Application Report
SWRA288B–June 2009–Revised September 2015

Enhanced HCILL: Four-Wire Power Management Protocol

ABSTRACT

The host controller interface, low level (HCILL) is a proprietary Texas Instruments protocol. The HCILL provides a baseband controller and a **Bluetooth**® host with a deterministic way to independently enter respective standby modes. This document discusses an enhanced version of the HCILL protocol designed to enable easier implementation on the host side while maintaining full compatibility with existing installations.

Contents

1 Introduction 1
2 Protocol Description 2
3 Alternative Implementations 10
4 Baseband Controller Hardware Behavior 11
5 Performing HCI_Reset 11

List of Figures

1 Deep-Sleep Basic State-Machine 4
2 Wake-Up by Baseband Controller 5
3 Wake-Up by Host 6
4 Wake-Up Collision Type 1 7
5 Wake-Up Collision Type 2 8

List of Tables

1 Terms and Abbreviations 2
2 Related Documentation 2
3 List of HCILL Commands 2

1 Introduction

The HCILL is a TI proprietary protocol, providing the baseband controller and the Bluetooth host with a deterministic way to independently enter respective standby modes. To maintain synchronization between the host and the **Bluetooth** device, each side must know when the other enters standby mode.

The enhanced HCILL (eHCILL) protocol includes several additions to the original HCILL specification that provide easier implementation on the host side and, at the same time, continuing to offer backward compatibility so that modifications are not required for a host that has implemented the original HCILL.

1.1 Terms and Abbreviations

Table 1 lists many of the terms and abbreviations used in this document.

Table 1. Terms and Abbreviations

Abbreviation /Term	Meaning / Explanation
Baseband Controller	TI Bluetooth single-chip solution
FW	Firmware
GPIO	General-purpose input / output
Host	The host processor that controls the baseband controller
HCI	Host controller interface
HCILL	Host controller interface, low level
eHCILL	Enhanced host controller interface, low level
HW	Hardware
SW	Software

1.2 Reference Documents

Table 2 lists reference documents.

Table 2. Related Documentation

Document	Reference
Bluetooth Specifications	1.1, 1.2, 3.0, and 4.0
BRF6150 HCI Vendor-Specific Commands	BT-SW-0026
BRF6300 HCI Vendor-Specific Commands	BT-SW-0030
BRF6350 HCI Vendor-Specific Commands	SWRU115B
Bluetooth (6450/127x/5500) HCI Vendor-Specific Commands	SWRU193
Bluetooth 18xx HCI Vendor-Specific Commands	SWRU303A

2 Protocol Description

2.1 HCILL Deep Sleep Commands List

Table 3 lists the HCILL commands related to deep-sleep mode.

Table 3. List of HCILL Commands

HCILL Command	Opcode
HCILL_GO_TO_SLEEP_IND	0x30
HCILL_GO_TO_SLEEP_ACK	0x31
HCILL_WAKE_UP_IND	0x32
HCILL_WAKE_UP_ACK	0x33

2.2 HCILL Protocol Commands Description

HCILL_GO_TO_SLEEP_IND

Only the baseband controller sends this command (the host does not need to send it) to request entry into Sleep state.

HCILL_GO_TO_SLEEP_ACK

The host sends this command in response to an HCILL_GO_TO_SLEEP_IND command to indicate that both the host and baseband controller can go to a low-power mode.

HCILL_WAKE_UP_IND

This command wakes up the sleeping party (host or controller).

HCILL_WAKE_UP_ACK

This message is a response to HCILL_WAKE_UP_IND to acknowledge a state of being awake and ready to communicate.

2.3 eHCILL Rules for Host

These simple rules allow easy understanding of the host behavior when deep sleep mode is enabled. The case situations are described from the host point of view.

Situation: HCILL_GO_TO_SLEEP_IND received.
Host should: Pull Request to Send (RTS) high, enable wakeup from Clear to Send (CTS), and send HCILL_GO_TO_SLEEP_ACK.

Situation: Wakeup from CTS.
Host should: Disable CTS interrupt and release RTS.

Situation: HCILL_WAKE_UP_IND received.
Host should: Send HCILL_WAKE_UP_ACK (1) (for one exception, see the following situation).

Situation: HCILL_WAKE_UP_IND sent and HCILL_WAKE_UP_IND received. (2)
Host should: Go directly to AWAKE state without sending an ACK.

