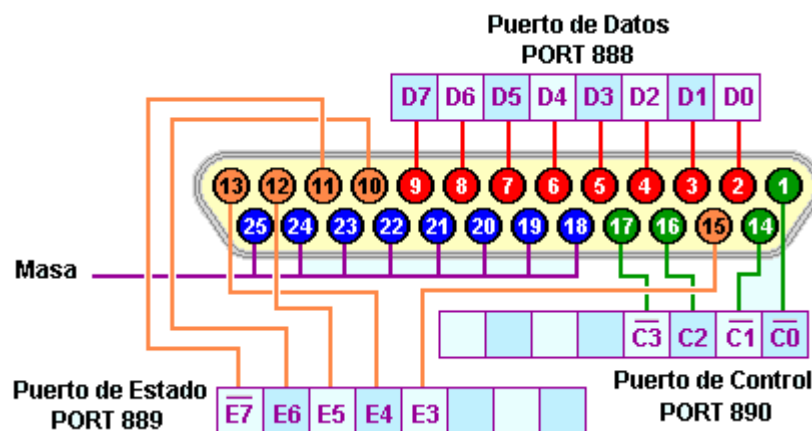


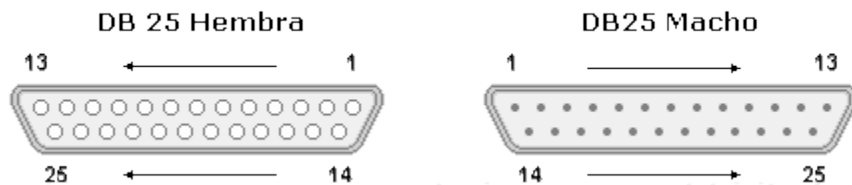
CONTROL DEL PUERTO PARALELO CON VISUAL BASIC 6.0

Hace ya mucho tiempo que realice unas prácticas con el puerto paralelo (LTP) y realice este pequeño tutorial donde se proporcionan los pasos básicos para controlarlo utilizando Visual Basic 6.0. Lo que se pretende es poder enviar pulsos eléctricos por medio de este puerto.

Empecemos con el puerto paralelo. Este es un conector DB25 obvio con 25 pines lo cuales están numerados y tienen una función específica cada uno de ellos. Desde el punto de vista hardware, el puerto es un conector hembra DB25 con doce salidas y cinco entradas, con 8 líneas de masa (GND). A continuación una imagen clara de las distribución del puerto paralelo.



Conectores DB25 Hembra y Macho.



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 computadora.

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.

Es importante mencionar que no se debe de conectar al revés las cosas en el puerto ya que existe la posibilidad de dañarlo.

El puerto paralelo maneja direcciones de memoria asignadas por el sistema operativo, estas direcciones están numeradas en código hexadecimal y son utilizadas normalmente para enviar ceros o unos a una impresora.

Por ejemplo: H378

Donde: H indica que es hexadecimal y 378 es el número hexadecimal.

