Application Note

ATK-MS7620 模块使用说明

高性能手势识别模块

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	2023/03/07	新增阿波罗 F429 与 F767 硬件连接描述
V1.2	2023/04/17	新增阿波罗 H743 硬件连接描述

目 录

1，硬件连接

1.1 正点原子 MiniSTM32F103 开发板

ATK-MS7620 模块可直接与正点原子 MiniSTM32F103 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
MiniSTM32F103 开发板	3.3V/5V	GND	PD2	PC12	-	-

表 1.1.1 ATK-MS7620 模块与 MiniSTM32F103 开发板连接关系

1.2 正点原子精英 STM32F103 开发板

ATK-MS7620 模块可直接与正点原子精英 STM32F103 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
精英 STM32F103 开发板	3.3V/5V	GND	PB11	PB10	-	-

表 1.2.1 ATK-MS7620 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-MS7620 模块可直接与正点原子战舰 STM32F103 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
战舰 STM32F103 开发板	3.3V/5V	GND	PB11	PB10	-	-

表 1.3.1 ATK-MS7620 模块与战舰 STM32F103 开发板连接关系

注意，若要使用正点原子战舰 STM32F103 开发板的 ATK MODULE 接口连接 ATK-MS7620 模块，需要用跳线帽将开发板板载的 P8 接线端子的 PB10(TX)和 GBC_RX 以及 PB11(RX)和 GBC_TX 用跳线帽进行短接，如下图所示：

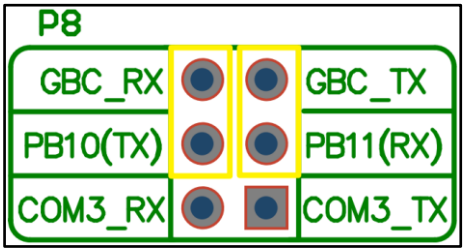

图 1.3.1 战舰 STM32F103 开发板 P8 接线端子

1.4 正点原子探索者 STM32F407 开发板

ATK-MS7620 模块可直接与正点原子探索者 STM32F407 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
探索者 STM32F407 开发板	3.3V/5V	GND	PB11	PB10	-	-

表 1.4.1 ATK-MS7620 模块与探索者 STM32F407 开发板连接关系

注意，若要使用正点原子探索者 STM32F407 开发板的 ATK MODULE 接口连接 ATK-MS7620 模块，需要用跳线帽将开发板板载的 P2 接线端子的 PB10(TX)和 GBC_RX 以及 PB11(RX)和 GBC_TX 用跳线帽进行短接，如下图所示：

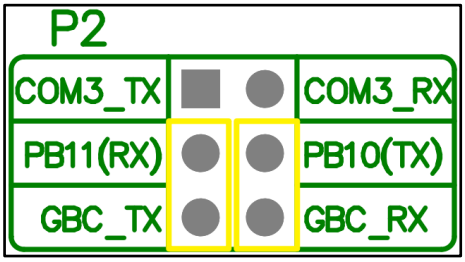

图 1.4.1 探索者 STM32F407 开发板 P2 接线端子

1.5 正点原子 F407 电机控制开发板

ATK-MS7620 模块可直接与正点原子 F407 电机控制开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
F407 电机控制开发板	3.3V/5V	GND	PC11	PC10	-	-

表 1.5.1 ATK-MS7620 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-MS7620 模块可直接与正点原子 MiniSTM32H750 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
MiniSTM32H750 开发板	3.3V/5V	GND	PA3	PA2	-	-

表 1.6.1 ATK-MS7620 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-MS7620 模块可直接与正点原子阿波罗 STM32F429 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
阿波罗 STM32F429 开发板	3.3V/5V	GND	PB11	PB10	-	-

表 1.7.1 ATK-MS7620 模块与阿波罗 STM32F429 开发板连接关系

注意，若要使用正点原子阿波罗 STM32F429 开发板的 ATK MODULE 接口连接 ATK-MS7620 模块，需要用跳线帽将开发板板载的 P9 接线端子的 PB10(TX)和 GBC_RX 以及 PB11(RX)和 GBC_TX 用跳线帽进行短接，如下图所示：

