brainchip™
User Guide

23041 Avenida De La Carlota,
Suite 250, Laguna Hills,
CA 92653

+1 949 784 0040

AKD1000 and AKD1500 Driver

User Guide

version 1.0

© 2026 BrainChip Holdings Ltd. All rights reserved.

23041 Avenida De La Carlota,
Suite 250, Laguna Hills,
CA 92653

+1 949 784 0040

1 Introduction

The Akida Driver supports all AKD1000 and AKD1500 modules including PCIe and M.2 form factors. It will work out-of-the-box using Ubuntu 16.04, 18.04, and 20.04. The Driver supports many Ubuntu-based Linux distributions running on x86-64 and aarch64 architecture, although some kernel versions are not officially supported. In this user guide, we will use PCIe as an example. Terminal examples in this guide show the prompt **`bcdemo@mypi`**, where **`bcdemo`** is the Linux user account and **`mypi`** is the host name of the example machine. Your own prompt will show your username and host name instead.

© 2026 BrainChip Holdings Ltd. All rights reserved.

23041 Avenida De La Carlota,
Suite 250, Laguna Hills,
CA 92653

+1 949 784 0040

2 Installation steps

2.1 Plug in the board

After turning off the power supply of your host machine, you can plug in your Akida PCIe card on your system.

2.2 Check board presence

Open a terminal on your system and check if the PCIe card is plugged in correctly, entering

```
lspci | grep Co-processor
```

You should get this output

```
bcdemo@mypi:~$ lspci | grep Co-processor
01:00.0 Co-processor: Device 1e7c:bca1 (rev 01)
```

Or

```
bcdemo@mypi:~$ lspci | grep Co-processor
01:00.0 Co-processor: Brainchip Inc AKD1000 Neural Network Coprocessor [Akida] (rev 01)
```

2.3 Install the driver

Go to Brainchip GitHub: https://github.com/Brainchip-Inc/akida_dw_edma and follow the instructions given in the ***README.md*** file.

Or do the following:
Open a terminal and open a folder where you can put Akida PCIe driver
Enter

```
git clone https://github.com/Brainchip-Inc/akida_dw_edma
```

Then install needed libraries

```
sudo apt install build-essential linux-headers-$(uname -r)
```

And finally, get into the newly created folder and enter

```
sudo ./install.sh
```

You should get this output:

```
bcdemo@mypi:~$ mkdir WorkHere

bcdemo@mypi:~/WorkHere$ git clone https://github.com/Brainchip-Inc/akida_dw_edma
Cloning into 'akida_dw_edma'...
[…]
Resolving deltas: 100% (111/111), done.

bcdemo@mypi:~/WorkHere$ sudo apt install build-essential linux-headers-$(uname -r)
```


© 2026 BrainChip Holdings Ltd. All rights reserved.

23041 Avenida De La Carlota,
Suite 250, Laguna Hills,
CA 92653

+1 949 784 0040

```
[sudo] password for bcdemo:
Reading package lists... Done
[…]
1 upgraded, 1 newly installed, 0 to remove and 56 not upgraded.

bcdemo@mypi:~/WorkHere$ cd akida_dw_edma/

bcdemo@mypi:~/WorkHere/akida_dw_edma$ sudo ./install.sh
```

2.4 Check driver installation

Open a terminal on your system and check if Akida board driver is correctly installed, entering

`lsmod | grep akida`

You should get this output

```
bcdemo@mypi:~/WorkHere$ lsmod | grep akida
akida_pcie             49152  0
```

2.5 Install MetaTF

Please refer to https://doc.brainchipinc.com/installation.html for an entire Akida installation, but if you only want the Akida library to check if Akida 'sees' the board, you can simply enter

`pip install akida`

And then

`akida devices`

You should get this output

```
ubuntu@mypi:~/WorkHere$ akida devices
Available devices:
PCIe/NSoC_v2
```


© 2026 BrainChip Holdings Ltd. All rights reserved.

23041 Avenida De La Carlota,
Suite 250, Laguna Hills,
CA 92653

+1 949 784 0040

3 Notes and Troubleshooting

3.1 Previously installed driver

The steps given in section 1.3 will remove the old installed driver versions, if any, and install the new one. It will also configure modules to load it at every boot and provide read/write access on ***/dev/akida**** to every user.

3.2 Help needed

If you face any issues along the way, please contact us in the Support Page (https://developer.brainchip.com/support/) by filling the form.

3.3 Sudoers

If you get this error during installation

```
bcdemo is not in the sudoers file.  This incident will be reported.
```

It means that the user account you are currently logged in with (`bcdemo` in this example) does **not** have `sudo` (administrator) privileges.

The `sudoers` file is a system configuration file that defines which users are allowed to execute commands with elevated (root) privileges. Many installation and system configuration steps require these privileges.

To resolve this issue, add the user to the `sudo` group using an account that already has administrator privileges:

```
sudo usermod -aG sudo bcdemo
```

Alternatively, the equivalent command is:

```
sudo usermod -a -G sudo bcdemo
```

After running the command, log out and log back in (or reboot the system) for the group membership change to take effect. You can verify that the user has been added to the `sudo` group by running: groups `bcdemo`

The output should include the `sudo` group. The user will then be able to execute commands with administrator privileges using the `sudo` command.

In all of the commands above, `bcdemo` is the example username — replace it with your own, which is the name shown before the `@` in your terminal prompt.

© 2026 BrainChip Holdings Ltd. All rights reserved.