



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA**

FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE UN
TERMOPLUVIÓGRAFO DIGITAL**

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

ELÉCTRICA - SISTEMAS ELECTRÓNICOS

P R E S E N T A :

JORGE ACOSTA HUERTA



DIRECTOR DE TESIS:
M. I. LAURO SANTIAGO CRUZ

2007

JURADO ASIGNADO:

Presidente: Dr. Vicente Vivas Esau

Secretario: M.I. Haro Ruíz Luis Arturo

Vocal: M.I. Santiago Cruz Lauro

1^{er.} Suplente: M.I. Quintana Thierry Sergio

2^{do.} Suplente: M.I. Tovar Medina Roberto

Lugar o lugares donde se realizó la tesis:
INSTITUTO DE INGENIERÍA

TUTOR DE TESIS:

M.I. Lauro Santiago Cruz

FIRMA

Dedicatoria

A mi esposa Maru y a mis hijos Ericksen Juan y Nelly Daniela, quienes siempre han sido el motivo de mi vida, de mi amor, de mis alegrías y de todos los propósitos a los que he anhelado.

A mi padre Don Fide y a mi madre Natalia, por la oportunidad de vida que me otorgaron.

A Chuchita, por el amor y cuidados que me proporcionó durante el tiempo que dios le permitió.

A mis hermanos y en particular a Jaime, por todo el tiempo que pasamos juntos en las buenas y en las malas, así como en las alegrías y en los sinsabores.

A mis suegros Don Alfre y Doña Male, por su sincera comprensión, cariño y amistad; pero sobretodo, por haberme proporcionado la motivación principal, que es Maru.

A mis cuñados, por el cariño y la confianza que siempre me han demostrado.

A mis amigos, por el tiempo que hemos disfrutado juntos.

AGRADECIMIENTOS

A dios, por sobre todas las cosas.

A Maru, Juanito y Danielita, por su amor.

A mis padres, por la vida.

A toda la familia, por el apoyo mostrado.

Al Ing. Agustín Pérez, por el ánimo proporcionado al inicio de la carrera.

Al M.I. Bernardo Frontana de la Cruz, por el apoyo, motivación y amistad en la Facultad de Ingeniería.

Al M.I. Lauro Santiago Cruz, por el apoyo y comprensión durante el presente trabajo.

A mis sinodales:

Dr. Vicente Vivas Esau

M.I. Haro Ruíz Luis Arturo

M.I. Santiago Cruz Lauro

M.I. Quintana Thierry Sergio

M.I. Tovar Medina Roberto

A todos los profesores de la Facultad de Ingeniería que participaron en mi formación personal y académica.

A la Facultad de Ingeniería, por proporcionarme los medios educativos.

Al Instituto de Ingeniería, por el apoyo a lo largo de este proyecto.

A la Universidad Nacional Autónoma de México, por ser nuestra máxima casa de estudios.

ÍNDICE

ÍNDICE	i
ÍNDICE DE FIGURAS	iii
ÍNDICE DE TABLAS	vii
PRÓLOGO	I
INTRODUCCIÓN	III
CAPÍTULO I. ANTECEDENTES	1
I.1. Estaciones de mediciones meteorológicas	1
I.2. Planteamiento del problema	5
I.3. Propuesta de solución	6
I.4. Metodología de diseño y desarrollo	8
CAPÍTULO II. FUNDAMENTOS TEÓRICOS	11
II.1. Pluviometría	11
II.1.1. Componentes del termopluviógrafo	12
II.1.1.1. Pluviómetro tipo balancín	12
II.1.1.2. Sistema de registro de eventos	14
II.2. Microcontroladores	15
II.2.1. Componentes de los microcontroladores	17
II.2.2. Arquitectura de los microcontroladores	22
II.2.3. Software en los microcontroladores	23
II.3. Comunicaciones alámbricas	23
II.3.1. Estándar RS-232	24
II.3.1.1. Especificaciones del estándar	25
II.3.2. Estándar USB	38
II.3.2.1. Especificaciones del estándar	42
II.4. Comunicaciones inalámbricas	49
II.4.1. Estándar IRDA	50
II.4.1.1. Tecnología de IrDA	51
II.4.1.2. Especificaciones del estándar	52
II.4.2. Estándar <i>Bluetooth</i>	61

II.4.2.1. Tecnología <i>Bluetooth</i>	63
II.4.2.2. Especificaciones de <i>Bluetooth</i>	64
II.4.2.3. Perfiles de <i>Bluetooth</i>	78
II.4.2.4. Seguridad en <i>Bluetooth</i>	81
II.5. Lenguaje de alto nivel	83
CAPÍTULO III. DISEÑO Y DESARROLLO DEL SISTEMA	85
III.1. Descripción del sistema	85
III.2. Diseño y desarrollo de los módulos	86
III.3. Módulo de comunicación alámbrica	99
III.4. Módulo de comunicación inalámbrica	104
III.5. Software del sistema	110
CAPÍTULO IV. APLICACIÓN DE PRUEBAS	117
IV.1. Evaluación	117
CAPÍTULO V. RESULTADOS Y CONCLUSIONES	127
V.1. Resultados	127
V.2. Conclusiones	129
BIBLIOGRAFÍA	131
APÉNDICES	
APÉNDICE A. MODELO OSI	A-1
APÉNDICE B. MANUAL DE USUARIO	B-1
APÉNDICE C. SOFTWARE	C-1
APÉNDICE D. GLOSARIO DE TÉRMINOS	D-1

ÍNDICE DE FIGURAS

CAPÍTULO I

Figura I.1. Observatorios ubicados en la Republica Mexicana	4
Figura I.2. Sistema de medición y alerta pluviométrica e hidrométrica	5
Figura I.3. Diagrama esquemático del termopluviógrafo	6

CAPÍTULO II

Figura II.1. Pluviómetro tipo balancín	13
Figura II.2. Elementos del sistema de registro de eventos	14
Figura II.3. Estructura básica de un microcontrolador	16
Figura II.4. Oscilador de cristal	21
Figura II.5. Arquitectura Harvard	22
Figura II.6. Tipos de conectores	26
Figura II.7. Conexión de la UART hacia el exterior	32
Figura II.8. Protocolo de transmisión serie	33
Figura II.9. Codificación NRZ-L y niveles de voltaje EIA-232	35
Figura II.10. Conexión física EIA-232	36
Figura II.11. Conexiones EIA-232 sin control de flujo de datos	37
Figura II.12. Conexiones EIA-232 con control de flujo de datos	37
Figura II.13. Conexiones EIA-232 de DTE a DCE	38
Figura II.14. Tipos de conectores de bus USB	42
Figura II.15. Estructura de capas del bus USB	45
Figura II.16. Concentrador o hub de 7 puertos	46
Figura II.17. Periféricos conectados por USB	47
Figura II.18. Código NRZ-I	48
Figura II.19. Estándar de comunicación por infrarrojos	51
Figura II.20. Arquitectura de Protocolos de IrDA	52
Figura II.21. Modulación para SIR	53
Figura II.22. Modulación para MIR	53
Figura II.23. Modulación 4PPM para FIR	54
Figura II.24. Formato de un paquete de datos en IrDA	57
Figura II.25. Símbolo y logotipo de <i>Bluetooth</i>	63
Figura II.26. <i>Scatternet</i>	63
Figura II.27. Arquitectura de protocolos de <i>Bluetooth</i>	65
Figura II.28. Técnica de salto de frecuencias FHSS	67
Figura II.29. Modulación GFSK	68
Figura II.30. Tipos de conexión	69
Figura II.31. Formato de paquete de datos en <i>Bluetooth</i>	70

Figura II.32. Estructura de los perfiles de <i>Bluetooth</i>	79
CAPÍTULO III	
Figura III.1. Diagrama en bloques del sistema de medición	87
Figura III.2. Pluviómetro tipo balancín	87
Figura III.3. Acondicionamiento de la señal del pluviómetro	88
Figura III.4. Acondicionamiento de señal del sensor de temperatura	89
Figura III.5. Diagrama a bloques del microcontrolador	91
Figura III.6. Distribución de terminales del microcontrolador	93
Figura III.7. Sistema de desarrollo AVR	94
Figura III.8. Interconexión de las memorias con el microcontrolador	95
Figura III.9. Esquema de memoria EEPROM	96
Figura III.10. Visualizador alfanumérico de 2 líneas X 16 caracteres	96
Figura III.11. Tecla de operación tipo <i>push-button</i>	97
Figura III.12. Configuración del circuito elevador de voltaje	98
Figura III.13. Configuración del regulador lineal	99
Figura III.14. Comunicación alámbrica	100
Figura III.15. Transceptor	100
Figura III.16. Implementación de la interfaz EIA-232	101
Figura III.17. Arquitectura del circuito convertidor	102
Figura III.18. Implementación del estándar USB a EIA-232	103
Figura III.19. Implementación de la interfaz USB	104
Figura III.20. Comunicación inalámbrica	104
Figura III.21. Transceptor óptico	105
Figura III.22. Diagrama funcional del transceptor óptico	106
Figura III.23. Implementación del estándar USB a IrDA	107
Figura III.24. Implementación de la interfaz IrDA	107
Figura III.25. Diagrama funcional de bloques del módulo <i>Bluetooth</i>	108
Figura III.26. Implementación de la interfaz <i>Bluetooth</i>	110
Figura III.27. Diagrama general del sistema de registro	111
Figura III.28. Configuración del visualizador	113
Figura III.29. Diagrama general del código de interfaz gráfica	115
CAPÍTULO IV	
Figura IV.1. Pruebas a dispositivo electrónico en una <i>proto-board</i>	118
Figura IV.2. Implementación de circuitos en tarjeta perforada	118
Figura IV.3. Circuito integrado de montaje de superficie	119
Figura IV.4. Estándar EIA-232	120
Figura IV.5. Evaluación del estándar de comunicación USB	121
Figura IV.6. Transmisión de datos por IrDA	121
Figura IV.7. Tamaño del módulo <i>Bluetooth</i>	122
Figura IV.8. Pluviómetro tipo balancín	123
Figura IV.9. Pruebas de operación al termopluviógrafo digital	123
Figura IV.10. Presentación errónea de datos en forma de texto	124

Figura IV.11. Presentación de datos verídicos en formato de texto	125
Figura IV.12. Prueba de transmisión y recepción de datos por USB	126

APÉNDICE B

Figura B.1. Asistente de empaquetado y distribución	B-2
Figura B.2. Entrada al programa a través de un acceso directo	B-2
Figura B.3. Entrada al programa a través del menú de programas	B-3
Figura B.4. Conexión a la PC	B-3
Figura B.5. Formulario principal del programa	B-4
Figura B.6. Error de comunicación serie	B-4
Figura B.7. Selección de estándar de comunicación	B-5
Figura B.8. Configuración del puerto de comunicación serie	B-5
Figura B.9. Error en la elección del puerto de comunicación serie	B-6
Figura B.10. Ventana de comunicación establecida	B-6
Figura B.11. Ventana de parámetros de operación	B-7
Figura B.12. Ventana de modificación de parámetros	B-7
Figura B.13. Opción de permanencia de datos de eventos	B-8
Figura B.14. Adquisición de datos	B-9
Figura B.15. Presentación de datos en diferentes formatos	B-9
Figura B.16. Presentación de datos en formato de hoja de cálculo	B-10
Figura B.17. Presentación de datos en formato de archivo de texto	B-11
Figura B.18. Intervalo de días por graficar	B-12
Figura B.19. Opción de gráfica	B-12
Figura B.20. Presentación de datos en un hietograma	B-13
Figura B.21. Presentación de datos en una curva de masas	B-13
Figura B.22. Hietograma de un día	B-14
Figura B.23. Curva de masas de un día	B-14
Figura B.24. Teclas de operación del termopluviógrafo digital	B-15
Figura B.25. Letrero que identifica al sistema de registro	B-16
Figura B.26. Letreros del sistema de registro	B-17
Figura B.27. Modificación del letrero capacidad	B-17
Figura B.28. Letrero de capacidad de balancín modificado	B-18

ÍNDICE DE TABLAS

Tabla I.1. Metodología de diseño y desarrollo	8
Tabla II.1. Especificación de terminales en conectores EIA-232	27
Tabla II.2. Especificación de terminales en conector USB	43
Tabla II.3. Distancia de transmisión para cables	43
Tabla II.4. Características de estándares IrDA, WLAN y <i>Bluetooth</i>	50
Tabla III.1. Características del sensor de temperatura	89
Tabla III.2. Características del microcontrolador	90
Tabla III.3. Funciones alternas del puerto B	92
Tabla III.4. Funciones alternas del puerto D	92
Tabla III.5. Interconexión de terminales	93
Tabla III.6. Características de la memoria EEPROM	95
Tabla III.7. Características del visualizador	96
Tabla III.8. Funciones de las teclas de operación	97
Tabla III.9. Características del circuito elevador de voltaje	98
Tabla III.10. Características de circuito proveedor de energía alterno	99
Tabla III.11. Características del convertidor	101
Tabla III.12. Características del transceptor óptico	106
Tabla III.13. Características principales del módulo <i>Bluetooth</i>	109
Tabla V.1. Especificaciones del termoplúviógrafo digital	128
Tabla B.1. Función de las teclas de operación	B-16

PRÓLOGO

La comunicación de información en la vida diaria se vuelve cada vez más indispensable y exigente, la transferencia de información de un dispositivo a otro y viceversa involucra todos los aspectos de seguridad y confiabilidad. La distancia está de por medio en una transmisión de datos, porque cuanto más separados se encuentren los dispositivos habrá mayor dificultad para el envío y recepción de información en forma eficiente.

Eventos climatológicos como: huracanes, terremotos, inundaciones, etc., pueden ser estudiados a través de equipos de medición. Estos dispositivos detectan señales del parámetro de interés por medio de sensores, posteriormente se hace la conversión de las señales adquiridas a otras que puedan ser procesadas en un equipo de cómputo.

Los dispositivos que se colocan en lugares apartados con el fin de adquirir información de un parámetro, tal como: temperatura, sonido, velocidad de viento, cantidad de lluvia (precipitación pluvial), presión, movimiento de tierra, luz, etc., tienen la necesidad de mandar la información recabada hacia otros equipos que harán el procesamiento de datos, que a su vez, serán usados para observar el comportamiento de un fenómeno natural.

Desafortunadamente, los fenómenos naturales se presentan cuando menos se les espera, y es aquí, donde las dependencias encargadas de elaborar estudios correspondientes a estos, deben hacer predicciones y tomar decisiones para prevenir desastres y tragedias que dañen a la sociedad en general.

Los desastres generados por la naturaleza, que ocurren en México y en todo el mundo, se podrían prever para evitar daños mayores en las vidas y bienes de las personas. Para realizar esta prevención es necesario desarrollar equipo electrónico que nos permita estar informados respecto al comportamiento climatológico; para ello hay empresas que venden o desarrollan dicho equipo. En este último aspecto, la compañía IROSA solicitó al Instituto de Ingeniería de la UNAM (Universidad Nacional Autónoma de México) diseñar y desarrollar (o implementar mejoras a) equipos de medición para colaborar en la prevención de eventos climatológicos.

El objetivo del presente trabajo de tesis de maestría es desarrollar un termopluviógrafo digital. El termopluviógrafo digital es un equipo de medición que adquiere datos que representan precipitación pluvial (lluvia) en regiones específicas de estudio, proporcionando además la temperatura.

El termopluviógrafo digital tendrá la capacidad de almacenar los datos adquiridos en una memoria no volátil, los cuales podrán ser enviados a una computadora personal o a un equipo portátil (*Lap Top* y PDA), por medio de un protocolo de comunicación.

Los lugares en donde se instalan estos dispositivos pueden ser desde partes geográficas accesibles hasta regiones de difícil acceso; por lo tanto se tiene la necesidad de tener comunicación alámbrica e inalámbrica con tales equipos, para extraer la información que ha recabado en un periodo de tiempo y así poderla procesar.

Al termopluviógrafo se le agregarán tecnologías modernas y avanzadas para comunicación de datos. El estándar de comunicación podrá ser elegido entre un sistema alámbrico (RS-232 y USB) o inalámbrico (IrDA y *Bluetooth*).

El diseño del termopluviógrafo involucra *hardware* y *software* para lograr una operación óptima del sistema. Igualmente se pretende un sistema que sea de bajo costo, soporte técnico accesible y, sobre todo, contribuir al desarrollo de la tecnología nacional a través de la UNAM.

Jorge Acosta Huerta

INTRODUCCIÓN

La Universidad Nacional Autónoma de México, a través de sus institutos, facultades y demás entidades; se ha mantenido a la vanguardia en el acontecer político, social, económico, tecnológico, científico, etc.

Los desastres climatológicos en territorio nacional no han pasado desapercibidos para la institución, por lo que el Instituto de Ingeniería, a través del personal docente y sus estudiantes, ha desarrollado proyectos de investigación, diseño y elaboración de equipos de medición; esto es con el fin de contar con herramientas que proporcionen información relacionada con los fenómenos naturales, como es el de este proyecto que tratará el tema de la precipitación pluvial.

La presente tesis comprende diseñar y desarrollar un termopluviógrafo digital, capaz de efectuar mediciones de precipitaciones pluviales y temperatura ambiente. En el trabajo escrito se describirá el *hardware* y *software* necesario para la captura de datos de temperatura y lluvia, además de la comunicación y procesamiento de datos en equipos de cómputo, sean estos fijos o portátiles.

El termopluviógrafo digital tendrá la función principal de adquirir y almacenar la información de temperatura y lluvia colectada, el registro se hará en memoria no volátil. Tales datos podrán ser enviados a través de algún protocolo de comunicación. Una vez que se tenga el equipo de medición, incluyendo el protocolo de comunicación RS-232, se ampliarán los protocolos de comunicación, de tal manera que se abarcarán los tipos de tecnologías alámbricas e inalámbricas más modernas y más utilizadas. Estas tecnologías constituirán los siguientes medios de comunicación:

Comunicación alámbrica: estándares RS-232 y USB.

Comunicación inalámbrica: estándares IrDA y *Bluetooth*.

Los estándares anteriores comunicarán el termoplviógrafo digital a un equipo de cómputo mediante un protocolo seleccionado.

El termoplviógrafo digital estará constituido por las siguientes partes:

- Un pluviómetro, el cual se encargará de coleccionar el agua de lluvia.
- Un sistema de adquisición de datos, el cual se hará cargo de la recepción y procesamiento de datos que representan la cantidad de lluvia, así como de la temperatura del medio ambiente en que se colocará este equipo.
- Un programa de *software*, que será el enlace entre un equipo de cómputo y el sistema de adquisición de datos. Este *software* proporcionará la forma de obtener la información y presentarla para su posterior análisis.

Los parámetros principales que se tomarán en consideración para el diseño y desarrollo de este proyecto son los siguientes: tamaño, acceso, autonomía, costo, operación y manejo de información eficiente.

En cuanto al trabajo escrito de la tesis, éste estará estructurado de la siguiente manera:

Capítulo I. Aquí se realiza una observación de algunas estaciones meteorológicas en el país, así como la instrumentación utilizada en éstas; se hace una introducción a los equipos de medición, pero principalmente aquellos que se relacionan con la problemática de la precipitación pluvial en México. Se hace el planteamiento del problema, así como la propuesta de solución mediante la realización de un termoplviógrafo digital; también describe la metodología empleada en el desarrollo del *hardware* y *software* para el funcionamiento del dispositivo.

Capítulo II. En este capítulo se proporcionan los conceptos generales relacionados con la precipitación pluvial, es decir, de la pluviometría. De la misma manera se tratan los microcontroladores, así como el lenguaje ensamblador, el cual es utilizado para la programación de estos. Se hace la descripción de los estándares de comunicación empleados para las diferentes formas de transmisión de datos. Finalmente se presenta el lenguaje de programación *Visual Basic*.

Capítulo III. En esta parte del trabajo se establecerá: el análisis, el diseño y el desarrollo de cada módulo contenido en el termopluviógrafo digital. Se implementará el pluviómetro, el sistema de adquisición de datos, y la comunicación alámbrica e inalámbrica. De igual manera se presentarán cada uno de los componentes electrónicos que estarán involucrados en el proyecto, así como los esquemas y diagramas más importantes del termopluviógrafo. Por último se hará la aplicación de la programación del *software* para el microcontrolador y para el sistema que hará interfaz de comunicación entre el adquirente de datos y un equipo de cómputo.

Capítulo IV. Aquí se presentarán las pruebas de *hardware* y *software* a las que serán sometidos los diferentes módulos que integran el termopluviógrafo digital.

Capítulo V. Este capítulo proporcionará los resultados a los que se llegaron, con el fin de tener un panorama del funcionamiento del proyecto, posteriormente se proporcionarán las conclusiones como una consecuencia de los resultados.

Bibliografía. Este apartado es el que corresponde a las listas de los diferentes libros y manuales, así como páginas de *internet* consultadas para la elaboración del presente trabajo.

Apéndices. En esta parte se finaliza el escrito proporcionando material de referencia. En el apéndice A se presentará de manera resumida el modelo OSI, que se considera como arquitectura modelo de protocolos para los estándares IrDA y *Bluetooth*. Se continúa con el apéndice B, el cual presentará un manual de usuario, mostrando la operación del programa desarrollado de la interfaz de comunicación usuario-equipos de cómputo-termopluviógrafo, diseñado y elaborado bajo *Visual Basic*. Posteriormente, el apéndice C contendrá parcialmente el *software* implementado para controlar la operación del termopluviógrafo, desarrollado en lenguaje ensamblador, así como el código de la interfaz gráfica desarrollado en lenguaje *Visual Basic*. Por último se presentará el apéndice D, que estará conformado por un glosario de términos utilizados en este trabajo.

CAPÍTULO I

ANTECEDENTES

En este capítulo se mencionan algunos sistemas de medición que operan en las estaciones de registro en México, enfocando principalmente el relacionado con la precipitación pluvial. También se presenta un panorama en la problemática de la precipitación pluvial y desencadena en el planteamiento del problema que ocupará este trabajo. Se hace la propuesta de la realización de un termopluviógrafo digital. Por último, se describe la metodología que se empleará para el diseño del sistema de comunicaciones.

I.1. Estaciones de mediciones meteorológicas

Una estación meteorológica es una instalación destinada a medir y registrar diversas variables meteorológicas. Estos datos se utilizan principalmente para la elaboración de predicciones meteorológicas. Los instrumentos comúnmente utilizados en estas estaciones, así como la variable de medición correspondiente, se presentan a continuación:

- Termómetro, que mide la temperatura.
- Barómetros, que proporcionan medidas de presión atmosférica.
- Pluviómetro, que realiza mediciones de la cantidad de precipitación pluvial.
- Psicrómetro, realiza medidas de la humedad relativa del aire y la temperatura del punto de rocío.

- Piranómetro, presenta mediciones de la insolación solar.
- Heliógrafo, proporciona medidas de las horas de sol.
- Anemómetro, mide la velocidad del viento y veleta para registrar su dirección.
- Ceilómetro, muestra mediciones de la altura de las nubes en el punto donde se encuentre instalado.

México cuenta con varios tipos de redes de estaciones remotas de medición o Estaciones Meteorológicas Automatizadas (EMA), algunas de las cuales enunciaremos a continuación.

Red de Monitoreo de Partículas PM_{2.5}

Esta red vigila y monitorea partículas suspendidas menores de 2.5 micrómetros (PM_{2.5}), que son contaminantes del aire y están constituidos por material sólido o líquido (ozono, monóxido de carbono, dióxido de azufre, dióxido de nitrógeno y plomo). Algunas PM_{2.5} se emiten directamente a la atmósfera, como son las provenientes de la combustión de vehículos diesel y de gasolina. Las PM_{2.5} también se forman a partir de reacciones químicas de gases emitidos a la atmósfera formando aerosoles de nitratos y sulfatos y de compuestos orgánicos. En menor proporción contribuyen las fuentes geológicas (suelo) y biológicas (polen).

La red de PM_{2.5} está integrada por 8 estaciones remotas automáticas y 7 manuales, ubicadas en zonas densamente pobladas de la ciudad, éstas son:

- *Estaciones Remotas Automáticas:* Coyoacán, UAM-Iztapalapa, La Merced, San Juan de Aragón, Nezahualcoyotl, San Agustín, Camarones, Tlalnepantla.
- *Estaciones Remotas Manuales:* Coyoacán, La Merced, Pedregal, San Agustín, Tlanepantla, UAM-Iztapalapa y Xalostoc.

Servicio de Meteorología Aeronáutica

Conformado por cincuenta y seis estaciones de observación que están instaladas a lo largo de la República Mexicana. Ahí se realizan observaciones de las condiciones climatológicas que imperan en cada aeropuerto. La información generada se reúne en el Centro de Análisis y Pronósticos Meteorológicos (CAPMA) que opera en el Aeropuerto Internacional de la Ciudad de México (AICM) y es utilizada para la elaboración y edición de los pronósticos meteorológicos de área, ruta y terminal.

Red de monitoreo del Popocatépetl

Esta red realiza el monitoreo de la actividad volcánica del Popocatépetl bajo tres esquemas principales:

- *La vigilancia sísmica:* se requiere de un sismógrafo, quizás es la más importante, porque permite conocer la actividad a través de la microactividad sísmica.
- *El monitoreo geodésico:* se colocan inclinómetros para estudiar la deformación del terreno.
- *El monitoreo geoquímico:* se realiza a través de un muestreo de gases de fumarolas y aguas termales, temperatura y análisis químicos de manantiales y de los gases del suelo.

Esta red de monitoreo consta de 15 estaciones remotas y una estación central de proceso y adquisición de datos ubicada en el Centro Nacional de Prevención de Desastres (CENAPRED).

Servicio Meteorológico Nacional

Es el organismo que se encarga de proporcionar información sobre el estado del tiempo en una escala local y nacional en nuestro país, depende de la Comisión Nacional del Agua (CNA), la cual forma parte de la Secretaría del Medio Ambiente y Recursos Naturales (SEMARNAT). La función principal de este servicio es vigilar y emitir información acerca de las condiciones atmosféricas del país, así como pronosticar y alertar sobre eventos hidrometeorológicos que puedan ocasionar daños a la población o a las actividades productivas en el territorio nacional.

El presente trabajo se centrará en la problemática de eventos hidrometeorológicos y más estrictamente en lo relacionado con las precipitaciones pluviales. Por lo anterior, se harán las observaciones a las actividades que realiza el Servicio Meteorológico Nacional (SMN).

El Servicio Meteorológico Nacional está conformado por una red de 80 observatorios, algunos de los cuales se muestran en la figura I.1. Estos centros de observación son los encargados de generar la información meteorológica y climatológica del país; para dar servicio tanto a instituciones nacionales privadas y de gobierno, así como a organismos internacionales, tal como la Organización Mundial de Meteorología (OMM). Esta información es usada para el análisis y elaboración de pronósticos de tiempo, aplicadas en áreas agrometeorológicas hidrológicas, climatológicas, etc.

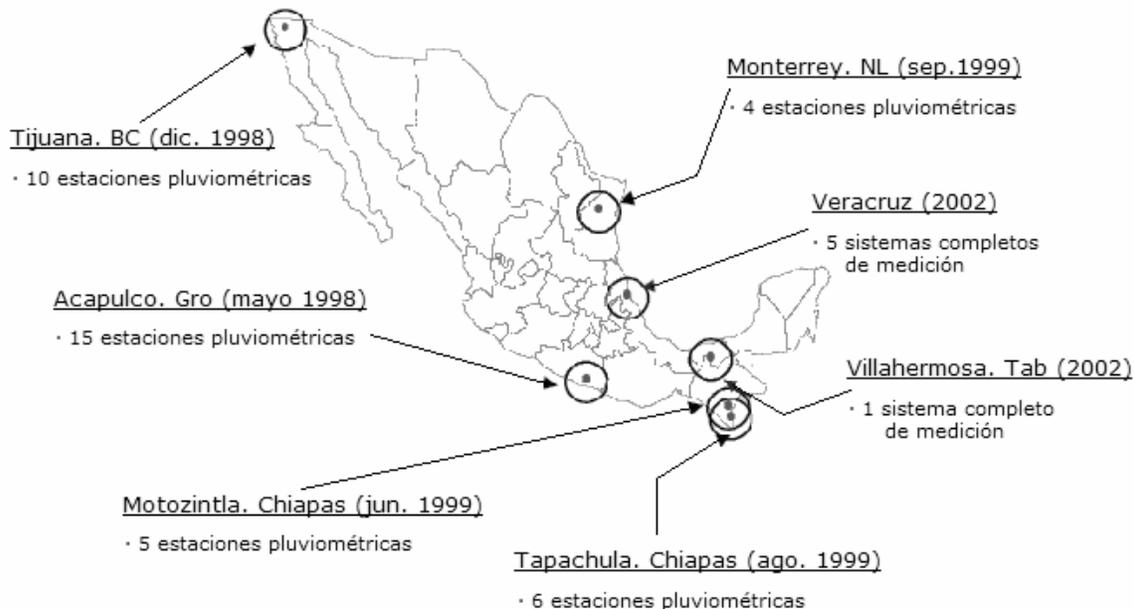


Figura I.1. Observatorios ubicados en la Republica Mexicana.

También se observan estados que cuentan con más o menos estaciones de medición pluviométrica, así como estados que tienen cantidades diferentes de sistemas completos de medición (la velocidad del viento, la dirección del viento, la presión, la altura de las nubes, la temperatura, la humedad, la radiación solar y la precipitación pluvial).

Todos los observatorios meteorológicos de la red deben trabajar las 24 horas del día ininterrumpidamente durante los 365 días del año.

Sistemas de medición remota y de alerta. Estos sistemas de medición están basados en un conjunto de estaciones pluviométricas e hidrométricas, ubicadas en diferentes regiones hidrológicas de estudio en que se divide el país; miden principalmente la precipitación acumulada, la intensidad de lluvia y los niveles de los cauces en los ríos, y envían la información adquirida por telemetría a una estación central de registro (Figura I.2).

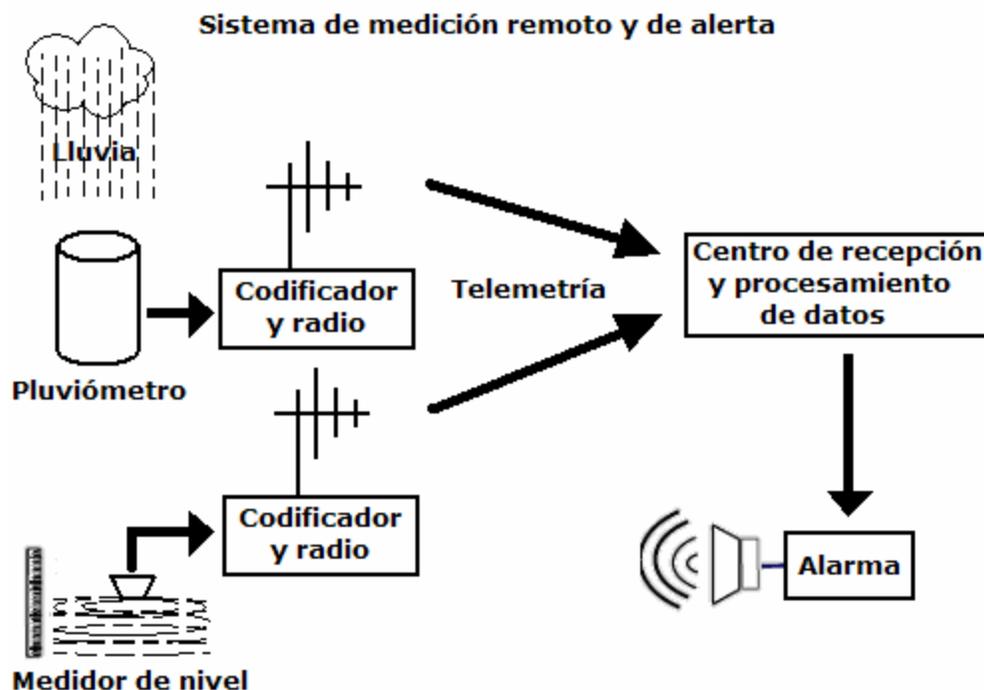


Figura 1.2. Sistema de medición y alerta pluviométrica e hidrométrica.

I.2. Planteamiento del problema

En los últimos años, México ha presentado eventos de precipitación extrema, que han puesto en evidencia la vulnerabilidad de la población a inundaciones, principalmente en regiones montañosas cercanas a las cuencas de ríos. De la misma forma, un déficit de lluvia, por ejemplo la sequía ocurrida en 1998, que ha sido una de las más severas que se tiene registrada en México.

Los eventos hidrometeorológicos extremos se presentan en dos formas:

- Lluvias intensas: aquellas que pueden provocar inundaciones, pérdidas humanas y económicas.
- Sequías: se reduce la cantidad de agua disponible, principalmente afecta al sector agrícola, también al sector urbano, generando un incremento en la demanda de servicios que requieren de agua.

Algunas estadísticas que presenta la CENAPRED son las siguientes:

- En el periodo de 1990-1999 ocurrieron en México 101 desastres naturales, en donde el fenómeno hidrometeorológico fue muy representativo, con el 59 %.
- 2767 muertos de 1980 a 1999 debido a este tipo de desastres.

- Daños materiales (en millones de pesos): directos \$44 023, indirectos \$1 449.

En México como en todo el mundo surgen empresas que desarrollan tecnología o que encargan proyectos a universidades y centros de investigación para la realización de equipos de medición de ciertos parámetros. Una vez hecho el sistema de medición, las empresas lo presentan a las entidades interesadas en su aplicación.

De esta manera nace la necesidad de esta tesis de maestría y se plantea la oportunidad de hacer un termopluviógrafo digital. ¿Cómo diseñar y desarrollar un dispositivo que permita obtener mediciones del agua de la lluvia, introducir éstas a un sistema de adquisición de datos, procesar esta información y enviarla hacia un equipo de cómputo, bajo un estándar de comunicación? ¿Qué tipos de dispositivos electrónicos usar? ¿Cuál de los lenguajes de programación es el más conveniente por utilizar? ¿Cuántos y cuáles estándares de comunicación se integrarán al sistema por realizar?. Estas preguntas y muchas más se irán presentando y se darán solución durante el avance del trabajo.

I.3. Propuesta de solución

Se propone realizar el diseño y desarrollo de un termopluviógrafo digital, esto es con el objetivo de contar con un equipo de medición que ofrecerá información referente a precipitaciones pluviales en lugares determinados.

La información adquirida por el termopluviógrafo se podrá revisar por medio de un *display* o a través de una computadora y llevar a cabo modificaciones en aquellos datos que se requieran. En la figura I.3 se muestra un esquema general de la propuesta de solución, zona punteada.

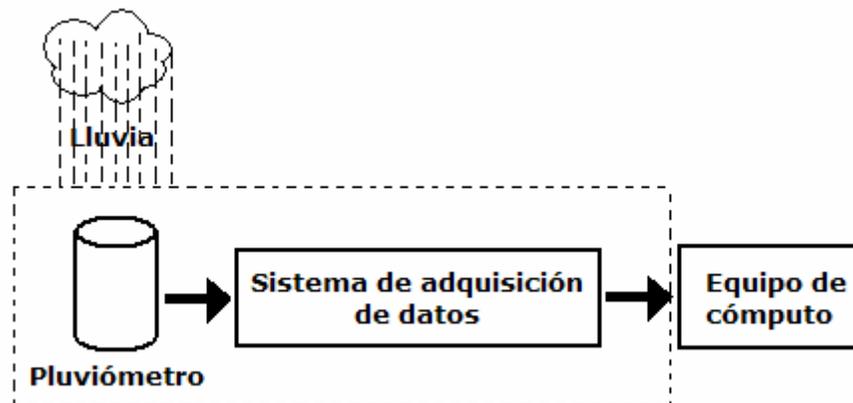


Figura I.3. Diagrama esquemático del termopluviógrafo.

Las características más importantes que deberá cumplir dicho dispositivo son las siguientes:

- Memoria electrónica de registro de datos para 20,000 eventos aproximadamente
- Autonomía con una batería tamaño C de 1.5 Volts
- Temperatura de operación de $-20\text{ }^{\circ}\text{C}$ a $70\text{ }^{\circ}\text{C}$
- *Display* de 2 líneas X 16
- Comunicación alámbrica serie RS-232 y USB
- Comunicación inalámbrica IrDA y *Bluetooth*

El sistema de medición estará conformado por los siguientes elementos:

- Un pluviómetro, que se encargará de coleccionar el agua de lluvia así como del envío de una señal de detección de evento.
- Un sensor que proporcione el parámetro de temperatura.
- Un microcontrolador, que será el encargado de procesar todos los datos.
- Memoria no volátil para el aseguramiento de los datos.
- Un *display* para la obtención de información del estado del sistema.
- Estándares de comunicación para la transmisión y recepción de datos hacia un equipo de cómputo en el exterior.
- Un módulo suministrador de energía que tendrá la característica de contar con circuitos que proporcionen tecnología de bajo consumo.

El termopluviógrafo digital estará formado por un sistema de registro y un pluviómetro tipo balancín, donde dichos elementos conformarán el *hardware* del sistema y además contará con un programa (*software*), el cual es una interfaz entre el usuario, el sistema de registro y un equipo de cómputo. Este *software* permitirá la programación del sistema de registro y la descarga de los datos de precipitación pluvial que se encuentren almacenados.

El termopluviógrafo digital de bajo consumo permitirá la adquisición y almacenamiento de datos de precipitación y de temperatura ambiente. Estará construido alrededor de un microcontrolador, con interfaz de entrada/salida para la adquisición, presentación y recuperación de información.

I.4. Metodología de diseño y desarrollo

Para el diseño y desarrollo del termoplúviógrafo digital, se utilizará una metodología, la cual contempla dos aspectos fundamentales: *hardware* y *software*.

El procedimiento de desarrollo del *hardware* y *software* del sistema de medición, así como el *software* para el equipo de cómputo, se ejecutará en un ciclo que consta de 4 etapas, las cuales son: diseño, implementación, medición y evaluación.

El resultado de cada etapa es la base de la que sigue, incluyendo la última, es decir, los resultados de la etapa de evaluación se toman para rediseñar el prototipo, implementarlo nuevamente y así sucesivamente. Debido a esa repetición o retroalimentación se le llama diseño iterativo.

En la tabla I.1 se muestra cada etapa, así como las descripciones de las tareas de cada una de ellas.

ETAPA	DESCRIPCIÓN DE TAREAS
DISEÑO	<p><i>HARDWARE</i></p> <p>Organización de tiempos y tareas Análisis por módulos del termoplúviógrafo digital Investigación y selección correspondiente a los diferentes microcontroladores, así como de componentes electrónicos Investigación de los protocolos de comunicación</p> <p><i>SOFTWARE</i></p> <p>Determinación de requerimientos del <i>software</i> Organización de tareas Propuestas de rediseño</p>
IMPLEMENTACIÓN	<p><i>HARDWARE</i></p> <p>Adquisición e implementación de los dispositivos electrónicos involucrados en el proyecto Generación de prototipos electrónicos</p> <p><i>SOFTWARE</i></p> <p>Evaluación y selección del lenguaje de programación de alto y bajo nivel para la interfaz y el microcontrolador Rediseño del programa de interfaz</p>
MEDICIÓN	<p><i>HARDWARE</i></p> <p>Aplicación de pruebas al prototipo del termoplúviógrafo Obtención de resultados</p> <p><i>SOFTWARE</i></p> <p>Aplicación de pruebas al programa de interfaz Rediseño de lenguajes de alto nivel: Ensamblador y <i>Visual Basic</i> Pruebas y resultados en el intercambio de datos</p>

Tabla I.1. Metodología de diseño y desarrollo. (Continúa)

EVALUACIÓN	En la etapa de evaluación están considerados tanto el <i>software</i> como el <i>hardware</i> del prototipo del termoplúviógrafo digital Análisis de los datos adquiridos tanto en el termoplúviógrafo digital como en el equipo de cómputo Verificación de los resultados y comparación con las diferencias Conclusiones Generación de nuevos objetivos
------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 1.1. Metodología de diseño y desarrollo.

Hasta ahora se ha hecho una descripción breve del entorno relacionado con la pluviometría, así como un bosquejo del termoplúviógrafo digital que se pretende realizar. A continuación se darán los conceptos teóricos de los elementos involucrados en este proyecto.

CAPÍTULO II

FUNDAMENTOS TEÓRICOS

Este capítulo comprenderá una introducción a los conceptos y elementos que contiene la pluviometría; también se tratará de manera generalizada a los microcontroladores. De modo similar se tratarán los siguientes estándares de comunicación: RS-232, USB, IrDA y *Bluetooth*.

II.1. Pluviometría

La Pluviometría es una parte de la meteorología que se encarga de estudiar la distribución geográfica y estacional de las precipitaciones pluviales.

Entre los conceptos que destacan en la pluviometría se encuentran principalmente: la precipitación, la intensidad de precipitación, el hietograma, el pluviómetro, el pluviógrafo y el termopluviógrafo.

Las definiciones correspondientes a cada uno de estos conceptos son:

Precipitación. Se define como cualquier agua atmosférica recogida sobre la superficie terrestre. Esto incluye básicamente: lluvia, nieve, granizo, así como rocío y escarcha, puesto que las dos últimas en algunas regiones constituyen una parte pequeña pero apreciable de la precipitación total.

La manera en que se define la unidad de medición de la precipitación pluvial es la siguiente: si el agua de la lluvia se acumulara en un terreno plano (superficie horizontal de 1 m^2) sin escurrir ni evaporarse, la precipitación se mediría en función de la altura de la capa de agua. Por ejemplo, la medición de 1 mm de altura representaría 1 litro de agua precipitada en 1 m^2 de la superficie.

Intensidad de precipitación. La intensidad de precipitación es la precipitación que se tiene en un intervalo de tiempo dado.

Pluviómetro. Un pluviómetro es un instrumento que mide la cantidad de agua precipitada en un determinado lugar. La unidad de medición es en milímetros (mm) y representa la precipitación pluvial. De acuerdo a sus dimensiones físicas, los pluviómetros tienen diferentes capacidades, por ejemplo de: 0.1 mm, 0.2 mm, 0.5 mm, etc.

Pluviógrafo o hietógrafo. Un pluviógrafo es en realidad un pluviómetro al que se le ha agregado un sistema de registro.

Hietograma. Un hietograma (del griego Hietos, lluvia, y grama, gráfica) es un histograma (gráfica de barras) que expresa la precipitación pluvial en función del tiempo.

Termopluviógrafo. Es un pluviógrafo que además obtiene la lectura de la temperatura.

II.1.1. Componentes del termopluviógrafo

Los principales elementos que componen el termopluviógrafo son: el pluviómetro y el sistema de registro. Aunque existen en el mercado diferentes tipos de pluviómetros, para nuestra aplicación se utilizará el pluviómetro tipo balancín.

II.1.1.1. Pluviómetro tipo balancín

El pluviómetro tipo balancín (Figura II.1) obtiene la precipitación de lluvia que cae en una área determinada y transmite la información que se genera al sistema de registro.

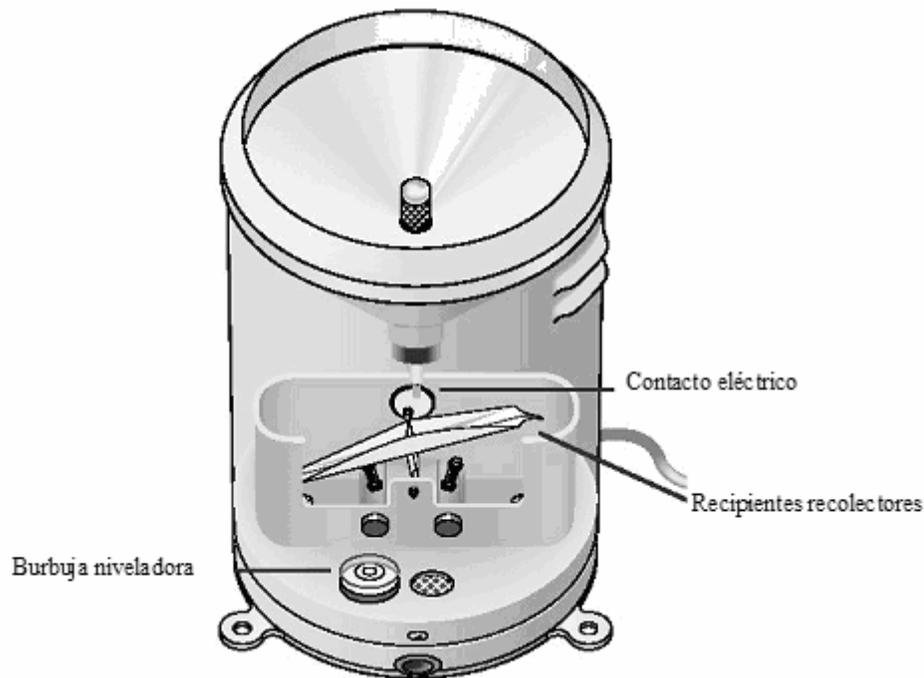


Figura II.1. Pluviómetro tipo balancín.

Operación del pluviómetro

El pluviómetro de balancín tiene dos colectores de agua calibrados que se encuentran suspendidos sobre un eje de rotación a manera de balanza. Mediante un embudo, la lluvia que se colecta es conducida hacia uno de los colectores. Cuando uno de los colectores se llena (cantidad de agua establecida y calibrada en fábrica), el peso del agua hace que por gravedad el balancín caiga vaciando su contenido (un evento) y colocando al otro colector en posición para recibir agua. Así sucesivamente se van llenando ambos recipientes alternándose mientras continúa la lluvia.

Normalmente los pluviómetros tipo balancín están calibrados de fábrica y establecen un valor de medición, esto se puede explicar con un ejemplo: si un balancín es de 0.5 mm y aplica un vaciado de agua, significa que cada evento de vaciado de agua del balancín, representa medio litro de agua vertida en una superficie plana cerrada de una área de 1 m², y la altura de la capa de agua es precisamente de 0.5 mm.

El movimiento del balancín en su transición se aprovecha para registrar un evento, es decir, la cantidad de agua almacenada en el colector. Esto se logra a través de un acoplamiento mecánico entre el balancín y un interruptor de mercurio, a partir del cual se genera y detecta el evento.

II.1.1.2. Sistema de registro de eventos

El sistema de registro de eventos (Figura II.2) se conformará con los siguientes elementos: microcontrolador, memoria de datos pluviales, sensor de temperatura, módulo de despliegue (*display*), sistema de alimentación de energía, acondicionamiento de señales y puertos de comunicación.

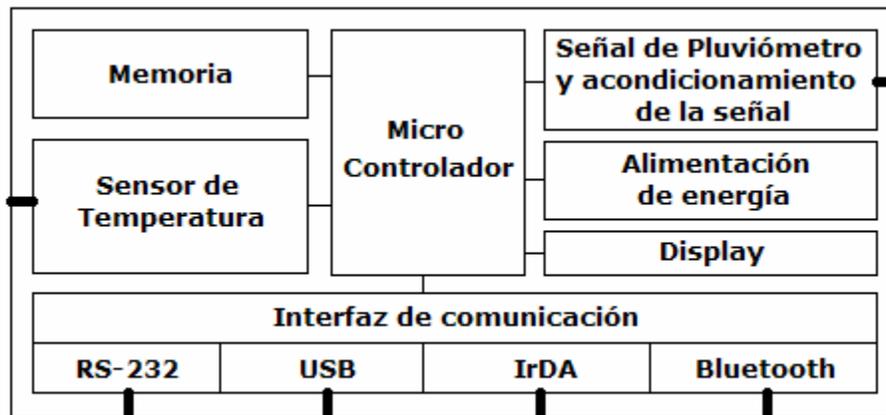


Figura II.2. Elementos del sistema de registro de eventos.

Las características principales de cada uno de los elementos mencionados son:

Microcontrolador. Es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de entrada y salida, todos incluidos en un circuito integrado. Es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario y son introducidos en éste a través de un programador.

Memoria. Las memorias son dispositivos de almacenamiento de información binaria permanente o volátil. El sistema de adquisición de datos requerirá el almacenamiento permanente de un gran número de datos binarios y disponer de ellos cuando sea necesario.

Sensor de temperatura. Es un dispositivo electrónico que se coloca en un medio ambiente y nos proporciona un voltaje o una corriente eléctrica equivalente a la temperatura existente.

Display. Es un visualizador o un dispositivo que permite mostrar información al usuario. Para el caso del termopluviógrafo digital, presentará los parámetros que se quieran monitorear.

Sistema de alimentación de bajo consumo. Este módulo suministrará la energía suficiente al sistema para que pueda operar normalmente y proporcionará además una alternativa de bajo consumo, es decir, un ahorro en la energía.

Acondicionamiento de señales. Este módulo será el encargado de acondicionar las señales de entrada hacia el adquisidor de datos, provenientes del sensor de temperatura y del pluviómetro. Se realizará esta acción para que los datos de eventos que se adquieran sean confiables. Debido a lo anterior se describe en que consistirán tales acondicionamientos.

Acondicionamiento de señal de temperatura. El sensor de temperatura debería proporcionar una señal de voltaje que sea función únicamente de la temperatura, pero la realidad es que la salida se presenta inestable, esto es debido a señales de ruido generado por varias fuentes (perturbaciones en fuente de alimentación, ruido térmico, etc.). Para estabilizar esta señal se realizará un circuito de acondicionamiento.

Acondicionamiento de señal del pluviómetro. Se deberá de acondicionar esta señal porque al ocurrir un evento en el pluviómetro, se producen rebotes mecánicos y estos si no se acondicionan proporcionarán información falsa.

Interfaz de comunicación. Es el medio a través del cual se realiza la comunicación de datos entre el adquisidor de datos y el equipo de cómputo.

Operación del sistema de registro de eventos

El sistema de registro del termopluviógrafo digital operará de la siguiente manera: cuando se produzca un evento, el pluviómetro enviará un pulso hacia el sistema de registro. Éste se encargará de tomar y procesar la información correspondiente y la almacenará en una memoria de estado sólido, de donde posteriormente se podrán transferir los datos a un equipo de cómputo, a través del puerto de comunicación para su procesamiento. De la misma manera el sensor evaluará constantemente la temperatura, enviará la señal analógica al microcontrolador para que la convierta a digital y la procese.

II.2. Microcontroladores

Desde la aparición del transistor la evolución de la electrónica ha sido sorprendente. Actualmente podemos encontrar dispositivos cada vez

más complejos, ubicados en encapsulados cada vez más pequeños; un ejemplo de lo anterior son los microcontroladores.

Un microcontrolador es un dispositivo electrónico encapsulado en un circuito integrado, capaz de ejecutar un programa. El microcontrolador reúne en un sólo integrado (Figura II.3): microprocesador, memoria de programa, memoria de datos y puertos de entrada/salida (E/S). Además, también dispone de otras características especiales como: puertos serie y paralelo, comparadores, *timer*, convertidores analógico/digital (A/D), etc.

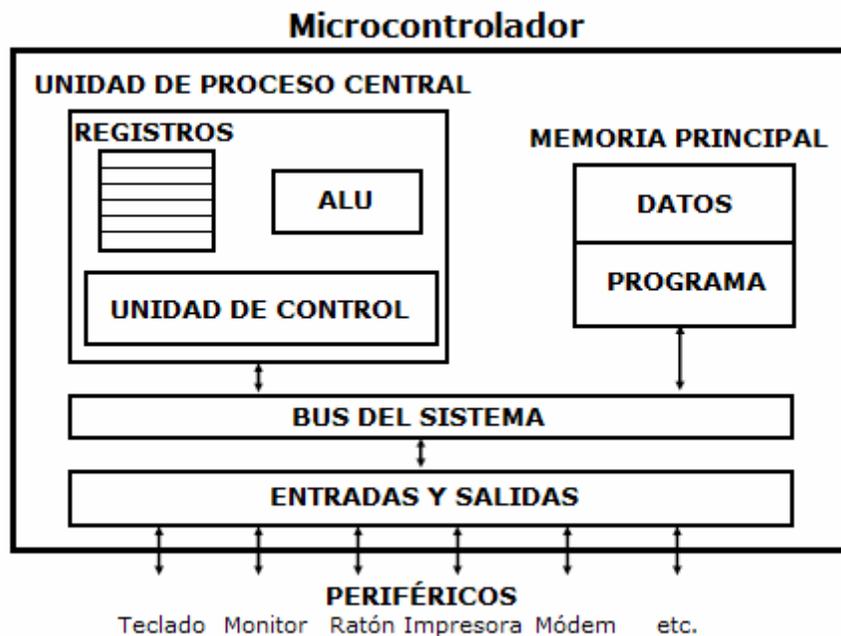


Figura II.3. Estructura básica de un microcontrolador.

En el mercado existe una gran variedad de microcontroladores de marcas y características distintas que se pueden utilizar dependiendo de la aplicación a realizar.

Un microcontrolador ejecuta instrucciones. Las instrucciones son leídas de la memoria del programa para ejecutarlas una detrás de otra. La memoria de programa contiene las instrucciones que queremos que el microcontrolador ejecute. El conjunto de instrucciones se denomina programa.

