ff

BRF Enhanced HCILL

4-Wire Power Management Protocol

August 15, 2004

BT-SW-0024

Revision 1.2

Copyright © 2003, Texas Instruments Inc.

PRELIMINARY: documents contain information on a product under development and are issued for evaluation purposes only. Features characteristic data and other information are subject to change.

Bluetooth is a trademark of Bluetooth SIG, Inc. and is licensed to Texas Instruments Incorporated. All other trademarks are the property of their owners.

All information presented in this document is confidential.

Table of Content

Table of Figures

Table of Tables

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

1. Introduction

The HCILL is a TI proprietary protocol, providing the BRF and the Bluetooth Host with a deterministic way to independently go into their standby modes. In order to keep the synchronization between the Host and the BRF, it is important for each side to know when the other is going into standby mode.

The Enhanced HCILL includes a few additions to the original HCILL spec making it easier to implement on the host side and at the same time keeps backward compatibility so that hosts already implementing the original HCILL do not need to be changed.

2. Terms & Abbreviations

Abbreviation /Term	Meaning / Explanation
Baseband controller	BRF- TI's Bluetooth Single Chip
FW	Firmware
GPIO	General Purpose Input/Output
Host	The host processor which controls the baseband controller
HCI	Host Controller Interface
HCILL	Host Controller Interface Low Level
HW	Hardware
SW	Software

Table 1: Terms & Abbreviations

3. Reference Documents

Document	Reference
Bluetooth Specifications	BT Spec 1.1, 1.2
BRF6150 HCI Vendor Specific Command	BT-SW-0026
BRF6300 HCI Vendor Specific Command	TBD

Table 2: Reference Documents

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4. Protocol Description

4.1 HCILL deep sleep commands list

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

Table 3: List of HCILL Commands

4.2 HCILL protocol Commands description

HCILL_GO_TO_SLEEP_IND
Sent by BRF only – Indicating that the BRF is requesting to go to the Sleep state

HCILL_GO_TO_SLEEP_ACK
Sent by host in response to a HCILL_GO_TO_SLEEP_IND indicating that both host and BRF can go to a low power mode

HCILL_WAKE_UP_IND
Sent to wake up the sleeping party

HCILL_WAKE_UP_ACK
Response to HCILL_WAKE_UP_IND for acknowledging being awake and ready to communicate

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.3 HCILL rules

Following are some simple rules that allow easy understanding of the Host behavior when deep sleep is enabled.

Situation: **HCILL_GO_TO_SLEEP_IND** received
Host should: Pull RTS high, Enable wakeup from 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** (with only one exception – see below)

Situation: **HCILL_WAKE_UP_IND** sent and **HCILL_WAKE_UP_IND** received
Host should: Go directly to *AWAKE* state without sending an Ack

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

Important Note: If an HCI command is sent just before an HCILL_SLEEP_IND is received, the host should complete the sleep preparation (Pull RTS high, enable wakeup from CTS and send HCILL_SLEEP_ACK). The BRF will then go through the wakeup procedure for sending back the HCI_Command_Complete_Event.

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.4 Deep Sleep State Machine

This basic host state machine diagram illustrates the host Bluetooth deep sleep mechanism implementation.

Diagram explanation:
The state machine states are logical states e.g. a device can be in sleep state but not in actual low power mode.