Situation: Host is in sleep state and has data or a command to send.
Host should: Disable wakeup from CTS, send HCILL_WAKE_UP_IND, and lower RTS.

NOTE: If an HCI command is sent immediately before an HCILL_SLEEP_IND message is received, the host must complete the sleep preparation procedure (pull RTS high, enable wakeup from CTS, and send HCILL_SLEEP_ACK). The baseband controller then goes through the wakeup procedure to return the HCI_Command_Complete_Event.

2.4 Host Deep Sleep State Machine

Figure 1 shows the basic host state-machine diagram in the host Bluetooth deep-sleep implementation. As part of the eHCILL, the host is not required to send SLEEP_IND (see the Inactivity TimeOut feature in Section 2.7, *Configuring eHCILL*) but can if desired. Also, the host is programmed to wake up from asserted CTS and assert RTS on going to sleep to prevent the baseband controller from sending data.

(1) If HCILL_WAKE_UP_ACK is not received by the baseband controller, another HCILL_WAKE_UP_IND is sent to the host according to the HCI_VS_HCILL_Parameters command (see Section 2.7, *Configuring eHCILL*).
(2) This is an exception to the previous situation.

As Figure 1 shows, the state-machine states are logical states; that is, a device can be in sleep state but not in actual low-power mode.

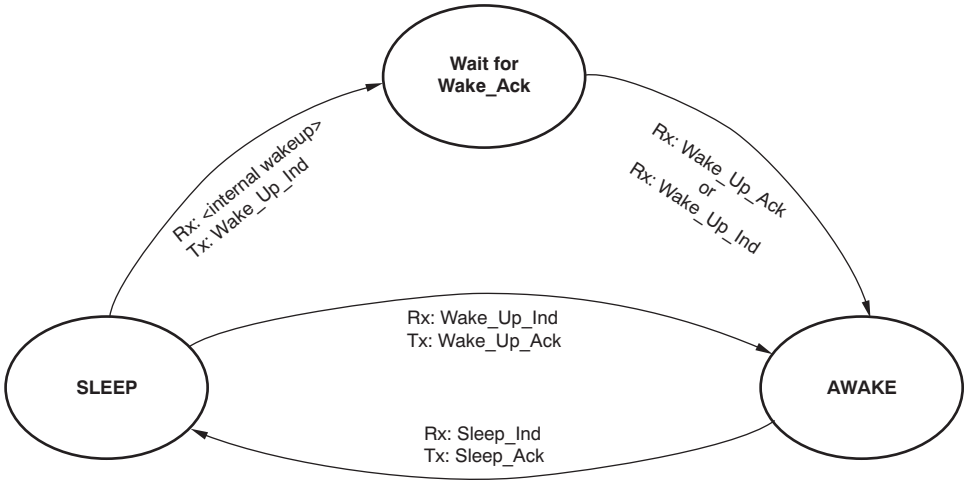

Figure 1. Deep-Sleep Basic State-Machine

2.5 eHCILL Implementation Guidelines

The following summary is a general description of the sleep procedure, the wakeup procedure, and the CTS interrupt service routine (ISR) from a host perspective.

Wakeup ISR on CTS

- Inform the power manager system (if applicable) that **Bluetooth** is awake (for example, turn on the UART clock).
- Release RTS.

Sleep Procedure

- HCILL_SLEEP_IND is received from baseband controller (see the Inactivity TimeOut feature in Section 2.7, *Configuring eHCILL*).
- Enable the Wakeup CTS ISR.
- Pull RTS high.
- Reply with HCILL_SLEEP_ACK→change HCILL state to *Sleep*.
- Inform power manager system (if applicable) that **Bluetooth** is asleep (for example, the UART clock can be shut off).

Wakeup by Baseband Controller

- Wakeup ISR is executed because of CTS interrupt (see the RTS pulse width feature in Section 2.7, *Configuring eHCILL*).
- Disable the Wakeup CTS ISR.
- Release RTS.
- HCILL_WAKEUP_IND is received.
- Reply with HCILL_WAKEUP_ACK→change HCILL state to *Awake* (see the WAKEUP_IND retransmission feature in Section 2.7, *Configuring eHCILL*).

Wakeup by Host

- Disable Wakeup ISR.
- Turn UART on (if turned off).
- Release RTS.
- Send HCILL_WAKEUP_IND.
- Wait for HCILL_WAKEUP_ACK (or HCILL_WAKEUP_IND in case of a collision type 1; see

Section 2.6.3, *Wakeup Collision 1*, and Section 2.6.4, *Wakeup Collision 2*) →change HCILL state to *Awake*.

