User's Guide

CC256x TI Bluetooth Stack SPPLEDemo 应用

摘要

本文档详细讨论了 SPPLE 应用程序。借助该应用程序，用户能够通过控制台，使用低功耗蓝牙 (BLE) 在两个 BLE 设备之间建立连接、发送蓝牙命令并通过 BLE 交换数据。

内容

商标

所有商标均为其各自所有者的财产。

1 引言

此应用程序演示了基于 BR/EDR SPP 的应用程序以及基于低功耗蓝牙的自定义应用程序 SPPLE，该应用程序在功能上与 BR/EDR 应用程序类似。SPPLE 配置文件与 SPP 配置文件类似，不同之处在于与 SPP 配置中的 BR/EDR 传输相比，SPPLE 使用低功耗传输。SPP 配置文件模拟串行电缆连接。此配置文件中定义了两个角色。第一个角色是运行 SPPLE 服务并打开服务器端口的服务器。客户端是连接到服务器的设备。然后，这两个设备可以相互交换数据。

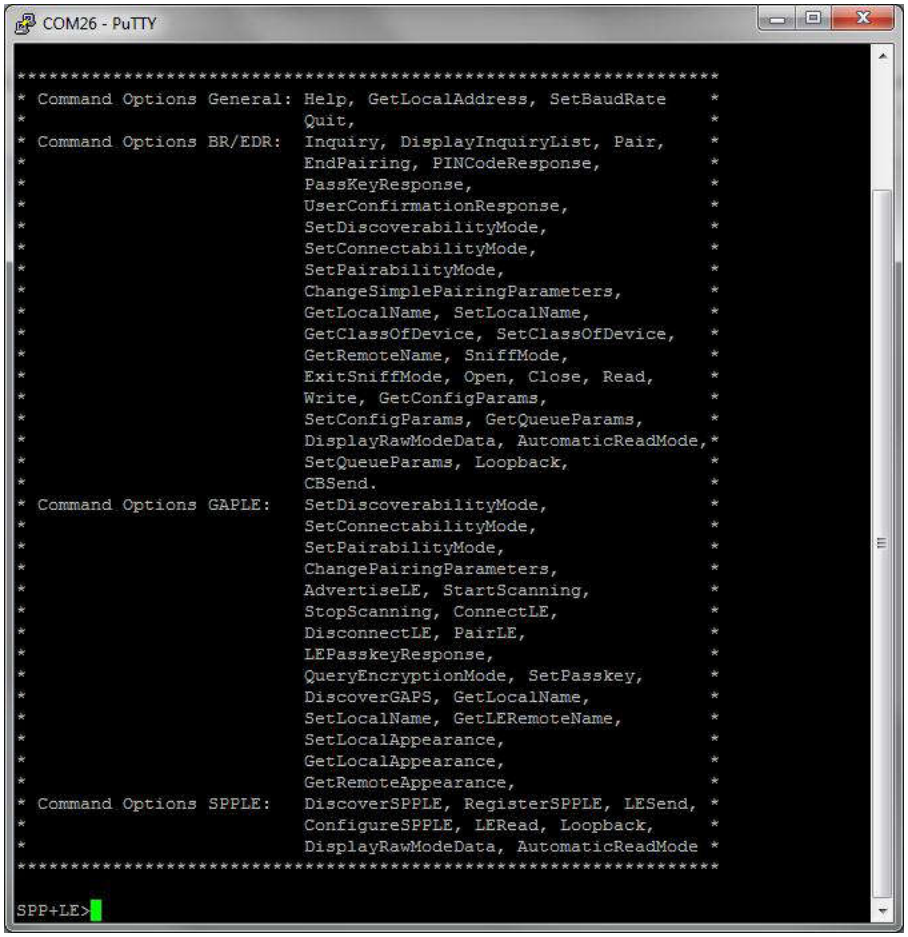
请先参阅基于 MSP432™ MCU 的 TI 双模 Bluetooth® Stack 或基于 STM32F4 MCU 的双模 Bluetooth® Stack 文档，然后试用本文档介绍的应用程序。

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。该驱动程序在设备管理器的“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。使用串行参数 115200 波特率（对于 MSP430 为 9600）、8、n、1。连接后，使用“Reset S3”按钮（位于 MSP430 的 Mini USB 连接器旁边）复位设备，并观察终端上的协议栈初始化情况，此时显示帮助屏幕，其中显示了所有命令。该设备成为服务器。通过 miniUSB 或 microUSB 电缆连接第二块电路板，并按照之前在第一块电路板上运行蓝牙代码时执行的相同步骤进行操作。连接到计算机的第二个设备是客户端。

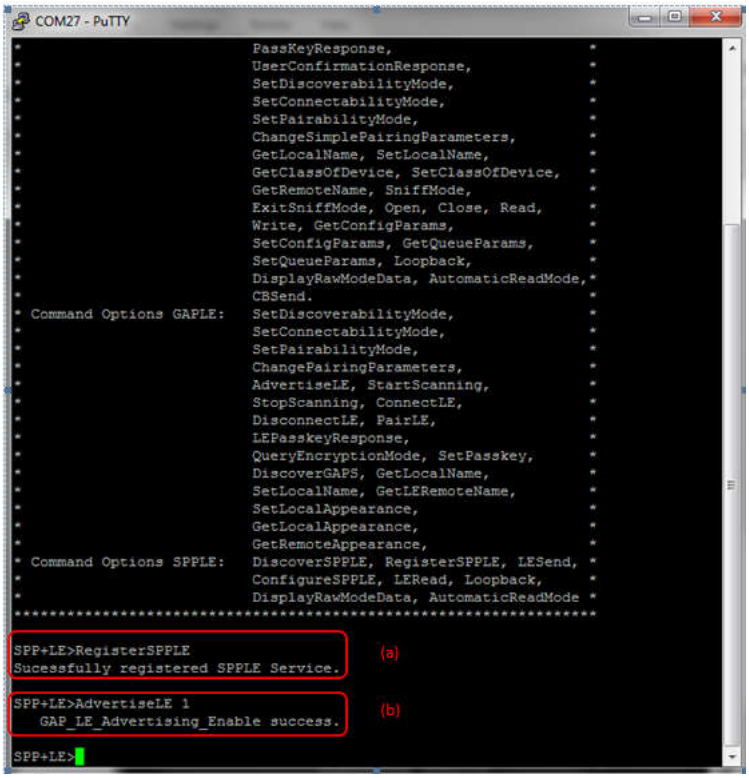


3 演示应用程序

下面介绍了如何使用演示应用程序来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 演示应用程序上的设备 1（服务器）设置

1. 首先，键入以下命令将设备置于服务器模式：Server on the console。然后，可以通过运行 RegisterSPPLE 来启动 SPP-LE 服务。
2. 接下来，充当服务器的设备需要向其他设备进行广播。这可以通过运行 AdvertiseLE 1 来完成。

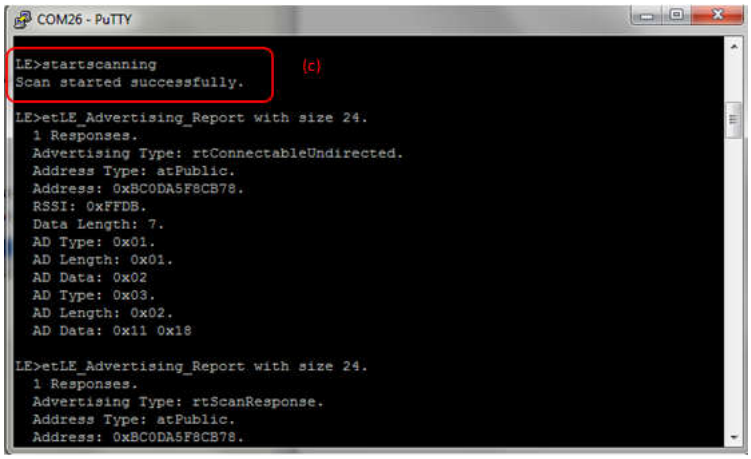


3.2 演示应用程序上的设备 2（客户端）设置

1. 通过在控制台上键入 Client 将设备置于客户端模式。
2. 客户端 LE 设备可以尝试使用以下命令查找附近有哪些 LE 设备：StartScanning。
3. 找到设备后，您可以使用以下命令停止扫描：StopScanning。

