



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Instalación del escaner Epson Perfection 1240U (USB) (26806 lectures)

Per **Carlos Cortes Cortes**, [carcoco](http://bulma.net/~carcoco/) (<http://bulma.net/~carcoco/>)

Creado el 29/11/2001 09:01 modificado el 29/11/2001 09:01

El **Epson Perfection 1240U** es un scanner USB, que funciona a las mil maravillas con **Linux**, aunque previamente tendremos que configurarlo, algo realmente sencillo y rápido si lees estas sencillas instrucciones

He dividido el artículo en 3 partes:

- Una primera donde describo los pasos a seguir en distribuciones con todo a punto "*Out of the box*".
- Una segunda para la mayoría de las *distribuciones* más o menos actualizadas.
- y otra donde explico *todos* los pasos lógicos a seguir para poder escanear partiendo de cero.

NOTA: El artículo explica cómo configurar este modelo concreto de Scanner USB en linux (**Epson Perfection 1240U**), otros escaners USB se configuran de forma totalmente equivalente.

NOTA ESPECIAL: El nuevo modelo **Epson Perfection 1250** NO es igual que el 1240U, ya que aunque es USB, utiliza el chip LM983x, este chip usa una especie de puerto paralelo sobre USB.

No compres este modelo porque el fabricante no da soporte para el mismo, aunque parece por el número que es igual al 1240U, no tiene nada que ver, ni en precio ni en rendimiento ni en calidad.

Please note the Epson Perfection 1250, unlike other Epson scanners, does not use the Epson Scanner Language and is not supported (it is reported to use a LM983x chipset, and may eventually be supported by a backend for such scanners)

Configuración Mega Express:

En este caso, lo único que tendremos es que instalar los paquetes necesarios, para el soporte del USB y del escaner: **usbmgr**, **sane**, **xsane** y **Gimp** (opcional):

Una vez instalados, los pasos a seguir serán tan sencillos como:

1. Encender el escaner
2. Ejecutar XSane o Gimp
3. Escanear

Configuración Express:

Se supone que tenemos instalado los siguientes paquetes: **usbmgr**, **sane** y **xsane** (**Gimp** opcional).

Modificar el fichero **/etc/sane.d/epson.conf**, de forma que tenga una única entrada para el escaner USB, el resto estarán comentadas con el carácter almohadilla (#):

BULMA: Instalación del escaner Epson Perfection 1240U (USB)



```
# here are examples for how to configure the EPSON backend
# scsi EPSON
# option color_correction 1 2 3 4 5 6 7 8 9
#
#pio 0x278
#pio 0x378
#pio 0x3BC
#scsi EPSON
# /dev/scanner
usb /dev/usb/scanner0
```

Podemos ver que tenemos este dispositivo creado en el directorio `/dev`:

```
$ ls -l /dev/usb/scanner0
crw-rw-rw- 1 root root 180, 48 Jan 19 2001 /dev/usb/scanner0
```

Para comprobar que tenemos en marcha correctamente el bus USB en Linux y que se ha detectado correctamente el scanner, ejecutaremos:

```
$ cat /proc/bus/usb/devices
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 1.00 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 0.00
S: Product=USB UHCI Root Hub
S: SerialNumber=dc00
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 8 Iv1=255ms
T: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 2 Spd=12 MxCh= 0
D: Ver= 1.00 Cls=ff(vend.) Sub=ff Prot=ff MxPS=64 #Cfgs= 1
P: Vendor=04b8 ProdID=010b Rev= 1.04
S: Manufacturer=EPSON
S: Product=Perfection1240
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 2mA
I: If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
E: Ad=81(I) Atr=02(Bulk) MxPS= 64 Iv1= 0ms
E: Ad=02(O) Atr=02(Bulk) MxPS= 64 Iv1= 0ms
```

Donde podemos comprobar que el escaner es detectado perfectamente.

