

CONFIDENTIAL B

MEDIATEK

Introduction to Flash AWB 2.5 Tuning

2015/08



Support Chip

- MT6595
- MT6752
- MT6735
- MT6580
- MT6755

Revision

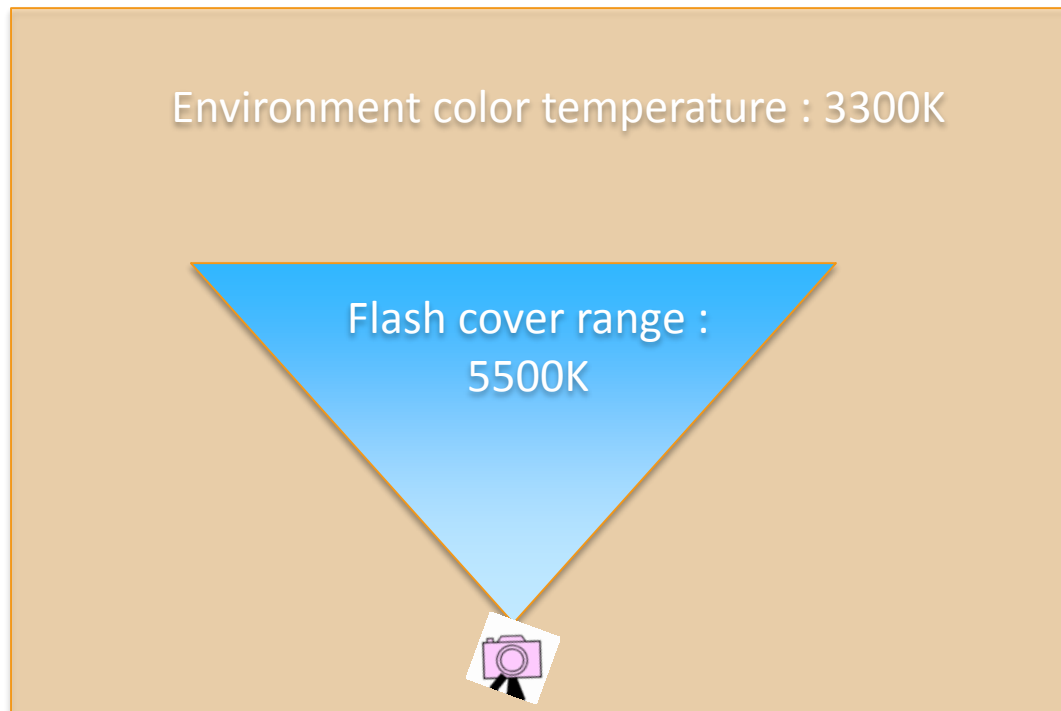
Date	version	description
2015/8/9	V1.0	First version
2015/8/26	V1.1	Modify location map UI

Outline

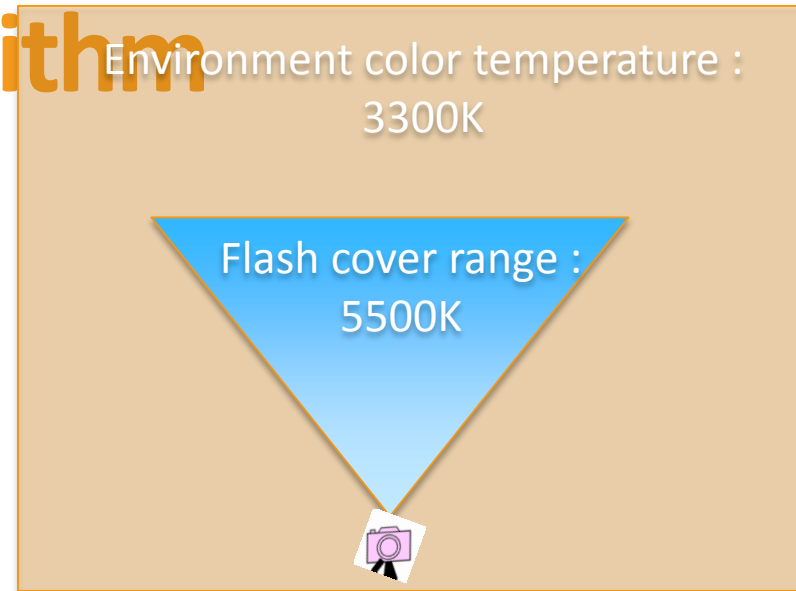
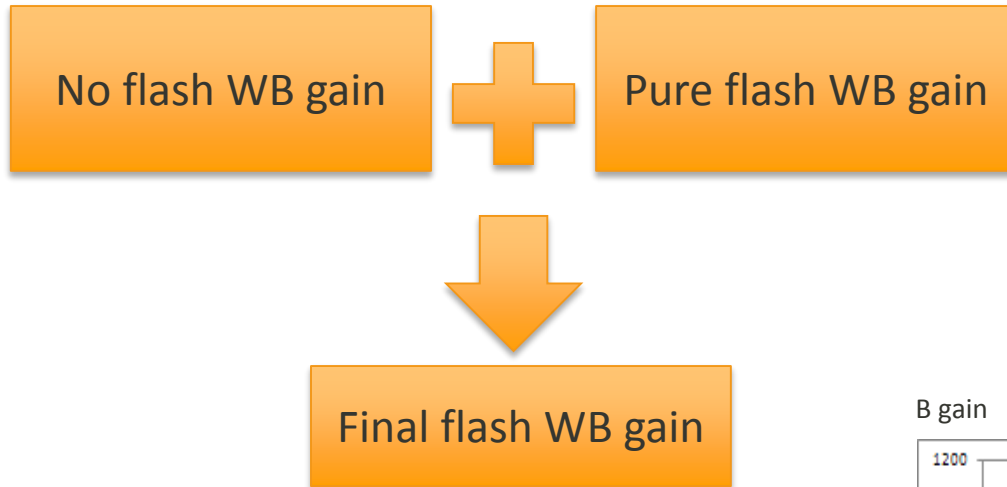
- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- Log analysis
- Issue analysis flow

