EAIDK-310 用户使用手册

2019/7/29

OPEN AI LAB

版本 v0.9.0

变更记录
(Reversion Record)

日期 (Date)	版本 (Rev)	说明 (Change Description)	作者 (Author)
2019.6.19	0.2.0	初稿	路明
2019.6.26	0.3.0	根据内部review意见修改	路明
2019.7.23	0.8.0	根据综合意见修改	路明
2019.7.29	0.9.0	格式调整，更新图片	路明
2019.8.1	0.9.1	更新产品名称	路明

目录(catalog)

1 前言 4

1.1 目的 4
1.2 术语 4

2 EAIDK-310 概述 5

2.1 硬件规格 6
2.2 软件规格 6

3 硬件介绍 8

3.1 硬件总览 8
3.2 调试接口 9
3.3 电源模块 10
3.4 存储模块 11
3.4.1 内存 11
3.4.2 EMMC 11
3.4.3 TF 卡 12
3.5 HDMI 显示接口 12
3.6 音频模块 13
3.7 USB 模块 13
3.8 网络通讯 14
3.8.1 以太网 14
3.8.2 WIFI/BT 14
3.9 扩展 IO 接口 15

4 人工智能开发平台 AID 16

4.1 嵌入式深度学习框架 TENGINE 16
4.2 嵌入式计算机视觉库 BLADECV 16
4.3 异构计算库 HCL 17

5 软件及开发 18

5.1 系统登陆 18
5.2 软件下载 18
5.3 环境设置 18
5.3.1 添加源（默认已经配置） 18
5.3.2 安装 RPM 包（默认已经安装） 18
5.4 串口调试 19

5.4.1 Windows 平台（以 SecureCRT 为例） 19
5.4.2 Linux 平台（以 Minicom 为例） 20

6 附录 固件烧写 22

6.1 WINDOWS 主机烧写 22
6.2 LINUX 主机烧写 24

1 前言

1.1 目的

本文档主要介绍 EAIDK-310 产品基本功能，硬件特性，软件特点及软件调试操作方法，旨在帮助开发人员更快、更好熟悉 EAIDK-310 硬件平台（EAIDK-310）和使用 EAIDK-310 开发套件。

1.2 术语

- **EAIDK**：Embedded AI Development Kit。嵌入式人工智能开发套件。
- **AID**：AID 是 OPEN AI LAB 开发的一个面向嵌入式平台前端智能，跨 SoC 的 AI 核心软件平台。
- **BladeCV**：BladeCV 是 OPEN AI LAB 开发的，在嵌入式平台上替代 OpenCV 的计算机视觉开发包，包含计算机视觉算法、图像获取和图形界面三部分。

2 EAIDK-310 概述

EAIDK（Embedded AI Development Kit），是以 Arm SoC 为硬件平台、Tengine（Arm 中国周易平台）为核心的人工智能基础软件平台 AID、集成典型应用算法，所形成的 **“软硬一体化”的 AI 开发套件；**是专为 AI 开发者精心打造，面向边缘计算的人工智能开发套件。

硬件平台具备语音、视觉等传感器数据采集能力，及适用于多场景的运动控制接口。

软件平台支持视觉处理与分析、语音识别、语义分析、SLAM 等应用和主流开源算法，满足 AI 教育、算法应用开发、产品原型开发验证等需求。

EAIDK-310 是 EAIDK 产品系列中第二款套件，主芯片采用具备主流性能 Arm SoC 的 RK3228H，搭载 OPEN AI LAB 嵌入式 AI 开发平台 AID（包含支持异构计算库 HCL、嵌入式深度学习框架 Tengine、以及轻量级嵌入式计算机视觉加速库 BladeCV）。为 AI 应用提供简洁、高效、统一的 API 接口，加速终端 AI 产品的场景化应用落地。

2.1 硬件规格

表 2-1 EAIDK-310 硬件规格

SoC	RK3228H
CPU	ARM 4 核 64 位处理器 四核 Cortex-A53，最高 1.3GHz
GPU	ARM Mali-450 MP2 GPU 支持 OpenGL ES 1.1/2.0、OpenVG 1.1
运行内存	LPDDR3 1GB
内置存储	8GB 高速 emmc
扩展存储	MicroSD，最大支持 128GB
有线网络	RJ45，10/100M 自适应
WIFI	802.11 ac/a/b/g/n, 2.4G/5GHz
蓝牙	Bluetooth 5.0
USB	1xUSB3, 3xUSB2, 1xMicro-USB
HDMI	2.0, 1xType-A, 最高 4Kx2K@60Hz
调试接口	UART（TTL 电平）
其他接口	2xI2C 1xSPI 9xGPIO 2xGPIO（仅输出） 1xSpeaker
电源	Micro USB 5V / 2A

2.2 软件规格

表 2-2 EAIDK-310 软件规格

操作系统	Linux	Fedora 28
	Android	Rockchip 官方 Android8.1
嵌入式 AI 开发平台 AID	嵌入式深度学习框架 Tengine	针对 ARM CPU 及 ARM Mali GPU 优化 支持 Caffe/TensorFlow/MXnet/ONNX 模型文件 兼容 Caffe/TensorFlow API

		以插件方式支持底层算子扩展 支持 INT8 量化
	嵌入式计算机视觉加速库 BladeCV	具备常用的图像处理、计算机视觉、模式识别的算子/算法 支持常见格式的图像和视频文件读写，常见编码视频码流的解码 支持图形化界面的图像/视频显示和信息叠加 针对 ARM neon 指令及 GPU 优化 支持基于 RK3228H 的硬件视频解码及简单图像操作的硬件加速
	异构计算库 HCL	支持异构计算的嵌入式 NN 计算库
视频编解码 API	硬解码	H264/H265：4K@60fps
	硬编码	H264/H265：1080p@30fps

