3.1 Commonly Used Tools

In this chapter, you need to connect to the same LAN as jetbot mini

1. PC(Host) side --- WinSCP

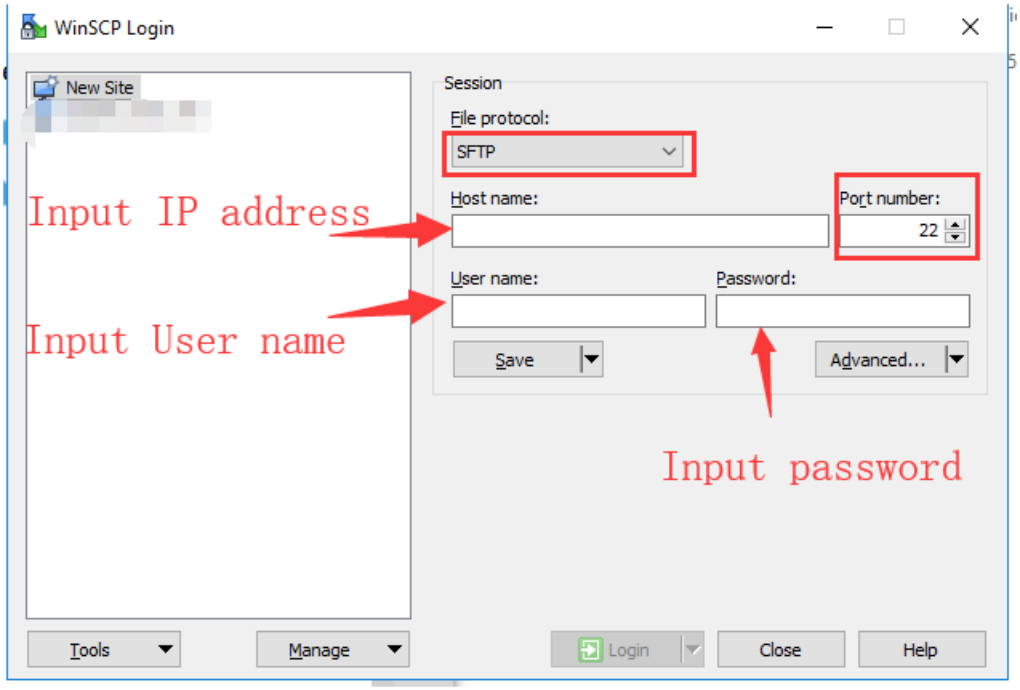
Install WinSCP:

Path of package : [JetBot-Mini-Robot-Car] --> [Annex] --> [Tools]---> [WinSCP-5.15.3.exe]

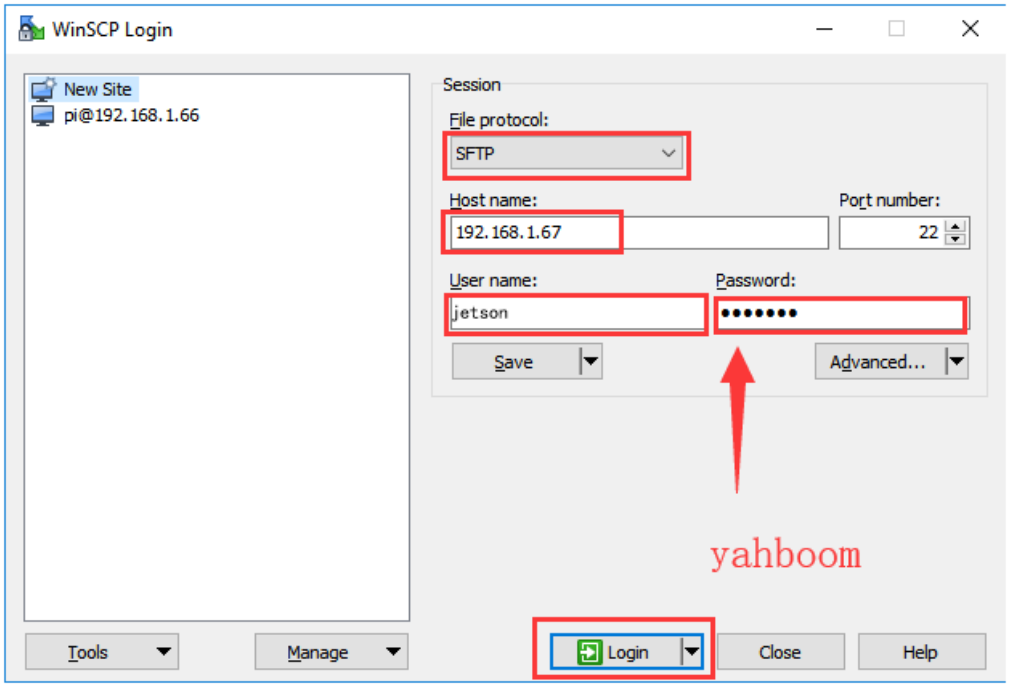
WinSCP is an open-source graphical SFTP client that uses SSH in a Windows environment and supports the SCP protocol. Its main function is to securely transfer files between local and remote computers.

We need to configure the host name IP address, port number, username and password, choose to remember the password and save the connection credentials.

As long as the next time you use, the IP address of the Jetbot Mini in the LAN does not change, we do not need to enter the IP address and password again.