What is Flash AWB

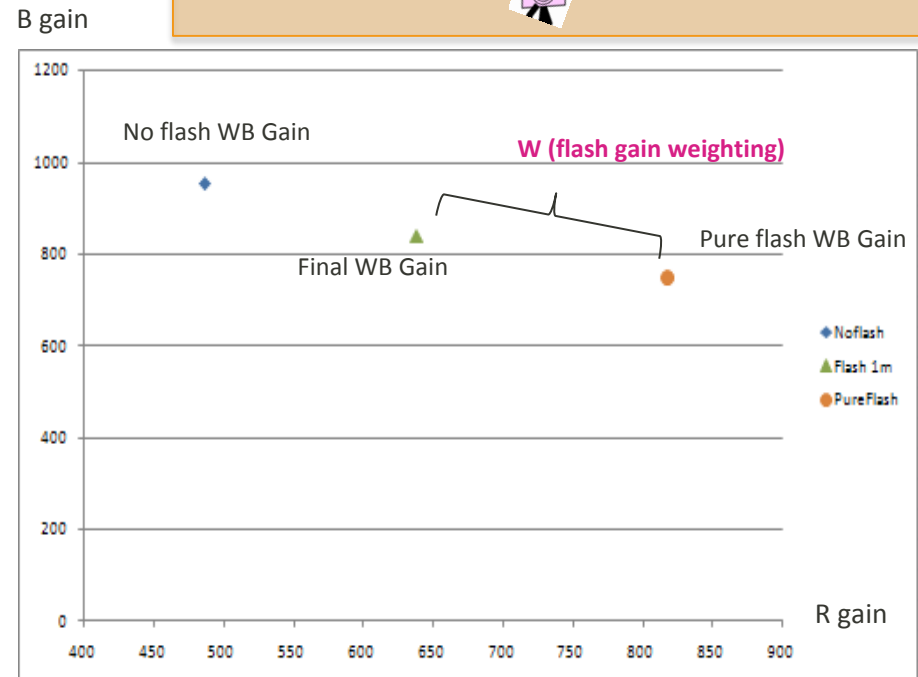
There are two kinds of light source from this picture. If we want to do AWB in this picture, we should make the AWB gain cover environment light and flash light.



Basic Algorithm



The final color temperature with flash firing is mixed by Environment light (No flash WB gain) and flash light (pure flash WB gain).



Basic Algorithm

■ Flash Weighting Table

- Flash Gain Weighting is the combination of the following three weightings:

- Foreground Weighting
- Location Weighting
- Luma. Weighting

There are 3 weighting table to let tuner to customize their preference.

Detail : p12 ~ p14



Outline

- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- Log analysis
- Issue analysis flow

Terms

- Flash Gain Weighting
 - Definition: Ratio of Flash Gain in Final Gain
 - Usage: Calculating the ratio of Flash Gain and AWB Gain mixing together
- Flash Weighting Table
 - Definition: Weighting of each block in the 24×18 blocks of an image
 - Usage: Calculating Flash Gain Weighting by multiplying it by Flash Map

Terms

■ Flash Map

- Definition: Map of different flash strength corresponding to different reflection strength of a scene (24×18)
- Usage: Calculating Flash Gain Weighting by multiplying it by Flash Weighting Table

■ Foreground Weighting Table

- Definition: The weighting value of the foreground of an image
- Usage: Calculating the weighting for balancing the foreground and background

Terms

■ Location Weighting Table

- Definition: The weighting value of the central part of an image
- Usage: Increasing the white balance weighting of the central part

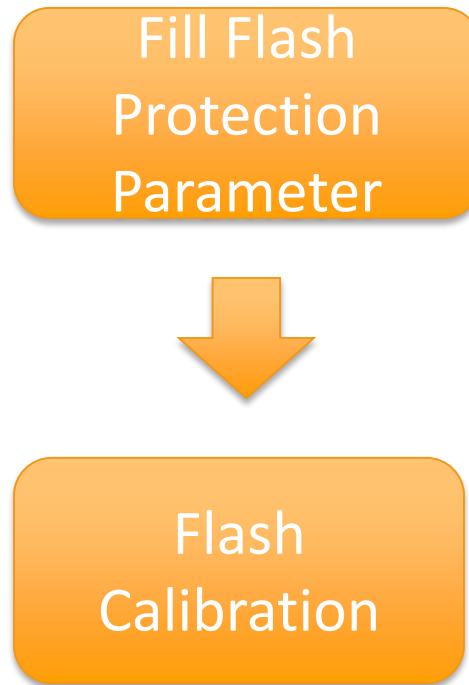
■ Luma. Weighting Table

- Definition: The weighting value of the brightness
- Usage: Reducing the impact of relatively darker part in an image on the result (the statistic of a part that is too dark is easily affected by noise and is less accurate).

