

君正®

算法授权版本 2 使用说明书 V1.0

Date: 2019/12/16



北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

君正®

算法授权版本2使用说明书-V2.0

Copyright © Ingenic Semiconductor Co. Ltd 2019. All rights reserved.

Release history

Date	Revision	Change
Aug. 2019	1.0	First release
Dec. 2019	2.0	Add based count authority

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

合肥君正科技有限公司

地址: 安徽省合肥高新区望江西路 800 号创新园 C3 楼 9 层

电话: 86-10-82826661

传真: 86-10-82825845

Http: //www.ingenic.cn

目录

1 概述.....	3
1.1 授权条件.....	3
1.2 授权资源.....	3
1.3 授权模式.....	4
1.4 授权单位.....	4
1.5 授权方法.....	4
1.6 授权参数.....	4
1.7 支持的算法列表.....	4
2 算法授权流程.....	5
2.1 算法授权流程.....	5
2.2 算法授权注意事项.....	5
3 算法授权 API 使用方法.....	7
3.1 算法授权接口概述-头文件 iaac.h.....	7
3.2 算法授权错误码.....	7
3.3 算法授权 API 调用流程.....	8
3.4 自定义授权单位天数 IAAC_SetAuthIntervalDays.....	8

1 概述

ADK2.0 商业模式由三种算法授权模式和两种算法购买单位组成。

三种算法授权模式分为：

1) 串号包年/月授权模式：在所包年/月内可以对所有算法发放新的授权；当所包年/月到期后，不再发放新的授权；

2) 算法包年/月授权模式：在所包年/月内可以对该算法发放新的授权；当所包年/月到期后，自动切换到算法信用卡授权模式；

3) 算法信用卡授权模式：在信用额度天数或数量未用完前都可以对该算法发放新的授权；信用额度天数或数量用完后，不再对该算法发放新的授权，直到付费提升该算法的信用额度天数或信用额度数量；若该算法的信用额度天数或信用额度数量为 0，则相当于预付费模式。

两种算法购买单位分为：

1) 按使用算法的芯片个数购买，一次授权终生可激活，简称为按个数购买；

2) 按使用算法的总天数购买，一次授权可激活授权单位天数，简称按天数购买

注意：客户在购买时必须明确每一个芯片一次授权可激活的天数，即授权单位天数。客户可以通过调用 `IAAC_SetAuthIntervalDays` 来自定义实际每个芯片上每个算法每次授权可激活的天数，若不调用 `IAAC_SetAuthIntervalDays` 来自定义，则每次授权可激活的天数为购买时确定的授权单位天数。

