The following document contains information on Cypress products. Although the document is marked with the name "Broadcom", the company that originally developed the specification, Cypress will continue to offer these products to new and existing customers.

CONTINUITY OF SPECIFICATIONS

There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

CONTINUITY OF ORDERING PART NUMBERS

Cypress continues to support existing part numbers. To order these products, please use only the Cypress Ordering Part Number listed in the table.

Broadcom Ordering Part Number	Cypress Ordering Part Number
BCM20706	CYW20706
BCM920706V2_EVAL	CYW920706V2_EVAL
BCM20729	CYW20729
BCM20719	CYW20719

FOR MORE INFORMATION

Please visit our website at www.cypress.com or contact your local sales office for additional information about Cypress products and services.

OUR CUSTOMERS

Cypress is for true innovators – in companies both large and small.

Our customers are smart, aggressive, out-of-the-box thinkers who design and develop game-changing products that revolutionize their industries or create new industries with products and solutions that nobody ever thought of before.

ABOUT CYPRESS

Founded in 1982, Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, home automation and appliances, consumer electronics and medical products. Cypress's programmable systems-on-chip, general-purpose microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first.

Cypress is committed to providing customers with the best support and engineering resources on the planet enabling innovators and out-of-the-box thinkers to disrupt markets and create new product categories in record time. To learn more, go to www.cypress.com.

Software User Manual

WICED

Sample HCI UART Control Protocol

BROADCOM
MASS MARKET
PLATFORM

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

WICED-SWUM115-R
January 12, 2017

Revision History

Revision	*Date*	*Change Description*
WICED-SWUM115-R	12/06/2016	**Added:** • HF Accept/Reject Audio Connection Command • HF Audio Connection Request Event • LE Connection Parameters **Updated:** • Table 152: HF Response Event Details
WICED-SWUM114-R	12/06/2016	**Updated:** • Audio Data
WICED-SWUM113-R	10/10/2016	**Updated:** • AVRC Controller Setting Change
WICED-SWUM112-R	09/26/2016	**Added:** • GATT Event Read Error • GATT Event Write Request Error
WICED-SWUM111-R	09/21/2016	**Updated:** • Table 32: GATT Command Read Response
WICED-SWUM110-R	09/19/2016	**Added:** • ANCS Connect • ANCS Disconnect • AMS Commands— HCI_CONTROL_GROUP_AMS • ANCS Command Status • ANCS Service Found • ANCS Connected • ANCS Disconnected • AMS Events—HCI_CONTROL_GROUP_AMS **Updated:** • Introduction • Table 1: WICED HCI Control Protocol Command and Event Groups • Table 159: Audio Connected Event
WICED-SWUM109-R	08/18/2016	**Added:** • LE Peer MTU **Updated:** • GATT Characteristic Discovered • GATT Descriptor Discovered • Audio Data Request • Audio Connected • AVRC Target Connected
WICED-SWUM108-R	08/08/2016	**Added:** • LE Set Connection Parameters • Start Bond • Read Buffer Pool Usage Statistics **Updated:** • AVRC Target Disconnect • AVRC Controller Disconnect
WICED-SWUM107-R	07/19/2016	**Updated:** • AVRC Controller Current Track Info • AVRC Target Connected
WICED-SWUM106-R	07/13/2016	**Updated:** • SPP TX Complete • LE Re Pair • Hyperlinks

WICED-SWUM105-R	07/12/2016	**Updated:** • AVRC Controller Play Position • AVRC Controller Settings Available • Trace Enable • Test Commands— HCI_CONTROL_GROUP_TEST • Command Status • Pairing Completed • Encryption Changed **Added:** • Test Events— HCI_CONTROL_GROUP_TEST • Maximum Number of Paired Devices Reached
WICED-SWUM104-R	07/07/2016	**Updated:** • BT-SDK 3.7 naming updates
WICED-SWUM103-R	06/07/2016	**Updated:** • Table 1: "WICED HCI Control Protocol Command and Event Groups," on page 25 • "AV Remote Control Target Commands— HCI_CONTROL_GROUP_AVRC_TARGET" on page 57 • "AV Remote Control Controller Commands — HCI_CONTROL_GROUP_AVRC_CONTROLLER" on page 61 • Table 110: "Command Status Event," on page 83 • "Device Events—HCI_CONTROL_GROUP_DEVICE" on page 83 • "AV Remote Control Controller Events— HCI_CONTROL_GROUP_AVRC_CONTROLLER" on page 105 **Added:** • "Read Local Bluetooth Device Address" on page 30 • "LE Set Channel Classification" on page 35 • "IAP2 Get Auth Chip Info" on page 74 • "SPP Command Status" on page 101 • "Audio Command Status" on page 102 • "AV Remote Control Target Events— HCI_CONTROL_GROUP_AVRC_TARGET" on page 110 • "IAP2 Auth Chip Info" on page 124
WICED-SWUM102-R	02/20/2016	**Added:** • "LE Get Identity Address" on page 34 • "LE Identity Address" on page 83
WICED-SWUM101-R	02/19/2016	**Added:** • "Enable Coexistence" on page 29 • "Disable Coexistence" on page 29
WICED-SWUM100-R	01/29/2016	Initial release

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Table of Contents

List of Figures

List of Tables

About This Document

Purpose and Audience

This document provides information on an HCI UART control protocol. The protocol is an implementation example of how a host microcontroller unit (MCU) can communicate with a Broadcom® WICED™ device via HCI UART.

This document is intended for application developers creating and testing designs based on Broadcom WICED™ Bluetooth Software Development Kit (BT SDK) Bluetooth devices.

Scope

Several paragraphs in the document refer the reader to variables and data structures that are not described in this document. For information on the variables and data structures mentioned in this document, see the API Reference Guide that is provided with the SDK.

References

The references in this section may be used in conjunction with this document.

Note: Broadcom provides technical documentation and software from the Broadcom Support Community website (community.broadcom.com).

For Broadcom documents, replace the "xx" in the document number with the largest number available in the repository to ensure that you have the most current version of the document.

Document (or Item) Name	Number	Source
Broadcom Items		
[1] WICED BT SDK Development System, Quick Start Guide	WICED-BT-SDK-2070x-QSG100-R	Broadcom community website.
Other Items		
[2] Bluetooth Core Specification, Version 4.2	–	www.bluetooth.org

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

For a comprehensive list of acronyms and other terms used in Broadcom documents, go to: http://www.broadcom.com/press/glossary.php.

Technical Support

Broadcom provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates.

Customers who work directly with Broadcom, can acquire technical documentation and software from the Broadcom CSP (support.broadcom.com). For a CSP account, contact your Broadcom Sales or Engineering support representative.

Customers who work with Broadcom through a distribution channel can acquire technical documentation and software from the Broadcom Support Community website (community.broadcom.com).

Hardware and Software Prerequisites

To fully use the content provided in this document, readers will need the following items:

- One BCM20706-based Bluetooth (BT) device and a second BT device, which can also be BCM20706-based.
- Version 3.7.x of the Broadcom WICED BT SDK, which includes several applications that use the HCI control protocol defined in this document.
- A Broadcom-supplied ClientControl.exe sample application (from the WICED BT SDK).
- A PC running Windows 7 or higher.

Note: A PC running a Windows OS and the ClientControl.exe application is used in place of an external MCU to send commands to and receive replies and asynchronous events from a BCM20706.

To prepare a BCM20706-based BLE device, build an application from the WICED BT SDK that uses the HCI control protocol defined in this document. For help doing such a build, see the WICED BT SDK Development System Quick Start Guide *[1]*.

Note: Throughout the document, references to the hci_control application are actually references to the application running on the BCM20706 that supports the HCI control protocol defined in this document. The WICED BT SDK contains several such applications (for example, hci_handsfree_plus and hci_av_source_plus).

Section 1: Introduction

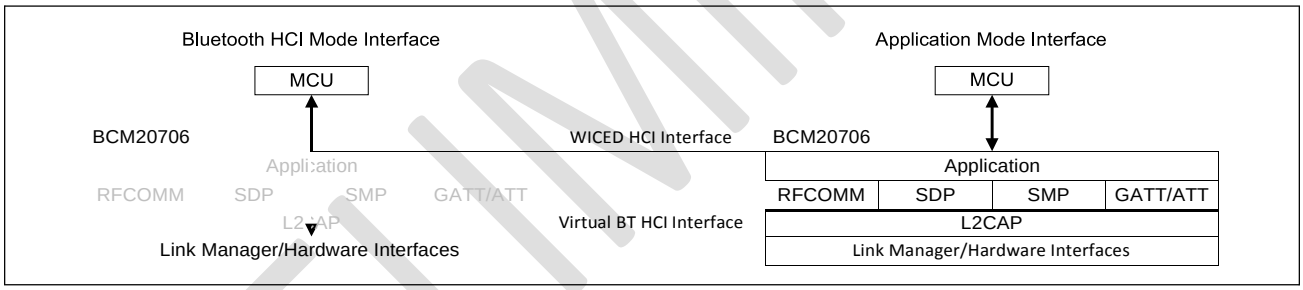
The Broadcom WICED BT SDK includes sample applications that can be executed on BCM20706 devices.

A real Bluetooth product could have an onboard MCU that uses the BCM20706 to provide Bluetooth functionality. For such a product, MCU software would likely be used to control the BCM20706 through a UART or SPI interface via a protocol that allows the MCU to send and receive commands, events, and data. This document describes a sample protocol for communication between an MCU and a BCM20706.

The BCM20706 supports two operating modes: the Bluetooth Host Controller Interface (HCI) mode and the Application mode. In the Bluetooth HCI mode, the embedded stack in the BCM20706 is not exercised and the BCM20706 behaves as a standard Bluetooth HCI controller. A standard Bluetooth HCI controller supports the Bluetooth UART HCI interface as defined in the Bluetooth Core specification *[2]*. In the Application mode, the embedded stack in the BCM20706 is used and the BCM20706 does not behave as a standard Bluetooth controller.

Figure 1 shows the Bluetooth HCI Mode and Application Mode logical interfaces. In the Bluetooth HCI mode, the MCU communicates to the BCM20706 using the standard Bluetooth HCI protocol. In the application mode, the MCU uses the WICED™ HCI protocol defined in this document.

Figure 1: BCM20706 MCU Interfaces in the Bluetooth HCI and Application Modes



This document provides a sample UART control protocol, referred to as the WICED HCI Control protocol, that can be used in the Application mode to support communication between an MCU (host) and an application running on the BCM20706 (controller). The combination of the ClientControl.exe application (hereinafter referred to as ClientControl) running on a PC and the application running on a BCM20706 provides a sample implementation of the WICED HCI Control protocol.

When the BCM20706 powers on, boot logic determines whether a serial flash is connected and, if so, whether it contains a valid application image. If there is a valid application, the BCM20706 loads and executes the application. If there is no serial flash, then the BCM20706 boots into and stays in the Bluetooth HCI mode where it waits for MCU (host) commands. While in Bluetooth HCI mode, the standard Bluetooth HCI protocol is used to download an application to the BCM20706 and change the device mode to Application mode.

The procedure for downloading an application is described in "Downloading an Application and Configuration Data Using ClientControl.exe". The procedure is not applicable when serial flash contains a valid application.

The WICED HCI Control protocol is defined in "WICED HCI Control Protocol Definition".

Section 2: Downloading an Application and Configuration Data Using ClientControl.exe

Introduction

This section shows how to use a ClientControl.exe application executing on a host PC (in place of a host MCU) to download an embedded application and its associated configuration data to a BCM20706.

Note: The procedure in this section assumes use of the Broadcom BCM920706V2_EVAL evaluation board. For devices that have On-Chip Flash (OCF), e.g. the BCM20729, or BCM20719, this section does not apply. The On-Chip Flash shall be programmed via the SDK using the “download” make directive.

Downloading the BCM20706 Controller Target Application

To download a target application to the BCM20706 controller on a BCM920706V2_EVAL board, perform the following steps:

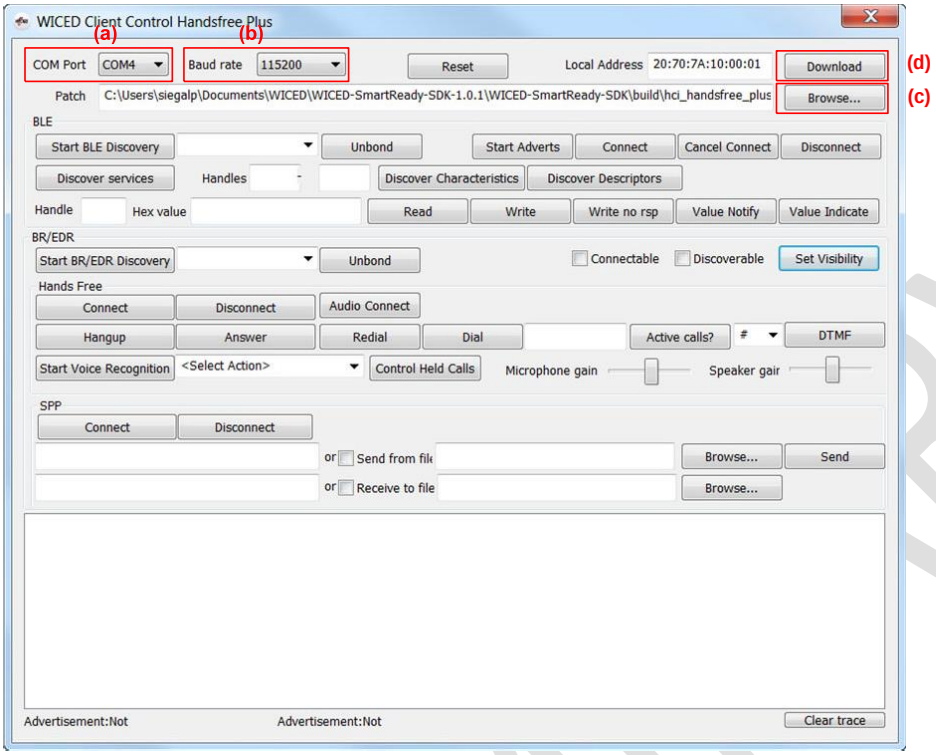
1. Build the BCM20706 controller target application using the WICED BT SDK.

   To do this, double-click an hci_control application target in the Make Target pane. For example, double-click:

   `hci_handsfree_plus-BCM920706_P49 DIRECT_LOAD=1 build`

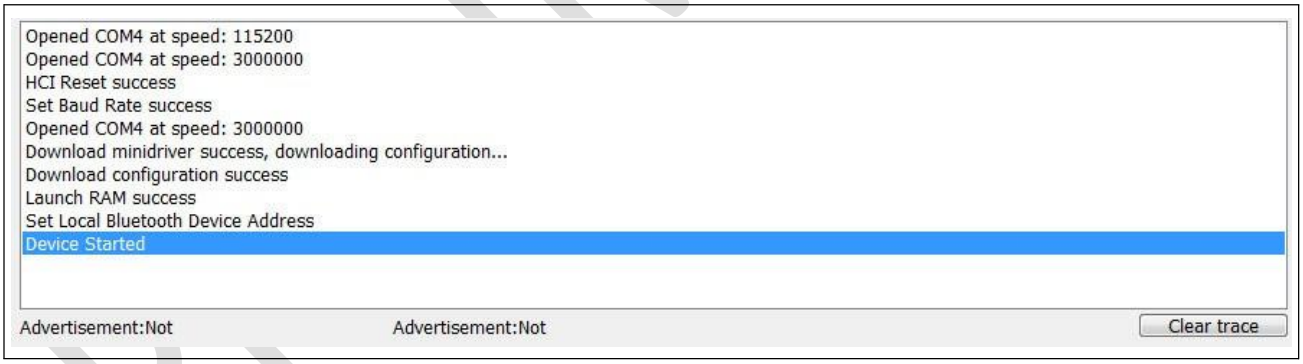
2. In the WICED BT SDK, navigate to: Apps/hci_handsfree_plus/ClientControl/Release and double- click **ClientControl.exe**.
3. In the ClientControl application:
   a. In the **COM Port** menu, select the COM port associated with the BCM20706 HCI UART.
   b. Set the baud rate to 115200.
   c. Click **Browse** and select the (*.hcd) file built earlier (in Step 1 above). On a Windows system, if the default installation path was used, the file will be located under the My Documents (or Documents) folder at a path similar to the following:
   ...WICED\WICED-BT-SDK-3.7\BCM20706-A2\build\hci_handsfree_plus-BCM920706_P49-rom-ram-Wiced-release\hci_handsfree_plus-BCM920706_P49-rom-ram-Wiced-release.hcd.

d. Click **Download**.



After clicking **Download**, messages similar to those shown in Figure 2 will appear in the Client Control console:

Figure 2: Client Control Console Showing Messages Following a Successful Download



The commands as well as the responses and events behind the console messages shown in Figure 2 are all conveyed using the Bluetooth UART HCI protocol as defined in the Bluetooth Core specification *[2]*. Information on the console messages shown in Figure 2 is provided in HCI Commands and Events During a Download.

HCI Commands and Events During a Download

After a download is initiated (by clicking **Download** in the ClientControl application), host and controller messages are exchanged in the following sequence (which is represented by the console messages in Figure 2):

1. The PC (MCU) host issues the following standard Bluetooth `HCI_RESET` command:
   `01 03 0C 00`

   The following response is expected from the BCM20706 controller within 100 ms:
   `04 0E 04 01 03 0C 00`

2. To speed up application downloading, the MCU host commands the BCM20706 to communicate at a new, higher rate by issuing the following Vendor-Specific `UPDATE_BAUDRATE` command:
   `01 18 FC 06 00 00 xx xx xx xx`

   In the above command, the `xx xx xx xx` bytes specify the 32-bit little-endian value of the new rate in bits per second. For example, 115200 is represented as `00 C2 01 00`.

   The following response to the `UPDATE_BAUDRATE` command is expected within 100 ms:
   `04 0E 04 01 18 FC 00`

3. The host switches to the new baud rate after receiving the response at the old baud rate.

4. If successful, the host issues the following `DOWNLOAD_MINIDRIVER` command:
   `01 2E FC 00`

   The following response is expected from the BCM20706 controller within 100 ms:
   `04 0E 04 01 2E FC 00`

5. After successfully downloading the mini-driver, the host writes application code and configuration data to the BCM20706 by sending `WRITE_RAM` commands. Since the writes are destined for BCM20706 RAM, the destination addresses in the `WRITE_RAM` commands are absolute RAM locations.

