QtSixA/sixad Manual

Contents

Intro

QtSixA is an interface to 'sixad', an application that can connect PS3 hardware (Sixaxis/DualShock3, Keypads and Remotes) to a Linux-compatible machine.

"QtSixA" is the GUI (interface), while "sixad" is the backend C++ code.

Currently sixad supports:
- Sixaxis/DualShock3 (buttons, axis, accelerometers, leds and rumble)
- PS3 Keypads
- PS3 BD Remotes

There's also other small utilities that can be helpful for users:
- hidraw-dump
Dumps event information from hidraw devices, just like hcidump does for bluetooth
- sixpair/sixpair-kbd
Pairs the sixaxis (or *-kbd for keypads) to the current machine's bluetooth adapter, or the 1st command-line argument.
Note that BD Remotes don't need pairing with sixad.
- sixad-jack
Makes a sixaxis available as a MIDI keyboard.
- sixad-raw
Registers a new joystick using a sixaxis hidraw interface, useful to get accelerometers working on USB mode.

These tools are more deeply explained later in this file.

The Sixaxis in Linux

The sixaxis works out-of-the box in Linux; just get yourself an USB cable, connect the sixaxis to the computer, and press the 'PS' button on the sixaxis to activate it.
You won't get all sixaxis features this way though – leds will keep blinking, accelerometers and rumble won't work.
Note: when connected over USB the sixaxis leds will stop blinking when the internal battery is fully charged.

There is already work in progress to make the leds and rumble work by default in Linux, it's just a matter of time now.
Same thing applies to bluetooth too (through BlueZ).

QtSixA solves this by implementing a custom driver (based on linux uinput) that supports leds, accelerometers and rumble. The default way to handle sixaxis in QtSixA is bluetooth, but it can be used for USBas well (no leds or rumble support though).

The sixad driver

"sixad" is a small driver based on good old 'hidd' (back on the bluez 3.xx days...).
It currently supports sixaxis, keypads and remotes. The keypad support will actually load the system default driver, so wherever it will work or not depends on each system.
Please note that sixad is a bluetooth application, not meant for USB. For USB related tasks, check the 'sixad-raw' utility.

Another thing worth knowing is that sixad uses a different bluetooth method than the new BlueZ 4.xx stack (present on many recent distros), so they will probably conflict each other. sixad tries to work around this, but it's not always possible. Check the 'Known issues' section if you find any problems.

Usage

When ran with no command-line arguments, sixad will print this:

```
usage: sixad <command>

command can be:
    -h, --help          Show help (this message)
    -v, --version       Show sixad version

    -s, --start         Start sixad
        --stop          Stop sixad
        --remote        BD Remote mode

    -r, --restore       Restore regular bluetooth

        --boot-yes      Auto-starts sixad at boot time
        --boot-no       Does not auto-start sixad at boot time


You can also check: sixad-raw, sixad-notify
```

The 'help' command will simply display this message, while 'version' will display the current sixad version.
All other commands will require root access.

The 'start' command will start sixad (and stop regular bluetooth if needed). ~~'hcid' will be used here to keep the regular bluetooth from picking up bluetooth packets, which are needed for sixad.~~
Once you've connected all your sixaxis and keypads, you can use the 'restore' command to restore the regular bluetooth if needed.

The 'stop' command will stop sixad and all services (sixad-remote and sixad-sixaxis), disconnecting all sixad-related devices in the process. After stopping sixad, regular bluetooth is restored.

The 'remote' command is used to connect PS3 BD Remotes to the computer (note that pairing is not needed here).
After running this command, hold Enter+Start buttons on the remote to connect. Only 1 remote can be connected at a time.
Please note that, when searching for remotes, the bluetooth device will be very busy; if you have other bluetooth devices currently connected, they will seem slower during this time.
When sixad finds a remote, it will stop searching (remember – only 1 remote at a time), so you're good again after that.
But, if sixad finds a remote and it somehow disconnects, sixad will start the search again.
Note: You should probably connect the remotes first, then the sixaxis and keypads. Much easier this way...

The 'restore' command will restore the regular bluetooth. It's the same as in 'stop', but sixad and connected devices won't be affected.

The 'boot-yes' and 'boot-no' will enable sixad to start (or not) during boot. When this method is activated, pressing the PS button will connect any sixaxis or keypad right away, but regular bluetooth won't work.

Note: before you can connect over bluetooth, you need to pair your sixaxis or keypad to the current computer. Check the sixad utilities section, sixpair, for that.

Configure sixad

The sixad global configuration is stored in '/etc/default/sixad'. It currently has two options – DEBUG and LEGACY.
To change them, use something like 'DEBUG=1', where the value can be 0 or 1 (off or on, respectively).
When DEBUG is on, sixad will print additional messages when ran.
When LEGACY is on, sixad will on longer manage the sixaxis. Instead the default joystick driver will be used (ie, no leds, accelerometers or rumble).

