

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA ELECTRÓNICA Y ELÉCTRICA

E.A.P. DE INGENIERIA ELECTRÓNICA

Supervisión y monitoreo de procesos utilizando mensajes de texto desde teléfonos celulares

TESIS Para optar el Título Profesional de INGENIERO ELECTRÓNICO

AUTOR

Luis Humberto Pérez Urteaga

ASESOR Guillermo Tejada Muñoz

LIMA – PERÚ 2006

Dedicación:

A mi hijo Luis Eduardo, mi mayor motivo para seguir siempre adelante y a todos mis seres queridos ausentes y presentes.

Luis Humberto Pérez Urteaga.

AGRADECIMIENTOS

A mis padres y hermanos, por todo el apoyo brindado durante el desarrollo de este trabajo de Tesis, ya que sus consejos y sugerencias fueron un gran impulso para seguir avanzando en mi investigación. A mi primer maestro en Electrónica, mi abuelo, el Ingeniero Edwin Pérez Villareal, quien despertó en mi el interés por esta profesión. Al Ingeniero Guillermo Tejada Muñoz quien como asesor de este trabajo, siempre me brindó en calidad de amigo, todo el apoyo, ideas y la claridad necesaria para culminar este trabajo. Al laboratorio e Instituto de Investigación de la Facultad de Ingeniería Electrónica y Eléctrica, especialmente a la profesora Lita Soto Nieto y al Ingeniero Esequiel Zavala Huavel por brindarme su apoyo y consejos, además de las facilidades necesarias dentro de sus instalaciones. A la Facultad de Ingeniería Química y a la Profesora Gloria Contreras quienes brindaron todas las facilidades para realizar pruebas empleando sus equipos. Finalmente también quiero agradecer a todas las personas que de una u otra manera, directa o indirectamente apoyaron para la realización de este trabajo.

INDICE GENERAL

Lista de Figuras

Lista de Tablas

Resumen

I. INTRODUCCIÓN.....	1
1.1 Objetivos.....	3
1.2 Justificación.....	3
1.3 Antecedentes	4
1.4 Metodología.....	5
1.5 Aportes	7
1.6 Descripción del sistema.....	7
II. ASPECTOS TEÓRICOS.....	12
2.1 Los teléfonos celulares	12
2.2 Comunicaciones seriales	14
2.3 El modelo OSI.....	18
2.3.1 La capa física.....	19
2.3.2 La capa de enlace.....	19
2.3.3 La capa de red	20
2.3.4 La capa de transporte	21
2.3.5 La capa de sesión.....	22
2.3.6 La capa de presentación.....	22
2.3.7 La capa de aplicación	23
2.4 Descripción del protocolo FBUS	24
2.5 El servicio de mensajes de texto cortos (SMS).....	27

2.5.1	Arquitectura de red para el servicio SMS	27
2.5.2	Protocolo SM - TP	29
2.5.3	Codificación y decodificación de caracteres para SMS	31
2.5.3.1	Codificación	32
2.5.3.2	Decodificación.....	34

III. ANÁLISIS DEL PROTOCOLO FBUS..... 36

3.1	Herramientas de análisis.....	36
3.1.1	Hardware de interfaz para la serie Nokia 33XX	37
3.1.2	Hardware canalizador de la comunicación.....	38
3.1.3	Software SMS Manager	38
3.1.4	Software para captura de datos seriales	40
3.1.5	Software de envío de tramas hacia el teléfono celular	42
3.1.6	Software codificador y decodificador de caracteres para SMS	43
3.2	Identificación de las tramas en la comunicación.....	44
3.2.1	Conclusiones de la identificación de las tramas.....	46
3.3	Identificación de los tipos de tramas.....	48
3.3.1	Análisis del contenido de las tramas	49
3.3.1.1	Lectura del directorio telefónico	50
3.3.1.2	Cantidad de mensajes de texto grabados en memoria.....	51
3.3.1.3	Lectura de mensajes de texto	53
3.3.1.4	Borrado de mensajes de texto	56
3.3.1.5	Detección de la llegada de mensajes de texto nuevos	57
3.3.1.6	Envío de mensajes de texto	58
3.3.2	Conclusiones de los detalles de las tramas	61

IV. IMPLEMENTACIÓN DEL PROTOCOLO FBUS 65

4.1	Capa física.....	66
4.2	Capa de enlace	67
4.2.1	Composición / Descomposición de los campos de la trama recibida.....	67
4.2.2	Detección del inicio y final de una trama válida	69
4.2.3	Verificación de checksums en las tramas provenientes del teléfono	71

4.2.4	Cálculo de checksums para las tramas por enviar.....	73
4.2.5	Detección y generación de la trama Acknowledge	73
4.2.6	Cálculo de la secuencia de las tramas por enviar.....	74
4.3	Capa de presentación	75
4.3.1	Obtención de la información contenida en el bloque de datos	75
4.3.2	Generalización de tramas para solicitar acciones al teléfono celular	79
4.4	Capa de aplicación.....	79

V. APLICACIONES PROTOTIPO PARA DEMOSTRAR LA UTILIDAD DE LA IMPLEMENTACIÓN DEL PROTOCOLO..... 80

5.1	Matrícula vía SMS (Short Message Service).....	80
5.1.1	Descripción de la interfaz gráfica.....	82
5.1.1.1	Ventana Conexión	83
5.1.1.2	Ventana Mensajes de texto	84
5.1.1.3	Ventana Consultas SQL.....	85
5.1.2	Pruebas del prototipo de Matrícula.....	85
5.1.3	Limitaciones	86
5.2	Supervisión y control de un proceso industrial vía SMS	87
5.2.1	Descripción de la interfaz gráfica.....	88
5.2.1.1	Ventana Comunicación	88
5.2.1.2	Ventana Nombres de I/Os.....	90
5.2.1.3	Ventana Notificaciones	91
5.2.1.4	Ventana Valores I/Os.....	92
5.2.2	Formato para el envío de ordenes.....	93
5.2.3	Pruebas del prototipo Monitor-Supervisor	94
5.2.3.1	Con la Unidad de adsorción de gases en alto vacío de la Facultad de Ingeniería Química.....	95
5.2.3.2	Con el módulo de nivel de líquidos del laboratorio de la Facultad de Ingeniería Electrónica.....	98
5.2.4	Limitaciones	102
5.3	Procesamiento de los mensajes de texto en las aplicaciones descritas	102

VI. CONCLUSIONES Y RECOMENDACIONES.....	105
6.1 Conclusiones.....	105
6.2 Recomendaciones	105
VII. BIBLIOGRAFÍA	107
APÉNDICE	109

Lista de Figuras.

Figura 1.1. Interfaz gráfica Matrícula	2
Figura 1.2. Interfaz gráfica Monitor – Supervisor.....	2
Figura 1.3. Esquema del sistema desarrollado	8
Figura 1.4. Diagrama esquemático de la tarjeta I/O	10
Figura 1.5. Descripción de los conectores y borneras de la tarjeta I/O	11
Figura 2.1. Diagrama de bloques de un teléfono celular GSM	12
Figura 2.2. Puertos de comunicación del teléfono celular	13
Figura 2.3. Comunicación serial unidireccional de dos hilos	14
Figura 2.4. Esquema de bits en una comunicación serial asíncrona.....	15
Figura 2.5. Niveles de voltaje en el RS-232C	16
Figura 2.6. Esquema de bits en el RS-232C	16
Figura 2.7. Conectores DB-25 y DB-9 Macho de la computadora	17
Figura 2.8 Capas del modelo OSI	18
Figura 2.9. Detalles de la comunicación entre el teléfono celular y la computadora.....	24
Figura 2.10. Estructura del protocolo FBUS Versión 2.....	25
Figura 2.11. Trama de Acknowledge en la versión 2 del protocolo FBUS	26
Figura 2.12. Esquema del servicio SMS	27
Figura 2.13. Arquitectura de red para el servicio SMS	28
Figura 2.14. Arquitectura de capas para el servicio SMS	29
Figura 2.15. Estructura de la trama SMS - SUBMINT	30
Figura 3.1. Diagrama del hardware de interfaz para el teléfono celular Nokia 3390	37
Figura 3.2. Interfaz BBS-8:0630261.....	37
Figura 3.3. Circuito canalizador de la comunicación	38
Figura 3.4. Ventana principal del programa SMS Manager	39
Figura 3.5. Ventana del software de captura para datos seriales	40
Figura 3.6. Diagrama de flujo del programa Grabador de Datos seriales	41
Figura 3.7. Ventana del programa para enviar tramas al teléfono celular	43
Figura 3.8. Diagrama de flujo para la codificación y decodificación de caracteres	43
Figura 3.9. Ventana del programa codificador y decodificador de caracteres.....	44
Figura 3.10. Obtención de los datos de la comunicación.....	45
Figura 3.11. Procedimiento manual de envío de tramas.....	49

Figura 4.1. Comparación de la comunicación FBUS con el modelo OSI.....	65
Figura 4.2. Flujo de la comunicación en el protocolo FBUS	66
Figura 4.3. Cantidad de ciclos de transmisión para enviar una trama.....	69
Figura 4.4. Diagrama de flujo del proceso de obtención de una trama completa	71
Figura 4.5. Diagrama de flujo para el cálculo y comparación de checksums.....	72
Figura 4.6. Diagrama de flujo para generar la trama de Acknowledge	74
Figura 4.7. Secuencias en tramas de Acknowledge y de solicitudes	75
Figura 4.8. Detección del tipo de mensaje y extracción de la función	76
Figura 4.9. Obtención de los datos útiles del tipo de mensaje 02h.....	77
Figura 4.10. Obtención de los datos útiles del tipo de mensaje 14h.....	78
Figura 4.11. Obtención de los datos útiles del tipo de mensaje 7Fh	79
Figura 5.1. Ventana Conexión.....	83
Figura 5.2. Ventana Mensajes de texto	84
Figura 5.3. Ventana Consultas SQL	85
Figura 5.4. Operaciones realizadas en el prototipo de matrícula vía SMS.	86
Figura 5.5. Ventana Comunicación.....	89
Figura 5.6. Ventana Nombres de I/Os	90
Figura 5.7. Ventana Notificaciones	92
Figura 5.8. Ventana Valores I/Os.....	93
Figura 5.9. Conexión de la unidad de adsorción de gases con la tarjeta I/O	96
Figura 5.10. Configuración de la ventana Nombres I/Os	96
Figura 5.11. Configuración de la ventana Notificaciones	97
Figura 5.12. Prueba número 1 con la unidad de adsorción de gases en alto vacío	97
Figura 5.13. Prueba número 2 con la Unidad de Adsorción de gases en alto vacío	98
Figura 5.14. Interconexión entre los diversos elementos	99
Figura 5.15. Configuración de la ventana Nombres I/Os	100
Figura 5.16. Configuración de la ventana Notificaciones	100
Figura 5.17. Notificación de la parada del proceso	101
Figura 5.18. Arranque remoto del proceso	101
Figura 5.19. Parada remota del proceso	102
Figura 5.20. Procesamiento de los mensajes de texto.....	104

Lista de Tablas

Tabla 2.1	Puertos del teléfono celular	13
Tabla 2.2	Características de comunicacion de los FBUS y MBUS	14
Tabla 2.3	Terminales de los conectores RS232	17
Tabla 2.4	Caracteres disponibles en el servicio SMS	31
Tabla 3.1	Trama obtenida del registro de la comunicación	46
Tabla 3.2	Funciones del teléfono celular agrupados por tipos de mensaje	47
Tabla 3.3	Valores constantes en las tramas enviadas y recibidas.....	48
Tabla 3.4	Trama de solicitud de lectura del directorio telefonico.....	50
Tabla 3.5	Trama de respuesta a la lectura del directorio telefonico.....	51
Tabla 3.6	Trama de solicitud de cantidad de mensajes grabados en memoria.....	51
Tabla 3.7	Trama de respuesta de cantidad de mensajes grabados en memoria.....	52
Tabla 3.8	Trama de solicitud de lectura de mensajes de texto	53
Tabla 3.9	Trama de respuesta a la lectura de mensajes de texto.....	54
Tabla 3.10	Trama de continuacion del mensaje de texto	56
Tabla 3.11	Trama de borrado de mensajes de texto	56
Tabla 3.12	Trama de lectura del numero de la central de mensajes de texto	58
Tabla 3.13	Trama de respuesta al numero de la central de mensajes de texto	59
Tabla 3.14	Trama de solicitud de envío de mensajes de texto	60
Tabla 3.15	Valores hexagesimales de las funciones analizadas	62
Tabla 3.16	Generalizacion de las solicitudes	63
Tabla 4.1	Lista de tipos de mensajes	68

RESUMEN

El presente trabajo de tesis, del área de Sistemas Digitales, tiene como meta principal convertir al teléfono celular en un periférico de la computadora, con esta sociedad, vista como una unidad, la computadora adquiere la capacidad de procesar datos que son enviados a grandes distancias vía mensajes de textos desde teléfonos celulares, potenciando las aplicaciones de ambos dispositivos.

Para lograr la fusión entre el celular y la computadora ha sido necesario analizar, implementar y validar el funcionamiento del protocolo de comunicación que se debe establecer entre el teléfono celular y la computadora.

La diversidad de aplicaciones que pueden ser desarrolladas es amplia, pudiendo ser empleada en áreas de telecontrol, telemedicina, domótica, entre otras, así pues, para demostrar la utilidad de la implementación del protocolo, se ha validado su funcionamiento mediante dos aplicaciones prototipo, la primera de ellas denominada *Matrícula*, la cual permite realizar la matrícula de alumnos a través de mensajes de texto y la segunda denominada *Monitor – Supervisor*, para aplicaciones de telecontrol y envío de alertas vía mensajes de textos, las cuales han sido probadas obteniéndose los resultados esperados. En el segundo prototipo ha sido necesario diseñar una tarjeta que comunica el puerto paralelo de la computadora con los sensores y actuadores de los procesos seleccionados.

Como consecuencia, al describir todos los detalles para el análisis e implementación del protocolo y sus aplicaciones, se provee a otros investigadores interesados la mayor cantidad de información sobre este tema, la cual es escasa en nuestro medio.

I. INTRODUCCIÓN.

El presente trabajo de tesis pertenece al área de Sistemas Digitales ya que se centra en estudiar la manera de intercambiar y procesar información, así como la de buscar aplicaciones de Ingeniería dentro del plano físico ubicado entre el teléfono celular y la computadora, dejando de lado los detalles referentes a la comunicación entre el teléfono y la red de telefonía del proveedor del servicio, que es competencia de estudio de otras áreas de ingeniería.

Mediante el presente estudio, se ha logrado convertir al teléfono celular en un periférico de la computadora brindándole un nuevo medio de comunicación inalámbrico a distancia, se ha optado por trabajar con la marca de celulares Nokia por ser de amplia difusión además de permitir la interconexión de equipos computarizados mediante el puerto serial (RS232) a través de un circuito de interfaz. Este trabajo también pretende ser una guía para la implementación de protocolos utilizados por teléfonos celulares de otras marcas.

La implementación del protocolo FBUS, ha permitido diseñar dos aplicaciones prototipo para demostrar su utilidad en aplicaciones de ingeniería, la primera denominada Matrícula, cuya ventana principal se muestra en la figura 1.1, que interactúa con servidores Microsoft SQL Server permitiendo la matrícula de alumnos vía SMS (Short Message Service – Servicio de mensajes cortos) y la segunda denominada Monitor–Supervisor cuya ventana principal se muestra en la figura 1.2, para aplicaciones de Telecontrol y envío de alertas vía SMS.

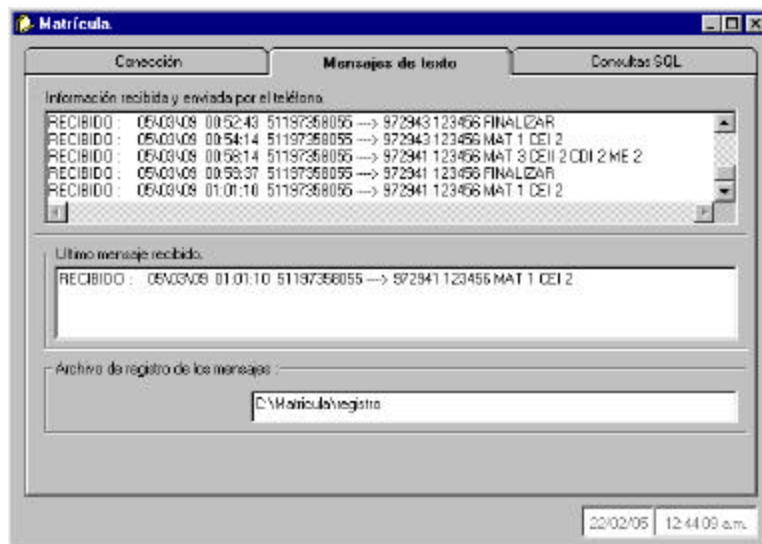


Figura 1.1. Interfaz gráfica Matrícula

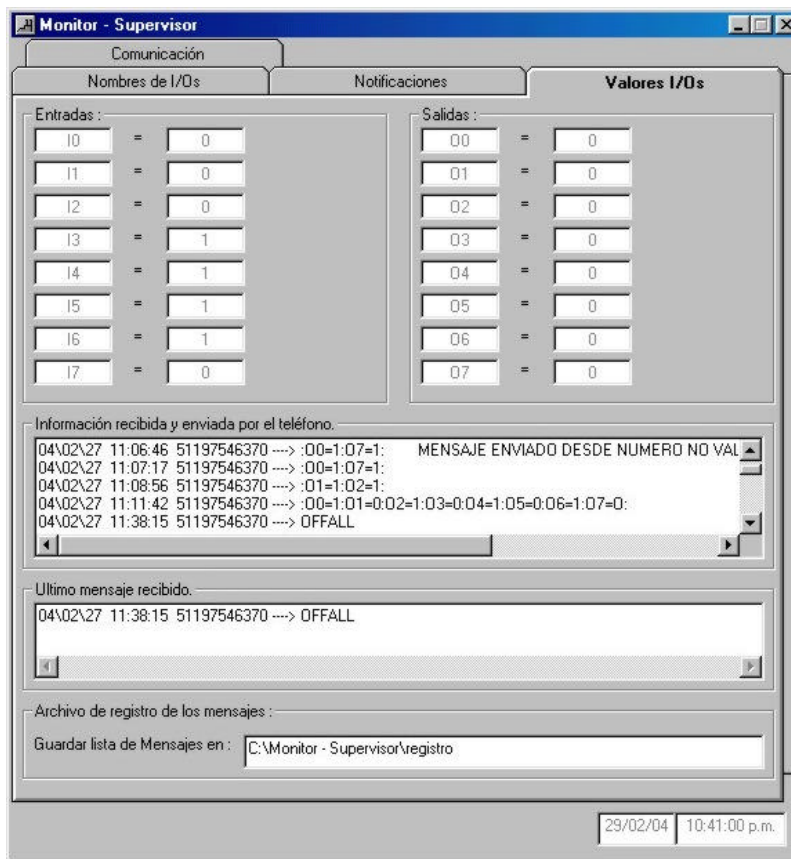


Figura 1.2. Interfaz gráfica Monitor – Supervisor

Este trabajo de tesis pretende facilitar a las personas interesadas en el tema toda la información necesaria para poder desarrollar con facilidad múltiples aplicaciones las cuales dependerán de sus necesidades e imaginación, así pues, en el capítulo II se ha incluido todos los aspectos teóricos relacionados al trabajo; en el capítulo III, se describe el análisis del protocolo FBUS así como también las diversas herramientas hardware y software que han sido necesario implementar para realizar una completa descripción de cada byte de los diversos tipos de tramas del formato del protocolo. En el capítulo IV, se encuentra la implementación del protocolo tomando como referencia al modelo OSI. Se encuentran también detallados, los diversos algoritmos y diagramas de flujo empleados para cada nivel de la comunicación lo cual facilita cualquier futura implementación del protocolo en otras plataformas o lenguajes de programación. En el capítulo V, se describen dos prototipos de aplicación en donde se demuestra el rendimiento del protocolo. La primera aplicación prototipo, permite realizar la matrícula de los alumnos de la FIE y muestra la potencialidad para interactuar con bases de datos. Así mismo esta aplicación también puede ser aprovechada por la Facultad o la Universidad para ofrecer un servicio de matrícula vía teléfonos celulares para casos

especiales. La segunda aplicación prototipo está desarrollada para aplicaciones de telecontrol y envío de alertas y puede ser fácilmente integrada a diversos procesos o sistemas de alarmas en general. Estos prototipos de aplicación han sido probados, obteniéndose los resultados esperados. En el capítulo VI, se describen las conclusiones y recomendaciones.

1.1 Objetivos

- Implementar el protocolo FBUS, con el fin de convertir al teléfono celular en un periférico adicional de la computadora de tal manera de poder potenciar las aplicaciones empleando ambos dispositivos.
- Validar el correcto funcionamiento del protocolo implementado, así como su utilidad práctica mediante dos aplicaciones prototipo.
- Motivar a otros investigadores para que tomando este trabajo como base puedan perfeccionar las presentadas como prototipos o desarrollar aplicaciones en otras áreas.

1.2 Justificación

Debido a la difusión masiva de los teléfonos celulares y computadoras, las cuales operando independientemente, una de otra es justificable realizar investigaciones para facilitar su interconexión de tal manera que con ambos dispositivos trabajando juntos se puedan desarrollar nuevas aplicaciones en Ingeniería.

Debido a que la información teórica sobre protocolos de comunicación entre computadoras y teléfonos celulares es muy escasa, este trabajo se justifica por dar a conocer los detalles de estos protocolos.

1.3 Antecedentes

Existen equipos comerciales para aplicaciones en Domótica, los cuales ofrecen la posibilidad de notificar al usuario hechos como la activación de alarmas, equipos, etc. Así mismo, también se brinda la posibilidad de realizar la activación y desactivación remota de equipos; este es el caso del sistema InControl Plus por ejemplo, el cual es

comercializado en Argentina y brinda la posibilidad de controlar y monitorear remotamente artefactos eléctricos, mirar y escuchar en tiempo real por medio de cámaras lo que viene sucediendo en el hogar u oficina; también actúa como un sistema de seguridad que envía información para notificar al usuario desde su casa directamente al teléfono celular o por medio de un correo electrónico.

En el año 2003 Entel PCS y ADT de Chile, lanzó el servicio de seguridad casa inteligente, el cual ofrece a los usuarios mantener una comunicación bidireccional para conocer hechos como la activación de alarmas y realizar procesos como desconexión de aparatos eléctricos a distancia.

MobiServer el cual interactúa con servidores de cómputo brindándole la posibilidad de compartir unidades y directorios, generar nuevos usuarios o cambiar las contraseñas, reiniciar el servidor, entre otras cosas haciendo uso de teléfonos celulares, empleado la tecnología WAP (Wireless Application Protocol - protocolo de aplicaciones inalámbricas).

Existen en la actualidad interfaces que pueden ser conectados con PLCs (autómatas programables) para brindarles la característica de notificar la activación de alarmas y realizar la activación de salidas vía mensajes de texto desde teléfonos celulares.

Existen programas denominados data suites los cuales permiten acceder a las funciones del teléfono celular desde una computadora personal, tal es el caso del programa SMS Manager del grupo Software Cave, Proyecto Gnokii y el programa Logo Manager, inclusive el grupo Software Cave proporciona librerías para Visual Basic y otros lenguajes de programación para realizar la comunicación con el teléfono celular y tener el acceso a las funciones del teléfono.

Si bien como se menciona, existe en internet librerías para algunos lenguajes de programación las cuales facilitan la tarea de comunicarse con el teléfono celular, éstas son escasas y no brindan la información necesaria acerca de las técnicas y algoritmos empleados para esta labor, convirtiéndose en "cajas negras". Inclusive la información relacionada a estos aspectos es limitada.

1.4 Metodología

Inicialmente, se realizó una búsqueda de información sobre la conectividad entre teléfonos celulares y computadoras personales así como el hardware necesario para este propósito. Se encontró que cada marca y modelo de teléfonos utilizan tipos diferentes de comunicación en los cuales el hardware requerido es distinto para cada marca y modelo. Se optó por trabajar con teléfonos celulares de la marca Nokia, debido a que son de amplio uso mundial y a que fue posible encontrar información referente a los protocolos de comunicación que emplean. Dentro de la marca Nokia se eligió el modelo 3390 debido a que utiliza los dos tipos de protocolos desarrollados por Nokia (MBUS y FBUS). Según las fuentes bibliográficas consultadas, se encontró que el protocolo MBUS se utiliza sólo para aplicaciones de servicio técnico, por este motivo se optó por el FBUS, el cual se utiliza para aplicaciones extendidas de muchos modelos de la marca (Nokia 6110, 6130, 6150, 6190, 5110, 5130, 5150, 5190, 3210, 3310, 3330, 3390, 3391, 3395, etc).

Con la información técnica encontrada, se implementó el hardware interfaz de comunicación para el teléfono celular Nokia 3390, que permite interconectar el teléfono celular con la computadora (ver ítem 3.1.1).

Fue necesario verificar y complementar la información insuficiente encontrada sobretodo en lo que respecta a la estructura de las tramas, se optó por verificar detalles del funcionamiento del protocolo FBUS mediante algún software existente que hiciera uso de este protocolo. Es así como se eligió el software de distribución gratuita SMS Manager (ver ítem 3.1.3).

Se diseñó un software para la captura de datos seriales (ver ítem 3.1.4) y un hardware canalizador de la comunicación (ver ítem 3.1.2) el cual, combina las señales de transmisión y recepción del puerto serial para que ambas se dirijan en un mismo sentido de transmisión hacia otro puerto serial donde fue grabada para ser ordenada, impresa y comparada con la información teórica que se quiso verificar. De esta manera, se pudo descifrar características importantes del funcionamiento del protocolo FBUS.

Luego que se complementó la información sobre las tramas del protocolo, se diseñó un software para envío de datos desde la PC hacia el teléfono celular (ver ítem 3.1.5) permitiendo visualizar la respuesta del teléfono. Paralelamente con los resultados

obtenidos se programó en Visual Basic las rutinas necesarias para realizar la comunicación entre la computadora y el teléfono utilizando el protocolo FBUS. Para la implementación del protocolo, se optó por utilizar como referencia al modelo OSI (ver ítem 2.2), estableciéndose 4 niveles para la comunicación FBUS (ver capítulo IV) .

Fue necesario además, realizar un estudio de publicaciones de la ETSI (European Telecommunications Standards Institute), específicamente la GSM 03.38 y GSM 03.40 para saber los detalles de la codificación y decodificación de mensajes de texto y el empaquetado de datos para el envío por mensajes de texto en sistemas GSM (Global System for Mobil Communications).

Con las rutinas implementadas para la comunicación FBUS, se diseñó dos prototipos de aplicación para demostrar la utilidad de la implementación del protocolo. El primer prototipo denominado Matrícula permite realizar la matrícula de alumnos vía SMS, registrando los datos en un servidor de base de datos. El segundo prototipo denominado Monitor – Supervisor fue desarrollado para aplicaciones de Telecontrol y envío de alertas. En ambos casos, se obtuvo resultados satisfactorios, lo cual demostró la validez del estudio, es decir del análisis e implementación del protocolo y su utilidad para aplicaciones de ingeniería.

El proyecto se realizó empleando una computadora Pentium III con un microprocesador de 1Ghz, los prototipos de aplicación han sido probados además en una computadora Pentium I con un microprocesador de 233 Mhz funcionando de manera correcta.

1.5 Aportes

- Contribuir con la tecnología nacional, al divulgar con los resultados de la Tesis el procedimiento por el cual se puede utilizar al teléfono celular como un periférico adicional de la computadora.
- Utilizar la capacidad instalada de las redes de telefonía celular de amplia cobertura, para implementar diversas aplicaciones de ingeniería, tales como telemetría, telecontrol, domótica, etc.

- Presentar a la Facultad, un prototipo que permita realizar la matrícula de alumnos empleando mensajes de texto.
- Presentar un prototipo de telecontrol y envío de alertas a través de mensajes de texto que pueda integrarse con facilidad a múltiples sistemas que lo requieran.

1.6 Descripción del sistema

El sistema, ha sido desarrollado de tal manera que permite integrarse con facilidad a computadoras personales para una diversidad de aplicaciones como el manejo de bases de datos, telecontrol, telemetría, envío de alertas (alarmas), etc. está constituido como se muestra en la figura 1.3 por los siguientes elementos:

- Computadora.
- Teléfono celular.
- Hardware de interfaz.
- Software instalador del sistema.
- Tarjeta de entradas y salidas (tarjeta I/O).

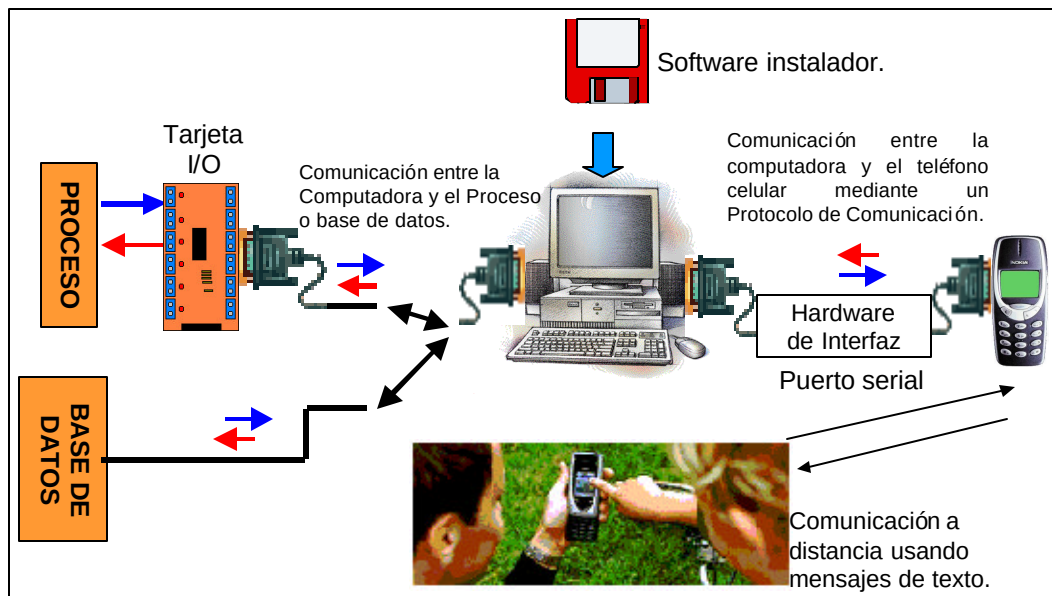


Figura 1.3. Esquema del sistema desarrollado

El teléfono celular se comunica con la computadora por medio del puerto serial en donde el hardware de interfaz acondiciona los niveles de voltaje entre ambos dispositivos, estableciéndose un medio físico por el cual los mensajes de texto son transmitidos a la computadora para su procesamiento y ejecución. También es el encargado de enviar la información de respuesta que el sistema reporta al usuario.

El software instalador del sistema permite instalar en la computadora las aplicaciones prototipos Matrícula o Monitor – Supervisor según sea el caso. En ambas aplicaciones, el protocolo FBUS realiza la comunicación entre la computadora y el teléfono celular.

La tarjeta I/O que se muestra en la figura 1.4, se utiliza con la aplicación prototipo Monitor – Supervisor , está conectada al puerto paralelo de la Pc, desde donde se recibe y entrega las señales del proceso que se desea controlar, Esta tarjeta posee en la etapa de salida, relés que son manejados por transistores los cuales se saturan de acuerdo al estado lógico de los bits de datos del puerto paralelo, Los relés tienen la capacidad de corriente como para poder activar o desactivar equipos eléctricos o electrónicos. En la etapa de entrada mediante el circuito integrado 74LS241 los bits de entrada del proceso son introducidos al puerto de estado del puerto paralelo. Se adicionó en cada entrada diodos zenner de 5.1 voltios con el fin de poder aceptar voltajes superiores a los 5 voltios a modo de protección.

La figura 1.5 muestra los conectores y borneras de la tarjeta I/O, las cuales brevemente se describen a continuación:

- 8 salidas tipo relés. Los contactos de los relés, son accesibles por medio de las borneras identificadas como O0 hasta O7, cada una de ellas tiene 3 terminales, uno que es el COMUN, otro normalmente cerrado (NC) y otro normalmente abierto (NA).
- 5 borneras de entrada de datos identificadas desde I3 hasta I7 de dos terminales, uno de los terminales recibe la señal y el otro terminal denominado GND, el cual está conectado al punto de referencia.
- Conector DB25 (P1), que permite la interconexión de la tarjeta con el puerto paralelo de la computadora.
- Conector CP00, para alimentar a la tarjeta. La fuente de alimentación puede variar entre 7.5 y 15 voltios, y debe entregar una corriente mínima de 300mA.