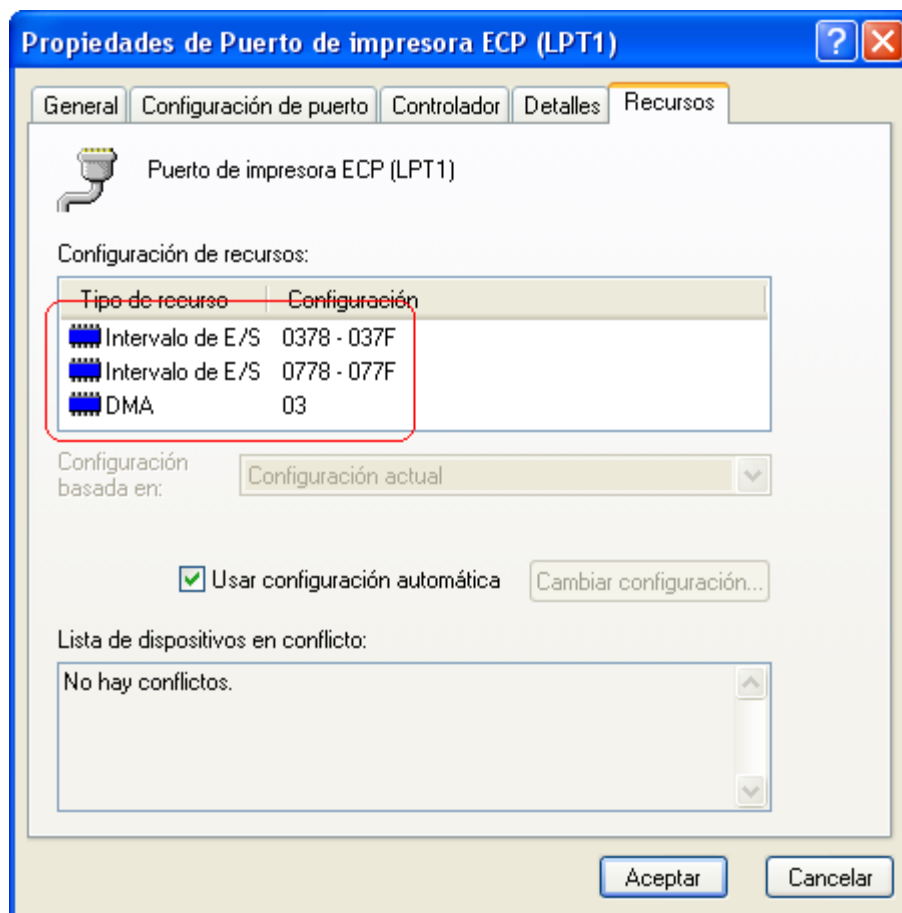
Una vez dicho y entendido esto se debe tener alguna forma de controlar esto desde el lenguaje visual Basic, desafortunadamente al ser un lenguaje de alto nivel visual no tiene la capacidad de controlar hardware por sí solo, este tipo de circunstancias se controlan con C o C++ ya que estos lenguajes tiene esa capacidad.

Afortunadamente para nosotros se puede aprovechar la característica de Windows de utilizar DLL (*Dynamic Linking Library* o Bibliotecas de Enlace Dinámico en español) y ya que alguien experto en programación nos ofrece una DLL en C++ para controlar este dispositivo nosotros podemos utilizarla en Visual Basic.

Esta DLL antes mencionada se llama inpout32 la cual tiene la capacidad de mandar y recibir señales o pulsos por el puerto paralelo. Se debe mencionar que la librería viene incluido con este tutorial y se tiene que copiar y pegar en la siguiente dirección: C:\WINDOWS\system32.

Una vez comprendido esto hablemos de las direcciones de memoria que anteriormente se menciona, la dirección de memoria que utiliza el puerto paralelo para funcionar se puede ver de la siguiente manera:

Botón INICIO/Mi PC/Click derecho del ratón /Propiedades /Hardware/ Administrador de dispositivos/Puertos (COM & LPT)/Doble clic en tu puerto LPT/Recursos. Y deberá aparecer el siguiente cuadro.



Como se ve en la imagen los primeros intervalos corresponden a las salidas del puerto paralelo es decir:

Pin	2	3	4	5	6	7	8	9
Memoria	378	379	37A	37B	37C	37D	37E	37F

El otro intervalo es para la lectura del puerto paralelo y funciona de la misma manera. Cabe mencionar que estos valores pueden variar ya que no son los mismos en todas las computadoras. Recomendando siempre checar las direcciones para no presentar errores ni confusiones, este simple y sencillo paso puede ahorrarte muchos dolores de cabeza.

Para la programación se necesita comprender la forma en la que serán enviados los datos, señales o pulsos al puerto y esto es de la siguiente forma.

Se envían datos binarios o decimales y dependiendo del valor son los pines utilizados:

Pin	2	3	4	5	6	7	8	9
Valor Binario	1	10	100	1000	10000	100000	1000000	10000000
Valor decimal	1	2	4	8	16	32	64	128

Para comprender un poco mejor la forma en la que se mandaran los datos ponemos los siguientes ejemplos:

1. Si queremos mandar señales por el Pin 2 y el 4 pues se hace la suma. $2 + 4 = 6$. Y ese valor es el que se envía por el puerto ya sea en binario “110” o decimal “6”.
2. Digamos que necesitamos el pin 5 y 9 la suma es: $8 + 128 = 136$. Y en binario es 10001000.
3. Si queremos prender todos los LED's conectados al puerto paralelo pues se pone: Binario “1111111” o Decimal “255”.

4. Si queremos apagar todos los LED's conectados al puerto paralelo pues se pone: "o"

NOTAS:

- Se puede ver que el pin de menos significativo es el numero 2 por lo que hay que empezar a contar por ahí.
- Normalmente el puerto paralelo está enviando señal por todos los pines.
- No es necesario enviar el dato binario ya que la DLL inpout32 los convierte basta con enviar el dato decimal.

