SDIO Host-Controller Interface 6300 Application Note

BT-AN-0057
Revision 0.5

May 23, 2006

Abstract

This document describes the SDIO Host-Controller interface:

- Supported Features
- The sleep/ wake-up protocol between the SDIO Host and the BRF device
- Shared SDIO bus Mode

Revision Control

Author Name	Description	Revision	Date	Approved By	Date
Dana Ram	Creation	0.1	November 8, 2005		
Dana Ram	Implemented R&D comments regarding the deep-sleep protocol	0.2	November 28, 2005		
Dana Ram	Added genral description and shared SDIO	0.3	December 6, 2005		
Dana Ram	Implemented comments from AE, FAE and R&D	0.4	December 22, 2005		
Dana Ram	Added Type-A header Issue description and fix	0.5	May 23, 2006		

Table of Contents

1 Overview

BRF6300 provides an SDIO interface as an additional transport layer to the UART.

SDIO stands for Secure Digital Input/Output. It is based on and compatible with SD memory card technology. The compatibility includes mechanical, electrical, power, signaling and software. The intent of SDIO card is to provide high speed data I/O with low power consumption for mobile electronic device.

Note: *the reader of this document should be well acquainted with the SDIO card specification Ver1.00 and with the Type-A Specification for Bluetooth ver1.00.*

2 Supported features

- SDIO Spec 1.0 compliant:
  - SDIO 1-bit SD mode
  - SDIO mandatory commands and functions
  - SDIO Card Type-A for Bluetooth compliant
- Only1.8V IO voltage
- up to 25MHz external clock rate
- Number of Functions supported: 1
- Interrupt to host supported (for all modes)
- Maximum number of data bytes transferred in one transaction: **128 bytes**
- Abort supported only for CMD53 Read transactions
- Read Wait NOT supported
- Suspend and resume NOT supported
- Block Basis mode not supported
- Multi-block data transfer NOT supported
- Infinite data transfer NOT supported
- Data transfer to function 1 via CMD53 only
- Shared SDIO bus mode

2.1 Supported Bluetooth Protocol layers in the SDIO Card Type-A for Bluetooth

Figure 1 depicts the supported Bluetooth protocol layer in the SDIO Card Type-A for Bluetooth. The radio, Baseband, LMP and HCI interface reside in the card.

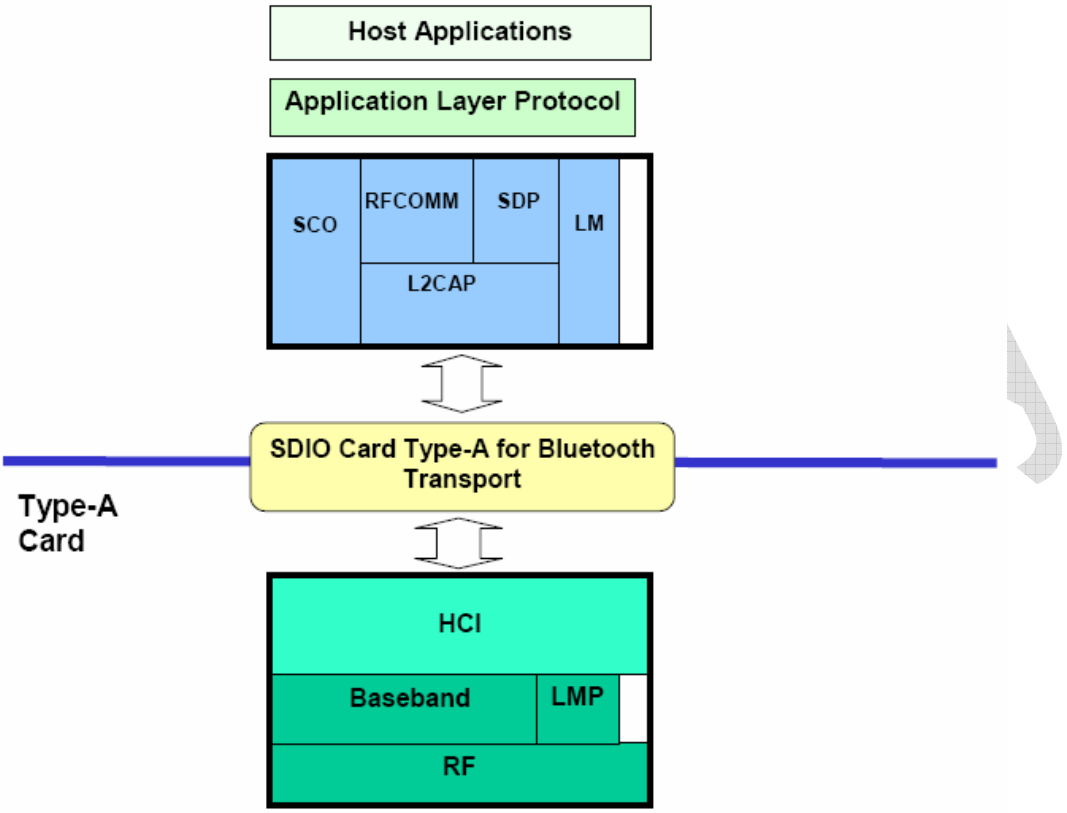

Figure 1 : Typical Application in Data Communications (from the Type-A Specification for Bluetooth ver1.00)

