Application Notes
ZHCA841 – Dec. 2018

基于 CC2640R2F 的 BLE HID 浅析与应用

Yan Zhang/Holly Gu *China Central FAE/ East China FAE*

摘要

随着智能化产品的普及，无线键盘、鼠标、手柄等基于 BLE HID 协议的设备越来越受到消费者的欢迎，相较传统蓝牙键盘鼠标，具有功耗更低，配对连接更快的特点。在客户现有的鼠标、键盘上，可以增加 BLE 实现产品的快速升级，而 BLE HID 协议因为相对复杂，常常需要开发者花费大量精力。同时，BLE 的 HID 设备开发通常没有一个很好的协议调试方式，专用的 sniffer 工具成本太高，应用并不广泛。本文从客户的实际问题出发，基于 TI 公司出的最新一代超低功耗 BLE SOC CC2640R2F，浅析 HID 在 BLE 协议上应用开发的关键知识点，同时根据实际应用的痛点，介绍如何灵活运用 TI 提供的 PC 上位机软件 BTool，根据 BLE 和 BLE HID 的协议规范，通过手动操作这些协议来进行通信和数据交互，为基于 CC2640R2F 的 BLE HID 应用开发过程中协议调试提供一种低成本和高效率的解决方案。更可以通过学习本文，增加对 BLE HID 协议和实际应用的理解。

目录

图表

1 BLE HID 规范简介

BLE 的 HID 全称是 HOGP (HID Over GATT Profile)，CC2640R2F 支持完整的 HOGP。完整的协议技术文档可以从 SIG (蓝牙技术联盟) 的官网上下载：HID Service 文档，HOGP 文档。BLE profile 的定义是由 service 组成，所以分为 service 文档和 profile 文档。

1.1 BLE HID 的特征值和 UUID

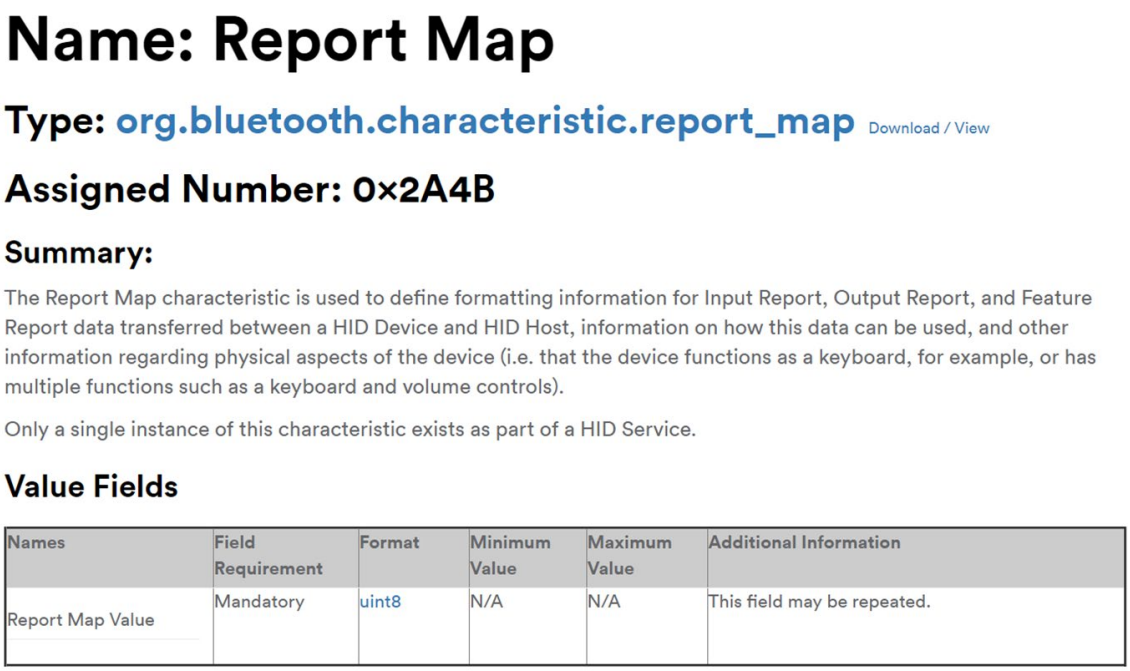
对于任何 BLE profile 来说，核心就是对其定义的特征值进行一系列操作来完成数据通信，达到产品定义的功能，HOGP 也不例外。注意前面所说的 HID Service 文档和 HOGP 文档并不包含特征值和 UUID 的定义，对于这两者的定义在 SIG 的官网上有独立说明。为了让读者更好地理解实际代码和后面章节用 BTool 进行调试，本节将对 HOGP 定义的几个重要的特征值进行一下说明。下面将要说明的是 Report Map 和 Report 这两个特征值以及他们对应的 UUID。

1.2 Report Map 特征值和它的 UUID

BLE HOGP 的 Report Map 特征值在 SIG 官网有定义：Report Map。Report Map 特征值：

- 用于 HID 设备定义各种 Report 数据的格式，包括 Input Report，Output Report，Feature Report。同时定义了这些 Report 数据在 HID 设备和主机之间的通信协议。

- 用于 HID 设备类型的定义，比如遥控器，鼠标，键盘等等。
- UUID：0x2A4B



Name: Report Map

Type: org.bluetooth.characteristic.report_map Download / View

Assigned Number: 0x2A4B

Summary:

The Report Map characteristic is used to define formatting information for Input Report, Output Report, and Feature Report data transferred between a HID Device and HID Host, information on how this data can be used, and other information regarding physical aspects of the device (i.e. that the device functions as a keyboard, for example, or has multiple functions such as a keyboard and volume controls).

Only a single instance of this characteristic exists as part of a HID Service.

Value Fields

Names	Field Requirement	Format	Minimum Value	Maximum Value	Additional Information
Report Map Value	Mandatory	uint8	N/A	N/A	This field may be repeated.

图1. Report Map 特征值

Report Map 特征值的具体数据结构源自于 USB HID 的定义。在 BLE 中，Report Map 以特征值的形式，在连接建立并且加密后，通过主机 Read 的方式发送给 HID 的主机设备，以此来交互告诉主机设备连上的是什么类型的设备，支持哪些功能。比如说连上的是个键盘设备，支持哪些键盘按键等等。

1.3 Report 特征值和它的 UUID

BLE HOGP 的 Report 特征值在 SIG 官网有定义：Report。Report 特征值：

- 包括 Input Report，Output Report，Feature Report，这些 Report 用于 HID 设备和 HID 主机之间数据交换。其中 Input 和 Output 都是对于 HID 主机而言，Input 就是发送到 HID 主机的数据，Output 就是从 HID 主机发送出来向 HID 设备的数据。
- UUID：0x2A4D。

Name: Report

Type: org.bluetooth.characteristic.report Download / View

Assigned Number: 0x2A4D

Abstract:

The Report characteristic is used to exchange data between a HID Device and a HID Host.

Summary:

The Report characteristic value contains Input Report, Output Report or Feature Report data to be transferred between the HID Device and HID Host.

Value Fields

Names	Field Requirement	Format	Minimum Value	Maximum Value	Additional Information
Report Value	Mandatory	uint8	N/A	N/A	This field may be repeated.

图2. Report 特征值

Report 的定义也是源自于 USB 的 HID 定义，其中 Input Report 在 HOGP 中通常是以外设 Notification 的方式发送到主机，而 Output Report 将会以 Write 的形式从主机发送到外设。特别是 Input Report，比如鼠标，键盘，遥控器的按键，光标信息，都是通过 Input Report 发送给主机的。

2 基于 CC2640R2F 的 HID 工程软件解析

CC2640R2F 的 SDK 支持完整的 BLE HOGP (HID Over GATT Profile) 。目前默认的 SDK 里面，并未包含相关工程，但 TI 提供了额外的 SDK BLE example pack，在这个里面提供了 BLE HID 的示例代码。本文以里面的 hid_emu_kbd 工程为基础展开，此工程可直接用 IAR 或者 CCS 打开，路径如图 3 所示。

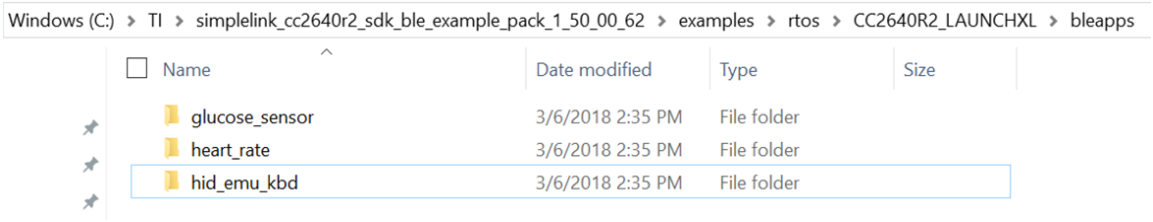


图3. hid_emu_kbd 工程路径

下文将以此工程为基础演示通过 BTool 调试 HID 的按键功能，并对应地看 HOGP 的 Report Map 特征值和 Report 特征值的具体实现，同时为了更好地演示 BTool 的 HID 调试功能，手动添加了一个 HID 的 Consumer page 到工程的 Report Map 里去。Consumer page 里面包含了音量加减，播放暂停等功能，普遍用于 BLE 遥控器设计中。

2.1 Report Map 特征值的实现

Report Map 在 HOGP 中，也是以特征值的形式存在于 HOGP 的 HID Service 中。在 hid_emu_kbd 工程里，HOGP 的 service 和特征值定义在 hidkbdservice.c 文件的 hidAttrTbl 数组里。

