

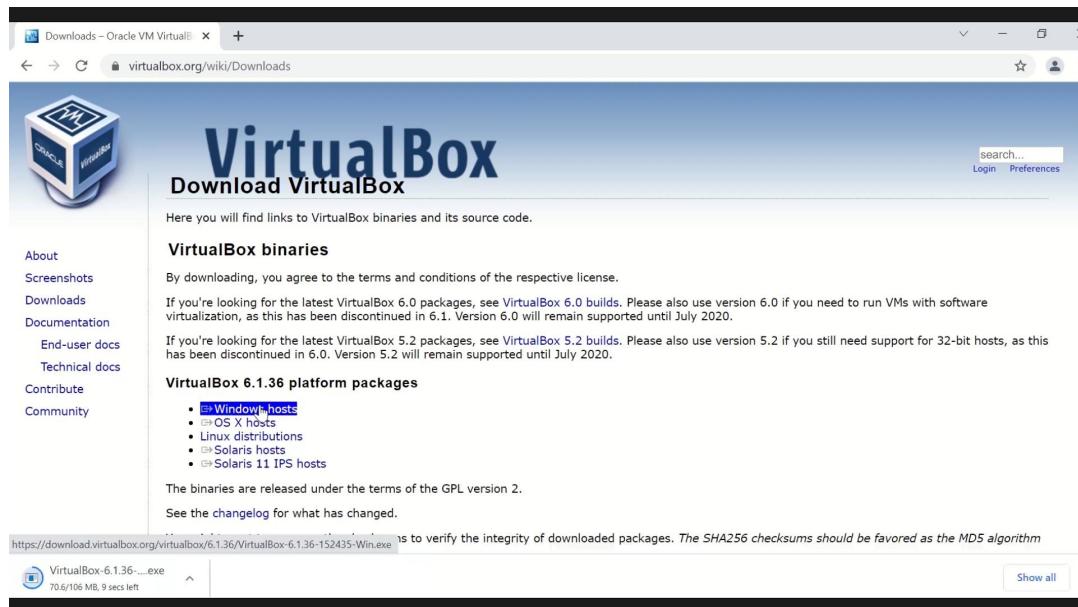
Creality Sonic Pad

User-defined model firmware compilation guide

- 1、Download and install VirtualBox
- 2、Download Ubuntu
- 3、Create a new virtual machine and install Ubuntu system
- 4、System upgrade and install git
- 5、Download Klipper firmware
- 6、Configure printer firmware
- 7、Upgrade printer firmware via SD/TF card
- 8、Upgrade printer firmware via USB

1. Download and install VirtualBox

Download VirtualBox from www.VirtualBox.org

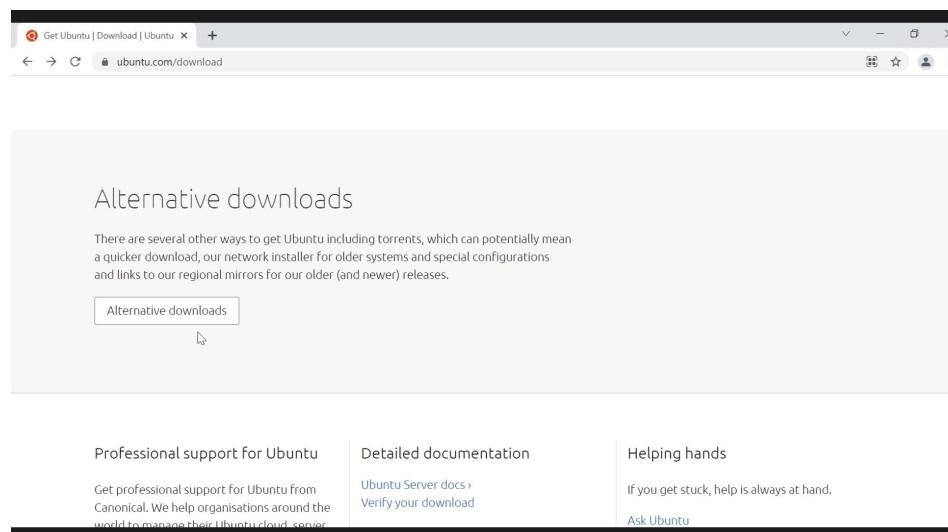
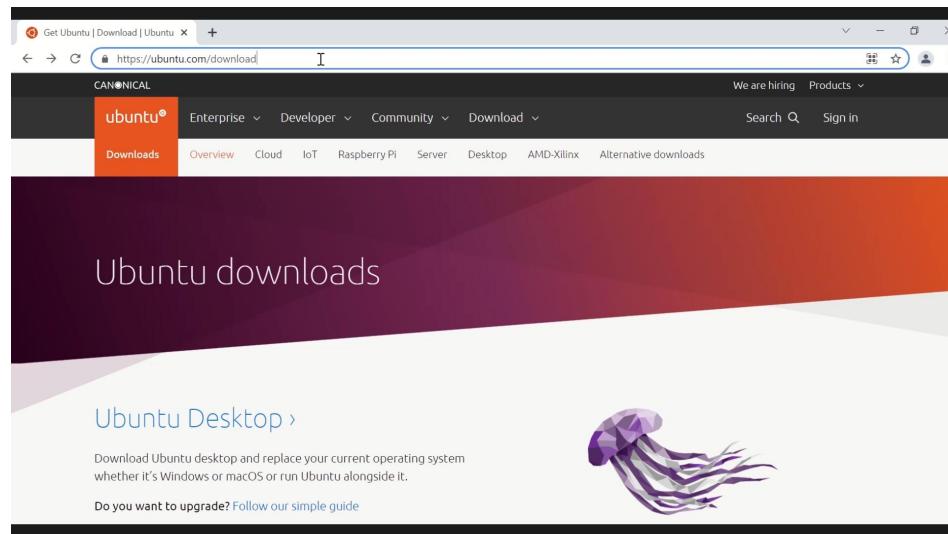


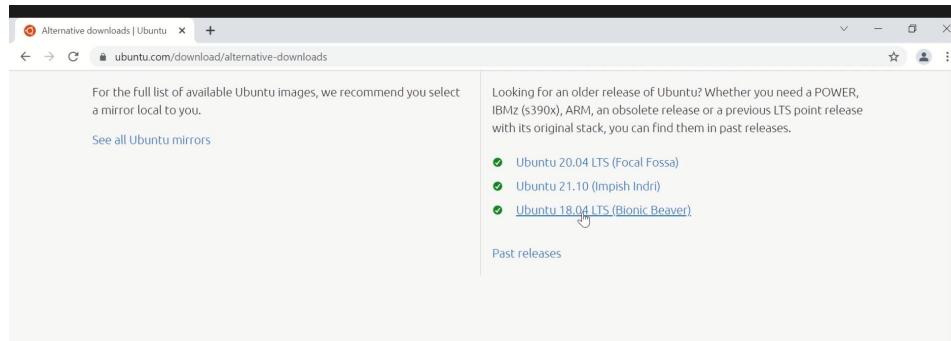
Download and install VirtualBox (Do not open it yet after installation)



2. Download Ubuntu

Download Ubuntu 18.04.LTS from www.ubuntu.com/download





Enterprise ready

Canonical delivers the leading Kubernetes distribution.

[Charmed Kubernetes >](#)

<https://releases.ubuntu.com/18.04/> Stack is the fastest way to build

Installation guides

If you need some help installing Ubuntu, please check out our step-by-step guides.

[Installing Ubuntu Desktop](#)
[Installing Ubuntu Server](#)
[Install On Focal](#)

Helping hands

If you get stuck, help is always at hand.
[Ask Ubuntu](#)
[Ubuntu Forums](#)
[Launchpad Answers](#)

A screenshot of a web browser window titled "Ubuntu 18.04 LTS (Bionic Beaver)". The URL in the address bar is "releases.ubuntu.com/18.04.6/?_ga=2.19672063.698931284.1659757208-699225595.1659408051". The page title is "Select an image".

Ubuntu is distributed on three types of images described below.

Desktop image The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MB of RAM to install from this image.	64-bit PC (AMD64) desktop image Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.
Server install image The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.	64-bit PC (AMD64) server install image Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

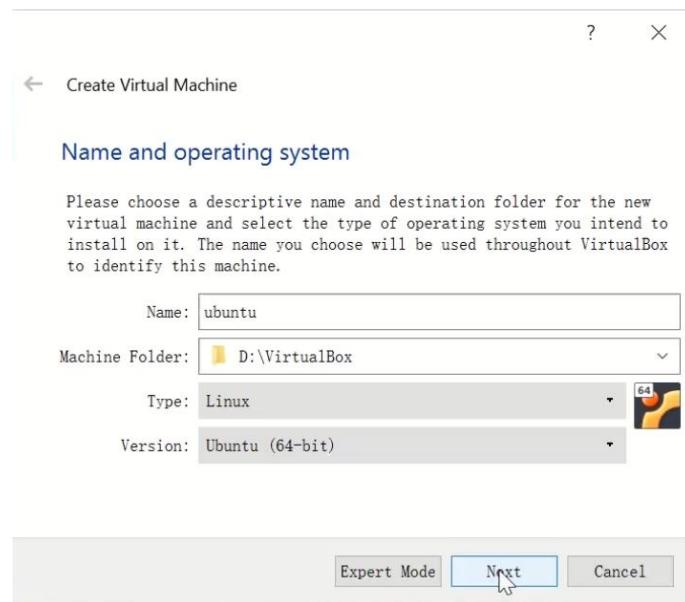
3、Create a new virtual machine and install Ubuntu system

Open VirtualBox and create a new virtual machine

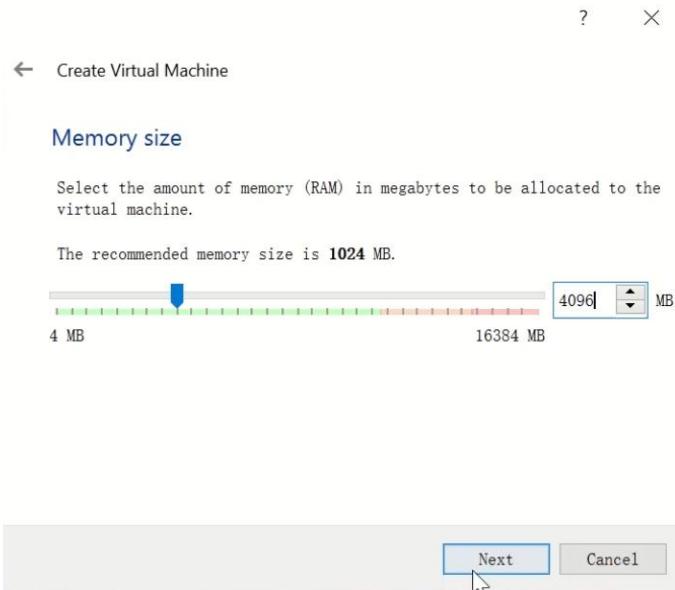


Select destination folder、Enter a name,

Type: Select Linux、Version: Select Ubuntu (64-bit)