3 SDIO Interface Description

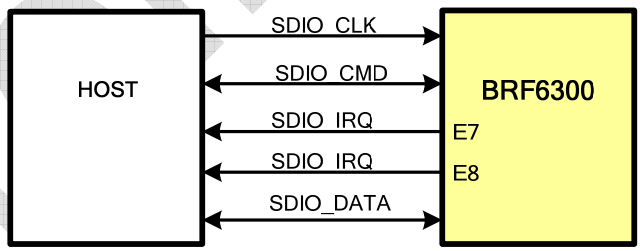

Figure 2 : SDIO Interface Signals

Table 1 : SDIO Interface Signals Description

Port name	In/Out	Description
SDIO_CLK	In	SDIO interface clock (0 to 25 MHz)
SDIO_CMD	Bi	CMD line in SDIO, Din in SPI mode (Single pins with different nomenclatures for SDIO modes respectively.
SDI_IRQ	out	IRQ
SDI_DATA	Bi	Data line

SDIO_IRQ is output on both E7 and E8 pins. This provides the following options to the host:

- Use E7 only to provide both the chip-initialization-complete signal and the IRQ signal. E7 first goes low as chip-initialization-complete signal. Thereafter E7 will function as SDIO_IRQ. Note that in this scheme, SDIO_IRQ will be low initially, even though the BRF6300 may not want to interrupt the host.
- Use Both E7 and E8: In this case E8 may be used as a dedicated SDIO_IRQ signal. E7 is used as the chip-initialization-complete signal as described above. Note, however, that E7 will not remain low, but will also toggle as SDIO_IRQ signal .

Both E7 and E8 will be set high after chip powered. Only E7 will go low to indicate chip initialization complete.

3.1 Initialization

After reset or power-up all the I/O functions are disabled and the SDIO will not respond to any commands from host except for CMD5.

All the SDIO transactions occur with respect to the POSITIVE EDGE of the SDIO clock.

The initialization sequence is as described in the SDIO card Specification Ver1.00 and to the Type-A Specification for Bluetooth ver1.00.

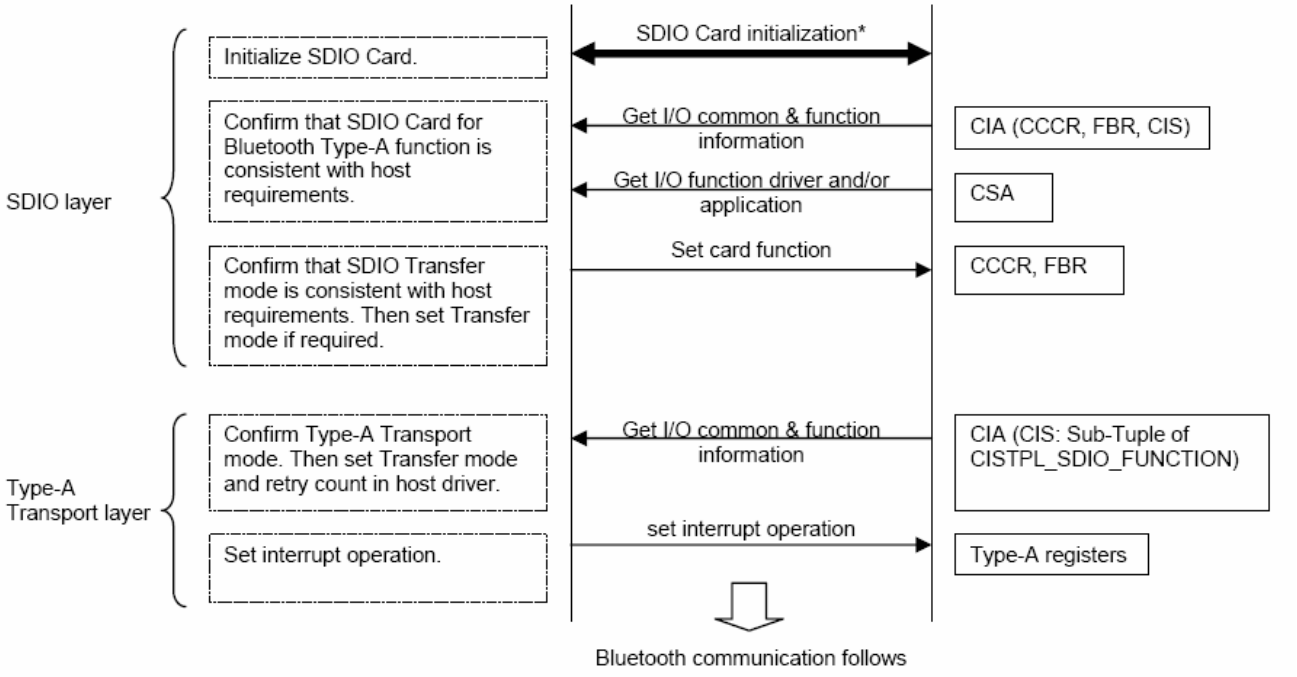

Figure 3 : Example of SDIO Card Type-A for Bluetooth Initialization Procedure

3.2 Type-A Header Issue

There may be a problem in the header interpretation by the BRF6300 V1.11/V1.21. On WINCE platforms we have experienced the following problem:

Type A header is 4 byte long and is composed of 3 bytes of the packet length and 1 byte of the Service ID.
BRF expects to get the whole header LSB first, meaning Service ID first and then Packet Length, but the WINCE Host sends Packet Length and then Service ID.

Let's say the header four bytes are: 0071 (packet-length is 7 and Service ID is 1)
BRF expects to get whole header: 1700, but host sends the packet length and then the service ID: 7001.

There is a relatively easy fix to the SDIO Host driver.
Please refer to 0Appendix B: Code example for fixing the Type-A Header Issue for a code example.

4 Deep-Sleep Protocol

4.1 General Description

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 low power mode. However the SDIO [Ver 1.00] specification doesn't explicitly define a low power mode protocol between a Host and a device.

This section describes how the SDIO Host and the BRF device inform each other of their power mode state.

