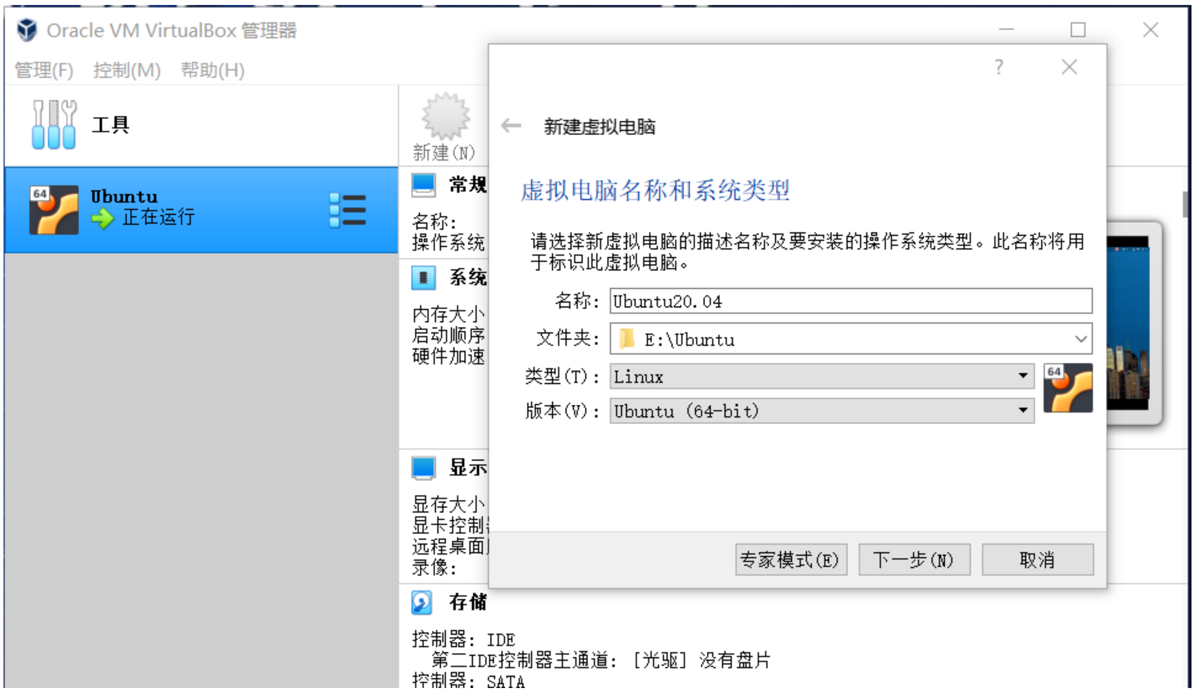
软件安装及编译环境搭建

软件安装及编译环境搭建

点击下载 Virtualbox VM 和 Ubuntu 20.04 LTS

1、于VirtualBoxVM安装Ubuntu

建立新的VM，并加以命名



新建虚拟电脑

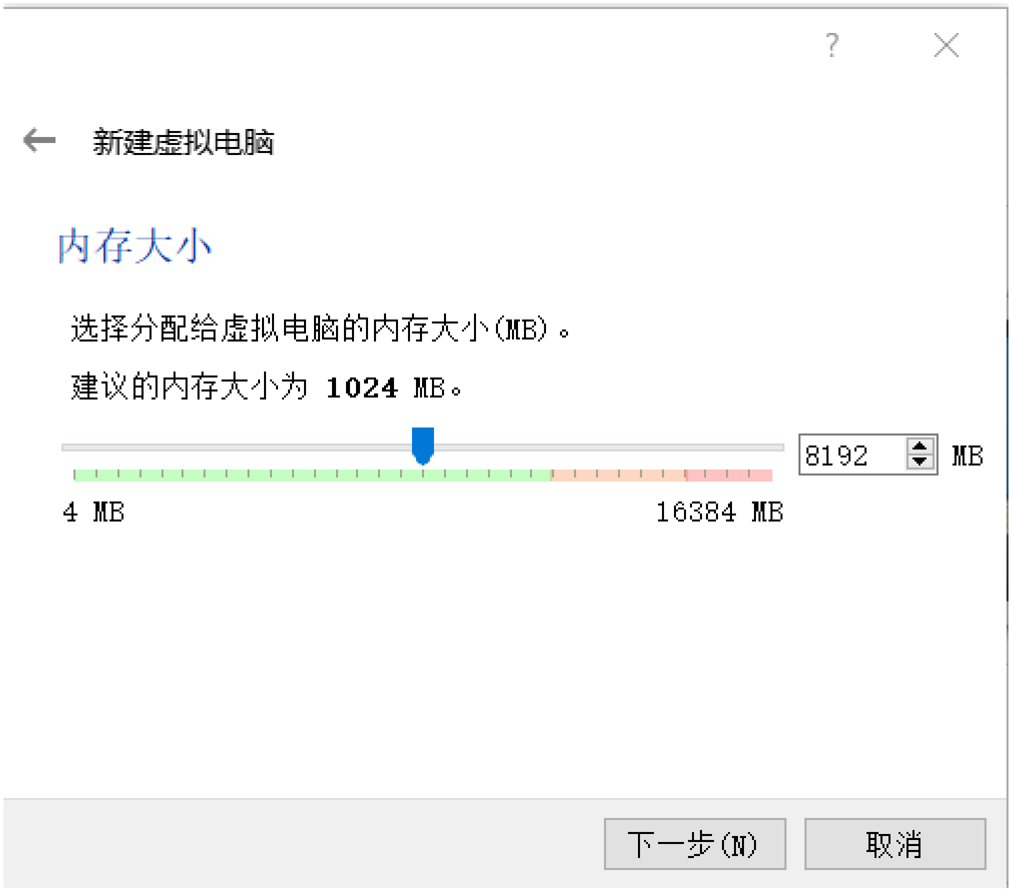
虚拟电脑名称和系统类型

请选择新虚拟电脑的描述名称及要安装的操作系统类型。此名称将用于标识此虚拟电脑。

名称: Ubuntu20.04
文件夹: E:\Ubuntu
类型(T): Linux
