User's Guide

CC256x TI 蓝牙协议栈 HIDDemo 应用

TEXAS INSTRUMENTS

摘要

内容

商标

1 演示概述

人机接口设备支持主机连接和控制 HID 设备。此配置文件中定义了两种角色：主机和设备。主机发送控制和报告请求，第二个是响应主机请求的设备。主机是计算机或平板电脑等设备，而设备是键盘或鼠标等 I/O 设备。

此应用允许用户使用支持低功耗蓝牙 (BLE) 的控制台在两个 BLE 设备之间建立连接，控制设备并获取报告和协议。

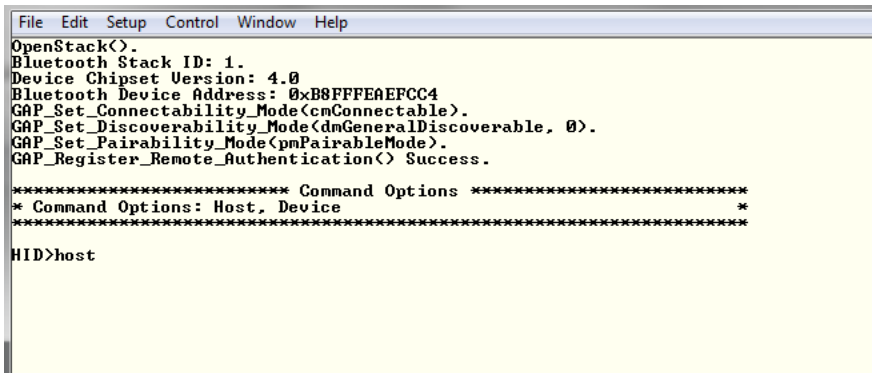
尝试本页介绍的应用之前，建议用户先访问基于 MSP432 MCU 的 ®TI 双模 Bluetooth™ 堆栈或基于 STM32F4 MCU 的双模 Bluetooth® 协议栈页面。

备注

可以使用相同的指令在 Tiva、MSP432 或 STM32F4 平台上运行此演示。

2 运行蓝牙代码

刷写代码后，使用 miniUSB 或 microUSB 电缆将电路板连接至 PC。连接后，等待安装驱动程序。这在设备管理器的“端口（COM 和 LPT）”下显示为 MSP-EXP430F5438 USB 串行端口 (COM x)、Tiva 虚拟 COM 端口 (COM x)、适用于 MSP432 的 XDS110 类应用程序/用户 UART (COM x)。将一个诸如 PuTTY 的终端程序连接到电路板的串行端口 x。要使用的串行参数为 115200 波特（MSP430 为 9600 波特）、8、n、1。连接后，使用“Reset S3”按钮（位于 MSP430 的 Mini USB 连接器旁边），并观察终端上的堆栈初始化情况。此时显示帮助屏幕，其中显示了所有命令。

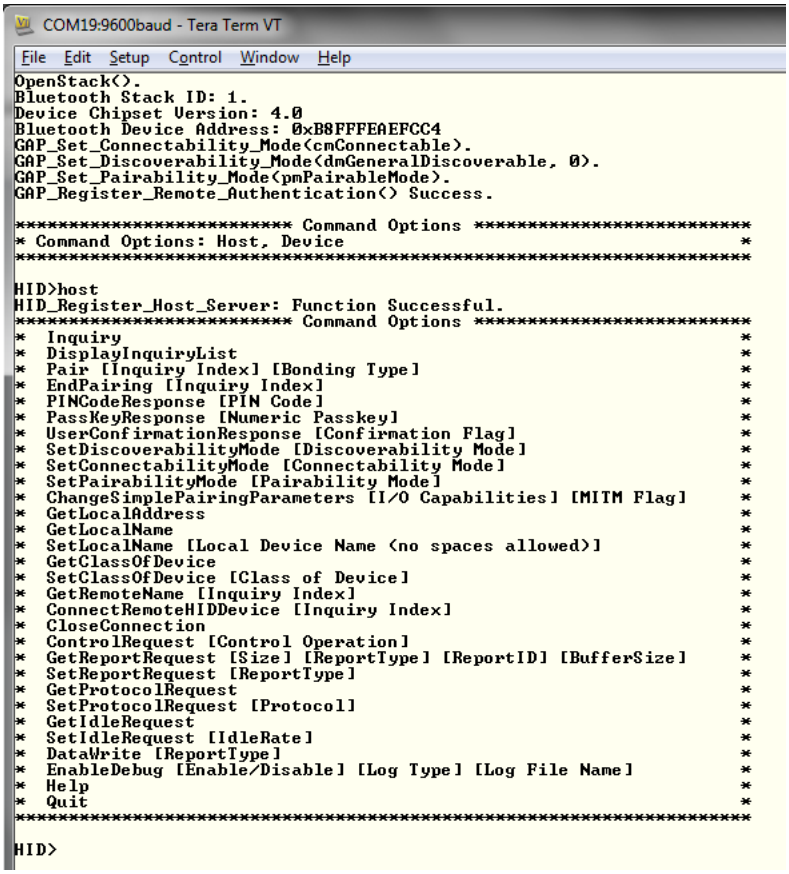


3 演示应用程序

演示应用介绍了如何使用演示应用连接两个已配置的电路板并通过 BluetoothLE 进行通信。当对协议栈进行初始化时，所包含的应用程序会在电路板上注册一项自定义服务。

3.1 演示应用中的设备 1（主机/HID 主机）设置

1. 将第一块电路板设置为主机。执行前面“运行蓝牙代码”一节中提到的步骤来初始化应用程序。初始化后，请记下服务器的蓝牙地址。稍后会用它从设备或客户端发起连接。
2. 在“Choose mode >”提示符处，输入“Host”。
3. 观察此时主机的所有可能命令的列表。在“Host >”提示符处输入“Help”，即可随时查看该列表。