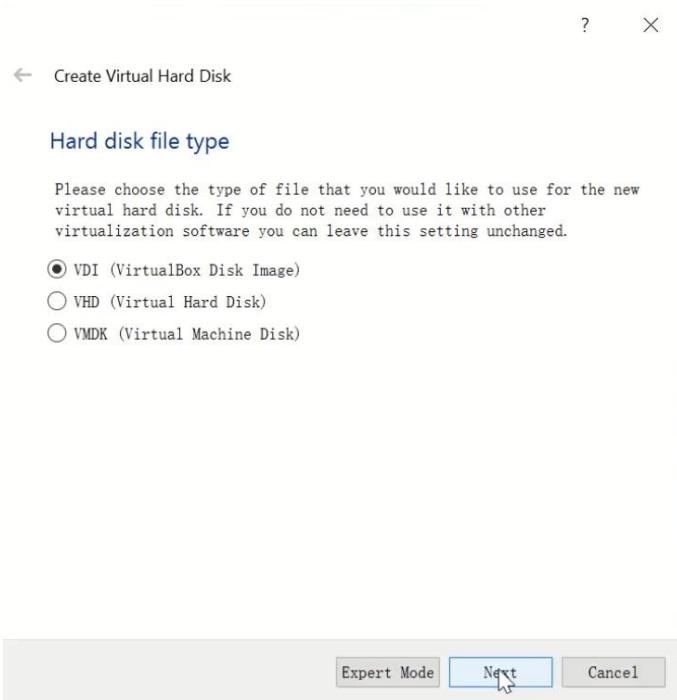
Resize memory to 4096MB



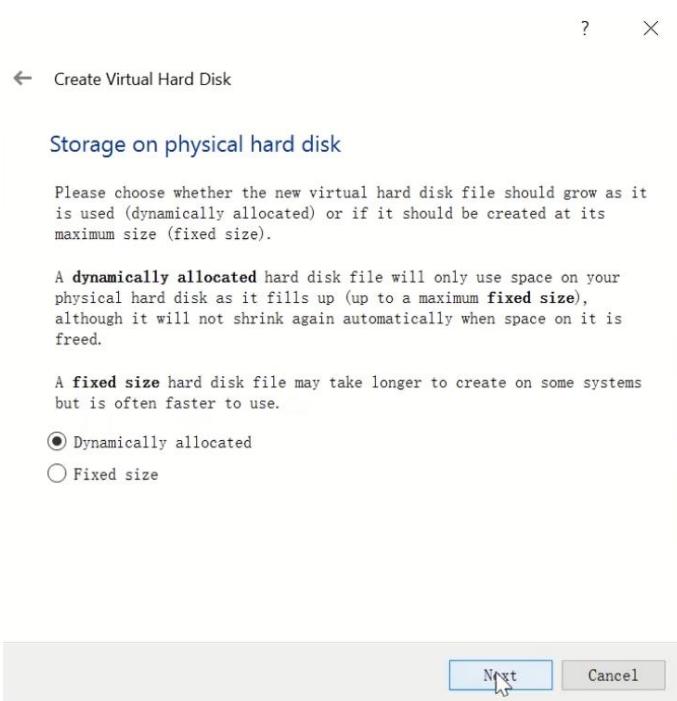
Create a virtual hard disk



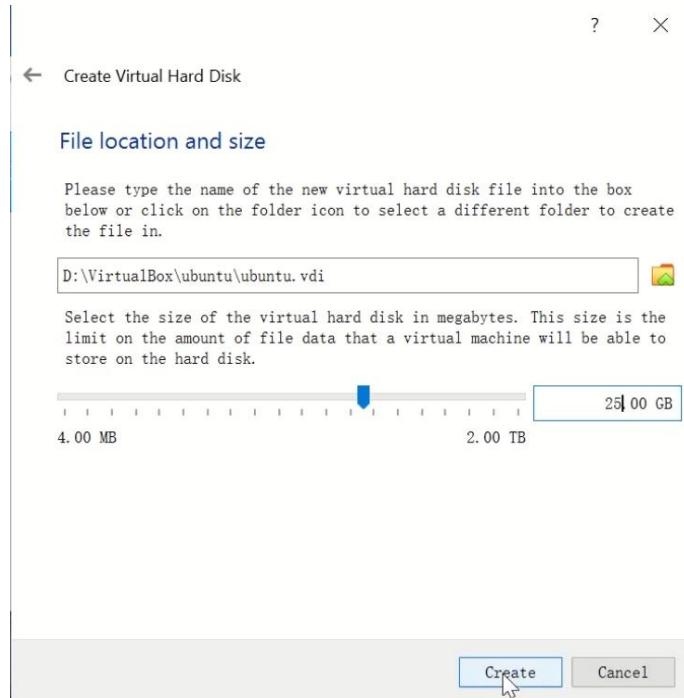
Select VDI



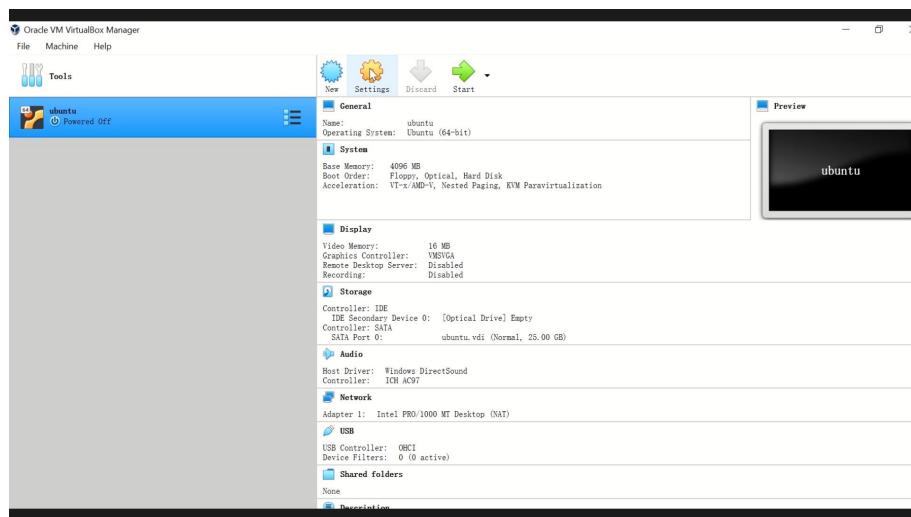
Select dynamic allocation



Resize the virtual hard disk to 25GB

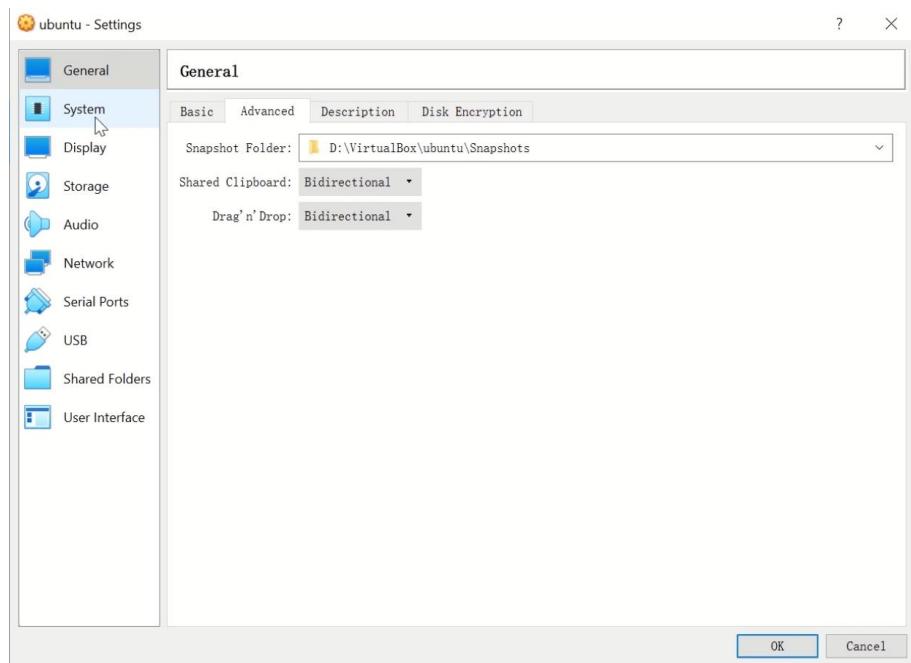


Open the settings

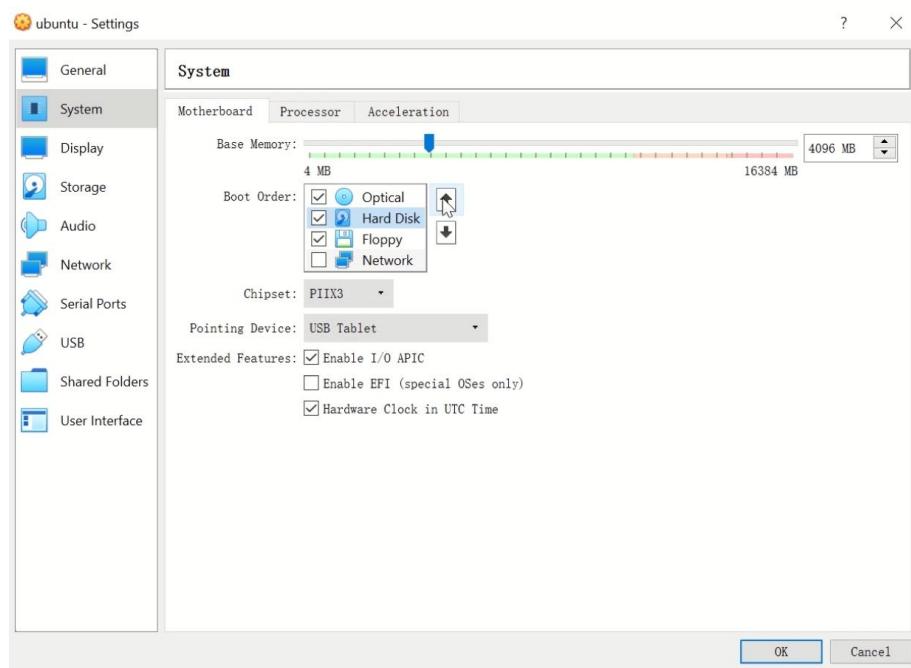


General Settings – Advanced,

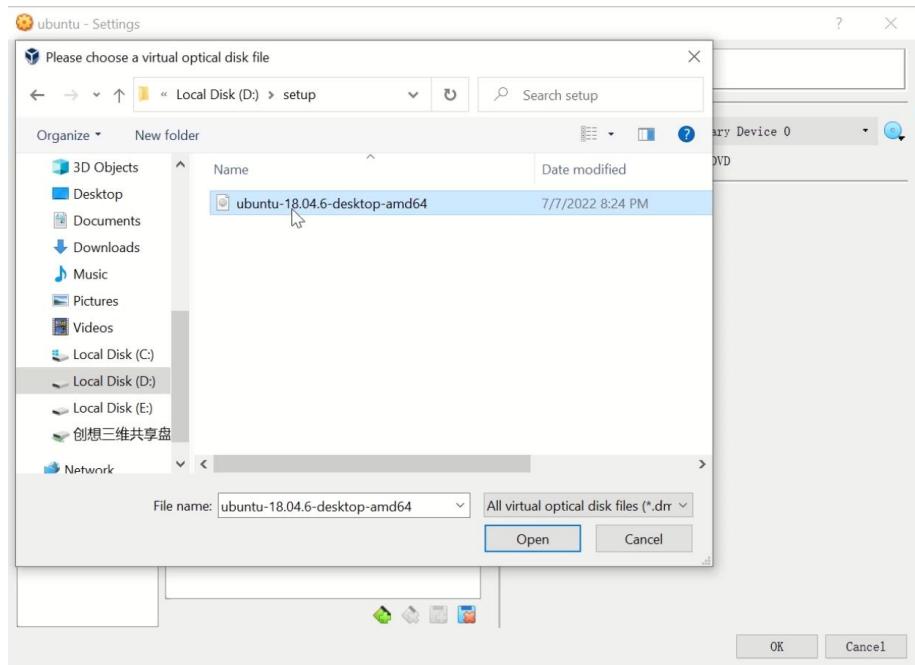
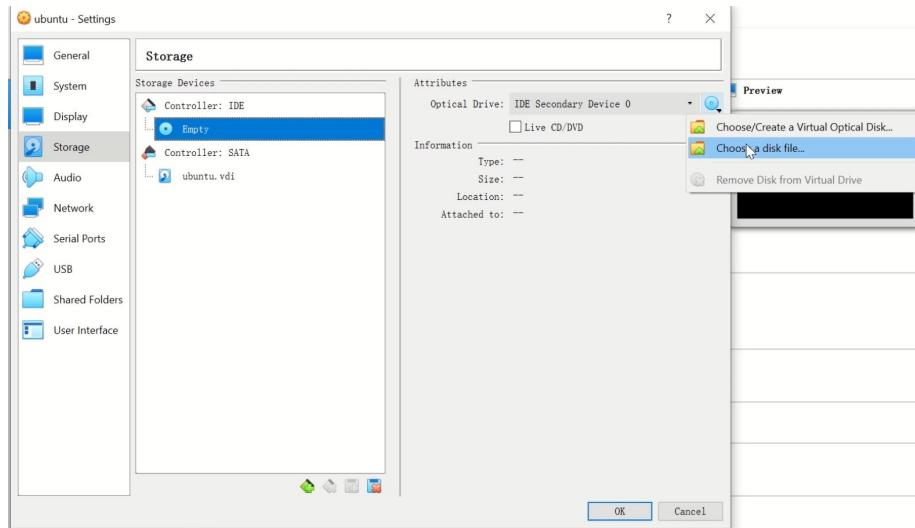
Change the shared clipboard and the drag and drop to bidirectional



