



Apoidea xHiveAI RK3588 Box 使用手册

2025 年 02 月

群蜂信息技术（上海）有限公司

文档修订历史

版本	修改描述	修改日期	变更人
V0.1	初稿	2024-03-19	Apoidea
V0.2	添加程序开发指南	2024-03-29	Apoidea
V0.3	统一文档的接口描述	2024-05-08	Apoidea
V0.4	增加双网口配置	2024-07-31	Apoidea
V0.5	更新 otg 烧写固件内容	2024-08-23	Apoidea
V0.6	完善更新 otg 烧写固件的步骤	2024-11-18	Apoidea
V0.7	添加盒子调试技巧	2024-11-19	Apoidea
V0.8	添加第二章：硬件接口使用	2025-01-14	Apoidea
V0.9	添加 emmc flash 读写的监控和保护逻辑	2025-02-18	Apoidea
V1.0	添加硬件功能使用和软件主要功能说明	2025-04-01	Apoidea

前言

本文档是群蜂信息技术（上海）有限公司开发的 xHiveAI RK3588 Box 使用手册。
本文档描述了客户在收到盒子后如何连接和使用 RK3588 盒子。

本手册是供专业技术人员、运维人员、产品经理、系统集成商参考研究，本产品应由经验丰富的技术人员进行安装和维护。

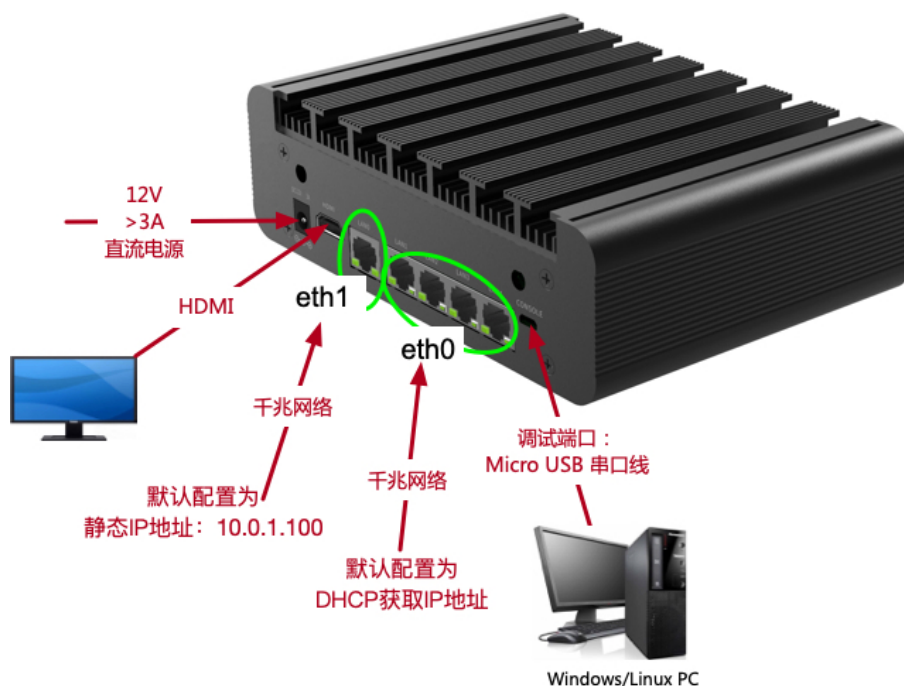
目录

1 盒子配置步骤	5
1.1 第一步：启动进入图形界面	5
1.2 第二步：配置盒子的 IP, Netmask 和 DNS 地址	6
1.2.1 配置静态 IP 地址	6
1.2.2 使用 DHCP 动态获取 IP 地址	7
1.2.3 修改 Mac 地址	8
1.2.4 配置双网卡	8
1.2.5 从双网卡配置恢复出厂网络配置	9
1.3 第三步：配置虚拟桌面，断开显示器连接	9
1.4 故障调试：恢复显示器的图形界面显示	10
1.5 程序开发指南	10
1.5.1 存储磁盘的使用	10
1.5.2 emmc flash 的读写监控	11
2 硬件接口的使用	11
2.1 DI 和 DO 接口的使用	11
2.2 STATUS led 灯控制	12
2.3 安装 M.2 SSD 存储卡	13
2.4 安装 Mini TF 存储卡	13
2.5 安装 4G PCIe 卡	13
3 软件开发指南	15
3.1 AI 大模型算法	15
3.2 支持硬件编解码功能的 ffmpeg	15
3.3 Opencv: 支持 ffmpeg, gstreamer, java 和 python 调用	16
3.4 VLC Player	16
3.5 内置 lightdm and weston UI 系统	16
4 Q&A	17
4.1 Docker 启动失败	17
4.2 是否有方法升级或者重置盒子的系统?	17
5 附录	17
5.1 使用调试串口诊断问题盒子	17
5.2 通过 ssh 连接	19
5.3 使用扩展 Nvme SSD Flash	19
5.4 使用 OTG 接口重烧或者升级固件	19
5.5 调试盒子的技巧	20
5.5.1 快速更新 kernel, dtb 或者 modules 文件	20

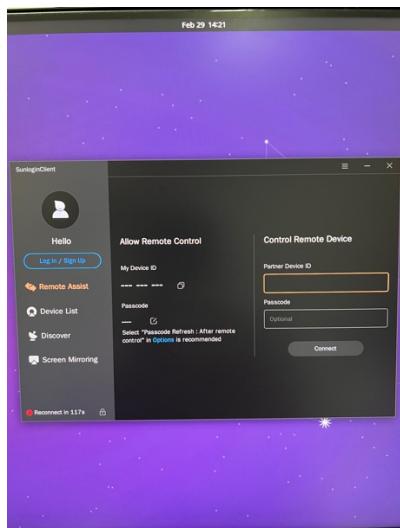
1 盒子配置步骤

客户在对盒子第一次上电启动前，请按照下图连接 HDMI LCD 显示器，usb 鼠标和 usb 键盘。如果配置静态 IP 地址，网线请插入 eth1 接口。如果使用 DHCP 方式自动获取 IP 地址方式，网线请插入 eth0 四个接口中的任意一个。

