CONFIDENTIAL B

P40 Color Engine New Flexibility Usage

Agenda

- Smooth Color Coding Guidelines
- Troubleshooting

SMOOTH COLOR CODING GUIDELINES

Block Diagram

MT6771_NVRAM_IF_*.xlsx
Sheet : UserTable
Sheet : M! COLOR

Scenario and
CT/LV mapping table

NVRAM

camera_isp_regs_*.h

```
.Lv_Env={
.IDX_Partition ={-30, -10, 10, 50, 100, 120}},
.Ct_Env={
.IDX_Partition ={2000, 2500, 3000, 3500,
4000, 4500, 5000, 5500, 6000, 6500}},
```

***_Scenario_COLOR.cpp**

```
const ISP_NVRAM_SCOLOR_PARAM_T
Sensormipiraw_COLOR_0000 = {
   .FD_COLOR = {...}
   .COLOR_TBL = {...}
:
const ISP_NVRAM_SCOLOR_PARAM_T
Sensormipiraw_COLOR_0059 = {
   .FD_COLOR = {...}
   .COLOR_TBL = {...}
const ISP_NVRAM_COLOR_PARAM_T
Sensormipiraw_COLOR_PARAM_0000 =
{...}
```

Smooth Color
*Per frame active

Interpolation by CT → Modulation by LV → IIR

HW

Color Engine

mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/MT6771_NVRAM_IF_**Sensor**mipiraw.xlsx
mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/camera_isp_regs_**Sensor**mipiraw.h
mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/**Scenario**/**Sensor**mipiraw_**Scenario**_COLOR.cpp

MT6771_NVRAM_IF_Sensormipiraw.xlsx (Sheet : UserTable)

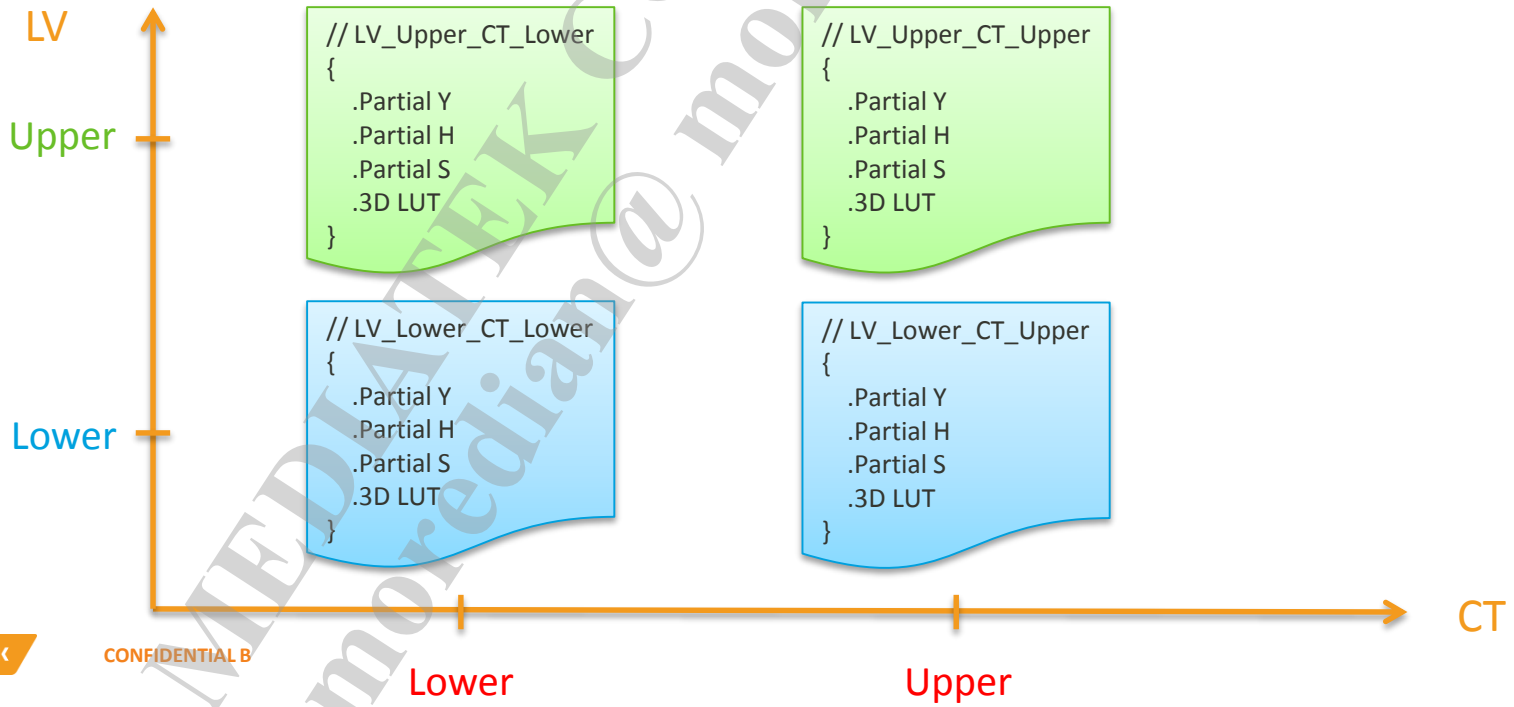
Scenario	CCM	COLOR
Scene_Capture	**Scene_Capture**	**Scene_Capture**
Face_Capture	**Face_Capture**	**Face_Capture**
Zoom_Capture	Scene_Capture	Scene_Capture
Professional_Capture	Scene_Capture	Scene_Capture
Flash_Capture	**Flash_Capture**	**Flash_Capture**
FaceBeauty_Capture	**FaceBeauty_Capture**	**FaceBeauty_Capture**
HDR_Capture	**HDR_Capture**	**HDR_Capture**
Panorama_Capture	Scene_Capture	Scene_Capture
Video_Capture	Scene_Capture	Scene_Capture
Capture_Preview	Scene_Capture	Scene_Capture
Capture_Preview_Zoom1	Scene_Capture	Scene_Capture
Capture_Preview_Zoom2	Scene_Capture	Scene_Capture
Video_Preview	Scene_Capture	Scene_Capture
Video_Preview_Zoom1	Scene_Capture	Scene_Capture
Video_Preview_Zoom2	Scene_Capture	Scene_Capture
Video_Recording	Scene_Capture	Scene_Capture
Video_Recording_Zoom1	Scene_Capture	Scene_Capture
Video_Recording_Zoom2	Scene_Capture	Scene_Capture
FaceBeauty_Preview	FaceBeauty_Capture	FaceBeauty_Capture
WeChatQQ	**WeChatQQ**	**WeChatQQ**
3rd_1080P	Scene_Capture	Scene_Capture
3rd_720P	Scene_Capture	Scene_Capture
3rd_480P	Scene_Capture	Scene_Capture

MT6771_NVRAM_IF_Sensormipiraw.xlsx (Sheet : M! COLOR)

LV	CT	ISO	Index	Folder	File	Scenario
IDX_00	IDX_00		0	Scene_Capture	COLOR	Scene_Capture
IDX_00	IDX_01		1	Scene_Capture	COLOR	Scene_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_00	IDX_09		9	Scene_Capture	COLOR	Scene_Capture
IDX_01	IDX_00		10	Scene_Capture	COLOR	Scene_Capture
IDX_01	IDX_01		11	Scene_Capture	COLOR	Scene_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_01	IDX_09		19	Scene_Capture	COLOR	Scene_Capture
IDX_05	IDX_00		50	Scene_Capture	COLOR	Scene_Capture
IDX_05	IDX_01		51	Scene_Capture	COLOR	Scene_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_05	IDX_09		59	Scene_Capture	COLOR	Scene_Capture
IDX_00	IDX_00		60	Face_Capture	COLOR	Face_Capture
IDX_00	IDX_01		61	Face_Capture	COLOR	Face_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_05	IDX_09		119	Face_Capture	COLOR	Face_Capture
IDX_00	IDX_00		0	Scene_Capture	COLOR	Zoom_Capture
IDX_00	IDX_01		1	Scene_Capture	COLOR	Zoom_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_05	IDX_09		59	Scene_Capture	COLOR	Zoom_Capture
⋮ ⋮	⋮ ⋮		⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮
IDX_00	IDX_00		120	Flash_Capture	COLOR	Flash_Capture
IDX_00	IDX_01		121	Flash_Capture	COLOR	Flash_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_05	IDX_09		179	Flash_Capture	COLOR	Flash_Capture
⋮ ⋮	⋮ ⋮		⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮

NVRAM by Scenario

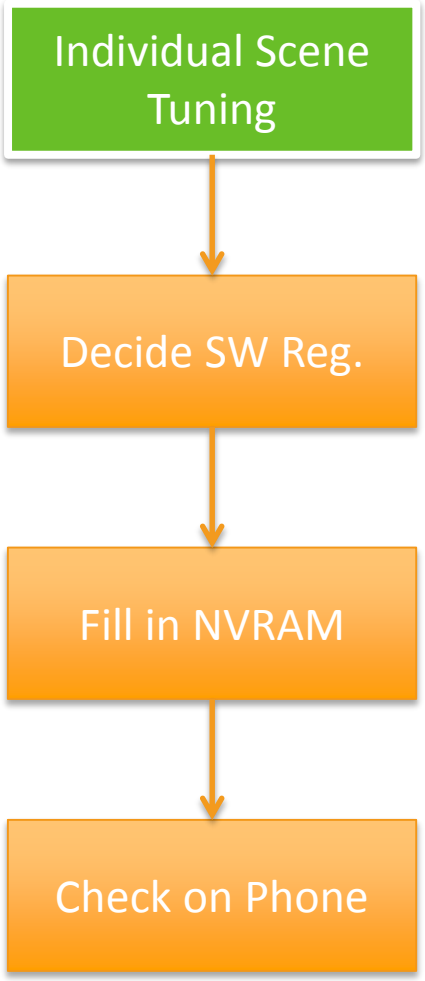
- Each Scenario has 60 COLOR_TBLs, the output COLOR_TBL is bilinear interpolation result by 1/CT(Mired) and LV
  - 6 LV partitions
  - 10 CT partitions



Individual Scene Tuning

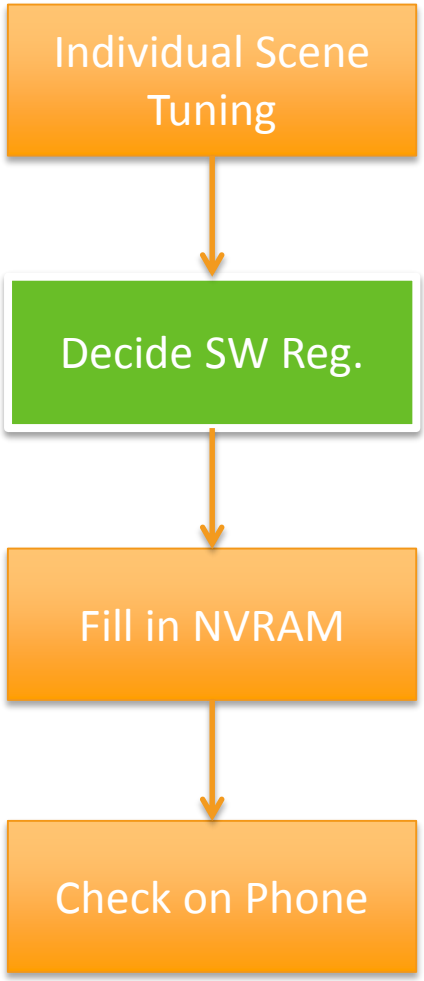
- Database
  - Different CT
  - Different lighting condition
- Scene
  - Indoor

