
brickhid - Direct connection between Bluetooth gamepads and LEGO Mindstorms EV3

Copyright © 2014 pabr@pabr.org
All rights reserved.

We show how to interface LEGO Mindstorms EV3 directly with off-the-shelf Bluetooth gamepads and joysticks such as the PS3 SIXAXIS and DualShock 3 controllers.

READ THE HYPERTEXT VERSION HERE:

<http://www.pabr.org/bricks/brickhid/brickhid.en.html>

Revision History		
1.1	2015-10-21	Added recommendation about alternative operating systems.
1.0	2014-10-24	Initial release.

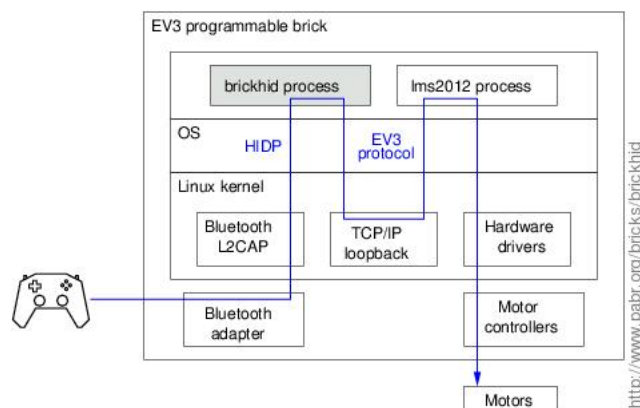
Table of Contents

1. Introduction	3
2. Requirements	3
2.1. LEGO parts	3
2.2. Bluetooth gamepad	4
2.3. WiFi	5
2.4. Computer and software tools	5
3. Software: brickhid	5
3.1. Environment	5
3.2. Bluetooth interface	5
3.3. EV3 interface	6
3.4. Configuration file	6
4. Operation	6
4.1. WiFi setup	6
4.2. Installing and running brickhid	7
4.3. Pairing the Bluetooth gamepad with the EV3 brick	8
4.4. Using brickhid	8
5. Known problems	8
6. Perspectives and future work	8
Bibliography	8

1. Introduction

The LEGO Mindstorms EV3 platform brings significant enhancements in CPU power, programmability and connectivity over the previous generation, Mindstorms NXT. Actually, there are so many new features that some of them are not fully exploited out-of-the-box. In particular, the EV3 brick contains a generic Bluetooth protocol stack, but it only uses it to communicate with Mindstorms software over a dedicated protocol.

In this project we show that a small C program running on the EV3, **brickhid**, can handle Bluetooth communications with a PlayStation 3 controller and translate stick positions and button presses into motor commands.



Others have already demonstrated the use of Bluetooth and USB gamepads with Mindstorms EV3 and NXT, but usually with proxy software running on a third device (PC, laptop, tablet or single-board computer). Here we establish a direct Bluetooth connection between the gamepad and the EV3 brick.

More generally, this project suggests that the EV3 brick can interact with a variety of Bluetooth devices, even without replacing the stock firmware:

- other gamepads, keyboards, mice, fitness trackers, health sensors, smart watches, home automation devices, GPS receivers
- the standard Bluetooth functions of smartphones: file transfer, PIM data exchange, call control, mobile networking
- possibly also audio headsets and other multimedia streaming devices.

Note . A better way to take advantage of all the features of the EV3 platform is to install an alternative operating system such as EV3DEV. See for example [BRICKFPV].

2. Requirements

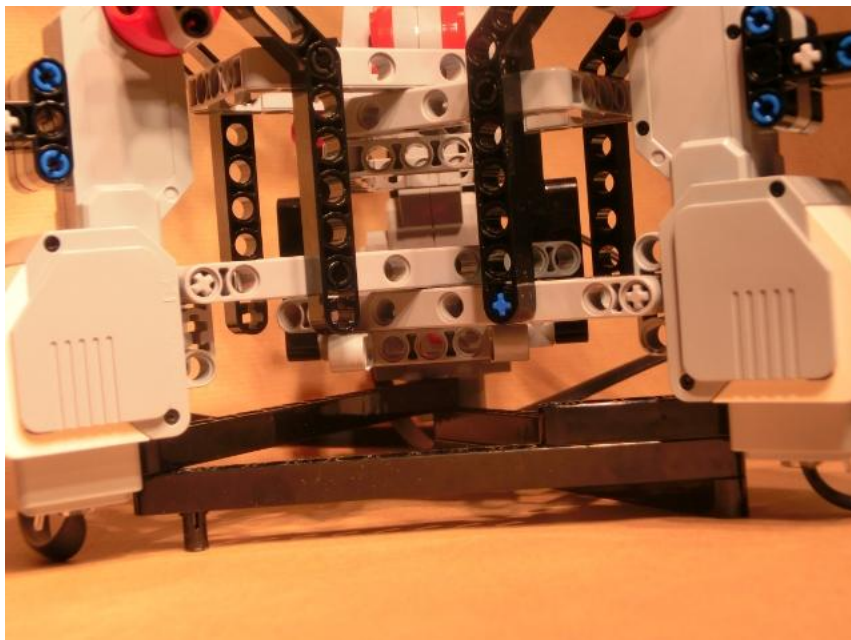
2.1. LEGO parts

The tracked vehicle in the demo video is the standard TRACK3R model from the Mindstorms EV3 base set (31313), except that the programmable brick has been shifted to expose the USB Host port.

