Application Note

ATK-MW8266D 模块使用说明

高性能 UART-WIFI 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
1.1 正点原子 MiniSTM32F103 开发板.....1
1.2 正点原子精英 STM32F103 开发板.....1
1.3 正点原子战舰 STM32F103 开发板.....1
1.4 正点原子探索者 STM32F407 开发板.....2
1.5 正点原子 F407 电机控制开发板.....2
1.6 正点原子 MiniSTM32H750 开发板.....2
1.7 正点原子阿波罗 STM32F429 开发板.....2
1.8 正点原子阿波罗 STM32F767 开发板.....3
1.9 正点原子阿波罗 STM32H743 开发板.....3
2，实验功能.....5
2.1 ATK-MW8266D 模块 TCP 透传实验.....5
2.1.1 功能说明.....5
2.1.2 源码解读.....5
2.1.3 实验现象.....10
2.2 ATK-MW8266D 模块原子云连接实验.....14
2.2.1 功能说明.....14
2.2.2 源码解读.....14
2.2.3 实验现象.....16
3，其他.....21

1，硬件连接

1.1 正点原子 MiniSTM32F103 开发板

ATK-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
MiniSTM32F103 开发板	3.3V/5V	GND	PD2	PC12	PC4	-

表 1.1.1 ATK-MW8266D 模块与 MiniSTM32F103 开发板连接关系

1.2 正点原子精英 STM32F103 开发板

ATK-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
精英 STM32F103 开发板	3.3V/5V	GND	PB11	PB10	PA4	-

表 1.2.1 ATK-MW8266D 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-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
战舰 STM32F103 开发板	3.3V/5V	GND	PB11	PB10	PA4	-

表 1.3.1 ATK-MW8266D 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-MW8266D 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：

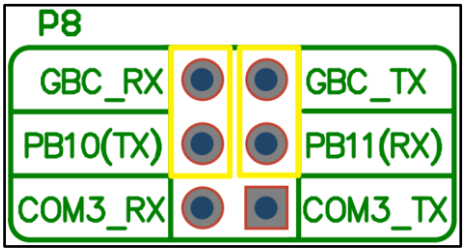

图 1.3.1 战舰 STM32F103 开发板 P8 接线端子

1.4 正点原子探索者 STM32F407 开发板

ATK-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
探索者 STM32F407 开发板	3.3V/5V	GND	PB11	PB10	PF6	-

表 1.4.1 ATK-MW8266D 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-MW8266D 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：

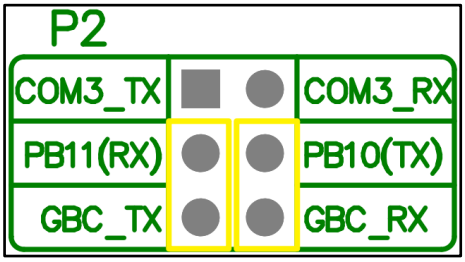

图 1.4.1 探索者 STM32F407 开发板 P2 接线端子

1.5 正点原子 F407 电机控制开发板

ATK-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
F407 电机控制开发板	3.3V/5V	GND	PC11	PC10	PI10	-

表 1.5.1 ATK-MW8266D 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-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
MiniSTM32H750 开发板	3.3V/5V	GND	PA3	PA2	PC2	-

表 1.6.1 ATK-MW8266D 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-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
阿波罗 STM32F429 开发板	3.3V/5V	GND	PB11	PB10	PI11	-

表 1.7.1 ATK-MW8266D 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-MW8266D 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：

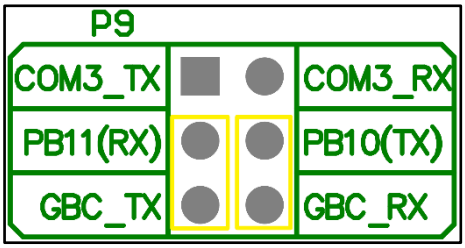

图 1.7.1 阿波罗 STM32F429 开发板 P9 接线端子

1.8 正点原子阿波罗 STM32F767 开发板

ATK-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
阿波罗 STM32F767 开发板	3.3V/5V	GND	PB11	PB10	PI11	-

表 1.8.1 ATK-MW8266D 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-MW8266D 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：

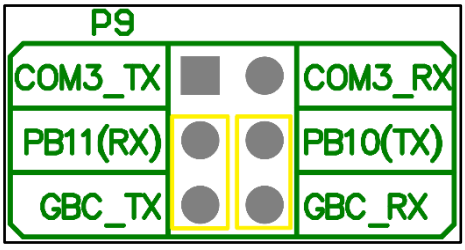

图 1.8.1 阿波罗 STM32F767 开发板 P9 接线端子

1.9 正点原子阿波罗 STM32H743 开发板

ATK-MW8266D 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-MW8266D 模块	VCC	GND	TXD	RXD	RST	IO_0
阿波罗 STM32H743 开发板	3.3V/5V	GND	PB11	PB10	PI11	-

表 1.9.1 ATK-MW8266D 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-MW8266D 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：

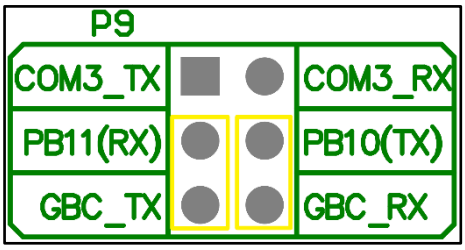

图 1.9.1 阿波罗 STM32H743 开发板 P9 接线端子

2，实验功能

2.1 ATK-MW8266D 模块 TCP 透传实验

2.1.1 功能说明

在本实验中，开发板主控芯片通过串口与 ATK-MW8266D 模块进行通讯，并在上电后自动根据配置信息，配置 ATK-MW8266D 模块连接 WIFI、TCP 服务器（与 ATK-MW8266D 模块所连接 WIFI 在同一局域网的电脑作为 TCP 服务器），成功连接 WIFI 后，将在串口调试助手和 LCD 上显示 ATK-MW8266D 模块的 IP 地址，随后便可通过按键对 ATK-MW8266D 模块进行 AT 指令测试和配置其进入或退出透传模式，AT 指令测试的测试结果将显示在串口调试助手上，当模块进入透传模式后，可通过按键发送数据至 TCP 服务器，同时也可接收来自 TCP 服务器的数据，并实时显示在串口调试助手上。

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_MW8266D 子文件夹，该文件夹中就包含了 ATK-MW8266D 模块的驱动文件，如下图所示：

```
./Drivers/BSP/ATK_MW8266D/
|-- atk_mw8266d.c
|-- atk_mw8266d.h
|-- atk_mw8266d_uart.c
`-- atk_mw8266d_uart.h
```