   The following `WRITE_RAM` command is an example:
   `01 4C FC nn xx xx xx xx yy yy yy …`

   In the above `WRITE_RAM` command:

   - `nn` is 4 + N, which represents 4 address bytes plus N payload bytes.
   - `xx xx xx xx` is the 4-byte, absolute RAM address in little-endian order.
   - `yy yy yy …` are the N payload bytes to be loaded into the addressed RAM location.

   The following response to each `WRITE_RAM` command is expected within 200 ms:
   `04 0E 04 01 4C FC 00`

6. After the host has written all application and configuration data to RAM, it sends a LAUNCH_RAM command with the address stored in the last record of the hardware configuration data (HCD) file.

   An example LAUNCH_RAM command is shown here:

   ```
   01 4E FC 04 xx xx xx xx
   ```

   In the above LAUNCH_RAM command, xx xx xx xx is the 4-byte, absolute RAM address of the last HCD record in little-endian order. Typically, the last address is 0xFFFFFFFF.

   The following response to the LAUNCH_RAM command is expected within 200 ms:

   ```
   04 0E 04 01 4E FC 00
   ```

Note: Following a successful LAUNCH_RAM command, the device is in the Application mode and the application is running.

Note: In the application mode, the UART configuration depends on the application. If the application sets the baud rate to 3 Mbps at start-up then the MCU or ClientControl.exe running on a Windows PC must also configure the UART for 3 Mbps operation to successfully communicate with the BCM20706. The application sets the baud rate using the following command: `uart_SetBaudrate(0, 0, 3000000)`. To set the UART rate via the host, see "Set Baud Rate".

Section 3: WICED HCI Control Protocol Definition

While in the Application mode, the BCM20706 uses the following 5-byte packet header for command/event exchanges with the host MCU.

Packet Type	*Command/ Event Code*	*Group Code*	*Packet Length*	
HCI_WICED_PKT(0x19)	HCI_CONTROL_ COMMAND_...	HCI_CONTROL_ GROUP_...	Low byte	High byte

The protocol follows the standard Bluetooth HCI rules for parameter byte ordering. For example, the attribute handle 0x210 is sent in two bytes, 0x10 followed by 0x02.

All commands and events are split into groups. Table 1 shows the groups defined by the WICED HCI Control protocol.

Table 1: WICED HCI Control Protocol Command and Event Groups

Group Name	*Group Value*	*Description*
HCI_CONTROL_GROUP_DEVICE	0x00	General control of BCM20706 management and Bluetooth functionality
HCI_CONTROL_GROUP_LE	0x01	LE device-related commands and events
HCI_CONTROL_GROUP_GATT	0x02	GATT commands and events
HCI_CONTROL_GROUP_HF	0x03	Hands-free profile commands, events, and data
HCI_CONTROL_GROUP_SPP	0x04	Serial port profile commands, events, and data
HCI_CONTROL_GROUP_AUDIO	0x05	Audio/video (AV) commands, events, and data
HCI_CONTROL_GROUP_HIDD	0x06	HID device (HIDD) commands and events
HCI_CONTROL_GROUP_AVRC_TARGET	0x07	AV remote control (AVRC) target commands and events
HCI_CONTROL_GROUP_TEST	0x08	Test commands
HCI_CONTROL_GROUP_TIME	0x0A	Current time client application events
HCI_CONTROL_GROUP_ANCS	0x0B	Apple Notification Center Service (ANCS) commands and events
HCI_CONTROL_GROUP_ALERT	0x0C	Immediate Alert Service (IAS) events
HCI_CONTROL_GROUP_LN	0x0D	Location and navigation commands and events.
HCI_CONTROL_GROUP_IAP2	0x0E	iPod Accessory Protocol implementation (iAP2) commands and events
HCI_CONTROL_GROUP_AG	0x0F	Hands-free Audio Gateway (AG) commands and events
HCI_CONTROL_GROUP_AIO_SERVER	0x10	Automation IO (AIO) server commands and events
HCI_CONTROL_GROUP_AIO_CLIENT	0x10	AIO client commands and events
HCI_CONTROL_GROUP_AVRC_CONTROLLER	0x11	AV remote control (AVRC) controller commands and events
HCI_CONTROL_GROUP_AMS	0x12	Apple Media Service (AMS) commands and events
HCI_CONTROL_GROUP_MISC	0xFF	Miscellaneous commands and events

See "WICED HCI Control Protocol Commands," for information on the WICED HCI Control protocol commands.

See "WICED HCI Control Protocol Events" for information on the WICED HCI Control protocol events.

Section 4: WICED HCI Control Protocol Commands

Device Commands— HCI_CONTROL_GROUP_DEVICE

The device commands allow the host to manage the behavior of the BCM20706.

Reset

The Reset command causes the BCM20706 to restart. After initialization completes, the BCM20706 sends a Device Started event (see "Device Started").

Table 2: Reset Command

Item	*Description*
Operating code	0x01
Parameters	–

Trace Enable

The Trace enable command instructs the BCM20706 to start or stop forwarding the WICED logs and virtual HCI traces.

The BCM20706 provides the following two trace types:

- An output of the WICED_BT_TRACE statements.
- A binary dump of the virtual HCI commands, events, and data packets between the embedded host stack and the BCM20706 controller.

The WICED_BT_TRACE output is forwarded in the HCI_CONTROL_EVENT_WICED_TRACE when a corresponding trace is enabled.

The virtual HCI traces are sent over UART using HCI_CONTROL_EVENT_HCI_DATA.

Table 3: Trace Enable Command

Item	*Description*	
Operating code	0x02	
Parameter	Bluetooth HCI trace enable (1 byte)	If true, HCI traces are routed through the WICED HCI interface to the host.
	WICED trace route (1 byte)	0: Traces are not generated. 1: Traces are forwarded to the WICED UART. 2: Traces are forwarded to the HCI UART. 3: Traces are forwarded to the debug UART. 4: Traces are forwarded to the peripheral UART.

Set Local Bluetooth Device Address

The Set Local Bluetooth Device Address command configures the BCM20706 to use a new Bluetooth device address. An MCU host typically sends this command during a start-up operation. The address is passed as a parameter in little-endian format.

Table 4: Set Local Bluetooth Device Address Command

Item	Description
Operating code	0x03
Parameter	A 6-byte Bluetooth device address

Set Baud Rate

The Set Baud Rate command instructs the BCM20706 to use a new baud rate for UART communications. An MCU may send this command during a start-up operation. It might not be needed if the MCU knows that the application on the BCM20706 sets a specific baud rate during start-up.

Table 5: Set Baud Rate Command

Item	Description	
Operating code	0x04	
Parameter	New baud rate (4 bytes)	As all other parameters in the HCI control protocol, the parameter is passed in little endian order. Example: To set a baud rate of 3 Mbps, send 0x2DC6C0 (3,000,000) as the following four-byte sequence: 0xC0 0xC6 0x2D 0x00.

Push NVRAM Data

If a BCM20706 does not have an embedded NVRAM, it relies on the MCU to save application-specific NVRAM data, which the BCM20706 can provide in NVRAM Data events (see "NVRAM Data"). At start-up, the MCU host should push all saved NVRAM information to the BCM20706 before the BCM20706 establishes any Bluetooth connections.

Table 6: Push NVRAM Data Command

Item	Description	
Operating code	0x05	
Parameters	nvram_id (2 bytes)	ID of an NVRAM information chunk
	nvram_data (variable bytes)	Data corresponding to nvram_id

Delete NVRAM Data

An application running on an MCU host may request the BCM20706 to delete NVRAM information for a specific nvram_id.

Table 7: Delete NVRAM Data Command

Item	Description	
Operating code	0x06	
Parameter	nvram_id	2-byte ID of an NVRAM information chunk

Inquiry

The Inquiry command lets an application cancel or start a Bluetooth Inquiry procedure.

If a device is found during an inquiry, the BCM20706 sends an Inquiry Result event (see "Inquiry Result").

When an Inquiry procedure completes, the BCM20706 sends an Inquiry Complete Event (see "Inquiry Complete").

Table 8: Inquiry Command

Item	Description	
Operating code	0x07	
Parameter	Enable (1 byte)	0: Cancel the Inquiry procedure.
		1: Start an Inquiry procedure.

Set Visibility

The Set Visibility command allows the host to turn Discoverability and Connectability on and off. After a BCM20706 restart, it is not discoverable (non-discoverable) and not connectable (non-connectable).

Note: Attempts to make the BCM20706 discoverable and non-connectable will be rejected because, according to the Bluetooth specifications, a discoverable device should also be connectable.

After the BCM20706 receives this command, it reports command success or failure in the Command Status event (see "Command Status"). .

Table 9: Set Visibility Command

Item	Description	
Operating code	0x08	
Parameters	Discoverability (1 byte)	0: Not discoverable
		1: Discoverable
	Connectability (1 byte)	0: Not connectable
		1: Connectable

Set Pairing Mode

The MCU can set the BCM20706 to be pairable or not pairable using this command. A BR/EDR connection will be rejected if a device is not pairable and there is no link key to secure the connection. Similarly, while a device is not pairable, access to LE characteristics requiring security will fail. While pairable, a pairing attempt from a peer device will be accepted.

After the BCM20706 receives this command, it reports command success or failure in the Command Status event (see "Command Status").

Table 10: Set Pairing Mode Command

Item	Description	
Operating code	0x09	
Parameters	Pairing mode (1 byte)	0: Not pairable
		1: Pairable

Unbond

The MCU can use this command to instruct the BCM20706 to remove bonding information (that is, security keys) for the device whose Bluetooth device address is passed as a parameter.

After the BCM20706 receives this command, it reports command success or failure in the Command Status event (see "Command Status").

Table 11: Unbond Command

Item	Description	
Operating code	0x0A	
Parameters	Address (6 bytes)	Bluetooth device address

User Confirmation

The MCU should send this command after it receives a User Confirmation Request event (see "User Confirmation Request") from the BCM20706 to accept or reject pairing. It is assumed that an MCU will display the numeric comparison code provided in the User Confirmation Request event and a user will provide the yes/no input that will be passed to the BCM20706 as the User Confirmation command.

Table 12: User Confirmation Command

Item	Description	
Operating code	0x0B	
Parameters	Address (6 bytes)	Bluetooth device address
	Accept/Reject (1 byte)	0: Reject pairing, or the numeric comparison code does not match.
		1: Accept pairing

Enable Coexistence

This command allows an MCU to enable the coexistence functionality in designs that include BT/BLE and WiFi applications.

Table 13: Enable Coexistence Command

Item	Description
Operating code	0x0C
Parameter	–

Disable Coexistence

This command allows an MCU to disable the coexistence functionality in designs that include BT/BLE and WiFi applications.

Table 14: Disable Coexistence Command

Item	Description
Operating code	0x0D
Parameter	–

Set Battery Level

This miscellaneous command allows the MCU to set the battery level in the GATT database of the BCM20706. A connected peer device can read the battery level using a standard ATT read operation.

Table 15: Set Battery Level Command

Item	Description
Operating code	0x0E
Parameter	Battery level (1 byte) Remaining battery capacity as a percentage (1 to 100).

Read Local Bluetooth Device Address

The MCU can send this command to read the local Bluetooth Device Address of the BCM20706. When the BCM20706 receives this command it responds with the Read Local BDA Event message containing the Bluetooth Device Address.

Table 16: Read Local Bluetooth Device Address Command

Item	Description
Operating code	0x0F
Parameters	–

Start Bond

The MCU can send this command to initiate bonding with an unbonded device.

Table 17: Start Bond Command

Item	*Description*	
Operating code	0x10	
Parameters	Address (6 bytes)	
	Transport (1 byte)	1 = BR/EDR, 2 = LE
	Address Type (1 bytes)	0 = Public, 1 = Random (LE Only)

Read Buffer Pool Usage Statistics

The MCU can send this command to read the buffer pool usage statistics to understand the buffer pool usage by the application running on the 20706, and to identify if there is a possibility of buffers running out for a given application use case. The Buffer Pool Usage Statistics event will be sent from the 20706 to the MCU which includes the buffer pool usage statistics.

Table 18: Read Buffer Pool Usage Statistics Command

Item	*Description*
Operating code	0x11
Parameters	–

LE Commands— HCI_CONTROL_GROUP_LE

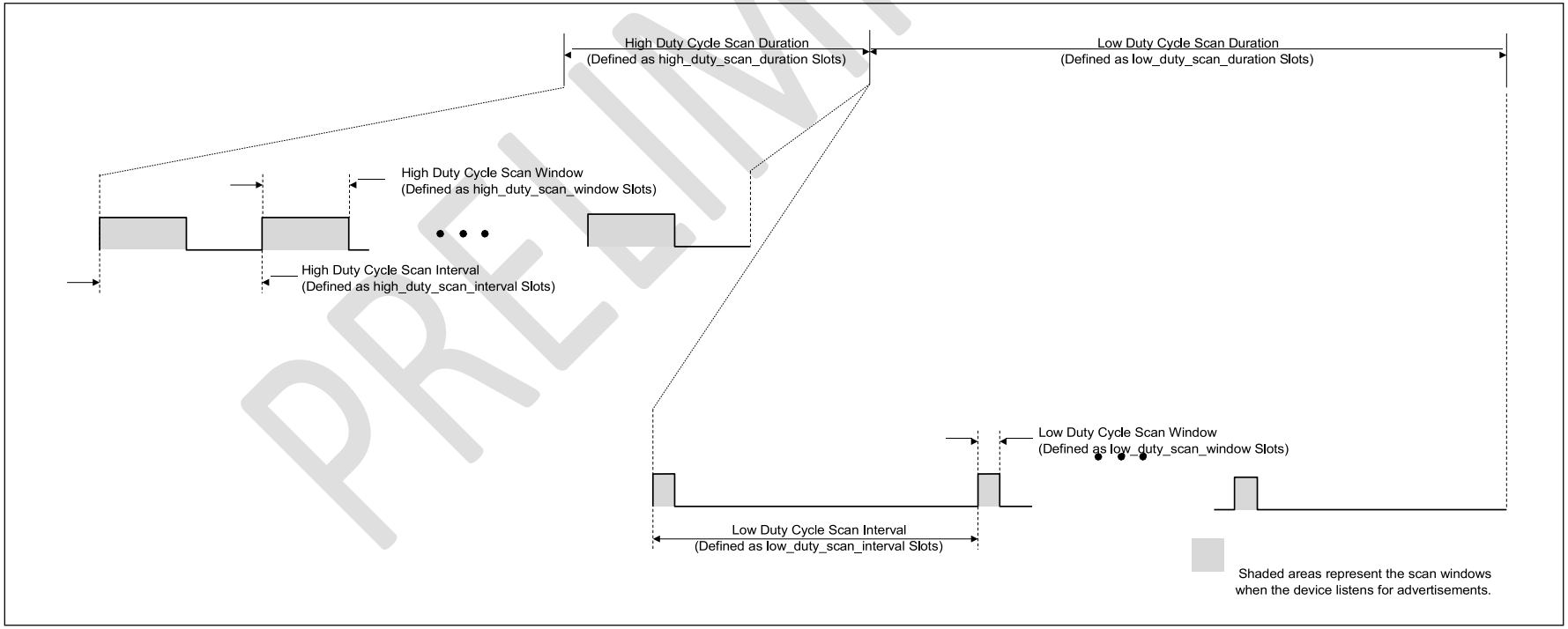
The LE commands let the MCU perform various LE Generic Access Profile (GAP) procedures using the BCM20706.

LE Scan

The LE Scan command instructs the BCM20706 to start or stop device discovery. The scan mode, window, interval, and duration are configured locally in the application running on the BCM20706 (see the wiced_bt_cfg.c file in the WICED BT SDK). When the device starts scanning, it executes a high- duty-cycle scan where it listens for advertisements during programmed windows occurring at programmed intervals for a programmed duration. Unless canceled by the application, the device then automatically switches to a low-duty-cycle scan. The device stops scanning after the low-duty-cycle scan duration.

Figure 3 shows an advertisement scanning cycle.

Figure 3: Advertisement Scanning



When the BCM20706 receives and processes this command, it reports the scan state change in the Scan Status event (see "LE Scan Status"). Scan Status events are also sent when the BCM20706 switches from a high-duty-cycle scan to a low-duty-cycle scan and from a low-duty-cycle scan to not scanning.

Table 19: LE Scan Command

Item	*Description*	
Operating code	0x01	
Parameters	Enable (1 byte)	0: Stop device-discovery scanning. 1: Start device-discovery scanning.
	Filter duplicates (1 byte)	0: Do not filter duplicate advertisements. 1: Filter duplicate advertisements.

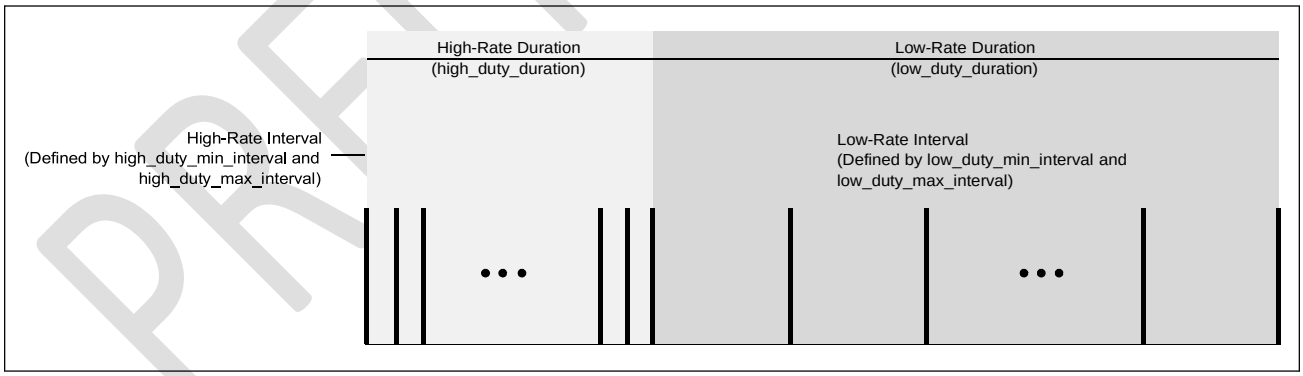
LE Advertise

The LE Advertise command instructs the BCM20706 to stop or start sending advertisements. Typically, advertisements are sent so that a central-device peer can discover and optionally connect to a peripheral-device peer. When a BCM20706 receives this command, it sends advertisements based on parameters configured in the wiced_bt_cfg_ble_advert_settings_t structure of the wiced_bt_cfg_settings_t structure (which is in the wiced_bt_cfg.c file of the WICED BT SDK).