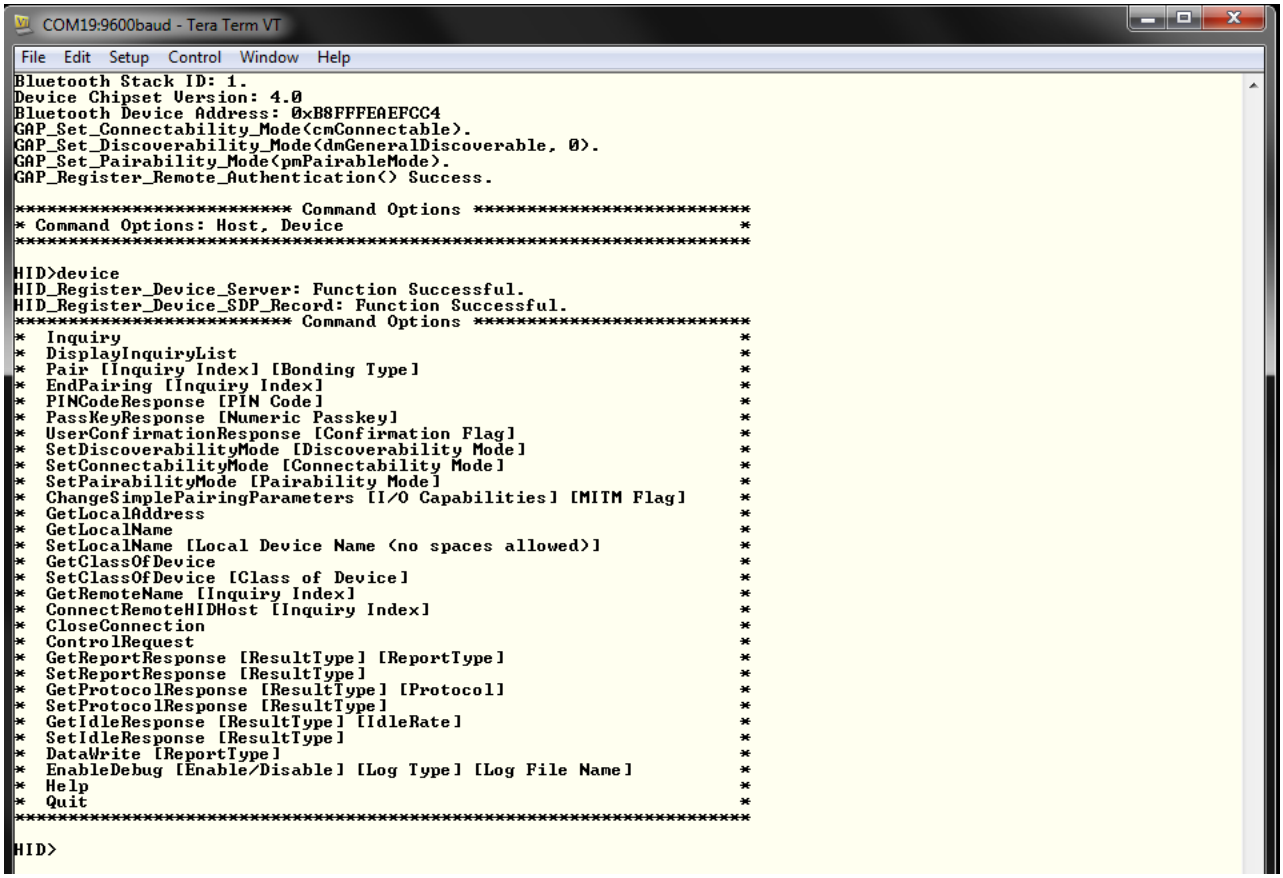


4. 在“Host >”提示符处，输入“Inquiry”。这将启动查询流程。完成后，观察所有已发现设备的列表。
5. 在“Host >”提示符处选择“DisplayInquiryList”，即可随时查看该列表。

```
HID>inquiry
Return Value is 0 GAP_Perform_Inquiry() SUCCESS.

HID>
GAP Inquiry Entry Result: 0xB8FFFEA92617.

HID>
GAP Inquiry Entry Result: 0x000272212389.

HID>
GAP Inquiry Entry Result: 0x30144A39DDB7.

HID>
GAP Inquiry Entry Result: 0x0007808031B7.

HID>
GAP Inquiry Entry Result: 0xBC0DA5F8D90E.

HID>
GAP Inquiry Entry Result: 0x0002724614C5.

HID>
GAP Inquiry Entry Result: 0x0007808031D1.

HID>
GAP Inquiry Entry Result: 0x00190E05483B.

HID>
GAP Inquiry Entry Result: 0xCDABDEADBEEF.

HID>
GAP Inquiry Entry Result: 0x001638392F2F.

HID>
GAP Inquiry Entry Result: 0x00190E0D47A3.

HID>
GAP Inquiry Entry Result: 0x7C8EE4001E72.

HID>
GAP Inquiry Entry Result: 0x000272D69F2D.

HID>
GAP Inquiry Entry Result: 0xB8FFFEAEFCC4.

HID>
GAP Inquiry Entry Result: 0x000272D69F2E.

HID>
GAP Inquiry Entry Result: 0x10BF48ED2C24.

HID>
GAP Inquiry Entry Result: 0x000272D69F33.

HID>
GAP Inquiry Entry Result: 0x000272D6A6E2.

HID>
GAP Inquiry Entry Result: 0x000780803205.

HID>
GAP Inquiry Entry Result: 0x000272D6A563.

HID>
GAP_Inquiry_Result: 20 Found.
GAP Inquiry Result: 1, 0xB8FFFEA92617.
GAP Inquiry Result: 2, 0x000272212389.
GAP Inquiry Result: 3, 0x30144A39DDB7.
GAP Inquiry Result: 4, 0x0007808031B7.
GAP Inquiry Result: 5, 0xBC0DA5F8D90E.
GAP Inquiry Result: 6, 0x0002724614C5.
GAP Inquiry Result: 7, 0x0007808031D1.
GAP Inquiry Result: 8, 0x00190E05483B.
GAP Inquiry Result: 9, 0xCDABDEADBEEF.
GAP Inquiry Result: 10, 0x001638392F2F.
GAP Inquiry Result: 11, 0x00190E0D47A3.
GAP Inquiry Result: 12, 0x7C8EE4001E72.
GAP Inquiry Result: 13, 0x000272D69F2D.
GAP Inquiry Result: 14, 0xB8FFFEAEFCC4.
GAP Inquiry Result: 15, 0x000272D69F2E.
```

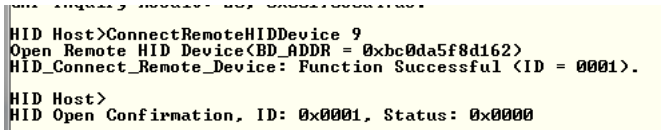
3.2 演示应用上的设备 2（客户端/HID 设备）设置

1. 将第二块电路板设置为设备。执行前面运行蓝牙代码一节中提到的步骤来初始化应用程序。在“选择模式 >”提示符处，输入“Device”。
2. 观察此时设备的所有可能命令的列表。在“Device >”提示符处输入“Help”，即可随时查看该列表。

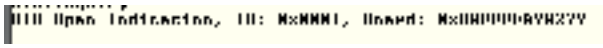


3.3 从 HID 主机发起连接

1. 记下配置为 HID 设备的第二块电路板的索引号。[如果屏幕上未显示列表，请在客户端上发出 DisplayInquiryList 命令以显示已发现设备的列表。]
2. 从设备发出 ConnectRemoteHIDDevice <Inquiry Index> 命令。
3. 等待 HID Open 确认。



4. 当客户端成功连接到服务器时，服务器会看到打开指示。



备注

首次连接时，可以从 HID 主机设备发起连接。否则，连接会被拒绝，因为 HID 主机不允许来自未知 HID 设备的连接。

备注

状态 0x0000 表示连接成功。任何其他状态均表示连接未成功。

3.4 从 HID 设备发起连接

从 HID 设备发起连接的过程与从主机发起连接的过程相同，只是我们会在 HID 设备上运行查询（如上文设备 1 的步骤 3 所述），并发出 ConnectRemoteHIDHost <Inquiry Index>，其中 Inquiry Index 是运行查询时对应于主机 BD-ADDR 的编号。

```
HID>ConnectRemoteHIDHost 12
Open Remote HID Host(BD_ADDR = 0xB8FFFEAEFCC4)
HID_Connect_Remote_Host: Function Successful (ID = 0001).

HID>
HID Open Confirmation, ID: 0x0001, Status: 0x0000
```

3.5 主机与设备之间的通信

1. 建立连接后，主机与设备就可以相互通信了。
2. 我们可以使用 ControlRequest 命令从主机或设备发送控制操作。对于主机，我们发出 Controlrequest < parameter-number > 命令。控制请求的选项包括 0= hcNop、1= hcHardReset、2= hcSoftReset、3= hcSuspend、4= hcExitSuspend、5= hcVirtualCableUnplug。当我们在主机上键入 Control Request 5 时，会收到以下消息：

```
HID>ControlRequest 5
HID_Control_Request: Function Successful.

HID>
HID Close Indication, ID: 0x0002
```

在设备侧，将显示 hcVirtualCableUnplug 指示。

```
HID>
HID Control Indication, ID: 0x0001, Control Operation: hcVirtualCableUnplug

HID>
HID Close Indication, ID: 0x0001
```

对于没有参数的 DeviceControlRequest，默认情况下会执行 hcVirtualCableUnplug。

```
HID>ControlRequest
HID_Control_Request: Function Successful.

HID>
HID Close Indication, ID: 0x0001
```

在主机侧，将显示 hcVirtualCableUnplug 指示。

```
HID>
HID Control Indication, ID: 0x0003, Control Operation: hcVirtualCableUnplug

HID>
HID Close Indication, ID: 0x0003
```

3. 通过发出 GetReportRequest 命令来发出报告请求。这需要 3 个或 4 个参数。第一个是 Size，0 表示使用报告大小，1 表示使用自定义缓冲区大小。第二个是 ReportType，0 表示 rtOther，1 表示 rtInput，2 表示 rtOutput，3 表示 rtFeature。第三个是 ReportID。如果在第一个参数中使用了自定义缓冲区大小，请在此处指定大小。例如，发送具有报告大小且 rtInput 和 ReportID 为 2 的报告请求。观察报告请求成功指示。

```
HID>GetReportRequest 0 1 2
HID_Get_Report_Request: Function Successful.
```

在设备上，观察报告指示，包括报告类型、ID、大小和缓冲区大小。

```
HID>
HID Get Report Indication, ID: 0x0003, ReportType: rtInput, ReportID: 2, Size: grSizeOfReport, BufferSize: 0
```

设备可以使用 GetReportResponse 响应 GetReportRequest。这需要 ResultType (0 表示 rtSuccessful，1 表示 rtNotReady，2 表示 rtErrInvalidReportID，3 表示 forrtErrUnsupportedRequest，4 表示 rtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData) 和 ReportType (0 表示 rtOther，1 表示 rtInput，2 表示 rtOutput，3 表示 rtFeature) 作为参数。例如，使用 rtData 作为 ResultType，rtInput 作为 ReportType，响应主机发出的上述 rtInput 请求。

```
HID>SetReportResponse 7 1
HID Set Report Response: Function Successful.
```

主机将收到报告确认。

```
HID>
HID Get Report Confirmation, ID: 0x0004, Status: rtData, ReportType: rtInput, ReportLength: 4,
Report: 0x02 0x80 0x50 0x00
```

4. 从主机运行 SetReportRequest。所需的唯一参数是 ReportType，0 表示 rtOther，1 表示 rtInput，2 表示 rtOutput，3 表示 rtFeature。

```
HID>SetReportRequest 1
HID_Set_Report_Request: Function Successful.
```

设备将收到具有 ReportType 的 SetReportIndication。

```
HID>
HID Set Report Indication, ID: 0x0004, ReportType: rtInput, ReportLength: 4,
Report: 0x02 0x80 0x50 0x00
```

设备可通过发出 SetReportResponse 命令来响应 SetReportRequest。唯一需要的参数是 ResultType（0 表示 rtSuccessful，1 表示 rtNotReady，2 表示 forrtErrInvalidReportID，3 表示 rtErrUnsupportedRequest，4 表示 rtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData）。例如，使用 rtSuccessful 响应上述 rtInputReport 请求。

```
HID>SetReportResponse 0
HID_Set_Report_Response: Function Successful.
```

