SUMMIT
DATA COMMUNICATIONS

Bluetooth

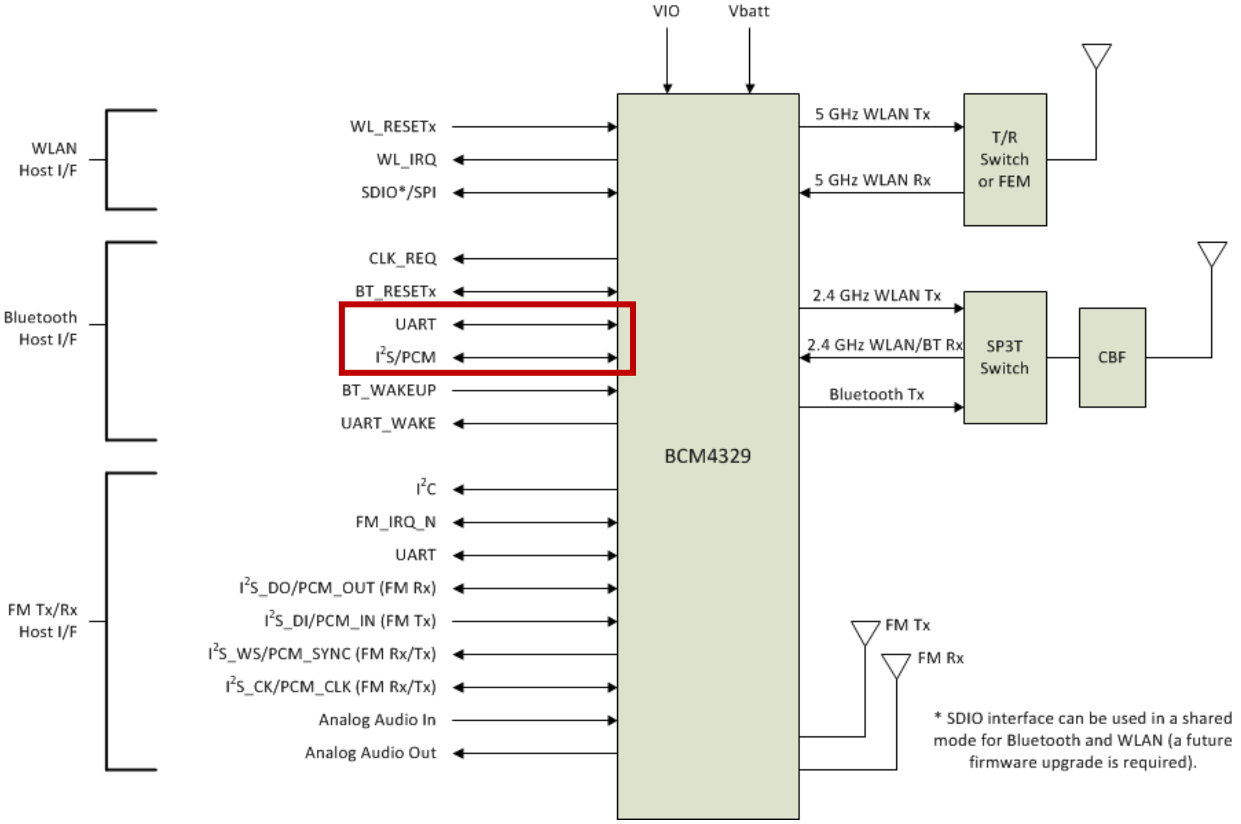
Application Note *April 2012*

Revision History

Version	Revision Date	Change Description
1.0	08/01/11	Initial Release of Version 1.0
1.1		
1.2	12/19/11	Added ***Set_Sleepmode_Param*** and ***Read_Sleepmode_Param*** entries
1.3	04/25/12	Updated Figure 2. Added **Frame Sync** and **Data Formatting** entries Add PCM Default formatting

Contents

Hardware Interface



* SDIO interface can be used in a shared mode for Bluetooth and WLAN (a future firmware upgrade is required).

Bluetooth Specifications

Feature	Description
Bluetooth Interface	Host Controller Interface (HCI) using High Speed UART and PCM for audio data
Extended Synchronous Connections (eSCO)	Supports eSCO for enhanced voice quality by allowing for retransmission of dropped packets.
Bluetooth Media	Frequency Hopping Spread Spectrum (FSSS)
Bluetooth Standards	Bluetooth Core Specification version 2.1 with Enhanced Data Rate (EDR) (up to 3 Mbps) compliant with provisions for supporting future specifications.
Bluetooth Data Rates Supported	1, 2, 3 Mbps
Bluetooth Modulation	GFSK@ 1 Mbps π/4-DQPSK@ 2 Mbps 8-DPSK@ 3 Mbps
Bluetooth Transmitter Operation	Class 1 or Class 2
Compliance ***Note:*** *These regulatory domain certifications are pending.*	**ETSI Regulatory Domain** EN 300 328 v1.7.1 (BT 2.1) **FCC Regulatory Domain** FCC 15.247 DSS – BT 2.1 **Industry Canada** RSS-210 – BT 2.1 **MIC (Japan) Regulatory Domain** (formerly TELEC) Article 2-1 Item 19-2 (BT 2.1)

Note: Users should have UART and PCM interfaces only. HCI commands are recommended for power management.

UART Interface

This physical interface is a standard, four-wire interface that includes the following four signals: RX, TX, RTS, and CTS. It has adjustable baud rates ranging from 9600 bps to 4.0 Mbps and defaults to a baud rate of 115.2 Kbps (out of reset). A high-speed baud rate may be selected by changing the ***baud*** registry key of the appropriate BT stack

Note: Although the Summit module is listed as supporting a baud rate up to 4 Mbps, this ability is dependent on how the module is integrated into the host device. Achieving a 3 Mbps baud rate is a more reasonable goal.

To support EDR, the BCM4329 has a 480-byte Received FIFO (First In, First Out) and a 480-byte Transmit FIFO. The UART interface also supports the Bluetooth 2.1 UART HCI (H4) specification.

Note: The H4 default baud rate is 115.2 kbaud.

To support both high and low baud rates efficiently, the UART clock can be selected as either 24 or 48 MHz. The 48 MHz clock is typically required for baud rates over three Mbaud; the 24 MHz clock can be used to for a more accurate baud rate under three Mbaud.

The UART baud rate is typically set by a configuration record that is downloaded after reset or by automatic baud rate detection (where the host does not need to adjust the baud rate). Support for changing the baud rate during normal HCI UART operation is included through a vendor-specific command that allows the host to adjust the contents of the baud rate registers.

For the UART to operate correctly with the host UART, the combined baud rate error of the two devices must be within two percent.

The following table describes the Bluetooth UART signals:

Ball Pad	Pin Name	Type	Description
M11	BT_UART_CTS_N	I	Clear-to-send signal for the Bluetooth UART interface, active low.
M12	BT_UART_RTS_N	O	Request-to-send signal for the Bluetooth UART interface, active low.
M13	BT_UART_TXD	O	Bluetooth UART Serial Output.
M14	BT_UART_RXD	I	Bluetooth UART Serial Input.

Figure 1: Bluetooth UART signals

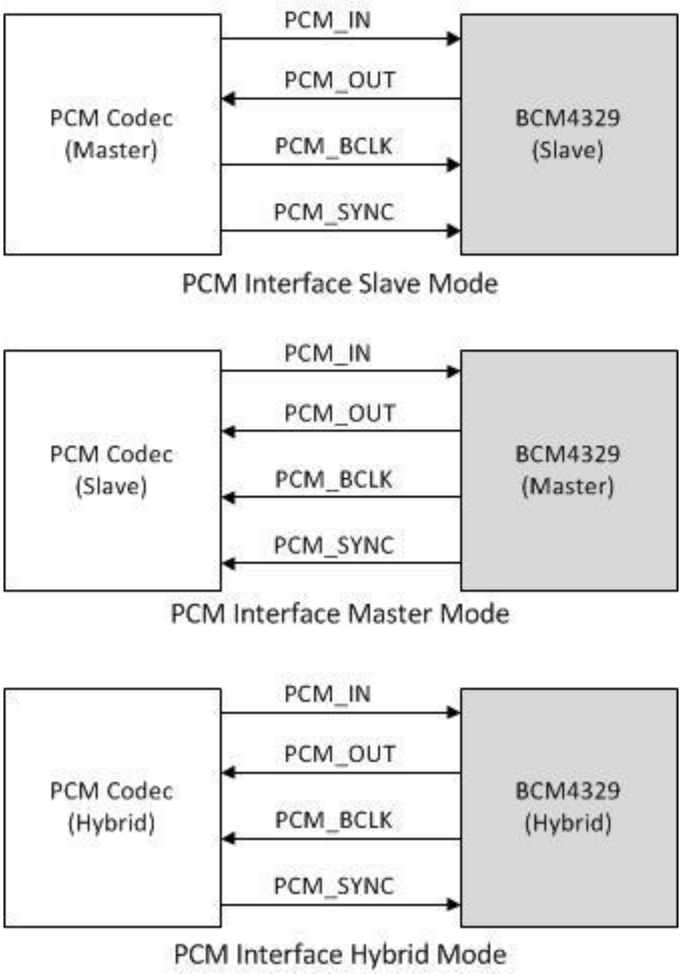
PCM Interface (Bluetooth and SCO Audio)

The PCM (Pulse Code Modulation) Interface can connect to linear PCM Codec devices in master or slave mode:

- Master mode – The BCM4329 generates the PCM_CLK and PCM_SYNC signals
- Slave mode – PCM_CLK and PCM_SYNC signals are provided by another master on the PCM interface and are inputs to the BCM4329.