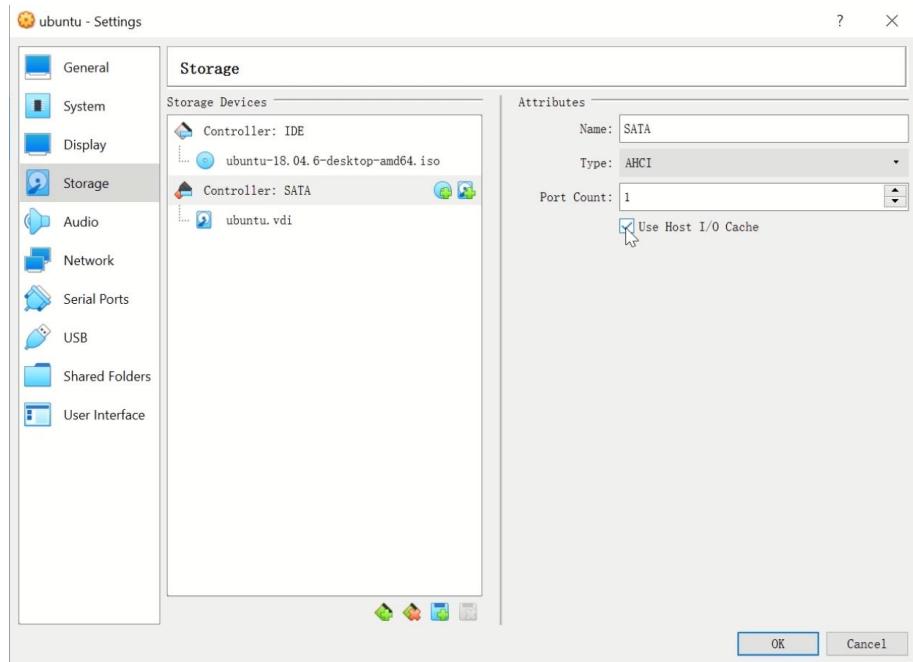
System settings - adjust the boot order to optical drive - hard disk - floppy drive



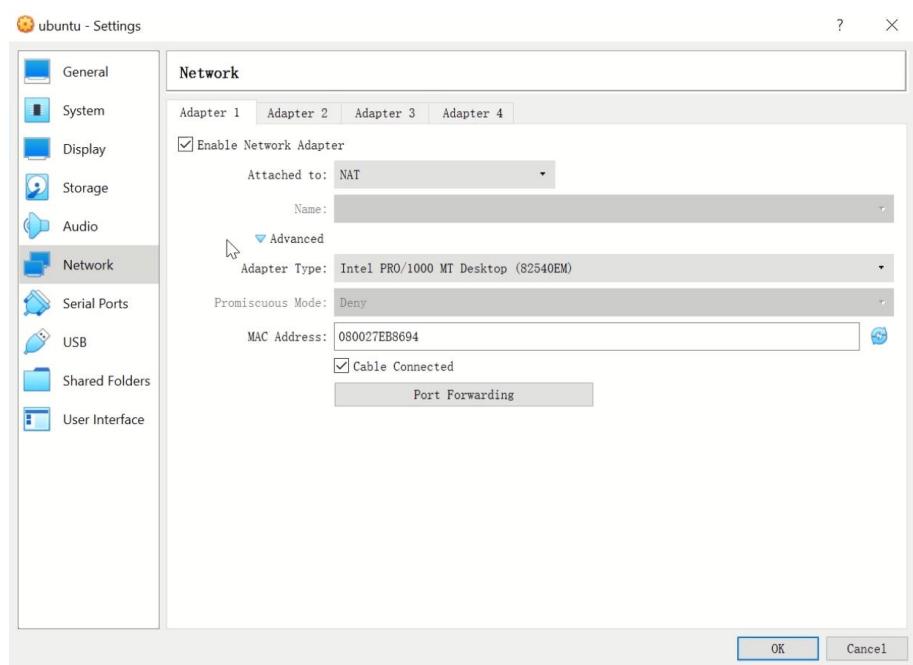
Storage Settings--IDE--Select Downloaded Ubuntu 18.04.6LTS



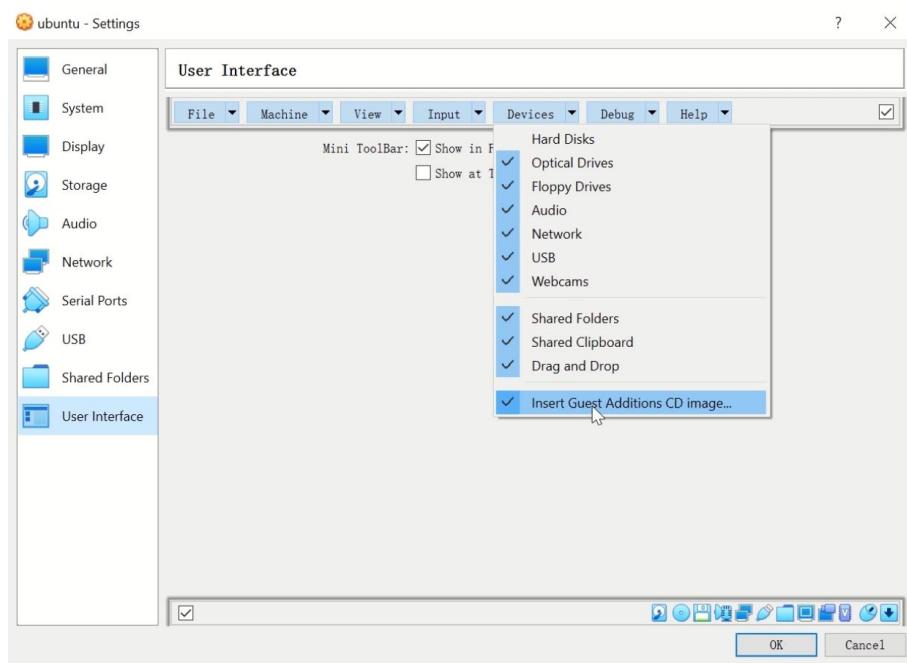
SATA--tick Use Host I/O Cache



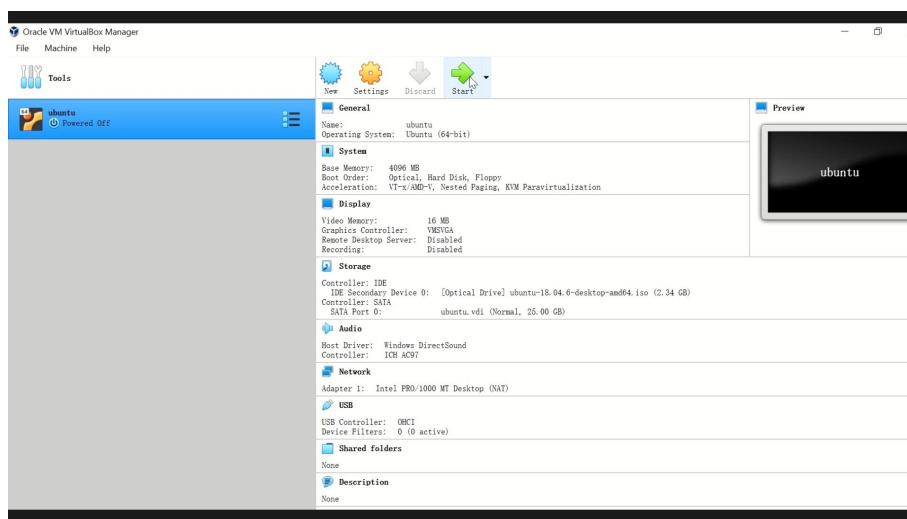
Network Settings - Advanced - Make sure the network is connected



User Interface -Device-Tick Insert Guest additions CD image



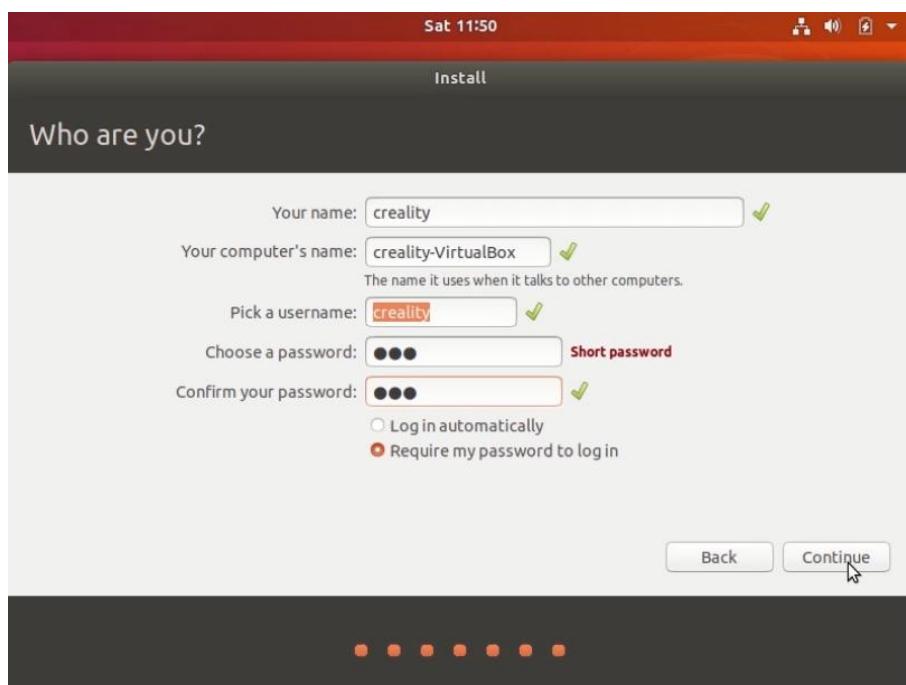
Start the virtual machine after confirmation



