



基于 MQTT 海思开发板的智能炫彩灯 最佳实践

文档版本 001

发布日期 2020/8/5

版权所有 © 上海海思技术有限公司 2020。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为上海海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

上海海思技术有限公司

地址：上海市青浦区金泽镇（西岑）水秀路 318 号 101 室 邮编：201718

网址：<http://www.hisilicon.com>

目 录

1 基于 MQTT 海思开发板的智能炫彩灯1

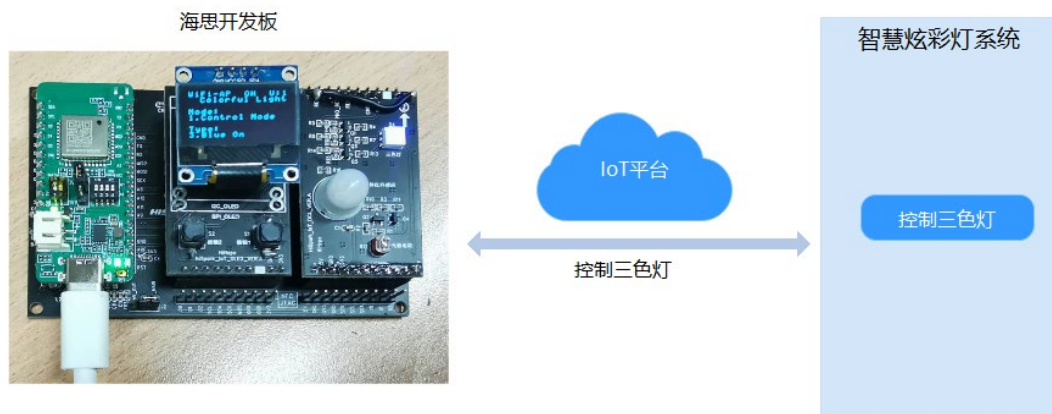
1 基于 MQTT 海思开发板的智能炫彩灯

场景说明

你是否觉得生活的色彩过于单调？好像除了天空的蓝、白云的白、森林的绿，暗夜的黑，没有其他想拥入怀中的色彩。

智慧炫彩灯能给平淡的生活增添一丝色彩，轻松调配出您专属的绚丽色彩、精彩生活、惊艳时光。

在该文档中，我们基于华为一站式开发工具平台（LiteOS Studio），从设备（海思开发套件）、平台（IoTDA 华为物联网平台）、端到端构建一款智慧炫彩灯样例，带您体验十分钟快速上云。

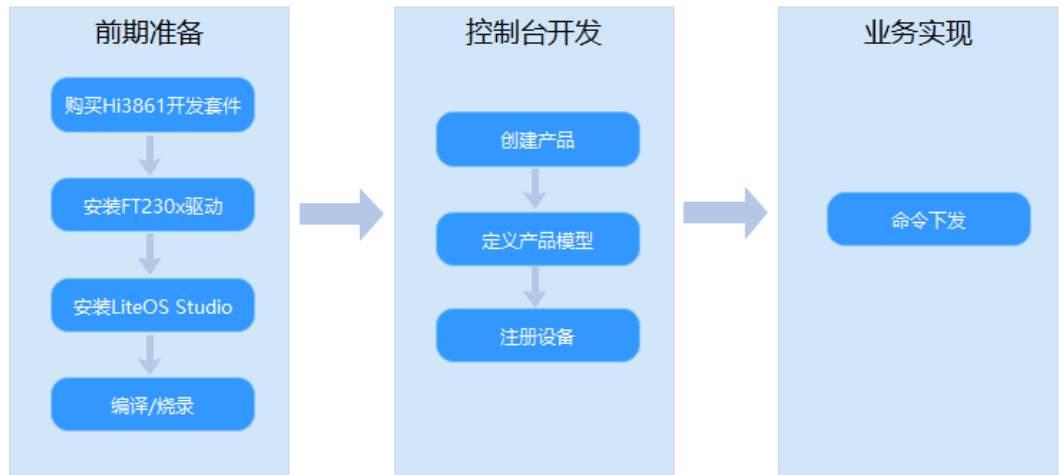


开发环境

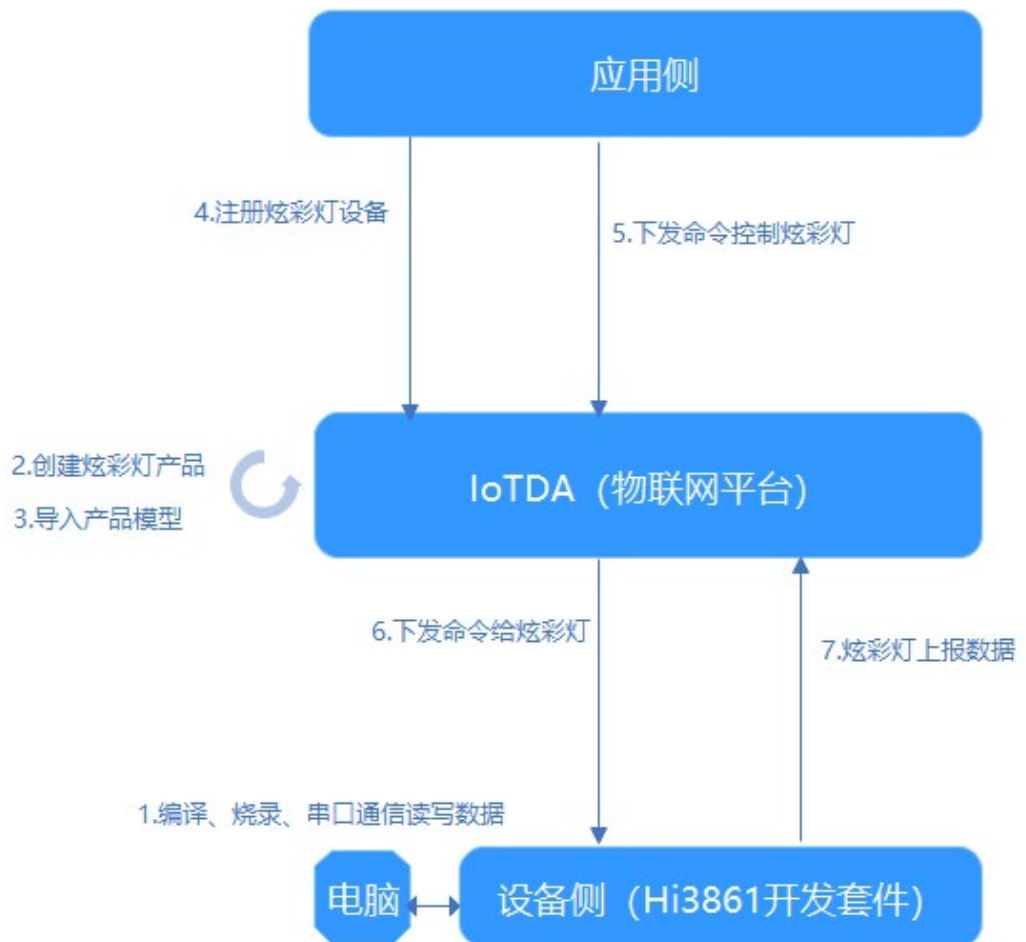
- 硬件：海思 WiFi 芯片 Hi3861 开发套件（包含 Hi3861 主板、OLED 屏板、智能炫彩灯板、通用底板、USB 数据线等）。
- 软件：Python、FT230x 驱动、LiteOS Studio、华为云（已开通[设备接入服务](#)）、win7 及以上版本的 64 位操作系统（本文以 windows 10 64-bit 系统调试）。