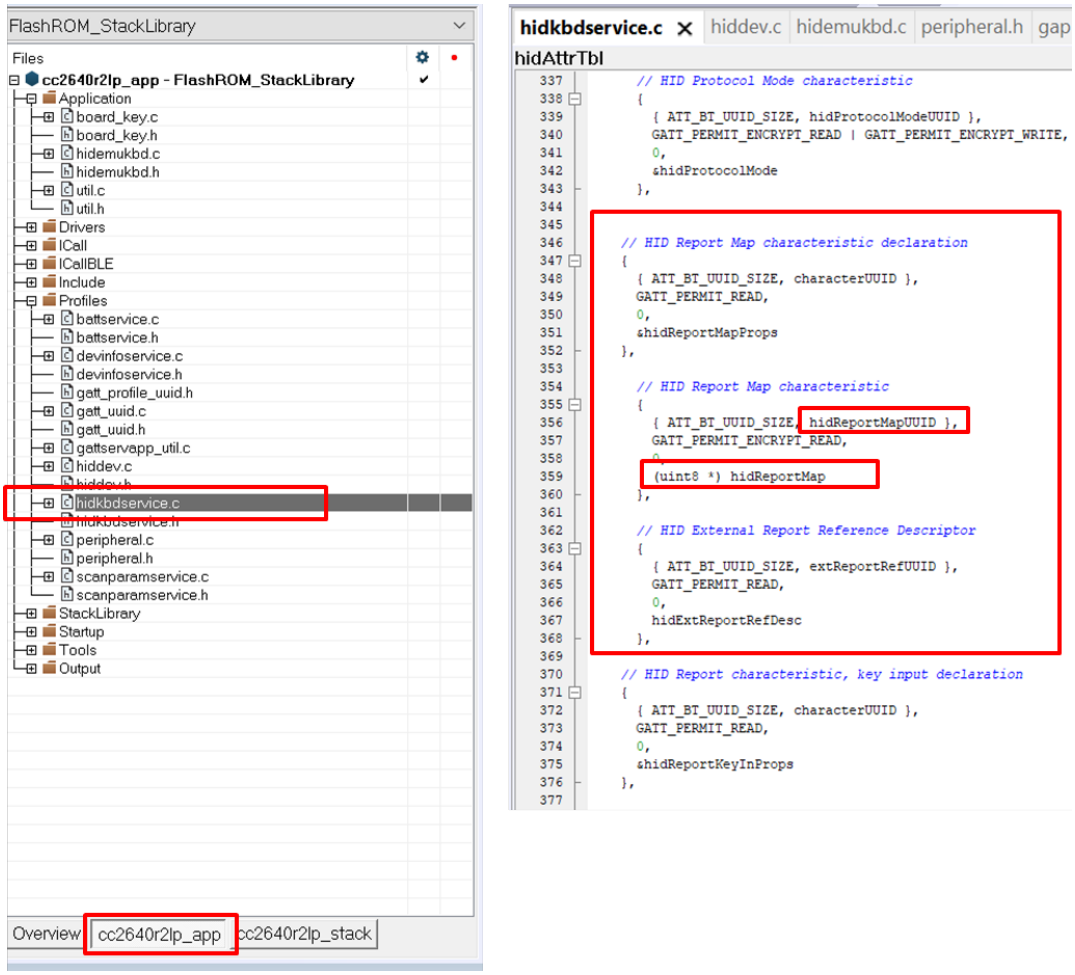


图4. hidkbdservice.c 中 HOGP 的 service 和特征值定义

HOGP 的 Report Map 特征值定义如下代码所示，代码中红色标注的就是前面提到过的 HOGP 所对应的 UUID 和对应的 Report Map 特征值的内容。需要特别注意的是，HOGP 必须是在链路已经加密的情况下进行数据交互，所以在 Report Map 特征值的属性是 GATT_PERMIT_ENCRYPT_READ。

```
1.  // HID Report Map characteristic declaration
2.  {
3.    { ATT_BT_UUID_SIZE, characterUUID },
4.    GATT_PERMIT_READ,
5.    0,
6.    &hidReportMapProps
7.  },
8.
9.    // HID Report Map characteristic
10.   {
11.     { ATT_BT_UUID_SIZE, hidReportMapUUID }, // Report Map UUID: 0x2A4B
```

```
    GATT_PERMIT_ENCRYPT_READ,               // 必须是在链路加密的情况下，由 HID 主机来读取 Report Map 的内容
    0,
    (uint8 *) hidReportMap                  // Report Map 的内容  hidReportMap
  },

  // HID External Report Reference Descriptor
  {
    { ATT_BT_UUID_SIZE, extReportRefUUID },
    GATT_PERMIT_READ,
    0,
    hidExtReportRefDesc
  },
```

Report Map 特征值的内容存储在 hidReportMap 数组中，Report Map 特征值将会以主机 Read 的方式发送至主机：

```
static CONST uint8 hidReportMap[] =
{
  0x05, 0x01,     // Usage Pg (Generic Desktop)
  0x09, 0x06,     // Usage (Keyboard)
  0xA1, 0x01,     // Collection: (Application)
                  //
                  // 224 到 231 是键盘上的功能按键值，包括 Ctrl, Shift，Alt，Gui 等总共 8 个。
  0x05, 0x07,     // Usage Pg (Key Codes)
  0x19, 0xE0,     // Usage Min (224)
  0x29, 0xE7,     // Usage Max (231)
  0x15, 0x00,     // Log Min (0)
  0x25, 0x01,     // Log Max (1)
                  //
                  // Modifier byte
                  // Input Report 第一个 Byte，这个 Byte 代表键盘上功能按键的开启标志位。
  0x75, 0x01,     // Report Size (1)
  0x95, 0x08,     // Report Count (8)
  0x81, 0x02,     // Input: (Data, Variable, Absolute)
                  //
                  // Reserved byte
                  // Input Report 第二个 Byte（size 是 1 个 bit，总共 8 个 bit），这是个保留 Byte。
  0x95, 0x01,     // Report Count (1)
  0x75, 0x08,     // Report Size (8)
  0x81, 0x01,     // Input: (Constant)
                  //
                  // LED report
                  // Output Report 总共一个字节，为 LED Report。前 5 个 bit 代表 LED 指示，值为 1 到 5。
  0x95, 0x05,     // Report Count (5)
  0x75, 0x01,     // Report Size (1)
  0x05, 0x08,     // Usage Pg (LEDs)
  0x19, 0x01,     // Usage Min (1)
  0x29, 0x05,     // Usage Max (5)
  0x91, 0x02,     // Output: (Data, Variable, Absolute)
                  //
                  // LED report padding
                  // Output Report 总共一个字节，前 5 个 bit 代表 LED 指示，后 3 个 bit 为保留位，不用。
  0x95, 0x01,     // Report Count (1)
  0x75, 0x03,     // Report Size (3)
  0x91, 0x01,     // Output: (Constant)
                  //
                  // Key arrays (6 bytes)
                  // Input Report 的第三个到第八个 Byte，总共 6 个，代表键盘的按键值。
```

```
    0x95, 0x06,     // Report Count (6)
    0x75, 0x08,     // Report Size (8)
    0x15, 0x00,     // Log Min (0)
    0x25, 0x65,     // Log Max (101)
    0x05, 0x07,     // Usage Pg (Key Codes)
    0x19, 0x00,     // Usage Min (0)
    0x29, 0x65,     // Usage Max (101)
    0x81, 0x00,     // Input: (Data, Array)
                    //
    0xC0            // End Collection
};
```

从 hidReportMap 数组的定义我们可以看到，Input Report 总共有 1（Modifier 功能按键）+1（保留字节）+6（键盘按键值）=8 个字节，这是一个用于按键的 Input Report。Output 共 1 个字节。对照 Hidemukbd.c 中 Input Report 发送函数，Input Report 发送格式如下：

```
static void HidEmuKbd_sendReport(uint8_t keycode)
{
  uint8_t buf[HID_KEYBOARD_IN_RPT_LEN];

  buf[0] = 0;         // Modifier keys
  buf[1] = 0;         // Reserved
  buf[2] = keycode;   // Keycode 1
  buf[3] = 0;         // Keycode 2
  buf[4] = 0;         // Keycode 3
  buf[5] = 0;         // Keycode 4
  buf[6] = 0;         // Keycode 5
  buf[7] = 0;         // Keycode 6

  HidDev_Report(HID_RPT_ID_KEY_IN, HID_REPORT_TYPE_INPUT,
                HID_KEYBOARD_IN_RPT_LEN, buf);
}
```

2.2 HID Service 中 Report 特征值的实现

Report 包括了 Input Report，Output Report，Feature Report，这三者都有对应的特征值在 HOGP 的 HID Service 中。在 hid_emu_kbd 工程中的 HOGP service 和特征值定义在 hidkbdservice.c 文件的 hidAttrTbl 数组里，和 Report Map 特征值一起定义。

有 1 个 Input Report 特征值：Key input。如何添加另一个重要 Input Report 到系统中会在后文中说明。

```
// HID Report characteristic, key input declaration
{
  { ATT_BT_UUID_SIZE, characterUUID },
  GATT_PERMIT_READ,
  0,
  &hidReportKeyInProps                    // 按键特征值的属性是 GATT_PROP_READ 和 GATT_PROP_NOTIFY，重点是 Notify
},

  // HID Report characteristic, key input
  {
    { ATT_BT_UUID_SIZE, hidReportUUID },    // Report UUID: 0x2A4D
    GATT_PERMIT_ENCRYPT_READ,
    0,
    &hidReportKeyIn                         // Key input report 的内容，就是 Report Map 中 8 个字节的那个 Input Report
  },
```

```
16.
17.   // HID Report characteristic client characteristic configuration
18.   {
19.     { ATT_BT_UUID_SIZE, clientCharCfgUUID },
20.     GATT_PERMIT_READ | GATT_PERMIT_ENCRYPT_WRITE,
21.     0,
22.     (uint8 *) &hidReportKeyInClientCharCfg  // Key input 的 CCC，notify 的使能
23.   },
24.
25.   // HID Report Reference characteristic descriptor, key input
26.   {
27.     { ATT_BT_UUID_SIZE, reportRefUUID },
28.     GATT_PERMIT_READ,
29.     0,
30.     hidReportRefKeyIn
31.   },
```

