

IDO-EVB3562-V2 Linux开发手册

1 上手指南

1.介绍

2 Linux开发

2.1.获取 SDK

2.2.校验md5值

2.3.解压

2.4.SDK目录介绍

2.5.配置文件介绍

2.6 分区说明

3.编译

3.1.选择项目

3.2.自动编译

3.3.分步编译

3.3.1.编译uboot

3.3.2.编译kernel

3.3.3.编译buildroot系统

3.3.4.编译ubuntu镜像

3.3.5.编译debian镜像

3.3.6.编译recovery

3.3.7.固件打包

4 烧录说明

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

4.1.4.点击升级

4.2.分包烧录

4.2.1.更新下载地址

4.2.2.烧录

5.驱动开发

5.1.CAN

5.1.1.CAN简介

5.1.2.DTS 节点配置

配置CAN0

通信测试

5.2.Ethernet

公共配置

板级的配置

5.3.LCD

开启MIPI显示

关闭MIPI显示

注意事项

调试方法

5.4.RTC

5.5.UART

DTS配置

收发测试



www.industio.cn

IDO-EVB3562-V2

Linux开发手册

深圳触觉智能科技有限公司

www.industio.cn

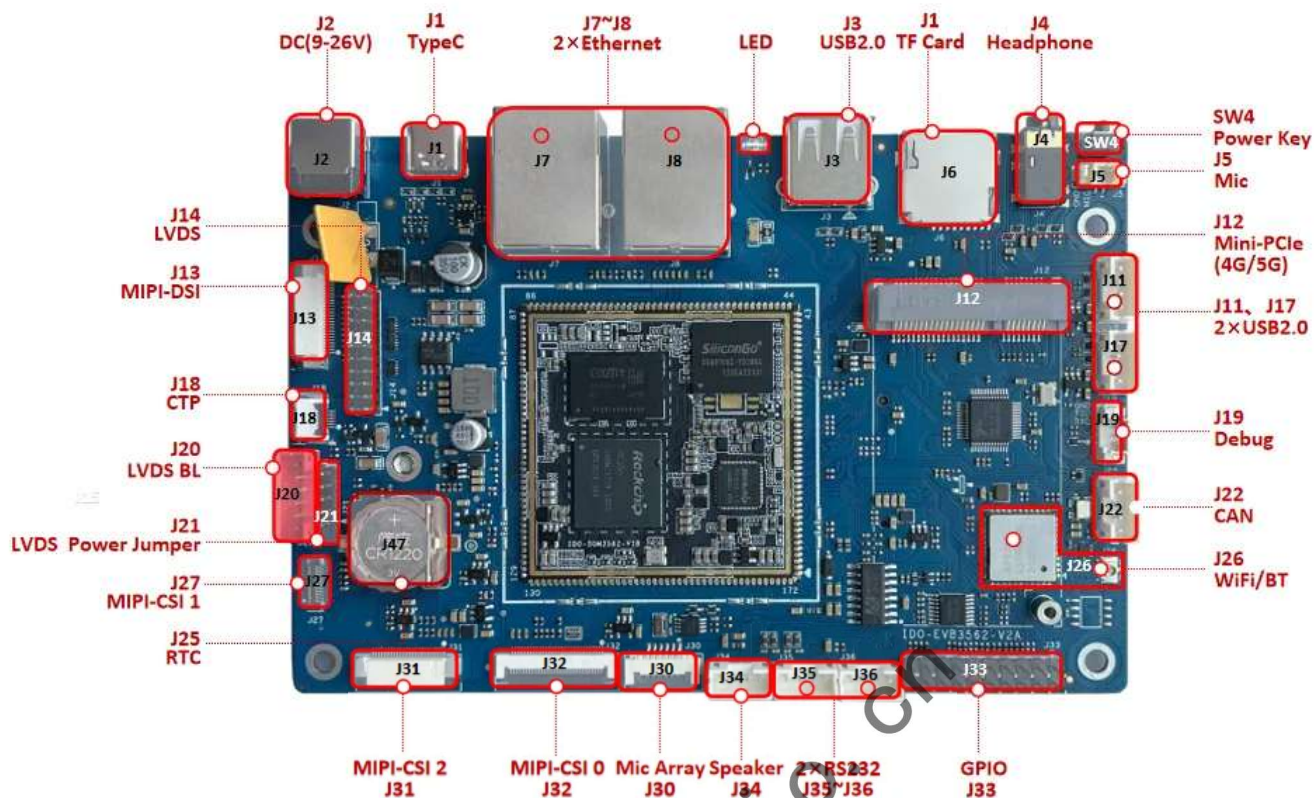
文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V2A	创建文档	MHK	IDO	2025/04/01
V1.1	V2A	优化文档	MHK	IDO	2025/04/25