版本(V): Ubuntu (64-bit)

控制器: IDE
第二IDE控制器主通道: [光驱] 没有盘片
控制器: SATA

规划8GB内存供VM使用。



新建虚拟电脑

内存大小

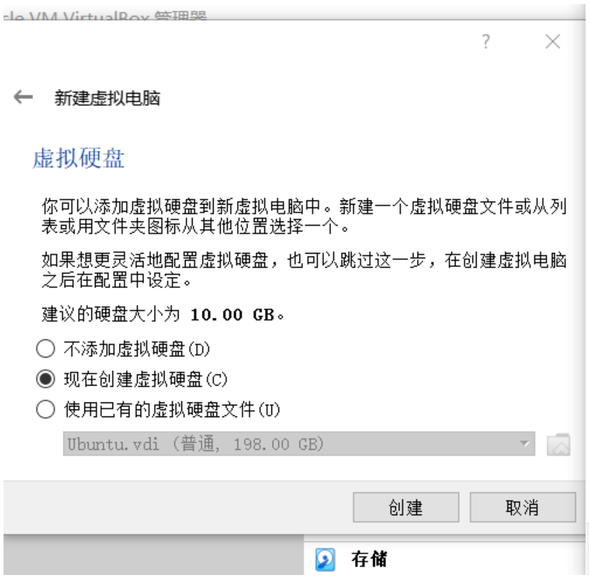
选择分配给虚拟电脑的内存大小(MB)。

建议的内存大小为 **1024** MB。

8192 MB

4 MB　　16384 MB

创建硬盘



虚拟硬盘

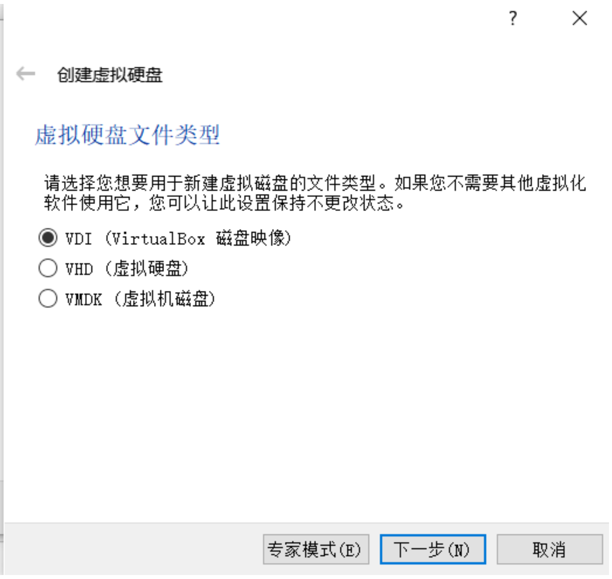
你可以添加虚拟硬盘到新虚拟电脑中。新建一个虚拟硬盘文件或从列表或用文件夹图标从其他位置选择一个。

