



---

# 喵喵派使用手册

---



2024-8-17

目录

喵喵派简介 .....2

喵喵派实物图 .....2

网卡驱动安装 .....3

DEMO 程序的使用 .....7

    USB 网卡连接电脑 .....7

    KMBOX NET 功能设置 .....8

        改端口 .....8

        移动方式 .....9

程序更新 .....10

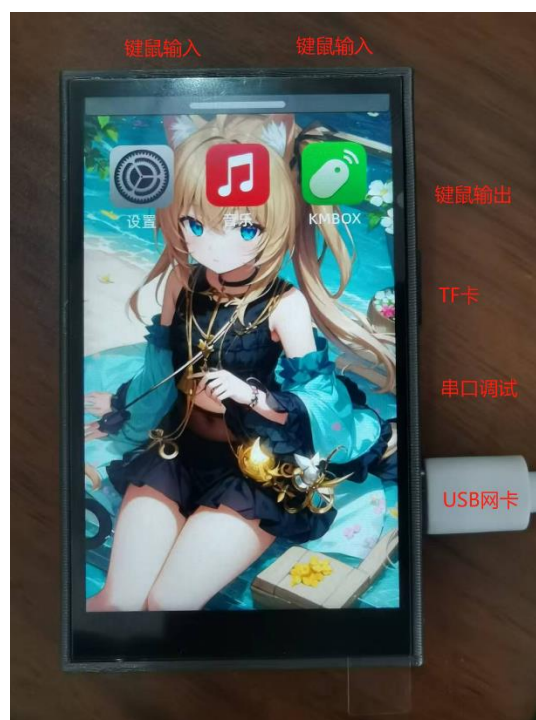
设置 usb 网卡属性 .....12

烧写系统 .....13

# 喵喵派简介

喵喵派是一款小巧且功能强大的 Linux 开发板，搭载了双核 Cortex-A7 处理器和 128MB 的运行内存。该开发板配备了 2 个 USB 主机接口、1 个 HID 键鼠输出、1 个调试串口，以及 1 个 USB 网卡。每个接口均配有自恢复保险丝和静电保护芯片，增强了设备的稳定性和安全性。Linux 系统预装了 OpenCV 等多种常用库，支持 C、C++、Python 等编程语言。开发板还配有 3.1 英寸高清显示屏，适用于 LVGL 和 QT 等用户界面的开发。此外，预装的 DEMO 程序完全开源，包含多个示例，且会不断添加新功能。