图 2.1.2.1 ATK-MW8266D 模块驱动代码

2.1.2.1 ATK-MW8266D 模块接口驱动

在图 2.1.2.1 中，atk_mw8266d_uart.c 和 atk_mw8266d_uart.h 是开发板与 ATK-MW8266D 模块通讯而使用的 UART 驱动文件，关于 UART 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 UART 对应的章节。

值得一提的是，由于 ATK-MW8266D 模块通过 UART 发送给主控芯片的数据的长度是不固定的，因此主控芯片就无法直接通过接收到数据的长度来判断 ATK-MW8266D 模块传来的一帧数据是否完成。对于这种通过 UART 接收不定长数据的情况，可以通过 UART 总线是否空闲来判断一帧的传输是否完成，恰巧 STM32 的 UART 提供了总线空闲中断功能，因此可以开启 UART 的总线空闲中断，并在中断中做相应的处理，具体的实现过程可以查看 ATK-MW8266D 模块的模块接口驱动代码，这里不做过多的描述。

2.1.2.2 ATK-MW8266D 模块驱动

在图 2.1.2.1 中，atk_mw8266d.c 和 atk_mw8266d.h 是 ATK-MW8266D 模块的驱动文件，包含了 ATK-MW8266D 模块初始化、硬件复位、发送 AT 指令的相关 API 函数和部分 AT 指令的封装函数。函数比较多，下面仅介绍几个重要的 API 函数。

1. 函数 atk_mw8266d_init()

该函数用于初始化 ATK-MW8266D 模块，具体的代码，如下所示：

```
/**
 * @brief   ATK-MW8266D 初始化
 * @param   baudrate: ATK-MW8266D UART 通讯波特率
```

```
 * @retval  ATK_MW8266D_EOK  : ATK-MW8266D 初始化成功，函数执行成功
 *          ATK_MW8266D_ERROR: ATK-MW8266D 初始化失败，函数执行失败
 */
uint8_t atk_mw8266d_init(uint32_t baudrate)
{
    atk_mw8266d_hw_init();                              /* ATK-MW8266D 硬件初始化 */
    atk_mw8266d_hw_reset();                             /* ATK-MW8266D 硬件复位 */
    atk_mw8266d_uart_init(baudrate);                    /* ATK-MW8266D UART 初始化 */
    if (atk_mw8266d_at_test() != ATK_MW8266D_EOK)       /* ATK-MW8266D AT 指令测试 */
    {
        return ATK_MW8266D_ERROR;
    }

    return ATK_MW8266D_EOK;
}
```

从上面的代码中可以看出，函数 atk_mw8266d_init()会对 ATK-MW8266D 模块进行硬件复位（拉低 ATK-MW8266D 模块的 RST 引脚，随后拉高），然后初始化主控芯片与 ATK-MW8266D 模块的 UART，最后进行 AT 指令测试，若 AT 指令测试成功，则说明 ATK-MW8266D 模块及其通讯接口初始化成功，反之，则初始化失败。

2. 函数 atk_mw8266d_at_test()

该函数用于对 ATK-MW8266D 模块进行 AT 指令测试，可以由此判断主控与 ATK-MW8266D 模块的通讯是否无误，具体的代码，如下所示：

```
/**
 * @brief   ATK-MW8266D AT 指令测试
 * @param   无
 * @retval  ATK_MW8266D_EOK     : AT 指令测试成功
 *          ATK_MW8266D_ERROR   : AT 指令测试失败
 */
uint8_t atk_mw8266d_at_test(void)
{
    uint8_t ret;
    uint8_t i;

    for (i=0; i<10; i++)
    {
        ret = atk_mw8266d_send_at_cmd("AT", "OK", 500);
        if (ret == ATK_MW8266D_EOK)
        {
            return ATK_MW8266D_EOK;
        }
    }

    return ATK_MW8266D_ERROR;
}
```

从上面的代码中可以看出，该函数会通过 UART 向 ATK-MW8266D 模块发送“AT”字符串（函数 atk_mw8266d_send_at_cmd()会根据通讯规则在字符串末尾添加换行符，通讯规则见《ATK-MW8266D 模块用户手册》），并在一段时间内等待 ATK-MW8266D 模块的“OK”响应，如果收到 ATK-MW8266D 模块的“OK”响应，说明主控芯片与 ATK-MW8266D 模块的 UART 通讯正常，AT 指令测试成功，反之，则说明 AT 指令测试失败，主控芯片不能与 ATK-MW8266D 模块进行正常的通讯。

3. 函数 atk_mw8266d_send_at_cmd()

该函数主要实现主控芯片与 ATK-MW8266D 模块的 AT 指令传输，本驱动代码中的大部分驱动函数都是基于该函数实现的，但由于 ATK-MW8266D 的 AT 指令众多，在驱动代码中无法一一实验，因此在使用 ATK-MW8266D 模块的时候，可以根据《ATK-MW8266D 模块用户手册》中列出的 AT 指令，并参考驱动文件中的驱动函数，对函数 atk_mw8266d_send_at_cmd() 进行简单的封装，即可实现相应的功能。函数 atk_mw8266d_send_at_cmd()的具体代码，如下所示：

```
/**
 * @brief   ATK-MW8266D 发送 AT 指令
 * @param   cmd     : 待发送的 AT 指令
 *          ack     : 等待的响应
 *          timeout : 等待超时时间
 * @retval  ATK_MW8266D_EOK         : 函数执行成功
 *          ATK_MW8266D_ETIMEOUT    : 等待期望应答超时，函数执行失败
 */
uint8_t atk_mw8266d_send_at_cmd(char *cmd, char *ack, uint32_t timeout)
{
    uint8_t *ret = NULL;

    atk_mw8266d_uart_rx_restart();
    atk_mw8266d_uart_printf("%s\r\n", cmd);

    if ((ack == NULL) || (timeout == 0))
    {
        return ATK_MW8266D_EOK;
    }
    else
    {
        while (timeout > 0)
        {
            ret = atk_mw8266d_uart_rx_get_frame();
            if (ret != NULL)
            {
                if (strstr((const char *)ret, ack) != NULL)
                {
                    return ATK_MW8266D_EOK;
                }
                else
```

```
                {
                    atk_mw8266d_uart_rx_restart();
                }
            }
            timeout--;
            delay_ms(1);
        }

        return ATK_MW8266D_ETIMEOUT;
    }
}
```

从上面的代码中可以看出，函数 atk_mw8266d_send_at_cmd()函数会将待发送的 AT 指令加上换行符后通过 UART 发送至 ATK-MW8266D 模块，随后等待 ATK-MW8266D 模块的响应，并判断响应中是否包含期望等待的响应，如果有，则说明本次 AT 指令传输成功。