⚠ 切勿将 eth0 和 eth1 同时接入同一网络，会因为 IP 地址冲突而导致盒子死机！！



1.1 第一步：启动进入图形界面

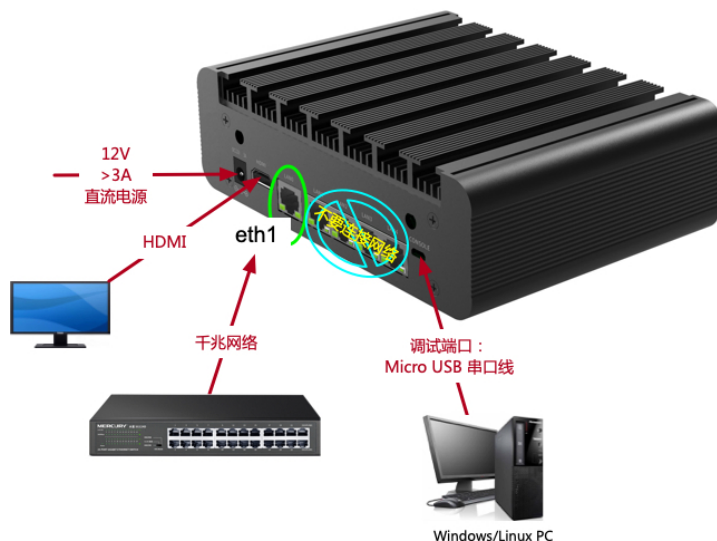


《向日葵》控制软件会默认启动，但是因为没有配置合适的 IP 地址，所以盒子无法连接 internet。结果《向日葵》软件的“My Device ID”和“Passcode”都为空，这是正常的。

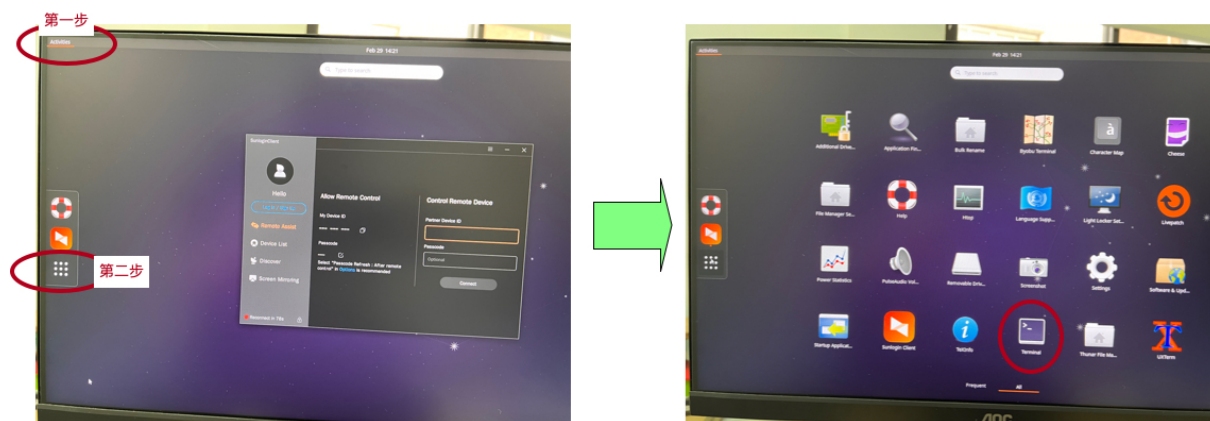
1.2 第二步：配置盒子的 IP，Netmask 和 DNS 地址

1.2.1 配置静态 IP 地址

按照下图连接盒子的网络：



按照下面操作进入命令行执行界面：



执行“Terminal”应用程序，进入命令行执行界面

执行如下命令修改网络配置：

```
sudo vi /etc/netplan/eth1-netcfg.yaml
```

按照下图修改对应的网络参数：

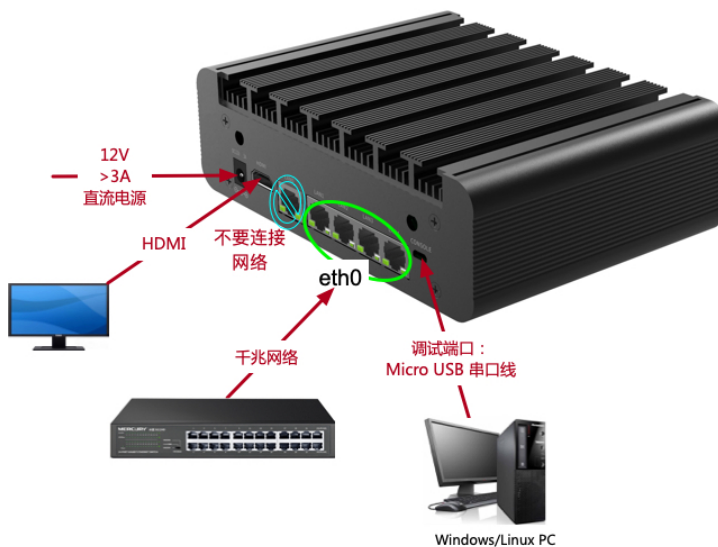
```
network:
  version: '2'
  renderer: networkd
  ethernets:
    eth1:
      dhcp4: no
      addresses:
        - 10.0.1.100/24 ← 修改静态IP地址和Netmask
```

```
macaddress: 88:5F:E8:10:03:00 ← 不要手动修改, 否则可能会导致盒子启动失败
gateway4: 10.0.1.1 ← 修改网关IP地址
nameservers:
  addresses:
    - 8.8.8.8 ← 修改或者添加DNS Server IP
    - 10.0.1.1 ← 修改或者添加DNS Server IP
```