## 喵喵派实物图



键鼠输入：USB 设备接口，用于连接 HUB、键盘、鼠标、U 盘、采集卡等设备。

键鼠输出：输出全键无冲键盘和高回报率鼠标信号。

TF 卡槽：用于启动 Linux 系统及存储程序数据。

调试串口：用于系统和程序的调试，一般不常用，也可作为辅助供电接口。

USB 网卡：免驱 USB 网卡，支持模拟为其他 USB 设备。



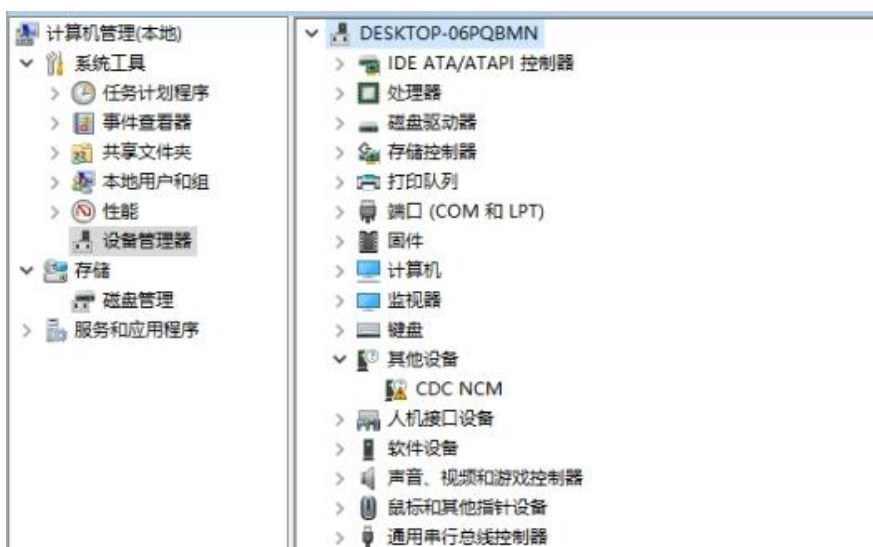
备用按钮：三个备用按钮，用户一般不需要使用。

- **BOOT1**：用于内部 flash 系统的烧写。
- **RES**：系统复位按钮。
- **BOOT2**：用于副芯片系统的烧写

## 网卡驱动安装

开发板网卡驱动 win10, win11 自带，为免驱网卡，在 win11 下插入会自动安装，win10 需要手动安装。以下是 win10 手动安装的教程。

打开设备管理器，在其他设备可以看到 CDC NCM,这个就是我们的网卡。



右键，点击属性。



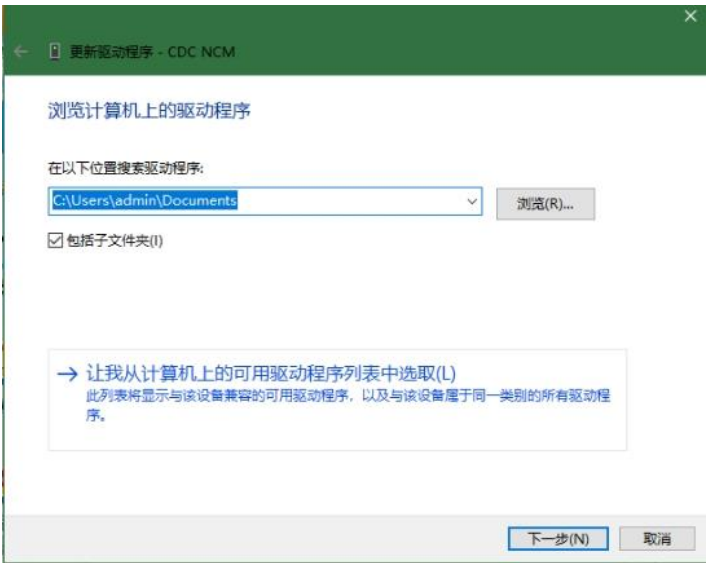
点击 更新驱动程序。



