



**ED-AIC2020**

**人工智能相机**

**用户手册**

**上海晶珩电子科技有限公司**

**2023年5月**

# 联系我们

非常感谢您购买和使用我们公司的产品，我们将竭诚为您提供服务。

我们是 Raspberry Pi 的全球设计合作伙伴之一，致力于提供基于 Raspberry Pi 技术平台的物联网、工业控制、自动化、绿色能源和人工智能的硬件解决方案。

您可以通过以下方式联系我们：

上海晶珩电子科技有限公司

EDA Technology Co.,LTD

地址：上海市嘉定区嘉罗公路 1661 号 24 栋 301 室

邮箱：[sales@edatec.cn](mailto:sales@edatec.cn)

手机：+86-18627838895

网站：<https://www.edatec.cn>

**技术支持：**

邮箱：[support@edatec.cn](mailto:support@edatec.cn)

微信：zzw\_1998-

# 版权声明

ED-AIC2020 及其相关知识产权为上海晶珩电子科技有限公司所有。

上海晶珩电子科技有限公司拥有本文件的版权并保留所有权利。未经上海晶珩电子科技有限公司的书面许可，不得以任何方式和形式修改、分发或复制本文件的任何部分。

# 免责声明

上海晶珩电子科技有限公司不保证本手册中的信息是最新的、正确的、完整的或高质量的。上海晶珩电子科技有限公司也不对这些信息的进一步使用作出保证。如果由于使用或不使用本手册中的信息，或由于使用错误或不完整的信息而造成的物质或非物质相关损失，只要没有证明是上海晶珩电子科技有限公司的故意或过失，就可以免除对上海晶珩电子科技有限公司的责任索赔。上海晶珩电子科技有限公司明确保留对本手册的内容或部分内容进行修改或补充的权利，无需特别通知。

# 前言

## 相关手册

产品包含的各类产品文档，如下表所示，用户可以根据需要选择查看对应的文档。

文档	说明
ED-AIC2020 Datasheet	介绍 ED-AIC2020 的产品形态、软硬件规格、尺寸和订购编码，帮助用户了解产品的整体系统参数。
ED-AIC2020 用户手册	介绍 ED-AIC2020 的外观、安装、启动和配置，帮助用户更好的使用该产品

用户可访问以下网址获取更多资料：

<https://www.edatec.cn>

## 读者范围

本手册适用以下读者对象：

- ◆ 机械工程师
- ◆ 电器工程师
- ◆ 软件工程师
- ◆ 系统工程师

## 相关约定

### 符号约定

符号	说明
	提示符号，提示重要的特点或操作。
	注意符号，可能会对人身造成伤害，或给系统造成损害，或造成信号中断/丢失。
	可能会对人身造成重大伤害。

# 安全说明

- ◆ 本产品应在符合设计规格要求的环境下使用，否则可能造成故障，因未遵守相关规定引发的功能异常或部件损坏等不在产品质量保证范围之内。
- ◆ 因违规操作产品引发的人身安全事故、财产损失等，我司将不承担任何法律责任。
- ◆ 请勿私自修改设备，修改设备可能会使设备故障。
- ◆ 安装设备时，需要固定好设备，防止跌落。
- ◆ 如果设备带有天线，正常使用时，请与设备至少保持20cm的距离。
- ◆ 请勿使用液体清洁设备，应远离液体和易燃物品。
- ◆ 本产品仅支持在室内环境使用。

# 目录

联系我们 .....	ii
版权声明 .....	iii
免责声明 .....	iv
前言 .....	i
相关手册 .....	i
读者范围 .....	i
相关约定 .....	ii
符号约定 .....	ii
安全说明 .....	iii
目录 .....	iv
1 产品描述 .....	1-1
1.1 概述 .....	1-2
1.2 包装清单 .....	1-4
1.3 外观 .....	1-5
1.4 接口 .....	1-6
1.4.1 电源接口 .....	1-6
1.4.2 通信接口 .....	1-7
1.5 按键和指示灯 .....	1-8
1.6 光源和摄像机镜头 .....	1-9
2 安装设备 .....	2-1

3	启动设备.....	3-1
3.1	连接线缆.....	3-2
3.2	首次启动设备.....	3-3
4	安装操作系统.....	4-1
4.1	镜像下载.....	4-2
4.2	打开设备外壳.....	4-3
4.3	eMMC 烧录.....	4-4
4.4	关闭设备外壳.....	4-8
5	配置设备.....	5-1
5.1	查找设备 IP.....	5-2
5.1.1	登录路由器来查询.....	5-2
5.1.2	使用 nmap 工具扫描.....	5-2
5.2	安装 BSP(可选).....	5-3
5.3	远程登录.....	5-4
5.4	编译 Camera 演示.....	5-6
5.5	aic.h 文件.....	5-7
5.6	libaic.so 文件.....	5-8
5.7	Camera API.....	5-9
5.7.1	AR0234 Camera.....	5-9
5.8	IO API.....	5-10

