OV7725 Camera Module

Software Application Note

Ymïr

Last Modified: Dec 12th, 2007

Document Revision: 1.04

OmniVision Technologies, Inc. reserves the right to make changes without further notice to any product herein to improve reliability, function or design. OmniVision does not assume any liability arising out of the application or use of any project, circuit described herein; neither does it convey any license under its patent nor the right of others.

Sensor datasheet is the official document of OmniVision. Software/hardware/dual camera applicaton notes are application guide lines for reference. If there are any difference between sensor datasheet and application notes, please follow sensor datasheet and kindly report the difference to OVT FAE.

Table of Contents

1. Select Output format

 OV7725 support 4 output formats: YCbCr, RGB565/RGB555, Bayer raw RGB and Processed raw RGB. How to choose the right output format for camera phone design or other applications? Let's look at the backend chip first.

 The general diagram of backend chip is as below:
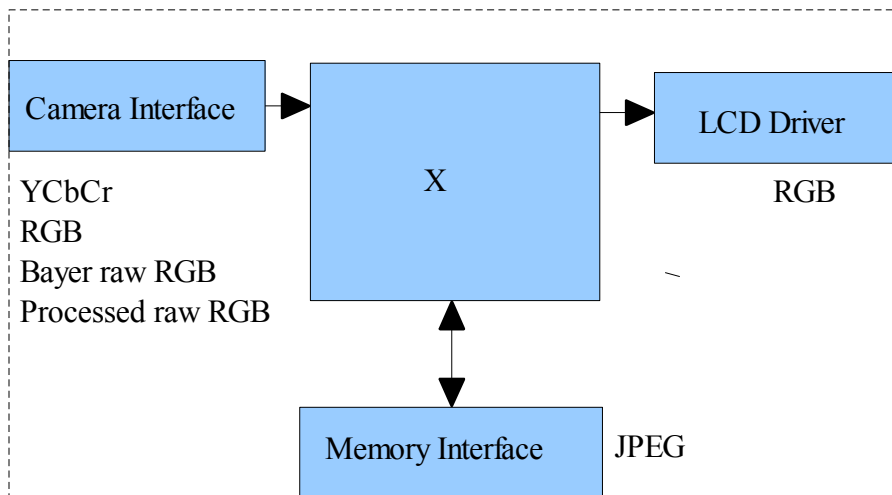
 The data format at LCD driver are always RGB. For example, RGB444, RGB565, RGB555, RGB888 etc. The data format and memory interface are always Compressed. The Compression data is compressed from YCbCr data. So Both RGB and YCbCr data are needed inside the backend chip. The "X" block is different for different backend chips.

1.1 Backend with full ISP

 This kind of backend has full ISP. It takes raw RGB input, doing interpolation to generate RGB24 and doing translation to generate YCbCr. This kind of backend could take either Bayer raw RGB or processed raw RGB.

 The advantage of processed raw RGB over Bayer raw RGB is the output data are processed. Sensor functions such as defect pixel correction, lens correction, gamma, color matrix, de-noise, sharpness, BLC etc. could be applied. Since the life time of backend chip is longer than image sensor, sometimes backend chips could not fix defects of new sensors if taken Bayer raw RGB. But the defects of new sensors could be fixed in processed raw RGB output.

 If backend take Bayer raw RGB format from sensor, all the image process operations such as defect pixel correction, lens correction, gamma, color matrix, de-noise, sharpness, BCL etc should be done by backend. If backend take processed raw RGB format from sensor, the image process operations such as defect pixel correction, lens correction, gamma, color matrix, de-noise,

sharpness, BCL etc could be done either inside sensor or by backend chips. In other words, user could select the image process operation be done by which side.

1.2 Backend with YCbCr ISP

This kind of backend has ISP, but could take only YCbCr format. The ISP could convert YCbCr to RGB format for LCD display and compress YCbCr for storage.

