

[Skip to content](#)
[Imaginary Industries](#)

Electronics, Photography, and boredom.

- [Home](#)
- [Computers](#)
- [Electronics](#)
- [Photography](#)
- [Python](#)
- [Uncategorized](#)
- [Venting](#)

Turning an Arduino Leonardo into a joystick.

2012 November 3
by Connor

I recently bought a Arduino Leonardo in the interest of updating the interface in an old arcade-style controller I occasionally use to play [MAME](#) games.

As such, I needed to figure out how to get the leonardo to properly act like a joystick.

Anyways, this is heavily based on drake250's work, which is on the freetronics forum [here](#). HelmPCB's [USB Joystick](#) site was also a useful reference.

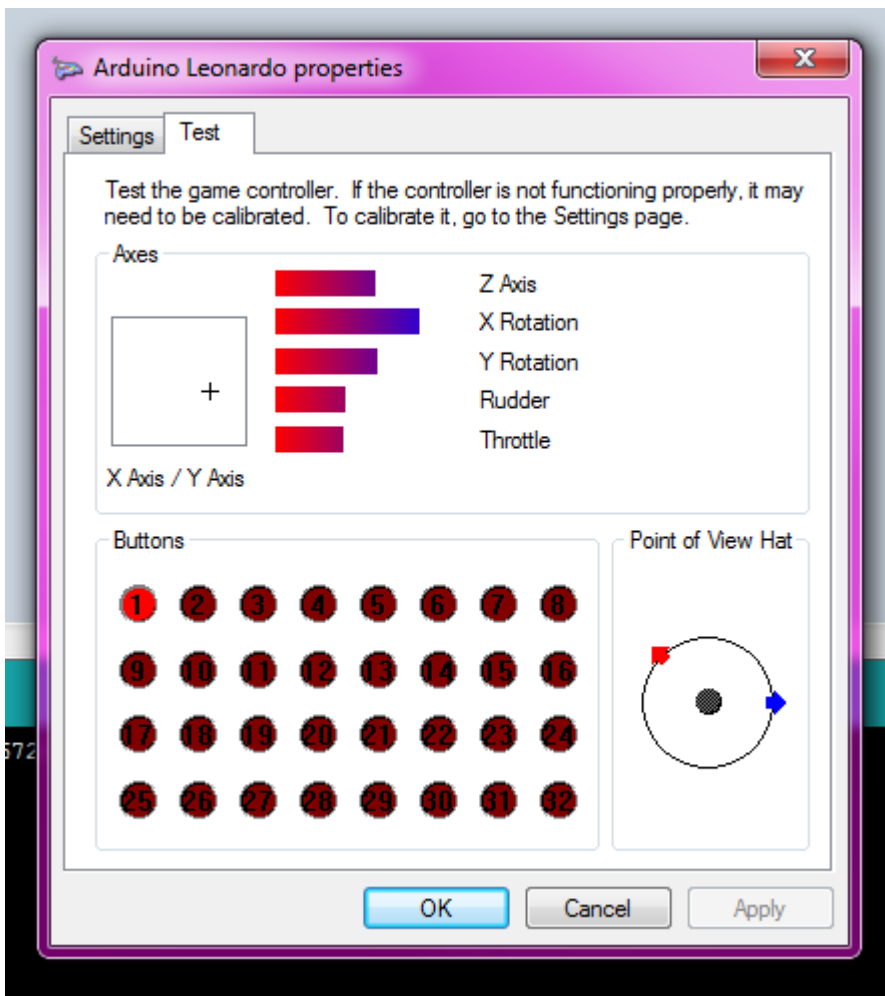
There is a useful list of HID descriptors [here](#).

Also, the "HID Descriptor Tool" from usb.org is also very useful for seeing how HID descriptors are put together. [Get it here](#).

In the end, I wound up rewriting most of the drake250's HID descriptor myself, based on some of the USB.org docs, and the HID Descriptor Tool.

Anyways, here are my modified files you would need to turn your own leonardo into a joystick. These files support an 8-axis joystick, with two hat-switches, and 32 buttons.