保存文件后, 重启盒子后, 用 `ifconfig` 来验证 `eth1` 配置的 IP 地址是否生效。

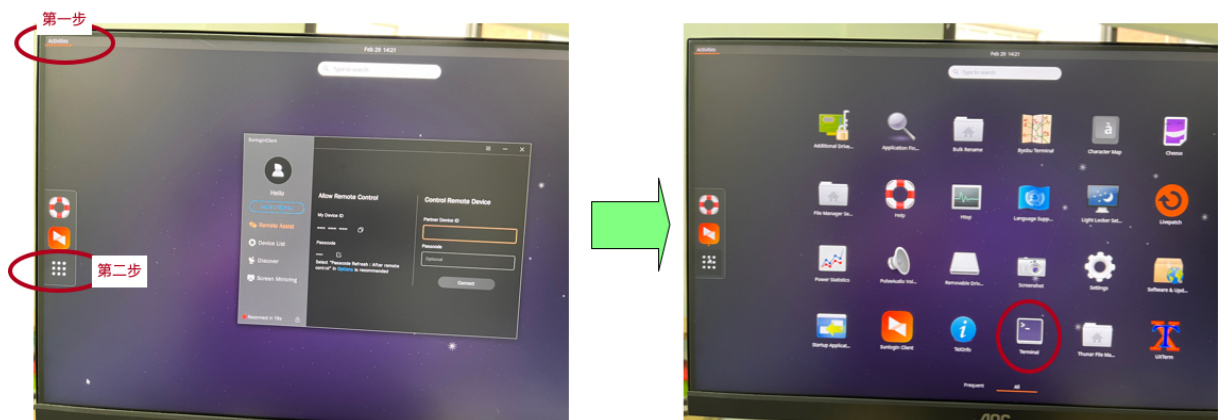
1.2.2 使用 DHCP 动态获取 IP 地址

按照下图连接盒子的网络：



连接好盒子后, 上电启动盒子, 应该可以正常通过 DHCP 获取 IP 地址。

按照下面操作进入命令行执行界面：



执行 “Terminal” 应用程序, 进入命令行执行界面, 用 `ifconfig` 来验证 `eth0` 配置的 IP 地址是否生效。

可以执行以下命令来修改 `eth0` 的 ip 配置：

```
sudo vi /etc/netplan/eth0-netcfg.yaml
```

1.2.3 修改 Mac 地址

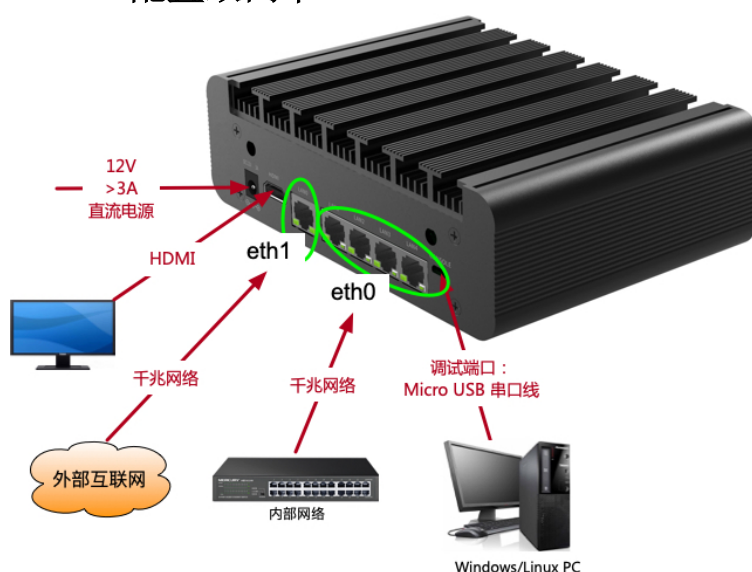
盒子出厂的时候，会默认配置一个 mac 地址，且 eth0 和 eth1 是一样的，所以 eth0 和 eth1 不能同时使用。如果项目需要同时使用 eth0 和 eth1，客户可以自行配置 mac 地址，和获取 ip 地址的方式。

配置 mac 地址的方式如下：

```
echo xx:xx:xx:xx:xx:xx > /userdata/persist/eth0_mac  
echo xx:xx:xx:xx:xx:xx > /userdata/persist/eth1_mac  
sync
```

重启盒子后，用 ifconfig 查看新设置的 mac address。

1.2.4 配置双网卡



按照上图连接内外网。按照以下步骤来修改网络配置：

第一步：保持 eth1 的网络配置不变

第二步：备份 eth0 配置文件

```
sudo cp /etc/netplan/eth0-netcfg.yaml /etc/netplan/eth0-netcfg.yaml.orig
```

