



摘要

免提配置文件允许用户在嵌入式设备上演示免提配置文件的用法。免提配置文件用于将耳机或扬声器与移动设备连接，以提供远程控制和语音连接。免提配置文件支持两种角色：免提和音频网关。本文档将演示如何使用免提配置文件的音频网关角色。在尝试本页中介绍的应用程序之前，请访问 [MSP432 套件设置入门指南](#) 或 [STM32F4 套件设置入门指南](#) 页面。

备注

可以使用相同的指令在 MSP432 或 STM32F4 平台上运行此演示。

备注

必须将外部编解码器连接到 CC256x I2S/PCM 接口才能播放和录制音频。

内容

1 演示应用程序	3
2 运行蓝牙代码	3
3 使用 HFP 演示应用程序进行服务器设置	4
4 演示应用程序上的客户端设置	5
5 示例：带商用耳机的音频网关	10
6 应用程序命令	12
6.1 通用访问配置文件命令.....	12
6.2 帮助 (DisplayHelp).....	12
6.3 查询.....	12
6.4 配对.....	13
6.5 结束配对.....	14
6.6 PIN 码响应.....	15
6.7 通行密钥响应.....	15
6.8 用户确认响应.....	16
6.9 设置可发现性模式.....	17
6.10 设置可连接性模式.....	18
6.11 设置可配对性模式.....	18
6.12 更改简易配对参数.....	19
6.13 获取本地地址.....	20
6.14 设置本地名称.....	20
6.15 获取本地名称.....	21
6.16 设置设备类别.....	22
6.17 获取设备类别.....	22
6.18 获取远程名称.....	23
7 免提配置文件命令	25
7.1 服务发现.....	25
7.2 打开音频网关客户端.....	25
7.3 管理音频.....	26
7.4 更新控制指示器.....	27
7.5 呼叫等待.....	28
7.6 设置语音识别激活.....	29
7.7 设置扬声器增益.....	30

7.8 设置麦克风增益.....	31
7.9 禁用远程声音增强.....	32
7.10 发送来电显示通知.....	33
7.11 设置振铃指示.....	33
7.12 振铃指示.....	34
7.13 发送来电状态.....	35
7.14 关闭 AG 客户端.....	36
7.15 发送运营商信息.....	37
7.16 发送用户号码.....	38
7.17 发送呼叫列表.....	39
8 参考文献.....	40
9 修订历史记录.....	40

商标

Bluetooth® is a registered trademark of Bluetooth.

所有商标均为其各自所有者的财产。

1 演示应用程序

本节介绍如何使用演示应用程序连接两块已配置的电路板并通过 Bluetooth® 进行通信。蓝牙 HFP 是一种简单的客户端-服务器连接过程，其中一侧以音频网关角色运行，另一侧以免提角色运行。需将其中一块电路板设置为音频网关客户端 (HFPAGDemo 充当音频网关)，并将另一块电路板设置为免提服务器 (使用 HFP 演示)。这样将通过蓝牙在两个设备之间启动连接并发送指示。现在使用第二块板，按照 HFP 演示 (使用 Tiva 或 MSP430 作为服务器时) 或 HFPDemo_HF (使用 MSP432 或 STM32F4 作为服务器时) 中的步骤进行操作。请在第一块电路板上运行音频网关蓝牙代码之前执行这些步骤。连接到计算机的第二个设备将是免提服务器。

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。对于 MSP432，此驱动程序在设备管理器的“Ports (COM & LPT)”下显示为“XDS110 ClassApplication/User 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、8、n、1。连接后，使用“Reset S3”按钮将设备复位，并在终端程序上观察初始化的堆栈，此时会显示帮助屏幕，其中显示所有命令。此设备现在成为音频网关。

3 使用 HFP 演示应用程序进行服务器设置

使用 HFP 演示应用程序进行服务器设置

1. 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。
2. 通过发出 SetLocalName 来为平台命名。在此示例中，名称为 hfpserver。
3. 通过发出命令“OpenHfServer”打开 HfPServer。使用 OpenHfServer 打开端口 1（这是默认的第一个端口）。

```

COM44:115200baud - Tera Term VT
File Edit Setup Control Window Help

OpenStack().
Bluetooth Stack ID: 1.
WBS Support initialized.
Device Chipset Version: Unknown (greater 4.0)
Bluetooth Device Address: 0x0017e9505f95
GAP_Set_Connectability_Mode(cmConnectable).
GAP_Set_Discoverability_Mode(dmGeneralDiscoverable, 0).
GAP_Set_Pairability_Mode(pmPairableMode).
GAP_Register_Remote_Authentication() Success.

*****
* Command Options: Inquiry, DisplayInquiryList, Pair,
* EndPairing, PINCodeResponse, PassKeyResponse,
* UserConfirmationResponse,
* SetDiscoverabilityMode, SetConnectabilityMode,
* SetPairabilityMode,
* ChangeSimplePairingParameters,
* GetLocalAddress, GetLocalName, SetLocalName,
* GetClassOfDevice, SetClassOfDevice,
* GetRemoteName, OpenHfServer, CloseHfServer
* ManageAudio, AnswerCall, HangUpCall, Close,
* Help
*****

HFRE16>SetLocalName hfpserver
Local Device Name set to: hfpserver.

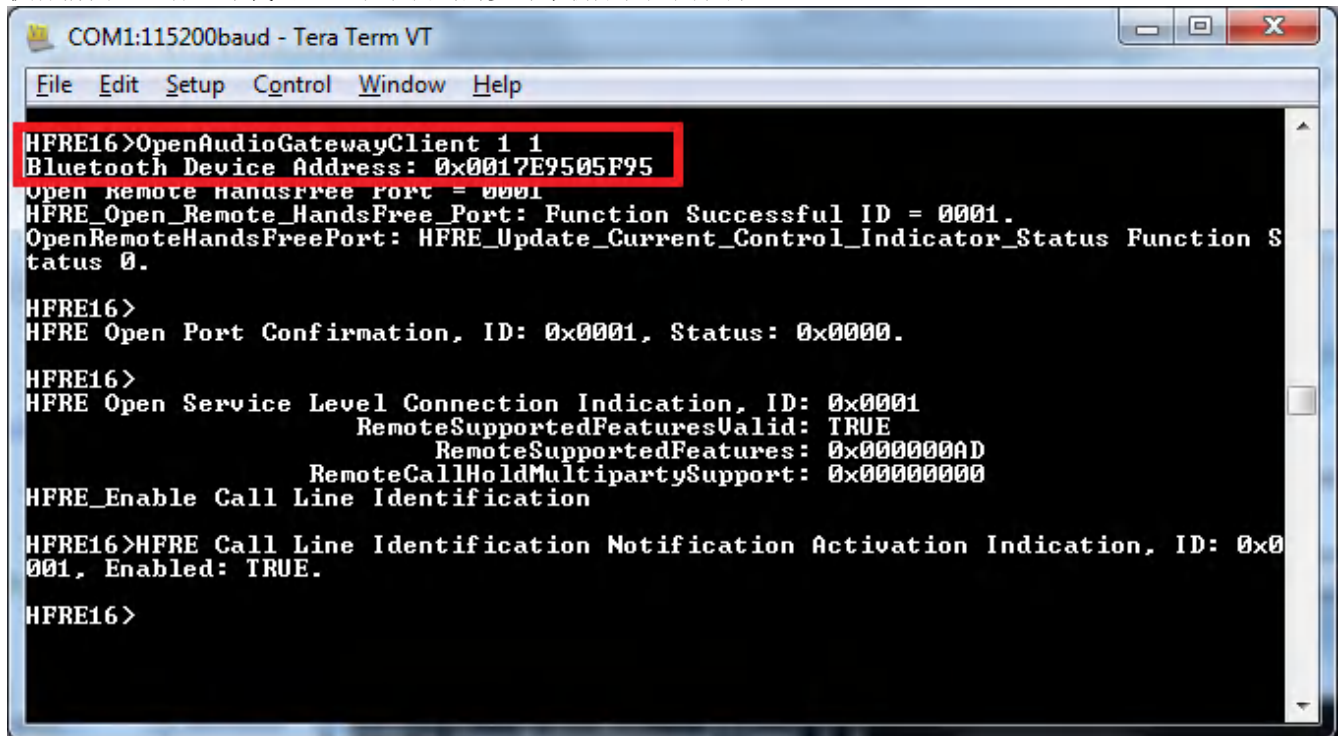
HFRE16>OpenHfServer 1
HFRE_Open_HandsFree_Server_Port: Function Successful.
HFRE_Register_HandsFree_SDP_Record: Function Successful.

HFRE16>
  
