



## Rover X-R0S 环境说明及配置指南



编写	杨秋生、毛伟、肖俊杰
日期	2023 年 7 月
版本	V1.0, 第 1 次发布
审核	程军
批准	汪步云

芜湖云擎机器人科技有限公司

# 目录

1. 软硬件介绍.....	3
1.1. 工控机.....	3
1.2. Ubuntu 18.04.....	3
1.3. ROS Melodic .....	3
2. 环境配置.....	4
2.1. 配置虚拟机.....	4
2.1.1. 安装 Ubuntu 虚拟机 .....	4
2.1.2. 配置 ROS 环境.....	11
2.2. 配置工控机.....	16
2.2.1. 安装 Ubuntu 系统 .....	16
2.2.2. 配置 ROS 环境.....	18
2.3. 配置工控机与虚拟机的 ROS 通信 .....	18
2.3.1. 工控机配置.....	18
2.3.2. 虚拟机配置.....	19
2.3.3. 测试 ROS 通信 .....	21
2.3.4. 设置静态 IP .....	22

# 1. 软硬件介绍

## 1.1. 工控机

为了满足计算需求，Rover X 选用高性能工控机作为 SLAM 与定位导航的计算设备。Rover X 所使用的工控机搭载 Intel AlderLake N100 CPU，低功耗高性能，四核四线程，最大睿频 3.4GHz。工控机配备 8GB DDR5 内存以及 128GB SSD，同时配有双网口、4 个 USB 口、2 个视频输出接口(HDMI、DP)。

## 1.2. Ubuntu 18.04

Ubuntu 是 ROS 官方提供二进制包支持 Linux 发行版。Ubuntu 是 Linux 发行版之一，拥有广泛的用户，可以得到庞大的社区支持。Ubuntu 由 Canonical 公司维护支持，每 6 个月发布一个新版本，每两年发布一个 LTS（长期支持）版本，其中 Ubuntu 18.04 就是 LTS 版本。

## 1.3. ROS Melodic

ROS(Robot Operating System)即机器人操作系统，于 2007 年由美国加州的 Willow Garage 机器人研究实验室发起。目标是开发一种灵活的系统，以支持各种各样的机器人软件开发任务，最初主要是用于支持 Willow Garage 的 PR2 机器人平台。经过 ROS 的不断发展，ROS 已经成为全球范围内，用于学术研究、企业开发及机器人爱好者等领域的首选框架，对于学术界和工业界的机器人研究起到了推动作用。ROS 并非传统意义上的“操作系统”，它更像一个中间件或平台，提供硬件抽象、低级设备控制、实现常用功能的公用库、可视化、消息传递服务和包管理等功能，为机器人开发提供便利。

ROS Melodic 是 ROS 官方于 2018 年发布的长期支持版本，在 Ubuntu18.04 系统上提供二进制包维护支持。

## 2. 环境配置

### 2.1. 配置虚拟机

#### 2.1.1. 安装 Ubuntu 虚拟机

##### (1) 安装 VirtualBox 虚拟机软件

首先打开链接：<https://download.virtualbox.org/virtualbox/7.0.10/VirtualBox-7.0.10-158379-Win.exe> 下载 VirtualBox 虚拟机软件。

下载完成后，打开下载文件开始安装 VirtualBox，如下图所示。



点击下一步保持默认配置安装即可。

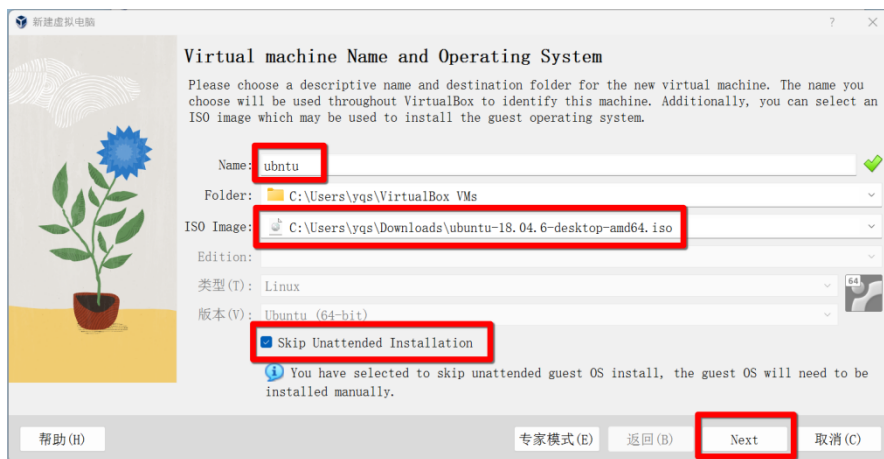
##### (2) 虚拟机挂载 Ubuntu 镜像

打开连接 <https://mirrors.ustc.edu.cn/ubuntu-releases/18.04.6/ubuntu-18.04.6-desktop-amd64.iso> 下载 Ubuntu18.04 镜像。下载完成后，打开 VirtualBox，按照下列图片所示安装 Ubuntu18.04。

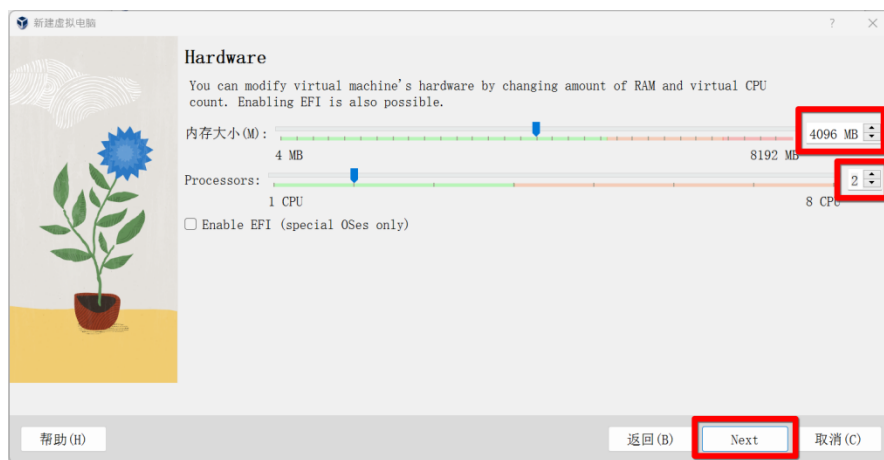
点击新建：



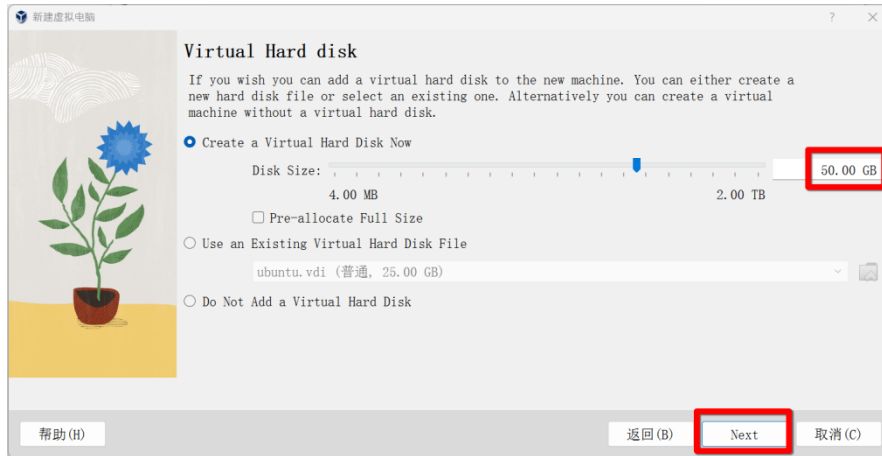
输入名称并选择下载的 Ubuntu 18.04 镜像，注意勾选”Skip Unattended Installation”。接着点击”Next”：