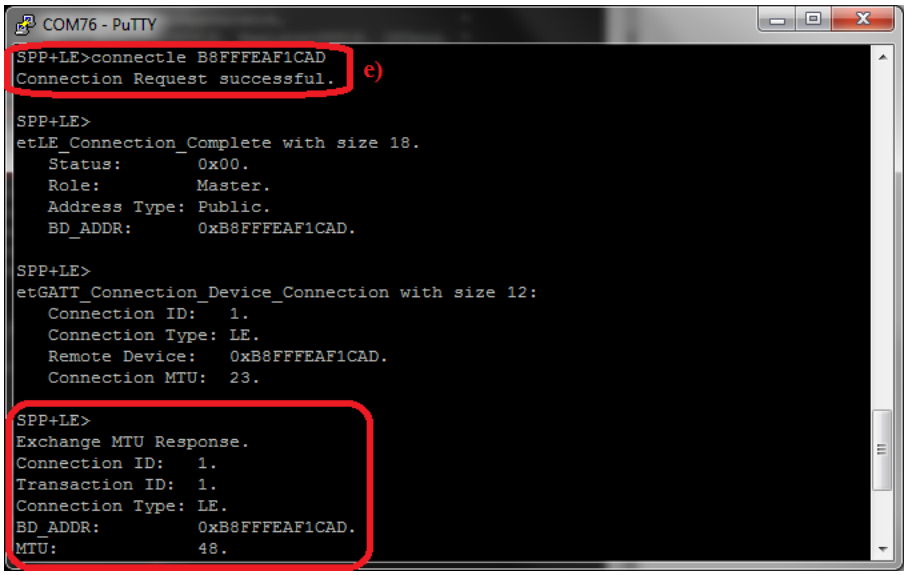
备注

如果蓝牙地址已知，则步骤 2 和 3 是可选的。

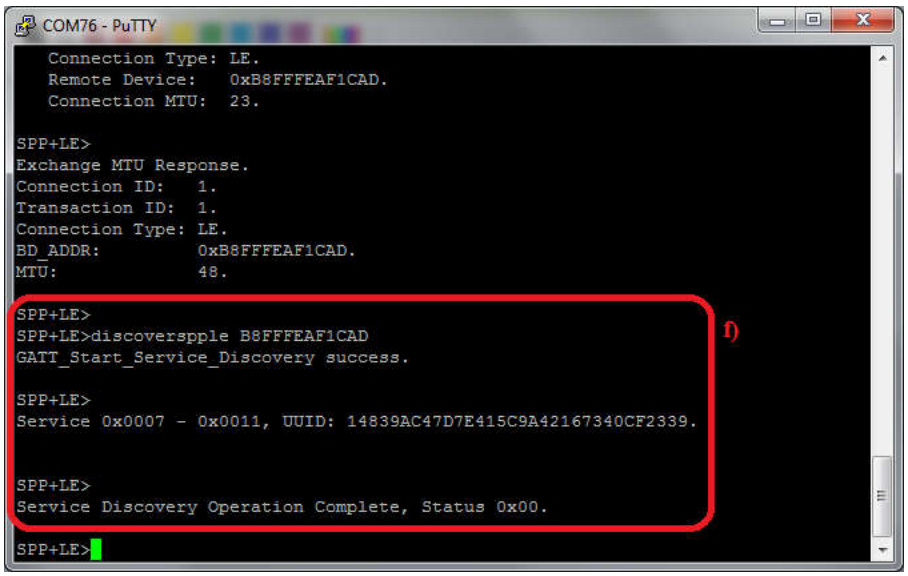


3.3 从设备 2 启动连接

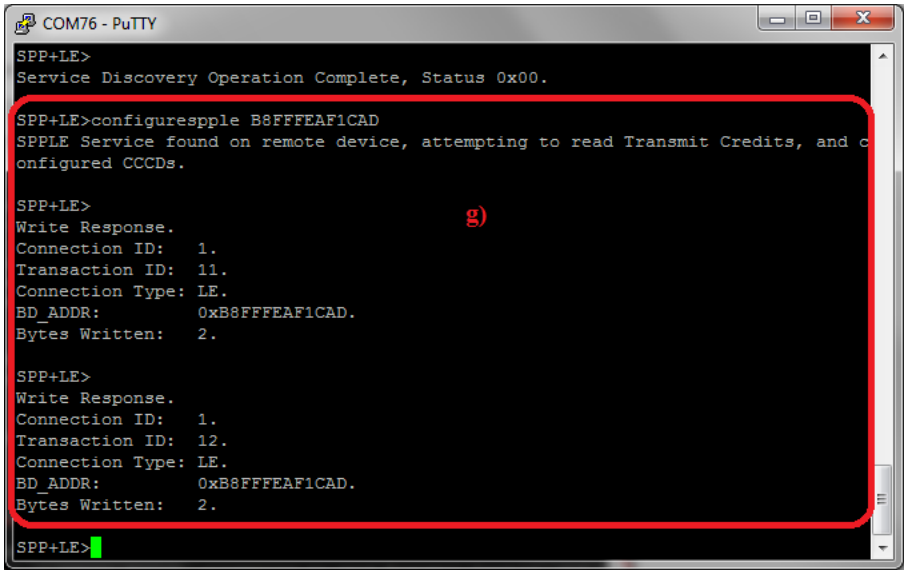
1. 一旦客户端上的应用程序知道正在广播的设备的蓝牙地址，应用程序就可以使用以下命令连接到该设备：ConnectLE <蓝牙地址>



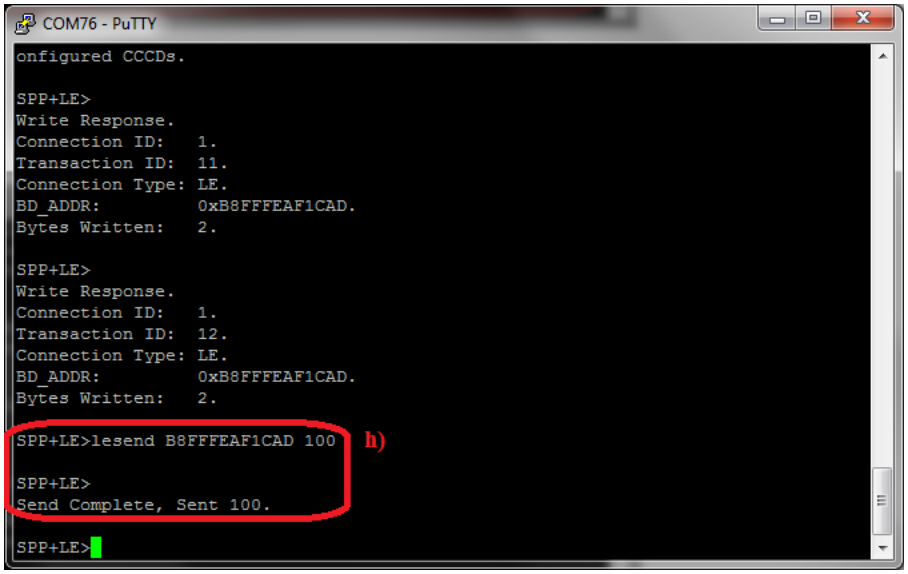
3.4 确定支持的服务



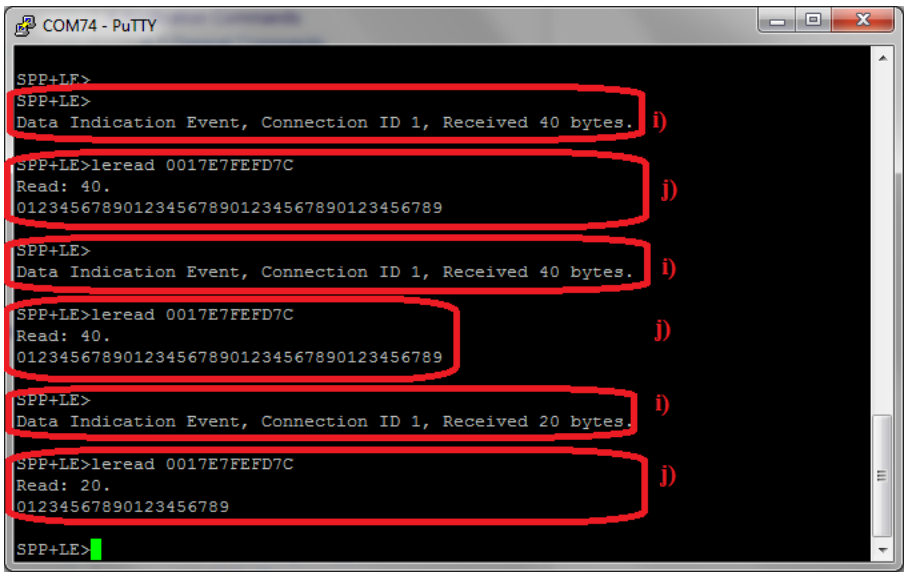
1. 初始化后，设备需要查明是否支持 SPP 服务。为此，请在客户端上运行 DiscoverSPPLE <服务器 BD 地址>。
2. 查明支持 SPP-LE 之后，我们需要配置 SPP-LE。这是通过在客户端上运行 ConfigureSPPLE <服务器 BD 地址> 来完成的。



3.5 客户端和服务器之间的数据传输



1. 配置完成后，我们就可以在客户端和服务器之间发送数据了。要发送数据，请使用 LESend <远程设备 BD 地址> <字节数>。
2. 一旦其他设备接收到数据，该设备也会接收到数据指示事件。
3. 然后，接收设备可以读取使用以下命令发送的数据：LERead <远程设备 BD 地址>
4. 这会打印出发送的数据。此类数据是使用示例应用程序中 SPPLE 的自定义服务通过 BluetoothLE 发送的。



3.6 多个 SPPLE 连接指南

两个 SPPLE 连接

1. 在此版本中，我们测试与 MSP430 的两个同时 SPPLE 连接。远程设备用作外设，而 MSP430 用作中央设备。
2. 连接到第一个设备，并在第一个设备上发现和配置服务。当发现服务和配置服务时，我们必须指定我们连接到的远程 BD_ADDR。
3. 同样，连接到第二个设备，并在第二个设备上发现和配置服务。
4. 为了将数据发送到第一个远程设备数据，我们使用 LeSend <BD-ADDR> <要发送的字节数>。
5. 为了将数据发送到第二个远程设备数据，我们使用 LeSend <BD-ADDR> <要发送的字节数>。
6. 为了从第一个远程设备数据中读取数据，我们使用 LeRead <BD-ADDR>。
7. 为了从第二个远程设备数据中读取数据，我们使用 LeRead <BD-ADDR>。
8. 当 Automaticreadmode、DisplayRawmodedata 或 Loopback 打开时，会同时对这两个连接打开。

一个 SPP 和一个 SPPLE 连接

1. 在此版本中，同时测试与 MSP430 的 SPP 连接和 SPPLE 连接。其中一个远程设备用作外围 LE 设备，而远程设备则用作 SPP 客户端。
2. 连接到第一个设备，并在第一个设备上发现和配置服务。当发现服务和配置服务时，我们必须指定我们连接到的远程 BD_ADDR。
3. 打开 SPP 服务器并让第二个远程设备进行连接。
4. 为了将数据发送到第一个远程设备数据，我们使用 LeSend <BD-ADDR> <要发送的字节数>。
5. 要将数据发送到第二个远程设备数据，我们使用 CBSend <要发送的字节数> <串行端口 ID>。如果我们要写入少量数据，我们使用命令 Write <串行端口 ID>。
6. 为了从第一个远程设备数据中读取数据，我们使用 LeRead <BD-ADDR>。
7. 为了从第二个远程设备数据中读取数据，我们使用 Read。
8. 打开 Automaticreadmode、DisplayRawmodedata 或 Loopback（同时对两个连接打开）。

4 使用 LightBlue 应用在 iOS 设备上演示 SPP LE

4.1 LightBlue 概述

LightBlue 应用是一款免费的 iOS 应用，它使用低功耗蓝牙 (BLE) 测试和演示 GATT 配置文件。此应用创建自定义服务，还与具有自定义服务的服务器进行交互。该应用同时支持 GATT 的客户端角色和服务器角色。下一节将介绍如何将该应用与 SPPLEDemo 应用程序结合使用。

4.2 SPP LE 服务概述

SPP LE 不是官方蓝牙服务。这是一项自定义服务，旨在演示如何使用低功耗蓝牙以与 ClassicBluetooth 的 SPP 配置文件类似的方式发送和接收数据。它使用基于额度的协议来发送和接收数据。要让设备使用 SPP LE 协议将数据发送到远程设备，远程设备必须已向设备提供“额度”。这些额度指定允许设备发送的数据量。当设备已发送最大额度数时，设备必须等待远程设备提供更多额度，然后才能发送。在此应用程序中，1 个额度相当于 1 个字节（八位字节）的数据。

4.2.1 特性

SPP LE 使用 GATT 特性实现基于额度的协议。SPP LE 服务有 4 个特性：

名称	UUID	用途
Rx 特性	0x8B00ACE7-EB0B-49B0-BBE9-9AEE0A26E1A3	客户端使用此特性通过 ATT 写入请求向服务器发送数据。
TX 额度特性	0xBA04C4B2-892B-43BE-B69C-5D13F2195392	客户端使用此特性通过 ATT 写入请求向服务器发送额度。
Tx 特性	0x0734594A-A8E7-4B1A-A6B1-CD5243059A57	服务器使用此特性通过 ATT 句柄值通知向客户端发送数据。
Rx 额度特性	0xE06D5EFB-4F4A-45C0-9EB1-371AE5A14AD4	服务器使用此特性通过 ATT 句柄值通知向客户端发送额度。

客户端和服务器使用这些特性来发送和接收数据和额度。以下是 SPPLEDemo 作为服务器和 LightBlue 作为客户端的演示。从 App Store 下载 LightBlue 应用并在 iOS 设备上打开蓝牙。

备注

有关特性、ATT 写入请求和 ATT 句柄值通知的更多信息，请参阅蓝牙核心规范（可在蓝牙 SIG 的网站上找到）中的属性协议 (ATT) 和通用属性配置文件 (GATT) 规范

