



**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS**

**Perfil "Realidad Virtual"**

## **Trabajo de Diploma**

**Título:** Propuesta sobre como mejorar la transmisión de la información en el Simulador C10.

**Autores:**

- Nayla Yamila Sire Vila
- Arlin Maury Cruz Pérez

**Tutores:** Ing. Gadied Carrero Sotolongo

Ciudad de la Habana  
2007

# Agradecimientos

---

Autores: Nayla Yamila Sire Vila  
Arlin Maury Cruz Perez

## Resumen

En este trabajo que tiene como tema “ Propuesta sobre como mejorar la transmisión de la información en el Simulador C10”, ha surgido la necesidad de perfeccionar el funcionamiento de la transmisión de datos de este Simulador, debido a que la velocidad de transmisión de la información es lenta y puede que se pierdan datos en el momento de la transmisión. Para el desarrollo de esta tesis se tuvo como objetivo: Proponer una transmisión de la información de Simulador C10 que funcione de forma eficiente en el proyecto de Simpro, y para darle cumplimiento al objetivo propuesto se establecieron una serie de tareas que fueron llevadas a cabo durante el desarrollo de esta investigación. El resultado más relevante obtenido fue la obtención de una velocidad de transmisión de la información del Simulador C10 rápida a través de una comunicación por puerto USB, y la transferencia de información se lleva a cabo sobre los comandos anteriormente diseñados para la transmisión en el C10 adaptados al nuevo protocolo USB llamado cola de transferencia.

# Índice

<b>INTRODUCCIÓN.....</b>	<b>6</b>
<b>CAPÍTULO 1 : FUNDAMENTACIÓN TEÓRICA .....</b>	<b>10</b>
1.1 ESTADO DEL ARTE: .....	10
1.2 COMO FUNCIONA EL PROCESO DE LA COMUNICACIÓN: .....	11
1.3 PUERTO SERIE. ....	12
1.3.1 Modos de Transmisión de Datos.....	13
1.3.2 Tipos de comunicaciones en serie.....	14
1.3.3 Detectar errores en la comunicación.....	17
1.3.4 Parámetros que se emplean para establecer una comunicación por puerto serie: .....	18
1.3.5 Conectores de puerto serie .....	20
1.3.6 Características de las comunicaciones en serie: .....	21
1.3.7 Ubicación en el sistema informativo: .....	21
1.3.8 Interrupciones y direcciones .....	22
(Para controlar al puerto serie, la CPU emplea direcciones de puertos de E/S y líneas de interrupción (IRQ).).....	22
1.3.9 Transmisión de datos. ....	23
1.3.10 Señales .....	24
1.3.11 Chip controlador: La UART .....	25
<b>CAPÍTULO 2 DOCUMENTACIÓN SOBRE PUERTO SERIE RS232 Y PUERTO USB .....</b>	<b>26</b>
2.1 PUERTO SERIE RS-232 .....	26
2.1.1 Concepto del puerto Serie RS232 .....	27
2.1.2 Características generales del estándar RS-232 .....	27
2.1.3 Tipos de conectores que emplea RS232 .....	27
2.1.4 Conexiones entre pines del DTE y el DCE. Señales de entrada y salida:.....	29
2.1.5 Niveles de voltaje: .....	30
2.1.6 Características funcionales: .....	30
2.1.7 Tipo de conexiones entre el ordenador y el dispositivo por puerto serie RS232: .....	31
2.1.8 UART: Universal Asynchronous Receiver Transmitter .....	31
2.1.9 Transmisión en RS232 : .....	32
2.1.10 Control de Flujo en el RS-232 .....	34
2.1.11 Configuración del protocolo:.....	35
2.2 ESTÁNDAR USB  .....	36
2.2.1 Concepto del Bus Serie Universal (USB): .....	36
2.2.2 Objetivos de USB .....	37
2.2.3 Funcionamiento .....	37
2.2.4 Descripción del Sistema USB.....	39
2.2.5 Señales y conectores .....	43
2.2.6 Nivel de voltaje. ....	44
2.2.7 Topología del BUS y Protocolo USB para la transmisión de datos. ....	45
2.2.8 Características generales del USB .....	47
2.2.9 Explicación detallada de las versiones de USB .....	49
<b>CAPÍTULO 3 CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>52</b>
3.1 SIMULADOR C10.....	52
3.2 PROTOCOLO DE COMUNICACIÓN DE TRANSFERENCIA DE INFORMACIÓN DEL SIMULADOR C10(PTDC10).....	53
3.3 BLOQUES DE DATOS. COMANDOS QUE GENERAN AL PTDC10: .....	53

3.4 FORMATO DE LOS BLOQUES DE DATOS.....	54
3.5 OPERACIONES Y SEÑALES DE ENTRADA/SALIDA EN LA PARTE DEL SOFTWARE. CAPACIDAD DE ENVÍO Y RECEPCIÓN DE DATOS. ....	56
3.6 OBJETO DE ESTUDIO: TRANSFERENCIA DE INFORMACIÓN .....	57
3.7 DISPOSITIVOS DONDE SE ESTABLECE LA CONEXIÓN:.....	58
3.8 CONECTORES, CABLES Y PUERTO SERIE EMPLEADO PARA EL ESTABLECIMIENTO DE LA CONEXIÓN. ...	58
3.9 CONEXIÓN Y CONFIGURACIÓN PARA EL ESTABLECIMIENTO DE LA TRANSMISIÓN. ....	58
3.10 VELOCIDAD DE TRANSMISIÓN:.....	59
3.11 CONTROL DE FLUJO: .....	59
3.12 FLASHLITE186.....	60
3.13 PROBLEMÁTICAS QUE HAY ACTUALMENTE EN EL SIMULADOR .....	60
3.14 VARIANTES DE SOLUCIÓN DEL PROBLEMA:.....	61
3.15 OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN (UCI). ....	61
3.16 INFORMACIÓN QUE SE MANEJA. ....	61
3.17 COMPARACIÓN ENTRE RS-232 Y USB:.....	62
3.18 SE PUEDE MODIFICAR EL MÉTODO DE TRANSMISIÓN DE LA INFORMACIÓN DEL PTDC10 EMPLEANDO PUERTO USB? .....	63
3.19 BREVE DESCRIPCIÓN SOBRE LA PROPUESTA DE SISTEMA. ....	63
<b>CAPÍTULO 4 PROPUESTA DEL SISTEMA CON CONEXIÓN USB.....</b>	<b>64</b>
4.1 PUERTO SERIE PROPUESTO PARA LA CONEXIÓN. ....	64
4.2 CONECTORES Y CABLES A EMPLEAR PARA LA CONEXIÓN: .....	64
4.3 FLASHLITE CON PUERTO USB .....	65
4.4 COMPARACIÓN ENTRE FLASHLITE186 Y OMNIFLASH: .....	66
4.5 TRANSMISIÓN DE LA INFORMACION POR USB .....	67
4.6 HIDMAKER FS - USB THE EASY WAY : CÓDIGO FUENTE PARA LA TRANSMISIÓN POR USB .....	69
4.7 IDENTIFICACIÓN Y DIAGNÓSTICO .....	69
4.8 DESCRIPCIÓN GENERAL DE LA PROPUESTA DE SISTEMA. ....	71
<b>CONCLUSIONES .....</b>	<b>72</b>
<b>RECOMENDACIONES .....</b>	<b>73</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>74</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>82</b>

## Introducción

En Cuba se ha confeccionado un simulador de tiro lanzacohetes llamado "Simulador C10", que es un sistema que intenta replicar o simular la experiencia de tirar cohetes antiaéreos de la forma más precisa y realista posible. Este simulador permite entrenar al artillero en condiciones totalmente realistas, donde el practicante pierde noción de que se trata de una simulación, ya que se utilizan blancos con desplazamiento y aparición repentina en distintos lugares geográficos. Para ello, se emplean lanzacohetes reales adaptados a un simulador, respetando su configuración de fábrica. El Simulador C10 emplea una comunicación por puerto RS232 para el establecimiento de la transmisión.

### **Antecedentes:**

El protocolo RS232, protocolo que emplea el puerto serie RS232 existente en todos las PCs, es el más utilizado para la transmisión de datos entre ordenadores y periféricos. Muchos simuladores emplean este tipo de protocolo, como por ejemplo: "El sistema inalámbrico de control de un brazo ROBOT", el cual trata de un brazo mecánico robot programable conectado a un PC a través del puerto serie RS232 en el cual reside una aplicación servidor que permanece en escucha esperando "clientes" solicitando tomar el control del equipo. Otro simulador es el "Monitor de Pulsos Cardíacos con Interfaz a PC" que es un sistema que cuenta con una interfaz serie RS232 para poder tener la misma señal sobre un Computador Personal, en el cual con software de visualización y administración se puede implementar un sistema de monitoreo centralizado. Como se pudo ver la transmisión de la información por protocolo RS232 es muy empleada en la confección de muchos sistemas que necesitan comunicarse. Ahora, el empleo del puerto RS232 para el establecimiento de la comunicación para sistemas donde se transmitan grandes datos no es muy recomendado, debido que estos sistemas requieren de una mayor velocidad de transmisión, la cual por puerto RS232 no se consigue ya que la velocidad de transmisión de información que emplea es muy reducida.

### **Actualidad y necesidad del trabajo:**

Uno de los defectos de los puertos serie iniciales como el RS232 era su lentitud, sin embargo, con el paso del tiempo, están apareciendo multitud de puertos serie con una alta velocidad que los hace muy interesantes ya que tienen la ventaja de un menor cableado y solucionan el problema de la velocidad; son más baratos pues usan la técnica del par trenzado, por ello, el puerto RS-232 e incluso multitud de

puertos paralelos están siendo reemplazados por nuevos puertos serie como el USB. Un ejemplo de simuladores que emplean protocolo USB es: El Aerofly e G3 que es una interfaz de juegos, en la cual la instalación es muy fácil de usar, es compatible con las plataformas Windows 98, Xp, y 2000, es rápida la transferencia de información lo que proporciona que las acciones ejecutadas en el juego se realicen de forma rápida. Hay muchos simuladores actuales que emplean este tipo de puerto por las grandes ventajas que presenta, además de que muchos fabricantes de hoy en día solo emplean el puerto USB en los ordenadores. En el Simulador C10 con comunicación serial RS232 ha surgido la necesidad de proponer una transmisión de la información más eficiente, debido a que la velocidad de transferencia de datos que emplea es muy reducida, y puede ocurrir que se pierdan datos cuando se transmita información.

**Situación problemática:**

En Cuba ha surgido la necesidad de renovar la transmisión de información del Simulador C10 (lanzacohetes), ya que el mismo actualmente no responde a las necesidades del usuario final. El Simulador C10 presenta fallas en el sistema, debido a que la velocidad de transmisión es muy lenta y puede que exista pérdida de la información durante la transmisión. En la Universidad de las Ciencias Informáticas se ha estado trabajando en el proyecto de Realidad Virtual para en conjunto con la empresa SIMPRO lograr la incorporación de un entorno virtual y la interacción del hardware con estos elementos virtuales, y por la nuevas demandas que han surgido sobre este simulador se quiere obtener una propuesta para mejorar la transmisión de la información del Simulador C10, que actualmente funciona mediante un puerto serie RS-232.

**Problema científico**

¿Constituye la sustitución del puerto RS232 por el puerto USB las causas de la obtención de un método que garantice que la transmisión de la información que ejerce el PTDC10 se realice de forma eficiente en los laboratorios de realidad virtual?

**Objeto de estudio:**

Transferencia de información del Simulador C10 por puerto RS232.

**Objetivo del trabajo:**

Proponer una transmisión de la información del Simulador C10 que funcione de forma eficiente en los laboratorios de realidad virtual.

**Campo de acción:**

Proceso de obtención de una transmisión de información del Simulador C10 eficiente en los laboratorios de realidad virtual

**Tareas desarrolladas para darles cumplimiento al objetivo propuesto:**

- Estudio de las características y conceptos de los puertos series RS232 y USB.
- Estudio de las características y funcionamiento del protocolo que emplea el puerto serie RS232 y el USB para la transmisión.
- Estudio de las necesidades, características y funcionamiento del PTDC10.
- Estudio sobre la búsqueda de un método que garantice la eficiencia en la transmisión de la información del Simulador C10.
- Estudio referente a una Flashlite que emplee entrada por puerto USB.
- Estudio sobre como leer los datos que se implementan en el Software a través del puerto USB.

**Aportes prácticos esperados**

Se espera que con el desarrollo de esta investigación se obtenga una propuesta que garantice que la transmisión de la información del Simulador C10 funcione eficientemente, y que los usuarios finales puedan usar un sistema que funcione de forma rápida, y que responda a todas las acciones solicitadas o datos entrados por el usuario final sin error alguno.

### **Estructuración del contenido con una breve explicación de sus partes**

El trabajo esta dividido en tres capítulos, que contienen la información sobre la investigación realizada, los cuales están organizados de la siguiente forma:

#### **Capitulo1: Fundamentación teórica**

En este capitulo se realiza un estudio sobre el estado del arte del tema tratado, a nivel internacional, nacional y de la Universidad, de las tecnologías, metodologías y software usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta, sobre los que es necesario profundizar. También se realiza un estudio sobre los protocolos de puertos serie mas utilizados, y de los puertos serie que se emplean para el establecimiento de la comunicaron entre la computadora y los periféricos.

#### **Capitulo2: Documentación de puerto serie RS232 y el puerto serie USB**

#### **Capitulo3: Características del sistema**

#### **Capitulo4: Propuesta del Sistema con conexión USB.**

## Capítulo 1 : Fundamentación Teórica

### Introducción

En este capítulo se aborda todo lo referente al estado actual sobre el empleo y confección de protocolos de transmisión de datos de los puertos serie en el ámbito mundial, nacional y en la universidad, además de un estudio de las tecnologías y metodologías en las que se apoya para la solución del problema planteado.

Durante todos estos años han sido objeto de estudio muchos software que necesitan ser modificados, debido a que dichos sistemas anteriormente fueron creados con escasa tecnología para ser empleados por sectores de la sociedad y que actualmente no responden a las necesidades de esta. La Universidad de las Ciencias Informáticas ha hecho realidad una revolución en la renovación de la transmisión de datos de simuladores que presentan problemas, para ser modificados y empleados por la sociedad en óptimas condiciones y que satisfaga las necesidades del usuario final.

### 1.1 Estado del arte:

Los protocolos de transmisión de datos de puertos seriales han adquirido gran importancia en la sociedad, ofreciendo así, la posibilidad de comunicarse y obtener información necesaria en un reducido período de tiempo. En todos estos años se han creado muchos protocolos sobre la base de puerto serie que ayudan a la transferencia la información de una PC a un periférico. A continuación se mostrará un resumen de la incidencia de los protocolos de puerto serie en el ámbito nacional, internacional y de la Universidad:

#### Ámbito Internacional:

A mediados de los 90 se hizo casi imprescindible disponer de una nueva interfaz estándar para la conexión de dispositivos externos al PC que pudiera conectar diferentes periféricos como impresoras, escáneres, ratones o cámaras digitales de una forma rápida, sencilla y eficaz que estuviera basada en comunicaciones serie. La solución se llamó USB que con el tiempo su utilidad

ha aumentado a un ritmo vertiginoso. En 1995, siete fabricantes (Compaq, Digital Equipment, IBM, Intel, Microsoft, NEC y Northern Telecom.) definieron la especificación USB para unas funciones bastante más limitadas que las que tiene actualmente. Tres años más tarde nació USB 1.1 y en 1999 una segunda versión que se llamó USB 2.0. El estándar más empleado a nivel mundial para la conexión entre la PC y periféricos es el RS232, pero con el apareamiento del USB fue reemplazado, ya que por medio de una comunicación por USB se logra una transmisión de la información de forma rápida, además de que presenta múltiples ventajas. Existen muchos simuladores que emplean protocolos por puerto serie para la transmisión de la información, los más empleados a nivel mundial son el RS232, y actualmente la comunicación por USB..

### **Ámbito Nacional:**

En Cuba se han confeccionado simuladores que emplean protocolos de transmisión de datos de puertos serie. Como es el caso del Simulador C10 con comunicación por puerto RS232 y un protocolo RS232 para la transmisión de datos definido por este puerto y diseñado estructuralmente hace algunos años por un especialista. Actualmente se está viendo la posibilidad de proponer una transmisión de datos que funcione de forma eficiente en este simulador.

### **En la Universidad:**

En la Universidad de las Ciencias Informáticas se ha llevado a cabo una revolución para el perfeccionamiento de simuladores que antes fueron coincidos sin la tecnología adecuada. Uno de estos simuladores llamado "Simulador C10" funciona mediante un protocolo de transmisión por puerto serie RS232. Por las necesidades que presenta este simulador se está estudiando el establecimiento de una propuesta para mejorar la transmisión que ejerce el protocolo de transmisión de información del Simulador C10 (PTDC10).

## **1.2 Como funciona el proceso de la comunicación:**

El objetivo principal de las comunicaciones es intercambiar información entre dos entidades.

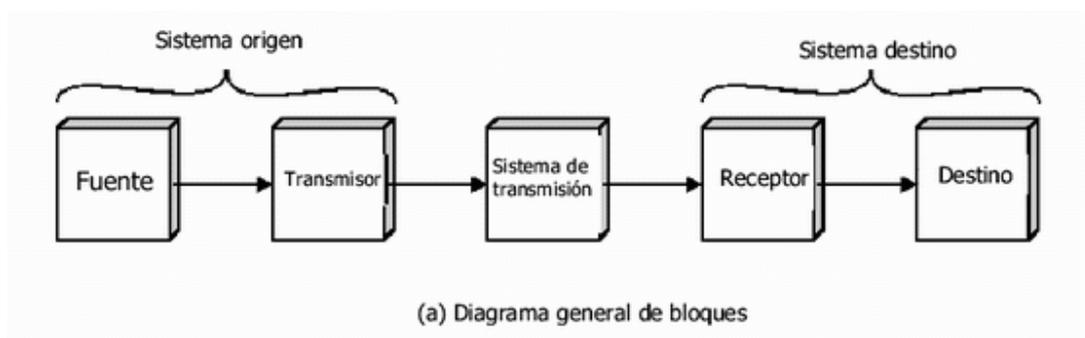
La fuente: este es el dispositivo que genera los datos a transmitir.

El transmisor: normalmente los datos generados no son transmitidos así como son generados. El transmisor transforma y codifica la información produciendo señales electromagnéticas para ser transmitidas a través de algún sistema de transmisión.

El sistema de transmisión: que puede ser un cable desde una simple línea de transmisión hasta una compleja red que conecte la fuente con el destino.

El receptor: que acepta la señal proveniente del sistema de transmisión y la convierte de tal manera que pueda ser manejada por el dispositivo destino.

El destino: que toma los datos del receptor.