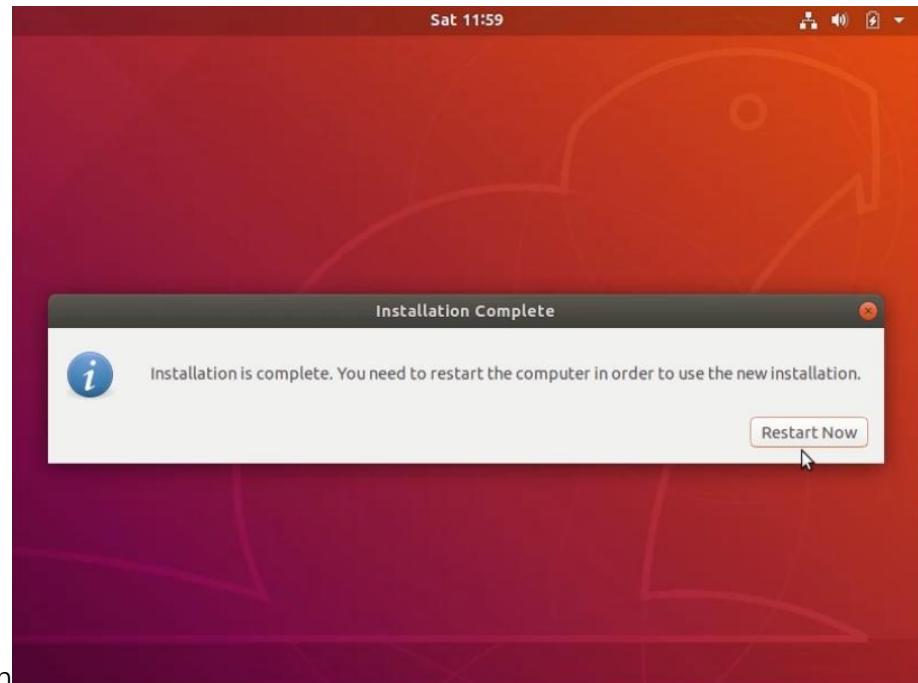
Install Ubuntu



Add username and system password



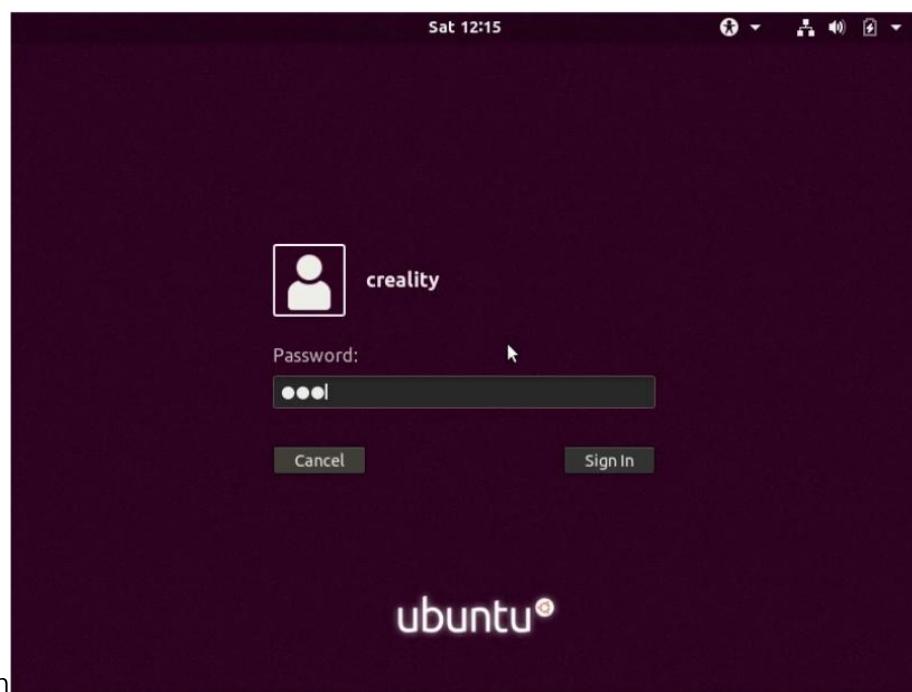
Wait for the restart after



confirmation

4. System upgrade and install git

Enter the system password you just set to enter the



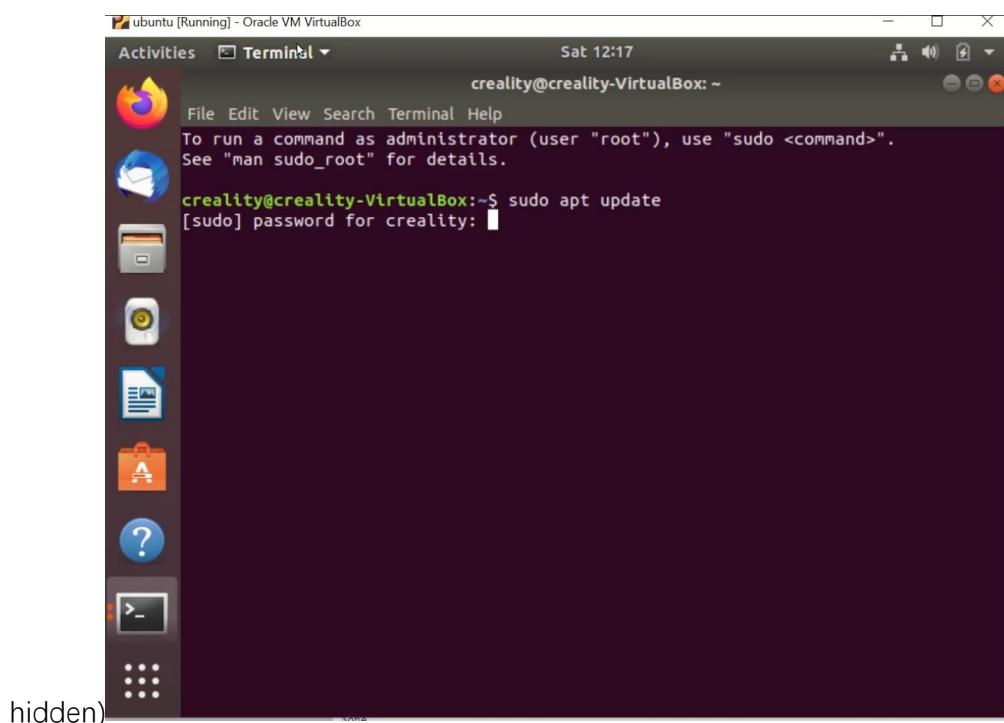
system

Desktop - right click - open

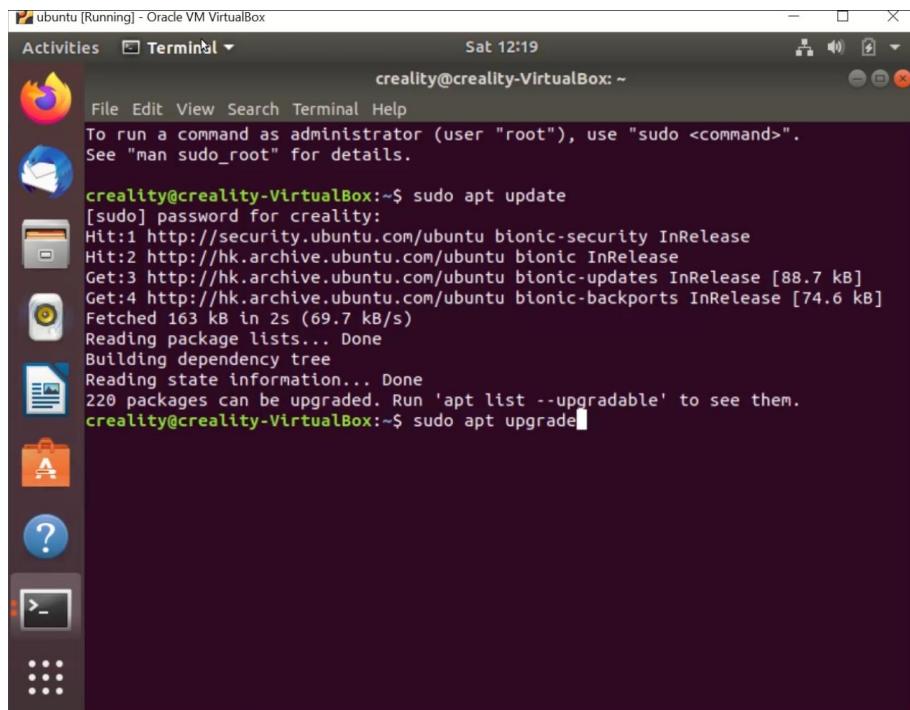


Enter "sudo apt update" and press Enter (list all updatable software)

Enter the system password and press Enter (the password is automatically



Enter "sudo apt upgrade" and press Enter (upgrade the package)



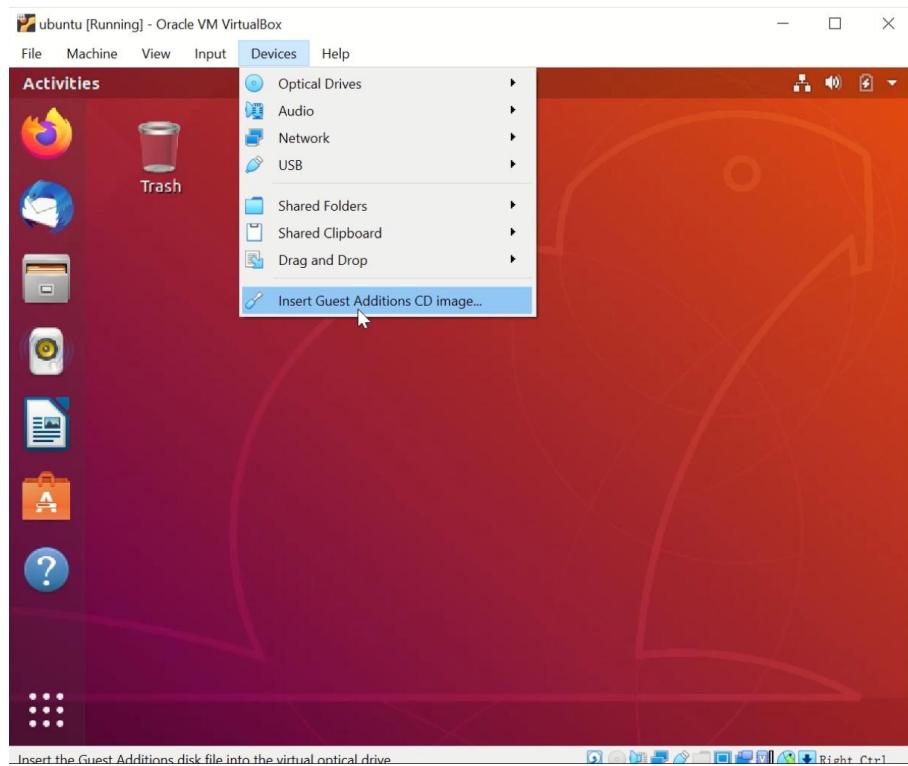
A screenshot of a terminal window titled "ubuntu [Running] - Oracle VM VirtualBox". The window shows a terminal session with the following text:

```
creality@creality-VirtualBox:~$ sudo apt update
[sudo] password for creality:
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://hk.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://hk.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://hk.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 163 kB in 2s (69.7 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
220 packages can be upgraded. Run 'apt list --upgradable' to see them.
creality@creality-VirtualBox:~$ sudo apt upgrade
```

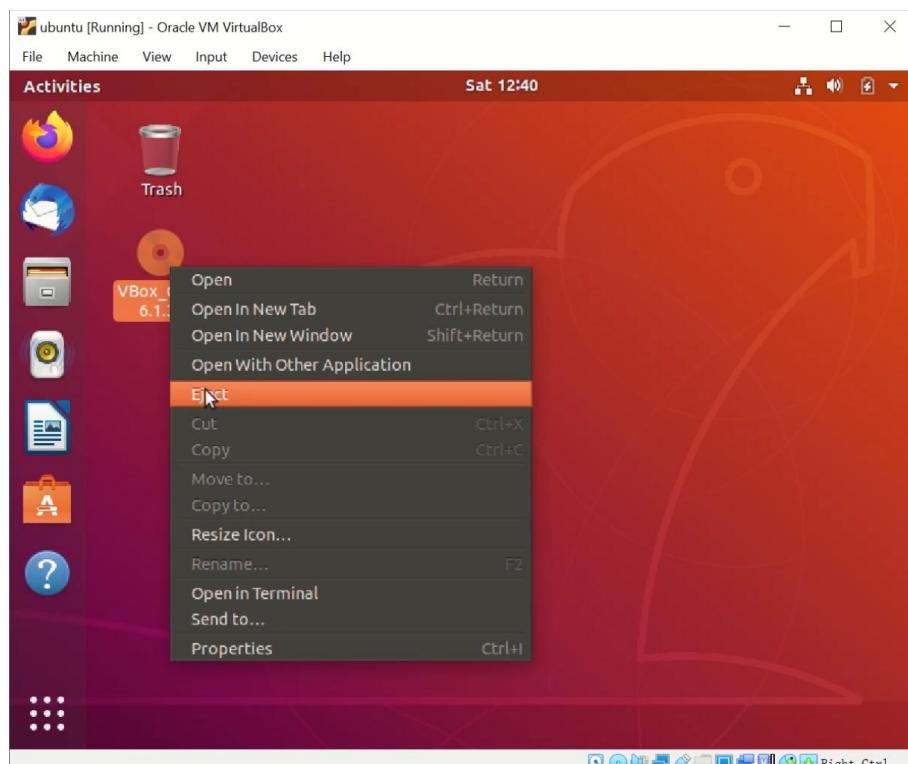
The terminal window has a dark background with light-colored text. The title bar indicates the session is running in Oracle VM VirtualBox. The left sidebar shows various application icons.

