
XRD R818 开发指南

XRD LINUX

V1.0
2022/04/15

目录

第 1 章 上手教程.....	4
1.1 入手指南.....	4
1.1.1 配件及启动.....	4
1.1.2 开机与重启.....	4
1.2 串口调试.....	5
1.2.1 选购适配器.....	5
1.2.2 硬件连接.....	5
1.2.3 连接参数.....	5
1.2.4 Windows 上使用串口调.....	6
1.2.5 Ubuntu 上使用串口调试:	7
1.3 使用 ADB 调试.....	7
1.3.1.下载 windows 的 adb 工具.....	7
1.3.2.配置环境变量.....	8
1.4 固件升级.....	10
1.4.1 准备工作.....	10
1.4.2 MICRO USB OTG 升级.....	10
1.4.4 MICRO SD 卡升级.....	14
第 2 章 linux ubuntu 教程.....	15
2.1 R818 系统介绍.....	15
2.1.1 特性.....	15

2.1.2 基础信息.....	15
2.2 ubuntu 界面系统.....	15
2.2.1 系统软件更新.....	15
2.2.2 虚拟键盘.....	16
2.2.3 串口工具.....	16
2.2.4 Qt 环境安装.....	16
2.2.5 服务命令操作.....	16
2.2.5 添加用户.....	17
2.2.6 开机启动应用程序.....	17
2.2.7 导出根文件系统.....	18
2.3 网络设备配置.....	19
2.3.1 以太网静态配置.....	19
2.3.2 以太网动态配置.....	19
2.3.2 wifi 配置.....	20
2.3.3 ssh.....	20
2.4 常用外设使用方法.....	21
2.4.1 RTC 使用.....	21
2.4.2 GPIO 使用.....	21
2.4.3 PWM 使用.....	22
2.4.4 背光控制.....	23