Initially, advertisements are sent out using a programmed high-duty-cycle advertisement profile. After the high-duty-cycle duration (for example, high_duty_duration) expires, advertisements are sent out in accordance with a programmed low-duty-cycle advertisement profile, which also has a duration (for example, low_duty_duration). After the low_duty_duration, the BCM20706 stops sending advertisements.

Figure 4 shows the high-duty-cycle and low-duty-cycle advertisement-sending profiles.

Figure 4: Advertisement-Sending Profile



When the BCM20706 receives and processes this command, it reports advertisement state changes in the Advertisement State event (see "LE Advertisement State"). Advertisement State events are also sent when the BCM20706 controller switches from the high-duty-cycle advertisements to low-duty-cycle advertisements and from low-duty-cycle advertisements to no advertisements.

Table 20: LE Advertise Command

Item	Description	
Operating code	0x02	
Parameter	Enable (1 byte)	0: Disable the ability to be discovered (that is, don't send advertisements). 1: Enable the ability to be discovered (that is, send advertisements).

LE Connect

The LE Connect command instructs the BCM20706 to try establishing a connection to a specified peer device.

When the BCM20706 receives and processes this command, it reports status back in the Command Status event (see "Command Status").

When a connection is established, the BCM20706 sends a Connected event (see "LE Connected").

Table 21: LE Connect Command

Item	Description
Operating code	0x03
Parameters	Address type (1 byte)
	Device address (6 bytes)

LE Cancel Connect

The LE Cancel Connect command instructs the BCM20706 to stop a connection-establishment attempt.

When the BCM20706 receives and processes this command, it reports No Scan in the Scan Status event (see "LE Scan Status").

Table 22: LE Cancel Connect Command

Item	Description
Operating code	0x04
Parameters	–

LE Disconnect

The LE Disconnect command terminates a previously established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected"), which gets sent by the BCM20706 upon a successful connection.

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist, it reports Not Connected in the Command Status event (see "Command Status").
- If the connections exists:
  - It reports Success in the Command Status event.
  - It starts the disconnection process.
  - It reports the Disconnected event when the disconnection process finishes.

Table 23: LE Disconnect Command

Item	*Description*
Operating code	0x05
Parameter	Connection handle (2 bytes)

LE Re Pair

This command instructs the BCM20706 to delete link keys associated with a previously paired device and re-initiate a pairing sequence with that same device.

The NVRAM ID parameter should match the value reported to the MCU after the successful pairing in the NVRAM Data event (see "NVRAM Data").

Table 24: LE Re Pair Command

Item	*Description*	
Operating code	0x06	
Parameter	NVRAM ID (2 bytes)	ID associated with the address of the device from the original pairing.

LE Get Identity Address

When an initial connection with a peer is established, the MCU will receive a private random address (if a private random address is used) of the device in the LE Connection Up event message. The LE Get Identity Address command can be used by the MCU to retrieve the Identity Address, which is a public or a static random address of the peer device.

If an MCU attempts to retrieve the resolved identity address of the peer, then this command can be used. The resolved identity address of the peer will be returned in the LE Identity Address event message (see "LE Identity Address").

Table 25: LE Get Identity Address Command

Item	Description	
Operating code	0x07	
Parameter	Device address (6 bytes)	Address of the peer device.

LE Set Channel Classification

The MCU can send this command to the BCM20706 and set the channel classification for data channels. This channel classification is only applicable to connections where the BCM20706 is the master. This command contains 37 1-bit fields which correlate to the value for the link layer channel index 0 -36.

Table 26: LE Set Channel Classification Command

Item	Description	
Operating code	0x08	
Parameters	BLE Channel Map (5 bytes)	This parameter contains 37 1-bit fields for the link layer channel indexes 0 -36. Channel n is bad = 0 Channel n is unknown = 1 At least one channel should be marked as unknown.

LE Set Connection Parameters

The MCU can send this command to the BCM20706 and change the connection parameters (interval min/max, latency and timeout) of an LE link.

Table 27: LE Set Connection Parameters Command

Item	Description	
Operating code	0x09	
Parameters	Connection handle (2 bytes)	
	Connection Interval Minimum (2 bytes)	Time = N * 1.25 msec
	Connection Interval Maximum (2 bytes)	Time = N * 1.25 msec
	Slave Latency (2 bytes)	In number of connection events
	Timeout (2 bytes)	Time = N * 10 msec

GATT Commands—HCI_CONTROL_GROUP_GATT

The GATT commands let an MCU perform various Generic Attribute Profile (GATT) procedures using the BCM20706.

GATT Discover Services

The GATT Discover Services command enables service discovery over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

The hci_control application uses the Discover All Primary Services GATT procedure. The start and end handles are passed to the GATT Read By Group Type Request.

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status") .
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the discovery process.

The BCM20706 sends a GATT Service Discovered event (see "GATT Service Discovered") for each discovered service. When a peer reports that there are no more services, the GATT Discovery Complete event (see "GATT Discovery Complete") is issued. The MCU should not send any new discovery, read, or write commands until after receiving the GATT Discovery Complete event.

Table 28: GATT Discover Services Command

Item	*Description*
Operating code	0x01
Parameters	Connection handle (2 bytes)
	Start handle (2 bytes)
	End handle (2 bytes)

GATT Discover Characteristics

The GATT Discover Characteristics command enables characteristic discovery over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

The hci_control application uses the Discover All Characteristics of a service GATT procedure. The start and end handles are passed to the GATT Read By Type Request.

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the discovery process.

The BCM20706 reports a GATT Characteristic Discovered event (see "GATT Characteristic Discovered") for each discovered characteristic. When a peer reports that there are no more characteristics, the GATT Discovery Complete event (see "GATT Discovery Complete") is issued. The MCU should not send any new discovery, read, or write commands until after receiving the GATT Discovery Complete event.

Table 29: GATT Discover Characteristics Command

Item	*Description*
Operating code	0x02
Parameters	Connection handle (2 bytes)
	Start handle (2 bytes)
	End handle (2 bytes)

GATT Discover Descriptors

The GATT Discover Descriptors command enables characteristic-descriptors discovery over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

The hci_control application uses the Discover All Characteristic Descriptors GATT procedure. The start and end handles are passed to the Find Info Request.

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the discovery process.

The BCM20706 reports a GATT Descriptor Discovered event (see "GATT Descriptor Discovered") for each discovered characteristic descriptor. When a peer reports that there are no more descriptors, the BCM20706 sends a GATT Discovery Complete event (see "GATT Discovery Complete"). The MCU should not send any new discovery, read, or write commands until after receiving the GATT Discovery Complete event.

Table 30: GATT Discover Descriptors Command

Item	*Description*
Operating code	0x03
Parameters	Connection handle (2 bytes)
	Start handle (2 bytes)
	End handle (2 bytes)

GATT Command Read Request

The GATT Command Read Request command enables MCU reading of a characteristic value or a descriptor value over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the read process.

When a GATT Command Read Request is received over the UART, the hci_control application sends the Read Request for the attribute handle received in the command.

Figure 5 shows an example message sequence that takes place when one device (represented as the combination of MCU1 and BT1) requests a static attribute such as the BT device name from a second device (represented as the combination of MCU2 and BT2). In this scenario, BT2 has the attribute value and returns it.

Figure 5: Reading a Static Attribute from a Peer

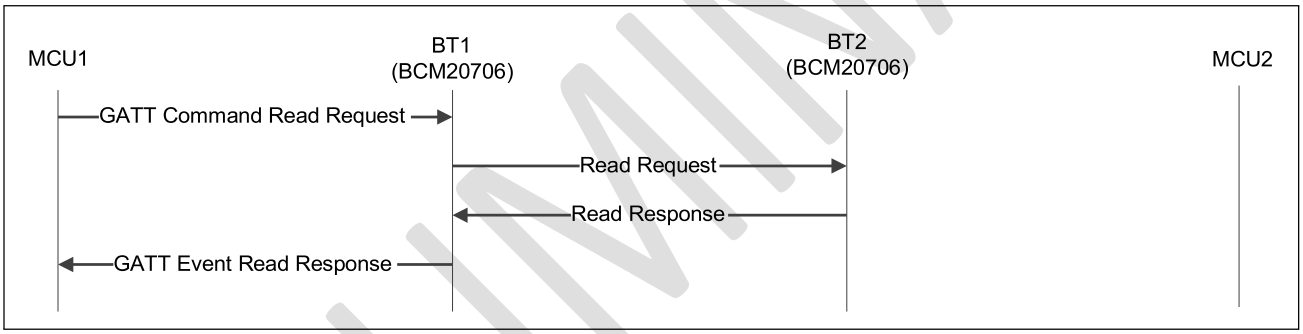
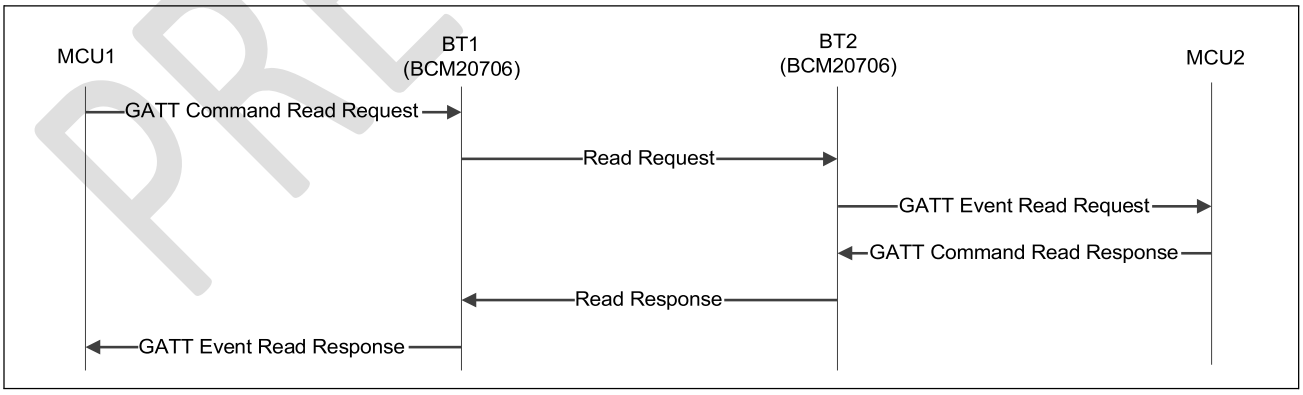


Figure 6 shows an example where BT2 must get an attribute value from MCU2.

Figure 6: Reading a Dynamic Attribute from a Peer



When a GATT Read Response or a GATT Error Response is received over the Bluetooth link, the hci_control application sends the GATT Event Read Response (see "GATT Event Read Response"). The MCU should not send any new discovery, read, or write commands until after receiving the GATT Event Read Response.

Table 31: GATT Command Read Request

Item	Description
Operating code	0x04
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)

GATT Command Read Response

The GATT Command Read Response is sent by an MCU in response to a GATT Event Read Request (see Figure 6 in GATT Command Read Request). The connection and attribute handles are the same 2- byte values that were sent in the GATT Event Read Request.

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist, then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists, then it reports Success in the Command Status event and sends the response to the connected Bluetooth device.

Table 32: GATT Command Read Response

Item	Description
Operating code	0x05
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)
	Read Status (1 byte)
	Data (variable bytes)

GATT Command Write

The GATT Command Write command enables MCU scheduling of transmissions over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the write process.

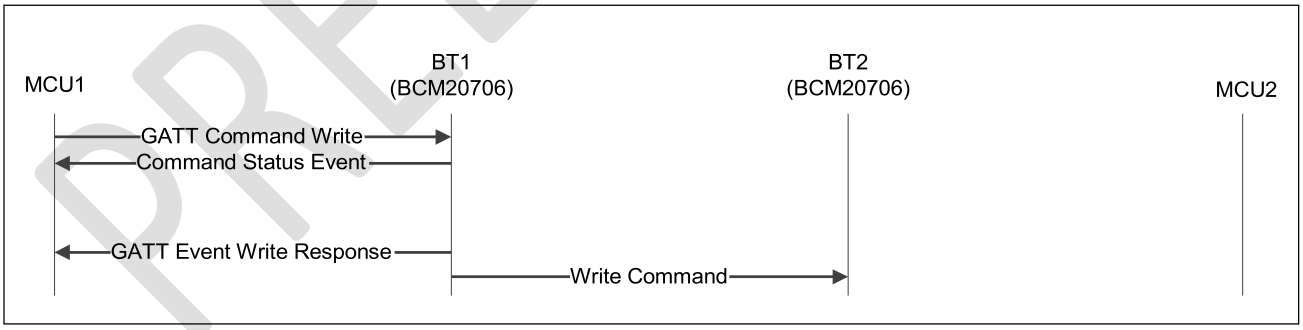
The BCM20706 has a limited number of transmit buffers. If the hci_control application is able to allocate a buffer and schedule it for transmission, then the write operation is considered complete and the hci_control application sends the GATT Event Write Response (see "GATT Event Write Response"). If all transmit buffers are already allocated and, thus, unavailable, then the hci_control application saves the data received in the command and delays sending the GATT Event Write Response until a transmit buffer becomes available and the data gets scheduled for transmission. The MCU should not send any new discovery, read, or write commands until after receiving the GATT Event Write Response.

Table 33: GATT Command Write Command

Item	*Description*
Operating code	0x06
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)
	Data (variable bytes)

Figure 7 shows an example GATT Command Write message sequence.

Figure 7: GATT Command Write message sequence



GATT Command Write Request

The GATT Command Write Request enables MCU writing of a characteristic value or a descriptor value over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the write process.

When the command is received over the UART, the hci_control application sends a GATT Write Request for the attribute handle received in the command. When a GATT Write Response or a GATT Error Response is received from a connected peer device, the hci_control application sends the GATT Event Write Response (see "GATT Event Write Response") to a connected MCU. The MCU should not send any new discovery, read, or write commands until after receiving the GATT Write Completed event.

Table 34: GATT Command Write Request

Item	*Description*
Operating code	0x07
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)
	Data (variable bytes)

Figure 8 shows a GATT Command Write Request sequence where the peer device does not require involvement from its MCU.

Figure 8: GATT Command Write Request – Peer MCU Not Involved in the Write

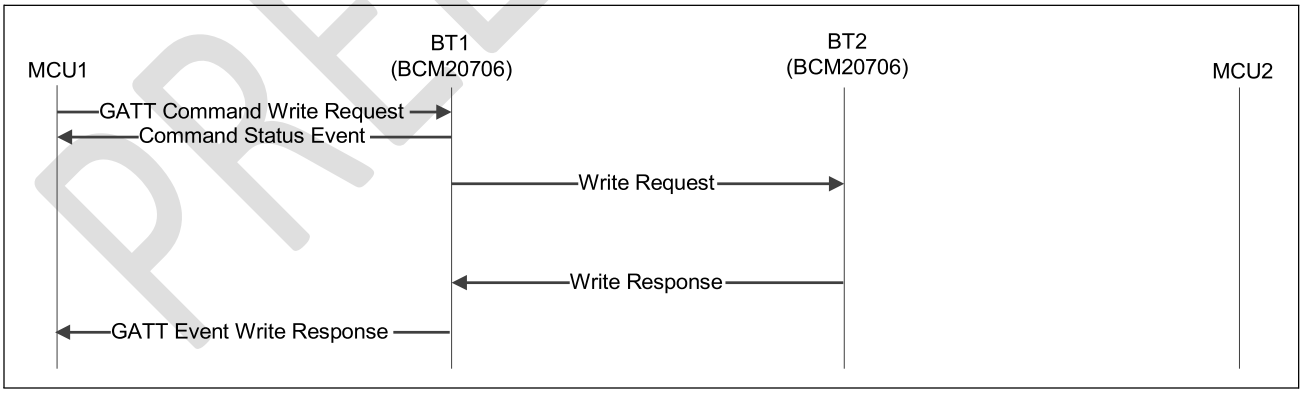
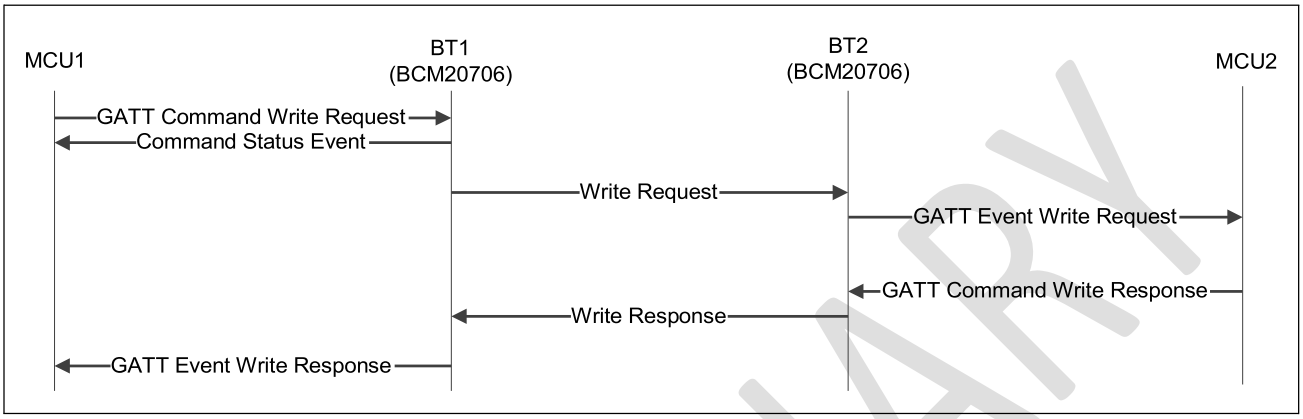


Figure 9 shows a GATT Command Write Request sequence where the peer device requires involvement from its MCU before executing the write.

Figure 9: GATT Command Write Request – MCU Is Involved in a Write



GATT Command Write Response

The GATT Command Write Response command is used to confirm a received Write Request from a peer device. The connection handle and attribute handle should match the parameters received in GATT Event Write Request (see "GATT Event Write Request"). See Figure 9 for an example message sequence where this command is used.

When the command is received over the UART, the hci_control application sends a GATT Event Write Response for the attribute handle received in the command.

Table 35: GATT Command Write Response

Item	*Description*
Operating code	0x08
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)
	Status (1 byte) **Note:** Application status codes are typically 0x80 and higher.

GATT Command Notify

The GATT Command Notify lets an MCU schedule the sending of a Notify packet over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the notification process.

The hci_control application sends a notification with the attribute handle received in the command.