### 1.3 Puerto serie.

#### Concepto de puerto Serie

El puerto serie de un ordenador es un adaptador asíncrono utilizado para poder intercomunicar varios ordenadores entre sí. Un puerto serie recibe y envía información fuera del ordenador mediante un determinado software de comunicación o un driver del puerto serie. El software envía la información al puerto carácter a carácter, convirtiéndolo en una señal que puede ser enviada por un cable serie o un módem. Cuando se ha recibido un carácter, el puerto serie envía una señal por medio de una interrupción indicando que el carácter está listo. Cuando el ordenador ve la señal, los servicios del puerto serie leen el carácter.

El puerto serie es un elemento hardware que permite el flujo de información en una sola línea de comunicación. Es una conexión por medio de la cual se envían datos a través de un solo conducto. Las computadoras tienen dos puertos seriales: COM1 y COM2.

**Serial:** método para transmitir datos secuencialmente, es decir, bit por bit. Ejemplos de dispositivos de conexión serial (COM1, COM2) son: el mouse, el fax módem externo, etc.

**Com1:** Generalmente es de "9 pin macho".

**Com2:** Suele ser de "25 pin macho" como también de "9 pin macho".

Los dispositivos seriales usan distintos pines para recibir y transmitir datos. Los puertos seriales dependen de un chip especial como controlador, el **Universal Asynchronous Receiver/Transmitter (UART)**, para funcionar correctamente. Esencialmente, los puertos serie proveen un conector estándar y un protocolo que te permite conectar dispositivos, tales como módems, microcontroladores a tu computadora. Para conectar un dispositivo a un PC (o sistema operativo) se necesita seleccionar un puerto serie y el cable apropiado para conectar al dispositivo serie. Ambos dispositivos deben concordar en los niveles de voltaje (HIGH y LOW), en la velocidad de transmisión, y en la interpretación de los bits transmitidos. Es decir, que deben de tener el mismo protocolo de comunicación serie (conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre dispositivos).

### 1.3.1 Modos de Transmisión de Datos

En los canales de comunicación existen tres tipos de transmisión:

#### Simplex

En este caso el transmisor y el receptor están perfectamente definidos y la comunicación es unidireccional. Este tipo de comunicaciones se emplean usualmente en redes de radiodifusión, donde los receptores no necesitan enviar ningún tipo de dato al transmisor.



### Duplex o Semi-duplex

En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor y los datos se desplazan en ambos sentidos pero no simultáneamente. Este tipo de comunicación se utiliza habitualmente en la interacción entre terminales y un computador central.



### Full Duplex

El sistema es similar al duplex, pero los datos se desplazan en ambos sentidos simultáneamente. Para ello ambos transmisores poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados, mientras que la comunicación semi-duplex necesita normalmente uno solo.

Para el intercambio de datos entre computadores este tipo de comunicaciones son más eficientes que las transmisiones semi-duplex.

### 1.3.2 Tipos de comunicaciones en serie

#### ➤ Comunicación asíncrona:

Consiste en agregar marcadores dentro del flujo de bits para ayudar a seguir cada bit de datos. Al introducir un bit de inicio que indica el inicio de un corto flujo de datos, la posición de cada bit puede ser determinado cronometrando los bits a intervalos regulares, enviando bits de inicio al comienzo de cada flujo de 8 bits los dos sistemas pueden no estar sincronizados por una señal de reloj, lo único que es importante es que ambos sistemas estén configurados a la misma velocidad. Cuando el dispositivo receptor de la comunicación recibe el bit de inicio comienza un temporizador de corto tiempo. Al mantener los flujos cortos no hay suficiente tiempo para que el reloj salga de sincronía.

Este método es conocido como comunicación asíncrona debido a que las terminales de la comunicación no están sincronizadas por una línea de señal.

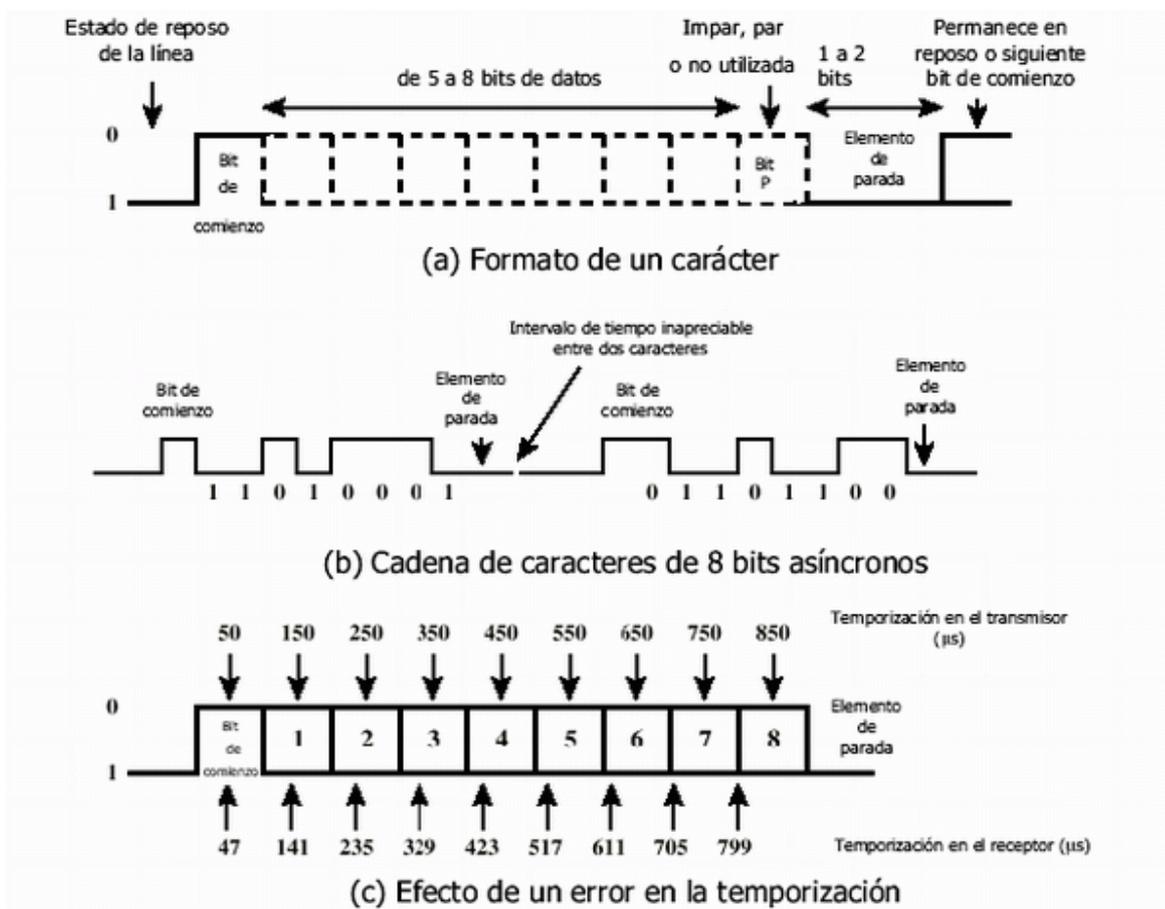
Cada flujo de bits es separado en grupos de 5 a 8 bits llamados palabras. Usualmente en el ambiente de la computación se encuentran palabras de 7 y 8 bits, el primero es para acomodar todas las letras mayúsculas y minúsculas del alfabeto en código ASCII, con un total de 127 caracteres.

Las palabras de 8 bits son utilizadas para corresponder a un byte. Por convención, el bit menos significativo de la palabra es enviado primero y el más significativo al final. Durante la comunicación, el transmisor codifica cada palabra agregándole un bit de inicio al principio y 1 o 2 bits al final. Algunas ocasiones agregará un bit de paridad entre el último bit de la palabra y el primer bit de paro, esto es utilizado como verificación de integridad de datos. Al paquete de bits que se transmiten usualmente se le llama marco de datos.

Se pueden utilizar 5 tipos diferentes de bits de paridad:

- Bit marcador de paridad) siempre es un 1 lógico.
- Bit de paridad de espacio) siempre es un 0 lógico.
- Bit de paridad par es puesto en 1 lógico si cuando al contar el número de bits en la palabra el resultado es par.
- Bit de paridad non es puesto en 1 lógico si cuando al contar el número de bits en la palabra el resultado es non.
- Marco sin bit de paridad, ó sin paridad, es cuando se elimina el bit de paridad del marco.

Según se conforme el marco de datos es como se referencia al marco mismo. Por ejemplo: Palabras de 8 bits, con paridad Non y 1 bit de paro es conocido como 8N1, y la trama es como se muestra en la siguiente imagen:



### Marco de datos de comunicación serial asíncrona

Otro factor importante de toda señal serial asíncrona es la velocidad de comunicación, que es la velocidad a la que los datos son transmitidos. Las velocidades a las que se envía la información comienzan en los 50 bits por segundo (bps) y casi de manera estándar se van duplicando 50, 100, 150, 300, 600, 1200, 2400, 4800, 9600, 19200.... Hasta 384,000.

En un principio el control de software de los puertos seriales resultaba en una gran sobrecarga para el procesador, hasta los 19200 bps, ya que los transportes de datos estaban limitados a 8 y 16 bits. Con la llegada de los transportes de datos MCA, EISA, VLBus y PCI, llegó la ventaja de la posesión de transporte de datos y el acceso directo a memoria elevando la velocidad de transmisión eliminando la sobrecarga del procesador.

Al eliminar la utilización del BIOS y controlando el puerto directamente se comenzaron a obtener velocidades de 115200 bps y superiores, especialmente con la llegada de puertos inteligentes que por sí mismos pueden controlar la comunicación.

Todo el esquema de comunicación serial cambió con la introducción del Bus Serie Universal, mejor conocido por las siglas de su nombre en inglés USB por Universal Serial Bus en 1996, que en sí ya deja de ser un puerto convirtiéndose en un bus con todos los beneficios de esto.

➤ **Comunicación síncrona:**

La comunicación síncrona es aquella en donde los dispositivos de envío y recepción de la comunicación son sincronizados utilizando un reloj que cronometra con precisión el tiempo que separa cada bit. Al verificar el tiempo el dispositivo receptor puede determinar si un bit se ha perdido o si un bit extra, usualmente inducido eléctricamente, ha sido introducido en el flujo de bits. Cualquiera de los dispositivos pierde la señal de tiempo la comunicación es terminada.

### 1.3.3 Detectar errores en la comunicación

Cuando se escriben o se envían datos, pueden producirse errores, entre otras cosas, por ruidos inducidos en las líneas de transmisión de datos. Es por tanto necesario comprobar la integridad de los datos transmitidos mediante algún método que permita determinar si se ha producido un error.

En un caso típico, si al transmitirse un mensaje se determina que se ha producido un error, el receptor solicita de nuevo el mensaje al emisor. Se pueden detectar errores de acuerdo a la forma de transmisión:

1. Transmisión asíncrona:
  - a. Paridad.
  - b. Sobre escritura.
  - c. Error de encuadre (framing).
2. Transmisión síncrona:
  - a. Paridad.
  - b. Sobre escritura.

### 1.3.4 Parámetros que se emplean para establecer una comunicación por puerto serie:

Los parámetros que caracterizan estas comunicaciones en serie son: velocidad; paridad; bits de datos y bits de parada. En la literatura sobre el tema es frecuente expresar estos datos en forma resumida. Por ejemplo: **1200 8 N 1** para indicar una transmisión de 1200 baudios con 8 bits de datos sin paridad y un BIT de Stop. Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra.

Los parámetros anteriores están relacionados con la forma en que se transmite la información serie. En esta comunicación cada carácter va incluido en un cuadro ("**Frame**"); generalmente el comienzo es un BIT de inicio (siempre un 1); después le sigue el dato, que puede ser de 5 a 8 bits de longitud; después puede haber un BIT de control de paridad, y por último un BIT final (siempre es un 1) de longitud variable (el equivalente a 1, 1.5 o 2 bits). Para que dos puertos se puedan comunicar, es necesario que las características (parámetros) sean iguales:

- **Velocidad de transmisión** ("Connection speed") es la cantidad de datos transmitidos en unidad de tiempo. Se expresa en bits por segundo (**bps**) o baudios. Baudio se define como el número de veces que cambia la portadora en un segundo. La velocidad que puede emplearse depende en gran medida de la calidad del medio de transmisión (calidad de la línea), que si (como es frecuente) se trata de líneas telefónicas, depende a su vez de la distancia. Los primeros dispositivos serie operaban a velocidades muy bajas, del orden de 110 a 1200 baudios. Las comunicaciones telefónicas serie actuales están muy cerca del máximo teórico que pueden soportar los pares de cobre utilizados en la telefonía estándar. Además, para aumentar el rendimiento de la comunicación se utilizan técnicas de compresión de los datos a transmitir, y velocidades variables, que pueden ser negociadas entre los equipos que comunican en función del estado de la línea en cada momento.

El número de bits por segundo que se transfieren, se mide en baudios (*bauds*). Por ejemplo, 300 baudios representan 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión. Por ejemplo, si el protocolo hace una llamada a 4800 ciclos de reloj, entonces el reloj está corriendo a 4800 Hz, lo que significa que el puerto serial está muestreando las líneas de transmisión a 4800 Hz. Las velocidades de transmisión

más comunes para las líneas telefónicas son de 14400, 28800, y 33600. Es posible tener velocidades más altas, pero se reduciría la distancia máxima posible entre los dispositivos. Las altas velocidades se utilizan cuando los dispositivos se encuentran uno junto al otro, como es el caso de dispositivos GPIB.

- **Bits de datos:** Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Por ejemplo, el ASCII estándar tiene un rango de 0 a 127, es decir, utiliza 7 bits; para ASCII extendido es de 0 a 255, lo que utiliza 8 bits. Si el tipo de datos que se está transfiriendo es texto simple (ASCII estándar), entonces es suficiente con utilizar 7 bits por paquete para la comunicación. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usar para referirse a todos los casos.
- **Longitud del carácter** ("Char length"). Se han utilizado caracteres de 5, 6, 7 y 8 bits, aunque actualmente los datos son enviados como caracteres ASCII, por lo que pueden utilizarse 7 u 8 bits según se trate del juego de caracteres US-ASCII o el extendido. El conjunto de bits que componen un carácter se denominan **bits de dato** ("Data bits")
- **Paridad** ("Parity"). Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Para paridad par e impar, el puerto serial fijará el bit de paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado alto lógico. Por ejemplo, si la información a transmitir es 011 y la paridad es par, el bit de paridad sería 0 para mantener el número de bits en estado alto lógico como par. Si la paridad seleccionada fuera impar, entonces el bit de paridad sería 1, para tener 3 bits en estado alto lógico. La paridad marcada y espaciada en realidad no verifican el estado de los bits de datos; simplemente fija el bit de paridad en estado lógico alto para la marcada, y en estado lógico bajo para la espaciada. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados. Para poder comprobar la calidad de la transmisión se suele utilizar un sistema de control de paridad que añade un bit a los bits de datos. Los sistemas utilizados son:

- Paridad par ("**E**ven")
- Paridad impar ("**O**dd")
- Paridad marca ("**M**ark")
- Paridad espacio ("**S**pace")
- Sin paridad (no se añade ningún bit de paridad al data grama)

Los sistemas de paridad **par** e **impar** cuentan el número de unos contenidos en los bits de datos y añade un uno o un cero según el resultado. Por su parte la paridad **Mark** indica que se incluirá siempre una marca (bit de valor "1") como bit de paridad, mientras que la paridad **Space** añade siempre un espacio ("0"). Evidentemente estos dos últimos sistemas no aportan absolutamente ninguna información, por lo que son usados muy raramente.

- **Bits de parada** ("Stop bits"). Después que se envía un carácter se envía un bits de parada, que tienen el valor "1" (marca); la duración de este bit puede ser 1, 1.5 o 2 periodos. Este es usado para indicar el fin de la comunicación de un solo paquete. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo la transmisión será más lenta.

### 1.3.5 Conectores de puerto serie

Dispositivo o enchufe que se emplea para la conexión entre el computador y un periférico. El conector más empleado en comunicaciones seriales es el de 25 pines (**DB25**) que tiene una longitud de alrededor de 38 mm; y otro de 9 pines (**DB9**) que tiene una longitud de 17 mm.



### 1.3.6 Características de las comunicaciones en serie:

1. Normalmente estos suelen ser 2 en una placa base y son denominados COM 1 y COM 2.
2. Estos puertos funcionan con un chip llamado UART, que es un controlador serie.
3. El término serie quiere decir que la comunicación con este tipo de conector se realiza sólo en una dirección: o envío, o recepción de datos, pero no las dos al mismo tiempo debido a que envía los datos uno detrás de otro.
4. El puerto serie utiliza direcciones y una línea de señales, un IRQ para llamar la atención del procesador. Además el Software de control debe conocer la dirección.
5. La mayoría de los puertos serie utilizan direcciones Standard predefinidas. Éstas están descritas normalmente en base hexadecimal.
6. Para el protocolo de transmisión de datos, sólo se tiene en cuenta dos estados de la línea, 0 y 1, también llamados Low y High.
7. El conector tiene sus extremos en ángulo de manera que el enchufe podrá introducirse de una manera solamente.

### 1.3.7 Ubicación en el sistema informático:

Se ubican en la parte trasera del case, se podrán identificar estos puertos por los nombres COM 1, COM 2, COM 3. La cantidad de puertos de serie dependen de la tarjeta, ya que hay algunas tarjetas que son capaces de tener 4 u 8 puertos.

### 1.3.8 Interrupciones y direcciones

(Para controlar al puerto serie, la CPU emplea direcciones de puertos de E/S y líneas de interrupción (IRQ).)

Una **petición de interrupción IRQ** ("Interrupt Request") es una señal que se origina en un dispositivo hardware (por ejemplo, un periférico), para indicar al procesador que algo requiere su atención inmediata, se solicita al procesador que suspenda lo que está haciendo para atender la petición. Los puertos serie utilizan recursos del sistema tales como solicitudes de interrupción (IRQ) y direcciones de puertos de entrada/salida, cuando llega una señal al ordenador a través de uno de los puertos, el puerto utiliza una IRQ para indicar al procesador que necesita su atención. Los recursos dedicados a los puertos serie, son casi siempre los mismos en todos los ordenadores.

Las **interrupciones** juegan un papel fundamental, en especial en la operación de dispositivos E/S, ya que les permite enviar estas peticiones a la CPU. Sin ellas el sistema debería chequear constantemente los dispositivos para comprobar su actividad, pero las interrupciones permiten que los dispositivos puedan permanecer en silencio hasta el momento que requieren atención del procesador.