```

图 3-1. 使用 HFP 演示应用程序进行服务器设置

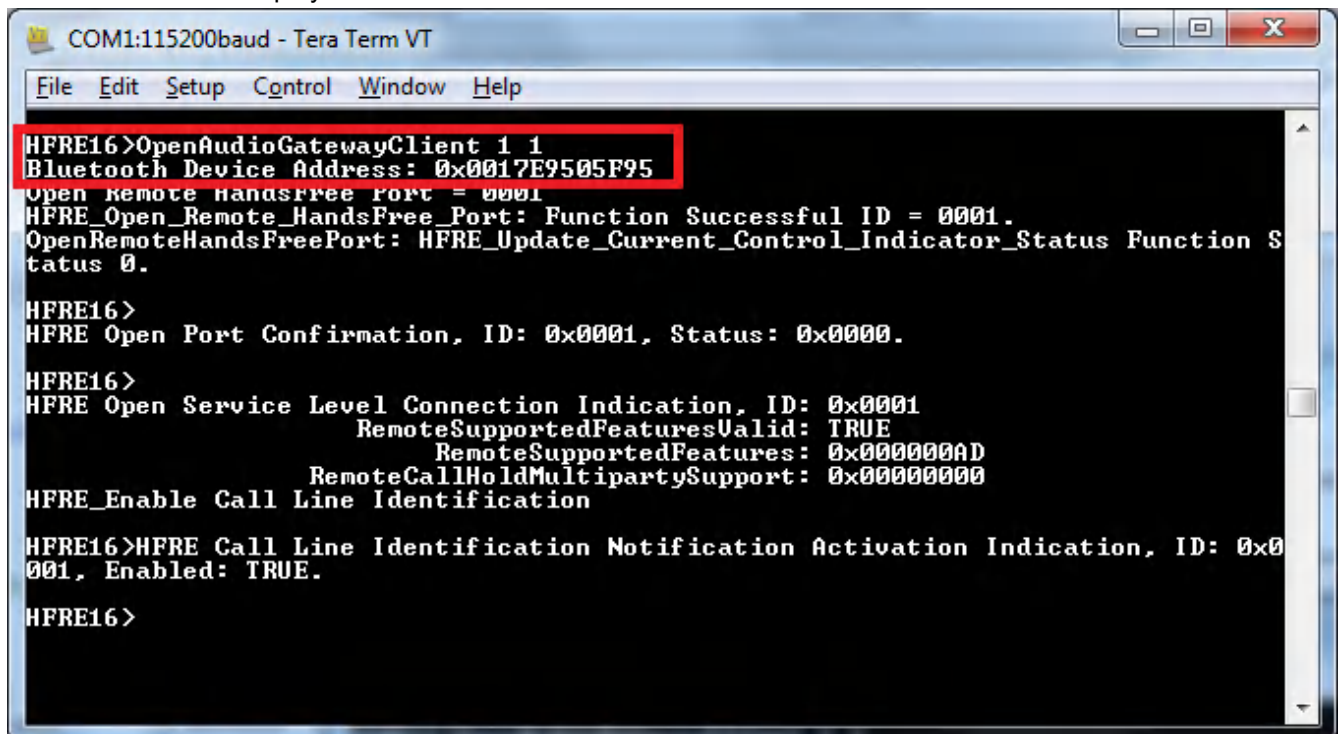
4 演示应用程序上的客户端设置

1. 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。



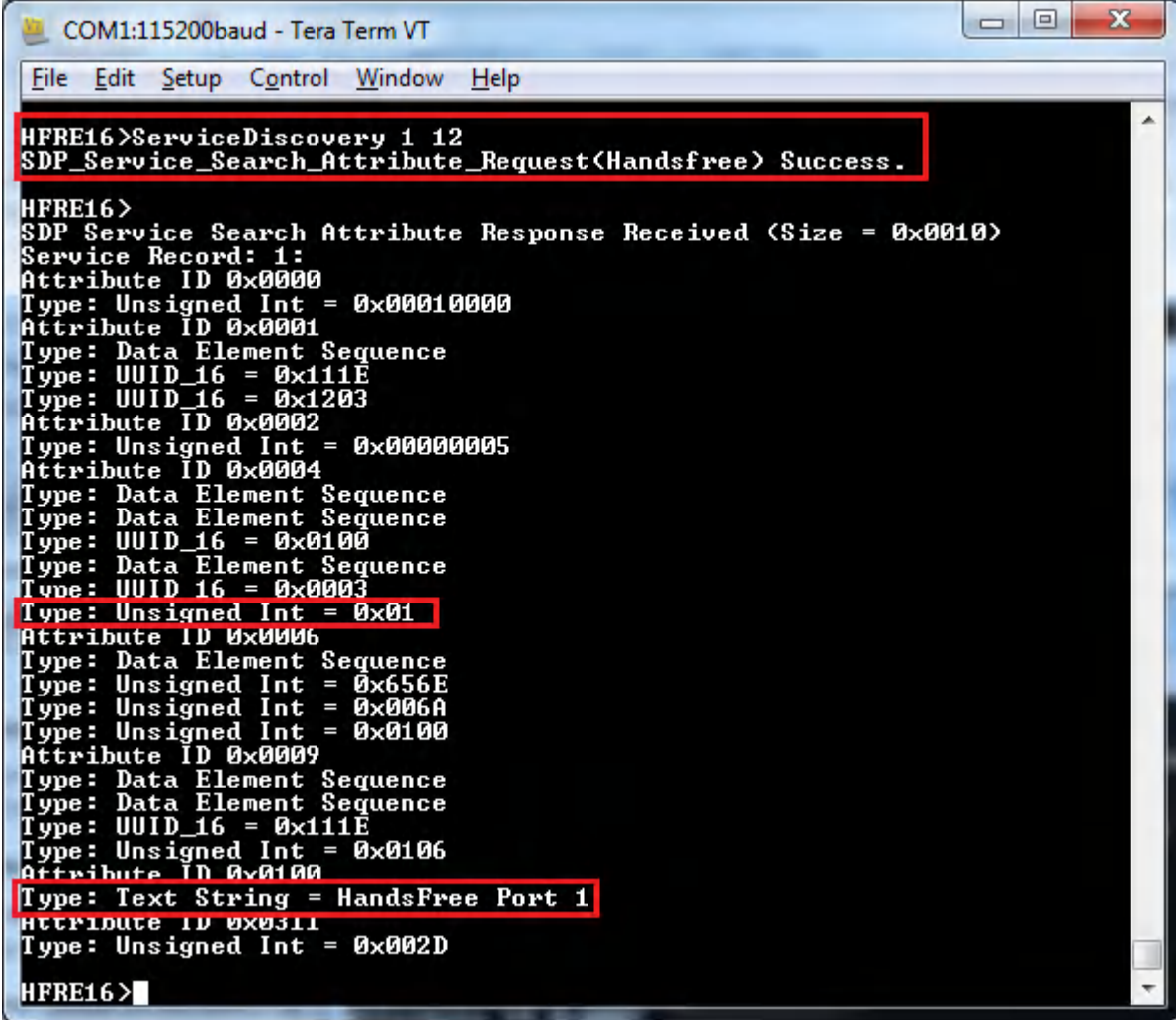
```
COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>OpenAudioGatewayClient 1 1
Bluetooth Device Address: 0x0017E9505F95
Open Remote Handsfree Port = 0001
HFRE_Open_Remote_HandsFree_Port: Function Successful ID = 0001.
OpenRemoteHandsFreePort: HFRE_Update_Current_Control_Indicator_Status Function S
tatus 0.
HFRE16>
HFRE Open Port Confirmation, ID: 0x0001, Status: 0x0000.
HFRE16>
HFRE Open Service Level Connection Indication, ID: 0x0001
RemoteSupportedFeaturesValid: TRUE
RemoteSupportedFeatures: 0x000000AD
RemoteCallHoldMultipartySupport: 0x00000000
HFRE_Enable Call Line Identification
HFRE16>HFRE Call Line Identification Notification Activation Indication, ID: 0x0
001, Enabled: TRUE.
HFRE16>
```

2. 对 HFP 服务器发出 Inquiry 命令。



```
COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>OpenAudioGatewayClient 1 1
Bluetooth Device Address: 0x0017E9505F95
Open Remote Handsfree Port = 0001
HFRE_Open_Remote_HandsFree_Port: Function Successful ID = 0001.
OpenRemoteHandsFreePort: HFRE_Update_Current_Control_Indicator_Status Function S
tatus 0.
HFRE16>
HFRE Open Port Confirmation, ID: 0x0001, Status: 0x0000.
HFRE16>
HFRE Open Service Level Connection Indication, ID: 0x0001
RemoteSupportedFeaturesValid: TRUE
RemoteSupportedFeatures: 0x000000AD
RemoteCallHoldMultipartySupport: 0x00000000
HFRE_Enable Call Line Identification
HFRE16>HFRE Call Line Identification Notification Activation Indication, ID: 0x0
001, Enabled: TRUE.
HFRE16>
```

