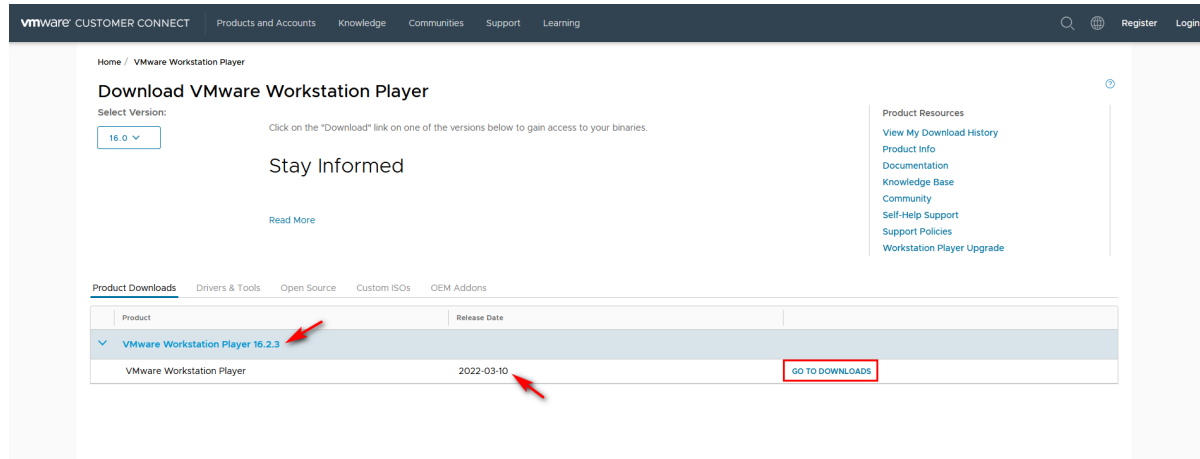


OpenCPU用户开发环境搭建手册

1 虚拟机工具下载安装

1.1 下载

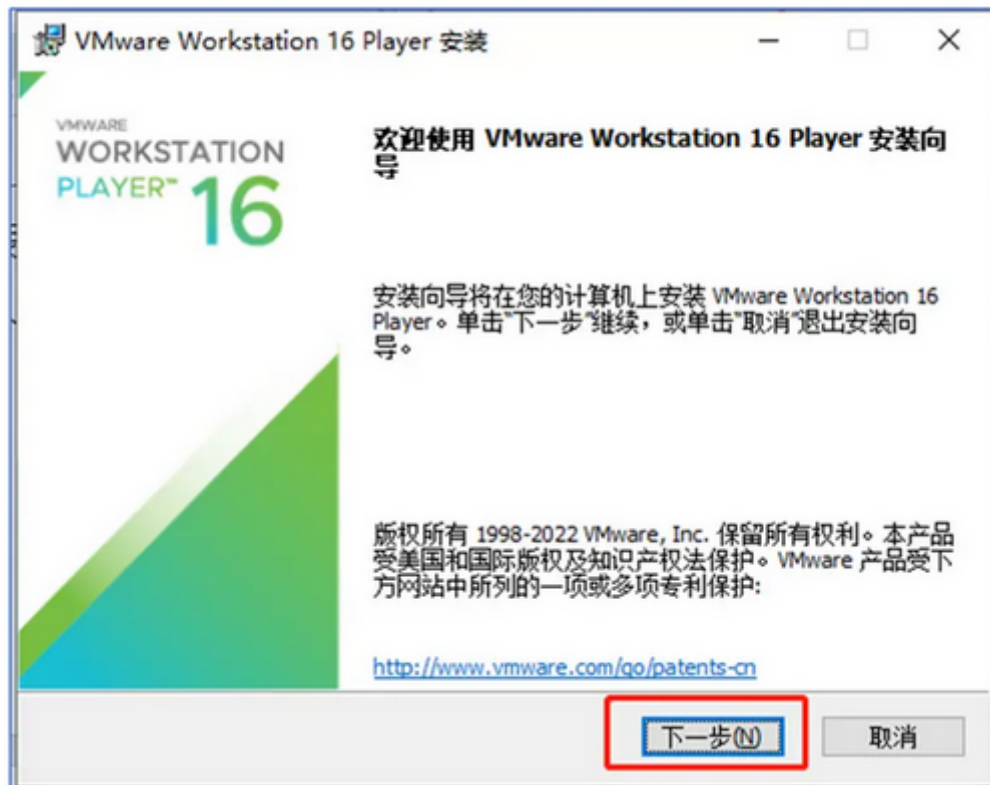
在vmware官方网站下载免费版本的[vmware.player](#):

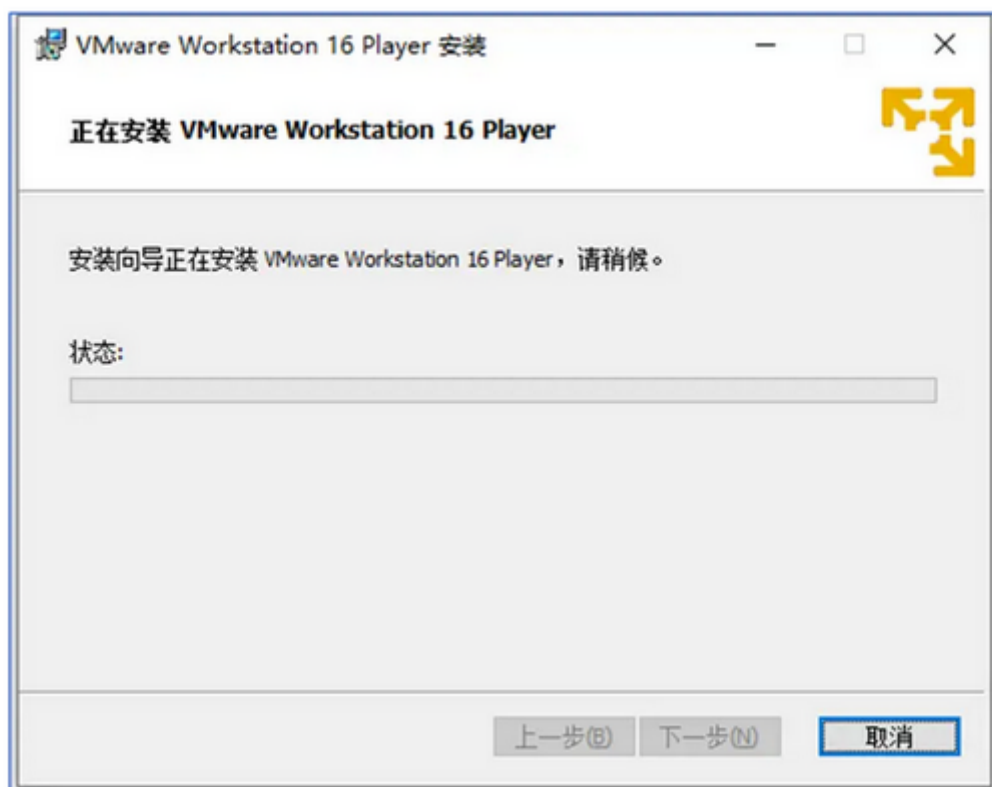
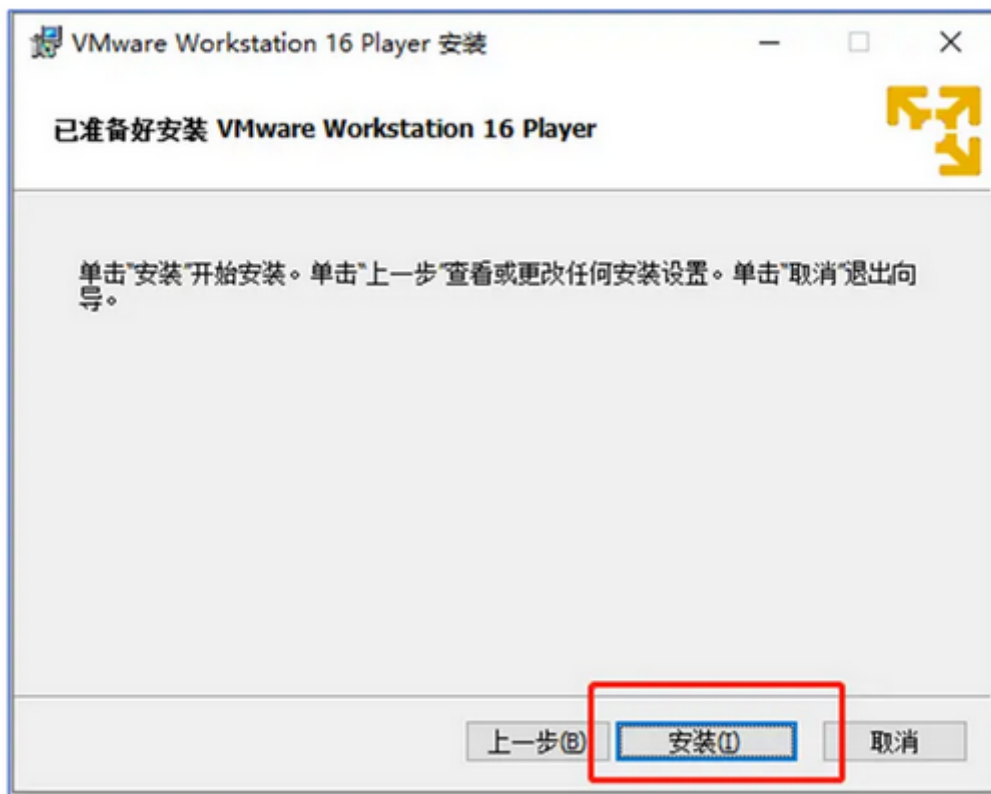


