OTP Check.dll 规范手册

**修改时间：2017年10月25日**

**版 本：Version 2.1**

**版本List**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **内容** | **编辑人** | **审核人** | **完成日期** | **备注** |
| Version 1.0 | 初始版本 | 刘剑平/陈志南 | 范敦贵 | 2017-3-23 | 欧菲所有 |
| Version 1.1 | 增加注册表卡控 | 刘剑平/陈志南 | 范敦贵 | 2017-5-8 | 欧菲所有 |
| Version 2.0 | 更新OTPCheck函数接口、变更配置文件名及新增测试项检测接口 | 刘剑平/陈志南 | 范敦贵 | 2017-8-3 | 欧菲所有 |
| Version 2.1 | 增加AEC接口 | 孔祥圆 | SW Team | 2017-10-25 | 欧菲所有 |

**目 录**

[OTP Check.dll 规范手册 1](#_Toc496705896)

[1 总纲 6](#_Toc496705897)

[1.1 OTPCheck规范的主要内容和结构安排 6](#_Toc496705898)

[1.2 OTPcheck文档 7](#_Toc496705899)

[1.3 OTPCheck API接口 7](#_Toc496705900)

[2 防呆 9](#_Toc496705901)

[2.1 机种防呆Check内容 9](#_Toc496705902)

[2.2 规格防呆 9](#_Toc496705903)

[2.3 其它防呆 9](#_Toc496705904)

[3 测试项接口 12](#_Toc496705905)

[3.1 测试项接口说明 12](#_Toc496705906)

[3.1.1 双摄项目 12](#_Toc496705907)

[3.1.2 非双摄项目 13](#_Toc496705908)

[3.2 各框架使用方法 13](#_Toc496705909)

[3.2.1 OTS3.0 13](#_Toc496705910)

[3.2.2 IsMedia（包含S30、S31、S22、SK10等） 14](#_Toc496705911)

[3.2.3 R3或四焦段 14](#_Toc496705912)

[3.2.4 其它架构 16](#_Toc496705913)

[4 Module Information Check 17](#_Toc496705914)

[4.1 Check模组基本信息 17](#_Toc496705915)

[4.2 卡控规范 17](#_Toc496705916)

[5 AF Check 19](#_Toc496705917)

[5.1 Check 模组AF值 19](#_Toc496705918)

[5.2 卡控规范 19](#_Toc496705919)

[6 AWB Check 20](#_Toc496705920)

[6.1 Check模组AWB值 20](#_Toc496705921)

[6.2 卡控方法 21](#_Toc496705922)

[7 LSC Check 22](#_Toc496705923)

[7.1 Check模组LSC值 22](#_Toc496705924)

[7.2 高通平台卡控方法 22](#_Toc496705925)

[7.2.1 新算法卡控 23](#_Toc496705926)

[7.2.2 锥形卡控 25](#_Toc496705927)

[7.2.3 国内标准卡控 26](#_Toc496705928)

[7.3 MTK平台 27](#_Toc496705929)

[7.4 LSI工具 27](#_Toc496705930)

[7.5 Hisi工具 27](#_Toc496705931)

[7.6 无平台（烧Sensor） 27](#_Toc496705932)

[7.7 卡控规范 27](#_Toc496705933)

[8 PDAF Check 28](#_Toc496705934)

[8.1 Check 模组PDAF值 28](#_Toc496705935)

[8.2 卡控规范 28](#_Toc496705936)

[8.2.1 高通PDAF 29](#_Toc496705937)

[8.2.2 MTK PDAF 30](#_Toc496705938)

[8.2.3 其它PDAF 30](#_Toc496705939)

[9 OC Check 31](#_Toc496705940)

[9.1 Check模组OC值 31](#_Toc496705941)

[9.2 卡控规范 31](#_Toc496705942)

[10 SFR Check 32](#_Toc496705943)

[10.1 Check模组SFR值 32](#_Toc496705944)

[10.2 卡控规范 32](#_Toc496705945)

[11 HDC Check 33](#_Toc496705946)

[11.1 Check 模组HDC值 33](#_Toc496705947)

[11.2 卡控规范 33](#_Toc496705948)

[12 AEC Check 34](#_Toc496705949)

[12.1 Check 模组AEC值 34](#_Toc496705950)

[12.2 卡控规范 34](#_Toc496705951)

[13 SPC Check 35](#_Toc496705952)

[13.1 Check 模组SPC值 35](#_Toc496705953)

[13.2 卡控规范 35](#_Toc496705954)

[14 CrossTalk Check 36](#_Toc496705955)

[14.1 Check 模组CrossTalk值 36](#_Toc496705956)

[14.2 卡控规范 36](#_Toc496705957)

[15 OIS Check 37](#_Toc496705958)

[15.1 Check 模组OIS值 37](#_Toc496705959)

[15.2 卡控规范 37](#_Toc496705960)

[**附录 1 SetOTPCheckSpec.ini中接口总清单** 38](#_Toc496705961)

# 

# 总纲

## OTPCheck规范的主要内容和结构安排

本规范介绍了OTPCheck dll调用方法和API接口函数的适用方法，同时根据目前厂内在做的项目，进行分类介绍。

第一章：编译和使用说明，介绍OTPCheck dll，SW编译和TE使用方法。

第二章：机种防呆，归纳目前采用的防呆方式。

第三章：测试项接口说明，介绍“SetOTPCheckSpec.ini”文档中的各个接口。

第四章：Module Information Check，采用一对一的匹配方式，当出现不匹配时将进行报错提示。

第五章：OC Check，针对烧录OC的模组将进行卡控。采用卡控数值上下限的方式进行。

第六章：AWB Check，卡控R,B,G相应Gain值的上下限，同时进行防呆卡控。

第七章：LSC Check，分平台用不同方式卡控。

第八章：PDAF Check，针对GainMap进行上下限和锥形卡控，针对DCCMap采用卡控上下限的方式。

第九章：SFR Check，卡控是否存在连续相同的值。

第十章：AF Check，卡控AF code值大小和差值，及注册表检测不允许多颗模组烧录相同的AF值。

第十一章：HDC Check，卡控是否存在连续相同的值。

第十二章：AEC Check，卡控是否存在连续相同的值。

第十三章：SPC Check，卡控SPC上下限值。

第十四章：CrossTalk Check，卡控是否存在连续相同的值。

第十五章：OIS Check，卡控是否存在连续相同的值。

第十六章：Others Check，根据具体的Guide 卡控。

第十七章：OTPCheck 管理

附录

## OTPcheck文档

OTPCheck基础文档有：OTPCheck.dll，OTPCheck.lib，OTPCheck.h，SetOTPCheckSpec.ini。

* SW调用文档

各项目组在首次调用OTPCheck.dll时需添加“OTPCheck.dll”、“OTPCheck.lib”、“OTPCheck.h”进行编译，若已经调用过，则无需重新编译，直接放在Release中使用。

* TE调用文档

TE需把“OTPCheck.dll”和“SetOTPCheckSpec.ini”放在在与EXE同一个文件夹下使用，如下图（以R3架构为例）。

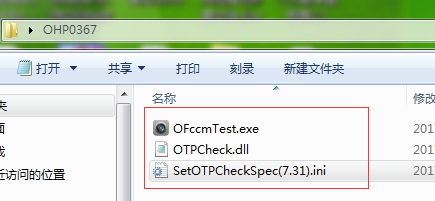


图2.1 OTPCheck放置

## OTPCheck API接口

* OTPcheck(char\* ErrorMsg,BYTE \*InputOTPData=NULL, int Flag=0)