Outline

- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- Log analysis
- Issue analysis flow

Calibration flow



Set protection mechanism

- The protection mechanism settings can be different based on different flash strengths.
 - `para.dutyNum = 10; //total 10 duties`
 - `para.coolDelayPara.tabNum = 3; //number of sets`
 - `para.coolDelayPara.tabId[0]=0; //setting of 0th flash segment`
 - `para.coolDelayPara.tabId[1]=4; //setting of 4th flash segment`
 - `para.coolDelayPara.tabId[2]=32; //setting of 32rd flash segment`
 - `para.coolDelayPara.coolingTM[0]=0; //no rest after main flash`
 - `para.coolDelayPara.coolingTM[1]=2; //time of main flash resting for 2x`
 - `para.coolDelayPara.coolingTM[2]=5; //time of main flash resting for 5x`
 - `para.coolDelayPara.timOutMs[0]=ENUM_FLASH_TIME_NO_TIME_OUT;`
 - `para.coolDelayPara.timOutMs[1]=100; //flash time cannot be longer than 100ms`
 - `para.coolDelayPara.timOutMs[2]=500; //flash time cannot be longer than 500ms`
- With interpolation

Set protection mechanism

Example

- Setting

```
para.coolDelayPara.tabNum = 3; //number of sets
para.coolDelayPara.tabId[0]=0; //setting of 0th flash segment
para.coolDelayPara.tabId[1]=4; //setting of 4th flash segment
para.coolDelayPara.tabId[2]=32; //setting of 32rd flash segment
para.coolDelayPara.coolingTM[0]=0; //no rest after main flash
para.coolDelayPara.coolingTM[1]=2; //time of main flash resting for 2x
para.coolDelayPara.coolingTM[2]=5; //time of main flash resting for 5x
para.coolDelayPara.timOutMs[0]=ENUM_FLASH_TIME_NO_TIME_OUT;
para.coolDelayPara.timOutMs[1]=100; //flash time cannot be longer than 100ms
para.coolDelayPara.timOutMs[2]=500; //flash time cannot be longer than 500ms
```

- Flash = 4

- ⇒ The next flash cannot appear after the flash rests for at least “2x flash time”. (Refer to settings [1] above.)
- ⇒ The flash time cannot be longer than 100ms. (Refer to settings [1] above.)

- Flash = 2 (Interpolation will operate itself if the setting is not included in the settings above.)

- ⇒ The next flash cannot appear after the flash rests for at least “1x flash time”. (The result of 0 and 2 interpolated)
- ⇒ The flash time cannot be longer than 50ms. (The result of 0 and 100ms interpolated)

Set protection mechanism

Longest Main Flash Exposure Time is Limited

- `para.maxCapExpTimeUs = 100000;`
- If the main flash is longer than this time, the exposure time will be automatically adjusted to prevent the flash time from exceeding it.

Set protection mechanism

How to Calculate Protection Time

Input:

1. The longest exposure time
2. Main flash current
3. Datasheet

For 1A LED, ~300ms is suggested.

Example:

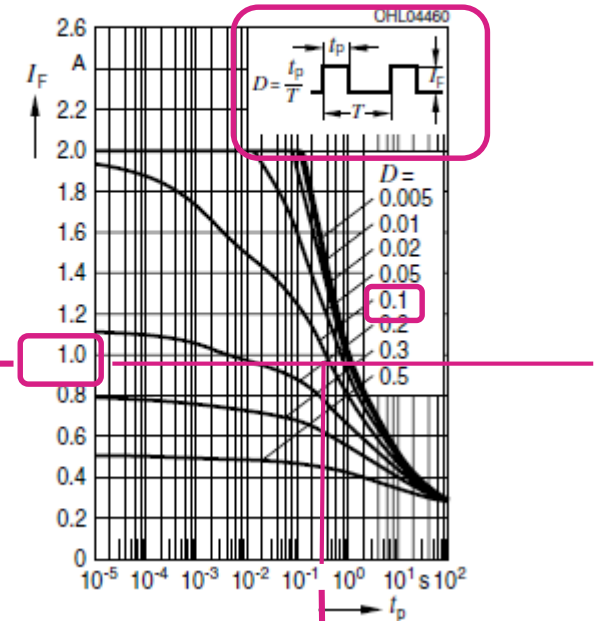
- Longest exposure time = $1/10 = 100\text{ms}$
- Main flash current = 1A
- $D = t_p/T$

\Rightarrow Longest flash on time = longest exposure t
 $= 300\text{ms} = 3 \cdot 10^{-1} = t_p$

$\Rightarrow D = t_p/T \sim 0.1$

\Rightarrow Cooling time = $(T - t_p)/t_p$
 $= 1/D - 1 \sim 9$

Permissible Pulse Handling Capability
 Zulässige Impulsbelastbarkeit $I_F = f(t_p)$
 $R_{thJA\text{ el}} = 150\text{K/W}$, $T_A = 25^\circ\text{C}$, still air, FR4



Flash AE/AWB calibration



Step	Point	Description
1	Environment prepare	Setup calibration environment : In dark room, Gray card put in front of camera in 10cm ~ 15cm
2	ADB commend	Connect USB to your computer, and then send ADB commend through USB. “adb shell setprop z.flash_ratio 1”
3	Pre-check	Lunch camera APP and press capture key in low light environment to check ADB commend work well or not. check : 1. Flash will fire several times with different duty 2. There are two files (fwb_code.txt, eng_code.txt)will save in the root folder.
4	Calibration	If pre-check ok, go to the correct environment as step 1, then Lunch camera APP and press capture key.
5	Get result	Go to root folder to get fwb_code.txt, eng_code.txt
6	Code gen	Merge fwb_code.txt into amera_flash_awb_para_[sensor].h (next page)

Outline

- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- Log analysis
- Issue analysis flow

How to Tune Parameters

- Parameter files
 - flashawb_tuning_custom.cpp
 - Path: alps\mediatek\custom\project\hal\camera\camera
 - Usage: Enables/Disables Flash AWB v2.1
 - camera_flash_awb_para_\$sensor.cpp
 - Path: alps\mediatek\custom\project\hal\imgsensor\sensor
 - Usage: Flash AWB preference (NVRAM)

How to Tune Parameters

- Parameter file: flashawb_tuning_custom.cpp
 - // Flash AWB v2.1 flag
 - Enables/Disables Flash AWB v2.1 (True: v2.1, False: v1.0)

```
MBOOL  
isFlashAWBv2Enabled()  
{  
    return MTRUE;  
}
```

How to Tune Parameters

- Parameter file: camera_flash_awb_para_\$sensor.cpp
 - // Foreground / Background Percentage Ratio
 - Controls the percentage of foreground and background
 - The bigger the foreground percentage, the more part in an image judged as foreground.
 - The bigger the background percentage, the more part in an image judged as background.

