Application Note

ATK-MD0130 模块使用说明

高性能 1.3’LCD 显示模块

使用说明

正点原子

广州市星翼电子科技有限公司

修订历史

版本	日期	原因
V1.0	2022/06/25	第一次发布
V1.1	2023/03/11	添加对阿波罗 STM32F429 开发板的阿波罗 STM32F767 开发板的支持
V1.2	2023/04/15	添加对阿波罗 STM32H743 开发板的支持

目 录

1，硬件连接

1.1 正点原子 MiniSTM32F103 开发板

ATK-MD0130 模块可直接与正点原子 MiniSTM32F103 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
MiniSTM32F103 开发板	GND	3.3V	PA4	PC4	PA5	PA7	PA6	PA1

表 1.1.1 ATK-MD0130 模块与 MiniSTM32F103 开发板连接关系

1.2 正点原子精英 STM32F103 开发板

ATK-MD0130 模块可直接与正点原子精英 STM32F103 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
精英 STM32F103 开发板	GND	3.3V	PG8	PG7	PB13	PB15	PB14	PG6

表 1.2.1 ATK-MD0130 模块与精英 STM32F103 开发板连接关系

1.3 正点原子战舰 STM32F103 开发板

ATK-MD0130 模块可直接与正点原子战舰 STM32F103 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
战舰 STM32F103 开发板	GND	3.3V	PG8	PG7	PB13	PB15	PB14	PG6

表 1.3.1 ATK-MD0130 模块与战舰 STM32F103 开发板连接关系

1.4 正点原子探索者 STM32F407 开发板

ATK-MD0130 模块可直接与正点原子探索者 STM32F407 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
探索者 STM32F407 开发板	GND	3.3V	PG6	PG7	PB3	PB5	PB4	PG8

表 1.4.1 ATK-MD0130 模块与探索者 STM32F407 开发板连接关系

1.5 正点原子 MiniSTM32H750 开发板

ATK-MD0130 模块可直接与正点原子 MiniSTM32H750 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
MiniSTM32H750 开发板	GND	3.3V	PC0	PE0	PB13	PB15	PB14	PC3

表 1.5.1 ATK-MD0130 模块与 MiniSTM32H750 开发板连接关系

1.6 正点原子阿波罗 STM32F429 开发板

ATK-MD0130 模块可直接与正点原子阿波罗 STM32F429 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
阿波罗 STM32F429 开发板	GND	3.3V	PG12	PG10	PB13	PB15	PB14	PI11

表 1.6.1 ATK-MD0130 模块与阿波罗 STM32F429 开发板连接关系

1.7 正点原子阿波罗 STM32F767 开发板

ATK-MD0130 模块可直接与正点原子阿波罗 STM32F767 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
阿波罗 STM32F767 开发板	GND	3.3V	PG12	PG10	PB13	PB15	PB14	PI11

表 1.7.1 ATK-MD0130 模块与阿波罗 STM32F767 开发板连接关系

1.8 正点原子阿波罗 STM32H743 开发板

ATK-MD0130 模块可直接与正点原子阿波罗 STM32H743 开发板板载的 WIRELESS 模块接口（SPI 接口）进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系							
ATK-MD0130 模块	GND	3V3	PWR	CS	SCK	SDA	WR	RST
阿波罗 STM32H743 开发板	GND	3.3V	PG12	PG10	PB13	PB15	PB14	PI11

表 1.8.1 ATK-MD0130 模块与阿波罗 STM32H743 开发板连接关系

2，实验功能

2.1 ATK-MD0130 模块测试实验

2.1.1 功能说明

在本实验中，开发板主控芯片通过 SPI 接口与 ATK-MD0130 模块进行通讯，从而完成对 ATK-MD0130 的初始化配置以及操作 ATK-MD0130 模块的 LCD 显示各种内容。

2.1.2 源码解读

打开本实验的工程文件夹，能够在./Drivers/BSP 目录下看到 ATK_MD0130 子文件夹，该文件夹中就包含了 ATK-MD0130 模块的驱动文件，如下图所示：

```
./Drivers/BSP/ATK_MD0130/
|-- atk_md0130.c
|-- atk_md0130.h
|-- atk_md0130_font.h
|-- atk_md0130_spi.c
`-- atk_md0130_spi.h
```