参数1：ErrorMsg：当OTPCheck fail 时输出具体的错误信息，左边为读出来的数据，右边为设置的规格，建议开辟的Buffer应大于1024，如ErrorMsg[1024]；

参数2：InputOTPData：传入模组的OTP数据，需严格按照OTPMap/guide要求存储，依实际EEprom大小开辟Buffer内存，至少大于等于0x2000，如：InputOTPData[0x2000]；

**约定：R3和R、IsMedia、Hyvsion、一拖二、一拖八、一拖三等框架，必须传入数据，DLL内部不作数据读取。目前只针对OTS3.0框架在使用本DLL时，DLL内部会做数据读取动作。**

参数3：Flag：InputOTPData的标志位，当其大于0时表示有数据传入，对于双摄项目，约定Flag=1时表示传入主摄数据，Flag=2时表示传入副摄数据；

* 函数返回值

该函数通常返回三个值：

1. 返回TRUE，代表该模组测试结果正确，输出测试结果PASS信息。

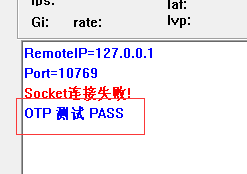


图2.2 打印测试结果

1. 返回FALSE，代表该模组测试失败，可能在某处无法满足规格，输出相应的错误位置，方便TE锁定错误和调试规格；左边是烧录的值，右边是设置的规格。

**但对于Checksum，规定：左边是烧录的值，右边计算出来的值。**

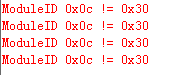


图2.3 打印Erromsg（ModuleID）

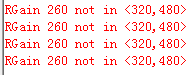


图2.4 打印Erromsg（RGain range）

1. 返回-1，表示该DLL与机种不匹配，输出报错信息信息。

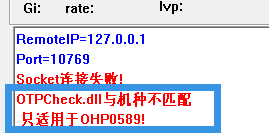


图2.5 打印Erromsg

* 其它接口

后续将根据大家需求封装其它函数接口。将会在OTPCheck.h中对新封装接口进行说明和调用方法。

# 防呆

## 机种防呆Check内容

查找SetOTPCheckSpec.ini中[Project1\_OTPCheck]及[Project2\_OTPCheck]下是否包含该机种名，若包含则会继续检测，若机种名不匹配，则会在打印出Error信息。

例如：机种名是OHP0589，但ProjectName1 != OHP0589

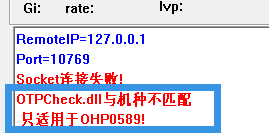


图3.1 Dialog框显示

## 规格防呆

主要是针对AWB进行防呆，防止RGain与BGain的值混淆。

例如：

* 烧录Golden值的，则必须满足，若RGain Golden 大于BGain Golden，则RGain必须大于BGain。反之亦然。
* 对于不烧录Golden值的，则增加一个RBGain值设定，如果烧录的RGain大于BGain，那么把RBGain设置成大于1000。反之则小于1000。

## 其它防呆

* 针对无法卡控上下限的数据，采用检测连续多少个数值是否相同的方式；
* 增加卡控连续多个模组烧录相同的值；

目前主要针对AF和AWB两种数值。

**这里以卡控AWB为例（AFcode值类同）：**

1、创建注册表（程式自动创建）

在注册表Software\\Ofilm\\OTPCheck 下找到机种名+OTPCheck，如：

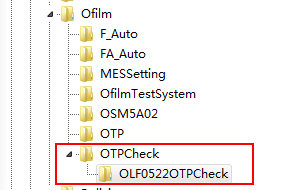


图3.2 注册表文件夹

在OLF0522OTPCheck中增加：

RGain的计数器：RGainEqualCount；

RGain的参照值：RGainEqualRef；

BGain的计数器：BGainEqualCount；

BGain的参照值：BGainEqualRef；

GGain的计数器：GGainEqualCount；

GGain的参照值：GGainEqualRef；

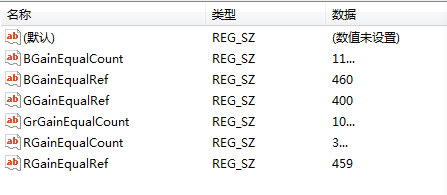


图3.2 注册表数值

在SetOTPCheckSpec.ini中增加一个EqualCountSpec 关键词接口。

2、测试流程

程式首先记录第一颗模组的AWB值，做为Ref值写入注册表中，当测试下一颗模组时，获取当前模组AWB值与上颗模组Ref值进行对比。

若相等，则Count将会加1，写入注册表中。当注册表中的Count值大于SetOTPCheckSpec.ini中EqualCountSpec设定的上限值，则程式会报错提醒。

若当前模组的AWB值与Ref值不相等，则Count清零，当前模组的AWB值做为新的Ref值写入注册表中。

**注：AWB中的各个值是独立累加，但规格统一，只要有一个超规格，则报错提醒。**

# 测试项接口

双摄项目使用[Project1\_OTPCheck]及[Project2\_OTPCheck]接口，非双摄项目仅使用[Project1\_OTPCheck]接口。

**注：接口名严禁私自修改，如需要增加，必须申请，升级该文档，并邮件通知Team成员。**

## 测试项接口说明

### 双摄项目

* **主摄**

[Project1\_OTPCheck]

1. ProjectName1：依据实际情况，填入项目1名称；
2. TestStation：依据Test Flow填入测试站位名称，目前仅支持:AF、OTP、PDAF、AEC、标定、终检；

**·**OTP： 目前OTP站位主要烧录ModuleInfomation、AWB、LSC等数据；

**·**标定：目前有虹软标定、Hisi标定等双摄标定；

**·**终检：烧录完所有数据的Check站位均当做终检，例如Image终检、OQC等；

1. SaveOTPCheckData：等于“1”表示只保存当前一颗模组的测试数据，保存路径为exe同等目录下，有两种格式（ReadOTPData1.ini、ReadOTPData1.bin）；等于“0”表示不保存测试数据；
2. EqualCountSpec：用于AWB及AF等数据，通过读写注册表方式，卡控连续多颗模组是否存在相同的值，若存在则NG；
3. HisiMasterFlag：华为Hisi专用，有Hisi Master模组时将此接口设置为“1”，没有则设置为“0”。

* **副摄**

[Project2\_OTPCheck]

1.ProjectName2：依据实际情况，填入项目2名称；

2.TestStation：同主摄；

3.SaveOTPCheckData：等于“1”表示只保存当前一颗模组的测试数据，保存路径为exe同等目录下，有两种格式（ReadOTPData2.ini、ReadOTPData2.bin）；等于“0”表示不保存测试数据；

4.EqualCountSpec：同主摄；

5.HisiMasterFlag：同主摄；

### 非双摄项目

非双摄项目则只使用[Project1\_OTPCheck]接口：

[Project1\_OTPCheck]