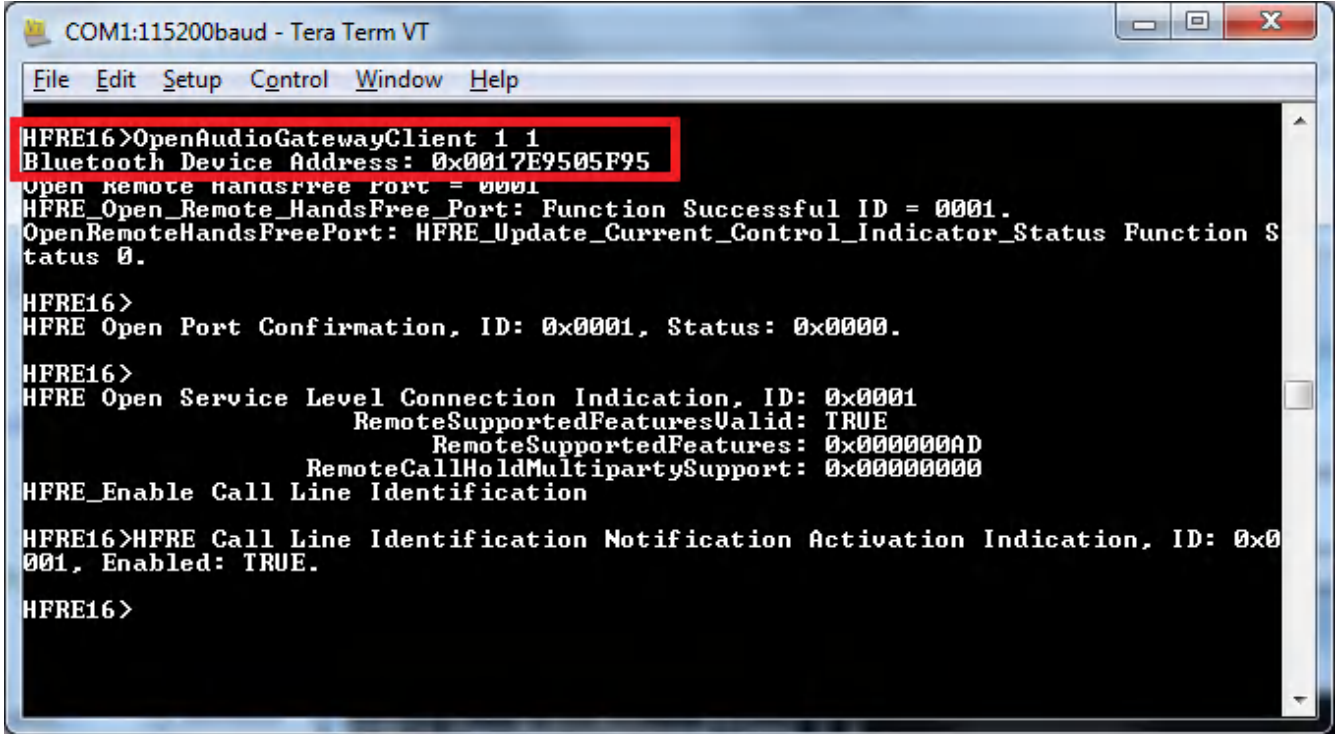
3. 通过发出 ServiceDiscovery 1 12 命令来获取端口号以发现远程 HFP 服务器的服务。



```

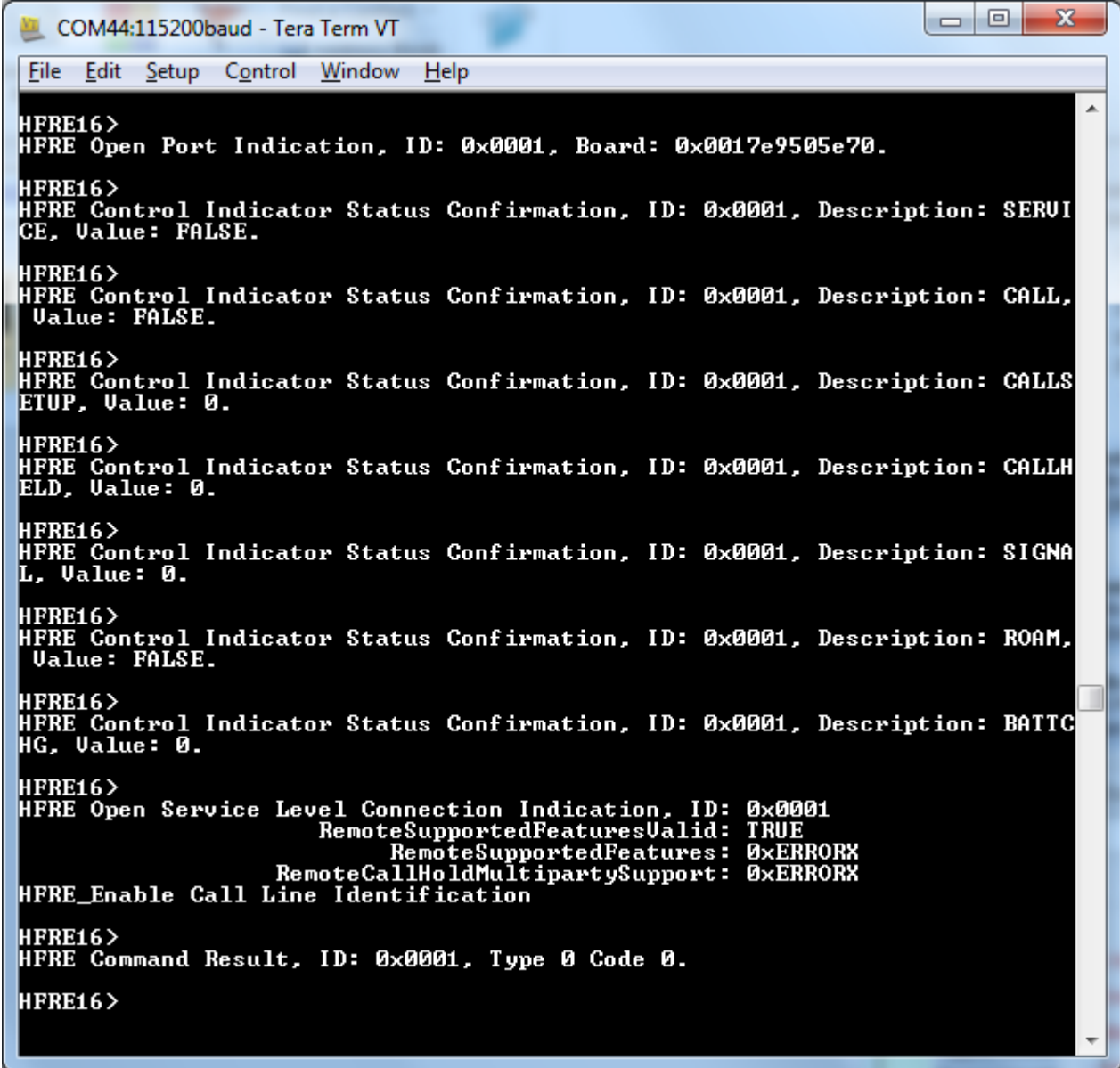
COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>ServiceDiscovery 1 12
SDP_Service_Search_Attribute_Request<Handsfree> Success.
HFRE16>
SDP Service Search Attribute Response Received (Size = 0x0010)
Service Record: 1:
Attribute ID 0x0000
Type: Unsigned Int = 0x00010000
Attribute ID 0x0001
Type: Data Element Sequence
Type: UUID_16 = 0x111E
Type: UUID_16 = 0x1203
Attribute ID 0x0002
Type: Unsigned Int = 0x00000005
Attribute ID 0x0004
Type: Data Element Sequence
Type: Data Element Sequence
Type: UUID_16 = 0x0100
Type: Data Element Sequence
Type: UUID_16 = 0x0003
Type: Unsigned Int = 0x01
Attribute ID 0x0006
Type: Data Element Sequence
Type: Unsigned Int = 0x656E
Type: Unsigned Int = 0x006A
Type: Unsigned Int = 0x0100
Attribute ID 0x0009
Type: Data Element Sequence
Type: Data Element Sequence
Type: UUID_16 = 0x111E
Type: Unsigned Int = 0x0106
Attribute ID 0x0010
Type: Text String = HandsFree Port 1
Attribute ID 0x0011
Type: Unsigned Int = 0x002D
HFRE16>
    
```

4. 通过发出 OpenAudioGatewayClient 1 1 命令启动与远程 HFP 服务器的连接。



```
COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>OpenAudioGatewayClient 1 1
Bluetooth Device Address: 0x0017E9505F95
Open Remote Handsfree port = 0001
HFRE_Open_Remote_HandsFree_Port: Function Successful ID = 0001.
OpenRemoteHandsFreePort: HFRE_Update_Current_Control_Indicator_Status Function S
tatus 0.
HFRE16>
HFRE Open Port Confirmation, ID: 0x0001, Status: 0x0000.
HFRE16>
HFRE Open Service Level Connection Indication, ID: 0x0001
RemoteSupportedFeaturesValid: TRUE
RemoteSupportedFeatures: 0x000000AD
RemoteCallHoldMultipartySupport: 0x00000000
HFRE_Enable Call Line Identification
HFRE16>HFRE Call Line Identification Notification Activation Indication, ID: 0x0
001, Enabled: TRUE.
HFRE16>
```