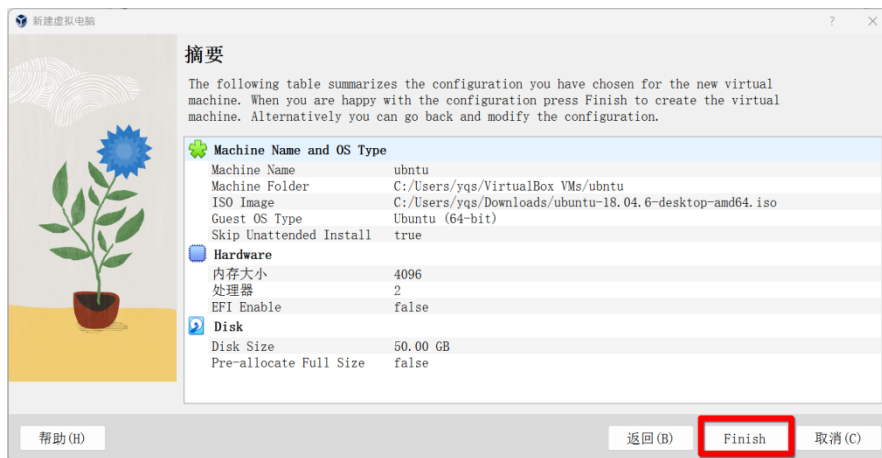
根据自己电脑的配置设置合适的虚拟机配置，接着点击”Next”：



添加虚拟硬盘，接着点击”Next”：



点击”Finish”，开始安装 Ubuntu 18.04 虚拟机：

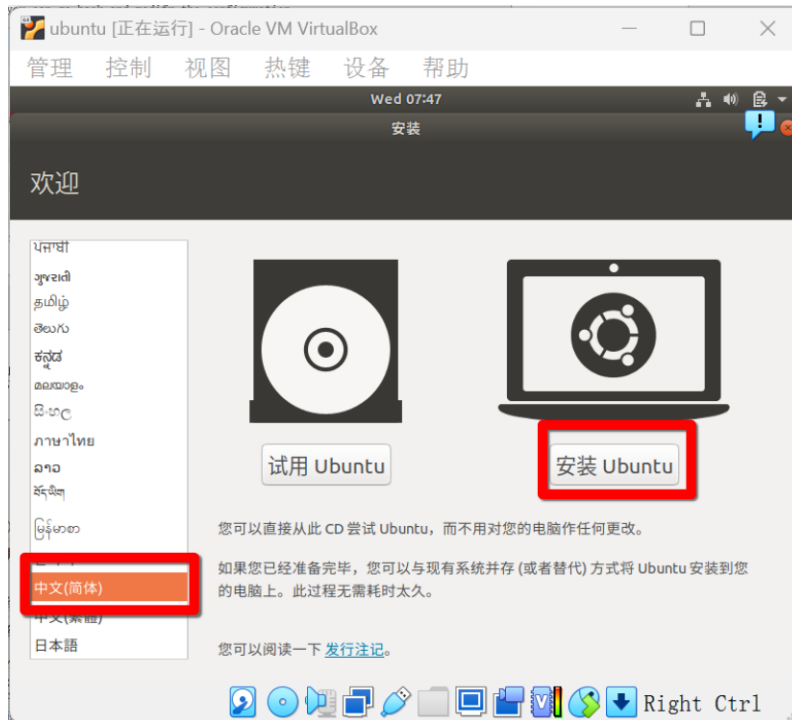


如果没有自动启动的话，点击”启动”并等待：

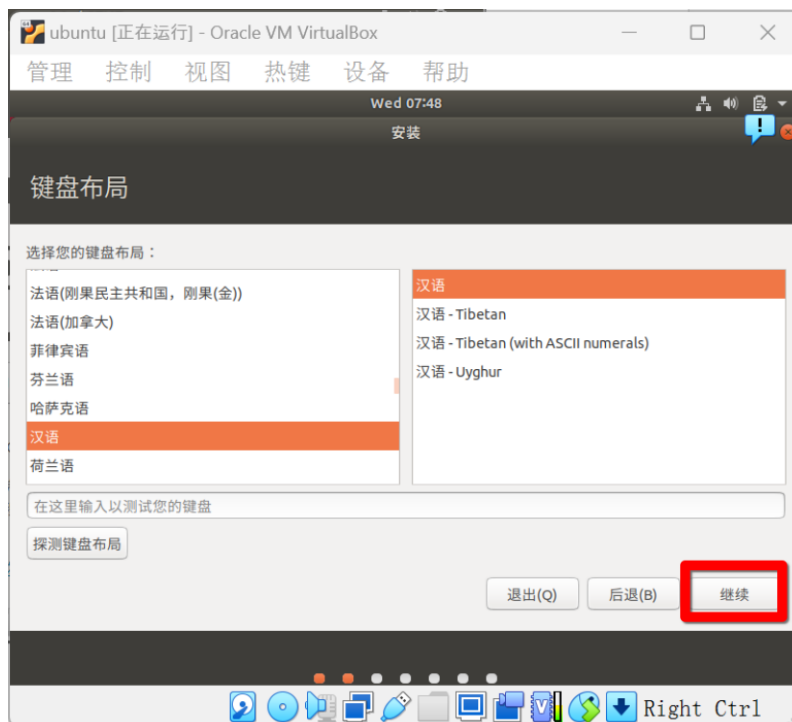


### (3) 安装 Ubuntu

选择”中文简体”，并点击”安装 Ubuntu”：



点击“继续”：



选择“最小安装”并取消勾选“安装 Ubuntu 时下载更新”，接着点击“继续”：



