



## :: Puerto Paralelo - Introducción

Bienvenidos a este breve tutorial

No podía ser de otra manera, la verdad, ja! breve...!!!, sólo por ahora, pronto verás hasta donde llegaremos...

En fin, así es como nos iniciaremos, con 5 lecciones describiendo algunas de las características de este puerto y sus posibles aplicaciones, es bueno tener algún conocimiento de electrónica digital, por si no lo tienes consulta la página correspondiente, o busca otras si con ésta no te es suficiente, sabes que Internet dispone de muchísimo material al respecto.

Quiero aclarar algo, cualquier daño ocasionado a tu ordenador es sólo tu responsabilidad, por mi parte los circuitos y programas que se encuentran en esta página fueron probados en mi AMDk5, y todavía vive...!!!

Por esta razón, no me vengas con quejas, de acuerdo...?

Algunos circuitos fueron extraídos de la www.... pero solo los que más me gustaron están aquí, al menos conmigo funcionaron. Los programas que no son de mi autoría llevan el nombre de sus Creadores y no pretendo lucrar con ellos ni mucho menos, lo que sí creo, deberíamos hacer todos es agradecerles el valioso tiempo invertido en esa tarea, y lo que es más, colocarlos en la red a disposición de todos nosotros. Saludos para todos ellos...!!!

Cualquier mensaje apreciación o comentario ya sabes, a mi correo...!!! y si deseas contactar conmigo también.

Bueno, ya está, ahora sí podemos comenzar...

## :: Puerto Paralelo - Índice

---

### General

[Introducción](#)

[Registros del Puerto Paralelo](#)

[Enviando Señales por el Registro de Datos](#)

[Enviando Señales por el Registro de Contros](#)

[Recibiendo Señales por el Registro de Estado](#)

[Comentarios finales](#)

### Apéndice

[Numeración binaria y Bits](#)

[Numeración binaria, BIT, Byte y Nibbles - En detalle...!!!](#)

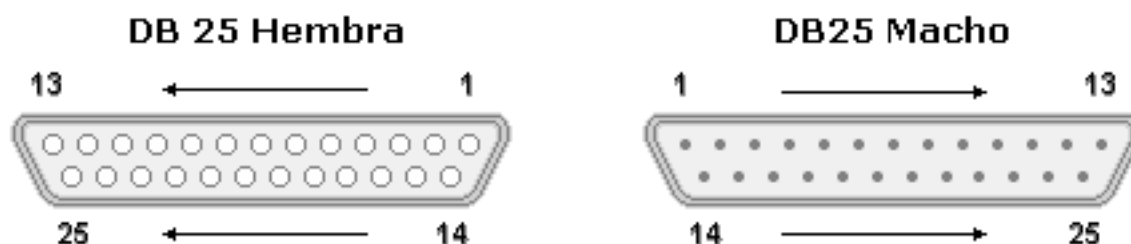
## :: Puerto Paralelo - Lección 1

Introducción...

En verdad se merecía toda una sección aparte a la cual le brindaremos una atención especial e iremos ampliándola cada vez más, ya que sus prestaciones son muchísimas, y es bueno que lo conozcas, te sorprenderás de las cosas que se pueden hacer...

Si estas leyendo este tutorial es por que tienes una PC en estos momentos y seguramente una impresora o un Scanner conectados a ella. Bien, la ficha en la cual están conectados estos dispositivos es conocida como puerto paralelo. Si la desconectas veras que tiene 25 contactos (pines de ahora en más), su nombre es Conector DB25.

Aquí tienes una imagen de los conectores



Cada pin de los conectores tiene un número asignado por lo cual es muy importante que al armar tus propios cables los identifiques correctamente o sufrirás unos terribles dolores de cabeza...!!!

Este puerto dispone de tres registros de 8 bit cada uno (un byte).

*Que no sabes lo que es un bit o un byte...?*, de acuerdo, ingresa [aquí](#) te sacas las dudas y regresas...

Sigamos con lo nuestro... cada uno de estos registros se denominan puertos o PORT., y cada uno de sus bits, representa un pin determinado del puerto. Los pin's que van del 18 al 25 (ambos inclusive): Son para masa, y sirven para conectar las descargas de los circuitos.