The BCM4329 supports up to three SCO or eSCO channels through the PCM interface; each channel can be independently mapped to any of the available slots in a frame.

The configuration of the PCM interface may be adjusted by the host through the use of vendor-specific HCI commands.



PCM Interface Slave Mode

PCM Interface Master Mode

PCM Interface Hybrid Mode

Figure 2: BCM4329 PCM Interface with Linear PCM Codec

Frame Sync

The BCM2349 supports both short and long frame sync types in both master and slave configurations. In the short frame sync mode, the frame sync signal is an active-high pulse at the 8-kHz audio frame rate that is a single-bit period in width and synchronized to the rising edge of the bit clock. The PCM slave looks for a high on the failing edge of the bit clock and expects the first bit of the first slot to start at the next rising edge of the clock. In the long frame sync mode, the frame sync signal is again an active-high pulse at the 8-kHz audio frame rate; however, the duration is three bit periods and the pulse starts coincident with the first bit of the first slot.

Data Formatting

The BCM4329 may be configured to generate and accept several different data formats. The BCM4329 uses 13 of the 16 bits in each PCM frame. The location and order of these 13 bits is configurable to support various data formats on the PCM interface. The remaining three bits are ignored on the input, and may be filled with 0s, 1s, sign bit, or a programmed value on the output. The default format is 13-bit 2's complement data, left-justified, and clocked MSB first.

Host Controller Power Management

When in UART mode, the BCM4329 can be configured so that dedicated signals are used for power management hand-shaking between the BCM4329 and the host. The hand-shaking signals support basic power saving functions including the standard Bluetooth defined power savings modes and standby modes of operation (see Table 3 below).

Pin Name	Mapped to Pin	I/O	Description
BT_WAKE_B	BT_GPIO_0	I	Bluetooth device wake-up: Signal from the host to the SDC-SSD40NBT indicating that the host requires attention. Asserted: Bluetooth device must wake-up or remain awake Deasserted: Bluetooth device may sleep when sleep criteria are met The polarity of this signal is software configurable and can be asserted high or low.
BT_HOST_WAKE_B	BT_GPIO_1	O	Host Wake-up Signal from the SDC-SSD40NBT to the host indicating that the radio requires attention. Asserted: Host device must wake-up or remain awake. Deasserted: Host device may sleep when sleep criteria are met The polarity of this signal is software configurable and can be asserted high or low.

Figure 3: Power Control Pin Definitions

BT Stacks

With our Bluetooth module, Summit recommends that you use our Summit BT stack (although you may use another stack such as the MSFT BT stack). Because the initial Summit release does not include the Summit BT stack, refer to the following document for assistance with integrating the MSFT stack:

Platform Builder OS Design Template.pdf

Updating Firmware

If the firmware is not updated, you may see basic connection problems such as non-returned inquiries or no neighboring BT devices returned although they are present.

The firmware is stored in a HCD file in the Summit CAB file. Summit software automatically updates the firmware file from the Summit sdc_gina.exe file. Because of this, you may notice a 6-10 second delay after a warm boot, suspend, and resume.

BT MAC Address

After the firmware update is complete, our sdc_gina.exe file writes the BD_ADDR. BD_ADDR is the BT MAC address. It is always the Wi-Fi MAC address plus one.

Note: If you install Summit files from the Summit CAB file, you do not need to worry about updating the firmware or BT MAC address.
However, if you are interested in doing this yourself, you can call the HCI commands directly to update firmware and to write the BT MAC address. See the *Summit HCI Commands* section for additional information.
For the BD_ADDR: *Write_BD_ADDR*.
For the firmware update: *Download_Minidriver* and *WriteRAM*.

Summit HCI Commands

FCC Testing

Super_Peek_Poke 0xFC0A

Allows the RF registers and memory space (XDATA space for 8051-based chips) to be read or written. Parameters are dependent on what memory space is being read or written and whether the access is a read or a write.

Command Parameters

Access_Type

Type	uint8
Size	1 byte
Description	Selects which memory or register space type is to be read or written (and whether it is to be read or written).
Values ***Note:*** *May not be combined bitwise.*	0x0 – 8051_XDATA_Peek 0x1 – BlueRF_Peek 0x2 – BlueRF_Poke 0x3 – 8051_XDATA_Poke 0x4 – ARM_Memory_Peek 0x5 – ARM_Memory_Poke 0x6 – PMU_Peek 0x7 – PMU_Poke

XDATA_Address

Type	uint16 (big endian)
Size	2 bytes (if present)
Description	Present for XDATA reads and writes on 8051-based chips. Specifies the XDATA address to be read or written.

BlueRF_Address

Type	uint8
Size	1 byte (if present)
Description	Specifies the RF register address to be read or written.

ARM_Memory_Address

Type	uint32 (little endian)
Size	4 bytes (if present)
Description	Present for memory reads and writes on ARM-based chips. Specifies the memory address to be read or written. For writes, the memory address must fall on a 32-bit address boundary.

PMU_Reg_Offset

Type	uint8
Size	1 byte (if present)
Description	Present for PMU register reads and writes. Specifies the PMU register address to be read or written.

Value8

Type	uint8
Size	1 byte (if present)
Description	Present for XDATA writes on 8051-based chips, or RF or PMU register writes on all chips. Specifies the value to be written.

Value32

Type	uint32 (little endian)
Size	4 bytes (if present)
Description	Present for memory writes on ARM-based chips. Specifies the value to be written.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

XDATA_Address

Type	uint8
Size	1 byte
Description	The value read; only valid if the access type was a read.

Set_Tx_Carrier_Frequency_ARM 0xFC14

Puts the radio into manual mode for Tx tests.

Command Parameters

Carrier_Enable

Type	uint8
Size	1 byte
Description	Used to turn the carrier on and off.
Values ***Note:*** *May not be combined bitwise.*	0x0 – Carrier on 0x1 – Carrier off

Carrier_Frequency_Encoded

Type	uint8
Size	1 byte
Description	If Carrier_Enable is 0, Bluetooth frequency minus 2400

Carrier_Frequency

Type	uint16 (little endian)
Size	0 byte (abstract)
Description	An abstraction of Carrier_Frequency_Encoded as Bluetooth frequency in Mhz.

Mode

Type	uint8
Size	1 byte
Description	If Carrier_Enable is 0, selects unmodulated or modulated with pattern.
Values ***Note:*** *May not be combined bitwise.*	0x0 – Unmodulated 0x1 – PRBS9 0x2 – PRBS15 0x3 – All Zeros 0x4 – All Ones 0x5 – Incrementing Symbols

Modulation Type

Type	uint8
Size	1 byte
Description	If Carrier_Enable is 0, 1 Mbps, 2 Mbps, or 3 Mbps selection, ignored if Mode is Unmodulated.
Values ***Note:*** *May not be combined bitwise.*	0x0 GFSK 0x1 QPSK 0x2 8PSK

Transmit_Power

Type	uint8
Size	1 byte
Description	If Carrier_Enable is 0, requested power table index.
Values ***Note:*** *May not be combined bitwise.*	0x0 0 dBm 0x1 -4 dBm 0x2 -8 dBm 0x3 -12 dBm 0x4 -16 dBm 0x5 -20 dBm 0x6 -24 dBm 0x7 -28 dBm 0x8 Specify Power in dBm 0x9 Specify Power Table index

Transmit_Power_dBm

Type	int8
Size	1 byte
Description	If Carrier_Enable is 0 and Transmit_Power is Integer override, requested table index.

Transmit_Power_Table_Index

Type	uint8
Size	1 byte
Description	If Carrier_Enable is 0 and Transmit_Power is Table index override, requested table index.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Tx_Test 0xFC51

Connectionless transmit test to send Bluetooth packets.

Command Parameters

Local_Device_BD_ADDR

Type	Bluetooth device address
Size	6 bytes
Description	BD_ADDR for this device.

