

## Search for drivers



There are often cases when a couple of devices does not work properly in your computer out of the box under Linux. The reason for this may be too new hardware (not yet implemented in the kernel), the absence of necessary Linux drivers (not provided by hardware vendors), too obsolete hardware, incompatible devices (e.g. storage controller and drive model, etc.) or a defect. According to data from the [Linux-Hardware.org](http://Linux-Hardware.org), at least 10% of Linux users encounter such problems. According to our [statistics](#), the most problematic devices are:

- WiFi cards
- Bluetooth cards
- Card readers
- Fingerprint readers
- Smart card readers
- Printers
- Scanners
- Modems
- Graphics cards
- Webcams
- DVB cards
- Multimedia controllers

If a device does not work, then this does not mean that it cannot be configured properly. Sometimes you can find a more suitable kernel or a third-party driver for a device on the Internet. To search for the required kernel, you can use the [LKDDb](#) database, where all Linux kernel versions are indexed for supported device drivers, or search for a solution on forums and similar resources.

Today we are launching a new way to find drivers — by creating of hardware probes! If a driver was not loaded for some device in your computer probe, then the database engine [will offer](#) a suitable kernel version or known third-party drivers. The same information is [presented](#) on each PCI/USB device page in the database.

A probe can be created by the following command (there is an [AppImage](#) for other Linux distributions):

```
hw-probe -all -upload
```

The drivers search is carried out on the basis of [LKDDb](#) database for kernels from 2.6.24 to the newest 5.0 version. Also we have indexed the following third-party drivers:

- nvidia — NVIDIA graphics
- wl — Broadcom WiFi-cards
- fglrx — AMD/ATI graphics
- hsfmodem — modems
- sane — scanners
- foomatic — printers
- gutenprint — printers
- and about 100 drivers from Github for WiFi cards and other devices

Yes, we have indexed fglrx for old AMD/ATI graphics cards. In some cases, the performance of this driver was higher than the free radeon driver, but you need to install a previous version of your Linux distribution, because the fglrx driver is not supported by newest Linux kernels and Linux distributions.

This is not the only example when you need a rollback to an old kernel version. For some devices, the drivers become obsolete and are removed from the kernel. In such cases, you need to pick up one of the old versions of your Linux distribution with an appropriate old kernel, and then manually update necessary software if not provided in the repositories (e.g. browser).

Posted by [Andrey.ponomarenko](#) on 5 March 2019 at 05:40 (GMT).

## Review of hardware probes



Did you manage to configure a hardware device that did not work out of the box? Did you find the right driver? The device does not work and you don't know what to do? Write a note about your experience right now in your hardware probe!

Registration is not needed — authorization of your computer is done while creating a probe. Just create a probe and immediately open it in your browser. You'll see a big green **REVIEW** button on the probe page for creating a review.

In the review, you can adjust the automatically detected operability status of a hardware device and write a comment for any device in [the Markdown](#) language.

Statuses and comments are assigned to the corresponding devices in the open [database](#). Other users will be able to quickly learn from your experience if they have the same problem by creating probes of their computers and following the links to the database on the probe page.