Veamos ahora los tres registros

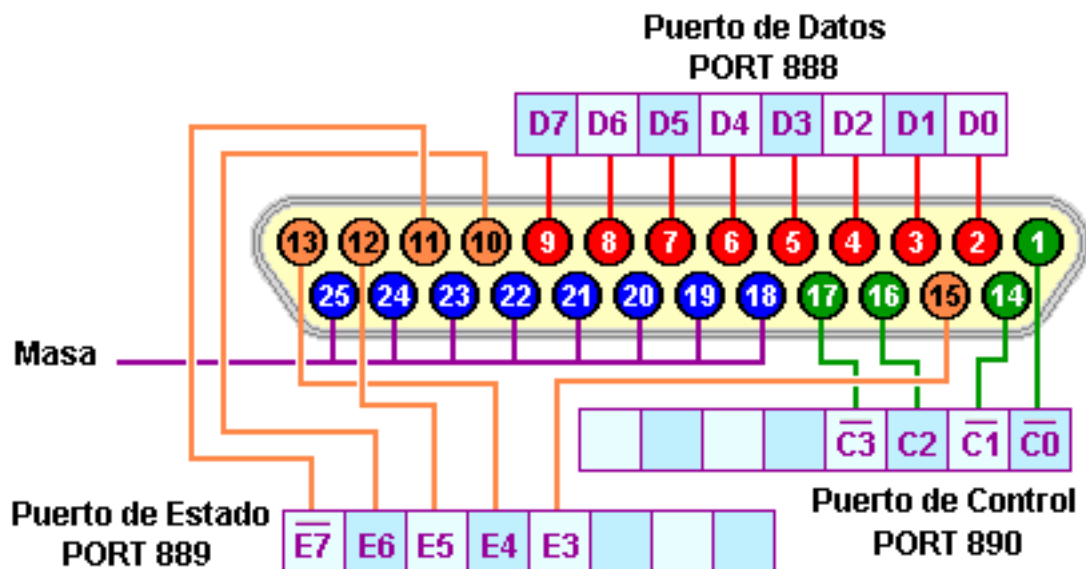
**Puerto de datos (Pin 2 al 9):** Es el PORT 888 y es de solo escritura, por este registro enviaremos los datos al exterior de la pc, cuidado...!!!, no

envíes señales eléctricas al ordenador por estos pines.

**Puerto de estado (Pin 15, 13, 12, 10 y 11):** Es el PORT 889 y es de solo lectura, por aquí enviaremos señales eléctricas al ordenador, de este registro solo se utilizan los cinco bits de más peso, que son el bit 7, 6, 5, 4 y 3 teniendo en cuenta que el bit 7 funciona en modo invertido.

**Puerto de control (Pin 1, 14, 16 y 17):** Es el correspondiente al PORT 890, y es de lectura/escritura, es decir, podremos enviar o recibir señales eléctricas, según nuestras necesidades. De los 8 bits de este registro solo se utilizan los cuatro de menor peso o sea el 0, 1, 2 y 3, con un pequeño detalle, los bits 0, 1, y 3 están invertidos.

En esta imagen puedes ver los tres registros, sus bits y los pines asignados a cada uno de ellos. La imagen corresponde a un conector DB-25 (Hembra).



Algo más... La tensión de trabajo del puerto es de 5 voltios, por lo que necesitamos una fuente estabilizada o regulada de tensión, esto es importante tenerlo en cuenta, ya que estaremos enviando señales al puerto. Por otro lado, si bien puedes utilizar la PC para enviar señales al exterior sin necesidad de una fuente externa, es recomendable utilizarla y así no exiges demasiado al puerto y te evitas problemas.

Ahora bien, si activas un bit de salida por el puerto, este permanecerá así hasta que lo cambies, es decir que estarás enviando 5V de forma continua hasta que lo pongas a 0.