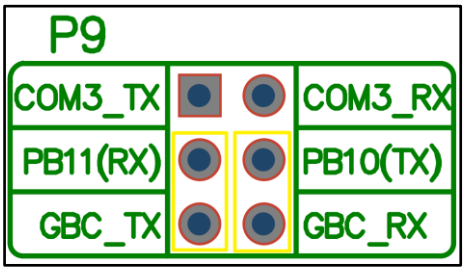

图 1.7.1 阿波罗 STM32F429 开发板 P9 接线端子

1.8 正点原子阿波罗 STM32F767 开发板

ATK-MS7620 模块可直接与正点原子阿波罗 STM32F767 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
阿波罗 STM32F767 开发板	3.3V/5V	GND	PB10	PB11	-	-

表 1.8.1 ATK-MS7620 模块与阿波罗 STM32F767 开发板连接关系

注意，若要使用正点原子阿波罗 STM32F767 开发板的 ATK MODULE 接口连接 ATK-MS7620 模块，需要用跳线帽将开发板板载的 P9 接线端子的 PB10(TX)和 GBC_RX 以及 PB11(RX)和 GBC_TX 用跳线帽进行短接，如下图所示：

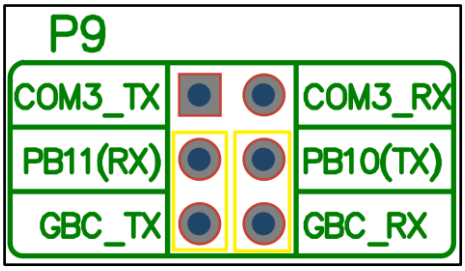

图 1.8.1 阿波罗 STM32F767 开发板 P9 接线端子

1.9 正点原子阿波罗 STM32H743 开发板

ATK-MS7620 模块可直接与正点原子阿波罗 STM32H743 开发板板载的 ATK 模块接口（ATK MODULE）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-MS7620 模块	VCC	GND	SDA	SCL	INT	NC
阿波罗 STM32H743 开发板	3.3V/5V	GND	PB10	PB11	-	-

表 1.9.1 ATK-MS7620 模块与阿波罗 STM32H743 开发板连接关系

注意，若要使用正点原子阿波罗 STM32H743 开发板的 ATK MODULE 接口连接 ATK-MS7620 模块，需要用跳线帽将开发板板载的 P9 接线端子的 PB10(TX)和 GBC_RX 以及 PB11(RX)和 GBC_TX 用跳线帽进行短接，如下图所示：

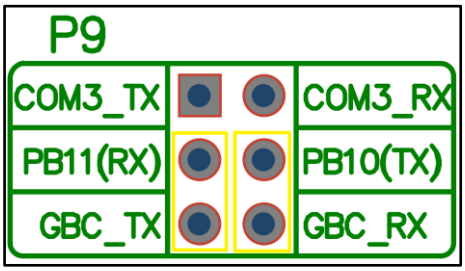

图 1.9.1 阿波罗 STM32H743 开发板 P9 接线端子

2，实验功能

2.1 ATK-MS7620 模块接近检测测试实验

2.1.1 功能说明

在本实验中，开发板主控芯片通过模拟 IIC 与 ATK-MS7620 模块进行通讯，从而获取 ATK-MS7620 模块在接近检测模式下检测到的物体亮度和大小，并将结果通过串口打印至串口调试助手。

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_MS7620 子文件夹，该文件夹中就包含了 ATK-MS7620 模块的驱动文件，如下图所示：

```
./Drivers/BSP/ATK_MS7620/
|-- atk_ms7620.c
|-- atk_ms7620.h
|-- atk_ms7620_iic.c
`-- atk_ms7620_iic.h
```

图 2.1.2.1 ATK-MS7620 模块驱动代码

2.1.2.1 ATK-MS7620 模块接口驱动

在图 2.1.2.1 中，atk_ms7620_iic.c 和 atk_ms7620_iic.h 是开发板与 ATK-MS7620 模块通讯而使用的模拟 IIC 驱动文件，关于模拟 IIC 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 IIC 对应的章节。

2.1.2.2 ATK-MS7620 模块驱动

在图 2.1.2.1 中，atk_ms7620.c 和 atk_ms7620.h 是 ATK-MS7620 模块的驱动文件，包含了 ATK-MS7620 模块初始化、读写寄存器的相关 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_ms7620_init()

