
 cambia el estado del proceso al indicado en el mensaje

P5.2 Ejecuta procedimiento

Este módulo se encarga de realizar la acción a seguir en caso de diagnóstico o de error no fatal.

1. Recibe el resultado de la evaluación del tipo de error indicado en el mensaje
2. Si es error no fatal
 despliega la leyenda correspondiente al número de error encontrado
 construye un mensaje y lo envía al sistema de cómputo a través del buzón BMAC
3. Si es diagnóstico
 selecciona el color correspondiente al estado indicado del proceso
 actualiza en el área correspondiente de la ventana IHM el estado del proceso

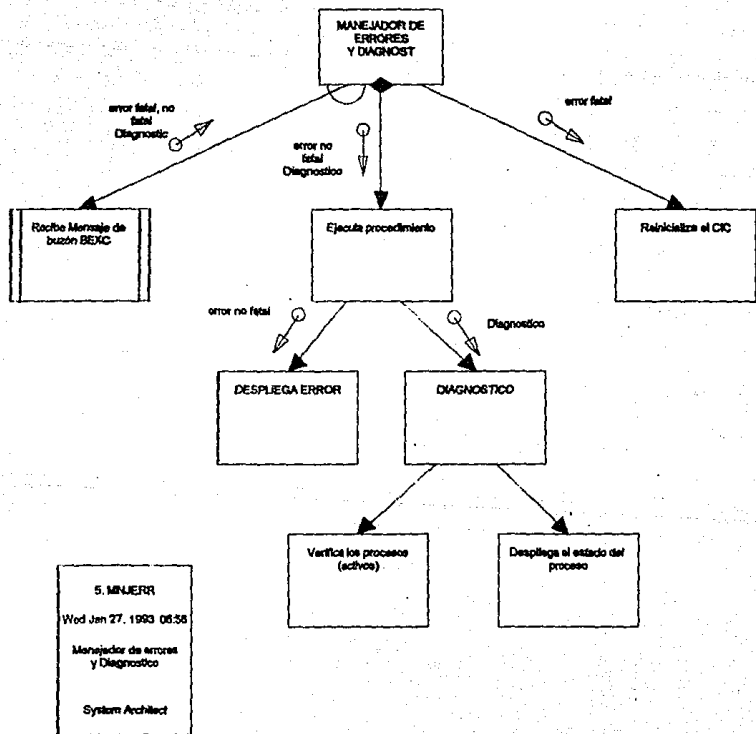


FIGURA 3.7 Proceso Manejador de errores

P5.3 Reinicializa el CIC

En este módulo se hace una llamada para solicitar al sistema de cómputo que envíe los datos necesarios para la reinicialización del CIC-V2

1. Construye un mensaje indicando al sistema de cómputo que ocurrió un error fatal
2. Envía el mensaje anterior por el buzón BMAC
3. Realiza las operaciones de inicialización del CIC
 - Espera BDD del maestro
 - Espera BDD de esclavos

3.3.6. Proceso de enlace al sistema de cómputo (lectura)

Este es el proceso manejador encargado de la recepción de mensajes directamente del puerto de enlace al sistema de cómputo y de la construcción del mensaje a enviar al buzón BDIS. En la figura 3.8 se muestra el diagrama de estructura obtenido para este proceso.

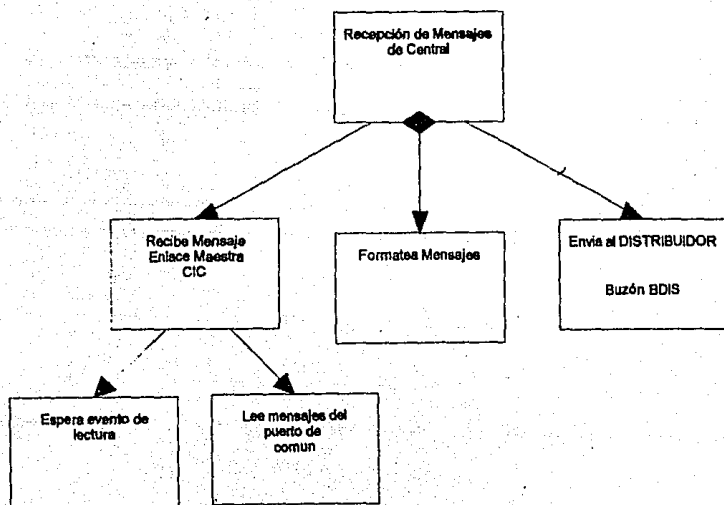
P6.0 Recepción de Mensajes de Maestra

Este es el módulo principal del proceso de enlace de lectura hacia el sistema de cómputo.

1. Envía al manejador de errores una señal indicando que el proceso está activo
2. Recibe la señal que indica que existen datos en el FIFO listos para ser leídos
3. Lee "byte" por "byte" del FIFO y construye un mensaje con ellos
4. Envía el mensaje al proceso DISTRIBUIDOR mediante el buzón BDIS
5. Envía al manejador de errores una señal indicando que el proceso terminó un ciclo
6. Regresa al paso 1.

P6.1 Enlace Maestra - CIC

En este módulo se espera la señal de interrupción de la tarjeta PC-851 indicando que el FIFO tiene datos listos para ser leídos, entre cada uno de los "bytes" leídos tiene que esperar un tiempo para asegurar la sincronización con el FIFO de la tarjeta. Lee tantos datos como los indicados por los dos primeros "bytes" que se encuentran y los datos leídos los va almacenando en un arreglo para su uso posterior.



6. LECT1
Wed Jan 27, 1993 07:03
Nivel 02
Manejador encargado de recepción de Mensajes directamente del Puerto
System Architect

FIGURA 3.8 Proceso de enlace al sistema de cómputo (lectura)

P6.2 Formatea Mensajes

En este módulo se construye el mensaje a partir de los datos contenidos en el arreglo; el formato del nuevo mensaje se convierte a la forma *encabezado - datos*, el cual queda listo para ser transmitido al buzón BDIS.

P6.3 Envía al DISTRIBUIDOR

En esta parte se envía el mensaje construido al proceso DISTRIBUIDOR a través del buzón BDIS y se libera la memoria utilizada por el mensaje de manera que el siguiente mensaje recibido se coloque en la misma área de memoria, optimizando de esta manera la transmisión de mensajes del sistema de cómputo al proceso de lectura.

3.3.7. Proceso de enlace al sistema de cómputo (escritura)

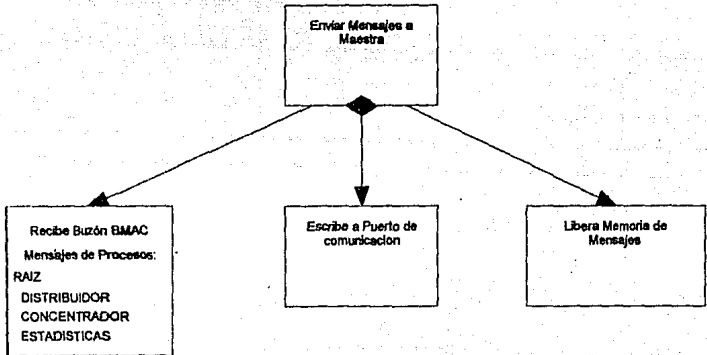
Este proceso es el encargado de enviar los mensajes al sistema de cómputo a través de la tarjeta PC-851 destinada para tal fin. La comunicación se realiza cuando el módulo maestro deposita el mensaje en el FIFO de la tarjeta. La figura 3.9 ilustra el diagrama de estructura de este proceso.

P7.0 Enviar Mensajes a sistema de cómputo

Este es el módulo principal del proceso de enlace hacia el sistema de cómputo en lo concerniente a escritura.

- Ejecuta en un ciclo infinito

1. Envía al proceso manejador de error una señal indicando que el proceso se encuentra activo
2. Espera a recibir un mensaje a través del buzón BMAC
3. Escribe byte por byte al FIFO de la tarjeta PC-851
4. Envía al proceso manejador de error una señal indicando que el proceso terminó un ciclo



7. ESCR1
 Wed Jan 27, 1993 07:13
 Nivel 02
 Manejador de escritura
 de mensajes para enlace
 CIC - Estación Maestra
 System Architect

FIGURA 3.9 Proceso de enlace al sistema de cómputo (escritura)

P7.1 Recibe buzón BMAC

En este módulo se espera el arribo de un mensaje por el buzón BMAC, mismo que puede provenir de cuatro procesos diferentes (RAIZ, DISTRIBUIDOR, CONCENTRADOR y ESTADISTICAS). El mensaje recibido se coloca en un área de memoria local y queda listo para ser enviado al sistema de cómputo. Una vez recibido el mensaje se envía al proceso manejador de errores un mensaje indicando el principio del ciclo de escritura de este proceso.

P7.2 Escribe al puerto de comunicación

Este módulo se encarga de realizar la escritura del mensaje al FIFO de la tarjeta PC-851, lo cual hace "byte" por "byte" con un tiempo de espera entre cada uno de ellos para asegurar la sincronización con la tarjeta a la que se escribe.

P7.3 Libera memoria de mensajes

En este módulo se libera la memoria asignada al mensaje en caso de que la transmisión del mismo se halla realizado por referencia y se envía al proceso manejador de errores un mensaje indicando que el proceso terminó un ciclo.

3.3.8. Proceso de enlace hacia las UTR (lectura)

Este proceso es funcionalmente semejante al descrito en el punto 3.3.6 con la única variante en las señales que maneja se esperan de la tarjeta con el enlace hacia las UTR y el resultado se envía al buzón BRES para su procesamiento posterior. En la figura 3.10 se muestra el diagrama de estructura de este proceso.

3.3.9. Proceso de enlace hacia las UTR (escritura)

Este es el último proceso del CIC-V2, en el cual se tiene una estructura muy similar a la descrita con el enlace al sistema de cómputo (punto 3.3.8), solo que ahora los datos tienen que escribirse al FIFO de la tarjeta de comunicación hacia las UTR. La figura 3.11 muestra el diagrama de estructura correspondiente.

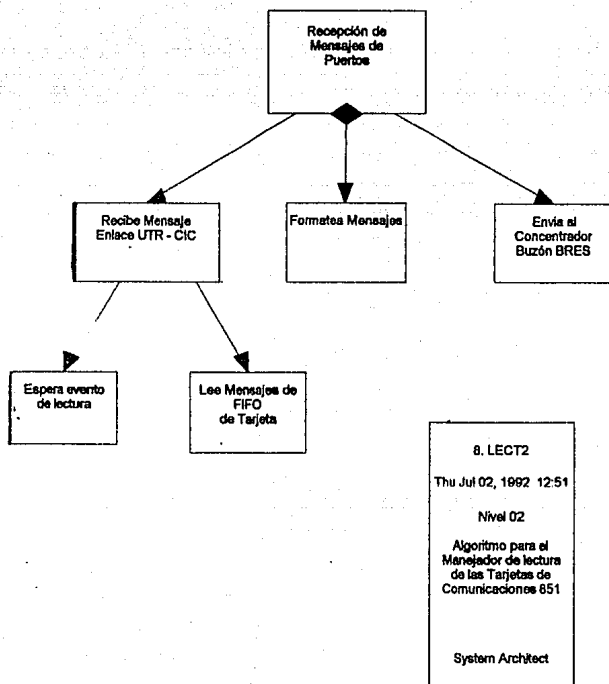
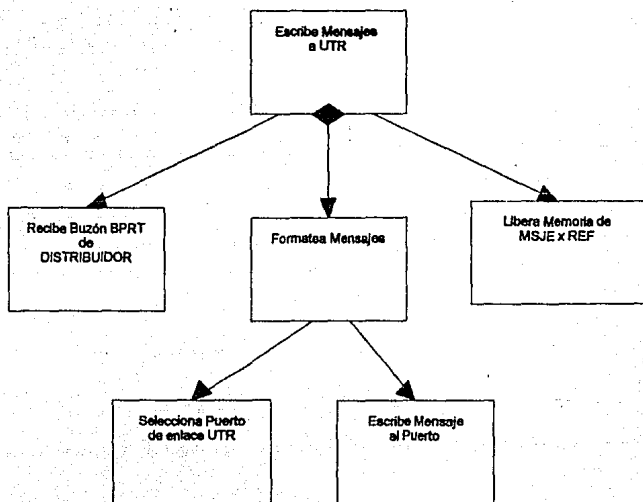


FIGURA 3.10 Proceso de enlace hacia las UTR (lectura)



9. ESCR2
 Thu Jul 02, 1992 12:44
 Nivel 02
 Algoritmo para el manejador de escritura de Mensajes a las Memorias FIFO
 System Architect

FIGURA 3.11 Proceso de enlace hacia UTR (escritura)

Capítulo 4

PLAN DE PRUEBAS Y VERIFICACION FUNCIONAL DEL MODULO MAESTRO DEL CIC

Se presenta en este capítulo el procedimiento para probar el funcionamiento del Controlador Inteligente de Comunicaciones en computadora personal (CIC-V2) principalmente enfocado hacia los programas del módulo maestro. El plan de pruebas describe el alcance, recursos y técnicas de las actividades de pruebas, identificando los elementos de prueba, las tareas ejecutadas y las acciones a seguir de acuerdo a los resultados de las pruebas

Asimismo, se presentan los resultados de la verificación funcional del sistema.

4.1. Proceso de pruebas de software

El plan de pruebas para el CIC - V2 esta basado en una parte de la norma ANSI/IEEE Std 829 - 1983 (*IEEE Standard for Software Test Documentation*) [IEE 83] la cual describe un conjunto de elementos básicos asociados con los aspectos dinámicos de las pruebas de software. El propósito de este plan es describir el alcance, técnicas, y recursos; así como la identificación de los elementos de prueba, las tareas de pruebas ejecutadas, y los riesgos asociados con este plan.