Pero bueno.. eso es programación, y lo veremos a su debido momento, por lo pronto deberemos armarnos de unos circuitos para verificar el

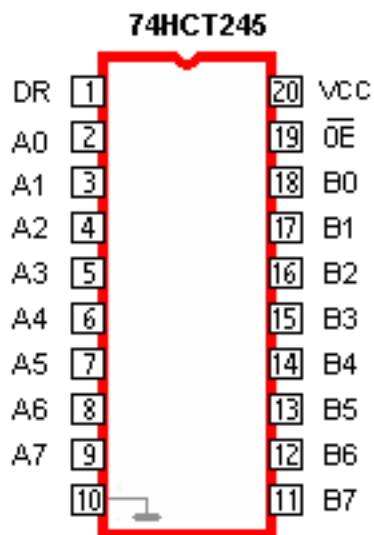
**funcionamiento del puerto. Aclarado esto pasemos a su construcción...**

## :: Puerto Paralelo - Lección 2

Enviando datos por el puerto paralelo...

Para comenzar construiremos un circuito que nos permita enviar señales por el puerto paralelo y lo haremos utilizando diodos LEDs, como el consumo de los LEDs es superior al que nos envía el puerto utilizaremos un buffer como el 74HC244 o el 74HCT245, prefiero este último ya que la construcción de la placa será mas sencilla, aquí tienes los datos del integrado

### Non-inverting 3-state outputs



### Tabla de Verdad

INPUTS		INPUTS/OUTPUTS	
$\overline{OE}$	DIR	$A_n$	$B_n$
L	L	A = B	inputs
L	H	inputs	B = A
H	X	Z	Z

Nota:

X = Sin importancia

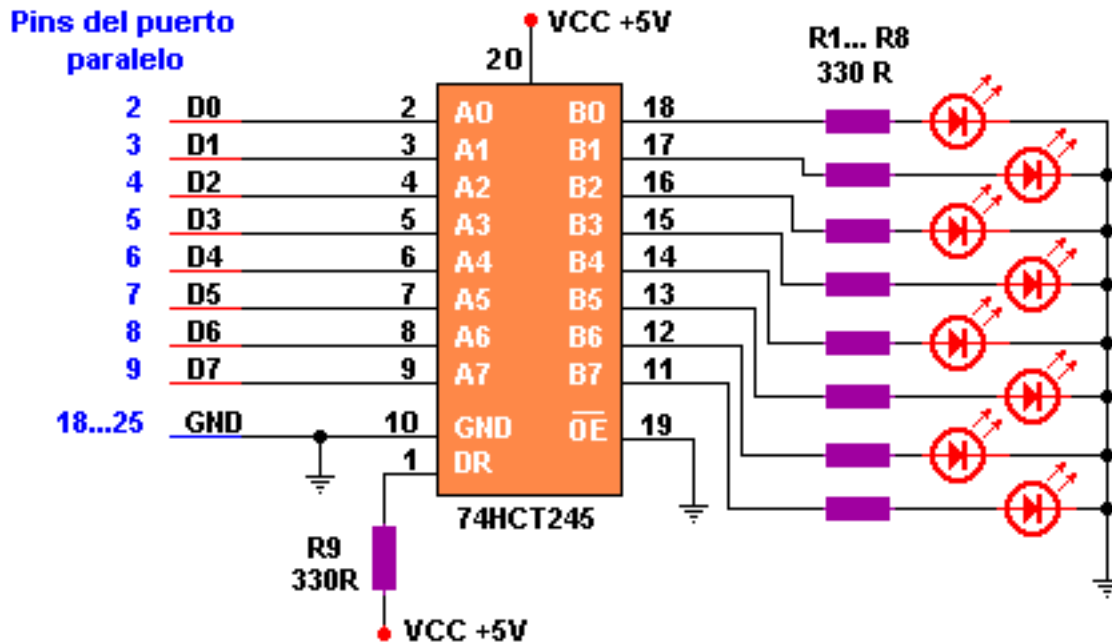
Z = Alta impedancia

Este integrado tiene la ventaja de ser bidireccional, es decir todos los pin's A pueden ser entradas y los B salidas si DIR (pin1 DR) se encuentra a nivel bajo (L), el modo invertido se obtiene con DR a nivel alto. Si el terminal OE (pin 19) se encuentra a nivel alto (H) el integrado pone los pin's A y B con alta impedancia, es decir ni entra ni sale señal alguna, en ese caso el estado de DR no tiene importancia. En la siguiente tabla tienes los nombres de los pines y sus funciones correspondientes.

