



# 六叶树产品教程-x86\_64 下 Docker 使用教程

| 类别  | 内容                                   |
|-----|--------------------------------------|
| 关键词 | Docker 使用教程                          |
| 摘要  | 本教程讲解如何在 Docker 环境下使用六叶树 USBCANFD 产品 |

相关环境介绍:

| 类别          | 内容          |
|-------------|-------------|
| Docker 宿主系统 | Ubuntu22.04 |
| Docker 版本   | 27.4.1      |

文档记录

| 版本    | 日期      | 说明                                           |
|-------|---------|----------------------------------------------|
| V1.00 | 2025.01 | 创建文档                                         |
| V1.01 | 2025.01 | 1.新增 Docker 安装教程<br>2.新增 CanAssistant 镜像使用教程 |



## 目录

|                                     |    |
|-------------------------------------|----|
| 六叶树产品教程-Docker 使用教程 .....           | 1  |
| 1、 Docker 教程 .....                  | 3  |
| 1.1 Docker 安装 .....                 | 3  |
| 1.2 Docker 镜像的导入 .....              | 5  |
| 1.3 Docker 容器创建 .....               | 6  |
| 1.4 Docker 容器启动 .....               | 7  |
| 1.5 Docker 容器停止 .....               | 7  |
| 1.6 Docker 进入容器 .....               | 7  |
| 2、 linux 通用用例使用说明 .....             | 7  |
| 第一步，启动并进入容器 .....                   | 7  |
| 第二步，工程代码下载 .....                    | 8  |
| 第三步，代码编译与运行 .....                   | 8  |
| 常见问题: .....                         | 10 |
| 3、 CanAssistant Docker 镜像使用说明 ..... | 11 |
| 第一步，导入镜像 .....                      | 11 |
| 第二步，创建并启动容器 .....                   | 11 |
| 3 常用链接 .....                        | 18 |
| 4 联系方式 .....                        | 18 |



## 1、Docker 教程

### 1.1 Docker 安装

#### 1.1.1 x86\_64 架构

演示安装环境:

| 项目     | 版本          |
|--------|-------------|
| CPU 架构 | X86_64      |
| 操作系统   | Ubuntu22.04 |

```
root@lys-virtual-machine:/home/lys# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 22.04.5 LTS
Release:      22.04
Codename:     jammy
root@lys-virtual-machine:/home/lys#
```

注意:所有指令使用具有 root 权限的账户

### 1.更新软件包索引

```
apt-get update
apt-get -y upgrade
```

### 2.允许 APT 使用 HTTPS

apt-get install apt-transport-https ca-certificates curl software-properties-common

```
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys# apt-get install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203-22.04.1).
ca-certificates set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https curl
0 upgraded, 2 newly installed, 0 to remove and 8 not upgraded.
Need to get 196 kB of archives.
After this operation, 625 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Ign:1 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.13
Ign:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.20
Get:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.13 [1,510 B]
Get:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.20 [194 kB]
Fetched 196 kB in 45s (4,304 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 187334 files and directories currently installed.)
```

### 3.添加 Docker 官方 GPG 密钥

```
curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
```



```
root@lys-virtual-machine:/home/lys#  
root@lys-virtual-machine:/home/lys# curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
root@lys-virtual-machine:/home/lys#
```

## 4. 添加 Docker 软件源

add-apt-repository "deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu \$(lsb\_release -cs) stable"

```
root@lys-virtual-machine:/home/lys# curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
root@lys-virtual-machine:/home/lys# add-apt-repository "deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb_release -cs) stable"  
Repository: 'deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy stable'  
Description:  
Archive for codename: jammy components: stable  
More info: http://mirrors.aliyun.com/docker-ce/linux/ubuntu  
Adding repository.  
Press [ENTER] to continue or Ctrl-c to cancel.  
Adding deb entry to /etc/apt/sources.list.d/archive_uri-http_mirrors.aliyun.com_docker-ce_linux_ubuntu-jammy.list  
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-http_mirrors.aliyun.com_docker-ce_linux_ubuntu-jammy.list  
Hit:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy InRelease  
Hit:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates InRelease  
Hit:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-backports InRelease  
Ign:4 http://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy InRelease  
Ign:5 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Get:4 http://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy InRelease [48.8 kB]  
Get:6 http://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages [42.5 kB]  
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Fetched 91.3 kB in 24s (3,836 B/s)  
Reading package lists... Done  
W: http://mirrors.aliyun.com/docker-ce/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
root@lys-virtual-machine:/home/lys#
```