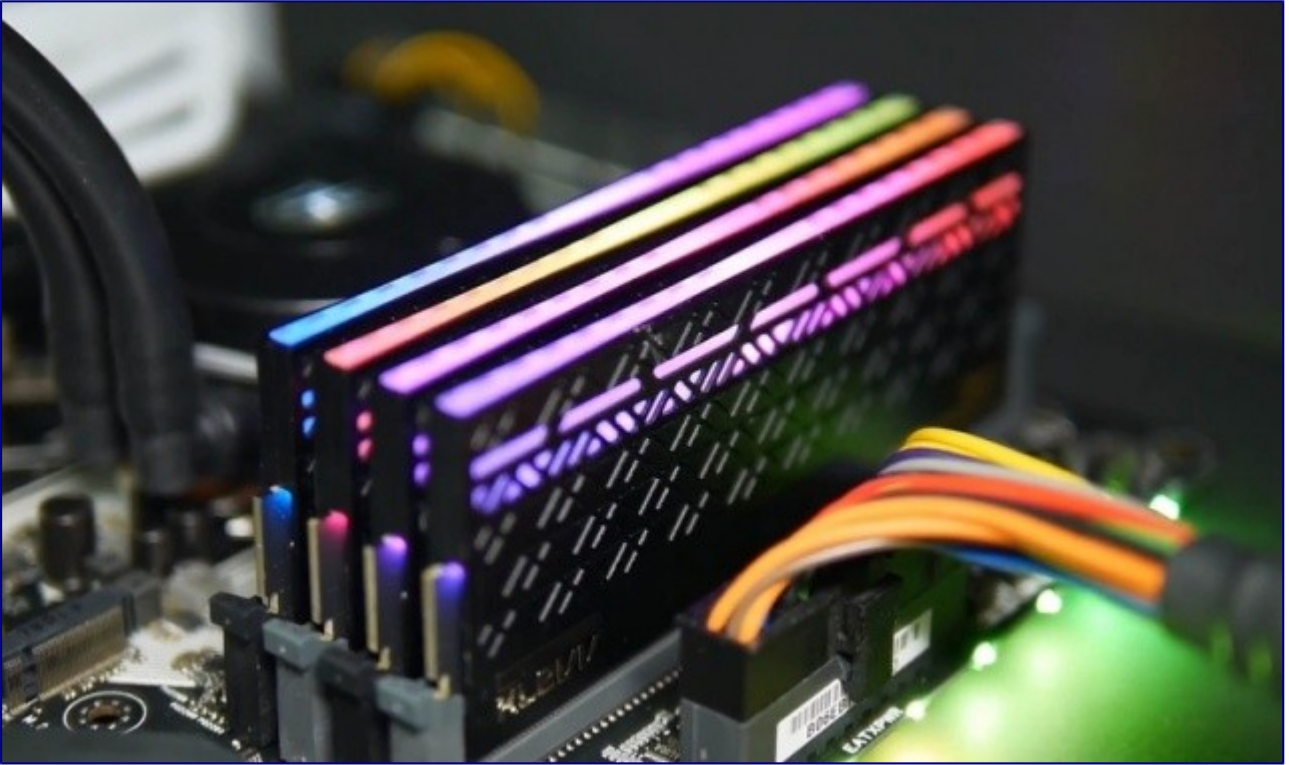
One can create a probe by the command:

```
hw-probe -all -upload
```

Or by the **Hardware Probe** application in the [SimpleWelcome](#) start menu of the ROSA Fresh Linux distribution ([KDE4](#) and [Plasma](#) editions). There is also portable [AppImage](#), [Docker](#), [Snap](#) and [Flatpak](#) to use on any other Linux distribution.

Posted by [Andrey.ponomarenko](#) on 13 February 2019 at 06:30 (GMT).

## Hardware database for all Linux distributions



The [Linux-Hardware.org](https://linux-hardware.org) database has been divided recently into a set of databases, one per each Linux distro. You can choose your favorite Linux distribution on the front page and hide probes and information collected from other Linux distributions.

Anyone can contribute to the database with the help of the [hw-probe](#) command:

```
hw-probe -all -upload
```

Hardware failures are highlighted in the collected logs (important SMART attributes, errors in dmesg and xorg.log, etc.). Also it's handy to search for particular hardware configurations in the community and review errors in logs to check operability of devices on board (for some devices this is done automatically by hw-probe — see statuses of devices in your probe).

Hardware stats and raw data are dumped to several Github repositories: <https://github.com/linuxhw>

Thanks to all for attention and new computer probes!

Posted by [Andrey.ponomarenko](#) on 23 October 2018 at 09:41 (GMT).



## Checking devices operability



We've implemented automated operability checks for devices via analysis of collected system logs in probes. We check if the driver is loaded and used for each device in the probe and if the device performs basic functions. For network cards we check received packets, for graphics cards we check absence of critical errors in the Xorg log and dmesg, for drives we check S.M.A.R.T. test results, for monitors we check the EDID and for batteries we check the remaining capacity.

The operability status is detected for the following devices:

- Graphics cards
- WiFi cards
- Ethernet cards
- Bluetooth cards
- Modems
- Hard drives
- Monitors
- Batteries
- Smart card controllers

For the following devices we can only detect if the device is failed to operate:

- Sound cards
- Card readers

- Fingerprint readers
- TV cards
- DVB cards

You can check all your previous probes now — the statuses are already updated!

If you are a new user, then you can create a probe by the [hw-probe](#) command:

```
hw-probe -all -upload
```

Nonworking devices are collected in the Github repository: <https://github.com/linuxhw/HWInfo>

Thanks to all for attention and new computer probes!

Posted by [Andrey.ponomarenko](#) on 25 July 2018 at 07:00 (GMT).

## EDID repository



The largest open repository of monitor characteristics has been created recently containing *EDID* structures for more than 9000 monitors: <https://github.com/linuxhw/EDID>

EDID (Extended Display Identification Data) is a metadata format for display devices to describe their capabilities to a video source. The data format is defined by a standard published by VESA. EDID data structure includes manufacturer name and serial number, product type, phosphor or filter type, timings supported by the display, display size, luminance data and (for digital displays only) pixel mapping data.