No. PIN	Nombre	Función
1	DIR	Control de dirección
2, 3, 4, 5, 6, 7, 8, 9	A0-A7	Entrada/Salida de datos
10	GND	Fuente (0V)
18, 17, 16, 15, 14, 13, 12, 11	B0-B7	Entrada/Salida de datos
19	$\overline{OE}$	Habilitación (Activo (L))
20	Vcc	Fuente (+5V)

En el siguiente esquema no se representaron todos los pines del puerto, sino los

correspondientes al puerto de datos y los de masa, que son los que utilizaremos en esta primera práctica...



Lo que queda ahora es ver como funciona, y lo haremos sencillito, para ello recurrimos a nuestros superiores en programación *ya que de eso yo no se nada... :o*)).

Es broma...!!! Para no meterte en líos con las sentencias de programación las cuales puedes programarlas tu mismo en Basic, en Pascal, en C, en Delphi y todos los demás lenguajes que quieras, yo hice uno en **C** y este es el código fuente...

```

/*Programa que envía señales al puerto de datos ingresando un número decimal*/

#include<dos.h>
#include<stdio.h>
#include<conio.h>

int a;      /*variable, valor que enviare al puerto*/
void main()
{
a=0;      /*asignando 0 a la variable a*/
outp(888,a); /*apago todos los LED's*/
clrscr(); /*limpio la pantalla*/

printf("Ingrese el número en decimal para enviar al puerto.\n");
printf("El número máximo permitido es 255:\n");

scanf("%d",&a); /*tomo el valor y se lo paso a "a"*/
outp(888,a); /*lo envío al puerto de datos*/

```

```

printf("tachannnn... Dato enviado...!!!\n\n");
printf("Se acabó, presione una tecla para salir");

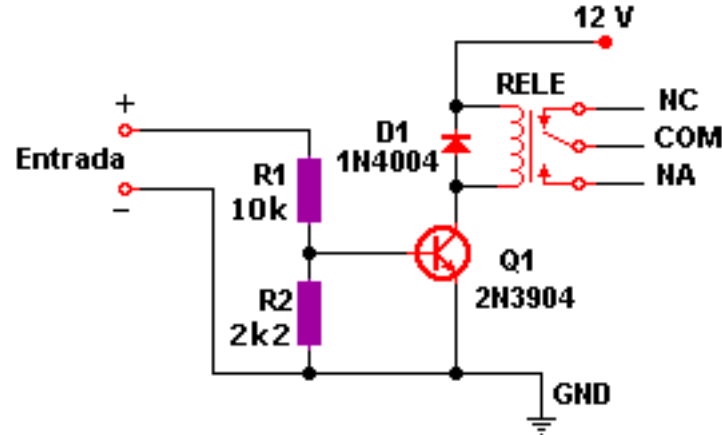
getch();          /*...y hasta luego*/
outp(888,0);     /*apago todos los LED's*/
}

```

Creo que no es necesario comentarlo, el ejecutable, está [aquí](#), y para los amantes de VBasic, un paquete...!!! con código fuente, comentarios y el ejecutable...



Deseas encender algún aparato de casa...? No hay problema, sólo le agregas una etapa de potencia a cada salida, algo como ésto...



Antes de que me olvide... Me encontré con un pequeño programa que te chequea el estado del puerto, supongo que su autor es Fred Bulback, al menos eso dice en el About, en concreto el programa es [Parallel Port Monitor](#)

Y con esto es suficiente, ya no necesitaras armar los circuitos para verificar tus programas, aunque si deseas utilizar el puerto no hay de otra tendrás que hacerlos...

: -P

## :: Puerto Paralelo - Lección 3

Enviando datos por el puerto de control...

Este puerto es de lectura-escritura. En este caso lo utilizaremos para enviar señales desde la PC hacia el exterior, y el circuito que se utilizara será similar al empleado en el puerto de datos, de ese modo es posible tener 12 salidas, lo que si se debe tener en cuenta es que el primero el segundo y el cuarto bit de este registro están invertidos, pero esta dificultad es superada con la codificación del programa que controla este puerto

Si deseas ver como funcionan las salidas, arma el circuito anterior y ya podrás observar las variaciones en los led's.