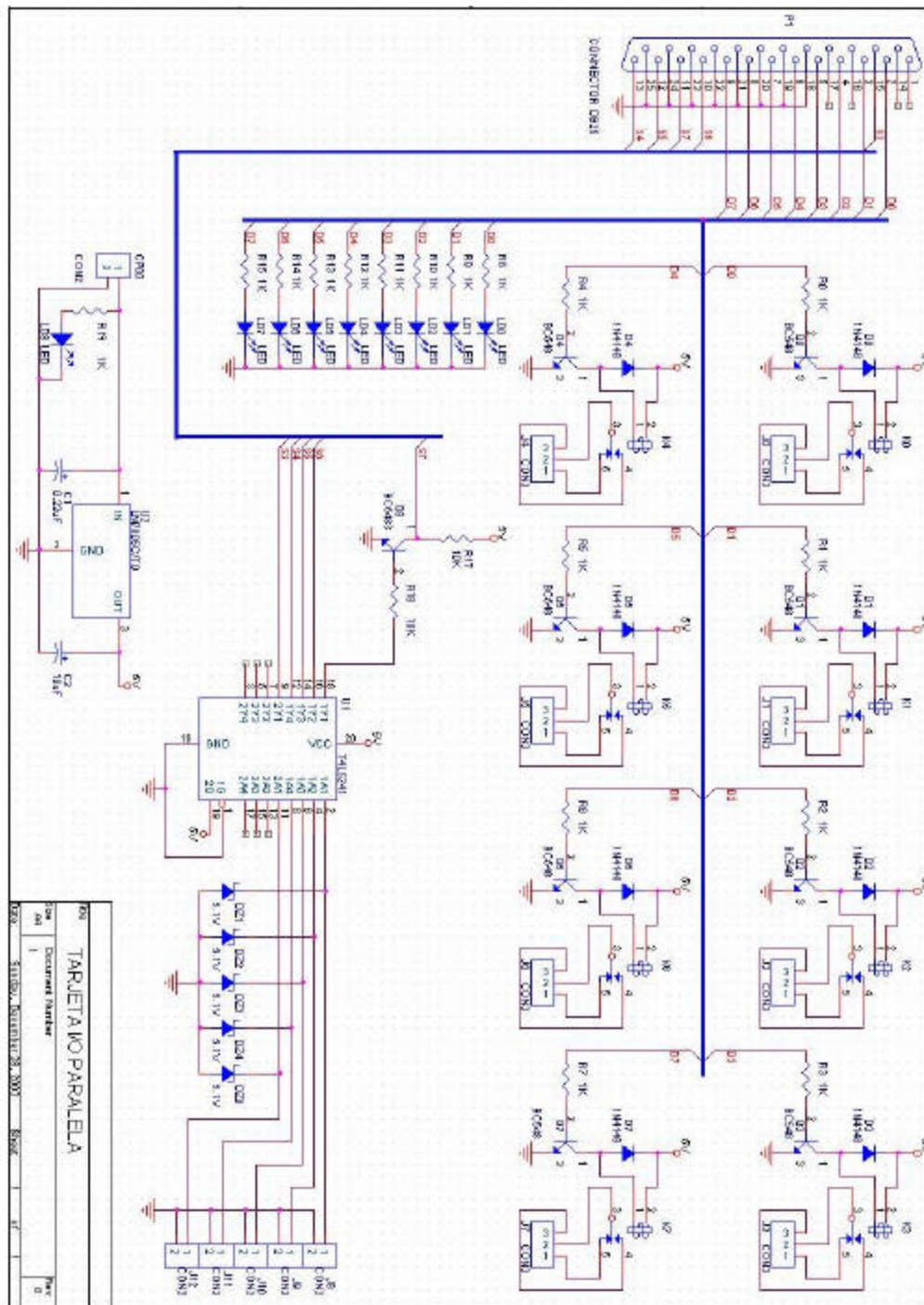


Figura 1.4. Diagrama esquemático de la tarjeta I/O

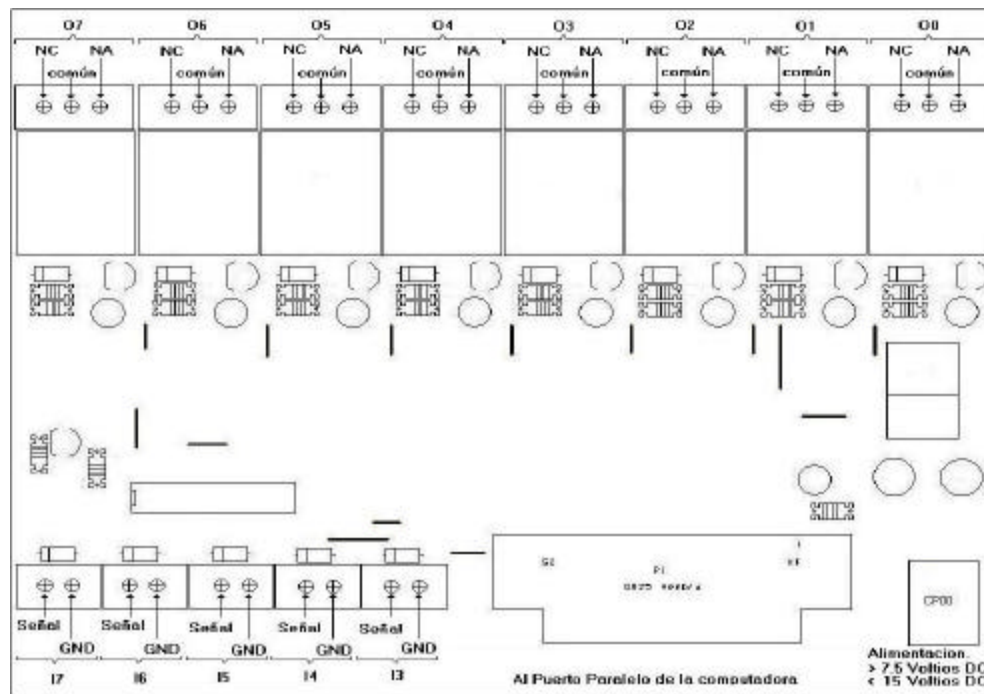


Figura 1.5. Descripción de los conectores y borneras de la tarjeta I/O

II. ASPECTOS TEÓRICOS

2.1 Los teléfonos celulares

En un teléfono celular se puede distinguir unidades bien diferenciadas, donde cada una realiza funciones específicas para el funcionamiento del teléfono. En la figura 2.1, se aprecia el diagrama de bloques de un teléfono celular GSM, el cual contiene las siguientes unidades:

- Unidad de RF, conformado por el transmisor y el receptor.
- Unidad de procesamiento de información, conformado por el DSP o microprocesador, memoria y codificador/decodificador de voz;
- Unidad de alimentación del sistema; conformado por la fuente de alimentación DC, batería y el cargador.
- Unidad de Interfaz con el usuario, conformado por el teclado y el LCD.
- Unidad de entrada/salida, conformado por los puertos de entrada/salida y el conector de la tarjeta SIM.

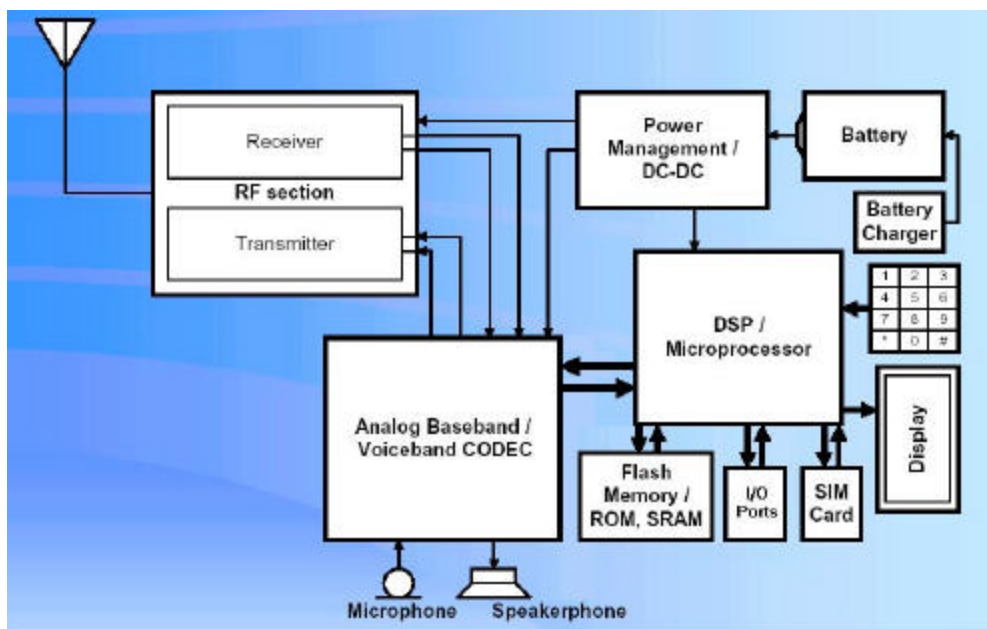


Figura 2.1. Diagrama de bloques de un teléfono celular GSM

Para facilitar su programación, los teléfonos celulares, disponen de puertos de interfaz; la comunicación por estos puertos se realiza mediante protocolos de comunicación definidos por el fabricante los cuales pueden ser desde protocolos muy simples hasta protocolos mas o menos complejos. Para el caso del teléfono celular utilizado en este trabajo, se dispone de puertos de comunicación, los cuales se encuentran situados en la parte posterior del teléfono [8]. La figura 2.2 muestra los detalles de los puertos, mientras que la tabla 2.1 detalla los niveles de voltaje aceptados por el teléfono.



Figura 2.2. Puertos de comunicación del teléfono celular

TABLA 2.1
PUERTOS DEL TELÉFONO CELULAR

Pin	Name	Parameter	Min	Typ	Max	Unit	Remark
1	FBUS_TX	Data ack. to the Prommer	0 2.24	logic low logic high	0.62 2.85	V	Transmit Data from MAD2WD1 (AccTxData) to Prommer (@ VBB=2,85V)
2	FBUS_RX	Serial data from the Prommer	0 1,68	logic low logic high	0.84 2.85	V	Receive Data from Prommer to MAD2WD1 (AccRxData) (@ VBB=2,85V)
3	GND	GND	0		0	V	Ground
4	MBUS	Serial clock from the Prommer	0 1,68	logic low logic high	0.84 2.85	V	Prommer detection and Serial Clock for synch. Comm. to MAD2WD1 (MBUS) (@ VBB=2,85V)

De acuerdo a esta información, los terminales corresponden a una línea de transmisión de datos (TX FBUS), una línea de recepción de datos (RX FBUS), una línea bidireccional de

transmisión y recepción de datos (MBUS) y una línea de referencia (0 voltios o GND). Estos terminales entregan y reciben señales cuadradas del orden de 2.85 Vpp en sus ciclos de transmisión y recepción de datos y constituyen puertos de comunicación seriales cuyas velocidades de comunicación son de 9600Bps y 115200Bps según sea el caso del puerto utilizado.

Los dos puertos se comunican con características diferentes las cuales han sido definidas por el fabricante (Nokia). La tabla 2.2 muestra las características respecto a los parámetros de comunicación de los dos puertos.

TABLA 2.2
CARACTERÍSTICAS DE COMUNICACION DE LOS FBUS Y MBUS

	Puerto MBUS	Puerto FBUS
Característica	Half duplex	Full duplex
Velocidad de Comunicación	9600bps	115200bps
Bits de datos	8	8
Paridad	odd (impar)	N (Ninguna)
Bits de parada	1	1
Versiones del Protocolo	MBUS Version 1 y 2	FBUS Version 1 y 2

2.2 Comunicaciones seriales

Hoy en día, es muy común encontrar equipos electrónicos que realizan transmisiones de datos a grandes distancias empleando medios de comunicación rápidos y confiables. Cuando esto ocurre, las comunicaciones de datos seriales a diferencia de las en paralelo, permiten reducir costos y simplificar la implementación de la misma debido a la utilización de una menor cantidad de hilos tal como muestra la figura 2.3.

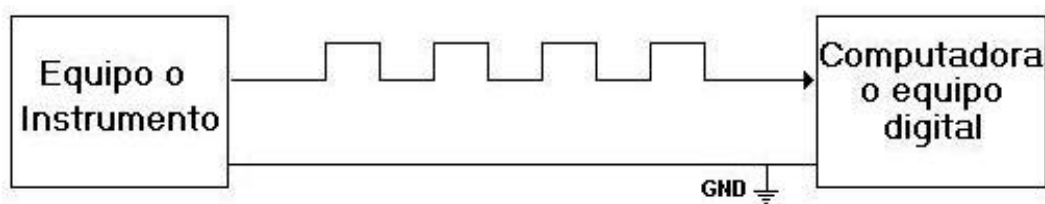


Figura 2.3. Comunicación serial unidireccional de dos hilos

Se pueden construir sistemas de transmisión serie para operar en modos simplex, semi - duplex (half duplex) y duplex completo (Full duplex). La transmisión simplex está constituida por dos hilos y es unidireccional, esto quiere decir que un sistema diseñado para trabajar en este modo sólo tiene un transmisor en un extremo y un receptor en el otro. Si la interconexión tiene un transmisor y un receptor en cada extremo, se puede diseñar el sistema para trabajar en modo semi – duplex (con dos hilos) o duplex completo (con tres hilos), en el primer caso la información es enviada en una dirección a la vez, mientras que en el segundo caso los transmisores de ambos extremos pueden trabajar al mismo tiempo [1].

La velocidad con la cual se realiza la comunicación en los sistemas seriales, se define por el número de bits que pueden ser transmitidos por segundo (siendo la unidad el Baudio o bps), para los sistemas de comunicación antiguos, la velocidad de transmisión puede estar comprendida entre 50 y 9600 baudios, mientras que en sistemas modernos la velocidad puede superar los 128000 baudios.

El método de transmisión de datos más empleado es la transmisión asíncrona la cual es relativamente más rápida que la síncrona debido a que no se necesita de una señal de reconocimiento por cada carácter transmitido o recibido. En la figura 2.4 se observa las características de los bits de una comunicación serial asíncrona, al inicio de la transmisión se encuentra un bit de arranque, seguido de los bits que conforman el dato, un bit de paridad para determinar errores y al final 1 o 2 bits de parada.

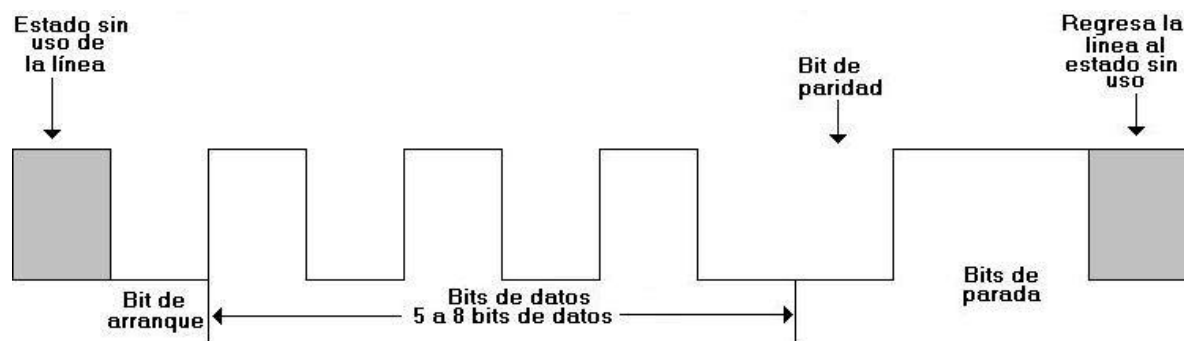


Figura 2.4. Esquema de bits en una comunicación serial asíncrona

Se han establecido múltiples codificaciones para transmisión de información, tales como el código Baudot (de 5 bits por carácter), el código EBCDIC (de 8 bits por carácter), entre

otros que poco a poco han sido dejados de lado, hoy en día el más utilizado, es el código ASCII [2].

Al referirse a comunicaciones seriales asíncronas empleadas en computadoras personales, el estándar empleado es el RS-232C, el cual constituye la tercera revisión de la norma RS-232, propuesta por la EIA (Electronic Industries Assosiation) para comunicaciones seriales.

Para el RS-232C, dependiendo de la velocidad de comunicación, el cable que une el transmisor con el receptor puede tener longitudes de hasta 15 metros asegurando comunicaciones sin perdidas de datos. Las señales de transmisión y recepción, emplean niveles de voltaje de +12V para el 0 lógico y -12V para el 1 lógico tal como se puede apreciar en la figura 2.5. El estado de reposo de las líneas de transmisión y recepción se encuentran siempre en 1 lógico (-12V).

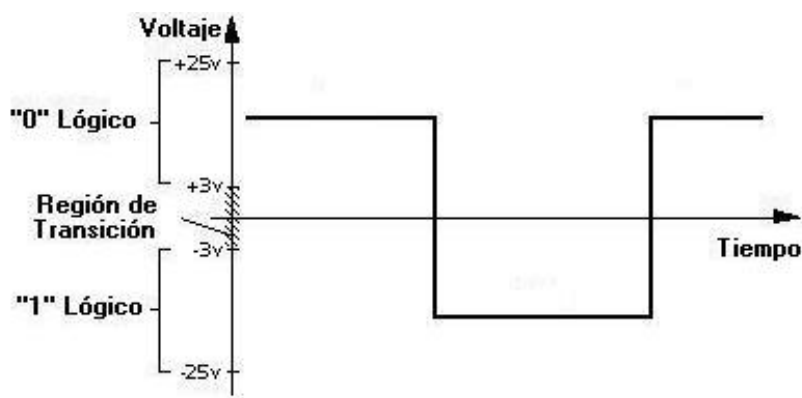


Figura 2.5. Niveles de voltaje en el RS-232C

El RS-232C puede transmitir los datos agrupados de 5, 6, 7 u 8 bits a velocidades determinadas. Después de la transmisión de datos, se genera un bit opcional de paridad par o impar y luego uno o dos bits de parada (figura 2.6).

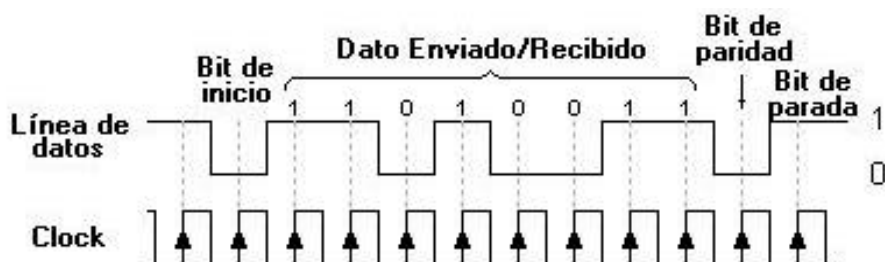


Figura 2.6. Esquema de bits en el RS-232C

Para controlar al puerto serial (COM) de la computadora es necesario especificar su dirección y el número de interrupción (IRQ) utilizado. En el AT-286, se tiene para el COM1 la dirección 3F8h e IRQ 4, y para el COM2 2F8h e IRQ 3; Hasta aquí llega el estándar de las PCs, por lo que si se desea añadir otros puertos seriales como el COM3 y COM4, se encuentran disponibles las direcciones 3E8h y 2E8h mientras que las IRQ pueden ser IRQ3 o IRQ4. No importa compartir una misma IRQ para dos puertos seriales mientras ambos puertos no sean utilizados al mismo tiempo.

Mediante los puertos de entradas y salidas se puede enviar y recibir datos, mientras que las IRQs producen interrupciones para indicar a la CPU que ha ocurrido algún evento como por ejemplo la llegada de algún dato o el cambio de estado de alguna de las señales del puerto. La CPU debe responder lo más rápido posible ante una interrupción para guardar y/o procesar la información recibida en el búfer de la UART (Universal Asynchronous Receiver Transmitter) antes que ésta sea rescrita con nuevos datos recibidos. Las versiones del UART 16550A incluye 2 buffers tipo FIFO de 16 bytes para guardar datos de recepción esperando que el CPU los lea; esto permite disminuir el número de interrupciones por unidad de tiempo generadas por el puerto serial.

En las computadoras personales el conector empleado para las comunicaciones vía RS-232 se encuentra normado al uso de conectores DB-25 y/o DB-9, macho (figura 2.7) dentro del cual, se encuentran las señales indicadas en la tabla 2.3.

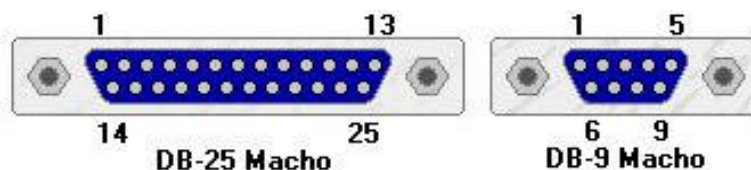


Figura 2.7. Conectores DB-25 y DB-9 Macho de la computadora

TABLA 2.3
TERMINALES DE LOS CONECTORES RS232

Número	de Pin	Señal	Descripción	E / S
DB - 25	DB - 9			
1	-	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC	Transmit Clock	S
17	-	RxC	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC	Transmit/Receive Clock	S

2.3 El modelo OSI

En 1977, la Organización Internacional de Estándares (ISO), integrada por industrias representativas, creó un subcomité para desarrollar estándares de comunicación de datos que promovieran la accesibilidad universal y una interoperabilidad entre productos de diferentes fabricantes. El resultado de estos esfuerzos es el Modelo de Referencia de Interconexión de Sistemas Abiertos (OSI) [3]. Fue creado con los siguientes objetivos:

- Obtener un modelo de referencia estructurado en varios niveles.
- Desarrollar un modelo en el cual cada nivel define un protocolo que realiza funciones específicas diseñadas para atender el protocolo de la capa superior.
- No especificar detalles de cada protocolo.
- Especificar la forma de diseñar familias de protocolos, esto es, definir las funciones que debe realizar cada capa.

El modelo OSI consta de 7 capas como se puede apreciar en la figura 2.8:

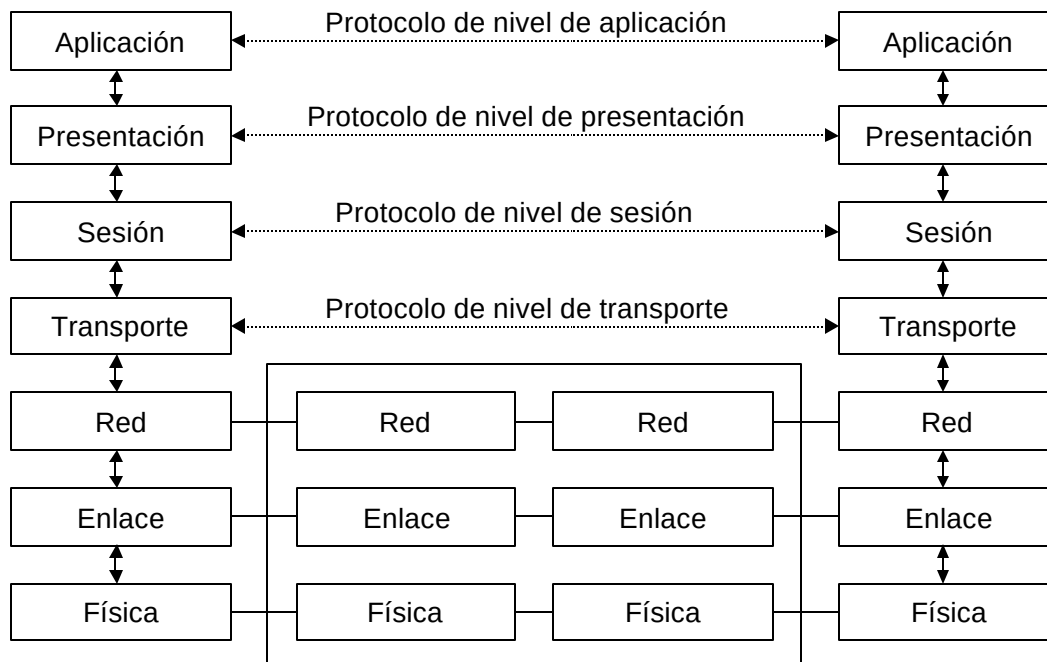


Figura 2.8 Capas del modelo OSI

2.3.1 La capa física

La misión principal de esta capa es transmitir bits por un canal de comunicación, de manera que cuanto envíe el emisor, la información llegue sin alteración al receptor. La capa física proporciona sus servicios a la capa de enlace de datos, definiendo las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales, relacionando la agrupación de circuitos físicos a través de los cuales los bits son transmitidos. Sus principales funciones son:

- Definir las características materiales (componentes y conectores mecánicos) y eléctricas (niveles de tensión) que se van a usar en la transmisión de los datos por los medios físicos.
- Definir las características funcionales de la interfaz (establecimiento, mantenimiento y liberación del enlace físico).

- Transmitir el flujo de bits a través del medio.
- Manejar voltajes y pulsos eléctricos.
- Especificar cables, conectores y componentes de interfaz con el medio de transmisión, polos en un enchufe, etc.
- Garantizar la conexión (aunque no la fiabilidad de ésta).

2.3.2 La capa de enlace

La capa de enlace proporciona sus servicios a la capa de red, suministrando un tránsito de datos confiable a través de un enlace físico. Se ocupa del direccionamiento físico, la topología de red, el acceso a la misma, la notificación de errores, la formación y entrega ordenada de datos y control de flujo. Su principal misión es convertir el medio de transmisión en un medio libre de errores de cualquier tipo, realizando para ello las siguientes funciones:

- Establecer los medios necesarios para una comunicación confiable y eficiente entre dos máquinas en red.
- Agregar una secuencia especial de bits al principio y al final de los paquetes de datos, estructurando este flujo bajo un formato predefinido, denominado trama, que suele ser de unos cientos de bytes.
- Sincronizar el envío de las tramas, transfiriéndolas de una forma confiable libre de errores. Para detectar y controlar los errores se añaden bits de paridad, se usan CRC (Códigos Cíclicos Redundantes) y envío de acuses de recibo positivos y negativos, y para evitar tramas repetidas se usan números de secuencia en ellas.
- Controlar la congestión de la red.
- Regular la velocidad de tráfico de datos.
- Controlar el flujo de tramas mediante protocolos que prohíben que el remitente envíe tramas sin la autorización explícita del receptor, sincronizando así su emisión y recepción.
- Encargarse del acceso de los datos al medio (soportes físicos de la red).

2.3.3 La capa de red

La capa de red proporciona sus servicios a la capa de transporte, siendo una capa compleja que proporciona conectividad y selección de la mejor ruta para la comunicación entre máquinas que pueden estar ubicadas en redes geográficamente distintas. Es la responsable de las funciones de conmutación y enrutamiento de la información (direccionamiento lógico), proporcionando los procedimientos necesarios para el intercambio de datos entre el origen y el destino, por lo que es necesario que conozca la topología de la red (forma en que están interconectados los nodos), con objeto de determinar la ruta más adecuada. Sus principales funciones son:

- Dividir los mensajes de la capa de transporte (segmentos) en unidades más complejas, denominadas paquetes, a los que asigna las direcciones lógicas de los host que se están comunicando.
- Conocer la topología de la red y manejar el caso en que la máquina origen y la máquina destino estén en redes distintas.
- Encaminar la información a través de la red en base a las direcciones del paquete, determinando los métodos de conmutación y enrutamiento a través de dispositivos intermedios (routers).
- Enviar los paquetes de nodo a nodo usando un circuito virtual o datagramas.
- Ensamblar los paquetes en el host destino.
- En esta capa es donde trabajan los routers, dispositivos encargados de encaminar o dirigir los paquetes de datos desde el host origen hasta el host destino a través de la mejor ruta posible entre ellos.

2.3.4 La capa de transporte

La capa de transporte proporciona sus servicios a la capa de sesión, efectuando la transferencia de datos entre dos entidades de sesión. Para ello, divide los datos originados en el host emisor en unidades apropiadas, denominadas segmentos, que vuelve a reensamblar en el sistema del host receptor.

Mientras que las capas de aplicación, presentación y sesión están relacionadas con aspectos de las aplicaciones de usuario, las tres capas inferiores se encargan del transporte de datos. Además, la capa de transporte es la primera que se comunica directamente con su capa par de destino, ya que la comunicación de las capas anteriores es de tipo máquina a máquina.

La capa de transporte intenta suministrar un servicio de transporte de datos que aísle las capas superiores de los detalles del mismo, encargándose de conseguir una transferencia de datos segura y económica y un transporte confiable de datos entre los nodos de la red. Para ello, la capa de transporte establece, mantiene y termina adecuadamente los circuitos virtuales, proporcionando un servicio confiable mediante el uso de sistemas de detección y recuperación de errores de transporte. Se conocen con el nombre de circuitos virtuales a las conexiones que se establecen dentro de una red. En ellos no hay la necesidad de tener que elegir una ruta nueva para cada paquete, ya que cuando se inicia la conexión se determina una ruta de la fuente al destino, ruta que es usada para todo el tráfico de datos posterior. Las funciones de la capa de transporte pueden ser resumidas en los siguientes puntos:

- Controlar la interacción entre procesos usuarios en las máquinas que se comunican.
- Incluir controles de integración entre usuarios de la red para prevenir pérdidas o doble procesamiento de transmisiones.
- Controlar el flujo de transacciones y el direccionamiento de procesos de máquina a procesos de usuario.
- Asegurar que se reciban todos los datos y en el orden adecuado, realizando un control de extremo a extremo.
- Aceptar los datos del nivel de sesión, fragmentándolos en unidades más pequeñas aptas para el transporte confiable, llamadas segmentos, que pasa luego a la capa de red para su envío.
- Realizar funciones de control y numeración de las unidades de información (los segmentos).
- Reensamblar los mensajes en el host destino, a partir de los segmentos que lo forman.
- Garantizar la transferencia de información a través de la red.

2.3.5 La capa de sesión

La capa de sesión proporciona sus servicios a la capa de presentación, proporcionando el medio necesario para que las entidades de presentación de dos hosts que se están comunicando por red organicen y sincronicen su diálogo y procedan al intercambio de datos. Sus principales funciones son:

- Establecer, administrar y finalizar las sesiones entre dos hosts (máquinas en red) que se están comunicando.
- Si por algún motivo una sesión falla por cualquier causa ajena al usuario, restaurar la sesión a partir de un punto seguro y sin pérdida de datos o, si esto no es posible, terminar la sesión de una manera ordenada, chequeando y recuperando todas sus funciones, evitando así problemas en sistemas transaccionales.
- Sincronizar el diálogo entre las capas de presentación de los dos hosts y administrar su intercambio de datos, estableciendo las reglas o protocolos para el diálogo entre máquinas, regulando quien habla y por cuánto tiempo.
- Conseguir una transferencia de datos eficiente y un registro de excepciones acerca de los problemas de la capa de sesión, presentación y aplicación.
- Manejar tokens, es decir aquellos objetos abstractos y únicos que se usan para controlar las acciones de los participantes en la comunicación, base de ciertos tipos de redes, como Token Ring o FDDI.
- Hacer checkpoints, que son puntos de recuerdo en la transferencia de datos, necesarios para la correcta recuperación de sesiones perdidas.

2.3.6 La capa de presentación

La capa de presentación proporciona sus servicios a la capa de aplicación, garantizando que la información que envía la capa de aplicación de un sistema pueda ser entendida y utilizada por la capa de aplicación de otro, estableciendo el contexto sintáctico del diálogo. Su tarea principal es aislar a las capas inferiores del formato de los datos de las aplicaciones específicas, transformando los formatos particulares (ASCII, EBCDIC, etc.) en un formato común de red, entendible por todos los sistemas y apto para ser enviado

por red. Es también la responsable de la obtención y de la liberalización de la conexión de sesión cuando existan varias alternativas disponibles. Para cumplir estas funciones, la capa de presentación realiza las siguientes operaciones:

- Traducir entre varios formatos de datos utilizando un formato común, estableciendo la sintaxis y la semántica de la información transmitida. Para ello convierte los datos desde el formato local al estándar de red y viceversa.
- Definir la estructura de los datos a transmitir. Por ejemplo, en el caso de un acceso a base de datos, definir el orden de transmisión y la estructura de los registros.
- Definir el código a usar para representar una cadena de caracteres (ASCII, EBCDIC, etc).
- Dar formato a la información para visualizarla o imprimirla; comprime los datos si es necesario.
- Aplicar a los datos procesos criptográficos cuando sea necesario.

2.3.7 La capa de aplicación

La capa de aplicación es la capa del modelo OSI más cercana al usuario, y está relacionada con las funciones de mas alto nivel, proporcionando soporte a las aplicaciones o actividades del sistema, suministrando servicios de red a las aplicaciones del usuario y definiendo los protocolos usados por las aplicaciones individuales. Es el medio por el cual los procesos y las aplicaciones de usuario acceden a la comunicación por red mediante el entorno OSI, proporcionando los procedimientos precisos para ello. Los procesos de las aplicaciones se comunican entre sí por medio de entidades de aplicación propias, estando controladas por protocolos específicos de la capa de aplicación, que a su vez utilizan los servicios de la capa de presentación, situada inmediatamente debajo en el modelo.

Difiere de las demás capas debido a que no proporciona servicios a ninguna otra capa OSI, sino solamente a aplicaciones que se encuentran fuera del modelo (procesadores de texto, hojas de cálculo, navegadores web, etc.).

La capa de aplicación establece la disponibilidad de los diversos elementos que deben participar en la comunicación, sincroniza las aplicaciones que cooperan entre sí y establece acuerdos sobre los procedimientos de recuperación de errores y control de la integridad de los datos.

2.3 Descripción del protocolo FBUS

En los teléfonos celulares Nokia, la comunicación entre la computadora y el teléfono se realiza mediante protocolos especificados por el fabricante. Así se tiene el MBUS versión 1, MBUS versión 2, el FBUS versión 1 y el FBUS versión 2/Direct IRDA [4].