4.2 The Deep- Sleep Protocol Registers

The SDIO deep-sleep protocol uses a master-slave configuration. The master initiates all transactions and it is therefore the master's responsibility to put the device into and out of low-power mode.

Two new registers were defined that are dedicated to the SDIO deep-sleep protocol. The existinf registers in the Bluetooth register map (defined by the Type-A Specification for Bluetooth) were not modified. The following table is taken from the Type-A Specification for Bluetooth.

Table 2: SDIO Card Type-A for Bluetooth Register Map

Add	RW	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	RO	Receiver Data	RDAT							
	WO	Transmitter Data	TDAT							
0x10	WO	Read Packet Control	0	0	0	0	0	0	0	PC RRT
0x11	WO	Write Packet Control	0	0	0	0	0	0	0	PC WRT
0x12	RO	Retry Control Status	0	0	0	0	0	0	0	RTC STAT
	WO	Retry Control Set	0	0	0	0	0	0	0	RTC SET
0x13	RO	Interrupt Identification	0	0	0	0	0	0	0	INTRD
	WO	Interrupt Clear	0	0	0	0	0	0	0	CL INTRD
0x14	RW	Interrupt Enable	0	0	0	0	0	0	0	EN INTRD
0x20	RO	Bluetooth Mode Status	0	0	0	0	0	0	0	MD STAT

The two new registers are in addition to the registers described in Table 2.
The first register called **Bluetooth Sleep Command** is used by the Host for two purposes. The Host will wake-up the BRF using this register and will also indicate to the BRF that it is allowed to enter low-power mode.

The second register called **Bluetooth Sleep State** is used by the Host to read the device power mode. By reading this register, the Host can establish if the BRF is in deep-sleep mode.

The two registers are depicted in Table 3.

Table 3: SDIO Deep-Sleep protocol registers.

Add	R/W	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x40	W/O	Bluetooth Sleep Command	0	0	0	0	0	0	0	SLP CMD
0x42	R/O	Bluetooth Sleep State	0	0	0	0	0	0	0	SLP STAT

4.3 Entering Deep-Sleep Mode

The BRF can enter deep-sleep mode (low-power mode) only if the Host allows it to do so. The Host will issue CMD52 that *writes* '1' to address 0x40 – asserting the SLP_CMD bit in the **Bluetooth Sleep Command** register. Table 4 shows the required command format.

Table 4 : CMD52- SDIO Host allows the BRF to go to deep-sleep mode

S	D	Command Index	R/W flag	Function Number	RAW flag	Stuff	Register Address	Stuff	Write Data or Stuff Bits	CRC	E
0	**1**	**110100b**	**1**	**001b**	**0**		**0x40**		**0x1**		**1**

When the BRF is in sleep task it will poll the SLP_CMD bit in the **Bluetooth Sleep Command** register. Only after this bit is asserted, the BRF will enter deep-sleep mode.
Upon entering deep-sleep mode the BRF asserts (writes '1') the SLP_STAT bit of the **Bluetooth Sleep State** register.

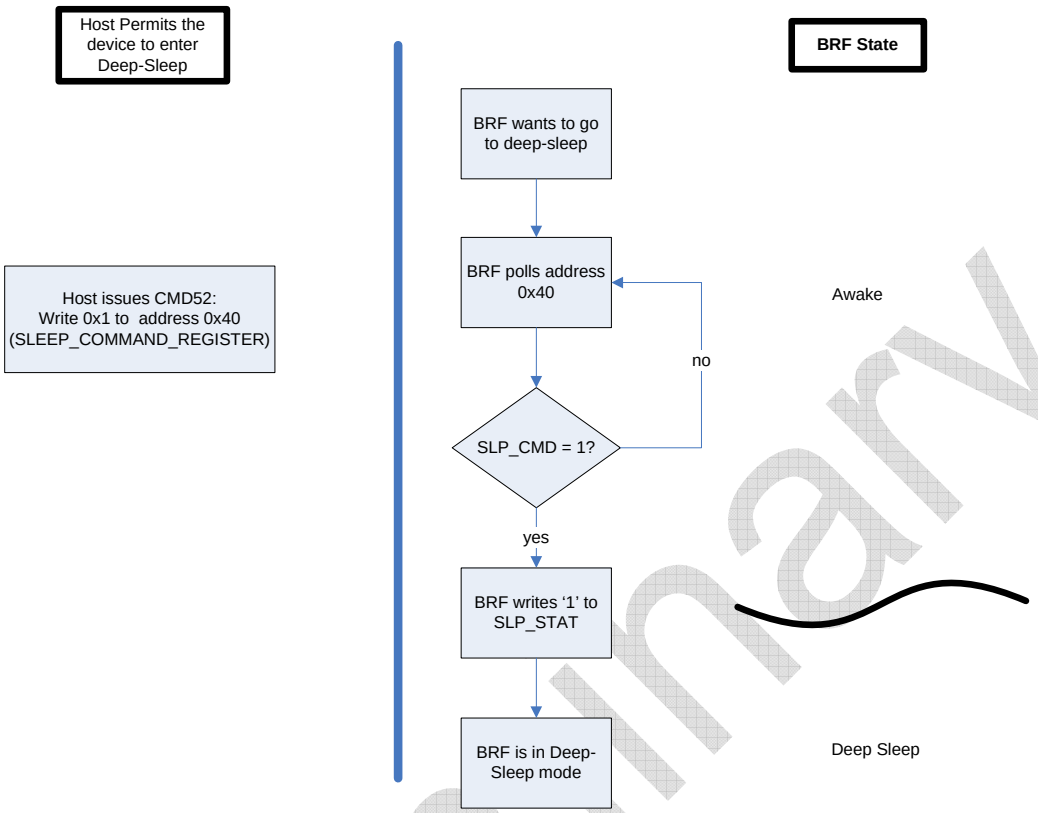

Figure 4 : Device enters deep-sleep mode

4.4 Exiting Deep-Sleep Mode

4.4.1 Host Initiates Wake-Up

The SDIO Host will wake-up the BRF by issuing CMD52 that *writes* '0' to address 0x40 – de-asserting the SLP_CMD bit in the **Bluetooth Sleep Command** register. Table 5 shows the required command format.

