

AT32WB415 SPP Application Note

前言

这篇应用笔记描述了如何通过AT32WB415的蓝牙模块来实现SPP协议，可以通过SPP协议来与主机进行通讯。

支持型号列表：

支持型号	AT32WB415
------	-----------

目录

1	概述	5
2	AT32WB415 SPP 程序设计	6
	2.1 BLE 模块软件设计	6
	2.2 MCU 软件设计	7
3	SPP 案例使用	8
	3.1 硬件资源.....	8
	3.2 软件资源.....	8
	3.3 SPP demo 使用	8
4	版本历史	16

表目录

表 1. 文档版本历史 16

图目录

图 1. SPP	5
图 2. 点击添加 BLE 烧录文件	9
图 3. 修改 BLE 模块下载起始地址	9
图 4. 电脑设置	10
图 5. 添加蓝牙设备	10
图 6. 连接配对 AT32WB415_SPP	11
图 7. 连接配对成功	11
图 8. 蓝牙虚拟串口测试	12
图 9. 开发板端通过串口收发数据	12
图 10. 连接 AT32WB415_SPP	13
图 11. 搜索蓝牙设备	13
图 12. 连接 AT32WB415_SPP	14
图 13. 允许 AT32WB415_SPP 配对	14
图 14. 上位机与 AT32WB415_SPP 进行数据收发测试	15
图 15. 开发板端通过串口收发数据	15

1 概述

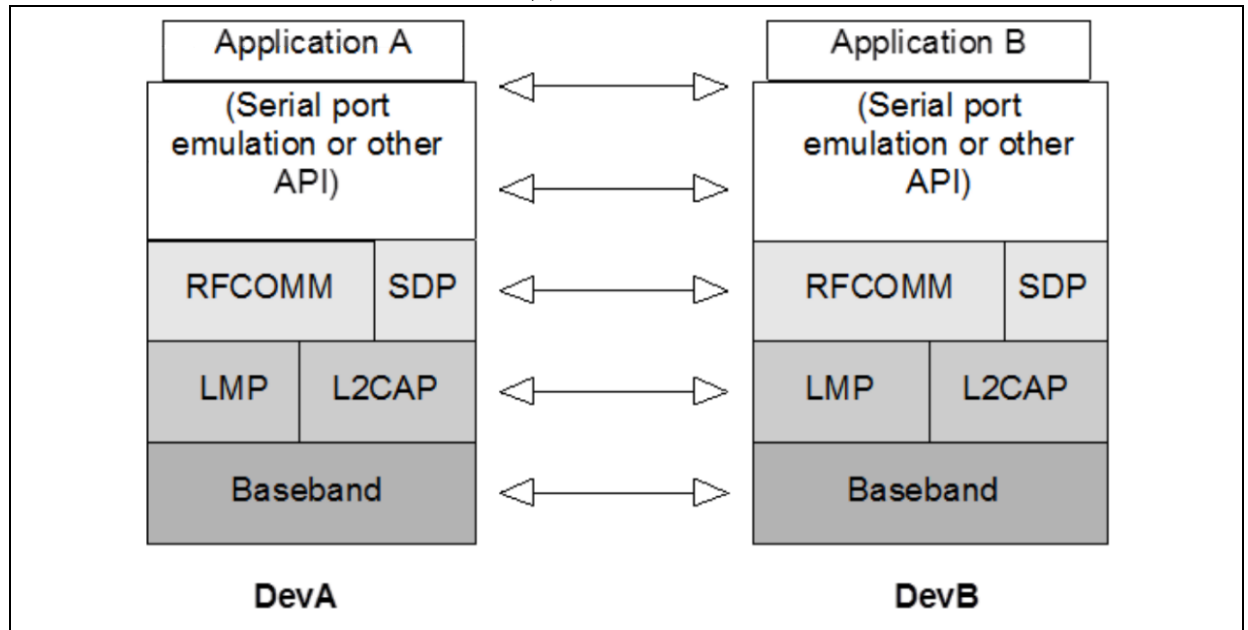
SPP (Serial Port Profile) 串口协议，是一个传输协议子集，它规定了在两个对等设备之间建立 RFCOMM 通信所需要的基本操作。简单来说就是在蓝牙设备之间建立虚拟的串口进行数据通信。串口应用基于通用访问应用 GAP 和服务发现应用 SDAP 应用模型之上，又是其他许多应用框架的基础。SPP 可以直接映射到 RFCOMM 协议中，当两个设备间需要通过 RFCOMM 仿真的串口进行通信时，SPP 将对其中必备的服务和过程做出详细的规定。

SPP 协议中包含两个设备：

- 设备 A：这个设备会发起一个连接另外设备的请求
- 设备 B：这个设备等待另外一个设备发起连接请求

SPP 在整个蓝牙中的架构图如下：

图 1. SPP



2 AT32WB415 SPP 程序设计

2.1 BLE 模块软件设计

对于 BLE 模块部分的软件，BLE 模块的功能就是通过蓝牙来发送和接收上位机软件的数据，主要分为以下几个部分：

1) 系统初始化，含协议栈、GPIO、串口等初始化内容，初始化函数如下：

```
void SYS_Initialise(void)
{
    u_int32 flags;
    SYSirq_Disable_Interrupts_Save_Flags(&flags);
    icu_init();
    flash_init();
    GPIO_Initial();

    UART_Initial();

#ifdef UART2_LOG
    uart2_init(256000);
#endif

    xver_initial();
    bt_clk_init();
    Timer_Initial();
    rw_ram_init();
    read_env_config();

    SYSconfig_Initialise();
    TRAhcit_Set_Register_Function(TRAhcit_Register_UART_Driver);

    bprintf("x5\r\n");
    BT_Initialise();
    REG_AHB0_ICU_INT_STATUS_FLAG = 0xffffffff;
    bprintf("start\r\n");
    SYSirq_Enable_Interrupts_Save_Flags(&flags);
    Enable_ICU_Intr();
}
```

2) SPP 协议配置，配置函数如下：

```
void SYS_Host_Initialise(void)
{
    memset((void*)&btj_adapter, 0x00, sizeof(DEVICEBTJ_ADAPTER_T));
    send_HCI_command(Com_Table_Reset, 4);
    {
        memset((void*)&spp_service, 0x00, sizeof(DEVICEBTJ_SERVICE_T));
        DeviceSdpdb_Doreg(&btj_adapter, &spp_service, DEVICEBTJ_OBJECT_SPP);
        spp_service.adapter = &btj_adapter;
        spp_service.rfc_context.rfc_mtu = RFC_FRAME_SIZE;
        spp_service.ctx_state = SERVICE_STATE_IDLE;
        btj_adapter.app_service_header = &spp_service;
    }
}
```

```

}
auto_connect_cnt = 0;
}