主机会收到具有 ResultType 的 SetReportConfirmation 指示。

```
HID>
HID Set Report Confirmation, ID: 0x0004, Status: rtSuccessful
```

5. 使用 GetProtocolRequest 发送协议请求。这不需要任何参数。

```
HID>GetProtocolRequest
HID_Get_Protocol_Request: Function Successful.
```

设备会收到协议指示。

```
HID>
HID Get Protocol Indication, ID: 0x0003
```

设备可通过发出 GetProtocolResponse 命令来响应协议请求。这需要两个参数：ResultType（0 表示 rtSuccessful，1 表示 rtNotReady，2 表示 forrtErrInvalidReportID，3 表示 rtErrUnsupportedRequest，4 表示 rtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData）和 Protocol（0 表示 ptBoot，1 表示 forptReport）。例如，响应上一个具有 rtData 和 ptBoot 的请求。

```
HID>GetProtocolResponse 7 0
HID_Get_Protocol_Response: Function Successful.

HID>
```

主机会收到具有 ResultType 和 Protocol 的协议确认。

```
HID>
HID Get Protocol Confirmation, ID: 0x0001, Status: rtData, Protocol: ptBoot
```

6. 从主机发出 SetProtocolRequest。唯一需要的参数是 Protocol（0 表示 ptBoot，1 表示 ptReport）。例如，发送一个具有 ptReport 的请求。

```
HID>SetProtocolResponse 0
HID_Set_Protocol_Response: Function Successful.
```

主机会收到设置协议指示和协议。这可通过发出 SetProtocolResponse 命令来响应，该命令需要 ResultType 作为参数（0 表示 forrtSuccessful，1 表示 rtNotReady，2 表示 rtErrInvalidReportID，3 表示 rtErrUnsupportedRequest，4 表示 rtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData）。

```
HID>
HID Set Protocol Indication, ID: 0x0001, Protocol: ptReport

HID>SetProtocolResponse 0
HID_Set_Protocol_Response: Function Successful.
```

主机中具有 ResultType 的协议确认。

7. 设置发出 GetIdleRequest 命令的空闲请求。这不需要任何参数。

```
HID>GetIdleRequest
HID_Get_Idle_Request: Function Successful.
```

设备会收到 GetIdleIndication。

```
HID>
HID Get Idle Indication, ID: 0x0001
```

这可通过 GetIdleResponse 来响应，需要 ResultType（0 表示 rtSuccessful，1 表示 rtNotReady，2 表示 rtErrInvalidReportID，3 表示 rtErrUnsupportedRequest，4 表示 forrtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData）和 Idle Rate 作为参数。例如，响应时，ResultType 为 rtData，Idle Rate 为 50。

```
HID>GetIdleResponse 7 50
HID_Get_Idle_Response: Function Successful.
```

主机会收到空闲确认。

```
HID>
HID Get Idle Confirmation, ID: 0x0002, Status: rtData, IdleRate: 50
```

8. 使用主机发出的 SetIdleRequest 设置 Idle Rate，该请求需要将 Idle Rate 作为唯一参数。例如，我们在这里将 Idle Rate 设置为 50。

```
HID>SetIdleRequest 50
HID_Set_Idle_Request: Function Successful.
```

设备会收到设置空闲指示。这可使用“设置空闲响应”来响应，需要 ResultType（0 表示 rtSuccessful，1 表示 rtNotReady，2 表示 rtErrInvalidReportID，3 表示 forrtErrUnsupportedRequest，4 表示 rtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData）作为一个参数。

```
HID>
HID Set Idle Indication, ID: 0x0001, IdleRate: 50

HID>SetIdleResponse 0
HID_Set_Idle_Response: Function Successful.
```

主机会收到具有 ResultType 的 SetIdleConfirmation。

```
HID>
HID Set Idle Confirmation, ID: 0x0002, Status: rtSuccessful
```

9. 使用 DataWrite 在设备之间写入数据。需要的一个参数是 ReportType（0 表示 rtOther，1 表示 rtInput，2 表示 rtOutput，3 表示 rtFeature）。在以下示例中，从设备写入数据。

```
HID>DataWrite 1
HID_Data_Write: Function Successful.
```

主机会收到数据指示。

```
HID>
HID Data Indication, ID: 0x0002, ReportType: rtInput, ReportLength: 4,
Report: 0x02 0x80 0x50 0x00
```

4 应用程序命令

使用 GetProtocolRequest 发送协议请求。这不需要任何参数。设备收到协议指示。设备可通过发出 GetProtocolRequest 命令来响应协议请求。这需要两个参数，一个是 Result Type（0 表示 rtSuccessful，1 表示 rtNotReady，2 表示 rtErrInvalidReportID，3 表示 rtErrUnsupportedRequest，4 表示 rtErrInvalidParameter，5 表示 rtErrUnknown，6 表示 rtErrFatal，7 表示 rtData），另一个是 Protocol（0 表示 ptBoot，1 表示 ptReport）。在下面的示例中，对上一个请求的响应是 rtData 和 ptBoot。主机会收到一份协议确认，其中包含结果类型和协议。

有关该应用程序和其他应用程序的概述，请访问基于 MSP432™ MCU 的 TI 双模 Bluetooth® Stack 和基于 STM32F4 MCU 的双模 Bluetooth® Stack。

本页描述了应用程序用户可以使用的各种命令。每个命令都是 TI Bluetooth Stack API 的包装器，可使用用户选择的参数进行调用。这是用户可用的 API 的子集。TI 的 Bluetooth Stack API 文档（即 TI_Bluetooth_Stack_Version-Number\Documentation，对于 STM32F4，为 TI_Bluetooth_Stack_Version-Number\RTOS_VERSION\Documentation）详细介绍了所有 API。

5 Gap 命令

通用访问配置文件定义了与发现和连接蓝牙设备相关的标准过程。这定义了所有设备通用的运行模式，并考虑了使用这些模式来决定设备如何与其他蓝牙设备交互的过程。可发现性、可连接性、可配对性、可绑定模式和安全模式都可以使用通用访问配置文件过程进行更改。所有这些模式都会影响两个设备之间的交互。GAP 还定义了有关如何绑定两个蓝牙设备的过程。

5.1 帮助 (DisplayHelp)

说明

Help 命令负责显示串行端口客户端或串行端口服务器的当前命令选项。此函数的输入参数被完全忽略，只需要传入即可，因为可以在提示符下输入的所有命令都会传入解析的信息。此命令显示当前可用的命令选项，并且始终返回零。

参数

使用此命令时不需要包含参数。参数对命令的结果没有影响。

可能的返回值

该命令始终返回 0。

5.2 查询

说明

Inquiry 命令负责执行一般查询以发现蓝牙设备。此命令要求在运行之前存在有效的蓝牙协议栈 ID。如果调用成功，此命令将返回零；如果执行期间发生错误，此命令将返回负值。查询会持续 10 秒 — 除非在该时间限制之前找到 20 个设备 (MAX_INQUIRY_RESULTS)。

参数

使用此命令时不需要包含参数。参数对查询的结果没有影响。

可能的返回值

- (0) 成功完成查询过程
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- (-58) BTPS_ERROR_INVALID_MODE

API 调用

GAP_Perform_Inquiry(BluetoothStackID, itGeneralInquiry, 0, 0, 10, MAX_INQUIRY_RESULTS, GAP_Event_Callback, (unsigned long) NULL)

API 原型

int BTPSAPI GAP_Perform_Inquiry(unsigned int BluetoothStackID, GAP_Inquiry_Type_t GAP_Inquiry_Type, unsigned int MinimumPeriodLength, unsigned intMaximumPeriodLength, unsigned int InquiryLength, unsigned int MaximumResponses, GAP_Event_Callback_t GAP_Event_Callback, unsigned long CallbackParameter)

API 说明

提供此函数是为了能够启动查询扫描过程。此函数的第一个参数是要执行查询的蓝牙设备的蓝牙协议栈。第二个参数是要执行的查询类型。第三个参数是最小周期长度，第四个参数是最大周期长度，以秒为单位 (仅在执行定

期查询的情况下有效）。第五个参数是执行查询的时间长度，以秒为单位指定。第六个参数是要等待的响应数。最后两个参数表示在完成指定查询后要调用的回调函数（和参数）。如果成功，此函数返回零；如果无法执行查询，则返回负的错误代码。在任何给定时间只能执行一次查询。如果正在进行未完成的查询，则调用此函数将会失败。调用方可以调用 GAP_Cancel_Inquiry() 函数来取消当前正在执行的查询过程。

最小和最大查询参数是可选参数，如果指定，则表示最小和最大定期查询周期。如果使用简单查询过程（非定期），则被调用方必须将这两个值均设置为零。如果指定了这两个参数，则必须满足以下条件：

MaximumPeriodLength > MinimumPeriodLength > InquiryLength

5.3 显示查询列表

说明

存在 DisplayInquiryList 命令，用于显示带有索引的当前查询列表。当用户忘记了要与之交互的特定蓝牙设备的查询索引时，此命令非常有用。此函数在成功执行时返回零，而在出现任何错误时返回负值。该命令要求在运行之前存在有效的蓝牙协议栈 ID，并在使用 Inquiry 命令后调用，因为列表为空，尚未发现设备。

