

[Ubuntu Forums](#) > [The Ubuntu Forum Community](#) > [Other Community Discussions](#)
> [Tutorials & Tips](#) > [How To] Installing/Using Nostromo N52 with Linux

[Log in](#)

View Full Version : [\[How To\] Installing/Using Nostromo N52 with Linux](#)

imlost

October 15th, 2008, 04:12 PM

The Nostromo N52. If you're a gamer and you have this baby, chances are you won't game without it, trust me I know. This is why when I ditched Windows for Ubuntu Ultimate 1.9 Linux, getting my Nostromo to work with full functionality in Linux was a VERY big deal for me.

There are 2 ways you can get the Nostromo N52 to work in Linux.

Download the Nostromo 0.1.3 or 0.1.4 drivers and follow one of the several guides out there that try to explain how to get it working.

Or

Download Pystromo (<https://launchpad.net/pystromo/+download>) from Raumkraut (<http://raumkraut.net/>) and follow my guide [here](#).

The down and dirty on the Nostromo Linux drivers from Sourceforge is that they work but are glitchy when it comes to macros and repeating. I have my Nostromo spam the button I am pressing every 100 milliseconds. With the Nostromo Linux drivers this will freeze up my game. At 200 milliseconds it does work but will glitch out and lock into a spam and not stop spamming the key when I release the key. I then have to press the key again to get it to stop.

That being said... I suggest you steer clear of the Nostromo Linux drivers!

Raumkraut has worked very hard to create this program and give it to us for free, don't let his hard work go to waste.

Raumkraut has included a README file with his Pystromo that details how to install it and what files to look into to learn how to configure it. However for those of us who, like me, need lengthy step by step instructions to get the job done, keep reading.

Raumkraut's README file is basically the installation instruction file.

His test.map file located in the config directory is his instructions on how to program Pystromo.

His constants.py script shows every key Pystromo understands and can bind to

your Nostromo.

Here goes the tutorial.

Raumkraut's website for Pystromo is located here.

<https://launchpad.net/pystromo> (<https://launchpad.net/pystromo>)

1. Download version 0.6.0 (<http://launchpad.net/pystromo/trunk/0.6.0/+download/pystromo-0.6.0.tar.gz>) of Pystromo.

2. Go to `/home/username/.config`

3. Create a folder in this directory called "pystromo" (without quotes)

4. Extract the files from the 0.6.0 tar you downloaded to this new pystromo directory.

5. Open a terminal window and do the following commands.

```
sudo gedit /etc/modules
```

This will open the file "modules" with the gedit program.

6. In there add "uinput" (without the quotes) to the file and save and close it.

7. Open another terminal window and do the following command.

```
sudo modprobe uinput
```

This will eliminate the need to reboot after adding that line to the modules file.

8. Now close that terminal window.

9. Now if you are running Ubuntu open another terminal window and do the following command.

```
sudo cp /home/username/.config/pystromo/config/52-pystromo-debian.rules  
/etc/udev/rules.d/
```

10. Now close that terminal window and open another terminal window and do the following.

*****ATTENTION***** -This section is only for those using this guide to install Pystromo to use with another game controller. If you are using the Nostromo N52 or Nostromo N50 skip this section. ***NOTE*** If you are using the Nostromo N52te version, you will need to follow this section of the guide. (To be removed later: If someone using this guide has a Nostromo N52te and has success with it, please

respond to my post with the vendor and product number you received when following this guide.)

lsusb

This should output a list of USB devices connected to the computer. Now I use a Belkin Nostromo N52 and the line for my controller looks like this:

Bus 002 Device 004: ID 050d:0815 Belkin Components

But yours should say something with the name or brand of your controller in it.

What we are looking for is the Vendor number and Product number. Notice the parts I have in red and blue in the above line. The red part is the vendor number and the blue part is the product number.

Write this part from your game controller line down. We will be adding it to one of Pystromo's config files.