Hopping_Mode

<table>
<tr><td align="right"><b>Type</b></td><td colspan="2">uint8</td></tr>
<tr><td align="right"><b>Size</b></td><td colspan="2">1 byte</td></tr>
<tr><td align="right"><b>Description</b></td><td colspan="2">Sets the hopping mode. The Fixed pattern hopping mode transmits a repeating pattern.</td></tr>
<tr><td align="right" rowspan="3"><b>Values</b>
<b><i>Note:</i></b> <i>May not be combined bitwise.</i></td><td>0x0</td><td>79 channel</td></tr>
<tr><td>0x1</td><td>Single frequency</td></tr>
<tr><td>0x2</td><td>Fixed pattern</td></tr>
</table>

Frequency

Type	uint8					
Size	1 byte					
Description	When Hopping Mode is Single Frequency, this parameter sets the frequency.					
Values ***Note:*** *May not be combined bitwise.*	0x00	2402 MHz	0x1B	2429 MHz	0x36	2456 MHz
	0x01	2403 MHz	0x1C	2430 MHz	0x37	2457 MHz
	0x02	2404 MHz	0x1D	2431 MHz	0x38	2458 MHz
	0x03	2405 MHz	0x1E	2432 MHz	0x39	2459 MHz
	0x04	2406 MHz	0x1F	2433 MHz	0x3A	2460 MHz
	0x05	2407 MHz	0x20	2434 MHz	0x3B	2461 MHz
	0x06	2408 MHz	0x21	2435 MHz	0x3C	2462 MHz
	0x07	2409 MHz	0x22	2436 MHz	0x3D	2463 MHz
	0x08	2410 MHz	0x23	2437 MHz	0x3E	2464 MHz
	0x09	2411 MHz	0x24	2438 MHz	0x3F	2465 MHz
	0x0A	2412 MHz	0x25	2439 MHz	0x40	2466 MHz
	0x0B	2413 MHz	0x26	2440 MHz	0x41	2467 MHz
	0x0C	2414 MHz	0x27	2441 MHz	0x42	2468 MHz
	0x0D	2415 MHz	0x28	2442 MHz	0x43	2469 MHz
	0x0E	2416 MHz	0x29	2443 MHz	0x44	2470 MHz
	0x0F	2417 MHz	0x2A	2444 MHz	0x45	2471 MHz
	0x10	2418 MHz	0x2B	2445 MHz	0x46	2472 MHz
	0x11	2419 MHz	0x2C	2446 MHz	0x47	2473 MHz
	0x12	2420 MHz	0x2D	2447 MHz	0x48	2474 MHz
	0x13	2421 MHz	0x2E	2448 MHz	0x49	2475 MHz
	0x14	2422 MHz	0x2F	2449 MHz	0x4A	2476 MHz
	0x15	2423 MHz	0x30	2450 MHz	0x4B	2477 MHz
	0x16	2424 MHz	0x31	2451 MHz	0x4C	2478 MHz
	0x17	2425 MHz	0x32	2452 MHz	0x4D	2479 MHz
	0x18	2426 MHz	0x33	2453 MHz	0x4E	2480 MHz
	0x19	2427 MHz	0x34	2454 MHz		
	0x1A	2428 MHz	0x35	2455 MHz		

Modulation_Type

Type	uint8	
Size	1 byte	
Description	Sets the data pattern that is continuously transmitted.	
Values ***Note:*** *May not be combined bitwise.*	0x1	0x00 8-bit Pattern
	0x2	0xFF 8-bit Pattern
	0x3	0xAA 8-bit Pattern
	0x9	0xF0 8-bit Pattern
	0x4	PRBS9 Pattern

Logical_Channel

Type	uint8	
Size	1 byte	
Description	Sets the logical channel. Only ACL Basic is supported if the Hopping_Mode is Fixed pattern.	
Values ***Note:*** *May not be combined bitwise.*	0x0	ACL EDR
	0x1	ACL Basic
	0x2	eSCO EDR
	0x3	eSCO Basic
	0x4	SCO Basic

BB_Packet_Type

Type	uint8	
Size	1 byte	
Description	Sets the packet type. Only DH1/2-DH1 is supported if the Hopping_Mode is Fixed pattern.	
Values ***Note:*** *May not be combined bitwise.*	0x0	NULL
	0x1	POLL
	0x2	FHS
	0x3	DM1
	0x4	DH1/2-DH1
	0x5	HV1
	0x6	HV2/2-EV3

	0x7	HV3/EV3/3-EV3
	0x8	DV/3-DH1
	0x9	AUX1/PS
	0xA	DM3/2-DH3
	0xB	DH3/3-DH3
	0xC	EV4/2-EV5
	0xD	EV5/3-EV5
	0xE	DM5/2-DH5
	0xF	DH5/3-DH5

BB_Packet_Length

Type	uint16 (little endian)
Size	2 bytes
Description	The length in bytes for the BB Packet type selected.

Tx_Power_Level

Type	uint16 (little endian)	
Size	1 byte	
Description	Sets the Tx output power in dbm.	
Values ***Note:*** *May not be combined bitwise.*	0x0	0 dBm
	0x1	-4 dBm
	0x2	-8 dBm
	0x3	-12 dBm
	0x4	-16 dBm
	0x5	-20 dBm
	0x6	-24 dBm
	0x7	-28 dBm
	0x8	Specify Power in dBm
	0x9	Specify Power Table index

Transmit_Power_dBm

Type	int8 (little endian)
Size	1 byte
Description	If Tx_Power_Level is Integer override, requested output power in dBm.

Transmit_Power_Table_Index

Type	uint8 (little endian)
Size	1 byte
Description	If Tx_Power_Level is Table index override, requested table index.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_Receive_Only 0xFC2B

This command sets receive frequency. Used for RF testing only.

Command Parameters

Receive_Frequency_Encoded

Type	uint8
Size	1 byte
Description	This is a Byte value indicating the frequency, which the receiver camps on as an offset in MHz from 2400 MHz. Valid values range from 2 to 80.

Receive_Frequency

Type	uint16
Size	0 bytes (abstract)
Description	An abstraction of Receive_Frequency_Encoded to indicate the actual frequency, from 2402 to 2480 MHz to simplify use of the command from scripts or the user interface.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Patchram

Download_Minidriver 0xFC2E

This command triggers the device to reboot into a state where it is prepared to receive the download of a minidriver.

Command Parameters

none

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_RAM 0xFC4C

This command writes data into the ARM's 32-bit linear address space or EEPROM. It is primarily intended for use when the device has received a Download_Minidriver vendor-specific HCI command, placing it into download mode. A minidriver would typically be downloaded to RAM to facilitate reading or writing firmware and/or config data, and then the minidriver would receive Write_RAM commands to write to flash or EEPROM. When receiving configuration data to RAM, a minidriver is unnecessary and Write_RAM commands containing the configuration data are typically issue immediately after the Download_Minidriver command.

Command Parameters

Address

Type	uint32 (little endian)
Size	4 bytes
Description	This indicates the address to be written to. Addresses greater than or equal to 0xFF000000 represent a virtual address in an EEPROM, with address 0xFF000000 corresponding to address zero in the EEPROM.

Data

Type	uint8 arrary
Size	Up to 251 bytes
Description	The data to write to the target address.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Launch_RAM 0xFC4E

Jumps into the target address in thumb mode, typically a minidriver entry point, or in the case of an 0xFFFFFFFF target address, an implied Bluetooth mode reentry vector with acceptance of runtime mode (having received a Download_Minidriver vendor-specific HCI command) into a minidriver which has just been received, to reset the device by jumping to the reset vector when the minidriver is no longer needed, or in the case of an 0xFFFFFFFF target address, to transition from download mode back to bluetooth mode with acceptance of configuration data which was downloaded by Write_RAM vendor-specific HCI commands.

Command Parameters

Address

Type	uint32 (little endian)
Size	4 bytes
Description	This indicates the address to be written to. A value of 0xFFFFFFFF indicates that a prior series of Write_RAM HCI commands contained a runtime RAM configuration data image, and that the firmware should reboot into Bluetooth mode, using that configuration data.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

BD_ADDR Writing

Write_BD_ADDR 0xFC01

This command writes the value for the device's Bluetooth device address.

Command Parameters

BD_ADDR

Type	Bluetooth device address
Size	6 bytes
Description	The Bluetooth device address.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Edit_BD_ADDR 0x1009

This command writes the value for the device's Bluetooth device address.

Command Parameters

Read_BD_ADDR

Type	None
Size	None
Description	None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte
Type	Bluetooth device address
Size	6 bytes

PCM

Write_SCO_PCM_Int_Param 0xFC1C

This command writes SCO and PCM interface parameters.