## 5. 再次更新软件包索引

apt-get update

## 6. 安装 Docker CE (社区版)

apt-get install -y docker-ce docker-ce-cli containerd.io

## 7. 验证 Docker 是否安装成功并运行

systemctl status docker

```
root@lys-virtual-machine:/home/lys# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
   Active: active (running) since Tue 2025-01-07 16:02:35 CST; 9s ago  
 Triggered By: ● docker.socket  
    Docs: https://docs.docker.com  
   Main PID: 47098 (dockerd)  
     Tasks: 8  
    Memory: 21.0M  
       CPU: 324ms  
    CGroup: /system.slice/docker.service  
            └─47098 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
1月 07 16:02:32 lys-virtual-machine systemd[1]: Starting Docker Application Container Engine...  
1月 07 16:02:32 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:32.417437761+08:00" level=info msg="Starting up"  
1月 07 16:02:32 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:32.418696799+08:00" level=info msg="OTEL tracing is not configured"  
1月 07 16:02:32 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:32.419029006+08:00" level=info msg="detected 127.0.0.53 nameserver"  
1月 07 16:02:32 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:32.682524266+08:00" level=info msg="Loading containers: start."  
1月 07 16:02:34 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:34.983789366+08:00" level=info msg="Loading containers: done."  
1月 07 16:02:35 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:35.044507792+08:00" level=info msg="Docker daemon commit=c710b"  
1月 07 16:02:35 lys-virtual-machine dockerd[47098]: time="2025-01-07T16:02:35.079927240+08:00" level=info msg="API listen on /run/docker.s"  
1月 07 16:02:35 lys-virtual-machine systemd[1]: Started Docker Application Container Engine.  
lines 1-22/22 (END)
```

## 8. 查看版本

docker --version



```
root@lys-virtual-machine:/home/lys#  
root@lys-virtual-machine:/home/lys#  
root@lys-virtual-machine:/home/lys# docker --version  
Docker version 27.4.1, build b9d17ea  
root@lys-virtual-machine:/home/lys# ^C  
root@lys-virtual-machine:/home/lys#
```

## 1.2 Docker 镜像的导入

### 方式 1:通过 Dockerfile 文件

通过使用六叶树提供的 Dockerfile 文件导入到 Docker 环境。

进入到 Dockerfile 文件所在的目录执行命令:

```
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo# ls  
Dockerfile  
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo#
```

语法:

`docker build -t "镜像名称:版本" ./`

镜像名称和版本是自定义的。

示例:

`docker build -t lys.ubuntu.18.04.usbcan.demo:20250106 ./`

```
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo# docker build -t lys.ubuntu.18.04.usbcan.demo:20250106 ./  
[+] Building 0.3s (12/12) FINISHED  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 431B  
=> [internal] load metadata for docker.io/library/ubuntu:18.04  
=> [internal] load .dockerignore  
=> => transferring context: 2B  
=> [1/8] FROM docker.io/library/ubuntu:18.04  
=> CACHED [2/8] RUN mkdir /home/lys  
=> CACHED [3/8] RUN cd /home/lys  
=> CACHED [4/8] RUN apt update && apt -y upgrade  
=> CACHED [5/8] RUN apt install -y ca-certificates  
=> CACHED [6/8] RUN sed -i "s@(\archive|security\).ubuntu.com@mirrors.tuna.tsinghua.edu.cn@" /etc/apt/sources.list  
=> CACHED [7/8] RUN apt update && apt -y upgrade  
=> CACHED [8/8] RUN apt install -y zip g++ gcc usbutils make wget libusb-1.0-0  
=> exporting to image  
=> => exporting layers  
=> => writing image sha256:53d932d8566fcd2d5d2622ea465260be656d57296ac974e2b73db9a7a2144586  
=> => naming to docker.io/library/lys.ubuntu.18.04.usbcan.demo:20250106  
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo#
```

完成

查看当前系统里的 Docker 镜像:

命令:

`docker images`

```
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo#  
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo#  
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo# docker images  
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE  
lys.ubuntu.24.04.usbcan.demo  20250106           9fd1082852dd       About an hour ago   391MB  
lys.ubuntu.18.04.usbcan.demo  20250106           53d932d8566f       15 hours ago       318MB  
ubuntu               20.04              6013a61a63c2       2 months ago       72.8MB  
ubuntu               22.04              97271d29cb79       3 months ago       77.9MB  
ubuntu               18.04              f9a80a55f492       19 months ago      63.2MB  
root@lys-VMware-Virtual-Platform:/home/lys/usbcanDemo#
```





## 方式 2:通过 Docker 镜像文件

通过使用六叶树提供的 Docker 镜像文件导入到 Docker 环境。

```
root@lys-VMware-Virtual-Platform:/home/lys/docker#  
root@lys-VMware-Virtual-Platform:/home/lys/docker# ls  
lys.x86_64.ubuntu.18.04.usbcan.demo.tar  
root@lys-VMware-Virtual-Platform:/home/lys/docker#
```

六叶树docker镜像文件

进入到 Docker 镜像文件所在的目录执行命令:

语法:

docker load -i “六叶树镜像文件名称”

示例:

```
docker load -i lys.x86_64.ubuntu.18.04.usbcan.demo.tar  
root@lys-VMware-Virtual-Platform:/home/lys/docker# docker load -i lys.x86_64.ubuntu.18.04.usbcan.demo.tar  
Loaded image: lys.ubuntu.18.04.usbcan.demo:20250106  
root@lys-VMware-Virtual-Platform:/home/lys/docker#
```