3 硬件介绍

3.1 硬件总览

EAIDK-310 正面如图 3-1 所示，背面如图 3-2 所示。

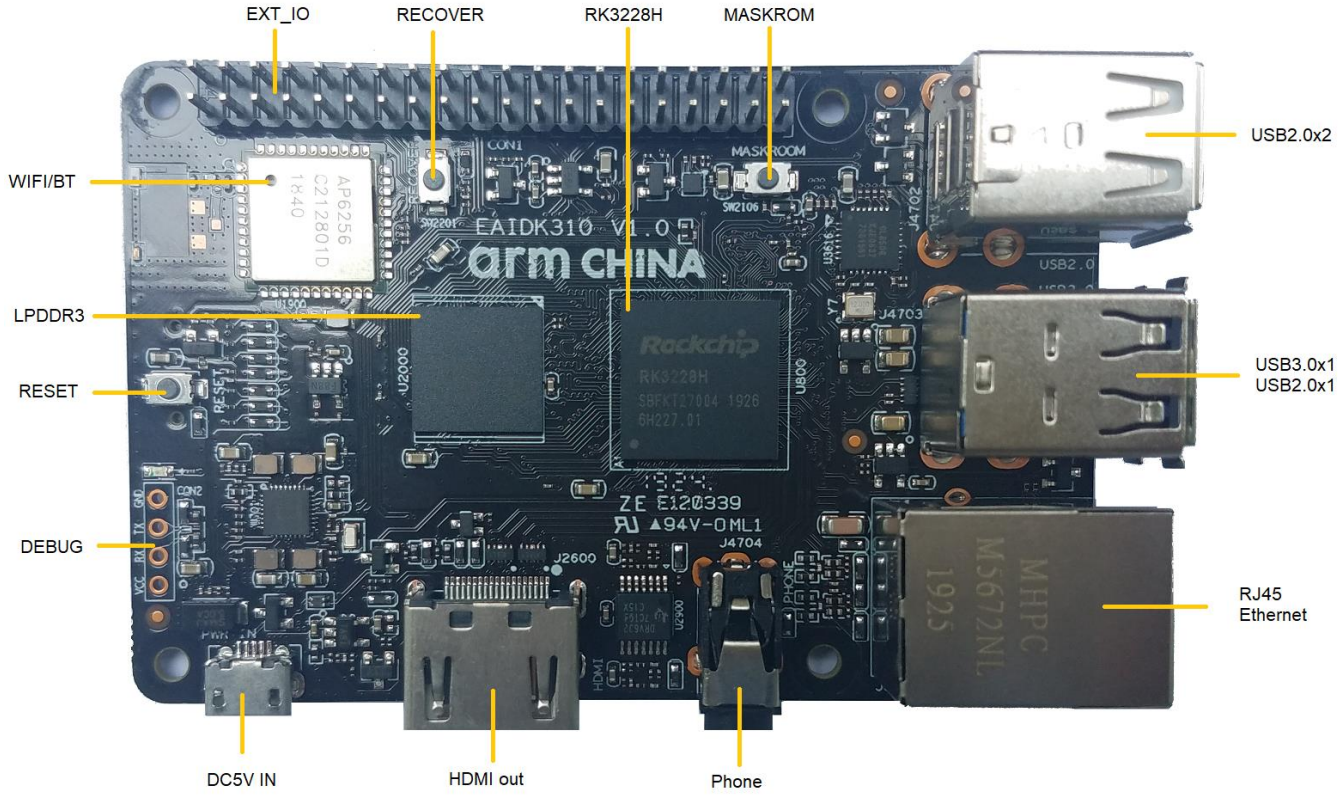

图 3-1 Top Layer 接口布局图

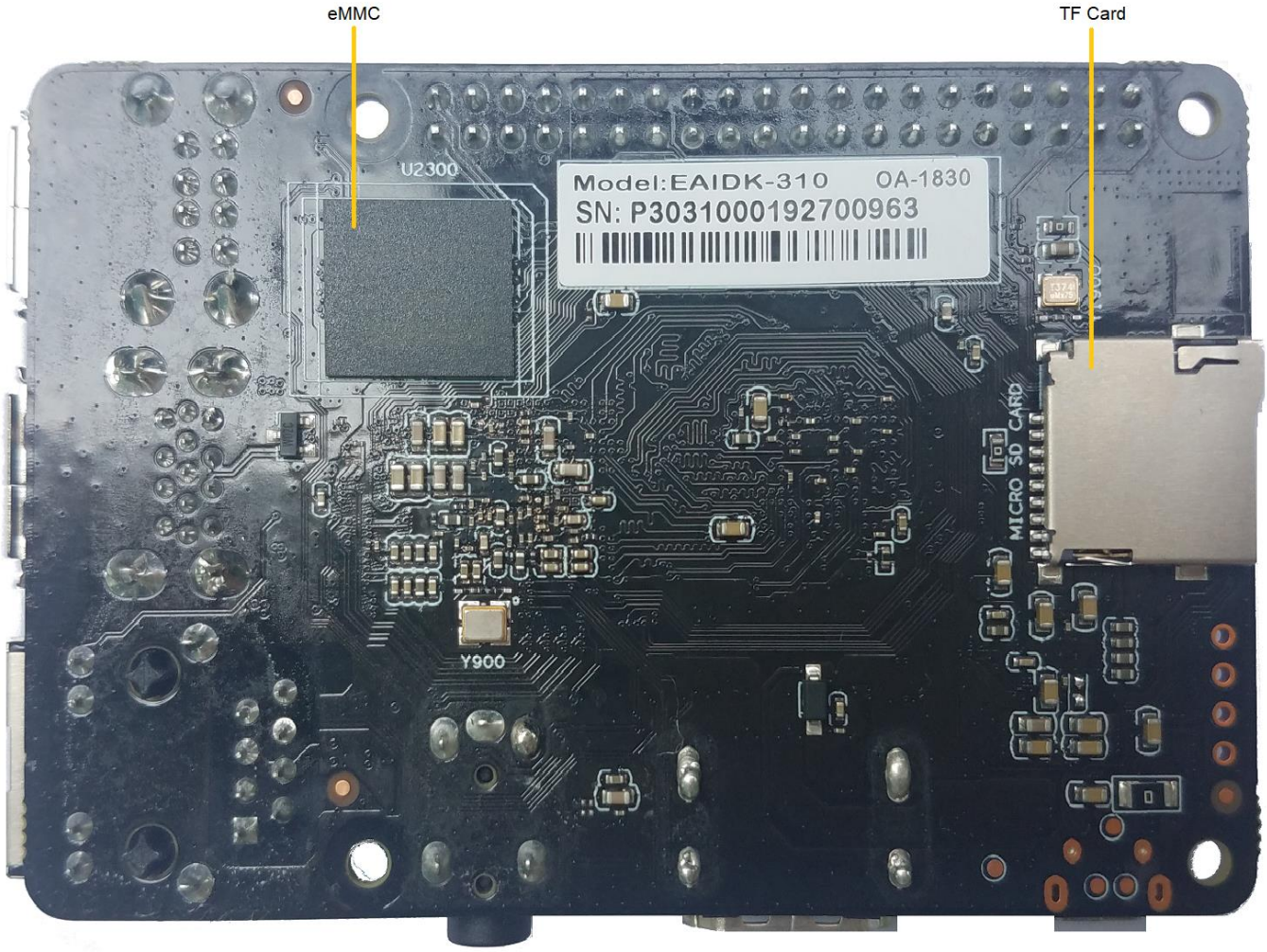

图 3-2 Bottom Layer 接口布局图

3.2 调试接口

开发板提供调试串口供开发调试使用。调试串口连接主控的 UART2 接口，通过板上提供的 TTL 电平接口与外部相连接。

说明：RK3228H 的调试串口的波特率为 1500000。

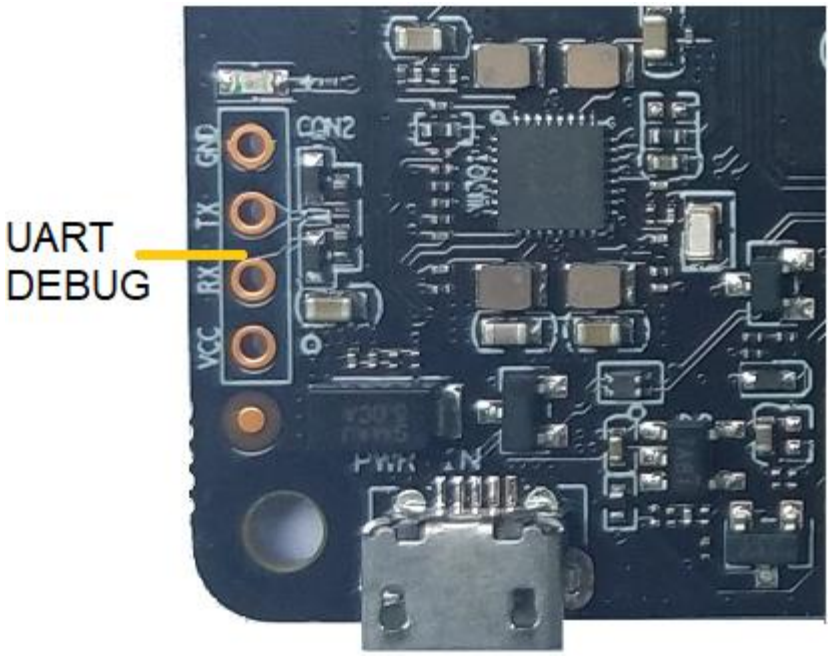

图 3-3 Uart Debug 接口示意图

3.3 电源模块

EAIDK-310 开发板的电源模块采用 PMIC RK805 为核心芯片，配合外围的 Buck、LDO 组成。

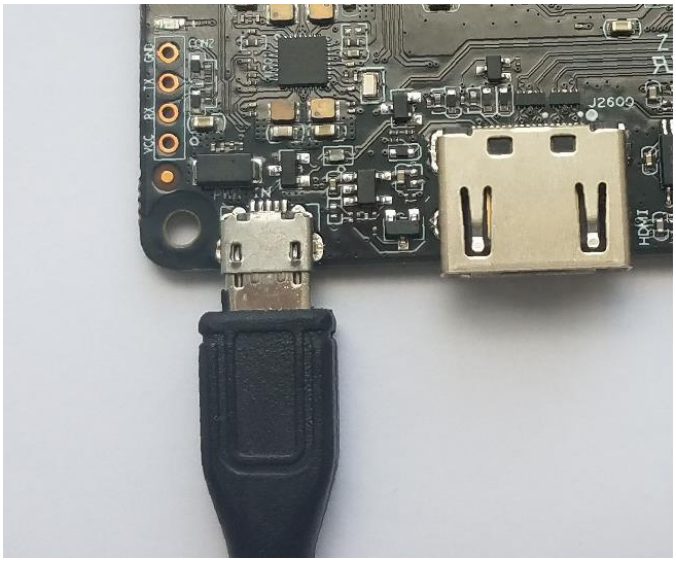

图 3-4 DC 输入接口示意图

3.4 存储模块

3.4.1 内存

EAIDK-310 开发板采用 1 颗 1GB LPDDR3 颗粒。

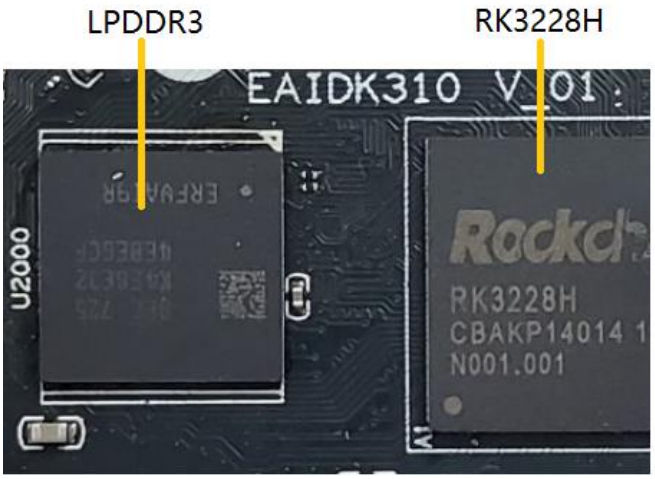