Las aplicaciones del protocolo MBUS se encuentran orientadas al servicio técnico, mientras que el protocolo FBUS se orienta en aplicaciones extendidas. El protocolo FBUS ha sido incluido en una gran cantidad de teléfonos de la marca Nokia (6110, 6130, 6150, 6190, 5110, 5130, 5150, 5190, 3210, 3310, 3330, 3390, 3391, 3395, etc) para permitir al usuario interactuar con su teléfono de una manera amigable a través de las aplicaciones denominadas “Data Suites” las cuales son conocidas también como aplicaciones de usuario.

El protocolo FBUS versión 2 permite en la capa física la comunicación entre el teléfono celular y la computadora por medio de un puerto serial (RS-232) así como también por medio de un puerto infrarrojo (solo si este puerto se encuentra disponible en el modelo de teléfono). La figura 2.9 muestra la interacción de las tramas entre el teléfono celular y la computadora en el nivel de enlace, para el ejemplo, el solicitante de la información es A (computadora) la cual solicita la información a B (teléfono celular).

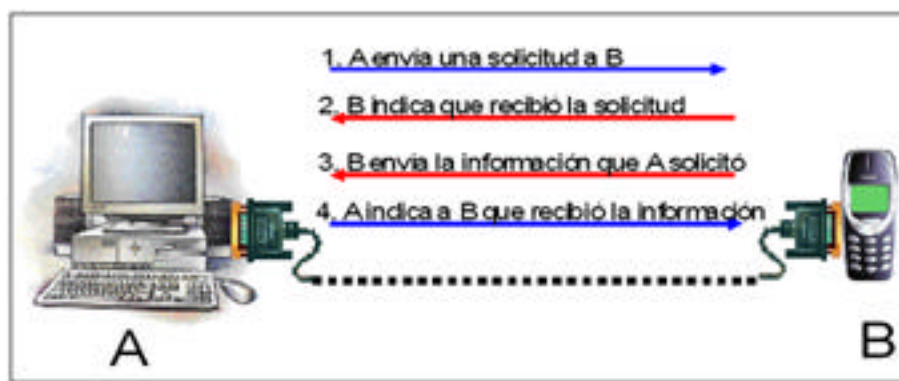


Figura 2.9. Detalles de la comunicación entre el teléfono celular y la computadora

La estructura de las tramas del protocolo FBUS versión 2 en el nivel de enlace es la que se indica en la figura 2.10.

FrameID	DestDEV	SrcDEV	MsgType	00h	FrameLength	{block}
FramesToGo		SeqNo	PaddingByte	ChkSum1	ChkSum2	

Figura 2.10. Estructura del protocolo FBUS Versión 2

En donde:

FrameID:	1Ch: IR / FBUS (Infrarrojo) 1Eh: Serial / FBUS
DestDev,SrcDev:	00h: Teléfono celular 0Ch: Pc.
MsgType:	Tipo de mensaje. Se refiere al tipo de solicitud que se está realizando o recibiendo. En el siguiente capítulo, se detallará los aspectos relacionados a este y los demás campos de las tramas FBUS.
00h:	Este valor siempre se mantiene constante.
FrameLength:	Es la longitud de los datos útiles. Su valor corresponde a la longitud del campo {Block }+2 bytes debido a que este se incluye la longitud del bloque de datos más los campos FramesToGo y SeqNo.
{block}:	Bloque de datos. Contiene los datos útiles de la trama. El valor en Hexagesimal de los dos primeros bytes de este campo siempre serán: 00 01 : Cuando se envía información al teléfono. 01 08 : Cuando se recibe información del teléfono. Los siguiente bytes dentro del bloque de datos corresponden a la información útil contenida entro de la trama.

FramesToGo: Es la cantidad de tramas por enviar. Por lo general es sólo una y tendrá el valor 01h pero en casos cuando la información a transmitir sea numerosa será un valor mayor.

SeqNo: [XYh]

X: 4 : Al enviar una sola trama o la primera de varias.

0 : Al enviar una segunda trama.

Y: Secuencia; va de 0 a 7.

PaddingByte: Su función es hacer que la cantidad de bytes pares y la cantidad de bytes impares sea igual. Se utiliza adicionando el valor 00h para lograr dicha igualdad. Si la cantidad de bytes pares e impares ya es igual, entonces este elemento ya no se presenta en la trama.

ChkSum1: XOR de todos los bytes impares de la trama.

ChkSum2: XOR de todos los bytes pares de la trama.

Para el caso del acuse de envió en el protocolo FBUS versión 2 (Acknowledge), la estructura de la trama es la que se muestra en la figura 2.11.

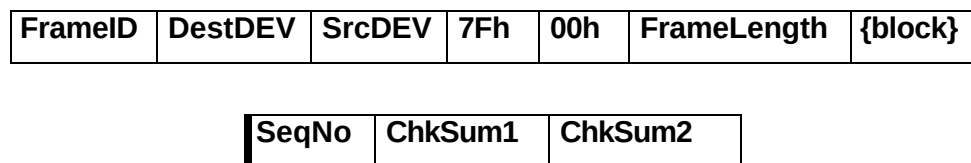


Figura 2.11. Trama de Acknowledge en la versión 2 del protocolo FBUS

Donde:

FrameLength: Siempre es 02h

{Block}: Tipo de mensaje que se recibió.

SeqNo: [0Yh]

Y: Secuencia; va de 0 a 7

Las tramas en estos protocolos tienen una longitud máxima de 130 bytes. Si durante un tiempo de medio segundo el teléfono celular no recibe la confirmación de que la información transmitida fue recibida (Acknowledge o acuse de envío), éste vuelve a enviarla, el proceso se repite sólo 3 veces y luego deja de transmitir la respuesta asumiendo que hay un error en la comunicación [5].

2.4 El servicio de mensajes de texto cortos (SMS)

En este ítem se pretende hacer un estudio general sobre la forma cómo trabaja una red SMS a fin de entender el proceso que se realiza en la red cuando una persona envía un mensaje de texto. No se profundizará en este tema debido a que escapa a los objetivos del trabajo de tesis.

El servicio SMS permite transferir un mensaje de texto entre una estación móvil (MS) y otra entidad (SME) a través de un centro de servicio (SC) tal como se muestra en la figura 2.12. El servicio final ofrecido es una comunicación extremo - extremo entre la estación móvil (mobil station - MS) y otra entidad (station mobil entiti - SME) a través de un centro de servicio (SC). La entidad puede ser otra estación móvil o puede estar situado en una red fija. En el caso de envío de un mensaje entre dos móviles, ambas partes son estaciones móviles [6].

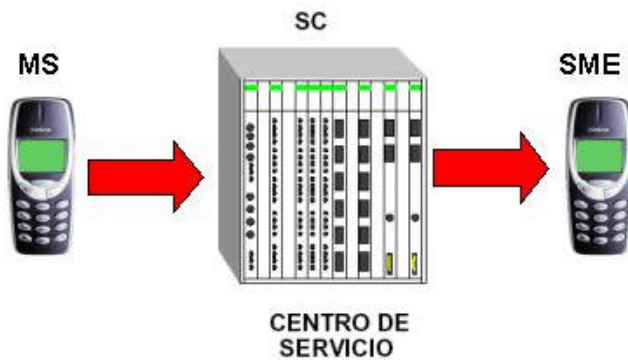


Figura 2.12. Esquema del servicio SMS

2.5.1 Arquitectura de red para el servicio SMS

La arquitectura básica de una red para el servicio SMS consta de diversas entidades que permiten el correcto envío y recepción de los mensajes e texto así como también el registro de los eventos producidos en la red SMS. Dichas entidades son las siguientes:

- MS (Mobil Station) - Estación móvil, la cual está formada por el teléfono celular y por la tarjeta SIM (Subscriber Identity Module), la cual permite identificar al abonado independientemente del teléfono celular utilizado.
- MSC (Mobile Switching Center) - Centro de conmutación móvil, el cual se encarga de administrar el enrutamiento de las llamadas dentro de la red. También controla los accesos a ciertas características de los sistemas y accesos a las bases de datos de la red. El MSC también se encarga de coordinar los cambios de una celda a otra, también envía alertas a los teléfonos móviles, registra los momentos en que cada celular es encendido y administra las conexiones con la red.
- SMS-GMSC (SMS Gateway/Interworking Mobile Switching Center) - Centro de Conmutación Móvil SMS. Es un centro de conmutación de mensajes encargado de recibir SMS del SMSC, interrogar al registro de localización local por la información de encaminamiento, y entregarlo al MSC que da servicio a la estación móvil.
- SMS-IWMSC (Interworking MSC for Short Message Service) - MSC de interconexión entre PLMN (Public Land Mobile Network) y el SC.
- SC (Service Center) - Centro de servicio.
- HLR (Home Location Register). Registro de localización de usuarios domésticos.
- VLR (Visitor Location Register). Registro de localización del visitante.

La interrelación entre estas entidades se aprecia en la figura 2.13.

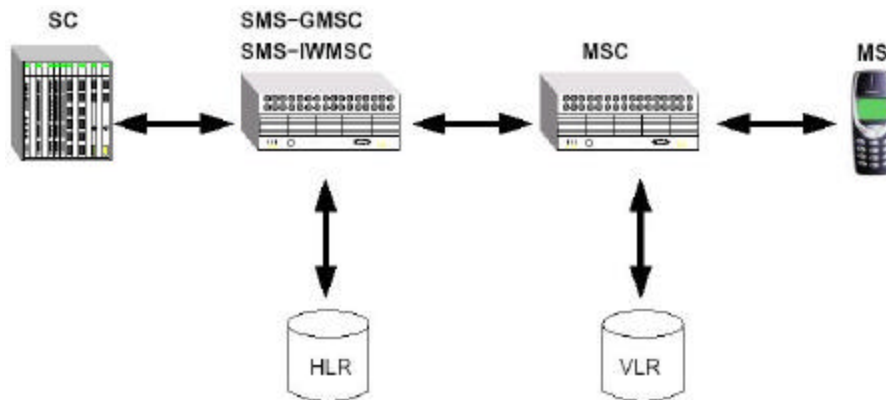


Figura 2.13. Arquitectura de red para el servicio SMS

Para la descripción detallada de la arquitectura, se utiliza un modelo de capas, en el que cada capa o nivel proporciona un servicio a la capa superior, el servicio SMS se implementa mediante el protocolo correspondiente. La arquitectura se divide en 4 capas las cuales se indican en la figura 2.14.

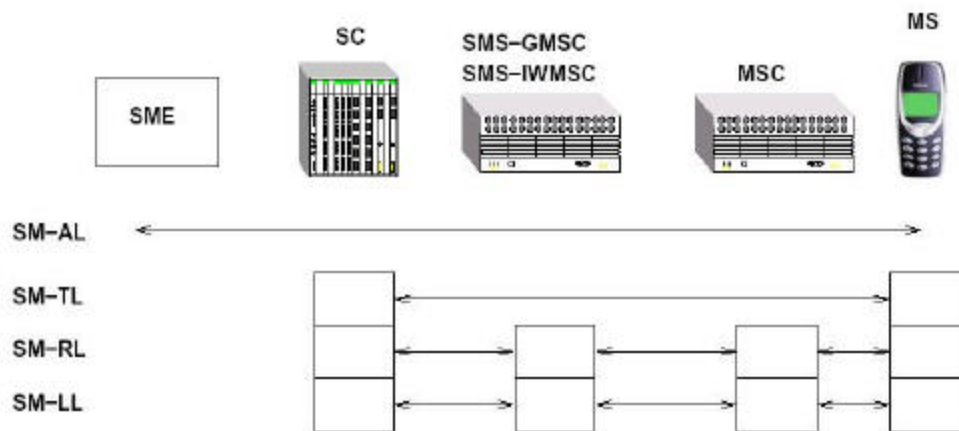


Figura 2.14. Arquitectura de capas para el servicio SMS

Las cuales son:

- SM-AL (Short Message Application Layer): Capa de aplicación. Envía mensajes de texto cortos entre los diversos usuarios de la red.
- SM-TL (Short Message Transfer Layer): Capa de transferencia. Servicio de transferencia de un mensaje corto entre una MS y un SC (en ambos sentidos) y obtención de los correspondientes informes sobre el resultado de la transmisión.

Este servicio hace abstracción de los detalles internos de la red, permitiendo que la capa de aplicación pueda intercambiar mensajes.

- SM-RL (Short Message Relay Layer): Capa de repetición. Proporciona un servicio a la capa de transferencia que le permite enviar TPDU (Transfer Protocol Data Units) a su entidad gemela.
- SM - LL (Short Message Lower Layers): Capas inferiores.

2.4.2 Protocolo SM - TP

El protocolo SM – TP trabaja a nivel de la capa SM - TL de la red SMS, este protocolo mediante sus PDUs se encarga de controlar el tráfico del servicio SMS enviando y recibiendo los mensajes de texto así como también informando el estado de entrega de los mismos. Consta de 6 PDUs las cuales son:

- SMS - DELIVER: Transmitir un mensaje desde el SC al MS.
- SMS - DELIVER-REPORT: Error en la entrega (si es que sucede).
- SMS - SUBMIT: Trasmistir un mensaje corto desde el MS al SC.
- SMS - SUBMIT-REPORT: Error en la transmisión (si es que sucede).
- SMS - STATUS-REPORT: Transmitir un informe de estado desde el SC al MS.
- SMS - COMMAND: Transmitir un comando desde el MS al SC.

La PDU SMS – SUBMINT es la que se encarga de transmitir el mensaje de texto en la red. En ella se encuentran todos los datos que debe contener el mensaje para poder llegar a su destino. La estructura de las tramas de esta PDU es como se muestra en la figura 2.15.

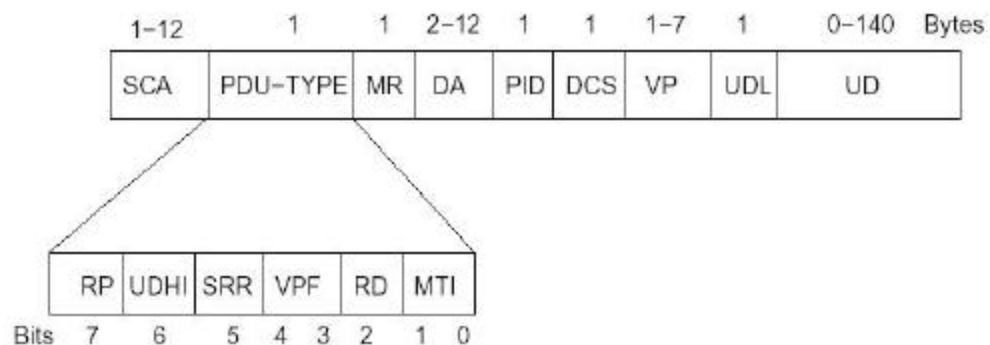


Figura 2.15. Estructura de la trama SMS - SUBMINT

Los campos que componen esta trama son los siguientes:

- **SCA** : Número de teléfono del Centro de Servicio (SC). Consta de los campos longitud (número de dígitos del teléfono del SC), tipo de número (81h Nacional, 91h Internacional), Dígitos BCD (número de teléfono del SC, en dígitos BCD).
- **PDU - TYPE**: Contiene información sobre el tipo de PDU, consta de los campos RP, que indica si existe camino de respuesta, UDHI, que indica si el campo UD contiene sólo el mensaje corto (UDHI=0) o si existe una cabecera antes del mensaje corto (UDHI=1), SRR, que solicita informes de estado, VPF que indica si el campo VP está o no presente, RD, que rechaza o no datos duplicados, MTI, que es el tipo de mensaje.
- **MR** : Parámetro que identifica el mensaje.
- **DA** : Dirección del SME destino (número de tlf).
- **PID** : Identifica el protocolo de la capa superior.
- **DCS** : Identifica el tipo de codificación dentro de los datos de usuario.
- **VP** : Periodo de validez del mensaje.
- **UDL** : Longitud del campo UD.
- **UD** : Datos de usuario.

2.4.3 Codificación y decodificación de caracteres para SMS

La codificación y decodificación de caracteres para el envío y recepción de mensajes de texto son tareas correspondientes a la capa de presentación del protocolo FBUS. Estos aspectos se encuentran normados por la ETSI; en la publicación referenciada [7], se establece la forma como se deben codificar y decodificar los mensajes de texto en los sistemas GSM mediante un procedimiento para empaquetarlos empleando 7 bits por carácter.

La ETSI ha codificado los caracteres de manera que los caracteres básicos coincidan con los del código ASCII, mientras que los caracteres alfanuméricos que se encuentran entre el rango del 128 y 255 del código ASCII, se encuentran ordenados en posiciones que

corresponden a caracteres no imprimibles del código ASCII. La tabla 2.4 muestra los caracteres soportados por el servicio de mensajes de texto así como su valor.

TABLA 2.4
CARACTERES DISPONIBLES EN EL SERVICIO SMS

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0	i	P	¿	p
0	0	0	1	1	£	_	!	1	A	Q	a	q
0	0	1	0	2	\$	Φ	"	2	B	R	b	r
0	0	1	1	3	¥	Γ	#	3	C	S	c	s
0	1	0	0	4	è	Λ	□	4	D	T	d	t
0	1	0	1	5	é	Ω	%	5	E	U	e	u
0	1	1	0	6	ù	Π	&	6	F	V	f	v
0	1	1	1	7	ì	Ψ	'	7	G	W	g	w
1	0	0	0	8	ò	Σ	(	8	H	X	h	x
1	0	0	1	9	Ç	Θ	)	9	I	Y	i	y
1	0	1	0	10	LF	Ξ	*	:	J	Z	j	z
1	0	1	1	11	Ø	1)	+	;	K	Ä	k	ä
1	1	0	0	12	ø	Æ	,	<	L	Ö	l	ö
1	1	0	1	13	CR	æ	-	=	M	Ñ	m	ñ
1	1	1	0	14	Å	ß	.	>	N	Ü	n	ü
1	1	1	1	15	å	É	/	?	O	Š	o	à

2.5.3.1 Codificación

El proceso de codificación de caracteres se muestra realizando como ejemplo la codificación de la palabra HOLA.

H O L A Valor de entrada:

ASC		Hexa		Binario
H	=	48	=	1001000
O	=	4F	=	1001111
L	=	4C	=	1001100
A	=	41	=	1000001

Los 7 bits obtenidos por cada caracter son ordenados de menos significativo a más significativo siendo agrupados de 4 en 4; si la agrupación de 4 bits no es completada, la posición vacía será donde se empiece a ordenar el siguiente carácter de la siguiente manera:

1a 1b 1c 1d 1e 1f 1g 2a 2b 2c 2d 2e 2f 2g 3a 3b ...

donde 1a, 1b, 1c, 1d, 1e, 1f y 1g son los bits del primer carácter, 2a, 2b, 2c, 2d, 2e, 2f, 2g el segundo y así sucesivamente

Se asignará una letra a cada grupo de 4 bits para explicar mejor el ejemplo.

A1	A2	B1	B2	C1	C2	D1
0001	0011	1110	0100	1100	1100	0001

La cantidad de bits debe ser múltiplo de 8, por lo tanto, los bits faltantes se completan con 0:

A1	A2	B1	B2	C1	C2	D1	D2
0001	0011	1110	0100	1100	1100	0001	0000

Se invierte A2 con A1, B2 con B1 y sucesivamente para los demás valores:

0011 0001 0100 1110 1100 1100 0000 0001

Las agrupaciones son codificadas utilizando BCD comenzando por el dígito menos significativo de acuerdo a los siguientes valores:

0000 = "0" 1000 = "1" 0100 = "2" 1100 = "3" 0010 = "4" 1010 = "5" 0110 = "6" 1110 = "7" 0001 = "8" 1001 = "9" 0101 = "A" 1101 = "B" 0011 = "C" 1011 = "D" 0111 = "E" 1111 = "F"

De donde C 8 2 7 3 3 0 8 son los valores obtenidos cada cuatro bits, los cuales deben ser agrupados de 2 en 2, obteniéndose **C8 27 33 08**, que es el resultado de codificar la palabra HOLA para su envío por mensaje de texto.

Se debe tener en cuenta la relación existente entre la cantidad de bytes antes de realizar la codificación y la cantidad de bytes generados después de la codificación.

Una regla práctica para realizar esta conversión es aplicar la siguiente formula:

$$\# \text{ Caracteres (7Bits)} = \frac{16}{14} \times \# \text{ Caracteres (8Bits)}$$

El valor obtenido debe ser redondeado, al número entero inmediato inferior. Ejemplos:

Para el caso de la palabra HOLA que consta de 4 bytes, se desea saber la cantidad de caracteres generados al realizar la codificación, se aplica la formula y se tiene:

$$\# \text{ Caracteres (7Bits)} = \frac{16}{14} \times 4 = 4.5714$$

Se redondea el valor obtenido al número entero inmediato inferior, y se obtiene 4, que es la cantidad de caracteres generados luego del proceso de codificación.

Si se codifican 17 bytes y se desea saber cual es la cantidad de caracteres que se generan luego de la conversión, entonces:

$$\# \text{ Caracteres (7Bits)} = \frac{16}{14} \times 17 = 19.4285$$

Se redondea el valor obtenido al número entero inmediato inferior, y se obtiene 19, que es la cantidad de caracteres generados luego del proceso de codificación.

2.5.3.2 Decodificación

Se realiza el proceso inverso del caso mencionado en el ítem 2.4.4.1 para decodificar el mensaje contenido en la cadena C8 27 33 08 (HOLA).

C8 27 33 08 Cadena Hexagesimal

Se codifica cada dígito usando BCD comenzando por el dígito menos significativo de acuerdo a los siguientes valores:

0000 = "0" 1000 = "1" 0100 = "2" 1100 = "3" 0010 = "4" 1010 = "5" 0110 = "6" 1110 = "7"
0001 = "8" 1001 = "9" 0101 = "A" 1101 = "B" 0011 = "C" 1011 = "D" 0111 = "E"
1111 = "F"

de donde se obtiene:

A1 A2 B1 B2 C1 C2 D1 D2
0011 0001 0100 1110 1100 1100 0000 0001

Se invierte A2 con A1, B2 con B1 y sucesivamente para los demás valores:

0001 0011 1110 0100 1100 1100 0001 0000

Se agrupan los bits de 7 en 7:

0001001

1111001

0011001

1000001

0000, se descarta los ceros que quedan al final.

Los bits agrupados de 7 en 7 son invertidos y convertidos a caracteres según la tabla 2.2.

1001000	=	48	=	H
1001111	=	4F	=	O
1001100	=	4C	=	L
1000001	=	41	=	A

HOLA es la decodificación de C8 27 33 08.

Si se desea saber el número de bytes que se generan luego del proceso de decodificación de los datos, se debe aplicar:

$$\# \text{ Caracteres (8Bits)} = \frac{14}{16} \times \# \text{ Caracteres (7Bits)}$$

El resultado debe ser redondeado al número entero inmediato superior. Ejemplos:

Para el caso de C8 27 33 08, que representa 4 bytes de caracteres codificados a 7 bits.
Para saber la cantidad de bytes generados luego de la codificación se tiene:

$$\# \text{ Caracteres (8Bits)} = \frac{14}{16} \times 4 = 3.5$$

Se procede a redondear el valor 3.5 al número entero inmediato superior, obteniéndose 4, que es la cantidad de bytes que se generan luego del proceso de decodificación.

Si se desea saber cual es la cantidad de bytes que se generan luego de decodificar 22 caracteres codificados a 7 bytes, se hará:

$$\# \text{ Caracteres (8Bits)} = \frac{14}{16} \times 22 = 19.25$$

Se procede a redondear el valor 19.25 al número entero inmediato superior obteniéndose, 20, que es la cantidad de bytes que se generan luego del proceso de decodificación.

III. ANÁLISIS DEL PROTOCOLO FBUS

Debido a la carencia de literatura, se buscó información relativa a la comunicación entre la computadora y el teléfono celular, sin embargo, la información encontrada no fue muy confiable debido a que existían muchas contradicciones.

Es así que este capítulo tiene por finalidad describir la manera como se realizó el análisis del protocolo, las herramientas de hardware y software que fueron necesarias crear y/o utilizar para describir los detalles de la comunicación así como complementar la información encontrada.

Las herramientas de hardware utilizadas fueron:

- Hardware de interfaz para la serie Nokia 33XX (implementado en la Tesis).
- Hardware canalizador de la comunicación (diseñado en la Tesis).

Las Herramientas de Software fueron:

- Software SMS Manager (disponible en Internet).
- Software para captura de datos seriales (diseñado en la Tesis).
- Software de envío de tramas hacia el teléfono celular (diseñado en la Tesis).
- Software codificador y decodificador de caracteres para SMS (diseñado en la Tesis).

3.1 Herramientas de análisis

En este ítem se describen las herramientas empleadas en el análisis del protocolo. Para esto, se tuvo que conseguir algún programa que realice la comunicación entre la computadora y el teléfono celular, y desarrollar herramientas que fueron indispensables para registrar, canalizar, ordenar, interpretar y estudiar las tramas involucradas en la comunicación.

3.1.1 Hardware de interfaz para la serie Nokia 33XX

Esta herramienta tiene como finalidad interconectar al teléfono celular con la computadora. Para esto, es necesario disponer de un circuito de interfaz que acondicione los niveles de voltaje del puerto serial a valores no mayores a 3 voltios, para evitar daños en el teléfono. Luego de analizar los circuitos disponibles en [9], se optó por implementar el circuito mostrado en la figura 3.1 obteniéndose resultados satisfactorios.

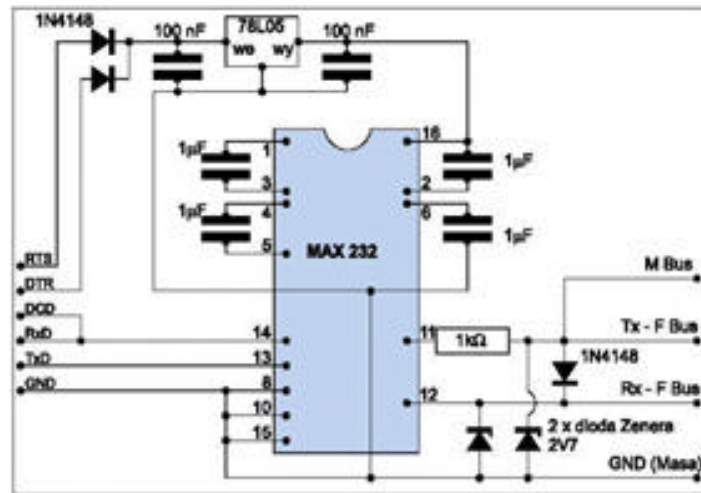


Figura 3.1. Diagrama del hardware de interfaz para el teléfono celular Nokia 3390

Existe también interfaces entre la computadora y el teléfono celular utilizadas en servicio técnico, estos interfaces presentan puertos MBUS y FBUS. La figura 3.2 muestra la interfaz de servicio de Nokia (BBS-8:0630261) [10].



Figura 3.2. Interfaz BBS-8:0630261

3.1.2 Hardware canalizador de la comunicación

Esta herramienta tiene como finalidad, combinar las señales de transmisión y recepción del puerto serial para que ambas se dirijan en un mismo sentido de transmisión hacia otro puerto serial con el fin de ser registradas o grabadas. El circuito diseñado para este fin es el que se muestra en la figura 3.3.

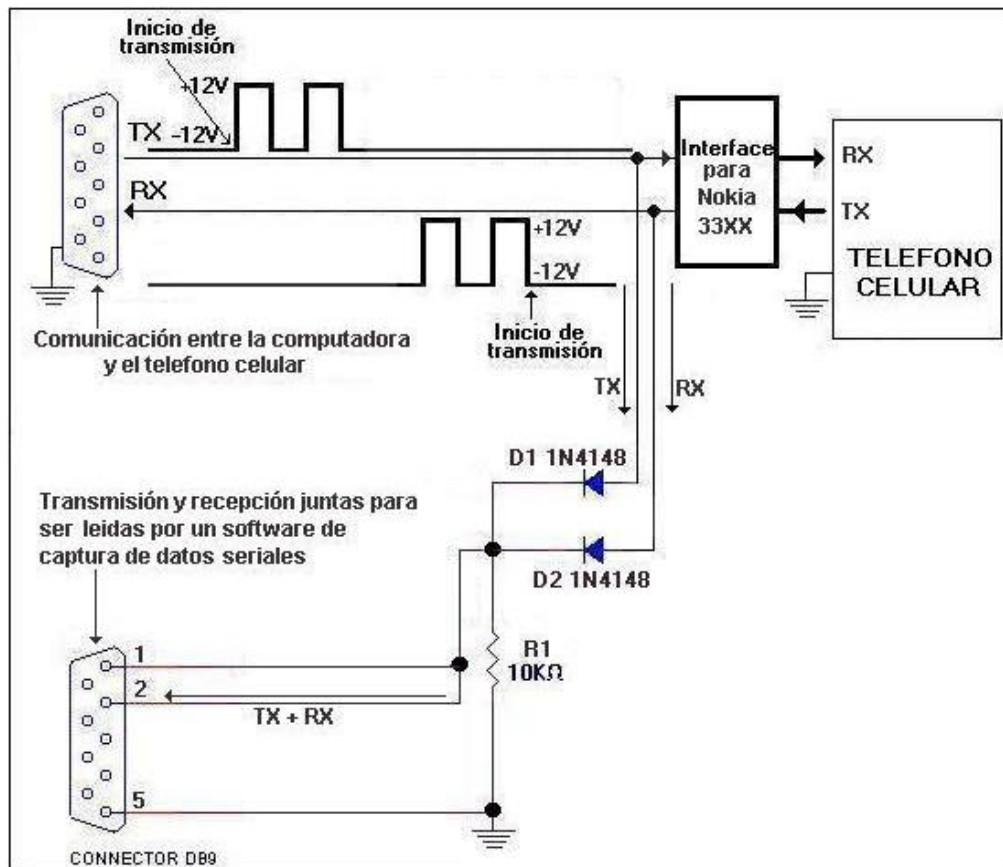


Figura 3.3. Circuito canalizador de la comunicación

3.1.3 Software SMS Manager

El programa SMS Manager [11], cuya ventana principal se muestra en la figura 3.4 realiza la comunicación entre el teléfono celular y la computadora permitiendo al usuario administrar sus mensajes de texto y directorio telefónico. Fue empleado para establecer una comunicación entre la computadora y el teléfono celular a fin de ser grabada para ser analizada.

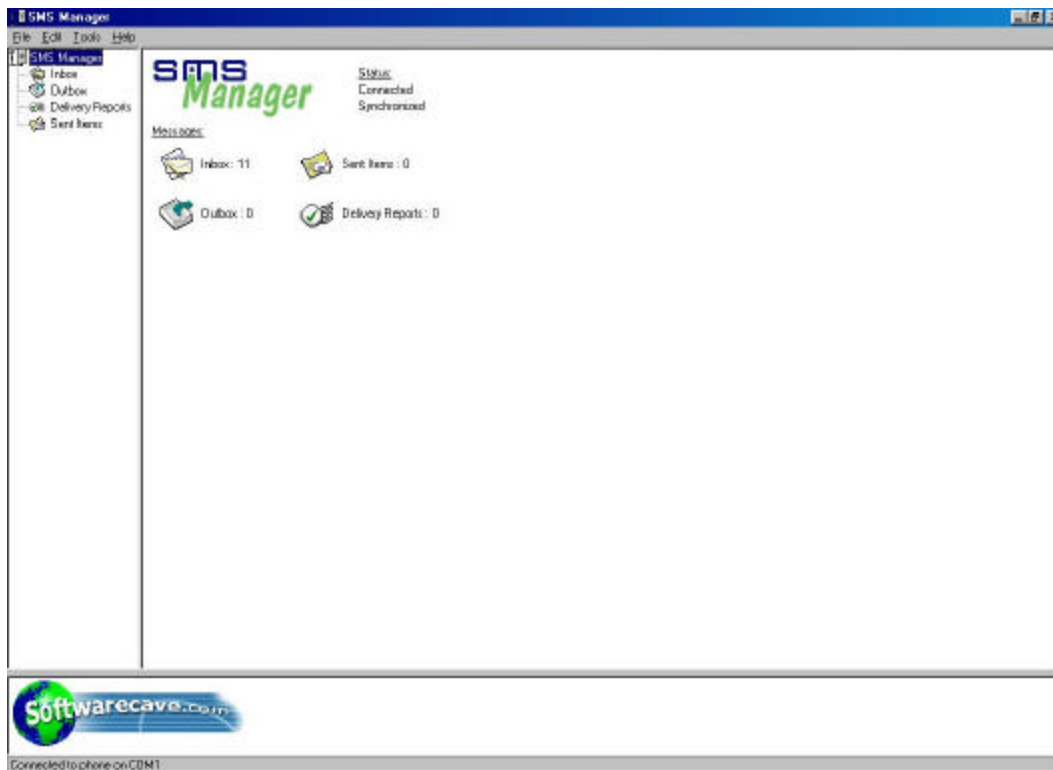


Figura 3.4. Ventana principal del programa SMS Manager

El Software está dividido en dos secciones:

- Sección izquierda, donde se tiene las siguientes herramientas:
Inbox (lista de mensajes recibidos), *OutBox* (lista de mensajes por enviar), *Delivery reports* (lista de reportes de los mensajes enviados), *Send Items* (lista de mensajes enviados).
- Sección derecha, donde se muestra el resumen de la cantidad de mensajes recibidos, por enviar, etc.