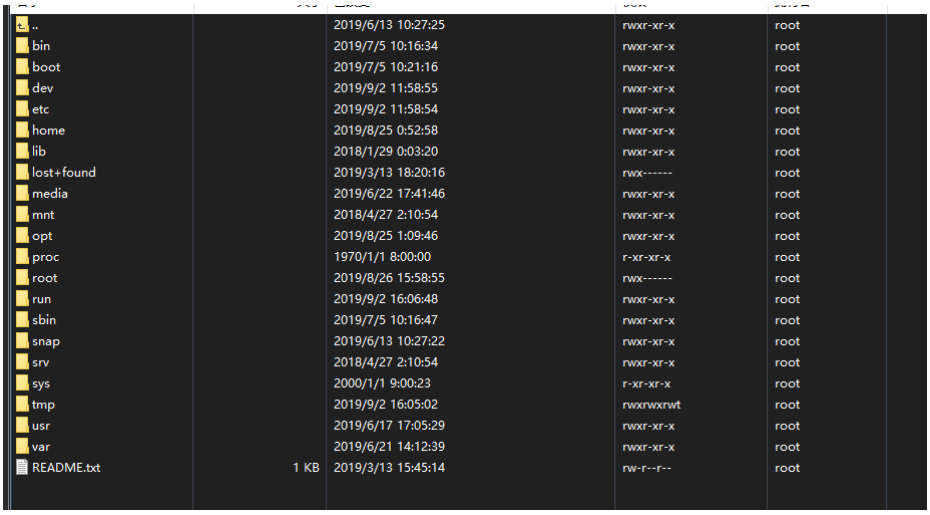
As shown below.

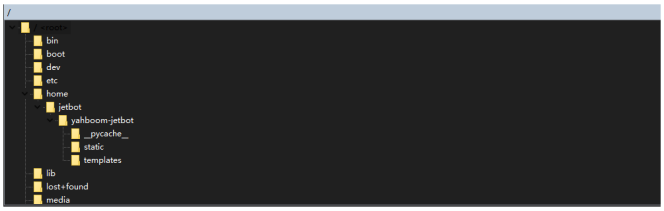


For example: my IP address is 192.168.1.67.

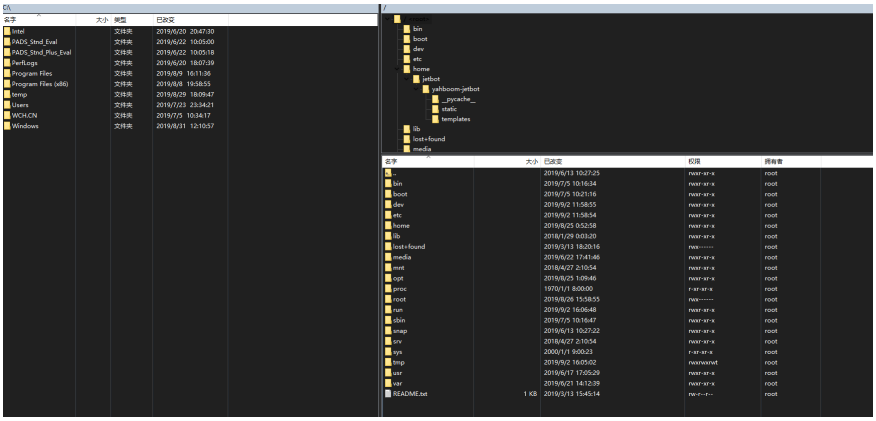
If you use image we provided, **the user name :jetson password: yahboom**

(The next course is to use this IP address and this image we provided.)





If you need to transfer any file between Windows and the remote Linux file system, just drag and drop to the target folder.



Note: Because WinSCP's encoding format and indentation rules may be inconsistent with your original file, resulting in an error after modification, so please do not modify the file directly here.

2. PC(Host) side --- Putty/Xshell

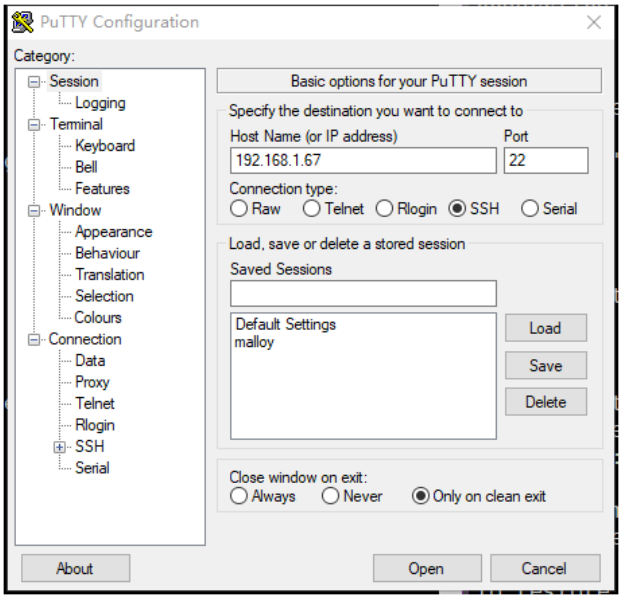
Install Putty **:**

Path of package : [JetBot-Mini-Robot-Car] --> [Annex] --> [Tools]---> [PuTTY.exe]

Install Xshell **:**

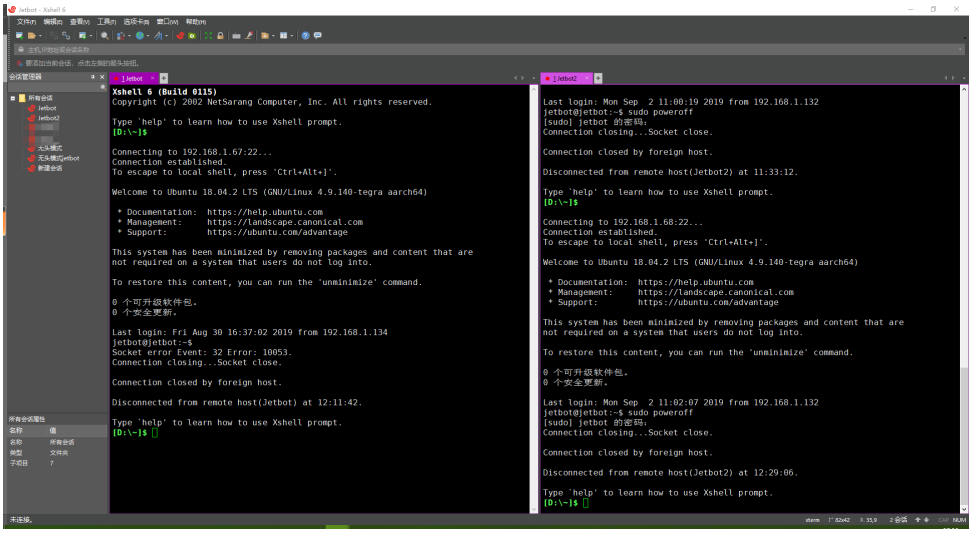
Path of package : [JetBot-Mini-Robot-Car] --> [Annex] --> [Tools]---> [xshell-6]