如果想更灵活地配置虚拟硬盘，也可以跳过这一步，在创建虚拟电脑之后在配置中设定。

建议的硬盘大小为 **10.00 GB**。

- ○ 不添加虚拟硬盘(D)
- [✓ 现在创建虚拟硬盘(C)]
- ○ 使用已有的虚拟硬盘文件(U)

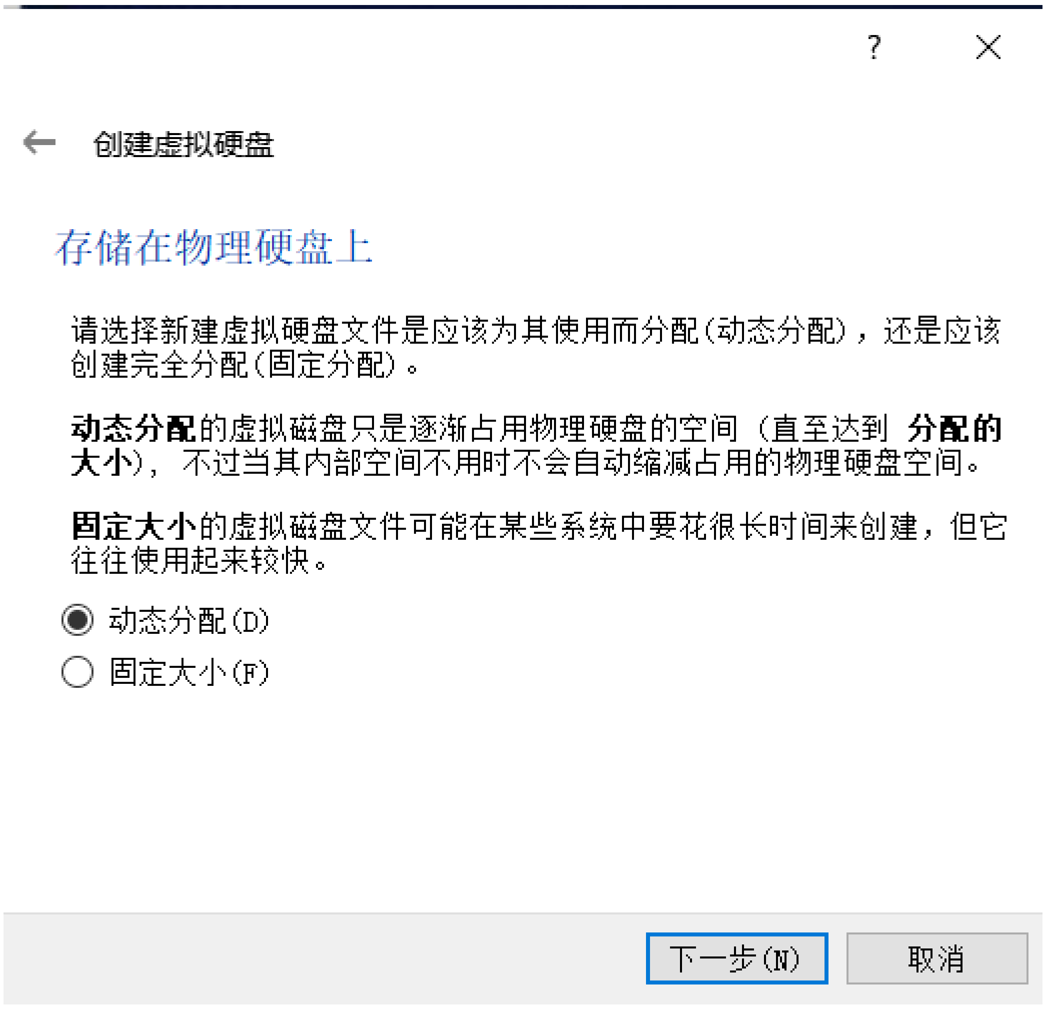
Ubuntu.vdi（普通，198.00 GB）



虚拟硬盘文件类型

请选择您想要用于新建虚拟磁盘的文件类型。如果您不需要其他虚拟化软件使用它，您可以让此设置保持不更改状态。

- [✓ VDI（VirtualBox 磁盘映像）]
- ○ VHD（虚拟硬盘）
- ○ VMDK（虚拟机磁盘）



存储在物理硬盘上

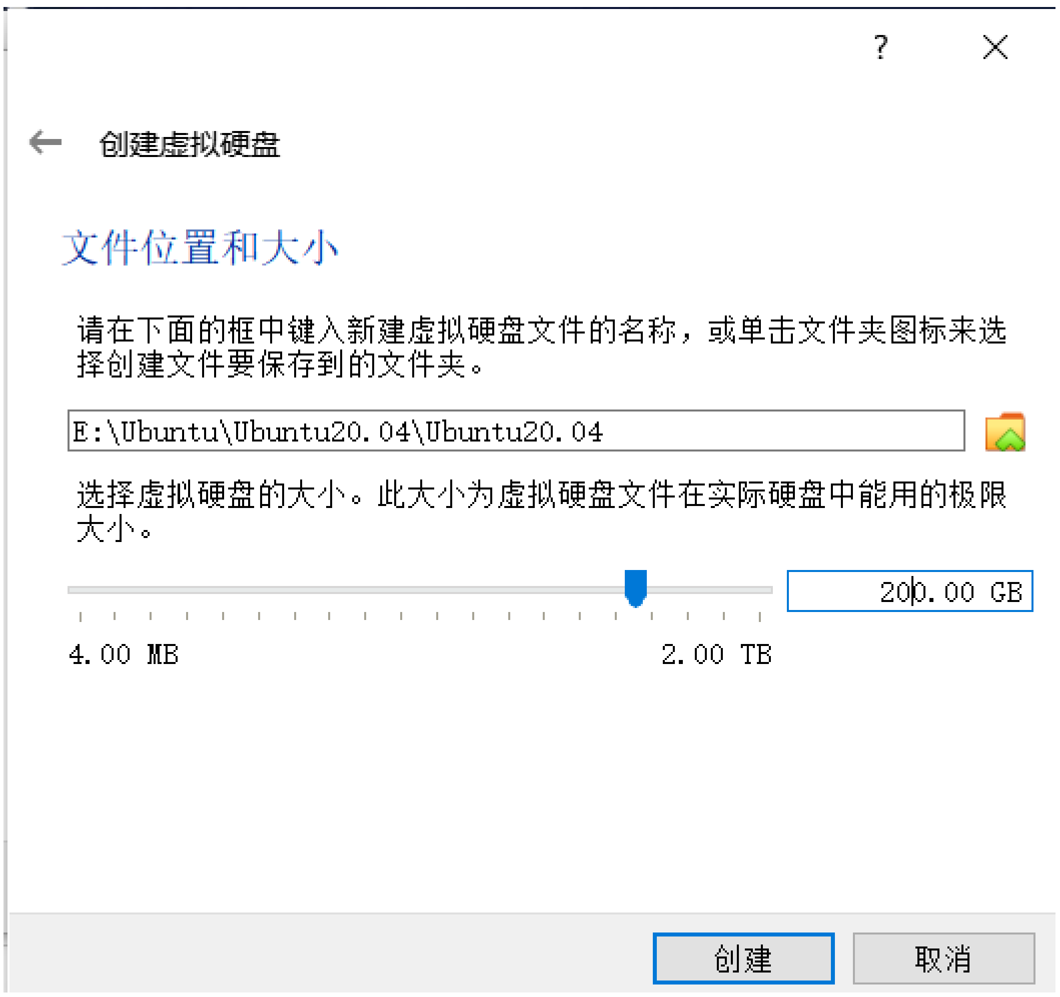
请选择新建虚拟硬盘文件是应该为其使用而分配(动态分配)，还是应该创建完全分配(固定分配)。