The most famous analogue of the repository is the [EDID.tv](https://www.edid.tv/) project, which also contains quite a lot of information about monitors.

The repository is replenished automatically based on recent hardware probes. One can participate in the replenishment of the repository by executing of one simple command in the terminal:

```
hw-probe -all -upload
```

The hw-probe utility is pre-installed in the ROSA Linux distribution. Users of other systems may use [AppImage](#), [Docker-image](#), [LiveCD](#) or [other techniques](#) to create probes.

Posted by [Andrey.ponomarenko](#) on 20 June 2018 at 07:00 (GMT).

## HW Probe 1.4

# HW PROBE 1.4: anonymization on the client-side



Friends, I'd like to introduce new hw-probe 1.4.

Most significant change in this release is the anonymization of probes on the client-side. Previously "private data" (like IPs, MACs, serials, hostname, username, etc.) was removed on the server-side. But now you do not have to worry how server will handle your "private data", since it's not uploaded at all. You can now upload probes from any computers and servers w/o the risk of security leak.

The update is available in repositories.

Other changes:

- Up to 3 times faster probing of hardware
- Collect SMART info from drives connected by USB
- Initial support for probing drives in MegaRAID
- Improved detection of LCD monitors and drives
- Collect info about MMC controllers
- Probe for mcelog and cpuid
- Etc.

You can, as before, create a probe of your computer via the application in SimpleWelcome menu or from the console by a simple command:



[hw-probe](#) -all -upload

Thanks to all for attention and new probes of computers!

Posted by [Andrey.ponomarenko](#) on 26 April 2018 at 05:00 (GMT).

## Highlighting important SMART attributes in probes

We've started to highlight most important SMART attributes in computer probes, that correlate with real mechanical failures according to Google and Backblaze studies.

Green highlights the zero value of important attributes, red — any positive value. You can review your probes now!

You can find links to the studies, as well as a complete list of highlighted important attributes here: <https://github.com/linuxhw/SMART>

You can, as before, create a probe of your computer via the application in SimpleWelcome menu or from the console by a simple command:

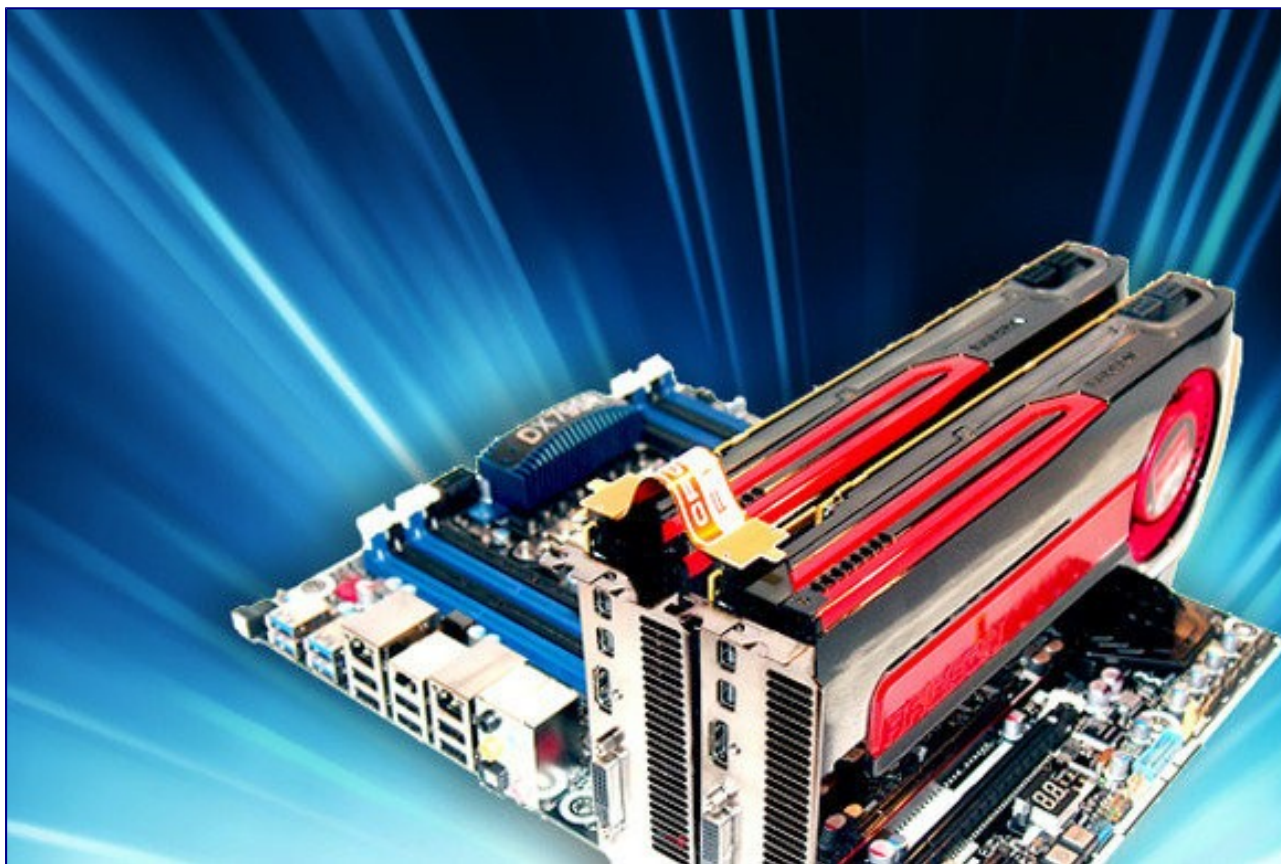
```
hw-probe -all -upload
```

Thank you all for the attention and new hardware probes!

SMART Attributes Data Structure revision number: 16							
Vendor Specific SMART Attributes with Thresholds:							
ID#	ATTRIBUTE NAME	FLAGS	VALUE	WORST	THRESH	FAIL	RAW_VALUE
1	Raw_Read_Error_Rate	P0-R--	100	100	050	-	0
2	Throughput_Performance	P-S---	100	100	050	-	0
3	Spin_Up_Time	POS--K	100	100	001	-	1006
4	Start_Stop_Count	-0--CK	100	100	000	-	12292
5	Reallocated_Sector_Ct	P0--CK	100	100	050	-	12
7	Seek_Error_Rate	P0-R--	100	100	050	-	0
8	Seek_Time_Performance	P-S---	100	100	050	-	0
9	Power_On_Hours	-0--CK	077	077	000	-	9438
10	Spin_Retry_Count	P0--CK	253	100	030	-	0
12	Power_Cycle_Count	-0--CK	100	100	000	-	11172
192	Power-Off_Retract_Count	-0--CK	100	100	000	-	247
193	Load_Cycle_Count	-0--CK	080	080	000	-	209208
194	Temperature_Celsius	-0--K	100	100	000	-	36 (Min/Max 5/55)
196	Reallocated_Event_Count	-0--CK	100	100	000	-	12
197	Current_Pending_Sector	-0--CK	100	100	000	-	0
198	Offline_Uncorrectable	---CK	100	100	000	-	0
199	UDMA_CRC_Error_Count	-0--CK	200	200	000	-	0
220	Disk_Shift	-0----	100	100	000	-	48
222	Loaded_Hours	-0--CK	079	079	000	-	8471
223	Load_Retry_Count	-0--CK	100	100	000	-	0
224	Load_Friction	-0--K	100	100	000	-	0
226	Load-in_Time	-0S--K	100	100	000	-	322
240	Head_Flying_Hours	P-----	100	100	001	-	0
_ _ _ _ _ K auto-keep							
_ _ _ _ _ C event count							
_ _ _ _ _ R error rate							
_ _ _ _ _ S speed/performance							
_ _ _ _ _ O updated online							
_ _ _ _ _ P prefailure warning							

Posted by [Andrey.ponomarenko](#) on 19 April 2018 at 07:30 (GMT).

## List of devices with poor Linux-compatibility



A new project has been created to collect the list of computer hardware devices with poor Linux compatibility based on the [Linux-Hardware.org](https://github.com/linuxhw/HWInfo) data for 4 years: <https://github.com/linuxhw/HWInfo>

There are about 26 thousands of depersonalized hwinfo reports in the repository from computers in various configurations (different kernels, OS — mostly ROSA Fresh). The device is included into the list of poorly supported devices if there is at least one user probe in which the driver for this device was not found. The column 'Missed' indicates the percentage of such probes. If number of such probes is small, it means that the driver was already added in newer versions of the OS. In this case we show minimal version of the Linux kernel in which the driver was present.