Al probar los sistemas en tiempo real, como el CIC-V2, existen problemas peculiares provocados por las interacciones entre los diversos procesos del sistema. Los errores pueden depender del tiempo y surgir sólo cuando los procesos del sistema se encuentran en un estado determinado. El estado exacto de los procesos del sistema, cuando ocurrió el error, puede ser imposible de reproducir. De aqui la importancia de la especificación del proceso de pruebas. [SOM 88]

Muchas veces, la prueba de los sistemas de tiempo real se complica aún más, porque los requisitos de confiabilidad de esos sistemas suelen ser mayores que los requisitos para sistemas que no dependen del tiempo. Las consecuencias de una falla en el sistema son potencialmente desastrosas; debido a ello, la prueba del sistema debe ser lo más exhaustiva y rigurosa posible.

Para el CIC-V2, las actividades de prueba comprenden únicamente pruebas a nivel laboratorio, excluyendo en este esquema las pruebas de aceptación y de validación en campo. El plan de pruebas hace énfasis en los siguientes puntos:

- Verificar que cada componente cumpla con las especificaciones del diseño
- Verificar cada componente contra sus requerimientos de funcionalidad
- Verificar que los componentes de interfaz transfieran la información apropiadamente

4.2. Plan de pruebas para el CIC - V2

4.2.1. Objetivos y alcance del plan de pruebas

El plan de pruebas para el CIC-V2 tiene los siguientes objetivos y alcance.

- *Objetivos*

1. Detallar las actividades requeridas para preparar y conducir las pruebas del sistema.
2. Comunicar a las partes responsables las tareas que se ejecutarán y la secuencia a seguir para su realización
3. Definir las fuentes de información usadas para preparar el plan
4. Definir las herramientas de prueba y las necesidades del medio ambiente requeridas para ejecutar las pruebas

- *Alcance*

Este plan cubre la totalidad de las pruebas a ejecutar para este software. Incluye las de procedimientos, de procesos (como cajas negras), del sistema completo y de integración con otros equipos, se analiza su robustez, su respuesta en tiempo, la capacidad para manejar sobrecarga de información, su interfaz con el usuario y su funcionalidad.

4.2.2. Elementos de prueba

Los elementos que constituyen al CIC-PC considerados para las pruebas son los siguientes:

- *Programas del Módulo Maestro del CIC-V2*

Los módulos de programas probados en la computadora personal se identifican de la siguiente manera:

Nombre	Tipo	Archivos
Proceso Concentrador	Código Fuente	CONCENT.C CONPROC.C
Proceso Distribuidor	Código Fuente	DISTCIC.C DISPROC.C
Proceso Estadísticas	Código Fuente	ESTADIST.C ESTPROC.C
Proceso Manejador de error	Código Fuente	MNJERROR.C
Proceso Manejador de Lectura a PC-851	Código Fuente	LECT1.C LECT2.C
Proceso Manejador de Escritura a PC-851	Código Fuente	ESCR1.C ESCR2.C
Proceso Raíz	Código Fuente	RAIZ.C
CIC-PC	Archivo ejecutable	CIC-PC.EXE

- Procedimientos de control de la tarjeta PC-851

Los procedimientos de control para la PC-851 se identifican de la siguiente manera:

Nombre	Tipo
Protocolo de enlace con la Unidad Central de Procesamiento	Programas de aplicación grabados en una memoria EPROM (en PC-851)
Protocolo de enlace con alguna UTR	Programas de aplicación grabados en una memoria EPROM (en PC-851)
Comunicación a través del bus de la PC	Programa manejador de interrupción con lectura y escritura a FIFO's Código fuente (en PC)

4.2.3. Características probadas y excluidas

- Características probadas

La siguiente lista describe las características principales a probar:

- 1) El correcto funcionamiento de cada módulo de acuerdo a la especificación
- 2) Conversión y cargado de la base de datos del sistema de cómputo al CIC
- 3) Cargado de la base de datos a los puertos de la tarjeta PC-851
- 4) El correcto enlace hacia el sistema de cómputo
- 5) El correcto enlace hacia las Unidades Terminales Remotas
- 6) Funcionalidad de todo el sistema

- 7) Tiempo de respuesta desde que llega un mensaje al CIC hasta que lo retransmite o contesta
- 8) El comportamiento del CIC en caso de error
- 9) El comportamiento del CIC con sobrecarga de información
- 10) Interfaz con el usuario

- Características excluidas

Los casos de prueba no cubren la totalidad de combinaciones posibles de comunicación de mensajes entre procesos, solo aquellos casos que resultaron significativos para su evaluación. La totalidad de las pruebas se realizan en laboratorio, no incluyendo pruebas en campo ni de aceptación y validación.

4.2.4. Técnicas

El propósito de estas técnicas es especificar las principales actividades y herramientas que se utilizan para probar las características de los grupos designados de prueba. Para cada grupo se requiere identificar las etapas necesarias para su realización.

- Prueba del correcto funcionamiento de cada módulo

Esta prueba se realiza a nivel del código fuente primero (prueba de escritorio) y después considerando el archivo ejecutable. En la primera etapa se revisa que el código programado realice las funciones especificadas en el documento de diseño, se revisan todos los procedimientos que se codificaron para cada uno de los procesos y se asegura que realizan correctamente su función.

Posteriormente se considera a los procesos como "cajas negras" y se ejecuta el programa para observar su comportamiento con ciertos casos de prueba seleccionados previamente de acuerdo a las funciones realizadas. La filosofía a seguir es enviar un mensaje a un buzón esperando que el proceso se encargue de redirigirlo o tomar la acción pertinente, pudiendo en cualquier momento examinar el contenido de los buzones o de la base de datos para verificar que la acción ejecutada cumple completamente con la especificación inicial. Los casos de prueba se seleccionan cuidadosamente, de manera que sin probar la totalidad de casos posibles se indica con bastante claridad el comportamiento del sistema.

- Prueba de enlace hacia la Unidades Terminales Remotas

La primera etapa consiste en probar el funcionamiento de las rutinas de lectura y escritura hacia la tarjeta de comunicaciones PC-851, es decir, probar las rutinas de reconocimiento de interrupción hacia la tarjeta simulando la señal de interrupción por medio de una interrupción de teclado, lo que reditúa en contar con la seguridad de que los manejadores de interrupción funcionan correctamente y que la comunicación hacia la tarjeta es correcta.

Como segunda etapa de prueba se verifica que el software de aplicación que contiene los protocolos y el manejo interno de la tarjeta PC-851 funciona al colocarle la memoria EPROM que actualmente se utiliza en las tarjetas MAC-851 de la Estación Maestra MAC-5000 [VAR 92]. Esta verificación se realiza utilizando un programa comercial de comunicaciones llamado "*PROCOMM PLUS*", el cual permite enviar y recibir datos a través del canal serie; los programas de prueba contienen rutinas que emulan el funcionamiento de una terminal remota DTE-532 DL con el protocolo de comunicación COM-832. Al realizar el enlace de la tarjeta PC-851 con una computadora que esté ejecutando el *PROCOMM PLUS* se pueden efectuar todas las funciones básicas de exploración desde el módulo maestro del CIC-V2 hacia la UTR emulada y se verifica el comportamiento de las comunicaciones desde la IHM local del CIC-V2.

- Prueba de enlace hacia el sistema de cómputo

Esta prueba es muy similar a la anterior. La única diferencia es que los programas de aplicación contenidos en la EPROM son diferentes, ya que el protocolo hacia el sistema de cómputo difiere de los protocolos utilizados para las UTR.

Estas pruebas se realizan conectando la tarjeta de comunicaciones PC-851 con una terminal la cual permite enviar mensajes a través de la PC-851 y recibir mensajes enviados por el CIC, visualizando en todo momento tanto los datos de entrada como los de salida (fig. 4.1).

Las principales características a probar son:

- El correcto funcionamiento de los procesos de enlace.
- El cargado de la BDD del módulo maestro.
- La correcta conversión de la BDD del módulo maestro al formato MAC cuando se requiera un cargado de imagen por el sistema de cómputo.

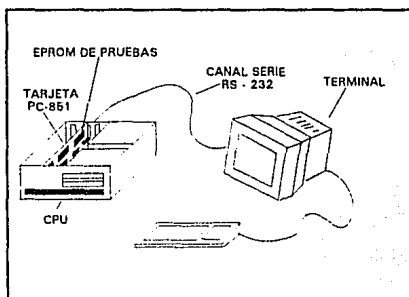


FIGURA 4.1 Prueba de funcionamiento de la tarjeta PC-851

- Funcionalidad del sistema

1ª Etapa. Simulación

Después de efectuar las pruebas de enlace hacia la Maestra y hacia las UTR se realiza una simulación conectando un canal serie al sistema de cómputo y otro a una computadora en donde se ejecuta PROCOMM PLUS, de manera que el sistema de cómputo envía comandos de exploración, los cuales se reciben en la computadora con PROCOMM, programada con las respuestas a cada comando.

2ª Etapa. Conexión real

Una vez que el sistema se prueba y depura bajo la simulación, se procede a realizar la conexión hacia el sistema de cómputo y hacia la UTR ó a un equipo de diagnóstico (Diagnoster) para verificar el correcto funcionamiento del sistema bajo condiciones reales de trabajo. El Diagnoster es un equipo que emula la mayoría de las funciones realizadas por la UTR, así como los protocolos de los distintos fabricantes de UTR.

- Tiempo de respuesta

El tiempo de respuesta se mide incluyendo variables al programa fuente para que registre la hora de ocurrencia de cuando se recibe el mensaje y la hora de cuando termina de enviarse, la diferencia de tiempo nos da el retardo real del CIC-V2 para realizar los procedimientos. Este tiempo se estima de acuerdo a los mensajes más largos, los mensajes promedio y los mensajes cortos.

- Comportamiento del CIC-V2 en caso de error

Una vez logrado el funcionamiento correcto del CIC-V2 se le envían mensajes incorrectos conocidos para observar si el manejo de los mismos es adecuado.

- Conversión y cargado de la base de datos de la Estación Maestra

En esta etapa se procede a realizar la conexión real con la Estación Maestra MAC-5000 y se verifica que la base de datos se localice en el área de memoria respectiva, se hace la comparación de los datos que envía la Estación Maestra con los datos que esta recibiendo el CIC-V2.

- Comportamiento del CIC-V2 con sobrecarga de información

Esta prueba se realiza con una conexión real con la Estación Maestra MAC-5000 y con las UTR (o el Diagnoster) y consiste en enviar hacia el CIC-V2 la mayor cantidad de mensajes de comunicación entre Estación Maestra y UTR (o Diagnoster), de manera que se pueda tomar una conclusión del comportamiento del CIC-V2 cuando se encuentra con una gran demanda de trabajo; para lograr esto los mensajes transmitidos hacia el CIC-V2 deberán ser enviados con la mayor frecuencia posible.

- Interfaz con el usuario

Como en la versión anterior del CIC no se contaba con una interfaz hombre máquina, en el CIC-V2 se ejecutan pruebas a la interfaz diseñada, la cual contempla el diagnóstico de los procesos del CIC-V2. Esta IHM no es indispensable para la ejecución del sistema, pero si es útil para la depuración.

4.2.5. Criterios de aceptación y rechazo

El sistema debe cumplir con lo especificado en el capítulo de especificación y diseño. La aceptación del sistema se basa en el hecho de ser funcionalmente igual o mejor que la versión anterior del CIC.

Como cada proceso cuenta con una serie de rutinas asociadas a él, se considera como válido solamente cuando cumple con todas las funciones asignadas, de no ser así se tiene que verificar la programación de cada función errónea hasta cumplir perfectamente con todos los requerimientos del proceso.

4.2.6. Criterios de suspensión y requerimientos de continuación

El sistema debe satisfacer el principal requerimiento que es el de tiempo de respuesta, si no cuenta con un tiempo adecuado se nulifica la interfaz con el usuario para darle mayor velocidad a los procesos, aunque se puede seguir utilizando para checar la funcionalidad de las rutinas implementadas.

Otro punto crítico es el cargado de la base de datos del sistema de cómputo y el cargado de la base de datos hacia los puertos, si alguno falla, se trata inmediatamente de corregir el error puesto que las demás funciones dependen en su mayoría de la base de datos.

Las interrupciones de la tarjeta de comunicaciones también resultan un factor importante para las pruebas ya que si no existe una perfecta comunicación, la integridad del sistema no se garantiza.

Si se detectan errores o modificaciones mayores que afecten al sistema se realiza una nueva versión del software y se repiten todas las pruebas desde el inicio; si las fallas sólo afectan al proceso bajo prueba únicamente se repiten las pruebas para este proceso y los que interactúan con él.

4.2.7. Requerimientos

Se necesita una computadora PC Olivetti M280 con procesador 80286 a 12 MHz en la que correrá el CIC - V2, adicionalmente se requiere lo siguiente:

Etapa 1.

- 2 Tarjetas PC-851
- 1 EPROM conteniendo el programa de pruebas a la tarjeta
- 1 terminal de video
- Información de los procedimientos de prueba en la EPROM
- Cables y conectores

Etapa 2.

- 1 EPROM con los protocolos empleados actualmente en la tarjeta MAC-851, los cuales son transportables a la PC-851
- La Estación Maestra MAC-5000
- Una UTR o el equipo Diagnoster
- Cables y conectores

- Riesgos y contingencias

En caso de que la velocidad de procesamiento requerida del CIC no le permita tener una interfaz con el usuario se puede reemplazar la computadora por una más veloz (con procesador 80386 por ejemplo) para hacer pruebas antes de deshabilitar esa función.