在开发阶段没有商务介入的情况下，默认按算法信用卡授权模式，采用按个数购买 40 个。

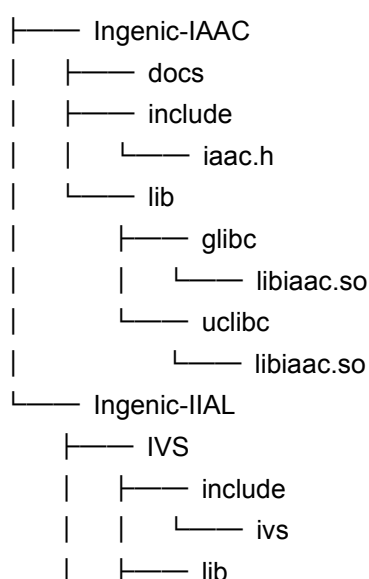
1.1 授权条件

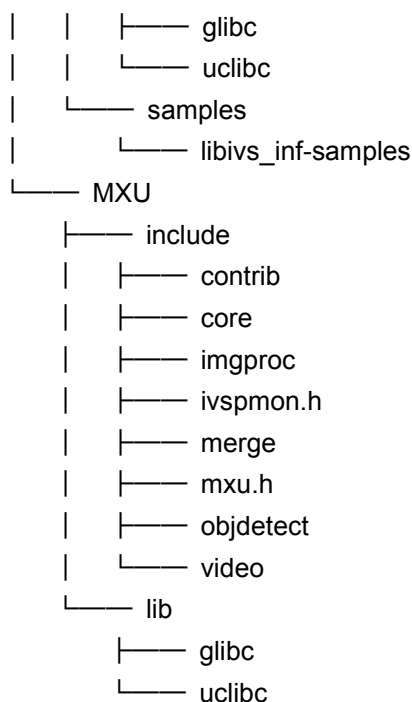
君正 T 和 X 系列芯片；

每次授权或激活必须可连接外网，检验方式如 ping www.baidu.com

1.2 授权资源

ADK2.0 相关资源，包括：Ingenic-IAAC 2.0 库和 Ingenic-IIAL 库，列表如下：





1.3 授权模式

串号包年/月授权：在所包年/月内可以获取任意算法的授权；

算法包年/月授权：在所包年/月内可以获取该算法的授权；

算法信用卡授权：在付费和信用额度天数内，可以获取该算法的授权；

1.4 授权单位

按个数授权；

按天数授权

1.5 授权方法

若未授权或已授权但授权已逾期，则会自动联网重新到 IAAS 获取新的授权；

1.6 授权参数

串号 SN

格式：由[0-9a-f]组成的 32 个 16 进制数字组成；

提供方式有两种：1) 使用君正提供的授权号，2) 提供授权串号给君正授权；

获取方式：与君正商务过程获取；

License 文件路径

必须可读可写；

每个算法占 72 个字节，请根据实际需求算法数量适当预留；

生成的 **License.txt** 不管什么情况下都不要删除，不要删除，不要删除

1.7 支持的算法列表

具体一直在变动，请参考发布的 ADK2.0 库 IVS 部分；

2 算法授权流程

2.1 算法授权流程

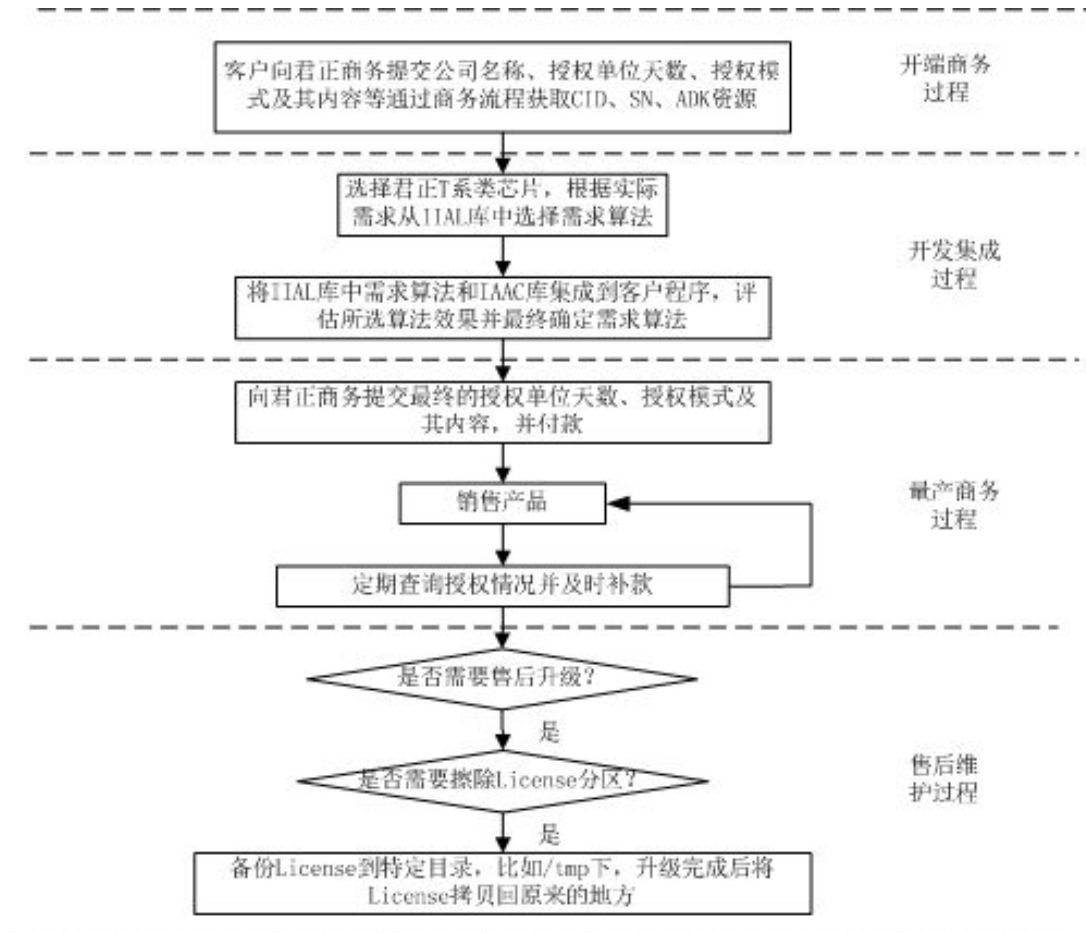


图 2-1：算法授权操作流程

2.2 算法授权注意事项

- 1) 商务过程公司信息尽量要全，并通过邮件的形式发送到君正商务；
- 2) SN 若需自定义，也需要在商务过程提交，SN 号由 32 个 16 进制字符组成；
- 3) 目前授权只支持君正 T 系列芯片；
- 4) 在开发阶段没有商务介入的情况下，默认按算法信用卡授权模式，采用按个数购买 40 个，若不够，请向君正商务或市场申请购买；
- 5) 量产过程中请合理选择授权模式及其内容以免频繁申请授权；
- 6) 请及时查询剩余授权时间，并及时反馈君正商务提供的需缴费信息，以免算法不能激活，影响客户体验；
- 7) 授权和激活时都必须连接外网；

