Application Note

ATK-MD0280 模块使用说明

高性能 2.8’TFTLCD 电阻触摸屏模块

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	2023/02/06	新增 2.8 寸 LCD 模块型号 7789（Chip ID: 0x8552）
V1.2	2023/03/11	添加对阿波罗 STM32F429 开发板的阿波罗 STM32F767 开发板的支持
V1.3	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-MD0280 模块可直接与正点原子 MiniSTM32F103 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
MiniSTM32F103 开发板	PC9	PC8	PC7	PC6	RESET	PB0	PB1	PB2	PB3
模块对应开发板	连接关系								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
MiniSTM32F103 开发板	PB4	PB5	PB6	PB7	PB8	PB9	PB10	PB11	PB12
模块对应开发板	连接关系								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
MiniSTM32F103 开发板	PB13	PB14	PB15	GND	PC10	3.3V	3.3V	GND	GND
模块对应开发板	连接关系								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
MiniSTM32F103 开发板	5V	PC2	PC3	PC1	-	PC13	PC0	-	-

表 1.1.1 ATK-MD0280 模块与 MiniSTM32F103 开发板连接关系

1.2 正点原子精英 STM32F103 开发板

ATK-MD0280 模块可直接与正点原子精英 STM32F103 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
精英 STM32F103 开发板	PG12	PG0	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	连接关系								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
精英 STM32F103 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	连接关系								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
精英 STM32F103 开发板	PD8	PD9	PD10	GND	PB0	3.3V	3.3V	GND	GND
模块对应开发板	连接关系								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
精英 STM32F103 开发板	5V	PB2	PF9	PF10	-	PF11	PB1	-	-

表 1.2.1 ATK-MD0280 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-MD0280 模块可直接与正点原子战舰 STM32F103 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
战舰 STM32F103 开发板	PG12	PG0	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
战舰 STM32F103 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
战舰 STM32F103 开发板	PD8	PD9	PD10	GND	PB0	3.3V	3.3V	GND	GND
模块对应开发板	**连接关系**								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
战舰 STM32F103 开发板	5V	PB2	PF9	PF10	-	PF11	PB1	-	-

表 1.3.1 ATK-MD0280 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-MD0280 模块可直接与正点原子探索者 STM32F407 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
探索者 STM32F407 开发板	PG12	PF12	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
探索者 STM32F407 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
探索者 STM32F407 开发板	PD8	PD9	PD10	GND	PB15	3.3V	3.3V	GND	GND
模块对应开发板	**连接关系**								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
探索者 STM32F407 开发板	5V	PB2	PF11	PB1	-	PC13	PB0	-	-

表 1.4.1 ATK-MD0280 模块与探索者 STM32F407 开发板连接关系

1.5 正点原子 F407 电机控制开发板

ATK-MD0280 模块可直接与正点原子 F407 电机控制开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系

ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
F407 电机控制开发板	PG12	PG0	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
F407 电机控制开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
F407 电机控制开发板	PD8	PD9	PD10	GND	PH9	3.3V	3.3V	GND	GND
模块对应开发板	**连接关系**								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
F407 电机控制开发板	5V	PD11	PH8	PH7	-	PG1	PH6	-	-

表 1.5.1 ATK-MD0280 模块与 F407 电机控制开发板连接关系

1.6 正点原子 MiniSTM32H750 开发板

ATK-MD0280 模块可直接与正点原子 MiniSTM32H750 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	**连接关系**								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
MiniSTM32H750 开发板	PD7	PE3	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
MiniSTM32H750 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
MiniSTM32H750 开发板	PD8	PD9	PD10	GND	PB5	3.3V	3.3V	GND	GND
模块对应开发板	**连接关系**								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
MiniSTM32H750 开发板	5V	PD6	PB3	PB1	-	PC5	PB0	-	-

表 1.6.1 ATK-MD0280 模块与 MiniSTM32H750 开发板连接关系

1.7 正点原子阿波罗 STM32F429 开发板

ATK-MD0280 模块可直接与正点原子阿波罗 STM32F429 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	**连接关系**								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
阿波罗 STM32F429 开发板	PD7	PD13	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	**连接关系**								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
阿波罗 STM32F429 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15