```

3) 等待 IO 拉低后上位机连接 BLE 模块，完成配对和连接，

```

void driver_mouse_key_scan_paired(void)
{
    if((((REG_APB5_GPIO0_DATA >> 8) & 0x80) == 0)) //PAIR_DETECT
    {
        if((CKECK_PAIR_scan_START) == 0)
        {
            CKECK_PAIR_scan_START = 1;
            PAIR_scan_START = 1;
        }
    }
    else
    {
        CKECK_PAIR_scan_START = 0;
        PAIR_scan_START = 0;
    }
}

```

4) UART 串口收到 MCU 数据后通过 SPP 发送到上位机软件，关键代码如下：

```

if((sys_flag & FLAG_UART_INTERRUPT) && (sys_flag & FLAG_CONNECTED))
{
    sys_flag &= ~FLAG_UART_INTERRUPT;
    bprintf("buff length: %d\r\n", uart_rx_buff_header);
    spp_data_send((uint8 *)uart_rx_tx_buff, uart_rx_buff_header);
    uart_rx_buff_header = 0;
    sys_flag |= FLAG_KEY_ACTIVE;
}

```

5) 蓝牙收到 SPP 接收数据时通过 UART 将上位机数据发送到 MCU 端，关键代码如下

```

case DEVICEBTJ_EVENT_SPP_RECEIVE:
    bprintf("Length:%d\r\n", MsgLength);
    bprintf("M1:%x\r\n", MsgBuffer[0]);
    bprintf("M2:%x\r\n", MsgBuffer[1]);
    bprintf("M3:%x\r\n", MsgBuffer[2]);
    uart_data_send(MsgBuffer, MsgLength);
    sys_flag |= FLAG_KEY_ACTIVE;
    enter_sniff_cnt = 10000;
    break;

```

2.2 MCU 软件设计

上面提到 BLE 端接收蓝牙数据后会通过串口将数据发送到 MCU 端，MCU 端对接收到的数据处理方式如下：

- 1) 通过串口 USART3 接收 BLE 模块发送的蓝牙数据，并通过串口 USART2 将数据发送到 PC 串口工具软件进行显示
- 2) 通过串口 USART2 接收 PC 串口工具发送的数据，将数据通过串口 USART3 发送给 BLE 模块

3 SPP 案例使用

本案例将展示如何使用 AT32WB415 通过 SPP 协议来与主机进行通讯。

3.1 硬件资源

- 1) AT-START-WB415实验板
- 2) 一台PC电脑或含SPP测试软件智能手机

3.2 软件资源

- 1) ICP Programmer Tool, 可在官网下载最新版本
- 2) BlueSPP Android APP, 安装包路径位于: `Software\BlueSPP.apk`
- 3) BLE模块源码
 - `wb415_ble_spp`, BLE源程序, 用于BLE模块app程序, 工程路径位于:
`AN0156_AT32WB415_SPP_Application_Note_V2.0.0\SourceCode\wb415_ble_spp_2.0.0\wb415_ble_spp`
- 4) MCU端源码
 - `wb415_mcu_project`, MCU源程序, 用于BLE数据处理, 工程路径位于:
`AN0156_AT32WB415_SPP_Application_Note_V2.0.0\SourceCode\wb415_mcu_spp_2.0.0\utilities\wb415_mcu_project\mdk_v5`

注: 本案例仅提供Keil v5版本工程, 若用户需要在其他编译环境上使用, 请参考

`AT32WB415_Firmware_Library_V2.x.x\project\at_start_wb415\templates`中各种编译环境(例如IAR6/7/8, keil 4/5, `eclipse_gcc`)进行对应修改即可。

3.3 SPP demo 使用

在使用 SPP demo 前需要在 AT32WB415 中烧录支持 SPP 协议的源码, 步骤如下:

- 1) 打开BLE端的`wb415_ble_spp`源程序
- 2) 编译后会生成一系列bin文件, 位于`SourceCode\wb415_ble_spp_2.0.0\output`路径, 其中`wb415_ble_spp_merge.bin`, 融合了底层蓝牙协议栈和上层应用程序
- 3) 打开ICP工具下载上述路径中的`wb415_ble_spp_merge.bin`至BLE模块, 如下图

图 2. 点击添加 BLE 烧录文件

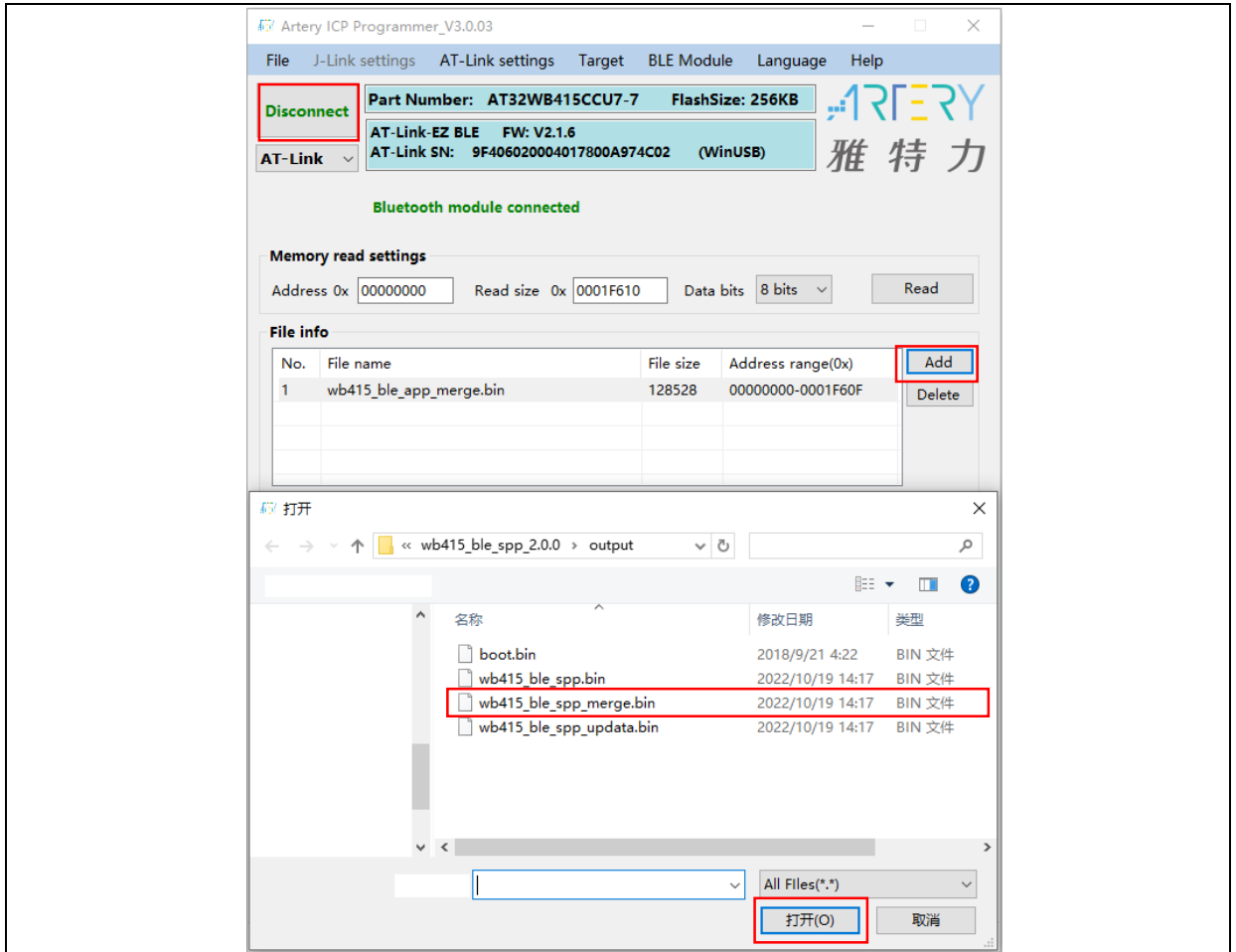
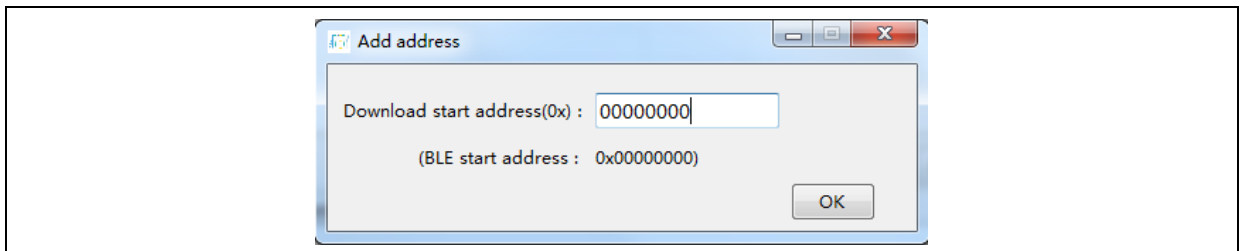


