

IDO-EVB5301-V1 Linux开发手册



IDO-EVB5301-V1 Linux开发手册

深圳触觉智能科技有限公司

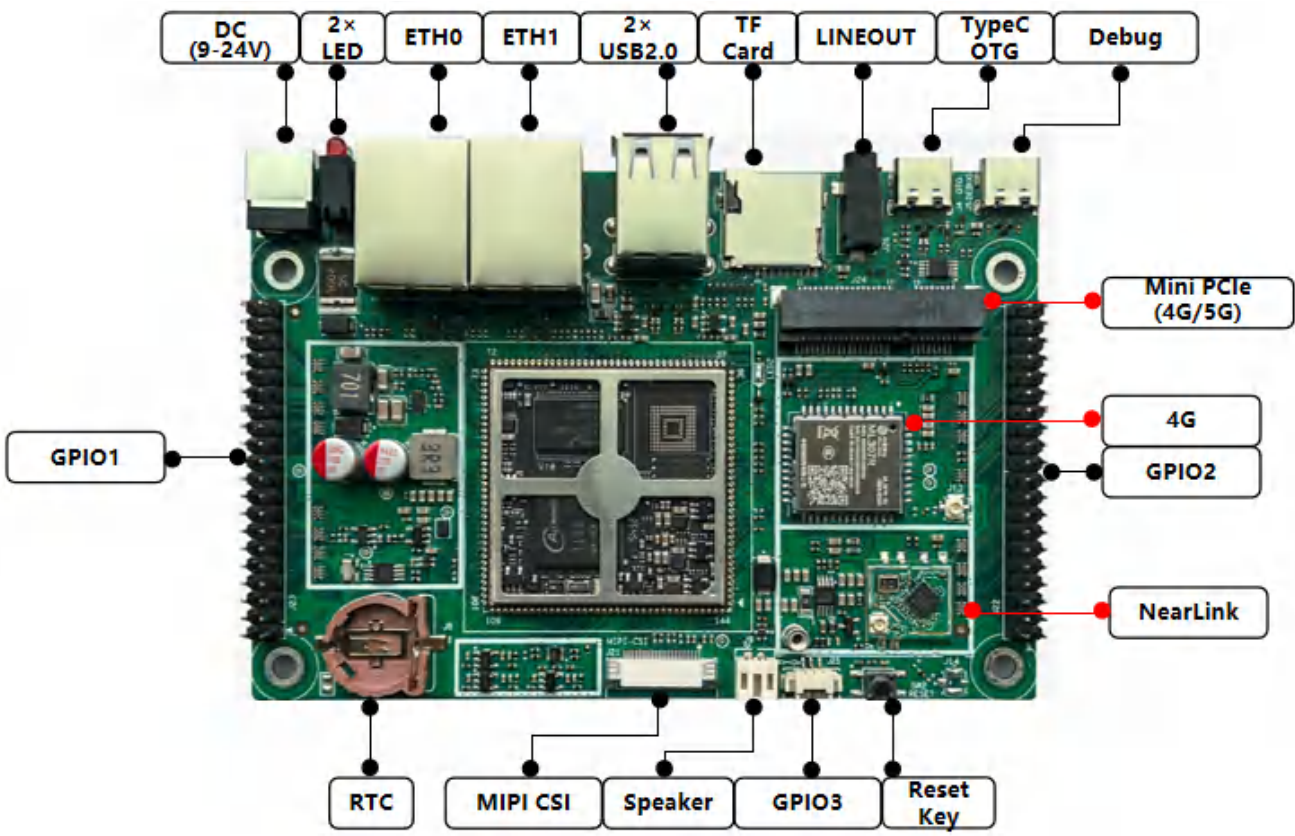
www.industio.cn

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1A	创建文档	LJH	IDO	2025/10/21

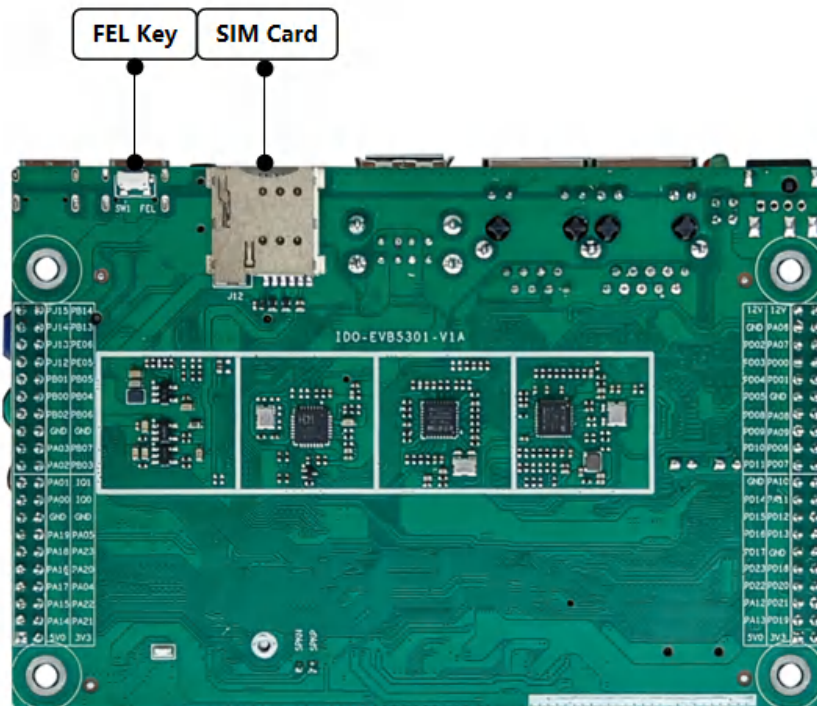
1 上手指南

1.介绍

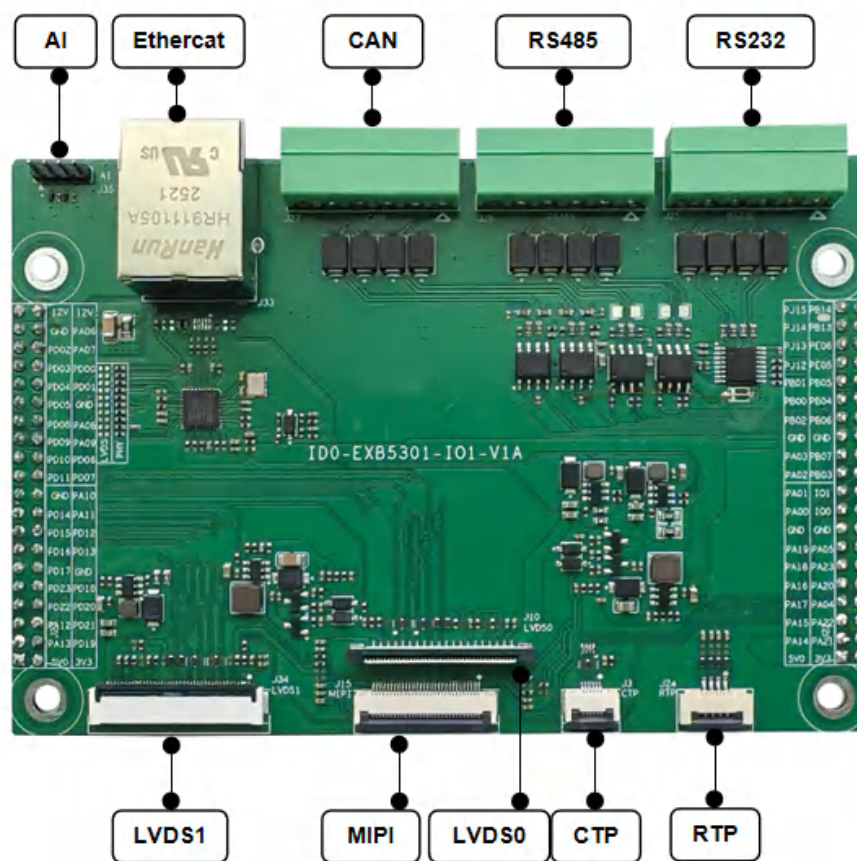
IDO-EVB5301-V1采用 Allwinner T153（四核 Cortex-A7, 主频最高1.6GHz）设计的评估开发板，支持 DDR3/DDR3L/DDR4内存，T153可提供强大的计算性能和快速响应，完美胜任高要求的自动化任务。还配备一路千兆以太网和两路百兆以太网接口、两个CAN_FD接口和localbus，支持高吞吐量网络连接，满足复杂数据驱动型应用需求。集成的图像信号处理器和显示引擎可为精密制造流程管理提供清晰的实时视觉反馈。适用于可编程逻辑控制器(PLC)、人机界面(HMI)机器人等场景。