1.ProjectName1：依据实际情况，填入项目名称；

2.TestStation：同上；

3.SaveOTPCheckData：同上；

4.EqualCountSpec：同上；

5.HisiMasterFlag：同上；

## 各框架使用方法

### OTS3.0

1、OTS3.0架构程序界面示例如下：



图4.1 OTS 3.0框架图

2、架构调用OTPCheck.dll：

**·**OTPCheck.dll驱动VCM；

**·**OTPCheck.dll读取SensorID；

**·**使用OTPCheck.dll读取数据，读取完后保存单颗模组数据并check 每个地址烧录的数据。

### IsMedia（包含S30、S31、S22、SK10等）

* 1. IsMedia架构程序界面示例如下：

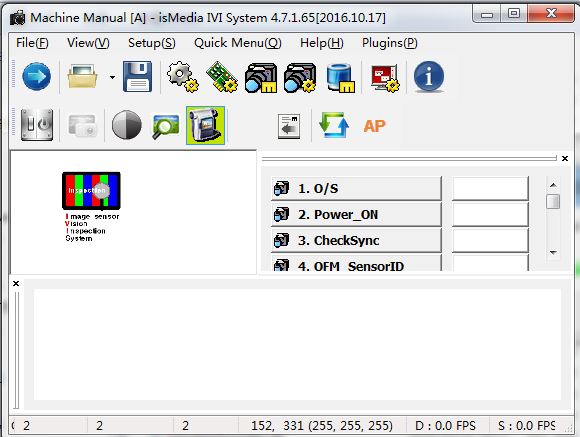


图4.2 IsMedia框架图

2.架构调用OTPCheck.dll：

通过IsMedia架构读取数据，读取完后通过Buffer方式传入OTPCheck.dll，并通过Flag判断为主摄或副摄数据（Flag > 0有数据传入，Flag = 1表示传入主摄或单射数据，Flag = 2表示传入副摄数据），OTPCheck.dll接收数据后进行保存，且按照SetOTPCheckSpec.ini中设置的规格进行检测。

### R3或四焦段

1、R3或四焦段架构程序界面如下：



图4.3 RX框架图

2、架构调用RX\_OTPCheck.dll及OTPCheck.dll：

**·**RX\_OTPCheck.dll调用OTPCheck.dll；

**·**RX\_OTPCheck.dll有Dialog，其中可选择使用R3、四焦段架构读取数据，；

**·**通过RX架构读取数据，读取完后通过Buffer方式传入OTPCheck.dll，并通过Flag判断为主摄或副摄数据（Flag > 0有数据传入，Flag = 1表示传入主摄或单射数据，Flag = 2表示传入副摄数据），OTPCheck.dll接收数据后进行保存，且按照SetOTPCheckSpec.ini中设置的规格进行Check。

3、RX\_OTPCheck.dll Dialog界面及参数说明：

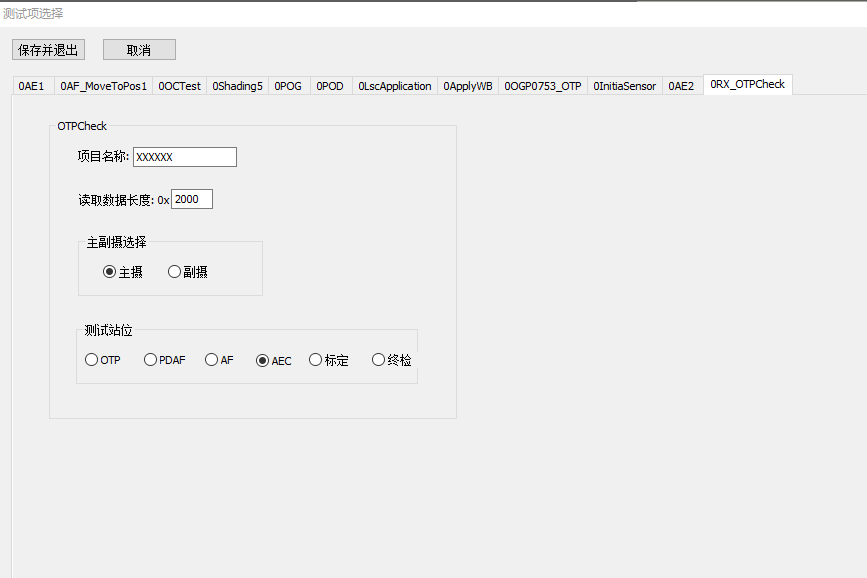


图4.3 RX\_OTPCheck Dialog

* 双摄项目

1. 项目名称：依据模组实际情况，填入主摄或副摄项目名称；
2. 读取数据长度：设置读取数据的长度；
3. 主副摄选择：依据模组实际情况，选择为主摄或副摄。选择“主摄”则程序会读取主摄数据，使用主摄规格来进行检测，反之读取副摄数据使用副摄规格进行检测；
4. 测试站位：依据实际使用情况选择测试站位。目前仅支持:AF、OTP、PDAF、AEC、标定、终检；

* 非双摄项目

1、项目名称：填入项目名称。

2、读取数据长度：设置读取数据的长度。

3、主副摄选择：请只选择主摄。

4、测试站位：依据实际使用情况选择测试站位。目前仅支持:OTP、AF、PDAF、AEC、标定、终检等。

### 其它架构

参照IsMedia框架的适用方式。

# Module Information Check

## Check模组基本信息

1. 基本信息烧录标志位（ModulInfoFlag：0x01）

2.模组编号（ModuleCode：0x05）

3.模组厂编号（SupplierCode：0x07）

4.镜头编号（LensID：0x02）

5.马达编号（VCMID：0x0F）

6.驱动芯片编号（DriverIcID：0x01）

7.传感器名称编号（SensorNameID：0x0E）

8.模组版本（ModuleVersion：0x09）

9.软件版本（SoftVersion：0x03）

10.AWB或LSC站位（AWBLSCStation：0x11）

11.PDAF站位（PDAFStation：0x0B）

12.AF站位（AFStation：0x32）

13.寄存器编号（EEPROMID：0x01）

14.Fnumber编号（FnumberID：0x01）

15.模组ID（ModuleID：0x07）

16.LSC工具版本（LSCToolVersion：0x11）

17.PDAF工具版本（PDAFToolVersion：0x0B）

18.烧录色温编号（ColorTemperatureID：0x32）

19.第一个像素（FirstPixel：0x06）

20.反射或翻转（MirrorFlip：0x07）

21.光源标志（LightSourceFlag：0x18）

22.AF或FF标志（AForFF\_Flag：0x1A）

23.IR编号（IRBGID：0x01）

24.平台编号（PlatformID：0x01）

25.双摄标志（DualCameraFlag：0x01）

26.产品烧录日期（年/月/日）

27.Check Sum 检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini中设置相应接口，TE填入实际固定值，程序内部进行Check。

* Check固定值：根据Guide要求设定相应固定值，烧录值须与设定值一致，满足则Pass，反之则Fail；
* Check时间信息：Year（2017-2022）、Month（1-12）、Day（1-31）、Hour（0-23）、Minute（0-59）、Second（0-59），满足则Pass，反之则Fail；
* Check CheckSum：依据Guide中的计算公式，程序内部计算并与相应位置烧录值做对比，一致则Pass，反之则Fail。

# AF Check

## Check 模组AF值

1.AF烧录标志位（AFFlag：0x01）

1. AF方向（AFCalDirection: 0x02）
2. AFCode1000cm最大值（MaxAFCode1000cm：400）
3. AFCode1000cm最小值（MinAFCode1000cm：300）
4. AFCode500cm最大值（MaxAFCode500cm：400）
5. AFCode500cm最小值（MinAFCode500cm：300）
6. AFCode10cm最大值（MaxAFCode10cm：400）
7. AFCode10cm最小值（MinAFCode10cm：300）
8. AFCode8cm最大值（MaxAFCode8cm：400）
9. AFCode8cm最小值（MinAFCode8cm：300）
10. AFCode7cm最大值（MaxAFCode7cm：400）
11. AFCode7cm最小值（MinAFCode7cm：300）
12. AFCode差值上限（MaxCodeDiff：200）
13. AFCode差值下限（MinCodeDiff：100）
14. 起始Code值上限（MaxStartCode：500）
15. 起始Code值下限（MinStartCode：300）
16. Check sum检测