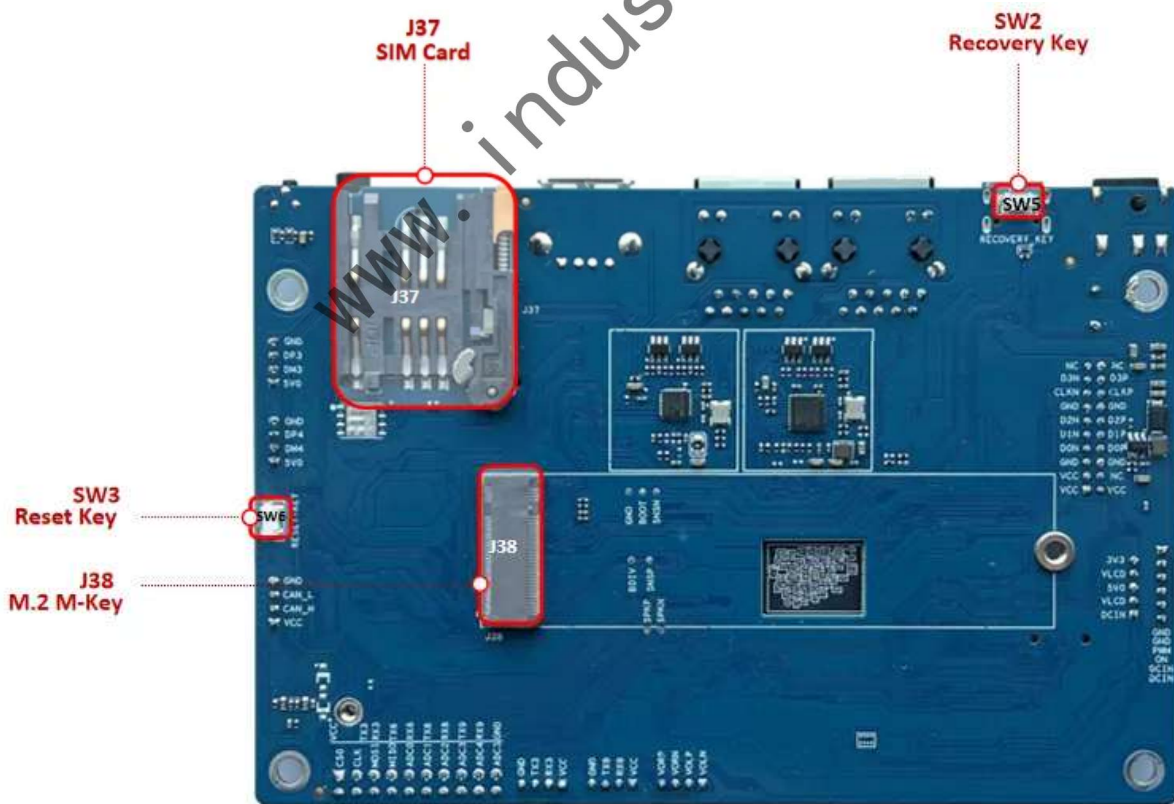
1 上手指南

1.介绍

IDO-EVB3562是搭载 瑞芯微的RK3562 SoC是一款高性能、低功耗的四核应用处理器，专为消费电子应用而精心打造。它采用了四核ARMv8架构的ARM Cortex-A53处理器，主频高达2.0GHz，集成NPU（1TOPS算力）和Mali-G52 GPU，支持工业级与商业级应用场景。



IDO-EVB3562-V2A 正面



IDO-EVB3562-V2A 背面

2 Linux开发

2.1.获取 SDK

下载路径：百度网盘/Linux/软件资料/SDK/

2.2.校验md5值

```
1  $ md5sum rk3562_linux5.10_rkr4_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~/evb3562为例

```
1  $ mkdir ~/evb3562
2  $ cd ~/evb3562
3  $ tar -xvf rk3562_linux5.10_rkr4_v1.tar.gz -C ./
4
5  $ cd rk3562_linux5.10_rkr4_v1
6
7  #查看分支(默认分支为ido-evb3562)
8  $ git branch
9  * ido-evb3562
10     ido-evb3562-preempt-rt
11     ido-evb3562-xenomai-rt
12
13  #获取代码
14  $ git reset --hard
15  正在更新文件： 20% (42627/212711)
```

2.4.SDK目录介绍

```
1  $ tree -L 1
2  .
3  ├── app
4  ├── buildroot
5  ├── build.sh -> device/rockchip/common/scripts/build.sh
6  ├── common -> device/rockchip/common
7  ├── Copyright_Statement.md -> docs/licenses/LICENSE
8  ├── debian
9  ├── device
10 ├── docker.sh
11 ├── docs
12 ├── external
13 ├── hal
14 ├── kernel -> kernel-5.10
15 ├── kernel-5.10
16 ├── Makefile -> device/rockchip/common/Makefile
17 ├── output
18 ├── prebuilts
19 ├── README.md -> device/rockchip/common/README.md
20 ├── rkbin
21 ├── rkflash.sh -> device/rockchip/common/scripts/rkflash.sh
22 ├── rockdev -> output/firmware
23 ├── rootfs
24 ├── rtos
25 ├── tools
26 ├── u-boot
27 └── yocto
```

```
1  app: 存放上层应用 APP, 主要是一些应用Demo。
2  buildroot: 基于 Buildroot (2021) 开发的根文件系统。
3  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。
4  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。
5  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。
6  kernel: 存放 Kernel 开发的代码。
7  output: 存放每次生成的固件信息、编译信息、XML、主机环境等。
8  prebuilts: 存放交叉编译工具链。
9  rkbin: 存放 Rockchip 相关二进制和工具。
10 rockdev: 存放编译输出固件, 实际软链接到 output/firmware 。
11 tools: 存放 Linux 和 Window 操作系统下常用工具。
12 u-boot: 存放基于 v2017.09 版本进行开发的 U-Boot 代码。
13 rtos: 存放rtos相关源码。
14 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/rk3562` 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。ido-evb3562-v2a*_defconfig 作为基础的配置。其他版型的配置文件会引用 ido-evb3562-v2a*_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：

```
1  RK_WIFIBT_CHIP="AP6256"
2  RK_UBOOT_SPL=y
3  RK_KERNEL_CFG="rockchip_linux_defconfig"           # kernel 编译配置
4  RK_KERNEL_DTS_NAME="ido-evb3562-v2a-mipi-800x1280" # 板型 dts 名称
5  RK_PARAMETER="parameter-evb3562.txt"               # 打包使用的分区表
6  RK_USE_FIT_IMG=y
7  RK_ROOTFS_SYSTEM_BUILDROOT=y                       #使用的rootfs类型
```

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：

```
1  FIRMWARE_VER: 1.0
2  MACHINE_MODEL: RK3562
3  MACHINE_ID: 007
4  MANUFACTURER: RK3562
5  MAGIC: 0x5041524B
6  ATAG: 0x00200800
7  MACHINE: 0xffffffff
8  CHECK_MASK: 0x80
9  PWR_HLD: 0,0,A,0,1
10 TYPE: GPT
11 GROW_ALIGN: 0
12 CMDLINE: mtdparts=:0x00002000@0x00004000(uboot),0x00002000@0x00006000(misc),0x00020000@0x00008000(boot),0x00040000@0x00028000(recovery),0x00010000@0x00068000(backup),0x00400000@0x00078000(userdata),0x00040000@0x00478000(overlay),-@0x004B8000(rootfs:grow)
13 uuid:rootfs=614e0000-0000-4b53-8000-1d28000054a9
```