按键发送就是通过 Key input 特征值把 Report Map 里 8 字节的按键信息以 notify 的方式发送给主机。

2.3 在 HID 工程里里加入 Consumer Control Report

为了更好地演示和理解 BTool 的 HID 调试功能，在工程中额外添加了一个 HID 的 Consumer Control Report。Consumer Control Report 里包含了音量加减，播放暂停等功能，普遍用于 BLE 遥控器设计中。

2.3.1 Report Map 中添加 Consumer Control Report 信息

在 hidReportMap 数组的最后，添加包含 Consumer Control 的控制按键信息映射的代码。值得注意的是，这个 Input Report 的大小只有 1 个 Byte，Consumer Control Report 是以在数组中按顺序的 Index 方式发送给 HID 主机的，而不是直接发送键值。

```
1.   0x05, 0x0c,  // USAGE_PAGE (Consumer Devices)
2.   0x09, 0x01,  // USAGE (Consumer Control)
3.   0xa1, 0x01,  // COLLECTION (Application)
4.   0x85, 0x02,  // REPORT_ID (2)
5.                //
6.   //要添加的 consumer control 键值 0x09 开头的是 consumer 键值，可以按照实际需要的顺序排列
7.   0x09, 0x82,  //   USAGE (KEY_BOOKMARKS)
8.   0x09, 0xCD,  //   USAGE (Play/Pause)
9.   0x09, 0xB7,  //   USAGE (Stop)
10.  0x09, 0xB5,  //   USAGE (Skip track)
11.  0x09, 0xB6,  //   USAGE (Previous track)
12.  0x09, 0xB3,  //   USAGE (Fast forward)
13.  0x09, 0xB4,  //   USAGE (Rewind)
14.  0x09, 0xB2,  //   USAGE (Record)
15.  0x09, 0xE9,  //   USAGE (Volume up)
16.  0x09, 0xEA,  //   USAGE (Volume down)
17.  0x09, 0xE2,  //   USAGE (Mute)
18.  0x15, 0x01,  //   LOGICAL_MINIMUM (1)
19.  0x25, 0x0B,  //   LOGICAL_MAXIMUM (11)
20.               //   Input Report 只有一个 Byte，这个 Byte 代表上面定义的 Consumer Control 的按键序列。
21.  0x95, 0x01,  //   REPORT_COUNT (1)
22.  0x75, 0x08,  //   REPORT_SIZE (8)
23.  0x81, 0x00,  //   INPUT (Data,Ary,Abs)
24.               //
```

```
25.  0xC0         // END_COLLECTION
```

基于 Report Map 定义，在 hidemukbd.c 中定义 Consumer Control 按键的 Index，注意一定要和 Report Map 里定义的顺序一致。这些 Index 值才是 Consumer Control Report 发送的真正的值：

```
1.   // HID Consumer Control keycodes
2.   // (based on the HID Report Map characteristic value)
3.   #define HID_CC_RPT_POWER                  1
4.   #define HID_CC_RPT_PLAY_PAUSE             2
5.   #define HID_CC_RPT_STOP                   3
6.   #define HID_CC_RPT_SCAN_NEXT_TRK          4
7.   #define HID_CC_RPT_SCAN_PREV_TRK          5
8.   #define HID_CC_RPT_FAST_FWD               6
9.   #define HID_CC_RPT_REWIND                 7
10.  #define HID_CC_RPT_RECORD                 8
11.  #define HID_CC_RPT_VOLUME_UP              9
12.  #define HID_CC_RPT_VOLUME_DOWN            10
13.  #define HID_CC_RPT_MUTE                   11
```

2.3.2 HID Servcie 中添加 Consumer Control Report 需要的特征值

在 hidkbdservice.c 中添加特征值定义所需要的变量：

```
1.   // HID Report characteristic, consumer control input
2.   static uint8 hidReportCCInProps = GATT_PROP_READ | GATT_PROP_NOTIFY;
3.   static uint8 hidReportCCIn;
4.   static gattCharCfg_t *hidReportCCInClientCharCfg;
5.
6.   // HID Report Reference characteristic descriptor, consumer control input
7.   static uint8 hidReportRefCCIn[HID_REPORT_REF_LEN] =
8.               { HID_RPT_ID_CC_IN, HID_REPORT_TYPE_INPUT };
```

在 hidAttrTbl 数组最后添加如下代码：

```
1.   // HID Report characteristic declaration, consumer control
2.   {
3.     { ATT_BT_UUID_SIZE, characterUUID },
4.     GATT_PERMIT_READ,
5.     0,
6.     &hidReportCCInProps                  // 特征值的属性是 GATT_PROP_READ 和 GATT_PROP_NOTIFY，重点是 Notify
7.   },
8.
9.     // HID Report characteristic, consumer control
10.    {
11.      { ATT_BT_UUID_SIZE, hidReportUUID },
12.      GATT_PERMIT_ENCRYPT_READ,
13.      0,
14.      &hidReportCCIn                     // Consumer Control Report 的内容，1 个字节的 Input Report
15.    },
16.
17.    // HID Report characteristic client characteristic configuration, consumer control
18.    {
19.      { ATT_BT_UUID_SIZE, clientCharCfgUUID },
20.      GATT_PERMIT_READ | GATT_PERMIT_ENCRYPT_WRITE,
21.      0,
22.      (uint8 *) &hidReportCCInClientCharCfg  // Consumer Control Report 的 CCC，notify 的使能
```

```
  },

// HID Report Reference characteristic descriptor, consumer control
{
  { ATT_BT_UUID_SIZE, reportRefUUID },
  GATT_PERMIT_READ,
  0,
  hidReportRefCCIn
}
```

在 hidkbdservice.c 的全局 enum 数组的最后为 hidAttrTbl 添加 Consumer Control Report 的 Index：

```
HID_REPORT_CC_IN_DECL_IDX,      // HID Report characteristic declaration, consumer control
HID_REPORT_CC_IN_IDX,           // HID Report characteristic, consumer control
HID_REPORT_CC_IN_CCCD_IDX,      // HID Report characteristic client characteristic configuration, consumer control
HID_REPORT_REF_CC_IN_IDX       // HID Report Reference characteristic descriptor, consumer contrl
```

在 HidKbd_AddService()函数中，在其他 CCC 初始化最后添加 Consumer Control Report 的 CCC 初始化：

```
hidReportCCInClientCharCfg = (gattCharCfg_t *)ICall_malloc(sizeof(gattCharCfg_t) *
                                                           linkDBNumConns);
if (hidReportCCInClientCharCfg == NULL)
{
  ICall_free(hidReportKeyInClientCharCfg);

  ICall_free(hidReportBootKeyInClientCharCfg);

  ICall_free(hidReportBootMouseInClientCharCfg);

  return ( bleMemAllocError );
}

GATTServApp_InitCharCfg(INVALID_CONNHANDLE,
                        hidReportCCInClientCharCfg);
```

在该函数最后将 Consumer Control Report 初始化相应的信息添加到 hidRptMap 数组中。

```
hidRptMap[6].id = hidReportRefCCIn[0];
hidRptMap[6].type = hidReportRefCCIn[1];
hidRptMap[6].handle = hidAttrTbl[HID_REPORT_CC_IN_IDX].handle;
hidRptMap[6].pCccdAttr = &hidAttrTbl[HID_REPORT_CC_IN_CCCD_IDX];
hidRptMap[6].mode = HID_PROTOCOL_MODE_REPORT;
```

同时，应当把 hidRptMap 数组的长度修改为 8，为 Consumer Control Report 添加一个 ID，并和其他 Input Report 区分，此处定为 2。这些在 hidkbdservice.h 中定义：

```
// Number of HID reports defined in the service
#define HID_NUM_REPORTS          8//7 //在原来的基础上+1，很重要

// HID Report IDs for the service
#define HID_RPT_ID_KEY_IN        0  // Keyboard input report ID
#define HID_RPT_ID_MOUSE_IN      1  // Mouse input report ID
#define HID_RPT_ID_LED_OUT       0  // LED output report ID
#define HID_RPT_ID_FEATURE       0  // Feature report ID
```

```
9.
10.  //给个 ID，和其他 Input Report 区分开来
11.  #define HID_RPT_ID_CC_IN         2  // Consumer Control input report ID
```

这样就成功地在 HID Service 中添加了 Consumer Control Report。

2.3.3 按键发送代码的修改

为了调试方便，需要改动 Launchpad 对应的按键发送代码。在 hidemukbd.c 中，首先添加 Consumer Control Report 的按键发送函数，因为发送的是对应的 Index 值，Consumer Control Report 发送只需要一个字节。

```
1.   static void HidEmuKbdConCtrl_sendReport(uint8_t keycode)
2.   {
3.     uint8_t buf[1];
4.
5.     buf[0] = keycode;   // keycode
6.
7.     HidDev_Report(HID_RPT_ID_CC_IN, HID_REPORT_TYPE_INPUT,
8.                   HID_CC_IN_RPT_LEN, buf);
9.   }
```

按键处理函数 HidEmuKbd_handleKeys()的 KEY_LEFT 处理改为发送 Consumer Control 的信息，按键下按后发送的是 Volume Up 的命令，实际值对应 Report Map 里定义为 9。注意按键命令发送完毕后必须在发送一次释放命令，不然主机会以为按键一直没释放：

```
1.   if (keys & KEY_LEFT)
2.   {
3.     // Key Press.
4.     HidEmuKbdConCtrl_sendReport(HID_CC_RPT_VOLUME_UP);
5.
6.     // Key Release.
7.     // NB: releasing a key press will not propagate a signal to this function,
8.     // so a "key release" is reported immediately afterwards here.
9.     HidEmuKbdConCtrl_sendReport(KEY_NONE);
10.
11.  }
```

