

# OPD2x 及 OPD2xM 软件说明文档

版本 V5

## 版本信息

版本	时间	备注
V1	2014. 8. 30	
V2	2014. 10. 11	增加设置外部信源描述
V3	2015. 3. 20	增加 CAIC 功能
V4	2016. 9. 26	增加 GPIO 控制, 温度检测
V5	2016. 11. 1	增加梯形校正描述 删除复位寄存器设置

说明: OPD2x 是 .3 720P TRP 系列光机 (OPD20, OPD21, OPD22, ...), 不带 DPP343x 主控板的。

OPD2xM 是 .3 720P TRP 系列光机(OPD20, OPD21, OPD22, ...), 带 DPP343x 主控板的。

## 一. GPIO PIN 脚定义

此项只针对带主控板的 OPD2xM 系列光机, 只供参考, 具体请参考各型号的规格书

PIN	NAME	I	Control System on/off	1:Open System 0:Close System	
PIN37	PROJ_ON	-	Control System on/off	1:Open System 0:Close System	
PIN38	NC	-	NC	NC	
PIN39	NC	-	NC	NC	
PIN40	NC	-	NC	NC	
PIN41	NC	-	NC	NC	
PIN42	IIC_SDA	I	IIC_SDA	IIC	note5
PIN43	IIC_SCL	I	IIC_SCL	IIC	note5
PIN44	GND	GND	Ground	GND	
PIN45	NC	-	NC	NC	
PIN46	NC	-	NC	NC	
PIN47	TMEP_ERR	O	System Temp Status	ADC OUTPUT	note6
PIN48	LED_CTL	I	central LED on/off	1:Open Led lamp 0:Close Led lamp	note4
PIN49	NC	-	NC	NC	
PIN50	GND	GND	GROUND	GND	
PIN51	GND	GND	GROUND	GND	

PIN37 PROJ\_ON: 系统开关控制 高为开, 低为关

PIN42 IIC\_SDA, PIN43 IIC\_SCL: IIC 总线

PIN47 TMEP\_ERR: 温度检测脚, ADC 口, 对应的 10K 热敏电阻, 型号 NCP15XH103F03RC

PIN48 LED\_CTL: 背光控制, 高为开, 低为关

备注: PIN48 LED\_CTL 在 V5 版软件中已取消, 还请注意, 不用控制。

## 二. 开关机

此项开关机只针对带主控板的 **OPD2xM** 系列光机

在 OPD2xM 机型上，开关机是通过 51pin 管脚的 PIN37 PROJ\_ON 来控制的，该 GPIO 口设置为高时开机，开机后可以通过 I2C 命令来进行配置。把该 GPIO 口设置为低时系统会关机。在给光机供电时，PIN37 要先为低，保持大概 1 秒左右，再进行拉高开机动作。开机后 2 秒之内不要发送 I2C 命令，因为此时光机内部正在初始化。

推荐的开关机流程

开机： PROJ\_ON 拉高（系统开启）， 延迟 2s， 发送 IIC 命令（电流设置，显示设置等）。

关机： PROJ\_ON 拉低。

### 三. I2C 命令格式

前端 I2C 接口必须能够满足能读写不同长度字节数据的要求，比如说能读写单字节，也能读写 8 字节。具体格式如下：

操作	器件地址(8 位)(注①)	子地址 (8 位)	数据 (单或多字节)
写	0x36 (0x3A)	寄存器地址	寄存器值
读命令	0x36 (0x3A)	寄存器地址	寄存器参数 (注②)
读响应	0x37 (0x3B)	无	返回的寄存器值

注①： 由于 OPD20 机型的硬件器件地址是可配置的，所以有两个地址可选（写是 0x36 或 0x3A，对应的读为 0x37 或 0x3B），如果没有特殊说明，在 OPD20 的机型中，所使用的硬件地址为 0x36（读为 0x37）。

注②： 读取寄存器的值时，需要先进行写操作，把寄存器地址和相关参数(一般都不需要，只有特殊几个寄存器才需要参数)发送给光机；然后再进行读操作，读取光机返回的寄存器值。这个是一般 I2C 器件不一样的地方，需要注意。