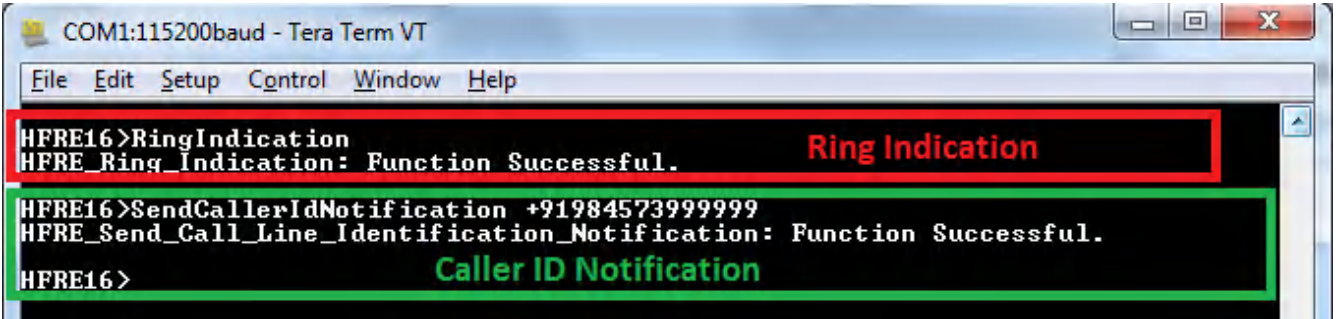
观察下面来自 HFP 服务器的输出。



```

COM44:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>
HFRE Open Port Indication, ID: 0x0001, Board: 0x0017e9505e70.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: SERVICE, Value: FALSE.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: CALL, Value: FALSE.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: CALLSETUP, Value: 0.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: CALLHOLD, Value: 0.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: SIGNAL, Value: 0.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: ROAM, Value: FALSE.
HFRE16>
HFRE Control Indicator Status Confirmation, ID: 0x0001, Description: BATTCHG, Value: 0.
HFRE16>
HFRE Open Service Level Connection Indication, ID: 0x0001
RemoteSupportedFeaturesValid: TRUE
RemoteSupportedFeatures: 0xERRORX
RemoteCallHoldMultipartySupport: 0xERRORX
HFRE_Enable Call Line Identification
HFRE16>
HFRE Command Result, ID: 0x0001, Type 0 Code 0.
HFRE16>
    
```

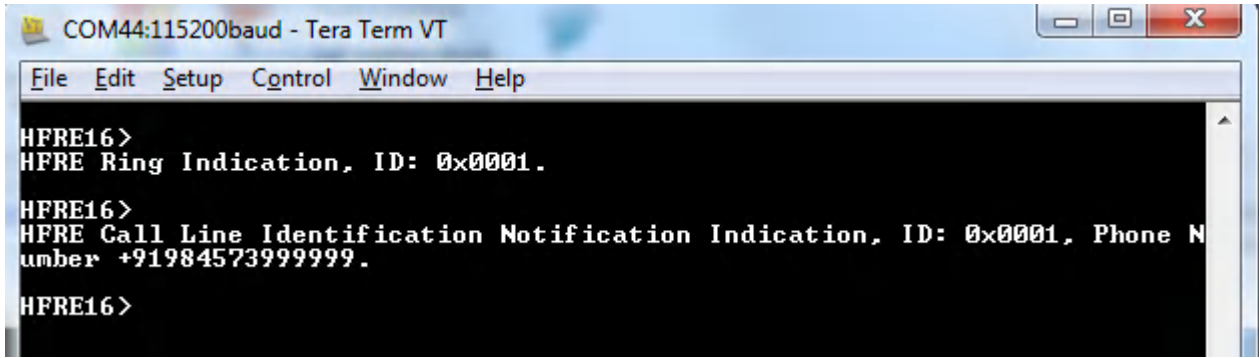
5. 发送指示：发出 RingIndication 或 SendCallerIdNotifications +9198787899889 命令。



```

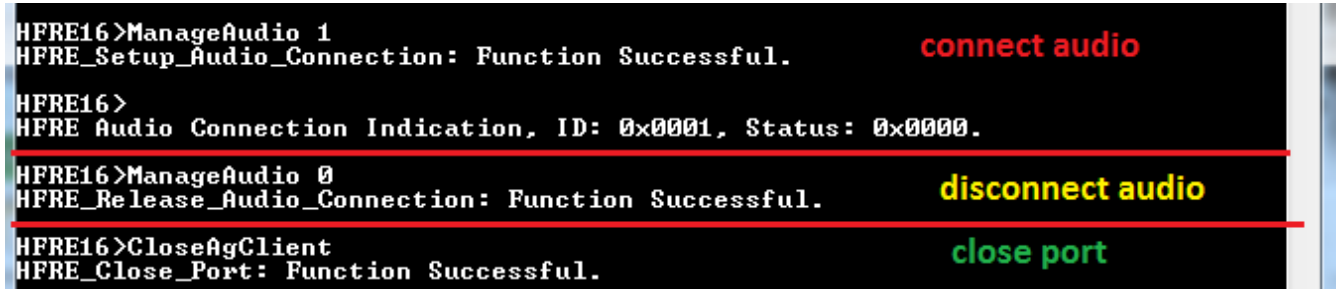
COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>RingIndication
HFRE_Ring_Indication: Function Successful. Ring Indication
HFRE16>SendCallerIdNotification +919845739999999
HFRE_Send_Call_Line_Identification_Notification: Function Successful.
HFRE16> Caller ID Notification
    
```

观察下面来自 HFP 服务器的输出。



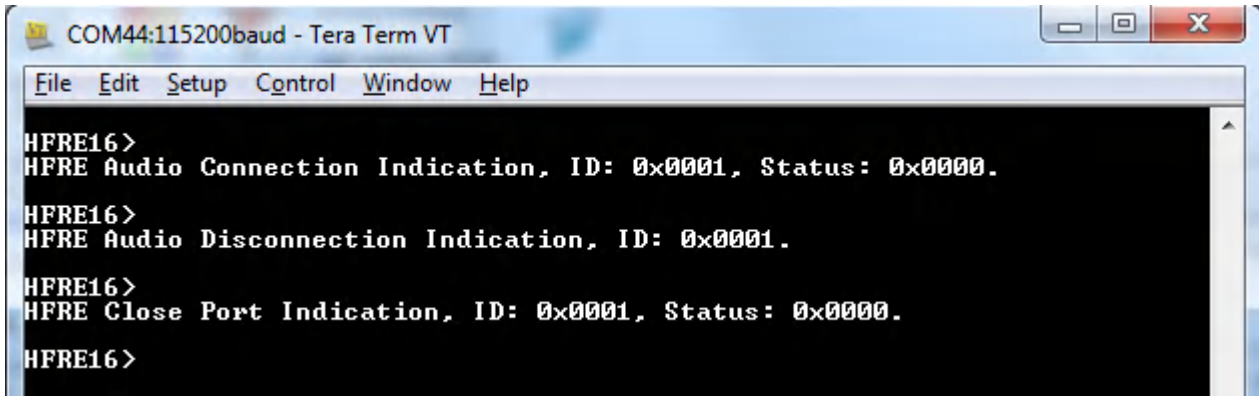
```
COM44:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>
HFRE Ring Indication, ID: 0x0001.
HFRE16>
HFRE Call Line Identification Notification Indication, ID: 0x0001, Phone Number +91984573999999.
HFRE16>
```

6. 通过发出 `ManageAudio <STATE>` 和 `CloseAgClient` 命令启动音频连接/断开连接并关闭 HFP 连接。



```
HFRE16>ManageAudio 1
HFRE_Setup_Audio_Connection: Function Successful. connect audio
HFRE16>
HFRE Audio Connection Indication, ID: 0x0001, Status: 0x0000.
HFRE16>ManageAudio 0
HFRE_Release_Audio_Connection: Function Successful. disconnect audio
HFRE16>CloseAgClient
HFRE_Close_Port: Function Successful. close port
```

观察下面来自 HFP 服务器的输出。



```
COM44:115200baud - Tera Term VT
File Edit Setup Control Window Help
HFRE16>
HFRE Audio Connection Indication, ID: 0x0001, Status: 0x0000.
HFRE16>
HFRE Audio Disconnection Indication, ID: 0x0001.
HFRE16>
HFRE Close Port Indication, ID: 0x0001, Status: 0x0000.
HFRE16>
```

备注

远程免提设备上的端口 ID 为 0x01 (无符号整数)，如下图突出所示 (属性 ID 0x0004)。此端口 ID 在以下 `OpenAudioGatewayClient` 命令中用作第二个参数。