至此 hid_emu_kbd 工程里便有两个 Input Report，并且 Launchpad 的左键，定义为发送 Consumer Control 命令，右键定义为键盘 Key 发送命令，以此作为 Input Report 的调试。

3 BTool 工具简介和实验平台搭建

3.1 BTool 工具

BTool 是 TI 提供的一个强大的 PC 上位机工具，用作 BLE 调试，TI 的每一个 BLE SDK 都随带这个工具。BTool 在 SDK 目录的 tools 子目录下能找到。以 CC2640R2 的 SDK 为例，其位置在：*SDK 目录\tools\blestack\btool*。

- 主要功能

BTool 可以当作 BLE 的 central 设备，去调试 BLE 的 peripheral 外设设备。BTool 可以配合 CC2640R2F LaunchPad 来作为 central 设备，只需要 CC2640R2F LaunchPad 上烧录 SDK 中的 Host Test 工程，或者现成的已经编译好的 Host Test 工程的 hex，让它运行在能接受 TI 的 HCI vender specific 命令的 network processor 模式下。

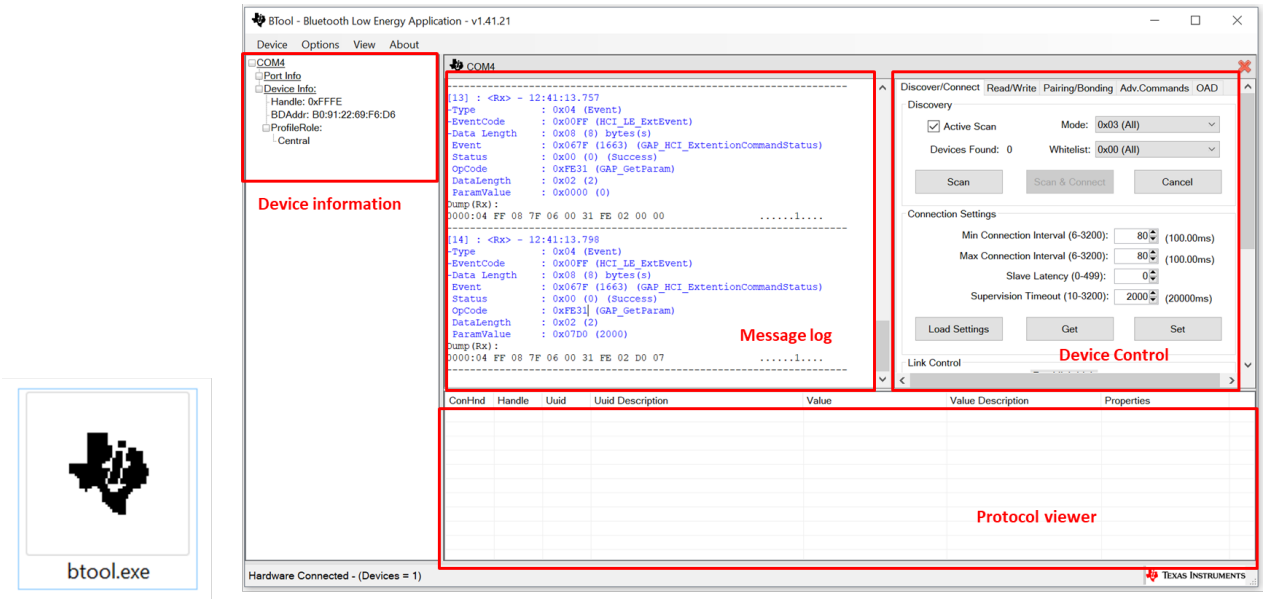


图5. BTool

- BTool 的连接

进行调试时，BTool 在 PC 上通过 UART 和 LaunchPad 连接。LaunchPad 的 Micro USB 除了用作调试，也是 USB 转 UART 工具，插上 PC 就能在设备管理器里看到对应 UART 端口，BTool 用的是 User UART 端口：

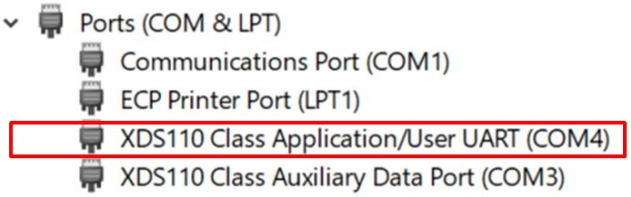


图6. UART 端口

完整连接调试框图如下，外设可以是 LaunchPad，也可以是具体其他 BLE 外设产品：

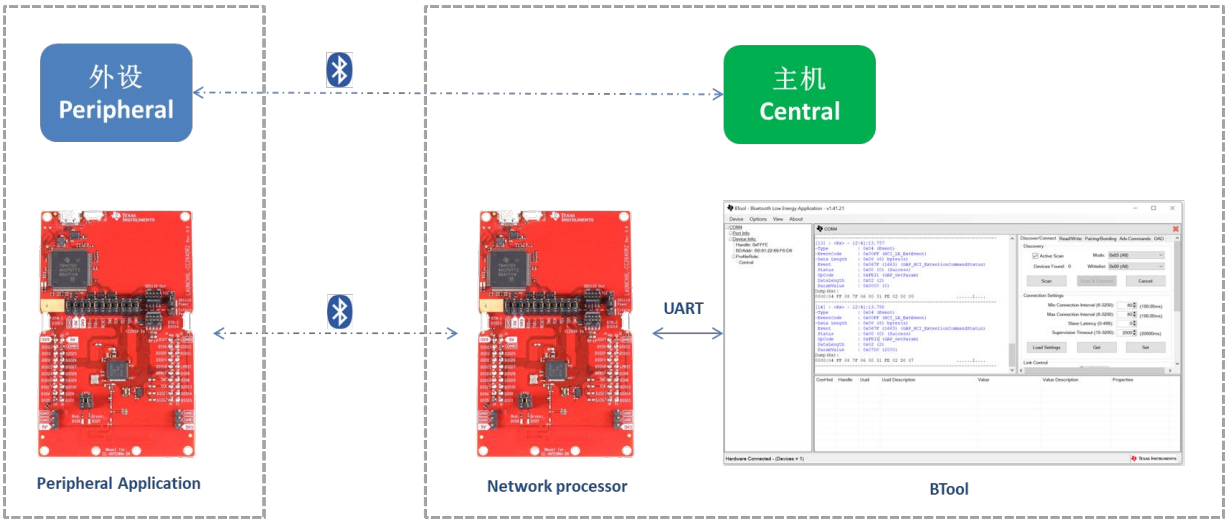


图7. BTool 连接方式

3.2 host_test 工程和 Network Processor 模式

Launchpad 需要烧录 host_test 固件才能和 BTool 联动。

host_test 固件可以通过编译 host_test 工程得到：

SDK 目录\ examples\rtos\CC2640R2_LAUNCHXL\blestack\host_test

也可以直接烧录 SDK 提供的.hex 文件：

SDK 目录\ examples\rtos\CC2640R2_LAUNCHXL\blestack\ hexfiles\ cc2640r2lp_host_test.hex

烧录好 host_test 固件之后，Launchpad 就处在 Network Processor 模式，这个时候就能和 BTool 通信了。打开 BTool：*SDK 目录\ tools\blestack\btool\btool.exe*，选择正确的 COM 口，就能看到如下图，Launchpad 的设备信息正常显示，表示 BTool 和 Launchpad 已经成功通信。