Close the terminal window when completed,

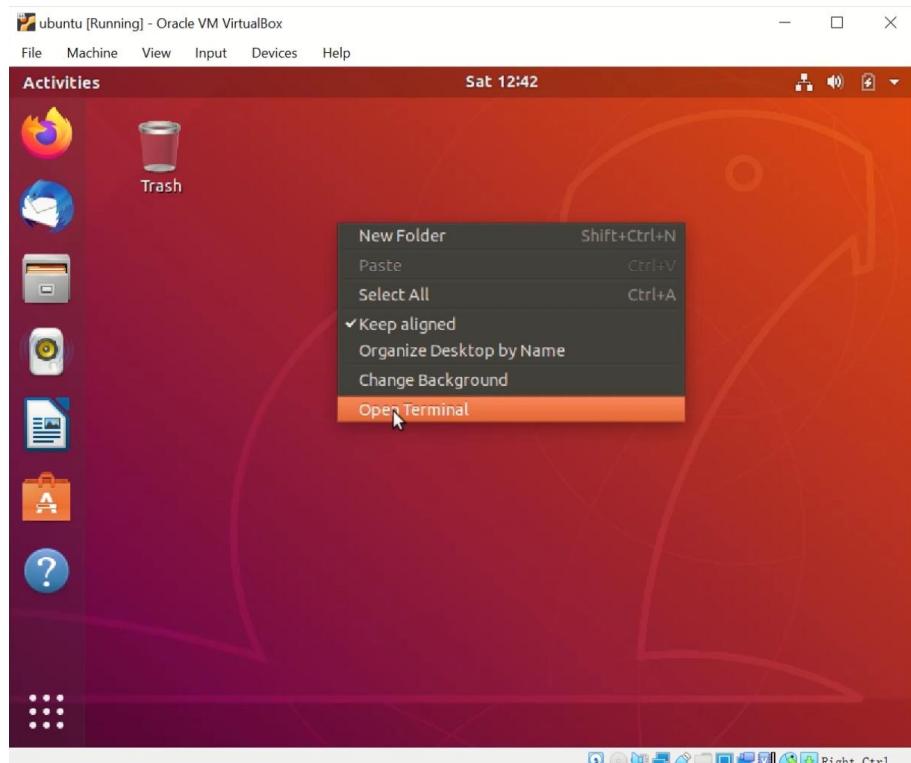
Devices -- Install Insert Guest additions CD image



Eject VBox_Gas after installation is completed

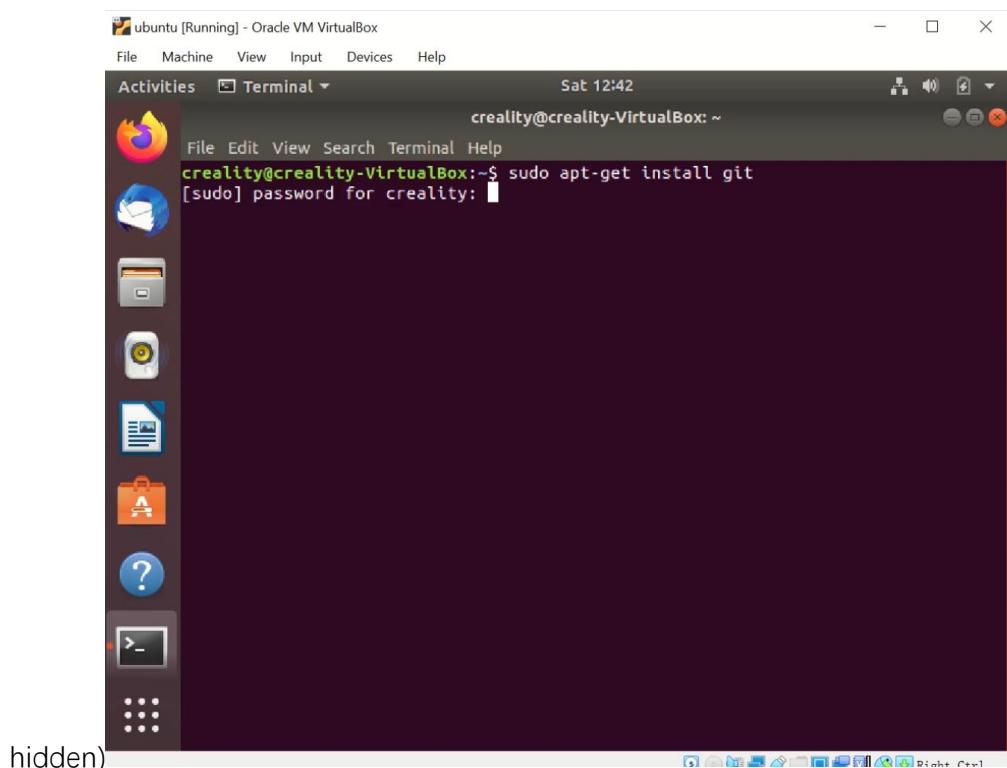


Desktop--right-click--open terminal



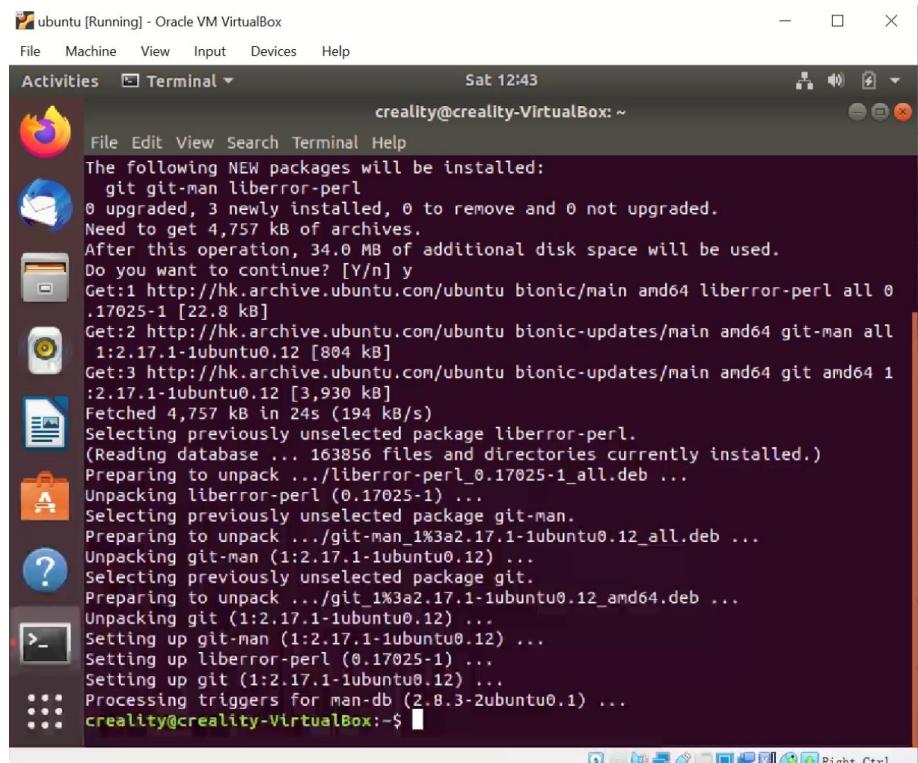
Enter "sudo apt-get install git" and press Enter (install git)

Enter the system password and press Enter (the password is automatically



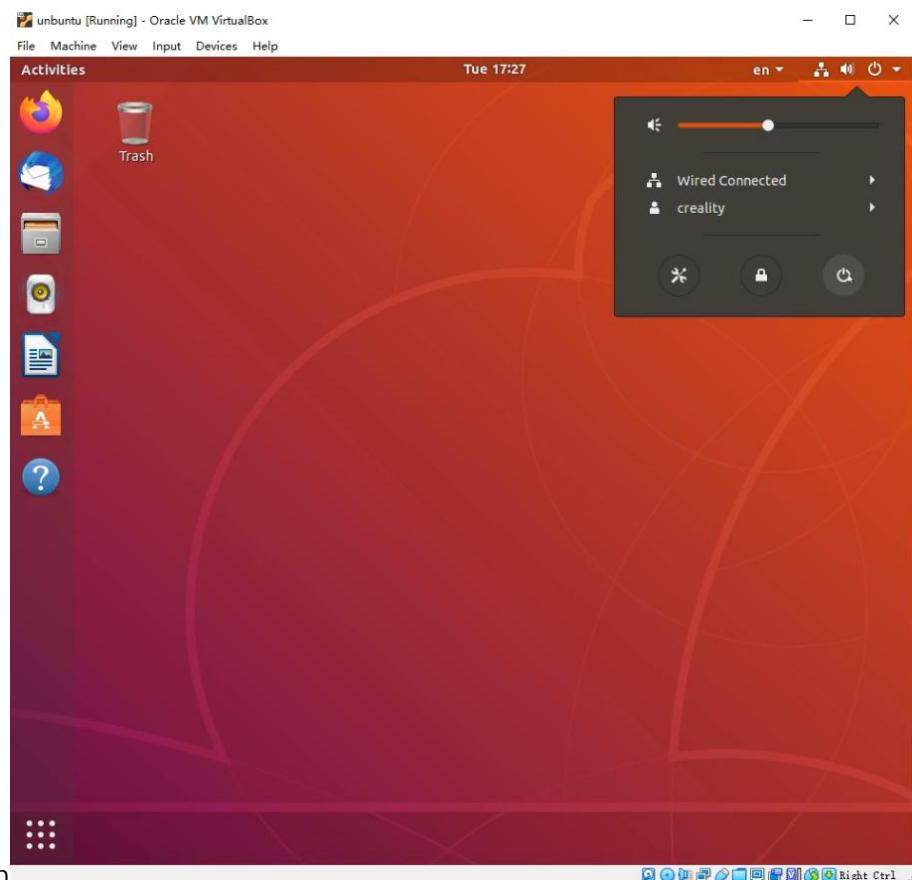
hidden)

Installation is completed



The following NEW packages will be installed:
git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,757 kB of archives.
After this operation, 34.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://hk.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0
.17025-1 [22.8 kB]
Get:2 http://hk.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all
1:2.17.1-1ubuntu0.12 [804 kB]
Get:3 http://hk.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1
2:1.17.1-1ubuntu0.12 [3,930 kB]
Fetched 4,757 kB in 24s (194 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 163856 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.17.1-1ubuntu0.12_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.12) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.12_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.12) ...
Setting up git-man (1:2.17.1-1ubuntu0.12) ...
Setting up liberror-perl (0.17025-1) ...
Setting up git (1:2.17.1-1ubuntu0.12) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
creality@creality-VirtualBox:~\$

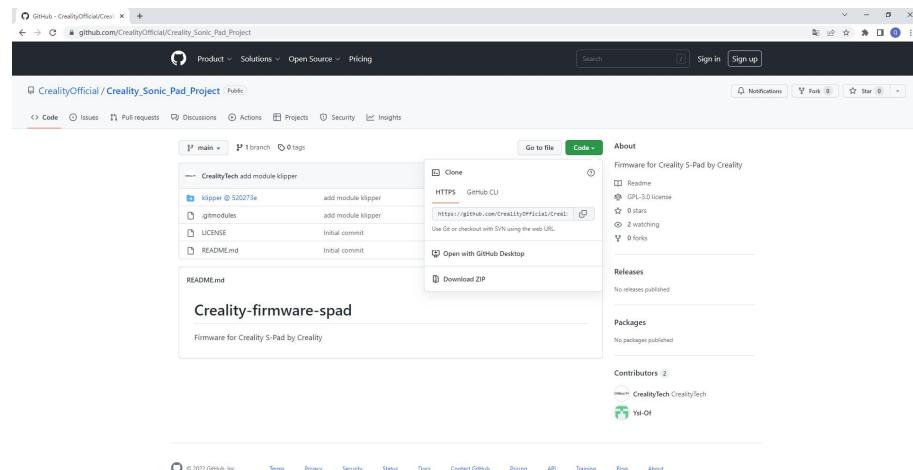
Restart the



system

5、Download Klipper firmware

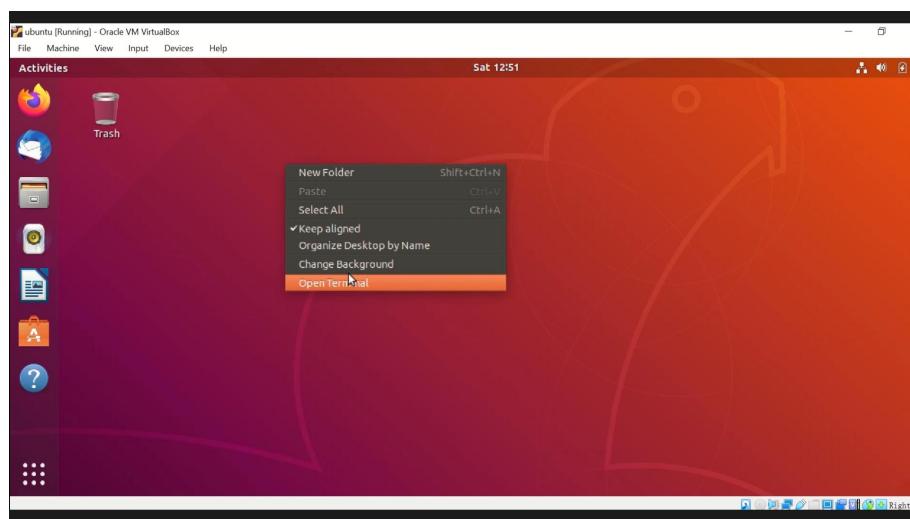
Copy the Klipper firmware address link at www.github.com/CreativityOfficial/Creativity_Sonic_Pad_Project



