

## Como usar un drive de memoria USB

Hoy en día existe un dispositivo mucho mejor que los antiguos disquetes, que permite fácilmente almacenar varios megabytes de información, en un medio muy pequeño y bastante fácil de transportar. Se trata de los dispositivos de memoria USB. Existen diferentes sistemas, como el sistema flash o el sistema memory stick.

Los venden como un pequeño bolígrafo que se conecta en los puertos USB. Muchas cámaras digitales también usan ese sistema y con un cable USB puedes conectar tu sistema de ficheros (con tus fotos o películas) a tu PC. Estos sistemas funcionan perfectamente con el kernel Linux. Basta montarlos como si fueran discos SCSI, así que normalmente estarán en /dev/sda1, o si existen otros discos SCSI, estará /dev/sdb1 o /dev/sdc1. Las tarjetas de algunos fabricantes quedan visibles en /dev/sda4 en vez de /dev/sda1 y en otros casos /dev/sda. Será fácil descubrirlo viendo cual de esos dispositivos se puede montar: como root, usa algo así

```
mount /dev/sda1 /mnt
```

Montaje fácil del dispositivo

Para facilitar el montaje del dispositivo cada vez que se quiera utilizar, tan sólo hay que modificar el archivo /etc/fstab, insertando una línea que monte la unidad en algún directorio.

**Los pasos a seguir son:**

1. Iniciamos una sesión como root.
2. Nos movemos al directorio raíz con el comando
3. `cd /`

Aquí crearemos el directorio en el que montaremos nuestra llave de memoria. Este directorio lo puedes llamar como te plazca; yo lo he llamado jflash porque la marca de mi llave de memoria es Jet Flash. El comando sería:

```
mkdir jflash
```

4. Editamos el archivo /etc/fstab como root. Aquí deberemos añadir una línea para que el sistema reconozca el punto de montaje.
5. Esta línea es muy similar a la que contiene el punto de montaje del floppy. La línea debe quedar más o menos así:
6. `/dev/sda1 /jflash vfat defaults,user,noauto,showexec,umask=022 0 0`

la opción `user` permitirá que los usuarios normales puedan montar y desmontar el dispositivo. La opción `noauto` es para que no sea montado automáticamente cada vez que se arranca, pues puede que el dispositivo no esté conectado, o si se trata de una cámara fotográfica, también deberá estar encendida. La opción `umask=022` es para que los usuarios diferentes de `root` no puedan modificar el contenido. Puedes poner el valor de `umask` que quieras.

7. Tras realizar estos pasos, entraremos con un usuario normal, pinchamos la llave de memoria y montamos la unidad haciendo

8. `mount /jflash`

(o el nombre que le hayamos puesto).

9. Ya podemos acceder al contenido de la memoria USB, o a tus fotos, entrando en el directorio `jflash` (o el que hayas puesto) con el comando

10. `cd /jflash`

11. Antes de retirar la llave de memoria deberemos desmontar la unidad tecleando

12. `umount /jflash`

De esta manera desmontamos la unidad y podemos retirar la llave de memoria del puerto USB o desconectar la cámara.

### Montaje super-fácil con KDE

Si usas KDE, puedes montar, inspeccionar y desmontar tu llave de memoria USB de forma aún más fácil: creas un icono en tu escritorio, que te permitirá montar la memoria y ver su contenido con un simple "click" sobre él. (me imagino que en Gnome será algo similar, pero no lo conozco; si alguien sabe por favor diga :).

Usando Konqueror entra en tu área de escritorio. Por ejemplo, en mi caso como mi username es `villate` estará en:

`file:/home/villate/Desktop`

En el menú "Edit" del Konqueror tienes un submenú "Create new". Allí usarás la opción "Floppy Device" que es la que mas se parece al dispositivo que vas a montar. Te aparecerá una ventana con algunas secciones que te permiten definir el nombre y el icono que quieres usar, el dispositivo que vas a montar, el directorio donde se montará e los permisos que vas a dar a los usuarios.

Después basta pinchar el icono con el botón izquierdo del ratón. El botón

derecho del ratón muestra otras opciones que permiten, por ejemplo, remontar el dispositivo con autorización de escritura, o desmontarlo si ya está montado.