5 示例：带商用耳机的音频网关

下面的示例演示了如何设置客户端以连接到商用耳机。

1. 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。

```
OpenStack().Bluetooth Stack ID: 1WBS support enabled.Device Chipset: Unknown (greater 4.1)BTPS
Version : 4.2.1.0Project Type : 6FW Version : 12.12App Name : HFPDemo_AGApp Version : 0.3LOCAL
BD_ADDR:
0x88C255D1D645GAP_Set_Connectability_Mode(cmConnectable).GAP_Set_Discoverability_Mode(dmGeneralD
iscoverable, 0).GAP_Set_Pairability_Mode(pmPairableMode).GAP_Register_Remote_Authentication()
Success.Command Options: Inquiry, DisplayInquiryList, Pair, EndPairing, PINCodeResponse,
PassKeyResponse, UserConfirmationResponse, SetDiscoverabilityMode, SetConnectabilityMode,
SetPairabilityMode, ChangeSimplePairingParameters, GetLocalAddress, SetLocalName,
GetLocalName, SetClassOfDevice, GetClassOfDevice, GetRemoteName, ManageAudio,
OpenAudioGatewayClient, CloseAgClient, ServiceDiscovery, UpdateControlIndicators,
CallWaiting, SendCallerIdNotification, SetRingIndication, RingIndication,
SetVoiceRecognitionActivation, SetSpeakerGain, SetMicroPhoneGain,
DisableRemoteSoundEnhancement, SendOperatorInfo, SendIncomingCallState, SendSubNumber,
SendCallList, Help
```

2. 对 HFP 服务器发出 Inquiry 命令。

```
HFP AG>InquiryReturn Value is 0 GAP_Perform_Inquiry() SUCCESS.HFP AG>GAP Inquiry Entry Result:
0x244B03F712D3.HFP AG>GAP Inquiry Entry Result: 0x3402862C9AE9.HFP AG>GAP Inquiry Entry Result:
0x340286605044.HFP AG>GAP Inquiry Entry Result: 0x484520902A4E.HFP AG>GAP Inquiry Entry Result:
0x08DF1F99F8D0.HFP AG>GAP Inquiry Entry Result: 0x002500F84FAB.HFP AG>GAP_Inquiry_Result: 6
Found.GAP Inquiry Result: 1, 0x244B03F712D3.GAP Inquiry Result: 2, 0x3402862C9AE9.GAP Inquiry
Result: 3, 0x340286605044.GAP Inquiry Result: 4, 0x484520902A4E.GAP Inquiry Result: 5,
0x08DF1F99F8D0.GAP Inquiry Result: 6, 0x002500F84FAB.
```

3. 通过发出 ServiceDiscovery 5 12 命令来获取端口号以发现远程 HFP 服务器的服务。

```
HFP AG>ServiceDiscoveryUsage: SERVICEDISCOVERY [Inquiry Index] [Profile Index] [16/32 bit UUID
(Manual only)]. Profile Index: 0) Manual (MUST specify 16/32 bit UUID) 1) L2CAP 2) Advanced
Audio 3) A/V Remote Control 4) Basic Imaging 5) Basic Printing 6) Dial-up Networking 7) FAX 8)
File Transfer 9) Hard copy cable repl. 10) Health Device 11) Headset 12) Handsfree 13) HID 14)
LAN Access
15) Message Access 16) Object Push 17) Personal Area Network 18) Phonebook Access 19) SIM
Access 20) Serial Port 21) IrSYNCFUNCTION Error.HFP AG>ServiceDiscovery 5
12SDP_Service_Search_Attribute_Request(Handsfree) Success.HFP AG>SDP Service Search Attribute
Response Received (Size = 0x0010)Service Record: 1:Attribute ID 0x0000Type: Unsigned Int =
0x00010000Attribute ID 0x0001Type: Data Element SequenceType: UUID_16 = 0x111EType: UUID_16 =
0x1203Attribute ID 0x0004Type: Data Element SequenceType: Data Element SequenceType: UUID_16 =
0x0100Type: Data Element SequenceType: UUID_16 = 0x0003Type: Unsigned Int = 0x0AAttribute ID
0x0006Type: Data Element SequenceType: Unsigned Int = 0x656EType: Unsigned Int = 0x006AType:
Unsigned Int = 0x0100Attribute ID 0x0009Type: Data Element SequenceType: Data Element
SequenceType: UUID_16 = 0x111EType: Unsigned Int = 0x0106Attribute ID 0x0100Type: Text String =
Hands-Free unitAttribute ID 0x0311Type: Unsigned Int = 0x003F
```

4. 通过发出 OpenAudioGatewayClient 5 10 命令启动与远程 HFP 服务器的连接。

```
HFP AG>OpenAudioGatewayClientUsage: OPENAUDIOWAYCLIENT [Inquiry Index] [Port
Number].Function Error.HFP AG>OpenAudioGatewayClient 5 10Bluetooth Device Address:
0x08DF1F99F8D0Open Remote HandsFree Port = 000AHFRE_Open_Remote_HandsFree_Port: Function
Successful ID = 0001.OpenRemoteHandsFreePort: HFRE_Update_Current_Control_Indicator_Status
Function Status 0.
```

观察下面来自 HFP 服务器的输出。

```
HFP AG>atLinkKeyRequest: 0x08DF1F99F8D0GAP_Authentication_Response() Success.HFP
AG>atPINCodeRequest: 0x08DF1F99F8D0Respond with the command: PINCodeResponseHFP
AG>PINCodeResponse 0000PINCodeResponse.GAP_Authentication_Response(), Pin Code Response
Success.HFP AG>atLinkKeyCreation: 0x08DF1F99F8D0Link Key:
0x4AF49AD7072771919BAC62840F1F985DLink Key Stored locally.HFP AG>HFRE Open Port Confirmation,
ID: 0x0001, Status: 0x0000.HFP AG>HFRE Available Codec List Indication, ID: 0x0001 NumCodecs:
2 [ 1 2 ]HFP AG>HFRE Open Service Level Connection Indication, ID: 0x0001
RemoteSupportedFeaturesValid: TRUE RemoteSupportedFeatures: 0x000000BF
RemoteCallHoldMultipartySupport: 0x00000000HFRE_Enable Call Line IdentificationHFP AG>HFRE
Disable Sound_Enhancement Indication, ID: 0x0001HFRE_Send_Terminating_Response (erOK) :: Res =
0HFP AG>HFRE Speaker Gain Indication, ID: 0x0001, Speaker Gain 0x000A.HFP AG>HFRE Call Line
Identification Notification Activation Indication, ID: 0x0001, Enabled: TRUE.HFP AG>HFRE Call
Waiting Notification Activation Indication, ID: 0x0001, Enabled: TRUE.HFP AG> HFRE Response
Hold Status Indication, ID: 0x0001HFRE_Send_Incoming_Call_State (csNone) :: Res = 0HFP
AG>HFRE_Current_Calls_List_IndicationHFRE_Send_Terminating_Response (erOK) :: Res = 0
```

5. 通过发出 `ManageAudio <STATE>` 命令来启动音频连接，此命令默认选择改良型子带编码 (MSBC)。

```
HFP AG>ManageAudioUsage: Audio [Release = 0, set-up = 1].Function Error.HFP AG>ManageAudio
1HFRE_Send_Select_Codec:: Codec = 2, Res = 0
```

观察下面来自 HFP 服务器的输出。

```
HFP AG>HFRE Codec Select Confirmation, ID: 0x0001 AcceptedCodec=2set-up WBS with Audio for ACL
handle 0x0001HFRE_set-up_Audio_Connection: Function Successful.HFRE_set-up_Audio_Connection::
Res = 0HFP AG>HFRE Audio Connection Indication, ID: 0x0001, BDADDR=0x08DF1F99F8D0, Status:
0x0000.
```

备注

远程免提设备上的端口 ID 为 `0xA` (无符号整数)，来自属性 ID `0x0004`。此端口 ID 转换为十进制数 (10) 后将在以下 `OpenAudioGatewayClient` 命令中用作第二个参数。

6 应用程序命令

TI 的蓝牙协议栈是蓝牙协议栈上层的一种实现。TI 的 **Bluetooth Stack** 是一款强大而灵活的开发工具，可在主机控制器接口 (HCI) 之上实现蓝牙协议和配置文件。TI 蓝牙协议栈的应用程序编程接口 (API) 可用于访问上层协议和配置文件，并且可以直接与蓝牙控制器芯片连接。

MSP432 和 **STM32F4** 附带的基本蓝牙应用程序是一个免提音频网关应用程序。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。

6.1 通用访问配置文件命令

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。

6.2 帮助 (DisplayHelp)

说明

`DisplayHelp` 命令显示“Command Options”菜单。根据设备 (服务器或客户端) 的 `UI_MODE`，在某些情况下可以使用不同的命令。打开和关闭命令会根据设备所处的模式来改变其用途。

参数

使用此命令时不需要参数。参数对帮助菜单的结果没有影响。

可能的返回值

返回值始终为 0。

6.3 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 int MaximumPeriodLength, 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。在未完成的查询正在进行时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。如果要使用简单查询过程（非定期），这两个值都必须设置为零。如果指定了这两个参数，则必须满足以下条件：`MaximumPeriodLength > MinimumPeriodLength > InquiryLength`。