1.3 Backend without ISP

This kind of backend doesn't have ISP built-in. It can not convert from one format to another by hardware. Actually the format conversion is done by software. There are 3 possible solution for this kind of backend chips.

a. Sensor output YCbCr. Backend convert YCbCr to RGB for display by software.
b. Sensor output RGB565. Backend convert RGB565 to YCbCR for compression.
c. Sensor output RGB565 for preview, output YCbCr for capture (compression).

Solution a. provide the best picture quality. Since the input data is 24-bit RGB equivalent. It could converted to RGB888 for LCD display. Solution b. provide the worst picture quality. Since the input data is only 16-bit RGB565, even it is converted to YCbCr, the color depth is still 16-bit. The solution c. provide similar picture quality as solution a. But since preview is RGB565, capture is YCbCr, preview picture may looks a little different than capture picture.

1.4 Equations to Convert from One Format to Another

YCbCr to RGB24

$Y = 0.299R + 0.587G + 0.114B$

$Cb = 0.568(B\text{-}Y) + 128 = -0.172R - 0.339G + 0.511B + 128$

$Cr = 0.713(R\text{-}Y) + 128 = 0.511R - 0.428G - 0.083B + 128$

$Y = ((77 * R + 150 * G + 29 * B) >> 8);$

$Cb = ((-43 * R - 85 * G + 128 * B) >> 8) + 128;$

$Cr = ((128 * R - 107 * G - 21 * B) >> 8) + 128;$

RGB24 to YCbCr
$R = Y + 1.371(Cr - 128)$

$G = Y - 0.698(Cr - 128) - 0.336(Cb - 128)$

$B = Y + 1.732(Cb - 128)$

R = Y + (351*(Cr – 128)) >> 8

G = Y – (179*(Cr – 128) + 86*(Cb – 128))>>8

B = Y + (443*(Cb – 128)) >> 8

2. Select Output Resolution

2.1 Backend with ISP

If Backend chip has built-in ISP (Full ISP or YCbCr ISP), the ISP could do image scale. So OV7725 outputs only VGA format. ISP scaled VGA image to other resolution that mobile device needed.

2.2 Backend without ISP

If backend chip doesn't have image scale capability, then the LCD scaler of OV7725 must be used to scale output resolution exactly the LCD size. For example, if the LCD size is 176x220, then the LCD scaler will scale the output size to 176x220.

3. Adjust frame rate

The recommended frame rates are 30fps and 15fps for 60Hz light environment, 25fps and 14.3fps for 50Hz light environment. The frame rate for night mode is lower, we'll discuss night mode later.

Reference settings for above frame rates are listed below.

3.1 Frame Rate Adjustment for 24Mhz input clock

30 fps, PCLK = 24Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x00);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

15 fps, PCLK = 12Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x03);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x00);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

25fps, PCLK = 24Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x66);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

14.3fps, PCLK = 12Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x03);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x1a);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

3.2 Frame Rate Adjustment for 26 Mhz input clock

30 fps, PCLK = 26Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);

write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x2b);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

15 fps, PCLK = 13Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x03);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x2b);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

25fps, PCLK = 26Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x99);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

14.3fps, PCLK = 13Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x03);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x46);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

3.3 Frame rate adjustment for 13 Mhz input clock

30 fps, PCLK = 26Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x00);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x2b);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

15 fps, PCLK = 13Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x2b);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

25fps, PCLK = 26Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x00);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);
write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x99);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

14.3fps, PCLK = 13Mhz

SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0d, 0x41);
write_SCCB(0x2a, 0x00);

write_SCCB(0x2b, 0x00);
write_SCCB(0x33, 0x46);
write_SCCB(0x34, 0x00);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);
write_SCCB(0x0e, 0x65);

4. Night Mode

There are 2 types of settings for night mode. One type is set to fixed low frame rate, for example 3.75fps. The other type is set to auto frame rate, for example from 30fps to 3.75fps. When environment is bright, the frame rate is increased to 30fps. When environment is dark, the frame rate is decreased to 3.65fps.

