ENGINEERING SPECIFICATION

Bismuth Display Calibration ERS

OBJECTIVES:	Detail the Engineering Requirements Specification (ERS) for the Bismuth display calibration including the technical and design conditions to be met. The ERS is used in conjunction with Reliability Requirements, Mechanical Control Outline (MCO), Outgoing Quality Control, and First Article Inspection documents for a comprehensive view of the adapter requirements.
BUILD LEVEL:	Proto1
AUTHORS:	Yencheng Peter Kim (hckim@google.com) Steven Lee (seungjunlee@google.com)
APPROVED BY:	
REVISION:	
RELEASE DATE:	10/06/2021
AGILE CONTROL:	xxx-xxxxx-xxx
NOTES:	This specification assumes that the reader has some familiarity with display test methodology, design and manufacturing. This document is intended for: Contract manufacturers who are involved in the design, assembly, and test of the display panel. Google engineers who are designing the display for Bismuth. Google engineers who are designing electrical or mechanical assemblies connecting to, or interacting with, the display system.

Revision History

Date	Rev. No	Contents	Authors
10/06/2021	0		Peter Kim

Table of Contents

1. Reference Documents

- TBD
- TBD
- TBD

2. Test Coverage and Specification

2.1 backlight calibration

Detailed test coverage and limit change history can be found at: go/z1-testcoverage

No.	Test Item	Description	Test Command
1	Backlight_calibration	Measure @ LCD bl current 0%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-lcd probe $ ./z1_fct -p /dev/ttyUSB0 shell fct-bl init $ ./z1_fct -p /dev/ttyUSB0 shell fct-lcd fill 0xffffff $ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 0
2		Measure @ LCD bl current 10%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 10
3		Measure @ LCD bl current 25%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 25
4		Measure @ LCD bl current 50%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 50
5		Measure @ LCD bl current 75%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 75
6		Measure @ LCD bl current 85%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 85
7		Measure @ LCD bl current 90%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 90
8		Measure @ LCD bl current 95%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 95
9		Measure @ LCD bl current 97%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 97
10		Measure @ LCD bl current 98%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 98
11		Measure @ LCD bl current 100%	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 100
12		write measured raw brightness data into device	$ ./z1_fct -p /dev/ttyUSB0 shell env set bl_raw_samples + value_list_str
13		Fit a,b: y = a*(e^bx - 1)	
14		Write coefficient a into device	$ ./z1_fct -p /dev/ttyUSB0 shell sysenv set LCD_BRIGHTNESS_COEFFICIENT = xxx
15		Write coefficient b into device	$ ./z1_fct -p /dev/ttyUSB0 shell sysenv set LCD_BRIGHTNESS_EXP_COEFFICIENT = xxx
16	backlight_validation	Measure @ brightness 0 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 0
17		Measure @ brightness 20 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness

			20
18		Measure @ brightness 50 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 50
19		Measure @ brightness 100 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 100
20		Measure @ brightness 150 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 150
21		Measure @ brightness 200 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 200
22		Measure @ brightness 250 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 250
23		Measure @ brightness 300 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 300
24		Measure @ brightness 350 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 350
25		Measure @ brightness 400 nits	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_brightness 400
26	center_chromaticity (chromaticity is tested at 20mA)	set backlight current at 20mA	$ ./z1_fct -p /dev/ttyUSB0 shell fct-bl set_current 96
27		Red (x, y)	# Red $ ./z1_fct -p /dev/ttyUSB0 shell fct-lcd fill 0xff0000
28		Green (x, y)	# Green $ ./z1_fct -p /dev/ttyUSB0 shell fct-lcd fill 0x00ff00
29		Blue (x, y)	# Blue $ ./z1_fct -p /dev/ttyUSB0 shell fct-lcd fill 0x0000ff
30		White (x,y)	# White $ ./z1_fct -p /dev/ttyUSB0 shell fct-lcd fill 0xffffff