查看当前系统里的 Docker 镜像:

命令:

```
docker images  
root@lys-VMware-Virtual-Platform:/home/lys/docker# docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE  
lys.ubuntu.24.04.usbcan.demo 20250106    9fd1082852dd     About an hour ago 391MB  
lys.ubuntu.18.04.usbcan.demo 20250106    53d932d8566f     15 hours ago    318MB  
ubuntu               20.04       6013ae1a63c2     2 months ago    72.8MB  
ubuntu               22.04       97271d29cb79     3 months ago    77.9MB  
ubuntu               18.04       f9a80a55f492     19 months ago   63.2MB  
root@lys-VMware-Virtual-Platform:/home/lys/docker#
```

成功

## 1.3 Docker 容器创建

### 1.创建容器:

docker create -it --device=/dev/bus/usb:/dev/bus/usb --privileged --name lys-container lys.ubuntu.18.04.usbcan.demo:20250106

```
root@lys-VMware-Virtual-Platform:/home/lys#  
root@lys-VMware-Virtual-Platform:/home/lys# docker create -it --device=/dev/bus/usb:/dev/bus/usb --privileged --name lys-container lys.ubuntu.18.04.usbcan.demo:20250106  
27f36f541564d95daceac68ff8a5c8aa6e02da62c0a2ca937d9940f8291  
root@lys-VMware-Virtual-Platform:/home/lys#
```

### 2.查看到上面创建的容器:

命令:

```
docker ps -a  
root@lys-VMware-Virtual-Platform:/home/lys# docker ps -a  
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS          NAMES  
27f36f541564   lys.ubuntu.18.04.usbcan.demo:20250106  "/bin/bash"            About a minute ago  Created                          lys-container  
root@lys-VMware-Virtual-Platform:/home/lys#
```

**注意:容器创建指令一定要加参数:“--device=/dev/bus/usb:/dev/bus/usb --privileged”, 否则程序访问六叶树设备会报错。**



### 1.4 Docker 容器启动

命令:

`docker start lys-container`

```
root@lys-VMware-Virtual-Platform:/home/lys# docker start lys-container
lys-container
root@lys-VMware-Virtual-Platform:/home/lys#
```

查看容器状态:

命令:

`docker ps -a`

```
root@lys-VMware-Virtual-Platform:/home/lys# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS         PORTS   NAMES
27f36f541564   lys.ubuntu.18.04.usbcan.demo:20250106  "/bin/bash"            4 minutes ago   Up 41 seconds  已启动 lys-container
root@lys-VMware-Virtual-Platform:/home/lys#
```

### 1.5 Docker 容器停止

命令:

`docker stop lys-container`

### 1.6 Docker 进入容器

命令:

`docker exec -it lys-container /bin/bash`

```
root@lys-VMware-Virtual-Platform:/home/lys#
root@lys-VMware-Virtual-Platform:/home/lys# docker exec -it lys-container /bin/bash
root@27f36f541564:/#
```

## 2、linux 通用用例使用说明

六叶树提供的 Dockerfile 和镜像里是不包含通用用例代码的，只是构建后了用例编译以及运行所依赖的环境，代码的下载和编译、运行需要用户自己操作。

以下简要介绍 x86\_64 平台下 linux 通用用例工程的使用。

### 第一步，启动并进入容器

见上面的 Docker 教程。



## 第二步，工程代码下载

下载工程代码到 Docker 容器。

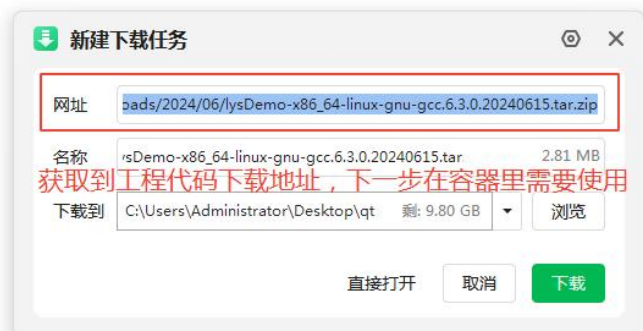
六叶树 USBCAN 适配器二次开发用例下载地址:

[http://www.liuyeshu.cn/?page\\_id=440](http://www.liuyeshu.cn/?page_id=440)

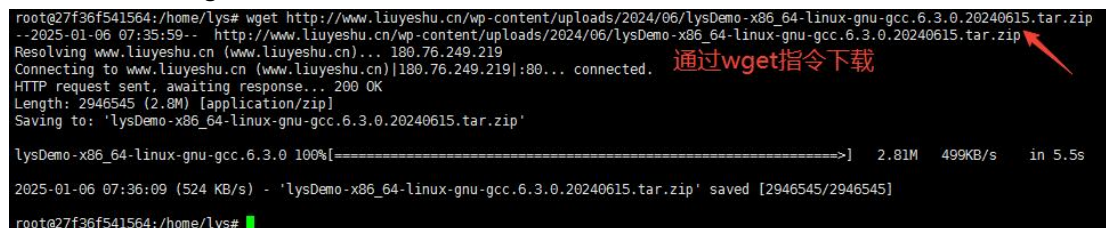
x86/64.