第 1 章 上手教程

1.1 入手指南

1.1.1 配件及启动

标准：R818 主板一块。

可选购的配件：

- 12/2A 的直流电源适配器
- 欣瑞达串口模块
- micro usb OTG
- usb 卡套,micro SD 卡

1.1.2 开机与重启

将电源适配器插入带电的插座上，电源线接口插入开发板，开发板加电开机。开机是开发板的核心板会有蓝色的电源指示灯亮起来，设备启动 logo15s 左右，进入系统。

重启：

- 可长按 power 键 5s
- 或者直接掉电重新上电。
- 通过终端或者控制台指令发送 reboot

1.2 串口调试

1.2.1 选购适配器

调试串口可购买我司的串口工具，我司串口工具兼容 232 电平和 TTL 电平，串口 debug 口的电平我司默认为串口 0，默认的电平为 232。

也可以购买网上有很多 usb 转串口的适配器，购买时先确保是否可以支持 232 电平，按芯片来分，有以下几种：

- PL2303
- CH340

一般来说采用 CH340 芯片的适配器，性能比较稳定，价格上贵一些。

1.2.2 硬件连接

串口转 USB 适配器，有四根不同颜色的连接线

1. GND，串口的地线，接开发板串口的 GND 针
2. TXD，串口的输出线，接开发板串口的 TX0 针
3. RXD，串口的输入线，接开发板串口的 RX0 针

1.2.3 连接参数

使用以下串口参数：

- 波特率：115200
- 数据位：8

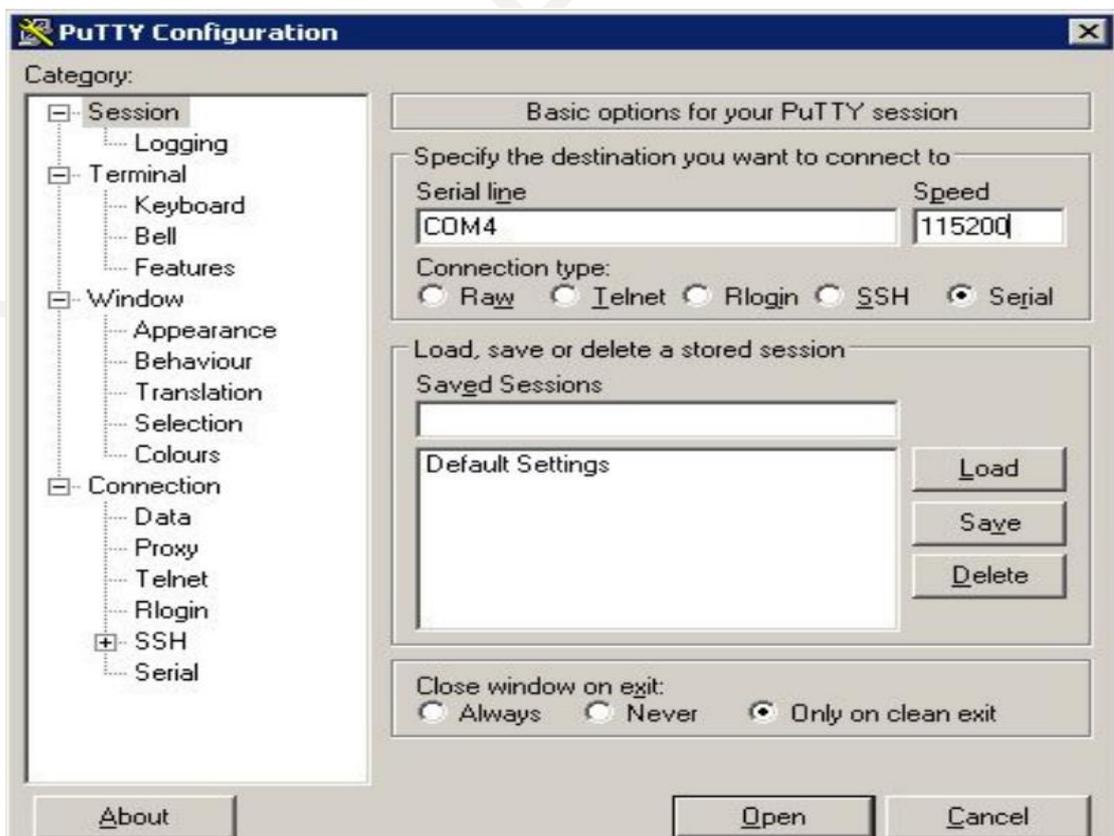
- 停止位: 1
- 奇偶校验: 无
- 流控: 无

1.2.4 Windows 上使用串口调

Windows 上一般用 putty 或 SecureCRT。其中 putty 是开源软件，在这里介绍一下， SecureCRT

下载 putty，建议下载 putty.zip，它包含了其它有用的工具。

解压后运行 PUTTY.exe，选择 Connection type (连接类型) 为 Serial (串口)，将 Serial line (串口线) 设置成设备管理器所看到的 COM 口，并将 Speed (波特率) 设置为 115200，按 Open (打开) 即可，如图所示。



1.2.5 Ubuntu 上使用串口调试:

在 Ubuntu 上可以有多种选择:

- Cutecom
- Minicom
- Kermit
- Picocom

在开发板上同样可以安装以上的的串口通信工具

安装方式: `apt install cutecom`

1.3 使用 ADB 调试

1.3.1. 下载 windows 的 adb 工具

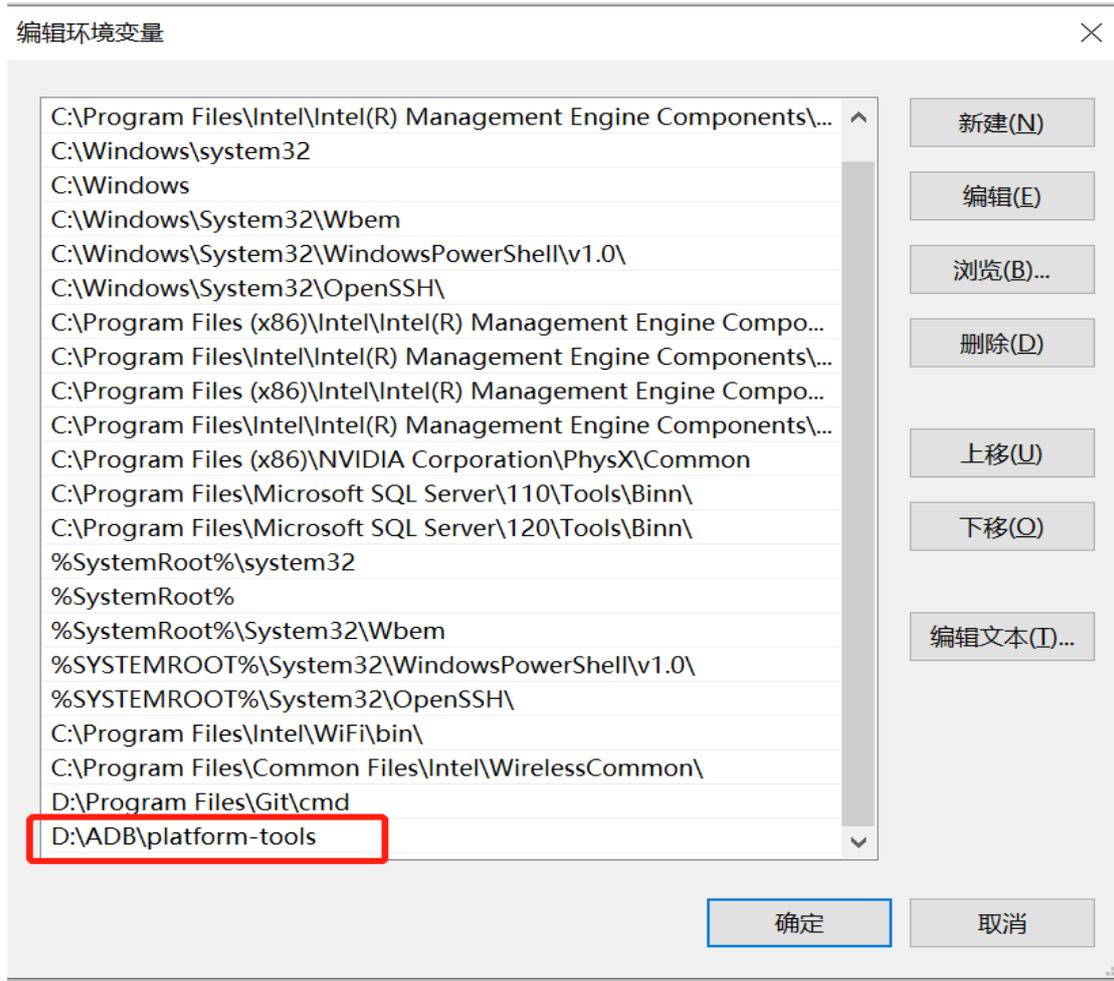
链接:

- Windows 版本 :
<https://dl.google.com/android/repository/platform-tools-latest-windows.zip>
- Mac 版本 : <https://dl.google.com/android/repository/platform-tools-latest-windows.zip>
- Linux 版本 : <https://dl.google.com/android/repository/platform-tools-latest-linux.zip>

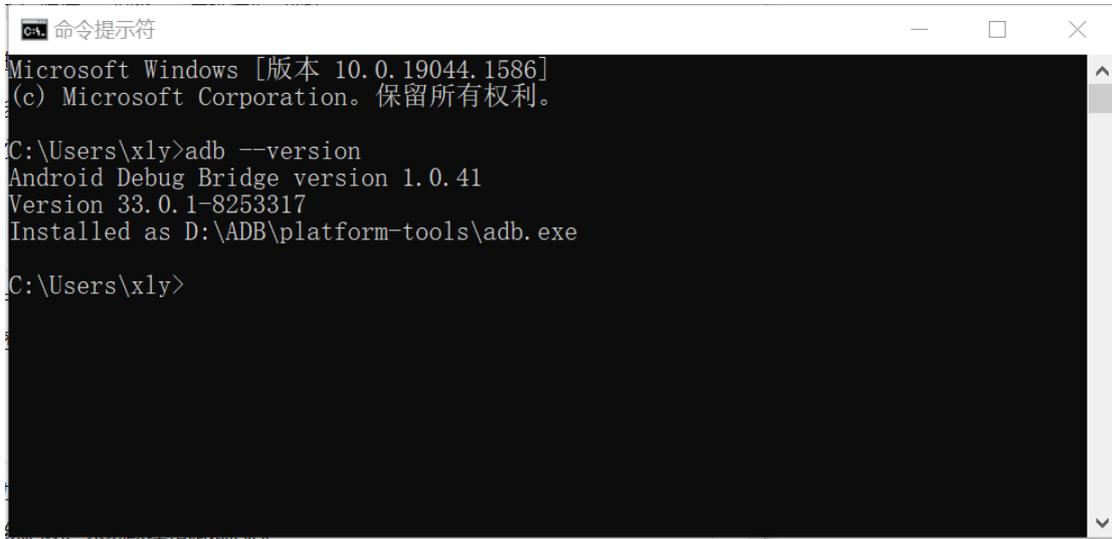
1.3.2.配置环境变量

1.3.2.1. 按键 windows+r 打开运行，输入 sysdm.cpl，回车。

高级 >>环境变量>>系统变量>>path



1.3.2.2. 查看是否安装成功，输入 adb --version



```
命令提示符
Microsoft Windows [版本 10.0.19044.1586]
(c) Microsoft Corporation。保留所有权利。

C:\Users\xly>adb --version
Android Debug Bridge version 1.0.41
Version 33.0.1-8253317
Installed as D:\ADB\platform-tools\adb.exe

C:\Users\xly>
```