整体流程

智能炫彩灯案例的端到端整体流程如下图所示。



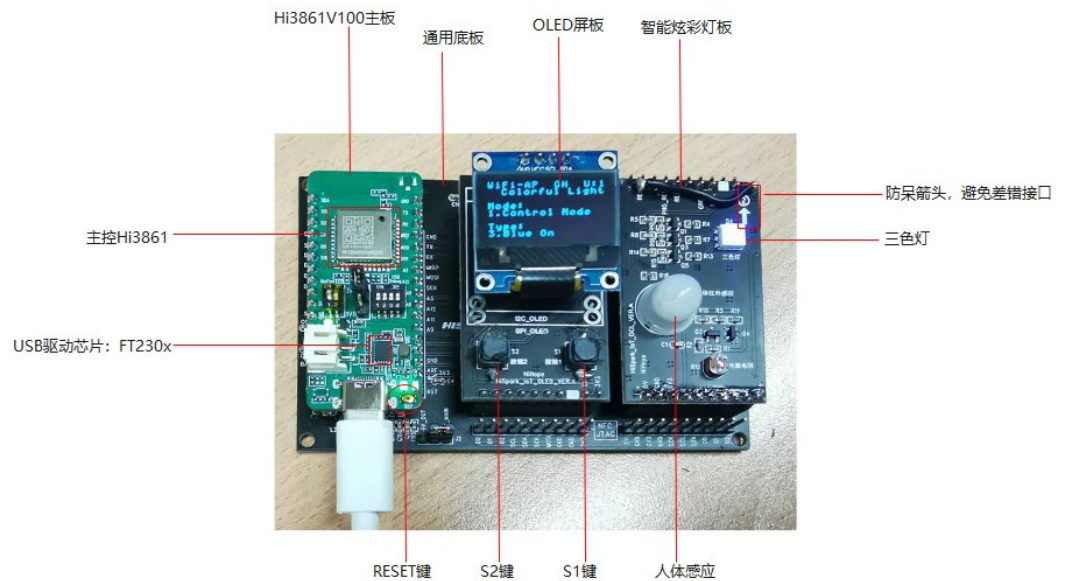
该场景主要描述的是设备可以通过 MQTT 协议（Wifi 模块）与物联网平台进行交互，应用侧可以给设备侧下发命令。



硬件连接

1. 将 Hi3861 主板按照如下图所示，安装在底板上。
2. 将 OLED 屏板安装在底板上。
3. 根据防呆箭头，将智能彩色灯板安装在底板上。

4. 使用 USB 线连接开发板和电脑，三色灯被点亮，表示通电正常。



OLED 显示屏说明:

- **WiFi-AP ON U:1:** 表示 WiFi 的 AP 状态开启, OFF 表示 WiFi 的 AP 状态关闭, U 表示有几台设备与主板建立连接。
- **Colorful Light:** 表示当前演示的是哪一个 demo。
- **Control Mode:** 表示当前在哪个模式下。
- **Blue on:** 表示当前蓝灯亮。

安装 Python 及配置环境变量

步骤 1 访问 [Python 官网](#), 下载 Python 软件。(本文以 win10-x64 Python3.7.6 为例)。

| Version | Operating System | Description | MD5 Sum | File Size | GPG |
|---|------------------|-----------------------------|----------------------------------|-----------|---------------------|
| Gzipped source tarball | Source release | | 3ef90f064506dd85b4b4ab87a7a83d44 | 23148187 | SIG |
| XZ compressed source tarball | Source release | | c08fbee72ad5c2c95b0f4e44bf6fd72c | 17246360 | SIG |
| macOS 64-bit/32-bit installer | Mac OS X | for Mac OS X 10.6 and later | 0dfc4cd9404cf0f5274d063eca4ea71 | 35057307 | SIG |
| macOS 64-bit installer | Mac OS X | for OS X 10.9 and later | 57915a926caa15f03ddd638ce714dd3b | 28235421 | SIG |
| Windows help file | Windows | | 8b915434050b29f9124eb93e3e97605b | 8158109 | SIG |
| Windows x86-64 embeddable zip file | Windows | for AMD64/EM64T/x64 | 5f84f4f62a28d3003679dc693328f8fd | 7503251 | SIG |
| Windows x86-64 executable installer | Windows | for AMD64/EM64T/x64 | cc31a9a497a4ec8a5190edec5cdd303 | 26802312 | SIG |
| Windows x86-64 web-based installer | Windows | for AMD64/EM64T/x64 | f9c11893329743d77801a7f49612ed87 | 1363000 | SIG |
| Windows x86 embeddable zip file | Windows | | accb8a137871ec632f581943c39cb566 | 6747070 | SIG |
| Windows x86 executable installer | Windows | | 9e73a1b27bb894f87fdce430ef88b3d5 | 25792544 | SIG |
| Windows x86 web-based installer | Windows | | c7f474381b7a8b90b6f07116d4d725f0 | 1324840 | SIG |