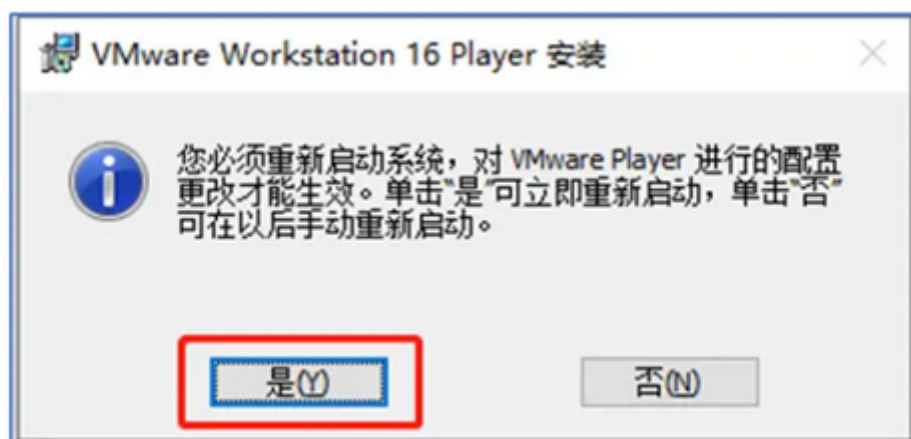
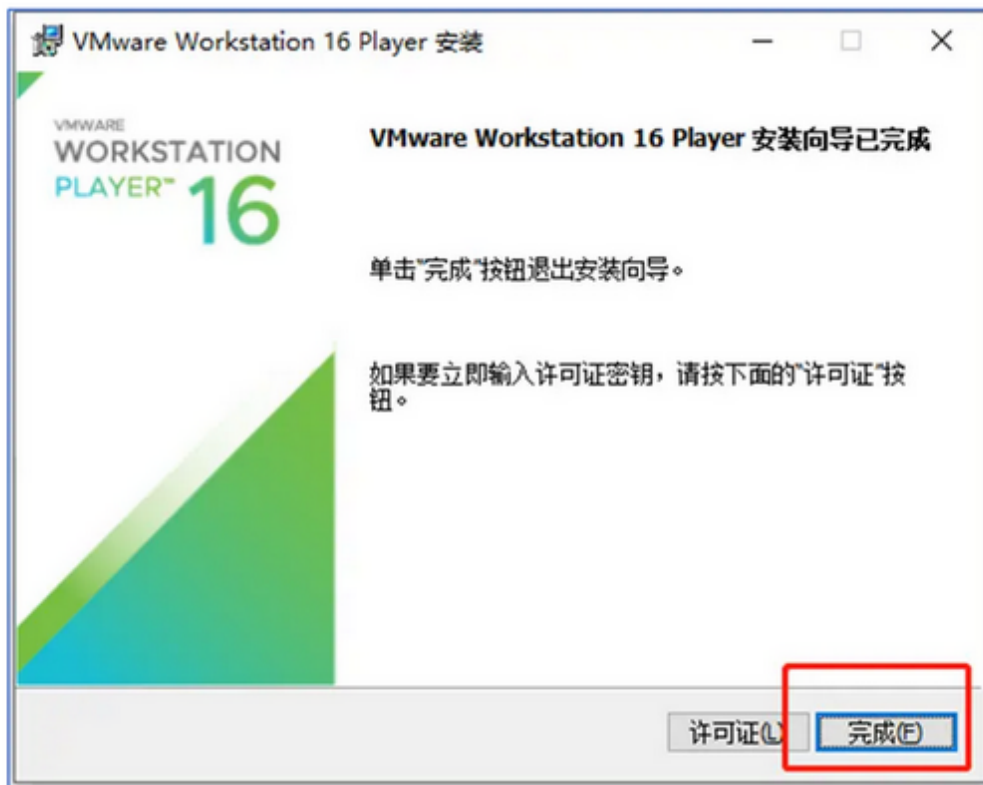
注意：版本可能会随时间更新，不影响使用

1.2 安装

1. 打开安装程序，按照向导开始安装：







2. 安装完成后重启系统。

2 Ubuntu系统下载安装

2.1 下载

到ubuntu官网下载[Ubuntu20.04 Desktop](#)，如下图：

Download Ubuntu Desktop

Ubuntu 20.04.4 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

- ✓ 2 GHz dual core processor or better
- ✓ 4 GB system memory
- ✓ 25 GB of free hard drive space
- ✓ Internet access is helpful
- ✓ Either a DVD drive or a USB port for the installer media



Download

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases [see our alternative downloads](#).

2.2 安装

1. 打开vmware player，新建一个linux虚拟机，如下图：



欢迎使用新建虚拟机向导

虚拟机如同物理机，需要操作系统。您将如何安装客户机操作系统？

安装来源：

☐ 安装程序光盘(D):

无可可用驱动器

点击浏览，选择下载的ISO文件

☒ 安装程序光盘映像文件(iso)(M):

ork\tools\download\ubuntu-20.04.3-desktop-amd64.iso

浏览(R)...

 已检测到 Ubuntu 64 位 20.04.3。
该操作系统将使用简易安装。([这是什么?](#))

☐ 稍后安装操作系统(S)。

创建的虚拟机将包含一个空白硬盘。

帮助

< 上一步(B)

下一步(N) >

取消

简易安装信息

这用于安装 Ubuntu 64 位。

个性化 Linux

全名(F):

ubuntu20.04

用户名(U):

密码(P):

确认(C):

帮助

< 上一步(B)

下一步(N) >

取消

新建虚拟机向导

×

命名虚拟机

您希望该虚拟机使用什么名称？

虚拟机名称(V):

Ubuntu20.04

位置(L):

C:\work\vm\ubuntu20.04

浏览(R)...

点击浏览选择虚拟机文件存放位置

< 上一步(B)

下一步(N) >

取消

指定磁盘容量

磁盘大小为多少?

虚拟机的硬盘作为一个或多个文件存储在主机的物理磁盘中。这些文件最初很小，随着您向虚拟机中添加应用程序、文件和数据而逐渐变大。

最大磁盘大小 (GB)(S):

针对 Ubuntu 64 位 的建议大小: 20 GB

☐ 将虚拟磁盘存储为单个文件(O)☒ 将虚拟磁盘拆分成多个文件(M)

拆分磁盘后，可以更轻松地在计算机之间移动虚拟机，但可能会降低大容量磁盘的性能。

设置ubuntu系统磁盘大小，在硬盘空间足够的情况下，建议尽量设置大一些

帮助

< 上一步(B)

下一步(N) >

取消

已准备好创建虚拟机

单击“完成”创建虚拟机，并开始安装 Ubuntu 64 位 和 VMware Tools。

将使用下列设置创建虚拟机：

名称：	Ubuntu20.04
位置：	C:\work\vm\ubuntu20.04
版本：	Workstation 16.x
操作系统：	Ubuntu 64 位
硬盘：	60 GB, 拆分
内存：	4096 MB
网络适配器：	NAT
其他设备：	2 个 CPU 内核, CD/DVD, USB 控制器, 打印机, 声卡

自定义硬件(C)...

☒ 创建后开启此虚拟机(P)

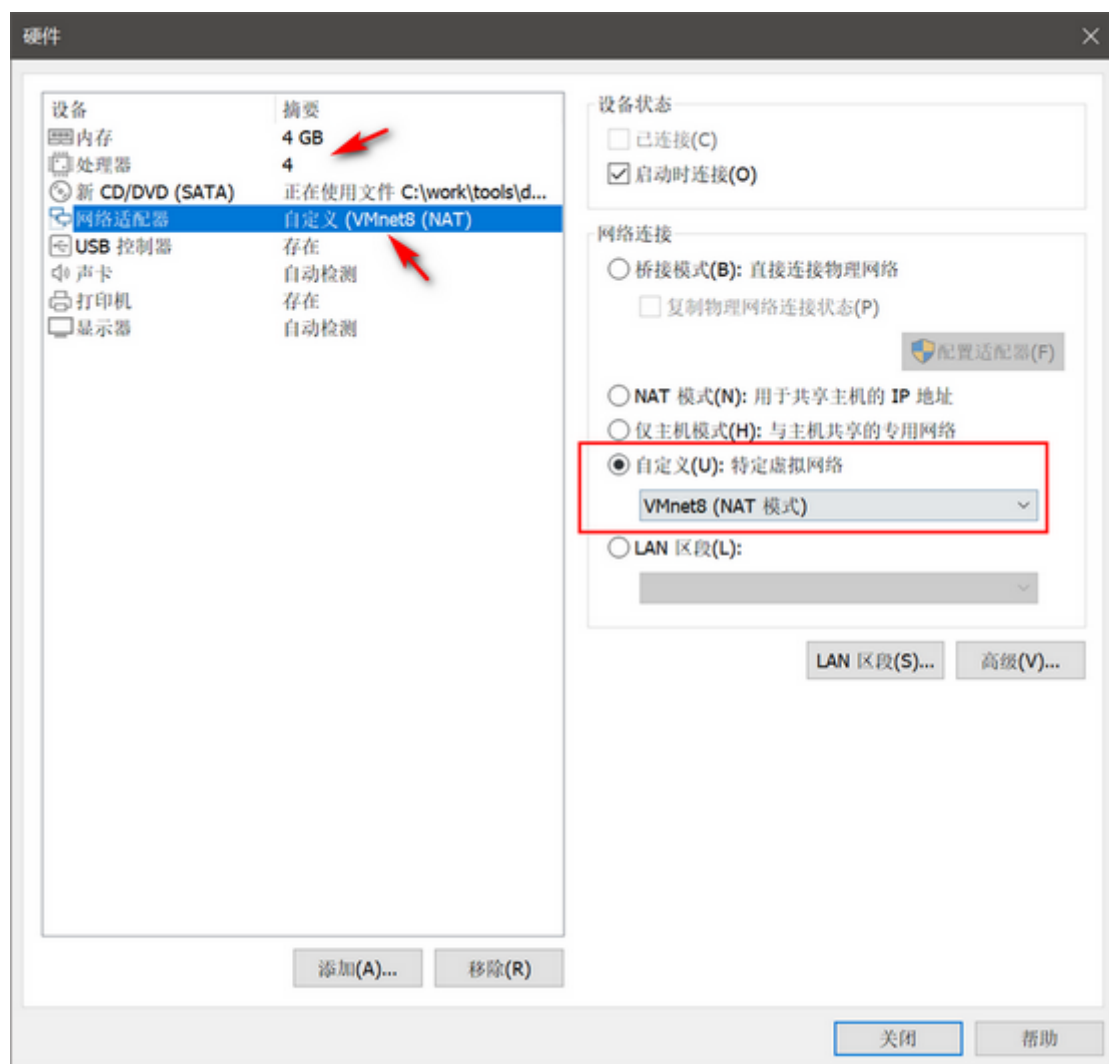
自定义硬件，根据主机的实际硬件情况配置

勾选后，点击完成开始安装

< 上一步(B)