图 2.1.2.1 ATK-MD0130 模块驱动代码

2.1.2.1 ATK-MD0130 模块接口驱动

在图 2.1.2.1 中，atk_md0130_spi.c 和 atk_md0130_spi.h 是开发板与 ATK-MD0130 模块通讯而使用的 SPI 驱动文件，关于 SPI 的驱动介绍，请查看正点原子各个开发板对应的开发指南中 SPI 对应的章节。

2.1.2.2 ATK-MD0130 模块字体文件

在图 2.1.2.1 中，atk_md0130_font.h 是驱动 ATK-MD0130 模块在 LCD 上显示 ASCII 字符时需要的字体取模文件，该文件支持字号为 12、16、24 和 32 的 ASCII 字符。

2.1.2.3 ATK-MD0130 模块驱动

在图 2.1.2.1 中，atk_md0130.c 和 atk_md0130.h 是 ATK-MD0130 模块的驱动文件，包含了 ATK-MD0130 模块初始化、LCD 清屏、LCD 画点、LCD 画线、LCD 显示字符、LCD 显示字符串、LCD 显示数字等相关的 ATK-MD0130 模块操作 API 函数。函数比较多，下面仅介绍几个重要的 API 函数。

1. 函数 atk_md0130_init()

该函数用于初始化 ATK-MD0130 模块，具体的代码，如下所示：

```
/**
 * @brief   ATK-MD0130 模块初始化
 * @param   无
 * @retval  无
 */
void atk_md0130_init(void)
{
    /* ATK-MD0130 模块硬件初始化 */
    atk_md0130_hw_init();
```

```
    /* ATK-MD0130 模块硬件复位 */
    atk_md0130_hw_reset();
    /* ATK-MD0130 模块 SPI 接口初始化 */
    atk_md0130_spi_init();
    /* ATK-MD0130 模块寄存器初始化 */
    atk_md0130_reg_init();
    /* 设置 ATK-MD0130 模块地址 */
    atk_md0130_set_address( 0,
                            0,
                            ATK_MD0130_LCD_WIDTH - 1,
                            ATK_MD0130_LCD_HEIGHT - 1);
    /* 开启 ATK-MD0130 模块 LCD 背光 */
    atk_md0130_display_on();
    /* ATK-MD0130 模块 LCD 清屏 */
    atk_md0130_clear(ATK_MD0130_WHITE);
}
```

从上面的代码中可以看出，函数 atk_md0130_init()初始化 ATK-MD0130 模块主要就是初始化与 ATK-MD030 模块的 SPI 通讯接口，SPI 通讯接口初始化完成后就可以通过 SPI 通讯接口初始化 ATK-MD0130 模块的寄存器，以完成 ATK-MD0130 模块的初始化。

2. 函数 atk_md0130_draw_point()

该函数用于在 ATK-MD0130 模块的 LCD 上画一个点，理论上只要通过该函数就能够完成对 ATK-MD0130 模块 LCD 的所有显示操作，该函数的具体代码，如下所示：

```
/**
 * @brief   ATK-MD0130 模块 LCD 画点
 * @param   x       : 待画点的 X 坐标
 *          y       : 待画点的 Y 坐标
 *          color   : 待画点的颜色
 * @retval  无
 */
void atk_md0130_draw_point(uint16_t x, uint16_t y, uint16_t color)
{
    atk_md0130_set_address(x, y, x, y);
    atk_md0130_write_dat_16b(color);
}
```

从上面的代码中可以看出，在 ATK-MD0130 模块的 LCD 上画点需要两个步骤，首先就是确认待画点的位置，接着就是确认待画点的颜色，因为在 ATK-MD0130 模块初始化配置其寄存器的时，配置 ATK-MD0130 模块 LCD 显示的颜色格式为 RGB565，因此一个点的颜色数据就需要占用两个字节。

3. 函数 atk_md0130_fill()