About Putty:

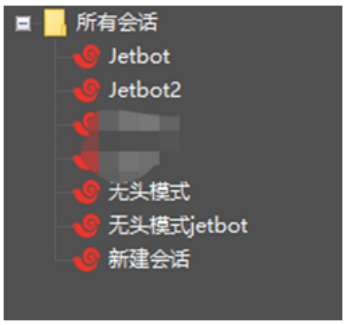


About Xshell:

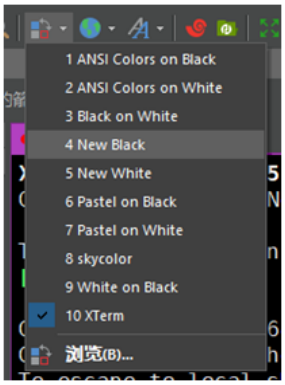
1) Multiple window operations can be performed by copying a conversation or opening a new connection while opening multiple consoles.



2) Ability to record multiple sets of login information, one-click connection.



3) Personalized color scheme.



3. PC(Host) side --- VNC

Install VNC **:**

Path of package : [JetBot-Mini-Robot-Car] --> [Annex] --> [Tools]---> [VNC-Viewer.exe]

If you use image we provided, **the user name :jetson password: yahboom**,You can directly skip to step 6 and log in to VNC according to the current IP address.

Note that the Software Updater software update is displayed after connecting to the desktop. In order to avoid compatibility problems, it is best not to update

1.Install vino

```
sudo apt update
sudo apt install vino
```

```
jetson@jetson-desktop:~$ sudo apt install vino
Reading package lists... Done
Building dependency tree
Reading state information... Done
vino is already the newest version (3.22.0-3ubuntu1.1).
vino set to manually installed.
The following packages were automatically installed and are no long
  apt-clone archdetect-deb bogl-bterm busybox-static cryptsetup-bi
  kwayland-data kwin-common kwin-data kwin-x11 libdebian-installer
  libkf5completion5 libkf5declarative-data libkf5declarative5 libk
  libkf5jobwidgets-data libkf5jobwidgets5 libkf5kcmutils-data libk
  libkf5package-data libkf5package5 libkf5plasma5 libkf5quickaddons
```

2.Set enable VNC service (at this time, you can manually open VNC server)

```
sudo ln -s ../vino-server.service  /usr/lib/systemd/user/graphical-
session.target.wants
```

```
jetson@jetson-desktop:~$ sudo ln -s ../vino-server.service   /usr/lib/systemd/user/graphical-session.target.wants
```

Configure VNC server:

```
gsettings set org.gnome.Vino prompt-enabled false
gsettings set org.gnome.Vino require-encryption false
```

```
jetson@jetson-desktop:~$ gsettings set org.gnome.Vino prompt-enabled false
jetson@jetson-desktop:~$ gsettings set org.gnome.Vino require-encryption false
```

Edit org.gnome, Recover the lost "enabled" parameter, enter the command to enter the file, and add the following key content to the back of the file. Save and exit.

```
sudo vi /usr/share/glib-2.0/schemas/org.gnome.Vino.gschema.xml
```

```
jetson@jetson-desktop:~$ sudo vi /usr/share/glib-2.0/schemas/org.gnome.Vino.gschema.xml
```

```
<key name='enabled' type='b'>
   <summary>Enable remote access to the desktop</summary>
   <description>
     If true, allows remote access to the desktop via the RFB
     protocol. Users on remote machines may then connect to the
     desktop using a VNC viewer.
   </description>
   <default>false</default>
</key>
```

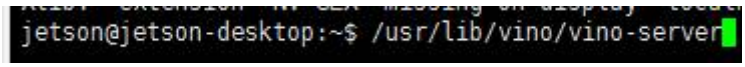
```
    <key name='disable-xdamage' type='b'>
      <summary>Whether we should disable the XDamage extension of X.org</summary>
      <description>
        If true, do not use the XDamage extension of X.org. This extension does
        not work properly on some video drivers when using 3D effects.
        Disabling it will make Vino work in these environments, with slower
        rendering as a side effect.
      </description>
      <default>false</default>
    </key>

    <key name='notify-on-connect' type='b'>
      <summary>Notify on connect</summary>
      <description>
        If true, show a notification when a user connects to the system.
      </description>
      <default>true</default>
    </key>
    <key name='enabled' type='b'>
      <summary>Enable remote access to the desktop</summary>
      <description>
         If true, allows remote access to the desktop via the RFB
         protocol. Users on remote machines may then connect to the
         desktop using a VNC viewer.
      </description>
      <default>false</default>
    </key>
  </schema>
</schemalist>
```

Set to Gnome compilation mode

```
sudo glib-compile-schemas /usr/share/glib-2.0/schemas
```

Now the screen sharing panel works in the unit control center... But that's not enough for vino to run! Therefore, you need to add a program: vino server when the session starts. Use the following command line:

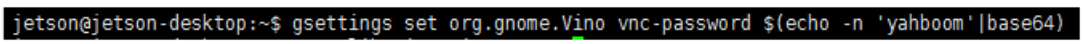
```
/usr/lib/vino/vino-server
```



This is a manual startup. If manual startup is required every time, it will be troublesome. The form of startup and self startup will be set below.