我们在此定义两个函数，方便对下面的寄存器说明做一些示例。

```
//写寄存器， reg 表示寄存器地址 data 是发送的数据（数组）， data_size 是发送的数据
//长度
write_opd20_i2c(uint08 reg, uint08 *data, uint08 data_size)
```

```
//读寄存器， reg 表示寄存器地址， param 表示参数（数组）， param_size 表示参数长度，
//data 是读取的数据（数组）， data_size 是读取数据的长度
read_opd20_i2c(uint08 reg, uint08 *param, uint08 param_size, uint08 *data, uint08 data_size)
```

## 四. 寄存器说明

注意：OPD20 的寄存器是读和写分开的，同一个寄存器只能读或者只能写，不能又读又写。

### 4.1 输入信源选择（05h） 只写

数据： 1 字节 默认值： 01h

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2)---保留位

b(1:0)---输入信源

- 00: 外部 TTL 信源
- 01: 内部测试图片
- 02: 内部 LOGO
- 03: 保留

### 4.2 输入信源选择（06h） 只读

数据： 1 字节

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2)---保留位

b(1:0) ---输入信源

- 00: 外部信源
- 01: 内部测试图片
- 02: 内部 LOGO
- 03: 保留

#### 4.3 外部信源格式选择（07h） 只写

数据： 1 字节 默认值： 43h

Byte1: 见表 1

40h	Parallel	16	RGB565	16	1
41h	Parallel	18	RGB666	18	1
42h	Parallel	24	RGB888	8	3
43h	Parallel	24	RGB888	24	1
50h	Parallel	18	YCbCr666	18	1
51h	Parallel	24	YCbCr888	24	1
60h	Parallel	16	YCbCr422	8	2
61h	Parallel	16	YCbCr422	16	1
70h	CPU	16	RGB565	8	2
71h	CPU	16	RGB565	16	1
72h	CPU	18	RGB666	8	3
73h	CPU	18	RGB666	16	1.5
74h	CPU	18	RGB666	18	1
75h	CPU	24	RGB888	8	3
76h	CPU	24	RGB888	16	1.5
77h	CPU	24	RGB888	24	1
80h	CPU	18	YCbCr666	8	TBD
81h	CPU	18	YCbCr666	16	TBD
82h	CPU	18	YCbCr666	18	TBD
83h	CPU	24	YCbCr888	8	3
84h	CPU	24	YCbCr888	16	1.5
85h	CPU	24	YCbCr888	24	1
90h	CPU	16	YCbCr422	8	2
91h	CPU	16	YCbCr422	16	1
A0h	BT656	16	YCbCr422	8	2

表 1

#### 4.4 外部信源格式选择（08h） 只读

数据： 1 字节

Byte1: 见表 2

Value	Port	Bits/pixel	Data Type	Bus width	CLKs/pixel
21h	DSI	16	RGB565	24	
22h	DSI	18	RGB666	24	
23h	DSI	18	RGB666 (L)	24	
24h	DSI	24	RGB888	24	
30h	DSI	16	YCbCr422	16	

40h	Parallel	16	RGB565	16	1
41h	Parallel	18	RGB666	18	1
42h	Parallel	24	RGB888	8	3
43h	Parallel	24	RGB88	24	1
50h	Parallel	18	YCbCr666	18	1
51h	Parallel	24	YCbCr888	24	1
60h	Parallel	16	YCbCr422	8	2
61h	Parallel	16	YCbCr422	16	1
70h	CPU	16	RGB565	8	2
71h	CPU	16	RGB565	16	1
72h	CPU	18	RGB666	8	3
73h	CPU	18	RGB666	16	1.5
74h	CPU	18	RGB666	18	1
75h	CPU	24	RGB888	8	3
76h	CPU	24	RGB888	16	1.5
77h	CPU	24	RGB888	24	1
80h	CPU	18	YCbCr666	8	TBD
81h	CPU	18	YCbCr666	16	TBD
82h	CPU	18	YCbCr666	18	TBD
83h	CPU	24	YCbCr888	8	3
84h	CPU	24	YCbCr888	16	1.5
85h	CPU	24	YCbCr888	24	1
90h	CPU	16	YCbCr422	8	2
91h	CPU	16	YCbCr422	16	1
A0h	BT656	16	YCbCr422	8	2