La mayoría de los puertos serie utilizan direcciones standard predefinidas. Éstas están descritas normalmente en base hexadecimal. Cuando se instala un nuevo puerto, normalmente se mueve un jumper o switch para seleccionar un puerto (COM1, COM2, COM3, etc., con lo que se asigna una dirección y una interrupción usada por la tarjeta del puerto. Las direcciones e IRQ usadas por los puertos serie fueron definidas al diseñar el computador originalmente ; sin embargo, las del COM3 y COM4 no se han definido oficialmente, aunque están aceptadas por convenios.

Generalmente, las direcciones de puerto e interrupciones utilizadas (  [2.4](#)), son las siguientes:

Nombre [4]	Dirección	IRQ
COM1 / ttyS0	3F8-3FF	IRQ4
COM2 / ttyS1	2F8-2FF	IRQ3
COM3 / ttyS2	3E8-3EF	IRQ4

COM4 / ttyS3

2E8-2EF

IRQ3

### 1.3.9 Transmisión de datos.

Para el protocolo de transmisión de datos, sólo se tienen en cuenta dos estados de la línea, 0 y 1, también llamados Low y High. Cuando no se transmite ningún carácter, la línea está High. Si su estado pasa a Low, se sabe que se transmiten datos. Por convenio se transmiten entre 5 y 8 bits, pero la BIOS sólo soporta una anchura de datos de 7 u 8 bits. Durante la transmisión, si la línea está Low, se envía un 0 y si está High indica un bit a uno. Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse. A continuación se envía un bit de paridad para detectar errores. Finalmente se mandan los bits de stop, que indican el fin de la transmisión de un carácter. El protocolo permite utilizar 1, 1,5 y 2 bits de stop. El número de bits que componen cada palabra dependen del código a emplear, así, si se quieren transmitir datos ASCII normales, serán de 7 bits, pero si se quiere enviar el juego completo de caracteres, cada una debería estar formada por 8 bits. Dependiendo de esto, si se emplea paridad y el número de bits de stop, se mandarán un determinado número de caracteres dependiendo de la velocidad en baudios (bits por segundo) por el puerto serie.

La especificación de procedimientos es el protocolo, esto es, la secuencia permitida de sucesos. El protocolo se basa en pares de acción-reacción. Por ejemplo, cuando la Terminal establece petición de envío el dispositivo contesta libre para enviar, siempre y cuando este en condiciones de aceptar datos.

Uno de los formatos más usados para el envío de datos es el ASCII.

Tabla Ejemplos ASCII		
ASCII	Hex	Caracter
00 00 00 00	00	(nulo)
00 11 00 00	30	0
00 11 00 01	31	1
01 00 00 01	41	a
01 10 00 01	61	A
01 11 10 10	7B	{

ASCII son las siglas de American Standard Code for Information Interchange o Código Estándar Americano para el Intercambio de información.

El formato ASCII (ilustrado en la Tabla de Ejemplos ASCII) toma un conjunto de 7 bits para representar cualquier carácter, bien sea una letra, un número o un símbolo.

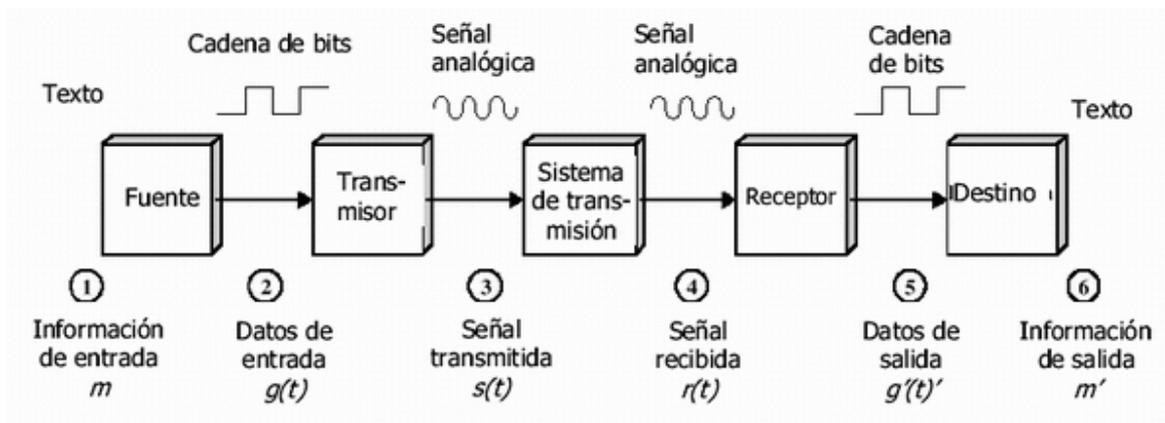
Cuando usted tiene 7 bits, los puede combinar de 128 maneras diferentes, por tanto, ASCII reconoce hasta 128 caracteres codificados. Hay una nueva versión de ASCII que incluye un octavo bit, para un total de 256 caracteres codificados.

La transmisión de datos entre un emisor y un receptor siempre se realiza a través de un medio de transmisión. Se pueden clasificar como guiados y no guiados. En ambos casos, la comunicación se realiza con ondas electromagnéticas. En los medios guiados, como por ejemplo en los pares trenzados, los cables coaxiales y las fibras ópticas, las ondas se transmiten confinándolas a lo largo del camino físico.

Por el contrario, los medios no guiados proporcionan una forma de transmitir las ondas electromagnéticas pero sin encauzarlas, como por ejemplo, en la propagación a través del aire, el mar o el vacío. Los datos se propagan de un computador a un periférico mediante señales eléctricas.

### **1.3.10 Señales**

En un sistema de comunicaciones, los datos se propagan de un punto a otro mediante señales eléctricas. Una señal analógica es una onda electromagnética que varía continuamente. Una señal digital es una secuencia de pulsos de tensión que se pueden transmitir a través de un cable; por ejemplo, un nivel de tensión positiva constante puede representar un 1 binario y un nivel de tensión negativa constante puede representar un 0.



### 1.3.11 Chip controlador: La UART

El corazón del sistema de comunicaciones serie es la **UART**, acrónimo de **Universal Asynchronous Receiver-Transmitter**. Es un chip cuya misión principal es convertir los datos recibidos del bus del PC en formato paralelo, a un formato serie que será utilizado en la transmisión hacia el exterior. También realiza el proceso contrario: transformar los datos serie recibidos del exterior en un formato paralelo entendible por el bus.

La UART es un dispositivo programable en el que pueden establecerse las condiciones que se utilizarán para la transmisión (velocidad, paridad, longitud y bits de parada). En los primeros PCs, eran circuitos integrados 8250 de National Semiconductor (un chip de 40 patillas DIP -Dual-In-Line-), pero se han ido utilizando otros a lo largo de la evolución del PC. Los más significativos se señalan a continuación:

- **NS 8250 (bastante antiguo, con fallos, solo llega a 9600 baudios)**. La UART del IBM PC original; este circuito tenía un pequeño "bug" que fue corregido mediante las oportunas modificaciones en las rutinas BIOS. Fue sustituido por el 8250-B.
- **NS 8250A** Este chip corregía los problemas de los predecesores, sin embargo no podía ser utilizado en los modelos XT precisamente porque sus BIOSes trataban de corregir un problema inexistente. En cualquier caso este chip no funcionaba más allá de 9600 bps.
- **NS 8250B** Este chip fue el último de la saga de los 8250, y reinstaló el antiguo bug, de forma que pudiera funcionar correctamente con las BIOSes de los XT. Todos los de esta serie tenían

un acceso muy lento, lo que obligaba a incluir estados de espera en el procesador. Tampoco funcionaba por encima de 9600 bps.

- **NS 16450 (versión corregida del 8250, llega hasta 115.200 baudios)** . Fue el chip elegido para acompañar a los procesadores i286 de clase AT. Funcionaba bien a 9600 bps, e inauguró la época de módems de alta velocidad. Sin embargo no funcionaba correctamente en los sistemas XT, debiendo ser sustituido por el 16550.
- **NS 16550** Fue el primer chip de su clase dotado con búferes FIFO [4] para transmisión y recepción de 16 bytes, que le hacen especialmente indicado para comunicaciones rápidas. Era más rápido que el 16450, operando por encima de los 9600 bps, pero seguía adoleciendo de algunos problemas, especialmente en los búferes, por lo que también fue sustituido.
- **NS 16550A (con buffers de E/S)**. Es la más rápida y fiable de las UARTs; adecuada para operar con módems de alta velocidad, puede operar a 115 Kbps. Su lógica es compatible pin a pin con el 16450, al que puede sustituir. Además puede utilizar canales DMA.

A partir de aquí existen versiones mejoradas de NS o compatibles con el 16550A de otros fabricantes. En la actualidad, muchas placas-base incluyen uno o dos puertos serie cuyas UART están integradas en el chipset de la propia placa.

## Capítulo 2 Documentación sobre Puerto serie RS232 y puerto USB

### Introducción

En este capítulo se abordará todo lo referente a conceptos y características de los puertos serie USB y RS232, a la transmisión de la información a través de cada uno de estos puertos y los protocolos que emplean el USB y el RS232 para la transmisión de datos.

#### 2.1 Puerto serie RS-232

### 2.1.1 Concepto del puerto Serie RS232

Este estándar fue diseñada para conectar DTEs (Data Terminal Equipment) o equipos terminales de datos como un ordenador con DCEs (data communication equipment) o equipos de comunicación de datos como módems u otro periférico. Esta norma define la interfase mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

**Data Terminal Equipment (DTE):** es el equipo que solicita la conexión, por ejemplo: un ordenador PC, un router. La interfaz serie física que poseen es **MACHO**.

**Data Communication (o Circuit) Equipment (DCE):** es el equipo que ofrece el soporte para la conexión, por tanto tiene acceso a la portadora, a la sincronización, etc. La interfaz serie física que poseen en **HEMBRA**.

Cuando se realiza una conexión serie, siempre hay un equipo que se comporta como un DTE y otro como DCE. Los pines de salida de un conector DTE serán de entrada en DCE y viceversa. Aunque la norma sólo define este tipo de conexión, la interfaz RS232 se ha establecido como estándar general para transmisiones de datos en serie a través de cortas distancias.

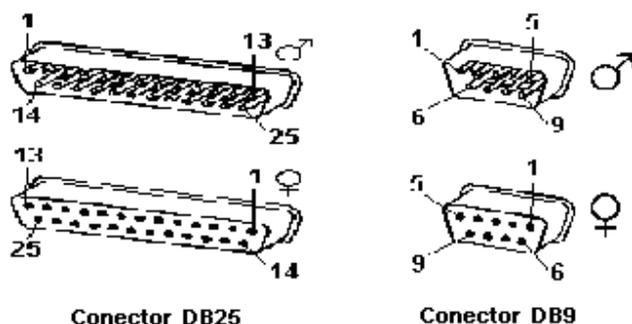
### 2.1.2 Características generales del estándar RS-232

- El 0 lógico (llamado Espacio "Space") comprendido entre +3 y +25 Voltios. Preferiblemente +12V.
- El 1 lógico (llamado Marca "Mark") comprendido entre -3 y -25 Voltios. Preferiblemente -12V. Este constituye el estado que ha de tener de transmisión cuando no se están enviando datos.
- Velocidad de transmisión de hasta 128 000 baudios
- La distancia máxima entre los equipos es de 15 metros.

### 2.1.3 Tipos de conectores que emplea RS232

El RS232 consiste en un conector tipo DB-25 de 25 pines y tipo DB-9 de nueve pines (macho para el DTE y hembra para el DCE), el DB-9 mas barato. En cualquier caso, los PCs no suelen emplear más de 9 pines en el conector DB-25. El RS232 está pensado para usar un cable de nueve hilos conductores en un conector DB9; la longitud del cable es de hasta 15m.

En la figura siguiente se muestran los dos tipos de conectores:



#### Funciones del Conector DB9. Cada ping emplea una función:

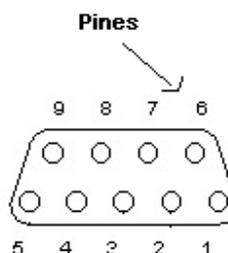
1. **Carrier Detect** (Portador detector). El DCE indica al DTE que se ha establecido una buena conexión y que es adecuado transmitir.
2. **Receive Data** (Receptor)- Envío primario de datos del DCE al DTE. Se mantiene en 1 lógico mientras no haya transmisión. En una transición de ON a OFF de la línea RTS se mantiene en OFF por un corto intervalo para permitir que se complete la transmisión.
3. **Transmit Data** (Transmisor)- Envío primario de datos desde el DTE al DCE. El dispositivo que actúa de DTE mantiene esta línea como un 1 lógico mientras no haya transmisión. Para que exista transmisión en esta línea es necesario que los pines CTS, DTR, DSR y DCD estén en estado de 1 lógico.
4. **Data Terminal Ready** - Cuando esta línea se encuentra en encendido indica al DTE que el DCE está listo para recibir. Esta señal debe estar en encendido antes que el DCE pueda encender el DSR de esta manera señala que existe una conexión para la comunicación.
5. **Signal Ground** (Tierra)- Esta es la tierra lógica. Esta señal se utiliza como el punto de referencia para el resto de señales. Debe estar siempre presente.
6. **Data Set Ready** - El Módem le dice a la computadora que esta listo para hablar. El DCE indique al DTE que el canal de comunicación está disponible.
7. **Request To Send** (Solicitar para envió)- El DTE utiliza esta línea cuando desea transmitir al DCE. Junto con el CTS se coordina la transmisión entre DTE y DCE. Un 0 lógico mantiene al DCE en modo de transmisión

8. **Clear To Send** - Cuando esta señal se encuentra activa le indica al DTE que puede realizar la transmisión. Cuando las líneas TD RTS DSR y DTR están en ON el DTE tiene asegurada la transmisión de la información en caso de estar en OFF es una indicación que el DTE no esta listo.
9. **Ring Indicator** -. En esta línea el DCE indica al DTE que existe una conexión entrante. La señal se pone en ON cuando existe una llamada.

### 2.1.4 Conexiones entre pines del DTE y el DCE. Señales de entrada y salida:

La conexión entre DTE y DCE es simple. Cada pin conecta con su par (el 1 con el 1, el N con el N). Cada pin tiene asignado una función tal y como se muestra en la figura.

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



(\*) = Normalmente no conectados en el DB-25

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Tierra de Señal). Finalmente, existen otras señales como RI (Indicador de Llamada), y otras poco comunes que no se explican en este artículo por rebasar el alcance del mismo. Las señales con la que actúa el puerto son digitales (0 - 1)

Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las más importantes son:

Pin	Función
TXD	(Transmitir Datos)
RXD	(Recibir Datos)
DTR	(Terminal de Datos Listo)
DSR	(Equipo de Datos Listo)
RTS	(Solicitud de Envío)
CTS	(Libre para Envío)
DCD	(Detección de Portadora)

### 2.1.5 Niveles de voltaje:

Todas las normas RS-232 cumplen con los siguientes niveles de voltaje:

- Un "1" lógico es un voltaje comprendido entre  $-5v$  y  $-15v$  en el transmisor y entre  $-3v$  y  $-25v$  en el receptor.
- Un "0" lógico es un voltaje comprendido entre  $+5v$  y  $+15v$  en el transmisor y entre  $+3v$  y  $+25v$  en el receptor.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el puerto serial del ordenador, ya que son diferentes a los que utilizan los microcontroladores y los demás circuitos integrados.

El 1 lógico (llamado Marca "Mark") comprendido entre  $-3$  y  $-25$  Voltios. Preferiblemente  $-12V$ . Este constituye el estado que ha de tener de transmisión cuando no se están enviando datos. El 0 lógico (llamado Espacio "Space") comprendido entre  $+3$  y  $+25$  Voltios. Preferiblemente  $+12V$ , constituye el estado que ha de tener de transmisión cuando se están enviando datos

### 2.1.6 Características funcionales:

Cuando la Terminal o computadora se enciende, establece (es decir, pone en un 1 lógico) la línea Terminal de datos preparada (pin 20). Cuando el periférico se enciende, establece la línea conjunto de datos preparado (pin 6). Cuando el periférico detecta una portadora en la línea telefónica establece la línea de detección de portadora (pin 8). La petición de envío (pin 4) indica que la Terminal quiere

enviar datos. Libre para enviar (pin 5) indica que el periférico esta preparado para aceptar datos. Los datos se transmiten por el circuito transmitir (pin2) y se reciben por el circuito recibir ( pin 3).

### 2.1.7 Tipo de conexiones entre el ordenador y el dispositivo por puerto serie RS232:

#### ➤ Conexión tipo: modem nulo

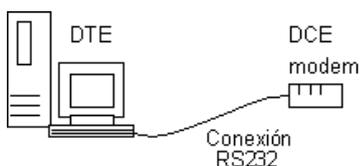
- Conexión entre DTE.
- Protocolo para adaptar velocidades mediante las señales (DTR, DSR, CTS)



**FIGURA 6: Conexión directa mediante cable NULL-MODEM**

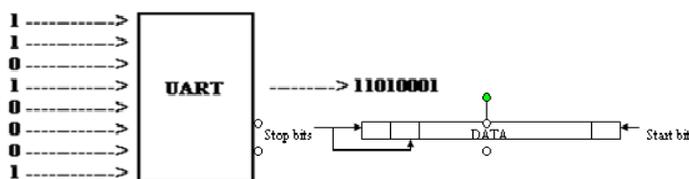
#### ➤ Conexión mediante módem

Los módem se conectan con el ordenador a través de un puerto de comunicaciones del primero. Estos puertos siguen comúnmente la norma RS232.



### 2.1.8 UART: Universal Asynchronous Receiver Transmitter

Es el circuito electrónico utilizado para enviar y recibir información por el puerto serie. Su función es convertir **bytes en bits serie** para transmitirse a bits por segundo.



**PC's BUS Slot**

**Serial port**

Circuitos integrados UART más utilizados (*todos son compatibles*): 8250 UART sin buffer , 16450 UART con 1 byte buffer , 16550 UART - 16 byte buffer.

140

La información mandada por la UART **es asíncrona**.

**2.1.9 Transmisión en RS232 :**

Cuando se trasmite utilizando este estándar se debe tomar en cuenta algunos factores, como la velocidad de transmisión del puerto, control de flujo y paridad.

En una maquina con Windows, la velocidad máxima que se puede llegar a utilizar es de hasta 38400 bps en transmisiones seriales. La RS-232 permite la transmisión síncrona y asíncrona. La subnorma asíncrona es sin duda la más frecuente. La transmisión asíncrona se lleva a cabo tal y como se dice en epígrafe anteriores. En concreto utiliza:

- 5, 6, 7 ó 8 bits de datos.
- 0 ó 1 bit de paridad (la paridad puede ser "par"(Even), "impar"(Odd), "siempre a cero"(Reset) y "siempre a uno"(Set).
- 1, 1.5 o 2 bits de STOP.