。

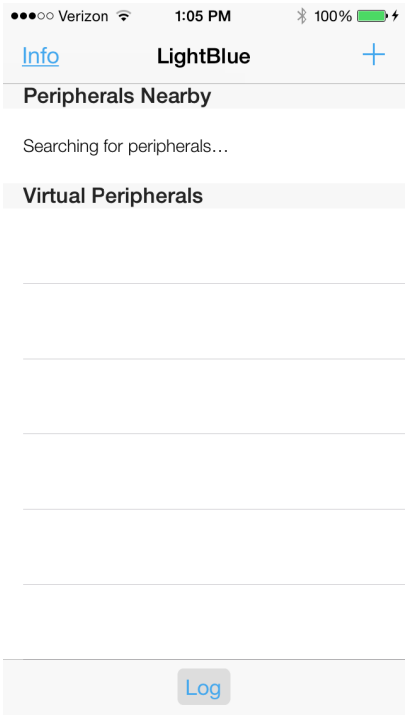
备注

以下说明已在装有 iOS 8.1.3 的 iPhone 5 上运行的 LightBlue 版本 2.2.0 中得到确认。这些指令可以与任何 TI 蓝牙 SDK 中的 SPPLEDemo 应用一起使用，但本例使用 Tiva v1.2 R2 SDK 中的 SPPLEDemo 应用。

5 LightBlue 作为客户端/SPPLEDemo 作为服务器

5.1 连接设备

首先，在设备之间建立连接。为此，请打开 LightBlue 应用，观察类似于以下内容的屏幕：

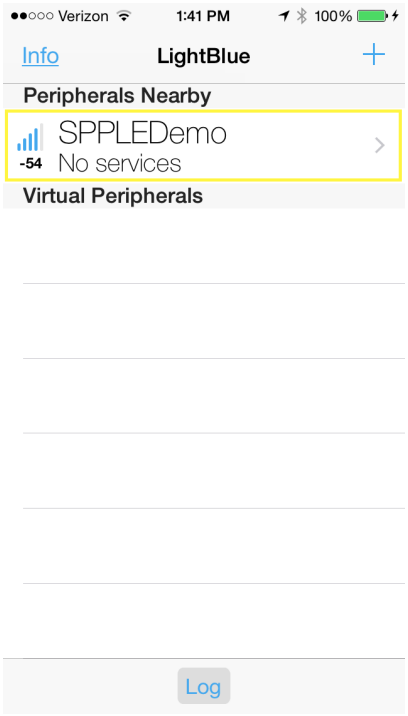


在 SPPLEDemo 终端中，将该应用作为服务器启动，注册 SPP LE 服务，然后使用 Server、RegisterSPPLE 和 AdvertiseLE 1 命令开始广播。在终端中观察到以下内容：

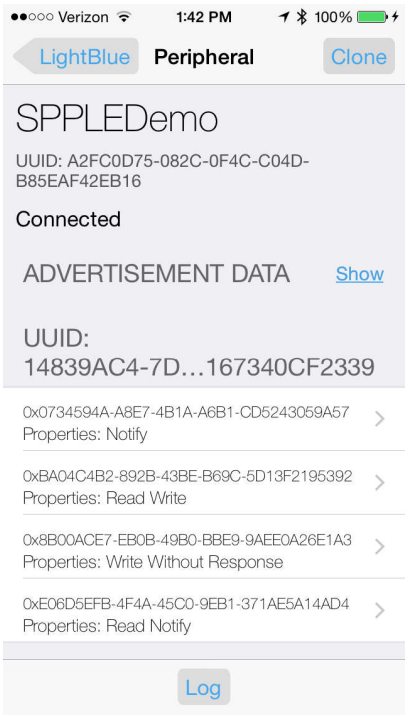
```
OpenStack().
Bluetooth Stack ID: 1.
Device Chipset: 4.1.
BD_ADDR: 0x0017e9d3581a

Command Options: Server, Client, Help

SPP+LE>Server

 Command Options General: Help, GetLocalAddress, SetBaudRate
 Quit,
 Command Options BR/EDR: Inquiry, DisplayInquiryList, Pair,
 EndPairing, PINCodeResponse,
 PassKeyResponse,
 UserConfirmationResponse,
 SetDiscoverabilityMode,
 SetConnectabilityMode,
 SetPairabilityMode,
 ChangeSimplePairingParameters,
 GetLocalName, SetLocalName,
 GetClassOfDevice, SetClassOfDevice,
 GetRemoteName, SniffMode,
 ExitSniffMode, Open, Close, Read,
 Write, GetConfigParams,
 SetConfigParams, GetQueueParams,
 SetQueueParams, Loopback,
 DisplayRawModeData, AutomaticReadMode,
 CBSend.
 Command Options GAPLE: SetDiscoverabilityMode,
 SetConnectabilityMode,
 SetPairabilityMode,
 ChangePairingParameters,
 AdvertiseLE, StartScanning,
 StopScanning, ConnectLE,
 DisconnectLE, PairLE,
```

```
 LEPasskeyResponse,
 QueryEncryptionMode, SetPasskey,
 DiscoverGAPS, GetLocalName,
 SetLocalName, GetLERemoteName,
 SetLocalAppearance,
 GetLocalAppearance,
 GetRemoteAppearance,
 Command Options SPPLE: DiscoverSPPLE, RegisterSPPLE, LESend,
 ConfigureSPPLE, LERead, Loopback,
 DisplayRawModeData, AutomaticReadMode

SPP+LE>RegisterSPPLE
Sucessfully registered SPPLE Service.
SPP+LE>AdvertiseLE 1
GAP_LE_Advertising_Enable success.
```

现在 SPPLEDemo 正在进行广播，请观察 LightBlue 中显示的设备：



接下来，在 LightBlue 中选择 SPPLEDemo 设备，然后观察以下屏幕：



在 SPPLEDemo 终端中，观察以下内容：

```
etLE_Connection_Complete with size 16.
Status: 0x00.
Role: Slave.
Address Type: Random.
BD_ADDR: 0x5cfc3252180b.
SPP+LE>
etGATT_Connection_Device_Connection with size 16:
Connection ID: 2.
Connection Type: LE.
Remote Device: 0x5cfc3252180b.
Connection MTU: 23.
```

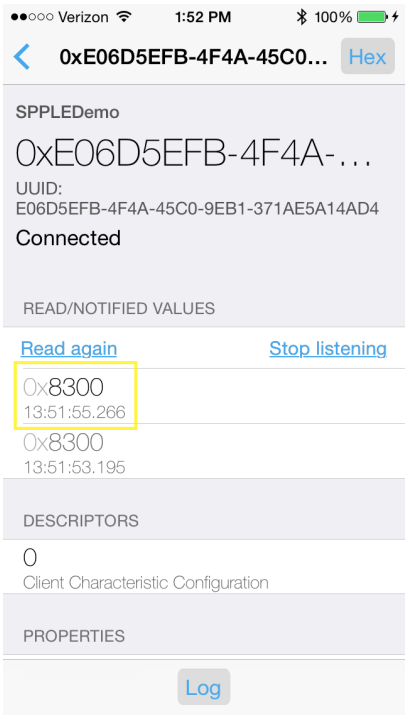
设备现已连接。

5.2 启用通知

要在 LightBlue 中启用有关 Tx 特性和 Rx 额度特性的通知，请执行以下操作：

1. 打开 Tx 特性 (0x0734594A-A8E7-4B1A-A6B1-CD5243059A57)。
2. 选择监听通知。
3. 按左上角的后退按钮。
4. 打开 Rx 额度特性 (0xE06D5EFB-4F4A-45C0-9EB1-371AE5A14AD4)。
5. 选择监听通知。
6. 按左上角的后退按钮。

观察到在对 Rx 额度特性 (0xE06D5EFB-4F4A-45C0-9EB1-371AE5A14AD4) 启用通知后，SPPLEDemo 向 LightBlue 发送初始额度，并观察到应用程序中显示两次 0x8300：



备注

显示 0x8300 的第一个实例是因为 LightBlue 在首次建立连接时自动读取该特性。

备注

这里的数据以小端字节顺序显示，实际额度数为十六进制的 0083，十进制的 131。

5.3 从 LightBlue 中发送数据/在 SPPLEDemo 中接收数据

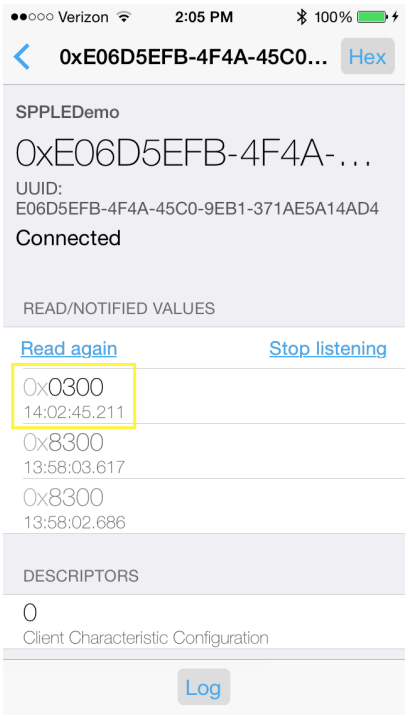
此时，客户端 (LightBlue) 可以向服务器 (SPPLEDemo) 发送数据。要将数据从 LightBlue 发送到 SPPLEDemo，请执行以下操作：

1. 打开 Rx 特性 (0x8B00ACE7-EB0B-49B0-BBE9-9AEE0A26E1A3)。
2. 选择“write new value”。
3. 键入 414243（ASCII 格式的 ABC）。
4. 选择“Done”。

在 SPPLEDemo 终端中，观察数据指示事件。要读取数据，请运行 LERead 5cfc3252180b 命令，然后在终端中观察以下内容：

```
Data Indication Event, Connection ID 1, Received 3 bytes.
SPP+LE>LERead 5cfc3252180b
Read: 3.
ABC
```

打开 Rx 额度特性 (0xE06D5EFB-4F4A-45C0-9EB1-371AE5A14AD4)，观察到 SPPLEDemo 又为 LightBlue 提供了 3 个额度：



5.4 从 SPPLEDemo 中发送数据/在 LightBlue 中接收数据

将数据从 SPPLEDemo 发送到 LightBlue。首先，LightBlue 需要向 SPPLEDemo 提供传输额度。要为 SPPLEDemo 提供传输额度，请在 LightBlue 中执行以下操作：

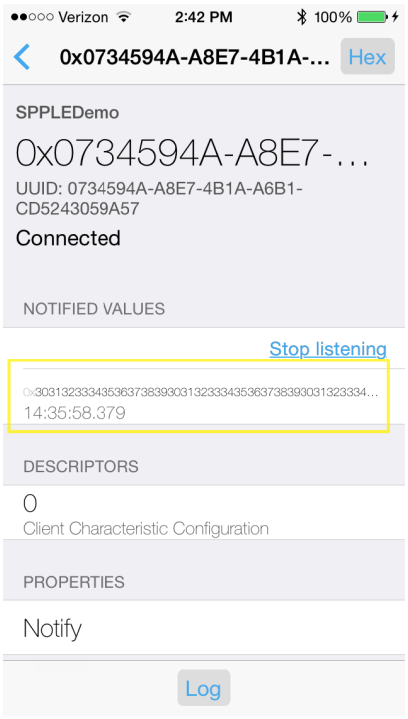
1. 打开 Tx 额度特性 (0xBA04C4B2-892B-43BE-B69C-5D13F2195392)。
2. 选择“Write new value”。
3. 键入 “6400”。（100 个额度= 0x0064 小端）。
4. 选择“done”。
5. 按左上角的后退按钮。

