

## Comenzando con el Puerto Paralelo Uso, Aplicación y Simulación

### Descripción:

Bueno, este es el primer tutorial que escribo de cómo usar el Puerto Paralelo. En casi todas las Pc's que conozco hay un solo puerto y tiene como Registro Base 378H (Así q por ahora lo tomaré como un Standard). Para poder acceder al Puerto, se usaría fácilmente en C o en Asm `Inxx(Port)` y `Outxx(Port,Dato)`. Pero actualmente los sistemas operativos son **mezquinos** así q para poder acceder a ellos debemos hacerlo mediante Drivers. (ver [Aquí](#))

Pero para no preocuparnos por eso. Yo voy a usar la DLL `InpOut32.dll` que mediante 2 rutinas `Inp32` y `Out32`, Resuelve el problema.

Además aquí voy a usar 3 tipos de codificación: Assembler, C++ y Visual Basic.

### TERMINOLOGIA:

A lo largo de este tutorial, voy a ser uso de ciertos términos que considero que seria muy bueno que se conocieran antes.

Bit '1' : Significa que la salida de esta puerta es 5V.

Bit '0' : Significa que la salida es 0v.

Bit 'X' : Significa que no importa el valor de este bit, puede ser '1' o '0'

Bit '~X' : Significa que este dato es negado.

Registro: Arreglos de 8 Bits, que sirven para la configuración del Puerto Paralelo

### Un Poco de Teoría.

#### Algo de Logica: And, Or, Xor:

Para comenzar, Estas operaciones son muy importantes, ya que nos permiten el manejo del Puerto, cambiar Bits, invertirlos, enmascararlos. Como se vera a continuación:

#### And: Dato = Op1 And Op2 = Op1 & Op2

En pocas palabras, si Op2='1', no importa como sea Op1='X', la respuesta seria Dato='X' ; Ahora si Op2='0' el dato='0'

Es decir que si tenemos un Byte=(D7) (D6) (D5) (D4) (D3) (D2) (D1) (D0). Y queremos quedarnos con el dato D7, D5, y D4. y que el resto sea '0'. Lo que tendríamos que hacer es:

```
Result=Byte And 10110000b;// Result = Byte & B0;
```

Y nos quedaria:

Byte =	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	Operador
	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	& = And
Result =	<b>D7</b>	<b>0</b>	<b>D5</b>	<b>D4</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	//Muy interesante

Es decir que nos sirve para colocar los bits que deseamos a '0'

Indiciándose en la Programación y simulación con el Puerto Paralelo

### Or: Dato = Op1 Or Op2 = Op1 | Op2

En pocas palabras, si Op2='1', no importa como sea Op1='X', la respuesta seria Dato='1'; Ahora si Op2='0' el dato='X'

Es decir que si tenemos un Byte=(D7) (D6) (D5) (D4) (D3) (D2) (D1) (D0). Y queremos que D7,D6, D4,D2 sean '1' y el resto quede igual tendríamos que hacer:

```
Result=Byte or 11010100b;// Result= Byte | 0xD4
```

Y nos quedaría:

Byte =	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	Operador
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	= or
Result =	<b>1</b>	<b>1</b>	<b>D5</b>	<b>1</b>	<b>D3</b>	<b>1</b>	<b>D1</b>	<b>D0</b>	//Muy interesante

Es decir, que mediante una Or podemos obligar a que la respuesta sea '1'.

### Xor: Dato = Op1 Xor Op2 = Op1 ^ Op2

En pocas palabras, si Op2='1', no importa como sea Op1='X', la respuesta seria Dato='~X'; Ahora si Op2='0' el dato='X'. Es decir que invierte el Bit donde encuentra Op='1'

Es decir que si tenemos un Byte=(D7) (D6) (D5) (D4) (D3) (D2) (D1) (D0). Y queremos invertir D7, D5, y D4. lo que tendríamos que hacer es:

```
Result = Byte Xor 10110000b;// Result = Byte ^ B0
```

Y nos quedaria:

Byte =	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	Operador
	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	^ =Xor
Result =	<b>~D7</b>	<b>D6</b>	<b>~D5</b>	<b>~D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	//Muy interesante

Es decir, nos permite invertir el Bit donde esta con '1'

Bueno, Espero que este claro lo de las operaciones lógicas. Si no revisen su libro de principios de sistemas digitales. :D ;)

## Aprendiendo como funciona mi Puerto Paralelo:

Aquí, no voy a redundar en los pines, es decir, no voy a detallar cuales son y su numeración, por que de eso ya hay bastante en la Web, y ya he puerto información en la sección [Puertos](#) de mi web.