参数

使用此命令时不需要包含参数。参数对显示的查询列表的结果没有影响。

可能的返回值

- (0) 成功显示查询列表
- (-8) INVALID_STACK_ID_ERROR

5.4 配对

说明

Pair 命令负责启动与远程蓝牙设备的绑定。此函数在成功执行时返回零，而在出现任何错误时返回负值。在尝试配对之前，必须存在蓝牙协议栈 ID，并且该设备不能事先连接到任何设备（包括该设备尝试配对的设备）。请注意，要连接到远程设备，必须在调用 Pair 之前使用 Inquiry 命令。通用和专用绑定均受支持。

参数

Pair 命令需要一个或两个具有特定值的参数才能成功运行。第一个参数是远程蓝牙设备的查询索引。此参数始终是必需的。该值可以在查询后找到，或者在使用命令 DisplayInquiryList 时显示。如果所需的远程设备未出现在列表中，则无法进行配对。第二个参数是用于配对过程的绑定类型。这是一个可选参数，仅在连接需要通用绑定时才需要此参数。必须将值指定为 0（表示专用绑定）或 1（表示通用绑定）。如果仅提供一个参数，则绑定类型为专用绑定。

命令调用示例

- “Pair 5 0”尝试使用专用绑定与第五个查询索引处的远程设备进行配对。
- “Pair 5”操作与上述示例完全相同。如果没有参数，则绑定类型为专用。
- “Pair 8 1”尝试使用通用绑定与第八个查询索引处的远程设备进行配对。

可能的返回值

- (0) 配对成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-59) BTPS_ERROR_ADDING_CALLBACK_INFORMATION
- (-8) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Initiate_Bonding(BluetoothStackID, InquiryResultList[(TempParam->Params[0].intParam - 1)], BondingType, GAP_Event_Callback, (unsigned long)0)

API 原型

int BTPSAPI GAP_Initiate_Bonding(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Bonding_Type_t GAP_Bonding_Type, GAP_Event_Callback_tGAP_Event_Callback, unsigned long CallbackParameter)

API 说明

提供此函数是为了能够启动绑定过程。此函数可以根据请求的绑定类型执行通用绑定和专用绑定。该函数将以下内容作为输入：本地蓝牙设备的蓝牙协议栈 ID（用于执行绑定）、要绑定的设备的远程蓝牙地址、要执行的绑定类型，以及 GAP 事件回调信息（用于处理此函数成功时将产生的身份验证事件）。如果该函数成功，则会通过注册的 GAP 事件回调返回所有进一步的信息。请注意，如果此函数成功返回结果，并不意味着远程设备已成功与本地设备绑定，仅意味着远程设备绑定过程已启动。仅当不存在与指定远程蓝牙设备的物理 DisplayInquiryList 配对连接时，此函数才会成功。此函数将连接到蓝牙设备并开始绑定过程。

如果指定了通用绑定并维持了链路，则在调用 GAP_End_Bonding 函数之前不会终止连接。这允许执行在同一物理链路上需要的任何更高级别的初始化。

如果执行了专用绑定，则在身份验证过程完成后，链路会自动终止。由于此过程的异步性质，指定的 GAP 事件回调将向调用方通知身份验证过程中出现的任何事件和/或数据。可随时调用 GAP_Cancel_Bonding 函数以结束绑定过程并终止链路（无论执行的是哪种绑定方法）。使用通用绑定时，如果通过此函数启动的蓝牙链路建立了 L2CAP 连接，则在发出 L2CAPDisconnect 请求（或响应）时，蓝牙协议栈可能会也可能不会终止物理链路。如果发生这种情况，则调用 GAP_End_Bonding 函数不起作用（在这种情况下，GAP_End_Bonding 函数会返回一个错误代码）。

5.5 结束配对

说明

EndPairing 命令负责结束先前启动的与远程设备的绑定会话。此函数在成功执行时返回零，而在出现任何错误时返回负值。尝试结束配对之前，必须存在蓝牙协议栈 ID，并且该设备必须已连接至一个远程设备。请注意，要断开与远程设备的连接，必须在调用 EndPairing 之前使用 Pair 和 Inquiry 命令。

参数

EndPairing 命令需要一个参数，即远程蓝牙设备的查询索引。该值可以在查询后找到，或者在使用命令 DisplayInquiryList 时显示。除非配对后调用了新的查询，否则该值与 Pair 命令中使用的第一个参数的值相同。如果是这种情况，请找到 Pair 命令中使用的设备的蓝牙地址。

命令调用示例

- “EndPairing 5”尝试结束与第五个查询索引处的远程设备的配对。
- “EndPairing 8”尝试结束与第八个查询索引处的远程设备的配对。

可能的返回值

- (0) 成功结束配对
- (-2)BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1)BTPS_ERROR_INVALID_PARAMETER
- (-58)BTPS_ERROR_INVALID_MODE
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR

API 调用

GAP_End_Bonding(BluetoothStackID, InquiryResultList[(TempParam->Params[0].intParam – 1)])

API 原型

int BTPSAPI GAP_Initiate_Bonding(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Bonding_Type_t GAP_Bonding_Type, GAP_Event_Callback_tGAP_Event_Callback, unsigned long CallbackParameter)

API 说明

提供此函数是为了能够终止通过调用 GAP_Initiate_Bonding 函数（指定通用绑定作为待执行的绑定类型）建立的连接。如果使用专用绑定启动了绑定过程（或设备已断开连接），则此函数不起作用。此函数使用指定要绑定（通用绑定）的远程蓝牙设备的地址。此函数将终止已建立的 ACL 连接。不会对原始 GAP_Initiate_Bonding 函数调用中指定的 GAP 事件回调发出 GAP 事件回调（如果此函数成功返回）。

5.6 PIN 码响应

说明

PINCodeResponse 命令负责发出 GAP 身份验证响应，通过输入参数指定其 PIN 码值。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。该设备还必须正在进行由本地设备或远程设备启动的配对操作。

参数

PINCodeResponse 命令需要一个参数，即用于对连接进行身份验证的 PIN 码。这是一个字符串值，最长可达 16 位数字。配对发起方查看在配对过程中显示的消息来调用此命令。在发起方输入 PIN 码后，响应方会收到一条调用此命令的消息。

命令调用示例

- “PINCodeResponse 1234”尝试将 PIN 码设置为“1234”。
- “PINCodeResponse 5921302312564542”尝试将 PIN 码设置为“5921302312564542”。该值表示最长的 16 位 PIN 码值。

可能的返回值

- (0) 成功的 PIN 码响应
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Authentication_Response(BluetoothStackID, CurrentRemoteBD_ADDR, &GAP_Authentication_Information)

API 原型

*int BTPSAPI GAP_Authentication_Response(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Authentication_Information_t *GAP_Authentication_Information)*

API 说明

提供此函数是为了能够让本地设备响应 GAP 身份验证事件。此函数用于为指定的蓝牙设备指定身份验证信息。该函数将以下内容作为输入：已请求身份验证操作的蓝牙设备的蓝牙协议栈 ID 以及身份验证响应信息（由调用方指定）。

5.7 通行密钥响应

说明

PassKeyResponse 命令负责发出 GAP 身份验证响应，通过输入参数指定其通行密钥值。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。该设备还必须正在进行由本地设备或远程设备启动的配对操作。

参数

PassKeyResponse 命令需要一个参数，即用于对连接进行身份验证的通行密钥。这是一个字符串值，最长可达 6 位数字（值介于 0 和 999999 之间）。

命令调用示例

- “PassKeyResponse 1234”尝试将通行密钥设置为“1234”。
- “PassKeyResponse 999999”尝试将通行密钥设置为“999999”。该值表示最长的 6 位通行密钥值。

可能的返回值

- (0) 成功的通行密钥响应
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Authentication_Response(BluetoothStackID, CurrentRemoteBD_ADDR, &GAP_Authentication_Information)

API 原型

int BTPSAPI GAP_Authentication_Response(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Authentication_Information_t *GAP_Authentication_Information)

API 说明

提供此函数是为了能够让本地设备响应 GAP 身份验证事件。此函数用于为指定的蓝牙设备指定身份验证信息。该函数将以下内容作为输入：已请求身份验证操作的蓝牙设备的蓝牙协议栈 ID 以及身份验证响应信息（由调用方指定）。

5.8 用户确认响应

说明

UserConfirmationResponse 命令负责发出 GAP 身份验证响应，通过输入参数指定其用户确认值。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。该设备还必须正在进行由本地设备或远程设备启动的配对操作。

参数

UserConfirmationResponse 命令需要一个参数，即用于对连接进行身份验证的用户确认值。这是一个整数值，必须为 1（以确认连接）或为 0（以不确认身份验证并停止配对过程）。

命令调用示例

- “UserConfirmationResponse 0”尝试拒绝与远程蓝牙设备建立的连接并取消身份验证过程。
- “UserConfirmationResponse 1”尝试接受与远程蓝牙设备建立的连接并确认身份验证过程。

可能的返回值

- (0) 成功的用户确认响应
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Authentication_Response(BluetoothStackID, CurrentRemoteBD_ADDR, &GAP_Authentication_Information)

API 原型

*int BTPSAPI GAP_Authentication_Response(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Authentication_Information_t *GAP_Authentication_Information)*