Now from a terminal window do the following command.

```
gedit /home/username/.config/pystromo/default.map
```

In this file put the following info.

```
# Descriptive name of your controller.  
[Device:mypad]  
vendor=0x(Put in the red numbers I had you right down)  
product=0x(Put in the blue numbers I had you right down)
```

```
[Map:mypad]
```

End Optional Section

Those who followed the above optional section due to having an originally unsupported controller should skip this next part.

11. Now from a terminal window do the following command.

```
gedit /home/username/.config/pystromo/default.map
```

In this file put the following info.

```
# Nostromo N52 Speedpad  
[Device:n52]  
vendor=0x050d
```

```
product=0x0815
```

```
[Map:n52]
```

****Everyone should be following the guide now****

12. Now save and close that file.

13. Open another terminal window and do the following command.

```
gedit /home/username/.config/pystromo/monitor.conf
```

14. Now copy and paste the following information.

```
# This is the configuration file for Pystromo's monitoring script.
# Each line herein specifies a mapping file to use when certain
# processes are detected as running.
# With the exception of "*", whitespace in the key IS NOT ignored.
# "=" is an invalid character in keys.
```

```
# The * defines mapping files to use all the time
*=default.map
```

15. Now save and close this file.

16. Now open another terminal window and do the following command.

```
gedit /home/username/.config/pystromo/default.map
```

Now here we want to put the information all after the last line in the current file, the [Map:n52] part.

Basically what we are doing now is telling Pystromo first what the button on the Nostromo we are wanting to configure is bound to by default.

17. The easiest way to do this is write the following to the file.

```
KEY_
```

18. You will then press the button on the Nostromo that you want to configure. This will output a character to your document or it will do something else, such as shift or capslock or tab.

If you want to know *precisely* what value to use (even for keys with non-visible output), do the following.

Open a terminal window and put in these commands.

```
gedit /home/username/.config/pystromo/output.map
```

In this file put the following text and save and close it.

```
# Nostromo N52 Speedpad  
[Device:n52]  
vendor=0x050d  
product=0x0815
```

```
[Map:n52]
```

Now open a terminal window and put in the following commands.

```
cd /home/username/.config/pystromo  
./pystromo-remap.py -m output.map -vv
```

You can now press any button on the Nostromo and it will output data in this terminal window telling you what exactly is bound to that key.

Now back to configuring our default.map file.

Let's say it output the character "A", but the character will be lower case so it will look like this:

```
KEY_a
```

19. Delete the lower case "a" and replace it with an upper case "A"

```
KEY_A
```

20. We will now tell it what we want to bind that key to.

```
KEY_A:KEY_1
```

This will bind the key "1" to the Nostromo key we are programming.

21. Do this for each key that you want to program. For a list of keys Pystromo can understand and bind to your Nostromo open up the constants.py file with a text editor. That file is located here:

```
/home/username/.config/pystromo/lib/constants.py
```

Now for specific programming such as delays or having the key repeat itself every so many milliseconds take a look at the test.map file located at: /home/username/.config/pystromo/config/test.map

It gives a list of commands you can use to configure it.

22. Once you have every key you want configured on the Nostromo open a terminal window and type the following commands.

```
cd /home/username/.config/pystromo
./pystromo-remap.py -m default.map
```

23. Now minimize this terminal window.

24. Open a text editor and test your Nostromo by pressing its keys. If all went well it should now be outputting whatever key you programmed to each key on the Nostromo!

Note if this still does not cause your Nostromo to use the keys you programmed to it, find the python process in your System Monitor and kill it. Then run the following commands from a terminal

```
cd /home/username/.config/pystromo
sudo ./pystromo-remap.py -m config/default.map
```

The reason it wouldn't run without sudo has something to do with the .rules file. Raumkraut would know about that, he made Pystromo... I personally have no clue how to program and am only providing a detailed guide to his excellent work.

If you were able to run pystromo-remap.py without the "sudo" command then you can follow this next section to have Pystromo run when a certain process starts.