3.Set VNC login password ('the password 'is changed to your own password)

```
gsettings set org.gnome.Vino authentication-methods "['vnc']"
gsettings set org.gnome.Vino vnc-password $(echo -n 'thepassword'|base64)
```



4.Restart the machine and verify whether VNC is set successfully

```
sudo reboot
```

5.Set startup and self startup VNC server

VNC server is available only after you log in to Jetson locally. If you want VNC to be automatically available, use the system settings application to enable automatic login.

```
gsettings set org.gnome.Vino enabled true
mkdir -p ~/.config/autostart
vi ~/.config/autostart/vino-server.desktop
```

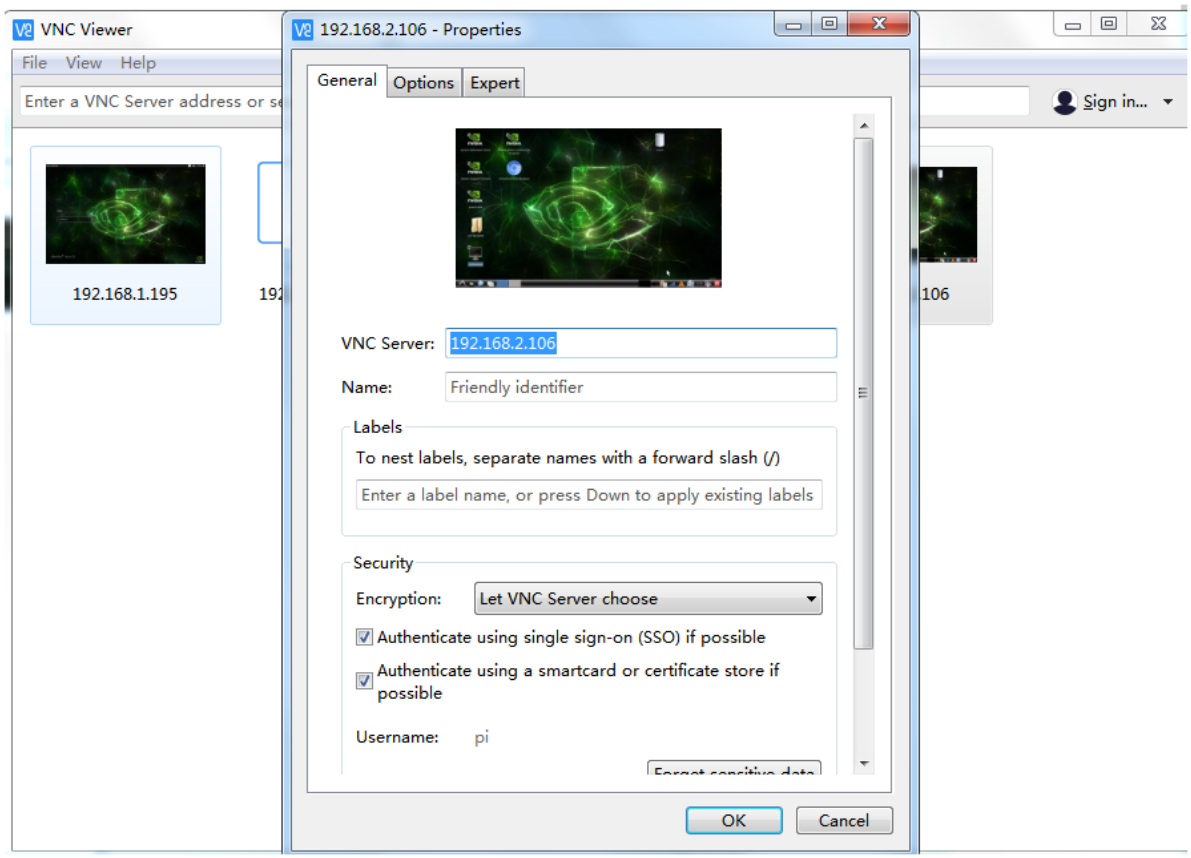
Add the following content to the file, save and exit.