Para agilizar el lenguaje se suele emplear una nomenclatura abreviada como, por ejemplo, "8N1" que indica que la transmisión serie RS-232 se ha configurado para transmitir 8 bits de datos, No paridad y 1 bit de STOP. Otro ejemplo sería "6E2" que indica 6 bits de datos, paridad par y 2 bits de STOP. La transmisión de datos mediante el hardware se realiza a través de las señales antes mencionadas. Por ejemplo, para conectar dos ordenadores personales (dispositivos DTE) con señales de acoplamiento, sería necesario efectuar las conexiones descritas en la figura 2. En estos esquemas, la dirección de las flechas indica realmente el sentido en que se mueve la información, es decir, el emisor y el receptor de la señal.

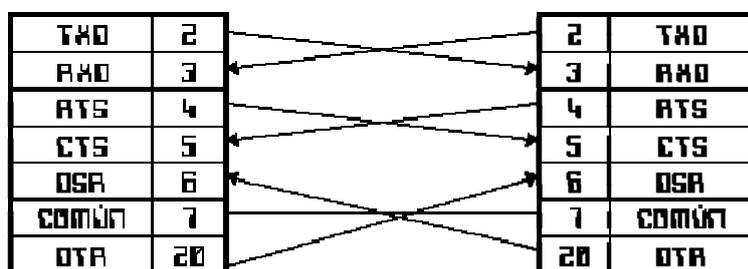


Fig. 2. Conexión estándar entre dos equipos RS-232-C DTE.

En el caso de no desear utilizar estas señales de acoplamiento, puede optarse por proporcionarlas por un medio físico, pues algunos programas de comunicación pueden requerir su presencia. Un posible esquema para esta conexión, puede ser el indicado en la figura 3. Se trata de un esquema más sencillo, pero puede funcionar en una gran parte de equipos, siempre que no se desee trabajar al límite de la capacidad de los dispositivos.

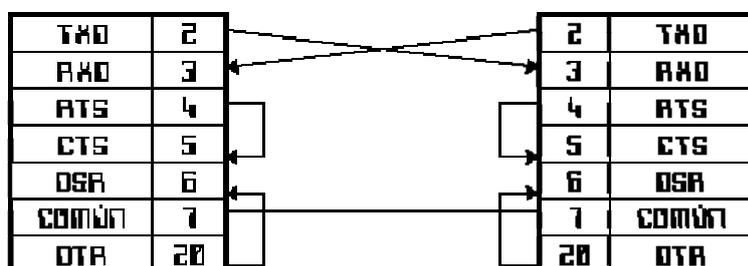


Fig. 3. Conexión entre dos equipos RS-232-C DTE sin utilizar acoplamientos

La transmisión de datos mediante software se realiza con la apertura e inicialización del puerto RS232, después se implementan los métodos que son las operaciones que generan a cada uno de los comandos diseñados en el protocolo para ejercer alguna función como transmitir datos, entrada de datos, cambiar datos, salida de datos, etc. El transmisor es el encargado de codificar esta información cogiendo una pequeña trama del paquete de datos o comando y codificándola respecto a: Bit de datos, bit de inicio, bit de stop, y bit de paridad. Esta información para ser enviada hacia el exterior pasa hacia el chip controlador que es el encargado de convertir el formato paralelo en un formato serie para que

sea entendido por el conector y cable RS232, el receptor es encargado después de cambiar este formato a un formato paralelo.

**Velocidad de transmisión:** Esta interface está hecha para tasa de datos de hasta 20 Kbps, así como cables de hasta 15m. La capacidad máxima que se alcanza en este tipo de puerto es de 20Kb/s. En cuanto a la velocidad, en el caso de los puertos RS-232 la unidad de medida es el **Baudio**, en lugar de utilizar el más habitual hoy en día de **bit por segundo**, siendo el ratio de entre 75 baudios y 128000 baudios, aunque los más utilizados son 9600, 14400 y 19200 baudios.

#### 2.1.10 Control de Flujo en el RS-232

En RS-232 el control de flujo se puede hacer de dos maneras: por hardware (RTS/CTS) o por software (Xon/Xoff).

- Por Software: El carácter Xoff es utilizado por el receptor para decir que su buffer está lleno y el emisor debe esperar. Cuando vuelva a tener espacio en el buffer, enviara el carácter Xon diciéndole que puede volver a transmitir.
- Por hardware: a diferencia del anterior, este tipo de control de flujo requiere que entre su PC y su periférico se conecten dos hilos: RTS y CTS. Cuando el buffer del receptor se llena (suponga que es un periférico "lento") le dice al PC que espere, desactivando la señal CTS. Cuando vuelva a tener espacio en el buffer, activa nuevamente el CTS para decir que está nuevamente listo (esto puede pasar cuando la UART es más rápida que el periférico). Las líneas DSR y DTR son las primeras en activarse cuando existe alimentación en los equipos. Estando ambas activadas, cuando el DTE quiere mandar información lo hace activando la línea RTS. Si el DCE tiene establecida conexión y puede cursar la información, activa la línea CTS. Si quiere deshabilitar el envío, desactiva CTS y el DTE deja de mandar información.

Por otro lado se utiliza la opción control de flujo para asegurarse que el otro dispositivo esta recibiendo correctamente la información. Este control de envío de información también se puede hacer por software o por Hardware, incluso por una combinación de los dos.

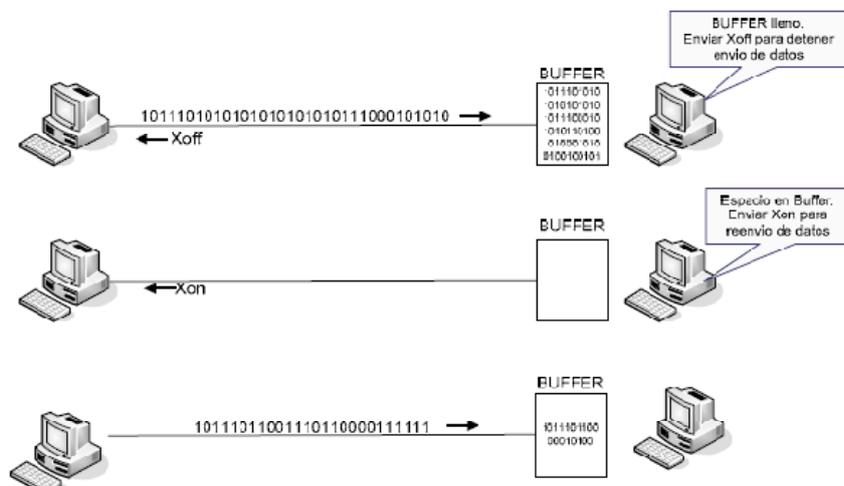


FIGURA 7: Control de flujo

Un importante aspecto de las comunicaciones seriales es el concepto de Control de flujo. Esto es la habilidad de un dispositivo para decirle a otro que deje de enviar datos por un tiempo. Los comandos Request to Send (RTS), Clear to Send (CTS), Data Terminal Ready (DTR) y Data Set Ready (DSR) son usados para activar el control de flujo.

### 2.1.11 Configuración del protocolo:

Antes de iniciar cualquier comunicación con el puerto RS232 se debe de determinar el protocolo a seguir dado que el estándar del protocolo no permite indicar en que modo se esta trabajando, es la persona que utiliza el protocolo el que debe decidir y configurar ambas partes antes de iniciar la transmisión de datos.

Siendo los parámetros a configurar los siguientes:

- Protocolo serie (número bits-paridad-bits stop).
- Velocidad de puerto.
- Protocolo de control de flujo (RTS/CTS o XON/XOFF).

## 2.2 Estándar USB



### 2.2.1 Concepto del Bus Serie Universal (USB):

El bus universal en serie, consiste en una norma para bus periférico, USB permite adjuntar dispositivos periféricos al ordenador rápidamente, sin necesidad de reiniciarlo ni de volver a configurar el sistema. Los dispositivos con USB se configuran automáticamente tan pronto como se han conectado físicamente. Este puerto utiliza: cable de 4 hilos, 2 para datos y 2 para alimentación. La versión actual del protocolo USB es la 2.0, pero su especificación ha tenido varias revisiones a lo largo de la historia, las cuales se muestran en la siguiente tabla.

<u>Estándar</u>	<u>Publicación</u>	<u>Detalles</u>
USB 0.9	Noviembre, 1995	Primer borrador
USB 1.0	Enero, 1996	Velocidad baja y velocidad completa
USB 1.1	Setiembre, 1998	Añade detalles y precisiones a la versión 1.0.
USB 2.0	Abril, 2000	Incorpora modo de alta velocidad

La versión 1.1 es el estándar mínimo que debe cumplir todo dispositivo USB hoy en día. La última versión (2.0) soporta tasas de transferencia de altas velocidades, comparables (o incluso superiores) a la de un disco duro o almacenamiento magnético, lo cual ha permitido ampliar el uso del USB a aplicaciones de video y almacenamiento masivo. Una de las razones a la cual se atribuye su gran popularidad es que todas las versiones del protocolo son compatibles hacia atrás. Es decir, que cualquier dispositivo 2.0 puede ser conectado a un dispositivo 1.0, aunque funcionará la velocidad del más lento. Las tres velocidades de transferencia soportadas son:

<u>Velocidad</u>	<u>bytes/s</u>	<u>bits/s</u>
Baja	183 Kbytes/s	1.5Mbit/s
Velocidad completa	1.4 Mbytes/s	12Mbit/s
Alta velocidad	57 Mbytes/s	480 Mbit/s

**Tabla 3.2** Velocidades de transferencias USB

El tipo de Alta velocidad fue introducido en la revisión USB 2.0, mientras que los otros ya existían desde la 1.0. Los de baja velocidad generalmente son dispositivos de interacción con la computadora como ratones, teclados y joysticks. **USB 1.1** alcanza los **12 Mb/s** y hasta los **480 Mb/s (60 MB/s)** para **USB 2.0**, mientras un puerto serie o paralelo tiene una velocidad de transferencia inferior a **1 Mb/s**.

**USB** (Universal Serial Bus) es un estándar en los ordenadores de última generación que incluyen al menos dos puertos **USB 1.1**, o puertos **USB 2.0** en los más modernos.

El puerto **USB 2.0** es compatible con los dispositivos **USB 1.1**. El cable **USB** permite también alimentar dispositivos externos a través de él, el consumo máximo de este controlador es de 5 voltios. Los dispositivos se pueden dividir en dispositivos de bajo consumo (hasta 100 mA) y dispositivos de alto consumo (hasta 500 mA) para dispositivos de más de 500 mA será necesario alimentación externa.

Hay que tener en cuenta que la longitud del cable no debe superar los 5 mts. El USB debe cumplir las especificaciones del Standard USB iguales para la 1.1 y la 2.0. Los dispositivos **USB 2.0** sólo pueden ser conectados a puertos **USB 2.0** aunque los dispositivos **USB 1.1** sí pueden ser conectados a puertos **USB 2.0** además del propio **USB 1.1**

### 2.2.2 Objetivos de USB

El objetivo del USB es paliar las carencias del puerto serie RS-232 (diez veces más lento) y del puerto paralelo (manguera con muchos hilos de longitud limitada), además ambos puertos sólo permiten conectar un dispositivo al mismo tiempo. El USB, con una manguera flexible de pocos hilos consigue velocidades muy por encima de las que se pueden transmitir con ambos tipos de puertos.

### 2.2.3 Funcionamiento

La placa base tiene normalmente dos conectores estandarizados que sirven para conectar dos dispositivos USB, pero para conectar hasta 127 dispositivos posibles se necesita utilizar HUBS (concentradores) USB con varios puertos USB cada uno, hasta llegar a totalizar como máximo 127 dispositivos, de modo que un dispositivo USB se puede conectar directamente al conector de la placa base o a un conector de HUB, sin variar para nada su funcionamiento. De hecho, algunos dispositivos pueden funcionar como HUBs al tener conectores USB incorporados, como los teclados. También se puede conectar un dispositivo a un hub, que a su vez esté conectado a otro hub que está conectado al conector de la placa base y el funcionamiento del dispositivo será igual

que estando conectado directamente al conector de la placa base. El cable de los dispositivos USB es un cable de 4 hilos con una longitud máxima de 5 metros por dispositivo o HUB, con lo que los dispositivos conectados no tienen por qué estar amontonados encima de una mesa. Tan pronto se conecta un dispositivo USB (no hace falta apagar el equipo) el controlador USB detecta una diferencia de voltaje en el puerto USB e intenta identificar el dispositivo nuevo y sus características. Si no es capaz de identificarlo, el host pide que se introduzca el disco de instalación, y una vez instalado el drive ya se puede empezar a funcionar con el nuevo dispositivo, normalmente sin reiniciar el equipo. El puerto USB utiliza una única IRQ para todos los dispositivos conectados (hasta 127), pero para identificarlos, a cada uno le asigna una ID (única a cada uno), no hay necesidad de configurarlos, pues todos los dispositivos comparten los mismos recursos del controlador USB de la placa base. Cuando se desconecta un dispositivo USB, de nuevo sin necesidad de reiniciar el equipo, el controlador USB detecta de nuevo una diferencia de voltaje, hace una búsqueda de dispositivos para averiguar cuál ha sido conectado e inmediatamente descarga los drivers del dispositivo para así ahorrar memoria del sistema.

El controlador USB instalado en el ordenador, denominado **controlador de host**, o **concentrador raíz** ("Root hub"), proporciona un enlace entre el bus de la placa-base, por ejemplo PCI, y una o más conexiones iniciales con el exterior (generalmente 2 conectores del tipo "A"). A partir de estas, utilizando concentradores adicionales, pueden conectarse más dispositivos.

Actualmente la mayoría de las placas-base incluyen un controlador USB integrado en el chipset. Para sistemas antiguos que no dispongan de USB pueden instalarse tarjetas PCI que incluyen un controlador y uno o dos conectores de salida.

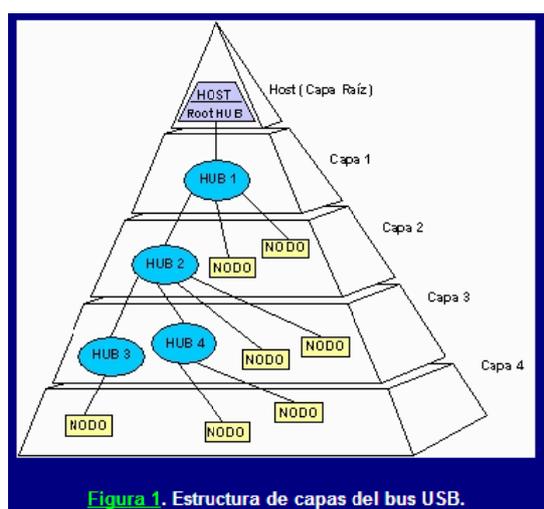
Las operaciones entre el dispositivo USB y el controlador USB se realizan a través de una cola de transferencia. Este método juega un papel importante en el sistema USB.

El funcionamiento del USB está centrado en el host, todas las transacciones se originan en él; es el controlador host el que decide todas las acciones, incluyendo el número asignado a cada dispositivo (esta asignación es realizada automáticamente por el controlador "host" cada vez que se inicia el sistema o se añade, o elimina, un nuevo dispositivo en el bus).

## 2.2.4 Descripción del Sistema USB

Compuesto por tres partes diferenciadas:

- Controlador o host.
- Dispositivos USB
- Interconexión



### ➤ Host USB.

A diferencia de los dispositivos y los hubs, existe tan solo un host dentro del sistema USB, que como ya se dijo es el ordenador mismo, particularmente una porción del mismo denominado Controlador USB del Host. Este tiene la misión de hacer de interfaz entre el ordenador mismo y los diferentes dispositivos. Existen algunas particularidades respecto a este controlador. Su implementación es una combinación de hardware y software todo en uno, es decir **Firmware**. Puede proveer de uno o dos puntos de conexión iniciales, denominados Hub raíz, a partir de los cuales y de forma ramificada irán conectándose los periféricos.

#### • Host USB - Hardware y Software

- El HOST es responsable a nivel hardware dentro del sistema USB de:
  - Detectar y configurar nuevos dispositivos
  - Administrar y controlar flujo de datos y de control entre el host y los dispositivos USB.

- Obtener información de los elementos del sistema.
- Proveer una cantidad de energía eléctrica limitada para aquellos dispositivos que pueden abastecerse con tan solo la energía proveniente del computador.
- o A nivel de software:
  - Enumeración y configuración de los dispositivos del sistema.
  - Administración de control de transferencias de información.
  - Administración avanzada de suministro eléctrico a los dispositivos
  - Adiministración de la información del bus y los dispositivos USB.

### ➤ **Dispositivos USB**

En cuanto a los tipos de dispositivos se encuentran dos clases:

- **Hubs**, que proporcionan los puntos de acople adicionales al USB.
- **Funciones**, que le dan al sistema la funcionalidad (impresoras, unidades de almacenamiento...)

Los diferentes tipos de dispositivos USB se agrupan en dos grupos bien diferenciados que son los: periféricos y los Hub's.

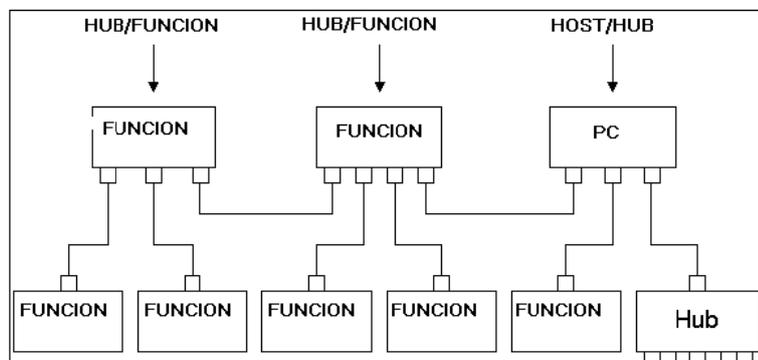
#### • **Funciones USB o Periféricos:**

Dentro de la terminología USB, todos los dispositivos que pueden ser conectados a este bus, a excepción de los Hubs, se denominan Funciones. Son funciones típicas: el ratón, el monitor, modem, etc.

Las funciones o dispositivos periféricos, son capaces de recibir y transmitir información, ya sea del usuario o de control. El común denominador de todas las funciones USB es su cable y el conector del mismo, diseñado y fabricado de acuerdo a las especificaciones del bus, por lo que no cabe preocuparse por la compatibilidad entre equipos de diferentes fabricantes.