第三步：按照实际的网络环境配置 eth0

【配置 static ip address】

```
sudo vi /etc/netplan/eth0-netcfg.yaml
```

按照下图修改对应的网络参数（以下为例配置）：

```
network:  
  version: '2'  
  renderer: networkd  
  ethernets:  
    eth0:
```

```

dhcp4: no
addresses:
  - 10.0.1.100/24 ← 修改静态IP地址和Netmask
macaddress: 88:5F:E8:10:03:00 ← 修改Mac地址, 可以简单+1, 修改成88:5F:E8:10:03:01
optional: yes
gateway4: 10.0.1.1 ← 修改网关IP地址
nameservers:
  addresses:
    - 8.8.8.8 ← 修改或者添加DNS Server IP
    - 10.0.1.1 ← 修改或者添加 DNS Server IP

```

【配置 dhcp】

```
sudo vi /etc/netplan/eth0-netcfg.yaml
```

按照下图修改对应的网络参数（以下为例配置）：

```

---
network:
  ethernets:
    eth0:
      dhcp4: yes
      macaddress: 88:5F:E8:10:05:8C ← 修改Mac地址, 可以简单+1, 修改成88:5F:E8:10:05:8D
      optional: yes
      renderer: networkd
      version: '2'

```

第四步 : `rm -f /userdata/persist/eth0_mac`

第五步 : `reboot`

1.2.5 从双网卡配置恢复出厂网络配置

按照以下步骤来恢复网络的出厂配置：

第一步 : 保持 eth1 的网络配置不变

第二步 : 恢复 eth0 配置文件

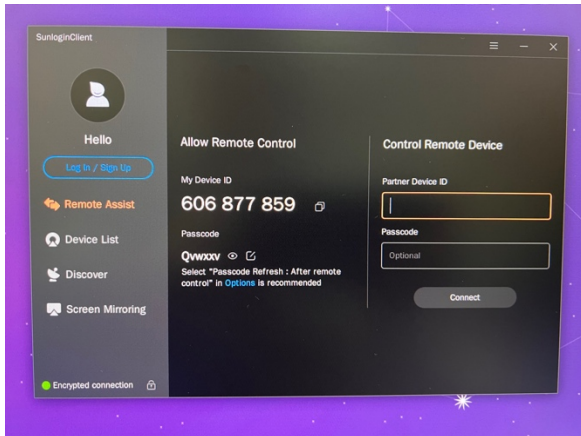
```
sudo mv /etc/netplan/eth0-netcfg.yaml.orig /etc/netplan/eth0-netcfg.yaml
```

第三步 : `cp /userdata/persist/eth1_mac /userdata/persist/eth0_mac`

第四步 : `reboot`

1.3 第三步：配置虚拟桌面，断开显示器连接

重启进入系统后，《向日葵》软件应该能够获取到正确的“My Device ID”和“Passcode”，**请务必记录下来，以后每次连接都使用这个 ID 和密码**。例子如下：



问题#1: 如果仍然拿不到正确的“My Device ID”和“Passcode”，请测试网络是否可以访问 internet，在命令行上执行：ping www.baidu.com，应该可以 ping 通，否则检查网络参数设置和连接（譬如，是否在公司的防火墙上开通盒子访问外网的权限）。

问题#2: 如果在可以访问外部 internet 的情况下仍然无法获得“My Device ID”和“Passcode”，可以先执行命令：pkill -9 -f sunlogin，强行关闭向日葵软件，然后再重启盒子。

这个时候，盒子必须要连接显示器，远程的《向日葵》控制软件才能正常显示桌面，如果断开显示器连接，就会显示黑屏。

按照下面步骤来启动虚拟桌面：

第一步：连接 usb 键盘

第二步：同时按下 **Ctrl + Shift + F1** 键

第三步：盒子会重新启动，如果连接显示器，屏幕上不会再有图形界面的显示。这个时候可以安全断开显示器连接。《向日葵》软件仍然可以正常控制。

1.4 故障调试：恢复显示器的图形界面显示

在某些情况下，盒子出现故障或者向日葵也无法远程访问，可以通过下面方法来恢复盒子在显示器上的桌面显示，以便进行故障排除：

第一步：连接 HDMI 显示器，并且连接 usb 键盘

第二步：同时按下 **Ctrl + Shift + F2** 键

第三步：盒子会重新启动，图形桌面会在显示器上正常显示

1.5 程序开发指南

1.5.1 存储磁盘的使用

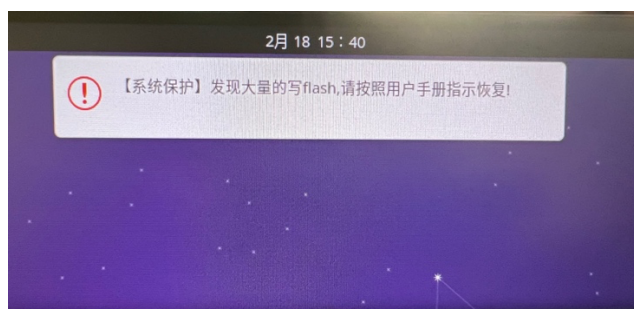
盒子有两个主要的存储设备，一个是安装了 Ubuntu 操作系统的根目录文件系统，另外一个是在 `/data` 下的外接 SSD Flash 设备。在软件开发过程中，**尽量将大容量的数据文件放在 `/data` 目录下面**，以免系统盘容量耗尽后，可能会导致的各种系统异常。

1.5.2 EMMC FLASH 的读写监控

因为 emmc flash 的写入是有寿命限制的，频繁的大量写入会导致 emmc flash 的损坏，最终导致盒子死机，无法启动。如果发生该情况，只有返厂修理，而且不属于保修范围。所以用户务必谨慎地规划应用程序的写入量，以期延长 emmc flash 的使用寿命。