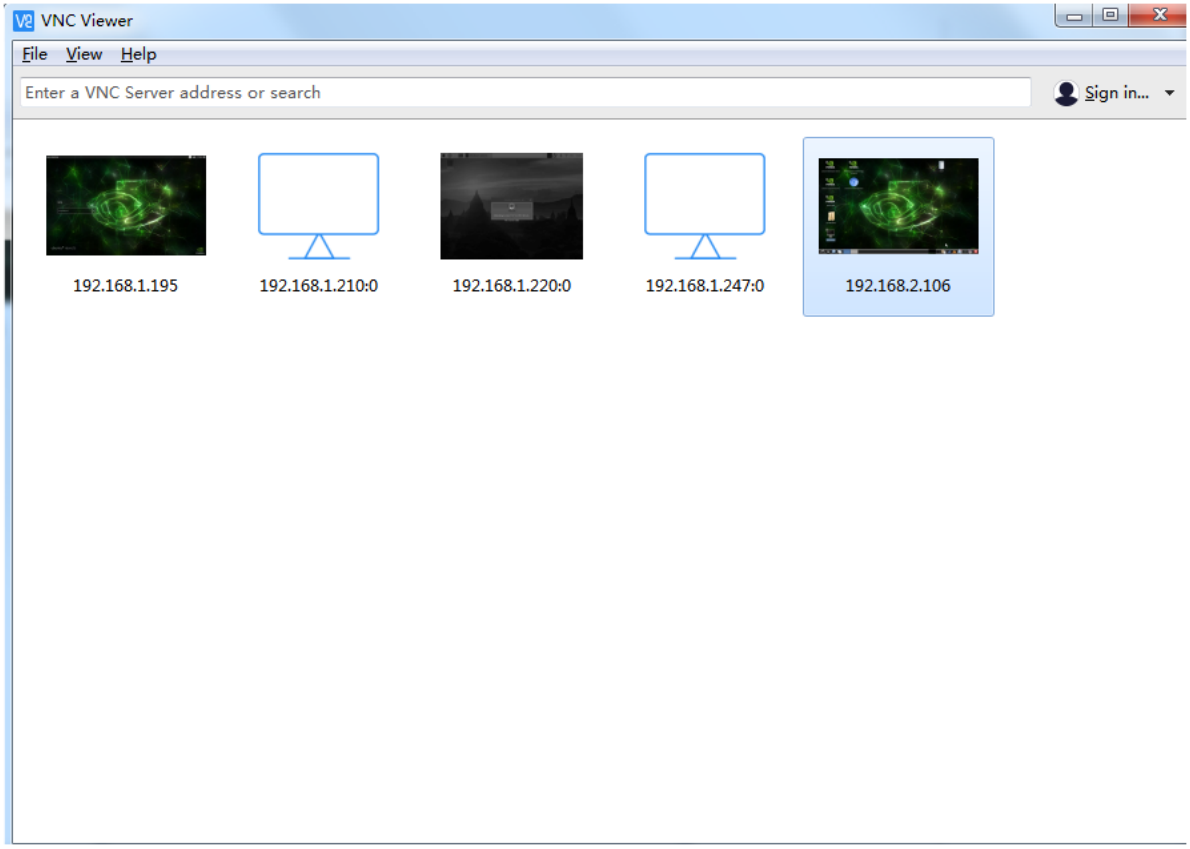
```
[Desktop Entry]
Type=Application
Name=Vino VNC server
Exec=/usr/lib/vino/vino-server
NoDisplay=true
```

If the system is set to enter the user password before entering the desktop, the above change script will not start until it enters the desktop. It is recommended to set the system to automatically log in to the desktop.

6.Connect to VNC server

For VNC connection using VNC viewer software, first you need to query the IP address. What I found here is 192.168 2.106. After entering the IP address, click OK, double-click the corresponding VNC user, enter the password, and finally enter the VNC interface.





4. Jetbotmini side - jtop

Input this command to install:

```
sudo pip3 install jetson-stats
```

Input this command to update:

```
sudo -H pip install -U jetson-stats
```

Input this command to run:

```
sudo jtop
```

Input this command to uninstall:

```
sudo pip3 uninstall jetson-stats
```

About Jtop:

Jtop is a system monitoring utility that can run on the terminal to view and control the status of NVIDIA Jetson products in real time.

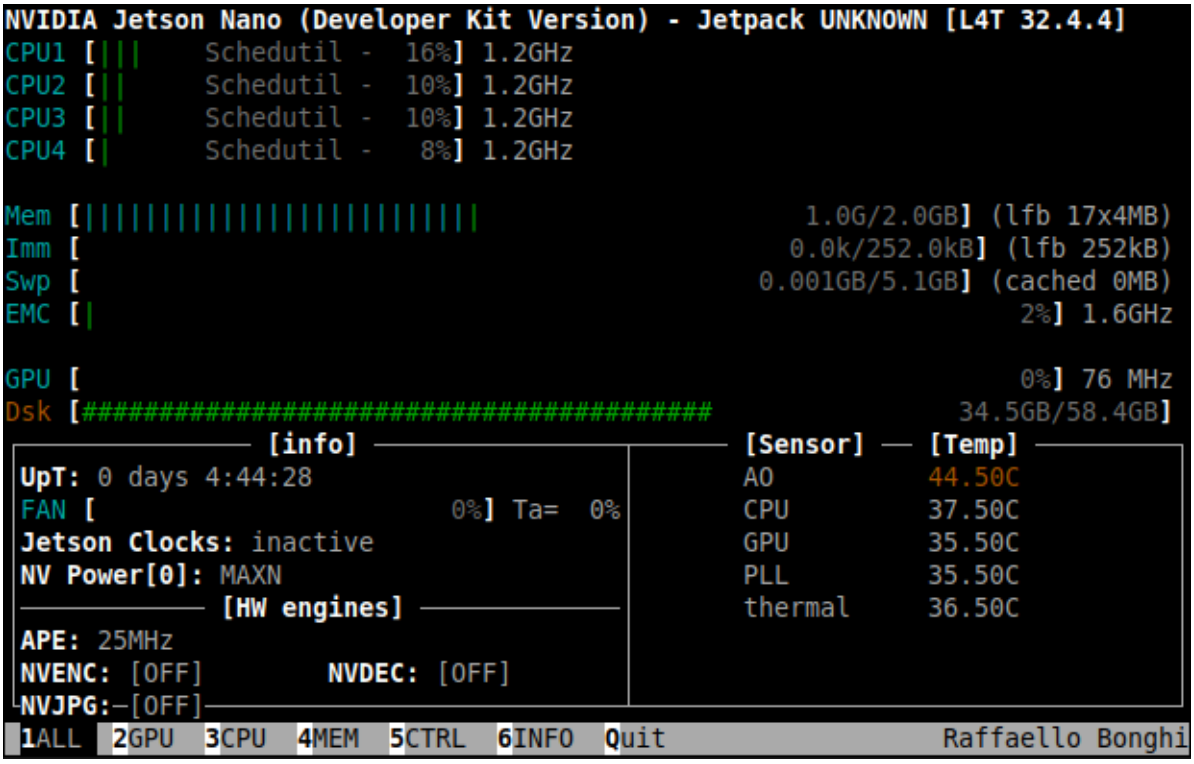
Control Jtop page:

Use the left and right arrow keys on your keyboard to switch between different status display interfaces.

