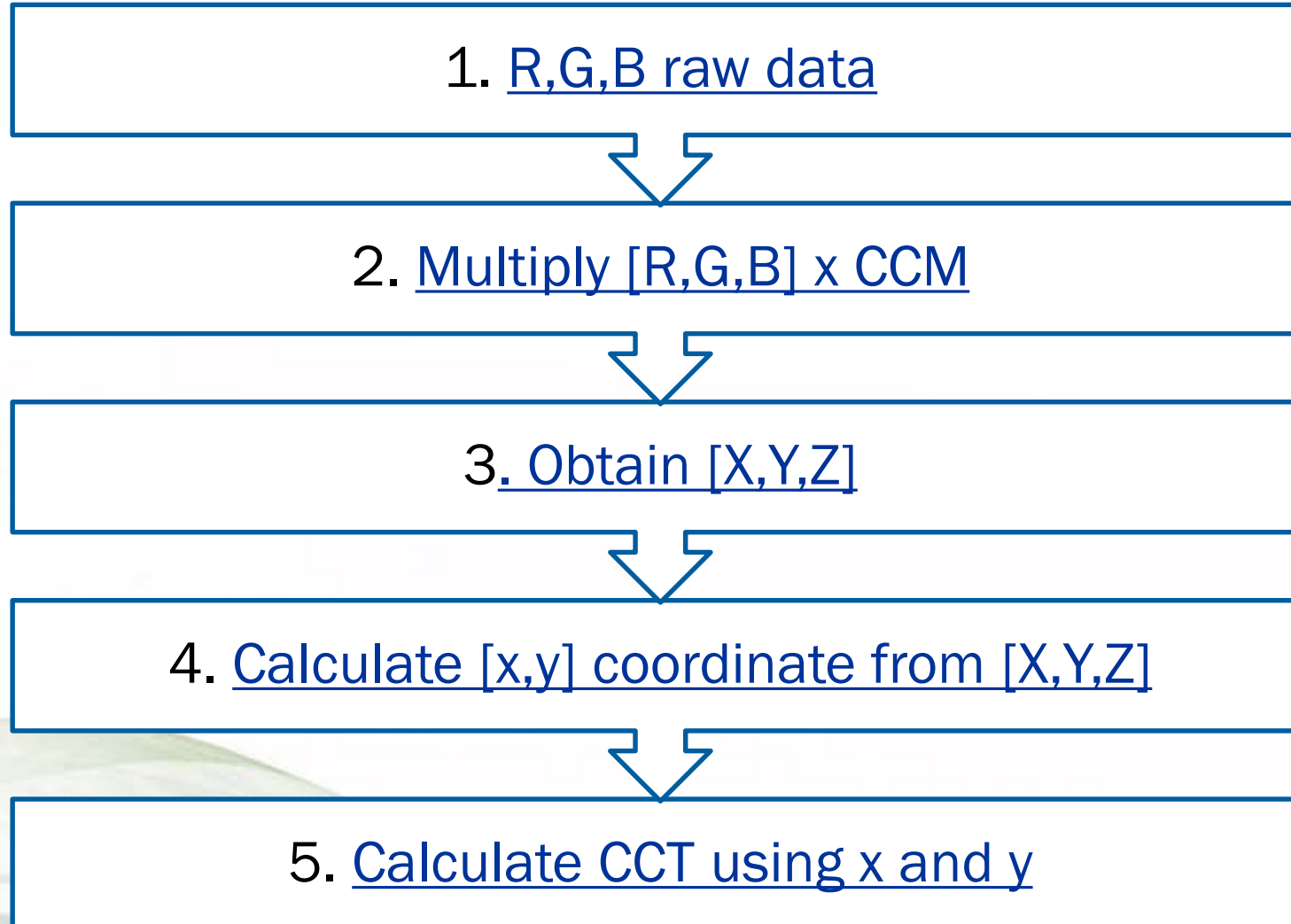


# ISL29125 CCT (Color Correlated Temperature) calculation

## CCT calculation diagram flow – pre-requisite: CCM (system dependant)



# 1. Raw data

- See datasheet p13

**Data Register (Address: 0x09, 0x0A, 0xB, 0xC, 0xD and 0xE)**

TABLE 20. CONFIGURATION-3

NAME	REGISTER ADDRESS		REGISTER BITS								DEFAULT	ACCESS
	DEC	HEX	B7	B6	B5	B4	B3	B2	B1	B0		
GREEN Data - Low Byte	9	0x09	GREEN[7]	GREEN[6]	GREEN[5]	GREEN[4]	GREEN[3]	GREEN[2]	GREEN[1]	GREEN[0]	0x00	RW
GREEN Data - High Byte	10	0x0A	GREEN[15]	GREEN[14]	GREEN[13]	GREEN[12]	GREEN[11]	GREEN[10]	GREEN[9]	GREEN[8]	0x00	RW
RED Data - Low Byte	11	0x0B	RED[7]	RED[6]	RED[5]	RED[4]	RED[3]	RED[2]	RED[1]	RED[0]	0x00	RW
RED Data - High Byte	12	0x0C	RED[15]	RED[14]	RED[13]	RED[12]	RED[11]	RED[10]	RED[9]	RED[8]	0x00	RW
RED Data - Low Byte	13	0x0D	BLUE[7]	BLUE[6]	BLUE[5]	BLUE[4]	BLUE[3]	BLUE[2]	BLUE[1]	RED[0]	0x00	RW
RED Data - High Byte	14	0x0E	BLUE[15]	BLUE[14]	BLUE[13]	BLUE[12]	BLUE[11]	BLUE[10]	BLUE[9]	RED[8]	0x00	RW

The ISL29125 has two 8-bit read-only registers to hold the higher and lower byte of the ADC value. The lower and higher bytes are accessed at address, respectively. For 16-bit resolution, the data is from D0 to D15; for 12-bit resolution, the data is from D0 to D11. The registers are refreshed after every conversion cycle. The default register value is 0x00 at power-on. Because all the register are double buffered the data is always valid on the data registers.

- It's the ADC count for each RGB channel

## 2. Multiply RGB x CCM

- **CCM = Color Correction Matrix**
- **The CCM needs to be calculated for the system on the final product**
  - Need to use the final opto-mechanical system
- **To calculate the CCM**