Table 5 : CMD52- SDIO Host allows the BRF to go to deep-sleep mode

S	D	Command Index	R/W flag	Function Number	RAW flag	Stuff	Register Address	Stuff	Write Data or Stuff Bits	CRC	E
0	1	110100b	1	001b	0		0x40		0x0		1

Upon waking-up the BRF de-asserts (writes '0') the SLP_STAT bit of the **Bluetooth Sleep State** register. (There is a hardware mechanism incorporated within the BRF that enables the SDIO Host to write to the device's registers even when the device is in deep-sleep mode.)

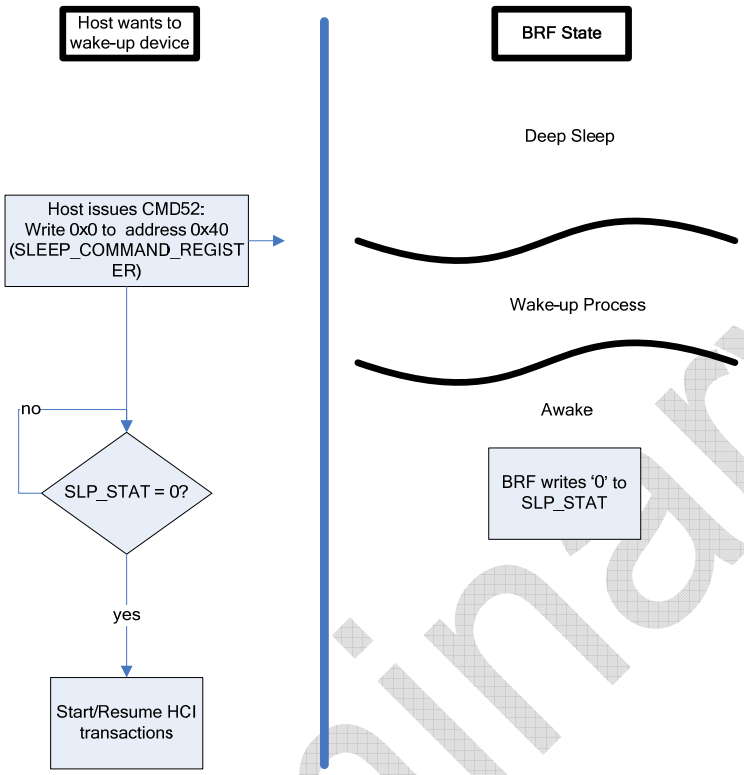

Figure 5 : Host wakes-up device

4.4.2 Device Initiates Wake-Up

Whenever the device want to exit the low power mode (i.e. it has an event to send), it asserts the interrupt line (IRQ). The host then must write '0' to SLP_CMD bit in the **Bluetooth Sleep Command** register. Polling the SLP_STAT is not required in this scenario.

Note: The behavior described above assumes that the BRF can signal the Host of a wake-up event asynchronously (even when the SD clock is not supplied to the BRF).

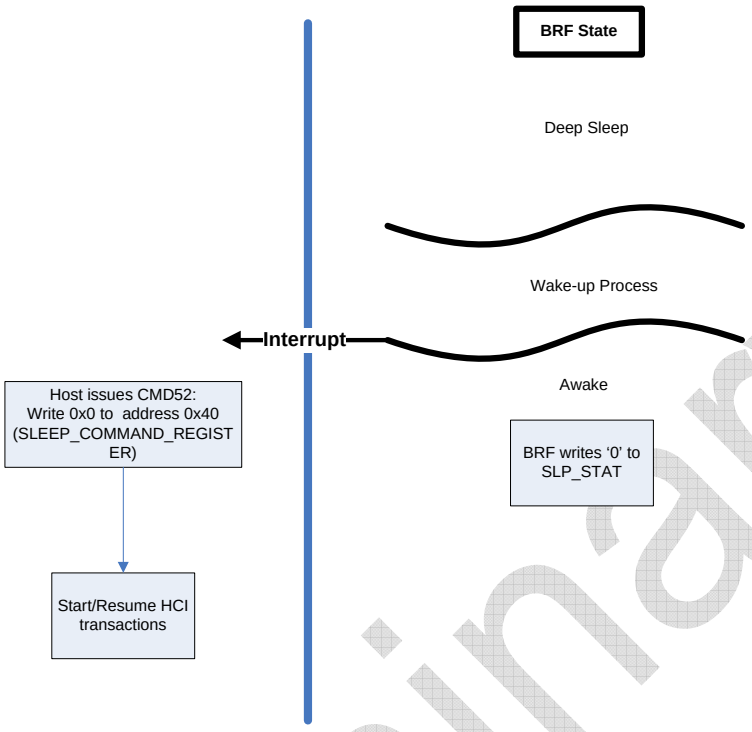

Figure 6 : Device initiates wake-up

4.5 BRF Power Mode Status

The SDIO Host may inquire as to the power mode of the BRF by reading the SLP_STAT bit of the **Bluetooth Sleep State** register. The Host will issue CMD52 that *reads* from address 0x42. Table 6 shows the required command format.

Table 6 : CMD52- SDIO Host reads the BRF power mode status

S	D	Command Index	R/W flag	Function Number	RAW flag	Stuff	Register Address	Stuff	Write Data or Stuff Bits	CRC	E
0	1	110100b	0	001b	0		0x42				1

Note: The host *must* check that the device is awake by reading SLP_STAT bit, as only then it can start an HCI transaction. Whenever the Host suspects the device to be in deep-sleep mode it must poll the SLP_STAT bit until receives a wake-up indication.