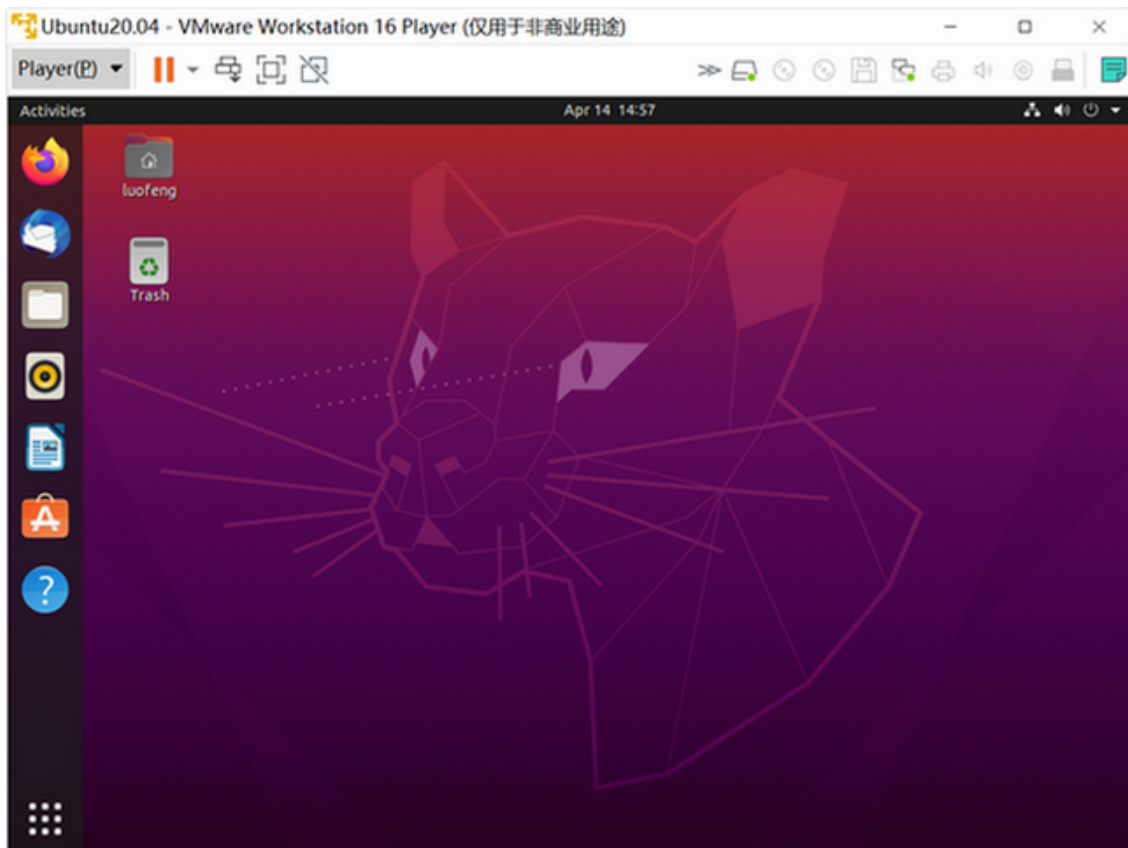
完成

取消



2. 完成虚拟机创建后，点击 完成 开始安装Ubuntu系统：



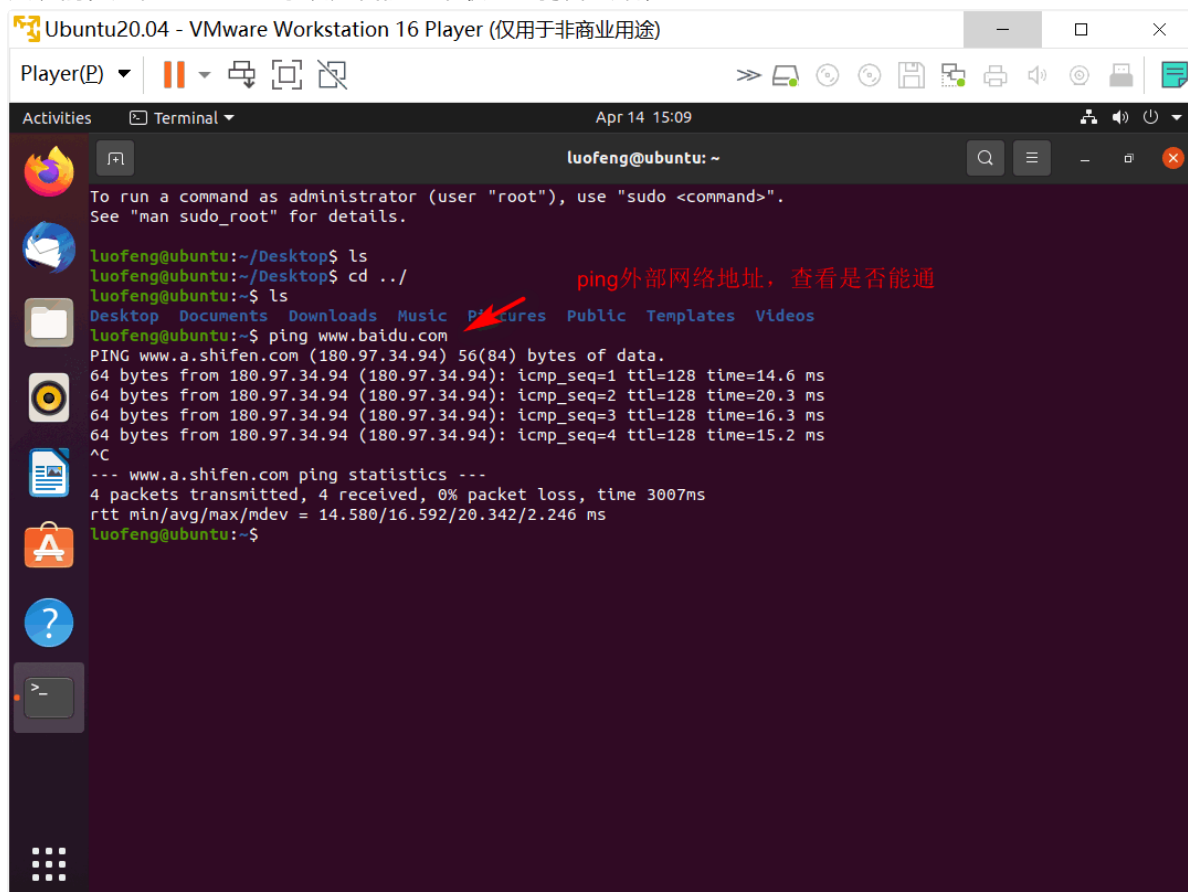


3 软件工具下载安装

3.1 Ubuntu工具

ubuntu环境主要负责软件编译，因此在该环境下需要在线安装开发环境依赖软件。

安装前，先检查ubuntu系统是否能正常联网。打开终端，



如果网络不通，可以上网查找如何配置虚拟机网络，寻找解决方案。

1. 网络正常的前提下，执行以下命令，安装开发环境依赖软件：

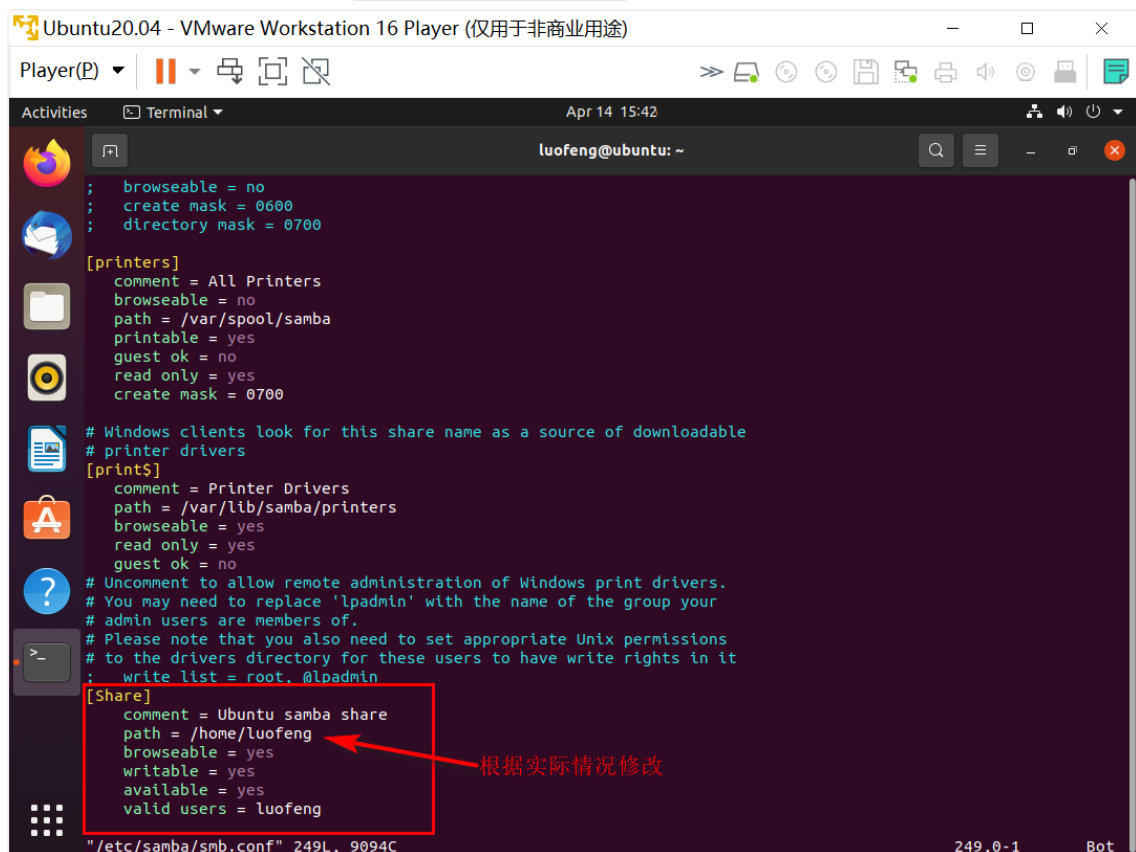
```
cd ~
sudo apt install build-essential python3 python3-tk qtbase5-dev
sudo apt install libc:i386 libstdc++6:i386 zlib1g:i386
sudo apt install net-tools
sudo wget http://security.ubuntu.com/ubuntu/pool/main/i/icu/libicu55_55.1-7_amd64.deb
sudo dpkg -i libicu55_55.1-7_amd64.deb
```

