IDO-EVB7608-V1

Linux开发手册

深圳触觉智能科技有限公司

www.industio.cn

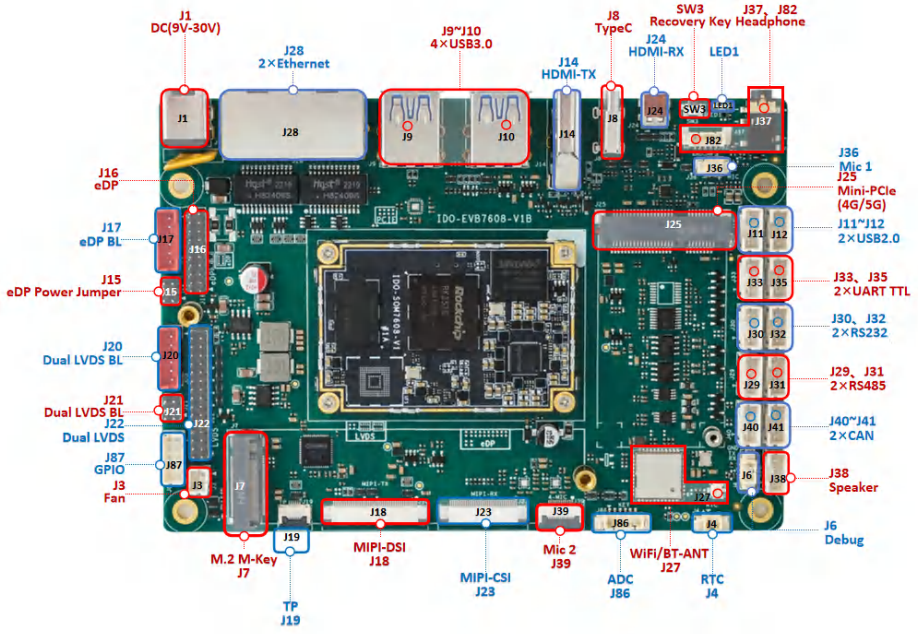
文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档	MHK	IDO	2025/05/27

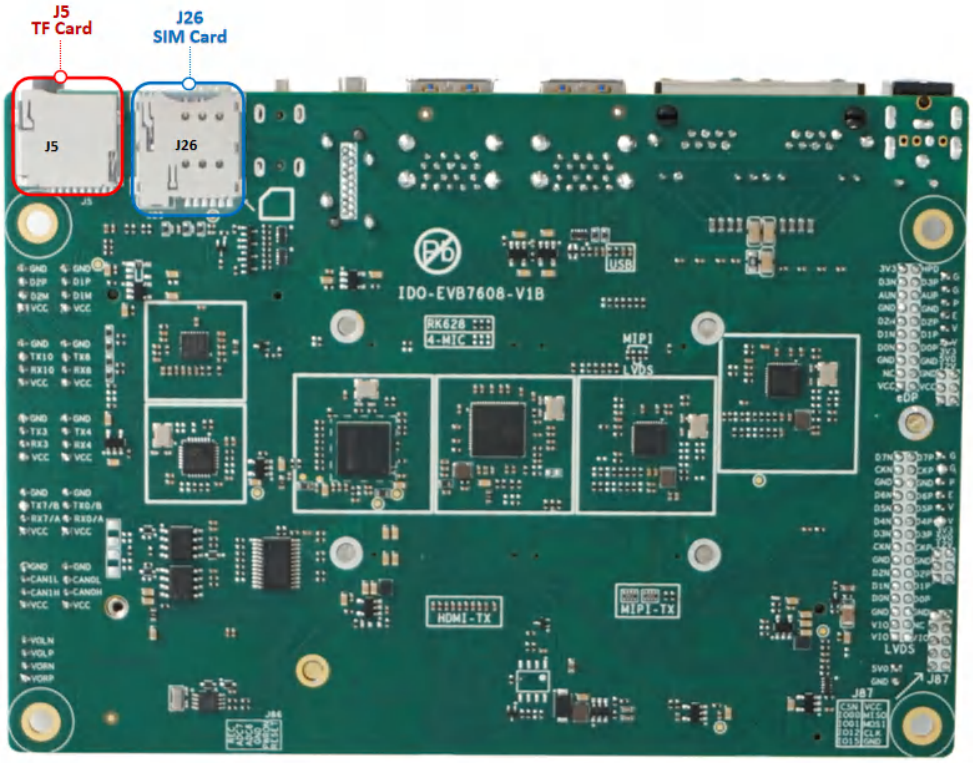
1 上手指南

1.介绍

IDO-EVB7608 配置 Rockchip 八核 64 位 AIOT 处理器 RK3576，采用先进工艺制程，高性能低功耗，内置 ARM Mali G52 MC3 GPU，集成6 TOPS算力NPU，支持主流大模型的私有化部署。具备 4K@120fps 高清高帧率显示能力，支持外部看门狗，具有实时网络、Flexbus、硬件资源隔离、DSMC 等工业新特性，满足不同的应用需求。



