
linmctool - Linux motion-sensing controller tool

Copyright © 2010, 2011 pabr@pabr.org
All rights reserved.

linmctool provides minimal support for popular Bluetooth motion-sensing game controllers on Linux: PlayStation Motion Controller (PS Move), Sixaxis, DualShock3, Wii Remote (Wiimote).

READ THE HYPERTEXT VERSION HERE:
<http://www.pabr.org/linmctool/linmctool.en.html>

Revision History		
1.1	2011-03-04	Timestamps. PS Move LEDs. Wii MotionPlus. Added mcalibrate.
1.0	2010-11-17	Initial release.

Table of Contents

1. Introduction	3
2. Disclaimer	3
3. Limitations	3
4. Build	4
5. Operation	4
5.1. Wiimote	4
5.2. SIXAXIS, DS3, PS Move	4
6. Troubleshooting	5
7. mccalibrate - Calibration tool	6
7.1. Interactive calibration	6
7.2. Applying calibration values to a data stream	7
8. mctrack - Motion tracking demo	7
9. Acknowledgements	7
Bibliography	8

1. Introduction

linmctool is a command-line utility which handles Bluetooth communications with several models of motion-sensing game controllers, and outputs sensor data in plain text. It is intended as a simple and portable tool for educational and research purposes in relation to such topics as 3D coordinate systems, Newtonian physics, robotics, sensor fusion, motion tracking and new user interfaces. Motion-sensing game controllers are a great opportunity to introduce children and students to a wide range of scientific topics, from the basic concept of inertial forces all the way to Kalman filtering.

The following devices are supported:

- Wiimote (including infrared sensor and gyros)
- SIXAXIS
- DS3
- PS Move (excluding PS Eye camera)

Sample output follows:

```
0 00:25:xx:xx:xx:xx WIIMOTE aX=-11 aY=-10 aZ=14 ir0x=966 ir0y=406 irlx=747 irly=419 \
  ir2x=554 ir2y=430 ir3x=467 ir3y=389 gX=209 gY=-298 gZ=232
1 00:19:xx:xx:xx:xx SIXAXIS aX=-3 aY=-21 aZ=-111 gZ=-21
2 00:06:xx:xx:xx:xx PSMOVE seq=0 aX=-2613 aY=-329 aZ=3271 gX=5 gY=8 gZ=8 mX=-197 mY=
```

2. Disclaimer

These game controllers are marketed as peripherals with Bluetooth connectivity (Bluetooth is an open standard which promotes interoperability between computers and wireless devices) and **linmctool** communicates with their open interfaces via the standard Bluetooth HID protocol. There is no need to modify the devices or circumvent any protection. Still, since this software is not endorsed by manufacturers, we cannot guarantee that it configures the hardware exactly as intended. There is always a risk of voiding the warranty, damaging the hardware by operating it beyond safe ratings, or worse. Use at your own risk!

For example, it is unknown whether the PS Move autonomously regulates the power of its RGB LEDs or whether the Bluetooth host is supposed to do it based on temperature readings. As a precaution, **linmctool** will limit brightness to 25 %.

3. Limitations

- **linmctool** is not a full-featured driver for HID peripherals. It does not support buttons. It does not provide an efficient binary API for applications.
- Since there is no documentation for these devices, it is likely that **linmctool** does not take advantage of their full functionality or accuracy.
- In order to avoid the limitations of the **hidraw** interface on some kernels, **linmctool** bypasses the Linux Bluetooth HID layer entirely. It must run with root privileges to take control of HID L2CAP PSMs 17 and 19. Other Bluetooth HID devices such as keyboards and mice cannot be used simultaneously. When used without root privileges, only the Wiimote is supported.
- **linmctool** does not use Bluetooth authentication nor encryption.
- **linmctool** supports any number of simultaneous devices, but system and hardware resource limitations apply. With some configurations, it may be difficult to reach the Bluetooth limit of 7 devices per adapter. Besides, with multiple devices transmitting simultaneously, packet loss may occur.