**COMANDO con el PUERTO PARALELO**

Patita	Señal	Función	Conexión	Registro	Bit	Estado
1	STB	Strobe	I/O	Control	0	Y
2	DO	Bit 0	O	Datos	0	N
3	D1	Bit 1	O	Datos	1	N
4	D2	Bit 2	O	Datos	2	N
5	D3	Bit 3	O	Datos	3	<b>N</b>
6	D4	Bit 4	O	Datos	4	N
7	D5	Bit 5	O	Datos	5	N
8	D6	Bit 6	O	Datos	6	N
9	D7	Bit 7	O	Datos	7	N
10	ACK	Acknowledge	I	Status	6	N
11	BSY	PrinterBusy	I	Status	7	Y
12	PE	PaperEnd	I	Status	5	N
13	SEL	PrinterSelect	I	Status	7	N
14	AUTOLF	AutomLineFeed	I/O	Control	4	N
15	ERR	Error	I	Status	3	N
16	INIT	InicialicePrinter	I/O	Control	2	N
17	SELIN	SelectPrinter	I/O	Control	3	Y
18 - 25	GND	Ground				

Datos BASE	Status BASE + 1	Control BASE + 2
3BCH	3BDH	3BEH
378H	379H	37AH
278H	279H	27AH

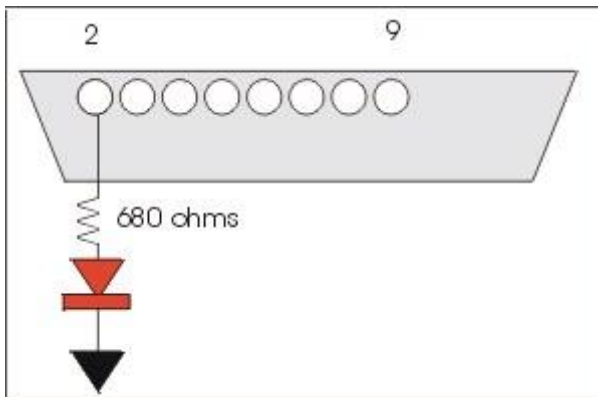
- El puerto paralelo nos permite manejar una cantidad de artefactos y dispositivos
- En un programa de la PC podemos además de conectar y desconectar la ejecución de
  - una orden, controlar tiempos y secuencia de esas ordenes
- En la salida del puerto paralelo tenemos 8 bornes que sin ningún agregado entrega
  - una tensión que puede estar entre 0 y 3.5 - 4.8 voltios según la computadora
- Si instalamos un programa de la familia de los Basic por ejemplo Quick Basic
  - podemos trabajar en la programación
- Los puertos paralelos se denominan **378** o **278** o **3bc** según la PC
- 
- Los terminales del puerto que nos interesa son los terminales 2 a la 9 y se llaman
 

Terminal	2	3	4	5	6	7	8	9
Salida	1	2	4	8	16	32	64	128
- Los números llamados salidas son las denominaciones de los BIT que maneja el puerto paralelo
- Para hacer visible las ordenes que demos desde la PC armemos el siguiente circuito

Soldamos un cable en cada terminal **cuidando que queden bien aislados entre si y con respecto a chasis**

- En el extremo libre del los cables soldamos una resistencia de 680 ohms y conectamos

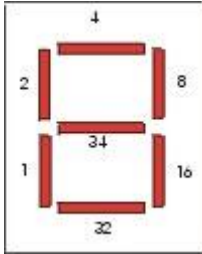
- un LED si es posible rectangular y el extremo libre del led lo conectamos a chasis



- Abrimos el programa Basic , escribimos el nombre del puerto Ej.: **out &h378,1**
- y oprimimos la tecla F5 que ejecuta el programa
- Inmediatamente se encenderá el led conectado a la salida 2 o BIT 1
- Si ponemos **out &h378,19** se encenderán los led de los terminales 2-3 y 6
- Para comprenderlo mejor escribamos **out &h378,1+2+16** lo que nos da la suma de los
- BIT encendidos

***Para no mezclar datos manejemos en lo sucesivo el numero del BIT***

- La sugerencia de pedir inicialmente la instalación de LEDS rectangulares es por esto
- Alineemos los Leds formando un rectángulo vertical usando siete es decir dos
- a la izquierda, dos a la derecha, uno arriba uno abajo y el séptimo en el centro
- Tenemos un DISPLAY NUMÉRICO de 7 segmentos



- Si escribimos el comando **out &h378,8+16 ó 24** veremos escrito el numero 1
- Si escribimos **out &h378,4+8+64+1+32 ó 109** veremos escrito el numero 2
- Si escribimos **out &h378,1+2+4+8+16+32+64 ó 127** veremos escrito el numero 8
- Si escribimos **out &h378,1+2+4+8+16+64 ó 95** veremos escrita la letra A
- Si escribimos **out &h378,2+1+32+16+8 ó 59** veremos escrita la letra U
- Es fácil determinar los BIT a encender para hacer visible un numero o letra
- Avancemos un paso en lo que es posible realizar
- **Si instalamos un secuenciador y un control de tiempo entre cada orden**
- **podemos leer un numero de varios dígitos o una palabra**