图 3. 修改 BLE 模块下载起始地址



MCU 端工程烧录步骤:

- 1) 打开MCU端的wb415_mcu_project源程序
- 2) 通过Keil v5编译后直接下载即可

BLE模块和MCU端代码都下载完成后即可开始进行SPP demo测试，PC电脑测试步骤如下：

- 1) 将引脚P07接地，此引脚拉低后会触发配对；
- 2) 打开电脑设置->设备->蓝牙和其他设备，点击添加蓝牙或其他设备（如图4）；
- 3) 进入添加设备页面，点击蓝牙（如图5）；
- 4) 点击AT32WB415_SPP设备进行添加（如图6），成功后会显示设备准备就绪（如图7）；
- 5) 在成功连接AT32WB415_SPP设备的PC上打开串口工具，找到蓝牙虚拟串口（串口号每台设备不同，示例中是COM23），进行虚拟串口收发测试（如图8）；

- 6) MCU端通过开发板上的AT-LINK-EZ使用串口USART2_TX (PA2), USART2_RX (PA3) 进行数据收发测试。

图 4. 电脑设置

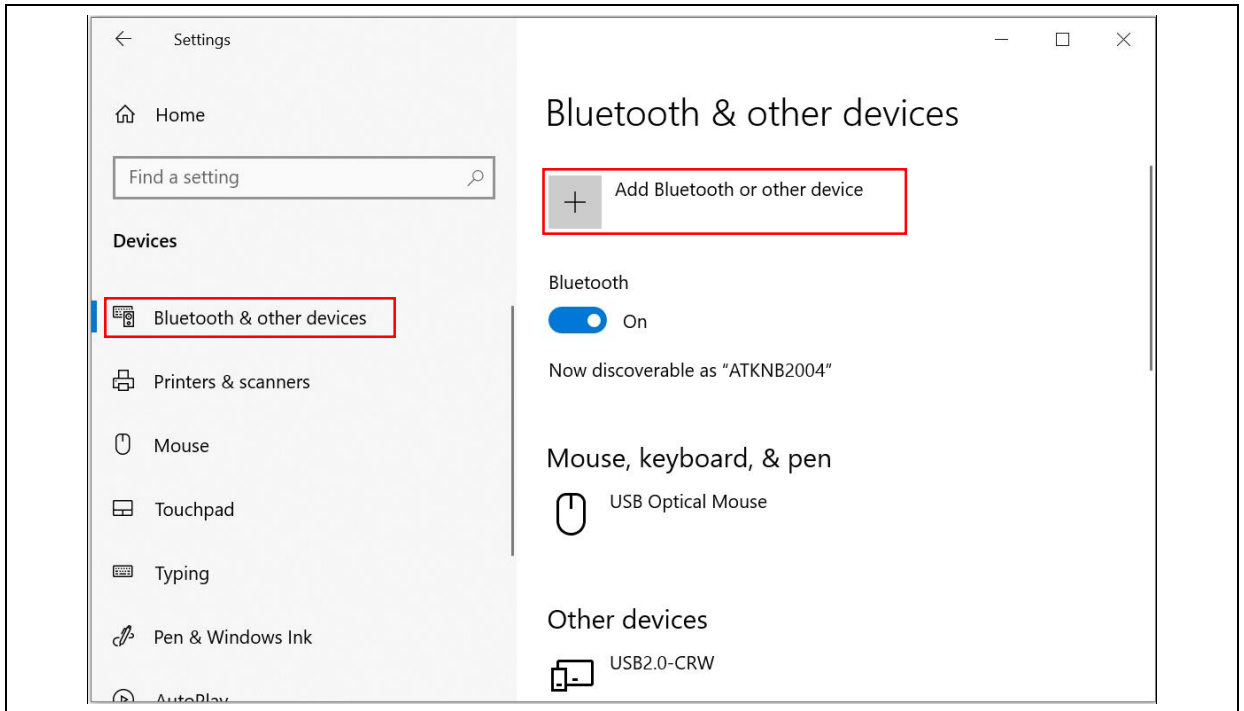


图 5. 添加蓝牙设备