• Foreground percentage + background percentage can over 100

```
// Flash AWB tuning parameter
{
    9, //foreground percentage
    95, //background percentage
    2, //FgPercentage_Th1
    5, //FgPercentage_Th2
    10, //FgPercentage_Th3
    15, //FgPercentage_Th4
    250, //FgPercentage_Th1_Val
    250, //FgPercentage_Th2_Val
    250, //FgPercentage_Th3_Val
    250, //FgPercentage_Th4_Val
}
```

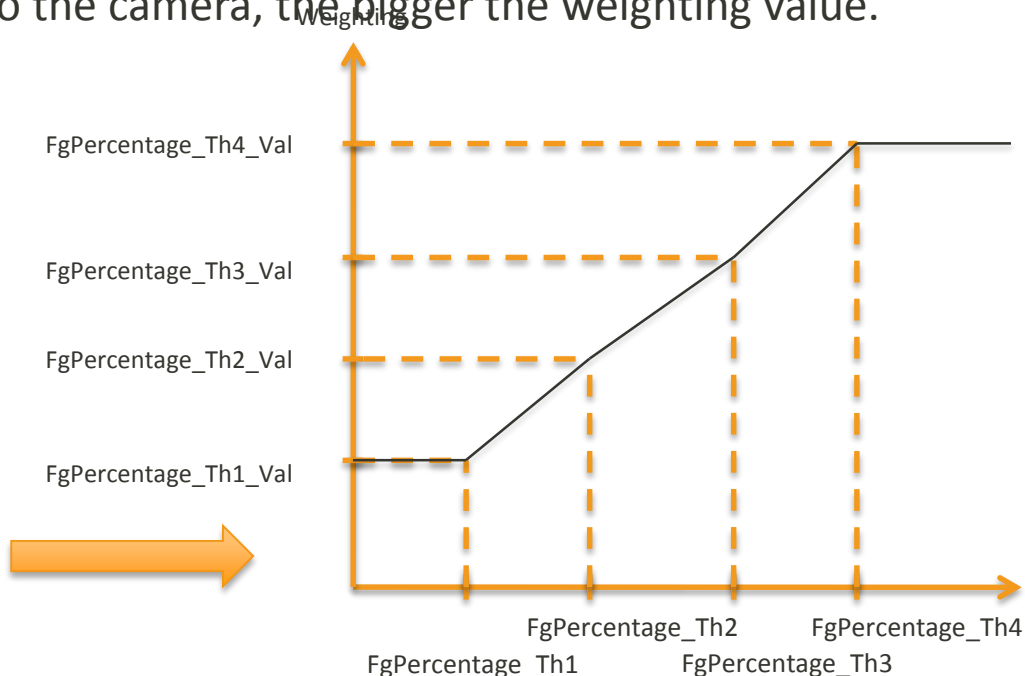
0<=Foreground Percentage<50
50<Background Percentage<=100

How to Tune Parameters

Foreground weighting

- Parameter file: camera_flash_awb_para_\$(sensor).cpp
 - // Foreground Weighting Table
 - You can get Foreground Weighting Table by looking up the table.
 - The closer the object is to the camera, the bigger the weighting value.

```
// Flash AWB tuning parameter
{
    9, //foreground percentage
    95, //background percentage
    2, //FgPercentage_Th1
    5, //FgPercentage_Th2
    10, //FgPercentage_Th3
    15, //FgPercentage_Th4
    250, //FgPercentage_Th1_Val
    250, //FgPercentage_Th2_Val
    250, //FgPercentage_Th3_Val
    250, //FgPercentage_Th4_Val
}
```



$FgPercentage_Th1 \leq FgPercentage_Th2 \leq FgPercentage_Th3 \leq FgPercentage_Th4$

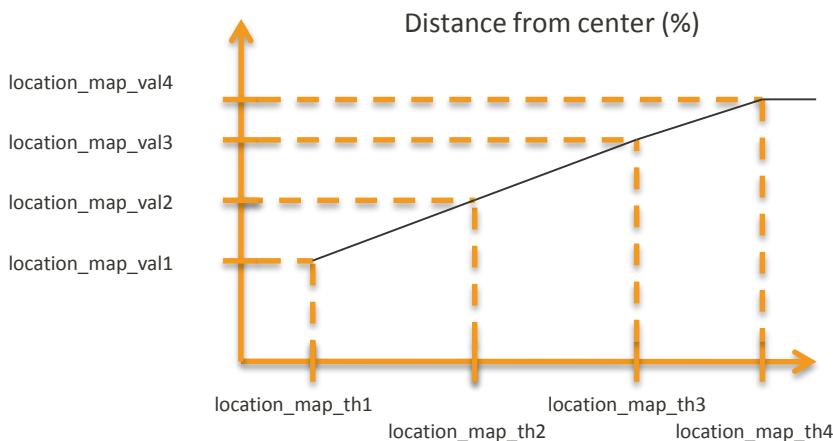
$FgPercentage_Th1_Val \leq FgPercentage_Th2_Val \leq FgPercentage_Th3_Val \leq FgPercentage_Th4_Val$