El programa de prueba lo tienes aquí.

```

/*Programa que envía señales al puerto de control ingresando un número decimal*/

#include< dos.h>
#include< stdio.h>
#include< conio.h>

int a;      /*variable, valor que enviare al puerto*/
void main()
{
a= 11;      /* 11 en binario es 1011 los valores 1 corresponden*/
           /*a los pines invertidos*/
outp(890,a); /*apago todos los LED's*/
clrscr();   /*limpio la pantalla*/

printf("Ingrese el número en decimal para enviar al puerto.\n");
printf("El número máximo permitido es 15:\n");

scanf("%d",&a); /*tomo el valor y se lo paso a "a"*/
outp(890,a);   /*lo envío al puerto de estado*/

printf("tachannnn... Dato enviado...!!!\n\n");
printf("Se acabó, presione una tecla para salir");

getch();      /*...y hasta luego*/
outp(890,11); /*apago todos los LED's*/
}

```

Tendré que hacer algunas aclaraciones...?, no creo verdad...? bueno, sólo ten en

cuenta los bits invertidos cuando pruebes este programa, que por cierto aquí lo tienes ya compilado se llama **Pcontrol.exe**.

Y como no podía faltar, para los amantes de Visual Basic también les dejo el código, el exe y sus comentarios.



Algo que olvidaba es la distribución de los pines de control, para que sueldes en el conector DB25, y son los siguientes.

**C0 --> pin 1**  
**C1 --> pin 14**  
**C2 --> pin 16**  
**C3 --> pin 17**  
**Masa --> pin 18 al**  
**25**

Bien... Con ésto terminamos el envío de señales, luego ampliaremos más, por ahora hasta aquí, espero logres familiarizarte con este tema ya que tiene más posibilidades de las que vimos hasta ahora, te sorprenderías de las cosas que se pueden hacer...!

Pasemos a otro tema, la recepción de señales...!!!

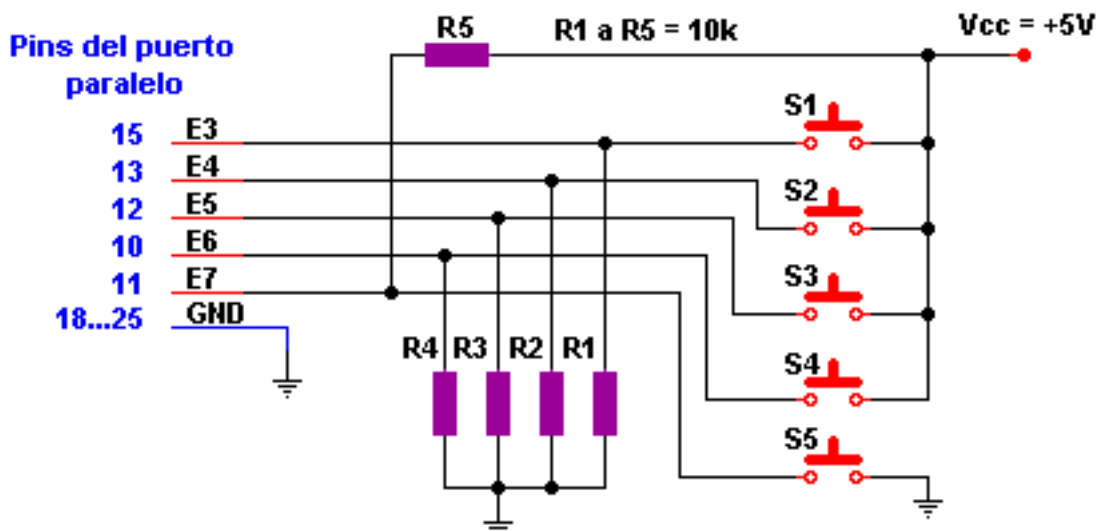
## :: Puerto Paralelo - Lección 4

Recibiendo datos por el puerto paralelo...

Ahora si ya estamos en condiciones de continuar...