API 说明

提供此函数是为了能够让本地设备响应 GAP 身份验证事件。此函数用于为指定的蓝牙设备指定身份验证信息。该函数将以下内容作为输入：已请求身份验证操作的蓝牙设备的蓝牙协议栈 ID 以及身份验证响应信息（由调用方指定）。

5.9 设置可发现性模式

说明

SetDiscoverabilityMode 命令负责设置本地设备的可发现性模式。此命令在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。如果将设备设置为“有限可发现”，则该设备的可发现时间为 60 秒；而“一般可发现”设备始终是可发现的。

参数

此命令只需要一个表示可发现性模式的整数值参数。该值必须指定为 0（表示不可发现模式）、1（表示有限可发现模式）或 2（表示一般可发现模式）。

命令调用示例

- “SetDiscoverabilityMode 0”尝试将本地设备的可发现性模式更改为不可发现。
- “SetDiscoverabilityMode 1”尝试将本地设备的可发现性模式更改为有限可发现。
- “SetDiscoverabilityMode 2”尝试将本地设备的可发现性模式更改为一般可发现。

可能的返回值

- (0) 成功设置可发现性模式
- (-4) FUNCTION_ERROR

- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-5) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-58) BTPS_ERROR_INVALID_MODE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- (-64) BTPS_ERROR_INTERNAL_ERROR
- (-1) BTPS_ERROR_INVALID_PARAMETER

API 调用

GAP_Set_Discoverability_Mode(BluetoothStackID, DiscoverabilityMode, (DiscoverabilityMode == dmLimitedDiscoverableMode)?60:0)

API 原型

int BTPSAPI GAP_Set_Discoverability_Mode(unsigned int BluetoothStackID, GAP_Discoverability_Mode_t GAP_Discoverability_Mode, unsigned int Max_Discoverable_Time)

API 说明

提供此函数是为了设置由蓝牙协议栈 ID 指定的蓝牙协议栈所指定的本地蓝牙设备的可发现性模式。第二个参数指定要将本地蓝牙设备置于的可发现性模式，第三个参数指定要将本地蓝牙设备置于指定的可发现模式的时间长度（以秒为单位）（如果未指定为不可发现模式）。在这段时间结束时（假设时间不是无限的），本地蓝牙设备将返回到不可发现模式。

5.10 设置可连接性模式

说明

SetConnectabilityMode 命令负责设置本地设备的可连接性模式。此命令在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

此命令只需要一个表示可发现性模式的整数值参数。该值必须指定为 0（表示非连接）或 1（表示可连接）。

命令调用示例

- “SetConnectabilityMode 0”尝试将本地设备的可连接性模式设置为“非连接”。
- “SetConnectabilityMode 1”尝试将本地设备的可连接性模式设置为“可连接”。

可能的返回值

- (0) 成功设置可连接性模式
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-5) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-58) BTPS_ERROR_INVALID_MODE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Set_Connectability_Mode(BluetoothStackID, ConnectableMode)

API 原型

int BTPSAPI GAP_Set_Connectability_Mode(unsigned int BluetoothStackID, GAP_Connectability_Mode_t GAP_Connectability_Mode)

API 说明

提供此函数是为了设置由蓝牙协议栈 ID 指定的蓝牙协议栈所指定的本地蓝牙设备的可连接性模式。第二个参数指定要将本地蓝牙设备置于的可连接性模式。

5.11 设置可配对性模式

说明

SetPairabilityMode 命令负责设置本地设备的可配对性模式。此命令在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

此命令只需要一个表示可配对性模式的整数值参数。该值必须指定为 0（表示不可配对）、1（表示可配对）或 2（表示安全简易配对）。

命令调用示例

- “SetPairabilityMode 0”尝试将本地设备的可配对性模式设置为不可配对。
- “SetPairabilityMode 1”尝试将本地设备的可配对性模式设置为可配对。
- “SetPairabilityMode 2”尝试将本地设备的可配对性模式设置为安全简易配对。

可能的返回值

- (0) 成功设置可配对性模式
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-5) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-58) BTPS_ERROR_INVALID_MODE

API 调用

GAP_Set_Pairability_Mode(BluetoothStackID, PairabilityMode)

API 原型

int BTPSAPI GAP_Set_Pairability_Mode(unsigned int BluetoothStackID, GAP_Pairability_Mode_t GAP_Pairability_Mode)

API 说明

提供此函数是为了设置本地蓝牙设备的可配对性模式。第二个参数指定要将本地蓝牙设备置于的可配对性模式。如果指定了安全简易配对 (SSP) 配对模式，则*必须*将 SSP 用于所有配对操作。在此之后，可将设备置于不可配对模式，但如果重新启用配对，则*必须*在启用 SSP 后将其设置为可配对。

5.12 更改简易配对参数

说明

ChangeSimplePairingParameters 命令负责在使用安全简易配对（安全级别 4）时更改配对过程中交换的安全简易配对参数。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。IOCapability 和 MITMProtection 值存储在用于安全简易配对的静态全局变量中。

参数

此命令需要两个参数，分别是“I/O 功能”和“MITM 要求”。第一个参数必须指定为 0（表示“仅显示器”）、1（表示“显示器是/否”）、2（表示“仅键盘”）或 3（表示“无输入/输出”）。第二个参数必须指定为 0（表示“无 MITM”）或 1（表示“需要 MITM”）。

命令调用示例

- “ChangeSimplePairingParameters 3 0”尝试将“I/O 功能”设置为“无输入/输出”并关闭“MITM 保护”。
- “ChangeSimplePairingParameters 2 1”尝试将“I/O 功能”设置为“仅键盘”并激活“MITM 保护”。
- “ChangeSimplePairingParameters 1 1”尝试将“I/O 功能”设置为“显示器是/否”并激活“MITM 保护”。

可能的返回值

- (0) 成功更改配对参数
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR

5.13 获取本地地址

说明

GetLocalAddress 命令负责查询本地蓝牙设备的蓝牙设备地址。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

使用此命令时不需要包含参数。参数对查询的结果没有影响。

可能的返回值

- (0) 成功查询本地地址
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-8) INVALID_STACK_ID_ERROR
- (-4) FUNCTION_ERROR

API 调用

GAP_Query_Local_BD_ADDR(BluetoothStackID, &BD_ADDR)

API 原型

*int BTPSAPI GAP_Query_Local_BD_ADDR(unsigned int BluetoothStackID, BD_ADDR_t *BD_ADDR)*

API 说明

此函数负责查询（和报告）本地蓝牙设备的设备地址。第二个参数是指向缓冲区的指针，该缓冲区用于接收本地蓝牙设备的设备地址。如果该函数成功，则 BD_ADDR 参数指向的缓冲区将填充从本地蓝牙设备读取的设备地址。如果此函数返回负值，则无法查询本地蓝牙设备的设备地址（错误情况）。

5.14 设置本地名称

说明

SetLocalName 命令负责将本地蓝牙设备的名称设置为指定名称。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

此命令需要一个参数。指定的设备名称必须是唯一的参数。

备注

请勿在名称中添加空格，否则只会设置名称的第一部分。

命令调用示例

- “SetLocalName New_Bluetooth_Device_Name”尝试将本地设备名称设置为“New_Bluetooth_Device_Name”。
- “SetLocalName New Bluetooth Device Name”尝试将本地设备名称设置为“New Bluetooth Device Name”，但仅设置第一个参数，这将使本地设备名称变为“New”。
- “SetLocalName MSP430”尝试将本地设备名称设置为“MSP430”。

可能的返回值

- (0) 成功设置本地设备名称
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-8) INVALID_STACK_ID_ERROR
- (-4) FUNCTION_ERROR
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Set_Local_Device_Name(BluetoothStackID, TempParam->Params[0].strParam)

API 原型

*int BTPSAPI GAP_Set_Local_Device_Name(unsigned int BluetoothStackID, char *Name)*

API 说明

提供此函数是为了允许更改本地蓝牙设备的设备名称。名称参数必须是指向以 NULL 为终止符的 ASCII 字符串的指针，且长度最大为 MAX_NAME_LENGTH (不包括尾部的 NULL 终止符) 。如果成功更改了本地设备名称，此函数返回零；如果出现错误情况，则返回负的错误代码。

5.15 获取本地名称

说明

此函数负责查询本地蓝牙设备的名称。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

使用此命令时不需要包含参数。参数对查询的结果没有影响。

可能的返回值

- (0) 成功查询本地设备名称
- (-8) INVALID_STACK_ID_ERROR
- (-4) FUNCTION_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- (-65) BTPS_ERROR_INSUFFICIENT_BUFFER_SPACE

API 调用

GAP_Query_Local_Device_Name(BluetoothStackID, 257, (char *)LocalName)

API 原型

*int BTPSAPI GAP_Query_Local_Device_Name(unsigned int BluetoothStackID, unsigned int NameBufferLength, char *NameBuffer)*

API 说明

此函数负责查询（和报告）本地蓝牙设备的用户友好名称。此函数的最后几个参数指定要接收本地设备名称的缓冲区和缓冲区长度。NameBufferLength 参数必须至少为 (MAX_NAME_LENGTH+1)，才能容纳允许的最长设备名称（加上用于容纳 NULL 终止符的单个字符）。如果此函数成功，则此函数返回零，并且 NameBuffer 指向的缓冲区将填充本地设备名称以 NULL 为终止符的 ASCII 表示形式。如果此函数返回负值，则无法查询本地设备名称（错误情况）。