2. 安装配置samba共享环境，方便windows主机与ubuntu虚拟机之间共享访问SDK工程，执行以下命令：

```
sudo apt install samba samba-common
sudo smbpasswd -a <username>
```

<usrname> 填写自己安装ubuntu系统时使用的用户名。

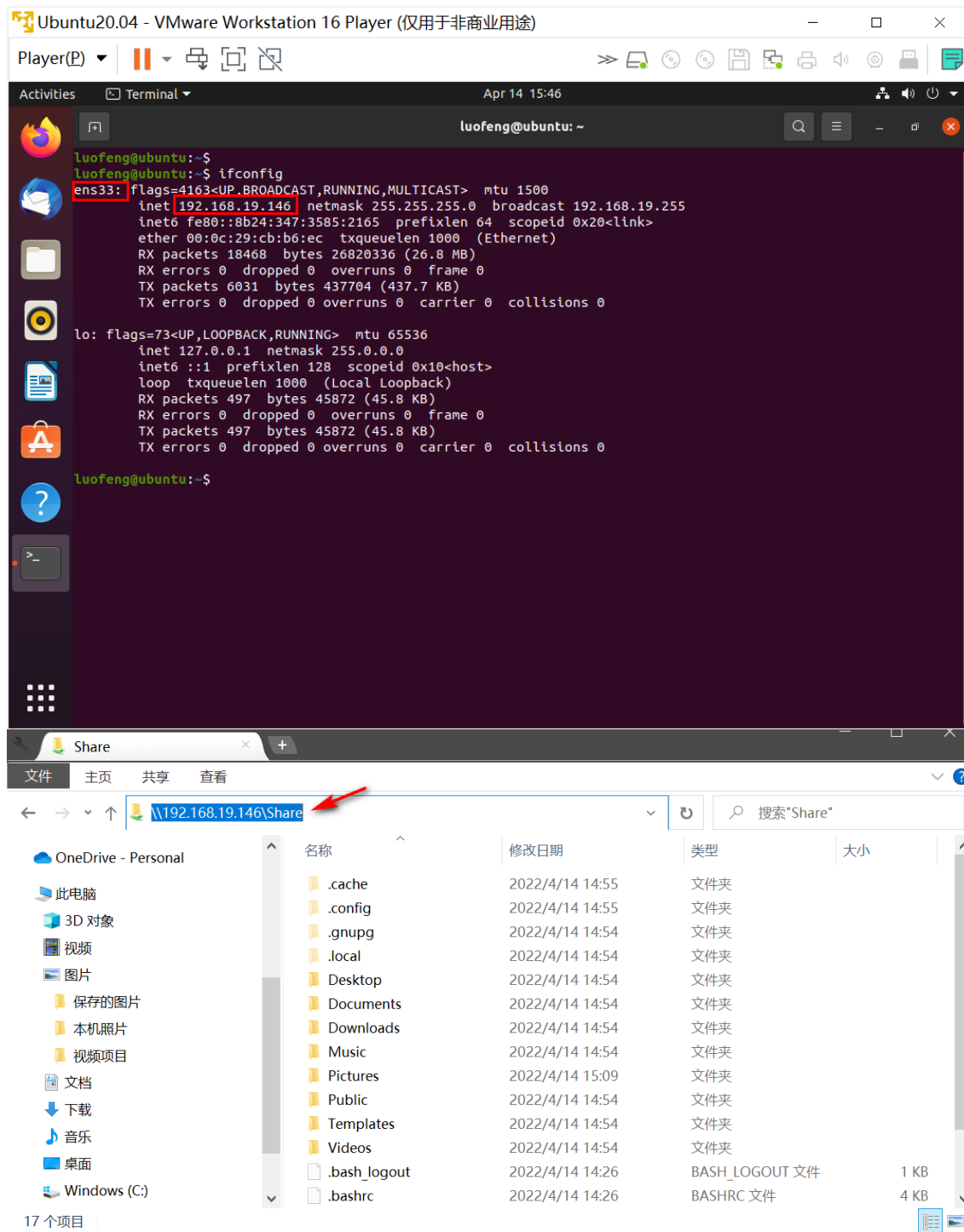
samba服务安装完成后，修改 /etc/samba/smb.conf，添加共享目录：



执行以下命令重启samba服务：

```
sudo service smbd restart
```

查看Ubuntu系统IP，并在Windows下使用samba访问共享目录



如果Windows下无法访问共享目录，可上网查找如何配置Ubuntu samba共享

3. 安装并启动SSH服务，执行以下命令：

```
sudo apt install openssh-server
sudo ufw allow ssh
```

3.2 Windows工具

参考文档：FIBOCOM MA510-GL-00-90 Series FibocomOpen Application Guide_V1.0.1.pdf

日志工具：<https://drive.weixin.qq.com/s?k=AGQAugfWAAkODaGQn9>

驱动下载：<https://drive.weixin.qq.com/s?k=AGQAugfWAAk9Y9moGe>

APP烧录工具(QPST)：[https://drive.weixin.qq.com/s?k=AGQAugfWAAkoFcV83s#/#](https://drive.weixin.qq.com/s?k=AGQAugfWAAkoFcV83s#/)