IDO-EVB7608-V1B 正面



IDO-EVB7608-V1B 背面

2 Linux开发

2.1.获取 SDK

下载路径：百度网盘/EVB7608-Linux/4.软件资料/SDK/

2.2.校验md5值

```Bash
$ md5sum rk3576_linux6.1_rkr3_v1.tar.gz
```

2.3.解压

首先准备一个空文件夹用于存放 SDK，建议在 home 目录下，本文以~/evb7608为例

```bash
$ mkdir ~/evb7608
$ cd ~/evb7608
$ tar -xvf rk3576_linux6.1_rkr3_v1.tar.gz -C ./

$ cd rk3576_linux6.1_rkr3_v1

#查看分支(默认分支为ido-evb7608)
$ git branch
* ido-evb7608
  ido-evb7608-preempt-rt
  ido-evb7608-xenomai-rt

#获取代码
$ git reset --hard

```

2.4.SDK目录介绍

Bash

```
$ tree -L 1
.
├── app
├── buildroot
├── build.sh -> device/rockchip/common/scripts/build.sh
├── common -> device/rockchip/common
├── Copyright_Statement.md -> docs/licenses/LICENSE
├── debian
├── debian_rootfs
├── device
├── docs
├── external
├── kernel -> kernel-6.1
├── kernel-6.1
├── Makefile -> device/rockchip/common/Makefile
├── output
├── prebuilts
├── README.md -> device/rockchip/common/README.md
├── rkbin
├── rkflash.sh -> device/rockchip/common/scripts/rkflash.sh
├── rockdev -> output/firmware
├── tools
├── u-boot
└── yocto
```

Bash

```
app：存放上层应用 APP，主要是一些应用Demo。
buildroot：基于 Buildroot（2024）开发的根文件系统。
device/rockchip：存放芯片板级配置以及SDK编译和打包固件的脚本和文件等。
docs：存放通用开发指导文档、芯片平台相关文档、Linux 系统开发相关文档、其他参考文档等。
external：存放第三方相关仓库，包括显示、音视频、摄像头、网络、安全等。
kernel：存放 Kernel 开发的代码。
output：存放每次生成的固件信息、编译信息、XML、主机环境等。
prebuilts：存放交叉编译工具链。
rkbin：存放 Rockchip 相关二进制和工具。
rockdev：存放编译输出固件,实际软链接到 output/firmware 。
tools：存放 Linux 和 Window 操作系统下常用工具。
u-boot：存放基于 v2017.09 版本进行开发的 U-Boot 代码。
yocto: 存放yocto相关编译脚本和代码。
```

注意：

1. SDK 采用交叉编译，所以要在 X86_64 电脑上使用 SDK，不要将 SDK 下载到板子上。

2. 编译环境请使用 Ubuntu22.04（真机或 虚拟机），如果使用其他版本可能导致编译出错。
3. 不要在虚拟机共享文件夹以及非英文目录存放、解压SDK。
4. 获取、编译 SDK 请全程使用普通用户，不允许也不需要使用 root 权限（除非需要 apt 安装软件）

2.5.配置文件介绍

在 `device/rockchip/rk3576` 目录下，有不同板型的配置文件(xxxx_defconfig)，用于管理 SDK 每个环节的编译配置。rockchip_rk3576_ido_evb7608_v1b*_defconfig 作为基础的配置。其他版型的配置文件会引用 rockchip_rk3576_ido_evb7608_v1b*_defconfig 并且覆盖其中的一些变量配置。 以ido-evb3562-v2a-mipi-800x1280-buildroot_defconfig 做介绍：

Bash

```
RK_BUILDROOT_BASE_CFG="rk3576"
RK_CHIP_FAMILY="rk3576"                          #芯片类型
# RK_WIFIBT is not set
RK_UBOOT_SPL=y
RK_KERNEL_DTS_NAME="ido-evb7608-v1b_hdmi"        # 板型 dts 名称
RK_USE_FIT_IMG=y
RK_KERNEL_CFG="rockchip_linux_defconfig"         # kernel 编译配置
```

2.6 分区说明

parameter-flash.txt 文件中包含了固件的分区信息，以 parameter-evb3562.txt 为例：

```
FIRMWARE_VER: 1.0
MACHINE_MODEL: RK3576
MACHINE_ID: 007
MANUFACTURER: RK3576
MAGIC: 0x5041524B
ATAG: 0x00200800
MACHINE: 0xffffffff
CHECK_MASK: 0x80
PWR_HLD: 0,0,A,0,1
TYPE: GPT
GROW_ALIGN: 0
CMDLINE: mtdparts=:0x00002000@0x00004000(uboot),0x00002000@0x00006000(misc),0x00020000@0x00008000(boot),0x00040000@0x00028000(recovery),0x00010000@0x00068000(backup),0x00800000@0x00078000(userdata),0x00040000@0x00878000(oem),-@0x008B8000(rootfs:grow)
uuid:rootfs=614e0000-0000-4b53-8000-1d28000054a9
uuid:boot=7A3F0000-0000-446A-8000-702F00006273
```