请遵守以下的编程规范：

- 1) 对于需要大量，频繁写入的临时数据操作，请使用/run 目录。该目录 mount 的是运行在内存中的 tmpfs 文件系统，大量的读写不会导致盒子的物理损耗。另外由于数据的读写是在内存中，速度也很快。但是这个文件系统只有大约 750MB，请及时删除旧数据，防止使用过量而导致的系统崩溃！另外，该目录下写入的数据由于是在内存中的，盒子重启会导致数据丢失！
- 2) 对于需要长时间保持的数据，可以写入/data 或者/userdata 目录中，这些数据可以长久保存，并且不会因为盒子断电或重启而丢失。但是注意不要频繁，大量的写入。通常我们设定的写入上限是 500MB/小时。
- 3) 盒子启动后，emmc 的数据监控程序会每小时统计 emmc flash 的写入量。如果任意连续的 6 小时，每个小时的写入量都超过了 500MB，那么盒子会进入保护模式，在 UI 上显示告警消息：



并且用户串口或者 ssh 登录的界面上会显示：

来自 root@xhivaei (pts/0) (Tue Feb 18 15:40:37 2025) 的广播消息：

【系统保护】发现大量的写 flash,请按照用户手册指示恢复！

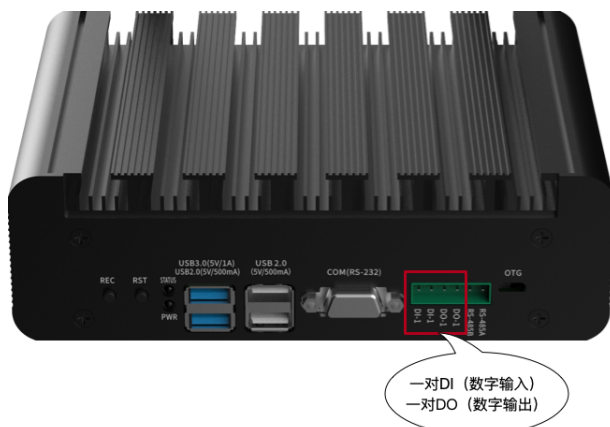
这个时候，整个/data 或者/userdata 会进入写保护状态，用户程序无法继续写入任何数据，但是读操作可以继续进行。请检查应用程序，彻底解决大量的写 emmc flash 问题后，执行如下命令恢复系统：

```
protect_system.sh stop
```

该命令执行完成后，emmc flash 的写入会恢复正常。监控程序会继续监控 flash 的写入量，如果再次达到上述标准后，系统会再次被保护！

2 硬件接口的使用

2.1 DI 和 DO 接口的使用



【DO 的配置和使用】

DO 开关可以通过执行下面代码，来控制外面电路的连通和断开。

登陆盒子后

```
cd /sys/class/gpio
echo 121 > export
cd gpio121
echo out > direction
echo 1 > value //断开连接
echo 0 > value //接通连接
```

【DI 的配置和使用】

DI 可以接收输入的数字信号：1 或者 0

登陆盒子后

```
cd /sys/class/gpio
echo 24 > export
cd gpio24
cat direction
in
cat value
1: 表示外部电路断开
0: 表示外部电路连通
```

2.2 STATUS led 灯控制

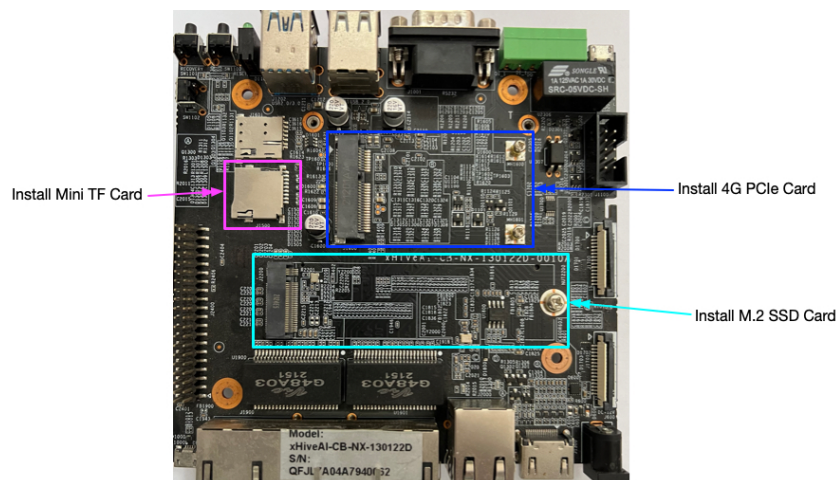
在盒子前面板有一个"STATUS" led 灯，应用程序可以控制其显示：

登陆盒子后

```
cd /sys/class/gpio/
echo 105 > export
echo out > direction
echo 1 > value //点亮 led 灯
```

```
echo 0 > value //熄灭 led 灯
```

2.3 安装 M.2 SSD 存储卡



拆开盒子外壳后，可以在底板上安装 M.2 ssd 存储卡。

系统启动后，执行命令：

```
lsblk
```

```
nvme0n1    259:0    0 931.5G 0 disk /data
```

会看到/dev/nvme0n1 设备，在系统启动的时候自动 mount 到/data 目录下。

2.4 安装 Mini TF 存储卡

拆开盒子外壳后，可以在底板上插入 mini tf 存储卡。

系统启动后，执行命令：

```
lsblk
```

```
mmcblk1    179:96   0 476.7G 0 disk
└─mmcblk1p1 179:97   0 476.7G 0 part
```