Existen dos formas para ello

- Escribimos **out &h378,24**
- Luego **sleep 1**
- Luego **out &h378,109**
- Luego **sleep 1**
- En este caso se vera el numero 1 y un segundo después el numero 2 (el comando sleep
- controla tiempo de a un segundo si ponemos sleep 2 se vera el segundo numero
- 2 segundos después )

La segunda forma es

- **out &h378,24**

- **for a# = 1 to 400000 step 1 : next a#**
- **out &h378,109**
- El comando for genera un bucle que detiene el programa hasta que no sume de 1 a 400000
- un numero por vez ( si step es 2 suma de a dos números por vez )
- Cambiando el numero 400000 por uno mayor o menor se aumenta o disminuye la espera

Además queremos que esto se repita N cantidad de veces, entonces escribimos

- **for z# =1 to 50 step 1**
- **out &h378,24**
- **for a# = 1 to 400000 step 1 : next a#**
- **out &h378,109**
- **next z#**
- Ahora el comando anterior esta dentro de un bucle y se repetirá 50 veces antes de
- desconectarse, el numero 50 indica la cantidad de veces que se repite

En una maquina de 266 Mc de velocidad contar el numero **400000** demora aprox. 1 segundo

Si deseamos leer HOLA programamos

- **out &h378,1+2+18+64+16 ó 91** (letra H)
- **sleep 1**
- **out &h378,1+2+4+8+16+32 ó 63** (letra O)
- **sleep 1**
- **out &h378,1+2+32 ó 35** (letra L)
- **sleep 1**
- **out &h378,1+2+4+8+16+64 ó 125** (letra A)
- **sleep 1**

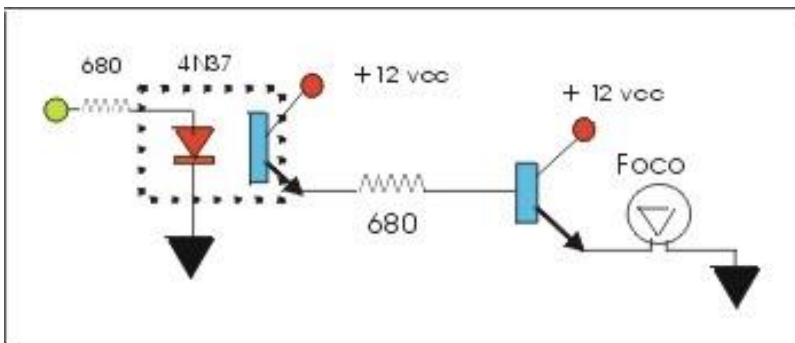
Para repetirlo indefinidamente escribimos

- **do**
- **out &h378,1+2+18+64+16 ó 91** (letra H)
- **sleep 1**
- **out &h378,1+2+4+8+16+32 ó 63** (letra O)
- **sleep 1**

- `out &h378,1+2+32 ó 35` (letra L)
- `sleep 1`
- `out &h378,1+2+4+8+16+64 ó 125` (letra A)
- `sleep 1`
- `loop`

Los comandos `do` y `loop` repiten los mandos encerrados entre ellos

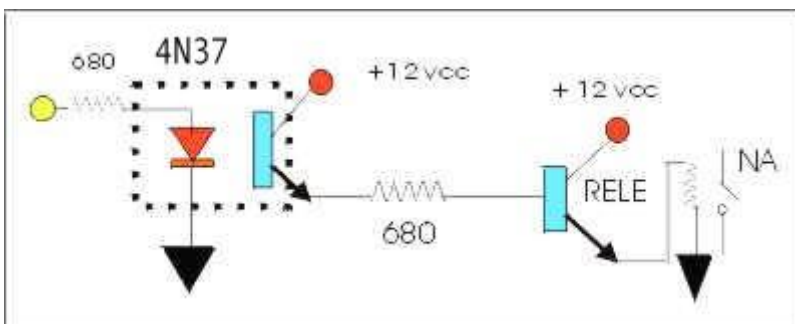
para detener el programa se oprimen las teclas `CONTROL` Y `PAUSA` simultáneamente



El grafico mostrado acopla la salida del puerto paralelo a una carga de C Continua en este

caso un foco. El circuito permite manejar cargas de hasta 1 Amper y 10-12 voltios