How to Tune Parameters

Location weighting

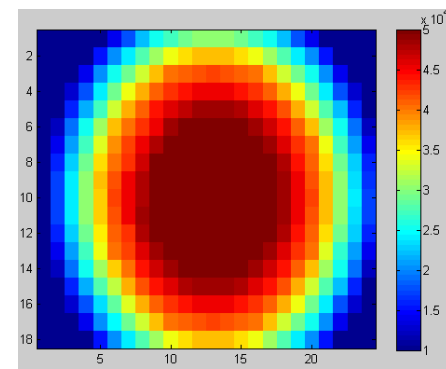
- Parameter file: camera_flash_awb_para_\${sensor}.cpp
 - // Location Weighting Table
 - You can get Location Weighting Table by looking up the table.
 - The closer to the center of the image, the bigger the weighting value.

```
10, //location_map_th1
20, //location_map_th2
40, //location_map_th3
50, //location_map_th4
100, //location_map_val1
100, //location_map_val2
100, //location_map_val3
100, //location_map_val4
```



location_map_th1 <= location_map_th2 <= location_map_th3 <= location_map_th4

location_map_val1 <= location_map_val2 <= location_map_val3 <= location_map_val4



How to Tune Parameters

Luma weighting

- Parameter file: camera_flash_awb_para_\${sensor}.cpp
 - //Luma. Weighting Table

- Reduces the weighting value of the block that is

```

5, //YPrimeWeightTh[0]
9, //YPrimeWeightTh[1]
11, //YPrimeWeightTh[2]
13, //YPrimeWeightTh[3]
15, //YPrimeWeightTh[4]
1, //YPrimeWeight[0]
3, //YPrimeWeight[1]
5, //YPrimeWeight[2]
7, //YPrimeWeight[3]
    
```



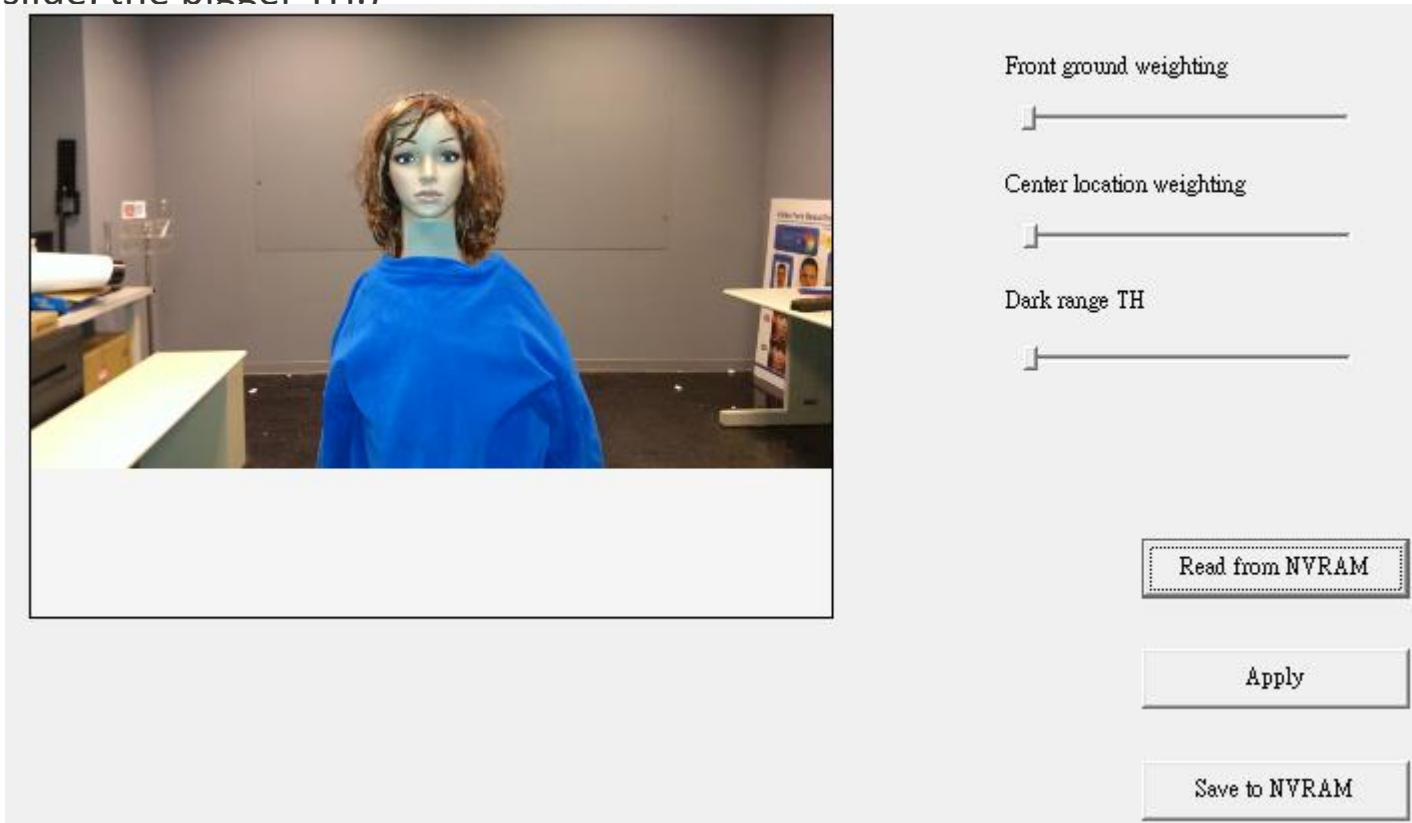
$YPrimeWeight[0] \leq YPrimeWeight[1] \leq YPrimeWeight[2] \leq YPrimeWeight[3]$

$YPrimeWeightTH[0] \leq YPrimeWeightTH[1] \leq YPrimeWeightTH[2] \leq YPrimeWeightTH[3] \leq YPrimeWeightTH[4]$

How to Tune Parameters

CCT Tuning

- **Front ground weighting:** The more the slide, the more white balance in foreground.
- **Center location weighting:** The more the slide, the more white balance in center.
- **Dark range TH:** The block with $Y < TH$ will have a reduced weighting value. (The more the slide, the bigger TH.)