25. Open your System Monitor and minimize the window.

26. Run the program or game you wish to use your Nostromo with.

27. Find this program or game's process in the System Monitor. Write it down.

28. Open a terminal window and do the following commands.

```
cd /home/username/.config/pystromo
gedit monitor.conf
```

The file should look like this from before:

```
# This is the configuration file for Pystromo's monitoring script.
# Each line herein specifies a mapping file to use when certain
# processes are detected as running.
# With the exception of "*", whitespace in the key IS NOT ignored.
# "=" is an invalid character in keys.
```

```
# The * defines mapping files to use all the time
```

```
*=default.map
```

29. Edit the "`*=config/default.map`" and replace the "*" with the process name of the program you wish to use your Nostromo with. For instance if you wanted to use it with the process "wow.exe" it would look like this:

```
# This is the configuration file for Pystromo's monitoring script.  
# Each line herein specifies a mapping file to use when certain  
# processes are detected as running.  
# With the exception of "*", whitespace in the key IS NOT ignored.  
# "=" is an invalid character in keys.
```

```
# The * defines mapping files to use all the time  
wow.exe=default.map
```

30. Save and close this window.

31. Right click on your desktop and select "Create Launcher".

Name it whatever you want but put the following line in the command text box.

```
/home/username/.config/pystromo/./pystromo-mon.py
```

32. The last step of all, but the most important... *****MOST IMPORTANT***** Enjoy your Nostromo in Linux and be sure to stop by Raumkraut's website (<http://raumkraut.net/>) and thank him for his hard work!

Good luck and happy gaming!!

Raumkraut

October 16th, 2008, 12:43 AM

2. Go to `/home/username/.config`

3. Create a folder in this directory called "pystromo" (without quotes)

4. Extract the files from the 0.6.0 tar you downloaded to this new pystromo directory.

You don't actually have to put the Pystromo program itself in here. In fact, you don't even have to use that particular location at all, that's just the default place it looks for config files. Arbitrary locations for the config files can be specified at the command line.

```
modprobe uinput
```

This'll need to be done under sudo, as well.

```
gedit /home/username/.config/pystromo/config/default.map
```

The 'config' directory of Pystromo was intended for out-of-the-box mappings and such. While there's no reason you can't put your own files in there, the intended location for such files was "/home/username/.config/pystromo".

I never intended the program itself to go in that directory, just the configuration - hence the ".config" part! :)

```
gedit /home/username/.config/pystromo/config/default.conf
```

Arf, that default.conf file is something of a relic from the days when the "device" and "map" sections of the mapping files were actually two separate files.

I keep it around for if/when Pystromo ever needs some non-mapping configuration options. :)

18. You will then press the button on the Nostromo that you want to configure. This will output a character to your document or it will do something else, such as shift or capslock or tab.

If you want to know *precisely* what value to use (even for keys with non-visible output), you can run the `pystromo-remap.py` script using a mapping file with at least a relevant [device] section, and the `-vv` option. This will show you all of the 'events' coming from your input device. Eg:

```
$ ./pystromo-remap.py -m /path/to/my.map -vv
```

```
Using output: <lib.devices.OutputDevice object at 0x825ea8c>
```

```
Loading mappings
```

```
config/test.map
```

```
Using device: <InputDevice "Genius-KB21e" on 0 device nodes>
```

```
Using device: <InputDevice "hand-track" on 0 device nodes>
```

```
Using device: <InputDevice "n52" on 2 device nodes>
```

```
Using device: <InputDevice "n50" on 0 device nodes>
```

```
Using device: <InputDevice "mx510" on 1 device nodes>
```

```
Using device: <InputDevice "saitek-p2500" on 0 device nodes>
```

```
Incoming event: <Event timestamp=1224130728.5547221, type='EV_KEY',  
code='KEY_Q', value=1>
```

```
Incoming event: <Event timestamp=1224130728.618726, type='EV_KEY',  
code='KEY_Q', value=0>
```

```
Incoming event: <Event timestamp=1224130731.130868, type='EV_KEY',  
code='KEY_CAPSLOCK', value=1>
```

```
Incoming event: <Event timestamp=1224130731.210865, type='EV_KEY',  
code='KEY_CAPSLOCK', value=0>
```

```
Incoming event: <Event timestamp=1224130733.499002, type='EV_KEY',  
code='KEY_LEFTSHIFT', value=1>
```

```
Incoming event: <Event timestamp=1224130733.5790069, type='EV_KEY',  
code='KEY_LEFTSHIFT', value=0>
```