## 卡控规范

1、依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

* 1. 注册表卡控多颗模组烧录相同的AF值，参见第三章AWB值注册表卡控方法

2、Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；

3、Check非固定值：根据模组烧录值，拉取CPK，设置相应的卡控范围，若在设定范围内，则Pass，否则Faill。

4、Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置实际烧录值做对比，若一致则Pass，反之则Fail。

# AWB Check

## Check模组AWB值

1. AWB标志位（AWBFlag：0x01）
2. RGain最大值（MaxRGain：TBD）
3. RGain最小值（MinRGain：TBD）
4. BGain最大值（MaxBGain：TBD）
5. BGain最小值（MinBGain：TBD）
6. GGGain最大值（MaxGGGain：TBD）
7. GGGain最小值（MinGGGain：TBD）
8. RBGain最大值（MaxRBGain：TBD）
9. RBGain最小值（MinRBGain：TBD）
10. RG Golden值（Typical\_RG：TBD）
11. BG Golden值（Typical\_BG：TBD）
12. GG Golden值（Typical\_GG：TBD）
13. R最大值（MaxRValue：TBD）
14. R最小值（MaxRValue：TBD）
15. B最大值（MaxRValue：TBD）
16. B最小值（MaxRValue：TBD）
17. Gr最大值（MaxGrValue：TBD）
18. Gr最小值（MaxGrValue：TBD）
19. Gb最大值（MaxGbValue：TBD）
20. Gb最小值（MaxGbValue：TBD）
21. R Golden值（Typical\_R：TBD）
22. B Golden值（Typical\_B：TBD）
23. Gr Golden值（Typical\_Gr：TBD）
24. Gb Golden值（Typical\_Gb：TBD）
25. ISO最大值（MaxISO：TBD）
26. ISO最小值（MinISO：TBD）
27. Check Sum检测

## 卡控方法

依据Guide要求在SetOTPCheckSpec.ini中设置相应接口，TE填入实际固定值及设置其他数据上下限范围，程序内部进行Check。

* 1. 注册表卡控多颗模组烧录相同的AWB值，参见3.3。
  2. Check非固定值：根据模组CPK，设置上下限卡控范围，数据满足范围则Pass，反之则Fail；
  3. RBGain值检测
     + 获取RBGain值

若模组未烧录RG/BG/GG Golden值，则通过RGain/BGain值\*1000得到RBGain值。

* 卡控思想

若实际RBGain值大于1000，说明该项目模组RGain > BGain，则设置MinRBGain > 1000接口,MaxRBGain依据模组CPK设置，当出现某颗模组烧录的RGain < BGain时，则RBGain < 1000,通过MinRBGain、MaxRBGain上下限范围就能将该NG模组卡出；

若实际RBGain值小于1000，说明该项目模组RGain < BGain，则设置MaxRBGain < 1000接口,MinRBGain依据模组CPK设置，当出现某颗模组烧录的RGain > BGain时，则RBGain > 1000,通过MinRBGain、MaxRBGain上下限范围也能将该NG模组卡出；

* 1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
  2. Check CheckSum：依据Guide中的计算公式，程序内部计算并与实际烧录值做对比，若一致则Pass，反之则Fail。

# LSC Check

## Check模组LSC值

1. LSC烧录标志位（LSCFlag：0x01）
2. LSC数据大小（LSCSize：0x074C）
3. 每行相邻递增、递减差值上限（ToleranceValueError：25）
4. 每通道总错误数上限（ToleranceCountError：5）
5. R通道LSC最大值（LSC\_R\_Max：1023）
6. R通道LSC最小值（LSC\_R\_Min：200）
7. Gr通道LSC最大值（LSC\_Gr\_Max：1023）
8. Gr通道LSC最小值（LSC\_Gr\_Min：200）
9. Gb通道LSC最大值（LSC\_Gb\_Max：1023）
10. Gb通道LSC最小值（LSC\_Gb\_Min：200）
11. B通道LSC最大值（LSC\_B\_Max：1023）
12. B通道LSC最小值（LSC\_B\_Min：200）
13. 与Standrd.ini对位差值上限(LSCDistanceSpec：10）
14. 中心9个点中允许出现最大值的个数上限（CenterValueEqualcount：5）
15. 每行允许递增、递减的误差值上限（Curvediff：15）
16. 每行错误数上限（ErrorCount：2）
17. 每通道总错误数上限（TotalErrorCount：6）
18. 第1,2,16,17列与中心9列比值下限（edg\_center\_LimitRatio：0.4）
19. 第1,2,16,17列与中心9列比值上限（edg\_center\_HighRatio：0.7）
20. 对称列差值上限（MaxSymColDiff：10）
21. 对称行差值上限（MaxSymRowDiff：15）
22. Check sum检测