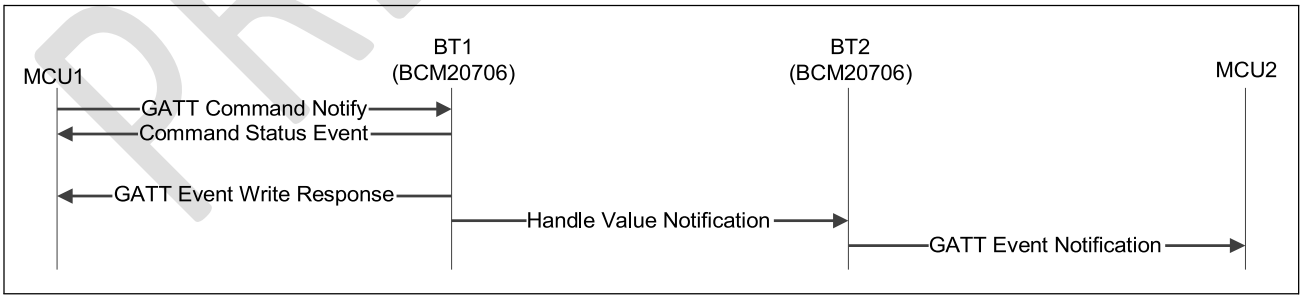
The BCM20706 has a fixed number of transmit buffers. If the hci_control application allocates a buffer and schedules it for transmission, then the GATT Command Notify operation is considered complete and the hci_control application sends the GATT Event Write Response (see "GATT Event Write Response"). If no transmit buffers are available, then the hci_control application saves the notification data and delays sending the GATT Event Write Response until a transmit buffer becomes available and the data is scheduled for transmission. The MCU should not send new discovery, read, or write commands until after receiving the GATT Event Write Response.

Table 36: GATT Command Notify

Item	*Description*
Operating code	0x09
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)
	Data (variable bytes)

Figure 10 shows a GATT Command Notify message sequence where a peer server (BT1) is prompted by its MCU (MCU1) to send a characteristic value notification to the client (BT2).

Figure 10: GATT Command Notify Message Sequence



GATT Command Indicate

The GATT Command Indicate lets an MCU perform an Value Indication procedure over an established Bluetooth LE connection. The connection handle is a two-byte value reported in the LE Connected event (see "LE Connected").

When the BCM20706 receives and processes this command, it takes one of the following actions:

- If the connection does not exist or the device is busy performing another action, such as discovery, reading, writing, etc., then it reports the relevant status in the Command Status event (see "Command Status").
- If the connection exists and the device is not busy performing another action, then it reports Success in the Command Status event and starts the indication process.

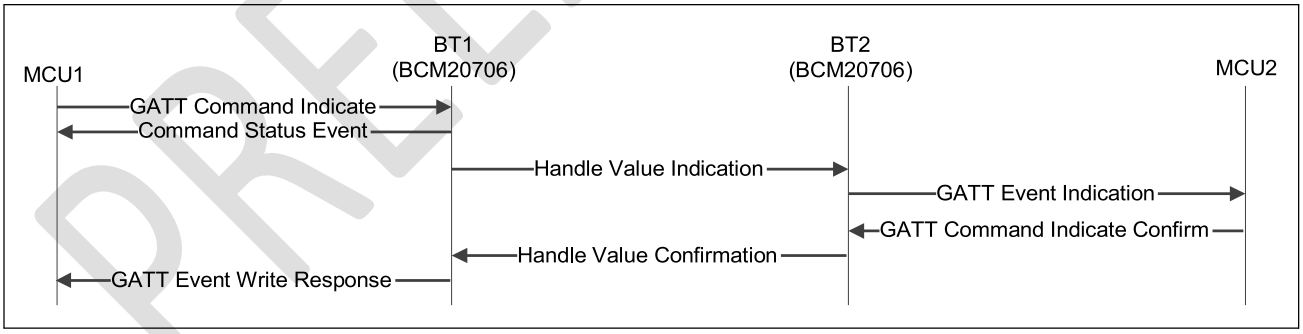
The hci_control application sends a Handle Value Indication with the attribute handle received in the command. When a Handle Value Confirmation is received from the connected device, the hci_control application sends the GATT Event Write Response (see "GATT Event Write Response"). The MCU should not send new discovery, read, or write commands until after receiving the GATT Event Write Response.

Table 37: GATT Command Indicate

Item	*Description*
Operating code	0x0A
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)
	Data

Figure 11 shows a GATT Command Indicate message sequence where a peer server (BT1) is prompted by its MCU (MCU1) to send a characteristic value indication to the client (BT2).

Figure 11: GATT Command Indicate Message Sequence



GATT Command Indicate Confirm

The GATT Command Indicate Confirm lets an MCU send a confirmation to an indication received from a peer device. The connection handle and attribute handle should match the parameters received in the GATT Event Indicate event (see "GATT Event Indication").

When the command is received over the UART, the hci_control application sends a Handle Value Confirmation (see Figure 11) for the attribute handle received in the command.

Table 38: GATT Command Indicate Confirm

Item	*Description*
Operating code	0x0B
Parameters	Connection handle (2 bytes)
	Attribute handle (2 bytes)

Hands-Free Commands— HCI_CONTROL_GROUP_HF

The Hands-Free (HF) commands let an MCU perform various HF procedures using the BCM20706.

HF Connect

The HF Connect command instructs the BCM20706 to try establishing a connection to a specified Audio Gateway (AG), which is typically a phone.

When a connection is established, the BCM20706 sends an HF Open event. The status field of that event tells whether the connection could be established or not.

Table 39: HF Connect Command

Item	*Description*
Operating code	0x01
Parameter	AG Bluetooth device address (6 bytes)

When the BCM20706 receives and processes this command, it:

- Allocates a handle for the connection.
- Starts paging the AG using the passed-in address.
- Establishes the Hands-free Profile-defined Service Level Connection (SLC) if the connection is created.
- Sends an HF Open event with the connection assigned handle and success/failure status.
- Sends an HF Connected event if the SLC gets established.

HF Disconnect

The HF Disconnect command instructs the BCM20706 to remove an existing connection to an AG.

Table 40: HF Disconnect Command

Item	*Description*
Operating code	0x02
Parameter	Connection handle (2 bytes)

When the BCM20706 receives and processes this command, it disconnects the connection identified by the passed handle. When the connection is disconnected, it sends an HF Closed event.

HF Open Audio

The HF Open Audio command instructs the BCM20706 to create an audio connection on the AG identified by the connection handle.

Table 41: HF Open Audio Command

Item	*Description*
Operating code	0x03
Parameter	Connection handle (2 bytes)

When the BCM20706 receives and processes this command, it attempts to open an audio connection on the AG identified by the passed connection handle. When an audio connection is established, it sends an HF Audio Open event.

HF Close Audio

The HF Close Audio command instructs the BCM20706 to close the audio connection on the AG identified by the connection handle.

Table 42: HF Close Audio Command

Item	*Description*
Operating code	0x04
Parameter	Connection handle (2 bytes)

When the BCM20706 receives and processes this command, it attempts to close an audio connection on the AG identified by the passed handle connection. When the audio connection is closed, it sends an HF Audio Close event.

HF Accept/Reject Audio Connection

The HF Accept/Reject Audio Connection command instructs the BCM20706 to accept/reject the SCO connection request on the AG identified by SCO index.

Table 43 HF Accept/Reject Audio Connection Command

Item	*Description*	
Operating code	0x05	
Parameter	SCO index (2 bytes)	
	Flag (1 byte)	0: Reject Audio Connection Request 1: Accept Audio Connection Request

When the BCM20706 receives and processes this command, it attempts to accept/reject the SCO connection on the AG identified by the passed handle connection.

HF AT Commands

Each HF AT Command instructs the BCM20706 to send a specific AT command to an AG.

Table 44: HF AT Command

Item	*Description*
Operating code	See Table 45.
Parameters	Connection handle (2 bytes)
	Command code (1 byte)
	Numeric value (2 bytes)
	Optional supporting character string (variable bytes)

Table 45 shows various available settings for the command code, numeric value, and optional string parameters of the HF AT Command (see Table 44).

Table 45: HF AT Command Parameters

Command Code			
Code	*Description*	*Numeric Value*	*Optional String*
0x20	Speaker gain	0–15	–
0x21	Microphone gain	0–15	–
0x22	Answer incoming call	–	–
0x23	Get number from voice tag	1	–
0x24	Voice recognition	0: Disable 1: Enable	–
0x25	Last number redial	–	–
0x26	Call hold	0: Release all held calls 1: Release all active calls 2: Swap active and held calls 3: Hold active call	–
0x27	Hang up	–	–
0x28	Read indicator status	–	–
0x29	Retrieve subscriber number	–	–
0x2A	Dial	–	Number to dial terminated by a semicolon. For example, “nnnnnnnnn;”
0x2B	Noise/Echo control	0: Disable 1: Enable	–
0x2C	Transmit DTMF tone	–	Digits to transmit via DTMF. For example, “nnnn#*”
0x2D	Response and hold	0: Hold incoming call 1: Accept held incoming call 2: Reject held incoming call	–
0x2E	Get operator information	–	–
0x2F	Extended result codes	1: Enable	–

Table 45: HF AT Command Parameters (Cont.)

Command Code			
Code	***Description***	***Numeric Value***	***Optional String***
0x30	Get current call list	–	–
0x31	Indicator control	–	1's (enable) and 0's (disable) corresponding to each indicator. For example, "0,1,1,0,0,0,0"
0x32	Send HF indicator	–	The indicator and value. For example, "2,1"
0x33	Send proprietary AT command	–	The full AT command string

When the BCM20706 receives and processes this command, it attempts to send the corresponding AT command to the AG identified by the connection handle. When a response is received from the AG, it is sent back via an HF Response event (see "HF Response"). Another command should not be sent until after the response event is received.

Serial Port Profile Commands—HCI_CONTROL_GROUP_SPP

The Serial Port Profile (SPP) commands let an MCU establish an SPP connection to a peer and send data.

SPP Connect

The MCU can send an SPP Connect command to the BCM20706 to establish an SPP connection to a specified device. Upon receiving the command, the BCM20706 establishes an ACL data connection, performs a Service Discovery Protocol (SDP) search for the RFCOMM service, and establishes an RFCOMM connection to that service.

If the operation is successful, the BCM20706 will send the SPP Connected event back to the MCU. If the operation fails, the SPP Connection Failed or SPP Service Not Found event is sent.

Table 46: SPP Connect Command

Item	*Description*
Operating code	0x01
Parameter	Bluetooth device address of the peer device (6 bytes)

SPP Disconnect

The MCU can send an SPP Disconnect command to disconnect a previously established SPP connection.

Table 47: SPP Disconnect Command

Item	*Description*	
Operating code	0x02	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the SPP Connected event.

SPP Data

The MCU issues the SPP Data command to send data over an established SPP connection.

Upon receiving an SPP Data command, the BCM20706 attempts to allocate a buffer and queue a data packet for transmission. After the packet is enqueued, the BCM20706 sends the TX Completed event. If the queue is full because data is received over the UART faster than it can be delivered to the peer, then the TX Completed event is delayed until the operation can be completed.

Table 48: SPP Data Command

Item	*Description*	
Operating code	0x03	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the SPP Connected event.
	Data (variable bytes)	–

Audio Commands— HCI_CONTROL_GROUP_AUDIO

The audio commands let an MCU establish an AV source connection to a peer device over the AVDT protocol and then send data.

Audio Connect

The MCU can send an Audio Connect command to the BCM20706 to establish an AV Source connection to a specified device. Upon receiving the command, the BCM20706 establishes an ACL data connection, performs Service Discovery Protocol (SDP) searches for the A2DP service, and establishes an AVDTP signaling connection and the data channel.

If the operation succeeds, the BCM20706 will send the Audio Connected event back to the MCU.

If the operation fails, the Audio Connection Failed, or Audio Service Not Found event is sent.

Table 49: Audio Connect Command

Item	*Description*	
Operating code	0x01	
Parameter	Address (6 bytes)	Bluetooth device address of the peer device.
	Audio route (1 byte)	0: I^2S
		1: UART
		2: Sine (sends a sine wave)

Audio Disconnect

The MCU can send an Audio Disconnect command to disconnect a previously established AV source connection.

Table 50: Audio Disconnect Command

Item	*Description*	
Operating code	0x02	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event.

Audio Start

The MCU can send an Audio Start command to the BCM20706 to start streaming audio from the MCU to the remote device. Upon receiving the command, if the BCM20706 determines that it's appropriate and necessary, it reconfigures the channel for a new sampling frequency and/or channel mode. If successful, it begins requesting raw audio data from the MCU.

The MCU can send an Audio Start command only after an audio connection to the peer device has been established; that is, after an Audio Connected event has been received (see "Audio Connected").

If the MCU was previously streaming data and it issued the Audio Stop command (see "Audio Stop"), it should not send another Audio Start command until after it receives the Audio Stopped event (see "Audio Stopped").

Sending the Audio Start command configures the BCM20706 for specific stream settings, including sample frequency and channel mode. Configured parameters will persist across stream suspend and resume.

If the peer device disconnects and then reconnects (see "Audio Connected" and "Audio Disconnected"), the BCM20706 will not start streaming until the MCU resends the Audio Start command.

If the operation is successful, then the BCM20706 will send the Audio Started event (see "Audio Started") back to the MCU. If the operation fails, then the Audio Stopped event (see "Audio Stopped") will be sent.

Table 51: Audio Start Command

Item	*Description*	
Operating code	0x03	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event.
	Sampling frequency (1 byte)	0: 16 kHz
		1: 32 kHz
		2: 44.1 kHz
		3: 48 kHz.
	Channel mode (1 byte)	0: Mono
		1: Stereo

Audio Stop

The MCU can send an Audio Stop command to the BCM20706 to stop streaming audio from the MCU, through the platform, to the remote device. Upon receiving the command, the BCM20706 stops requesting audio data buffers from the MCU. When the BCM20706 finishes sending queued data, it will send the Audio Stopped event (see "Audio Stopped") to the MCU and, upon timeout (if not restarted), it will place the AVDTP connection in a suspended state.

Table 52: Audio Stop Command

Item	*Description*	
Operating code	0x04	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event.

After sending an Audio Stop command, an MCU should not send an Audio Start command until after it receives an Audio Stop event.

Audio Data

The MCU can send an Audio Data command in response to an Audio Data Request event (see "Audio Data Request"). The Audio Data Request indicates the bytes per packet and number of packets that the MCU needs to send.

The Audio Data command from the MCU carries high-priority, real-time data. The type of raw PCM data (stereo/mono, sampling frequency) is set by the MCU in the Audio Start Command (see "Audio Start").

Table 53: Audio Data Command

Item	*Description*	
Operating code	0x06	
Parameters	PCM data packet length (2 bytes)	–
	PCM data (variable bytes)	Each 16-bit audio sample is 2 bytes.

HID Device Commands—HCI_CONTROL_GROUP_HIDD

The HID Device (HIDD) commands let an MCU perform various HIDD-related procedures using the BCM20706.

HID Accept Pairing

The HID Accept Pairing command instructs the BCM20706 to enter or exit a discoverable and connectable mode. When the BCM20706 is in a discoverable and connectable mode, peer devices can find the device and establish a bonding relationship with it.

Table 54: HID Accept Pairing Command

Item	*Description*
Operating code	0x01
Parameter	Enable (1 byte)

When a peer device establishes a connection, the HID Opened event (see "HID Opened") will be sent to the MCU. At that time, the MCU can start sending HID reports.

HID Send Report

When a connection is established, the MCU can send a HID report over the HID interrupt or control channel. The report should be a fully formatted packet, including the Report ID and the data.

Table 55: HID Send Report Command

Item	*Description*	
Operating code	0x02	
Parameters	Report channel (1 byte)	0: Control 1: Interrupt
	Report type (1 byte)	0: Other 1: Input 2: Output 3: Feature
	Report data (variable bytes)	

If the BCM20706 is not connected to a paired host when it receives a HID Send Report command, it will try to establish a HID connection. When this happens, the report will be lost.

HID Push Pairing Host Info

If the BCM20706 is not connected to external serial flash, then the MCU is responsible for storing the paired host information. At start-up, the MCU should download the paired host information that it previously received in an NVRAM Data event (see "NVRAM Data").

Table 56: HID Push Pairing Host Info Command

Item	Description	
Operating code	0x03	
Parameter	Data (variable bytes)	Data received in the NVRAM Data event.

HID Connect

The HID Connect command instructs the BCM20706 to try establishing a connection to a previously paired HID host. Prior to issuing this command, information about the host, including the Bluetooth device address and link key, should be downloaded to the BCM20706 using the HID Push Pairing Host Info command.

When a connection is established, the BCM20706 sends a HID Opened event (see "HID Opened").

Table 57: HID Connect Command

Item	Description	
Operating code	0x04	
Parameter	Address (6 bytes)	Bluetooth device address of the HID host to which a connection is made.

AV Remote Control Target Commands— HCI_CONTROL_GROUP_AVRC_TARGET

AVRC Target Connect

Note: This command should only be used in the case of PTS testing. Target side connections are made in conjunction with the Audio Source connections.

The MCU can send this to the BCM20706 to establish an AV remote control target connection to a specified device. Upon receiving this command, the BCM20706 establishes an ACL data connection if one does not exist yet, performs the Service Discovery Protocol (SDP), searches for the AVRC service, and establishes an AVCTP channel.

If the operation succeeds, the BCM20706 sends the AVRC Connected event (see "AVRC Controller Connected") back to the MCU. If the operation fails, the BCM20706 sends the AVRC Disconnected event (see "AVRC Controller Disconnected").

Table 58: AVRC Target Connect Command

Item	*Description*	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth address of the peer device.

AVRC Target Disconnect

Note: This command should only be used in the case of PTS testing. Target side connections are made in conjunction with the Audio Source connections.

The MCU can send this command to disconnect a previously established AV remote control connection.

Table 59: AVRC Target Disconnect Command

Item	*Description*
Operating code	0x02
Parameters	-

AVRC Target Track Information

The MCU can send this command to the BCM20706 to inform it of updates to the track information for the currently playing track. The MCU shall send information about all changed attributes in a single command. The command can include all attributes or it can limited to one or several attributes. For example one could send Title and Track Number if only those attributes have changed. If an attribute is not available for the new track, the MCU should include the attribute with length of zero.