该函数用于对 ATK-MD0130 模块 LCD 的某一区域填充指定的单一颜色，虽然画点函数 atk_md0130_draw_point()能够完成 ATK-MD0130 模块 LCD 显示的所有操作，但是对于大面积填充的场景，每画一个点都要确定点的位置和颜色，这导致用画点函数在这种场景下的效率不高。因为 ATK-MD0130 模块支持先确定一个填充区域，然后自动将连续的颜色数据顺序填充进确定好的区域，因此就有了在大面积填充的场景下效率更高的方法，函数

atk_md0130_fill()的具体代码，如下所示：

```
/**
 * @brief   ATK-MD0130 模块 LCD 区域填充
 * @param   xs      : 区域起始 X 坐标
 *          ys      : 区域起始 Y 坐标
 *          xe      : 区域终止 X 坐标
 *          ye      : 区域终止 Y 坐标
 *          color   : 区域填充颜色
 * @retval  无
 */
void atk_md0130_fill(   uint16_t xs,
                        uint16_t ys,
                        uint16_t xe,
                        uint16_t ye,
                        uint16_t color)
{
    uint32_t area_size;
    uint32_t area_remain = 0;
    uint16_t buf_index;

    area_size = (xe - xs + 1) * (ye - ys + 1) * sizeof(uint16_t);
    if (area_size > ATK_MD0130_LCD_BUF_SIZE)
    {
        area_remain = area_size - ATK_MD0130_LCD_BUF_SIZE;
        area_size = ATK_MD0130_LCD_BUF_SIZE;
    }

    atk_md0130_set_address(xs, ys, xe, ye);
    ATK_MD0130_WR(1);
    while (1)
    {
        for (buf_index=0; buf_index<area_size / sizeof(uint16_t); buf_index++)
        {
            g_atk_md0130_lcd_buf[buf_index * sizeof(uint16_t)] =
                                                (uint8_t)(color >> 8) & 0xFF;
            g_atk_md0130_lcd_buf[buf_index * sizeof(uint16_t) + 1] =
                                                (uint8_t)color & 0xFF;
        }

        atk_md0130_spi_send(g_atk_md0130_lcd_buf, area_size);

        if (area_remain == 0)
        {
            break;
```

```
        }

        if (area_remain > ATK_MD0130_LCD_BUF_SIZE)
        {
            area_remain = area_remain - ATK_MD0130_LCD_BUF_SIZE;
        }
        else
        {
            area_size = area_remain;
            area_remain = 0;
        }
    }
}
```

从上面的代码中可以看出，函数 atk_md0130_fill()在一次往 ATK-MD0130 模块 LCD 填充颜色的过程中只会通过函数 atk_md0130_set_address()确定一次填充的区域范围，然后将颜色数据一次性连续地发送至 ATK-MD0130 模块即可，这样一来，就大大地提高了大面积填充颜色的效率。

2.1.2.4 实验测试代码

实验的测试代码为文件 demo.c，在工程目录下的 User 子目录中。测试代码的入口函数为 demo_run()，具体的代码，如下所示：

```
/**
 * @brief   例程演示入口函数
 * @param   无
 * @retval  无
 */
void demo_run(void)
{
    /* 初始化 ATK-MD0130 模块 */
    atk_md0130_init();
    /* ATK-MD0130 模块 LCD 清屏 */
    atk_md0130_clear(ATK_MD0130_WHITE);
    /* ATK-MD0096 模块 OLED 显示字符串 */
    atk_md0130_show_string( 10,
                            10,
                            "STM32",
                            ATK_MD0130_LCD_FONT_32,
                            TK_MD0130_RED);
    atk_md0130_show_string( 10,
                            42,
                            "ATK-MD0130",
                            ATK_MD0130_LCD_FONT_24,
                            ATK_MD0130_RED);
    atk_md0130_show_string( 10,
                            66,
```

```
                            "ATOM@ALIENTEK",
                            ATK_MD0130_LCD_FONT_16,
                            ATK_MD0130_RED);

    while (1)
    {
        /* 演示立方体 3D 旋转 */
        demo_show_cube();
    }
}
```