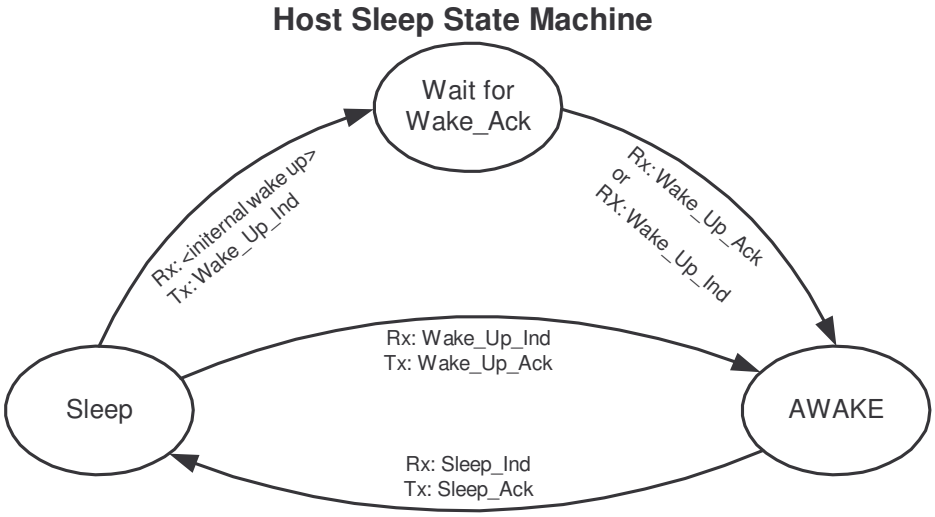

Figure 1: Deep Sleep Basic State Machine

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.5 HCILL Implementation Guidelines

Following are general descriptions of preparation for sleep procedure, wakeup procedure and the CTS Interrupt Service Routine content.

Wakeup Interrupt Service Routine (ISR) upon CTS

- Inform Power Manager system (if applicable) that BT is awake (turn on UART clock for example)
- Release RTS

Sleep procedure

- HCILL_Sleep_Ind received from BRF
- 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 BT is asleep (can now shut off UART clock for example)

Wakeup by BRF

- Wakeup ISR is run due to CTS interrupt
- Disable the Wakeup CTS ISR
- HCILL_Wakeup_Ind received
- Reply with HCILL_Wakeup_Ack -> change HCILL state to 'Awake'

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) -> change HCILL state to 'Awake'

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.6 Sleep and wakeup diagrams

4.6.1 Wakeup by BRF

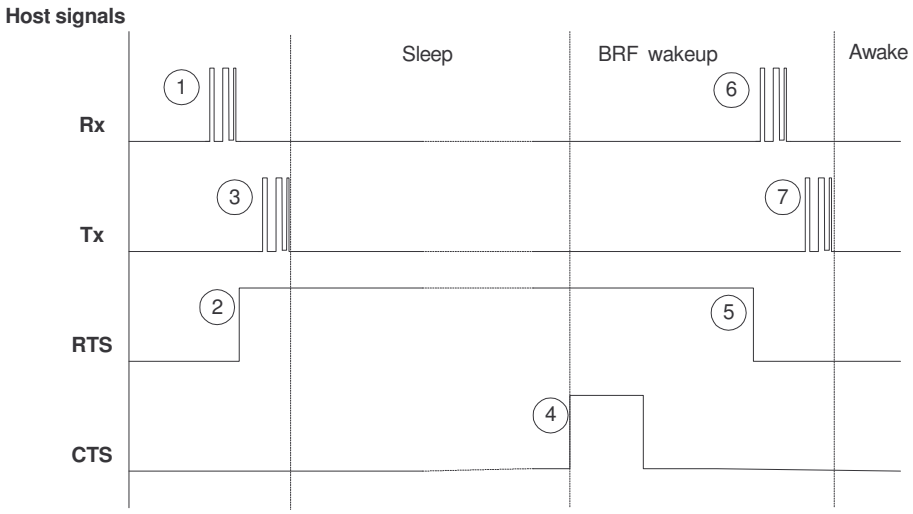

Figure 2: Wakeup by host

Diagram explained:

1. **HCILL_SLEEP_IND** received from BRF
2. Host Pulls **RTS** high and enables wakeup from CTS. Pulling RTS is so the BRF 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. BRF wakes up host by asserting **CTS** for 150uS
5. Host wakes up and lowers its **RTS** so the BRF can send the HCILL_WAKEUP_IND
6. **HCILL_WAKEUP_IND** received from BRF once RTS is low
7. Host sends **HCILL_WAKEUP_ACK**

Important Notes:

1. If the host cannot control RTS level, there may be a need to change the RTS pin's functionality to GPIO and assert/de-assert it as needed. When awake, change back to RTS functionality.
2.