Si no lo hubiera detectado o el escaner estuviera apagado obtendríamos tan solo esto:

```
$ cat /proc/bus/usb/devices
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 1.00 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 0.00
S: Product=USB UHCI Root Hub
S: SerialNumber=dc00
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 8 Iv1=255ms
```

Vamos a ver si ya funciona el scanner:

```
$ scanimage -L
```

```
No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
find-scanner tool (if appropriate). Please read the documentation which came
with this software (README, FAQ, manpages).
```

Lo que nos está diciendo es que no ha detectado ningún escaner, pero si acabamos de ver que lo había detectado. El problema es que nos falta cargar el módulo del escaner en el kernel. Como root ejecutamos :

```
# modprobe scanner
```



Pero sigue sin funcionar ...

```
$ scanimage -L
```

No scanners were identified. If you were expecting something different, check that the scanner is plugged in, turned on and detected by the find-scanner tool (if appropriate). Please read the documentation which came with this software (README, FAQ, manpages).

El problema, es que tenemos que pasarle un par de parametros al cargar el modulo: el número de vendedor y el código de producto. Esta información la hemos obtenido previamente, en mi caso **Vendor=04b8 ProdID=010b**.

Por lo tanto será: **modprobe scanner vendor=0x04b8 product=0x010b**

Antes deberemos descargar el modulo, también como root:

```
# rmmod scanner
```

Para volverlo a cargar, pero esta vez pasándole los parametros:

```
# modprobe scanner vendor=0x04b8 product=0x010b
```

Si tenemos instalado y correctamente configurado **usbmng**, que es un *daemon* que se encarga de cargar y descargar los modulos USB del kernel, nos ahorraremos este último paso de la carga del modulo *scanner*.

```
$ lsmod
Module                Size  Used by
scanner              4768   0 (unused)
ppp_deflate           40352   0 (autoclean)
bsd_comp               4080   0 (autoclean)
...
```

Podemos ahorrarnos el teclear estos parametros cada vez que cargamos el modulo, con tal solo modificar el fichero */etc/modules.conf*, añadiendo una entrada como la que sigue:

```
options scanner          vendor=0x04b8 product=0x010b
```

Como podemos ver, el escaner esta plenamente funcional en Linux gracias al proyecto **Sane**:

```
scanimage -L
device `epson:/dev/usb/scanner0' is a Epson Perfection1240 flatbed scanner
```

```
$ scanimage -T
scanimage: scanning image of size 2544x3509 pixels at 1 bits/pixel
scanimage: acquiring gray frame, 1 bits/sample
scanimage: reading one scanline, 318 bytes...    PASS
scanimage: reading one byte...                PASS
scanimage: stepped read, 2 bytes...            PASS
scanimage: stepped read, 4 bytes...            PASS
scanimage: stepped read, 8 bytes...            PASS
scanimage: stepped read, 16 bytes...           PASS
scanimage: stepped read, 32 bytes...           PASS
scanimage: stepped read, 64 bytes...           PASS
scanimage: stepped read, 128 bytes...          PASS
scanimage: stepped read, 256 bytes...          PASS
scanimage: stepped read, 512 bytes...          PASS
scanimage: stepped read, 511 bytes...          PASS
scanimage: stepped read, 255 bytes...          PASS
scanimage: stepped read, 127 bytes...          PASS
scanimage: stepped read, 63 bytes...           PASS
scanimage: stepped read, 31 bytes...           PASS
scanimage: stepped read, 15 bytes...           PASS
scanimage: stepped read, 7 bytes...            PASS
scanimage: stepped read, 3 bytes...            PASS
```

Si deseamos obtener más información sobre el scanner, tendremos que ejecutar:



```
$ SANE_DEBUG_EPSON=255 scanimage -L
[sanei_init_debug]: Setting debug level of epson to 255.
[epson] sane_init: sane 1.0.6
[epson] sane_init, ># scsi EPSON<
[epson] sane_init, >usb /dev/usb/scanner0 <
[epson] sane_init, >usb /dev/usb/scanner0 <
[epson] attach(usb /dev/usb/scanner0)
[epson] SANE Epson Backend v0.3.01ek - 2001-11-08
[epson] attach(usb /dev/usb/scanner0)
[epson] attach: opening usb /dev/usb/scanner0
[epson] send buf, size = 2
[epson] buf[0] 1b .
[epson] buf[1] 40 @
[epson] receive buf, expected = 1, got = 1
[epson] buf[0] 06 .
[epson] get_identity_information()
[epson] send buf, size = 2
...
```