Klipper firmware address

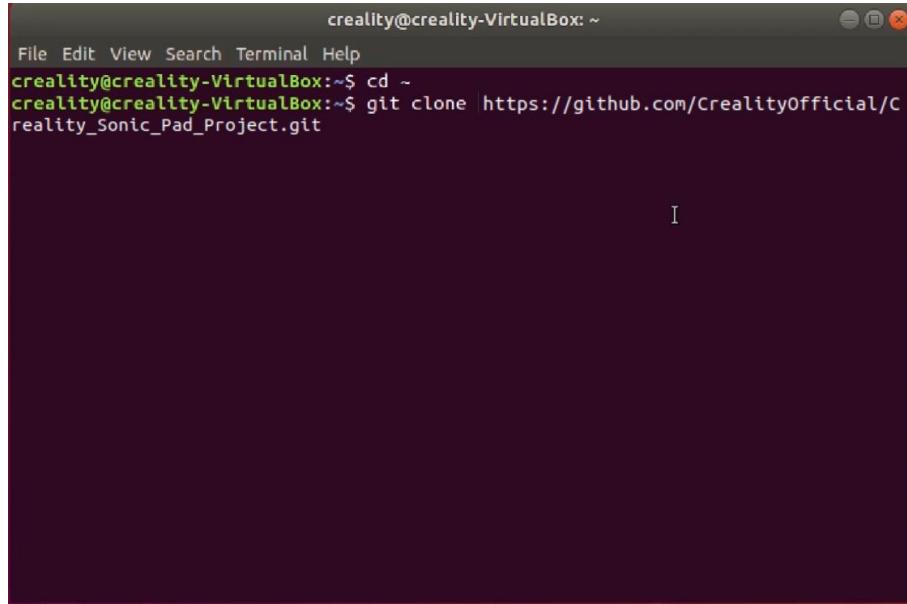
https://github.com/CreativityOfficial/Creativity_Sonic_Pad_Project.git

Desktop--right-click--open terminal



Enter "cd ~" and press Enter (switch the working directory to the home directory)

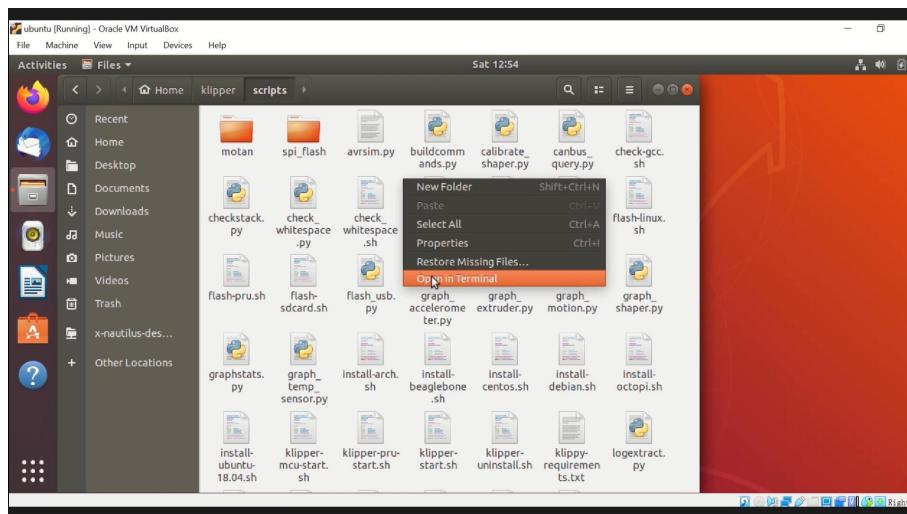
Enter "git clone" and paste the Klipper firmware address link -- Enter



```
creality@creality-VirtualBox:~$ cd ~
creality@creality-VirtualBox:~$ git clone https://github.com/CrealityOfficial/creality_Sonic_Pad_Project.git
```

6、Configure printer firmware

Open the file manager-Klipper folder-scripts folder-Right click-open in terminal

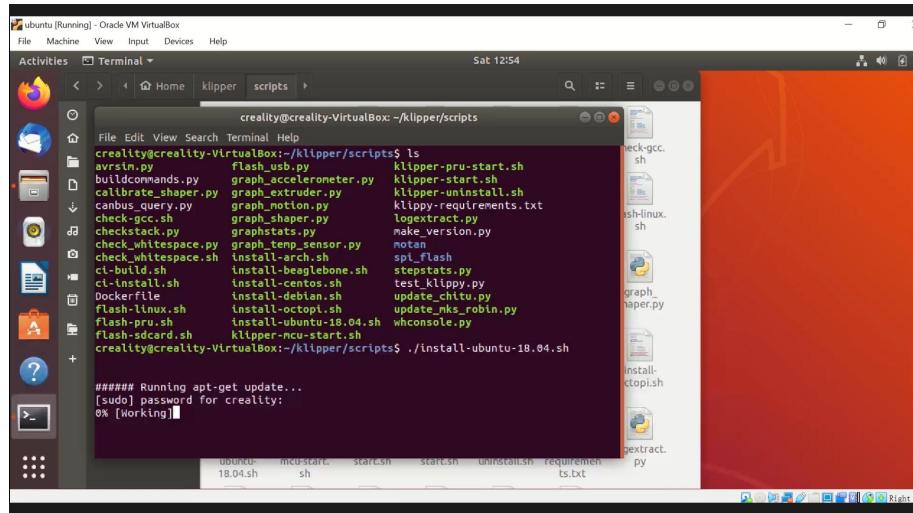


Enter "ls" and press Enter

(confirm that the content in the current directory is consistent with that in the folder)

Enter "./install-ubuntu-18.04.sh" and press Enter (install ubuntu)

Enter the system password and press Enter (the password is automatically hidden)

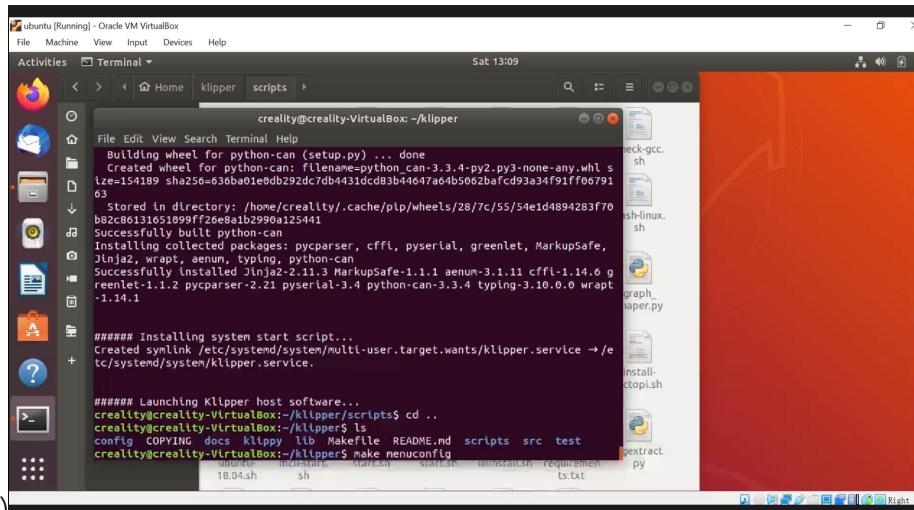


Enter "cd .." and then press Enter (switch to the upper directory)

Enter "ls" and press Enter

(confirm that the content in the current directory is consistent with that in the folder)

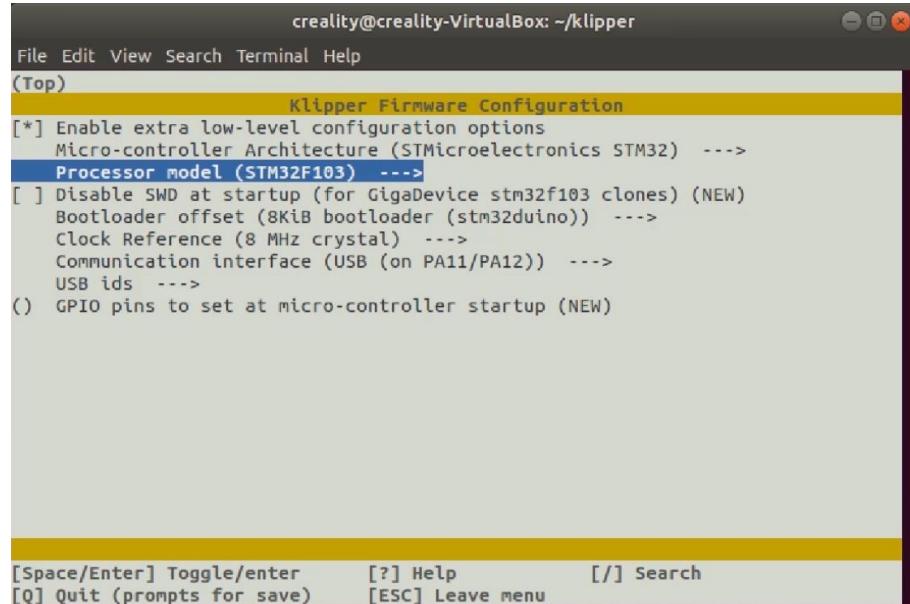
Enter "make menuconfig" and press Enter (configure the printer



motherboard)

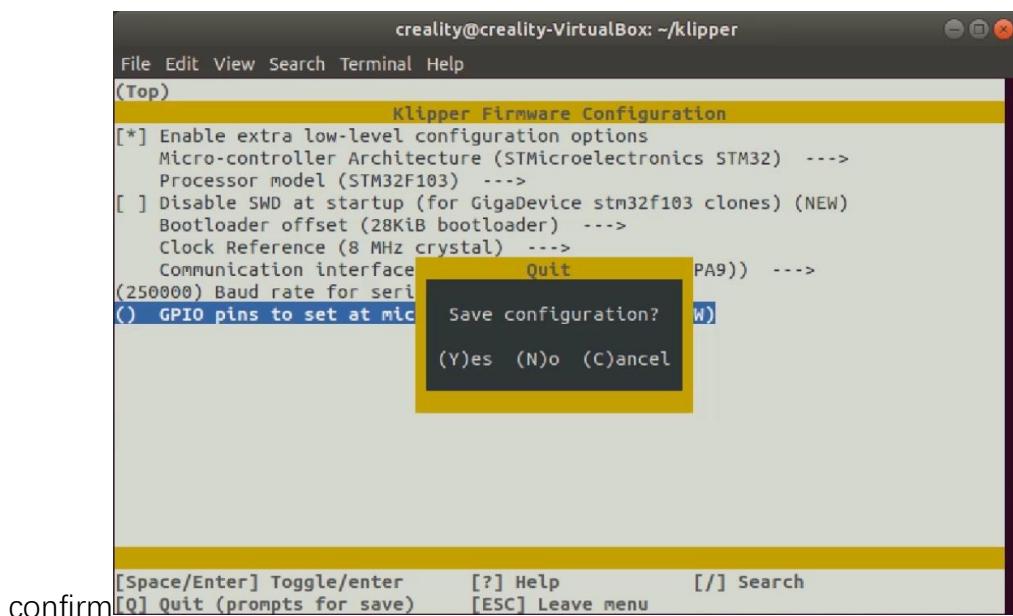
Select the Micro-controller Architecture、Processor model、Bootloader Offset、Clock Reference、Communication interface etc. parameters

(Example, please configure it according to the motherboard)