  - Outdoor



Decide SW Register

- LUM/HUE/SAT SPEED
  - Unit: 1 code/frame
  - Default: 4 code/frame
- LSP LV TH
  - LSP effective range by LV



mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/**Scenario**/**Sensor**mipiraw_Scenario_COLOR.cpp

```
const ISP_NVRAM_COLOR_PARAM_T Sensormipiraw_COLOR_PARAM_0000 = {
   0,  //DC OFFSET : unused
   4,  //LUM SPEED
   4,  //HUE SPEED
   4,  //SAT SPEED
   30, //LSP LV TH
   0,  //OUTDOOR EN : unused
   4,  //OUTDOOR SPEED : unused
   16  //OUTDOOR RATIO : unused
};
```

Fill in NVRAM

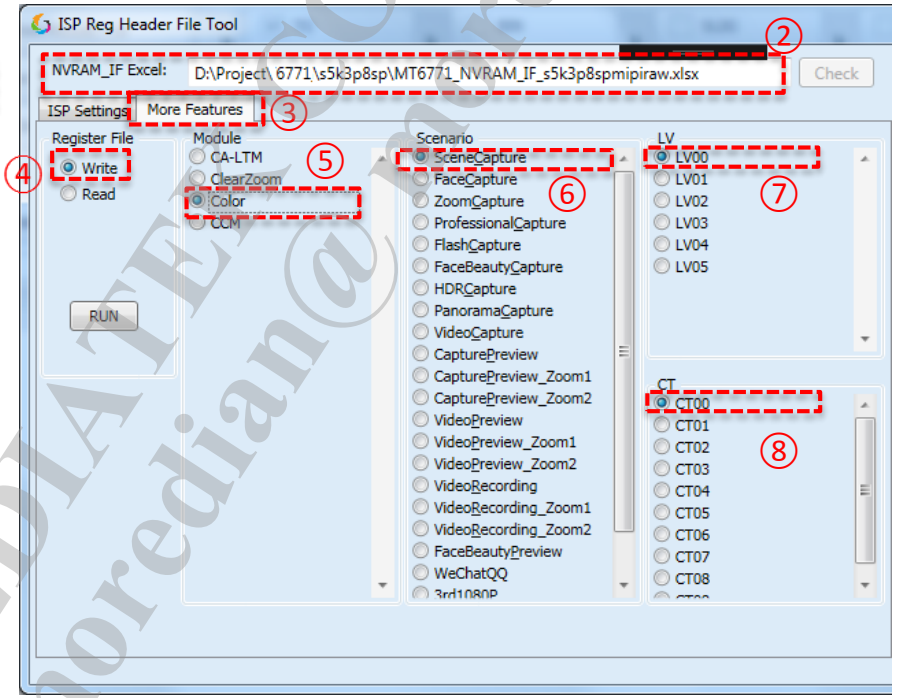
- Step

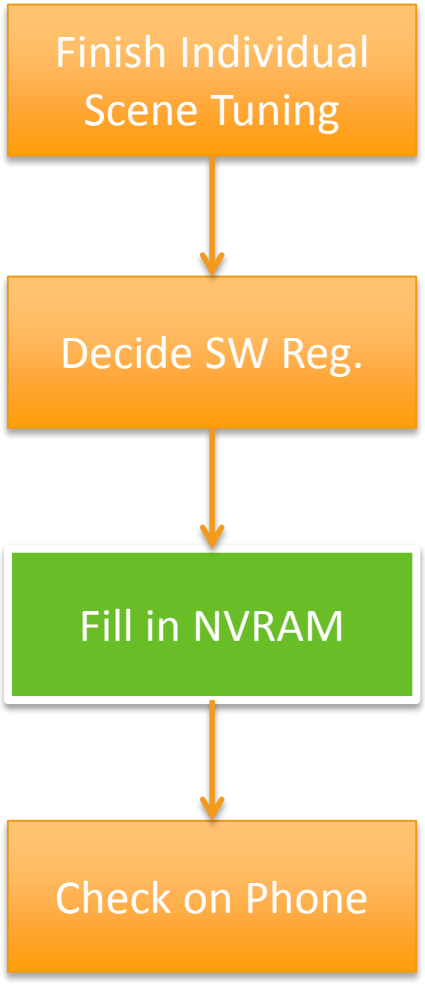
  Tool → ISP Reg Header File Tool

  → Open “NVRAM_IF.xlsx” → Select “More Feature”

  → Select “Write” → Select “Color” → Select “Scenario”

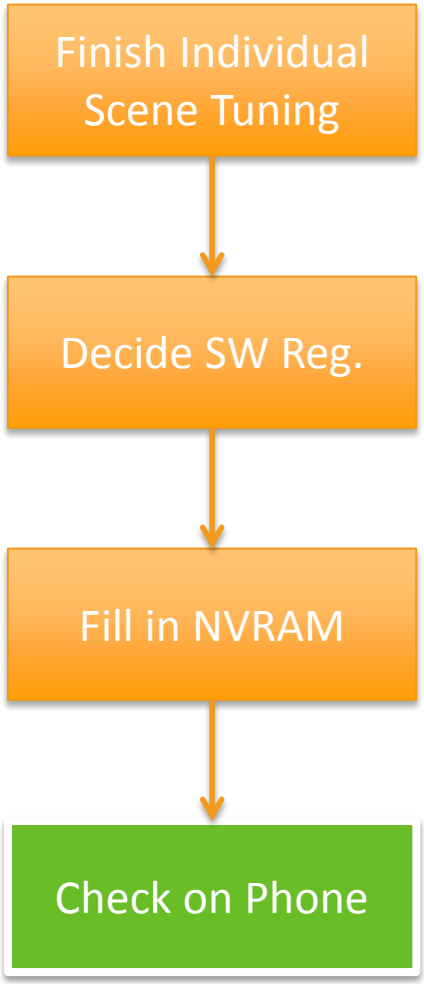
  → Select “LV” → Select “CT” → Run





Check on Phone

- Field test
  - Check setting in EXIF
  - Check preview screen
- Troubleshooting



TROUBLESHOOTING

Troubleshooting

- Bypass Y/S/H Engine and Memory Color
- Dump Input Parameters
- Dump Interpolation Results
  - Dump Global Brightness/Contrast/Saturation
  - Dump Partial Y
  - Dump Partial H
  - Dump Partial S
  - Dump S Gain by Y
  - Dump LSP
  - Dump Memory Color

Bypass Y/S/H Engine and Memory Color

Be used to debug abnormal color effect in preview or video

- Step 1: Turn on debug enable
  - adb shell setprop debug.smooth_color.enable 1
- Step 2: Bypass **Y/S/H engine and memory color**
  - adb shell setprop debug.smooth_color.bypass.yeng 1
  - adb shell setprop debug.smooth_color.bypass.heng 1
  - adb shell setprop debug.smooth_color.bypass.seng 1
  - adb shell setprop debug.smooth_color.bypass.cm 1

Dump Input Parameters

- Step 1: Turn on debug enable
  - adb shell setprop debug.smooth_color.dump 1
- Step 2: Check the debug info (=>Note)

```
[SmoothCOLOR()] u4RealCT = 3164, u4LowerCT = 3000, u4UpperCT = 3500, i4RealLV = 27, i4LowerLV = 10, i4UpperLV = 50
=> CT/LV info
[SmoothCOLOR()] IsCapture = 0, IsBoot = 0
=> If IsCapture = 1 or IsBoot = 1, there is no IIR
[SmoothCOLOR()] In Ram: 000, 004, 004, 004, 030, 000, 004, 016
=> See below description
[SmoothCOLOR()] LSP = (1, 1)
=> (1,1) : COLOR_LSP_EN = 1, otherwise COLOR_LSP_EN = 0
[SmoothCOLOR()] Interpolated mode !!
=> Output “Interpolated mode !!” or “Force Lock !!”
```

mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/**Scenario**/**Sensor**mipiraw_Scenario_COLOR.cpp

```
const ISP_NVRAM_COLOR_PARAM_T Sensormipiraw_COLOR_PARAM_0000 = {
   0,  //DC OFFSET : unused
   4,  //LUM SPEED
   4,  //HUE SPEED
   4,  //SAT SPEED
   30, //LSP LV TH
   0,  //OUTDOOR EN : unused
   4,  //OUTDOOR SPEED : unused
   16  //OUTDOOR RATIO : unused
};
```

Dump Global Brightness/Contrast/Saturation

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.global 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamGlobal()]

```
[InterParamGlobal()]  u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamGlobal()] (LV Lower, Mired Lower)  B:128,C:128,S:128
[InterParamGlobal()] (LV Lower, Mired Upper)  B:128,C:128,S:128
[InterParamGlobal()] (LV Upper, Mired Lower) B:128,C:128,S:128
[InterParamGlobal()] (LV Upper, Mired Upper) B:128,C:128,S:128
[InterParamGlobal()] (Target)  B:128,C:128,S:128
[InterParamGlobal()] (Final )  B:128,C:128,S:128
```

Dump Partial Y

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.lum 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamPartialY()]

```
[InterParamPartialY()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamPartialY()] PartialY LV Lower, Mired Lower : 128,128,128,128,130,131,134,144,128,128,128,128,128,128,128,128
[InterParamPartialY()] PartialY LV Lower, Mired Upper: 128,128,128,128,112,112,112,128,128,128,128,128,128,128,128,128
[InterParamPartialY()] PartialY LV Upper, Mired Lower : 160,160,160,160,120,120,120,160,160,144,160,160,160,160,160,160
[InterParamPartialY()] PartialY LV Upper, Mired Upper: 107,104,109,107,139,138,137,106,104,116,106,109,107,104,109,107
[InterParamPartialY()] PartialY Target : 128,127,128,128,121,120,121,131,127,128,128,128,128,127,128,128
[InterParamPartialY()] PartialY Final : 128,127,128,128,121,120,121,131,127,128,128,128,128,127,128,128
```

Dump Partial H

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.hue 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamPartialH()]

```
[InterParamPartialH()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamPartialH()] PartialH LV Lower, Mired Lower : 128,128,128,128,130,131,134,144,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialH()] PartialH LV Lower, Mired Upper : 128,128,128,128,130,131,134,144,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialH()] PartialH LV Upper, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialH()] PartialH LV Upper, Mired Upper : 128,128,128,128,138,154,138,140,128,128,128,128,128,128,128,128,128,144,132,128
[InterParamPartialH()] PartialH Target : 128,128,128,128,131,134,134,144,128,128,128,128,128,128,128,128,128,131,129,128
[InterParamPartialH()] PartialH Final : 128,128,128,128,131,134,134,144,128,128,128,128,128,128,128,128,128,131,129,128
```

Dump Partial S 1/3

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.sat 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamPartialS()]

Dump Partial $S_{2/3}$

[InterParamPartialS()]u4RealMired = 200, u4UpperMired = 181, u4LowerMired = 250, i4RealLV = 60, i4UpperLV = 100, i4LowerLV = 50
[InterParamPartialS()] PartialS point1 Lv Lower Mired Lower : 020,020,020,020,020,020,010,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Lv Lower Mired Upper : 020,020,020,020,020,020,010,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Lv Upper Mired Lower : 020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Lv Upper Mired Upper : 020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point2 Lv Lower Mired Lower : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Lv Lower Mired Upper : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Lv Upper Mired Lower : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Lv Upper Mired Upper : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS gain1 Lv Lower Mired Lower : 128,128,128,128,112,112,112,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Lv Lower Mired Upper : 128,128,128,128,112,112,112,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Lv Upper Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Lv Upper Mired Upper : 128,128,128,128,120,112,116,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain2 Lv Lower Mired Lower : 140,140,140,140,120,120,120,144,144,132,140,140,112,120,144,140,140,140,140,140
[InterParamPartialS()] PartialS gain2 Lv Lower Mired Upper : 136,144,144,144,120,120,120,144,144,132,144,144,144,144,144,144,144,144,136,128
[InterParamPartialS()] PartialS gain2 Lv Upper Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain2 Lv Upper Mired Upper : 138,138,138,138,120,112,116,128,128,128,138,142,148,142,138,138,128,128,128,128
[InterParamPartialS()] PartialS gain3 Lv Lower Mired Lower : 120,119,121,120,139,138,136,114,113,124,120,121,139,134,118,120,120,121,119,120
[InterParamPartialS()] PartialS gain3 Lv Lower Mired Upper : 123,116,118,118,139,138,136,114,113,124,117,118,117,116,118,118,117,119,122,128
[InterParamPartialS()] PartialS gain3 Lv Upper Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain3 Lv Upper Mired Upper : 121,121,122,121,136,143,139,128,128,128,121,120,115,118,122,121,128,128,128,128