Programar un microcontrolador consiste en introducir el programa en la memoria de éste. Las instrucciones son operaciones simples como: sumar, restar, escribir en un puerto, activar un *bit* de un dato, etc. Mediante estas instrucciones básicas se pueden realizar operaciones más complejas y así llegar al objetivo de la aplicación.

II.2.1. Componentes de los microcontroladores

Los microcontroladores están compuestos por: la Unidad de Procesamiento Central (CPU), memorias, registros, puertos de entrada-salida, osciladores, temporizador/contador (T/C) y *watchdog*.

Unidad de Procesamiento Central

La Unidad de Procesamiento Central es el corazón del microcontrolador. Lee la instrucción almacenada en la memoria del programa, la interpreta (decodifica) y la ejecuta. La CPU está compuesta de: registros, la Unidad Lógica Aritmética (ALU), decodificador de instrucciones y circuitos de control.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

- *Procesador CISC* (Computadora con *Set* de Instrucciones Complejas): una gran cantidad de procesadores utilizados en los microcontroladores se basan en la filosofía CISC. Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.
- *Procesador RISC* (Computadora con *Set* de Instrucciones Reducidas): tanto la industria de las computadoras comerciales como la de los microcontroladores están dirigiéndose hacia la filosofía RISC. En estos procesadores, el conjunto de instrucciones máquina es muy reducido y las instrucciones son simples, y generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el *hardware* y el *software* del procesador.
- *Procesador SISC* (Computadora con *Set* de Instrucciones Específicas): en los microcontroladores destinados a aplicaciones muy concretas, el conjunto de instrucciones SISC, además de ser reducido, es específico, o sea, las instrucciones se adaptan a las necesidades de aplicaciones previstas.

Memoria

La memoria es el lugar donde se almacenan datos. Dependiendo de la arquitectura del microcontrolador existen dos tipos de memorias: la memoria del programa y la memoria de datos. Estas dos memorias son

independientes entre ellas, teniendo tamaño y longitudes de palabra distintas. Cada bloque posee su propio *bus*, de tal forma que el acceso a cada uno puede producirse durante el mismo ciclo del oscilador.

Memoria de programa. A esta memoria también se le denomina memoria de instrucciones y memoria de solo lectura (ROM). Aquí es donde se escriben los programas. Existen diferentes tipos de memorias: ROM, PROM, EPROM, EEPROM y FLASH.

Las características de cada una de las memorias mencionadas son:

- *Memorias ROM.* Las memorias ROM son utilizadas para almacenar datos permanentemente. Las grabaciones regularmente se hacen en fábrica.
- *Memorias PROM.* Estas memorias son semejantes a las ROM, además se graban eléctricamente y lo puede realizar el usuario después de su fabricación. Esta memoria permite sólo una grabación.
- *Memorias EPROM.* Este tipo de memoria puede ser grabada por el usuario y hacerlo tantas veces como sea necesario. Cuando esta memoria ha sido grabada, los datos se pueden borrar si la ventana del circuito integrado se expone a la luz ultravioleta.
- *Memorias EEPROM.* Son memorias similares a las EPROM, sólo que la operación de grabación es totalmente eléctrica.
- *Memoria FLASH.* Son memorias nombradas *FLASH* por la rapidez con que se pueden reprogramar. Utilizan un borrado eléctrico selectivo a nivel de bloque de *bytes*.

La cantidad de memoria depende de la gama de microcontroladores, así como de los fabricantes. Para programas simples se opta por disponer de microcontroladores con menor memoria y más baratos, mientras que para programas de mayor tamaño se seleccionan microcontroladores de más memoria.

Memoria de datos. La memoria que se constituye de datos también se denomina memoria de lectura y escritura o memoria de acceso aleatorio (RAM), ésta es utilizada por el microprocesador, así como por otros dispositivos de *hardware*. La memoria RAM se clasifica en dos tipos de memorias diferentes, las cuales son DRAM y SRAM.

- *Memorias DRAM.* Estas memorias son del tipo dinámico y almacenan los datos similarmente a la carga de un capacitor. Por lo tanto tienden a descargarse, por lo que es necesario refrescar la

información contenida en ellas periódicamente. Su fabricación es barata y su operación es lenta.

- *Memorias SRAM.* Las memorias SRAM son del tipo estático y almacenan los datos en circuitos biestables, por lo tanto, no se necesita refrescar los datos. Estas memorias son más rápidas que las de tipo DRAM y su fabricación resulta más cara.

La memoria de datos se organiza en bancos, y la cantidad de estos depende del tipo de microcontrolador, así como del fabricante. Esta memoria de datos funciona de forma similar al banco de registros de un procesador.

Registros. Los microcontroladores utilizan una arquitectura basada en registros. Esto significa que todos los objetos del sistema (posiciones de memoria, temporizadores, puertos de entrada/salida, etc.) están implementados físicamente como registros. Todos los registros están ubicados en una posición específica de la memoria, por lo cual sus posiciones implementan registros de propósito general y registros de propósito especial.

- *Registros de propósito general:* realizan operaciones con datos y guardan direcciones, por ejemplo los puertos de E/S.
- *Registros de propósito especial:* son de funcionamiento y configuración, tales como: el registro Contador de Programa (PC) y el registro *stack*.

Puertos de entrada-salida. Para los microcontroladores son un recurso importante los puertos de entrada/salida, con los cuales se comunican a los periféricos del mundo exterior; dependiendo del microcontrolador que se utilice, estos tendrán más o menos puertos de E/S. Dichos puertos pueden ser digitales, analógicos, multiplexados, etc.

Algunos microcontroladores permiten utilizar a los puertos como comparadores y convertidores, pero esas características en particular dependen de cada microcontrolador. Los puertos de entrada/salida utilizan la denominación de puerto A, puerto B, etc., dependiendo de la cantidad de puertos que tengan.

Contador de programa. Este registro es totalmente equivalente al de todos los microprocesadores y contiene la dirección de la próxima instrucción a ejecutar. Automáticamente se incrementa al ejecutar cada instrucción, de manera que la secuencia natural de ejecución del programa es lineal, es decir, una instrucción después de la otra.

Al reinicializar el microprocesador, todos los *bits* del registro PC toman valor '1', de manera que la dirección de arranque del programa es siempre la última posición de la memoria de programa. En esta posición se deberá poner una instrucción de salto al punto donde verdaderamente se inicia el programa.

Stack. Es una memoria interna dedicada de tamaño limitado, tiene un puntero interno que lo direcciona; separada de las memorias de datos y de programa, inaccesible al programador, y organizada en forma de pila. Como en toda memoria tipo pila, los datos son recuperados de manera tal que el primero que entra es el último que sale. Es utilizada solamente para guardar las direcciones de retorno de subrutinas e interrupciones en forma automática. Permite guardar una copia completa del contador de programa.

Las posiciones de la pila en un microcontrolador están limitadas a no hacer uso intensivo del anidamiento de subrutinas. Sólo se pueden anidar dos niveles de subrutinas, es decir, una subrutina que es llamada desde el programa principal, puede a su vez llamar a otra subrutina, pero esta última no puede llamar a una tercera, porque se desborda la capacidad del *stack*, que sólo puede almacenar dos direcciones de retorno. Aunque esto sea un problema es una buena solución de compromiso, ya que los microcontroladores están diseñados para aplicaciones de alta velocidad en tiempo real, en las que las demoras adicionales que ocasiona un excesivo anidamiento de subrutinas, son inaceptables.

Oscilador externo

Todo microcontrolador requiere un circuito externo que le indique la velocidad a la que debe trabajar, es decir, sincroniza el funcionamiento del sistema. Este circuito se conoce con el nombre de oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema.

Los microcontroladores pueden utilizar cuatro tipos diferentes de osciladores y son los siguientes.

- RC: oscilador con resistencia y condensador.
- XT: cristal de cuarzo.
- HS: cristal de alta velocidad.
- LP: cristal para baja frecuencia y bajo consumo de potencia.

Por ejemplo: un microcontrolador que ejecuta una instrucción en un ciclo de reloj, si utiliza un cristal de 1 MHz, la frecuencia efectiva de trabajo es de 1 MHz, por lo que cada instrucción se ejecuta en 1 μ s. El cristal debe instalarse acompañado de dos condensadores y se conecta como se muestra en la figura II.4.

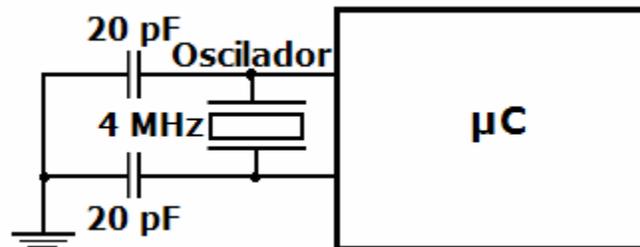


Figura II.4. Oscilador de cristal.

Temporizadores

La cantidad de temporizadores depende del microcontrolador que se esté utilizando, si un microcontrolador dispone de dos temporizadores, uno de ellos actúa como principal y sobre él recae el control de tiempos en las operaciones del sistema, mientras el otro recibe el nombre de *Watchdog*.

Temporizador/Contador. Una exigencia en las aplicaciones de control es la regulación estricta de los tiempos que duran las diversas acciones que realiza el sistema. El dispositivo típico destinado a gobernar los tiempos recibe el nombre de temporizador o *timer* y básicamente consiste en un contador ascendente o descendente que determina un tiempo específico entre el valor que se carga y el momento en que se produce su desbordamiento o paso por 0.

Watchdog. El *Watchdog* vigila que el programa no entre en una malla cerrada y dejen de ejecutarse las instrucciones secuenciales del mismo, tal como lo ha previsto el diseñador. Para hacer esta labor de vigilancia el *Watchdog* realiza un recorrido por la CPU cada tiempo determinado y comprueba si el programa se ejecuta normalmente; en caso contrario, por ejemplo si el programa está detenido en un bucle infinito o a la espera de algún suceso que no se produce, el *Watchdog* genera una señal y provoca un *reset* del sistema, reiniciándolo todo.

II.2.2. Arquitectura de los microcontroladores

Los microcontroladores se pueden clasificar por la forma en que se tiene acceso a las memorias de datos y de programa. De ahí nacen las arquitecturas Von Neumann y Harvard.

Arquitectura Von Neumann

Tradicionalmente, las computadoras y microprocesadores siguen el modelo propuesto por John Von Neumann, en el cual la unidad de procesamiento central está conectada a una memoria única que contiene las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones está fijado por el ancho del *bus* de la memoria. Esto limita la velocidad de operación del microprocesador, ya que no se puede buscar en la memoria una nueva instrucción antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior.

Arquitectura Harvard

Este tipo de arquitectura conecta de forma independiente y con dos *buses* distintos la memoria de instrucciones del programa y la de datos. Permite a la CPU acceder simultáneamente a las dos memorias (Figura II.5). Esto proporciona mayor velocidad además de numerosas ventajas al funcionamiento del sistema.

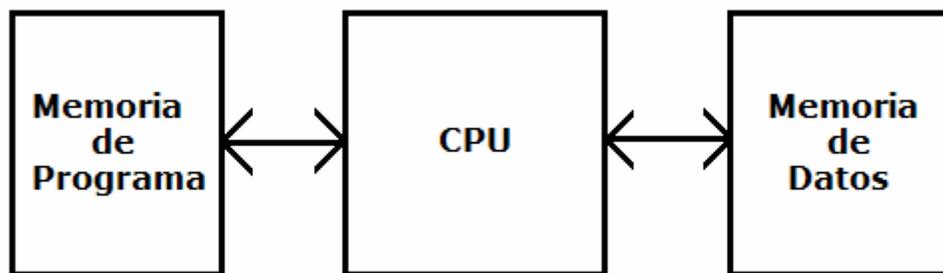


Figura II.5. Arquitectura Harvard.

Por su fabricación, los microcontroladores se pueden clasificar en microcontroladores de arquitectura cerrada y de arquitectura abierta.

Microcontroladores de arquitectura cerrada

En este tipo de arquitectura el microcontrolador contiene recursos específicos, los cuales no permiten ningún tipo de modificación, es decir, no admiten ningún tipo de variaciones ni de ampliaciones. La aplicación a la que se destina debe encontrar en su estructura todo lo que precisa.

Microcontroladores de arquitectura abierta

Este tipo de microcontroladores, además de tener una estructura interna determinada, permiten la ampliación de sus puertos para sacar al exterior los *buses* de datos, dirección y control, con lo que se posibilita la ampliación de la memoria y las E/S con dispositivos externos.

II.2.3. Software en los microcontroladores

Para programar un microcontrolador se requiere un *software*, este *software* es un lenguaje de programación de nombre ensamblador, también suele utilizarse otro, el lenguaje C; con dicho lenguaje se tiene un dominio absoluto del funcionamiento del sistema.

Un programa está compuesto de instrucciones, éstas pueden realizar operaciones, manipular y trasladar datos de una localidad a otra, a través de sus modos de direccionamiento, y trabajar con interrupciones.

Cuando se realiza la programación de un microcontrolador, al momento de especificar los datos y operandos, es posible hacerlo de formas distintas, llamadas: modos de direccionamiento. Estos modos de direccionamiento son principalmente: inmediato, directo e indirecto.

- *Modo Inmediato*. Es cuando se utiliza el valor literal, colocado inmediato a la instrucción.
- *Modo Directo*. En la instrucción se apunta la localidad de memoria que contiene el dato que se va a utilizar.
- *Modo Indirecto*. Dentro de la instrucción se utiliza un registro como operando que accesa a una localidad de memoria que contiene el dato.

Una interrupción consiste en una detención del programa en curso, para ejecutar una determinada rutina que atienda la causa que ha provocado la interrupción. Tras terminar la rutina de interrupción, se regresa al programa principal en el punto en que se abandonó.

II.3. Comunicaciones alámbricas

Ante el desarrollo de la tecnología aplicada a las comunicaciones, se ha hecho necesario diseñar y realizar especificaciones para el correcto uso y funcionamiento de equipos para la transmisión de información, y que son fabricados por diferentes empresas. Después de hacer los estudios correspondientes a la interfaz de comunicaciones que se deben establecer, se finaliza con la creación de estándares para tales equipos

de comunicación. Estos estándares especifican desde los parámetros más sencillos hasta los más complejos; por ejemplo, podemos mencionar: voltaje de alimentación, tipo de comunicación, velocidad de transmisión de datos, tipo de interfaz, características físicas, etc.

Los estándares de comunicación en forma alámbrica, que actualmente son más utilizados en equipos de cómputo, funcionan bajo los estándares seriales RS-232, USB y el estándar paralelo. En nuestro proyecto son de particular interés los estándares seriales.

II.3.1. Estándar RS-232

El RS-232 es uno de los estándares más utilizados frecuentemente como interfaz en la comunicación entre computadoras con equipos externos. Este estándar existe en todas las computadoras del mundo, actualmente es el sistema más común para el intercambio de información, todos los equipos tienen como mínimo uno.

Definición

RS-232 es una interfaz que designa una norma para el intercambio de datos binarios en forma serial entre un equipo terminal de datos (DTE), por ejemplo una computadora, y un equipo de comunicación de datos (DCE), por ejemplo, un módem. Existen ocasiones en que interesa realizar otro tipo de conexiones, por ejemplo, computadora con computadora. Evidentemente, en el caso de interconexión entre el mismo tipo de dispositivos, se requerirá la conexión de un DTE con otro DTE.

Objetivos

La necesidad de unificar criterios en la comunicación fue la que condujo a la realización de este estándar, teniendo como principales objetivos:

- Asegurar una comunicación confiable.
- Habilitar la interconexión de equipos fabricados por diferentes empresas.

Historia

En 1960, reunida la Asociación de Industrias Electrónicas (EIA), elaboró un estándar para equipos de comunicación de datos. En aquel tiempo, el intercambio de información entre computadoras originaba errores en la

transmisión y recepción de datos analógicos, además, estas actividades eran realmente complejas.

A más de 40 años que el estándar nació, han sido publicadas tres modificaciones, la más reciente en 1997 llamada EIA-232F. En esta modificación se cambió el nombre de RS-232 a EIA-232, así que, de aquí en adelante se hará la referencia del nombre actual.

II.3.1.1. Especificaciones del estándar

Con base en los dos puntos mencionados como objetivos, se pretendía fomentar los beneficios en la producción en serie, así como la competencia entre los fabricantes de los equipos de cómputo. De estas ideas surgió el estándar RS-232, implementando las siguientes especificaciones:

- Voltajes de señales
- Tiempos en las señales
- Función de señales
- Protocolo para el intercambio de datos
- Conexiones mecánicas

El estudio de las especificaciones del estándar de comunicación EIA-232 se va a realizar clasificándolo en tres aspectos fundamentales: mecánico, eléctrico y funcional.

Aspecto mecánico

El aspecto mecánico hace referencia a todo lo relacionado con lo físico, es decir: los conectores, tipos de conectores, las terminales, definición de las terminales, etc. Lo anterior se describe en las especificaciones físicas y especificaciones de los conectores.

Especificaciones físicas. La interfaz EIA-232 está diseñada para llevar al cabo comunicaciones en distancias cortas, aproximadamente de unos 15 m o menos.

La interfaz puede trabajar en comunicación síncrona o asíncrona y con los siguientes tipos de canal:

- *Simplex*: en un canal *simplex* los datos siempre viajarán en una dirección, por ejemplo desde un dispositivo DCE a un DTE.
- *Half duplex*: en un canal *half duplex*, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de

tiempo; después la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección.

- *Full duplex*: en un canal *full duplex*, los datos pueden viajar en ambos sentidos simultáneamente. Las terminales de control de flujo de datos de la EIA-232 se usan para resolver los problemas asociados con este modo de operación, por ejemplo: la dirección en que los datos deben viajar en un instante determinado.

El estándar EIA-232 consiste en un conector tipo DB-25 (25 terminales), aunque es normal encontrar la versión DB-9 (9 terminales), más barato e incluso más extendido para cierto tipo de periféricos y de aplicaciones. En la figura II.6 se pueden observar físicamente los conectores mencionados.

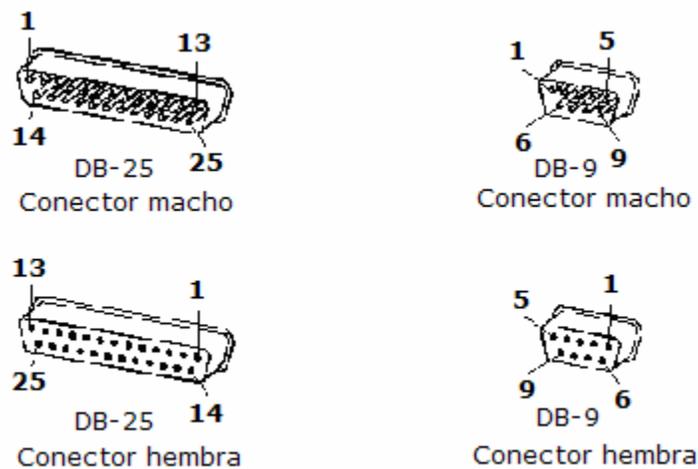


Figura II.6. Tipos de conectores.

La especificación del estándar EIA-232, dice que una conexión está definida por la unión de un cable desde un dispositivo hasta el otro. Hay 25 conexiones en la especificación completa, pero regularmente se encuentran menos de la mitad de éstas en una interfaz determinada. La respuesta es simple, una interfaz *full duplex* puede obtenerse con la conexión de únicamente 3 cables.

Especificaciones de los conectores. La tabla II.1 muestra la especificación y descripción de cada una de las terminales utilizadas en el estándar EIA-232, para los conectores de 25 y 9 terminales.

Señal	Clave	DB 25	DB 9	Función	Descripción
<i>Chasis Ground</i>	CG	1		Tierra de Chasis	Es un mecanismo para asegurar que los chasis de los dispositivos estén al mismo potencial, y para impedir una descarga eléctrica al operador. Este circuito no es usado como referencia por ningún otro voltaje y es optativo. Es la tierra de seguridad del sistema.
<i>Transmitted Data</i>	TxD	2	3	Salida	Es la trayectoria de los datos que se envían desde el DTE al DCE, es decir, es la línea de transmisión de datos.
<i>Received Data</i>	RxD	3	2	Entrada	Esta línea es el recorrido por medio del cual los datos se envían desde el DCE al DTE. Línea de recepción de datos.
<i>Ready To Send</i>	RTS	4	7	Salida	Es la señal que indica que el DTE desea enviar datos al DCE. En operación normal, la línea de RTS estará en <i>OFF</i> ('1' lógico o MARCA). Una vez que el DTE tiene los datos para enviar y se ha determinado que el canal no está ocupado, colocará RTS en <i>ON</i> ('0' lógico o ESPACIO) y esperará un estado <i>ON</i> en el CTS desde el DCE, al tiempo que puede comenzar a enviar datos. Una vez que el DTE culminó el envío, vuelve a fijar RTS a <i>OFF</i> .
<i>Clear To Send</i>	CTS	5	8	Entrada	Esta línea es la señal que indica que el DCE está preparado para aceptar datos desde el DTE. En operación normal, la línea CTS estará en la condición <i>OFF</i> . Cuando el DTE activa a RTS, el DCE permitirá que los datos sean enviados. En este momento, el DCE colocará al CTS en <i>ON</i> , entonces el DTE enviará los datos. Cuando la línea RTS desde el DTE retorna a la condición <i>OFF</i> , el DCE descarga el canal y entonces CTS se restaura en <i>OFF</i> .

Tabla II.1. Especificación de terminales en conectores EIA-232. (Continúa)

<i>Data Set Ready</i>	DSR	6	6	Entrada	Esta terminal es la señal que informa al DTE que el DCE está activo. Es puesta normalmente en <i>ON</i> por el DCE al encenderse éste. Un DTE típico deberá tener un DSR entrante a fin de desempeñarse normalmente.
<i>Signal Ground</i>	SG	7	5	Señal de Tierra	Esta terminal corresponde al voltaje de tierra, al que todos los otros voltajes están referenciados. Debe estar presente en toda interfaz EIA-232.
<i>Data Carrier Detect</i>	DCD	8	1	Entrada	Es la señal o línea en la que el DCE informa al DTE que tiene una portadora entrante. Puede ser usado por el DTE para determinar si el canal está desocupado, y que el DTE pueda pedir un RTS.
<i>Pos. Test Voltage</i>		9		Entrada	No aplica
<i>Neg. Test Voltage</i>		10		Entrada	No aplica
(No tiene uso)		11		Sin uso	No aplica
<i>Secondary Received Line Signal Detector</i>	SDCD	12		Entrada	Esta señal es equivalente a la de la terminal 8, pero se refiere al canal secundario. Canal primario: terminales 2, 3, 4 y 5. Canal secundario: terminales 13, 14, 16 y 19.
<i>Secondary Clear To Send</i>	SCTS	13		Entrada	Esta señal es equivalente a la señal 5 correspondiente del primer canal de comunicación. La velocidad de transmisión de datos es mucho menor en el canal secundario, para incrementar la confiabilidad de la comunicación
<i>Secondary Transmitted Data</i>	STxD	14			Esta señal es equivalente a la señal 2 correspondiente del primer canal de comunicación. La velocidad de transmisión de datos es mucho menor en el canal secundario para incrementar la confiabilidad de la comunicación.

Tabla II.1. Especificación de terminales en conectores EIA-232. (Continúa)

<i>Transmit Clock</i>	TxC	15		Entrada	Esta terminal proporciona el reloj para la sección de transmisión de un DTE síncrono. Debe estar presente sobre las interfaces síncronas. Puede o no funcionar a la misma velocidad que corre el receptor.
<i>Secondary Received Data</i>	SRxD	16		Entrada	Esta terminal es equivalente a la señal 3 correspondiente del primer canal de comunicación. La velocidad de envío de información es mucho menor en el canal secundario, para incrementar la confiabilidad de la comunicación.
<i>Receive Clock</i>	RxC	17		Entrada	Esta terminal proporciona el reloj para la sección de receptor de un DTE síncrono. Ambos TxC y RxC son provistos por el DCE. Al igual que TxC, este circuito debe estar presente sobre las interfaces síncronas. Puede o no funcionar a la misma velocidad que corre el transmisor.
<i>Local Loopback</i>	LL	18		No tiene Uso	Esta señal es generada por el DTE y es usada para colocar el módem dentro de un estado de prueba.
<i>Secondary Request To Send</i>	SRTS	19		Salida	Esta terminal es equivalente a la señal 4 correspondiente del primer canal de comunicación. En el canal secundario la velocidad de transmisión es mucho menor para incrementar la confiabilidad de la comunicación.
<i>Data Terminal Ready</i>	DTR	20	4	Salida	Esta línea proporciona la señal que informa al DCE que el DTE está activo. Es normalmente puesta en <i>ON</i> por el DTE al encenderse éste. Un DCE típico deberá tener un DTR entrante a fin de desempeñarse normalmente.
<i>Remote Loopback</i>	RL	21		Entrada	Esta señal es generada por el DTE y es usada para colocar el módem remoto dentro de un estado de espera.

Tabla II.1. Especificación de terminales en conectores EIA-232. (Continúa)

<i>Ring Indicator</i>	RI	22	9	Entrada	Esta señal es importante cuando el DCE es un módem, y es mantenido en voltaje positivo ('0' lógico), cuando una señal sonora está siendo recibida desde la línea telefónica. Mientras la señal suene, esta terminal se mantiene en '0' lógico.
<i>Data Signal Rate Selector</i>	DSR	23		Salida	Tal señal puede originarse en DTE o DCE, pero no en ambos, y es usada para seleccionar una o dos velocidades de transmisión. Si mantiene un nivel lógico '0', selecciona la transmisión más alta.
<i>External Transmitter Clock</i>	ETC	24		Salida	Esta señal es de regulación de tiempo, proporcionada por el DTE para uso del módem. Utilizada sólo cuando TxC y RxC no están en uso. La transición del '1' lógico al '0' lógico (voltaje negativo a voltaje positivo) indica el tiempo central del dato. Esta señal es proporcionada cuando el DTE está encendido, sin tener en cuenta las condiciones de otras señales.
<i>Test Mode</i>	TM	25		Salida	Esta señal es relevante sólo cuando el DCE es un módem. Si su nivel lógico es '0', indica que el módem está en una condición <i>Loopback</i> o <i>Remote Loopback</i> . También puede estar en una autoprueba interna.

Tabla II.1. Especificación de terminales en conectores EIA-232.

Generalmente en las comunicaciones bajo el estándar EIA-232, se utilizan tres terminales para la transmisión y recepción de la información: dos son canales portadores de datos (TxD y RxD) y el tercero es un conductor de señal de tierra (SG), que sirve como camino de retorno de la corriente de la señal. Las 22 terminales restantes sirven como terminales de control.

Aspecto eléctrico

Esta parte especifica las características eléctricas que debe cumplir el estándar.

Especificaciones eléctricas. El estándar EIA-232 utiliza una lógica bipolar negativa, en donde una señal de voltaje negativa representa un '1'

lógico, y un voltaje positivo representa un '0' lógico. El protocolo de comunicación EIA-232 define los niveles de voltaje de la siguiente manera: de -25 Volts a -3 Volts (típicamente se considera -12 Volts) para representar al '1' lógico y de +3 Volts a +25 Volts (típicamente se considera +12 Volts) para representar el '0' lógico. Dependiendo de la funcionalidad del circuito de intercambio de datos, los valores eléctricos se manejan como datos binarios o como señales de control *ON* y *OFF*. Cualquier voltaje entre -3 Volts y +3 Volts tiene un nivel lógico no definido.

Aspecto funcional

El aspecto funcional trata la explicación de todo lo relacionado con el funcionamiento del estándar EIA-232. Se describen las características de: intercambio de datos, comunicación serial, la velocidad del puerto, el tiempo de flujo de un *bit*, codificación de las señales y las principales formas de conexiones físicas.

Protocolo para el intercambio de datos. Un dispositivo muy importante dentro del estándar EIA-232 es la unidad Transmisora Receptora Asíncrona Universal (UART), que transmite y recibe datos utilizando la interfaz EIA-232.

Generalmente, la computadora controla el puerto de comunicaciones seriales por medio de la UART o de una unidad Transmisora Receptora Asíncrona-Síncrona Universal (USART). El principal dispositivo utilizado para construir una interfaz EIA-232 es la UART, la cual es totalmente programable.

Es en la UART donde se colocan los parámetros que regirán la comunicación serial, como son: la velocidad de transmisión (*baud rate*), el número de *bits* de datos, el número de *bits* de inicio, el número de *bits* de paro, el tipo de paridad y la manipulación para el control de líneas en la entrada y la salida (*handshaking*).

La UART se diseñó para adaptar las señales que maneja la CPU y transmitir las al exterior. La función principal de este circuito es transformar el flujo de información paralela a un flujo de información serie y viceversa.

La UART es totalmente compatible con dispositivos de tecnología TTL (0 Volts y 5 Volts). Estos niveles TTL no son compatibles con el equipo periférico que utilice la interfaz EIA-232, y consecuentemente algún tipo de circuito de interfaz es necesario para la conversión entre voltajes TTL

y EIA-232. Un circuito integrado que convierte el nivel de voltaje de la UART al de la interfaz EIA-232 se denomina convertidor (*driver*), por ejemplo podemos mencionar el dispositivo MC1488 de *Motorola*. Otro dispositivo que se utiliza para recibir (receptor) una señal y convertir un nivel de voltaje de interfaz EIA-232 a un voltaje TTL es el MC1489 de *Motorola*.

Típicamente se utilizan los siguientes modelos del circuito integrado UART: *Intel* 8250 (bastante antiguo, con fallas, sólo llega a 9,600 *bauds*), *Intel* 16450 (versión corregida del 8250, llega hasta 115,200 *bauds*, con 16 *bits*), *Motorola* MC8650, *Motorola* MC68681, *National* INS 8250 y 16550A (con *buffer* de E/S).

En las computadoras a partir de la gama *Pentium*, los circuitos UART de la placa base son de alta velocidad, contienen la UART 16550A.

Las líneas de salida de un circuito UART son: TxD, DTR y RTS; mientras que las líneas de entrada son: RxD, DSR, DCD y CTS.

La figura II.7 muestra la conexión de las líneas de una UART, a través de las compuertas del *driver* y del conector receptor/transmisor (R/T).

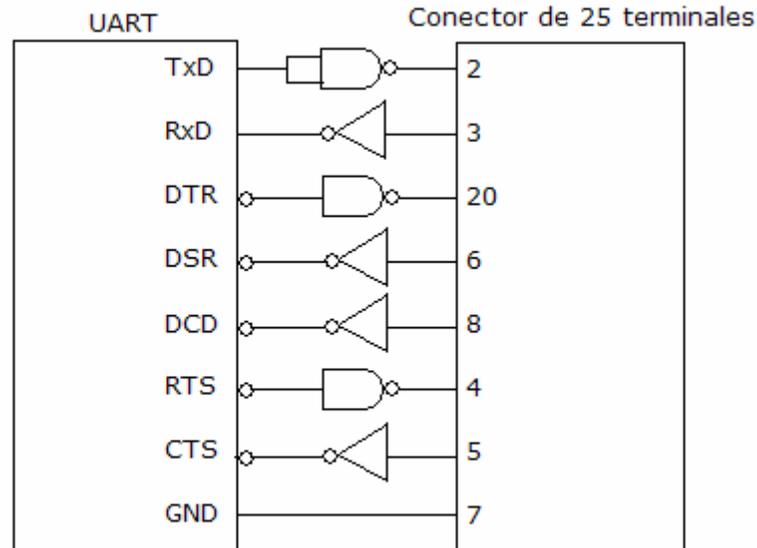


Figura II.7. Conexión de la UART hacia el exterior.

Protocolo de comunicación serial. Antes de iniciar cualquier comunicación con el puerto EIA-232, se debe determinar el protocolo a seguir, dado que el estándar del protocolo no permite indicar en que modo se está trabajando; la persona que utiliza la interfaz es la que

debe decidir y configurar ambas partes antes de iniciar la transmisión de datos.

Los parámetros a configurar son los siguientes:

- Protocolo serie (*bit* de inicio, número de *bits* de datos, paridad y de paro).
- Velocidad del puerto.
- Protocolo para controlar el flujo de datos (RTS/CTS o *XON/XOFF*).

En la transmisión de los datos se pueden enviar datos de 5, 6, 7 u 8 *bits*. Se comienza con un *bit* de inicio (*start*), que corresponde al valor binario 0, le siguen los *bits* para cada dato por transmitir, después un *bit* opcional de paridad que indica si el número de *bits* transmitidos es par o impar para detectar fallas, y por último 1 ó 2 *bits* de parada (*stop*).

Normalmente, el protocolo utilizado es 8N1 (8 *bits* de datos, sin paridad y con 1 *bit* de *stop*).

En la figura II.8 se presenta el diagrama de transmisión de un dato (formato: 1 *bit* de inicio, 7 *bits* por dato, 1 *bit* de paridad y 2 *bits* de paro). La comunicación se establece cuando el transmisor a través de la terminal RTS envía una señal de salida (*OM*) hacia el receptor, indicando que el dato está listo para ser transmitido. Como respuesta el transmisor recibe por medio de la terminal CTS una señal de entrada (*OM*) indicando que el receptor está listo para recibir datos, e inicia la comunicación; el emisor envía un *bit* de *START* (nivel alto), siguen los datos, después un *bit* de paridad y al final dos *bits* de *STOP* (nivel bajo).

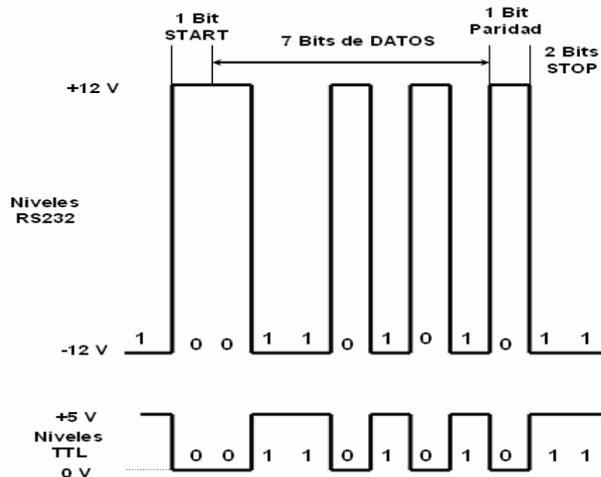


Figura II.8. Protocolo de transmisión serie.

Una vez comenzada la transmisión de un dato, los *bits* llegan uno detrás de otro a una velocidad constante y en determinados instantes. Por eso se dice que el EIA-232 es asíncrono por carácter y síncrono por *bit*.

Velocidad del puerto. Las velocidades de comunicación para este estándar son relativamente bajas, no más de 20 kbps, los valores comúnmente usados son: 300, 1200, 2400, 9600 y 19200 bps. La velocidad se mide en *bits/segundo*. Una velocidad de transmisión normalmente usada es de 9600 *bits/segundo*.

También hay otra velocidad de transmisión o tasa de transmisión que se expresa en pulsos/segundos o *Bauds*. Esto se aclara con un simple ejemplo: si un codificador emite 100 *Bauds*, quiere decir que está emitiendo 100 pulsos/segundo; estos pulsos no son necesariamente binarios. Aclarando: si la señal es binaria, 100 *Bauds* equivalen a 100 *bits/seg*. Si la señal es cuaternaria, cada pulso contiene dos *bits* y entonces 100 *Bauds* equivalen a 200 *bits/seg*. Si la señal es de 8 niveles, cada pulso contiene tres *bits* y entonces 100 *Bauds* equivalen a 300 *bits/seg.*, y así sucesivamente.

Tiempo de flujo de un bit. El tiempo en el que cada *bit* es transmitido dada una velocidad en *Baud*, puede ser determinada utilizando la siguiente formula:

$$\text{Periodo de tiempo de cada bit} = 1/\text{Velocidad.}$$

Por ejemplo: si la velocidad de transmisión es de 9600 *Baud*, entonces el tiempo en el que se transmite cada *bit* es de 104 μ s.

Protocolo para el control del flujo de datos. El control de flujo de datos se puede aplicar por *hardware* o *software*.

El control de flujo de datos por *hardware* se realiza si un dispositivo de los que están conectados a una interfaz EIA-232 procesa los datos a una velocidad menor de la que los recibe, para lograr esto, deben de conectarse las líneas *handshaking*. Las líneas de *handshaking* que permiten hacer este control de flujo, son las terminales RTS y CTS, así como DTR y DSR. Realizada la conexión de las terminales anteriores, se logra que al dispositivo más lento le de tiempo de procesar los datos.

En el caso de control de flujo de datos por *software* se hace uso de la comunicación de dos caracteres ASCII, uno de inicio y otro de paro. Estos caracteres son los siguientes: *X-ON* (definida como \wedge s, Control+S o ASCII 11) y *X-OFF* (definida como \wedge Q, Control+Q o ASCII 13).

Cuando el transmisor recibe un carácter *X-OFF* cesa la comunicación hasta que le sea enviado un carácter *X-ON*. Este tipo de control de flujo es normalmente usado cuando el transmisor y el receptor pueden procesar datos relativamente rápidos. Normalmente, el receptor tendrá un *buffer* muy grande para la introducción de los datos, cuando este *buffer* se ha llenado, se transmite un *X-OFF* hacia el transmisor, a través de un microprocesador en el receptor. Posteriormente, cuando el transmisor recibe un *X-ON*, continúa la transmisión de datos hacia el receptor.

Protocolo de codificación de señales. Las señales necesarias en la comunicación EIA-232 se codifican de acuerdo al formato No Retorno a Nivel Cero (NRZ-L). En donde al nivel del '0' lógico le corresponde un nivel de voltaje alto, al nivel '1' lógico le corresponde un voltaje bajo y la señal nunca está en el nivel de cero Volts (Figura II.9). La palabra nivel (*Level*) indica que únicamente hay un nivel de voltaje para el '0' lógico y otro para el '1' lógico.

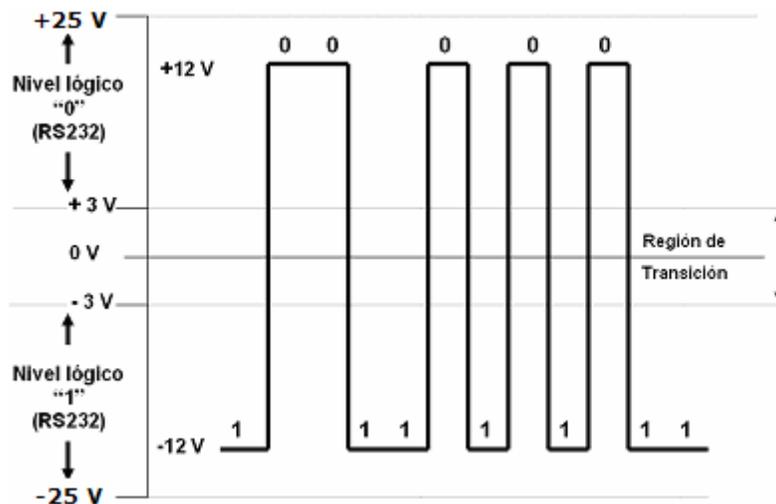


Figura II.9. Codificación NRZ-L y niveles de voltaje EIA-232.

La forma más simple de transmitir señales digitales es mediante la utilización de un nivel diferente de tensión para cada uno de los *bits*. De entre los diferentes formatos de codificación posibles, el más sencillo es el código NRZ. El código NRZ-L es el más utilizado en la práctica y por definición el estándar de comunicación serie EIA-232 asigna un nivel negativo al '1' lógico y un nivel positivo al '0' lógico.

Las ventajas que se tienen de este código son las siguientes:

- No tiene componente directa porque matemáticamente hay que integrar la señal y se puede observar que la integral resulta ser

cero, esto se ve a simple vista porque la probabilidad de que haya 1's y 0's es del 50% respectivamente. Al no tener componente directa, el código es favorable para viajar por un cable de cobre.

- Este código no permite la propagación de errores, es decir, cada *bit* se reconoce por sí mismo, sin ayuda de los demás.
- Es el código idóneo y confiable en la transmisión de datos a bajas frecuencias.

Una desventaja que se puede apreciar en este formato es:

- El código no tiene la capacidad para detectar errores, esto es, si un nivel alto es interpretado por el receptor como nivel bajo o viceversa, el código no permite detectar el error.

Conexiones físicas. Son las principales formas de implementar una conexión entre dos dispositivos para establecer una comunicación serial, un ejemplo de una conexión se da entre una computadora y una línea telefónica a través de un módem, la cual se muestra en la figura II.10.

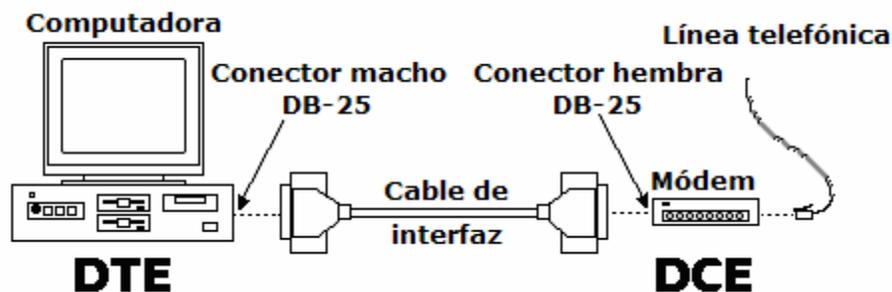


Figura II.10. Conexión física EIA-232.

Los tipos de conexiones más importantes que se realizan entre equipos de comunicación serial son: comunicación simple sin control de flujo de datos, comunicación con control de flujo de datos y comunicación de DTE-DCE.

Comunicación simple sin control de flujo de datos. Este tipo de comunicación funciona con el control *X-OFF* y *X-ON*. Como no se tiene control de flujo con *handshaking*, entonces las terminales se conectan como se muestra en la figura II.11. Para la conexión mostrada, se están considerando dos posibilidades de conexión a través de los conectores EIA-232. Lo anterior significa que un equipo de comunicación tiene un tipo de conector DB-9 y por parte del otro equipo contiene también un conector tipo DB-9. De la misma manera se muestra que un equipo tenga un conector tipo DB-9 y el otro equipo contenga un conector tipo DB-25.

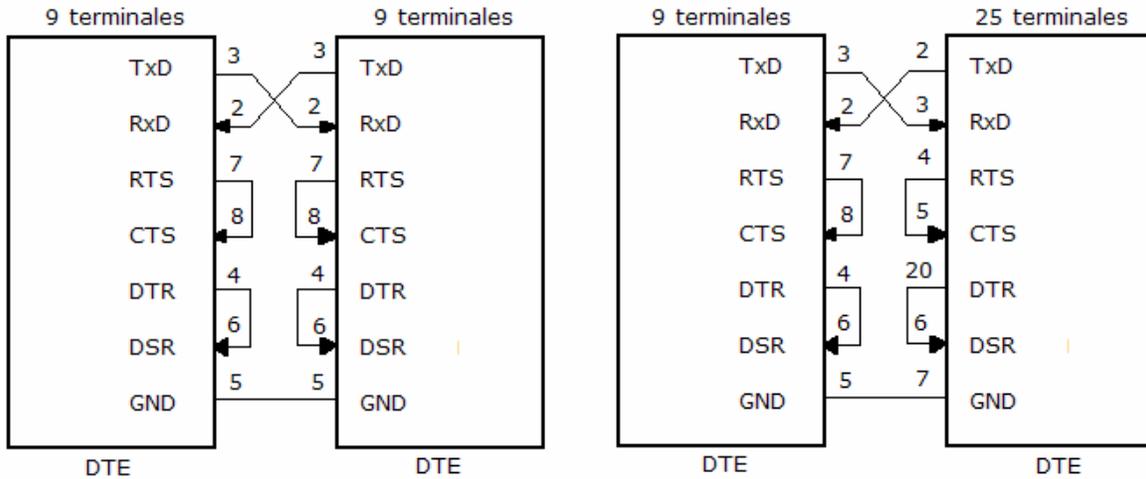


Figura II.11. Conexiones EIA-232 sin control de flujo de datos.

La figura muestra el ejemplo de una conexión entre un DTE (tal como una PC) con otro DTE (una impresora).

Comunicación con control de flujo de datos. Para la comunicación con control de flujo de datos seriales, se utiliza el control de flujo por hardware (terminales *handshaking*). La instalación de las terminales se muestra en la figura II.12.

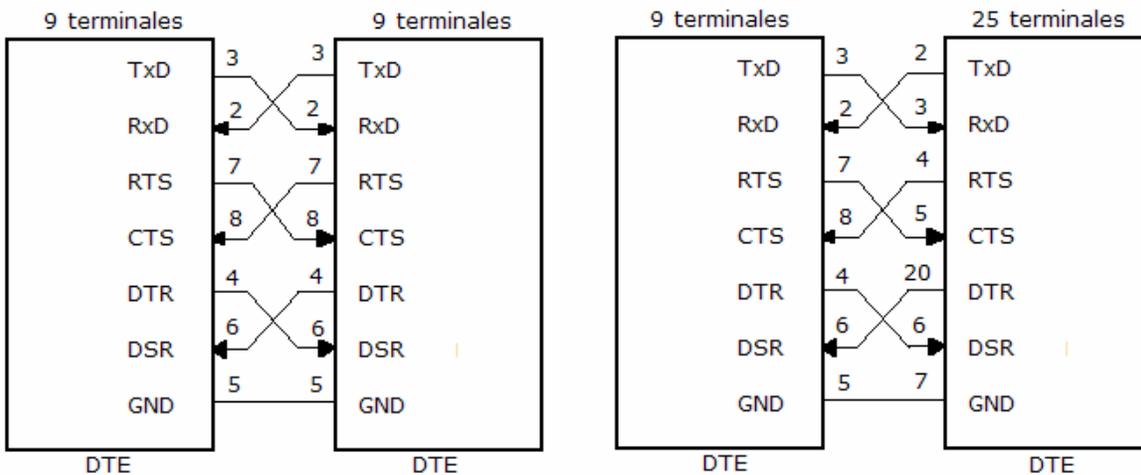


Figura II.12. Conexiones EIA-232 con control de flujo de datos.

De la misma forma se presentan dos posibilidades de conexión, esto es, un equipo que tiene conector DB-9 con otro equipo con conector DB-9 y un equipo con conector DB-9 hacia otro equipo con conector DB-25.

Comunicación DTE-DCE. En la conexión entre un DTE y un DCE, por ejemplo de una PC hacia un módem, se hace la instalación de las terminales de la forma como se muestra en la figura II.13.

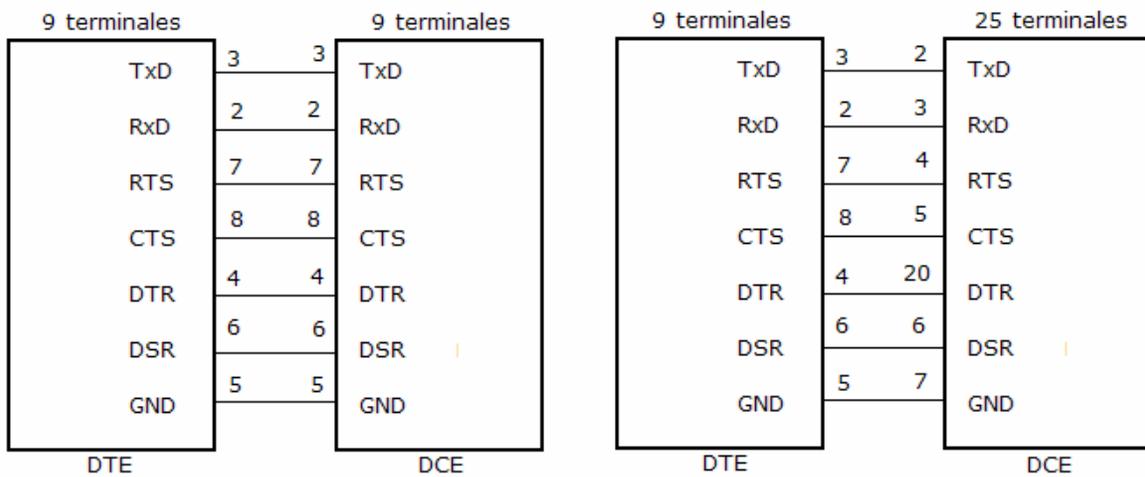


Figura II.13. Conexiones EIA-232 de DTE a DCE.

En este tipo de conexión, la velocidad de transmisión de datos del DTE es la velocidad de operación que hay entre la computadora y el módem al cual está conectado. Existe la velocidad de transmisión del DCE, la cual es aquella entre el módem y el otro equipo al cual está transfiriendo la comunicación.

Como se ha visto en las tres conexiones anteriores, son las más usuales e importantes dentro de este estándar de comunicación, no todas las terminales se utilizan puesto que es suficiente con las que se muestran en las figuras.

II.3.2 Estándar USB

En el campo de las comunicaciones, una interfaz es el camino (puerto o circuito físico) a través del cual se envían o reciben señales desde un sistema o subsistemas hacia otros.

No existe una interfaz universal, sino que existen diferentes estándares, por ejemplo: interfaz ISA (Arquitectura Estándar de la Industria), interfaz PCI (Interconexión de Componentes Periféricos), interfaz EISA (Arquitectura Estándar de la Industria Extendida), etc.; las cuales establecen especificaciones concretas, donde la interconexión sólo es posible utilizando la misma interfaz en origen y destino a través de un protocolo común a ambos.

Las tarjetas de expansión utilizadas en las computadoras personales son tarjetas que se conectan a través de un *bus* interno de expansión, esto se hace con el fin de mejorar el desempeño de un sistema de cómputo. Estas tarjetas mejoran las posibilidades de la computadora, agregando por este medio: memorias, video, controladores de discos flexibles, impresoras, puertos para módem, comunicaciones seriales, etc.

Con la intención de ya no hacer las conexiones internamente, sino que realizarlas desde el exterior, sin abrir la computadora, se desarrolló el *Bus Serial Universal (USB)*.

Definición

USB es una interfaz que proporciona un estándar de *bus* serie para conectar dispositivos a una computadora.

Objetivos de USB

Las empresas que desarrollaron este proyecto tuvieron como objetivos los siguientes puntos:

- Eliminar la necesidad de adquirir tarjetas separadas para colocar en los puertos ISA, EISA o PCI de las computadoras.
- Permitir a los dispositivos ser conectados o desconectados (*plug and play*) a la computadora sin necesidad de reiniciar el sistema operativo.
- Los usuarios no tienen que abrir el gabinete para instalar nuevos dispositivos de E/S.
- Sólo debe haber un tipo de cable para la conexión de dispositivos.
- Los dispositivos de E/S tienen que obtener su energía mediante el cable de conexión.
- Conexión máxima de 127 dispositivos a una sola computadora.
- El sistema deberá apoyar dispositivos de tiempo real.
- Los dispositivos podrán instalarse mientras la computadora esté funcionando.
- El nuevo *bus* debe ser de bajo costo.

Historia

En 1980 la empresa Máquinas de Negocios Internacional (IBM) fabricó su primera computadora personal (PC), la cual contenía un *bus* de expansión conocido como XT (*PC bus*), que funcionaba con las características de los procesadores Intel 8086 y 8088 (4.77 MHz, con un *bus* de datos de 8 *bits* y un *bus* de dirección de 20 *bits*). Con el

desarrollo de los microprocesadores el *bus* de la PC no garantizaba que la comunicación fuera fluida hacia el exterior, esto era debido a que aumentaban sus características: velocidad, *bus* de datos, *bus* de dirección, etc.

Con la introducción de Tecnología Avanzada (AT) apareció el nuevo *bus* ISA en el mundo de la PC, en el cual el *bus* de datos era de 16 *bits*, además era compatible con su antecesor. Las anteriores tarjetas de 8 *bits* podían por tanto manejarse con las nuevas tarjetas de 16 *bits* en un mismo dispositivo. De todas maneras las tarjetas de 16 *bits* eran considerablemente más rápidas, ya que transferían la misma cantidad de datos en comparación con las tarjetas de 8 *bits* en la mitad de tiempo. Además, en el *bus* ISA, el *bus* de direcciones se amplió hasta 24 *bits*, de manera que se podía dirigir al AT con memoria de 16 MB. También se incremento la frecuencia, de 4.77 MHz se pasó a 8.33 MHz.

El *bus* ISA de IBM, tenía varias limitantes, por lo que se trabajó en una nueva tecnología de *bus* que aplicó a las computadoras PS/2. El estudio dio como resultado el *bus* síncrono llamado Arquitectura de Interfaz de Microcanal (MCA), que poseía un *bus* de datos de 32 *bits* y una velocidad de reloj más elevada de 10 MHz, con una velocidad de transferencia máxima de 20 Mbps frente a los 8 Mbps del *bus* ISA. El *bus* MCA era totalmente incompatible con las tarjetas de expansión que incluía el *bus* ISA.

En la búsqueda de otro *bus*, se originó el *bus* EISA con el objetivo de competir con el *bus* MCA, también estaba basado en la idea de controlar el *bus* desde el microprocesador y ampliar el *bus* de datos hasta 32 *bits*. EISA mantuvo compatibilidad con las tarjetas de expansión ISA ya existentes, por lo cual se obligo a funcionar a una velocidad de 8 MHz (exactamente 8.33 MHz).