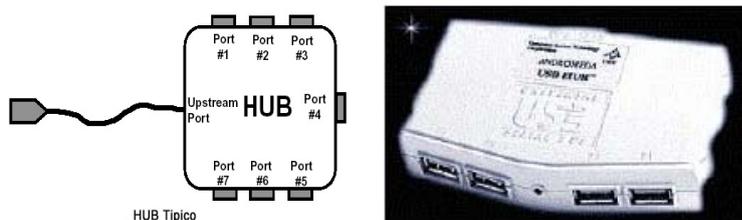
Un aspecto interesante de las funciones, es que pueden ser a la vez nuevos hubs. La siguiente figura muestra un esquema en el que el PC tiene tres puertos, la siguiente función que puede ser un monitor 4, el siguiente 3 y adicionalmente un Hub, provee 4 puertos mas, es un esquema tan sencillo, donde

existen 14 puertos disponibles para todo tipo de periféricos, entre los que se pueden citar: ratón, lápiz óptico, teclado, impresora, un teléfono ISDN, etc.



• **Los Hub's USB.**

Los hubs son un elemento clave en la arquitectura [Plug and Play](#) del USB.



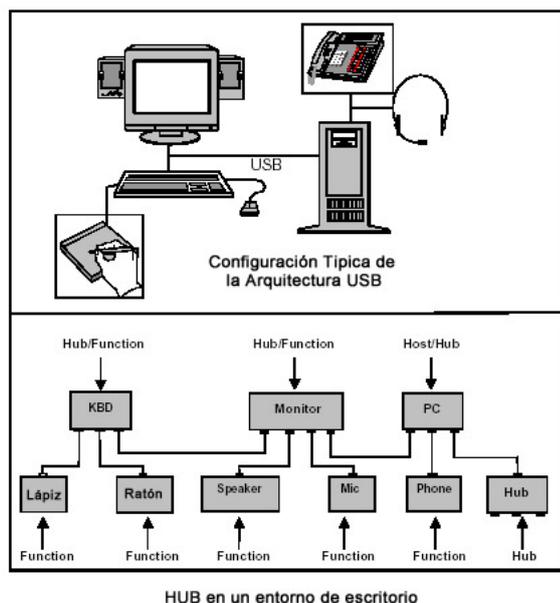
Un bus tradicional puede ser dividido en segmentos de bus individuales conectados por puentes. Cada segmento tiene un número limitado de puntos de conexión debido a la limitada energía y la carga del bus. En la arquitectura USB, el número de puntos de conexión de dispositivos se expande añadiendo dispositivos únicos llamados **HUBs** o concentradores. Estos dispositivos expanden la arquitectura del USB de dos formas:

- Incrementando los puntos de conexión a través de puertos adicionales
- Proporcionando energía a los dispositivos al expandir el bus

Los HUBs sirven para simplificar la conectividad USB desde la perspectiva del usuario y proporcionar robustez y complejidad a un precio relativamente bajo. Según esta arquitectura, se pueden expandir el bus acoplando hubs adicionales, interconectando dichos hubs mediante enlaces. Cada hub convierte

un punto simple de conexión en múltiples puntos, donde estos puntos de conexión se les llaman puertos.

En la siguiente foto se puede observar un ejemplo de un uso típico de la arquitectura USB con la conexión de diversos dispositivos.



### ➤ Interconexión en el BUS USB

- Topología en estrella estratificada piramidalmente.
- Cada centro de la estrella es un HUB: dispositivo que, por un lado se conecta a un PC o a otro HUB, y por el otro permite conectar varios dispositivos ó más hubs.

### 2.2.5 Señales y conectores

Las señales USB son transmitidas en un par trenzado (cuyos hilos son denominados D+ y D-) utilizando señalización diferencial half-duplex minimizando el ruido electromagnético en tramos largos. El diseño eléctrico permite un largo máximo de 5 metros, sin necesidad de un repetidor intermediario.

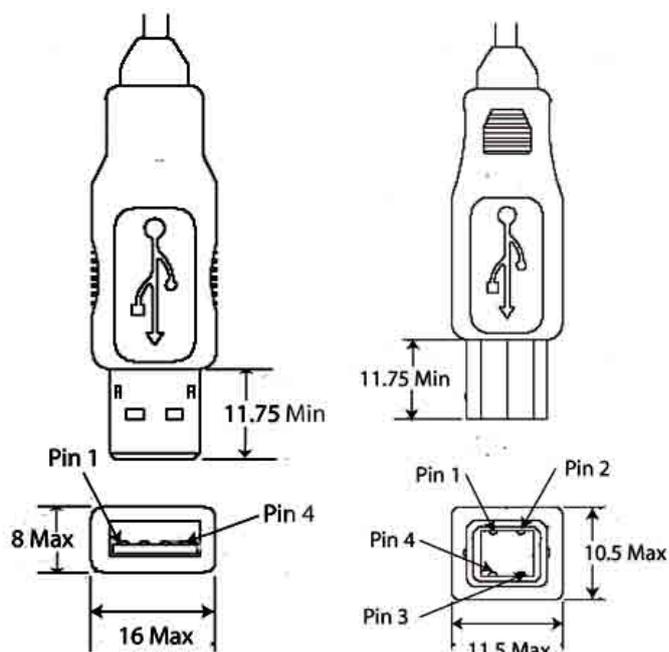
Existen dos tipos de conectores: estándar y mini. Los estándar son los que típicamente se encuentran en un computador y vienen en dos tipos: A (puerto del ordenador) y B (puerto del dispositivo). El tipo A es el que es chato y se encuentra del lado del host controlador, mientras que el tipo B es el cuadrado y se encuentra del lado del dispositivo. Todos los cables son machos, mientras que los enchufes (ya sea en la computadora o los dispositivos) son hembra. No existen intercambiadores de género puesto que las conexiones cíclicas no están permitidas en un bus USB. Los conectores mini siguen la misma política que los estándares pero son utilizados para dispositivos pequeños como celulares.

Los pines de un cable USB son los siguientes:

<u>Pin</u>	<u>Color</u>	<u>Función</u>
1	Rojo	V.BUS (4.4 - 5.25 V)
2	Blanco	D-
3	Verde	D+
4	Negro	Tierra
5	en corto con pin 4 en conector Mini-A, utilizado para USB On-The-Go	

**Tabla 3.3** Pines del bus USB

A continuación se muestra un diagrama de los conectores (las medidas están en mm) y los números de los pines se corresponden con la tabla anterior.

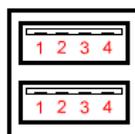


### 2.2.6 Nivel de voltaje.

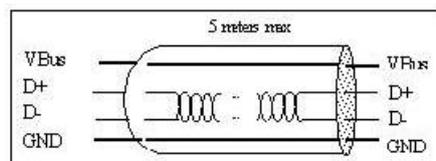
Los pines del conector se identifican a continuación, como también el nivel de voltaje:

Conector Pin Señal

- 1 +5V
- 2 Datos -
- 3 Datos +
- 4 a masa



El Bus Serial Universal transfiere señales de información y energía eléctrica a través de 4 cables, cuya disposición se muestra en las siguientes figuras.



Los dos cables VBus y GND tienen la misión de llevar suministro eléctrico a los dispositivos, con potencial de +5V para VBus. Los cables USB permiten una distancia que va de los pocos centímetros a varios metros, más específicamente 5 metros de distancia máxima entre un dispositivo USB y el siguiente. El cable **USB** permite también alimentar dispositivos externos a través de él, el consumo máximo de este controlador es de 5 voltios. Es importante indicar que los cables USB tienen protectores de voltaje a fin de evitar cualquier daño a los equipos, son estos mismos protectores los que permiten detectar un dispositivo nuevo conectado al sistema y su velocidad de trabajo.

### **2.2.7 Topología del BUS y Protocolo USB para la transmisión de datos.**

La forma física en la que los elementos se interconectan dentro del sistema USB, puede asemejarse a la topología en estrella. El centro de cada estrella es un hub, un dispositivo que por un lado se conecta al computador o a otro hub y por otro lado, permite conectar al mismo varios dispositivos o en su defecto nuevos hubs.

**El protocolo que lleva a cabo USB se describe a continuación:**

#### **Cola de transferencia:**

Las operaciones entre el dispositivo USB y el controlador USB del PC, se realizan a través de un mecanismo denominado cola de transferencia. Este método juega un papel importante en el sistema USB. Mediante software, se construye una cola de operaciones como se muestra en la figura A.1.

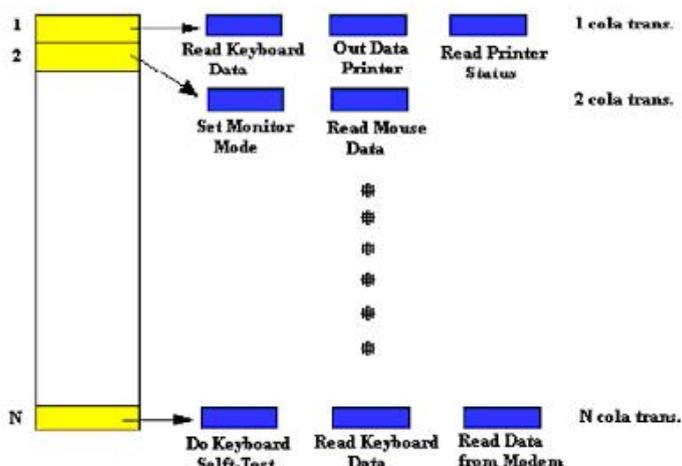


Figura A.1

Antes de que el controlador inicie una operación con un dispositivo, el driver USB, coloca la petición de operación en la cola. El controlador inspecciona la cola de peticiones de transferencia y estas son llevadas a cabo de forma secuencial “FIFO”, o sea, por orden de entrada.

Una petición de transacción consiste en: una petición de transferencia de información útil para el controlador USB del PC.

- La petición de transferencia puede ser para enviar un comando USB al dispositivo, o lectura / escritura de datos a un dispositivo o desde este.
- Información útil para el controlador como: la dirección del dispositivo, situación del buffer, tipo de transferencias (sincrónica, asincrónica, interrupción, etc.), máximo de bytes del dato, etc.

**Transacción:**

Cuando el controlador USB ejecuta una transacción u operación específica de la cola de transferencia, se envían hasta tres paquetes en la transmisión. Cada transacción comienza con el envío de un paquete que describe el tipo y dirección de la transacción, la dirección del dispositivo USB y el “end point number”, que indica al dispositivo.

La transacción se realiza, enviando los datos desde el dispositivo al controlador o viceversa, dependiendo del tipo de transacción. Para finalizar la transacción, el receptor de los datos (el

dispositivo o el controlador) manda el tercer paquete, que se utiliza como confirmación del éxito de la comunicación. Cuando una petición se realiza, se pasa a la siguiente de la cola de transferencia, y se opera de la misma manera.

El controlador debe realizar esta operación en el período de un milisegundo; si los datos a enviar requieren más tiempo, entonces se colocara una petición extra cola de transferencia, para realizarla posteriormente.

Una transacción USB esta contenida en una de estas dos categorías:

- Transferencia de control / estado de un dispositivo.
- Transferencia de datos.

La transferencia de control / estado, se usa para que el controlador pueda configurar un dispositivo. Sin embargo, la transferencia de datos sirve para el envío de un flujo de datos. La transferencia de datos se puede realizar tanto desde el controlador, como desde el periférico. Mientras que la transferencia de control siempre se realiza desde el controlador hacia el dispositivo.

Un ejemplo muy sencillo seria el envío de un comando que configure un altavoz USB en moda stereo, realizándose una transferencia de control. A continuación se envía los datos de sonido, realizando una transferencia de datos.

### **2.2.8 Características generales del USB**

La especificación del USB proporciona una serie de características que pueden ser distribuidas en categorías. Estas características son comunes para todas las versiones (desde la 1.0 hasta la 2.0).

- Todos los dispositivos USB deben tener el mismo tipo de cable y el mismo tipo de conector, más allá de la función que cumplan.
- Los detalles de consumo y administración eléctrica del dispositivo deben ser completamente transparentes para el usuario.
- El computador debe identificar automáticamente un dispositivo agregado mientras opera, y por supuesto configurarlo.
- Los dispositivos pueden ser desconectados mientras el computador está en uso.
- Deben poder compartir un mismo bus tanto dispositivos que requieren de unos pocos Kbps como los que requieren varios Mbps.

- Más de 127 dispositivos diferentes pueden estar conectados simultáneamente y operando con un mismo ordenador sobre el Bus Serial Universal.
- El bus debe permitir periféricos multifunción, es decir aquellos que pueden realizar varias tareas a la vez, como lo son algunas impresoras que adicionalmente son fotocopiadoras y máquinas de fax.
- Capacidad para manejo y recuperación de errores producidos por un dispositivo cualquiera.
- Soporte para la arquitectura Conectar y Operar (Plug&Play).
- Bajo costo.
- No se necesita un cable extra de alimentación - la mayoría de los periféricos USB obtienen la alimentación del bus USB, con lo cual no requieren un cable de alimentación adicional.
- Más rápido: USB transfiere los datos 10 veces más rápido que los puertos serie tradicionales.

**Ahora se va a proceder a la breve explicación de alguna de las características anteriores:**

### **PLUG & PLAY**

El puerto USB, y por tanto todos los dispositivos con conexión USB, son verdaderamente Plug & Play, es decir, el dispositivo es detectado automáticamente al conectarlo al equipo y el sistema operativo instala el driver adecuado o pide el disco de instalación. Aún más, no es necesario apagar, ni siquiera reiniciar el equipo para conectar o desconectar los dispositivos, cargándose y descargándose automáticamente de memoria el driver correspondiente.

### **RECURSOS DEL DISPOSITIVO**

Quizá una de las mayores ventajas para los equipos actuales es el hecho de que el puerto USB solamente necesita una IRQ y una dirección de memoria y todos los dispositivos conectados a él, solamente necesitan una ID para su identificación (como en el bus SCSI) dentro de la cadena de 127 dispositivos, sin necesitar más recursos. Si se tiene en cuenta que los puertos estándar (dos serie, un paralelo, un PS/2 para ratón, uno para teclado, un puerto para joystick) consumen 5 IRQs, algunas DMA y múltiples direcciones de memoria, al utilizar dispositivos USB se están ahorrando valiosos recursos del sistema.

### **SIMPLICIDAD**

El manejo de los dispositivos USB se hace por software, concretamente por el propio sistema operativo, por lo que los dispositivos USB son más fáciles de fabricar y por tanto más baratos. Además, USB es una tecnología abierta por la que no hay que pagar derechos, lo que siempre abarata los costos de fabricación.

## DISPOSITIVOS

Se pueden conectar hasta un total de 127 dispositivos en cadena o utilizando HUBs USB (concentradores de puertos USB), y cada dispositivo puede tener un cable de hasta 5 metros de longitud, frente a 1 metro para el puerto serie y 4 metros para el puerto paralelo. Además, conectándolos en cadena, el último dispositivo puede estar a 635 metros del ordenador. Longitud máxima del cable entre dispositivos es de 5 metros.

## VELOCIDAD

El puerto serie es capaz de transmitir hasta 112,5 KB/s y el puerto paralelo entre 600KB/s y 15MB/s, pero el puerto USB es capaz de llegar a alcanzar entre 1,5MB/s , 12MB/s y 480 Mbit/s.

Seguido se mostrarán algunos detalles de cada una de las versiones, que es lo que las diferencia unas de otras. Estas diferencias serán más bien escasas en cuanto a características generales.

### 2.2.9 Explicación detallada de las versiones de USB

#### Versión USB 1.1

El USB 1.1 es un bus externo que soporta tasas de transferencia de datos de 12 Mbps. Empezó a utilizarse en 1996, algunos fabricantes de ordenadores empezaron a incluir soporte para USB en sus nuevas máquinas. Se espera que sustituya totalmente a los puertos de serie y paralelos.

#### RENDIMIENTO

#### VELOCIDAD BAJA

Dispositivos Interactivos

#### APLICACIONES

Teclado y ratón

Periféricos de entretenimiento (juegos y

#### ATRIBUTOS

Bajo coste

de Conexión en caliente y

realidad virtual)

10-100 Kb/s	Configuración del Monitor	Fácil uso Múltiples periféricos
-------------	---------------------------	------------------------------------

### VELOCIDAD MEDIA

Teléfono	<u>ISDN/RDSI</u>	Bajo coste
Audio	<u>PBX</u>	(Des)Conexión en caliente
Video comprimido	<u>POTS</u>	Fácil uso
500 Kb/s - 10 Mb/s	Audio	Múltiples dispositivos Garantizada la latencia Garantizado el ancho de banda

### USB 2.0

La especificación del USB 2.0 fue lanzada en abril de 2000. También conocido como *USB de alta velocidad*, el USB 2.0 es un bus externo que soporta tasas de transferencia de datos de hasta 480Mbps.

El USB 2.0 es una extensión del USB 1.1, utiliza los mismos cables y conectadores y es completamente compatible con USB 1.1.

Hewlett-Packard, Intel, Lucent, Microsoft, NEC y Philips tomaron juntos la iniciativa para desarrollar una tasa de transferencia de datos más alta que la del USB 1.1 para resolver las necesidades de ancho de banda de las nuevas tecnologías.

En total USB 2.0 soporta tres tipos distintos de velocidad:

- Hi-Speed 480 Mbps.
- Full-Speed 12 Mbps.
- Low-Speed 1.5 Mbps.

El alcance de uso para la versión USB 2.0 se ve incrementada notablemente respecto a sus predecesoras. Así quedaría la tabla anterior modificada.

<u>RENDIMIENTO</u>	<u>APLICACIONES</u>	<u>ATRIBUTOS</u>
<b>VELOCIDAD BAJA</b>	Teclado y ratón	Bajo coste
Dispositivos Interactivos	Periféricos de entretenimiento (juegos y (Des) realidad virtual)	Conexión en caliente
10-100 Kb/s	Configuración del Monitor	Fácil uso Múltiples periféricos
<b>VELOCIDAD MEDIA</b>		
Teléfono	<u>ISDN/RDSI</u>	Bajo coste
Audio	<u>PBX</u>	(Des) Conexión en caliente
Video comprimido	<u>POTS</u>	Fácil uso
500 Kb/s - 10 Mb/s	Audio	Múltiples dispositivos Garantizado el ancho de banda Garantizada la latencia
<b>VELOCIDAD ALTA</b>		
Video	Video	Gran ancho de banda
Unidades de almacenamiento	Discos duros y grabadoras	Latencia garantizada
Video Comprimido		Fácil uso
25-500 Mb/s		

Básicamente, USB 2.0 incluye todo lo que ofrece USB 1.1 y añade el modo de alta velocidad. USB 2.0 también usa el mismo tipo de cables y conectores para conectar los dispositivos de alta velocidad, sin embargo los hubs USB clásicos ralentizarán los dispositivos USB 2.0. Otro requisito es que es

necesario un controlador de host para USB 2.0 si se quiere tener disponible la conexión de alta velocidad con un dispositivo de este tipo.

