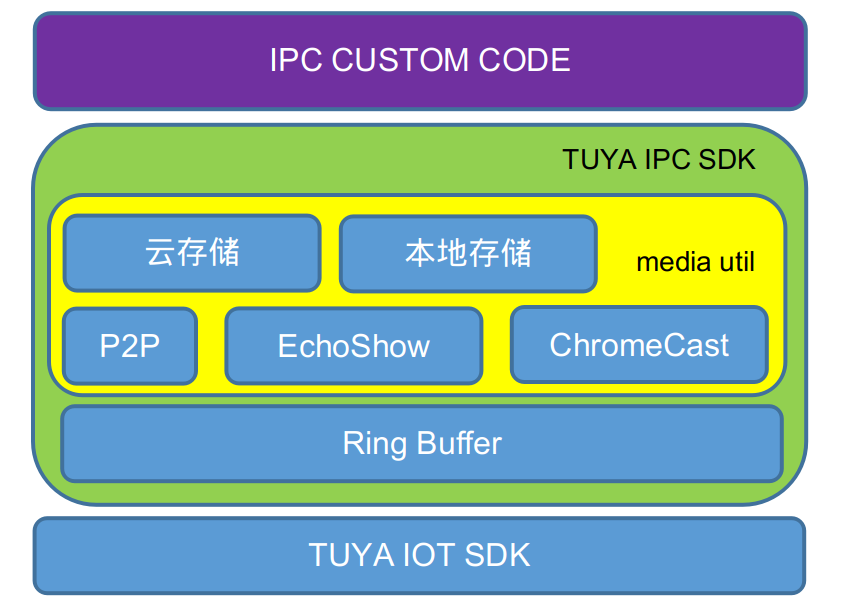
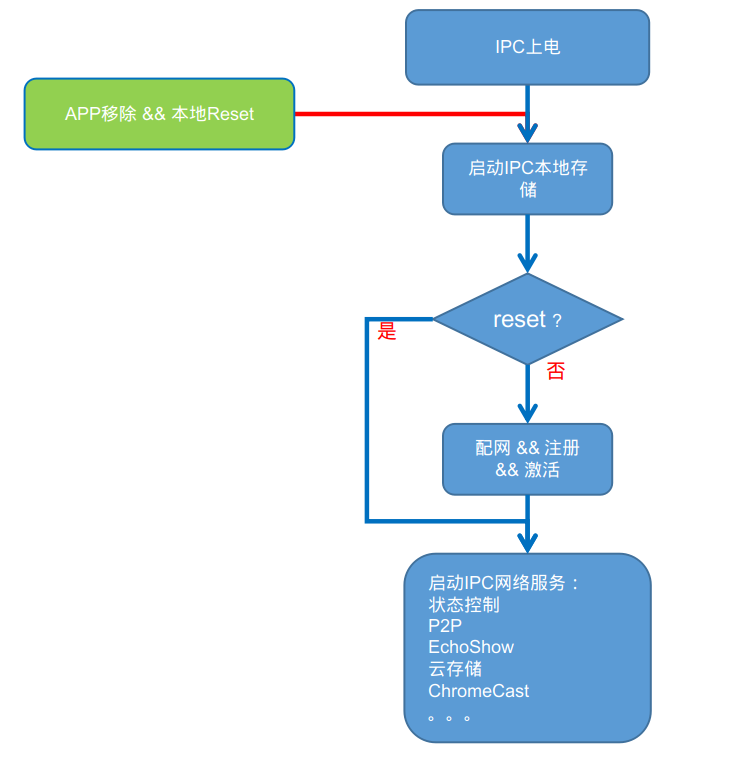
涂鸦TUYA\_IPC\_SDK使用说明 v3.0.X

# SDK概述

涂鸦TUYA\_IPC\_SDK运行在嵌入式Linux平台上，为用户提供APP交互、IPC本地存储、云存储、EchoShow、ChromeCast、P2P传输等服务。SDK整体框架如下：



# SDK工作流程介绍



当IPC启动后，IPC会首先读取本地的配置信息，识别和判断本地的状态，如果是第一次启用或者被重置，则进入WIFI配网流程，通过声音或者LED提示用户，同时用户使用涂鸦智能APP，使用【 AP模式/二维码配网/Smartcfg配网 】将WIFI信息和用户信息告知IPC，IPC连接WIFI后进行注册，激活，登录TUYA MQTT服务器等操作。整个流程完成后，IPC将最新的状态数据通过MQTT服务上报到云端和APP，同时开启P2P传输，Echoshow，云存储，ChromeCast等音视频服务。

细节澄清：

1. 配网方式可能会根据IPC SDK和APP的升级发生变动。
2. P2P传输，Echoshow，云存储，ChromeCast等音视频服务可能会根据IPC SDK的升级发生变动。

## 设备配网

由于IPC没有键盘和显示器，因此IPC的WIFI配置需要通过手机APP来实现。

目前有如下几种方式：

### AP模式

IPC建立一个本地AP热点，手机APP连接后通过局域网通讯方式将SSID和PASSWD传送给IPC。

### 二维码配网

APP生成一个二维码，里面包含SSID和PASSWD，IPC通过扫描二维码获取SSID和PASSWD。

### Smartcfg配网

APP通过一定的编码格式将SSID和PASSWD编码成特定的网络报文，并通过路由器在网络中发送，IPC通过网络抓包，识别并分析出里面包含的SSID和PASSWD。