CMDLINE 属性是我们关注的地方，以 uboot 为例，0x00002000@0x00002000(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 |
1  mhkwork@dell-PowerEdge-R430:~/work/rk/rk3562/rk3562_linux-250415$ ./build.
   sh lunch
2  Log colors: message notice warning error fatal
3
4  Log saved at /mnt/5c953a98-9ee6-45b9-8cea-a2898eb313d7/mhk/rk/rk3562/rk356
   2_linux-250415/output/sessions/2025-04-25_17-50-49
5  Pick a defconfig:
6
7  1. rockchip_defconfig
8  2. ido-evb3562-v2a-lvds-1280x800-buildroot_defconfig
9  3. ido-evb3562-v2a-lvds-1280x800-debian_defconfig
10 4. ido-evb3562-v2a-lvds-1280x800-ubuntu_defconfig
11 5. ido-evb3562-v2a-mipi-800x1280-buildroot_defconfig
12 6. ido-evb3562-v2a-mipi-800x1280-debian_defconfig
13 7. ido-evb3562-v2a-mipi-800x1280-ubuntu_defconfig
14 8. rockchip_rk3562_dictpen_test3_v20_defconfig
15 9. rockchip_rk3562_evb1_lp4x_v10_amp_defconfig
16 10. rockchip_rk3562_evb1_lp4x_v10_defconfig
17 11. rockchip_rk3562_evb2_ddr4_v10_amp_defconfig
18 12. rockchip_rk3562_evb2_ddr4_v10_defconfig
19 13. rockchip_rk3562_robot_evb1_lp4x_v10_defconfig
20 14. rockchip_rk3562_robot_evb2_ddr4_v10_defconfig
21 Which would you like? [1]: 2
22
```

选择对应的数字即可。

注意：

1.项目命名规则介绍：

```
▼ Bash |
1  主板丝印：ido-evb3562-v2a。
2  显示接口及屏幕分辨率：mipi-800x1280, lvds-1280x800。
3  文件系统支持列表：buildroot, debian, Ubuntu。
```

3.2.自动编译

▼

Bash

```
1  $ ./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

```
1  ./build.sh uboot
```

固件编译后生成路径：rockdev/uboot.img

3.3.2.编译kernel

▼

Bash

```
1  ./build.sh kernel
```

固件编译后生成路径：rockdev/boot.img

3.3.3.编译buildroot系统

▼

Bash

```
1  ./build.sh buildroot
```

固件编译后生成路径：rockdev/rootfs.img

3.3.4.编译ubuntu镜像

▼

Bash

```
1  ./build.sh ubuntu
```

3.3.5.编译debian镜像

```
▼ Bash |
1  ./build.sh debian
```

3.3.6.编译recovery

```
▼ Bash |
1  ./build.sh recovery
```

固件编译后生成路径: rockdev/recovery.img

3.3.7.固件打包

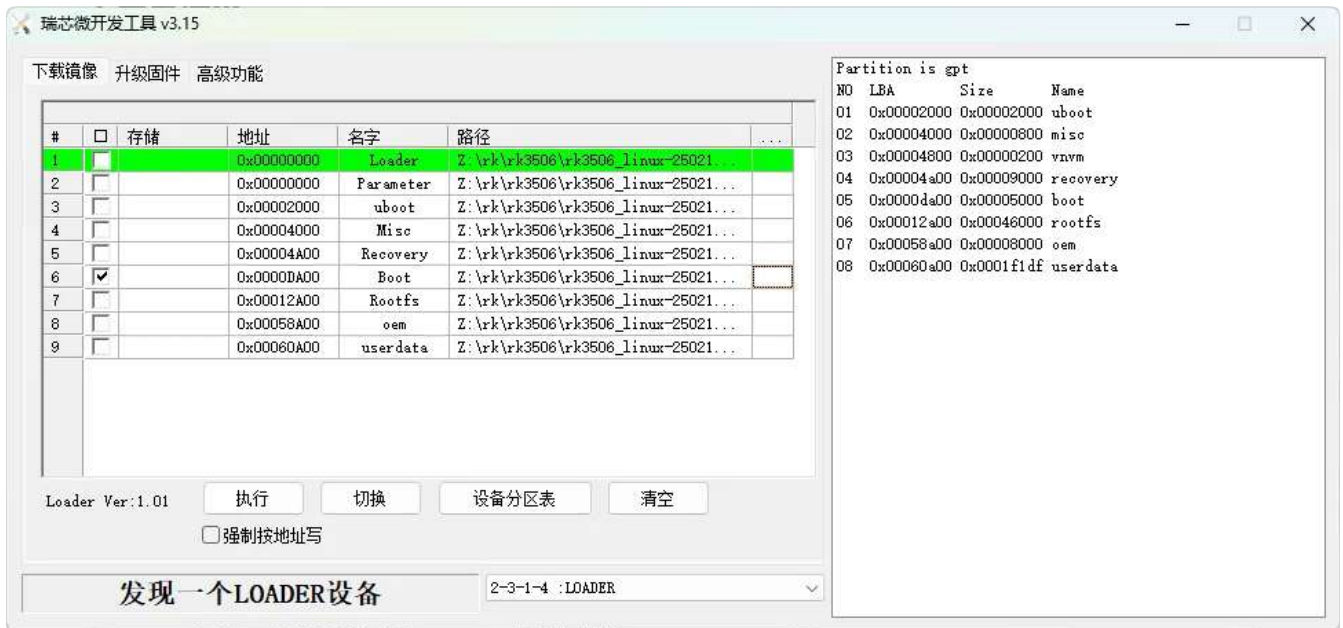
```
▼ Bash |
1  ./build.sh updateimg
```

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-EVB3562-V2A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