会看到/dev/mmcblk1 设备，应用程序可以根据需要使用它。

2.5 安装 4G PCIe 卡

拆开盒子外壳后，可以在底板上插入 PCIe 硬件接口的 4G 模块。

下面例子中使用的是 Quectel EC20F 4G 模块。

系统启动后，首先查看安装的 4G 模块是否被正确识别到：

```
root@xHiveAIBox:~# dmesg | grep ttyUSB*
```

```
[6.197286] usb 1-3.4: GSM modem (1-port) converter now attached to ttyUSB0
[6.198385] usb 1-3.4: GSM modem (1-port) converter now attached to ttyUSB1
```

```
[6.200094] usb 1-3.4: GSM modem (1-port) converter now attached to ttyUSB2
[6.204078] usb 1-3.4: GSM modem (1-port) converter now attached to ttyUSB3
```

实验中的这款硬件使用的是/dev/ttyUSB2

AT 命令测试：

1、测试 AT 是否正常

```
root@xHiveAIBox:~#echo -e "AT\r\n" >/dev/ttyUSB2
Quectel
EC20F
Revision: EC20CEHDLGR06A07M1G
OK
```

2、获取模块型号

```
root@xHiveAIBox:~#echo -e "AT+CGMM\r\n" > /dev/ttyUSB2
EC20F
OK
```

3、获取厂商信息

```
root@xHiveAIBox:~#echo -e "AT+CGMI\r\n" >/dev/ttyUSB2
Quectel
OK
```

4、查询 SIM 卡的 ICCID

```
root@xHiveAIBox:~#echo -e "AT+QCCID\r\n" >/dev/ttyUSB2
+QCCID: 898600810914F4052657
OK
```

5、查询信号强度

```
root@xHiveAIBox:~#echo -e "AT+CSQ\r\n" >/dev/ttyUSB2
root@xHiveAIBox:~#
+CSQ: 13,99
OK
```

6、查询网络状态

```
root@xHiveAIBox:~#echo -e "AT+QNWINFO\r\n" >/dev/ttyUSB2
+QNWINFO: "TDD LTE","46000","LTE BAND 41",41134
OK
```

7、查 SIM 卡运营商

```
root@xHiveAIBox:~# echo -e "AT+COPS?\r\n" >/dev/ttyUSB2
+COPS: 0,0,"CHINA MOBILE",7
OK
```

8、开启 SIM 卡检测功能

```
root@xHiveAIBox:~# echo -e "AT+QSIMDET=1,1\r\n" >/dev/ttyUSB2
OK
```

9、电话打进的信息提示

```
root@xHiveAIBox:~#
+CIEV: 7,1
+CRING: VOICE
+CLIP: "173XXXX2397",128,,0
```

获取 ttyUSB2 上接收到的信息

```
cat /dev/ttyUSB2 &
```

3 软件开发指南

3.1 AI 大模型算法

最新的软件版本支持包括 Deepseek, LLAMA, Qwen 等大模型 AI 算法。查看 NPU 的版本，如果是 0.9.8 或以上就表示支持，否则需要升级软件后支持：

```
cat /sys/kernel/debug/rknpu/version
RKNPU driver: v0.9.8
```

3.2 支持硬件编解码功能的 ffmpeg

盒子中适配的 ffmpeg 完全支持 rk3588 的视频编解码功能，通过如下命令查看：

```
ffmpeg configure -codecs | grep rkmpp
```

DEV.LS h264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (decoders: h264 h264_v4l2m2m h264_rkmpp) (encoders: h264_v4l2m2m)

DEV.L. hevc H.265 / HEVC (High Efficiency Video Coding) (decoders: hevc hevc_rkmpp hevc_v4l2m2m) (encoders: hevc_v4l2m2m)

DEV.L. vp8 On2 VP8 (decoders: vp8 vp8_rkmpp vp8_v4l2m2m) (encoders: vp8_v4l2m2m)

D.V.L. vp9 Google VP9 (decoders: vp9 vp9_rkmpp vp9_v4l2m2m)

1) 测试 rkmpp 的视频 decoder

```
ffmpeg -c:v h264_rkmpp -i <h.264 编码的 mp4 文件> /data/out.yuv
```

2) 测试 rkmpp 的视频 encoder

输入上面的 yuv 文件，编码成 h.264 格式的文件

```
ffmpeg -f rawvideo -vcodec rawvideo -s 960x540 -r 30 -pix_fmt nv12 -i out.yuv -vcodec h264_rkmpp output.h264
```

转码视频文件

```
ffmpeg -hwaccel rkmp -hwaccel_output_format drm_prime -c:v h264_rkmp -i <h264 格式 mp4 文件>
-vf "scale_rkrga=format=yuv420p" -c:v h264_rkmp output.mp4
```

3) 叠加汉字在视频文件上

```
ffmpeg -i in.mp4 -vf "drawtext=fontfile=/usr/local/share/fonts/truetype/msyh.ttf:text='My 你好新年快乐'
:fontcolor=white:fontsize=24:box=1:boxcolor=black@0.5:boxborderw=5:x=(w-text_w)/2:y=(h-text_h)/2"
-codec:a copy out.mp4
```

3.3 Opencv: 支持 ffmpeg, gstreamer, java 和 python 调用

显示 opencv 支持的功能列表：

```
import cv2
print(cv2.getBuildInformation())
GUI:          GTK3
  GTK+:       YES (ver 3.24.20)
  GThread :   YES (ver 2.64.6)
  GtkGExt:    NO
  OpenGL support: NO
  VTK support: NO
Video I/O:
  DC1394:     YES (2.2.5)
  FFMPEG:     YES
    avcodec:   YES (60.31.102)
    avformat:  YES (60.16.100)
    avutil:    YES (58.29.100)
    swscale:   YES (7.5.100)
    avresample: NO
  GStreamer:  YES (1.16.3)
  v4l/v4l2:   YES (linux/videodev2.h)
Java:
  ant:        NO
  Java:       YES (ver 1.8.0.422)
  JNI:        /usr/lib/jvm/java-8-openjdk-arm64/include /usr/lib/jvm/java-8-openjdk-arm64/include/linux
  Java wrappers: YES
  Java tests:  NO
```