6.4 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）。还需要注意的是，要连接到远程设备，必须在调用 `Pair` 命令之前使用 `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，则无法与该设备配对。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” 尝试执行与上述示例完全相同的操作。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 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：执行绑定的本地蓝牙设备的蓝牙协议栈 ID、要绑定的设备的远程蓝牙地址、要执行的绑定类型，

以及用于处理此函数成功时将产生的身份验证事件的 **GAP** 事件回调信息。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。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。仅当与指定远程蓝牙设备的物理连接尚不存在时，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，则会维持链路，并且在调用 **GAP_End_Bonding** 函数之前不会终止链路。这将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。**GAP_Cancel_Bonding** 函数可随时由 **Pair** 命令调用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** 函数将不起作用（在这种情况下，**GAP_End_Bonding** 函数将返回错误代码）。

6.5 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。还需要注意的是，调用 **EndPairing** 之前必须使用 **Pair** 命令和 **Inquiry** 命令才能断开与远程设备的连接。

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_t GAP_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，并且不会向原始 **GAP_Initiate_Bonding** 函数调用中指定的 **GAP** 事件回调发出任何 **GAP** 事件回调（如果此函数返回成功结果）。

6.6 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）。

6.7 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
- (-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）。

6.8 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）。

6.9 设置可发现性模式

设置可发现性模式

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。

6.10 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。

6.11 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。在此之后可以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。

6.12 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

6.13 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）。

6.14 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（这意味着名称中不应有空格，否则只会设置该名称的第一部分）。

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）。如果成功更改本地设备名称，此函数将返回零；如果出现错误情况，则返回负的错误代码。

6.15 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 参数至少应为 (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）。如果成功，则此函数将返回零，并且 NameBuffer 指向的缓冲区将填充本地设备名称的 ASCII 表示形式（以 NULL 为终止符）。如果此函数返回负值，则无法查询本地设备名称（错误情况）。

6.16 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。该值最好是以“0x”开头，然后使用六位数值。如果不这样做，写入的设备类别将假定为十进制，然后将转换为十六进制格式并更改给定的值。

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。

6.17 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, 并且 **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。

6.18 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。此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 long CallbackParameter);

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，如果取消回调并且操作尝试取消，则会取消回调（即，**GAP** 模块仍执行远程名称请求，但始终不发出回调）。

7 免提配置文件命令

7.1 服务发现

说明

以下函数负责向远程 SDP 服务器发出服务搜索属性请求。如果成功，此函数将返回零；如果发生错误，则返回负值。

参数

此命令需要两个参数。第一个参数是查询索引，第二个参数是配置文件索引。

命令调用示例

- “ServiceDiscovery 1 12” 尝试根据查询索引 1 和配置文件索引 12 (免提) 打开音频网关客户端端口。
- “ServiceDiscovery 3 12” 尝试根据查询索引 3 和配置文件索引 12 (免提) 打开音频网关客户端端口。

可能的返回值

- (-1) BTPS_ERROR_INVALID_PARAMETER (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-40) BTPS_ERROR_MEMORY_ALLOCATION_ERROR (-54) BTPS_ERROR_EXPECTED_UUID_ENTRY (-64) BTPS_ERROR_INTERNAL_ERROR (-103)
- BTPS_ERROR_FEATURE_NOT_AVAILABLE

API 调用

SDP_Service_Search_Attribute_Request(BluetoothStackID, InquiryResultList[(TempParam->Params[0].intParam - 1)], 1, &SDPUUIDEntry, 1, &AttributeID, SDP_Event_Callback,(unsigned long)0)

API 原型

BTPSAPI_DECLARATION int BTPSAPI SDP_Service_Search_Attribute_Request(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, unsigned int NumberServiceUUID,SDP_UUID_Entry_t SDP_UUID_Entry[], unsigned int NumberAttributeListElements, SDP_Attribute_ID_List_Entry_t AttributeIDList[], SDP_Response_Callback_tSDP_Response_Callback, unsigned long CallbackParameter)

API 说明

以下函数负责向指定的 BD_ADDR 发出 SDP 服务搜索属性请求。此函数返回在调用此函数时指定的 SDP 响应回调中的服务搜索属性请求的结果。该函数将以下内容作为输入：SDP 客户端所在的蓝牙协议栈的 ID、要远程连接到的蓝牙板地址 (远程 SDP 服务器位于此 BD_ADDR 上)、要搜索的服务 UUID 数量、要实际搜索的服务 UUID、要查询的属性列表中的条目数、要在查询中实际使用的属性列表、SDP 响应回调函数，以及 SDP 响应回调函数回调参数。如果成功，此函数将返回一个正的非零代码；如果出现错误，则返回一个负的错误代码。如果该函数成功，用户可以调用 SDP_Cancel_Service_Request() 函数以提前取消 SDP 服务搜索请求。UUID 数量参数必须至少为 1，并且服务 UUID 参数必须指向至少包含已指定的 UUID 数量的列表。最后，BD_ADDR 参数和 SDP_Reponse_Callback 参数必须有效，否则对此函数的调用将失败。属性列表中的属性数量必须至少为 1，并且属性 ID 列表参数必须指向至少包含已指定的属性列表条目数量的属性 ID 列表。最后，BD_ADDR 参数和 SDP_Reponse_Callback 参数必须有效，否则对此函数的调用将失败。

7.2 打开音频网关客户端

说明

以下函数负责打开音频网关客户端端口。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

此命令需要两个参数。第一个参数是查询索引，第二个参数是远程设备端口号。

命令调用示例

- “OpenAudioGatewayClient 1 1” 尝试根据查询索引 1 和远程端口号 1 打开音频网关客户端端口。
- “OpenAudioGatewayClient 2 3” 尝试根据查询索引 2 和远程端口号 3 打开音频网关客户端端口。

可能的返回值

- (0) 成功打开 AG 客户端
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Open_Remote_Hands-free_Port(BluetoothStackID, InquiryResultList[(TempParam->Params[0].intParam-1)], TempParam->Params[1].intParam, Hands-free Profile Commands, ServiceDiscovery, OpenAudioGatewayClient, DEFAULT_AG_SUPPORTED_FEATURES, DEFAULT_CALL_HOLDING_SUPPORT, 0, NULL, HFRE_Event_Callback, (unsigned long)0)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Open_Remote_Hands-free_Port(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, unsigned int RemoteServerPort, unsigned long SupportedFeaturesMask, unsigned long CallHoldSupportMask, unsigned int NumberAdditionalIndicators, HFRE_Control_Indicator_Entry_t AdditionalSupportedIndicators[], HFRE_Event_Callback_t EventCallback, unsigned long CallbackParameter)

API 说明

以下函数负责在指定的远程设备上打开一个远程免提端口。此函数将蓝牙协议栈的蓝牙协议栈 ID (用于打开 HFRE 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)。第三个参数指定本地音频网关支持的功能。下一个参数是位掩码, 指定支持的呼叫保持和多方处理功能。此函数的第五个参数是出现在前一个参数表示的列表中的指示器名称的数量。下一个参数是此音频网关支持的其他指示器的列表。如果“其他指示器”参数为 NULL 且“其他指示器的数量”为零, 则不支持其他参数。最后两个参数分别指定 HFRE 事件回调函数和 HFRE 事件回调的回调参数, 此回调用于处理与指定远程端口 (正在打开状态、关闭状态等) 的任何进一步交互。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, 则返回值表示 HFRE 端口 ID, 可将该 ID 传递给所有需要它的其他函数。一旦打开远程免提装置, 只能通过调用 HFRE_Close_Port() 函数来关闭该装置 (传递此函数的返回值)。

备注

必需免提指示器 (call、service 和 call_set-up) 会自动添加到列表中, 无需指定为其他指示器。

7.3 管理音频

说明

以下函数负责建立或释放音频连接。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

“管理音频”命令只需要一个表示“管理音频”模式的整数值参数。该值必须指定为 0 (表示释放) 或 1 (表示建立)。

命令调用示例

- “ManageAudio 0” 尝试释放音频连接。
- “ManageAudio 1” 尝试建立音频连接。

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_set-up_Audio_Connection(BluetoothStackID, CurrentClientPortID) 或
HFRE_Release_Audio_Connection(BluetoothStackID, CurrentClientPortID)

API 原型

int BTPSAPI HFRE_set-up_Audio_Connection(unsigned int BluetoothStackID, unsigned int HFREPortID) 或 int
BTPSAPI HFRE_Release_Audio_Connection(unsigned int BluetoothStackID, unsigned int HFREPortID)

API 说明

此函数负责在本地和远程设备之间建立音频连接。此函数可由存在有效服务级别连接的音频网关或免提装置使用。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 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。

(或) 此函数负责释放先前由远程设备或通过调用 HFRE_set-up_Audio_Connection() 函数建立的音频连接。此函数可由音频网关或免提装置使用。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 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。

7.4 更新控制指示器

说明

以下函数负责更新远程免提装置上控制指示器的当前状态。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。指示器名称：呼叫状态、呼叫建立、服务可用性、信号指示器、漫游指示器、批量指示器和呼叫保持。