图 3-5 LPDDR3 位置示意图

3.4.2 EMMC

EAIDK-310 开发板采用 EMMC 作为系统盘，默认容量 8GB。

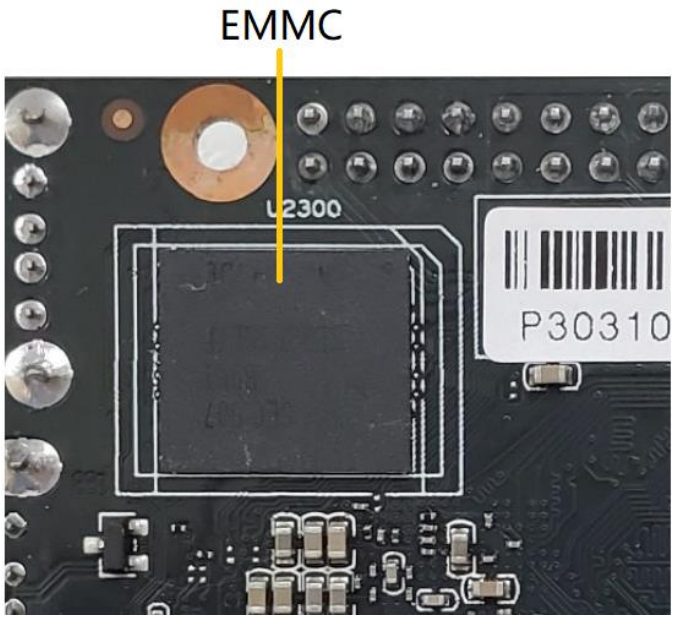

图 3-6 EMMC 位置示意图

3.4.3 TF 卡

EAIDK-310 开发板带有 TF Card 卡座，连接 RK3228H SDMMC0。数据总线宽带为 4bit，支持热插拔。

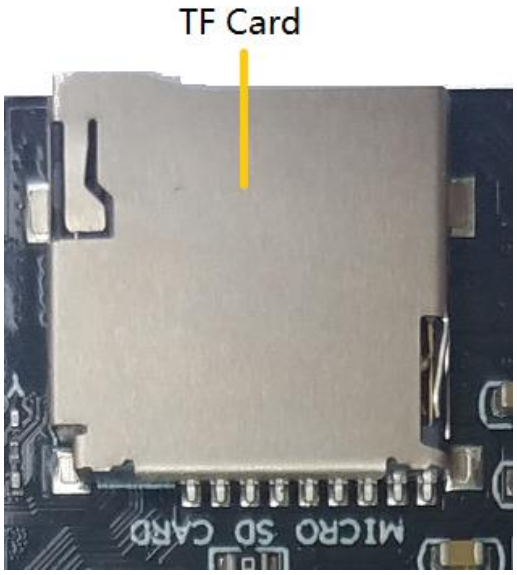

图 3-7 TFcard 位置示意图

3.5 HDMI 显示接口

EAIDK-310 开发板支持 HDMI 显示，采用 A 型接口。

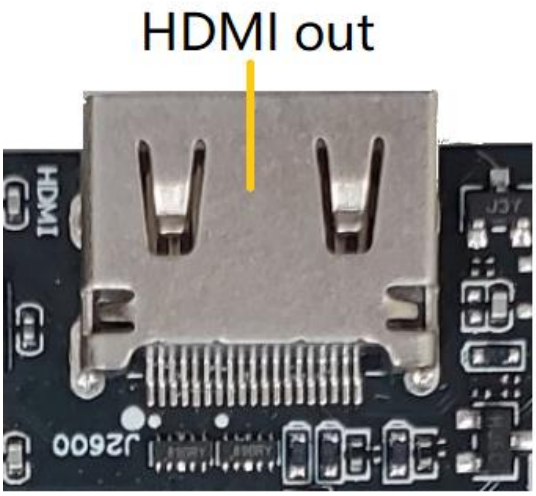

图 3-8 HDMI 位置示意图

3.6 音频模块

EAIDK-310 开发板支持立体声耳机无电容耦合输出及 HDMI 数字音频输出，并通过扩展接口插针可直连 Speaker 输出。

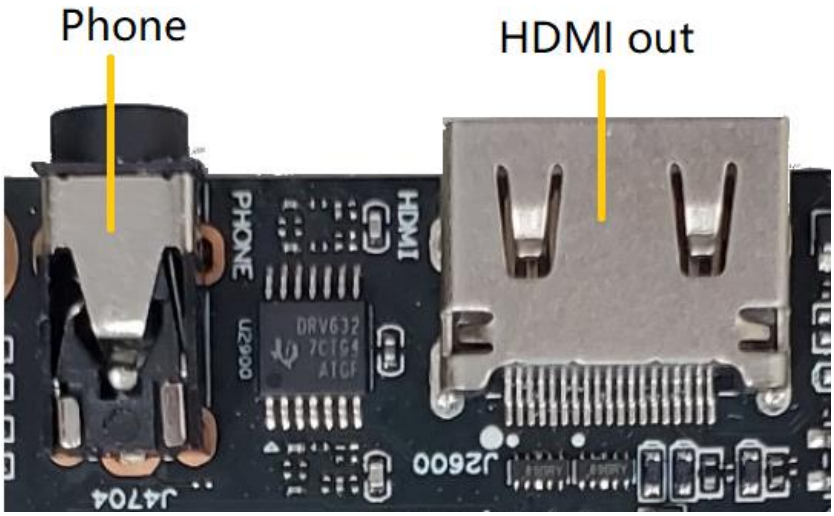

图 3-9 音频接口位置示意图

3.7 USB 模块

EAIDK-310 开发板集成 3 路 USB2.0 Host 和 1 路 USB3.0 Host。外接 USB 鼠标、键盘和 U 盘等多种的人机交互方式。