2.6 Sleep and Wakeup Diagrams

2.6.1 Wakeup by Baseband Controller

Figure 2 shows the process of the baseband controller waking up the device.

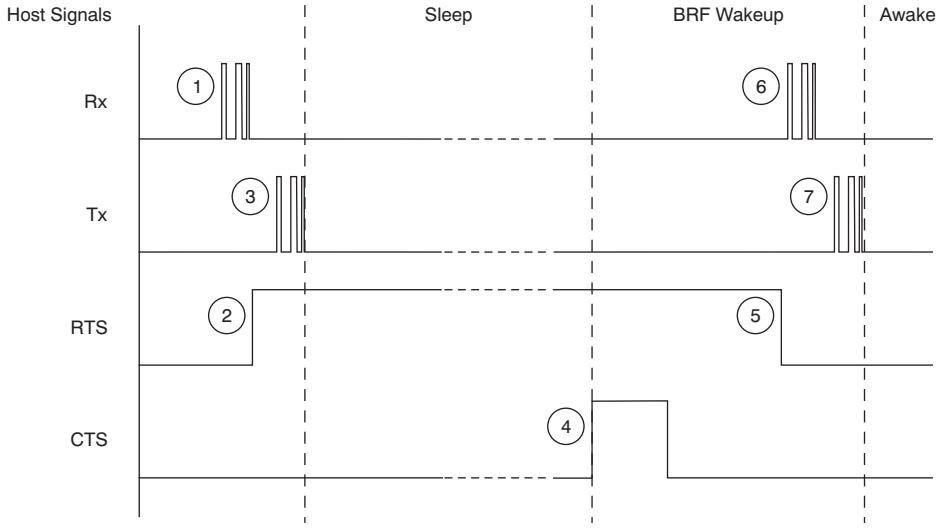

Figure 2. Wake-Up by Baseband Controller

1. HCILL_SLEEP_IND received from baseband controller.
2. Host pulls RTS high and enables wake-up from CTS. Pulling RTS is so the baseband controller can send the HCILL_WAKEUP_IND to the host only when the host is awake and ready for it.
3. Host sends HCILL_SLEEP_ACK.
4. Baseband controller wakes up host by asserting CTS for 150µs (recommended pulse width).
5. Host wakes up and lowers its RTS so the baseband controller can send the HCILL_WAKEUP_IND.
6. HCILL_WAKEUP_IND received from baseband controller once RTS is low.
7. Host sends HCILL_WAKEUP_ACK.

NOTE: If the host cannot control the RTS level, the RTS pin functionality might need to be changed to GPIO and asserted/deasserted as needed. When the device is awake, the pin functionality might need to be changed back to RTS.

2.6.2 Wakeup by Host

Figure 3 shows the procedure for the host to wake up the device.