Si la memoria EPROM que contiene los protocolos no funciona correctamente al colocarla en la tarjeta PC-851 se tiene que realizar mediante un estudio más profundo del protocolo en prueba los ajustes necesarios.

4.3. Casos de prueba

Los casos de prueba para los procesos se llevan a cabo sólo cuando los demás procesos no están activos, de manera que se pueda observar los mensajes que se envían a los otros procesos con solo leerlos del buzón respectivo.

4.3.1. Proceso concentrador

DATOS DE ENTRADA AL BUZÓN BRES	SALIDAS ESPERADAS
<p>Se envía el encabezado del mensaje para cada una de las funciones especificadas en función.</p> <p>longitud : 8 función : 11, 12, 14, 15, 16, 23, 28, 30, 36, 37, 38, 39, 41, 63, 64, 65, 66, 69, 70, 71 resu : 1 n_msje : 1 destino : 1 (para todos los mensajes)</p>	<p>Se espera el encabezado del mensaje en el buzón BMAC (Escr. Maestra)</p> <p>Se pueden cambiar la longitud del mensaje y enviar los datos que se deseen, esos datos deben ser recibidos por el buzón exactamente como fueron enviados.</p> <p>En todos los mensajes <i>destino</i> debe tener el valor de 49.</p>
<p>Se envía el encabezado del mensaje seguido del número de puerto al que se envía la inicialización</p> <p>longitud : 10 función : 13, 62 resu : 1 n_msje : 1 destino : 1 datos : 1 0 (para función 13) 2 0 (para función 62)</p>	<p>En el buzón BMAC se espera el mensaje enviado.</p> <p>La Base de Datos en su relación Puerto se modifica en el <i>sts_puerto</i> (debe tener el bit 1 = 0).</p> <p>El <i>destino</i> debe tener el valor de 49.</p>

DATOS DE ENTRADA AL BUZÓN BRES	SALIDAS ESPERADAS
<p>Se envía el encabezado del mensaje y 2 datos adicionales</p> <p>longit : 10 func : 3, 7, 29, 31, 33, 42, 48 resu : 1 n_msje : 1 destino : 1 datos : 10, 20</p>	<p>Se recibe el mensaje enviado en los buzones BMAC y BEST (Esc. a Maestra y Estadísticas)</p> <p>El <i>destino</i> tiene el valor de 49 en los dos buzones de recepción.</p> <p>Se pueden enviar la cantidad de datos adicionales al encabezado que se desee y se deben recibir exactamente como se enviarón</p>
<p>Se envía el encabezado del mensaje al buzón</p> <p>longit : 8 función : 67, 68 resu : 1 n_msje : 1 destino : 1</p>	<p>Se recibe el mensaje en los buzones BEST y BMAC tal y como se enviarón.</p> <p>El <i>destino</i> tiene el valor de 49 en ambos mensajes recibidos</p>
<p>Se envía el siguiente mensaje al buzón</p> <p>longit : 10 función : 1 resu : 32, 5 n_msje : 1 destino : 1 datos : 01 /* utr */ 03 /* n_grupo */</p>	<p>Se recibe el mensaje en el buzón BEST con 49 en <i>destino</i>.</p> <p>Modifica la relación ADQ_GPOS en su campo GPO_ACT (00 o FF) y envía el mensaje a BMAC por referencia o como : (dependiendo del estado de la BDD)</p> <p>longit : 11 (LON_FALLA_EXP) func : 21 (FUN_FALLA_EXP) resu : 0 (NO) n_msje : 0 destino : 48 (DESMAC) datos : 01 /* utr */ 03 /* n_gpo */ ?? /* gpo_act */</p>

DATOS DE ENTRADA AL BUZÓN BRES	SALIDAS ESPERADAS
longitud : 10 función : 1 resu : 1 n_msje : 1 destino : 1 datos : 01 /* utr */ 01 /* t_barr = LENTO */	Modifica en la relación UTR el <i>t_barr</i> , en la relación ESTADISTICAS actualiza el tiempo en que la UTR ha estado en determinado barrido. Envía el mensaje al buzón de <i>escr.</i> a central (BMAC) con el <i>destino</i> con un valor de 49
longitud : 9 función : 24, 25 resu : 1 n_msje : 1 destino : 1 datos : 01 /* utr */	Obtiene el tipo de barrido y actualiza en la relación ESTADISTICAS el tiempo en que la UTR ha estado en ese barrido. En la relación UTR actualiza el <i>sts_utr</i> . Envía el mensaje al buzón BMAC con valor de 49 para <i>destino</i> .
longitud : 10 función : 26, 27 resu : 1 n_msje : 1 destino : 1 datos : 01 /* utr */ 03 /* n_gpo */	Actualiza en ADQ_GPOS el campo <i>sts_adq</i> : 0 para función 26 1 para función 27 Envía el mensaje al buzón BMAC cambiando la variable <i>destino</i> a 49
longitud : 9 función : 34, 35 resu : 1 n_msje : 1 destino : 1 datos : 01 /* puerto */	Encuentra el tipo de barrido y actualiza en la relación ESTADISTICAS el tiempo en que la UTR ha estado en ese tipo de barrido. Actualiza en la relación PUERTO el campo <i>sts_prio</i> de acuerdo a la función que se trate: 0 para función 34 1 para función 35 Envía el mensaje al buzón BMAC cambiando la variable <i>destino</i> a 49

4.3.2. Proceso estadísticas

DATOS DE ENTRADA AL BUZÓN BEST	SALIDAS ESPERADAS
longitud : 10 función : 7, 29, 31, 33, 42, 48 resu : 40 n_msj : 1 destino : 1 datos : 01 /* utr */ 01 /* t_barr */	Incrementa en la relación ESTADISTICAS los siguientes datos: INT_COM_UTR EXI_COM_UTR Obtiene el tipo de barrido y arma un nuevo mensaje con los siguientes datos: longit : LON_BAR_UTR func : FUN_BAR_UTR resu : NO n_msje : 0 destino : destino * 10 t_barr : BAR_NORMAL Envía el mensaje al buzón BPRT
longitud : 10 función : 68 resu : 40 n_msje : 1 destino : 1 utr : 02 puerto : 02	Incrementa en la relación ESTADISTICAS los siguientes datos: INT_COM_UTR EXI_COM_UTR
longitud : 10 función : 67 resu : 40 n_msje : 1 destino : 1 utr : 02 puerto : 02	De la relación ESTADISTICAS incrementa y actualiza los siguientes campos: INT_EXP EXI_EXP
longitud : 10 función : 1 resu : 1 n_msje : 1 destino : 1 utr : 01	De la relación ESTADISTICAS incrementa y actualiza los siguientes campos: INT_EXP FAL_EXP Realiza el procedimiento <i>estadísticas de comunicaciones (estcom)</i>
longitud : 10 función : 3 resu : 40 n_msje : 1 destino : 1 utr : 02 datos : 01	De la relación ESTADISTICAS incrementa y actualiza los siguientes campos: INT_CTR FAL_CTR CON_ERR_COM

4.3.3. Proceso distribuidor

DATOS DE ENTRADA AL BUZON BDIS	SALIDAS ESPERADAS
longitud : 10 función : 3, 24, 25, 26, 27, 28, 29, 31, 33, 42, 48 resu : 1 n_msje : 1 destino : 1 datos : 01 /* utr */ 02 /* adicional */	Si el protocolo del puerto obtenido con el valor de <i>utr</i> en el mensaje es mayor o igual que PCS y menor o igual que MSTRRA envía el mensaje al buzón BPRT.
longitud : 10 función : 23, 30, 34, 35, 36, 37, 38, 39 resu : 1 n_msje : 1 destino : 1 datos : 02 /* puerto */ 02 /* adicional */	Obtiene el protocolo del puerto indicado en el mensaje y si es mayor o igual que PCS y menor o igual que MSTRRA envía el mensaje a BPRT cambiando el valor de destino a 34.
longitud : 10 función : 11, 12, 13, 14, 15, 16, 62, 63, 64, 65, 66, 69, 70, 71 resu : 1 n_msje : 1 destino : 1 datos : 02 /* puerto */ 02 /* adicional */	Envía el mensaje al buzón BPRT y realiza el reset físico al puerto enviándole un pulso al registro de identificación del puerto.

4.3.4. Sistema integrado

Los casos de prueba mencionados anteriormente para cada uno de los procesos pueden utilizarse para conjuntar una ruta de prueba para el sistema completo. Se muestra a continuación las rutas propuestas de pruebas.

- Caso 1 (Comunicación de tarjeta de enlace a la EM a módulo maestro del CIC)

Se probará la comunicación entre la tarjeta PC-851 que sirve de enlace a la Estación Maestra y el módulo maestro del CIC-PC a través de los procesos LECT1, ESCR1 y DISTRIBUIDOR. En esta prueba se enviará un mensaje desde la terminal conectada a la PC-851, de manera que pase por los procesos anteriormente mencionados y regrese a la tarjeta después de haberse procesado los cambios en el mensaje (ver figura 4.2).

Los mensajes a enviarse para realizar esta prueba son los siguientes:

ENTRADA	SALIDA
longit : 8 func : 54 Diagnostico CIC-PC resu : 0 n_msje : 0 destino : 1	Se recibe el mensaje enviado sin ningún cambio en sus datos.
longit : 9 func : 50, 52, 53 resu : 0 n_msje : 1 destino : 1 utr : 1 Solicitud de estadísticas	Se recibe el encabezado del mensaje por referencia longit : 13 LON_MSJE_XREF func : 0xFFE0 MSJE_XREF resu : 0 NO n_msje : 0 destino : 1 tipo : 1 ptr_mem : XXXX
longit : 9 func : 51 resu : 0 n_msje : 1 destino : 1 utr : 1	Se recibe el mismo mensaje que se envió y además todos los valores de la relación ESTADIST se ponen en cero

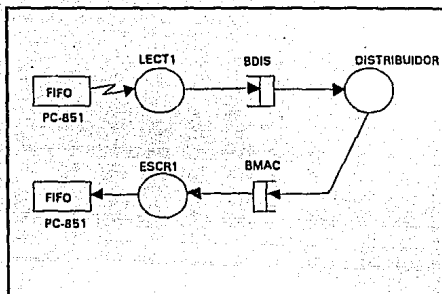


FIGURA 4.2 Ruta sugerida para prueba del sistema (caso 1)

-Caso 2 (Comunicación Estación Maestra - UTR)

Se probará en este punto la comunicación entre la tarjeta que sirve de enlace a la Estación Maestra y la tarjeta de enlace hacia las UTR, suponiendo que la de enlace hacia la Estación Maestra es la que envía el mensaje y la otra lo recibe (ver figura 4.3). Los mensajes seleccionados para esta prueba son :

ENTRADA	SALIDA
longit : 11 func : 23, 30, 34, 35, 36, 37, 38, 39, 41, 71 resu : 1 n_msje : 1 destino : 1 puerto : 2 datos : 1, 2	Para todos los mensajes se espera que los datos enviados no se modifiquen, todas estas funciones pertenecen a la función <i>puertos_func</i>
longit : 10 func : 11, 12, 13, 14, 15, 16, 62, 63, 64, 65, 66, 69, 70 resu : 0 n_msje : 1 destino : 1 puerto : 1 datos : 1	Se recibe el mensaje exactamente como se envío, la única diferencia es que pasa por la función <i>ini_puerto</i> . Esta referencia se da por si se desea ver los efectos que tiene en la base de datos alguna de las funciones indicadas

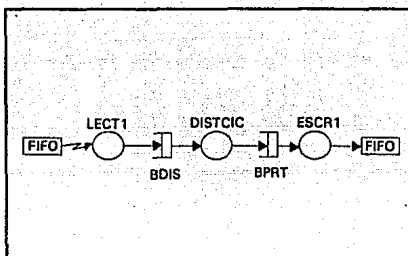


FIGURA 4.3 Ruta sugerida para prueba (caso 2)

- Caso 3 (Comunicación UTR - Estación Maestra)

Esta prueba es muy similar a la anterior, solo que ahora la tarjeta de enlace hacia las UTR es la que envía el mensaje y la de enlace hacia la Estación Maestra es la que lo recibe después de pasar por los procesos correspondientes y de haber sufrido los cambios necesarios (ver figura 4.4). Los mensajes para esta prueba son:

```

longit : 10
func : 11, 12, 13, 14, 15, 16, 23, 28, 30, 36, 37,
38, 39, 41, 62, 63, 64, 65, 66, 69, 70, 71
resu : 1
n_msje : 1
destino : 1
puerto : 2
datos : 1
  
```

Se reciben los mensajes tal y como se enviaron, estos no son todos los casos, ya que del proceso concentrador todos los mensajes se redirigen a BMAC

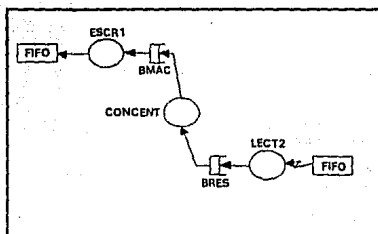


FIGURA 4.4 Ruta sugerida para pruebas (caso 3)

- Caso 4 (Estadísticas de los mensajes)

Este caso de prueba es parecido al anterior con excepción de que ahora los mensajes se dirigen al proceso de estadísticas para registrar los elementos necesarios en la relación ESTADISTICAS de la base de datos (ver figura 4.5). Los mensajes a utilizar son los siguientes.

```

longit : 9
func : 3, 7, 29, 31, 33, 42, 48
resu : 2
n_msje : 3
destino : 2
datos : 1
  
```

Se recibe el mismo mensaje en los buzones BMAC y BEST

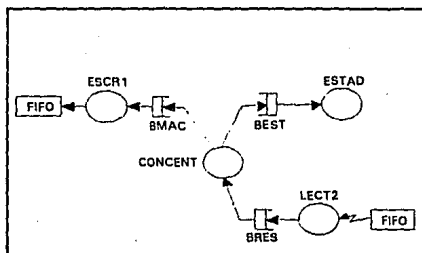


FIGURA 4.5 Ruta sugerida para prueba (caso 4)

- Caso 5 (Comunicación de la tarjeta de enlace hacia UTR con módulo maestro del CIC)

En esta última prueba se realizarán estadísticas de los mensajes enviados desde el puerto de escritura de la tarjeta de enlace a las UTR hacia el puerto de lectura de la misma tarjeta, el mensaje enviado debe coincidir con el recibido, con excepción de los datos que se modifican en el trayecto (ver figura 4.6). Los mensajes de prueba son:

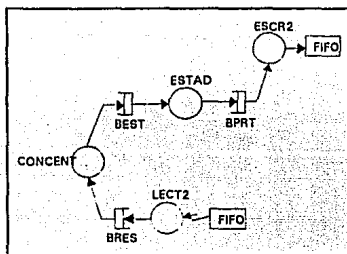


FIGURA 4.6 Ruta sugerida para prueba (caso 5)

```

longit : 9
func : 7, 29, 31, 33, 42, 48
resu : 2
n_msje : 3
destino : 2
datos : 1
  
```

Se recibe el mismo mensaje en los
buzones BMAC y BEST

4.4. Resultados de las pruebas

- Prueba del correcto funcionamiento de cada módulo

Se aplicaron dos técnicas de análisis informal para las pruebas de los módulos que forman al módulo maestro del CIC. Estas técnicas organizan las actividades de prueba mediante un razonamiento informal para la verificación de los programas.