When the SLP_STAT bit of the **Bluetooth Sleep State** register is '0' - it indicates the BRF is awake.

When the SLP_STAT bit of the **Bluetooth Sleep State** register is '1' - it indicates the BRF is in deep-sleep mode.

Note: The BRF will update the SLP_STAT bit whenever it exits sleep even when the Host did not initiate it. This will happen whenever the device is in scan activities. In such a case the

SLP_CMD=1 and the SLP_STAT=0. If the Host wishes to interact with the BRF it *must* write '0' to SLP_CMD to prevent the BRF to go back to deep-sleep mode.

4.6 Enabling the SDIO Deep Sleep Protocol

The default setting at power up is that Deep Sleep operation is disabled. In order to enable the BRF Deep Sleep feature and make the Deep Sleep protocol active, the Host must first send an HCI_VS_Set_Sleep_Mode command (see "BRF6300 HCI Vendor Specific Command" document).

Send_HCI_VS_Sleep_Mode_Configurations 0xFD0C, 1, 1, 0x07, 0xFF, 0xFF, 0xFF, 0, 100

Wait_HCI_Command_Complete_VS_Sleep_Mode_Configurations_Event 5000, any, HCI_VS_Sleep_Mode_Configurations, 0x00

4.6.1 HCI_VS_Sleep_Mode_Configurations Command(0xFD0C)

Command	Opcode	Command Parameters	Return Parameters
HCI_VS_Sleep_Mode_Configurations	0xFD0C	Big Sleep Enable Deep Sleep Enable Deep Sleep protocol mode Output IO select (Reserved) Output pull enable Input pull enable Input IO select (Reserved)*	status

Description:
This command configures the sleep mode to be used.
Note that prior to sending this command deep sleep is disabled.

Default values:
Big Sleep is enabled by default
Deep Sleep is disabled by default.

Command Parameters:

Big sleep enable	Size: 1 Byte
Value	**Parameter Description**
0x0	Big sleep is disabled
0x1	Big sleep is enabled

Deep sleep enable	Size: 1 Byte
Value	**Parameter Description**
0x0	Deep sleep is disabled
0x1	Deep sleep is enabled

Deep sleep protocol mode	Size: 1 Byte
Value	**Parameter Description**
0x0	HCILL
0x1	Reserved
0x2	Reserved
0x3	Reserved
0x4	Reserved
0x5	Reserved
0x6	Reserved
0x7	SDIO Protocol
0x8	Reserved

Output IO select	Size: 1 Byte

Value	Parameter Description
0x00 - 0xFF	**Value must be set to 0xFF**, Reserved for future usage.

Output pull Enable	Size: 1 Bytes
Value	**Parameter Description**
0x0	Output pull is disabled
0x1	Output pull is enabled
0xFF	Don't change

Input pull Enable	Size: 1 Bytes
Value	**Parameter Description**
0x0	Input pull is disabled
0x1	Input pull is enabled
0xFF	Don't change

Input IO select	Size: 1 Byte
Value	**Parameter Description**
0x00 - 0xFF	**Value must be set to 0x00**, Reserved for future usage.

Host_Wake deassertion timer	Size: 2 Bytes
Value	**Parameter Description**
0x00	Reserved
0x0001 - 0xFFFF	Reserved

Return Parameters:

Status:	Size: 1 Byte
Value	Parameter Description
0x00 0x01-0xFF	Command Succeeded. Command failed. See Appendix B for HCI error codes.

Events Generated:

Command Complete Event

5 Shared SDIO Bus Mode

5.1 General Description

This section describes TI solution for a system in which the SDIO Host Controller interfaces with Bluetooth (BRF6300) and WLAN (WL1251) devices (and possibly a standard SD card).

The topology is of one SDIO master and an SDIO bus shared by several slaves (one of which might be a legacy SD-card).

Standard SD/SDIO devices are referred to as *legacy* devices. The BT and WLAN devices are commonly named *non-legacy* SDIO devices i.e. they support the shared SDIO functionality that is hereby described.

5.2 Shared SDIO Bus Topology

The bus topology has a single master (HOST) and multiple slaves (SD memory card, WLAN and Bluetooth devices).

The following lines are common to all SDIO devices in the system:

- SDIO_CLK
- SDIO_CMD
- SDIO_Dat0

The Power and Ground signals might be shared as well depending on the electrical characteristics of the I/O cells in the HOST, the BRF6300, and the WL1251.

The shared SDIO bus is depicted in Figure 7.