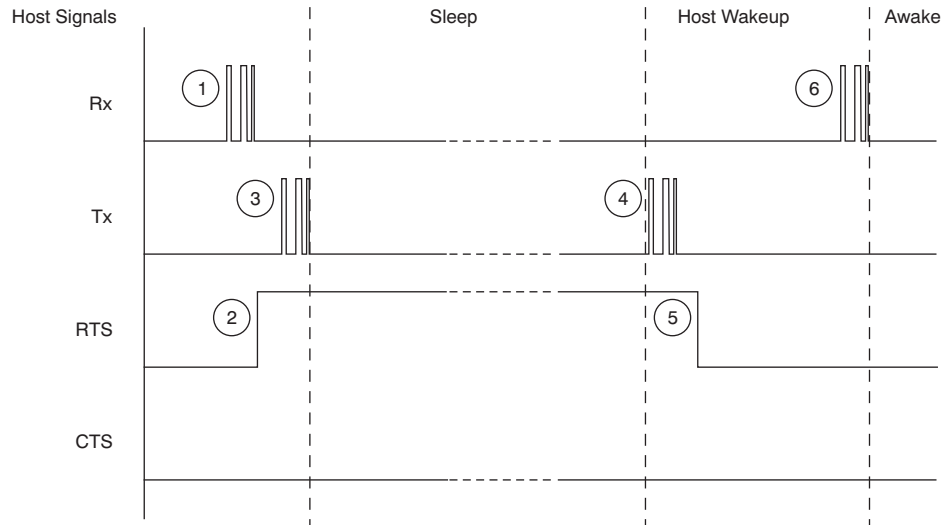

Figure 3. Wake-Up by Host

1. HCILL_SLEEP_IND is received from baseband controller.
2. Host pulls RTS high and enables wake-up from CTS so that the baseband controller can send HCILL_WAKEUP_IND to the host only when the host is awake and ready.
3. Host sends HCILL_SLEEP_ACK.
4. Host wakes up the baseband controller by sending a HCILL_WAKEUP_IND message.
5. Host lowers RTS so that the baseband controller can reply with HCILL_WAKEUP_ACK.
6. Host receives HCILL_WAKEUP_ACK once the baseband controller physically wakes up.

2.6.3 Wakeup Collision 1

Figure 4 shows the process of a wakeup collision type 1.

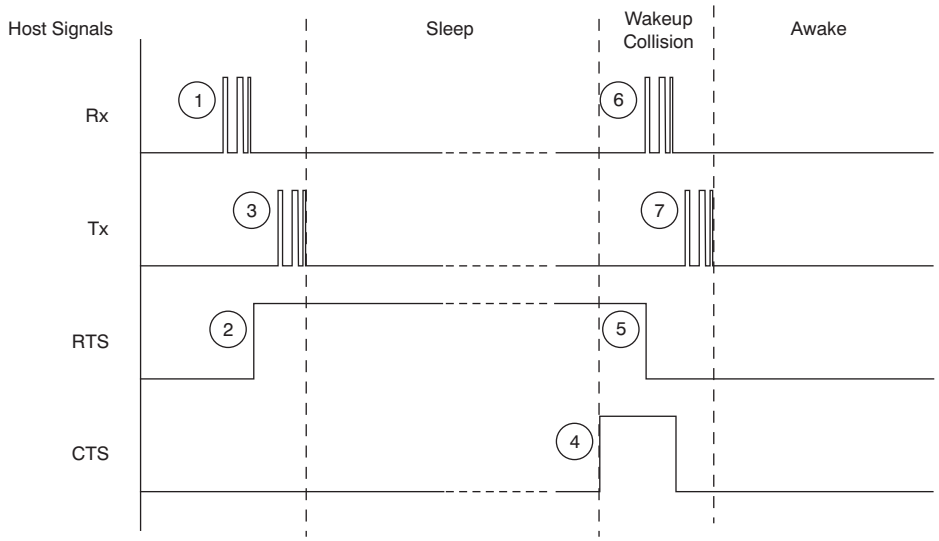

Figure 4. Wake-Up Collision Type 1

1. HCILL_SLEEP_IND is received from baseband controller.
2. Host pulls RTS high and enables wake-up from CTS so that the baseband controller can send HCILL_WAKEUP_IND to the host only when the host is awake and ready.
3. Host sends HCILL_SLEEP_ACK.
4. Baseband controller wakes up host by asserting CTS for 150 µs.
5. Host lowers RTS so that the baseband controller can reply with HCILL_WAKEUP_ACK (it is not aware this is a collision).
6. HCILL_WAKEUP_IND is received from the baseband controller (once RTS is low).
7. Host wakes up the baseband controller by sending a HCILL_WAKEUP_IND (delayed until CTS is low).

NOTE: If the host sends an HCILL_WAKEUP_IND, the host should consider reception of an HCILL_WAKEUP_IND as an HCILL_WAKEUP_ACK and go directly to the Wakeup state. Sending an HCILL_WAKEUP_ACK to the baseband controller in this case does not harm its state-machine.

2.6.4 Wakeup Collision 2

Figure 5 shows the process of a wakeup collision type 2.

