
BLE 实战教程

——疯壳·开发板系列

Wolverine-Team

2015/5/24

目录

术语和定义.....	4
一、与手机 APP 通信实验.....	4
二、手机 APP 控制 LED 灯实验.....	9
三、手机 APP 读取步伐实验.....	11
四、手机 APP 与 PC 串口通信.....	14
五、如何修改 BLE 的一些参数.....	19
5.1 修改服务及特征值的 UUID.....	19
5.2 修改广播数据.....	19
5.3 修改 MAC 地址.....	20
5.4 修改设备名称.....	20

官网地址: <http://www.fengke.club>
购买链接: <http://shop115904315.taobao.com/>
官方 QQ 群: 193836402

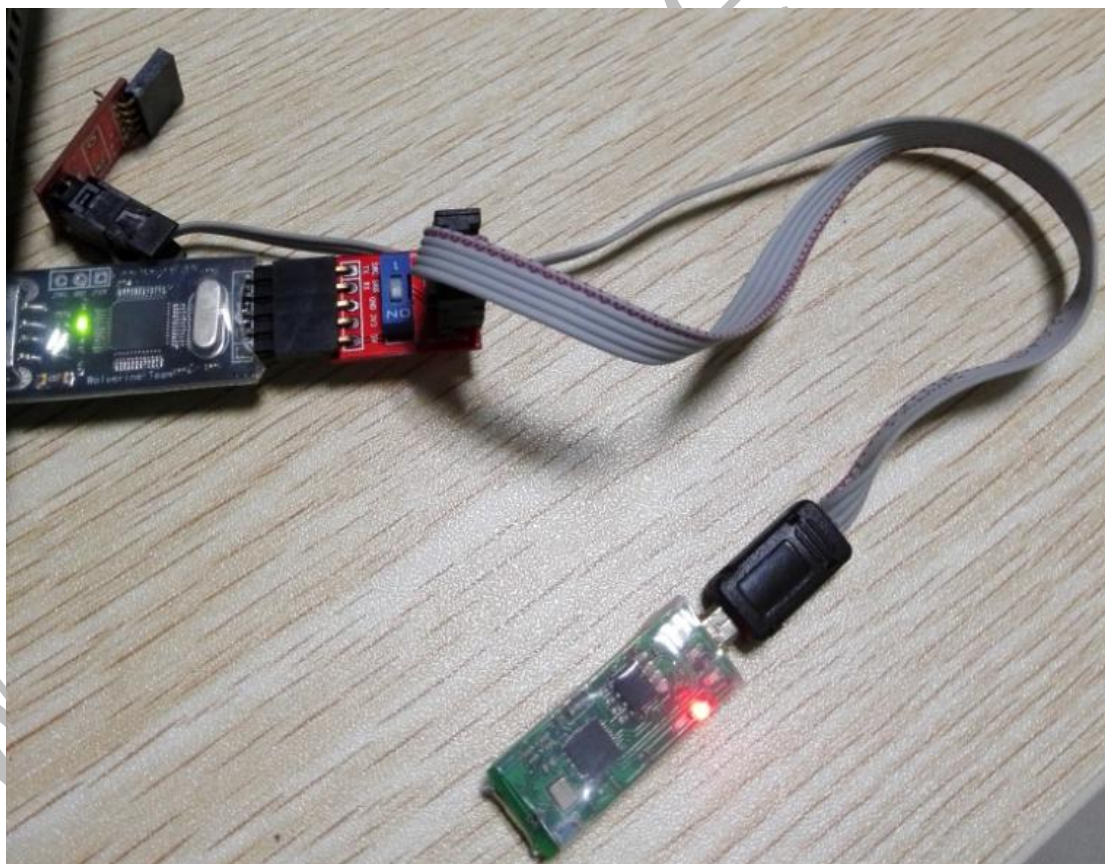
术语和定义

BLE	低功耗蓝牙 (Bluetooth Low Energy)
SDK	软件开发工具包 (Software Development Kit)
UUID	通用唯一标识符 (Universally Unique Identifier)
APP	手机应用软件 (Application)
MAC	媒体访问控制 (Media Access Control)
GPIO	通用输入输出 (General Purpose Input/Output)
INT	中断 (Interrupt)
ADC	模数转换 (Analog to Digital Converter)
I2C	集成电路通信总线 (Inter-Integrated Circuit bus)
SPI	串行外设接口 (Serial Peripheral Interface)
UART	通用异步收发器 (Universal Asynchronous Receiver/Transmitter)

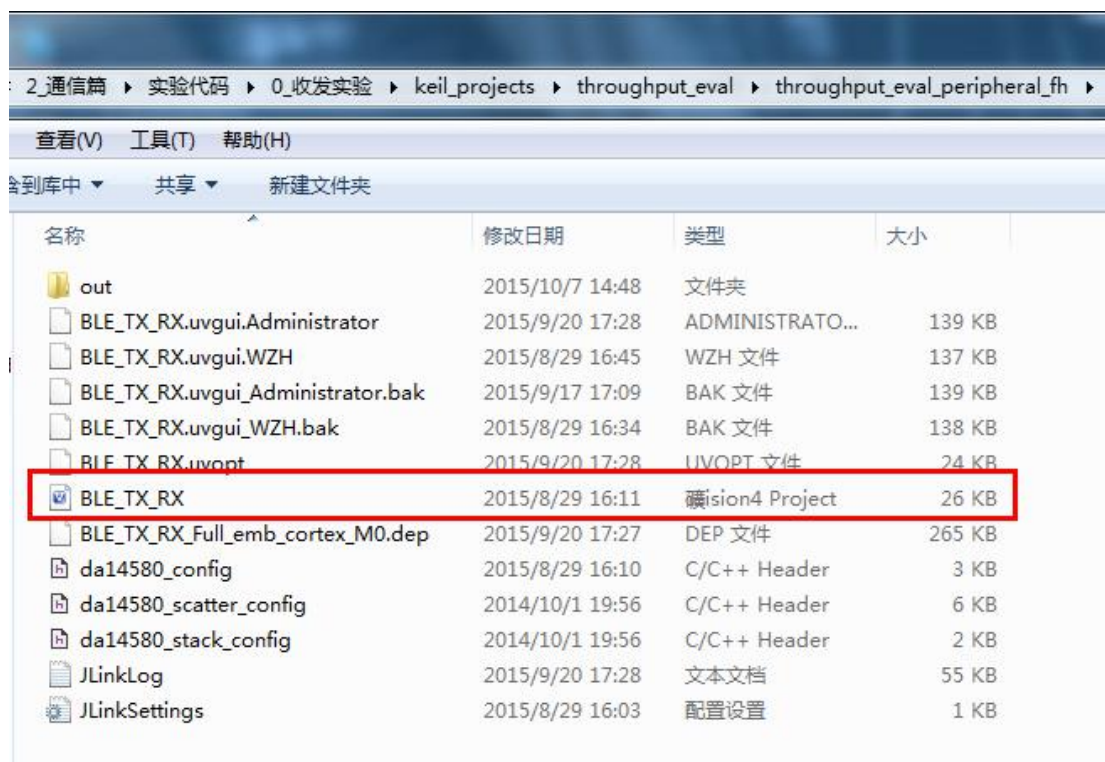
一、与手机 APP 通信实验

实验需要使用的模块有：手环，Jlink 调试工具，一根手环下载调试线。

将 JLINK 通过下载调试线连接到手环的 USB 调试接口，JLINK 插在有拨码开关的一端，注意丝印标注一一对应，将 JLINK 插上电脑的 USB 口，如下图所示：