4.1 Night Mode with Fixed Frame Rate

For 24Mhz/26Mhz Clock Input

3.75fps night mode for 60Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x0f);
write_SCCB(0x0e, 0x65);

3.125fps night mode for 50Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x0f);
write_SCCB(0x0e, 0x65);

For 13Mhz Clock Input

3.75fps night mode for 60Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x07);
write_SCCB(0x0e, 0x65);

3.125fps night mode for 50Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x07);
write_SCCB(0x0e, 0x65);

4.2 Night Mode with Auto Frame Rate

For 24Mhz/26Mhz Clock Input

30fps ~ 3.75fps night mode for 60Hz light environment
SCCB_salve_Address = 0x42;

write_SCCB(0x11, 0x01);
write_SCCB(0x0e, 0xf5);

15fps ~ 3.75fps night mode for 60Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x03);
write_SCCB(0x0e, 0xe5);

25fps ~ 3.125fps night mode for 50Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0e, 0xf5);

14.3fps ~ 3.6fps night mode for 50Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x03);
write_SCCB(0x3b, 0xe5);

For 13Mhz Clock Input

30fps ~ 3.75fps night mode for 60Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x00);
write_SCCB(0x0e, 0xf5);

15fps ~ 3.75fps night mode for 60Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0e, 0xe5);

25fps ~ 3.125fps night mode for 50Hz light environment
SCCB_salve_Address = 0x42;
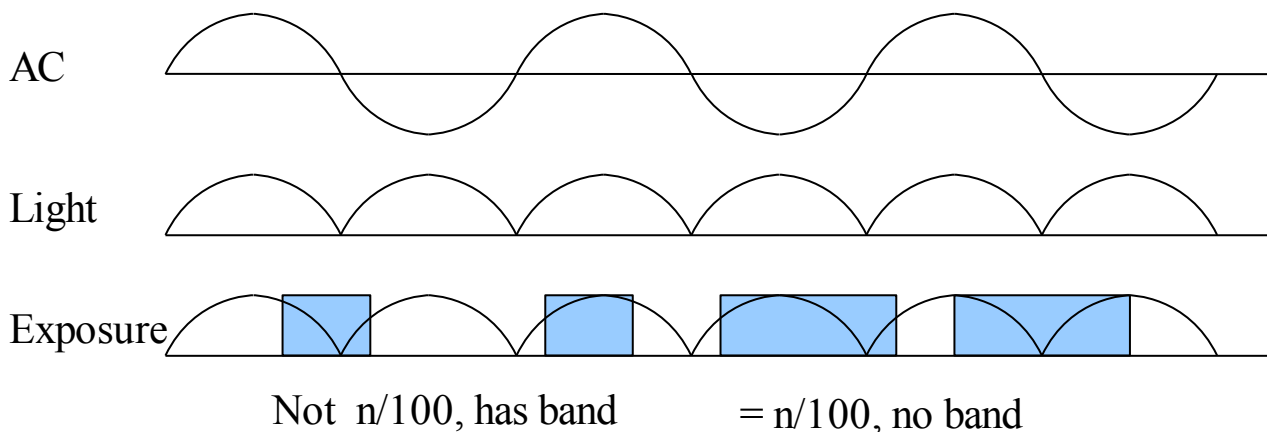write_SCCB(0x11, 0x00);
write_SCCB(0x0e, 0xf5);

14.3fps ~ 3.6fps night mode for 50Hz light environment
SCCB_salve_Address = 0x42;
write_SCCB(0x11, 0x01);
write_SCCB(0x0e, 0xe5);

Note:
   When OV7725 is set to low frame rate, there may be many white pixels shown on LCD of mobile phone or on PC diaplay.