Bueno, para manejar el puerto paralelo, hay q saber acerca de sus registros. Que Son 3 **Base**, **Estado** y **Control**.

La dirección de la mayoría de Pc's q conozco, esta en 378h. Aunque puede variar dependiendo del SO q uses. Yo lo voy ha usar como un Estándar para este tutorial.

## Descripción de Registros:

### Registro de Datos. (Su dirección es 378h = Dirección Base).

D7 D6 D5 D4 D3 D2 D1 D0

Este registro es de 8 bits. Es por aquí q enviamos la información hacia el exterior.

Ejm Activamos un Led, un Relé, Motores DC, Motores a pasos.

Una Alarma. Lo q nos imaginemos. O también recibimos datos. Ejm: Señal de un ADC para digitalizar una señal externa.

Por defecto, solo funciona como Salidas, pero es posible Hacerla funcionar como entrada. Para esto hay q modificar en Bit 5 del Registro de Control.

Esto se explica mas abajo con mucho mas cuidado

Dependiendo del lenguaje de programación q uses, la sintaxis es :

Outxxx (378h,Dato).

Ejemplo en C++ Usando la Librería InpOut32.dll

```
#include "stdafx.h"
#include "Iostream.h"
//Definimos los prototipos de nuestras funciones
//Recuerdese q se debe agregar a la etapa de Link, el archivo InpOut32.lib
short _stdcall Inp32(short PortAddress);
void _stdcall Out32(short PortAddress, short data);
void main(void)
{ short data;
data=Inp32(0x378);
cout<<"El Registro Estado es: "<<data<<"\n";
return; }
```

Ejemplo en Assembler – Para DOS

```
.model small
.code
    mov al,11111111b; Mandamos q todas las salidas sean '1'
    mov dx,378h
    in dx,al
    mov ah,4ch
    int 21h
end
```

### Registro de Estado (Su dirección es 379h) (Dirección Base +1).

~S7 S6 S5 S4 S3 X X X

Este registro es de solo Entradas, pero solo es de 5 Bits.

Donde:

~S7 :Significa que este dato esta invertido.

X :Significa que su valor es indeterminado.

## Indiciándose en la Programación y simulación con el Puerto Paralelo

Para obtener el Valor Real de Este registro, tenemos que enmascararlo, Haciendo un Xor con 128 o su equivalente 80h ó 0x80 para C o &H80 en Basic o en binario 10000000b , donde se observa q el bit a cambiar debe ser el 8vo. Y luego un And por 248=0xF8=1111000b (En este ultimo se observa que los valores con los que te debes quedar son los 5 ultimos).

Este Registro lo podemos usar por ejemplo para ver la entrada de sensores.

Donde para obtener el valor Real su Sintaxis seria:

```
Estado = Inxxx(379H) Xor 0x80 And 0xF8.
```

En C: Estado = (Inp32(0x379) ^ 0x80) & 0xF8

Ejemplo en C para obtener: Valor Real del Registro de Datos

```
#include "stdafx.h"
#include "Iostream.h"
//observar que se debe agregar al Link el archivo InpOut32.lib
short _stdcall Inp32(short PortAddress);
void _stdcall Out32(short PortAddress, short data);
void main(void)
{ short estado;
estado=Inp32(0x379);
//Este codigo es el Equivalente en Asm para
// Estado = (Inp32(0x379) ^ 0x80) & 0xF8
_asm xor estado,10000000b
_asm and estado,0xF8
cout<<"El Registro Estado es: "<<estado<<"\n";
return; }
```

### Registro de Control. (37Ah) (Dirección Base +2)

Este registro, es bidireccional. Sus datos son

X X X X ~C3 C2 C1 C0

Ahora para invertir las señales C3, C1 y C0

Tenemos que hacerle un xor con 00001011b y para quedarnos con el dato valido, le hacemos un And con 0xF=00001111b

Ejm:

```
control = (Outxx(37AH) Xor 1011b ) And 0xF.
```

En C: control = (Out32(37AH) ^0x0B) & 0xF.