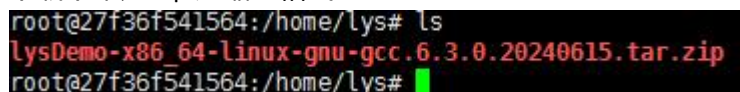
获取到下载地址:



回到容器通过 wget 进行下载



下载下来是 zip 压缩包格式



## 第三步，代码编译与运行

Zip 解压:

unzip lysDemo-x86\_64-linux-gnu-gcc.6.3.0.20240615.tar.zip





## Docker 使用教程

```
root@27f36f541564:/home/lys# unzip lysDemo-x86_64-linux-gnu-gcc.6.3.0.20240615.tar.zip
Archive: lysDemo-x86_64-linux-gnu-gcc.6.3.0.20240615.tar.zip
  inflating: ^q USBCAN^p'o^`_x^3d^2CAN&CANFD).pdf
  inflating: lysDemo-x86_64-linux-gnu-gcc.6.3.0.20240615.tar.gz
root@27f36f541564:/home/lys# ls
lysDemo-x86_64-linux-gnu-gcc.6.3.0.20240615.tar.gz
lysDemo-x86_64-linux-gnu-gcc.6.3.0.20240615.tar.zip
'$'\301\371\322\266\312\367' 'USBAN'$'\266\376\264\316\277\252\267\242\272\257\312\375\275\323\277\332\31
AN&CANFD).pdf'
```

gz 解压：

tar -xvf lysDemo-x86\_64-linux-gnu-gcc.6.3.0.20240615.tar.gz

```
root@27f36f541564:/home/lys# tar -xvf lysDemo-x86_64-linux-gnu-gcc.6.3.0.20240615.tar.gz
lysDemo/canframe.h
lysDemo/config.h
lysDemo/ControlCAN.h
lysDemo/libControlCAN.a
lysDemo/libControlCAN.so
lysDemo/libusb-1.0.a
lysDemo/libusb-1.0.la
lysDemo/libusb-1.0.so
lysDemo/libusb-1.0.so.0
lysDemo/libusb-1.0.so.0.1.0
lysDemo/libusb.h
lysDemo/lyscan-test
lysDemo/lyscan-test.c
lysDemo/lysfd-test
lysDemo/lysfd-test.c
lysDemo/Makefile
lysDemo/typedef.h
```

进入到工程目录：

cd lysDemo

```
root@27f36f541564:/home/lys# cd lysDemo
root@27f36f541564:/home/lys/lysDemo# ls
ControlCAN.h  canframe.h  libControlCAN.a  libusb-1.0.a  libusb-1.0.so  libusb-1.0.so.0.1.0  lyscan-test  lysfd-test  typedef.h
Makefile      config.h    libControlCAN.so  libusb-1.0.la  libusb-1.0.so.0  libusb.h             lyscan-test.c  lysfd-test.c
```

编译：

make

```
root@27f36f541564:/home/lys/lysDemo# make
rm -vf lyscan-test lysfd-test
removed 'lyscan-test'
removed 'lysfd-test'
gcc -o lyscan-test lyscan-test.c -L ./ -pthread -lrt -lControlCAN -lusb-1.0
gcc -o lysfd-test lysfd-test.c -L ./ -pthread -lrt -lControlCAN -lusb-1.0
root@27f36f541564:/home/lys/lysDemo# ls
ControlCAN.h  canframe.h  libControlCAN.a  libusb-1.0.a  libusb-1.0.so  libusb-1.0.so.0.1.0  lyscan-test  lysfd-test  typedef.h
Makefile      config.h    libControlCAN.so  libusb-1.0.la  libusb-1.0.so.0  libusb.h             lyscan-test.c  lysfd-test.c
```

直接make

生成的可执行程序

运行：

export LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:./

./lyscan-test

```
root@27f36f541564:/home/lys/lysDemo# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:./
root@27f36f541564:/home/lys/lysDemo# ./lyscan-test
设置CAN0波特率成功
DEMO功能: CAN1波特率1M, 接收到什么数据就回发什么数据。
```

启动成功



### 常见问题:

#### 1.此设备不存在

```
root@27f36f541564:/home/lys/lysDemo# ./lyscan-test  
此设备不存在  
root@27f36f541564:/home/lys/lysDemo# █
```

原因:六叶树 USBCANFD 设备没有接到主机或没有接到容器

可以通过 lsusb 指令查看 USB 设备已连接:

命令:

lsusb

```
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse  
Bus 001 Device 006: ID 0471:1200 Philips (or NXP)  
Bus 001 Device 004: ID 0e0f:0008 VMware, Inc.  
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub  
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub  
root@27f36f541564:/home/lys/lysDemo# █
```