Su mayor ventaja con respecto al *bus* MCA es que EISA era un sistema abierto, puesto que fue desarrollado por la mayoría de fabricantes de computadoras compatibles PC que no aceptaron el monopolio que intentó ejercer IBM. Estos fabricantes fueron: *AST, Compaq, Epson, Hewlett Packard, NEC, Olivetti, Tandy, Wyse* y *Zenith*.

Intel presentó en 1992 un *bus* moderno, síncrono e independiente de la frecuencia de reloj, éste es el *bus* PCI. El *bus* PCI es independiente de la CPU, puesto que entre la CPU y el *bus* PCI se instala siempre un controlador del *bus* PCI. También es posible instalarlo en sistemas que no estén basados en el procesador *Intel* (Pentium).

El *bus* PCI transfiere datos usando el reloj del sistema, puede operar a 32 *bits* o a 64 *bits* de datos. Su frecuencia de reloj es de 33 MHz o de 50 MHz. Las tarjetas de expansión PCI trabajan eficientemente en todos los sistemas y pueden ser intercambiadas de la manera que se desee. Si se instalara una CPU más rápida en la computadora, no hay problema porque con la separación del *bus* PCI de la CPU no influye la frecuencia de reloj.

Resumiendo: para el manejo de dispositivos externos desde una PC, se hacía uso de las ranuras internas de ésta, conocidas como *bus* ISA, *bus* PCI, etc. Resultaba engorroso el proceso de incorporación de un dispositivo externo, puesto que había que abrir la computadora para realizar la conexión y en algunas ocasiones había que hacer ciertos ajustes. Esto llevó a pensar en la posibilidad de crear una tecnología que permitiera llevar a cabo la incorporación de un dispositivo externo desde afuera, es decir, sin abrir la computadora. Este pensamiento desencadenaría en la creación de un nuevo *bus* de interfaz.

En la década de los 90's se reunieron representantes de siete compañías (*Microsoft, Compaq, Digital Equipment Corporation, IBM, Intel, NEC y Northern Telecom*) para diseñar y desarrollar una mejor manera de conectar dispositivos de E/S de baja velocidad a una computadora. El resultado fue el nacimiento del estándar USB.

La evolución cronológica de USB está dada de la siguiente manera:

- USB 0.7: noviembre de 1994.
- USB 0.8: diciembre de 1994.
- USB 0.9: abril de 1995.
- USB 0.99: agosto de 1995.
- USB 1.0: se anuncia en noviembre de 1995.
- USB 1.0: enero de 1996. Con las siguientes especificaciones de velocidad de transmisión de datos: baja velocidad de 1.5 Mbps y la velocidad máxima, denominada velocidad completa de 12 Mbps.
- USB 1.1: septiembre de 1998.
- USB 2.0: abril de 2000. Con las siguientes especificaciones: alta velocidad de transmisión de datos de 480 Mbps y es compatible con la versión anterior.
- USB 2.0: revisión en diciembre de 2002. Agrega corrección de especificaciones.
- USB-*On-The-Go* Suplemento 1.0: diciembre de 2001. Es una extensión de USB, permite a un puerto actuar como servidor o como cliente, esto se determina por el tipo de conexión que se

realice. *USB-On-The-Go* también ha diseñado 2 tipos de conectores pequeños, el mini-A y el mini-B.

- *USB-On-The-Go* Suplemento 1.0a: junio de 2003.
- *USB-On-The-Go* Suplemento 1.2: abril de 2006.
- Micro USB: anunciada en enero de 2007.

II.3.2.1. Especificaciones del estándar

Las especificaciones del estándar de comunicación USB se van a presentar de la siguiente manera: aspectos mecánicos, aspectos eléctricos y aspectos funcionales.

Aspectos mecánicos

Los aspectos mecánicos se refieren a todo lo relacionado con lo físico, es decir: los conectores, tipos de conectores, las terminales, definición de las terminales, etc.

Especificaciones físicas. El estándar USB utiliza un tipo de cable pequeño, sencillo, económico y fácil de instalar. Únicamente existe un tipo de cable, pero con dos tipos de conectores diferentes (Figura II.14) en los extremos: tipo A y tipo B. La conexión de estos dos tipos de conectores hace imposible la colocación de manera errónea. El conector tipo A es utilizado hacia el sistema *host* y el tipo B es utilizado hacia el dispositivo conectado USB.

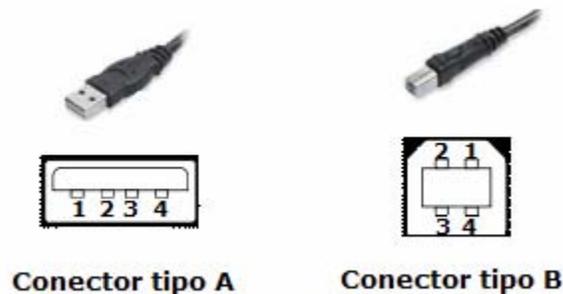


Figura II.14. Tipos de conectores de bus USB.

El cable tipo A presenta las cuatro terminales correspondientes a los cuatro hilos alineados en un plano, mientras que los conectores tipo B presentan las terminales distribuidas en dos planos paralelos, dos en cada plano.

En la tabla II.2 se muestra la definición de las terminales del conector USB.

Número de terminal	Color de cable	Nombre	Descripción
1	Rojo	BUS	(4.75-5.25) VDC
2	Blanco	D-	Dato-
3	Verde	D+	Dato+
4	Negro	GND	Tierra

Tabla II.2. Especificación de terminales en conector USB.

El cable USB consta de cuatro conductores, dos van trenzados y son para transmitir una señal diferencial (D+ y D-), uno para alimentación de +5 Volts y uno para tierra, estos dos últimos hilos están rodeados de una capa de blindaje para evitar interferencias.

En una conexión USB la longitud del cable no debe exceder de 5 metros, esto es debido a la degradación que sufriría el nivel de voltaje utilizado, aunque es posible encadenar 4 *hubs* usando 5 cables periféricos, con lo que se pueden lograr distancias de comunicación de hasta 25 metros.

El cable blindado que se utiliza es con las siguientes especificaciones:

Para datos: 28 AWG trenzado.

Para energía: de 28 AWG a 20 AWG no trenzado.

Si es cable no blindado el que se usa, las especificaciones son:

Para datos: 28 AWG no trenzado.

Para energía: de 28 AWG a 20 AWG no trenzado.

En la tabla II.3 se observa la comparación del número de cable con la distancia a la que puede transmitir.

Tamaño de cable (AWG)	28	26	24	22	20
Máxima longitud	0.81 m	1.31 m	2.08 m	3.33 m	5.0 m

Tabla II.3. Distancia de transmisión para cables.

Aspecto eléctrico

Este aspecto especifica las características eléctricas que debe cumplir el estándar USB.

Especificaciones eléctricas. El voltaje de operación del estándar USB es de 5 Volts.

Como se ha escrito anteriormente USB usa un par de cables trenzados para transmitir señales diferenciales (D+ y D-), estas señales utilizan el formato de codificación No Regresa a Cero Invertida (NRZ-I). Si se está transmitiendo en baja velocidad o velocidad completa, se tiene lo siguiente: un '1' diferencial es transmitido en D+, con un voltaje mayor de 2.8 Volts, con una resistencia de 1.5 k Ω conectada a tierra y D- debajo de 0.3 Volts, con una resistencia de 1.5 k Ω conectada a 3.6 Volts. Por otro lado un '0' diferencial es transmitido en D- con un voltaje mayor de 2.8 Volts y un D+ menor de 0.3 Volts, con las mismas resistencias conectadas.

El receptor define un '1' diferencial en D+ con 200 mV mayor que D- y un '0' diferencial en D+ con 200 mV menor que D-. Los términos que se utilizan como estados J y K definen los niveles lógicos '0' diferencial y '1' diferencial respectivamente.

Los dispositivos disponen de un transmisor diferencial, receptor diferencial y resistencias de terminación con los que pueden transmitir y detectar varios estados eléctricos distintos en la línea, los cuales son:

- Transmisión/recepción diferencial de *bits*: estados J y K.
- Finalizado simple 0 (SE0), ambas señales D+ y D- a 0 Volts. Se utiliza para detectar la conexión/desconexión de dispositivos, para indicar el fin de paquete (EOP) y para generar *reset*.
- Reposo (IDLE): línea en alta impedancia, necesario para permitir transferencia de datos, detectar conexión/desconexión de dispositivos y selección entre dispositivos de baja velocidad o de velocidad completa. El principio de paquete (SOP) se indica mediante una transición IDLE a K. El EOP se indica mediante una secuencia SE0(2 *bits*)+J(1 *bit*)+ IDLE.
- Detección de dispositivo y selección de baja velocidad y velocidad completa: cuando el transmisor deja la línea en IDLE, si hay un dispositivo conectado, su polarización fuerza a un estado J, si es de baja velocidad, o K, si es de velocidad completa, y si no lo hay, la polarización del transmisor fuerza a un estado SE0.

Aspecto funcional

El aspecto funcional trata la explicación de todo lo relacionado con el funcionamiento del estándar USB. Se describen las características de: el funcionamiento del *bus* USB, la transferencia de datos, el formato de codificación de la información y la velocidad de transmisión de los datos.

Funcionamiento del bus USB. Un *bus* USB emplea una estructura jerárquica en estrella (Figura II.15), conformada por capas apiladas, que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas está el controlador anfitrión (*host*), que controla la transmisión de datos que circulan por el *bus*. Esta estructura permite a muchos dispositivos conectarse a un único *bus* lógico, sin que los dispositivos que se encuentran más abajo en la pirámide se vean afectados.

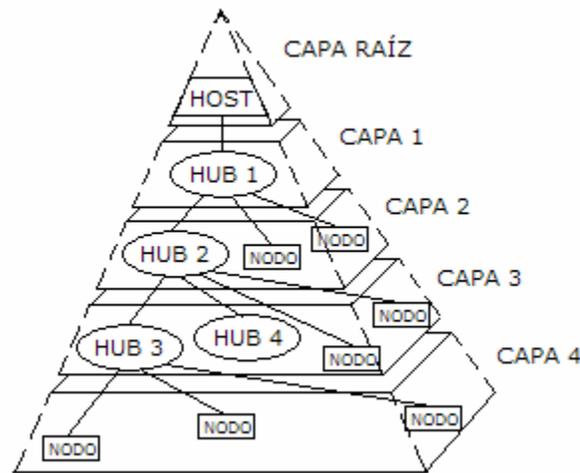


Figura II.15. Estructura de capas del bus USB.

USB trabaja como una interfaz en la transmisión de información, así como para la distribución de energía.

El sistema USB consta de tres componentes:

- Controlador
- Concentradores
- Periféricos

Controlador (host). El controlador de comunicación de datos, mediante el *bus* USB, se encuentra en el interior de la computadora y es el que se encarga de controlar el flujo de datos entre los periféricos USB y la CPU. Detecta una conexión o desconexión de un periférico, así como la admisión de éste en el *bus*. Cuando un periférico es admitido, el controlador determina su tipo y le asigna una dirección lógica, para utilizarla siempre en la comunicación de ambos. Si se presentan errores durante la conexión, el controlador lo comunica a la CPU y ésta lo transmite al usuario. Establecida la conexión correctamente, el controlador asigna al periférico los recursos del sistema que necesita para su funcionamiento.

Concentradores (hubs). Los concentradores son distribuidores inteligentes de datos, así como alimentadores de voltaje, y hacen posible la conexión a un único puerto USB de 127 dispositivos. Se encargan de repartir selectivamente los datos y alimentación hacia sus puertos descendentes y permiten la comunicación hacia su puerto de retorno o ascendente. Por ejemplo, un *hub* de 7 puertos (Figura II.16) acepta datos de la PC hacia un periférico por su puerto de retorno o ascendente y los distribuye a los 7 puertos descendentes si fuera necesario. Los concentradores también permiten las comunicaciones desde los dispositivos periféricos hacia la PC, aceptando datos en los 7 puertos descendentes y enviándolos hacia la PC por el puerto de retorno.

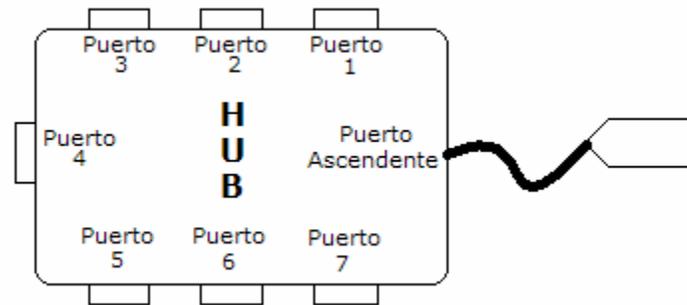


Figura II.16. Concentrador o hub de 7 puertos.

Si la PC tiene un único puerto USB y se le conecta un concentrador de 4 puertos, la PC se queda sin más puertos disponibles. Sin embargo, el *hub* de 4 puertos permite realizar 4 conexiones descendentes. Conectando otro *hub* de 4 puertos a uno de los 4 puertos del primero, habrá creado un total de 7 puertos a partir de un puerto de la PC. De esta forma se agregan concentradores.

Periféricos. El estándar USB soporta periféricos de diferentes velocidades (Figura II.17). Empleando dos velocidades para la transmisión de datos, 1.5 Mbps y 12 Mbps, se consigue una utilización más eficiente de sus recursos.

Los periféricos que funcionan con baja velocidad (1.5 Mbps) son: teclados, ratones, *joysticks*, y otros periféricos para juegos.

Los periféricos que utilizan velocidades más altas (12 Mbps) son: monitores, impresoras, módems, *scanner*, equipos de audio, etc.

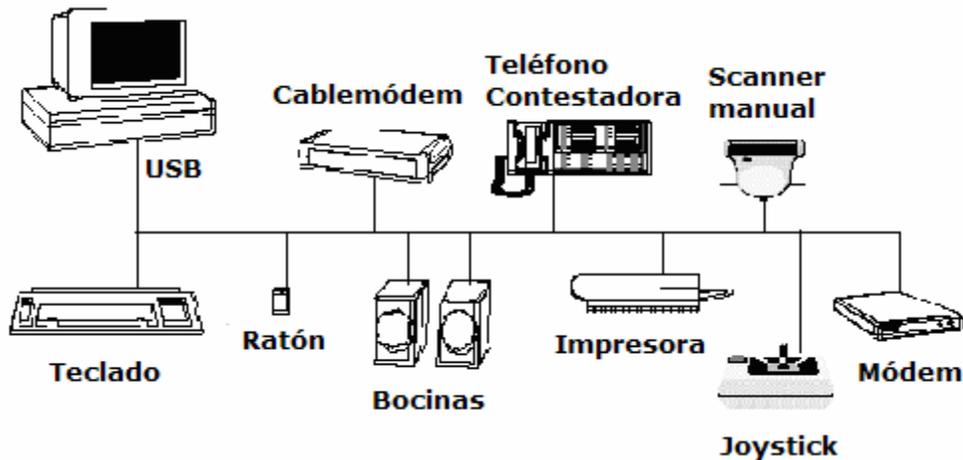


Figura II.17. Periféricos conectados por USB.

Transferencia de datos. Los dispositivos conectados al *bus* USB, así como los *hubs*, tienen asociados canales lógicos. Estos canales son sinónimo de transmisión de datos, están conectados desde el controlador *host* hasta una entidad lógica en el periférico, nombrado punto final. El término punto final es utilizado para referirse a la entrada del dispositivo instalado.

Los puntos finales se encuentran en ambas direcciones pero actúan hacia una sola dirección a la vez, se direccionan por un número y por el sentido en que se transfiere la información, es decir, son de entrada si transmiten datos hacia el sistema y de salida si transfieren datos hacia el dispositivo conectado. Los puntos finales también pueden ser vistos como la interfaz entre el *hardware* del dispositivo conectado y el *software* de aplicación que está corriendo en el dispositivo.

Estos puntos finales y sus respectivos canales son numerados del 0 al 15 en cada dirección, así que un dispositivo conectado al *host* puede tener hasta 32 canales activos, 16 dentro del controlador *host* y 16 en el dispositivo conectado.

El punto final 0 es reservado para administrar el *bus* en ambas direcciones, así que ocupa 2 de los 32 puntos finales (todos los dispositivos USB implementan un punto final 0), por lo tanto, siempre hay un canal lógico numerado 0 interno *host* y un externo en cualquier dispositivo dado.

En estos canales, los datos son transmitidos en paquetes de longitud variable, típicamente 2^n bytes, así que un paquete USB contendrá 8, 16,... hasta 512 bytes.

El estándar USB reconoce los canales de comunicación por el tipo de transferencia de datos, estos se clasifican en cuatro categorías diferentes:

- *Transferencia de control.* Envían comandos cortos y simples a un dispositivo, por ejemplo: configurar dispositivos, preguntar por su estado y dar órdenes.
- *Transferencia isócrona.* En algunas velocidades, la transmisión se garantiza pero con posible pérdida de datos hacia dispositivos de tiempo real, por ejemplo: audio y video.
- *Transferencia de interrupción.* Utilizadas para dispositivos que necesitan una respuesta rápida garantizada, por ejemplo un teclado.
- *Transferencia de volumen.* Son para transferencias esporádicas grandes de datos, utilizando el restante del ancho de banda disponible, por ejemplo: transferencia de archivos e impresoras.

Formato de codificación de datos. El estándar USB utiliza la codificación de datos NRZ-I. En este formato, el nivel de voltaje se mantiene constante en la duración de un *bit*. La codificación de datos se presenta mediante la presencia o ausencia de un cambio en el nivel de la señal al principio del intervalo de duración de un *bit* (Figura II.18).

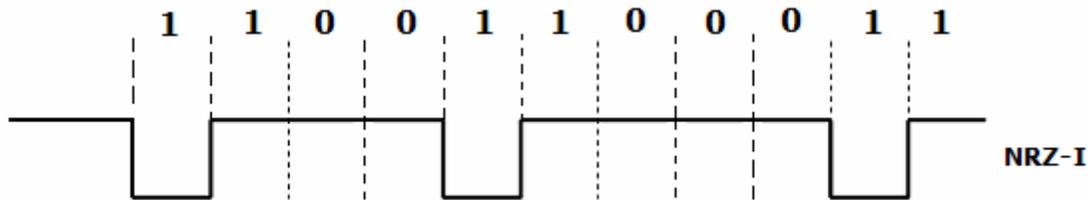


Figura II.18. Código NRZ-I.

En este formato, también conocido como NRZ marca, la información no va en el nivel de los pulsos, sino en los cambios de nivel, esto es: si se va a transmitir un '1' lógico, la señal cambia de nivel, y si se va a transmitir un '0' lógico, la señal no cambia de nivel.

Las ventajas que ofrece este formato son:

- Teniendo en cuenta que la información está en los cambios de nivel, el mensaje no se pierde aunque la señal se invierta.
- Este formato es adecuado cuando se tiene la certeza de que el mensaje binario tiene muchos más 1's que 0's, lo cual va a originar muchos cambios de nivel.

Las desventajas de usar este código, son las siguientes:

- Hay propagación de errores porque: si debido a la deformación de la señal, el receptor interpreta mal un nivel, el siguiente *bit* también será interpretado incorrectamente.
- Contiene una componente de corriente directa.

Una vez que se han revisado las ventajas y desventajas se puede decir que: el formato de codificación NRZ-I es confiable para utilizarlo en este estándar por la cantidad de cambios de nivel, ya que esto garantiza la confiabilidad en la información.

Velocidad de transmisión de la información. USB especifica la velocidad de transmisión de datos de la siguiente manera:

- *Alta velocidad de transmisión:* 480 Mbps (en USB 2.0 ó más).
- *Velocidad completa de transmisión:* 12 Mbps (requiere cable blindado).
- *Baja velocidad de transmisión:* 1.5 Mbps (cable sin blindar).

II.4. Comunicaciones inalámbricas

El crecimiento de la electrónica, así como el de las comunicaciones, ha permitido a la industria reemplazar el tipo de conexión con alambres por sistemas de interconexión inalámbricos, en búsqueda de verdadera flexibilidad y comodidad para el usuario final. De tal manera que en años recientes, la industria de las comunicaciones inalámbricas ha percibido un crecimiento extraordinario.

Las comunicaciones inalámbricas de largo alcance hacen uso de las radiofrecuencias (RF), y generalmente utilizan partes del espectro de RF con licencia. Las comunicaciones inalámbricas de corto alcance pueden utilizar ya sea RF o infrarrojos y típicamente utilizan partes del espectro de frecuencias sin licencia (libres).

En el presente trabajo hacemos uso de la comunicación inalámbrica de corto alcance. Existen varios estándares inalámbricos de corto alcance, pero los tres más importantes son:

- Asociación de Datos Infrarrojos (IrDA).
- Tecnología *Bluetooth* inalámbrica.
- Redes de Área Local Inalámbrica (WLAN). Especificado y conocido como IEEE 802.11, el cual contiene tres principales variantes:

802.11b y 802.11g, que operan en 2.4 GHz, y 802.11a, que opera en 5 GHz.

La tabla II.4 presenta las principales características de las tecnologías mencionadas.

Función	IrDA	WLAN	Bluetooth
Tipo de Conexión	Infrarrojo Haz estrecho Línea de mira	Espectro amplio esférico	Espectro amplio esférico
Espectro	Óptico 850-900 nm.	RF 2.4 GHz. (5 GHz para 802.11a)	RF 2.4 GHz
Potencia	40-500 mW/Sr	100 mW	10-100 mW
Velocidad de Transmisión	De 9600 bps a 16 Mbps (con variaciones)	11 Mbps 54 Mbps para 802.11a y 802.11g	1 Mbps
Alcance	1 m	100 m	10-100 m
Dispositivos Soportados	2	Conectados a través de un punto de acceso	8 (activo) 200 (pasivo)
Canales de voz	No soporta	VoIP	3
Direccionamiento	32 bits	48 bits	48 bits

Tabla II.4. Características de estándares IrDA, WLAN y Bluetooth.

En el presente estudio únicamente se van a describir los estándares con tecnología inalámbrica IrDA y *Bluetooth*.

II.4.1. Estándar IrDA

Como se ha descrito anteriormente, la tendencia a conectar dispositivos con ausencia de cables provoca una exagerada comodidad, pero también una competencia entre las diferentes tecnologías para hacer prevalecer sus técnicas. En este escenario IrDA, como una tecnología de comunicaciones, compite con sus homólogas con grandes ventajas, no obstante también presenta desventajas.

Definición

IrDA conforma un conjunto de especificaciones que constituyen el estándar internacional para el desarrollo de sistemas de comunicaciones a través de rayos infrarrojos. Es una interfaz de comunicación inalámbrica entre dispositivos.

Objetivos

Desarrollar una interfaz inalámbrica, utilizando para ello la luz infrarroja como medio de comunicación.

Historia

IrDA fue establecida en 1993, está patrocinada por más de 160 industrias con el objetivo de crear las especificaciones y estándares para los equipos, así como los protocolos empleados en enlaces por medio de luz infrarroja.

En 1995 comenzaron a aparecer en el mercado equipos portátiles con puertos IrDA integrados. *Microsoft* presentó el soporte IrDA para su sistema operativo *Windows 95*.

II.4.1.1. Tecnología de IrDA

Los estándares de IrDA definen comunicaciones bidireccionales punto a punto empleando un haz de luz infrarroja (en la banda de 850-900 nm) que requiere línea de vista, un ángulo no mayor de 30° y una distancia que no supera un metro (Figura II.19), para obtener velocidades de transmisión de datos entre 9600 bps y 16 Mbps dependiendo del entorno en que se ubiquen los dispositivos.

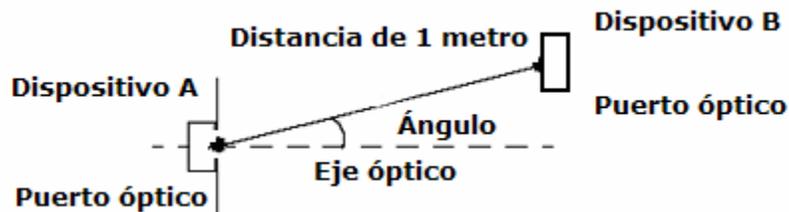


Figura II.19. Estándar de comunicación por infrarrojos.

El estándar IrDA se clasifica por su funcionamiento en dos partes:

- *IrDA-Datos*. Empleado básicamente para transferencias bidireccionales de información en forma inalámbrica y con altas tasas de transmisión entre dispositivos de comunicación.
- *IrDA-Control*. Fue establecido para realizar comunicaciones de control entre dispositivos periféricos como teclados, ratones, *joysticks* o controles remotos. Esta parte no se describirá en el presente trabajo.

II.4.1.2. Especificaciones del estándar

La transmisión de datos a través de luz infrarroja necesita de algún dispositivo (*hardware*) y el uso de un protocolo de comunicación especializado. Este protocolo y el *hardware* requerido son descritos en detalle por la especificación del estándar IrDA.

Similar al modelo OSI (Apéndice A), la tecnología IrDA se encuentra estructurada en bloques funcionales con responsabilidades específicas.

La arquitectura IrDA (Figura II.20) define protocolos fundamentales, que son necesarios en todas las implementaciones, así como protocolos opcionales que se incluyen sólo en algunas implementaciones dependiendo del tipo de aplicación.

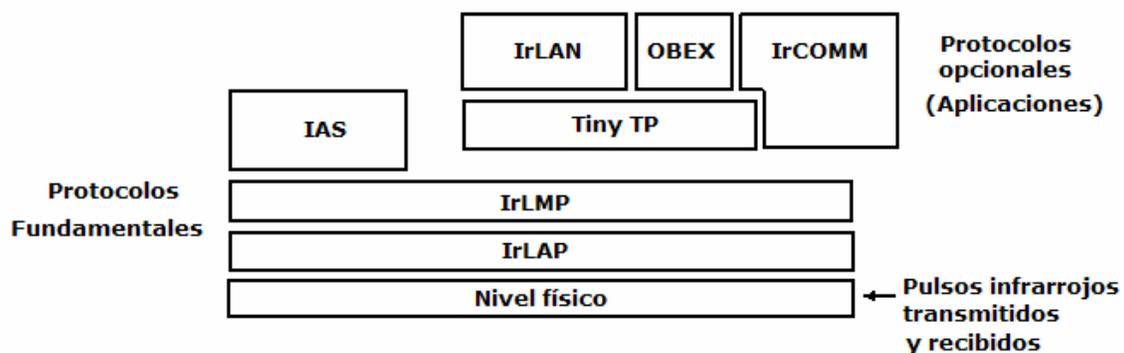


Figura II.20. Arquitectura de Protocolos de IrDA.

Nivel Físico

Es en el nivel físico, o capa física, en donde se realiza la conversión de datos seriales en pulsos de luz infrarroja y viceversa. El problema principal que tiene el transceptor IrDA es que no puede transmitir ni recibir datos al mismo tiempo, es decir, su especificación de comunicación es *half duplex*.

Una capa física se puede implementar completamente en *hardware*, con dispositivos especiales que cumplan con las especificaciones de IrDA: una UART, un modulador/demodulador y un transceptor.

El entramador (*framer*) es un *software* que realiza la presentación de la información recibida por el puerto infrarrojo a la capa superior en un formato específico, así como la construcción de tramas (paquetes de datos) con la información de la capa superior para posteriormente transmitir las hacia su destino.

La capa física es la encargada de enviar y recibir cadenas de *bits* a través del aire, esta capa se relaciona con la velocidad de transmisión de datos y la especificación para codificar la información.

Velocidad de transmisión de la información. La velocidad de comunicación de datos en el estándar IrDA se clasifica en:

- Infrarrojo Bajo (*SIR*): hasta 115 kbps.
- Infrarrojo Medio (*MIR*): 1.152 Mbps.
- Infrarrojo Rápido (*FIR*): 4 Mbps.
- Infrarrojo Muy Rápido (*VFIR*): pretende alcanzar 16 Mbps.

Codificación de datos. Para las tasas de transmisión de datos *SIR* se usa la modulación Regreso a Cero Invertido (RZI), donde un '0' lógico se representa por un pulso de luz cuya duración está normalmente entre $1.6 \mu\text{s}$ y $3/16$ de la duración de un *bit*, y un '1' lógico se codifica con la ausencia de pulso, como se muestra en la figura II.21.

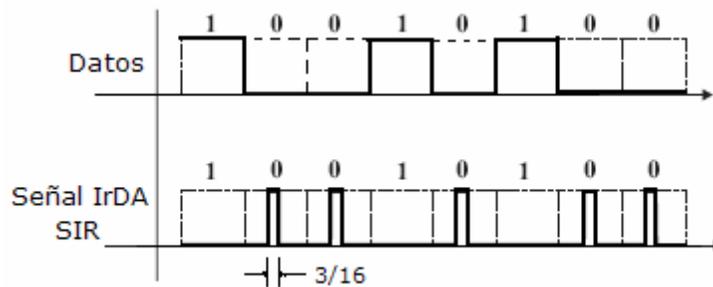


Figura II.21. Modulación para *SIR*.

Para la tasa de transmisión *MIR* se utiliza el mismo formato de codificación de *SIR*, sólo que el pulso óptico tiene una duración nominal de $1/4$ de la duración del *bit*, como se muestra en la figura II.22.

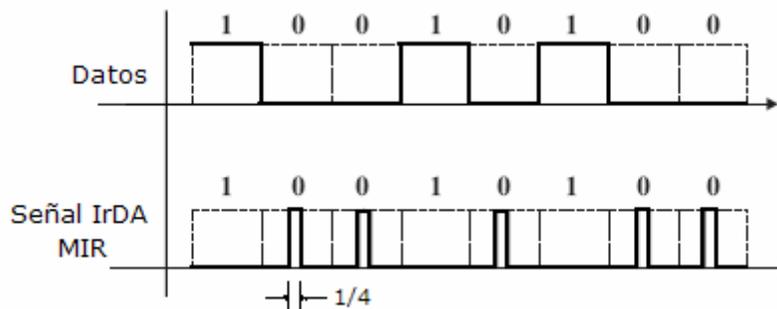


Figura II.22. Modulación para *MIR*.

Para la tasa de transmisión FIR, se utiliza el formato de codificación Modulación por Posición de Pulso (4PPM), donde se toma cada par de *bits* del flujo de datos y se representa con un pulso de luz infrarroja emitido en una de las cuatro posiciones que componen un símbolo 4PPM (Figura II.23).

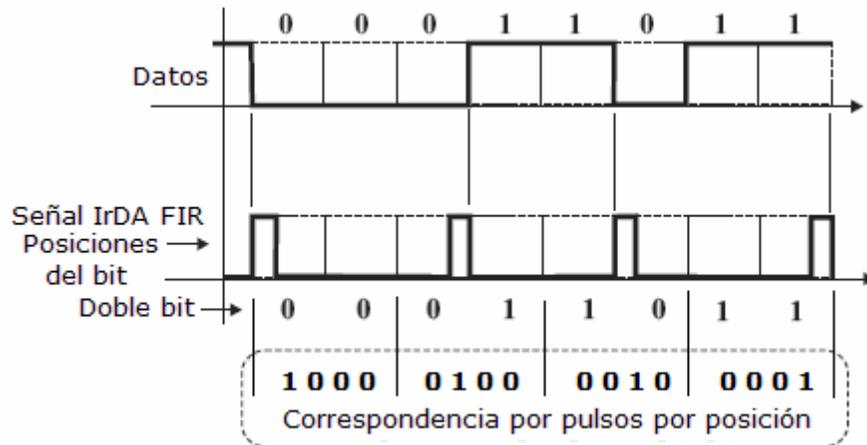


Figura II.23. Modulación 4PPM para FIR.

Un *bit* tiene una duración de 250 ns, por lo que el pulso infrarrojo emitido que representa un '1' lógico en cierta posición es de 125 ns.

Nivel IrLAP

El Protocolo de Acceso al Enlace IrDA (IrLAP), es el nivel que se ubica arriba de la capa física y desempeña actividades relacionadas con las especificaciones de comunicación.

Dentro de las actividades desarrolladas en el nivel IrLAP se encuentran los procesos de control de flujo de información de bajo nivel y los tipos de operación en los dispositivos, además de proporcionar la estructura de los paquetes de información que se transmiten.

Funcionamiento del nivel IrLAP. Cuando se desea establecer la comunicación entre dos dispositivos, en el nivel IrLAP de ambos dispositivos, ésta se realiza a través de primitivas de servicio.

Una operación se inicia con una petición de servicio desde la estación primaria, esta petición viaja en una trama a través del enlace, posteriormente el dispositivo secundario genera una respuesta y la regresa para convertirse en el extremo inicial en una confirmación.

Los servicios más importantes de este nivel se listan a continuación:

- *Descubrimiento de vecinos*: inspecciona el entorno cercano buscando señales infrarrojas para identificar dispositivos y determinar el estado en que se encuentran.
- *Conexión*: selecciona un dispositivo específico para establecer las condiciones de comunicación para ambos extremos.
- *Envío de datos*: es la principal función de este nivel, recibir y transmitir datos.
- *Desconexión*: la comunicación actual se cierra y regresa al modo de desconexión normal, para preparar una futura conexión en el momento que sea necesario.

Control de flujo de información. IrLAP está basado principalmente sobre los protocolos Control de Enlace de Datos de Alto Nivel (HDLC) y Control de Enlace de Datos Síncronos (SDLC), sólo que se realizaron algunas adaptaciones en las características que se requieren para transmisiones por infrarrojos y sus factores del entorno.

IBM realizó las especificaciones del protocolo SDLC para que sus productos trabajaran bajo ese esquema. Posteriormente ISO en respuesta propuso nuevas especificaciones, dando como resultado el protocolo HDLC.

Las características principales del protocolo SDLC son:

- Es un protocolo para enlaces síncronos.
- Está basado en un entorno centralizado (por sondeo).
- La comunicación entre dispositivos es realizada punto a punto y punto a multipunto.
- Utiliza una estrategia de envío de datos continuos.
- Orientados a *bit*, es decir, utilizan un *bit* de control.
- El protocolo tiene tres etapas de comunicación, en las cuales se trata el control de flujo de datos, el control de errores, permite un sondeo de terminales y la utilización de un formato general de paquete de datos.

Las características más importantes del protocolo HDLC:

- Es un protocolo de enlace síncrono.
- La comunicación entre dispositivos se puede realizar punto a punto y punto a multipunto.
- Define tres tipos de estaciones: primaria, secundaria y combinada.

- Dos configuraciones de enlace: configuración no balanceada, formada por una estación primaria y una o más secundarias; configuración balanceada, tiene dos estaciones combinadas.
- Permite transmisión *full duplex* y *half duplex*.
- Tres modos de transferencia de datos: modo de respuesta normal, utiliza configuración no balanceada; modo balanceado asíncrono, usa configuración balanceada y modo de respuesta asíncrono, emplea configuración no balanceada.
- Orientación por *bit*.

Aunque los protocolos SDLC y HDLC parecen análogos, no existe una correspondencia directa, ya que las arquitecturas para las cuales fueron desarrolladas son completamente diferentes. Hechas las especificaciones para el control de transferencia de datos en un entorno de infrarrojos, IrLAP define las siguientes características.

Las conexiones son punto a punto. Los dispositivos que se van a comunicar deben estar frente a frente, a un metro de distancia, para realizar intercambio de información.

Comunicaciones half duplex. En los dispositivos que se van a comunicar, los datos son enviados en uno de dos sentidos, alternándose el turno para transmitir.

Cono angosto de infrarrojos. La transmisión de pulsos infrarrojos es direccional dentro de un ángulo medio de 15 grados.

Interferencia. Componentes de luz infrarroja que están contenidos en luces fluorescentes, el sol e inclusive la luna, así como dispositivos que estén próximos a los equipos en comunicación.

No detección de colisiones. El *hardware* no puede detectar las coaliciones, es el *software* empleado para cada aplicación quien debe controlar estos inconvenientes.

Dispositivos maestro y esclavo. Los dispositivos que interactúan en una comunicación se pueden definir como maestro y esclavo, transmisor y receptor o estación primaria y estación secundaria respectivamente. IrDA define a estos elementos de la siguiente manera:

- *La estación primaria (dispositivo maestro):* es la encargada de enviar los comandos de inicio de conexión, garantiza el flujo organizado y el control de los datos, así como el tratamiento de los errores en la transmisión. Los dispositivos maestros son típicamente

computadoras personales, agendas electrónicas o elementos que requieran imprimir o realizar algún tipo de envío de información a otro.

- *La estación secundaria* (dispositivo esclavo): transfiere las respuestas a los requerimientos de inicio de conexión y envío de datos, realizadas por el otro extremo. Los dispositivos esclavos son impresoras u otros periféricos que son usados por otros componentes.

Tipos de operación de dispositivos. Existen dos modos de operación posibles para el nivel IrLAP, de desconexión normal y de respuesta normal.

Modo de Desconexión Normal. Es el estado por omisión de los dispositivos que no se encuentran ejecutando una comunicación, estos elementos se encuentran revisando si otros equipos están realizando comunicaciones; esta operación se debe hacer en intervalos de tiempo no menores a 500 ms, para decidir si es posible o no realizar una solicitud de conexión.

Existen dispositivos que carecen de interfaz de usuario para configurar los parámetros de comunicación, por lo tanto, las transmisiones en este estado de operación se realizan con una configuración fija: asíncrona, 9600 bps, 8 bits y sin control de paridad. Después, durante el proceso de negociación en la conexión, se pueden establecer nuevos y mejores parámetros.

Modo de Respuesta Normal. En este modo, los dispositivos se encuentran conectados realizando una comunicación con los parámetros establecidos por mutuo acuerdo.

Estructura de los paquetes de información. El protocolo IrLAP proporciona la estructura para los paquetes de datos. La figura II.24 muestra como IrLAP tiene organizada una trama.

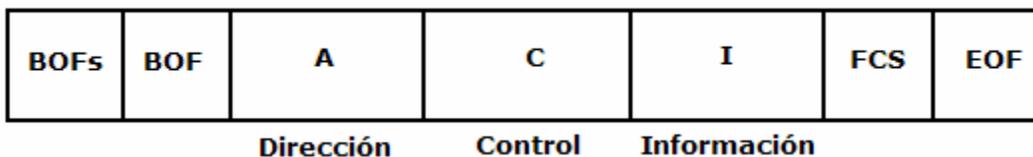


Figura II.24. Formato de un paquete de datos en IrDA.

La trama inicia con un número de caracteres de inicio (BOF). Si se envían varios BOF, es un aviso para que el otro dispositivo sepa que está entrando un paquete de datos. Posteriormente la trama continúa con un *byte* de dirección A y un *byte* de control C. El *byte* de control es

para diferenciar entre diferentes tipos de paquetes de datos y también es utilizado para su conteo. Una trama puede llevar datos o comandos. El protocolo de IrLAP tiene una sintaxis de comando propio y este comando es parte del *byte* de control. Si contiene datos el paquete, estos se encuentran en el campo I. La integridad de la trama está asegurada con 16 *bits*, referida como secuencia de revisión de trama (FCS). El fin de la trama (EOF) es marcado con un carácter. La estructura del paquete de datos descrita es usada para todas las versiones del protocolo estándar de IrDA en velocidades SIR.

Nivel IrLMP

Cuando dos dispositivos contienen los estándares IrDA de conexión, hay generalmente uno (maestro) que tiene un trabajo por hacer y el otro (esclavo) tiene los recursos para realizarlo. Por ejemplo una *lap top* tiene un archivo para imprimir y una impresora posee los recursos para imprimirlo. En el momento que los dispositivos se conectan, el maestro investiga las especificaciones del esclavo para determinar si es capaz de hacer el trabajo.

Esta determinación la hace el dispositivo primario realizando al dispositivo secundario una serie de preguntas. Dependiendo de las respuestas, el maestro puede o no elegir la conexión al esclavo. Este intercambio es ejecutado entre los dispositivos, a través de su Protocolo de Administración de Enlace en IrDA (IrLMP) respectivo. Las respuestas a tales preguntas pueden ser encontradas en el nivel Servicio de Acceso a la Información (IAS) del dispositivo secundario.

En la medida en que puede haber varias conexiones IrLMP sobre un único canal IrLAP, existe un nivel de direccionamiento de complejidad mayor, el cual involucra los siguientes conceptos:

- *Puntos Lógicos de Acceso al Servicio (LSAP)*. Es un punto donde se puede tener acceso a un servicio o aplicación dentro de IrLMP.
- *Selector de Puntos Lógicos de Acceso al Servicio (LSAP-SEL)*. Es un *byte* que identifica un LSAP, lo que equivale a la dirección del servicio dentro del multiplexor de IrLMP.

Nivel IAS

El nivel IAS actúa como un directorio para un dispositivo, de modo que cada punto de este directorio debe tener los servicios y aplicaciones disponibles para las conexiones entrantes. Este servicio se utiliza para determinar las direcciones de los servicios o selectores LSAP y LSAP-SEL

y puede ser consultado para obtener información adicional sobre los mismos.

Nivel TinyTP

El nivel Protocolo de Transporte Pequeño (TinyTP) es opcional y tiene asociadas dos funciones:

- El control de flujo de información en las conexiones hechas sobre IrLMP.
- La segmentación y regeneración de los paquetes de información.

El control de flujo de los datos es la tarea más importante de TinyTP, esta acción es aplicada sobre cada una de las conexiones de IrLMP, a través de un sistema de permisos (créditos) para transmitir, con el cual se logra que uno de los extremos pueda detenerse para procesar la información sin afectar la comunicación en sentido opuesto.

Una vez en conexión, cada dispositivo ofrece los créditos y al mismo tiempo se habilita para recibir paquetes de determinado tamaño. El número de créditos ofrecidos depende de la capacidad que tiene el dispositivo receptor para almacenar paquetes en la memoria de recepción.

Así mismo, en la medida que se reciben paquetes y se va liberando memoria, se realiza un nuevo ofrecimiento de estos por parte del receptor, lo que significa que es el receptor quien tiene el control de la comunicación. Entonces, puede presentarse un problema en el rendimiento cuando el transmisor tenga que esperar mucho tiempo en la recepción de nuevos créditos, considerando que sin existir estos, no habrá transmisión de información.

La otra tarea a cargo de TinyTP es la segmentación y regeneración de los paquetes, es decir, los paquetes muy grandes son divididos en fragmentos cuyo tamaño se define durante la negociación de la comunicación con el protocolo IrLAP. Posteriormente, en el lado del receptor la tarea es reensamblar los fragmentos en el orden adecuado para obtener el paquete original.

Nivel IrOBEX

El nivel Intercambio de Objetos en IrDA (IrOBEX) es opcional dentro de la arquitectura de protocolos de IrDA. Su función es permitir a dispositivos de diferentes características intercambiar datos y comandos

en un modo estandarizado de acuerdo a los recursos presentes en cada uno, y así, hacer del intercambio de archivos o mensajes un procedimiento transparente para la aplicación del usuario.

Nivel IrCOMM

El nivel Comunicación IrDA (IrCOMM) es simplemente una sintaxis que permite al dispositivo primario considerar al dispositivo secundario como un dispositivo serial. IrCOMM tiene como objetivo dentro de la arquitectura de IrDA, permitir que las interfaces seriales y paralelas de los antiguos dispositivos periféricos, puedan operar a través de infrarrojos sin ningún problema. El protocolo de nivel IrCOMM se desarrolló para ofrecer el uso de estas interfaces en la tecnología IrDA, aunque es un nivel opcional.

IrCOMM define cuatro tipos de servicios:

- *3 Hilos Puro*: simulación de la interfaz serial y paralela para enviar solo datos, sin control del canal y soportado en TinyTP.
- *3 Hilos*: emulación de la interfaz serial y paralela con mínimo uso de control del canal y soportado en TinyTP.
- *9 Hilos*: representación serial únicamente con control del canal para estado del estándar EIA-232 y soportado en TinyTP.
- *Centronics*: especificación paralela únicamente con control del canal para estado de los circuitos de *centronics*.

Nivel IrLAN

El nivel Redes de Área Local en IrDA (IrLAN) permite que dispositivos con tecnología IrDA accedan a redes de área local, las cuales se han definido con una arquitectura cliente/servidor. El cliente es el dispositivo quien descubre y se conecta con el dispositivo servidor, inicia la instalación a través de la información de servicios y aplicaciones que le proporciona el nivel IAS, después realiza la conexión por medio de un canal de control negociando las características de un canal de datos exclusivo para la transferencia de paquetes LAN.

Existen tres maneras para realizar una conexión a través de IrLAN.

- *Modo de punto de acceso*: permite a una computadora acceder a una red local a través de un dispositivo llamado punto de acceso, que consiste en un adaptador de infrarrojos a la red cableada.
- *Modo igual a igual*: permite a dos computadoras interactuar como si fueran parte de una red local formada por ellas mismas.

- *Modo de host*: una computadora que accesa a la red de área local, a través de otro equipo de cómputo que está instalado en la red.

II.4.2. Estándar *Bluetooth*

La transferencia de información en forma inalámbrica se puede realizar por medio de la tecnología *Bluetooth*, que es una tecnología muy noble y tiene en la actualidad una gran aceptación dentro de las empresas que desarrollan telecomunicaciones. Esta tecnología ha revolucionado el mercado de la conectividad, ya que es capaz de comunicar cualquier dispositivo que cumpla con las normas inalámbricas del estándar *Bluetooth*. La principal ventaja que ofrece esta tecnología es la conexión sin cables en los dispositivos involucrados.

Definición

Bluetooth es la especificación que define un estándar mundial en las comunicaciones inalámbricas para la recepción y transmisión de voz y datos entre diferentes equipos, mediante el enlace por radiofrecuencia en un medio móvil y estático. *Bluetooth* es una tecnología orientada a la conectividad inalámbrica entre diversos dispositivos existentes, tales como: electrodomésticos, PC's, teléfonos celulares, PDA's, etc.

Bluetooth es el término utilizado para describir el protocolo de un enlace de radio por salto de frecuencias de corto alcance entre dispositivos.

Objetivos de *Bluetooth*

La tecnología *Bluetooth* comprende *hardware*, *software* y requerimientos de interoperabilidad, por lo que para su desarrollo ha sido necesaria la participación de los principales fabricantes en los sectores de las telecomunicaciones y la informática.

La tecnología *Bluetooth* tiene como principales objetivos los siguientes:

- Ser un sistema basado en un protocolo robusto y seguro.
- Utilizar microcircuitos de bajo costo.
- El emisor de radio debe consumir poca energía, puesto que va a estar operando en dispositivos alimentados por baterías.
- El sistema deberá ser capaz de establecer comunicación entre dispositivos que cumplan las especificaciones de *Bluetooth*, sin importar su naturaleza.
- Ser de baja complejidad.
- Ser un estándar universal.

Historia de Bluetooth

En 1994, *Ericsson Comunicación Móvil*, la compañía global de telecomunicaciones con base en Suecia, comenzó un estudio con base en la viabilidad de crear una interfaz de radio de baja potencia y de bajo costo entre teléfonos celulares y otros dispositivos, con el fin de eliminar los cables. Hasta entonces, la tecnología de comunicaciones estaba basada en la conexión por medio de cables y funcionaba con eficiencia, pero su instalación y configuración resultaba muy engorrosa.

El estudio era parte de un proyecto más amplio que investigaba como conectar diferentes dispositivos de comunicaciones a la red, a través de un teléfono celular. La compañía determinó que el enlace final en este tipo de conexión debería de ser un enlace de radio de corto alcance.

A medida que se desarrollaba el proyecto, se hizo evidente que este tipo de enlace de corto alcance podría utilizarse en muchas aplicaciones. Conforme avanzaba el proyecto de *Ericsson* fue despertando el interés de otros fabricantes, así que, a principios de 1998 se creó el Grupo de Interés Especial (SIG), el cual estaba formado por *Ericsson, Nokia, IBM, Toshiba* e *Intel*, con el objetivo de desarrollar, promover, definir y publicar las especificaciones para esta nueva tecnología inalámbrica de corta distancia.

En mayo del mismo año, se invitó a otras empresas al grupo: *Microsoft Corporation, 3Com Corporation, Lucent Technologies* y *Motorola Inc.* Hoy en día, este grupo está formado por más de 2,000 empresas del sector de la informática y las telecomunicaciones. El propósito principal del grupo SIG es establecer un estándar para la interfaz aérea junto con su *software* de control, la finalidad es asegurar la interoperabilidad de los diversos equipos de diferentes fabricantes.

El nombre *Bluetooth* es en honor a un rey Danés del siglo X llamado Harald Blatand (traducido al idioma inglés como Harold Bluetooth), conocido por unificar las tribus en guerra de Noruega, Suecia y Dinamarca e iniciar el proceso de cristianización de la sociedad vikinga. Fue rey de Dinamarca entre los años 940-981 D.C. La elección para nombrar como *Bluetooth* a esta nueva tecnología, se debe a que de la misma manera, se pretende unificar a los diferentes dispositivos, tales como: computadoras, teléfonos celulares, PDA's, etc.

Cabe comentar que el logotipo de *Bluetooth* es representado por la combinación de dos caracteres de los antiguos alfabetos germánico y escandinavo (denominados runas): Hagalas, transcrito como H y

Berkana, transcrito como B. Dichos caracteres se juntaron en un símbolo único mostrado en la figura II.25.



Figura II.25. Símbolo y logotipo de Bluetooth.

II.4.2.1. Tecnología *Bluetooth*

La comunicación entre dispositivos *Bluetooth* se puede establecer cuando dos o más equipos de comunicación se encuentran dentro del mismo radio de cobertura. Un conjunto de hasta 8 dispositivos *Bluetooth* que pueden enlazarse para intercambiar información se denomina *piconet*. Los dispositivos que conforman una *piconet* solamente pueden obtener un papel, maestro (M) o esclavo (S). La unión de varias *piconet* se denomina *Scatternet* (Figura II.26).

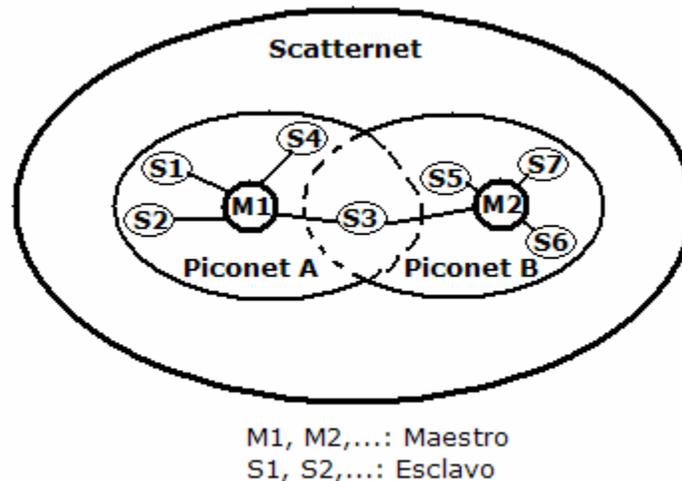


Figura II.26. Scatternet.

En cada *piconet* sólo puede haber una unidad maestra que normalmente es quien inicia la conexión, el resto de unidades *bluetooth* se denominan esclavas. El maestro es el dispositivo de una *piconet* que establece una sincronización hacia todos los demás esclavos. Cuando se establece la conexión, se realiza una adaptación a la frecuencia del reloj de la unidad esclava para poder sincronizarse con el reloj del maestro. El reloj de la unidad maestra mantiene siempre constante su frecuencia, así que, la adaptación producida por las unidades esclavas para sincronizarse con el maestro, sólo son válidos mientras dura la conexión.

Dentro de la misma área pueden coexistir diversas *piconets*, ya que cada *piconet* tiene una unidad maestra distinta con su propia sincronización. A medida que tenemos más *piconets* en la misma superficie de cobertura, la probabilidad de colisión aumenta, produciendo una degradación del espectro y reduciendo el rendimiento del sistema.

Una unidad *bluetooth* puede participar en varias *piconets*. Esto es posible siempre y cuando la unidad sólo esté activa en una *piconet* a la vez. Para realizar la incorporación a otra *piconet*, debe realizar el proceso de sincronía a la nueva *piconet*.

Cuando una unidad abandona una *piconet*, la esclava informa al maestro actual que ésta no estará disponible por un determinado periodo de tiempo y que se activará en otra *piconet*. Durante su ausencia, la comunicación en la *piconet* entre el maestro y otros esclavos continúa con normalidad.

Una unidad maestra también puede cambiar de *piconet*, pero en este caso la comunicación se desactiva, reiniciándose hasta el regreso de ésta. La maestra que entra en una nueva *piconet*, al principio lo hace como esclava, posteriormente, puede solicitar actuar como maestra.

Especificación del protocolo Bluetooth. Toda la estructura de la tecnología *Bluetooth* está clasificada en dos secciones:

1. *Especificaciones de Bluetooth.* Describe como trabaja la tecnología a través de una pila de protocolos llamada arquitectura de protocolos de *Bluetooth*.
2. *Perfiles de Bluetooth.* Describe como es utilizada la tecnología. Las partes que componen las especificaciones pueden ser usadas para obtener una función o la aplicación deseada de un dispositivo *Bluetooth*.