该函数用于初始化 ATK-MS7620 模块，具体的代码，如下所示：

```
/**
 * @brief   ATK-MS7620 模块初始化
 * @param   无
 * @retval  ATK_MS7620_EOK      : ATK-MS7620 模块初始化成功
 *          ATK_MS7620_ERROR    : ATK-MS7620 模块初始化失败
 */
uint8_t atk_ms7620_init(void)
{
    uint8_t ret;

    delay_ms(1);                                /* 等待至少 700us */
    atk_ms7620_iic_init();                      /* 初始化 IIC 接口 */
    ret = atk_ms7620_get_wakeup_status();       /* 获取唤醒状态 */
```

```
    if (ret != ATK_MS7620_EOK)
    {
        return ATK_MS7620_ERROR;
    }
    atk_ms7620_initial_register();          /* 初始化寄存器配置 */

    return ATK_MS7620_EOK;
}
```

从上面的代码中可以看出，对 ATK-MS7620 模块进行初始化操作之前，先有一个延时，这个延时是必须的，且延时时长应不小于 700us，其目的是为了等待 ATK-MS7620 模块上电后稳定。延时过后才是初始化于 ATK-MS7620 模块的模拟 IIC 通讯接口，接下来就是对 ATK-MS7620 模块的一系列初始化操作了。

2. 函数 atk_ms7620_get_obj_brightness()

该函数用于在 ATK-MS7620 处于接近检测模式下，获取 ATK-MS7620 模块测量的物体亮度，具体的代码，如下所示：

```
/**
 * @brief   ATK-MS7620 模块获取物体亮度
 * @param   brightness: 物体亮度，范围 0~255
 * @retval  ATK_MS7620_EOK      : 获取物体亮度成功
 *          ATK_MS7620_ERROR    : 获取物体亮度失败
 *          ATK_MS7620_EINVAL   : 函数参数有误
 */
uint8_t atk_ms7620_get_obj_brightness(uint8_t *brightness)
{
    uint8_t ret;
    uint8_t _brightness[1];

    if (brightness == NULL)
    {
        return ATK_MS7620_EINVAL;
    }

    atk_ms7620_switch_reg_bank(ATK_MS7620_BANK0);
    ret = atk_ms7620_read_byte( ATK_MS7620_IIC_ADDR,
                                ATK_MS7620_REG_OBJ_BRIGHTNESS,
                                _brightness);
    if (ret != ATK_MS7620_EOK)
    {
        return ATK_MS7620_ERROR;
    }

    *brightness = _brightness[0];

    return ATK_MS7620_EOK;
```

```
}
```

从上面的代码中可以看出，获取 ATK-MS7620 模块在接近检测模式下测量的物体亮度，主要就是读取 ATK-MS7620 模块指定寄存器的值，该寄存器的地址由代码中的宏 ATK_MS7620_REG_OBJ_BRIGHTNESS 定义。

3. 函数 atk_ms7620_get_obj_size()

该函数用于在 ATK-MS7620 处于接近检测模式下，获取 ATK-MS7620 模块测量的物体大小，具体的代码，如下所示：

```
/**
 * @brief   ATK-MS7620 模块获取物体大小
 * @param   size: 物体大小，范围 0~900
 * @retval  ATK_MS7620_EOK      : 获取物体大小成功
 *          ATK_MS7620_ERROR    : 获取物体大小失败
 *          ATK_MS7620_EINVAL   : 函数参数有误
 */
uint8_t atk_ms7620_get_obj_size(uint16_t *size)
{
    uint8_t ret;
    uint8_t _size[2];

    if (size == NULL)
    {
        return ATK_MS7620_EINVAL;
    }

    atk_ms7620_switch_reg_bank(ATK_MS7620_BANK0);
    ret  = atk_ms7620_read_byte(    ATK_MS7620_IIC_ADDR,
                                    ATK_MS7620_REG_OBJ_SIZE_1,
                                    &_size[0]);
    ret += atk_ms7620_read_byte(    ATK_MS7620_IIC_ADDR,
                                    ATK_MS7620_REG_OBJ_SIZE_2,
                                    &_size[1]);

    if (ret != ATK_MS7620_EOK)
    {
        return ATK_MS7620_ERROR;
    }

    *size = (((uint16_t)_size[1] << 8) & 0x0F00) | _size[0];

    return ATK_MS7620_EOK;
}
```

从上面的代码中可以看出，获取 ATK-MS7620 模块在接近检测模式下测量的物体大小，主要就是读取 ATK-MS7620 模块指定寄存器的值，涉及到了两个寄存器，这两个寄存器的地址分别为在程序中由宏 ATK_MS7620_REG_OBJ_SIZE_1 和宏