3.4 VLC Player

支持文件, rtsp, rtmp 等视频流播放

3.5 内置 lightdm and weston UI 系统

系统默认启动后使用的是 lightdm UI 系统，可以通过下面命令切换到 weston UI 系统：

```
systemctl stop lightdm.service
systemctl disable lightdm.service
systemctl start weston.service
systemctl enable weston.service
```

切换回 lightdm UI 系统的命令如下：

```
systemctl stop weston.service
systemctl disable weston.service
systemctl start lightdm.service
systemctl enable lightdm.service
```

4 Q&A

4.1 Docker 启动失败

在某些情况下，`sudo systemctl status docker.service` 显示失败。

解决方法：`rm -f /etc/docker/daemon.json`

`sudo systemctl restart docker.service`

4.2 是否有方法升级或者重置盒子的系统？

盒子出厂的时候安装标准的 Ubuntu 20.04.5 LTS 操作系统，用户可以根据需要自行安装软件。但是该系统是和具体的 soc 和硬件适配过的，无法进行常规的升级或者系统重置操作。

5 附录

5.1 使用调试串口诊断问题盒子

按照前图用 Micro USB 线连接盒子和 PC 机。

如果使用 Linux 服务器，请首先安装 minicom 软件，然后配置 minicom 到对应的串口：

Ubuntu 机器：`sudo apt-get install minicom`

Centos 机器：`sudo yum install minicom lrzsz`

如果能够识别串口，在系统的 `/dev` 目录下会找到设备：`ls /dev/ttyUSB0`

`sudo minicom -s`

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup           |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+-----+-----+-----+
```

```

+-----+
| A - Serial Device      : /dev/ttyUSB0 |
| B - Lockfile Location  : /var/lock    |
| C - Callin Program     :              |
| D - Callout Program    :              |
| E - Bps/Par/Bits       : 115200 8N1  |
| F - Hardware Flow Control : No       |
| G - Software Flow Control : No       |
| H - RS485 Enable       : No          |
| I - RS485 Rts On Send  : No          |
| J - RS485 Rts After Send : No        |
| K - RS485 Rx During Tx : No          |
| L - RS485 Terminate Bus : No         |
| M - RS485 Delay Rts Before: 0        |
| N - RS485 Delay Rts After : 0        |
|                               |
| Change which setting? [ ]          |
+-----+

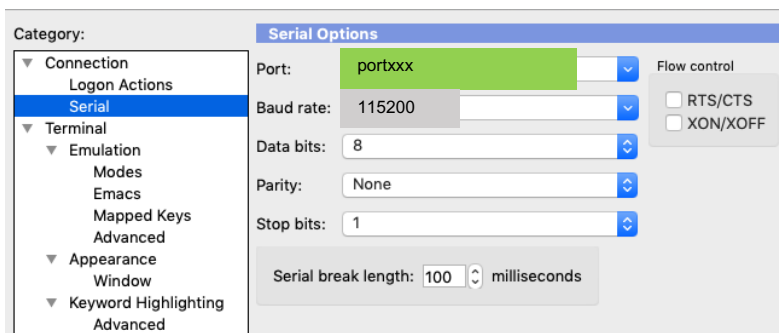
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols     |
| Serial port setup           |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..             |
| Exit                         |
| Exit from Minicom           |
+-----+

```

最后选择 “Exit” 后，进入串口模式：

⚠ “Serial Device” 的填写要按照开发系统实际检测到的 usb 设备名称。上图中的/dev/ttyUSB0 只是一个示例！！

如果使用 Windows PC，请首先安装一个串口工具，推荐使用“SecureCRT”。下图为 SecureCRT 的串口配置界面：



盒子上电后，应该在串口有打印信息。等待出现以下打印后，登录 Ubuntu 系统。

```

Ubuntu 20.04.5 LTS xhiveai ttyFIQ0
xhiveai login: root (automatic login)

Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.10.160 aarch64)

 * Documentation:  https://www.apoidea.io
 * Management:    https://www.apoidea.io

System information as of Mon Nov 27 17:08:49 CST 2023

System load: 0.96          Swap usage: 0%          Users logged in: 0
Usage of /: 15.4% of 13.22GB Temperature: 32.4 C
Memory usage: 3%          Processes: 241
Last login: Mon Nov 27 15:10:42 CST 2023 on ttyFIQ0
root@xhiveai:~#

```

5.2 通过 ssh 连接

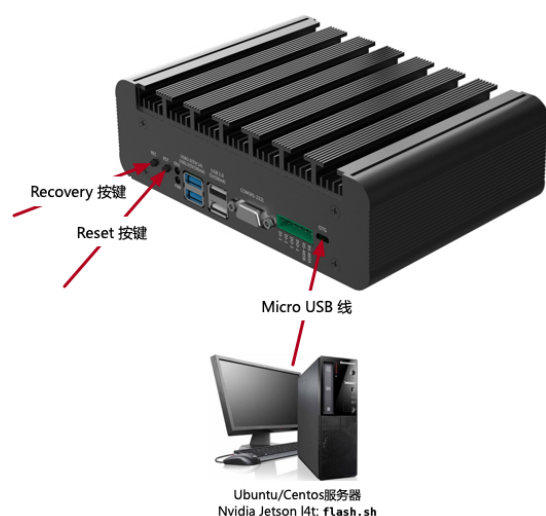
ssh root@<盒子 IP 地址>

密码：apoidea_xhiveai

5.3 使用扩展 Nvme SSD Flash

某些盒子出厂的时候按照客户的要求扩展安装 M.2 接口的 Nvme SSD flash. 系统在启动的时候会自动检测是否存在外接的 ssd，如果有，自动 mount 到/data 目录下来使用。

5.4 使用 OTG 接口重烧或者升级固件



第一步：向技术支持索要升级的软件包，在提供公司的 email 地址后，升级的软件包会以邮件的方式发给客户

第二步：给个出厂的盒子会分配一个 Mac 地址，需要记录下该 Mac 地址，在升级固件后，重新配置该 Mac 地址。

在盒子的图形终端上，执行 ifconfig，记录下 Mac 地址。例子：`ether 88:5f:e8:10:05:95`

第三步：用 Micro USB 线连接盒子的 otg 接口和 Ubuntu x86 PC 机的 USB 接口

⚠️可以是物理的 x86 PC 机，也可以是 windows 上的虚拟机。安装 Ubuntu 16.04 或者更新的版本

第四步：在 Ubuntu PC 机上安装固件烧写软件。

- 1) 下载 Linux_Upgrade_Tool.tar.bz2
- 2) tar jxf Linux_Upgrade_Tool.tar.bz2
- 3) 执行：`./upgrade_tool -h`，应该正常显示帮助信息

第五步：给盒子上电后。一直按住 Recovery 按键不放的同时点击 Reset 按键，过 5 秒后，再松开 Recovery 按键。

判断是否进入 Recovery mode :

执行命令 :