1.3.2.3. 连接板子 MICRO USB OTG。

1.3.2.4. 查看 adb devices,看是否连接上板子

1.3.2.5. 输入 adb shell。进入开发板的 rootfs

1.4 固件升级

前言：本章将介绍两种固件更新方式。一种是 MICRO USB OTG，一般是开发者的烧录的方式，一种是 SD 卡更新方式，这种方式主要用于量产。

1.4.1 准备工作

- R818 开发板
- 固件
- 主机
- 良好的 Micro USB OTG 线(MICRO USB OTG 升级)
- Micro SD 卡 (SD 标准固件大小需大于等于 8g)
- 固件文件一般有两种：
- 单个统一固件 update.img, 将启动加载器、参数和所有分区镜像都打包到一起，用于固件发布。

1.4.2 MICRO USB OTG 升级

1. 获取软件包 PhoenixSuit-v1.1.3.rar 并解压。具体烧录细节可参见帮助文档。
2. 运行 PhoenixSuit.exe 启动软件
3. 选择一键刷机



4. 加载固件



5. 如果只想烧录内核或者单独烧录 rootfs，则可以勾选了单或多分区来选择下载。



6. 按住 boot 键之后再给开发板上电。如果不勾选单或多分区下载，则会弹出是否格式化烧录，直接选择是的。如果勾选了则会直接下载。

PhoenixSuit 一键刷机工具

首页 一键刷机 设备管理 资讯页面

开始烧写固件

PhoenixSuit

提示信息: 是否强制格式化?

强制格式化会导致文件丢失, 请事先备份重要文件!

选择Yes, 进入格式化升级模式。
选择No, 进入普通升级模式。(推荐使用)