Table 60: AVRC Target Track Information Command

Item	*Description*	
Operating code	0x05	
Parameter	Attribute Id of the first attribute (1 byte)	1: Title 2: Artist 3: Album 4: Track number 5: Number of tracks 6: Genre 7: Playing time
	Attribute Length of the first attribute (1 byte)	Attribute string Length
	Attribute Value of the first attribute (n bytes as defined above)	Attribute value expressed as a character string.
	Attribute Id of the second attribute (1 byte)	(see list above)
	Attribute Length of the second attribute (1 byte)	Attribute string Length
	Attribute Value of the second attribute (n bytes as defined above)	Attribute value expressed as a character string.
	… (up to 7 entries)	…

AVRC Target Player Status

The MCU can send this command to the BCM20706 with the player play state and track position information. It is mandatory for the MCU to send this status update for every change in the playback status of the local player If last reported status is *playing*, the BCM20706 will assume that the track position on the player is continuously updating 1000 ms every second. The MCU must send the Track Position only if changes are not due to the standard playback. For example, the command needs to be sent regularly if the player is performing fast forward or rewind operations, or if the position jumps due to the local update on the player.

Table 61: AVRC Target Player Status Command

Item	*Description*	
Operating code	0x06	
Parameter	Play State	0x00: STOPPED 0x01: PLAYING 0x02: PAUSED 0x03: FWD_SEEK 0x04: REV_SEEK
	Track Length (4 bytes)	Length of the current track in milliseconds
	Track Position (4 bytes)	Position in the current track in ms within Length defined above.

AVRC Target Repeat Mode Changed

The MCU can send this command to the BCM20706 to inform the BCM20706 of a change in the mode of the local player repeat setting.

Table 62: AVRC Target Repeat Mode Changed Command

Item	*Description*	
Operating code	0x07	
Parameter	Repeat Mode	0x01: OFF 0x02: Single Track Repeat 0x03: All Track Repeat 0x04: Group Repeat

AVRC Target Shuffle Mode Changed

The MCU can send this command to the BCM20706 to inform the BCM20706 of a change in the mode of the local player shuffle setting.

Table 63: AVRC Target Shuffle Mode Changed Command

Item	Description	
Operating code	0x08	
Parameter	Shuffle Mode	0x01: OFF 0x02: All Track Shuffle 0x04: Group Shuffle

AVRC Target Equalizer Status Changed

The MCU can send this command to the BCM20706 to inform the BCM20706 of a toggle in the On/Off status of the local player equalizer.

Table 64: AVRC Target Equalizer Status Changed Command

Item	Description	
Operating code	0x09	
Parameter	Equalizer status	0x01: OFF 0x02: On

AVRC Target Scan Mode Changed

The MCU can send this command to the BCM20706 to reflect the change of the status of the local player scan control setting.

Table 65: AVRC Target Scan Mode Changed Command

Item	Description	
Operating code	0x0A	
Parameter	Scan Mode	0x01: OFF 0x02: All Track Scan 0x04: Group Scan

AV Remote Control Controller Commands — HCI_CONTROL_GROUP_AVRC_CONTROLLER

The AV Remote Control controller group of the commands are used by the MCU when implementing a remote control application. For example the MCU can send play, pause and other commands to the remote connected Bluetooth player.

AVRC Controller Connect

The MCU can send this to the BCM20706 to establish an AV remote control connection to a specified device. Upon receiving this command, the BCM20706 establishes an ACL data connection if one does not exist yet, performs the Service Discovery Protocol (SDP), searches for the AVRC service, and establishes an AVCTP channel.

If the operation succeeds, the BCM20706 sends the AVRC Connected event back to the MCU. If the operation fails, the BCM20706 sends the AVRC Disconnected event.

Note: This command should only be used in the case of a standalone AVRC Controller application. If remote controller functionality is combined with the speaker, the AVRC command will be established automatically when audio connection is established.

Table 66: AVRC Controller Connect Command

Item	*Description*	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth address of the peer device

AVRC Controller Disconnect

The MCU can send this command to disconnect a previously established AV remote control connection.

Table 67: AVRC Controller Disconnect Command

Item	*Description*	
Operating code	0x02	
Parameters	bd_addr (6 bytes)	Bluetooth address of the peer device.

AVRC Controller Play

The MCU sends this command to start playing audio on the connected Bluetooth media player.

Table 68: AVRC Controller Play Command

Item	Description	
Operating code	0x03	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Stop

The MCU sends this command to stop playing audio on the connected Bluetooth media player.

Table 69: AVRC Controller Stop Command

Item	Description	
Operating code	0x04	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Pause

The MCU sends this command to pause playing audio on the connected Bluetooth media player.

Table 70: AVRC Controller Pause Command

Item	Description	
Operating code	0x05	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Begin Fast Forward

The MCU sends this command to begin fast forward operation on the connected Bluetooth media player. Unlike most of the other AVRC commands, this command initiates the mode where the player plays audio at high speed. Use the AVRC End Fast Forward command to terminate this mode.

Table 71: AVRC Controller Begin Fast Forward Command

Item	Description	
Operating code	0x06	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller End Fast Forward

The MCU sends this command to terminate fast forward operation on the connected Bluetooth media player.

Table 72: AVRC Controller End Fast Forward Command

Item	Description	
Operating code	0x07	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Begin Rewind

The MCU sends this command to begin rewind operation on the connected Bluetooth media player. Unlike most of the other AVRC commands, this command initiates the mode where the player plays audio in reverse at high speed. Use the AVRC End Rewind command to terminate this mode.

Table 73: AVRC Controller Begin Rewind Command

Item	Description	
Operating code	0x08	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller End Rewind

The MCU sends this command to terminate rewind operation on the connected Bluetooth media player.

Table 74: AVRC Controller End Rewind Command

Item	Description	
Operating code	0x09	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Next Track

The MCU sends this command to instruct the player to move to the next track on the connected Bluetooth media player.

Table 75: AVRC Controller Next Track Command

Item	Description	
Operating code	0x0A	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Previous Track

The MCU sends this command to instruct the player to move to the previous track on the connected Bluetooth media player.

Table 76: AVRC Controller Previous Track Command

Item	Description	
Operating code	0x0B	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Volume Up

The MCU can send this command to the BCM20706 to request a volume increase on a connected AV player.

Table 77: AVRC Controller Volume Up Command

Item	Description	
Operating code	0x0C	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Volume Down

The MCU can send this command to the BCM20706 to request a volume decrease on a connected AV player.

Table 78: AVRC Controller Volume Down Command

Item	Description	
Operating code	0x0D	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Get Track Information

This is an optional command that an MCU can send to a BCM20706 in order to retrieve the current track information from the target player. The BCM20706 sends a request for the current track attributes to the peer. When the player responds the BCM20706 will send an event to the MCU for each of the track elements that it has retrieved. This can be invoked at any time or the MCU can choose to do so when informed by the BCM20706 of a track change (see “AVRC Controller Track Change”).

Table 79: AVRC Controller Get Track Information Command

Item	*Description*	
Operating code	0x0E	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event.
	Number of attributes (1 byte)	Number of attributes to return. 0 to return all attributes.
	Attributes (1 to 8 bytes)	Each byte represents an attribute to retrieve. Attribute values indicate the following options: 1: Title 2: Artist 3: Album 4: Track number 5: Number of tracks 6: Genre 7: Playing time

AVRC Controller Set Equalizer Status

The MCU can send this command to the BCM20706 to toggle the on/off state of the target player equalizer. The BCM20706 reports the initial state of the equalizer and subsequent state changes using the AVRC Setting Change event (see "AVRC Controller Setting Change").

Table 80: AVRC Controller Set Equalizer Status Command

Item	Description	
Operating code	0x0F	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event (see "Audio Connected").

AVRC Controller Set Repeat Mode

The MCU can send this command to the BCM20706 to change the repeat mode of the target player. Each command submitted by the MCU will change the setting to the next available on the remote, cycling through all possible settings one at a time. The BCM20706 reports the initial repeat-mode state and subsequent state changes using the AVRC Setting Change event (see "AVRC Controller Setting Change").

Table 81: AVRC Controller Set Repeat Mode Command

Item	Description	
Operating code	0x10	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event (see "Audio Connected").

AVRC Controller Set Shuffle Mode

The MCU can send this command to the BCM20706 to change the shuffle mode of the target player. Each command submitted by the MCU will change the setting on the remote to the next available setting, cycling through all possible settings one at a time. The BCM20706 reports the initial shuffle-mode state and subsequent state changes using the AVRC Setting Change event (see "AVRC Controller Setting Change").

Table 82: AVRC Controller Set Shuffle Mode Command

Item	Description	
Operating code	0x11	
Parameter	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event (see "Audio Connected").

AVRC Controller Set Scan Status

The MCU can send this command to the BCM20706 to change the scan status of the target player. Each command submitted by the MCU will change the setting on the remote to the next available setting, cycling through all possible settings one at a time. The BCM20706 reports the initial scan status and subsequent status changes in the AVRC Setting Change event (see "AVRC Controller Setting Change").

Table 83: AVRC Controller Set Scan Status Command

Item	Description	
Operating code	0x12	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event (see "Audio Connected").

AVRC Controller Set Volume

The MCU can send this command to the BCM20706 to pass a new volume setting to the connected AV sink device. An MCU should use this command only if the *Absolute Volume Capable* flag is true as indicated in the Audio Connected event (see "Audio Connected" on page 102).

Table 84: AVRC Controller Set Volume Command

Item	Description	
Operating code	0x13	
Parameters	Connection handle (2 bytes)	The connection handle as reported in the Audio Connected event (see "Audio Connected").
	Volume level (1 byte)	The percentage (0 to 100) of the maximum volume level to be used by a connected peer device.

Test Commands— HCI_CONTROL_GROUP_TEST

The Test commands allow the host to execute various tests on the BCM20706.

Encapsulated HCI Command

Primarily for manufacturing test purposes, this test command allows the host to send encapsulated HCI commands to the BCM20706 to control the BT controller for RF test purposes. For example, Bluetooth LE RF testing usually requires the support of the LE Transmitter Test, LE Receiver Test, and LE Test End HCI commands (see BLUETOOTH SPECIFICATION Version 4.2 [Vol 2, Part E], Section 7.8.28, 7.8.29, and 7.8.30 [2] for details). All of which can be formatted into this Encapsulated HCI Command.

The BCM20706 also provides support for Vendor Specific commands (Radio_Tx_Test and Radio_Rx_Test) which enable a connectionless transmit and receive mode to send and receive respectively Bluetooth packets at a specified frequency. Please refer to the WMBT tool included in the SDK under Tools\wmbt.

When the BCM20706 receives a test command, it is put into a Test Mode. While in the Test Mode all the events received from the controller are passed to the MCU as Encapsulated HCI Events (see Encapsulated HCI Event) and not processed by the stack. Because of that the BCM20706 must be reset and reinitialized to continue normal application operation.

Table 85: Encapsulated HCI Command

Item	*Description*	
Operating code	0x10	
Parameters	HCI Command (variable bytes)	Fully formatted HCI Command.

ANCS Commands— HCI_CONTROL_GROUP_ANCS

The Apple Notification Control Service (ANCS) commands let an MCU perform various ANCS-related procedures using the BCM20706. Refer to the following Apple developer link for more information:

https://developer.apple.com/library/ios/documentation/CoreBluetooth/Reference/AppleNotificationCenterServiceSpecification/Specification/Specification.html

ANCS Action

This command instructs the BCM20706 to pass a positive or negative action with respect to a specific notification sent by the iOS device. The command is sent after the BCM20706 reports the notification in the ANCS Notification event (see "ANCS Notification").

Table 86: ANCS Action Command

Item	Description	
Operating code	0x01	
Parameters	Notification ID (4 bytes)	The Notification ID as reported in the ANCS Notification Event.
	Action (1 byte)	0: Positive action. 1: Negative action.

ANCS Connect

This command instructs the BCM20706 to activate the ANCS service on the iOS device connected to the given LE Connection Handle. The MCU should not send this command until after it has received the ANCS Service Found event and has verified that the LE connection is Encrypted since the ANCS service requires Authentication.

Table 87: ANCS Connect Command

Item	Description	
Operating code	0x02	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.

ANCS Disconnect

This command instructs the BCM20706 to deactivate the ANCS service on the iOS device connected to the given LE Connection Handle by unsubscribing to notifications for the service.

Table 88: ANCS Disconnect Command

Item	*Description*	
Operating code	0x03	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.

AMS Commands— HCI_CONTROL_GROUP_AMS

The Apple Media Service (AMS) commands let an MCU perform various AMS-related procedures using the BCM20706. Refer to the following Apple developer link for more information:

https://developer.apple.com/library/ios/documentation/CoreBluetooth/Reference/AppleMediaService_Reference/Specification/Specification.html

AMS Connect

This command instructs the BCM20706 to activate the AMS service on the iOS device connected to the given LE Connection Handle. The MCU should not send this command until after it has received the AMS Service Found event and has verified that the LE connection is Encrypted since the AMS service requires Authentication.

Table 89: AMS Connect Command

Item	*Description*	
Operating code	0x01	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.

AMS Disconnect

This command instructs the BCM20706 to deactivate the AMS service on the iOS device connected to the given LE Connection Handle by unsubscribing to notifications for the service.

Table 90: AMS Disconnect Command

Item	*Description*	
Operating code	0x02	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.

iAP2 Commands—HCI_CONTROL_GROUP_IAP2

The Apple iPod Accessory Protocol (iAP2) commands allows an MCU to establish and send data over an iAP2 External Accessory (EA) session implemented on a BCM20706.

IAP2 Connect

The MCU can send this command to the BCM20706 to establish an EA session with a specified device. Upon receiving the command, the BCM20706 establishes an ACL data connection, performs an SDP search for the iAP2 service, and establishes an EA session to the iAP2 service.

After the EA session is successfully established, the BCM20706 will send an IAP2 Connected event (see "IAP2 Connected") back to the MCU. If the operation fails, then either the IAP2 Connection Failed event (see "IAP2 Connection Failed") or IAP2 Service Not Found event (see "IAP2 Service Not Found") is sent.

Table 91: IAP2 Connect Command

Item	Description	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth address of the peer device.

IAP2 Disconnect

The MCU can send this command to disconnect a previously established EA session.

Table 92: IAP2 Disconnect Command

Item	Description	
Operating code	0x02	
Parameters	Session handle (2 bytes)	The session handle as reported in the IAP2 Connected event (see "IAP2 Connected").

IAP2 Data

An MCU issues this command to send data over an established EA session.

Upon receiving this command, the BCM20706 attempts to allocate a buffer and queue a data packet for transmission. After successfully enqueuing a packet, the BCM20706 sends the IAP2 TX Complete event (see "IAP2 TX Complete"). If the queue is full because data is received over the UART faster than it can be delivered to the peer, then the sending of the TX Completed event is delayed until after the packet is successfully enqueued.

Table 93: IAP2 Data Command

Item	*Description*	
Operating code	0x03	
Parameters	Session handle (2 bytes)	The session handle as reported in the IAP2 Connected event (see "IAP2 Connected").
	Data (variable bytes)	Data to be transmitted to the iOS device.

IAP2 Get Auth Chip Info

The MCU can send this command to read the chip information from the authentication coprocessor connected to the BCM20706.

Table 94: IAP2 Get Auth Chip Info Command

Item	*Description*
Operating code	0x04
Parameters	–

Hands-free AG Commands—HCI_CONTROL_GROUP_AG

The Hands-free AG commands let an MCU establish signaling and audio connections to a peer hands-free device. The current version of the protocol defined in this document supports a simple implementation that can be used only for voice control and not for actual calls, conferences, and more.

AG Connect

An MCU can send this command to the BCM20706 to establish an hands-free audio gateway connection to a specified device. Upon receiving the command, the BCM20706 establishes an ACL data connection, performs an SDP search for the RFCOMM service, establishes a connection with the RFCOMM service, and establishes a signaling connection with the specified hands-free device.

After an AG connection is successfully established, the BCM20706 will send the AG Connected event (see “AG Connected") back to the MCU.

Table 95: AG Connect Command

Item	*Description*	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth address of the peer device.

AG Disconnect

An MCU can send this command to disconnect a previously established AG signaling connection.

Table 96: AG Disconnect Command

Item	*Description*	
Operating code	0x02	
Parameters	Session handle (2 bytes)	The session handle as reported in the AG Connected event (see “AG Connected").

AG Audio Connect

An MCU can send this command to establish an audio channel over a previously established AG signaling connection.

Table 97: AG Audio Connect Command

Item	*Description*	
Operating code	0x03	
Parameters	Session handle (2 bytes)	The session handle as reported in the AG Connected event (see "AG Connected").

AG Audio Disconnect

An MCU can send this command to disconnect the audio channel previously established over the AG signaling connection.

Table 98: AG Audio Disconnect Command

Item	*Description*	
Operating code	0x04	
Parameters	Session handle (2 bytes)	The session handle as reported in the AG Connected event (see "AG Connected").

AIO Server Commands—HCI_CONTROL_GROUP_AIO_SERVER

The Automation IO (AIO) server commands let an MCU perform various AIO server procedures using the BCM20706.

AIO Digital Input

This command allows an MCU to simulate a change in an AIO server digital input.

Table 99: AIO Digital Input Command

Item	*Description*	
Operating code	0x01	
Parameters	Index (1 byte)	Index of digital IO, starting with 0.
	Data (variable bytes)	An array of 2-bit values in a bit field in little endian order, which identifies the state of the digital input. 00: Inactive state 01: Active state 10: Tristate 11: Unknown state

After a BCM20706 receives this command, it sets the new value in the database and, if a value/time trigger is set and the condition is met, sends a notification or indication with the new value to the AIO client.

AIO Analog Input

This command allows an MCU to indicate a change in an AIO server analog input value.

Table 100: AIO Analog Input Command

Item	*Description*	
Operating code	0x02	
Parameters	Index (1 byte)	Index of analog IO, starting with 0.
	Data (2 bytes)	The value of the analog signal as an unsigned 16-bit integer.

After a BCM20706 receives this command, it sets the new value in the database and, if a value/time trigger is set and the condition is met, sends a notification or indication with the new value to the AIO client.

AIO Client Commands - HCI_CONTROL_GROUP_AIO_CLIENT

The Automation IO Client commands let an MCU perform various AIO client procedures using the BCM20706.

AIO Connect

This command instructs the AIO client on a BCM20706 to connect to an AIO server.

Table 101: AIO Connect Command

Item	*Description*	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth address of the AIO server to which a connection is made.

After the BCM20706 receives this command, it tries to establish a connection to the specified AIO server. If a Bluetooth device address is not specified or set to all zeros, it starts LE scanning and connects to the first AIO server it finds. After a connection is established, the BCM20706 performs characteristic and characteristic descriptor discoveries.

AIO Read

This command instructs the AIO client on a BCM20706 to read a value from the AIO server.

Table 102: AIO Read Command

Item	*Description*	
Operating code	0x02	
Parameters	Type (1 byte)	1: Analog IO 2: Digital IO 3: Aggregate IO
	Index (1 byte)	Index of the analog, digital, or aggregate IO, starting with 0.