5. Remove Light Band

5.1 Light Band

AC

Light

Exposure

Not n/100, has band   = n/100, no band

The strength of office light is not even. It changes with AC frequency. For example, if the AC frequency is 50Hz, the light changes strength at 100hz.
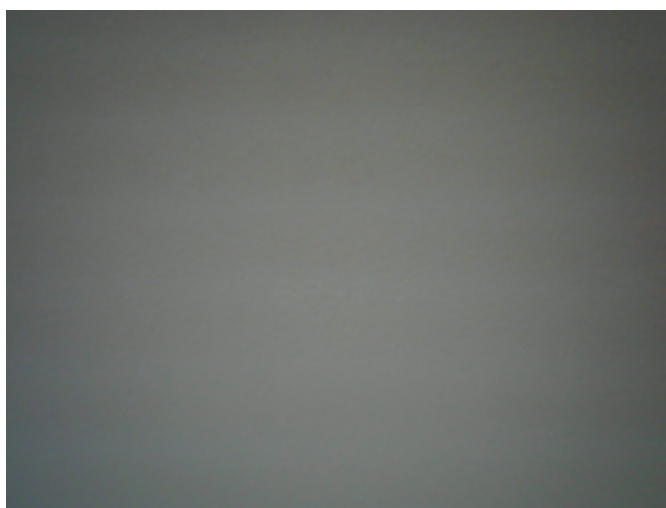
5.2 Remove Light band

Light band is removed by set exposure to n/100 (n/120 for 60Hz)seconds. The banding filter value tell OV7725 how many lines is 1/100 (1/120 for 60Hz) seconds.

5.3 Select Banding Filter by Region Information

The region information of mobile phone could be used to select banding filter values. A light frequency table is built to indicate which region uses 50Hz light and which region uses 60Hz light. When region information is got, the light frequency information could be get from the table.

Different frame rate could be used for different light frequency. So the frame rate is optimized for

both 50hz light condition and 60hz light condition.

Banding Filter Setting for 24Mhz Input Clock

30fps for 60Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x7f); //60Hz banding filter
write_SCCB(0x23, 0x03); //4 step for 60hz

15fps for 60Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x3f); //60Hz banding filter
write_SCCB(0x23, 0x07); //8 step for 60hz

25fps for 50Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x98); //50Hz banding filter
write_SCCB(0x23, 0x03); //4 step for 50hz

14.3fps for 50Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x4c); //50Hz banding filter
write_SCCB(0x23, 0x06); //7 step for 50hz

Banding Filter Setting for 13Mhz/26Mhz Input Clock
30fps for 60Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x89); //60Hz banding filter
write_SCCB(0x23, 0x03); //4 step for 60hz

15fps for 60Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x44); //60Hz banding filter
write_SCCB(0x23, 0x07); //8 step for 60hz

25fps for 50Hz light frequency
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0xa5); //50Hz banding filter
write_SCCB(0x23, 0x03); //4 step for 50hz

14.3fps for 50Hz light frequency

SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //banding filter enable
write_SCCB(0x22, 0x52); //50Hz banding filter
write_SCCB(0x23, 0x06); //7 step for 50hz

5.4 When Light Band can not be Removed

Normally the light band is removed by banding filter.

But there is some special conditions such as mix light of sun light and office light, take picture of florescent light, the light band can not removed. The reason is the exposure time is less than 1/100 second for 50hz light environment and less than 1/120 second for 60hz light environment, so the light band can not be removed.

The light band is this conditions could not be removed for all CMOS sensors, not only OV7725. So there is no way to remove light band in this condition.

6. White Balance

OV7725 support simple white balance and advanced balance.

6.1 Simple White Balance

Simple white balance assume "gray world". Which means the average color of world is gray. It is true for most environment.

Advantage of simple AWB
Simple white balance is not depend on lens. A general setting for simple white balance could applied for all modules with different lens.

Disadvantage of simple AWB
The color is not accurate in conditions where "gray world" not true. For example the background has a huge red, blue or green etc. the color of the foreground is not accurate. If the camera target single color such as red, blue, green, the simple white balance will make the single color gray.

Settings
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //AWB on
write_SCCB(0x69, 0x5d); // Simple AWB

6.2 Advanced White Balance

Advanced white balance uses color temperature information to detect white area and do the white balance.