表 2

#### 4.5 内部测试画面选择（0Bh） 只写

数据： 6 个字节 注③

Byte1: TPG 编号

Byte2: 前景/背景色

Byte3: 参数 1

Byte4: 参数 2

Byte5: 参数 3

Byte6: 参数 4

注③：针对不同类型内部图片，送的字节数是不一样的。

Byte1 默认值: 00h

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7)---图片边缘

0: 关

1: 开

b(6:4)---保留位

b(3:0)---图片类型

00h: 单色图（红绿蓝白等）

01h: 水平灰阶图

02h: 垂直灰阶图

03h: 水平线条

04h: 对角（斜）线条

05h: 垂直线条

06h: 方格

07h: Checkboard

08h: Color Bars

09h—0fh: 保留

Byte2 默认值: 70h

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7)---保留位

b(6:4)---前景色

00h: 黑

01h: 红

02h: 绿

03h: 蓝

04h: 青绿（绿+蓝）

05h: 品红（红+蓝）

06h: 黄（红+绿）

07h: 白

b(3)---保留位

b(2:0)---背景色

00h: 黑

01h: 红

02h: 绿

03h: 蓝

04h: 青绿（绿+蓝）

05h: 品红（红+蓝）

06h: 黄（红+绿）

07h: 白

针对产生不同类型图片的应用范围：

图片类型	前景色	背景色
单色图	有效	无效
水平灰阶图	有效	无效
垂直灰阶图	有效	无效
水平线条	有效	有效
对角线条	有效	有效
垂直线条	有效	有效
方格	有效	有效
CheckBoard	有效	有效
Color Bars	无效	无效

Byte3—Byte6

图片类型	Byte6	Byte5	Byte4	Byte3
单色图	n/a	n/a	n/a	n/a
水平灰阶图	n/a	n/a	结束值	开始值
垂直灰阶图	n/a	n/a	结束值	开始值
水平线条	n/a	n/a	背景线宽	前景线宽
对角线条	n/a	n/a	垂直间隔	水平间隔
垂直线条	n/a	n/a	背景线宽	前景线宽
方格	垂直背景线宽	垂直前景线宽	水平背景线宽	水平前景线宽
CheckBoard	垂直棋盘格数	垂直棋盘格数	水平棋盘格数	水平棋盘格数
Color Bars	n/a	n/a	n/a	n/a

针对不同图片类型，发送的字节数：

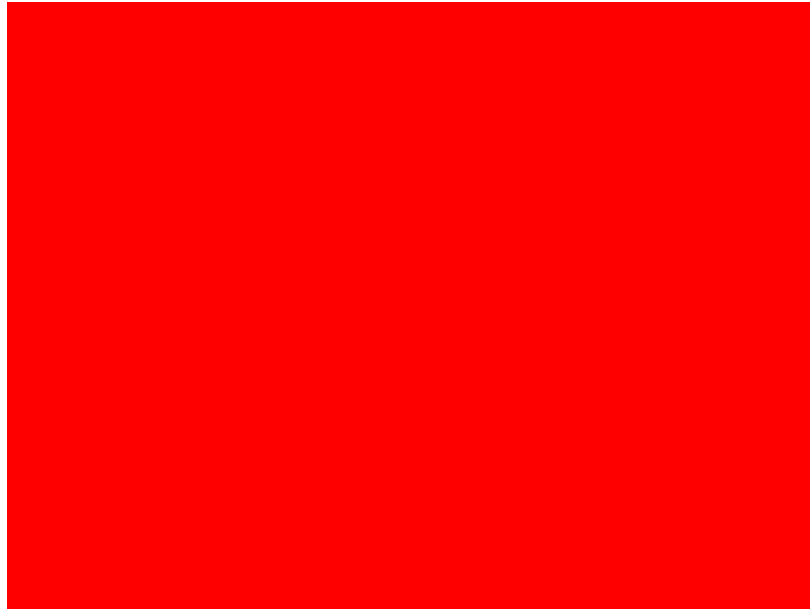
图片类型	字节数
单色图	2
水平灰阶图	4
垂直灰阶图	4
水平线条	4
对角线条	4
垂直线条	4
方格	6
CheckBoard	6
Color Bars	1