II.4.2.2. Especificaciones de *Bluetooth*

Las especificaciones de *Bluetooth* utilizan una arquitectura de protocolos que clasifica las diversas funciones de red en un sistema de niveles o capas, que en su conjunto desarrollan todo lo necesario para el intercambio de información entre las aplicaciones diseñadas de acuerdo a cierta especificación, así como la interoperabilidad entre los dispositivos de diferentes fabricantes. La arquitectura de protocolos de *Bluetooth* se basa en el modelo de referencia OSI de ISO para la interconexión de sistemas abiertos (Apéndice A).

La arquitectura o pila de protocolos de *Bluetooth* (Figura II.27) se divide en dos partes, cada una de las cuales se implementa en un procesador:

- *Módulo Bluetooth*: es el *hardware* encargado de las actividades relacionadas con el intercambio de información a través de la interfaz de radiofrecuencia.
- *Host Bluetooth*: es el *software* encargado de la parte relacionada con los niveles o capas superiores de enlace y las aplicaciones del dispositivo.

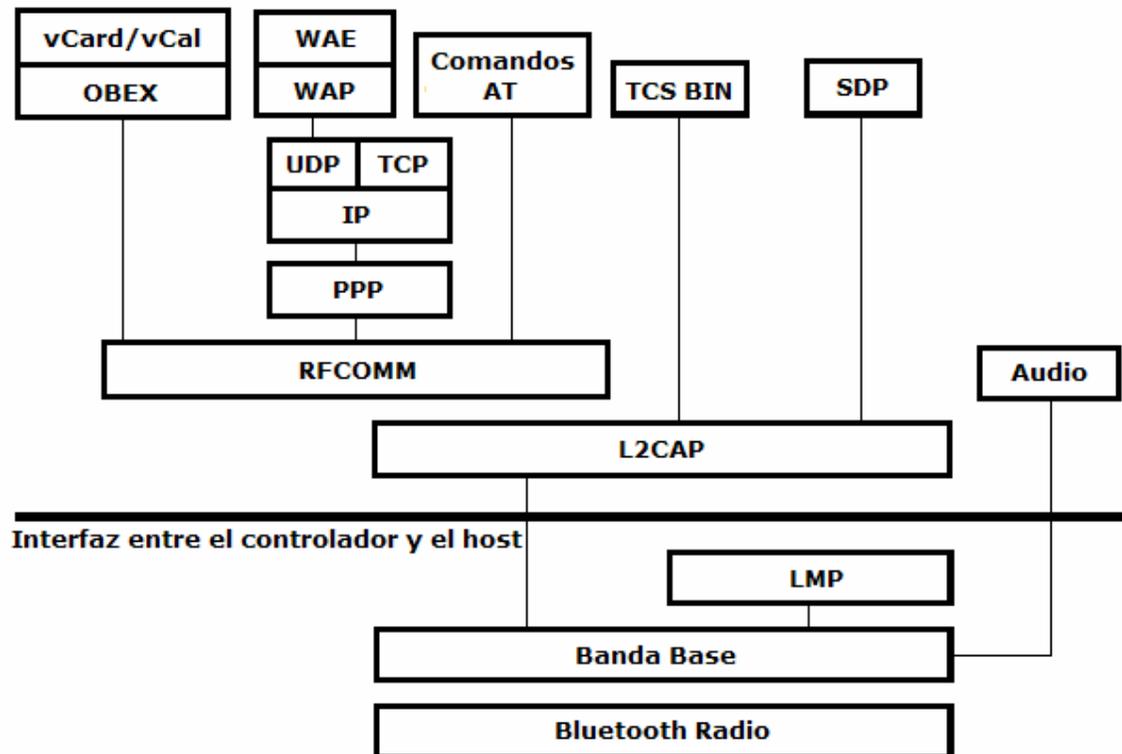


Figura II.27. Arquitectura de protocolos de Bluetooth.

La zona intermedia que comunica ambas partes es la capa Interfaz entre el Controlador y el *Host* (HCI).

Cada fabricante puede implementar su propia capa de protocolos, de esta manera, la especificación de *Bluetooth* amplía la cantidad de aplicaciones que ofrece esta tecnología inalámbrica.

De una manera sencilla, toda la arquitectura de protocolos se podría resumir en cuatro niveles o capas, considerando la aplicación de los dispositivos así como su funcionamiento, estos son: capa de protocolos

del núcleo de *Bluetooth*, capa de protocolos de reemplazo de cable, capa de protocolos de control de telefonía y capa de protocolos adoptados.

Capa de protocolos del núcleo de Bluetooth: son los protocolos requeridos por casi la totalidad de dispositivos con tecnología *Bluetooth*: Banda base, LMP, L2CAP y SDP.

Capa de protocolo de reemplazo de cable: es el protocolo RFCOMM.

Capa de protocolos de control de telefonía: son los protocolos que se utilizan para aplicaciones telefónicas: TCS BIN y AT *Commands*.

Capa de protocolos adoptados: son protocolos que se usan en ciertas aplicaciones, tales como PPP, UDP/TCP/IP, OBEX, WAP, vCard, vCal, IrMC y WAE.

La arquitectura o pila de protocolos del estándar de comunicación inalámbrica *Bluetooth* está compuesta de capas, entre las capas más importantes se pueden mencionar las siguientes: capa *Bluetooth* Radio, capa Banda Base de *Bluetooth*, capa audio, capa Protocolo Administrador de Enlace (LMP), capa Interfaz entre el Controlador y el *host* (HCI), capa Protocolo de Adaptación y Control de Enlace Lógico (L2CAP), capa Comunicación por Radiofrecuencia (RFCOMM), capa Protocolo de Descubrimiento de Servicios (SDP), capa Intercambio de Objetos (OBEX), formatos de contenido vCard y vCal, capa Protocolo Punto a Punto (PPP), capa TCP/UDP/IP, capa Protocolo de Acceso Inalámbrico (WAP), capa Protocolo de Especificación Binaria y Control de Telefonía (TCS BIN) y la capa Comandos de Atención (AT).

Las capas anteriormente nombradas no son todas las que contiene el estándar *Bluetooth*, pero si se consideran dentro de las más fundamentales, así que solamente se describirán las mencionadas.

Capa Bluetooth Radio

El nivel *Bluetooth* radio es la parte más baja de las especificaciones y define los requerimientos del dispositivo transceptor *Bluetooth*, es decir, permite el enlace físico por radiofrecuencia entre dispositivos dentro de una *piconet*, realizando actividades de modulación y demodulación en las señales de radiofrecuencia que están siendo transmitidas por aire.

Banda de frecuencias y arreglo de canales. *Bluetooth* define las siguientes especificaciones técnicas:

- La frecuencia de radio en la que trabaja *Bluetooth* se sitúa en el rango de 2.402 a 2.480 GHz de la banda Medica Científica Industrial (ISM), que está disponible a nivel mundial y que no requiere licencia de operador.
- La banda ISM tiene muchas interferencias, para solucionar este problema con otros protocolos que operan en la misma banda de frecuencias, *Bluetooth* emplea la técnica de salto de frecuencias de amplio espectro (FHSS) mostrada en la figura II.28. Esta técnica consiste en dividir la banda de frecuencias en 79 canales de 1 MHz cada uno, estos canales se accesan de acuerdo a una secuencia de saltos aleatoria la cual la determina el dispositivo maestro; para la transmisión de datos cada canal está compuesto por intervalos de tiempo de 625 μs , nombrados *slots*, donde cada salto de frecuencia es ocupado por un *slot*. Esto da lugar a una frecuencia de salto de 1600 veces por segundo, en la que un paquete de datos ocupa un *slot* para la emisión y otro para la recepción y que pueden ser usados alternadamente, dando lugar a un esquema de tipo División de Tiempo *Duplex* (TDD). De esta manera se consiguen transceptores de banda estrecha con una gran inmunidad a las interferencias.

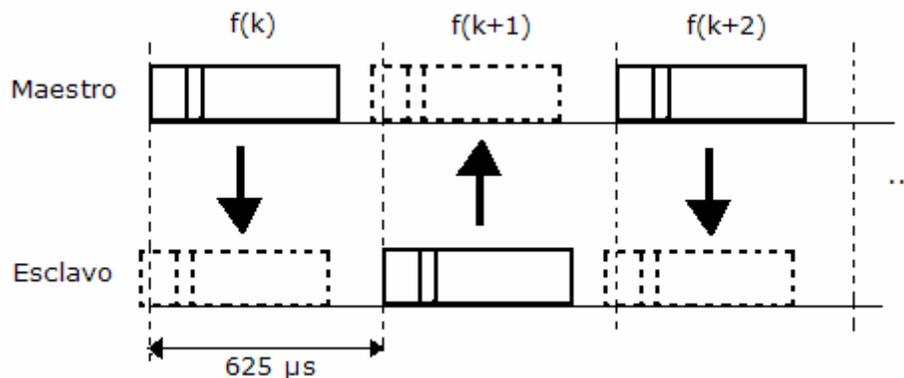


Figura II.28. Técnica de salto de frecuencias FHSS.

Los dispositivos *Bluetooth* se clasifican por su potencia o por la velocidad de transmisión.

Potencia de dispositivos Bluetooth. De acuerdo con su potencia, los dispositivos *Bluetooth* se clasifican en clases:

- *Clase 1:* dispositivos de largo alcance (aproximadamente 100 m) con potencia máxima de 20 dBm (100 mW).
- *Clase 2:* dispositivos de medio alcance (aproximadamente 10 m) con potencia máxima de 4 dBm (2.5 mW).

- *Clase 3*: dispositivos de corto alcance (aproximadamente 1 m) con potencia máxima de 0 dBm (1 mW).

La velocidad de transmisión. La versión de los dispositivos *Bluetooth* indica su velocidad de transmisión de datos:

- *Versión 1.1*: 723.1 kbps.
- *Versión 1.2*: 1.0 Mbps.
- *Versión 2.0 + EDR*: 2.1 a 3.0 Mbps.

Codificación de datos. El módulo de radio de *Bluetooth* versión 1.2 utiliza el formato de codificación de Modulación por Desplazamiento de Frecuencia Gaussiana (GFSK) para una velocidad máxima de transmisión de datos de 1 Mbps. El formato GFSK es una versión mejorada del formato FSK. El formato GFSK actúa de la siguiente manera: la información es pasada por un filtro Gaussiano, donde un filtro Gaussiano es una de las formas estándar de reducir el ancho de banda; posteriormente a las señales que salen del filtro se les aplica modulación FSK. Dando como resultado un espectro de energía más angosto, permitiendo mayores velocidades de transmisión de datos por el mismo canal.

GFSK proporciona una gran inmunidad al ruido y confiabilidad en la transmisión de datos. Un '1' lógico es representado por un incremento en la frecuencia de la onda portadora y un '0' lógico por un decremento de la misma. La figura II.29 muestra la modulación GFSK.

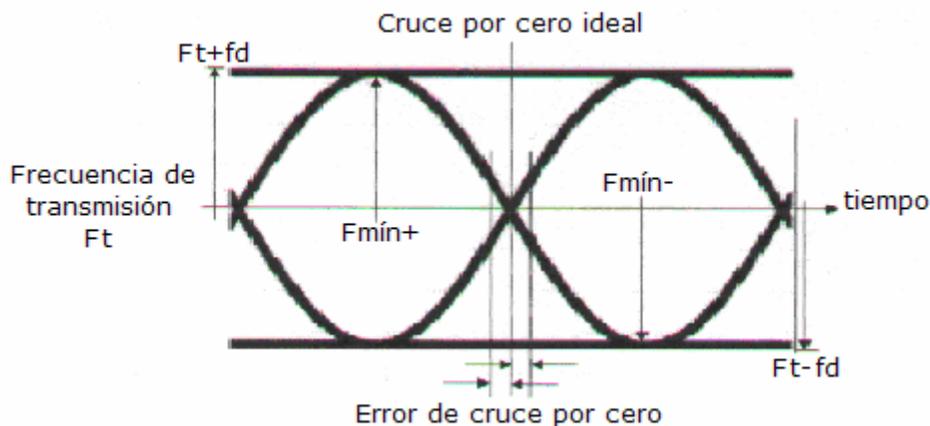


Figura II.29. Modulación GFSK.

Capa Banda Base de Bluetooth

La banda base es la capa física de *Bluetooth*. Maneja los canales físicos y enlaza a distancia los servicios, tales como: corrección de errores, selección de saltos y seguridad de *Bluetooth*.

El protocolo de banda base es implementado como un controlador de enlace, maneja los paquetes de datos, así como los enlaces síncronos y asíncronos y ejecuta el protocolo de preguntas para establecer el acceso a dispositivos *Bluetooth* en las áreas circunvecinas.

Un transceptor en banda base usa el esquema TDD, esto es, cuando el maestro y el esclavo transmiten alternadamente. El maestro comenzará su transmisión de datos en *slots* de tiempo numerados en par y el esclavo iniciará la transferencia solamente en *slots* de tiempo numerados en impar.

Enlace Físico. En la especificación de banda base de *Bluetooth* se definen dos tipos de enlace, como se muestra en la figura II.30.

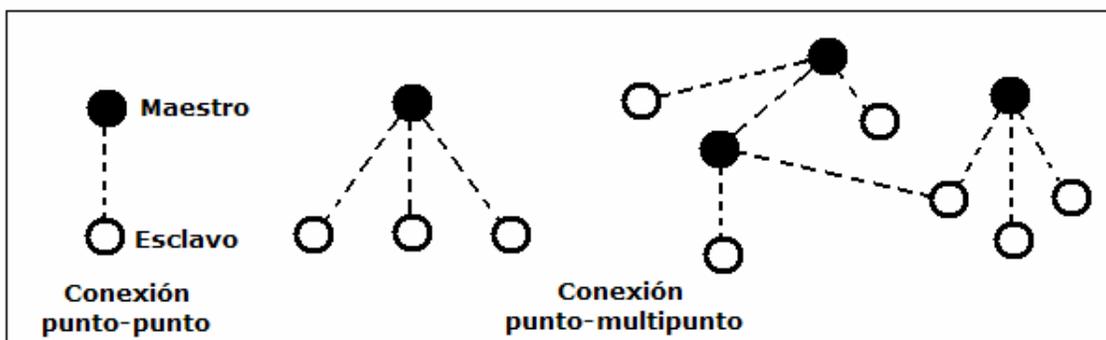


Figura II.30. Tipos de conexión.

Enlace de Conexión Orientada Síncrona (SCO), donde sus principales características son:

- Conexión punto a punto entre maestro y esclavo.
- El maestro usa *slots* de tiempo reservados a intervalos regulares.
- El maestro puede soportar más de 3 enlaces simultáneos mientras que los esclavos 3 como máximo.
- Los paquetes de configuración nunca son retransmitidos.
- Este tipo de enlace se utiliza principalmente para transmitir información de voz con una velocidad de 64 kbps.

Enlace de Conexión Baja Asíncrona (ACL), cuyas características son:

- Conexiones punto a multipunto entre maestro y esclavo.

- Enlace utilizado para la transmisión de datos.
- Se aplica retransmisión de paquetes.
- Definidas para este tipo de conexión los *slots* 1,3 y 5.
- La máxima velocidad de transferencia de datos es de 721 kbps en una dirección y 57.6 kbps en la otra.

Canales lógicos. *Bluetooth* tiene cinco canales lógicos, los cuales son utilizados para transferir información del nivel de enlace, así como información del usuario.

Direccionamiento del dispositivo. Una unidad *Bluetooth* puede asignar cuatro tipos de direccionamiento, los cuales son: la dirección del dispositivo *Bluetooth*, la dirección del miembro activo, la dirección del miembro en espera y la dirección del que requiere acceso.

Paquetes de datos. Los dispositivos *Bluetooth* intercambian paquetes de datos entre ellos, y cada paquete está formado por un conjunto de *slots*. Existen diferentes tipos de paquetes y están definidos en la banda base del sistema *Bluetooth*.

Formato de paquetes de datos, figura II.31. El formato que tienen los paquetes de datos está constituido por tres campos de información, los cuales son: el código de acceso, la cabecera y la información.

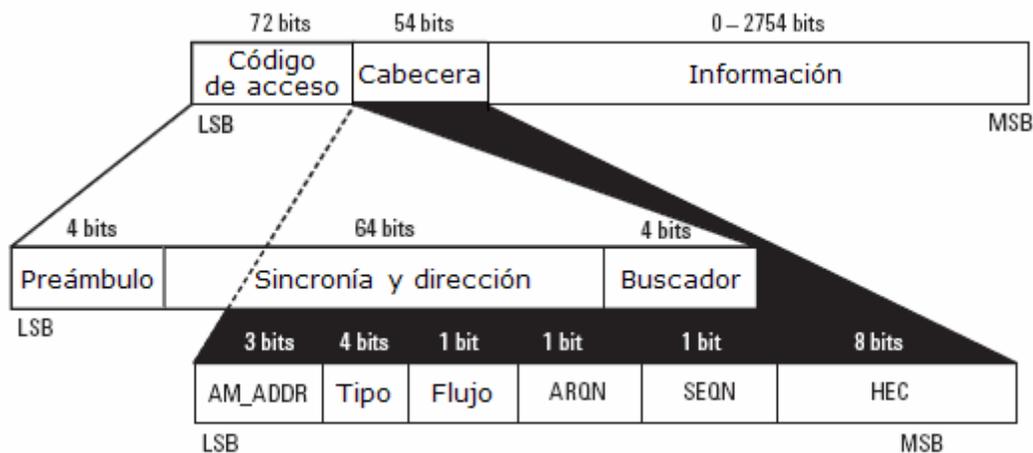


Figura II.31. Formato de paquete de datos en Bluetooth.

Código de acceso: contiene de 68 a 72 bits, inicia con un preámbulo de comunicación, sincroniza y da la dirección del dispositivo y por último realiza la búsqueda de dispositivos. Existen tres diferentes tipos de código de acceso: código de acceso de canal, que identifica una *piconet* única; código de acceso a dispositivo, que es utilizado para preguntas y

respuestas, y el código de acceso de petición, que se utiliza para solicitar información del dispositivo.

Cabecera: contiene 54 *bits*, en los cuales se puede hallar información de control; entre la información más importante se puede destacar la siguiente: tres *bits* de acceso de dirección al dispositivo activo (AM-ADDR), cuatro *bits* que representa el tipo de paquete que es enviado, un *bit* de control de flujo de datos, un *bit* de reconocimiento de paquete recibido (ARQN), un *bit* para evitar retransmisión en el receptor (SEQN) y ocho *bits* de revisión de errores en la transmisión de datos (HEC).

Información: su longitud varía entre 0 y 2754 *bits* y puede contener información de voz, datos o ambos.

Los paquetes de datos están protegidos por un esquema de repetición automática de consulta, es decir, los paquetes son automáticamente retransmitidos. Aún así, con este sistema, si un paquete de datos no llegara a su destino, sólo una pequeña parte de la información se perdería.

La voz no se retransmite nunca, sin embargo, se utiliza un esquema de codificación muy robusto, el cual sigue la forma de la onda de audio y es muy resistente a los errores de *bits*. Estos errores son percibidos como ruido de fondo, que se intensifica si los errores aumentan.

Los receptores de la *piconet* comparan las señales que reciben con el código de acceso, si éstas no coinciden, el paquete recibido no es considerado como válido en el canal y el resto de su contenido es ignorado.

Controlador de estados de comunicación. El controlador del estado de comunicación en *Bluetooth* tiene la siguiente clasificación:

- *Estado de espera:* este estado está colocado por omisión, posee baja potencia, sólo su reloj interno corre y no hay interacción con algún otro dispositivo.
- *Estado de conexión:* se presenta cuando el maestro y el esclavo intercambian paquetes de información. Este estado puede permanecer en alguno de los cuatro modos siguientes: modo activo, donde el maestro organiza la comunicación basada en la demanda de transmisión desde y hacia los esclavos; modo de olfateo, donde los dispositivos en la *piconet* están en su actividad reducida, este modo es programable y depende de la aplicación; modo de agarre, donde los dispositivos sincronizados en una

piconet reducen la potencia en la cual su actividad es muy baja y modo estacionado, donde un dispositivo está sincronizado en una *piconet* pero no está participando en la transmisión de información.

Procedimiento de conexión. Normalmente una conexión entre dos dispositivos ocurre de dos maneras: cuando un dispositivo remoto es absolutamente desconocido y si el dispositivo remoto es conocido.

Dispositivo desconocido: para este caso un procedimiento de conexión hace el reconocimiento y habilita al dispositivo que se encuentra dentro de su alcance, determina el reloj y la dirección de éste. El procedimiento lo inicia la unidad maestra, enviando un paquete de preguntas. La unidad esclava estará a la espera del paquete de preguntas, posteriormente entrará al estado de respuestas, es decir, enviará las respuestas a las preguntas formuladas por la unidad maestra.

Dispositivo conocido: en este caso el procedimiento de conexión sólo requiere la dirección del dispositivo para realizar la conexión; adaptará el reloj y acelerará el proceso de conexión. La unidad que establece una conexión dirigirá el procedimiento y automáticamente será el maestro. El maestro busca al dispositivo destinatario. El destinatario recibe la búsqueda y envía una respuesta al maestro. El maestro transmite un paquete al destinatario y éste regresa una segunda respuesta al maestro. El destinatario y el maestro entonces intercambian los parámetros del canal de comunicación.

Capa audio

Los datos de audio pueden ser transferidos hacia uno o más dispositivos *Bluetooth*, teniendo varias posibilidades de uso, los datos de audio en paquetes SCO son transmitidos directamente hacia y desde el protocolo Banda base y no cruzan a través del protocolo L2CAP.

Una aplicación de audio es relativamente sencilla, puesto que dos dispositivos *Bluetooth* cualesquiera pueden enviar y recibir datos de sonido sólo abriendo un enlace de audio.

Capa LMP

La capa Protocolo de Administrador de Enlace (LMP) está encargada de configurar y llevar a cabo el control de enlace entre dispositivos *Bluetooth*. Todos los equipos *Bluetooth* tienen un Administrador de Enlace (LM). El administrador o gestor de enlace de un dispositivo se

comunica con otro a través de mensajes por medio del protocolo LMP, sólo si se encuentra dentro de su alcance. Tales mensajes realizan el enlace entre estos dispositivos.

El protocolo LMP se encarga de realizar las funciones que se relacionan con la seguridad, tales como la autenticación y encriptamiento, en las cuales se realiza la generación, intercambio y comprobación de las claves de enlace y de encriptado.

Capa HCI

La capa Interfaz entre el Controlador y el *Host* (HCI) actúa como una línea divisoria entre el *hardware* (representado por el módulo *Bluetooth* llamado controlador) y el *software* (representado por el *host Bluetooth*). Es una capa de *software*, la cual pasa todos los datos desde la computadora hacia el módulo *Bluetooth*. Por ejemplo, si se está tratando de comunicar inalámbricamente desde una computadora (*host*), y se tiene un dispositivo *Bluetooth* (controlador) sujetado o conectado al puerto USB, entonces se necesitará esta capa que puede entender y manipular el estándar USB y enviar la información hacia las capas superiores.

Todos los datos pasan forzosamente a través de la capa HCI. La capa HCI proporciona una interfaz de comandos entre el controlador de banda base y el administrador de enlace, es decir, estos comandos realizan la comunicación entre el dispositivo y el *firmware* del módulo *Bluetooth*.

Por su funcionamiento la capa HCI se define en tres partes separadas:

- *HCI Firmware*: está localizado en la sección controladora de *host*, es decir, en el dispositivo (módulo *Bluetooth*). El *firmware* es un *software* de aplicación e implementa los comandos en el módulo *Bluetooth*.
- *HCI Driver*: es el *software* que se utiliza para controlar el funcionamiento del módulo *Bluetooth*, así como la comunicación con dispositivos exteriores.
- *Capa de transporte entre el controlador y el host*: el *driver* y el *firmware* se comunican por medio del nivel HCI. Esta capa intermedia deberá tener la habilidad de transferir datos sin entenderlos. Varias capas entre el controlador y el *host* pueden ser utilizadas, de las cuales tres han sido definidas por *Bluetooth*: USB, UART y EIA-232.

Comandos de host. La capa HCI proporciona una metodología simple de comandos para acceder a las posibilidades del *hardware* de dispositivos *Bluetooth*. Los comandos de enlace de HCI proporcionan al *host* la habilidad de controlar la conexión en la capa de enlace hacia otros dispositivos *Bluetooth*. Estos comandos involucran normalmente al administrador de enlace (LM) para intercambiar comandos (LMP) con dispositivos *Bluetooth* remotos, así que, la información que se intercambia es específica.

A continuación son enlistados los tipos de comandos:

- Comandos para el control de enlace.
- Comandos con la norma de enlace.
- Comandos de banda base y controlador de *host*.
- Parámetros de información: referente al módulo *Bluetooth* y son fijos, programados de fábrica.
- Parámetros de estado.
- Comandos de prueba.

Capa L2CAP

La capa Protocolo de Adaptación y Control de Enlace Lógico (L2CAP) se encuentra en una capa superior del protocolo de banda base y reside en la capa de enlace de datos. Las funciones del protocolo L2CAP son las siguientes:

- La segmentación y reconstrucción de los paquetes, esto es necesario para permitir la utilización de protocolos que usan paquetes de mayor tamaño que los soportados por la capa de la banda base. Cuando el transmisor envía paquetes grandes de información, el protocolo L2CAP los segmenta, es decir, los parte en paquetes más pequeños de formato adecuado al protocolo de banda base. Al arribar al receptor, estos paquetes se reensamblan en el protocolo L2CAP y se regenera el paquete original. El protocolo L2CAP permite paquetes de información de hasta 64 *kBytes* de longitud.
- Se encarga de la multiplexión de protocolos, ya que el protocolo de banda base no puede identificar un protocolo de nivel superior al que quiere transmitir información, por ejemplo el RFCOMM.
- El protocolo L2CAP sólo está definido para realizar enlaces ACL.
- La implementación del protocolo L2CAP, en cada uno de los dispositivos *Bluetooth*, es para llevar a cabo un control de los recursos utilizados por el protocolo y asegurar el cumplimiento de la calidad de servicio.

Capa RFCOMM

La capa Comunicación por Radiofrecuencia (RFCOMM) proporciona una simulación de puertos seriales EIA-232 sobre el protocolo L2CAP. Este protocolo está basado en un estándar del Instituto de Estandarización de Telecomunicaciones Europeas (ETSI TS 07.10).

- El protocolo RFCOMM es un transporte sencillo que soporta hasta 60 conexiones simultáneas (canales de RFCOMM) entre dos dispositivos *Bluetooth*.
- RFCOMM sustituye al cable serie, ya que emula las señales de control y datos EIA-232 sobre la banda base y proporciona capacidad en el transporte a los servicios de niveles superiores.
- El protocolo RFCOMM soporta aquellas aplicaciones que utilizan el puerto serie de los dispositivos en donde se ejecutan.
- Existen dos tipos de dispositivos que RFCOMM puede acoplar:
Tipo 1: dispositivos que comunican hacia un punto final, por ejemplo: computadoras e impresoras.
Tipo 2: dispositivos que se encuentran como parte de una línea de comunicación, por ejemplo: el módem.

Capa SDP

La capa Protocolo de Descubrimiento de Servicios (SDP) realiza dos operaciones que están encaminadas en el descubrimiento de servicios en los dispositivos *Bluetooth*:

- *Búsqueda de servicios:* es la operación en la cual se busca un servicio específico en un dispositivo *Bluetooth*.
- *Enumeración de servicios:* es la operación en la que un dispositivo *Bluetooth* muestra los servicios que está ofreciendo.

Un servicio es la actividad o trabajo que puede realizar un dispositivo *Bluetooth*, por ejemplo: una impresora con tecnología *Bluetooth* puede ofrecer el servicio de imprimir información.

Clases de servicios SDP. Las clases de servicios que son soportados por dispositivos *Bluetooth* son los siguientes:

- Posicionamiento: identificación de localización.
- Redes: LAN, etc.
- Interpretación: impresora, bocinas, etc.
- Captura: *scanner*, micrófono, etc.
- Transferencia de objetos: *v-inbox*, *v-folder*, etc.

- Audio: bocinas, micrófono, servicio de audífonos, etc.
- Telefonía: telefonía inalámbrica, módem, etc.
- Información: servidor WEB, servidor WAP, etc.

Los dispositivos *Bluetooth* incorporan en la cabecera de sus paquetes de datos en la banda base un espacio que contiene información de sus clases o tipos de servicios. Conociendo las clases de servicios de un dispositivo *Bluetooth*, se puede determinar el conjunto de servicios genéricos o específicos que soporta dicho dispositivo.

Grabación de servicio. Toda la información, relacionada con un servicio que proporciona un dispositivo *Bluetooth*, está contenida en un registro individual de servicio. Un servicio grabado es una lista de atributos que describen las características de un servicio, y son los siguientes:

- Nombre del servicio
- Descripción del servicio
- Nombre del proveedor
- Gestor de grabación del servicio
- Lista identificadora de clase de servicio
- Estado de grabación de servicio
- Identificador de servicio
- Lista de descripción de protocolo
- Lista que agrupa numeración
- Lista identificadora de atributos de lenguaje base
- Tiempo de duración de servicio
- Disponibilidad de servicio
- Lista descriptora de perfil de *Bluetooth*

Capa OBEX

La capa Intercambio de Objetos (OBEX) es un protocolo de sesión desarrollado por la asociación IrDA con el nombre de IrOBEX, donde su objetivo es intercambiar objetos, tales como archivos, carpetas, etc., en forma sencilla y espontánea.

- Define un objeto llamado lista de archivos, el cual es usado para investigar el contenido de archivos en dispositivos remotos.
- Se basa en el modelo cliente/servidor y es independiente del transporte, aunque en la implementación de OBEX en la especificación sólo se utiliza RFCOMM como medio de transporte.

Formatos de contenido vCard y vCal

Los formatos de contenido *vCard* y *vCal* son especificaciones abiertas y ahora controladas por el consorcio de correos de *Internet*, estas especificaciones definen el formato de una tarjeta electrónica de negocios y entrada de calendarios personales e información de organización, respectivamente. *vCard* y *vCalendar* no definen tipo de transporte, únicamente el formato donde los datos son transportados.

Otros formatos contenidos, los cuales son transferidos por OBEX en dispositivos *Bluetooth*, son: *vMessage* y *vNote*. Estos formatos contenidos de igual manera son estándares abiertos y se utilizan en el intercambio de mensajes y notas. Los formatos anteriores son definidos en la especificación de Comunicaciones Móviles de Infrarrojos (IrMC), la cual define un formato para archivos de notas que se necesita cuando se sincronizan datos entre dispositivos *Bluetooth*.

Capa PPP

La capa Protocolo Punto a Punto (PPP) es diseñada para funcionar sobre el protocolo RFCOMM y realizar una conexión o enlace punto a punto. Es un protocolo utilizado como medio de acceso a redes. El protocolo PPP define la transmisión de datos de *Internet* (IP) sobre enlaces punto a punto, garantiza la interoperabilidad de dispositivos *Bluetooth* con aplicaciones basadas en protocolos TCP y UDP.

Capa TCP/UDP/IP

Los protocolos estándar TCP/UDP/IP son definidos y usados comúnmente en comunicaciones a través de *Internet*, es decir, el uso de estos protocolos permite la comunicación con cualquier otro dispositivo conectado a *Internet* con especificaciones de *Bluetooth*, por ejemplo: impresoras, computadoras, teléfonos celulares, etc.

Capa WAP

La capa Protocolo de Acceso Inalámbrico (WAP) proporciona un protocolo para que un dispositivo *Bluetooth* realice una conexión hacia otro en forma inalámbrica, por ejemplo proporciona funciones como el control remoto y realiza una búsqueda de datos desde la PC hacia otro dispositivo *Bluetooth*.

Capa TCS BIN

La capa Protocolo de Especificación Binaria y Control de Telefonía (TCS BIN) es un protocolo de *bit* orientado, proporciona la definición de señalización en el control de una llamada telefónica para establecer una conversación y transferir información entre dispositivos *Bluetooth*.

Capa Comandos AT

Los comandos de atención (AT) son instrucciones codificadas que conforman un lenguaje de comunicación entre un usuario y un módem telefónico, se denominan de esta manera por la palabra *Attention*.

Los comandos AT fueron desarrollados en 1977 por Dennis Hayes como una interfaz de comunicación con un módem, para configurarlo y así suministrarle instrucciones. Posteriormente las compañías *Microcomm* y *US Robotics* continuaron su desarrollo, hasta convertirlo en un estándar universal.

La telefonía celular con Sistema Global para Comunicaciones Móviles (GSM) también ha adoptado este estándar para la comunicación con sus terminales, así que, todos los teléfonos con tecnología GSM contienen un conjunto de comandos especiales que actúan como interfaz y proporcionan instrucciones a las terminales.

II.4.2.3. Perfiles de *Bluetooth*

Los perfiles de *Bluetooth* definen los protocolos y características que soporta un modelo de uso particular, o sea, una aplicación, esto es para asegurar la interoperabilidad entre dispositivos *Bluetooth* de diferentes fabricantes. Si cumplen con la misma especificación del perfil, se espera que interactúen correctamente en ese uso particular.

El perfil define mensajes específicos y procedimientos utilizados para implementar características, algunas de éstas son obligatorias y otras pueden ser opcionales. Los perfiles definen los roles y capacidades para aplicaciones específicas. Diferentes perfiles pueden abarcar diferentes niveles y protocolos para diferentes grados de seguridad.

En el siguiente cuadro (Figura II.32) puede verse como se encuentran estructurados los perfiles que se definen en el estándar *Bluetooth*.

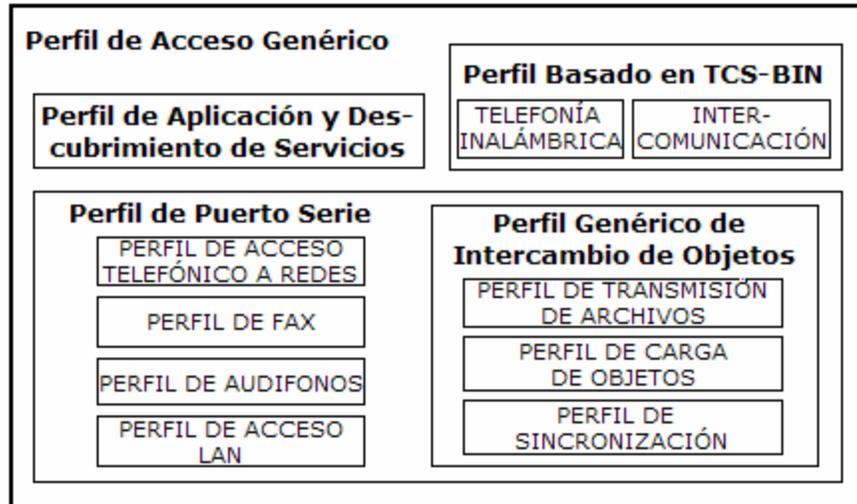


Figura 11.32. Estructura de los perfiles de Bluetooth.

Se definen cuatro perfiles genéricos o generales que contienen la definición de los perfiles específicos:

- Perfil de Acceso Genérico (GAP)
- Perfil de Puerto Serie (SPP)
- Perfil Genérico de Intercambio de Objetos (GOEP)
- Perfil de Aplicación y Descubrimiento de Servicios (SDAP)

Perfil de Acceso Genérico

El perfil GAP define las siguientes especificaciones:

- Procedimientos generales para descubrir los dispositivos *Bluetooth*.
- Los procedimientos de administración de enlace para establecer una conexión entre dispositivos *Bluetooth*.

El perfil GAP debe implementarse en todo aquel dispositivo que contenga tecnología *Bluetooth* para asegurar una interoperabilidad básica y la coexistencia con otros dispositivos, sin importar la aplicación que soporten. Los dispositivos que además cumplan otro perfil, pueden emplear adaptaciones de los procedimientos genéricos.

Perfil de Puerto Serie

El perfil SPP se utiliza cuando se quiere sustituir el cable físico por tecnología inalámbrica *Bluetooth*. Este perfil está construido sobre el perfil GAP, y define como deben configurarse los dispositivos *Bluetooth*

para emular una conexión a través de un cable serie utilizando el protocolo RFCOMM.

Una aplicación puede ser ejecutada sobre cualquiera de dos dispositivos *Bluetooth* utilizando el puerto serie virtual, como si se conectara por medio de un cable físico en un puerto EIA-232.

Perfil Genérico de Intercambio de Objetos

El perfil GOEP se basa en el modelo cliente/servidor, un cliente carga o envía objetos de datos a un servidor mediante la operación PUT del protocolo OBEX, o bien descarga o recibe objetos de datos desde un servidor mediante la operación GET del protocolo OBEX. El perfil GOEP define como deben soportar los dispositivos *Bluetooth* las aplicaciones de intercambio de objetos.

Perfil de Aplicación y Descubrimiento de Servicios

El perfil SDAP describe las características y procedimientos utilizados para descubrir servicios registrados en otros dispositivos *Bluetooth*, así como obtener información de esos servicios. El perfil SDAP utiliza el protocolo SDP para localizar los servicios disponibles en dispositivos situados dentro de su radio de enlace.

Sobre los perfiles genéricos (GAP, SPP, SDAP y GOEP) descritos anteriormente se definen los diferentes perfiles específicos para las aplicaciones. Los perfiles de aplicación son múltiples y variados y se presentan de manera opcional e independiente por cada fabricante y para cada tipo de dispositivo.

Los siguientes perfiles para aplicaciones que se definen están especificados dentro de la tecnología *Bluetooth* versión 1.0.

Perfil de Acceso Telefónico a Redes. Este perfil define protocolos y procedimientos usados por dispositivos, tales como el módem y teléfonos celulares para implementar una aplicación denominada puente de *Internet*. La aplicación más común es utilizar el teléfono celular como módem inalámbrico para conectar una PC a un servicio de acceso telefónico a *Internet*.

Perfil de Fax. Define los protocolos y procedimientos utilizados por dispositivos que implementan la parte de fax en la aplicación llamada Acceso a Redes Inalámbricas (WAN). Una computadora puede utilizar un

dispositivo *Bluetooth* (tal como un teléfono celular) como un dispositivo fax inalámbrico para recibir y enviar mensajes de fax.

Perfil de Audífonos. Define los protocolos y procedimientos para aplicaciones que permiten utilizar un dispositivo auricular como interfaz de entrada y salida de audio hacia otro dispositivo, tal como una PC o un teléfono celular.

Perfil de Acceso a Red. Este perfil define los protocolos y procedimientos para que un dispositivo *Bluetooth* pueda acceder a los servicios de una Red de Área Local (LAN). Este perfil utiliza el protocolo PPP sobre el protocolo RFCOMM.

Perfil de Transmisión de Objetos. Perfil que define los protocolos y procedimientos para soportar la aplicación de transferencia de objetos a través del protocolo OBEX, el cual tiene la capacidad de trasladar objetos de datos (archivos y carpetas) de un dispositivo *Bluetooth* a otro, así como navegar por el contenido de las carpetas del dispositivo remoto.

Perfil de Carga de Objetos. Es el perfil que define los protocolos y procedimientos para tener la capacidad de cargar y descargar objetos de datos de un dispositivo *Bluetooth* a otro, también se utiliza para la transferencia rápida de archivos. Este perfil está limitado a cierta clase de objetos, tal como *vCards*.

Perfil de Sincronización. Es el perfil que define los protocolos y procedimientos utilizados por las aplicaciones de sincronización de datos entre dos dispositivos *Bluetooth* habilitados.

II.4.2.4. Seguridad en *Bluetooth*

Bluetooth es una tecnología que contiene sistemas de seguridad, lo que la convierte en una tecnología inalámbrica de comunicaciones muy fiable y robusta ante ataques que se pudieran dar al contenido de información en un dispositivo. Las especificaciones de seguridad se realizan en los siguientes niveles de protocolos:

- A nivel de banda base
- A nivel de enlace

Seguridad a nivel de banda base. *Bluetooth* emplea la técnica de salto de frecuencias FHSS, estudiada anteriormente, como elemento de seguridad.

Seguridad a nivel de enlace. En el nivel de enlace existen tres técnicas de seguridad:

- Autenticación
- Autorización
- Encriptación

La autenticación: es un proceso en donde un dispositivo se identifica con otro para poder tener acceso a los servicios que este puede ofrecer. La seguridad en el nivel de enlace define todas sus funciones a través de claves de enlace. Las claves de enlace son números pseudoaleatorios de 128 *bits* almacenados en dispositivos *Bluetooth*, así que, esta función se ejecuta para autenticar dispositivos.

El emparejamiento es un procedimiento de inicialización entre dos dispositivos que realizan su primer intento de comunicación y se crea una clave de enlace común.

En la primera conexión de los dispositivos, el procedimiento de emparejamiento requiere que el usuario de cada dispositivo *Bluetooth* introduzca un código de seguridad de hasta 16 *bytes* y debe ser el mismo en ambos dispositivos, es decir, un primer usuario introduce el código de seguridad y el segundo confirma el mismo código.

La autorización: es el proceso en el cual se determinan los derechos que tiene un dispositivo *Bluetooth* para tener acceso a los servicios que presta un sistema. La autorización se lleva a cabo a través de niveles de confianza, los cuales especifican la capacidad de acceso a los servicios:

- Acceso a los servicios en forma total
- Acceso a los servicios en forma parcial
- Acceso a los servicios en forma restringida o nula

La encriptación: protege la información que se transmite en el enlace de dos dispositivos *Bluetooth*. Así se asegura la confidencialidad de la información, de tal manera que si un usuario captura el paquete de datos y no tiene la clave para desencriptar, entonces no podrá descifrar el contenido de la información.

Rutina de Encriptado Rápido y Seguro SAFER+. Este proceso de seguridad es un algoritmo simétrico de encriptado de datos y utiliza bloques de 128 *bits*, el algoritmo se utiliza cuando se generan las claves de autenticación y de encriptamiento. Los dispositivos *Bluetooth* utilizan el algoritmo

4LSFR, que es un encriptador de flujo adecuado para encriptar datos en forma rápida.

II.5. Lenguaje de alto nivel

El equipo de cómputo necesita de un *software* para poder comunicarse e intercambiar información con un periférico externo, tal como un adquisidor de datos. El *software* que se mencionará es un lenguaje de programación llamado *Visual Basic*, tal lenguaje posee las características siguientes: compatibilidad con la plataforma *Windows*, así como instrucciones para el manejo de comunicaciones seriales.

El lenguaje *Visual Basic* es considerado un lenguaje de programación visual, también se le denomina lenguaje de cuarta generación; esto significa que una gran cantidad de tareas pueden realizarse sin la necesidad de escribir código.

Visual Basic es un lenguaje de programación para aplicaciones de *Windows* de *Microsoft*, este lenguaje está diseñado específicamente para crear aplicaciones con interfaz gráfica de forma rápida y sencilla. *Visual Basic* se desarrolla bajo dos tipos de objetos: ventanas y controles.

Para realizar una aplicación se crean ventanas, llamadas formularios, y sobre ellas se dibujan objetos, denominados controles; estos pueden ser: imágenes, etiquetas, cajas de texto, botones de órdenes, botones de selección, cuadros de verificación, listas desplegadas, etc. Realizado lo anterior se desarrolla un código fuente, el cual se relaciona con cada objeto (ventanas y controles) que permanece inactivo hasta que un suceso lo activa.

Una característica importante en este lenguaje de programación es que proporciona una interfaz con un conjunto estándar de comandos de comunicaciones, permitiendo establecer una conexión con un puerto de comunicación serie.

En este capítulo se ha visto lo más importante de la teoría que se involucrará en este proyecto, se continuará con el diseño y el desarrollo del mismo.

CAPÍTULO III

DISEÑO Y DESARROLLO DEL SISTEMA

En el presente capítulo se va a tratar la implementación de los componentes que integran el termopluviógrafo digital, así como los diferentes módulos de comunicación que conforman este dispositivo. De la misma manera se describirán el *hardware* y el *software* diseñados y desarrollados.

III.1. Descripción del sistema

El termopluviógrafo que se va a desarrollar tiene como principal objetivo el registro de la cantidad de precipitación pluvial y de la temperatura ambiente. Las principales características que debe obedecer el sistema son las siguientes:

- Las señales que entrarán al sistema de adquisición son las provenientes del pluviómetro tipo balancín y la del sensor de temperatura, las cuales serán acondicionadas para evitar posibles errores.
- El cerebro del sistema será un microcontrolador, que se encargará de realizar el control de todas las actividades, tales como: el registro de la temperatura y la hora en que sucede un evento de precipitación, así como almacenar estos datos en una memoria no

volátil, presentar la información en un visualizador alfanumérico al usuario, realizar la transferencia de la información hacia un equipo de cómputo y administrar el consumo de energía al sistema.

- El módulo de almacenamiento de información podrá registrar hasta 20,000 datos, que corresponderán a la precipitación y a la temperatura, y será habilitado al momento de presentarse un evento o bien cuando se transfieran datos hacia el equipo de cómputo.
- Contendrá un visualizador que permitirá al usuario acceder a la información almacenada en el sistema de adquisición, este visualizador empleará tecnología de fabricación de bajo consumo de potencia y se interactuará con él a través de un teclado.
- Contará con un teclado, mediante el cual se tendrá la capacidad de modificar ciertos parámetros de operación del sistema.
- El sistema tendrá una autonomía de funcionamiento aproximadamente de 6 meses, por lo que la tecnología de fabricación de los componentes empleados será de bajo consumo de potencia. Esta característica se debe a que el sistema operará con sólo una batería y se requiere su máxima duración. La fuente de alimentación estará constituida por una pila tamaño C de 1.5 Volts. Se requerirá de un circuito elevador de voltaje, por lo que se utilizará un módulo de elevación, que proporcionará el voltaje necesario al sistema para su correcta operación.
- El sistema contará con un módulo de comunicación, que tendrá la capacidad de transferir los datos almacenados en la memoria hacia un equipo de cómputo de forma alámbrica e inalámbrica, a través de los estándares RS-232, USB e IrDA y *Bluetooth*.
- Al sistema se le incorporará un programa que se utilizará como interfaz entre el usuario, el equipo de cómputo y el termopluviógrafo digital.

Todas las características enunciadas anteriormente se presentan en la figura III.1.

III.2. Diseño y desarrollo de los módulos

Los módulos a diseñar y desarrollar son los correspondientes a: el pluviómetro de balancín, de temperatura, microcontrolador, almacenamiento de datos, despliegue, teclado y alimentación de energía.

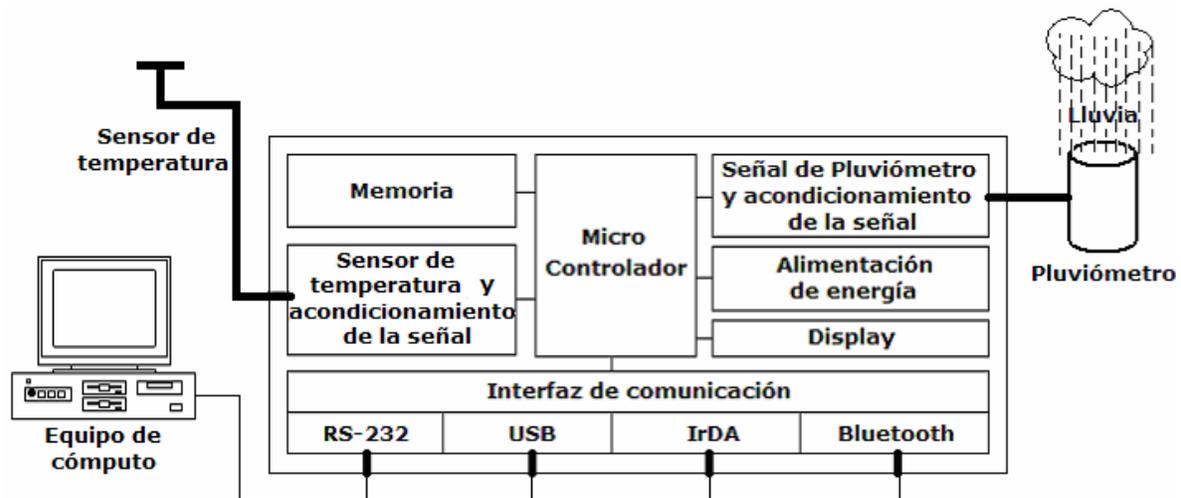


Figura III.1. Diagrama en bloques del sistema de medición.

Pluviómetro tipo balancín

El dispositivo más común para obtener mediciones pluviales es el pluviómetro tipo balancín mostrado en la figura III.2. Este tipo de pluviómetro es el que se utilizará en este proyecto y su valor de medición calibrado en fábrica es de 0.5 mm de registro de lluvia.

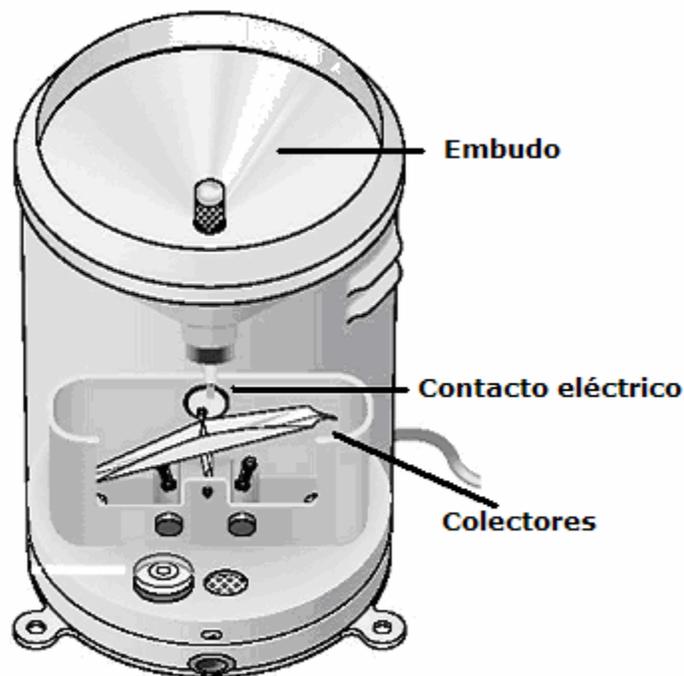


Figura III.2. Pluviómetro tipo balancín.

Como se recordará, el pluviómetro de balancín tiene dos colectores de agua calibrados que se encuentran suspendidos sobre un eje de rotación a manera de balanza. Mediante un embudo, la lluvia colectada es conducida hacia uno de los colectores. Cuando uno de los colectores se llena, el peso del agua hace que por gravedad el balancín caiga vaciando su contenido, colocando al otro colector en posición para recibir agua. Así sucesivamente se van llenando ambos recipientes alternándose mientras continúa la lluvia.

Cada vuelco del balancín (colectores) se registra como un evento, la información generada se transmite hacia el sistema de registro. Esto se logra a través de un acoplamiento mecánico entre el balancín y un interruptor de mercurio, a partir del cual se genera y detecta el evento.

La señal del evento proveniente del pluviómetro requiere de un acondicionamiento, esto es debido a que después de que ocurre un evento existen rebotes mecánicos u otras características entre las superficies de contacto, las cuales se presentan cuando se vuelca el balancín. Si no es acondicionada la señal, el microcontrolador puede considerar dichos rebotes como señales de eventos y registrarlos, dando como resultado eventos falsos.

El acondicionamiento de la señal se realiza por medio de *hardware* (Figura III.3), donde se utiliza un filtro RC para asegurar que sólo un pulso por evento se va a registrar. También se colocan dos diodos para obtener una protección de sobrevoltaje, el cual podría dañar la terminal de entrada del microcontrolador. Posteriormente se hace una conexión hacia una compuerta con entradas de disparador Schmitt, este circuito utiliza retroalimentación interna, desplazando el umbral de conmutación, el cual es conocido como histéresis, dependiendo de si la entrada está cambiando de bajo a alto o viceversa. La característica de este tipo de circuito es que da una protección adicional con respecto al ruido.

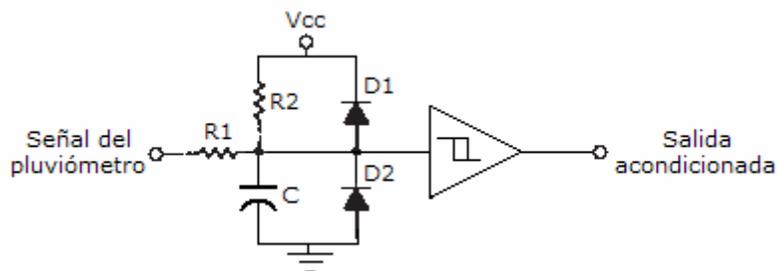


Figura III.3. Acondicionamiento de la señal del pluviómetro.

Sensor de temperatura basado en semiconductor

Regularmente los sensores de temperatura basados en semiconductor presentan en la salida una señal de voltaje analógica, cuyo valor es una función lineal de la temperatura. Los sensores de temperatura basados en semiconductor están disponibles en diferentes escalas de medición, como pueden ser: en grados Kelvin (K), en grados Celsius ($^{\circ}\text{C}$) y en grados Fahrenheit ($^{\circ}\text{F}$).

Para este proyecto se eligió un sensor de temperatura de semiconductor, cuyo voltaje de salida es linealmente proporcional a la temperatura en grados Celsius, algunas de sus características se muestran en la tabla III.1.