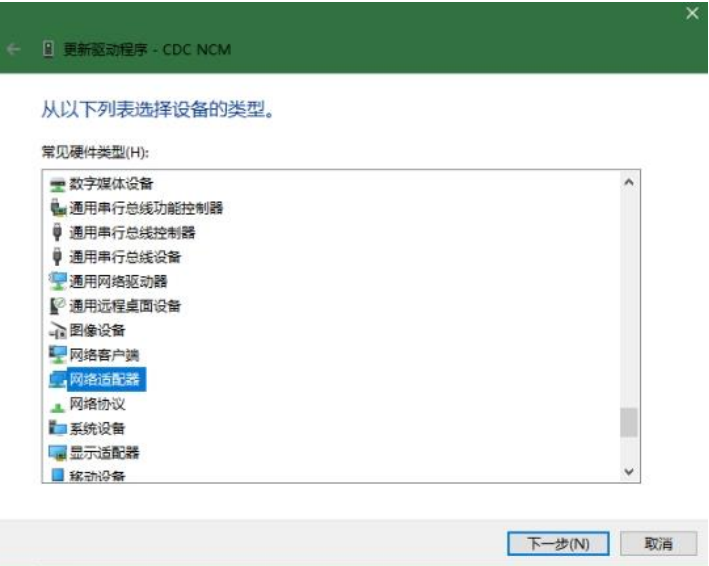
点击 浏览我的电脑以查找驱动程序。



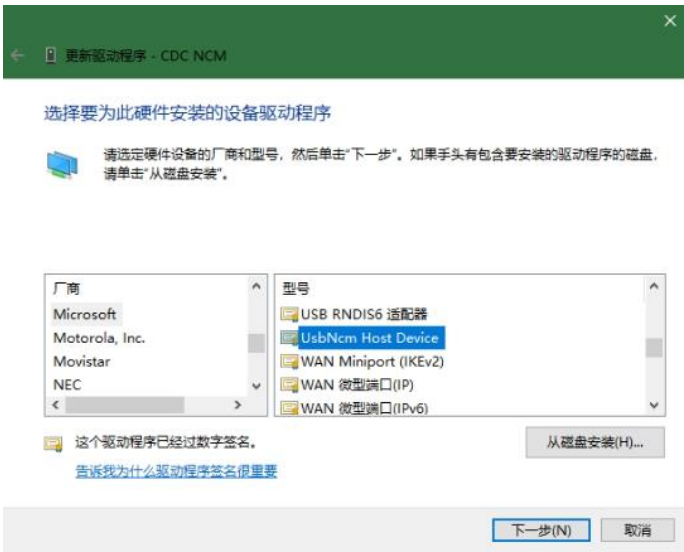
点击让我从计算机上的可用驱动程序列表中选择。



点击网络适配器。



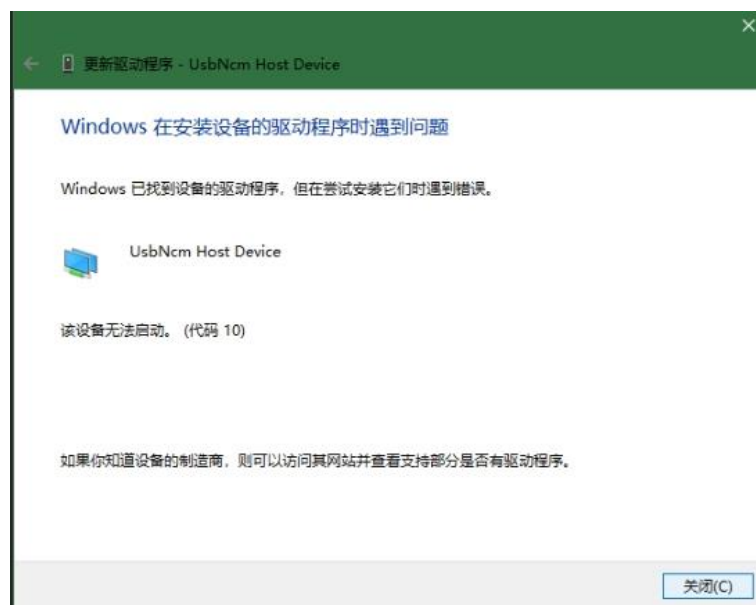
找到 UsbNcm Host Device ， 选中后点击 下一步。



点击 是。



点击关闭。



重新插拔 usb 网卡的数据线。即可安装完成。



# DEMO 程序的使用

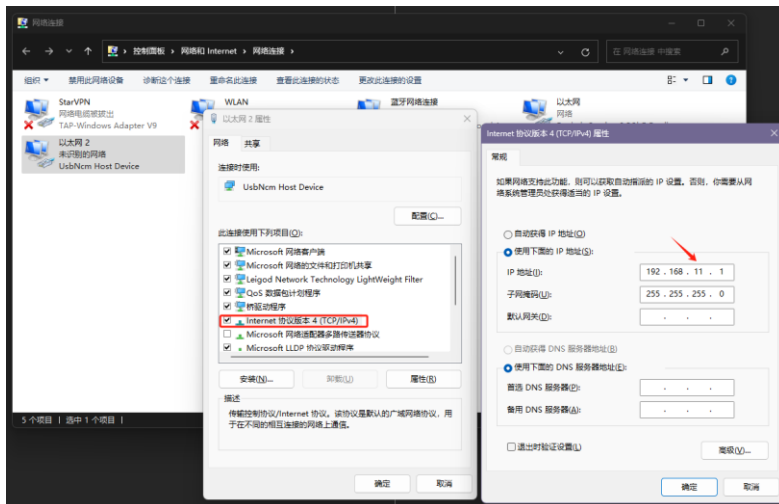
## USB 网卡连接电脑

收到开发板后，首先需要为其分配一个独立的 IP 地址。将 USB 网卡通过数据线连接到电脑，点击“设置 - IP”，进入网卡 IP 设置界面。



在界面中，通过滚动方式调整 IP 地址的最后两位数字，然后点击“确定”完成 IP 设置（注意，必须点击“确定”以保存设置）。

接下来，在电脑的控制面板中，设置网卡的网段，并确保其与开发板的网段相同。



修改完 IP 后，可以使用 ping 命令测试主机与开发板之间的网络连接。



```
Windows PowerShell
版权所有 (C) Microsoft Corporation, 保留所有权利。

安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

PS C:\Users\Wing> ping 192.168.11.5

正在 Ping 192.168.11.5 具有 32 字节的数据:
来自 192.168.11.5 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.11.5 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.11.5 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.11.5 的回复: 字节=32 时间<1ms TTL=64

192.168.11.5 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 1ms, 平均 = 0ms
PS C:\Users\Wing> |
```