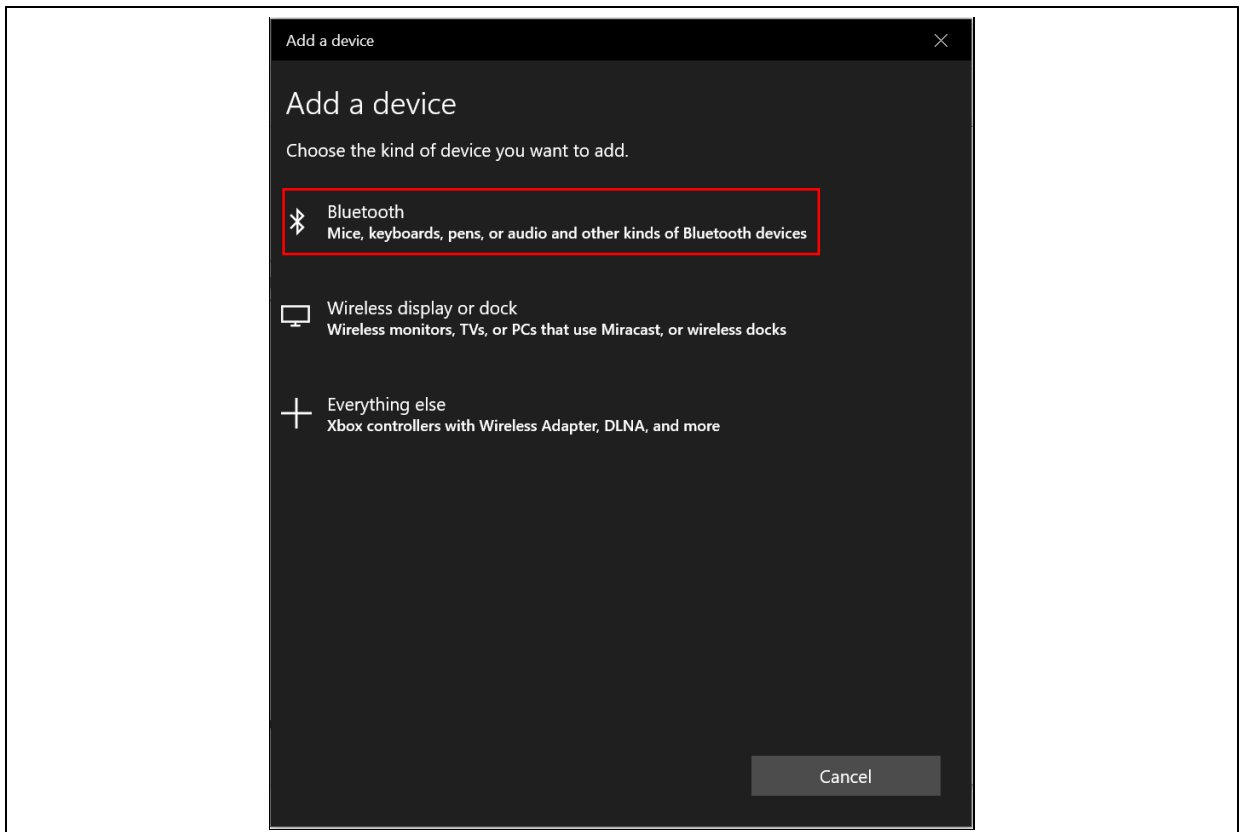


图 6. 连接配对 AT32WB415_SPP

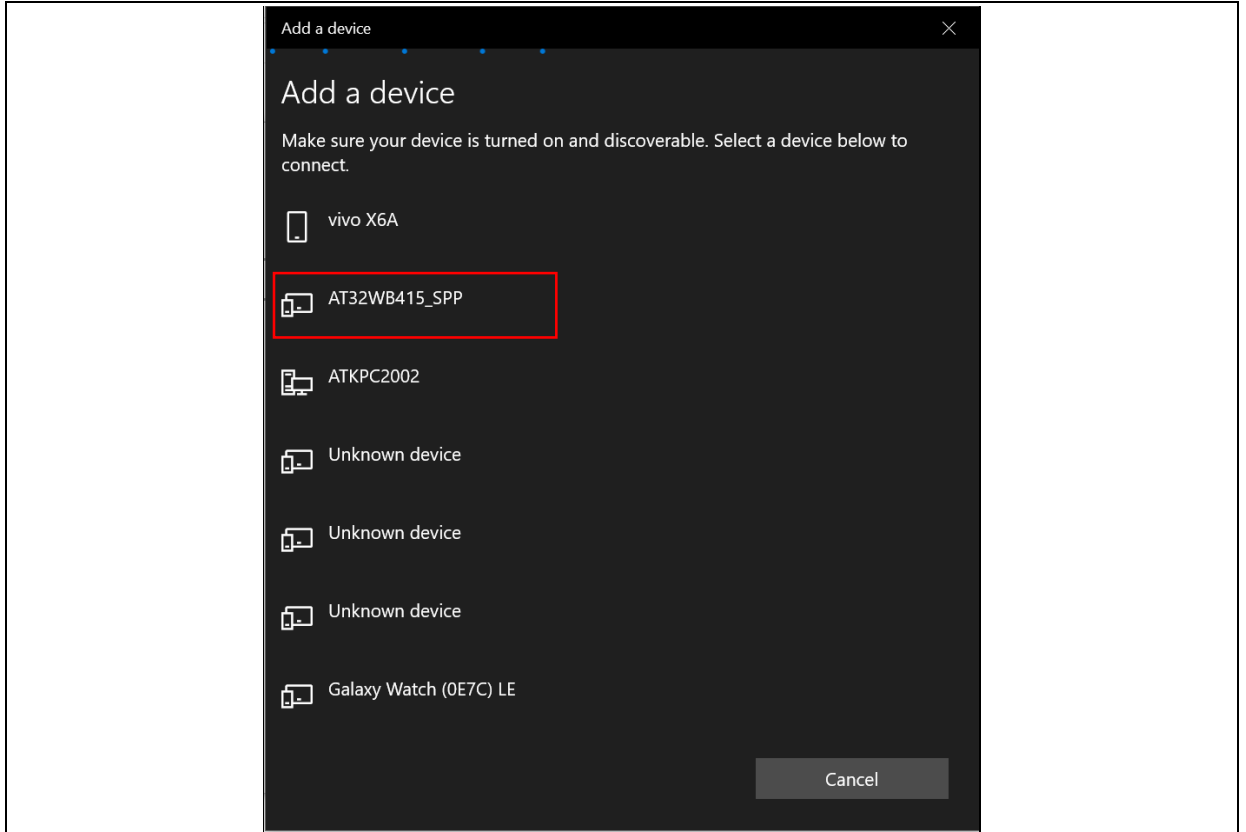


图 7. 连接配对成功

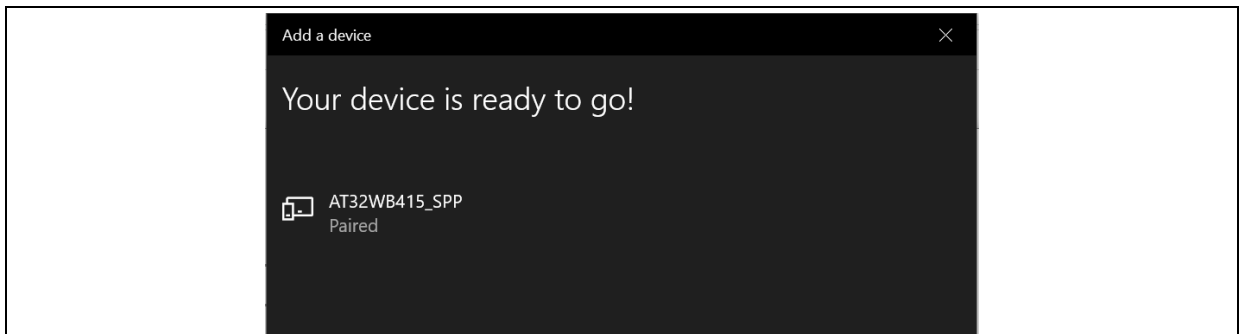


图 8. 蓝牙虚拟串口测试

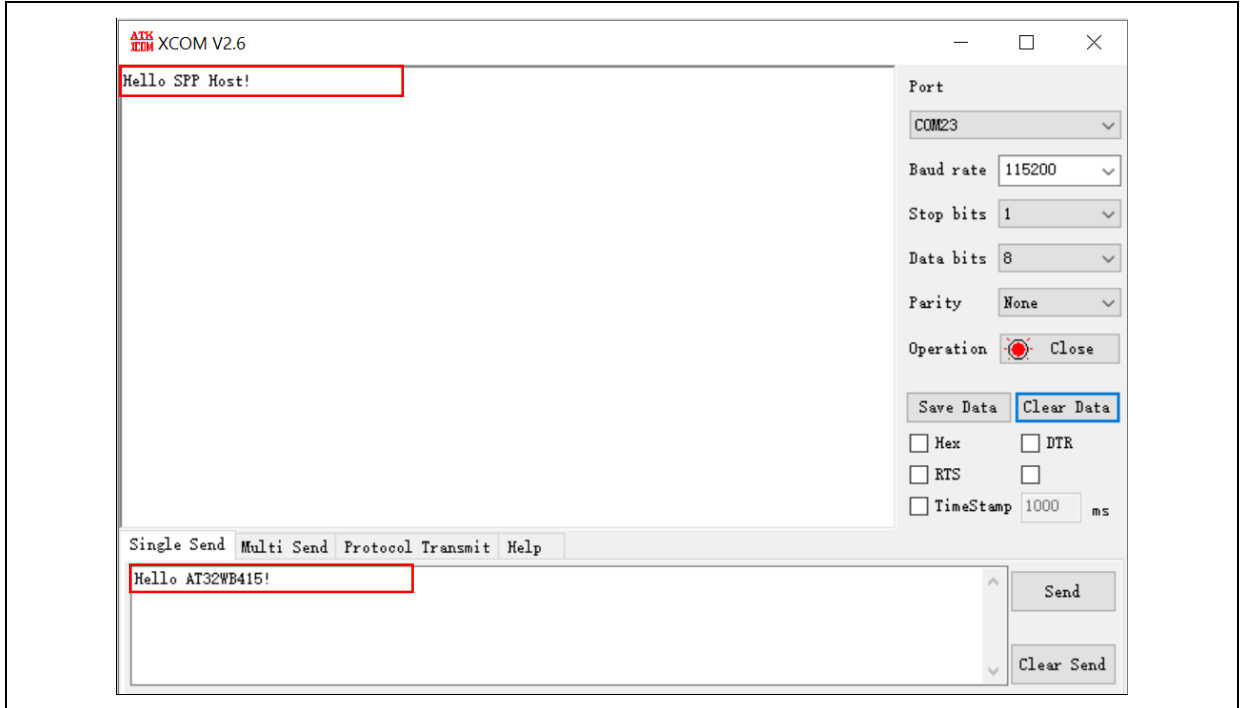
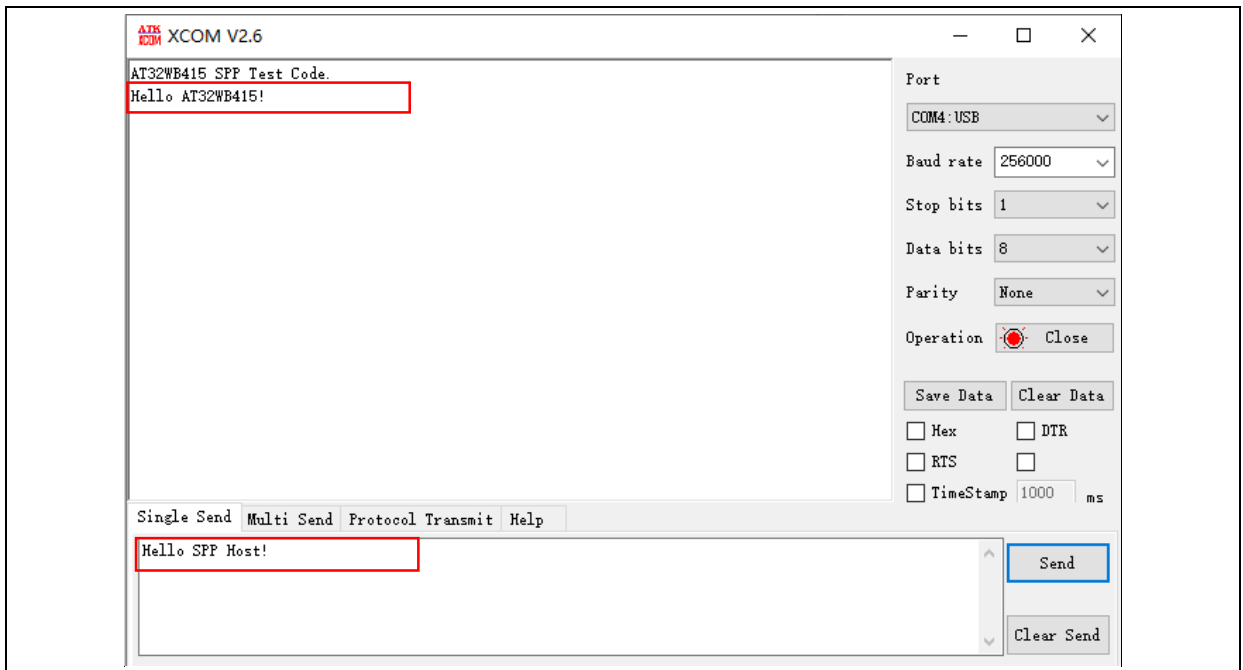