Advantage of Advanced AWB

Color is more accurate than simple white balance. Even the background is single color, the camera will not make the single color gray.

Disadvantage of Advanced AWB

Advanced white balance setting is depend on lens. The setting must be adjusted for every module with new lens. The adjustment must be done by OmniVision FAE in optical lab with some optical equipment such as light box, color checker etc.

Settings

Contact with OmniVision local FAE.

6.3 How to select?

Generally, for low resolution camera module such as CIF, VGA and 1.3M, simple AWB is selected. For high resolution camera module such as 2M, 3M, advanced AWB is selected.

7. Defect Pixel Correction

Defect pixel include dead pixel and wounded pixel.

Dead pixel include white dead pixel and black dead pixel. White dead pixel is always white no matter the actual picture is bright or dark. Black dead pixel is always black no matter the actual picture is bright or dark.

Wounded pixel may change with light, but not as much as normal pixel. White wounded pixels are much brighter then normal pixels, but not complete white. Black wounded pixels are much darker than normal pixels, but not complete black.

OV7725 has built-in defect pixel correction function. If OV7725 output YCbCr, RGB565, Processed raw RGB, the defect pixel correction function could be enabled to fix defect pixels. But if Bayer raw RGB is used, the defect pixel correction function of sensor could not be used. The defect pixel correction of backend chip should be used instead.

Please pay attention to the defect pixel correction function of backend chip. Some backend chip may not be able to correct all defect pixels of OV7725.

8. BLC

The function of Black Level Calibration (BLC) is to product accurate color in the dark area of picture. There is automatic BLC function built-in OV7725. It should always be turned on.

9. Video Mode

Video mode need high frame rate, usually fixed 15fps. There is no night mode for video mode.

10. Digital zoom

If OV7725 output image smaller than QVGA, it may support digital zoom. For example

VGA	not digital zoom supported
QVGA	1x, 2x
QQVGA	1x, 2x, 4x
QCIF	1x, 1.8x
QQCIF	1x, 2x, 3.6x

If backend chip support scale up, then more zoom level could be supported.

11. OV7725 Functions

11.1 Light Mode

Auto
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //AWB on
write_SCCB(0x0e, 0x65);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);

Sunny
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xfd); //AWB off
write_SCCB(0x01, 0x5a);
write_SCCB(0x02, 0x5c);
write_SCCB(0x0e, 0x65);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);

Cloudy
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xfd); //AWB off
write_SCCB(0x01, 0x58);
write_SCCB(0x02, 0x60);
write_SCCB(0x0e, 0x65);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);

Office
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xfd); //AWB off
write_SCCB(0x01, 0x84);
write_SCCB(0x02, 0x4c);
write_SCCB(0x0e, 0x65);
write_SCCB(0x2d, 0x00);

write_SCCB(0x2e, 0x00);

Home
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xfd); //AWB off
write_SCCB(0x01, 0x96);
write_SCCB(0x02, 0x40);
write_SCCB(0x0e, 0x65);
write_SCCB(0x2d, 0x00);
write_SCCB(0x2e, 0x00);

Night
SCCB_salve_Address = 0x42;
write_SCCB(0x13, 0xff); //AWB on
write_SCCB(0x0e, 0xe5);

11.2 Color Saturation

The color saturation of OV7725 could be adjusted. High color saturation would make the picture looks more vivid, but the side effect is the bigger noise and not accurate skin color.

Saturation + 4
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x80);
write_SCCB(0xa8, 0x80);

Saturation + 3
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x70);
write_SCCB(0xa8, 0x70);

Saturation + 2
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x60);
write_SCCB(0xa8, 0x60);

Saturation + 1
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x50);
write_SCCB(0xa8, 0x50);

Saturation 0
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x40);
write_SCCB(0xa8, 0x40);

Saturation -1
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x30);
write_SCCB(0xa8, 0x30);

Saturation - 2
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x20);
write_SCCB(0xa8, 0x20);

Saturation - 3
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x10);
write_SCCB(0xa8, 0x10);

Saturation - 4
SCCB_salve_Address = 0x42;
write_SCCB(0xa7, 0x00);
write_SCCB(0xa8, 0x00);

11.3 Brightness

The brightness of OV7725 could be adjusted. Higher brightness will make the picture more bright. The side effect of higher brightness is the picture looks frogy.