CMDLINE 属性是我们关注的地方，以 uboot 为例， 0x00002000@0x00004000(uboot) 中 0x00002000 为uboot 分区的起始位置，0x00004000 为分区的大小，以此类推。

3.编译

3.1.选择项目

```Bash
mhkwork@dell-PowerEdge-R430:~/work/rk/rk3576/rk3576_linux6.1_rkr3_v1$ ./build.sh lunch
Log colors: message notice warning error fatal

Log saved at /mnt/5c953a98-9ee6-45b9-8cea-a2898eb313d7/mhk/rk/rk3576/rk3576_linux6.1_rkr3_v1/output/sessions/2025-05-27_17-31-12
Pick a defconfig:

1. rockchip_rk3576_ido_evb7608_v1_edp_1920x1080_defconfig
2. rockchip_rk3576_ido_evb7608_v1_hdmi_defconfig
3. rockchip_rk3576_ido_evb7608_v1_lvds_1920x1080_defconfig
4. rockchip_rk3576_ido_evb7608_v1_mipi_1920x1200_defconfig
Which would you like? [1]: 2
Switching to defconfig: /mnt/5c953a98-9ee6-45b9-8cea-a2898eb313d7/mhk/rk/rk3576/rk3576_linux6.1_rkr3_v1/device/rockchip/.chip/rockchip_rk3576_ido_evb7608_v1b_hdmi_defconfig
#
# configuration written to /mnt/5c953a98-9ee6-45b9-8cea-a2898eb313d7/mhk/rk/rk3576/rk3576_linux6.1_rkr3_v1/output/.config
#
Using last kernel version(6.1)
Using rootfs system(ubuntu) from environment
```