动态分配的虚拟磁盘只是逐渐占用物理硬盘的空间（直至达到 **分配的大小**），不过当其内部空间不用时不会自动缩减占用的物理硬盘空间。

固定大小的虚拟磁盘文件可能在某些系统中要花很长时间来创建，但它往往使用起来较快。

- [✓ 动态分配(D)]
- ○ 固定大小(F)

预留200GB硬盘空间，供后续存放SDK用。



创建虚拟硬盘

文件位置和大小

请在下面的框中键入新建虚拟硬盘文件的名称，或单击文件夹图标来选择创建文件要保存到的文件夹。

E:\Ubuntu\Ubuntu20.04\Ubuntu20.04

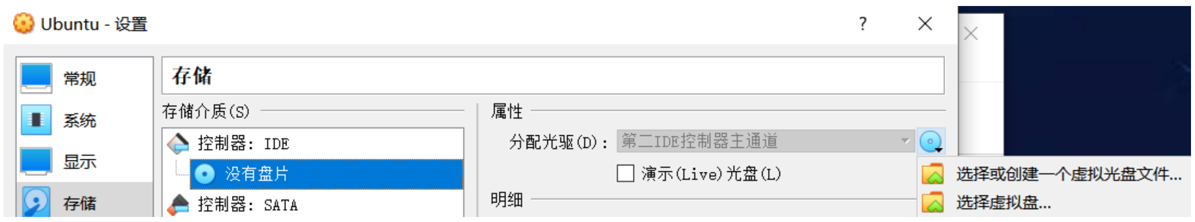
选择虚拟硬盘的大小。此大小为虚拟硬盘文件在实际硬盘中能用的极限大小。

200.00 GB

4.00 MB　　2.00 TB

2、Ubuntu开机设定

第一次开机需要挂载安装光盘ISO档案



Ubuntu - 设置

存储

存储介质(S)

