



# 软件应用指南

## ECB32 系列



## 目录

|                    |    |
|--------------------|----|
| 免责声明和版权公告 .....    | 1  |
| 1. 概述 .....        | 2  |
| 1.1. 软件资源 .....    | 2  |
| 2. 使用前准备 .....     | 2  |
| 2.1. 串口软件安装 .....  | 2  |
| 3. 快速开始 .....      | 2  |
| 3.1. 串口配置 .....    | 3  |
| 3.2. 镜像刻录 .....    | 3  |
| 3.3. 上电开机 .....    | 6  |
| 4. 快速体验 .....      | 6  |
| 4.1. CPU .....     | 6  |
| 4.2. GPU .....     | 9  |
| 4.3. NPU .....     | 10 |
| 4.4. 视频解码 .....    | 10 |
| 4.5. 内存 .....      | 11 |
| 4.6. eMMC 测试 ..... | 14 |
| 4.7. 看门狗 .....     | 16 |
| 4.8. 电源管理 .....    | 16 |
| 4.9. RTC .....     | 17 |
| 4.10. 网络 .....     | 18 |
| 4.11. 显示 .....     | 21 |
| 4.12. 摄像头 .....    | 23 |
| 4.13. TF 卡 .....   | 23 |
| 4.14. PCIE .....   | 24 |
| 4.15. USB .....    | 26 |
| 4.16. 音频 .....     | 29 |
| 4.17. LED .....    | 31 |
| 4.18. 按键 .....     | 32 |
| 4.19. 风扇 .....     | 34 |
| 4.20. GPIO .....   | 34 |
| 5. 参考资料 .....      | 35 |
| 6. 修订说明 .....      | 35 |
| 7. 关于我们 .....      | 35 |

## 免责声明和版权公告

本文中的信息，如有变更，恕不另行通知。文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中所得测试数据均为亿佰特实验室测试所得，实际结果可能略有差异。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

最终解释权归成都亿佰特电子科技有限公司所有。

### 注 意：

由于产品版本升级或其他原因，本手册内容有可能变更。亿佰特电子科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，成都亿佰特电子科技有限公司尽全力在本手册中提供准确的信息，但是成都亿佰特电子科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

## 1. 概述

本文着重介绍如何测试并使用亿佰特 ECB32 系列单板机提供的功能，若用户还需在此基础上移植适配特定功能，也请参考《ECK32\_T527\_Software\_Development\_Guide》，后文以“软件开发指南”指代该文档。本文可作为前期评估指导使用，也可以作为通用系统开发的测试指导书使用。

### 1.1. 软件资源

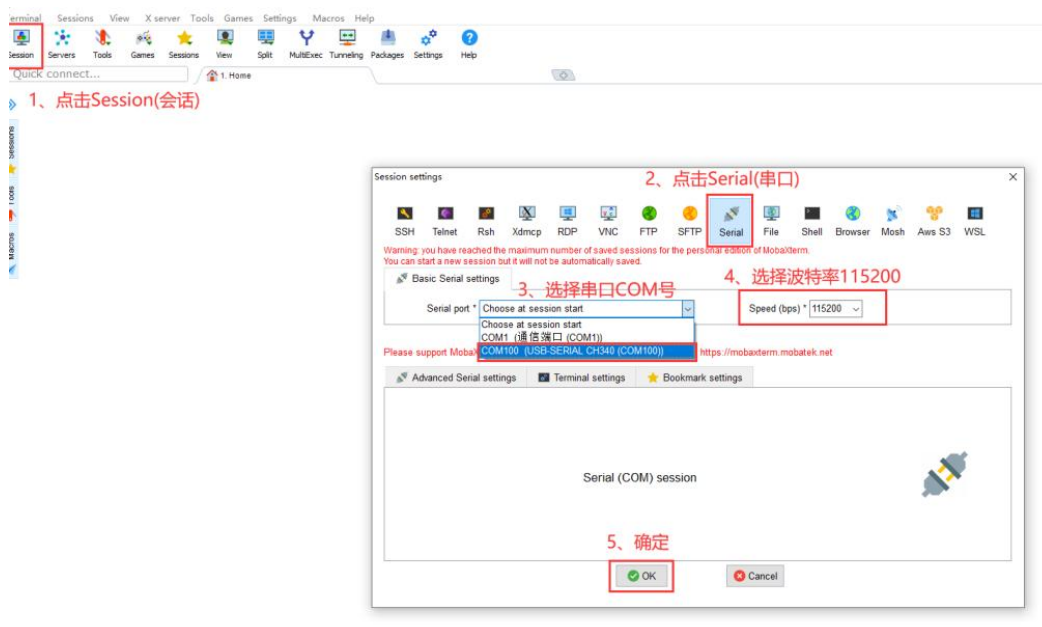
ECB32 搭载基于 Linux 5.15 版本操作系统，单板机与 Pi 板出厂附带嵌入式 Linux 系统开发所需要的交叉编译工具链，U-boot 源代码，Linux 内核和各驱动模块的源代码，以及适用于 Windows 桌面环境和 Linux 桌面环境的各种开发调试工具，应用开发样例等。

## 2. 使用前准备

### 2.1. 串口软件安装

使用串口前需要安装 CH340 串口驱动和 MobaXterm 串口终端。具体安装与使用方法可见软件开发指南。

连接好串口之后，安装 MobaXterm 后进行配置，按如下方式打开串口。

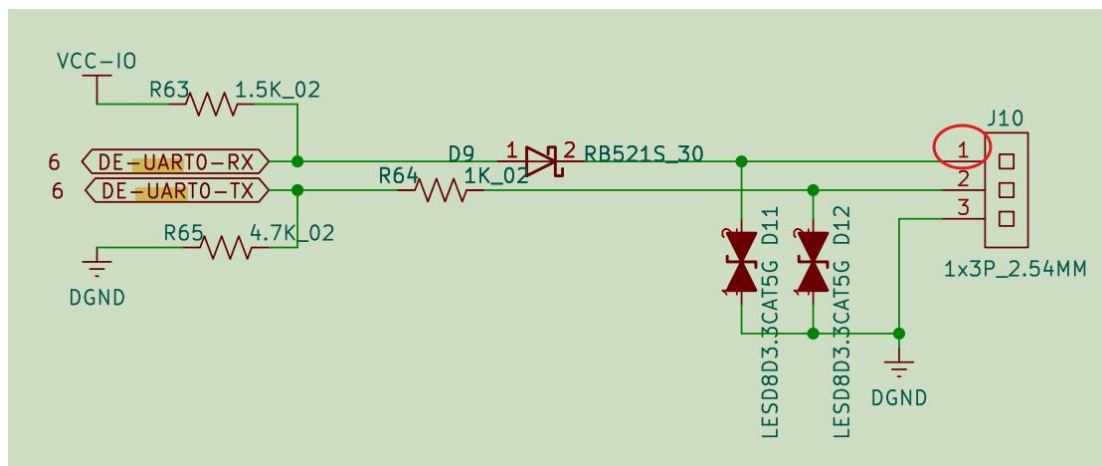


## 3. 快速开始

## 3.1. 串口配置

T527 使用的串口使用的波特率为 115200，无硬件流控。

对于 Pi 板，使用 TTL 串口连接，位于耳机座旁边，丝印有一个小箭头的对应下面原理图中 1 脚。



对于单板机，直接使用 USB 数据线即可，无需区分连线。

## 3.2. 镜像刻录

亿佰特出厂的单板机与 Pi 板默认不带启动固件，需要用户自行烧录固件到板上 eMMC 或者 SD 卡。用户可以根据单板机型号来选择烧录固件，这里我们主要帮助用户快速启动单板机，只讲解使用官方工具烧录固件到 eMMC 的方法。更多烧录方法和烧录中的问题见软件开发指南。