这里用的是USBCAN2设备



## 3、CanAssistant Docker 镜像使用说明

### 第一步，导入镜像

见上面的 Docker 教程。

```
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys# ls
Desktop Documents Downloads lys.x86_64.ubuntu.18.04.canassistant.tar Music Pictures Public snap Te
root@lys-virtual-machine:/home/lys# docker load -i lys.x86_64.ubuntu.18.04.canassistant.tar
548a79621a42: Loading layer [=====>] 65.53MB/65.53MB
dad707c7717d: Loading layer [=====>] 2.048kB/2.048kB
5f70bf18a086: Loading layer [=====>] 1.024kB/1.024kB
405d3062ba2e: Loading layer [=====>] 46.89MB/46.89MB
60091a5505b1: Loading layer [=====>] 6.843MB/6.843MB
4e6fe5af33bc: Loading layer [=====>] 5.632kB/5.632kB
d015e79d93ed: Loading layer [=====>] 45.73MB/45.73MB
3cd3a4acdf47: Loading layer [=====>] 158.3MB/158.3MB
234c6d3ab71c: Loading layer [=====>] 506.1MB/506.1MB
78269a40c441: Loading layer [=====>] 11.26kB/11.26kB
7da7ccbfd88a: Loading layer [=====>] 18.94kB/18.94kB
Loaded image: lys.ubuntu.18.04.canassistant:2025-01-07
root@lys-virtual-machine:/home/lys#
```

```
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys#
root@lys-virtual-machine:/home/lys# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
lys.ubuntu.18.04.canassistant  2025-01-07         74bba10706a0       7 hours ago        807MB
root@lys-virtual-machine:/home/lys#
```

### 第二步，创建并启动容器

#### 方式 1:用宿主机的显示设备

要求 Docker 宿主机是带界面的操作系统，容器里的运行的软件将直接在宿主机上显示。

##### 1.宿主机安装 x11-server

sudo apt-get install x11-xserver-utils

##### 2.宿主机开启 X Server 权限

命令: ( 在宿主机图形终端执行，不要在 ssh 连接的终端执行 )

sudo xhost +



```
TX packets 35056 bytes 2564718 (2.5 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 493 bytes 47396 (47.3 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 493 bytes 47396 (47.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lys@lys-virtual-machine:~/Desktop$
lys@lys-virtual-machine:~/Desktop$
lys@lys-virtual-machine:~/Desktop$
lys@lys-virtual-machine:~/Desktop$
lys@lys-virtual-machine:~/Desktop$ su
Password:
root@lys-virtual-machine:~/Desktop#
root@lys-virtual-machine:~/Desktop#
root@lys-virtual-machine:~/Desktop#
root@lys-virtual-machine:~/Desktop# sudo xhost +
access control disabled, clients can connect from any host
root@lys-virtual-machine:~/Desktop#
```

成功

## 3.启动容器

命令: ( 在宿主机图形终端执行 )

```
docker run -it --rm -e DISPLAY=$DISPLAY \
-v /tmp/.X11-unix:/tmp/.X11-unix \
--device=/dev/bus/usb:/dev/bus/usb \
--privileged lys.ubuntu.18.04.canassistant:2025-01-07 \
/home/lys/start.sh
```

```
lys@lys-virtual-machine:~/Desktop$ su
Password:
root@lys-virtual-machine:~/Desktop#
root@lys-virtual-machine:~/Desktop#
root@lys-virtual-machine:~/Desktop# docker run -it --rm -e DISPLAY=$DISPLAY \
-v /tmp/.X11-unix:/tmp/.X11-unix \
--device=/dev/bus/usb:/dev/bus/usb \
--privileged lys.ubuntu.18.04.canassistant:2025-01-07 \
/home/lys/start.sh
#####Lys CanAssistant#####
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
"log/.."
"log/.."
"log/dd_log"
```