模块对应开发板	连接关系								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
阿波罗 STM32F429 开发板	PD8	PD9	PD10	GND	PB5	3.3V	3.3V	GND	GND
模块对应开发板	连接关系								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
阿波罗 STM32F429 开发板	5V	PG3	PI3	PH7	-	PI8	PH6	-	-

表 1.7.1 ATK-MD0280 模块与阿波罗 STM32F429 开发板连接关系

1.8 正点原子阿波罗 STM32F767 开发板

ATK-MD0280 模块可直接与正点原子阿波罗 STM32F767 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
阿波罗 STM32F767 开发板	PD7	PD13	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	连接关系								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
阿波罗 STM32F767 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	连接关系								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
阿波罗 STM32F767 开发板	PD8	PD9	PD10	GND	PB5	3.3V	3.3V	GND	GND
模块对应开发板	连接关系								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
阿波罗 STM32F767 开发板	5V	PG3	PI3	PH7	-	PI8	PH6	-	-

表 1.8.1 ATK-MD0280 模块与阿波罗 STM32F767 开发板连接关系

1.9 正点原子阿波罗 STM32H743 开发板

ATK-MD0280 模块可直接与正点原子阿波罗 STM32H743 开发板板载的 TFTLCD 模块接口进行连接，具体的连接关系，如下表所示：

模块对应开发板	连接关系								
ATK-MD0280 模块	CS	RS	WR	RD	RST	D0	D1	D2	D3
阿波罗 STM32H743 开发板	PD7	PD13	PD5	PD4	RESET	PD14	PD15	PD0	PD1
模块对应开发板	连接关系								
ATK-MD0280 模块	D4	D5	D6	D7	D8	D9	D10	D11	D12
阿波罗 STM32H743 开发板	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15
模块对应开发板	连接关系								
ATK-MD0280 模块	D13	D14	D15	GND	BL	VDD	VDD	GND	GND
阿波罗 STM32H743 开发板	PD8	PD9	PD10	GND	PB5	3.3V	3.3V	GND	GND
模块对应开发板	连接关系								
ATK-MD0280 模块	V5	MI	MO	PEN	NC	TCS	CLK	-	-
阿波罗 STM32H743 开发板	5V	PG3	PI3	PH7	-	PI8	PH6	-	-

表 1.9.1 ATK-MD0280 模块与阿波罗 STM32H743 开发板连接关系

2，实验功能

2.1 ATK-MD0280 模块测试实验（FMC&FSMC）

2.1.1 功能说明

在本实验中，开发板主控芯片通过 FMC 或 FSMC 接口与 ATK-MD0280 模块进行通讯，从而完成对 ATK-MD0280 模块的初始化配置以及操作 ATK-MD0280 模块的 LCD 显示各种内容，同时通过模拟 SPI 接口与 ATK-MD0280 模块进行通讯，从而获取 ATK-MD0280 模块的触摸数据。

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_MD0280 子文件夹，该文件夹中就包含了 ATK-MD0280 模块的驱动文件，如下图所示（以 FSMC 为例）：

```
./Drivers/BSP/ATK_MD0280/
|-- atk_md0280.c
|-- atk_md0280.h
|-- atk_md0280_font.h
|-- atk_md0280_fsmc.c
|-- atk_md0280_fsmc.h
|-- atk_md0280_touch.c
|-- atk_md0280_touch.h
|-- atk_md0280_touch_spi.c
`-- atk_md0280_touch_spi.h
```

图 2.1.2.1 ATK-MD0280 模块驱动代码

2.1.2.1 ATK-MD0280 模块接口驱动

在图 2.1.2.1 中，atk_md0280_fsmc.c 和 atk_md0280_fsmc.h 是开发板与 ATK-MD0280 模块通讯而使用的 FSMC 驱动文件，关于 FSMC 和 FMC 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 FSMC 和 FMC 对应的章节。