Dump Partial $S_{3/3}$

[InterParamPartialS()] PartialS output1 Lv Lower Mired Lower : 020,020,020,020,018,018,009,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output1 Lv Lower Mired Upper : 020,020,020,020,018,018,009,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output1 Lv Upper Mired Lower : 020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output1 Lv Upper Mired Upper : 020,020,020,020,019,018,018,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output2 Lv Lower Mired Lower : 064,064,064,064,056,056,056,066,066,062,064,064,055,058,065,064,064,064,064,064
[InterParamPartialS()] PartialS output2 Lv Lower Mired Upper : 063,065,065,065,056,056,056,066,066,062,065,065,065,065,065,065,065,065,063,060
[InterParamPartialS()] PartialS output2 Lv Upper Mired Lower : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS output2 Lv Upper Mired Upper : 063,063,063,063,057,053,054,060,060,060,063,064,066,064,063,063,060,060,060,060
[InterParamPartialS()] PartialS point1 Target : 020,020,020,020,020,020,012,012,012,012,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Final : 020,020,020,020,020,020,012,012,012,012,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point2 Target : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Final : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS output1 Target : 020,020,020,020,018,018,011,012,012,012,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output2 Target : 063,065,065,065,056,056,056,066,066,062,065,065,063,064,065,065,065,065,063,061
[InterParamPartialS()] PartialS gain1 Target : 128,128,128,128,115,115,117,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Final : 128,128,128,128,115,115,117,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain2 Target : 138,144,144,144,122,122,120,144,144,133,144,144,138,141,144,144,144,144,138,131
[InterParamPartialS()] PartialS gain2 Final : 138,144,144,144,122,122,120,144,144,133,144,144,138,141,144,144,144,144,138,131
[InterParamPartialS()] PartialS gain3 Target : 122,116,118,118,137,136,136,115,114,123,117,118,122,119,118,118,117,119,121,126
[InterParamPartialS()] PartialS gain3 Final : 122,116,118,118,137,136,136,115,114,123,117,118,122,119,118,118,117,119,121,126

Dump S Gain by $Y_{1/2}$

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.sgy 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamSGainByY()]

Dump S Gain by Y 2/2

[InterParamSGainByY()]u4RealMired = 200, u4UpperMired = 181, u4LowerMired = 250, i4RealLV = 60, i4UpperLV = 100, i4LowerLV = 50
[InterParamSGainByY()] SGainByY000 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 LV Lower, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 LV Upper, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 Target : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 Final : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY064 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY064 LV Lower, Mired Upper : 127,127,121,122,122,121,119,120,120,120,123,128,123,122,131,130,131,116,117,125
[InterParamSGainByY()] SGainByY064 LV Upper, Mired Lower : 127,127,123,122,131,130,131,116,117,125,121,122,122,121,119,120,120,120,123,128
[InterParamSGainByY()] SGainByY064 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY064 Target : 127,127,124,125,125,124,123,123,123,123,125,128,125,125,131,130,131,121,122,126
[InterParamSGainByY()] SGainByY064 Final : 127,127,124,125,125,124,123,123,123,123,125,128,125,125,131,130,131,121,122,126
[InterParamSGainByY()] SGainByY128 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY128 LV Lower, Mired Upper : 124,122,122,121,120,122,126,125,124,125,127,128,125,125,131,131,131,127,126,127
[InterParamSGainByY()] SGainByY128 LV Upper, Mired Lower : 124,122,125,125,131,131,131,127,126,127,122,121,120,122,126,125,124,125,127,128
[InterParamSGainByY()] SGainByY128 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY128 Target : 126,125,125,124,123,125,127,126,126,126,127,128,126,126,131,131,131,127,127,127
[InterParamSGainByY()] SGainByY128 Final : 126,125,125,124,123,125,127,126,126,126,127,128,126,126,131,131,131,127,127,127
[InterParamSGainByY()] SGainByY192 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY192 LV Lower, Mired Upper : 125,121,128,127,127,127,122,122,124,124,123,128,120,120,140,140,138,116,116,126
[InterParamSGainByY()] SGainByY192 LV Upper, Mired Lower : 125,121,120,120,140,140,138,116,116,126,128,127,127,127,122,122,124,124,123,128
[InterParamSGainByY()] SGainByY192 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY192 Target : 126,124,128,127,128,128,125,124,125,126,125,128,123,123,137,137,135,121,121,127
[InterParamSGainByY()] SGainByY192 Final : 126,124,128,127,128,128,125,124,125,126,125,128,123,123,137,137,135,121,121,127
[InterParamSGainByY()] SGainByY256 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 LV Lower, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 LV Upper, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 Target : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 Final : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128

Dump LSP

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.lsp 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamLSP()]

```
[InterParamLSP()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamLSP()] LSP1 LV Lower, Mired Lower : 000,000,127,000
[InterParamLSP()] LSP1 LV Lower, Mired Upper : 000,000,127,000
[InterParamLSP()] LSP1 LV Upper, Mired Lower : 000,000,127,000
[InterParamLSP()] LSP1 LV Upper, Mired Upper : 000,000,127,000
[InterParamLSP()] LSP1 Target : 000,000,127,000
[InterParamLSP()] LSP1 Final  : 000,000,127,000
[InterParamLSP()] LSP2 LV Lower, Mired Lower : 127,000,127,127
[InterParamLSP()] LSP2 LV Lower, Mired Upper : 127,000,127,127
[InterParamLSP()] LSP2 LV Upper, Mired Lower : 127,000,127,127
[InterParamLSP()] LSP2 LV Upper, Mired Upper : 127,000,127,127
[InterParamLSP()] LSP2 Target : 127,000,127,127
[InterParamLSP()] LSP2 Final  : 127,000,127,127
```

Dump Memory Color 1/2

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.dump.cm 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParam3DLUT()]

Dump Memory Color 2/2

Only show W3 Hue

[InterParam3DLUT()]u4RealMired = 200, u4UpperMired = 181, u4LowerMired = 250, i4RealLV = 60, i4UpperLV = 100, i4LowerLV = 50
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Lower Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Lower Mired Upper : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Upper Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Upper Mired Upper : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Target : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Final : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Lower Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Lower Mired Upper : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Upper Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Upper Mired Upper : 0806,0847,0888,0929,0934,0939,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Target : 0806,0832,0857,0883,0903,**0000**,0944
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Lower Mired Lower : 128,128,128,128,128,128
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Lower Mired Upper : 128,128,128,128,128,128
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Upper Mired Lower : 128,128,128,128,128,128
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Upper Mired Upper : 230,230,230,026,026,026
[InterParam3DLUT()] 3DLUT W3 Hue Slope Target : 145,139,145,111,**000,255**
[InterParam3DLUT()] 3DLUT W3 Hue Slope Final : 145,139,145,111,**000,255**
[InterParam3DLUT()] 3DLUT W3 Hue Lv Lower Mired Lower : WGT_LSLOPE = 000, WGT_USLOPE = 000
[InterParam3DLUT()] 3DLUT W3 Hue Lv Lower Mired Upper : WGT_LSLOPE = 016, WGT_USLOPE = 016
[InterParam3DLUT()] 3DLUT W3 Hue Lv Upper Mired Lower : WGT_LSLOPE = 016, WGT_USLOPE = 016
[InterParam3DLUT()] 3DLUT W3 Hue Lv Upper Mired Upper : WGT_LSLOPE = 016, WGT_USLOPE = 016
[InterParam3DLUT()] 3DLUT W3 Hue Target : WGT_LSLOPE = 014, WGT_USLOPE = 014
[InterParam3DLUT()] 3DLUT W3 Hue Final : WGT_LSLOPE = 014, WGT_USLOPE = 014

APPENDIX : P23/P40 COLOR ENGINE

CONFIDENTIAL B

P23/P40 Color Engine

Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

Support Chip

- P23
- P40

Document Revision

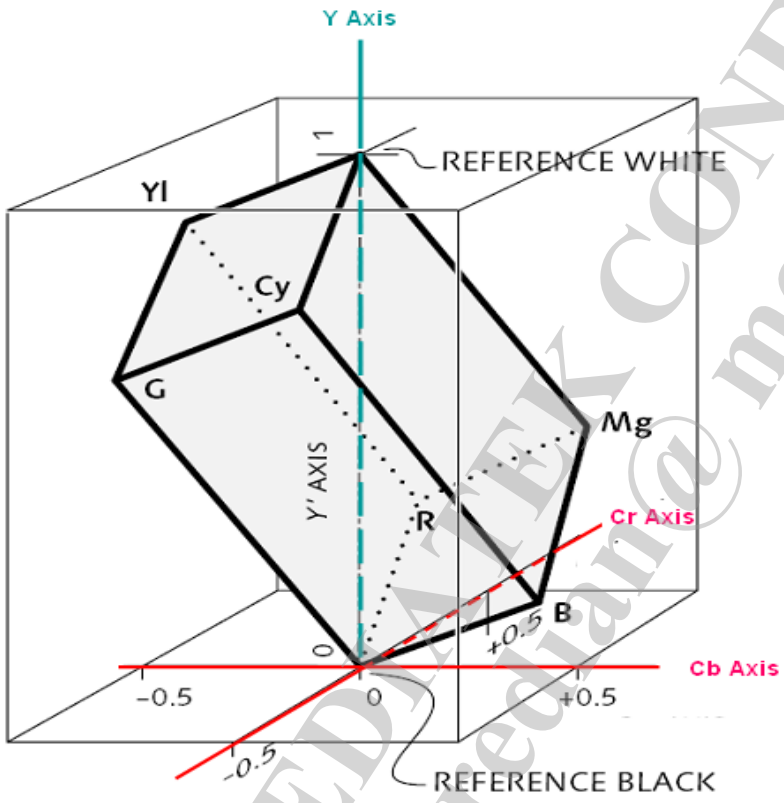
Date	Note	Modify by
2017/02/18	Initial draft	Jay Huang
2017/03/08	Update new UI (Partial S/LSP)	Jay Huang
2017/05/09	Add Memory Color	Jay Huang

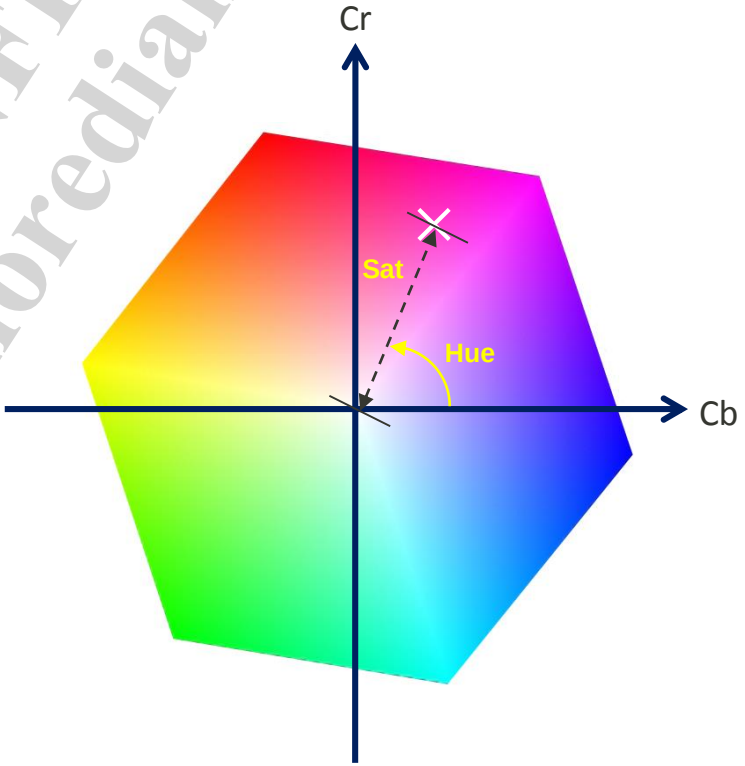
Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

Function Introduction

- Color Engine can process color in YSH domain.