选择对应的数字即可。

3.2.自动编译

```Bash
$ ./build.sh
```

固件编译后生成路径：rockdev/update.img

3.3.分步编译

3.3.1.编译uboot

```Bash
./build.sh uboot
```

固件编译后生成路径：rockdev/uboot.img

3.3.2.编译kernel

```Bash
./build.sh kernel
```

固件编译后生成路径：rockdev/boot.img

3.3.3.编译buildroot系统

```Bash
./build.sh buildroot
```

固件编译后生成路径：rockdev/rootfs.img

3.3.4.编译ubuntu镜像

```Bash
./build.sh ubuntu
```

3.3.5.编译debian镜像

```Bash
./build.sh debian
```

3.3.6.编译recovery

```bash
./build.sh recovery
```

固件编译后生成路径：rockdev/recovery.img

3.3.7.固件打包

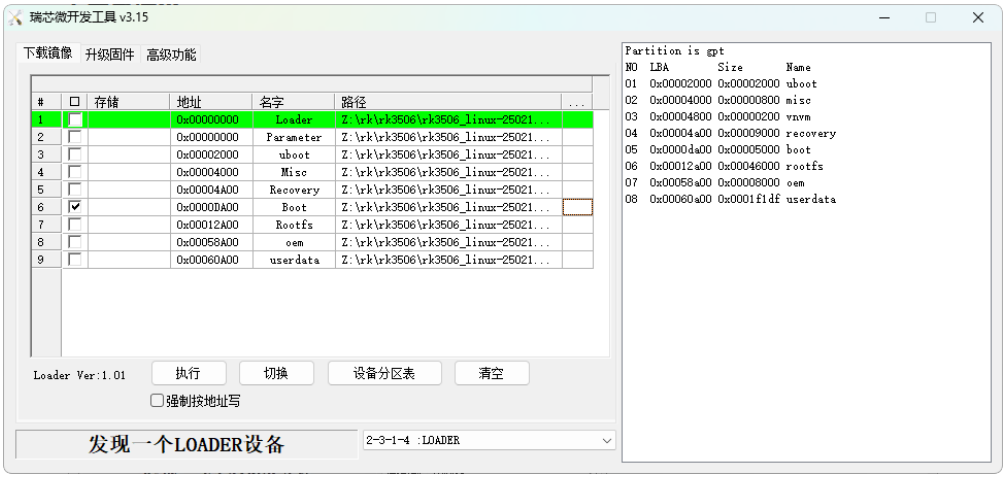
```bash
./build.sh updateimg
```

4 烧录说明

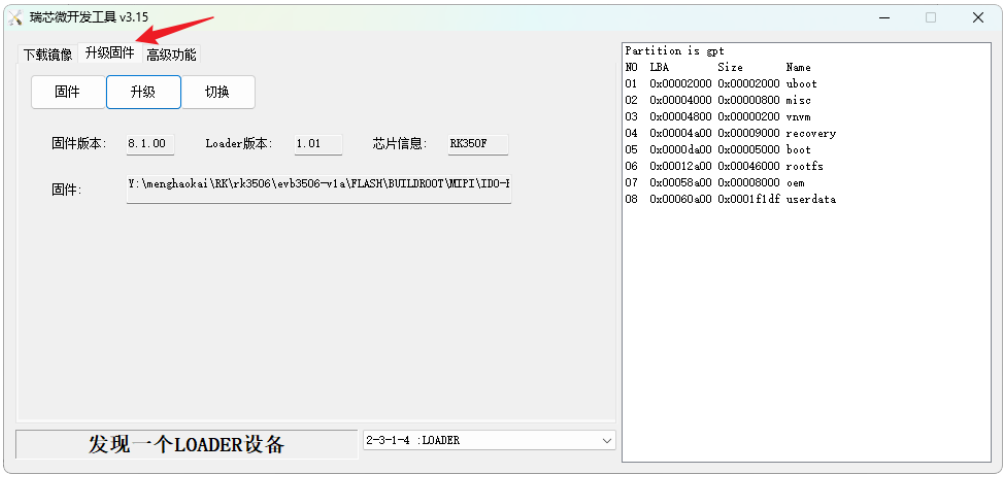
注意：板子需要进入到烧录模式，具体方法可以参考：《IDO-EVB7608-V1B 开发板固件烧录手册》

