```
                         joystick.txt
                   Linux Joystick driver v2.0.0
             (c) 1996-2000 Vojtech Pavlik <vojtech@ucw.cz>
                        Sponsored by SuSE
```
----

0. Disclaimer
~~~~~~~~~~~~~~~

   This program is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by the Free
Software Foundation; either version 2 of the License, or (at your option)
any later version.

   This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for
more details.

   You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc., 59
Temple Place, Suite 330, Boston, MA 02111-1307 USA

   Should you need to contact me, the author, you can do so either by e-mail
- mail your message to <vojtech@ucw.cz>, or by paper mail: Vojtech Pavlik,
Simunkova 1594, Prague 8, 182 00 Czech Republic

   For your convenience, the GNU General Public License version 2 is included
in the package: See the file COPYING.

1. Intro
~~~~~~~~~~

   The joystick driver for Linux provides support for a variety of joysticks
and similar devices. It is based on a larger project aiming to support all
input devices in Linux.

   Should you encounter any problems while using the driver, or joysticks
this driver can't make complete use of, I'm very interested in hearing about
them. Bug reports and success stories are also welcome.

   The input project website is at:

        http://atrey.karlin.mff.cuni.cz/~vojtech/input/

   There is also a mailing list for the driver at:

        listproc@atrey.karlin.mff.cuni.cz

send "subscribe linux-joystick Your Name" to subscribe to it.

2. Usage
~~~~~~~~~~

   For basic usage you just choose the right options in kernel config and
you should be set.

2.1 inpututils
~~~~~~~~~~~~~~~~

For testing and other purposes (for example serial devices), a set of

utilities is available at the abovementioned website. I suggest you download
and install it before going on.

2.2 Device nodes
~~~~~~~~~~~~~~~~~

For applications to be able to use the joysticks,
you'll have to manually create these nodes in /dev:

```
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
```

For testing with inpututils it's also convenient to create these:

```
mknod input/event0 c 13 64
mknod input/event1 c 13 65
mknod input/event2 c 13 66
mknod input/event3 c 13 67
```

2.4 Modules needed
~~~~~~~~~~~~~~~~~~~

   For all joystick drivers to function, you'll need the userland interface
module in kernel, either loaded or compiled in:

```
        modprobe joydev
```

   For gameport joysticks, you'll have to load the gameport driver as well;

```
        modprobe ns558
```

   And for serial port joysticks, you'll need the serial input line
discipline module loaded and the inputattach utility started:

```
        modprobe serport
        inputattach -xxx /dev/tts/X &
```

   In addition to that, you'll need the joystick driver module itself, most
usually you'll have an analog joystick:

```
        modprobe analog
```

   For automatic module loading, something like this might work - tailor to
your needs:

```
        alias tty-ldisc-2 serport
        alias char-major-13 input
        above input joydev ns558 analog
        options analog map=gamepad,none,2btn
```

2.5 Verifying that it works

For testing the joystick driver functionality, there is the jstest program in the utilities package. You run it by typing:

        jstest /dev/js0

And it should show a line with the joystick values, which update as you move the stick, and press its buttons. The axes should all be zero when the joystick is in the center position. They should not jitter by themselves to other close values, and they also should be steady in any other position of the stick. They should have the full range from -32767 to 32767. If all this is met, then it's all fine, and you can play the games. :)

If it's not, then there might be a problem. Try to calibrate the joystick, and if it still doesn't work, read the drivers section of this file, the troubleshooting section, and the FAQ.

2.6. Calibration

For most joysticks you won't need any manual calibration, since the joystick should be autocalibrated by the driver automagically. However, with some analog joysticks, that either do not use linear resistors, or if you want better precision, you can use the jscal program

        jscal -c /dev/js0

 included in the joystick package to set better correction coefficients than what the driver would choose itself.

After calibrating the joystick you can verify if you like the new calibration using the jstest command, and if you do, you then can save the correction coefficients into a file

        jscal -p /dev/js0 > /etc/joystick.cal

And add a line to your rc script executing that file

        source /etc/joystick.cal

This way, after the next reboot your joystick will remain calibrated. You can also add the jscal -p line to your shutdown script.

3. HW specific driver information

In this section each of the separate hardware specific drivers is described.

3.1 Analog joysticks

The analog.c uses the standard analog inputs of the gameport, and thus supports all standard joysticks and gamepads. It uses a very advanced routine for this, allowing for data precision that can't be found on any other system.

It also supports extensions like additional hats and buttons compatible
with CH Flightstick Pro, ThrustMaster FCS or 6 and 8 button gamepads. Saitek
Cyborg 'digital' joysticks are also supported by this driver, because
they're basically souped up CHF sticks.