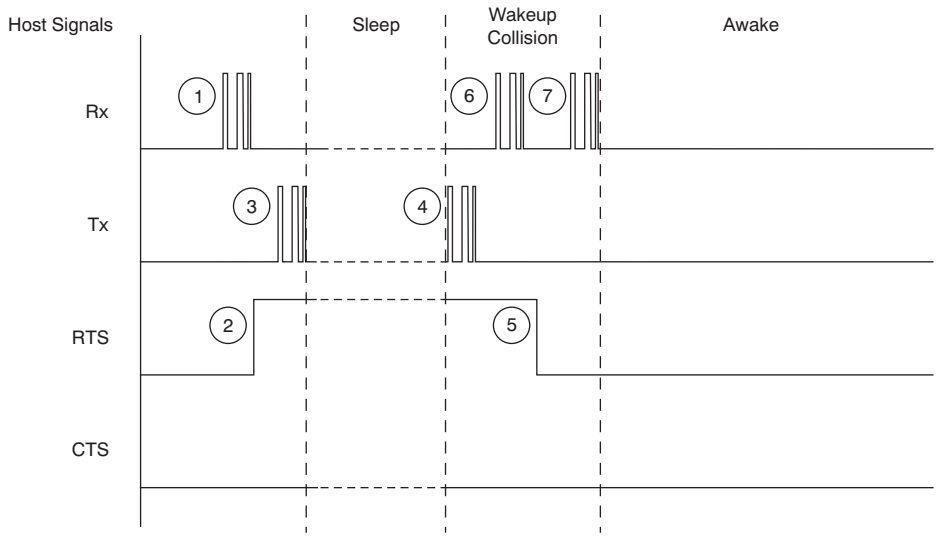

Figure 5. Wake-Up Collision Type 2

1. HCILL_SLEEP_IND is received from baseband controller.
2. Host pulls RTS high and enables wake-up from CTS so the baseband controller can send the HCILL_WAKEUP_IND to the host only when the host is awake and ready.
3. Host sends HCILL_SLEEP_ACK.
4. Host wakes up the baseband controller by sending a HCILL_WAKEUP_IND message.
5. Host lowers RTS so that the baseband controller can reply with HCILL_WAKEUP_ACK.
6. HCILL_SLEEP_IND is received from the baseband controller. This command has been in the queue and must be ignored by the host (sent until RTS goes low; it is not aware of a collision).
7. Host must wait to receive HCILL_WAKEUP_ACK (acknowledge to Step 4) to wake up.

2.7 Configuring eHCILL

As part of the eHCILL, the following three features must be configured: Inactivity timeout, WAKEUP_IND retransmission, and RTS pulse width.

- Inactivity timeout is a timer implemented in the baseband controller that determines when to send the HCILL_SLEEP_IND. If there is no activity on the UART line, the baseband controller is assumed to allow the host to go to sleep. This time is separate from the actual power mode the baseband controller can enter because, in some cases, the host can enter a low-power mode even though the baseband controller cannot (an active voice call, for example).
- WAKEUP_IND retransmission specifies the interval time to retransmit the WAKEUP_IND message until acknowledged (until WAKEUP_ACK is received successfully at baseband controller) to increase protocol robustness. This feature can also be used on hosts that do not support a wakeup interrupt on the CTS line (see Section 3.1, *Host Cannot Wakeup from the CTS Signal*). In this case, the UART RX is routed to an internal interruptible GPIO. The first WAKEUP_IND message is used for the wakeup interrupt and the second WAKEUP_IND is accepted as a WAKEUP byte on the host UART RX.
- RTS pulse width sets the minimum pulse width of the baseband RTS that is followed by a WAKEUP_IND message. The minimum pulse width should be set to 150 µs in case high baud rates require waking up some HLOS platforms.

The following command sets up eHCILL (for more information, see the relevant *HCI Vendor-Specific Commands* document or consult your local TI support):

HCI_VS_HCILL_Parameters(Inactivity_Timeout, WakeUp_Ind_Retransmission_TimeOut, RTS_pulse_width)

where:

- Inactivity_TimeOut is in **Bluetooth** frame units (1.25 ms); that is, an Inactivity_TimeOut value of 80 sets a timeout of 100 ms (80 × 1.25 ms). Default value is 100 ms.
- WakeUp_Ind_Retransmission_TimeOut is the timeout, in **Bluetooth** frame units (1.25 ms), for resending WAKEUP_IND if no acknowledgement is received. A value of 0 means no retransmission. Default value is 500 ms (400 × 1.25 ms).
- RTS_pulse_width sets the minimum pulse width of RTS (µs). A 0 value disables the pulse. Default value is 1 µs; however, 150 µs should be used.

2.8 Enabling Deep Sleep

Deep-sleep operation is disabled by default at power up. To enable the baseband controller deep-sleep feature and activate the eHCILL protocol, the host must send the HCI_VS_Sleep_Mode_Configurations command first (see the relevant *HCI Vendor-Specific Commands* document).