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.6.2 Wakeup by Host

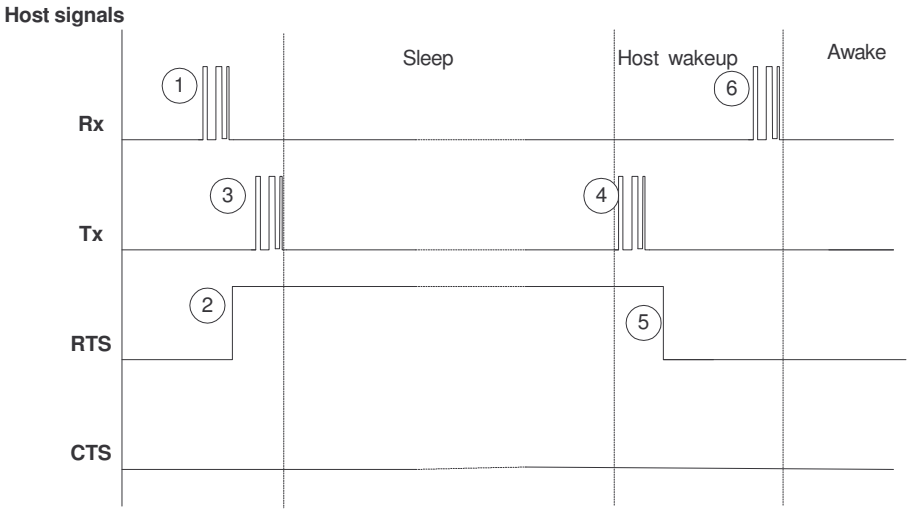

Figure 3: Wakeup by host

Diagram explained:

1. **HCILL_SLEEP_IND** received from BRF
2. Host Pulls **RTS** high and enables wakeup from CTS. Pulling RTS is so the BRF 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. Host wakes up the BRF by sending it an **HCILL_WAKEUP_IND**
5. Host lowers **RTS** so that the BRF can reply with HCILL_WAKEUP_ACK
6. Host receives **HCILL_WAKEUP_ACK** Once the BRF physically wakes up

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.6.3 Wakeup collision

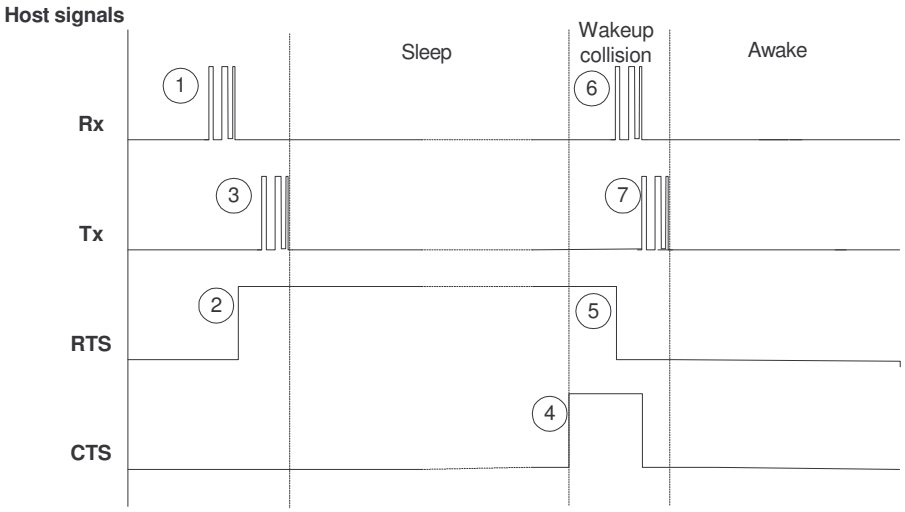

Figure 4: Wakeup collision

Diagram explained:

1. **HCILL_SLEEP_IND** received from BRF
2. Host Pulls **RTS** high and enables wakeup from CTS. Pulling RTS is so the BRF 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. BRF wakes up host by asserting **CTS** for 150uS
5. Host lowers **RTS** so that the BRF can reply with HCILL_WAKEUP_ACK (not aware this is a collision)
6. **HCILL_WAKEUP_IND** received from BRF (once RTS is low)
7. Host wakes up the BRF by sending it an **HCILL_WAKEUP_IND** (delayed until CTS is low)

Important Note: If the hosts sends an HCILL_WAKEUP_IND, it 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 BRF in this case will not harm the BRF state machine.

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

4.7 Inactivity timer

The Inactivity timer is a timer implemented in the BRF that determines when it will send the HCILL_SLEEP_IND. It is assumed that if there is no activity on the UART line, the BRF can allow the host to go to sleep. This is detached from the actual power mode the BRF can go into since in some cases the host can go into a low power mode even though the BRF cannot (active voice call for example).