SPPLEDemo 有 100 个额度，可以使用 LESend 5cfc3252180b 100 命令在 SPPLEDemo 中发送数据。在终端中观察到以下内容：

```
SPP+LE>LESend 5cfc3252180b 100
Send Complete, Sent 100.
```

要检查 LightBlue 是否收到数据，请执行以下操作：

1. 打开 Tx 特性 (0x0734594A-A8E7-4B1A-A6B1-CD5243059A57)。
2. 观察“NOTIFIED VALUES”列表中收到的数据，即较长的 0x30313233... 字符串，如下所示：



LightBlue 已收到数据，需要将传输额度返回给 SPPLEDemo。这可以通过重复上述序列并将 0x6400 重新写入 Tx 额度特性 (0xBA04C4B2-892B-43BE-B69C-5D13F2195392) 来完成。

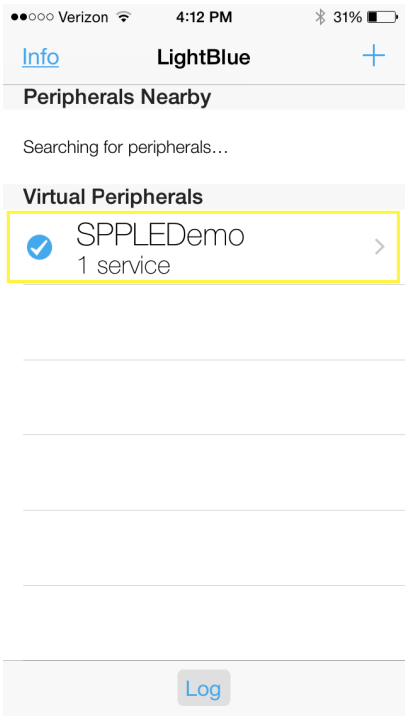
6 LightBlue 作为服务器/SPPLEDemo 作为客户端

备注

承担服务器角色的 LightBlue 不支持显示特性的更新值，即使在写入时也是如此。所以，LightBlue 无法将数据从 LightBlue 发送到 SPPLEDemo，而 SPPLEDemo 能够将数据发送到 LightBlue，但该数据不会显示在应用中。这是 LightBlue 的一个限制。

6.1 连接设备

连接设备的第一步是将 SPP LE 服务和特性添加到 LightBlue 中。要手动执行此操作，请在 LightBlue 中创建一个空白的虚拟外设，然后添加必要的服务和特性。另一个选项是在 SPPLEDemo 充当服务器的同时克隆 SPPLEDemo。要克隆 SPPLEDemo，首先如上所述连接 2 个设备。连接 2 个设备后，选择显示屏右上角的“Clone”选项。应用返回到设备列表，您将观察到 SPPLEDemo 列为虚拟外设，如下所示：



备注

确保选中 SPPLEDemo 左侧的复选框，如图所示。如果未选中，iDevice 未进行广播并且 SPPLEDemo 无法连接。

克隆后，SPP LE 服务现在可以与设备连接。接下来，重新启动 SPPLEDemo，并在出现提示时启动该应用作为客户端。下一步，使用 StartScanning 命令进行扫描以查找 iOS 设备。找到 iOS 设备后，使用 StopScanning 命令停止扫描。现在使用 ConnectLE 5c75524c733a 1 命令连接到 iOS 设备。之后，在 10 秒时间范围内运行 DiscoverSPPLE 5c75524c733a 命令。SPP LE 服务发现完成后，在 25 秒时间范围内运行 ConfigureSPPLE 5c75524c733a。如果命令未在这些时间范围内运行，iOS 设备将断开与 SPPLEDemo 的连接。配置 SPP LE 特性后，2 个应用保持连接，但请注意，如果 iOS 设备进入睡眠状态，则会关闭连接。运行刚刚介绍的命令后，在 SPPLEDemo 的终端中将观察到类似以下内容的输出：

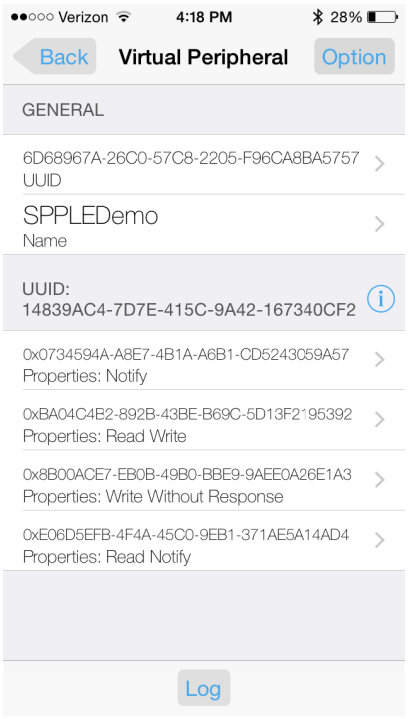
```
OpenStack().
Bluetooth Stack ID: 1.
Device Chipset: 4.1.
BD_ADDR: 0xd03972cdab68

 Command Options: Server, Client, Help

SPP+LE>Client

 Command Options General: Help, GetLocalAddress, SetBaudRate
 Quit,
 Command Options BR/EDR: Inquiry, DisplayInquiryList, Pair,
 EndPairing, PINCodeResponse,
 PassKeyResponse,
 UserConfirmationResponse,
 SetDiscoverabilityMode,
 SetConnectabilityMode,
 SetPairabilityMode,
 ChangeSimplePairingParameters,
 GetLocalName, SetLocalName,
 GetClassOfDevice, SetClassOfDevice,
 GetRemoteName, SniffMode,
 ExitSniffMode, Open, Close, Read,
 Write, GetConfigParams,
 SetConfigParams, GetQueueParams,
 DisplayRawModeData, AutomaticReadMode,
 SetQueueParams, Loopback,
 CBSend.
 Command Options GAPLE: SetDiscoverabilityMode,
 SetConnectabilityMode,
 SetPairabilityMode,
 ChangePairingParameters,
 AdvertiseLE, StartScanning,
 StopScanning, ConnectLE,
 DisconnectLE, PairLE,
 LEPasskeyResponse,
 QueryEncryptionMode, SetPasskey,
 DiscoverGAPS, GetLocalName,
 SetLocalName, GetLERemoteName,
 SetLocalAppearance,
 GetLocalAppearance,
 GetRemoteAppearance,
 Command Options SPPLE: DiscoverSPPLE, RegisterSPPLE, LESend,
 ConfigureSPPLE, LERead, Loopback,
 DisplayRawModeData, AutomaticReadMode

SPP+LE>StartScanning
Scan started successfully.
SPP+LE>
etLE_Advertising_Report with size 36.
1 Responses.
Advertising Type: rtConnectableUndirected.
Address Type: atRandom.
Address: 0x5c75524c733a.
RSSI: -71.
Data Length: 21.
AD Type: 0x01.
AD Length: 0x01.
AD Data: 0x1a
AD Type: 0x07.
AD Length: 0x10.
AD Data: 0x39 0x23 0xcf 0x40 0x73 0x16 0x42 0x9a 0x5c 0x41 0x7e 0x7d 0xc4 0x9a 0x83 0x14
SPP+LE>
etLE_Advertising_Report with size 36.
1 Responses.
Advertising Type: rtScanResponse.
Address Type: atRandom.
Address: 0x5c75524c733a.
RSSI: -71.
Data Length: 11.
AD Type: 0x09.
AD Length: 0x09.
AD Data: 0x53 0x50 0x50 0x4c 0x45 0x44 0x65 0x6d 0x6f
SPP+LE>StopScanning
Scan stopped successfully.
SPP+LE>ConnectLE 5c75524c733a 1
Connection Request successful.
SPP+LE>
```

```
etLE_Connection_Complete with size 16.
Status: 0x00.
Role: Master.
Address Type: Random.
BD_ADDR: 0x5c75524c733a.
SPP+LE>
etGATT_Connection_Device_Connection with size 16:
Connection ID: 1.
Connection Type: LE.
Remote Device: 0x5c75524c733a.
Connection MTU: 23.
SPP+LE>
Exchange MTU Response.
Connection ID: 1.
Transaction ID: 1.
Connection Type: LE.
BD_ADDR: 0x5c75524c733a.
MTU: 131.
SPP+LE>
SPP+LE>DiscoverSPPLE 5c75524c733a
GATT_Start_Service_Discovery success.
SPP+LE>
Service 0x000f - 0x001b, UUID: 14839ac47d7e415c9a42167340cf2339.
SPP+LE>
Service Discovery Operation Complete, Status 0x00.
SPP+LE>ConfigureSPPLE 5c75524c733a
SPPLE Service found on remote device, attempting to read Transmit Credits, and configured CCCDs.
SPP+LE>
Write Response.
Connection ID: 1.
Transaction ID: 15.
Connection Type: LE.
BD_ADDR: 0x5c75524c733a.
Bytes Written: 2.
SPP+LE>
Write Response.
Connection ID: 1.
Transaction ID: 16.
Connection Type: LE.
BD_ADDR: 0x5c75524c733a.
Bytes Written: 2.
```

备注

当 SPPLEDemo 充当服务器时，用户必须使用 LightBlue 应用手动启用通知；但是，当运行 ConfigureSPPLE 命令时，SPPLEDemo 会自动处理启用通知。

现在，2 个设备已连接并配置完毕，设备可以在它们之间发送和接收数据。现在，选择 LightBlue 中的 SPPLEDemo 虚拟外设，以查看虚拟外设的特性。将在 iDevice 的显示屏上观察到以下内容或类似内容：



6.2 从 LightBlue 中发送数据/在 SPPLEDemo 中接收数据

要确认在 ConfigureSPPLE 完成运行后 SPPLEDemo 已向 LightBlue 提供传输额度，请打开 SPPLEDemo 虚拟外设并选择额度特性 (0xBA04C4B2-892B-43BE-B69C-5D13F2195392)。如上所述，在写入更新的特性值时，LightBlue 不会显示这些值，并且用户无法确认 LightBlue 是否收到了数据。即使没有确认 LightBlue 已经收到传输额度，数据仍然可以从 LightBlue 发送到 SPPLEDemo，因为 LightBlue 主要只是 GATT 配置文件演示，对所使用的 SPP LE 协议一无所知。LightBlue 不知道存在传输额度，因此，无论是否有传输额度，数据都可以从 LightBlue 发送到 SPPLEDemo。要将数据发送到 SPPLEDemo，请使用 Tx 特性 (0x0734594A-A8E7-4B1A-A6B1-CD5243059A57') 并在 LightBlue 中执行以下操作：

1. 打开 Tx 特性并选择“No value/hex”选项。
2. 键入 414243。
3. 选择“Done”。

在 SPPLEDemo 中观察数据指示。要读取数据，请使用 LERead 5c75524c733a 命令。观察终端中显示的 ABC，如下所示：

```
Data Indication Event, Connection ID 1, Received 3 bytes.
SPP+LE>LERead 5c75524c733a
Read: 3.
ABC
```

6.3 从 SPPLEDemo 中发送数据/在 LightBlue 中接收数据

备注

如前所述，LightBlue 不支持在写入时显示更新的特性值。目前无法确认 LightBlue 是否已收到数据。

要从 SPPLEDemo 发送数据，LightBlue 必须首先提供额度。这可以在 LightBlue 中使用以下过程来实现：

- 打开 Rx 额度特性 (0xE06D5EFB-4F4A-45C0-9EB1-371AE5A14AD4)。
- 键入 6400。（100 个额度= 0x0064 小端）
- 选择“Done”。