2.1.2.2 ATK-MD0280 模块字体文件

在图 2.1.2.1 中，atk_md0280_font.h 是驱动 ATK-MD0280 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-MD0280 模块触摸接口驱动

在图 2.1.2.1 中，atk_md0280_touch_spi.c 和 atk_md0280_touch_spi.h 是开发板与 ATK-MD0280 模块通讯而使用的模拟 SPI 驱动文件，主要用于获取 ATK-MD0280 模块的触摸状态，关于模拟 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模拟 SPI 对应的章节。

2.1.2.4 ATK-MD0280 模块驱动

在图 2.1.2.1 中，atk_md0280.c 和 atk_md0280.h 是 ATK-MD0280 模块的驱动文件，包含了 ATK-MD0280 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-MD0280 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_md0280_init()

该函数用于初始化 ATK-MD0280 模块，具体的代码，如下所示：

```
/**
 * @brief   ATK-MD0280 模块初始化
 * @param   无
 * @retval  ATK_MD0280_EOK      : ATK_MD0280 模块初始化成功
 *          ATK_MD0280_ERROR    : ATK_MD0280 模块初始化失败
 */
uint8_t atk_md0280_init(void)
{
    uint16_t chip_id;

    atk_md0280_hw_init();               /* ATK-MD0280 模块硬件初始化 */
    atk_md0280_fsmc_init();             /* ATK-MD0280 模块 FSMC 接口初始化 */
    chip_id = atk_md0280_get_chip_id(); /* 获取 ATK-MD0280 模块驱动器 ID */
    if ((chip_id != ATK_MD0280_CHIP_ID1) && (chip_id != ATK_MD0280_CHIP_ID2))
    {
        return ATK_MD0280_ERROR;
    }
    else
    {
        g_atk_md0280_sta.chip_id = chip_id;
        g_atk_md0280_sta.width = ATK_MD0280_LCD_WIDTH;
        g_atk_md0280_sta.height = ATK_MD0280_LCD_HEIGHT;
    }
    atk_md0280_reg_init();
    atk_md0280_set_disp_dir(ATK_MD0280_LCD_DISP_DIR_0);
    atk_md0280_clear(ATK_MD0280_WHITE);
    atk_md0280_display_on();
    atk_md0280_backlight_on();
#if (ATK_MD0280_USING_TOUCH != 0)
    atk_md0280_touch_init();
#endif

    return ATK_MD0280_EOK;
}
```

从上面的代码中可以看出，函数 atk_md0280_init()初始化 ATK-MD0280 模块主要就是初始化与 ATK-MD030 模块的 FSMC 通讯接口，FSMC 通讯接口初始化完成后就可以通过 FSMC 通讯接口初始化 ATK-MD0280 模块的寄存器，以完成 ATK-MD0280 模块的初始化，同时还通过宏定义 ATK_MD0280_USING_TOUCH 来使能或禁用 ATK-MD0280 模块的触摸驱动，若使能了 ATK-MD0280 模块的触摸驱动，还会调用函数 atk_md0280_touch_init()进行触摸的相关初始化，这个函数在下面会进行介绍。

2. 函数 atk_md0280_draw_point()

该函数用于在 ATK-MD0280 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-MD0280 模块 LCD 的所有显示操作，该函数的具体代码，如下所示：

```
/**
 * @brief   ATK-MD0280 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_md0280_draw_point(uint16_t x, uint16_t y, uint16_t color)
{
    atk_md0280_set_column_address(x, x);
    atk_md0280_set_page_address(y, y);
    atk_md0280_start_write_memory();
    atk_md0280_fsmc_write_dat(color);
}
```

从上面的代码中可以看出，在 ATK-MD0280 模块的 LCD 上画点需要三个步骤，首先就是确认待画点的位置（设置列地址和页地址），接着就是发送开始写显存命令，最后就可以写入待画点的颜色数据了。

3. 函数 atk_md0280_fill()