# 1 产品描述

介绍产品的概述、包装、外观、接口、指示灯和光源等内容。

- ✓ 概述
- ✓ 包装清单
- ✓ 外观
- ✓ 接口
- ✓ 按键和指示灯
- ✓ 光源和摄像机镜头

## 1.1 概述

ED-AIC2020 是一款高度集成的工业人工智能摄像机，配有固定或液态镜头、LED 照明、工业级图像传感器和 Raspberry Pi 计算机模块 4 (CM4)。

- ◆ ED-AIC2020 配备了变焦镜头和 LED 照明，大大减少了安装和维护工作。
- ◆ ED-AIC2020 是一款一体式、紧凑耐用的工业人工智能摄像机，非常适合各种 AI 视觉应用。
- ◆ 以预装 QT、Python、OpenCV SDK 和 AI 算法演示应用为特色，可以大大加速 AI 视觉应用的开发和部署。



ED-AIC2020 主要应用于以下场景：

- ◆ 固定式工业条形码扫描仪
- ◆ 物体位置
- ◆ 测量
- ◆ 光学字符识别
- ◆ 基于人工智能的对象检测
- ◆ 基于人工智能的对象分割
- ◆ 基于人工智能的缺陷检测



## 1.2 包装清单

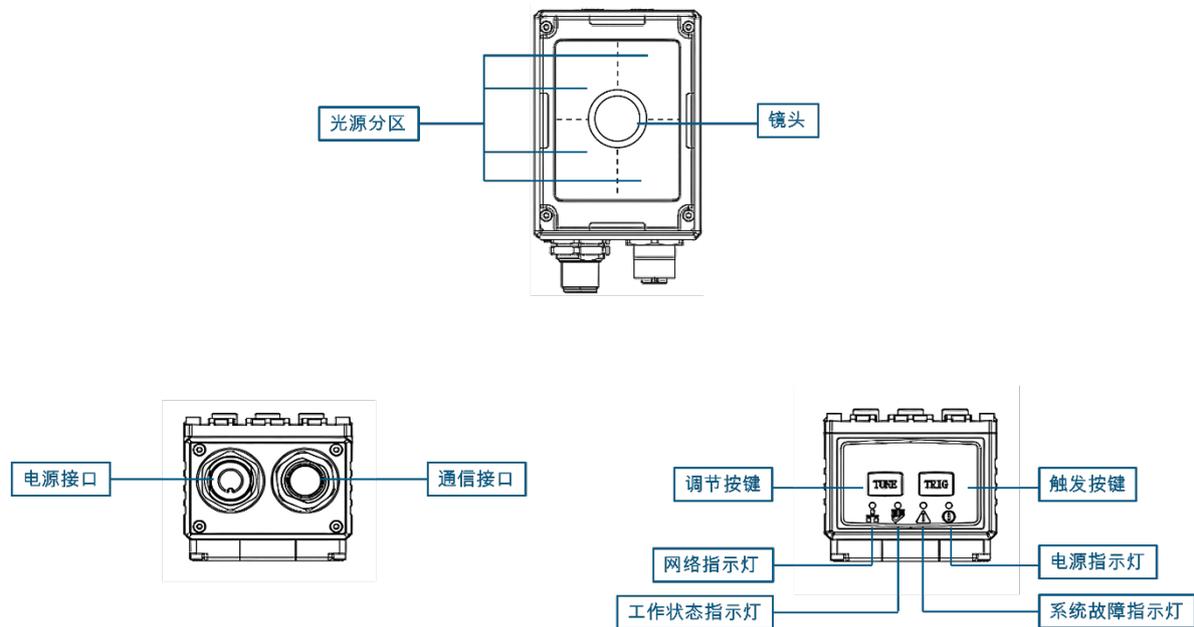
1 x ED-AIC2020 人工智能相机

### 可选配件

图例	订购编码	描述
	ED-ACC-AIC-ETH-A-03 ED-ACC-AIC-ETH-A-05 ED-ACC-AIC-ETH-A-10	M12 8 芯千兆网口转 RJ45 线
	ED-ACC-AIC-PWR-A-03 ED-ACC-AIC-PWR-A-05 ED-ACC-AIC-PWR-A-10	M12 12 芯电源 IO 线
	ED-ACC-AIC-DW-FIX001	支架，用于产品和型材之间固定
	ED-ACC-AIC-DW-FIX002	1 x 螺丝 M4* 8 3 x 垫片 1 x 螺钉 M6*10 1 x T 型螺母(连接型材)

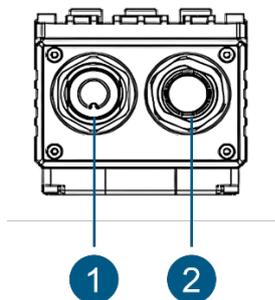
## 1.3 外观

介绍产品接口、按键和指示灯的分布。



## 1.4 接口

ED-AIC2020 包含通信和电源两个接口，均采用 M12 航空连接器，支持 IP65 防水等级。



编号	定义	描述
1	通信接口	千兆以太网接口，M12 8-Pin A-code 航空连接器
2	电源接口	电源接口、IO 接口和 RS232 串口，M12 12-Pin 航空连接器