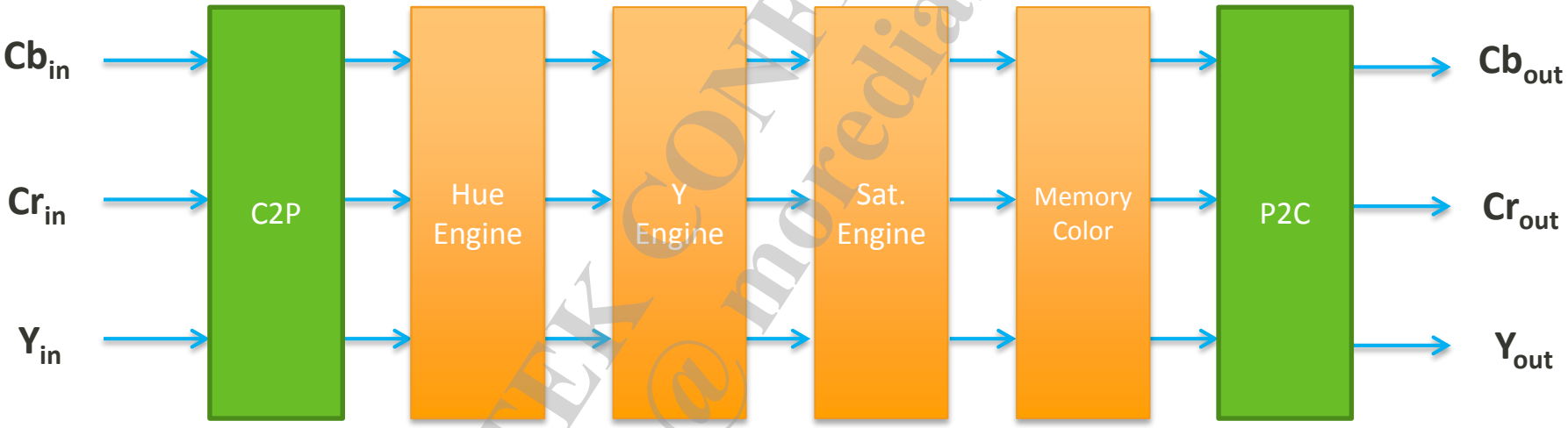




Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

Block Diagram

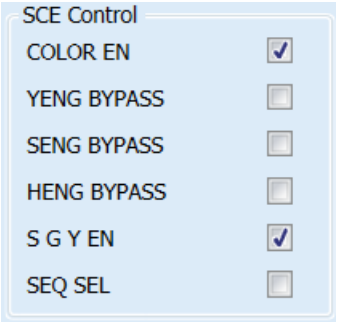


Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

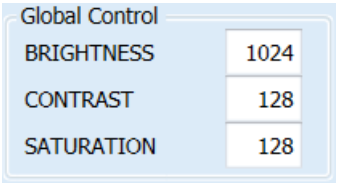
Global Control

- COLOR EN (Default = 1)
  - Module enable
- YENG/SENG/HENG BYPASS (Default = 0)
  - Bypass sub modules
- S G Y EN (Default = 1)
  - Enable Saturation Gain by Y
- SEQ SEL (Default = 0)
  - Switch between original hue (0) and processed hue (1)



Global Contrast / Brightness / Sat.

- BRIGHTNESS
  - 0.0 x : 0
  - 1.0 x : 1024
  - 2.0 x : 2047
- CONTRAST
  - 0.0 x: 0
  - 1.0 x : 128
  - 7.992 x : 1023
- SATURATION
  - 0.0 x: 0
  - 1.0 x : 128
  - 7.992 x : 1023

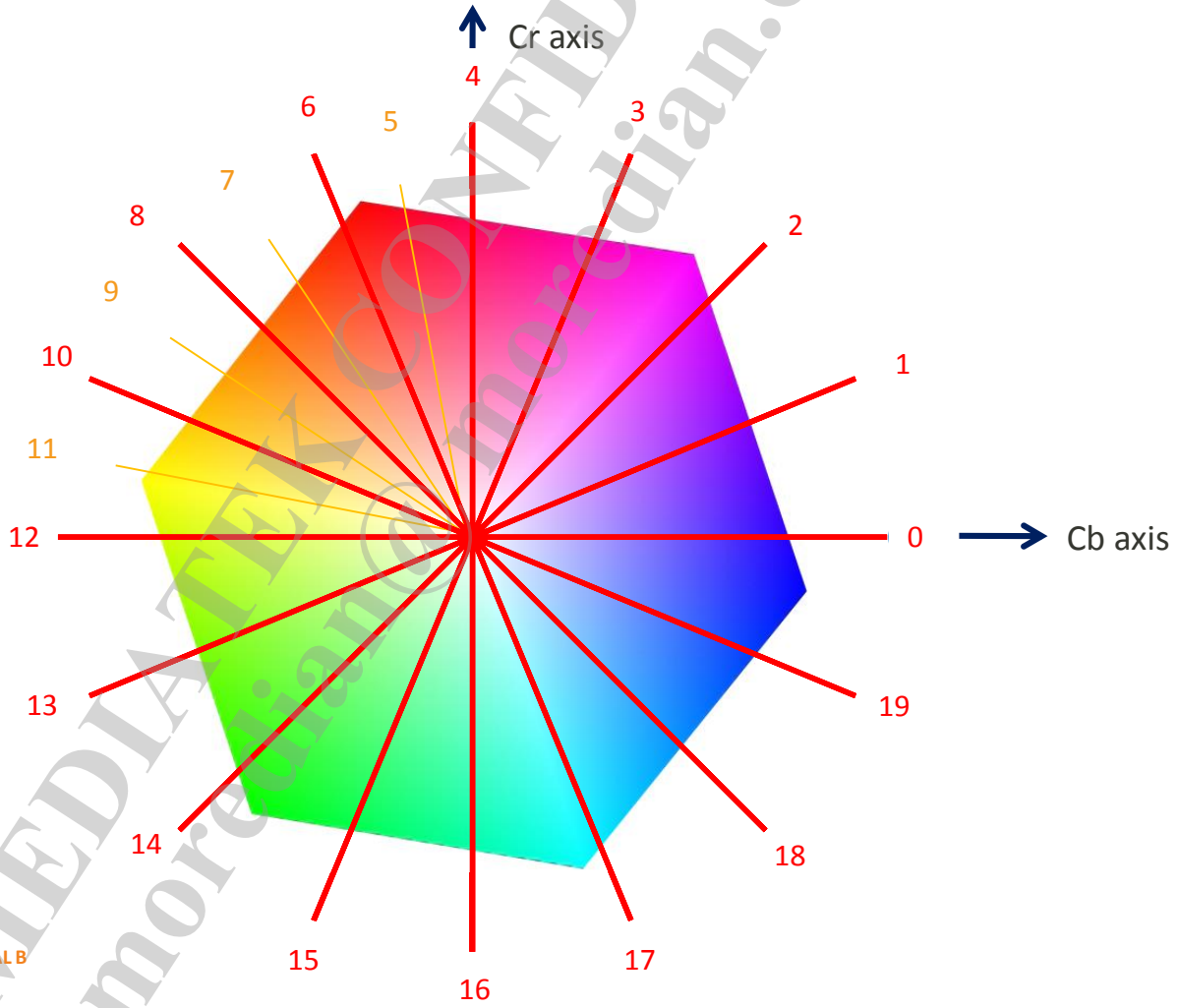


Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

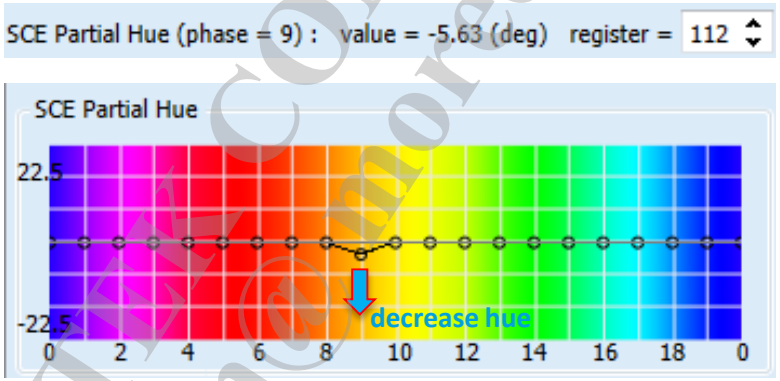
Hue Phase Division

- 20 hue phases, denser around skin tone

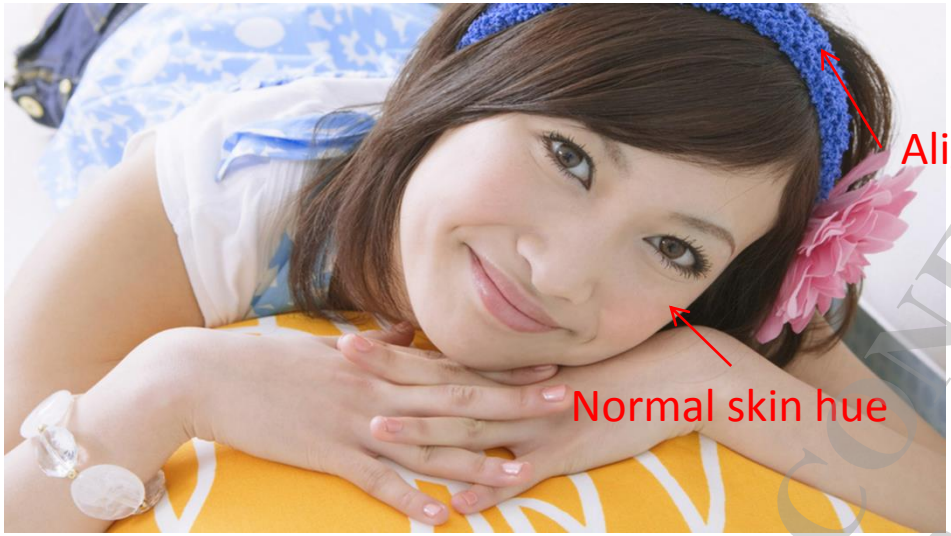


Partial Hue

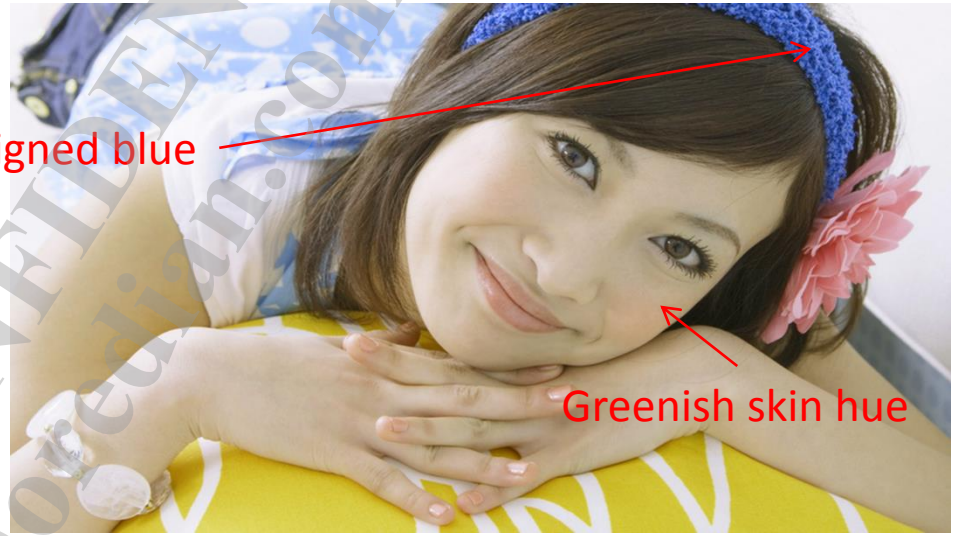
- Modify hue angle for specific hue phase
  - Register adjust range is 0 ~ 255
  - Corresponds to -45° ~ +44.65°