ATK_MS7620_REG_OBJ_SIZE_2 定于，其中寄存器 ATK_MS7620_REG_OBJ_SIZE_1[7:0]组成物体大小数据的低 8 位，寄存器 ATK_MS7620_REG_OBJ_SIZE_2[3:0]组成物体大小数据据的第 8~11 位。

4. 函数 atk_ms7620_get_gesture()

该函数用于在 ATK-MS7620 处于手势检测模式下，获取 ATK-MS7620 模块检测出的手势，具体的代码，如下所示：

```
/**
 * @brief   ATK-MS7620 模块获取手势
 * @param   gesture: 手势
 * @retval  ATK_MS7620_EOK      : 获取手势成功
 *          ATK_MS7620_ERROR    : 获取手势失败
 *          ATK_MS7620_EINVAL   : 函数参数有误
 */
uint8_t atk_ms7620_get_gesture(atk_ms7620_gesture_t *gesture)
{
    uint8_t ret;
    union
    {
        uint8_t byte[2];
        uint16_t halfword;
    }_flag;

    if (gesture == NULL)
    {
        return ATK_MS7620_EINVAL;
    }

    atk_ms7620_switch_reg_bank(ATK_MS7620_BANK0);
    ret = atk_ms7620_read_byte(     ATK_MS7620_IIC_ADDR,
                                    ATK_MS7620_REG_INT_FLAG_1,
                                    &_flag.byte[0]);
    ret += atk_ms7620_read_byte(    ATK_MS7620_IIC_ADDR,
                                    ATK_MS7620_REG_INT_FLAG_2,
                                    &_flag.byte[1]);
    if (ret != ATK_MS7620_EOK)
    {
        return ATK_MS7620_ERROR;
    }

    switch (_flag.halfword)
    {
        case ATK_MS7620_GES_UP_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_UP;
```

```
            break;
        }
        case ATK_MS7620_GES_DOWN_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_DOWN;
            break;
        }
        case ATK_MS7620_GES_LEFT_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_LEFT;
            break;
        }
        case ATK_MS7620_GES_RIGHT_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_RIGHT;
            break;
        }
        case ATK_MS7620_GES_FORWARD_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_FORWARD;
            break;
        }
        case ATK_MS7620_GES_BACKWARD_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_BACKWARD;
            break;
        }
        case ATK_MS7620_GES_CLOCKWISE_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_CLOCKWISE;
            break;
        }
        case ATK_MS7620_GES_ANTICLOCKWISE_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_ANTICLOCKWISE;
            break;
        }
        case ATK_MS7620_GES_WAVE_FLAG:
        {
            *gesture = ATK_MS7620_GESTURE_WAVE;
            break;
        }
        default:
        {
```

```
            return ATK_MS7620_ERROR;
        }
    }

    return ATK_MS7620_EOK;
}
```

从上面的代码中可以看出，获取 ATK-MS7620 模块在手势检测模式下检测的手势，主要就是读取 ATK-MS7620 模块指定寄存器的值，涉及到了两个寄存器，这两个寄存器的地址分别为在程序中由宏 ATK_MS7620_REG_INT_FLAG_1 和宏 ATK_MS7620_REG_INT_FLAG_2 定于，这两个寄存器中的不同比特位表示了检测到的一种手势，通过判断这两个寄存器中指定的比特位是否被置 1，即可获取检测到的手势。

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)
{
    uint8_t ret;
    uint8_t brightness;
    uint16_t size;

    /* 初始化 ATK-MS7620 模块 */
    ret = atk_ms7620_init();
    if (ret != 0)
    {
        printf("ATK-MS7620 init failed!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }

    /* 配置 ATK-MS7620 模块为接近检测模式 */
    ret = atk_ms7620_mode_config(ATK_MS7620_MODE_PS);
    if (ret != 0)
    {
        printf("ATK_MS7620 config failed!\r\n");
        while (1)
        {
```