2.1.2.3 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;
    char     ip_buf[16];
    uint8_t  key;
    uint8_t  is_unvarnished = 0;

    /* 初始化 ATK-MW8266D */
    ret = atk_mw8266d_init(115200);
    if (ret != 0)
    {
        printf("ATK-MW8266D init failed!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }

    printf("Joining to AP...\r\n");
    ret  = atk_mw8266d_restore();                          /* 恢复出厂设置 */
    ret += atk_mw8266d_at_test();                          /* AT 测试 */
```

```
    ret += atk_mw8266d_set_mode(1);                             /* Station 模式 */
    ret += atk_mw8266d_sw_reset();                              /* 软件复位 */
    ret += atk_mw8266d_ate_config(0);                           /* 关闭回显功能 */
    ret += atk_mw8266d_join_ap(DEMO_WIFI_SSID, DEMO_WIFI_PWD);/* 连接 WIFI */
    ret += atk_mw8266d_get_ip(ip_buf);                          /* 获取 IP 地址 */
    if (ret != 0)
    {
        printf("Error to join ap!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }
    demo_show_ip(ip_buf);

    /* 连接 TCP 服务器 */
    ret = atk_mw8266d_connect_tcp_server(   DEMO_TCP_SERVER_IP,
                                            DEMO_TCP_SERVER_PORT);
    if (ret != 0)
    {
        printf("Error to connect tcp server!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }

    /* 重新开始接收新的一帧数据 */
    atk_mw8266d_uart_rx_restart();

    while (1)
    {
        key = key_scan(0);

        switch (key)
        {
            case KEY0_PRES:
            {
                /* 功能测试 */
                demo_key0_fun(is_unvarnished);
                break;
            }
```

```
            case KEY1_PRES:
            {
                /* 透传模式切换 */
                demo_key1_fun(&is_unvarnished);
                break;
            }
            default:
            {
                break;
            }
        }

        /* 发送透传接收自 TCP Server 的数据到串口调试助手 */
        demo_upload_data(is_unvarnished);

        delay_ms(10);
    }
}
```

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。但要注意的是，函数 demo_run()首先会配置 ATK-MW8266D 模块连接 WIFI 和 TCP 服务器，其中 WIFI 的名称和密码、TCP 服务器的地址和端口均在 demo.c 文件中通过宏的方式定义，具体的代码，如下所示：