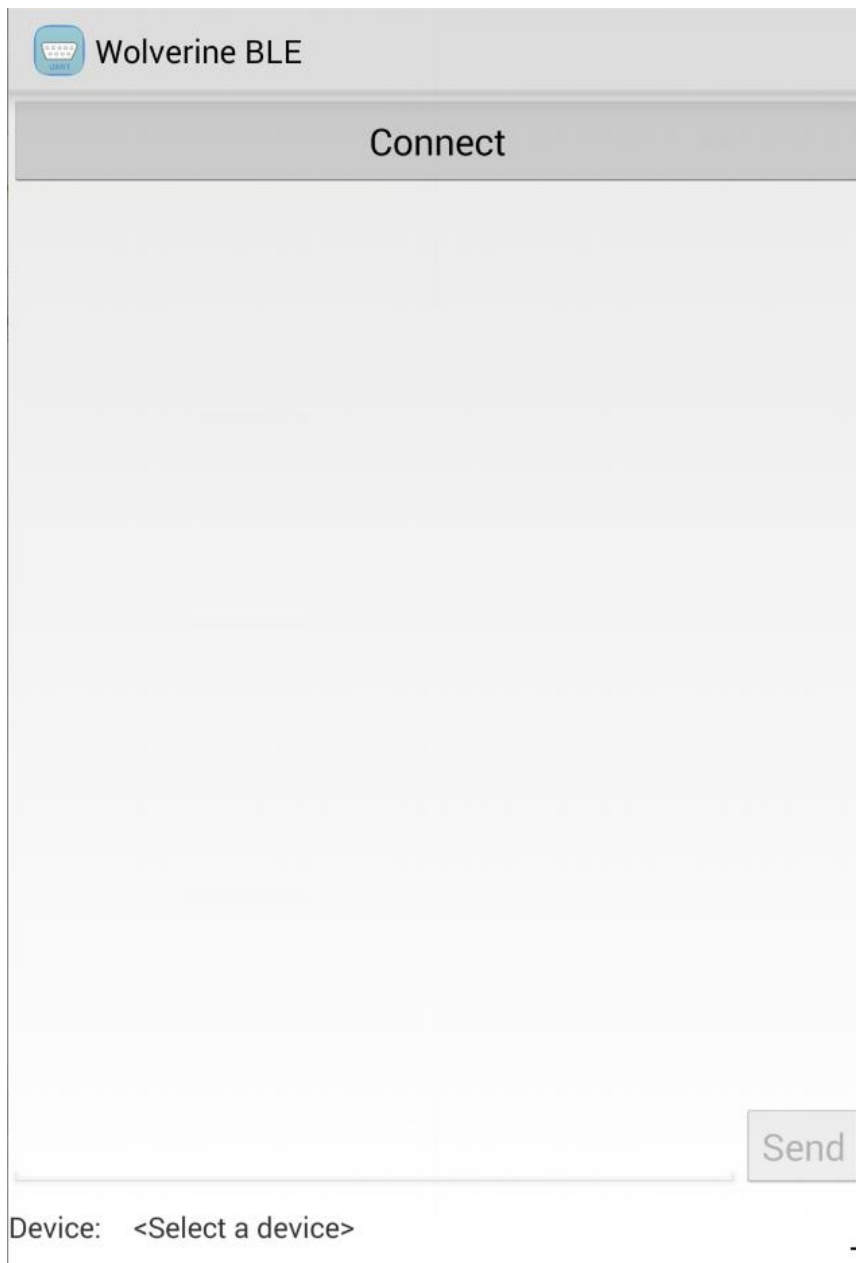
打开收发实验的Keil工程BLE_TX_RX.uvproj，位于目录：..\4_实战教程\2_通信篇\实验代码\0_收发实验\keil_projects\throughput_eval\throughput_eval_peripheral_fh。如下图所示：