### 1.4.1 电源接口

电源接口采用 M12 12-Pin 航空连接器，包含 1 路电源接口、1 路串口和 6 路通用 GPIO，引脚定义如下。

引脚 ID	线颜色	引脚名称
2	白色/黄色	DC+
3	棕色	COMMON_IN
4	白色/棕色	DI1
5	紫色	Trigger
6	白色/紫色	COMMON_OUT
7	红色	External Strobe
8	绿色	DO1
9	黑色	DO2
10	橙色	RS232_GND
11	蓝色	RS232_TX
12	灰色	RS232_RX

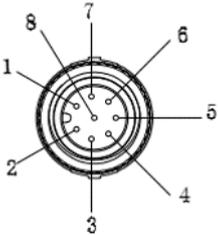
The diagram shows a circular M12 12-pin connector with 12 pins numbered 1 through 12. The pins are arranged in a circular pattern. Pin 1 is at the top, and pin 12 is at the bottom. The diagram is used to illustrate the pinout for the power interface.

DI 和 DO 引脚定义如下。

信号	CM4 引脚
DI1	GPIO17
DO1	GPIO22
DO2	GPIO27

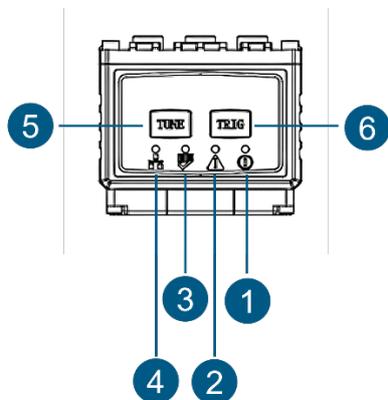
## 1.4.2 通信接口

通信接口采用 M12 8-Pin 航空连接器，引脚定义如下。

	引脚 ID	引脚名称
	1	TRD0+
	2	TRD0-
	3	TRD1+
	4	TRD2+
	5	TRD2-
	6	TRD1-
	7	TRD3+
	8	TRD3-

## 1.5 按键和指示灯

ED-AIC2020 包含 2 个按键和 4 个指示灯。



编号	描述
1	电源指示灯 <ul style="list-style-type: none"> <li>● 常亮：设备已上电</li> <li>● 熄灭：设备未上电</li> </ul>
2	系统故障指示灯
3	工作状态指示灯
4	网络连接指示灯
5	调节按键，一键式自动对焦按钮或用户自定义按钮
6	触发按键，用于相机触发或用户定义的一键式按钮

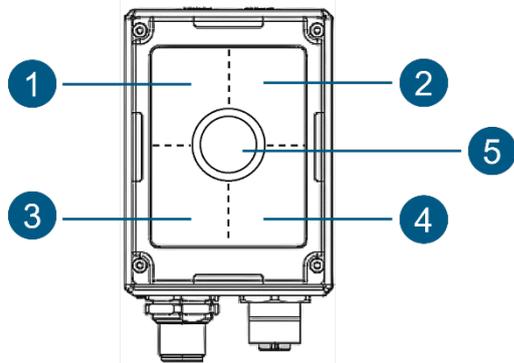
### 按键引脚定义

按键	CM4 引脚
调节按键	GPIO20
触发按键	GPIO12

### 指示灯引脚定义

指示灯	CM4 引脚
电源指示灯	N/A
故障指示灯	GPIO21
工作状态指示灯	GPIO7(异常) GPIO16(正常)
网络指示灯	N/A

## 1.6 光源和摄像机镜头



编号	描述
1	光源部分 1，支持单独启用和禁用
2	光源部分 2，支持单独启用和禁用
3	光源部分 3，支持单独启用和禁用
4	光源部分 4，支持单独启用和禁用
5	镜头，支持固定镜头和液态镜头

### 光源控制命令

信号	CM4 引脚 (GPIO 8/9, /dev/ttyAMA2)
控制光源部分 1	RS232 命令 <ul style="list-style-type: none"> <li>● 启用: 1-1</li> <li>● 禁用: 1-0</li> </ul>
控制光源部分 2	RS232 命令 <ul style="list-style-type: none"> <li>● 启用: 2-1</li> <li>● 禁用: 2-0</li> </ul>
控制光源部分 3	RS232 命令 <ul style="list-style-type: none"> <li>● 启用: 3-1</li> <li>● 禁用: 3-0</li> </ul>
控制光源部分 4	RS232 命令 <ul style="list-style-type: none"> <li>● 启用: 4-1</li> <li>● 禁用: 4-0</li> </ul>