该函数用于对 ATK-MD0280 模块 LCD 的某一区域填充指定的单一颜色，虽然画点函数 atk_md0280_draw_point()能够完成 ATK-MD0280 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-MD0280 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_md0280_fill()的具体代码，如下所示：

```
/**
 * @brief   ATK-MD0280 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_md0280_fill(   uint16_t xs,
                        uint16_t ys,
                        uint16_t xe,
                        uint16_t ye,
                        uint16_t color)
{
    uint16_t x_index;
    uint16_t y_index;
```

```
    atk_md0280_set_column_address(xs, xe);
    atk_md0280_set_page_address(ys, ye);
    atk_md0280_start_write_memory();
    for (y_index=ys; y_index<=ye; y_index++)
    {
        for (x_index=xs; x_index<= xe; x_index++)
        {
            atk_md0280_fsmc_write_dat(color);
        }
    }
}
```

从上面的代码中可以函数，区域填充函数 atk_md0280_fill()与画点函数 atk_md0280_draw_point()很相似，只是画点函数在确定列地址和页地址时，确定的是一个点，而填充函数确定的是一个区域，画点函数在发送颜色数据的时候，发送的是一个点的颜色数据，而填充函数则是连续发送一个区域的颜色数据，这样一来，就大大地提高了大面积填充颜色的效率。

2.1.2.5 ATK-MD0280 模块触摸驱动

在图 2.1.2.1 中，atk_md0280_touch.c 和 atk_md0280_touch.h 是 ATK-MD0280 模块的触摸驱动文件，包含了 ATK-MD0280 模块触摸初始化、校准和扫描等相关的 ATK-MD0280 模块触摸 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_md0280_touch_init()

该函数用于初始化 ATK-MD0280 模块的触摸，具体的代码，如下所示：

```
/**
 * @brief   ATK-MD0280 模块触摸初始化
 * @param   无
 * @retval  无
 */
void atk_md0280_touch_init(void)
{
    atk_md0280_touch_hw_init();
    atk_md0280_touch_spi_init();
    atk_md0280_touch_calibration();
}
```

从上面的代码中可以看出，函数 atk_md0280_touch_init()初始化 ATK-MD0280 模块的触摸功能主要就是初始化与 ATK-MD030 模块触摸相关的模拟 SPI 通讯接口，模拟 SPI 通讯接口初始化完成后就可以通过模拟 SPI 通讯接口操作 ATK-MD0280 模块的触摸了，由于 ATK-MD0280 模块使用的是电阻触摸，因此需要对 ATK-MD0280 模块的触摸进行校准。

2. 函数 atk_md0280_touch_scan()

该函数用于扫描 ATK-MD0280 模块的触摸，具体的代码，如下所示：

```
/**
 * @brief   ATK-MD0280 模块触摸扫描
 * @param   x: 扫描到触摸的 X 坐标
 *          y: 扫描到触摸的 Y 坐标
 * @retval  ATK_MD0280_TOUCH_EOK    : 扫描到有效的触摸
```

```
 *          ATK_MD0280_TOUCH_ERROR  : 触摸坐标无效
 *          ATK_MD0280_TOUCH_EMPTY  : 未扫描到有效的触摸
 */
uint8_t atk_md0280_touch_scan(uint16_t *x, uint16_t *y)
{
    uint16_t x_adc;
    uint16_t y_adc;
    atk_md0280_lcd_disp_dir_t dir;
    uint16_t x_raw;
    uint16_t y_raw;

    if (ATK_MD0280_TOUCH_READ_PEN() == 0)
    {
        x_adc = atk_md0280_touch_get_adc2(ATK_MD0280_TOUCH_CMD_X);
        y_adc = atk_md0280_touch_get_adc2(ATK_MD0280_TOUCH_CMD_Y);

        x_raw = (int16_t)(x_adc - g_atk_md0280_touch_sta.center.x) /
                    g_atk_md0280_touch_sta.fac.x + ATK_MD0280_LCD_WIDTH / 2;
        y_raw = (int16_t)(y_adc - g_atk_md0280_touch_sta.center.y) /
                    g_atk_md0280_touch_sta.fac.y + ATK_MD0280_LCD_HEIGHT / 2;

        if((x_raw >= ATK_MD0280_LCD_WIDTH) || (y_raw >= ATK_MD0280_LCD_HEIGHT))
        {
            return ATK_MD0280_TOUCH_ERROR;
        }

        dir = atk_md0280_get_disp_dir();
        switch (dir)
        {
            case ATK_MD0280_LCD_DISP_DIR_0:
            {
                *x = x_raw;
                *y = y_raw;
                break;
            }
            case ATK_MD0280_LCD_DISP_DIR_90:
            {
                *x = y_raw;
                *y = atk_md0280_get_lcd_height() - x_raw;
                break;
            }
            case ATK_MD0280_LCD_DISP_DIR_180:
            {
                *x = atk_md0280_get_lcd_width() - x_raw;
```

```
                *y = atk_md0280_get_lcd_height() - y_raw;
                break;
            }
            case ATK_MD0280_LCD_DISP_DIR_270:
            {
                *x = atk_md0280_get_lcd_width() - y_raw;
                *y = x_raw;
                break;
            }
        }

        return ATK_MD0280_TOUCH_EOK;
    }

    return ATK_MD0280_TOUCH_EMPTY;
}
```