Front ground weighting

Center location weighting

Dark range TH

Read from NVRAM

Apply

Save to NVRAM

Advanced Tuning Method

Scene A



Scene B



Customer preference 1:

Both A & B: Foreground color balance

Customer preference 2:

Both A & B: Background color balance

Customer preference 3:

A: Foreground color balance

B: Background color balance

Check average reflection in JPEG EXIF parser “FL_T_FLAWB_REFLECTION”

FL_T_FLAWB_AVG_REFLECTION

= 0.361

FL_T_FLAWB_AVG_REFLECTION

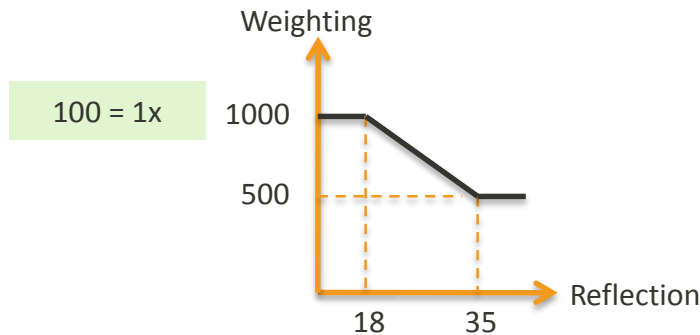
= 0.169

Advanced Tuning Method

[Customer preference 1]

Both A & B: Foreground color balance

- In a scene of lower reflection rate (B), tune high the weighting value of foreground (as below).



FgPercentage_Th1; = 18
FgPercentage_Th2; = 24
FgPercentage_Th3; = 30
FgPercentage_Th4; = 35
FgPercentage_Th1_Val; = 1000
FgPercentage_Th2_Val; = 900
FgPercentage_Th3_Val; = 750
FgPercentage_Th4_Val; = 500

Scene A Result



Scene B Result

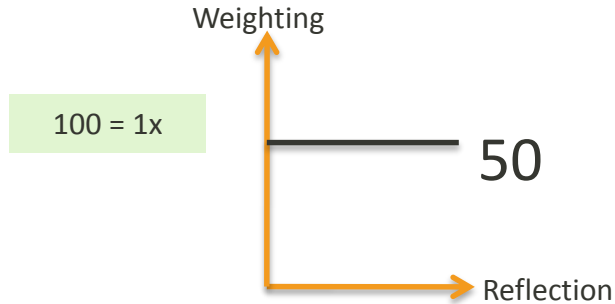


Advanced Tuning Method

[Customer preference 2]

Both A & B: Background color balance

- Reduce the weighting value of foreground (as below)



FgPercentage_Th1; = 18
FgPercentage_Th2; = 24
FgPercentage_Th3; = 30
FgPercentage_Th4; = 35
FgPercentage_Th1_Val; = 50
FgPercentage_Th2_Val; = 50
FgPercentage_Th3_Val; = 50
FgPercentage_Th4_Val; = 50

Scene A Result



Scene B Result



To make A light become pure gray,
disable A light preference in AWB.

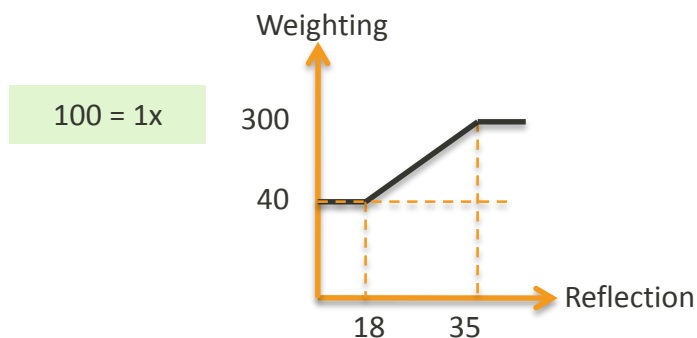
Advanced Tuning Method

[Customer preference 3]

A: Foreground color balance

B: Background color balance

- In a scene of higher reflection rate (A), tune high the weighting value of foreground (as below).



FgPercentage_Th1; = 18
FgPercentage_Th2; = 24
FgPercentage_Th3; = 30
FgPercentage_Th4; = 35
FgPercentage_Th1_Val; = 40
FgPercentage_Th2_Val; = 125
FgPercentage_Th3_Val; = 210
FgPercentage_Th4_Val; = 300