## 2 安装设备

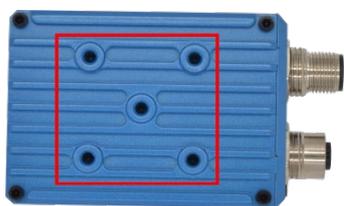
介绍安装设备的操作步骤。

### 前提条件:

- ◆ 已准备安装支架及安装螺钉(M4\*8 带垫片、M6\*10 带垫片)
- ◆ 已准备 M4 和 M6 的内六角螺丝刀

### 操作步骤:

1. 确定相机上安装孔的位置，如下图红框位置，一般用中央螺钉和一颗外围螺钉固定即可。



2. 将安装支架放置在相机安装孔的上方，使支架(带 M4 螺孔的一侧)和相机的中央螺丝孔位对齐，通过螺丝刀使用带垫圈的 M4 螺丝将支架固定在相机上，如下图所示。



3. 旋转支架，调整安装方向，按需选择顶部安装或者侧面安装，效果如下图:

顶部安装



侧面安装



4. 使用 M6 螺钉固定支架和其他设备，建议使用中心螺钉和一颗外围螺钉固定。



**提示：**

根据现场不同的工程要求，调整合适的安装位置和角度。

## 3 启动设备

介绍连接线缆和启动设备的操作步骤。

- ✓ 连接线缆
- ✓ 首次启动系统

## 3.1 连接线缆

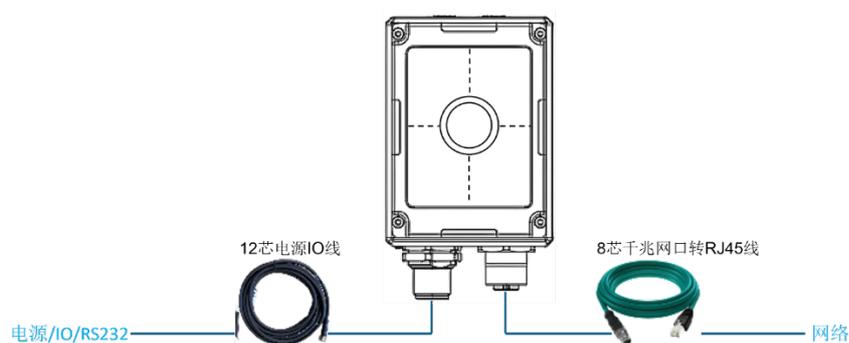
介绍线缆的连接方法。

**前提条件：**

- ◆ 已获取待连接的摄像机、8 芯千兆网口转 RJ45 线、12 芯电源 IO 线。
- ◆ 已准备 DC 12V 2A 的电源适配器和辅助接线的连接器。
- ◆ 因为 12 芯电源 IO 线的一端是 12 芯裸线，按需准备连接器和接线工具。

**连接线缆示意图**

各接口的引脚定义及连线的具体方法，请参见 [1.4 接口](#)。



## 3.2 首次启动设备

ED-AIC2020 无开关电源，接入电源后，设备将会开始启动。

- ◆ 电源指示灯点亮，表示设备已正常供电。
- ◆ 工作状态指示灯闪烁，表示系统正常启动。

使用 64 位 Raspberry Pi 操作系统，系统启动后，需要使用默认用户名和密码进行登录，因为摄像机无法连接显示器，故需要通过 PC 远程登录系统，具体的操作参考 [5.2 远程登录](#)。

## 4 安装操作系统

介绍镜像下载、打开/关闭设备外壳和烧录的操作步骤。

- ✓ 镜像下载
- ✓ 打开设备外壳
- ✓ eMMC 烧录
- ✓ 关闭设备外壳

## 4.1 镜像下载

最新版本的 64 位 Raspberry Pi 操作系统(Desktop)的下载地址为: [ED-AIC2020 Raspberry Pi OS\(64 位\)](#)。

## 4.2 打开设备外壳

烧录系统需要通过内部的 Micro USB 来进行，故烧录前需要打开外壳。

**前提条件：**

- ◆ 已准备一把 M1.5 的内六角螺丝刀
- ◆ 已准备一把 M2 的内六角螺丝刀

**操作步骤：**

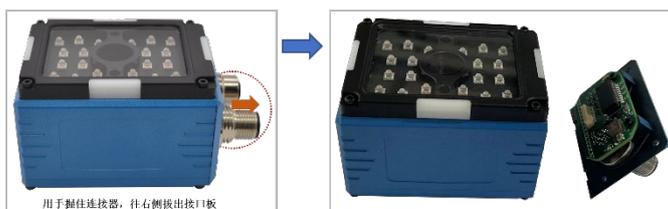
1. 使用 M2 的内六角螺丝刀逆时针拧下后面板的 4 颗螺丝，如下图红色标记的位置。