Command Parameters

SCO_Routing

Type	uint8	
Size	1 byte	
Description	Specifies whether the SCO path is through PCM interface or transport.	
Values ***Note:*** *May not be combined bitwise.*	0x0	PCM
	0x1	Transport
	0x2	CODEC
	0x3	I2S

PCM_Interface_Rate

Type	uint8	
Size	1 byte	
Description	Specifies the PCM clock frequency.	
Values ***Note:*** *May not be combined bitwise.*	0x0	128 KBps
	0x1	256 KBps
	0x2	512 KBps
	0x3	1024 KBps
	0x4	2048 KBps

Frame_Type

Type	uint8	
Size	1 byte	
Description	Specifies the PCM frame type: short frame or long frame	
Values ***Note:*** *May not be combined bitwise.*	0x0	Short
	0x1	Long

Sync_Mode

Type	uint8
Size	1 byte
Description	Specifies whether Bluetooth module is to be the master or slave for PCM_SYNC signal.
Values ***Note:*** *May not be combined bitwise.*	0x0 Slave 0x1 Master

Clock_Mode

Type	uint8
Size	1 byte
Description	Specifies whether Bluetooth module is to be the master or slave for PCM_CLK signal.
Values ***Note:*** *May not be combined bitwise.*	0x0 Slave 0x1 Master

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_SCO_PCM_Int_Param 0xFC1D

This command reads SCO and PCM interface parameters.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_PCM_Data_Format_Param 0xFC1E

This command writes PCM data format parameters.

Command Parameters

LSB_First

Type	boolean
Size	1 byte
Description	Specifies whether or not the LSB bit is shifted out first for serial PCM data.

Fill_Bits

Type	uint8
Size	1 byte
Description	Specifies the value with which to fill unused bits if Fill_Method is set to programmable.

Fill_Method

Type	uint8
Size	1 byte
Description	Specifies the method of filling unused data bits.
Values **Note:** *May not be combined bitwise.*	0x0 0’s 0x1 1’s 0x2 Signed 0x3 Programmable

Fill_Num

Type	uint8
Size	1 byte
Description	Specifies the number of bits to be filled.

Right_Justify

Type	boolean
Size	1 byte
Description	Specifies the data to right justify (fill data shifted out first) or not. Default is left justify (fill data shifted out last).

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_PCM_Data_Format_Param 0xFC1F

This command reads PCM data format parameters.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

LSB_First

Type	boolean
Size	1 byte
Description	Indicates whether or not the LSB bit is shifted out first for serial PCM data.

Fill_Bits

Type	uint8
Size	1 byte
Description	Specifies the value with which to fill unused bits if Fill_Method is set programmable.

Fill_Method

Type	uint8	
Size	1 byte	
Description	Indicates the method of filling unused data bits.	
Values ***Note:*** *May not be combined bitwise.*	0x0	0’s
	0x1	1’s
	0x2	Signed
	0x3	Programmable

Fill_Num

Type	uint8
Size	1 byte
Description	Indicates the number of bits to be filled.

Right_Justify

Type	boolean
Size	1 byte
Description	Indicates the data to right justify (fill data shifted out first) or not. Default is left justify (fill data shifted out last).

Read_PCM_Loopback_Mode 0xFC25

This command reads PCM Loopback Mode.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

PCM_Data_Dump_Enable

Type	unsigned integer	
Size	7 bits {7:1} in uint8	
Description	PCM data dump vendor-specific debug events enable/disable.	
Values ***Note:*** *May not be combined bitwise.*	0x00	PCM data dump disable
	0x40	POM data dump enable

PCM_Loopback_Mode

Type	boolean
Size	1 bit {0} in uint8
Description	Loopback on/off selection.

Values ***Note:*** *May not be combined bitwise.*	0 Loopback off 1 Loopback on

Write_PCM_Mute_Param 0xFC5E

This command writes the PCM Mute settings.

Command Parameters

Mute_Settings

Type	uint8
Size	1 byte
Description	PCM Mute settings.
Values ***Note:*** *May not be combined bitwise.*	0x1 Mute the From_PCM_To_SCO Direction 0x2 Mute the From_SCO_To_PCM Direction

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_PCM_Mute_Param 0xFC5F

This command reads the PCM Mute settings.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Mute_Settings

Type	uint8	
Size	1 byte	
Description	PCM Mute settings.	
Values	0x1	Mute the From_PCM_To_SCO Direction
Note: *May not be combined bitwise.*	0x2	Mute the From_SCO_To_PCM Direction

Write_I2SPCM_Interface_Param 0xFC6D

This command configures the I2S/PCM interface.

Command Parameters

I2S_Enable

Type	uint8	
Size	1 byte	
Description	Turn on or off the I2S/PCM interface.	
Values	0x0	Disable
Note: *May not be combined bitwise.*	0x1	Enable

Is_Master

Type	uint8	
Size	1 byte	
Description	Select master/slave.	
Values	0x0	Slave
Note: *May not be combined bitwise.*	0x1	Master

Sample_Rate

Type	uint8	
Size	1 byte	
Description	Sample rate.	
Values	0x0	8 KHz
Note: *May not be combined bitwise.*	0x1	16 KHz
	0x2	4 KHz

Clock_Rate

Type	uint8	
Size	1 byte	
Description	I2S Interface Clock Rate	
Values ***Note:*** *May not be combined bitwise.*	0x0	128 KHz
	0x1	256 KHz
	0x2	512 KHz
	0x3	1024 KHz
	0x4	2048 KHz

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_SCO_PCM_Int_Param 0xFC1C

This command writes SCO and PCM interface parameters.

Command Parameters

SCO_Routing

Type	uint8	
Size	1 byte	
Description	Specifies whether the SCO path is through PCM interface or transport.	
Values ***Note:*** *May not be combined bitwise.*	0x0	PCM
	0x1	Transport
	0x2	CODEC
	0x3	I2S

PCM_Interface_Rate

Type	uint8
Size	1 byte
Description	Specifies the PCM clock frequency.

Values ***Note:*** *May not be combined bitwise.*	0x0	128 KBps
	0x1	256 KBps
	0x2	512 KBps
	0x3	1024 KBps
	0x4	2048 KBps

Frame_Type

Type	uint8	
Size	1 byte	
Description	Specifies the PCM frame type: short frame or long frame.	
Values ***Note:*** *May not be combined bitwise.*	0x0	Short
	0x1	Long

Sync_Mode

Type	uint8	
Size	1 byte	
Description	Specifies whether Bluetooth module is the master or slave for PCM_SYNC signal.	
Values ***Note:*** *May not be combined bitwise.*	0x0	Slave
	0x1	Master

Clock_Mode

Type	uint8	
Size	1 byte	
Description	Specifies whether Bluetooth module is the master or slave for PCM_CLK signal.	
Values ***Note:*** *May not be combined bitwise.*	0x0	Slave
	0x1	Master

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_SCO_PCM_Int_Param 0xFC1D

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

SCO_Routing

Type	uint8	
Size	1 byte	
Description	Indicates whether the SCO path is through PCM interface or transport.	
Values ***Note:*** *May not be combined bitwise.*	0x0	PCM
	0x1	Transport
	0x2	CODEC
	0x3	I2S

PCM_Interface_Rate

Type	uint8	
Size	1 byte	
Description	Indicates the PCM clock frequency.	
Values ***Note:*** *May not be combined bitwise.*	0x0	128 KBps
	0x1	256 KBps
	0x2	512 KBps
	0x3	1024 KBps
	0x4	2048 KBps

Frame_Type

Type	uint8
Size	1 byte
Description	Indicates the PCM frame type: short frame or long frame.
Values ***Note:*** *May not be combined bitwise.*	0x0 Short 0x1 Long

Sync_Mode

Type	uint8
Size	1 byte
Description	Indicates whether the Bluetooth module is the master or slave for PCM_SYNC signal.
Values ***Note:*** *May not be combined bitwise.*	0x0 Slave 0x1 Master

Clock_Mode

Type	uint8
Size	1 byte
Description	Indicates whether the Bluetooth module is the master or slave for PCM_CLK signal.
Values ***Note:*** *May not be combined bitwise.*	0x0 Slave 0x1 Master

Write_PCM_Data_Format_Param 0xFC1E

This command writes PCM data format parameters.

Command Parameters

LSB_First

Type	boolean
Size	1 byte
Description	Specifies whether or not the LSB bit is shifted out first for serial PCM data.

Fill_Bits

Type	uint8
Size	1 byte
Description	Specifies the value with which to fill unused bits if Fill_Method is set to programmable.

Fill_Method

Type	uint8
Size	1 byte
Description	Specifies the method of filling unused data bits.
Values ***Note:*** *May not be combined bitwise.*	0x0 0’s 0x1 1’s 0x2 Signed 0x3 Programmable

Fill_Num

Type	uint8
Size	1 byte
Description	Specifies the number of bits to be filled.

Right_Justify

Type	boolean
Size	1 byte
Description	Specifies the data to right justify (fill data shifted out first) or not. Default is left justify (fill data shifted out last).

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_PCM_Data_Format_Param **0xFC1F**

This command reads PCM data format parameters.

Command Parameters

None

Type	boolean
Size	1 byte
Description	Specifies whether or not the LSB bit is shifted out first for serial PCM data.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

LSB_First

Type	boolean
Size	1 byte
Description	Specifies whether or not the LSB bit is shifted out first for serial PCM data.