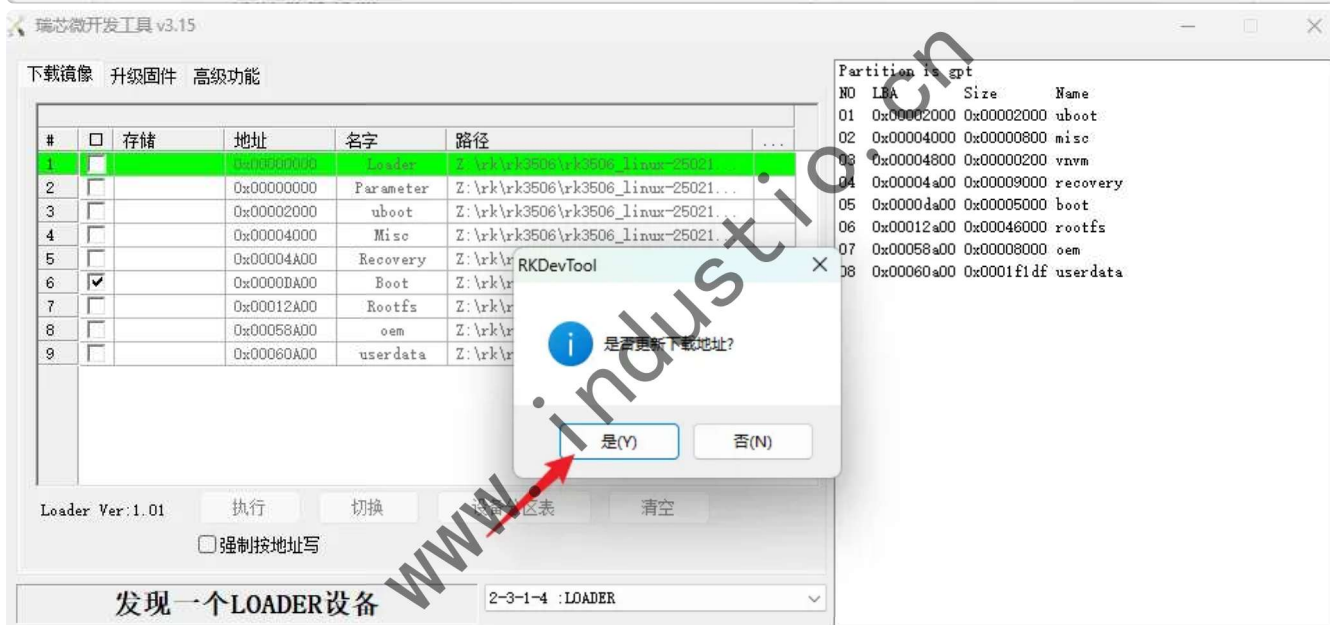
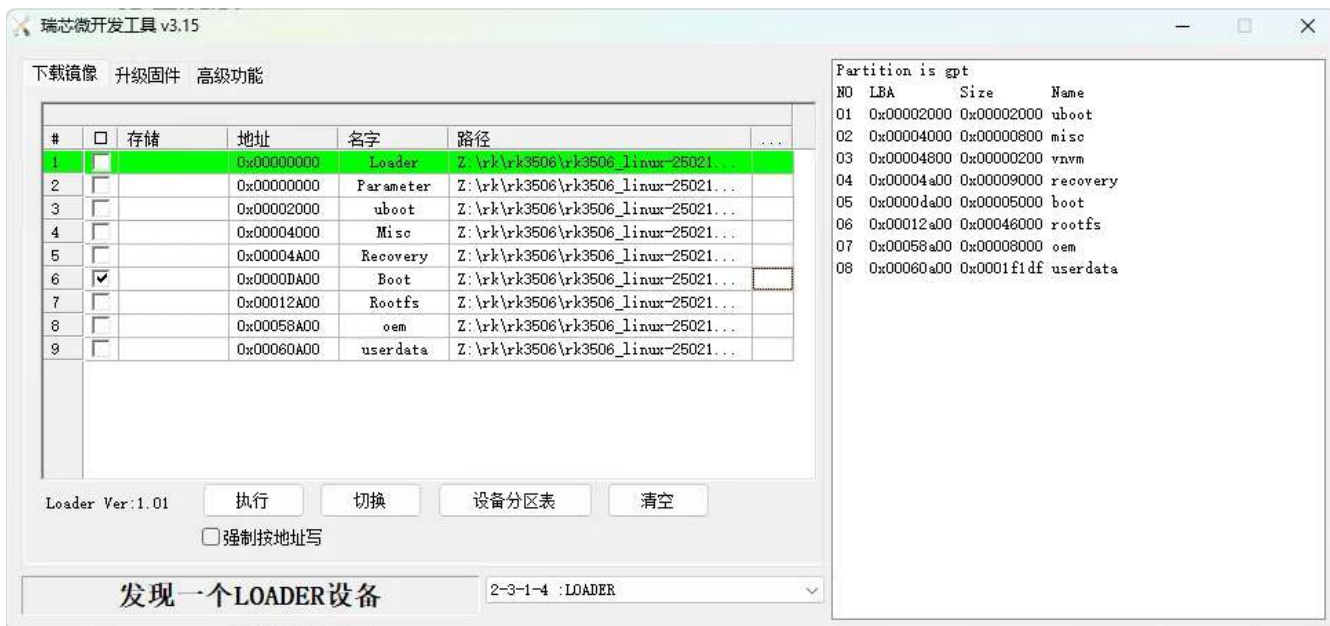
4.1.4. 点击升级