- **linmctool** may fail to reliably distinguish a SIXAXIS from a DS3. If your DS3 rumbles continuously, use **linmctool --force-ds3**; if your SIXAXIS gyro does not work, use **linmctool --force-sixaxis**; and please send the output of **linmctool --dump-readable** to the author.

4. Build

Unlike other approaches which tend to face compatibility and integration issues, **linmctool** is designed to compile and run easily on any Linux distribution, with barely any dependency.

```
# wget http://www.pabr.org/linmctool/linmctool-20110304.c [linmctool-20110304.c]
# gcc --std=gnu99 -Wall linmctool-20110304.c -lusb -o linmctool
```

For systems without **libusb**, compile with **-DWITHOUT_USB**. USB pairing, if needed, will have to be performed on another host (see [Pairing](#)).

```
# gcc --std=gnu99 -Wall -DWITHOUT_USB linmctool-20110304.c -o linmctool
```

5. Operation

5.1. Wiimote

The Wiimote is capable of powering up and initiating a Bluetooth connection when any button is pressed, but this feature cannot be supported with all models of Bluetooth adapters. Therefore **linmctool** requires a more complex procedure.

- **Obtain the Bluetooth address of the Wiimote** . Put it in discoverable mode by simultaneously pressing buttons 1 and 2, then run **hcitool scan** while the blue LEDs are flashing. Write down the Bluetooth address formatted as **xx:xx:xx:xx:xx:xx**.
- **Establish the Bluetooth connection** . Pass the Bluetooth address(es) to **linmctool** as command-line arguments. Within seconds before starting **linmctool**, put the Wiimote(s) in discoverable mode again by simultaneously pressing buttons 1 and 2.

```
# ./linmctool --ir --wmp 00:25:xx:xx:xx:xx
Waiting for Bluetooth connections.
Connecting to 00:25:xx:xx:xx:xx
New device 0 00:25:xx:xx:xx:xx is a Wiimote
0 00:25:xx:xx:xx:xx WIIMOTE aX=-11 aY=-10 aZ=14 ir0x=966 ir0y=406 ir1x=747 ir1y=419
ir2x=554 ir2y=430 ir3x=467 ir3y=389 gX=209 gY=-298 gZ=232
```

5.2. SIXAXIS, DS3, PS Move

- **Disabling the Bluetooth HID layer on old Linux distributions with hidd** . Simply terminate **hidd**:

```
# killall hidd
```

- **Disabling the Bluetooth HID layer on recent Linux distributions with bluetoothd** . Add **input** to the directive **DisablePlugins** in `/etc/bluetooth/main.conf`:

```
[General]
# List of plugins that should not be loaded on bluetoothd startup
```

```
DisablePlugins = network,input
```

Then restart **bluetoothd**:

```
# service bluetooth restart
```

- **Pairing** . These devices must be paired with a Bluetooth adapter of the Linux host. They apparently do not support the standard Bluetooth pairing procedure; instead, pairing is performed over USB, which is arguably simpler and more secure. Pairing is persistent: there is no need to repeat this procedure, unless the device has been paired with another host.

- Connect the device with a USB cable.

- Run **linmctool**:

```
# ./linmctool
Waiting for Bluetooth connections.
USB: PS MOVE
    Changing master from xx:xx:xx:xx:xx to yy:yy:yy:yy:yy
    Now press the PS button.
```

- Disconnect the USB cable. (The PS Move can be used in Bluetooth mode while plugged in, but this is not recommended, for obvious reasons.)