Los hubs USB 2.0 tienen ahora mucho más trabajo que hacer que en el USB 1.1 ya que necesitan manejar todo el tráfico de tres tipos de dispositivos con velocidades distintas. Conectando un dispositivo USB 1.1 en uno USB 2.0 funcionaría bien, pero no lo haría si se hiciera al revés, además de ralentizarse a 12 Mbps, posiblemente, el sistema operativo avisaría de su mal uso.

La forma de distinguir si tu ordenador tiene USB de alta velocidad es mediante los logotipos registrados por el grupo promotor del USB (el [USB-IF](#)), que como se puede ver más arriba está etiquetado en rojo con Hi-Speed. Dicho logotipo sólo lo pueden usar las empresas desarrolladas de hardware que hayan pasado los tests del USB-IF

Cabe destacar que el USB 2.0 es simplemente una extensión y nunca llegará a reemplazar completamente al 1.1 ya que hay productos como teclados genéricos, ratones, joysticks o altavoces que no requieren la gran velocidad que ofrece la tecnología USB 2.0. Sólo dispositivos de alta velocidad como webcams o sistemas de alta capacidad necesitarán el máximo de velocidad, aunque los ordenadores los fabriquen sólo con puertos USB 2.0.

## **Capítulo 3 Características del sistema.**

### **Introducción**

En este capítulo se realiza una comparación entre los puertos USB y RS232, y un estudio detallado sobre la comunicación y transmisión de datos del simulador C10 en cuanto al puerto que utiliza este para la comunicación, la estructura del protocolo que se emplea para la transferencia en la parte del software, la capacidad de envío y recepción, longitud del cable, conectores que utiliza para la conexión, velocidad de transmisión, etc.

### **3.1 Simulador C10**

El simulador C10, es un sistema que intenta simular la acción de lanzar un cohete desde un lanzacohetes real adaptado a un simulador que representa el cielo con los posibles blancos de aviones

a tirar. Este simulador permite entrenar al artillero en condiciones totalmente realistas, donde el practicante pierde noción de que se trata de una simulación, ya que se utilizan blancos con desplazamiento y aparición repentina en distintos lugares geográficos.

En la siguiente figura se muestra un lanzacohetes real:



### 3.2 Protocolo de Comunicación de transferencia de información del Simulador C10( PTDC10).

El protocolo que emplea el Simulador C10 para la transmisión de la información esta definido por los comandos que más adelante se mostrarán y sus respectivas operaciones, que son implementados en la parte de software, y leídos posteriormente por los conectores que emplea el puerto RS232 para la comunicación entre la computadora y la Flashlite 186 . Esta última a su vez procesa la información y envía una respuesta, siendo un dispositivo que permite adaptarse a todas las señales entrantes de forma que puede registrar varios sensores al mismo tiempo convirtiéndolos en una sola salida digital. Desde un punto de vista, se puede observar como una caja negra en la cual por un lado le entran varias señales analógica y por otro sale una sola señal digital. A continuación se mostrarán las operaciones y comandos que conforman el PTDC10:

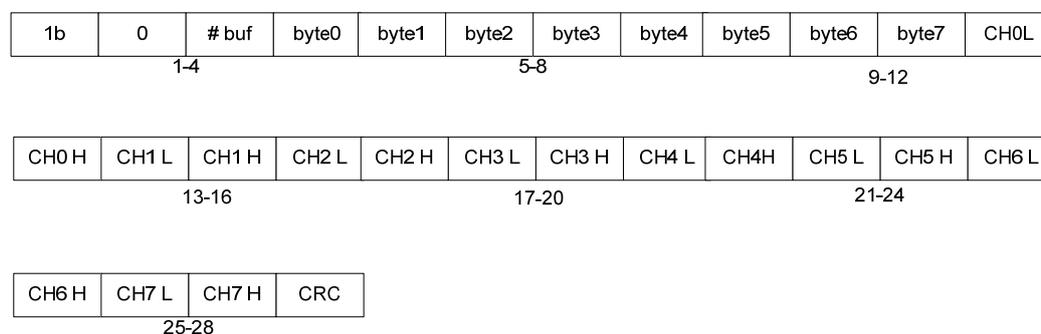
### 3.3 Bloques de Datos. Comandos que generan al PTDC10:

Nombre Comando	No.	Bytes	Observaciones
----------------	-----	-------	---------------

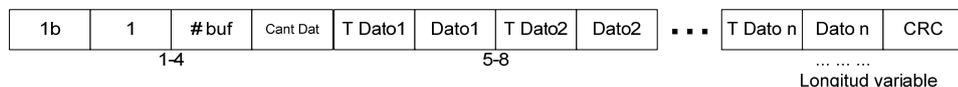
	Block		
Enviar todo los Datos de Entrada	0	24	Transmite los 64 estados de los interruptores y conmutadores acoplados a la tarjeta de entrada.
Cambio de los datos de Entrada	1	Variable	Transmite <b>N</b> cambios de los interruptores que cambiaron de estado
Repetir Bloque	2	4	Solicita que repitan el último bloque recibido
Parar Transmisión	3	4	Detiene la Transmisión de la FlashLite
Solicitud de los datos de entrada	4	4	Solicita a la FlashLite que comience la Tx con un bloque 0 y que se mantenga transmitiendo los cambios cada vez que los mismos ocurran.
Comenzar transmisión de los cambios	5	4	Hasta que no se de este comando no se permite la transmisión de los cambios de entrada que se detectan en el SimC10
Reservado	6		
Reservado	7		
Reservado	8		
Reservado	9		
Todos los Datos de Salida	10	12	Contiene los 64 estados de los bombillos y LEDs acoplados a la tarjeta de Salida para que sean actualizados sus valores.
Actualizar los Datos de Salidas	11	Variable	Contiene los <b>N</b> estados de los bombillos y LEDs que se deben cambiar.
Actualizar los Datos de todos los Instrumentos	12	7	Contiene los valores que deben indicar cada uno de los instrumentos

### 3.4 Formato de los Bloques de Datos

BLOQUE 0: Enviar todos los datos de entrada



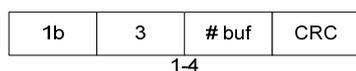
BLOQUE 1: Cambios de los datos de entrada



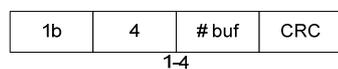
En dependencia del valor de T Dato x el dato que viene a continuación tiene una longitud de 1 ó 2 byte como se muestra en la siguiente tabla.

T Dato x	Longitud Dato x	Longitud Total	Tipo de Dato
<10	1	2	Entrada
>10	2	3	Canal Analógico

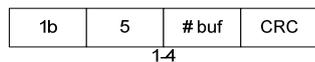
BLOQUE 3: Parar la Transmisión



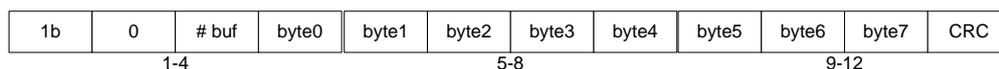
BLOQUE 4: Solicitud de los datos de entrada



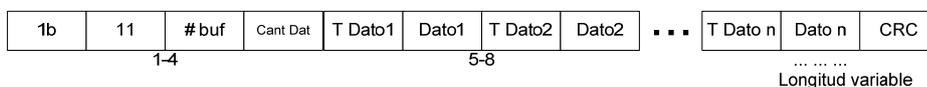
BLOQUE 5: Comenzar transmisión de los cambios



BLOQUE 10: Todos los Datos de Salida



## BLOQUE 11: Actualizar los Datos de Salida



## BLOQUE 12: Actualizar los Datos de todos los Instrumentos



### 3.5 Operaciones y señales de entrada/salida en la parte del software. Capacidad de envío y recepción de datos.

Las señales de entrada que controla el Simulador C10 son interruptores de 1byte cada uno y 8 canales de conversor análogo digital de 2 byte cada uno. El cliente es quien solicita los datos de entrada y determina los datos de salida los cuales son: bombillas de 1 byte cada uno y existen 4 instrumentos de los cuales los datos para cambiar el valor de los instrumentos su longitud es de 1 nibble (4 bits) por tanto con tres byte se puede mandar los datos para la actualización de los 4 instrumentos. Y la flashlite186 es quien procesa todas las acciones. Las operaciones básicas que debe contemplar el protocolo para el intercambio de información en la parte del software son las siguientes:

- Solicitar todos los datos de entrada (interruptores y canales analógico)
- Solicitar transmisión de cambios de los datos de entrada.
- Parar la transmisión de cambios.
- Actualizar todos los datos de salida ( bombillos e instrumentos)
- Actualizar datos de salida específico (puede ser bombillos o instrumentos)

Cada una de estas operaciones genera un comando (paquete o bloque) y su respuesta correspondiente, las cuales son implementadas en la parte del software.

### 3.6 Objeto de estudio: Transferencia de información

El PTDC10 realiza la transmisión asincrónica de forma secuencial, es decir, bit por bit. El flujo de la información que permite el puerto serie RS232 por el cual se establece la comunicación de la computadora con la FlashLite 186 es en una sola línea de comunicación, es decir es una conexión por la cual se envían datos a través de un solo conducto. Para el establecimiento de la transmisión mediante software se diseñaron comandos con sus respectivas operaciones adaptadas a la parte hardware por donde transita la información mediante el puerto RS232. Las operaciones o métodos que se implementan en la parte de software son las que generan a cada uno de los comandos diseñados en el protocolo. De cada comando en dependencia de la operación que se realice, el transmisor será el encargado de codificar esta información cogiendo un pequeño flujo de datos y codificándolo en cuanto a: bits de datos, bit de inicio, bit de parada y bit de paridad. Cada bit es enviado al receptor de forma secuencial como se dijo anteriormente, bit a bit. En la parte del hardware la información transita a través de los pines del conector DB 9. Cada uno de estos pines tiene su propia función que son las señales que se emplean para la transmisión de la información. En la comunicación serie RS232 están las siguientes señales:

Tx: transmisión de datos

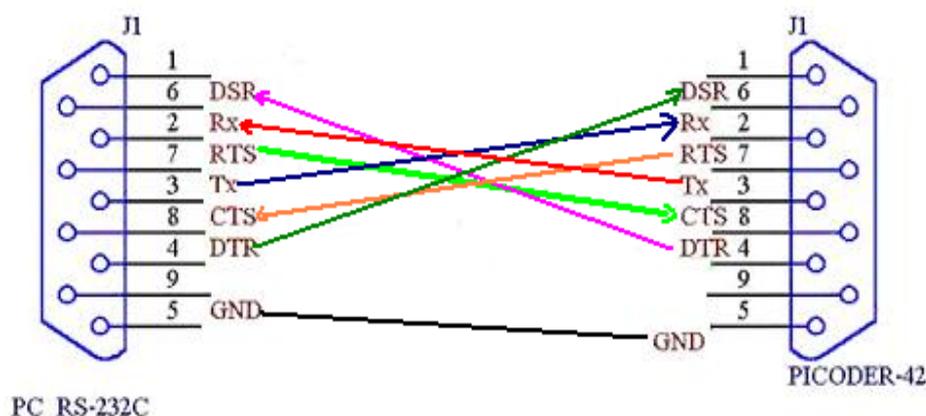
Rx: Recepción de datos

RTS: Request to send (solicitud de transmisión de datos)

DSR: Data send ready (listo para enviar datos)

CTS: Clear to send

DTR: Data terminal ready (listo para recibir datos)



Este diagrama muestra como las señales se entrecruzan para el intercambio de información necesaria antes de inicializar la transmisión (recuerdan que esta es una transmisión asincrónica). El que va a transmitir los datos a través de la línea RTS solicita permiso de transmisión, que llega a la otra parte por CTS. La otra parte cuando está disponible para recibir le envía la señal a través de DTR le dice que está lista para recibir los datos, que llega por DSR entrada en la otra parte. Después de esto a través de la salida TX se transmite los datos que llegan a la otra parte por Rx.

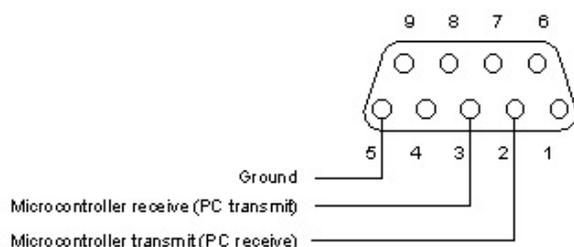
Ambos dispositivos (PC y Flashlite 186) cuando se establece la conexión deben leer el mismo protocolo para dar inicio a la transmisión. En el epígrafe 2.4 se abordará más información respecto al tema.

### 3.7 Dispositivos donde se establece la conexión:

Una Computadora que funciona como Terminal(DTE) que es el dispositivo que solicita la conexión y una Flash lite 186 que funciona como equipo de datos(DCE) es el que ofrece el soporte para la conexión, por tanto tiene acceso a la portadora, a la sincronización. Hay un equipo que se comporta como un DTE y otro como DCE. Los pines de salida de un conector DTE serán de entrada en DCE y viceversa.

### 3.8 Conectores, cables y puerto serie empleado para el establecimiento de la conexión.

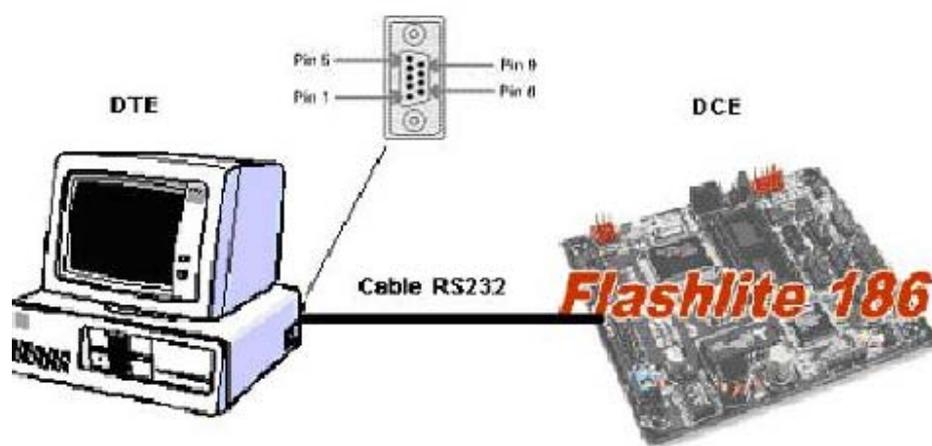
La conexión es realizada mediante puerto serie RS-232 con el empleo de un cable de 2 metros RS232, y conectores de tipo DB9.



### 3.9 Conexión y configuración para el establecimiento de la transmisión.

Para que ocurra la transmisión, mediante software se diseñaron comandos con sus respectivas operaciones. En la parte física para la transmisión primeramente se establece una conexión entre los

dos dispositivos mediante el puerto RS232, y se debe configura ambas dispositivos ya que los dos deben leer el mismo protocolo para la transferencia de la información, siendo los parámetros a seguir: la velocidad de transmisión del puerto, control de flujo y bit transmitidos. Después que se realiza la conexión y la configuración por medio de la persona que la establece; ya que es quien debe decidir y configurar ambas partes antes de iniciar la transmisión, se realiza la transmisión de la información.



### 3.10 Velocidad de transmisión:

El PTDC10 tiene una velocidad de transmisión fija de 38400 baudios o bit por segundos, siendo el rango de velocidad entre 75 hasta 128000 baudios para la transmisión en puerto Serie RS-232 que es el mismo que emplea Simulador C10 para la conexión.

### 3.11 Control de Flujo:

Con el flujo de control, el dispositivo puede parar el flujo de datos de la computadora, si el buffer del dispositivo está lleno o procesando aún la información. Estas señales RTS y CTS se usan debido a hay casos en que la velocidad de transmisión de la computadora puede ser mayor que la del dispositivo, es decir la computadora realiza las acciones más rápido que el dispositivo. Por tanto, el

dispositivo puede parar el flujo de datos de la computadora desactivando la señal CTS, cuando este lista para que le envíen datos la activa de nuevo. El control de flujo procede de la siguiente forma la computadora le envía al dispositivo una señal RTS y el dispositivo activa el CTS si no tiene el buffer lleno informando que esta libre para envío.

### 3.12 Flashlite186

Para solamente **\$69** por unidad, la sola computadora del tablero de Flashlite 186 ofrece una solución rentable para la adquisición de muchos datos, las comunicaciones y los usos industriales del control. Construido alrededor de un procesador compatible 33MHz 186, un regulador de 44 líneas digitales configurables de entrada-salida, la COPITA 512K y la memoria de destello 512K, dos puertos seriales y una consola/elimina errores del puerto serial. Las características adicionales incluye el regulador de voltaje de 7-34V (potencia cc 5V o 7-34V), dos 16 contadores de tiempo del pedacito, un contador de tiempo del perro guardián, puerto de la extensión de la entrada-salida, así como un zócalo para ampliar memoria permanente usando los M-Sistemas DiskOnChip.

**El kit del desarrollo de Flashlite 186 (99-0080)** incluye la solos computadora del tablero de Flashlite 186 guía, cable de la consola, kit del conector, adaptador de 110 VAC, disposición, diagrama esquemático y CD con el compilador de Borland C/C++ V4.52, la documentación y bibliotecas seriales del conductor. (kit 99-0081 del desarrollo 220Vin disponible.). El kit del desarrollo de Flashlite 186 incluye el tablero y el software para **SOLAMENTE \$99**.

### 3.13 Problemáticas que hay actualmente en el simulador

El Simulador C10 presenta fallas en el sistema debido a que la transmisión de la información es muy lenta, y existe la posibilidad de que se pierdan datos cuando se realice la misma. El PTDC10 emplea una velocidad de transmisión de 38400baudios que es muy reducida. Esto influye cuando se envíe un comando que emplee una capacidad grande se retarde la velocidad de transmisión. También el medio de comunicación (puerto RS232) que emplea este simulador tiene la desventaja de ser susceptible a perder datos en el momento de la transmisión. Por tal motivo se optó una variante para solucionar estos problemas que presenta el simulador C10.

### **3.14 Variantes de solución del problema:**

Existen dos variantes por las que se puede resolver el problema de transmisión de la información que presenta el simulador C10:

1. Debido a que en el diseño del PTDC10 se emplean algunos comandos con estructura muy larga. Podría ser una vía desfragmentar cada uno de estos comandos, mediante estudios realizados a partir del diseño o confección de la estructura de comandos de otros protocolos empleados para la transmisión de la información por puerto serie.
2. Otra alternativa sería, a través de estudios realizados sobre puerto serie. Obtener un puerto mas eficiente que sustituya al estándar RS232 que es el que emplea el Simulador C10 para la transmisión de la información y adaptarlo al sistema.