是 否

注意: 刷机前, 请检查设备电量。切勿在刷机过程中, 拔出设备。

开始烧写固件 PhoenixSuit V1.13

PhoenixSuit 一键刷机工具

首页 一键刷机 设备管理 资讯页面

正在烧写固件: 37% 耗时[0]分[5]秒

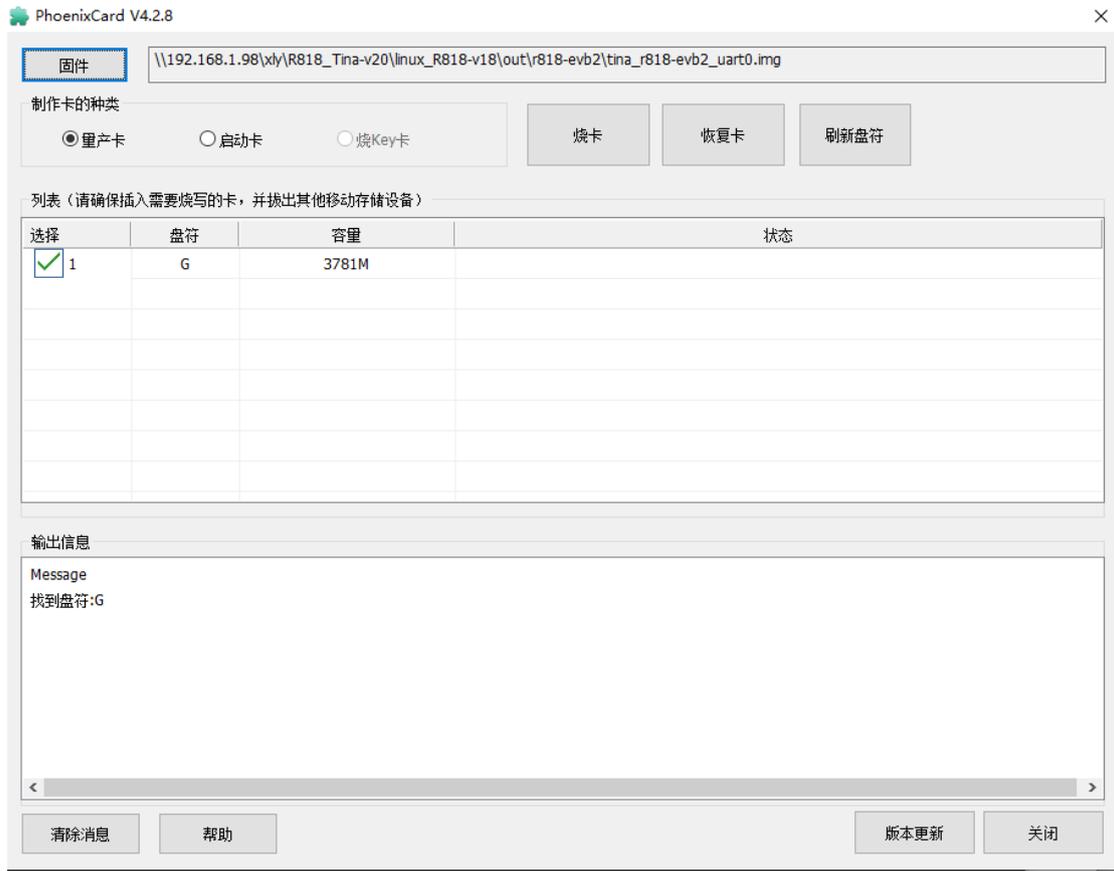
立即升级

注意: 刷机前, 请检查设备电量。切勿在刷机过程中, 拔出设备。

开始烧写固件 PhoenixSuit V1.13

1.4.4 MICRO SD 卡升级

1. 获取卡烧录软件包。
2. 运行 PhoenixCard.exe



3. 插入 micro sd 卡到主机上，选择量产模式。然后点击烧卡，等待完成
4. 烧录好之后，直接将卡插入到开发板，然后上电。上电之后进入更新模式，此时屏幕会有进度条显示。等待进度条拉满即烧录成功。
5. 如果选择卡启动烧卡，则是上电之后程序在 micro sd 卡启动，不跑 emmc。

第 2 章 linux ubuntu 教程

2.1 R818 系统介绍

2.1.1 特性

R818 Ubuntu Desktop 系统基于 Ubuntu 32bit 系统构建，目前主要维护 Ubuntu18.04 版本。具有以下特性：

- 桌面环境采用 XFCE，简洁美观。
- Xserver 使用 GPU + RGA 进行 2D 加速，运行流畅，占用 CPU 资源少。
- 针对嵌入式平台，精简系统服务。
- 提供基于 Arm Mali GPU 的 OpenGL、OpenCL 支持。
- 适配 QT、Docker、Electron 等开发框架。

2.1.2 基础信息

Linux Desktop 系统开机启动后，自动登录到 xrd 用户。

xrd 用户密码： 123

root 用户密码： 123

2.2 ubuntu 界面系统

2.2.1 系统软件更新

第一次安装软件前需要联网后在终端输入命令

```
apt update
```

2.2.2 虚拟键盘

系统安装了虚拟键盘工具，在没有键盘的情况下，可以点击打开左上角的菜单中的应用找到：matchbox-keyboard，点击打开，即可触控虚拟键盘输入。

2.2.3 串口工具

系统默认安装了 cutecom 工具，可在菜单中找到直接打开，界面方便测试。

2.2.4 Qt 环境安装

输入命令：`apt install qt5-default make g++`

等待安装成功。

查看 qmake 版本：`qmake --version`

18.04ubuntu 默认为 qt5.9

2.2.5 服务命令操作

1. 使能服务命令 `systemctl enable demo.service`
2. 同步服务命令：`systemctl daemon-reload`
3. 开启服务命令：`service rockchip start`
4. 查看服务状态：`systemctl status rockchip.service`
5. 停止服务命令 `systemctl enable demo.service`