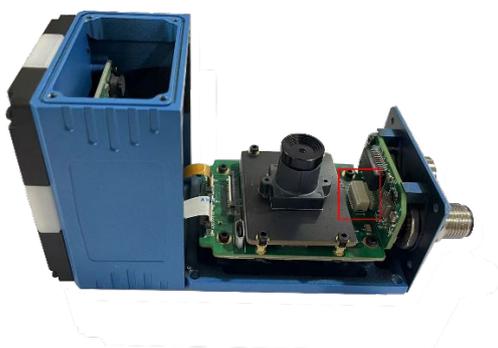
2. 使用 M1.5 的内六角螺丝刀逆时针拧下底部的 4 颗螺丝，如下图红色标记的位置。



3. 用手握住连接器，往右侧拔出接口板，如下图所示。



4. 向上打开上盖，用手握住接口板的连接器，对准两端的接口(红框标记端口)，再插入接口板，如下图所示。



## 4.3 eMMC 烧录

建议使用 Raspberry Pi 官方烧录工具，下载路径如下：

- ◆ Raspberry Pi Imager : [https://downloads.raspberrypi.org/imager/imager\\_latest.exe](https://downloads.raspberrypi.org/imager/imager_latest.exe)
- ◆ SD Card Formatter : <https://www.sdcardformatter.com/download/>
- ◆ Rpiboot : [https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot\\_setup.exe](https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot_setup.exe)

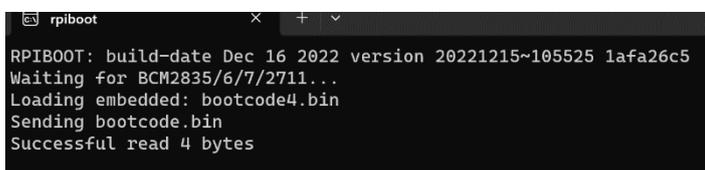
### 前提条件

- 已完成烧录工具的下载和并安装至电脑。
- 已准备一根 Micro USB 转 USB-A 线。
- 已打开设备外壳，具体操作请参见 [4.2 打开设备外壳](#)。
- 已获取待烧录的镜像文件。

### 操作步骤

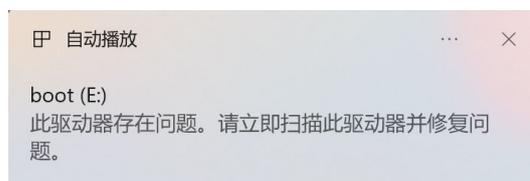
操作步骤以 Windows 系统为例进行说明。

1. 连接好电源线和 USB 烧录线(Micro USB 端连接至 ED-AIC2020 内部的 Micro USB 端口，另一端连接至 PC)。
2. 断开 ED-AIC2020 电源，再重新上电。
3. 打开已安装的 Rpiboot 工具，自动进行盘符化。

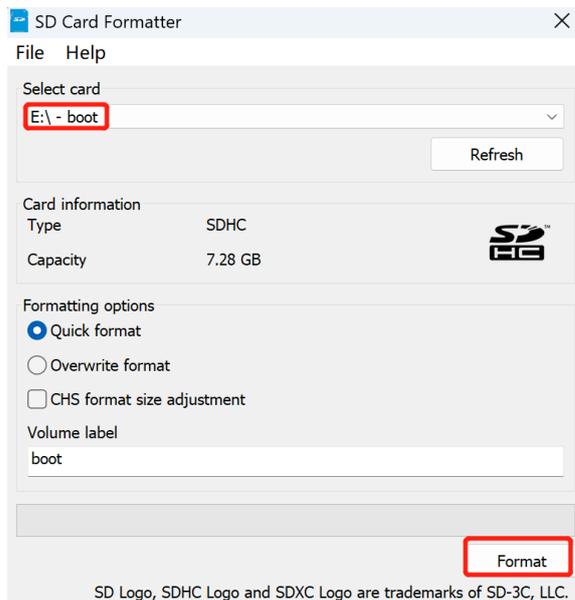


```
rpiboot
RPiBOOT: build-date Dec 16 2022 version 20221215~105525 1afa26c5
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Sending bootcode.bin
Successful read 4 bytes
```

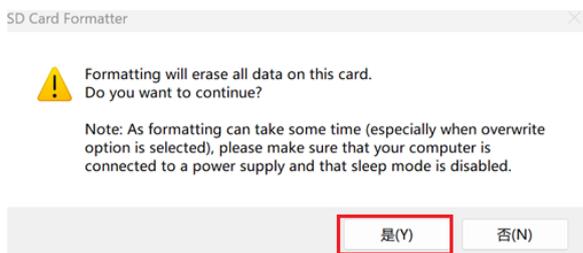
4. 待盘符化完成后，电脑右下角会弹出盘符，如下图中的 E 盘。



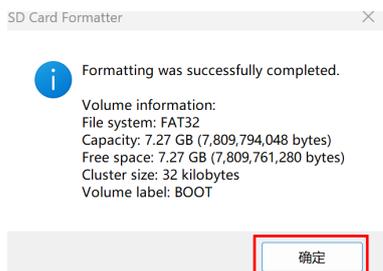
5. 打开 SD Card Formatter，选择被格式化的盘符，单击右下方“Format”进行格式化。