Las funciones que puede realizar este programa se encuentran agrupadas dentro de una barra de menús desplegable que contiene lo siguiente:

- *File*, que contiene las siguientes funciones:
New SMS Message (enviar mensaje), *Connect to Pone* (conectarse al teléfono), *Disconnect form Pone* (desconectarse del teléfono), *Connection Parameters* (configurar parámetros de la comunicación), *Exit* (salir del programa).
- *Edit*, que contiene las siguientes funciones:

Mark as Read (marcar mensaje como leído), *Mark as Unread* (marcar mensaje de texto como no leído), *Delete* (borrar mensaje en el programa), *Remove from Phone* (borrar mensaje en teléfono), *Remove all from Phone* (borrar todos los mensajes en el teléfono).

- *Tools*, que contiene la función:
Synchronize All (sincronizar la comunicación)

3.1.4 Software para captura de datos seriales

Este software fue diseñado e implementado con la finalidad de grabar en un archivo de registro toda la información que se reciba por el puerto serial. El programa de captura de datos cuya ventana se muestra en la figura 3.5, lee un determinado puerto serial, convierte los bytes recibidos a su equivalente en Hexagesimal y código ASCII para visualizar los valores que conforman la trama recibida en una forma que sea fácil de analizar.

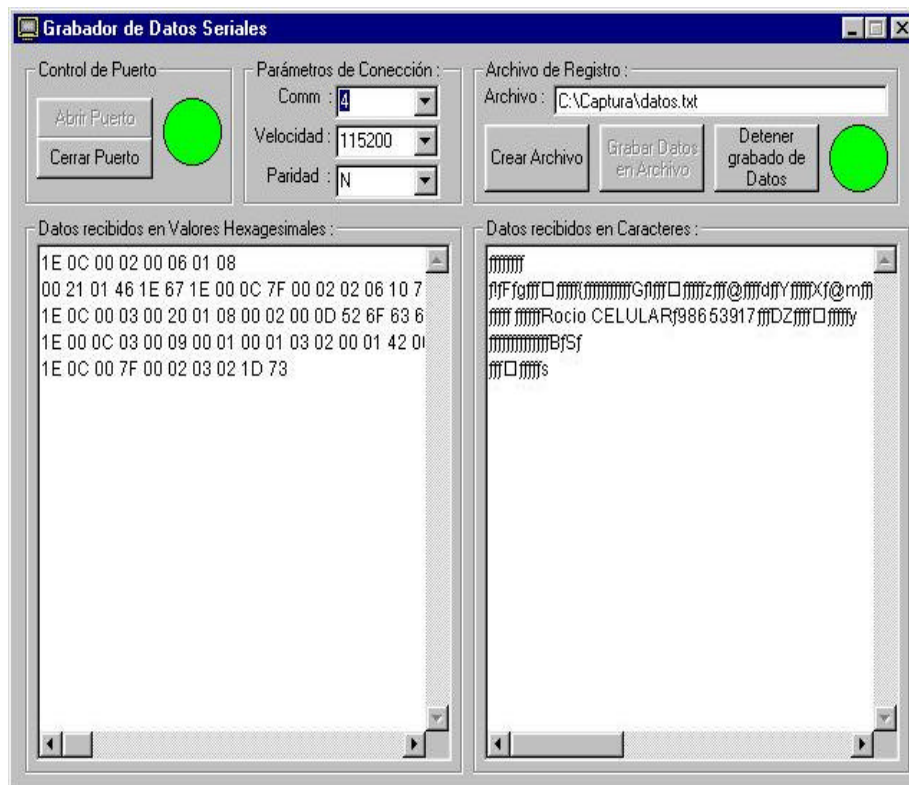


Figura 3.5. Ventana del software de captura para datos seriales

El programa ha sido desarrollado en Visual Basic 6.0 y su componente más importante es el objeto MSComm que permite leer los datos del puerto serial de la computadora. En este objeto, al configurar la propiedad RTHReshold al valor de 1, es posible detectar la interrupción del puerto serial para que el programa inmediatamente atienda los datos recibidos.

En la figura 3.6, muestra el diagrama de flujo del software de captura de datos seriales, donde se puede apreciar que al momento de recibirse datos por el puerto serial, el programa ejecuta a una subrutina que permite grabar los datos en el archivo de registro, terminando este proceso, el programa continua esperando la llegada de nuevos datos.

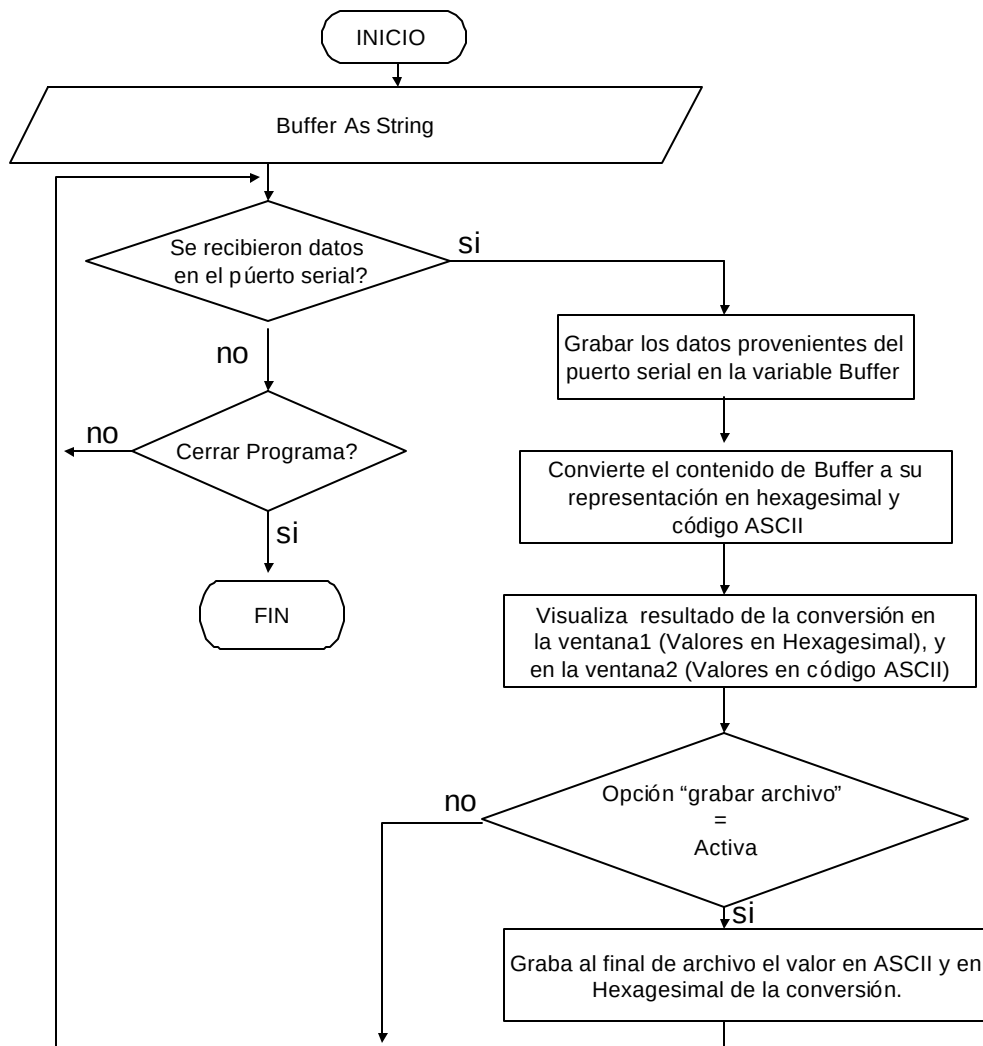


Figura 3.6. Diagrama de flujo del programa Grabador de Datos seriales

El software de captura de datos consta de las siguientes secciones:

- **Control de Puerto**
 - Abrir puerto.
 - Cerrar puerto.
 - Estado de actividad del puerto (verde = abierto, rojo = cerrado).

- **Parámetros de Conexión.-**
 - Comm.- Puerto serial utilizado (COM1, COM2, COM3 o COM4).
 - Velocidad.- Velocidad de la comunicación.
 - Paridad.- Tipo de paridad a utilizar (Par, Impar, Ninguna).

- **Archivo de Registro.-**
 - Ruta del archivo.
 - Crear archivo.
 - Grabar datos en archivo.
 - Detener grabado de datos.
 - Indicador de grabación de datos (verde = grabando, rojo = detenido).

- **Visualización**
 - Datos recibidos en valores Hexagesimal.
 - Datos recibidos en código ASCII.

3.1.5 Software de envío de tramas hacia el teléfono celular

Este software fue implementado con la finalidad de enviar tramas hacia el teléfono para luego visualizar la trama de respuesta obtenida.

En la figura 3.7 se puede apreciar el software para enviar tramas. Al iniciar el programa, se debe introducir una trama sin Checksums, mediante el botón *Agregar Checksums* se calcula y agrega estos campos a la trama por enviar. Al presionar el botón *Enviar Trama*, la información es enviada hacia el teléfono visualizándose a continuación la respuesta.

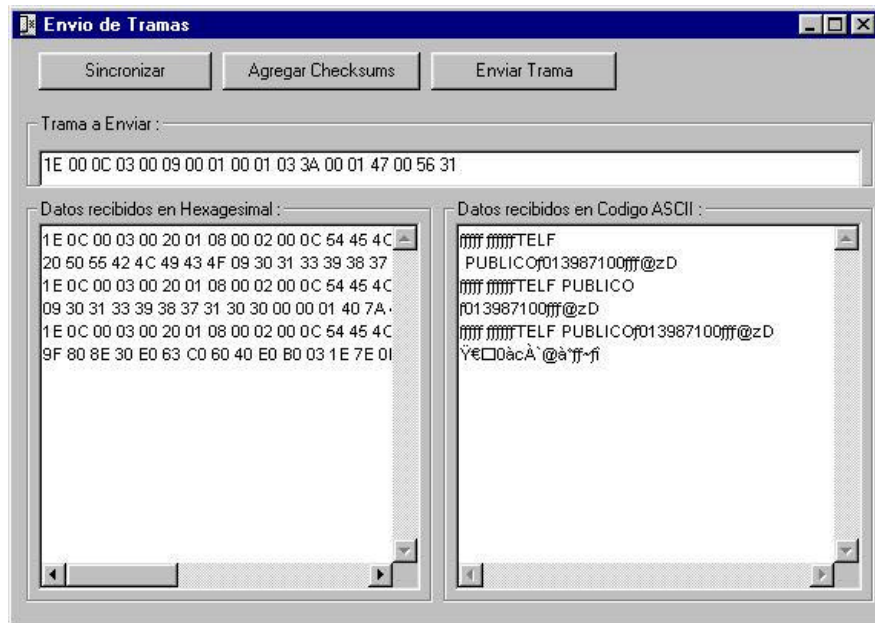


Figura 3.7. Ventana del programa para enviar tramas al teléfono celular

3.1.6 Software codificador y decodificador de caracteres para SMS

Este software fue implementado con la finalidad de codificar los mensajes de texto antes de ser enviados al teléfono así como también para decodificar los mensajes contenidos en las tramas de respuesta. Los procedimientos para codificar y decodificar los mensajes de texto se encuentran descritos en los ítems 2.5.3.1 y 2.5.3.2. La figura 3.8 muestra el funcionamiento de este software.

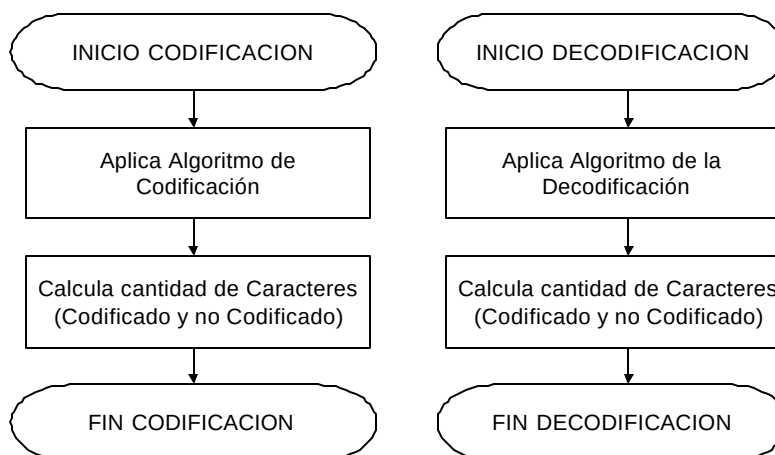


Figura 3.8. Diagrama de flujo para la codificación y decodificación de caracteres

En la figura 3.9 se aprecia el software para codificar y decodificar los mensajes de texto. Para realizar una codificación, se ingresa el mensaje en la ventana Caracteres del Mensaje de Texto, a continuación se presiona el botón *Codifica*. El resultado de la codificación se visualizará en la ventana Valores Codificados. El proceso inverso también es posible haciendo uso del botón *Decodifica*.

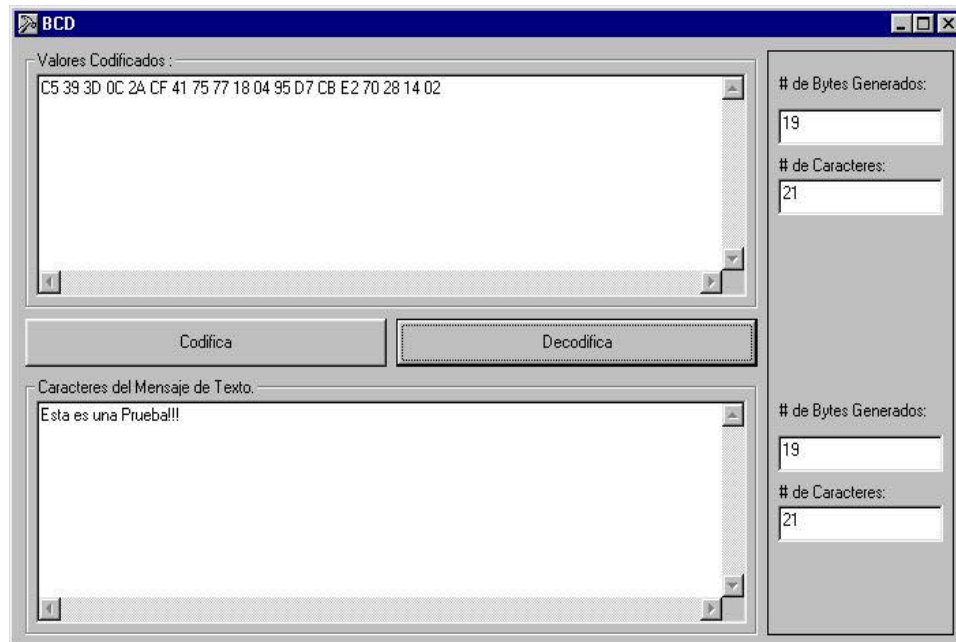


Figura 3.9. Ventana del programa codificador y decodificador de caracteres

3.2 Identificación de las tramas en la comunicación

El objetivo de este análisis, fue identificar las tramas de la comunicación a fin de verificar y validar la insuficiente información encontrada, se optó por grabar la comunicación entre algún programa que haga uso del protocolo FBUS cuando se comunique con la computadora, se eligió para este propósito el software SMS Manager que como fue descrito en el ítem 3.1.3, es gerenciador de mensajes de texto y del directorio telefónico.

Para grabar las tramas de la comunicación entre el teléfono y la computadora se realizó el procedimiento indicado en la figura 3.10:

- Se empleó dos computadoras PC-A y PC-B, el teléfono celular, el hardware de interfaz y el hardware canalizador de la comunicación.

- PC-A ejecuta el software SMS manager para comunicarse con el teléfono celular. PC-A y el teléfono se encuentran conectados mediante el hardware de interfaz descrito en el ítem 3.1.1.
- PC-B ejecuta el software para captura de datos seriales descrito en el ítem 3.1.4 y su función fue la de grabar los bits generados en la comunicación entre el teléfono y PC-A. Para interconectar esta computadora a la línea RS232, por donde fluía la comunicación entre el teléfono y PC-A, se empleó el circuito canalizador de la comunicación descrito en el ítem 3.1.2.

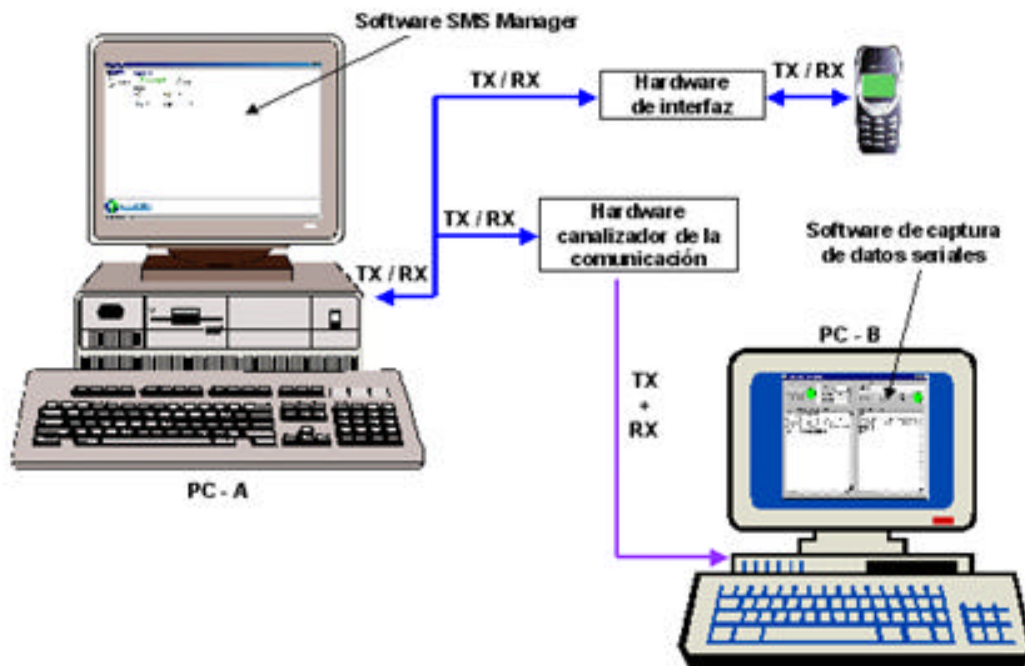


Figura 3.10. Obtención de los datos de la comunicación.

Mediante este procedimiento, se obtuvo tramas reales producto de la comunicación entre la computadora y el software SMS Manager las cuales fueron impresas, analizadas y comparadas con la información obtenida de las fuentes bibliográficas, permitiendo validar la información útil y descartar la información errónea por medio de comparaciones. Así mismo, también se descubrió detalles propios de la comunicación que no eran mencionados en las fuentes bibliográficas.

3.2.1 Conclusiones de la identificación de las tramas

- Del análisis de los datos grabados, se pudo deducir que para iniciar y sincronizar la comunicación se debe utilizar una cadena de 128 bytes con el valor 55h.
- Comparando la estructura de la trama FBUS y las observadas de la grabación, se puede deducir la tabla 3.1

TABLA 3.1
TRAMA OBTENIDA DEL REGISTRO DE LA COMUNICACIÓN

Byte :	00	01	02	03	04	05	06	07	08	09	10	11	12	13
Datos :	1E	00	0C	04	00	06	00	01	00	01	01	40	13	42

Donde:

Byte 00	:	FrameID.
Byte 01	:	DestDEV.
Byte 02	:	SrcDEV.
Byte 03	:	MsgType.
Byte 04	:	00h.
Byte 05	:	FrameLength.
Byte 06 al 09	:	{block}.
Byte 10	:	FramesToGo.
Byte 11	:	SeqNo.
Byte 12	:	ChkSum1.
Byte 13	:	ChkSum2.

Por lo cual la estructura de las tramas FBUS indicada en el ítem 2.3 coincide con las tramas capturadas.

- El valor para el campo FrameID siempre es igual a 1E (comunicación FBUS)
- Para los campos DestDEV y SrcDEV (fuente y destino de la comunicación) el valor 00h corresponde al teléfono celular mientras que 0Ch corresponde a la computadora.

- El valor del campo MsgType siempre consta de un solo byte y su valor depende del tipo de solicitud que se realice o se atienda. El valor de este campo corresponde a los indicados en la tabla 3.2

TABLA 3.2

FUNCIONES DEL TELÉFONO CELULAR AGRUPADOS POR TIPOS DE MENSAJE

Tipo de Mensaje	Función
02h: Manipulación de mensajes de texto	Solicitar envío de mensaje de texto Mensaje de texto enviado sin error Envío de Mensaje de texto fallido Mensaje de texto nuevo recibido. Leer mensaje de texto recibido Obtener número de la central de mensajes Número de la central de mensajes recibido Error al leer número de la central de mensajes
03h: Funciones del directorio telefónico	Solicitar posición de memoria del directorio Respuesta a solicitud de posición de memoria Error al leer posición de memoria Escribir en posición de memoria Escritura en posición de memoria sin error Error al escribir en posición de memoria Solicitar estado de las posiciones de memoria Respuesta a solicitud de estado de posiciones de memoria Error al leer estado de las posicines de memoria
14h: Control de mensajes de Texto	Escribir mensaje de texto en la SIM Marcar Mensaje como Leído Leer mensaje de texto recibido Posición de mensaje de texto vacía Borrar mensaje de texto Mensaje de texto borrado con éxito Solicitar estado de los mensajes de texto Respuesta a la soliciatud de estado de los mensajes de texto Error al leer estado de los mensajes de texto

- El campo FrameLength corresponde a la cantidad de bytes del bloque de datos (campo {block}) más dos bytes (FrameToGo y SeqNo)
- El Checksum1 corresponde a realizar un XOR a todos los elementos impares de la trama mientras que el Checksum2 corresponde a un XOR de todos los elementos pares de la trama, entonces el campo PaddingByte sólo se encontrará presente si

la cantidad de bytes impares es diferente a la cantidad de bytes pares, caso contrario, este campo no aparece en la trama.

- Al comparar las tramas enviadas y las tramas recibidas (ver tabla 3.3), siempre se encuentran en la misma posición 2 bytes con los valores 00 01 para el caso de las tramas enviadas y 01 08 para el caso de las tramas recibidas. Su posición siempre es constante (bytes 6 y 7 de la trama).

TABLA 3.3
VALORES CONSTANTES EN LAS TRAMAS ENVIADAS Y RECIBIDAS EN LA PC.

ENVIADAS															
1E	00	0C	04	00	06	00	01	00	01	01	40	13	42		
1E	00	0C	02	00	07	00	01	00	20	02	01	42	00	52	25
1E	00	0C	64	00	06	00	01	00	10	01	44	13	37		
1E	00	0C	40	00	06	00	01	64	01	01	45	77	03		
1E	00	0C	40	00	08	00	01	8F	04	09	C4	01	46	95	CF
1E	00	0C	40	00	08	00	01	8F	00	00	00	01	47	9C	0E
1E	00	0C	03	00	07	00	01	00	07	03	01	40	00	51	03
RECIBIDAS															
1E	0C	00	02	00	06	01	08	00	0E	01	45	1E	4B		
1E	0C	00	02	00	06	01	08	00	21	01	46	1E	67		

- Si el teléfono celular al enviar una trama no recibe el acuse de envío respectivo (Acknowledge) por parte de la computadora, repite la información transmitida tres veces cada medio segundo. Si en estos tres intentos no se recibe respuesta, el teléfono interrumpe la comunicación.

3.3 Identificación de los tipos de tramas

Este análisis se realizó con el objetivo de descifrar la función de cada byte dentro de las tramas FBUS. Para esto, se procedió a extraer tramas del archivo de registro obtenido en el procedimiento descrito en el ítem 3.2. Se modificó los campos SeqNo y checksums (CHKSum1 y 2) de las tramas y fueron enviadas hacia el teléfono celular mediante el software para envío de tramas descrito en el ítem 3.1.5, registrándose la respuesta para ser analizada. La figura 3.11 detalla este procedimiento en el cual la computadora mediante la ejecución del software envía las tramas a fin de visualizar y grabar la respuesta del teléfono.

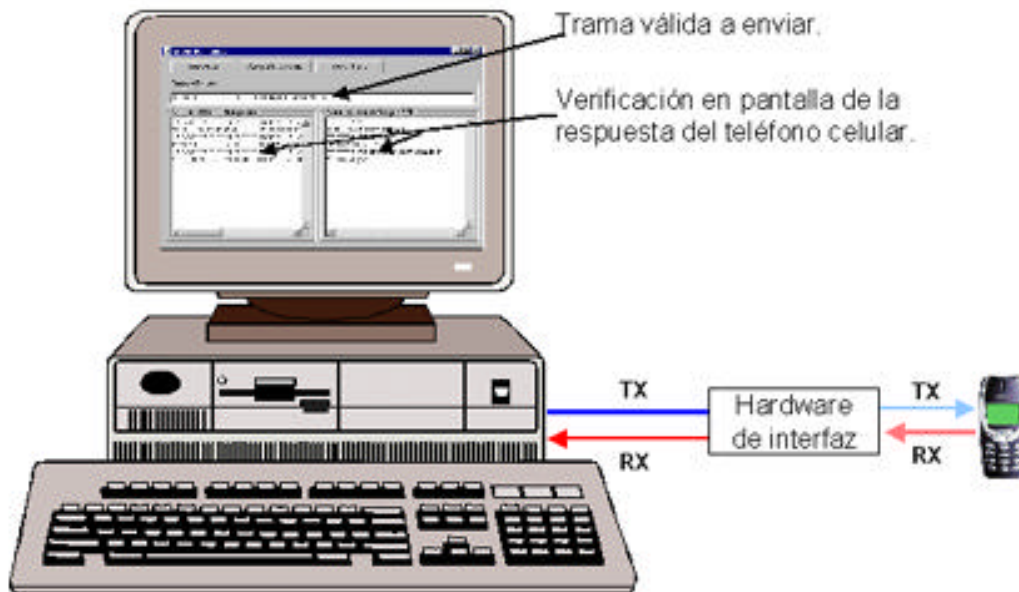


Figura 3.11. Procedimiento manual de envío de tramas

Los siguientes ítem describen las pruebas de envío de tramas realizadas.

3.3.1 Análisis del contenido de las tramas

Con el procedimiento indicado en el ítem 3.3, se procedió a solicitar al teléfono las siguientes acciones:

- Lectura del directorio telefónico
- Cantidad de mensajes de texto grabados en memoria
- Lectura de mensajes de texto
- Borrado de mensajes de texto
- Detección de la llegada de mensajes de texto nuevos
- Envío de mensajes de texto

Con las respuestas obtenidas por parte del teléfono, fue posible determinar la función de cada byte dentro de las tramas debido a que la información ya era conocida (por ejemplo, el contenido del directorio telefónico, los mensajes de texto, ambos grabados en el teléfono). En algunos casos fue necesario realizar comparaciones entre diversas tramas del mismo tipo para poder llegar a una conclusión general. Los siguientes ítem describen las pruebas realizadas con los tipos de tramas.

3.3.1.1 Lectura del directorio telefónico

Del archivo de registro obtenido, se observó que las tramas referentes al directorio telefónico, permiten familiarizarse con el protocolo FBUS debido a que son tramas pequeñas y fáciles de entender, por este motivo fueron utilizadas en un primer momento, para entender el funcionamiento del protocolo FBUS. A continuación se describe la trama para realizar la solicitud de lectura y la trama de respuesta.

a. Solicitud de lectura del directorio telefónico

De acuerdo a las observaciones del archivo de registro de la comunicación entre la computadora y el teléfono, se puede observar que la estructura de este tipo de trama es la que se muestra en la tabla 3.4.

TABLA 3.4
TRAMA DE SOLICITUD DE LECTURA DEL DIRECTORIO TELEFONICO

BYTE:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
DATO:	1E	00	0C	03	00	09	00	01	00	01	TM	00	NE	01	SQ	00	C1	C2

Donde se tiene que:

TM	:	Tipo de memoria.
NE	:	Número de la entrada de directorio.
SQ	:	Secuencia de la trama.
C1	:	Checksum 1.
C2	:	Checksum 2.

El byte 03 corresponde al tipo de mensaje para acceder al directorio telefónico, mientras que los bytes 8 y 9 constituyen la función 00 01 (lectura del directorio telefónico).

Los valores de las memorias donde puede ser leídas las entradas de números telefónicos son: 01h: Teléfono y tarjeta SIM juntas, 02h: En el teléfono, 03h: En la tarjeta SIM, 05h: Número del teléfono, 07h: Lista de llamadas realizadas, 08h: Lista de llamadas perdidas, 09h: Lista de llamadas recibidas, 0Bh: mensajes de voz.

b. Respuesta a la lectura del directorio telefónico

La respuesta del teléfono ante la lectura del directorio telefónico es la que se muestra a continuación en la tabla 3.5.

TABLA 3.5

TRAMA DE RESPUESTA A LA LECTURA DEL DIRECTORIO TELEFONICO

1E	0C	00	03	00	20	01	08	00	02	00	0D	LN	[Nombre]	Ln	[Número]	00	01	SQ	C1	C2
----	----	----	----	----	----	----	----	----	----	----	----	----	----------	----	----------	----	----	----	----	----

Donde se tiene que:

LN	:	Longitud del nombre grabado.
Nombre	:	Nombre grabado en Código ASCII.
Ln	:	Longitud del número telefónico.
Número	:	Número grabado en código ASCII.
SQ	:	Secuencia de la trama.
C1	:	Checksum 1.
C2	:	Checksum 2.

3.3.1.2 Cantidad de mensajes de texto grabados en memoria

Se envió hacia el teléfono celular la trama mostrada en la tabla 3.6. solicitando la cantidad de mensajes de texto grabados en memoria.

TABLA 3.6

TRAMA DE SOLICITUD DE CANTIDAD DE MENSAJES GRABADOS EN MEMORIA

Byte :	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Datos:	1E	00	0C	14	00	07	00	01	00	36	64	01	40	00	36	25

Donde se tiene que:

Byte 00 al 02	:	Cabecera de la trama.
Byte 03	:	Tipo de mensaje.
Byte 04	:	Siempre es 00h.
Byte 05	:	Longitud de la trama.
Byte 06 al 07	:	Siempre es 00 01.
Byte 08 al 09	:	Función (estado de las posiciones de memoria).
Byte 10	:	Siempre es 64h.

Byte 11 : Cantidad de tramas que se van a enviar.
 Byte 12 : Secuencia de la trama.
 Byte 13 : Padding byte.
 Byte 14 : Checksum de bytes impares.
 Byte 15 : Checksum de bytes pares.

La respuesta obtenida fue la trama indicada en la tabla 3.7.

TABLA 3.7

TRAMA DE RESPUESTA DE CANTIDAD DE MENSAJES GRABADOS EN MEMORIA

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Datos:	1E	0C	00	14	00	0E	01	08	00	37	01	00	00	19	00	00	13	00	01
Byte:	19	20	21																
Datos:	73	43	0C																

Donde se tiene que:

Byte 00 al 02 : Cabecera de la trama.
 Byte 03 : Tipo de mensaje.
 Byte 04 : Siempre es 00h.
 Byte 05 : Longitud de la trama.
 Byte 06 al 07 : Siempre es 01 08.
 Byte 08 al 09 : Función (estado de los mensajes de texto).
 Byte 10 al 12 : No hay información acerca de este campo.
 Byte 13 : Cantidad total de mensajes que se pueden grabar.
 Byte 14 al 15 : No hay información acerca de este campo.
 Byte 16 : Cantidad de mensajes grabados en el teléfono.
 Byte 17 : Cantidad de mensajes no leídos grabados en el teléfono.
 Byte 18 : Cantidad de tramas que se van a recibir.
 Byte 19 : Secuencia de la trama.
 Byte 20 : Checksum de bytes impares.
 Byte 21 : Checksum de bytes pares.

3.3.1.3 Lectura de mensajes de texto

Las tramas referentes a la lectura de mensajes de texto no sólo contienen la información del mensaje, también contienen información de la hora y fecha en la que fue enviado, número del remitente, etc

a. Solicitud de lectura de mensajes de texto

Para solicitar al teléfono la lectura del mensaje de texto, se envió la trama mostrada en la tabla 3.8.

TABLA 3.8
TRAMA DE SOLICITUD DE LECTURA DE MENSAJES DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
Datos:	1E	00	0C	14	00	0A	00	01	00	07	02	02	01	64	01	46	10	38

Donde se tiene que:

Byte 00 al 02	:	Cabecera de la trama.
Byte 03	:	Tipo de mensaje.
Byte 04	:	Siempre es 00h.
Byte 05	:	Longitud de la trama.
Byte 06 al 07	:	Siempre es 00 01.
Byte 08 al 09	:	Función (leer mensaje de texto de la memoria SIM).
Byte 10	:	Siempre es 02h.
Byte 11	:	Número de la posición del mensaje de texto a leer.
Byte 12 al 13	:	Siempre es 01 64.
Byte 14	:	Cantidad de tramas que se van a enviar.
Byte 15	:	Secuencia de la trama.
Byte 16	:	Checksum de bytes impares.
Byte 17	:	Checksum de bytes pares.

b. Respuesta a la lectura de mensajes de texto.

La respuesta obtenida como respuesta del teléfono fue la indicada en la tabla 3.9.