## 注册 && 激活

设备注册指设备向涂鸦云注册设备信息，设备激活是指设备和用户APP上的账号进行绑定。设备只有正常的注册，激活后才能与APP、涂鸦云进行数据音视频交互。

在此过程中，设备需要向涂鸦云提供以下信息：

### PID/ProductKey

产品ID，用来标示某一类产品，同一种类型的IPC设备共享同一个产品ID。

当IPC产品需求确定后，涂鸦产品经理会根据需求提供产品ID或者用户通过涂鸦开发平台自行生成。

### UUID && AUTHKEY && P2PID

设备授权ID，用来标示某一台产品，每台IPC都需要唯一的授权ID。开发阶段涂鸦会提供部分授权ID进行开发和测试。

# SDK使用要求

## WIFI托管

在设备配网阶段，SDK需要根据配网方式对WIFI芯片进行配置，比如开启/关闭AP模式，开启/关闭Sniffer模式，设置/获取WIFI信道，抓取WIFI管理帧等，在设备正常工作后，SDK会使用SSID和PASSWD进行连接WIFI。为了防止操作冲突，嵌入式平台需要禁用掉外部其他的WIFI管理功能，并且提供TUYA\_IPC\_SDK操作WIFI芯片所需要的各类操作函数。

## 存储配置

TUYA\_IPC\_SDK运行时需要存储和管理自己的状态信息，因此需要嵌入式平台提供一个可读写且掉电不丢失的文件路径来存放配置文件。

# SDK使用说明

为了最大化减少嵌入式开发者使用SDK时的工作量，最快速度将设备接入涂鸦云平台，TUYA\_IPC\_SDK在基础API的基础上，封装了大量工具函数，并以源码的方式进行提供，开发者可以根据这些工具函数和源码非常快速的将设备接入涂鸦云，同时又能根据自身业务进行高度定制化，增加了开发者使用SDK的灵活性。

## 文件目录

├── build\_app.sh

├── demo\_linux\_tutk\_all

│   ├── include

│   │   ├── tuya\_ipc\_dp\_handler.h

。。。。。。。。。。。。

│   │   └── tuya\_linux\_wifi.h

│   └── user

│   ├── tuya\_ipc\_dp\_handler.c

│   ├── tuya\_ipc\_dp\_utils.c

│   ├── tuya\_ipc\_media\_handler.c

│   ├── tuya\_ipc\_media\_utils.c

│   ├── tuya\_ipc\_mgr\_handler.c

│   ├── tuya\_ipc\_mgr\_utils.c

│   ├── tuya\_linux\_wifi.c

│   └── user\_main.c

├── include

│   ├── cJSON.h

。。。。。。。。。。。。

│   └── tuya\_ring\_buffer.h

├── libs

│   ├── libAVAPIs.a

│   ├── libIOTCAPIs.a

│   ├── libtuya\_ipc.a

│   └── libtuya\_ipc.a.stripped

└── Makefile

为了方便开发者理解和使用SDK，SDK在函数定义和使用方式上均进行了大量注释，开发者可以根据注释和实例代码进行参考开发。

## 部分文件介绍

### build\_app.sh && Makefile

DEMO编译工具，修改Makefile第一行的交叉编译器路径到正确的位置，然后执行【 sh build\_app.sh demo\_linux\_tutk\_all 】，就可以生成对应的bin，在嵌入式平台下可以直接运行。

### \*.h

工具函数和SDK标准API，具体函数定义和返回值请参考代码注释。

### tuya\_ipc\_dp\_handler.c

IPC状态上报和命令下发实现接口，代码框架已经搭建完成，开发者根据自身的业务填补函数空白即可。

### tuya\_ipc\_dp\_utils.c

IPC状态上报和命令下发工具代码，开发者可以不用关心。

### tuya\_ipc\_media\_handler.c

IPC音视频信息注册和音频对讲实现接口，代码框架已经搭建完成，开发者根据自身的业务填补函数空白即可。

### tuya\_ipc\_media\_utils.c

IPC音视频工具代码，开发者可以不用关心。

### tuya\_ipc\_mgr\_handler.c

IPC固件管理接口，代码框架已经搭建完成，开发者根据自身的业务填补函数空白即可。

### tuya\_ipc\_mgr\_utils.c

IPC音视频工具代码，开发者可以不用关心。

### tuya\_linux\_wifi.c

WIFI配置接口，代码框架已经搭建完成，开发者根据自身的业务填补函数空白即可。

### user\_main.c

用户实例。

### \*.a

TUYA\_IPC\_SDK和P2P的SDK。

# 常见问题

1. SDK合入后如何修改编译脚本？

链接时加上 -ltuya\_ipc -lpthread -lAVAPIs -lIOTCAPIs

1. WIFI\_INIT\_DEBUG模式如何使用？

目前SDK是需要配网激活后才能正常使用，而配网一般都需要实现SmartCfg（快连）或者二维码方式获取SSID和Passwd后连接WIFI，如果用户需要并行开发或者配网有问题，同时开发环境可以直接连接互联网，那么可以使用DEBUG方式跳过配网，直接激活设备，token获取方式为：进入涂鸦APP，选择【智能摄像机】-【二维码配网】，使用第三方工具识别二维码，里面的”t”字段即为token。这里请注意token只有10分钟有效期。