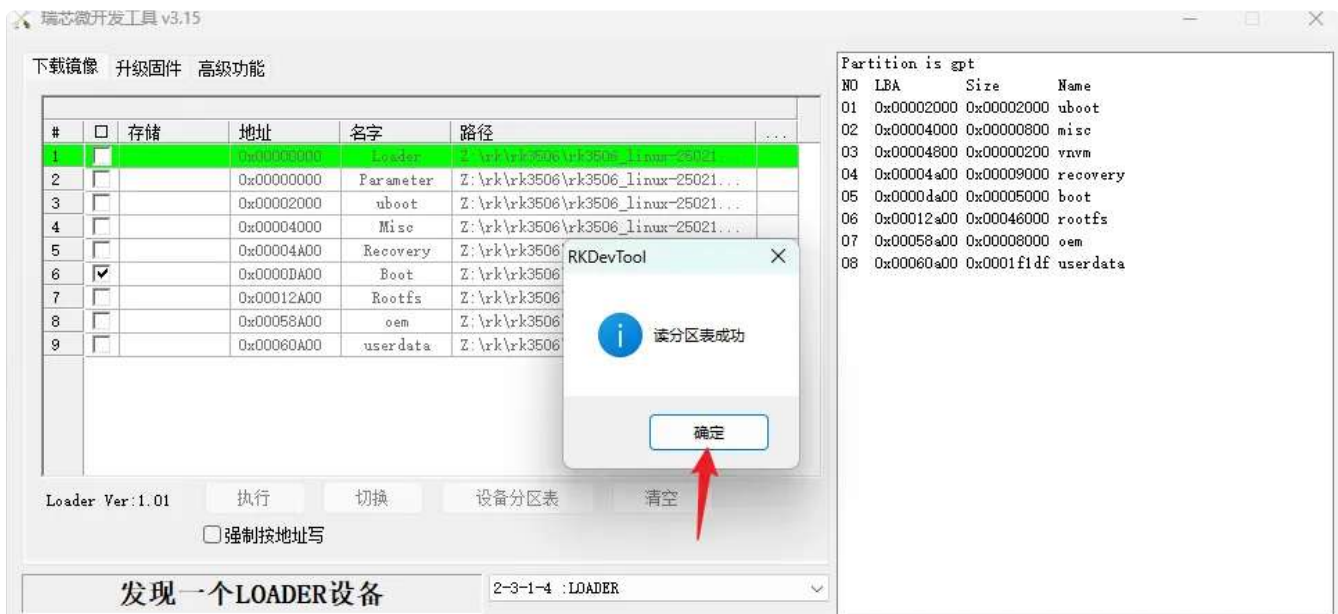


点击升级后，右侧会开始下载固件；下载完成后，板子会自动启动。

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/rk3562.dtsi

```
1 can0: can@ff600000 {
2     compatible = "rockchip,rk3562-can";
3     reg = <0x0 0xff600000 0x0 0x1000>;
4     interrupts = <GIC_SPI 64 IRQ_TYPE_LEVEL_HIGH>;
5     clocks = <&cru CLK_CAN0>, <&cru PCLK_CAN0>;
6     clock-names = "baudclk", "apb_pclk";
7     resets = <&cru SRST_CAN0>, <&cru SRST_P_CAN0>;
8     reset-names = "can", "can-apb";
9     status = "disabled";
10 };
11
12 can1: can@ff610000 {
13     compatible = "rockchip,rk3562-can";
14     reg = <0x0 0xff610000 0x0 0x1000>;
15     interrupts = <GIC_SPI 65 IRQ_TYPE_LEVEL_HIGH>;
16     clocks = <&cru CLK_CAN1>, <&cru PCLK_CAN1>;
17     clock-names = "baudclk", "apb_pclk";
18     resets = <&cru SRST_CAN1>, <&cru SRST_P_CAN1>;
19     reset-names = "can", "can-apb";
20     status = "disabled";
21 };
```

配置CAN0

板级配置：ido-evb3562-v2a.dtsi

```
1 &can0 {  
2     status = "okay";  
3     pinctrl-names = "default";  
4     pinctrl-0 = <&can0m2_pins>;  
5 };
```

通信测试

```
1 #关闭can0设备  
2 ip link set can0 down  
3  
4 #普通can协议 (1.0)  
5 ip link set can0 type can bitrate 125000 triple-sampling on  
6  
7 #查看can信息, 波特率等等  
8 ip -details link show can0  
9  
10 #启动can0  
11 ip link set can0 up  
12  
13 #执行candump, 阻塞等待can0接收  
14 candump can0  
15  
16 #can格式发送  
17 cansend can0 123#1122334455667788
```

5.2.Ethernet

公共配置

```
kernel/arch/arm64/boot/dts/rk3562.dtsi
```

```

1  gmac0: ethernet@ffa80000 {
2      compatible = "rockchip,rk3562-gmac", "snps,dwmac-4.20a";
3      reg = <0x0 0xffa80000 0x0 0x10000>;
4      interrupts = <GIC_SPI 73 IRQ_TYPE_LEVEL_HIGH>,
5                  <GIC_SPI 70 IRQ_TYPE_LEVEL_HIGH>;
6      interrupt-names = "macirq", "eth_wake_irq";
7      rockchip,grf = <&sys_grf>;
8      rockchip,php_grf = <&ioc_grf>;
9      clocks = <&cru CLK_GMAC_125M_CRU_I>, <&cru CLK_GMAC_50M_CRU_I>,
10             <&cru PCLK_GMAC>, <&cru ACLK_GMAC>;
11      clock-names = "stmmaceth", "clk_mac_ref",
12                  "pclk_mac", "aclk_mac";
13      resets = <&cru SRST_A_GMAC>;
14      reset-names = "stmmaceth";
15      rockchip,csu = <&csu CSU_GMAC_ACLK>, <&csu CSU_GMAC_PCLK>;
16      rockchip,csu-names = "aclk", "pclk";
17
18      snps,mixed-burst;
19      snps,tso;
20
21      snps,axi-config = <&gmac0_stmmac_axi_setup>;
22      snps,mtl-rx-config = <&gmac0_mtl_rx_setup>;
23      snps,mtl-tx-config = <&gmac0_mtl_tx_setup>;
24      status = "disabled";
25
26  mdio0: mdio {
27      compatible = "snps,dwmac-mdio";
28      #address-cells = <0x1>;
29      #size-cells = <0x0>;
30  };
31
32  gmac0_stmmac_axi_setup: stmmac-axi-config {
33      snps,wr_osr_lmt = <4>;
34      snps,rd_osr_lmt = <8>;
35      snps,blen = <0 0 0 0 16 8 4>;
36  };
37
38  gmac0_mtl_rx_setup: rx-queues-config {
39      snps,rx-queues-to-use = <1>;
40      queue0 {};
41  };
42
43  gmac0_mtl_tx_setup: tx-queues-config {
44      snps,tx-queues-to-use = <1>;
45      queue0 {};