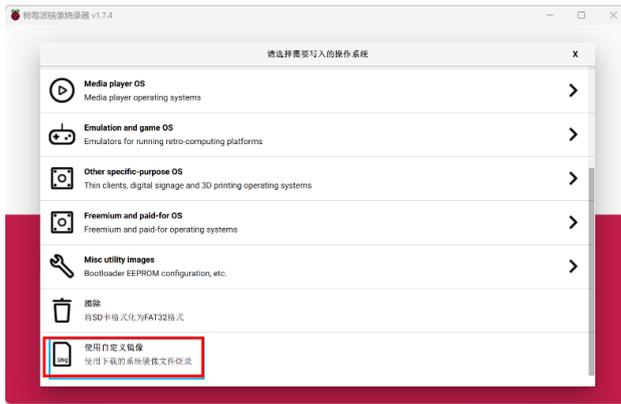
6. 在弹出的提示框中，单击“是”。



7. 格式化完成后，在提示框中单击“确定”。



8. 关闭 SD Card Formatter。
9. 打开 Raspberry Pi Imager，单击“选择操作系统”，在弹出的窗格中选择“使用自定义镜像”。



10. 根据提示，在自定义路径下选择下载好的镜像文件，并返回至烧录主界面。
11. 单击“选择 SD 卡”，在“存储卡”界面选择默认的 SD 卡，并返回至烧录主界面。



12. 单击“烧录”，在弹出的提示框中选择“是”，开始写入镜像。



13. 待镜像写入完成后，会进行文件的验证。



14. 验证完成后，弹出“烧录成功”提示框，单击“继续”完成烧录。



15. 关闭 Raspberry Pi Imager，取下 USB 连接线，重新给设备上电。



**提示：**

正常使用时，需要关闭设备外壳，具体操作请参考 [4.4 关闭设备外壳](#)。

## 4.4 关闭设备外壳

烧录完系统后需要关闭设备外壳，便于正常的使用。

**前提条件：**

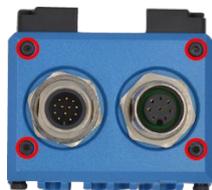
- ◆ 已准备一把 M1.5 的内六角螺丝刀
- ◆ 已准备一把 M2 的内六角螺丝刀

**操作步骤：**

1. 拔出接口板，向下盖上上盖板。
2. 用手握住接口板的连接器，对准两端的接口(红框标记处)，插入接口板。



3. 插入接口板，直到听到一声响，表示插入成功。
4. 使用 M1.5 的内六角螺丝刀顺时针拧紧底部的 4 颗螺丝，如下图红色标记的位置。



5. 使用 M2 的内六角螺丝刀顺时针拧紧后面板的 4 颗螺丝，如下图红色标记的位置。



## 5 配置设备

介绍系统配置的相关操作。

- ✓ 查找设备 IP
- ✓ 安装 BSP(可选)
- ✓ 远程登录
- ✓ 编译 Camera 演示
- ✓ aic.h 文件
- ✓ libaic.so 文件
- ✓ Camera API
- ✓ IO API

## 5.1 查找设备 IP

介绍获取设备 IP 的方法。

### 5.1.1 登录路由器来查询

**前提条件：**

- ◆ 摄像机已通过路由器接入网络
- ◆ 已获取所在网络的路由器的 IP 和网络密码，IP 地址如 192.168.X.X

**操作步骤：**

1. 打开浏览器，在地址栏中输入摄像机所在网络的路由器 IP：192.168.X.X，按 Enter 键进入路由器登录界面。
2. 按照界面提示，输入网络密码，进入路由器管理界面。
3. 在管理界面的终端设备中找到摄像机的 IP 地址。

### 5.1.2 使用 nmap 工具扫描

nmap 支持 Linux、macOS、Windows 等多个平台。如果希望使用 nmap 扫描 192.168.3.0~255 的网段，则可以使用以下命令：

**前提条件：**

- ◆ 摄像机已通过路由器接入网络
- ◆ 已获取所在网络的路由器的 IP 网段和掩码，例如 192.168.X.X/24，其中 24 为掩码

**操作步骤：**

1. 打开 nmap 工具，扫描 192.168.X.X/24 网段中的主机。

nmap 工具在不同的操作系统中，操作不同，根据实际界面或者命令的提示来操作即可。

2. 根据扫描出的结果，获取摄像机 IP。

## 5.2 安装 BSP(可选)

若使用的最新版本的 64 位 Raspberry Pi 操作系统(Desktop)，则无需安装。

若使用的是出厂默认的 64 位 Raspberry Pi 操作系统(Lite)，则需要安装。

**操作步骤:**

1. 下载 GPG 密钥并添加源列表。

```
curl -sS https://apt.edatec.cn/pubkey.gpg | sudo apt-key add -  
echo "deb https://apt.edatec.cn/raspbian stable main" | sudo tee /etc/apt/sources.list.d/edatec.list
```