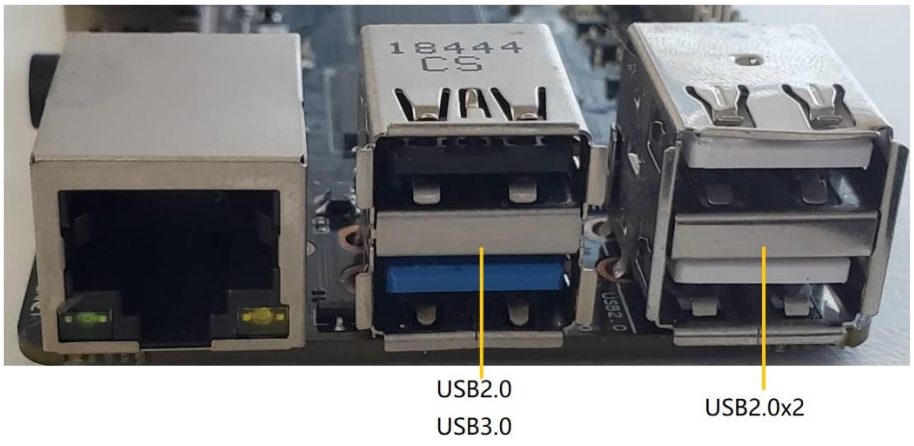

图 3-10 USB 位置示意图

3.8 网络通讯

3.8.1 以太网

EAIDK-310 开发板支持 RJ45 接口，可提供百兆快速以太网连接功能，选用内置 PHY 其特性如下：

- 兼容 IEEE802.3 标准，支持全双工和半双工操作，支持交叉检测和自适应
- 支持 10/100 数据速率。
- 接口采用具有指示灯和隔离变压器的 RJ45 接口。

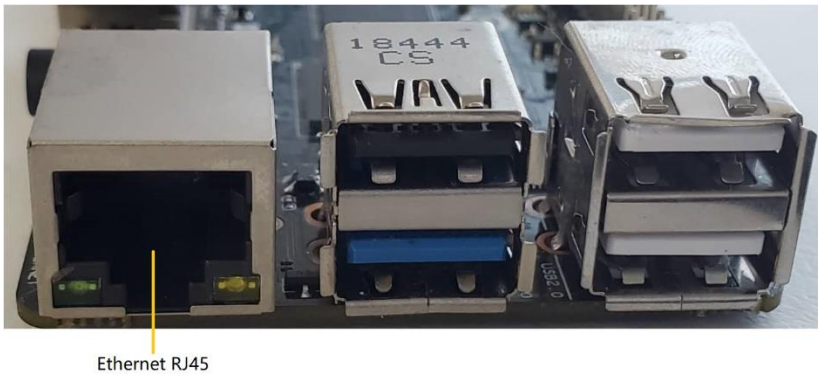

图 3-11 RJ45 位置示意图

3.8.2 WIFI/BT

开发板上 WIFI+BT 模组的特性如下：

- 支持 WIFI 2.4G 和 5G，IEEE802.11a/b/g/n/ac，采用 4bits SDIO 通讯。
- 支持 BT5.0 功能，采用 UART 通讯，具备 PCM 接口，可语音通话。

图 3-12 WIFI/BT 模组示意图

3.9 扩展 IO 接口

EAIDK-310 集成 40 Pins IO 扩展接口，包含一组 UART，一组 SPI，两组 I2C，一组扬声器输出，9 个 GPIO，2 个仅可用于输出的 GPIO。

图 3-13 扩展 IO 示意图

扩展 IO 接口的管脚定义如表 3-3

表 3-3 GPIO_EXT 管脚定义表

Pin	Name	Pin	Name
1	VCC_IO	**2**	VCC_SYS
3	I2C0_SDA	**4**	VCC_SYS
5	I2C0_SCL	**6**	GND
7	GPIO2_B7	**8**	UART1_TX
9	GND	**10**	UART1_RX
11	GPIO0_D3	**12**	GPIO2_A2
13	GPIO2_C2	**14**	GND
15	GPIO2_C1	**16**	GPIO3_A5
17	VCC_IO	**18**	GPIO2_C3
19	SPI_TXD_M2	**20**	GND
21	SPI_RXD_M2	**22**	GPIO3_A7
23	SPI_CLK_M2	**24**	SPI_CS0N_M2
25	GND	**26**	SPI_CS1N_M0
27	I2C1_SDA_PMIC	**28**	I2C1_SCL_PMIC
29		**30**	GND
31		**32**	
33		**34**	GND
35		**36**	GPIO2_C4
37	RK805_GPIO2	**38**	RK805_GPIO1
39	SPEAK_N	**40**	SPEAK_P

