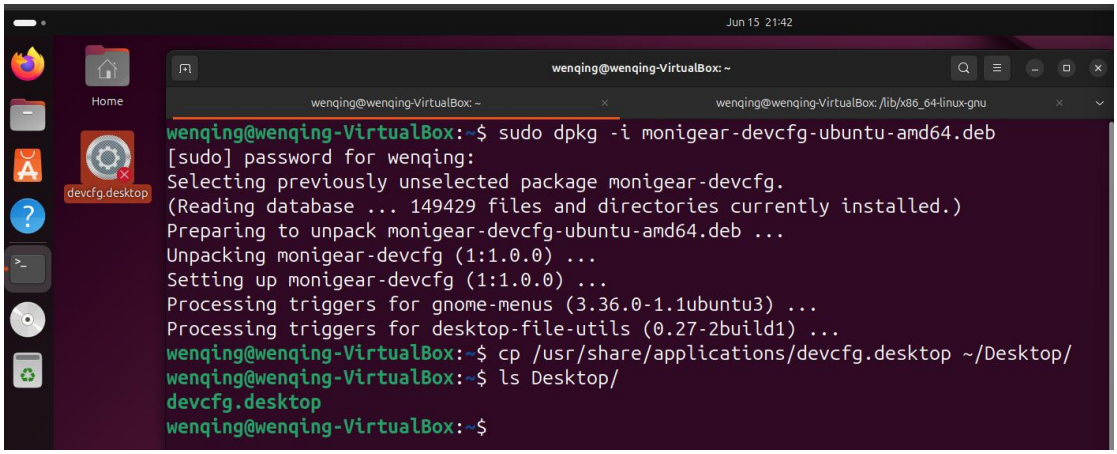
1) Installation process of Linux version device configuration tool

Take ubuntu24.04LTS as an example

A) Download the deb file of the installation package to the user's home directory

B) Execute: sudo dpkg -i monigear-devcfg-ubuntu-amd64.deb

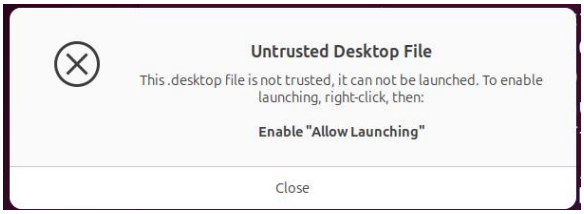
C) Normally, the installation is completed as shown in the figure below, and the command line runs /opt/monigear/devcfg/rundevcfg.sh to start the program.



D) If you want to add a desktop icon, execute

cp /usr/share/applications/devcfg.desktop ~/Desktop/

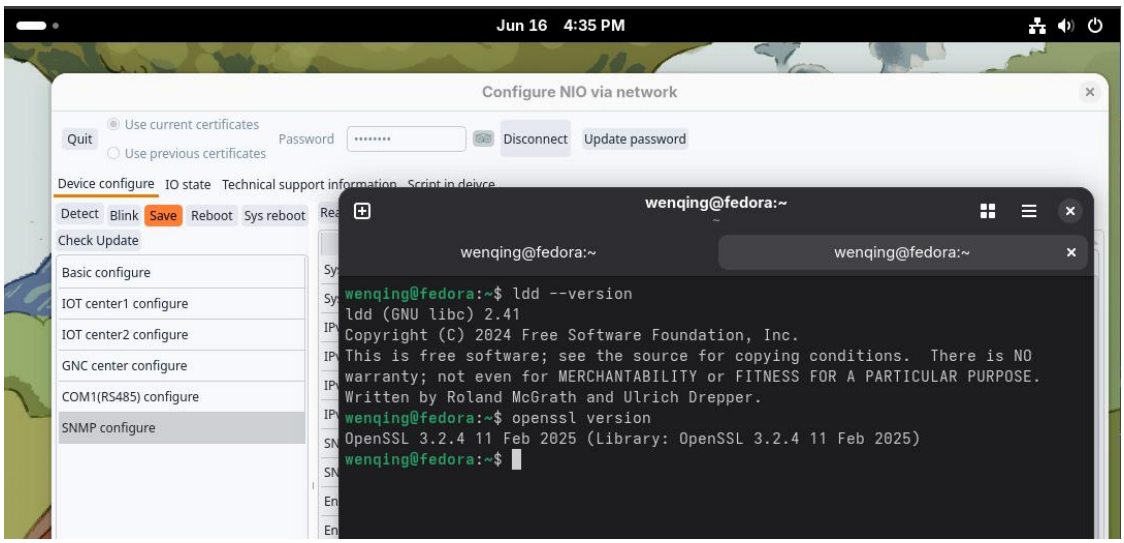
and then you can see the configuration tool icon appear on the desktop, as shown above. Double-click it and the following instructions will appear.