Our delta robot uses additional parts from the EV3 Education Expansion set (45560), as well as an extra EV3 large motor (45502).



Positioning accuracy is disappointing (at least by delta robot standards), even though care has been taken to make the chassis very rigid with three layers of beams and rectangular frames interlocking at 120° . Still, this model is a good introduction to parallel robot kinematics. For better accuracy, gear reduction and pre-tensioning should be considered.



2.2. Bluetooth gamepad

This project has been tested with an old SIXAXIS controller from 2007. The DualShock 3 should work as well.

Second-hand units should be massively available and inexpensive in the next few years as newer gaming consoles reach the market.

2.3. WiFi

We use WiFi to upload files to the EV3 brick and access its command-line. This is the simplest way to test **brickhid**. Note however that the stock EV3 firmware supports only one specific WiFi dongle (WNA1100 0846:9030).

It might be possible to run programs from a microSD card containing a customized firmware, but we have not investigated this approach.

2.4. Computer and software tools

A generic Linux PC with an ARM cross-compilation toolchain is recommended to compile **brickhid**, access the EV3 command-line, transfer files, pair the Bluetooth gamepad and start the program.

Users without a cross-compilation toolchain can either use our pre-compiled **brickhid** binary, or compile and run **brickhid** on a PC rather than on the EV3 brick. Communications with the Mindstorms runtime will take place over WiFi rather than over the loopback interface.

This project does not use the Mindstorms graphical programming environment for Windows/OSX.

3. Software: brickhid

Download source code: [brickhid-1.0.tar.gz](#)

Browse source code: [brickhid-1.0/](#)

Compile from source:

```
user@PC $ wget http://www.pabr.org/bricks/brickhid/brickhid-1.0.tar.gz
user@PC $ tar zxf brickhid-1.0.tar.gz
user@PC $ cd brickhid-1.0
user@PC $ make
```

3.1. Environment

The EV3 brick runs a generic embedded Linux OS on an ARM SoC.

The platform has only 64 MB of RAM, most of which is already used by the filesystem image (cramfs), the operating system, the Mindstorms runtime, and EV3 user programs (loaded in a ramdisk). The Mindstorms runtime process consumes about 50% of CPU power when idle. So **brickhid** must be kept as lean and efficient as possible. Hence the decision to use the C programming language.

3.2. Bluetooth interface

The EV3 brick contains a generic Bluetooth transceiver and the firmware includes Bluez, the standard Linux Bluetooth protocol stack.

The Bluetooth standard specifies "profiles", which are essentially dedicated protocols for various types of devices. For keyboards, mice and gamepads, the relevant profile is called HIDP (Human Interface Device Profile). HIDP is supported on most Linux distributions, which is why Bluetooth keyboards and mice usually work out-of-the-box. The Linux kernel also has an interface called HIDRAW which allows applications to access more exotic HID devices such as game controllers.

Unfortunately the Linux kernel in the stock EV3 firmware has been compiled without support for HIDP and HIDRAW, presumably to save memory. The following procedure can be used to check whether this situation changes in future releases:

```
root@EV3:~# zcat /proc/config.gz | grep 'HIDP\|HIDRAW'
# CONFIG_BT_HIDP is not set
# CONFIG_HIDRAW is not set
```

As a consequence, **brickhid** must interface with the Bluetooth stack at a lower layer, namely L2CAP. **brickhid** creates Bluetooth L2CAP sockets and accepts connections on PSMs 17 and 19 which are allocated to HIDP.

brickhid does not implement the full HID specification. The current version has hard-coded support for the message format of the PS3 gamepad. But it is very easy to add decoders for other HID devices.

3.3. EV3 interface

brickhid does not control the motor drivers directly. Rather, it sends EV3 bytecodes (such as motor commands) to the Mindstorms runtime, **lms2012**. This is done over a loopback TCP connection, with the same EV3 protocol that is used over WiFi.