- HCI_VS_Sleep_Mode_Configurations 0xFD0C, 1, Deep_Sleep_Enable, Deep_Sleep_Mode, 0xFF, 0xFF, 0xFF, 0xFF, 100
  - Deep_Sleep_Enable = 0 (disable), 1 (enable).
    - Default = 0 (disable).
  - Deep_Sleep_Mode = 0 (HCILL),… ,0xFF (Don't change).
    - Default = 0 (HCILL).

For the current deep-sleep status, send *HCI_VS_Get_System_Status 0xFE1F, 0xFE1F* (for more information, see the relevant *HCI Vendor-Specific Commands* document).

2.9 Minimum Sleep Time

In cases where a baseband controller holds one or more connections that are all in Sniff or Park, the controller internally determines whether to enter deep sleep during the intervals.

The baseband controller can change the default value for this decision using the HCI_VS_Set_Min_Sleep_Time command (see the *HCI Vendor-Specific Commands* document). Normally, the default value should not be changed.

NOTE: This command does not directly affect HCILL protocol but only sets the condition for physical sleep of the baseband controller when deep sleep is enabled.

3 Alternative Implementations

Previous sections describe the recommended way to implement the eHCILL mechanism on the host. This section provides more implementation options that can be used when it is difficult to use the host RTS and CTS for purposes other than flow control.

3.1 Host Cannot Wakeup from the CTS Signal

There are three possibilities in this case, from a host perspective:

1. Normally, each UART line can be multiplexed with a GPIO line. Thus, instead of using the CTS input pin as CTS, first change its functionality to be GPIO (interruptible). Then, apply the wake-up ISR previously described for CTS to this new GPIO. In some cases, the hardware flow control might need to be disabled first.
2. At hardware design time, connect the host CTS signal to a separate GPIO and apply the wake up ISR described previously for CTS to this new GPIO.
3. If the CTS cannot be controlled, the UART RX can be routed to an internal interruptible GPIO. In this case, the first WAKEUP_IND message is used for the wakeup interrupt and the second WAKEUP_IND is accepted as a WAKEUP byte on the host UART RX. WAKEUP_IND retransmission value in HCI_VS_HCILL_Parameters command must be configured accordingly (see Section 2.7, *Configuring eHCILL*).

3.2 Host Cannot Control the RTS Line

There are two possibilities in this case:

1. Normally, each UART line can be multiplexed with a GPIO line. Thus, instead of controlling the RTS output pin as RTS, first change its functionality to be GPIO. Asserting or de-asserting a GPIO should not be a problem. In some cases, the hardware flow control might need to be disabled first.
2. The purpose of asserting the host RTS line during sleep mode is so the host can wake up from a HCILL_WAKEUP_IND byte only if awake and ready. If the RTS is not asserted (during sleep mode) and baseband controller must wake up, it sends the HCILL_WAKEUP_IND immediately after sending a CTS wakeup pulse whether the host is ready or not. In this case, the host must respond with a HCILL_WAKEUP_ACK as a result of the CTS wake-up indication regardless whether the HCILL_WAKEUP_IND is received or not. Recall that the baseband controller may keep sending HCILL_WAKEUP_IND messages until acknowledged (see Section 2.7, *Configuring eHCILL*).

4 Baseband Controller Hardware Behavior

The baseband controller hardware behaves according to these parameters:

- The baseband controller wakes up on a falling edge of the RX_HCI signal.
- The first byte received while in deep sleep (HCI_WAKE_UP_IND) wakes up the device but is discarded because reception is usually corrupted (first few bits are lost).
- To allow the host to send the HCI_WAKE_UP_IND byte, RTS is always low while in deep-sleep mode (enabling data reception).

5 Performing HCI_Reset

The power management protocol relies on both the baseband controller and the host being aware of the counterpart state. Breaking this synchronization can cause the baseband controller to seem inaccessible to the host. During normal operation, power management sequencing performs smoothly if the rules discussed here are followed.

However, to ensure that power management operation continues to run after an HCI_Reset, the host must implement the following steps as the reset procedure:

Step 1. Disable Deep-Sleep mode (see Section 2.8 *Enabling Deep Sleep*).

Step 2. Wait for Command Complete event.

Step 3. Send HCI_Reset.

Step 4. Wait for Command Complete event (and wait for CTS indication that baseband controller is active again).

Revision History

Changes from A Revision (March 2013) to B Revision	Page
• Removed NDA banner	1

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265