  - follow the procedure in the excel file embedded in the GUI package: “CCM calibration worksheet”
  - The customer will need 3 light sources + the final opto-mechanical system
  - We recommend to use at least 5 devices to have an improved statistical data for CCM (goal is to compensate for manufacturing variation of the ink, panel, placement, airgap...)

- **Example of CCM and calculation**

$(X,Y,Z) = \text{color correction matrix (CCM)} * (R,G,B)$

$$\begin{matrix} X \\ Y \\ Z \end{matrix} = \begin{matrix} 0.2241 & 1.029 & -0.3835 \\ 5.864*10^{-2} & 1 & -6.308*10^{-2} \\ -0.5194 & 0.3655 & 1.118 \end{matrix} \begin{matrix} R \\ G \\ B \end{matrix}$$

- **The CCM calculation needs to be for the 2 ranges of the device: low range and high range**

### 3. Obtain X,Y,Z

- **Example of CCM and calculation**

(X,Y,Z) = color correction matrix (CCM) \* (R,G,B)

- **CCM** = 
$$\begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix}$$

- **[X,Y,Z] = CCM \* [R,G,B] = 
$$\begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix} * [R,G,B]$$**

$$X = A0*R+B0*G+ C0*B$$

$$Y = A1*R+B1*G+ C1*B$$

$$Z = A2*R+B2*G+ C2*B$$

## 4. Calculate (x,y)

$$x = \frac{X}{X + Y + Z}$$
$$y = \frac{Y}{X + Y + Z}$$

## 5. Calculate CCT

- $CCT(x, y) = -449n^3 + 3525n^2 - 6823.3n + 5520.33$

- $n = (x - x_e) / (y - y_e)$

- $(x_e = 0.3320, y_e = 0.1858)$

- Note: This equation is the industry standard to calculate CCT

- Reference paper:

[https://www.usna.edu/Users/oceano/raylee/papers/RLee\\_A0\\_CCTpaper.pdf](https://www.usna.edu/Users/oceano/raylee/papers/RLee_A0_CCTpaper.pdf)

# CCT calculation sample code

```
static u32 cal_cct(struct isl29124_data_t *dat)
{
    //s32 tmp;
    s32 cct;
    s64 X0, Y0, Z0, sum0;
    s64 x,y,n, xe, ye;
    u8 Range;
    u8 bits;
    u16 als_r, als_g, als_b;
    s64 tmp;
```