从上面的代码中可以看出，函数 atk_md0280_touch_scan()首先会判断是否有触摸，如果没有触摸那么就直接返回相应的错误，如果有触摸就会获取触摸的 ADC 值，然后根据校准的值，转化为坐标值，最后在根据屏幕的旋转方向，转换为屏幕上对应旋转方向的实际坐标值。

2.1.2.6 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)
{
    uint8_t ret;

    /* 初始化 ATK-MD0280 模块 */
    ret = atk_md0280_init();
    if (ret != 0)
    {
        printf("ATK-MD0280 init failed!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }
```

```
    /* ATK-MD0280 模块 LCD 清屏 */
    atk_md0280_clear(ATK_MD0280_WHITE);
    /* ATK-MD0280 模块 LCD 显示字符串 */
    atk_md0280_show_string( 10,
                            10,
                            ATK_MD0280_LCD_WIDTH,
                            32,
                            "STM32",
                            ATK_MD0280_LCD_FONT_32,
                            ATK_MD0280_RED);
    atk_md0280_show_string( 10,
                            42,
                            ATK_MD0280_LCD_WIDTH,
                            24,
                            "ATK-MD0280",
                            ATK_MD0280_LCD_FONT_24,
                            ATK_MD0280_RED);
    atk_md0280_show_string( 10,
                            66,
                            ATK_MD0280_LCD_WIDTH,
                            16,
                            "ATOM@ALIENTEK",
                            ATK_MD0280_LCD_FONT_16,
                            ATK_MD0280_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-MD0280 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-MD0280 模块的 LCD 显示屏上画线段，画线的本质实际上也就是画点，同时根据扫描的触摸坐标值，实时的修改立方体的位置。

2.1.3 实验现象

将 ATK-MD0280 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，在 ATK-MD0280 模块初始化前，会先通过串口显示本实验的相关信息，如下图所示：