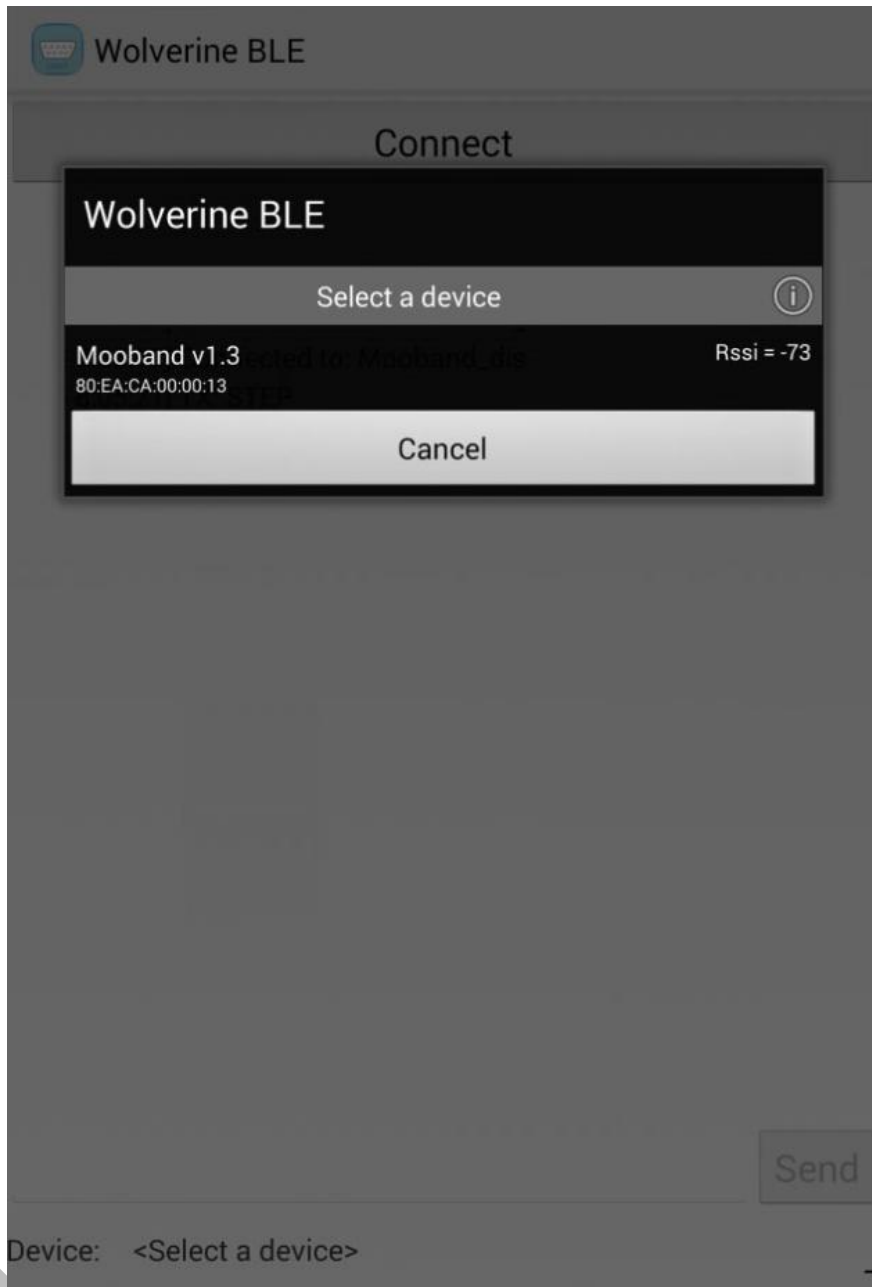


打开工程 KEIL 之后，点击编译，点击 DEBUG，然后点击全速运行，如下图所示：

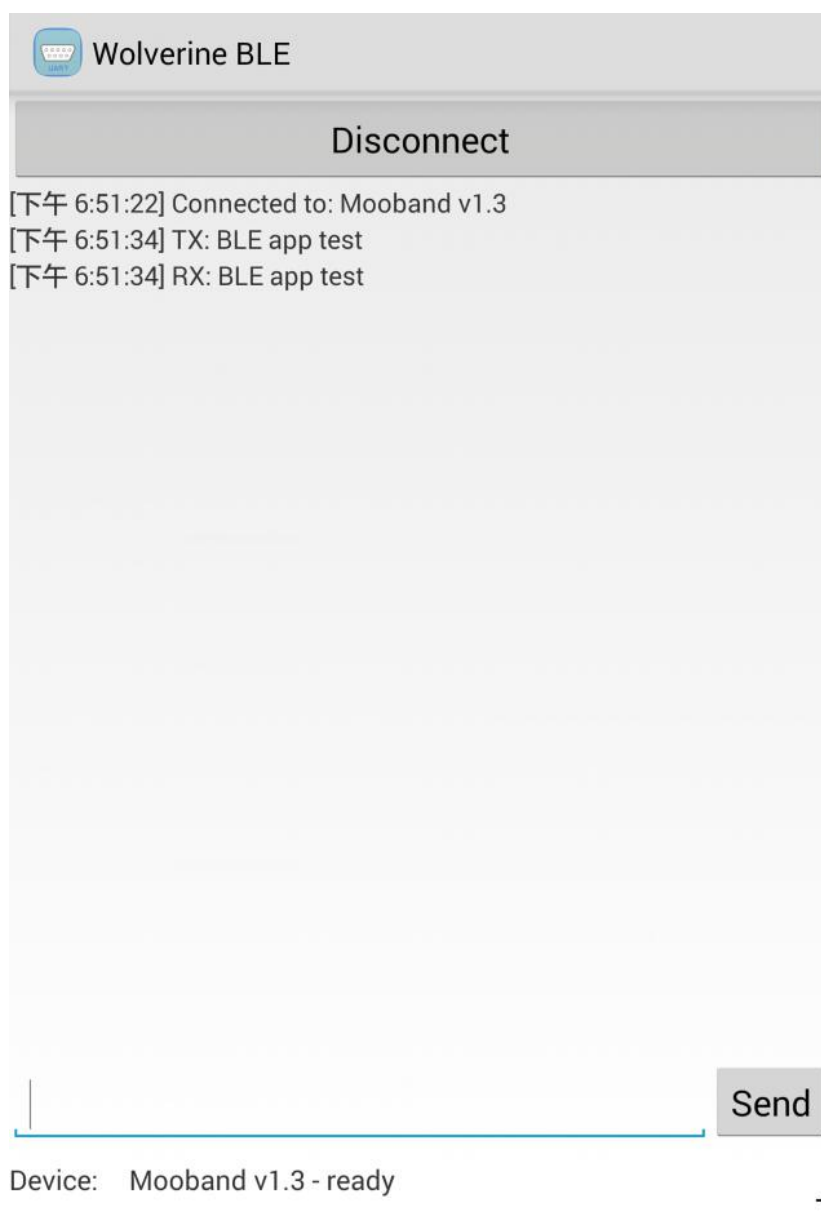


打开手机APP，点击Connect搜索到Mooband v1.3，点击链接，显示Connected to:Mooband v1.3，发送数据，就会接收到同样的数据，如下图所示：





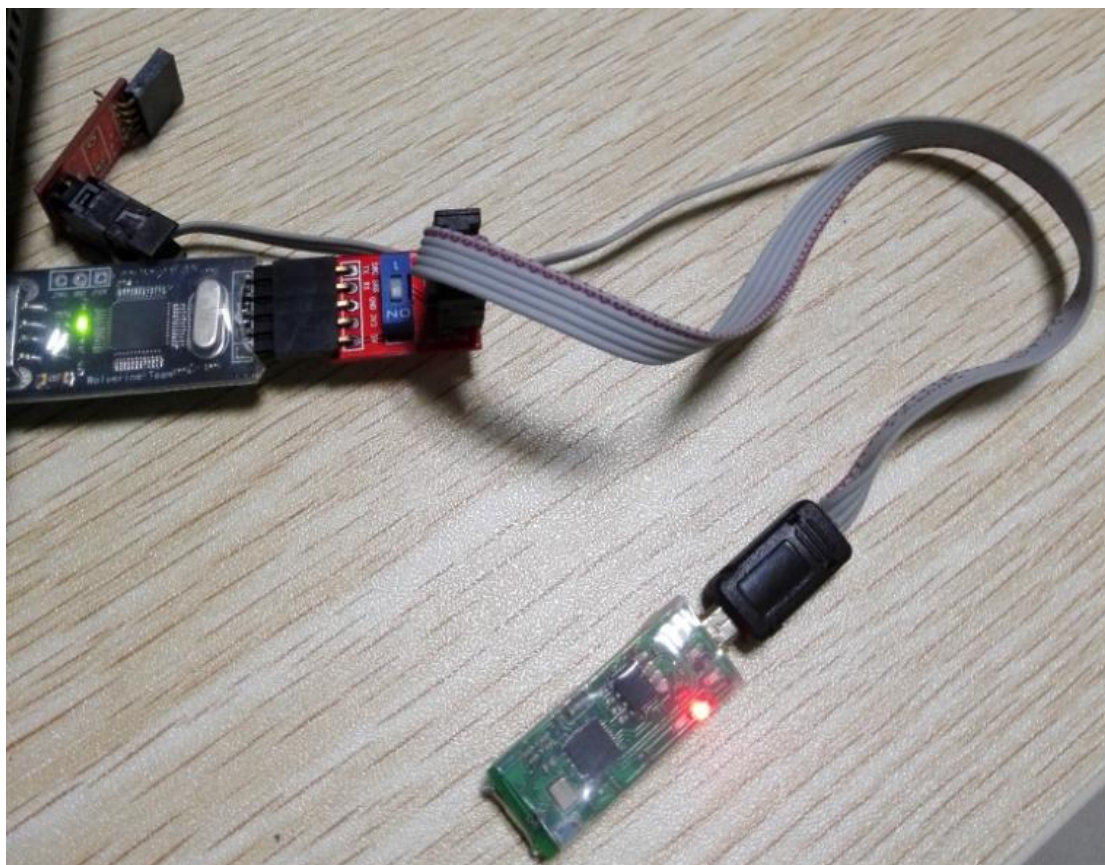




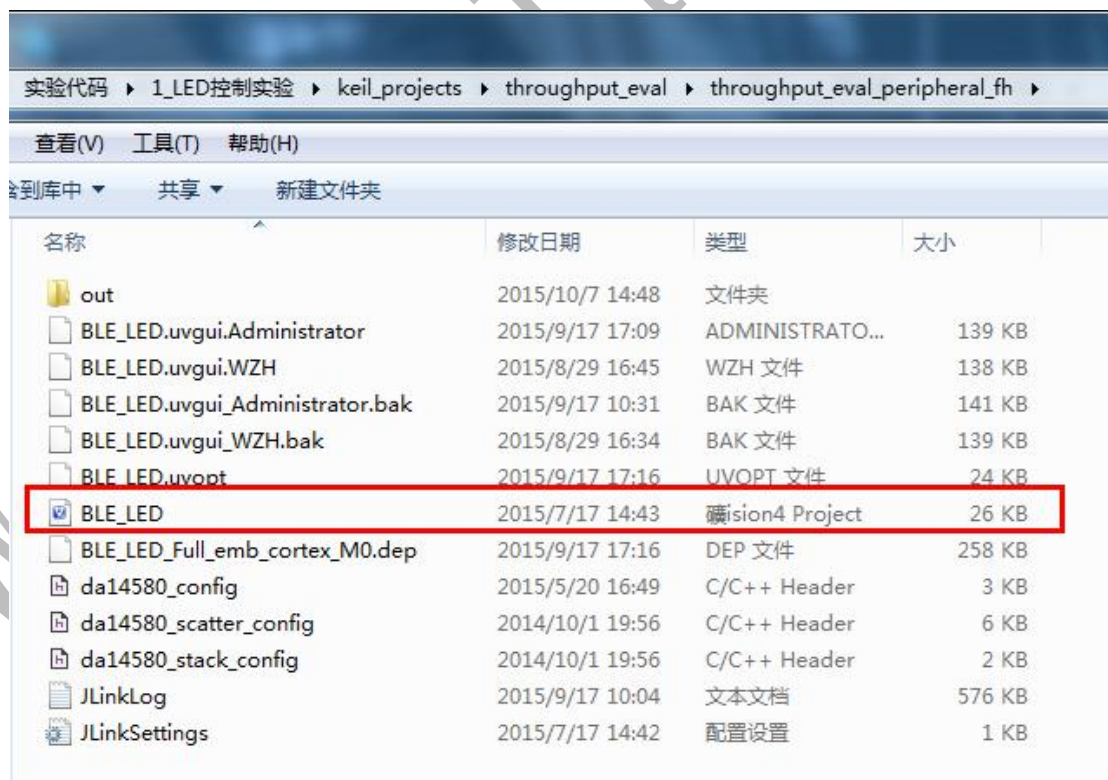
二、手机 APP 控制 LED 灯实验

实验需要使用的模块有：手环，Jlink 调试工具，一根手环下载调试线。

将 JLINK 通过下载调试线连接到手环的 USB 调试接口，JLINK 插在有拨码开关的一端，注意丝印标注一一对应，将 JLINK 插上电脑的 USB 口，如下图所示：