IDO-EVB5301-V1正面



IDO-EVB5301-V1背面



IDO-EVB5301-V1 拓展板

2 Linux开发

2.1.获取 SDK

下载路径：[百度网盘/EVB5301/EVB5301-Linux/4.软件资料/SDK-V1.0（或SDK-V2.0）/系统SDK](#)

2.2.校验md5值

```
1 $ cat ./ido_evb5301_v0.95_a* > ido_evb5301_v0.95.tar.gz
2 $ md5sum ido_evb5301_v0.95.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~/evb5301为例

```
1 $ mkdir ~/evb5301
2 $ cd ~/evb5301
3 $ tar -xvf ido_evb5301_v0.95.tar.gz -C ./
```

2.4.SDK目录介绍

```

1  /
2  |— brandy //uboot和boot0代码
3  |   |— brandy-1.0
4  |   |— brandy-2.0
5  |— bsp
6  |   |— config //soc芯片平台公共配置
7  |   |— drivers //驱动接口相关代码
8  |   |— modules //nand和gpu驱动目录
9  |   |— ramfs //mini-sys文件系统
10 |— build //编译打包脚本
11 |   |— bin //制作文件系统工具
12 |   |— createkeys //创建安全方案密钥工具
13 |   |— envsetup.sh //配置环境变量
14 |   |— Makefile
15 |   |— mkcmd.sh //主要编译脚本
16 |   |— mkcommon.sh //编译脚本
17 |   |— mksetup.sh
18 |   |— pack //打包脚本
19 |   |— toolchain //编译工具链
20 |   |— top_build.sh
21 |— build.sh -> build/top_build.sh //顶级编译脚本，链接到build/top_build.sh
22 |— device //方案配置文档
23 |   |— config
24 |   |   |— chips //板级配置
25 |   |   |   |— a40i //A40I配置
26 |   |   |   |   |— bin //uboot和烧写程序
27 |   |   |   |   |— boot-resource //启动资源文件，如bootlogo
28 |   |   |   |   |— config //方案板配置
29 |   |   |   |   |   |— default //默认配置和通用配置
30 |   |   |   |   |   |— p3 //a40i p3方案板配置
31 |   |   |   |   |   |   |— buildroot //linux sdk配置
32 |   |   |   |   |   |   |— swupdate //linux OTA升级配置文件
33 |   |— product -> ./config/chips/a40i
34 |   |— target //方案配置目录
35 |— kernel //各个版本的内核文件
36 |   |— linux-4.9
37 |   |— linux-5.4
38 |   |— linux-5.10
39 |— out
40 |   |— gcc-linaro-* //kernel交叉编译工具链
41 |   |— pack_out //打包输出目录
42 |   |— a40i //a40i输出目录
43 |   |   |— p3 //a40i p3输出目录
44 |   |   |   |— bsp //bsp输出目录
45 |   |— a40i_linux_p3_uart0.img //a40i生成固件

```

```
46 |— test
47 |   |— dragonboard //dragonboard测试系统
48 |   |— SATA //sata测试系统
49 |— tools //pc工具
50 |   |— build
51 |   |— codecheck //代码检查工具
52 |   |— pack
53 |   |— tools_win //windows软件工具
```

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 root 权限（除非需要 apt 安装软件）

2.5.配置文件介绍

当 `./build.sh config` 完成之后，编译系统会生成一个方案的编译规则，其保存在SDK根目录中的 `.buildconfig` 文件里面。下面是一个规则的说明。

```
1 LICHEE_PLATFORM //编译平台
2 LICHEE_LINUX_DEV //编译的方案
3 LICHEE_IC // 编译的IC
4 LICHEE_BOARD //编译的板级配置
5 LICHEE_FLASH //编译的flash配置
6 LICHEE_CHIP //编译的芯片代号名称
7 LICHEE_KERN_VER //编译的内核版本
8 LICHEE_KERN_DEFCNF //编译内核的默认配置
9 LICHEE_BUILDING_SYSTEM //编译的构造系统
10 LICHEE_BR_VER //buildroot的版本
11 LICHEE_BR_DEFCNF //buildroot的默认配置
12 LICHEE_BR_RAMFS_CONF //buildroot的ramfs默认配置
13 LICHEE_BRANDY_VER //uboot的版本
14 LICHEE_BRANDY_DEFCNF //brandy的默认配置
15 LICHEE_CROSS_COMPILER //编译使用的交叉编译链
16 LICHEE_CHIP_CONFIG_DIR //编译系统的IC配置目录
17 LICHEE_OUT_DIR //编译的输出目录
```

上面只列出了部分重要的配置，详细的配置可以查看该文件。

而某个方案的大部分编译规则都是由 BoardConfig.mk，其存放在每个方案的板级配置目录中，同一个板型的不同系统拥有不用的 BoardConfig.mk 配置，例如 T153 bsp 验证系统和 Linux SDK 系统拥有各自的 BoardConfig.mk 配置。

2.6 分区说明

`<SDK_TOP_PATH>/device/config/chips/t153/configs/{board}/buildroot/sys_partition.fex` 文件中包含了固件的分区信息：


```

1  ;-----
   ;
2  ; 说明： 脚本中的字符串区分大小写，用户可以修改"="后面的数值，但是不要修改前面的字符串
3  ;-----
   ;
4
5
6  ;-----
   ;
7  ;                                     固件下载参数配置
8  ;-----
   ;
9  ;*****
   ;*****
10 ;    mbr的大小，以Kbyte为单位
11 ;*****
   ;*****
12 [mbr]
13 size = 16384
14
15 ;*****
   ;*****
16 ;                                     分区配置
17 ;
18 ;
19 ;  partition 定义范例：
20 [partition]                ; //表示是一个分区
21 ;  name          = USERFS2    ; //分区名称
22 ;  size          = 16384      ; //分区大小 单位： 扇区.分区表示个数最多2^31 *
   512 = 2T
23 ;  downloadfile = "123.fex"    ; //下载文件的路径和名称，可以使用相对路径，相对
   是指相对于image.cfg文件所在分区。也可以使用绝对路径
24 ;  keydata      = 1            ; //私有数据分区，重新量产数据将不丢失
25 ;  encrypt      = 1            ; //采用加密方式烧录，将提供数据加密，但损失烧录
   速度
26 ;  user_type    = ?            ; //私有用法
27 ;  verify       = 1            ; //要求量产完成后校验是否正确
28 ;
29 ; 注： 1、name唯一，不允许同名
30 ;      2、name最大12个字符
31 ;      3、size = 0， 将创建一个无大小的空分区
32 ;      4、为了安全和效率考虑，分区大小最好保证为16M字节的整数倍
33 ;*****
   ;*****
34 [partition_start]

```



```

35
36 ▾ [partition]
37     name          = boot-resource
38     size           = 65536
39     downloadfile   = "boot-resource.fex"
40     user_type      = 0x8000
41
42
43 ▾ [partition]
44     name          = env
45     size           = 1024
46     downloadfile   = "env.fex"
47     user_type      = 0x8000
48
49 ▾ [partition]
50     name          = env-redund
51     size           = 1024
52     downloadfile   = "env.fex"
53     user_type      = 0x8000
54
55 ▾ [partition]
56     name          = boot
57     size           = 65536
58     downloadfile   = "boot.fex"
59     user_type      = 0x8000
60
61 ▾ [partition]
62     name          = dtbo
63     size           = 2048
64     downloadfile   = "dtbo.fex"
65     user_type      = 0x8000
66
67 ▾ [partition]
68     name          = dtbo-r
69     size           = 2048
70     downloadfile   = "dtbo.fex"
71     user_type      = 0x8000
72
73 ▾ [partition]
74     name          = rootfs
75     size           = 1048576
76     downloadfile   = "rootfs.fex"
77     user_type      = 0x8000
78
79 ▾ [partition]
80     name          = private
81     size           = 248
82     ro             = 0

```