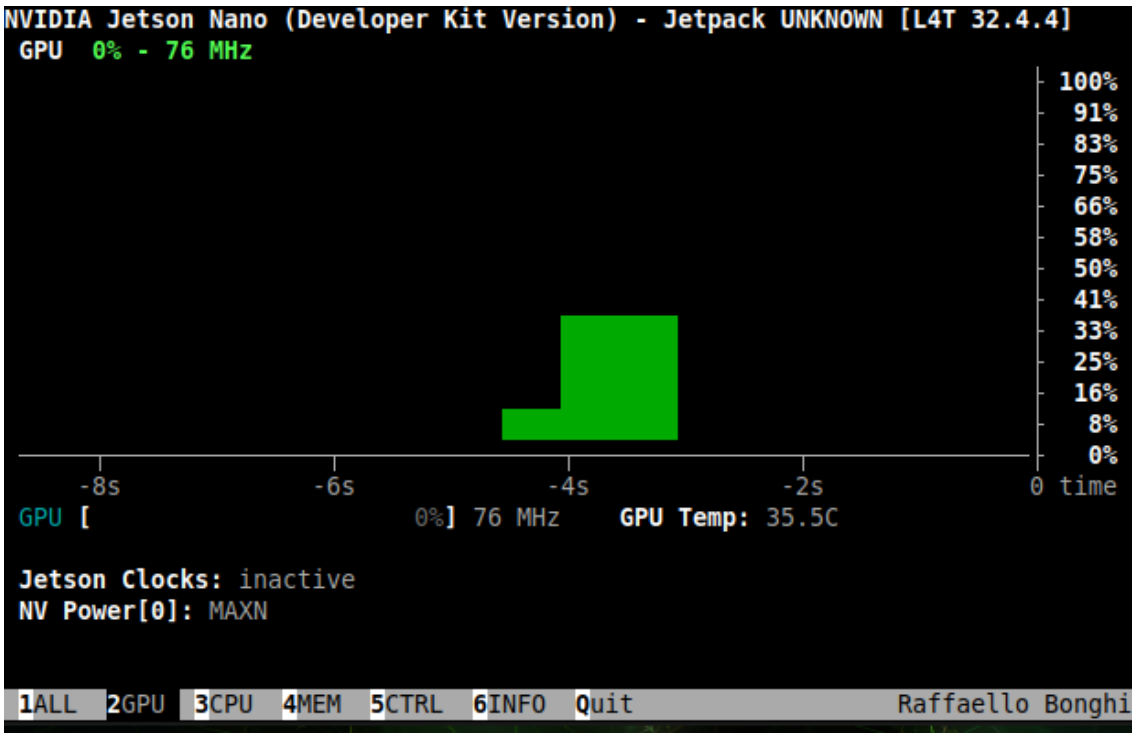
Jtop has four different pages to monitor your Jetbot's running status:

About four interfaces of Jtop:

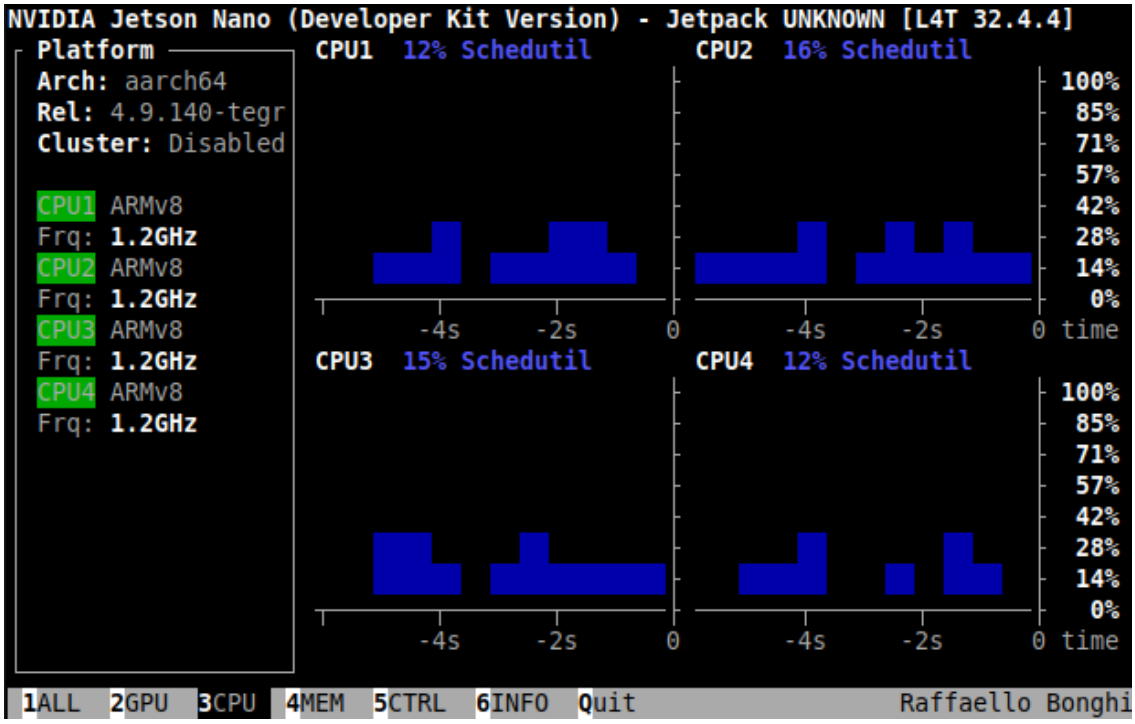
1. **ALL**: Collect all the information about development board: CPU, Memory, GPU, Disk, Fan and all the status about jetson_clocks, NVPmodel and others status.As shown below.