SPPLEDemo 现在有 100 个传输额度。接下来，要在 SPPLEDemo 中发送数据，请使用 LESend 5c75524c733a 100 命令。在终端中观察到以下内容。

```
SPP+LE>LESend 5c75524c733a 100
Send Complete, Sent 100.
```

7 应用程序命令

TI 的 Bluetooth Stack 用于实现蓝牙协议栈的上层。TI 的 Bluetooth Stack 是一款强大而灵活的软件开发工具，可在主机控制器接口 (HCI) 之上实现蓝牙协议和配置文件。TI 的 Bluetooth Stack 应用程序编程接口 (API) 可用于访问上层协议和配置文件，并且可以直接与蓝牙芯片连接。

MSP-EXP430F5438、Tiva DK-TM4C129X、MSP432 和 STM32F4 附带的基本蓝牙应用程序是一个串行端口配置文件应用程序。

另请查看以下文档：基于 MSP432™ MCU 的双模 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 文档详细介绍了所有 API。

8 常规命令

8.1 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。此命令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。

8.2 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，才能尝试调用此命令。

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
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_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）。

8.3 设置波特率

说明

SetBaudRate 命令负责更改当前用于与无线电通信的波特率。此函数仅配置 TI 蓝牙芯片组的波特率。此命令要求存在有效的蓝牙协议栈 ID。

参数

此命令需要一个参数。该值是一个整数，表示用于波特率的值。选项为 0（表示波特率为 115200）、1（表示波特率为 230400）、2（表示波特率为 460800）、3（表示波特率为 921600）、4（表示波特率为 1843200）或 5（表示波特率为 3686400）。最大波特率默认值为 921600，因此选项 4 和 5 被禁用。

命令调用示例

- “SetBaudRate 0”尝试将波特率设置为 115200。
- “SetBaudRate 1”尝试将波特率设置为 230400。
- “SetBaudRate 2”尝试将波特率设置为 460800。
- “SetBaudRate 3”尝试将波特率设置为 921600。

可能的返回值

- (0) 成功设置了波特率
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID

API 调用

HCI_Reconfigure_Driver(BluetoothStackID, FALSE, &(Data.DriverReconfigureData))

API 原型

*int BTPSAPI HCI_Reconfigure_Driver(unsigned int BluetoothStackID,Boolean_t ResetStateMachines,HCI_Driver_Reconfigure_Data_t *DriverReconfigureData)*

API 说明

此函数向 HCI 驱动程序发出适当的调用，以请求 HCI 驱动程序使用相应的配置信息重新配置自身。

8.4 退出

使用此命令返回到初始命令屏幕。

9 BR/EDR 命令

有关 BR/EDR 命令，请参阅文档 SPP 配置文件 (http://processors.wiki.ti.com/index.php/CC256x_MSP430_TI's_Bluetooth_Stack_Basic_SPPDemo_APP) 中的以下部分：“GenericAccess 配置文件命令”和“串行端口配置文件命令”。

10 GAPLE 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。

10.1 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。低功耗模式下的可发现性模式仅在广播时适用。如果设备未在广播，则该设备不可发现。此命令设置的值用作命令 AdvertiseLE 中的一个参数。

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) 成功设置了可发现性模式参数
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR

10.2 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。低功耗模式下的可连接性模式仅在广播时适用。如果设备未在广播，则该设备非连接。此命令设置的值用作命令 AdvertiseLE 中的一个参数。

参数

该命令仅需要一个参数，该参数是一个表示可连接性模式的整数值。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) 成功设置了可连接性模式参数
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR

10.3 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。

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（表示可安全简易配对）。

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。

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

API 调用

GAP_LE_Set_Pairability_Mode(BluetoothStackID, PairabilityMode)

API 原型

int BTPSAPI GAP_LE_Set_Pairability_Mode(unsigned int BluetoothStackID, GAP_LE_Pairability_Mode_t PairableMode)

API 说明

提供此函数是为了让本地主机能够更改本地主机使用的可配对性模式。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。

10.4 更改配对参数

说明

ChangePairingParameters 命令负责更改在配对过程中交换的 LE 配对参数。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。

参数

该命令需要五个参数，分别是 I/O 功能、绑定类型、MITM 要求、SC 启用和 P256 调试模式：

1. 第一个参数必须指定为 0（对应于“仅显示器”）、1（对应于“显示器是/否”）、2（对应于“仅键盘”）、3（对应于“无输入/输出”）或 4（对应于“键盘/显示器”）。
2. 第二个参数必须指定为 0（对应于“无绑定”）或 1（对应于“绑定”），当至少一个设备设置为“无绑定”时，将不会存储 LTK。
3. 第三个参数必须指定为 0（对应于“无 MITM”）或 1（对应于“需要 MITM”）。
4. 第四个参数必须指定为 0（对应于”SC 禁用“）或 1（对应于”SC 启用“），使用”SC 禁用“时会发生传统配对过程。

5. 第五个参数必须指定为 0（对应于”禁用调试模式“）或 1（对应于”启用 P256 调试模式“），但仅限于使用 SC 配对时。P256 调试模式在设置后是相关的，P256 私钥和公钥的值是根据蓝牙规范预先定义的，而不是随机的。

命令调用示例

- “ChangeSimplePairingParameters 3 0 0 0 0”尝试将“I/O 功能”设置为“无输入/输出”、“绑定类型”设置为“无绑定”、关闭 MITM 保护、禁用安全连接并禁用调试模式。
- “ChangeSimplePairingParameters 2 0 1 1 0”尝试将“I/O 功能”设置为“仅键盘”，将“绑定类型”设置为“无绑定”，激活 MITM 保护，启用安全连接并禁用调试模式。
- “ChangeSimplePairingParameters 1 1 1 1 1”尝试将“I/O 功能”设置为“显示器是/否”、将“绑定类型”设置为“绑定”、激活 MITM 保护、启用安全连接并启用调试模式。

可能的返回值

- (0) 成功设置可配对性模式
- (-6) INVALID_PARAMETERS_ERROR
- (-8) INVALID_STACK_ID_ERROR

10.5 广播 LE

说明

AdvertiseLE 命令负责启用 LE 广播。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。

参数

唯一必需的参数决定是发送还是禁用广播报告。要禁用，请使用 0 作为第一个参数；要启用，请改用 1。

命令调用示例

- “AdvertiseLE 1”尝试在本地蓝牙设备上启用低功耗广播。
- “AdvertiseLE 0”尝试在本地蓝牙设备上禁用低功耗广播。

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
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

根据第一个参数值：

- *GAP_LE_Advertising_Disable(BluetoothStackID)*
- *GAP_LE_Set_Advertising_Data(BluetoothStackID, (Advertisement_Data_Buffer.AdvertisingData.Advertising_Data[0] + 1), &(Advertisement_Data_Buffer.AdvertisingData))*
- *GAP_LE_Set_Scan_Response_Data(BluetoothStackID, (Advertisement_Data_Buffer.ScanResponseData.Scan_Response_Data[0] + 1), &(Advertisement_Data_Buffer.ScanResponseData))*

- *GAP_LE_Advertising_Enable(BluetoothStackID, TRUE, &AdvertisingParameters, &ConnectabilityParameters, GAP_LE_Event_Callback, 0)*

API 原型

- *int BTPSAPI GAP_LE_Advertising_Disable(unsigned int BluetoothStackID)*
- *int BTPSAPI GAP_LE_Set_Advertising_Data(unsigned int BluetoothStackID, unsigned int Length, Advertising_Data_t *Advertising_Data)*
- *int BTPSAPI GAP_LE_Set_Scan_Response_Data(unsigned int BluetoothStackID, unsigned int Length, Scan_Response_Data_t *Scan_Response_Data)*
- *int BTPSAPI GAP_LE_Set_Advertising_Data(unsigned int BluetoothStackID, unsigned int Length, Advertising_Data_t *Advertising_Data)*
- *int BTPSAPI GAP_LE_Set_Advertising_Data(unsigned int BluetoothStackID, unsigned int Length, Advertising_Data_t *Advertising_Data)*

API 说明

- 提供的 GAP_LE_Advertising_Disable 函数使本地主机能够取消（停止）正在进行的广播过程。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。
- 提供的 GAP_LE_Set_Advertising_Data 使本地主机能够设置在广播过程（通过 GAP_LE_Advertising_Enable 函数启动）中使用的广播数据。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。
- 提供的 GAP_LE_Set_Scan_Response_Data 函数使本地主机能够设置在广播过程（通过 GAP_LE_Advertising_Enable 函数启动）中使用的广播数据。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。
- 提供的 GAP_LE_Set_Advertising_Data 函数使本地主机能够设置在广播过程（通过 GAP_LE_Advertising_Enable 函数启动）中使用的广播数据。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。

10.6 启动扫描

说明

StartScanning 命令负责启动 LE 扫描过程。如果成功，此命令将返回零；如果发生错误，则返回负值。该命令调用用于执行扫描的 StartScan（在 BluetoothStackID 中未签名）函数。

参数

使用此命令时不需要包含参数。参数对扫描的结果没有影响。

可能的返回值

- (0) 成功启动了 LE 扫描过程
- (-1)蓝牙协议栈 ID 在 StartScan() 调用期间无效
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-66) BTPS_ERROR_INSUFFICIENT_RESOURCES
- (-105) BTPS_ERROR_SCAN_ACTIVE
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_LE_Perform_Scan(BluetoothStackID, stActive, 10, 10, latPublic, fpNoFilter, TRUE, GAP_LE_Event_Callback, 0)

API 原型

int BTPSAPI GAP_LE_Perform_Scan(unsigned int BluetoothStackID, GAP_LE_Scan_Type_t ScanType, unsigned int ScanInterval, unsigned int ScanWindow,GAP_LE_Address_Type_t LocalAddressType, GAP_LE_Filter_Policy_t FilterPolicy, Boolean_t FilterDuplicates, GAP_LE_Event_Callback_t GAP_LE_Event_Callback, unsignedlong CallbackParameter)

API 说明

提供 GAP_LE_Perform_Scan 函数是为了让本地主机能够开始 LE 扫描过程。此过程在概念上类似于蓝牙 BR/EDR 中的查询过程，因为这可用于发现已被指示进行广播的设备。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。

10.7 停止扫描

说明

StopScanning 命令负责停止 LE 扫描过程。如果成功，此命令将返回零；如果发生错误，则返回负值。该命令调用用于停止扫描的 StopScan（在 BluetoothStackID 中未签名）函数。

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

- (0) 成功停止了 LE 扫描过程
- (-1) 蓝牙协议栈 ID 在 StartScan() 调用期间无效
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR

API 调用

GAP_LE_Cancel_Scan(BluetoothStackID)

API 原型

int BTPSAPI GAP_LE_Cancel_Scan(unsigned int BluetoothStackID)

API 说明

提供 GAP_LE_Cancel_Scan 函数是为了让本地主机能够取消（停止）正在进行的扫描过程。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。

10.8 连接 LE

说明

ConnectLE 命令负责连接到 LE 设备。如果成功，此命令将返回零；如果发生错误，则返回负值。此命令使用 ConnectLEDevice (BluetoothStackID, BD_ADDR, FALSE) 调用 ConnectLEDevice (unsigned in BluetoothStackID, BD_ADDR_t BD_ADDR, Boolean_t UseWhiteList) 函数。