```
83      user_type      = 0x8000
84
85
```

2.7 ddr配置说明

开发板默认适配了以下两款ddr芯片

A3T4GF40BBF-HPI, DDR3, 容量512M, x16, 1866Mbps, 96BGA, 工作温度-40℃~+95℃, Zentel(力积电)

IS43TR16128DL-125KBLI, DDR3L, 容量256M, 128Mb x 16, 1600Mbps, 96BGA, 工作温度-40℃~+95℃, ISSI(北京君正)

两款ddr芯片的初始化配置并不是统一的, ddr初始化配置位于以下两个配置文件中 `device/config/chips/t153/configs/bga_demo/sys_config.fex`, `device/config/chips/t153/configs/bga_demo_nand/sys_config.fex`

编译固件默认会识别 `[dram_para]` 下的ddr配置。

```
1 ▾ [dram_para]
2   dram_clk           = 900
3   dram_type          = 3
4   dram_dx_odt         = 0x00000404
5   dram_dx_dri         = 0x00000808
6   dram_ca_dri         = 0x0c0c0c
7   dram_para0          = 0x00009898
8   dram_para1          = 0x30fa
9   dram_para2          = 0x0001
10  dram_mr0            = 0x840
11  dram_mr1            = 0x42
12  dram_mr2            = 0x8
13  dram_mr3            = 0x0
14  dram_mr4            = 0x0
15  dram_mr5            = 0x0
16  dram_mr6            = 0x0
17  dram_mr11           = 0x0
18  dram_mr12           = 0x0
19  dram_mr13           = 0x0
20  dram_mr14           = 0x0
21  dram_mr16           = 0x0
22  dram_mr17           = 0x0
23  dram_mr22           = 0x0
24  dram_tpr0           = 0x0
25  dram_tpr1           = 0x0
26  dram_tpr2           = 0x0
27  dram_tpr3           = 0x0
28  dram_tpr6           = 0x40
29  dram_tpr10          = 0x00001e00
30  dram_tpr11          = 0x00009494
31  dram_tpr12          = 0x13131313
32  dram_tpr13          = 0x08007071
33  dram_tpr14          = 0x810700f5
```

```
1 ▾ [dram_para]
2   dram_clk           = 800
3   dram_type          = 3
4   dram_dx_odt         = 0x00000404
5   dram_dx_dri         = 0x00000808
6   dram_ca_dri         = 0x0c0c0c
7   dram_para0         = 0x00009898
8   dram_para1         = 0x30ea
9   dram_para2         = 0x0001
10  dram_mr0           = 0x840
11  dram_mr1           = 0x42
12  dram_mr2           = 0x8
13  dram_mr3           = 0x0
14  dram_mr4           = 0x0
15  dram_mr5           = 0x0
16  dram_mr6           = 0x0
17  dram_mr11          = 0x0
18  dram_mr12          = 0x0
19  dram_mr13          = 0x0
20  dram_mr14          = 0x0
21  dram_mr16          = 0x0
22  dram_mr17          = 0x0
23  dram_mr22          = 0x0
24  dram_tpr0          = 0x0
25  dram_tpr1          = 0x0
26  dram_tpr2          = 0x0
27  dram_tpr3          = 0x0
28  dram_tpr6          = 0x40
29  dram_tpr10         = 0x00001e00
30  dram_tpr11         = 0x00009293
31  dram_tpr12         = 0x0000191B
32  dram_tpr13         = 0x08007071
33  dram_tpr14         = 0x810700f5
```

3.编译

3.1.选择项目

```

1  ./build.sh config
2  10-24 16:25:17.805 158713 D mkcommon : =====ACTION List: mk_config ;=
=====
3  10-24 16:25:17.806 158713 D mkcommon : options :
4  All available platform:
5      0. android
6      1. linux
7  ▾ Choice [linux]: 1 //选择编译linux固件, 按实际选择
8  All available linux_dev:
9      0. bsp
10     1. buildroot
11     2. ubuntu
12 ▾ Choice [buildroot]: 1 //选择bsp方案, 按实际选择
13 All available ic:
14     0. t153
15 ▾ Choice [t153]: 0
16 All available board:
17     0. bga_demo
18     1. bga_demo_amp_nand
19     2. bga_demo_nand
20     3. bga_demo_nor
21     4. demo
22     5. demo2_nand
23     6. demo_amp_nand
24     7. demo_nand
25     8. demo_nor
26     9. demo_qa
27 ▾ Choice [bga_demo]: 0 //选择bsp方案, 目前支持0和2;0: emmc版本;2: nand_flash版
本
28 All available flash:
29     0. default
30     1. nor
31 ▾ Choice [default]: 0 //选择存储介质, 按实际选择
32 All available kern_name:
33     0. linux-5.10-origin
34     1. linux-5.10-rt
35     2. linux-5.10-xenomai
36 ▾ Choice [linux-5.10-origin]: 0 //选择linux内核版本, 按实际选择

```

选择对应的数字即可。

3.2.自动编译

▼

Bash |

```
1  $ ./build.sh
```

3.3.分步编译

3.3.1.编译uboot

▼

Bash |

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

3.3.2.编译kernel

▼

Bash |

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

3.3.3.编译buildroot系统

▼

Bash |

```
1  ./build.sh buildroot_rootfs
```

3.3.4.固件打包

▼

Bash |

```
1  ./build.sh pack
```

编译生成的固件在SDK中 /out 目录下 t53_linux_bga_demo_uart0.img

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-EVB5301-V1开发板固件烧录手册》

4.1.整包烧录

4.1.1.运行全志烧录工具



4.1.2.点击一键刷机



4.1.3.选择固件



选择: `out/t153_linux_{board}_uart0.img`

4.1.4.点击升级



勾选全盘擦除升级后，点击立即升级。



工具提示是否进行对设备镜像烧写工作，点击是，自动开始烧录完整固件。

4.2.分包烧录

4.2.1.选择整包固件





选择单或多分区下载，勾选需要单独烧录的分区，并点击立刻升级即可。

注：这里的分区由整包固件的分区表决定，烧录前需保证分区表一致。

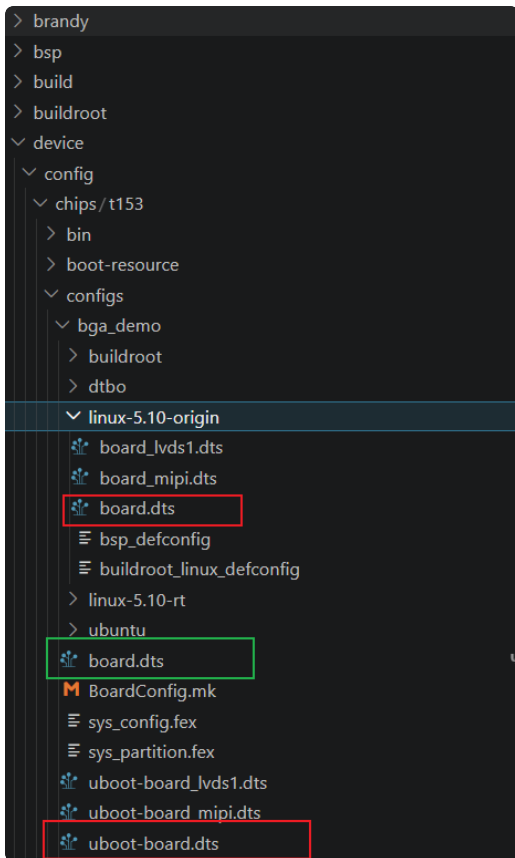
4.2.2.烧录



以单独烧录boot分区为例，单独勾选 `boot` 选项后，手动选择立即升级即可。

5.驱动开发

T153 DTS节点说明



根据编译时选择的 bsp 方案和内核的不同，在 参考 board 为 bga_demo ， kern_name 为 linux-5.10-origin 时

<SDK_TOP_PATH>/device/config/chips/t153/configs/chips/t153/{board}/board.dts 为连接到选择的 kern_name 的内核设备树

<SDK_TOP_PATH>/device/config/chips/t153/configs/chips/t153/{board}/u-boot-board.dts 为 u-boot 的设备树

对于 mipi 屏幕和 lvds 屏幕的内核设备树修改后，还需对应修改 u-boot 的设备树。

对应目录下的 *board_lvds.dts 与 *board_mipi.dts 为修改后的屏幕参考设备树。

5.1.CAN

5.1.1.CAN 简介

CAN(Controllor 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