Fill_Bits

Type	uint8
Size	1 byte
Description	Specifies the value with which to fill unused bits if Fill_Method is set to programmable.

Fill_Method

Type	uint8
Size	1 byte
Description	Specifies the method of filling unused data bits.
Values ***Note:*** *May not be combined bitwise.*	0x0 0’s 0x1 1’s 0x2 Signed 0x3 Programmable

Fill_Num

Type	uint8
Size	1 byte
Description	Indicates the number of bits to be filled.

Right_Justify

Type	boolean
Size	1 byte
Description	Indicates the data to right justify (fill data shifted out first) or not. Default is left justify (fill data shifted out last).

Write_PCM_Loopback_Mode 0xFC24

This command writes PCM Loopback Mode.

Command Parameters

PCM_Data_Dump_Enable

Type	unsigned integer
Size	7 bits {7:1} in uint8
Description	PCM data dump vendor_specific debug events enable/disable.
Values ***Note:*** *May not be combined bitwise.*	0x00 PCM data dump disable 0x40 POM data dump enable

PCM_Loopback_Mode

Type	boolean
Size	1 bit {0} in uint8
Description	Loopback on/off selection.
Values	0 Loopback off 1 Loopback on

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_PCM_Loopback_Mode 0xFC25

This command reads PCM Loopback Mode.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

PCM_Data_Dump_Enable

Type	unsigned integer
Size	7 bits {7:1} in uint8
Description	PCM data dump vendor_specific debug events enable/disable.
Values ***Note:*** *May not be combined bitwise.*	0x00 PCM data dump disable 0x40 POM data dump enable

PCM_Loopback_Mode

Type	boolean
Size	1 bit {0} in uint8
Description	Loopback on/off selection.
Values.	0 Loopback off 1 Loopback on

Write_PCM_Mute_Param 0xFC5E

This command writes the PCM Mute settings.

Command Parameters

Mute_Settings

Type	uint8
Size	1 byte

Description	PCM Mute settings.
Values ***Note:*** *May be combined bitwise.*	0x1 — Mute the From_PCM_To_SCO Direction 0x2 — Mute the From_SCO_To_PCM Direction

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_PCM_Mute_Param 0xFC5F

This command reads the PCM Mute settings.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Mute_Settings

Type	uint8
Size	1 byte
Description	PCM Mute settings.
Values ***Note:*** *May be combined bitwise.*	0x1 — Mute the From_PCM_To_SCO Direction 0x2 — Mute the From_SCO_To_PCM Direction

Write_I2SPCM_Interface_Param 0xFC6D

This command configures the I2S/PCM interface.

Command Parameters

I2S_Enable

Type	uint8
Size	1 byte

Description	Turn on or off the I2S/PCM interface.	
Values ***Note:*** *May not be combined bitwise.*	0x0	Disable
	0x1	Enable

Is_Master

Type	uint8	
Size	1 byte	
Description	Select master/slave.	
Values ***Note:*** *May not be combined bitwise.*	0x0	Slave
	0x1	Master

Sample_Rate

Type	uint8	
Size	1 byte	
Description	Sample rate.	
Values ***Note:*** *May not be combined bitwise.*	0x0	8 KHz
	0x1	16 KHz
	0x2	4 KHz

Clock_Rate

Type	uint8	
Size	1 byte	
Description	I2S Interface Clock Rate.	
Values ***Note:*** *May not be combined bitwise.*	0x0	128 KHz
	0x1	256 KHz
	0x2	512 KHz
	0x3	1024 KHz
	0x4	2048 KHz

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Voice

Update_UART_Baud_Rate 0xFC18

This command changes the baud rate at which the UART transport communicates. A Command Complete event will be generated at the previous baud rate. If the Command Complete event indicates success, the UART transport will then shift to the new baud rate.

Command Parameters

Encoded_Baud_Rate

Type	uint16 (little endian)
Size	2 bytes
Description	The hardware register representation of the new baud rate. The value is determined by a formula described below. For the BCM2045, the value of Encoded_Baud_Rate can be 0x0000, which implies that the encoded form is not used and the actual baud rate follows. When using the UART transport of the BCM2045 in three-wire SLIP (H5) mode, if Update_UART_Baud_Rate is issued, use of the encoded form is disallowed. If the encoded form is used, the value is determined as follows:

```
// Upper four bits
uint32 encoded_segment = ( ( (24000000 / baud_rate) % 16 ) / 2 );
encoded_baud_rate = ( ( (uint16)encoded_segment ) << 4 );

// Lower four bits of high nibble
encoded_segment = ( ( ( (24000000 / baud_rate) % 16 ) / 2 ) +
( ( (24000000 / baud_rate) % 16 ) % 2 ) );
encoded_baud_rate |= (uint16)encoded_segment;

// Lower byte
encoded_segment = ( 256 - ( (24000000 / baud_rate) / 16 ) );
encoded_baud_rate |= (encoded_segment << 8);
```

Note: The above formula is only based on 24 MHz UART clock.

Use_Encoded_Form

Type	boolean
Size	0 bytes (abstract)
Description	An abstraction of whether the encoded form is to be used. If not using the encoded form of baud rate, the Encoded_Baud_Rate parameter is 0x0000 and a four-byte baud rate (in integer form) follows in the Explicit_Baud_Rate parameter.

Explicit_Baud_Rate

Type	uint32 (big endian)
Size	4 bytes (if present)
Description	If Encoded_Baud_Rate is 0x0000 and therefore the encoded form of baud rate is not being used, the Explicit_Baud_Rate parameter will be present, containing the baud rate in integer form.

Baud_Rate

Type	uint32 (big endian)
Size	0 bytes (abstract)
Description	An abstraction of the baud rate, whether present in the command in encoded form in the Encoded_Baud_Rate or included as Explicit_Baud_Rate.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_SCO_Time_Slot 0xFC22

This command writes SCO time slot for the next SCO connection.

Command Parameters

SCO_Time_Slot

Type	uint8
Size	1 byte
Description	TS Assignment, 4 bit value (Time Slot 0-15)

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_SCO_Time_Slot 0xFC23

This command reads SCO time slot for the next SCO connection.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

SCO_Time_Slot

Type	uint8
Size	1 byte
Description	TS Assignment, 4 bit value (Time Slot 0-15)

Delay_Peripheral_SCO_Startup 0xFC2A

This command controls delay of SCO peripheral. This command should only be used with UART transport.

Command Parameters

Enable_Delay

Type	boolean	
Size	1 byte	
Description		
Values	1	Delay SCO peripheral startup
	0	Do not delay SCO peripheral startup

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_Dynamic_SCO_Routing_Change 0xFC83

This command writes dynamic SCO routing enable.

Command Parameters

SCO_Routing

Type	uint8	
Size	1 byte	
Description	Specifies whether the SCO path is through PCM interface or transport.	
Values ***Note:*** *May not be combined bitwise.*	0x00	PCM
	0x01	Transport
	0x02	CODEC
	0x03	I2S
	0xFF	No Change

SCO_Connections

Type	uint8	
Size	1 byte	
Description	Specifies the SCO connection number.	
Values ***Note:*** *May not be combined bitwise.*	0x0	0
	0x1	1
	0x2	2
	0x3	3

Connection_Handle[0]

Type	uint16 (little endian)
Size	2 bytes (if present)
Description	Specifies the SCO connection handle.

SCO_Routing[0]

Type	uint8	
Size	1 byte (if present)	
Description	Specifies whether the SCO path is through PCM interface or transport.	
Values ***Note:*** *May not be combined bitwise.*	0x0	PCM
	0x1	Transport
	0x2	CODEC
	0x3	I2S

PCM_Time_Slot[0]

Type	uint16 (little endian)
Size	2 bytes (if present)
Description	TS Assignment, 4 bit value (Time Slot 0-15)

Input_Coding[0]

Type	unsigned integer	
Size	8 bits {15:8} in little endian uint16 (if present)	
Description	Input coding	
Values ***Note:*** *May not be combined bitwise.*	0x0	Linear
	0x1	u-Law
	0x2	A-law

Input_Data_Format[0]

Type	unsigned integer	
Size	2 bits {7:6} in little endian uint16 (if present)	
Description	Input date format	
Values ***Note:*** *May not be combined bitwise.*	0x0	1’s complement
	0x1	2’s complement
	0x2	Sign-magnitude
	0x3	Unsigned

Sample_Size[0]

Type	unsigned integer
Size	6 bits {5:0} in little endian uint16 (if present)
Description	Sample size.
Values ***Note:*** *May not be combined bitwise.*	0x00 | 8-bit 0x20 | 16-bit

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Update_UART_Baud_Rate 0xFC18

This command changes the baud rate at which the UART transport communicates. A Command Complete event is generated at the previous baud rate. If the Command Complete event indicates success, the UART transport shifts to a new baud rate.

Command Parameters

Encoded_Baud_Rate

Type	uint16 (little endian)
Size	2 bytes
Description	Specifies the hardware register representation of the new baud rate. The value is determined by a formula (described below).