```
            LED0_TOGGLE();
            delay_ms(200);
        }
    }

    printf("ATK-MS7620 config succedded!\r\n");

    while (1)
    {
        /* 获取物体亮度和大小 */
        ret  = atk_ms7620_get_obj_brightness(&brightness);
        ret += atk_ms7620_get_obj_size(&size);
        if (ret == ATK_MS7620_EOK)
        {
            printf("Object brightness: %d, size: %d\r\n", brightness, size);
        }
    }
}
```

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，就是先初始化 ATK-MS7620 模块，然后将 ATK-MS7620 模块配置为接近检测模式，接下来就可以获取 ATK-MS7620 模块在接近检测模式下测量到的物体亮度和大小了，并将获取到的结果实时地通过串口输出。

2.1.3 实验现象

将 ATK-MS7620 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-MS7620 模块初始化并配置成功，则会在串口调试助手上显示相应的提示，如下图所示：

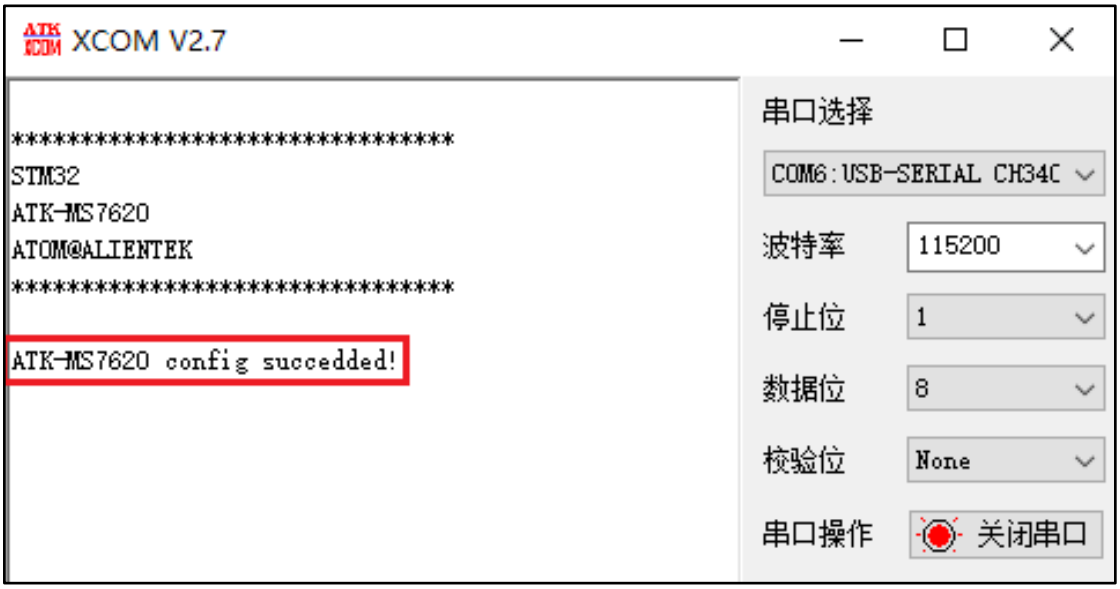

图 2.1.3.3 ATK-MS7620 模块初始化并配置成功

接下来就能够获取 ATK-MS7620 模块在接近检测模式下测量到的物体亮度和大小了，如下图所示：

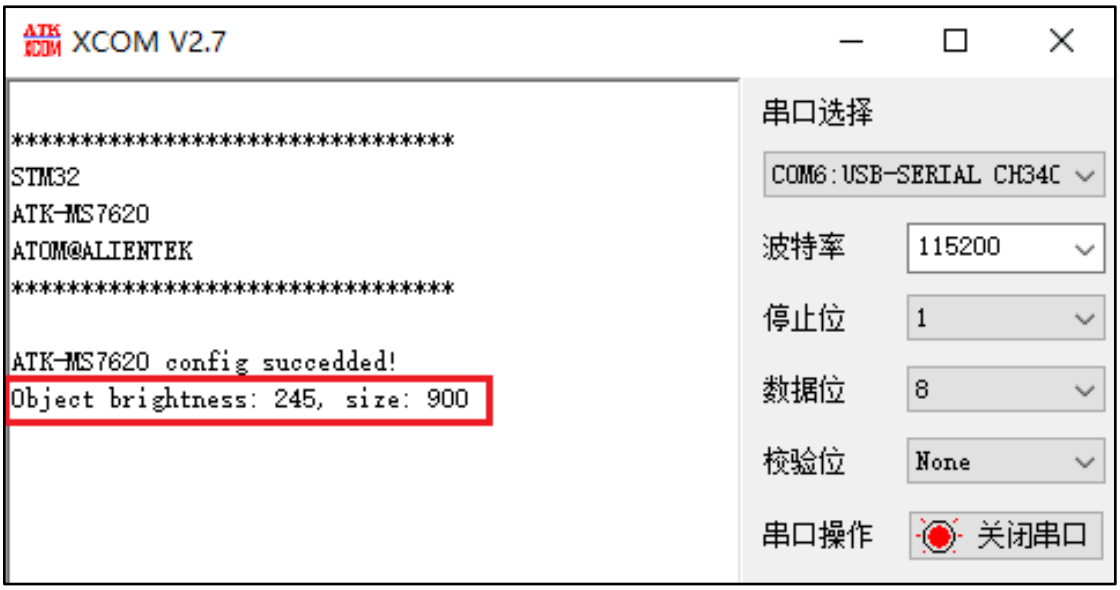

图 2.1.3.4 获取物体亮度和大小

2.2 ATK-MS7620 模块手势检测测试实验

2.2.1 功能说明

在本实验中，开发板主控芯片通过模拟 IIC 与 ATK-MS7620 模块进行通讯，从而获取 ATK-MS7620 模块在手势检测模式下检测到的手势，并将结果通过串口打印至串口调试助手。

2.2.2 源码解读

2.2.2.1 ATK-MS7620 模块接口驱动

本实验中，ATK-MS7620 模块接口的驱动代码与 2.1 小节“ATK-MS7620 模块接近检测实验”中的接口驱动代码一致，请见第 2.1.2.1 小节“ATK-MS7620 模块接口驱动”。

2.2.2.2 ATK-MS7620 模块驱动

本实验中，ATK-MS7620 模块的驱动代码与 2.1 小节“ATK-MS7620 模块接近检测实验”中的驱动代码一致，请见第 2.1.2.2 小节“ATK-MS7620 模块驱动”。

2.2.2.4 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)
{
    uint8_t ret;
    atk_ms7620_gesture_t gesture;
```

```
/* 初始化 ATK-MS7620 模块 */
ret = atk_ms7620_init();
if (ret != 0)
{
    printf("ATK-MS7620 init failed!\r\n");
    while (1)
    {
        LED0_TOGGLE();
        delay_ms(200);
    }
}

/* 配置 ATK-MS7620 模块为手势检测模式 */
ret = atk_ms7620_mode_config(ATK_MS7620_MODE_GESTURE);
if (ret != 0)
{
    printf("ATK_MS7620 config failed!\r\n");
    while (1)
    {
        LED0_TOGGLE();