Simulation Result



Partial Hue output



Global Hue output

Input

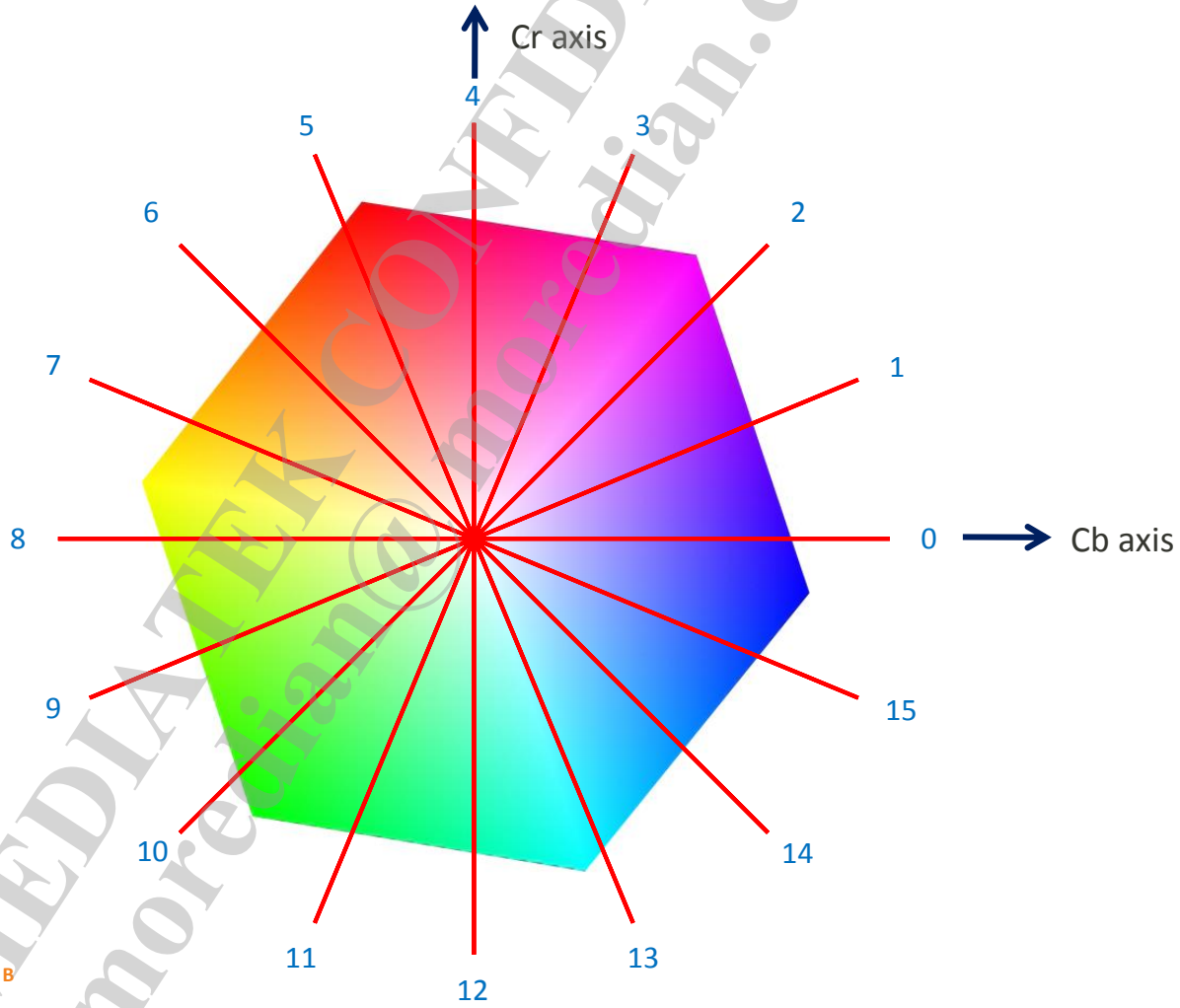
- Adjustment in **Hue**

Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

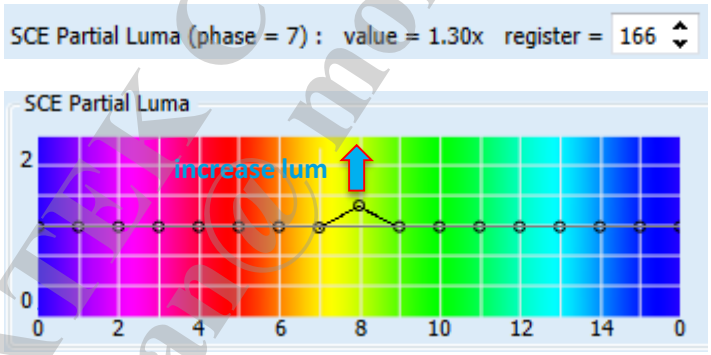
Hue Phase Division

- 16 hue phases

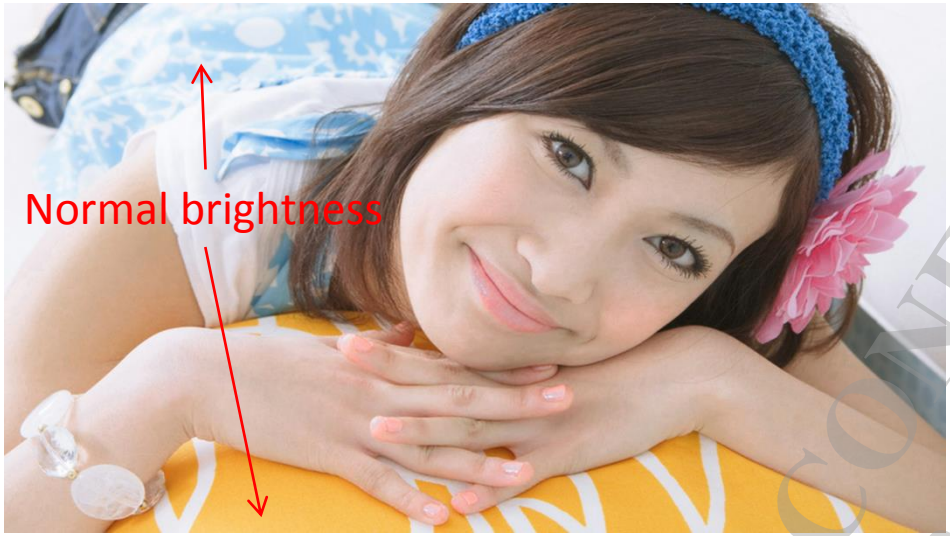


Partial Y

- Modify Y for specific hue phase
  - 16 hue phases available
  - Register adjust range is 0 ~ 255
  - Corresponds to 0x ~ 1.992x



Simulation Result



Partial Y output

Global Y output

Input

- Adjustment in **Luma**

Index

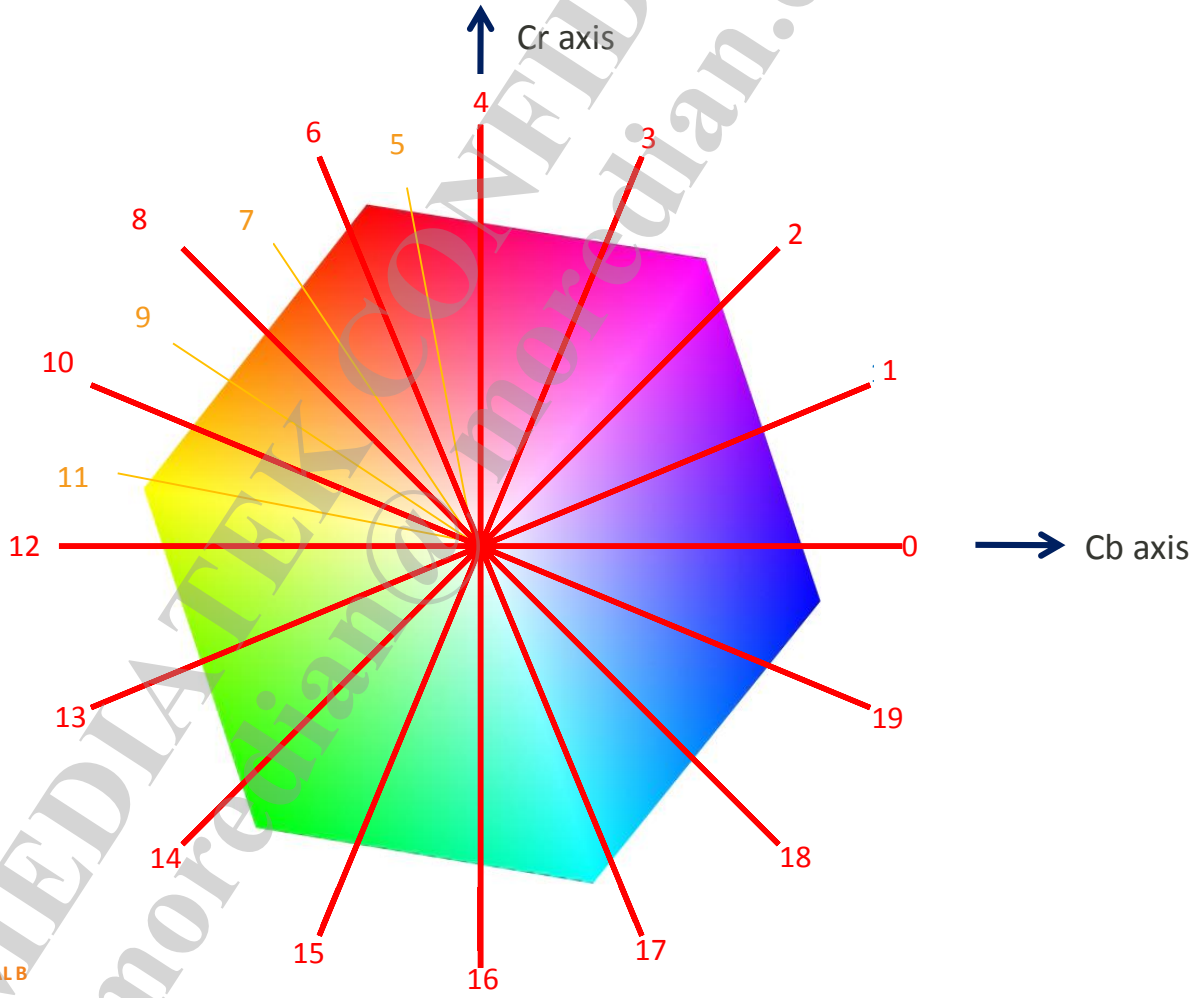
- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

S Engine

- Partial Saturation
- Saturation Gain by Y (SGainByY)
- Low Saturation Protection (LSP)