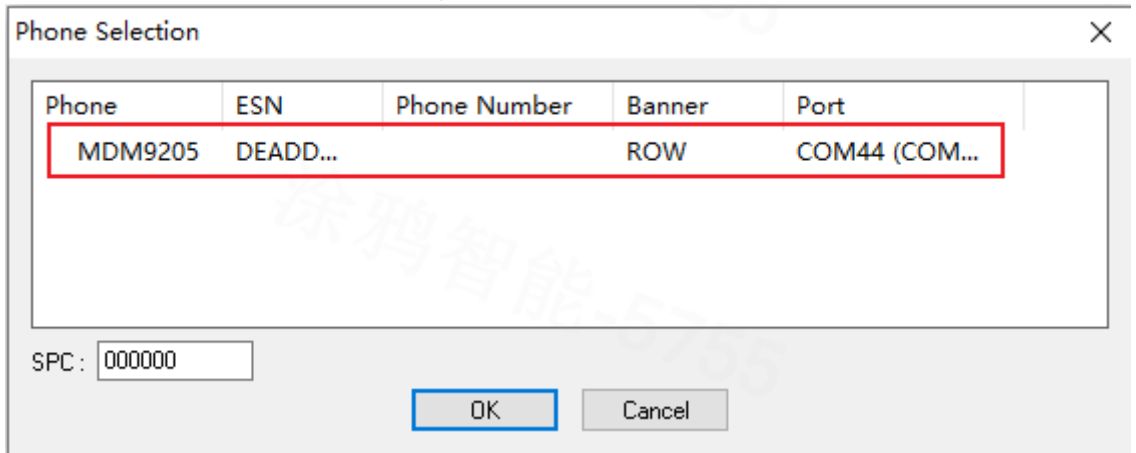
3.2.1 驱动安装

1. 驱动安装，一直点下一步即可
2. 安装完成后，即可在设备管理器看到模块端口：

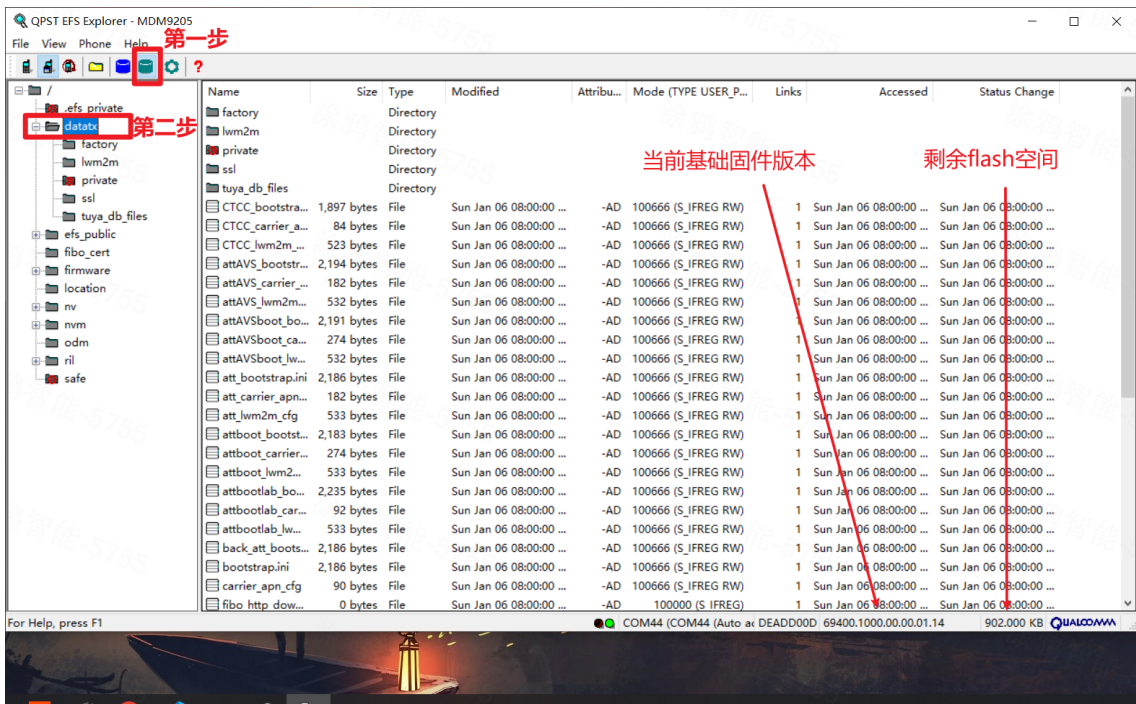


3.2.2 APP烧录工具(QPST)使用说明

1. 设备连上USB之后，打开QPST EFS Explorer.exe可以看到设备信息



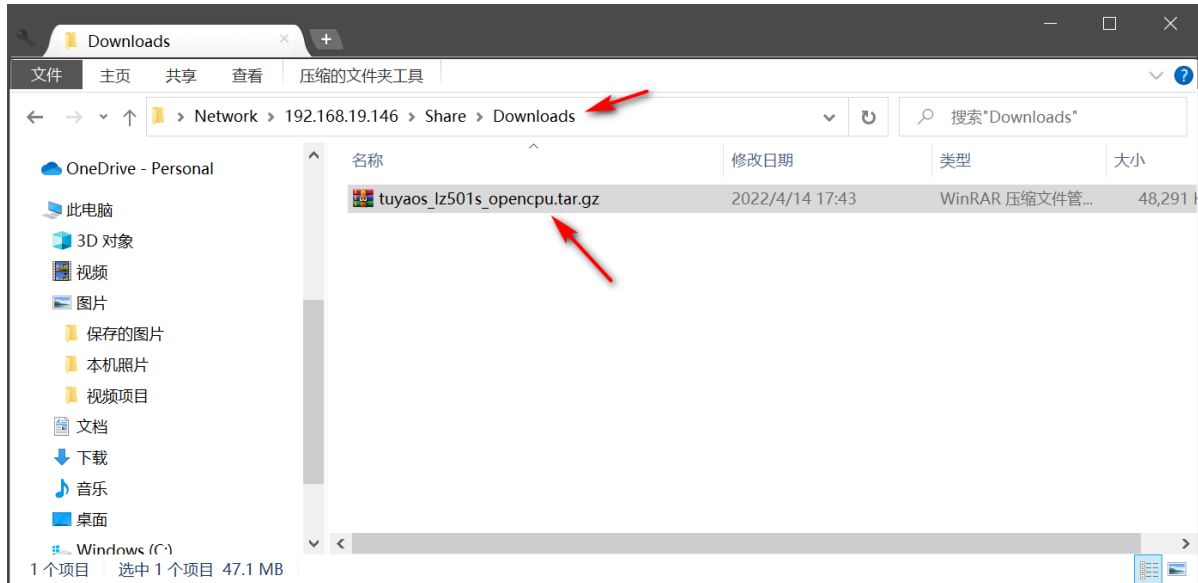
2. 选中设备后可以看到模块文件系统
3. 按下列步骤将应用程序.bin文件及启动ini文件拖入datatx文件即完成烧录



4 开发板及SDK获取

4.1 SDK获取

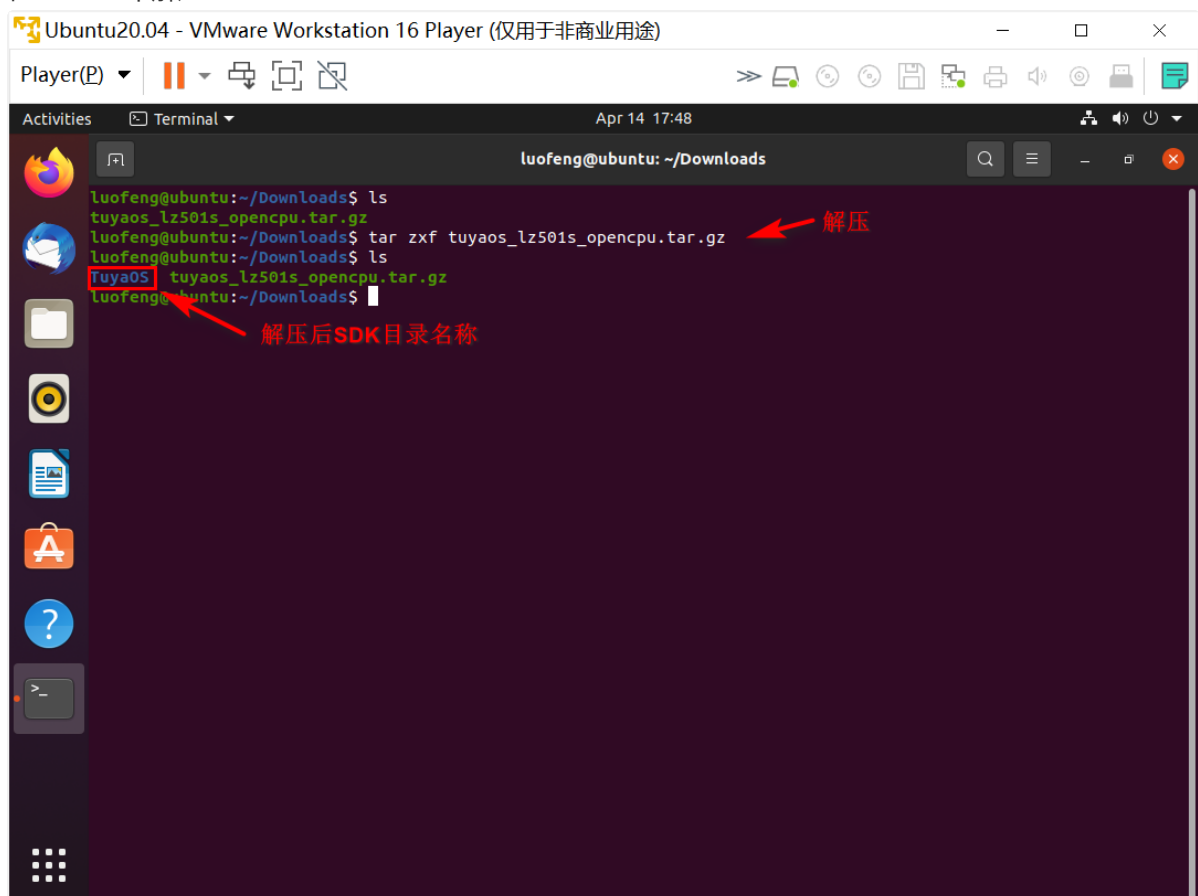
在涂鸦开发者平台或者通过涂鸦IDE工具，完成正常的开发者账号和产品创建流程后，即可获取到模组SDK：



图中下载的是LZ501S模组的SDK压缩包，包名称为临时名称，下载后放到Ubuntu共享目录下

各型号模组最终发布的SDK包名称稍有差异，不能以上图作为参考

在Ubuntu中解压SDK：



5 程序开发及编译

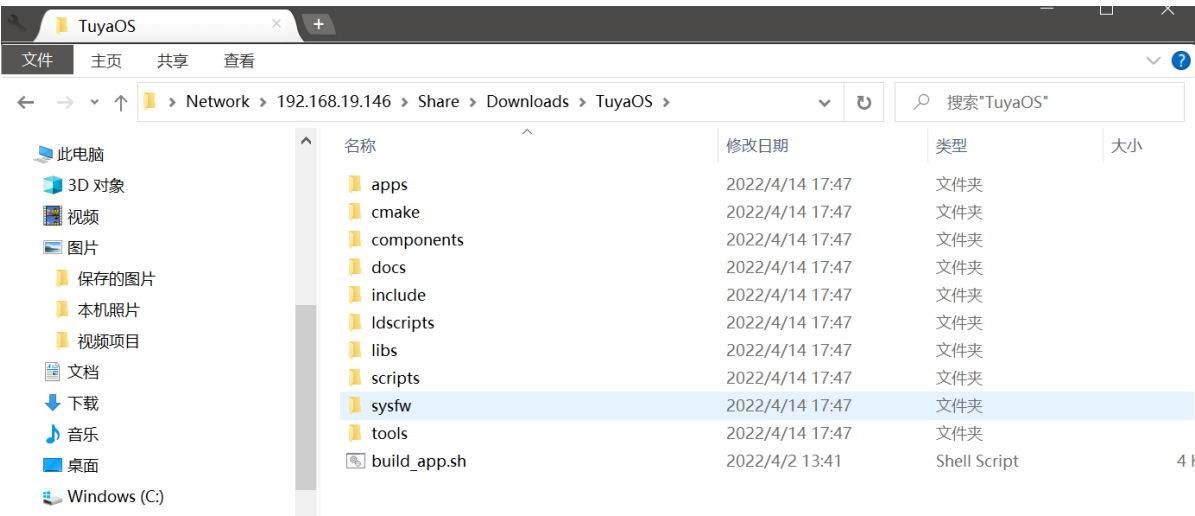
5.1 程序开发

用户开始应用编程之前，需要了解各型号模组提供给用户使用的Flash空间情况，当应用程序超出Flash空间限制时，将编译失败。各型号模组提供给用户的Flash空间大小如下：

模组型号	应用程序Flash空间大小
MA510GL	

5.1.1 SDK说明

SDK包解压完成后，进入TuyaOS目录，目录结构如下：

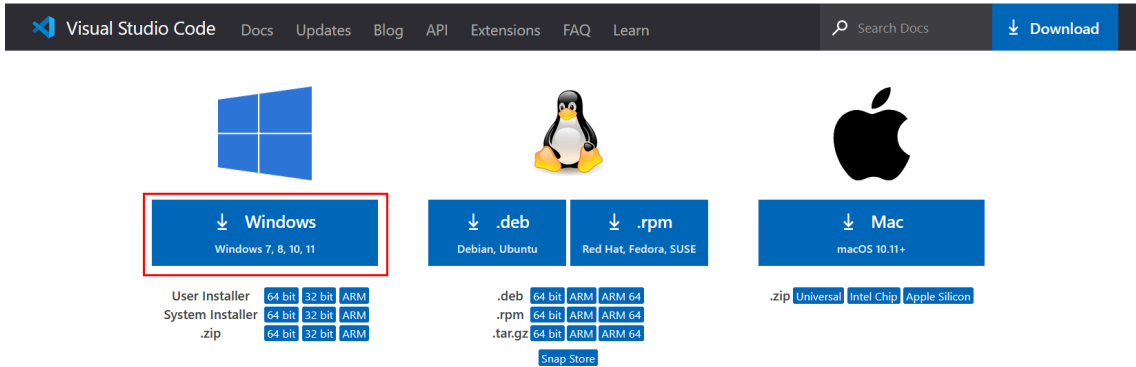


目录	说明
apps	应用工程目录，所有应用程序代码工程放在该目录下
cmake	cmake基础脚本目录
components	系统库及头文件目录
docs	帮助文档目录
include	涂鸦头文件目录
ldscripts	编译链接脚本目录
libs	涂鸦SDK开发库目录
scripts	通用shell脚本目录
sysfw	系统固件目录
tools	SDK工具目录
build_app.sh	工程编译启动脚本