- Inspección del código

En la inspección del código, el análisis se enfocó principalmente en descubrir algún error de sintaxis. En otras palabras el código se revisó para evitar la presencia de errores, en lugar de simular su ejecución.

Ya que muchos de los errores en los programas se pueden clasificar de acuerdo a categorías bien conocidas, se revisó el código del CIC-V2 para evitar algunos de los errores clásicos de programación, tales como:

- Uso de variables no inicializadas
- Saltos dentro de un ciclo
- Asignaciones incompatibles
- Ciclos que no terminan
- Índices de arreglos fuera de límites
- Asignación de memoria inadecuada
- Error en los parámetros formales en llamadas a procedimientos
- Etc.

En general se concentran las pruebas en aquello que no puede ser inspeccionado automáticamente por el compilador (errores de lógica).

- Revisión del código por pasos

Esta es una técnica de análisis de código informal en la que se seleccionan algunos casos de pruebas y se simula la ejecución del código en una prueba de escritorio. Las variables locales se registran en una hoja de papel y la memoria compartida en un pizarrón.

Durante el procedimiento de pruebas del código no se encontraron errores tanto en la inspección de código como en la revisión del código por pasos, el manejo de los mensajes de un procedimiento a otro es correcto; por lo que se asume que el código es aceptable.

En esta etapa se ejecutaron también todas las pruebas de los casos seleccionados previamente (descritos en la sección 4.3) y se encontró que para todos los mensajes enviados, la respuesta a los mismos es la que se esperaba desde un principio, con lo que podemos asegurar que el funcionamiento interno del modulo maestro del CIC-V2 está correcto y cumple con las especificaciones requeridas en el capítulo 3. Estas pruebas se repitieron un número de veces para verificarlas.

- Prueba del puerto de enlace hacia las Unidades Terminales Remotas

Para la realización de estas pruebas se colocó en la computadora del CIC una tarjeta PC-851 y se procede en primera instancia a realizarle pruebas de funcionamiento, y en una segunda etapa se prueba la comunicación entre la tarjeta y el modulo maestro.

La primera etapa es realizar pruebas de funcionamiento de los elementos de las tarjetas PC-851, se tenía un programa de pruebas realizado anteriormente para las tarjetas MAC-851 y dada la compatibilidad, se ejecutó este programa para diagnosticar el estado de las tarjetas. Los resultados se muestran a continuación.

<<<< PROGRAMA DE PRUEBA PARA TARJETA MAC-851 >>>>
(IIE- DEPARTAMENTO DE ELECTRONICA VERSION 1.0)

OPCIONES DE PRUEBA

- (0) PRUEBA DE CANAL SERIE BIT-BIT
- (1) PRUEBA DE MEMORIA RAM
- (2) PRUEBA DE MEMORIAS FIFO
- (3) PRUEBA DE INDICADORES
- (4) PRUEBA DEL CIRCUITO NSC-810A
- (5) PRUEBA DEL DUART MC-2881
- (6) PRUEBA DE INTERRUPCIONES
- (7) PRUEBA DEL TIMER DUART 80C31
- (8) PRUEBA DE LA TARJETA MX-802

Los resultados obtenidos al ejecutar algunas de las opciones mostradas fueron:

*0

----> PRUEBA DE CANAL SERIE BIT - BIT

----> <<OK>>

----> oprima ESPACIO para continuar

*1

----> PRUEBA DE MEMORIA RAM

----> MEMORIA RAM <<OK>>

----> Oprima ESPACIO para continuar

*2

----> PRUEBA DE MEMORIA FIFO DE SALIDA

----> MEMORIA FIFO DE SALIDA <<OK>>

----> oprima ESPACIO para CONTINUAR

----> PRUEBA DE MEMORIA FIFO DE ENTRADA

Escribir en la memoria FIFO de entrada desde la tarjeta MAESTRA hasta 512 bytes.

----> oprima ESPACIO para CONTINUAR

----> LEYENDO MEMORIA FIFO DE ENTRADA

01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17

----> oprima ESPACIO para CONTINUAR

*4

***** <<< PRUEBA DEL CIRCUITO NSC810-A >>> *****

----> MEMORIA RAM <<< OK >>>

----> <<< PRUEBA DE PUERTO B >>>

opríma ESPACIO para leer DIPSWITCH

opríma RETURN para continuar

1F_

*5

*****> PRUEBA DE DUART

(1)---->PRUEBA INTERNA

(2)---->CANALES RS-485

(3)---->CANALES RS-232

*1

PRUEBA INTERNA DEL DUART

TXA/RXA ----> <<OK>>

TXB/RXB ----> <<OK>>

*3

PRUEBA DE LINEAS RS-232

TXDA/RXDA ----><<OK>>

TXDB/RXDB ----><<OK>>

RTSA/CTSA ----><<OK>>

RTSB/CTSB ----><<OK>>

DTRA/DSRA ----><<OK>>

DTRB/DSRB ----><<OK>>

Para realizar las pruebas de enlace del modulo maestro con la tarjeta PC-851 se le coloca a la tarjeta una memoria EPROM conteniendo el protocolo COM-832 utilizado para comunicarse con una UTR tipo DTE-532 DL y se le envía una base de datos con la información de dos UTR y se espera su respuesta.

La secuencia a seguir para el cargado de la base de datos es la siguiente:

1. Envía los datos para el cargado de líneas, modem y canal. En este punto se configura 1 línea, 1 modem funcionando a 300 baud con un tiempo de espera por CTS de 60 seg, y 1 canal con tiempo de inversion de Rx-Tx de 250 ms, con Tx-Rx de 210 ms.
2. Se envían dos mensajes, cada uno de ellos conteniendo la información referente a una UTR la cual indica su número, estado, número de línea de comunicación, dirección, velocidad de comunicación, grupos, estado inicial de cada grupo y los puntos que maneja
3. Como tercer punto se envían las imágenes para cada uno de los puntos definidos en la UTR, se indica el número de punto y el estado o imagen.
4. Por último se le envía la información para el cargado de grupos DTE indicando las características del grupo y se envían las funciones de exploración, las cuales contienen los tiempos programados de exploración de cada UTR.

Una vez realizado el cargado de la base de datos se efectúan las siguientes operaciones en el puerto:

1. Cambiar el periodo de exploración a 10 segundos
2. Meter el grupo en exploración
3. Meter UTR en exploración
4. Poner el puerto en modo normal (empieza a comunicarse con la UTR)

A continuación se muestra la relación de mensajes enviados y la respuesta obtenida para cada uno de ellos.

ENVIO DE LA BASE DE DATOS	
MENSAJE	RESPUESTA
<p>1. Cargado Línea, Modem Canal</p> <p>longitud: 33 función: 13 resultado: 0 n_msje: 0 destino: 144 (puerto 09) 09 01 01 01 01 01 01 01 01 00 02 3C 00 01 01 00 00 FA 00 D2 00 D7 00 00</p>	<p>longitud: 9 función: 13 resultado: 0 n_msje: 90 destino: 48 09</p> <p>Indica que la operación se llevo a cabo con éxito en el puerto 09.</p>
<p>2. Cargado UTR-1</p> <p>longitud: 46 función: 11 resultado: 0 n_msje: 0 destino: 144 09 01 01 00 01 01 02 01 00 01 00 00 00 00 00 00 01 00 01 00 00 00 07 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00</p>	<p>longitud: 10 función: 11 resultado: 0 n_msje: 0 destino: 48 09 01</p> <p>El resultado igual a cero indica operación exitosa.</p>
<p>3. Cargado UTR-2</p> <p>longitud: 46 función: 11 resultado: 0 n_msje: 0 destino: 144 09 06 01 00 01 06 02 01 00 01 00 00 00 00 00 00 01 00 02 00 00 00 07 00 08 00 09 00 0A 00 0B 00 0C 00 0D 00 0E 00</p>	<p>longitud: 10 función: 11 resultado: 0 n_msje: 0 destino: 48 09 06</p> <p>Operación exitosa</p>
<p>4. Carga imagenes digitales UTR-1</p> <p>longitud: 40 función: 14 resultado: 0 n_msje: 0 destino: 144 09 01 07 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00</p>	<p>longitud: 10 función: 14 resultado: 0 n_msje: 0 destino: 48</p> <p>Operación exitosa en UTR 1</p>

<p>5. Carga imagenes digitales UTR-2</p> <p>longitud: 40 función: 14 resultado: 0 n_msje: 0 destino: 144 09 06 07 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00 0C 00 00 00 0D 00 00 00 0E 00 00 00</p>	<p>longitud: 10 función: 14 resultado: 0 n_msje: 0 destino: 48 09 06</p> <p>Operación exitosa en UTR 2</p>
<p>6. Cargado de Mapas para grupos DTE en UTR 1</p> <p>longitud: 16 función: 15 resultado: 0 n_msje: 0 destino: 144 09 01 01 00 00 07 01 07</p>	<p>longitud: 11 función: 15 resultado: 0 n_msje: 0 destino: 48 09 01 01</p> <p>Operación exitosa en UTR 1</p>
<p>7. Cargado de Mapas para grupos DTE en UTR 2</p> <p>longitud: 16 función: 15 resultado: 0 n_msje: 0 destino: 144 09 06 01 00 00 07 01 07</p>	<p>longitud: 11 función: 15 resultado: 0 n_msje: 0 destino: 48 09 06 01</p> <p>Operación exitosa en UTR 2</p>
<p>8. Funciones de exploración</p> <p>longitud: 22 función: 12 resultado: 0 n_msje: 0 destino: 144 09 02 00 00 00 30 01 00 01 00 00 30 06 00</p>	<p>longitud: 9 función: 12 resultado: 0 n_msje: 0 destino: 48 09</p> <p>Operación exitosa</p>

COMANDOS ENVIADOS HACIA EL PUERTO

MENSAJE	RESPUESTA
<p>1. Cambio de tiempo de canal</p> <p>longitud: 18 función: 23 resultado: 0 n_msje: 0 destino: 144 09 01 10 00 10 00 D0 07 00 00</p>	<p>longitud: 10 función: 23 resultado: 0 n_msje: 0 destino: 48 09 01</p> <p>Operación exitosa</p>

<p>2. Cambio de barrido a grupo</p> <p>longitud: 13 función: 28 resultado: 0 n_msje: 0 destino: 144 06 00 00 00 10</p>	<p>longitud: 13 función: 28 resultado: 0 n_msje: 0 destino: 48</p> <p>Operación exitosa</p>
<p>3. Meter grupo a exploración</p> <p>longitud: 10 función: 26 resultado: 0 n_msje: 0 destino: 144 06 00</p>	<p>longitud: 10 función: 26 resultado: 0 n_msje: 0 destino: 48 06 00</p>
<p>4. Meter UTR a exploración</p> <p>longitud: 9 función: 24 resultado: 0 n_msje: 0 destino: 144 06</p>	<p>longitud: 9 función: 24 resultado: 0 n_msje: 0 destino: 48 06</p>
<p>5. Poner el puerto en modo normal</p> <p>longitud: 9 función: 34 resultado: 0 n_msje: 0 destino: 144 09</p>	<p>longitud: 9 función: 34 resultado: 0 n_msje: 0 destino: 48 09</p>

Después del envío del último comando el puerto mandaba mensajes periódicos cada 10 segundos indicando en un código de error para la respuesta de exploración:

"Error de CTS en la comunicación con la UTR"

Este error se generaba dado que no existía un modem conectado a la tarjeta de comunicación PC-851, de manera que el puerto se encontraba funcionando perfectamente, pero no podía establecer la comunicación hacia la UTR.