5.16 设置设备类别

说明

SetClassOfDevice 命令负责将本地蓝牙设备的设备类别设置为一个设备类别值。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

唯一需要的参数是新的设备类别值。该值以“0x”开头，后跟一个六位数值。如果不这样做，则写入的设备类别将假定为十进制，并转换为十六进制格式，从而更改给定的值。

命令调用示例

- “SetClassOfDevice 0x123456”尝试将本地蓝牙设备的设备类别设置为“0x123456”。
- “SetClassOfDevice 123456”尝试将本地蓝牙设备的设备类别设置为“0x01E240”，这相当于十进制值 123456。

可能的返回值

- (0) 成功设置本地设备类别
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-8) INVALID_STACK_ID_ERROR
- (-4) FUNCTION_ERROR
- (-5) BTPS_ERROR_GAP_NOT_INITIALIZED

API 调用

GAP_Set_Class_of_Device(BluetoothStackID, Class_of_Device)

API 原型

int BTPSAPI GAP_Set_Class_Of_Device(unsigned int BluetoothStackID, Class_of_Device_t Class_of_Device)

API 说明

提供此函数是为了允许更改本地蓝牙设备的设备类别。Class_of_Device 参数表示要写入本地蓝牙设备的设备类别值。如果成功更改设备类别，此函数将返回零；如果出现错误情况，则返回负的错误代码。

5.17 获取设备类别

说明

GetClassOfDevice 命令负责查询本地蓝牙设备的蓝牙设备类别。此函数在成功执行时返回零，而在出现任何错误时返回负值。必须存在蓝牙协议栈 ID，才能尝试调用此函数。

参数

使用此命令时不需要包含参数。参数对查询的结果没有影响。

可能的返回值

- (0) 成功查询本地设备类别
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-8) INVALID_STACK_ID_ERROR
- (-4) FUNCTION_ERROR
- (-1) BTPS_ERROR_INVALID_PARAMETER

API 调用

GAP_Query_Class_Of_Device(BluetoothStackID, &Class_of_Device)

API 原型

*int BTPSAPI GAP_Query_Class_Of_Device(unsigned int BluetoothStackID, Class_of_Device_t *Class_of_Device)*

API 说明

此函数负责查询（和报告）本地蓝牙设备的设备类别。第二个参数是指向设备类别缓冲区的指针，该缓冲区用于接收本地设备的蓝牙设备类别。如果成功，则此函数将返回零，而 Class_Of_Device 指向的缓冲区将填充从本地蓝牙设备读取的设备类别。如果出错，则此函数将返回负值，并且本地蓝牙设备的设备类别不会复制到指定的输入缓冲区中。

5.18 获取远程名称

说明

GetRemoteName 命令负责查询远程设备的蓝牙设备名称。此函数在成功执行时返回零，而在出现任何错误时返回负值。该命令要求在运行之前存在有效的蓝牙协议栈 ID，并在使用 Inquiry 命令后调用。在这种情况下，DisplayInquiryList 命令可用于查找哪个远程设备与哪个查询索引关联。

参数

GetRemoteName 命令需要一个参数，即远程蓝牙设备的查询索引。该值可以在查询后找到，或者在使用命令 DisplayInquiryList 时显示。

命令调用示例

- “GetRemoteName 5”尝试查询位于第五个查询索引处的远程设备的设备名称。
- “GetRemoteName 8”尝试查询位于第八个查询索引处的远程设备的设备名称。

可能的返回值

- (0) 成功查询远程名称
- (-6) INVALID_PARAMETERS_ERROR
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-59) BTPS_ERROR_ADDING_CALLBACK_INFORMATION
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_Query_Remote_Device_Name(BluetoothStackID, InquiryResultList[(TempParam->Params[0].intParam - 1)], GAP_Event_Callback, (unsigned long)0)

API 原型

int BTPSAPI GAP_Query_Remote_Device_Name(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Event_Callback_t GAP_Event_Callback, unsigned longCallbackParameter)

API 说明

提供此函数是为了能够查询指定远程蓝牙设备的用户友好的蓝牙设备名称。该函数将以下内容作为输入：远程蓝牙设备的地址（用于查询该设备的名称）以及 GAP 事件回调信息（在远程设备名称查询过程完成时需要使用）。如果成功，此函数返回零；如果无法提交远程名称请求，则返回负的错误代码。如果此函数返回成功结果，则在确定远程名称信息后（或存在错误时），将通过指定的回调通知调用方。此函数无法用于确定本地蓝牙设备的用户友好名称。GAP_Query_Local_Name 函数不得用于查询本地蓝牙设备的用户友好名称。由于此函数本质上是异步的（指定远程设备地址），因此该函数通过指定的回调向调用方通知结果。通过发出 GAP_Cancel_Query_Remote_Name 函数并指定蓝牙设备的蓝牙设备地址（在对该函数的原始调用中指定），调用方可以随时取消远程名称请求。请注意，取消回调后，该操作会发出取消尝试，然后回调会被取消（即，GAP 模块仍可执行远程名称请求，但从未发出回调）。

6 人机接口演示配置文件

6.1 主机

6.1.1 连接远程 HID 设备

说明

以下函数负责连接到远程 HID 设备。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

此命令需要查询索引号，查询完成后可以使用 DisplayInquiryList 命令找到该编号。

可能的返回值

- (0)HID_Connect_Remote_Device：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1004)BTHID_ERROR_INSUFFICIENT_RESOURCES

API 调用

HID_Connect_Remote_Device(BluetoothStackID, InquiryResultList[(TempParam->Params->intParam-1)], &HIDConfiguration, HID_Event_Callback, 0)

API 原型

int BTPSAPI HID_Connect_Remote_Device(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, HID_Configuration_t *HIDConfiguration, HID_Event_Callback_tEventCallback, unsigned long CallbackParameter)

API 说明

以下函数负责在指定的蓝牙设备上打开与远程 HID 设备的连接。此函数使用蓝牙协议栈的 ID（用于打开 HID 连接）作为第一个参数。第二个参数指定要连接的远程蓝牙设备的板地址（非 NULL）。此函数的第三个参数是 HID 配置规范，将用于协商与此设备客户端关联的 L2CAP 通道。最后两个参数分别指定 HID 事件回调的 HID 事件回调函数和回调参数，HID 事件回调用于处理与此设备客户端关联的任何其他事件。如果成功，此函数将返回非零正值；如果不成功，则返回负的错误代码。如果该函数成功，则返回值代表 HID ID，可将该 ID 传递给所有需要它的其他函数。一旦打开与远程设备的连接，只能通过调用 HID_Close_Connection() 函数来关闭连接（将成功调用此函数后的返回值作为 HID ID 输入参数传入）。

6.1.2 关闭连接

说明

以下函数负责关闭任何正在进行的连接。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

使用此命令时不需要包含参数。参数对 close 命令的结果没有影响。

可能的返回值

- (0) HID_Close_Connection：成功运行
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1004)BTHID_ERROR_INSUFFICIENT_RESOURCES
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Close_Connection(BluetoothStackID, HIDID)

API 原型

int BTPSAPI HID_Close_Connection(unsigned int BluetoothStackID, unsigned int HIDID)

API 说明

以下函数负责关闭通过连接到注册服务器而建立的 HID 连接，或关闭通过调用 HID_Open_Remote_Device() 或 HID_Open_Remote_Host() 函数而建立的连接。此函数使用蓝牙协议栈的 ID 作为输入，第二个参数指定的 HID ID 对该蓝牙协议栈有效。如果成功，此函数将返回零；如果发生错误，则返回负的错误代码。请注意，如果使用本地服务器的 HID ID 调用此函数，则服务器保持注册状态，但关闭与指定 HID ID 关联的连接。

6.1.3 控制请求

说明

以下函数负责向远程实体发送 HID_CONTROL 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

如果设备是主机，则需要 Control Operation（0= hcNop，1= hcHardReset，2= hcSoftReset，3= hcSuspend，4= hcExitSuspend，5= hcVirtualCableUnplug）参数。如果设备不是主机，则使用此命令时不需要参数，因为参数对 ControlRequest 的结果没有影响。

可能的返回值

- (0) HID_Control_Request：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Control_Request(BluetoothStackID, HIDID, (HID_Control_Operation_Type_t)((!IsHost)? (hcVirtualCableUnplug):(TempParam->Params->intParam)))

API 原型

int BTPSAPI HID_Control_Request (unsigned int BluetoothStackID, unsigned int HIDID, HID_Control_Operation_Type_t ControlOperation)

API 说明

以下函数负责向远端发送 HID_CONTROL 事务。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送请求）和已建立连接的 HID ID。第三个参数是 Control Operation。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输通常包括两个阶段：主机请求和设备响应。然而，HID 控制事务不需要响应阶段。请注意，处理其他事务期间不允许发出 HID 控制请求，除非 Control Operation 类型为 hcVirtualCableUnplug（可随时发送）。