示例:

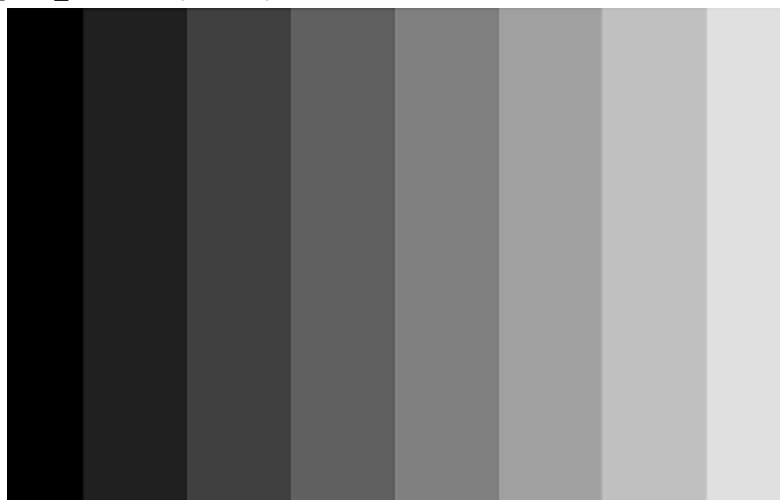
#### 4.5.1 单色图(红色)

```
data[0] = 0x00;    //单色图类型
data[1] = 0x10;    //前景色红, 背景色黑
Write_opd20_i2c(0x0b, data, 2);
```



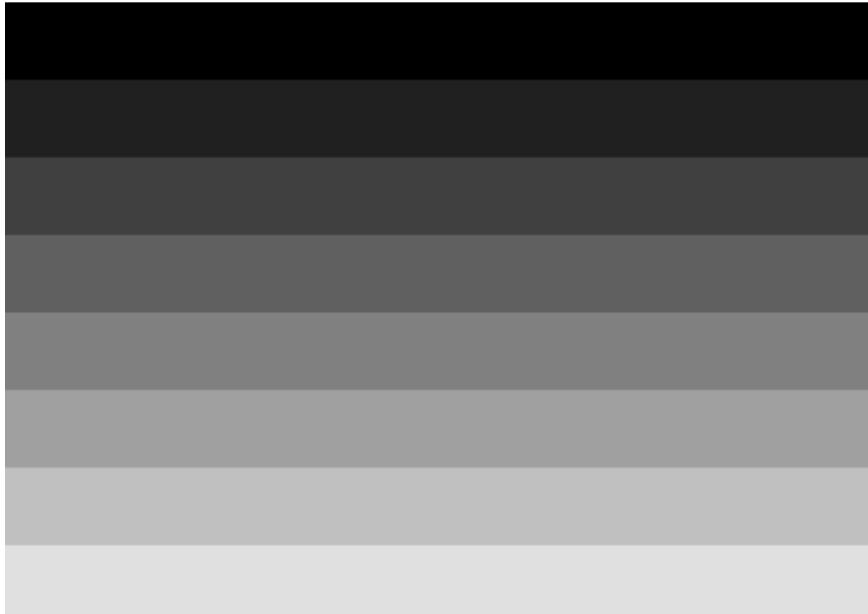
#### 4.5.2 水平灰阶

```
Data[0] = 0x01; // 水平灰阶
Data[1] = 0x70; //前景色白, 背景色黑
Data[2] = 0;    //0 阶
Data[3] = 0xff; //255 阶
write_opd20_i2c(0x0b, data, 4);
```



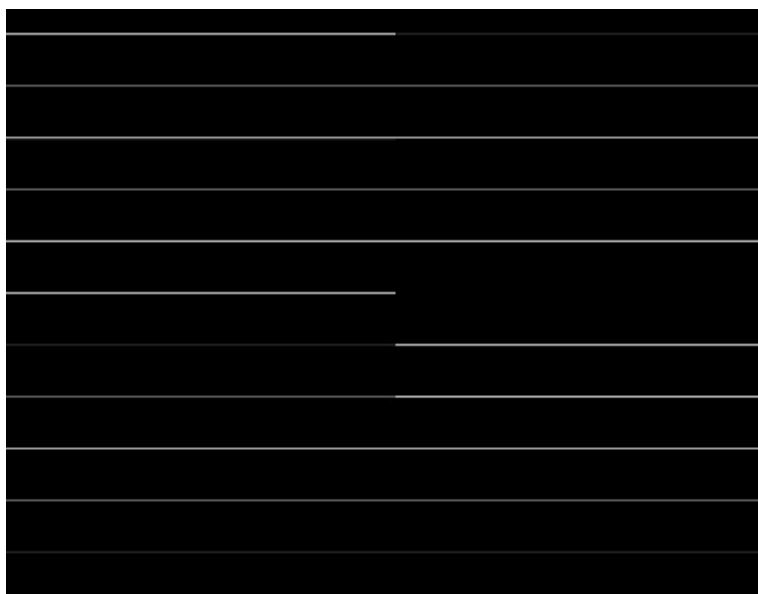
#### 4.5.3 垂直灰阶

```
Data[0] = 0x02; // 水平灰阶  
Data[1] = 0x70; //前景色白，背景色黑  
Data[2] = 0;    //0 阶  
Data[3] = 0xff; //255 阶  
write_opd20_i2c(0x0b, data, 4);
```



#### 4.5.4 水平线条

```
data[0] = 0x03; //水平线条  
data[1] = 0x70; //前景色白，背景色黑  
data[2] = 1;    //1 个像素的前景色  
data[3] = 9;    //9 个像素的背景色  
write_opd20_i2c(0x0b, data, 4);  
//每隔 9 个像素宽度的黑色水平线条就有 1 个像素的白色水平线条
```



4.6 内部测试图片 (0Ch) 只读

数据: 6 个字节

Byte1:

Byte2:

Byte3:

Byte4:

Byte5:

Byte6:

参见 0Bh 寄存器的值。

4.7 图像翻转 (14h) 只写

数据: 1 个字节

Byte1

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3)---保留位

b(2)---垂直翻转

0: 不翻转

1: 翻转

b(1)---水平翻转

0: 不翻转

1: 翻转

b(0)---旋转

0: 不旋转

1: 逆时针旋转 90 度



不旋转



逆时针旋转 90 度



不翻转



水平翻转



不翻转



垂直翻转

#### 4.8 图像翻转（15h） 只读

数据： 1 个字节

Byte1

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3)---保留位

b(2)---垂直翻转

0: 不翻转

1: 翻转

b(1)---水平翻转

0: 不翻转

1: 翻转

b(0)---旋转

0: 不旋转

1: 逆时针旋转 90 度

#### 4.9 显示幕布（16h） 只写

数据： 1 个字节 默认值： 00h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4)---保留位

b(3:1)---幕布颜色

00h: 黑

01h: 红

02h: 绿

03h: 蓝

04h: 青绿

05h: 品红

06h: 黄

07h: 白

b(0)---幕布开关

0: 关

1: 开

如果把幕布打开并且设定颜色的话，整个显示画面就是设定的幕布颜色。效果和内部图片的单色图是一样的。

#### 4.10 显示幕布（17h） 只读

数据： 1 个字节

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4)---保留位  
b(3:1)---幕布颜色  
00h: 黑  
01h: 红  
02h: 绿  
03h: 蓝  
04h: 青绿  
05h: 品红  
06h: 黄  
07h: 白  
b(0)---幕布开关  
0: 关  
1: 开

4.11 冻结画面（1Ah） 只写  
数据： 1 个字节 默认值： 00h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1)---保留位  
b(0)---画面冻结  
0: 关  
1: 开

4.12 冻结画面（1Bh） 只读  
数据： 1 个字节

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1)---保留位  
b(0)---画面冻结  
0: 关  
1: 开

4.13 LED 开关（52h） 只写  
数据： 1 个字节 默认值： 07h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3)---保留位  
b(2)---蓝色 LED  
0: 关  
1: 开

b(1)---绿色 LED

0: 关

1: 开

b(0)---红色 LED

0: 关

1: 开

#### 4.14 LED 开关 (53h) 只读

数据: 1 个字节

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3)---保留位

b(2)---蓝色 LED

0: 关

1: 开

b(1)---绿色 LED

0: 关

1: 开

b(0)---红色 LED

0: 关

1: 开

#### 4.15 LED 电流控制 (54h) 只写

数据: 6 个字节

Byte1: 红色 LED 的低字节

Byte2: 红色 LED 的高字节

Byte3: 绿色 LED 的低字节

Byte4: 绿色 LED 的高字节

Byte5: 蓝色 LED 的低字节

Byte6: 蓝色 LED 的高字节

调整电流时相应的光机亮度会改变，但是同时也会影响到白平衡（白场的色温），所以更改电流时还请注意。

#### 4.16 LED 电流值 (55h) 只读

数据: 6 个字节

Byte1: 红色 LED 的低字节

Byte2: 红色 LED 的高字节

Byte3: 绿色 LED 的低字节

Byte4: 绿色 LED 的高字节

Byte5: 蓝色 LED 的低字节

Byte6: 蓝色 LED 的高字节

#### 4.17 设置图像修剪 (Write Image Crop 10h) 只写

数据: 8 个字节

Byte1: 截取初始像素点坐标 x 的低字节

Byte2: 截取初始像素点坐标 x 的高字节

Byte3: 截取初始像素点坐标 y 的低字节

Byte4: 截取初始像素点坐标 y 的高字节

Byte5: 每行像素点的低字节

Byte6: 每行像素点的高字节

Byte7: 行数的低字节

Byte8: 行数的高字节

#### 4.18 读取图像修剪 (Read Image Crop 11h) 只读

数据: 8 个字节

Byte1: 截取初始像素点坐标 x 的低字节

Byte2: 截取初始像素点坐标 x 的高字节

Byte3: 截取初始像素点坐标 y 的低字节

Byte4: 截取初始像素点坐标 y 的高字节

Byte5: 每行像素点的低字节

Byte6: 每行像素点的高字节

Byte7: 行数的低字节

Byte8: 行数的高字节

#### 4.19 设置显示区域 (Write Display Size 12h) 只写

数据: 8 个字节 默认值: 720P (DMD 分辨率)

Byte1: 初始像素点坐标 x 的低字节

Byte2: 初始像素点坐标 x 的高字节

Byte3: 初始像素点坐标 y 的低字节

Byte4: 初始像素点坐标 y 的高字节

Byte5: 每行像素点的低字节

Byte6: 每行像素点的高字节

Byte7: 行数的低字节

Byte8: 行数的高字节

#### 4.20 读取显示区域 (Read Display Size 13h) 只读

数据: 8 个字节

Byte1: 初始像素点坐标 x 的低字节

Byte2: 初始像素点坐标 x 的高字节

Byte3: 初始像素点坐标 y 的低字节

Byte4: 初始像素点坐标 y 的高字节

Byte5: 每行像素点的低字节

Byte6: 每行像素点的高字节

Byte7: 行数的低字节

Byte8: 行数的高字节

4.21 设置外部输入信源分辨率 (Write External Input Image Size 2Eh) 只写

数据: 4 个字节 默认: 854×480

Byte1: 每行像素点的低字节

Byte2: 每行像素点的高字节

Byte3: 行数的低字节

Byte4: 行数的高字节

4.22 读取外部输入信源分辨率 (Write External Input Image Size 2Fh) 只读

数据: 4 个字节 默认: 854×480

Byte1: 每行像素点的低字节

Byte2: 每行像素点的高字节

Byte3: 行数的低字节

Byte4: 行数的高字节



设置外部信源需要用到以下几个寄存器（都是只写寄存器）：

输入信源选择（05h），外部信源选择（07h），显示屏幕（16h），冻结画面（1Ah），设置图像修剪（10h），设置显示区域（12h），设置外部输入信源分辨率（2Eh）。

参考示例（以外部信源 720P 为例）：

```
data[0] = 0x01;
write_opd20_i2c(0x16, data, 1);    //黑屏
data[0] = 0x01;
write_opd20_i2c(0x1A, data, 1);    //冻结画面

data[0] = 0;
data[1] = 0;
data[2] = 0;
data[3] = 0;
data[4] = 0x00;
data[5] = 0x05;
data[6] = 0xD0;
data[7] = 0x02;
write_opd20_i2c(0x10, data, 8);    //设置画面修剪（720P，相当于没有修剪）

data[0] = 0;
data[1] = 0;
data[2] = 0;
data[3] = 0;
data[4] = 0x00;
data[5] = 0x05;
data[6] = 0xD0;
data[7] = 0x02;
write_opd20_i2c(0x12, data, 8);    //设置显示区域（720P）

data[0] = 0x00;
data[1] = 0x05;
data[2] = 0xD0;
data[3] = 0x02;
write_opd20_i2c(0x2E, data, 4);    //设置外部输入信源分辨率（720P）

data[0] = 0x43;
write_opd20_i2c(0x07, data, 1);    //设置信源格式为 RGB888
data[0] = 0x00;
write_opd20_i2c(0x05, data, 1);    //把信源设置为外部信源
data[0] = 0x00;
write_opd20_i2c(0x1A, data, 1);    //解冻画面
data[0] = 0x00;
write_opd20_i2c(0x16, data, 1);    //显示屏幕
```

4.23 CAIC（动态调整电流）功能（Write LED Output Control Method 50h）只写

数据： 1 个字节 默认值： 01h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2)---保留位

b(1:0)---LED 控制方法

- 00: 手动写电流（CAIC 功能关闭）
- 01: 动态调整电流（CAIC 功能开启）
- 10: 保留位
- 11: 保留位

4.24 CAIC（动态调整电流）功能（Write LED Output Control Method 51h）只读

数据： 1 个字节 默认值： 01h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2)---保留位

b(1:0)---LED 控制方法

- 00: 手动写电流（CAIC 功能关闭）
- 01: 动态调整电流（CAIC 功能开启）
- 10: 保留位
- 11: 保留位

4.25 选择 look（write look select 22h） 只写

数据： 1 个字节 默认值 00h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0): look 的定位（第几条）。

备注： 光机默认调好白平衡参数的 look 是第 0 条，还请不要随意修改。

3D look 固定在第 8 条，所以在切换 3D 前，请把 look 切换到 3D look。

#### 4.26 读取当前 look (write look select 23h) 只读

数据: 1 个字节 默认值 00h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0): look 的定位 (第几条)。

#### 4.27 设置 3D (write 3-D control 20h) 只写

数据: 1 个字节 默认值 00h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7)---保留位

b(6)---3D 参考极性

0: 校正---不需要反向

1: 不校正---需要反向

b(5)---Frame Dominance

0: Left Dom.

1: Right Dom.

b(4:2)---保留位

b(1)---3D 参考源

0: 内部

1: 外部

b(0)---保留位

一般这个寄存器设置 0x22 或者 0x02 即可

#### 4.28 读取 3D 配置 (read 3-D control 21h) 只读

数据: 1 个字节 默认值 00h

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

#### 4.29 设置梯形校正功能 (write Keystone Correction control 88h) 只写

数据: 5 个字节

Byte1: 功能开关

Byte2: 光学投射比的低字节

Byte3: 光学投射比的高字节

Byte4: 光学偏移 (offset) 低字节

Byte5: 光学偏移 (offset) 高字节

Byte1:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1)---保留位

b(0)---功能开关

- 0: 关梯形校正
- 1: 开梯形校正

Byte2: 光学投射比的低字节

Byte3: 光学投射比的高字节

在 OPD2x (OPD20/OPD21/OPD22) 的投射比是 1.66, 所以 Byte2=A9, Byte3=1

Byte4: 光学偏移 (offset) 低字节

Byte5: 光学偏移 (offset) 高字节

在 OPD2x (OPD20/OPD21/OPD22) 的 offset 是 100%, 所以 Byte4=0, Byte5=1

4.30 读取梯形校正功能配置 (write Keystone Correction control 89h) 只读数据: 5 个字节

- Byte1: 功能开关
- Byte2: 光学投射比的低字节
- Byte3: 光学投射比的高字节
- Byte4: 光学偏移 (offset) 低字节
- Byte5: 光学偏移 (offset) 高字节

4.31 设置梯形校正角度 (write Keystone Projection Pitch Angle BBh) 只写数据: 2 个字节 默认值: 0000h

- Byte1: 梯形校正角度的低字节
- Byte2: 梯形校正角度的高字节

寄存器值是实际角度×256, 最高位为正负

b15 -2'	b14 2 <sup>0</sup>	b13 2 <sup>1</sup>	b12 2 <sup>2</sup>	b11 2 <sup>3</sup>	b10 2 <sup>4</sup>	b9 2 <sup>5</sup>	b8 2 <sup>6</sup>	b7 2 <sup>7</sup>	b6 2 <sup>8</sup>	b5 2 <sup>9</sup>	b4 2 <sup>10</sup>	b3 2 <sup>11</sup>	b2 2 <sup>12</sup>	b1 2 <sup>13</sup>	b0 2 <sup>14</sup>
------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

备注:

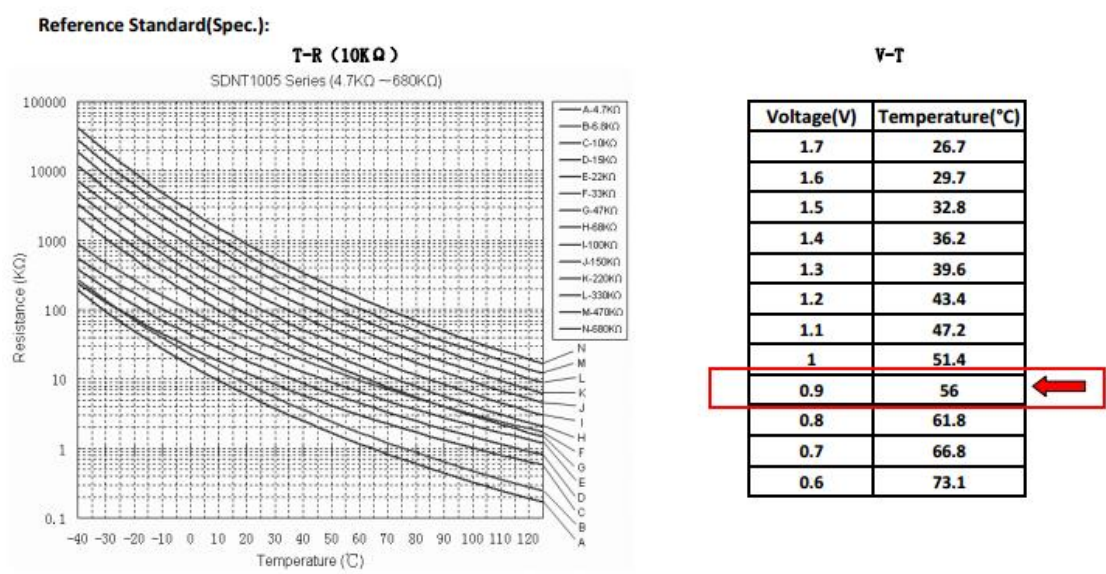
- 1. 梯形校正角度范围从-40 度到 40 度;
- 2. 这个寄存器必须和设置梯形校正功能寄存器 (88h) 配合使用, 也就是必须开梯形校正功能后才能设置角度。

4.32 读取梯形校正角度 (write Keystone Projection Pitch Angle BCh) 只读数据: 2 个字节 默认值: 0000h

- Byte1: 梯形校正角度的低字节
- Byte2: 梯形校正角度的高字节

五. 温度检测

OPD2xM 和 OPD2x 系列都在 LED 灯上装有热敏电阻，以便于监控温度。



如果是光机本身带主控板，那只要读取 51pin 接口中的 PIN47 TMEP\_ERR 的 ADC 值即可。  
如果光机本身不带主控板，那就需要前端板接一个 10K 电阻上拉 3.3V。