After a BCM20706 receives this command, it sends a read request to the AIO server. After a read response comes back from the AIO server, the BCM20706 will send the value back to the MCU in an AIO Read Response event (see “AIO Read Response”).

AIO Write

This command instructs the AIO client on a BCM20706 to write a value to the AIO server.

Table 103: AIO Write Command

Item	*Description*	
Operating code	0x03	
Parameters	Type (1 byte)	1: Analog IO 2: Digital IO
	Index (1 byte)	Index of the analog or digital IO, starting with 0.
	Data (variable bytes)	An unsigned 16-bit integer for analog IO, or an array of 2-bit values in a bit field for digital IO.

After the BCM20706 receives this command, it sends a write request to the AIO server.

AIO Register for Notification

This command instructs the AIO client on a BCM20706 to register for notification or indication on the AIO server.

Table 104: AIO Register for Notification Command

Item	*Description*	
Operating code	0x04	
Parameters	Type (1 byte)	1: Analog IO 2: Digital IO 3: Aggregate IO
	Index (1 byte)	Index of the analog, digital, or aggregate IO, starting with 0.
	Value (1 byte)	0: Unregister notification/indication 1: Register for notification 2: Register for indication

After a BCM20706 receives this command, it sends a write request to the AIO server to set a client characteristic configuration descriptor. The notification and/or indication configuration is set through a combination of the client characteristic configuration descriptor and the value and/or time trigger settings. See “AIO Set Value Trigger” and “AIO Set Time Trigger” for information regarding setting value and time triggers.

AIO Set Value Trigger

This command instructs the AIO client on a BCM20706 to set a value trigger on the AIO server.

Table 105: AIO Set Value Trigger Command

Item	*Description*	
Operating code	0x06	
Parameters	Type (1 byte)	1: Analog IO 2: Digital IO
	Index (1 byte)	Index of the analog or digital IO, starting with 0.
	Condition (1 byte)	0: Value changed 1: Crossed boundary 2: On the boundary 3: Value change exceeds a set value 4: Mask then compare 5: Crossed boundaries 6: On the boundaries 7: No value trigger
	Values(variable bytes)	These bytes are a function of the condition set. They represent one or more boundaries or a set value.

After a BCM20706 receives this command, it sends a write request to an AIO server to set a value trigger descriptor.

AIO Set Time Trigger

This command instructs the AIO client on a BCM20706 to set a time trigger on the AIO server.

Table 106: AIO Set Time Trigger Command

Item	*Description*	
Operating code	0x07	
Parameters	Type (1 byte)	1: Analog IO 2: Digital IO
	Index (1 byte)	Index of the analog or digital IO, starting with 0.
	Condition (1 byte)	0: No time trigger 1: Periodic 2: Not more often than a set time 3: Value changed N times, where N is a count that can be set.
	Values(variable bytes)	These bytes are a function of the condition set.

After a BCM20706 receives this command, it sends a write request to the AIO server to set a time trigger descriptor.

AIO Set User Description

This command instructs the AIO client on a BCM20706 to set the user description on the AIO server.

Table 107: AIO Set User Description Command

Item	*Description*	
Operating code	0x08	
Parameters	Type (1 byte)	1: Analog IO 2: Digital IO
	Index (1 byte)	Index of the analog or digital IO, starting with 0.
	Description (variable bytes)	User description

AIO Disconnect

This command instructs the AIO client on a BCM20706 to disconnect from the AIO server.

Table 108: AIO Disconnect Command

Item	*Description*
Operating code	0x09
Parameters	–

After a BCM20706 receives this command, it terminates its connection with the AIO server.

Miscellaneous Commands— HCI_CONTROL_GROUP_MISC

The miscellaneous commands allow the host to send the general commands as defined by the BCM20706.

Ping Request

This miscellaneous command sends a Ping Request to the BCM20706. The application running on the BCM20706 is expected to respond back with a Ping Reply event (see "Ping Request Reply"). The Ping Reply event is expected to return back the data sent as part of the Ping Request.

Table 109: Ping Request Command

Item	*Description*
Operating code	0x01
Parameter	Data (variable bytes)

Get Version

This miscellaneous command requests the BCM20706 to report the SDK version used to build the embedded application. The application running on the BCM20706 is expected to respond back with a Version Info event (see "Version Info").

Table 110: Get Version Command

Item	*Description*
Operating code	0x02

Section 5: WICED HCI Control Protocol Events

Device Events—HCI_CONTROL_GROUP_DEVICE

The device events are general events and state transitions reported by the BCM20706.

Command Status

The Command Status event indicates to the MCU that execution of the command has been started or that a command has been rejected due to the state of the hci_control application.

Table 111: Command Status Event

Item	*Description*		
Operating code	0x01		
Parameter	Status (1 byte)	0:	Execution of the command has started.
		1:	The command has been rejected because the previous command is still executing.
		2:	The Connect command has been rejected because the specified device is already connected.
		3:	The Disconnect command has been rejected because the connection is down.
		4:	The handle parameter in the command is invalid.
		5:	The Discover, Read, or Write command has been rejected because the previous command has not finished executing.
		6:	Invalid parameters passed in the command.
		7:	Bluetooth stack on BCM20706 failed to execute the command.
		8:	Embedded application loaded on the BCM20706 does not support processing of the commands of the requested group.
		9:	Embedded application loaded on BCM20706 does not support the command requested by the MCU.
		10:	LE application cannot send notification or indication because the GATT client is not registered.
		11:	Out of memory.
		12:	Operation disallowed.

WICED Trace

When tracing is enabled (see "Trace Enable"), the BCM20706 sends WICED_BT_TRACE statements over UART for the MCU to display.

Table 112: WICED Trace Event

Item	Description
Operating code	0x02
Parameters	WICED_BT_TRACE statements (ASCII string)

HCI Trace

When tracing is enabled (see "Trace Enable"), the BCM20706 sends binary data with the HCI commands, events, and data over UART for the MCU to display.

Table 113: HCI Trace Event

Item	Description	
Operating code	0x03	
Parameters	Type (1 byte)	0: HCI event
		1: HCI command
		2: Incoming HCI data
		3: Outgoing HCI data
	Raw HCI bytes (variable bytes)	Data formatted according to the Bluetooth Core Specification Vol. 2 Part E. [2]

NVRAM Data

For the situations when the BCM20706 does not have internal persistent storage, an application running on the BCM20706 can send data to the MCU in the NVRAM Data events.

Table 114: NVRAM Data Event

Item	Description	
Operating code	0x04	
Parameters	nvram_id (2 bytes)	ID of the NVRAM information chunk
	nvram_data (variable bytes)	Data corresponding to the nvram_id

Device Started

The hci_control application sends a Device Started event at the end of application initialization. Upon receiving the event, the MCU can assume that there are no active connections. The application logic determines the initial BLE scanning or advertising state and whether the Bluetooth device is discoverable and/or connectable.

Table 115: Device Started Event

Item	*Description*
Operating code	0x05
Parameters	–

Inquiry Result

The hci_control application sends an Inquiry Result event when the BCM20706 is performing the inquiry procedure and information is received about a discoverable peer device.

Table 116: Inquiry Result Event

tem	*Description*
Operating code	0x06
Parameters	Address (6 bytes)
	Class of device (CoD) (3 bytes)
	RSSI (1 byte)
	Extended inquiry response (EIR) data (variable bytes)

Inquiry Complete

The hci_control application sends an Inquiry Complete event on completion of the inquiry process.

Table 117: Inquiry Complete Event

Item	*Description*
Operating code	0x07
Parameters	–

Pairing Completed

The hci_control application sends a Pairing Completed event when a secure bond with the peer device has been established or when an attempt to establish a bond has failed.

Table 118: Pairing Complete Event

Item	*Description*		
Operating code	0x08		
Parameter	Pairing result (1 byte):	0:	Success
		1:	Passkey Entry Failure
		2:	OOB Failure
		3:	Pairing Authentication Failure
		4:	Confirm Value Failure
		5:	Pairing Not Supported
		6:	Encryption Key Size Failure
		7:	Invalid Command
		8:	Pairing Failure Unknown
		9:	Repeated Attempts
		10:	Internal Pairing Error
		11:	Unknown I/O Capabilities
		12:	SMP Initialization Failure
		13:	Confirmation Failure
		14:	SMP Busy
		15:	Encryption Failure
		16:	Bonding Started
		17:	Response Timeout
		18:	Generic Failure
		19:	Connection Timeout
	Bluetooth device address (6 bytes)		

Encryption Changed

The hci_control application sends an Encryption Changed event when a link to the peer device has been encrypted or when encryption has been turned off.

Table 119: Encryption Changed Event

Item	*Description*		
Operating code	0x09		
Parameter	Encryption status (1 byte):	0:	Encryption enabled
		Else:	Encryption disabled
	Bluetooth device address (6 bytes)		

Connected Device Name

The application running on the BCM20706 can send this command to inform the MCU of the friendly name of the connected device.

Table 120: Connected Device Name Event

Item	*Description*
Operating code	0x0A
Parameter	A variable length UTF-8 string representing a peer's device name.

User Confirmation Request

The application running on the BCM20706 device can be written to support numerical-comparison pairing or require a user permission to pair with another device. For these cases, the application sends this event to the MCU.

Table 121: User Confirmation Request Event

Item	*Description*
Operating code	0x0B
Parameter	Bluetooth device address (6 bytes)
	Numeric comparison code (4 bytes)

Device Error

The BCM20706 sends this event when it runs into a situation where it cannot proceed and needs to reset in order to recover. This can occur if the controller or the embedded application detects that it has entered a bad state.

Table 122: Device Error Event

Item	*Description*	
Operating code	0x0C	
Parameters	Application Error Code (1 byte)	Error code reported by Application
	Firmware Error Code (1 byte)	Error code reported by controller

Local Bluetooth Device Address

The BCM20706 sends this event in response to the Read Local Bluetooth Device Address Command.

Table 123: Local Bluetooth Device Address Event

Item	*Description*
Operating code	0x0D
Parameters	A 6-byte Bluetooth device address

Maximum Number of Paired Devices Reached

The BCM20706 sends this event if the maximum amount of keys stored for paired devices is reached. When this event occurs, the BCM20706 will also disable pairing since there are no more buffers available to store more pairing keys. The host will need to delete one or more NVRAM entries and enable pairing in order to pair with more devices.

Table 124: Maximum Number of Paired Devices Reached Event

Item	*Description*
Operating code	0x0E
Parameters	-

Buffer Pool Usage Statistics

The BCM20706 sends this event when the Read Buffer Pool Usage Statistics is received. The Buffer Pool Usage Statistics event message provides the buffer pool usage since the start of the application running on the BCM20706. This event message provides all buffer pool information such as the number of buffers allocated at the instance of receiving the Read Buffer Pool Usage Statistics command, the maximum number of buffers in use at a given time since the start of the system, and the total number of buffers in a pool. The actual number of pools are application dependent.

Table 125: Buffer Pool Usage Statistics Event

Item	*Description*
Operating code	0x0F
Parameters	Pool ID (1 byte)
	Pool Buffer Size (2 byte)
	Current Allocated Count (2 bytes)
	Maximum Allocated Count (2 bytes)
	Total Allocated Count (2 bytes)

LE Events—HCI_CONTROL_GROUP_LE

The LE events are related to the LE GAP profile and reported by the BCM20706.

LE Command Status

This event indicates to the MCU that LE command execution has started or that a command has been rejected due to the state of the hci_control application.

Table 126: LE Command Status Event

Item	Description	
Operating code	0x01	
Parameter	Status (1 byte)	See "Command Status Event"

LE Scan Status

The hci_control application sends a Scan Status event when the BCM20706 enters a new scanning state. A scanning state transition can be caused by a received LE Scan Command or internal application or stack logic.

Table 127: LE Scan Status Event

Item	Description		
Operating code	0x02		
Parameter	State[a]	0	No scan
		1	High-duty-cycle scan
		2	Low-duty-cycle scan

a. The high-duty-cycle and low-duty-cycle scan parameters for each state are defined in the wiced_bt_cfg.c file, which is included in every application.

LE Advertisement Report

The hci_control application sends an LE Advertisement Report event when the BCM20706 is scanning and it receives an advertisement or a scan response from a peer device.

Table 128: LE Advertisement Report Event

Item	*Description*
Operating code	0x03
Parameters	Event type indicating the type of advertisement report (1 byte)
	Address type indicating the Bluetooth address type (1 byte)
	Bluetooth device address (6 bytes)
	RSSI of the advertisement (1 byte)
	Advertisement data (variable bytes)

LE Advertisement State

The hci_control application sends an Advertisement State event when the BCM20706 enters a new advertisement state. An advertisement state change can be caused by an Advertisement Command received from the MCU or by internal application or stack logic.

Table 129: LE Advertisement State Event

Item	*Description*	
Operating code	0x04	
Parameter	State[a]	0 Not Discoverable
		1 High-duty-cycle discoverable
		2 Low-duty-cycle discoverable

a. The advertisement intervals and durations for each state are defined in the wiced_bt_cfg.c file, which is included in every application.

LE Connected

The hci_control application sends the LE Connected event when the BCM20706 establishes a connection with a peer Bluetooth LE device identified by address type and address. The connection handle identifies the connection and can be used in consecutive requests to disconnect or transfer data. If the Role parameter is zero, then the BCM20706 is a Master/Central in a newly established connection. Otherwise, the BCM20706 performs as a Slave/Peripheral. If the BCM20706 is performing as a GATT client, then the MCU can issue the GATT Command Read Request, GATT Command Write, or GATT Command Write Request commands to send data to the peer. Otherwise, the GATT Command Notify or GATT Command Indicate commands should be used.

Table 130: LE Connected Event

Item	*Description*	
Operating code	0x05	
Parameters	Type (1 byte)	Bluetooth-device address type.
	Address (6 bytes)	Bluetooth-device address
	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Role (1 byte)	The role is either peripheral or central.

LE Disconnected

When the Bluetooth LE connection with a peer device is disconnected, the hci_control application sends the LE Disconnected event. The connection handle and disconnection reason are passed as parameters.

Table 131: LE Disconnected Event

Item	*Description*	
Operating code	0x06	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Disconnection reason (1 byte)	–

LE Identity Address

When the LE Get Identity Address is called, the resolved Identity Address of the peer is returned via this event message.

Table 132: LE Identity Address Event

Item	*Description*	
Operating code	0x07	
Parameters	Address (6 bytes)	Resolved Identity address

LE Peer MTU

When the BCM20706 receives a Client MTU Request, this event will be passed to the MCU indicating the negotiated MTU size.

Table 133: LE Peer MTU Event

Item	*Description*	
Operating code	0x08	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event
	MTU size (2 bytes)	

LE Connection Parameters

When the BCM20706 receives a connection update complete event from a peer device, this LE Connection Parameters event will be passed to the MCU indicating the negotiated connection parameters or error code reflected by the Status byte.

Table 134: LE Connection Parameters Event

Item	*Description*	
Operating code	0x09	
Parameters	Status (1 byte)	0: Success, Else: Failure
	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Connection Interval (2 bytes)	–
	Connection Latency (2 bytes)	
	Supervision Timeout (2 bytes)	

GATT Events

The GATT events are related to the GATT profile and reported by the BCM20706.

GATT Command Status

This event indicates to the MCU that GATT command execution has started or that a command has been rejected due to the state of the hci_control application.

Table 135: GATT Command Status Event

Item	*Description*	
Operating code	0x01	
Parameter	Status (1 byte)	See "Command Status Event"

GATT Discovery Complete

The GATT Discovery Complete event indicates to an MCU that all results from a previously issued GATT Discover Services, GATT Discover Characteristics, or GATT Discover Descriptors command have been delivered. After receiving this event, the MCU can start a new discovery procedure.

Table 136: GATT Discovery Complete Event

Item	*Description*	
Operating code	0x02	
Parameter	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.

GATT Service Discovered

While performing a service discovery, the hci_control application sends the GATT Service Discovered event for every service found on a peer device. The connection handle identifies the connection to the peer device. The start and end handles identify the handles used by the service. The UUID identifies the remote service and can be either 2 or 16 bytes.

Table 137: GATT Service Discovered Event

Item	Description	
Operating code	0x03	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	UUID (2 or 16 bytes)	The UUID of the discovered service.
	Start handle (2 bytes)	The start handle of the service.
	End handle (2 bytes)	The end handle of the service.

GATT Characteristic Discovered

While performing a characteristic discovery, the hci_control application sends the GATT Characteristic Discovered event for every characteristic discovered on the peer device. The connection handle identifies the connection to the peer device. The value handle can be used by the MCU in consecutive GATT Read, GATT Write Command, GATT Write Request, GATT Notify, or GATT Indicate calls to send data to the peer. The UUID identifies the remote characteristic and can be either 2 or 16 bytes.

Table 138: GATT Characteristic Discovered Event

Item	Description	
Operating code	0x04	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Characteristic handle (2 bytes)	–
	UUID (2 or 16 bytes)	The UUID of the characteristic found.
	Characteristic properties (1 byte)	A bit mask of the properties supported by the discovered characteristic.
	Value handle (2 bytes)	The characteristic-value handle that can be used in consecutive reads and write.

GATT Descriptor Discovered

While performing a characteristic descriptor discovery, the hci_control application sends the GATT Descriptor Discovered event for every characteristic descriptor discovered on the peer device. The connection handle identifies the connection to the peer device. The handle can be used by the MCU in consecutive GATT Read or GATT Write Request commands to set or get a descriptor value. The UUID identifies the remote descriptor and can be either 2 or 16 bytes

Table 139: GATT Descriptor Discovered Event

Item	Description	
Operating code	0x05	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	UUID (2 or 16 bytes)	The descriptor UUID.
	Handle (2 bytes)	The descriptor handle, which can be used in subsequent reads and writes.

GATT Event Read Request

The GATT Event Read Request can be sent to the MCU to provide the value of the specific attribute. The connection handle identifies the connection to the peer device requesting the operation and the attribute handle identifies the attribute requested by the peer device. Upon receiving this request, the MCU should send the GATT Command Read Response (see “GATT Command Read Response”).

Table 140: GATT Event Read Request

Item	Description	
Operating code	0x06	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Attribute handle (2 bytes)	The attribute handle of the value being read.

See Figure 6: Reading a Dynamic Attribute from a Peer for a message sequence example where the GATT Event Read Request is used.

GATT Event Read Response

The GATT Event Read Response indicates to the MCU that the execution of the GATT Command Read Request has completed. The event includes the received data. The connection handle identifies the connection to the peer device for which the read procedure has been performed.