从上面的代码中可以看出，整个测试代码的逻辑还是比较简单的，就是在 ATK-MD0130 模块的 LCD 上显示了一些实验信息，然后就调用函数 demo_show_cube()显示立方体 3D 旋转的演示，函数 demo_show_cube()实际上就是通过 LCD 画线函数在 ATK-MD0130 模块的 LCD 显示屏上画线段，画线的本质实际上也就是画点。

2.1.3 实验现象

将 ATK-MD0130 模块按照第一节“硬件连接”中介绍的连接方式与开发板连接，并将实验代码编译烧录至开发板中，如果此时开发板连接 LCD，那么 LCD 显示的内容，如下图所示：

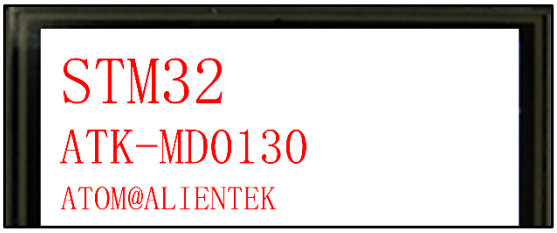

图 2.1.3.1 LCD 显示内容一

同时，通过串口调试助手输出实验信息，如下图所示：

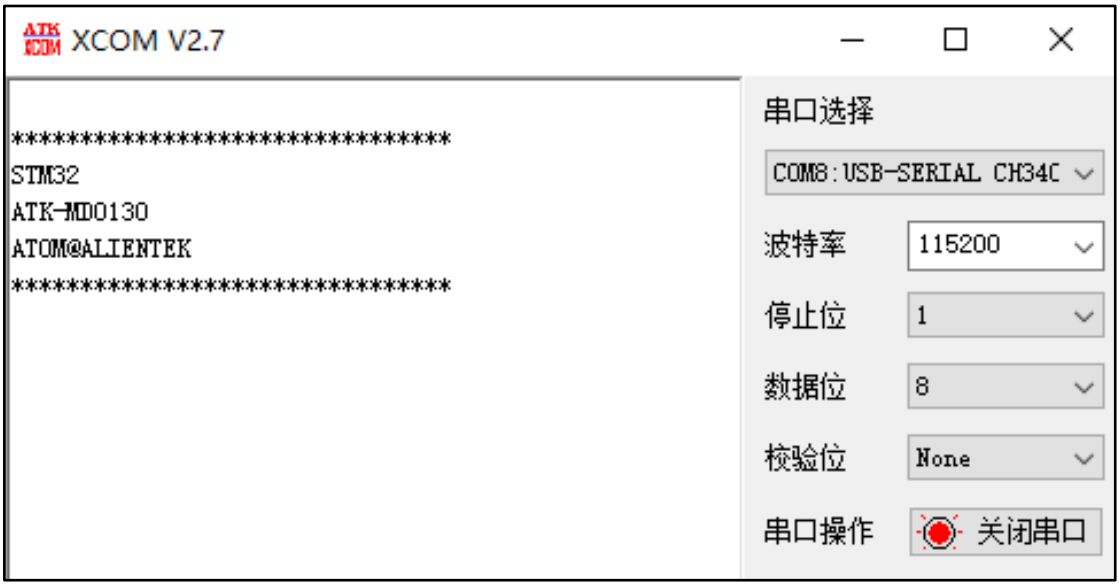

图 2.1.3.2 串口调试助手显示内容一

接下来，如果 ATK-MD0130 模块初始化成功，则会在 ATK-MD0130 模块的 LCD 上显示一些实验信息，和立方体 3D 旋转的演示，如下图所示：

图 2.1.3.3 ATK-MD0130 模块显示内容

3，其他

1、 购买地址：

天猫：https://zhengdianyuanzi.tmall.com

淘宝：https://openedv.taobao.com

2、 资料下载

模块资料下载地址：http://www.openedv.com/docs/modules/lcd/1.3-lcd.html

3、 技术支持

公司网址：www.alientek.com

技术论坛：http://www.openedv.com/forum.php

在线教学：www.yuanzige.com

B 站视频：https://space.bilibili.com/394620890

传真：020-36773971

电话：020-38271790