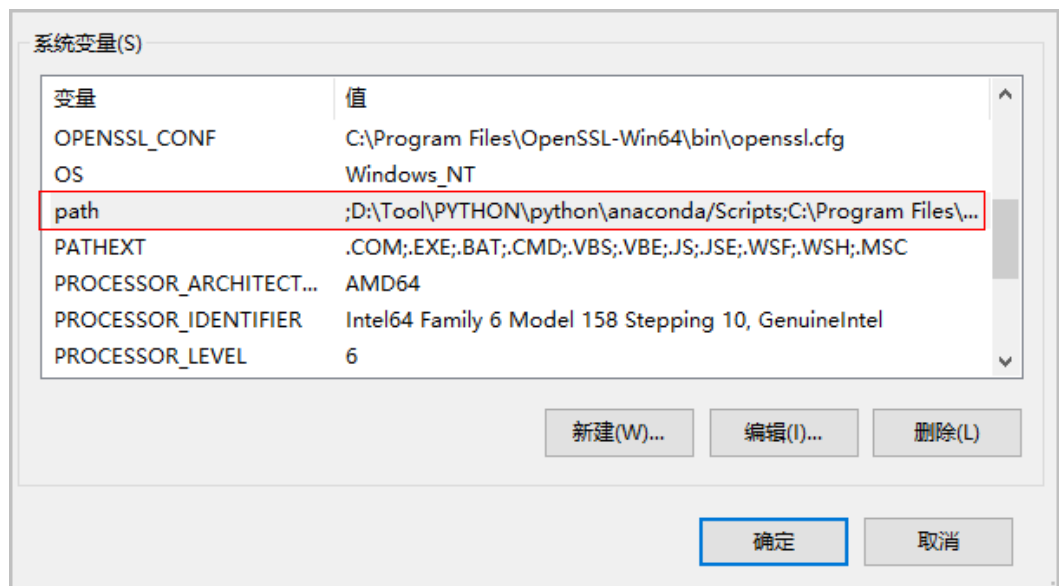
步骤 2 运行 exe 文件, 根据界面提示进行安装。

注: 注意保存好安装目录, 配置环境变量时用。

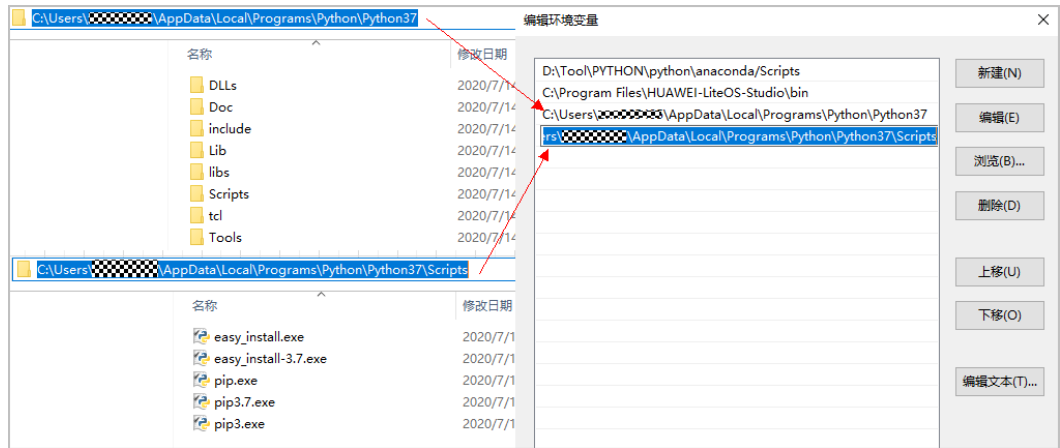
步骤 3 以 win10 为例, 添加环境变量: 在运行窗口中输入 `pc`, 单击“属性”, 选择“高级系统配置”, 然后单击“环境变量”。



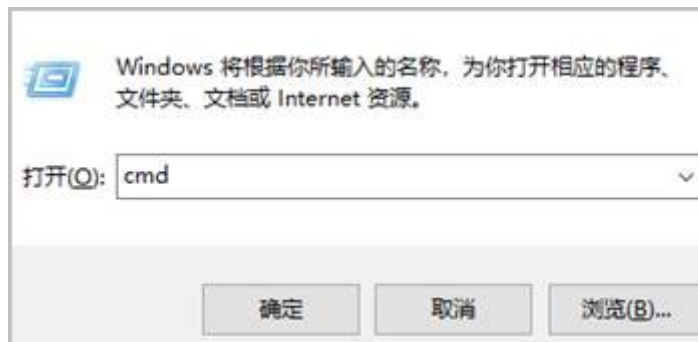
步骤 4 双击“path”的系统变量，进入编辑环境变量页面。



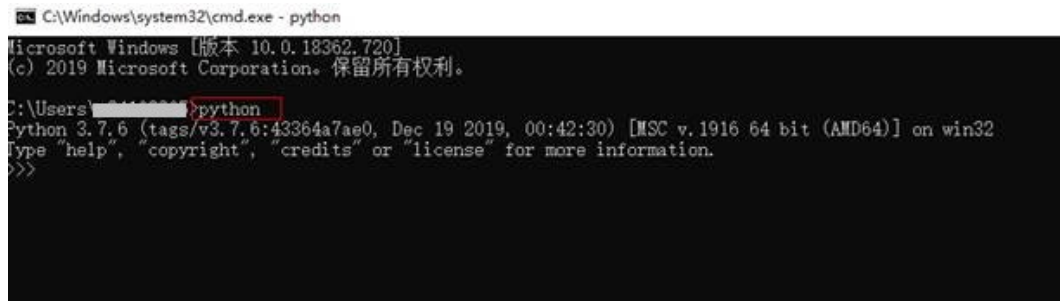
步骤 5 将“x:\xxx\Python37”和“x:\xxx\Python37\ Scripts”加入系统的 Path 环境变量，然后单击“确定”，如 C:\Python37\Scripts 加入环境变量。



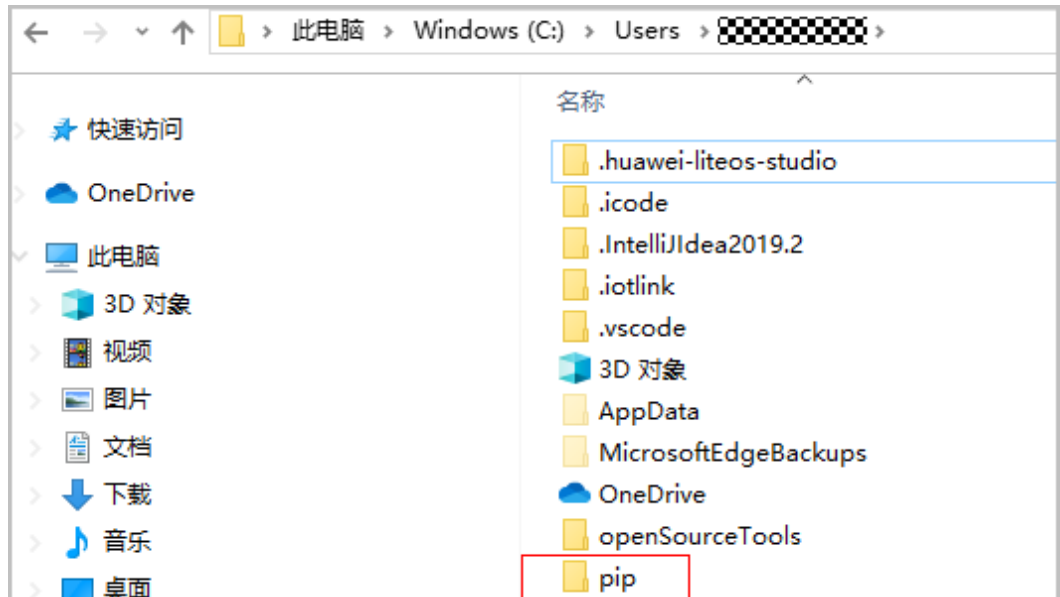
步骤 6 按“win”+“R”输入 cmd 进入 dos 模式:



步骤 7 输入“Python”，如果输出下面信息表示安装 python 成功。



步骤 8 在“本地电脑 > Windows > Users > XXX”用户下创建一个名为 pip 的文件夹。



步骤 9 访问 pip 文件，在 pip 文件夹下新建一个名为“pip”的 txt 文件，并将以下内容复制拷贝在 txt 中。

```
[global]
index-url = https://repo.huaweicloud.com/repository/pypi/simple/
trusted-host = repo.huaweicloud.com
timeout = 120
```