        delay_ms(200);
    }
}

printf("ATK-MS7620 config succedded!\r\n");

while (1)
{
    /* 获取手势 */
    ret = atk_ms7620_get_gesture(&gesture);
    if (ret == ATK_MS7620_EOK)
    {
        switch (gesture)
        {
            case ATK_MS7620_GESTURE_UP:
            {
                printf("Gesture: Up\r\n");
                break;
            }
            case ATK_MS7620_GESTURE_DOWN:
            {
                printf("Gesture: Down\r\n");
                break;
            }
```

```
                case ATK_MS7620_GESTURE_LEFT:
                {
                    printf("Gesture: Left\r\n");
                    break;
                }
                case ATK_MS7620_GESTURE_RIGHT:
                {
                    printf("Gesture: Right\r\n");
                    break;
                }
                case ATK_MS7620_GESTURE_FORWARD:
                {
                    printf("Gesture: Forward\r\n");
                    break;
                }
                case ATK_MS7620_GESTURE_BACKWARD:
                {
                    printf("Gesture: Backward\r\n");
                    break;
                }
                case ATK_MS7620_GESTURE_CLOCKWISE:
                {
                    printf("Gesture: Clockwise\r\n");
                    break;
                }
                case ATK_MS7620_GESTURE_ANTICLOCKWISE:
                {
                    printf("Gesture: Anticlockwise\r\n");
                    break;
                }
                case ATK_MS7620_GESTURE_WAVE:
                {
                    printf("Gesture: Wave\r\n");
                    break;
                }
                default:
                {
                    break;
                }
            }
        }
    }
}
```