Scene A result



Scene B result

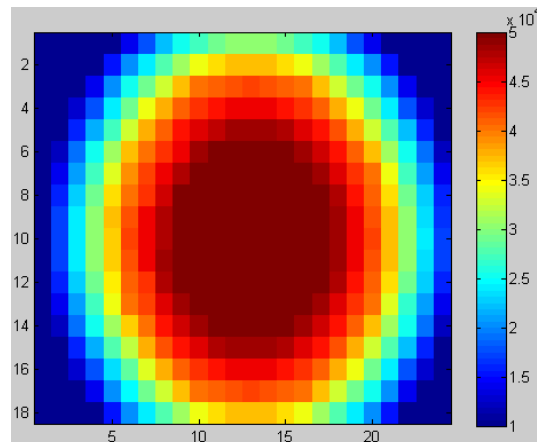


Advanced Tuning Method

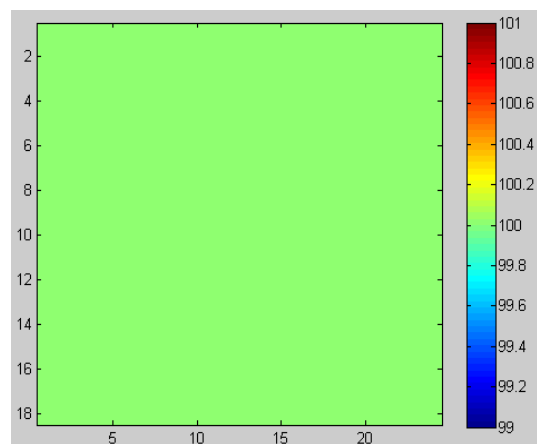
Location Weighting Tuning

- If you tune high the weighting of the center of location weighting table, the center of the image will be balanced to be whiter.

Variable	
location_map_th1	
location_map_th2	
location_map_th3	
location_map_th4	
location_map_val1	
location_map_val2	
location_map_val3	
location_map_val4	



Th1 = 10
Th2 = 20
Th3 = 50
Th4 = 70
Val1 = 100
Val2 = 100
Val3 = 400
Val4 = 500



Th1 = 10
Th2 = 20
Th3 = 50
Th4 = 70
Val1 = 100
Val2 = 100
Val3 = 100
Val4 = 100

Outline

- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- Log analysis
- Issue analysis flow

Debug Parser Tag

Flash AWB Tuning Parameters

Foreground/Background
percentage (try not to
tune it)

FL_T_FLAWB_FG_PERCENTAGE
FL_T_FLAWB_BG_PERCENTAGE

Flash Weighting Table

FL_T_FGPERCENTAGE_TH1
FL_T_FGPERCENTAGE_TH2
FL_T_FGPERCENTAGE_TH3
FL_T_FGPERCENTAGE_TH4
FL_T_FGPERCENTAGE_TH1_VAL
FL_T_FGPERCENTAGE_TH2_VAL
FL_T_FGPERCENTAGE_TH3_VAL
FL_T_FGPERCENTAGE_TH4_VAL

Location Weighting Table

FL_T_LOCATION_MAP_TH1
FL_T_LOCATION_MAP_TH2
FL_T_LOCATION_MAP_TH3
FL_T_LOCATION_MAP_TH4
FL_T_LOCATION_MAP_TH1_VAL
FL_T_LOCATION_MAP_TH2_VAL
FL_T_LOCATION_MAP_TH3_VAL
FL_T_LOCATION_MAP_TH4_VAL

Luma. Weighting Table

FL_T_YPRIME_WEIGHT_TH0
FL_T_YPRIME_WEIGHT_TH1
FL_T_YPRIME_WEIGHT_TH2
FL_T_YPRIME_WEIGHT_TH3
FL_T_YPRIME_WEIGHT_TH4
FL_T_YPRIME_WEIGHT_0
FL_T_YPRIME_WEIGHT_1
FL_T_YPRIME_WEIGHT_2
FL_T_YPRIME_WEIGHT_3

Debug Parser Tag

WB Gain Info (In AWB Page)

FL_AWB_VERSION	Flash AWB Version: should be 201
FL_AWB_PURE_FLASH_RGAIN	Pure flash gain
FL_AWB_PURE_FLASH_GGAIN	
FL_AWB_PURE_FLASH_BAIN	
FL_AWB_NO_FLASH_RGAIN	
FL_AWB_NO_FLASH_GGAIN	Preview gain
FL_AWB_NO_FLASH_BGAIN	
FL_AWB_FINAL_WEIGHT	
FL_AWB_FINAL_WEIGHT	Final weight (should be the same as Strobe page "FL_T_FLAWB_FINAL_WEIGHT")
FL_AWB_RESULT_RGAIN	Pure flash AWB gain
FL_AWB_RESULT_GGAIN	
FL_AWB_RESULT_BGAIN	

•Error Code check provide tuner to check abnormal result of flash AWB

Flash AWB Process Info

FL_T_FLAWB_AVG_REFLECTION	Average reflection
FL_T_FLAWB_FL_STRENGTH	Flash eng (Y)
FL_T_FLAWB_FG_WEIGHT	Foreground weighting
FL_T_FLAWB_FINAL_WEIGHT	Result weighting
FL_T_FLAWB_ERROR_CODE	Error code

Flash AWB Error Code

0x0000	OK
0x0001	FG/BG percentage out of range
0x0002	Foreground weighting table error
0x0004	Location weighting table error
0x000F	Luma. weighting table error
Other	Contact MediaTek engineer.

Outline

- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- [Log analysis](#)
- Issue analysis flow

Log analysis

log :Flash AWB tuning parameter

- flash_awb_para for:9 bk: 95
 - ForeGroundPercentage/ BackGroundPercentage
- lash_awb_para th1,2,3,4:2 5 10 15
 - FgPercentage_Th1/FgPercentage_Th2/FgPercentage_Th3/FgPercentage_Th4

```
01-02 06:23:27.275 343 6934 D flash_mgr_m.cpp: [Flash Awb] Prepare to do flash awb init
01-02 06:23:27.276 343 6934 D nvbuf_util.cpp: nvbuf_util.cpp getBufAndRead+ In=561 ramId=1 sensorDev=1
01-02 06:23:27.276 343 6934 D flash_mgr_m.cpp: flash_awb_para for:9 bk: 95
01-02 06:23:27.276 867 867 D IAudioFlinger: queryEffect() return
01-02 06:23:27.276 343 6934 D flash_mgr_m.cpp: flash_awb_para th1,2,3,4:2 5 10 15
01-02 06:23:27.276 343 6934 D FlashAwb.cpp: [Flash Awb] In Flash_Awb_Init()
```