### 3.2.1. 镜像选择

镜像说明表

| 镜像名                                        | 说明                                     |
|--------------------------------------------|----------------------------------------|
| t527_linux_ebyte_linux_t527_uart0_hdmi.img | 单板机使用 HDMI 和 DP 接口显示，Debian 11 系统      |
| t527_linux_ebyte_linux_t527_uart0_mipi.img | 单板机使用双 MIPI-DSI 接口显示，Debian 11 系统      |
| a527_linux_pi_ebyte_a527_uart0_hdmi.img    | A527 Pi 板使用 HDMI 显示，Debian 12 系统       |
| a527_linux_pi_ebyte_a527_uart0_mipi.img    | A527 Pi 板使用 MIPI 显示，无需显示屏，Debian 12 系统 |
| a527_linux_pi_ebyte_a527_uart0_nor.img     | A527 Pi 板用于 nor-flash 启动的镜像，使用 ramfs   |
| t527_linux_pi_ebyte_t527_uart0_hdmi.img    | T527 Pi 板使用 HDMI 显示，Debian 12 系统       |
| t527_linux_pi_ebyte_t527_uart0_mipi.img    | T527 Pi 板使用 MIPI 显示，无需显示屏，Debian 12 系统 |
| t527_linux_pi_ebyte_t527_uart0_nor.img     | T527 Pi 板用于 nor-flash 启动的镜像，使用 ramfs   |

**HDMI 后缀镜像必须连接显示器使用，nor-flash 镜像不支持显示与 NPU 功能。**

对于 nor-flash, 可以修改 uboot 的 env 变量指定 rootfs 的路径, 通过 ramfs 跳转到新的系统。

### 3.2.2. 烧录板载 eMMC

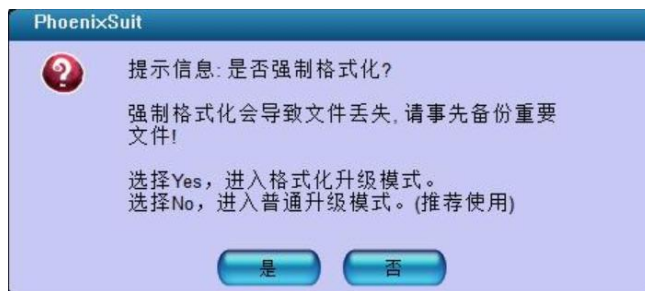
首先需要使用 usb 数据线连接单板机和电脑, 接入单板机的 OTG 接口。然后使用全志官方工具 PhoenixSuit 实现 USB 快速烧录。



点击上方的“一键刷机”项, 进入到镜像选择, **注意不要点击立即升级按钮。**

接下来我们使用 type-c 数据线连接 OTG 接口与 PC, 连接好之后, 我们需要进入烧写模式。

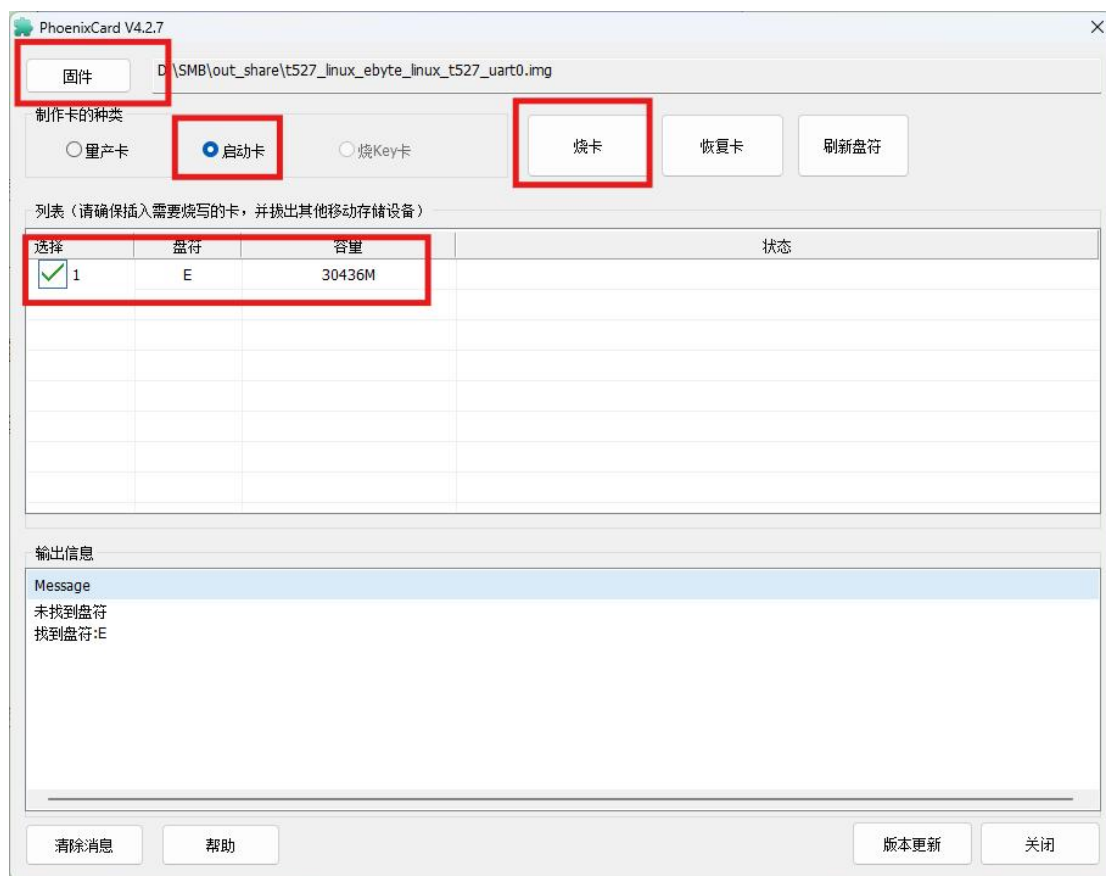
ECB32 引出了 FEL 按键, 用户可以按下 FEL 然后保持, 再按下复位按键或者给板子上电, 即可进入烧写模式, 弹出烧写询问框。若用户不方便按 FEL 按键, 可以先打开串口终端软件, 然后在串口终端按住小键盘 2, 然后再按下复位按键或者给板子上电, 即可进入烧写模式, 烧写工具会弹出烧写询问框。



用户按需选择是否格式化升级，等待烧录完成即可，一般默认使用 Yes 即可。

### 3.2.3. 烧录 SD 卡

除了直接烧录到板上的 eMMC 中，还可以选择烧写到 SD 卡后启动。这里我们使用全志提供的 PhoenixCard 来进行 SD 卡的烧录。



用户请依次选择待烧写固件，固件根据烧录的板子选择对应镜像即可。选择卡的种类为“启动卡”，而后选择待烧写的 SD 卡设备，该软件会自动扫描可用的移动存储设备，请注意不要选择了错误的设备。最后点击“烧卡”按钮，烧卡过程中会实时显示当前进度，直到进度条变为绿色表示烧录完毕。

需要注意 T527 启动顺序为 SD -> eMMC -> other(nor-flash etc)。也即只要存在可以启动的 SD 卡，就不会从 eMMC 启动。烧录到板上 FLASH 之后，拔掉 SD 卡才能从板上 eMMC 启动。

### 3.3.上电开机

需要注意，如果选择的显示接口不是 **mipi-dsi** 作为显示输出，则在上电之前需要连接其中至少一种显示器，否则 **uboot** 无法引导至内核或进入内核后系统由于无法提供显示功能出错。