After the parameter configuration is completed, press "Q" to save,

Press "Y" to



Enter "make" and press Enter (generate firmware)

```
creality@creality-VirtualBox: ~/klipper
File Edit View Search Terminal Help
63
Stored in directory: /home/creality/.cache/pip/wheels/28/7c/55/54e1d4894283f70
b82c86131651099ff26e8a1b2990a125441
Successfully built python-can
Installing collected packages: pycparser, cffi, pyserial, greenlet, MarkupSafe,
Jinja2, wrapt, aenum, typing, python-can
Successfully installed Jinja2-2.11.3 MarkupSafe-1.1.1 aenum-3.1.11 cffi-1.14.6 g
reenlet-1.1.2 pycparser-2.21 pyserial-3.4 python-can-3.3.4 typing-3.10.0.0 wrapt
-1.14.1

##### Installing system start script...
Created symlink /etc/systemd/system/multi-user.target.wants/klipper.service → /e
tc/systemd/system/klipper.service.

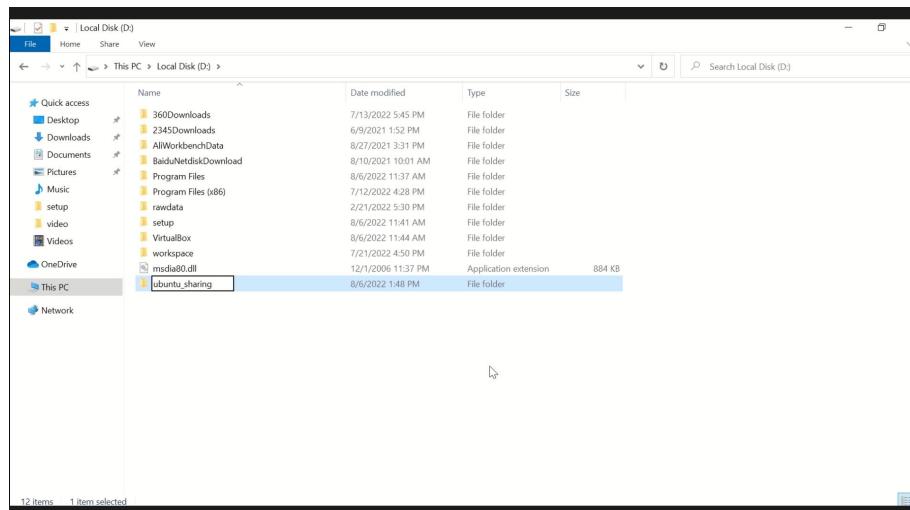
##### Launching Klipper host software...
creality@creality-VirtualBox:~/klipper/scripts$ cd ..
creality@creality-VirtualBox:~/klipper$ ls
config COPYING docs klippy lib Makefile README.md scripts src test
creality@creality-VirtualBox:~/klipper$ make menuconfig
Loaded configuration '/home/creality/klipper/.config'
Configuration saved to '/home/creality/klipper/.config'
creality@creality-VirtualBox:~/klipper$ make
```

Find the firmware storage location

```
creality@creality-VirtualBox: ~/klipper
File Edit View Search Terminal Help
Compiling out/src/tmcuart.o
Compiling out/src/neopixel.o
Compiling out/src/pulse_counter.o
Compiling out/src/stm32/watchdog.o
Compiling out/src/stm32/gpio.o
Compiling out/src/generic/crc16_ccitt.o
Compiling out/src/generic/armcm_boot.o
Compiling out/src/generic/armcm_irq.o
Compiling out/src/generic/armcm_reset.o
Compiling out/src/..lib/stm32f1/system_stm32f1xx.o
Compiling out/src/stm32/stm32f1.o
Compiling out/src/generic/armcm_timer.o
Compiling out/src/stm32adc.o
Compiling out/src/stm32/i2c.o
Compiling out/src/stm32/spi.o
Compiling out/src/stm32/serial.o
Compiling out/src/generic/serial_irq.o
Compiling out/src/stm32/hard_pwm.o
Building out/compile_time_request.o
Version: v0.10.0-169-g520273e5
Preprocessing out/src/generic/armcm_link.ld
Linking out/klipper.elf
Creating hex file out/klipper.bin
creality@creality-VirtualBox:~/klipper$
```

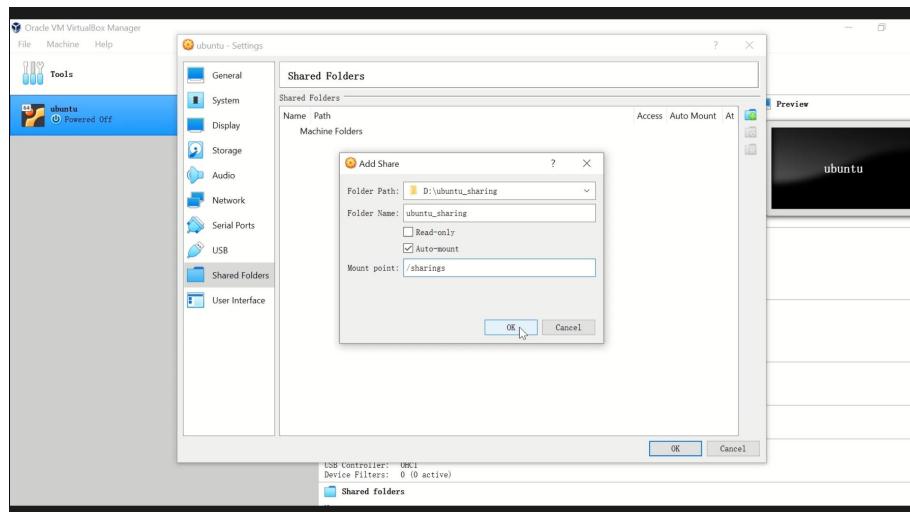
7、Upgrade printer firmware via SD/TF card

Create a new "ubuntu_sharing" on the computer disk

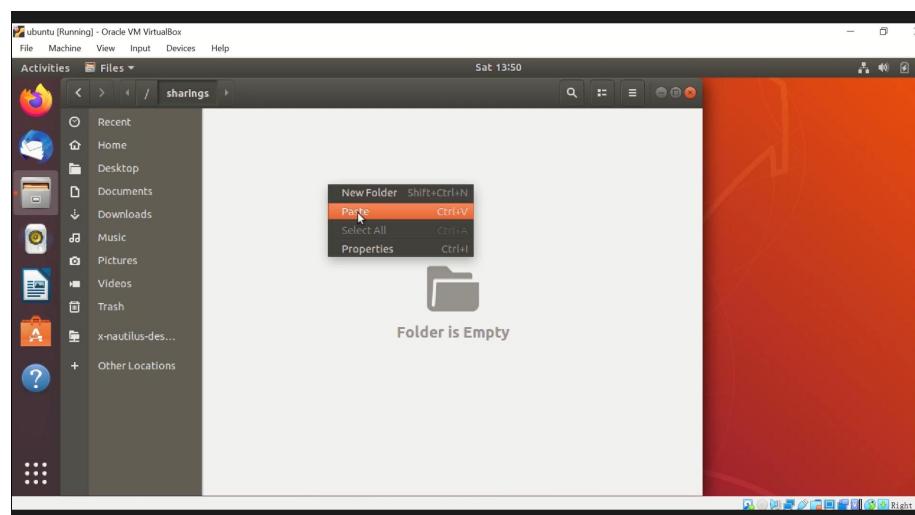
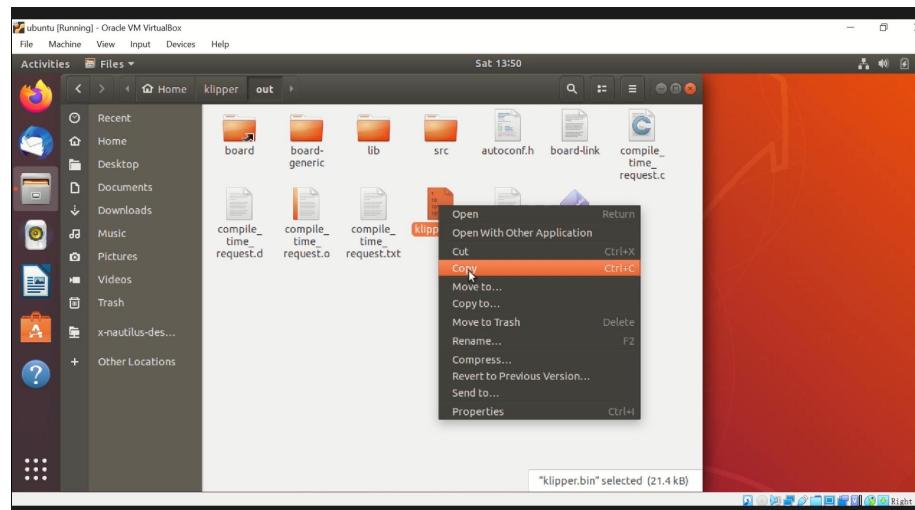


Open VirtualBox--Settings--Shared Folders

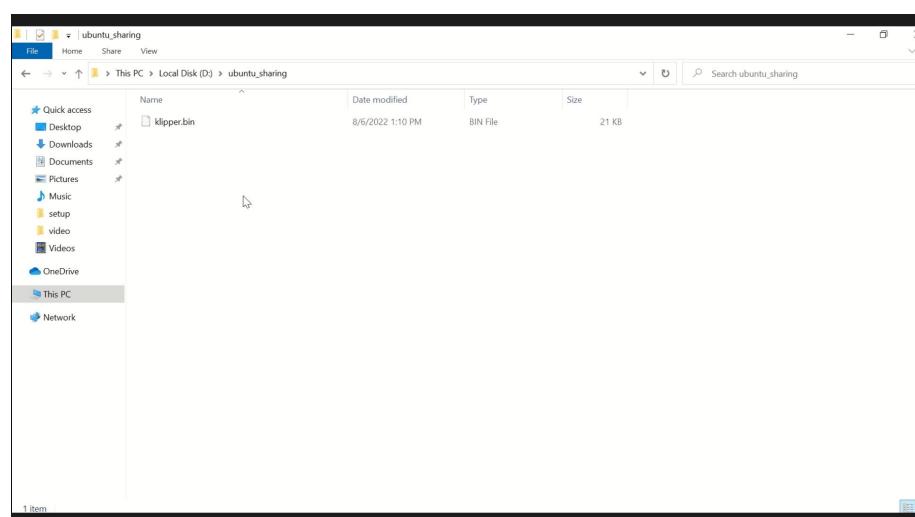
Add a shared folder (tick auto mount), Mount point type "/sharings"



Copy the klipper.bin file just generated to the sharings folder



Copy klipper.bin in the ubuntu_sharing folder of the computer disk to the TF/SD

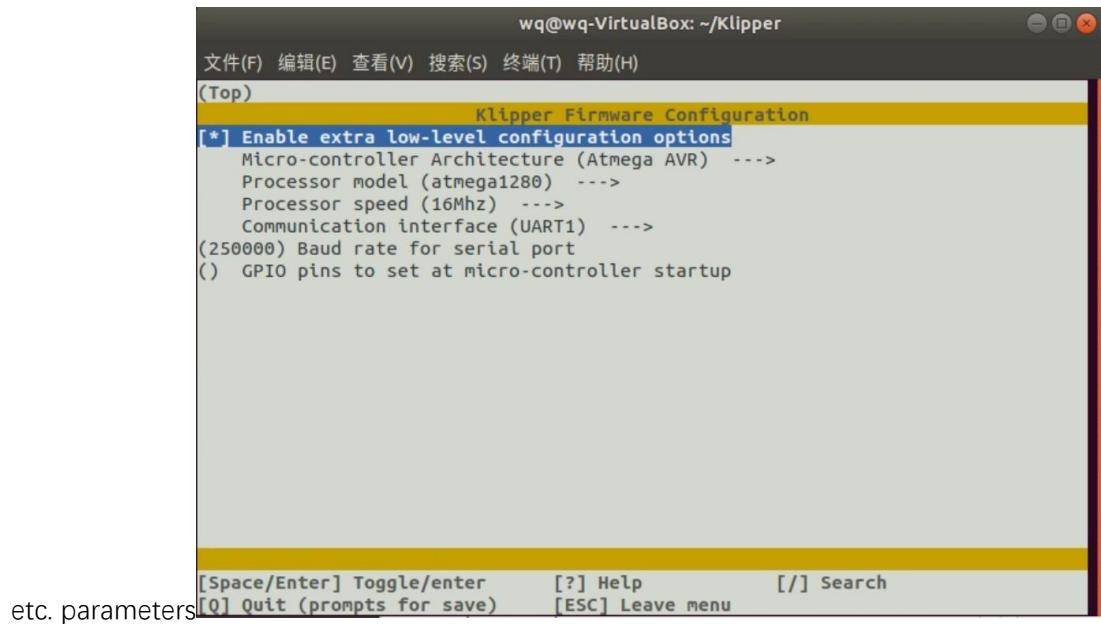


8、Upgrade printer firmware via USB

(The motherboard has no Bootloader and does not support SD card upgrades)

Please use a USB cable to connect the printer to the computer where the virtual machine is located. Please keep the connection during the firmware upgrade process