If you want fewer buttons/axes/whatever, you can either just ignore the things you don't need, or modify the HID descriptor yourself. As it is, for everything the HID descriptor currently specifies, it only needs 12 bytes per PC-update, so I'm not too worried about the extra axes causing issues or slowing things down.



The windows joystick info dialog only seems to be able to display 7 axes. I'm *fairly* sure the eighth axis there. However, since the built-in windows dialog works *almost* the time, I have not found any tools for looking at joystick readouts other than the built-in windows dialog. I mean, how many people have an 8-axis joystick anyways?

Also, the HID documents seem to also support the belief that the HID spec allows a maximum of 63 buttons. However, again, the windows tools for inspecting a joystick only support 32. You could add more quite easily, but you would have to test their functionality yourself.

Modified arduino files (replace the files in {arduino executable dir}\hardware\arduino\cores\arduino\ with these):

[USBAPI.h](#)
[HID.cpp](#)

And the main source file:

[leoJoy.ino](#)

It's contents:

```
7 JoyState_t joySt;           // Joystick state structure
8
9 void setup()
10 {
11     pinMode(13, OUTPUT);
```

```
12
13     joySt.xAxis = 0;
14     joySt.yAxis = 0;
15     joySt.zAxis = 0;
16     joySt.xRotAxis = 0;
17     joySt.yRotAxis = 0;
18     joySt.zRotAxis = 0;
19     joySt.throttle = 0;
20     joySt.rudder = 0;
21     joySt.hatSw1 = 0;
22     joySt.hatSw2 = 0;
23     joySt.buttons = 0;
24
25 }
```

The interesting thing here is JoyState_t, which is a struct that stores the state for all the joystick controls.

The struct is defined in USBAPI.h as such:

```
53 typedef struct JoyState // Pretty self explanatory. Simple state to
54 {
55     uint8_t      xAxis;
56     uint8_t      yAxis;
57     uint8_t      zAxis;
58
59     uint8_t      xRotAxis;
60     uint8_t      yRotAxis;
61     uint8_t      zRotAxis;
62
63     uint8_t      throttle;
64     uint8_t      rudder;
65
66     uint8_t      hatSw1; // 0-7 correspond to 0-315° in 45°
67     uint8_t      hatSw2; // All other values are invalid
68
69     uint32_t      buttons; // 32 general buttons (Each bit co
70
71 } JoyState_t;
```

Continuing on with the contents of leoJoy.ino:

```
29 void loop()
30 {
31
32
33
34     joySt.xAxis = random(255);
35     joySt.yAxis = random(255);
36     joySt.zAxis = random(255);
37     joySt.xRotAxis = random(255);
38     joySt.yRotAxis = random(255);
39     joySt.zRotAxis = random(255);
40     //joySt.throttle = random(255);
41     joySt.rudder = random(255);
42 }
```

```
43     joySt.throttle++;
44
45
46     joySt.buttons <=& 1;
47     if (joySt.buttons == 0)
48         joySt.buttons = 1;
49
50     joySt.hatSw1++;
51     joySt.hatSw2--;
52
53     if (joySt.hatSw1 > 8)
54         joySt.hatSw1 = 0;
55     if (joySt.hatSw2 > 8)
56         joySt.hatSw2 = 8;
57
58     delay(100);
59
60     if (joySt.throttle > 127)
61         digitalWrite(13, HIGH);
62     else
63         digitalWrite(13, LOW);
64
65
66     // Call Joystick.move
67     Joystick.setState(&joySt);
68
69 }
```

This section just generates random values and sets the analog axes to them. Each bit in `buttons` is also set in sequence, using bit-shifting. It also steps through the different hat-switch positions. It steps `hatSw1` forwards, and `hatSw2` backwards. Lastly, the throttle axis is linearly ramped.

The last (and most important thing) here is

```
67     Joystick.setState(&joySt);
```

This is the function that actually sends the values in the `joySt` struct to the computer.

At this point, all that is needed to make a useable joystick is to replace the contents of `loop()` with something that actually reads hardware buttons.