## 4. 快速体验

本章节将按照功能模块进行测试，提供一种初步的功能运用方式，在不明确指明使用的系统时，单板机默认使用 **buildroot** 构建的系统进行验证，而 **Pi** 板默认使用 **debian** 系统验证。

### 4.1.CPU

#### 4.1.1. 查看 CPU 信息

使用如下命令查看 CPU 信息。

```
# cat /proc/cpuinfo

processor      : 0

BogoMIPS      : 48.00

Features       : fp asimd aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid a
simdrdm lrpc dcpop asimddp

CPU implementer : 0x41

CPU architecture: 8

CPU variant    : 0x2

CPU part       : 0xd05

CPU revision   : 0


processor      : 1

BogoMIPS      : 48.00

Features       : fp asimd aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid a
```

```

simdrdm lrpc dcpop asimddp

CPU implementer : 0x41

CPU architecture: 8

CPU variant      : 0x2

CPU part         : 0xd05

CPU revision     : 0

.....

processor        : 7

BogoMIPS         : 48.00

Features         : fp asimd aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid a
simdrdm lrpc dcpop asimddp

CPU implementer : 0x41

CPU architecture: 8

CPU variant      : 0x2

CPU part         : 0xd05

CPU revision     : 0
  
```

**processor:** 系统中逻辑处理核的编号，对于多核处理器则可以是物理核、或者使用超线程技术虚拟的逻辑核。

**BogoMIPS:** 在系统内核启动时粗略测算的 CPU 每秒运行百万条指令数（Million Instructions Per Second）

**Features:** CPU 支持的指令功能

#### 4.1.2. 查看 CPU 使用率

使用如下命令查看 CPU 使用率

```

# htop

0[          0.0%]  4[*****100.
0%]

1[          0.0%]  5[          0.
0%]

2[          0.0%]  6[          0.
  
```

|        |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|--------|--------|--------|-----|----|-------------|----------------------------------------|-------|---|------|-------|---------|-----------------|
| 0%]    |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 3[     |        |     |    |             | 0.0%]                                  | 7[    |   |      |       | 0.      |                 |
| 0%]    |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | Mem[   | #*@@@@ |     |    | 258M/1.88G] | Tasks: 20, 45 thr, 124 kthr; 1 running |       |   |      |       |         |                 |
|        | Swp[   |        |     |    | 0K/0K]      | Load average: 1.41 0.31 0.10           |       |   |      |       |         |                 |
|        |        |        |     |    |             | Uptime: 00:00:13                       |       |   |      |       |         |                 |
|        | [Main] | [I/O]  |     |    |             |                                        |       |   |      |       |         |                 |
|        | PID    | USER   | PRI | NI | VIRT        | RES                                    | SHR   | S | CPU% | -MEM% | TIME+   | C               |
| ommand |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 626    | root   | 20  | 0  | 4268        | 2552                                   | 2068  | R | 66.7 | 0.1   | 0:00.08 | htop            |
|        | 1      | root   | 20  | 0  | 3316        | 1688                                   | 1552  | S | 0.0  | 0.1   | 0:00.63 | init            |
|        | 178    | root   | 20  | 0  | 3316        | 528                                    | 456   | S | 0.0  | 0.0   | 0:00.02 | /sbin/syslogd - |
|        | 182    | root   | 20  | 0  | 3316        | 556                                    | 488   | S | 0.0  | 0.0   | 0:00.03 | /sbin/klogd -n  |
|        | 194    | root   | 20  | 0  | 15968       | 2676                                   | 1656  | S | 0.0  | 0.1   | 0:00.33 | /sbin/udev      |
|        | 284    | root   | 20  | 0  | 3744        | 228                                    | 0     | S | 0.0  | 0.0   | 0:00.00 | dbus-daemon -   |
| -s     |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 288    | root   | 20  | 0  | 2536        | 176                                    | 0     | S | 0.0  | 0.0   | 0:00.00 | /usr/sbin/rpcbi |
|        | 303    | root   | 20  | 0  | 404M        | 77012                                  | 59384 | S | 0.0  | 3.9   | 0:00.41 | weston --back   |
| en     |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 307    | root   | 20  | 0  | 3196        | 2364                                   | 1784  | S | 0.0  | 0.1   | 0:00.04 | dhcpcd: [mana   |
| ge     |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 327    | root   | 20  | 0  | 404M        | 77012                                  | 59384 | S | 0.0  | 3.9   | 0:00.00 | weston --back   |
| en     |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 328    | root   | 20  | 0  | 404M        | 77012                                  | 59384 | S | 0.0  | 3.9   | 0:00.00 | weston --back   |
| en     |        |        |     |    |             |                                        |       |   |      |       |         |                 |
|        | 329    | root   | 20  | 0  | 404M        | 77012                                  | 59384 | S | 0.0  | 3.9   | 0:00.00 | weston --back   |
| en     |        |        |     |    |             |                                        |       |   |      |       |         |                 |

### 4.1.3. 查看 CPU 温度

使用如下命令查看 CPU 当前温度:

```
# cat /sys/class/thermal/thermal_zone*/type

cpul_thermal_zone

cpub_thermal_zone

gpu_thermal_zone

npu_thermal_zone

ddr_thermal_zone

axp2202-usb

# cat /sys/class/thermal/thermal_zone*/temp

38702

38036

38480

38480

33374

32900
```

通过 type 的类型, 可以发现我们需要重点关注的模块对应的节点是哪一个, 然后该对应节点的 temp 返回值, 我们除以 1000 就是实际温度, 比如 cpub\_thermal\_zone 对应的 temp 值为 38036, 实际温度为 38 摄氏度。

## 4.2.GPU

对于 T527 携带的 GPU(mali-g57), 当连接了对应显示屏同时桌面启动后, 执行如下命令可以查看 GPU 占用情况:

```
# 查看 gpu 占用

# mtop
```

在 buildroot 构建的系统上可以通过如下命令测试 gpu 性能:

```
# export XDG_RUNTIME_DIR="/tmp/wayland"
```

```
# glmark2-es2-wayland
```

在 Debian bookworm 构建的系统上可以通过如下命令测试 gpu 性能:

```
# export XDG_RUNTIME_DIR="/run/user/0"
# glmark2-es2-wayland
```

在 Debian bullseye 构建的系统上可以通过如下命令测试 gpu 性能:

```
# export DISPLAY=:0.0
# glmark2
```

对于 Debian bullseye, 由于使用的 X11 进行测试, 其测试成绩会弱于 bookworm, 但实际性能区别不大, 都是使用的 drm + Wayland 运行。

## 4.3.NPU

本司部分型号中提供了一个 2 TOPS 的 NPU, 相关测试使用流程可以查看开发笔记中的《t527\_npu》文档, 本文不在赘述。

## 4.4.视频解码

当前使用 gstreamer 提供视频解码功能, 用户可以使用如下命令播放一个 H264 的视频:

在 buildroot 构建的系统上可以通过如下命令测试 gpu 性能:

```
# export XDG_RUNTIME_DIR="/tmp/wayland"
# gst-play-1.0 your.mp4
```

在 Debian 构建的系统上可以通过如下命令测试 gpu 性能:

```
# export XDG_RUNTIME_DIR="/run/user/0"
# gst-play-1.0 your.mp4
```

对于 debian 系统, 可以安装使用 totem 播放器, 该播放器后端使用 gstreamer 提供解码支持, 对于超出 gstreamer 的部分, 使用软件解码。