For the BCM2045, the value of Encoded_Baud_Rate can be 0x0000, which implies that the encoded form is not used and the actual baud rate follows. When using the UART transport of the BCM2045 in three-wire SLIP (H5) mode, if Update_UART_Baud_Rate is issued, use of the encoded form is disallowed.

If the encoded form is used, the value is determined as follows:

```
// Upper four bits
uint32 encoded_segment = ( ( (24000000 / baud_rate) % 16 ) / 2 );
encoded_baud_rate = ( ( (uint16)encoded_segment ) << 4 );


// Lower four bits of high nibble

encoded_segment = ( ( ( (24000000 / baud_rate) % 16 ) / 2 ) +
                  ( ( (24000000 / baud_rate) % 16 ) % 2 ) );
encoded_baud_rate |= (uint16)encoded_segment;

// Lower byte
encoded_segment = ( 256 - ( (24000000 / baud_rate) / 16 ) );
encoded_baud_rate |= (encoded_segment << 8);
```

Note: The formula above is only based on a 24 MHz UART clock; it does not work with a 48MHz UART clock.

Use_Encoded_Form

Type	boolean
Size	0 bytes (abstract)
Description	An abstraction of whether the encoded form is to be used. If not using the encoded form of baud rate, the Encoded_Baud_Rate parameter is 0x0000 and a four-byte baud rate in integer form follows in the Explicit_Baud_Rate parameter.

Explicit_Baud_Rate

Type	uint32 (big endian)
Size	4 bytes (if present)
Description	If Encoded_Baud_Rate is 0x0000 and therefore the encoded form of baud rate is not being used, the Explicit_Baud_Rate parameter will be present, containing the baud rate in integer form.

Baud_Rate

Type	uint32 (big endian)
Size	0 bytes (abstract)
Description	An abstraction of the baud rate, whether present in the command in encoded form in the Encoded_Baud_Rate or included as Explicit_Baud_Rate.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_SCO_Time_Slot 0xFC22

This command writes SCO time slot for the next SCO connection.

Command Parameters

SCO_Time_Slot

Type	uint8
Size	1 byte
Description	TS Assignment, 4 bit value (Time Slot 0-15)

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_SCO_Time_Slot

This command reads SCO time slot for the next SCO connection.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

SCO_Time_Slot

Type	uint8
Size	1 byte
Description	TS Assignment, 4 bit value (Time Slot 0-15)

Delay_Peripheral_SCO_Startup 0xFC2A

This command controls delay of SCO peripheral. This command should only be used with UART transport.

Command Parameters

Enable_Delay

Type	boolean
Size	1 byte
Values	1 Delay SCO peripheral startup 0 Do not delay SCO peripheral startup

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Write_Dynamic_SCO_Routing_Change 0xFC83

This command writes dynamic SCO routing enable.

Command Parameters

SCO_Routing

Type	uint8
Size	1 byte
Description	Specifies whether the SCO path is through PCM interface or transport.
Values ***Note:*** *May not be combined bitwise.*	0x00 PCM 0x01 Transport 0x02 CODEC 0x03 I2S 0xFF No Change

Power Management

Set_Sleepmode_Param 0xFC27

This command activates the selected sleep mode algorithm and specifies respective timer thresholds.

Command Parameters

Sleep_Mode

Type	uint8	
Size	1 byte	
Description	Sleep mode algorithm selection.	
Values ***Note:*** *May not be combined bitwise.*	0x0	No sleep mode
	0x1	UART
	0x2	UART with messaging
	0x3	USB
	0x4	H4IBSS
	0x5	USB with host wake
	0x6	SDIO
	0x7	UART CS-N
	0x8	SPI
	0x9	H5
	0xA	H4DS
	0xC	UART with BREAK

Idle_Threshold_Host

Type	uint8
Size	1 byte
Description	Host Idle Threshold, applicable to Sleep Modes 1, 2, 5, and 7. This is the number of firmware loops executed with no activity before the Host wake line is deasserted. Activity includes HCI traffic excluding certain sleep mode commands and the presence of SCO connections if the *Allow Host Sleep During SCO* flag is not set to 1. Each count of this parameter is roughly equivalent to 300 ms. For example, when this parameter is set to 16 (0x10), the Host wake line will be deasserted after approximately 4.8 seconds of inactivity.

Idle_Threshold_HC

Type	uint8
Size	1 byte
Description	Host Controller Idle Threshold, applicable to Sleep Modes 1, 2, 3, 4, 5, 6, 7, and 9. This is the number of firmware loops executed with no activity before the HC is considered idle. Depending on the mode, HC may then attempt to sleep. Activity includes HCI traffic excluding certain sleep mode commands and the presence of ACL/SCO connections. Each count of this parameter is roughly equivalent to 300 ms. For example, when this parameter is set to 16 (0x10), the HC will be considered idle after approximately 4.8 seconds of inactivity.

BT_WAKE_Active_Mode

Type	uint8
Size	1 byte
Description	Applicable to Sleep Modes 1, 2, and 7. This flag indicates whether the BT_WAKE line is active low or high. GPIO0 is typically used for BT_WAKE.
Values ***Note:*** *May not be combined bitwise.*	0x0 Active Low 0x1 Active High

HOST_WAKE_Active_Mode

Type	uint8
Size	1 byte
Description	Applicable to Sleep Modes 1, 2, 5, 7. This flag indicates whether the HOST_WAKE line is active low or high. GPIO3 is typically used for HOST_WAKE on BCM2035, and GPIO1 on ARM-based chips.
Values ***Note:*** *May not be combined bitwise.*	0x0 Active Low 0x1 Active High

Allow_Host_Sleep_During_SCO

Type	boolean
Size	1 byte
Description	Applicable to Sleep Modes 1, 2, 3, 5, 7. When this flag is set to 0, the host is not allowed to sleep while an SCO is active. In modes 1 and 2, the device will keep the host wake line asserted while an SCO is active. In mode 3, the device will immediately issue a USB RESUME if the host issues a SUSPEND. When this flag is set to 1, the host can sleep while an SCO is active. This flag should only be set to 1 if SCO traffic is directed to the PCM interface.

Combine_Sleep_Mode_And_LPM

Type	boolean
Size	1 byte
Description	Applicable to Sleep Modes 1, 2, 3, 5, 7. In Mode 0, always set byte 7 to 0. In all sleep modes, device always requires permission to sleep between scans / periodic inquiries regardless of the setting of this byte. In Modes 1 and 2, if byte is set, device must have "permission" to sleep during the low power modes of sniff, hold, and park. If byte is not set, device can sleep without permission during these modes. Permission to sleep in Mode 1 is obtained if the BT_WAKE signal is not asserted. Permission to sleep in Mode 2 occurs after the Sleep Request / Sleep Request ACK exchange. In Mode 3 and 5, if the byte is set to 0, the device will not be able to sleep during the low power modes. If it is set to 1, the device will be able so sleep during the low power modes.

Enable_Tristate_Control_Of_UART_Tx_Line

Type	boolean
Size	1 byte
Description	Applicable to Sleep Modes 1, 2, 7. When set to 0, the device will not tristate its UART TX line before going to sleep. When set to 1, the device will tristate its UART TX line before going to sleep.

Active_Connection_Handling_On_Suspend

Type	uint8
Size	1 byte
Description	Suspend Behavior, applicable To Sleep Modes 3, 5. When set to 0, this flag indicates that upon detecting a USB suspend, the device should sleep whenever possible. This means that the device will stay up if necessary to maintain active ACL and/or SCO connections, and will wake up to perform

	any scheduled periodic activities if configured to do so. When set to 1, the device will immediate go to sleep upon detecting a USB SUSPEND and will not wake up until USB RESUME is detected. This will cause all connections (whether ACL or SCO, parked or not, etc.) to be dropped. All periodic activity will also be suspended. When this flag is set to 1, all other parameters are ignored.
Values ***Note:*** *May not be combined bitwise.*	0x1 — Maintain connections; sleep when timed activity allows 0x2 — Sleep until resume is detected

Resume_Timeout

Type	uint8
Size	1 byte
Description	Applicable to Sleep Modes 3, 5. After the device issues a USB RESUME, it will wait this many seconds for the Host to resume USB operations before issuing another USB RESUME. If this value is set to 0, the device will never reissue RESUME and will instead wait forever for the host to act on the initial RESUME.

Enable_BREAK_To_Host

Type	boolean
Size	1 byte
Description	Applicable to Sleep Mode 12. If 0 and sleep mode 12 is selected, disables setting a break condition to the host, making the sleep mechanism unidirectional.

Pulsed_HOST_WAKE

Type	boolean
Size	1 byte
Description	Applicable to Sleep Modes 1, 12. After asserting BT_WAKE (mode 1) or setting or clearing a BREAK condition (mode 12), if the host does not wake up, clear the condition and retry.

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Read_Sleepmode_Param 0xFC28

This command reads back sleep mode related parameters.