Bueno y eso es todo ... ahora solo tendremos que ejecutar el comando **xsane** y podremos empezar a escanear.

```
$ xsane
```

También se puede configurar el **Gimp**, para que utilice al **xsane** como un extra, de forma que lo que escaneemos pase directamente al Gimp, sin necesidad de guardar la imagen en un fichero.

```
$ ln -s /usr/local/bin/xsane ~/.gimp-1.2/plug-ins/
```

Ahora para arrancar el **xsane** desde el **GIMP**, iremos a la opción *File/Acquire*.

Configurando el Scanner desde Cero

El primer paso será una visita al proyecto [Sane](#)⁽¹⁾ (Scanner Access Now Easy), donde nos informaremos si nuestro escaner esta soportado por linux, (realmente si el fabricante ha facilitado documentación lo estará).

Nos bajaremos la última versión del mismo en el momento de escribir esto SANE-1.0.6 (del 05.11.2001). Podemos obtener el paquete (rpm, deb) de nuestra distribución e instalar directamente el binario. Pero también podemos bajarnos el fuente, compilarlo e instalarlo de la forma habitual: (*./configure; make; make install*):

Necesitamos :

sane-backends-1.0.6.tar.gz, En este archivo es donde estan las librerías principales del sane y todo el código de los diferentes modelos de escaner.

Si conseguimos un nuevo backend ("SANE Epson Backend v0.2.14 - 2001-11-11") para nuestro modelo concreto de escaner y es mas actual que el incluido en el Sane ("SANE Epson Backend v0.2.12 - 2001-10-28") , es aqui donde tendremos que usarlo para parchear el sane.

Por ejemplo, **sane_epson_20011111.tar.gz**, contiene:

backend/epson.c

backend/epson.h.

sane-frontends-1.0.6.tar.gz, aqui encontramos las aplicaciones **xscanimage** y **xcam**, que son 2 programas que permiten usar el escaner, aunque es preferible usar el mucho más potente **xsane**.

Estos son los escaner soportados por el **sane epson backend**: (+ info en linux/driver/usb/scanner.h y backend/epson.c)

- (0x04b8, 0x0103) - Perfection 610
- (0x04b8, 0x0101) - Perfection 636
- (0x04b8, 0x010c) - Perfection 640
- (0x04b8, 0x0104) - Perfection 1200U/Photo
- (0x04b8, 0x010b) - Perfection 1240
- (0x04b8, 0x0112) - Perfection 2450 (GT-9700 en el mercado japones)
- (0x04b8, 0x0106) - Stylus Scan 2500
- (0x04b8, 0x0107) - Expression 1600



- (0x04b8, 0x010a) - Expression 1640SU/Photo
- (0x04b8, 0x0110) - Perfection 1650
- (0x04b8, 0x010e) - Expression 1680

Vuelvo a repetir que **NO** esta soportado el **Perfection 1250**, porque no usa el lenguaje estandar de epson para escaners USB, al llevar un nuevo chip.

Tenemos tres excelentes tablas donde podremos consultar si existe y el estado del soporte de nuestro hardware:

- Sane BackEnds: <http://panda.mostang.com/sane/sane-backends.html>⁽²⁾
- Listado Usb Scanners on Linux: <http://www.buzzard.org.uk/jonathan/scanners-usb.html>⁽³⁾
- Listado de Scanner puerto paralelo y SCSI: <http://www.buzzard.org.uk/jonathan/scanners.html>⁽⁴⁾

Kernel

Vamos a suponer que nuestro kernel no tiene soporte para USB, por lo que tendremos que recompilarlo, con las siguientes opciones:

(Esto es valido tanto para las series 2.2.X, como para las series 2.4.X).

```
# USB support
CONFIG_USB=y
CONFIG_USB_DEBUG=y
CONFIG_USB_DEVICEFS=y
CONFIG_HOTPLUG=y
CONFIG_USB_UHCI=m
CONFIG_USB_SCANNER=m
```

En este caso le decimos que compile como parte del kernel el soporte USB, y que compile como un modulo independiente el soporte para el scanner.