除此之外, 也可以使用 gst-lunch-1.0 播放, 提供更详细的配置。

## 4.5.内存

### 4.5.1. 内存信息

使用如下命令查看内存信息:

```
# cat /proc/meminfo

MemTotal:          1968224 kB
MemFree:           1491108 kB
MemAvailable:      1661780 kB
Buffers:            5272 kB
Cached:            180740 kB
SwapCached:         0 kB
Active:            15232 kB
Inactive:          261972 kB
Active(anon):       520 kB
Inactive(anon):    108548 kB
Active(file):       14712 kB
Inactive(file):    153424 kB
Unevictable:        0 kB
Mlocked:            0 kB
SwapTotal:          0 kB
SwapFree:           0 kB
Dirty:              0 kB
Writeback:          0 kB
AnonPages:          91280 kB
Mapped:            110508 kB
Shmem:              17876 kB
KReclaimable:       27216 kB
Slab:               87728 kB
```

|               |              |
|---------------|--------------|
| SReclaimable: | 27216 kB     |
| SUnreclaim:   | 60512 kB     |
| KernelStack:  | 3104 kB      |
| PageTables:   | 1872 kB      |
| NFS_Unstable: | 0 kB         |
| Bounce:       | 0 kB         |
| WritebackTmp: | 0 kB         |
| CommitLimit:  | 984112 kB    |
| Committed_AS: | 630164 kB    |
| VmallocTotal: | 259653632 kB |
| VmallocUsed:  | 6476 kB      |
| VmallocChunk: | 0 kB         |
| Percpu:       | 864 kB       |
| CmaTotal:     | 65536 kB     |
| CmaFree:      | 64356 kB     |

**MemTotal:** 总内存量。这里显示的值为 498788 kB，表示系统总共有大约 487 MB 的物理内存可用。由于此处换算方式为 1024，而常规销售过程中一般使用 1000 换算，故此处大小也可看做 510.76MB，在加上部分内存留作它用，此处大小也算正常。

**MemFree:** 空闲内存量。这里显示的值为 369352 kB，表示当前系统中有大约 360 MB 的内存是空闲的，未被使用。

**MemAvailable:** 可用内存量。这里显示的值为 396952 kB，表示当前可供系统使用的内存总量，包括已缓存的内存和可用的内存。

**Buffers:** 缓冲区使用量。这里显示的值为 0 kB，表示系统当前没有使用任何内存作为缓冲区。

**Cached:** 缓存的内存量。这里显示的值为 39208 kB，表示系统当前用于缓存的内存量。

**SwapCached:** 交换缓存的内存量。这里显示的值为 0 kB，表示当前没有被缓存到交换空间中的内存。

**Active:** 活跃的内存量。这里显示的值为 8576 kB，表示当前正在使用的内存量。

## 4.5.2. 内存测试

使用如下命令获取可用于测试的内存大小:

```
# free -h
```

|       | total | used   | free | shared | buff/cache | available |
|-------|-------|--------|------|--------|------------|-----------|
| Mem:  | 1.9G  | 257.2M | 1.4G | 17.5M  | 208.2M     | 1.6G      |
| Swap: | 0     | 0      | 0    |        |            |           |

此处可用于测试的内存大小最大为 1.6G，但是考虑到后续其他程序的稳定运行，下面以 100MB 进行一次内存测试为例:

```
# memtester 100M 1

memtester version 4.5.0 (64-bit)

Copyright (C) 2001-2020 Charles Cazabon.

Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xfffffffffff000
want 100MB (104857600 bytes)
got 100MB (104857600 bytes), trying mlock ...locked.

Loop 1/1:

Stuck Address      : ok
Random Value       : ok
Compare XOR        : ok
Compare SUB        : ok
Compare MUL        : ok
Compare DIV        : ok
Compare OR         : ok
Compare AND        : ok
Sequential Increment: ok
```

```
Solid Bits      : ok
Block Sequential : ok
Checkerboard     : ok
Bit Spread       : ok
Bit Flip         : ok
Walking Ones     : ok
Walking Zeroes   : ok
8-bit Writes     : ok
16-bit Writes    : ok
```

Done.

可以看到此处全部测试项都是`ok`状态，表示测试通过，内存无明显错误。

**注意：**对于 Debian 系统，默认不带 memtester，需要手动通过 apt 安装。

## 4.6.eMMC 测试

### 4.6.1. 查看 eMMC 信息

使用如下命令获取 eMMC 设备容量大小与分区情况表，对于使用 SD 卡启动的用户，使用/dev/mmcblk1 查看 eMMC 信息：

```
# fdisk -l /dev/mmcblk0

Disk /dev/mmcblk0: 14.58 GiB, 15655239680 bytes, 30576640 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: gpt

Disk identifier: AB6F3888-569A-4926-9668-80941DCB40BC
```

| Device | Start | End | Sectors | Size | Type |
|--------|-------|-----|---------|------|------|
|--------|-------|-----|---------|------|------|

```
/dev/mmcblk0p1    73728    139263    65536    32M Microsoft basic data
/dev/mmcblk0p2    139264    172031    32768    16M Microsoft basic data
/dev/mmcblk0p3    172032    368639    196608    96M Microsoft basic data
/dev/mmcblk0p4    368640    17145855 16777216    8G Microsoft basic data
/dev/mmcblk0p5    17145856 30576606 13430751    6.4G Microsoft basic data
```

此处 eMMC 分为了五个分区，p4 分区用于文件系统。可通过下面命令获取分区类型：

```
# lsblk -f

NAME                FSTYPE FSVER LABEL  UUID
FSAVAIL FSUSE% MOUNTPOINTS

mmcblk0

|-mmcblk0p1 vfat   FAT16 Volumn

|-mmcblk0p2

|-mmcblk0p3

|-mmcblk0p4 ext4    1.0          57f8f4bc-abf4-655f-bf67-946fc0f9f25b 7.2G
7% /var/lib/docker

|

/

`-mmcblk0p5
```

#### 4.6.2. eMMC 读写

可以使用如下命令测试 eMMC 读写性能：

```
# time dd if=/dev/zero of=test.bin bs=1M count=1024 conv=fsync

1024+0 records in

1024+0 records out

real    0m9.172s
```

```
user    0m0.023s

sys     0m3.991s

# echo 3 > /proc/sys/vm/drop_caches

# time dd if=test.bin of=/dev/null bs=1M

1024+0 records in

1024+0 records out

real    0m4.573s

user    0m0.000s

sys     0m1.192s
```

根据上面的内容，我们可以得知，当前单板机的写入速度为  $1024/9.172 = 111.6\text{MB/s}$ ，读取速度为  $1024/4.572=223.9\text{MB/s}$ 。

同时上面的`echo 3 > /proc/sys/vm/drop\_caches`是为了消除系统缓存的影响，获取实际的读取速度。

## 4.7.看门狗

使用如下命令配置看门狗超时时间为 5 秒并操作让单板机超时重启：

```
# watchdog -T 5 -t 10 /dev/watchdog
```

在该流程中，会在 5s 左右的时间后系统自动重启。

## 4.8.电源管理

本章节演示 Linux 电源管理的 Suspend 功能，让板子睡眠，通过外部事件唤醒。

Linux 内核提供了多种 Suspend 方式，在 ECB32 系列单板机上，支持如下：

```
# cat /sys/power/state

freeze mem
```

在用户空间向"/sys/power/state"文件分别写入"freeze"和"mem"，即可进入对应休眠模式。进入该模式后会发现功率降低。**需要注意：**如果使用 mem 进入休眠，单板机中只有 PL 组

引脚还会持续供电，如果需要额外的唤醒功能，请使用这部分引脚实现。如不需要，可短按 power 键唤醒：

```
# 提供 freeze 休眠后唤醒功能，休眠后任意从串口输入即可唤醒

# echo enabled > /sys/class/tty/ttyAS0/power/wakeup

# echo freeze > /sys/power/state

# echo mem > /sys/power/state
```

