

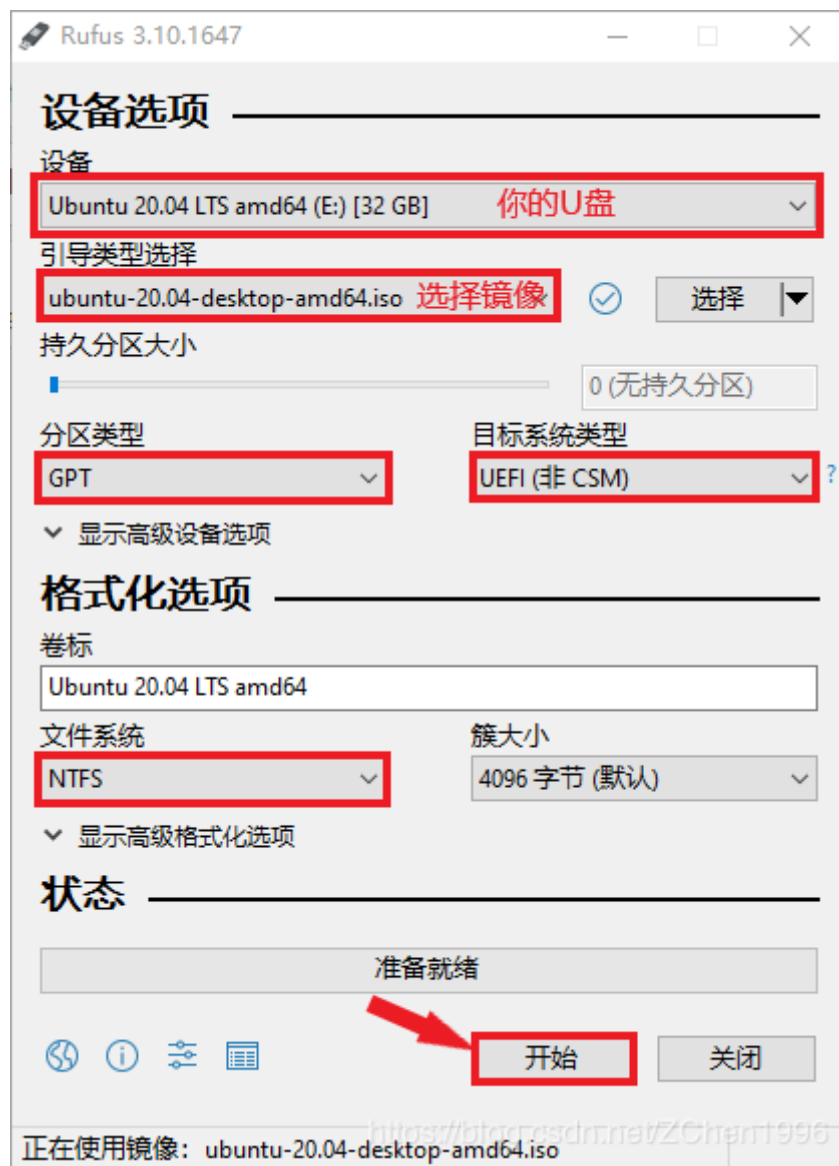
# Ubuntu20.04 Robocup教程

## 1.Ubuntu20.04安装教程【针对已经安装过Ubuntu的用户】

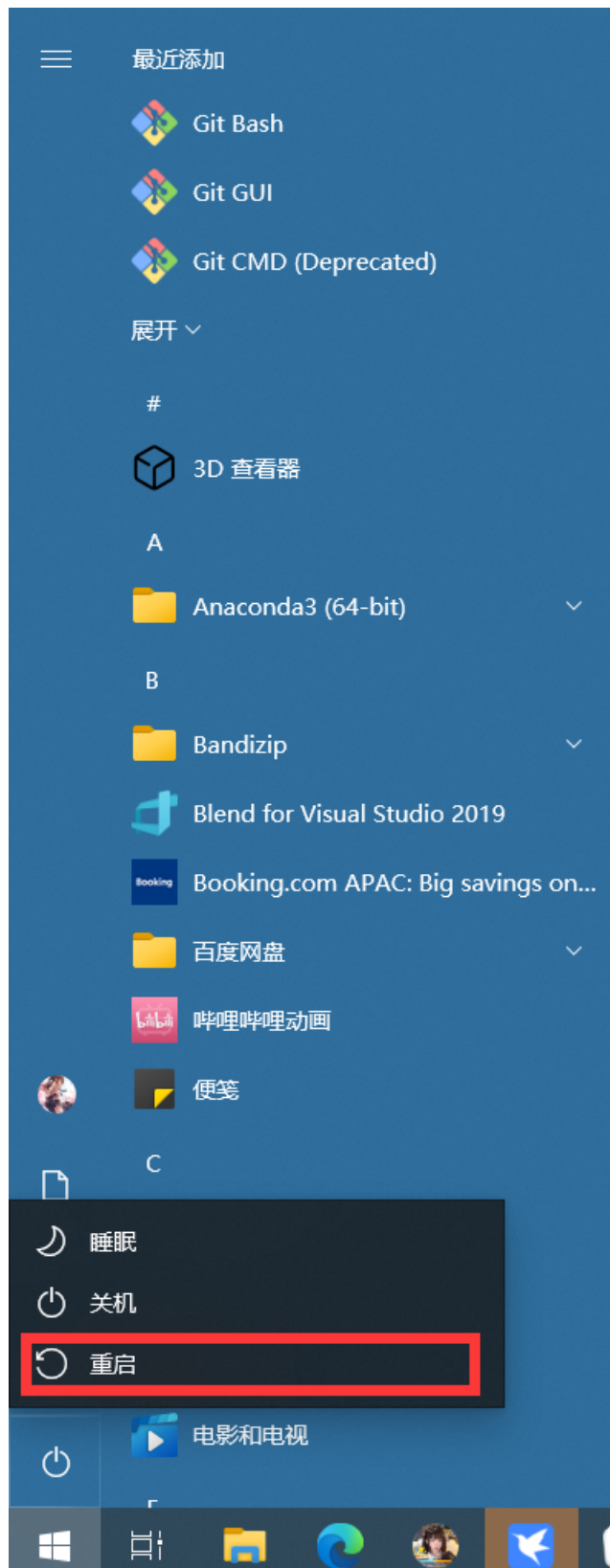
(1) 下载桌面图像 [ubuntu.com](https://releases.ubuntu.com/20.04/) 点击进入



(2) Rufus烧录文件 (exe见打包文件夹)



(3) 按住SHIFT键鼠标点击重启，进入U盘启动界面



(4) 走完一系列Ubuntu的基本配置即可（不多赘述）

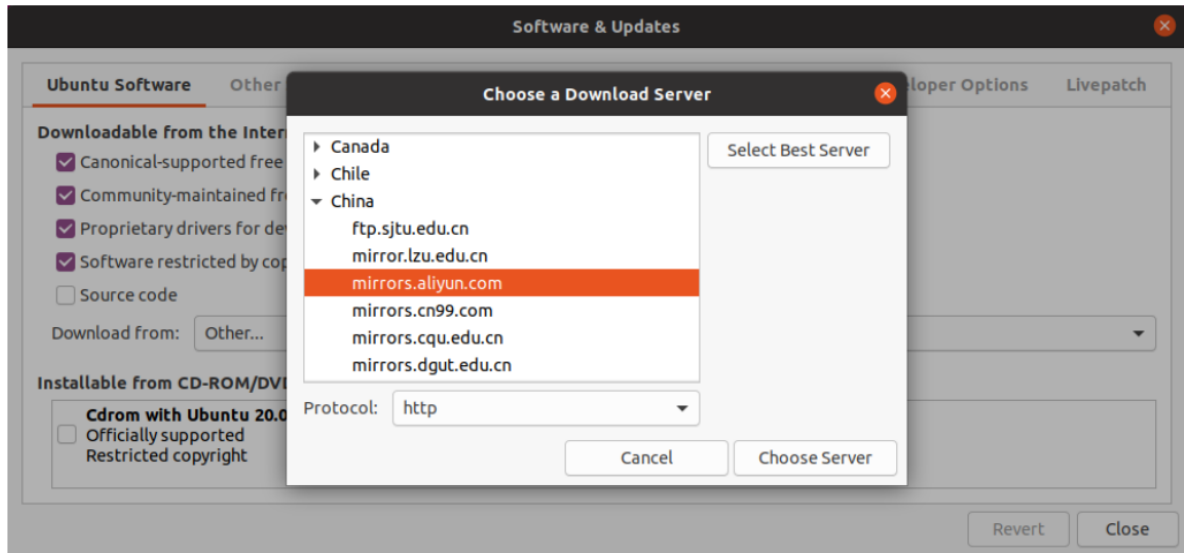
## 2.Ros Noetic安装【请记牢这个版本的名字】

(1) 设置ros源，让每次sudo apt-get update都找到这个仓库

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

(2) 切换系统源，采用国内的镜像网站下载，速度会快很多

打开软件与更新，选择【中国的服务器】，确认后【自动】打开这个界面，建议选择【清华/阿里】



### (3) 添加ROS密钥

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

### (4) 更新源【至此，系统可以找到ROS的所有安装包】

```
$ sudo apt update
```

### (5) 安装ROS包

```
$ sudo apt-get install ros-noetic-desktop-full  
$ sudo apt-get install ros-noetic-rqt*
```

### (6) 初始化rosdep

```
$ sudo rosdep init  
$ rosdep update
```

这个地方100%报错，解决方法如下：

-----问题1：无法连接到某个网络位置-----

【1.克隆仓库】【注意：在非中文路径处克隆】

```
$ sudo apt-get install git 【如果你已经下载了git就不用执行这一步】  
$ git clone https://github.com/ros/rosdistro.git
```