2. 安装 BSP 和 SDK 包。

```
sudo apt update  
sudo apt install ed-aic2020-bsp ed-aic2020-sdk
```

3. 安装 WiringPi

```
git clone https://github.com/WiringPi/WiringPi.git  
cd WiringPi  
./build
```

## 5.3 远程登录

因为摄像机无法连接显示器，故需要通过 PC 远程登录系统。

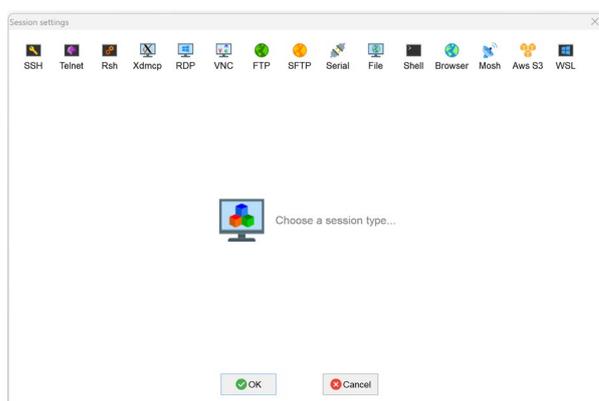
远程登录的方式由用户自己决定，下文以通过 MobaXterm 登录为例进行说明。

### 前提条件：

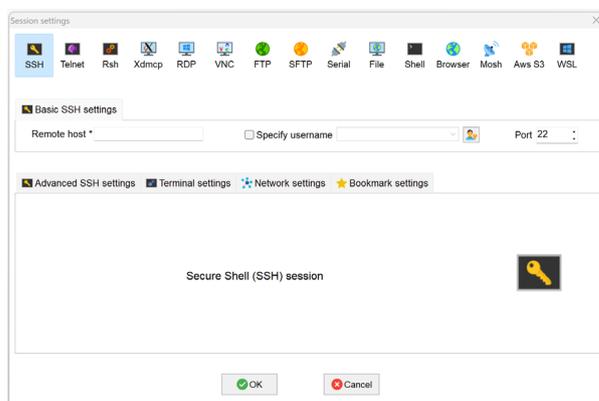
- ◆ 已在 PC 上安装 MobaXterm 工具。
- ◆ 摄像机已通过路由器接入网络。
- ◆ 已获取摄像机的 IP 地址。

### 操作步骤：

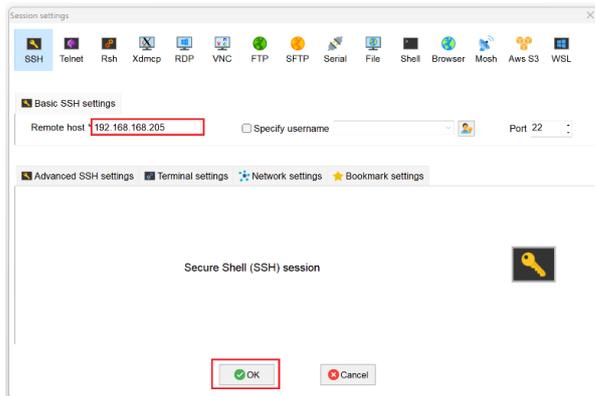
1. 打开 MobaXterm，单击  Session，打开创建连接的窗口，如下图所示。



2. 单击左上角的  SSH，打开 SSH 连接界面。



3. 输入已获取的摄像机的 IP 地址后，单击“OK”。



4. 在弹出的提示框中单击“Accept”，进入系统登录界面。
5. 根据提示输入用户名和密码，完成登录后进入系统。



提示：

默认用户名：**pi**，默认密码：**raspberry**。

```
login as: pi
pi@192.168.168.205's password:

• MobaXterm Personal Edition v23.0 •
(SSh client, X server and network tools)

▶ SSh session to pi@192.168.168.205
• Direct SSh : ✓
• SSh compression : ✓
• SSh-browser : ✓
• X11-forwarding : ✓ (remote display is forwarded through SSh)
▶ For more info, ctrl+click on help or visit our website.

Linux raspberrypi 5.10.92-v8+ #1514 SMP PREEMPT Mon Jan 17 17:39:38 GMT 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
last login: Sat May 6 10:07:19 2023 from 192.168.168.227

SSh is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$
```

## 5.4 编译 Camera 演示

如何编译启动测试 Camera 示例。

```
cd /usr/share/ed-aic2020-sdk/example/cd WiringPi

sudo make all

sudo /usr/share/ed-aic2020-sdk/example/build/test_camera
# -h help
sudo /usr/share/ed-aic2020-sdk/example/build/test_camera -h
```