```
./upgrade_tool LD
```

List of rockusb connected(1)

```
DevNo=1      Vid=0x2207,Pid=0x350b,LocationID=17 Mode=Loader  SerialNo=
```

如果有找到设备，代表盒子成功进入了 recovery mode。如果没有发现设备，重新执行上述步骤。

第六步：下载烧写的固件并烧写固件。

仔细阅读 split_merge.readme，参考[merge]来完成完整固件的制作，然后按照[otg burning]来烧写固件到盒子。

第七步：将 config_aibox.sh 下载到一台 Linux PC 上，确保该机器可以通过网络和盒子连通。

在配置完盒子的 ip 地址后，在 Linux PC 上执行如下命令：

```
./config_aibox.sh <remote_ip> <mac_addr>
```

<remote_ip>：AI 盒子的 ip 地址

<mac_addr>：记录下的盒子已分配 IP 地址

完成后，重启盒子两次后，用 ifconfig 查看更新的 Mac 地址是否生效。

5.5 调试盒子的技巧

5.5.1 快速更新 KERNEL, DTB 或者 MODULES 文件

有时候客户的应用程序的功能依赖于 kernel，可能需要添加 kernel 的配置，修改 dtb 文件或者把某些模块编译成 modules ko 等。在通过技术支持获取更新的 kernel，dtb 或者 modules ko 文件后，按照下面步骤来更新。

1) 通过 ssh 登陆 AI 盒子(确保下面操作过程中，AI 盒子不会中途断电或者 reset)

【更新 kernel 文件】

```
cd /boot
```

```
rm -f Image-5.10.160 //老盒子
```

OR

```
rm -f Image-5.10.198 //新盒子
```

- 在 Ubuntu/Centos host 机器上，执行

```
// Image-5.10.160 为老盒子的 kernel，Image-5.10.198 为新盒子的 kernel
```

```
md5sum Image-5.10.160 / Image-5.10.198
```

```
<xxxxxxxxxxxxxxxx> Image-5.10.160 / Image-5.10.198 //记录下校验码，后面确保 kernel 文件的远程 copy 是正确的！！
```

```
scp Image-5.10.160 / Image-5.10.198 root@<AI 盒子 ip 地址>:/boot
```

- 在 AI 盒子上确认 kernel 文件完好无损。

```
// Image-5.10.160 为老盒子的 kernel，Image-5.10.198 为新盒子的 kernel
```

```
cd /boot
```

```
md5sum Image-5.10.160 / Image-5.10.198
```

```
<xxxxxxxxxxxxxxxx> Image-5.10.160 / Image-5.10.198 //确保该校验码与前面获取的一直，以确保拷贝的完整性！！
```

【更新 kernel modules 文件】

```
cd /boot
```

```
rm -rf lib
```

- 在 Ubuntu/Centos host 机器上，执行

```
scp kernel_modules.tar.bz2 root@<AI 盒子 ip 地址>:/boot
```

- 在 AI 盒子上部署 kernel modules。

```
cd /boot
```

```
tar jxf kernel_modules.tar.bz2
```

```
<查看是否生成 lib 目录>
```

```
rm -f kernel_modules.tar.bz2
```

【更新 dtb 文件】

```
cd /boot
```

```
rm -f rk3588-xhiveai-lp4-v10-linux.dtb
```

- 在 Ubuntu/Centos host 机器上，执行

```
md5sum rk3588-xhiveai-lp4-v10-linux.dtb
```

```
<xxxxxxxxxxxxxxxx> rk3588-xhiveai-lp4-v10-linux.dtb //记录下校验码，后面确保 dtb 文件的远程 copy 是正确的！！
```

```
scp rk3588-xhiveai-lp4-v10-linux.dtb root@<AI 盒子 ip 地址>:/boot
```

- 在 AI 盒子上确认 dtb 文件完好无损。

```
cd /boot
```

```
md5sum rk3588-xhiveai-lp4-v10-linux.dtb
```

```
<xxxxxxxxxxxxxxxx> rk3588-xhiveai-lp4-v10-linux.dtb //确保该校验码与前面获取的一直，以确保拷贝的完成性！！
```

2) 需要升级的 kernel 文件都升级完成后，重启盒子，使新的 kernel 生效。

```
reboot
```