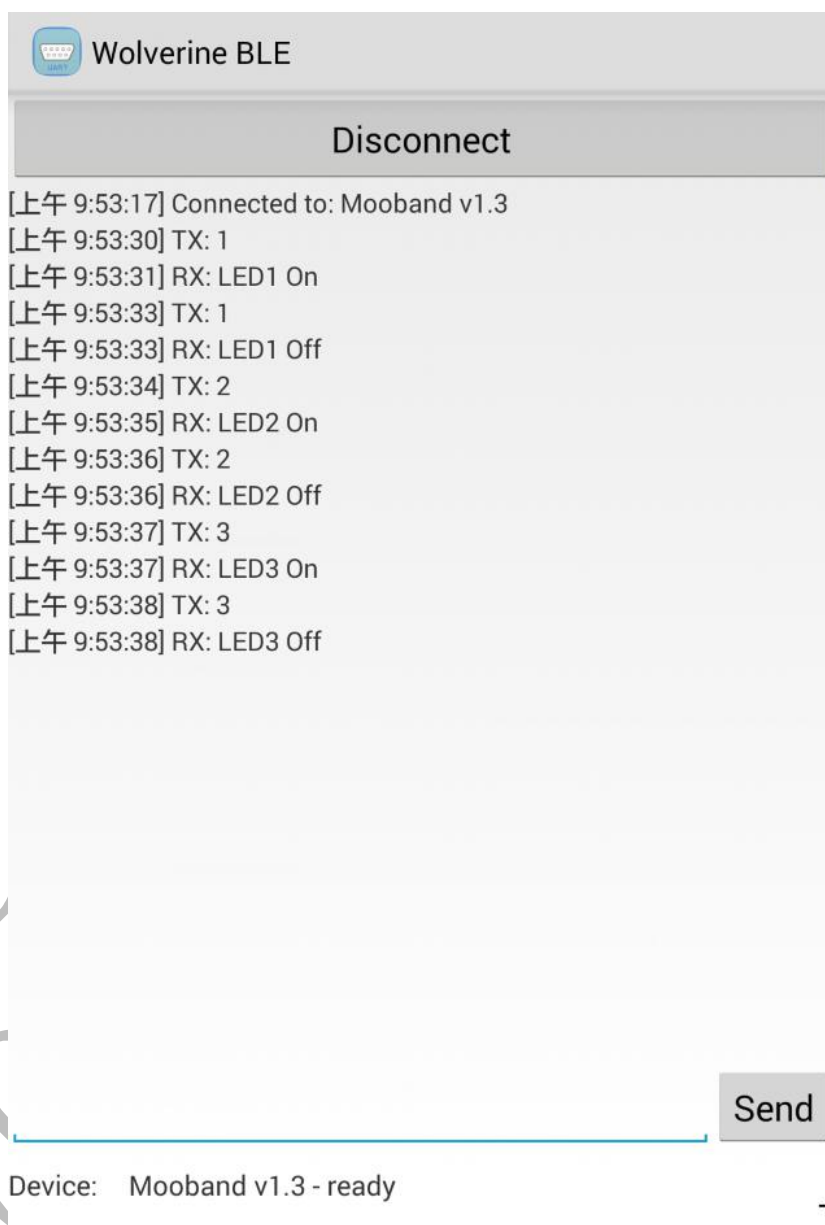
打开LED控制实验的Keil工程BLE_LED.uvproj, 位于目录: ..\4_实战教程\2_通信篇\实验代码\1_LED控制实验\keil_projects\throughput_eval\throughput_eval_peripheral_fh。如下图所示:



打开 KEIL 工程之后，点击编译，点击 DEBUG，然后点击全速运行，如下图所示：



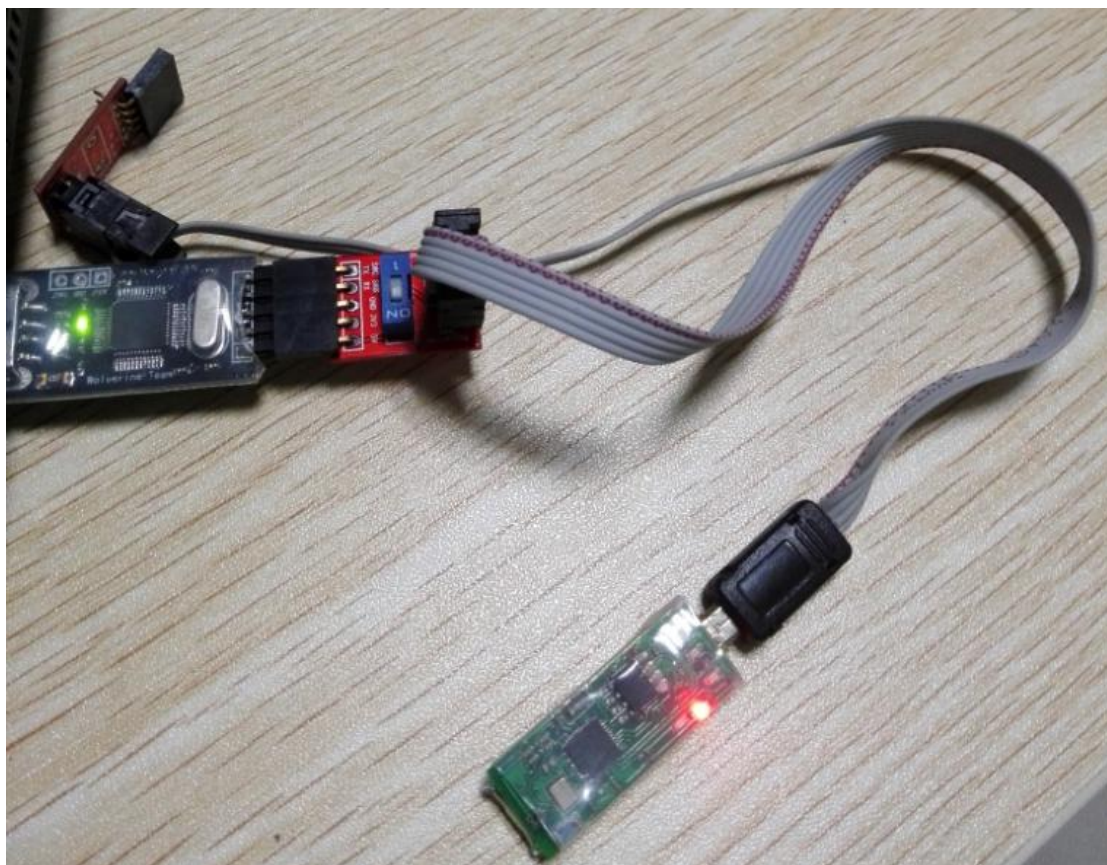
打开手机APP，点击连接手环，手机APP发送1，控制LED1的亮灭，发送2，控制LED2的亮灭，发送3，控制LED3的亮灭，如下图所示：