Devices will be paired with the first active Bluetooth adapter, as reported by `hcitool dev`. In order to pair with another adapter, or with an adapter on another machine, obtain its Bluetooth address by running `hcitool dev`, and pass it as follows:

```
# ./linmctool --master xx:xx:xx:xx:xx:xx
```

- **Establishing the Bluetooth connection** . Press the PS button at any time while **linmctool** is running and the USB cable is unplugged.

```
# ./linmctool
Waiting for Bluetooth connections.
Incoming connection...
New device 0 00:06:xx:xx:xx:xx is a PS Move
0 00:06:xx:xx:xx:xx PSMOVE seq=0 aX=4232 aY=-135 aZ=652 gX=4 gY=-9 gZ=-7 \
    mX=-94 mY=50 mZ=-91
```

- **PS Move bulb control** . R,G,B components can be specified on the command-line and updated per-device by typing commands on standard input. Changes are sent to the device within 3 seconds.

```
# ./linmctool --rgb 0,10,0
Waiting for Bluetooth connections.
Incoming connection...
New device 0 00:06:xx:xx:xx:xx is a PS Move
0 rgb 10,0,0
...
```

6. Troubleshooting

Here are debugging techniques which can help improve support for these devices.

- To see the HID descriptors of a USB device in plain text, simply use **lsusb -v**. Do this after running **linmctool**; otherwise **lsusb** will print "Report Descriptors: ** UNAVAILABLE **".

- **binhistogram** displays bit-wise and byte-wise histograms of input reports.

```
# wget http://www.pabr.org/linmctool/binhistogram.c [binhistogram.c]
# gcc --std=gnu99 -Wall binhistogram.c -o binhistogram
# ./linmctool --binary | ./binhistogram
```

For example, with the PS Move, this tool reveals that slowly depressing the trigger produces evenly spaced analog values, but in alternating fields:

```
6 41 29 00##### #_____#_____#____#___#___#___#_____
7 43 2b 00##### #_#_____#_____#____#___#___#___#_____
```

This suggests that two samples are sent per input report. The same effect can be observed with the accelerometers and gyros (but not with the magnetometers). The sampling rate is about 175 Hz.

- **linmctool --dump-readable** dumps all 1024 HID reports of the first detected device (either USB or Bluetooth).

linmctool --dump-readable --repeat-dump 5 detects readable reports with a dynamic content.

linmctool --dump-writable detects writable reports (by trying to write 16 null bytes, which could be dangerous).

With the PS Move connected via USB, these tools reveal that report number 0x04 contains Bluetooth addresses. Report 0x04 is not writable, but its neighbours 0x03 and 0x05 are. Also, the first byte of most (but not all) reports is the report ID.

- Use **hcidump -t -V -x** to diagnose Bluetooth connection failures.
- The HID standard is specified in [USBHID] and [HIDP].

7. mccalibrate - Calibration tool

mccalibrate turns raw data from **linmctool** into physically-relevant metric values.

The current implementation simply performs a linear mapping. Two reference values (± 1 g, ± 1 rad/s) for each sensor axis are stored in the working directory.

```
# wget http://www.pabr.org/linmctool/mccalibrate-20110304.c [mccalibrate-20110304.c]
# gcc --std=gnu99 -Wall mccalibrate-20110304.c -lm -o mccalibrate
```

7.1. Interactive calibration

mccalibrate --static calibrates the static sensors of the PS Move (accelerometers and magnetometers). The device must be manually put in six distinct orientations (for example: normal, upside-down, roll left, roll right, pitch up, pitch down). **mccalibrate** determines the reference values by fitting an ellipsoid to the data.

```
# ./linmctool --nostdin | ./mccalibrate --static
Incoming connection...
New device 0 00:06:xx:xx:xx:xx is a PS Move
Read calibration from 00:06:xx:xx:xx:xx.cal
Put PS MOVE in at least 6 orientations.
Press RETURN to sample when it is stationary.
Please wait 1s... Got 1 samples, please continue.
Please wait 1s... Got 2 samples, please continue.
```

```
Please wait 1s... Got 3 samples, please continue.
Please wait 1s... Got 4 samples, please continue.
Please wait 1s... Got 5 samples, please continue.
Please wait 1s... Computing...
ACC: 6 samples, coverage 99% 99% 99%, err 19.792511
MAG: 6 samples, coverage 98% 95% 99%, err 0.741071
A0  min   +656 ppm   max  +1339 ppm
A1  min  -2325 ppm   max   +204 ppm
A2  min   -496 ppm   max  -2189 ppm
G0  min     +0 ppm   max     +0 ppm
G1  min     +0 ppm   max     +0 ppm
G2  min     +0 ppm   max     +0 ppm
M0  min -101515 ppm   max  +78074 ppm
M1  min -72260 ppm   max +102799 ppm
M2  min -98625 ppm   max  +91781 ppm
Wrote 00:06:xx:xx:xx:xx.cal
```