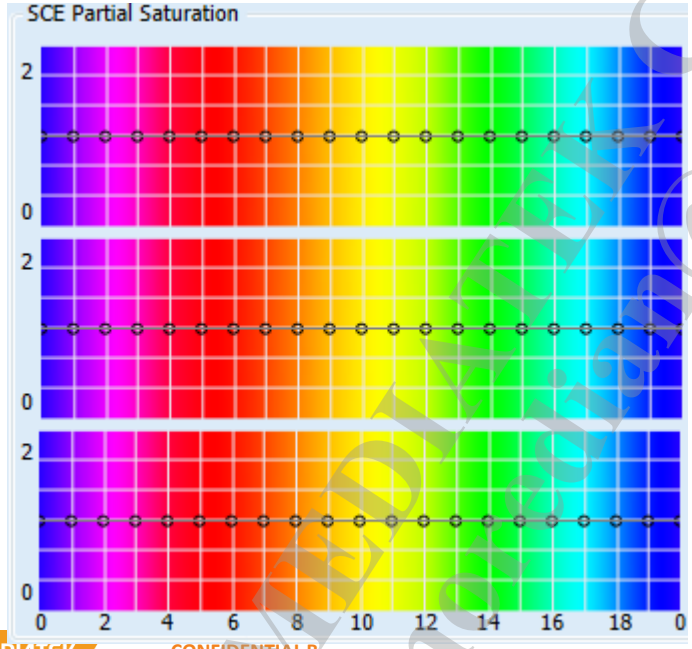
Hue Phase Division

- 20 hue phases, denser around skin tone



Partial Saturation

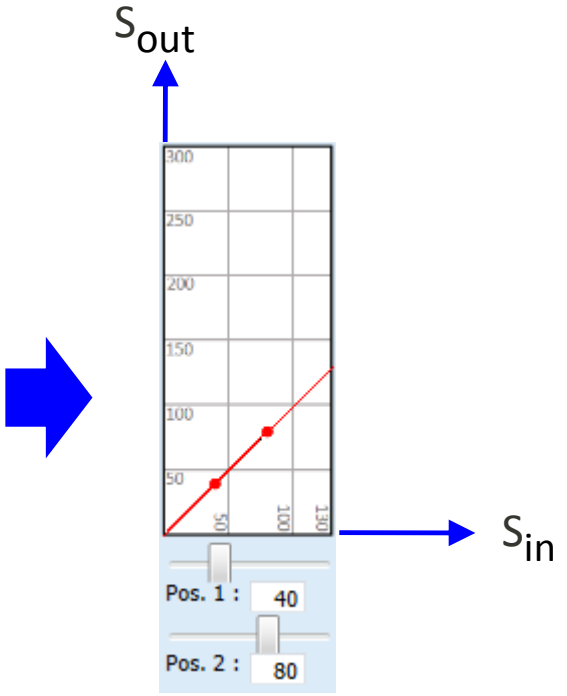
- Modify saturation for specific hue phase
  - 3 gains & 2 control points for each phase
  - Gain adjust range is 0x ~ 1.992x (Register : 0 ~ 255)
  - P1/P2 adjust range is 0 ~ 130



G1: S_{in} < P1

G1: P1 < S_{in} < P1

G1: P2 < S_{in}



Simulation Result

Partial S output

Global S output

Input

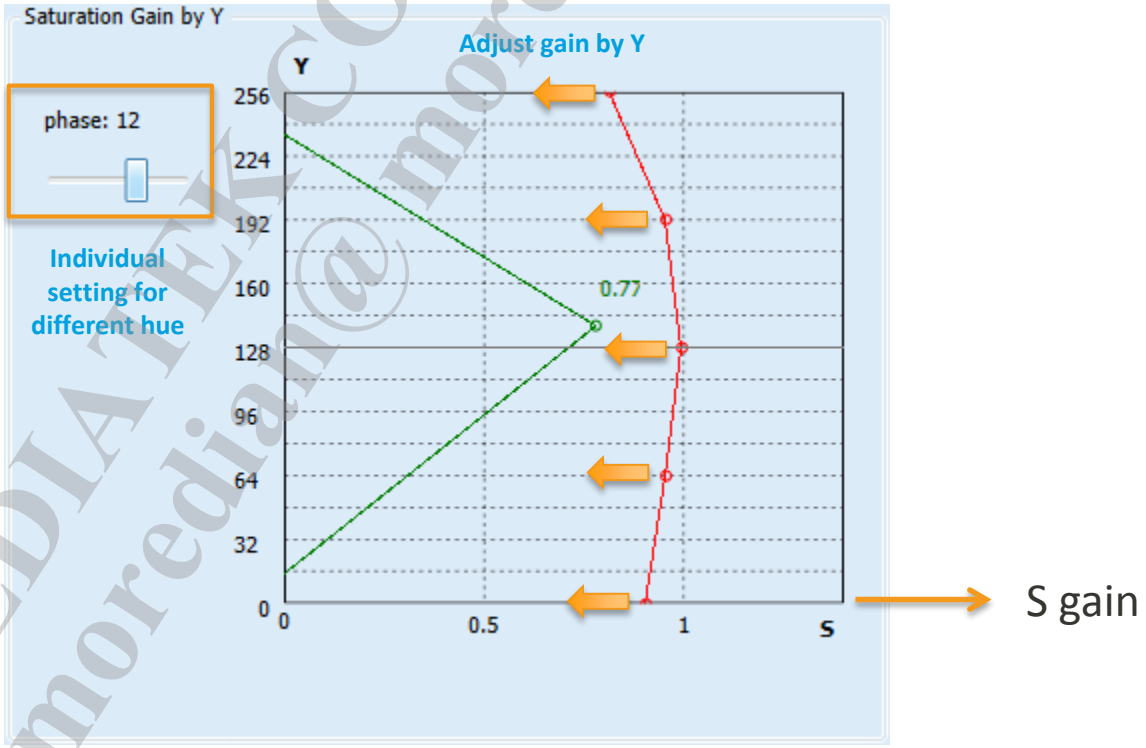
- Adjustment in **Saturation**

S Engine

- Partial Saturation
- Saturation Gain by Y (SGainByY)
- Low Saturation Protection (LSP)

Saturation Gain by Y

- 5 control gains for **each phase**
- The gain interpolated based on Y is applied to S
  - Adjust range is 0x ~ 1.992x (Register : 0 ~ 255)



Simulation Result

W/O SGainByY

: Over-saturation

Input

Same Color enhancement

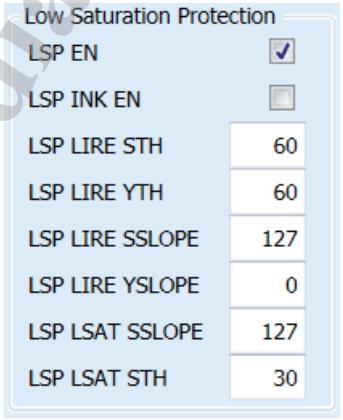
W/ SGainByY

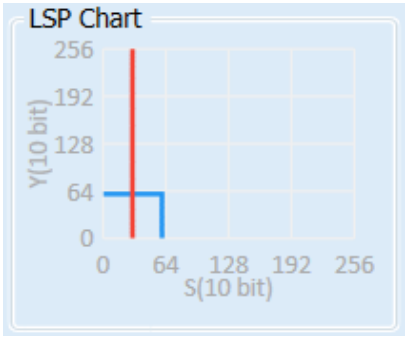
S Engine

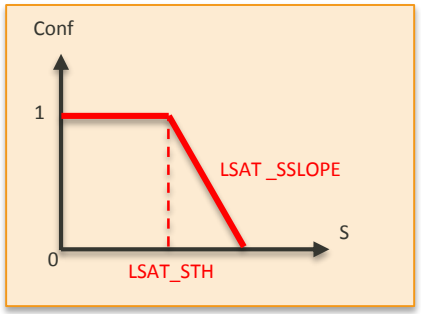
- Partial Saturation
- Saturation Gain by Y (SGainByY)
- Low Saturation Protection (LSP)

Low Saturation Protection (LSP)$_{1/2}$

- Control registers shared by **all hue phases**
  - LSP EN
    - Enable LSP
  - LSP INK EN
    - Enable ink mode of LSP
  - LSP LSAT STH
    - Low sat. region threshold
  - LSP LSAT SSLOPE
    - Low sat. region slope

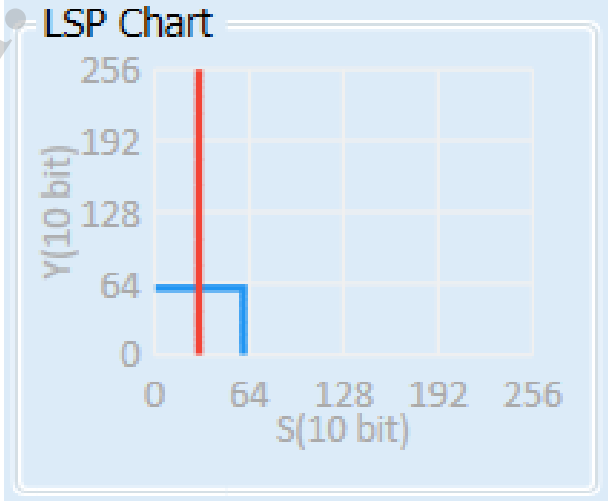


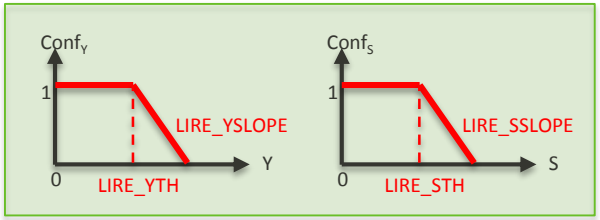




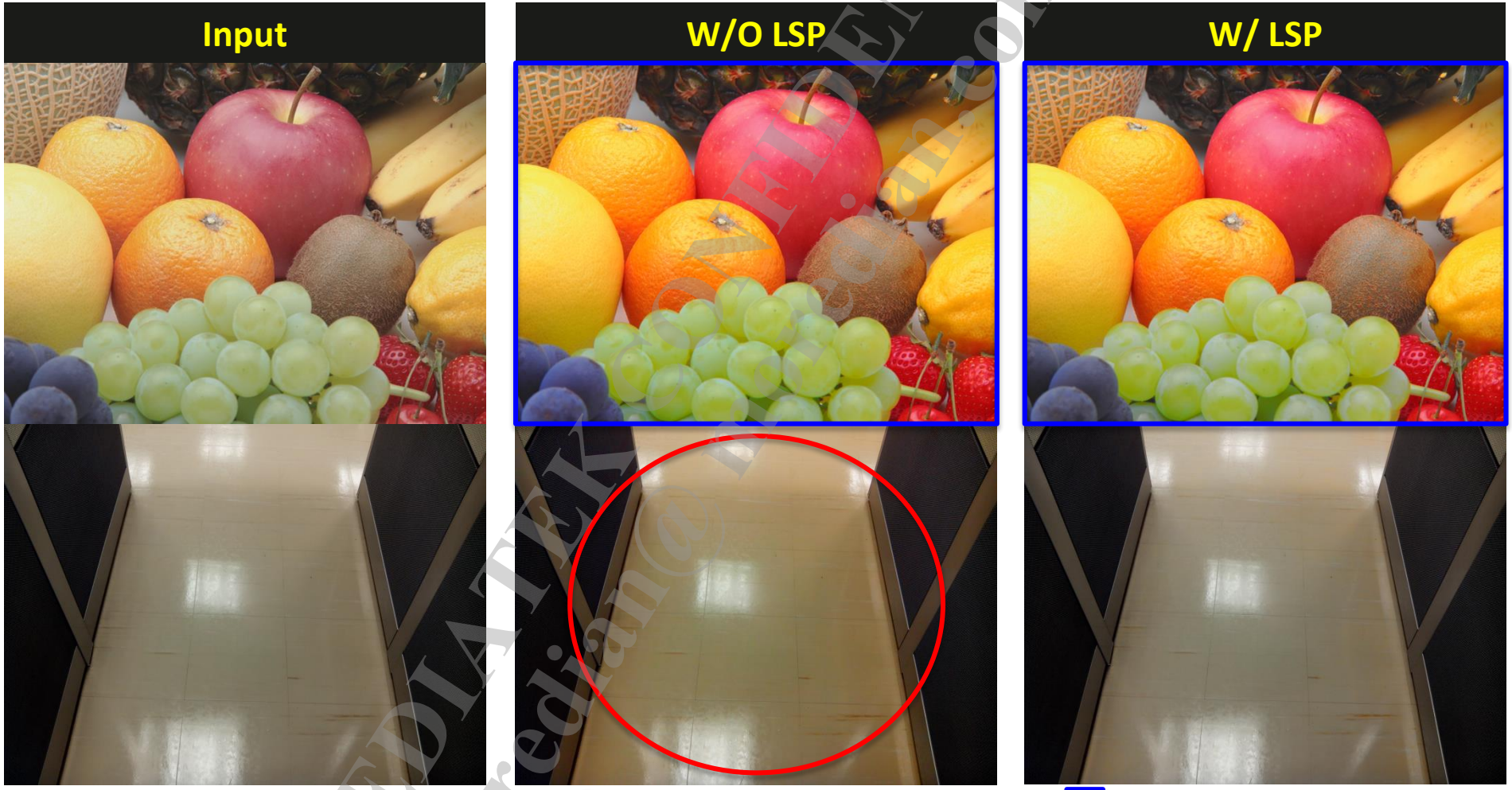
Low Saturation Protection (LSP)$_{2/2}$

- (Cont.)
  - LSP LIRE STH
    - Low IRE region saturation threshold
  - LSP LIRE YTH
    - Low IRE region luma threshold
  - LSP LIRE SSLOPE
    - Low IRE region saturation slope
  - LSP LIRE YLOPE
    - Low IRE region saturation slope