如果使用 Debian 系统，按了 power 键后自动关机，请务必配置 power button 不做任何事，如 gnome 桌面下，在 settings -> power -> Power Button Behavior，配置为`Nothing`。

## 4.9.RTC

Linux 系统中分系统时钟（软件时钟）和 RTC 时钟（硬件时钟），系统时钟掉电即会消失，RTC 时钟在安装电池的情况下会长期运行。如需使用外部 RTC 时钟，请将 ML2032（3V 可充）或 CR2032（3V 不可充）电池安装至 RTC 纽扣电池座，单板机上标识为`VBAT`。

1. 设置当前时间：

```
# date -s "2010-10-10 10:10:10"

Sun Oct 10 10:10:10 CST 2010
```

2. 写入系统时间到外部 RTC，并查询是否写入：

```
# hwclock -w -f /dev/rtc1

# hwclock -f /dev/rtc1

Sun Oct 10 10:10:12 2010 0.000000 seconds
```

3. 关机拔出供电并等待一定时间后重新上电查看 RTC 时间：

```
# hwclock -f /dev/rtc1

Sun Oct 10 10:11:24 2010 0.000000 seconds
```

## 4.10.网络

### 4.10.1. ETH

当前单板机存在两个网口, Pi 板存在一个网口, 使用 net-tools 工具包中的 ifconfig 对网络进行手动配置, 首先通过 ifconfig 命令查看网络设备信息如下:

```
# ifconfig

docker0  Link encap:Ethernet  HWaddr 02:42:29:6D:EC:BD

        inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0

        UP BROADCAST MULTICAST  MTU:1500  Metric:1

        RX packets:0 errors:0 dropped:0 overruns:0 frame:0

        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

        collisions:0 txqueuelen:0

        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)


eth0      Link encap:Ethernet  HWaddr D2:09:15:64:BA:64

        inet addr:192.168.20.31  Bcast:192.168.20.255  Mask:255.255.255.0

        inet6 addr: fe80::2a5:c1ad:5f72:115a/64 Scope:Link

        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

        RX packets:538 errors:0 dropped:0 overruns:0 frame:0

        TX packets:102 errors:0 dropped:0 overruns:0 carrier:0

        collisions:0 txqueuelen:1000

        RX bytes:274417 (267.9 KiB)  TX bytes:8614 (8.4 KiB)

        Interrupt:98


eth1      Link encap:Ethernet  HWaddr 46:3E:24:75:A2:77

        UP BROADCAST MULTICAST  MTU:1500  Metric:1

        RX packets:0 errors:0 dropped:0 overruns:0 frame:0

        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```

```

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

Interrupt:99

lo    Link encap:Local Loopback

    inet addr:127.0.0.1  Mask:255.0.0.0

    inet6 addr: ::1/128 Scope:Host

    UP LOOPBACK RUNNING  MTU:65536  Metric:1

    RX packets:130 errors:0 dropped:0 overruns:0 frame:0

    TX packets:130 errors:0 dropped:0 overruns:0 carrier:0

    collisions:0 txqueuelen:1000

    RX bytes:9612 (9.3 KiB) TX bytes:9612 (9.3 KiB)
  
```

当前的根文件系统配置了自动 DHCP 功能，但用户也可以手动配置一个静态 ip。

#### 4.10.1.1. 静态地址

下面介绍在 buildroot 系统下给 eth0 手动配置 IP 地址 192.168.1.253 的方法，命令如下：