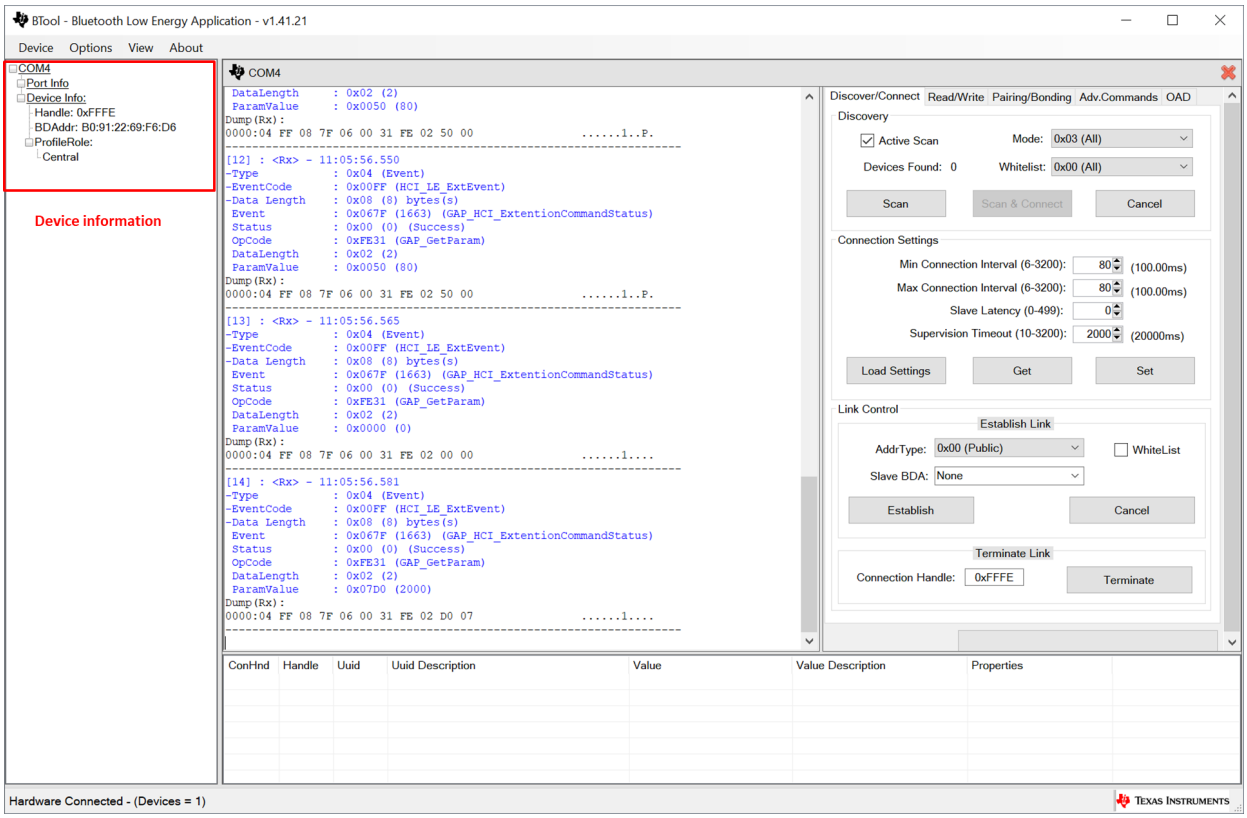


图8. BTool 连接 Launchpad

3.3 BTool 和 HID 外设的连接和配对加密

烧录前面改过的 hid_emu_kbd 工程到另外一块 CC2640R2F 的 Launchpad 上，用来作为 HID 外设。

需要注意的是 HOGP 通信必须是在链路加密的情况下，这是协议规定的，所以 BTool 和 HID 外设建立连接后，要进行调试的话必须进行链路加密。这部分 BTool 没法自动完成，必须进行手动操作。

3.3.1 HID 外设的连接

根据下图的步骤，查找连接 HID 外设。

- 选择 Discover/Connect 界面。
- 点击 Scan 查找周围设备。
- 找到需要的 HID 外设，选择对应的 MAC 地址，一般用 Public 地址，除非有特别设定。
- 点击 Establish 建立连接。
- 成功建立连接，显示连上设备的设备信息。

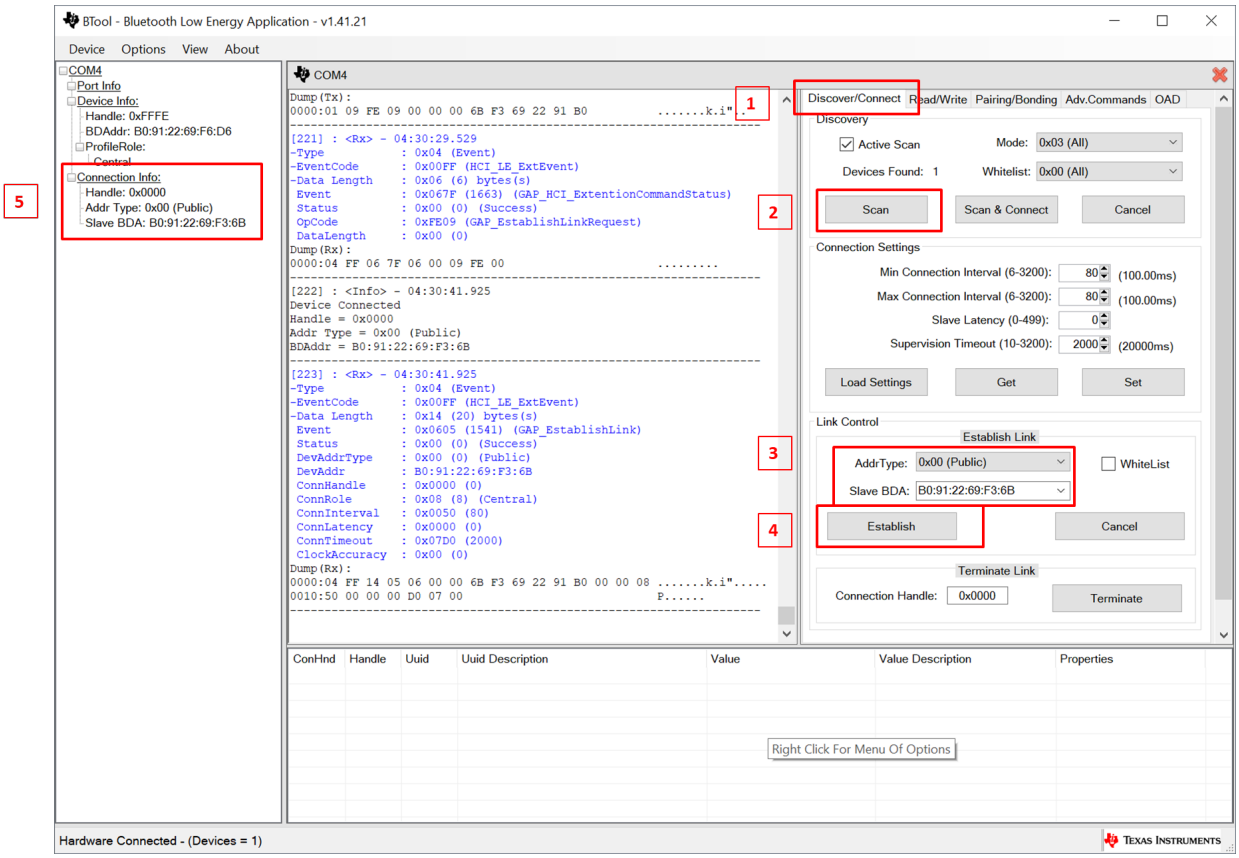


图9. BTool 查找并连上 HID 外设

3.3.2 配对和链路加密

协议规定 HOGP 通信必须是在加密的 BLE 链路上。所以必须进行配对和链路加密。

- 选择 Pairing/Bonding 界面。
- 点击 Send Pairing Request。注意这里其他选项并不会影响链路的加密。其中 Bonding Enabled 会使能绑定，下次连接的时候就不需要配对了，链路直接加密。MITM Enabled 就是使能 Man In The Middle 方式的配对。Connection Handle 对应的是 HID 外设，由于我们调试只连接一个设备，所以就用 0x0000。
- 稍等，就会出现 Long-Term Key 的显示，链路加密完成。中间对话框可以不用理会，是用 Long-Term Key 加密链路的选项。如果前面一步选择的是 Bonding Enable，那么第二次建立连接之后来到链路加密步骤就可以用这里的方法，导入第一次连接时候的 Long-Term Key，然后点击 Encrypt Link 链路就加密了。注意，我们调试可以不用 Bonding，Key 都是一次性的，所以不需要保存这里的 Long-Term Key，每次进来重新 Send Pairing Request 就行。

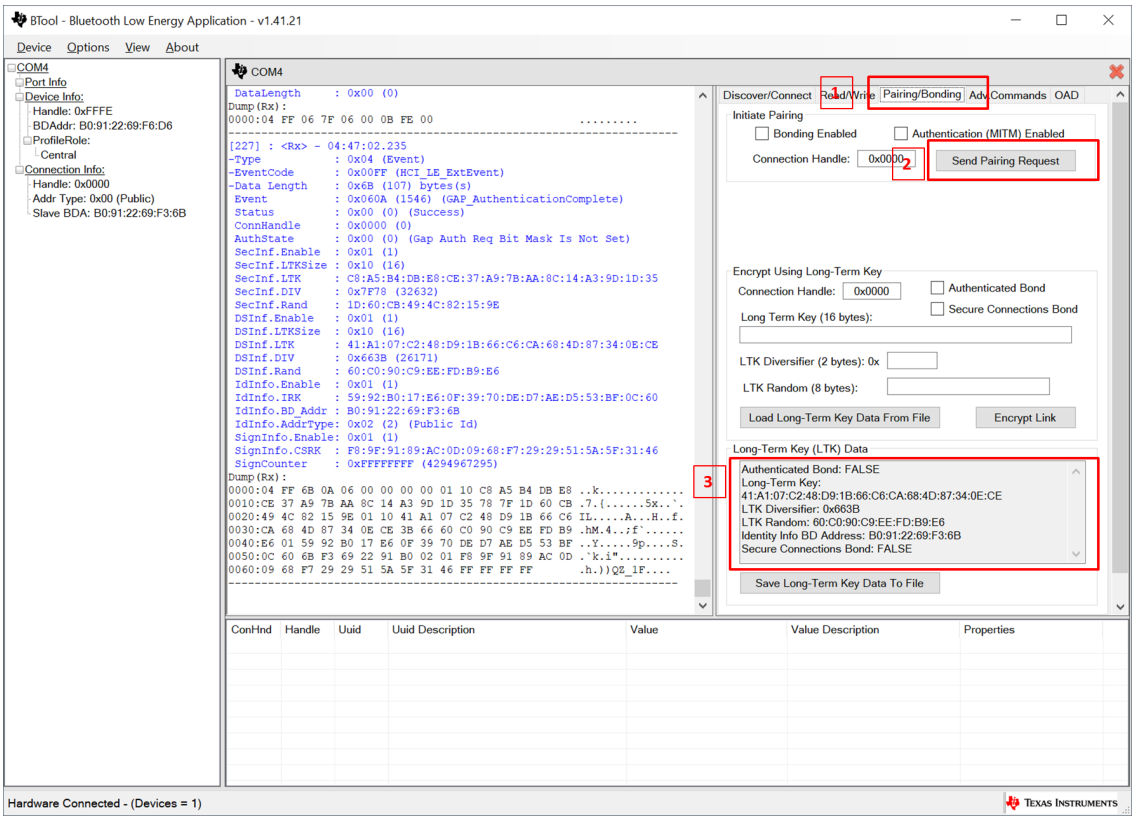


图10. HID 外设链路加密

至此，实验所需的调试平台就搭建完成了。

4 BTool 调试 HID Report

链路加密之后，就可以进行 HID 外设的功能调试，通过手动找到对应特征值的 handle 进行操作。

4.1 Report 特征值 handle 的查找

首先要做的就是找到所有的 HID Report 特征值的 handle。

- 选择 Read/Write 界面。这个界面是手动操作特征值非常全面的界面。
- 选择 Read Using Characteristic UUID 方式，并且在 Characteristic UUID 框中填入 4D:2A。前面已经介绍过，HOGP 的 Report UUID 的定义是 0x2A4D，包括了 Input Report，Output Report 和 Feature Report。小端在前，所以这里填入 4D:2A。
- 点击 Read，返回 Success。
- 在 BTool 最下面的 Protocol Viewer 窗口就会列出所有对应 Report 特征值对应的 handle，对这些 Report 的操作就得通过这些 handle。