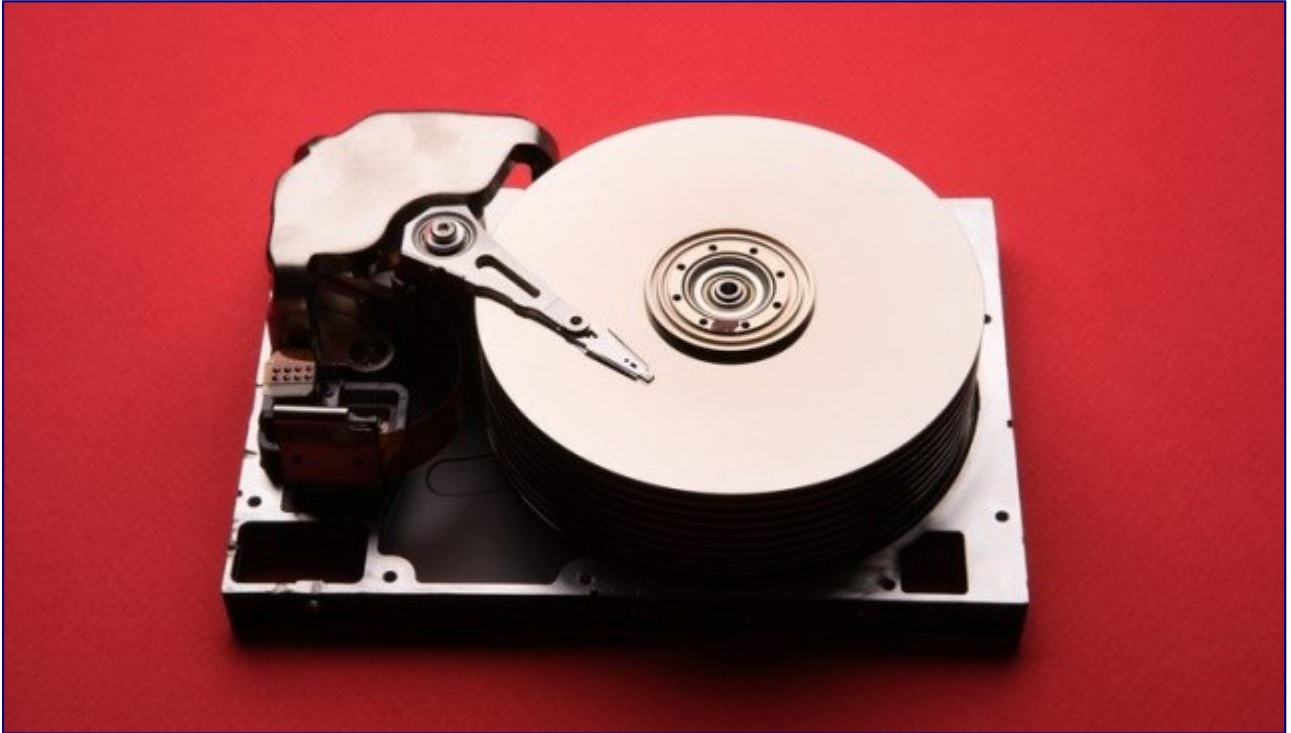
Devices are divided into categories. For each category we calculate the ratio of poorly supported devices to the total number of devices tested in this category.

At the moment, the study is limited only to PCI and USB devices. In the future, it is planned to include the rest.

Please check the presence of known unsupported devices in the table. The device ID can be taken from the output of the 'lspci -vvnn' command in square brackets, for example [1002:9851].

Posted by [Andrey.ponomarenko](#) on 15 March 2018 at 07:30 (GMT).

## Real-life reliability test for hard drives



A new open project has been created to estimate reliability of hard drives (HDD/SSD) in real-life conditions based on the SMART data collected in the [Linux-Hardware.org](https://linux-hardware.org) database. The initial data (SMART reports), analysis methods and results are publicly shared in a new github repository: <https://github.com/linuxhw/SMART>. Everyone can contribute to the report by uploading probes of their computers by the hw-probe tool!

The primary aim of the project is to find drives with longest "power on hours" and minimal number of errors. We use the following formula as a measure of reliability:  $\text{Power\_On\_Hours} / (1 + \text{Number\_Of\_Errors})$ , i.e. time to the first error/between errors.

Please be careful when reading the results table. Pay attention not only to the rating, but also to the number of checked model samples. If rating is low, then look at the number of power-on days and number of errors occurred. New drive models will appear at the end of the rating and will move to the top in the case of long error-free operation.

You can, as before, create a probe of your computer via the application in SimpleWelcome menu or from the console by a simple command:

```
hw-probe -all -upload
```

Posted by [Andrey.ponomarenko](#) on 15 February 2018 at 07:30 (GMT).