6.1.4 获取报告请求

说明

以下函数负责向远程 HID 设备发送 GET_REPORT 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

此函数需要三个或四个参数。第一个是 size（0 = grSizeOfReport，1 = grUseBufferSize），第二个是 ReportType（0 = rtOther，1 = rtInput，2 = rtOutput，3 = rtFeature），第三个是 ReportId。如果 size 参数为 1，则需要指定第四个参数 Buffersize。

可能的返回值

- (0) HID_Get_Report_Request：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION

API 调用

HID_Get_Report_Request(BluetoothStackID, HIDID, (HID_Get_Report_Size_Type_t)TempParam->Params[0].intParam, (HID_Report_Type_Type_t)TempParam->Params[1].intParam, (Byte_t)(TempParam->Params[2].intParam), (Word_t)(TempParam->Params[3].intParam))

API 原型

HID_Get_Report_Request(unsigned int BluetoothStackID, unsigned int HIDID, HID_Get_Report_Size_Type_t Size, HID_Report_Type_Type_t ReportType, Byte_t ReportID,Word_t BufferSize)

API 说明

以下函数负责向远程设备发送 GET_REPORT 事务。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送请求）和已建立连接的 HID ID。第三个参数是指示设备如何确定主机已分配的缓冲区大小的描述符。第四个参数是所请求的报告的类型。第五个参数是由设备的 SDP 记录确定的报告 ID。将此参数设置为零表示不使用此参数，

并从事务有效载荷中排除相应字节。使用的第五个参数基于作为大小传递的参数。如果主机指示分配的缓冲区大小小于报告请求的缓冲区大小，则此参数将用作返回的报告的大小。否则，事务有效载荷中不包含相应的字节。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输包括两个阶段：主机请求和设备响应。一次只允许一个主机控制通道请求未处理。接收到 HID Get Report Confirmation 事件表示已收到响应，并且控制通道现在可供进一步的事务使用。

6.1.5 设置报告请求

说明

以下函数负责向远程 HID 设备发送 SET_REPORT 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

SetReportRequest 使用一个参数，即 ReportType（0 = rtOther，1 = rtInput，2 = rtOutput，3 = rtFeature）。

可能的返回值

- (0)HID_Set_Report_Request：成功运行。
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Set_Report_Request(BluetoothStackID, HIDID, (HID_Report_Type_Type_t)TempParam->Params[1].intParam, size of(GenericMouseReport), GenericMouseReport)

API 原型

int BTPSAPI HID_Set_Report_Request(unsigned int BluetoothStackID, unsigned int HIDID, HID_Report_Type_Type_t ReportType, Word_t ReportPayloadSize, Byte_t*ReportDataPayload)

API 说明

以下函数负责向远程设备发送 SET_REPORT 请求。此函数接受蓝牙协议栈的蓝牙协议栈 ID（用于发送事务）和已建立连接的 HID ID 作为输入。第三个参数是要发送的报告的类型。请注意，rtOther 是无效报告类型，不能与此函数一起使用。此函数的最后两个参数是要发送的报告有效载荷的长度和指向要发送的报告有效载荷的指针。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输包括两个阶段：主机请求和设备响应。一次只允许一个主机控制通道请求未处理。接收到 HID Set Report Confirmation 事件表示已收到响应，并且控制通道现在可供进一步的事务使用。

6.1.6 获取协议请求

说明

以下函数负责向远程 HID 设备发送 GET_PROTOCOL 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

使用此命令时不需要包含参数。参数对 GetProtocolRequest 的结果没有影响。

可能的返回值

- (0)HID_Get_Protocol_Request：成功运行。
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Get_Protocol_Request(BluetoothStackID, HIDID)

API 原型

int BTPSAPI HID_Get_Protocol_Request (unsigned int BluetoothStackID, unsigned int HIDID)

API 说明

以下函数负责向远程 HID 设备发送 GET_PROTOCOL 事务。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送请求）和已建立连接的 HID ID。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输包括两个阶段：主机请求和设备响应。一次只允许一个主机控制通道请求未处理。接收到 HID Get Protocol Confirmation 事件表示已收到响应，并且控制通道现在可供进一步的事务使用。

6.1.7 设置协议请求

说明

以下函数负责向远程 HID 设备发送 SET_PROTOCOL 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

SetProtocolRequest 需要一个参数，即 Protocol（0= ptReport，1= ptBoot）。

可能的返回值

- (0) HID_Set_Protocol_Request：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR

- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Set_Protocol_Request(BluetoothStackID, HIDID, (HID_Protocol_Type_t)TempParam->Params[0].intParam)

API 原型

int BTPSAPI HID_Set_Protocol_Request(unsigned int BluetoothStackID, unsigned int HIDID, HID_Protocol_Type_t Protocol)

API 说明

以下函数负责向远程 HID 设备发送 SET_PROTOCOL 事务。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送请求）和已建立连接的 HID ID。最后一个参数是要设置的 Protocol。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输包括两个阶段：主机请求和设备响应。一次只允许一个主机控制通道请求未处理。接收到 HID Set Protocol Confirmation 事件表示已收到响应，并且控制通道现在可供进一步的事务使用。

6.1.8 获取空闲请求

说明

以下函数负责向远程 HID 设备发送 GET_IDLE 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

使用此命令时不需要包含参数。参数对 GetIdleRequest 的结果没有影响。

可能的返回值

- (0) HID_Get_Idle_Request：成功运行。
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Get_Idle_Request(BluetoothStackID, HIDID)

API 原型

int BTPSAPI HID_Get_Idle_Request(unsigned int BluetoothStackID, unsigned int HIDID)

API 说明

以下函数负责向远程 HID 设备发送 GET_IDLE 事务。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送请求）和已建立连接的 HID ID。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输包括两个阶段：主机请求和设备响应。一次只允许一个主机控制通道请求未处理。接收到 HID Get Idle Confirmation 事件表示已收到响应，并且控制通道现在可供进一步的事务使用。

6.1.9 设置空闲请求

说明

以下函数负责向远程 HID 设备发送 SET_IDLE 事务。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

SetIdleRequest 需要一个参数，即 Idlerate。

可能的返回值

- (0) HID_Set_Idle_Request：成功运行
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1004)BTHID_ERROR_INSUFFICIENT_RESOURCES
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Set_Idle_Request(BluetoothStackID, HIDID, (Byte_t)TempParam->Params[0].intParam)

API 原型

int BTPSAPI HID_Set_Idle_Request(unsigned int BluetoothStackID, unsigned int HIDID, Byte_t IdleRate)

API 说明

以下函数负责向远程 HID 设备发送 SET_IDLE 事务。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送请求）和已建立连接的 HID ID。最后一个参数是待设置的 Idle Rate。Idle Rate LSB 加权至 4ms（即 Idle Rate 分辨率为 4ms，范围为 4ms 至 1.020s）。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

备注

控制通道传输包括两个阶段：主机请求和设备响应。一次只允许一个主机控制通道请求未处理。接收到 HID Set Idle Confirmation 事件表示已收到响应，并且控制通道现在可供进一步的事务使用。

6.1.10 数据写入

说明

以下函数负责将中断通道上的数据事务发送到远程实体。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

需要一个称为 ReportType 的参数，0 = rtOther，1 = rtInput，2 = rtOutput，3 = rtFeature。

可能的返回值

- (0) HID_Control_Request：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION

API 调用

HID_Data_Write(BluetoothStackID, HIDID, (HID_Report_Type_Type_t)TempParam->Params[0].intParam, sizeof(GenericMouseReport), GenericMouseReport)

API 原型

int BTPSAPI HID_Data_Write(unsigned int BluetoothStackID, unsigned int HIDID, HID_Report_Type_Type_t ReportType, Word_t ReportPayloadSize, Byte_t *ReportDataPayload)

API 说明

以下函数负责通过中断通道发送报告。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送 ReportData）和已建立连接的 HID ID。第三个参数是要发送的报告的类型。最后两个参数是要发送的报告有效载荷的长度和指向要发送的报告有效载荷的指针。请注意，rtOther 和 rtFeature 是无效报告类型，不能与此函数一起使用。另请注意，rtInput 报告必须从设备发送至主机，而 rtOutput 报告必须从主机发送至设备。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

6.2 客户端

6.2.1 获取报告响应

说明

以下函数负责将未处理的 GET_REPORT 事务的响应发送到远程 HID 主机。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

GetReportResponse 需要两个参数，一个是 ResultType（0= rtSuccessful，1= rtNotReady，2= rtErrInvalidReportID，3= rtErrUnsupportedRequest，4= rtErrInvalidParameter，5= rtErrUnknown，6= rtErrFatal，7= rtData），另一个是 ReportType（0 = rtOther，1 = rtInput，2 = rtOutput，3 = rtFeature）。

可能的返回值

- (0) HID_Get_Report_Response：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID

- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Get_Report_Response(BluetoothStackID, HIDID, (HID_Result_Type_t)TempParam->Params[0].intParam, (HID_Report_Type_Type_t)TempParam->Params[1].intParam,sizeof(GenericMouseReport), GenericMouseReport)

API 原型

int BTPSAPI HID_Get_Report_Response(unsigned int BluetoothStackID, unsigned int HIDID, HID_Result_Type_t ResultType, HID_Report_Type_Type_t ReportType, Word_tReportPayloadSize, Byte_t *ReportDataPayload)