## 高通平台卡控方法

目前高通LSC卡控方法存在三种：

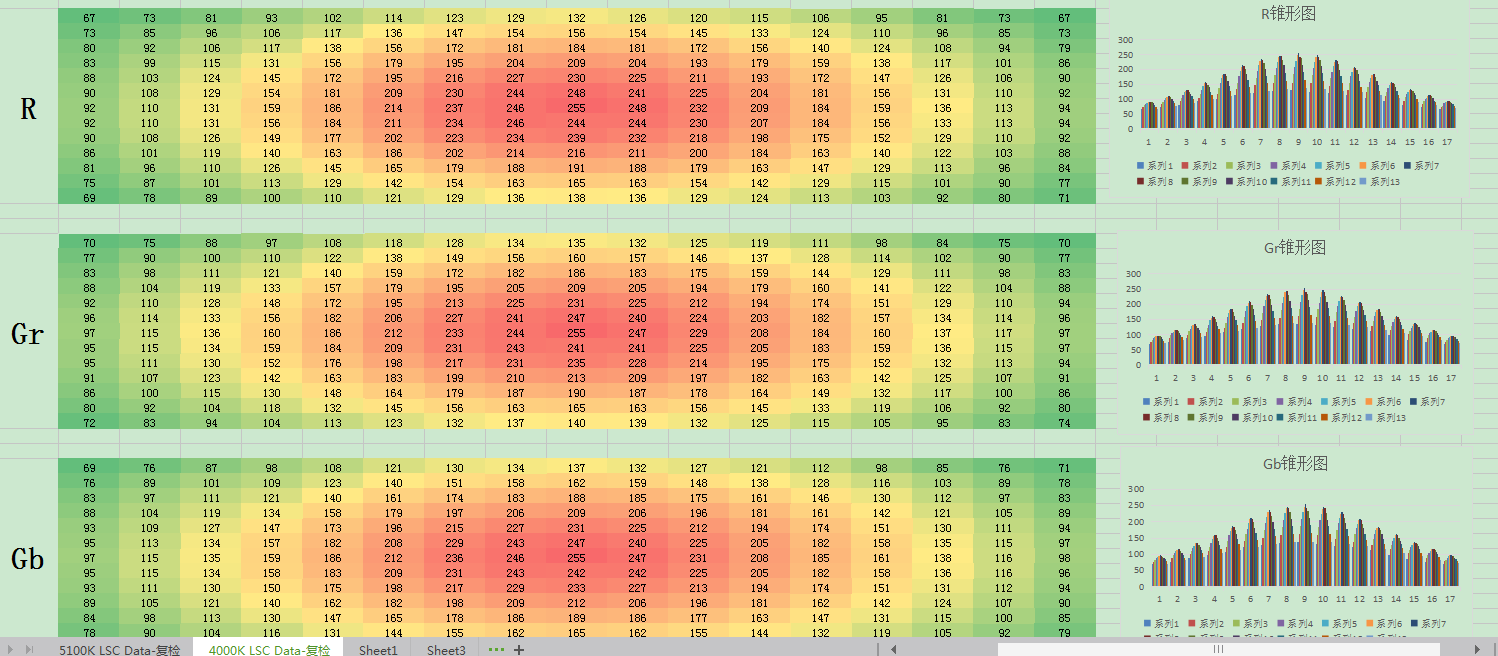
①．新算法卡控；

②．锥形卡控；

③．国内标准卡控。

设置接口LSCCurveCheck，等于“1”表示采用新算法及国内标准卡控，等于“2”表示采用锥形及国内标准卡控，等于非“1”、“2”表示仅采用国内标准卡控。

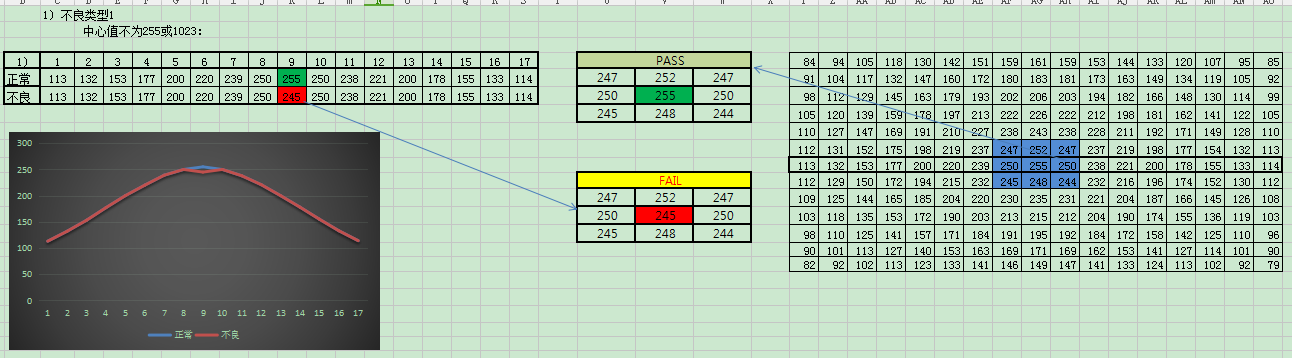
高通LSC数据规律：



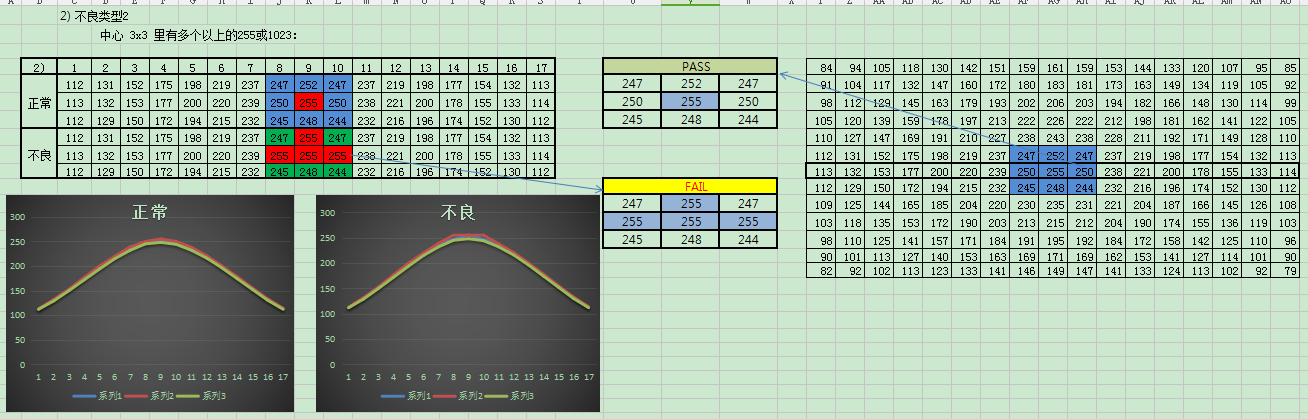
### 新算法卡控

1、中心最大值由OTP站计算LSC时采用的Ratio值（如255或1023）决定，具体参照Guide，在中心3\*3 范围内必有最大值且不允许存在多个最大值。

Ex：中心9个点内不存在255或1023，则Test Fail。



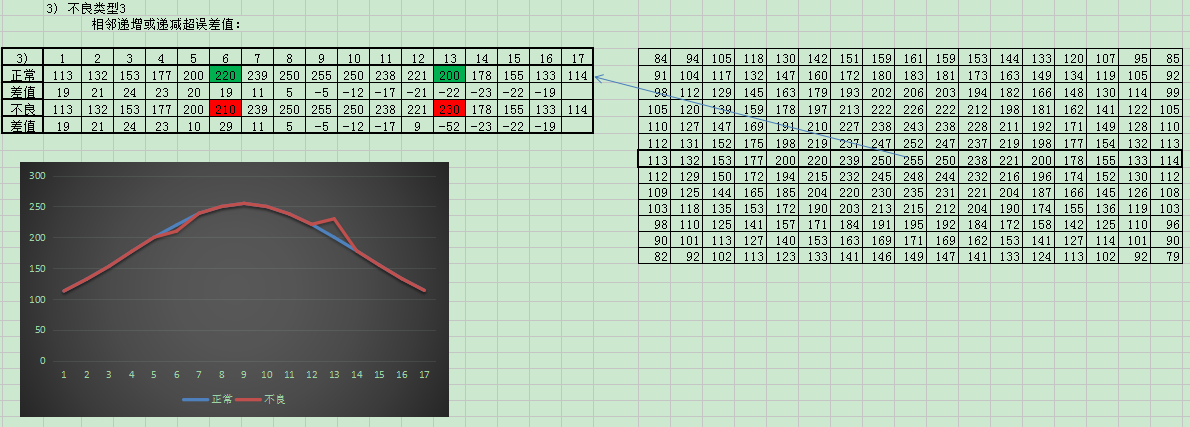
Ex：设置接口CenterValueEqualcount = 3，若中心9个点内存在超过3个255或1023值，则Test Fail。