2.2 backlight calibration

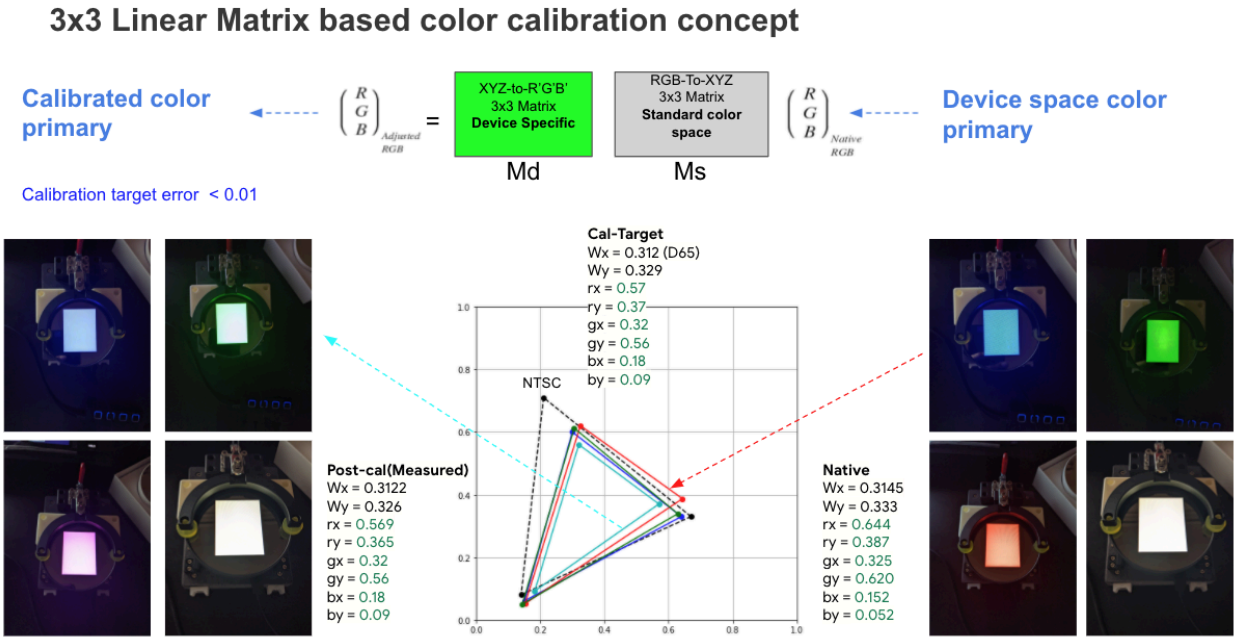
Backlight calibration target is defined by combining module IQC build data with lens transmittance tolerance. Detailed analysis link can be found here: LINK. Since each lens has a different transmittance level and variation, each lens should use a different target luminance level for 100% current level. P1 tentative POR target is defined as follows.

	Black	Silver	Warm gold
Target luminance	97 nits	307 nits	427 nits

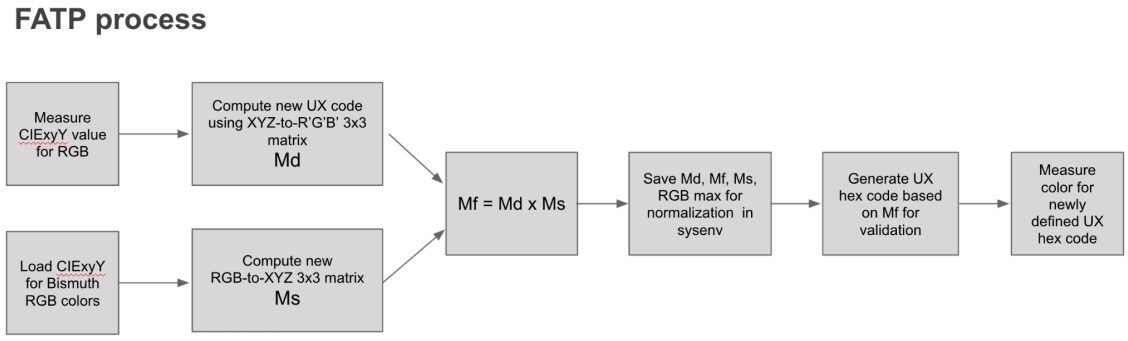
2.3 Color calibration

2.2.1 Introduction

- Thermostat has larger color variation due to different lens configs, lens process variation and display panel variation
  - Bismuth with warm gold lens will cause larger color variation than Z1 project did
- Existing per-config color code can't reduce the wide color variation
  - Hard to defining color code for each lens config without knowing distribution