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，就是先初始化

ATK-MS7620 模块，然后将 ATK-MS7620 模块配置为接近检测模式，接下来就可以获取 ATK-MS7620 模块在手势检测模式下检测到的手势，并将获取到的结果实时地通过串口输出。

2.2.3 实验现象

将 ATK-MS7620 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 2.2.3.1 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 2.2.3.2 串口调试助手显示内容一

接下来，如果 ATK-MS7620 模块初始化并配置成功，则会在串口调试助手上显示相应的提示，如下图所示：

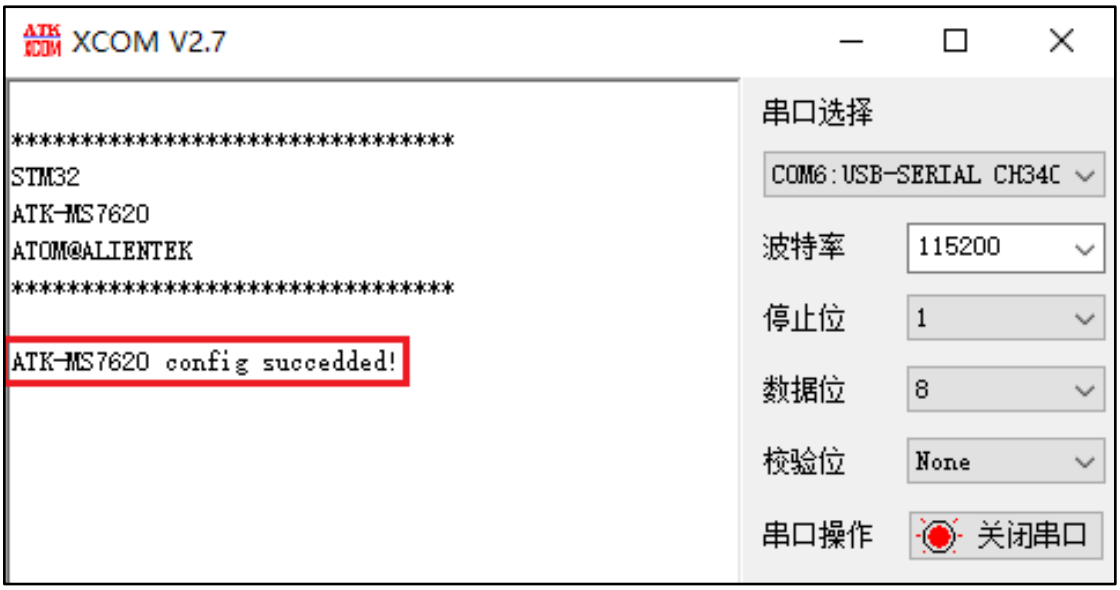

图 2.2.3.3 ATK-MS7620 模块初始化并配置成功

接下来就能够获取 ATK-MS7620 模块在手势检测模式下检测到的手势结果了，如下图所示：

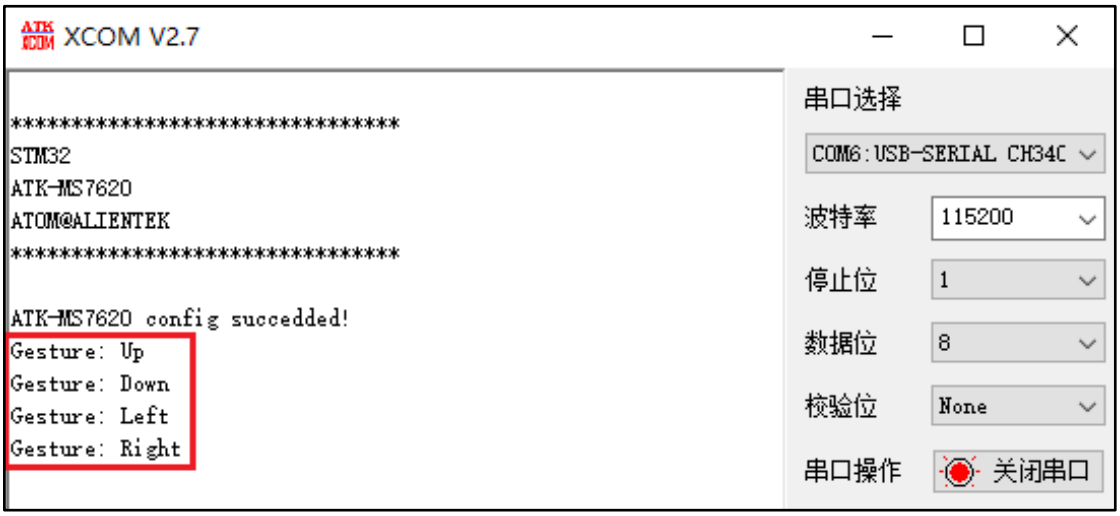

图 2.2.3.4 获取物体亮度和大小

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/other/ATK-PAJ7620.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