2、每行相邻的递增或递减误差值，不予许超过一个上限值。

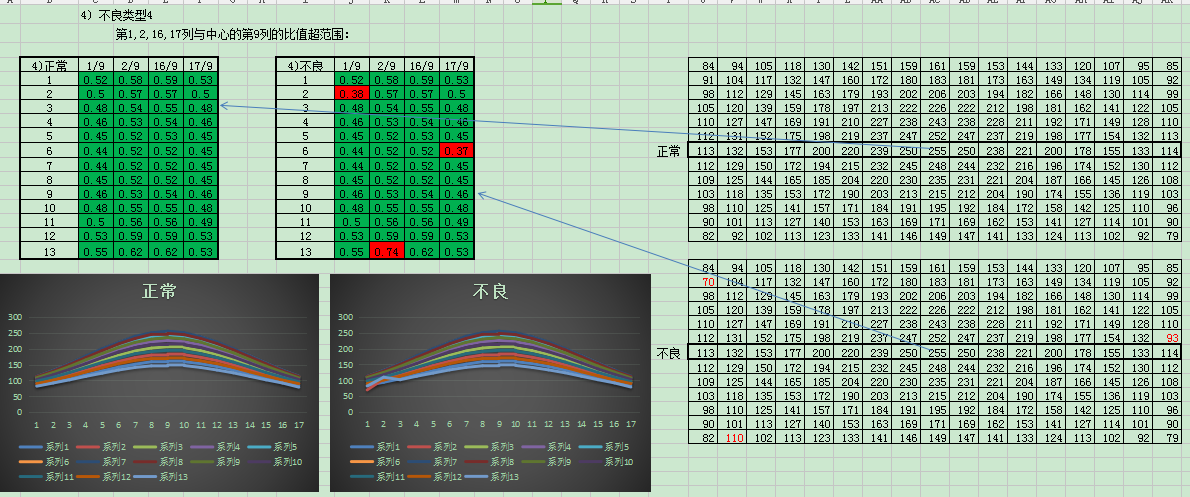
误差：只有当应递增（递减）时却不满足递增（递减）趋势时，才算做误差。

Ex：设置接口Curvediff = 25，允许每行相邻的递增或递减误差值为25，当误差超过Curvediff值时，则ErrorCount +1，当ErrorCount值超过spec值，则Test Fail，还设置每个Channel的阀值接口TotalErrorCount = 5，若每个Channel“错误数”> TotalErrorCount，则Test Fail。



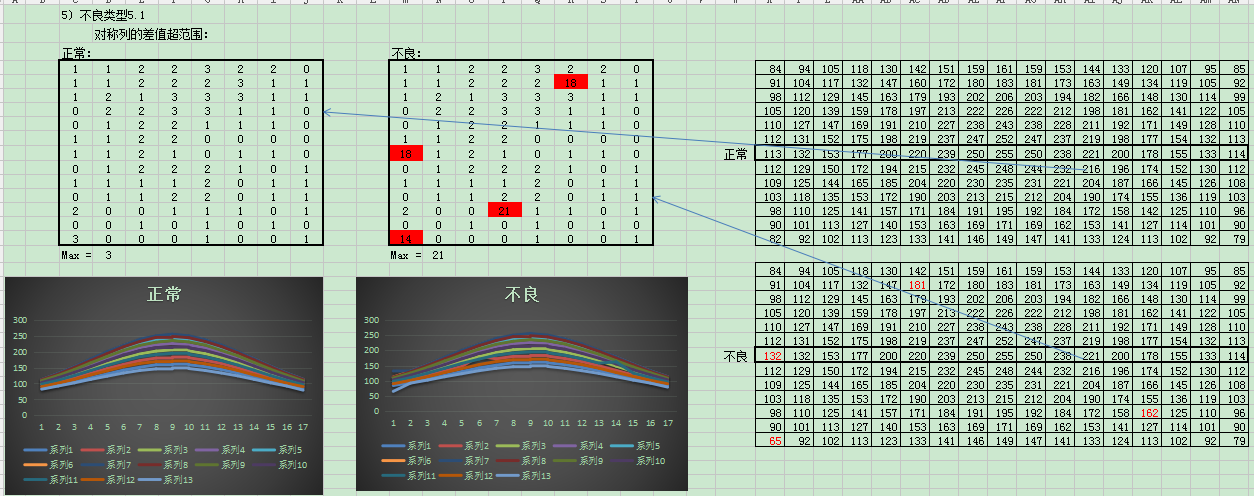
3、第1,2,16,17列与中心的第9列的比值，在一定范围内。

Ex：设置接口edg\_center\_LimitRatio = 0.4、edg\_center\_HighRatio = 0.7，若1,2,16,17列与9列比值任意一个超出0.4~0.7范围，则Test Fail。



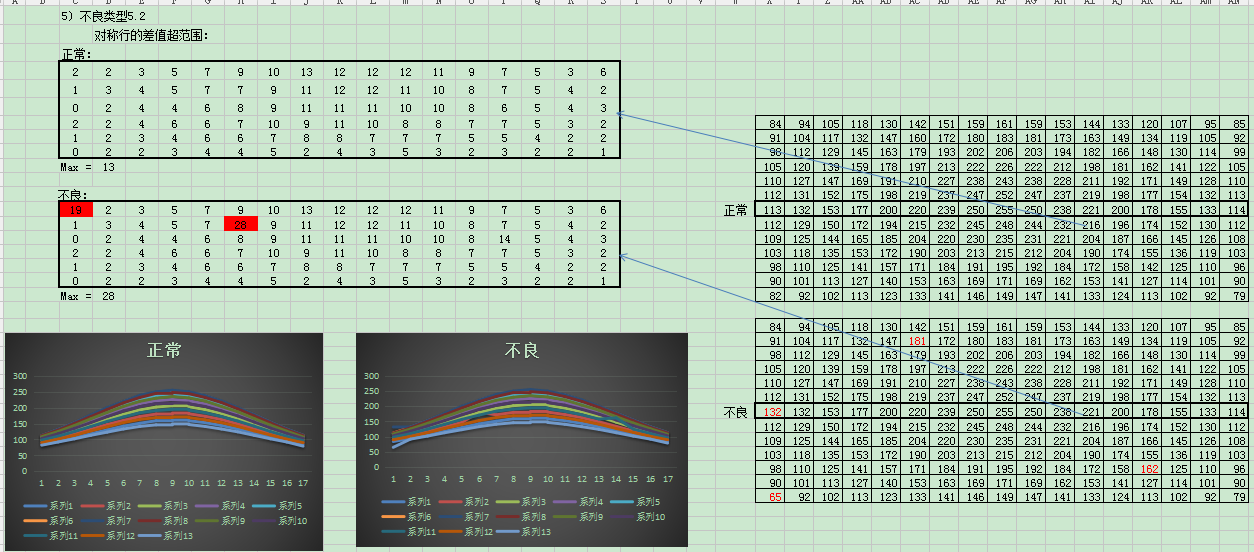
4、对称列差值，在一定范围内。

Ex：设置接口MaxSymColDiff = 10，若对称列做差的绝对值超过10，则Test Fail。



5、对称行差值，在一定范围内。

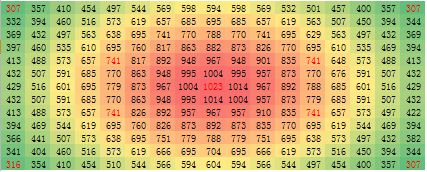
Ex：设置接口MaxSymRowDiff = 15，若对称行做差的绝对值超过15，则Test Fail。



6、卡控0.5视场、1.0视场的均匀性（初步按<10%以内进行管控）。

对于同一个视场，最大值与最小值的差值，再与四个值的平均值做比值。比如下图

Ratio = (（316-307）\*4/(316+307+307+307) )\*100%= 2.91%



另：为方便调设新算法卡控规格，程序设有保存上述表格的功能。

将SaveLSCALLDataEn置为“1”，即可保存所有模组数据，**但由于NG时会打印大量Log存在程序卡死风险，建议只在收集数据时开启，生产时请关闭；**若只收集OK模组数据，可将SaveLSCOKDataEn置为“1”。