控制器: IDE

没有盘片

控制器: SATA

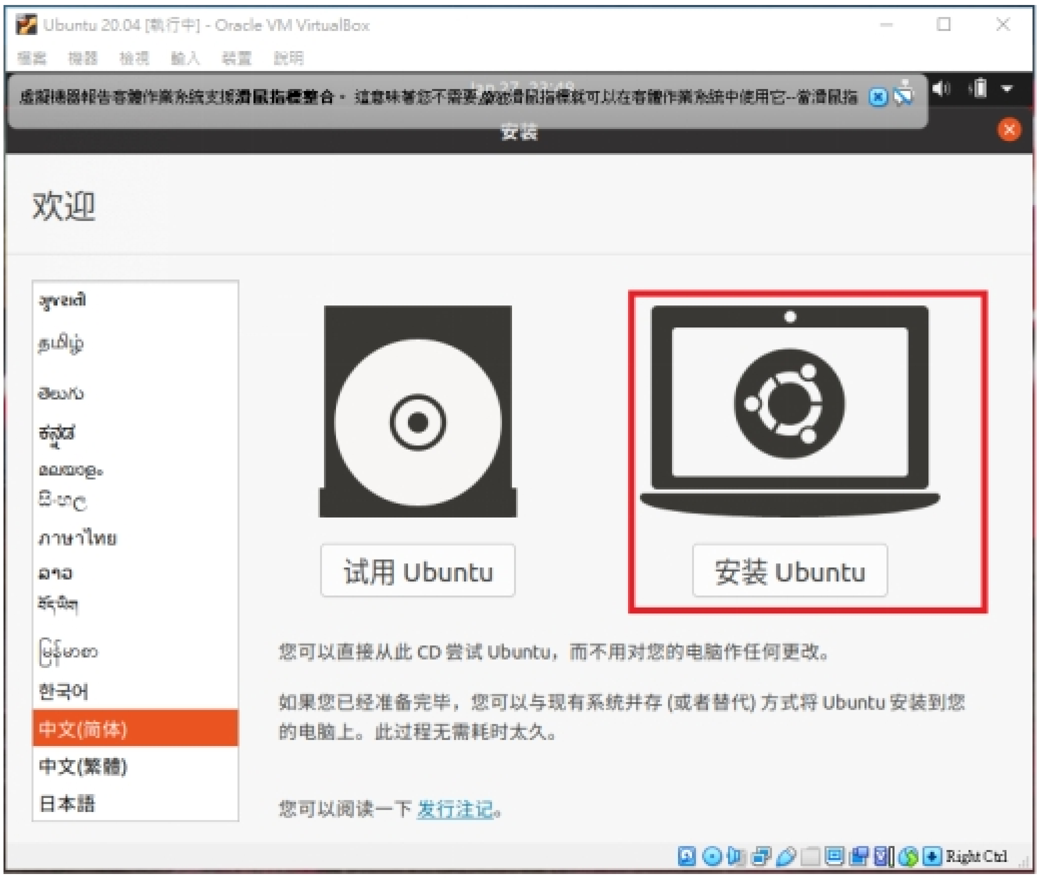
属性

分配光驱(D): 第二IDE控制器主通道

☐ 演示(Live)光盘(L)