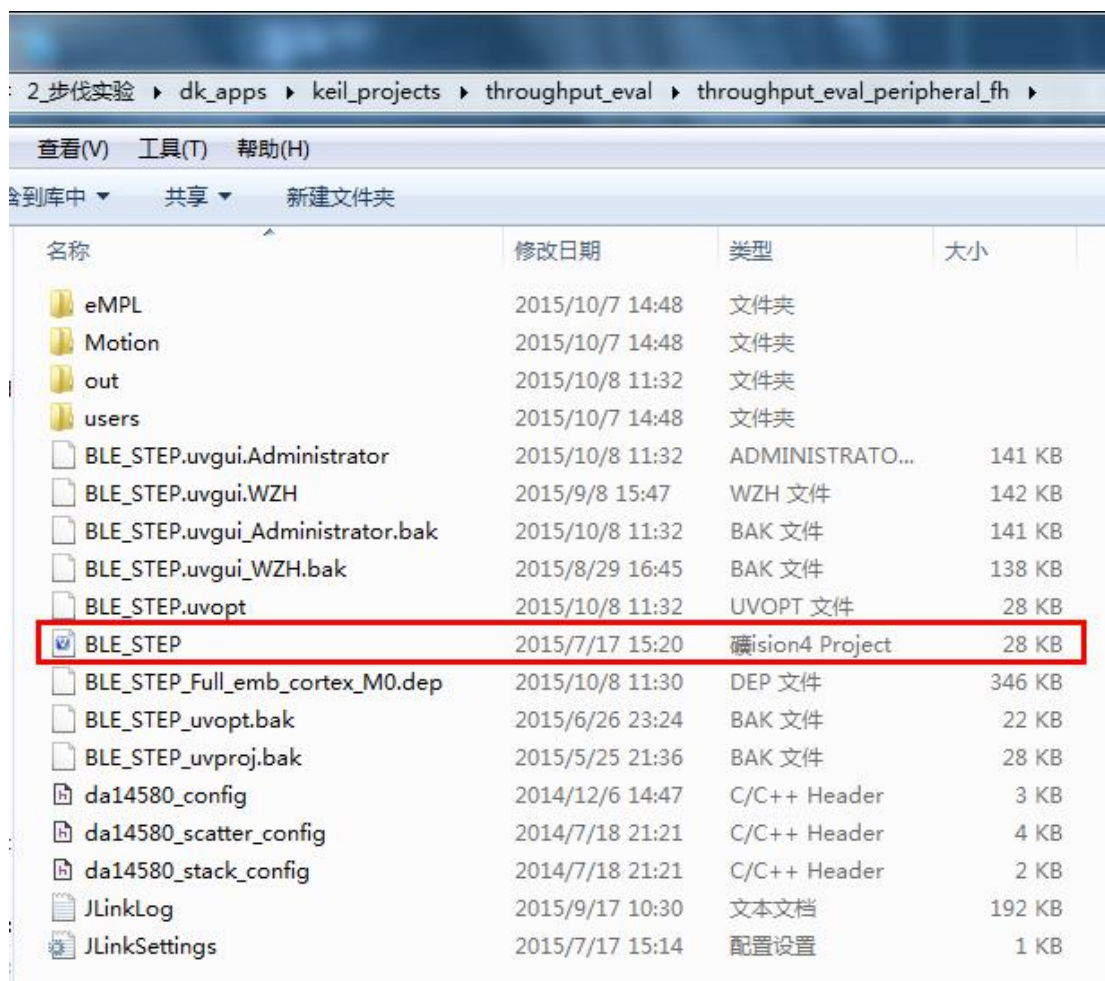
三、手机 APP 读取步伐实验

实验需要使用的模块有：手环，Jlink 调试工具，一根手环下载调试线。

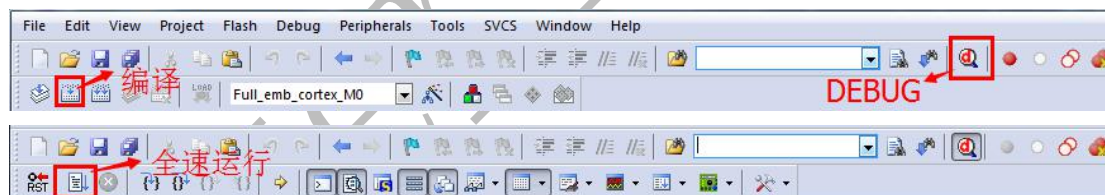
将 JLINK 通过下载调试线连接到手环的 USB 调试接口，JLINK 插在有拨码开关的一端，注意丝印标注一一对应，将 JLINK 插上电脑的 USB 口，如下图所示：



打开步伐实验的Keil工程BLE_STEP.uvproj，位于目录：..\4_实战教程\2_通信篇\实验代码\2_步伐实验\dk_apps\keil_projects\throughput_eval\throughput_eval_peripheral_fh。如下图所示：



打开 KEIL 工程之后，点击编译，点击 DEBUG，然后点击全速运行，如下图所示：



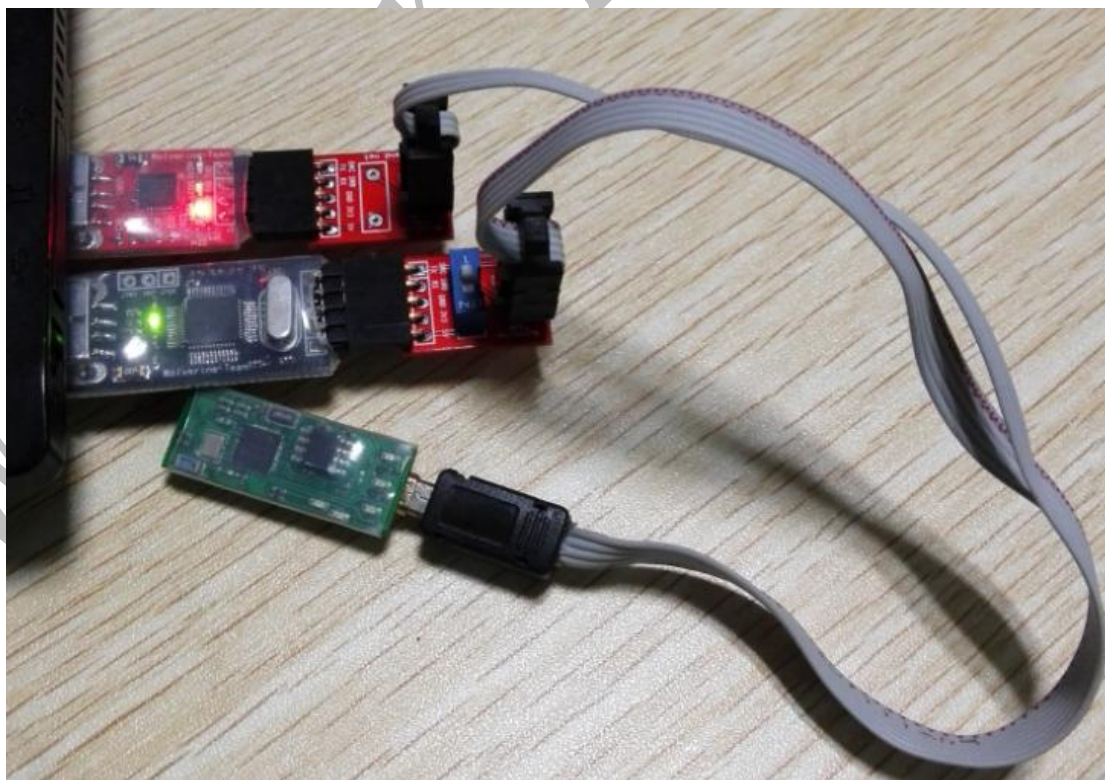
打开手机APP，连接手环，发送“STEP”读取步伐卡路里，如下图所示：



四、手机 APP 与 PC 串口通信

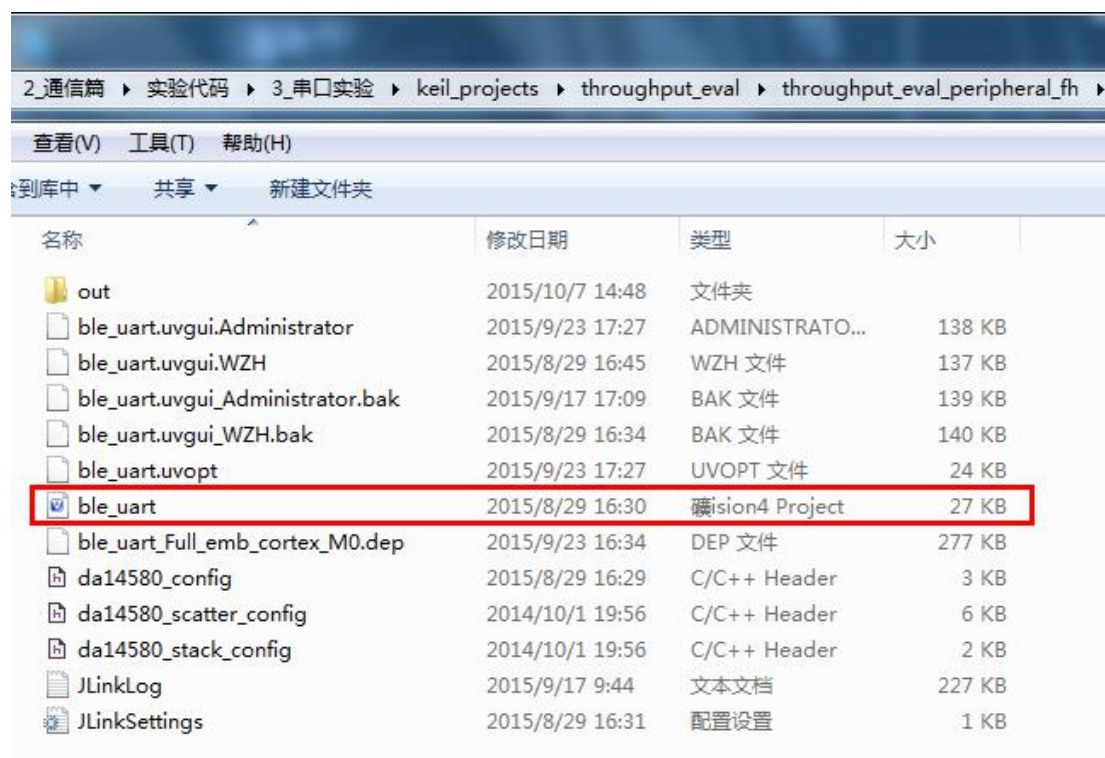
实验需要使用的模块有：手环，Jlink 调试工具，USB 转串模块，一根手环下载调试线。