**Este Registro es muy importante**, ya nos permite colocar al puerto paralelo en modo Bidireccional. Es decir se puede tambien recibir datos por el Registro de Datos. Por ahora siempre lo mandare a 0, es decir q sea Salida. Pero si quisiera hacerlo entrada, tendría q colocar el Bit5 equivalente a C5 en '1'.

La Sintaxis para hacerlo Bidireccional es:

```
Int Caso=1 o 0 ;Según quiera 1=Input, 0=Output
```

```
control= Outxx(0x37A,((caso <<5) or (Inxxx(0x37A) And 11011111))
```

En C: control =Out32(0x37A,((caso<<5) | (Inp32(0x37A) & 0xDF

Con caso<<5, significa rotamos 5 veces a la Izq a caso .

## Indiciándose en la Programación y simulación con el Puerto Paralelo

```
//Ejemplo en C para Obtener el Valor, que tiene el Registro de Control.
#include "stdafx.h"
#include "Iostream.h"
//observar que se debe agregar al Link el archivo InpOut32.lib
short _stdcall Inp32(short PortAddress);
void _stdcall Out32(short PortAddress, short data);
void main(void)
{
short control;
control=Inp32(0x37A);
//El codigo que viene es Equivalente
// control = (Out32(37AH) ^0x0B ) & 0xF
_asm xor control,1011b
_asm and control,0xF //Para Quedarnos con los 4 Primeros Bits
cout<<"El Registro Control es: "<<control<<"\n";
return;
}
```

Prueba del Puerto de cómo hacerlo Bidireccional, este programa te mostrara como queda el control si lo pones en modo Bidireccional. No hace el cambio por obvios motivos. Pero si lo quieres probar, al final coloca `Out32(0x37A,control);` Y ya estara en modo Bidireccional:

```
// bidire.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include "Iostream.h"
//observar que se debe agregar al Link el archivo InpOut32.lib
short _stdcall Inp32(short PortAddress);
void _stdcall Out32(short PortAddress, short data);
void main(void)
{
short caso,control;
caso=1;//Suponiendo q lo quiero hacer Bidireccional
// Dato= Outxx(0x37A,((caso <<5) or (Inxxx(0x37A) And 11011111)
control=Inp32(0x37A);
cout<<"Tenemos en Control: "<<control<<"\n";
_asm and control,11011111b
_asm rol caso,5 //caso=caso>>5; //Roto caso 5 bits
_asm mov ax,caso //control =control or caso
_asm or control,ax
cout<<"Lo que enviariamos seria: "<<control<<"\n";
return;
}
```

Mas información de esto lo puedes encontrar en mi web, o puedes escribirme a mi e-Mail : [yoelocmin@hotmail.com](mailto:yoelocmin@hotmail.com)

<http://proyectosfie.tk>

Indiciándose en la Programación y simulación con el Puerto Paralelo

### Haciendo lo que vinimos:

Para comenzar, he hecho un pequeño programa. En visual Basic.

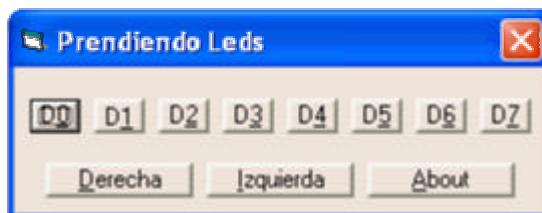
### Pero como!!!