```
<SDK_TOP_PATH>/device/config/chips/t153/configs/{board}/board.dts
```

```

1      can0 {
2          compatible = "allwinner,sun8i-t153-can";
3          reg = <0x0 0x0453C800 0x0 0x400>,
4                <0x0 0x04538000 0x0 0x4400>,
5                <0x0 0x04530000 0x0 0x1000>,
6                <0x0 0x04534000 0x0 0x1000>;
7          reg-names = "can", "message_ram", "can_top", "can_dma";
8          interrupts = <GIC_SPI 36 IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 35
IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 37 IRQ_TYPE_LEVEL_HIGH>;
9          interrupt-names = "int0", "int_top", "int1";
10         clocks = <&ccu CLK_CAN0>, <&ccu CLK_BUS_CAN0>, <&ccu CLK_M
BUS_CAN_GATE>;
11         clock-names = "can_clk", "can_bus", "can_mbus";
12         resets = <&ccu RST_BUS_CAN0>, <&ccu RST_BUS_CAN_SYS>;
13         reset-names = "can_rst", "can_sys_rst";
14         clock-frequency = <40000000>;
15         allwinner,ram-cfg = <0x0 0 0 64 64 64 32 32>;
16         pinctrl-0 = <&can0_pins_active>;
17         pinctrl-1 = <&can0_pins_sleep>;
18         pinctrl-names = "default", "sleep";
19         status = "disabled";
20     };
21
22     can1 {
23         compatible = "allwinner,sun8i-t153-can";
24         reg = <0x0 0x04541800 0x0 0x400>,
25               <0x0 0x0453D000 0x0 0x4400>,
26               <0x0 0x04531000 0x0 0x1000>,
27               <0x0 0x04535000 0x0 0x1000>;
28         reg-names = "can", "message_ram", "can_top", "can_dma";
29         interrupts = <GIC_SPI 144 IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 1
46 IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 145 IRQ_TYPE_LEVEL_HIGH>;
30         interrupt-names = "int0", "int_top", "int1";
31         clocks = <&ccu CLK_CAN1>, <&ccu CLK_BUS_CAN1>, <&ccu CLK_M
BUS_CAN_GATE>;
32         clock-names = "can_clk", "can_bus", "can_mbus";
33         resets = <&ccu RST_BUS_CAN1>, <&ccu RST_BUS_CAN_SYS>;
34         reset-names = "can_rst", "can_sys_rst";
35         clock-frequency = <40000000>;
36         allwinner,ram-cfg = <0x0 0 0 64 64 64 32 32>;
37         pinctrl-0 = <&can1_pins_active>;
38         pinctrl-1 = <&can1_pins_sleep>;
39         pinctrl-names = "default", "sleep";
40         status = "disabled";
41     };

```

```
42
43 ▾ &pio {
44
45
46     };
```

配置CAN1

```
▾ Bash |
1 ▾ &can0 {
2     status = "disabled";
3 };
4
5 ▾ &can1 {
6     status = "okay";
7 };
```

双CAN配置

板级配置: ido_evb3506_v1a-emmc.dtsi

```
▾ Bash |
1 ▾ &can0 {
2     status = "okay";
3 };
4
5 ▾ &can1 {
6     status = "okay";
7 };
```

通信测试

```
1  #检查can设备
2  $ ip link show
3
4  #在收发端关闭can0设备
5  $ ip link set can0 down
6
7  #设置仲裁段1M波特率，数据段1M波特率
8  $ ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
9  [ 63.834376] rk3576_canfd ff330000.can can0: bitrate error 0.3%
10 [ 63.834512] rk3576_canfd ff330000.can can0: bitrate error 0.3%
11
12 #在收发端打开can0设备
13 $ ip link set can0 up
14
15 #在接收端执行candump, 阻塞等待报文
16 $ candump can0
17
18 #在发送端执行cansend, 发送报文
19 $ cansend can0 123#1122334455667788
20
21 #检查是否启用成功
22 $ ip link show can0
```

总结调试过程中遇到的几个问题及解决方法：

1.无法正常收发

检查总线 CAN_H 和 CAN_L， 杜邦线是否松动或者接反。

2.CAN时钟频率配置

如果CAN的比特率低于等于3M建议修改CAN时钟到100M,信号更稳定。高于3M比特率的, 时钟设置200M就可以。

CAN时钟频率修改方法参考如下：

```
1 ▾ &can0 {  
2     .....  
3     -         clock-frequency = <300000000>;  
4     +         clock-frequency = <200000000>;  
5     .....  
6     };  
7  
8 ▾ &can1 {  
9     .....  
10    -         clock-frequency = <300000000>;  
11    +         clock-frequency = <200000000>;  
12    .....  
13    };
```

在某些时钟频率下，CAN的bitrate无法获得准确的速率，可以自行调整clock-frequency去解决。

查看是否得到所需的bitrate：

```
1 ip -d link show can0
```

5.2.Ethernet

公共配置

```
bsp/configs/{kernel}/sun8iw22p1.dtsi
```

```

1  gmac0: ethernet@4500000 {
2      compatible = "allwinner,sunxi-gmac-211";
3      reg = <0x0 0x04500000 0x0 0x8000>, <0x0 0x04508000 0x0 0x1000>;
4      interrupts = <GIC_SPI 46 IRQ_TYPE_LEVEL_HIGH>;
5      interrupt-names = "macirq";
6      clocks = <&ccu CLK_BUS_GMAC0_AHB>, <&ccu CLK_MBUS_BUS_GMAC0>, <&ccu
u CLK_GMAC0_PHY>, <&ccu CLK_GMAC0_PTP_REF>,
7      <&ccu CLK_BUS_GMAC0_AHB_SW_CFG>, <&ccu CLK_MBUS_BUS_GMAC0_SW_CFG>;
8      clock-names = "stmmaceth", "pclk", "phy", "ptp_ref",
9      "ahb_sw_cfg", "mhub_sw_cfg";
10     assigned-clocks = <&ccu CLK_GMAC0_PHY>;
11     assigned-clock-rates = <25000000>;
12     resets = <&ccu RST_BUS_GMAC0_AXI>, <&ccu RST_BUS_GMAC0_AHB>;
13     reset-names = "stmmaceth", "ahb";
14     phy-mode = "rgmii";
15     phy-handle = <&gmac0_phy0>;
16     status = "disabled";
17
18     aw,rgmii-clk-ext;
19     snps,fixed-burst;
20     snps,en-tx-lpi-clockgating;
21     snps,axi-config = <&gmac_stmmac_axi_setup>;
22     snps,mtl-rx-config = <&gmac_mtl_rx_setup>;
23     snps,mtl-tx-config = <&gmac_mtl_tx_setup>;
24
25     mdio0: mdio0@0 {
26         compatible = "snps,dwmac-mdio";
27         #address-cells = <1>;
28         #size-cells = <0>;
29         gmac0_phy0: ethernet-phy@1 {
30             compatible = "ethernet-phy-ieee802.3-c22";
31             reg = <0x1>;
32             max-speed = <1000>; /* Max speed capability */
33             reset-gpios = <&pio PA 14 GPIO_ACTIVE_LOW>;
34             /* PHY datasheet rst time */
35             reset-assert-us = <10000>;
36             reset-deassert-us = <150000>;
37         };
38     };
39 };
40
41 gmac1: ethernet@4510000 {
42     compatible = "allwinner,sunxi-gmac-211";
43     reg = <0x0 0x04510000 0x0 0x8000>, <0x0 0x04518000 0x0 0x1000>;

```