Bueno ahora si es hora de comenzar con el software. Para poder enviar datos al puerto paralelo desde Visual Basic utilizando la DLL inpout32 se usa el comando Out y su sintaxis es la siguiente:

out (numero de pin desde donde se desea empezar a contar "cifra menos significativa", dato en decimal o binario).

Por ejemplo:

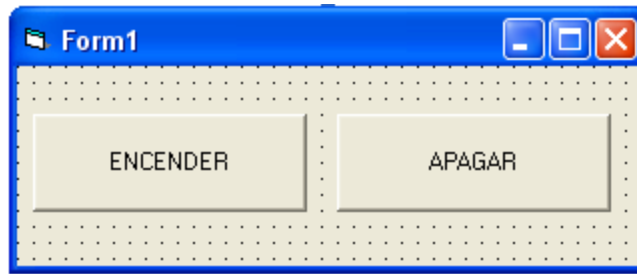
```
Out &H378, 255
```

Donde se omite el paréntesis. &H378 indica que el valor es hexadecimal y el dato de memoria antes mencionado. Se separa con ",". Y el 255 indica que pines serán usados en este caso todos encendidos.

Programación:

1. Creamos un nuevo proyecto EXE estándar en Visual Basic 6 en el formulario agregamos 2 botones.

Nombre	Texto
Command1	Encender
Command2	Apagar



2. Abrimos el código haciendo doble click en el botón y agregamos.

```
Private Sub Command1_Click()
```

```
    Out &H378, 0
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Out &H378, 1
```

```
End Sub
```