Bueno, Sabemos que Visual Basic no permite manejar el Puerto Paralelo, pero nuestra DLL **Sí**, ES por eso que lo adjuntaremos, mediante un archivo obj:

### InpOut32.bas

```
'Declaramos las Variables de la Dll
'Aquí declaramos como función a Inp32
'Ya q lo que hace es devolvernos el valor q indica la dirección que se le Asigne.
Public Declare Function Leer Lib "Inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Byte
'Aquí declaramos como una Subrutina a Out32
'Ya que no devuelve ningun valor.
Public Declare Sub Escribir Lib "Inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Byte)
'Aquí estoy declarando esta Api de Windows, para usarlo en los retardos.
Public Declare Sub Sleep Lib "Kernel32.dll" (ByVal PortAddress As Integer)
```

Nuestro Programita tiene la Forma:



Las características basicas de los controles son:

Control	Nombre	Caption
Form	FrmLeds	Prendiendo Leds
Button	CmdD0	D&0
Button	CmdD1	D&1
Button	CmdD2	D&2
Button	CmdD3	D&3
Button	CmdD4	D&4
Button	CmdD5	D&5
Button	CmdD6	D&6
Button	CmdD7	D&7
Button	CmdEfe1	&Derecha
Button	CmdEfe1	&I zquierda
Button	CmdAbout	&About

## Indiciándose en la Programación y simulación con el Puerto Paralelo

Donde su Código Correspondiente es:

### **Form1.frm**

```
'Declaramos las variables generales
Dim Dato As Byte 'Dato es usado por todos las Subrutinas
Private Sub CmdAbout_Click()
'Muestro Mensaje en Pantalla: el vbCrLf es equivalente a ENTER
MsgBox "Prendiendo Leds " + vbCrLf + vbCrLf + " por:" + vbCrLf + " Yoel Ocmín " +
vbCrLf + vbCrLf + "yoelocmin@hotmail.com" + vbCrLf + "http://proyectosfie.tk",
vbQuestion, "Acerca..."
End Sub

Private Sub CmdD0_Click()
Dato = Leer(&H378) Xor &H1 'Cambio el Bit 1=Lsb=D0
Envia
End Sub
Private Sub CmdD1_Click()
Dato = Leer(&H378) Xor &H2 'Cambio el Bit 2
Envia
End Sub
Private Sub CmdD2_Click()
Dato = Leer(&H378) Xor &H4 'Cambio el Bit 3
Envia
End Sub
Private Sub CmdD3_Click()
Dato = Leer(&H378) Xor &H8 'Cambio el Bit 4
Envia
End Sub
Private Sub CmdD4_Click()
Dato = Leer(&H378) Xor &H10 'Cambio el Bit 5
Envia
End Sub
Private Sub CmdD5_Click()
Dato = Leer(&H378) Xor &H20 'Cambio el Bit 6
Envia
End Sub
Private Sub CmdD6_Click()
Dato = Leer(&H378) Xor &H40 'Cambio el Bit 7
Envia
End Sub
Private Sub CmdD7_Click()
Dato = Leer(&H378) Xor &H80 'Cambio el Bit 8 =MsB=D7
Envia
End Sub

Private Sub CmdEfe1_Click()
'Efecto '1':
'Rotamos un Bit del LSB -> MSB
'VB, no tiene la funcion rotar Bit, haci que lo q
'Colocar en Dato=00000001 y multiplicarlo por 2
```

## Indiciándose en la Programación y simulación con el Puerto Paralelo

```
' Dato=00000010 ->Roto un Bit
'Si lo multiplico nuevamente.
' Dato=00000100 ->Roto un Bit
'Y asi sucesivamente hasta q llega al Octavo Bit.
Dato = 1
While Dato < &H80
Envia
Sleep 250 'Ocasiono un retardo de 250 milisegundos
Dato = Dato * 2
Wend
Envia ' como no se envio el ultimo bit se lo envia recien
Sleep 250 '
Dato = 0 'Reseteamos el Puerto.
Envia
Sleep 250
End Sub
```

```
Private Sub CmdEfe2_Click()
'Efecto '2':
'Rotamos un Bit del MSB -> LSB
'VB, no tiene la funcion rotar Bit, haci que lo q
'Colocar en Dato=10000000 y Dividimos por 2
' Dato=01000000 ->Roto un Bit
'Si lo multiplico nuevamente.
' Dato=00100000 ->Roto un Bit
'Y asi sucesivamente hasta q llega al Primer Bit.
Dato = &H80
While Dato > &H1
Envia
Sleep 250
Dato = Dato / 2
Wend
Envia
Dato = 0
Sleep 250
Envia
Sleep 250
End Sub
```

```
Private Sub Form_Load()
'Al iniciar a cargar el Form, reiniciamos el Puerto
Escribir &H378, 0
End Sub
Private Sub Envia()
'Solo envia el Dato al Puerto Paralelo
Escribir &H378, Dato
End Sub
```

Bueno Eso es Todo con Respecto a Visual Basic.