2.3.2 Test Flow Overview



Note: The adjusted RGB is in linear space. It needs to be converted to digital count considering the gamma 2.2. In other words, the new UX RGB code will be as

2.3.3 Calibration Output (ETC : 10/31)

1. Md(XYZ to RGB matrix) = [x, x, x ; x, x, x; x, x, x] , matrix element x = floating point number (32 bits)
2. Ms (RGB to XYZ matrix) = [x, x, x ; x, x, x; x, x, x] , matrix element x = floating point number (32 bits)
3. Mf (RGB to RGB matrix) = [x, x, x ; x, x, x; x, x, x] , matrix element x = floating point number (32 bits)
4. RGB max for normalization: 1 scalar , floating point (32 bits)
   Comment: Need FCT comment for calibration process.

2.4 Verification patterns and spec

G255 White (x, y)

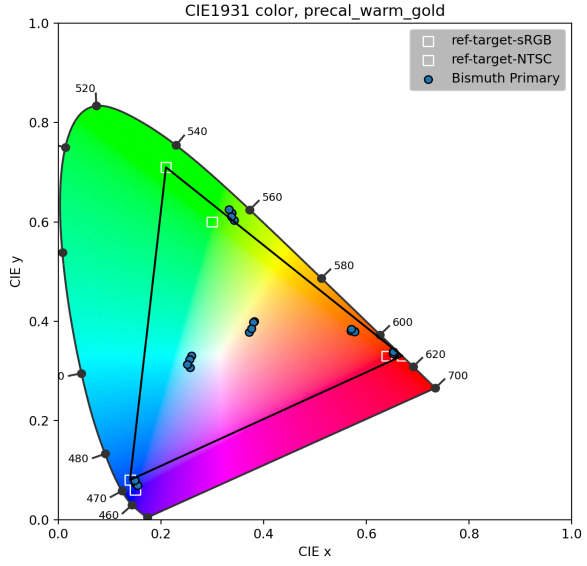
Lens info.	Tongda Selen Silver	CIExyY	Tongda Selen Black	CIExyY	Tongda Selen Gold	CIExyY
Colors	**Silver**		**Black**		**Warm Gold**	
White	#FFFFFF	0.3072 +/-0.01 0.3227 +/-0.01	#FFFFFF	0.3136 +/-0.01 0.3401 +/-0.01	#FFFFFF	0.3778 +/-0.01 0.3903 +/-0.01
Orange	#FF7343		#FF6632		#FF905C	
Blue	#45A6FF	0.149 +/-0.01 0.0567 +/-0.01	#56AEFE	0.1496 +/-0.01 0.0530 +/-0.01	#4EAAFF	0.1547 +/-0.01 0.0735 +/-0.01
Green	#40D267	0.3292 +/-0.01 0.6038 +/-0.01	#3CDF68	0.3234 +/-0.01 0.6227 +/-0.01	#40D267	0.3377 +/-0.01 0.6031 +/-0.01
Green - Apollo	#28FF8B		#28FF8B		#00F17A	
Red	#F83538	0.6393 +/-0.01 0.3396 +/-0.01	#FF4548	0.64 +/-0.01 0.3306 +/-0.01	#F83538	0.6520 +/-0.01 0.3366 +/-0.01
Yellow	#FFDB55		#FFDB55		#F6D455	
Maroon (Air)	#CA293A		#E94052		#B12D2D	
Purple (Air)	#BB3E90		#D445A3		#A53E90	
Gray - Light	#969DA5		#A0A7AF		#A0A7AF	