Following are the commands for controlling the Inactivity timer: By default this timer is set as previously default in BRF6150, hence 100ms. For more info please see the relevant "HCI Vendor Specific Commands" document or consult with your local support.

- **BRF6150** (Init script 31.22 and later) – HCI_VS_Set_Just_Wakeup_Time(time) , where 'time' is in BT frames (1.25ms). Default inactivity time before initiating HCILL_SLEEP_IND is 100ms.
- **BRF63xx** (available in BRF6300 version 3.11 and later) - HCI_VS_ HCILL_Parameters(time, 0, 150) , where 'time' is in BT frames (1.25ms). Default inactivity time before initiating HCILL_SLEEP_IND is 100ms.

4.8 Enabling deep sleep

The default setting at power up is that Deep Sleep operation is disabled. In order to enable the BRF Deep Sleep feature and making the HCILL protocol active, the Host must send an HCI_VS_Set_Sleep_Mode command first (see "BRF6XXX HCI Vendor Specific Command" document)

4.9 Minimum Sleep Time

In cases where a BRF holds one or more connection that are all in either Sniff or Park, the BRF internally makes a decision whether to go to deep sleep during the intervals.

In BRF6150 there is a way to change the default value for this decision. This is achieved by using the HCI_VS_Set_Min_Sleep_Time command (see the "HCI Vendor Specific Command" document).

Normally there should be no reason to change the BRF default value for this.

Important Note: This command does not directly affect HCILL protocol, all it sets is the condition for physical sleep of the BRF when deep sleep is enabled.

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

5. Additional implementation options and alternatives

Previous sections describe the recommended way of implementing HCILL mechanism on the host. This section will try and provide some more implementation options that can be used in cases where there is difficulty using RTS and CTS for uses other than flow control.

5.1 Host cannot wakeup from the CTS signal

There 2 possibilities in this case:

1. Normally each UART line can be multiplexed with a GPIO line. So instead of using the CTS input pin as CTS, first change it's functionality to be GPIO. Then apply the wake up Interrupt Service Routine described earlier for CTS to this new GPIO. In some cases there may be a need to disable hardware flow control first.
2. At HW design time, connect the host CTS signal to a separate GPIO and apply the wake up Interrupt Service Routine described earlier for CTS to this new GPIO.

5.2 Host cannot control the RTS line

There 2 possibilities in this case:

1. Normally each UART line can be multiplexed with a GPIO line. So instead of controlling the RTS output pin as RTS, first change it's functionality to be GPIO. There should be no problem asserting de-asserting a GPIO. In some cases there may be a need to disable hardware flow control first.
2. The purpose of the assertion of RTS is so the HCILL_WAKEUP_IND byte is received by the host only after when it is awake and ready for it. If the RTS is not asserted, the BRF will send the HCILL_WAKEUP_IND immediately after the CTS wakeup pulse whether the host is ready for it or not. In this case, the host should respond with a HCILL_WAKEUP_ACK as a result of the CTS wake up regardless to whether the HCILL_WAKEUP_IND was actually received or not.

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

6. BRF Hardware behavior

- The BRF 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 since its reception is usually corrupted (first few bits are lost).
- RTS is always low while in deep sleep mode (enabling data reception). This is in order to allow the host to send the HCI_WAKE_UP_IND byte.

7. Performing HCI_Reset

The power management protocol relies on both BRF and host being aware of their counterpart state. Breaking of this synchronization can cause the BRF to seem inaccessible to the host. During normal operation, power management sequencing will perform smoothly if the above rules are kept.

However, in order to make sure that power management operation continues to run after an HCI_Reset, it is important that the Host implements the following steps as the Reset procedure:

- Disable Deep Sleep mode (HCI_VS_Set_Sleep_Mode, see “BRF HCI Vendor Specific Command” document)
- Wait for Command Complete event
- Send HCI_Reset
- Wait for Command Complete event (and wait for CTS indication that BRF is active again)
- Enable Deep Sleep again mode (HCI_VS_Set_Sleep_Mode, see “BRF HCI Vendor Specific Command” document)

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.

Important Notice

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE–SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, the customer to minimize inherent or procedural hazards must provide adequate design and operating safeguards.

TI assumes no liability for applications assistance or customer product design. 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 of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 2003, Texas Instruments Inc.
All information presented in this document is confidential.