4 人工智能开发平台 AID

4.1 嵌入式深度学习框架 Tengine

Tengine 是由 OPEN AI LAB 开发的一款轻量级模块化的高性能神经网络推理引擎，针对 Arm 嵌入式设备进行优化。

Tengine 支持各类常见卷积神经网络，包括 SqueezeNet，MobileNet，AlexNet，ResNet 等，支持层融合、8 位量化等优化策略。并且通过调用针对不同 CPU 微构架优化的 HCL 库，将 Arm CPU 的性能充分挖掘出来。除了对单核性能的挖掘，Tengine 在多线程加速上也针对 Arm SoC 系统构架做了专门的优化，并且通过优化调度，双线程加速比达到 170%，四线程加速比达到 270%。为了满足嵌入式应用场景对低功耗、小内存的需求，Tengine 在内存也做了深度优化，轻松运行极度消耗内存的 VGG16 网络，并且通过增加缓存利用率减少不必要的内存访问，从而降低功耗。

Tengine 以 wrapper 的形式支持 Caffe API，原本基于 caffe 开发的应用只需要简单的重新编译就能在 Tengine 上二次部署，后续 Tengine 还将支持 Tensorflow API。

作为一款开放计算平台，Tengine 具备强大的扩展能力，除了支持自带的 HCL 库以外，还支持业界常见的 Openblas, Arm Compute Library 库。同时，开发者可以很容易拓展自己的算子库，提升 Tengine 适应力。Tengine 的扩展性还表现在适配芯片上 GPU 等各类硬件资源甚至是 DLA 硬件加速器。只需要简单的在 Tengine 上注册设备并挂载上驱动，就能充分利用芯片上的硬件资源提升计算能力。

4.2 嵌入式计算机视觉库 BladeCV

BladeCV 是由 OPEN AI LAB 开发的面向嵌入式平台的一款轻量级计算机视觉库，针对 Arm 的 CPU 和 GPU 进行深度优化。BladeCV 在嵌入式平台可用于完成 OpenCV 在 Intel 平台相应的功能，用于算法的开发、调试等任务。其有如下特点：

1. 具备常用的图像处理、计算机视觉、模式识别的算子/算法/接口。
2. 支持常见格式的图像和视频文件读写，能对常见编码格式的视频码流进行解码，也能对未压缩的图像序列进行编码。
3. 支持图形化界面的图像/视频显示和信息叠加。

4. 针对 ARM neon 指令及 GPU 优化。
5. 在一些芯片平台上支持硬件视频编解码及图像操作的硬件加速。EAIDK-310 的版本能够充分使用 RK3228H 的硬件编解码能力，也能利用 RGA 功能进行图像的缩放，旋转，截取，格式转换等硬件加速。

4.3 异构计算库 HCL

HCL 是由 OPEN AI LAB 研发的异构计算库，专用于加速嵌入式平台神经网络计算，HCL 量化计算库也是 Tengine 的高性能计算插件。其重要特点有：

(1) 加速最耗时算子，手工调优汇编函数，针对 CPU 微构架做极致优化

(2) 支持多线程加速，采用 Dataflow 计算构架，双线程加速比 175%，四线程加速比达 300%

(3) 支持多精度数据格式计算

HCL 量化计算库基于 Arm CPU 实现，使用 8 位整型（INT8）量化数据大幅提高计算速度，同时显著降低内存占用和系统内存带宽消耗。相比于标准 32 位浮点型（FP32）数据计算性能提升 50~90%。HCL 采用智能动态量化技术，在不需要 retraining 的前提下，实现推理计算精度基本不变。

HCL 已经适配主流 Arm CPU Cortex-A7/A17/A53/A72/A73。

5 软件及开发

5.1 系统登陆

EAIDK-310 预装 Fedora 28 及轻量级桌面系统 LXDE。缺省登录账号为 openailab，密码为 openailab

5.2 软件下载

EAIDK-310 的固件和源码都可以从 EAIDK 的官方网站获取，其地址为 http://www.eaidk.com

5.3 环境设置

5.3.1 添加源（默认已经配置）

1. 安装 RK3228H 硬件相关的系统库 RPM 包的源：

```
sudo yum -y localinstall --nogpgcheck  http://www.eaidk.net/rockchip/rockchip-repo-1.0-1.fc28.aarch64.rpm
```

2. 安装 Openailab 的系统和应用 RPM 包的源：

```
sudo yum -y localinstall --nogpgcheck http://www.eaidk.net/openailab/openailab-repo-1.0-1.fc28.aarch64.rpm
```

3. 安装第三方非开源的 RPM 包的源（可选）：

```
sudo yum localinstall --nogpgcheck http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-28.noarch.rpm

sudo yum localinstall --nogpgcheck http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-28.noarch.rpm
```

5.3.2 安装 RPM 包（默认已经安装）

1. 安装编译工具：
   sudo dnf -y install gcc gcc-c++ cmake automake

1. 安装 openssl(开发板上编译内核需要):
   sudo dnf -y install openssl-devel

5.4 串口调试

5.4.1 Windows 平台（以 SecureCRT 为例）

1. 将 EAIDK-310 开发板的 Debug 口与主机用 USB 转 4 针串口线连接，打开设备管理器获取 USB Serial Port 的端口号，如下图所示:

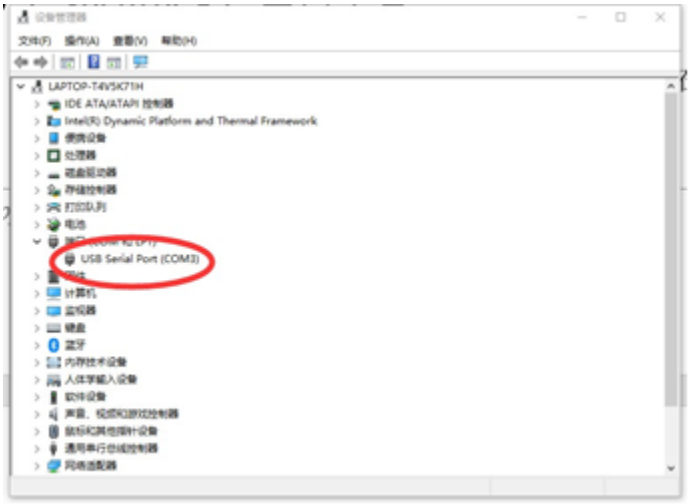

图 6-1 获取当前端口 COM 号

注意：如果设备管理器里面显示驱动异常信息，请选择更新驱动信息即可。

2. 打开串口工具“SecureCRT”，点击“快速连接”按钮

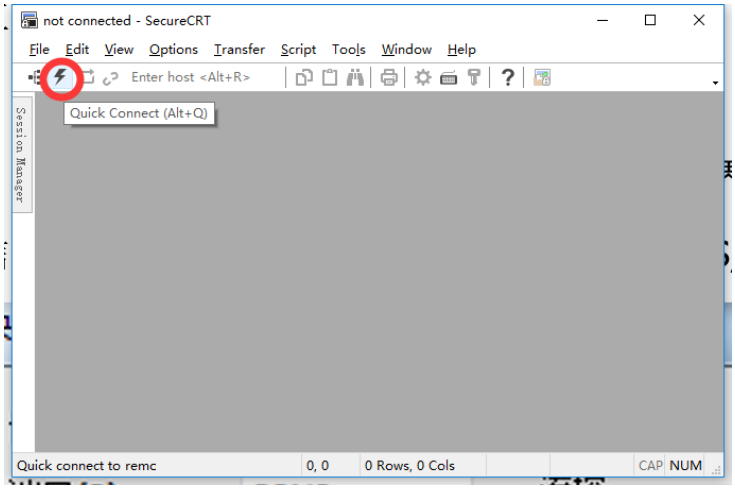

图 6-2 串口工具 SecureCRT 界面

3. 配置串口信息，端口选择连接开发板的端口号，设置波特率为 1500000，不勾选流控 RTS/CTS， 如下图所示：

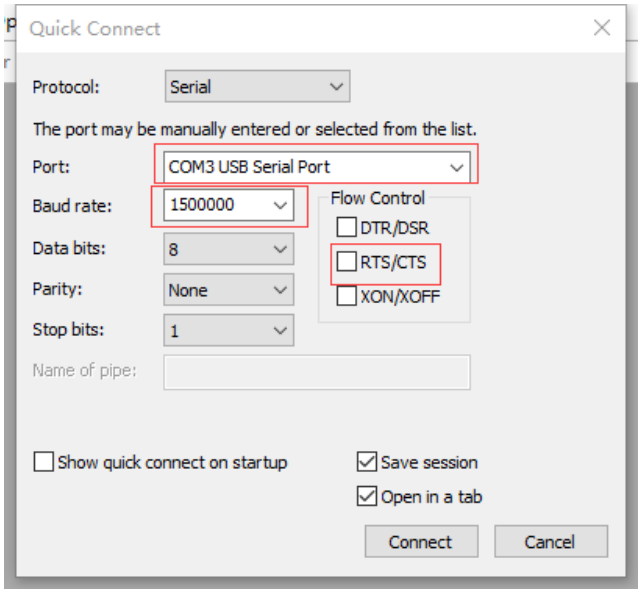

图 6-3 配置串口信息

4. 点击连接，就可以正常查看系统调试信息和输入用户命令。

5.4.2 Linux 平台（以 Minicom 为例）

1. 将 EAIDK-310 开发板的 Debug 口与主机用 USB 转 4 针串口线连接。
2. 以 root 权限打开 minicom：sudo minicom -s
3. 打开 Minicom 菜单：输入 CTRL-A + z。
4. 进入 Minicom 配置界面：输入“O”选择“cOnfigure Minicom”。
5. 进入串口设置：选择“Serial port setup”。
6. 设置串口设备：输入“A”，写入“/dev/ttyUSB0”，按回车确定。
7. 禁止流控：输入“F”，按回车确定。
8. 设置波特率：输入“E”，再输入“A”直到显示“Current 1500000 8N1”，然后回车确定。
9. 保存设置：选择“Save setup as dfl”。

10. 退出设置：选择“Exit”。

6 附录 固件烧写

6.1 Windows 主机烧写

烧写前需要安装 Windows PC 端 USB 驱动：

双击 Tools\Windows\DriverAssitant_v4.5\ DriverInstall.exe 打开安装程序，点击“驱动安装”按提示安装驱动即可，安装界面如下所示：

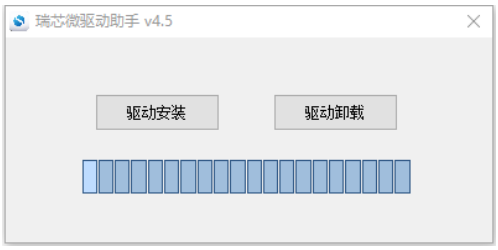

图 6-1 安装界面图

驱动安装完成后，固件烧写步骤如下：

1. Micro-USB 线连接主机端的 USB 接口和 EAIDK-310 开发板的 Micro-USB 接口。
2. 双击烧录工具 AndroidTool.exe 打开程序。
3. 按鼠标右键选择导入配置，文件，如下图所示：

图 6-2 导入配置

4. 在烧录工具的文件夹下选择相关的配置文件（注：烧录 Linux 镜像选择 config_linux,烧录安卓镜像选择 config_android_baseparameter），如下图所示：