Característica	Descripción
Intervalo de medición	-55 a +150 $^{\circ}\text{C}$
Voltaje de polarización	De 4 a 30 Volts
Máximo de corriente drenada	60 μA
Factor de linealidad	+10.0 mV/ $^{\circ}\text{C}$
Exactitud de medición	0.5 $^{\circ}\text{C}$ (a 25 $^{\circ}\text{C}$)
Salida de baja impedancia	0.1 Ω para una carga de 1 mA

Tabla III.1. Características del sensor de temperatura.

La medición de la temperatura requiere de una etapa de acondicionamiento. Dicho acondicionamiento consiste en agregar un filtro RC al dispositivo sensor de temperatura (Figura III.4), con el objetivo de estabilizar la señal que pueda presentarse inestable debido a perturbaciones internas o externas al circuito, tal como el ruido. El sensor de temperatura proporciona una señal analógica, por lo que se requiere para su posterior procesamiento un convertidor analógico-digital.

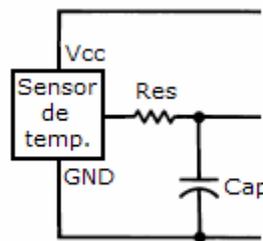


Figura III.4. Acondicionamiento de señal del sensor de temperatura.

Microcontrolador

Antes de seleccionar el microcontrolador que se utilizó en este proyecto, se hizo un análisis comparativo entre los diferentes microcontroladores

que existen en el mercado, así como de diferentes fabricantes. Entre los fabricantes de los microcontroladores estaban: *Nacional Semiconductor*, *Microchip*, *Motorola* y *Atmel*. Como resultado de este estudio se observó que, las características, los precios, los atributos, etc., eran semejantes. Pero considerando que en el Instituto de Ingeniería se tiene experiencia y se cuenta con un sistema de desarrollo basado en los microcontroladores de la familia de los AVR de la empresa *Atmel*, se optó por este fabricante.

En el diseño y desarrollo del termoplúviógrafo digital haremos uso de un microcontrolador de la familia AVR, que posee las características que se muestran en la tabla III.2.

Características	Descripción
Memoria de programa <i>flash</i>	8 <i>kBytes</i>
Memoria de datos SRAM	512 <i>Bytes</i>
Memoria de datos EEPROM	512 <i>Bytes</i>
Microcontrolador AVR	Alto rendimiento de 8 <i>bits</i>
Periféricos	32 líneas de E/S programables
Convertidor analógico-digital	De 10 <i>bits</i> , 8 canales
Comparadores	1
Temporizadores/Contadores	2 de 8 <i>bits</i> 1 de 16 <i>bits</i> 1 contador de tiempo real (RTC) de 8 <i>bits</i> 1 <i>Watchdog</i> Timer
Entradas y salidas seriales	USART I ² C SPI
Velocidad máxima de operación	8 MHz
Modos de bajo consumo	Seis: <i>Idle</i> , <i>ADC Noise Reduction</i> , <i>Power-save</i> , <i>Power-down</i> , <i>Standby</i> and <i>Extended Standby</i>
Voltaje de operación	2.7 a 5.5 Volts
Arquitectura	RISC avanzada: 130 instrucciones 32 registros de trabajo de propósito general de 8 <i>bits</i>

Tabla III.2. Características del microcontrolador.

El microcontrolador seleccionado es de 8 *bits*, de baja potencia, con tecnología CMOS, basado en la arquitectura AVR RISC. Ejecuta las instrucciones en un simple ciclo de reloj.

En la figura III.5 se muestra un diagrama de bloques de dicho microcontrolador.

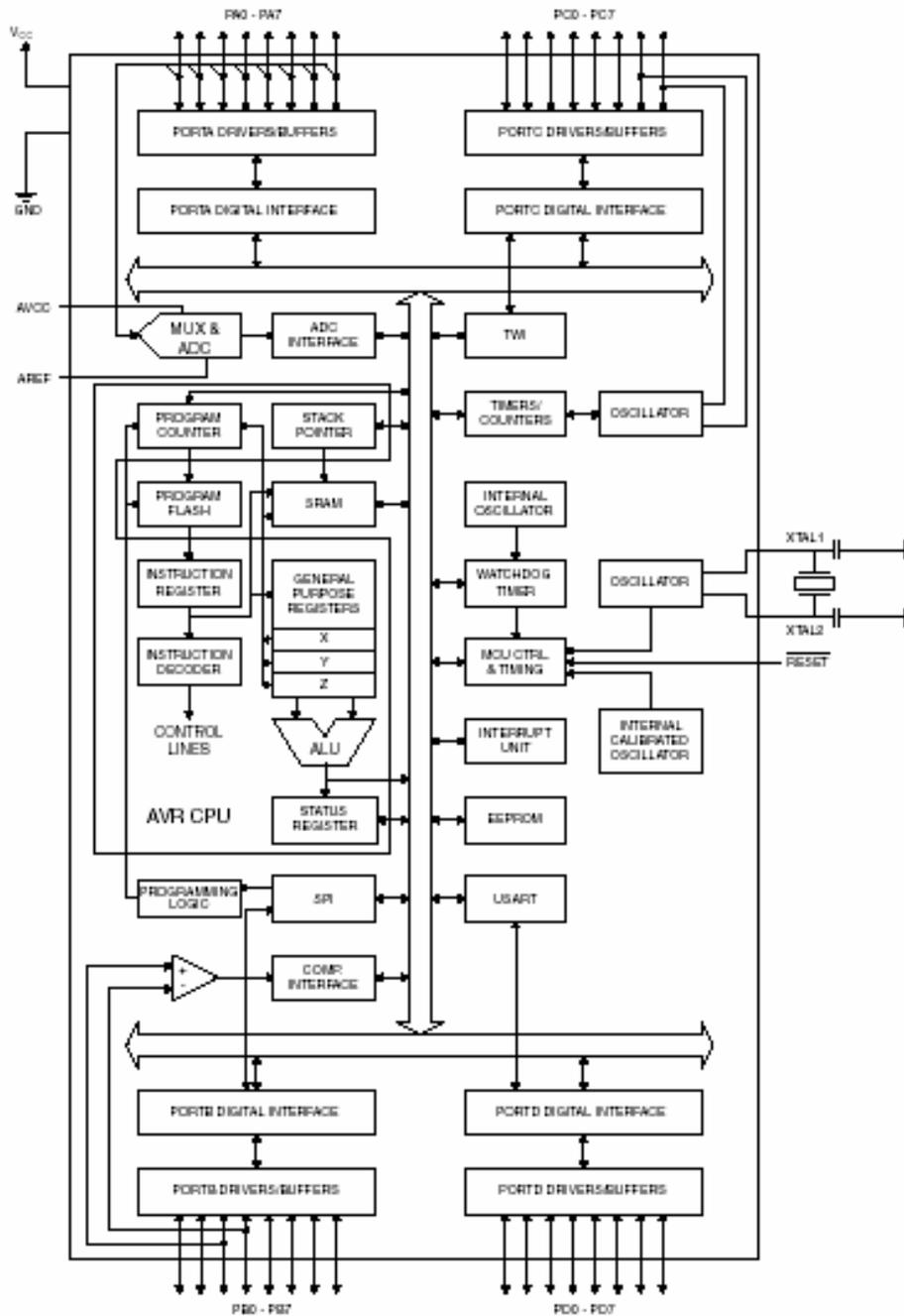


Figura III.5. Diagrama a bloques del microcontrolador.

La comunicación entre el microcontrolador y los dispositivos periféricos es a través de terminales de entrada/salida digitales, estas terminales reciben el nombre de puertos. Todos los puertos digitales tienen resistencias individuales llamadas *pull-up*, así como la capacidad de proporcionar hasta 20 mA de corriente en la salida, permitiendo manejar en forma directa *leds* y visualizadores. Estos puertos son programables por medio de instrucciones.

El microcontrolador cuenta con 32 terminales de E/S para controlar los periféricos, los cuales se encuentran distribuidos en cuatro puertos bi-direccionales de E/S, estos son: puerto A, puerto B, puerto C y puerto D, cada uno de 8 *bits*.

Los puertos de E/S pueden tener funciones alternas. Por ejemplo, el puerto A funciona como entrada analógica hacia el convertidor analógico-digital (ADC) interno del microcontrolador.

El puerto B también puede desempeñar funciones alternas, las cuales se describen en la tabla III.3.

Terminal	Función alterna
PB0	T0 (Entrada externa del temporizador/contador 0)
PB1	T1 (Entrada externa del temporizador/contador 1)
PB2	AIN0 (Entrada positiva del comparador analógico)
PB3	AIN1 (Entrada negativa del comparador analógico)
PB4	SS (Entrada de selección del dispositivo esclavo del bus serial SPI)
PB5	MOSI (Salida del maestro/entrada del esclavo del bus serial SPI)
PB6	MISO (Entrada del maestro/salida del esclavo del bus serial SPI)
PB7	SCK (Señal de reloj del bus serial SPI)

Tabla III.3. Funciones alternas del puerto B.

Una función alterna en el puerto C se presenta cuando se habilita el reloj asíncrono del *Timer/Counter2* para funcionar de manera asíncrona, donde las terminales PC6 y PC7 son desconectadas del puerto. En este modo un oscilador de cristal de 32.768 kHz es conectado a las terminales del puerto.

El puerto D cuenta con funciones alternas, las cuales se presentan en la tabla III.4.

Terminal	Función alterna
PD0	RXD (Terminal de entrada de la UART)
PD1	TXD (Terminal de salida de la UART)
PD2	INT0 (Entrada de la interrupción externa 0)
PD3	INT1 (Entrada de la interrupción externa 1)
PD4	OC1B (Salida de comparación B del temporizador/contador B)
PD5	OC1A (Salida de comparación A del temporizador/contador A)
PD6	ICP (Terminal de captura de entrada al contador 1)
PD7	OC2 (Salida del temporizador 2 en modo de comparación)

Tabla III.4. Funciones alternas del puerto D.

En la figura III.6 se muestra la distribución de terminales del microcontrolador a utilizar.

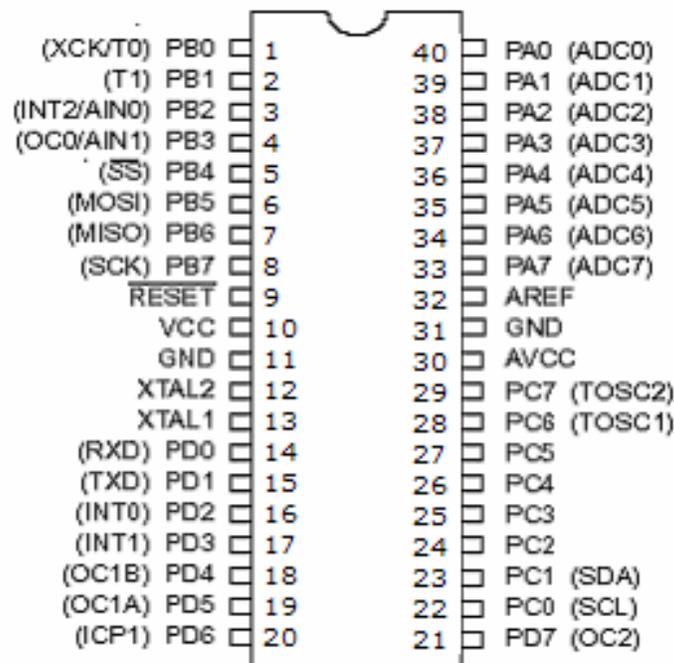


Figura III.6. Distribución de terminales del microcontrolador.

En la tabla III.5 se muestra una relación de las conexiones realizadas entre las terminales del microcontrolador y cada uno de los módulos que conforman el sistema de adquisición de datos.

Módulo conectado	Terminales del microcontrolador
Sensor de temperatura	PA1
Pluviómetro	PD2
Memorias EEPROM	PB1 a PB7
Visualizador	PA4 a PA7 y PC3 a PC5
Teclas de operación	PD3 a PD7
Suministro de energía	VCC
Interfaz de comunicación	PD0 y PD1

Tabla III.5. Interconexión de terminales.

El sistema de desarrollo AVR (Figura III.7) que se dispone en el Instituto de Ingeniería, lo usamos en el diseño y desarrollo de este proyecto y se empleó principalmente en: la realización de pruebas en el diseño de la transmisión de datos entre el microcontrolador y la interfaz RS-232, pero fundamentalmente se utilizó en la grabación del programa principal en ensamblador introducido al microcontrolador.

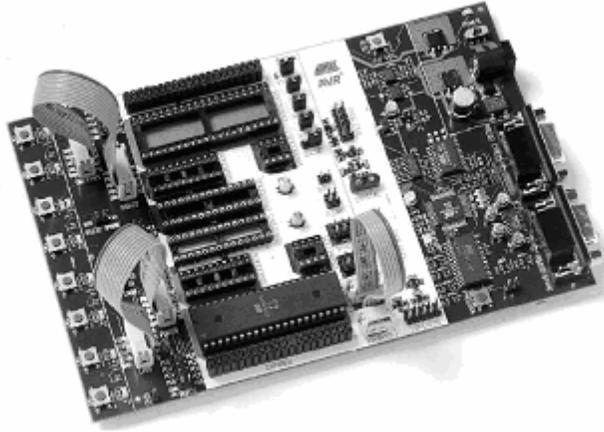


Figura III.7. Sistema de desarrollo AVR.

Las características principales de este *kit* de pruebas son:

- Cuenta con la interfaz RS-232
- Contiene zócalos (*sockets*) de 8, 20, 28 y 40 terminales para dispositivos AVR.
- 8 botones y 8 *leds* de propósito general
- Puertos de E/S
- Alimentación regulada de 10 a 15V

Módulo de almacenamiento de datos

Los sistemas digitales tienen la capacidad de almacenar grandes cantidades de información. En este proyecto haremos uso de memorias EEPROM seriales, que son reprogramables y empleadas principalmente en configuración y programación de datos en diferentes aplicaciones. Su característica principal es mantener los datos almacenados aún si la energía eléctrica falla. Otra de sus características es su operación con bajo voltaje, por consecuencia reduce el consumo de potencia, por lo tanto, estas memorias se pueden usar en sistemas donde su fuente de alimentación es con baterías.

Las memorias que se van a emplear operan con el protocolo de comunicación serie síncrono (SPI), el cual es ampliamente utilizado en el entorno del microcontrolador. En la figura III.8 se muestra la interconexión de las memorias con el microprocesador. El intercambio de datos se lleva a cabo de la siguiente manera: la EEPROM es habilitada por el microcontrolador a través de una terminal de selección del circuito integrado (CS) y se tiene acceso a ella por medio de una interfaz de tres terminales, que consisten en una entrada de datos

seriales (SI), una salida de datos seriales (SO) y una señal de reloj (SCK).

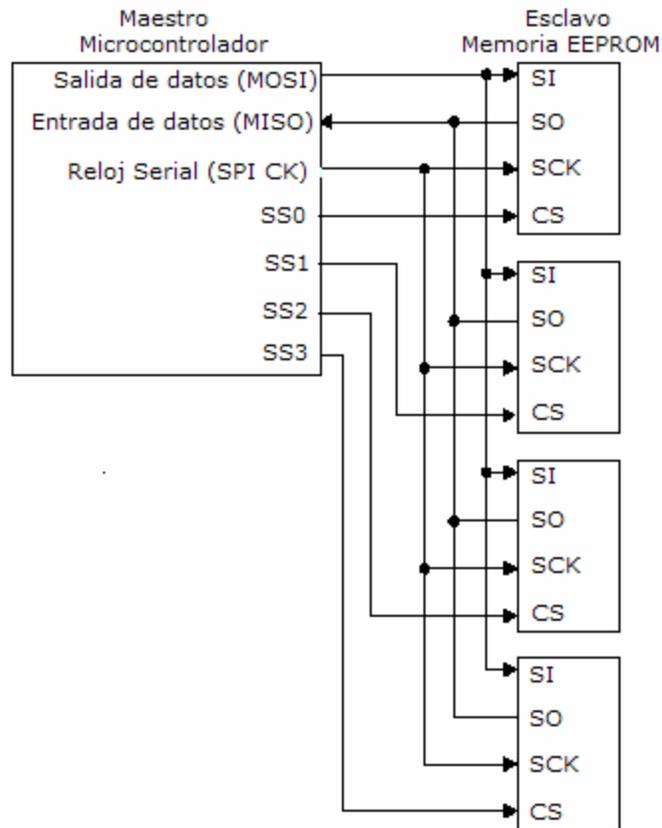


Figura III.8. Interconexión de las memorias con el microcontrolador.

Las memorias que se utilizaron en el proyecto son de la empresa *Atmel*, cuyas características se muestran en la tabla III.6.

Característica	Descripción
Interfaz de soporte	Interfaz de Periférico Serial (SPI)
Voltaje de operación	2.7 a 5.5 Volts
Protección por <i>hardware</i>	Terminal de protección de escritura (WP)
Protección por <i>software</i>	Instrucciones para deshabilitar escritura
Tiempo de retención de datos	100 años
Capacidad de almacenamiento	8192 <i>bytes</i> de 8 <i>bits</i> cada uno (65536 <i>bits</i>)

Tabla III.6. Características de la memoria EEPROM.

El esquema del circuito integrado de *Atmel* que representa una memoria EEPROM se muestra en la figura III.9.

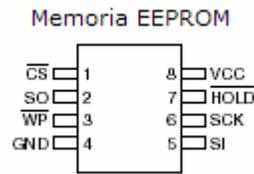


Figura III.9. Esquema de memoria EEPROM.

Módulo de despliegue y teclas operación

El sistema de medición permite mostrar ciertos datos de operación, tales como la hora, la fecha, el número de eventos acumulados, entre otros. Esta cualidad se logra a través de un visualizador alfanumérico de cristal líquido y las teclas de operación. El visualizador que se utilizó para el proyecto es un módulo integrado de cristal líquido de 2 líneas y 16 caracteres por línea (Figura III.10).

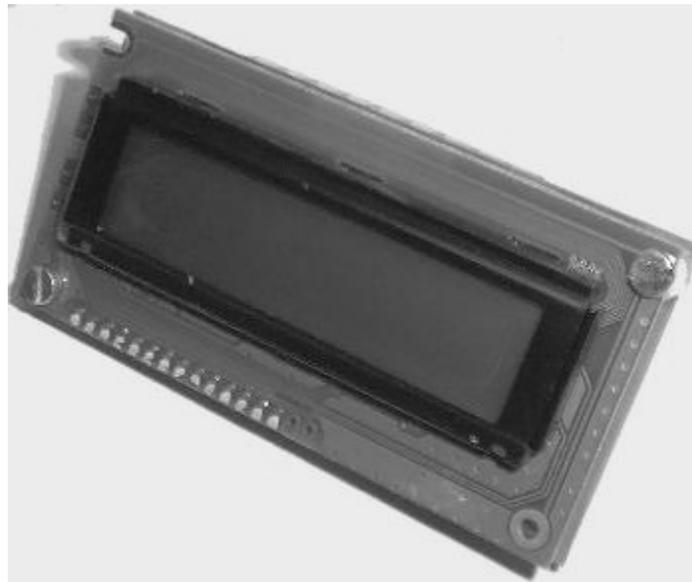


Figura III.10. Visualizador alfanumérico de 2 líneas X 16 caracteres.

El visualizador tiene integrados los circuitos necesarios para su operación y no se necesita de algún otro componente para ello. Las características de este visualizador se muestran en la tabla III.7.

Características	Descripción
Voltaje de alimentación	5 Volts
Total de caracteres desplegados	32

Tabla III.7. Características del visualizador. (Continúa)

Terminales utilizadas	4 para datos y 3 para control
Comandos para control	11
Interfaz hacia microcontrolador	4 u 8 <i>bits</i>

Tabla III.7. Características del visualizador.

La interfaz del microcontrolador con el visualizador se hizo con 4 *bits*, debido a que de esta forma la cantidad de líneas por utilizar en el microcontrolador es menor.

El termoplúviógrafo digital contiene 5 teclas de operación, las cuales son del tipo *push-button* (PB), (Figura III.11).



Figura III.11. Tecla de operación tipo *push-button*.

La conexión de las teclas se realiza de manera directa hacia las terminales del microcontrolador, utilizando las resistencias *pull-up* de éste. Las funciones que realizan las teclas de operación se muestran en la tabla III.8.

Tecla de operación	Función
PB1	Muestra el cursor para modificar los parámetros.
PB2	Selecciona en forma cíclica las opciones del menú del registrador en el modo de visualización.
PB3	Habilita el visualizador y también se utiliza para incrementar en uno el valor de un dato.
PB4	Cada vez que se presiona recorre el cursor una posición hacia la derecha para modificar datos.
PB5	Acepta o cancela los datos modificados.

Tabla III.8. Funciones de las teclas de operación.

Alimentación del sistema

La alimentación del termoplúviógrafo digital se realiza a partir de una pila tipo C, de 1.5 Volts. Posteriormente este voltaje es proporcionado a un sistema de elevación de voltaje. Para este último efecto se utiliza un circuito elevador convertidor DC-DC, conocido como *step-up*. Este

circuito nos proporciona un voltaje fijo de 5 Volts en la salida, el cual es adecuado para que los demás circuitos trabajen correctamente.

Las características del circuito elevador de voltaje utilizado se muestran en la tabla III.9.

Características	Descripción
Voltaje de alimentación	De 0.7 a 5.5 Volts
Voltaje de salida	Fijo de 3.3 Volts Ajustable de 2.7 a 5.5 Volts
Corriente de suministro máxima	1.4 A
Eficiencia	Hasta del 95 %
Consumo en modo de ahorro de energía	150 μ W
Modos de operación	Dos: Modulación por frecuencia de pulso (PFM) Modulación por ancho de pulso (PWM)

Tabla III.9. Características del circuito elevador de voltaje.

Este circuito elevador de voltaje contiene dos modos de operación, los cuales son: el modo de modulación por pulso de frecuencia (PFM), en donde su consumo de potencia es de 150 μ W (este modo es el utilizado) y el modo de modulación por ancho de pulso (PWM), el cual funciona para una frecuencia fija de 300 kHz y consume más energía.

En la figura III.12 se muestra el circuito elevador de voltaje que proporciona la energía necesaria para el sistema.

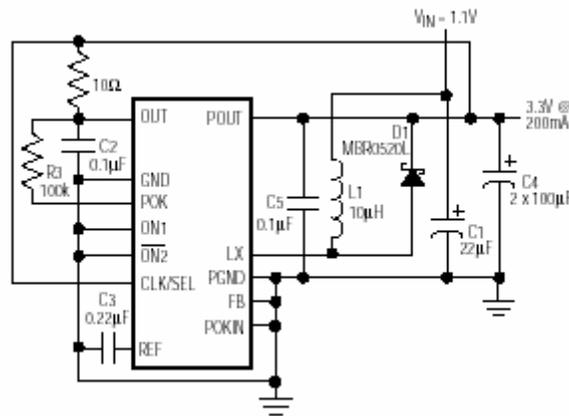


Figura III.12. Configuración del circuito elevador de voltaje.

Para controlar la característica de bajo consumo se necesita deshabilitar algunos circuitos. En el termoplúviógrafo digital, el dispositivo que consume más energía es el visualizador, por lo que hay que considerar un circuito de suministro de energía alterno. Dicho circuito se podrá

conectar/desconectar, para que cuando el visualizador no esté en uso no consume energía, esto sin afectar a los demás circuitos.

Las características del circuito utilizado se muestran en la tabla III.10.

Características	Descripción
Transistor de paso	Mosfet de canal P
Flujo de corriente regulada	11 μ A
Corriente a máxima carga	200 mA
Corriente en modo de ahorro de energía	Máximo de 1 μ A
Protección	Corriente inversa y sobrecarga térmica

Tabla III.10. Características del circuito proveedor de energía alterno.

El circuito mencionado anteriormente cuenta con una terminal *OFF*, la cual es controlada desde el microcontrolador a través de una de sus líneas, esto con el fin de deshabilitarlo, y lograr así una reducción en la demanda de corriente a menos de 1 μ A, desconectando de esta forma al visualizador. El circuito también cuenta con una terminal de detección de batería baja, con un limitador de corriente, protección contra corrientes inversas y protección contra sobrecarga. En la figura III.13 se muestra su configuración.

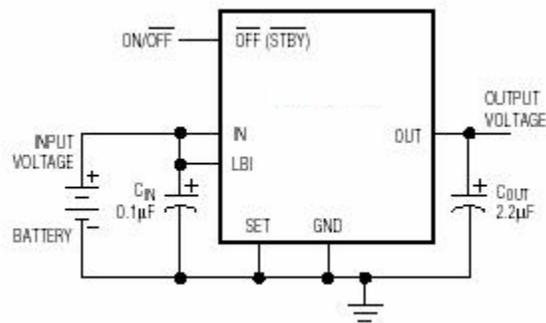


Figura III.13. Configuración del regulador lineal.

III.3. Módulo de comunicación alámbrica

En el presente trabajo se va a realizar la transferencia de información entre el termopluviógrafo digital y un equipo de comunicación PC, o una computadora portátil (*lap top*) por medio de los estándares alámbricos EIA-232 y USB.

En el diagrama de la figura III.14 se muestra un esquema de comunicación entre los dispositivos involucrados en el diseño.

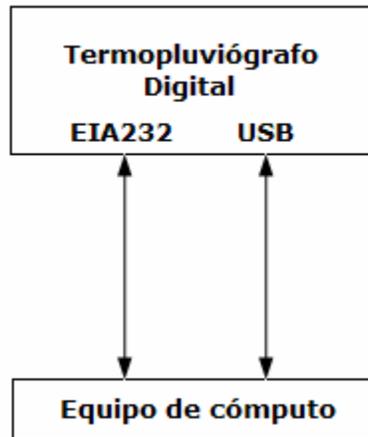


Figura III.14. Comunicación alámbrica.

Interfaz EIA-232

Como se ha visto, el diseño del termoplúviógrafo digital está basado en un microcontrolador que entre sus propiedades posee una interfaz de transmisión de datos seriales. Para la transferencia de información, el microcontrolador contiene integrada una UART, con base en la cual desarrolla la comunicación en forma serial a través de la interfaz EIA-232.

Para hacer compatibles los niveles lógicos de la UART con los del estándar EIA-232, se utilizó un circuito integrado como transceptor, (Figura III.15).

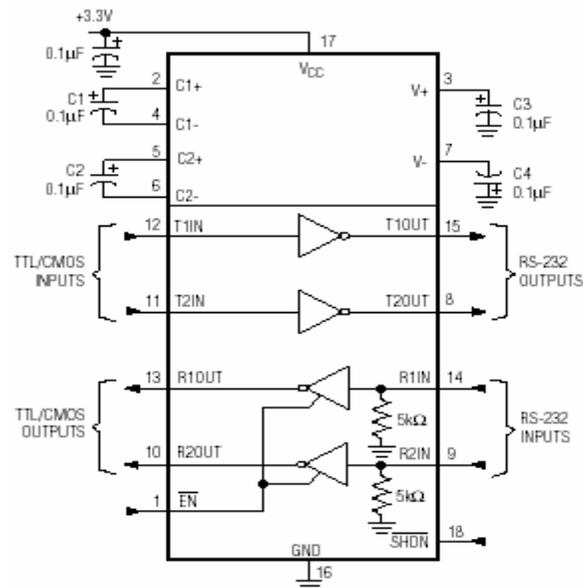


Figura III.15. Transceptor.

Para su operación el microcontrolador se programa en lenguaje ensamblador y con ello es capaz de controlar la comunicación serie del termoplúviógrafo digital hacia el equipo de cómputo.

La conexión para la transferencia de datos entre el termoplúviógrafo digital y el equipo de cómputo, a través de la interfaz EIA-232, se realiza como se muestra en la figura III.16.

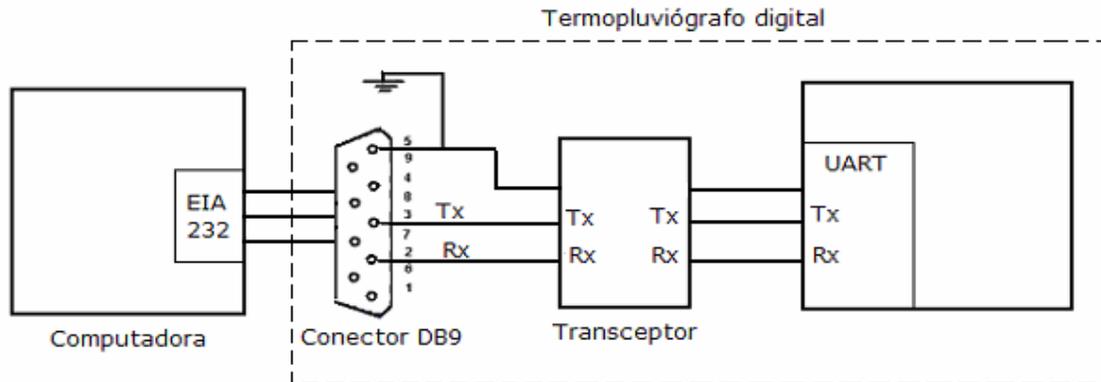


Figura III.16. Implementación de la interfaz EIA-232.

Interfaz USB

El diseño del circuito que permite comunicar el termoplúviógrafo digital hacia un equipo de cómputo, a través de una interfaz USB, se llevó a cabo utilizando un circuito integrado que fue desarrollado para dicho proceso.

En la tabla III.11 se reportan las características principales del circuito integrado utilizado, también denominado como convertidor.

Características	Descripción
Versión USB 2.0	Soporta 12 Mbps
Contiene un microcontrolador 8052	RAM de 256 X 8, para datos internos ROM de 10k X 8 RAM de 16k X 8, para espacio de código RAM 2k X 8, reservada para <i>buffer</i> de datos, configuración de paquetes de datos, y para bloques descriptores de punto final (EDB) <i>Watchdog</i> de 128 ms
Operación de cristal	12 MHz
Voltaje de operación	3.3 Volts
Control de flujo	Programable <i>XON/XOFF</i> Programable auto RTS/DTR y CTS/DSR

Tabla III.11. Características del convertidor. (Continúa)

Selección de estándar IrDA	Hasta 115.2 kbps
Baud rate seleccionable de software	De 50 a 921.6 kbps
Programación interfaz serie	5, 6, 7 o 8 bits de datos bit de detección de paridad Genera bit de paro de 1, 1.5 ó 2
Conversiones	USB a RS-232 USB a IrDA USB a RS-485

Tabla III.11. Características del convertidor.

En la figura III.17 se observa la arquitectura del circuito convertidor.

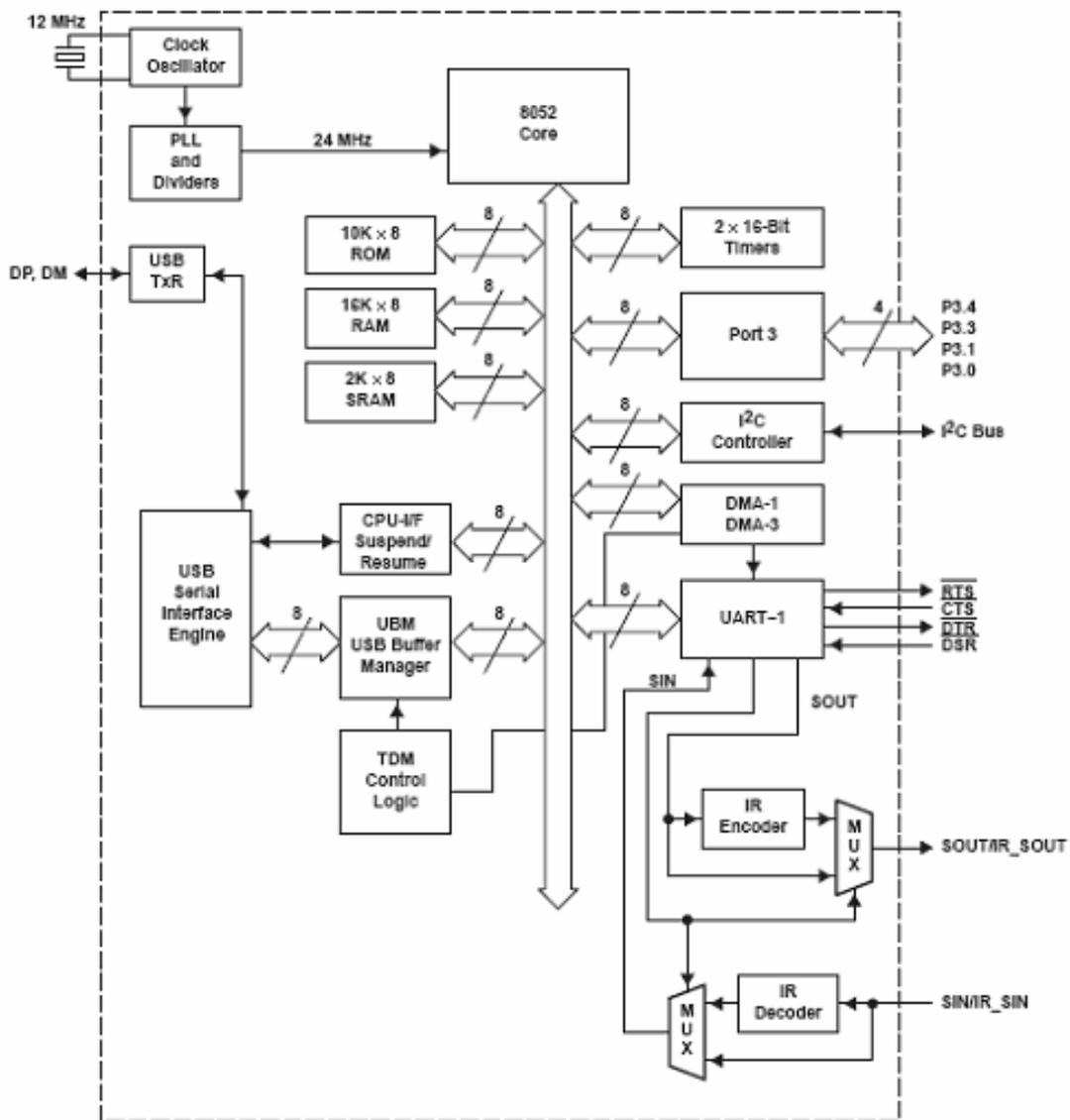


Figura III.17. Arquitectura del circuito convertidor.

El circuito convertidor cuenta con las siguientes funciones.

- Decodificación y codificación de las señales del convertidor USB a EIA-232 y EIA-232 a USB.
- Configuración de la UART.
- Manejo de errores que son administrados por el *firmware* (programa interno del circuito) del microcontrolador.

En la figura III.18 se muestra la conexión electrónica implementada del circuito convertidor.

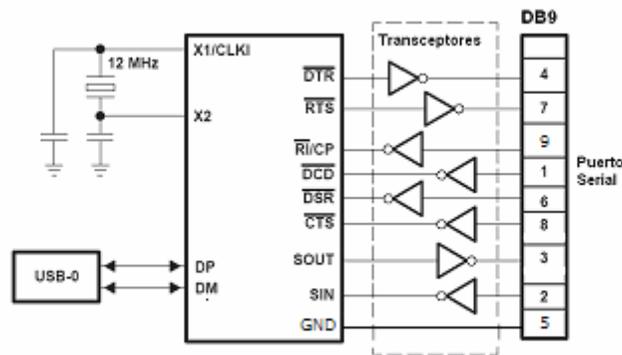


Figura III.18. Implementación del estándar USB a EIA-232.

Las terminales que se utilizan para la implementación del estándar USB a EIA-232 son: RTS, CTS, DTR y DSR, con sus transceptores para los niveles de voltaje. Las terminales de control son configuradas desde el microcontrolador. Las terminales SOUT y SIN corresponden al TxD y RxD del estándar EIA-232 respectivamente.

El circuito convertidor funciona de la siguiente manera: la información que es emitida desde el equipo de cómputo, a través de su puerto USB, es suministrada a la entrada USB del circuito convertidor, y después es enviada por medio de la línea SOUT al puerto serie del termoplúviógrafo digital. De la misma forma, los datos suministrados hacia la computadora, a través del circuito convertidor, son transmitidos desde el puerto serie del termoplúviógrafo digital por la terminal SIN.

La figura III.19 presenta la implementación de diseño del estándar USB que se llevó a la práctica.

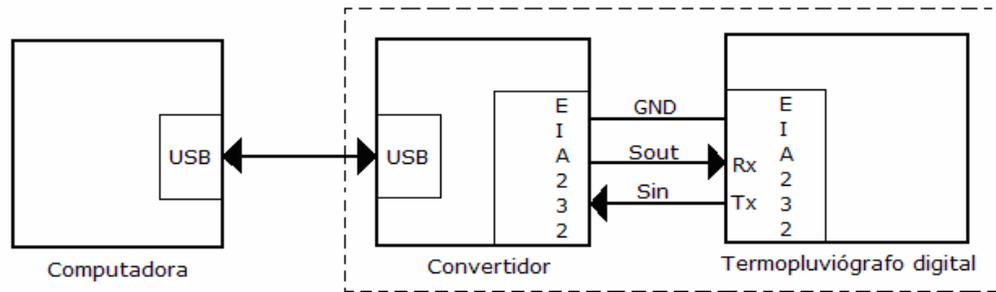


Figura III.19. Implementación de la interfaz USB.

El convertidor utilizado es de tipo bidireccional y contiene la lógica necesaria para comunicarse con una computadora a través de la interfaz USB. Una observación importante es que, para que el circuito funcione correctamente, es necesario suministrar un *software*, es decir unos *drivers* que se obtienen de la página www.ti.com.

III.4. Módulo de comunicación inalámbrica

Con la intención de evitar el uso de conexiones alámbricas, y hacer que el intercambio de datos fuese más rápido, se requirió de una interfaz inalámbrica.

Así que, para diseñar la comunicación inalámbrica entre un equipo de cómputo y el termoplúviógrafo digital, se va a considerar la comunicación por medio de luz infrarroja (IrDA) y por radiofrecuencia (*Bluetooth*), como muestra la figura III.20.

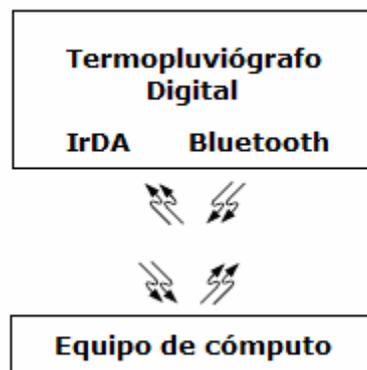


Figura III.20. Comunicación inalámbrica.

Interfaz IrDA

El circuito que se diseñará para lograr la comunicación inalámbrica por medio de luz infrarroja contendrá elementos de *hardware* y *software*,

mediante los cuales se obtendrá la comunicación del termopluviógrafo digital hacia equipos de cómputo, tales como PC, *Lap top* y PDA.

Para lograr la comunicación infrarroja, haciendo uso del estándar IrDA, se utilizó el mismo tipo de circuito de la interfaz USB.

En relación a la comunicación infrarroja, el convertidor cuenta con las siguientes funciones.

- Decodificación y codificación de las señales del convertidor USB a IrDA e IrDA a USB.
- Configuración de la UART.
- Manejo de errores que son administrados por el *firmware* (programa interno del circuito) del microcontrolador.

Para implementar el estándar IrDA se requiere un dispositivo electrónico auxiliar denominado transceptor óptico. El transceptor óptico es un dispositivo que proporciona la interfaz entre la lógica y las señales infrarrojas para establecer la transferencia y la recepción de datos en un sistema de comunicación inalámbrica por medio de infrarrojos. Este dispositivo realiza la conversión de pulsos infrarrojos a señales eléctricas y viceversa, cumpliendo con las especificaciones del estándar IrDA.

El dispositivo que se seleccionó es un transceptor fabricado por la empresa *Vishay*, figura III.21.



Figura III.21. Transceptor óptico.

El transceptor contiene un *led* de alta velocidad y alta eficiencia, de arseniuro de galio, un fotodiodo y un circuito integrado. Este circuito integrado contiene el manejador del *led*, un amplificador y un cuantizador.

En la figura III.22 se muestra un diagrama funcional del transceptor óptico.

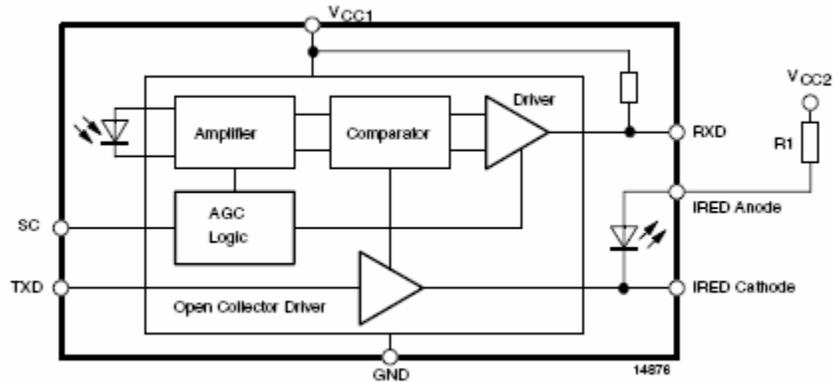


Figura III.22. Diagrama funcional del transceptor óptico.

Para transmitir y recibir datos se requieren principalmente tres terminales del transceptor óptico, una para la recepción de datos (RXD), otra para la transmisión (TXD) y la última para habilitar el dispositivo (*SHUTDOWN*), ya que no se encontrará operando en todo momento. El circuito transceptor tiene una terminal de control de sensibilidad (SC), ésta permite la detección mínima de irradiancia; cuando el nivel lógico colocado en esta terminal sea igual a uno, se incrementa la sensibilidad de las señales infrarrojas y el rango de comunicación de datos aumenta hasta 3 metros. Sin embargo, poner esta terminal en nivel lógico 1, hace que el transmisor y el receptor sea más susceptible de presentar errores en la transmisión, debido al incremento de la sensibilidad hacia fuentes de luz fluorescente.

En la tabla III.12 se presentan las características más importantes del transceptor óptico utilizado.

Características	Descripción
Voltaje de operación	2.7 a 5.5 Volts
Componentes externos	2 capacitores y 1 resistencia
Especificación IrDA	Nivel físico
Velocidad de transmisión	Hasta 115.2 kbps
Paquete de montaje superficial	Universal

Tabla III.12. Características del transceptor óptico.

La figura III.23 muestra el diagrama de conexión de los circuitos convertidor y transceptor óptico.

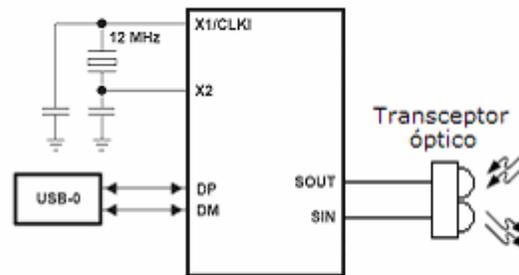


Figura III.23. Implementación del estándar USB a IrDA.

El funcionamiento de la interfaz óptica es el siguiente: la información que es emitida del termoplúviógrafo digital a través de su puerto USB, es suministrada a la entrada USB del circuito convertidor, a su vez, el convertidor la envía por medio de la línea SOUT hacia el transceptor óptico, y éste emitirá los pulsos infrarrojos hacia el transceptor óptico del equipo de cómputo. De la misma forma, los datos suministrados desde la computadora, a través de su circuito óptico, son transmitidos desde el transceptor óptico del convertidor integrado al termoplúviógrafo digital por la línea SIN.

Finalmente, en la figura III.24 se muestra el diagrama completo de la implementación de la interfaz del estándar IrDA.

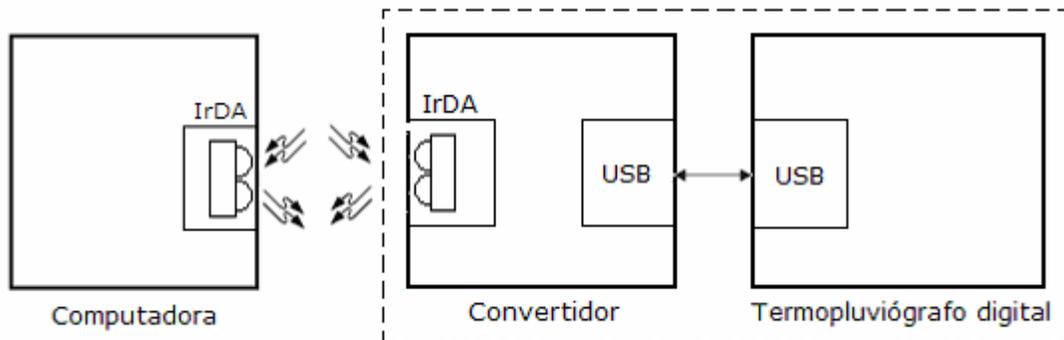


Figura III.24. Implementación de la interfaz IrDA.

Interfaz Bluetooth

El circuito por desarrollar para obtener la comunicación inalámbrica a través de radiofrecuencia (*Bluetooth*) contendrá la implementación de *hardware* y *software*, mediante los cuales se logrará la comunicación del termoplúviógrafo digital hacia un equipo de cómputo que contenga el estándar de comunicación *Bluetooth*.

En la elaboración del diseño del circuito que se comunica por medio del estándar inalámbrico *Bluetooth*, se escogió un dispositivo fabricado por

la empresa *National Semiconductor*. El dispositivo utiliza como base de comunicación la UART del microcontrolador, la cual se encuentra contenida en el termoplúvgrafo digital para realizar la recepción y transmisión de datos hacia el equipo de cómputo.

El módulo *Bluetooth* es un dispositivo que proporciona una solución totalmente implementada en un solo circuito integrado para obtener el estándar de comunicación *Bluetooth 1.2*.

En la figura III.25 se presenta un diagrama funcional de bloques del módulo *Bluetooth*.

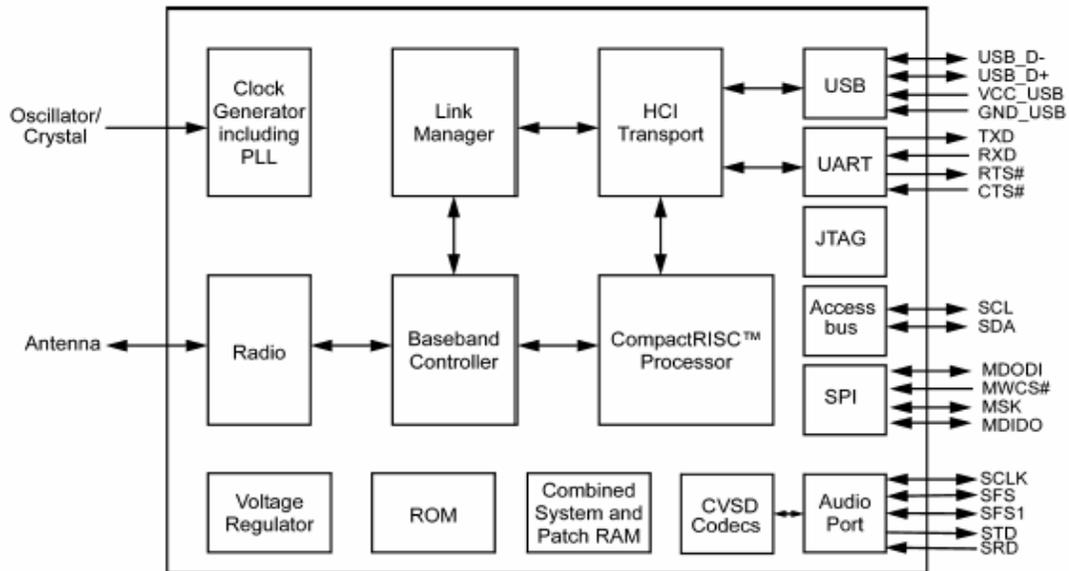


Figura III.25. Diagrama funcional de bloques del módulo *Bluetooth*.

El módulo *Bluetooth* es un dispositivo de comunicación inalámbrica que está basado en un procesador de 16 bits con arquitectura RISC compacta, este módulo está optimizado para manipular audio, datos y los requerimientos de procesamiento del administrador de enlace y control de especificaciones del estándar *Bluetooth*.

El módulo *Bluetooth* contiene cinco interfaces seriales con el fin de comunicarse hacia periféricos externos, estos son: un puerto USB, un puerto que funciona en base a una UART, un puerto *Access bus*, un puerto serial (SPI) y un puerto exclusivamente para audio. Además se encuentra integrado un controlador de banda base que se relaciona directamente con el radio de 2.4 GHz. El radio emplea un filtro de antena integrado y un conmutador de transmisión y recepción de datos, para minimizar el número de componentes externos; también contiene un amplificador de potencia.

En el módulo *Bluetooth* se encuentran integradas memorias ROM y RAM, además, un aspecto importante es que se simplifica el diseño, puesto que el *firmware* está integrado en la memoria ROM. El *firmware* soporta en su totalidad las capas de protocolos de *Bluetooth*, el administrador de enlace y HCI, proporcionando comunicación a través de la interfaz UART o USB. Puede implementar enlaces punto a punto y punto a multipunto, con una velocidad de transmisión hasta de 723 kbps.

Otros elementos que conforman el módulo *Bluetooth* son un generador de reloj, que se conecta a un oscilador externo, así como un regulador de voltaje, que proporciona la energía necesaria para que el dispositivo funcione correctamente.

En la tabla III.13 se muestran las principales características del módulo *Bluetooth*.

Características	Descripción
Especificación	Versión 1.2
Potencia	Clase 2
Voltaje de operación	De 2.5 a 3.6 Volts
Rango de temperatura	-40 C a +85 C
Memoria integrada	ROM y RAM
Enlaces	Soporta hasta 7 enlaces de datos
Enlaces de voz	Soporta 2 enlaces simultáneos de voz
Velocidad de transmisión	Soporta hasta 723 kbps
Control de entorno de operación	Modo de <i>Bluetooth</i> por <i>default</i> Modo de programación del sistema interno (ISP)
<i>Firmware</i>	Integrado en el circuito, con protocolos de capas completo
Reloj externo o entrada de cristal	De 10 a 20 MHz
Oscilador secundario	32.768 kHz
Sintetizador	VCO integrado Proporciona todo el reloj para el radio y funciones de banda base
Puerto de antena	50 Ohms impedancia nominal (incrustada)
Conmutador integrado para transmisión y recepción de datos	Operación <i>Full duplex</i> por medio del puerto para antena
Sensitividad típica de entrada	-80 dBm
Potencia típica de salida	0 dBm
Terminales	60 en un micro módulo paquete BGA (6.1mm X9.1mm X1.2mm)

Tabla III.13. Características principales del módulo *Bluetooth*.

Finalmente, en la figura III.26 se muestra el diagrama completo de la implementación de la interfaz del estándar *Bluetooth*.

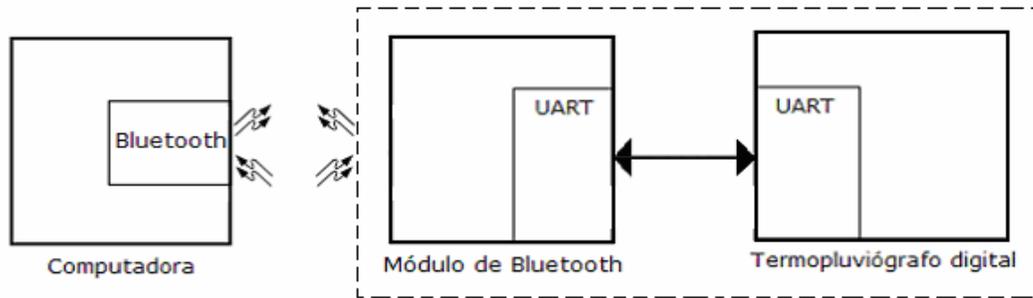


Figura III.26. Implementación de la interfaz Bluetooth.