TABLA 3.9
TRAMA DE RESPUESTA A LA LECTURA DE MENSAJES DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Datos:	1E	0C	00	14	00	7A	01	08	00	08	01	02	02	00	07	91	15	91	97
Byte:	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Datos:	09	00	F0	00	00	00	00	04	39	00	9A	0B	85	15	A6	40	A8	65	CA
Byte:	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Datos:	E5	72	B9	40	20	90	8E	08	A2	97	41	65	B7	91	97	09	00	F0	60
Byte:	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
Datos:	32	90	02	30	50	91	71	73	85	00	A8	65	0A	84	5A	3D	FD	06	B5
Byte:	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
Datos:	EB	63	F4	7B	0E	12	97	E7	EF	39	0B	44	2F	83	C6	F5	B2	9B	FE
Byte:	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113
Datos:	06	C5	EB	65	90	FD	9D	07	85	41	69	39	28	0C	1A	B3	C3	F3	F2
Byte:	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129			
Datos:	1C	04	0F	CB	CB	E3	32	28	0E	0A	B3	CF	02	42	8C	81			

Donde se tiene que:

- Byte 00 al 02 : Cabecera de la trama.
- Byte 03 : Tipo de mensaje.
- Byte 04 : Siempre es 00h.
- Byte 05 : Longitud de la trama.
- Byte 06 al 07 : Siempre es 01 08.
- Byte 08 al 09 : Función (leer mensaje de texto de la memoria SIM).
- Byte 10 : Estado del mensaje de Texto: 01=Leído, 02=Memoria no válida, 03=No leído, 07=Posición vacía.
- Byte 11 : Siempre es 02h.
- Byte 12 : Número del mensaje de texto.

Byte 13	:	Siempre es 00h.
Byte 14	:	Cantidad de bytes del número del SMS Center.
Byte 15	:	Tipo de mensaje: 81=Desconocido, 91=Internacional, A1=Nacional.
Byte 16 al 25	:	Número del SMS Center. El número se debe leer byte a byte considerando el dígito más significativo el dígito de la derecha y menos significativo el de la izquierda. Ejemplo: 15 91 97 09 00 F0 = 51 19 79 90 00 0, deben ser en total 12 bytes, los faltantes, se completan con 00h.
Byte 26 al 44	:	No hay información acerca de este campo.
Byte 45	:	Cantidad de dígitos del número de origen.
Byte 50	:	No hay información acerca de este campo.
Byte 51 al 55:	:	Número de origen. Se decodifica igual que el número del SMS Center.
Byte 38 al 40	:	No hay información acerca de este campo.
Byte 57 al 60	:	Indica la fecha en la que se envió el mensaje. Se decodifica de la misma manera que el número del SMS Center.
Byte 62 al 65	:	Hora a la que se envió el Mensaje. Se decodifica de la misma manera que los casos anteriores.
Byte 66	:	Siempre es 00h, Se utiliza para separar los datos del mensaje de texto (fecha, hora, número, etc) del mensaje de texto en sí.
Byte 67 al 125	:	Mensaje de texto codificado.
Byte 126	:	Es la cantidad de tramas que el teléfono va a transmitir.
Byte 127	:	Es la secuencia de la trama.
Byte 128	:	Checksum de los bytes impares.
Byte 129	:	Checksum de los bytes pares.

Debido a que el mensaje era muy extenso para una sola trama, fue recibida una segunda con el resto del mensaje, la cual se detalla en la tabla 3.10.

TABLA 3.10
TRAMA DE CONTINUACION DEL MENSAJE DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Datos:	1E	0C	00	14	00	3C	75	F7	7B	0E	82	CB	DF	E6	F2	9C	05	72	BF
Byte:	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Datos:	41	E8	F0	B8	EC	06	A1	EB	65	F6	39	CC	02	D9	C3	ED	F7	1C	14
Byte:	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Datos:	06	D9	CB	72	50	BC	5E	06	A1	C3	79	17	08	04	02	51	8B	A0	60
Byte:	57	58	59	60	61	62	63	64	65	66	67								
Datos:	F3	19	0A	85	40	BB	14	01	03	76	02								

Donde se tiene que:

- Byte 00 al 02 : Cabecera de la trama
- Byte 03 : Tipo de mensaje.
- Byte 04 : Siempre es 00h
- Byte 05 : Longitud de la trama
- Byte 06 al 63 : Continuación del mensaje de texto de la trama anterior.
- Byte 64 : Cantidad de tramas a enviar.
- Byte 65 : Secuencia de la trama.
- Byte 66 : Checksum de los bytes impares.
- Byte 67 : Checksum de los bytes pares.

3.3.1.4 Borrado de mensajes de texto

Un aspecto muy importante a tener en cuenta es que los mensajes de texto luego de ser recibidos y atendidos, inmediatamente sean borrados para así evitar que las posiciones de memoria se llenen y no se pueda seguir recibiendo mensajes. Para obtener los detalles de este tipo de tramas, se envió al teléfono la trama indicada en la tabla 3.11 solicitando el borrado de un mensaje.

TABLA 3.11
TRAMA DE BORRADO DE MENSAJES DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Datos:	1E	00	0C	14	00	08	00	01	00	0A	02	01	01	01	42	11

Donde se tiene que:

Byte 00 al 02	:	Cabecera de la trama.
Byte 03	:	Tipo de mensaje.
Byte 04	:	Siempre es 00h.
Byte 05	:	Longitud de la trama.
Byte 06 al 07	:	Siempre es 00 01.
Byte 08 al 09	:	Función (borrar mensaje de texto).
Byte 10	:	Siempre es 02h.
Byte 11	:	Número del mensaje que se quiere borrar.
Byte 12	:	Cantidad de tramas que se van a enviar.
Byte 13	:	Secuencia de la Trama.
Byte 14	:	Checksum de bytes impares.
Byte 15	:	Checksum de bytes pares.

3.3.1.5 Detección de la llegada de mensajes de texto nuevos

Durante la experimentación, se observó que al llegar mensajes de texto nuevos el teléfono envía automáticamente una trama indicando este suceso. En esta trama, figura el número de la posición de memoria en la que se grabó el mensaje, además del contenido del mensaje entre otros datos. En esta trama, los datos no tienen una longitud ni posición fija como en el caso de las tramas para solicitar los mensajes de texto. Este detalle complica la labor de obtener el mensaje de texto desde este tipo de tramas debido a que no se puede generalizar con facilidad. A continuación se muestra sólo los 13 primeros bytes de este tipo de trama:

Byte :00 01 02 03 04 05 06 07 08 09 10 11 12
Datos:1E 0C 00 02 00 7A 01 08 00 10 02 09 00

En donde los bytes 8 y 9 (00h y 10h) representan la función recibida (mensaje de texto nuevo recibido) y el byte 11 es el que indica el número de la posición de memoria que se debe leer. En este caso, se indica que en la posición 09 (byte 11) ha sido grabado el mensaje recibido.

3.3.1.6 Envío de mensajes de texto

Para realizar el envío del mensaje de texto es necesario conocer el número del SMS Center (central de mensajes de texto) la cual será quien redirija el mensaje al destino. Si se desea obtener este número en caso de no ser conocido, existe una trama para solicitárselo al teléfono como se ve a continuación.

a. Lectura del número de la central de mensajes de texto

Para obtener el número de la central de mensajes de texto, se envió al teléfono la trama que se muestra en la tabla 3.12

TABLA 3.12
TRAMA DE LECTURA DEL NUMERO DE LA CENTRAL DE MENSAJES DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Datos:	1E	00	0C	02	00	08	00	01	00	33	64	01	01	40	77	79

Donde se tiene que:

Byte 00 al 02	:	Cabecera de la trama.
Byte 03	:	Tipo de mensaje.
Byte 04	:	Siempre es 00h.
Byte 05	:	Longitud de la trama.
Byte 06 al 07	:	Siempre es 00 01.
Byte 08 al 09	:	Función (leer número del SMS Center).
Byte 10	:	Siempre es 64h.
Byte 11	:	No hay información acerca de este campo.
Byte 12	:	Cantidad de tramas que se van a enviar.
Byte 13	:	Secuencia de la trama.
Byte 14	:	Checksum de bytes impares.
Byte 15	:	Checksum de bytes pares.

La trama recibida, fue la indicada en la tabla 3.13.

TABLA 3.13

TRAMA DE RESPUESTA AL NUMERO DE LA CENTRAL DE MENSAJES DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Datos:	1E	0C	00	02	00	24	01	08	00	34	01	FD	00	00	00	00	00	00	00	00
Byte:	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Datos:	00	00	00	00	00	00	00	06	91	15	71	99	00	00	00	00	00	00	00	00
Byte:	40	41	42	43																
Datos:	01	47	FF	26																

Donde se tiene que:

- Byte 00 al 02 : Cabecera de la trama
- Byte 03 : Tipo de mensaje.
- Byte 04 : Siempre es 00h
- Byte 05 : Longitud de la trama
- Byte 06 al 07 : Siempre es 01 08
- Byte 08 al 09 : Función (número del SMS Center)
- Byte 10 : Siempre es 01h
- Byte 11 al 26 : No hay información acerca de este campo.
- Byte 27 al 38 : Número del SMS Center (12 bytes). Tal como es obtenido puede aprovecharse para ser insertado en el bloque que corresponde al número del SMS Center en la trama para enviar mensajes de texto.
- Byte 39 : No hay información acerca de este campo.
- Byte 40 : Cantidad de tramas que se van a enviar.
- Byte 41 : Secuencia de la trama.
- Byte 42 : Checksum de bytes impares.
- Byte 43 : Checksum de bytes pares.

b. Envío de mensajes de texto

Como ya se mencionó, esta trama debe contener un conjunto de información para el envío del mensaje de texto y no sólo los números telefónicos, también deben incluirse otros valores como el periodo de validez del mensaje, el esquema de la codificación de los datos, etc. En la publicación de referencia [13] se encuentra la estandarización de los parámetros útiles para el envío de mensajes de texto en sistemas GSM. La trama

enviada para experimentar el envío de mensajes de texto, es la que se muestra en la tabla 3.14.

TABLA 3.14
TRAMA DE SOLICITUD DE ENVÍO DE MENSAJES DE TEXTO

Byte:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Datos:	1E	00	0C	02	00	56	00	01	00	01	02	00	06	91	15	71	99	00	00
Byte:	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Datos:	00	00	00	00	00	11	00	00	F1	30	08	81	79	24	75	98	00	00	00
Byte:	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Datos:	00	00	00	A9	00	00	00	00	00	00	C5	39	3D	0C	2A	CF	41	75	77
Byte:	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
Datos:	18	04	97	D7	CB	E2	30	88	5C	06	95	DD	F6	F4	1B	44	2E	83	DA
Byte:	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	
Datos:	65	F7	3C	AC	2E	CF	41	E4	32	88	5E	C6	D3	DF	01	41	BB	63	

Donde se tiene que:

- Byte 00 al 02 : Cabecera de la trama.
- Byte 03 : Tipo de mensaje.
- Byte 04 : Siempre es 00h.
- Byte 05 : Longitud de la trama.
- Byte 06 al 07 : Siempre es 00 01.
- Byte 08 al 09 : Función (envía mensaje de texto).
- Byte 10 al 11 : Siempre es 02 00.
- Byte 12 al 23 : Número del SMS Center. (12 bytes).
 - Byte 12 : Longitud del número del SMS Center.
 - Byte 13 al 23 : Número del SMS Center.
- Byte 24 al 28 : TPDU Transfer Protocol Data Unit (5 bytes).
 - Byte 24 : Tipo de mensaje (utilizar siempre 11h).
 - Byte 25 : Referencia del mensaje (utilizar siempre 00h).
 - Byte 26 : Protocol ID; para este caso utilizar 00h.
 - Byte 27 : Esquema de codificación de datos, para este caso utilizar siempre 00h.

Byte 28	:	Cantidad de caracteres que tiene el mensaje.
Para más detalles acerca de este punto revisar la publicación GSM 03.40 de la ETSI.		
Byte 29 al 40	:	Número destino (12 bytes)
Byte 29	:	Cantidad de dígitos del número destino.
Byte 30	:	Ubicación: 81= desconocido, 91 = internacional, A1 = nacional.
Byte 31 al 40	:	Número del remitente. Se encuentra codificado de la misma manera que en todos los casos anteriores. Si quedan de este bloque bytes sin completar, se rellenan con 00.
Byte 41 al 47	:	SCTS Service Center Time Stamp (7 bytes).
Byte 41	:	Periodo de validez del mensaje: 0Bh=1 hora, 47h=6 horas, A7h=24 horas, A9h=72 horas, ADh=1 semana, FFh = tiempo máximo. Por defecto utilizar A9 (72 horas).
Byte 42 al 47	:	Rellenar con ceros (00 00 00 00 00 00).
Para mayores detalles acerca de este punto se recomienda revisar la publicación GSM 03.40 de la ETSI.		
Byte 48 al 89	:	Mensaje de texto codificado.
Byte 90	:	Cantidad de tramas que se van a enviar.
Byte 91	:	Secuencia de la trama.
Byte 92	:	Checksum de bytes impares.
Byte 93	:	Checksum de bytes pares.

Al enviar el mensaje, el teléfono respondió con el tipo de mensaje 02h y la función 00 02 (mensaje fue enviado sin errores), en los casos cuando el mensaje no pudo ser enviado por algún error, el teléfono respondió con la función 00 03 (error al enviar mensaje).

3.3.2 Conclusiones de los detalles de las tramas

- Las tramas FBUS pueden tener como máximo una longitud de 130 bytes.

- Las tramas de los mensajes de texto no sólo contienen la información del mensaje, también contienen información como número del remitente, número de la central de mensajes de texto, fecha, hora de envió, etc.
- Este análisis permitió documentar los tipos de solicitudes de interés para el trabajo de tesis, así como también las funciones disponibles. La tabla 3.15 detalla estas conclusiones.

TABLA 3.15
VALORES HEXAGESIMALES DE LAS FUNCIONES ANALIZADAS

Tipo de Mensaje	Función	Función en Hexagesimal
02h	Solicitar envió de mensaje de texto	00 01
	Mensaje de texto enviado sin error	00 02
	Envío de Mensaje de texto fallido	00 03
	Solicitar Mensaje de texto recibido	00 07
	Mensaje de texto nuevo recibido	00 10
	Obtener número de la central de mensajes	00 33
	Número de la central de mensajes recibido	00 34
	Error al leer número de la central de mensajes	00 35
03h	Solicitar posición de memoria del directorio	00 01
	Respuesta a solicitud de posición de memoria	00 02
	Error al leer posición de memoria	00 03
	Escribir en posición de memoria	00 04
	Escritura en posición de memoria sin error	00 05
	Error al escribir en posición de memoria	00 06
	Solicitar estado de las posiciones de memoria	00 07
	Respuesta a solicitud de estado de posiciones de memoria	00 08
	Error al leer estado de las posicines de memoria	00 09
14h	Escribir mensaje de texto en la SIM	00 04
	Leer mensaje de texto recibido	00 07
	Respuesta a la lectura de mensaje de texto	00 08
	Posición de mensaje de texto vacia	00 09
	Borrar mensaje de texto	00 0A
	Mensaje de texto borrado con exito	00 0B
	Solicitar estado de los mensajes de texto	00 36
	Respuesta a la soliciatud de estado de los mensajes de texto	00 37
	Error al leer estado de los mensajes de texto	00 38

- Se consiguió realizar una generalización para los diversos tipos de tramas para realizar solicitudes al teléfono, dicha información es la que se muestra en la tabla 3.16.

TABLA 3.16
GENERALIZACION DE LAS SOLICITUDES

Enviar mensaje de texto.	1E 00 0C 02 00 + Longitud + 00 01 00 01 02 00 + SMSC + TPDU + Destino + Vp + SCTS + MensajedeTextoCodificado + 01 + SeqNo + PaddingByte + ChkSum1 + ChkSum2
Solicitar número de la central de mensajes de texto.	1E 00 0C 02 00 08 00 01 00 33 64 01 01 + SeqNo + PaddingByte + ChkSum1 + ChkSum2
Solicitar mensaje de texto.	1E 00 0C 14 00 0A 00 01 00 07 02 + Número de mensaje + 01 64 01 + SeqNo + PaddingByte + ChkSum1 + ChkSum2
Solicitar cantidad de mensajes totales y no leídos.	1E 00 0C 14 00 07 00 01 00 36 64 01 + SeqNo + PaddingByte + ChkSum1 + ChkSum2
Borrar mensaje de texto.	1E 00 0C 14 00 08 00 01 00 0A 02 número de mensaje a borrar + PaddingByte + ChkSum1 + ChkSum2
Solicitar entrada de directorio.	1E 00 0C 03 00 09 00 01 00 01 03 + número de entrada + 00 01 + SeqNo + PaddingByte + ChkSum1 + ChkSum2
Solicitar estado de memorias del directorio telefónico.	1E 00 0C 03 00 07 00 01 00 07 + Tipo_de_Memoria + 01 + SeqNo + PaddingByte + ChkSum1 + ChkSum2
Solicitar versión de hardware y software.	1E 00 0C D1 00 07 00 01 00 03 00 + SeqNo + PaddingByte + ChkSum1 + ChkSum2

Donde : SeqNo = Secuencia de la trama.
 PaddingByte = (00) En caso de que el número de bytes impares
 sea mayor a la de bytes pares.

- Los mensajes de texto al recibirse, no son grabados por el teléfono en forma consecutiva, sino en la primera posición vacía que el teléfono encuentra dentro de su memoria. El teléfono, siempre antes de grabar el mensaje, realiza una exploración del estado de las posiciones (libres y vacías) para localizar la primera posición vacía.
- El número de caracteres que podrán ser enviados por mensaje de texto, no depende de la trama que se genere ni del modelo o marca del teléfono celular sino de las características del servicio de telefonía celular que esté siendo utilizado (Telefónica, TIM, Bellsouth para el caso de Perú). Para sistemas GSM (TIM) la cantidad de caracteres es de 160.

IV. IMPLEMENTACIÓN DEL PROTOCOLO FBUS

En el modelo OSI, cada capa realiza una función específica para lograr la comunicación entre dos equipos, de acuerdo a las particularidades de las capas de OSI, el protocolo FBUS puede ser descompuesto en cuatro capas como se muestra en la figura 4.1 a fin de facilitar su implementación. Este modelamiento por capas permite asignar tareas específicas a cada capa del protocolo FBUS, el cumplimiento de estas tareas provee de información a las capas contiguas y permite realizar la interacción entre el sistema y el usuario final.

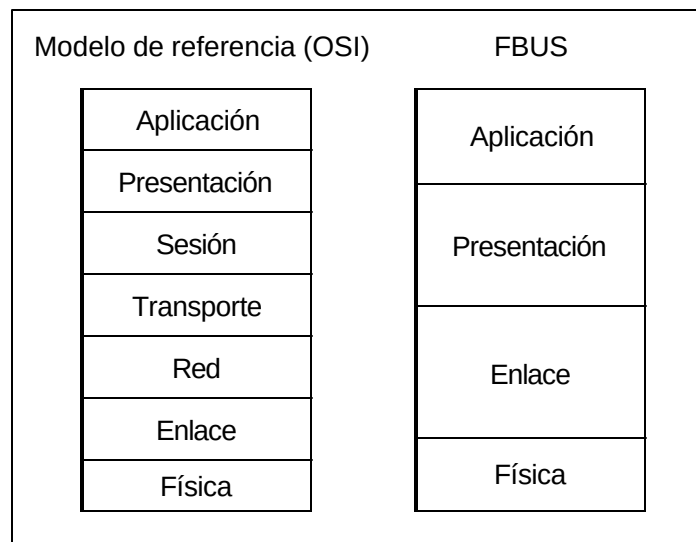


Figura 4.1. Comparación de la comunicación FBUS con el modelo OSI

El protocolo FBUS, no presenta una capa de red debido a que la comunicación es punto a punto (de la computadora hacia el teléfono celular y viceversa). También carece de una capa de sesión debido a que no es necesario ningún tipo de validación para iniciar el flujo de información entre los dispositivos. Tampoco es conveniente implementar una capa de transporte debido a que la única tarea propia de esta capa, para los fines del trabajo de la tesis, hubiera sido la segmentación de las tramas en grupos de 130 bytes, tarea que fue asignada a la capa de enlace.

La implementación del protocolo FBUS consiste en elaborar un programa de software que realiza la interacción entre la capa de aplicación y la capa de enlace. El programa debe reconocer eficientemente las tramas recibidas y separar la información útil que se encuentre contenida en ellas, así como también atender las solicitudes que el teléfono realiza según

sea el caso. El cumplimiento de estas tareas, principalmente, se realiza mediante la capa de enlace y la capa de presentación, los cuales son los puntos centrales de este capítulo. El protocolo de la capa física es el RS232C medio por el cual el FBUS transfiere bytes entre el teléfono y la computadora. La transferencia de datos entre las diversas capas se realiza según la figura 4.2.

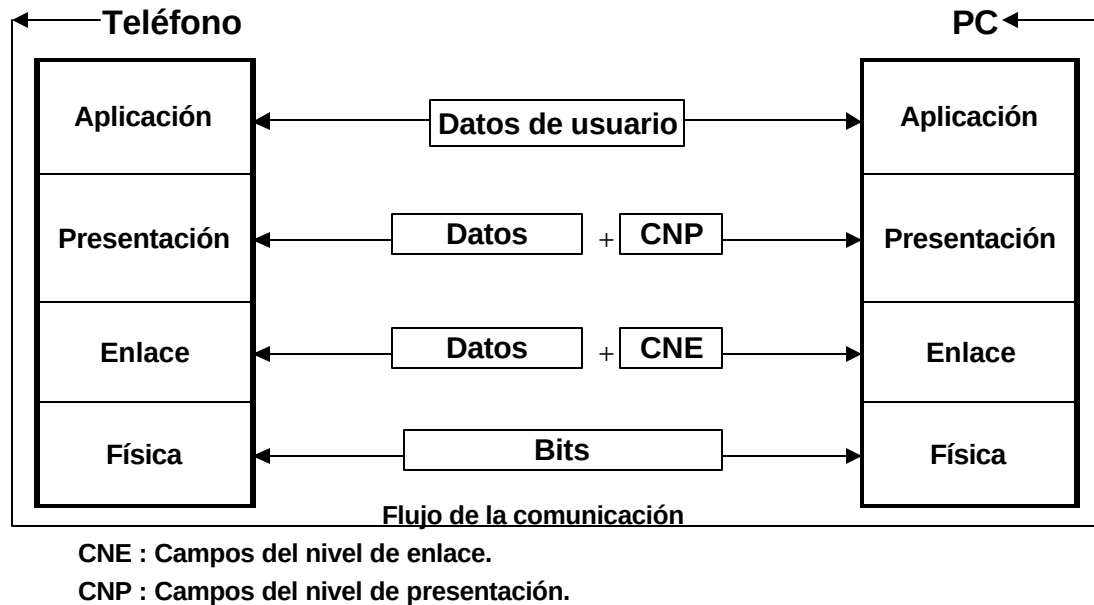


Figura 4.2. Flujo de la comunicación en el protocolo FBUS

En los siguientes puntos se describe la implementación de las capas física, enlace, presentación y aplicación de la comunicación FBUS, enfatizando las capas de enlace y presentación.

4.1 Capa física

En esta capa, se establece la interconexión entre la computadora y el teléfono celular. El medio de comunicación es el puerto serial de comunicaciones bajo es estándar RS232C empleando como norma 115200 Bps, 8 bits de datos, sin bit de paridad y un bit de parada.

Para la implementación de esta capa, se construyó el circuito de interfaz para el teléfono celular Nokia 3390 descrito en el ítem 3.1.1.

4.2 Capa de enlace

En esta capa, los datos binarios provenientes de la capa física son vistos como tramas de datos. Las tramas que interactúan en la capa de enlace, son las que fueron deducidas en el ítem 3.2. Las funciones que realiza esta capa y que son implementadas por software son las siguientes:

- Composición / Descomposición de los campos de la trama recibida.
- Detección del inicio y final de las tramas.
- Verificación / Generación de los checksums en las tramas.
- Detección y envío de la trama Acknowledge.
- Cálculo y verificación de las secuencias de las tramas.

4.2.1 Composición / Descomposición de los campos de la trama recibida

Esta tarea permite al nivel de enlace reconocer el tipo de trama con la que va a interactuar, así como también poder reconocer el inicio y final de las tramas, generar o verificar los checksums, etc.

Con el fin de reducir la cantidad de variables utilizadas al momento de programar las rutinas del protocolo FBUS, se optó por agrupar algunos campos en cabeceras, tipo de mensaje, longitud de la trama, bloque de datos y checksums como se detalla a continuación.

- **Cabeceras.-** El software implementado, lee los tres primeros bytes de las cadenas recibidas de la capa física para encontrar el inicio de las tramas. Esta cadena contiene la información del tipo de identificador de protocolo, el destino y la fuente de la comunicación. El valor a detectar siempre será constante debido a que las tramas a recibirse (por la computadora) siempre provendrán del teléfono celular. El valor de la cabecera siempre coincidirá con la siguiente cadena "1E 00 0C", la cual indica que la comunicación se realiza mediante el protocolo FBUS (1E) y la información va desde el teléfono (00) hacia la computadora (0C).
- **Tipo de mensaje.-** El Software implementado, lee el cuarto byte de la trama, que corresponde dentro de la trama al campo MsgType. Cada posible valor de este byte se

encuentra relacionado con una determinada función del teléfono celular. La tabla 4.1, indica los valores de los diferentes tipos de mensaje utilizados.

TABLA 4.1
LISTA DE TIPOS DE MENSAJES

Tipo de mensaje	Descripción
02h	Manipulación de mensajes de texto
03h	Directorio telefónico
14h	Control de los mensajes de texto
7Fh	Acknowledge

Al leer el código del tipo de mensaje, se proporciona al nivel de enlace información acerca de cual es el tipo de trama con el que va a trabajar o para generar el acuse de envío.

- **Longitud de la trama.-** Lee el sexto byte de la trama. Este campo va a ser el indicador de la cantidad de bytes que contiene el Bloque de datos incluyendo también los campos secuencia de la trama y el FramestoGo.
- **Bloque de datos.-** En este campo, se encuentra la información útil recibida; La obtención de estos datos corresponde a leer comenzando desde el séptimo byte de la trama hasta “n” bytes, siendo “n” el valor de la longitud de la trama. Dentro de estos bytes también se encuentran los campos FramesToGo (anteúltimo byte de este campo) y SeqNo. (último byte de este campo), los cuales son separados para proporcionar el nivel de presentación solamente el campo Bloque de datos.

Se debe tener en cuenta que cuando la información proviene del teléfono celular, los dos primeros bytes del bloque de datos siempre serán “01 08”, mientras que si la información proviene de la computadora, siempre serán “00 01”.

- **Checksums.-** Corresponde a leer el valor de los dos últimos bytes de la trama, siendo checksum1 el penúltimo byte de la trama y checksum2 el último byte de la trama.

4.2.2 Detección del inicio y final de una trama válida

Cuando el teléfono celular transmite una trama hacia la computadora, la acción puede ser realizada en más de un proceso de transmisión. Como consecuencia, para completar una trama de datos, deben ser unidos los datos recibidos en dos o más ciclos de transmisión, este proceso es sincronizado por el inicio de la trama (cabecera) y el final de la misma (checksums).

Esto puede ser apreciado claramente si se observa al detalle el contenido de la información que se obtiene al recibir tramas desde el teléfono celular con el programa desarrollado para este fin. La figura 4.3 describe este hecho. Como se aprecia cada una de las dos tramas recibidas se encuentran visualizadas en dos líneas, lo que significa, que fueron recibidas cada una en ciclos de transmisión distintos.

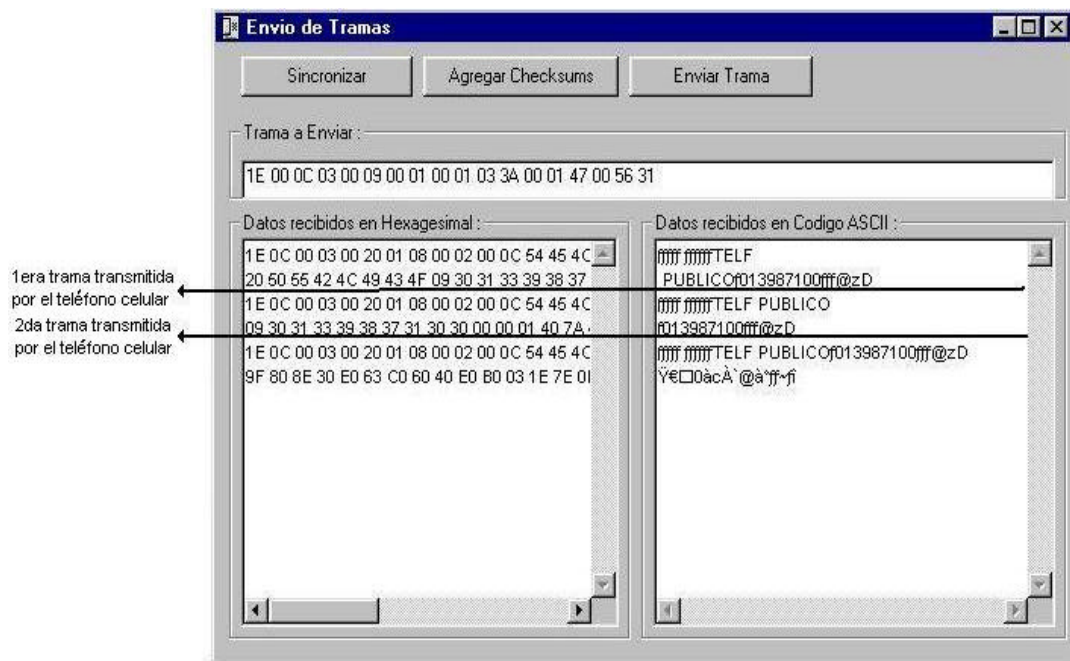


Figura 4.3. Cantidad de ciclos de transmisión para enviar una trama

Para reconstruir en la PC la trama enviada por el teléfono, se realiza el siguiente proceso:

- Detectar el inicio de la trama (cadena "1E 0C 00").
- Calcular los checksums de los datos recibidos y comparar el resultado con los últimos dos bytes de la trama, si los valores coinciden, la trama se encuentra completa.

- Si el valor de los checksums no coincide, se almacena los datos recibidos hasta el momento en una variable temporal y al detectarse nuevamente la llegada de datos, se concatena los datos guardados con los recién recibidos para repetir el proceso hasta que coincidan los checksums.
- Después de una determinada cantidad de procesos de transmisión, si no hay un resultado correcto, los datos deben ser descartados asumiendo que hay error en la recepción de los datos.

El diagrama de flujo en la figura 4.4, describe este proceso, donde Contatramas es una variable que lleva el conteo de cuantos ciclos de transmisión se van empleando para obtener la trama. Cuando el valor de esta variable es mayor a 10, se descartan todos los datos iniciándose el proceso nuevamente con el fin de descartar los datos erróneos recibidos y esperar la llegada de una nueva trama.

TramaFBUSRecibidaTemp es una variable en la que se almacena los datos de la trama inconclusa recibida hasta el momento. FlagtramaFBUSRecibidaTemp es una variable que indica mediante el valor de 1 que el contenido de la variable TramaFBUSRecibidaTemp no se encuentra vacío y que los datos que sean recibidos por el puerto serial deben ser concatenados con el contenido de esta variable. En cada ciclo de recepción de datos, el contenido de la variable Buffer se verifica para ver si es una trama FBUS mediante la búsqueda de la cabecera (1E 0C 00) y la verificación de los checksums.

Una vez que la trama FBUS es reconocida, los datos contenidos en la misma son analizados para su respectiva separación y/o utilización por el siguiente nivel de la comunicación.