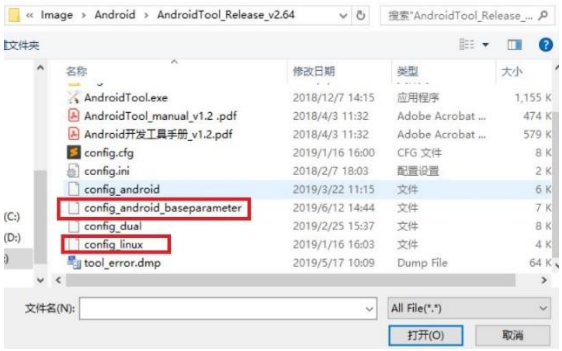

图 6-3 配置文件选择

5. 长按 EAIDK-310 开发板上 RECOVER 按键的同时短按 RESET 键，直到系统进入 Loader 模式，FlashTool 显示如下所示：

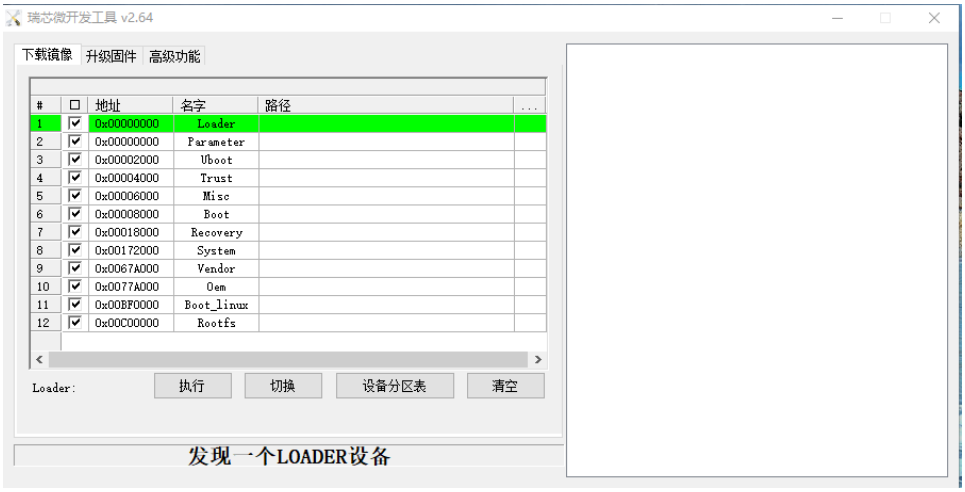

图 6-4 AndroidTool 显示界面图

注意：如果 uboot 启动异常而无法进入 Loader 模式，需要长按 Maskrom 按键的同时短按 RESET 键让系统进入 Maskrom 模式;AndroidTool 提示：Found one Maskrom Device!

6. 选择相关的镜像文件点击执行按钮，如下图所示(左图为 Linux,右图为 Andorid)：

图 6-5 固件烧写

7. 烧录成功如下图所示(左图为 Linux，右图为 Android)：

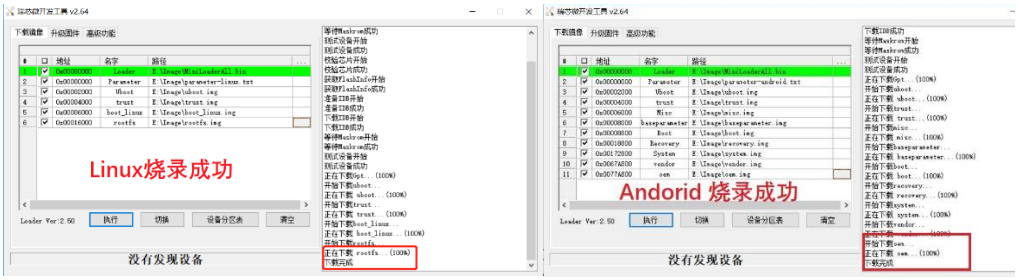

图 6-6 烧写成功

6.2 Linux 主机烧写

1. Micro-USB 线连接主机端的 USB 接口和 EAIDK-310 开发板的 Micro-USB 接口。
   说明：EAIDK-310 的 fedora28 系统也可以作为开发主机给其他开发板烧写固件。
2. 长按 EAIDK-310 开发板上 RECOVER 按键后重启机器，进入 Loader 模式。
3. 进入 flashTool 目录：cd flashTool
4. 将固件拷贝到 flashTools/images 目录。
5. 执行命令烧写固件：
   sudo python flash.py
   1) 烧写 Android 固件：

sudo python flash.py -a all:	烧写所有 Android 固件
sudo python flash.py -a boot:	烧写 Android 的 boot.img
sudo python flash.py -a uboot:	烧写 Android 的 trust.img 和 uboot.img。

sudo python flash.py -a system:	烧写 Android 的 system.img

2) 烧写 Linux 固件：

sudo python flash.py -l all：	烧写所有 Linux 固件
sudo python flash.py -l boot:	烧写 Linux 的 boot_linux.img
sudo python flash.py -l uboot：	烧写 Linux 的 trust.img 和 uboot.img。
sudo python flash.py -l rootfs：	烧写 fedora 根文件系统。

3) 烧写双系统固件：

sudo python flash.py -d all:	烧写所有双系统固件
sudo python flash.py -d boot:	烧写 boot.img 和 boot_linux.img
sudo python flash.py -a uboot:	烧写 trust.img 和 uboot.img。
sudo python flash.py -a system:	烧写 system.img
sudo python flash.py -a rootfs：	烧写 fedora 根文件系统。