5.1.2 IDE开发环境安装

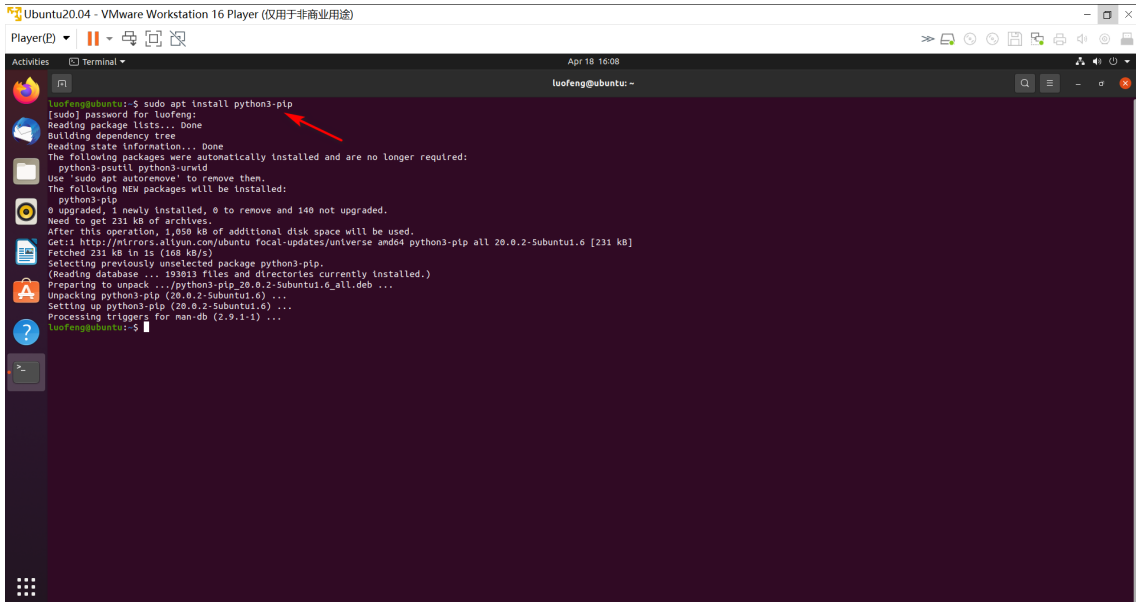
- SDK解压后，用户可在 `apps` 目录下建立自己的应用开发工程。SDK发布时，在该目录下集成了若干demo工程，用户可以参考demo工程的样式创建新的应用工程。
- 完成应用工程创建之后，用户可以使用自己喜欢的代码编辑工具（如vscode, source insight, vim, sublime text, atom等），完成应用工程的代码开发。除此之外，为方便用户在涂鸦SDK工程中开发应用，涂鸦也为用户提供了一套专用的IDE工具，下面介绍windows系统中IDE工具的安装及使用。

1. 下载并安装 [vscode studio](#)



2. Ubuntu下安装python3.8及pip3

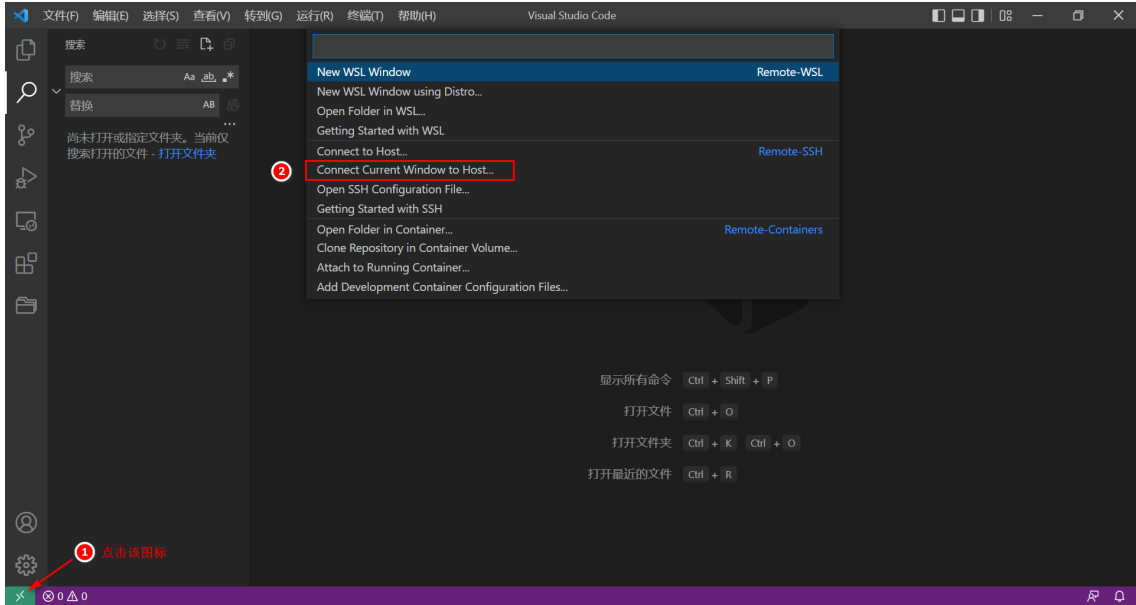
```
sudo apt install python3.8
sudo apt install python3-pip
```



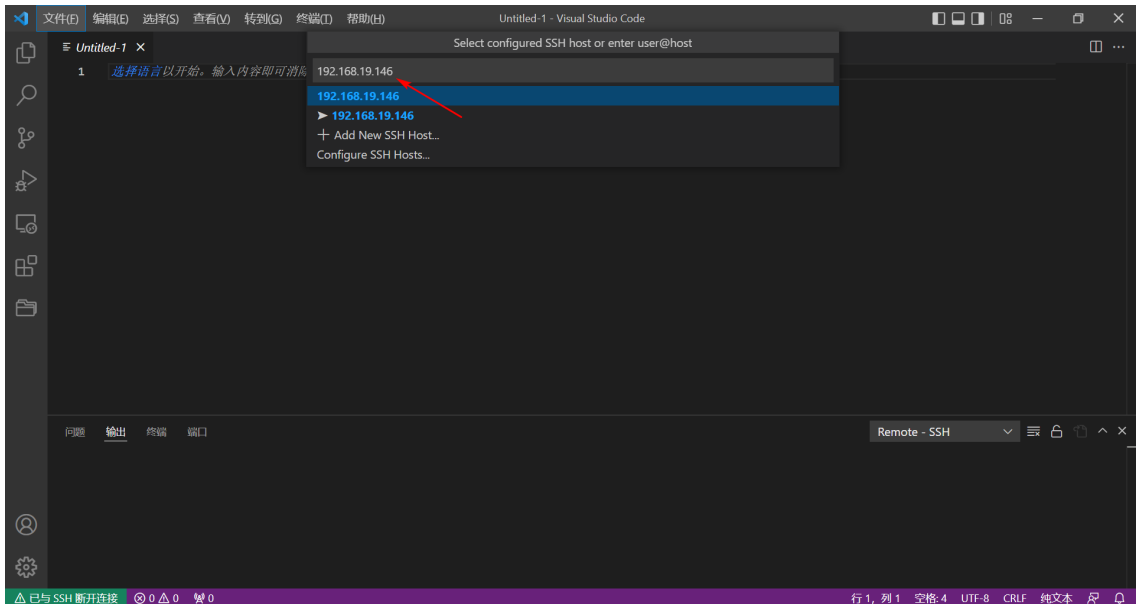
3. 重启电脑，打开 vscode studio，安装 Remote SSH 插件