步骤 10 将“pip.txt”文件的后缀名改为“.ini”。

步骤 11 重新进入 dos 模式，分别执行以下命令，安装 Scons。

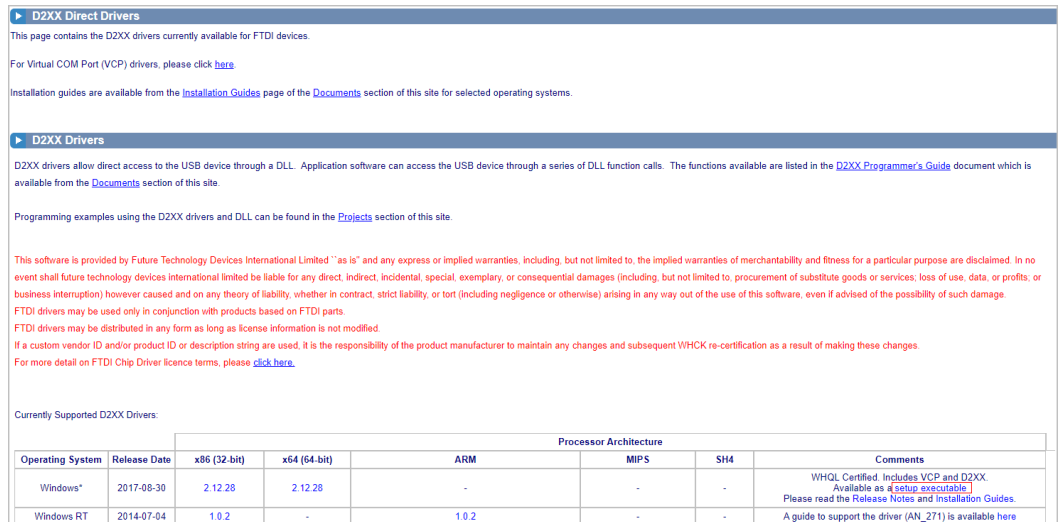
```
pip install pycryptodome
pip install ecdsa
pip install pywin32
```

至此，环境变量配置全部完成。

---结束

安装驱动

步骤 1 访问 [FTDI 官网](#)，点击“setup executable”下载驱动软件。



步骤 2 解压后运行 exe 文件，根据界面提示进行安装。

步骤 3 (可选) 开 PC 设备管理器可查看对应的驱动是否安装成功。若下图所示，表明驱动安装成功。



----结束

安装 LiteOS Studio

步骤 1 访问[这里](#)，下载 LiteOS Studio 安装包。

LiteOS Studio 是基于 LiteOS 轻量级操作系统开发的工具。它提供了代码编辑、编译、烧录、调试及 Trace 跟踪等功能，可以对系统关键数据进行实时跟踪及保存与回放。



步骤 2 运行安装文件，根据界面提示进行安装。（本文以 LiteOS Studio 1.45.10 为例）。

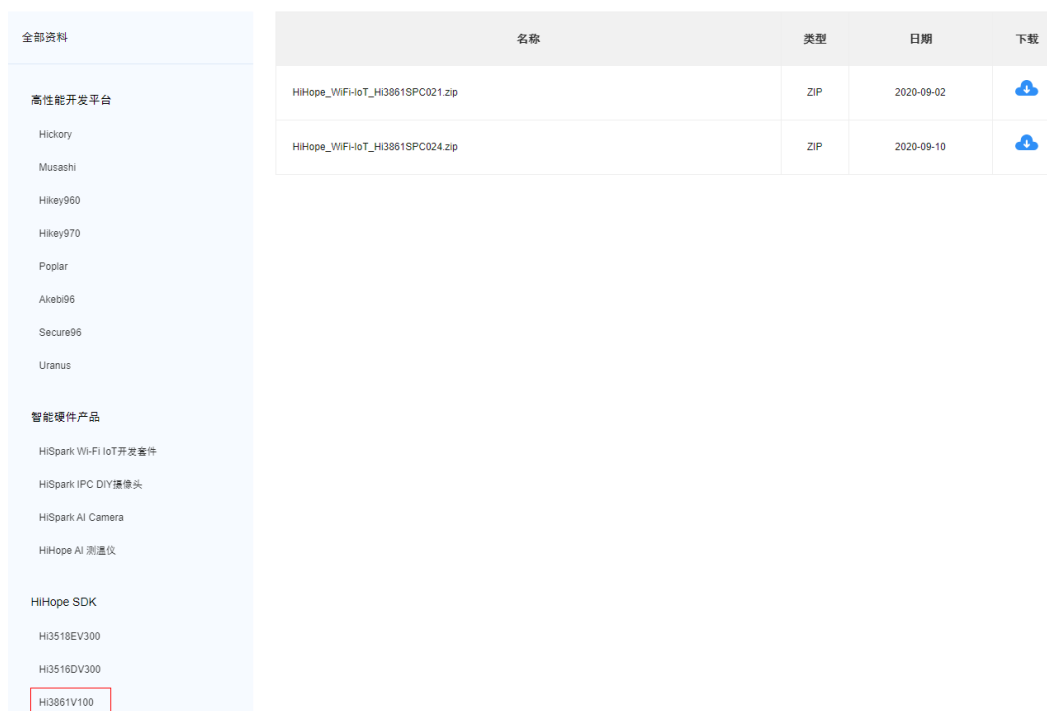
📖 说明



安装过程中不要有其他操作，避免安装失败。

---结束

导入代码样例及配置工程

步骤 1 访问 [HiHope 官网](#)，下载 Hi3861V100 代码样例。



| 名称 | 类型 | 日期 | 下载 |
|----------------------------------|-----|------------|---|
| HiHope_WiFi-IoT_Hi3861SPC021.zip | ZIP | 2020-09-02 |  |
| HiHope_WiFi-IoT_Hi3861SPC024.zip | ZIP | 2020-09-10 |  |

全部资料

高性能开发平台

- Hickory
- Musashi
- Hikey960
- Hikey970
- Poplar
- Akebi96
- Secure96
- Uranus

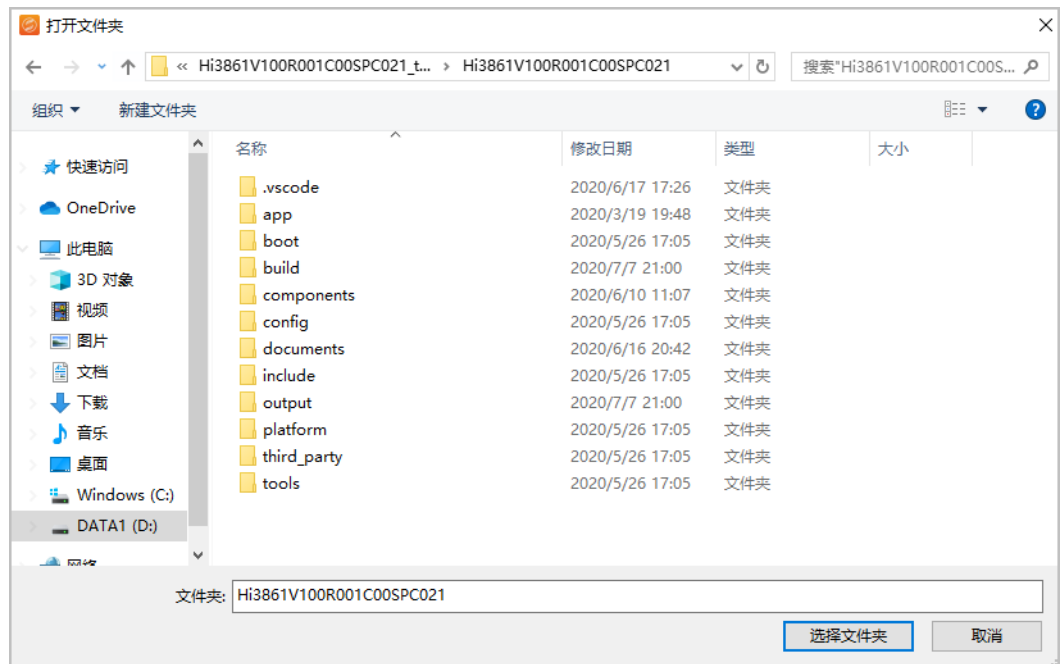
智能硬件产品

- HiSpark Wi-Fi IoT 开发套件
- HiSpark IPC DIY 摄像头
- HiSpark AI Camera
- HiHope AI 测速仪