4.1.整包烧录

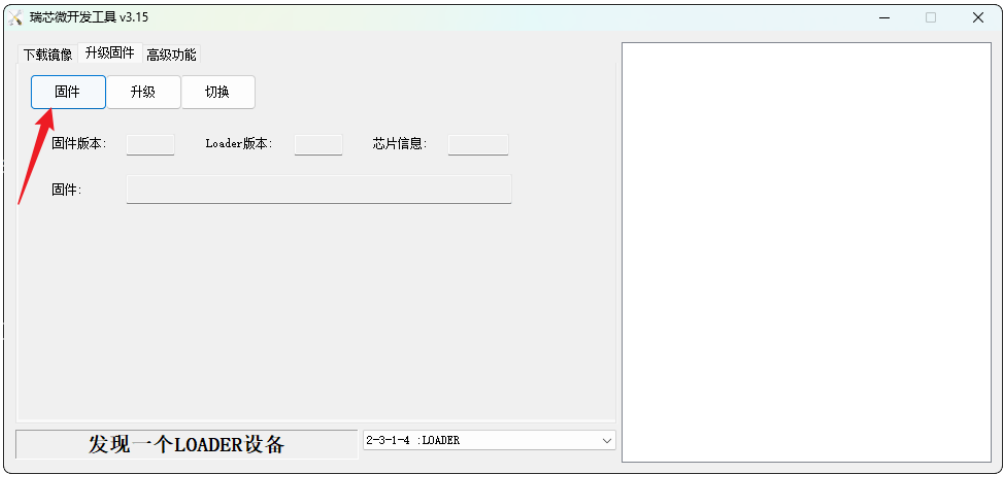
4.1.1.运行RK开发工具



4.1.2.点击升级固件

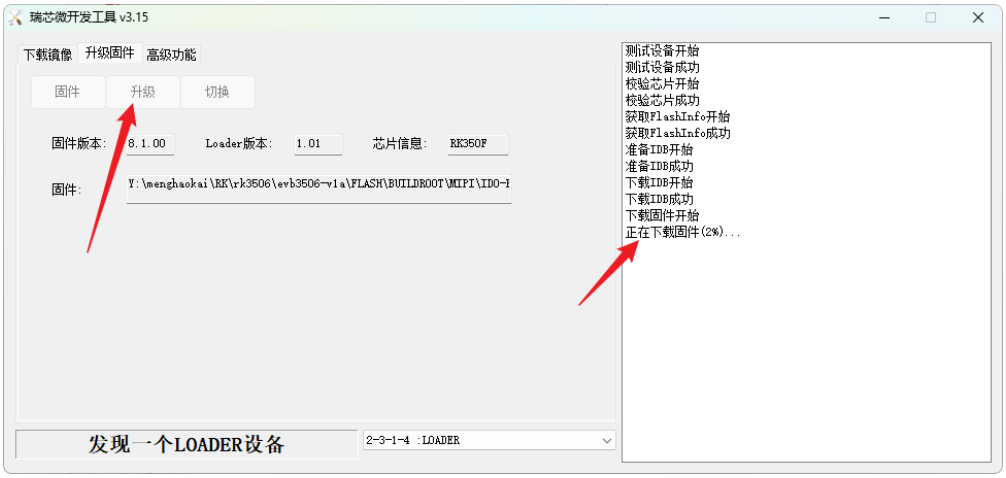


4.1.3.点击固件



选择：rockdev/update.img

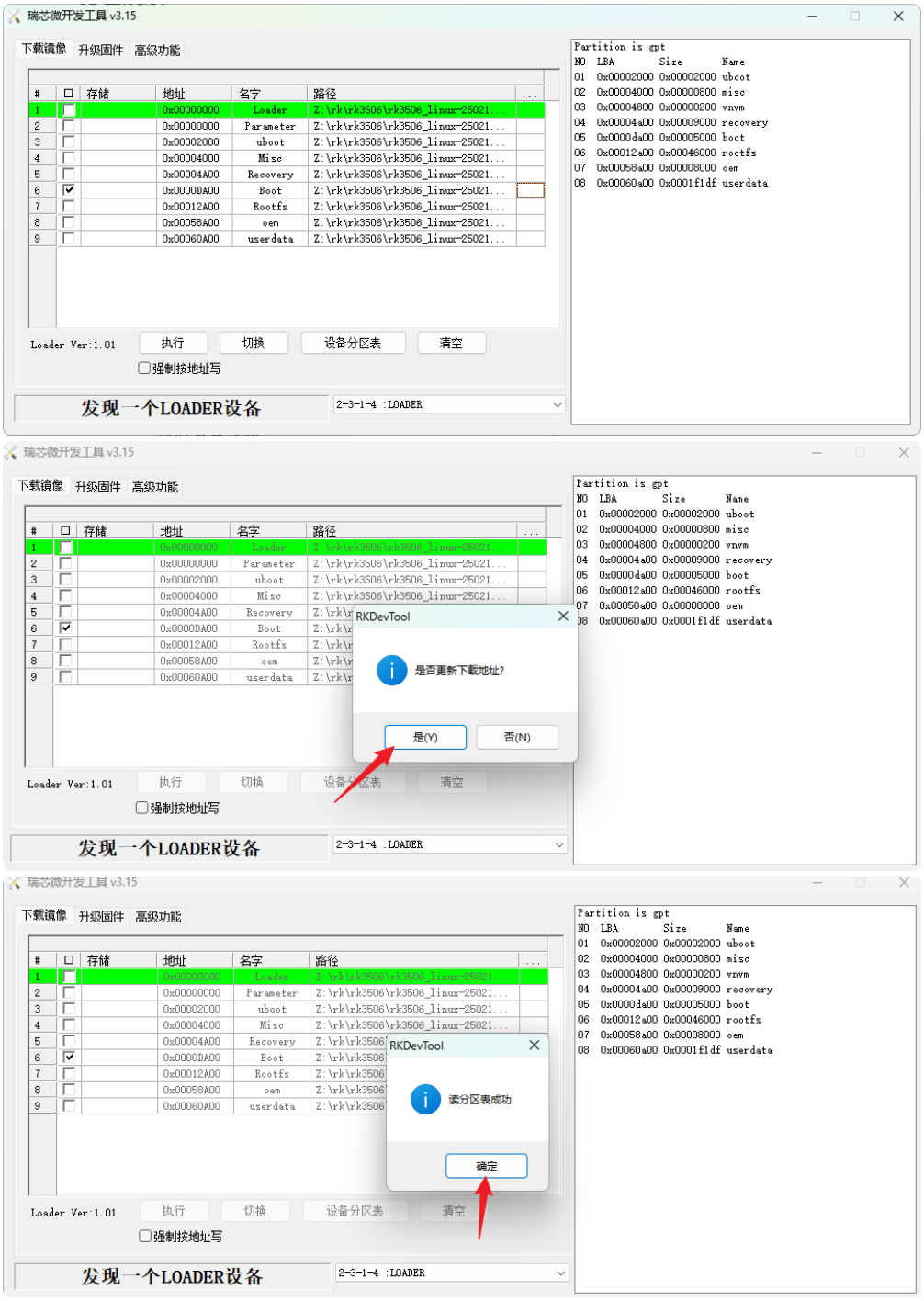
4.1.4.点击升级



点击升级后，右侧会开始下载固件；下载完成后，板子会自动启动。

4.2.分包烧录

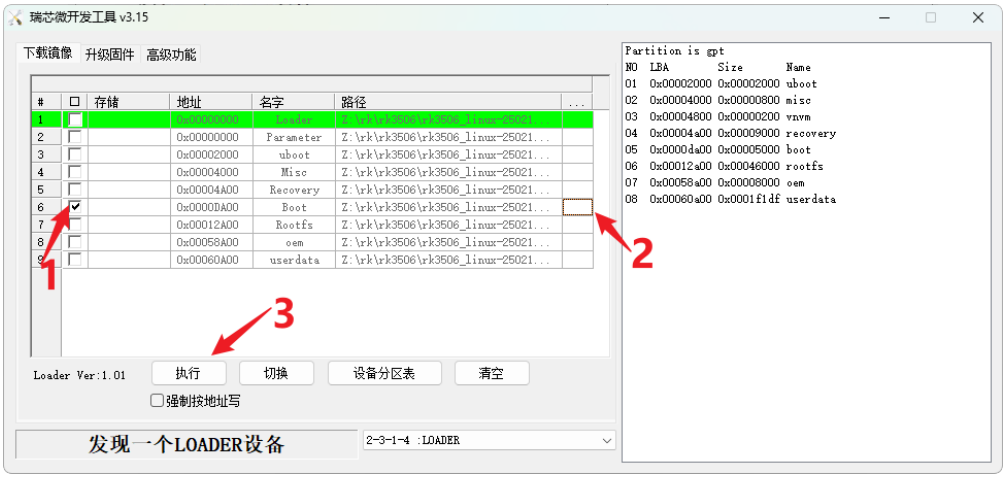
4.2.1.更新下载地址



4.2.2.烧录

1. 在对应的分区前选择对号
2. 然后点击右侧去选择对应的镜像

3. 点击执行即可烧录



5.驱动开发

5.1.CAN

5.1.1.CAN简介

CAN(Controller Area Network)总线，即控制器局域网总线，是一种有效支持分布式控制或实时控制的串行通信网络。CAN总线是一种在汽车上广泛采用的总线协议，被设计作为汽车环境中的微控制器通讯。

5.1.2.DTS 节点配置

公共配置：kernel/arch/arm64/boot/dts/rk3576.dtsi

```Bash
  can0: can@2ac00000 {
    compatible = "rockchip,rk3576-canfd";
    reg = <0x0 0x2ac00000 0x0 0x1000>;
    interrupts = <GIC_SPI 121 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru CLK_CAN0>, <&cru HCLK_CAN0>;
    clock-names = "baudclk", "apb_pclk";
    resets = <&cru SRST_CAN0>, <&cru SRST_H_CAN0>;
    reset-names = "can", "can-apb";
    dmas = <&dmac0 20>;
    dma-names = "rx";
    status = "disabled";
  };

  can1: can@2ac10000 {
    compatible = "rockchip,rk3576-canfd";
    reg = <0x0 0x2ac10000 0x0 0x1000>;
    interrupts = <GIC_SPI 122 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru CLK_CAN1>, <&cru HCLK_CAN1>;
    clock-names = "baudclk", "apb_pclk";
    resets = <&cru SRST_CAN1>, <&cru SRST_H_CAN1>;
    reset-names = "can", "can-apb";
    dmas = <&dmac1 21>;
    dma-names = "rx";
    status = "disabled";
  };

```

配置CAN0

板级配置：ido-evb7608_v1b.dtsi

```Bash
&can0 {
  assigned-clocks = <&cru CLK_CAN0>;
  assigned-clock-rates = <200000000>;
  pinctrl-names = "default";
  pinctrl-0 = <&can0m2_pins>;
  status = "okay";
};

&can1 {
  assigned-clocks = <&cru CLK_CAN1>;
  assigned-clock-rates = <200000000>;
  pinctrl-names = "default";
  pinctrl-0 = <&can1m3_pins>;