3. Agregamos un modulo nuevo “Proyecto/Agregar modulo” y en la ventana que se abre colocamos lo siguiente.

```
Public Declare Function Inp Lib "inpout32.dll" _
```

```
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Public Declare Sub Out Lib "inpout32.dll" _
```

```
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

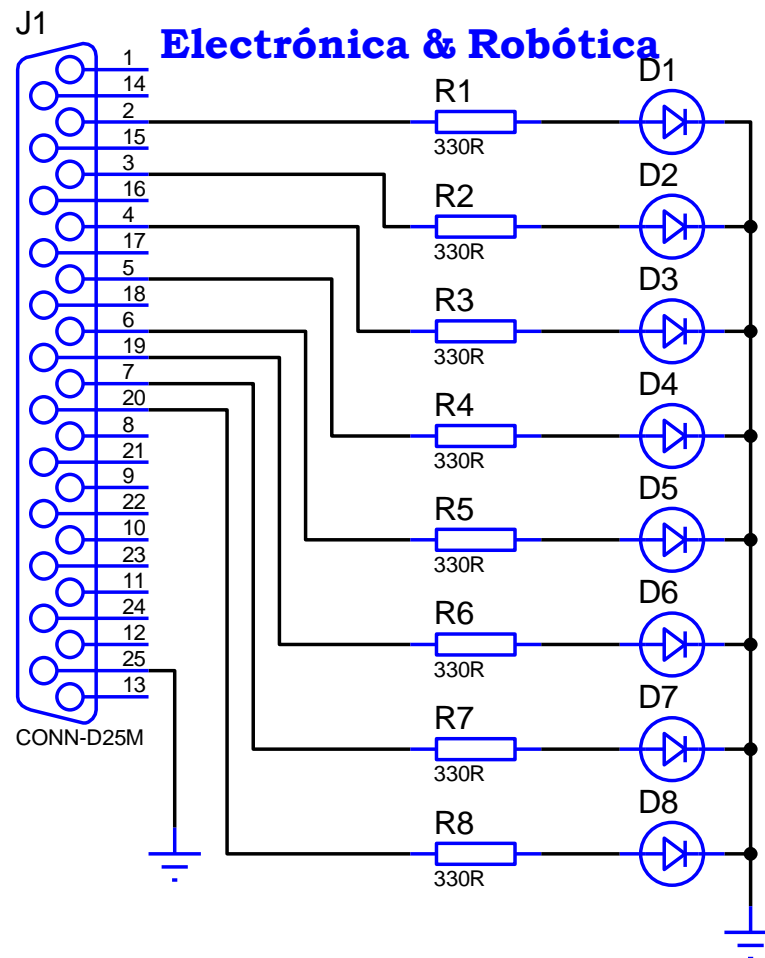
Es importante copiar y pegar la DLL inpout32 en C:\WINDOWS\system32 si ponemos el código en el modulo tal cual. Si te notifica que existe un error solo basta con buscar otra librería (inpout32) en internet, la reemplazan y listo.

4. Si se desea generar un archivo .EXE esto para poder ejecutar el proyecto en una computadora donde no se cuente con Visual Basic solo basta con ir “Archivo/ Generar .EXE”, lo guardan en algún lugar y listo.

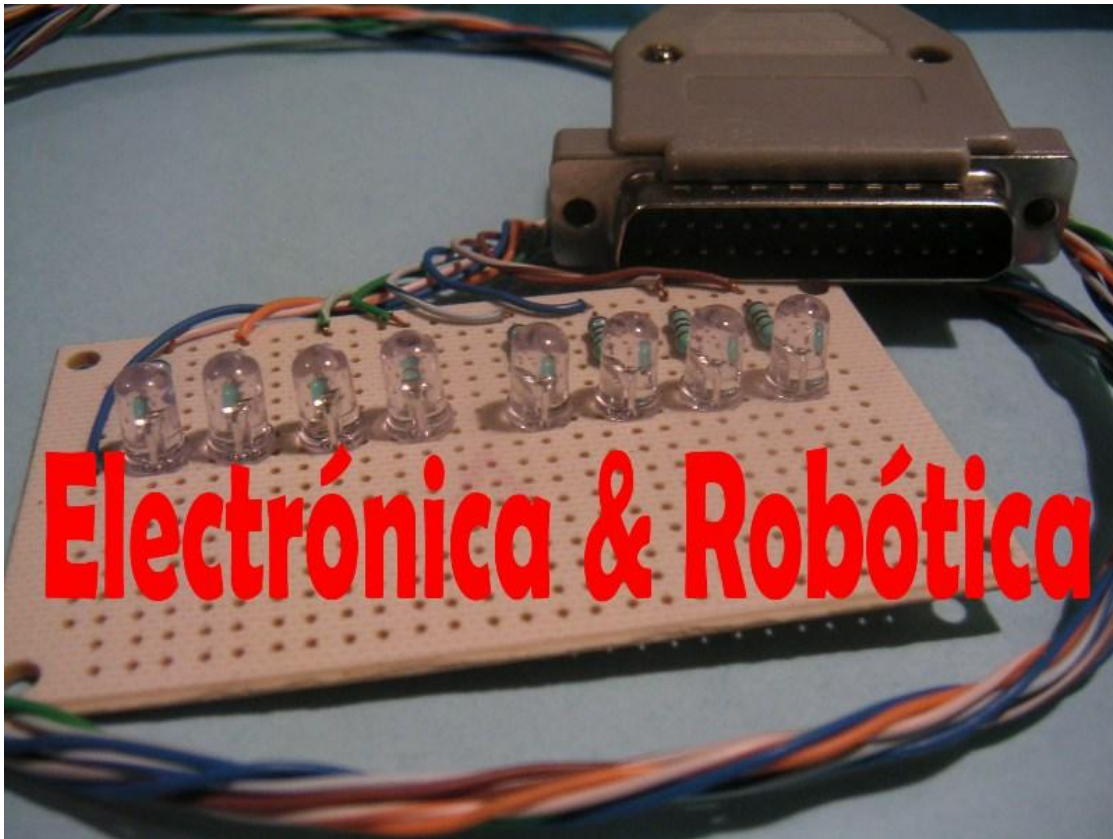
Circuito “Monitor de LED’s”:

Se trata de 8 LED’s conectados al Puerto de Datos (Pines del 2 al 9 del DB25) con su respectiva resistencia de 330Ω , esto para que en un futuro se pueda mandar datos y realizar sus propias pruebas con respecto a sus necesidades, lo hemos llamado monitor de LED’s ya que por este medio podemos ver los datos existente en el puerto LTP.

Dejo el esquemático por si se desea armar:



Aquí una imagen del monitor que arme en una vaquelita universal.



Es importante mencionar que si se desea trabajar con el puerto paralelo para activar cargas como relevadores, circuitos u otros tipos de dispositivos que consuman demasiada corriente es necesario utilizar un medio de acoplamiento (optoacopladores, transistores, arreglos darlington, etc), además de utilizar voltajes externos para evitar dañar el puerto LTP.