## 5.5 aic.h 文件

“aic.h”文件提供了如何操作 Camera 的接口，该文件的存放路径为“/usr/lib/include/aic.h”。

## 5.6 libaic.so 文件

“libaic.so”库文件的存放路径为“/usr/lib/libaic.so”。

## 5.7 Camera API

Camera API 提供了 AI Camera Sensor 的控制函数，支持打开和关闭传感器、设置传感器工作模式、设置曝光时间和设置增益，下文分别按照不同的 Camera 场景进行说明。

### 5.7.1 AR0234 Camera

AR0234 Camera API 提供的 AI Camera Sensor 的控制函数如下表，包含获取图像数据、打开摄像头、关闭摄像头、设置工作模式、设置曝光时间和设置增益等。

函数	功能	参数取值
int register_imgReady_callback(img_Callback callback)	获取图像数据	—
virtual int open() = 0	打开摄像头	—
virtual int close() = 0	关闭摄像头	—
virtual int set_mode(int mode) = 0	设置工作模式	<ul style="list-style-type: none"> <li>• 0: 连续工作模式</li> <li>• 1: 外部触发模式</li> </ul>
virtual int get_mode(int *mode) = 0	获取工作模式	<ul style="list-style-type: none"> <li>• 0: 连续工作模式</li> <li>• 1: 外部触发模式</li> </ul>
virtual int set_geometry(int x_offset, int y_offset,int width,int height) = 0	设置窗口信息	—
virtual int get_geometry(int *x_offset, int *y_offset,int *width, int *height) = 0	获取窗口信息	—
virtual int set_exposure(int exp_value) = 0	设置曝光时间	—
virtual int get_exposure(int *exp_value) = 0	获取曝光时间	—
virtual int set_gain(int gain_value) = 0	设置增益	取值范围: 0~64
virtual int get_gain(int *gain_value) = 0	获取增益	取值范围: 0~64

#### 获取 AR0234 实例

```
#include "CameraManger.h"
#include "camera_0234.h"

void test()
{
    eda::Camera *t_camera = eda::create_ar0234();
    if(t_camera){
        eda::Camera_0234 *t_camera_1 = static_cast<eda::Camera_0234*>(t_camera);
    }
    ...
}
```

## 5.8 IO API

IO API 提供了 AI Camera IO 的控制函数，支持控制指示灯、控制激光、控制侧灯和控制输出等。

### 控制函数-C 语言环境

函数	功能
eda::Edalo *em = eda::Edalo::getInstance()	获取 IO 控制实例
void setup()	初始化 IO 设置
void openLaser()	打开激光
void closeLaser()	关闭激光
void setScanStat(bool good)	设置状态指示灯
void openAlarm()	打开警告指示灯
void closeAlarm()	关闭警告指示灯
void setDo1High(bool high)	设置 output1 输出
void setDo2High(bool high)	设置 output2 输出
void registerInput(IoTrigger callback)	注册 input 触发回调函数
void registerTrigger(IoTrigger callback)	注册 register 按键回调函数
void registerTune(IoTrigger callback)	注册 Tune 按键回调函数
void setRgbLight(uint8_t light)	设置 RGB 灯光

### 控制函数-Python3 语言环境

函数	功能
eda = Edalo.singleton()	获取 IO 控制实例
eda.setup()	初始化
registerTrigger(func_trigger)	注册 Trigger 按键回调
registerInput(func_trigger)	注册 Input 输入回调
registerTune(func_trigger)	注册 Tune 按键回调
eda.openLaser()	打开激光
eda.closeLaser()	关闭激光
eda.setScanStat(True)	设置状态指示灯
eda.openAlarm()	打开警告指示灯
eda.closeAlarm()	关闭警告指示灯
eda.setDo1High(True)	设置 output1 输出
eda.setDo2High(False)	设置 output1 输出
eda.setRgbLight(1)	设置侧灯

## 操作说明

### ◆ 初始化

1. 操作 IO 前需要获取 IO 实例 `eda::Edalo *em = eda::Edalo::getInstance();`
2. 对实例初始化 `em->setup();`

### ◆ IO 控制支持对事件注册回调函数:

- input 输入事件 `em->registerInput(trigger_input);`
- Trigger 按键 `em->registerTrigger(trigger_trigger);`
- Tune 按键 `em->registerTune(trigger_tune);`

### ◆ 控制 IO (必须先完成初始化)

- 控制激光 `em->openLaser();` 和 `em->closeLaser();`
- 控制状态指示灯 `em->setScanStat(true)` 和 `em->setScanStat(false)`
- 控制警报指示灯 `em->openAlarm()` 和 `em->openAlarm()`
- 控制两路输出 `em->setDo1High(false);` 和 `em->setDo2High(false);`

### ◆ 控制灯光 (必须先完成初始化)

- 控制侧灯颜色 `em->setRgbLight(1);`
  - 0: 关闭
  - 1: 红色
  - 2: 绿色
  - 3: 蓝色
- 控制灯源
  - 使能灯源 (默认已使能) `em->enableLightSection(1);` 取值范围 1~4, 对应不同分区
  - 禁用灯源 `em->disableLightSection(1);` 取值范围 1~4, 对应不同分区