   However the only types that can be autodetected are:

* 2-axis, 4-button joystick
* 3-axis, 4-button joystick
* 4-axis, 4-button joystick
* Saitek Cyborg 'digital' joysticks

   For other joystick types (more/less axes, hats, and buttons) support
you'll need to specify the types either on the kernel command line or on the
module command line, when inserting analog into the kernel. The
parameters are:

        analog.map=<type1>,<type2>,<type3>,....

   'type' is type of the joystick from the table below, defining joysticks
present on gameports in the system, starting with gameport0, second 'type'
entry defining joystick on gameport1 and so on.

Type	Meaning
none	No analog joystick on that port
auto	Autodetect joystick
2btn	2-button n-axis joystick
y-joy	Two 2-button 2-axis joysticks on an Y-cable
y-pad	Two 2-button 2-axis gamepads on an Y-cable
fcs	Thrustmaster FCS compatible joystick
chf	Joystick with a CH Flightstick compatible hat
fullchf	CH Flightstick compatible with two hats and 6 buttons
gamepad	4/6-button n-axis gamepad
gamepad8	8-button 2-axis gamepad

   In case your joystick doesn't fit in any of the above categories, you can
specify the type as a number by combining the bits in the table below. This
is not recommended unless you really know what are you doing. It's not
dangerous, but not simple either.

Bit	Meaning
0	Axis X1
1	Axis Y1
2	Axis X2
3	Axis Y2
4	Button A
5	Button B
6	Button C
7	Button D
8	CHF Buttons X and Y
9	CHF Hat 1
10	CHF Hat 2
11	FCS Hat
12	Pad Button X

```
13 │ Pad Button Y
14 │ Pad Button U
15 │ Pad Button V
16 │ Saitek F1-F4 Buttons
17 │ Saitek Digital Mode
19 │ GamePad
20 │ Joy2 Axis X1
21 │ Joy2 Axis Y1
22 │ Joy2 Axis X2
23 │ Joy2 Axis Y2
24 │ Joy2 Button A
25 │ Joy2 Button B
26 │ Joy2 Button C
27 │ Joy2 Button D
31 │ Joy2 GamePad
```

3.2 Microsoft SideWinder joysticks

Microsoft 'Digital Overdrive' protocol is supported by the sidewinder.c
module. All currently supported joysticks:

* Microsoft SideWinder 3D Pro
* Microsoft SideWinder Force Feedback Pro
* Microsoft SideWinder Force Feedback Wheel
* Microsoft SideWinder FreeStyle Pro
* Microsoft SideWinder GamePad (up to four, chained)
* Microsoft SideWinder Precision Pro
* Microsoft SideWinder Precision Pro USB

are autodetected, and thus no module parameters are needed.

There is one caveat with the 3D Pro. There are 9 buttons reported,
although the joystick has only 8. The 9th button is the mode switch on the
rear side of the joystick. However, moving it, you'll reset the joystick,
and make it unresponsive for about a one third of a second. Furthermore, the
joystick will also re-center itself, taking the position it was in during
this time as a new center position. Use it if you want, but think first.

The SideWinder Standard is not a digital joystick, and thus is supported
by the analog driver described above.

3.3 Logitech ADI devices

Logitech ADI protocol is supported by the adi.c module. It should support
any Logitech device using this protocol. This includes, but is not limited
to:

* Logitech CyberMan 2
* Logitech ThunderPad Digital
* Logitech WingMan Extreme Digital
* Logitech WingMan Formula
* Logitech WingMan Interceptor
* Logitech WingMan GamePad
* Logitech WingMan GamePad USB
* Logitech WingMan GamePad Extreme
* Logitech WingMan Extreme Digital 3D

   ADI devices are autodetected, and the driver supports up to two (any
combination of) devices on a single gameport, using an Y-cable or chained
together.

   Logitech WingMan Joystick, Logitech WingMan Attack, Logitech WingMan
Extreme and Logitech WingMan ThunderPad are not digital joysticks and are
handled by the analog driver described above. Logitech WingMan Warrior and
Logitech Magellan are supported by serial drivers described below. Logitech
WingMan Force and Logitech WingMan Formula Force are supported by the
I-Force driver described below. Logitech CyberMan is not supported yet.