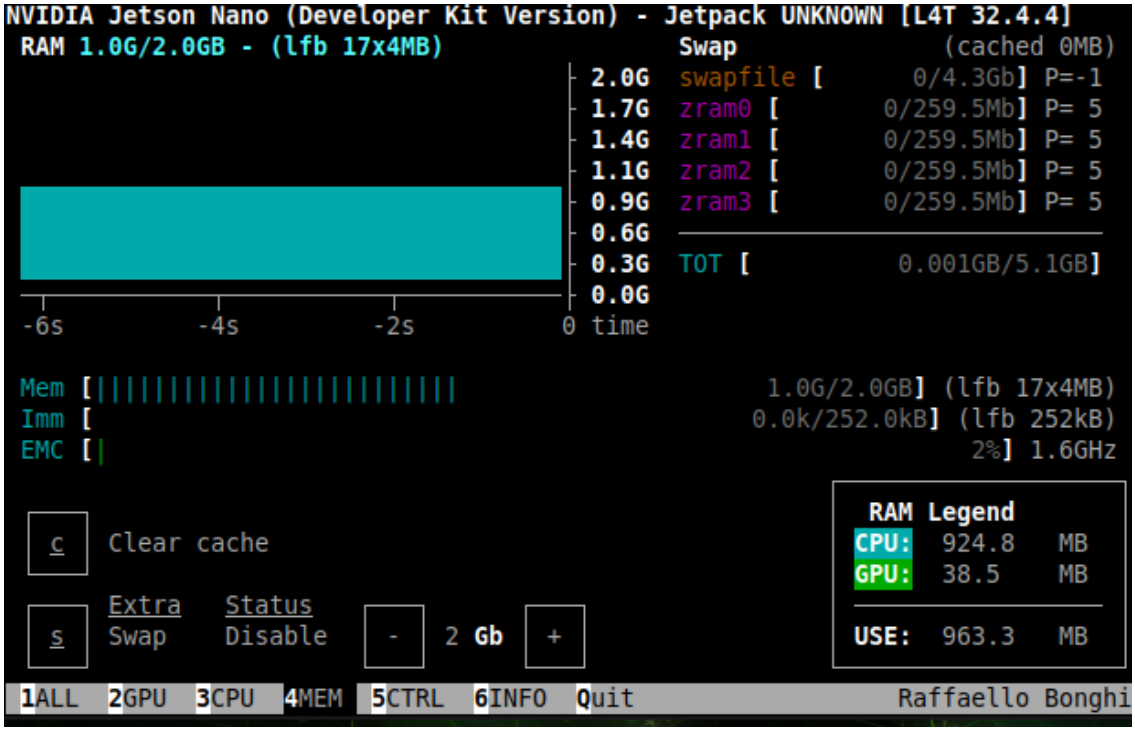
2. **GPU:** About your NVIDIA Jetson's real-time GPU history.



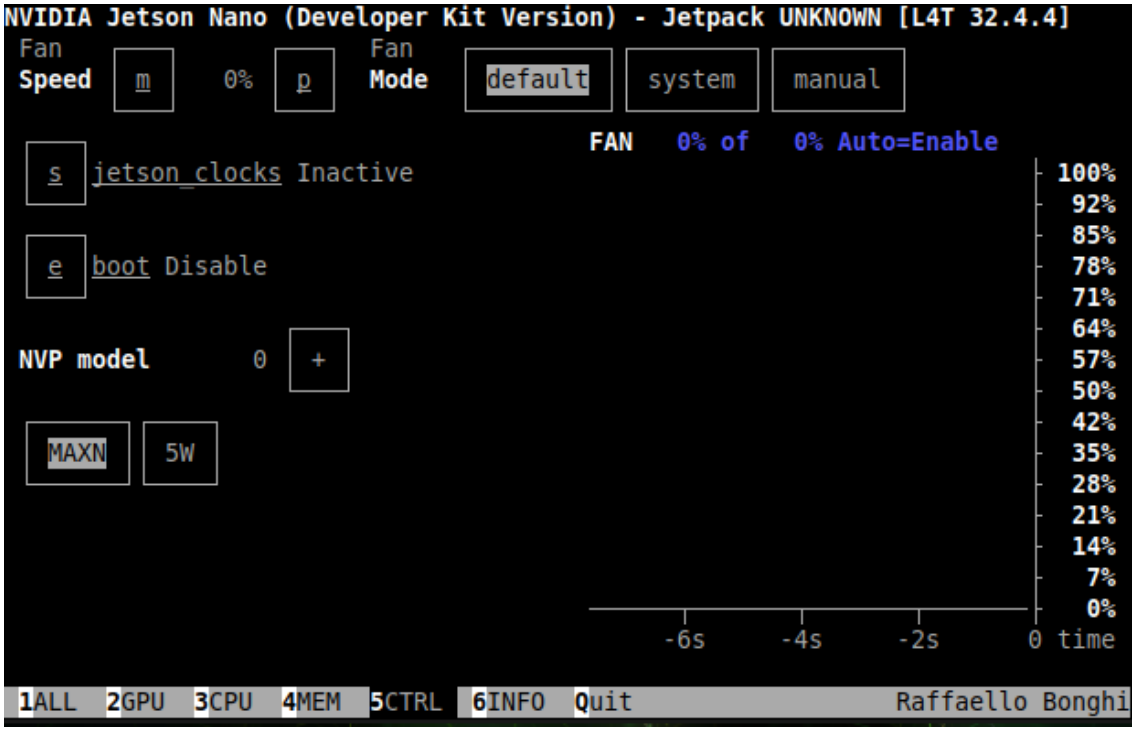
3. **CPU:** Display the dominant frequency and occupancy of 4-core CPU