System configuration is stored in '/var/lib/sixad/', as follows:
`/var/lib/sixad/` - Main sixad folder
`/var/lib/sixad/00:XX:00:XX:00:XX/` - Specific bluetooth adapter settings folder, ignore

`/var/lib/sixad/profiles/` - Device profiles folder
`/var/lib/sixad/profiles/default` - Default sixad configuration file
`/var/lib/sixad/profiles/hidraw` - Default sixad configuration file for hidraw devices
`/var/lib/sixad/profiles/00:XX:00:XX:00:XX` - Specific device configuration file

If some of these files don't exist yet, sixad will use internal defaults.
All the users in the group 'sixad' will have read+write permissions over those files.

sixad will try to load the specifc device configuration first. If not found it tries to load the 'default' one. If that is not found either, it will use the internal default configuration.
sixad-raw will first try to load the 'hidraw' file, then 'default'.

The sixad configuration file specification

The files inside '`/var/lib/sixad/`' folow these rules:
- Lines started with # are ignored
- Empty lines are ignored
- Lines with data must be "`string_of_config<space>value`", like this: "`enable_leds 1`"
- 'string_of_config' must only contain ASCII characters (ie, no 'ç' or 'é')
- 'value' must always be a number (0/1 is used for off/on)
- only 1 data per line

sixad will handle missing data, but resolve to internal defaults. If a specific device configuration doesn't have "`enable_leds`", sixad will load the internal default value for that, even if the 'default' file still has it.
Example:
File '`/var/lib/sixad/profiles/00:XX:00:XX:00:XX`' is missing some configuration, but the file '`/var/lib/sixad/profiles/default`' has it. Still, in this case, the missing configuration in the 1st file will not be loaded from the 2nd one, but from internal defaults.

The internal defaults are:
- Enable leds, auto-set by joystick number
- Enable joystick (buttons, axis, sensible buttons and accelerometers)
- Enable rumble, new mode
- Enable connection animation (led+rumble)
- Enable all remote features (numeric, dvd, direcctional and multimedia)
- Enable auto-disconnect if no action after 30 minutes
- Disable input (mouse and keyboard)

A complete file (with all currently available settings) is shown in the next page.

```
# ##########################
# sixad configuration file #
########################## #

# Features
enable_leds 1
enable_joystick 0
enable_input 1
enable_remote 0
enable_rumble 0
enable_timeout 0

# LED
led_n_auto 1
led_n_number 1
led_anim 0

# Joystick
enable_buttons 1
enable_sbuttons 1
enable_axis 1
enable_accel 1
enable_accon 0
enable_speed 0
enable_pos 0

# Input - None
key_select 0
key_l3 0
key_r3 0
key_start 0
key_up 0
key_right 0
key_down 0
key_left 0
key_l2 0
key_r2 0
key_l1 0
key_r1 0
key_tri 0
key_cir 0
key_squ 0
key_cro 0
key_ps 0
axis_left_type 0
axis_left_up 0
axis_left_right 0
axis_left_down 0
axis_left_left 0
axis_right_type 0
axis_right_up 0
axis_right_right 0
axis_right_down 0
axis_right_left 0
axis_speed 6
use_lr3 0

# Remote
remote_numberic 1
remote_dvd 1
remote_directional 1
remote_multimedia 1

# Rumble
old_rumble_mode 0

# Timeout
timeout_mins 30

# Misc
out_of_reach_disconnects 0
```

Here's a description of the values:

String	Possible values	Information
Features		
enable_leds	0 or 1	Wherever to use leds. If not, all leds will be off
enable_joystick	0 or 1	Wherever to register a joystick for this device
enable_input	0 or 1	Wherever to register a mouse+keyboard for this device
enable_remote	0 or 1	Wherever to use specific remote features (remote only)
enable_rumble	0 or 1	Wherever to use rumble
enable_timeout	0 or 1	Wherever to auto-disconnect after unused for some time
LED		
led_n_auto	0 or 1	Wherever to use led according to joystick number
led_n_number	1<->10	Specific led number (if led_n_auto is off)
led_anim	0 or 1	Wherever to animate new connections (will also rumble, if active)
Joystick		
enable_buttons	0 or 1	Wherever to handle button events
enable_sbuttons	0 or 1	Wherever to handle sensible button events
enable_axis	0 or 1	Wherever to handle axis events
enable_accel	0 or 1	Wherever to handle accelerometer events
enable_accon	0 or 1	Wherever to handle "acceleration" events (calculated with accelerometers)
enable_speed	0 or 1	Wherever to handle "speed" events (calculated with accelerometers)
enable_pos	0 or 1	Wherever to handle "position" events (calculated with accelerometers)
Input		
key_*	0<->MAX	What key this button will provide (see your kernel's input.h for a list)
axis_left_type	0, 2 or 3	0 - don't use axis; 2 - use for keys; 3 - use for mouse
axis_left_up	0<->MAX	In keys mode, used as key_*: in mouse mode, see below
axis_left_right	0<->MAX	In keys mode, used as key_*: in mouse mode, see below
axis_left_down	0<->MAX	In keys mode, used as key_*: in mouse mode does nothing
axis_left_left	0<->MAX	In keys mode, used as key_*: in mouse mode does nothing
axis_right_*	(same)	(same thing as with axis_left_*)
axis_speed	1<->9	Axis speed multiplicator; default value is 6
use_lr3	0 or 1	When on, L3 will enable/disable keys while R3 does it for mouse
Remote		
remote_numeric	0 or 1	Wherever to use the numberic keys on the remote
remote_dvd	0 or 1	Wherever to use the DVD keys on the remote
remote_directional	0 or 1	Wherever to use the directional keys on the remote
remote_multimedia	0 or 1	Wherever to use the multimedia keys on the remote, see below
Rumble		
old_rumble_mode	0 or 1	Wherever to use the old linux rumble mode or not; usually off
Timeout		
timeout_mins	1<->+inf	Timeout in minutes used to disconnect when no unused
Misc		
out_of_reach_disconnects	0 or 1	Wherever to auto-disconnect when device goes too far (sixaxis only, experimental)