图 9. 开发板端通过串口收发数据



除了上述PC电脑测试方法外，还可以通过手机SPP上位机软件进行测试，步骤如下：

- 1) 将引脚P07接地，此引脚拉低后会触发配对；
- 2) 打开SPP测试APP并点击“CONNECT”按钮（如图10），点击搜索图标（如图11）；
- 3) 点击名称为AT32WB415_SPP的设备进行配对（如图12、13）；
- 4) 配对成功后即可通过SPP上位机通讯软件与AT32WB415_SPP进行数据收发测试（如图14）；
- 5) 开发板端则通过USART2_TX（PA2），USART2_RX（PA3）进行数据收发（如图15）。

注：AT-START-WB415实验板已将USART2串口连接到AT-LINK-EZ，可以直接进行使用，无需USB转串口工具。

图 10. 连接 AT32WB415_SPP

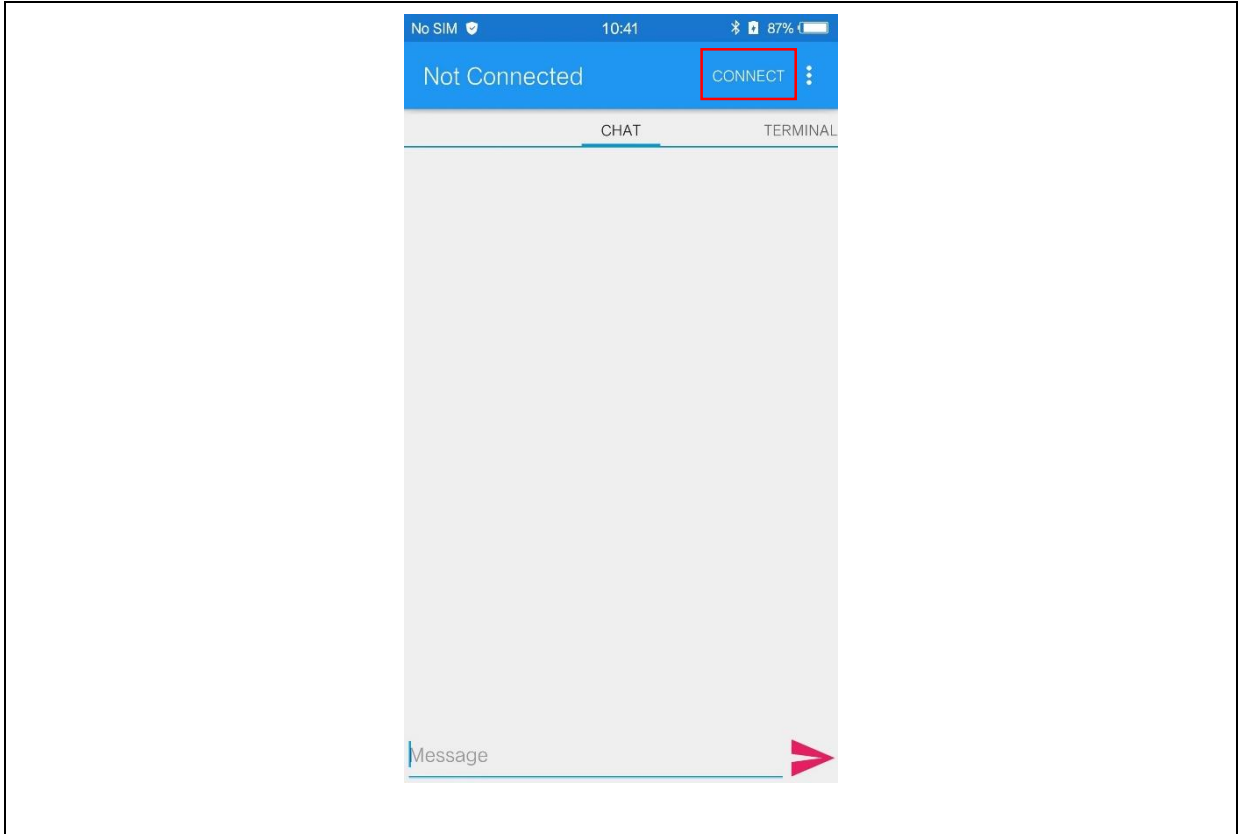


图 11. 搜索蓝牙设备

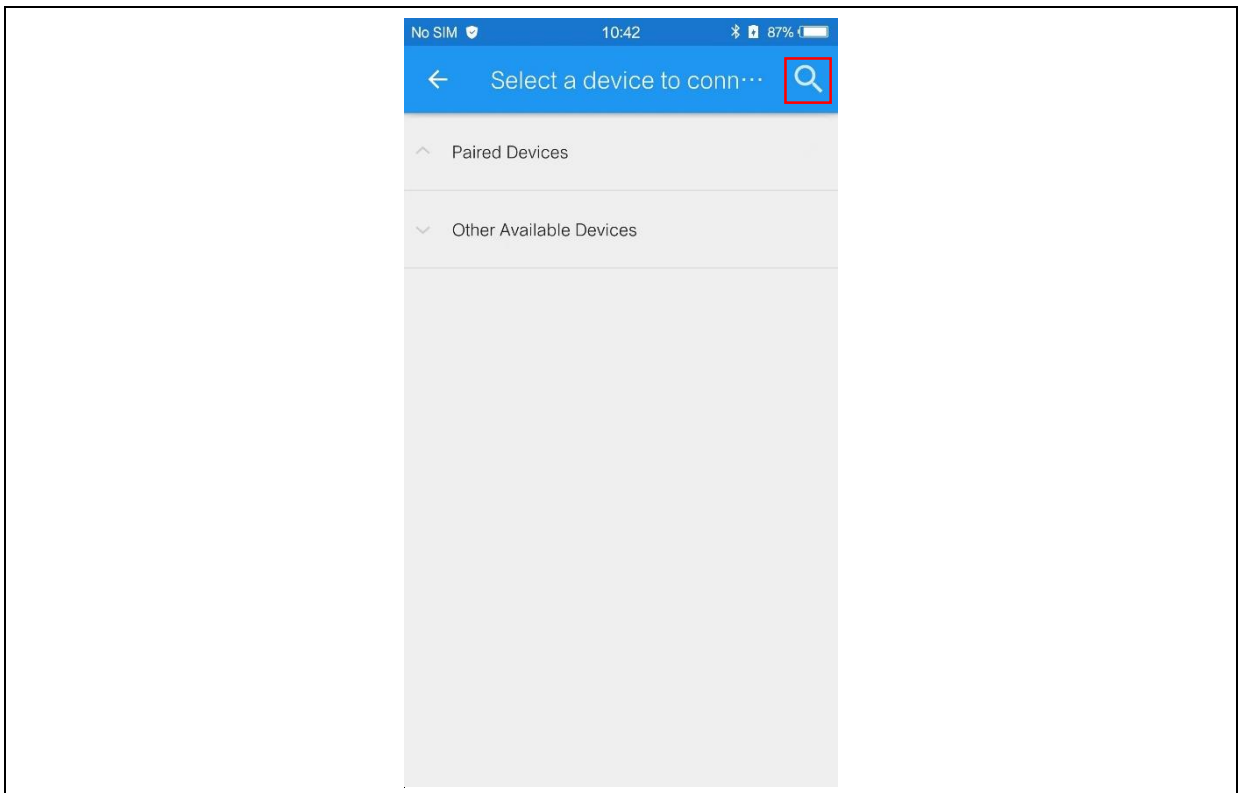


图 12.连接 AT32WB415_SPP

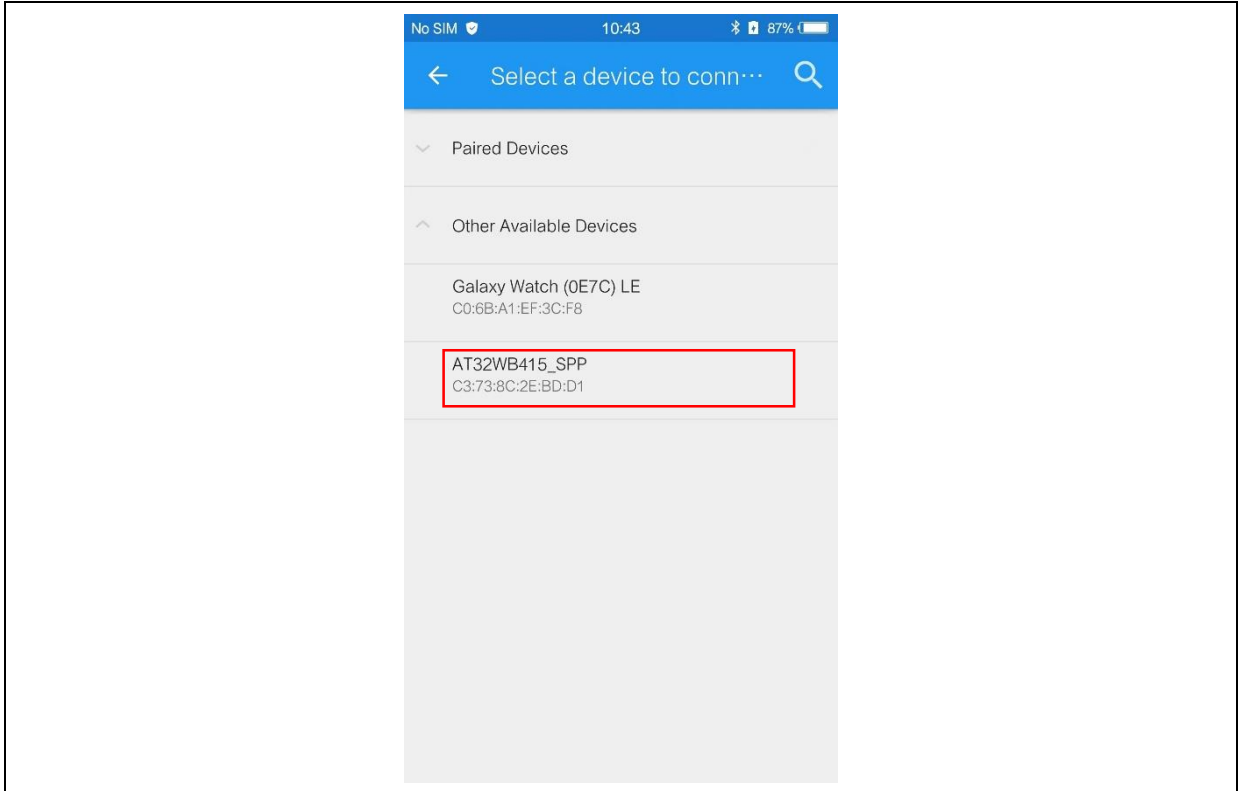


图 13.允许 AT32WB415_SPP 配对

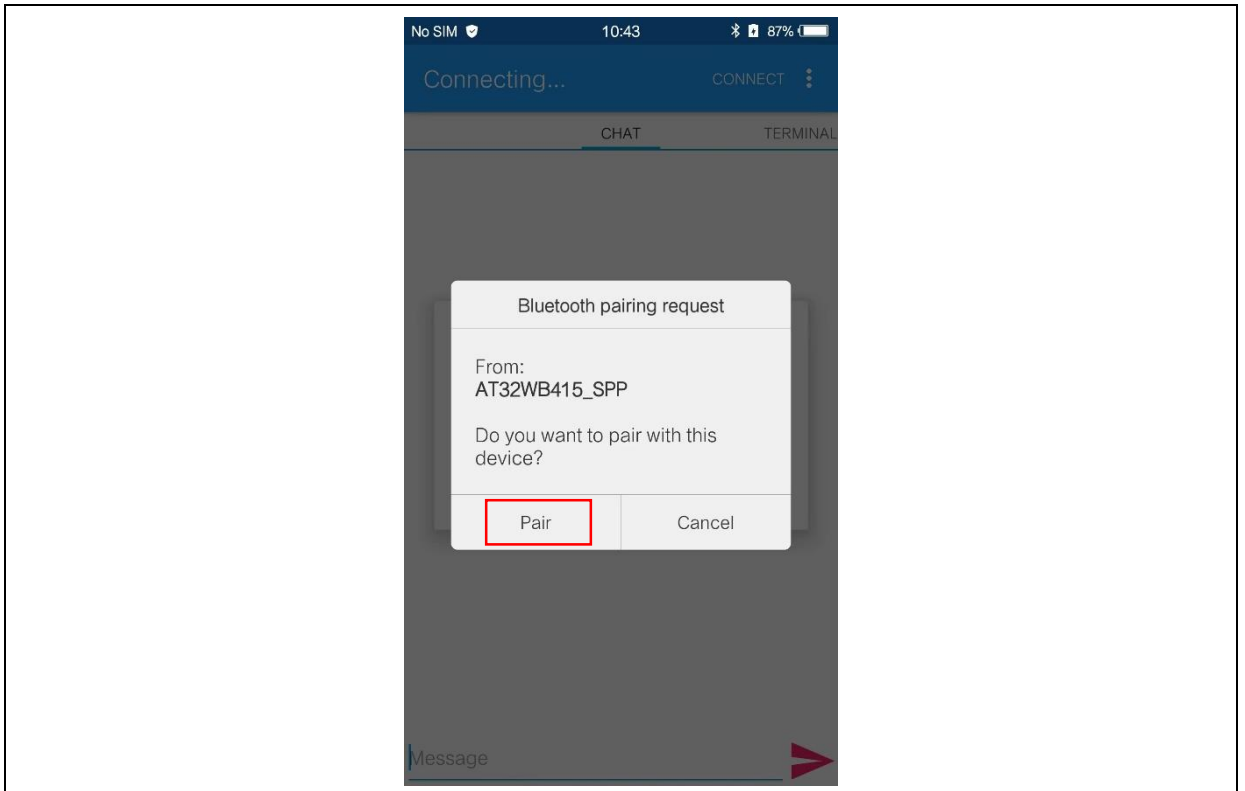