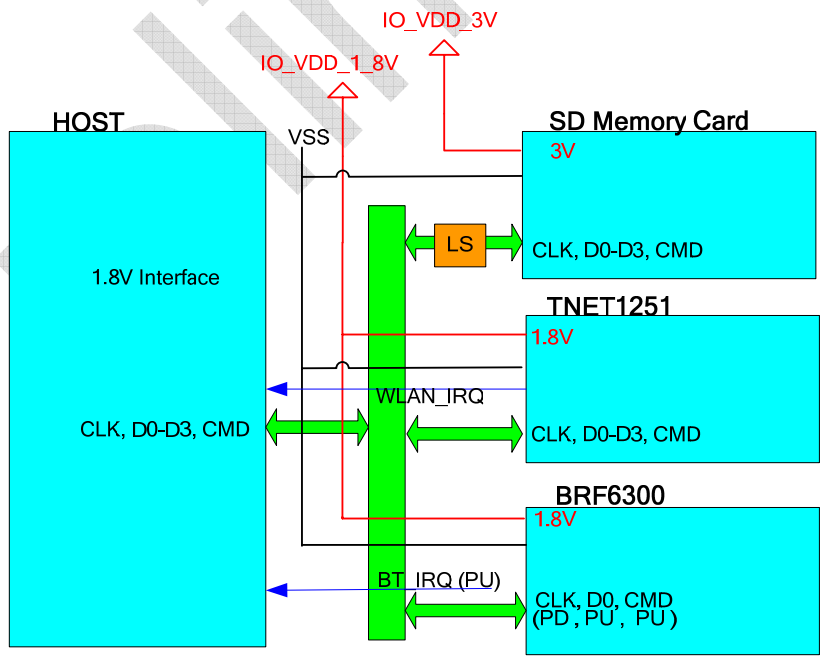

Figure 7 : Shared SDIO Bus Topology

5.3 Command Format

Each function, in each non-legacy device, is assigned a vendor specific command. In this way the separation between devices and addressing the different functions in each device is assured.
Table 7 depicts the format of a *generic* SDIO command.

Table 7 : SDIO Command Format

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

The BRF6300 and WL1251 response to the SDIO packet will be in the regular SDIO format as a legacy device.

All the mandatory commands for SDIO devices directed to a non-legacy device are re-mapped through a vendor specific command. Three bits in the command argument define the command remapping, as shown in Table 8. Commands which are common to all functions are issued through the vendor specific command directed to the appropriate function 0. In this way the non-legacy devices are fully separated from each other, and from the legacy device.
In addition, the SDIO slave in each non-legacy device holds a mechanism for auto-detection between shared and non-shared mode of operation. This will be done upon initialization described in Section 5.4.

Table 8 depicts the SDIO *VS* command format of the BRF6300 and WL1251.

Table 8 : SDIO VS Command Format

Field Name	**Description**	
Start bit	Start Bit *always '0'*	
Transmission Bit	Direction, 1- Indicated Host to Card 0- Indicated Card to Host	
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). The default should be: 60 (TBD) BRF6300 - CIA Function '0' 61 (TBD) BRF6300 - Data Function '1' 62 (TBD) WL1251 - CIA Function '0' 63 (TBD) WL1251 - Data Function '1'	
Argument	Includes all the parameters of the SDIO remapped command. The function number that will be used for indicating the remapped command number according to the following table:	
	SDIO CMD Number (3 bits) Identify the remapped SDIO command:	
	000	CMD0

	001	CMD3
	010	CMD5
	011	CMD7
	100	CMD15
	101	CMD52
	110	CMD53
	111	CMD59
CRC7	Seven bits of CRC Data	
E	Stop bit, *always '1'*	

5.3.1 Examples:

Example #1**:** Sending CMD53 write command to BRF6300 function 1 sent from Host to card:

	S	**D**	**VS Command Index**	**R/W Flag**	**Remapped SDIO CMD Number**	**Block Mode**	**OP Code**	**Register Address**	**Byte/Block Count**	**CRC7**	**E**
#of bits	1	1	6	1	3	1	1	17	9	7	1
value	0	1	111101	1	110	0	0	0xXXXX	000000001	0xXXX	1

Figure 8 – CMD53 to BRF6300 Function-1 Example

Example #2: Sending CMD5 command from Host to WL1251 function 1:

	S	**D**	**VS Command Index**	**Stuff**			**IO OCR**	**CRC7**	**E**
#of bits	1	1	6	8			24	7	1
value					CMD#				
	0	1	111111	0	010	0000	0xXXXXXX	0xXXX	1

Figure 9 – CMD5 from WL1251 Function-1 to the Host Example

Example #3**:** Sending CMD52 read command from BRF6300 function 0 sent from Host to card:

	S	**D**	**VS Command Index**	**R/W Flag**	**Remapped SDIO CMD Number**	**RAW Flag**	**Stuff**	**Register Address**	**Stuff**	**Write Data or Stuff Bits**	**CRC7**	**E**
#of bits	1	1	6	1	3	1	1	17	1	8	7	1
value	0	1	111100	0	101	0	0	0xXXXX	0	00000000	0xXXX	1

Figure 10 – CMD52 to BRF6300 Function-0 Example

5.3.2 Legacy and Non-Legacy Modes Inter-Operability

There are two modes of operation for non-legacy devices:

1) Legacy device – only one device
2) Non-legacy device - shared mode

The SDIO slave module on the chip supports both legacy and non legacy mode by default.

In legacy mode, the device behaves normally and responds to standard SDIO commands.

In non-legacy mode, the device responds only to vendor specific commands as described above. For this purpose a special mechanism is incorporated into the slave device to auto-detect the mode of operation during the initialization phase (refer to Section 5.4).

In order to prevent falsely identifying another device's response as a legal command there are a few mechanism implemented within TI devices. However there are still a more actions that need to be taken by the host.

1. The host must send VS-CMD7 to the non-legacy devices before sending a command that generates a R2 response.
2. When interfacing with a legacy device, the host must de-select the device by using CMD7 before interacting with the non-legacy devices.
3. Before and after data transaction with a non-legacy device TI recommends that the Host will first select and then de-select the device via CMD7. But in cases when higher data rate is of issue CMD7 can be eliminated.

5.4 Initialization Process

The initialization phase is critical for the shared bus to come up successfully.

Note that if the initialization sequence is initiated through vendor-specific commands the device will cease to respond in legacy mode and will act only as a non-legacy device. However, if the initialization is done through CMD5 the device will come up in legacy mode.