Here we see the output of sequentially pressing buttons #02 (KEY_Q), #06 (KEY_CAPSLOCK) and #11 (KEY_LEFTSHIFT) on my n52.


```
/home/username/.config/pystromo/lib/constrants.py
```

I think you mean "constants.py" :)

```
/home/username/.config/pystromo/./pystromo-mon.py
```

If you want to run the monitor without arguments, you need to use a config file called monitor.conf located in /home/username/.config/pystromo (rather than config/default.conf as specified earlier). You can however specify an arbitrary config file using the -c command-line argument.

And don't forget to at least mention the included README file! It may not as detailed as this HOWTO, but it's guaranteed* to be accurate!

* - guarantee not valid outside of my dream-world. ;)

imlost

October 16th, 2008, 01:13 AM

Thank you for the corrections! I will fix those points in my guide tomorrow.

imlost

October 17th, 2008, 02:36 PM

Ok, I went back through the guide a few times and believe I have got all the kinks and errors out of it. If you are using my guide and come across a discrepancy in it, let me know.

If you are using my guide and follow everything correctly and it still doesn't work or gives you an error, contact Raumkraut at his Pystromo (<https://launchpad.net/pystromo>) website.

jeffster23

November 20th, 2008, 10:43 AM

I found this site helpful:

<http://micolous.id.au/papers/speedpad-n52/>

It lists the default N52 key mappings in Linux, which I have translated below to N52 parlance. Figured it might save some people out there a few minutes. Of course, these entries do not yet have the actual key I want mapped to each.

```
# Nostromo N52 Speedpad
[Device:n52]
vendor=0x050d
product=0x0815
```

```
[Map:n52]
#Top row, from left
KEY_TAB:
```

```
KEY_Q:
KEY_W:
KEY_E:
KEY_R:

#Middle row, from left
KEY_CAPSLOCK:
KEY_A:
KEY_S:
KEY_D:
KEY_F:

#Bottom row, from left
KEY_LEFTSHIFT:
KEY_Z:
KEY_X:
KEY_C:

#Wheel (acts as mouse wheel)
REL_WHEEL>=
# Remap the negative wheel motion to "-"
REL_WHEEL<=
BTN_MIDDLE:

#D-Pad - "up" is more like forward
KEY_UP:
KEY_LEFT:
KEY_DOWN:
KEY_RIGHT:

#orange button
KEY_LEFTALT:

#button under thumb
KEY_SPACE:
```

Sugi

January 4th, 2009, 01:30 AM

This application is amazing and thanks for the guide. I had some issues with it at first, but except for that everything went smoothly. Pystromo works great with my N52te gamepad. :D

Thanks,
Sugi

BrokenPoet

January 14th, 2009, 12:44 AM

In response to the earlier request for N52TE information:

Bus 004 Device 005: ID 050d:0200 Belkin Components

Only the Product ID changed, apparently.

nafriel

January 29th, 2009, 08:07 PM

```
cd /home/username/.config/pystromo
./pystromo-remap.py -m output.map -vv
```

You can now press any button on the Nostromo and it will output data in this terminal window telling you what exactly is bound to that key.