参数

唯一需要的参数是远程设备的蓝牙地址。如果在扫描期间广播设备位于附近，则可以使用 StartScanning 命令轻松找到该地址。

命令调用示例

- “ConnectLE 001bdc05b617”尝试向 BD_ADDR 为 001bdc05b617 的蓝牙设备发送连接请求。
- “ConnectLE 000275e126FF”尝试向 BD_ADDR 为 000275e126FF 的蓝牙设备发送连接请求。

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
- (-116) BTPS_ERROR_RANDOM_ADDRESS_IN_USE
- (-111) BTPS_ERROR_CREATE_CONNECTION_OUTSTANDING
- (-66) BTPS_ERROR_INSUFFICIENT_RESOURCES
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- GAP_LE_ERROR_WHITE_LIST_IN_USE

API 调用

- *GAP_LE_Create_Connection(BluetoothStackID, 100, 100, Result?fpNoFilter:fpWhiteList, latPublic, Result? &BD_ADDR:NULL, latPublic, &ConnectionParameters,GAP_LE_Event_Callback, 0)*
- *GAP_LE_Remove_Device_From_White_List(BluetoothStackID, 1, &WhiteListEntry, &WhiteListChanged)*
- *GAP_LE_Add_Device_To_White_List(BluetoothStackID, 1, &WhiteListEntry, &WhiteListChanged)*

备注

通常可以忽略这两个 API，除非在对 ConnectLEDevice 的调用中启用了白名单。

API 原型

- *int BTPSAPI GAP_LE_Create_Connection(unsigned int BluetoothStackID, unsigned int ScanInterval, unsigned int ScanWindow, GAP_LE_Filter_Policy_t InitatorFilterPolicy,GAP_LE_Address_Type_t RemoteAddressType, BD_ADDR_t *RemoteDevice, GAP_LE_Address_Type_t LocalAddressType, GAP_LE_Connection_Parameters_t*ConnectionParameters, GAP_LE_Event_Callback_t GAP_LE_Event_Callback, unsigned long CallbackParameter)*
- *int BTPSAPI GAP_LE_Remove_Device_From_White_List(unsigned int BluetoothStackID, unsigned int DeviceCount, GAP_LE_White_List_Entry_t *WhiteListEntries, unsigned int*RemovedDeviceCount)*
- *int BTPSAPI GAP_LE_Add_Device_To_White_List(unsigned int BluetoothStackID, unsigned int DeviceCount, GAP_LE_White_List_Entry_t *WhiteListEntries, unsigned int*AddedDeviceCount)*

API 说明

提供的 GAP_LE_Create_Connection 函数使本地主机能够使用低功耗蓝牙无线电创建与远程设备的连接。连接过程本质上是异步的，当连接完成时，将通过 GAP LE 事件回调函数（在目前讨论的这个函数中指定）通知调用方。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。提供的 GAP_LE_Remove_Device_From_White_List 函数使本地主机能够从本地设备维护的白名单中删除一个（或多个）设备。此函数尝试（从指定的列表中）删除尽可能多的设备，并返回已删除的设备数量。
GAP_LE_Read_White_List_Size 函数可用于确定本地设备在白名单中（同时）支持多少设备。

10.9 断开 LE

说明

DisconnectLE 命令负责断开与 LE 设备的连接。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。此命令要求在运行之前存在有效的蓝牙协议栈 ID。

参数

唯一需要的参数是所连接的远程设备的蓝牙地址。

命令调用示例

- “DisconnectLE 001bdc05b617”尝试向 BD_ADDR 为 001bdc05b617 的蓝牙设备发送断开连接请求。
- “DisconnectLE 000275e126FF”尝试向 BD_ADDR 为 000275e126FF 的蓝牙设备发送断开连接请求。

可能的返回值

- (0) 成功断开了远程设备连接
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR

API 调用

GAP_LE_Disconnect(BluetoothStackID, BD_ADDR)

API 原型

int BTPSAPI GAP_LE_Disconnect(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR)

API 说明

GAP_LE_Disconnect 函数提供与远程设备断开连接的功能。如果成功，此函数返回零；如果出现错误情况，则返回负的错误代码。

10.10 LE 配对

说明

提供 PairLE 命令是为了与连接的设备进行配对（或者，如果是从设备，则请求安全性）。该命令使用 SendPairingRequest (ConnectionBD_ADDR, LocalDeviceIsMaster) 调用 SendPairingRequest (BD_ADDR_tBD_ADDR, Boolean_t ConnectionMaster) 函数。

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

- (0) 成功设置可配对性模式
- (-4) FUNCTION_ERROR
- (-6) INVALID_PARAMETERS_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE
- (-66) BTPS_ERROR_INSUFFICIENT_RESOURCES
- (-107) BTPS_ERROR_INVALID_DEVICE_ROLE_MODE

API 调用

- *GAP_LE_Pair_Remote_Device(BluetoothStackID, BD_ADDR, &Capabilities, GAP_LE_Event_Callback, 0)*
- *GAP_LE_Request_Security(BluetoothStackID, BD_ADDR, Capabilities.Bonding_Type, Capabilities.MITM, GAP_LE_Event_Callback, 0)*

API 原型

- *int BTPSAPI GAP_LE_Pair_Remote_Device(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_LE_Pairing_Capabilities_t *Capabilities, GAP_LE_Event_Callback_tGAP_LE_Event_Callback, unsigned long CallbackParameter)*
- *int BTPSAPI GAP_LE_Request_Security(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_LE_Bonding_Type_t Bonding_Type, Boolean_t MITM,GAP_LE_Event_Callback_t GAP_LE_Event_Callback, unsigned long CallbackParameter)*

API 说明

提供 GAP_LE_Pair_Remote_Device 函数是为了能够与连接的远程设备配对。该函数将以下内容作为输入：当前连接的要配对的设备的地址以及本地设备的配对功能。此函数还将要在配对过程中使用的 GAP LE 事件回调信息作为输入。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。该函数只能由连接的主设备（连接的发起方）发出。原因是从设备只能请求安全过程，而不能发起安全过程。提供 GAP_LE_Request_Security 函数是为了让从设备能够请求（连接的）主设备执行配对操作或重新建立先前的安全性。该函数只能由从设备调用。原因是从设备只能请求发起安全性，它无法自行发起安全过程。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。

10.11 LE 通行密钥响应

说明

LEPassKeyResponse 命令负责使用由输入参数指定的通行密钥值发出 GAP 身份验证响应。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。

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
- (-118) BTPS_ERROR_PAIRING_NOT_ACTIVE
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE
- (-66) BTPS_ERROR_INSUFFICIENT_RESOURCES
- (-107) BTPS_ERROR_INVALID_DEVICE_ROLE_MODE

API 调用

GAP_LE_Authentication_Response(BluetoothStackID, CurrentRemoteBD_ADDR, &GAP_LE_Authentication_Response_Information)

API 原型

*int BTPSAPI GAP_LE_Authentication_Response(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_LE_Authentication_Response_Information_t*GAP_LE_Authentication_Information)*

API 说明

提供此函数是为了能够让本地设备响应 GAP LE 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）。

10.12 LE 查询加密

说明

LEQueryEncryption 命令负责查询 LE 连接的加密模式。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。

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) 成功查询了加密模式
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE

API 调用

GAP_LE_Query_Encryption_Mode(BluetoothStackID, ConnectionBD_ADDR, &GAP_Encryption_Mode)

API 原型

*int BTPSAPI GAP_LE_Query_Encryption_Mode(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_Encryption_Mode_t *GAP_Encryption_Mode)*

API 说明

提供此函数是为了能够查询所指定 LE 连接的当前加密模式。

10.13 设置通行密钥

说明

SetPasskey 命令负责查询 LE 连接的加密模式。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。

备注

SetPasskey 命令仅在配对时有效。

参数

SetPasskey 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

- “SetPasskey 0”尝试删除通行密钥。
- “SetPasskey 1 987654”尝试将通行密钥设置为 987654。
- “SetPasskey 1”尝试将通行密钥设置为默认的固定通行密钥值。

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
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE

API 调用

根据第一个参数选择其中一个：

- *GAP_LE_Set_Fixed_Passkey(BluetoothStackID, &Passkey)*
- *GAP_LE_Set_Fixed_Passkey(BluetoothStackID, NULL)*

API 原型

*int BTPSAPI GAP_LE_Set_Fixed_Passkey(unsigned int BluetoothStackID, DWord_t *Fixed_Display_Passkey)*

API 说明

提供此函数是为了能够在配对操作期间，每当选择本地蓝牙设备显示通行密钥时，可以使用固定密钥。仅当根据远程 I/O 功能和本地 I/O 功能选择本地蓝牙设备来显示通行密钥时，才会使用此固定密钥。

10.14 发现 GAPS

说明

提供的 DiscoverGAPS 命令可通过一种简单的机制来启动服务发现过程，以发现所连接的远程设备上的通用访问配置文件服务。

参数

使用此命令时不需要包含参数。参数对服务发现的结果没有影响。

可能的返回值

- (0) 成功发现了通用访问配置文件服务。
- (-4) 函数错误（失败时）。

API 调用

GDIS_Service_Discovery_Start(BluetoothStackID, ConnectionID, (sizeof(UUID)/sizeof(GATT_UUID_t)), UUID, GDIS_Event_Callback, sdGAPS)

API 原型

*int BTPSAPI GDIS_Service_Discovery_Start(unsigned int BluetoothStackID, unsigned int ConnectionID, unsigned int NumberOfUUID, GATT_UUID_t *UUIDList,GDIS_Event_Callback_t ServiceDiscoveryCallback, unsigned long ServiceDiscoveryCallbackParameter)*

API 说明

GDIS_Service_Discover_Start 位于名为 GDIS 的应用程序模块中，提供的该模块旨在以简单的方式执行 GATT 服务发现。此模块可以修改以供客户使用。GDIS 模块调用此函数来启动服务发现操作。

10.15 获取本地名称

说明

GetLocalName 命令负责查询本地蓝牙设备的名称。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，才能尝试调用此命令。

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。该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 参数至少为 MAX_NAME_LENGTH+1，以保存允许的最长设备名称加上用于保留 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）。

10.16 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，才能尝试调用此命令。

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（这意味着名称中不应有空格。如果名称中包含空格，则仅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。

10.17 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。此命令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
- (-6) INVALID_PARAMETERS_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，则在已确定远程名称信息（或如果存在错误）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。取消回调时，仍会尝试执行此操作，然后取消回调（即 GAP 模块仍可以执行远程名称请求，但从不发出回调）。

10.18 LE 用户确认响应

说明

LEUserConfirmationResponse 命令负责发出 GAP LE 身份验证响应，其用户确认值通过输入参数指定。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。

参数

此命令需要一个参数来指示是否接受确认。0 = 拒绝，1 = 接受。

命令调用示例

- “LEUserConfirmationResponse 0”尝试以拒绝值进行响应。
- “LEUserConfirmationResponse 1”尝试以接受值进行响应。