Simulation Result

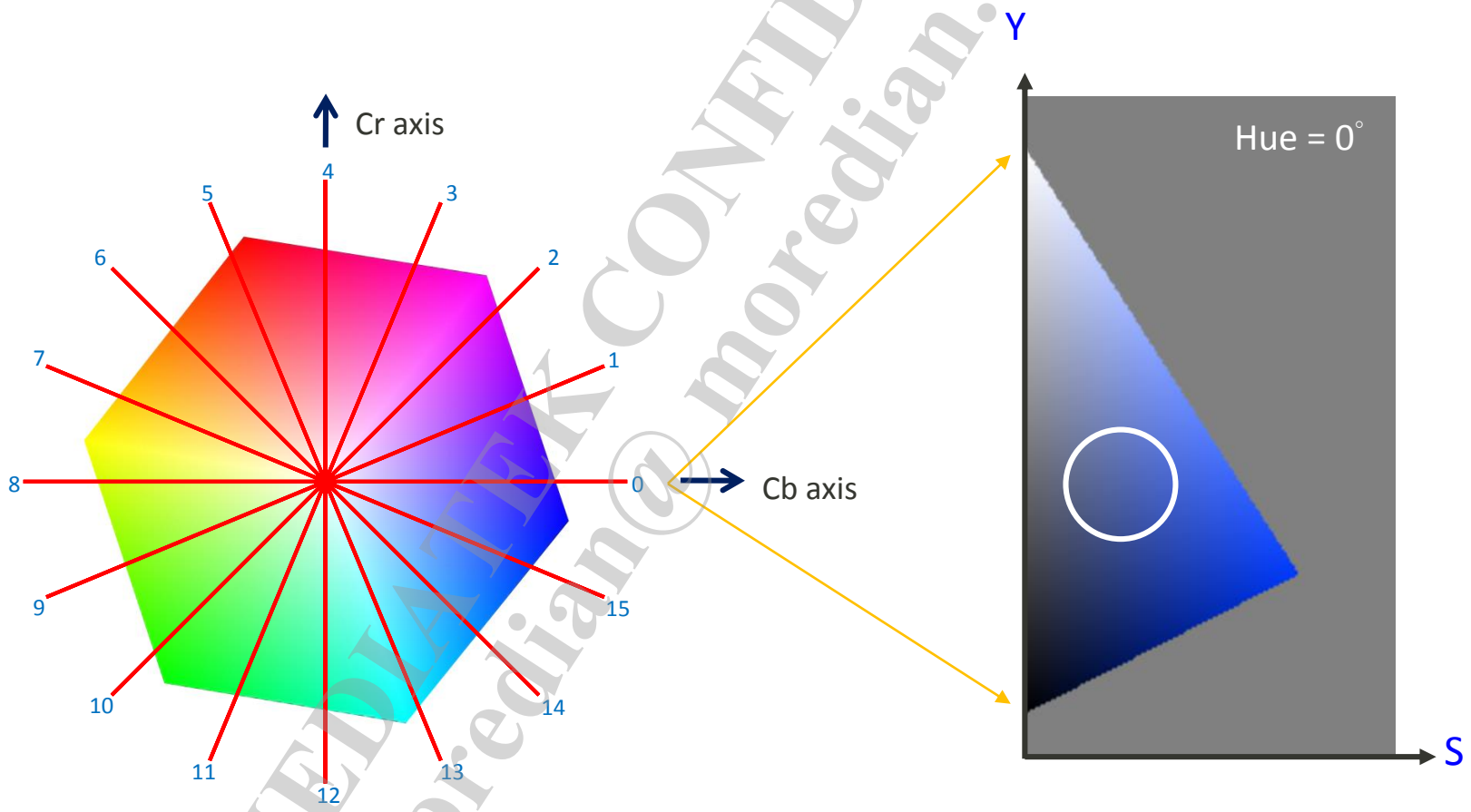


Index

- Support chip
- Document revision
- Function introduction
- Block diagram
- Function
  - Global control
  - H Engine
  - Y Engine
  - S Engine
  - Memory Color

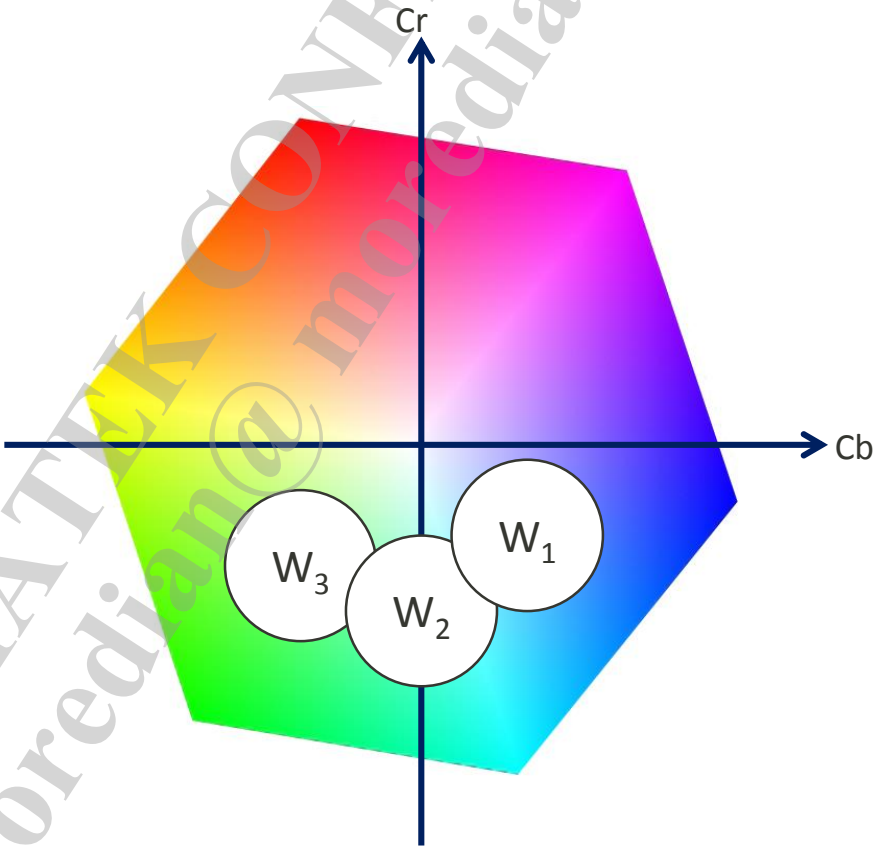
Hue-based Tuning Limitation

- One Hue phase as a processing unit



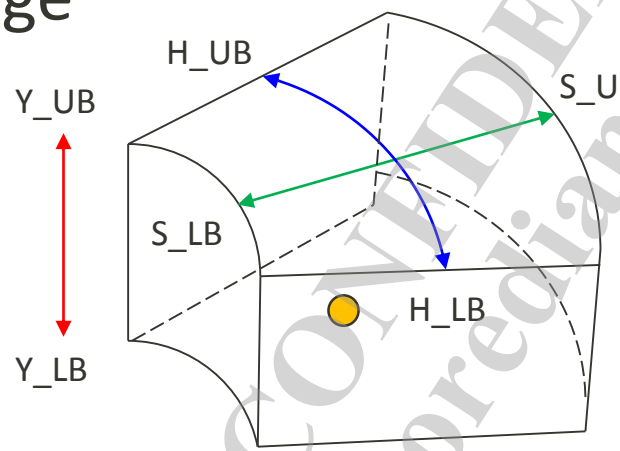
Memory Color 1/2

- Three sets of Color Mapping window
- Order: $W_1 > W_2 > W_3$



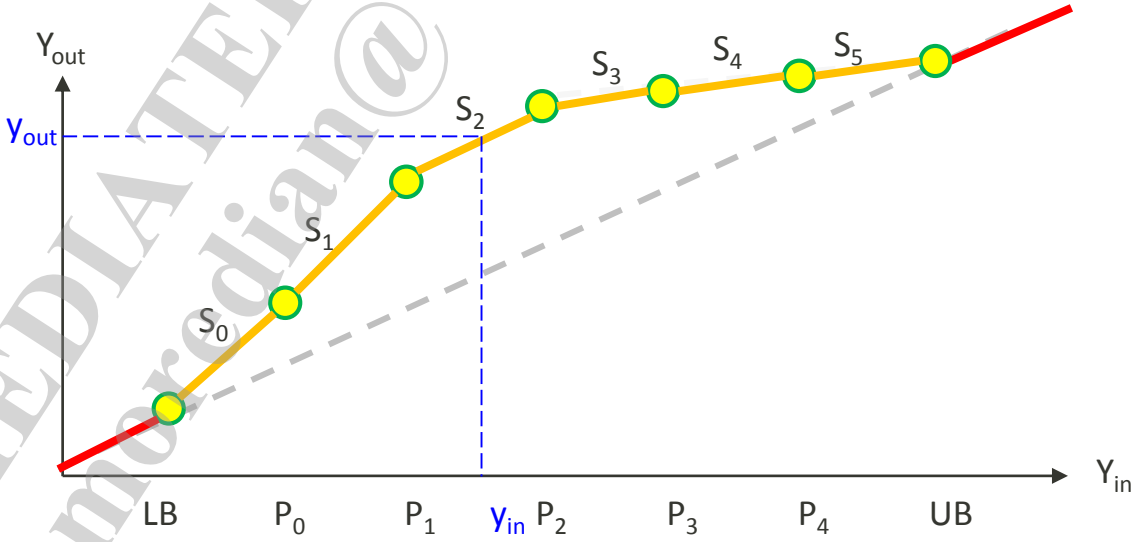
Memory Color 2/2

- Window range



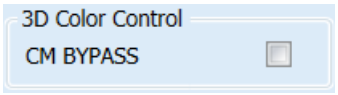
- UB: Upper boundary
- LB: Lower boundary