明细

开始安装



虛擬機器報告客體作業系統支援滑鼠指標整合。這意味著您不需要擷取滑鼠指標就可以在客體作業系統中使用它 -- 當滑鼠指

您可以直接从此 CD 尝试 Ubuntu，而不用对您的电脑作任何更改。

如果您已经准备完毕，您可以与现有系统并存 (或者替代) 方式将 Ubuntu 安装到您的电脑上。此过程无需耗时太久。

您可以阅读一下 发行注记。

（如需在虚拟机上使用共享剪贴板和共享文件夹等提升使用体验，请自行搜索方法，此处不再赘述）

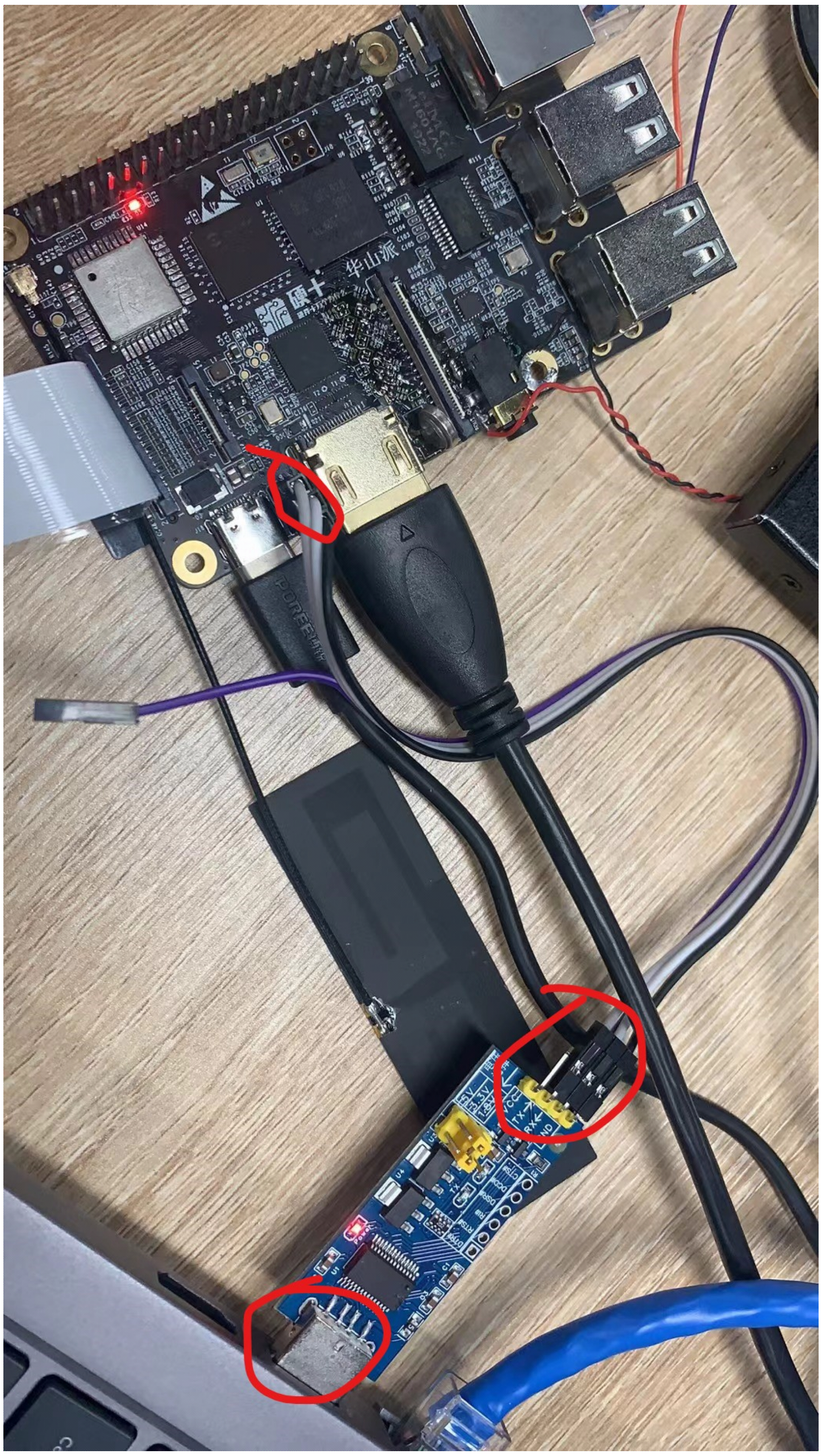
3、MobaXterm安装及开发板连接

官网地址：

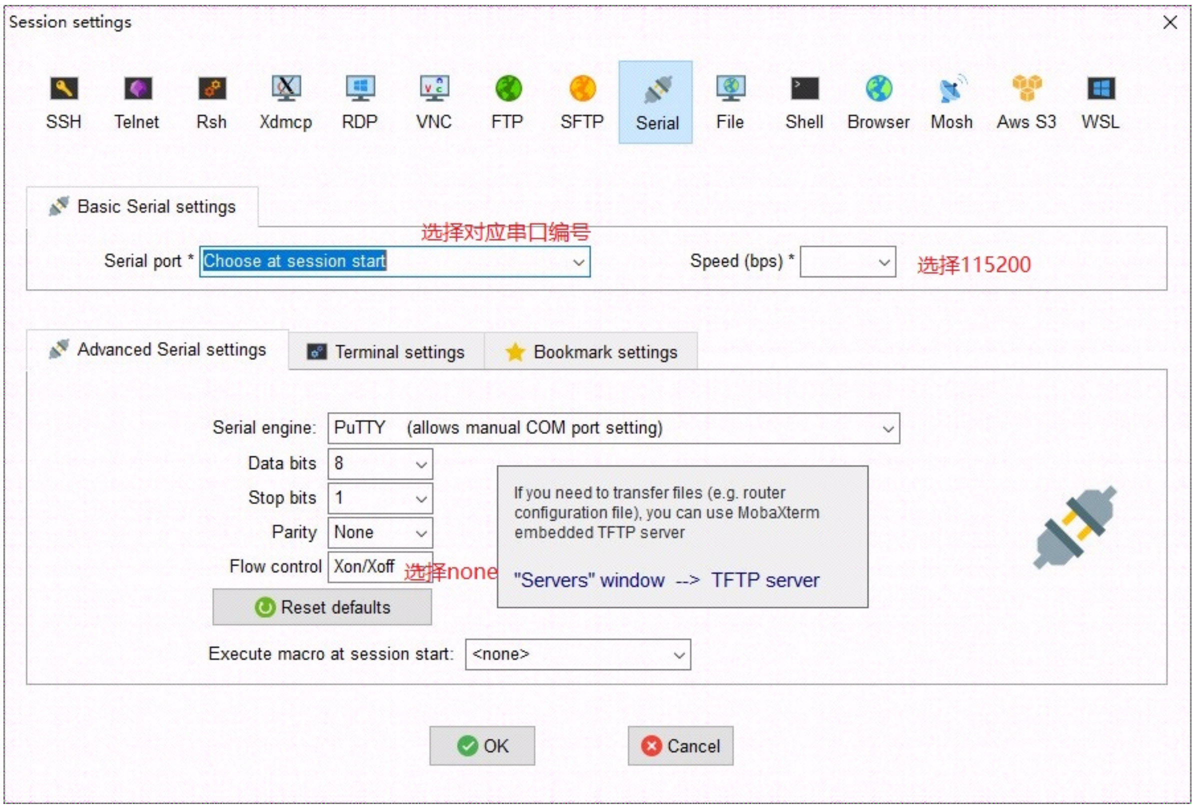
```
https://mobaxterm.mobatek.net/
```

将开发板上通过串口转TTL板与电脑连接

> 注意板端的RX脚接 `usb-ttl` 的TX脚，板端TX脚接 `usb-ttl` 的RX脚



使用mobaxterm配置串口终端



Serial port * Choose at session start — 选择对应串口编号

Speed (bps) * — 选择115200

Serial engine: PuTTY (allows manual COM port setting)

Data bits 8

Stop bits 1

Parity None

Flow control Xon/Xoff — 选择none

If you need to transfer files (e.g. router configuration file), you can use MobaXterm embedded TFTP server

"Servers" window --> TFTP server

Execute macro at session start: <none>

> 终端显示

```
[root@cvitek]~#
```

4、套件安装以及SDK编译

在编译SDK之前，Ubuntu需要安装以下套件：

```
sudo apt-get update
sudo apt-get install -y build-essential
sudo apt-get install -y ninja-build
sudo apt-get install -y automake
sudo apt-get install -y autoconf
sudo apt-get install -y libtool
sudo apt-get install -y wget
sudo apt-get install -y curl