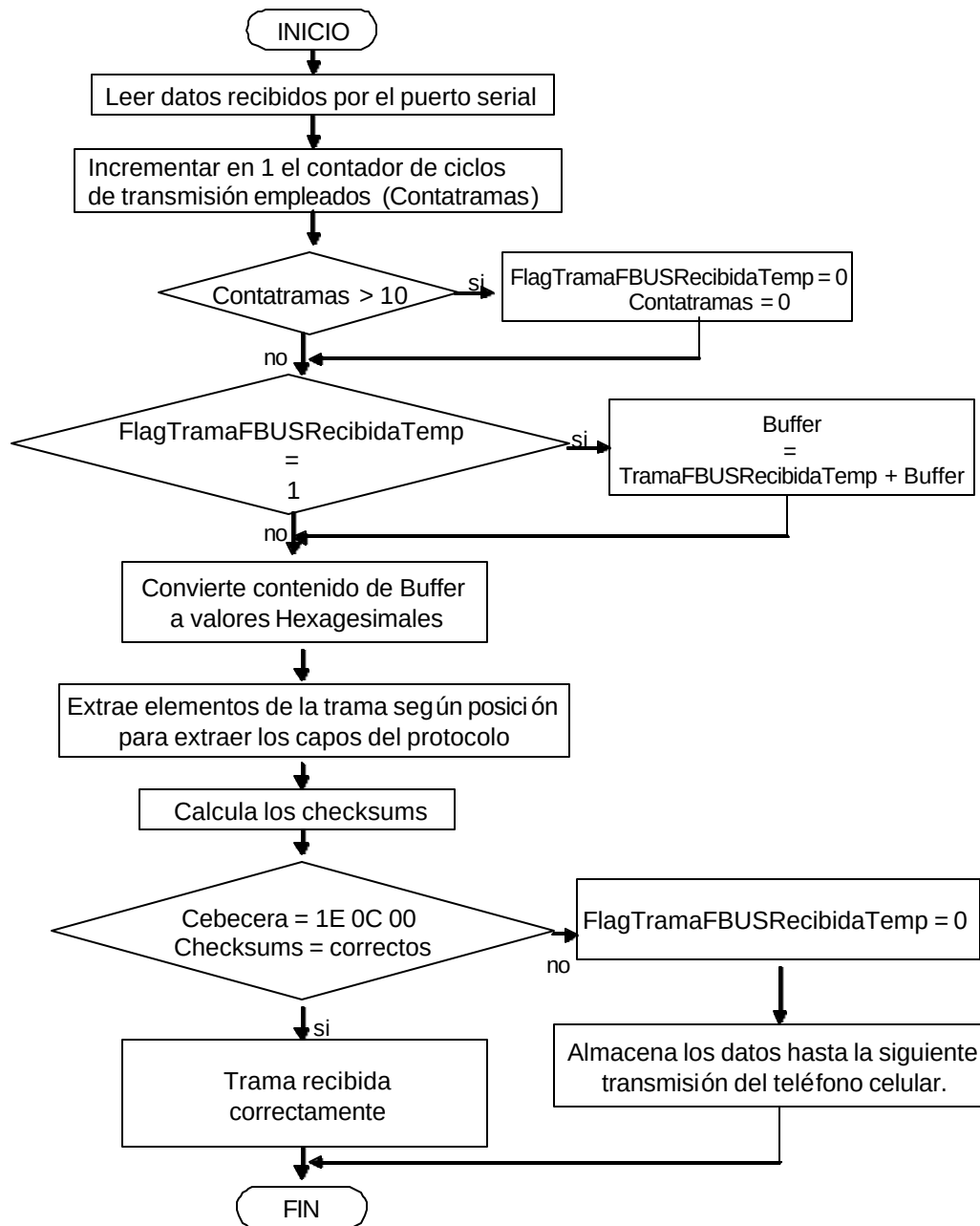


Figura 4.4. Diagrama de flujo del proceso de obtención de una trama completa

4.2.3 Verificación de checksums en las tramas provenientes del teléfono

El cálculo de los checksums es una operación muy importante pues permite detectar cuales tramas son válidas y cuales han sido transmitidas con error. También sirve para detectar el

final de una trama recibida como fue visto en el Ítem anterior. Se implementó una rutina que realiza este cálculo y almacena los resultados en las variables CHK1RX y CHK2RX. Para calcular los checksums, primero deben ser separados los bytes impares y los pares de la trama para luego aplicar a cada uno de estos conjuntos de bytes la operación XOR, no se debe incluir los dos últimos bytes que son los valores de los checksums que han sido recibidos y que servirán para ser comparados con el valor obtenido de las operaciones. Se incluyó en esta rutina una etapa de comparación para que el programa indique mediante el valor de una variable si el valor de los checksums calculados son iguales a los valores de los checksums recibidos (CHKOOK=1). El procedimiento descrito para realizar el calculo de los cheksums es el que se indica en el diagrama de flujo de la figura 4.5.

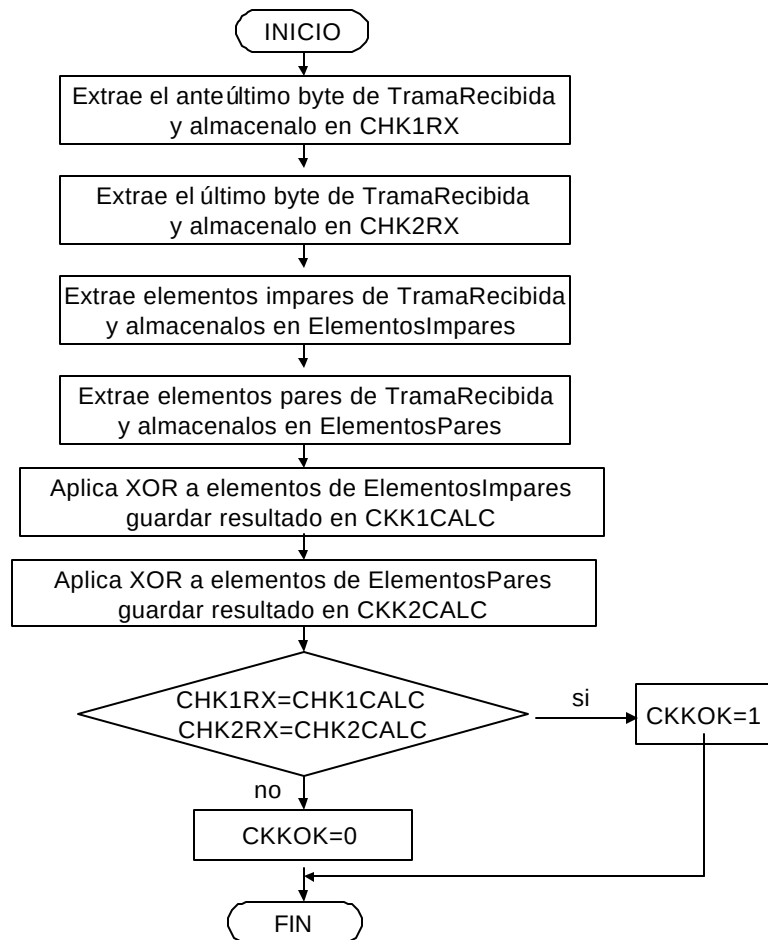


Figura 4.5. Diagrama de flujo para el cálculo y comparación de checksums

4.2.4 Cálculo de checksums para las tramas por enviar

El cálculo de los checksums 1 y 2 es un proceso que se realiza para completar los campos de las tramas generadas por la computadora en el nivel de presentación, antes de ser enviada por el nivel físico.

Para esto, se implementó una rutina que realiza la operación XOR a los bytes de orden impar de la trama por enviar y almacena el resultado en la variable CHECKSUM1, de igual manera se procede con los bytes de orden par de la trama por enviar para almacenarlos en otra variable por ejemplo CHECKSUM2. El contenido de estas dos variables se agrega a la trama por enviar. También verifica que la cantidad de bytes impares sea igual a la cantidad de bytes pares para generar si es necesario en campo PaddingByte.

4.2.5 Detección y generación de la trama Acknowledge

Para detectar la recepción de una trama de Acknowledge, se busca dentro de una trama completa que el valor del tipo de mensaje coincida con el valor 7Fh.

Por otro lado, para generar la trama de Acknowledge en respuesta a la información recibida, son necesarios los siguientes datos:

- Código del tipo de mensaje al que se va a responder.
- Secuencia de Acknowledge que corresponde.

En la figura 4.6 se encuentra el diagrama de flujo que describe la implementación de la rutina para generar la trama de Acknowledge, donde TramaFBUSaEnviar es la variable donde se almacena el contenido de la trama que debe ser enviada y Comando_a_responder es una variable donde previamente se almacenó el tipo de mensaje al que corresponde la trama recibida a la cual se desea hacer el acuse de envió.

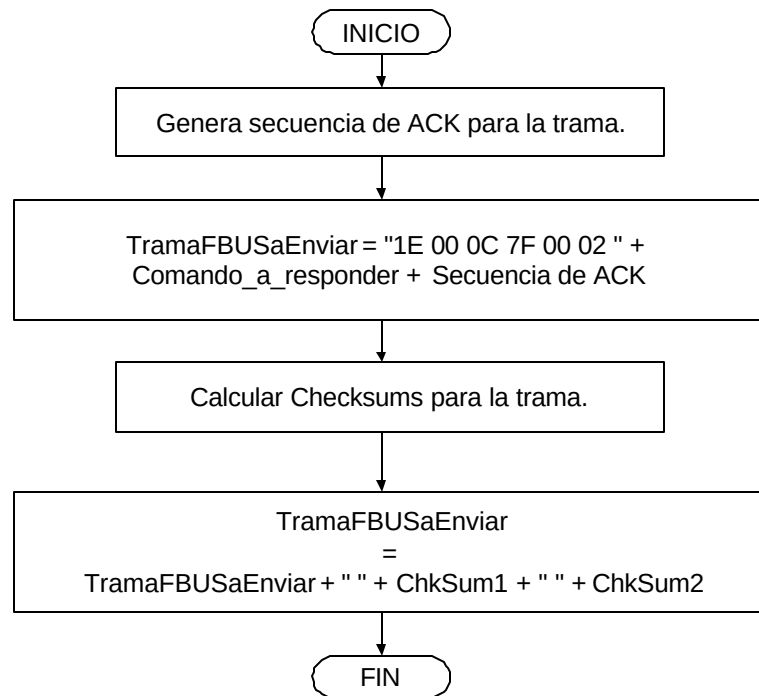


Figura 4.6. Diagrama de flujo para generar la trama de Acknowledge

4.2.6 Cálculo de la secuencia de las tramas por enviar

El protocolo FBUS, para la transmisión de tramas mantiene un riguroso orden con respecto al momento en que son enviadas mediante el campo **SeqNo** el cual está conformado por solo un byte. Si observamos los 5 primeros bits (del más significativo al menos significativo) se tiene que:

- 01000 : Cuando se envía una trama.
- 00000 : Cuando se envía un Acknowledge.

El conteo de la secuencia para las tramas se realiza mediante el incremento de los 3 últimos bits, en forma consecutiva entre 0 y 7 para luego volver a empezar. En la programación del protocolo FBUS, se debe considerar la secuencia de las tramas de solicitudes y la secuencia de las tramas de Acknowledge en forma separada debido a que el valor de la secuencia para cada uno de estos casos es independiente.

Este hecho se puede apreciar en la figura 4.7, donde se detalla el valor de la secuencia de las tramas solicitudes y el de las tramas de Acknowledge.

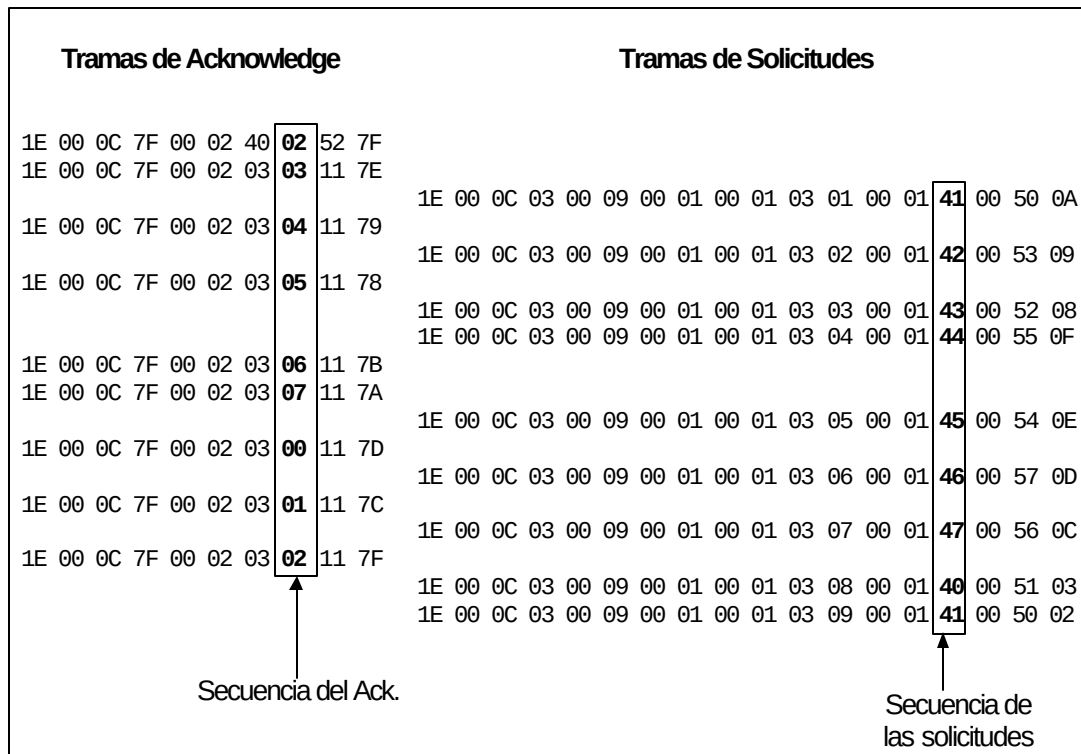


Figura 4.7. Secuencias en tramas de Acknowledge y de solicitudes

4.3 Capa de presentación

En esta capa es donde se procede a recuperar los datos validos contenidos en el bloque de datos de una trama FBUS; Es aquí donde también se generan tramas previas que serán completadas por el nivel de enlace para ser enviadas hacia el teléfono celular solicitándole acciones. La implementación de la capa de presentación está dividido en dos partes, obtención de los datos contenidos en el bloque de datos y generalización de tramas para solicitar acciones al teléfono celular.

4.3.1 Obtención de la información contenida en el bloque de datos

Cuando el nivel de presentación recibe el bloque de datos y el código del tipo de mensaje, procede a separar los datos útiles para recuperar la información. El código del tipo de mensaje indica a la capa de presentación la manera como debe ser separada la información útil contenida en el bloque de datos.

Existe dentro del bloque de datos un conjunto de bytes al que se le denominó “función” el cual indica que tipo de respuesta se ha obtenido, teniéndose que para cada tipo de mensaje, existen varias funciones posibles. La obtención de la función, corresponde a leer, desde el tercer hasta el cuarto byte del bloque de datos tal como fue visto en el capítulo anterior.

La figura 4.8, muestra el diagrama de flujo que refiere el procedimiento para detectar el código de mensaje recibido y el procedimiento en general para extraer los datos útiles comprendidos en el bloque de datos; este último, es detallado en las figuras 4.9, 4.10 y 4.11.

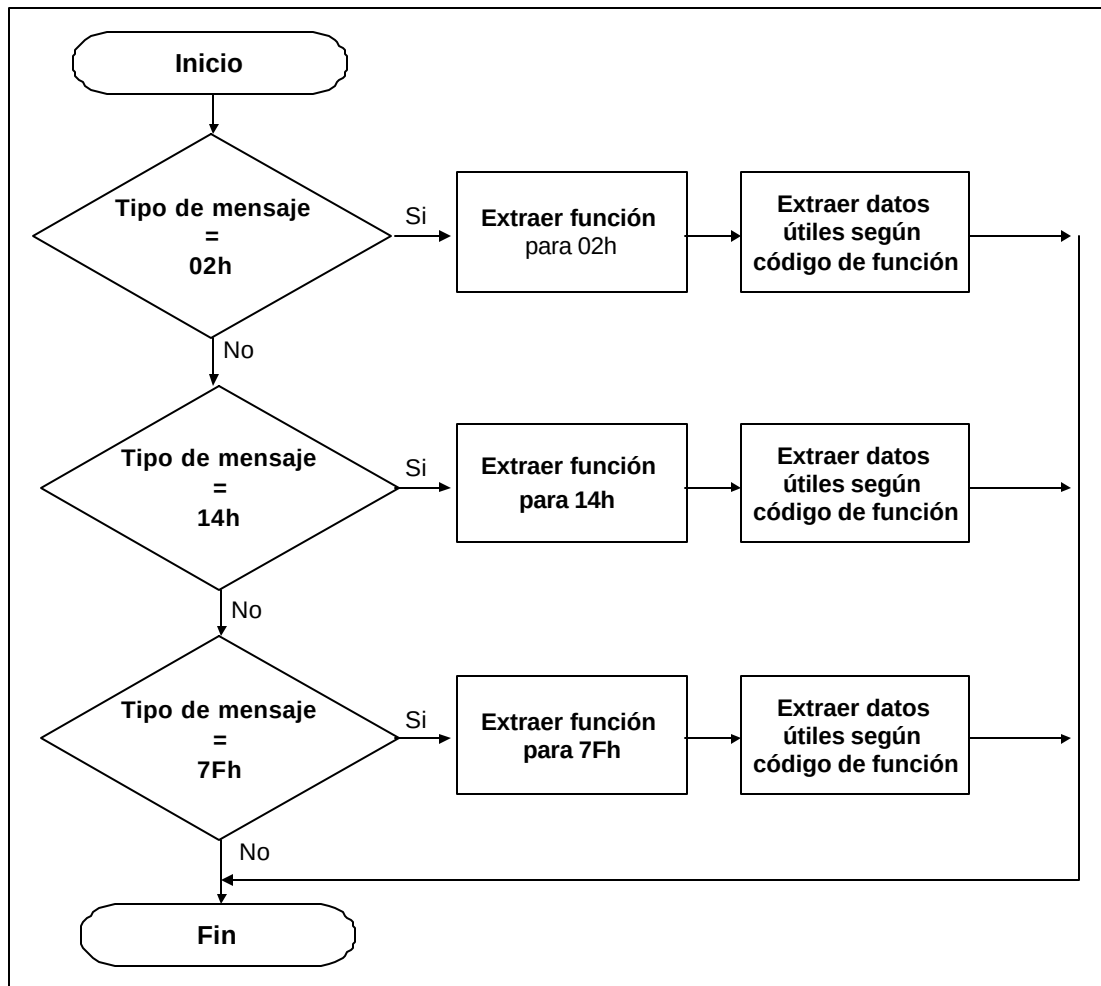


Figura 4.8. Detección del tipo de mensaje y extracción de la función

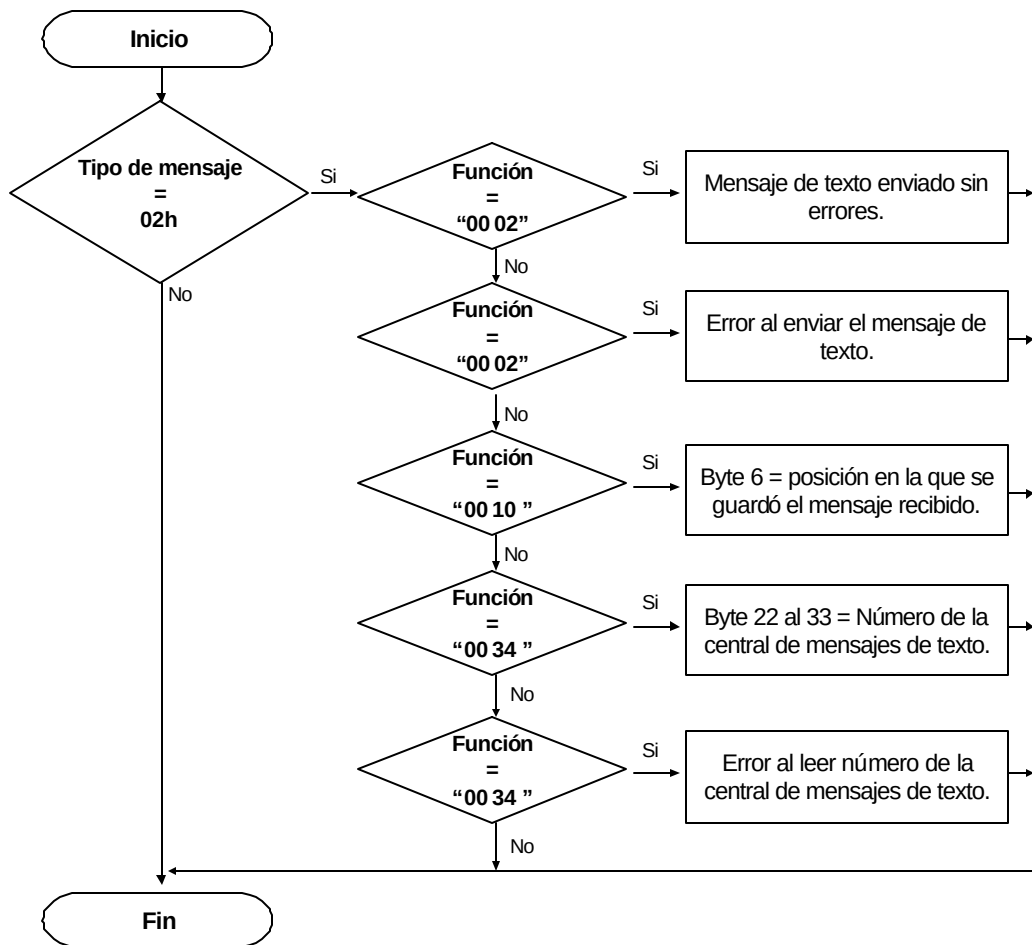


Figura 4.9. Obtención de los datos útiles del tipo de mensaje 02h

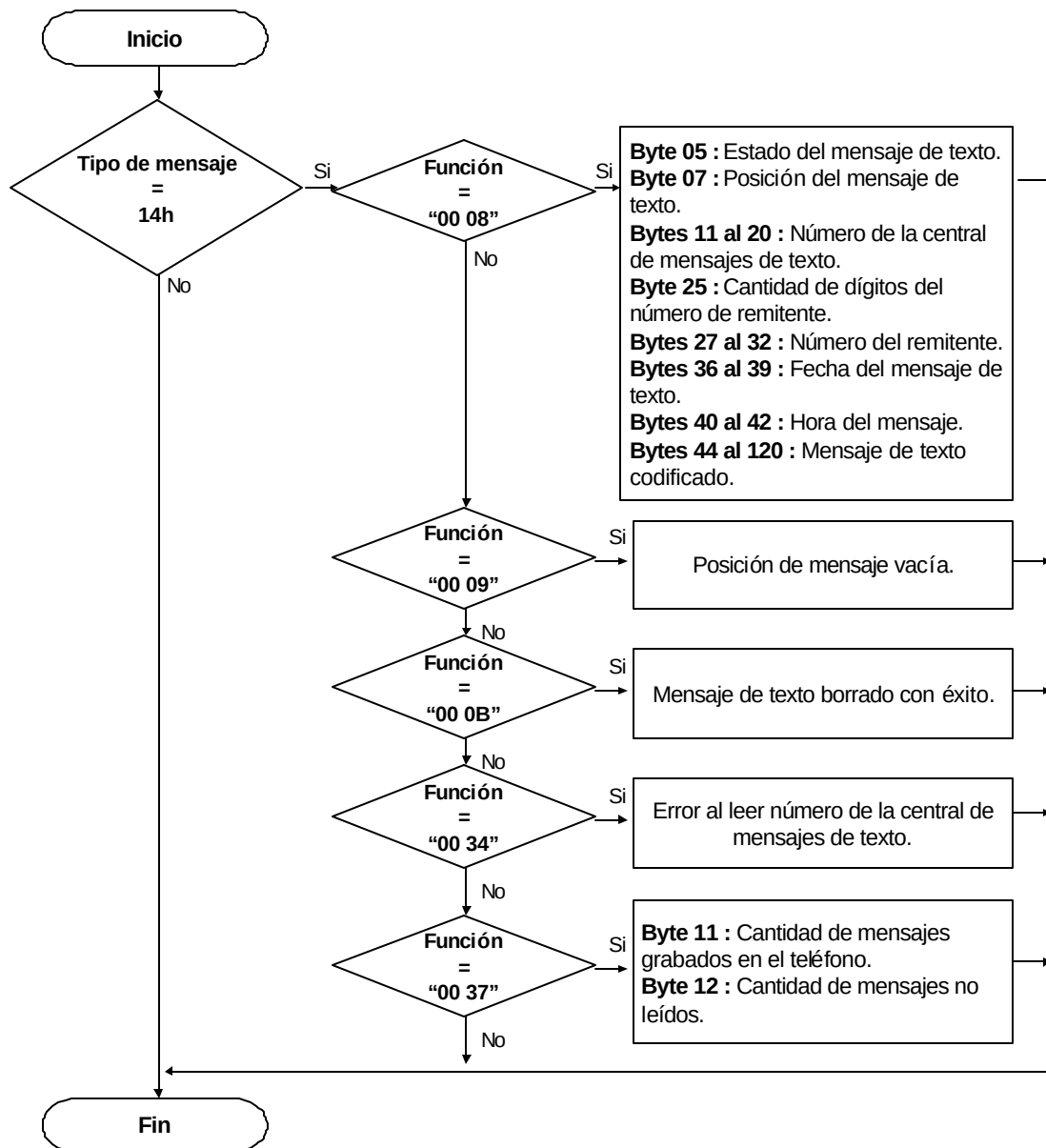


Figura 4.10. Obtención de los datos útiles del tipo de mensaje 14h

Cabe resaltar que en el caso de lectura de mensajes de texto, los valores obtenidos dentro del bloque de datos se encuentran codificados y deben ser decodificados antes de utilizarse.

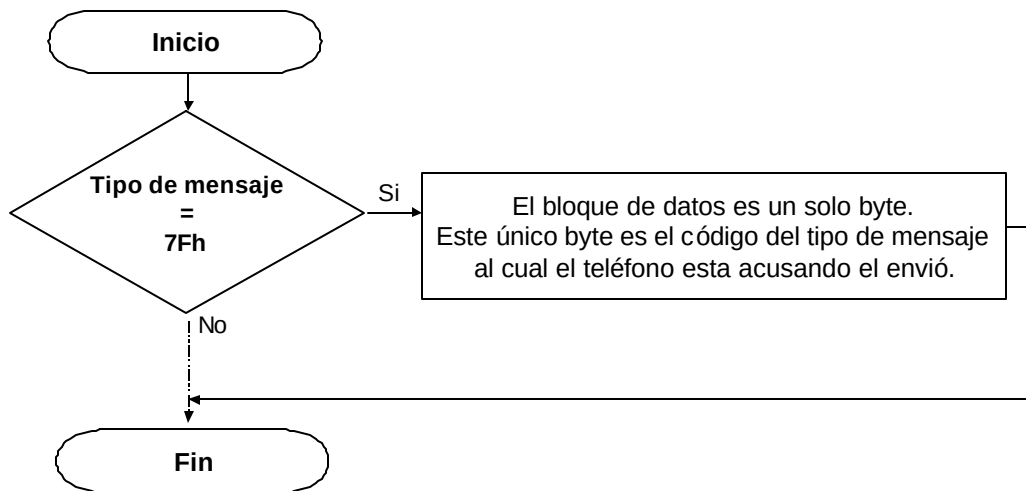


Figura 4.11. Obtención de los datos útiles del tipo de mensaje 7Fh

4.3.2 Generalización de tramas para solicitar acciones al teléfono celular

Con la generalización de las tramas para solicitudes realizada en el capítulo anterior (ver ítem 3.3.2) se implementó funciones a las cuales sólo es necesario proporcionar algunos valores para realizar las solicitudes al teléfono. La lista de funciones desarrolladas pueden ser apreciadas en la sección apéndice.

4.4 Capa de aplicación

Se desarrolló dos prototipos de los cuales el primero, denominado *Matrícula* (ítem 5.1), interactúa con bases de datos Microsoft SQL Server 2000, y el segundo, denominado *Monitor – Supervisor* (ítem 5.2), para aplicaciones de telecontrol y envío de alertas; en ambos casos la interacción entre el usuario y el prototipo se realiza por medio de mensajes de texto desde teléfonos celulares. Los prototipos se encuentran descritos en el capítulo V.

V. APLICACIONES PROTOTIPO PARA DEMOSTRAR LA UTILIDAD DE LA IMPLEMENTACIÓN DEL PROTOCOLO

La implementación del protocolo FBUS descrito en los ítems precedentes, ha dado la posibilidad de tener al teléfono celular como un periférico adicional a la computadora. La potencialidad que con este dispositivo adquiere la computadora es muy grande ya que las aplicaciones son diversas y están a disposición e imaginación del usuario. Así por ejemplo, es posible transmitir mediante mensajes de texto, señales de alarmas activadas por sistemas de seguridad instalados en casas u otros recintos, o señales que contengan información sobre el estado de funcionamiento de dispositivos y equipos electrónicos operando en la industria u hospitales.

Para demostrar la validez y flexibilidad de las aplicaciones que se pueden realizar con el protocolo implementado, se han realizado en Visual Basic 6.0 dos ejemplos prototipos que con posterioridad pueden ser, si fuera el caso, perfeccionados por cualquier investigador interesado en el tema. Se trata de:

- Matrícula vía mensajes de texto.
- Supervisión y corrección de un proceso industrial vía mensajes de texto.

5.1 Matrícula vía SMS (Short Message Service)

Este ejemplo, será aplicable para alumnos que estén dispuestos a pagar un costo adicional por su matrícula ya que desde la facultad, la computadora continuamente estaría respondiendo con mensajes a los usuarios. Se justifica el uso de este prototipo para aquellos alumnos que por una u otra razón no puedan realizar su matrícula personalmente, como por ejemplo al encontrarse en lugares inaccesibles del país o en el extranjero o cualquier otro caso similar. La matrícula por Internet es también otra alternativa, pero el teléfono celular es un servicio adicional y en algunos casos el único con el que contaría el alumno según su situación geográfica o de comodidad.

Los mensajes de texto enviados por los alumnos desde un teléfono celular remoto serán recibidos por un teléfono celular local. La interfaz gráfica, el teléfono celular y la computadora del servidor de base de datos pueden constituirse en una sola unidad o estar

separados en caso de configurarse una red local de recepción de mensajes. Luego que se ha recibido el mensaje en un formato establecido, donde figure el código de alumno, número del recibo de pago, y los cursos de interés, el sistema debe ser capaz de responder al usuario con otro mensaje de texto a fin de comunicarle si su matrícula es procedente y darle alternativas de solución. Se ha medido el tráfico que se puede soportar en los casos más críticos, lo que constituye las limitaciones del sistema.

Para el envío de ordenes desde los teléfonos celulares hacia el servidor se creó comandos para que el usuario los envíe a través de mensajes de texto, los cuales son:

"MAT" o "mat"	:	Matricular en curso(s)
"ELIM" o "elim"	:	Eliminar curso(s) matriculado(s).
"REP" o "rep"	:	Listar cursos matriculados.
"FINALIZAR" o "finalizar"	:	Finalizar proceso de matrícula.

Las cabeceras del formato del mensaje son las siguientes:

CODALUMNO COMPPAGO COMANDO CANTCURSOS CURSO1 GRUPOCUR1 CURSO2
GRUPOCUR2...

En Donde:

CODALUMNO	:	Código del alumno.
COMPPAGO	:	Número del comprobante de pago.
COMANDO	:	Acción a realizar.
CANTCURSOS	:	Cantidad de cursos contenidos dentro del mensaje.
CURSO1	:	Código del primer curso.
GRUPOCUR1	:	Grupo elegido para el primer curso.
CURSO2	:	Código del segundo curso.
GRUPOCUR2	:	Grupo elegido para el segundo curso.

Así mismo, se registró en la tabla correspondiente de la base de datos las abreviaciones de los nombres de los cursos disponibles para matrícula de la siguiente manera:

CI	:	Circuitos Eléctricos I.
CII	:	Circuitos Eléctricos II.
CEI	:	Circuitos Electrónicos I.
CEII	:	Circuitos Electrónicos II.

ME : Máquinas Eléctricas.

CDI : Circuitos Digitales I.

CDII : Circuitos Digitales II.

Entre otros.

Ejemplo:

972941 14002832 mat 3 CDI 2 CEII 2 ME 2

Esta instrucción inscribe al alumno cuyo código es 972941 en los cursos CDI grupo 2, CEII grupo 2, ME grupo 2. Si la matrícula procede, el alumno recibirá un mensaje indicándole los cursos en los que se ha matriculado. En caso de haber grupos llenos en el mensaje se visualizará que cursos debe cambiar.

972941 14002832 elim 3 CDI 2 CEII 2 ME 2

Esta instrucción elimina al alumno 972941 de los cursos CDI grupo 2, CEII grupo 2, ME grupo 2.

El alumno al terminar la matrícula debe enviar el comando finalizar de la siguiente manera.

CODALU COMPPAGO FINALIZAR

Ejemplo:

972941 14002832 finalizar

Una vez finalizada la matrícula, el alumno ya no puede solicitar reportes a través de mensajes de texto.

5.1.1 Descripción de la interfaz gráfica

La interfaz gráfica, consta de tres ventanas denominadas Conexión, Mensajes de Texto y Consultas, y es de uso exclusivo del administrador del sistema. Posee las siguientes características generales:

- Permite realizar la matrícula de alumnos mediante mensajes de texto almacenando la información en una base de datos.

- Verifica la seguridad de los datos por medio del número del comprobante de pago y código del alumno.
- Brinda la posibilidad de realizar consultas a la base de datos mediante instrucciones SQL.
- Guarda en un archivo histórico todos los mensajes de texto recibidos y enviados.

5.1.1.1 Ventana Conexión

En esta ventana mostrada en la figura 5.1 se realiza y configura la conexión con el servidor de base de datos y el teléfono celular, consta de tres secciones denominadas Base de datos, Autorización y Teléfono.

Figura 5.1. Ventana Conexión

Dentro de la sección Base de datos existen dos casillas de texto denominadas *Servidor*, en donde se especifica el nombre del servidor Microsoft SQL que va a ser utilizado y *Base de datos*, en donde se especifica el nombre de la base de datos a utilizar.

Dentro de la sección *Autorización*, se especifica el tipo de validación a utilizar ya sea la de un servidor Windows NT o la del servidor SQL. Los botones *Conectar* y *Desconectar* sirven para conectarse o dejar de hacerlo con el servidor SQL.

En la Sección Teléfono existe un objeto *Com*, en el cual se indica el número del puerto serial al que se encuentra conectado el teléfono celular. Mediante el botón *Probar comunicación* se puede realizar una prueba de conexión con el teléfono celular. El programa comprueba si el teléfono celular se encuentra conectado al sistema, si durante 10 segundos no se recibe respuesta se procederá a indicar el error.