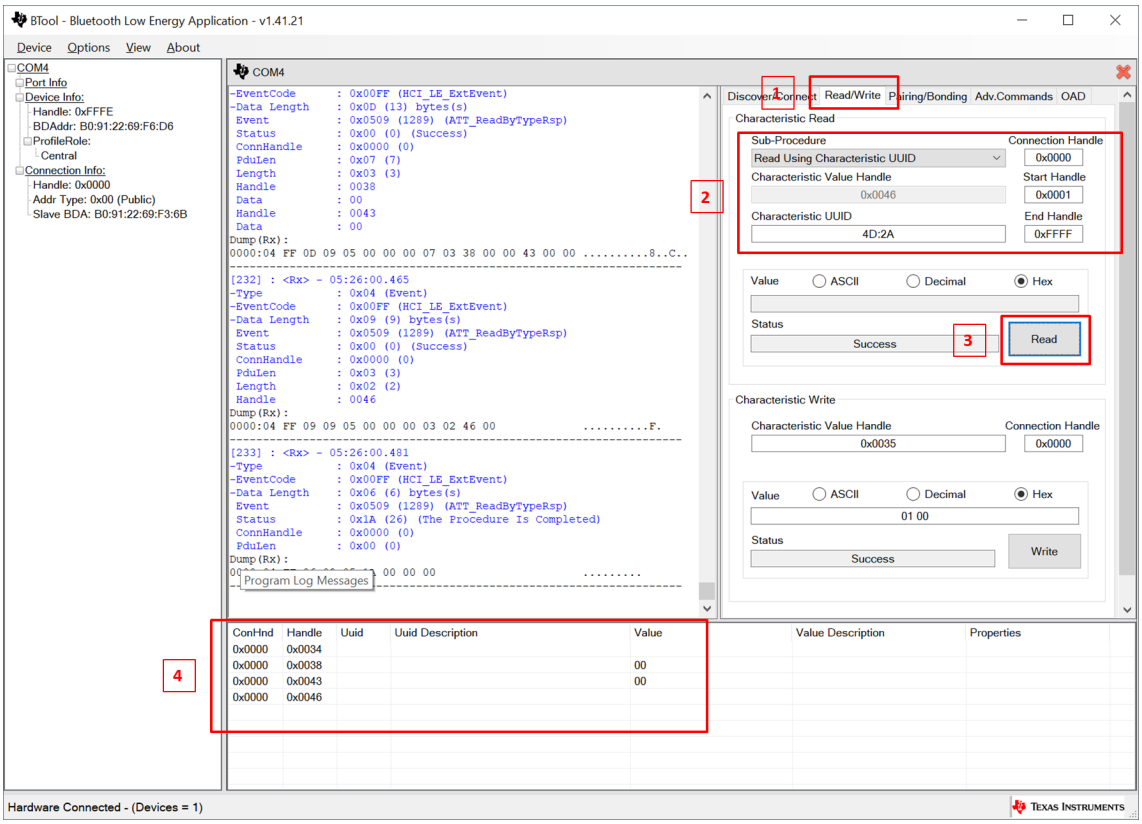


图11. Report 特征值的 handle 查找

从结果看，我们找到了 4 个 Report 特征值 handle。这都对应的是哪些 Report 呢？结合实际的 hid_emu_kbd 代码，服务和特征值的定义数组 hidAttrTbl，还有 Report Map 数组 hidReportMap 的定义我们知道我们依次定义了 Key Input，LED Output，Feature Report，另外额外增加了 Consumer Control Report，总共四个，所以这里读到的四个特征值 handle，就是依次对应着这四个 Report 特征值。

- Handle 0x0034： Key Input
- Handle 0x0038： LED Output
- Handle 0x0043： Feature Report
- Handle 0x0046： Consumer Control Report

其中原始的 Key Input 是个 Input Report，LED Output 是个 Output Report，Feature Report 就是 Feature Report。这就是为什么我们额外增加了 Consumer Control Report，因为这也是个 Input Report，这样我们在工程里就有 2 个 Input Report 了，可以更好演示如何用 BTool 查找和调试不同 Input Report。

4.2 Input Report 特征值的 notify 使能

工程里的两个 Input Report，都是以 notify 的形式发送按键信息到 HID 主机。BLE 的 notify 是需要通过主机端来使能的。Notify 的使能开关本身也是一个特征值，叫 Client Characteristic Configuration，简称 CCC，在协议定义中它是紧跟着它对应的特征值内容。所以它们的特征值 handle 也是紧跟着的。

- Key Input：0x0034，对应 CCC handle：0x0035
- Consumer Control Report：0x0046，对应 CCChandle：0x0047

LED Output 和 Feature Output 没有 notify 功能，不存在 CCC。

使能 Key Input 和 Consumer Control Report 的 Notify：

- Characteristic Value Hanlde 中填入 0x0035。
- Value 里填入 01:00，这是使能 CCC 的值。
- 点击 Write，等待 Success 返回。
- Characteristic Value Handle 改成 0x0047，重复上面步骤。

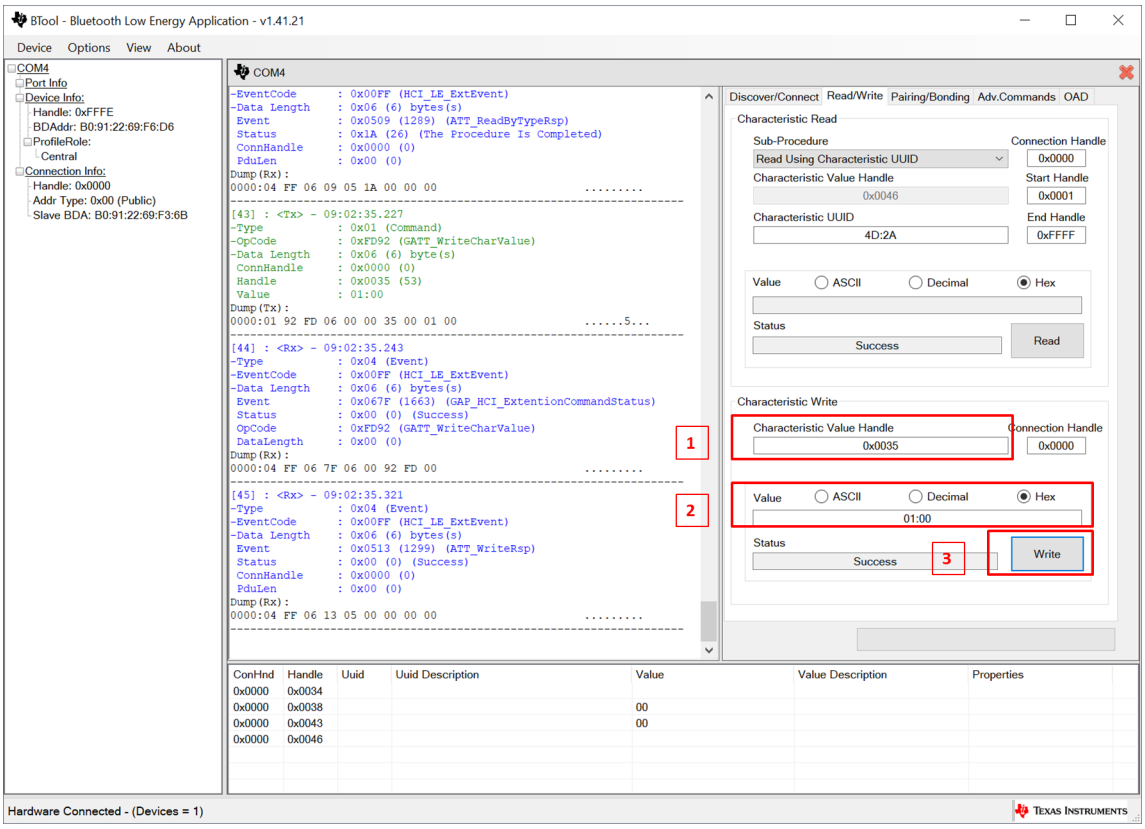


图12. Input Report 的 notify 使能

进行最后一步调试之前，需要确认 notify 功能已经成功打开，可以读一下两个 CCC 的值：

- 选择 Read Characteristic Value/Descriptor。
- Characteristic Value Handle 填入 0x0035
- 点击 Read，等待返回 Success。BTool 下面的窗口会显示 0x0035 对应的值，01:00 表示 CCC 值写入成功，notify 打开成功。
- Characteristic Value Handle 填入 0x0047，重复上面操作，检查一下 0x0047 对应的值是否是 01:00。