Gray - Dark	#79838C		#86919B		#80868B	
Black	#000000		#000000		#000000	

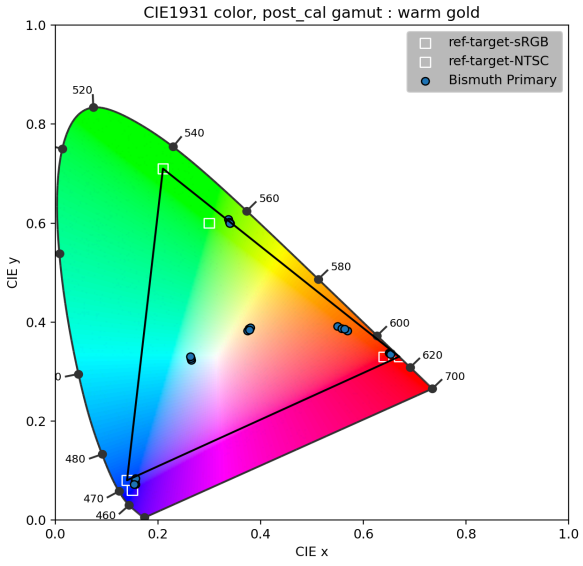
3. Lab validation results

- Calibration Result Summary

Pre Cal

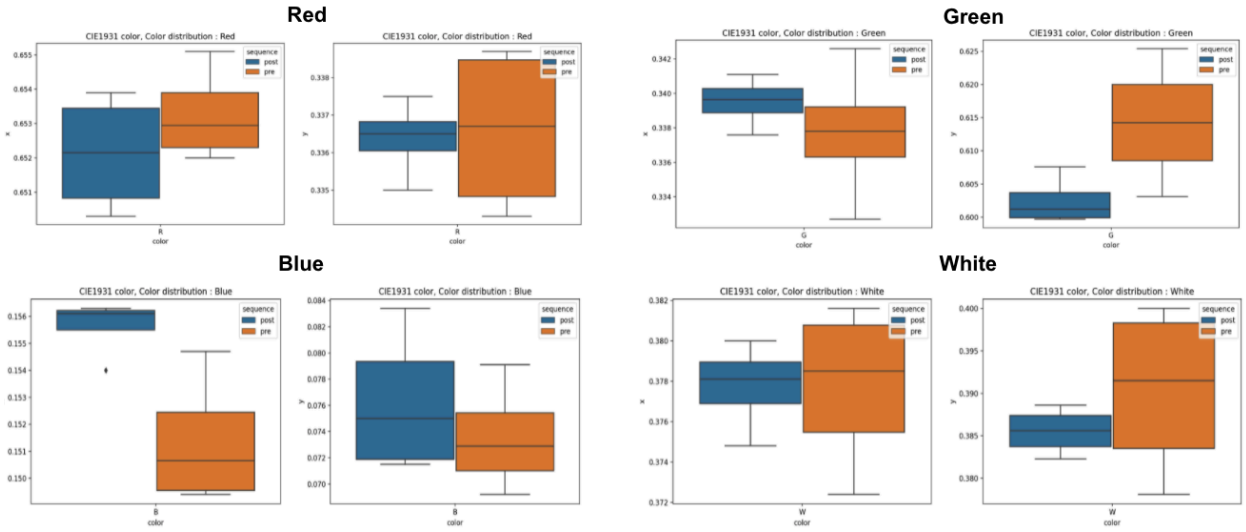


Post Cal



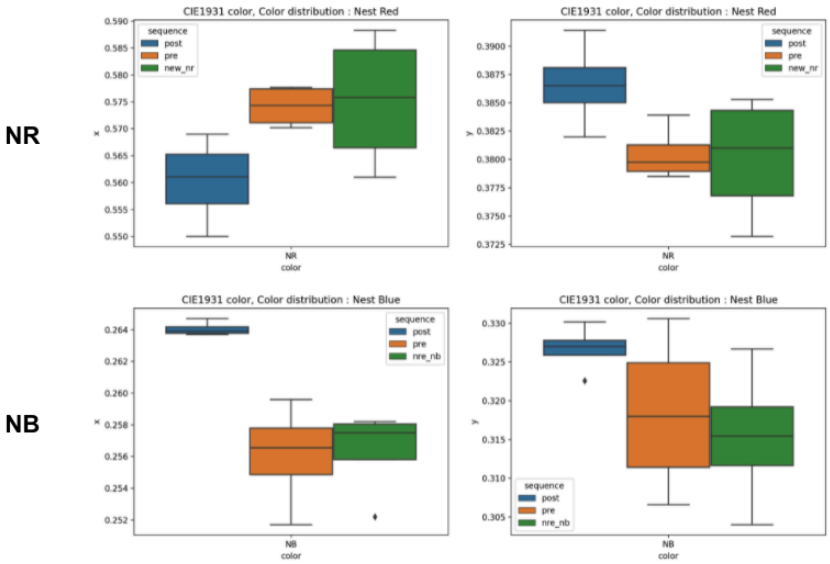
Pre vs Post Cal Color distribution : R,G,B