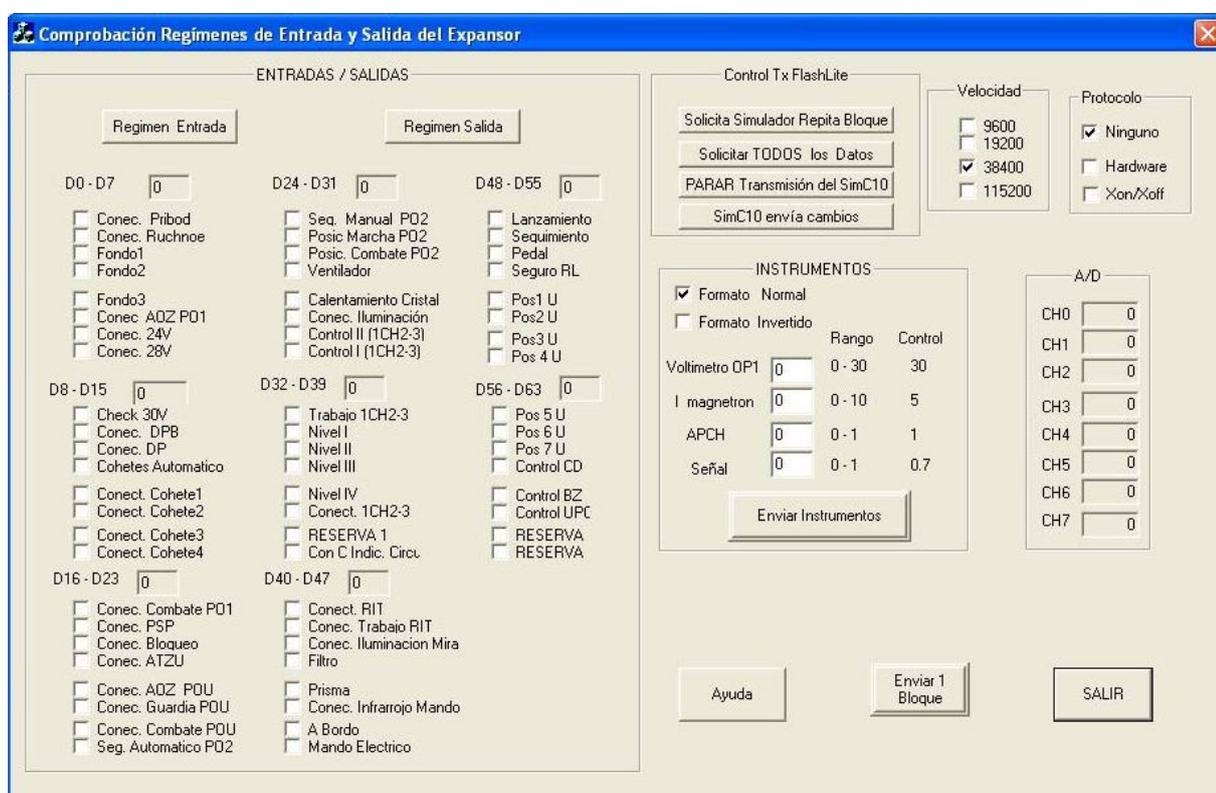
Para la solución de este problema, se escogió la segunda opción, ya que aumentando la velocidad de transmisión de la información, no perjudicaría en nada que los comandos sean largos, pues la transmisión se realizaría de forma rápida. El puerto que se determinó para sustituir por el RS232 es el USB que además que produce una velocidad de transmisión 10 veces mas rápida que el RS232 presenta muchas ventajas como en el capitulo anterior se muestran.

### **3.15 Objetivos estratégicos de la organización (UCI).**

Los objetivos estratégicos de la organización tratan sobre: la producción de software para ser empleados en cualquier esfera de la sociedad, ampliar el conocimiento sobre simuladores que representan eventos de la vida real y que sirvan de ayuda a la preparación de los clientes finales antes de ejercerlo realmente, el establecimiento de una propuesta para optimizar el sistema de transmisión de la información en un simulador para que sirva como manual de ayuda para investigaciones futuras y también para la confección de un simulador que responda a las necesidades que actualmente se imponen.

### **3.16 Información que se maneja.**

La comunicación establecida para la transmisión de la información es entre una computadora y una Flashlite186 como anteriormente se había dicho. La computadora contiene una aplicación llamada “Comprobación Regímenes de entrada y salida del expansor” donde se emplean opciones como datos de entrada que seria los datos de convertidores e interruptores, los datos de salida que serian instrumentos y canales analógicos, emplea un control de transmisión en la flashlite, el envío de instrumentos, el tipo de protocolo de control de flujo, y la velocidad de transmisión. Esta información al ser enviada es procesada por la Flashlite 186 quien ejecuta las acciones correspondientes en el simulador C10. A continuación se muestra la aplicación antes mencionada.



### 3.17 Comparación entre RS-232 y USB:

El puerto serie RS232 presenta desventajas respecto a su velocidad de transmisión que es muy reducida, los conectores son mas costosos que los de USB, es susceptible a perder datos y no es compatible con muchas plataformas nuevas. En cambio el puerto USB emplea una velocidad de

transmisión 10 veces mayor que la que emplea RS232, es empleado en todas las plataformas nuevas como mejor puerto de comunicación, los conectores son menos costosos, es plug and play, hot pluggable, etc. El puerto USB presenta características ventajosas que podrían ser una solución a los problemas de transmisión de información que presenta el Simulador C10. Los datos que se muestran en la tabla de comparación demuestran que el Puerto USB es más eficiente que el puerto RS232. [Ir a](#)

[Anexo 1](#)

### **3.18 Se puede modificar el método de transmisión de la información del PTDC10 empleando puerto USB?**

¡Se puede utilizar este método!

Una característica que se tuvo en cuenta para tomar esta decisión fue la compatibilidad con todas las plataformas, pues aunque muchos dispositivos viejos no presentan entrada USB, actualmente todos los fabricantes emplean este estándar debido a las múltiples ventajas que presentan como en el capítulo anterior se muestran. Por lo que sustituyendo USB por RS232 no influiría en nada la compatibilidad mediante el software o hardware.

### **3.19 Breve descripción sobre la propuesta de sistema.**

Se quiere establecer una conexión por puerto USB entre la computadora y FlashLite 186, aunque esta última no tiene entrada por puerto serie USB. Por tanto obteniendo una flashlite que tenga puertos de entrada USB se resolvería el problema. Con esto, y estableciendo una conexión con cable USB y conectores USB, y adaptando la parte del software para que pueda leer la transmisión de información para puerto serie USB, se resolvería el problema que presenta el Simulador C10 referente a la velocidad de transmisión, y la pérdida de información. El sistema que se propone en el próximo capítulo servirá de ayuda para que los usuarios finales a la hora de ejecutar el sistema y utilizarlo, tengan a su disposición un sistema eficiente, que ejecute las instrucciones rápidamente y sin riesgo a fallo a la hora de introducir los datos, y devolver la respuesta correspondiente.

## Capítulo 4 Propuesta del Sistema con conexión USB.

### Introducción

En este capítulo se realiza un estudio sobre una Flashlite que tenga entrada por puerto USB para posteriormente realizar una comparación con la Flashlite 186 que es la emplea el Simulador C10, y con esta información determinar la flashlite apropiada a utilizar para la conexión con la computadora. También se realiza una investigación sobre como leer los comandos (datos) por puerto USB, los tipos de conectores y cables a emplear para la conexión. Y se muestra una propuesta de como debe quedar el sistema.

### 4.1 Puerto serie propuesto para la conexión.

El USB será empleado como anteriormente se ha dicho para la conexión establecida entre la computadora y la Flashlite nueva. De las versiones que se generan de este puerto se puede escoger la más óptima de estas para la conexión que sería la versión USB 2.0 ya que es la que emplea una mayor tasa de transferencia de 480 Mbit/s, a diferencia de la versión USB 1.1 que es de 1.5 y 12 Mbit/s. El USB 2.0 es una extensión del USB 1.1, utiliza los mismos cables y conectores y es completamente compatible con USB 1.1. También se puede emplear para la conexión entre la PC y la Flashlite la versión 1.1, ya que de todas formas, empleando esta velocidad de transmisión sería aun superior a la que emplea RS232. Solo hay que tener en cuenta que conectando un dispositivo USB 1.1 en uno USB 2.0 funcionaría bien, pero no lo haría si se hiciera al revés, además de ralentizarse a 12 Mbps, posiblemente, el sistema operativo avisaría de su mal uso.

### 4.2 Conectores y cables a emplear para la conexión:

Los conectores empleados para el establecimiento de la conexión son: el de Tipo A (puerto del ordenador) y el del Tipo B (puerto del dispositivo).



**USB A (Huésped)**

El conector USB Tipo A tiene forma rectangular y es el conector que se coloca en el CPU o el Hub USB.



**USB B (Dispositivo)**

El conector USB Tipo B es más cuadrado y es el conector que se coloca directamente en el dispositivo (por ejemplo, en la cámara digital o en la impresora multifuncional).

USB 1.1 contempla en sus especificaciones los conectores: Tipo A y Tipo B

USB 2.0 contempla en sus especificaciones los conectores: Tipo A y Tipo B, y además el Mini-B.



**Mini-B**

Este conector fue introducido para permitirle a los consumidores de teléfonos celulares y PDA's gozar de las bondades del USB.

### 4.3 Flashlite con puerto USB

**OmiFlash** es la marca registrada para una [sola computadora](#) pequeña del [tablero](#) (SBC) basada en la viruta de procesador de la [lógica](#) EP9302 [ARM9 del cirro](#) [Linux encajado](#) funcionamiento. El regulador ofrece 16MB de la [memoria de destello](#) y 32MB de [SDRAM](#).



Fig.Sola computadora del tablero de OmniFlash

El USB, Ethernet, y las capacidades audio pueden ahora ser una parte de un diseño con el empleo OmniFlash. Su sistema operativo tamaño pequeño y de gran alcance hace ideal para la complejidad de crecimiento de usos de hoy en día.

En menos de 16 pulgadas cuadradas, el OmniFlash esta constituido por 2 puertos seriales, 2 puertos USB, In/Out audio, potencia cc 5V, un procesador ARM a 200MHz, 32MB de RAM ,16 líneas digitales de la entrada-salida, un reloj/el calendario, e incluso capacidades audio. Becomess de la extensión fáciles para cualquier otra cosa que puedes necesitar a través del interfaz del USB. Para manejar todas estas nuevas características, el OmniFlash viene equipado de un procesador del BRAZO 200Mhz, 32 megabytes de ESPOLÓN, y el regulador ofrece 16MB de la [memoria de destello](#). Contiene el regulador más rápido.

El microcontrolador single-board de OmniFlash viene cargado con el sistema operativo de Linux y todas las utilidades que necesitas comenzar el desarrollo. Nuestro kit barato del desarrollo contiene un CD bootable del desarrollo, eliminando los molestias de conseguir un ambiente de compilación cruzado instalado. El código de fuente para un número de programas de la demostración es además incluido en el CD. Gracias a los puertos USB la expansión del equipo es realmente sencilla.

**El kit del desarrollo de OmniFlash (99-0140)** incluye un regulador de OmniFlash, un cable de programación, un kit del conector, un adaptador de 110 VAC, una guía de la disposición, un CD esquemático y Bootable con GCC C/C++Compiler, programas de muestra y una documentación. El kit incluye el tablero y el software para **SOLAMENTE \$199**.

#### 4.4 Comparación entre Flashlite186 y Omniflash:

La Omniflash es más costosa que la Flashlite 186. Aunque esta sea una desventaja, la Omniflash presenta propiedades muy ventajosas, como son, el empleo de un regulador de voltaje y un procesador de información mucho más rápido que el emplea Flashlite 186, 2 puertos seriales USB los cuales hacen que la expansión del equipo sea realmente sencilla además de que aumenta la velocidad de transmisión de la información. Con esta información se puede concluir que indudablemente la Omiflash seria la tarjeta de procesamiento ideal para sustituirla por la Flashlite 186, ya que permite que el procesamiento de la información se realice de forma rápida en el Simulador C10. [Ir a Anexo 2](#)

#### **4.5 Transmisión de la información por USB**

##### **➤ Transmisión de la información, parte física:**

El puerto USB al que hace mención es el conector de entrada a un bus de tipo serie en el que es posible conectar múltiples dispositivos, tanto en frío como en caliente, siendo éstos por regla general reconocidos por de inmediato por sistemas operativos como Windows. Cuando el sistema detecta que se ha conectado un dispositivo al bus USB, lo primero que hace es solicitar una identificación para saber de qué dispositivo se trata. Si existe un controlador ya instalado que le permita comunicarse con él, lo utilizará directamente e, incluso, pondrá en marcha la aplicación que corresponda para acceder a dicho dispositivo. Si, por el contrario, no hay un controlador pre instalado en el sistema, aparecerá el típico cuadro de diálogo para solicitar un controlador adecuado. La comunicación del ordenador con los dispositivos USB es, por tanto, tarea de los controladores de dispositivo. Estos utilizarán un conjunto de funciones que forman parte de la especificación conocida como Universal Serial Bus Specification, Revisión 1.0 (proporciona los detalles técnicos para entender requisitos del USB y para diseñar productos compatibles del USB.), accediendo al puerto USB a bajo nivel. El controlador, a su vez, ofrecerá a las aplicaciones un API de alto nivel que les permita usar el dispositivo sin tener que preocuparse de los detalles de la comunicación USB. Si está usando una cámara fotográfica digital USB, por ejemplo, las aplicaciones accederían a ella mediante un conjunto de funciones en las que trabajaría sobre imágenes, y no sobre detalles de comunicación y flujos de datos. Si lo que quiere es comunicarse con un dispositivo en particular, lo mejor es que use el API de alto nivel correspondiente. Si desea acceder al bus USB a bajo nivel, deberá programar un controlador de dispositivo. Precisaré, por una parte, el DDK (Driver Development Kit) que corresponda a su versión de Windows. Estos paquetes de desarrollo de controladores son gratuitos y puede obtenerlos de [www.microsoft.com/ddk](http://www.microsoft.com/ddk).

En segundo lugar precisará toda la información sobre el bus USB que pueda encontrar, principalmente la especificación en la que se define el funcionamiento del bus y todas sus funciones. En [www.usg.orb/developers](http://www.usg.orb/developers) encontrará todos estos datos, además de una FAQ y algunas herramientas útiles.

➤ **Transmisión de la información mediante software:**

La transmisión de información es asincrónica. Las operaciones entre el dispositivo USB y el controlador USB se realizan a través de una cola de transferencia. Mediante software se construye una cola de operaciones, las cuales son llevadas a cabo antes de que el controlador inicie una operación con la OmniFlash. El controlador inspecciona la cola de transferencia o cola de peticiones, y esta se ejecuta por orden de llegada de las operaciones. La petición de transferencia es la de enviar un comando USB a la Omniflash. La Información útil que debe conocer el controlador es:

- La dirección del dispositivo
- Situación del buffer
- Tipo de transferencia de información (sincrona, asincrónica, etc).
- Máximo número de bytes de datos.

Cada petición de la cola de transferencia es ejecutada por el controlador en dependencia del orden de llegada. Cuando se ejecuta una petición se envían hasta tres paquetes que determinan la dirección del dispositivo donde se enviara la información, el comando que se transmitirá y el estado de la transferencia. Cada paquete describe el tipo y la dirección de la transacción, dirección el dispositivo USB y el “end point number” que identifica al dispositivo. La OmniFlash se encargará de descifrar los campos que componen el paquete.

La transacción se realiza enviando los datos desde la Omniflash al controlador, o viceversa, dependiendo del tipo de transmisión. Para finalizar la transacción, el receptor de datos (la omniflash o el controlador) enviará un tercer paquete que confirmará el éxito de la comunicación.

Una transacción esta contenida en estas dos categorías

- Transferencia de control
- Transferencia de datos

La transferencia de control es empleada para la configuración de un Omniflash. Sin embargo la transferencia de datos sirve para el envío de un flujo de datos, esta se puede realizar tanto desde el controlador como desde la Omniflash. Mientras que la transferencia de control siempre se realiza desde el controlador al dispositivo (Omniflash).

Un ejemplo podría ser el envío de un comando que configura la Omniflash, realizándose una transferencia de control. A continuación se envían los datos de las operaciones a realizar como por ejemplo "Enviar instrumentos".

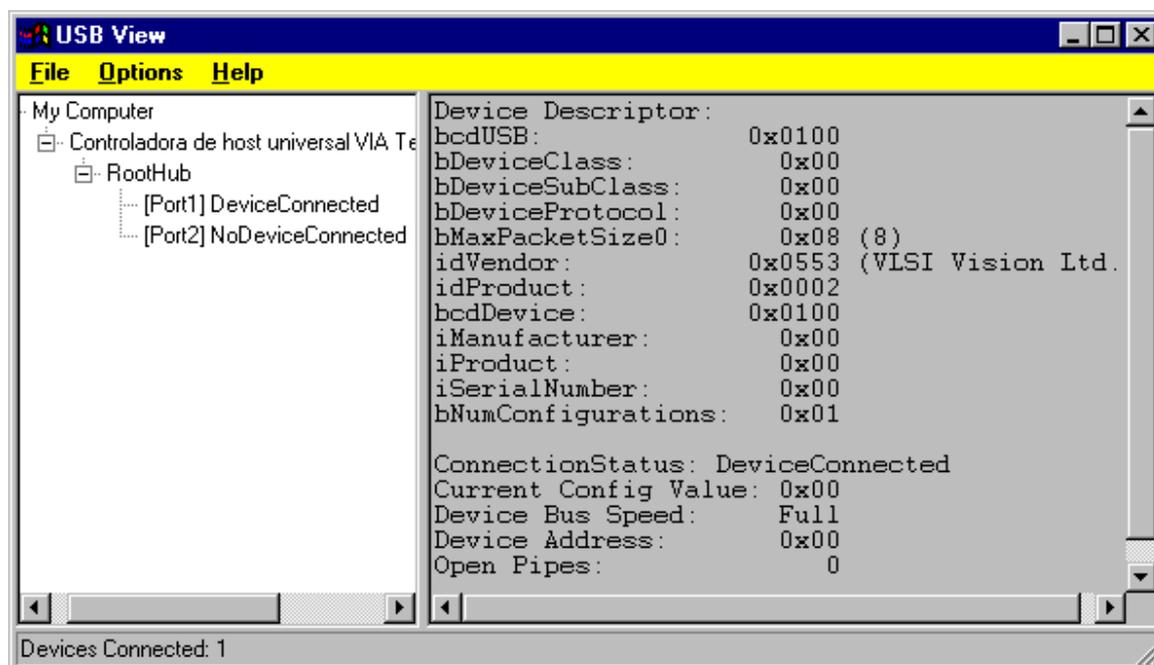
El controlador ejecuta cada petición de la cola de transferencia en dependencia del orden de llegada. Cuando se ejecuta una petición se envían hasta tres paquetes los cuales determinarán la dirección del dispositivo donde se enviará la información, el comando a enviar y el estado de cada transferencia.