Para recibir datos en nuestro ordenador por el puerto paralelo utilizaremos su registro de estado, recuerda que solo dispones de 5 bits (7, 6, 5, 4 y 3) y que el bit 7 funciona en modo invertido

El circuito que construiremos será uno de los mas sencillos, solo habrá que tener en cuenta que un bit puede tener dos estados posibles, ya sea 0 (0 voltios) o 1 (5 voltios) no podemos dejarlo librado al azar, razón por la cual, si no se envía señal alguna, éste deberá estar unido a masa, de acuerdo...? El esquema eléctrico es el siguiente;



Observa que para el bit 7 la conexión es invertida si lo pones a 0 el programa lo leerá como un 1, si no presionas el pulsador se leerá como 0 lógico.

Con respecto al programa, para no liar en lenguajes de programación hice uno que nos verifica el estado del puerto, puedes bajarlo si lo deseas, eso sí...!!! "no es eeeel programa", pero que va, sirve para estudiarlo, jejejejeje



Luego podrás hacer uno mucho más interesante que éste, eso espero...!!!

## :: Puerto Paralelo - Lección 5

---

Creo que haré la expansión a 32 salidas...!!!

Pero por ahora me tomaré un descanso y pasaremos a otra cosa, he visto por ahí gran difusión del tema así que con esto mas de ayuda ya tienes suficiente.

La verdad aquí en Jujuy, este tema se ha vuelto indispensable, por lo de la Fiesta Nacional de los estudiantes y sus desfiles de carrozas, así que lo reservaré un tiempo más hasta que me anime a presentarles todos sus detalles. Aún no quiero competencia...!!! jejejejeje

Recomendación...!!!

Si por algun motivo quieres trabajar con el puerto sin arriesgarte a dañarlo, en la sección de Descarga tienes un programa para monitorear el estado del Puerto Paralelo.

Saludos, y que Dios proteja sus PC's. :o))

***R-Luis...***

## :: Puerto Paralelo - Bit, Byte y esas yerbas

Pues bien, supongo que sabrás que el ordenador manda los datos en forma de bytes. Un byte son 8 bits y un bit es la mínima cantidad de información que un ordenador puede tratar. Es decir, un bit puede tomar sólo dos valores posibles, que son 0 ó 1.

Hablando en lenguaje eléctrico el 0 significa 0 voltios y el uno significa 5 voltios (*por aquello de la electrónica digital*). Cualquier valor que mandemos al puerto, el ordenador lo transformará en ceros y unos. Esto se conoce como **sistema binario de numeración**. Como tenemos datos de 8 bits (1 byte), aplicando algo de las matemáticas que aprendimos en EGB, vemos que con sólo 2 dígitos (el cero y el uno) podremos representar  $2^8$  valores. haciendo números tendremos 256 valores, que se corresponden a la siguiente tabla:

Decimal.	Binario.
<b>0</b>	<b>00000000</b>
<b>1</b>	<b>00000001</b>
<b>2</b>	<b>00000010</b>
<b>3</b>	<b>00000011</b>
<b>4</b>	<b>00000100</b>
<b>5</b>	<b>00000101</b>
<b>6</b>	<b>00000110</b>
<b>7</b>	<b>00000111</b>
<b>8</b>	<b>00001000</b>
<b>9</b>	<b>00001001</b>
<b>etc. etc. etc.</b>	<b>etc. etc. etc.</b>
<b>255</b>	<b>11111111</b>

Bueno, espero que lo hayas comprendido y si no es así, consulta [aquí](#) que esto ya es para gente mas difícil, *como yo!!!... jeje.*

## :: Puerto Paralelo - Bit, Byte y Esas Yerbas para Duros...!!!

### BIT.

El ordenador se compone de dispositivos electrónicos digitales, por lo tanto éstos solo pueden adoptar únicamente dos estados, que representamos matemáticamente por 0 y 1. Cualquiera de estas unidades de información se denomina BIT, *contracción de «binary digit» en inglés.*