图 14.上位机与 AT32WB415_SPP 进行数据收发测试

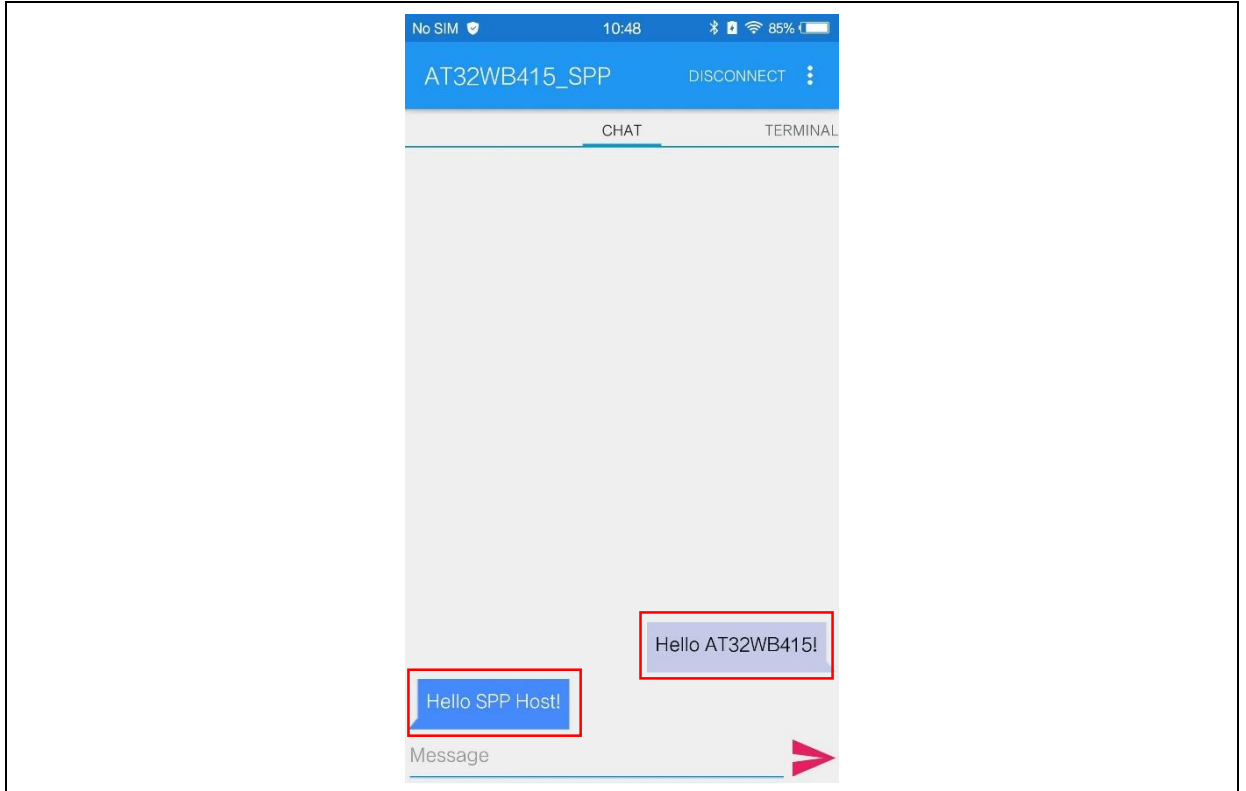
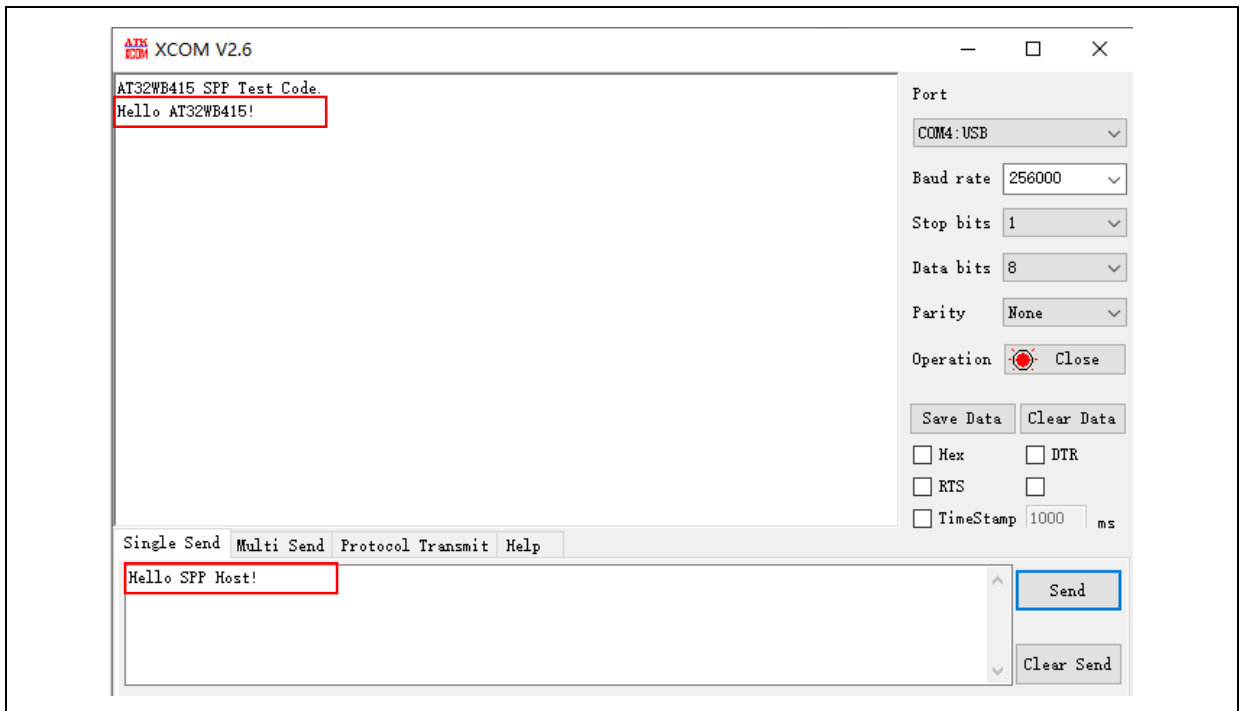


图 15. 开发板端通过串口收发数据



至此本案例的AT32WB415 SPP demo测试完成。

4 版本历史

表 1. 文档版本历史

日期	版本	变更
2022.10.19	2.0.0	初始版本
2022.11.03	2.0.1	增加电脑端蓝牙串口测试方法

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 保留所有权利