```

44     interrupts = <GIC_SPI 47 IRQ_TYPE_LEVEL_HIGH>;
45     interrupt-names = "macirq";
46     clocks = <&ccu CLK_BUS_GMAC1_AHB>, <&ccu CLK_MBUS_BUS_GMAC1>, <&ccu
47 CLK_GMAC1_PHY>, <&ccu CLK_GMAC1_PTP_REF>,
        <&ccu CLK_BUS_GMAC1_AHB_SW_CFG>, <&ccu CLK_MBUS_BUS_GMAC1_SW_CFG>;
48     clock-names = "stmmaceth", "pclk", "phy", "ptp_ref",
49                 "ahb_sw_cfg", "mhub_sw_cfg";
50     assigned-clocks = <&ccu CLK_GMAC1_PHY>;
51     assigned-clock-rates = <25000000>;
52     resets = <&ccu RST_BUS_GMAC1_AXI>, <&ccu RST_BUS_GMAC1_AHB>;
53     reset-names = "stmmaceth", "ahb";
54     phy-mode = "rgmii";
55     phy-handle = <&gmac1_phy0>;
56     status = "disabled";
57
58     aw,rgmii-clk-ext;
59     snps,fixed-burst;
60     snps,en-tx-lpi-clockgating;
61     snps,axi-config = <&gmac_stmmac_axi_setup>;
62     snps,mtl-rx-config = <&gmac_mtl_rx_setup>;
63     snps,mtl-tx-config = <&gmac_mtl_tx_setup>;
64
65     mdio1: mdio1@0 {
66         compatible = "snps,dwmac-mdio";
67         #address-cells = <1>;
68         #size-cells = <0>;
69         gmac1_phy0: ethernet-phy@1 {
70             compatible = "ethernet-phy-ieee802.3-c22";
71             reg = <0x1>;
72             max-speed = <1000>; /* Max speed capability */
73             /* PHY datasheet rst time */
74             reset-assert-us = <10000>;
75             reset-deassert-us = <150000>;
76         };
77     };
78 };
79
80     gmac2: ethernet@4520000 {
81         compatible = "allwinner,sunxi-gmac-211";
82         reg = <0x0 0x04520000 0x0 0x8000>, <0x0 0x04528000 0x0 0x1000>;
83         interrupts = <GIC_SPI 101 IRQ_TYPE_LEVEL_HIGH>;
84         interrupt-names = "macirq";
85         clocks = <&ccu CLK_BUS_GMAC2_AHB>, <&ccu CLK_MBUS_BUS_GMAC2>, <&ccu
86 CLK_GMAC2_PHY>, <&ccu CLK_GMAC2_PTP_REF>,
            <&ccu CLK_BUS_GMAC2_AHB_SW_CFG>, <&ccu CLK_MBUS_BUS_GMAC2_SW_CFG>;
87     };
        clock-names = "stmmaceth", "pclk", "phy", "ptp_ref",

```

```

88         "ahb_sw_cfg", "mhub_sw_cfg";
89
90     assigned-clocks = <&ccu CLK_GMAC2_PHY>;
91     assigned-clock-rates = <25000000>;
92     resets = <&ccu RST_BUS_GMAC2_AXI>, <&ccu RST_BUS_GMAC2_AHB>;
93     reset-names = "stmmaceth", "ahb";
94     phy-mode = "rgmii";
95     phy-handle = <&gmac2_phy0>;
96     status = "disabled";
97
98     aw,rgmii-clk-ext;
99     snps,fixed-burst;
100    snps,en-tx-lpi-clockgating;
101    snps,axi-config = <&gmac_stmmac_axi_setup>;
102    snps,mtl-rx-config = <&gmac_mtl_rx_setup>;
103    snps,mtl-tx-config = <&gmac_mtl_tx_setup>;
104
105    mdio2: mdio2@0 {
106        compatible = "snps,dwmac-mdio";
107        #address-cells = <1>;
108        #size-cells = <0>;
109        gmac2_phy0: ethernet-phy@1 {
110            compatible = "ethernet-phy-ieee802.3-c22";
111            reg = <0x1>;
112            max-speed = <1000>; /* Max speed capability */
113            /* PHY datasheet rst time */
114            reset-assert-us = <10000>;
115            reset-deassert-us = <150000>;
116        };
117    };

```

板级的配置


```

1 ▾ &gmac0_phy0 {
2     compatible = "ethernet-phy-id001c.c916";
3     reset-gpios = <&pio PJ 10 GPIO_ACTIVE_LOW>;
4     /* For RTL8211F: PHY datasheet rst time */
5     reset-assert-us = <10000>;
6     reset-deassert-us = <30000>;
7     reg = <0x1>;
8     max-speed = <100>;
9 };
10
11 ▾ &gmac0 {
12     phy-mode = "rmii";
13     pinctrl-names = "default", "sleep";
14     pinctrl-0 = <&gmac0_pins_default>;
15     pinctrl-1 = <&gmac0_pins_sleep>;
16     phy-handle = <&gmac0_phy0>;
17     status = "okay";
18 };
19
20 ▾ &gmac1_phy0 {
21     compatible = "ethernet-phy-id4f51.e91b";
22     reg = <0x1>;
23     reset-gpios = <&pio PE 8 GPIO_ACTIVE_LOW>;
24     /* For RTL8211F: PHY datasheet rst time */
25     reset-assert-us = <10000>;
26     reset-deassert-us = <150000>;
27 };
28
29 ▾ &gmac1 {
30     phy-mode = "rgmii";
31     pinctrl-names = "default", "sleep";
32     pinctrl-0 = <&gmac1_pins_default>;
33     pinctrl-1 = <&gmac1_pins_sleep>;
34     phy-handle = <&gmac1_phy0>;
35     aw,soc-phy-clk-en;
36     sunxi,phy-clk-type = <0>;
37     /delete-property/ aw,rgmii-clk-ext;
38     tx-delay = <4>;
39     rx-delay = <13>;
40     status = "okay";
41 };
42
43 ▾ &gmac2_phy0 {
44     compatible = "ethernet-phy-id0000.0128";
45     reset-gpios = <&pio PA 7 GPIO_ACTIVE_LOW>;

```

```

46      /* For RTL8211F: PHY datasheet rst time */
47      reset-assert-us = <10000>;
48      reset-deassert-us = <30000>;
49      reg = <0x1>;
50      max-speed = <100>;
51  };
52
53  &gmac2 {
54      phy-mode = "rmii";
55      pinctrl-names = "default", "sleep";
56      pinctrl-0 = <&gmac2_pins_default>;
57      pinctrl-1 = <&gmac2_pins_sleep>;
58      phy-handle = <&gmac2_phy0>;
59      status = "okay";
60  };
61
62  &pio {
63  &gmac0_pins_default: gmac0@0 {
64      pins = "PJ0", "PJ1", "PJ2", "PJ3", "PJ4", "PJ5", "PJ6",
65      "PJ7",
66      "PJ8", "PJ9";
67      function = "rgmii0";
68      drive-strength = <40>;
69      bias-pull-up;
70  };
71
72  &gmac0_pins_sleep: gmac0@1 {
73      pins = "PJ0", "PJ1", "PJ2", "PJ3", "PJ4", "PJ5", "PJ6",
74      "PJ7",
75      "PJ8", "PJ9";
76      function = "io_disabled";
77      bias-disable;
78  };
79
80  &gmac1_pins_default: gmac1@0 {
81      pins = "PG0", "PG1", "PG2", "PG3", "PG4", "PG6", "PG7",
82      "PG8", "PG9", "PG10", "PG11", "PG12", "PG13", "PG
83      14", "PG15";
84      function = "rgmii1";
85      drive-strength = <40>;
86      bias-pull-up;
87  };
88
89  &gmac1_pins_sleep: gmac1@1 {
90      pins = "PG0", "PG1", "PG2", "PG3", "PG4", "PG6", "PG7",
91      "PG8", "PG9", "PG10", "PG11", "PG12", "PG13", "PG
92      14", "PG15";
93      function = "io_disabled";

```

```

90             bias-disable;
91         };
92
93     gmac2_pins_default: gmac2@0 {
94         pins = "PD0", "PD1", "PD2", "PD3", "PD4", "PD5", "PD6",
95             "PD7", "PD8", "PD9";
96         function = "rgmii2";
97         drive-strength = <40>;
98         bias-pull-up;
99     };
100
101     gmac2_pins_sleep: gmac2@1 {
102         pins = "PD0", "PD1", "PD2", "PD3", "PD4", "PD5", "PD6",
103             "PD7", "PD8", "PD9";
104         function = "io_disabled";
105         bias-disable;
106     };
107
108 };

```

注：IDO-EVB5301开发板gmac2与lvds0、mipi硬件复用，如需配置lvds0、mipi，则应先禁用gmac2相关节点。

5.3.LCD

IDO-EVB5301-V1 有一路 MIPI DSI 显示输出接口和两路LVDS输出接口。

dts修改:

mipi-1024x600:

```

1  backlight0: backlight0 {
2      compatible = "pwm-backlight";
3      status = "okay";
4      brightness-levels = <
5          0   1   2   3   4   5   6   7
6          8   9  10  11  12  13  14  15
7          16  17  18  19  20  21  22  23
8          24  25  26  27  28  29  30  31
9          32  33  34  35  36  37  38  39
10         40  41  42  43  44  45  46  47
11         48  49  50  51  52  53  54  55
12         56  57  58  59  60  61  62  63
13         64  65  66  67  68  69  70  71
14         72  73  74  75  76  77  78  79
15         80  81  82  83  84  85  86  87
16         88  89  90  91  92  93  94  95
17         96  97  98  99 100 101 102 103
18        104 105 106 107 108 109 110 111
19        112 113 114 115 116 117 118 119
20        120 121 122 123 124 125 126 127
21        128 129 130 131 132 133 134 135
22        136 137 138 139 140 141 142 143
23        144 145 146 147 148 149 150 151
24        152 153 154 155 156 157 158 159
25        160 161 162 163 164 165 166 167
26        168 169 170 171 172 173 174 175
27        176 177 178 179 180 181 182 183
28        184 185 186 187 188 189 190 191
29        192 193 194 195 196 197 198 199
30        200 201 202 203 204 205 206 207
31        208 209 210 211 212 213 214 215
32        216 217 218 219 220 221 222 223
33        224 225 226 227 228 229 230 231
34        232 233 234 235 236 237 238 239
35        240 241 242 243 244 245 246 247
36        248 249 250 251 252 253 254 255>;
37      default-brightness-level = <150>;
38  //      enable-gpios = <&pio PA 9 GPIO_ACTIVE_HIGH>;
39      pwms = <&pwmcs0 0 25000 0>;
40  };
41
42  panel_0: panel_0@0 {
43      compatible = "allwinner,panel-dsi";
44      status = "okay";
45      enable0-gpios = <&pio PB 7 GPIO_ACTIVE_HIGH>;

```

```

46 enable-delay-ms = <10>;
47 reset-gpios = <&pio PA 6 GPIO_ACTIVE_HIGH>;
48 reset-on-sequence = <1 10>, <0 20>, <1 100>;
49 reset-off-sequence = <0 100>;
50 backlight = <&backlight0>;
51 dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_ASYNC_INCELL)>;
52 dsi,lanes = <4>;
53 dsi,format = <0>;
54 panel-init-sequence = [
55     15 00 02 80 5b
56     15 00 02 81 78
57     15 00 02 82 84
58     15 00 02 83 88
59     15 00 02 84 88
60     15 00 02 85 E3
61     15 00 02 86 88
62     05 78 01 11
63     05 14 01 29
64 ];
65 panel-exit-sequence = [
66     05 00 01 28
67     05 78 01 10
68 ];
69
70 display-timings {
71     native-mode = <&timing0>;
72     timing0: timing0 {
73         clock-frequency = <51000000>;
74         hactive = <1024>;
75         vactive = <600>;
76         hfront-porch = <160>;
77         hsync-len = <10>;
78         hback-porch = <160>;
79         vfront-porch = <23>;
80         vsync-len = <2>;
81         vback-porch = <12>;
82         hsync-active = <0>;
83         vsync-active = <0>;
84         de-active = <0>;
85         pixelclk-active = <0>;
86
87     /*
88         clock-frequency = <156408000>;
89         hback-porch = <40>;
90         hactive = <1200>;
91         hfront-porch = <80>;
92         hsync-len = <10>;
93         vback-porch = <16>;
94         vactive = <1920>;

```

```

94         vfront-porch = <20>;
95         vsync-len = <4>;
96     */
97     };
98 };
99 port {
100     #address-cells = <1>;
101     #size-cells = <0>;
102     panel0_in: endpoint@0 {
103         reg = <0>;
104         remote-endpoint = <&panel_output_0>;
105     };
106 };
107 };
108
109 &pio {
110     dsi0_4lane_pins_a: dsi0_4lane@0 {
111         pins = "PD0", "PD1", "PD2", "PD3", "PD4", "PD5", "PD6", "PD7", "PD8", "PD9"; // "PD17", "PD18", "PD19", "PD21";
112         function = "dsi";
113         drive-strength = <30>;
114         bias-disable;
115     };
116     dsi0_4lane_pins_b: dsi0_4lane@1 {
117         pins = "PD0", "PD1", "PD2", "PD3", "PD4", "PD5", "PD6", "PD7", "PD8", "PD9"; // "PD17", "PD18", "PD19", "PD21";
118         function = "io_disabled";
119         bias-disable;
120     };
121 };
122
123 &dsi0 {
124     status = "okay";
125     pinctrl-0 = <&dsi0_4lane_pins_a>;
126     pinctrl-1 = <&dsi0_4lane_pins_b>;
127     #address-cells = <1>;
128     // interrupts = <GIC_SPI 83 IRQ_TYPE_LEVEL_HIGH>;
129     #size-cells = <0>;
130     ports {
131         dsi0_out: port@1 {
132             dsi_out_panel: endpoint {
133                 remote-endpoint = <&panel_input>;
134             };
135         };
136     };
137     panel: panel@0 {
138         compatible = "allwinner,virtual-panel";
139         status = "okay";

```

```

140     reg = <0>;
141 ports {
142     #address-cells = <1>;
143     #size-cells = <0>;
144     panel_in: port@0 {
145         #address-cells = <1>;
146         #size-cells = <0>;
147         reg = <0>;
148         panel_input: endpoint@0 {
149             reg = <0>;
150             remote-endpoint = <&dsi_out_panel>;
151         };
152         panel_input1: endpoint@1 {
153             };
154     };
155     panel_out: port@1 {
156         reg = <1>;
157         #address-cells = <1>;
158         #size-cells = <0>;
159         panel_output_0: endpoint@0 {
160             reg = <0>;
161             remote-endpoint = <&panel0_in>;
162         };
163         panel_output_1: endpoint@1 {
164             };
165     };
166 };
167 };
168 };
169 };
170 &vo0 {
171     status = "okay";
172 };
173 };
174 &dlcd0 {
175     status = "okay";
176 };
177 };
178 &de {
179     chn_cfg_mode = <0>;
180     status = "okay";
181 };
182 };
183 &dsi0combophy {
184     status = "okay";
185 };
186 };
187 &sunxi_drm{

```

```
188     route{
189         route_dsi0{
190             status = "okay";
191         };
192     };
193 };
```

lvds-1024x600:


```

1  backlight0: backlight0 {
2      compatible = "pwm-backlight";
3      status = "okay";
4      brightness-levels = <
5          0   1   2   3   4   5   6   7
6          8   9  10  11  12  13  14  15
7          16  17  18  19  20  21  22  23
8          24  25  26  27  28  29  30  31
9          32  33  34  35  36  37  38  39
10         40  41  42  43  44  45  46  47
11         48  49  50  51  52  53  54  55
12         56  57  58  59  60  61  62  63
13         64  65  66  67  68  69  70  71
14         72  73  74  75  76  77  78  79
15         80  81  82  83  84  85  86  87
16         88  89  90  91  92  93  94  95
17         96  97  98  99 100 101 102 103
18        104 105 106 107 108 109 110 111
19        112 113 114 115 116 117 118 119
20        120 121 122 123 124 125 126 127
21        128 129 130 131 132 133 134 135
22        136 137 138 139 140 141 142 143
23        144 145 146 147 148 149 150 151
24        152 153 154 155 156 157 158 159
25        160 161 162 163 164 165 166 167
26        168 169 170 171 172 173 174 175
27        176 177 178 179 180 181 182 183
28        184 185 186 187 188 189 190 191
29        192 193 194 195 196 197 198 199
30        200 201 202 203 204 205 206 207
31        208 209 210 211 212 213 214 215
32        216 217 218 219 220 221 222 223
33        224 225 226 227 228 229 230 231
34        232 233 234 235 236 237 238 239
35        240 241 242 243 244 245 246 247
36        248 249 250 251 252 253 254 255>;
37      default-brightness-level = <150>;
38  //      enable-gpios = <&pio PB 14 GPIO_ACTIVE_HIGH>;
39      pwms = <&pwmcs0 0 25000 0>;
40  };
41
42  lvds_panel: lvds_panel@0 {
43      compatible = "sunxi-lvds";
44      status = "okay";
45      backlight = <&backlight0>;

```

```

46 bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;
47 // bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_JEIDA>;
48 // bus-format = <MEDIA_BUS_FMT_RGB666_1X7X3_SPWG>;
49 reset_gpios = <&pio PA 4 GPIO_ACTIVE_HIGH>; //复位脚
50 display-timings {
51     native-mode = <&lvds0_timing0>;
52     lvds0_timing0: timing0 {
53         clock-frequency = <51287040>;
54         hactive = <1024>;
55         vactive = <600>;
56         hback-porch = <160>;
57         hfront-porch = <136>;
58         hsync-len = <24>;
59         vback-porch = <23>;
60         vfront-porch = <12>;
61         vsync-len = <1>;
62     };
63 };
64 port {
65     lvds_panel_in: endpoint {
66         remote-endpoint = <&lvds_panel_out>;
67     };
68 };
69 };
70 };
71 leds {
72     compatible = "gpio-leds";
73     pinctrl-names = "default", "sleep";
74     status = "okay";
75
76     .....
77
78     lvds_ctl_en_PA9 {
79         gpios = <&pio PA 9 GPIO_ACTIVE_HIGH>;
80         default-state = "on";
81     };
82
83     lvds_ctl_pwr {
84         gpios = <&pio PD 20 GPIO_ACTIVE_HIGH>;
85         default-state = "on";
86     };
87
88     .....
89 };
90
91 &pio {
92     lvds1_pins_a: lvds1@0 {
93

```