5.1.1.2 Ventana Mensajes de texto

En esta ventana mostrada en la figura 5.2 se visualizan los mensajes de texto recibidos por el prototipo.

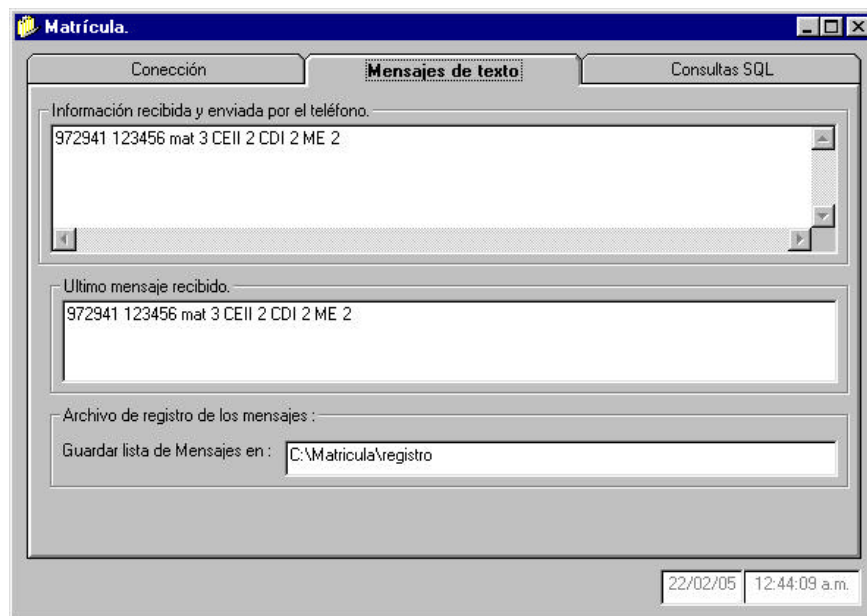


Figura 5.2. Ventana Mensajes de texto

En la casilla de texto *Información recibida y enviada por el teléfono*, se visualizan todos los mensajes recibidos y enviados por el prototipo. La casilla de texto *Último mensaje recibido*, visualiza el último mensaje atendido por el prototipo. En la sección *Archivo de registro de los mensajes*, se configura la ruta del archivo donde guardará todos los mensajes recibidos y enviados.

5.1.1.3 Ventana Consultas SQL

Esta ventana mostrada en la figura 5.3 sirve para realizar consultas al servidor de base de datos mediante instrucciones SQL.

En la casilla de texto *Consulta*, se puede escribir la instrucción SQL pertinente, mediante el botón *Ejecutar*, se envía la instrucción al servidor SQL, el resultado de la consulta se visualiza en la casilla de texto *Resultados*.

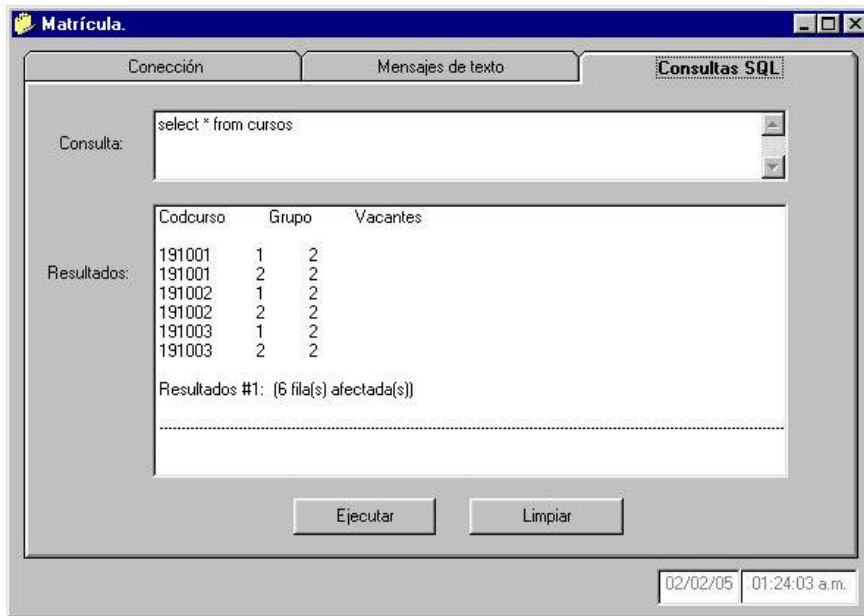


Figura 5.3. Ventana Consultas SQL

5.1.2 Pruebas del prototipo de Matrícula

El prototipo ha sido probado simulando un proceso de matrícula en el que intervinieron tres usuarios para lo cual se limitó la cantidad de cupos por curso y grupo a dos vacantes. Durante la prueba, los tres usuarios intentaron matricularse en los mismos grupos y cursos; el prototipo al detectar grupos llenos, ofreció a los usuarios la información correcta acerca del estado de su matrícula y los cambios que debía realizar, obteniéndose los resultados esperados. La figura 5.4 muestra las últimas operaciones realizadas durante la prueba del prototipo de matrícula.

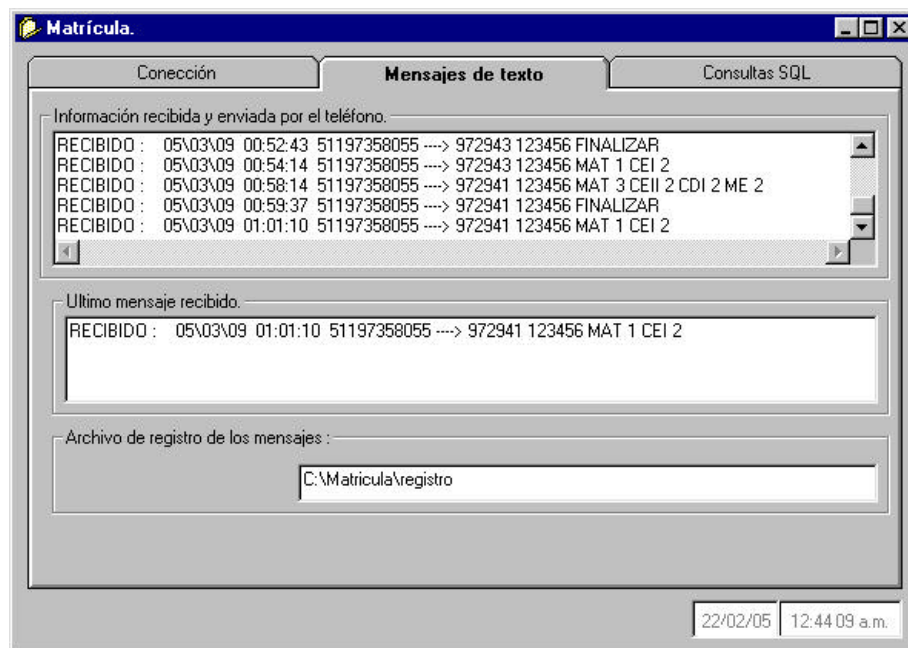


Figura 5.4. Operaciones realizadas en el prototipo de matrícula vía SMS.

5.1.3 Limitaciones

Cuando el servidor de base de datos se encuentra instalado en la misma computadora que contiene la interfaz gráfica, conformando una sola unidad, la atención por usuario durante el proceso de matrícula vía SMS demora aproximadamente de 1 a 2 segundos después de que el mensaje ha sido recibido. Este tiempo podría incrementarse de acuerdo al tráfico existente en la red de telefonía celular.

Con respecto al tipo de servicio para el teléfono celular que se utilice, se debe tener en cuenta que cuando se envían mensajes de texto (SMS) a un teléfono móvil, desde cualquier otro teléfono, éstos quedan almacenados en el servidor de mensajes de texto del operador móvil hasta que puedan ser entregados o descargados. La capacidad que tienen los servidores y por cuanto tiempo quedan almacenados dependerá tanto del fabricante del servidor como de la configuración que haga el operador del mismo.

Información obtenida del servicio de atención al cliente de Comunicaciones Móviles del Perú S.A. (Ex Bellsouth – Perú) refiere lo siguiente para el caso del funcionamiento de su centro de servicio de mensajes de texto:

Si el destinatario no se encuentra disponible, no recibirá el mensaje, por lo que la central de mensajes lo guardará y reintentará el envío cuando este se encuentre disponible, entonces se podrá recibir el mensaje. El tiempo que el mensaje permanecerá guardado será de ocho (08) horas, después de este tiempo, ya no podrá recibir el mensaje. Al cabo de las ocho (08) horas, el sistema borra (o expira) el mensaje que tenía guardado. Esto se llama Tiempo de Expiración del Mensaje = 8 horas, para el caso de los mensajes enviados desde la páginas web, el tiempo de expiración es menor.

La cantidad de mensajes que pueden ser guardados en la central, depende del tipo de servicio de telefónica celular. El almacenamiento estándar es de 100 mensajes de texto aproximadamente por equipo celular.

En teléfonos celulares GSM como el utilizado, el tiempo de expiración del mensaje también conocido como Periodo de Validez puede ser configurado por el usuario para tiempos de 1, 6, 24 horas, 3 días, 1 semana y tiempo máximo (los teléfonos celulares por lo general vienen configurados para un periodo de validez de 24 horas para los mensajes de texto). Así mismo, la cantidad de mensajes que pueden ser guardados en el centro de servicio, dependerá del tipo de servicio que tenga el teléfono celular (pre pago, post pago, etc.)

5.2 Supervisión y control de un proceso industrial vía SMS

En este caso conforman la unidad, la PC, el teléfono celular local y la tarjeta I/O, (descrita en el ítem 1.6) para comunicarse con el proceso industrial a supervisar. Los mensajes enviados desde un teléfono celular remoto hacia el teléfono celular local son transformados en ordenes para el proceso mediante el servidor local y la tarjeta I/O. Recíprocamente las señales de alarma provenientes del proceso son monitoreadas con la tarjeta I/O para que el servidor local las envíe como mensajes de texto hacia el teléfono celular remoto.

El sistema ha sido probado con la unidad de adsorción de gases en alto vacío de la Facultad de Ingeniería Química y con el modulo de nivel de líquidos del Laboratorio de la Facultad de Ingeniería Electrónica, ambos de nuestra Universidad. En los dos casos, se ha utilizado la misma interfaz gráfica que será detallada previamente.

5.2.1 Descripción de la interfaz gráfica

La interfaz gráfica es de uso exclusivo del administrador del sistema, consta de cuatro ventanas denominadas Comunicación, Nombres de I/Os, Notificaciones y Valores I/O que serán descritos en los ítems 5.2.1.1 a 5.2.1.4. Posee las siguientes características:

- Asigna a variables de entradas y salidas así como sus estados de actividad y no actividad, con nombres específicos. Así es posible definir por ejemplo, a la salida O7 de la tarjeta I/O con el nombre *de faja* pues se encuentra conectado a la línea de control de una faja transportadora, y definir sus estados de actividad como *corriendo* o *detenida* o cualquier otro nombre que se crea conveniente.
- Interpreta las órdenes recibidas en los mensajes de texto y de acuerdo a ellas, activa las salidas de la tarjeta I/O respectivas, asimismo al detectar cambios en las entradas de la tarjeta I/O envía mensajes de texto al número de teléfono celular que haya sido configurado notificando el hecho.
- Restringe o no los números telefónicos a los que el prototipo obedecerá.
- Presenta los mensajes recibidos en una ventana y los graba en un archivo detallándose el número del remitente, la fecha, hora, el mensaje recibido y si es o no un número válido para la ejecución de la orden.

5.2.1.1 Ventana Comunicación

En esta ventana mostrada en la figura 5.5 se configura los puertos para que la interfaz gráfica se comunique con el teléfono celular y con la tarjeta I/O.

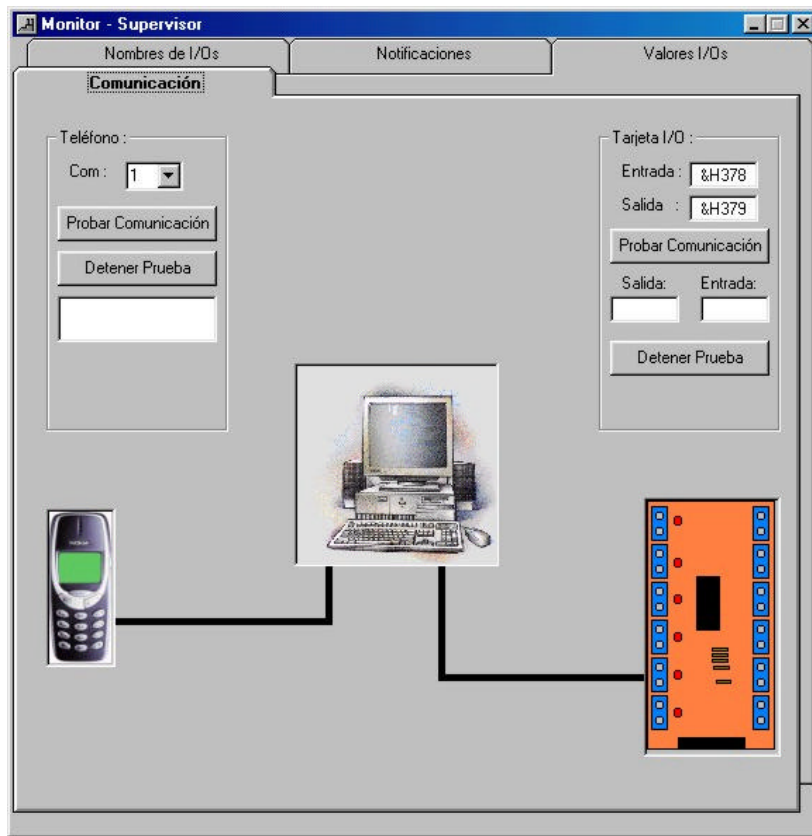


Figura 5.5. Ventana Comunicación

En la sección *Teléfono*, por medio del objeto desplegable *COM*, se selecciona el número del puerto serial utilizado para la comunicación con el teléfono celular. El botón llamado *Probar Comunicación*, realiza la lectura del directorio telefónico existente en el teléfono celular.

En la sección llamada *Tarjeta I/O*, se configuran los números de las direcciones para los puertos de entrada y salida de la tarjeta I/O. Sólo es necesario escribir las direcciones de entrada y salida en el cuadro de texto correspondiente (entrada o salida).

Mediante el botón *Probar Comunicación*, la interfaz gráfica lee el valor existente en el puerto de entrada de la tarjeta y simultáneamente envía por el puerto de salida valores que van desde 0 hasta 255 probando las salidas. Los valores leídos y enviados son visualizados en las casillas de texto *Entrada* y *Salida* de la parte inferior de esta sección. Al presionar el botón *Detener Prueba*, la lectura y escritura en los puertos de entrada y salida se detienen. Es necesario que esta prueba sea activada sin conectar la tarjeta al proceso.

5.2.1.2 Ventana Nombres de I/Os

En esta ventana mostrada en la figura 5.6 se configura la identificación literal o numérica de las entradas y salidas con sus estados de actividad. Inicialmente las entradas están identificadas desde I0 a I7, mientras que las salidas desde O0 a O7, los estados de actividad están definidos como 1 y los de no actividad como 0. La configuración puede ser cambiada en cualquier momento, cuando esto sucede, los cambios son actualizados automáticamente en toda la interfaz gráfica.

Es posible, grabar en archivos los nombres configurados con sólo indicar la ruta para el archivo y presionar el botón *Guardar Configuración* de la parte inferior. El archivo se guardará bajo la extensión CFG. Para recuperar el contenido del archivo se debe hacer un clic en el nombre del mismo (dentro de la lista de archivos que se pueden recuperar).

Entradas :			Salidas :				
Entradas :	Activo :	No Activo :	Salidas :	Activo :	No Activo :		
I0 =	Tanque1	Lleno	Vacio	O0 =	Motor1	Encendido	Apagado
I1 =	Tanque2	Lleno	Vacio	O1 =	Motor2	Encendido	Apagado
I2 =	Alarma1	Activa	Inactiva	O2 =	Motor3	Encendido	Apagado
I3 =	Alarma2	Activa	Inactiva	O3 =	Alarma	Activa	Inactiva
I4 =	Sensor1	Activo	Inactivo	O4 =	Luz	ON	OFF
I5 =	Sensor2	Activo	Inactivo	O5 =	Sistema1	Activo	Inactivo
I6 =	Motor1	RUN	STOP	O6 =	Generador	Encendido	Apagado
I7 =	Motor2	RUN	STOP	O7 =	Bomba	Activa	Inactiva

Salvar/Cargar Configuración :

Guardar :

C:\Monitor - Supervisor\IOdemo

Guardar Configuración

Cargar :

IO.CFG
IOdemo.CFG

Figura 5.6. Ventana Nombres de I/Os

5.2.1.3 Ventana Notificaciones

En la figura 5.7, se puede apreciar esta ventana, la cual tiene cuatro secciones; *Notificación para el estado de las Entradas*, *Validación de solicitudes*, *Acuse de Envíos* y *Salvar/Cargar Configuración*, que se describen a continuación:

- *Notificación para el estado de las Entradas*, en esta sección se configura las condiciones para las cuales se enviará una notificación a un número telefónico dependiendo de la activación de una de las variables de entrada. Para esto sólo hay que activar el check del lado izquierdo de la variable, indicar cual es el estado que debe cumplir y el número telefónico al que se debe realizar la notificación.
- *Validación de solicitudes*, aquí se configura la lista de números telefónicos habilitados para enviar mensajes con órdenes remotas para el proceso. También es posible activando el check *Aceptar todos los números*, que se acepte mensajes con ordenes de cualquier número telefónico.
- *Acuse de Envíos*, esta sección es para definir una palabra clave con la cual, la interfaz gráfica luego de recibir un mensaje con órdenes válidas, confirme al usuario la recepción y ejecución de la orden. Se incluyó una casilla de texto para anteponer un dígito a los números que realicen las notificaciones para casos en los que pueda ser necesario agregar código de ciudad o país, etc.
- *Salvar/Cargar Configuración*, esta sección es para grabar o recuperar listas de números telefónicos validos y configuraciones de notificaciones. Para guardar la configuración de las notificaciones o de los números telefónicos validos, se debe indicar en las casillas de texto correspondientes la ruta donde se va a grabar el archivo (Notificaciones o Lista de Números Validos) y presionar el respectivo botón *Guardar*. El archivo de notificaciones generado es guardado con la extensión NOT y NUM para el caso de la lista de números telefónicos validos. Para recuperar el contenido de un archivo de notificaciones o de lista de números telefónicos sólo se debe seleccionar el nombre del archivo de la lista correspondiente.

Monitor - Supervisor

Comunicación

Nombres de I/Os

Notificaciones

Valores I/Os

Notificación para el estado de las Entradas :

Entradas :	Cuando :	Al número :
☐ TANQUE1	=	-----
☐ TANQUE2	=	-----
☒ ALARMA1	= ACTIVA	96534126
☐ ALARMA2	=	-----
☒ TANQUE	= VACIO	97635229
☐ SENSOR2	=	-----
☐ MOTOR1	=	-----
☐ MOTOR2	=	-----

Validación de solicitudes :

Números Aceptados para recibir comandos :

972415245	972424345	972424345
973145345	972758645	972125365
978249705	972424345	972424345
972400045	981345345	978754835
-----	-----	-----
-----	-----	-----
-----	-----	-----
-----	-----	-----
-----	-----	-----

☐ Aceptar todos los números.

Acuse de Envíos :

Activar acuse de envío al recibir la palabra : VER

Anteponer número al notificar comando recibido :

Salvar/Cargar Configuración :

Notificaciones :

C:\Monitor - Supervisor\notificaciones

Guardar

Lista de Números :

C:\Monitor - Supervisor\numeros

Guardar

Cargar Notificaciones :

notificaciones.NOT

Cargar lista de Números :

numeros.NUM

Figura 5.7. Ventana Notificaciones

5.2.1.4 Ventana Valores I/Os

En la ventana que se muestra en la figura 5.8 se visualiza los estados actuales de las entradas y salidas de la tarjeta I/O, así como también la lista de mensajes de texto que se ha recibido. Contiene las siguientes secciones:

- *Entradas*, en esta sección se visualiza el estado de los bits del puerto de entrada de la tarjeta I/O.
- *Salidas*, en esta sección se visualiza el estado de los bits del puerto de salida de la tarjeta I/O.
- *Información recibida y enviada por el teléfono*, en esta sección es donde se visualiza todos los mensajes de texto que han sido enviados y recibidos. La lista indica el número

telefónico de donde proviene el mensaje o a quien ha sido enviado, la fecha, hora y si el número es válido.

- *Último mensaje recibido*, en esta sección se visualiza el último mensaje de texto que se ha recibido.
- *Archivo de registro de los mensajes*, en esta sección se indica la ruta donde se encuentra el archivo de registro de todos los mensajes enviados y recibidos. Es posible cambiar el nombre del archivo para crear nuevos registros en cualquier momento sin cerrar la interfaz grafica. Los archivos de registro de mensajes se generan con la extensión SAV.

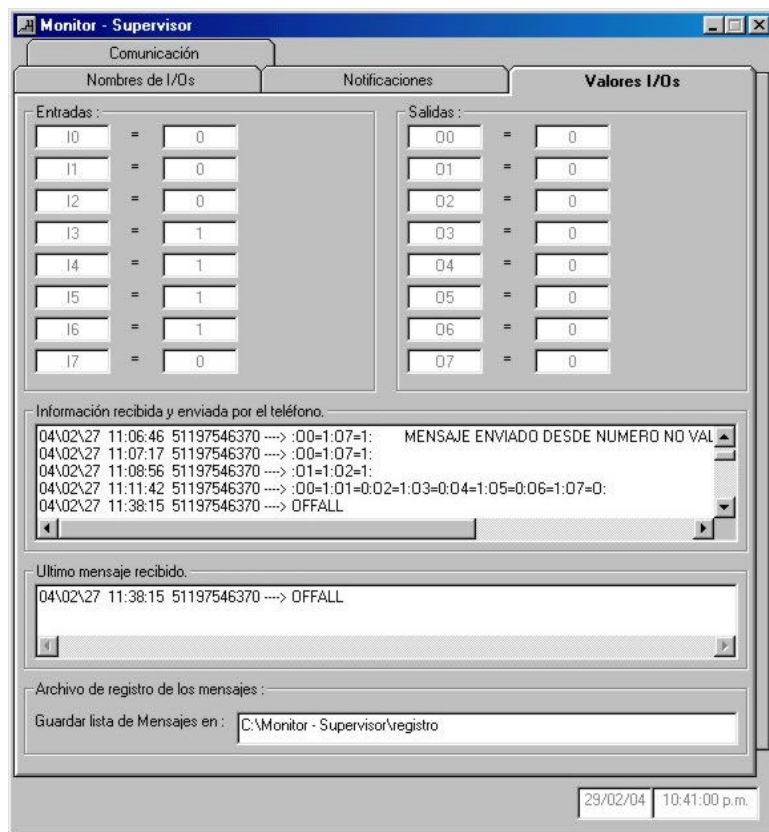


Figura 5.8. Ventana Valores I/Os

5.2.2 Formato para el envío de ordenes

Las cabeceras del formato del mensaje para el envío de órdenes remotas es el siguiente:

ACUSE:VARIABLE1=VALOR1:VARIABLE2=VALOR2:VARIABLE3=VALOR3:.....
.....:VARIABLE8=VALOR8:

O simplemente para activar una sola de las salidas:

ACUSE:VARIABLE=VALOR:

En ambos casos, el parámetro acuse (que se configura en la ventana de Notificaciones explicada en el ítem 5.2.1.3) es opcional y sirve para que la interfaz gráfica envíe un mensaje de confirmación al ejecutar la orden recibida. Los valores correspondientes a los parámetros VALOR y VARIABLE son los que se configuran en el programa dentro de la ventana Nombres de I/Os (descrito en el ítem 5.2.1.4).

Se dispone de comandos de acción rápida para casos de emergencia los cuales son:

ONALL	realiza la activación de todas las salidas.
OFFALL	realiza la desactivación de todas las salidas.

Para este caso también es posible activar la función de acuse de envío, de la siguiente manera:

ACUSE:OFFALL
ACUSE:ONALL

5.2.3 Pruebas del prototipo Monitor-Supervisor

Se realizó dos pruebas de funcionamiento, una en la Facultad de Ingeniería Química de la Universidad Nacional Mayor de San Marcos conectando el sistema a un proceso llamado Unidad de adsorción de gases en alto vacío y la segunda en el Laboratorio de Ingeniería Electrónica conectando el sistema al módulo de nivel de líquidos existente. En ambas pruebas las cuales son descritas en los ítems 5.2.3.1 y 5.2.3.2 se obtuvo los resultados esperados.

5.2.3.1 Con la Unidad de adsorción de gases en alto vacío de la Facultad de Ingeniería Química

La unidad de Adsorción de gases en alto vacío es un proceso utilizado por la Facultad de Ingeniería Química para realizar pruebas de medición del área de adsorbentes y obtener insumos para procesos de refinación petroquímica con la finalidad de prestar servicios a terceros a fin de generar recursos propios para esa Facultad. Los materiales necesarios para el funcionamiento de la unidad así como las muestras son de costo elevado y quedan inservibles al producirse alguna falla por tiempo prolongado. Los usuarios manifestaron que sus experimentos demoran aproximadamente 72 horas, tiempo durante el cual, necesitan estar informados de la ocurrencia de alguna falla en la unidad, por este motivo se experimentó la forma de dar una alternativa de solución que brinde la posibilidad de apagar o encender remotamente algunos equipos a fin de salvar los materiales y muestras empleadas o al menos de mantener informado al usuario donde quiera que se encuentre de cualquier suceso producido durante la ejecución del proceso. La alternativa propuesta, fue colocar un sensor de corriente en la unidad difusora y un sensor de voltaje en la bomba de vacío (ambos de gran importancia para el funcionamiento de la unidad) con la finalidad de monitorearlos para mantener informado al usuario de cualquier falla en su funcionamiento. De igual manera, se planteó conectar un relé en serie a la unidad difusora a fin de poder apagarla en forma remota ante algún desperfecto.

En la figura 5.9, se muestra las conexiones de los sensores que fueron contruidos debidamente a fin de detectar el funcionamiento de la unidad difusora y la bomba de vacío. Al alimentar la unidad difusora, el flujo de corriente producido, generó en los terminales de entrada del sensor de corriente una pequeña caída de tensión la cual es amplificada a una tensión suficiente para ser leída por la entrada I6 de tarjeta I/O, mientras que en el caso del sensor de voltaje, al alimentarse la bomba de vacío, el voltaje recibido en sus terminales se reduce a una tensión de 5 voltios para ser entregada a la entrada I7 de la tarjeta I/O. De esta manera, las dos variables mas críticas del proceso pueden ser leídas por la PC por medio de la tarjeta I/O. También se conectó en serie a la línea de alimentación de la unidad difusora el relé de la entrada O0 de la tarjeta I/O con la finalidad de poder apagarla remotamente.

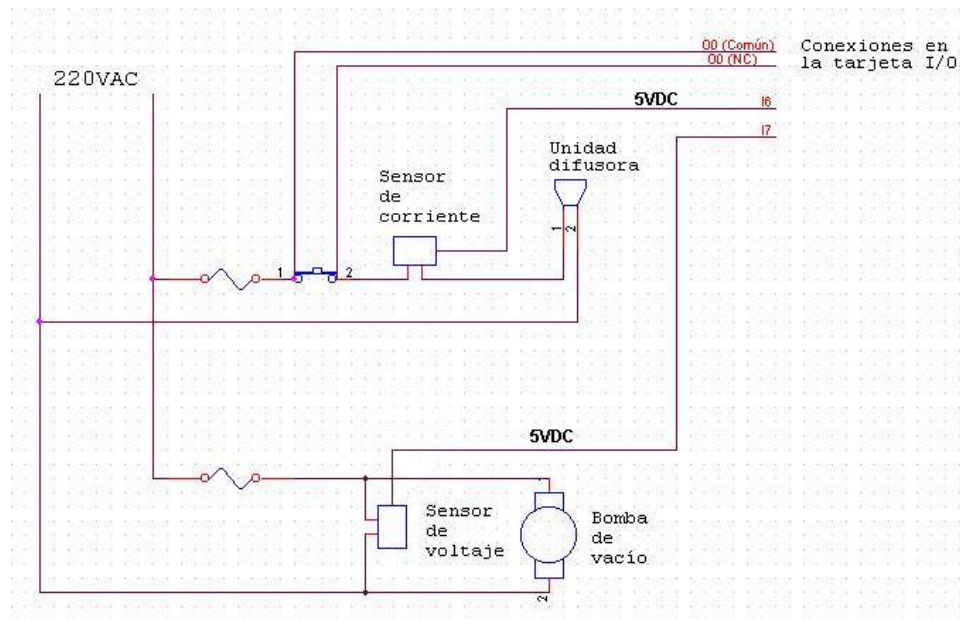


Figura 5.9. Conexión de la unidad de adsorción de gases con la tarjeta I/O

Se configuró la interfaz gráfica en la sección Nombres de I/Os y Notificaciones de la manera indicada en las figuras 5.10 y 5.11.

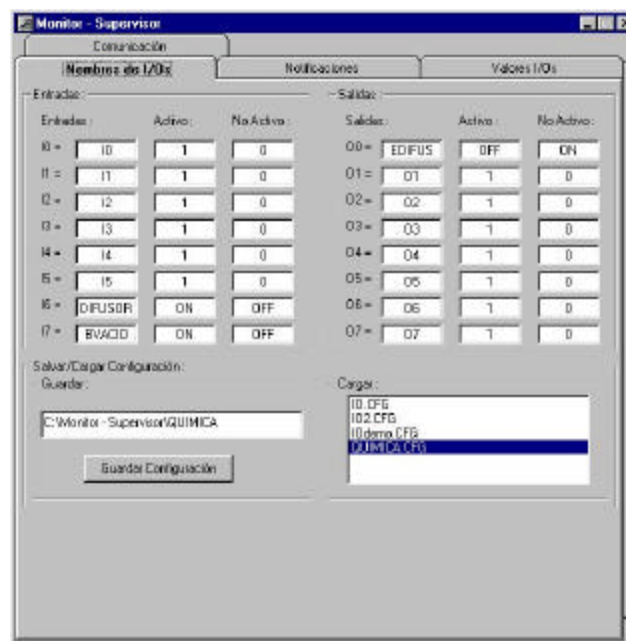


Figura 5.10. Configuración de la ventana Nombres I/Os

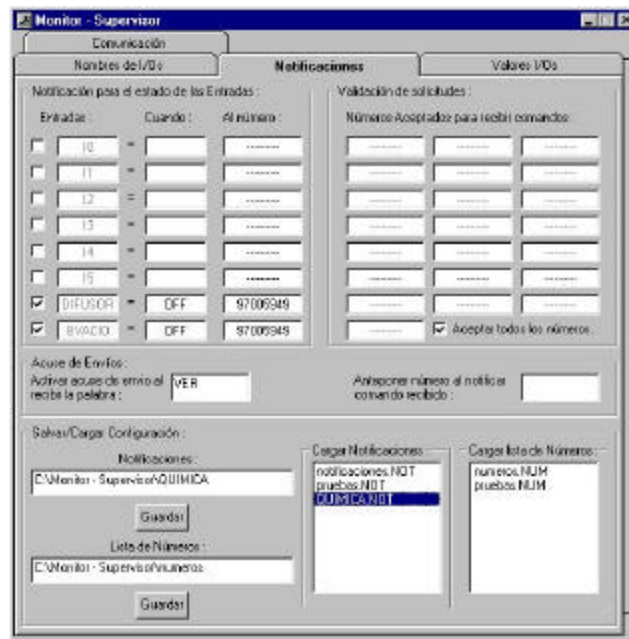


Figura 5.11. Configuración de la ventana Notificaciones

Al realizar la primera prueba de funcionamiento, se procedió a desconectar la unidad difusora (simulando una falla en dicha unidad), la interfaz gráfica detectó la desconexión del dispositivo por medio de la tarjeta I/O y procedió a realizar la notificación de manera correcta al número de celular remoto configurado (ver figura 5.12).

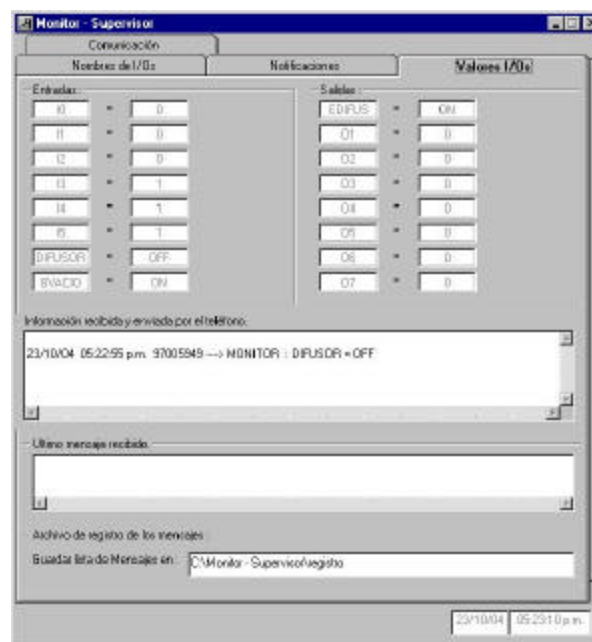


Figura 5.12. Prueba número 1 con la unidad de adsorción de gases en alto vacío

En una segunda prueba de funcionamiento, se procedió a desconectar la alimentación de la bomba de vacío, el sistema detectó el cambio de estado en la entrada respectiva de la tarjeta I/O procediendo a enviar la notificación al número telefónico configurado; una vez recibida la notificación de desconexión de la bomba de vacío mediante el mensaje de texto al usuario remoto, éste envió de vuelta otro mensaje al teléfono celular local ordenando la desconexión de la unidad difusora (:EDIFUS=OFF:); el mensaje fue recibido y se ejecutó desconectando la unidad difusora (ver figura 5.13).