参数

此命令需要两个参数：指示器名称和值。

命令调用示例

- "UpdateControllIndicators Call set-up 1"
- "UpdateControllIndicators Call Held 3"

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Update_Current_Control_Indicator_Status(BluetoothStackID, PortToUse, 1, &HFREIndicatorUpdate)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Update_Current_Control_Indicator_Status(unsigned int BluetoothStackID, unsigned int HFREPortID, unsigned int NumberUpdateIndicators, HFRE_Indicator_Update_t UpdateIndicators[])

API 说明

以下函数负责更新当前控制指示器状态。此函数只能由具有有效服务级别连接的音频网关或已接收到 HFRE_Control_Indicator_Request_Indication 事件的音频网关执行。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。此函数的第三个参数是列表中存在的名称/值对的数量。此函数的最后一个参数是要更新的指示器的名称/值对列表。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。

7.5 呼叫等待

备注

仅当远程 HF 设备支持呼叫等待功能时，此函数才有效。

说明

以下函数负责向远程免提装置发送呼叫等待通知。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。如果 HF 设备支持远程设备呼叫等待功能，则此函数有效。

参数

“呼叫等待”命令仅接受一个参数，即在此响应中要发送的电话号码。

命令调用示例

```
"CallWaiting +9198787899889"
```

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED

- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Send_Call_Waiting_Notification(BluetoothStackID, CurrentClientPortID, TempParam->Params->strParam)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Send_Call_Waiting_Notification(unsigned int BluetoothStackID, unsigned int HFREPortID, char *PhoneNumber)

API 说明

此函数负责向远程设备发送呼叫等待通知。此函数只能由已启用呼叫等待通知且具有有效服务级别连接的音频网关执行。此函数将 HFRE 端口 ID 作为第一个输入参数。最后一个参数是作为此响应中的参数之一所需的电话号码。此参数是指向以 NULL 为终止符的字符串（如果指定）的指针，且长度必须小于 HFRE_PHONE_NUMBER_LENGTH_MAXIMUM。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

为空白字符串的 PhoneNumber 参数传递 NULL 以指定不存在电话号码的做法是有效的。

7.6 设置语音识别激活

说明

以下函数负责停用音频网关上的语音识别激活以及更改免提装置上的语音识别激活状态。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

语音识别命令仅接受一个参数，该值必须指定为 0（表示停用）或 1（表示激活）。

命令调用示例

"SetVoiceRecognitionActivation 0"

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Set_Remote_Voice_Recognition_Activation(BluetoothStackID, CurrentClientPortID, (Boolean_t)TempParam->Params->intParam)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Set_Remote_Voice_Recognition_Activation(unsigned int BluetoothStackID, unsigned int HFREPortID, Boolean_tVoiceRecognitionActive)

API 说明

由免提装置调用时，此函数负责激活和停用位于远程音频网关上的语音识别。由音频网关调用时，此函数负责向远程免提装置通知本地语音识别功能的当前激活状态。此函数只能由为语音识别启用了受支持功能位的本地设备进行调用。该函数将以下内容作为输入参数：**HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID**。最后一个参数是一个布尔标志，指定此调用要在远程音频网关上激活或停用此功能，还是指定为远程免提装置在本地激活或停语音识别。激活时，音频网关上的语音识别功能将打开；未激活时，音频网关上的语音识别功能将关闭。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。

7.7 设置扬声器增益

说明

以下函数负责设置远程设备上的扬声器增益。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。如果 HF 设备支持远程音频音量控制功能，则此函数有效。

参数

此命令只需要扬声器增益整数值作为参数。

命令调用示例

- "SetSpeakerGain 5"
- "SetSpeakerGain 6"

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Set_Remote_Speaker_Gain(BluetoothStackID, CurrentClientPortID, TempParam->Params->intParam)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Set_Remote_Speaker_Gain(unsigned int BluetoothStackID, unsigned int HFREPortID, unsigned int SpeakerGain)

API 说明

此函数负责允许同步和设置远程设备扬声器增益。仅当存在有效服务级别连接时，才能执行此函数。由免提装置调用时，此函数用于向远程音频网关通知当前扬声器增益值。由音频网关调用时，此函数可为音频网关提供一种方法来控制远程免提装置的扬声器增益。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。最后一个参数是要发送到远程设备的扬声器增益。扬声器增益参数 *必须* 介于 HFRE_SPEAKER_GAIN_MINIMUM 和 HFRE_SPEAKER_GAIN_MAXIMUM 的值之间。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。

7.8 设置麦克风增益

说明

以下函数负责设置远程设备上的麦克风增益。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。如果 HF 设备支持远程音频音量控制功能，则此函数有效。

参数

此命令只需要麦克风增益整数值作为参数。

命令调用示例

- "SetMicroPhoneGain 5"
- "SetMicroPhoneGain 6"

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Set_Remote_Microphone_Gain(BluetoothStackID, CurrentClientPortID, TempParam->Params->intParam)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Set_Remote_Microphone_Gain(unsigned int BluetoothStackID, unsigned int HFREPortID, unsigned int MicrophoneGain)

API 说明

此函数负责允许同步和设置远程设备麦克风增益。仅当存在有效服务级别连接时，才能执行此函数。由免提装置调用时，此函数用于向远程音频网关通知当前麦克风增益值。由音频网关调用时，此函数可为音频网关提供一种方法来控制远程免提装置的麦克风增益。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。最后一个参数是要发送到远程设备的麦克风增益。麦克风增益参数 *必须* 介于 HFRE_MICROPHONE_GAIN_MINIMUM 和 HFRE_MICROPHONE_GAIN_MAXIMUM 的值之间。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。

7.9 禁用远程声音增强

备注

仅当存在有效服务级别连接但不存在音频连接时，才能执行此函数。

说明

以下函数负责禁用远程设备上的声音增强。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。仅当存在有效服务级别连接但不存在音频连接时，才能执行此函数。

参数

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

命令调用示例

```
"DisableRemoteSoundEnhancement"
```

可能的返回值

- (0) 命令发送成功
- (-2) `BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID`
- (-4) `FUNCTION_ERROR`
- (-8) `INVALID_STACK_ID_ERROR`
- (-1000) `BTHFRE_ERROR_INVALID_PARAMETER`
- (-1001) `BTHFRE_ERROR_NOT_INITIALIZED`
- (-1002) `BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID`
- (-1003) `BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR`
- (-1004) `BTHFRE_ERROR_INSUFFICIENT_RESOURCES`
- (-1005) `BTHFRE_ERROR_INVALID_OPERATION`
- (-1006) `BTHFRE_ERROR_INVALID_CODEC_ID`

API 调用

```
HFRE_Disable_Remote_Echo_Cancelation_Noise_Reduction(BluetoothStackID, CurrentClientPortID)
```

API 调用

```
BTPSAPI_DECLARATION int BTPSAPI HFRE_Disable_Remote_Echo_Cancelation_Noise_Reduction(unsigned int BluetoothStackID, unsigned int HFREPortID)
```

API 说明

此函数负责禁用远程设备上的回声消除和降噪。此函数可由免提装置和音频网关（存在有效服务级别连接但不存在音频连接）执行。该函数将以下内容作为输入参数：`HFRE` 端口 ID 有效的蓝牙协议栈 ID 以及 `HFRE` 端口 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

一旦禁用此功能便无法启用，因为相关规范未提供重新启用此功能的方法。此功能将保持禁用状态，直到断开当前服务级别连接。

7.10 发送来电显示通知

说明

以下函数负责向远程免提装置发送呼叫线路识别通知。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

此命令仅需要一个参数，即呼叫者 ID (电话号码)。

命令调用示例

```
"SendCallerIdNotification +9198787899889"
```

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

```
HFRE_Send_Call_Line_Identification_Notification(BluetoothStackID, CurrentClientPortID, TempParam->Params->strParam)
```

API 原型

```
BTPSAPI_DECLARATION int BTPSAPI HFRE_Send_Call_Line_Identification_Notification(unsigned int BluetoothStackID, unsigned int HFREPortID, char *PhoneNumber)
```

API 说明

此函数负责向远程设备发送呼叫线路识别通知。此函数只能由已启用呼叫线路识别通知且具有有效服务级别连接的音频网关执行。该函数将以下内容作为输入参数：**HFRE** 端口 ID 有效的蓝牙协议栈 ID 以及 **HFRE** 端口 ID。最后一个参数是作为此响应中的参数之一所需的电话号码。此参数应为指向以 **NULL** 为终止符的字符串的指针，且其长度 *必须* 介于 **HFRE_PHONE_NUMBER_LENGTH_MINIMUM** 和 **HFRE_PHONE_NUMBER_LENGTH_MAXIMUM** 的值之间。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。

7.11 设置振铃指示

说明

以下函数负责向远程免提装置发送启用或禁用带内振铃指示。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

此命令只需要一个参数。该值必须指定为 0 (表示禁用) 或 1 (表示启用)。

命令调用示例

- “SetRingIndication 1” 表示启用带内振铃指示
- “SetRingIndication 0” 表示禁用带内振铃指示

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Enable_Remote_InBand_Ring_Tone_Setting(BluetoothStackID, CurrentClientPortID, TempParam->Params[0].intParam? TRUE : FALSE)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Enable_Remote_InBand_Ring_Tone_Setting(unsigned int BluetoothStackID, unsigned int HFREPortID, Boolean_t EnableInBandRing)