III.5. Software del sistema

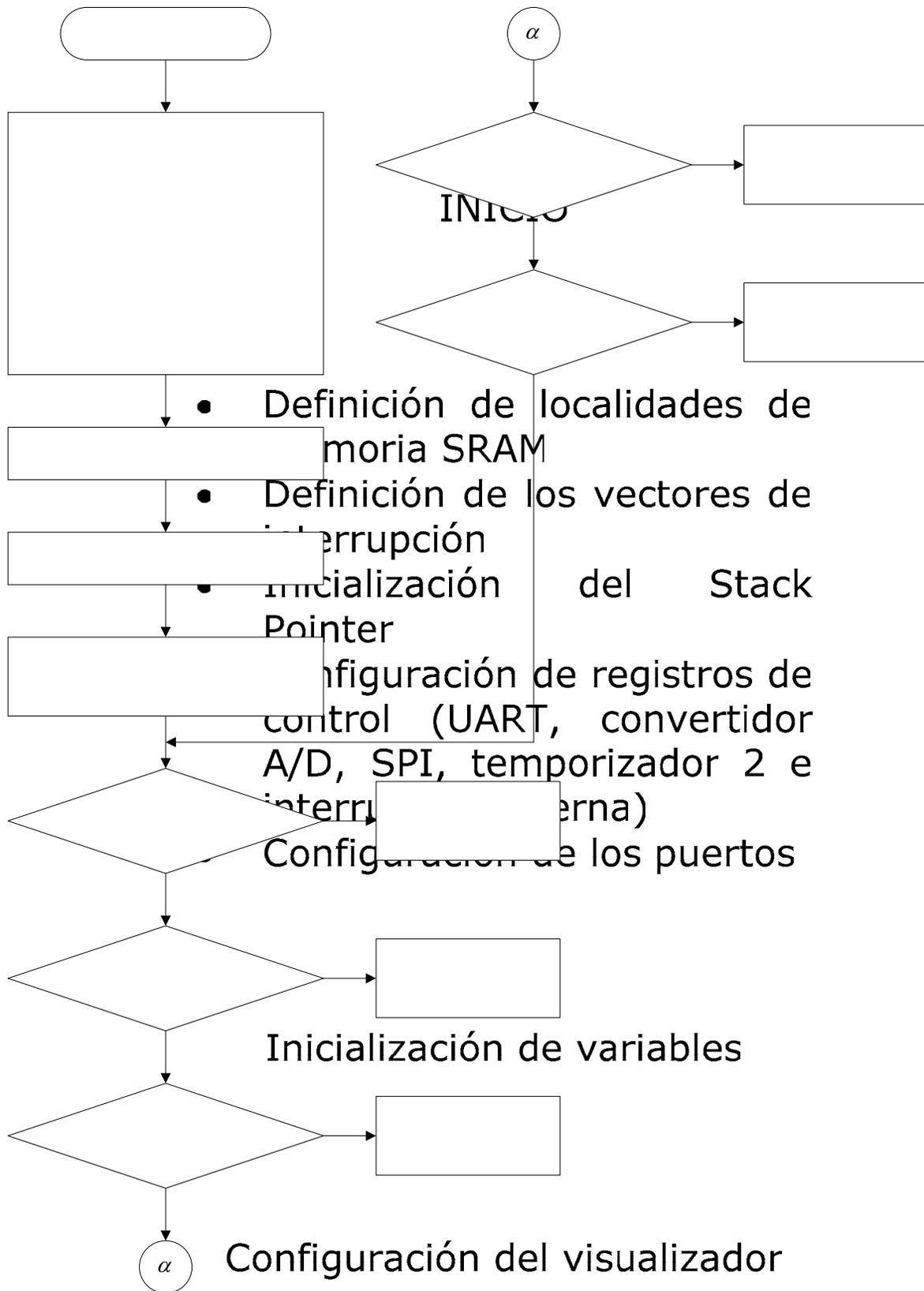
El termopluviógrafo digital utiliza un *software* para su operación, el cual está estructurado en dos partes: una es el *software* para el microcontrolador, que se desarrolló en lenguaje ensamblador y lo llamaremos programa del termopluviógrafo; la segunda es el *software* hecho para la comunicación entre el termopluviógrafo-computadora-usuario, el cual se realizó con lenguaje *Visual Basic* y lo denominaremos programa de aplicación.

Programa del termopluviógrafo

La programación en lenguaje ensamblador es introducida en un microcontrolador, con el objetivo de llevar a cabo el control de dispositivos electrónicos, los cuales no forman parte de los procesadores. El código resultante en este tipo de programas es muy confiable y eficiente.

El microcontrolador utilizado en el termopluviógrafo se programó por medio de lenguaje ensamblador utilizando el conjunto de instrucciones para microcontroladores AVR de *Atmel*. Este código se empleó para implementar los algoritmos de control de los diversos periféricos, entre ellos la comunicación serial, la adquisición de datos y otras rutinas implementadas. En el apéndice C se presentan parcialmente los códigos que conforman el programa del microcontrolador.

En la figura III.27 se presenta un diagrama de flujo, el cual nos proporciona el funcionamiento del sistema de una manera general, sin profundizar en detalles en cada uno de los procesos.



¿T

¿E

Figura III.27. Diagrama general del sistema de registro.

Como se puede observar, el microcontrolador inicia la tarea asignando localidades de memoria SRAM, éstas se utilizan en el almacenamiento de datos temporales. Como ejemplo de datos se pueden mencionar los datos que se muestran en el visualizador y las localidades usadas en la conversión de datos.

Después de haber definido las localidades de memoria, se procede a definir los vectores de interrupción que se emplearon. Las localidades de la memoria de programa del microcontrolador indican que las interrupciones se encuentran en el inicio de éstas. Las interrupciones por emplear son las siguientes: interrupción externa 0, interrupción por desbordamiento del temporizador 2, interrupción de recepción de datos en la UART y la interrupción del vector de *reset*.

Una vez que se han implementado los vectores de interrupción, se procede a especificar la dirección de almacenamiento de las direcciones de regreso de las subrutinas e interrupciones, es decir, se inicializa el *stack pointer*.

Se continúa con la configuración de los registros de control, los cuales permiten que los dispositivos internos del microcontrolador trabajen correctamente. Tales unidades son: la UART, el convertidor analógico-digital, la comunicación serial SPI, así como los temporizadores y las interrupciones.

Es importante destacar que se estableció el formato de comunicación para la UART en los siguientes términos: la velocidad de transmisión de datos en 9600 bps, 8 *bits* de datos, 1 *bit* de paro y no se empleó paridad.

Para el convertidor analógico-digital se define el canal de entrada por usar, en el cual se lleva a cabo la conversión. El modo de conversión utilizado es la conversión simple, puesto que sólo se requiere para conversión del dato de temperatura cuando se presenta un evento. Se establece la frecuencia de operación del convertidor A/D en 125 kHz, se realiza mediante la habilitación de los *bits* respectivos que determinan el factor de división de frecuencia de operación del microcontrolador y la frecuencia de entrada al convertidor A/D. Se empleó esta frecuencia de operación porque el tiempo de conversión es mínimo, es de aproximadamente de 65 μ s y se obtiene máxima resolución.

En la comunicación serial SPI se especificó el orden de transmisión de datos: si el *bit* del registro de orden de datos está habilitado, se

transmite primero el *bit* más significativo (MSB); en caso de estar deshabilitado, se transmite el *bit* menos significativo (LSB). El modo de operación del microcontrolador se establece como dispositivo maestro y las memorias utilizadas operan como dispositivos esclavos. Para las memorias se especifican la polaridad, la fase de reloj y su frecuencia. Para los dispositivos de memoria utilizados, se seleccionaron los siguientes valores: polaridad cero, fase cero y la relación entre la frecuencia de operación del microcontrolador y la frecuencia de transmisión en la comunicación SPI de 128.

El temporizador 2 se encarga de la programación del reloj de tiempo real, donde se establece la interrupción por desbordamiento. De esta manera se incrementan los registros de almacenamiento de los segundos, minutos, horas, días, mes y año. La interrupción por desbordamiento se presenta cada dos segundos, esto tiene la finalidad de mantener un consumo mínimo de energía del termoplúviógrafo, puesto que en estado de bajo consumo, una de las fuentes que deshabilita al microcontrolador de ese estado es por desbordamiento del temporizador y es conveniente que para obtener un mayor rendimiento en la batería, el microcontrolador pase el mayor tiempo en modo de bajo consumo.

Se establece que el modo en que se realizará la interrupción externa 0, se presente cuando se tiene un nivel bajo.

La configuración de registros finaliza con los puertos de E/S, en los cuales se especifica su modo de operación, habilitándolos ya sea como puerto de entrada o puerto de salida.

Terminada la configuración de registros y puertos, se continúa con la inicialización de variables, las cuales serán empleadas durante el control en la operación del dispositivo; posteriormente se realiza la configuración del visualizador. La configuración del visualizador hace la presentación de un mensaje en éste (Figura III.28), el cual identifica al termoplúviógrafo digital.



Figura III.28. Configuración del visualizador.

En esta parte, el microcontrolador se encuentra en estado de espera. Para que el microcontrolador salga de este modo tiene que suceder alguna de las siguientes interrupciones: un desbordamiento del temporizador 2, una interrupción externa, una interrupción provocada por la recepción de datos en la UART y como última interrupción si una tecla de operación es presionada.

Si se presenta una interrupción provocada por el desbordamiento del temporizador del microcontrolador, se ejecuta una actualización del reloj de tiempo real, es decir, se actualizan los datos correspondientes al segundo, minuto, hora, mes y año.

Si se presenta una interrupción por evento, el cual es realizado cuando uno de los balancines del pluviómetro se vuelca, esto lo detecta el microcontrolador a través de la interrupción externa 0. En este momento se lleva a cabo el registro de la hora, minuto y segundo; así como el registro de la temperatura ambiente en ese instante.

Si se genera una comunicación hacia el equipo de cómputo, es porque se presentó una interrupción de recepción de datos, en donde la computadora ha enviado un carácter al microcontrolador, indicándole que desea establecer una comunicación. A través de esta subrutina se pueden realizar tres tareas: la configuración de los parámetros de operación, la modificación de los parámetros y la transferencia de los datos pluviales almacenados en la memoria hacia el equipo de cómputo.

Otra interrupción se ejecuta cuando es presionada una tecla de operación y el visualizador se activa. Esta interrupción provoca que el microcontrolador salga del modo de bajo consumo, por lo tanto, dicha interrupción es generada por el usuario. Posteriormente se pueden llevar a cabo tareas, tales como realizar la revisión de los parámetros de operación o puede hacerles modificaciones. Se abunda la información al respecto en el apéndice B.

Si alguna de las interrupciones descritas anteriormente no se presenta y pasan dos minutos desde que el termoplúviógrafo entró en modo de espera, el microcontrolador entrará en modo de bajo consumo y saldrá de éste cuando alguna de las interrupciones se presente.

Programa de aplicación

Se requiere de un *software* que se utilice como interfaz entre el termoplúviógrafo, la computadora y el usuario; con el objetivo de intercambiar información. Este *software* está hecho con un lenguaje de

programación para desarrollar el sistema, el lenguaje de programación que se empleó fue *Visual Basic*.

El código fue desarrollado para implementar la interfaz gráfica, éste se empleó para diseñar los algoritmos de control de la operación del termoplúviógrafo, entre ellos se encuentran: la selección del puerto de comunicación, la adquisición de datos, la modificación de los parámetros de operación, la presentación de datos en diferentes formatos y otras rutinas implementadas. En el apéndice C se presentan parcialmente los códigos de procedimientos del sistema.

En la figura III.29 se presenta un diagrama de flujo, el cual nos proporciona el funcionamiento del sistema de una manera general, sin profundizar en detalles en cada uno de los procesos.

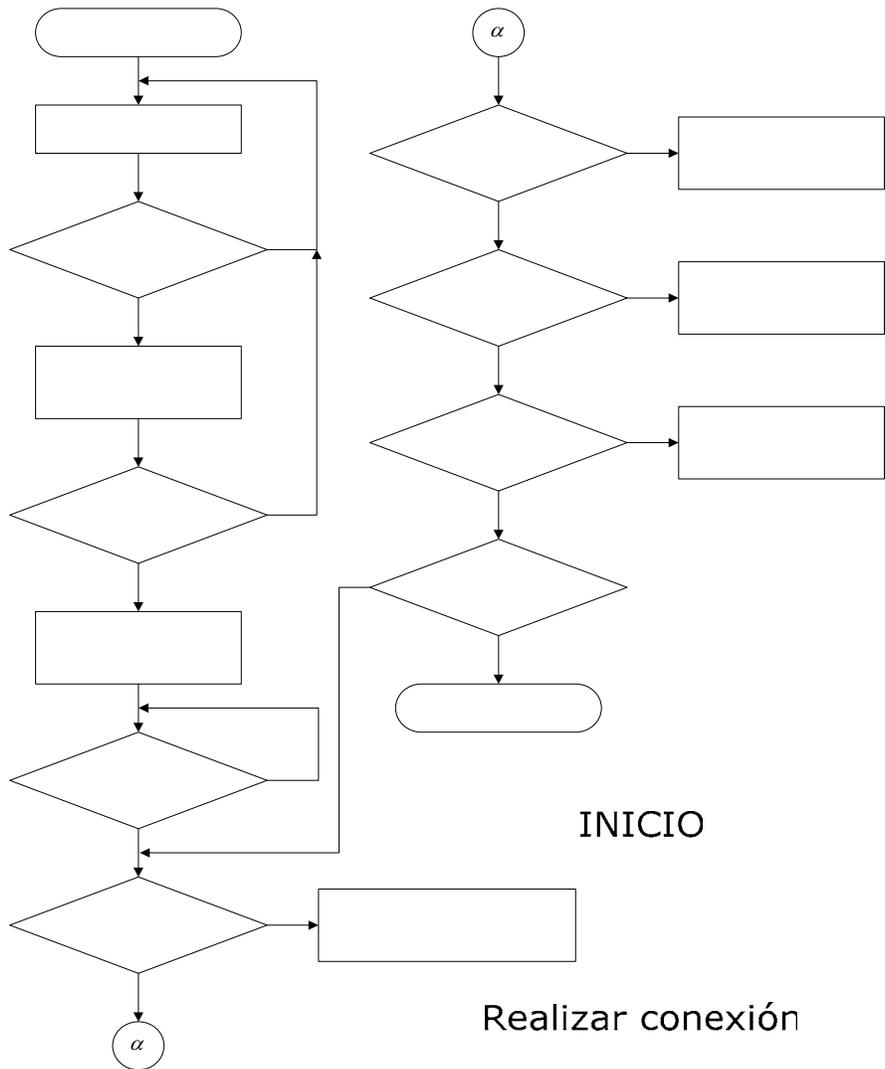


Figura III.29. Diagrama general del código de interfaz gráfica.

Cuando el usuario desea establecer una comunicación serie con el termoplúviógrafo digital, primeramente se debe de realizar la instalación física entre el termoplúviógrafo y el equipo de cómputo. En ese momento el visualizador del termoplúviógrafo muestra en su pantalla un rótulo que dice "COMUNICACIÓN PC", esto significa que el termoplúviógrafo está conectado y listo para establecer comunicación con la computadora.

El usuario puede ingresar al programa de comunicación, el cual ha sido previamente instalado en la computadora y se tiene acceso a él por medio de un acceso directo instalado en el escritorio o bien en el menú de programas. Ejecutando el programa termoplúviógrafo, aparece la ventana principal de éste, en la cual se podrá configurar y seleccionar el puerto serie de comunicación del equipo de cómputo. Si el sistema reconoce el procedimiento anterior, éste pasa a realizar las funciones más importantes o básicas del programa.

El programa realiza dos funciones básicas, las cuales son: la configuración o modificación de los parámetros de operación y la adquisición de datos almacenados en la memoria del termoplúviógrafo.

La configuración de los parámetros de operación define un algoritmo, el cual ejecutará las operaciones necesarias para hacer las modificaciones a los parámetros seleccionados.

La adquisición de datos pluviales es aquella función que obtiene los eventos registrados en la memoria del termoplúviógrafo. Cuando el equipo de cómputo ha adquirido los datos, estos pueden presentarse al usuario en tres formatos: un formato que requiere un procesador de hoja de cálculo, por ejemplo Excel; otro formato con el que se presenta la información en un archivo de texto y finalmente un formato donde la información se presenta gráficamente.

Una vez que se obtuvo la información por medio de un formato, se tiene la opción de regresar al sistema o dar por terminada la comunicación serial entre la computadora y el termoplúviógrafo digital.

La interfaz gráfica del sistema desarrollada en *Visual Basic*, se presenta ampliamente en el apéndice B como un manual de usuario.

En este capítulo se ha presentado el diseño y desarrollo de cada uno de los elementos que componen el termoplúviógrafo digital, abarcando el *hardware* y *software*, ahora pasaremos a presentar las evaluaciones y resultados del sistema.

CAPÍTULO IV

APLICACIÓN DE PRUEBAS

En el presente capítulo van a presentarse los diferentes tipos de pruebas que le fueron aplicadas al prototipo del termoplviógrafo digital.

IV.1. Evaluación

Entre las pruebas a las que fueron sometidos los diferentes módulos que integran el termoplviógrafo se encuentran las siguientes:

- Evaluación individual por componente en una tarjeta de prueba o *protoboard*.
- Pruebas de errores por *default*.
- Verificación en transmisión y recepción de datos.

Evaluación individual por componente en una protoboard. Para cada dispositivo que se seleccionó para formar parte del módulo correspondiente, se verificó su operación y funcionamiento en una tarjeta de prueba o *protoboard*, con el fin de ajustar parámetros y solucionar detalles que se pudieran presentar, relacionados con el circuito. Por ejemplo, en la figura IV.1 se puede observar el circuito adaptador de voltaje utilizado en el circuito convertidor USB a RS-232. Dicho circuito adaptador es sometido a las pruebas de operación y funcionamiento, en donde participan equipos de medición de laboratorio, con los cuales se verifican los valores esperados.

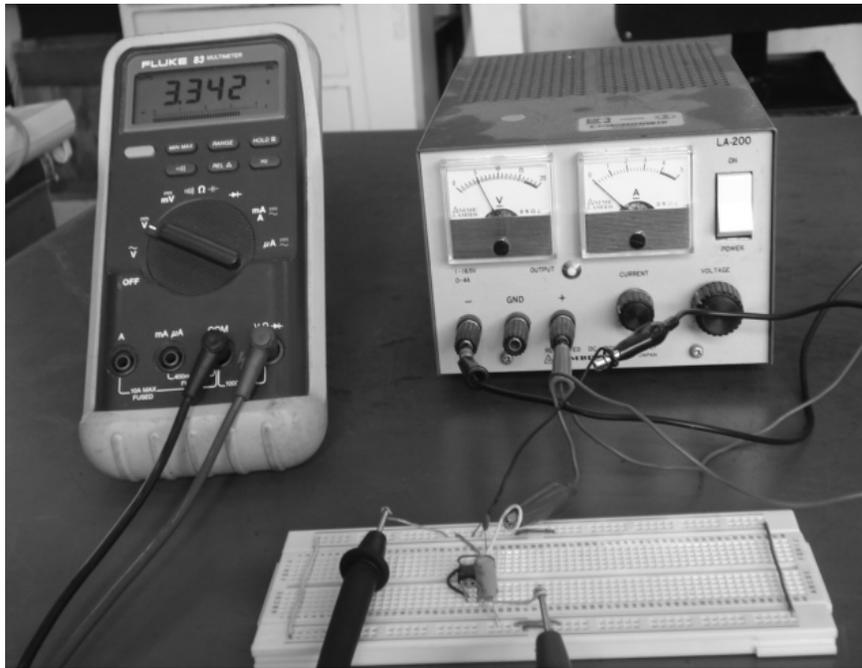


Figura IV.1. Pruebas a dispositivo electrónico en una protoboard.

Resuelta la evaluación del dispositivo en la tarjeta de prueba, se procedía a su implementación en una tarjeta perforada de cobre, como se muestra en la figura IV.2.

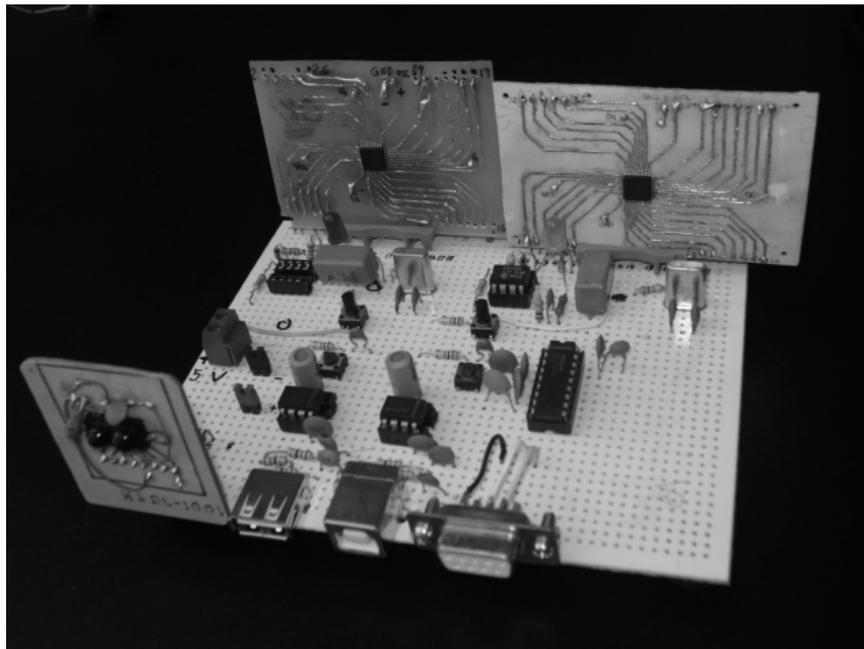


Figura IV.2. Implementación de circuitos en tarjeta perforada.

En la realización de estas pruebas se presentaban problemas imprevistos e incluso errores humanos, tales como, falsos contactos en la *proto-board*, falsos contactos en la tarjeta del prototipo final, soldaduras mal aplicadas, dispositivos que se descomponían al momento de la aplicación de soldadura, errores de conexión y dispositivos mal colocados.

Algunos de los circuitos que se trabajaron en montaje superficial presentaron muchos problemas para hacer su circuito impreso, por lo próximo y delgadas de las pistas de configuración, así como por el exceso de calor suministrado en la etapa de aplicación de soldadura. Una muestra de un dispositivo de montaje de superficie es el convertidor USB a RS-232, el cual se presenta en la figura IV.3. Varios circuitos integrados se echaron a perder por estos motivos.

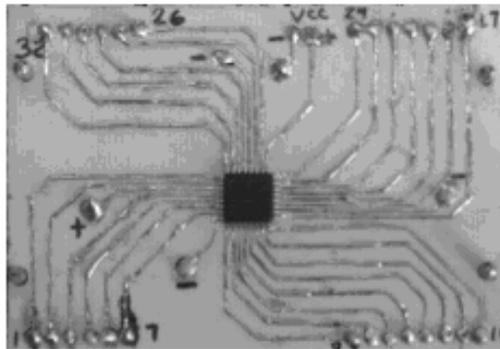


Figura IV.3. Circuito integrado de montaje de superficie.

Pruebas de errores por default. Son pruebas que se aplicaron a los componentes de los módulos, pruebas con fallas conocidas; es decir, se aplicaban pruebas las cuales iban a generar errores, así que se sabía el resultado que se iba a obtener. Algunas de estas pruebas son las siguientes: cables mal conectados, falsos contactos, fallas en el suministro de energía, fallas de *software*, etc.

Estas pruebas aunque estaban controladas y contempladas, no se podían exceder de ciertos intervalos de valores, porque si se sobrepasaban, entonces se estaba sentenciando a descomponer el dispositivo de prueba. Dentro de los parámetros por variar se pueden mencionar: el voltaje de alimentación, considerando que las hojas de especificación de cada dispositivo indican el intervalo permisible de operación sin ocurrir fallas; cables mal conectados, por ejemplo no podemos conectar GND con VCC puesto que se dañaría el circuito integrado, etc.

Verificación en la transmisión y recepción de datos. Esta prueba se realizó para cada uno de los estándares de comunicación, así como la comprobación de sus protocolos tanto alámbricos como inalámbricos.

Los módulos de comunicación fueron sometidos primeramente a pruebas independientes del termopluviógrafo, es decir, las pruebas que se les realizaron fueron como interfaz entre dos computadoras utilizando el puerto hiperterminal de cada una de éstas. Posteriormente una vez que se comprobó su funcionamiento, se realizaron pruebas de comunicación y de operación con el termopluviógrafo.

En las pruebas de comunicación de datos se simulaban posibles errores, como los siguientes: el error en la conexión del cable, la transferencia interrumpida de datos, datos erróneos, corte en la alimentación de energía al sistema y el error en algún parámetro mal configurado (por ejemplo, la selección del puerto de comunicación).

Para cada estándar de comunicación se le aplicaron pruebas semejantes, donde se obtuvieron los siguientes resultados:

EIA-232. El estándar EIA-232 funcionó correctamente cumpliendo con todas las especificaciones, así como los datos transmitidos fueron verificados y confirmados, por lo que esta prueba es confiable y resultó el estándar más sencillo de implementar. Recordemos que este estándar es el más importante del proyecto, puesto que a partir de él se podrán implementar los demás protocolos de comunicación. La figura IV.4 muestra el estándar EIA-232 (zona punteada) colocado en el prototipo final.

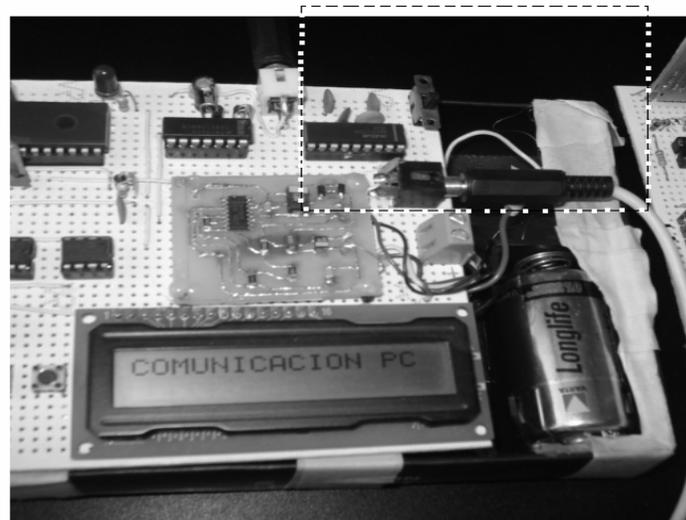


Figura IV.4. Estándar EIA-232. (zona punteada)

USB. El estándar USB fue el circuito que más se complicó para su implementación, desde el circuito impreso hasta las pruebas finales realizadas, pero finalmente cumplió con todas las expectativas establecidas de comunicación, teniendo éxito en la evaluación, tal como se muestra en la figura IV.5.



Figura IV.5. Evaluación del estándar de comunicación USB.

IrDA. En la comunicación inalámbrica por medio de infrarrojos, las pruebas realizadas, además de las descritas anteriormente, fueron las siguientes: la consideración del ángulo de alineación y la distancia de comunicación entre los dispositivos ópticos. En estas pruebas se comprobó que el ángulo al que puede llegar a transmitir el dispositivo sin ningún problema es de 15 grados, con una distancia máxima de un metro.

El estándar IrDA también resultó muy complicado para su implementación, puesto que no se contaba con circuitos ópticos para realizar las pruebas y los que había en existencia no se tenía la certeza de que estuvieran en buen estado. Finalmente, superados los inconvenientes, se logró la comunicación (Figura IV.6).

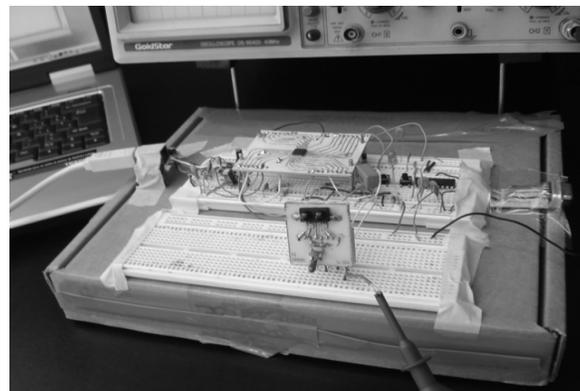


Figura IV.6. Transmisión de datos por IrDA.

Bluetooth. El estándar de comunicación inalámbrico *Bluetooth* no se pudo lograr por los siguientes aspectos: el dispositivo de *Bluetooth* por usar sólo se podía conseguir en Estados Unidos y tardaron en enviarlo varios meses, posteriormente el problema al que nos enfrentamos fue el no contar con la tecnología adecuada para realizar su circuito impreso y soldar el componente.

No se pudo realizar su tarjeta de circuito impreso por el tamaño, como se muestra en la comparación del módulo *Bluetooth* con una moneda y con una resistencia (Figura IV.7). En esta área mostrada del módulo *Bluetooth* se encuentran 60 terminales de conexión, con una distancia muy próxima entre ellas, de las cuales se tenían que soldar 37 para lograr su salida hacia una tarjeta impresa y así poder manejar las líneas para hacer su tarjeta de circuito impreso. Cabe mencionar que se contó con la ayuda del Instituto de Física para la elaboración de esta tarjeta de circuito impreso, pero aunque ellos cuentan con herramientas de precisión específicas para estos trabajos, tal como un equipo computarizado que realiza circuitos impresos, tampoco se logró hacerlo debido a la distancia entre las terminales del circuito integrado. Por consecuencia podemos decir que el estándar de *Bluetooth* no se pudo realizar por falta de tecnología.

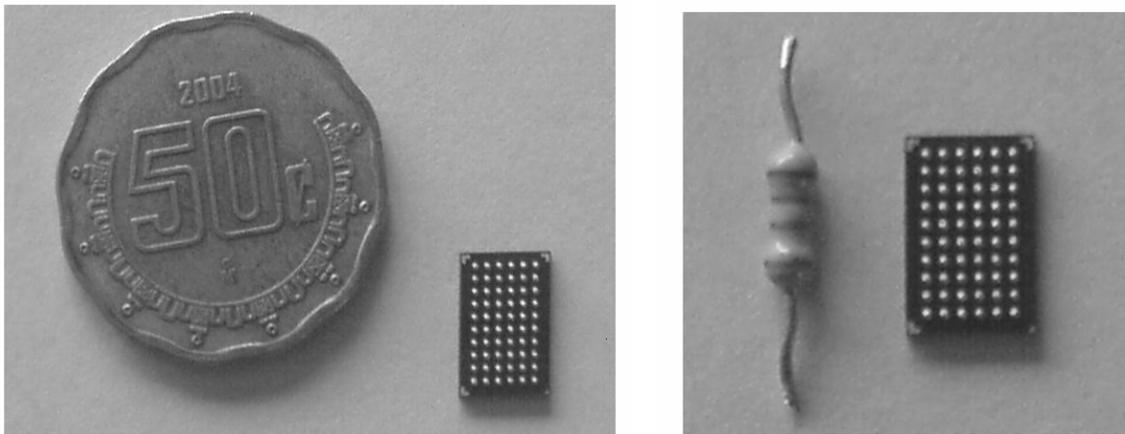


Figura IV.7. Tamaño del módulo *Bluetooth*.

Una vez hechas las evaluaciones a cada uno de los módulos, se procedió a realizar la integración de todos ellos para la conformación del prototipo final del termopluviógrafo digital y poder aplicarle pruebas.

Concluido el prototipo del termopluviógrafo digital, que está compuesto por el pluviómetro tipo balancín (Figura IV.8), el cual se colocó en la azotea del edificio del Instituto de Ingeniería de la UNAM, una tarjeta de

adquisición de datos y una tarjeta que contiene las interfaces de comunicación, se procedió a realizar pruebas de operación del termoplúviógrafo.



Figura IV.8. Pluviómetro tipo balancín.

Para la realización de pruebas de operación del termoplúviógrafo, se utilizó un equipo de cómputo, con el objetivo de verificar que efectivamente el *software* y el *hardware* desarrollados funcionaban correctamente. Como equipo de cómputo para las pruebas de operación con el termoplúviógrafo digital se empleó una *lap top*, esto se muestra en la figura IV.9.

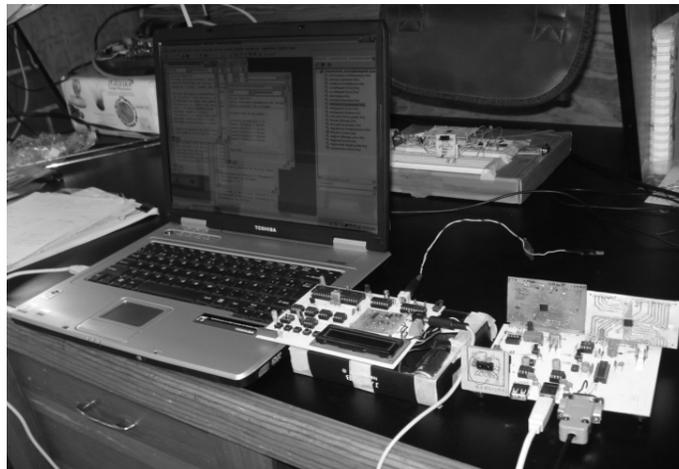


Figura IV.9. Pruebas de operación al termoplúviógrafo digital.

Las pruebas de operación realizadas al prototipo final consistieron en lo siguiente: aplicación de eventos pluviométricos y la transmisión y recepción de datos entre el equipo de cómputo y el termoplúviógrafo.

Aplicación de eventos pluviométricos. Estos eventos pluviométricos se aplicaban al balancín directamente, es decir se le echaba agua manualmente, para que éste mandara la señal al adquisidor de datos y a partir de ello comenzara a funcionar todo el sistema de medición, tanto en *hardware* como en *software*. Se llevó un registro de los eventos aplicados y se constató que la computadora adquiría bien la información y la presentaba correctamente en los formatos disponibles. Es importante mencionar que no todo salió bien en el primer intento, sino que se presentaron muchos problemas de *hardware* y *software*, pero se fueron resolviendo como iban apareciendo; a veces era rápida la solución y en otras ocasiones se llevaban semanas para su corrección.

Entre las pruebas es importante destacar que también se llevaron a cabo la obtención de registros de eventos producidos por lluvia natural, y de la misma manera se realizaron registros durante largos periodos de tiempo, finalmente se obtuvo el éxito esperado.

En la figura IV.10 se presenta un reporte hecho en formato de texto, donde se puede observar que los registros obtenidos durante un tiempo de prueba presentan datos erróneos, estos son: la fecha de inicio, la capacidad del balancín, el número de eventos, la precipitación acumulada, la hora en que ocurrió cada uno de los eventos y la temperatura; en este periodo de prueba el termopluviógrafo estaba fallando.

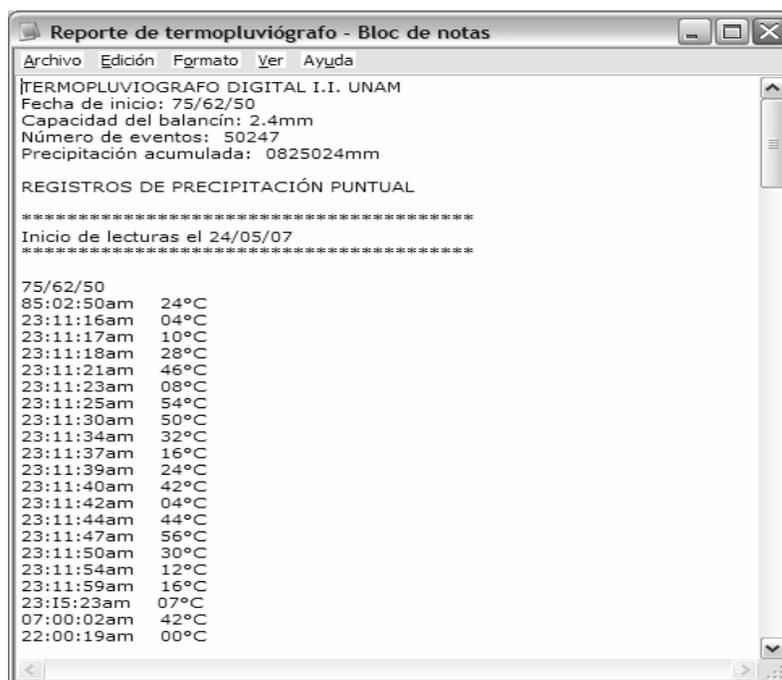


Figura IV.10. Presentación errónea de datos en formato de texto.

En la figura IV.11 se presenta un reporte hecho en formato de texto, donde se pueden observar los registros obtenidos durante un tiempo de prueba, los cuales presentan datos reales de precipitación pluvial.

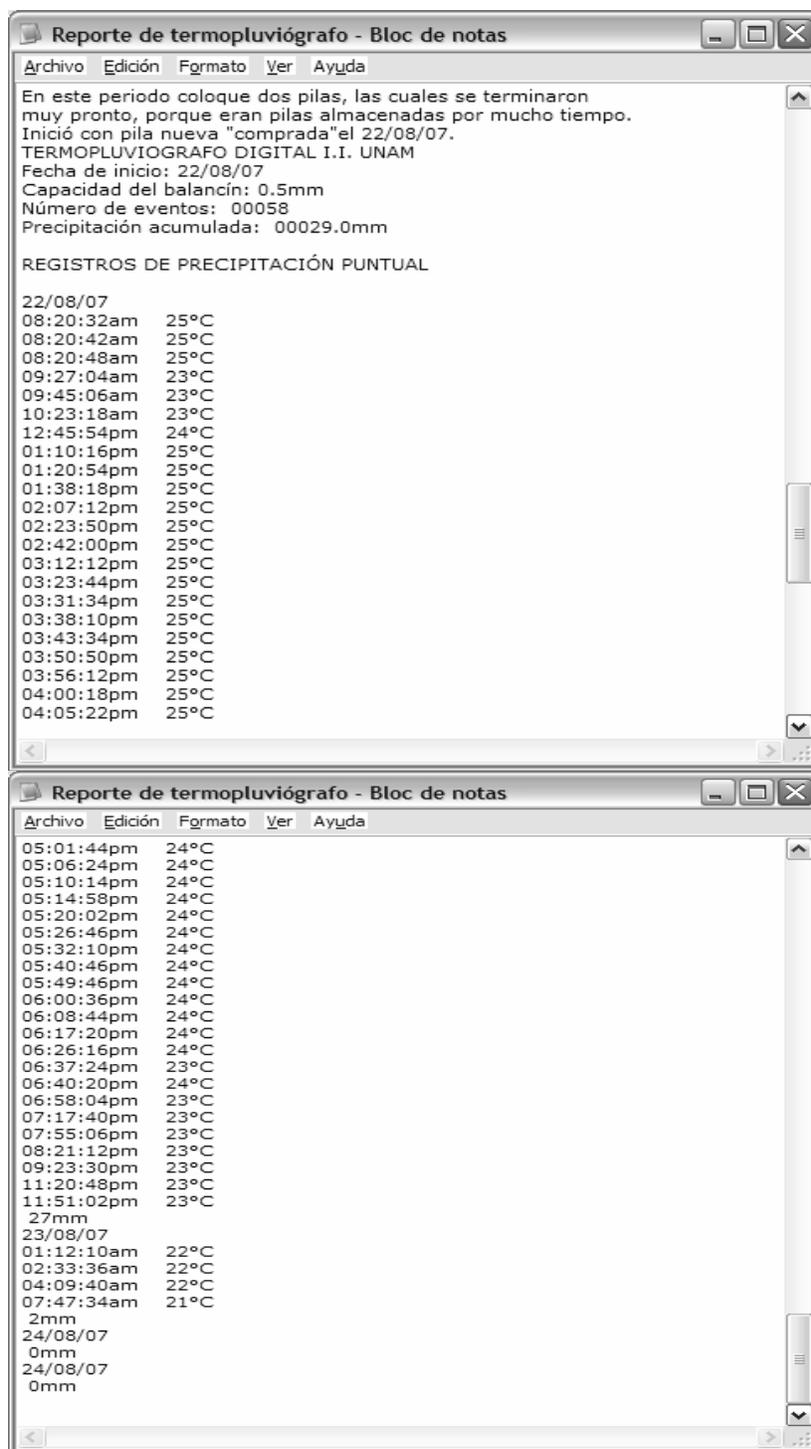


Figura IV.11. Presentación de datos verídicos en formato de texto.

Transmisión y recepción de datos. Otra de las pruebas fue hacer que el equipo portátil fuera capaz de establecer comunicación con el termoplúviógrafo digital, así como transmitir y recibir datos por el puerto de comunicación seleccionado, sin ningún problema, lo cual se logró puesto que de esa manera se obtuvieron los reportes mostrados anteriormente. Cuando se presentaron errores de *hardware* y *software* o de otro tipo, también se fueron solucionando.

Para cada uno de los estándares de comunicación desarrollados, se procedió a verificar que cada especificación establecida se cumplía, es decir, cada dispositivo utilizado para un protocolo de comunicación, actuaba de acuerdo con cada punto marcado en las especificaciones. La figura IV.12, muestra como un ejemplo la prueba de transmisión y recepción de información a través del estándar USB.

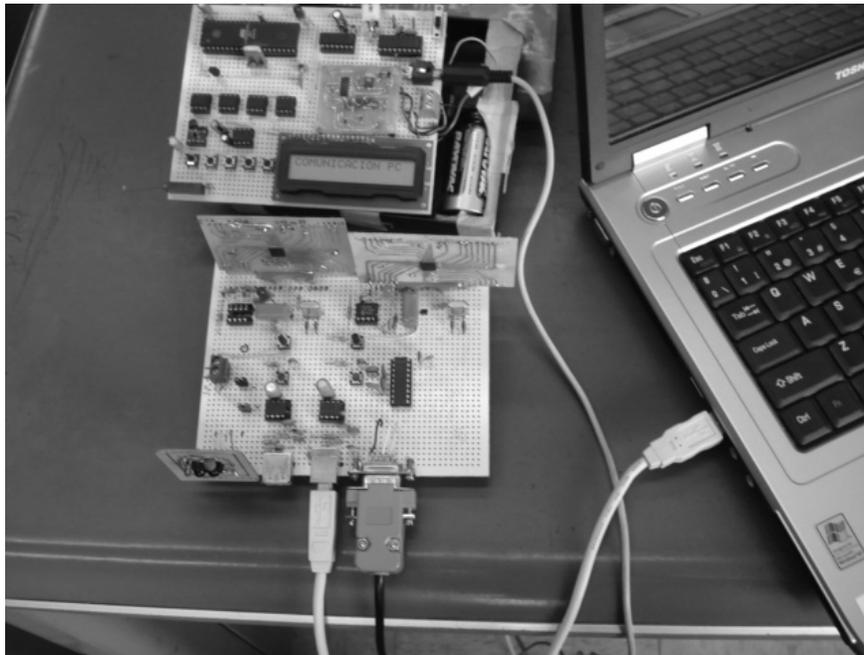


Figura IV.12. Prueba de transmisión y recepción de datos por USB.

Cabe comentar que todas estas pruebas se realizaron a nivel laboratorio.

En este capítulo se han presentado las pruebas a las que fue sometido el termoplúviógrafo digital, así que continuaremos con los resultados obtenidos y a las conclusiones a las que se llegaron.

CAPÍTULO V

RESULTADOS Y CONCLUSIONES

En este capítulo se presentan los resultados obtenidos y las conclusiones a las que se llegaron con respecto al diseño y desarrollo del termopluviógrafo digital.

V.1. Resultados

Haciendo un análisis del diseño y desarrollo del termopluviógrafo digital, podemos decir que para su realización se planteó una metodología consistente de varios pasos:

En el primero se realizó el análisis e investigación de las necesidades de realizar un termopluviógrafo digital que pudiera comunicarse con un equipo de cómputo (sea una PC, *lap top* o PDA), a través del puerto serie de comunicación RS-232. Realizado lo anterior se pudieran desarrollar convertidores de protocolos de comunicación alámbricos e inalámbricos y por estos medios intercambiar comunicación.

La segunda etapa correspondió al diseño y desarrollo de los módulos necesarios, tanto de *hardware* como de *software* del termopluviógrafo y del equipo de cómputo. Para el *hardware* fueron considerados el diseño y la selección de dispositivos electrónicos que integrarían el sistema.

Éste fue un punto muy importante para el desarrollo del prototipo, puesto que los dispositivos tienen ciertos parámetros importantes que se deben tomar en cuenta. Se destaca que también se desarrolló el *software* necesario para el microcontrolador que se encuentra en el sistema de adquisición, así como para el equipo de cómputo que hace las veces de interfaz de comunicación con el termopluviógrafo digital.

La tercera etapa consistió en la aplicación de las diferentes pruebas de laboratorio que se realizaron al prototipo final del termopluviógrafo digital, en esta fase se evaluó el desempeño del sistema en condiciones críticas y en condiciones óptimas de operación.

Una vez que se finalizó con éxito la etapa de pruebas, se obtuvo el prototipo del termopluviógrafo digital. El prototipo final del termopluviógrafo digital cumple con los objetivos especificados en el diseño y desarrollo de este proyecto.

Se comprobó el funcionamiento correcto de cada módulo integrante del termopluviógrafo digital tanto en *hardware* como en *software*.

El sistema completo opera de una forma sencilla, entendible y confiable para un usuario, sin tener la necesidad de una capacitación compleja para su operación.

Finalmente en la tabla V.1 se presentan las especificaciones técnicas del termopluviógrafo digital.

Especificación	Descripción
Alimentación de voltaje	1 pila tamaño C de 1.5 V
Duración de batería	6 meses
Parámetros de medición	Eventos pluviales y temperatura
Capacidad de medición	20,000 eventos de lluvia y de temperatura
Tipo de evento	Por volcamiento de balancín
Tipo de memoria	No volátil EEPROM
Modos de almacenamiento	Dos: <ul style="list-style-type: none"> • Modo lineal: cuando la memoria está llena se detiene el almacenamiento. • Modo sobrescribir: si la memoria está llena, sobrescribe la información en los datos más antiguos.

Tabla V.1. Especificaciones del termopluviógrafo digital. (Continúa)

Intervalo de temperatura de operación	-20°C a 70°C
Visualizador alfanumérico	Pantalla de cristal líquido de 2 X 16
Plataforma de operación	Windows 95/98/XP/CE
Protocolo de comunicación básico	EIA-232
Lenguajes de programación utilizados	Dos: <ul style="list-style-type: none"> • Ensamblador para el microcontrolador • <i>Visual Basic</i> para interfaz gráfica

Tabla V.1. Especificaciones del termopluviógrafo digital.

V.2. Conclusiones

Se cumplió el objetivo del proyecto, el cual fue diseñar e implementar un termopluviógrafo digital, teniendo incluidos tanto comunicaciones de tipo alámbricas como inalámbricas.

El termopluviógrafo digital cuenta con un sistema de comunicación que permite la transmisión de datos hacia un equipo de cómputo a través de diversos estándares de comunicación. Por lo tanto, la versatilidad de los módulos de comunicación hace que el usuario elija el medio de comunicación más óptimo, seguro y eficiente; dependiendo de las necesidades y posibilidades de los equipos de cómputo.

Mediante la evaluación del prototipo final se comprobó que es un sistema confiable para la adquisición datos pluviométricos.

Este termopluviógrafo digital es una primera versión en prototipo, que requiere de la fabricación de los circuitos impresos correspondientes, con sus pruebas y puesta a punto para tener una versión final en prototipo, listo para su fabricación.

Se recomienda el uso de componentes de montaje de superficie y de circuito de bajo consumo de energía, para hacer un uso más eficiente de la pila.

Este proyecto puede crecer considerablemente, puesto que se podrían agregar más estándares de comunicación, por ejemplo: comunicación por *internet*, por módem telefónico, por radio módem y por satélite.

APÉNDICE A

MODELO OSI

En el mundo existen seis agencias de estándares internacionales que definen las normas para los sistemas de comunicación de datos:

- Organización Internacional de Estándares (ISO)
- Comité Consultivo Internacional de Telegrafía y Telecomunicaciones (CCITT)
- Asociación de Industrias Eléctricas (EIA)
- Unión Internacional de Telecomunicaciones (ITU)
- Instituto Nacional de Estándares Americanos (ANSI)
- Instituto de Ingenieros Eléctricos y Electrónicos (IEEE)

ISO e IEEE definen los estándares para la conexión de computadoras hacia redes de áreas locales y CCITT (ahora incorporada con ITU) define estándares para la interconexión de redes nacionales e internacionales. EIA define estándares para la interconexión de computadoras utilizando comunicación en serie.

Dentro de las especificaciones de estándares es importante proporcionar algunas definiciones, las cuales se utilizan comúnmente en el entorno de las comunicaciones, tales como: protocolo y niveles o capas.

Protocolo: es el conjunto de normas y reglas, organizadas y convenidas de mutuo acuerdo entre todos los participantes en una comunicación.

Su función es hacer que la comunicación entre todas las computadoras de una red que están usando ese protocolo sea compatible y se pueda regular algún aspecto de la misma. Estos protocolos son estandarizados por las asociaciones u organizaciones de estandarización, y los fabricantes toman en cuenta estos estándares para la realización de dispositivos de comunicación.

Niveles o Capas: las redes de computadoras proporcionan al usuario una serie de servicios, e internamente ejecutan ciertas funciones. Todo esto es realizado por las capas o niveles de la arquitectura que posee el tipo de red. La arquitectura de las redes tiene una serie de capas superpuestas, una encima de otra, en la que cada una desempeña su función. Los niveles tienen las siguientes funciones y características:

- Permiten fraccionar el desarrollo del protocolo que usa.
- Las capas facilitan la comprensión del funcionamiento global de un protocolo.
- Facilitan las compatibilidades tanto de *software* como *hardware* de las diferentes computadoras conectadas.
- La arquitectura o estructura de capas son flexibles a la hora de modificarlas.

Una vez que se han definido los dos conceptos anteriores, en este proyecto solamente se hará una referencia básica a un estándar que fue creado por la agencia ISO, nos referimos al modelo Interconexión de Sistemas Abiertos (OSI).

Modelo OSI

En 1977, la Organización Internacional de Estándares creó un grupo de estudio para desarrollar estándares de comunicación de datos, con la finalidad de obtener los siguientes puntos.

- Promover la accesibilidad universal.
- Interoperabilidad entre productos de diferentes fabricantes.

El resultado final de este proyecto es el modelo de referencia OSI. El modelo OSI es un lineamiento funcional para los distintos tipos de comunicación, por lo tanto, no especifica un estándar de comunicación, sin embargo, muchos estándares y protocolos cumplen con los lineamientos del modelo OSI. El modelo OSI surge de la necesidad de estandarizar los elementos que participan en la solución del problema de comunicación entre equipos de cómputo de diferentes fabricantes. Las

diferencias que se presentan entre los diferentes equipos, involucran a los siguientes elementos:

- Unidad de Procesamiento Central
- Velocidad de comunicación
- Memoria
- Dispositivos de almacenamiento
- Interfaces para comunicaciones
- Códigos de caracteres
- Sistemas Operativos

Estas diferencias propician que el problema de comunicación entre computadoras no tenga una solución simple. Si el problema general de comunicación se clasifica en problemas específicos, facilitamos la obtención de una solución a dicho problema. Lo anterior proporciona una mejor comprensión del problema, la solución de cada problema específico puede ser optimizada individualmente.

Objetivos del modelo OSI

Entre los principales objetivos del modelo OSI se encuentran:

- Obtener un modelo de referencia estructurado en varios niveles, en los cuales se contemple desde el concepto *bit* hasta el concepto aplicación.
- 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.

Estructura del modelo OSI

El objetivo perseguido por OSI establece una estructura que presenta especificaciones en su estructura multinivel, en los puntos de acceso, en la dependencia de los niveles y en la transferencia de mensajes.

Estructura multinivel: la estructura multinivel está definida con la idea de que cada nivel resuelva una parte del problema de comunicación, esto es, cada nivel realiza funciones específicas. La comunicación internivel está bien definida, un nivel utiliza los servicios de los niveles inferiores, así como proporciona servicios a otro nivel superior. Cada nivel se comunica con su nivel similar de otras computadoras, pero debe

hacerlo enviando un mensaje a través de los niveles inferiores en la misma computadora.

Puntos de acceso: entre los diferentes niveles existen interfaces llamadas puntos de acceso a los servicios.

Dependencias de niveles: cada nivel es dependiente del nivel inferior y también del superior.

Transferencia de mensajes: en cada nivel se incorpora al mensaje un formato de control. Este elemento de control permite que un nivel en la computadora receptora se entere de que su similar en la computadora emisora está enviándole información a través de un mensaje. Un mensaje está constituido en dos partes, el encabezado y la información.

La incorporación de encabezados es necesaria aunque representa una cantidad extra de información, lo que implica que un mensaje corto pueda ser voluminoso. Sin embargo, como la computadora destino retira los encabezados en orden inverso a como fueron incorporados en la computadora origen, finalmente el usuario sólo recibe el mensaje original.

Niveles del modelo OSI. El modelo OSI está estructurado en siete capas o niveles, según se observa en el siguiente cuadro.

Nivel 7	Aplicación
Nivel 6	Presentación
Nivel 5	Sesión
Nivel 4	Transporte
Nivel 3	Red
Nivel 2	Enlace de datos
Nivel 1	Físico

Las características principales de cada uno de estos niveles son:

Nivel Físico: define el medio de comunicación utilizado para la transferencia de información. Contiene el control de este medio y especifica *bits* de control, mediante las siguientes definiciones:

- Define conexiones físicas entre computadoras.
- Describe el aspecto mecánico, eléctrico y funcional de la interfaz física.
- Define la técnica, velocidad y tipo de transmisión.
- Define la codificación de línea.

- Define el modo de operación de la línea de datos.

Nivel Enlace de Datos: proporciona la especificación para transmitir bloques de datos entre dos dispositivos que se encuentran conectados en una red, esto es, organiza los '1's y los '0's del nivel físico en formatos o grupos lógicos de información con los siguientes objetivos:

- Detectar errores en el nivel físico.
- Establecer esquemas de detección de errores en retransmisiones o reconfiguraciones de la red.
- Establecer el método de acceso que la computadora debe seguir para transmitir y recibir mensajes.
- Realizar la transferencia de datos a través del enlace físico.
- Enviar bloques de datos con el control necesario para la sincronía.