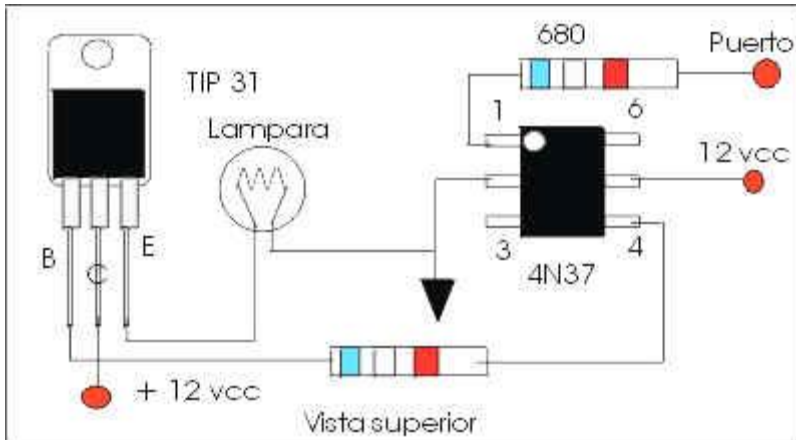
El optoacoplador es 4N47 y el transistor TIP 31





En este caso la salida es a un rele cuyo contacto puede ser NA-NC O ambos y conectar un artefacto de 220 V c Alternada con los cuidados necesarios

**Para los que se confunden las conexiones el circuito en diagrama**



**El optocoplador puede ser 4N35-4N36-4N37**

Por si interesa utilizar otro lenguaje de programación veamos en C

```
#include <dos.h>
void main ()
{
int puerto = 0x378;
int control = 0;
clrscr();
printf("se inicia el comando de puerto");
outportb(puerto,control +2);
sleep (3);
outportb(puerto,control + 8);
}
```

Cambiando el numero después de control cambiamos el bit o la combinación de ellos que deseamos

Sleep controla segundos en este caso 3

Los datos en rojo lo podemos repetir con distintos bit y distintos tiempos

Si deseamos tiempos mas cortos reemplazamos sleep por delay que mide

milésimos de  
segundos

Si controlan el lenguaje V Basic 5 mostramos un programa  
Los command son lo ejecutores  
los shape círculos rojos que se encienden  
Y si no lo tienen es necesario bajar un archivo que se carga en  
Windows/System  
o system 32 y hace que el puerto paralelo responda a ordenes desde V  
Basic

```
# if win32 then
```

```
Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32"  
(byvalPortAddress As Integer) As Integer  
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal portaddress  
As Integer, ByVal value As Integer)  
#End If
```

```
Private Sub Command1_Click()  
Out &H378, 1  
Shape1.FillStyle = 0  
End Sub
```

```
Private Sub Command2_Click()  
Out &H378, 2  
Shape2.FillStyle = 0  
End Sub
```

```
Private Sub Command3_Click()  
Out &H378, 4  
Shape3.FillStyle = 0  
End Sub
```

```
Private Sub Command4_Click()  
Out &H378, 8  
Shape4.FillStyle = 0
```

End Sub

```
Private Sub Command5_Click()  
Out &H378, 16  
Shape5.FillStyle = 0  
End Sub
```

```
Private Sub Command6_Click()  
Out &H378, 32  
Shape6.FillStyle = 0  
End Sub
```

```
Private Sub Command7_Click()  
Out &H378, 64  
Shape7.FillStyle = 0  
End Sub
```

```
Private Sub Command8_Click()  
Out &H378, 128  
Shape8.FillStyle = 0  
End Sub
```

```
Private Sub Command9_Click()  
Out &H378, 0  
Shape9.FillStyle = 0  
Shape1.FillStyle = 1  
Shape2.FillStyle = 1  
Shape3.FillStyle = 1  
Shape4.FillStyle = 1  
Shape5.FillStyle = 1  
Shape6.FillStyle = 1  
Shape7.FillStyle = 1  
Shape8.FillStyle = 1  
End Sub
```

```
Private Sub Form_Load()  
a# = Inp(&H378)
```

Print a#

End Sub

- Una vez copiado en el VB5 el programa abrir en la pantalla de proyectos (arriba derecha)
- la pantalla de objetos haciendo un clic sobre ella
- Ampliar la pantalla y cargar 9 controles command a la izquierda de la pantalla y de arriba a abajo
- Luego colocar 9 controles shape a la derecha de cada comando command
- Los command se llaman al instalarlos command 1-2-3 hasta 9
- Si se quiere cambiar el nombre para indicar una aplicación
- Como Ej. en la pantalla de objetos hacemos clic sobre el command 1
- En la pantalla de propiedades (derecha abajo) aparece una pantalla con el titulo "command 1
- command button"
- Entre las propiedades elegir caption
- Si hacemos doble clic sobre caption se pintara el nombre actual del botón, cambiarlo por el nombre deseado
- Podemos cambiar los colores y las formas de los shape haciendo clic sobre ellos aparecen las propiedades
- Elegimos fiillcolor hacemos clic y elegimos el color que tendrá cuando se conecte el bit del puerto

### ***ARCHIVOS DE CONTROL PARA BAJAR***

*Sin uno de estos archivos VB no reconoce el puerto paralelo*

Puerto paralelo 16 bit

Puerto paralelo 32  
bit