  status = "okay";
};

```

通信测试

```Bash
#关闭can0设备
ip link set can0 down

#普通can协议（1.0）
ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on

#查看can信息，波特率等等
ip -details link show can0

#启动can0
ip link set can0 up

#执行candump，阻塞等待can0接收
candump can0

#can格式发送
cansend can0 123#DEADBEEF12345679
```

5.2.Ethernet

公共配置

`kernel/arch/arm64/boot/dts/rk3576.dtsi`

```
gmac0: ethernet@2a220000 {
    compatible = "rockchip,rk3576-gmac", "snps,dwmac-4.20a";
    reg = <0x0 0x2a220000 0x0 0x10000>;
    interrupts = <GIC_SPI 293 IRQ_TYPE_LEVEL_HIGH>,
           <GIC_SPI 298 IRQ_TYPE_LEVEL_HIGH>;
    interrupt-names = "macirq", "eth_wake_irq";
    rockchip,grf = <&sdgmac_grf>;
    rockchip,php_grf = <&ioc_grf>;
    clocks = <&cru CLK_GMAC0_125M_SRC>, <&cru CLK_GMAC0_RMII_CRU>,
       <&cru PCLK_GMAC0>, <&cru ACLK_GMAC0>,
       <&cru CLK_GMAC0_PTP_REF>;
    clock-names = "stmmaceth", "clk_mac_ref",
           "pclk_mac", "aclk_mac",
           "ptp_ref";
    resets = <&cru SRST_A_GMAC0>;
    reset-names = "stmmaceth";
    power-domains = <&power RK3576_PD_SDGMAC>;

    dma-coherent;
    snps,mixed-burst;
    snps,tso;

    snps,axi-config = <&gmac0_stmmac_axi_setup>;
    snps,mtl-rx-config = <&gmac0_mtl_rx_setup>;
    snps,mtl-tx-config = <&gmac0_mtl_tx_setup>;
    status = "disabled";

    mdio0: mdio {
      compatible = "snps,dwmac-mdio";
      #address-cells = <0x1>;
      #size-cells = <0x0>;
    };

    gmac0_stmmac_axi_setup: stmmac-axi-config {
      snps,wr_osr_lmt = <4>;
      snps,rd_osr_lmt = <8>;
      snps,blen = <0 0 0 0 16 8 4>;
    };

    gmac0_mtl_rx_setup: rx-queues-config {
      snps,rx-queues-to-use = <1>;
      queue0 {};
    };

    gmac0_mtl_tx_setup: tx-queues-config {
```

```
        snps,tx-queues-to-use = <1>;
        queue0 {};
      };
    };
```