Just right-click on the icon and select "Allow launching".

The usage of the tool software is exactly the same as the Windows version, refer to the Windows version manual.

E) Firewall considerations: Due to different firewalls of different Linux distributions, such as ufw/iptable, to ensure that the program can communicate with the device, UDP port 6104 needs to be opened under the default settings.



Fedora running example

Take Centos Stream9 as an example to explain the installation of the rpm package

A) Download monigear-devcfg-1.0.0-2.x86_64.rpm

B) Run: rpm -ivh --nodeps monigear-devcfg-1.0.0-2.x86_64.rpm

If --nodeps is not used, it may prompt that the file dependency is missing, but it is not really missing. After adding this parameter, the installation can be used normally.

C) Open the udp port on the firewall and run the following two commands in the command line

```
sudo firewall-cmd --add-port=6104/udp --permanent
sudo firewall-cmd --reload
```

D) Run: /opt/monigear/devcfg/rundevcfg.sh to start the program.

2) Version differences

Since there are so many Linux distributions, it is difficult to adapt the program to all systems for smooth use. The following is the version available for download on our website, which is compiled and packaged for different systems.

Ubuntu (newer)	We compiled the packaged version on Ubuntu 22.04LTS and installed and verified it on the following systems: ● ubuntu 22.04LTS/24.04LTS(deb) ● Fedora42 workstation(rpm) These versions already use OpenSSL 3.0 or later by default, and the libc library version is 2.35 or later. If the distribution version you are using is similar to these versions, you can try to install it with this version.
Deepin (older)	We compiled the packaged version on Deepin and installed and verified it on the following systems: ● Deepin Desktop(deb) ● Kylin V10(deb) ● UOS Server V20(deb) ● Centos Stream9(rpm)

	Most of these versions use OpenSSL 1.1.1 by default. We have built-in openssl v3.0 dynamic library in the installation package. The libc library version is 2.28 and above. If the distribution version you are using is similar to this libc version, you can try to install it with this version.

In the future we may add compiled versions for different environments based on customer feedback.

3) Trouble shooting

Since there are so many Linux distributions, if you want to install this device management tool and encounter problems with your distribution, you can first report it to our technical support email address to obtain technical support. Another option is to try to solve it yourself. Here is a solution: Most of the running problems are caused by inconsistent dynamic library versions.

Running /opt/monigear/devcfg/rundevcfg.sh directly from the command line can display possible problems in the command line window, such as missing or library version conflicts.

Our tool software is developed by QT. The target path of the installation package is /opt/monigear/devcfg. All QT related libraries are located under this installation path. We do not copy any dynamic libraries to the system directory. The installation of our software should not affect your originally installed programs.

However, the system library may affect the operation of the library that our program depends on due to different versions. Sometimes this problem cannot be solved. For example, when I tried to install the Deepin version on ubuntu18.04LTS, it would prompt that libc version 2.28 was required while the system version was 2.27. There is no other solution to this situation. The only way to solve it is to compile and package it on such a platform. However, if libc can meet the requirements, but other libraries (non-QT libraries in the libs directory) conflict with the system libraries, you can try to move the conflicting libraries from /opt/monigear/devcfg/libs and install the corresponding version of the library provided by the system. This may solve some problem.