Known limitations:

- The cost function is algebraic (Euclidian distance might be better)
- Minimization is performed by a random walk (not guaranteed to converge)
- Axes are assumed perfectly orthogonal (no cross-axis sensitivity)
- Local gravity is assumed to be 9.81 m/s² (should be adjusted for altitude; will not work on the ISS).

7.2. Applying calibration values to a data stream

```
# ./linmctool | ./mccalibrate
Waiting for Bluetooth connections.
Incoming connection...
New device 0 00:06:xx:xx:xx:xx is a PS Move
Read calibration from 00:06:xx:xx:xx:xx.cal
0 00:06:xx:xx:xx:xx PSMOVE seq=12  aX=-0.10400 aY=9.82060 aZ=-0.00367  \
  gX=0.00000 gY=0.00500 gZ=0.03833  mX=0.1890  mY=0.9726  mZ=-0.3458
```

8. mctrack - Motion tracking demo

mctrack estimates and graphically displays the position and orientation of up to four devices. The current version works only with the PS Move.

```
# wget http://www.pabr.org/linmctool/mctrack-20110304.c [mctrack-20110304.c]
# gcc --std=gnu99 -Wall mctrack-20110304.c -lX11 -lm -o mctrack
# ./linmctool | ./mccalibrate | ./mctrack
```

There is plenty of published research on motion estimation with accelerometers, gyros and magnetometers. **mctrack** is intended as a basic example; it implements naive Euler integration of instantaneous rotation and acceleration, with long-term registration to a reference position and an absolute orientation defined by gravity and magnetic field. Damping factors are hard-coded.

9. Acknowledgements

The HID-over-L2CAP-socket code is inspired from the **hidd** daemon in Bluez [<http://www.bluez.org/>]. This approach, which achieves maximum portability, was explored earlier by Jim Paris [<http://ps3.jim.sh/sixaxis/bt/>].

WiiMote support is based on technical information published by the Wiibrew [<http://wiibrew.org/wiki/Wiimote>] project.

SIXAXIS/DS3 support is from [SIXLINUX], with additional support for LEDs, gyro and rumble.

The MoveOnPC [<http://code.google.com/p/moveonpc/>] project pioneered work toward open-source support for the PS Move, primarily on Windows, and with a wider scope including optical tracking with the PS Eye camera.

Thanks to Sony and Nintendo for adopting Bluetooth, an open standard, in their gaming systems. Although they are sometimes criticized for not fully complying with the HID specifications, their protocols remain reasonably open and intelligible. Things would have been much worse if the product designers had opted to actively impede third-party projects such as this one.

Bibliography

[SIXLINUX] *Using the PlayStation 3 controller in Bluetooth mode with Linux*. <http://www.pabr.org/sixlinux/sixlinux.en.html>.

[USBHID] *Universal Serial Bus. Device Class Definition for Human Interface Devices (HID)*. http://www.usb.org/developers/devclass_docs/HID1_11.pdf.

[HIDP] *Bluetooth Specification. Human Interface Device (HID) Profile*. http://www.bluetooth.com/Specification%20Documents/HID_SPEC_V10.pdf.