## Docker 使用教程



### 方式 2:通过 VNC 软件连接

要求其他电脑安装 VNC 软件，宿主机可以是不带界面的操作系统，容器里运行的软件将通过 VNC 软件显示，宿主机和安装 VNC 的主机必须在同一个网络里。

#### 1.安装 VNC 软件

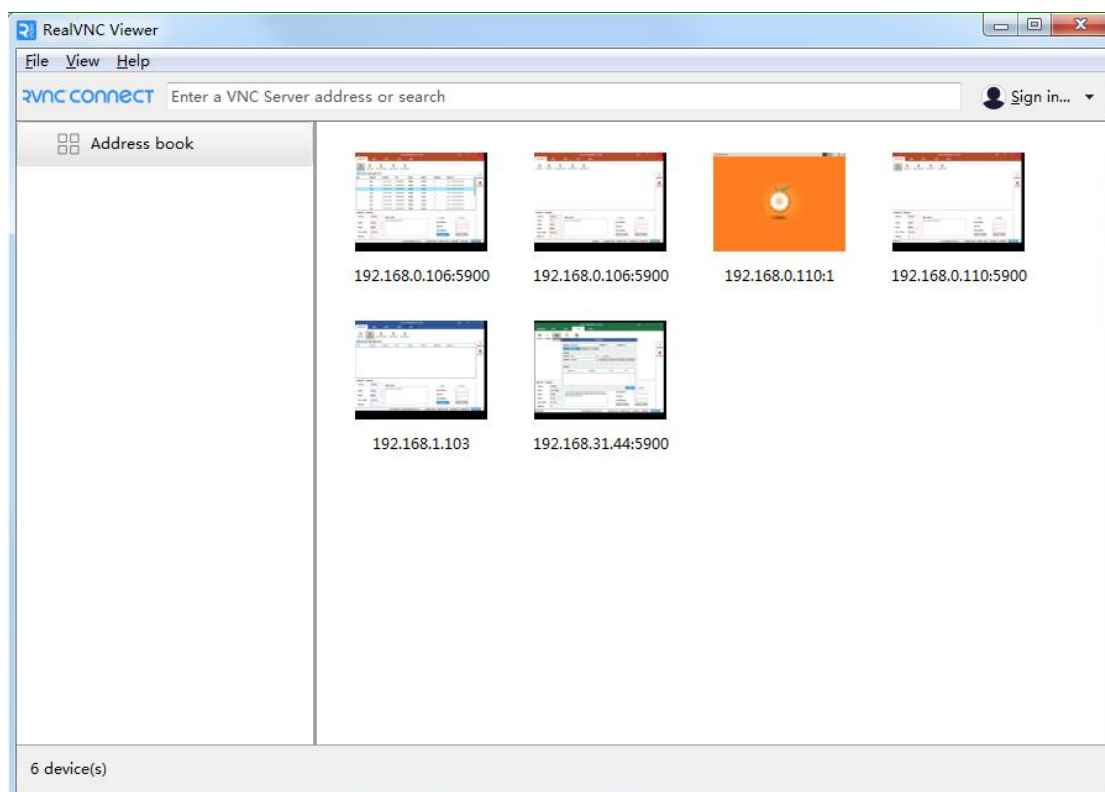
VNC 下载地址:

<https://www.123684.com/s/elmmTd-QuPX>





## Docker 使用教程



## 2. 查看宿主机 IP 地址

命令:

Ifconfig -a

```
root@lys-virtual-machine:/home/lys#  
root@lys-virtual-machine:/home/lys#  
root@lys-virtual-machine:/home/lys# ifconfig -a  
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255  
    inet6 fe80::42:6aff:fe62:dd23 prefixlen 64 scopeid 0x20<link>  
    ether 02:42:6a:62:dd:23 txqueuelen 0 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 31 bytes 4474 (4.4 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.103 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::5a80:7aec:5219:14a9 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:e6:57:37 txqueuelen 1000 (Ethernet)  
    RX packets 1900753 bytes 2797255855 (2.7 GB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 738646 bytes 53560864 (53.5 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 1501 bytes 155770 (155.7 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1501 bytes 155770 (155.7 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

这是演示的宿主机电脑IP，等下VNC软件会需要。

## 3.启动容器

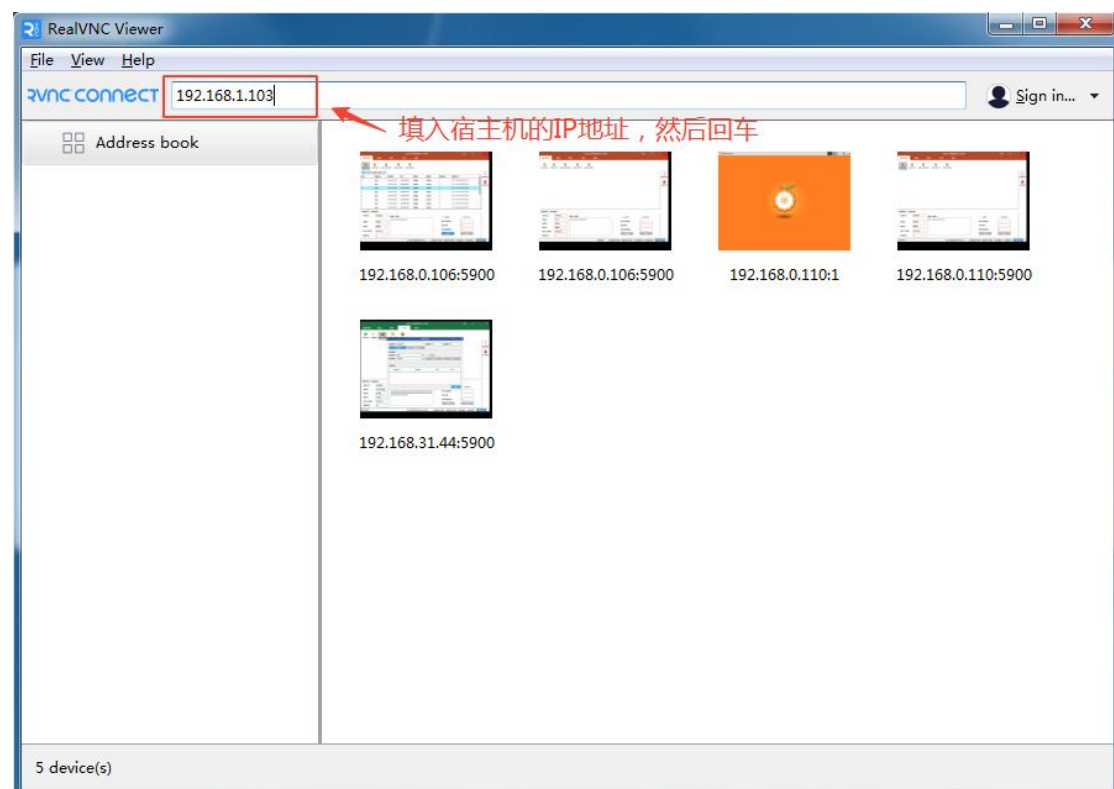
命令:

```
docker run -it --rm -e DISPLAY=$DISPLAY \  
-v /tmp/.X11-unix:/tmp/.X11-unix \  
--device=/dev/bus/usb:/dev/bus/usb \  
-p 5900:5900 \  
--privileged lys.ubuntu.18.04.canassistant:2025-01-07 \  
/home/lys/start.sh vnc
```

```
root@lys-virtual-machine:/home/lys# docker run -it --rm -e DISPLAY=$DISPLAY \  
-v /tmp/.X11-unix:/tmp/.X11-unix \  
--device=/dev/bus/usb:/dev/bus/usb \  
-p 5900:5900 \  
--privileged lys.ubuntu.18.04.canassistant:2025-01-07 \  
/home/lys/start.sh vnc  
#####lys CanAssistant#####  
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'  
"log/."  
"log/.."  
"log/dd.log"  
QVncServer created on port 5900  
█
```

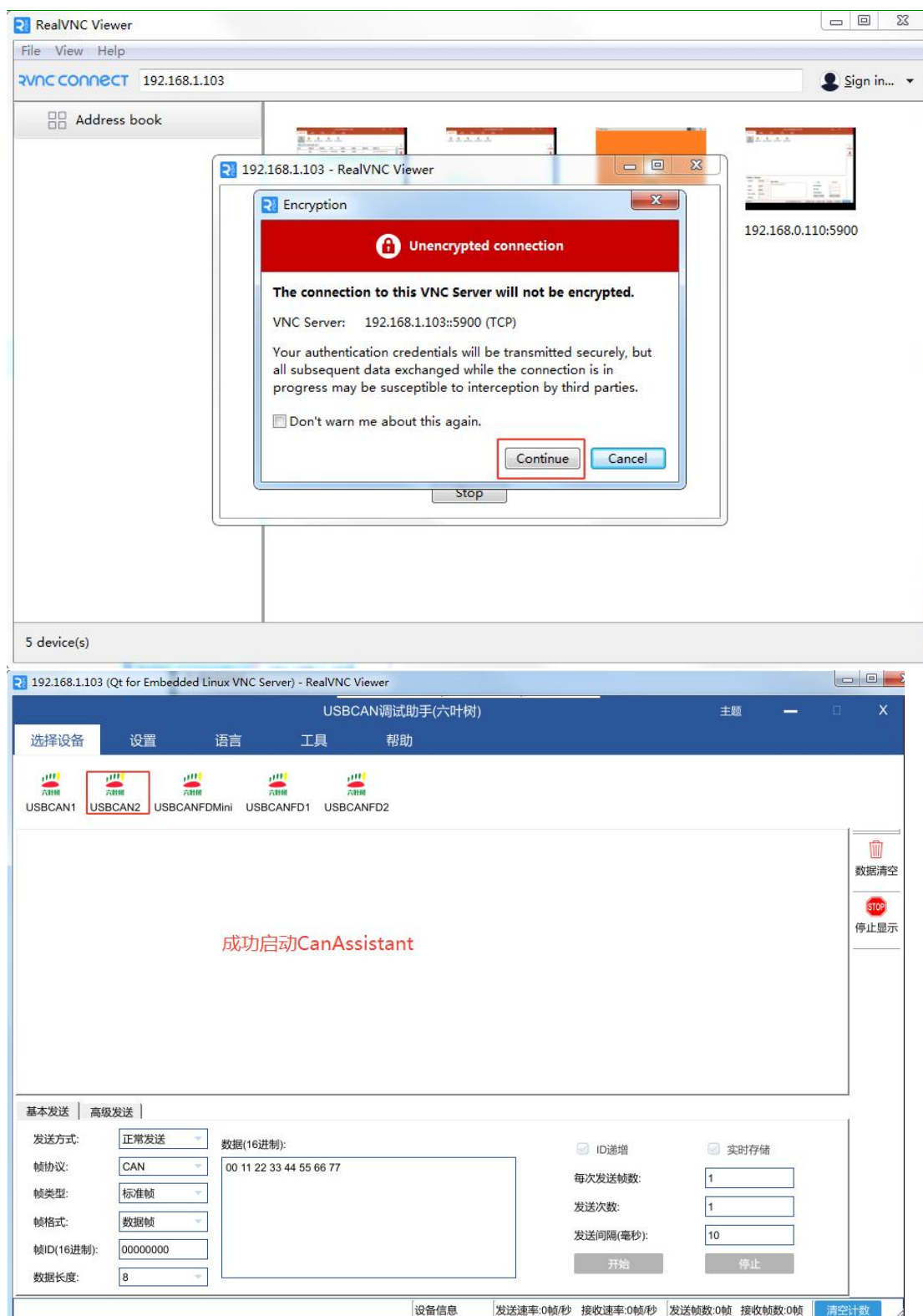
启动成功，下一步通过VNC进行连接

运行 VNC 软件:





## Docker 使用教程



六叶树 USBCAN2 高配版配置界面



## Docker 使用教程

USBCAN调试助手(六叶树)

选择设备 设置 语言 工具 帮助

USBCAN1 USBCAN2 USBCANFMini USBCANFD1 USBCANFD2

USBCAN2 配置

设备索引: 0 通道号: 0

☒ 同时打开全部CAN通道

波特率: 500kbps 定时器0(TIME0): 0x 00

☒ 自定义波特率 定时器1(TIME1): 0x 1C

滤波模式: 单滤波模式 验收码: 0x 00000000

工作模式: 正常 屏蔽码: 0x FFFFFFFF

确定 取消

数据清空

停止显示

基本发送 高级发送

发送方式: 正常发送

帧协议: CAN

帧类型: 标准帧

帧格式: 数据帧

帧ID(16进制): 00000000

数据长度: 8

数据(16进制): 00 11 22 33 44 55 66 77

☒ ID递增 ☒ 实时存储

每次发送帧数: 1

发送次数: 1

发送间隔(毫秒): 10

开始 停止

USBCAN2 设备信息 发送速率: 0帧/秒 接收速率: 0帧/秒 发送帧数: 0帧 接收帧数: 0帧 清空计数

配置成功:

192.168.1.103 (Qt for Embedded Linux VNC Server) - RealVNC Viewer

USBCAN调试助手(六叶树)

选择设备 设置 语言 工具 帮助