选择“清除整个磁盘并安装 Ubuntu”，并点击“现在安装”：



点击“继续”：

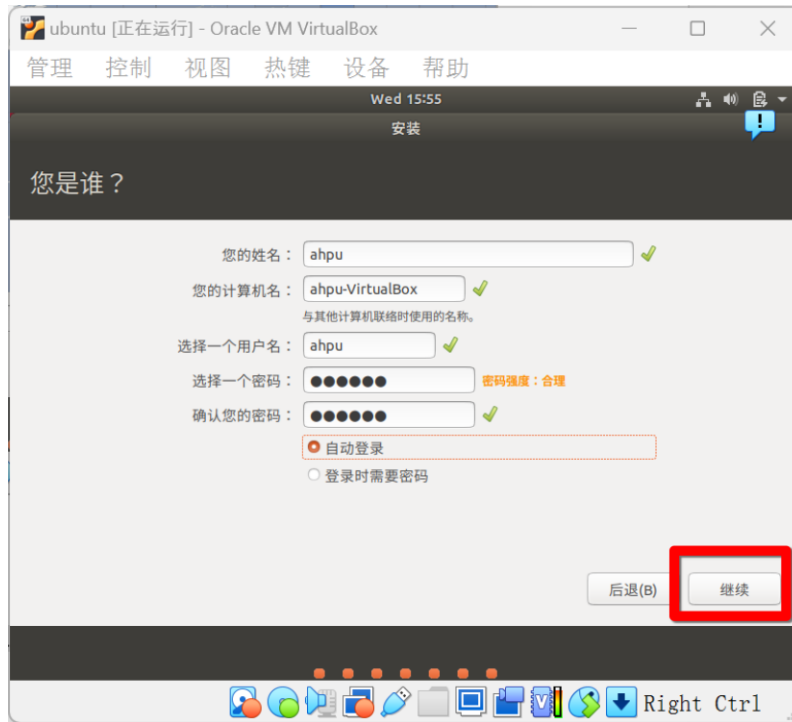




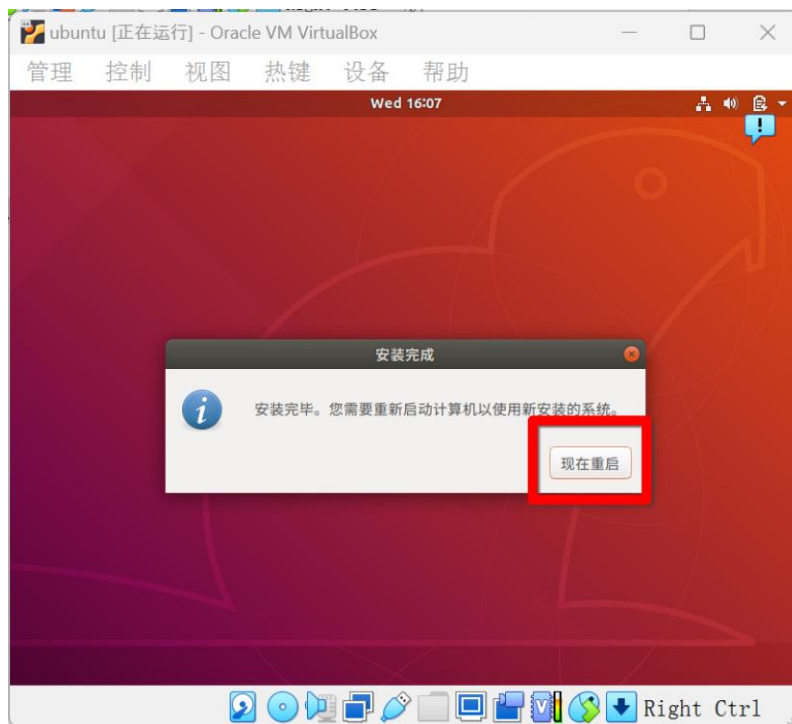
点击“继续”：



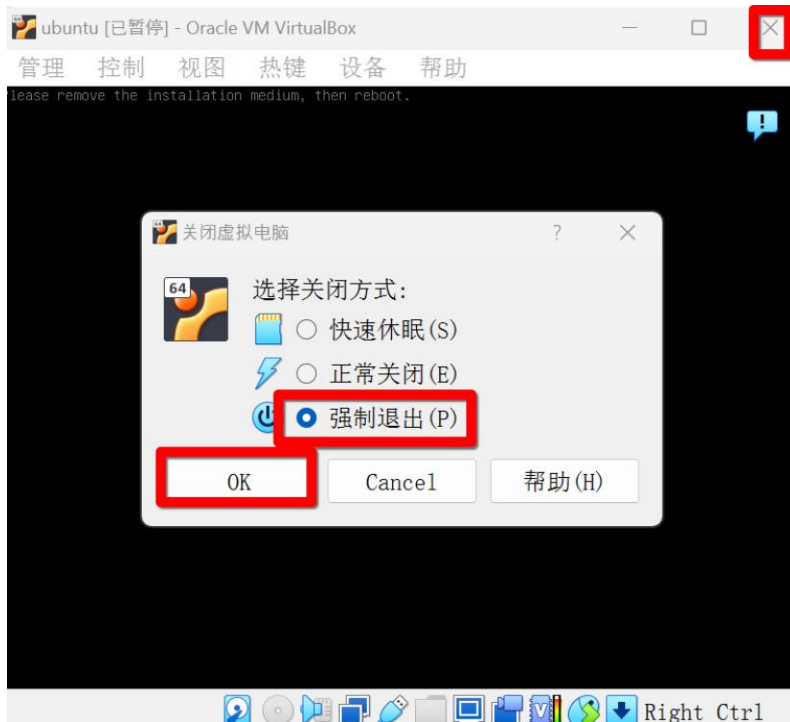
配置账号密码，接着点击“继续”，等待安装完成：



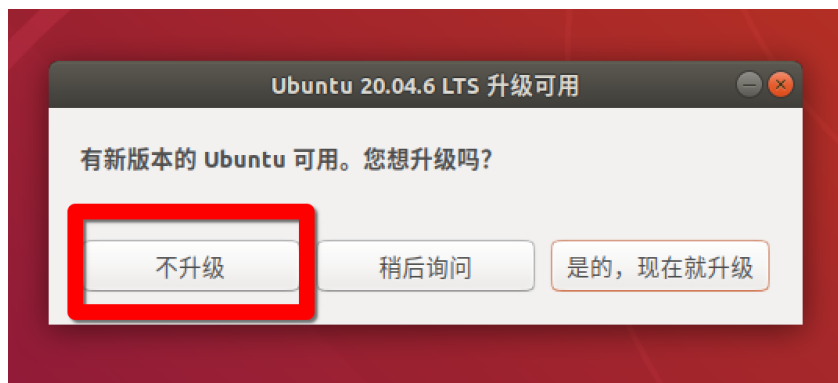
安装完成后，点击“现在重启”：



如果一直未重启成功，则点击右上角“X”并选择“强制退出”：



注意, 在后面打开使用 Ubuntu 虚拟机时, 如果遇到升级提醒, 选择”不升级”:



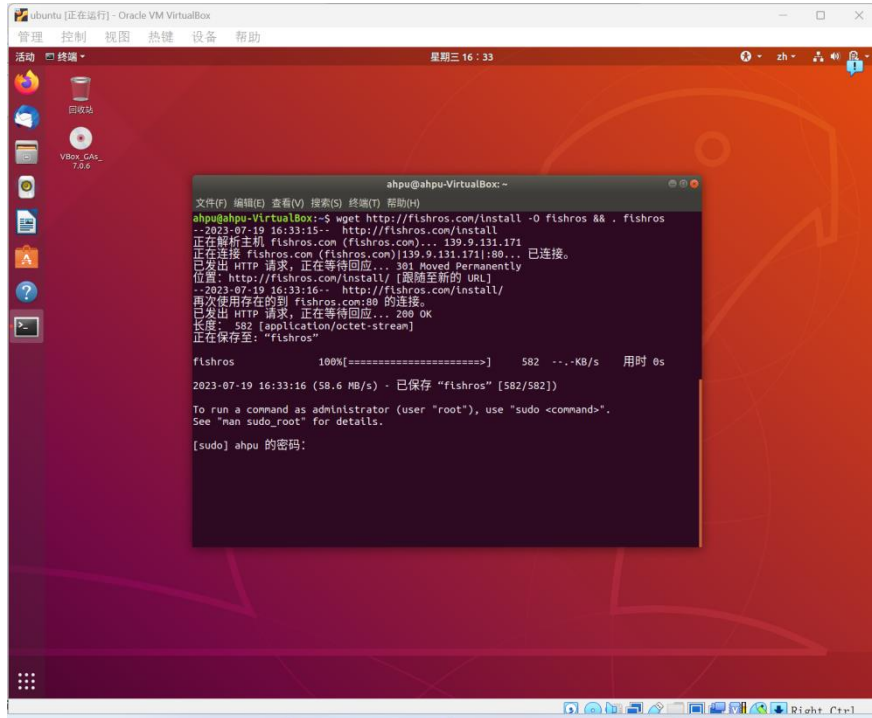
## 2.1.2. 配置 ROS 环境

### (1) 安装 ROS Melodic

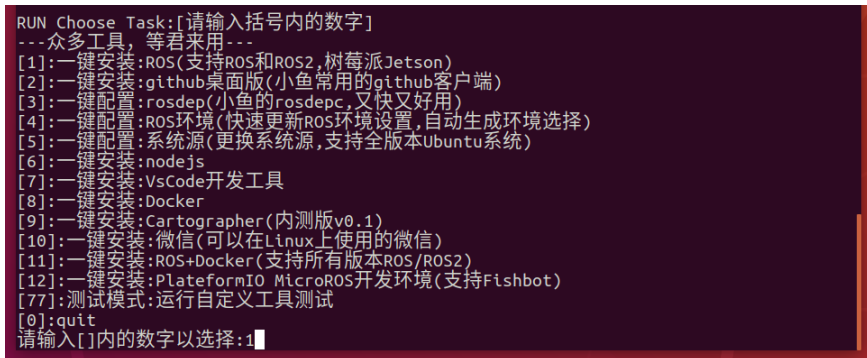
在 Ubuntu 虚拟机内按”Ctrl+Alt+t”或点击鼠标右键打开终端, 并执行以下 shell 命令:

```
wget http://fishros.com/install -O fishros && . fishros
```

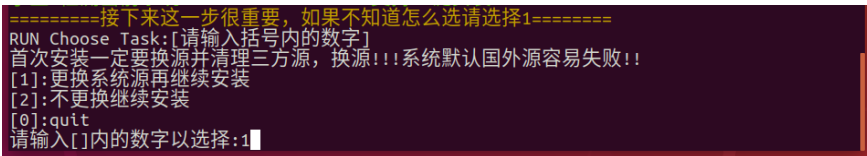
执行后如图, 接着输入密码并回车 (Linux 输入密码不会明文显示, 也不会显示\*等符号, 密码输完后回车即可):



选择”1”:



选择”1”:



选择”2”:

```
=====接下来这一步很重要，如果不知道怎么选请选择1=====
RUN Choose Task:[请输入括号内的数字]
首次安装一定要换源并清理三方源，换源!!!系统默认国外源容易失败!!
[1]:更换系统源再继续安装
[2]:不更换继续安装
[0]:quit
请输入[]内的数字以选择:1
欢迎使用一键换源工具，本工具由[鱼香ROS]小鱼贡献..
RUN Choose Task:[请输入括号内的数字]
请选择换源方式,如果不知道选什么请选2
[1]:仅更换系统源
[2]:更换系统源并清理第三方源
[0]:quit
请输入[]内的数字以选择:2
```

选择”1”:

```
RUN Choose Task:[请输入括号内的数字]
请选择你要安装的ROS版本名称(请注意ROS1和ROS2区别):
[1]:melodic(ROS1)
[2]:bouncy(ROS2)
[3]:crystal(ROS2)
[4]:dashing(ROS2)
[5]:eloquent(ROS2)
[0]:quit
请输入[]内的数字以选择:1
```

选择”1”，等待安装完成:

```
RUN Choose Task:[请输入括号内的数字]
请选择你要安装的ROS版本名称(请注意ROS1和ROS2区别):
[1]:melodic(ROS1)
[2]:bouncy(ROS2)
[3]:crystal(ROS2)
[4]:dashing(ROS2)
[5]:eloquent(ROS2)
[0]:quit
请输入[]内的数字以选择:1
RUN Choose Task:[请输入括号内的数字]
请选择安装的具体版本(如果不知道怎么选,请选1桌面版):
[1]:melodic(ROS1)桌面版
[2]:melodic(ROS1)基础版(小)
[0]:quit
请输入[]内的数字以选择:1
```

## (2) 创建 ROS 工作空间

首先打开终端执行以下命令创建文件夹:

```
mkdir -p ~/ahpu_robot/src
```

接着执行以下命令进入文件夹:

```
cd ~/ahpu_robot
```

执行以下命令编译:

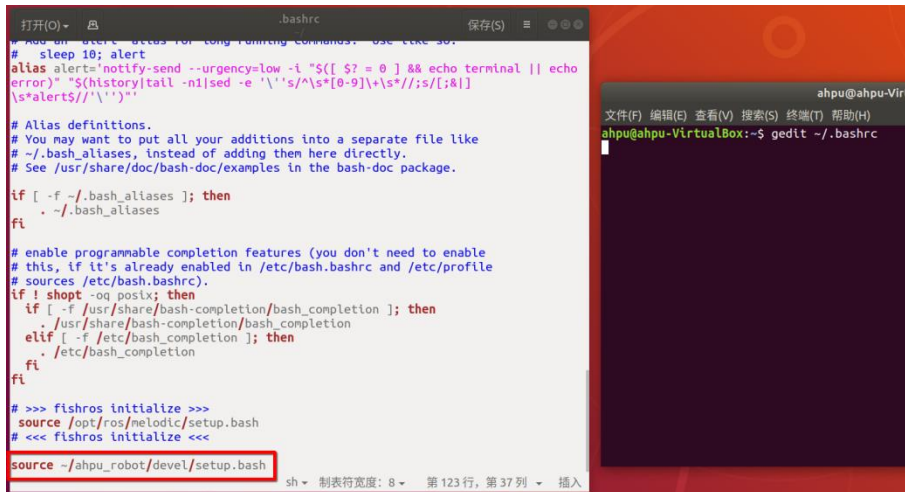
```
catkin_make
```

## (3) 添加环境变量

打开终端执行以下命令编辑.bashrc 文件:

```
gedit ~/.bashrc
```

在.bashrc 文件末尾添加”source ~/ahpu\_robot/devel/setup.bash”，如图所示:



```
.bashrc
# See an article titled 'alias for long running commands' Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${? = 0 }" && echo terminal || echo
error" "$(historytail -n|sed -e '\''s/^\s*[0-9]\+\s*//;s/;/&|
\&alert$//'\`)'
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq postfix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
# >>> fishros initialize >>>
source /opt/ros/melodic/setup.bash
# <<< fishros initialize <<<
source ~/ahpu_robot/develop/setup.bash
```