```

94     pins = "PD10", "PD11", "PD12", "PD13", "PD14", "PD15", "PD16", "PD17"
95 , "PD18", "PD19";
96     function = "lvds1";
97     drive-strength = <30>;
98 };
99
100     lvds1_pins_b: lvds1@1 {
101         pins = "PD10", "PD11", "PD12", "PD13", "PD14", "PD15", "PD16", "PD17"
102 , "PD18", "PD19";
103         function = "gpio_in";
104     };
105
106     leds_gpio1_pins_default: leds@0 {
107         pins = .....\
108         "PD20", "PA9";
109         function = "gpio_out";
110     };
111
112     leds_gpio1_pins_sleep: leds@1 {
113         pins = .....\
114         "PD20", "PA9";
115         function = "gpio_in";
116     };
117
118     &lvds0 {
119         status = "okay";
120         dual-channel = <2>;
121         pinctrl-0 = <&lvds1_pins_a>;
122         pinctrl-1 = <&lvds1_pins_b>;
123         pinctrl-names = "active","sleep";
124         ports {
125             #address-cells = <1>;
126             #size-cells = <0>;
127             port@1 {
128                 #address-cells = <1>;
129                 #size-cells = <0>;
130                 reg = <1>;
131                 lvds_panel_out: endpoint@0 {
132                     #address-cells = <1>;
133                     #size-cells = <0>;
134                     reg = <0>;
135                     remote-endpoint = <&lvds_panel_in>;
136                 };
137             };
138         };
139     };

```

```

140     &vo0 {
141         status = "okay";
142     };
143
144     &dlcd0 {
145         status = "okay";
146     };
147
148     &de {
149         chn_cfg_mode = <0>;
150         status = "okay";
151     };
152
153     &dsi0combophy {
154         status = "okay";
155     };
156
157     &sunxi_drm{
158         route{
159             route_lvds0{
160                 status = "okay";
161             };
162         };

```

5.4.RTC

IDO-EVB5301-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: `device/config/chips/t153/configs/bga_demo/linux-5.10-origin/boar`
`d.dts`

```
1
2 ▾ &twi2 {
3     clock-frequency = <400000>;
4     pinctrl-0 = <&twi2_pins_default>;
5     pinctrl-1 = <&twi2_pins_sleep>;
6     pinctrl-names = "default", "sleep";
7     /* For stability and backwards compatibility, we recommend settin
g 'twi_drv_used' to 1 */
8     twi_drv_used = <1>;
9     /* twi-supply = <&reg_cldo3>; */
10    status = "okay";
11
12 ▾    pcf8563: rtc@51 {
13        compatible = "nxp,pcf8563";
14        reg = <0x51>;
15    };
16 };
17
18 ▾ &pio {
19 ▾    twi2_pins_default: twi2@0 {
20        pins = "PK7", "PK8";
21        function = "twi2";
22        drive-strength = <10>;
23        bias-pull-up;
24    };
25
26 ▾    twi2_pins_sleep: twi2@1 {
27        pins = "PK7", "PK8";
28        function = "gpio_in";
29    };
30 };
```

驱动参考: `kernel/linux-5.10-origins/drivers/rtc/rtc-pcf8563.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/rtc0/date
6  2021-01-01
7
8  $ cat /sys/class/rtc/rtc0/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     pinctrl-names = "default", "sleep";
3
4     pinctrl-0 = <&uart3_pins_active>;
5     pinctrl-1 = <&uart3_pins_sleep>;
6     status = "okay";
7 }
8 ▼ &pio {
9 ▼     uart3_pins_active: uart3_pins@0 {
10         pins = "PK9", "PK10";
11         function = "uart3";
12     };
13
14 ▼     uart3_pins_sleep: uart3_pins@1 {
15         pins = "PK9", "PK10";
16         function = "io_disabled";
17     };
18 };
```

```
1 ▼ &uart2 {
2     pinctrl-names = "default", "sleep";
3     pinctrl-0 = <&uart2_pins_active>;
4     pinctrl-1 = <&uart2_pins_sleep>;
5     sunxi,uart-485pin_auto = <2>;
6     status = "okay";
7 };
8
9 ▼ &pio {
10 ▼     uart2_pins_active: uart2_pins@0 {
11         pins = "PB0", "PB1", "PB2";
12         function = "uart2";
13     };
14
15 ▼     uart2_pins_sleep: uart2_pins@1 {
16         pins = "PB0", "PB1", "PB2";
17         function = "io_disabled";
18     };
19 };
```

配置好串口后，硬件接口对应软件上的节点为：

▼

Bash

```
1  UART2:  /dev/ttyAS2
```

收发测试

▼ RS232

Bash

```
1  # cat /dev/ttyAS5 &
2  [1] 453
3  # echo 123123 > /dev/ttyAS7
4  123123
5
6  # echo 123123 > /dev/ttyAS7
7  123123
8
9  # kill 453
10 # cat /dev/ttyAS7 &
11 [2] 454
12 [1] Terminated          cat /dev/ttyAS5
13 # echo 123123 > /dev/ttyAS5
14 123123
15
16 # echo 123123 > /dev/ttyAS5
17 123123
18
19 # echo 123123 > /dev/ttyAS5
20 123123
21
22 #
```

测试RS485收发可以通过以下测试demo进行测试验证


```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<unistd.h>
4  #include<sys/types.h>
5  #include<sys/stat.h>
6  #include<fcntl.h>
7  #include<termios.h>
8  #include<errno.h>
9  #include<string.h>
10 #include<pthread.h>
11 #include<time.h>
12 #include <linux/serial.h>
13 #include <termios.h>
14 #include <sys/ioctl.h>
15
16 #define FALSE  -1
17 #define TRUE   0
18
19 volatile int flag = 0;
20
21 int uart_open(char* port)
22 {
23     int fd;
24     fd = open( port, O_RDWR|O_NOCTTY|O_NDELAY);
25     if (fd < 0)
26     {
27         printf("can't open serial port:%s\n",port);
28         return(FALSE);
29     }
30
31     if(fcntl(fd, F_SETFL, 0) < 0)
32     {
33         printf("fcntl failed!\n");
34         return(FALSE);
35     }
36
37     if(0 == isatty(STDIN_FILENO))
38     {
39         printf("standard input is not a terminal device\n");
40         //return(FALSE);
41     }
42
43     return fd;
44 }
45
```

```

46 void uart_close(int fd)
47 {
48     close(fd);
49 }
50
51 int uart_set(int fd,int speed,int flow_ctrl,int databits,int stopbits,int
52 parity)
53 {
54     int i;
55     int status;
56     struct serial_rs485 rs485conf;
57     int speed_arr[] = {B4000000, B1500000, B115200, B57600, B38400, B1920
0, B9600, B4800,
58                             B2400, B1800, B1200, B600, B300, B200, B150,
59                             B134, B110, B75, B50, B0};
60     int name_arr[] = {4000000, 1500000, 115200, 57600, 38400, 19200, 9600
, 4800, 2400, 1800,
61                             1200, 600, 300, 200, 150, 134, 110, 75, 50, 0};
62
63     struct termios options;
64
65     if ( tcgetattr( fd,&options) != 0)
66     {
67         perror("setupSerial 1");
68         return FALSE;
69     }
70
71     for ( i= 0; i < sizeof(speed_arr) / sizeof(int); i++)
72     {
73         if (speed == name_arr[i])
74         {
75             cfsetispeed(&options, speed_arr[i]);
76             cfsetospeed(&options, speed_arr[i]);
77         }
78     }
79
80     // RS485
81     if (ioctl(fd, TIOCGRS485, &rs485conf) < 0) {
82         return FALSE;
83     }
84     rs485conf.flags |= SER_RS485_ENABLED;
85     if (ioctl(fd, TIOCRS485, &rs485conf) < 0) {
86         return FALSE;
87     }
88
89     options.c_cflag |= CLOCAL; //修改控制模式, 保证程序不会占用串口
90     options.c_cflag |= CREAD; //修改控制模式, 使得能够从串口读取输入数据