- Post calibration color (blue) distribution is improved overall for all primary and white colors



Pre vs Post Cal Color distribution : NR, NB

- Chose Native NR/NB color as target so new NR/NB distribution is larger than Pre-cal color
  - Expect narrower distribution once adopting single NR/NB color target



		Pre				Post				New_NR/NB (Stored in Device)			
unit_#	color	hex	x	y	Y	hex	x	y	Y	hex	x	y	Y
0	NR	ff6240	0.5702	0.3804	115.4	fb7646	0.55	0.3914	118	ff7142	0.561	0.384	114.23

	NB	4aa8ee	0.2572	0.3066	111.7	52b5e0	0.264	0.327	125	55abe8	0.257	0.304	114
2	NR	ff6240	0.5773	0.3785	110.3	f8724a	0.569	0.382	110	fe7341	0.5683	0.3853	145
	NB	4aa8ee	0.2559	0.3230	100.62	59aeee	0.2647	0.3226	109.7	4dacec	0.2522	0.3142	142
1	NR	ff6240	0.5714	0.3839	146.8	fa774a	0.5581	0.387	153	f8663f	0.5883	0.3732	106
	NB	4aa8ee	0.2517	0.313	140.1	59b5e8	0.2638	0.327	161	489ddd	0.258	0.3167	88.7
3	NR	ff6240	0.5777	0.3791	111.6	f3704b	0.564	0.386	106.7	f4653f	0.5834	0.378	100.6
	NB	4aa8ee	0.2596	0.3306	106.7	54a9ee	0.2637	0.3302	108.3	479cdd	0.2582	0.3267	90.4

4. FATP Display Calibration Station Requirement

This station will check the important display performance parameters such as colors, brightness, and contrast ratio. These parameters defines the important and basic display performance. Thus this station requires highly accurate measurement instrument. In the test flow, unit should be tested in display color and brightness station first.

Instrument condition

- Color and luminance calibration is required based on high accuracy spectro(radio)meter such as PR6XX, CAS140 or equivalent instrument
    - Color accuracy required for calibration instrument: +-0.002
    - Precision required for calibration instrument: 0.00015
    - Luminance accuracy required for calibration instrument : <3%
- After calibration, calibrated instrument should be verified with 3x randomly selected units for verification
    - Color accuracy required for calibration instrument: +-0.004
    - Luminance accuracy required for calibration instrument : <5%
- Instrument should be regularly audited to correct any measurement data drift.
    - Engineering build: Every build
    - Mass production: Every two month (it can be adjusted based on the build data)

Display color calibration supplementary documents and algorithm

Bismuth calibration algorithm high level
Deck1
Deck2

Algorithm code link : Run “Bismuth_gamut_per_lens.py”
The following function generates 3x3 matrix.

```
# This function takes care of matrix generation
# M, Md, Mf = LM_gen_per_len(df_gold_r, df_gold_g, df_gold_b, df_gold_w, unit_number)
```

Instrument condition

- Luminance calibration is required based on high accuracy spectro(radio)meter such as PR6XX, CAS140 or equivalent instrument
    - Color accuracy : +-0.002
    - Precision: 0.00015
    - Luminance accuracy : <3%

- After calibration, calibrated instrument should be verified with 3x randomly selected units for verification and GR&R is required
    - Color accuracy required for calibration instrument: +-0.004
    - Luminance accuracy required for calibration instrument : <5%
    - GR&R condition: 3 Operators, 3 times and 10x samples

- Camera should be regularly audited to correct measurement data drift.
    - Engineering build: Beginning of each build
    - Mass production: Every two month (it can be adjusted based on the build data)