In non-legacy mode the initialization process is as follows:

- Issue vendor specific command to first non-legacy device (device #1), with command index 5. This will initialize this device in non-legacy mode.
- Issue vendor specific command to first non-legacy device (device #1), with command index 3 and then with command index 7 and RCA 0 to de-select the device.
- Issue vendor specific command to next non-legacy device (device #2), with command index 5. This will initialize this device in non-legacy mode.
- Issue vendor specific command to next non-legacy device (device #2), with command index 3 and then with command index 7 and RCA 0 to de-select the device.
- Issue CMD5 (sdio-card) or CMD55 (sd-card) on the bus. Only the legacy device will respond thanks to the auto-detect functionality of non-legacy devices. However, in case the legacy device is not from TI, it will respond with the bad CRC bit on because the response R4 to the VS-CMD5 that was sent does not contain a CRC values and the legacy device will mistake it for a valid command.
- Finish initialization procedure of the legacy device.
- Send CMD7 to legacy device to de-select it.
- Now all three devices have been initialized and de-selected.

It is critical to follow the sequence as described in order for the shared bus to come up successfully.

If the non legacy mode is disabled and a VS command has been received the device does not respond to the command. After power up both modes are enabled.

Figure 11 : Initialization and Data Transfer with Legacy and Non-Legacy devicesFigure 11 illustrated the sequence of commands when initializing a legacy and non-legacy device and the command sequence for data transaction for both types of devices.

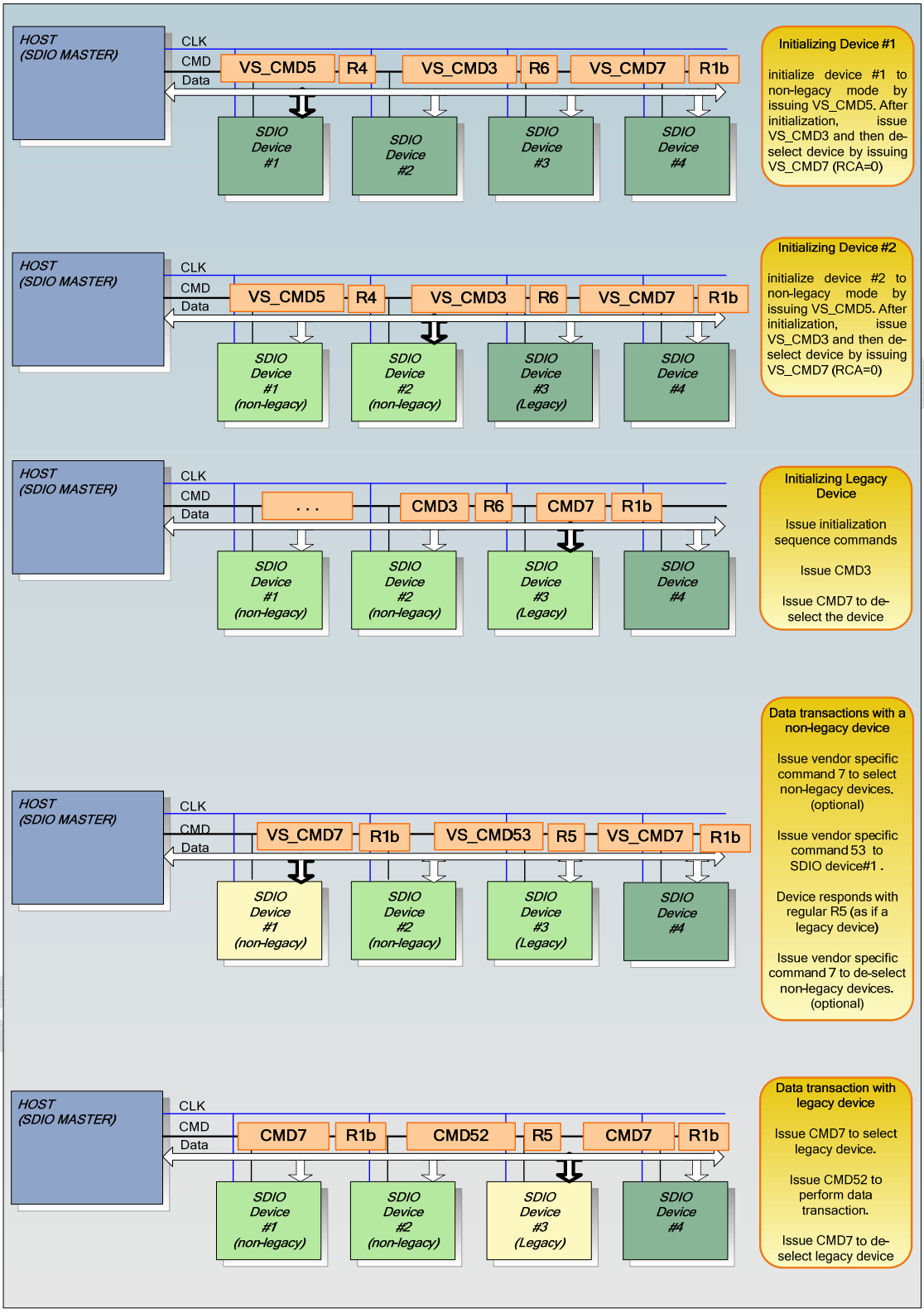

Figure 11 : Initialization and Data Transfer with Legacy and Non-Legacy devices

6 Reference Documents

Document	Reference
SDIO Card Specification ver1.00	SD Card Part E1, September 2002 ,SD Association
Type-A Specification for Bluetooth ver1.00	SDIO Card Part E2, October 2001, SD Association
BRF6300 Product Review Rev 0.41	BT-DS-0023
BRF3000 HCI Vendor Specific Command, Rev 0.4	BT-SW-0029

Appendix A: SDIO VS Commands

I. CMD0

BRF6300 & WL1251 SDIO VS Command 0 Format

	S	D	VS Command Index	Stuff Bits			CRC7	E
#of bits	1	1	6	32 [31:0]			7	1
value				1	3 CMD#	28		