```

```

91     switch(flow_ctrl)    //设置数据流控制
92     {
93
94         case 0 :
95             options.c_cflag &= ~CRTSCTS;    //不使用流控制
96             //options.c_iflag &= ~(IXON | IXOFF | IXANY);
97             break;
98
99         case 1 :
100            options.c_cflag |= CRTSCTS; //使用硬件流控制
101            break;
102        case 2 :
103            options.c_cflag |= IXON | IXOFF | IXANY; //使用软件流控制
104            break;
105    }
106
107
108    options.c_cflag &= ~CSIZE; //设置数据位    //屏蔽其他标志位
109
110    switch (databits)
111    {
112        case 5:
113            options.c_cflag |= CS5;
114            break;
115        case 6:
116            options.c_cflag |= CS6;
117            break;
118        case 7:
119            options.c_cflag |= CS7;
120            break;
121        case 8:
122            options.c_cflag |= CS8;
123            break;
124        default:
125            fprintf(stderr, "Unsupported data size\n");
126            return FALSE;
127    }
128
129    switch (parity)    //设置校验位
130    {
131        case 'n':
132        case 'N': //无奇偶校验位。
133            options.c_cflag &= ~PARENB;
134            options.c_iflag &= ~INPCK;
135            break;
136        case 'o':
137        case 'O': //设置为奇校验
138            options.c_cflag |= (PARODD | PARENB);

```

```

139         options.c_iflag |= INPCK;
140         break;
141     case 'e':
142     case 'E': // 设置为偶校验
143         options.c_cflag |= PARENB;
144         options.c_cflag &= ~PARODD;
145         options.c_iflag |= INPCK;
146         break;
147     case 's':
148     case 'S': // 设置为空格
149         options.c_cflag &= ~PARENB;
150         options.c_cflag &= ~CSTOPB;
151         break;
152     default:
153         fprintf(stderr, "Unsupported parity\n");
154         return FALSE;
155 }
156
157 switch (stopbits) // 设置停止位
158 {
159     case 1:
160         options.c_cflag &= ~CSTOPB; break;
161     case 2:
162         options.c_cflag |= CSTOPB; break;
163     default:
164         fprintf(stderr, "Unsupported stop bits\n");
165         return FALSE;
166 }
167
168 options.c_iflag &= ~(IXON | IXOFF | IXANY);
169 options.c_iflag &= ~(INLCR | ICRNL | IGNCR);
170 options.c_oflag &= ~(ONLCR | OCRNL);
171
172 options.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG);
173 options.c_oflag &= ~OPOST;
174
175 // 设置等待时间和最小接收字符
176 options.c_cc[VTIME] = 1; /* 读取一个字符等待1*(1/10)s */
177 options.c_cc[VMIN] = 0; /* 读取字符的最少个数为1 */
178
179 // 如果发生数据溢出, 接收数据, 但是不再读取 刷新收到的数据但是不读
180 // tcflush(fd, TCIFLUSH);
181
182 // 激活配置 (将修改后的termios数据设置到串口)
183 if (tcsetattr(fd, TCSANOW, &options) != 0)
184 {
185     perror("com set error!\n");
186     return FALSE;

```

```

187     }
188     return TRUE;
189 }
190 int uart_init(int fd, int speed, int flow_ctrl, int databits, int stopbits, i
nt parity)
191 {
192     int err;
193
194     if (uart_set(fd, speed, flow_ctrl, databits, stopbits, parity) == FALSE)
//设置串口数据帧格式
195     {
196         return FALSE;
197     } else {
198         return TRUE;
199     }
200 }
201
202
203 int uart_recv(int fd, char *rcv_buf, int data_len)
204 {
205     int len, fs_sel;
206     fd_set fs_read;
207
208     struct timeval time;
209
210     FD_ZERO(&fs_read);
211     FD_SET(fd, &fs_read);
212
213     time.tv_sec = 1;
214     time.tv_usec = 0;
215
216     fs_sel = select(fd+1, &fs_read, NULL, NULL, &time);
217     if(fs_sel > 0)
218     {
219         len = read(fd, rcv_buf, data_len);
220         return len;
221     } else if (fs_sel == 0){
222         return 0;
223     }
224 }
225
226 int uart_send(int fd, char *send_buf, int data_len)
227 {
228     int len = 0;
229
230     len = write(fd, send_buf, data_len);
231     if (len == data_len )
232     {

```

```

233         return len;
234     } else {
235         tcflush(fd,TCOFLUSH);
236         return FALSE;
237     }
238 }
239 static void * uart_tx_pthread(void *arg)
240 {
241     char test_cmd[] = {0x31, 0x32, 0x33};
242
243     uart_send(*(int*)arg, test_cmd, sizeof(test_cmd));
244 }
245
246 static void * uart_rx_pthread(void *arg)
247 {
248     char recv_buf[128];
249     int len = 0;
250     int i;
251     int ret;
252
253     memset(recv_buf, 0, sizeof(recv_buf));
254     while (1) {
255         ret = uart_recv(*(int*)arg, recv_buf+len, sizeof(recv_buf));
256         if (ret > 0){
257             len += ret;
258             printf(" len=%d\n", len);
259         }else if (ret == 0){
260             printf("timeout \n");
261             printf("recv:");
262             for (i=0; i<len; i++)
263             {
264                 printf("%02x \n",recv_buf[i]);
265             }
266             printf("\n");
267             break;
268         }
269         usleep(100000);
270     }
271
272     if ( (recv_buf[0] == 0x31) && (recv_buf[1] == 0x32) && (recv_buf[2] =
273 = 0x33) )
274     {
275         printf("check okay\n");
276         flag = 1;
277     }else{
278         printf("check fail\n");
279         flag = -0;
280     }

```

```

280 }
281
282
283 int main(int argc, char **argv)
284 {
285     int fd_r;
286     int fd_w;
287     int err;
288     int baud;
289     int databit;
290     int stopbit;
291     char parity;
292
293     if (argc != 3){
294         printf("usage: ./rs485_test /dev/ttyS2 /dev/ttyAS4\n");
295         return -1;
296     }
297
298     fd_r = uart_open(argv[1]);
299     if (fd_r < 0) {
300         printf("open %s fail!", argv[1]);
301         return fd_r;
302     }
303
304     fd_w = uart_open(argv[2]);
305     if (fd_w < 0) {
306         printf("open %s fail!", argv[2]);
307         close(fd_r);
308         return fd_w;
309     }
310     baud = 9600;
311     databit = 8;
312     stopbit = 1;
313     parity = 'N';
314
315     err = uart_init(fd_r,baud,0,databit, stopbit, parity);
316     if (err == FALSE){
317         printf("uart init err\n");
318         return -1;
319     }
320
321     err = uart_init(fd_w,baud,0,databit, stopbit, parity);
322     if (err == FALSE){
323         printf("uart init err\n");
324         return -1;
325     }
326
327     pthread_t pt1, pt2;

```

```

328     if (pthread_create(&pt1, NULL, uart_rx_thread, (void *)&fd_r) == -1)
329     {
330         printf("pthread_create failed\n");
331         return -1;
332     }
333     usleep(10000);
334     if (pthread_create(&pt2, NULL, uart_tx_thread, (void *)&fd_w) == -1)
335     {
336         printf("pthread_create failed\n");
337         return -1;
338     }
339     if (pthread_join(pt2, NULL))
340     {
341         printf("thread is not exit...\n");
342         return -2;
343     }
344     if (pthread_join(pt1, NULL))
345     {
346         printf("thread is not exit...\n");
347         return -2;
348     }
349     close(fd_r);
350     close(fd_w);
351     if (flag == 1){
352         return 0;
353     }else{
354         return -1;
355     }
356 }
357
358
359
360

```

6.开发案例

6.1.Qt5使用

详情参考

百度网盘：EVB5301-Linux/3.开发手册/《IDO-EVB5301-V1 QT应用开发手册》.pdf

6.2.MQTT使用

详情参考

百度网盘：EVB5301-Linux/3.开发手册/《IDO-EVB5301-V1 MQTT应用案例》.pdf

6.3.GDB使用

详情参考

百度网盘：EVB5301-Linux/3.开发手册/《IDO-EVB5301-V1 GDB开发手册》.pdf

6.4. LVGL使用

详情参考

百度网盘：EVB5301-Linux/3.开发手册/《IDO-EVB5301-V1 LVGL应用开发手册》.pdf