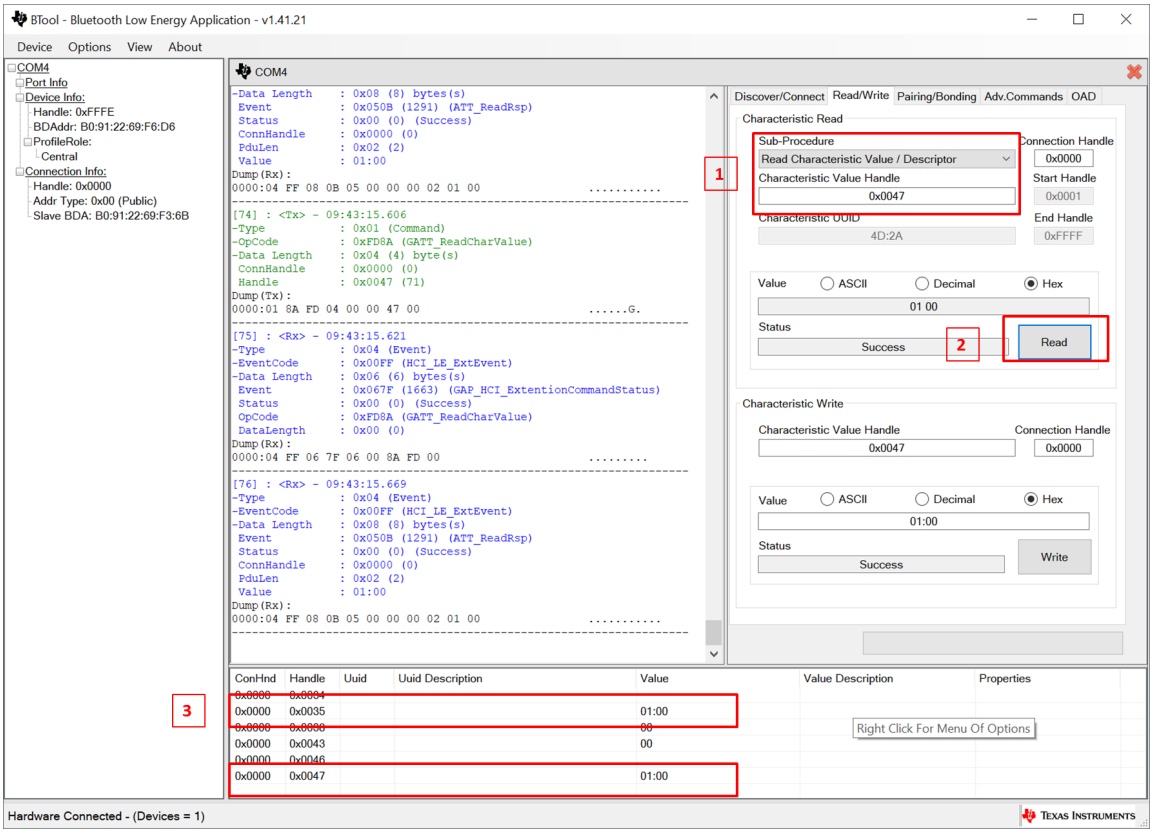


图13. notify 使能的检查

4.3 Input Report 的调试

工程里有两个 Input Report 了， Consumer Control Report 和 Input Key，首先来看一下 Input Key 调试：

- 按下 Launchpad 的右边按钮，然后释放。
- BTool 收到了两组 Notify 事件，每个事件包含 8 个字节，这是和前面的 Report Map 的定义完全吻合的，其中第 3 个字节是按键信息，第一组显示 0x4F，因为代码里让 Launchpad 的右变按钮模仿发送键盘的向右箭头按键，这个键值就似乎 0x4F。第二组的 notify 内容可以看到第 3 个字节就变成 0x00 了，这是因为 HID 协议规定按键发送完必须发送一个释放信号，这个 0x00 就代表前面 0x4F 的释放。

如果 HID 外设能让 BTool 显示如下，说明按键 Input Report 成功了。

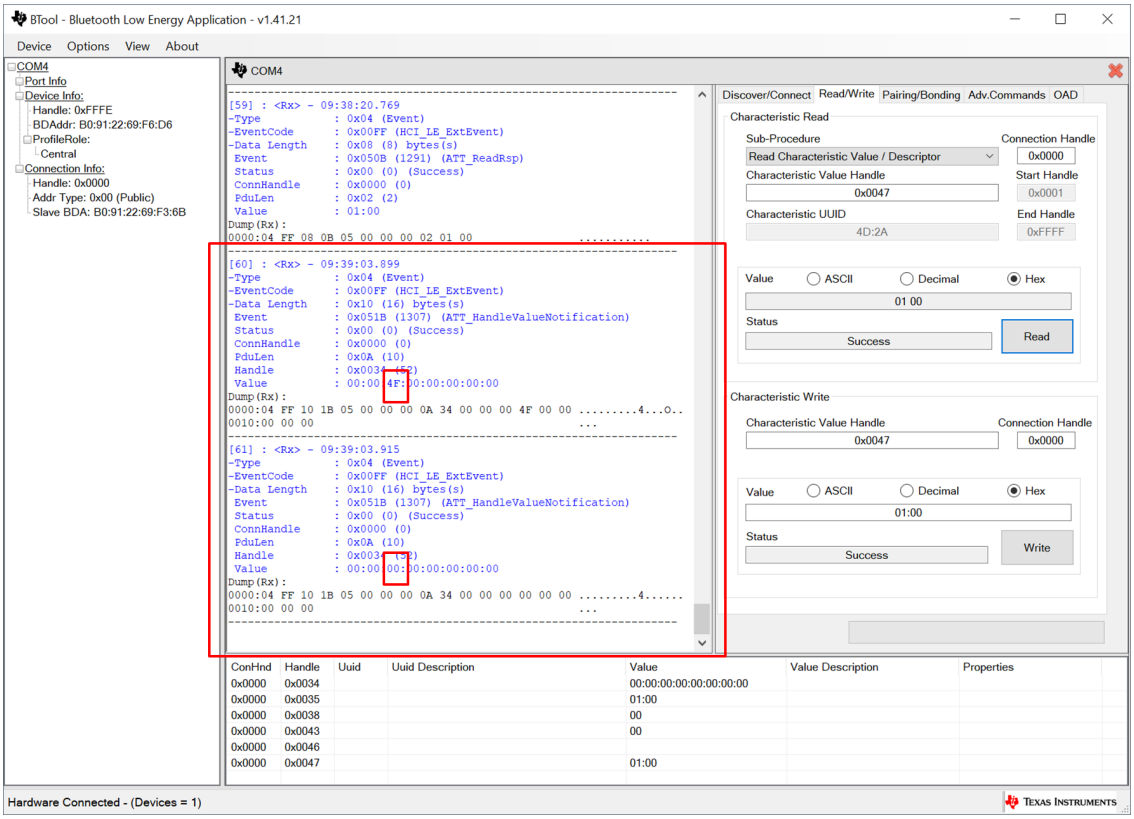


图14. Key input 调试

Consumer Control Report 的调试：

- 按下 Launchpad 的左边按钮，然后释放。
- 因为代码里把 Lauchpad 的左边按钮改成了触发 consumer Control 的 Volume Up 功能，对应的的 Index 是 9，所以 BTool 上会收到这个 Index 0x09，然后再接着一个代表释放的 0x00。

如果 HID 外设也能让 BTool 有这样的显示，那就说明按键 Consumer Control Report 成功了。

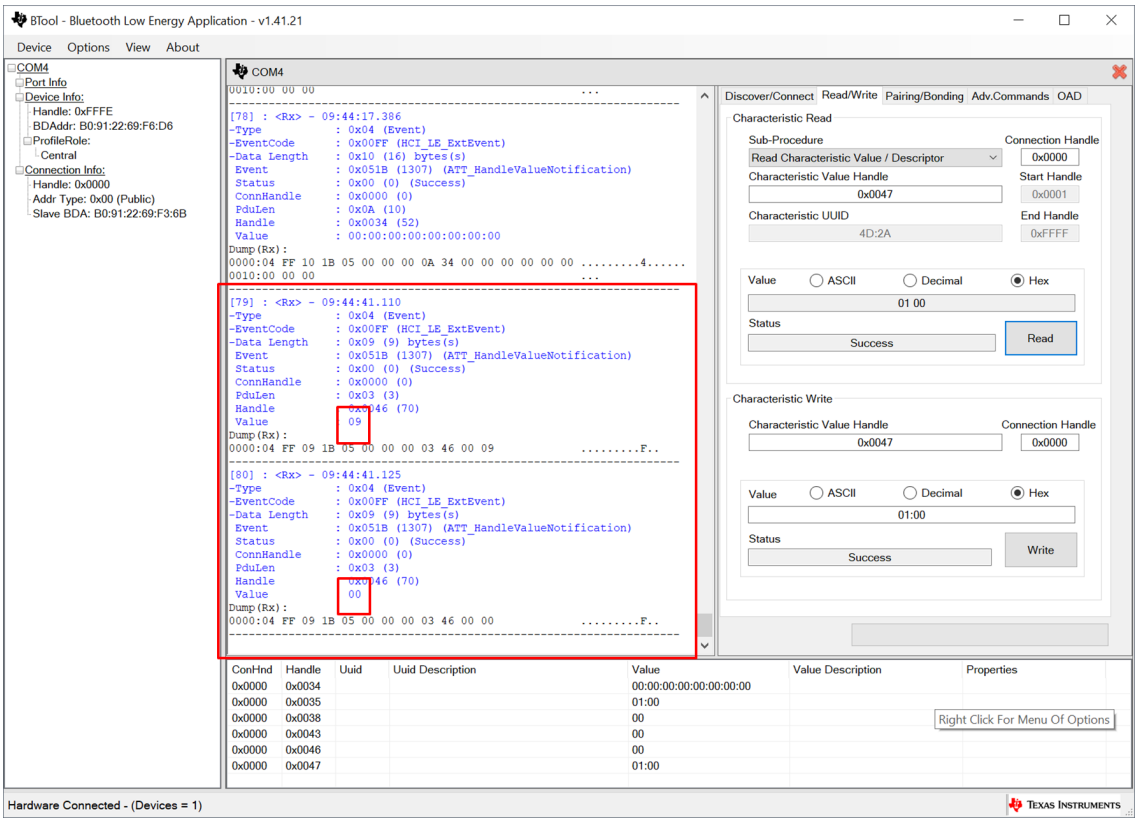


图15. Consumer Control Report 调试

到这里就基本了解了 Input Report 的调试方式，而且知道普通按键和 Consumer Control 的按键是不一样的形式发送的，普通按键是以直接按键值的方式发送，Consumer Control 则是以预先约定好的 Index 来发送。