- Tiempo de respuesta

Se realizó la medición del tiempo de respuesta de los procesos principales del módulo maestro del CIC-V2 (Distribuidor, Concentrador y Estadísticas). Estos son los procesos principales y en donde se realizan las operaciones indicadas por los mensajes adquiridos por los procesos de enlace; por consiguiente, es necesario conocer su comportamiento para evaluar si su tiempo de respuesta es adecuado para la aplicación.

Se enviaron los mensajes propuestos en los casos de prueba para cada uno de estos procesos y se realizaron 12 mediciones para cada uno de los procesos. Se consideró que el número de mediciones son suficientes para los objetivos de esta prueba. Adicionalmente, se envió al proceso distribuidor los mensajes con la base de datos del puerto, pues son de los que más información manejan.

Los resultados de esta prueba se presentan en las siguientes tablas:

PROCESO CONCENTRADOR		
HORA_INICIO	HORA_TERMINACION	DIFERENCIA (seg.)
16:09:01.57	16:09:01.57	-----
16:09:27.19	16:09:28.12	0.93
16:10:14.43	16:10:15.42	0.99
16:10:32.88	16:10:33.65	0.77
16:10:49.30	16:10:50.62	1.32
16:12:27.62	16:12:28.61	0.99
16:12:50.58	16:12:51.57	0.99
16:13:10.52	16:13:11.34	0.82
16:13:34.25	16:13:35.18	0.93
16:13:47.37	16:13:47.37	-----
16:14:11.76	16:14:11.81	0.05
16:14:50.70	16:14:50.70	-----

Tiempo de respuesta promedio : 0.865 segundos

PROCESO DISTRIBUIDOR		
HORA_INICIO	HORA_TERMINACION	DIFERENCIA (seg.)
16:05:20.85	16:05:20.90	0.05
16:05:20.90	16:05:20.96	0.06
16:05:20.96	16:05:21.01	0.05
16:05:21.01	16:05:21.07	0.06

16:05:21.12	16:05:21.18	0.06
16:05:21.18	16:05:21.23	0.05
16:05:21.23	16:05:21.29	0.06
16:05:21.29	16:05:21.34	0.05
16:06:20.17	16:06:22.42	2.25
16:06:22.42	16:06:22.48	0.06
16:06:22.53	16:06:22.59	0.06
16:06:22.59	16:06:22.64	0.05

Tiempo de respuesta promedio : 0.238 seg

PROCESO ESTADISTICAS		
HORA_INICIO	HORA_TERMINACION	DIFERENCIA (seg.)
16:10:15.42	16:10:15.47	0.05
16:10:33.65	16:10:33.65	----
16:10:50.62	16:10:50.62	----
16:12:28.61	16:12:28.66	0.05
16:12:51.57	16:12:51.62	0.05
16:13:11.34	16:13:11.34	----
16:13:35.12	16:13:35.18	0.06
16:15:20.20	16:15:20.20	----
16:15:37.94	16:15:37.94	----
16:15:52.05	16:15:52.05	----
16:16:07.87	16:16:09.25	1.38

Tiempo de respuesta promedio : 0.486 segundos

Los tiempos de respuesta promedio indican que el CIC-V2 tiene la suficiente capacidad para manejar las comunicaciones hacia las UTR, ya que la exploración más rápida se realiza en un segundo.

Las diferencia de tiempo entre procesos se debe a la manera en como se asigna el procesador a cada uno de ellos, se observa que al momento de realizar las pruebas el proceso Distribuidor fue el que mayor tiempo se ejecuto, ya que las respuestas son las más rápidas a pesar de que los mensajes fueron de los que más información contenían.

- Prueba de enlace hacia la Estación Maestra

Esta prueba quedo fuera del alcance de este trabajo por no contarse en el laboratorio con todos los elementos del sistema de cómputo.

CONCLUSIONES

En este trabajo se desarrollaron los programas de un equipo cuya función es la de controlar las comunicaciones entre una Estación Maestra y diferentes Unidades Terminales Remotas de un sistema de control supervisorio. Este equipo (denominado CIC-V2) es una pieza fundamental para este sistema, ya que de él depende que se obtenga una información precisa en tiempo real del estado del sistema a supervisar, en este caso una red eléctrica de transmisión o distribución.

La constante integración de sistemas de control supervisorio en el país resulta en una mejor calidad y continuidad del suministro y distribución de energía eléctrica a todos los sectores de la población.

Las conclusiones principales obtenidas de este trabajo respecto a la funcionalidad del CIC son:

1. Flexibilidad de uso, ya que soporta protocolos de seis diferentes fabricantes de UTR, los cuales se pueden combinar en la computadora que contiene al CIC-V2 sin ningún problema, pues el CIC interpreta estos protocolos y los convierte a un formato

interno de comunicación con la EM. Otros protocolos se pueden implementar fácilmente a esta arquitectura.

2. Completa compatibilidad con la Estación Maestra MAC-5000, lo cual lo considera como una opción para el manejo de las comunicaciones en los equipos que operan en las zonas de Morelia y Toluca.
3. Facilidad para integrar desarrollos de otros fabricantes. Como este CIC se implemento en un equipo comercial, resulta menor el esfuerzo para adaptar desarrollos de otros fabricantes, tal como la comunicación hacia una red local Ethernet. Para realizar esta comunicación se requiere cambiar únicamente el programa de enlace hacia el sistema de cómputo, lo cual resultaría en tener las UTR en comunicación con una red local la cual puede ser parte de una EM distribuida basada en Estaciones de Trabajo, como la EMIX-V1 en desarrollo en el Instituto de Investigaciones Eléctricas.
4. Su arquitectura permite la adaptación de este equipo a otras aplicaciones de control supervisorio o control distribuido diferentes a las ya mencionadas, como para el control de procesos industriales, plantas generadoras de energía eléctrica, etc.
5. Resalta también el aspecto de que el CIC puede comportarse como un traductor de protocolos ("gateway") con o sin procesamiento intermedio o como un dispositivo puente ("bridge") entre dos redes ya sea iguales o diferentes.

Por otro lado, la utilización de una metodología de desarrollo contribuyó en contar con una especificación formal del módulo maestro, la cual permite la correcta descripción de cada uno de los módulos sin perder de vista el objetivo del sistema; basandose en esta especificación y diseño se realizó la codificación de tal manera que su integración resultó un proceso claro y ordenado. Esta especificación también sirve como base para futuras modificaciones o para efectuar el mantenimiento del sistema.

La utilización de una determinada metodología depende en gran parte del tipo de sistema a desarrollar, para este caso se realizó un análisis resultando que la metodología de Ward & Mellor era la más adecuada para sistemas de tiempo real.

Las pruebas internas del módulo maestro del CIC resultaron bastante satisfactorias, y aunque no se efectuaron las pruebas de enlace con el equipo real debido a que no se disponía de todos los elementos necesarios, se puede asegurar que el CIC-V2 tendrá un comportamiento correcto ya que cumple con todos los requerimientos solicitados en el capítulo de especificación y diseño. La realización de estas pruebas queda fuera de los objetivos de este trabajo, pero sigue siendo un punto importante antes de realizar su implementación e integración a la Estación Maestra.

Finalmente, esta versión del controlador de comunicaciones CIC se puede aún mejorar realizando en trabajos futuros algunos de los puntos sugeridos a continuación:

1. Se puede mejorar la interfaz hombre-máquina local para que, además de presentar el comportamiento en ese instante de la UTR, permita efectuar acciones directas sobre las UTR notificando al sistema de cómputo, en todo momento, cada una de las operaciones realizadas mediante esta interfaz local.
2. Ya que la tendencia de los equipos de control supervisorio es encaminarse a los sistemas abiertos utilizando una red local Ethernet, como en el caso de la Estación Maestra EMIX-V1, se puede realizar este enlace y se pueden adaptar los programas de una manera relativamente sencilla al sistema operativo UNIX con extensiones para tiempo real, ya que como está codificado en C estándar solo se necesitaría verificar como se efectúan las tareas en el nuevo sistema operativo.
3. Mejorar las funciones de diagnóstico local, incorporando mecanismos de detección y corrección de errores, así como funciones de reconocimiento en cuanto a la presencia y configuración de todos los puertos del sistema.
4. Se requiere de un respaldo a disco periódicamente de la base de datos para que en caso de falla del sistema se pueda arrancar a partir de un estado reciente del sistema eléctrico y evitar la generación de múltiples alarmas. También debe considerarse la posibilidad de contar con una base de datos local al CIC-V2 en un esquema de bases de datos distribuidas.
5. Incluir un procesamiento de variables más avanzado de tal forma que el sistema de cómputo solo reciba la información estrictamente necesaria. (por ejemplo, procesamiento inicial de alarmas, indicación de límites, etc).

REFERENCIAS

- [DAT 81] Date, C.J., *An introduction to database systems*, 3rd ed. Addison Wesley, USA, 1981.
- [DAZ 90] Daza, U. I., *Sistema operativo para procesos concurrentes (SOPCO-51) orientado a tiempo real*, CENIDET, tesis de maestria, 1990
- [GHE 91] Ghezzi, C., Jazayeri, M., *Fundamentals of Software Engineering*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey 1991
- [GLO 89] Glow F. "Control Systems Design, Part 5 - Communications Orchestrate Process Control", *ISA INTECH*, vol. 36, No. 9, Sep 1989.
- [GUT 90] Gutiérrez G. "Principales características de la estación maestra MAC-5000", *Boletín IIE*, Marzo/Abril 1990, vol 14, num. 2, pp. 51-57.
- [IBA 89] Ibargüengoytia, P. "Comunicaciones para controladores", *Boletín IIE*, Septiembre/Octubre 1989, pp. 196-202.
- [IEE 83] IEEE Standard, "IEEE Standard for Software Test Documentation", ANSI/IEEE Std 829, 1983

- [IEE 88] IEEE Standard, "Portable Operating System Interface for Computer Environments" (POSIX), IEEE Std 1003.1 1988.
- [IEE 91] IEEE Tutorial Course, *Fundamentals of Supervisory Systems*. IEEE Power Engineering Society, 1991.
- [JIM 90] Jiménez, H.D., "Maestro del Controlador Inteligente de Comunicaciones para sistemas de control supervisorio", IIE, Depto. de Electrónica, Reporte Interno, Abril 1990.
- [MOL 90] Molina, P. "Equipos de adquisición y control", *IEEE Tercera Reunión de Verano de Potencia, RVP'90*, Acapulco, Gro., Julio 1990, pp. 83-86.
- [ORT 91] Ortega, S. C. "Tarjeta de comunicaciones PC-851 / Manual de usuario", IIE, Depto. de Electrónica, Reporte Interno, Agosto 1991
- [PIC 92] Picasso, B. C. "Sistema de comunicaciones para control supervisorio", *IEEE Quinta Reunión de Verano de Potencia, RVP'92*, Acapulco, Gro., Julio 1992, Tomo IV, pp 5-8.
- [SAU 92] Saucedo E., Valdivia R., "Metodología propuesta para el diseño de sistemas de cómputo en tiempo real", IIE, Departamento de Electrónica, Septiembre 1992
- [SOM 88] Sommerville, I. *Ingeniería de Software*, 2da. ed. Addison Wesley Iberoamericana, 1988
- [SYS 90] System Architect "User's Guide", Popkin Software & Systems Inc., 1990
- [TOP 91] TopSpeed "User's Guide", Jensen & Partners International, 1991
- [VAL 92] Valdivia B. R, Silva J.L.. "Tendencias tecnológicas en la automatización de la distribución", *IEEE Quinta Reunión de Verano de Potencia, RVP'92*, Acapulco, Gro., Julio 1992, Tomo III, pp.1-5.
- [VAR 92] Varela, J. H. "Puerto de comunicación maestro, Protocolo COM-832 para las terminales remotas DTE-532 DL", IIE, Depto. de Electrónica / Sintelec S.A. de C.V., Marzo 1992.

Apéndice A

CARACTERISTICAS DE LAS HERRAMIENTAS DE DESARROLLO

En este apéndice se describen las características de las herramientas utilizadas para el desarrollo del módulo maestro del CIC-V2, las cuales son:

1. **System Architect.** Herramienta CASE para la especificación y diseño del módulo maestro bajo la metodología "Data Flow Ward & Mellor".
2. **Top Speed C.** Ambiente de desarrollo con compilador en lenguaje C para manejo de concurrencia.

A.1 System Architect

El paquete "System Architect" es una herramienta de CASE ("Computer Aided Software Engineering") que soporta múltiples metodologías de diseño de programas de cómputo y ayuda a realizar los trabajos de una manera ordenada y más eficiente. Se ejecuta bajo la extensión gráfica Microsoft Windows del sistema operativo MS-DOS y soporta metodologías manejadas por procesos o manejadas por datos. Las metodologías que soporta son:

- Diagrama de flujo de datos Gane & Sarson
- Diagrama de flujo de datos DeMarco/Yourdon
- Diagramas de estructura
- Diagramas de Entidad - Relación
- Diagramas de flujo
- Diagramas de Descomposición
- Diagramas de transición de estados
- Transición de estados Ward & Mellor
- Diagrama de flujo de datos Ward & Mellor

También incluye soporte para sistemas de transacción y sistemas en tiempo real. En el apéndice B se encuentran los fundamentos de los métodos más importantes; opcionalmente, se dispone de una metodología orientada a objetos utilizando la notación Grady Booch. [SYS 90]

El "System Architect" utiliza todo el conjunto de manejadores de Microsoft Windows, así como la interfaz de usuario. Esta interfaz incluye menús desplegables, cuadros de diálogo e iconos; la amplia variedad de manejadores permiten que el "System Architect" sea utilizado con una gran variedad de equipos comerciales.