API 说明

此函数负责启用或禁用本地设备的带内铃声功能。此函数只能由存在有效服务级别连接的音频网关执行。仅当本地音频网关支持带内铃声功能时，才能使用此函数启用这些功能。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。最后一个参数是一个布尔标志，指定此调用要启用还是禁用该功能。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。

7.12 振铃指示

说明

以下函数负责向远程免提装置发送振铃指示。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参数。参数对 RingIndication 的结果没有影响。

命令调用示例

“RingIndication” 发送振铃指示。

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR

- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Ring_Indication(BluetoothStackID, CurrentClientPortID)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Ring_Indication(unsigned int BluetoothStackID, unsigned int HFREPortID)

API 说明

此函数负责向远程免提装置发送振铃指示。此函数只能由存在有效服务级别连接的音频网关执行。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 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。

7.13 发送来电状态

说明

以下函数负责向远程设备发送响应和保持命令 (+BTRH)。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

此命令只需要一个表示来电状态的整数值参数。该值必须指定为 0 (来电保持)、1 (接受保持呼叫) 和 2 (拒绝保持呼叫)。

命令调用示例

“SendIncomingCallState 1” — 如果有来电。

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Send_Incoming_Call_State(BluetoothStackID, CurrentClientPortID, (HFRE_Call_State_t) CallState)

API 原型

BTPSAPI_DECLARATION int BTPSAPI HFRE_Send_Incoming_Call_State(unsigned int BluetoothStackID, unsigned int HFREPortID, HFRE_Call_State_t CallState)

API 说明

以下函数负责发送有关来电状态的信息。此函数只能由具有有效服务级别连接的音频网关执行。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。此函数的最后一个参数指示要在该消息中设置的呼叫状态。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。

7.14 关闭 AG 客户端

说明

以下函数负责关闭任何打开的 HFP 端口。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

使用此命令不需要参数。参数对 Close 的结果没有影响。

可能的返回值

- (0) 客户端成功关闭
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

HFRE_Close_Port(BluetoothStackID, CurrentClientPortID)

API 原型

int BTPSAPI HFRE_Close_Port(unsigned int BluetoothStackID, unsigned int HFREPortID)

API 说明

可以使用以下函数来关闭先前由以下任何机制打开的 HFRE 端口：

- 成功调用 HFRE_Open_Remote_Hands-free_Port() 函数。
- 成功调用 HFRE_Open_Remote_Audio_Gateway_Port() 函数。
- 使用 HFRE_Open_Hands-free_Server_Port() 或 HFRE_Open_Audio_Gateway_Server_Port() 函数打开服务器端口的传入打开请求（免提或音频网关）。

该函数将以下内容作为输入：打开的 HFRE 端口所在蓝牙协议栈的 ID 以及要关闭的端口的 HFRE 端口 ID（上述 Open 函数之一的返回值）。如果成功，此函数返回零；如果出现错误，则返回负值。此函数不会从系统中取消注册 HFRE 服务器端口，而是仅断开当前在服务器端口上处于活动状态的任何连接。HFRE_Close_Server_Port() 函数可用于取消注册 HFRE 服务器端口。

7.15 发送运营商信息

备注

从远程免提设备收到查询网络运营商选择请求时，将执行此函数。

说明

以下函数负责发送命令以向远程免提设备发送运营商选择响应。从远程免提设备收到查询远程网络运营商选择请求时，将执行此函数。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。

参数

“发送运营商信息”命令只需要一个参数，即当前网络运营商。

命令调用示例

```
"SendOperatorInfo airtel"
```

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

```
HFRE_Send_Network_Operator_Selection(BluetoothStackID, CurrentClientPortID, 1, TempParam->Params[0].strParam)
```

API 原型

```
BTPSAPI_DECLARATION int BTPSAPI HFRE_Send_Network_Operator_Selection(unsigned int BluetoothStackID, unsigned int HFREPortID, unsigned int NetworkMode, char*NetworkOperator)
```

API 说明

以下函数负责发送网络运营商信息。此函数只能由已收到查询远程网络运营商选择请求的音频网关执行。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。此函数的第三个参数是当前网络模式。最后一个参数是当前网络运营商。此参数是指向以 NULL 为终止符的字符串（如果指定）的指针，且长度必须小于 HFRE_NETWORK_OPERATOR_LENGTH_MAXIMUM。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

为空白字符串的 NetworkOperator 参数传递 NULL 以指定不存在网络运营商的做法是有效的。

7.16 发送用户号码

备注

从远程免提设备收到查询用户号码信息请求时，将执行此函数。

说明

以下函数负责发送命令以向远程免提设备发送用户号码。从远程免提设备收到查询用户号码信息请求时，将执行此函数。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。

参数

“发送用户号码”命令只需要一个参数，即在此响应中要发送的电话号码。

命令调用示例

```
"SendSubNumber +9198787899889"
```

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

```
HFRE_Send_Subscriber_Number_Information(BluetoothStackID, CurrentClientPortID, TempParam->Params[0].strParam, 4, HFRE_DEFAULT_NUMBER_FORMAT, TRUE)
```

API 原型

```
BTPSAPI_DECLARATION int BTPSAPI HFRE_Send_Subscriber_Number_Information(unsigned int BluetoothStackID, unsigned int HFREPortID, char *PhoneNumber, unsigned intServiceType, unsigned int NumberFormat, Boolean_t FinalEntry)
```

API 说明

以下函数负责发送用户号码信息。此函数只能由已收到查询用户号码信息请求的音频网关执行。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。此函数的第三个参数是在此响应中要发送的电话号码。电话号码参数字符串长度 *必须* 介于 HFRE_PHONE_NUMBER_LENGTH_MINIMUM 和 HFRE_PHONE_NUMBER_LENGTH_MAXIMUM 的值之间。此函数的第四个参数是与指定电话号码相关的服务类型。此函数的第五个参数是用于该号码的号码格式。此函数的最后一个参数是一个布尔值，指示这是否是要发送的最后一个用户号码信息条目（因此还需要发送 OK）。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。

7.17 发送呼叫列表

备注

从远程免提设备收到查询当前呼叫列表信息请求时，将执行此函数。

说明

以下函数负责发送命令以向远程免提设备发送呼叫条目。从远程免提设备收到查询远程当前呼叫列表请求时，将执行此函数。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。

参数

“发送呼叫列表”命令接受六个参数：索引、呼叫方向、呼叫状态、呼叫模式、多方、电话号码（顺序相同）。

命令调用示例

```
"SendCallList 0 1 1 0 0 5551212"
```

可能的返回值

- (0) 命令发送成功
- (-2) BTPS_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-4) FUNCTION_ERROR
- (-8) INVALID_STACK_ID_ERROR
- (-1000) BTHFRE_ERROR_INVALID_PARAMETER
- (-1001) BTHFRE_ERROR_NOT_INITIALIZED
- (-1002) BTHFRE_ERROR_INVALID_BLUETOOTH_STACK_ID
- (-1003) BTHFRE_ERROR_LIBRARY_INITIALIZATION_ERROR
- (-1004) BTHFRE_ERROR_INSUFFICIENT_RESOURCES
- (-1005) BTHFRE_ERROR_INVALID_OPERATION
- (-1006) BTHFRE_ERROR_INVALID_CODEC_ID

API 调用

```
HFRE_Send_Current_Calls_List(BluetoothStackID, CurrentClientPortID, &CurrentCallListEntry, Final)
```

API 原型

```
BTPSAPI_DECLARATION int BTPSAPI HFRE_Send_Current_Calls_List(unsigned int BluetoothStackID, unsigned int HFREPortID, HFRE_Current_Call_List_Entry_t*CurrentCallListEntry, Boolean_t FinalEntry)
```

API 说明

以下函数负责向远程设备发送当前呼叫列表条目。此函数只能由已收到查询远程当前呼叫列表请求的音频网关执行。该函数将以下内容作为输入参数：HFRE 端口 ID 有效的蓝牙协议栈 ID 以及 HFRE 端口 ID。此函数的第三个参数是要发送的当前呼叫列表条目。此函数的最后一个参数是一个布尔值，指示这是否是要发送的最后一个呼叫列表条目（因此还需要发送 OK）。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

如果第三个参数指定为 NULL，则最后一个参数 *必须* 指定这是最后一个条目。在这种情况下，不能发送呼叫列表条目，但可以发送终止响应。

备注

此函数不会将电话簿名称作为呼叫列表条目的一部分发送。如果希望随条目一起发送电话簿名称，请使用 `HFRE_Send_Current_Calls_List_With_Phonebook_Name`。

8 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。

9 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

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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

版权所有 © 2022, 德州仪器 (TI) 公司

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

Copyright © 2023，德州仪器 (TI) 公司