```
#define DEMO_WIFI_SSID          "ALIENTEK-YF"
#define DEMO_WIFI_PWD           "15902020353"
#define DEMO_TCP_SERVER_IP      "192.168.1.3"
#define DEMO_TCP_SERVER_PORT    "8080"
```

在完成本实验的时候，需要根据具体的情况对这四个宏定义进行配置，才能顺利地完成本实验。

2.1.3 实验现象

将 ATK-MW8266D 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：

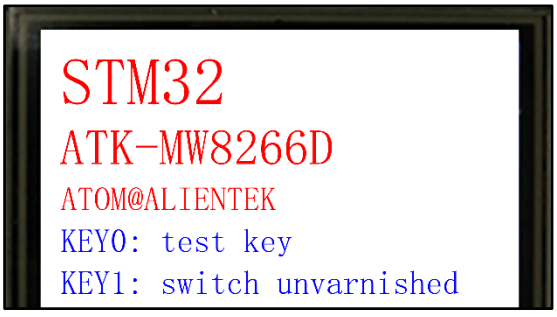

图 2.1.3.1 LCD 显示内容一

同时，通过串口调试助手输出实验信息，如下图所示：

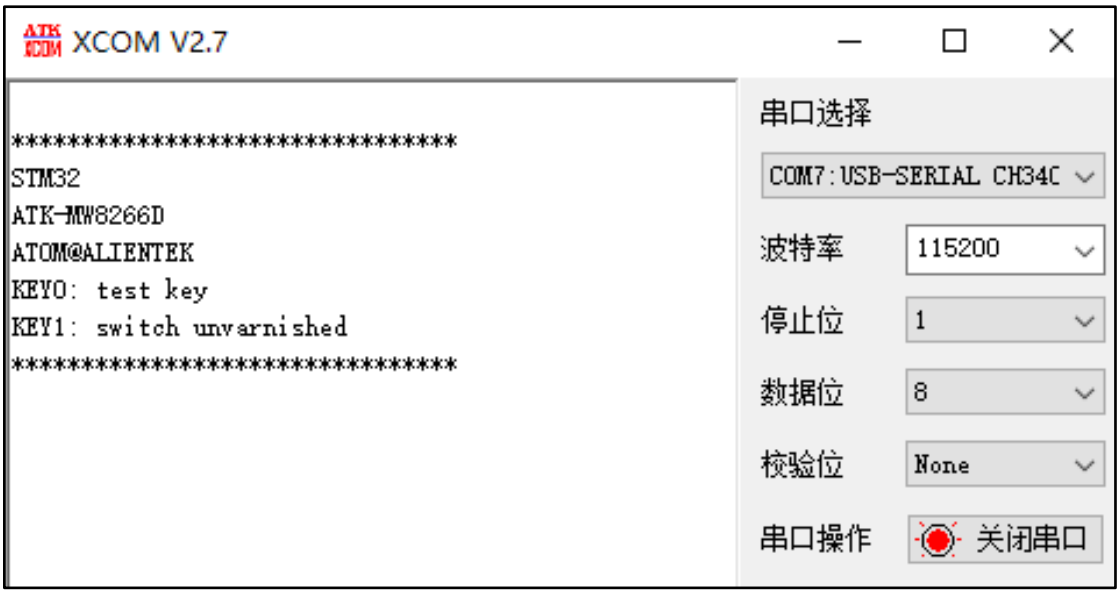

图 2.1.3.2 串口调试助手显示内容一

接下来，程序会初始化 ATK-MW8266D 模块并自动配置连接 WIFI 和 TCP 服务器，因此在此之前，需要先打开网络调试助手，设置好 TCP 服务器 IP 和款口号，并开启 TCP 服务器连接，否则 ATK-MW8266D 模块将无法成功连接 TCP 服务器，网络调试助手的设置界面，如下图所示：

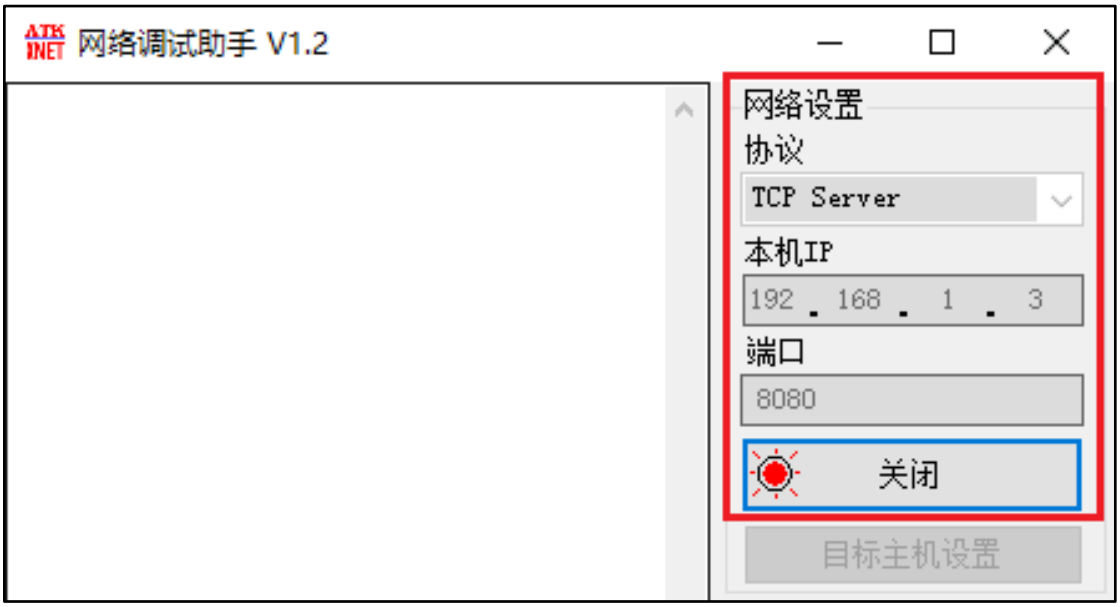

图 2.1.3.3 网络调试助手设置

如果 ATK-MW8266D 模块成功连接上 WIFI 和 TCP 服务器，便会在 LCD 和串口调试助手上显示 ATK-MW8266D 的 IP 地址，方便网络调试助手确认 TCP 客户端的 IP 地址，LCD 和串口调试助手显示的 IP 地址，如下图所示：

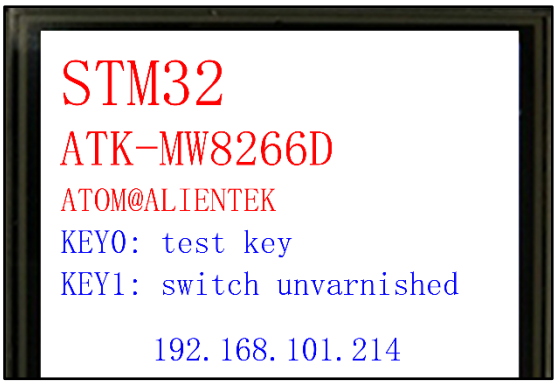

图 2.1.3.4 LCD 显示内容二

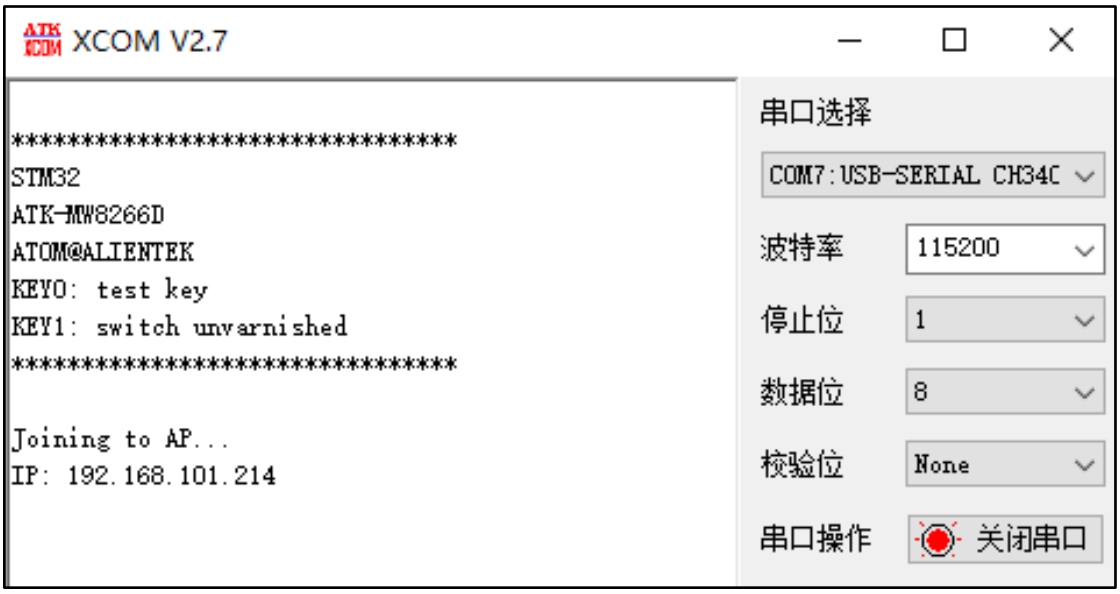

图 2.1.3.5 串口调试助手显示内容二

当 ATK-MW8266D 成功连接 WIFI 和 TCP 服务器后，就能够通过开发板上的按键来进行本实验的操作了。

在默认没有进入透传的模式下，按下按键 0，会对 ATK-MW8266D 进行 AT 指令测试，测试的结果会通过串口调试助手输出，如下图所示：

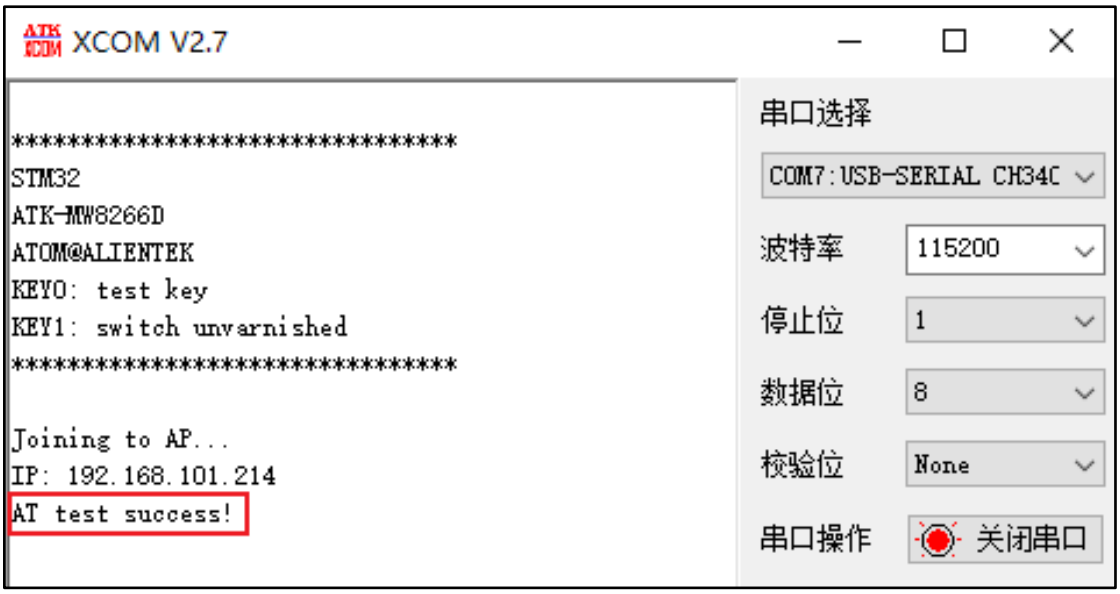

图 2.1.3.6 串口调试助手显示内容三

如果 AT 指令测试成功，那么说明开发板与 ATK-MW8266D 模块的通讯正常，那么接下来可以按下按键 1，配置 ATK-MW8266D 进入透传模式，在透传模式下，开发板上主控芯片通过 UART 发送给 ATK-MW8266D 模块的数据，会原封不动地通过网络发送给 TCP 服务器，同样的，TCP 服务器通过网络发送给 ATK-MW8266D 模块的数据，也会原封不动地通过 UART 发送给开发板上的主控芯片。

接下来按下按键 0，发送数据给 ATK-MW8266D 模块，此时，便可以在网络调试助手上看到，ATK-MW8266D 模块发送过来的消息，如下图所示：

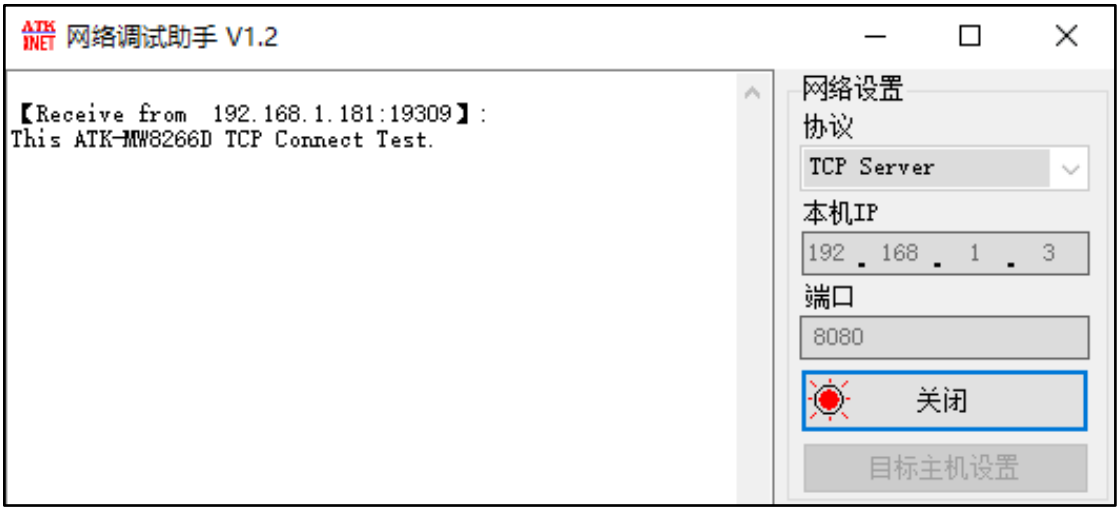

图 2.1.3.7 网络调试助手显示内容

这时通过网络助手发送消息给 ATK-MW8266D 模块，那么 ATK-MW8266D 模块就会将接收到的数据通过 UART 发送给开发板上的主控芯片，为了观察实验现象，本实验将接收到的信息打印在串口调试助手上，如下图所示：

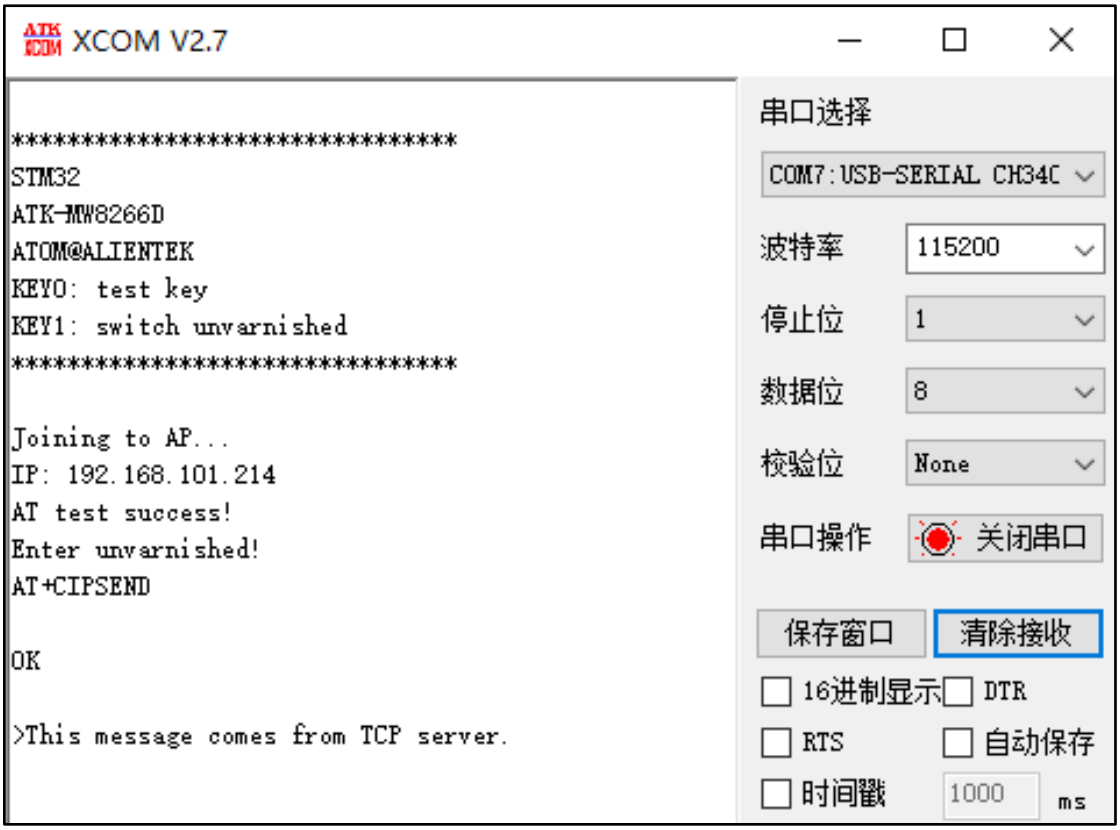

图 2.1.3.8 串口调试助手显示内容四

2.2 ATK-MW8266D 模块原子云连接实验

2.2.1 功能说明

在本实验中，开发板主控芯片通过串口与 ATK-MW8266D 模块进行通讯，并在上电后自动根据配置信息，配置 ATK-MW8266D 模块连接 WIFI，成功连接 WIFI 后，将在串口调试助手和 LCD 上显示 ATK-MW8266D 模块的 IP 地址，随后便可通过按键对 ATK-MW8266D 模块进行 AT 指令测试和连接原子云服务器（原子云服务器的连接方式，请见《ATK-MW8266D 模块用户手册》，在本实验中不再进行过多的描述），AT 指令测试的测试结果将显示在串口调试助手上，当模块成功连接原子云后，可通过按键发送数据至原子云服务器，同时也可接收来自原子云服务器的数据，并实时显示在串口调试助手上。