HiHope SDK

- Hi3518EV300
- Hi3516DV300
- Hi3861V100**

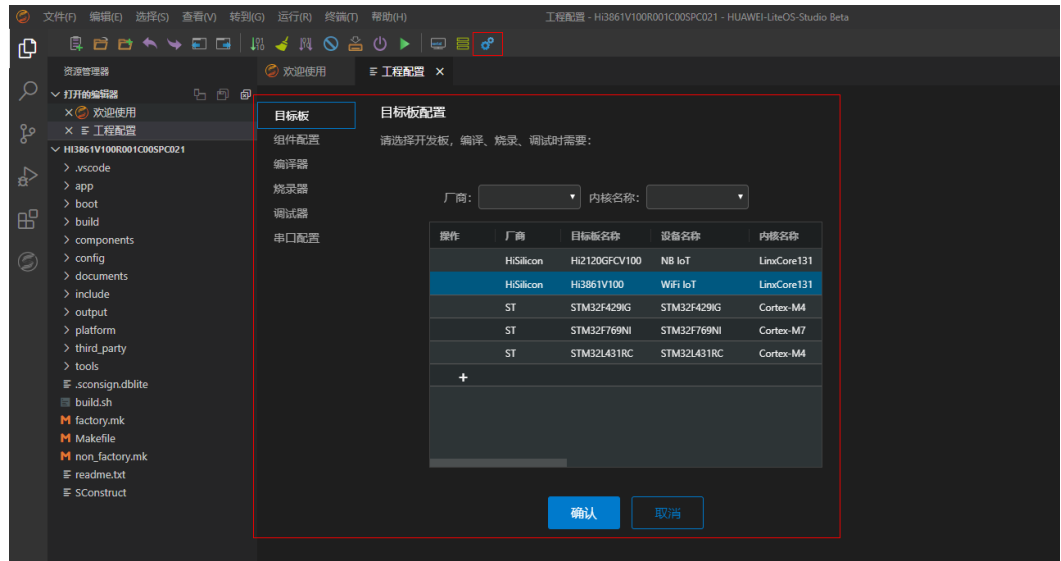
步骤 2 运行 LiteOS Studio 软件，单击“文件 > 打开文件夹”加载代码样例。



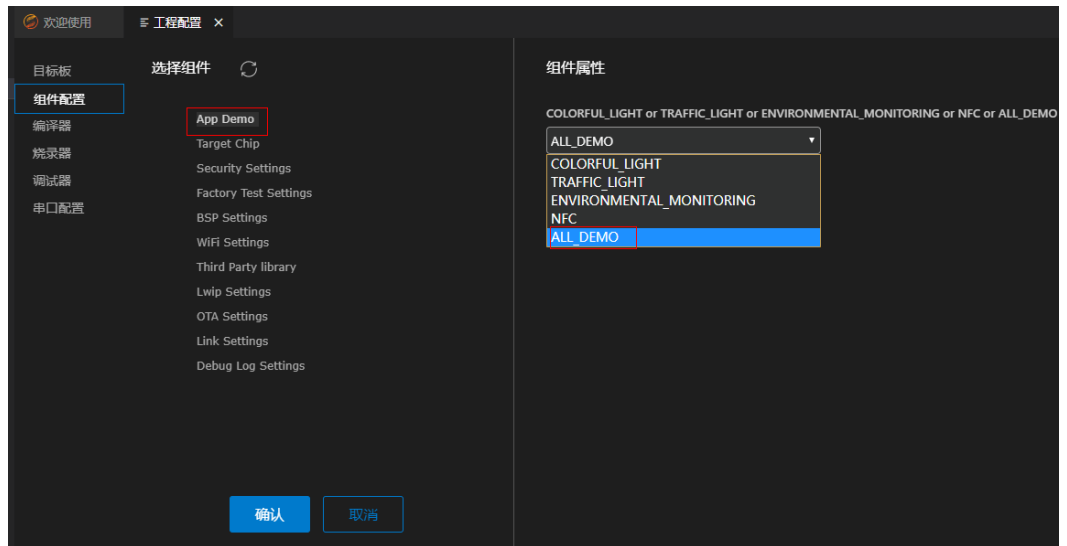
SDK 文件结构介绍:

- app: 存放 demo 示例文件;
- boot: 包含 commonboot/flashboot/loadboot, 包含 3861 的 drivers/startup 等;
- build: 编译配置
- components: Hi3861 组件, 如 wifi/histreaming/at/hilink/mcast6 等;
- config: 系统配置;
- documents: Hi3861 的开发指南等;
- include: Hi3861 包含的头文件;
- output: 编译后输出的编译文件 bin;
- platform: 包含 Hi3861 的外设驱动, 如 i2c/spi/uart; cpu 的内核 core;
- third_party: 第三方应用, 如 mqtt/cjson/coap 等;
- tool: 配置工具, 如 python 脚本/xml 文件等。