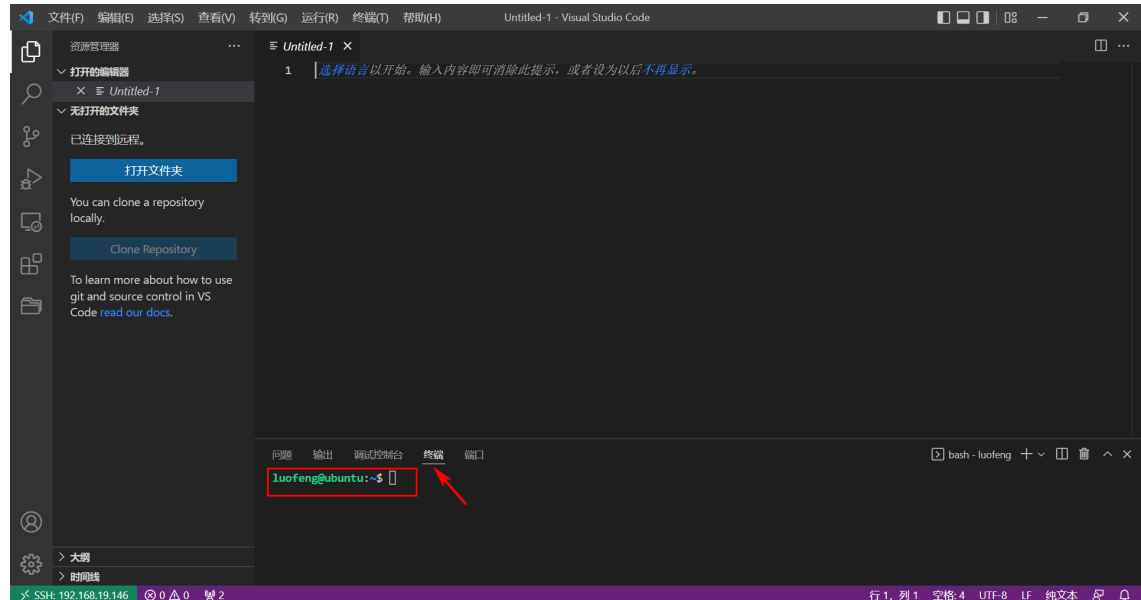
安装完成后，开始SSH连接Ubuntu



输入Ubuntu系统IP地址，开始连接，并根据提示输入用户名和密码

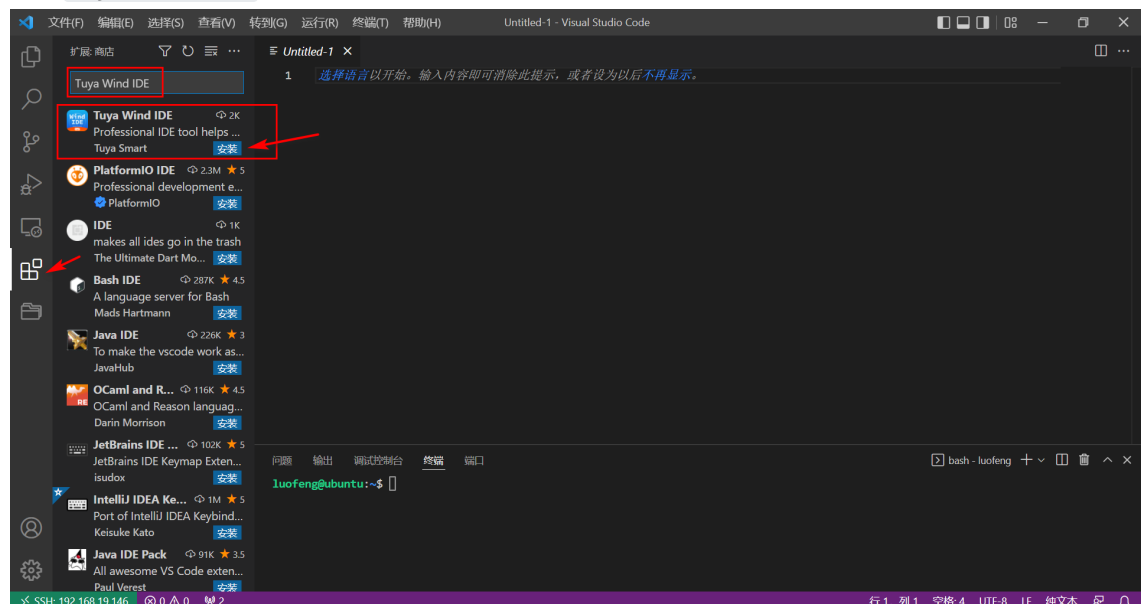


连接成功后，在终端下能看到Ubuntu命令行

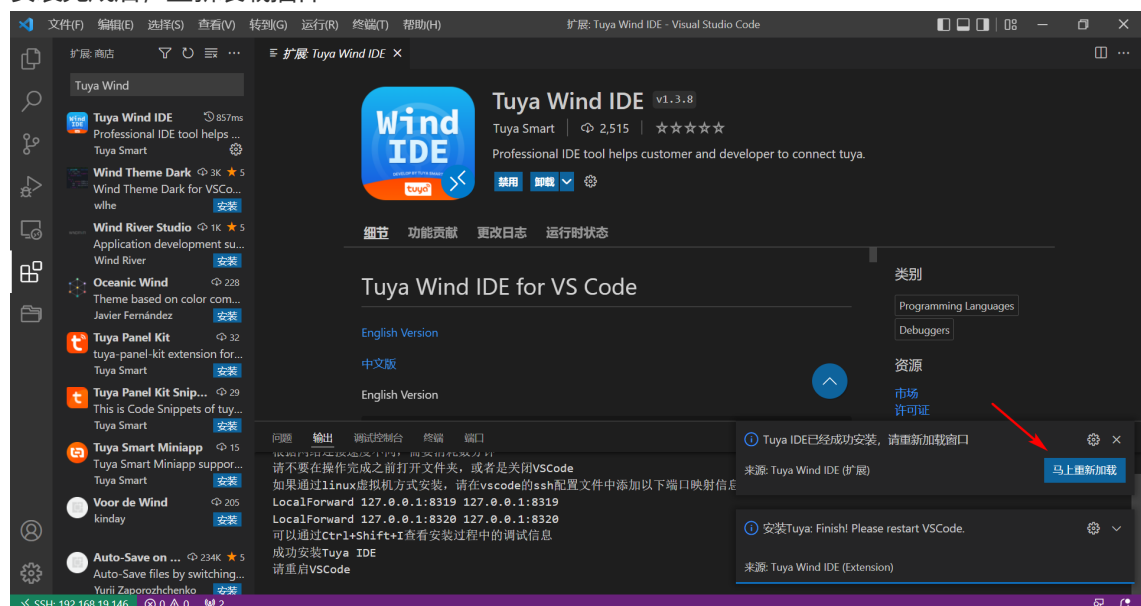


注意：如果SSH连接失败，可上网查找如何配置vscode SSH

4. 安装Tuya wind IDE 插件

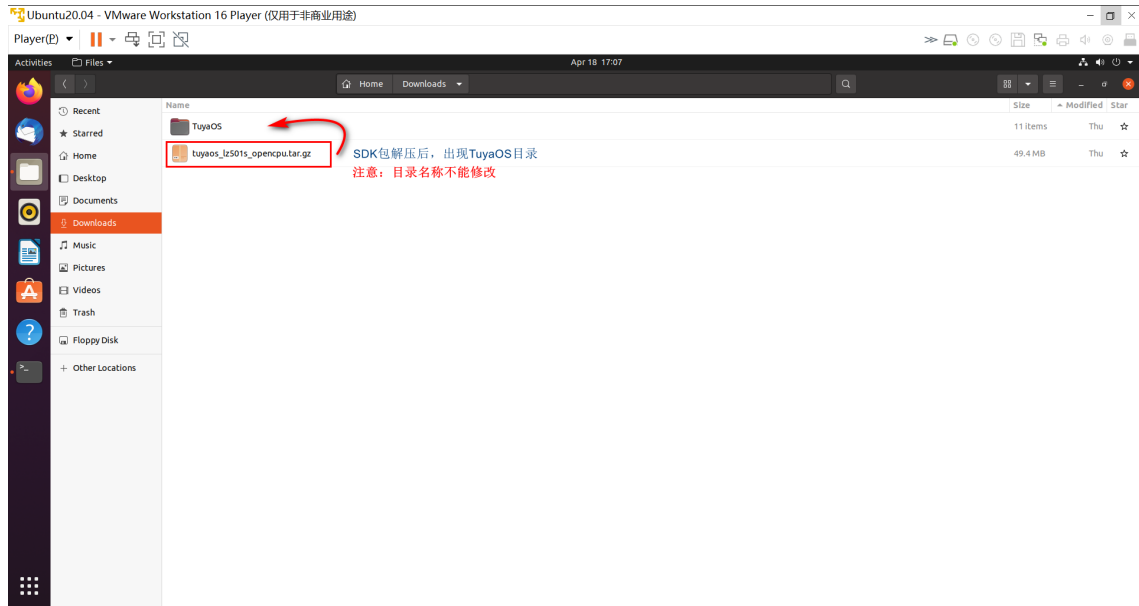


安装完成后，重新装载插件

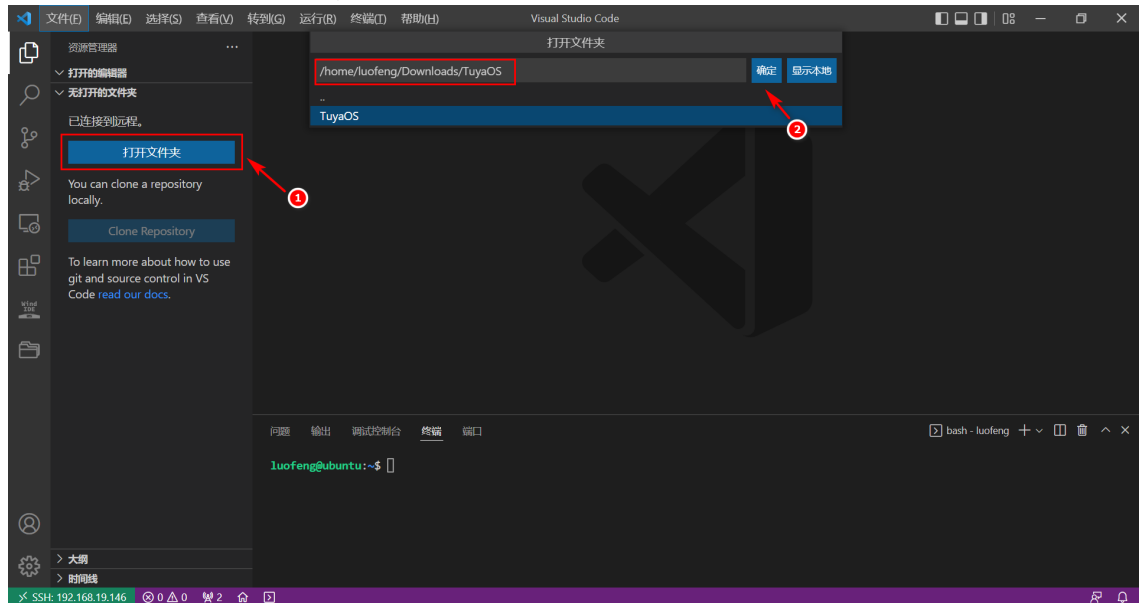


5.1.3 IDE开发环境使用

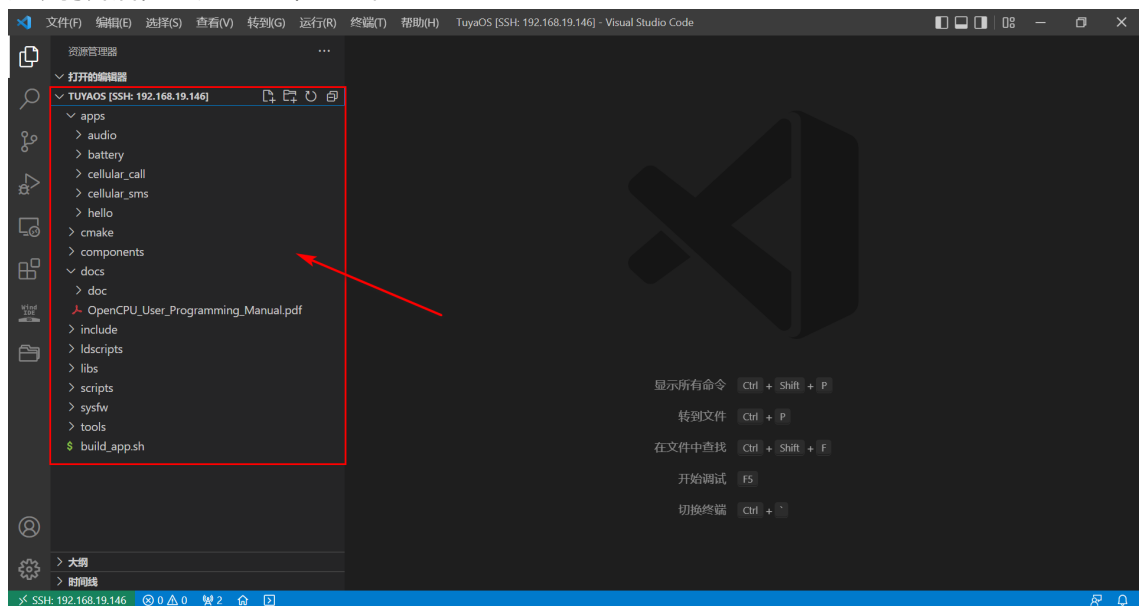
1. IDE安装完成后，先在Ubuntu系统下将获取的SDK压缩包解压



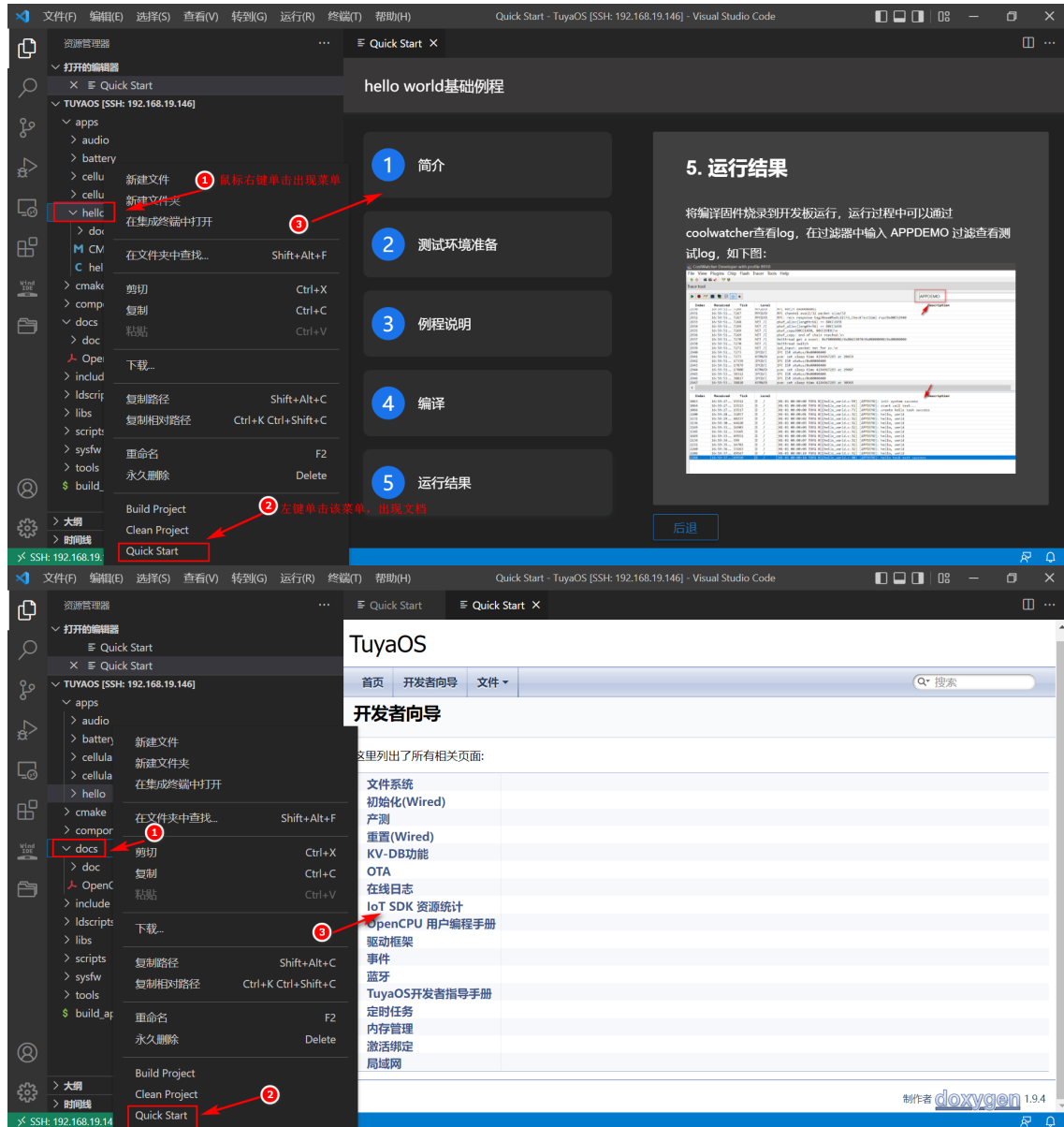
2. 打开vscode studio，选择TuyaOS



3. 成功打开后，出现SDK工程目录树



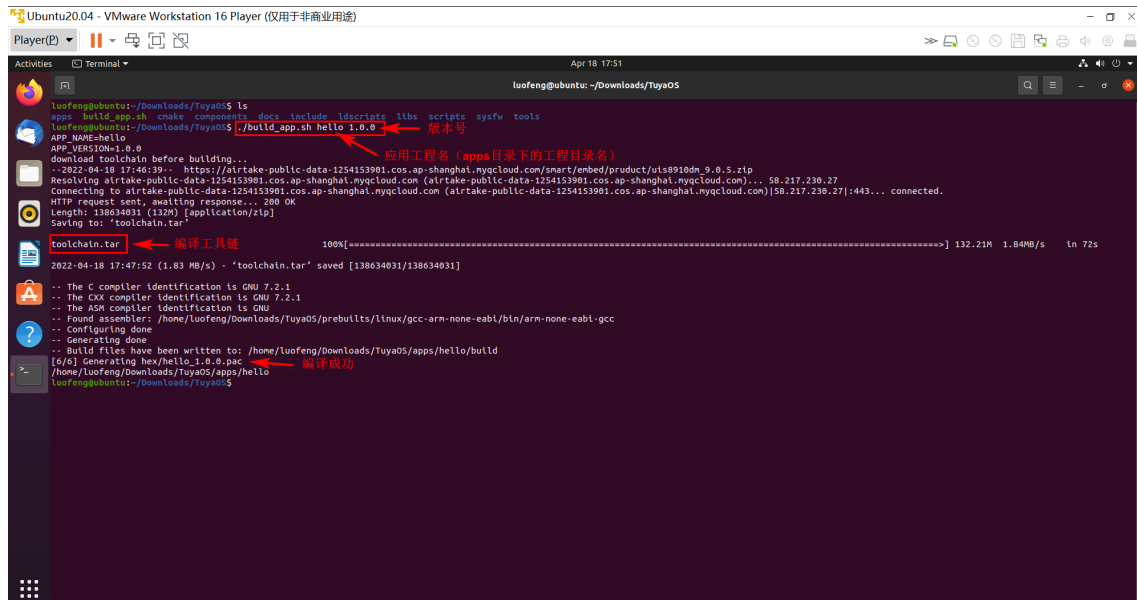
4. 通过菜单查看 demo 帮助文档，以及SDK手册



5.2 编译

支持两种编译方式:

1. Ubuntu系统命令行编译



第一次编译应用之前，会自动下载编译工具链并集成到SDK中。

2. Tuya Wind IDE菜单编译

The first screenshot shows the Tuya Wind IDE interface. The 'Build Project' menu option is highlighted in the 'Build' menu. A red arrow points to the 'Build Project' option with the text '鼠标单击出现菜单' (Click the mouse to show the menu). Another red arrow points to the 'Build Project' option with the text '单击选择该菜单' (Click to select this menu).

The second screenshot shows the version '1.0.0' being entered in the version input field. A red arrow points to the '1.0.0' text with the text '输入版本号' (Enter version number). Below the input field, the text '请输入应用的版本，版本格式为 x.x.x (按 "Enter" 以确认或按 "Esc" 以取消)' (Please enter the application version, version format is x.x.x (press "Enter" to confirm or press "Esc" to cancel)) is displayed.