## 1. Get R,G,B count

```
    als_r = dat->last_r;
    als_g = dat->last_g;
    als_b = dat->last_b;
```

```
    //bits = (dat->adc_resolution==0)? 1:0;
```

```
    bits = 0;
```

```
    Range=dat->als_range_using;
```

```
    if(Range == 0)
```

```
    {
```

```
        X0 = ( CCM_RangeLo[0][0]*als_r + CCM_RangeLo[0][1]*als_g + CCM_RangeLo[0][2] * als_b );
```

```
        Y0 = ( CCM_RangeLo[1][0]*als_r + CCM_RangeLo[1][1]*als_g + CCM_RangeLo[1][2] * als_b );
```

```
        Z0 = ( CCM_RangeLo[2][0]*als_r + CCM_RangeLo[2][1]*als_g + CCM_RangeLo[2][2] * als_b );
```

```
    }
```

```
    else
```

```
    {
```

```
        X0 = ( CCM_RangeHi[0][0]*als_r + CCM_RangeHi[0][1]*als_g + CCM_RangeHi[0][2] * als_b );
```

```
        Y0 = ( CCM_RangeHi[1][0]*als_r + CCM_RangeHi[1][1]*als_g + CCM_RangeHi[1][2] * als_b );
```

```
        Z0 = ( CCM_RangeHi[2][0]*als_r + CCM_RangeHi[2][1]*als_g + CCM_RangeHi[2][2] * als_b );
```

```
    }
```

## 2/3. Calculate X,Y,Z = CCM x [R,G,B]

If mode is low range → use CCM low range

If mode is high range → use CCM high range



# CCT calculation code continued

```

// X=X0/CCM_Gain[Range][bits];
// Y=Y0/CCM_Gain[Range][bits];
// Z=Z0/CCM_Gain[Range][bits];
// sum = X + Y + Z;
// x = X*1000/sum; y = Y*1000/sum;

sum0 = X0 + Y0 + Z0;
if (sum0 == 0)
{
    printk("sum0 value is 0");
    return -1;
}
//x = X0*1000*CCM_Gain[Range][bits]/sum0;
x = div64_s64(X0*10000, sum0);
//y = Y0*1000*CCM_Gain[Range][bits]/sum0;
y = div64_s64(Y0*10000, sum0);
xe=3320; // 0.3320
ye=1858; // 0.1858
if (y == 1858)
{
    printk("y-ye value is 0");
    return -1;
}
// n = (x-xe)/(y-ye)
//n = ( x - xe ) * 1000 / ( y - ye );
n = div64_s64(( x - xe ) * 10000, ( y - ye ));
//cct = n * ( n * ((-449 * n) / 1000 + 3525) / 1000 - 6823) / 1000 + 5520;
tmp = div64_s64(-449*n, 10000);
tmp = div64_s64((tmp+3525)*n, 10000);
tmp = div64_s64((tmp-6823)*n, 10000);
cct = tmp + 5520;
//n = (X<<31 - 712964572L *sum) / ( Y<<17 - 24354L * sum);
//cct = n * ( n * ((-449*n)/16384 + 3525)/16384 - 6823)/16384 + 5520;
dat->X = div64_s64( X0, CCM_Gain[Range][bits]);
dat->Y = div64_s64( Y0, CCM_Gain[Range][bits]);
dat->Z = div64_s64( Z0, CCM_Gain[Range][bits]);

//printk(KERN_ERR "CCM : X0 %lld, Y0:%lld, Z0:%lld, sum0:%lld, x:%lld, y:%lld, n:%lld, cct:%lld\n", X0, Y0, Z0, sum0, x, y, n,(long long int) cct );
if(cct < 0) cct = 0;

dat->cct = cct;
return cct;
}

```

4. Calculate x,y

xe and ye

5. Calculate CCT

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$

$$CCT(x, y) = -449n^3 + 3525n^2 - 6823.3n + 5520.33$$

$$n = (x - x_e)/(y - y_e)$$

$$(x_e = 0.3320, y_e = 0.1858)$$

# ISL29125 Driver including CCT code



CCT code  
example.



**intersil**<sup>TM</sup>  
A Renesas Company

[www.intersil.com](http://www.intersil.com)