当 ping 测试成功后，即可开始使用开发板的各项功能。

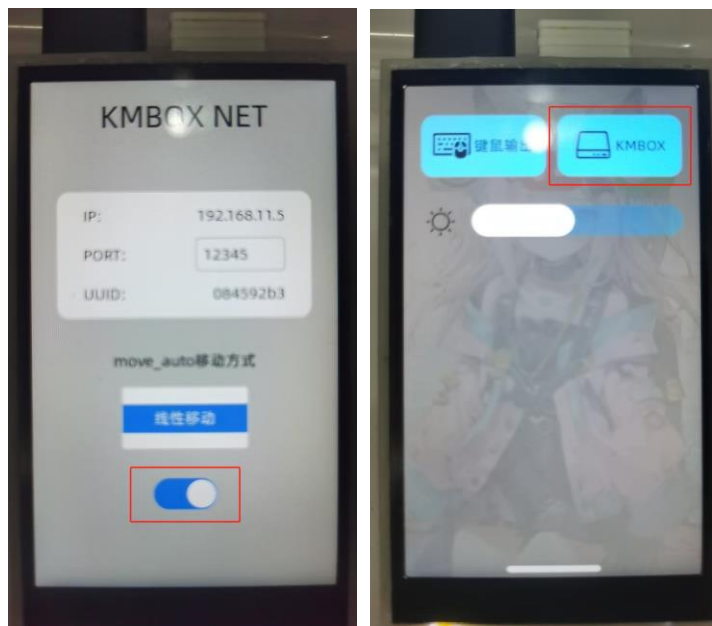
## KMBOX NET 功能设置

### 改端口

点击 KMBOX 图标进入 KMBOX 设置页面。然后，在端口输入框中输入所需的端口号（范围为 1 到 65536）。



端口设置完成后，需要重启 KMBOX 功能以使更改生效。可以通过 KMBOX 页面的开关或主界面下拉框中的快捷开关来完成此操作。

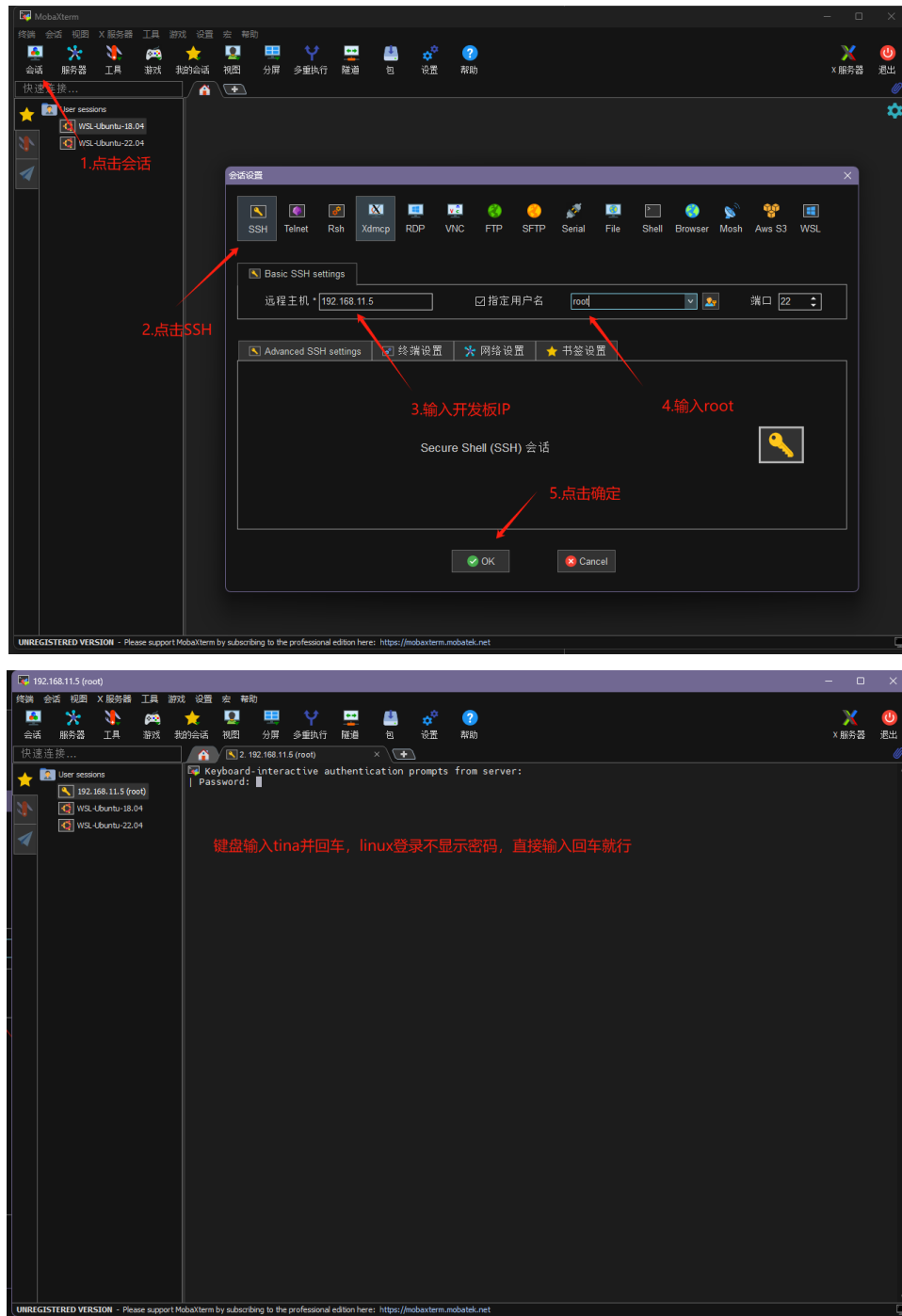


## 移动方式

目前，Move\_auto API 仅支持线性移动方式。未来的版本将会添加更多移动方式，敬请期待。

# 程序更新

打开 MobaXterm 软件并连接到开发板。



```

Mon Feb 6 00:00:00 GMT 2023
nodev debugfs
mount: mounting none on /sys/kernel/debug failed: Resource busy
root@TinaLinux:~# top
Mem: 39100K used, 78700K free, 0K shrd, 3004K buff, 11704K cached
CPU: 18% usr 0% sys 0% nic 81% idle 0% io 0% irq 0% irq
Load average: 0.00 0.00 0.00 1/58 499