API 说明

以下函数负责对未处理的 GET_REPORT 事务发送适当的响应。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送响应求）和已建立连接的 HID ID。此函数的第三个参数是要与该响应关联的 ResultType。rtSuccessful 结果类型不能与此函数一起使用。如果使用 rtNotReady 到 rtErrFatal 结果状态进行响应，则会发送具有指定错误条件的结果代码参数的 HANDSHAKE 响应。如果指定的 ResultType 为 rtData，则将使用 DATA 响应来响应 GET_REPORT 事务，该响应将报告（由最后一个参数指定）作为有效载荷。第四个参数是要发送的报告的类型。请注意，rtOther 是无效报告类型，不能与此函数一起使用。最后两个参数是要发送的报告有效载荷的长度和指向报告有效载荷的指针。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

6.2.2 设置报告响应

说明

以下函数负责将未处理的 SET_REPORT 事务的响应发送到远程 HID 主机。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

SetReportResponse 需要一个参数，即 ResultType（0= rtSuccessful，1= rtNotReady，2= rtErrInvalidReportID，3= rtErrUnsupportedRequest，4= rtErrInvalidParameter，5= rtErrUnknown，6= rtErrFatal，7= rtData）。

可能的返回值

- (0) HID_Set_Report_Response：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION

API 调用

HID_Get_Report_Response(BluetoothStackID, HIDID, (HID_Result_Type_t)TempParam->Params[0].intParam, (HID_Report_Type_Type_t)TempParam->Params[1].intParam,sizeof(GenericMouseReport), GenericMouseReport)

API 原型

int BTPSAPI HID_Set_Report_Response(unsigned int BluetoothStackID, unsigned int HIDID, HID_Result_Type_t ResultType)

API 说明

以下函数负责对未处理的 SET_REPORT 事务发送适当的响应。此函数接受蓝牙协议栈的蓝牙协议栈 ID（用于发送响应）和已建立连接的 HID ID 作为输入。此函数的第三个参数是要与该响应关联的 ResultType。rtData 结果类型不能与此函数一起使用。如果指定了 rtSuccessful 到 rtErrFatal 结果类型，则此函数将使用 HANDSHAKE 响应来响应 SET_REPORT 请求，该响应具有与指定结果类型匹配的结果代码参数。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

6.2.3 获取协议响应

说明

以下函数负责将未处理的 GET_PROTOCOL 事务的响应发送到远程 HID 主机。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

GetProtocolResponse 需要两个参数，一个是 ResultType（0= rtSuccessful，1= rtNotReady，2= rtErrInvalidReportID，3= rtErrUnsupportedRequest，4= rtErrInvalidParameter，5= rtErrUnknown，6= rtErrFatal，7= rtData），另一个是 Protocol（0= ptReport，1= ptBoot）。

可能的返回值

- (0) HID_Get_Protocol_Response：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1004)BTHID_ERROR_INSUFFICIENT_RESOURCES
- (-1005)BTHID_ERROR_INVALID_OPERATION

API 调用

HID_Get_Protocol_Response(BluetoothStackID, HIDID, (HID_Result_Type_t)TempParam->Params[0].intParam, (HID_Protocol_Type_t)TempParam->Params[1].intParam)

API 原型

int BTPSAPI HID_Get_Protocol_Response(unsigned int BluetoothStackID, unsigned int HIDID, HID_Result_Type_t ResultType, HID_Protocol_Type_t Protocol)

API 说明

以下函数负责对未处理的 GET_PROTOCOL 事务发送适当的响应。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送响应求）和已建立连接的 HID ID。此函数的第三个参数是要与该响应关联的 ResultType。rtSuccessful 结果类型不能与此函数一起使用。如果指定了 rtNotReady 到 rtErrFatal 结果类型，则此函数将使用 HANDSHAKE 响应来响应 GET_PROTOCOL 请求，该响应具有指定错误条件的结果代码参数。如果指定的 ResultType 为 rtData，则将使用 DATA 响应来响应 GET_PROTOCOL 事务，该响应将指定为最后一个参数的协议类型作为有效载荷。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

6.2.4 设置协议响应

说明

以下函数负责将未处理的 SET_PROTOCOL 事务的响应发送到远程 HID 主机。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

SetProtocolResponse 需要一个参数，即 ResultType（0= rtSuccessful，1= rtNotReady，2= rtErrInvalidReportID，3= rtErrUnsupportedRequest，4= rtErrInvalidParameter，5= rtErrUnknown，6= rtErrFatal，7= rtData）。

可能的返回值

- (0) HID_Set_Protocol_Response：成功运行
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Set_Protocol_Response(BluetoothStackID, HIDID, (HID_Protocol_Type_t)TempParam->Params[0].intParam)

API 原型

int BTPSAPI HID_Set_Protocol_Response(unsigned int BluetoothStackID, unsigned int HIDID, HID_Result_Type_t ResultType)

API 说明

以下函数负责对未处理的 SET_PROTOCOL 事务发送适当的响应。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送响应求）和已建立连接的 HID ID。此函数的第三个参数是要与该响应关联的 ResultType。rtData 结果类型不能与此函数一起使用。如果指定了 rtSuccessful 到 rtErrFatal 结果类型，则此函数将使用 HANDSHAKE 响应来响应 SET_PROTOCOL 事务，该响应具有与指定结果类型匹配的结果代码参数。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

6.2.5 获取空闲响应

说明

以下函数负责将未处理的 GET_IDLE 事务的响应发送到远程 HID 主机。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

GetIdleResponse 需要两个参数，一个是 ResultType（0= rtSuccessful，1= rtNotReady，2= rtErrInvalidReportID，3= rtErrUnsupportedRequest，4= rtErrInvalidParameter，5= rtErrUnknown，6= rtErrFatal，7= rtData），另一个是 IdleRate。

可能的返回值

- (0) HID_Set_Idle_Response：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION

API 调用

HID_Get_Idle_Response(BluetoothStackID, HIDID, (HID_Result_Type_t)TempParam->Params[0].intParam, (Byte_t)TempParam->Params[1].intParam)

API 原型

int BTPSAPI HID_Get_Idle_Response(unsigned int BluetoothStackID, unsigned int HIDID, HID_Result_Type_t ResultType, Byte_t IdleRate)

API 说明

以下函数负责对未处理的 GET_IDLE 事务发送适当的响应。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送响应求）和已建立连接的 HID ID。此函数的第三个参数是要与该响应关联的 ResultType。rtSuccessful 结果类型不能与此函数一起使用。如果指定了 rtNotReady 到 rtErrFatal 结果类型，则此函数将使用 HANDSHAKE 响应来响应 GET_IDLE 事务，该响应具有指定错误条件的结果代码参数。如果指定的 ResultType 为 rtData，则将使用 DATA 响应来响应 GET_IDLE 事务，该响应将指定为最后一个参数的 Idle Rate 作为有效载荷。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

6.2.6 设置空闲响应

说明

以下函数负责将未处理的 SET_IDLE 事务的响应发送到远程 HID 主机。此函数在成功执行时返回零，而在出现任何错误时返回负值。

参数

SetIdleResponse 需要一个参数，即 ResultType（0= rtSuccessful，1= rtNotReady，2= rtErrInvalidReportID，3= rtErrUnsupportedRequest，4= rtErrInvalidParameter，5= rtErrUnknown，6= rtErrFatal，7= rtData）。

可能的返回值

- (0) HID_Get_ Idle _Response：成功运行
- (-4) FUNCTION_ERROR
- (-6)INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-103)BTPS_ERROR_FEATURE_NOT_AVAILABLE
- (-1000)BTHID_ERROR_INVALID_PARAMETER
- (-1001)BTHID_ERROR_NOT_INITIALIZED
- (-1002)BTHID_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1005)BTHID_ERROR_INVALID_OPERATION
- (-1006)BTHID_ERROR_REQUEST_OUTSTANDING

API 调用

HID_Set_Idle_Response(BluetoothStackID, HIDID, (HID_Result_Type_t)TempParam->Params[0].intParam)

API 原型

int BTPSAPI HID_Set_Idle_Response(unsigned int BluetoothStackID, unsigned int HIDID, HID_Result_Type_t ResultType)

API 说明

以下函数负责对未处理的 SET_IDLE 事务发送适当的响应。该函数将以下内容作为输入：蓝牙协议栈的 ID（用于发送响应求）和已建立连接的 HID ID。此函数的第三个参数是要与该响应关联的 ResultType。rtData 结果类型不能与此函数一起使用。如果指定了 rtSuccessful 到 rtErrFatal 结果类型，则此函数将使用 HANDSHAKE 响应来响应 SET_IDLE 事务，该响应具有与指定结果类型匹配的结果代码参数。如果成功，此函数返回零；如果出现错误，则返回负的错误代码。

7 参考文献

- 德州仪器 (TI)，*基于 MSP432™ MCU 的 TI 双模 Bluetooth® Stack*，用户指南。
- 德州仪器 (TI)，*基于 STM32F4 MCU 的双模 Bluetooth® Stack*，用户指南。

8 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

日期	修订版本	说明
August 2023	*	初始发行版

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（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，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(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功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265