可能的返回值

- (0) 成功。
- (-4) FUNCTION_ERROR。
- (-6) INVALID_PARAMETERS_ERROR.
- (-1) BTPS_ERROR_INVALID_PARAMETER。
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID。
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED。
- (-57) BTPS_ERROR_DEVICE_HCI_ERROR。
- (-66) BTPS_ERROR_INSUFFICIENT_RESOURCES。
- (-98) BTPS_ERROR_DEVICE_NOT_CONNECTED。
- (-103) BTPS_ERROR_FEATURE_NOT_AVAILABLE。
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE。
- (-107) BTPS_ERROR_INVALID_DEVICE_ROLE_MODE。
- (-118) BTPS_ERROR_PAIRING_NOT_ACTIVE。
- (-119) BTPS_ERROR_INVALID_STATE。

- (-120) BTPS_ERROR_FEATURE_NOT_CURRENTLY_ACTIVE。
- (-122) BTPS_ERROR_NUMERIC_COMPARISON_FAILED。

API 调用

GAP_LE_Authentication_Response(BluetoothStackID, CurrentLERemoteBD_ADDR, &GAP_LE_Authentication_Response_Information)

API 原型

*int BTPSAPI GAP_LE_Authentication_Response(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_LE_Authentication_Response_Information_t*GAP_LE_Authentication_Information)*

说明

提供以下函数是为了让本地设备能够响应 GAP LE 身份验证事件。此函数用于设置指定蓝牙设备的身份验证信息。该函数将以下内容作为输入：蓝牙协议栈 ID，当前正在执行配对/身份验证过程的远程蓝牙设备地址，以及身份验证响应信息。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。

10.19 启用仅 SC

说明

EnableSCOnly 命令启用仅 LE 安全连接 (SC) 模式。如果启用此模式，则拒绝来自仅支持传统配对的对等方的配对请求。请注意，如果启用此模式，LE_Parameters 中的 SC 标志必须设置为 TRUE。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

此命令需要一个参数，该参数指示是否设置仅安全连接模式。0 = 仅 SC 模式关闭，1 = 仅 SC 模式打开。

命令调用示例

- “EnableSCOnly 0”禁用仅安全连接模式。
- “EnableSCOnly 1”启用仅安全连接模式。

可能的返回值

- (0) 成功。
- (-4) FUNCTION_ERROR。
- (-6) INVALID_PARAMETERS_ERROR。
- (-8) INVALID_STACK_ID_ERROR。
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID。
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED。
- (-103) BTPS_ERROR_FEATURE_NOT_AVAILABLE。
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE。
- (-120) BTPS_ERROR_FEATURE_NOT_CURRENTLY_ACTIVE。

API 调用

GAP_LE_SC_Only_Mode(BluetoothStackID, EnableSCOnly)

API 原型

int BTPSAPI GAP_LE_SC_Only_Mode (unsigned int BluetoothStackID, Boolean_t EnableSCOnly)

API 说明

提供的以下函数允许配置仅 LE 安全连接模式。上层在 LE SC 配对开始之前使用此函数，以防止该函数要求拒绝仅支持传统配对的设备。当设备具有高安全性比设备与不支持 SC 的设备保持向后兼容性更重要时，可以使用该模式。和该函数将以下内容作为参数：蓝牙设备的蓝牙协议栈 ID 以及启用或禁用仅 SC 模式的布尔值 EnableSCOnly。此函数在第一次配对过程之前使用一次。如果成功，此函数返回零；否则返回负的错误代码。

10.20 重新生成 P256 本地密钥

说明

以下函数允许用户生成新的 P256 私钥和本地密钥。该函数不得在配对过程中使用。这仅适用于 LE SecureConenctions 配对。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

无需任何参数。

命令调用示例

“RegenerateP256LocalKeys”尝试生成新的 P256 私钥和本地密钥。

可能的返回值

- (0) 成功。
- (-4) FUNCTION_ERROR。
- (-8) INVALID_STACK_ID_ERROR。
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID。
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED。
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE。
- (-117) BTPS_ERROR_PAIRING_ACTIVE。
- (-120) BTPS_ERROR_FEATURE_NOT_CURRENTLY_ACTIVE。

API 调用

GAP_LE_SC_Regenerate_P256_Local_Keys (BluetoothStackID)

API 原型

int BTPSAPI GAP_LE_SC_Regenerate_P256_Local_Keys (unsigned int BluetoothStackID)

API 说明

提供以下函数是为了能够重新生成 P-256 私钥和本地公钥。此函数仅适用于 LE SC 配对的情况。该函数将蓝牙设备的蓝牙协议栈 ID 作为参数。执行配对时不得使用此函数。如果成功，此函数返回零；否则返回负的错误代码。

10.21 SC 生成 OOB 本地参数

说明

要使用 OOB 方法执行 LE SC 配对，请在配对过程开始之前生成本地随机值和确认值。以下函数允许用户生成 OOB 本地参数。该函数不得在配对过程中使用。这仅适用于 LE SC 配对。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

无需任何参数。

命令调用示例

“SCGenerateOOBLocalParams”尝试在配对过程开始前生成本地随机值和确认值。

可能的返回值

- (0) 成功。
- (-4) FUNCTION_ERROR。
- (-8) INVALID_STACK_ID_ERROR。
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID。
- (-56) BTPS_ERROR_GAP_NOT_INITIALIZED。
- (-104) BTPS_ERROR_LOCAL_CONTROLLER_DOES_NOT_SUPPORT_LE。
- (-117) BTPS_ERROR_PAIRING_ACTIVE。
- (-120) BTPS_ERROR_FEATURE_NOT_CURRENTLY_ACTIVE。

API 调用

GAP_LE_SC_OOB_Generate_Parameters(BluetoothStackID, &OOBLocalRandom, &OOBLocalConfirmation)

API 原型

*int BTPSAPI GAP_LE_SC_OOB_Generate_Parameters(unsigned int BluetoothStackID, SM_Random_Value_t *OOB_Local_Rand_Result, SM_Confirm_Value_t*OOB_Local_Confirm_Result)*

API 说明

提供以下函数是为了能够在带外 (OOB) 关联方法中使用 LE 安全连接 (SC) 配对。上层使用此函数生成蓝牙规范中定义的本地 OOB 随机值和 OOB 确认值（ra/rb 和 Ca/Cb）。该函数将以下内容作为参数：蓝牙设备的蓝牙协议栈 ID 以及指向缓冲区的指针（用于接收生成的本地 OOB 随机值和 OOB 确认值）。如果成功，此函数返回零；否则返回负的错误代码。

10.22 设置本地外观

说明

提供 SetLocalAppearence 命令是为了设置由 GAP 服务 (GAP) 公开的本地设备外观。

参数

SetLocalAppearence 命令需要的一个参数是本地设备外观。

可能的返回值

- (0) 成功。
- (-4) 函数错误（失败时）。

API 调用

GAPS_Set_Device_Appearance(BluetoothStackID, GAPSInstanceID, Appearance)

API 原型

int BTPSAPI GAPS_Set_Device_Appearance (unsigned int BluetoothStackID, unsigned int InstanceID, Word_t DeviceAppearance)

API 说明

此函数旨在通过一种机制来设置作为 GAP 服务 API (GAPS) 的一部分公开的本地设备外观。

10.23 获取本地外观

说明

提供的 GetLocalAppearence 命令用于读取由 GAP 服务 (GAP) 公开的本地设备外观。

参数

使用此命令时不需要包含参数。参数对结果没有影响。

可能的返回值

- (0) 成功。
- (-4) 函数错误（失败时）。

API 调用

GAPS_Query_Device_Appearance(BluetoothStackID, GAPSInstanceID, &Appearance)

API 原型

*int BTPSAPI GAPS_Query_Device_Appearance(unsigned int BluetoothStackID, unsigned int InstanceID, Word_t *DeviceAppearance)*

API 说明

此函数旨在通过一种机制读取作为 GAP 服务 API (GAPS) 的一部分公开的本地设备外观。

11 SPPLE 命令

11.1 发现 SPPLE

说明

以下函数负责执行 SPPLE 服务发现操作。此函数在成功执行时返回零，而在出现错误时返回负值。

参数

唯一需要的参数是所连接的远程设备的蓝牙地址。

命令调用示例

- “DiscoverSPPLE 001bdc05b617”尝试发现 BD_ADDR 为 001bdc05b617 的蓝牙设备的服务。
- “DiscoverSPPLE 000275e126FF”尝试发现 BD_ADDR 为 000275e126FF 的蓝牙设备的服务。

可能的返回值

- (0) 成功启动了 SPP LE 服务发现。
- (-4) 函数错误（失败时）。

API 调用

GDIS_Service_Discovery_Start(BluetoothStackID, ConnectionID, (sizeof(UUID)/sizeof(GATT_UUID_t)), UUID, GDIS_Event_Callback, 0)

API 原型

*int BTPSAPI GDIS_Service_Discovery_Start(unsigned int BluetoothStackID, unsigned int ConnectionID, unsigned int NumberOfUUID, GATT_UUID_t *UUIDList,GDIS_Event_Callback_t ServiceDiscoveryCallback, unsigned long ServiceDiscoveryCallbackParameter)*

API 说明

GDIS_Service_Discover_Start 位于名为 GDIS 的应用程序模块中，提供的该模块旨在以简单的方式执行 GATT 服务发现。GDIS 模块调用此函数来启动服务发现操作。

11.2 注册 SPPLE

说明

以下函数负责注册 SPPLE 服务。此函数在成功执行时返回零，而在出现错误时返回负值。

参数

使用此命令时不需要包含参数。参数对注册 SPPLE 服务的结果没有影响。

可能的返回值

- (0) 成功注册了 SPPLE 服务。
- (-4) 函数错误（失败时）。

API 调用

*GATT_Register_Service(BluetoothStackID, SPPLE_SERVICE_FLAGS, SPPLE_SERVICE_ATTRIBUTE_COUNT, (GATT_Service_Attribute_Entry_t *)SPPLE_Service,&ServiceHandleGroup, GATT_ServerEventCallback, 0)*

API 原型

*int BTPSAPI GATT_Register_Service(unsigned int BluetoothStackID, Byte_t ServiceFlags, unsigned int NumberOfServiceAttributeEntries, GATT_Service_Attribute_Entry_t*ServiceTable, GATT_Attribute_Handle_Group_t *ServiceHandleGroupResult, GATT_Server_Event_Callback_t ServerEventCallback, unsigned long CallbackParameter)*

API 说明

提供以下函数是为了能够将 GATT 服务添加到本地 GATT 数据库。第一个参数是蓝牙设备的蓝牙协议栈 ID。第二个参数是一个位掩码字段，用于指定正在注册的服务类型，该字段必须为非零（即必须设置至少一位）。第三个参数是第四个参数指向的服务属性数组中的条目数。第四个参数是一个数组，其中包含正在注册的服务的属性。下一个参数是指向一个缓冲区的指针，该缓冲区存储已注册服务的属性句柄范围。最后两个参数指定 GATT 服务器回调和回调参数，只要本地 GATT 模块无法在内部满足客户端对 GATT 服务器的请求，就可以使用此参数。如果成功，此函数将返回一个正的非零服务 ID；如果出现错误，则返回一个负的错误代码。如果此函数成功返回，则 ServiceHandleGroupResult 缓冲区包含服务的属性句柄范围。

11.3 LE 发送

说明

以下函数负责向存在连接的远程设备发送多个字符。该函数接收一个参数，该参数指示要传输的字节数。此函数在成功执行时返回零，而在出现错误时返回负值。根据 SPPLE 的设备角色是服务器还是客户端，调用的 API 函数为 GATT_Handle_Value_Notification 或 GATT_Write_Without_Response_Request；这两个函数分别通知接收额度特性或向传输额度特性发送无响应的写入数据包。