When I try this part, I it fails. Here's the output I see:

```
Using output: <lib.devices.OutputDevice object at 0x7f25ed9d1690>
Loading mappings
output.map
Traceback (most recent call last):
File "./pystromo-remap.py", line 170, in <module>
inputs = getInputs(keymap, output)
File "./pystromo-remap.py", line 98, in getInputs
dev = devices.InputDevice(id=device, keymap=keymap, output=output, **params)
File "/home/eric/.config/pystromo/lib/devices.py", line 99, in __init__
node.grab()
File "/home/eric/.config/pystromo/lib/ioctl.py", line 211, in grab
fcntl.ioctl(self.fd, const.EVIOCGRAB, grab)
IOError: [Errno 16] Device or resource busy
```

22. Once you have every key you want configured on the Nostromo open a terminal window and type the following commands.

```
cd /home/username/.config/pystromo
./pystromo-remap.py -m default.map
```

I get the exact same response after this part. Can anyone help me figure out what I've done wrong?

Thanks much!

Raumkraut

January 30th, 2009, 03:49 PM

When I try this part, I it fails. Here's the output I see:

```
IOError: [Errno 16] Device or resource busy
```

Please try the proposed workaround in this bug report: <https://bugs.launchpad.net/pystromo/+bug/287457>

nafriel

January 30th, 2009, 07:52 PM

I followed the steps on that page exactly and now neither my keyboard or mouse respond in the Ubuntu UI. (I'm posting from my roommate's computer.) I'm assuming the Identifiers, Drivers and Options simply didn't match what i actually have... do you know how to find the right ones?

Thanks!

nafriel

January 30th, 2009, 08:17 PM

Okay, please disregard the last post... very much user error, there.

I got the following command to work:
`./pystromo-remap.py -m output.map -vv`

It now shows all the events in the console window as desired. I then tried:
`./pystromo-remap.py -m default.map`
and got the same error I was asking about initially.

I'm pretty new to Linux, so maybe this is pointless, but I'm going to try rebooting the computer and see if I can get the latter command to work if I haven't already entered the former.

nafriel

January 30th, 2009, 08:22 PM

My God! It worked!

Thankyouthankyouthankyou

For making pystromo, for replying to my cry for help...

Thank you so much!

nafriel

February 1st, 2009, 03:37 PM

31. Right click on your desktop and select "Create Launcher".

Name it whatever you want but put the following line in the command text box.

`/home/username/.config/pystromo/./pystromo-mon.py`

Just curious, what is the point of this part? I followed this step, and I tried double-clicking the launcher, but it doesn't seem to do anything...

B0rat

February 16th, 2009, 07:41 PM

I followed the steps on that page exactly and now neither my keyboard or mouse

respond in the Ubuntu UI.

This happens to me also! I had to boot into another distro and remove the offending fdi file from /usr/share/hal/fdi/policy/20thirdparty/

```
./pystromo-remap.py -m output.map -vv
Using output: <lib.devices.OutputDevice object at 0x9a84f2c>
Loading mappings
output.map
Traceback (most recent call last):
File "./pystromo-remap.py", line 170, in <module>
inputs = getInputs(keymap, output)
File "./pystromo-remap.py", line 98, in getInputs
dev = devices.InputDevice(id=device, keymap=keymap, output=output, **params)
File "/home/john/.config/pystromo/lib/devices.py", line 99, in __init__
node.grab()
File "/home/john/.config/pystromo/lib/ioctl.py", line 211, in grab
fcntl.ioctl(self.fd, const.EVIOCGRAB, grab)
IOError: [Errno 16] Device or resource busy
```

Even using sudo does nothing. Just as above.

snaphthetics

March 3rd, 2009, 09:56 AM

I have been racking my brains on this for a week. I tried to figure it up to my Ubuntu noobness. When I run ./pystromo-remap.py -m output.map -vv I get:

```
Using output: <lib.devices.OutputDevice object at 0x94b3f2c>
Loading mappings
output.map
Traceback (most recent call last):
File "./pystromo-remap.py", line 170, in <module>
inputs = getInputs(keymap, output)
File "./pystromo-remap.py", line 98, in getInputs
dev = devices.InputDevice(id=device, keymap=keymap, output=output, **params)
File "/home/justin/.config/pystromo/lib/devices.py", line 99, in __init__
node.grab()
File "/home/justin/.config/pystromo/lib/ioctl.py", line 211, in grab
fcntl.ioctl(self.fd, const.EVIOCGRAB, grab)
IOError: [Errno 16] Device or resource busy
```