将 JLINK 通过下载调试线连接到手环的 USB 调试接口，JLINK 插在有拨码开关的一端，注意丝印标注一一对应，将 JLINK 插上电脑的 USB 口。将 USB 转串模块插在手环现在调试线的另一端，注意丝印标注一一对应，然后将 USB 转串模块插在电脑的 USB 接口。如下图所示：



打开串口实验的Keil工程ble_uart.uvproj，位于目录：..\4_实战教程\2_通信篇\实验代码

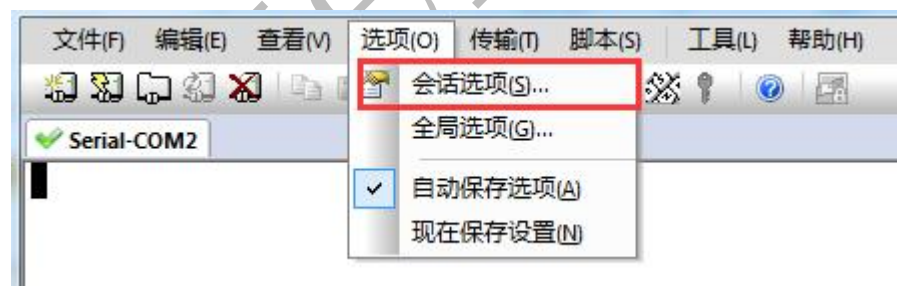
\\3_串口实验\keil_projects\throughput_eval\throughput_eval_peripheral_fh，如下图所示：