Command Parameters

None

Return Parameters

Status

Type	uint8 (error code per Bluetooth Core Specification)
Size	1 byte

Sleep_Mode

Type	uint8	
Size	1 byte	
Description	See *Set_Sleepmode_Param*.	
Values ***Note:*** *May not be combined bitwise.*	0x0	No sleep mode
	0x1	UART
	0x2	UART with messaging
	0x3	USB
	0x4	H4IBSS
	0x5	USB with host wake
	0x6	SDIO
	0x7	UART CS-N
	0x8	SPI
	0x9	H5
	0xA	H4DS
	0xC	UART with BREAK

Idle_Threshold_Host

Type	uint8
Size	1 byte
Description	See *Set_Sleepmode_Param*.

Idle_Threshold_HC

Type	uint8
Size	1 byte
Description	See *Set_Sleepmode_Param*.

BT_WAKE_Active_mode

Type	uint8
Size	1 byte
Description	See *Set_Sleepmode_Param*.
Values **Note: May not be combined bitwise.**	0x0 Active Low 0x1 Active High

HOST_WAKE_Active_mode

Type	uint8
Size	1 byte
Description	See *Set_Sleepmode_Param*.
Values **Note: May not be combined bitwise.**	0x0 Active Low 0x1 Active High

Allow_Host_Sleep_During_SCO

Type	boolean
Size	1 byte
Description	See *Set_Sleepmode_Param*.

Combine_Sleep_Mode_And_LPM

Type	boolean
Size	1 byte
Description	See *Set_Sleepmode_Param*.

Enable_Tristate_Control_Of_UART_Tx_Line

Type	boolean
Size	1 byte
Description	See *Set_Sleepmode_Param*.

Active_Connection_Handling_On_Suspend

Type	uint8
Size	1 byte
Description	See *Set_Sleepmode_Param*.
Values **Note: May not be combined bitwise.**	0x0 Maintain connections; sleep when timed activity allows 0x1 Sleep until resume is detected

Resume_Timeout

Type	uint8
Size	1 byte
Description	See *Set_Sleepmode_Param*.

Enable_BREAK_To_Host

Type	boolean
Size	1 byte
Description	See *Set_Sleepmode_Param*.

Pulsed_HOST_WAKE

Type	boolean
Size	1 byte
Description	See *Set_Sleepmode_Param*.

UART Timing Requirements

The following figure displays UART timing.

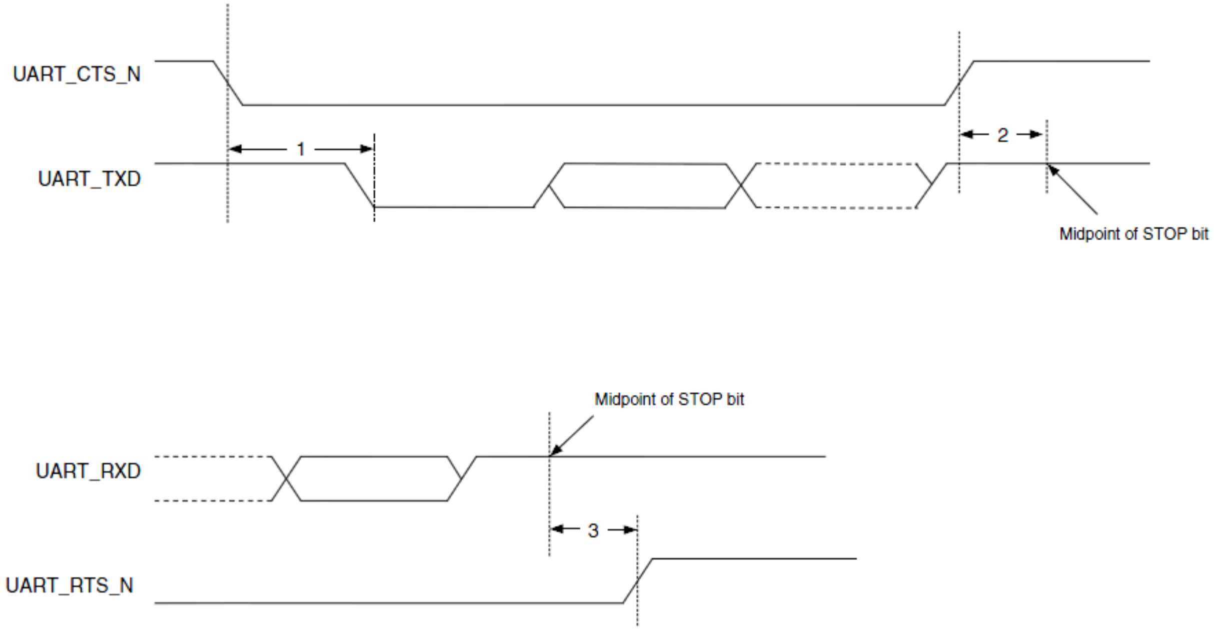


Figure 1: UART Timing Requirements

Notes: The UART 4-wire interface supports Bluetooth 2.1 HCI Specification.

Reference	Description	Min.	Typ.	Max.	Unit
1	Delay time, BT_UART_CTS_N low to UART_TXD valid	-	-	24	Baudout cycles
2	Setup time, BT_UART_CTS_N high before midpoint of stop bit	-	-	10	ns
3	Delay time, midpoint of stop bit to BT_UART_RTS_N high	-	-	2	Baudout cycles

PCM Interface Timing

Default PCM Format

The following is Summit’s default PCM format:

- SCO Routing – PCM
- Clock Mode – Master
- Sync Mode – Master
- Frame Type – Short
- Interface Rate – 512
- 8khz audio sample interval, 16 bit mono

Short Frame Sync, Master Mode

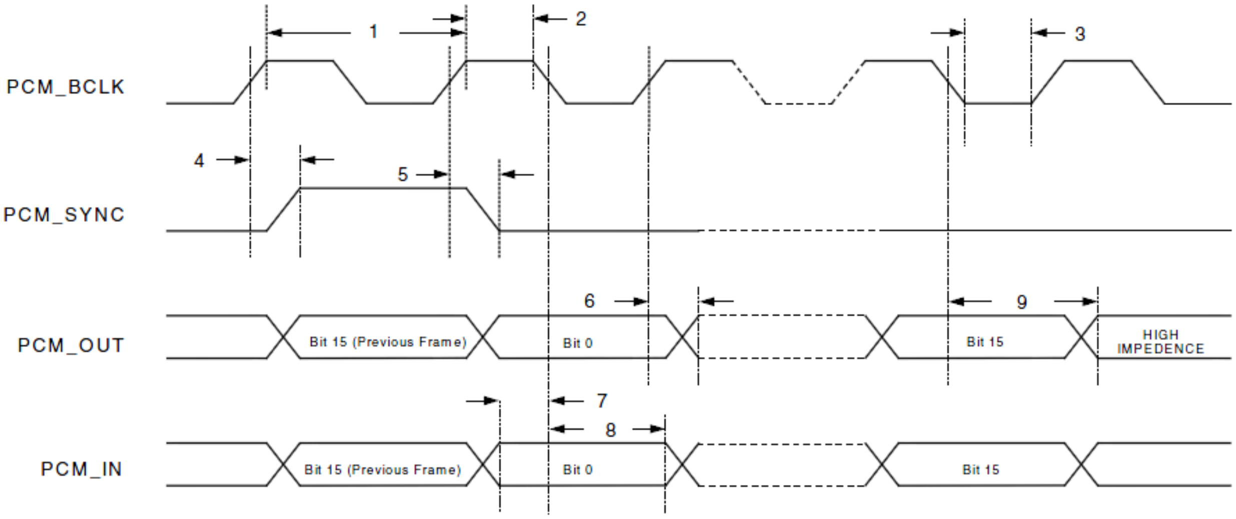


Figure 2: Short Frame Sync, Master Mode

Reference	Description	Min.	Typ.	Max.	Unit
1	PCM bit clock frequency	128	-	2048	kHz
2	PCM bit clock high time	128	-	-	ns
3	PCM bit clock low time	209	-	-	ns
4	Delay from BT_PCM_CLK rising edge to BT_PCM_SYNC high	-	-	50	ns
5	Delay from BT_PCM_CLK rising edge to BT_PCM_SYNC low	-	-	50	ns
6	Delay from BT_PCM_CLK rising edge to data valid on BT_PCM_OUT	-	-	50	ns

Reference	Description	Min.	Typ.	Max.	Unit
7	Setup time for BT_PCM_IN before BT_PCM_CLK falling edge	50	-	-	ns
8	Hold time for BT_PCM_IN after BT_PCM_CLK falling edge	10	-	-	ns
9	Delay from falling edge of BT_PCM_CLK during last bit period to BT_PCM_OUT becoming high impedance	-	-	50	ns