Existe soporte interno de un diccionario, el cual incluye elementos de datos, expresiones de estructuras de datos, flujo de datos, almacén de datos, descripciones de procesos, banderas de datos, de control, descripciones de los módulos y descripción de los requerimientos, entre otros. Toda la información de la enciclopedia se mantiene y se

integra dentro del diccionario, el cual incluye el almacenamiento de los símbolos gráficos, definiciones y diagramas.

El Diccionario/Enciclopedia se mantiene en un formato de "arquitectura abierta", lo que permite al usuario acceder toda la información del sistema. El Diccionario/Enciclopedia se almacena en formato dBase III Plus, dando flexibilidad adicional en el uso de estos datos, los cuales se pueden utilizar para comunicarse con otras herramientas CASE o para la generación personalizada de reportes. El diccionario puede ser importado y exportado a otros productos mediante el formato de archivo CSV ("Comma Separated Value").

Existe también soporte para revisión de reglas en los diagramas de flujo de datos, incluyendo las reglas de DeMarco/Yourdon y Gane & Sarson; cuenta con utilerías para reportar errores en línea y para realizar reportes del contenido del diccionario.

Se tiene una interfaz de usuario gráfica, la cual permite reducción, ampliación, vista de la página completa y otros modos de video más. Existe soporte para múltiples tipos de letras proporcionadas por otros fabricantes. Los tamaños de las plumas, de la cabezas de flechas y otros elementos de los diagramas se pueden personalizar. No existen limitaciones prácticas en el tamaño o el número de símbolos gráficos contenidos en un diagrama. Hojas de estilos permiten al usuario personalizar símbolos con respecto a muchas de sus características.

El sistema operativo Windows ofrece un medio ambiente gráfico de multi-ventanas manejado por eventos provenientes del mouse o del teclado. El sistema operativo permite al usuario cambiar de una aplicación a otra en cualquier momento sin ninguna pérdida de datos por el cambio de contexto, además, el "System Architect" permite que el texto y gráficas sean exportadas a otros programas compatibles con el ambiente Windows.

A.2 TopSpeed C

Aunque la implementación de un sistema en un lenguaje de programación no es la fase más crítica en el diseño, es la parte que permite que los programas se puedan ejecutar; de manera que el lenguaje de programación juega un papel importante en el desarrollo. Para el desarrollo del módulo maestro del CIC-V2 se utilizó el ambiente de desarrollo "TopSpeed" con compilador en lenguaje C y manejo de concurrencia en uno de sus modelos de memoria, en el que la mayoría de las funciones utilizadas son reentrantes. [TOP 91]

Tradicionalmente, la concurrencia no se soporta por el lenguaje de programación; sino se proporciona por el sistema operativo o accedendo los recursos internos de la máquina (por medio de interrupciones). Sin embargo, las características de un lenguaje con soporte de concurrencia permite la construcción de sistemas tales como sistemas de control supervisorio en una máquina convencionalmente secuencial como lo es una computadora personal sin necesidad de contar con un sistema operativo especial para el manejo de concurrencia en tiempo real. Las rutinas se ejecutan una a la vez, de manera de que no hay realmente unidades ejecutadas concurrentemente, sin embargo, se simula la concurrencia entrelazando la ejecución de estas rutinas. Por ejemplo, la rutina C1 se ejecuta por un determinado periodo de tiempo, después del cual se transfiere el control a la rutina C2 (cambio de contexto), la cual lo transfiere a C3 después de cierto tiempo y ésta a su vez lo transfiere a C1 completándose de esta manera un ciclo de ejecución de las rutinas. Existen variantes dentro de este esquema como lo son asignarles a las rutinas una prioridad de ejecución o mantener bloqueadas las rutinas mientras se espera la señal de un semáforo, etc.

El lenguaje utilizado para el desarrollo utiliza a un *proceso* (también llamado tarea) como la unidad básica de ejecución concurrente. Un proceso se puede ejecutar de manera asíncrona con respecto a otro; la concurrencia se simula compartiendo el mismo procesador de acuerdo a alguna política de asignación de tiempo de CPU.

Existe un módulo en el "TopSpeed" llamado *process*, encargado de proporcionar una ejecución de multi-tareas dentro de un módulo principal, bajo los sistemas operativos OS2 y MS-DOS. A cada tarea se le asigna un espacio de trabajo dentro de la pila y una

prioridad. El espacio de trabajo debe ser lo suficientemente grande para contener los requerimientos de pila de la tarea. Se recomienda un mínimo de 2000 bytes.

El módulo principal siempre tiene una tarea, *tarea_1*, que es el propio proceso cuando se ejecuta el programa. Las otras tareas se pueden iniciar con una llamada a la función que se encarga de colocarla en la cola de procesos esperando a ser activados. Antes de ejecutar las nuevas tareas, el programador de tareas se debe de inicializar con una llamada a la función que lo activa.

Las tareas se pueden comunicar mediante señales, implementadas como semáforos. Una tarea puede detener su ejecución hasta que se le envíe una señal o puede enviar una señal a cualquier tarea que este esperando.

Las tareas pueden proteger código no-reentrante u objetos de datos globales colocándole un candado a esa sección de código. Los candados pueden anidarse, pero se debe tener la llamada a su apertura correspondiente para asegurar que no ocurran tiempos muertos. Bajo el sistema operativo MS-DOS las llamadas a las funciones que llamen a rutinas del sistema o al BIOS deben protegerse con candados.

Se asume que todos los recursos tales como datos de E/S en archivos, E/S por consola y asignación de memoria son reentrantes y pueden ser utilizados sin restricción en múltiples tareas. La biblioteca de funciones del "TopSpeed" incluye funciones para realizar lo siguiente:

- Saber si un proceso esta en espera de una señal
- Generar un retraso de tiempo
- Conocer el identificador de un proceso
- Saber el número de tarea que se esta ejecutando
- Inicializar una señal de semáforo
- Colocar un candado a una sección de código
- Enviar una señal a un proceso
- Crear un nuevo proceso
- Inicializar el programador de tiempos
- Terminar el proceso actual
- Finalizar al programador de tiempos

- Abrir un candado
- Esperar una señal

A continuación se muestra un ejemplo de las funciones para el manejo de tareas en el "TopSpeed".

```

/* se le asigna la suficiente memoria a la pila para los procesos */
#pragma data(stack_size => 20000)

#include <stdio.h>      /* se incluyen los procedimientos de E/S estándar */
#include <conio.h>      /* se incluye el manejo de E/S por la consola */
#include <process.h>    /* se incluye las bibliotecas de manejo de concurrencia */

SIGNAL exit_signal;   /* inicializa el semáforo que utilizaremos como señal de
                       terminación */

/* declaración formal de la tareas que se ejecutarán concurrentemente */

void tarea1(void);
void tarea2(void);

/* Esta función repite la cuenta de 0-1000 y la presenta en las coordenadas (10, 10) hasta que
no se presione alguna tecla, posteriormente envia la señal de terminación */

void tarea1(void)
{
    int contador;
    while(!kbhit())
    {
        for(contador=0;contador<=1000;contador++)
        {
            Lock();          /* evita que haya un cambio de contexto (coloca un candado) */
            gotoxy(10,10);
            printf(" Contador interno de la tarea 1 : %d",contador);
            Unlock();        /* abre al candado para que otra tarea pueda activarse */
        }
    }
    SEND(exit_signal);      /* envia una señal al proceso que este aguardando por ella */
}

```

```
/* la tarea 2 ejecuta en un ciclo infinito la cuenta interna de 0 -1000 y la presenta en pantalla */  
  
void tarea2(void)  
{  
    int contador2;  
    while(1)  
    {  
        for(contador2=0;contador2<=1000;contador2++)  
        {  
            Lock(); /* mantiene el tiempo del CPU en esta tarea (coloca candado) */  
            gotoxy(20,20);  
            printf(" Contador interno de la tarea 2 : %d",contador2);  
            Unlock(); /* quita el candado para permitir que se asigne el CPU a  
                       otras tareas */  
        }  
    }  
}  
  
main()  
{  
    Init(&exit_signal); /* Inicializa la señal de semáforo utilizada  
                        para indicar la terminación del programa */  
    StartScheduler(); /* inicializa el programador de tiempos */  
    StartProcess(tarea1, 2000, 1); /* coloca a la tarea 1 en la cola de procesos  
                                  listos para activarse */  
    StartProcess(tarea2, 2000, 1); /* coloca a la tarea 2 en la cola de procesos  
                                  listos para activarse */  
    WAIT(exit_signal); /* bloquea al proceso hasta que llegue la  
                       señal de terminación y permite que tareas de  
                       menor prioridad se ejecuten (tarea 1 y 2) */  
    StopScheduler(); /* finaliza al programador de tiempos.*/  
}
```


Apéndice B

METODOLOGIA DE DESARROLLO

Se presenta un breve análisis de las metodologías más comunes para el diseño de sistemas de cómputo. Como resultado de este análisis se propone una metodología a seguir para el diseño de sistemas en tiempo real, basada en el método de diagramas de flujo de datos de Ward & Mellor y haciendo uso del paquete de CASE llamado SYSTEM ARCHITECT.
[SAU 92]

B.1. La fase de desarrollo de programas de cómputo

Una vez que se han establecido los requerimientos de los programas de cómputo, la fase de desarrollo comprende tres pasos distintos: diseño, codificación de código (ya sea manual o automático) y prueba. En cada paso se transforma la información de tal manera que el resultado final es un programa de cómputo probado y validado.

El flujo de información durante la fase de desarrollo se ilustra en la figura B.1.

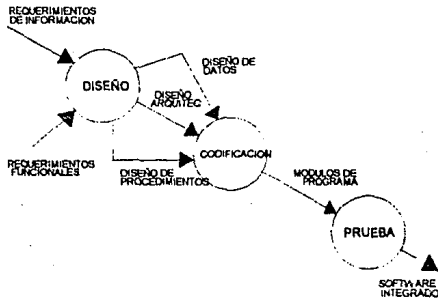


FIGURA B.1. Fase de desarrollo

El diseño de programas de cómputo se caracteriza por el dominio de la información y requerimientos funcionales del problema, los cuales se conducen utilizando una metodología adecuada que comprenda el diseño de datos, diseño de arquitectura, y diseño de procedimientos. El diseño de datos se enfoca a la definición de las estructuras de datos. El diseño de arquitectura define la relación entre los principales elementos estructurales del programa. El diseño de procedimientos transforma los elementos estructurales en una descripción del procedimiento de programas de cómputo. En este punto el código fuente se genera y se prueba para tener programas de cómputo integrados y validados.

La fase de desarrollo absorbe el 75 por ciento o más del costo de la ingeniería de software [PRE 88] (excluyendo el mantenimiento). Es aquí donde se hacen decisiones que afectarán finalmente el desarrollo de la implementación de programas de cómputo, e igualmente importante, la facilidad con la cual el programa será mantenido. Esas decisiones se hacen durante el diseño, haciéndolo un paso pivotal en la fase de desarrollo.

El diseño de programas de cómputo repercute en la calidad del producto, ya que es parte fundamental de las fases de desarrollo y mantenimiento.

El diseño de programas de cómputo es un proceso en el cual los requerimientos son trasladados en una representación gráfica del problema. Inicialmente la representación describe una vista amplia del programa. Las refinaciones subsecuentes llevan a una representación del diseño que es muy cercana al código fuente.

Desde un punto de vista de administración de proyecto, el diseño se conduce en dos pasos. (a) *Diseño preliminar*, al cual concierne la transformación de los requerimientos en arquitecturas de datos y de programas de cómputo y (b) *Diseño detallado* el cual se enfoca en refinamientos de la representación que lleven a estructuras de datos detalladas y representaciones algorítmicas para los programas.

B.2. Metodologías de diseño de programas de cómputo

Los modelos más utilizados para el desarrollo de programas de cómputo son:

- a) Modelado de procesos (Diagrama de Flujo de Datos).
- b) Modelo Entidad - Relación.
- c) Diagramas de estructura.
- d) Diagramas de Descomposición.
- e) Sistemas de Tiempo Real.

Cada una de estos modelos se describen brevemente a continuación.

B.2.1. Modelado de procesos (Diagrama de Flujo de Datos)

El modelado de procesos describe las características funcionales de un sistema. Estas funciones se representan como transformaciones de información, dibujadas como círculos. Una o más entradas, mostradas como flechas etiquetadas, manejan la transformación para producir información de salida. Este modelo puede ser aplicado al sistema completo o solo al programa, la clave es representar la información alimentada a/y producida por la transformación.

Un diagrama de flujo de datos (DFD) es una técnica que representa el flujo de información y las transformaciones que se aplican mientras los datos se mueven de la entrada a la salida. El DFD se puede usar para representar un sistema o programa de cómputo a cualquier nivel de abstracción. De hecho, los DFD se pueden particionar en niveles que representan el flujo de información y detalles funcionales.

Los estilos de los DFD varían en la forma de presentación pero tienen las mismas facilidades, por ejemplo, los DFD de Yourdon/DeMarco usan círculos como símbolos de procesos y flechas arqueadas como flujo de datos. Mientras el estilo Gane & Sarson usa un rectángulo con las esquinas redondeadas como símbolo de proceso y una notación adicional de etiquetas.

Este método es simple y se puede presentar en varios niveles por lo que se facilita el detallado del sistema, pudiéndose llegar al grado de detalle deseado.

B.2.2. Modelo Entidad - Relación

Esta es una técnica para representar las entidades y sus relaciones asociadas, esencialmente los elementos estáticos de un sistema. Puede usarse como complemento del modelado de procesos para la definición de las bases de datos.