3.4 Gravis GrIP
~~~~~~~~~~~~~~~~~~

   Gravis GrIP protocol is supported by the grip.c module. It currently
supports:

* Gravis GamePad Pro
* Gravis BlackHawk Digital
* Gravis Xterminator
* Gravis Xterminator DualControl

   All these devices are autodetected, and you can even use any combination
of up to two of these pads either chained together or using an Y-cable on a
single gameport.

GrIP MultiPort isn't supported yet. Gravis Stinger is a serial device and is
supported by the stinger driver. Other Gravis joysticks are supported by the
analog driver.

3.5 FPGaming A3D and MadCatz A3D
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

   The Assassin 3D protocol created by FPGaming, is used both by FPGaming
themselves and is licensed to MadCatz. A3D devices are supported by the
a3d.c module. It currently supports:

* FPGaming Assassin 3D
* MadCatz Panther
* MadCatz Panther XL

   All these devices are autodetected. Because the Assassin 3D and the Panther
allow connecting analog joysticks to them, you'll need to load the analog
driver as well to handle the attached joysticks.

   The trackball should work with USB mousedev module as a normal mouse. See
the USB documentation for how to setup an USB mouse.

3.6 ThrustMaster DirectConnect (BSP)
~~~~~~

   The TM DirectConnect (BSP) protocol is supported by the tmdc.c
module. This includes, but is not limited to:

* ThrustMaster Millenium 3D Inceptor
* ThrustMaster 3D Rage Pad
* ThrustMaster Fusion Digital Game Pad

   Devices not directly supported, but hopefully working are:

* ThrustMaster FragMaster
* ThrustMaster Attack Throttle

   If you have one of these, contact me.

   TMDC devices are autodetected, and thus no parameters to the module
are needed. Up to two TMDC devices can be connected to one gameport, using
an Y-cable.

3.7 Creative Labs Blaster

   The Blaster protocol is supported by the cobra.c module. It supports only
the:

* Creative Blaster GamePad Cobra

   Up to two of these can be used on a single gameport, using an Y-cable.

3.8 Genius Digital joysticks

   The Genius digitally communicating joysticks are supported by the gf2k.c
module. This includes:

* Genius Flight2000 F-23 joystick
* Genius Flight2000 F-31 joystick
* Genius G-09D gamepad

   Other Genius digital joysticks are not supported yet, but support can be
added fairly easily.

3.9 InterAct Digital joysticks

   The InterAct digitally communicating joysticks are supported by the
interact.c module. This includes:

* InterAct HammerHead/FX gamepad
* InterAct ProPad8 gamepad

   Other InterAct digital joysticks are not supported yet, but support can be
added fairly easily.

3.10 PDPI Lightning 4 gamecards

   PDPI Lightning 4 gamecards are supported by the lightning.c module.
Once the module is loaded, the analog driver can be used to handle the
joysticks. Digitally communicating joystick will work only on port 0, while
using Y-cables, you can connect up to 8 analog joysticks to a single L4
card, 16 in case you have two in your system.

3.11 Trident 4DWave / Aureal Vortex

   Soundcards with a Trident 4DWave DX/NX or Aureal Vortex/Vortex2 chipsets
provide an "Enhanced Game Port" mode where the soundcard handles polling the
joystick. This mode is supported by the pcigame.c module. Once loaded the

analog driver can use the enhanced features of these gameports..

3.13 Crystal SoundFusion
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

   Soundcards with Crystal SoundFusion chipsets provide an "Enhanced Game
Port", much like the 4DWave or Vortex above. This, and also the normal mode
for the port of the SoundFusion is supported by the cs461x.c module.

3.14 SoundBlaster Live!
~~~~~~~~~~~~~~~~~~~~~~~~~

   The Live! has a special PCI gameport, which, although it doesn't provide
any "Enhanced" stuff like 4DWave and friends, is quite a bit faster than
its ISA counterparts. It also requires special support, hence the
emu10k1-gp.c module for it instead of the normal ns558.c one.

3.15 SoundBlaster 64 and 128 - ES1370 and ES1371, ESS Solo1 and S3 SonicVibes
~~~~~~

   These PCI soundcards have specific gameports. They are handled by the
sound drivers themselves. Make sure you select gameport support in the
joystick menu and sound card support in the sound menu for your appropriate
card.

3.16 Amiga
~~~~~~~~~~~

   Amiga joysticks, connected to an Amiga, are supported by the amijoy.c
driver. Since they can't be autodetected, the driver has a command line.

        amijoy.map=<a>,<b>

   a and b define the joysticks connected to the JOY0DAT and JOY1DAT ports of
the Amiga.

        Value | Joystick type
        ----
          0 | None
          1 | 1-button digital joystick

   No more joystick types are supported now, but that should change in the
future if I get an Amiga in the reach of my fingers.

3.17 Game console and 8-bit pads and joysticks
~~~~~~

See joystick-parport.txt for more info.