2.2.2 源码解读

2.2.2.1 ATK-MW8266D 模块接口驱动

本实验中，ATK-MW8266D 模块接口的驱动代码与 2.1 小节中“ATK-MW8266D 模块 TCP 透传实验”中的接口驱动代码一致，请见第 2.1.2.1 小节“ATK-MW8266D 模块接口驱动”。

2.2.2.2 ATK-MW8266D 模块驱动

本实验中，ATK-MW8266D 模块的驱动代码与 2.1 小节中“ATK-MW8266D 模块 TCP 透传实验”中的驱动代码一致，请见第 2.1.2.2 小节“ATK-MW8266D 模块驱动”。

2.2.2.3 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;
    char     ip_buf[16];
    uint8_t  key;
    uint8_t  is_atkcld = 0;

    /* 初始化 ATK-MW8266D */
    ret = atk_mw8266d_init(115200);
    if (ret != 0)
    {
        printf("ATK-MW8266D init failed!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }

    printf("Joining to AP...\r\n");
    ret  = atk_mw8266d_restore();                                   /* 恢复出厂设置 */
    ret += atk_mw8266d_at_test();                                   /* AT 测试 */
    ret += atk_mw8266d_set_mode(1);                                 /* Station 模式 */
    ret += atk_mw8266d_sw_reset();                                  /* 软件复位 */
    ret += atk_mw8266d_ate_config(0);                               /* 关闭回显功能 */
    ret += atk_mw8266d_join_ap(DEMO_WIFI_SSID, DEMO_WIFI_PWD);/* 连接 WIFI */
    ret += atk_mw8266d_get_ip(ip_buf);                              /* 获取 IP 地址 */
    if (ret != 0)
    {
        printf("Error to join ap!\r\n");
        while (1)
        {
            LED0_TOGGLE();
            delay_ms(200);
        }
    }
    demo_show_ip(ip_buf);
```

```
    /* 重新开始接收新的一帧数据 */
    atk_mw8266d_uart_rx_restart();

    while (1)
    {
        key = key_scan(0);

        switch (key)
        {
            case KEY0_PRES:
            {
                /* 功能测试 */
                demo_key0_fun(is_atkcld);
                break;
            }
            case KEY1_PRES:
            {
                /* 切换原子云连接状态 */
                demo_key1_fun(&is_atkcld);
                break;
            }
            default:
            {
                break;
            }
        }

        /* 发送接收自原子云的数据到串口调试助手 */
        demo_upload_data(is_atkcld);

        delay_ms(10);
    }
}
```