I went on and followed the rest of the steps and my n52 stays with the factory default keys.

snaphthetics

March 3rd, 2009, 06:00 PM

Please try the proposed workaround in this bug report: <https://bugs.launchpad.net/pystromo/+bug/287457>

I followed these steps and lost control of my keyboard and mouse. What can I do about that?

B0rat

March 3rd, 2009, 06:06 PM

remove the offending fdi file from /usr/share/hal/fdi/policy/20thirdparty/

I did this by booting into another distro I had on my HD. You may have to use a livecd if you can't do this.

If you have a root account on your ubuntu then you could boot into single user mode and remove it. HTH.

snaphthletics

March 4th, 2009, 12:30 PM

I did this by booting into another distro I had on my HD. You may have to use a livecd if you can't do this.

If you have a root account on your ubuntu then you could boot into single user mode and remove it. HTH.

Ya, I was able to get back up and running. But still unable to map the n52. Is there a bug fix for the bug fix?

B0rat

March 4th, 2009, 01:21 PM

Some people have got this to work but not me.
Something is grabbing the n52 and not allowing it to be used. This has only happened since upgrading to Intrepid.

B0rat

March 4th, 2009, 04:45 PM

snaphthletics, look at this thread (https://sourceforge.net/forum/forum.php?thread_id=2058408&forum_id=207570)

It fixed my problem :)

Seems like xorg is to blame.

snaphthletics

March 5th, 2009, 07:46 PM

snaphthletics, look at this thread (https://sourceforge.net/forum/forum.php?thread_id=2058408&forum_id=207570)

It fixed my problem :)

Seems like xorg is to blame.

lol, This ended up helping in the end. I have used the corrections from all the links below and I am still not running in Pystromo.

"Device or resource busy" under Intrepid (<https://bugs.launchpad.net/pystromo/+bug/287457>)

Cant grab device after X is restarted (http://sourceforge.net/forum/forum.php?thread_id=2058408&forum_id=207570)

evdev mouse fails on hardy: cannot open input pEvdev (<https://bugs.launchpad.net/ubuntu/+source/xserver-xorg-input-evdev/+bug/173833>)

Originally I tried the Nostromo drivers. I was able to open the config window but not able to remap or communicate with the n52. A fresh install later I tried Pystromo and after the open bug fix suggested I lost control of all my devices. At that time I did not know how to fix it so another fresh install. Again ran through all of the steps of pystromo including bug fix but before I restarted I added the steps to the xorg file to try and keep from loosing my mouse and keyboard again. More success this time with Pystromo.

Now I was able to grab input keys from the command `./pystromo-remap.py -m output.map -vv`. However I was still not getting any response remapping my keys.

Now I decided since the issue was xorg then maybe the Nostromo drivers would work. Went through setup again and boom, remapped keys at the end.

I would have wrote a more detailed description of my exact steps except I really don't know what actually corrected what and I swear I have some adult ADD.

What were we talking about again?

MisteR2

March 24th, 2009, 08:22 PM

Hey everybody. Found a solution to evdev grabbing everything it can get it's hands on. It's a modification of a posting I found here. (<http://bbs.archlinux.org/viewtopic.php?id=60986>)

Basically I blacklisted my Nostromo to HAL, and it worked right away. Keyboard and mouse still work (outside of some issues external to this) and this is with a `xorg.conf` not too far from stock (ie. keyboard and mouse are pretty much undefined).

Lat.

vBulletin® v3.8.4, Copyright ©2000-2011, Jelsoft Enterprises Ltd.