```

```
46     };  
47 };
```

板级的配置

```
1 &gmac0 {  
2     phy-mode = "rgmii";  
3     clock_in_out = "input";  
4  
5     snps,reset-gpio = <&gpio3 RK_PC1 GPIO_ACTIVE_LOW>;  
6     snps,reset-active-low;  
7     /* Reset time is 20ms, 100ms for rtl8211f */  
8     snps,reset-delays-us = <0 50000 100000>;  
9  
10    //RTL8811  
11    // tx_delay = <0x3c>;  
12    // rx_delay = <0x2f>;  
13  
14    tx_delay = <0x2c>;  
15    rx_delay = <0x35>;  
16  
17    pinctrl-names = "default";  
18    pinctrl-0 = <&rgmiim0_miim,  
19                &rgmiim0_tx_bus2  
20                &rgmiim0_rx_bus2  
21                &rgmiim0_rgmii_clk  
22                &rgmiim0_rgmii_bus  
23                &rgmiim0_clk>;  
24  
25    phy-handle = <&rgmii_phy>;  
26    status = "okay";  
27 };  
28  
29 &mdio0 {  
30     rgmii_phy: phy@1 {  
31         compatible = "ethernet-phy-ieee802.3-c22";  
32         reg = <0x1>;  
33     };  
34 };
```

5.3.LCD

IDO-EVB3562-V2A 有一路 MIPI 和 一路 LVDS显示接口

开启MIPI显示

www.industio.cn

```

1 // SPDX-License-Identifier: (GPL-2.0+ OR MIT)
2 /*
3  * Copyright (c) 2022 Rockchip Electronics Co., Ltd.
4  *
5  */
6
7 #include <dt-bindings/display/drm_mipi_dsi.h>
8 #include <dt-bindings/gpio/gpio.h>
9 #include <dt-bindings/pinctrl/rockchip.h>
10 #include "ido-evb3562-v2a.dtsi"
11
12 / {
13     backlight: backlight {
14         compatible = "pwm-backlight";
15         pwms = <&pwm4 0 25000 0>;
16         brightness-levels = <
17             0 20 20 21 21 22 22 23
18             23 24 24 25 25 26 26 27
19             27 28 28 29 29 30 30 31
20             31 32 32 33 33 34 34 35
21             35 36 36 37 37 38 38 39
22             40 41 42 43 44 45 46 47
23             48 49 50 51 52 53 54 55
24             56 57 58 59 60 61 62 63
25             64 65 66 67 68 69 70 71
26             72 73 74 75 76 77 78 79
27             80 81 82 83 84 85 86 87
28             88 89 90 91 92 93 94 95
29             96 97 98 99 100 101 102 103
30             104 105 106 107 108 109 110 111
31             112 113 114 115 116 117 118 119
32             120 121 122 123 124 125 126 127
33             128 129 130 131 132 133 134 135
34             136 137 138 139 140 141 142 143
35             144 145 146 147 148 149 150 151
36             152 153 154 155 156 157 158 159
37             160 161 162 163 164 165 166 167
38             168 169 170 171 172 173 174 175
39             176 177 178 179 180 181 182 183
40             184 185 186 187 188 189 190 191
41             192 193 194 195 196 197 198 199
42             200 201 202 203 204 205 206 207
43             208 209 210 211 212 213 214 215
44             216 217 218 219 220 221 222 223
45             224 225 226 227 228 229 230 231

```

```

46         232 233 234 235 236 237 238 239
47         240 241 242 243 244 245 246 247
48         248 249 250 251 252 253 254 255
49     >;
50     default-brightness-level = <200>;
51 };
52
53 vcc3v3_lcd_n: vcc3v3-lcd0-n {
54     compatible = "regulator-fixed";
55     regulator-name = "vcc3v3_lcd_n";
56     regulator-boot-on;
57     regulator-state-mem {
58         regulator-off-in-suspend;
59     };
60 };
61 };
62
63 &display_subsystem {
64     status = "okay";
65 };
66
67 &dsi {
68     status = "okay";
69     //rockchip, lane-rate = <1000>;
70     dsi_panel: panel@0 {
71         status = "okay";
72         compatible = "simple-panel-dsi";
73         reg = <0>;
74         power-supply = <&vcc3v3_lcd_n>;
75         pinctrl-names = "default";
76         pinctrl-0 = <&lcd_rst_gpio>;
77         reset-gpios = <&gpio3 RK_PC4 GPIO_ACTIVE_LOW>;
78
79         backlight = <&backlight>;
80         reset-delay-ms = <60>;
81         enable-delay-ms = <60>;
82         prepare-delay-ms = <60>;
83         unprepare-delay-ms = <60>;
84         disable-delay-ms = <60>;
85         init-delay-ms = <60>;
86         dsi, flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
87             MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET)>;
88         dsi, format = <MIPI_DSI_FMT_RGB888>;
89         dsi, lanes = <4>;
90         panel-init-sequence = [
91             29 00 04 FF 98 81 03
92             29 00 02 01 00
93             29 00 02 02 00