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。但要主要的是，函数 demo_run()首先会配置 ATK-MW8266D 模块连接 WIFI 和原子云服务器，其中 WIFI 的名称和密码、原子云服务器设备的 ID 和密码均在 demo.c 文件中通过宏的方式定义，具体的代码，如下所示：

```
#define DEMO_WIFI_SSID      "ALIENTEK-YF"
#define DEMO_WIFI_PWD       "15902020353"
#define DEMO_ATKCLD_DEV_ID  "86112558784372656615"
#define DEMO_ATKCLD_DEV_PWD "12345678"
```

在完成本实验的时候，需要根据具体的情况对这四个宏定义进行配置，才能顺利地完成本实验。

2.2.3 实验现象

将 ATK-MW8266D 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：

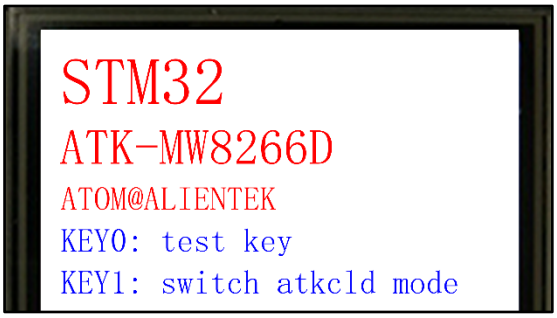

图 2.2.3.1 LCD 显示内容一

同时，通过串口调试助手输出实验信息，如下图所示：

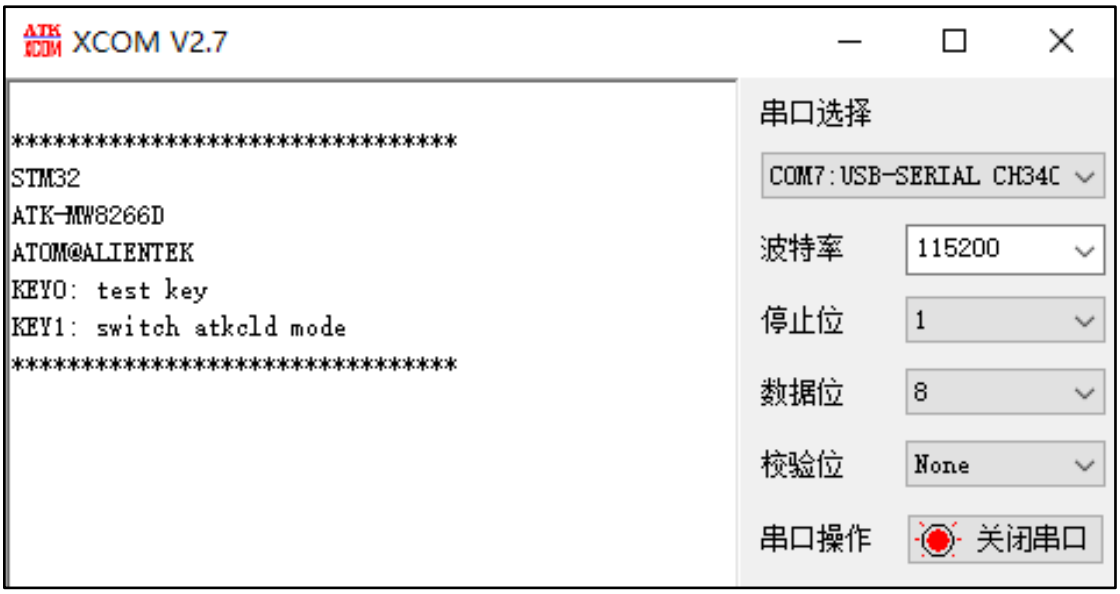

图 2.2.3.2 串口调试助手显示内容一

接下来，程序会初始化 ATK-MW8266D 模块并自动配置连接 WIFI，如果 ATK-MW8266D 模块成功连接上 WIFI，便会在 LCD 和串口调试助手上显示 ATK-MW8266D 的 IP 地址，LCD 和串口调试助手显示的 IP 地址，如下图所示：