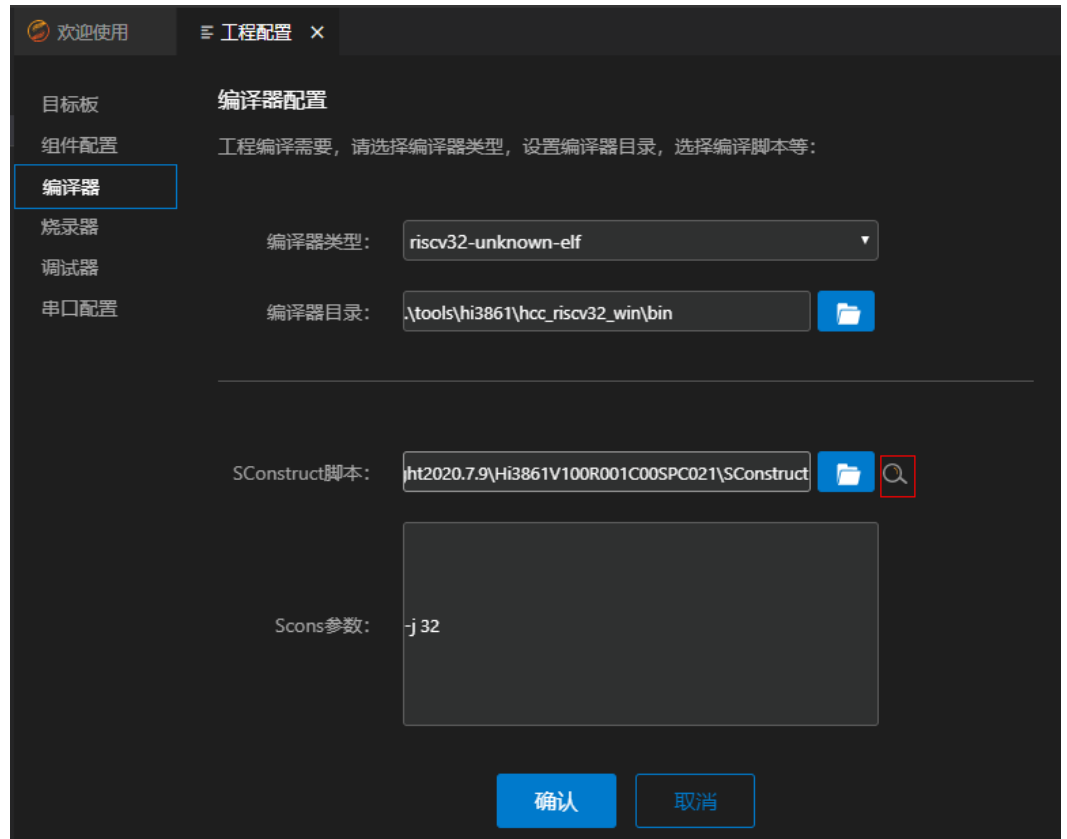
步骤 3 工程配置: 单击软件窗口上方“工程配置”按钮, 进入配置工程界面。



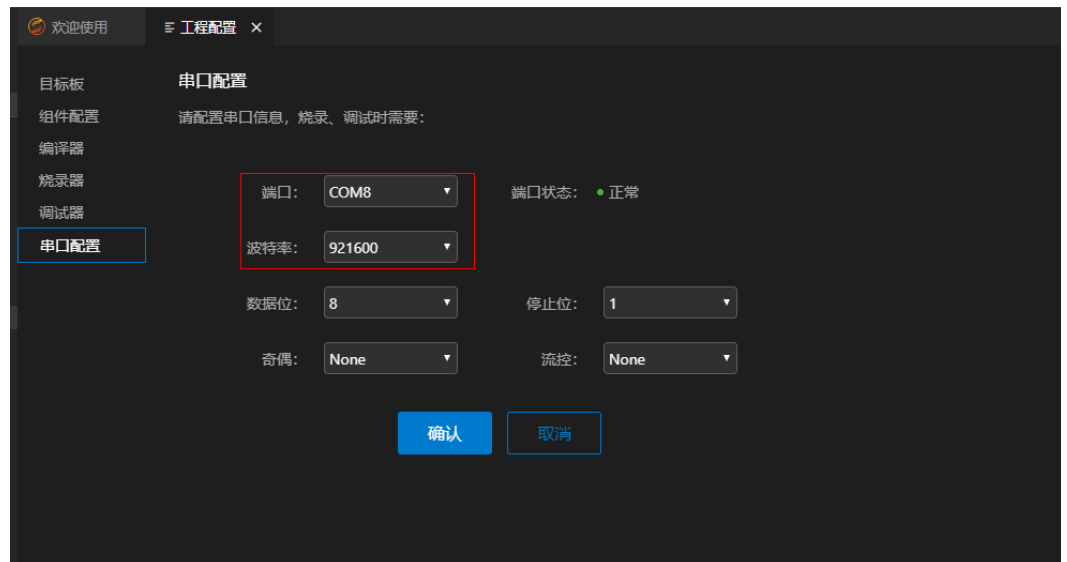
1. 目标板配置：已自动匹配目标板，不需要再单独配置。
2. 组件配置：单击 App Demo，下拉选择 ALL_DEMO，然后单击“确认”。



3. 编译器配置：单击“搜索框”，自动选择脚本文件，不需要手动选择。



4. 烧录器配置：保持默认值，不需要单独配置。
5. 调试器配置：保持默认值，不需要单独配置。
6. 串口配置：选择 3 中的端口号，设置波特率为 921600，然后单击“确认”。



至此，工程配置操作全部完成。

---结束

创建产品

- 步骤 1** 登录华为云官方网站，访问[设备接入服务](#)。
- 步骤 2** 单击“立即使用”进入设备接入控制台。
- 步骤 3** 单击左侧导航栏的产品，单击右上角下拉框，选择新建产品所属的资源空间。



- 步骤 4** 单击右上角的“创建产品”，创建一个基于 MQTT 协议的产品，参考下表填写参数后，单击“立即创建”。



| 基本信息 | |
|--------|-------------------|
| 所属资源空间 | 选择您需要归属的资源空间 |
| 产品名称 | 自定义，如“Wifi_IoT” |
| 协议类型 | 选择“MQTT” |
| 数据格式 | 选择“JSON” |
| 厂商名称 | 自定义，如“HiSilicon” |
| 功能定义 | |
| 选择模型 | 不勾选，本文使用离线导入产品模型。 |
| 所属行业 | 智慧城市 |
| 设备类型 | WifiIoT |

- 步骤 5** 产品创建成功后，单击“详情”进入产品详情页面，进行后续操作。

| 产品名称 | 产品ID | 设备类型 | 协议类型 | 操作 |
|-------------|--------------------------|---------------|------|---------------------------------------|
| Wifi_IoT | 5f0d714869c46102cb1e1a8c | WifiIoT | MQTT | 详情 删除 |
| MQTTProduct | 5f0d216cc1ee2202fc2ce6b0 | colorfullight | MQTT | 详情 删除 |

---结束

导入产品模型

产品模型是用来描述设备能力的文件，通过 JSON 的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备的功能。我们已提供开发好的产品模型，帮助您快速体验上云流程。

步骤 1 在产品详情，“功能定义”页面，单击“上传模型文件”。



步骤 2 在弹出的页面中，上传提供的产品模型，然后单击“确认”。



---结束

注册设备

本文介绍 MQTT 设备的注册方法。

步骤 1 在设备接入控制台，选择“设备 > 所有设备”，单击右上角的“注册设备”。

步骤 2 在弹出的页面中，按照如下表格填写参数后，单击“确定”。

设备认证类型 ✕

* 所属资源空间

* 所属产品

* 设备标识码

* 设备名称

设备认证类型 密钥 X.509证书

密钥

确认密钥

确定
取消

| 参数名称 | 说明 |
|--------|--|
| 所属资源空间 | 选择设备所属的资源空间。 |
| 所属产品 | 选择 此处 创建的产品。 |
| 设备标识码 | 即 nodeID，自定义，英文字母和数字的组合字符串。通过注册成功后生成的“设备 ID”（与设备标识码一一对应）和“设备密钥”接入平台。 |
| 设备名称 | 即 device_name，可自定义。 |
| 设备认证类型 | 密钥：设备通过密钥方式接入。 |
| 密钥 | 设备密钥，可自定义，不填写物联网平台会自动生成。 |