QtSixA – The interface

Important note: Although the interface is very useful for setting up things like profiles, it's being targetted for complete rewrite soon. For this reason it will not be explained in this manual.
Once ready, the manual will be updated, and a proper QtSixA section will be written.

Please continue to the next page.

The sixad utilities

Here is a detailed explanation of each utility that comes with sixad.

hidraw-dump

This utility will dump event information from hidraw devices, just like hcidump does for bluetooth.
Simply run:

```
$ [sudo] hidraw-dump /dev/hidrawX
```

Where 'X' is the hidraw interface number.

sixpair[-kbd]

This utility pair you sixaxis to the computer (*-kbd for keypad), required to connect over bluetooth.
It accepts one command-line argument, which is the MAC to pair with (if no arguments, it will try to get the default MAC from your default bluetooth adapter).
You need to connect your bluetooth adapter (and turn it on, if such switch exists), and your sixaxis or keypad via USB.
Once ready, run:

```
$ [sudo] sixpair[-kbd] [00:XX:00:XX:00:XX]
```

Use *-kbd for keypads. The 1st argument (00:XX...) is optional.
PS3 BD Remote don't need this.

sixad-jack

This utility will turn a sixaxis into a MIDI keyboard. This is made through JACK, so make sure JACK is running first.
It has one command-line option '-b'. When used, black keys will be handled (default is to use white keys only).

The key mapping is as follows:

Select	- Change current octave (from 3 to 6)
Start	- Start/Stop JACK transport
L3	- Transport backwards
R3	- Transport forwards
PS	- Panic button

(other keys will play notes, using sensible buttons)

Left Axis vertical	- CC Modulation
Right Axis horizontal	- CC Pan
Accelerometer X	- CC Misc 1 (0x10)
Accelerometer Y	- CC Misc 2 (0x11)
Accelerometer Z	- CC Misc 3 (0x12)

Make sure you enabled sensible buttons for this sixaxis, since that is used for the key presses (sensitive!)
You can enable/disable features in sixad (ie, disable accelerometers to disable CC Misc events)

Run it like this:

```
$ sixad-jack [-b] /dev/input/jsX
```

Where 'X' is the sixaxis joystick number. The '-b' argument is optional.

Note: You should use this utility together with sixad. Using this on regular sixaxis (USB) will not work properly.

sixad-raw

This utility will make a hidraw device of a sixaxis work with the sixad driver. It's useful for accelerometers and input (ie, mouse+keyboard). It will not handle leds and rumble though.
Simply run:

```
$ [sudo] sixad-raw /dev/hidrawX
```

Where 'X' is the hidraw interface number.

Connecting and disconnecting devices

Sixaxis/DualShock3

If you want to connect and work via USB, you're in the wrong place. Sixaxis work in Linux out-of-the-box, please read the intro.
If you want to access the accelerometers via USB, then it's a different story. In this case, configure sixad for 'hidraw' and use the sixad-raw utility (this information was explained before in this document).

To connect over bluetooth, if you first need to pair your sixaxis. For that, see the sixad utilities, sixpair section.
Once done, start sixad with 'sixad -s' and press the PS button.

To disconnect, press and hold the PS button for 12 seconds.

Keypad

Keypads are not currently working via USB on Linux. If you got it working via USB, let me know!

To connect over bluetooth, do the same thing as for the sixaxis, as explained above.
Note that sixad will use the system driver for the keypad, so wherever it will work fine or not depends on your current distribution.

To disconnect, just push the switch for it.
The keypad will auto-disconnect itself after being unused for 5 minutes though.

Remote

Remotes cannot be used on USB.
For bluetooth, follow the explanation in the main sixad section, 'remote' argument option.
There's no way to disconnect a remote via hardware. You'll have to use QtSixA or kill sixad to force it to disconnect.
Still, note that sixad will auto-disconnect any device after being unused for 30 minutes.