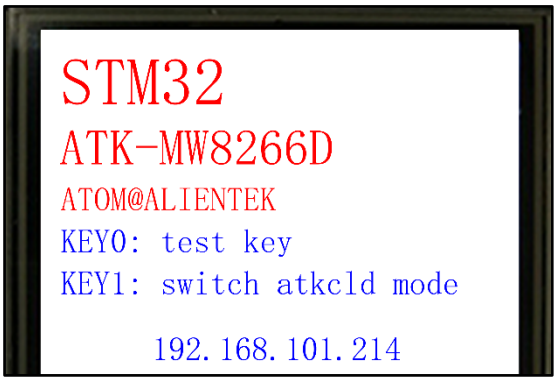

图 2.2.3.3 LCD 显示内容二

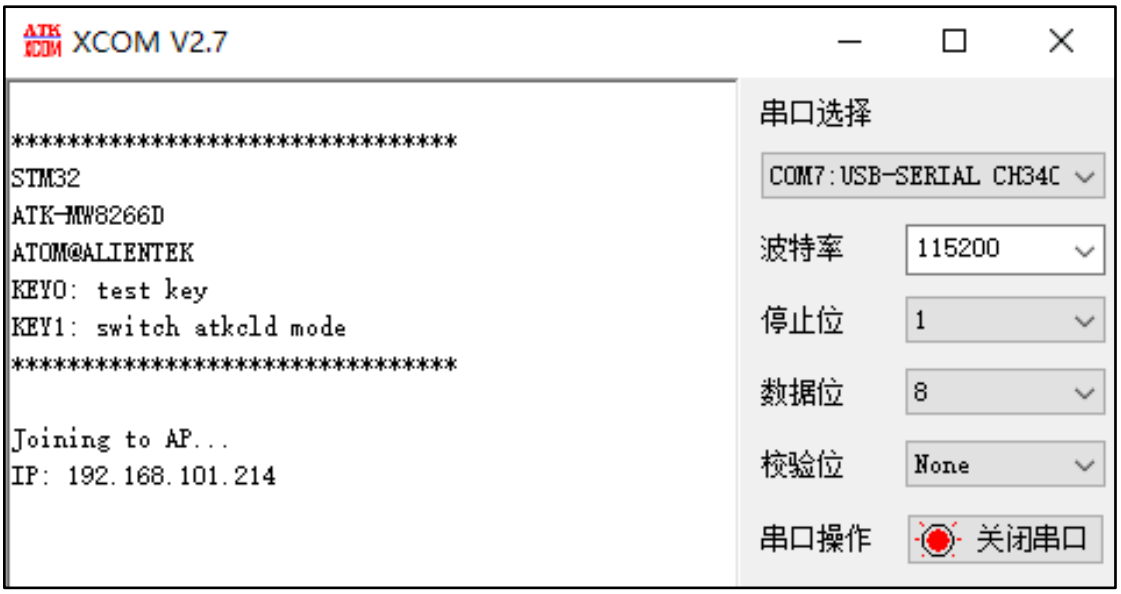

图 2.2.3.4 串口调试助手显示内容二

当 ATK-MW8266D 成功连接 WIFI 后，就能通过开发板上的按键来进行本实验的操作了。

在默认没有连接原子云的模式下，按下按键 0，会对 ATK-MW8266D 进行 AT 指令测试，测试的结果会通过串口调试助手输出，如下图所示：

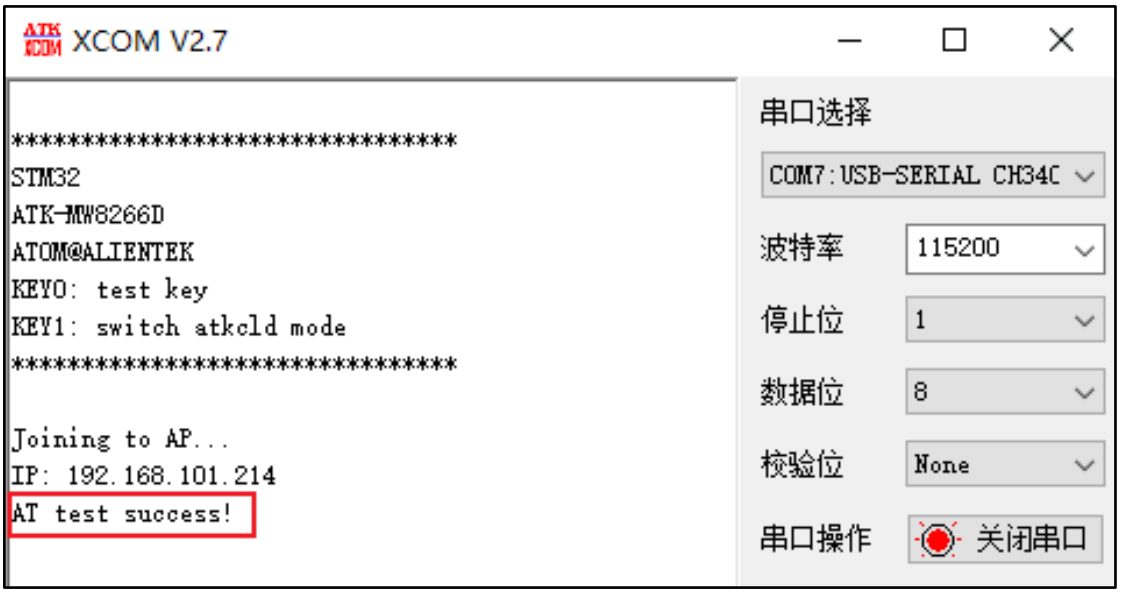

图 2.2.3.5 串口调试助手显示内容三

如果 AT 指令测试成功，那么说明开发板与 ATK-MW8266D 模块的通讯正常，那么接下来可以按下按键 1，配置 ATK-MW8266D 连接原子云，因为要连接原子云服务器，因此在此之前，需要在原子云上创建好设备，具体的创建过程，请见《ATK-MW8266D 模块使用说明》，否则 ATK-MW8266D 模块将无法成功连接原子云服务器，成功连接原子云服务器后，就能够在原子云的网页中开到设备的状态为“已连接”，如下图所示：