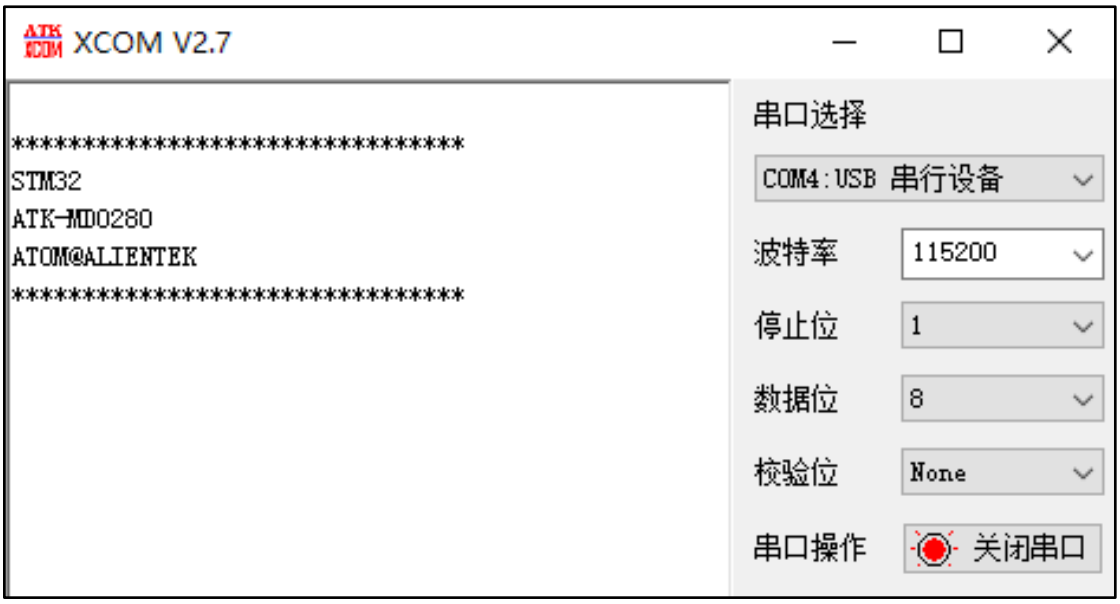

图 2.1.3.1 串口调试助手显示内容

ATK-MD0280 模块的初始化过程中，会在 ATK-MD0280 模块的 LCD 上显示与触摸校准的相关内容，如下图所示：

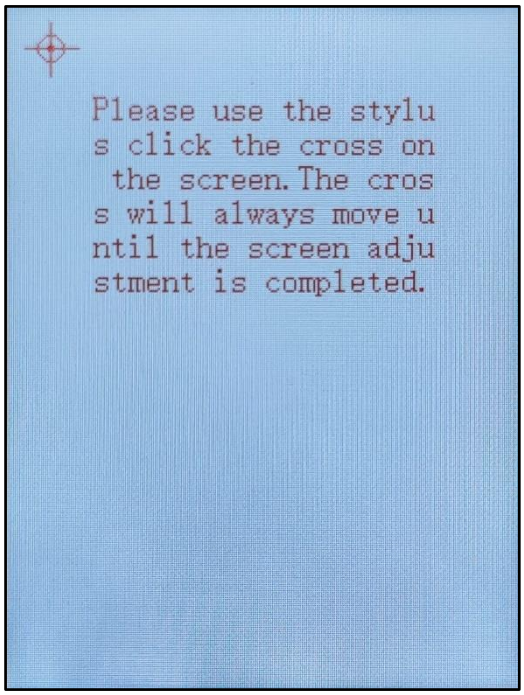

图 2.1.3.2 ATK-MD0280 模块 LCD 显示触摸校准相关内容

接下来请根据 ATK-MD0280 模块 LCD 上的内容提示，依次点击 ATK-MD0280 模块 LCD 上显示的 5 个点，以完成 ATK-MD0280 模块的触摸校准。