Anyways, everything is under the arduino license (I think it's GPL, but the only file I edited with a license actually looks to be BSD-ish. Anyways, do whatever, just don't be an ass).

from → [Electronics](#)

101 Responses [leave one](#) →



1. Colin [permalink](#)
July 16, 2014

I want to write a program is 5IO 25KEY ,but I don't know hwo to write,there is a example:

```
L2 = L3 = L4 = L5 = 1;L1 = 0;
```

```
delay (); if(L2==0) {
```

```
L1 = 1; //释放人工置低的线 if(L2==0) key = 22; else key = 1; } }
```

```
if(L3==0) { delay (); if(L3==0) { L1 = 1; if(L3==0) key = 23; else key = 2; } } if(L4==0) { delay (); if(L4==0) { L1 = 1; if(L4==0) key = 24; else key = 3; } } if(L5==0) { delay ();
```

```
if(L5==0) { L1 = 1; if(L5==0) key = 25; else key = 4; } }
```

```
L1 = L3 = L4 = L5 = 1;L2 = 0; if(L1==0) { delay (); if(L1==0) { L2 = 1; if(L1==0) key = 21; else key = 20; } } if(L3==0) { delay (); if(L3==0) {
```

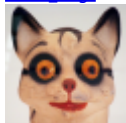
```
L2 = 1; if(L3==0) key = 23; else key = 5; } } if(L4==0) { delay ();
```

```
if(L4==0) { L2 = 1; if(L4==0) key = 24; else key = 6; } } if(L5==0) {
```

```
delay (); if(L5==0) { L2 = 1; if(L5==0) key = 25; else key = 7; } }
```

can you help me,thanks a lot

[Reply](#)



2. Vindolin [permalink](#)
August 15, 2014

Thanks for the cool blog post!

I made an Instructable based on the info here, for adding a little two axis joystick to a bigger joystick to controll vertical/lateral thrust in Elite Dangerous 😊

<http://www.instructables.com/id/Add-a-little-two-analog-axis-thumb-joystick-to-you/>

[Reply](#)



3. Doveman [permalink](#)
September 3, 2014

I've got a Thrustmaster F16 stick and want to wire up the buttons/hats to a Pro Micro to make it a USB device.

I found this post with code for doing this with a Teensy, which is almost compatible with the Pro Micro except for the Joystick. commands which are integrated into the Teensy code.

<http://forum.il2sturmovik.com/topic/1790-there-any-better-joystick-ms-sidewinder-force-feedback-2/#entry46472>

Would anyone be able to help me take that code and combine it with the USB HID code from here? I've already flashed the latter to my Pro Micro and that works fine and shows random data being received on the joystick setup screen in Windows

[Reply](#)



4.

patrick [permalink](#)
September 7, 2014

Hello, how do i modify the code for 10 bit analog ?

Can i just replace on of the uint8_t with uint10_t ?

[Reply](#)

[Newer Comments »](#)

Leave a Reply

Name: :

Email: :

Website:

Comment:

Note: You can use basic XHTML in your comments. Your email address will **never** be published.

[Subscribe to this comment feed via RSS](#)

Submit Comment

• Search

type and press enter

• Categories

- [Computers](#)
- [Electronics](#)
- [Photography](#)
- [Power-Meter Project](#)
- [Programming](#)
- [Python](#)
- [Uncategorized](#)
- [Venting](#)

• Archives

- [September 2014](#)
- [July 2014](#)
- [June 2014](#)
- [March 2014](#)
- [February 2014](#)
- [January 2014](#)
- [November 2013](#)
- [October 2013](#)
- [September 2013](#)
- [July 2013](#)
- [June 2013](#)
- [May 2013](#)
- [March 2013](#)
- [February 2013](#)
- [January 2013](#)
- [November 2012](#)
- [June 2012](#)
- [May 2010](#)
- [January 2010](#)
- [November 2009](#)
- [October 2009](#)
- [September 2009](#)

Copyright 2014

[Vigilance Theme](#) by [The Theme Foundry](#)