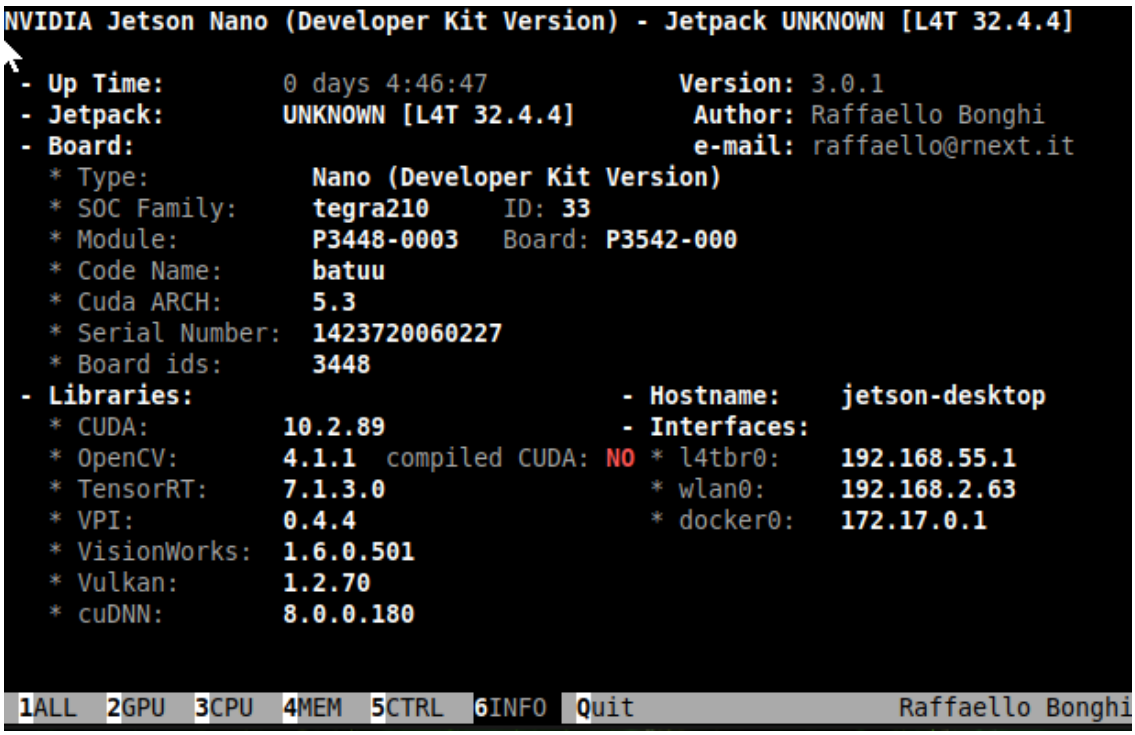
4. **MEM:** Memory status



5. **CTRL:** Control your status



6. **INFO:** Collect all information about library, CUDA, serial number, interface, etc.



About Control Jetbot:

We can use following key on the keyboard to control Jebot Mini.

a: Start/Stop jetson_clocks service（Note: jetson_clocks only starts after 60 seconds）

e: Start/Disable jetson_clocks onboard boot

+/-: Increase and decrease the NVP model

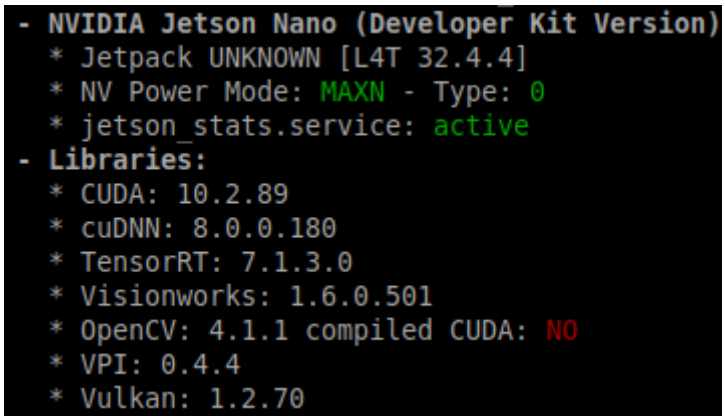
p/m: Increase and decrease speed of fan（But because Jetbot Mini uses a non-shiftable fan, this feature is negligible）

Others function:

Jetson_release:

Display information about the current version of Jetbot Mini environment component version, status, etc.

As shown below:



export | grep JETSON:

This script generates a simple environment variable to understand the hardware version of Jetson and the installed Jetpack.

As shown below:

```
jetson@jetson-desktop:~$ export | grep JETSON
declare -x JETSON_BOARD="P3542-000"
declare -x JETSON_BOARDIDS="3448"
declare -x JETSON_CHIP_ID="33"
declare -x JETSON_CODENAME="batuu"
declare -x JETSON_CUDA="10.2.89"
declare -x JETSON_CUDA_ARCH_BIN="5.3"
declare -x JETSON_CUDNN="8.0.0.180"
declare -x JETSON_JETPACK="UNKNOWN"
declare -x JETSON_L4T="32.4.4"
declare -x JETSON_L4T_RELEASE="32"
declare -x JETSON_L4T_REVISION="4.4"
declare -x JETSON_MACHINE="NVIDIA Jetson Nano (Developer Kit Version)"
declare -x JETSON_MODULE="P3448-0003"
declare -x JETSON_OPENCV="4.1.1"
declare -x JETSON_OPENCV_CUDA="NO"
declare -x JETSON_SERIAL_NUMBER="1423720060227"
declare -x JETSON_SOC="tegra210"
declare -x JETSON_TENSORRT="7.1.3.0"
declare -x JETSON_TYPE="Nano (Developer Kit Version)"
declare -x JETSON_VISIONWORKS="1.6.0.501"
```