Nivel de Red: este nivel define el enrutamiento y el envío de paquetes de datos entre redes. Es responsabilidad de este nivel establecer, mantener y terminar las conexiones. Proporciona el enrutamiento de mensajes, por ejemplo, si un mensaje se envía al nivel 4 (Nivel de transporte) o bien al nivel 2 (Enlace de datos).

Nivel de Transporte: este nivel actúa como un puente entre los tres niveles inferiores totalmente orientados a las comunicaciones y los tres niveles superiores totalmente orientados al procesamiento. Además, garantiza una entrega confiable de la información, asegura que la llegada de datos del nivel de red encuentre las características de transmisión y la calidad de servicio requerido por el nivel 5 (Nivel de sesión). El nivel de transporte define como direccionar la localidad física de los dispositivos de la red, puede soportar múltiples conexiones. Define la manera de habilitar y deshabilitar las conexiones entre los puntos de conexión. Determina el protocolo que garantiza el envío del mensaje. Establece la transparencia de datos, así como la confiabilidad en la transferencia de información entre dos sistemas.

Nivel Sesión: proporciona los servicios utilizados para la organización y sincronización del diálogo entre usuarios y el manejo e intercambio de datos.

- Establece el inicio y fin de la sesión.
- Recuperación de la sesión.
- Control del diálogo; establece el orden en que los mensajes deben fluir entre usuarios finales.
- Identifica a los dispositivos por nombre y no por dirección.

- Permite escribir programas que correrán en cualquier instalación de red.

Nivel Presentación: traduce el formato y asignan una sintaxis a los datos para su transmisión en la red.

- Determina la forma de presentación de los datos sin preocuparse de su significado o semántica.
- Establece independencia a los procesos de aplicación considerando las diferencias en la representación de datos.
- Proporciona servicios al nivel de aplicaciones para interpretar el significado de los datos intercambiados.
- Opera el intercambio.
- Opera la visualización.

Nivel Aplicación: este nivel implementa servicios al usuario final.

- Proporciona comunicación entre dos procesos de aplicación, tales como: programas de aplicación, aplicaciones de red, etc.
- Proporciona aspectos de comunicaciones para aplicaciones específicas entre usuarios de redes: manejo de la red, protocolos de transferencias de archivos, etc.

APÉNDICE B

MANUAL DE USUARIO

Este apéndice se presenta como un manual de usuario, el cual está constituido de dos partes, éstas son: interfaz termoplúviógrafo-computadora-usuario y la configuración a través de las teclas de operación.

Interfaz termoplúviógrafo-computadora-usuario

Esta interfaz permite la programación del sistema de registro y la obtención de los datos de precipitación y temperatura almacenados.

Estando terminado nuestro programa, pasa a ser un programa de aplicación. Se puede distribuir este programa hacia otras computadoras para que en éstas se pueda operar de la misma manera.

Para la distribución de este programa se requiere crear un instalador, para la creación de éste, *Visual Basic* contiene un asistente, el cual realiza un procedimiento de empaquetado y distribución (*Package and deployment Wizard*) proporcionando los archivos necesarios para su operación. El asistente realiza todo el procedimiento hasta obtener un disco de distribución, el cual podemos instalar en cualquier otra computadora que contenga *Windows* y de esta manera operar el programa de comunicación. En la figura B.1 se muestra el lugar donde se encuentra el asistente.

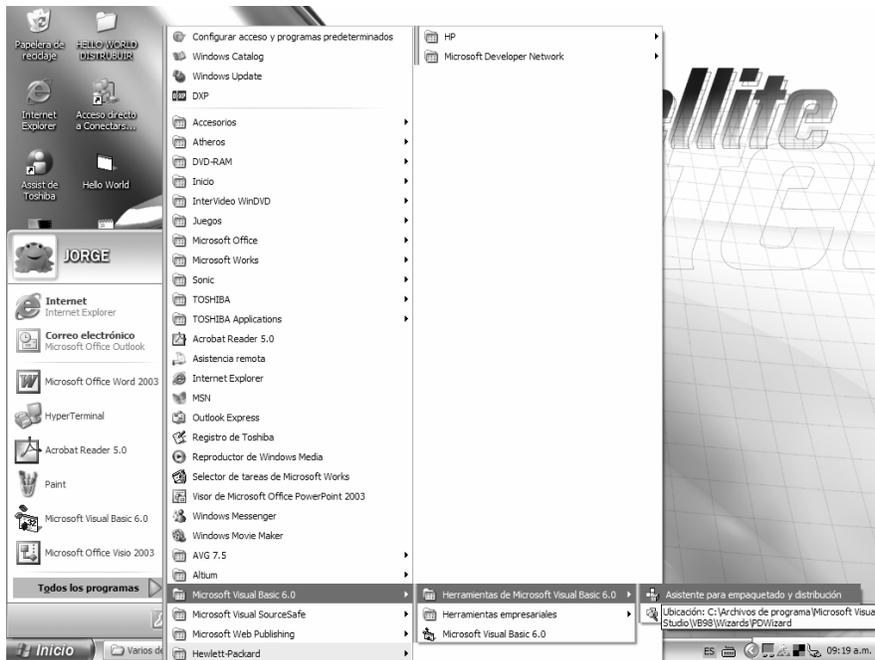


Figura B.1. Asistente de empaquetado y distribución.

Una vez que el programa está instalado en el equipo de cómputo, se puede establecer una comunicación serie con el termoplúviógrafo; el usuario puede acceder al programa a través de un acceso directo instalado en el escritorio, como lo muestra la figura B.2; o bien por medio del menú de programas, como se muestra en la figura B.3.

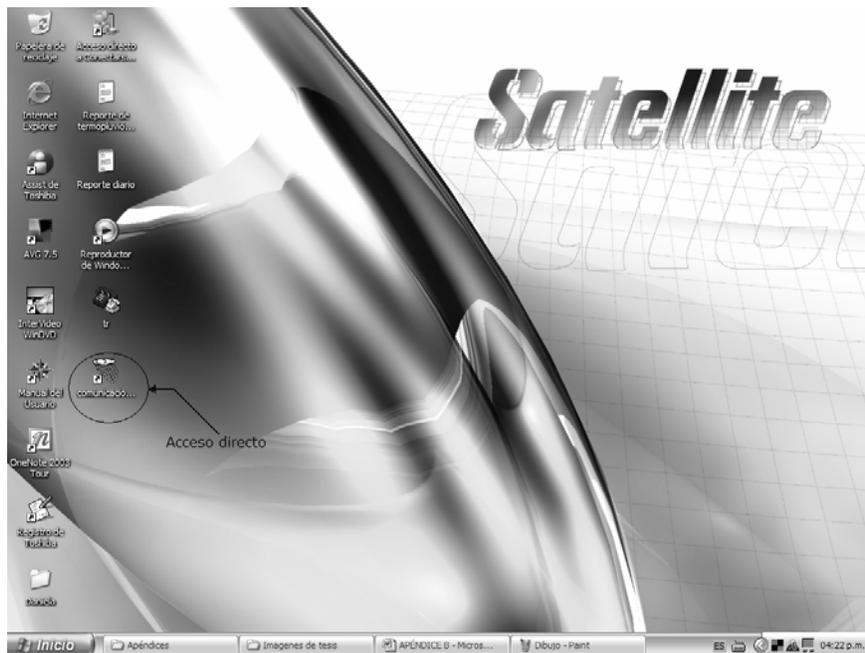


Figura B.2. Entrada al programa a través de un acceso directo.

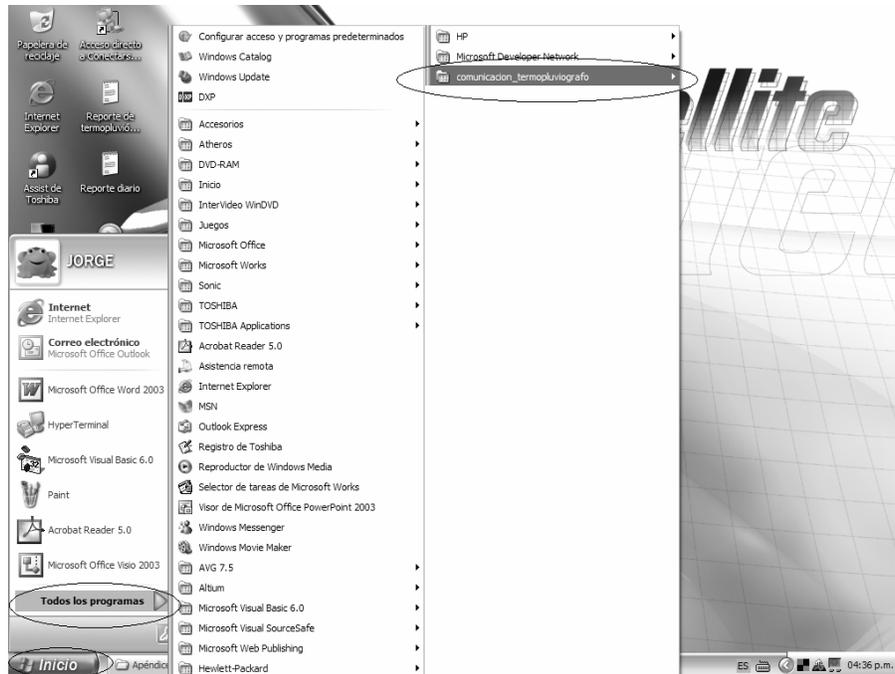


Figura B.3. Entrada al programa a través del menú de programas.

El procedimiento de comunicación que van a realizar el equipo de cómputo y el termopluiógrafo, se hace de la siguiente manera: primeramente se debe realizar la conexión física a través del estándar EIA-232 entre el termopluiógrafo y el equipo de cómputo, después se cierra un *switch* localizado en el termopluiógrafo, de esta manera queda establecida únicamente la conexión entre los dispositivos involucrados. Es importante mencionar que cuando se establece la conexión, el visualizador alfanumérico del termopluiógrafo presenta un letrero como el que se muestra en la figura B.4.

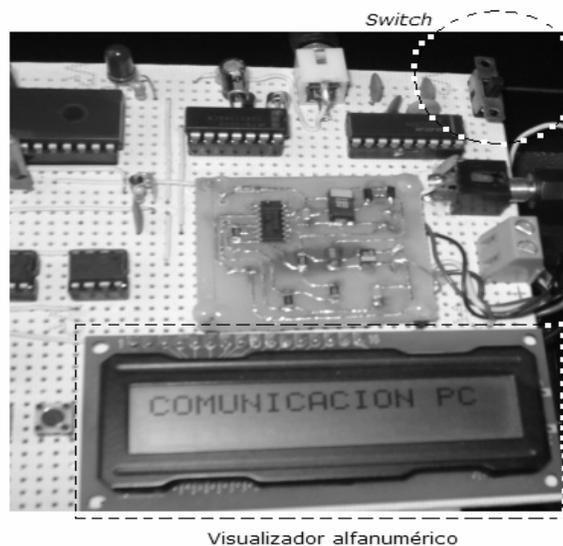


Figura B.4. Conexión a la PC.

Cuando es seleccionado el logotipo del programa, aparece la ventana principal, como se muestra en la figura B.5. En el cuadro se observan dos botones, uno es para configurar el puerto de comunicación y el siguiente es para iniciar la comunicación con el termoplúviógrafo.

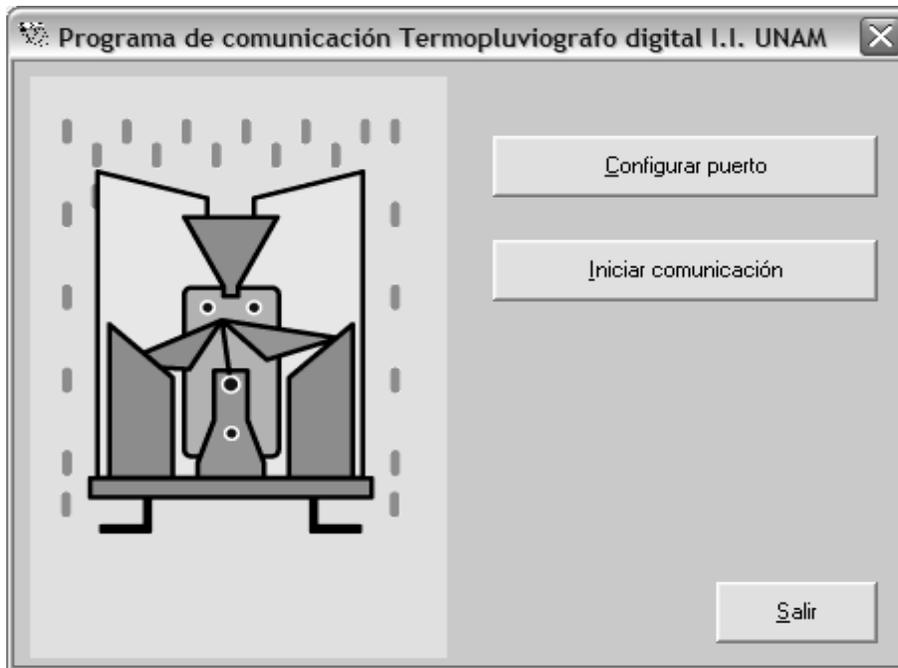


Figura B.5. Formulario principal del programa.

Primeramente se debe seleccionar el botón *Configurar puerto*, una vez que es seleccionado pueden suceder dos opciones:

- Si el termoplúviógrafo digital no está conectado a la computadora, mediante su cable de conexión, aparece una ventana que indicará *Error en la comunicación* (Figura B.6).

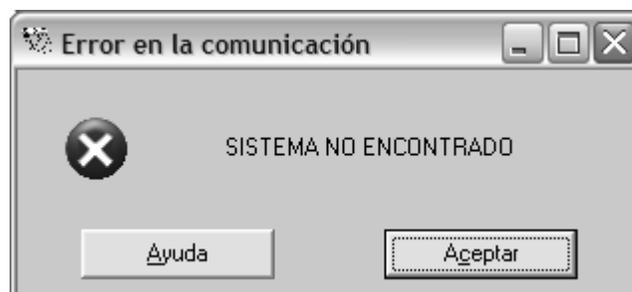


Figura B.6. Error de comunicación serie.

- Si se encuentra conectado el termoplúviógrafo con la computadora, a través de su cable, aparece una ventana con el

nombre de: *Selección de estándar de comunicación* (Figura B.7), en la que se seleccionará el estándar de comunicación por el cual se va a transmitir la información.

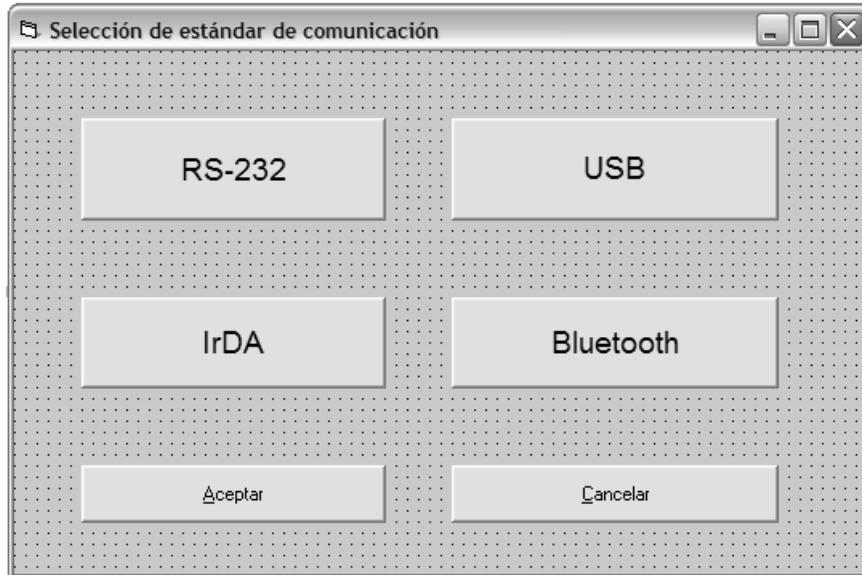


Figura B.7. Selección de estándar de comunicación.

- Una vez que se ha elegido un botón, se abre una ventana de *Configuración del puerto serie*, en la cual se va a indicar la configuración del puerto de comunicación por usar, tal como se muestra en la figura B.8.



Figura B.8. Configuración del puerto de comunicación serie.

Cuando se elige el puerto de comunicación existen dos opciones:

- Si se selecciona un puerto de comunicación al cual no se puede acceder, aparece la ventana *Error de comunicación*, como se muestra en la figura B.9, anunciando que el puerto elegido no puede abrirse y por lo tanto será necesario elegir otro.

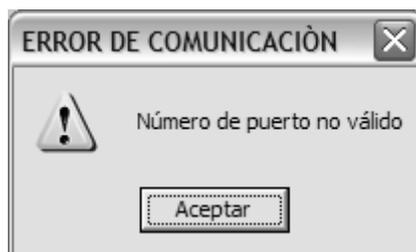


Figura B.9. Error en la elección del puerto de comunicación serie.

- Si se elige el puerto de comunicación correcto, aparecerá nuevamente la figura B.5, en la cual se pulsará el botón *Iniciar comunicación*, con lo que de esta manera el termoplúviógrafo y la computadora podrán iniciar el intercambio de información.

Al seleccionar el botón *Iniciar configuración* aparecerá la ventana *Comunicación establecida* (Figura B.10), que mostrará dos botones de selección, los cuales son: *Adquisición de datos* y *Parámetros de operación*, que realizarán la tarea de adquisición de datos y la configuración de parámetros de operación respectivamente.

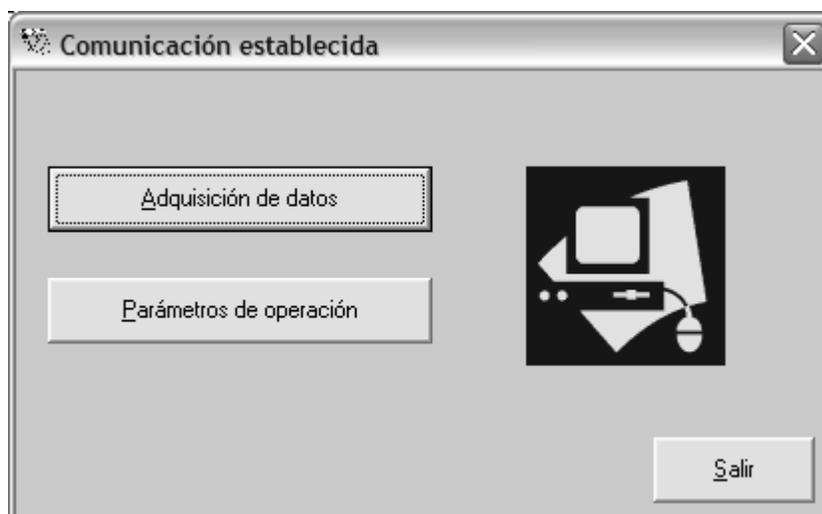


Figura B.10. Ventana de Comunicación establecida.

Cabe comentar que mediante la ventana *Parámetros de operación* se realiza la configuración, revisión o modificación de éstos. Al seleccionar este botón, se observa la ventana *Parámetros de operación del Termopluviógrafo digital* mostrada en la figura B.11, con los parámetros bajo los cuales está trabajando el sistema de registro.

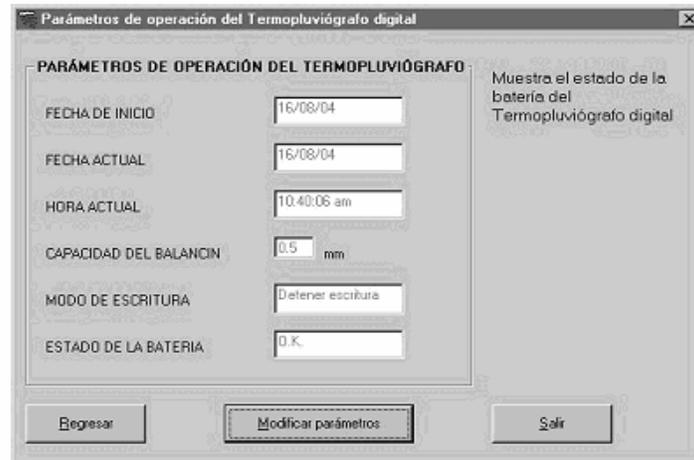


Figura B.11. Ventana de parámetros de operación.

Se observan en la ventana los siguientes parámetros: la fecha de inicio o bien la fecha de la última modificación de los parámetros de operación del sistema de registro, la fecha actual, la hora actual del termopluviógrafo digital (la cual es la hora en que se estableció la comunicación con el sistema de registro), la capacidad del balancín del pluviómetro, el modo de registro en memoria y el estado de la batería.

Si se desea realizar la configuración o modificación de los parámetros del termopluviógrafo, bastará con hacer la selección del botón *Modificar parámetros* y la ventana *Modificar parámetros de operación* aparecerá, como se muestra en la figura B.12.

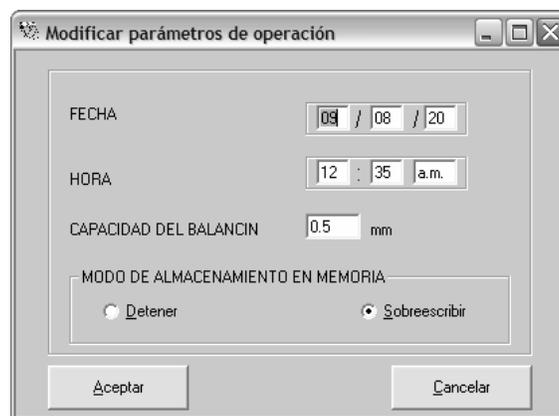


Figura B.12. Ventana de modificación de parámetros.

La parte que muestra la *fecha* y la *hora*, indica la fecha y la hora del sistema del equipo de cómputo, si éstas no son correctas, se pueden modificar directamente sin afectar la fecha y la hora del sistema de la computadora. La capacidad del balancín será siempre en milímetros y corresponde a la capacidad indicada en la placa de datos del pluviómetro a utilizar. Este valor está colocado por *default* con 0.5 mm.

Si se selecciona el botón *Detener*, que está contenido en el *modo de almacenamiento en memoria*, indica que cuando se hayan registrado 20,000 datos de eventos, el sistema de registro detendrá la escritura en la memoria de datos pluviales. Si se elige *Sobreescribir*, indica que después de haberse registrado 20,000 datos de eventos, se continuará con el registro sobrescribiendo en los datos más antiguos, registrados en la memoria de datos pluviales del termopluviógrafo.

Una vez que se han realizado las modificaciones necesarias, se deberá seleccionar el botón *Aceptar*, para que los parámetros modificados sean transmitidos al sistema de registro. Posterior a esta última acción aparece la ventana *Comunicaciones*, como se muestra en la figura B.13, donde preguntará si se desea o no que los datos almacenados anteriormente en la memoria de datos pluviales permanezcan.

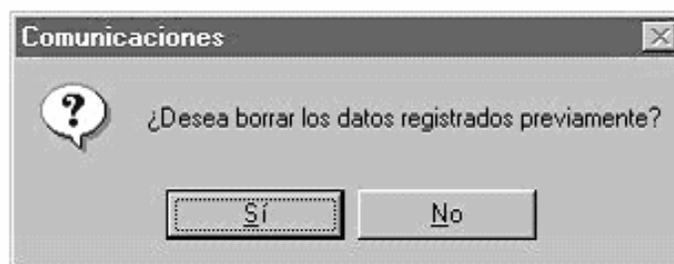


Figura B.13. Opción de permanencia de datos de eventos.

Con los pasos anteriores se ha llevado a cabo la configuración o modificación de parámetros de operación del termopluviógrafo digital. A continuación se explica la manera de adquirir los datos de eventos registrados en la memoria de datos pluviales.

Si se selecciona el botón *Adquisición de datos* de la figura B.10, se presentará una ventana (*Descargando datos*), que indica que la transferencia de datos desde el sistema de registro al equipo de cómputo se está llevando a cabo, figura B.14.



Figura B.14. Adquisición de datos.

Cuando se han transferido los datos del termopluviógrafo hacia la computadora, aparece la ventana *Presentación de datos en diferentes formatos*, como se muestra en la figura B.15. Esta ventana indica que los datos colectados se pueden presentar al usuario en tres formatos diferentes, los cuales se eligen a través de los tres botones que están en ésta, cuyos nombres son: *Abrir datos con Excel*, *Guardar datos en formato de texto* y *Revisar gráficas*.

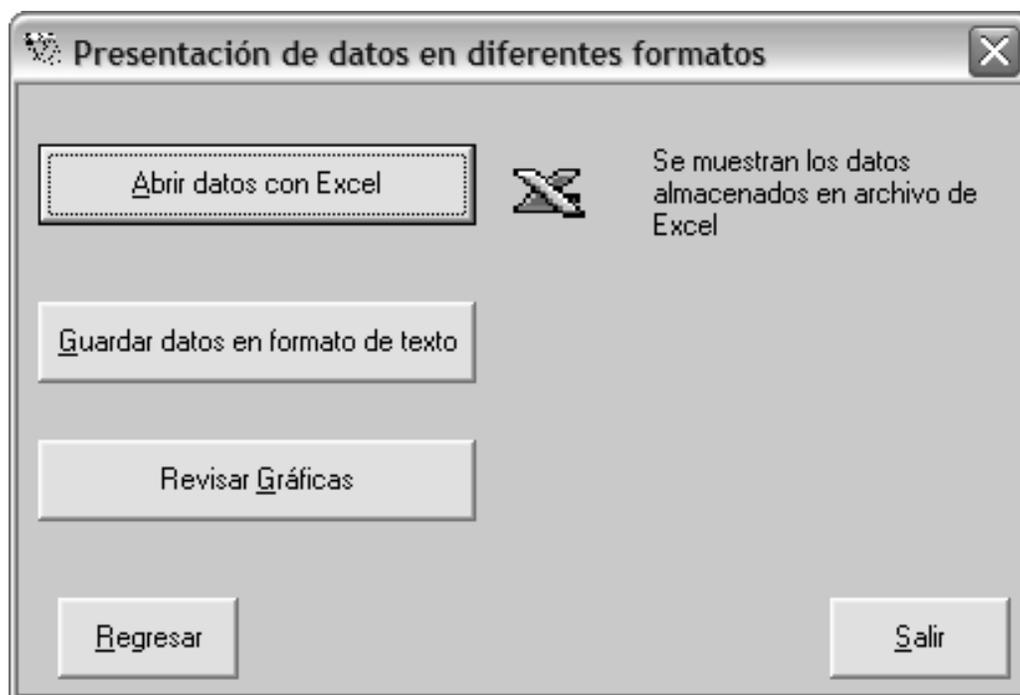


Figura B.15. Presentación de datos en diferentes formatos.

El botón *Abrir datos con Excel* muestra el formato en que la información puede ser presentada en un procesador de hojas de cálculo, tal como Excel. Los datos se organizan por columnas, donde cada columna representa un día de registro de datos. La figura B.16 presenta un ejemplo.

TERMOPLUVIOGRAFO DIGITAL II UNAM						
Fecha de inicio		28/08/07				
Capacidad del balancin		0.5mm				
Número de eventos		25				
Precipitación acumulada		12.5mm				
Registros de precipitación puntual						
28/08/07		29/08/07		30/08/07		
Hora del evento	Temp. Ext.	Hora del evento	Temp. Ext.	Hora del evento	Temp. Ext.	
11:50:42 a.m.	28°C	11:18:26 a.m.	27°C	10:58:32 p.m.	24°C	
11:50:48 a.m.	28°C	03:56:48 p.m.	27°C	.5mm		
11:50:54 a.m.	28°C	04:01:46 p.m.	26°C			
11:51:36 a.m.	28°C	04:06:14 p.m.	27°C			
05:04:24 p.m.	28°C	04:10:06 p.m.	26°C			
05:04:26 p.m.	28°C	04:17:04 p.m.	26°C			
3mm		09:17:42 p.m.	25°C			
		09:36:06 p.m.	25°C			
		10:21:32 p.m.	24°C			
		10:25:18 p.m.	24°C			
		10:28:48 p.m.	24°C			
		10:30:56 p.m.	24°C			
		10:33:46 p.m.	24°C			
		10:42:08 p.m.	24°C			
		7mm				

Figura B.16. Presentación de datos en formato de hoja de cálculo.

Si se selecciona el botón *Guardar datos en formato de texto*, se presenta otra forma con la que se puede revisar los datos de los eventos ocurridos, almacenándolos en un archivo con formato de texto, como se presenta en la figura B.17.

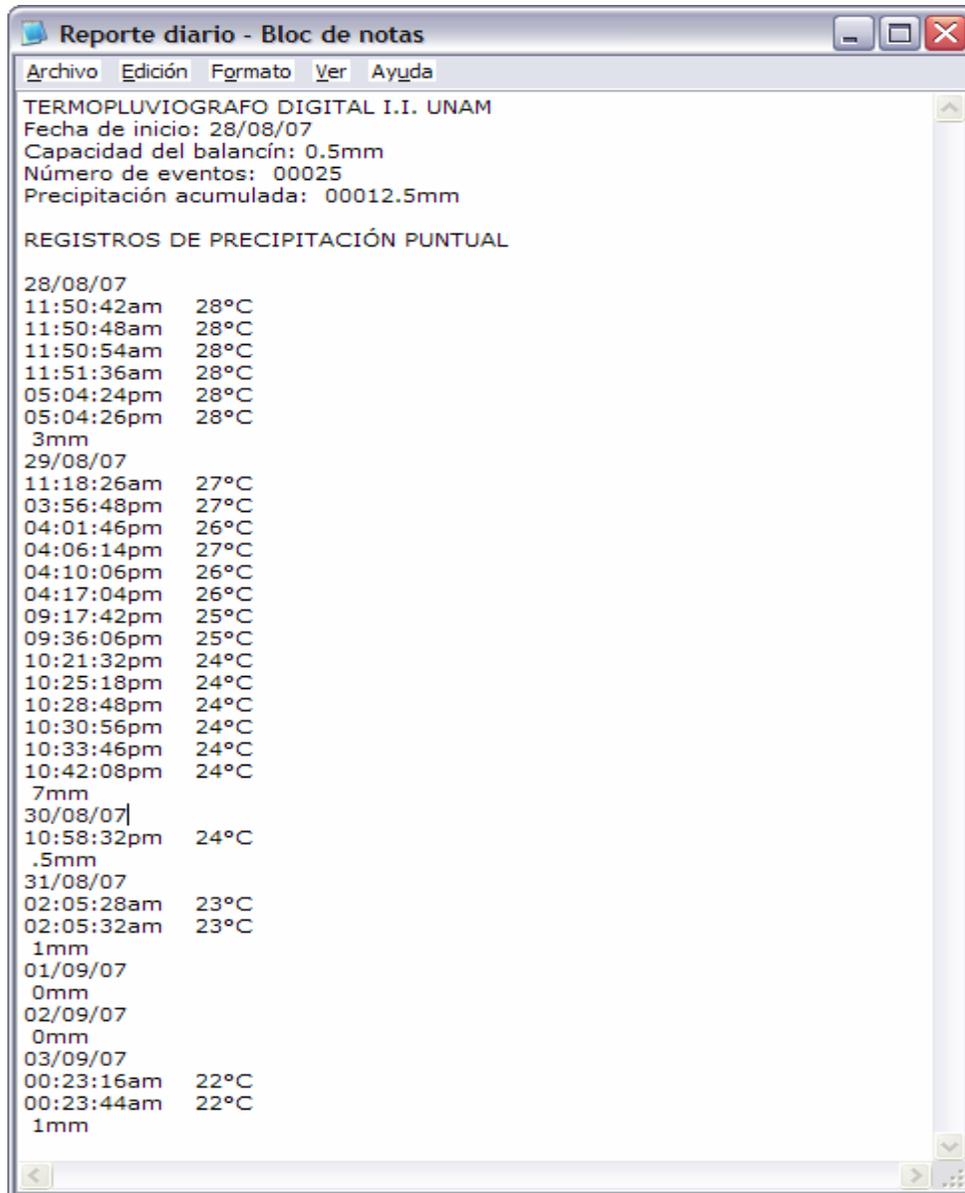


Figura B.17. Presentación de datos en formato de archivo de texto.

Los dos formatos anteriormente descritos en que pueden consultarse los datos de eventos son de manera puntual, es decir, se observa el dato del evento registrado por hora, minuto y segundo, así como la temperatura ambiente al momento del evento.

Si se elige el botón *Revisar gráficas*, se obtiene un formato en el que puede consultarse los datos en forma de gráficas. Si se selecciona este formato de presentación de datos, aparece la ventana *Intervalo de días a graficar*, observada en la figura B.18, en la cual se debe establecer el intervalo de días que se desea graficar.

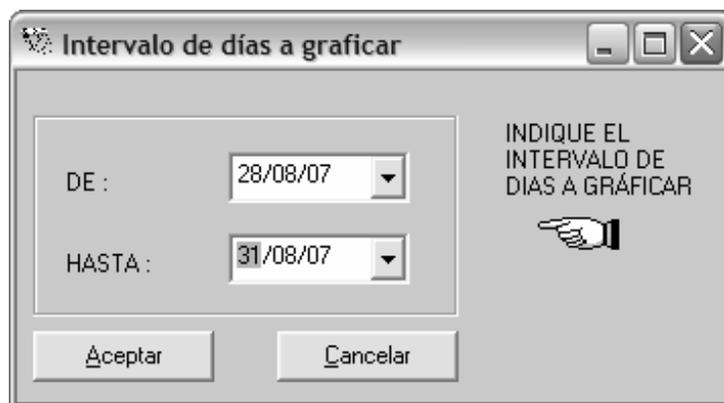


Figura B.18. Intervalo de días por graficar.

Una vez que se ha establecido el intervalo de días a graficar, se selecciona *Aceptar* y aparecerá la ventana *Opción de gráficas*, mostrada en la figura B.19, la cual permite la elección de una gráfica entre un *Hietograma* o una *Curva de masas*.



Figura B.19. Opción de gráfica.

Si se selecciona el botón *Hietograma*, presenta una gráfica que relaciona la intensidad de la lluvia contra un intervalo de tiempo, como se observa en la figura B.20.

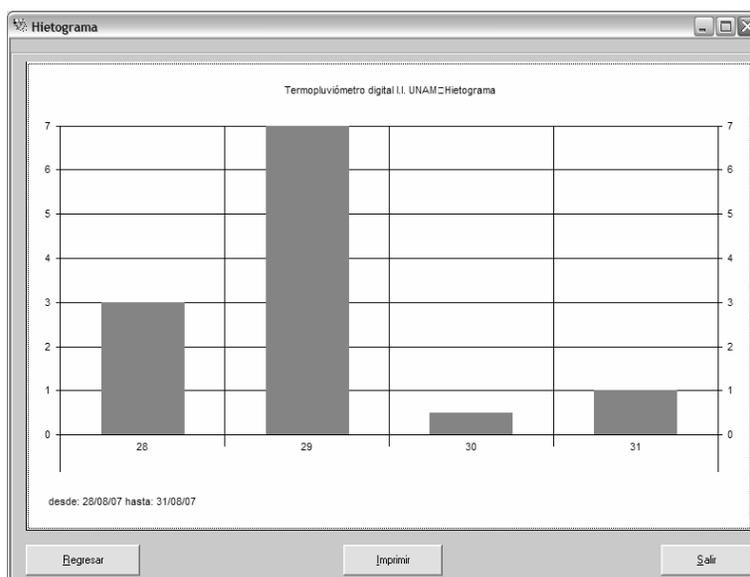


Figura B.20. Presentación de datos en un Hietograma.

Si se selecciona el botón *Curva de masas*, se graficará la precipitación acumulada contra el tiempo, como se muestra en la figura B.21.

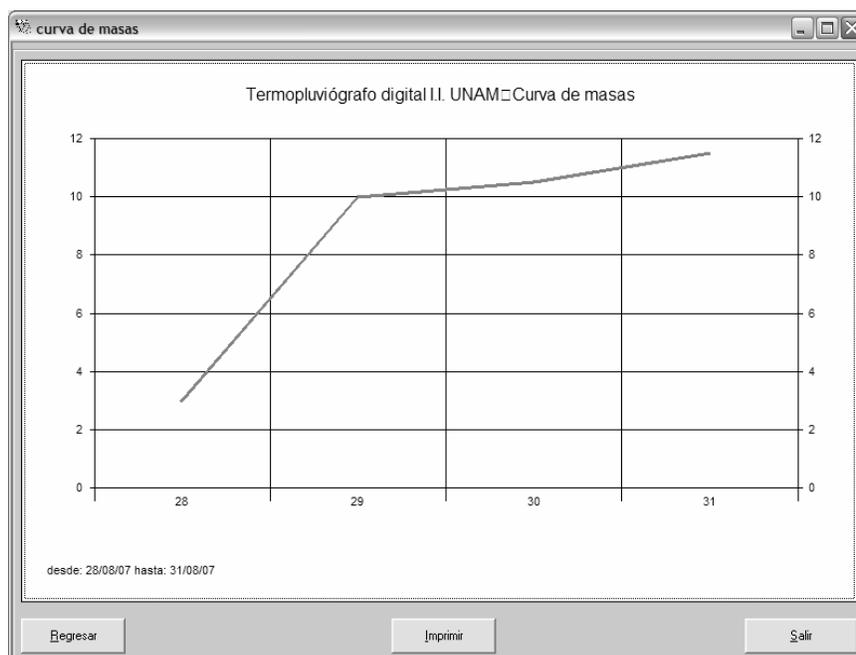


Figura B.21. Presentación de datos en una curva de masas.

En la ventana *Intervalo de días a graficar* (Figura B.18), también se puede establecer la presentación de gráficas por día, donde se muestran los datos ocurridos en las horas del día seleccionado.

La figura B.22 muestra un ejemplo al presentar los datos en el formato gráfico, donde se selecciona un Hietograma de un día especificado.

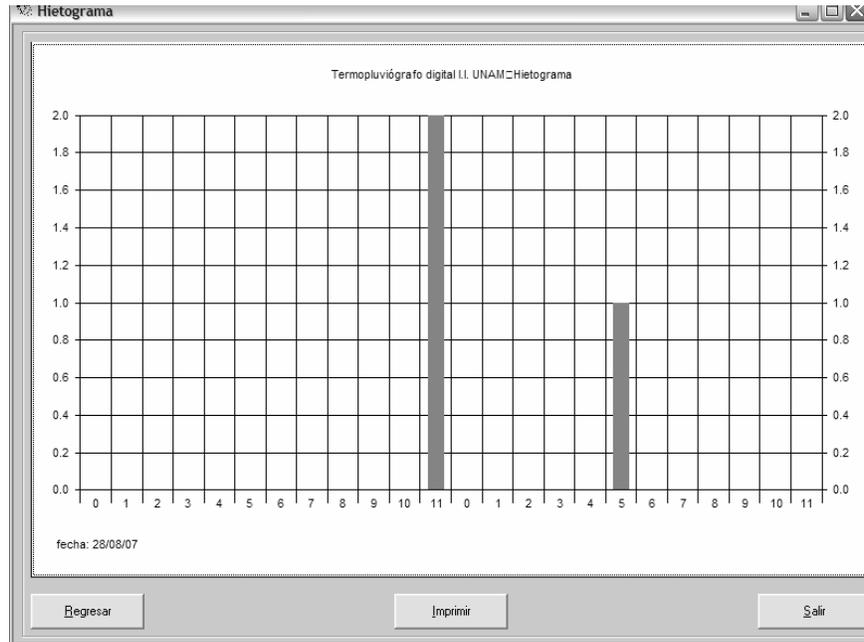


Figura B.22. Hietograma de un día.

La figura B.23 muestra la curva de masas para este mismo día determinado.

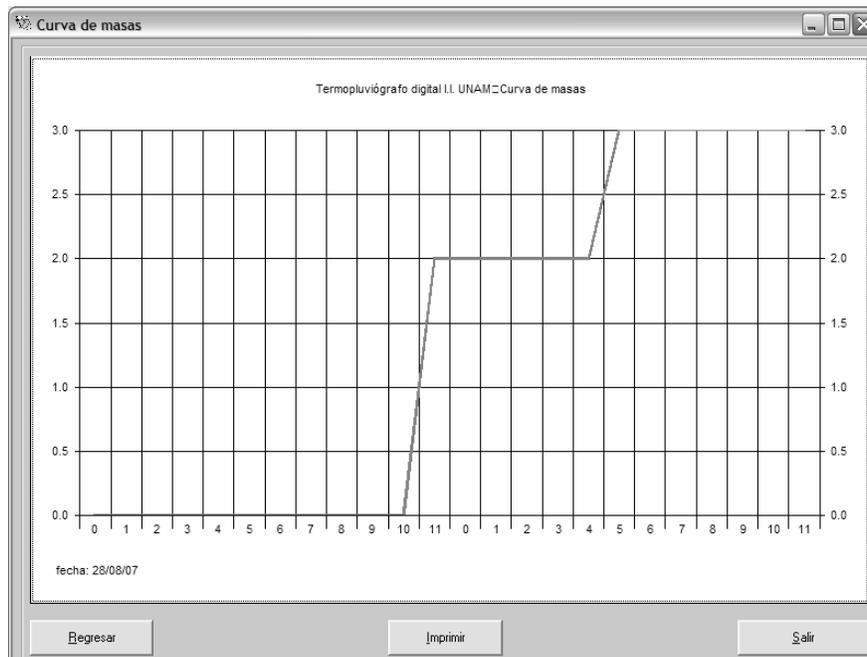


Figura B.23. Curva de masas de un día.

En las opciones antes mencionadas, se cuenta con un botón *Imprimir*, el cual se utiliza para realizar una impresión de la gráfica que interese al usuario.

Para terminar la transferencia y procesamiento de datos, así como salir del entorno de comunicación, es suficiente seleccionar el botón *Salir*.

Una observación importante es la siguiente: cuando el termopluviógrafo digital va a ser utilizado por primera vez, es necesario que la configuración de los parámetros de operación de éste se realice mediante el programa instalado en el equipo de cómputo. Esta operación se debe de realizar también cuando se ha cambiado la batería.

Hasta esta parte se ha visto el procedimiento de operación del *software* instalado en la computadora, ahora procederemos a explicar la configuración de parámetros de operación utilizando las teclas de operación, las cuales están integradas en el termopluviógrafo digital.

Configuración a través de las teclas de operación

El usuario también tiene la opción de realizar modificaciones a los parámetros de operación a través de las teclas de operación del termopluviógrafo, estas teclas se muestran en la figura B.24.

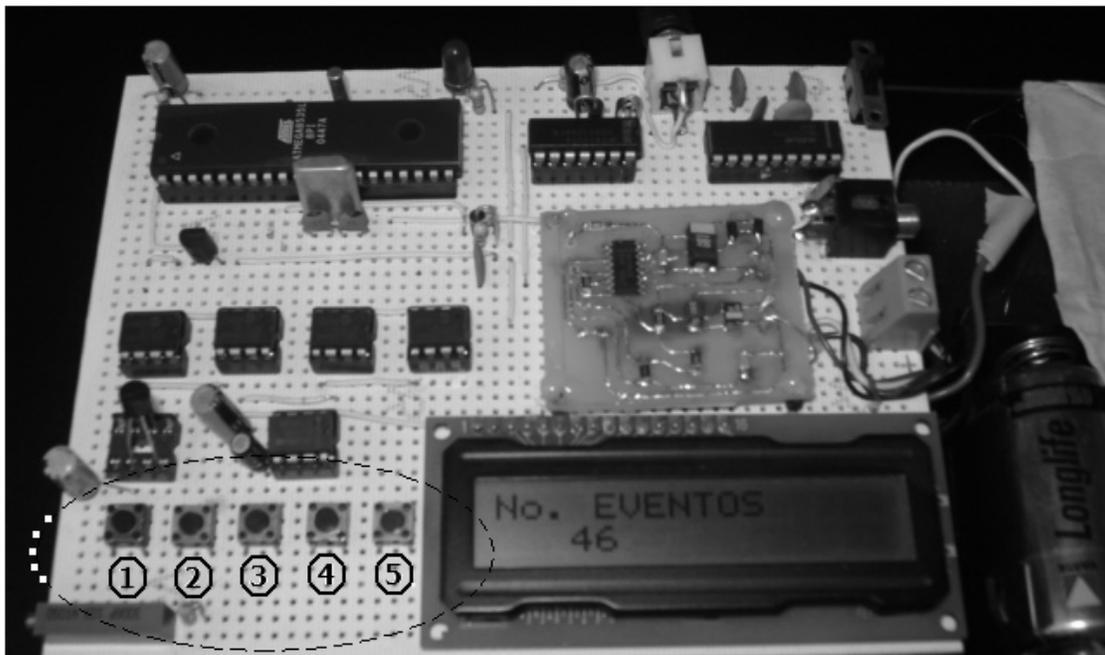


Figura B.24. Teclas de operación del termopluviógrafo digital. (Zona punteada)

La función de las teclas de operación se muestra en la tabla B.1.

NÚMERO	NOMBRE DE TECLA	FUNCIÓN
1	SEL	Muestra el cursor para indicar que se puede modificar el parámetro en cuestión
2	DESP	Con esta tecla se puede desplazar en forma cíclica a través de los diversos letreros que muestra el termoplúviógrafo digital
3	SET/UP	Con esta tecla se activa el visualizador alfanumérico y se saca al microcontrolador del modo de bajo consumo. También se utiliza para incrementar en uno el valor de un dato a modificar
4	SHIFT	Esta tecla se emplea para recorrer el valor del elemento resaltado
5	ENTER	Acepta la modificación de datos

Tabla B.1. Función de las teclas de operación.

Si el termoplúviógrafo no se encuentra conectado a una computadora o está en operación de bajo consumo de energía y el usuario desea ver la información contenida en él, o modificar los parámetros de operación, se realiza el siguiente procedimiento de operación con las teclas del termoplúviógrafo: se presiona la tecla *SET/UP* durante algunos segundos, enseguida aparece un letrero en el visualizador que identifica al mismo, como se observa en la figura B.25.

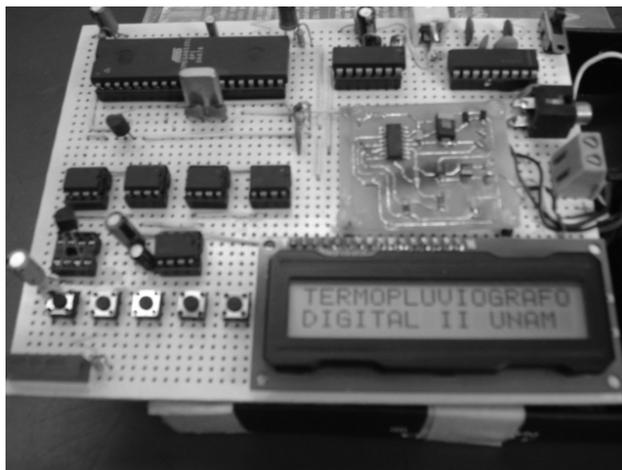


Figura B.25. Letrero que identifica al sistema de registro.

Posteriormente, a través de la tecla de operación *DESP*, el usuario puede desplazarse a través de los diversos letreros que el termoplúviógrafo muestra. En la figura B.26 se muestran los letreros disponibles cuando se presiona de manera sucesiva dicha tecla.

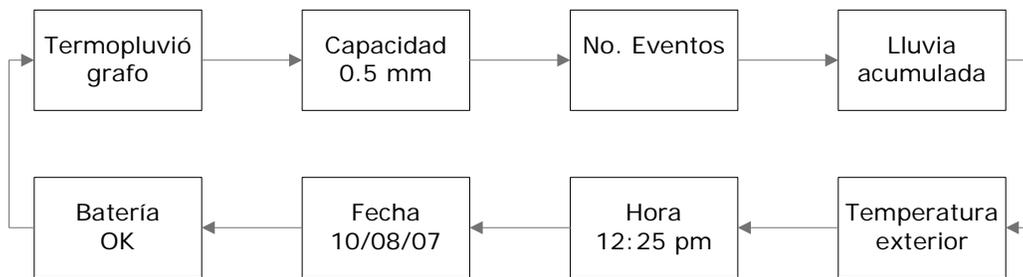


Figura B.26. Letreros del sistema de registro.

Con ayuda de las teclas de operación se pueden realizar modificaciones a los parámetros con los cuales opera el termopluviógrafo digital. Sólo es posible modificar las siguientes opciones: capacidad del balancín, hora y fecha. Como un ejemplo, vamos a modificar el valor de capacidad del balancín, de 0.5 mm a 0.2 mm.

Para realizar la modificación del parámetro de operación del termopluviógrafo mediante las teclas de operación se deben seguir los siguientes pasos.

- Si el termopluviógrafo se encuentra en modo de operación de bajo consumo o sin conexión, se debe de mantener presionada la tecla *SET/UP* hasta que el visualizador alfanumérico muestre el mensaje que identifica al termopluviógrafo, como se mostró en la figura B.25.
- Con ayuda de la tecla *DESP* desplazarse a través de los letreros hasta el letrero que muestre el parámetro que se desea modificar, para nuestro ejemplo seleccionamos el letrero capacidad (Figura B.27).

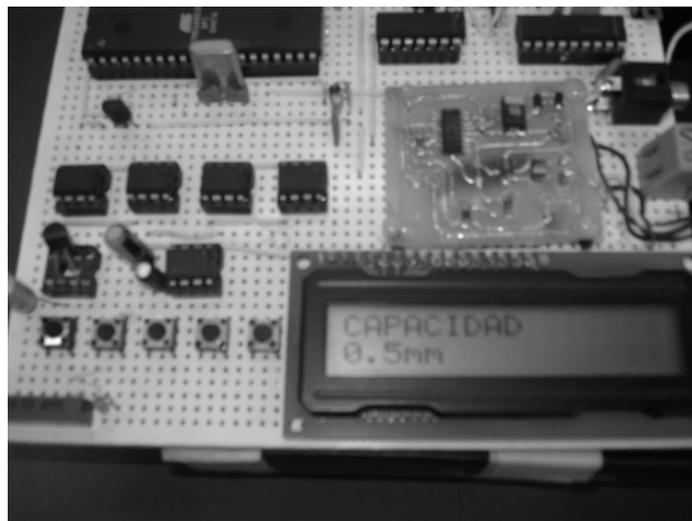


Figura B.27. Modificación del letrero capacidad.

- Con la tecla *SEL* se muestra el cursor en el primer dígito del parámetro a modificar.
- Para desplazarse a través del dato a modificar se presiona la tecla *SHIFT*.
- El dígito del dato a modificar se incrementa en uno presionando la tecla *SET/UP*.
- Si el dato ha tomado los nuevos valores es necesario presionar la tecla *ENTER*, para que éstos sean actualizados en la memoria de programa del microcontrolador, quedando finalmente modificado el letrero capacidad del balancín (Figura B.28).

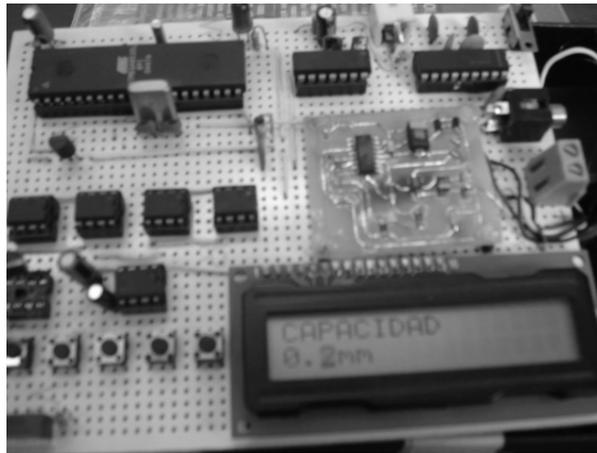


Figura B.28. Letrero de capacidad de balancín modificado.

Después de que hayan transcurrido dos minutos desde que el usuario terminó de usar las teclas de operación, el microcontrolador entrará en modo de operación de bajo consumo de energía. Para ello el visualizador alfanumérico se apagará y el termoplúviógrafo estará a la espera de que un evento ocurra.

APÉNDICE C

SOFTWARE

Este apéndice está compuesto de dos partes: en la primera se presenta un listado parcial del programa desarrollado en lenguaje ensamblador; en la segunda, se muestra un listado también parcial del programa desarrollado en lenguaje *Visual Basic*.