The third screenshot shows the compilation output in the terminal. The output includes the following lines:

```
luofeng@ubuntu:~/Downloads/TuyaOS$ cd /home/luofeng/Downloads/TuyaOS/
luofeng@ubuntu:~/Downloads/TuyaOS$ export TUYA_IDE_PYTHON_ENV=/home/luofeng/.tuya/venv/bin/python
luofeng@ubuntu:~/Downloads/TuyaOS$ cd /home/luofeng/Downloads/TuyaOS/
luofeng@ubuntu:~/Downloads/TuyaOS$ ./build_app.sh apps/hello hello 1.0.0
APP_NAME=hello
APP_VERSION=1.0.0
-- The C compiler identification is GNU 7.2.1
-- The CXX compiler identification is GNU 7.2.1
-- The ASM compiler identification is GNU
-- Found assembler: /home/luofeng/Downloads/TuyaOS/prebuilts/linux/gcc-arm-none-eabi/bin/arm-none-eabi-gcc
-- Configuring done
-- Generating done
-- Build files have been written to: /home/luofeng/Downloads/TuyaOS/apps/hello/build
[6/6] Generating hex/hello.1.0.0.pac
luofeng@ubuntu:~/Downloads/TuyaOS$
```

A red arrow points to the line '开始自动编译' (Start automatic compilation) which is the first line of the output. Another red arrow points to the line '编译结果' (Compilation result) which is the last line of the output.