板级的配置

Bash

```
&gmac0 {
  /* Use rgmii-rxid mode to disable rx delay inside Soc */
  phy-mode = "rgmii";
  clock_in_out = "output";

  snps,reset-gpio = <&gpio0 RK_PC7 GPIO_ACTIVE_LOW>;
  snps,reset-active-low;
  /* Reset time is 20ms, 100ms for rtl8211f */
  snps,reset-delays-us = <0 20000 100000>;

  pinctrl-names = "default";
  pinctrl-0 = <&eth0m0_miim
         &eth0m0_tx_bus2
         &eth0m0_rx_bus2
         &eth0m0_rgmii_clk
         &eth0m0_rgmii_bus >;
         //&ethm0_clk0_25m_out>;

  tx_delay = <0x1a>;
  rx_delay = <0x26>;

  phy-handle = <&rgmii_phy0>;
  status = "okay";
};
```

5.3.LCD

打开 LVDS

Bash

```
// SPDX-License-Identifier: (GPL-2.0+ OR MIT)
/*
 * Copyright (c) 2024 Rockchip Electronics Co., Ltd.
 *
 */

/dts-v1/;

#include "ido-evb7608-v1b.dtsi"

#include <dt-bindings/gpio/gpio.h>
#include <dt-bindings/pwm/pwm.h>
#include <dt-bindings/pinctrl/rockchip.h>
#include <dt-bindings/display/drm_mipi_dsi.h>
#include <dt-bindings/display/rockchip_vop.h>

/{
  backlight: backlight {
    compatible = "pwm-backlight";
    pwms = <&pwm2_8ch_5 0 25000 0>;
    pinctrl-0 = <&backligth_on>;
    enable-gpios = <&gpio2 RK_PB2 GPIO_ACTIVE_HIGH>;
    brightness-levels = <
        0  20  20  21  21  22  22  23
       23  24  24  25  25  26  26  27
       27  28  28  29  29  30  30  31
       31  32  32  33  33  34  34  35
       35  36  36  37  37  38  38  39
       40  41  42  43  44  45  46  47
       48  49  50  51  52  53  54  55
       56  57  58  59  60  61  62  63
       64  65  66  67  68  69  70  71
       72  73  74  75  76  77  78  79
       80  81  82  83  84  85  86  87
       88  89  90  91  92  93  94  95
       96  97  98  99 100 101 102 103
      104 105 106 107 108 109 110 111
      112 113 114 115 116 117 118 119
      120 121 122 123 124 125 126 127
      128 129 130 131 132 133 134 135
      136 137 138 139 140 141 142 143
      144 145 146 147 148 149 150 151
      152 153 154 155 156 157 158 159
      160 161 162 163 164 165 166 167
      168 169 170 171 172 173 174 175
```

```
        176 177 178 179 180 181 182 183
        184 185 186 187 188 189 190 191
        192 193 194 195 196 197 198 199
        200 201 202 203 204 205 206 207
        208 209 210 211 212 213 214 215
        216 217 218 219 220 221 222 223
        224 225 226 227 228 229 230 231
        232 233 234 235 236 237 238 239
        240 241 242 243 244 245 246 247
        248 249 250 251 252 253 254 255
      >;
      default-brightness-level = <200>;
    };
};

&pwm2_8ch_5 {
    status = "okay";
    pinctrl-0 = <&pwm2m1_ch5>;
};

/*
 * mipidcphy0 needs to be enabled
 * when dsi is enabled
 */
&dsi {
    status = "okay";
    //rockchip,lane-rate = <1000>;
    dsi_panel: panel@0 {
      status = "okay";
      compatible = "simple-panel-dsi";
      reg = <0>;
      power-supply = <&vcc3v3_lcd_n>;
      backlight = <&backlight>;
      reset-delay-ms = <10>;
      enable-delay-ms = <10>;
      prepare-delay-ms = <10>;
      unprepare-delay-ms = <10>;
      disable-delay-ms = <60>;
      init-delay-ms = <60>;
      width-mm = <68>;
      height-mm = <121>;
      dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
              MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_NO_EOT_PACKET)>;
      dsi,format = <MIPI_DSI_FMT_RGB888>;
      dsi,lanes  = <4>;
      panel-init-sequence = [
        29 00 02 27 AA
        29 00 02 48 02
```

```
        29 00 02 B6 20
        29 00 02 01 80
        29 00 02 02 38
        29 00 02 03 47
        29 00 02 04 84//hfp
        29 00 02 05 32//hsync
        29 00 02 06 58//hbp
        29 00 02 07 00
        29 00 02 08 04//vfp
        29 00 02 09 20//vsync
        29 00 02 0A 09//vbp
        29 00 02 0B 82
        29 00 02 0C 12
        29 00 02 0D 01
        29 00 02 0E 80
        29 00 02 0F 20
        29 00 02 10 20
        29 00 02 11 03
        29 00 02 12 1B
        29 00 02 13 63//53 //寄偶
        29 00 02 14 01
        29 00 02 15 23
        29 00 02 16 40
        29 00 02 17 00
        29 00 02 18 01
        29 00 02 19 23
        29 00 02 1A 40
        29 00 02 1B 00
        29 00 02 1E 46
        29 00 02 51 30
        29 00 02 1F 10
        //29 00 02 2A 4D //彩条
        29 00 02 2A 01

        05 78 01 11
        05 14 01 29
      ];

      panel-exit-sequence = [
        05 00 01 28
        05 00 01 10
      ];

      disp_timings0: display-timings {
        native-mode = <&dsi_timing0>;
        dsi_timing0: timing0 {
          clock-frequency = <148500000>;
          hactive = <1920>;
```

```
            vactive = <1080>;

            hfront-porch = <0x84>;
            hsync-len = <0x3c>;
            hback-porch = <0x58>;

            vfront-porch = <0x4>;
            vsync-len = <0x20>;
            vback-porch = <0x09>;

            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
          };
        };

        ports {
          #address-cells = <1>;
          #size-cells = <0>;

          port@0 {
            reg = <0>;
            panel_in_dsi: endpoint {
              remote-endpoint = <&dsi_out_panel>;
            };
          };
        };
      };

      ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@1 {
          reg = <1>;
          dsi_out_panel: endpoint {
            remote-endpoint = <&panel_in_dsi>;
          };
        };
      };

    };

    &dsi_in_vp0 {
      status = "disabled";
    };
```

```
&dsi_in_vp1 {
  status = "okay";
};

&dsi_in_vp2 {
  status = "disabled";
};

&dsi_in_vopl {
  status = "disabled";
};

&route_dsi {
  status = "okay";
};

&hdmi {
  status = "okay";
  enable-gpios = <&gpio1 RK_PD5 GPIO_ACTIVE_HIGH>;
  rockchip,sda-falling-delay-ns = <360>;
};

&hdmi_in_vp0 {
  status = "okay";
};

&hdptxphy_hdmi {
  status = "okay";
};

&route_hdmi {
  status = "okay";
  connect = <&vp0_out_hdmi>;
};
```