【2.更改文件】【文件一共6个，还有两个在rosdistro但是我忘记了】

下面的文件中，`https://raw.githubusercontent.com/ros/rosdistro/master` 字段，全部替换成【file:// + 仓库的绝对位置】

- `/etc/ros/rosdep/sources.list.d/20-default.list`
- `/usr/lib/python2.7/dist-packages/rosdep2/gbpdistro_support.py`
- `/usr/lib/python2.7/dist-packages/rosdep2/rep3.py`
- `/usr/lib/python2.7/dist-packages/rosdistro/__init__.py`

-----问题2: sudo rosdep init没有这个指令-----

```
$ rosdep init
```

会提示需要安装一个包，安装即可

(7) 安装其他依赖

```
$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator  
python-wstool build-essential
```

(8) 创建ros工作区以及编译

```
$ mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/src  
$ catkin_init_workspace  
  
$ cd ~/catkin_ws/  
$ catkin_make
```

(8) 配置环境变量

```
$ sudo apt-get install net-tools  
$ gedit ~/.bashrc
```

这个操作会打开一个文件，在文件下方输入下面的内容

```
# Set ROS noetic 【source命令可以让系统找到这些包】  
source /opt/ros/noetic/setup.bash  
source ~/catkin_ws/devel/setup.bash  
  
# Set ROS alias command 【快捷指令】  
alias cw='cd ~/catkin_ws'  
alias cs='cd ~/catkin_ws/src'  
alias cm='cd ~/catkin_ws && catkin_make'  
# 自定义一些常用的指令  
alias up2 = 'sudo apt-get update && sudo apt-get upgrade'
```

(9) 测试ros是否成功，打开三个不同的终端，每个终端一个指令

```
$ roscore  
$ rosrunc turtlesim turtlesim_node  
$ rosrunc turtlesim turtle_teleop_key
```

-----问题: sudo rosdep init时出现了问题2，然后roscore 或者roscrun 找不到指令-----

重新执行下面的命令

```
$ sudo apt-get install ros-noetic-desktop-full
```

## 3.安装Robocup项目所需的软件包

(1) 安装pytorch [点击进入\(pytorch.org\)](https://pytorch.org)

【查看系统显卡驱动】

```
$ nvidia-smi
```

【官网查看安装指令，cuda11.1以上都可以安装cuda11.1的pytorch】

PyTorch Build	Stable (1.9.0)		Preview (Nightly)	LTS (1.8.2)
Your OS	Linux		Mac	Windows
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.1	ROCm 4.2 (beta)	CPU
Run this Command:	pip3 install torch==1.9.0+cu111 torchvision==0.10.0+cu111 torchaudio==0.9.0 -f https://download.pytorch.org/whl/torch_stable.html			

(2) 物品识别第三方库安装：在 `\robocup\object_api` 目录下找到 `requirements.txt`，当前文件夹的终端

```
$ pip3 install -r requirements.txt
```

(3) 人脸识别第三方库安装：

```
$ pip3 install baidu-aip
```

(4) 运行 `main.py`，不出现错误即成功

## 4.安装ros项目的依赖包

(1) 将打包文件夹里面 `catkin_ws/src` 的内容复制粘贴到本地的目录中

(2) 安装依赖包

```
$ sudo apt install flex bison freeglut3-dev libbdd-dev python-catkin-tools  
ros-${ROS_DISTRO}-tf2-bullet  
$ sudo apt install ros-${ROS_DISTRO}-turtlebot3-gazebo ros-${ROS_DISTRO}-  
turtlebot3-navigation ros-${ROS_DISTRO}-move-base-msgs  
$ sudo apt-get install -y ros-${ROS_DISTRO}-navigation ros-${ROS_DISTRO}-teb-  
local-planner* ros-${ROS_DISTRO}-ros-control ros-${ROS_DISTRO}-ros-controllers  
ros-${ROS_DISTRO}-gazebo-ros-control ros-${ROS_DISTRO}-ackermann-msgs  
ros-${ROS_DISTRO}-serial qt4-default ros-${ROS_DISTRO}-effort-controllers  
ros-${ROS_DISTRO}-joint-state-controller ros-${ROS_DISTRO}-tf2-ros  
ros-${ROS_DISTRO}-tf
```

(3) 编译项目

```
$ cm 【这是之前的快捷指令】
```

-----问题1: gcc版本过高，报错提示一些cpp文件语法错误-----

-----问题2：依赖包不够，这个报错会很奇怪-----

【解决方法】首先把src里面的包，移除一部分到其他位置保存，相对于每次编译编一小部分，这样错误会变少，报错也明显很多

【需要亲自指导】