sudo apt-get install -y git
sudo apt-get install -y gcc
sudo apt-get install -y libssl-dev
sudo apt-get install -y bc
sudo apt-get install -y slib
sudo apt-get install -y squashfs-tools
sudo apt-get install -y android-sdk-libsparse-utils
sudo apt-get install -y android-sdk-ext4-utils
sudo apt-get install -y jq
sudo apt-get install -y cmake
sudo apt-get install -y python3-distutils
sudo apt-get install -y tclsh
sudo apt-get install -y scons
sudo apt-get install -y parallel
sudo apt-get install -y ssh-client
sudo apt-get install -y tree
sudo apt-get install -y python3-dev
sudo apt-get install -y python3-pip
sudo apt-get install -y device-tree-compiler
sudo apt-get install -y libssl-dev
sudo apt-get install -y ssh
sudo apt-get install -y cpio
sudo apt-get install -y squashfs-tools
sudo apt-get install -y fakeroot
sudo apt-get install -y libncurses5
sudo apt-get install -y flex
sudo apt-get install -y bison
```

检查cmake版本

```
cmake --version
```

如果小于版本号3.16则需要更新cmake，以3.16.5为例

```
# 注意先删除低版本cmake
sudo apt autoremove cmake

wget https://cmake.org/files/v3.16/cmake-3.16.5.tar.gz
tar zxvf cmake-3.16.5.tar.gz
cd cmake-3.16.5/
./configure
make
sudo make install
```