BRF6300 & WL1251 VS Command Fields description for CMD0

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
Stuff - SDIO CMD Number	Bits [30:28] Identify the remapped SDIO command. The rest of bits are '0' As described in SDIO VS Command Description
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

II. CMD3

BRF6300 & WL1251 SDIO VS Command 3 Format

	S	D	VS Command Index	Stuff Bits			CRC7	E
#of bits	1	1	6	32 [31:0]			7	1
value				1	3 CMD#	28		

BRF6300 & WL1251 VS Command Fields description for CMD3

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
Stuff - SDIO CMD Number	Bits [30:28] Identify the remapped SDIO command. The rest of bits are '0' As described in SDIO VS Command Description
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

III. CMD5

BRF6300 & WL1251 SDIO VS Command 5 Format

	S	D	VS Command Index	Stuff			IO OCR	CRC7	E
#of bits	1	1	6	8 [7:0]			24	7	1
value				1	3 CMD#	4			

BRF6300 & WL1251 VS Command Fields description for CMD5

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
Stuff - SDIO CMD Number	Bits [6:4] Identify the remapped SDIO command. The rest of bits are '0' As described in SDIO VS Command Description
IO OCR	Same functionality as described in SDIO Specification
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

Table 9

IV. CMD7

BRF6300 & WL1251 SDIO VS Command 7 Format

	S	D	VS Command Index	RCA (Stuff Bits)			Stuff Bits (RCA)	CRC7	E
#of bits	1	1	6	16 [15:0]			16	7	1
value				1	3 CMD#	12			

BRF6300 & WL1251 VS Command Fields description for CMD7

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
RCA	Will be used as the Stuff bits Bits [14:12] Identify the remapped SDIO command. The rest of bits are '0' As described in SDIO VS Command Description
Stuff - SDIO CMD Number	Will be used as the RCA field
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

V. CMD15

BRF6300 & WL1251 SDIO VS Command 15 Format

	S	D	VS Command Index	RCA (Stuff Bits)			Stuff Bits (RCA)	CRC7	E
#of bits	1	1	6	16 [15:0]			16	7	1
value				1	3 CMD#	12			

BRF6300 & WL1251 VS Command Fields description for CMD15

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
RCA	Will be used as the Stuff bits Bits [14:12] Identify the remapped SDIO command. The rest of bits are '0' As described in SDIO VS Command Description
Stuff - SDIO CMD Number	Will be used as the RCA field
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

VI. CMD52

BRF6300 & WL1251 SDIO VS Command 52 Format

	S	D	VS Command Index	R/W Flag	Remapped SDIO CMD Number	RAW Flag	Stuff	Register Address	Stuff	Write Data or Stuff Bits	CRC7	E
#of bits	1	1	6	1	3	1	1	17	1	8	7	1
value												

BRF6300 & WL1251 VS Command Fields description for CMD52

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
R/W Flag	Same functionality as described in SDIO Specification
SDIO CMD Number	Identify the remapped SDIO command. As described in SDIO VS Command Description
RAW Flag	Same functionality as described in SDIO Specification
Stuff	Same functionality as described in SDIO Specification
Resister Address	Same functionality as described in SDIO Specification
Stuff	Same functionality as described in SDIO Specification
Write Data or Stuff Bits	Same functionality as described in SDIO Specification
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

VII. CMD53

BRF6300 & WL1251 SDIO VS Command 53 Format

	S	D	VS Command Index	R/W Flag	Remapped SDIO CMD Number	Block Mode	OP Code	Register Address	Byte/Block Count	CRC7	E
#of bits	1	1	6	1	3	1	1	17	9	7	1
value											

BRF6300 & WL1251 VS Command Fields description for CMD53

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
R/W Flag	Same functionality as described in SDIO Specification
SDIO CMD Number	Identify the remapped SDIO command. As described in SDIO VS Command Description
Block Mode	Same functionality as described in SDIO Specification
OP Code	Same functionality as described in SDIO Specification
Resister Address	Same functionality as described in SDIO Specification
Byte/Block Count	Same functionality as described in SDIO Specification
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

VIII. CMD59

BRF6300 & WL1251 SDIO VS Command 59 Format

	S	D	VS Command Index	Stuff Bits			CRC7	E
#of bits	1	1	6	32 [31:0]			7	1
value				1	3 CMD#	28		

BRF6300 & WL1251 VS Command Fields description for CMD59

Field Name	Description
S	Same functionality as described in SDIO Specification
D	Same functionality as described in SDIO Specification
VS Command Index	Indicates Vendor Specific Command with the following values (Configurable). As described in SDIO VS Command Description
Stuff - SDIO CMD Number	Bits [30:28] Identify the remapped SDIO command. The rest of bits are '0' As described in SDIO VS Command Description
CRC7	Same functionality as described in SDIO Specification
E	Same functionality as described in SDIO Specification

Appendix B: Code example for fixing the Type-A Header Issue

```
/*
typedef struct _SD_TRANSPORT_HEADER {
   union {
      struct {
         UCHAR   PacketLength[3];
         UCHAR   ServiceID;
      } AsUCHAR;
      ULONG   AsULONG;
   } u;
} SD_TRANSPORT_HEADER, *PSD_TRANSPORT_HEADER;
*/

This is the code AFTER the fix
typedef struct _SD_TRANSPORT_HEADER {
   union {
                  struct {
                        UCHAR   ServiceID;
                        UCHAR   PacketLength[3];
                  } AsUCHAR;
      ULONG   AsULONG;
   } u;
} SD_TRANSPORT_HEADER, *PSD_TRANSPORT_HEADER; //krishna interchanged serviceid and PacketLength
```

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.