The EV3 protocol specifies a command for setting the angular position of a motor [https://github.com/mindboards/ev3sources/blob/master/lms2012/c_output/source/c_output.c#L38], but it is apparently not implemented [https://github.com/mindboards/ev3sources/blob/master/lms2012/d_pwm/Linuxmod_AM1808/d_pwm.c#L2492]. So **brickhid** implements a simple servo loop using the commands **OUTPUT_GET_COUNT** [https://github.com/mindboards/ev3sources/blob/master/lms2012/c_output/source/c_output.c#L47] and **OUTPUT_SPEED** [https://github.com/mindboards/ev3sources/blob/master/lms2012/c_output/source/c_output.c#L32]. This causes noticeable latency and overshoot in the delta robot demo.

3.4. Configuration file

On start-up **brickhid** reads a configuration file which specifies how buttons presses and stick positions should be converted into motor commands.

brickhid-1.0/ps3_tank.config

brickhid-1.0/ps3_delta.config

4. Operation

4.1. WiFi setup

- Plug the WiFi adapter into the USB Host port of the EV3 brick.
- Turn the EV3 brick on.
- In the EV3 menus:
 - Navigate to the "Tools" menu
 - Select "WiFi"
 - Select the "WiFi" checkbox and wait
 - Select Connections

- Select your WiFi access-point from the list, then wait
- Enter the WEP/WPA passphrase if needed
- Select "Connect"
- In the EV3 menus:
 - Navigate to the "Tools" menu
 - Select "Brick info"
 - Scroll to the bottom and note the IP address of the EV3 brick (e.g. 192.168.1.2) for future use.

4.2. Installing and running brickhid

- In a shell on a PC connected to the same WiFi network:

```
user@PC $ telnet 192.168.1.2
login: root
root@EV3:~#
```

(Replace 192.168.1.2 with the IP address of the EV3 brick from a previous step.)

```
root@EV3:~# dropbear
root@EV3:~# hcitool dev
Devices:
    hci0    11:22:33:44:55:66
```

Note the Bluetooth address of the EV3 brick (11:22:33:44:55:66 in this example) for future use.

- In another shell, if you have not compiled from source, download the ARM executable and a configuration file example:

```
user@PC $ wget http://www.pabr.org/bricks/brickhid/brickhid-1.0/brickhid.arm
user@PC $ wget http://www.pabr.org/bricks/brickhid/brickhid-1.0/ps3_tank.config
```

Then upload them to the EV3 brick:

```
user@PC $ scp brickhid.arm ps3_tank.config root@192.168.1.2:/var/volatile/
password: (press return)
```

(Replace 192.168.1.2 with the IP address of the EV3 brick from a previous step.)

- In the first shell:

```
root@EV3:~# killall -STOP udevd
root@EV3:~# killall bluetoothd
root@EV3:~# hciconfig hci0 up pscan
root@EV3:~# cd /var/volatile
root@EV3:~# chmod a+x brickhid.arm
root@EV3:~# ./brickhid.arm ps3_tank.config
```

brickhid immediately interfaces with the Mindstorms runtime and plays a low-pitched tone. Then it waits for an incoming Bluetooth connection.

4.3. Pairing the Bluetooth gamepad with the EV3 brick

For more information about this procedure, see [SIXLINUX].

- Connect the PS3 gamepad to a Linux PC with a USB cable.
- In a shell on the PC:

```
user@PC $ wget http://www.pabr.org/sixlinux/sixpair.c
user@PC $ gcc -o sixpair sixpair.c -lusb
user@PC $ sudo ./sixpair 11:22:33:44:55:66
Current Bluetooth master: xx:xx:xx:xx:xx:xx
Setting master bd_addr to 11:22:33:44:55:66
```

(Replace 11:22:33:44:55:66 with the Bluetooth address of the EV3 brick from a previous step.)

- Disconnect the PS3 gamepad from the PC.

4.4. Using brickhid

- Press the PS button of the PS3 controller. When the PS3 controller successfully connects to the EV3 brick, **brickhid** will play a high-pitched tone.
- From then on **brickhid** translates stick positions and button presses into motor commands as specified by its configuration file.
- You can terminate **brickhid** by pressing the PS button again. Alternatively, you can issue Control+C on the command-line or press the reset button on the back of the PS3 controller with a paper clip.

5. Known problems

- To use Mindstorms applications which connect over Bluetooth simultaneously with the gamepad, you may need to edit `/etc/bluetooth/main.conf`, uncomment the line which says **DisablePlugins=network,input**, and restart **bluetoothd** instead of disabling it entirely.
- The TCP interface of the Mindstorms runtime seems to be available only when WiFi is active. Disabling WiFi manually in the settings menu or unplugging the WiFi adapter will terminate **brickhid**.

6. Perspectives and future work

- Support more HID devices.
- Make HID peripherals appear like EV3 sensors to the runtime so that EV3 programs can use them.
- Support HID feedback functions such as rumble on the DualShock 3 and LEDs on the PlayStation Move.

Bibliography

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

[BRICKFPV] *First-person view and remote control with LEGO Mindstorms EV3*. <http://www.pabr.org/bricks/brickfpv/brickfpv.en.html> .