



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

## Pon un pad de la playstation en tu linux (46664 lectures)

Per **Javier Ferrer Gómez**, [EICoco](http://) (<http://>)

Creado el 09/04/2002 21:36 modificado el 09/04/2002 21:36

*El propio kernel del linux trae consigo una colección de drivers para infinidad de joysticks y pads, en este artículo veremos cómo podemos poner a punto nuestro joystick tomando como ejemplo práctico el fantástico pad de la PSX.*

### Montaje del pad

Primero de todo, para este tipo de pad se necesita realizar un montaje para poder conectarlo al PC a través del puerto paralelo, pero no os asusteis, no se requieren grandes conocimientos de electrónica para ello, tan solo saber soldar 4 cables...El montaje para este y otros pads lo podeis encontrar en la documentación del kernel o para más claridad y en castellano, pasaros por [emulatronia](#)<sup>(1)</sup>.

### Seleccionar los módulos apropiados

Bien, una vez que ya tenemos el pad bien montado, llega la parte software. Necesitamos los siguientes módulos:

- **parport y parport\_pc**: Soporte para puerto paralelo, los encontrarás en la sección *Parallel port support/Parallel port support* de la configuración del kernel.
- **joydev**: Este módulo es requerido por cualquier joystick, lo podemos encontrar en la sección *Input core support/joystick support* de la configuración del kernel.
- **gamecon**: Este es el módulo que contiene los drivers del pad de la PSX, lo podemos encontrar en la sección *character devices/joysticks/Multisystem, NES, SNES, N64, PSX joysticks and gamepads* de la configuración del kernel.

### Preparar el sistema

Para que las aplicaciones puedan usar el joystick, tienes que crear los nodos apropiados en /dev, en caso de que ya existan en un kernel superior o igual a 2.4 borralos ya que estarán mal, en los 2.4.x el char mayor pasa a ser 13 en vez de 15, haz lo siguiente:

```
cd /dev
rm js*
mkdir input
mknod input/js0 c 13 0
mknod input/js1 c 13 1
mknod input/js2 c 13 2
mknod input/js3 c 13 3
ln -s input/js0 js0
ln -s input/js1 js1
ln -s input/js2 js2
ln -s input/js3 js3
Para las utilidades de testeo (se explicará más adelante) poner lo siguiente:
mknod input/event0 c 13 64
mknod input/event1 c 13 65
mknod input/event2 c 13 66
mknod input/event3 c 13 67
```

### Cargar los módulos

Teóricamente basta con poner:  
modprobe joydev



`modprobe gamecon gc=0,7`

Cuando cargamos el **gamecon** se autocargarán automáticamente los módulos del puerto paralelo (`parport` y `parport_pc`), para asegurarte que se han cargado satisfactoriamente utiliza el comando **lsmod**. Para cargar el módulo **gamecon** no debe de estar cargado el módulo **lp** (el de la impresora) así que un **rmmod lp** antes de cargar **gamecon** estaría bien para asegurarse.

Los argumentos que le pasamos a **gamecon** tienen la siguiente sintaxis:

**gc=port,pad1,pad2,pad3,pad4,pad5**

Donde **port** es el número de la interface del puerto paralelo (si solo tienes uno es 0, ej. `parport0`) y **pad1,pad2...** son el tipo de pad que tienes conectado en cada pin de entrada del puerto paralelo (ver montaje). Si tienes más de un puerto paralelo puedes ampliar los argumentos con `gc_2,gc_3,etc.` Los diferentes tipos de pad son:

Tipo	Joystick/Pad
0	None
1	SNES pad
2	NES pad
4	Multisystem 1-button joystick
5	Multisystem 2-button joystick
6	N64 pad
7	Sony PSX controller

**gamecon** sirve para multitud de pads de videoconsolas, lo único que cambia respecto al pad de PSX es el [montaje](#)<sup>(1)</sup>. Los diferentes tipos de pad de PSX se autodetectan al cargar el módulo por lo que debe de estar conectado antes de cargarlo.

### Comprobar que todo funciona

Existen unas herramientas de testeo y calibración de joystick. Los debianeros hacen un **apt-get install joystick** y ya las tendréis listas, el resto de la gente se puede bajar el código de [aquí](#)<sup>(2)</sup>.

Con la herramienta **jstest** podremos saber si el joystick funciona correctamente, la sintaxis es:

**jstest /dev/jsX**

Si todo a ido bien te indicará el joystick conectado y te mostrará una serie de números los cuales se irán actualizando según los botones del pad que toques. Si obtienes un *no such device* como respuesta, es que probablemente no se hayan cargado los módulos correctamente, borra el módulo **gamecon** y vuelve a cargarlo, si sigue sin funcionar repasa todos los pasos.

La otra herramienta interesante es **jscal** la cual sirve para calibrar el joystick, el joystick ya se configura solo por defecto pero con esta herramienta puedes ajustar más los parámetros:

**jscal -c /dev/jsX**

Una vez calibrado puedes ver como va con **jstest** y si te gusta puedes guardar los parámetros en un archivo a parte con:

**jscal -p /dev/jsX > /etc/joystick.cal**

Así añadiendo la línea **source /etc/joystick.cal** a un script de inicio se cargará la configuración de calibrado automáticamente.

### Optimización del driver gamecon.c

Si habéis notado que aunque todo vaya bien, las aplicaciones pierden mucho rendimiento cuando el pad está conectado (jugar a un juego de cps-2 con el xmame es un buen ejemplo ;) ) se debe a que el driver está constantemente ocupando la CPU, para solucionar esto necesitamos tocar unas pocas líneas del código del driver **gamecon.c** el cual lo podéis encontrar en `/usr/src/linux/drivers/char/joystick`. Los cambios a realizar son muy simples:

- Cambiar en la constante `GC_PSX_DELAY` el valor de 60 por uno bastante menor. Ejemplo:  

```
#define GC_PSX_DELAY 15
```
- Cada vez que se use la variable observareis que se multiplica por 2, pues bien, quitar ese \* 2 y dejar la variable tal cual. Ejemplo:  

```
udelay(GC_PSX_DELAY);
```

Una vez realizados los cambios, recompila el módulo y compruebas que tal va. El valor de `GC_PSX_DELAY` es arbitrario, tienes que ir jugando con él hasta que encuentras uno que te vaya bien, yo primero probe 30 y me seguía



iendo demasiado lento, hasta que probé al 15 y me gusto como iba.

Pues bien con esto ya esta todo listo, la mejora no es perfecta, se sigue notando una bajada de frames pero no tan considerable como antes y hace que los juegos anteriormente injugables sean perfectamente jugables. Un ejemplo es que el juego **super street fighter 2** de la **super nintendo** con el emulador **zsnes** me daba 15/60 fps jugando con el pad, mientras que tras hacer la optimización me da 60/60, es decir, el mismo rendimiento que sin pad.

Casi toda la información aquí citada a sido extraida de la propia documentación del kernel, más concretamente de los archivos *joystick.txt* y *joystick-parport.txt*, así que si después de leer este artículo aun quieres saber más, no dejes de echar un vistazo a dichos documentos, los cuales están mucho más detallados técnicamente.

---

**Lista de enlaces de este artículo:**

1. <http://www.emulatronia.com/reportajes/directpad/index.htm>
2. <http://atrey.karlin.mff.cuni.cz/~vojtech/input/>

---

E-mail del autor: ElCoco8\_ARROBA\_teleline.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1268>