Table 141: GATT Event Read Response

Item	*Description*	
Operating code	0x07	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Data (variable bytes)	–

See Figure 5: Reading a Static Attribute from a Peer and Figure 6: Reading a Dynamic Attribute from a Peer for message sequence examples where the GATT Event Read Response is used.

GATT Event Write Request

The GATT Event Write Request indicates to the MCU that a write request from a connected peer has been received. The connection handle identifies the connection of the peer device that issued the write request and the attribute handle identifies the characteristic to be written.

The BCM20706 application can be designed to wait for the GATT Command Write Response (see "GATT Command Write Response") or to reply automatically to indicate the success of the write operation to the peer. Waiting for the GATT Command Write Response is required when the MCU needs to be able to reject peer write attempts.

Table 142: GATT Event Write Request

Item	*Description*	
Operating code	0x08	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Attribute handle (2 bytes)	The attribute handle of the value being written.
	Data (variable bytes)	–

See Figure 9: GATT Command Write Request – MCU Is Involved in a Write for a message sequence example where the GATT Event Write Request is used.

GATT Event Write Response

The GATT Event Write Response indicates to the MCU that the execution of a GATT Command Write, GATT Command Write Request, GATT Command Notify, or GATT Command Indicate has completed. The event includes the result of the write operation. The connection handle identifies the connection to the peer device for which the procedure has been performed.

For the GATT Command Write Request and GATT Command Indicate commands, issuance of the GATT Event Write Response indicates that the write has completed and that the peer has confirmed receiving the data. For the GATT Command Write and GATT Command Notify commands, issuance of the GATT Event Write Response indicates that the buffer has been allocated and a command has been scheduled for transmission.

Table 143: GATT Event Write Response

Item	*Description*	
Operating code	0x09	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Result (1 byte)	–

See Figure 9: GATT Command Write Request – MCU Is Involved in a Write for a message sequence example where the GATT Event Write Response is used.

GATT Event Indication

The GATT Event Indication event passes data received from a peer-sent GATT Indication to the MCU. The connection handle identifies the connection to the peer device from which the GATT Indication was received. The attribute handle identifies the characteristic value or descriptor to which data has been written.

The application running on the BCM20706 can behave in one of the following two ways after receiving a GATT Indication:

- It can reply automatically (with the success).
- In a flow-controlled scenario, it can pass the event up to the MCU and wait for the GATT Command Indicate Confirm from the MCU before replying.

Table 144: GATT Event Indication Event

Item	*Description*	
Operating code	0x0A	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Attribute handle (2 bytes)	This the handle of the attribute being accessed.
	Data (variable bytes)	–

See Figure 11: GATT Command Indicate Message Sequence for a message sequence example where the GATT Event Indication is used.

GATT Event Notification

The GATT Event Notification forwards data received from a peer-sent GATT Command Notify to the MCU. The connection handle identifies the connection to the peer device from which the GATT Command Notify was received. The attribute handle identifies the characteristic value to which data has been written.

Table 145: GATT Event Notification Event

Item	*Description*	
Operating code	0x0B	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Attribute handle (2 bytes)	This is the handle of the attribute being accessed.

See Figure 10: GATT Command Notify Message Sequence for a message sequence example where the GATT Event Notification is used.

GATT Event Read Error

The GATT Event Read Error message will be sent to the MCU in the case where a GATT Read Request command resulted in an error. This event message will include the received read result GATT error code, e.g. Insufficient Authentication.

Table 146: GATT Event Read Error

Item	*Description*	
Operating code	0x0C	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Read result (1 byte)	Received GATT error code.

GATT Event Write Request Error

The GATT Event Write Request Error message will be sent to the MCU in the case where a GATT Write Request command resulted in an error. This event message will include the received read result GATT error code, e.g. Insufficient Authentication.

Table 147: GATT Event Write Request Error

Item	*Description*	
Operating code	0x0D	
Parameters	Connection handle (2 bytes)	This is the connection handle reported in the LE Connected event.
	Read result (1 byte)	Received GATT error code.

HF Events—HCI_CONTROL_GROUP_HF

These events sent by the BCM20706 pertain to the functionality of the Hands-Free profile.

HF Open

This event is sent when an RFCOMM connection is established with an AG. At this point, the Service Level Connection (SLC) is still not established, so commands cannot yet be sent. The Bluetooth device address and connection handle are passed as parameters. The connection handle can be used by the MCU to send commands or to identify a peer device that caused the event.

Table 148: HF Open Event

Item	Description
Operating code	0x01
Parameters	Connection handle (2 bytes)
	Bluetooth device address of the AG (6 bytes)
	Status (1 byte)

HF Close

This event is sent when an RFCOMM connection with an AG is closed.

Table 149: HF Close Event

Item	Description
Operating code	0x02
Parameter	Connection handle (2 bytes)

HF Connected

This event is sent when the hands-free device and the AG have completed the protocol exchange necessary to establish an SLC. At this point, the application can send any commands to the BCM20706.

Table 150: HF Connected Event

Item	Description
Operating code	0x03
Parameters	Connection handle (2 bytes)
	32-bit mask of AG supported features

HF Audio Open

This event is sent when an audio connection with an AG is opened.

Table 151: HF Audio Open Event

Item	Description
Operating code	0x04
Parameter	Connection handle (2 bytes)

HF Audio Close

This event is sent when an audio connection with an AG is closed.

Table 152: HF Audio Close Event

Item	Description
Operating code	0x05
Parameter	Connection handle (2 bytes)

HF Audio Connection Request

This event is sent to the MCU on receiving an audio connection request from the AG. The MCU shall use the HF Accept/Reject Audio Connection command to accept/reject the connection request

Table 153: HF Audio Connection Request Event

Item	Description
Operating code	0x06
Parameters	Bluetooth device address of the AG (6 bytes)
	SCO Index (2 bytes)

HF Response

The HF Response events are sent when a response is received from the AG for a command sent by the application.

Table 154: HF Response Event Format

Item	Description
Operating code	See Table 155: HF Response Event Details
Parameters	Connection handle (2 bytes)
	Numeric value (2 bytes)
	Optional supporting character string

Table 155 shows various available values for the operating code, numeric value, and optional string parameters of Table 154.

Table 155: HF Response Event Details

Operating Code			
Code	***Description***	***Numeric Value***	***Optional String***
0x20	OK response	Command index of last command	–
0x21	Error response	Command index of last command	–
0x22	Extended error response	Command index of last command	Error code
0x23	Incoming call	–	–
0x24	Speaker gain	0–15	–
0x25	Microphone gain	0–15	–
0x26	Incoming call waiting	–	The calling party's number and number type. For example: "nnnnn, 128"
0x27	Call hold	0: Release all held calls 1: Release all active calls 2: Swap active and held calls 3: Hold active call	–
0x28	AG indicators	–	The AG indicators
0x29	Caller phone number	–	The caller's number
0x2A	AG indicator changed	–	The indicator number [1-7] and value. For example: "1,2" 1: Service indicator 2: Call status indicator 3: Call set up status indicator 4: Call hold status indicator 5: Signal Strength indicator 6: Roaming status indicator 7: Battery Charge indicator
0x2B	Number attached to voice tag	–	Phone number. For example: "nnnnnn"
0x2C	Voice recognition status	0: VR disabled in AG 1: VR enabled in AG	–
0x2D	In-band ring tone	0: No AG in-band ring tone 1: AG provides in-band ring tone	–
0x2E	Subscriber number	–	The subscriber number and number type. For example: "nnnnn, 128"
0x2F	Call hold status	0: AG put incoming call on hold 1: AG accepted held incoming call 2: AG rejected held incoming call	–
0x30	Operator information	–	–
0x31	Active call list	–	List of active calls
0x32	Supported HF indicators	–	–
0x33	Bluetooth Codec Selection	1: CVSD Codec 2: MSBC Codec	–
0x34	Unknown AT response	–	The unknown response that was received from the AG.

SPP Events—HCI_CONTROL_GROUP_SPP

These events sent by the BCM20706 pertain to the functionality of the Serial Port Profile (SPP).

SPP Connected

This event is sent when an SPP connection has been established with a peer device. The Bluetooth device address and connection handle are passed as parameters. The connection handle can be used by the MCU for future commands to send commands or data and to identify a peer device that has sent data.

Table 156: SPP Connected Event

Item	*Description*
Operating code	0x01
Parameters	Bluetooth device address (6 bytes)
	Connection handle (2 bytes)

SPP Service Not Found

This event is sent when a BCM20706 is able to connect to a peer device and perform SDP discovery, but the SPP service is not found.

Table 157: SPP Service Not Found Event

Item	*Description*
Operating code	0x02
Parameters	–

SPP Connection Failed

A BCM20706 sends this event when a connection attempt requested by an MCU is unsuccessful.

Table 158: SPP Connection Failed Event

Item	*Description*
Operating code	0x03
Parameters	–

SPP Disconnected

This event is sent when an SPP connection has been dropped.

Table 159: SPP Disconnected Event

Item	Description
Operating code	0x04
Parameter	Connection handle (2 bytes)

SPP TX Complete

A BCM20706 sends this event after a data packet received from an MCU, in an SPP Send Data command, has been queued for transmission. The MCU should not send another data packet until it has received this event for the previous packet.

Table 160: SPP TX Complete Event

Item	Description
Operating code	0x05
Parameters	Connection handle (2 bytes)
	Result (1 byte) 0 = Success, other result codes defined in SDK header file wiced_bt_rfcomm.h `wiced_bt_rfcomm_result_t` enum

SPP RX Data

A BCM20706 forwards SPP data received from a peer device in the SPP RX Data event.

Table 161: SPP RX Data Event

Item	Description
Operating code	0x06
Parameters	Connection handle (2 bytes)
	Data received from the peer

SPP Command Status

This event indicates to the MCU that a SPP command execution has started or that a command has been rejected due to the state of the hci_control application.

Table 162: SPP Command Status Event

Item	Description
Operating code	0x07
Parameters	Status (1 byte)
	See “Command Status Event".

Audio Events—HCI_CONTROL_GROUP_AUDIO

These events sent by the BCM20706 pertain to audio (A2DP) profile functionality.

Audio Command Status

This event indicates to the MCU that an Audio command execution has started or that a command has been rejected due to the state of the hci_control application.

Table 163: Audio Command Status Event

Item	Description
Operating code	0x01
Parameters	Status (1 byte)
	See "Command Status Event".

Audio Connected

This event is sent when an audio connection has been established with a peer device. The Bluetooth device address and connection handle are passed as parameters. The connection handle can be used by the MCU to send commands or data, and to identify a peer device that has sent data.

The *Absolute Volume Capable* flag indicates to the MCU whether a peer device can accept commands to set the volume.

Table 164: Audio Connected Event

Item	Description	
Operating code	0x02	
Parameters	Address (6 bytes)	Bluetooth device address of peer.
	Connection handle (2 bytes)	The handle to use during command and data exchanges.
	Absolute volume capable (1 byte)	1: Peer can accept commands to set volume. 0: Peer cannot accept commands to set volume.
	A2DP Features Flags (2 bytes)	The bitmap of the supported features published in the A2DP service of the connected AV sink device. Note that publishing of the features is optional. A value of zero indicates that the AV sink does not publish the features in the SDP record.

Audio Service Not Found

A BCM20706 sends this event when it is able to connect to a peer device and perform SDP discovery, but there is no A2DP service.

Table 165: Audio Service Not Found Event

Item	*Description*
Operating code	0x03
Parameters	–

Audio Connection Failed

A BCM20706 sends this event when a connection attempt requested by the MCU is unsuccessful.

Table 166: Audio Connection Failed Event

Item	*Description*
Operating code	0x04
Parameters	–

Audio Disconnected

A BCM20706 sends this event when an audio connection has been dropped.

Table 167: Audio Disconnected Event

Item	*Description*
Operating code	0x05
Parameter	Connection handle (2 bytes)

Audio Data Request

A BCM20706 sends this event when an audio stream is configured to send audio data over UART. The host is expected to maintain and send the number of packets requested as well as the number of bytes per packet.

Table 168: Audio Data Request Event

Item	*Description*	
Operating code	0x06	
Parameters	Bytes per packet (2 bytes)	
	Number of packets (1 byte)	
	Total Number of packets requested (2 bytes)	Total number of audio packets requested since the start of audio streaming, including the current Number of packets request
	Total Number of packets received (2 bytes)	Total number of audio packets received from the MCU

Audio Started

A BCM20706 sends this event when an audio stream has been started by an MCU-sent Audio Start command (see "Audio Start").

Table 169: Audio Started Event

Item	*Description*
Operating code	0x07
Parameter	Connection handle (2 bytes)

Audio Stopped

A BCM20706 sends this event when an audio stream has been stopped by an MCU-sent Audio Stop command (see "Audio Stop").

Table 170: Audio Stopped Event

Item	*Description*
Operating code	0x08
Parameter	Connection handle (2 bytes)

AV Remote Control Controller Events— HCI_CONTROL_GROUP_AVRC_CONTROLLER

AVRC Controller Connected

A BCM20706 sends the AVRC Connected event to an MCU when a peer device establishes an AVRC connection or after a connection requested by an AVRC Connect command has been successfully established.

Table 171: AVRC Controller Connected Event

Item	*Description*	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth address of the connected player.
	Status (1 byte)	Status of the connection establishment event. If 0, then the connection has been established successfully.
	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event.

AVRC Controller Disconnected

A BCM20706 sends the AVRC Disconnected event to an MCU to indicate that the AVRC connection has been terminated.

Table 172: AVRC Controller Disconnected Event

Item	*Description*	
Operating code	0x02	
Parameter	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event.

AVRC Controller Current Track Info

A BCM20706 sends this event when it receives information about new attributes of the track playing on the connected player. Each attribute reported by the player will be passed to the MCU in a separate AVRC Current Track Info event.

Table 173: AVRC Controller Current Track Info Event

Item	Description	
Operating code	0x03	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").
	Status (1 byte)	AVRC Response Status
	Attribute ID (1 byte)	1: Title 2: Artist 3: Album 4: Track number 5: Number of tracks 6: Genre 7: Playing time
	Attribute length (2 bytes)	The length of the attribute data string.
	Data (variable bytes)	Attribute data string.

AVRC Controller Play Status

A BCM20706 sends the AVRC Play Status event when a connected player reports a change in player status.

Table 174: AVRC Controller Play Status Event

Item	Description	
Operating code	0x04	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").
	Play status (1 byte)	0: Stopped 1: Playing 2: Paused 3: Forward seek 4: Reverse seek 255: Error

AVRC Controller Play Position

A BCM20706 sends an AVRC Play Status event when a connected player reports a change in the play position.

Table 175: AVRC Controller Play Position Event

Item	Description	
Operating code	0x05	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").
	Play position (4 bytes)	The play position in milliseconds since the beginning of the track.

AVRC Controller Track Change

A BCM20706 sends an AVRC Track Changed event when a connected player reports a track change. It is incumbent upon the MCU to request the updated track information.

Table 176: AVRC Controller Track Change Event

Item	Description	
Operating code	0x06	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Track End

A BCM20706 sends an AVRC Track End event when a connected player reports reaching the end of a track.

Table 177: AVRC Controller Track End Event

Item	Description	
Operating code	0x07	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Track Start

A BCM20706 sends an AVRC Track Start event when a connected player reports starting a new track.

Table 178: AVRC Controller Track Start Event

Item	Description	
Operating code	0x08	
Parameter	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").

AVRC Controller Settings Available

A BCM20706 sends an AVRC Settings Available event to report the player settings available for the connected player.

Table 179: AVRC Controller Settings Available Event

Item	*Description*	
Operating code	0x09	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").
	Settings (variable bytes)	An array of bytes indicating which attributes are supported by the connected player. Any value set in these bytes indicates that the setting is supported. The bits indicate the possible values for each setting: 1: The player supports an Equalizer. Bit 0: Unused Bit 1: Off supported Bit 2: On supported 2: The player supports Repeat mode. Bit 0: Unused Bit 1: Off supported Bit 2: Single Track repeat supported Bit 3: All Track repeat supported Bit 4: Group repeat supported 3: The player supports Shuffle mode. Bit 0: Unused Bit 1: Off supported Bit 2: All Track shuffle supported Bit 4: Group shuffle supported 4: The player supports Scan mode. Bit 0: Unused Bit 1: Off supported Bit 2: All track scan supported .

AVRC Controller Setting Change

A BCM20706 sends an AVRC Setting Change event to report the initial value or a settings change on a connected player.

Table 180: AVRC Controller Setting Change Event

Item	*Description*	
Operating code	0x0A	
Parameters	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event (see "AVRC Controller Connected").
	Number of Settings (1 byte)	Number of ID-Value Pairs
	Setting ID (1 byte)	The following values indicate the ID of the player setting: 1: Equalizer. 2: Repeat mode. 3: Shuffle mode. 4: Scan mode.
	Setting value (1 byte)	For ID = 1 (Equalizer): 1: On 2: Off For ID = 2 (Repeat mode): 1: Off 2: Repeat a single track 3: Repeat all tracks 4: Repeat a group of tracks For ID = 3 (Shuffle mode): 1: Off 2: Shuffle all tracks 3: Shuffle a group of tracks For ID = 4 (Scan mode): 1: Off 2: Scan all tracks 3: Scan a group of tracks

AVRC Controller Player Change

A BCM20706 sends an AVRC Player change event to report a change in the named connected player.

Table 181: AVRC Controller Player Change Event

Item	*Description*
Operating code	0x0B
Parameter	Name (n bytes). Character string that identifies the player by name.

AVRC Controller Command Status

This event indicates to the MCU that an AVRC command execution has started or that a command has been rejected due to the state of the hci_control application.

Table 182: AVRC Controller Command Status Event

Item	*Description*
Operating code	0xFF
Parameter	Status (1 byte). See “Command Status Event".

AV Remote Control Target Events— HCI_CONTROL_GROUP_AVRC_TARGET

AVRC Target Connected

A BCM20706 device sends the AVRC Connected event to an MCU when a peer device establishes an AVRC connection or after a connection requested by an AVRC Connect command has been successfully established.

Table 183: AVRC Target Connected Event

Item	Description	
Operating code	0x01	
Parameter	bd_addr (6 bytes).	Bluetooth address of the connected player.
	Session handle (2 bytes)	The session handle as reported in the AVRC Connected event.

AVRC Target Disconnected

A BCM20706 sends the AVRC Disconnected event to an MCU to indicate that the AVRC connection has been terminated.

Table 184: AVRC Target Disconnected Event

Item	Description	
Operating code	0x02	
Parameter	Session handle (2 bytes).	The session handle as reported in the AVRC Connected event.