Una base de datos contiene entidades de información que están relacionadas por medio de organización y asociación. La arquitectura lógica de una base de datos se define por un esquema que representa las definiciones de relaciones con las entidades de información. La arquitectura física de la base de datos depende de la configuración del

equipo. Sin embargo, el esquema (descripción lógica) y la organización (descripción física) deben conjuntarse para satisfacer los requerimientos funcionales para acceso, análisis y reportes.

La normalización de los archivos de datos y los análisis subsecuentes de la capacidad de la base de datos se deben acoplar con una técnica efectiva para representar las relaciones entre objetos de datos. El diagrama de Entidad -Relación y sus subsecuentes extensiones se usan frecuentemente como una herramienta gráfica para representar las relaciones entre los datos.

B.2.3. Diagramas de estructura

El diseño estructurado usa una herramienta gráfica, llamada diagrama de estructura, para hacer entendible fácilmente el sistema. El diagrama de estructura es el resultado de mostrar el sistema particionándolo en módulos, y construyendo el diagrama con esos símbolos para mostrar jerarquía, organización y comunicación entre módulos. En lugar de representar la jerarquía por niveles (añadiendo un nuevo diagrama a cada símbolo, como en el modelado de procesos), la jerarquía en el diagrama de estructura forma un árbol de módulo representado en un diagrama de una página.

La conexión entre los módulos son "llamadas", y para cada módulo existe una descripción/especificación que puede definir rigurosamente la función interna del módulo.

El diagrama de estructura (y la documentación que lo acompaña en el diccionario de datos) sirve como guía a la implementación y como herramienta de documentación. El diagrama de estructura puede guiar la asignación de tareas de codificación, y el orden en el cual son escritas y probadas.

B.2.4. Diagramas de descomposición

Esta metodología permite un análisis por medio de diagramas de organización (organigramas). El uso de esta técnica no ofrece mucha ventaja al desarrollar un sistema en tiempo real.

B.2.5. Sistemas de tiempo real

Es una extensión del modelado de procesos con adiciones y variaciones a la notación de los símbolos. Utiliza los diagramas de transición de estado para definir un tipo especial de proceso llamado "transformación de control"; tiene dos estilos; el primero usa círculos como estados y flechas como transiciones, mientras que el segundo es el estilo Ward & Mellor que es un poco más riguroso y completo que el anterior (posteriormente se describirá esta metodología). Asimismo se pueden crear más niveles de DFD como en el modelado de procesos.

B.3. Areas de aplicación del análisis orientado al flujo de datos

Cada metodología de diseño de programas de cómputo tiene fortalezas y debilidades. Un factor de selección importante para un método de diseño es la cantidad de aplicaciones en las cuales puedes ser utilizado. El diseño orientado al flujo de datos trata un amplio rango de áreas de aplicación. De hecho, como todo programa de cómputo puede ser representado por un diagrama de flujo de datos, un método de diseño que haga uso del diagrama se puede aplicar teóricamente en cualquier desarrollo.

Una técnica orientada al flujo de datos para diseño es particularmente útil cuando la información se procesa secuencialmente y no existe una estructura de datos jerárquica formal. Por ejemplo, aplicaciones de control por microprocesador, procedimientos complejos de análisis numérico, control de procesos, y muchas otras aplicaciones de programas de cómputo científico e ingeniería. Extensiones del diseño orientado a flujo de datos, adaptan la técnica al tiempo real y a aplicaciones manejadas por interrupción.

En la sección siguiente se describe en detalle este método de diseño orientado al flujo de datos.

B.4. Método de análisis orientado al flujo de datos

La información se transforma a medida que fluye a través de un sistema. El sistema acepta entradas en una gran variedad de formas; utiliza el equipo, los programas de cómputo y los elementos humanos para transformar las entradas en salidas y producirlas en una gran variedad de formas. La entrada puede ser una señal de control transmitida por un transductor, una serie de números teclados por un operador, un paquete de información transmitida en un enlace de red o datos de archivos voluminosos recuperados de almacenamiento secundario. Las transformaciones pueden comprender desde una simple comparación lógica, un algoritmo numérico complejo, o la técnica regla-inferencia de un sistema experto. La salida puede encender un simple LED o producir un reporte de 200 páginas. El modelo de flujo de datos puede ser aplicado a cualquier sistema basado en computadora sin importar su tamaño o complejidad.

Una técnica para representar el flujo de información a través de un sistema basado en computadora se muestra en la figura B.2.

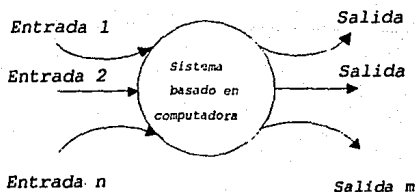


FIGURA B.2. Flujo de información

La función total del sistema se representa como una transformación de información, dibujada como una burbuja en la figura. Una o más entradas, mostradas como flechas etiquetadas, manejan la transformación para producir información de salida. Debe notarse que el modelo puede ser aplicado al sistema completo o sólo al programa de cómputo. La clave es representar la información alimentada a/y producida por la transformación.

B.4.1. Diagramas de flujo de datos (DFD)

La información se modifica por una serie de transformaciones. Un diagrama de flujo de datos (DFD) es una técnica gráfica que representa el flujo de información y las transformaciones que se aplican mientras los datos se mueven de la entrada a la salida. La forma básica de un DFD se muestra en la figura B.3.

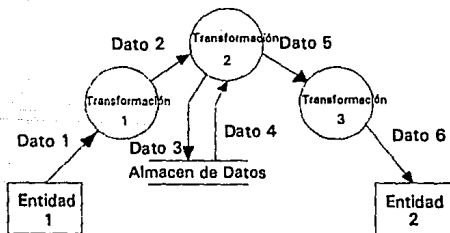


FIGURA B.3. Diagrama de flujo de datos.

El DFD se puede usar para representar un sistema a cualquier nivel de abstracción. De hecho, los DFD se pueden particionar en niveles que representan el flujo de información y detalles funcionales. El nivel 01 DFD, también llamado el modelo fundamental del sistema, representa el conjunto total del programa de cómputo como una

burbuja con una entrada y una salida de datos indicados por flechas entrando y saliendo respectivamente. Transformaciones adicionales y rutas de flujo de información se representan cuando el nivel 01 DFD se particiona para revelar más detalle.

La simbología del DFD se muestra en la figura B.4.

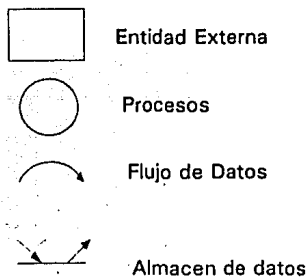


FIGURA B.4. Simbología

Un rectángulo se usa para representar una entidad externa, esto es, un elemento del sistema (ej. equipo, una persona, etc.) o algún otro sistema que produce información para su transformación por este programa de cómputo o recibe información producida por el mismo. Un círculo representa un proceso o transformación que es aplicado a los datos y los modifica de alguna manera. Una flecha representa uno o más datos. Todas las flechas en un diagrama de flujo de datos deben estar etiquetadas. La doble línea representa un almacén de datos - información almacenada que es usada por el programa. La simplicidad de la simbología DFD es una de las razones de la amplia utilización de las técnicas de análisis orientadas al flujo de datos.

Ejemplo 1:

Un ejemplo simple del DFD, considera el flujo de información para una típica llamada telefónica (ver Fig B.5(a)). El nivel 01 DFD indica que la salida es el sonido de la voz del que habla recibida por el que escucha. La entrada es la voz y el número telefónico. La figura B.5(b) ilustra el nivel 02 como refinamiento del nivel 01 DFD. En esta figura, se provee más información acerca del flujo de información y funciones del proceso (transformaciones). La acción del que llama al marcar en el teclado se transforma por la electrónica asociada a una serie de frecuencias audibles (tonos). Las frecuencias fluyen a un sistema conmutador que ejecuta el enrutamiento requerido y establece un enlace del que envía al receptor. La voz humana es transformada por un transductor de vibración que produce una señal como salida. El sistema conmutador mueve la señal de voz al receptor que la transforma de nuevo en sonido.

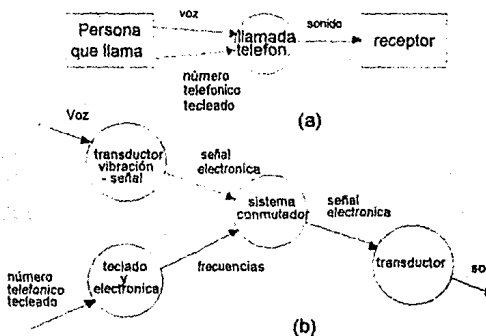


FIGURA B.5. DFD de llamada Telefónica (a) nivel 01 (b) nivel 02

Aunque el ejemplo anterior es una simplificación, el flujo de información, representado por el DFD es fácil de discernir. Cada transformación en el diagrama (los círculos) se puede refinar cada vez más para proveer mayores detalles acerca del procesamiento del teclado, transductores, o el sistema conmutador. Esto es, el diagrama se

puede desglosar para mostrar el nivel deseado de detalle. La figura B.6 ilustra este concepto.

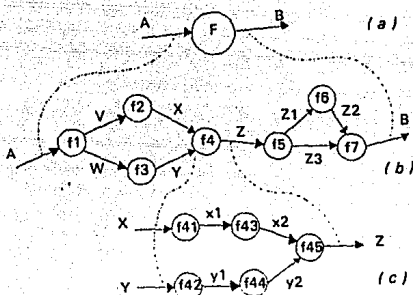


FIGURA B.6 Refinamientos del flujo de información

El modelo fundamental para el sistema F ilustrado en la figura B.6(a) indica la entrada primaria como A y la última salida como B . Se refinó el modelo F en transformaciones $f1$ a $f7$, tal como se muestra en la figura B.6(b). Nótese que la continuidad del flujo de información debe ser mantenida; esto es, la entrada y la salida del refinamiento debe ser la misma. Una refinación posterior de $f4$ (ver figura B.6(c)) representa detalles en la forma de las transformaciones $f41$ a $f45$. De nuevo, la entrada (x,y) y la salida (z) permanecen sin cambio.

Es importante notar que no hay indicación explícita de la secuencia de eventos (p.e., si el marcado se hace antes o después de la entrada de voz) dada por el diagrama. El procedimiento o la secuencia puede estar implícito en el diagrama.

El DFD es una herramienta gráfica que puede ser muy valiosa durante el análisis de requerimientos de programas de cómputo. Sin embargo, el diagrama puede causar confusión si sus funciones se confunden con el diagrama de flujo. Un DFD describe el flujo de información sin una notación explícita de control (ej. condiciones o ciclos).

B.4.2. Extensiones del modelo de flujo de datos

Los métodos de diseño y análisis orientados al flujo de datos se han extendido para el análisis y diseño con una notación apropiada de sistemas en tiempo real. Aunque se han desarrollado varios esquemas basados en el modelo de flujo de datos, la técnica propuesta por Ward & Mellor está ganando gran aceptación, la cual se describe a continuación.

Para la extensión de la notación clásica de análisis y diseño orientados al flujo de datos y para acomodarse a las demandas impuestas por un sistema en tiempo real, se debe desarrollar una notación para representar lo siguiente:

- a) Flujo de información que es adquirido o producido en tiempo continuo.
- b) Control de información que pasa a través del sistema y el procesamiento asociado de control.
- c) Estados del sistema y el mecanismo que causa transición entre estados.

En un porcentaje significativo de aplicaciones de tiempo real, el sistema debe monitorear información continua en el tiempo generada por algún proceso de tiempo real. Por ejemplo, un sistema de monitoreo de pruebas en tiempo real para una turbina de gas puede requerir monitorear la velocidad de la turbina, la temperatura del combustor, y una variedad de puntos de presión en una base continua. La notación convencional de flujo de datos no hace distinción entre datos discretos y datos continuos en el tiempo. Esto es, un examen del DFD no provee indicación de la continuidad en el tiempo del flujo de datos. Una extensión a la notación convencional de flujo de datos, mostrado en la figura B.7, provee un mecanismo para representar flujo de datos continuos en el tiempo.

La flecha con doble cabeza se utiliza para representar flujo continuo mientras una flecha sencilla se usa para indicar flujo de datos discretos. En esta figura, la temperatura se mide continuamente mientras que se provee un solo valor para el punto de ajuste de temperatura. El proceso mostrado produce una salida continua en el tiempo, *valor de corrección de nivel*.

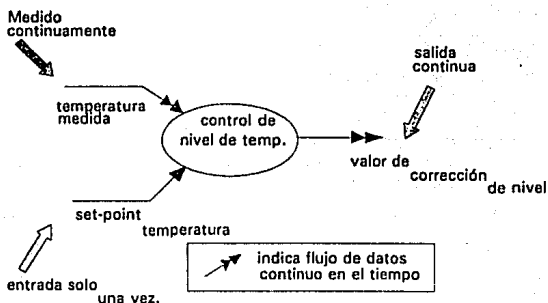


FIGURA B.7. Ejemplo de notación Ward & Mellor

Cuando el modelo físico o de implementación es creado, el diseñador debe establecer un mecanismo para la colección de datos continuos en el tiempo. Obviamente, el sistema digital adquiere datos en una manera cuasi-continua usando técnicas como muestreo cíclico ("polling") a alta velocidad. La notación indica donde se requerirá equipo convertidor analógico-a-digital y cuales transformaciones son las que demandarán programas de cómputo de alta funcionalidad.