```

94	29 00 02 03 73
95	29 00 02 04 13
96	29 00 02 05 00
97	29 00 02 06 0A
98	29 00 02 07 05
99	29 00 02 08 00
100	29 00 02 09 28
101	29 00 02 0a 00
102	29 00 02 0b 00
103	29 00 02 0c 00
104	29 00 02 0d 28
105	29 00 02 0e 00
106	29 00 02 0f 28
107	29 00 02 10 28
108	29 00 02 11 00
109	29 00 02 12 00
110	29 00 02 13 00
111	29 00 02 14 00
112	29 00 02 15 00
113	29 00 02 16 00
114	29 00 02 17 00
115	29 00 02 18 00
116	29 00 02 19 00
117	29 00 02 1a 00
118	29 00 02 1b 00
119	29 00 02 1c 00
120	29 00 02 1d 00
121	29 00 02 1e 40
122	29 00 02 1f 80
123	29 00 02 20 00
124	29 00 02 21 01
125	29 00 02 22 00
126	29 00 02 23 00
127	29 00 02 24 00
128	29 00 02 25 00
129	29 00 02 26 00
130	29 00 02 27 00
131	29 00 02 28 33
132	29 00 02 29 33
133	29 00 02 2a 00
134	29 00 02 2b 00
135	29 00 02 2c 04
136	29 00 02 2d 04
137	29 00 02 2e 05
138	29 00 02 2f 05
139	29 00 02 30 00
140	29 00 02 31 00
141	29 00 02 32 31

142 29 00 02 33 00
143 29 00 02 34 00
144 29 00 02 35 0A
145 29 00 02 36 00
146 29 00 02 37 08
147 29 00 02 38 00
148 29 00 02 39 00
149 29 00 02 3a 00
150 29 00 02 3b 00
151 29 00 02 3c 00
152 29 00 02 3d 00
153 29 00 02 3e 00
154 29 00 02 3f 00
155 29 00 02 40 00
156 29 00 02 41 00
157 29 00 02 42 00
158 29 00 02 43 08
159 29 00 02 44 00
160 29 00 02 50 01
161 29 00 02 51 23
162 29 00 02 52 44
163 29 00 02 53 67
164 29 00 02 54 89
165 29 00 02 55 ab
166 29 00 02 56 01
167 29 00 02 57 23
168 29 00 02 58 45
169 29 00 02 59 67
170 29 00 02 5a 89
171 29 00 02 5b ab
172 29 00 02 5c cd
173 29 00 02 5d ef
174 29 00 02 5e 11
175 29 00 02 5f 02
176 29 00 02 60 08
177 29 00 02 61 0E
178 29 00 02 62 0F
179 29 00 02 63 0C
180 29 00 02 64 0D
181 29 00 02 65 17
182 29 00 02 66 01
183 29 00 02 67 01
184 29 00 02 68 02
185 29 00 02 69 02
186 29 00 02 6a 00
187 29 00 02 6b 00
188 29 00 02 6c 02
189 29 00 02 6d 02

www.industio.cn

190 29 00 02 6e 16
191 29 00 02 6f 16
192 29 00 02 70 06
193 29 00 02 71 06
194 29 00 02 72 07
195 29 00 02 73 07
196 29 00 02 74 02
197 29 00 02 75 02
198 29 00 02 76 08
199 29 00 02 77 0E
200 29 00 02 78 0F
201 29 00 02 79 0C
202 29 00 02 7a 0D
203 29 00 02 7b 17
204 29 00 02 7c 01
205 29 00 02 7d 01
206 29 00 02 7e 02
207 29 00 02 7f 02
208 29 00 02 80 00
209 29 00 02 81 00
210 29 00 02 82 02
211 29 00 02 83 02
212 29 00 02 84 16
213 29 00 02 85 16
214 29 00 02 86 06
215 29 00 02 87 06
216 29 00 02 88 07
217 29 00 02 89 07
218 29 00 02 8A 02
219 29 00 04 FF 98 81 04
220 29 00 02 6E 1A
221 29 00 02 6F 37
222 29 00 02 3A A4
223 29 00 02 8D 1F
224 29 00 02 87 BA
225 29 00 02 B2 D1
226 29 00 02 88 0B
227 29 00 02 38 01
228 29 00 02 39 00
229 29 00 02 B5 02
230 29 00 02 31 25
231 29 00 02 3B 98
232 29 00 04 FF 98 81 01
233 29 00 02 22 0A
234 29 00 02 31 00
235 29 00 02 53 5A
236 29 00 02 55 3D
237 29 00 02 50 9E

```

238      29 00 02 51 99
239      29 00 02 60 06
240
241      29 00 02 62 20
242      29 00 02 A0 00
243      29 00 02 A1 17
244      29 00 02 A2 26
245      29 00 02 A3 13
246      29 00 02 A4 16
247      29 00 02 A5 29
248      29 00 02 A6 1E
249      29 00 02 A7 1F
250      29 00 02 A8 8B
251      29 00 02 A9 1D
252      29 00 02 AA 2A
253      29 00 02 AB 7B
254      29 00 02 AC 1A
255      29 00 02 AD 19
256      29 00 02 AE 4E
257      29 00 02 AF 24
258      29 00 02 B0 29
259      29 00 02 B1 4F
260      29 00 02 B2 5C
261      29 00 02 B3 23
262      29 00 02 C0 00
263      29 00 02 C1 17
264      29 00 02 C2 26
265      29 00 02 C3 13
266      29 00 02 C4 16
267      29 00 02 C5 29
268      29 00 02 C6 1E
269      29 00 02 C7 1F
270      29 00 02 C8 8B
271      29 00 02 C9 1D
272      29 00 02 CA 2A
273      29 00 02 CB 7B
274      29 00 02 CC 1A
275      29 00 02 CD 19
276      29 00 02 CE 4E
277      29 00 02 CF 24
278      29 00 02 D0 29
279      29 00 02 D1 4F
280      29 00 02 D2 5C
281      29 00 02 D3 23
282      29 00 04 FF 98 81 00
283      05 78 01 11
284      05 1E 01 29
285