#### (4) 测试 ROS 环境

首先打开终端执行以下命令进入工作空间下的 src 文件夹：

```
cd ~/ahpu_robot/src
```

执行以下命令创建 hello 功能包：

```
catkin_create_pkg hello roscpp
```

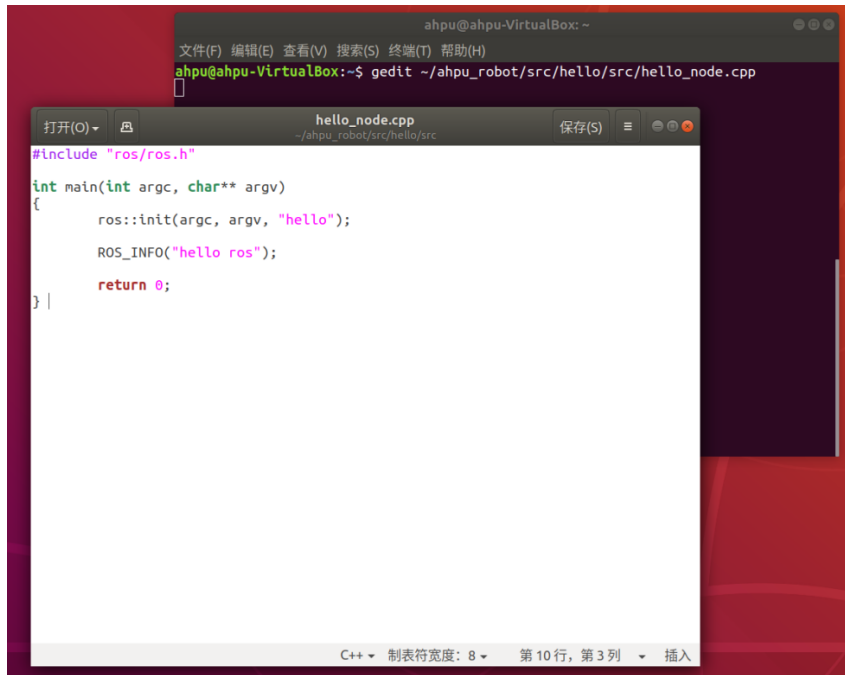
在 hello 功能包的 src 文件夹下新建 hello\_node.cpp 并编辑：

```
gedit ~/ahpu_robot/src/hello/src/hello_node.cpp
```

加入以下内容并保存：

```
#include "ros/ros.h"
int main(int argc, char** argv)
{
    ros::init(argc, argv, "hello");
    ROS_INFO("hello ros");
    return 0;
}
```

如图：



```
ahpu@ahpu-VirtualBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ahpu@ahpu-VirtualBox:~$ gedit ~/ahpu_robot/src/hello/src/hello_node.cpp

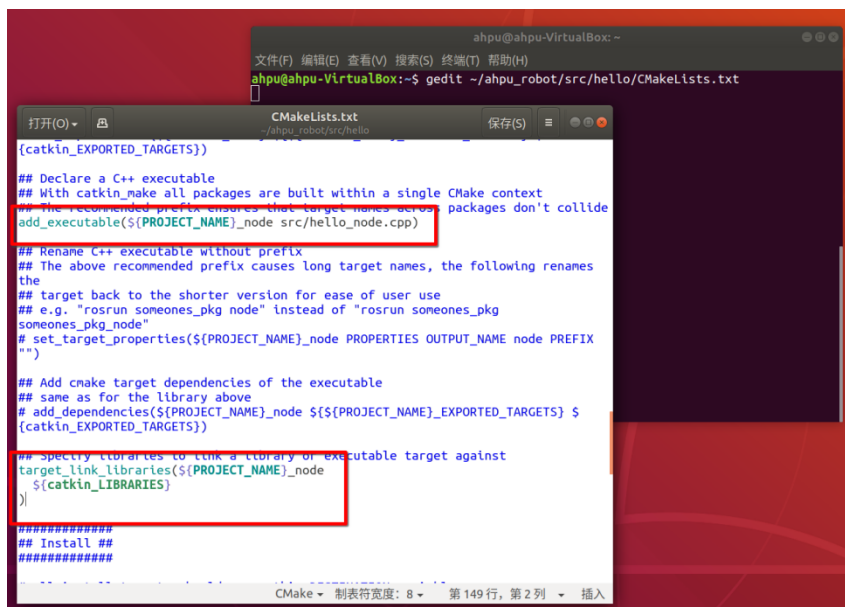
hello_node.cpp
~/ahpu_robot/src/hello/src

#include "ros/ros.h"

int main(int argc, char** argv)
{
    ros::init(argc, argv, "hello");
    ROS_INFO("hello ros");
    return 0;
}

C++ 制表符宽度: 8 第 10 行, 第 3 列 插入
```

接着修改 hello 功能包下的 CMakeLists.txt，取消如图所示的注释部分，即删除前面的”#”号：



```
ahpu@ahpu-VirtualBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ahpu@ahpu-VirtualBox:~$ gedit ~/ahpu_robot/src/hello/CMakeLists.txt

CMakeLists.txt
~/ahpu_robot/src/hello

[catkin_EXPORTED_TARGETS])

## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
add_executable(${PROJECT_NAME}_node src/hello_node.cpp)

## Rename C++ executable without prefix
## The above recommended prefix causes long target names, the following renames
## the
## target back to the shorter version for ease of user use
## e.g. "roslaunch someones_pkg node" instead of "roslaunch someones_pkg
## someones_pkg_node"
# set_target_properties(${PROJECT_NAME}_node PROPERTIES OUTPUT_NAME node PREFIX
"")

## Add cmake target dependencies of the executable
## same as for the library above
# add_dependencies(${PROJECT_NAME}_node ${${PROJECT_NAME}_EXPORTED_TARGETS} ${
catkin_EXPORTED_TARGETS})

## Specify libraries to link a library or executable target against
target_link_libraries(${PROJECT_NAME}_node
${catkin_LIBRARIES})

#####
## Install ##
#####

CMake 制表符宽度: 8 第 149 行, 第 2 列 插入
```

进入工作空间编译并编译：

```
cd ~/ahpu_robot/ && catkin_make
```

如图：

```
ahpu@ahpu-VirtualBox: ~/ahpu_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ahpu@ahpu-VirtualBox:~$ cd ~/ahpu_robot/ && catkin_make
Base path: /home/ahpu/ahpu_robot
Source space: /home/ahpu/ahpu_robot/src
Build space: /home/ahpu/ahpu_robot/build
Devel space: /home/ahpu/ahpu_robot/devel
Install space: /home/ahpu/ahpu_robot/install
####
#### Running command: "make cmake_check_build_system" in "/home/ahpu/ahpu_robot/build"
####
####
#### Running command: "make -j2 -l2" in "/home/ahpu/ahpu_robot/build"
####
Scanning dependencies of target hello_node
[ 50%] Building CXX object hello/CMakeFiles/hello_node.dir/src/hello_node.cpp.o
[100%] Linking CXX executable /home/ahpu/ahpu_robot/devel/lib/hello/hello_node
[100%] Built target hello_node
ahpu@ahpu-VirtualBox:~/ahpu_robot$
```

运行 hello 功能包下的节点测试:

```
roslaunch hello hello_node
```

如图:

```
ahpu@ahpu-VirtualBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ahpu@ahpu-VirtualBox:~$ roslaunch hello hello_node
[ INFO] [1689818711.236955247]: hello ros
ahpu@ahpu-VirtualBox:~$
```