初始化通过后，会在 ATK-MD0280 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：

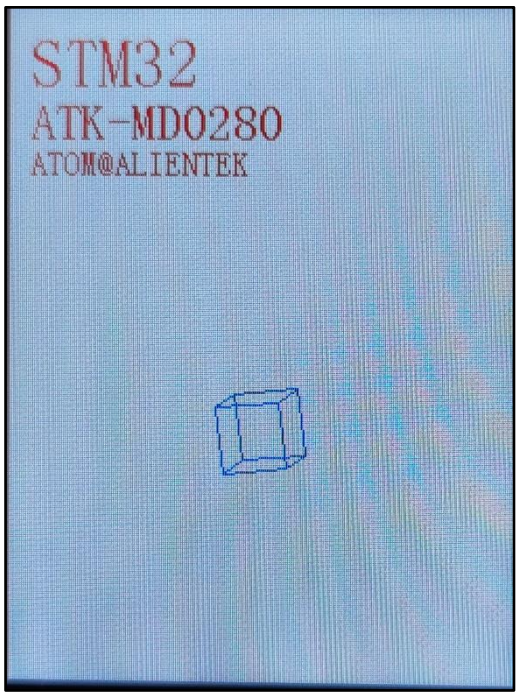

图 2.1.3.3 ATK-MD0280 模块 LCD 显示立方体 3D 旋转演示等信息

此时通过触摸屏幕，可以实时修改立方体的位置，如下图所示：

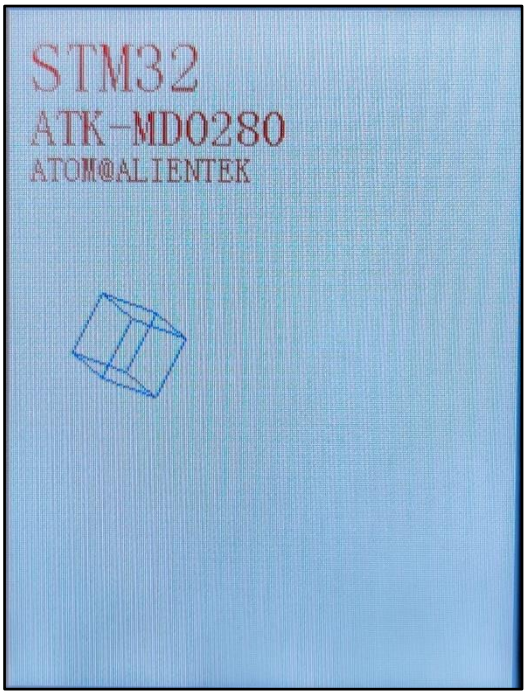

图 2.1.3.4 触摸修改立方体位置

2.2 ATK-MD0280 模块测试实验（GPIO）

2.2.1 功能说明

在本实验中，开发板主控芯片通过 GPIO 接口与 ATK-MD0280 模块进行通讯，从而完成对 ATK-MD0280 模块的初始化配置以及操作 ATK-MD0280 模块的 LCD 显示各种内容，

同时通过模拟 SPI 接口与 ATK-MD0280 模块进行通讯，从而获取 ATK-MD0280 模块的触摸数据。

2.2.2 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_MD0280 子文件夹，该文件夹中就包含了 ATK-MD0280 模块的驱动文件，如下图所示：

```
./Drivers/BSP/ATK_MD0280/
|-- atk_md0280.c
|-- atk_md0280.h
|-- atk_md0280_font.h
|-- atk_md0280_gpio.c
|-- atk_md0280_gpio.h
|-- atk_md0280_touch.c
|-- atk_md0280_touch.h
|-- atk_md0280_touch_spi.c
`-- atk_md0280_touch_spi.h
```

图 2.2.2.1 ATK-MD0280 模块驱动代码

2.2.2.1 ATK-MD0280 模块接口驱动

在图 2.1.2.1 中，atk_md0280_gpio.c 和 atk_md0280_gpio.h 是开发板与 ATK-MD0280 模块通讯而使用的 GPIO 驱动文件，对于没有 FMC 或 FSMC 接口的开发板（如正点原子 MiniSTM32F103 开发板），可以使用 GPIO 直接驱动 ATK-MD0280 模块的 LCD。关于使用 GPIO 驱动 TFTLCD 的介绍，请查看正点原子各个开发板对应的开发指南中，使用 GPIO 驱动 TFTLCD 对应的章节。

其余的源码，均与第 2.1 小节“ATK-MD0280 模块测试实验（FMC&FSMC）”中的源码类似，请查看第 2.1 小节“ATK-MD0280 模块测试实验（FMC&FSMC）”中对应的内容。

2.2.3 实验现象

本实验的实验现象与第 2.1 小节“ATK-MD0280 模块测试实验（FMC&FSMC）”一致，请查看第 2.1 小节“ATK-MD0280 模块测试实验（FMC&FSMC）”。

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/2.8-TFT_LCD-320240.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