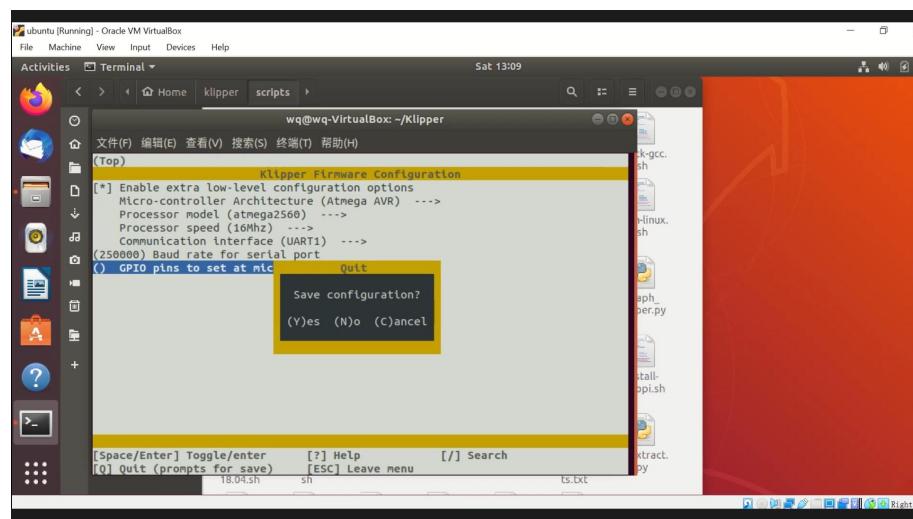
Refer to the previous video to enter the configuration motherboard interface
(Example, please configure it according to the motherboard)
Select the Micro-controller Architecture、Processor model、Clock Reference、Communication interface



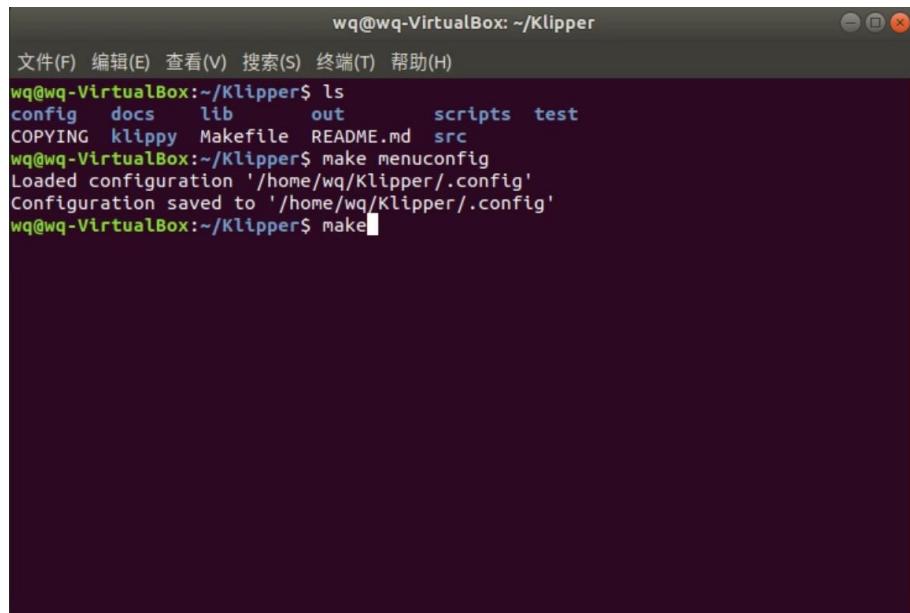
etc. parameters

After the parameter configuration is completed, press "Q" to save,

Press "Y" to confirm



Enter "make" and press Enter (generate firmware)

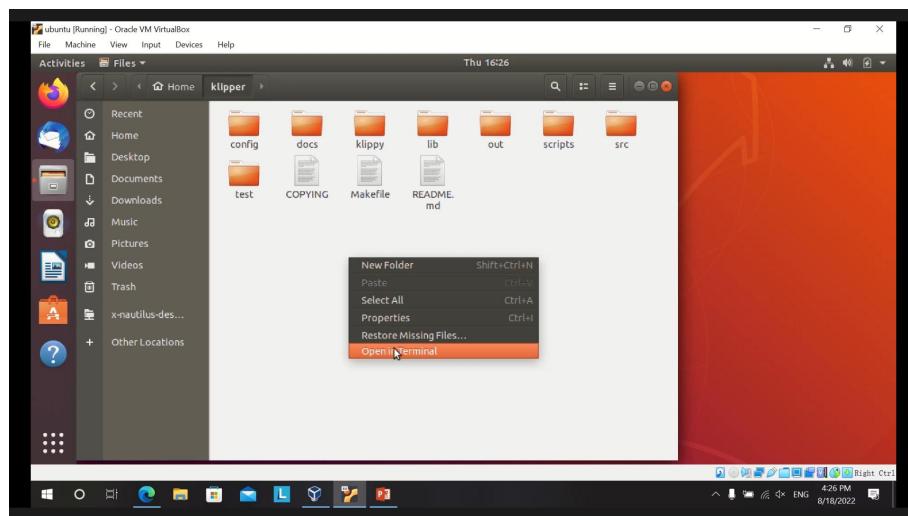


```
wq@wq-VirtualBox: ~/Klipper
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
wq@wq-VirtualBox:~/Klipper$ ls
config  docs  lib    out      scripts  test
COPYING  klippy  Makefile  README.md  src
wq@wq-VirtualBox:~/Klipper$ make menuconfig
Loaded configuration '/home/wq/Klipper/.config'
Configuration saved to '/home/wq/Klipper/.config'
wq@wq-VirtualBox:~/Klipper$ make
```

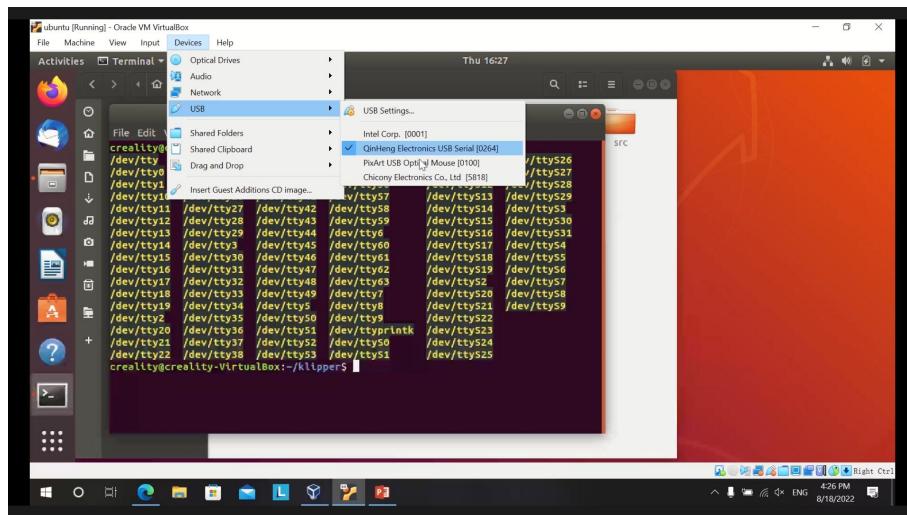
Find the firmware storage

```
wq@wq-VirtualBox: ~/Klipper
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
Compiling out/src/pwmcmds.o
Compiling out/src/spi_software.o
Compiling out/src/sensor_adxl345.o
Compiling out/src/lcd_st7920.o
Compiling out/src/lcd_hd44780.o
Compiling out/src/buttons.o
Compiling out/src/tmcuart.o
Compiling out/src/neopixel.o
Compiling out/src/pulse_counter.o
Compiling out/src/avr/main.o
Compiling out/src/avr/timer.o
Compiling out/src/avr/gpio.o
Compiling out/src/avr/adc.o
Compiling out/src/avr/spi.o
Compiling out/src/avr/i2c.o
Compiling out/src/avr/hard_pwm.o
Compiling out/src/avr/watchdog.o
Compiling out/src/avr/serial.o
Compiling out/src/generic/serial_irq.o
Building out/compile_time_request.o
Version: v0.10.0-169-g520273e5-20220913_155730-wq-VirtualBox
Linking out/klipper.elf
Creating hex file out/klipper.elf.hex
location wq@wq-VirtualBox:~/Klipper$
```

File Manager--Klipper Folder--Right click--open in terminal



Device--USB--Tick the USB option for connection to the printer



Enter "ls /dev/tty*" and press Enter

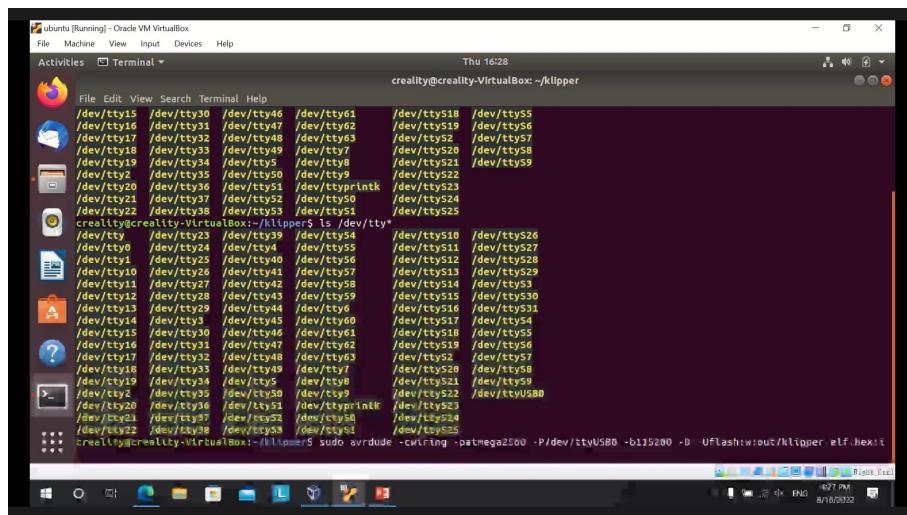
(Confirm that /dev/ttyUSB0 appears in the directory)

Enter "sudo avrdude -cwiring -patmega2560 -P/dev/ttyUSB0 -b115200 -D

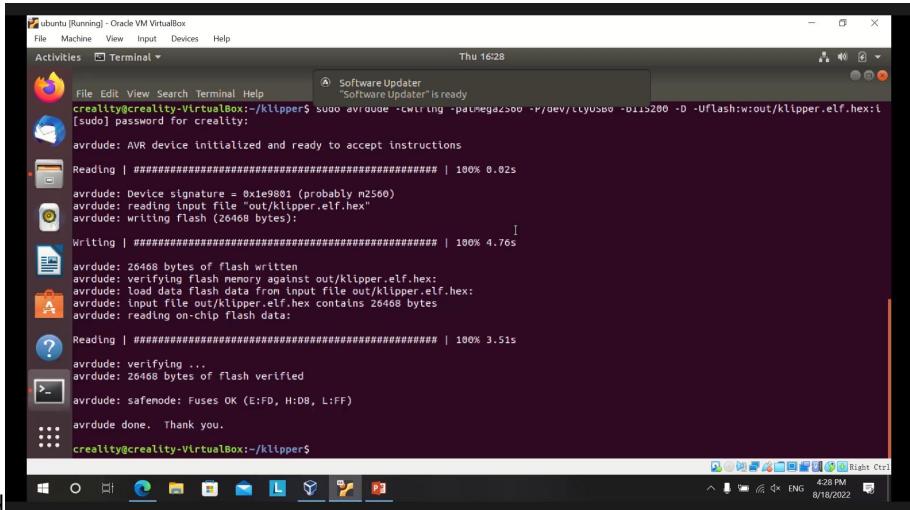
-Uflash:w:out/klipper.elf.hex:i" and press Enter

(Transfer the klipper firmware to the printer for upgrade,

Note the space before the "-" in the command)



Update



```
ubuntu [Running] - Oracle VM VirtualBox
File Edit View Search Terminal Help
Activities Terminal Thu 16:28
creality@creality-VirtualBox:~/klipper$ sudo avrdude -cwiring -patmega328 -P/dev/ttyUSB0 -D -Uflash:w:out/klipper.elf.hex:i
[sudo] password for creality:
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.02s
avrdude: Device signature = 0x1e9801 (probably m2560)
avrdude: reading input file "out/klipper.elf.hex"
avrdude: writing flash (26408 bytes):
Writing | ##### | 100% 4.76s
avrdude: 26408 bytes of flash written
avrdude: verifying flash memory against out/klipper.elf.hex:
avrdude: load data flash data from input file out/klipper.elf.hex:
avrdude: input file out/klipper.elf.hex contains 26408 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 3.51s
avrdude: verifying ...
avrdude: 26408 bytes of flash verified
avrdude: safemode: Fuses OK (E:FD, H:D8, L:FF)
avrdude done. Thank you.
creality@creality-VirtualBox:~/klipper$
```

completed