8) License 存储区域要可读得写，不能存储在易失性区域，如/tmp，因相同的授权可最大重复 7 次，若超过这个次数，会导致授权不成功；

9) 每个算法 license 占用空间为 72 个 Byte，n 个算法累计占用 $n*72\text{Byte}$ ，请合理预留 license 存储空间；

3 算法授权 API 使用方法

请仔细阅读 `iaac.h` 头文件的说明。

3.1 算法授权接口概述-头文件 `iaac.h`

主要结构体 `IAACInfo` 和函数：

```
typedef struct {
    char    *license_path;
    int     cid;
    int     fid;
    char    *sn;
    int     (*send_and_recv)(IAACAuthInfo *authInfo);
} IAACInfo;

int IAAC_Init(const IAACInfo *info); //授权初始化函数
int IAAC_DeInit(void);              //授权反初始化函数
```

`ADKAuthInfo` 成员解释：

- 1) `license_path`: 许可证 `license` 文件路径名，以此文件路径名建立的文件必须允许可读可写，每个算法占用 72 个字节，若一个产品上需要激活 N 个算法，则该文件大小最小为 $N \times 72$ 个字节；
- 2) `cid`: 客户编号 CID，由君正提供，客户必须保密以防信息流失，ADK2.0 起该成员未使用，可不填写，有此成员是为了兼容 ADK1.0；
- 3) `fid`: 功能编号 FID，由君正提供，ADK2.0 起该成员未使用，可不填写，有此成员是为了兼容 ADK1.0；
- 4) `sn`: 激活序列号 SN，最大 32 个字节，唯一不重复，可以由君正提供，也可以由客户在产品生产前统一向君正提供，并由君正授权
- 5) `send_and_recv`: 若君正芯片可连接外网，则置为 `NULL`，否则你需要自己实现并赋值可以与外网交互数据的 `send_and_recv` 接口函数，你必须根据 `authInfo` 里提供的信息将数据发送出去，并接受 `need_recv_data_len` 大小的数据存到 `need_recv_data` 空间里。
- 6) `IAAC_Init`: 必须在所有算法调用 `IMP_IVS_CreateChannel` 函数之前调用，否则后者会调用不成功；
- 7) `IAAC_DeInit`: 必须在算有算法调用 `IMP_IVS_DestroyChannel` 函数后调用，否则 IVS 相关接口可能不成功或出错；

3.2 算法授权错误码

表 3-1: 授权错误码表

错误码	解释
0x80000001	IAACInfo 参数错误
0x80000002	License 文件路径不可读写
0x80000003	不支持的芯片类型
0x80000004	不支持的芯片标识
0x90000001	不能得到 license

0x90000002	错误的 license
0x90000003	算法未付费
0x90000004	算法库错误

3.3 算法授权 API 调用流程

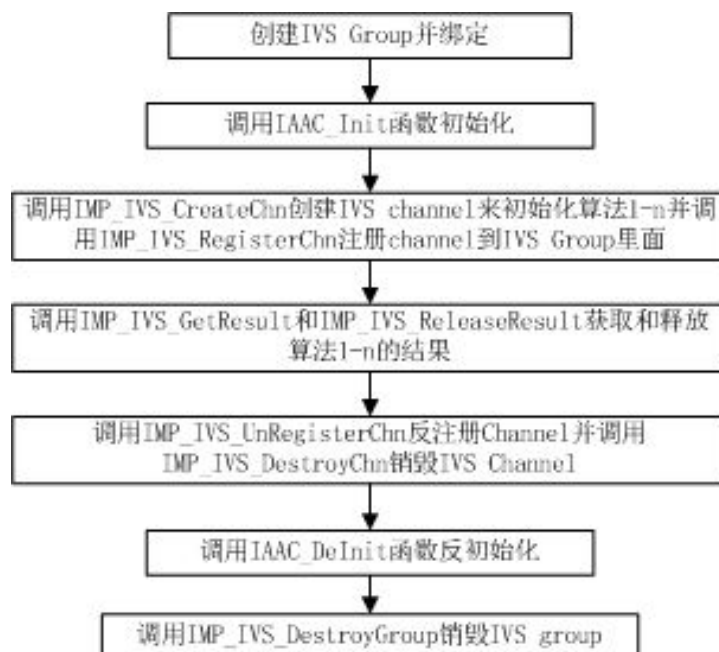


图 3-1: 授权 API 调用流程图

3.4 自定义授权单位天数 IAAC_SetAuthIntervalDays

若你想自定义授权单位天数，而不直接使用商务流程的授权单位天数，可以调用 `int IAAC_SetAuthIntervalDays(int days);`来自定义，但需注意以下几点：

- 1) 此接口必须放在每一个 `IMP_IVS_CreateChn` 函数之前调用；
- 2) 此接口设置的单位天数只有在需要授权的条件满足，即未授权过，或已授权但授权日期已逾期，才有效；
- 3) 此接口设置的单位天数有效期期为此接口调用后到 `IMP_IVS_CreateChn` 返回之前；
- 4) 若此接口不调用，或调用的 `days` 值为 0，则会直接使用商务流程获取到的单位授权天数。