USBCAN1 USBCAN2 USBCANFMini USBCANFD1 USBCANFD2

USBCAN2 设备: 0 通道: 0

| 序号 | 传输方向 | 时间标志 | 帧ID | 帧格式 | 帧类型 | 数据长度 | 数据(HEX) |
|----|------|------|-----|-----|-----|------|---------|
|----|------|------|-----|-----|-----|------|---------|

数据清空

停止显示

基本发送 高级发送

发送方式: 正常发送

帧类型: 标准帧

帧格式: 数据帧

帧ID(16进制): 00000000

数据长度: 8

数据(16进制): 00 11 22 33 44 55 66 77

☒ ID递增 ☒ 实时存储

每次发送帧数: 1

发送次数: 1

发送间隔(毫秒): 10

开始 停止

[CAN]中继状态->[关闭][正常模式][500Kbps] 发送速率: 0帧/秒 接收速率: 0帧/秒 发送帧数: 0帧 接收帧数: 0帧 清空计数



### 3 常用链接

六叶树官网:[www.liuyeshu.cn](http://www.liuyeshu.cn)

六叶树 CanAssistant 调试软件下载地址:[http://www.liuyeshu.cn/?page\\_id=492](http://www.liuyeshu.cn/?page_id=492)

六叶树 USBCANFD 产品驱动下载地址:[http://www.liuyeshu.cn/?page\\_id=485](http://www.liuyeshu.cn/?page_id=485)

六叶树 USBCANFD 适配器产品二次开发资料下载:[http://www.liuyeshu.cn/?page\\_id=440](http://www.liuyeshu.cn/?page_id=440)

六叶树 USBCAN 适配器产品 SOCKETCAN 专栏资料下载:[http://www.liuyeshu.cn/?page\\_id=1256](http://www.liuyeshu.cn/?page_id=1256)

六叶树 USBCAN 适配器产品 ROS 机器人操作系统专栏:[http://www.liuyeshu.cn/?page\\_id=1893](http://www.liuyeshu.cn/?page_id=1893)

六叶树 USBCANFD 产品常见软硬件问题解决方案专栏:[http://www.liuyeshu.cn/?page\\_id=593](http://www.liuyeshu.cn/?page_id=593)

### 4 联系方式



电话:15211065817(业务合作咨询)



邮箱:798746621@qq.com(业务咨询+技术支持)



微信:18673379565(技术支持)



官网:[www.liuyeshu.cn](http://www.liuyeshu.cn)(资料下载)



网上商城:<https://shop112408209.taobao.com>(产品购买)

淘宝店铺搜索:“六叶树教育科技有限公司”