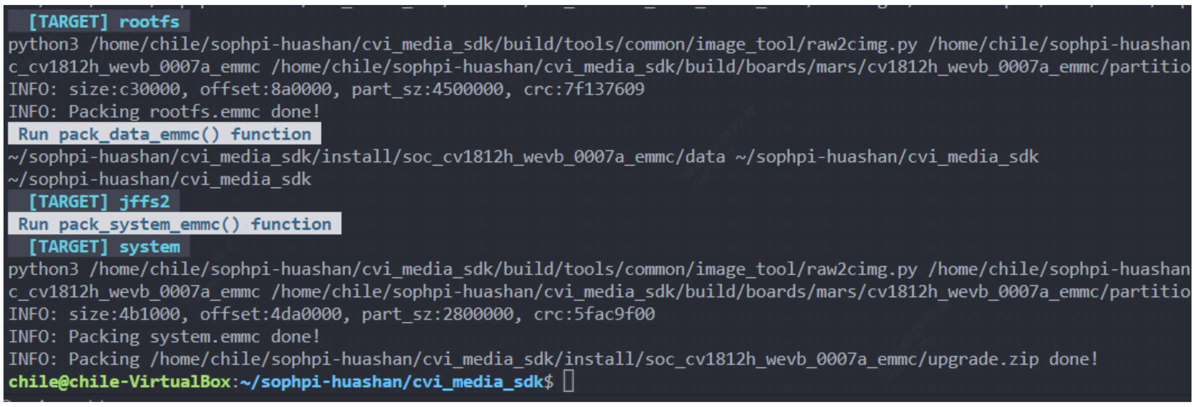
获取SDK源码

```
git clone https://github.com/sophgo/sophpi-huashan.git
```

编译

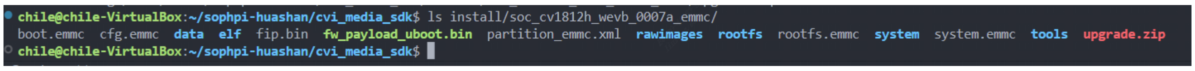
```
1.cd cvi_media_sdk/
2.source build/cvisetup.sh                 # 配置编译环境
3.defconfig cv1812h_wevb_0007a_emmc
4.build_all                                # 编译SDK
```

> 编译成功



```
 [TARGET] rootfs
python3 /home/chile/sophpi-huashan/cvi_media_sdk/build/tools/common/image_tool/raw2cimg.py /home/chile/sophpi-huashan
c_cv1812h_wevb_0007a_emmc /home/chile/sophpi-huashan/cvi_media_sdk/build/boards/mars/cv1812h_wevb_0007a_emmc/partitio
INFO: size:c30000, offset:8a0000, part_sz:4500000, crc:7f137609
INFO: Packing rootfs.emmc done!
 Run pack_data_emmc() function
~/sophpi-huashan/cvi_media_sdk/install/soc_cv1812h_wevb_0007a_emmc/data ~/sophpi-huashan/cvi_media_sdk
~/sophpi-huashan/cvi_media_sdk
 [TARGET] jffs2
 Run pack_system_emmc() function
 [TARGET] system
python3 /home/chile/sophpi-huashan/cvi_media_sdk/build/tools/common/image_tool/raw2cimg.py /home/chile/sophpi-huashan
c_cv1812h_wevb_0007a_emmc /home/chile/sophpi-huashan/cvi_media_sdk/build/boards/mars/cv1812h_wevb_0007a_emmc/partitio
INFO: size:4b1000, offset:4da0000, part_sz:2800000, crc:5fac9f00
INFO: Packing system.emmc done!
INFO: Packing /home/chile/sophpi-huashan/cvi_media_sdk/install/soc_cv1812h_wevb_0007a_emmc/upgrade.zip done!
chile@chile-VirtualBox:~/sophpi-huashan/cvi_media_sdk$
```

生成的镜像在`install/soc_cv1812h_wevb_0007a_emmc`下



```
chile@chile-VirtualBox:~/sophpi-huashan/cvi_media_sdk$ ls install/soc_cv1812h_wevb_0007a_emmc/
boot.emmc  cfg.emmc  data  elf  fip.bin  fw_payload_uboot.bin  partition_emmc.xml  rawimages  rootfs  rootfs.emmc  system  system.emmc  tools  upgrade.zip
chile@chile-VirtualBox:~/sophpi-huashan/cvi_media_sdk$
```