Indiciándose en la Programación y simulación con el Puerto Paralelo

## Simulación:

Bueno ahora, vamos a Hacer un Circuito y simular el programa.

### Como!!!

Bueno, uno puede simular el programa hecho en Visual Basic, con otro programa.

Yo voy a usar el Proteus Lite - I sis. Para ello voy a necesitar la librería y

modelo Respectivo. Se llaman Modelo : Port.dll y Library: Lpt02.lib

Se supone que debe estar junto a este archivo, en al versión comprimida, pero si no es así bajatela de la seccion [Proteus](#) de mi pagina web.

<http://proyectosfie.tk>

OJO:

Esa Librería y Modelo son de mi cosecha.

Otra cosa. El Esquemático que acompaña el archivo fue hecho con la versión 6.3 sp1, que la puedes descargar del enlace q' indica mi Web en la sección [Proteus](#)

Si te bajas esa versión no te olvides colocarle los 2 parches respectivos para su buen funcionamiento.

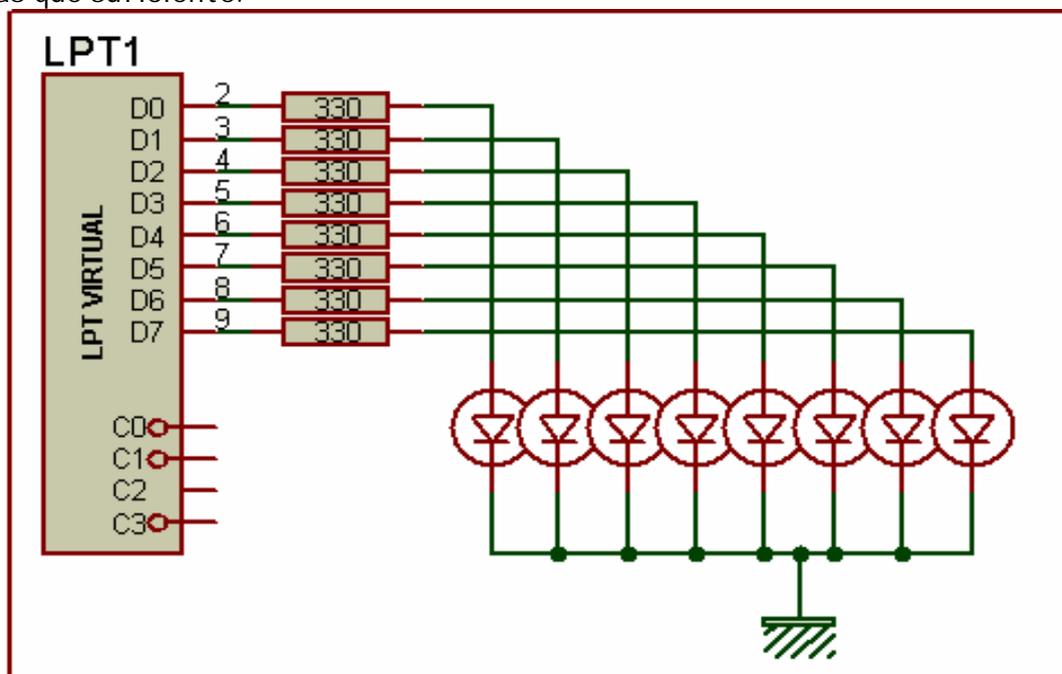
Por Ahora, creo que con poner el Esquema es Suficiente, para la simulación usa las Librerías:

{ACTIVE}: LED-YELLOW :Led amarillo

{ACTIVE}: RES :Resistencia de 330

{LPT02}:LPT02 :Puerto Paralelo Virtual

Ahora, no es necesario que explique el circuito, porque creo q con el esquema es mas que suficiente.



Indiciándose en la Programación y simulación con el Puerto Paralelo

Ahora dale a simular y Ejecuta el programa de prueba:

Y Ya Esta:

