



版本: 1.0.0 2021 年 8 月

AnyCloud37E MPU 屏调试手册

版本变更说明

以下表格对于本文档的版本变更做一个简要的说明。版本变更仅限于技术内容的变更，不包括版式、格式、句法等的变更。

版本	说明	完成日期
V1.0.0	首次发布	2021 年 8 月

Anyka Confidential For
CIMC Use Only

目 录

1 概述	4
2 读者对象	4
3 硬件说明	4
4 添加 MPU LCD	6
4.1 配置 MPU LCD 的管脚	6
4.2 LCD 控制器输入参数的配置	10
4.3 配置驱动 IC 初始化序列及 MPU 时序参数	12
4.3.1 如何获取 MPU 时序参数	12
4.3.2 如何配置	13
5 MPU 屏调试	17
5.1 硬件确认	17
5.2 参数配置	17
5.3 驱动 IC 刷图	18

1 概述

本文档介绍如何在 AK376xE 芯片解决方案上开发调试 MPU LCD 屏，以帮助客户有序快速开发 MPU LCD 业务。

2 读者对象

本文档主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

3 硬件说明

1、接口资源

AK376xE 芯片的 MPU 显示屏接口如表 3-1 所示。

表 3-1 AK376xE 的 MPU 显示屏管脚

序号	管脚名称	电源域
1	GPIO34_PU/RGB_VOAGATE/MPU_RD_N	VDDIO
2	GPIO35_PU/RGB_VOVSNC/MPU_A0	VDDIO
3	GPIO36_PU/RGB_VOHSNC/MPU_CS_N	VDDIO
4	GPIO37_PU/RGB_VOPLCK/MPU_WR_N	VDDIO
5	GPIO38_PU/RGB_D0/MPU_D0	VDDIO
6	GPIO39_PU/RGB_D1/MPU_D1	VDDIO
7	GPIO40_PU/RGB_D2/MPU_D2	VDDIO
8	GPIO41_PU/RGB_D3/MPU_D3	VDDIO
9	GPIO42_PU/RGB_D4/MPU_D4	VDDIO
10	GPIO43_PU/RGB_D5/MPU_D5	VDDIO
11	GPIO44_PU/RGB_D6/MPU_D6	VDDIO
12	GPIO45_PU/RGB_D7/MPU_D7	VDDIO
13	GPIO46_PU/RGB_D8/MPU_D8/I2C2_SCL	VDDIO
14	GPIO47_PU/RGB_D9/MPU_D9/I2C2_SDA	VDDIO
15	GPIO48_PU/RGB_D10/MPU_D10/JTAG_RSTN	VDDIO
16	GPIO49_PU/RGB_D11/MPU_D11/JTAG_TDI	VDDIO

序号	管脚名称	电源域
17	GPIO50_PU/RGB_D12/MPU_D12/JTAG_TMS/I2S1_LRCLK	VDDIO
18	GPIO51_PU/RGB_D13/MPU_D13/JTAG_TCLK/I2S1_BCLK	VDDIO_OSC
19	GPIO52_PU/RGB_D14/MPU_D14/JTAG_RTCK/I2S1_MCLK	VDDIO_OSC
20	GPIO53_PU/RGB_D15/MPU_D15/JTAG_TDO/I2S1_DIN	VDDIO_OSC
21	GPIO54_PU/RGB_D16/MPU_D16/PDM_CLK/I2C3_SCL	VDDIO_OSC
22	GPIO55_PU/RGB_D17/MPU_D17/PDM_DATA/I2C3_SDA	VDDIO_OSC

注意：

- (1) 高位和低位的电源域不同，电压注意要统一。
- (2) MPU 仅支持 8080 接口模式，不支持 6800 接口模式。
- (3) 屏基本都是小屏应用，支持 8bit、9bit、16bit 和 18bit 四种数据位宽。
- (4) 在硬件设计时按低位对齐方式处理，如图 3-1 所示。

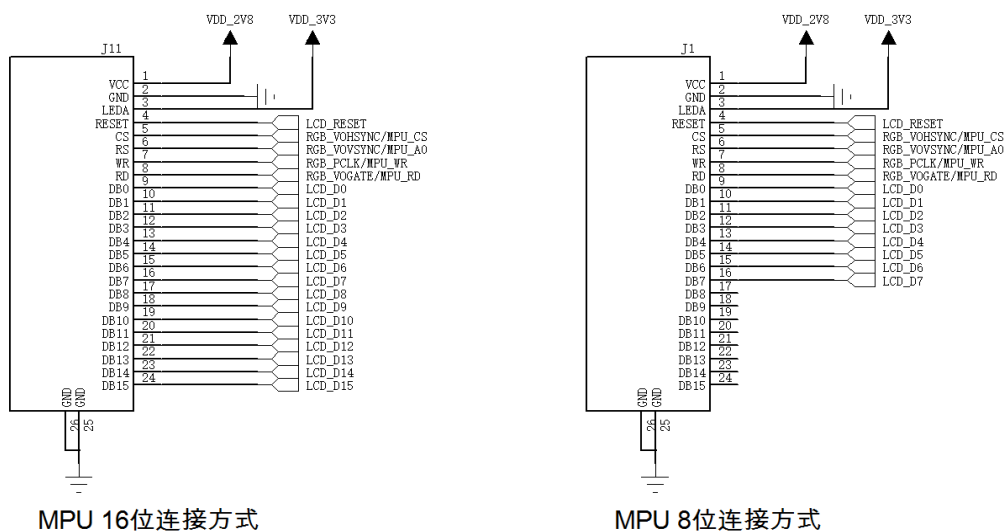


图 3-1 MPU 连接方式

4 添加 MPU LCD

4.1 配置 MPU LCD 的管脚

查看原理图，确认 MPU LCD 硬件连接后，需在 dts 中配置 MPU LCD，dts 文件存放在 arch/arm/boot/dts/EVB_CBDR_AK3760E_V1.0.1_mpu.dts。

默认配置如下：

```
&gpio {
    .....
    lcd_mpu_8pins: lcd_mpu_8pins {
        anyka,pins = < XGPIO_045
            XGPIO_044
            XGPIO_043
            XGPIO_042
            XGPIO_041
            XGPIO_040
            XGPIO_039
            XGPIO_038
            XGPIO_037
            XGPIO_036
            XGPIO_035
            XGPIO_034>;
        anyka,function = < XGPIO_045_FUNC_MPU_D7
            XGPIO_044_FUNC_MPU_D6
            XGPIO_043_FUNC_MPU_D5
            XGPIO_042_FUNC_MPU_D4
            XGPIO_041_FUNC_MPU_D3
            XGPIO_040_FUNC_MPU_D2
            XGPIO_039_FUNC_MPU_D1
            XGPIO_038_FUNC_MPU_D0
            XGPIO_037_FUNC_MPU_WR
            XGPIO_036_FUNC_MPU_CS
            XGPIO_035_FUNC_MPU_A0
            XGPIO_034_FUNC_MPU_RD
            >;
    };
};
```

```

anyka,pull = < 0x00010210

    0x00010210
    0x00010210
    0x00010210
    0x00010210
    0x00010210
    0x00010210
    0x00010210
    0x00010210
    0x00010110
    0x00010010
    0x00010010
    0x00010010>;

};

lcd_mpu_8pins_sleep: lcd_mpu_8pins_sleep {
    anyka,pins = < XGPIO_045
        XGPIO_044
        XGPIO_043
        XGPIO_042
        XGPIO_041
        XGPIO_040
        XGPIO_039
        XGPIO_038
        XGPIO_037
        XGPIO_036
        XGPIO_035
        XGPIO_034>;

    anyka,function = < XGPIO_045_FUNC_GPIO45
        XGPIO_044_FUNC_GPIO44
        XGPIO_043_FUNC_GPIO43
        XGPIO_042_FUNC_GPIO42
        XGPIO_041_FUNC_GPIO41
        XGPIO_040_FUNC_GPIO40
        XGPIO_039_FUNC_GPIO39
        XGPIO_038_FUNC_GPIO38
        XGPIO_037_FUNC_GPIO37
        XGPIO_036_FUNC_GPIO36

```

```

        XGPIO_035_FUNC_GPIO35
        XGPIO_034_FUNC_GPIO34

        >;

        anyka,pull = <0x00000000>;
    };

    lcd_reset_pins: lcd_reset_pins {
        anyka,pins = <XGPIO_026>;
        anyka,function = <XGPIO_026_FUNC_GPIO26>;
        anyka,pull = <0x01000110>;
    };

    lcd_power_pins: lcd_power_pins {
        anyka,pins = <XGPIO_073>;
        anyka,function = <XGPIO_073_FUNC_GPIO73>;
        anyka,pull = <0x00000010>;
    };

    .....
};

```

lcd_mpu_8pins、lcd_reset_pins 和 lcd_power_pins 是配置 MPU LCD 正常情况下的数据和制管脚。lcd_mpu_8pins_sleep 是配置 MPU LCD 休眠时的数据管脚和控制管脚。

正常情况下的 lcd_mpu_8pins 配置如下所示：

anyka,pins	anyka,function	anyka,pull
XGPIO_045	XGPIO_045_FUNC_MPU_D7	0x00010210
XGPIO_044	XGPIO_044_FUNC_MPU_D6	0x00010210
XGPIO_043	XGPIO_043_FUNC_MPU_D5	0x00010210
XGPIO_042	XGPIO_042_FUNC_MPU_D4	0x00010210
XGPIO_041	XGPIO_041_FUNC_MPU_D3	0x00010210
XGPIO_040	XGPIO_040_FUNC_MPU_D2	0x00010210
XGPIO_039	XGPIO_039_FUNC_MPU_D1	0x00010210
XGPIO_038	XGPIO_038_FUNC_MPU_D0	0x00010210
XGPIO_037	XGPIO_037_FUNC_MPU_WR	0x00010110
XGPIO_036	XGPIO_036_FUNC_MPU_CS	0x00010010

anyka,pins	anyka,function	anyka,pull
XGPIO_035	XGPIO_035_FUNC_MPU_A0	0x00010010
XGPIO_034	XGPIO_034_FUNC_MPU_RD	0x00010010

lcd_mpu_8pins 中 anyka,pins 和 anyka,function 用于配置 MPU LCD 管脚对应的 GPIO 及 GPIO 的工作模式，请参考[错误!未找到引用源。](#)进行配置。如果是非 8 位接口，请根据[错误!未找到引用源。](#)进行修改。

anyka,pull 用于配置管脚的上下拉、驱动能力等，配置方法请参考《AnyCloud37E 平台内核板级配置手册》。

注意，anyka,pull 中驱动能力等值的配置可能需要根据实际硬件情况进行调整，以免出现过冲或者上升/下降沿时间过长等不符合 MPU LCD 时序的问题。

休眠时 lcd_mpu_8pins_sleep 的配置如下所示：

anyka,pins	anyka,function	anyka,pull
XGPIO_045	XGPIO_045_FUNC_GPIO45	0x00000000
XGPIO_044	XGPIO_044_FUNC_GPIO44	0x00000000
XGPIO_043	XGPIO_043_FUNC_GPIO43	0x00000000
XGPIO_042	XGPIO_042_FUNC_GPIO42	0x00000000
XGPIO_041	XGPIO_041_FUNC_GPIO41	0x00000000
XGPIO_040	XGPIO_040_FUNC_GPIO40	0x00000000
XGPIO_039	XGPIO_039_FUNC_GPIO39	0x00000000
XGPIO_038	XGPIO_038_FUNC_GPIO38	0x00000000
XGPIO_037	XGPIO_037_FUNC_GPIO37	0x00000000
XGPIO_036	XGPIO_036_FUNC_GPIO36	0x00000000
XGPIO_035	XGPIO_035_FUNC_GPIO35	0x00000000
XGPIO_034	XGPIO_034_FUNC_GPIO34	0x00000000

lcd_mpu_8pins_sleep 用于配置 MPU LCD 休眠时的管脚状态。注意，所有管脚的 anyka,function 需要配置为 GPIO，anyka,pull 需要配置为 0，避免休眠时漏电，产生不必要的功耗。

lcd_reset_pins 和 lcd_power_pins 用于配置 MPU LCD 的 reset 引脚和电源引脚。这两个引脚的 anyka,function 需要配置为 GPIO。anyka,pull 的值可能需要根据实际硬件情况进行调整，以免出现不符合 MPU LCD 时序的问题。

4.2 LCD 控制器输入参数的配置

LCD 控制器的输入参数的配置也 arch/arm/boot/dts/EVB_CBDR_AK3760E_V1.0.1_mpu.dts 中：

```
/* RGB panel */
&lcdc {
    pinctrl-names = "default","sleep";
    pinctrl-0 = <&lcd_mpu_8pins &lcd_reset_pins &lcd_power_pins>;
    pinctrl-1 = <&lcd_mpu_8pins_sleep &lcd_reset_pins &lcd_power_pins>;
    reset-pins = <&gpio 26 1>;
    pwr-gpio = <&gpio 73 1>;
    backlight-gpio = <&gpio 27 1>;

    lcd-logo-width = <320>;
    lcd-logo-height = <240>;

    // [fmt1:fmt0] 00 16bits input (RGB565 or BGR565)
    // [fmt1:fmt0] 01 24bits input (RGB888 or BGR888)
    // [fmt1:fmt0] 10 or 11 32bits input (ARGB888 ABGR888 RGBA888 BGRA888)
    lcd-logo-fmt0 = <0>;
    lcd-logo-fmt1 = <0>;
    lcd-logo-rgb-seq = <1>; // 0 for BGR, 1 for RGB

    lcd-fb-type = <1>; // 0:single buffer; 1:double buffer
    status = "okay";
};
```

参数含义说明如表 4-1 所示：

表 4-1 LCD 控制器参数

节点	说明
reset-pins	LCD 屏硬件复位控制 GPIO，需要和 lcd_reset_pins

节点	说明
	中的配置一致
pwr-gpio	LCD 屏上电控制 GPIO，需要和 lcd_power_pins 中的配置一致
backlight-gpio	LCD 屏背光控制 GPIO
lcd_logo_width	LOGO 的宽度
lcd_logo_height	LOGO 的高度
lcd_logo_fmt0 lcd_logo_fmt1	LOGO 的格式 [fmt1: fmt0]: [0:0] 16bits input (RGB565 or BGR565) [0:1] 24bits input (RGB888 or BGR888) [1:0] or [1:1] 32bits input (ARGB888 or ABGR888 or RGBA888 or BGRA888) 注意，屏的输入格式和此处的设置是相同的，一旦设置好后，应用程序是不能通过修改输入格式的
lcd_logo_rgb_seq	LOGO 的 RGB 数据排列顺序 0: RGB 1: BGR 注意，屏的 RGB 数据排列顺序和此处的设置是相同的。
lcd_fb_type	Lcd 帧缓冲数量： 0: 单缓冲 1: 双缓冲 缓冲数量应满足配置的 LCD 分辨率所需要的内存大小要小于配置的 CMA_RESERVED_MEMORY_SIZE，若分辨率较大，设置双缓冲时所需内存可能会超过设置的 CMA_RESERVED_MEMORY_SIZE，导致加载驱动时，申请内存失败。 37E 平台 CMA_RESERVED_MEMORY_SIZE 配置是在/kernel/arch/arm/boot/dts/anycld_ak37e.dtsi 文件里，默认是 8M；
status	okay 表示打开，disable 表示关闭

关于 GPIO 口的配置有以下几点需要注意：

Uboot 下，背光控制 GPIO 引脚是由 CONFIG_LCD_BACKLIGHT_GPIO 宏定义的，该宏定义在 37E 平台是在/uboot/include/configs/anycld_ak37e.h 头文件里定义的。

4.3 配置驱动 IC 初始化序列及 MPU 时序参数

4.3.1 如何获取 MPU 时序参数

MPU 时序参数及初始化序列请咨询 IC 原厂 FAE 支持或自行查看 IC DATASHEET 确认。举例如下：

如图 2-1、图 2-2 所示，为 ST7789V MPU 屏的驱动 IC datasheet 中的相关参数说明，根据图中信息可知，该屏分辨率为 320*240，支持 262K 颜色，MPU 接口支持 8bit、9bit、16bit、18bit，一个写操作需要的时间最小为 66ns，一个写操作低电平需要的时间最小为 15ns。

Display Resolution: 240*RGB (H) *320(V)

Frame Memory Size: 240 x 320 x 18-bit = 1,382,400 bits

LCD Driver Output Circuits

- Source Outputs: 240 RGB Channels
- Gate Outputs: 320 Channels
- Common Electrode Output

Display Colors (Color Mode)

- Full Color: 262K, RGB=(666) max., Idle Mode Off
- Color Reduce: 8-color, RGB=(111), Idle Mode On

Programmable Pixel Color Format (Color Depth) for Various Display Data input Format

- 12-bit/pixel: RGB=(444)
- 16-bit/pixel: RGB=(565)
- 18-bit/pixel: RGB=(666)

MCU Interface

- Parallel 8080-series MCU Interface (8-bit, 9-bit, 16-bit & 18-bit)
- 6/16/18 RGB Interface(VSYNC, HSYNC, DOTCLK, ENABLE, DB[17:0])
- Serial Peripheral Interface(SPI Interface)
- VSYNC Interface

Signal	Symbol	Parameter	Min	Max	Unit	Description
D/CX	T _{AST}	Address setup time	0		ns	-
	T _{AHT}	Address hold time (Write/Read)	10		ns	
CSX	T _{CHW}	Chip select "H" pulse width	0		ns	-
	T _{CS}	Chip select setup time (Write)	15		ns	
	T _{RCS}	Chip select setup time (Read ID)	45		ns	
	T _{RCSFM}	Chip select setup time (Read FM)	355		ns	
	T _{CSF}	Chip select wait time (Write/Read)	10		ns	
	T _{CSH}	Chip select hold time	10		ns	
WRX	T _{WC}	Write cycle	66		ns	
	T _{WRH}	Control pulse "H" duration	15		ns	
	T _{WRL}	Control pulse "L" duration	15		ns	
RDX (ID)	T _{RC}	Read cycle (ID)	160		ns	When read ID data
	T _{RDH}	Control pulse "H" duration (ID)	90		ns	
	T _{RDL}	Control pulse "L" duration (ID)	45		ns	
RDX (FM)	T _{RCFM}	Read cycle (FM)	450		ns	When read from frame memory
	T _{RDHFM}	Control pulse "H" duration (FM)	90		ns	
	T _{RDLFM}	Control pulse "L" duration (FM)	355		ns	
D[17:0]	T _{DST}	Data setup time	10		ns	For CL=30pF

4.3.2 如何配置

LCD 控制器的输出参数 37E 平台在/kernel/arch/arm/boot/dts/anycld_lcd.dtsi 中配置。

4.3.2.1 LCD 输出端控制参数

```
st7789v: lcd-panel@11 {
    compatible = "mpu,st7789v";
    panel-if = <2>;
    panel-width = <240>;
    panel-height = <320>;
    panel-pclk-div = <2>;
    panel-data-width = <0>;
    panel-color-mode = <1>;
    panel-rs-polarity = <1>;
    panel-wr-cycle = <20>;
    panel-wr-low = <3>;
```

```

panel-direction = <0>;
panel-init-list = <0x11 0xFFFF
                    0x36 0x00 0xffff
                    0x3a 0x55 0xFFFF
                    0xc5 0x1a 0xffff
                    0xb1 0x00 0x00 0x00 0xFFFF
                    0xb2 0x0c 0x0c 0x00 0x33 0x33 0xFFFF
                    0xb7 0x35 0xFFFF
                    0xbb 0x35 0xFFFF
                    0xc0 0x2c 0xffff
                    0xC2 0x01 0xFFFF
                    0xC3 0x0b 0xFFFF
                    0xC4 0x20 0xFFFF
                    0xC6 0x0f 0xFFFF
                    0xd0 0xa4 0xa1 0xffff
                    0xe0 0xd0 0x00 0x02 0x07
                    0x0b 0x1a 0x31 0x54 0x40
                    0x29 0x12 0x12 0x12 0x17 0xFFFF
                    0xe1 0xd0 0x00 0x02 0x07
                    0x05 0x25 0x2d 0x44 0x45
                    0x1c 0x18 0x16 0x1c 0x1d 0xFFFF
                    // 0x29 0xffff
                    //0x11 0xFFFF
                    0x2a 0x00 0x00 0x00 0xef 0xFFFF
                    0x2b 0x00 0x00 0x01 0x3f 0xFFFF
                    0x2c 0xFFFF
                    0x29 0xFFFF>;

status = "disable";
};

```

参数含义说明如表 4-2 所示：

表 4-2 节点说明表

节点	说明
panel-if	外接 LCD 屏接口类型。2 表示 MPU。
panel-width	LCD 屏分辨率，代表宽。
panel-height	LCD 屏分辨率，代表高。
panel-pclk-div	<p>设置 pclk-div 分频系数。</p> <p>注意： 分频系数须为整数。</p> <p>在 37E 平台上芯片内部 PCLK 由 asic_pll 分频得到。</p> <p>计算公式为： $\text{pclk-div} = \text{asic_pll} / \text{PCLK} - 1;$ 其中 asic_pll = 500MHz。</p> <p>MPU 屏中设置 PCLK 是为了匹配写操作所需要的周期时长，所以要结合 panel-wr-cycle 决定设置值。</p>
panel-data-width	<p>总线宽度，由驱动 IC 原厂提供或自行查看驱动 IC datasheet 确认。</p> <p>0: 8bits 1: 16bits 2: 18bits 3: 9bits</p>
panel-color-mode	<p>LCD 屏支持的颜色数量，由驱动 IC 原厂提供或自行查看驱动 IC datasheet 确认。</p> <p>0: 4K 1: 64K/256K</p>
panel-rs-polarity	<p>MPU_A0 的极性。由驱动 IC 原厂提供或自行查看驱动 IC datasheet 确认。</p> <p>0: 低表示数据高表示命令 1: 低表示命令高表示数据</p>
panel-wr-cycle	<p>写一个点需要的周期，由驱动 IC 原厂提供或自行查看驱动 IC datasheet 确认。</p> <p>注意： 该参数单位为 PCLK 周期数量，而 IC 原厂提供的参数单位是周期时长，所以此处需要根据设置的 PCLK 进行换算。</p> <p>换算公式为： $\text{周期时长} = (1 / \text{PCLK}) * \text{周期数量}。$ </p>
panel-wr-low	<p>一次写操作低电平需要的周期，由驱动 IC 原厂提供或自行查看驱动 IC datasheet 确认。</p> <p>该参数单位为 PCLK 周期数量，而 IC 原厂提供的</p>

节点	说明
	<p>参数单位是周期时长，所以此处需要根据设置的 PCLK 进行换算。</p> <p>换算公式为： 周期时长=(1/PCLK)*周期数量。</p> <p>注意： wr-cycle - wr-low >= 2 (18/16-bit mode) wr-cycle/2 - wr-low >= 2 (8/9-bit mode)</p>
panel-direction	<p>数据刷新方向。</p> <p>0: 先从左到右，再从上往下 1: 先从上到下，再从左到右</p>
panel-init-list	IC 初始化设置参数，详细说明查看 2.3.2.2 节说明。
status	<p>该节点参数的使能状态。</p> <p>okay: 开启 disable: 关闭</p> <p>该参数在 dtsi 文件中建议配置为 disable, 由对应的板级配置的 dts 文件中配置为 okay 状态进行覆盖，使能方法如下：</p> <pre>&FRD350C25024 { status = "okay"; };</pre> <p>若同时打开多份节点配置，会导致上述的其他参数在驱动中被多次设置，导致时序异常，所以必须保证在一份配置中只能开启一个配置。</p>

注意：以上表格中未列出的参数，使用默认值即可。

4.3.2.2 驱动 IC 初始化参数

```
panel-init-list = <0x11 0xFFFF
    0x36 0x00 0xffff
    0x3a 0x55 0xFFFF
    0xc5 0x1a 0xffff
    0xb1 0x00 0x00 0x00 0xFFFF
    0xb2 0x0c 0x0c 0x00 0x33 0x33 0xFFFF
    0xb7 0x35 0xFFFF
    0xbb 0x35 0xFFFF
    0xc0 0x2c 0xffff
    0xc2 0x01 0xFFFF
    0xc3 0x0b 0xFFFF
```



```
0xC4 0x20 0xFFFF
0xC6 0x0f 0xFFFF
0xd0 0xa4 0xa1 0xffff
0xe0 0xd0 0x00 0x02 0x07
0x0b 0x1a 0x31 0x54 0x40
0x29 0x12 0x12 0x12 0x17 0xFFFF
0xe1 0xd0 0x00 0x02 0x07
0x05 0x25 0x2d 0x44 0x45
0x1c 0x18 0x16 0x1c 0x1d 0xFFFF
// 0x29 0xffff
//0x11 0xFFFF
0x2a 0x00 0x00 0x00 0xef 0xFFFF
0x2b 0x00 0x00 0x01 0x3f 0xFFFF
0x2c 0xFFFF
0x29 0xFFFF>;
```

每行第一个为 Byte1，后面的参数以此类推，参数说明如下：

Byte1：代表需要发送的命令。当设置为 0x11 时，表示延时控制，会延时 120ms。

Byte2~ByteN：代表需要发送的数据。根据长短包的大小，每行长短不一。

每一行参数末尾 0xFFFF：代表每一行的结束标识符。

5 MPU 屏调试

5.1 硬件确认

首先需确认提供给屏的 IOVCC、VCI、RESET 电压符合 IC 要求，具体要求需查看 IC 规格书，在此基础上才可进行后面的软件调试，若该部分电压不符合要求，会导致无法显示图像或者图像异常等问题。

5.2 参数配置

用户只需将 IC 原厂提供的参数填写到 DTS 文件中对应的参数节点即可。驱动加载后，会解析 DTS 中的参数，配置 LCD 控制器。

5.3 驱动 IC 刷图

完成参数配置后，编译固件烧录设备，通过加载驱动显示 LOGO 以及运行 vo_sample、vdec_sample 的方式查看显示是否正常。

若出现出图有时成功有时失败的问题，先排查写操作周期设置是否符合 IC 的时序要求，若不符合，可通过修改 panel-wr-cycle、panel-wr-low、panel-plck-div 参数进行调整，若调整后问题还存在，可尝试提高 MPU 相关管脚驱动能力。

若出现颜色不对的问题，可排查 IC 初始化命令中配置的 RGB 顺序是否正确。

5.4 调屏示例

假设要调一款 JPY23106 的 MPU 屏，该屏的驱动 IC 是 ILI9342C，分辨率是 320*240，使用 8 位 8080-I 的接口，图像格式是 RGB565，首先准备好一张 RGB565 的图片，图片在 MPU 屏上正常显示效果如图 5-1：

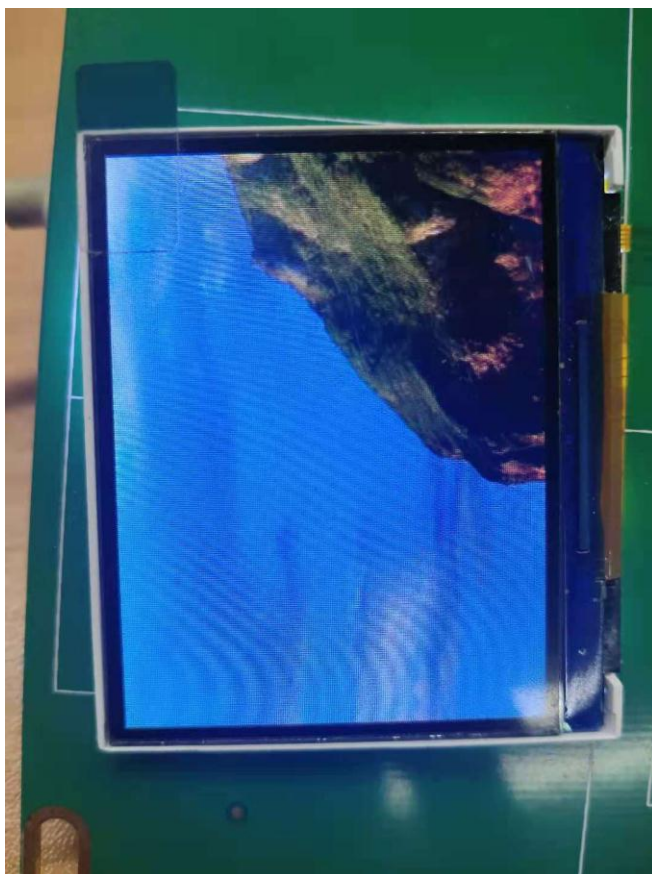


图 5-1

如上所述，调屏首先要对硬件进行检查和确认，查看屏的 datasheet 和找到 IC 原厂提供初始化序列，根据第 4 章的说明对 MPU 的管脚、LCD 控制器和 MPU 的时序及初始化序列进行配置，调试中如果出现图片看不清楚的问题，如[图 5-2](#)所示：

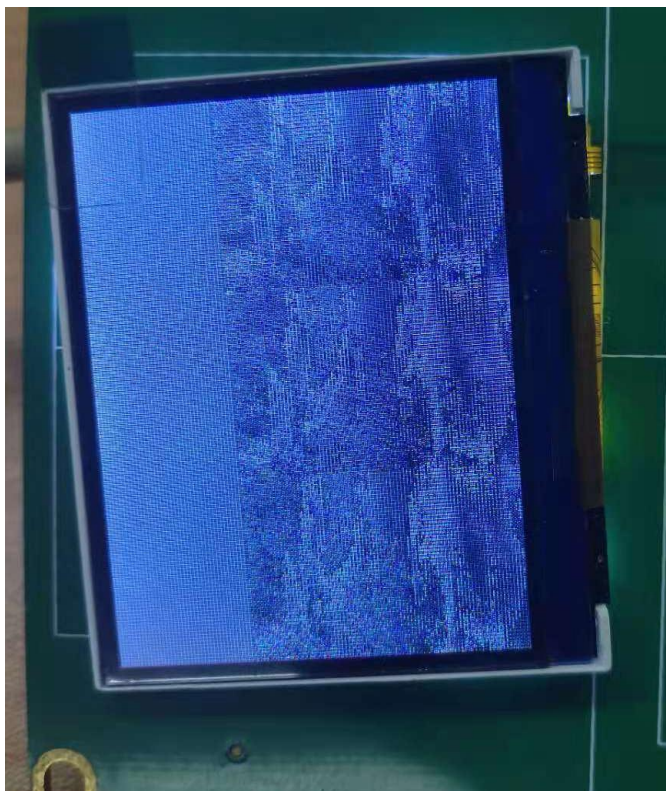


图 5-2

这通常是 MPU 屏的时序参数有问题，请使用示波器和屏的 datasheet，并参考第 4 章调整 panel-pclk-div、panel-wr-low 和 panel-wr-cycle 等时序参数。

如果出现如图 5-3 的颜色不对的情况，可检查初始化序列中 RGB 顺序的设置。

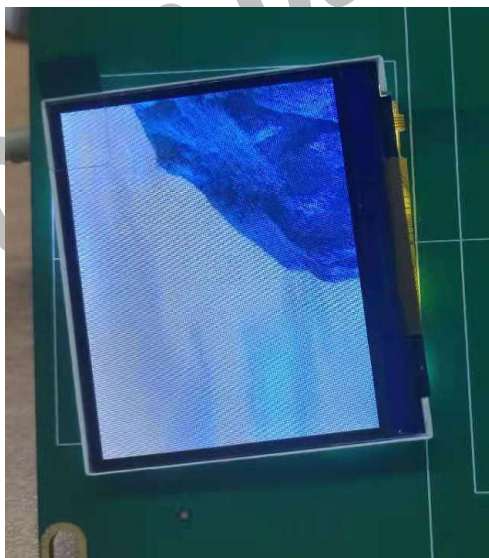


图 5-3