AVRC Target Play

The BCM20706 sends this event to the MCU when a play command is received from a connected AVRC controller.

Table 185: AVRC Target Play Event

Item	Description
Operating code	0x3
Parameter	Connection handle (2 bytes)

AVRC Target Stop

The BCM20706 sends this event to the MCU when a stop command is received from a connected AVRC controller.

Table 186: AVRC Target Stop Event

Item	Description
Operating code	0x04
Parameter	Connection handle (2 bytes)

AVRC Target Pause

The BCM20706 sends this event to the MCU when a pause command is received from a connected AVRC controller.

Table 187: AVRC Target Pause Event

Item	Description
Operating code	0x05
Parameter	Connection handle (2 bytes)

AVRC Target Next Track

The BCM20706 sends this event to the MCU when a next track command is received from a connected AVRC controller.

Table 188: AVRC Target Next Track Event

Item	Description
Operating code	0x06
Parameter	Connection handle (2 bytes)

AVRC Target Previous Track

The BCM20706 sends this event to the MCU when a previous track command is received from a connected AVRC controller.

Table 189: AVRC Target Previous Track Event

Item	Description
Operating code	0x07
Parameter	Connection handle (2 bytes)

AVRC Target Begin Fast Forward

The BCM20706 sends this event to the MCU when a connected AVRC controller starts fast-forward operation. The target application should continue the fast forward operation until the End Fast Forward event is received.

Table 190: AVRC Target Begin Fast Forward Event

Item	Description
Operating code	0x08
Parameter	Connection handle (2 bytes)

AVRC Target End Fast Forward

The BCM20706 sends this event to the MCU when a connected AVRC controller terminates fast-forward operation.

Table 191: AVRC Target End Fast Forward Event

Item	Description
Operating code	0x09
Parameter	Connection handle (2 bytes)

AVRC Target Begin Rewind

The BCM20706 sends this event to the MCU when a connected AVRC controller starts rewind operation. The MCU should continue the Rewind operation until the End Rewind event is received.

Table 192: AVRC Target Begin Rewind Event

Item	Description
Operating code	0x0A
Parameter	Connection handle (2 bytes)

AVRC Target End Rewind

The BCM20706 sends this event to the MCU when a connected AVRC controller terminates rewind operation.

Table 193: AVRC Target End Rewind Event

Item	Description
Operating code	0x0B
Parameter	Connection handle (2 bytes)

AVRC Target Volume Level

The BCM20706 sends this event to the MCU when it receives a volume-level indication from a connected AVRC controller.

Table 194: AVRC Target Volume Level Event

Item	*Description*
Operating code	0x0C
Parameter	Connection handle (2 bytes)
	Volume level (1 byte). The percentage (0 to 100) of the maximum volume level of the local audio player to be set.

AVRC Target Repeat Settings

The BCM20706 sends this event to the MCU when a connected remote controller changes the player repeat attribute settings value.

Table 195: AVRC Target Repeat Settings Event

Item	*Description*	
Operating code	0x0D	The following are possible values:
Parameter	Setting value (1 byte)	0x01: OFF
		0x02: Single Track Repeat
		0x03: All Track Repeat
		0x04: Group Repeat

AVRC Target Shuffle Settings

The BCM20706 sends this event to the MCU when a connected remote controller changes the player shuffle attribute settings value.

Table 196: AVRC Target Shuffle Event

Item	*Description*	
Operating code	0x0E	The following are possible values:
Parameter	Setting value (1 byte)	0x01: OFF
		0x02: All Track Shuffle
		0x03: Group Shuffle

AVRC Target Command Status

This event indicates to the MCU that an AVRC command execution has started or that a command has been rejected due to the state of the hci_control application.

Table 197: AVRC Target Command Status Event

Item	*Description*	
Operating code	0xFF	The following are possible values:
Parameter	Status (1 byte)	See “Command Status Event"

HID Device Events —HCI_CONTROL_GROUP_HIDD

These events sent by the BCM20706 pertain to HID device profile functionality.

HID Opened

This event is sent when a HID connection has been fully established with a peer device, including control and interrupt channels.

Table 198: HID Opened Event

Item	Description
Operating code	0x01
Parameters	–

HID Virtual Cable Unplugged

The BCM20706 sends this event when a connected host sends a Virtual Cable Unplug message over the HID control channel.

Table 199: HID Virtual Cable Unplugged Event

Item	Description
Operating code	0x02
Parameters	–

HID Data

The BCM20706 sends a HID data event after receiving a HID report on either the control or interrupt channel.

Table 200: HID Data Event

Item	Description
Operating code	0x03
Parameters	Report type (1 byte)
	Report data (variable bytes)

HID Closed

The BCM20706 sends this event when a HID connection has been disconnected.

Table 201: HID Closed Event

Item	Description
Operating code	0x04
Parameter	Reason (1 byte)

AIO Server Events —HCI_CONTROL_GROUP_AIO_SERVER

These events sent by a BCM20706 pertain to AIO server functionality.

AIO Digital Output

This event sends a digital output value to an MCU.

Table 202: AIO Digital Output Event

Item	*Description*	
Operating code	0x01	
Parameters	Index (1 byte)	Digital IO index, starting with 0.
	Data (variable bytes)	An array of 2-bit values in a bit field in little endian order.

AIO Analog Output

This event sends an analog output value to an MCU.

Table 203: AIO Analog Output Event

Item	*Description*	
Operating code	0x02	
Parameters	Index (1 byte)	Analog IO index, starting with 0.
	Data (2 bytes)	The value of the analog signal as an unsigned 16-bit integer.

AIO Client Events —HCI_CONTROL_GROUP_AIO_CLIENT

These events sent by a BCM20706 pertain to AIO client functionality.

AIO Command Status

This event indicates to an MCU that AIO command execution has started or that a command was rejected due to the state of the application.

Table 204: AIO Command Status Event

Item	*Description*	
Operating code	0x01	
Parameter	Status (1 byte)	0: Command execution has started. 1: Command rejected because the previous command is still executing. 2: Connect command rejected; the specified device is already connected. 3: Disconnect command rejected because the connection is down. 4: Characteristic is not found. 5: Characteristic Descriptor is not found. 6: Invalid parameters passed in the command

AIO Connected

This event instructs an MCU that a connection with an AIO server had been created.

Table 205: AIO Connected Event

Item	*Description*
Operating code	0x02
Parameter	Device address (6 bytes)

AIO Read Response

This event sends a read response to an MCU.

Table 206: AIO Read Response Event

Item	*Description*	
Operating code	0x03	
Parameters	Status (1 byte)	0: Success. 2: Read not permitted.
	Data (variable bytes)	An unsigned 16-bit integer for analog IO or an array of 2-bit values in a bit field for digital IO.

AIO Write Response

This event sends a write response to an MCU.

Table 207: AIO Write Response Event

Item	Description	
Operating code	0x04	
Parameters	Status (1 byte)	0: Success. 3: Write not permitted.
	Data (variable bytes)	An unsigned 16-bit integer for analog IO or an array of 2-bit values in a bit field for digital IO.

AIO Input

The AIO client sends this event to an MCU after it receives notification about an IO module input change on the server.

Table 208: AIO Input Event

Item	Description	
Operating code	0x05	
Parameter	Type (1 byte)	1: Analog IO. 2: Digital IO.
	Index (1 byte)	Analog or digital IO index, starting with 0.
	Data (variable bytes)	An unsigned 16-bit integer for analog IO or an array of 2-bit values in a bit field for digital IO.

AIO Disconnected

This event informs an MCU that an AIO server has been disconnected.

Table 209: AIO Disconnected Event

Item	Description
Operating code	0x06
Parameter	Reason (1 byte)

Current Time Events—HCI_CONTROL_GROUP_TIME

Time Update

An application running on a BCM20706 sends this event to an MCU when it can to connect to a peer device and retrieve the current time via a current-time service or when a current-time service running on a peer device sends a time update notification (for example, a notification that daylight savings time (DST] has taken effect).

The date and time values are the local date and time reported by the server device. The time the server device provides is normally the correct time for the location adjusted for time zone and DST.

Table 210: Time Update Event

Item	*Description*	
Operating code	0x01	
Parameters	Year (2 bytes)	Current year
	Month (1 byte)	Current month
	Day (1 bytes)	Current day of month
	Hour (1 byte)	Current hour
	Minutes (1 byte)	Current minutes
	Seconds (1 byte)	Current seconds
	Exact time 256 (1 byte)	Current seconds fraction. LSB = 1/256 seconds.
	Day of week (1 byte)	Current day of the week: 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Sunday
	Adjust Reason (1 byte)	Bit field indicating the reason for the change in the time on the server. Bit 0: Manual time update Bit 1: External reference time update Bit 2: Time zone change Bit 3: Daylight savings time change

Test Events— HCI_CONTROL_GROUP_TEST

The Test commands allow the host to execute various tests on the BCM20706.

Encapsulated HCI Event

While in the Test Mode, the application encapsulates all the HCI Events received from the controller in the Encapsulated HCI Events and sends them to the MCU.

Table 211: Encapsulated HCI Event

Item	*Description*	
Operating code	0x10	
Parameters	HCI Event (variable bytes)	Fully formatted HCI Event

ANCS Events—HCI_CONTROL_GROUP_ANCS

ANCS Notification

An application running on a BCM20706 sends this event to an MCU when it receives a notification from a connected iOS device.

Refer to the following link for more information on the fields of this event:

https://developer.apple.com/library/ios/documentation/CoreBluetooth/Reference/AppleNotificationCenterServiceSpecification/Specification/Specification.html

Table 212: ANCS Notification Event

Item	*Description*	
Operating code	0x01	
Parameters	Notification UID (4 bytes)	Notification Unique Identifier
	Event ID (1 byte)	0: Notification added 1: Notification modified 2: Notification removed
	Category (1 bytes)	0: Other 1: Incoming call 2: Missed call 3: Voicemail 4: Social 5: Schedule 6: Email 7: News 8: Health and fitness 9: Business and finance 10: Location 11: Entertainment
	Flags (1 byte)	Bit mask of event flags Bit 0: Silent Bit 2: Important Bit 3: Preexisting Bit 4: Positive action possible Bit 5: Negative action possible
	Title (variable bytes)	Zero terminated UTF8 string with notification title.
	Message (variable bytes)	Zero terminated UTF8 string with notification message.
	Positive Action (variable bytes)	Zero terminated UTF8 string with positive action that can be performed by the MCU.
	Negative Action (variable bytes)	Zero terminated UTF8 string with negative action that can be performed by the MCU.

ANCS Command Status

This event indicates to the MCU that ANCS command execution has started or that a command has been rejected due to the state of the application.

Table 213: ANCS Command Status Event

Item	Description	
Operating code	0x02	
Parameters	Status (1 byte)	See "Command Status Event".

ANCS Service Found

This event indicates to the MCU that the ANCS service has been found on the given LE Connection Handle.

Table 214: ANCS Service Found Event

Item	Description	
Operating code	0x03	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.

ANCS Connected

This event indicates to the MCU that ANCS service has started. The MCU can expect to start receiving ANCS Notification events after the ANCS Connected event has occurred.

Table 215: ANCS Connected Event

Item	Description	
Operating code	0x04	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.
	Result (1 byte)	Provides additional status information, see "Command Status Event".

ANCS Disconnected

This event indicates to the MCU that ANCS service has stopped or has been unsubscribed to. ANCS Notification events shall not occur after the ANCS service has been disconnected.

Table 216: ANCS Disconnected Event

Item	*Description*	
Operating code	0x05	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.
	Result (1 byte)	Provides additional status information, see "Command Status Event".

AMS Events—HCI_CONTROL_GROUP_AMS

AMS Command Status

This event indicates to the MCU that AMS command execution has started or that a command has been rejected due to the state of the application.

Table 217: AMS Command Status Event

Item	Description	
Operating code	0x01	
Parameters	Status (1 byte)	See “Command Status Event".

AMS Service Found

This event indicates to the MCU that the AMS service has been found on the given LE Connection Handle.

Table 218: AMS Service Found Event

Item	Description	
Operating code	0x02	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.

AMS Connected

This event indicates to the MCU that AMS service has started.

Table 219: AMS Connected Event

Item	*Description*	
Operating code	0x03	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.
	Result (1 byte)	Provides additional status information, see "Command Status Event"

AMS Disconnected

This event indicates to the MCU that AMS service has stopped or has been unsubscribed to.

Table 220: AMS Disconnected Event

Item	*Description*	
Operating code	0x04	
Parameters	Connection Handle (2 bytes)	The connection handle reported in the LE Connected event.
	Result (1 byte)	Provides additional status information, see "Command Status Event".

Alert Events—HCI_CONTROL_GROUP_ALERT

Alert Notification

An application running on a BCM20706 forwards alerts received from a peer device in this event.

Table 221: Alert Notification Event

Item	*Description*	
Operating code	0x01	
Parameter	Alert level (1 byte)	Alert level requested by the peer device. 0: No alert 1: Medium alert 2: High alert.

iAP2 Events—HCI_CONTROL_GROUP_IAP2

The BCM20706 uses Apple iPod Accessory Protocol (iAP2) events to provide an MCU with protocol status changes and data received over an iAP2 External Accessory (EA) session.

IAP2 Connected

This event is sent when an EA session has been established with a peer device. The Bluetooth device address and connection handle are passed as parameters. The connection handle can be used by the MCU when sending subsequent commands or data and for identifying a peer device that has sent data.

This event can be sent for a connection originated by the MCU or by a peer iOS device.

Table 222: IAP2 Connected Event

Item	*Description*	
Operating code	0x01	
Parameters	bd_addr (6 bytes)	Bluetooth device address of the peer device with which an EA session has been established.
	Handle (2 bytes)	iAP2 EA session handle.

IAP2 Service Not Found

A BCM20706 sends this event when it is able to connect to a peer device and perform SDP discovery, but the iAP2 service is not found.

Table 223: IAP2 Service Not Found Event

Item	*Description*
Operating code	0x02
Parameters	–

IAP2 Connection Failed

The BCM20706 sends this event when a connection attempt requested by the MCU is unsuccessful.

Table 224: IAP2 Connection Failed Event

Item	*Description*
Operating code	0x03
Parameters	–

IAP2 Disconnected

This event is sent when a previously established EA session is disconnected.

Table 225: IAP2 Disconnected Event

Item	Description	
Operating code	0x04	
Parameter	Connection handle (2 bytes)	Connection handle reported in an IAP2 Connected event.

IAP2 TX Complete

A BCM20706 sends this event after a data packet received from an MCU in an IAP2 Send Data command has been queued for transmission. After sending the IAP2 Send Data command, an MCU should not send another data packet until it has received this event.

Table 226: IAP2 TX Complete Event

Item	Description	
Operating code	0x05	
Parameter	Connection handle (2 bytes)	Connection handle reported in an IAP2 Connected event.

IAP2 RX Data

A BCM20706 sends this event to forward iAP2 data received from a peer device during an EA session.

Table 227: IAP2 RX Data Event

Item	Description	
Operating code	0x06	
Parameters	Connection handle (2 bytes)	Connection handle reported in an IAP2 Connected event.
	Data (variable bytes)	Data received from a peer.

IAP2 Auth Chip Info

The BCM20706 sends this event after successfully processing an IAP2 Get Auth Chip Info command with chip information received from the authentication coprocessor.

Table 228: IAP2 Auth Chip Info Event

Item	*Description*	
Operating code	0x07	
Parameters	Device version (1 byte)	Device version reported by the auth chip
	Firmware version (1 byte)	Firmware version reported by the auth chip
	Protocol version (Major) (1 byte)	Protocol version reported by the auth chip
	Protocol version (Minor) (1 byte)	Protocol version reported by the auth chip
	Device ID (4 bytes)	Device identification reported by the auth chip

AG Events— HCI_CONTROL_GROUP_AG

These events sent by the BCM20706 pertain to the functionality of the hands-free profile audio gateway.

AG Open

This event is sent when an RFCOMM connection is established with a hands-free device. At this point, the Service Level Connection (SLC) is still not established, so commands cannot yet be sent. The Bluetooth device address and connection handle are passed as parameters. The connection handle can be used by the MCU to send commands or to identify a peer device that caused the event.

Table 229: AG Open Event

Item	*Description*	
Operating code	0x01	
Parameters	Connection handle (2 bytes)	Connection handle reported in an AG Connected event.
	Address (6 bytes)	Bluetooth device address of the AG.
	Status (1 byte)	–

AG Close

This event is sent when an RFCOMM connection with a hands-free device is closed.

Table 230: AG Close Event

Item	*Description*	
Operating code	0x02	
Parameters	Connection handle (2 bytes)	Connection handle reported in an AG Connected event.

AG Connected

This event is sent when the hands-free device and the AG have completed the protocol exchange necessary to establish an SLC. At this point, the application can send a command to establish an audio connection to the BCM20706.

Table 231: AG Connected Event

Item	*Description*	
Operating code	0x03	
Parameters	Connection handle (2 bytes)	Connection handle reported in an AG Connected event.
	Mask (4 bytes)	Mask of hands-free supported features.

AG Audio Open

This event is sent when an audio connection with a hands-free device is opened.

Table 232: AG Audio Open Event

Item	*Description*	
Operating code	0x04	
Parameters	Connection handle (2 bytes)	Connection handle reported in an AG Connected event.

AG Audio Close

This event is sent when an audio connection with a hands-free device is closed.

Table 233: AG Audio Close Event

Item	*Description*	
Operating code	0x05	
Parameters	Connection handle (2 bytes)	Connection handle reported in an AG Connected event.

Miscellaneous Events— HCI_CONTROL_GROUP_MISC

These events sent by the BCM20706 pertain to miscellaneous group of commands.

Ping Request Reply

This miscellaneous event is sent when the host sends a Ping Request (see "Ping Request"). The BCM20706 device responds with the exact data received in the Ping Request.

Table 234: Ping Request Reply Event

Item	*Description*
Operating code	0x01
Parameter	Data (variable bytes)

Version Info

The Version Info miscellaneous event is sent in reply to the MCU sending Get Version command (see "Get Version").

Table 235: Version Info Event

Item	*Description*
Operating code	0x02
Parameters	Major version (1 byte)
	Minor version (1 byte)
	Revision number (1 byte)
	Build number (2 bytes)
	Chip ID (3 bytes)
	Power class (1 byte)

An application that runs on a BCM20706 with power class 1 and built using SDK 1.0.1.188 should report 0x01, 0x00, 01, 0xBC, 0x00, 0xE2, 0x50, 0x00, 0x01.

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

WICED-SWUM115-R January 12, 2017

Phone: 949-926-5000
Fax: 949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com