Una vez tenemos soporte USB en el kernel, utilizaremos el paquete **usbmgr** para controlar este soporte, el fichero principal se encuentra en **/etc/usbmgr/usbmgr.conf**, y debe contener la especificación para nuestro modelo concreto de escaner (si no estuviera la añadimos):

```
...
### SCANNER

# (scanner)
# Perfection 636U , 636Photo [Epson]
vendor 0x04b8 product 0x0101 module scanner
# Perfection 610 [Epson]
vendor 0x04b8 product 0x0103 module scanner
# Perfection 1200U , 1200Photo [Epson]
vendor 0x4b8 product 0x104 module scanner
# Stylus Scan 2500 [Epson]
vendor 0x4b8 product 0x106 module scanner
# Expression 1600 [Epson]
vendor 0x4b8 product 0x107 module scanner
# Perfection 1640SU, Perfection 1640SU Photo [Epson]
vendor 0x4b8 product 0x10a module scanner
# Perfection 1240U [Epson]
vendor 0x4b8 product 0x10b module scanner
# Perfection 640U [Epson]
vendor 0x4b8 product 0x10c module scanner
# Expression 1680 [Epson]
vendor 0x4b8 product 0x10e module scanner
```

Modificaremos el fichero **/etc/modules.conf**, añadiendole los parametros que necesita el módulo *scanner* cuando se carga:

```
options scanner          vendor=0x04b8 product=0x010b
```

Crearemos los dispositivos relativos al escaner en el directorio **/dev/**:

```
# cd /dev
```



```
# mknod /dev/usbscanner0 c 180 48
# chmod 666 /dev/usbscanner0
```

Ahora modificaremos el fichero `/etc/sane.d/epson.conf`, tal y como se explica en la sección Configuración Express.

```
usb /dev/usb/scanner0
```

El siguiente paso es montar el sistema de archivos USB `usbdevfs`:

```
# mount -t usbdevfs none /proc/bus/usb/
```

Podemos comprobar si se ha montado correctamente el sistema USB, con el comando `mount`:

```
$ mount
/proc/bus/usb on /proc/bus/usb type usbdevfs (rw)
```

En el registro del kernel, habra quedado registrada toda esta actividad, tal y como podemos ver en `/var/log/messages`:

```
Nov 28 23:03:47 linuxalcoy kernel: hub.c: USB new device connect
on bus1/1, assigned device number 3
Nov 28 23:03:47 linuxalcoy kernel: usb.c: kmalloc IF c7649580, numif 1
Nov 28 23:03:47 linuxalcoy kernel: usb.c: new device strings: Mfr=1,
Product=2, SerialNumber=0
Nov 28 23:03:47 linuxalcoy kernel: usb.c: USB device number 3
default language ID 0x409
Nov 28 23:03:47 linuxalcoy kernel: Manufacturer: EPSON
Nov 28 23:03:47 linuxalcoy kernel: Product: Perfection1240
Nov 28 23:03:47 linuxalcoy kernel: scanner.c: probe_scanner:
User specified USB scanner -- Vendor:Product - 4b8:10b
Nov 28 23:03:47 linuxalcoy kernel: usb.c: usbscanner driver claimed
interface c7649580
Nov 28 23:03:47 linuxalcoy usbmgr[2450]: vendor:0x4b8 product:0x10b
Nov 28 23:03:47 linuxalcoy usbmgr[2450]: class:0xf subclass:0xff
protocol:0xff
Nov 28 23:03:47 linuxalcoy usbmgr[2450]: USB device is matched
the configuration
Nov 28 23:03:47 linuxalcoy usbmgr[2450]: "scanner" was loaded
```

Ahora ya podemos comprobar si el escaner es detectado en el bus USB y si esta todo listo para se usado a través del sane, tal y como hemos visto anteriormente:

```
$ cat /proc/bus/usb/devices
...
P: Vendor=04b8 ProdID=010b Rev= 1.04
S: Manufacturer=EPSON
S: Product=Perfection1240
...

$ scanimage -L
device `epson:/dev/usb/scanner0' is a Epson Perfection1240 flatbed scanner
```