En representaciones convencionales orientadas al flujo de datos, el control o flujo de eventos no se representa explícitamente. Esta exclusión es restrictiva cuando se consideran aplicaciones de tiempo real y por esta razón se ha desarrollado una notación especializada para representar flujo de eventos y procesamiento de control. Continuando con la convención establecida por los diagramas de flujo de datos, el flujo de datos se representa con una flecha sólida. El flujo de control, sin embargo, se representa con una flecha punteada o sombreada.

Refiriéndose al ejemplo de la figura B.8, un proceso de transformación llamado *visualiza velocidad* acepta un valor discreto, *valor de velocidad*, y produce otra salida discreta, *velocidad mostrada*. *Visualiza velocidad* también recibe un flujo de evento, *ajuste sistema de medición*, el cual controla la utilización de un algoritmo de conversión inglés o métrico dentro de la transformación. En muchas aplicaciones de tiempo real, un

porcentaje significativo de todo el flujo de información se puede asociar con control y procesamiento de eventos.

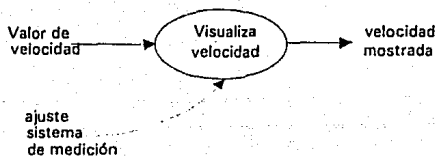


FIGURA B.8. Flujo de eventos

La figura B.9 ilustra el nivel más alto de un control de procesos para una celda de manufactura. Como los componentes se colocan en accesorios, se coloca un bit de status dentro de un registro con status de las partes que indica la presencia o ausencia de cada componente.

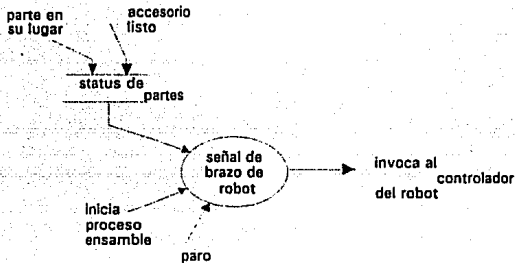


FIGURA B.9. Flujo de eventos y transformación de control

La información de eventos contenida en el status de partes se pasa a un control de procesos, *señal de brazo de robot*, que también recibe otro flujo de eventos como se muestra en la figura B.9. La salida del control de procesos es una señal de control que invoca a un controlador del robot.

Este flujo de control se expande posteriormente para ilustrar la combinación de flujo de eventos y de datos. Tal como lo muestra la figura B.10, la salida del proceso de control descrita en el párrafo precedente se usa para disparar el sistema de control de un robot ilustrado con nivel 01 del DFD. Es importante notar que la expansión del sistema de control del robot puede resultar en la representación de flujo de eventos adicionales y control de procesos, que se pueden combinar con procesos de transformación de datos y flujo de datos convencionales.

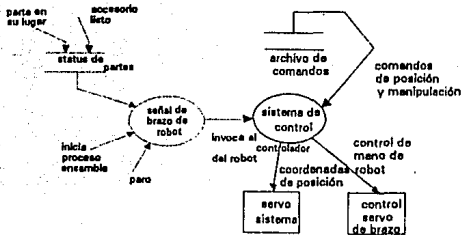


FIGURA B.10. Combinando Transformaciones de Datos y Control

En algunas situaciones pueden ocurrir varios casos del mismo proceso de control o transformación de datos en un sistema en tiempo real. Por ejemplo, un gran número de registros con status de partes se pueden monitorear de manera que diferentes robots se puedan señalar al tiempo apropiado. En suma, cada robot puede tener su propio sistema de control. La figura B.11 ilustra la notación que se puede usar para representar múltiples *casos equivalentes* del mismo proceso.

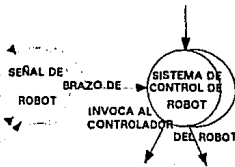


FIGURA B.11. Casos equivalentes

La representación diagramática de estados que se utiliza para extender el método orientado al flujo de datos toma la forma de la figura B.12; donde los estados del sistema se representan usando rectángulos.

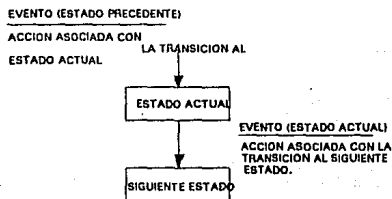


FIGURA B.12. Diagrama de transición de estados

Un beneficio importante de esta notación es que la representación diagramática de estados es menos confundible con la notación del flujo de datos. La transición de estados se representa por flechas que contienen información acerca del evento que dispara la transición y la acción producida como resultado de la transición.

B. Método propuesto para diseño de programas de cómputo

Combinando la notación convencional orientada al flujo de datos (sección B.5.1) con la extensión en tiempo real discutida en la sección B.5.2, se puede proponer un método para el diseño de sistemas en tiempo real, que consiste de los siguientes pasos.

1. Desarrollar el nivel 01 DFD (Diagrama de Flujo de Datos) que incluya todos los datos primarios y flujo de eventos, que muestre todas las fuentes de información, destinos y almacenes de datos. Para algunas aplicaciones en tiempo real, el flujo de eventos puede ser más importante que el flujo de datos en la caracterización del sistema. Por esta razón, todos los flujos de eventos deberán representarse en el nivel más alto.

2. Crear una lista de eventos que describa los principales eventos asociados con el sistema. La lista de eventos podrá incluir actividades orientadas al control tales como "activación del switch de control", o "la temperatura alcanzó el punto de ajuste" y eventos orientados a comandos como "petición del operador del status actual", o "el usuario proporciona el valor de calibración". La lista de eventos se usa para refinar el nivel 01 del modelo de flujo y como fuente de información para la creación de diagramas de estado.

3. Refinar el modelo de flujo al nivel 02 y 03 y detallar. Las recomendaciones del DFD se aplican igualmente para el refinamiento de sistemas en tiempo real. Se debe notar, sin embargo, que se refinan el flujo de datos y el flujo de eventos.

4. Crear una representación diagramática de estados. La lista de eventos puede asistir en la creación del diagrama de estados. Como en el modelo de flujo, el diagrama de estados se puede expandir a varios de niveles, con cada nivel subsecuente elaborado con la información contenida en el nivel precedente.

5. Revisar el modelo de flujo y diagramas de estado para asegurar que cada uno represente la función y flujo del sistema consistentemente. Los modelos de flujo para un sistema en tiempo real implican la estructura del diagrama de estados y, respectivamente, el diagrama de estado implica el flujo del sistema. Las representaciones para cada uno deben de ser consistentes.

6. Usar notación convencional para describir objetos de datos y contenido de procedimientos. Se puede proveer una descripción detallada de los objetos de datos con un diccionario de datos. Una narrativa de procesos para transformaciones de datos y control se puede desarrollar usando árboles de decisión o tablas de decisión.

7. La arquitectura del programa se puede derivar usando mapeo de transformación y transacción. Esos mapeos, se pueden aplicar a modelos de flujo de datos a cualquier nivel razonable de detalle. Cuando se presentan múltiples tareas o procesos, el mapeo se aplica al modelo de flujo para cada tarea o proceso individualmente.

8. Revisar todas las representaciones e iterar cuando se requiera.

B.5.1. Consideraciones en el proceso de diseño

El diseño orientado al flujo de datos permite una transición conveniente de representación de información a un diseño de la descripción de la estructura de un programa. La transición del flujo de información a la estructura se ejecuta como parte de un proceso de 5 pasos:

- 1) Se establece el tipo de flujo de información
- 2) Se indican las fronteras del flujo
- 3) Se proyecta a un programa estructurado el DFD
- 4) Se define por descomposición la jerarquía de control
- 5) Se refina la estructura resultante usando medidas de diseño y heurística.

Apéndice C

GLOSARIO DE TERMINOS

Barrido lento

Después de un cierto número de intentos de comunicación del CIC hacia la UTR, si ésta no contesta, el CIC la coloca en estado de "barrido lento", lo cual significa que la frecuencia de exploración hacia esa UTR tendrá un periodo más largo.

Barrido normal

Cuando la UTR se encuentra en este estado el CIC ejecuta la exploración a la UTR en el tiempo especificado en la base de datos, respetando el periodo que se le había asignado cuando se dio de alta.

BDD

Base de Datos. Se utiliza esta abreviación para hacer alguna referencia a la información almacenada en la estructura de datos manejada internamente por el programa.

Buzón

Un buzón es una estructura que permite el intercambio de mensajes entre procesos. Tiene asociadas dos colas en las cuales se encuentran los procesos bloqueados que esperan recibir un mensaje y los procesos que están bloqueados esperando enviar.

Cambio de contexto

Cuando el programador de tiempos decide que se debe ejecutar un proceso diferente al que está ejecutándose, se debe guardar toda la información del medio ambiente en que está trabajando y del estado de todos sus registros para que cuando le sea asignado nuevamente tiempo de CPU pueda continuar con la tarea que estaba efectuando, a esta operación se le conoce como cambio de contexto.

Canal

El canal de comunicación indica el medio de transmisión que se está empleando para comunicación con la UTR. El medio de comunicación puede ser radio, teléfono, hilo piloto, etc.

CASE

"Computer Aided Software Engineering". Se le da este nombre al conjunto de reglas que automatizan las actividades que involucra el seguir alguna metodología de las existentes en la ingeniería de software.

CD

Carrier Detect. Indica al puerto que el modem ha sido conectado con otro modem, bajo la norma RS-232C.

CIC-V2

Controlador Inteligente de Comunicaciones en computadora personal. Equipo encargado de realizar la comunicación entre una Estación Maestra de control

supervisorio y las Unidades Terminales Remotas que estan enlazadas a los elementos de campo. También se le refiere como únicamente CIC.

Concurrencia

Permite que un proceso sea ejecutado asincrónicamente con respecto a otro en un ambiente de múltiples procesadores. En el CIC-V2 la concurrencia es simulada compartiendo un único procesador de acuerdo a alguna política de asignación de procesos, ejecutando cambios de contexto.

CTS

Clear To Send. El modem indica al puerto que está listo para iniciar la transmisión de datos.

DFD

Diagrama de Flujo de Datos. Es una técnica gráfica para representar el flujo de información y las transformaciones que sufren los datos cuando circulan de entrada a salida.

Diagnoster

Equipo de diagnóstico que emula la mayoría de las funciones realizadas por la UTR, así como diversos protocolos

DSR

Data Set Ready. El modem informa al puerto que está energizado y listo. bajo la norma RS-232C

DTR

Data Terminal Ready. El puerto le informa al modem que está listo para iniciar la comunicación.

EM

Estación Maestra de control supervisorio. Se encarga de supervisar y controlar a una o más Unidades Terminales Remotas.

Entrada analógica

Los valores de entrada analógica se obtienen normalmente de transductores de entrada que convierten la cantidad medida (Watts, Vars, Volts, Amperes) a señales de baja energía (por ejemplo 0-1 mA o 4-20 mA) las cuales son fácilmente procesadas con equipo electrónico.

Entrada digital

Los datos de entrada digital se obtienen de la supervisión del estado de contactos externos (abierto/cerrado). Existen cuatro tipos de datos digitales, o información de la posición de los contactos: 1) Estado actual. 2) Estado actual con memoria, esto es, el número de cambios de contactos desde el último reporte a la maestra. 3) Secuencia de eventos y 4) Acumuladores de pulsos.

Exploración

La información del sistema eléctrico debe adquirirse en determinados periodos de tiempo; a la operación de la Estación Maestra de solicitar los datos a la UTR se le conoce como exploración

FIFO

Es una memoria (en este caso de 512 bytes) que sirve para el intercambio de información entre la computadora y el puerto (tarjeta PC-851). Maneja la norma de "primero en entrar, primero en salir", también es un elemento de sincronización de la comunicación entre puerto y módulo maestro.

Grupo de exploración

La información del sistema a controlar se agrupa por funciones para ser explorada con la periodicidad requerida, grupos de mayor prioridad, como pueden ser el estado de interruptores son solicitados más frecuentemente, las mediciones pueden ser adquiridas con menor frecuencia y los acumuladores de pulsos cada hora o cada 24 horas.

ISO

Organización internacional de estándares.

Modem

Dispositivo electrónico que consiste de un modulador y un demodulador, utilizado para la transmisión de información por un canal de comunicación.

OSI

Modelo de referencia conocido en la literatura como "Interconexión de sistemas abiertos" utilizado en las redes de comunicación de un sistema abierto.

Proceso ó tarea

Es una función o procedimiento en un ambiente multi-tareas que se ejecuta concurrentemente en un solo procesador, posee recursos propios y comparte otros administrados por el sistema operativo.

Puerto

Se denomina así al módulo esclavo perteneciente a la arquitectura del CIC que se encarga de la comunicación hacia la UTR, haciendo el protocolo completamente transparente para el módulo maestro.

RTS

Request To Send. El puerto le informa al modem que desea transmitir datos, bajo la norma RS-232C.

Rx

Línea de recepción de la información mediante el canal serie bajo la norma RS-232C.

SCADA

Sistema de control supervisorio y adquisición de datos. Adquiere y concentra la información en tiempo real de los dispositivos de campo para su supervisión y telecontrol por medio de un operador o de manera automática.

Semáforo

Un semáforo es una estructura que permite la sincronización entre procesos, son de tipo contador y tienen asociada una cola para procesos bloqueados. Se manejan dos operaciones en un semáforo: señal y espera.

Tx

Línea de transmisión de datos por el canal serie bajo la norma RS-232C

UTR

Unidades Terminales Remotas. Reciben la información directamente de los elementos de campo y se encargan de darle un formato para ser enviado a la Estación Maestra mediante un protocolo de comunicación, asimismo, reciben comandos de control de la Estación Maestra para accionar los dispositivos de campo como interruptores, válvulas, etc.