单颗模组表格路径为：D：\\“机种名”\\“时间（年月日）”\\FuseID\_LSC.xls

Ex：D：\\OHA0476\\20170512\\00303137380200201D19\_LSC.xls

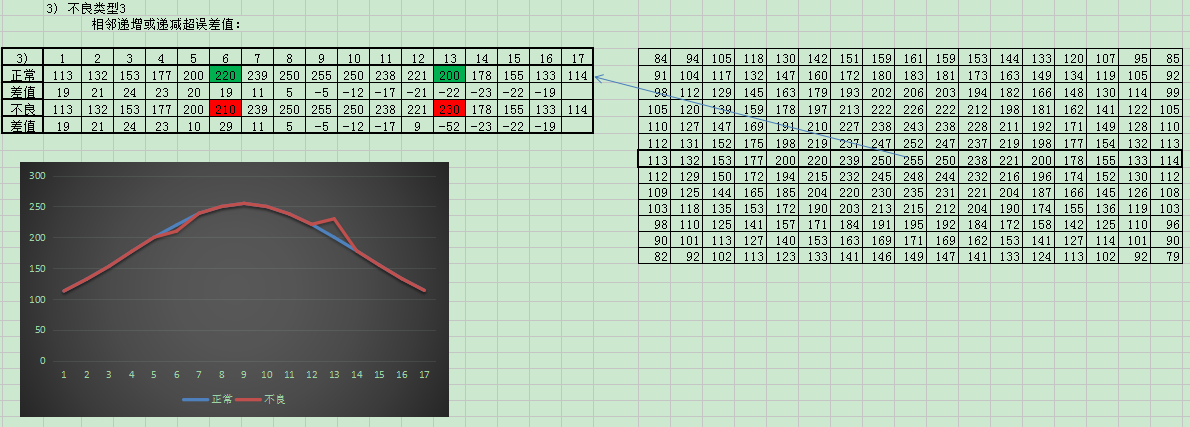
汇总表格路径为：D：\\“机种名”\\“时间（年月日）”\\AnalyzeSum\_LSC.xls

Ex：D：\\OHA0476\\20170512\\AnalyzeSum\_LSC.xls

### 锥形卡控

1.每行相邻的递增或递减差值，不予许超过一个上限值。

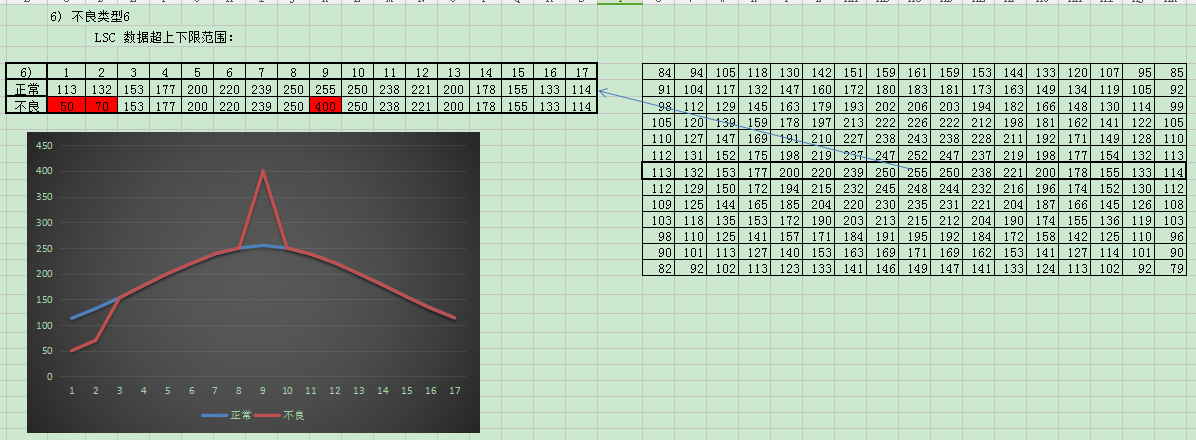
Ex：设置接口ToleranceValueError = 25，若每行相邻的递增或递减差值超过25，则“错误数”+1，再设置每个Channel的阀值接口ToleranceCountError = 5，若每个Channel“错误数”> ToleranceCountError，则Test Fail。



### 国内标准卡控

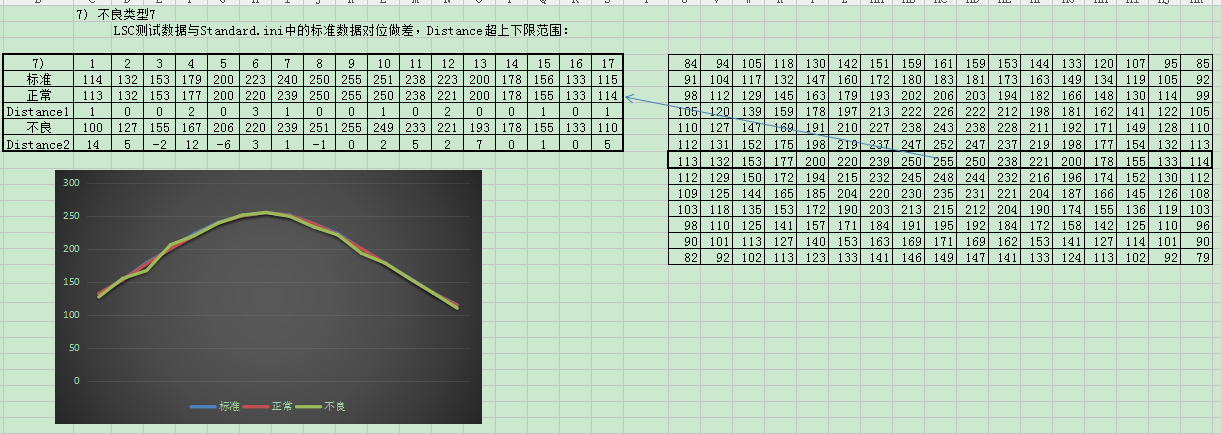
1. LSC R/Gr/Gb/B 四个Channel 的数据卡控上下限；

Ex：设置接口 LSC\_R/Gr/Gb/B\_Max = 255，LSC\_R/Gr/Gb/B\_Min = 75，若LSC R/Gr/Gb/B四个Channel任意一个数据超过75~255的卡控范围，则Test Fail。



2. LSC测试数据与Standard.ini中的标准数据对位做差，不允许超过一个上限值；

Ex：设置接口LSCDistanceSpec = 10，若LSC测试数据与Standard.ini中的标准数据对位做差的值若超过10，则Test Fail。



## MTK平台

数据无规律，采用不能存在多个连续相同的值进行卡控。

Ex：设置EqualCountSpec = 5，当MTK平台产生的LSC数据中有超过5个连续相同的值时，则Test Fail。

## LSI工具

数据无规律，采用不能存在多个连续相同的值进行卡控。

Ex：设置EqualCountSpec = 5，当LSI工具产生的LSC数据中有超过5个连续相同的值时，则Test Fail。

## Hisi工具

数据无规律，采用不能存在多个连续相同的值进行卡控。

Ex：设置EqualCountSpec = 5，当Hisi工具产生的LSC数据中有超过5个连续相同的值时，则Test Fail。

## 无平台（烧Sensor）

数据无规律，采用不能存在多个连续相同的值进行卡控。

Ex：设置EqualCountSpec = 5，当烧Sensor产生的LSC数据中有超过5个连续相同的值时，则Test Fail。

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini中设置相应接口，TE填入实际固定值及设置其他数据上下限范围，程序内部进行Check。

1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
2. Check非固定值：根据模组CPK，设置上下限卡控范围，数据满足范围则Pass，反之则Fail；