虚拟机 ROS 环境配置完成。

## 2.2. 配置工控机

### 2.2.1. 安装 Ubuntu 系统

#### (1) 制作启动 U 盘

打开链接 <https://www.ventoy.net/cn/download.html> 下载并解压 Ventoy 工具。



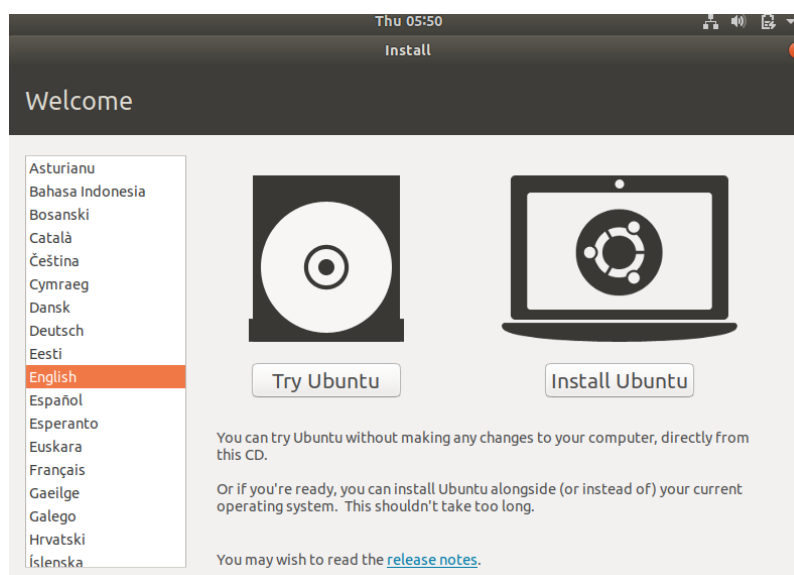
接着插上 U 盘到电脑，双击解压出的”Ventoy2Disk”打开。打开 Ventoy 工具后，选择要安装 Ventoy 的 U 盘并点击”安装”（**注意：给 U 盘安装 Ventoy 会清空 U 盘数据**），如图：



安装好后，把前面下好的 Ubuntu 18.04 镜像复制到 U 盘中。

## (2) 开机并安装 Ubuntu

U 盘插入工控机，开机后会进入 Ventoy 界面，选择 Ubuntu 18.04 镜像并回车。开机后进入 Ubuntu 安装界面：



后续安装步骤同虚拟机安装 Ubuntu。

## 2.2.2. 配置 ROS 环境

配置 ROS 环境步骤同虚拟机配置 ROS 环境。

## 2.3. 配置工控机与虚拟机的 ROS 通信

首先确保工控机与运行虚拟机的电脑连接同一 wifi。

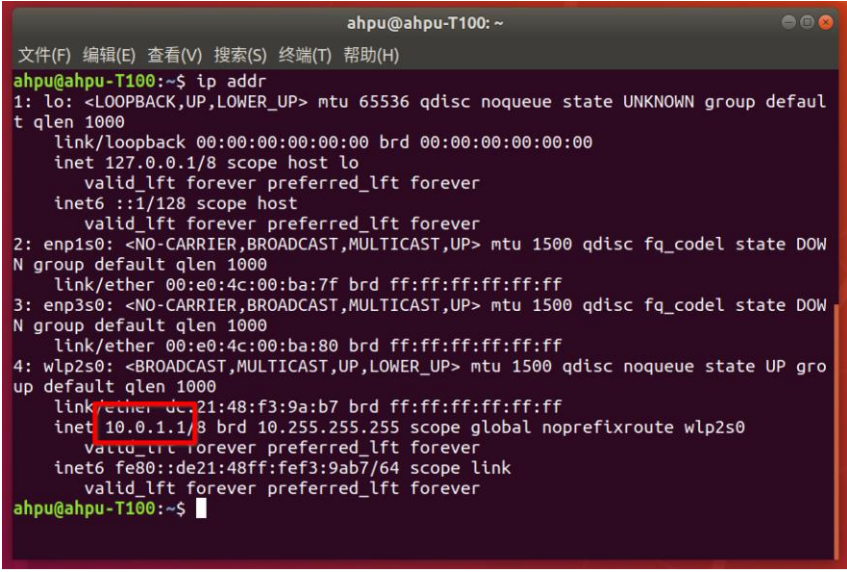
### 2.3.1. 工控机配置

#### (1) 查询 IP

进入 Ubuntu 打开终端，执行以下命令查询 IP：

```
ip addr
```

图中 10.0.1.1 即为工控机 IP：



```
ahpu@ahpu-T100: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
ahpu@ahpu-T100:~$ ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN  
    link/ether 00:e0:4c:00:ba:7f brd ff:ff:ff:ff:ff:ff  
3: enp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN  
    link/ether 00:e0:4c:00:ba:80 brd ff:ff:ff:ff:ff:ff  
4: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group  
    link/ether dc:21:48:f3:9a:b7 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.1.1/8 brd 10.255.255.255 scope global noprefixroute wlp2s0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::de21:48ff:fef3:9ab7/64 scope link  
        valid_lft forever preferred_lft forever  
ahpu@ahpu-T100:~$
```

#### (2) 配置环境变量

编辑.bashrc 并在末尾设置 ROS\_MASTER\_URI 与 ROS\_HOSTNAME 环境变量。添加以下内容：

```
export ROS_MASTER_URI=http://{主机 IP}:11311
```

```
export ROS_HOSTNAME={本机 IP}
```

把{主机 IP}和{本机 IP}替换为实际的 IP 地址，如图所示：

```
打开(O)  *.bashrc 保存(S)
error) $(hostnamectl | grep -e ^IP | sed -r 's/([0-9]{1,3}){3}/[&]/g')
\${*alert$/'\`'}"
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
# >>> fishros initialize >>>
source /opt/ros/melodic/setup.bash
# <<< fishros initialize <<<
source ~/ahpu_robot/devel/setup.bash
export ROS_MASTER_URI=http://10.0.1.1:11311
export ROS_HOSTNAME=10.0.1.1
```

其中 ROS\_MASTER\_URI 为 ROS 主节点地址，这里选用工控机运行 ROS 主节点，所以这里是工控机 IP；ROS\_HOSTNAME 为本机 IP 地址，因为是工控机上的配置，因此这里是工控机 IP 地址。