Y si todo a ido bien, ya podemos usar **xscanimage**, **XSane**, **Gimp**, **QuiteInsane** o el nuevo **EPSON Kowa's IScan! for Linux**, que es un programa creado por la **Epson Kowa Corporation**, para escanear en Linux usando exclusivamente los escaners USB de esta empresa:

- Perfection 1240U
- Perfection 1640SU
- Perfection 1650/Perfection 1650 Photo
- Perfection 2450 Photo
- Expression 1600
- Expression 1680
- Expression 1640XL



Esta disponible el código fuente, aunque el **IScan! for Linux** necesita para poder trabajar la instalación y configuración previa del **Sane**.

Se puede usar al igual que el **XSane** como un plug-ins de Gimp:

```
$ ln -s /usr/bin/iscan ~/.gimp-1.2/plugin-ins/
```

Por último comentar la gran potencia y calidad del **Sane**, que permite entre otras cosas, el **escanar** a través de la **red**, o escanear directamente desde la **línea de comandos** (ideal para, por ejemplo, crear scripts personalizados).

Referencias adicionales:

Sane Epson: <http://www.freecolormanagement.com/sane/>⁽⁵⁾

Suse 7.1 Epson USB scanner Howto: <http://www.freecolormanagement.com/sane/suse71.html>⁽⁶⁾

XSane: <http://www.xsane.org/>⁽⁷⁾

EPSON Kowa's IScan! for Linux: http://www.epkowa.co.jp/english/index_e.html⁽⁸⁾

QuiteInsane: <http://sourceforge.net/projects/quiteinsane>⁽⁹⁾

Linux scanner how-to. For beginners: <http://www.arrakis.es/~fsanta/sane/howto.html>⁽¹⁰⁾

Usb Scanner on GNU / Linux Mandrake 8.1: <http://michaelo.free.fr/linux/mdk81/usbscanner.php3>⁽¹¹⁾

Using a USB Scanner: <http://www.linux-usb.org/USB-guide/x722.html>⁽¹²⁾

Como scanear con el AGFA SNAPSCAN 1212 USB: <http://bulma.net/body.phtml?nIdNoticia=669>⁽¹³⁾

Usbmgr: <http://www.wonder.ne.jp/~shuu/usbmgr/>⁽¹⁴⁾

Fichero de configuración del usbmgr: <http://www.wondernetworkresources.com/staff/shuu/linux/usbmgr/>⁽¹⁵⁾

Linux USB Guide: <http://linuxusbguide.sourceforge.net/>⁽¹⁶⁾

--

Carlos Cortes(aka carcoco)

http://bulma.net/todos.phtml?id_autor=132⁽¹⁷⁾

Lista de enlaces de este artículo:

1. <http://panda.mostang.com/sane/>
 2. <http://panda.mostang.com/sane/sane-backends.html>
 3. <http://www.buzzard.org.uk/jonathan/scanners-usb.html>
 4. <http://www.buzzard.org.uk/jonathan/scanners.html>
 5. <http://www.freecolormanagement.com/sane/>
 6. <http://www.freecolormanagement.com/sane/suse71.html>
 7. <http://www.xsane.org/>
 8. http://www.epkowa.co.jp/english/index_e.html
 9. <http://sourceforge.net/projects/quiteinsane>
 10. <http://www.arrakis.es/~fsanta/sane/howto.html>
 11. <http://michaelo.free.fr/linux/mdk81/usbscanner.php3>
 12. <http://www.linux-usb.org/USB-guide/x722.html>
 13. <http://bulma.net/body.phtml?nIdNoticia=669>
 14. <http://www.wonder.ne.jp/~shuu/usbmgr/>
 15. <http://www.wondernetworkresources.com/staff/shuu/linux/usbmgr/>
 16. <http://linuxusbguide.sourceforge.net/>
 17. http://bulma.net/todos.phtml?id_autor=132
-

E-mail del autor: carcoco _ARROBA_ gmail.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1031>