#### **4.6 HIDmaker FS - USB the *Easy Way* : Código fuente para la Transmisión por USB**

Se encuentra en [http://www.tracesystemsinc.com/usb\\_tools.ivnu](http://www.tracesystemsinc.com/usb_tools.ivnu) con el precio de \$599.95 ! Todo lo que usted tiene que hacer es describir los datos que desea enviar entre el dispositivo periférico y el anfitrión de la PC, usando el bloc de octetos gráfico llamado el diseñador visual de los datos. Entonces, HIDmaker FS genera el código de fuente emparejado para el periférico y la PC en el mismo tiempo, en su opción de lenguajes de programación. Usted compila ese código de fuente usando los compiladores que usted posee ya, y en minutos, usted tiene el anfitrión de la PC y periférico de PIC que habla el uno al otro, intercambiando los datos que usted especificó.

#### **4.7 Identificación y diagnóstico**

Windows dispone de un programa específico para ver los puertos USB reconocidos y sus parámetros de configuración, es el programa **Usbview.exe**, que puede encontrarse en el CD de instalación de Windows 98, en el directorio Tools\Reskit\Diagnose



En la figura se muestra el resultado de conectar una cámara de videoconferencia en el puerto 1.

Por su parte, el foro de desarrolladores USB (USB-IF), ha publicado diversas herramientas para comprobación de las características de los dispositivos USB, entre las que se encuentran las siguientes:

**USBCV** ("USB Command Verifier"). Es una herramienta de diagnóstico y verificación para comprobar que dispositivos USB de alta y baja velocidad se ajustan a las normas.

**USB Check**. Esta herramienta permite verificar dos enlaces; uno para comprobar dispositivos de alta velocidad funcionando a alta velocidad, y otro para comprobar dispositivos de velocidad completa y baja; también dispositivos de alta velocidad funcionando a velocidad completa.

**USBHTT** ("USB2 Hub Transaction Translator Test Suite"). Es una herramienta de verificación para concentradores USB 2.0.

**USBHSET** ("USB High Speed Electrical Test Tool Kit"). Este sistema de prueba contiene software y procedimientos diseñados para verificar diversos parámetros eléctricos, incluyendo la calidad de señal

en dispositivos USB de alta velocidad. Incluye también procedimientos detallados para comprobación de **controladores host**, **concentradores** y **funciones** USB de alta velocidad. Este software también permite comprobar la calidad de señal en dispositivos de velocidad completa y baja, así como verificaciones de suministro energético.

Estas herramientas están disponibles para su descarga en:

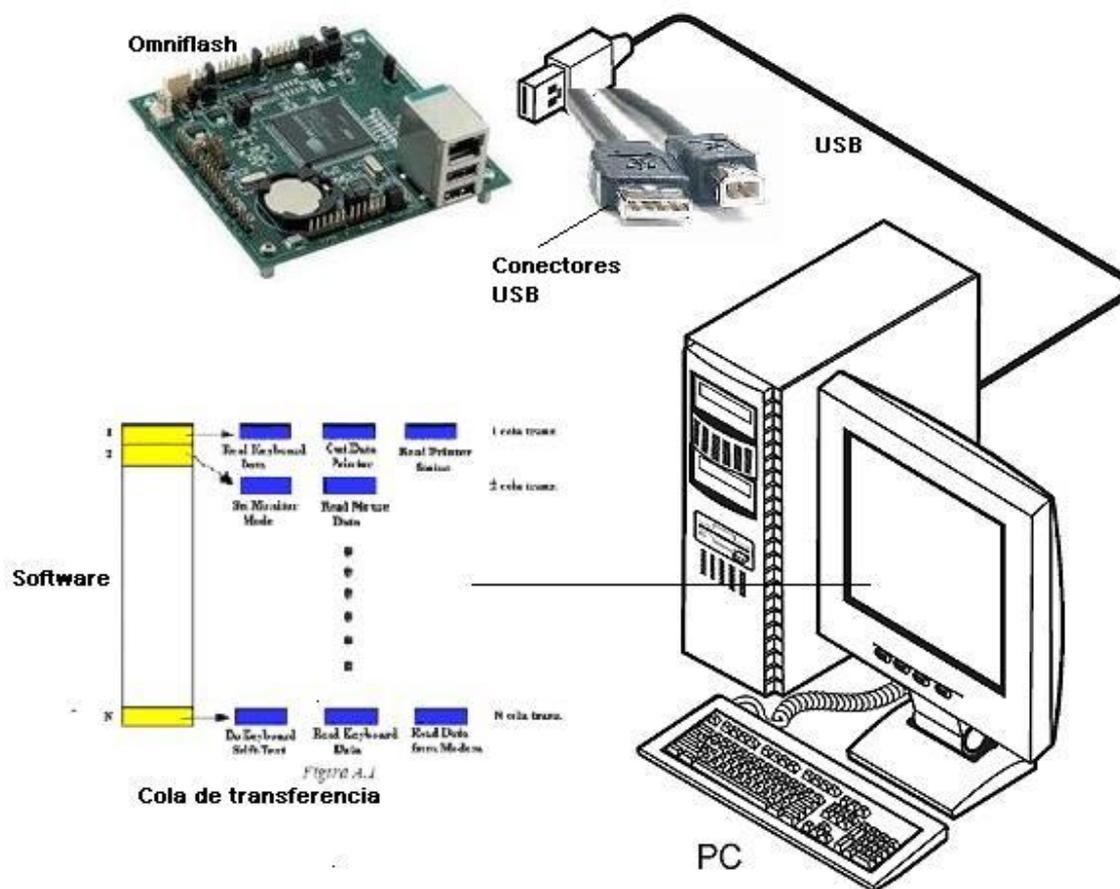
[www.usb.org/developers/tools.html#usbhset](http://www.usb.org/developers/tools.html#usbhset)

**SSTD** ("Single Step Transaction Debugger"). Herramienta de depuración y desarrollo de bajo nivel que permite enviar transacciones simples a un dispositivo de alta velocidad.

<http://www.usb.org/developers/usb20/>

#### **4.8 Descripción general de la propuesta de sistema.**

Se realiza la conexión por puerto USB entre la PC y la Omniflash, con conectores tipo A en la PC y tipo B en la Omni Flash, y cable de 5 metros. El controlador es el encargado de realizar la configuración del periférico (Omniflash) y es quien realiza la transmisión de la información entre la PC y la Omniflash mediante una cola de transferencia. EL método del puerto USB para transferir datos y configurar el sistema es muy sencillo, ya que el host controlador de la PC es quien configura el dispositivo, y el protocolo para la transmisión de la información es realizado mediante un software a través de una cola de transferencia la cual se abordó en el epígrafe 2.2. En la figura siguiente se mostrará el modelo de comunicación por USB:



## Conclusiones

Con la investigación realizada se logró cumplir el objetivo propuesto, ya que mediante los conocimientos obtenidos se llegó a la conclusión que empleando un puerto serie que en este caso es el USB, que tuviera una velocidad de transmisión mucho mayor que la que produce el puerto RS232, se resolverían los problemas de transmisión de información del simulador C10; ya que cuando se transmite bit a bit hacia el receptor la información llega a este mucho mas rápida y sin la mínima probabilidad de que el paquete enviado llegue al receptor con pérdida de información . Para el establecimiento de la propuesta del sistema, se tuvieron en cuenta las grandes ventajas que presenta este tipo de puerto serie, las cuales atribuyen propiedades muy valiosas en el establecimiento de conexión entre la PC y la Omniflash., se escogieron los medios (conectores USB, cable USB, dispositivo USB (Omniflash) y la PC) propicios para la conexión, y se desarrolló una explicación sobre el funcionamiento de la transmisión de la información por puerto serie USB mediante el hardware y mediante el software que empleará el Simulador C10 para transmisión de datos de forma eficiente

## **Recomendaciones**

Realizar el diseño del Simulador C10 poniendo en práctica la propuesta establecida, y optimizar en trabajos futuros el funcionamiento del simulador debido a que con el aparecimiento de nuevas tecnologías pueden surgir métodos más eficientes para la transferencia de información en el Simulador C10.

## Referencias bibliográficas

[1] mi sitio...

[2] la segunda

[20] la veinte

**¡Error! No se encuentra el origen de la referencia.**

## Anexos

### Anexo 1. Tabla de comparación entre RS232 y USB

	<b>Puerto RS-232</b>	<b>Puerto USB</b>
<b>#Máximo de dispositivos conectados</b>	1 dispositivo	127 dispositivos
<b>Cambio caliente (agregar o quitar dispositivos sin reiniciar el ordenador)</b>	No	Si
<b>Longitud del cable entre dispositivos</b>	15 metros	5 metros
<b>Velocidad de transferencia de datos</b>	75 a 128000 baudios	1.5Mbps (teclados, ratones) y 12Mbps CD Rom, altavoces) y hasta 480Mbit/s para audio y videos.
<b>Tipo de configuración</b>	No automática.	La PC es quien lo configura (automática)
<b>Conectores</b>	2 tipos de conectores: DB-9 y DB-25.	2 conectores: Tipo A y Tipo B
<b>Cantidad de hilos cada cable</b>	Unos pueden tener hasta 9 hilos y otros hasta 25 hilos.	Tiene 4 hilos: 2 para datos, y dos para alimentación.
<b>Soporte</b>	El RS232 es compatible con la mayoría de las plataformas viejas.	Responde a todas las necesidades de los usuarios con el mismo hardware para todas las plataformas. El estándar USB está actualmente soportado por más de 250 compañías líderes en la industria de los ordenadores, telecomunicaciones y software. Intel es miembro del USB Implementers Forum. Total en Windows 98 y por medio de drivers en Windows 95 y Windows NT

Autores: Nayla Yamila Sire Vila  
Arlin Maury Cruz Perez

**¡Error! No se encuentra el origen de la referencia.**

<b>Versiones</b>	RS-232C	USB1.0,USB1.1,USB2.0
<b>Caraterísticas de las versiones</b>	Las diferencias entre ambas versiones RS232 y RS232C son mínimas, por lo que casi siempre se habla refiriéndose al mismo estándar.	El USB1.0, USB1.1 y USB2.0 presentan las mismas características lo que los diferencian son las velocidades de transmisión.  <b>USB 1.1</b> alcanza los <b>12 Mb/s</b> .  <b>USB 2.0</b> alcanza los <b>480 Mb/s (60 MB/s)</b>  Mientras que un puerto serie o paralelo tiene una velocidad de transferencia inferior a <b>1 Mb/s</b> .
<b>Niveles de voltage</b>	-12v y 12 v	5v

**Anexo2. tabla de Comparacion entre la flashlite 186 y la Omniflash**

	<b>Flashlite186</b>	<b>Omniflash</b>
Costo	<b>\$99</b>	<b>\$199</b>
Puertos	2 puertos serials RS232	2 puertos del USB, 2 puertos seriales RS232.
Regulador de voltaje(Potencia)	cc 5V	cc 5V
Procesador	33 Mhz	200 Mhz

Autores: Nayla Yamila Sire Vila  
Arlin Maury Cruz Perez

**¡Error! No se encuentra el origen de la referencia.**

Kit de desarrollo	Incluye un cable de la consola, kit del conector, adaptador de 110 VAC, disposición, diagrama esquemático y CD con el compilador de Borland C/C++ V4.52, la documentación y bibliotecas seriales del conductor. (kit 99-0081 del desarrollo 220V en disponible.). El kit incluye el tablero y el software para <b>SOLAMENTE \$99.</b>	Incluye un regulador de OmniFlash, un cable de programación, un kit del conector, un adaptador de 110 VAC, una guía de la disposición, un CD esquemático y Bootable con GCC C/C++ Compiler, programas de muestra y una documentación. El kit incluye el tablero y el software para <b>SOLAMENTE \$199.</b>
-------------------	---	--

**DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**<nombre autor>****<nombre tutor>**\_\_\_\_\_  
Firma del Autor\_\_\_\_\_  
Firma del Tutor

**¡Error! No se encuentra el origen de la referencia.**

### OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **< título del trabajo de diploma >**, fue realizado en **<nombre de lugar>**. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente  
 Parcialmente en un \_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

---



---



---



---



---



---

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a **<valor en MN o USD del efecto económico>**

Y para que así conste, se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
 Representante de la entidad

\_\_\_\_\_  
 Cargo

\_\_\_\_\_  
 Firma

\_\_\_\_\_  
 Cuño

### OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Autores: Nayla Yamila Sire Vila  
 Arlin Maury Cruz Perez

Título: **<título del trabajo de diploma>**

Autor: **<nombres y dos apellidos del o los autores>**

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

**<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:**

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad>

**<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >**

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de **<nota>**. **<Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>**

**<nombre tutor>**

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Fecha

## Glosario de términos

**BUS** (informática): Conjunto de líneas (cables) para la transmisión de datos entre los componentes de un sistema informático. Un bus es en esencia una ruta compartida que conecta diferentes partes del sistema como el microprocesador, la controladora de unidad de disco, la memoria y los puertos de entrada, salida, permitiéndoles transmitir información.

**USB:** Bus serie universal, tecnología de conexión.

**DRIVERS:** Programas o software, que es capaz de controlar un dispositivo, para su correcto funcionamiento.

**HUBS:** dispositivo que permite la conexión de otros dispositivos a la vez, dado por sus numerosas entradas o puertas

**BYTE:** Conjunto de 8 bits. Representa un carácter en lenguaje binario.

**BIT:** Dígito binario, unidad mínima de información de los dos estados 0/1. Abreviación de Binary Digit que puede ser 0 o 1. Es la unidad básica de almacenamiento y proceso de una computadora. 8 bits = 1 byte.

**bps:** (bits por segundo) unidad de medida de transferencia de información.

bytes/s: (bytes por segundo) unidad de medida de transferencia de información en byte por segundo (un carácter por segundo).

**IRQ:** Canal de interrupción. Línea directa entre el microprocesador y la tarjeta periférica para que ésta solicite atención del CPU.

**Protocolo:** Conjunto de reglas establecidas para fijar la forma en que se realizan las transacciones.

**FIFO:** First In, First Out - primero en entrar, primero en salir). Método de estructuración de datos que es utilizado en colas (queue). Se suele implementar en programación con el uso de vectores o punteros.

**Firewire:** Bus serial de Apple; es un solo enchufe, pero pueden conectarse hasta 63 dispositivos. Transfiere información de hasta 400 MB por segundos.

**RAM:** (Random Access Memory - Memoria de acceso aleatorio). Tipo de memoria donde la computadora guarda información para que pueda ser procesada más rápidamente. En la memoria RAM se almacena toda información que está siendo usada en el momento. Su capacidad de almacenamiento se mide en megabytes y más recientemente en gigabytes.

**Chip:** Microchip o también llamado circuito integrado (CI). Placa de silicio pequeña en la que se encuentran miles de dispositivos electrónicos interconectados (diodos, transistores, resistencia, capacitores, etc). Los microprocesadores son ejemplos de chips muy avanzados. También lo son las memorias digitales.

**Microprocesadores:** Microchip más importante en un ordenador, es considerado el "cerebro" de una computadora. Está constituido por millones de transistores integrados. Este dispositivo se ubica en un zócalo especial en la placa madre y dispone de un sistema de enfriamiento (generalmente un ventilador).

Lógicamente funciona como la unidad central de procesos (CPU). En el microprocesador se procesan todas las acciones de la computadora.

**CPU** (Central Processing Unit - Unidad central de procesamiento): Es el cerebro de la computadora, ubicado específicamente en el microprocesador. El rendimiento, calidad y costo de una computadora está altamente relacionado a su CPU.

**Chipset:** También llamado Circuito Integrado Auxiliar. Conjunto de circuitos integrados (chips) encargados de realizar funciones que el microprocesador delega en ellos.

**Datagrama:** Es la estructura interna de un paquete de datos. Paquetes de datos que se transfieren en una conexión.

**Dirección de memoria** (memory address): En computación, la dirección de memoria es un identificador único para una ubicación de la memoria, con las cuales una CPU u otro dispositivo puede almacenar, modificar o recuperar datos de la misma.

**Hexadecimal:** Sistema numérico en base 16, esto significa que contiene 16 símbolos únicos para representar datos: los números del 0 al 9 y las letras de la A a la F.

**Módem** (Modulador-DEModulador): Periférico de entrada/salida, que puede ser interno o externo a una computadora, y sirve para conectar una línea telefónica con la computadora. Se utiliza para acceder a internet u otras redes, realizar llamadas, etc.

**Placa base** (Placa madre, placa base, tarjeta madre, mainboard, system board, logic board, motherboard): Tarjeta central de circuitos integrados para la interconexión entre el microprocesador, las ranuras o slots para conectar tarjeta de expansión, memorias RAM, la ROM, dispositivos de almacenamiento y cableados.

**HUB** (Concentrador): Dispositivos que expanden el número de puntos de conexión al bus mediante puertos adicionales.

**PBX** (Private Branch eXchange) :Red de teléfono privada usada dentro de una empresa cuyos usuarios pueden compartir un cierto número de líneas externas para hacer llamadas al exterior de la PBX.

**PnP o Plug & Play:** Sistema de reconocimiento por el que el sistema operativo detecta un nuevo hardware. Tras la detección, el sistema busca el software que lo gestione, y que suele estar en los discos que el fabricante ofrece con el dispositivos.

**RS-232:** Estándar de comunicación serie utilizado desde los primeros PCs hasta la actualidad.

**USB-IF:** USB Implementers Forum, Corporación fundada por un grupo de empresas que se unieron para el desarrollo de la especificación del USB.

**POTS** (plain old telephone service) Servicio telefónico estándar, también llamada red telefónica pública conmutada o public switched telephone network (PSTN).

**ISDN** (Integrated Services Digital Network) en español RDSI. Red que facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados.

**UART**(Universal Asynchronous Receiver-Transmitter): Chip de ciertos sistemas digitales cuyo principal objetivo es convertir los datos recibidos en forma paralela, a forma serial, con el fin de comunicarse con otro sistema externo. También realiza el proceso inverso. Todas las computadoras y algunos módems y adaptadoras RDSI tienen su componente UART. Los UART son programables, pudiendo configurarse la velocidad, paridad, longitud y bits de parada.

**PTDC10:** Protocolo de comunicación de transferencia de información del Simulador C10.

**FAQ** (frequently-asked questions): Preguntas más frecuentes. Sección que ofrece una recopilación de las preguntas y respuestas más solicitadas por los visitantes o usuarios de un sitio web. Aclaran dudas de forma rápida, sin leer grandes textos y además evitan que se recurra continuamente al servicio técnico o al webmaster.

**Puerto de comunicación:** Un puerto es una conexión o un enchufe, el cual es utilizado para conectar un dispositivo de Hardware como impresoras o Mouse, permitiendo el intercambio de datos con otro dispositivo. También existen puertos internos definidos mediante el Software.