### 2.3.2. 虚拟机配置

(1) 设置虚拟机网络为桥接模式

点击”设置”：



点击”网络”，选择”桥接网卡”，点击”OK”：



## (2) 查询 IP，配置环境变量

进入虚拟机打开终端，执行以下命令查询 IP：

```
ip addr
```

图中 10.0.1.251 即为虚拟机 IP：

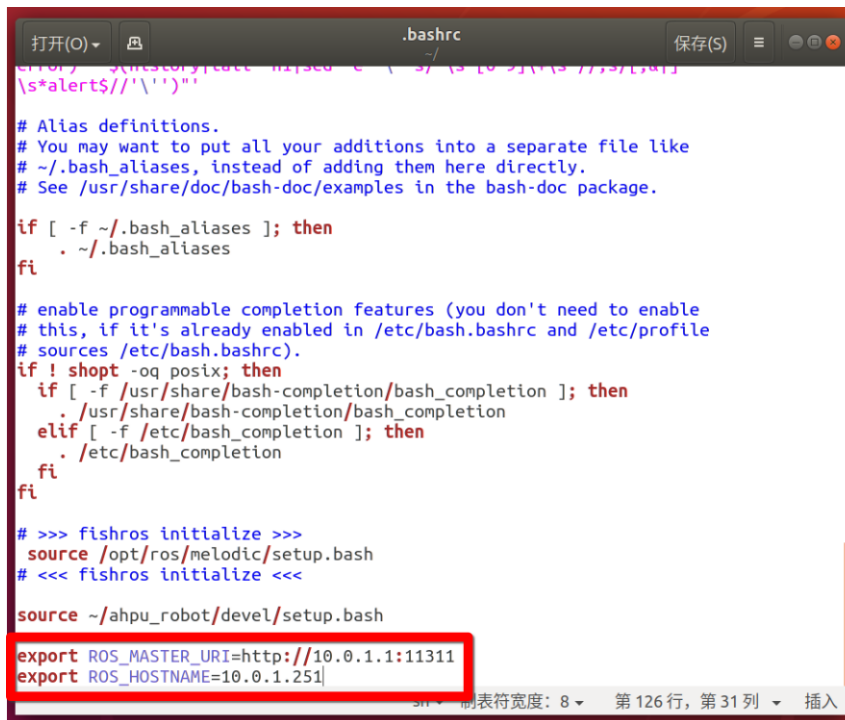
```
ahpu@ahpu-VirtualBox: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
ahpu@ahpu-VirtualBox:~$ ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 82:55:c2:7f:fc:2a:ca brd ff:ff:ff:ff:ff:ff  
    inet 10.0.1.251/24 brd 10.255.255.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 3575sec preferred_lft 3575sec  
    inet6 fe80::aae1:36e6:80f9:b9cf/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
ahpu@ahpu-VirtualBox:~$
```

编辑.bashrc 并在末尾设置 ROS\_MASTER\_URI 与 ROS\_HOSTNAME 环境变量。添加以下内容：

```
export ROS_MASTER_URI=http://{主机 IP}:11311
```

```
export ROS_HOSTNAME={本机 IP}
```

把{主机 IP}和{本机 IP}替换为实际的 IP 地址，如图所示：



```
打开(O)  .bashrc 保存(S)
error) $(testconfig) nlsed e \ / (s [0-9 ]*) [0-9 ]*)
\s*alert$/'\''"'
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

# >>> fishros initialize >>>
source /opt/ros/melodic/setup.bash
# <<< fishros initialize <<<

source ~/ahpu_robot/devel/setup.bash

export ROS_MASTER_URI=http://10.0.1.1:11311
export ROS_HOSTNAME=10.0.1.251
```

其中 ROS\_MASTER\_URI 为 ROS 主节点地址，这里选用工控机运行 ROS 主节点，所以这里是工控机 IP；ROS\_HOSTNAME 为本机 IP 地址，因为是虚拟机上的配置，因此这里是虚拟机 IP 地址。

### 2.3.3. 测试 ROS 通信

在工控机上执行以下命令，打开 ROS 主节点：

```
roscore
```

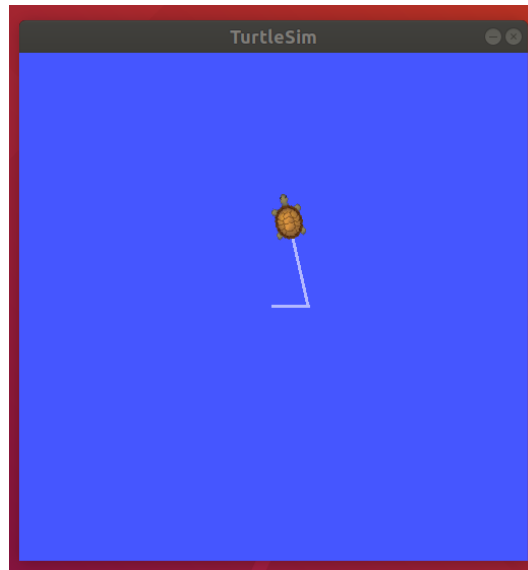
接着在工控机上执行以下命令运行小乌龟界面：

```
roslaunch turtlesim turtlesim_node
```