```

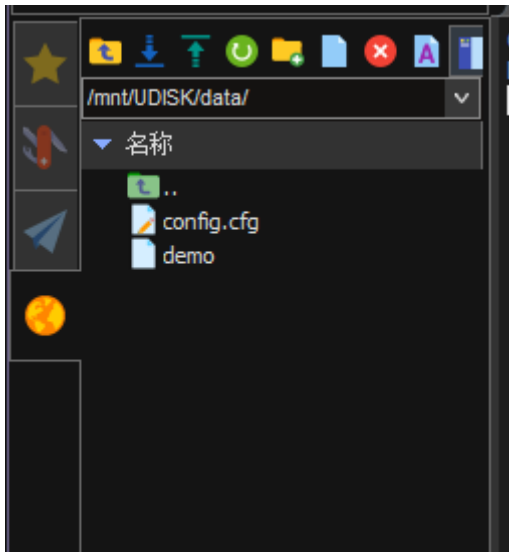
PID	PPID	USER	STAT	VSZ	%VSZ	%CPU	COMMAND
156	1	root	S	23756	20%	9%	/mnt/UDISK/data/demo
497	165	root	R	1024	1%	3%	top
179	155	root	S	3944	3%	0%	sshd: root@pts/0
183	155	root	S	3632	3%	0%	sshd: root@tty
155	1	root	S	2852	2%	0%	/usr/sbin/sshd
198	197	root	S	2176	2%	0%	/usr/lib/sftp-server
1	0	root	S	1020	1%	0%	/sbin/init
193	179	root	S	1020	1%	0%	ash -c while true; do sleep 1; head -v -n 8 /proc/meminfo; head -v -n 2 /proc/stat
158	1	root	S	1020	1%	0%	/bin/sh
185	179	root	S	1020	1%	0%	-ash
197	183	root	S	1020	1%	0%	ash -c /usr/lib/sftp-server
499	193	root	S	1020	1%	0%	sleep 1
9	2	root	SW	0	0%	0%	[ksoftirqd/0]
67	2	root	IW	0	0%	0%	[kworker/0:3-eve]
20	2	root	IW	0	0%	0%	[kworker/0:1-eve]
10	2	root	IW	0	0%	0%	[rcu_preempt]
54	2	root	SW	0	0%	0%	[spi0]
73	2	root	SW	0	0%	0%	[irq/54-ft6236]
64	2	root	SW	0	0%	0%	[irq/38-mmc0]
72	2	root	IW<	0	0%	0%	[kworker/0:1H-kb]
7	2	root	IW	0	0%	0%	[kworker/u4:0-ev]
52	2	root	IW	0	0%	0%	[kworker/u4:1-ev]
50	2	root	IW<	0	0%	0%	[kblockd]
23	2	root	SW	0	0%	0%	[kcompactd0]
51	2	root	SW	0	0%	0%	[ion_system_heap]
21	2	root	SW	0	0%	0%	[oom_reaper]
19	2	root	SW	0	0%	0%	[rcu_tasks_kthre]
18	2	root	SW	0	0%	0%	[kdevtmpfs]
22	2	root	IW<	0	0%	0%	[writeback]
15	2	root	PW	0	0%	0%	[ksoftirqd/1]
58	2	root	SW	0	0%	0%	[vsync_proc_1]
66	2	root	SW	0	0%	0%	[irq/39-mmc1]
65	2	root	SW	0	0%	0%	[irq/196-4020000]
55	2	root	SW	0	0%	0%	[kswapd0]
57	2	root	SW	0	0%	0%	[vsync_proc_0]
68	2	root	IW<	0	0%	0%	[ipv6_addrconf]
70	2	root	IW<	0	0%	0%	[mmc_complete]
69	2	root	SW<	0	0%	0%	[krfcomm]
75	2	root	IW<	0	0%	0%	[kworker/0:2H]
98	2	root	SW	0	0%	0%	[bd2/mmcblkp7-]