2.2.5 添加用户

以下是设置 xrd 账号的命令 demo

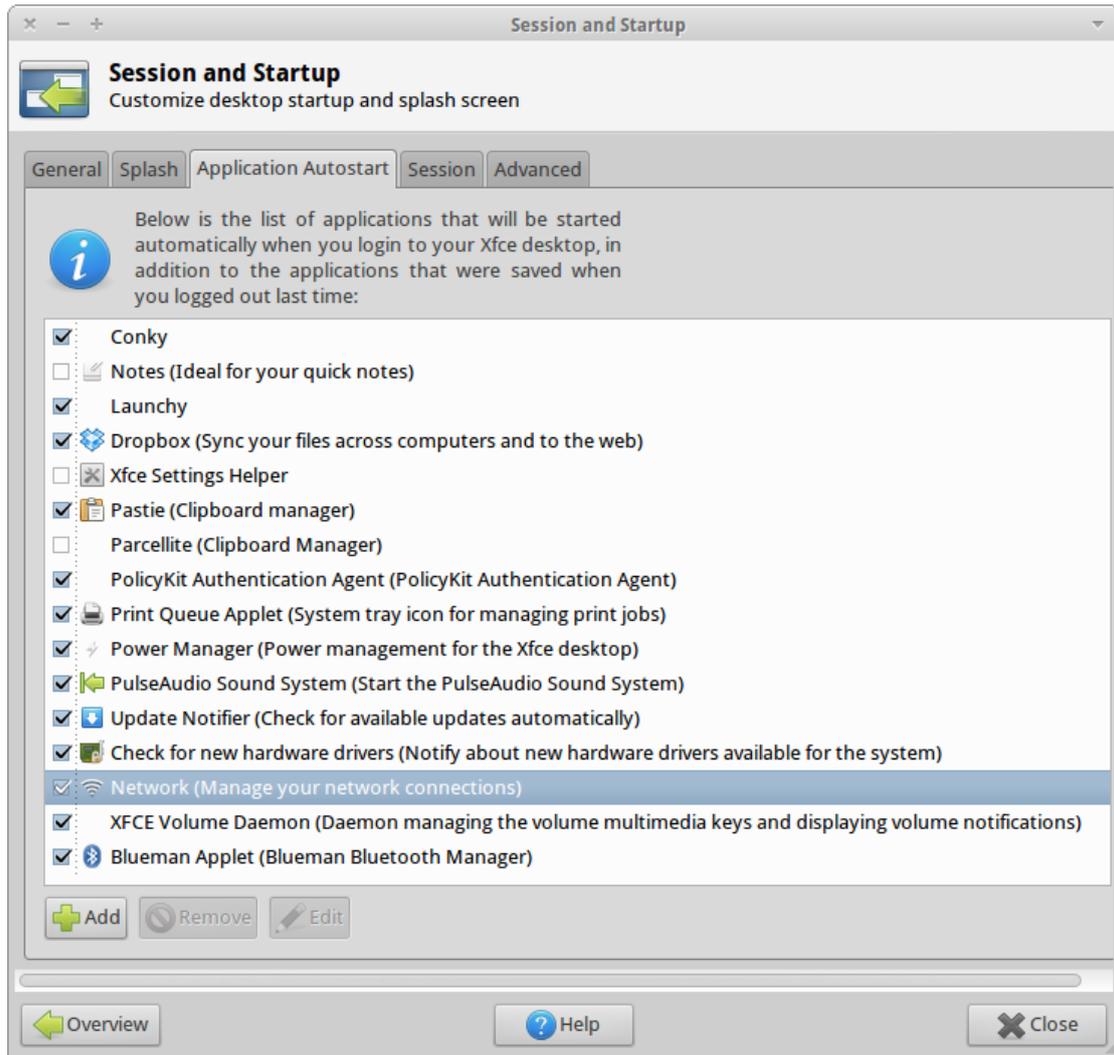
1. `echo "xrd" > /etc/hostname`
2. `echo "127.0.0.1 xrd" > /etc/hosts`
3. `useradd -s '/bin/bash' -m -G adm,sudo xrd`
4. `passwd xrd`

2.2.6 开机启动应用程序

在系统中，可以根据用户需求自行设置应用程序开机自动启动。

在系统桌面点击搜索的图标，搜 session and startup，然后 launch，则弹出一下界面。可

以选择你要设置的开机自动程序应用，也可以选择添加，添加上你想自启动的应用或者命令



2.2.7 导出根文件系统

根据客户修改 rootfs，只需要将修改后的 rootfs img 导出来，重新制作成定制的固件。

执行命令 `xrd_export` (导出路径: 如 `xrd_export /home/xrd/udisk`)，此导出过程花费时间比较长。

2.3 网络设备配置

2.3.1 以太网静态配置

系统安装了 netplan。

文件格式很严格。注意空格位置

打开所在文件：vim /etc/netplan/01-network-manager-all.yaml

```
network:

  version: 2

  # renderer: networkd

  ethernets:

    eth0:

      dhcp4: no

      optional: true

      addresses: [192.168.1.95/24] #其中的 24 指的是掩码位

      gateway4: 192.168.1.1
```