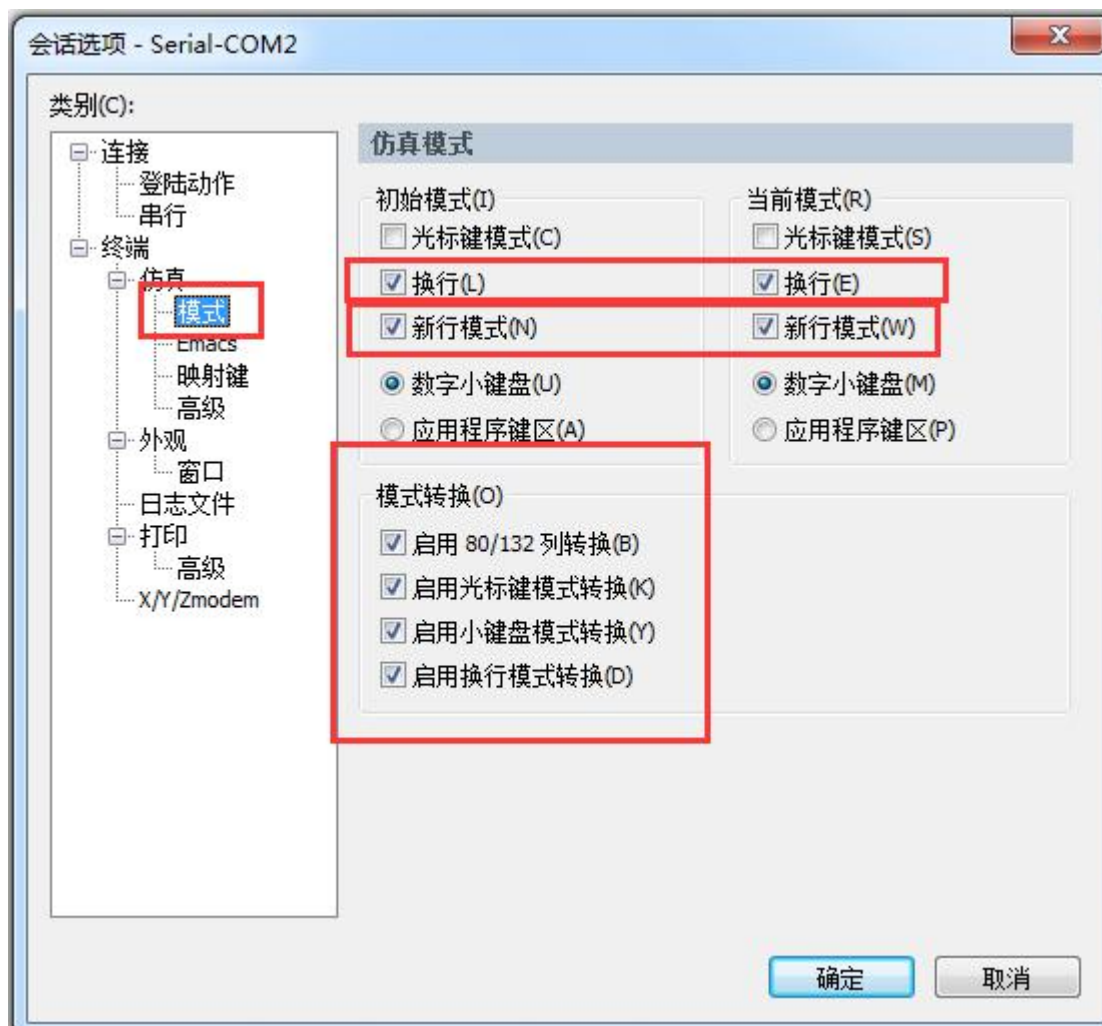


打开 KEIL 工程之后，点击编译，点击 DEBUG，然后点击全速运行，如下图所示：

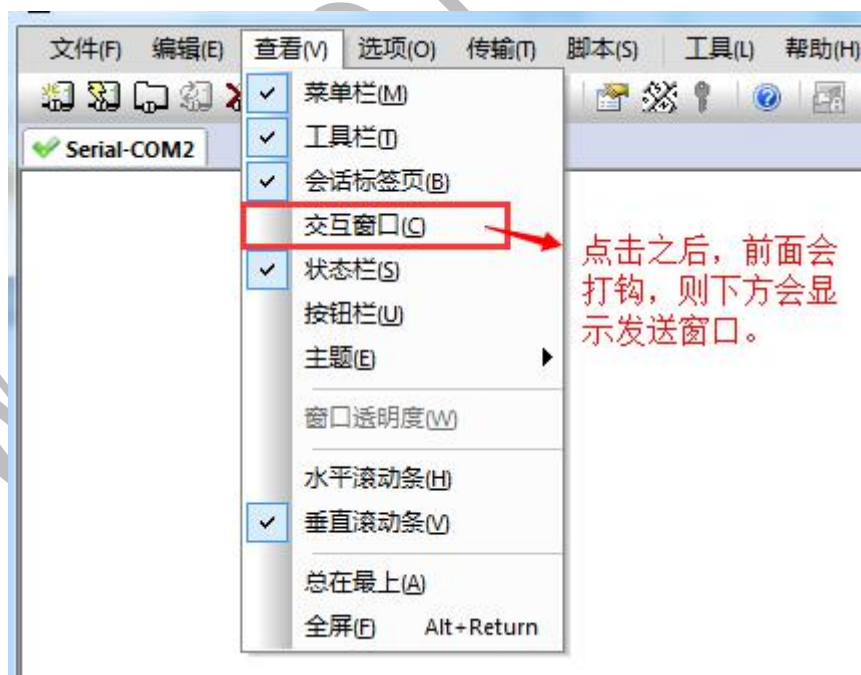


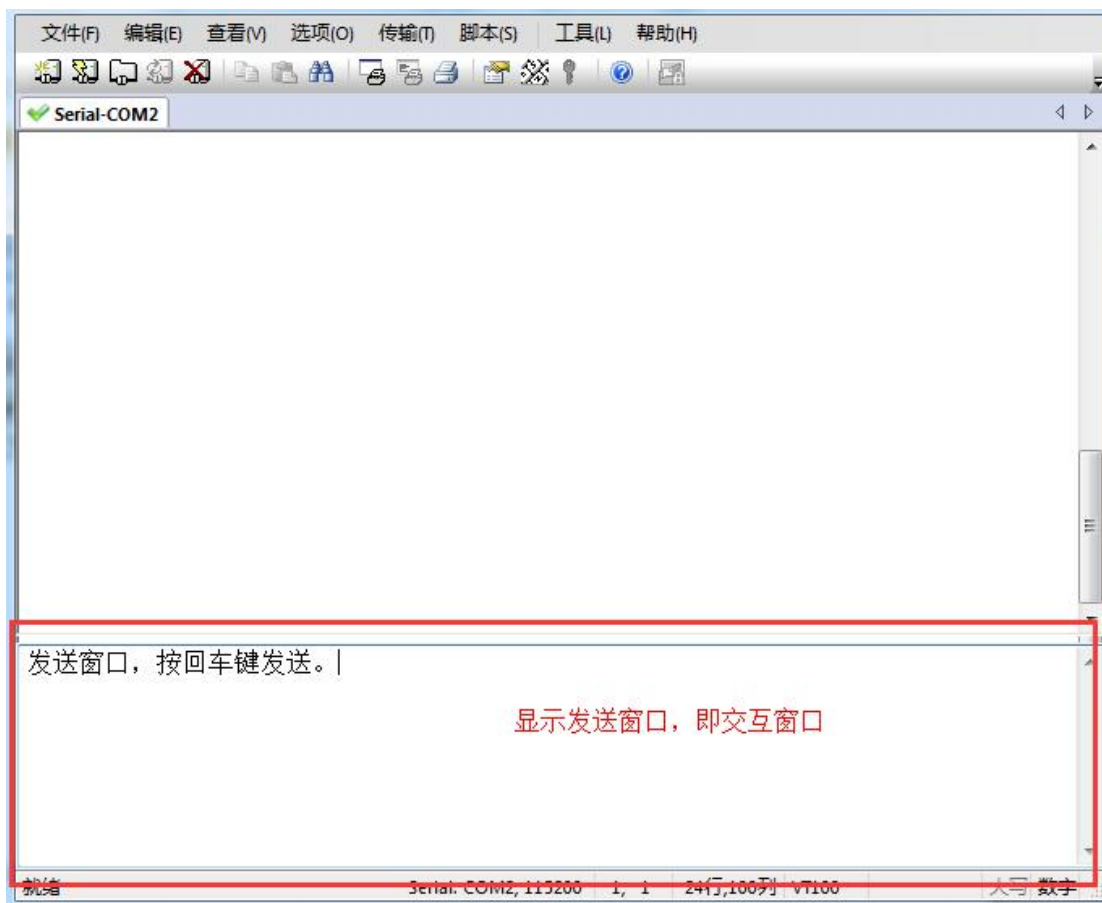
打开串口调试助手，波特率设置为115200。由于程序中串口接受中断需要检测回车换行来判断是否为一次字符串发送，对串口调试助手做如下设置，如图所示：