* Check CheckSum：依据Guide中的计算公式，程序内部计算并与实际烧录值做对比，若一致则Pass，反之则Fail。

# PDAF Check

## Check 模组PDAF值

1. PDAF烧录标志位（PDAFFlag：0x01）
2. PDAFStep1烧录标志位（PDAFStep1Flag：0x01）
3. PDAFStep1大小（PDAFStep1Size：0x06）
4. PDAFStep2烧录标志位（PDAFStep2Flag：0x01）
5. PDAFStep2大小（PDAFStep2Size：0x08）
6. PDAF工具版本编号（PDAFVersionNum：0x82）
7. GainMap宽度（GainMapWidth：0x11）
8. GainMap高度（GainMapHeight：0x0D）
9. GainMap数据最大值（MaxGainMap：800）
10. GainMap数据最小值（MinGainMap：400）
11. DCC烧录标志位（DCCFlag：0x01）
12. DCC版本编号（DCCFormat：0x01）
13. DCCMap宽度（DCCMapWidth：0x08）
14. DCCMap高度（DCCMapHeight：0x06）
15. DCC数据最大值（MaxDCC：1000）
16. DCC数据最小值（MinDCC：400）
17. Check sum检测

## 卡控规范

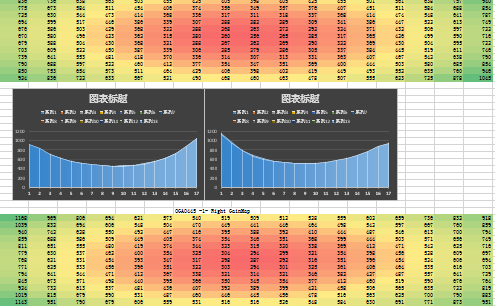
依据Guide要求在SetOTPCheckSpec.ini中设置相应接口，TE填入实际固定值及设置其他数据上下限范围，程序内部进行Check。

* 1. Check固定值：根据Guide要求设定相应固定值，烧录值须与设定值一致，满足则Pass，反之则Fail；
  2. Check非固定值：根据模组CPK，设置DCC和GainMap的上下限范围，数据满足范围则Pass，反之则Fail；
  3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与实际烧录值做对比，一致则Pass，反之则Fail。

### 高通PDAF

* GainMAP

对于Left/Right GainMap目前新增以下方式卡控：



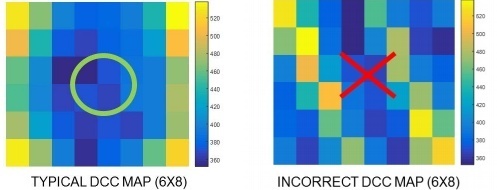
1. Right与Left Gainmap的曲面图均应该是平滑、连续过渡的，不存在突变，满足与LSC类似的曲线规律。
2. Right GainMap数值左高右低（根据sensor走）。
3. Left GainMap数值左低右高；
4. Right与Left Gainmap的平均数值差异应小于20%.

* DCC数据

目前采用卡控上下限的方式，而在实际生产时，需多前面几颗烧录的DCC数据进行肉眼判断是否满足以下规律：

1、数据长度都为48（8x6）；

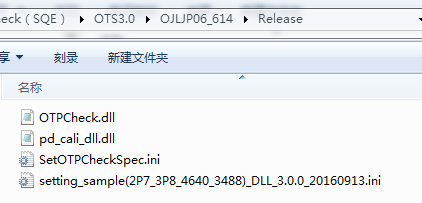
2、数据生成的3D图像是光滑、连续的，不存在突变，如下图：



### MTK PDAF

该PDAF采用MTK自带的工具进行卡控，会封装到OTPCheck DLL中。

这里以OJLJP06为例，对此需把以下文件放到与EXE相同的文件夹中。



### 其它PDAF

对于OV和Sony的PDAF，目前没有具体的卡控方式，均采用卡控上下限的方式进行卡控。

# OC Check

## Check模组OC值

1. OC标志位（OCFlag：0x01）
2. OCX最大值（MaxOCX：TBD）
3. OCX最小值（MinOCX：TBD）
4. OCY最大值（MaxOCY：TBD）
5. OCY最小值（MinOCY：TBD）
6. ShiftX最大值（MaxShiftX：TBD）
7. ShiftX最小值（MinShiftX：TBD）
8. ShiftY最大值（MaxShiftY：TBD）
9. ShiftY最小值（MinShiftY：TBD）
10. TiltX最大值（MaxTiltX：TBD）
11. TiltX最小值（MinTiltX：TBD）
12. TiltY最大值（MaxTiltY：TBD）
13. TiltY最小值（MinTiltY：TBD）
14. Rotation最大值（MaxRotation：TBD）
15. Rotation最小值（MinRotation：TBD）
16. Check Sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini中设置相应接口，TE填入实际固定值及设置其他数据上下限范围，程序内部进行Check。

* Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
* Check非固定值：根据模组CPK，设置上下限卡控范围，数据满足范围则Pass，反之则Fail；
* Check CheckSum：依据Guide中的计算公式，程序内部计算并与实际烧录值做对比，若一致则Pass，反之则Fail。

# SFR Check

## Check模组SFR值

1、Check sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
2. Check非固定值：根据模组实际情况，卡控相邻地址位相等数值的个数，该个数由程序内部写死，新项目有可能拉接口出来设置；
3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置烧录值做对比，一致则Pass，反之则Fail。

# HDC Check

## Check 模组HDC值

1、Check sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
2. Check非固定值：根据模组实际情况，卡控相邻地址位相等数值的个数，该个数由程序内部写死，新项目有可能拉接口出来设置；
3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置烧录值做对比，一致则Pass，反之则Fail。

# AEC Check

## Check 模组AEC值

1.Check sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
2. Check非固定值：根据模组实际情况，卡控相邻地址位相等数值的个数，该个数由程序内部写死，新项目有可能拉接口出来设置；
3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置烧录值做对比，一致则Pass，反之则Fail。

# SPC Check

## Check 模组SPC值

1. SPC最大值（MaxSPC：500）
2. SPC最小值（MinSPC：200）
3. Check sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

* 1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
  2. Check非固定值：根据模组烧录值，拉取CPK，设置相应的卡控范围，若在设定范围内，则Pass，否则Faill。
  3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置实际烧录值做对比，若一致则Pass，反之则Fail。

# CrossTalk Check

## Check 模组CrossTalk值

1.CrossTalk烧录标志位（CrossTalkFlag：0x01）

2.Check sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
2. Check非固定值：根据模组实际情况，卡控相邻地址位相等数值的个数，该个数由程序内部写死，新项目有可能拉接口出来设置；
3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置实际烧录值做对比，若一致则Pass，反之则Fail。

# OIS Check

## Check 模组OIS值

1.OISWideTag（OISWideTag：0x10）

2.OIS版本编号（OISVersion：0x01）

3.Check sum检测

## 卡控规范

依据Guide要求在SetOTPCheckSpec.ini相应接口设置固定值或卡控范围，以便程序内部进行Check。

1. Check固定值：根据Guide要求设定相应固定值，烧录值必须与设定值一致，满足则Pass，反之则Fail；
2. Check非固定值：根据模组实际情况，卡控相邻地址位相等数值的个数，该个数由程序内部写死；
3. Check CheckSum：依据Guide中的计算公式，程序内部计算并与checksum位置实际烧录值做对比，若一致则Pass，反之则Fail。

**对于SetOTPCheckSpec.ini中的关键字，不允许做任何增加或删减。**

# **附录 1 SetOTPCheckSpec.ini中接口总清单**