```

# ifconfig eth0 192.168.0.253 netmask 255.255.255.0 up

# ifconfig

eth0    Link encap:Ethernet  HWaddr 4E:42:0B:5A:EE:7F

    inet addr:192.168.0.253  Bcast:192.168.0.255  Mask:255.255.255.0

    UP BROADCAST MULTICAST  MTU:1500  Metric:1

    RX packets:118 errors:0 dropped:0 overruns:0 frame:0

    TX packets:92 errors:0 dropped:0 overruns:0 carrier:0

    collisions:0 txqueuelen:1000

    RX bytes:13846 (13.5 KiB) TX bytes:8610 (8.4 KiB)
  
```

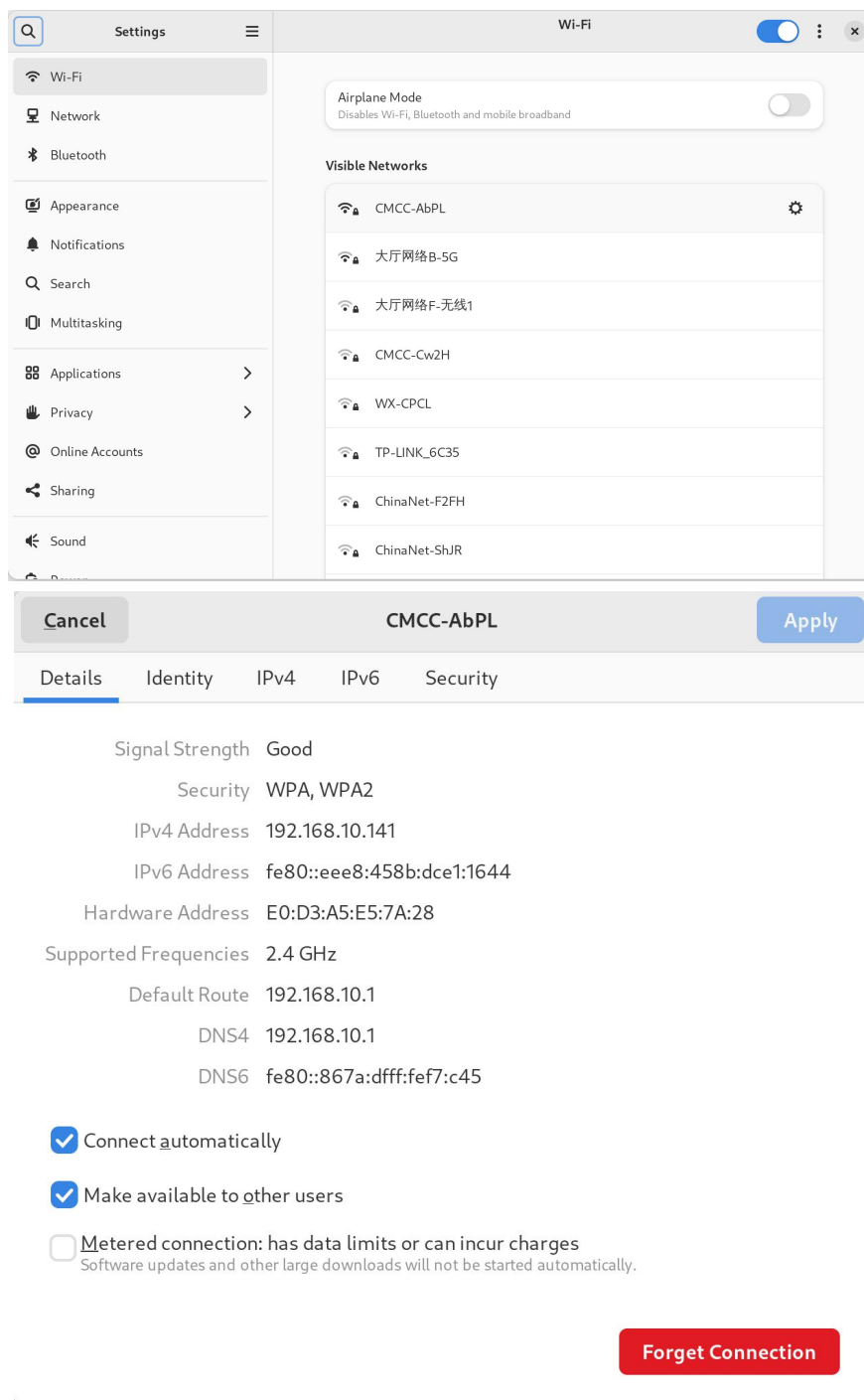
对于使用 Debian 系统的情况，使用 nmtui 或使用 gnome-settings 配置即可，无需使用命令配置。

## 4.10.2. WIFI/BT

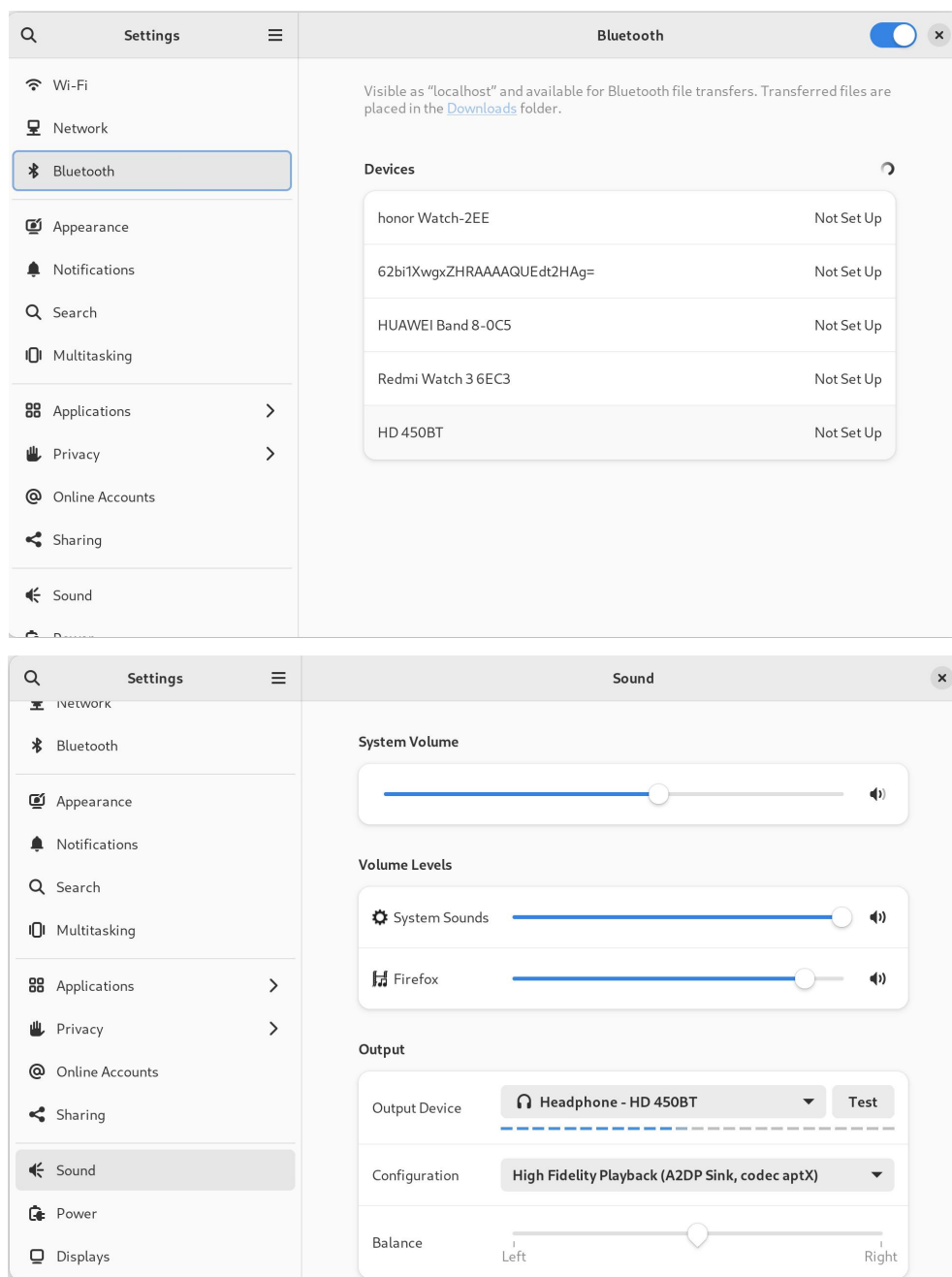
该节仅限于 Pi 板支持，单板机无相应器件，强烈建议添加天线使用，否则信号极弱且可能损坏器件。

该功能可以在 Debian 系统中通过 UI 画面配置，简单方便。对于使用 nmtui 与 bluez-utilities 的配置方式，本文不在赘述。

### 4.10.2.1. Wifi



## 4.10.2.2. Bluetooth



## 4.11.显示

Debian + gnome: 提供 Pi 板以及单板机的 HDMI 和 DP 显示。

Buildroot + weston: 提供单板机的 MIPI 显示，用户可自行配置为使用 Debian 系统。

### 4.11.1. Gnome

单板机支持 HDMI 与 DP 接口共用，而 Pi 板没有 DP 接口。在没有与 MIPI-DSI 构成双屏的时候，需要注意在上电之前必须连接 HDMI/DP 显示器，否则由于无法通过 EDID 获取屏幕参数，停止后续启动。考虑 HDMI 与 DP 的显示情况，使用 Debian 的 gnome 桌面提

供支持。

连接上屏幕后会通过 EDID 信息获取屏幕参数，自动选择最佳分辨率显示，用户也可进入系统后通过 `gnome` 配置不同的分辨率。

如果 `gnome` 出现错误，可以通过在串口中执行如下命令重启桌面

```
systemctl restart gdm3
```

注：由于 DP 接口需要做训练，如果系统启动后 DP 显示器无法显示内容，可尝试复位系统，重新训练，或可正常显示。但也不排除部分显示器不兼容问题。

#### 4.11.1.1. 中文界面

对于当前系统，默认显示为英文，用户可以通过如下命令配置使用中文显示

```
dpkg-reconfigure locales  
  
reboot
```

执行后会用户需要启用 `'zh_CN.UTF-8 UTF-8'` 支持，并选择使用其为默认，等待系统重启后即显示为中文界面。

此时在串口会也会打印中文，但可能存在部分乱码，可以在 `/root/.bashrc` 中添加如下命令强制指定在串口上显示英文：

```
if [ $XDG_SESSION_TYPE = "tty" ]; then  
    export LANG='C.UTF-8'  
fi
```

#### 4.11.2. Weston

单板机的 MIPI-DSI 使用的系统为 `buildroot` 构建，桌面环境为 `weston`，配置文件位于 `/etc/xdg/weston`。

正常进入系统后会显示如下图像：



其他更具体的 weston 使用请自行查询相关资料。

## 4.12. 摄像头

T527 提供了最高 8M@30fps RAW12 2F-WDR, size up to 3264(H) x 2448(V) 的 MIP I-CSI 接口, 详细使用方式参考开发笔记中的《T527\_CSI》文档, 本文不在赘述。

## 4.13. TF 卡

SD 卡位于单板机背面, 丝印为 TF Card。

### 4.13.1. 查看 TF 卡信息

通过如下命令可以查看当前 TF 卡的状态信息:

```
# mount -t debugfs debugfs /sys/kernel/debug/  
  
# cat /sys/kernel/debug/mmc*/ios  
  
# fdisk -l
```

其中, 查看 ios 信息后, 显示 SD 相关的节点表示为 SD 卡, MMC 相关信息的节点表示 eMMC, 记住此时 mmc 是 0 还是 1。如从 SD 卡启动, 则一般情况下 SD 卡对应的设备节点为 mmc0, 其对应的块设备为/dev/mmcblk0。

### 4.13.2. 测试 TF 卡速度

通过如下命令测试 SD 卡读写速度:

```
# mkdir -p /root/tmp

# mount /dev/mmcblk0p4 /root/tmp

# time dd if=/dev/zero of=/root/tmp/test.bin bs=1M count=4096 conv=fsync

# echo 3 > /proc/sys/vm/drop_caches

# time dd if=/mnt/test.bin of=/dev/zero bs=1M

# umount /root/tmp
```

## 4.14.PCIE

T527 系列支持 PCI 接口，可通过转接板连接 m.2 硬盘等。对于单板机，其 PCI 接口为 MIPI-PCI，还额外支持一路 USB 2.0 接口，可以连接 4G 模块等。

此处以连接硬盘为例使用。硬盘型号为长江存储，PC210。通过如下命令测试读写情况：

```
# fdisk /dev/nvme0n1

Welcome to fdisk (util-linux 2.38).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.