参数

LESend 需要两个参数。第一个是您要发送到的设备的远程蓝牙地址。第二个是要发送的字节数。该值必须大于 10。

命令调用示例

- “LeSend 0017E7FEFD7C 100”尝试向 0017E7FEFD7C 发送 100 字节的数据。
- “LeSend B8FFFEAF1CAD 25”尝试向 B8FFFEAF1CAD 发送 25 字节的数据。

可能的返回值

- (0) 成功发送了数据
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-67) BTPS_ERROR_RFCOMM_NOT_INITIALIZED
- (-85) BTPS_ERROR_SPP_NOT_INITIALIZED

API 调用

GATT_Handle_Value_Notification(BluetoothStackID, SPPLEServiceID, ConnectionID, SPPLE_TX_CHARACTERISTIC_ATTRIBUTE_OFFSET, (Word_t)DataCount, SPPLEBuffer)

或者

GATT_Write_Without_Response_Request(BluetoothStackID, ConnectionID, DeviceInfo->ClientInfo.Rx_Characteristic, (Word_t)DataCount, SPPLEBuffer)

API 原型

*int BTPSAPI GATT_Handle_Value_Notification(unsigned int BluetoothStackID, unsigned int ServiceID, unsigned int ConnectionID, Word_t AttributeOffset, Word_tAttributeValueLength, Byte_t *AttributeValue)*

或者

*int BTPSAPI GATT_Write_Without_Response_Request(unsigned int BluetoothStackID, unsigned int ConnectionID, Word_t AttributeHandle, Word_t AttributeLength, void*AttributeValue)*

API 说明

第一个 API 函数用于向远程 GATT 客户端发送句柄/值通知。此函数的第一个参数是本地蓝牙协议栈的 ID。第二个参数是发送句柄/值通知的服务的 ID。第三个参数指定接收句柄/值通知的连接的 ID。第四个参数指定正在通知的属性在服务表（通过调用 GATT_Register_Service() 函数注册）中的偏移量。第五个参数是正在通知的属性值的长度（以字节为单位）。第六个参数是指向要通知的实际属性值的指针。此函数返回一个非负值，该值表示已通知的属性值的实际长度；如果出现错误，则返回负的错误代码。

提供第二个 API 函数是为了能够针对指定的属性向远程设备执行无响应的写入请求。该函数的第一个参数是本地蓝牙协议栈的 ID，依次后跟所连接远程设备的连接 ID、要写入的属性的句柄、要写入的值数据的长度（以字节为单位）以及要写入的实际值。此函数返回成功写入的字节数或负的错误代码。

11.4 配置 SPPLE

说明

以下函数负责在远程设备上配置 SPPLE 服务。此函数在成功执行时返回零，而在出现错误时返回负值。以下函数根据指定的句柄通知正确的特性；根据 SPPLE 的设备角色是服务器还是客户端，调用的 API 函数为 GATT_Handle_Value_Notification 或 GATT_Write_Without_Response_Request；这两个函数分别通知接收额度特性或向传输额度特性发送无响应的写入数据包。

参数

唯一需要的参数是所连接的远程设备的蓝牙地址。

命令调用示例

- “ConfigureSPPLE 001bdc05b617”尝试配置 BD_ADDR 为 001bdc05b617 的蓝牙设备的服务。
- “ConfigureSPPLE 000275e126FF”尝试配置 BD_ADDR 为 000275e126FF 的蓝牙设备的服务。

可能的返回值

- (0) 成功配置了 SPPLE 服务。
- (-4) 函数错误（失败时）。

API 调用

GATT_Write_Request(BluetoothStackID, ConnectionID, ClientConfigurationHandle, sizeof(Buffer), &Buffer, ClientEventCallback, 0)

和

GATT_Handle_Value_Notification(BluetoothStackID, SPPLEServiceID, ConnectionID, SPPLE_RX_CREDITS_CHARACTERISTIC_ATTRIBUTE_OFFSET, WORD_SIZE, (Byte_t)&Credits)*

或者

GATT_Write_Without_Response_Request(BluetoothStackID, ConnectionID, DeviceInfo->ClientInfo.Tx_Credit_Characteristic, WORD_SIZE, &Credits)

API 原型

*int BTPSAPI GATT_Write_Request(unsigned int BluetoothStackID, unsigned int ConnectionID, Word_t AttributeHandle, Word_t AttributeLength, void *AttributeValue,GATT_Client_Event_Callback_t ClientEventCallback, unsigned long CallbackParameter)*

和

*int BTPSAPI GATT_Handle_Value_Notification(unsigned int BluetoothStackID, unsigned int ServiceID, unsigned int ConnectionID, Word_t AttributeOffset, Word_tAttributeValueLength, Byte_t *AttributeValue)*

或者

*int BTPSAPI GATT_Write_Without_Response_Request(unsigned int BluetoothStackID, unsigned int ConnectionID, Word_t AttributeHandle, Word_t AttributeLength, void*AttributeValue)*

API 说明

提供的第一个 API 函数能够针对指定的属性向远程设备执行写入请求。该函数的第一个参数是本地蓝牙协议栈的 ID，依次后跟所连接远程设备的连接 ID、要写入其值的属性的句柄、值的长度（以字节为单位）以及要写入的实际值数据。最后两个参数分别指定 GATT 客户端事件回调函数和回调参数，当从远程设备接收响应时可以调用这些参数。此函数返回请求的正的非零事务 ID 或负的错误代码。

第二个 API 函数用于向远程 GATT 客户端发送句柄/值通知。此函数的第一个参数是本地蓝牙协议栈的 ID。第二个参数是发送句柄/值通知的服务的 ID。第三个参数指定接收句柄/值通知的连接的 ID。第四个参数指定正在通知的属性在服务表（通过调用 GATT_Register_Service() 函数注册）中的偏移量。第五个参数是正在通知的属性值的长度（以字节为单位）。第六个参数是指向要通知的实际属性值的指针。此函数返回一个非负值，该值表示已通知的属性值的实际长度；如果出现错误，则返回负的错误代码。

提供第三个 API 函数，以便针对指定的属性向远程设备执行无响应的写入请求。该函数的第一个参数是本地蓝牙协议栈的 ID，依次后跟所连接远程设备的连接 ID、要写入的属性的句柄、要写入的值数据的长度（以字节为单位）以及要写入的实际值。此函数返回成功写入的字节数或负错误代码。

11.5 LE 读取

说明

以下函数负责读取由存在连接的远程设备发送的数据。此函数在成功执行时返回零，而在出现错误时返回负值。根据 SPPLE 的设备角色是服务器还是客户端，调用的 API 函数为 GATT_Handle_Value_Notification 或 GATT_Write_Without_Response_Request；这两个函数分别通知接收额度特性或向传输额度特性发送无响应的写入数据包。

参数

唯一需要的参数是所连接的远程设备的蓝牙地址。

命令调用示例

- “LeRead 001bdc05b617”尝试读取 BD_ADDR 为 001bdc05b617 的蓝牙设备的数据。
- “LeRead 000275e126FF”尝试读取 BD_ADDR 为 000275e126FF 的蓝牙设备的数据。

可能的返回值

- (0) 成功读取了数据
- (-6) INVALID_PARAMETERS_ERROR
- (-1) BTPS_ERROR_INVALID_PARAMETER
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-67) BTPS_ERROR_RFCOMM_NOT_INITIALIZED
- (-85) BTPS_ERROR_SPP_NOT_INITIALIZED
- (-86) BTPS_ERROR_SPP_PORT_NOT_OPENED

API 调用

GATT_Handle_Value_Notification(BluetoothStackID, SPPLEServiceID, ConnectionID, SPPLE_RX_CREDITS_CHARACTERISTIC_ATTRIBUTE_OFFSET, WORD_SIZE, (Byte_t)&Credits)*

或者

GATT_Write_Without_Response_Request(BluetoothStackID, ConnectionID, DeviceInfo->ClientInfo.Tx_Credit_Characteristic, WORD_SIZE, &Credits)

API 原型

*int BTPSAPI GATT_Handle_Value_Notification(unsigned int BluetoothStackID, unsigned int ServiceID, unsigned int ConnectionID, Word_t AttributeOffset, Word_tAttributeValueLength, Byte_t *AttributeValue)*

或者

*int BTPSAPI GATT_Write_Without_Response_Request(unsigned int BluetoothStackID, unsigned int ConnectionID, Word_t AttributeHandle, Word_t AttributeLength, void*AttributeValue)*

API 说明

第一个 API 函数用于向远程 GATT 客户端发送句柄/值通知。此函数的第一个参数是本地蓝牙协议栈的 ID。第二个参数是发送句柄/值通知的服务的 ID。第三个参数指定接收句柄/值通知的连接的 ID。第四个参数指定正在通知的属性在服务表（通过调用 GATT_Register_Service() 函数注册）中的偏移量。第五个参数是正在通知的属性值的长度（以字节为单位）。第六个参数是指向要通知的实际属性值的指针。此函数返回一个非负值，该值表示已通知的属性值的实际长度；如果出现错误，则返回负的错误代码。

提供第二个 API 函数是为了能够针对指定的属性向远程设备执行无响应的写入请求。该函数的第一个参数是本地蓝牙协议栈的 ID，依次后跟所连接远程设备的连接 ID、要写入的属性的句柄、要写入的值数据的长度（以字节为单位）以及要写入的实际值。此函数返回成功写入的字节数或负的错误代码。

11.6 环回

说明

Loopback 命令负责设置应用程序状态以支持环回模式。该命令在成功执行时返回零，而对于错误返回负值。

参数

此命令需要一个参数来指示是否可以支持环回。0 = 环回处于非活动状态，1 = 环回处于活动状态。

命令调用示例

- “Loopback 0”将环回支持设置为非活动状态。
- “loopback 1”将环回支持设置为活动状态。

可能的返回值

- (0) 成功设置了环回支持。
- (-6) INVALID_PARAMETERS_ERROR.

11.7 显示原始模式数据

说明

以下函数负责设置应用程序状态，以支持显示原始数据。此函数在成功执行时返回零，而在出现错误时返回负值。

参数

该命令使用一个参数指示是否支持显示原始数据模式。0 = 显示原始数据模式处于非活动状态，1 = 显示原始数据模式处于活动状态。

命令调用示例

- “DisplayRawModeData 0”将显示原始模式支持设置为非活动状态。

- “DisplayRawModeData 1”将显示原始模式支持设置为活动状态。

可能的返回值
- (0) 成功设置了显示原始数据模式支持。
- (-6) INVALID_PARAMETERS_ERROR。

11.8 自动读取模式

说明

AutomaticReadMode 命令负责设置应用程序状态，以支持自动读取通过 SPP 接收的所有数据。此函数在成功执行时返回零，而在出现错误时返回负值。

参数

此命令使用一个参数指示是否支持自动读取模式。0 = 自动读取模式处于非活动状态，1 = 自动读取模式处于活动状态。

命令调用示例
- “AutomaticReadMode 0”将自动读取模式支持设置为非活动状态。
- “AutomaticReadMode 1”将自动读取模式支持设置为活动状态。

可能的返回值
- (0) 成功设置了自动读取模式支持。
- (-6) INVALID_PARAMETERS_ERROR

12 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。

13 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