在虚拟机上执行以下命令，运行小乌龟控制节点：

```
roslaunch turtlesim turtle_teleop_key
```

在虚拟机上点击电脑键盘方向键，若工控机上的小乌龟运动则说明 ROS 通信配置成功，如图：

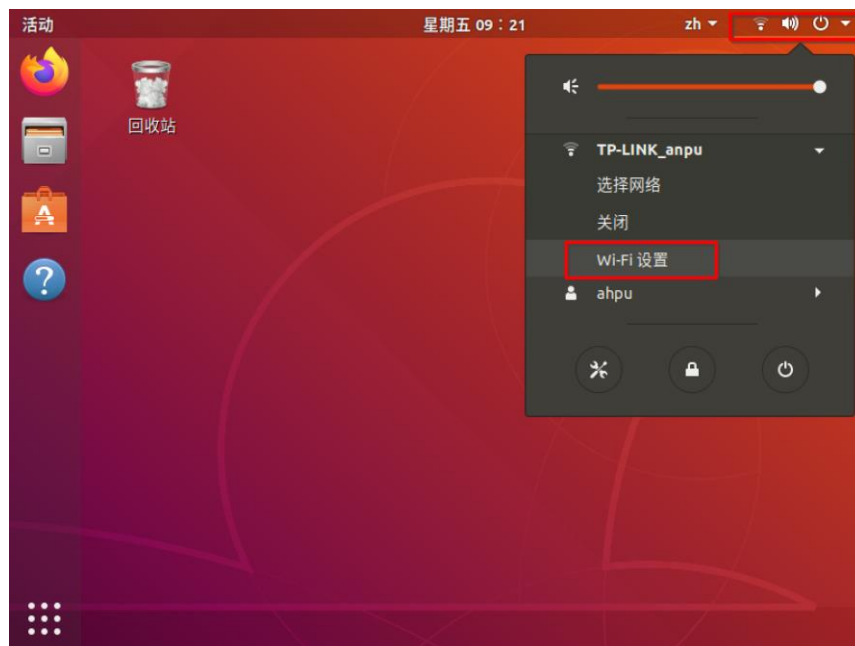


### 2.3.4. 设置静态 IP

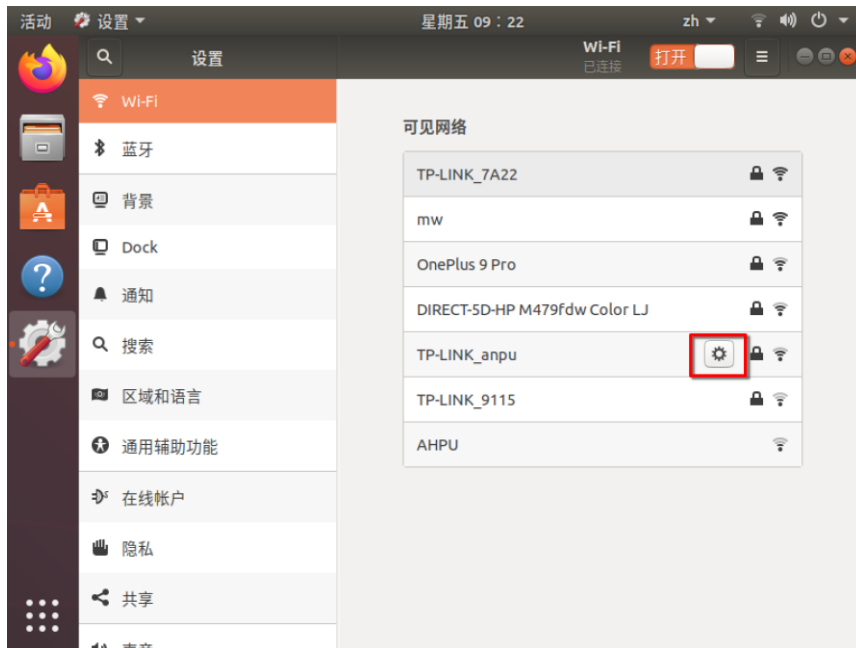
有时工控机或虚拟机的 IP 可能会变动，需要用新的 IP 重新设置 ROS\_MASTER\_URI 和 ROS\_HOSTNAME 环境变量。可以为工控机和虚拟机设置静态 IP 避免 IP 变动。

#### (1) 工控机设置静态 IP

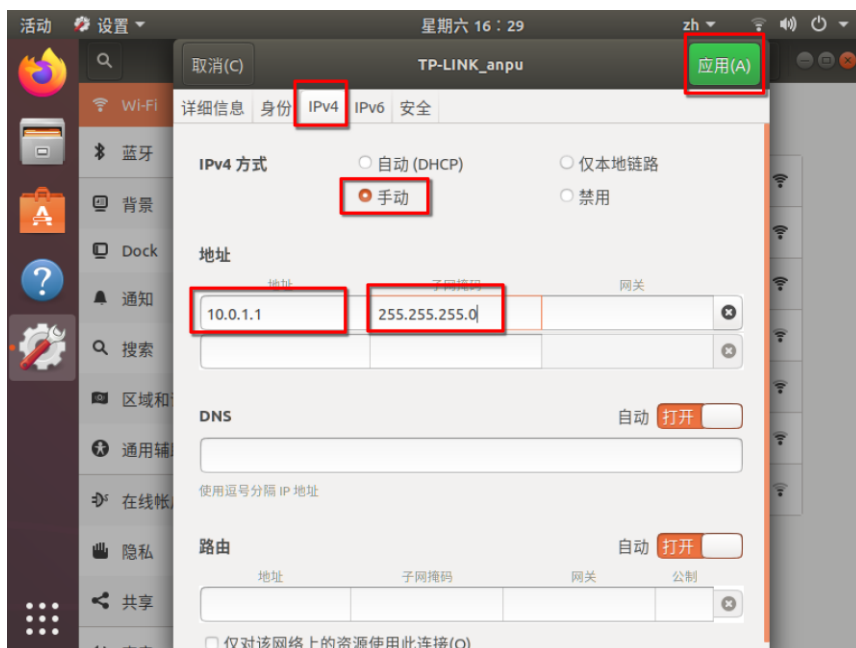
打开 wifi 设置：



点击所连接 wifi 的设置图标：

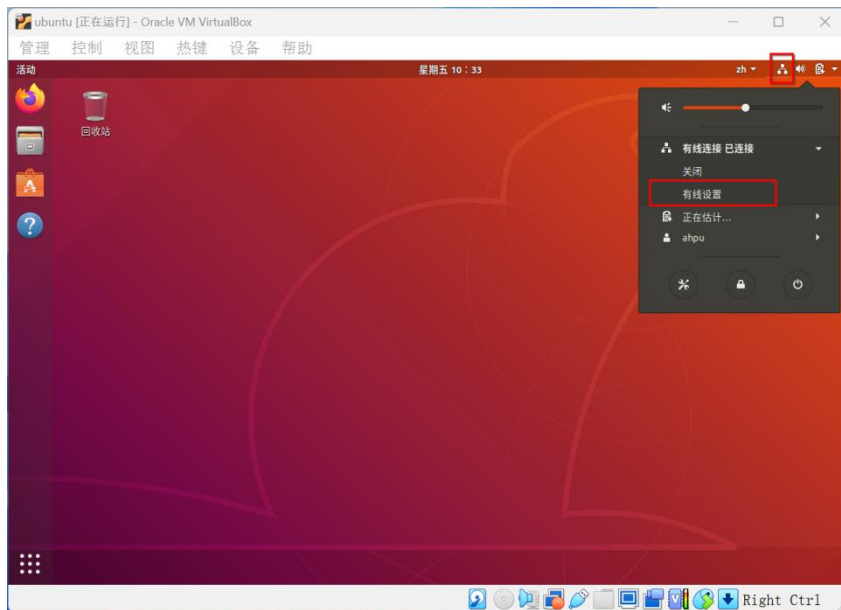


设置静态 IP，注意这里设置的 IP 最好与前面执行 `ip addr` 查询的 IP 一致。如果要改成别的，IP 的最后一位需要避免与其他局域网设备 IP 的最后一位冲突：

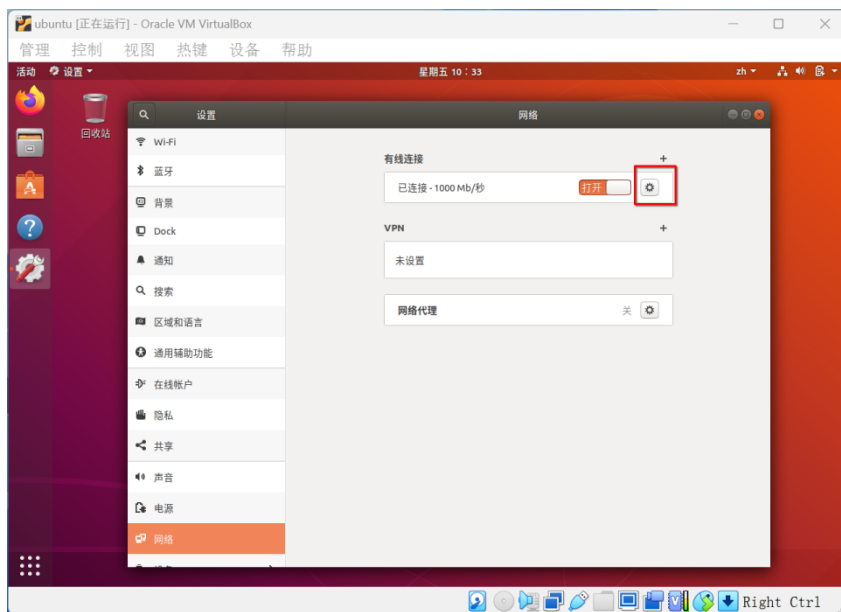


## (2) 虚拟机设置静态 IP

打开有线设置：



点击有线连接旁边的设置图标：



设置静态 IP，注意这里设置的 IP 最好与前面执行 `ip addr` 查询的 IP 一致。  
如果要改成别的，IP 的最后一位需要避免与其他局域网设备 IP 的最后一位冲突：