Lenguaje ensamblador. Este programa contiene las instrucciones con las que el microcontrolador ejecuta las tareas encomendadas, las cuales son necesarias para el funcionamiento adecuado del termopluviógrafo digital.

```
; Programa: TERMOPLUVIÓGRAFO DIGITAL
; Programa de comunicación del Termopluviógrafo digital
; con visualizador alfanumérico de cristal líquido LCD 16x2
; y teclas de operación tipo push button.
;
; Asignación de terminales
; PA0 ---> BATERIA      PB0 ---> OFFLCD      PC0 ---> LED        PD0 ---> RXD
; PA1 ---> TEMPERATURA  PB1 ---> CS1        PC1 --->            PD1 ---> TXD
; PA2 --->              PB2 ---> CS2        PC2 ---> INVALID   PD2 ---> EVENTO
; PA3 --->              PB3 ---> CS3        PC3 ---> RS         PD3 ---> S1
; PA4 ---> DB4          PB4 ---> CS4        PC4 ---> R/W       PD4 ---> S2
; PA5 ---> DB5          PB5 ---> MOSI       PC5 ---> E          PD5 ---> S3
; PA6 ---> DB6          PB6 ---> MISO       PC6 ---> TOSC1     PD6 ---> S4
; PA7 ---> DB7          PB7 ---> SCK        PC7 ---> TOSC2     PD7 ---> S5
```

```
;*****
```

```

.include "m8535def.inc"

.org $000
    rjmp  RESET                ;reset handler
.org $001
    rjmp  EXT_INT0            ;IRQ0 Handler Interrupción externa
.org $004
    rjmp  TIM2_OVF           ;Timer2 Overflow Handler
.org $00B
    rjmp  UART_RXC           ;UART RX Complete Handler
;*****
;Definición de registros utilizados en el programa
.def    temp    = r16        ;registro temporal de uso general
.def    contador_sleep = r10
;registros utilizados en conversión BCD2Binario
.def    tbin    =r20        ;resultado binario
.def    fBCDL  =r20        ;primer digito BCD
.def    fBCDH  =r21        ;segundo digito BCD
;definición de registro para conversión binario2BCD
.def    fbin    =r20        ;valor binario de 8-bit
.def    tBCDL  =r20        ;resultado MSD de BCD
.def    tBCDH  =r21        ;resultado LSD de BCD
;definición de registros para conversión binario 2 BCD 5
.equ    AtBCD0 =13         ;dirección de tBCD0
.equ    AtBCD2 =15         ;dirección de tBCD1
.def    tBCD0  =r13        ;valor de los digitos 1 y 0 de BCD
.def    tBCD1  =r14        ;valor de los digitos 3 y 2 de BCD
.def    tBCD2  =r15        ;valor de digito 4 de BCD
.def    fbinL  =r20        ;valor binario de Low byte
.def    fbinH  =r21        ;valor binario de High byte
.def    cnt16a =r22        ;contador de lazo
.def    tmp16a =r23
;definición de registros para conversión 24bits a BCD
.def    fbin_0 =r16        ;valor binario byte 0 (LSB)
.def    fbin_1 =r17        ;valor binario byte 1
.def    fbin_2 =r18        ;valor binario byte 2 (MSB)
.def    tBCD_0 =r20        ;BCD value digits 0 and 1
.def    tBCD_1 =r21        ;BCD value digits 2 and 3
.def    tBCD_2 =r22        ;BCD value digits 4 and 5
.def    tBCD_3 =r23        ;BCD value digits 6 and 7 (MSD)
;definición de registros para LCD
.def    temp1  =r21        ;registro temporal de subrutina LCD
.def    contador =r22      ;registro de subrutina LCD
.def    ddram  =r23        ;registro de subrutina LCD
.def    posición =r24      ;registro de subrutina LCD
;definición de registros para EEPROM interna escritura y lectura
.def    EEdata =r16        ;data byte to write to EEPROM
.def    EElow  =r17        ;address low byte to write to
.def    EEhigh =r18        ;address high byte to write to
;definición de registros para sistema SPI
.equ    SS1    = PB4        ;pin SS del sistema SPI
.equ    SS2    = PB3        ;pin SS del sistema SPI
.equ    SS3    = PB2        ;pin SS del sistema SPI

```

```

.equ   SS4   = PB1           ;pin SS del sistema SPI
;Variables utilizadas durante la conversión Horas _ Minutos*
.def   mc8u  =r16           ;multiplicando
.def   mp8u  =r17           ;multiplicador
.def   m8uL  =r17           ;result Low byte
.def   m8uH  =r18           ;result High byte
.def   mcnt8u=r19          ;loop counter
;Variables utilizadas durante en subrutina conversion_segundos
.def   mc16uL=r16          ;multiplicand low byte
.def   mc16uH=r17          ;multiplicand high byte
.def   mp16uL=r18          ;multiplier low byte
.def   mp16uH=r19          ;multiplier high byte
.def   m16u0=r18           ;result byte 0 (LSB)
.def   m16u1=r19           ;result byte 1
.def   m16u2=r20           ;result byte 2
.def   m16u3=r21           ;result byte 3 (MSB)
.def   mcnt16u=r22         ;loop counter
;Variables utilizadas durante la conversión_HH_MM_SS
.def   drem16uL=r14        ;
.def   drem16uH=r15        ;
.def   dres16uL=r16        ;
.def   dres16uH=r17        ;
.def   dd16uL=r16          ;
.def   dd16uH=r17          ;
.def   dv16uL=r18          ;
.def   dv16uH=r19          ;
.def   dcnt16u=r20         ;
;*****
RESET:
    ldi    temp,low(RAMEND)    ; inicialización del Stack Pointer
    out    SPL,temp
    ldi    temp,high(RAMEND)
    out    SPH,temp
;*****
;Configuración de puertos y registros de control
;*****
;Configuración de GICR General Interrupt Mask Register
                                ;GICR: INT1 INTO 5 4 3 2 1 0
    ldi    temp,0b01000000    ;      0  1  0 0 0 0 0 0
    out    GICR,temp          ;habilita interrupción externa 0
;Configuración de TIMSK Timer/Counter Interrupt Mask Register
                                ;TIMSK: 7 TOIE2 5 4 3 2 1 0
    ldi    temp,0b01000000    ;      0  1  0 0 0 0 0 0
    out    TIMSK,temp         ;habilita TOIE2
;Configuración de MCUCR del MCU
                                ;MCUCR: - SE SM1 SM0 ISC11 ISC10 ISCO1 ISCO0
    ldi    temp,0b01000000    ;      0 0 0 0 0 0 0 0
    out    mcucr,temp         ;interrupciones INTO en estado bajo
;Configuración de TCCR2 Timer/Counter2 control register
                                ;TCCR2: 7 6 5 4 3 CS22 CS21 CS20
    ldi    temp,0b00000110    ;      0 0 0 0 0 1 1 0
    out    TCCR2,temp         ;carga preescalador CK=256 Overflow c/2sec
;Configuración de Timer Counter2 TCNT2

```

```

ldi    temp,00          ;carga al timer con valor inicial = $00
out    TCNT2,temp
;Configuración de ASSR
;ASSR: 7 6 5 4 AS2 TCN2UB OCR2UB TCR2UB
ldi    temp,0b00001000 ; 0 0 0 0 1 0 0 0
out    ASSR,temp       ;selecciona fuente de reloj con AS2=1
;Configuración del sistema SPI para memoria EEPROM de almacenamiento de datos
;SPCR: SPIE SPE DORD MSTR CPOL CPHA SPR1 SPRO
ldi    temp,0b01010011 ; 0 1 0 1 0 0 1 1
out    SPCR,temp       ;SCK frequency = f/128
;Configuración de UART
;UCSRB: RXCIE TXCIE UDRIE RXEN TXEN CHR9 RXB8 TXB8
ldi    temp,0b10011000 ; 1 0 0 1 1 0 0 0
out    UCSRB,temp      ;Se habilita transmision, recepcion e interrupcion

ldi    temp,25          ;9.600bps@4MHz
out    UBRRL,temp      ;Baud Rate Register
;Configuración de registro de control y de estado ADCSRA
;ADCSRA: ADEN ADSC ADFR ADIF ADIE ADPS2 ADPS1 ADPS0
ldi    temp,0b00000101 ; 0 0 0 0 0 1 0 1
out    ADCSRA,temp     ;factor de división de 32=215kHz
;Configuración de DDRA para líneas de control de LCD y canales del ADC
;DDRA 7 6 5 4 0 0 ADC1 ADC0
ldi    temp,0b11110000 ; 1 1 1 1 0 0 0 0
out    DDRA,temp       ;Líneas del display, sensor de temperatura y nivel de bateria
;configuracion de DDRB y PORTB para sistema SPI
;DDRB SCK MISO MOSI SS1 SS2 SS3 SS4 offLCD
ldi    temp,0b10111111 ; 1 0 1 1 1 1 1 1
out    DDRB,temp       ;SCK=1, MOSI=1 Push Pull One Output
;PORTB SCK MISO MOSI SS1 SS2 SS3 SS4 offLCD
ldi    temp,0b01000000 ; 0 1 0 0 0 0 0 0
out    PORTB,temp      ;
;Configuración de DDRC y PORTC para líneas de datos de LCD y Led_Evento
;DDRC: TOSC2 TOSC1 E RW RS INVALID OFFTEMP LED
ldi    temp,0b00111001 ; 0 0 1 1 1 0 0 1
out    DDRC,temp       ;
sbi    portc,2          ;pinc2 como entrada de señal invalid
sbi    portc,0          ;pinc0 como salida para Led_Evento
;Configuración de DDRD y PORTD para Interrupción externa y push buttons
;DDRD sw5 sw4 sw3 sw2 sw1 INTO TXD RXD
ldi    temp,0b00000010 ; 0 0 0 0 0 0 0 1
out    DDRD,temp       ;
ldi    temp,0b11111101 ; 1 1 1 1 1 1 1 0
out    PORTD,temp      ;
sei                               ;se habilitan las interrupciones
;*****
;inicialización de registros
ldi    temp,0
sts    $0156,temp       ;permite escritura en memoria eeprom spi
sts    $0157,temp       ;indica que no existen datos sobreescritos
ldi    temp,$53         ;envia una "S"
sts    $0155,temp       ;permite sobrecribir los datos en spi
clr    contador_sleep   ;inicia en cero contador de tiempo de LCD

```

```

sbi portb,0 ;habilita regulador de VdcLCD
ldi temp,1 ;indica que la memoria en la que se están
sts $012F,temp ;escribiendo los datos es la memoria 1
;inicializa variables de hora, minuto y segundo cuando existe un reset
clr temp
sts $60,temp
sts $61,temp
sts $62,temp
;obtiene dato de capacidad desde memoria eeprom interna
clr temp ;almacena en localidades de SRAM
sts $0166,temp ;dirección en donde se escribirá la capacidad
ldi temp,$08 ;del balancín en eeprom interna
sts $0167,temp
rcall leer_memoria_eeprom_interna ;eeprom interna
sts $66,temp ;almacena en SRAM dato de capacidad anterior al reset
mov fbinL,temp ;carga en registro el valor de memoria eeprom
clr fbinH ;la parte alta del registro debe estar en blanco
rcall bin2BCD16 ;se realiza una conversión del dato en binario
rcall ASCII_precipitacion ;para poder mostrar el dato en el display
;los datos resultado de la conversión se encuentran en las localidades de SRAM
;$011F y $0120
;los datos a mostrar en la pantalla del display correspondientes a la capacidad del
;balancín son $98 y $9A
lds temp,$011F ;se almacena primer dígito de capacidad
sts $98,temp
lds temp,$0120 ;se almacena segundo dígito de capacidad
sts $9A,temp
;obtiene dato de número de eventos desde memoria interna
clr temp ;almacena en localidades de SRAM
sts $0166,temp ;dirección en donde se escribirá la capacidad
ldi temp,$09 ;del balancín en eeprom interna
sts $0167,temp
rcall leer_memoria_eeprom_interna ;eeprom interna
sts $67,temp ;almacena en SRAM dato de capacidad
anterior al reset
clr temp ;almacena en localidades de SRAM
sts $0166,temp ;dirección en donde se escribirá la capacidad
ldi temp,$0A ;del balancín en eeprom interna
sts $0167,temp
rcall leer_memoria_eeprom_interna ;eeprom interna
sts $68,temp ;almacena en SRAM dato de capacidad anterior al reset
rcall numero_eventos
;Obtiene dato de cantidad de precipitación acumulada desde eeprom interna
;El dato correspondiente a la precipitación acumulada se calcula considerando la
;capacidad del balancín y el número de eventos registrados
clr temp ;almacena en localidades de SRAM
sts $0166,temp ;dirección en donde se escribirá la capacidad
ldi temp,$0E ;del balancín en eeprom interna
sts $0167,temp
rcall leer_memoria_eeprom_interna ;eeprom interna
sts $69,temp ;almacena en SRAM dato de capacidad
anterior al reset
clr temp ;almacena en localidades de SRAM

```

```

sts    $0166,temp           ; dirección en donde se escribirá la capacidad
ldi    temp,$0F             ; del balancín en eeprom interna
sts    $0167,temp
rcall  leer_memoria_eeprom_interna ; eeprom interna
sts    $6A,temp            ; almacena en SRAM dato de capacidad anterior al reset
clr    temp                ; almacena en localidades de SRAM
sts    $0166,temp           ; dirección en donde se escribirá la capacidad
ldi    temp,$10            ; del balancín en eeprom interna
sts    $0167,temp
rcall  leer_memoria_eeprom_interna ; eeprom interna
sts    $6B,temp            ; almacena en SRAM dato de capacidad anterior al reset
; se obtienen datos correspondientes a la ultima fecha registrada en la memoria
; eeprom spi dato de dia
clr    temp                ; almacena en localidades de SRAM
sts    $0166,temp           ; dirección en donde se escribirá la capacidad
ldi    temp,$0B            ; del balancín en eeprom interna
sts    $0167,temp
rcall  leer_memoria_eeprom_interna ; eeprom interna
sts    $63,temp
; dato de mes
clr    temp                ; almacena en localidades de SRAM
sts    $0166,temp           ; dirección en donde se escribirá la capacidad
ldi    temp,$0C            ; del balancín en eeprom interna
sts    $0167,temp
rcall  leer_memoria_eeprom_interna ; eeprom interna
sts    $64,temp
; dato de año
clr    temp                ; almacena en localidades de SRAM
sts    $0166,temp           ; dirección en donde se escribirá la capacidad
ldi    temp,$0D            ; del balancín en eeprom interna
sts    $0167,temp
rcall  leer_memoria_eeprom_interna ; eeprom interna
sts    $65,temp
; *****
; para poner en estado activo al dispositivo esclavo (memoria eeprom)
; es requerido una transición de estado alto a bajo al momento de encender
; el instrumento
ldi    temp,$01            ; indica que se va a habilitar la memoria 2
sts    $0140,temp
rcall  habilita_EEPROM
cbi    portc,0             ; prende y apaga el display como indicación
rcall  delay                ; de encendido del instrumento
sbi    portc,0
; *****
configuracion_LCD:
; configuración de control del LCD después de salir del estado de bajo consumo
sbi    portb,0             ; habilita regulador VdcLCD
; Configuración de DDRA para líneas de control de LCD y canales del ADC
; DDRA 7 6 5 4 3 2 1 0
ldi    temp,0b11110000    ; DB7 DB6 DB5 DB4 0 0 ADC1 ADC0
out    DDRA,temp ; Líneas del display, sensor de temperatura y nivel de batería
; Configuración de DDRC y PORTC para líneas de datos de LCD y Led_Evento
; DDRC: 7 6 5 4 3 2 1 0

```

```

; TOSC2 TOSC1 E RW RS OFFLCD OFFTEMP LED
ldi temp,0b00111001 ; 0 0 0 0 0 0 0 0
out DDRC,temp ; 0 0 1 1 1 1 1 1
sbi portc,0 ;pinc0 como salida para Led_Evento
sbi portc,2 ;pinc2 como entrada de señal INVALID
; inicialización del display LCD
; (1)function set (wait more than 15ms)
rcall delay15msec
rcall function_set
; (2) function set (wait more than 4.1ms)
rcall delay5msec
rcall function_set
; (3) function set (wait more than 100ms)
rcall delay100msec
rcall function_set
; (4) function set (Set interface data lenght. Set 4-bit)
rcall delay5msec
rcall function_set1
;function set
rcall delay5msec
rcall function_set1
rcall delay5msec
rcall function_setNF
;display ON
rcall display_ON_OFF
;Display clear
rcall display_clear
;Entry Mode Set
rcall Entry_Mode
; conversión de datos de temperatura externa e interna y almacenamiento en EEPROM
; SPI
; Conversión de Temperatura Externa
rcall conversion_ext ;se llama a subrutina de conversión de temperatura
; externa
lds temp,$D4 ;almacena en SRAM primer dígito de valor de temperatura
subi temp,$30 ;ASCII - $30
mov fBCDH,temp ;carga en registro primer dígito BCD para conversión
lds temp,$D5 ;almacena en SRAM segundo dígito de valor de temperatura
subi temp,$30 ;ASCII - $30
mov fBCDL,temp ;almacena segundo dígito BCD para conversión
rcall BCD2bin8 ;llama a subrutina para conversión BCD2BIN8
mov r2,tbin ;el resultado lo almacena en registro r2
; envía dato correspondiente a sensor de temperatura externa en memoria EEPROM SPI
rcall WRITE_EEPROM ;se llama a subrutina de escritura en memoria
; EEPROM SPI
salir:
sbi portc,0 ;se apaga el led de registro de eventos
pop r31
pop r30
pop r25
pop r24
pop r23
pop r22

```

```

    pop    r16
    pop    r6
    pop    r1
    out    sreg,r0
    reti

;*****
;-----Subrutina de conversion HORA _ MINUTOS-----
;*****
mpy8u:
    clr    m8uH        ;clear result High byte
    ldi    mcnt8u,8    ;init loop counter
    lsr    mp8u        ;rotate multiplier

m8u_1:
    brcc   m8u_2       ;carry set
    add    m8uH,mc8u   ;add multiplicand to result High byte
m8u_2:
    ror    m8uH        ;rotate right result High byte
    ror    m8uL        ;rotate right result L byte and multiplier
    dec    mcnt8u     ;decrement loop counter
    brne   m8u_1      ;if not done, loop more
    ret

;*****
mpy16u:
    clr    m16u3       ;clear 2 highest bytes of result
    clr    m16u2
    ldi    mcnt16u,16  ;init loop counter
    lsr    mp16uH
    ror    mp16uL

m16u_1:
    brcc   noad8       ;if bit 0 of multiplier set
    add    m16u2,mc16uL ;add multiplicand Low to byte 2 of res
    adc    m16u3,mc16uH ;add multiplicand high to byte 3 of res
noad8:
    ror    m16u3       ;shift right result byte 3
    ror    m16u2       ;rotate right result byte 2
    ror    m16u1       ;rotate result byte 1 and multiplier High
    ror    m16u0       ;rotate result byte 0 and multiplier Low
    dec    mcnt16u     ;decrement loop counter
    brne   m16u_1     ;if not done, loop more
    ret

;*****
conversion_HHMMSS:
;SS (segundos)
    lds    dd16uL,$013A ;carga Lbyte de tiempo en segundos
    lds    dd16uH,$013B ;carga Hbyte de tiempo en segundos
    ldi    dv16uL,60    ;valor del divisor = 60
    clr    dv16uH       ;valor del divisor = 60
    rcall  div16u
    sts    $013B,drem16uL ;valor de SS
    sts    $013C,dres16uL ;valor del residuo de tiempo en segundos
    sts    $013D,dres16uH ;valor del residuo de tiempo en segundos

```

```

;MM (minutos)
    lds    dd16uL,$013C    ;carga Lbyte de residuo de tiempo en segundos
    lds    dd16uH,$013D    ;carga Hbyte de residuo de tiempo en segundos
    ldi    dv16uL,60      ;valor del divisor = 60
    clr    dv16uH          ;valor del divisor = 60
    rcall  div16u
    sts    $013E,drem16uL  ;valor de MM
    sts    $013F,dres16uL  ;valor de HH
    ret

;*****
;-----SUBROUTINAS DE ACTUALIZACION DE DATOS DE TEMPERATURA-----
;*****
conversion_ext:
;inicio de conversión de dato dummy para inicializar el ADC
;Configuración del convertido analogico digital ADMUX
    ldi    temp,0b00000001    ;se selecciona canal 1 para conversión
    out    ADMUX,temp
;Configuración de registro de control y de estado ADCSRA
    cbi    ADCSRA,ADFR
    sbi    ADCSRA,ADSC        ;inicia conversión ADSC = 1
fconv0:
    sbis   ADCSRA,ADIF
    rjmp  fconv0
    sbi    ADCSRA,ADIF        ;se limpia bandera de interrupción de fin de
                                ;conversión
    in     fbinL,ADCL        ;se almacena dato de conversión en registros
    in     fbinH,ADCH        ;para conversión a codigo ASCII
    rcall  bin2BCD16
    rcall  ASCII_precipitacion
    lds    temp,$011E
    sts    $D4,temp
    lds    temp,$011F
    sts    $D5,temp
    ret

;-----SUBROUTINAS DE RETRASO-----
;*****

delay:  ldi    r18,$ff
again:  ldi    r17,$ff
down:   dec    r17
        brne  down
        dec   r18
        brne  again
        ret

;-----Escribir a memoria EEPROM interna parámetros de operación-----
;*****
escribir_memoria_eeprom_interna:
EEPROM_write:
    sbic   EECR,EEWE        ; Wait for completion of previous write
    rjmp  EEPROM_write
;Dirección a la que se desea escribir
    lds    EEhigh,$0166    ;se indica dirección alta de EEPROM
    out    EEARH,EEhigh    ;a escribir

```

```

        lds    EElow,$0167      ;se indica dirección baja de EEPROM
        out    EEARL,EElow     ;a escribir
; Dato que se desea almacenar
        lds    temp,$0168
        out    EEDR,temp
; inicia proceso de escritura
        sbi    EECR,EEMWE
        sbi    E+ECR,EEWE
        ret    +
; *****
;-----Leer memoria EEPROM interna parámetros de operación-----
; *****
leer_memoria_eeprom_interna:
EEPROM_read:
        sbic   EECR,EEWE
        rjmp  EEPROM_read
; Dirección que desea leerse
        lds    EEhigh,$0166    ;se indica dirección alta de EEPROM
        out    EEARH,EEhigh   ;a escribir
        lds    EElow,$0167    ;se indica dirección baja de EEPROM
        out    EEARL,EElow    ;a escribir
; Inicia proceso de lectura
        sbi    EECR,EERE
; Se lee dato en registro
        in     temp,EEDR
        ret
; *****
escribir_memoria_interna_fecha:
; guarda dato de fecha (día, mes y año) en memoria eeprom interna
; se guardan los datos en las localidades $000B, $000C y $000D respectivamente
; se escribe dato de día
        clr    temp           ;almacena en localidades de SRAM
        sts    $0166,temp     ;dirección en donde se escribirá la capacidad
        ldi    temp,$0B      ;del balancín en eeprom interna
        sts    $0167,temp
        lds    r2,$63
        sts    $0168,r2      ;almacena en localidad de SRAM dato a guardar en
        rcall  escribir_memoria_eeprom_interna ;eeprom interna
; se escribe dato de mes
        clr    temp           ;almacena en localidades de SRAM
        sts    $0166,temp     ;dirección en donde se escribirá la capacidad
        ldi    temp,$0C      ;del balancín en eeprom interna
        sts    $0167,temp
        lds    r2,$64
        sts    $0168,r2      ;almacena en localidad de SRAM dato a guardar en
        rcall  escribir_memoria_eeprom_interna ;eeprom interna
; se escribe dato de año
        clr    temp           ;almacena en localidades de SRAM
        sts    $0166,temp     ;dirección en donde se escribirá la capacidad
        ldi    temp,$0D      ;del balancín en eeprom interna
        sts    $0167,temp
        lds    r2,$65
        sts    $0168,r2      ;almacena en localidad de SRAM dato a guardar en

```

```

        rcall  escribir_memoria_eeprom_interna      ;eeprom interna
        ret
;*****
;*****
;*****

```

Lenguaje Visual Basic. Este programa contiene las instrucciones necesarias que definen la interfaz de comunicación entre el termopluiógrafo-computadora-usuario, con el objetivo de tener acceso a los datos contenidos en la memoria del termopluiógrafo, para de esta manera hacer evaluaciones o procesamiento de la información de acuerdo con las necesidades del usuario.

```

'Programa: TERMOPLUIÓGRAFO DIGITAL
'Código fuente de la interfaz gráfica del Termopluiógrafo digital
-----

```

```

Option Explicit
Private nCid As Integer 'identificación del puerto
Private BufferSal As String 'buffer de salida
Const Tb = 16384 'tamaño del buffer de E y de S
-----

```

```

Private Sub Abrir_Com(DefPuertoCom As String)
'Si el puerto está abierto, cerrarlo
If MSCommX.PortOpen Then MSCommX.PortOpen = False
'Establecer el número del puerto
nCid = Val(Mid(DefPuertoCom, InStr(DefPuertoCom, ":") - 1, 1))
MSCommX.CommPort = nCid
'Establecer el tamaño del buffer de recepción y de transmisión
MSCommX.InBufferSize = 21845
MSCommX.OutBufferSize = 21845

'Abrir el puerto de comunicaciones. Si el número
'del puerto no es válido, se genera el error 68.
On Error Resume Next
MSCommX.PortOpen = True
If Err Then
    MsgError.Show
    MsgError.Label1.Caption = "NO SE PUEDE ABRIR EL PUERTO COM" & nCid
    Exit Sub
End If
'Establecer las características de comunicación
MSCommX.Settings = Mid(DefPuertoCom, InStr(DefPuertoCom, ":") + 1)
End Sub
-----

```

```

Private Sub Comunicacion_Click()
ConfigCom.Show
FormComm.Visible = False
End Sub
-----

```

```

Private Sub Comunicacion_GotFocus()
FormComm.Caption = "Configurar puerto de comunicaciones"
End Sub

```

```

-----
Private Sub Comunicacion_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
FormComm.Caption = "Configurar puerto de comunicaciones"
End Sub
-----

```

```

Private Sub Form_Load()
'Características por defecto del puerto
'DefCom = "COM1:9600,N,8,1"
Animation1.Stop
Animation1.Close
Animation1.Open App.Path & "\i_ani.avi"
Animation1.Play
End Sub
-----

```

```

Private Sub Iniciar_Click()
Dim msg As String, datos_puerto As String, capacidad As String
Dim cadena_caracteres As Integer
'Abrir_Com DefCom
cadena_caracteres = 0
If MSCommX.PortOpen Then
    alfa = FormComm.MSCommX.InBufferCount
    contador_datos = 0
    MSCommX.Output = "L"
    clock = 0
    Do
        clock = clock + 1
    Loop Until clock = Timer1.Interval
    Do
        datos_puerto = FormComm.MSCommX.Input
        contador_datos = contador_datos + 1
    Loop While contador_datos = alfa
    cadena_caracteres = Len(datos_puerto)
    If cadena_caracteres <= 44 Then
        MsgErrorSNE.Show
        FormComm.Visible = True
    ElseIf datos_puerto = "" Then
        MsgErrorSNE.Show
        FormComm.Visible = True
    Else
        datos.leer.Enabled = True
        datos.Show
        FormComm.Visible = False
    End If
End If
If Err Then
    FormComm.Visible = True
End If
End Sub
-----

```

```

Private Sub Iniciar_GotFocus()
FormComm.Caption = "Iniciar comunicación con el Termoplúviógrafo digital"
End Sub

```

```

-----
Private Sub Iniciar_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
FormComm.Caption = "Iniciar comunicación con el Termoplviógrafo digital"
End Sub

```

```

-----
Private Sub Salir_Click()
  If FormComm.MSCommX.PortOpen = True Then
    FormComm.MSCommX.PortOpen = False
  End If
End
End Sub

```

```

-----
Private Sub Salir_GotFocus()
FormComm.Caption = "Salir del programa"
End Sub

```

```

-----
Private Sub Salir_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
FormComm.Caption = "Salir del programa"
End Sub

```

```

*****
*****

```

```

Option Explicit
Private Puerto As String
Private Baudios As String, Paridad As String
Private BitsPorCarácter As Integer, BitsDeParada As Integer

```

```

-----
Private Sub Aceptar_Click()
  'El contenido de DefCom es de la forma "COMn:b,p,bc,bs"
  'DefCom = Puerto & ":" & 9600 & "," & "N" & "," & _
  '      8 & "," & 1
  On Error GoTo RUTINA_Error
  FormComm.MSCommX.PortOpen = 1 'True
  FormComm.Visible = True
  Unload ConfigCom
Exit Sub
RUTINA_Error:
MsgBox Err.Description, vbExclamation, "ERROR DE COMUNICACIÓN"
End Sub

```

```

-----
Private Sub Cancelar_Click()
  Unload ConfigCom
  FormComm.Visible = True
End Sub

```

```

-----
Private Sub Form_Load()
Dim X As String, DefPto As String
'Mostrar valores actuales de las comunicaciones.
'Puerto
'DefPto = DefCom 'de la forma: "COMn:b,p,bc,bs"
'X = Mid(DefPto, InStr(DefPto, ":") - 1, 1)
' ConfigCom.puertocom(Val(X)).Value = True

```

End Sub

```
Private Sub Option1_Click()
    FormComm.MSCommX.CommPort = 1
End Sub
```

```
Private Sub Option2_Click()
    FormComm.MSCommX.CommPort = 2
End Sub
```

```
Private Sub Option3_Click()
    FormComm.MSCommX.CommPort = 3
End Sub
```

```
Private Sub PuertoCom_Click()
    FormComm.MSCommX.CommPort = 1
End Sub
```

```
Private Sub Option4_Click()
    FormComm.MSCommX.CommPort = 4
End Sub
```

```
Private Sub Option5_Click()
    FormComm.MSCommX.CommPort = 5
End Sub
```

```
Private Sub Option6_Click()
    FormComm.MSCommX.CommPort = 6
End Sub
```

```
*****
*****
```

Option Explicit

```
Private Sub Actualizar_Click()
    Dim datos_puerto As String, capacidad As String
    Dim fecha_termoD As String, fecha_termoM As String, fecha_termoA As String
    Dim fecha_termo As String
    Dim modo_memoria As String
    Dim msg As String
    Dim numero_caracteres As Integer
    Dim hora_h As String, hora_m As String, hora_s As String, hora_Md As String
    Dim marca_am_pm As String
    Dim estado_bateria As String, bateria As String
    ActParametros.Caption = "Parámetros de operación"
    Dim Contador As Integer
    Contador = 0
    verificar_comunicacion:
    Contador = Contador + 1
    'If FormComm.MSCommX.PortOpen = False Then
    '    Abrir_Com DefCom
    'End If
    'se envia caracter para obtener los parametros bajo los cuales
    'está operando obteniendo así el valor del balancín
```

```

'la fecha de operación del sistema
'y el modo de almacenamiento de datos en memoria
FormComm.MSCommX.InBufferCount = 0
If FormComm.MSCommX.PortOpen = False Then
    FormComm.Show
Else
    clock = 0
    Do
        Timer1.Enabled = True
        clock = clock + 1
    Loop Until clock = Timer1.Interval
'Enviar caracter al buffer para obtener los datos desde la memoria
FormComm.MSCommX.Output = "L"
clock = 0
Do
    Timer1.Enabled = True
    clock = clock + 1
    alfa = FormComm.MSCommX.InBufferCount
    Counter = alfa
    Loop Until clock = Timer1.Interval
    contador_datos = 0
    Do
        datos_puerto = FormComm.MSCommX.Input
        contador_datos = contador_datos + 1
    Loop While contador_datos = alfa
End If
If datos_puerto = "" Then
    MsgErrorSNE.Show
    'msg = "SISTEMA NO ENCONTRADO" & vbCrLf
    'msg = msg & "Verifique que el sistema este conectado correctamente," & vbCrLf
    'MsgBox msg, 48, "Comunicaciones"
    datos.Hide
Else
    'MUESTRA DATO CORRESPONDIENTE A CAPACIDAD DEL BALANCIN
    capacidad = datos_puerto
    capacidad = Mid(capacidad, InStr(capacidad, ".") - 1, 3)
    ActParametros.Text5.Text = capacidad
    'MUESTRA DATO CORRESPONDIENTE A FECHA QUE TIENE EL TERMOPLUVIOGRAFO
    fecha_termoD = datos_puerto
    fecha_termoD = Mid(fecha_termoD, InStr(fecha_termoD, ".") + 8, 2)
    fecha_termoM = datos_puerto
    fecha_termoM = Mid(fecha_termoM, InStr(fecha_termoM, ".") + 10, 2)
    fecha_termoA = datos_puerto
    fecha_termoA = Mid(fecha_termoA, InStr(fecha_termoA, ".") + 12, 2)
    fecha_termo = fecha_termoD & "/" & fecha_termoM & "/" & fecha_termoA
    ActParametros.Text2.Text = fecha_termo
    'MUESTRA DATO CORRESPONDIENTE A LA HORA QUE TIENE EL
    TERMOPLUVIOGRAFO
    hora_h = datos_puerto
    hora_h = Mid(hora_h, InStr(hora_h, ".") + 14, 2)
    hora_m = datos_puerto
    hora_m = Mid(hora_m, InStr(hora_m, ".") + 16, 2)
    hora_s = datos_puerto

```

```

hora_s = Mid(hora_s, InStr(hora_s, ".") + 18, 2)
hora_Md = datos_puerto
hora_Md = Mid(hora_Md, InStr(hora_Md, ".") + 20, 1)
If hora_Md = "p" Then
marca_am_pm = "pm"
Else
marca_am_pm = "am"
End If
ActParametros.Text3.Text = hora_h & ":" & hora_m & ":" & hora_s & " " &
marca_am_pm
'MUESTRA ESTADO DE LA BATERIA
estado_bateria = datos_puerto
estado_bateria = Mid(estado_bateria, InStr(estado_bateria, ".") + 21, 1)
If estado_bateria = "k" Then
bateria = "O.K."
ActParametros.Image1.Visible = False
ElseIf estado_bateria = "b" Then
bateria = "Cambiar bateria"
ActParametros.Image1.Visible = True
Else
bateria = "O.K."
ActParametros.Image1.Visible = False
End If
ActParametros.Text4.Text = bateria
'MUESTRA DATO CORRESPONDIENTE A MODO DE ALMACENAMIENTO DE DATOS EN
'MEMORIA
modo_memoria = datos_puerto
modo_memoria = Mid(modo_memoria, InStr(modo_memoria, ".") + 22, 1)
If modo_memoria = "D" Then
ActParametros.Text1.Text = "Detener escritura"
Else
ActParametros.Text1.Text = "Sobreescribir datos"
End If
datos.Visible = False
ActParametros.Show
End If
If Err Then
ActParametros.Visible = False
End If
'datos.Visible = False
'ActParametros.Show
End Sub
-----
Private Sub Actualizar_GotFocus()
datos.Caption = "Parámetros de operación"
Animation1.Visible = False
Animation1.Stop
Animation1.Close
Animation2.Stop
Animation2.Close
Animation2.Visible = True
Animation2.Open App.Path & "\\comunicacion_pc_parametros.avi"
Animation2.Play

```

End Sub

```
Private Sub Actualizar_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
datos.Caption = "Parámetros de operación"
Animation1.Visible = False
Animation1.Stop
Animation1.Close
Animation2.Stop
Animation2.Close
Animation2.Visible = True
Animation2.Open App.Path & "\\comunicacion_pc_parametros.avi"
Animation2.Play
End Sub
```

```
Private Sub Cancelar_Click()
    If FormComm.MSCommX.PortOpen = True Then
        FormComm.MSCommX.PortOpen = False
    End If
End
End Sub
```

```
Private Sub Cancelar_GotFocus()
datos.Caption = "Salir del programa"
End Sub
```

```
Private Sub Cancelar_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
datos.Caption = "Salir del programa"
End Sub
```

```
Private Sub Form_Load()
Dim Contador As Integer
Contador = 0
Animation1.Visible = False
Animation2.Visible = False
End Sub
```

```
Private Sub Leer_Click()
Dim msg As String, datos_puerto As String
Dim Contador As Integer
Contador = 0
datos.Visible = False
animacion.Visible = True
verificar_comunicacion:
Contador = Contador + 1
FormComm.MSCommX.InBufferCount = 0
If FormComm.MSCommX.PortOpen = False Then
    FormComm.MSCommX.PortOpen = True
Else
    clock = 0
    Do
        clock = clock + 1
```

```

    Loop Until clock = 100000 'Timer1.Interval
'Enviar caracter al buffer para obtener los datos desde la memoria
    FormComm.MSCommX.Output = "L"
'habilita timer para permitir al microcontrolador enviar todos los
'caracteres disponibles
    'Timer1.Interval = 10000
    clock = 0
    Do
        clock = clock + 1
    Loop Until clock = 100000 'Timer1.Interval
    FormComm.MSCommX.InputLen = 0
    'Timer1.Interval = 10000
    clock = 0
    Do
        clock = clock + 1
        alfa = FormComm.MSCommX.InBufferCount
    Loop Until clock = 100000 'Timer1.Interval
    Counter = alfa
    'Timer1.Interval = 10000
    clock = 0
    Do
        clock = clock + 1
    Loop Until clock = 100000 'Timer1.Interval
    contador_datos = 0
    Do
        contador_datos = contador_datos + 1
        datos_puerto = FormComm.MSCommX.Input
    Loop While contador_datos = alfa
    contador_datos = Len(datos_puerto)
End If
If datos_puerto = "" Then
    If Contador > 10 Then
        MsgErrorSNE.Show
        datos.Hide
        FormComm.Show
    Else
        GoTo verificar_comunicacion
    End If
Else
    datos_evento = datos_puerto
End If
End Sub

```

```

Private Sub leer_GotFocus()
datos.Caption = "Adquisición de datos"
Animation2.Visible = False
Animation2.Stop
Animation2.Close
Animation1.Stop
Animation1.Close
Animation1.Visible = True
Animation1.Open App.Path & "\comunicacion_pc.avi"
Animation1.Play

```

End Sub

```
Private Sub Abrir_Com(DefPuertoCom As String)
Dim nCid As Integer
'Si el puerto está abierto, cerrarlo
If FormComm.MSCommX.PortOpen Then FormComm.MSCommX.PortOpen = False
'Establecer el número del puerto
nCid = Val(Mid(DefPuertoCom, InStr(DefPuertoCom, ":") - 1, 1))
FormComm.MSCommX.CommPort = nCid
'Establecer el tamaño de los buffer de recepción
'y de transmisión
FormComm.MSCommX.InBufferSize = 21845
FormComm.MSCommX.OutBufferSize = 21845
'Abrir el puerto de comunicaciones. Si el número
'del puerto no es válido, se genera el error 68.
On Error Resume Next
FormComm.MSCommX.PortOpen = True
If Err Then
    MsgError.Show
    MsgError.Label1.Caption = "NO SE PUEDE ABRIR EL PUERTO COM" & nCid
    Exit Sub
End If
'Establecer las características de comunicación
FormComm.MSCommX.Settings = Mid(DefPuertoCom, InStr(DefPuertoCom, ":") + 1)
End Sub
```

```
Private Sub leer_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
datos.Caption = "Adquisición de datos"
Animation2.Visible = False
Animation2.Stop
Animation2.Close
Animation1.Stop
Animation1.Close
Animation1.Visible = True
Animation1.Open App.Path & "\comunicacion_pc.avi"
Animation1.Play
End Sub
```


```
Option Explicit
Public contador_animacion As Integer
Public Contador As Integer
```

```
Private Sub Form_Load()
contador_animacion = 1
Image1.Visible = False
Image2.Visible = False
Image3.Visible = False
Image4.Visible = False
Image5.Visible = False
End Sub
```

```

Private Sub Timer1_Timer()
Contador = Contador + 1
Select Case contador_animacion
Case 1:
    Image1.Visible = True
    Image2.Visible = True
    Image5.Visible = True
    Image3.Visible = False
    Image4.Visible = False
    contador_animacion = 2
Case 2:
    Image1.Visible = True
    Image2.Visible = True
    Image4.Visible = True
    Image3.Visible = False
    Image5.Visible = False
    contador_animacion = 3
Case 3:
    Image1.Visible = True
    Image2.Visible = True
    Image3.Visible = True
    Image4.Visible = False
    Image5.Visible = False
    contador_animacion = 1
End Select
If Contador < 60 Then
    animacion.Visible = True
Else
    animacion.Visible = False
    Timer1.Enabled = False
    Download.Visible = True
    Unload animacion
End If
End Sub
*****
*****

```

APÉNDICE D

GLOSARIO DE TÉRMINOS

Este apéndice presenta un glosario de términos ordenados alfabéticamente, los cuales fueron utilizados en el transcurso del desarrollo de este proyecto.

- 4PPM (*Pulse Position Modulation*, Modulación por Posición de Pulso)
- ACL (*Asynchronous Connection Less*, Conexión Baja Asíncrona)
- ALU (*Arithmetic-Logic Unit*, Unidad Lógica Aritmética)
- ANSI (*American National Standards Institute*, Instituto de Estándares Nacionales Americanos)
- ASCII (*American Standard Code for Information Interchange*, Intercambio de Información para Códigos Estándares Americanos)
- AT (*Advanced Technology*, Tecnología Avanzada)
- AT (*Attention*, Atención)
- BOF (*Beginning Of Frame*, Trama de Inicio)
- CCITT (*International Telegraph and Telephone Consultative Committee*, Comité de Consulta Internacional de Telegrafía y Telefonía)
- CG (*Chasis Ground*, Tierra de Chasis)
- CISC (*Complex Instruction Set Computer*, Computadora con *Set* de Instrucciones Complejas)
- CPU (*Central Processing Unit*, Unidad de Procesamiento Central)
- CS (*Select Chip*, Selector de Circuito Integrado)
- CTS (*Clear To Send*, Limpio para Enviar)
- DCD (*Data Carrier Detect*, Detección de Portadora de Datos)
- DCE (*Data Circuit-terminating Equipment*, Equipo de Comunicación de Datos)
- DRAM (*Dynamic Random Access Memory*, Memoria de Acceso Aleatorio Dinámica)
- DSR (*Data Signal Rate Selector*, Selector de Velocidad de las Señales de los Datos)
- DSR (*Data Set Ready*, Listo para Configurar Datos)

DTE (*Data Terminal Equipment*, Equipo Terminal de Datos)
DTR (*Data Terminal Ready*, Terminal de Datos Lista)
EEPROM (*Electrically Erasable Programmable Read Only Memory*, Memoria de Sólo Lectura Programable Borrable Eléctricamente)
EIA (*Electronics Industries Association*, Asociación de Industrias Electrónicas)
EISA (*Extended Industry Standard Architecture*, Arquitectura Estándar de la Industria Extendida)
EOF (*End Of Frame*, Fin de Trama)
EPROM (*Erasable Programmable Read Only Memory*, Memoria de Sólo Lectura Programable Borrable)
ETC (*External Transmitter Clock*, Reloj del Transmisor Externo)
ETSI (*European Telecommunications Standards Institute*, Instituto de Estandarización de Telecomunicaciones Europeas)
FAX (*Fax Profile*, Perfil de Fax)
FHSS (*Frequency Hopping Spread Spectrum*, Espectro Amplio de Saltos de Frecuencia)
FIR (*Fast Infrared*, Infrarrojo Rápido)
FTP (*File Transfer Profile*, Perfil de Transferencia de Archivos)
GAP (*Generic Access Profile*, Perfil de Acceso Genérico)
GFSK (*Gaussian Frequency Shift Keying*, Modulación por Desplazamiento de Frecuencia Gaussiana)
GOEP (*Generic Object Exchange Profile*, Perfil Genérico de Intercambio de Objetos)
GSM (*Global System for Mobile communications*, Sistema Global para Comunicaciones Móviles)
HCI (*Host Controller Interface*, Interfaz entre el Controlador y el Host)
HDLC (*High-level Data Link Control*, Control de Enlace de Datos de Alto Nivel)
IAS (*Information Access Service*, Servicio de Acceso a la Información)
IBM (*International Business Machines Corporation*, Corporación Internacional de Máquinas de Negocios)
IP (*Internet Protocol*, Protocolo de Internet)
IrCOMM (*Infrared Communications*, Comunicación por Infrarrojos)
IrDA (*Infrared Data Association*, Asociación de Datos Infrarrojos)
IrLAN (*Infrared Local Area Network*, Redes de Área Local por Infrarrojos)
IrLAP (*Infrared Link Access Protocol*, Protocolo de Acceso al Enlace por Infrarrojos)
IrLMP (*Infrared Link Management Protocol*, Protocolo de Administración de Enlace por Infrarrojos)
IrMC (*Infrared Mobile Communications*, Comunicaciones Móviles por Infrarrojos)
IrOBEX (*Infrared Object Exchange*, Intercambio de Objetos por Infrarrojos)
ISA (*Industry Standard Architecture*, Arquitectura Estándar de la Industria)
ISM (*Industrial Scientific and Medical Bands*, Banda Medica Científica e Industrial)
ISO (*International Standards Organization*, Organización Internacional de Estándares)
L2CAP (*Logical Link Control and Adaptation Protocol*, Protocolo de Adaptación y Control de Enlace Lógico)
LAN (*Local Area Network*, Red de Área Local)
LMP (*Link Management Protocol*, Protocolo del Administrador de Enlace)
LL (*Local Loopback*, Retroalimentación Local)
MCA (*Micro Channel Architecture*, Arquitectura de Microcanal)
MIR (*Medium Infrared*, Infrarrojo Medio)
NRZI (*Non Return to Zero Inverted*, No Retorno a Cero Invertido)
NRZ-L (*Non Return to Zero Level*, No Retorno a Nivel Cero)
OBEX (*Object Exchange*, Intercambio de Objetos)
OSI (*Open Systems Interconnection*, Interconexión de Sistemas Abiertos)
PC (*Personal Computer*, Computadora Personal)
PC (*Program Counter*, Contador de Programa)

PCI (*Peripheral Component Interconnect*, Interconexión de Componentes Periféricos)
PPP (*Point to Point Protocol*, Protocolo Punto a Punto)
PROM (*Programmable Read Only Memory*, Memoria de Sólo Lectura Programable)
RAM (*Random Access Memory*, Memoria de Acceso Aleatorio)
RFCOMM (*Radio Frequency Communication*, Comunicación por Radiofrecuencia)
RI (*Ring Indicator*, Indicador de Campana)
RISC (*Reduced Instruction Set Computer*, Computadora con *Set* de Instrucciones Reducidas)
RL (*Remote Loopback*, Retroalimentación Remota)
ROM (*Read Only Memory*, Memoria de Sólo Lectura)
RTS (*Ready To Send*, Listo para Enviar)
RxC (*Receive Clock*, Reloj de Recepción)
RxD (*Received Data*, Receptor de Datos)
RZI (*Return to Zero Inverted*, Regreso a Cero Invertido)
SCDC (*Secondary Received Line Signal Detector*, Detector de Señal de Línea Receptora Secundario)
SCK (*Serial Clock*, Reloj Serial)
SCO (*Synchronous Connection Oriented*, Conexión Orientada Síncrona)
SCTS (*Secondary Clear To Send*, Limpio para Enviar Secundario)
SDAP (*Service Discovery Application Profile*, Perfil de Aplicación y Descubrimiento de Servicios)
SDLC (*Synchronous Data Link Control*, Control de Enlace de Datos Síncronos)
SDP (*Service Discovery Protocol*, Protocolo de Descubrimientos de Servicios)
SE0 (*Ending Single 0*, Finalizado Simple 0)
SG (*Signal Ground*, Señal de Tierra)
SI (*Serial Input*, Entrada Serial)
SIG (*Special Interest Group*, Grupo de Interés Especial)
SIR (*Serial Infrared*, Infrarrojo Bajo)
SISC (*Specific Instruction Set Computer*, Computadora con *Set* de Instrucciones Específicas)
SO (*Serial Output*, Salida Serial)
SPI (*Serial Peripheral Interface*, Interfaz Periférico Serial)
SPP (*Serial Port Profile*, Perfil de Puerto Serie)
SRAM (*Static Random Access Memory*, Memoria de Acceso Aleatorio Estática)
SRTS (*Secondary Request To Send*, Petición para Enviar Secundario)
SRxD (*Secondary Received Data*, Receptor de Datos Secundario)
STxD (*Secondary Transmitted Data*, Transmisor de Datos Secundario)
SYNCH (*Synchronization Profile*, Perfil de Sincronización)
TCP/IP (*Transmission Control Protocol/Internet Protocol*, Protocolo de Control de Transmisión/Protocolo de Internet)
TDD (*Time Division Duplex*, División de Tiempo Duplex)
TIA (*Telecommunications Industry Association*, Asociación de Industrias de Telecomunicaciones)
TinyTP (*Tiny Transport Protocol*, Protocolo de Transporte Pequeño)
TM (*Test Mode*, Modo de Prueba)
TxC (*Transmit Clock*, Reloj de Transmisión)
TxD (*Transmitted Data*, Transmisor de Datos)
UART (*Universal Asynchronous Receiver Transmitter*, Transmisor Receptor Asíncrono Universal)
USART (*Universal Synchronous-Asynchronous Receiver Transmitter*, Transmisor Receptor Asíncrono-Síncrono Universal)
USB (*Bus Serial Universal*, Bus Serie Universal)
VFIR (*Very Fast Infrared*, Infrarrojo Muy Rápido)

WAN (*Wireless Access Network*, Acceso a Redes Inalámbricas)

WAP (*Wireless Access Point*, Protocolo de Acceso Inalámbrico)

WLAN (*Wireless Local Area Network*, Redes de Área Local Inalámbricas)

BIBLIOGRAFÍA

LIBROS

Bernal Díaz Arcelia. *Diseño y desarrollo de sistemas de comunicación de datos para estaciones remotas*. UNAM.

Brey Barry B. *Los Microprocesadores INTEL, arquitectura, programación e interfaces*. México, Prentice Hall, 1994.

Ceballos Francisco J. *Enciclopedia de Microsoft ® Visual Basic 6*. Alfaomega, 1997.

Collazo Javier. *Diccionario enciclopédico de términos técnicos*. Mc Graw-Hill, 1981.

Fernández F. *Tecnología móvil, aplicaciones GSM, GPRS, UMTS y WIFI*. Madrid, Anaya Multimedia, 2003.

Fishbane, Gasiorowicz, Thornton, S. *Física para Ciencias e Ingeniería Volumen II*, México, Prentice Hall, 1994.

Hall Douglas V. *Microprocessors and interfacing programming and hardware*. E.E.U.U., Mc Graw Hill, 1992.

Huidobro J. *Manual de Telefonía, Telefonía Fija y Móvil*. Paraninfo, España, 1998.

Infrared data transmission. ELEKTOR ELECTRONICS, April 1996.

Joe Campbell. *The RS-232 Solution*. Sybex, 1989.

Kaaranen, Ahtiainen, et al. *UMTS Networks –Architecture, Mobility and Services*. England, Wiley, 2001.

Kate G. *Edición Especial Visual C++® 6*. España, Prentice Hall, 1999.

Mandado Enrique. *Sistemas electrónico digitales*. 7ª edición, México, Alfaomega, 1991.

Miller Brian. *My favorite family of micros*. CIRCUIT CELLAR # 34, August 2001.

Nava, I. *Diseño y Construcción de un Adquisidor de Datos y un Codificador Angular para Medición de Nivel de Agua*. México, UNAM, 2004.

Paul J. Kline, Timothy J. Thompson, C. Bala Kumar. *Bluetooth Application Programming*. Morga Kaufmann Publishers, 2004.

Peter W. Gofton. *Mastering Serial Communication*. Sybex, 1986.

Predko Michael. *Programming and customizing PICmicro microcontrollers*. Second edition, USA, McGraw-Hill, 2002.

Ramachandran Hari. *IrDA Technology*. Part 1: An overview, CIRCUIT CELLAR #111, October 1999.

Ramachandran Hari. *IrDA Technology*. Part 2: Protocol Layers, CIRCUIT CELLAR #112, November 1999.

Sendín A. *Fundamentos de los sistemas de comunicaciones móviles*. España, McGraw-Hill, 2004.

Shapiro J. *Visual Basic .NET*. España, Mc Graw Hill, 2003.

Serway R. *Física Tomo II*. México, Mc Graw Hill, 1993.

Stallings W. *Comunicaciones y Redes de Computadores*. España, Pearson Prentice Hall, 2004.

Tocci Ronald J. *Microprocessors and microcomputers/Hardware and software*. Fifth edition, USA, Prentice Hall, 2000.

Walraven K. *Working with surface mount devices*. ELEKTOR ELECTRONICS, May 2000.

William Buchanan. *PC Interfacing, Communications and Windows Programming*. Addison-Wesley 1999.

MANUALES

AVR Risc Microcontroller. ATMEL Corporation, August 1999.

Guía de instrumentos y métodos de observación meteorológicos. OMM No.8, quinta edición en inglés, 1983.

Microcontroller Data Book. AT89 Series, ATMEL Corporation, December 1997.

Microcontroller Data Book, MICROCHIP, 2005.

National Analog and Interface Products Data Book, National Semiconductor, 2002.

Nonvolatile data memory, ATMEL Corporation, December 1998.

PÁGINAS DE INTERNET

<http://www.ag.com>
<http://www.analogdevice.com>
<http://www.atmel.com>
<http://www.bluetooth.com>
<http://www.cna.gob.mx>
<http://www.df.gob.mx>
<http://www.google.com>
<http://www.intel.com>
<http://www.irda.com>
<http://www.maxim.com>
<http://www.microchip.com>
<http://www.microsoft>
<http://www.monografias.com>
<http://www.motorola.com>
<http://www.national.com>
<http://www.rs-232.com>
<http://www.smn.gob.mx>
<http://www.ti.com>
<http://www.unam.mx>

<http://www.usb.org>
<http://www.vishay.com>
<http://www.wikipedia.com>
<http://www.yahoo.com>