输入命令 top 查看demo的线程号

记住自己开发板demo的PID，我这里是156

按下 Ctrl + C 终止 top 命令的输出，然后输入命令 kill 156（将 156 替换为实际的进程 ID）。

在左侧的 SFTP 窗口中，导航至/mnt/UDISK/data/目录。



这是 DEMO 程序的存放目录。将新的 DEMO 程序文件拖入该目录，即可完成更新。要运行新的 DEMO 程序，可以直接断电重启开发板，或在命令行中输入以下命令：

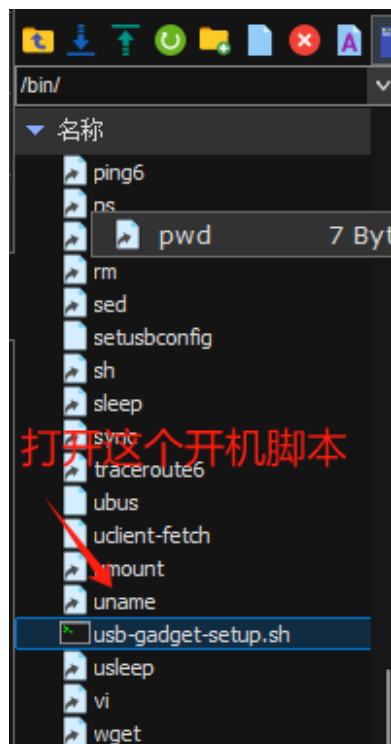
```

cd /mnt/UDISK/data/
./demo

```

程序将会自动启动。

# 设置 usb 网卡属性



此部分设置包括了与开机运行相关的内容，包含设置 USB 网卡的各种参数，SSH 的自启，demo 程序的自启。

```
usb-gadget-setup.sh
46 # 创建 USB Gadget 配置目录
47 mkdir -p /sys/kernel/config/usb_gadget/g1
48
49 # 设置 USB Gadget 的基本属性
50 echo 0x1d6b > /sys/kernel/config/usb_gadget/g1/idVendor # Linux Foundation
51 echo 0x0104 > /sys/kernel/config/usb_gadget/g1/idProduct # Multifunction Composite Gadget
52
53 # 设置设备描述符字符串
54 mkdir -p /sys/kernel/config/usb_gadget/g1/strings/0x409
55 echo "fedcba9876543210" > /sys/kernel/config/usb_gadget/g1/strings/0x409/serialnumber
56 echo "My Manufacturer" > /sys/kernel/config/usb_gadget/g1/strings/0x409/manufacturer
57 echo "My USB Device" > /sys/kernel/config/usb_gadget/g1/strings/0x409/product
58
59 # 创建 NCM 功能
60 mkdir -p /sys/kernel/config/usb_gadget/g1/functions/ncm.usb0
61 mkdir -p /sys/kernel/config/usb_gadget/g1/configs/c.1
62 mkdir -p /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
63 echo "Config 1" > /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409/configuration
64
65 # 绑定 NCM 功能到配置
66 ln -s /sys/kernel/config/usb_gadget/g1/functions/ncm.usb0 /sys/kernel/config/usb_gadget/g1/configs/c.1
67
68 # 绑定 UDC
69 echo "$(ls /sys/class/udc)" > /sys/kernel/config/usb_gadget/g1/UDC
70
71 # 启用 usb0 接口
72 ifconfig usb0 192.168.7.2 netmask 255.255.255.0 up
73
74 echo "USB gadget setup complete."
75
76
```

# 烧写系统

如果需要重新烧写系统，请准备一个 TF 卡读卡器。通常情况下，开发板出厂时已经预装了最新的系统，因此用户通常无需自行烧写。

如需烧写，请按照以下步骤操作：

1. 打开 **PhoenixCard** 软件。
2. 将 TF 卡插入读卡器，并将读卡器连接至电脑。
3. 在 PhoenixCard 软件中，点击“**固件**”，然后选择要烧写的系统镜像文件。
4. 点击“**启动卡**”以设置 TF 卡为启动盘。
5. 最后，点击“**烧卡**”开始烧写过程。

烧写完成后，您可以将 TF 卡插入开发板，正常启动系统。