关闭 LVDS

```Bash
&dsi {
  status = "disabled";
  ...
}

&dsi_in_vp1 {
  status = "okay";
};

&route_dsi {
  status = "okay";
};
```

把这些状态修改为disabled即为关闭LVDS显示。

5.4.RTC

IDO-EVB7608-V1　采用 HYM8563 作为RTC(*Real Time Clock*），需要接入 RTC 电池给 RTC 芯片供电才可以保证在短时间系统断电后 RTC 能正常运行。

DTS配置参考：`kernel/arch/arm64/boot/dts/ido-evb7608_v1b.dtsi`

驱动参考：`kernel/drivers/rtc/rtc-hym8563.c`

Linux下的rtc使用方法为：

```Bash
#同步网络时间
$ ntpdate cn.pool.ntp.org

#查看当前 RTC 的日期和时间:
$ cat /sys/class/rtc/rtc0/date
2025-05-27

$ cat /sys/class/rtc/rtc0/time
10:10:30

$ 查看系统时间
$ date
Tue May 27 18:10:41 CST 2025

#设置系统时间
$ date -s "2024-10-09 14:02:30"

#将rtc时间调整为与目前的系统时间一致
$ hwclock -w

#获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
$ hwclock -r
2024-10-09 14:02:35.945604+00:00

#将rtc时间设置为系统时间
hwclock --systohc
```

5.5.UART

DTS配置

```Bash
&uart3 {
    status = "okay";
    pinctrl-names = "default";
    pinctrl-0 = <&uart3m0_xfer>;
};
```

配置好串口后，硬件接口对应软件上的节点为：

```Bash
UART3:   /dev/ttyS3
```

收发测试

使用microcom可以进行串口收发测试，命令如下：

```Bash
root@rk3576:~# microcom -s 115200 -p /dev/ttyS3
```

注意：测试完成，按Ctrl+x退出。