```

```
];
```

```

286
287 panel-exit-sequence = [
288     05 00 01 28
289     05 00 01 10
290 ];
291
292 disp_timings0: display-timings {
293     native-mode = <&dsi_timing0>;
294     dsi_timing0: timing0 {
295         clock-frequency = <73400000>;
296         hactive = <800>;
297         vactive = <1280>;
298         hfront-porch = <52>;
299         hsync-len = <4>;
300         hback-porch = <12>;
301         vfront-porch = <60>;
302         vsync-len = <18>;
303         vback-porch = <50>;
304         hsync-active = <0>;
305         vsync-active = <0>;
306         de-active = <0>;
307         pixelclk-active = <1>;
308     };
309 };
310
311 ports {
312     #address-cells = <1>;
313     #size-cells = <0>;
314
315     port@0 {
316         reg = <0>;
317         panel_in_dsi: endpoint {
318             remote-endpoint = <&dsi_out_panel>;
319         };
320     };
321 };
322 };
323
324 ports {
325     #address-cells = <1>;
326     #size-cells = <0>;
327
328     port@1 {
329         reg = <1>;
330         dsi_out_panel: endpoint {
331             remote-endpoint = <&panel_in_dsi>;
332         };
333     };

```

```

334     };
335 };
336
337 &route_dsi {
338     status = "okay";
339 };
340
341 &dsi_in_vp0 {
342     status = "okay";
343 };
344
345 &video_phy {
346     status = "okay";
347 };
348
349 &pwm4 {
350     pinctrl-0 = <&pwm4m0_pins>;
351     status = "okay";
352 };
353
354 &i2c3 {
355     status = "okay";
356     gt911@14 {
357         compatible = "goodix,gt9xx";
358         reg = <0x14>;
359         pinctrl-names = "default";
360         pinctrl-0 = <&touch_gpio>;
361         goodix_irq_gpio = <&gpio0 RK_PB6 IRQ_TYPE_LEVEL_LOW>;
362         goodix_rst_gpio = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
363         touchscreen-swapped-x-y;
364         touchscreen-inverted-y;
365         status = "okay";
366     };
367 };
368
369 &pinctrl {
370     touch {
371         touch_gpio: touch-gpio {
372             rockchip,pins =
373                 <0 RK_PB6 RK_FUNC_GPIO &pcfg_pull_up>,
374                 <0 RK_PB5 RK_FUNC_GPIO &pcfg_pull_none>;
375         };
376     };
377
378     lcd {
379         lcd_rst_gpio: lcd-rst-gpio {
380             rockchip,pins = <3 RK_PC4 RK_FUNC_GPIO &pcfg_pull_none>;
381         };

```

```
382     };
383 };
384 };
385
```

关闭MIPI显示

```
1  &dsi {
2      status = "disabled";
3      ...
4  }
5
6  &video_phy {
7      status = "disabled";
8  };
9
10 &dsi_in_vp0 {
11     status = "disabled";
12 };
13
14 &route_dsi {
15     status = "disabled";
16 };
```

把这些状态修改为disabled即为关闭显示。

注意事项

在配置 MIPI DSI的时候，如果出现异常现象，如黑屏，画面拉伸，显示有噪点等，需要注意排查：

1. 显示时序是否配置正确，尤其是 DCLK 的配置。
2. 上下电时序是否正确，在文件 `kernel/drivers/gpu/drm/panel/panel-simple.c` 中的 `panel_simple_prepare` 和 `panel_simple_unprepare` 函数内，调用了设备树中所配置的上下电时序和 gpio 口。

如果选择在 uboot 阶段开启开机 logo，那么还需要排查 `u-boot/drivers/video/drm/rockchip_panel.c` 文件内的 `panel_simple_prepare` 和 `panel_simple_unprepare` 函数。

3. 搭示波器看一下上电时序是否正确，主要是确认 LCD 使能引脚、复位引脚和屏幕上电指令间的时序是否正确。

调试方法

获取系统中正在使用的 Video Port（与所连接的显示控制器） 信息：

Bash

```
1 root@rk3562:/# cat /sys/kernel/debug/dri/0/summary
2 Video Port0: ACTIVE
3     Connector:DSI-1 Encoder: DSI-127
4     bus_format[100a]: RGB888_1X24
5     overlay_mode[0] output_mode[0] color_space[0], eotf:0
6     Display mode: 800x1280p60
7     clk[73400] real_clk[73400] type[48] flag[a]
8     H: 800 852 856 868
9     V: 1280 1340 1358 1408
10    Esmart0-win0: ACTIVE
11    win_id: 0
12    format: XR24 little-endian (0x34325258) SDR[0] color_space[0] glb_alpha
    [0xff]
13    rotate: xmirror: 0 ymirror: 0 rotate_90: 0 rotate_270: 0
14    csc: y2r[0] r2y[0] csc mode[0]
15    zpos: 0
16    src: pos[0, 0] rect[800 x 1280]
17    dst: pos[0, 0] rect[800 x 1280]
18    buf[0]: addr: 0x00000000fa82a000 pitch: 3200 offset: 0
```

一般如果遇到屏幕无法显示的问题，都需要先执行上面的命令去看一下连接状态和分辨率是否正确。

5.4.RTC

IDO-EVB3562-V2 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-evb3562-v2a.dtsi`

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

Linux下的rtc使用方法为：

```
1  #同步网络时间
2  $ ntpdate cn.pool.ntp.org
3
4  #查看当前 RTC 的日期和时间:
5  $ cat /sys/class/rtc/rtc1/date
6  2021-01-01
7
8  $ cat /sys/class/rtc/rtc1/time
9  17:18:14
10
11 $ 查看系统时间
12 $ date
13 Wed Mar 12 18:46:59 CST 2025
14
15 #设置系统时间
16 $ date -s "2024-10-09 14:02:30"
17
18 #将rtc时间调整为与目前的系统时间一致
19 $ hwclock -w
20
21 #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
22 $ hwclock -r
23 2024-10-09 14:02:35.945604+00:00
24
25 #将rtc时间设置为系统时间
26 hwclock --systohc
```

5.5.UART

DTS配置

```
1 &uart3 {
2     status = "okay";
3     pinctrl-names = "default";
4     pinctrl-0 = <&uart3m1_xfer>;
5 };
```

配置好串口后，硬件接口对应软件上的节点为：

```
▼ Bash |
1  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 |
1  root@rk3562:~# microcom -s 115200 -p /dev/ttyS3
```

注意：测试完成，按Ctrl+x退出。

www.industio.cn