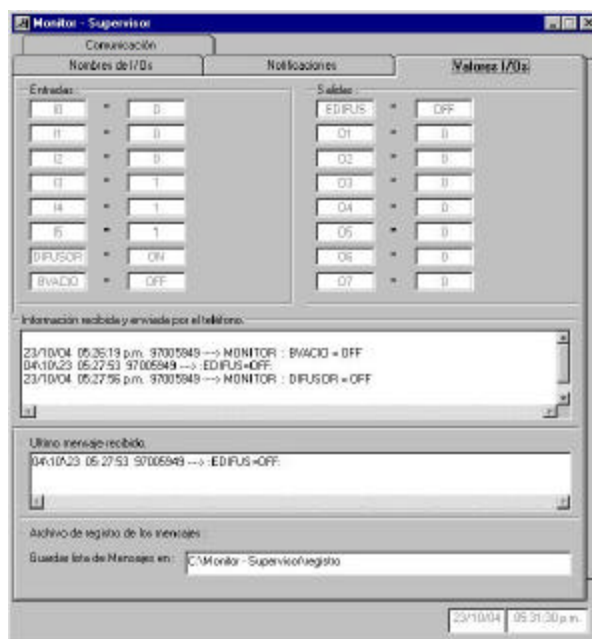


Figura 5.13. Prueba número 2 con la Unidad de Adsorción de gases en alto vacío

5.2.3.2 Con el módulo de nivel de líquidos del laboratorio de la Facultad de Ingeniería Electrónica

Para esta prueba se empleó el módulo de nivel de líquidos del laboratorio de la Facultad, el cual consiste en un conjunto de tanques, bombas de agua y electro válvulas interconectadas de manera conveniente permitiendo la circulación de líquidos en forma secuencial para simular procesos de mezcla de líquidos. El módulo está construido de tal manera que los sensores y actuadores pueden conectarse a un PLC. A fin de monitorear el proceso remotamente, se procedió a conectar las entradas y salidas de la tarjeta I/O a los pulsadores de arranque y parada del proceso así como también a una salida del PLC tal como se muestra en la figura 5.14.

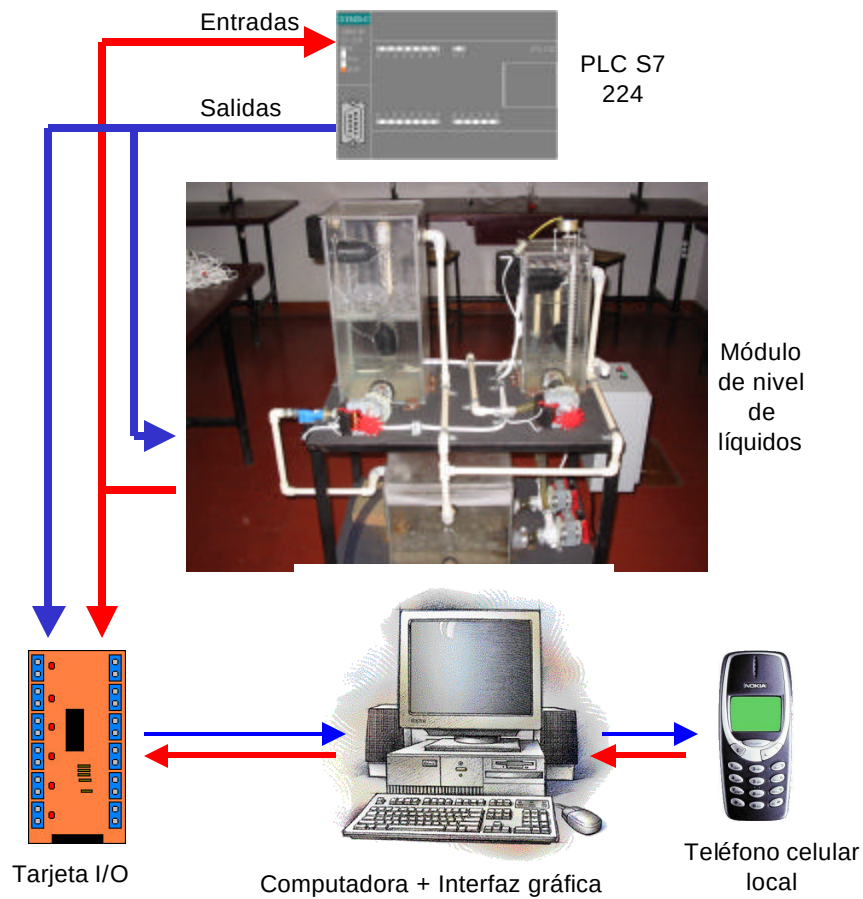


Figura 5.14. Interconexión entre los diversos elementos

Para esta prueba se procedió a realizar el inicio y parada remota del proceso mediante mensajes de texto. Además, se procedió a detectar la parada manual del proceso obteniéndose los resultados esperados en ambos casos.

Para la detección y notificación de la parada del módulo, se configuró en la ventana "Nombres I/Os" la entrada I7 como STOP, mientras que para el arranque y parada remota, se configuró I0 como START, I1 como STOP tal como se muestra en la figura 5.15, la figura 5.16 describe la configuración de la ventana "Notificaciones".

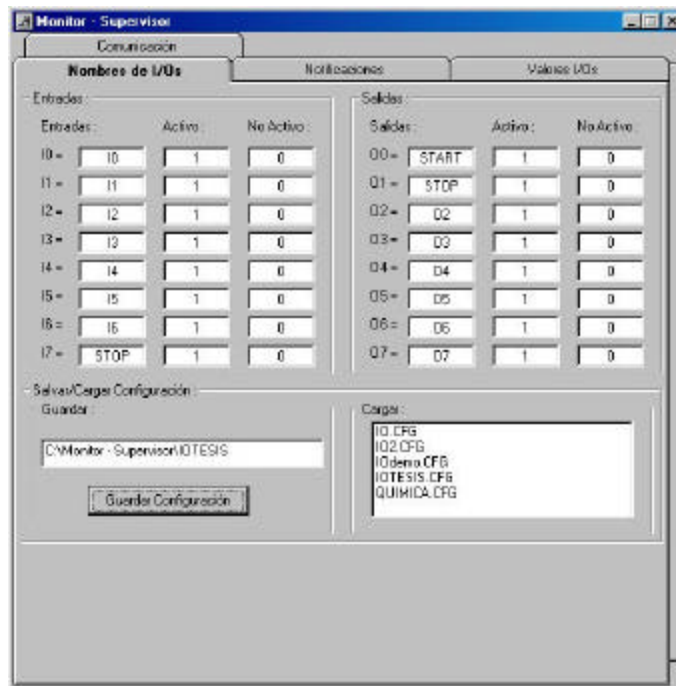


Figura 5.15. Configuración de la ventana Nombres I/Os

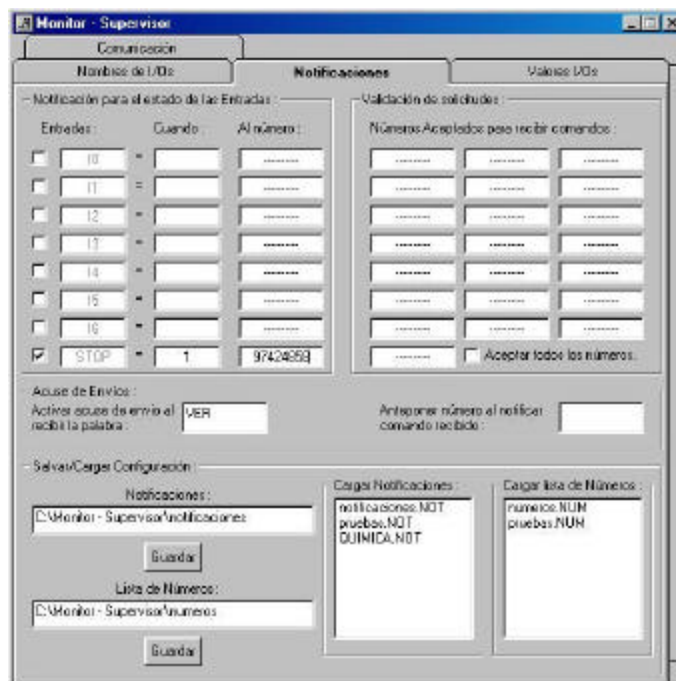


Figura 5.16. Configuración de la ventana Notificaciones

Durante la prueba, al ser presionado el botón de parada del proceso, el prototipo envió el mensaje de texto al número configurado indicando el hecho tal como se muestra en la figura 5.17.

Finalmente se envió la orden de parada (:STOP=1:) la cual detuvo el funcionamiento del proceso tal como se aprecia en la figura 5.19.

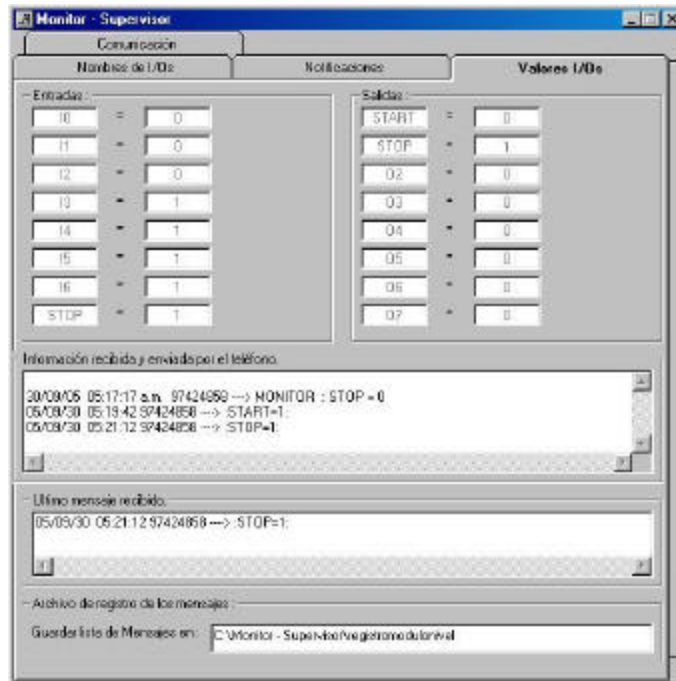


Figura 5.19. Parada remota del proceso

5.2.4 Limitaciones

El prototipo presenta las siguientes limitaciones:

- Sólo es posible monitorear 5 señales de entrada y controlar 5 variables de salida en estados de encendido y apagado (ON/OFF) con la tarjeta I/O implementada.
- En cuanto al tipo de servicio y al tipo de entrega del mensaje por parte de la central de mensajes de texto del operador móvil, las limitaciones son las mismas que para el prototipo Matrícula y que fueron detalladas en el ítem 5.1.4.

5.3 Procesamiento de los mensajes de texto en las aplicaciones descritas

El procesamiento de los mensajes de texto para las aplicaciones descritas en los ítems 5.1 y 5.2 se encuentra detallado en el diagrama de flujo de la figura 5.20. Cuando existen

mensajes de texto sin leer, el protocolo FBUS cambia el valor de la variable FlagSMSNuevosRecibidos de 0 a 1. Además existe un arreglo de tipo entero de 25 elementos llamado SMS(X) (donde X va de 1 a 25), el cual indica en que estado se encuentra cada posición de los mensajes de texto, asignando a cada posición de dicho arreglo los valores de 1 para los ya leídos, 3 para los no leídos y 7 para las posiciones vacías según sea el caso. Estas dos variables en conjunto permiten saber si existen mensajes nuevos y en que posición se encuentran grabados.

La sección del programa que se encarga de procesar los mensajes recibidos está realizada en base a un temporizador que cada 500 milisegundos comprueba el valor de la variable FlagSMSNuevosRecibidos y si la encuentra en 1, revisa cada una de las posiciones del arreglo SMS(X) verificando su valor para saber el estado de la posición de memoria. Si el programa detecta en alguna posición el valor 3 (mensaje no leído) procede a leer el mensaje de texto que se encuentra en dicha posición y analiza el contenido para extraer las órdenes remotas recibidas. Luego el mensaje de texto de esta posición es borrado y la implementación del protocolo FBUS, procede a cambiar su estado en el arreglo SMS(X) al valor 7 (posición vacía). Este proceso se repite hasta que no existan mensajes de texto nuevos grabados en el teléfono.

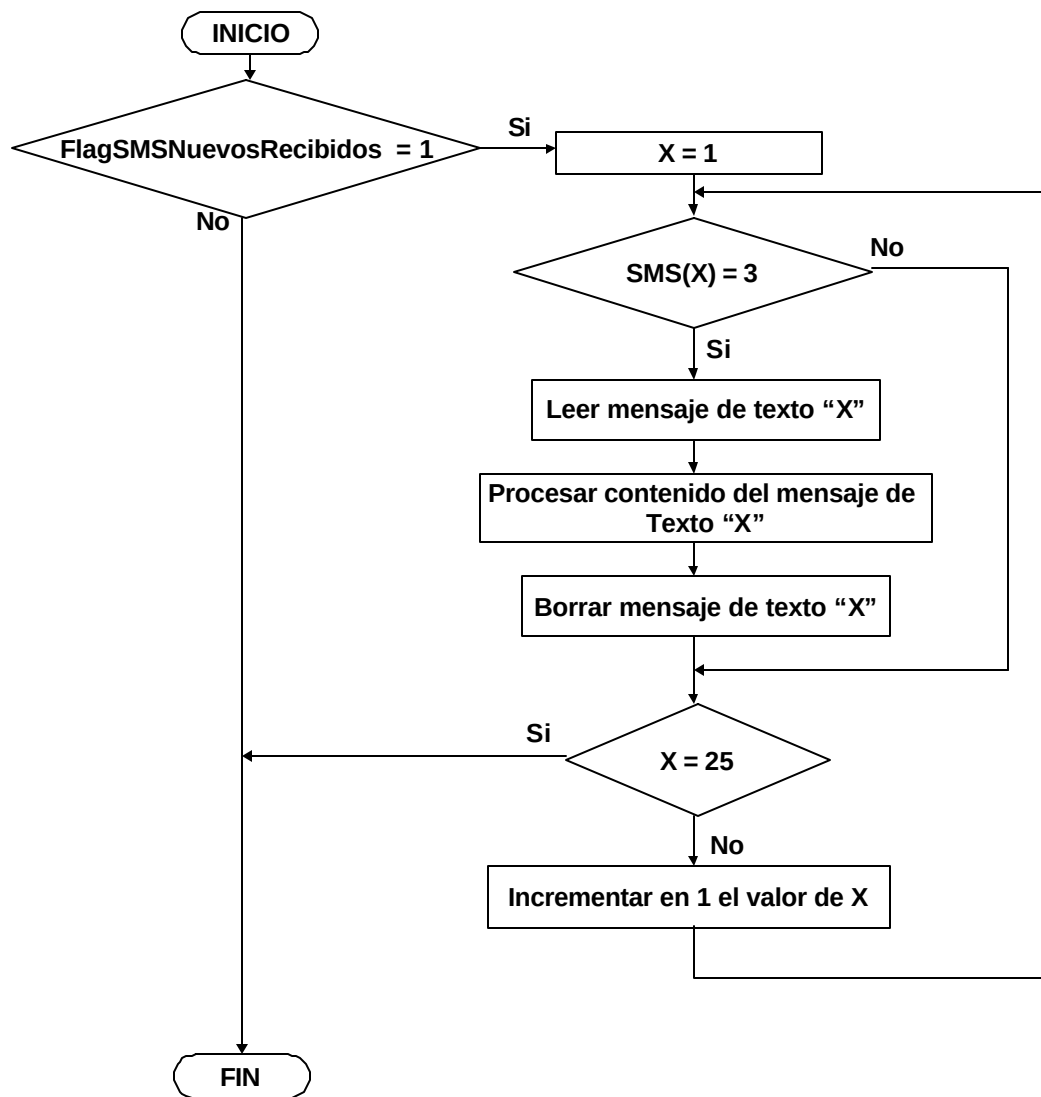


Figura 5.20. Procesamiento de los mensajes de texto

VI. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- Se ha logrado integrar al teléfono celular con la computadora mediante la implementación del protocolo FBUS, potenciando las aplicaciones de ambos dispositivos.
- Se ha validado el funcionamiento del protocolo implementado a través de aplicaciones prototipo.
- Se ha propuesto un procedimiento para el análisis y estudio del protocolo FBUS el cual se puede extender para analizar otros protocolos de comunicación.
- Se ha logrado documentar a través de las pruebas experimentales realizadas, la manera en que el teléfono celular intercambia información con la computadora mediante el protocolo FBUS detallando la función de los bytes dentro de cada tipo de trama.

6.2 Recomendaciones

- Siempre se debe tener en cuenta las limitaciones de este tipo de sistemas, las cuales fueron descritas en los ítem 5.1.3 y 5.2.4 del capítulo de pruebas de los prototipos de aplicación.
- Con los resultados del análisis y estudio del protocolo FBUS es posible implementar a este protocolo para ser utilizado en microcontroladores, esto permitiría desarrollar aplicaciones compactas que puedan ser llevadas a lugares pequeños sin depender de una computadora.
- Es conveniente continuar este estudio extendiéndolo a otros tipos de teléfonos celulares a fin de poder utilizar amplia variedad de modelos y marcas los cuales podrían trabajar con sistemas diferentes al GSM.
- Las aplicaciones que sean implementadas deben ser utilizadas en áreas donde el servicio de telefonía celular llegue sin problemas para asegurar el correcto funcionamiento.

VII. BIBLIOGRAFÍA

- [1] A. Manuel *et al.*, *Instrumentación Virtual*, Ediciones ALFAOMEGA 2002.
- [2] S. Wolf, R. Smith, *Guía Para Mediciones Electrónicas y Practicas de Laboratorio*, Pretince – Hall Hispanoamericana S.A. 1992.
- [3] A. Tanenbaum, *Redes de Computadoras*, Pretince – Hall México 1997.
- [4] B. Nagy, *et al.*, *Protocol used in Nokia phones*, The Linux Gnokii Project, Abril 2003.
<http://gnokii.org/> Fecha de acceso: Julio 2003.
- [5] B. Nagy, *et al.*, *Frames used in GSM/PCN Nokia 6110 and derivatives*, The Linux Gnokii Project. Abril 2003.
<http://gnokii.org/> Fecha de acceso: Julio 2003.
- [6] J. Gómez. *El Servicio SMS: un enfoque práctico*, Trabajo de doctorado para la asignatura “Nuevas tecnologías para las comunicaciones”, Escuela Técnica Superior de Ingeniería Informática UAM, Junio del 2002.
<http://www.learobotics.com/personal/juan/doctorado/sms/sms.html> Fecha de acceso: Noviembre 2004
- [7] European Telecommunications Standards Institute, *Digital cellular telecommunications system (Phase 2+); Alphabets and language-specific information (GSM 03.38) Versión 5.3.0*, ETSIS GSM Technical Specifications July 1996.
<http://webapp.etsi.org/key/queryform.asp> Fecha de acceso: Agosto 2003.
- [8] ©Nokia Mobile Phones Ltd, *Programmers After Market Services NHM-5NX Series Transeivers – System Module*, ©Nokia Mobile Phones Ltd 2000.
- [9] NuukiaWorld project, *FBUS & MBUS adapters Pictures*, NuukiaWorld project 2000 PUBLICACION ON-LINE.
<http://www.panuworld.net/nuukiaworld/hardware/cables/index.htm> Fecha de acceso: Mayo 2003.

- [10] ©Nokia Mobile Phones Ltd, *Programmers After Market Services NHM-5NX Series Transceivers – Service Tools*, ©Nokia Mobile Phones Ltd 2000.
- [11] SoftwareCave, *SMS Manager 1.1.3*.
<http://www.softwarecave.com> Fecha de acceso : Mayo 2003
- [12] R. Luna, *Programación en Visual Basic 6.0*, Editorial Macro 2002.
- [13] European Telecommunications Standards Institute, *Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point-to-Point (PP) (GSM 03.40) Versión 5.3.0*, ETSI GSM Technical Specifications July 1996.
<http://webapp.etsi.org/key/queryform.asp> Fecha de acceso: Agosto 2003.
- [14] R. Dutra. *Comunicação de dados em ambiente industrial: Um protocolo para automação e controle em tempo real*, Tesis de Maestria, Escuela de Ingenieria de San Carlos, Diciembre 1996.
- [15] B. Suarez, *SQL Server 2000*, Editorial Ritisa Graff 2004.

APÉNDICE

Rutinas implementadas para la comunicación FBUS

Para la implementación de los niveles de enlace y presentación de la comunicación FBUS, se implementó rutinas en Visual Basic las cuales realizan la interacción entre el teléfono celular y la computadora. La lista de rutinas implementadas se describe a continuación las cuales además se encuentran ordenadas de acuerdo al Módulo dentro del cual fueron programadas.

Módulo TextoaHex

Dentro de este módulo se encuentran las rutinas encargadas de la conversión de datos binarios a valores hexagesimales/código ASCII y viceversa, además de otra rutina encargada de la conversión de valores hexagesimales a valores enteros.

Rutina ConvierteStringaTextoaHex

Realiza la conversión de un tipo de dato binario a una cadena de caracteres que contiene el valor de los datos introducidos en formato Hexagesimal y en código ASCII. El formato de esta rutina es el siguiente:

ConvierteStringaTextoaHex(Datos_de_entrada)

Valores de entrada : Datos_de_entrada

Variable tipo cadena de caracteres que contiene los datos binarios que serán convertidos.

Valores de salida : ValorHex

Variable tipo cadena de caracteres que contiene los datos convertidos a valores hexagesimales.

ValorASC

Variable tipo cadena de caracteres que contiene los datos convertidos a código ASCII.

Rutina ConvierteTextoHexString_Y_Envia

Realiza la conversión de una cadena de caracteres que contiene valores de tipo Hexagesimal a datos de tipo binario y los envía por el puerto serial. El formato de esta rutina es el siguiente:

ConvierteTextoHexString_Y_Envia(TextoHex)

Valores de entrada : TextoHex

Variable tipo cadena de caracteres que contiene la cadena de caracteres de valores hexagesimales que serán convertidos a datos de tipo binario y enviados por el puerto serial.

Rutina HexToDec

Realiza la conversión de un dato tipo Hexagesimal a su equivalente en números enteros. El formato de esta rutina es el siguiente:

HexToDec(hexa)

Valores de entrada : hexa

Variable tipo cadena de caracteres que contiene el valor Hexagesimal que se desea convertir.

Valores de salida : ValorEntero

Variable tipo entero que contiene el resultado de la conversión.

Módulo CodificaDecodifica

Dentro de este módulo se encuentran las rutinas encargadas de la codificación y decodificación del mensaje de texto.

Rutina Decodifica_Mensaje_de_Texto

Realiza la decodificación de un mensaje de texto recibido. El formato de esta rutina es el siguiente:

Decodifica_Mensaje_de_Texto(CadenaHexdeEntrada)

Valores de entrada : CadenaHexdeEntrada

Variable tipo cadena de caracteres que contiene los bytes del mensaje de texto en formato Hexagesimal que deben ser decodificados.

Valores de salida : Cantidaddecaracteres7bits

Variable de tipo entero que contiene la cantidad de caracteres que resultaron luego del proceso de decodificación del mensaje de texto.

MensajedeTextodeCodificadoP1

Variable de tipo cadena de caracteres que contiene el contenido del mensaje de texto decodificado.

MensajedeTextodeCodificadoP2

Variable de tipo cadena de caracteres utilizada en caso de que el mensaje de texto sea enviado por el teléfono en 2 tramas. Contiene el contenido del mensaje de texto decodificado de la segunda trama recibida.

Rutina Codifica_Mensaje_de_Texto

Realiza la codificación del mensaje a enviar. El formato de esta rutina es el siguiente:

Codifica_Mensaje_de_Texto(MensajedeEntrada)

Valores de entrada : MensajedeEntrada

Variable tipo cadena de caracteres que contiene el mensaje que debe ser codificado.

Valores de salida : CantidaddeHexas

Variable de tipo entero que contiene la cantidad de bytes que resultaron luego del proceso de codificación.

MensajedeTextodeCodificadoP1

Variable tipo cadena de caracteres que contiene los bytes en formato Hexagesimal del mensaje codificado.

Módulo ChecksumFBUS

Dentro de este módulo se encuentran las rutinas encargadas del cálculo de los checksums de las tramas enviadas y recibidas.

Rutina CalculaChecksumFBUS

Realiza el cálculo y genera los checksums 1 y 2 para las tramas que van a ser enviadas. El formato de esta rutina es el siguiente:

CalculaChecksumFBUS(ValoresDeLaCadena)

Valores de entrada : ValoresDeLaCadena

Variable tipo cadena de caracteres que contiene los bytes en formato Hexagesimal de la trama sin checksums que va a ser enviada .

Valores de salida : FlagPaddingByte

Variable tipo entero que cuando su valor es 1, indica que la trama necesita además e los checksums el PaddingByte.

ChkSum1

Variable tipo cadena de caracteres que contiene el valor del checksum1 en formato Hexagesimal.

ChkSum2

Variable tipo cadena de caracteres que contiene el valor del checksum2 en formato Hexagesimal.

Rutina CalculaChecksumFBUS_Recibida

Realiza el cálculo y los checksums 1 y 2 de una trama recibida y comprueba que los valores sean correctos. El formato de esta rutina es el siguiente:

CalculaChecksumFBUS_Recibida(ValoresDeLaCadena)

Valores de entrada : ValoresDeLaCadena

Variable tipo cadena de caracteres que contiene la trama recibida en formato Hexagesimal.

Valores de salida : CHKok

Variable tipo entero que indica mediante el valor de 1 que los checksums 1 y 2 de la trama recibida son correctos.

Módulo ProtocoloFBUS

Dentro de este módulo se encuentran las rutinas encargadas del envío, recepción, generación e interpretación de tramas FBUS.

Rutina Sincroniza

Realiza la sincronización de los UARTs de la computadora y el teléfono celular. El formato de esta rutina es el siguiente:

Sincroniza

Rutina Detecta_Trama_Recibida_Válida

Detecta e indica la recepción de una trama FBUS completa y válida. El formato de esta rutina es el siguiente:

Detecta_Trama_Recibida_Válida(Trama_de_entrada)

Valores de entrada : Trama_de_entrada

Variable tipo cadena de caracteres que contiene los datos recibidos por el puerto serial en formato hexagesimal.

Valores de salida : FlagTramaFBUSRecibida

Variable de tipo entero que indica mediante el valor de 1 que la información recibida es una trama FBUS, se encuentra completa y es válida.

Rutina LeerTrama

Realiza la separación de los diversos campos que conforman la trama FBUS recibida. El formato de esta rutina es el siguiente:

LeerTrama(TramaRecibida)

Valores de entrada : TramaRecibida

Variable tipo cadena de caracteres que contiene la trama recibida en formato Hexagesimal.

Valores de salida : Cabecera

Variable tipo cadena de caracteres que contiene el contenido de la cabecera de la trama FBUS leída.

Comando

Variable tipo cadena de caracteres que contiene código del Tipo de mensaje de la trama FBUS recibida.

Bloque

Variable tipo cadena de caracteres que contiene el bloque de datos de la trama FBUS recibida.

Rutina Interpreta_Comando

Realiza la separación de los datos útiles contenidos dentro del bloque de datos de la trama FBUS recibida. El formato de esta rutina es el siguiente:

Interpreta_Comando(Comando_de_Entrada)

Valores de entrada : Comando_de_Entrada

Variable tipo cadena de caracteres que contiene el código del Tipo de mensaje de la trama recibida en formato Hexagesimal.

Valores de salida : Subcomando

Variable tipo cadena de caracteres que contiene la función de la trama FBUS recibida en formato hexagesimal. (ver tabla 3.2)

Rutina Genera_Secuencia

Genera el valor de la secuencia de una trama que va a ser enviada al teléfono celular. El formato de esta rutina es el siguiente:

Genera_Secuencia

Valores de salida : SeqNo

Variable tipo cadena de caracteres que contiene el valor de la secuencia de la trama FBUS que va a ser enviada en formato hexagesimal.

Rutina Solicitar_Entrada_de_Directorio_SIM

Realiza la solicitud de una entrada del directorio telefónico al teléfono celular. El formato de esta rutina es el siguiente:

Solicitar_Entrada_de_Directorio_SIM(Entrada)

Valores de entrada : Entrada

Variable tipo entero que contiene el número de la entrada del directorio telefónico que se desea leer.

Valores de salida : PBNombre

Variable tipo cadena de caracteres que contiene el nombre grabado en la entrada del directorio telefónico leído.

PBNúmero

Variable tipo cadena de caracteres que contiene el número grabado en la entrada del directorio telefónico leído

Rutina Solicitar_Estado_de_Directorio

Solicita al teléfono celular el estado de las entradas del directorio telefónico. El formato de esta rutina es el siguiente:

Solicitar_Estado_de_Directorio(Tipo_de_Memoria)

Valores de entrada : Tipo_de_Memoria

Variable tipo cadena de caracteres que contiene el código del tipo de memoria que se desea leer. Los valores aceptados para esta variable son:

01h: Teléfono y Tarjeta SIM juntas.

02h: En el teléfono.

03h: En la tarjeta SIM **(UTILIZADA POR DEFECTO)**.

05h: Número del teléfono utilizado.

07h: Lista de llamadas realizadas.

08h: Lista de llamadas perdidas.

09h: Lista de llamadas recibidas.

0Bh: mensajes de voz.

Valores de salida : MemoriasUsadas

Variable de tipo entero que contiene el número de posiciones utilizadas del directorio telefónico del teléfono celular.

MemoriasLibres

Variable de tipo entero que contiene el número de posiciones libres para grabar entradas en el directorio telefónico del teléfono celular.

Rutina Solicita_Número_del_SMSCENTER

Solicita al teléfono celular el número de la central de mensajes de texto para el envío de mensajes. Es necesario utilizar esta rutina antes de enviar un mensaje de texto. El formato de esta rutina es el siguiente:

Solicita_Número_del_SMSCENTER

Rutina Solicitar_Mensaje_de_Texto

Realiza la solicitud de los mensajes de texto al teléfono celular. El formato de esta rutina es el siguiente:

Solicitar_Mensaje_de_Texto(Número_de_mensaje)

Valores de entrada : Número_de_mensaje

Variable tipo entero que contiene el número de la posición en la que se encuentra grabado el mensaje de texto que se desea leer. Este valor debe estar comprendido entre 1 y 25.

Valores de salida : NúmeroqueEnvia

Variable tipo cadena de caracteres que contiene el número telefónico desde donde fue enviado el mensaje de texto.

Fecha de Mensaje

Variable tipo cadena de caracteres que contiene la fecha en la que se recibió el mensaje de texto.

Hora de Mensaje

Variable tipo cadena de caracteres que contiene la hora en la que se recibió el mensaje de texto.

SMS(25)

Arreglo de variables tipo entero que contiene el estado del mensaje de cada posición de memoria. Los posibles valores que puede tomar cada posición son:

01h = Leído

02h = Memoria no Válida.

03h = No Leído

07h = Posición Vacía.

SMSmensaje(25)

Arreglo de variables tipo cadena de caracteres que contiene el mensaje de texto de cada posición de memoria.

Númerodesms

Variable tipo entero que contiene el número de mensaje de texto que ha sido leído.

Rutina Envia_Mensaje_de_Texto

Solicita al teléfono celular el envío de un mensaje e texto. Antes de utilizar esta rutina es necesario haber introducido el número al que se va a enviar el mensaje de texto y el mensaje que se desea enviar en las variables NúmeroaqueEnviar y MensajeaEnviar respectivamente.

Envia_Mensaje_de_Texto

Valores de entrada : MensajeaEnviar

Variable tipo cadena de caracteres que contiene el mensaje de texto que se quiere enviar.

NúmeroaqueEnviar

Variable tipo cadena de caracteres que contiene el número del teléfono celular al que se quiere enviar el mensaje de texto.

Ejemplo de uso:

Solicita_Número_del_SMSCENTER

MensajeaEnviar="Esta es una prueba"

NúmeroaqueEnviar="97234766"

Envia_Mensaje_de_Texto

La rutina "Solicita_Número_del_SMSCENTER" debe haber sido ejecutada por lo menos una vez desde que se inició el programa antes del envío del mensaje de texto para que la acción se pueda realizar.

Rutina Solicita_Cantidad_de_Mensajes_no_Leídos

Solicita la cantidad de mensajes de texto grabados en el teléfono celular y cuantos de ellos no han sido leídos. El formato de esta rutina es el siguiente:

Solicita_Cantidad_de_Mensajes_no_Leídos

Valores de salida : CantidaddeMensajes

Variable de tipo entero que contiene la cantidad de mensajes de texto grabados en el teléfono celular.

CantidaddeMensajesNoLeídos

Variable de tipo entero que contiene la cantidad de mensajes de texto no leídos grabados en el teléfono celular.