2.3.2 以太网动态配置

```
network:

  version: 2

  # renderer: networkd

  ethernets:

    eth0:
```

dhcp4: true

optional: true

Waddresses: [192.168.1.95/24] #其中的 24 指的是掩码位

#gateway4: 192.168.1.1

2.3.2 wifi 配置

2.3.2.1 查看 wifi

wpa_cli -i wlan0 scan_result

2.3.2.2.连接 wifi

使用命令：nmcli

nmcli dev wifi connect wifi 名 password WiFi 密码

示例：nmcli dev wifi connect TP-LINK_XRD12131415 password XRD12131415

2.3.3 ssh

Ssh 远程调试与文件传输也是很方便的一种

在出厂系统中 ssh 默认是打开的，开发板连上网络之后，用 ifconfig 查看 ip 地址

即可在主机串口终端连接上并开始调试或者传输文件

2.4 常用外设使用方法

2.4.1 RTC 使用

插上电池，确保电池有电，进入系统后通过以下命令设置时间：

```
# date -s "2019-04-28 15:06:00"
```

```
Sun Apr 28 15:06:00 UTC 2019
```

```
# hwclock -w
```

```
# date
```

```
Sun Apr 28 15:06:09 UTC 2019
```

2.4.2 GPIO 使用

注：控制 gpio 需要保证该引脚已经配置成普通 gpio，例如控制 GPIOPH5

1. #echo 229 > /sys/class/gpio/export
2. #echo out > /sys/class/gpio/gpio229 /direction
3. #echo 1 > /sys/class/gpio/gpio229 /value
4. #echo 0 > /sys/class/gpio/gpio229 /value

说明:

- `/sys/class/gpio/export` 文件用于通知系统需要导出控制的 GPIO 引脚编号, `/sys/class/gpio/export` 写编号, 比如以下命令, 命令成功后生成 `/sys/class/gpio/gpio229` 目录, 如果没有出现相应的目录, 说明此引脚不可导出;
- 计算引脚编号, PH5 示例 引脚编号 = $32*(H-A) + 5 = 229$
- `value` 文件是端口的数值, 为 1 或 0, `echo` 写入高低电平之后可以使用万用表测量该引脚是否可控。
- `direction` 文件, 定义输入输入方向

2.4.3 PWM 使用

在命令行中设置 PWM 输出, 调出 `pwm0` 目录下设备节点, *为你要设的 pwm。 在命令行输入下列命令:

```
#echo 1 > /sys/class/pwm/pwmchip*/export
```

```
#echo 1 > /sys/class/pwm/pwmchip*/pwm0/enable //设置使能
```

```
#echo 1000000 > /sys/class/pwm/pwmchip*/pwm0/period //设置周期
```

```
#echo 500000 > /sys/class/pwm/pwmchip*/pwm0/duty_cycle //设置占空比
```

2.4.4 背光控制

在命令行中设置背光亮度:

```
/sys/class/disp/disp/attr/disp
```

```
/sys/class/disp/disp/attr/enhance_bright /* 亮度*/
```

```
/sys/class/disp/disp/attr/enhance_contrast /* 对比度*/
```

```
/sys/class/disp/disp/attr/enhance_saturation /* 饱和*/
```

```
/sys/class/disp/disp/attr/enhance_edge /* 边缘锐度*/
```

```
/sys/class/disp/disp/attr/enhance_detail /* 细节增强*/
```

```
/sys/class/disp/disp/attr/enhance_denoise /* 降噪*/
```

- ARGUMENTS

disp display channel, 比如 0: disp0, 1: disp1

enhance_xxx: 范围: 0~100, 数据越大, 调节幅度越大。

```
//设置 disp0 的图像亮度为 80
```

```
echo 0 > /sys/class/disp/disp/attr/disp;
```

```
echo 80 > /sys/class/disp/disp/attr/enhance_bright;
```

```
//设置 disp1 的饱和度为 50
```

```
echo 1 > /sys/class/disp/disp/attr/disp;
```

```
echo 50 > /sys/class/disp/disp/attr/enhance_saturation;
```