说明

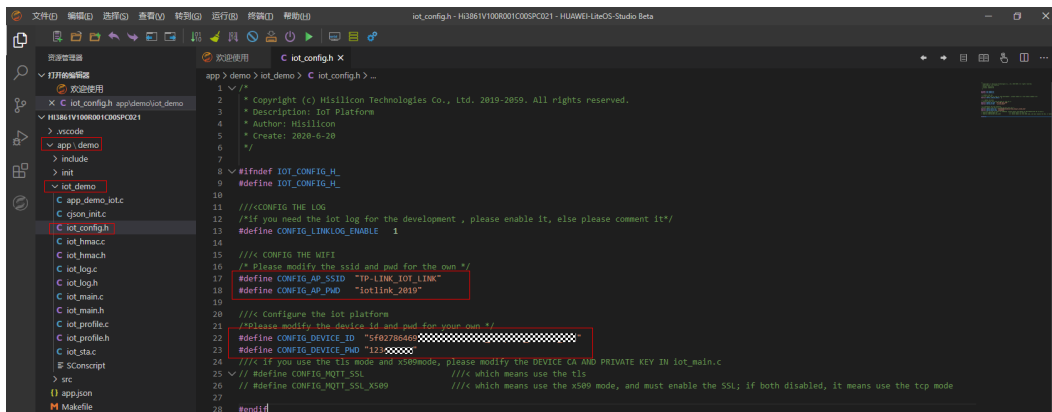
成功注册设备后，请保存好设备 ID 和设备密钥。

---结束

修改代码样例

提供的 demo 样例中需要设置设备 ID，设备密钥，以及 Wifi 账号和密码，以便设备连接华为云平台并上线。


步骤 1 访问“app > iot_demo > iot_config.h”，打开 iot_config.h 配置文件。

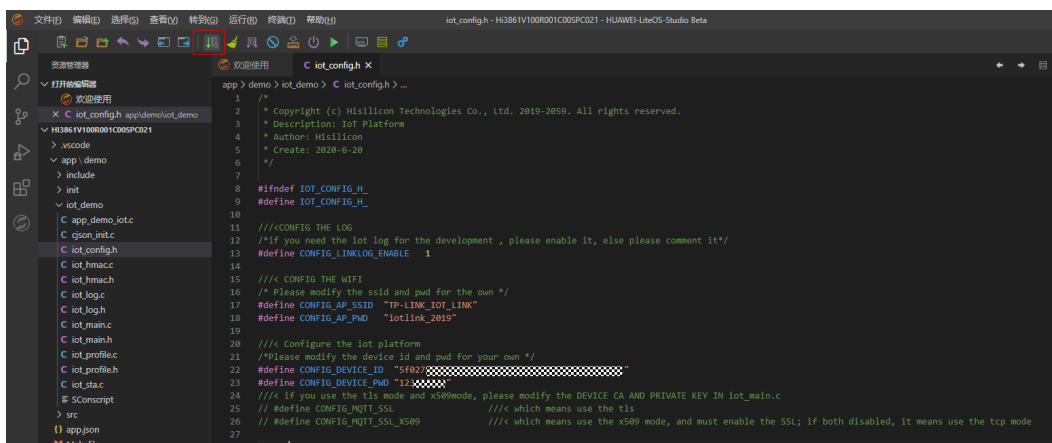


步骤 2 设置您的 Wifi 账号，密码，以及在 2 中生成的设备 ID 和设备密钥。

---结束


编译及烧录

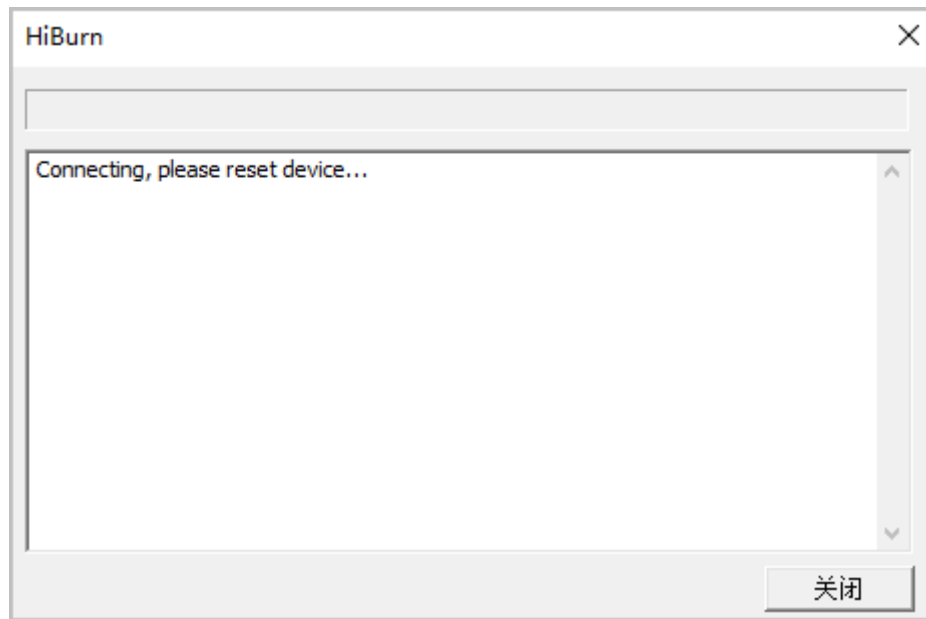
步骤 1 单击工具栏上的  进行编译。



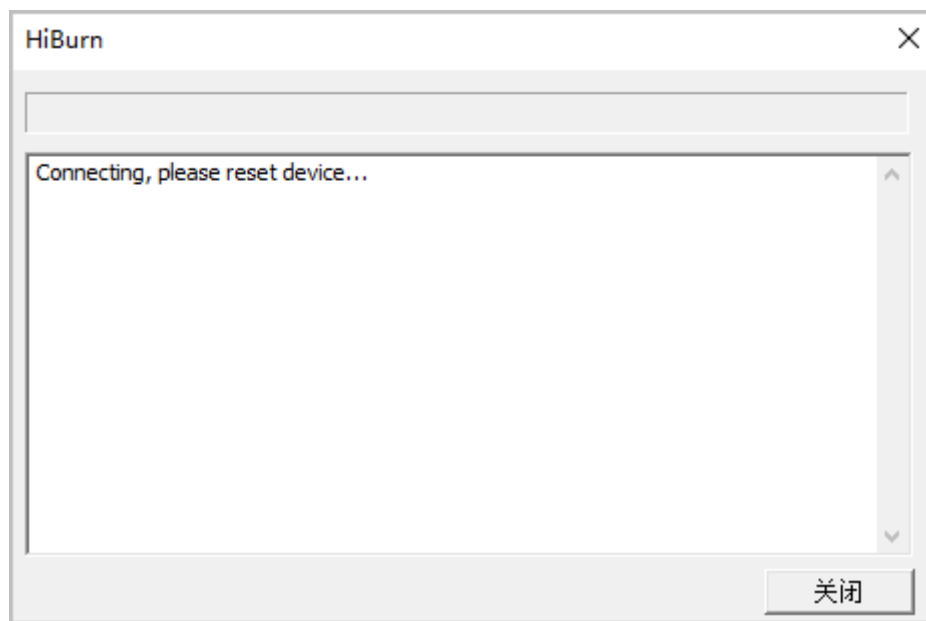
步骤 2 编译成功后，界面提示如下。



步骤 3 单击工具栏的  图标进行烧录，在弹出烧录进度框时，按下开发板上的 REST 键开始烧录。



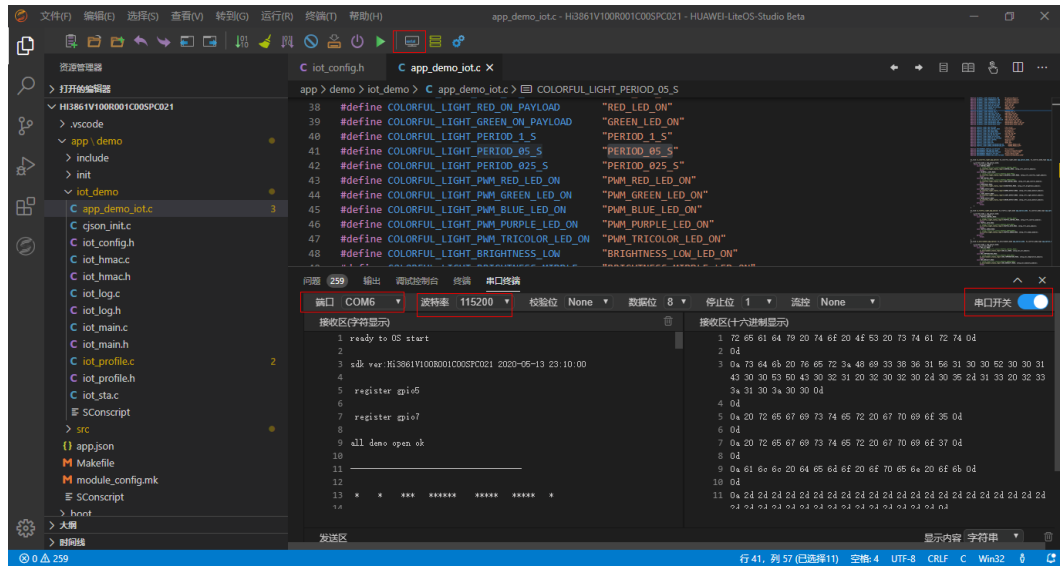
步骤 4 烧录成功后，单击“关闭”按钮，关闭烧录弹窗。



说明

如果没有单击关闭窗口，用户再次按下开发板上的 RESET 键会再次进入烧录过程。

步骤 5 烧录成功后，单击工具栏的“串口终端（SerialPort Terminal）”，选择 3 中的端口号，波特率设置为 115200，开启串口开关，按下开发板上的复位键，可以看到串口输出内容。