Brightness +4
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x48);
write_SCCB(0xab, 0x06);

Brightness +3
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x38);
write_SCCB(0xab, 0x06);

Brightness +2
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x28);
write_SCCB(0xab, 0x06);

Brightness +1
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x18);
write_SCCB(0xab, 0x06);

Brightness 0
SCCB_salve_Address = 0x42;

write_SCCB(0x9b, 0x08);
write_SCCB(0xab, 0x06);

Brightness -1
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x08);
write_SCCB(0xab, 0x0e);

Brightness -2
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x18);
write_SCCB(0xab, 0x0e);

Brightness -3
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x28);
write_SCCB(0xab, 0x0e);

Brightness -4
SCCB_salve_Address = 0x42;
write_SCCB(0x9b, 0x38);
write_SCCB(0xab, 0x0e);

11.4 Contrast

The contrast of OV7725 could be adjusted. Higher contrast will make the picture sharp. But the side effect is loosing dynamic range.

Contrast +4
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x30);

Contrast +3
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x2c);

Contrast +2
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x28);

Contrast +1
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x24);

Contrast 0
SCCB_salve_Address = 0x42;

write_SCCB(0x9c, 0x20);

Contrast -1
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x1c);

Contrast -2
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x18);

Contrast -3
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x14);

Contrast -4
SCCB_salve_Address = 0x42;
write_SCCB(0x9c, 0x10);

11.5 Special effects

OV7725 support some special effects such as B/W, negative, sepia, bluish, redish, greenish etc. If users need other special effects, it should be supported by backend chips.

Normal
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x06);
write_SCCB(0x60, 0x80);
write_SCCB(0x61, 0x80);

B&W
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x26);
write_SCCB(0x60, 0x80);
write_SCCB(0x61, 0x80);

Bluish
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x1e);
write_SCCB(0x60, 0xa0);
write_SCCB(0x61, 0x40);

Sepia
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x1e);
write_SCCB(0x60, 0x40);
write_SCCB(0x61, 0xa0);

Redish
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x1e);
write_SCCB(0x60, 0x80);
write_SCCB(0x61, 0xc0);

Greenish
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x1e);
write_SCCB(0x60, 0x60);
write_SCCB(0x61, 0x60);

Negative
SCCB_salve_Address = 0x42;
write_SCCB(0xa6, 0x46);

12. Deal with Lens

12.1 Light fall off

Light fall off means the corner of image is darker than center of image. It is caused by the lens. The lens shading correction function of OV7725 could be turned on to compensate the corner brightness and make the whole picture looks same bright.

12.2 Dark corner

Some lens may have dark corner. Dark corner means the color of picture looks almost black. It is not possible to correct dark corner with lens correction. So the module with dark corner is NG, it can not be used.

12.3 Resolution

The resolution of camera module depends on lens design, focus adjustment and sensor resolution as well. The focus adjustment is very important for camera module assembly.

For OV7725 the focus distance is about 50~60cm. The depth of field is about from 25~30cm to infinite. If checking resolution of camera module, the resolution chart should be placed 50~60 cm away.

12.4 Optical contrast

The optical contrast of lens is very important to picture quality. If the optical contrast of lens is not good, the picture would looks forgy. Though it could be improved by increase the sensor contrast to make the picture sharper, the higher sensor contrast would make the detail lost of dark area of the picture.

12.5 Lens Cover

   The lens cover is the cheapest part in optical path. But it could affect picture quality very much. The lens cover should be made with optical glass with AR coating at both side. Otherwise, the lens cover may cause sensitivity loss and/or stronger lens flare.

12.6 Lens Correction

   Lens Correction setting should be tunned with every module. Please contact with OmniVision local FAE foe lens correction tuning.

13. Reference Settings

13.1 YCbCr

```
//OV7720, YCbCr, VGA
//15fps at 24MHz input clock, 4x maximum gain
//2/27/2007

write_SCCB(0x12, 0x80); // Reset

Delay(2ms);

write_SCCB(0x3d, 0x03);
write_SCCB(0x17, 0x22);
write_SCCB(0x18, 0xa4);
write_SCCB(0x19, 0x07);
write_SCCB(0x1a, 0xf0);
write_SCCB(0x32, 0x00);
write_SCCB(0x29, 0xa0);
write_SCCB(0x2c, 0xf0);
write_SCCB(0x2a, 0x00);
write_SCCB(0x11, 0x01);

write_SCCB(0x42, 0x7f);
write_SCCB(0x4d, 0x09);
write_SCCB(0x63, 0xe0);
write_SCCB(0x64, 0xff);
write_SCCB(0x65, 0x20);
write_SCCB(0x66, 0x00);
write_SCCB(0x67, 0x48);

write_SCCB(0x13, 0xf0);
write_SCCB(0x0d, 0x41); // 0x51/0x61/0x71 for different AEC/AGC window
write_SCCB(0x0f, 0xc5);
write_SCCB(0x14, 0x11);
write_SCCB(0x22, 0x7f);
write_SCCB(0x23, 0x03);
write_SCCB(0x24, 0x40);
```

```
write_SCCB(0x25, 0x30);
write_SCCB(0x26, 0xa1);
write_SCCB(0x2b, 0x00);
write_SCCB(0x6b, 0xaa);
write_SCCB(0x13, 0xff);

write_SCCB(0x90, 0x05);
write_SCCB(0x91, 0x01);
write_SCCB(0x92, 0x03);
write_SCCB(0x93, 0x00);
write_SCCB(0x94, 0xb0);
write_SCCB(0x95, 0x9d);
write_SCCB(0x96, 0x13);
write_SCCB(0x97, 0x16);
write_SCCB(0x98, 0x7b);
write_SCCB(0x99, 0x91);
write_SCCB(0x9a, 0x1e);
write_SCCB(0x9b, 0x08);
write_SCCB(0x9c, 0x20);
write_SCCB(0x9e, 0x81);
write_SCCB(0xa6, 0x06);

//Gamma
write_SCCB(0x7e, 0x0c);
write_SCCB(0x7f, 0x16);
write_SCCB(0x80, 0x2a);
write_SCCB(0x81, 0x4e);
write_SCCB(0x82, 0x61);
write_SCCB(0x83, 0x6f);
write_SCCB(0x84, 0x7b);
write_SCCB(0x85, 0x86);
write_SCCB(0x86, 0x8e);
write_SCCB(0x87, 0x97);
write_SCCB(0x88, 0xa4);
write_SCCB(0x89, 0xaf);
write_SCCB(0x8a, 0xc5);
write_SCCB(0x8b, 0xd7);
write_SCCB(0x8c, 0xe8);
write_SCCB(0x8d, 0x20);

// for 30 fps, 60Hz
write_SCCB(0x33, 0x00);
write_SCCB(0x22, 0x7f);
write_SCCB(0x23, 0x03);

// for 25 fps, 50Hz
//write_SCCB(0x33, 0x66);
//write_SCCB(0x22, 0x99);
```

//write_SCCB(0x23, 0x03);

// Lens Correction, should be tuned with real camera module
write_SCCB(0x4a, 0x10);
write_SCCB(0x49, 0x10);
write_SCCB(0x4b, 0x14);
write_SCCB(0x4c, 0x17);
write_SCCB(0x46, 0x05);

write_SCCB(0x0e, 0x65);

13.2 RGB raw

Contact with OmniVision local FAE.

13.3 RGB565

Contact OmniVision local FAE