Log analysis

Log : Flash AWB 2.1 log

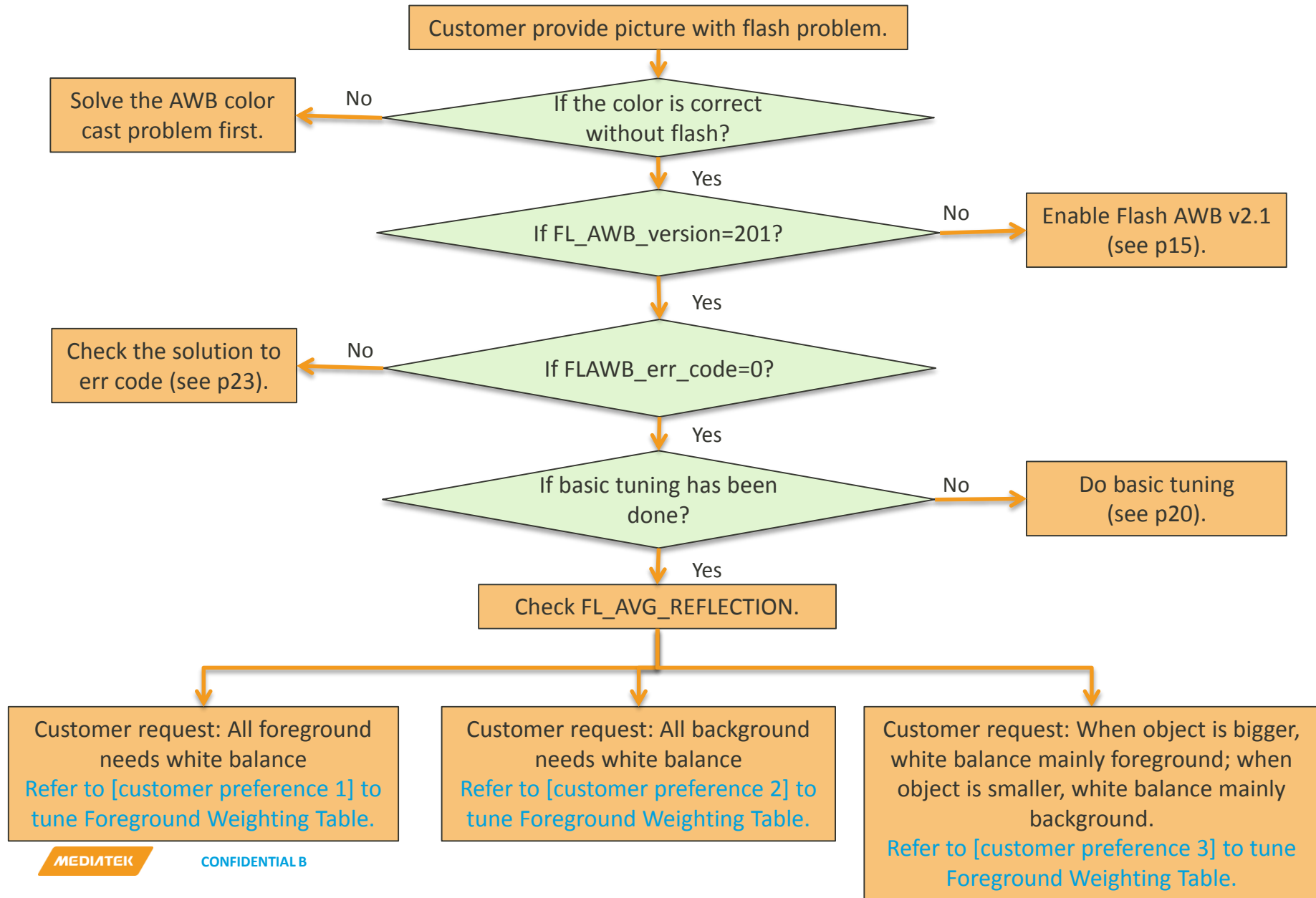
- m_flashAwbWeight
 - Final weight of Flash gain/ non flash gain
- pureFlash Rgain=777, Ggain=512, Bgain=802
 - Flash calibration gain with current duty

```
01-02 06:23:28.831 343 6934 D awb_mgr : [1][flashAWB 2.1] m_flashDuty=1, m_flashAwbWeight=99
01-02 06:23:28.831 343 6934 D awb_mgr : [1][flashAWB 2.1] pureFlash Rgain=777, Ggain=512, Bgain=802
01-02 06:23:28.831 343 6934 D awb_algo: m_PureFlashWeighting= 99
01-02 06:23:28.831 343 6934 D awb_algo: pureFlash Rgain=777, Ggain=512, Bgain=802
01-02 06:23:28.831 343 6934 D awb_algo: preview flash Rgain=953, Ggain=512, Bgain=643
01-02 06:23:28.831 343 6934 D awb_mgr : [1][flashAWB 2.1] In Awb_mgr.cpp - FlashAwbOutput Rgain=778, Ggain=512, Bgain=800
01-02 06:23:28.831 343 6934 D aaa_state_capture: !AwbMgr::getInstance().doCapAWB() END
```

Outline

- Introduction
 - What is Flash AWB
 - Basic algorithm
- Terms
- Flash AE/AWB Calibration
 - Set protection mechanism
 - Flash AE/AWB calibration
- How to tune parameters?
 - Foreground Weighting
 - Location Weighting
 - Luma. Weighting
 - CCT tuning
 - Advanced tuning method
- Debug Parser Tag
- Log analysis
- Issue analysis flow

Issue Analysis Flow





everyday genius