Command (m for help): n

Partition type

   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)


Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-250069679, default 2048): 2048

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-250069679, default 250069679):
+10G
```

Created a new partition 1 of type 'Linux' and of size 10 GiB.

Partition #1 contains a ext4 signature.

**Do you want to remove the signature? [Y]es/[N]o: Y**

The signature will be removed by a write command.

**Command (m for help): w**

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

```
# mkfs.ext4 /dev/nvme0n1p1
```

```
# mkdir -p /root/tmp
```

```
# mount /dev/nvme0n1p1 /root/tmp
```

```
# time dd if=/dev/zero of=/root/tmp/test.bin bs=1M count=4096 conv=fsync
```

```
# echo 3 > /proc/sys/vm/drop_caches
```

```
# time dd if=/mnt/test.bin of=/dev/zero bs=1M
```

```
4096+0 records in
```

```
4096+0 records out
```

```
real    0m10.157s
```

```
user    0m0.032s
```

```
sys     0m6.226s
```

```
# umount /root/tmp
```

通过上面的情况可以发现，该硬盘在当前环境下读取速度为  $4096/10.157=403\text{MB/s}$ 。

## 4.15.USB

### 4.15.1. OTG

#### 4.15.1.1.Buildroot

在单板机上存在一路 USB-OTG 接口，通过 typec 转 usb 连接器连接 usb 设备，将会自动挂载，比如鼠标，u 盘。

连接 PC 机与单板机的 OTG 接口，打开全志的烧写软件 PhoenixSuit，默认会发现如下信息则表明 OTG 功能正常使用：



#### 4.15.1.2.Debian

对于 Debian 系统，默认没有提供 adb 功能，用户可以连接电脑后使用如下脚本把自身模拟为一个 U 盘使用：

```
#!/bin/bash

if ! grep -q "/sys/kernel/config" /proc/mounts; then
    mount -t configfs configfs /sys/kernel/config
fi
```

```
if ! which mkfs.fat; then

    apt update && apt install -y dosfstools

fi

dd if=/dev/zero of=/root/a.bin bs=10M count=10

mkfs.fat -F 32 /root/a.bin

mkdir -p /sys/kernel/config/usb_gadget/g1

echo "0x18d1" > /sys/kernel/config/usb_gadget/g1/idVendor

echo "0x0001" > /sys/kernel/config/usb_gadget/g1/idProduct

mkdir -p /sys/kernel/config/usb_gadget/g1/strings/0x409

mkdir -p /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0

echo /root/a.bin > /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/lun.0/file

mkdir -p /sys/kernel/config/usb_gadget/g1/configs/c.1

echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes

echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower

mkdir -p /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409

ln -sf /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/

ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

想要结束模拟可以使用如下命令:

```
if [ -d /sys/kernel/config/usb_gadget/g1 ]; then

    echo "" > /sys/kernel/config/usb_gadget/g1/UDC 2>/dev/null

    rm -f /sys/kernel/config/usb_gadget/g1/configs/c.1/mass_storage.usb0

    rmdir /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0 2>/dev/null

    rmdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409 2>/dev/null

    rmdir /sys/kernel/config/usb_gadget/g1/configs/c.1 2>/dev/null

    rmdir /sys/kernel/config/usb_gadget/g1/strings/0x409 2>/dev/null

    rmdir /sys/kernel/config/usb_gadget/g1 2>/dev/null

fi
```

该功能不适用于 Pi 板，供电接口与 OTG 共用，模拟后会导致 PC 提供的供电错乱出现未知问题。同时大部分情况电脑 USB 接口提供的功率不够，需要专用充电器，Pi 板上该接口只推荐在烧录时连接 PC 其余情况请使用专用充电器作为供电使用。

## 4.15.2. HOST

本节通过相关命令或热插拔、USB HUB 验证 USB Host 驱动的可行性，实现读写 U 盘的功能、usb 枚举功能。

插入 U 盘后可通过如下命令查找存储设备。

```
# lsblk

NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                  8:0    1  28.7G  0 disk
`-sda1               8:1    1  28.7G  0 part
mmcblk0             179:0    0  14.6G  0 disk
|-mmcblk0p1         179:1    0    32M  0 part
|-mmcblk0p2         179:2    0    16M  0 part
|-mmcblk0p3         179:3    0    96M  0 part
|-mmcblk0p4         179:4    0     8G  0 part /var/lib/docker
|
|-mmcblk0p5         179:5    0    6.4G  0 part
nvme0n1             259:0    0 119.2G  0 disk
`-nvme0n1p1         259:1    0    10G  0 part /mnt
```

可以看到此处多出了一个 sda 设备，将 sda1 挂载到 /root/tmp 中，进入该目录可以发现 U 盘中存在的文件，且可以通过如下命令测试当前 U 盘的读写性能：

```
# dd if=/dev/zero of=/root/dd.bin bs=1M count=128 conv=fsync

# echo 3 > /proc/sys/vm/drop_caches

# time dd of=/dev/zero if=/root/tmp/dd.bin bs=1M

128+0 records in

128+0 records out
```

```
real    0m3.881s
user    0m0.000s
sys     0m0.236s
```

可以看到此时的 U 盘读取速度为 32.98MB/s, 符合一个 USB 2.0 U 盘的正常读取速度。

## 4.16. 音频

单板机中通过提供了音频功能, 支持从 HPOUT 输出音频, 同时通过 MIC1 录制音频, 默认已经设置完毕, 用户无需关心。**Pi** 板还额外提供从 **HDMI** 输出音频的功能, 但默认情况还是从 **HPOUT** 输出, 用户需要在相应的桌面环境中配置默认从 **HDMI** 输出。

通过如下命令可以查看当前 codec 支持的功能情况:

```
# aplay --dump-hw-params /dev/zero

Playing raw data '/dev/zero' : Unsigned 8 bit, Rate 8000 Hz, Mono

HW Params of device "PlaybackHP":
-----

ACCESS:  MMIO_INTERLEAVED RW_INTERLEAVED
FORMAT:  S16_LE S24_LE S32_LE
SUBFORMAT:  STD
SAMPLE_BITS: [16 32]
FRAME_BITS: [16 64]
CHANNELS: [1 2]
RATE: (0 4294967295]
PERIOD_TIME: (0 4294967295]
PERIOD_SIZE: [32 32768]
PERIOD_BYTES: [256 65536]
PERIODS: [1 8]
BUFFER_TIME: (0 4294967295]
BUFFER_SIZE: [32 65536]
BUFFER_BYTES: [256 131072]
```

```
TICK_TIME: ALL
```

```
-----  
aplay: set_params:1352: Sample format non available
```

Available formats:

- S16\_LE

- S24\_LE

- S32\_LE

用户接入耳机后，可以使用如下命令测试音频的输出功能：

```
# speaker-test -c 2 -t wav
```

```
speaker-test 1.2.6
```

```
Playback device is default
```

```
Stream parameters are 48000Hz, S16_LE, 2 channels
```

```
Using 16 octaves of pink noise
```

```
Rate set to 48000Hz (requested 48000Hz)
```

```
Buffer size range from 64 to 32768
```

```
Period size range from 64 to 16384
```

```
Using max buffer size 32768
```

```
Periods = 4
```

```
was set period_size = 8192
```

```
was set buffer_size = 32768
```

```
0 - Front Left
```

```
1 - Front Right
```

使用如下命令录制一个 5 秒的音频：

```
# arecord -f S16_LE -r 48000 -d 5 test.wav
```

```
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono
```

**注意:** 对于 Debian 系统, 由于音频接口使用 pipewire/pulseaudio 管理, 可能出现播放时资源被占用问题。需要在测试之前手动关闭相关服务。

```
# systemctl --user stop pipewire # debian bookworm  
# systemctl --user stop pulseaudio # debian bullseye
```

测试完毕后替换上面的 stop 为 restart 重新使能该功能即可。

## 4.17.LED

Linux 系统提供了一个独立的子系统以方便从用户空间操作 LED 设备, 该子系统以文件的形式为 LED 设备提供操作接口。这些接口位于 /sys/class/leds 目录下。在硬件资源列表中, 我们已经列出了单板机上所有的 LED。下面通过命令读写 sysfs 的方式对 LED 进行测试。

下述命令均为通用命令, 也是操控 LED 的通用方法。

# 查看当前系统配置的所有 LED

```
ls /sys/class/leds/
```

# 查看心跳灯当前状态

```
cat /sys/class/leds/run/brightness
```

# 配置心跳灯可控

```
echo none > /sys/class/leds/run/trigger
```

# 控制心跳灯亮灭变化

```
echo 1 > /sys/class/leds/run/brightness
```

```
echo 0 > /sys/class/leds/run/brightness
```

# 查看心跳灯支持根据什么外设的工作状态呼吸闪烁

```
cat /sys/class/leds/run/trigger
```

# 控制心跳灯按照 mmc0 工作状态呼吸闪烁

```
echo mmc0 > /sys/class/leds/run/trigger
```

# 测试 LED 是否按照 mmc0 工作状态闪烁。使用下面的方式当写入数据时 LED 会常亮

一小段时间，直到写入完成。

```
dd if=/dev/zero of=test.bin bs=1M count=10 conv=fsync
```

## 4.18.按键

板上共有三个按键：RESET，FEL 和 POWER，分别对应复位功能，烧录功能和开关机功能。

### 4.18.1. 复位

当按下 RESET 按键，系统会立刻重启。

```
# memtester 100M 1

memtester version 4.5.0 (64-bit)

Copyright (C) 2001-2020 Charles Cazabon.

Licensed under the GNU General Public License version 2 (only).


pagesize is 4096
pagesizemask is 0xfffffffffff000
want 100MB (104857600 bytes)
got 100MB (104857600 bytes), trying mlock ...locked.

Loop 1/1:

  Stuck Address      : ok

  Random Value       : [[80]HELLO! BOOT0 is starting!

[83]BOOT0 commit : ad8b4b44ac

[87]periph0 has been enabled

[90]set pll end

[92]PL gpio voltage : 3.3V

[96][pmu]: bus read error

[99]PMU: AXP2202

[104]PMU: AXP1530
```

```
[109]power mode:33, sys_vol:920  
  
[114]vaild para:1 select dram para0  
  
[117]board init ok  
  
[120]rtc[3] value = 0xb00f  
  
[123]rtc[7] value = 0x1  
  
[125]enable_jtag
```

从上面的 log 信息中可以发现, 系统还在执行 memtester, 但是立刻进入了 boot0 信息打印。此时就是笔者按下 RESET 按键的时机。

### 4.18.2. 电源

电源按键是用于当系统关机后, 开机使用, 或长按强制关机。当关机后, 系统无法通过 RESET 复位启动, 必须使用 POWER 按键启动。

```
Requesting system poweroff  
  
[ 146.085755] reboot: Power down  
  
[80]HELLO! BOOT0 is starting!  
  
[83]BOOT0 commit : ad8b4b44ac  
  
[87]periph0 has been enabled  
  
[90]set pll end  
  
[92]PL gpio voltage : 3.3V  
  
[96][pmu]: bus read error  
  
[99]PMU: AXP2202  
  
[104]PMU: AXP1530  
  
[109]power mode:33, sys_vol:920  
  
[114]vaild para:1 select dram para0  
  
[117]board init ok  
  
[120]rtc[3] value = 0xa300  
  
[122]rtc[7] value = 0x2  
  
[125]enable_jtag
```

上面的 log 信息就是一次系统关机到按下 POWER 按键重新开机的过程。

### 4.18.3. FEL

参考上文中的快速开始章节内容。

## 4.19. 风扇

该节仅限于 Pi 板支持，单板机无相应器件，但用户可自行参考 Pi 板进行移植，需要注意系统默认使用的 **default governor**，单板机默认是 **power\_allocator**，这种情况风扇会以最高转速等级运行。

```
cd $(dirname $(grep -rn fan /sys/class/thermal/cooling_device*/type | cut -d ":" -f 1))

cat cur_state      # 读取当前风扇转速等级

cat max_state      # 查看最高支持的风扇转速等级

echo 4 > cur_state  # 设置为最高风扇转速
```

## 4.20. GPIO

通过如下命令控制一个 IO，这里以 PK23 为例：

```
# mount -t debugfs debugfs /sys/kernel/debug/

# grep PK23 /sys/kernel/debug/pinctrl/2000000.pinctrl/pinmux-pins

pin 343 (PK23): UNCLAIMED

# cd /sys/class/gpio/

# echo 343 > export

# echo out > gpio343/direction

# echo 1 > gpio343/value
```

此时通过万用表查看电压并没有按照实际的情况发生变化，这是由于 PK 组引脚与 sensor 模块的冲突导致，重启后长按 `s` 进入 uboot 的命令行中执行如下命令关闭 sensor 功能即可：

```
=> fdt set /soc/vind status disabled

=> fdt set /soc/pd-vi-test status disabled
```

上面的命令等效于在设备树中关闭 vind0 和 pd-vi-test 节点。用户可参考《软件开发指

南》修改设备树实现同时支持摄像头与 PK 组其他引脚使用。

## 5. 参考资料

- ❖Linux kernel 开源社区：<https://www.kernel.org/>
- ❖全志开发社区：<https://bbs.aw-ol.com/>
- ❖Buildroot-manual：<https://buildroot.org/downloads/manual/manual.html>
- ❖Debian wiki：<https://wiki.debian.org/>

## 6. 修订说明

修订说明表

| 版本   | 修改内容        | 修改时间       | 编制  | 校对  | 审批  |
|------|-------------|------------|-----|-----|-----|
| V1.0 | 初稿          | 2025.03.19 | WYQ | HSL | WFX |
| V1.1 | 添加 Pi 板使用说明 | 2025.7.31  | WYQ | HSL | WFX |
|      |             |            |     |     |     |

## 7. 关于我们



销售热线：4000-330-990

技术支持：[support@cdebyte.com](mailto:support@cdebyte.com) 官方网站：<https://www.ebyte.com>

公司地址：四川省成都市高新西区西区大道 199 号 B5 栋

((( )))<sup>®</sup>  
**EBYTE 成都亿佰特电子科技有限公司**  
Chengdu Ebyte Electronic Technology Co.,Ltd.