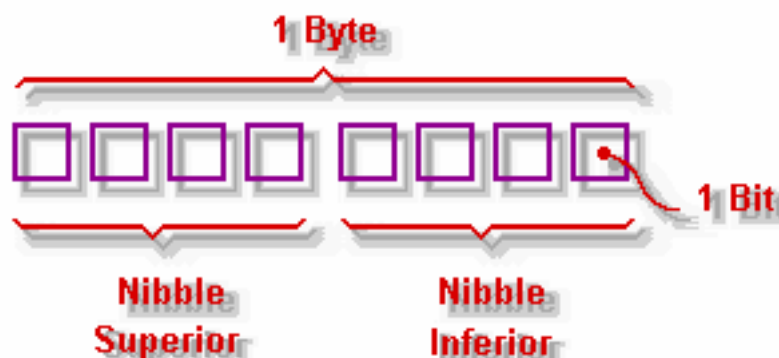
### BYTE.

Cada grupo de 8 bits se conoce como byte u octeto. Es la unidad de almacenamiento en memoria, la cual está constituida por un elevado número de posiciones que almacenan bytes. La cantidad de memoria de que dispone un sistema se mide en Kilobytes (1 Kb = 1024 bytes), en Megabytes (1 Mb = 1024 Kb), Gigabytes (1 Gb = 1024 Mb), Terabytes (1 Tb = 1024 Gb) o Petabytes (1 Pb = 1024 Tb).

Los bits en un byte se numeran de derecha a izquierda y de 0 a 7, correspondiendo con los exponentes de las potencias de 2 que reflejan el valor de cada posición. Un byte nos permite, por tanto, representar 256 estados (de 0 a 255) según la combinación de bits que tomemos.

### NIBBLE.

Cada grupo de cuatro bits de un byte constituye un nibble, de forma que los dos nibbles de un byte se llaman nibble superior (el compuesto por los bits 4 a 7) e inferior (el compuesto por los bits 0 a 3).



Veamos... Un bit es la posición que ocupa un número el cual será "0" o "1" ya que son los únicos valores que admite.

Si dispones de un bit solo tienes 2 posibilidades

1ra. posibilidad	1
2da. posibilidad	0

Si dispones de 2 bits tienes 4 posibles combinaciones.

1ra. combinación	00
2da. combinación	01
3ra. combinación	10
4ta. combinación	11

En fin si dispones de 4 bit tienes 16 posibles combinaciones que son

0000  
0001  
0010  
0011  
0100  
0101  
0110  
0111  
1000  
1001  
1010  
1011  
1100  
1101  
1110  
1111

Pues bien, estas son las que corresponden a un NIBBLE, esto es muy importante ya que cada nibble representa una cifra en el sistema hexadecimal que van desde el 0 al 9 y luego de la A a la F

Nibble en binario	Valor Hexadecimal	Valor Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Existe una forma sencilla de saber cuantas posibles combinaciones puedes obtener con una determinada cantidad de bits, como se trata de un sistema binario de numeración, este se organiza en base 2, entonces  $2^n$  nos da la cantidad de combinaciones que podemos realizar, ( $n$  en este caso, y solo en este caso lo tomaremos como la cantidad de bit disponibles, recuerda que solo es en este caso, esta...?) veamos un ejemplo;

Al disponer de 1 bits tendremos  $2^1 = 2$  combinaciones posibles

Al disponer de 2 bits tendremos  $2^2 = 4$  combinaciones posibles

Al disponer de 3 bits tendremos  $2^3 = 8$  combinaciones posibles

Al disponer de 4 bits tendremos  $2^4 = 16$  combinaciones posibles

Al disponer de 5 bits tendremos  $2^5 = 32$  combinaciones posibles

Al disponer de 6 bits tendremos  $2^6 = 64$  combinaciones posibles

Al disponer de 7 bits tendremos  $2^7 = 128$  combinaciones posibles

Al disponer de 8 bits tendremos  $2^8 = 256$  combinaciones posibles

En este ultimo caso estamos hablando de un byte (recuerdas, ...eso de los 8 bits...!!! que forman los dos nibbles)

El sistema de numeración hexadecimal agrupa los bits de a cuatro, es por eso que aparecen los nibbles (grupos de 4 bits), observa esta equivalencia de ejemplo; y verifica que sea verdad de acuerdo...?

<b>BYTE</b>	<b>Valor hexadecimal</b>
0111 0101	75

Uffffff...!!!, Fue de terror Verdad...???