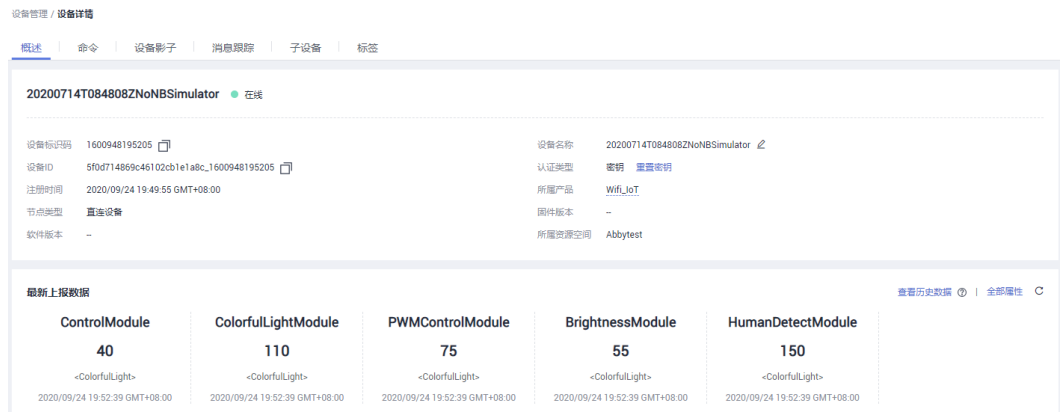


----结束

数据上报

平台和开发板建立连接后，根据烧录至开发板的代码，炫彩灯每隔 1 秒（上报频率可根据业务需要在 demo 中自行设置）上报三色灯（红、绿、蓝）状态、三色灯闪烁频率（1s、0.5s、0.25s）、呼吸灯（红、紫、白、绿、蓝）状态（由暗变亮）、白色灯的亮度、红外感应结果（人走动时，上报有人被检测到；人离开后，上报没人被检测到）、光照感应结果（光弱时候（黑暗），上报有灯光；光强时候（白天），上报没有灯光）、以及当黑夜有人时，上报有人结果；黑夜无人时，上报无人结果。

数据上报到平台后，您可以在设备详情页面，查看上报到平台的数据。



下发命令

下发命令之前，请按右侧 S1 键进入 Colorful light 模式。

步骤 1 在设备列表页面，单击设备右侧的“查看”，进入设备详情页面。



步骤 2 在“命令”页签，单击同步命令下发右侧的“命令下发”，在弹出的窗口中选择需要下发的命令并设置命令参数。例如下发一条 ControlModule，值为 BLUE_LED_ON 的命令，您可以看到三彩灯蓝灯被点亮，继续下发一条 ColorfulLightModule，值为 PERIOD_1_S 的命令，您可以看到间隔 1s，蓝灯不断闪烁。

图1-1 控制台下发命令

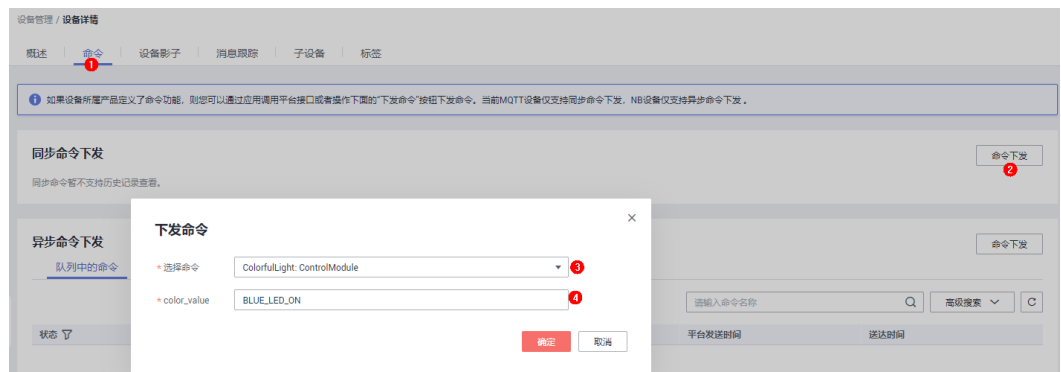
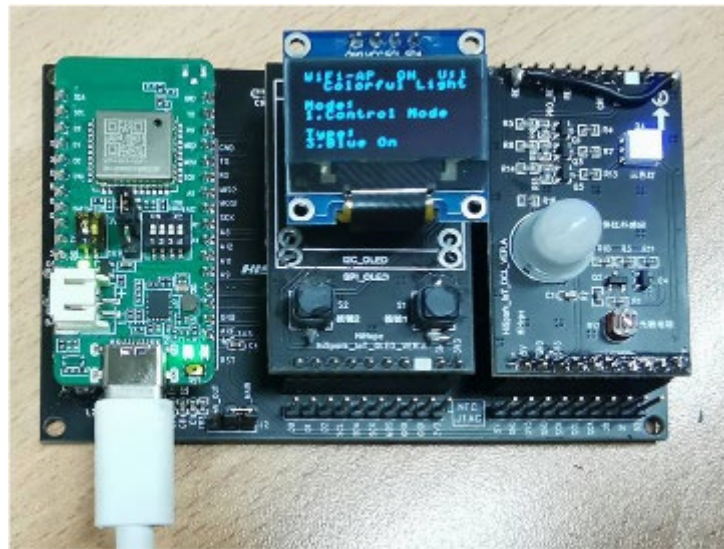


图1-2 蓝灯被点亮



---结束