图 2.2.3.6 原子云网页设备状态

接下来按下按键 0，发送数据给 ATK-MW8266D 模块，此时，便可以在原子云网页中的“消息收发”界面中查看到 ATK-MW8266D 模块发送过来的消息，如下图所示：

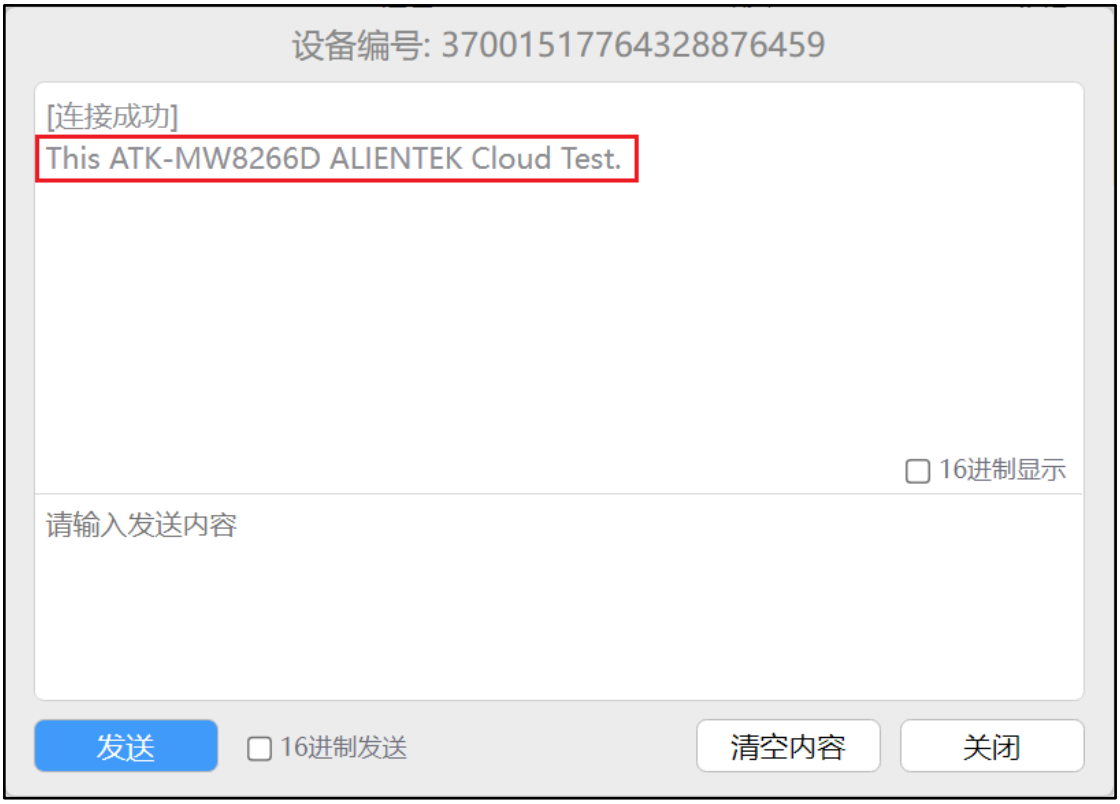

图 2.2.3.7 原子云网页设备“消息收发”界面

这时通过原子云网页设备的“消息收发”界面发送消息给 ATK-MW8266D 模块，那么 ATK-MW8266D 模块就会将接收到的数据通过 UART 发送给开发板上的主控芯片，为了观察实验现象，本实验将接收到的信息打印在串口调试助手上，如下图所示：

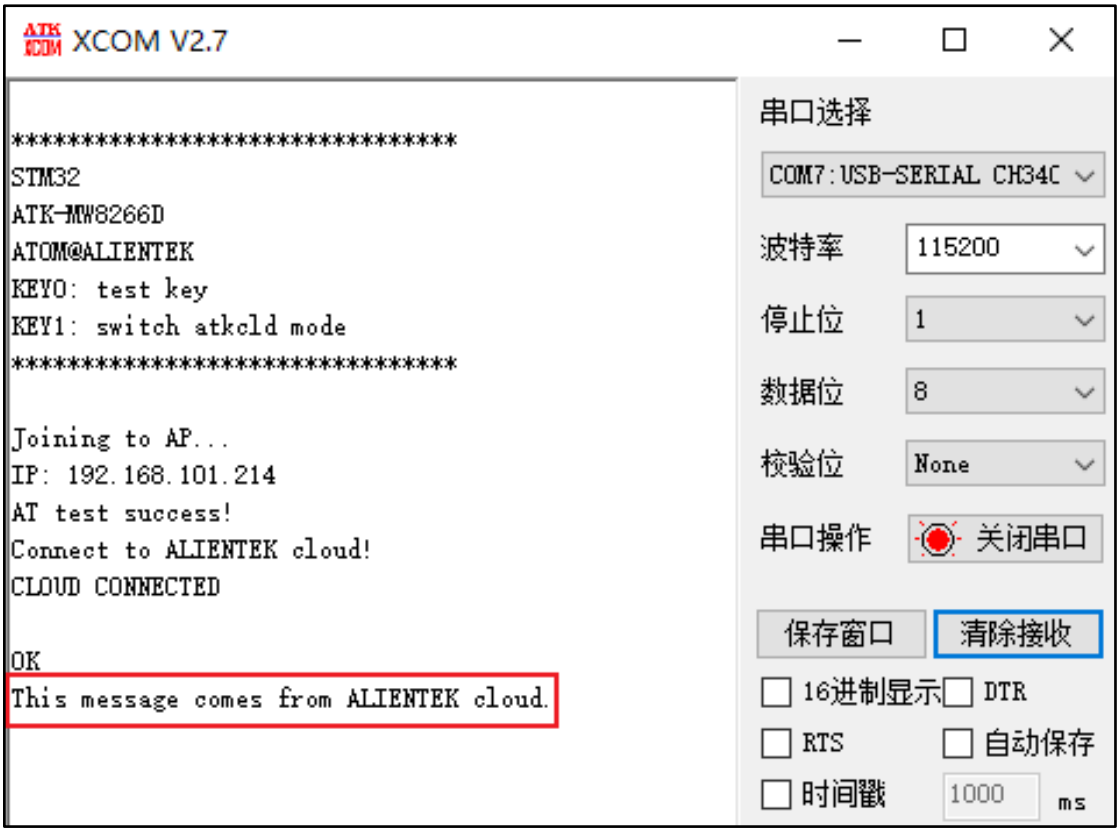

图 2.2.3.8 串口调试助手显示内容四

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/iot/atk-esp.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

正点原子