3.18 SpaceTec/LabTec devices
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

   SpaceTec serial devices communicate using the SpaceWare protocol. It is
supported by the spaceorb.c and spaceball.c drivers. The devices currently
supported by spaceorb.c are:

* SpaceTec SpaceBall Avenger
* SpaceTec SpaceOrb 360

Devices currently supported by spaceball.c are:

* SpaceTec SpaceBall 4000 FLX

In addition to having the spaceorb/spaceball and serport modules in the kernel, you also need to attach a serial port to it. to do that, run the inputattach program:

        inputattach --spaceorb /dev/tts/x &
or
        inputattach --spaceball /dev/tts/x &

where /dev/tts/x is the serial port which the device is connected to. After doing this, the device will be reported and will start working.

There is one caveat with the SpaceOrb. The button #6, the on the bottom side of the orb, although reported as an ordinary button, causes internal recentering of the spaceorb, moving the zero point to the position in which the ball is at the moment of pressing the button. So, think first before you bind it to some other function.

SpaceTec SpaceBall 2003 FLX and 3003 FLX are not supported yet.

3.19 Logitech SWIFT devices

The SWIFT serial protocol is supported by the warrior.c module. It currently supports only the:

* Logitech WingMan Warrior

but in the future, Logitech CyberMan (the original one, not CM2) could be supported as well. To use the module, you need to run inputattach after you insert/compile the module into your kernel:

        inputattach --warrior /dev/tts/x &

/dev/tts/x is the serial port your Warrior is attached to.

3.20 Magellan / Space Mouse

The Magellan (or Space Mouse), manufactured by LogiCad3d (formerly Space Systems), for many other companies (Logitech, HP, ...) is supported by the joy-magellan module. It currently supports only the:

* Magellan 3D
* Space Mouse

models, the additional buttons on the 'Plus' versions are not supported yet.

To use it, you need to attach the serial port to the driver using the

        inputattach --magellan /dev/tts/x &

command. After that the Magellan will be detected, initialized, will beep, and the /dev/input/jsX device should become usable.

3.21 I-Force devices

All I-Force devices are supported by the iforce module. This includes:

* AVB Mag Turbo Force
* AVB Top Shot Pegasus
* AVB Top Shot Force Feedback Racing Wheel
* Logitech WingMan Force
* Logitech WingMan Force Wheel
* Guillemot Race Leader Force Feedback
* Guillemot Force Feedback Racing Wheel
* Thrustmaster Motor Sport GT

   To use it, you need to attach the serial port to the driver using the

        inputattach --iforce /dev/tts/x &

command. After that the I-Force device will be detected, and the
/dev/input/jsX device should become usable.

   In case you're using the device via the USB port, the inputattach command
isn't needed.

   The I-Force driver now supports force feedback via the event interface.

   Please note that Logitech WingMan *3D devices are _not_ supported by this
module, rather by hid. Force feedback is not supported for those devices.
Logitech gamepads are also hid devices.

3.22 Gravis Stinger gamepad

   The Gravis Stinger serial port gamepad, designed for use with laptop
computers, is supported by the stinger.c module. To use it, attach the
serial port to the driver using:

        inputattach --stinger /dev/tty/x &

where x is the number of the serial port.

4. Troubleshooting

   There is quite a high probability that you run into some problems. For
testing whether the driver works, if in doubt, use the jstest utility in
some of its modes. The most useful modes are "normal" - for the 1.x
interface, and "old" for the "0.x" interface. You run it by typing:

        jstest --normal /dev/input/js0
        jstest --old   /dev/input/js0

   Additionally you can do a test with the evtest utility:

        evtest /dev/input/event0

   Oh, and read the FAQ! :)

5. FAQ

Q: Running 'jstest /dev/js0' results in "File not found" error. What's the
   cause?

A: The device files don't exist. Create them (see section 2.2).

Q: Is it possible to connect my old Atari/Commodore/Amiga/console joystick
   or pad that uses a 9-pin D-type cannon connector to the serial port of my
   PC?
A: Yes, it is possible, but it'll burn your serial port or the pad. It
   won't work, of course.

Q: My joystick doesn't work with Quake / Quake 2. What's the cause?
A: Quake / Quake 2 don't support joystick. Use joy2key to simulate keypresses
   for them.

6. Programming Interface
~~~~~~~~~~~~~~~~~~~~~~~~~

   The 1.0 driver uses a new, event based approach to the joystick driver.
Instead of the user program polling for the joystick values, the joystick
driver now reports only any changes of its state. See joystick-api.txt,
joystick.h and jstest.c included in the joystick package for more
information. The joystick device can be used in either blocking or
nonblocking mode and supports select() calls.

   For backward compatibility the old (v0.x) interface is still included.
Any call to the joystick driver using the old interface will return values
that are compatible to the old interface. This interface is still limited
to 2 axes, and applications using it usually decode only 2 buttons, although
the driver provides up to 32.