- 5 turning points & 6 slopes for each of Y/S/H

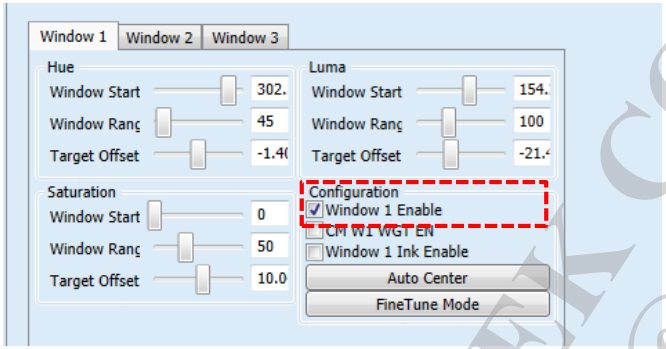


Operation Step

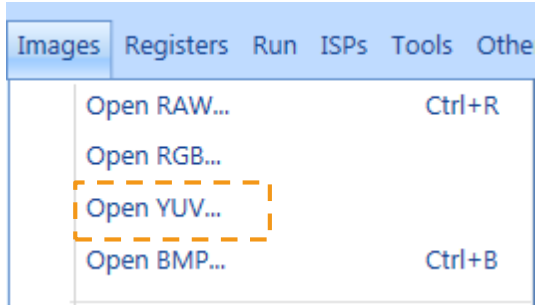
1) Turn on Memory Color

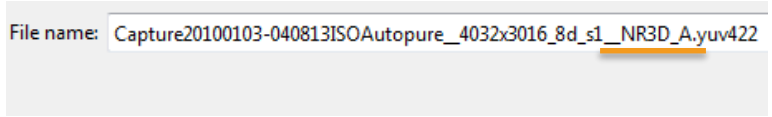


2) Enable Window



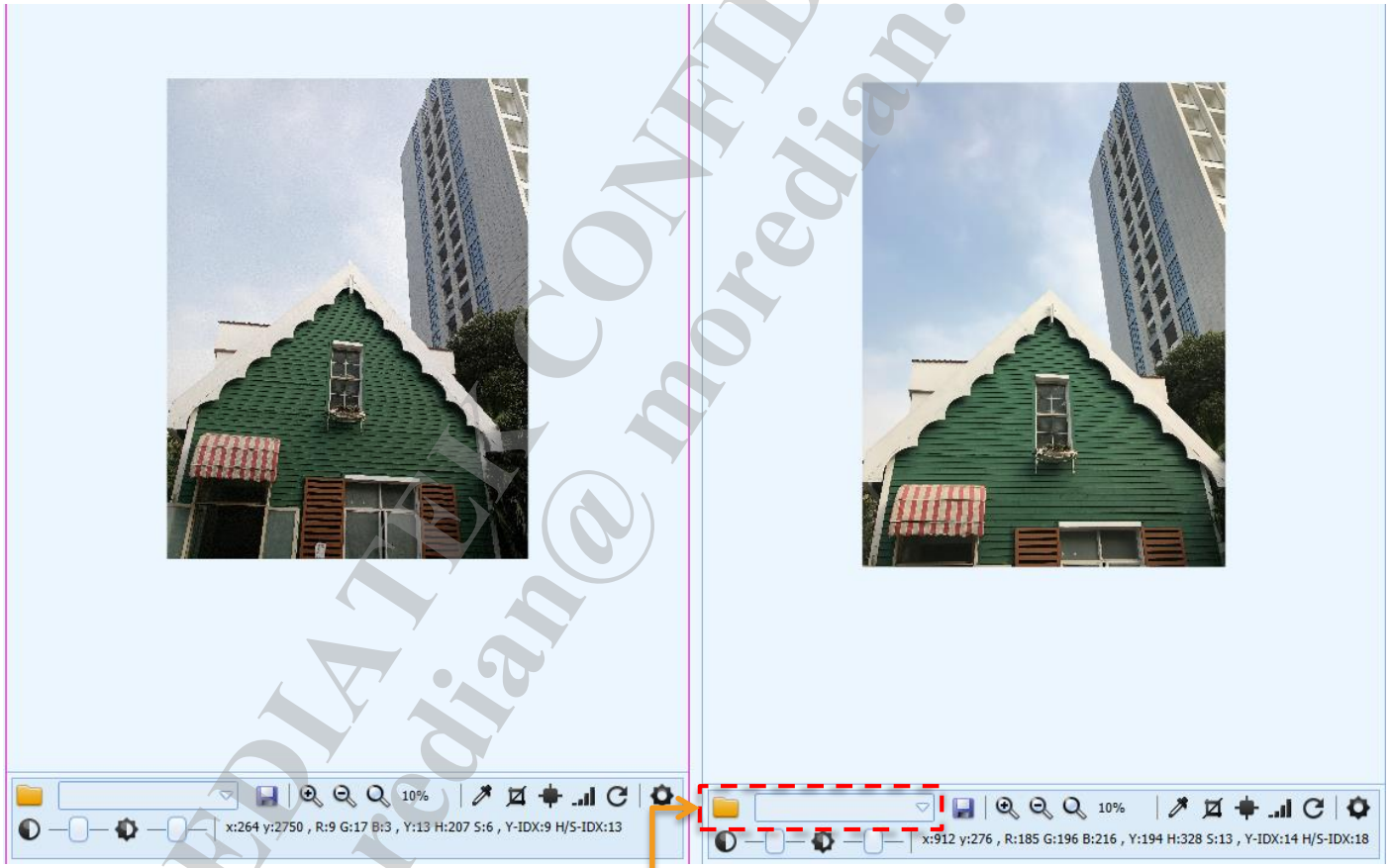
3) Use COLOR/NR3D output as new input





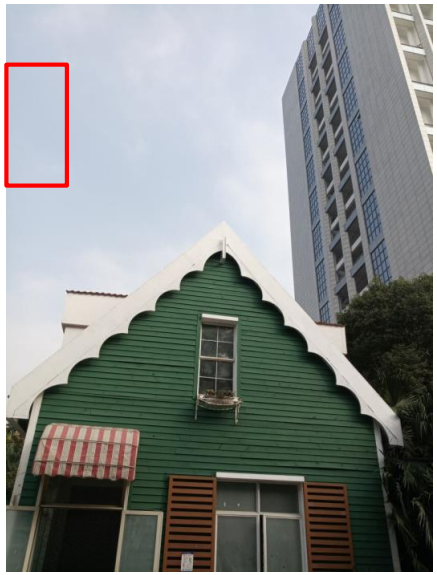
Operation Step

4) Load target image



Operation Step

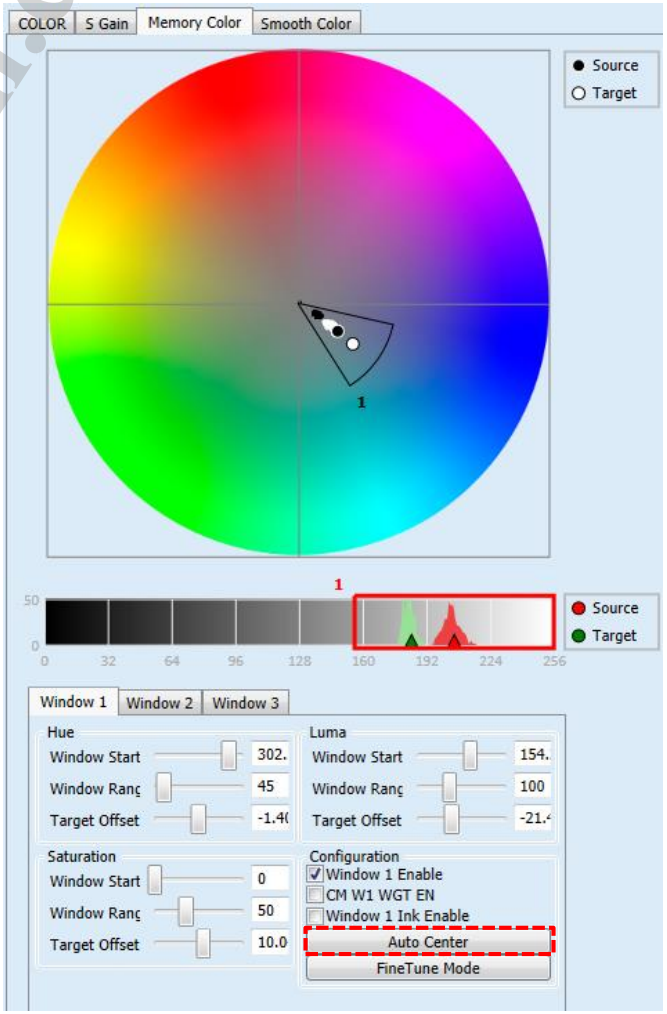
5) Crop ROI



Input

Target

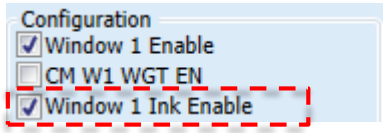
6) Press “Auto Center”



Operation Step

7) Check ink

- After confirm ROI is included, disable ink mode.



Ink mode

8) Check artifact effect

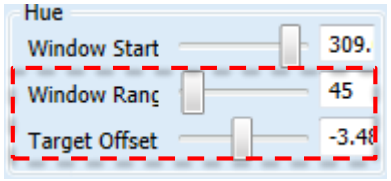
- Contour

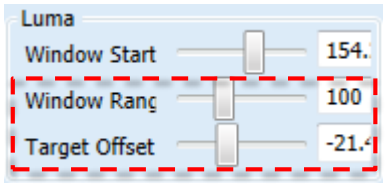
w/o Weighting

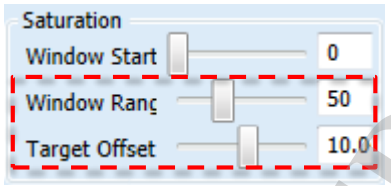
Operation Step

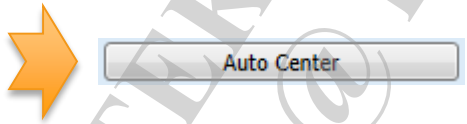
9) Fine tune window area

- Enlarge Window Range
- Adjust Target Offset
  - Target Offset ↑ , stronger effect



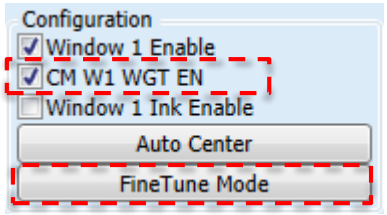




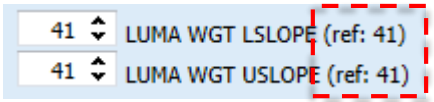


10) Fine tune effect

- Enable Weighting function
- Click FineTune Mode

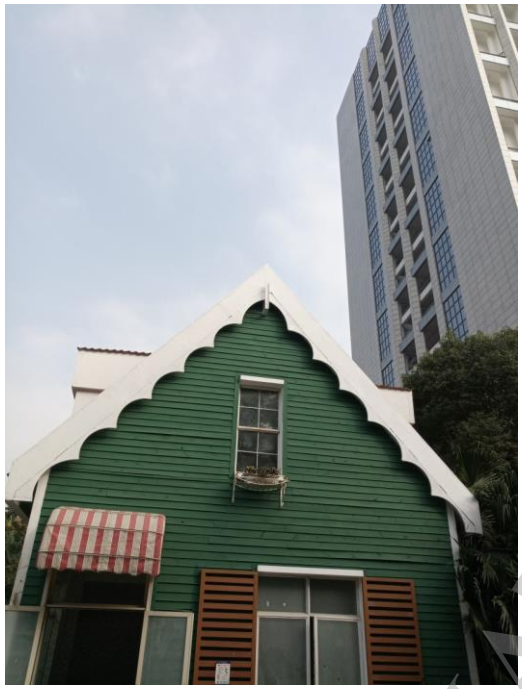


- Adjust WGT LSLOPE and USLOPE
  - SLOPE ↑ , stronger effect



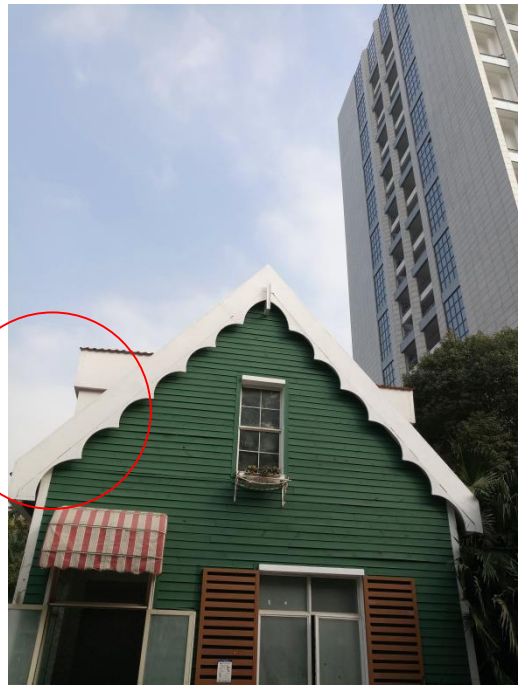
recommend value

Simulation Result



Input

w/o Weighting



w/ Weighting

everyday genius