4.4 Output Report 的调试

Output Report 则是以主机 Write 的方式向 HID 外设发送指令。比如这里的 LED Output，我们可以这样做：

- LED Output 特征值的 handle 是 0x0038 前面已知，填入 Characteristic Write 的 Characteristic Value Handle 。
- Value 填入 01。
- 点击 Write，等待 Success。

这样 Output Report 的过程就结束了，可以查看一下是否发送成功：

- 选择 Read Characteristic Value/Descriptor，填入 handle 值 0x0038。
- 点击 Read，等待 Success。
- BTool 返回前面写入的 0x01 值，表示成功。

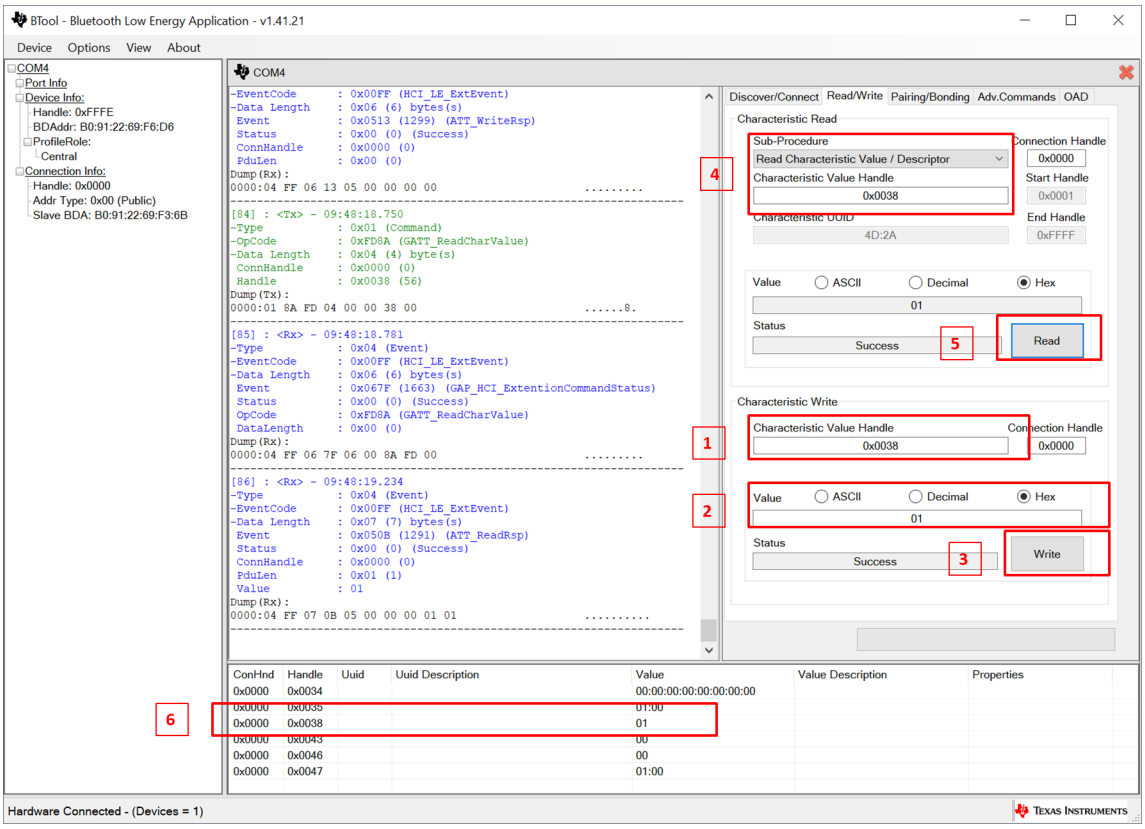


图16. Output Report 调试

4.5 Feature Report 的调试

Feature Report 的调试和 Output Report 类似，由于 Feature Report 也是可读可写，因此也能通过先写后读的方式判断其功能是否正常。

- 前面已知这个工程的 Feature Report 特征值 handle 是 0x0043，将其填入 Characteristic Write 的 Characteristic Value Handle 。
- Value 我们填入 0x06，也可以是其他。
- 点击 Write，等待 Success。

读取一下 Feature Report：

- 选择 Read Characteristic Value/Descriptor，填入 handle 值 0x0043。
- 点击 Read，等待 Success。
- BTool 返回前面写入的 0x06 值，表示成功。

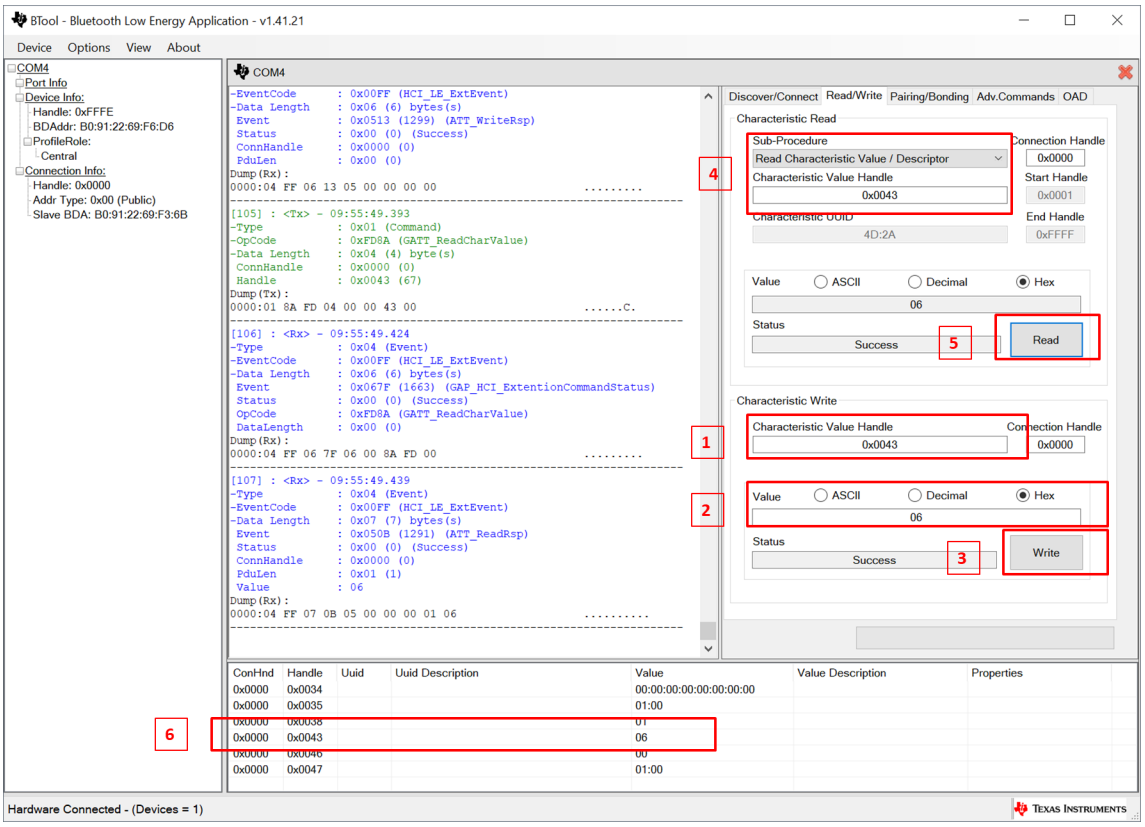


图17. Feature Report 调试

5 参考文献

1. *SimpleLink CC2640R2F BLE SOC 资源.*
2. *simplelink_cc2640r2_sdk_ble_example_pack_1_50_00_62, hid_emu_kbd Project.*
3. *SIG BLE HID Service specification.*
4. *SIG BLE HOGP (HID Over GATT Profile) specification.*
5. *SIG BLE GATT HID Report Map Definition.*
6. *SIG BLE GATT HID Report Definition.*
7. *USB HID information.*

重要声明和免责声明

TI 均以“原样”提供技术性及可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证其中不含任何瑕疵，且不做任何明示或暗示的担保，包括但不限于对适销性、适合某特定用途或不侵犯任何第三方知识产权的暗示担保。

所述资源可供专业开发人员应用TI 产品进行设计使用。您将对以下行为独自承担全部责任：(1) 针对您的应用选择合适的TI 产品；(2) 设计、验证并测试您的应用；(3) 确保您的应用满足相应标准以及任何其他安全、安保或其他要求。所述资源如有变更，恕不另行通知。TI 对您使用所述资源的授权仅限于开发资源所涉及TI 产品的相关应用。除此之外不得复制或展示所述资源，也不提供其它TI或任何第三方的知识产权授权许可。如因使用所述资源而产生任何索赔、赔偿、成本、损失及债务等，TI对此概不负责，并且您须赔偿由此对TI 及其代表造成的损害。

TI 所提供产品均受TI 的销售条款 (http://www.ti.com.cn/zh-cn/legal/termsofsale.html) 以及ti.com.cn上或随附TI产品提供的其他可适用条款的约束。TI提供所述资源并不扩展或以其他方式更改TI 针对TI 产品所发布的可适用的担保范围或担保免责声明。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122