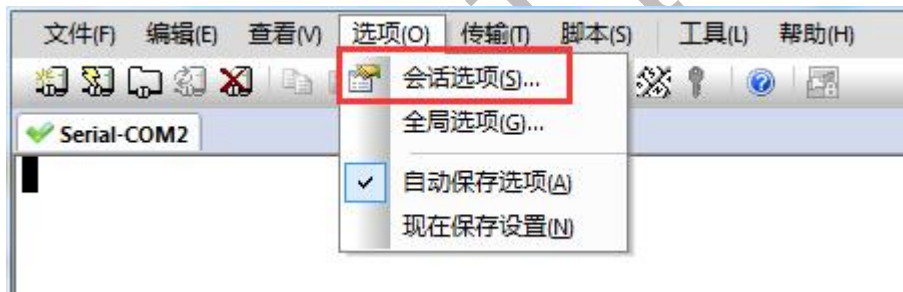


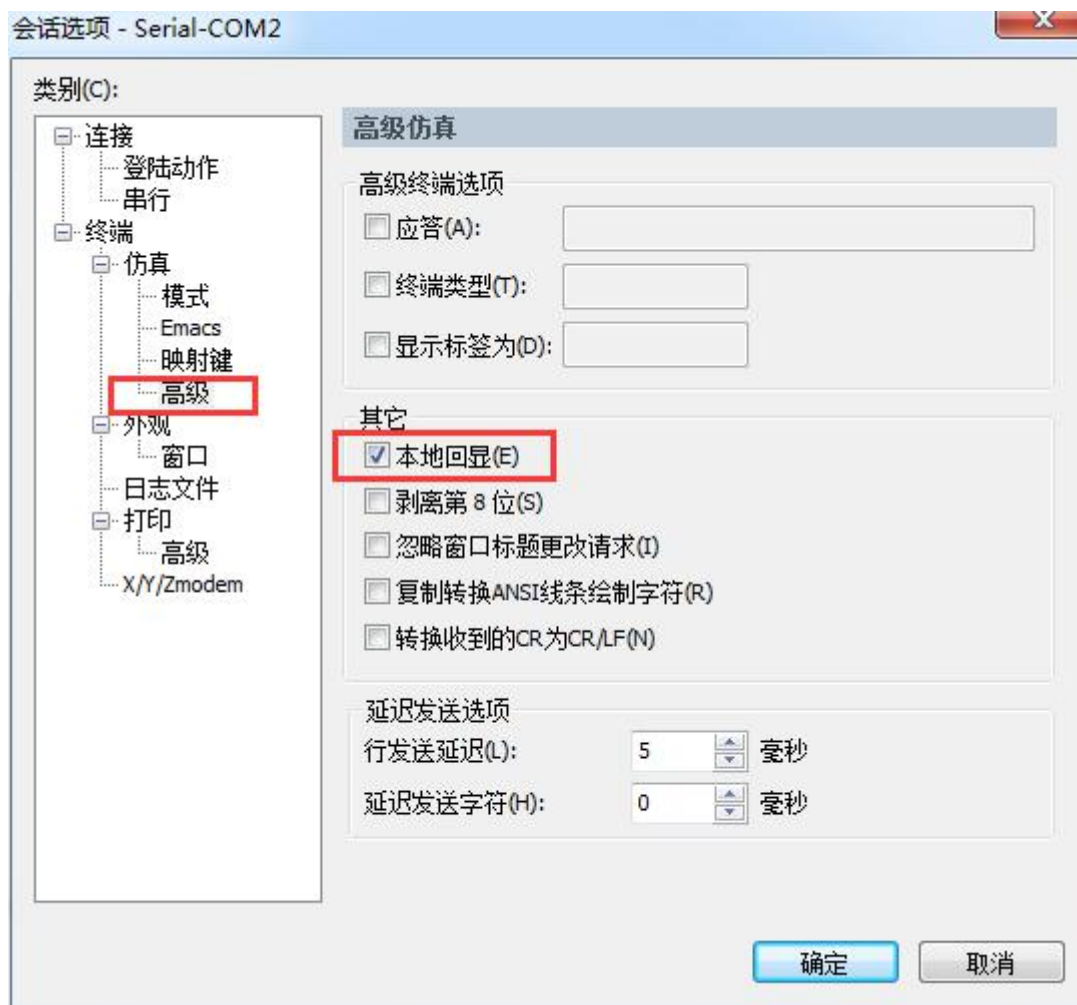
另外，如果需要显示发送串口，进行如下设置，如下图所示：



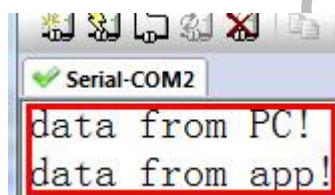


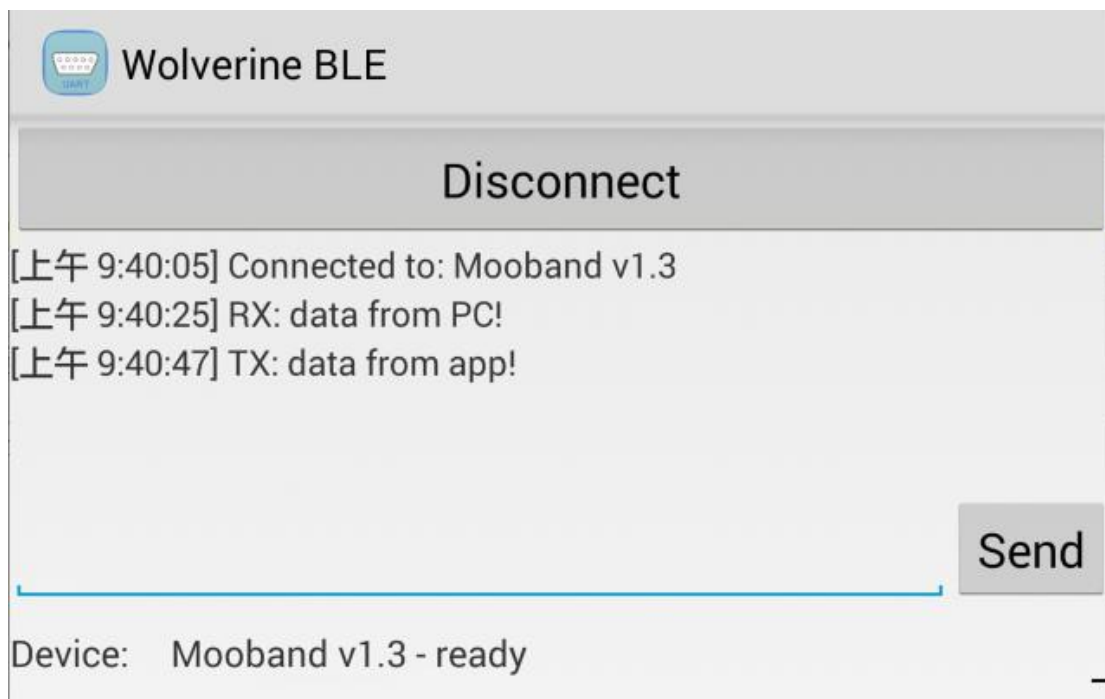
如果要在接收串口中显示自己发送的字符，进行如下设置，如下图所示：





打开手机APP，点击连接之后，发送的数据通过串口发送到电脑中的串口调试助手显示（一次性不超过20个字符），串口调试助手发送的数据在手机APP显示。结果如下图所示：





五、如何修改 BLE 的一些参数

5.1 修改服务及特征值的 UUID

BLE 协议栈自己规定的一些服务及特征值的 UUID 是固定的，不可以修改之外，其它的 UUID 都是可以修改的。注：因为手机蓝牙的记忆性，修改完 UUID 之后需要关闭手机蓝牙及 APP，重新打开搜索才能显示修改后的值。

UUID 就是唯一识别码，唯一代表对应的服务或者特征值。一般在对应的 profile 头文件中定义 UUID，如下图红框部分所示：

```
streamdatad.h
52
53 // Size of a single streamdatad packet i
54 #define STREAMDATAD_PACKET_SIZE (20)
55 #define MAX_TRANSMIT_BUFFER_PACKETS (10)
56
57 //extern uint8_t nb_buf_av;
58
59 // Proprietary UUIDs
60 enum
61 {
62     STREAMDATAD_SERVICE_UUID = 0xfff0,
63     STREAMDATAD_ENABLE_UUID = 0xFFF1,
64     STREAMDATAD_DO_UUID = 0xFFF4,
```

5.2 修改广播数据

广播函数 `app_adv_func(struct gapm_start_advertise_cmd *cmd)` 中，如下图红框部分所示：

```
app_throughput_peripheral_proj.c
261  */
262  void app_adv_func(struct gapm_start_advertise_cmd *cmd)
263  {
264
265  // Start advertising. Fill GAPM_START_ADVERTISE_CMD message
266
267  // Device Name Length
```

5.3 修改 MAC 地址

在 nvds.c 文件中定义了 BLE 相关的参数，NVDS_TAG_BD_ADDRESS 为 MAC 地址变量，如下图所示：

```
nvds.c
34  extern struct nvds_data_struct *nvds_data_ptr __attribute__((section("retention_mem_area0"), zero_init));
35
36  #ifdef READ_NVDS_STRUCT_FROM_OTP
37  const struct nvds_data_struct nvds_data_storage __attribute__((section("nvds_data_storage_area")));
38  #else
39  const struct nvds_data_struct nvds_data_storage __attribute__((section("nvds_data_storage_area"))) =
40  {
41  .NVDS_VALIDATION_FLAG           = 0xffff,
42  .NVDS_TAG_UART_BAUDRATE        = 115200,
43  .NVDS_TAG_DIAG_SW              = 0,
44  .NVDS_TAG_DIAG_BLE_HW         = 0,
45  .NVDS_TAG_NEB_ID               = 0,
46  .NVDS_TAG_LPCLK_DRIFT          = DRIFT_BLE_DFT,
47  .NVDS_TAG_SLEEP_ENABLE        = 1,
48  .NVDS_TAG_EXT_WAKEUP_ENABLE    = 0,
49  .NVDS_TAG_SECURITY_ENABLE      = 1,
50  .NVDS_TAG_APP_BLE_ADV_DATA     = "\x1A\xff\xd2\x00\x02\x15\x58\x5C\xDE\x93\x1B\x01\x42\xCC\x",
51  .NVDS_TAG_APP_BLE_SCAN_RESP_DATA = "\x09\xff\x00\x60\x52\x57\x2D\x42\x4C\x45",
52  .NVDS_TAG_DEVICE_NAME          = "Mooband v1.3",
53  .NVDS_TAG_BD_ADDRESS           = {0x13, 0x00, 0x00, 0xCA, 0xEA, 0x80},
54  .ADV_DATA_TAG_LEN              = 27,
55  .SCAN_RESP_DATA_TAG_LEN       = 10,
56  .DEVICE_NAME_TAG_LEN          = 12,
57  // Default Channel Assessment Timer duration (20s - Multiple of 10ms)
```

5.4 修改设备名称

设备名是在手机 APP 搜索设备时显示的名称，在 nvds.c 文件中，修改 NVDS_TAG_DEVICE_NAME 的值以及对应的名称长度值 DEVICE_NAME_TAG_LEN 即可。注：因为手机蓝牙的记忆性，修改完设备名称之后需要关闭手机蓝牙及 APP，重新打开搜索才能显示修改后的值。如下图所示：

```
nvds.c
34 extern struct nvds_data_struct *nvds_data_ptr __attribute__((section("retention_mem_area0"), zero_init)); //@R
35
36 #ifdef READ_NVDS_STRUCT_FROM_OTP
37 const struct nvds_data_struct nvds_data_storage __attribute__((section("nvds_data_storage_area")));
38 #else
39 const struct nvds_data_struct nvds_data_storage __attribute__((section("nvds_data_storage_area"))) =
40 {
41     .NVDS_VALIDATION_FLAG           = 0x1ffff,
42     .NVDS_TAG_UART_BAUDRATE        = 115200,
43     .NVDS_TAG_DIAG_SW              = 0,
44     .NVDS_TAG_DIAG_BLE_HW          = 0,
45     .NVDS_TAG_NEB_ID               = 0,
46     .NVDS_TAG_LPCLK_DRIFT           = DRIFT_BLE_DFT,
47     .NVDS_TAG_SLEEP_ENABLE         = 1,
48     .NVDS_TAG_EXT_WAKEUP_ENABLE    = 0,
49     .NVDS_TAG_SECURITY_ENABLE      = 1,
50     .NVDS_TAG_APP_BLE_ADV_DATA     = "\x1A\xff\xd2\x00\x02\x15\x58\x5C\xDE\x93\x1B\x01\x42\xCC\x9A\x1
51     .NVDS_TAG_APP_BLE_SCAN_RESP_DATA = "\x09\xff\x00\x60\x52\x57\x2D\x42\x4C\x45",
52     .NVDS_TAG_DEVICE_NAME           = "Mooband v1.3",
53     .NVDS_TAG_BD_ADDRESS            = {0x13, 0x00, 0x00, 0xCA, 0xEA, 0x80},
54     .ADV_DATA_TAG_LEN               = 27,
55     .SCAN_RESP_DATA_TAG_LEN        = 10,
56     .DEVICE_NAME_TAG_LEN           = 12,
57     /// Default Channel Assessment timer duration (20s - Multiple of 10ms)
58     .NVDS_TAG_BLE_CA_TIMER_DUR     = 2000,
59     /// Default Channel Reassessment Timer duration (Multiple of Channel Assessment Timer duration)
60     .NVDS_TAG_BLE_CRA_TIMER_DUR    = 6,
61     /// Default Minimal RSSI Threshold - -48dBm
62     .NVDS_TAG_BLE_CA_MIN_RSSI      = 0x40, //0xD0,
63     /// Default number of packets to receive for statistics
64     .NVDS_TAG_BLE_CA_NB_PKT        = 100,
65     /// Default number of bad packets needed to remove a channel
66     .NVDS_TAG_BLE_CA_NB_BAD_PKT    = 50,
67 };
68 #endif
```