Short Frame Sync, Slave Mode

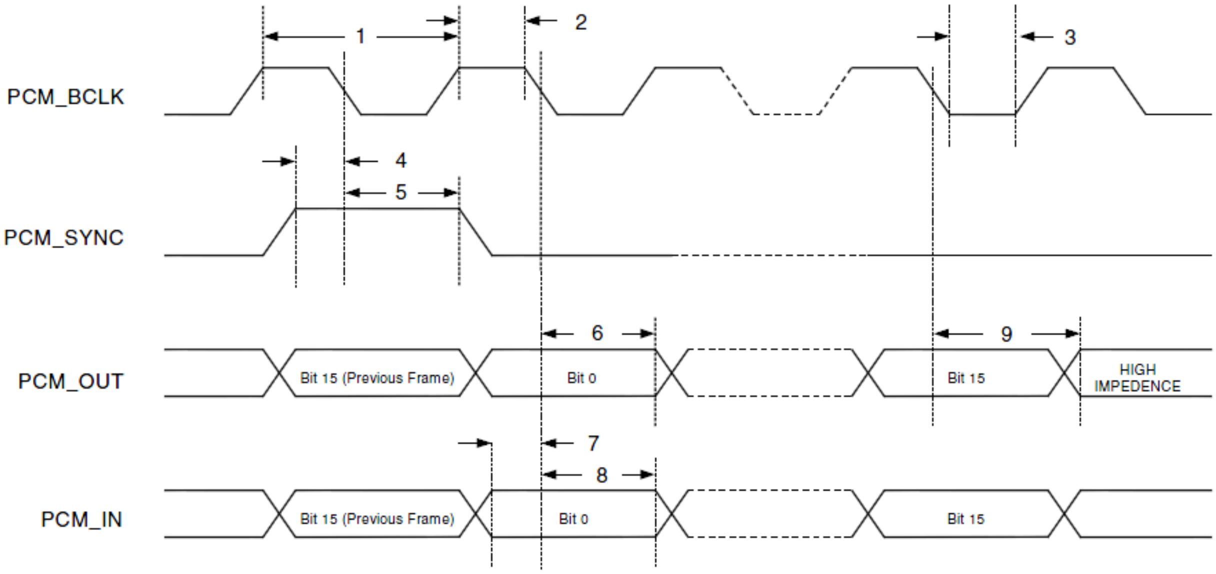


Figure 3: Short Frame Sync, Slave Mode

Reference	Description	Min.	Typ.	Max.	Unit
1	PCM bit clock frequency	128	-	2048	kHz
2	PCM bit clock high time	209	-	-	ns
3	PCM bit clock low time	209	-	-	ns
4	Setup time for BT_PCM_SYNC before falling edge of BT_PCM_BCLK	50	-	-	ns
5	Hold time for BT_PCM_SYNC after falling edge of BT_PCM_CLK	10	-	-	ns
6	Hold time of BT_PCM_OUT after BT_PCM_CLK falling time	-	-	175	ns

Reference	Description	Min.	Typ.	Max.	Unit
7	Setup time for BT_PCM_IN before BT_PCM_CLK falling edge	50	-	-	ns
8	Hold time for BT_PCM_IN after BT_PCM_CLK falling edge	10	-	-	ns
9	Delay from falling edge of BT_PCM_CLK during last bit period to BT_PCM_OUT becoming high impedance	-	-	100	ns

Long Frame Sync, Master Mode

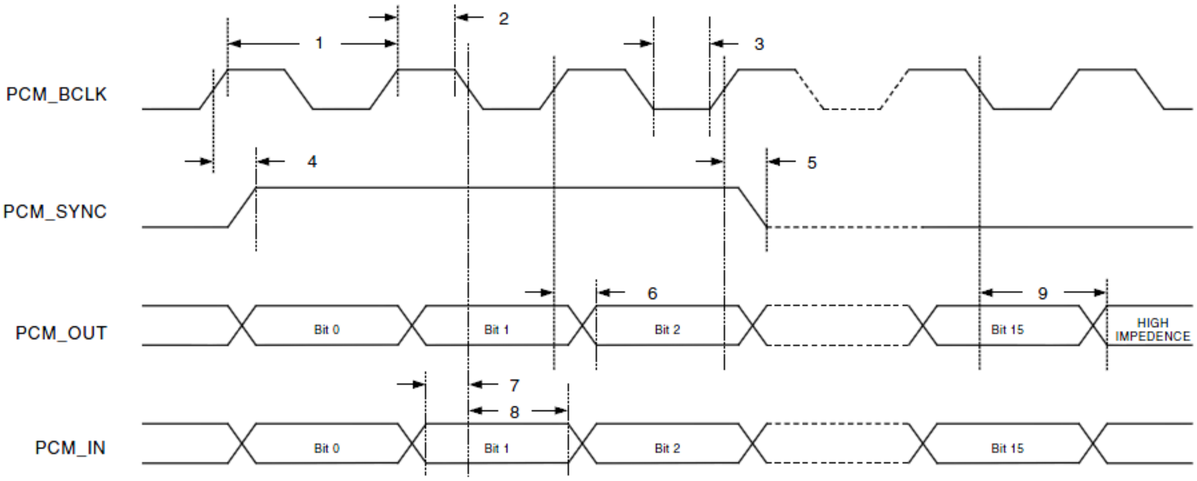


Figure 4: Long Frame Sync, Master Mode

Reference	Description	Min.	Typ.	Max.	Unit
1	PCM bit clock frequency	128	-	2048	kHz
2	PCM bit clock high time	209	-	-	ns
3	PCM bit clock low time	209	-	-	ns
4	Delay from BT_PCM_CLK rising edge to BT_PCM_SYNC high during first bit time	-	-	50	ns
5	Delay from BT_PCM_CLK rising edge to BT_PCM_SYNC low during third bit time	-	-	50	ns
6	Delay from BT_PCM_CLK rising edge to data valid on BT_PCM_OUT	-	-	50	ns
7	Setup time for BT_PCM_IN before BT_PCM_CLK falling edge	50	-	-	ns
8	Hold time for BT_PCM_IN after	10	-	-	ns

Reference	Description	Min.	Typ.	Max.	Unit
	BT_PCM_CLK falling edge				
9	Delay from falling edge of BT_PCM_CLK during last bit period to BT_PCM_OUT becoming high impedance	-	-	50	ns

Long Frame Sync, Slave Mode

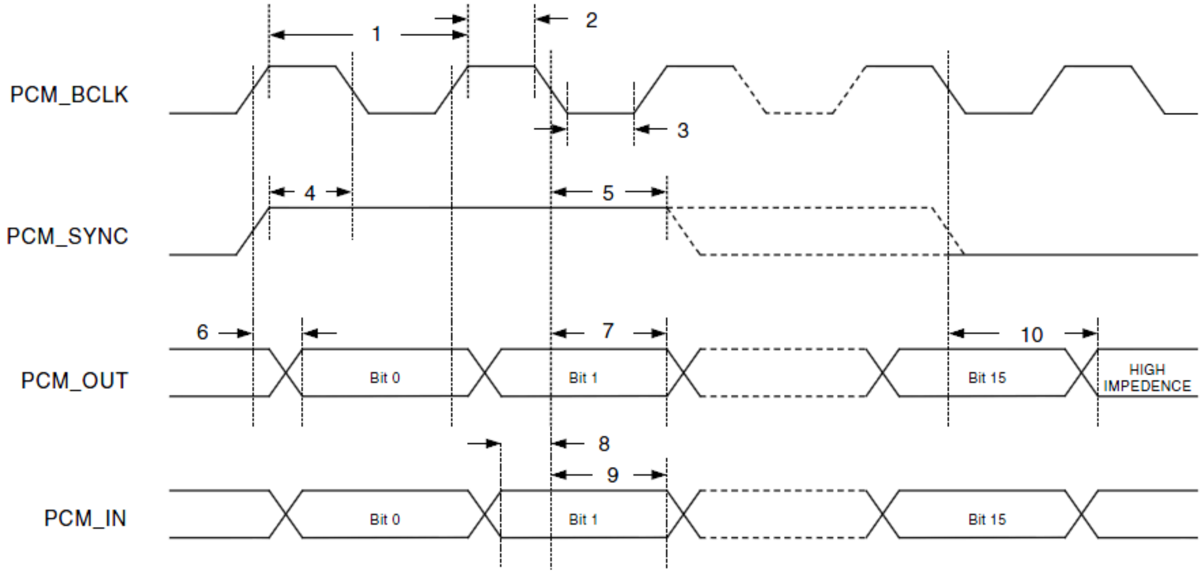


Figure 5: Long Frame Sync, Slave Mode

Reference	Description	Min.	Typ.	Max.	Unit
1	PCM bit clock frequency	128	-	2048	kHz
2	PCM bit clock high time	209	-	-	ns
3	PCM bit clock low time	209	-	-	ns
4	Setup time for BT_PCM_SYNC before falling edge of BT_PCM_CLK during first bit time	50	-	-	ns
5	Hold time for BT_PCM_SYNC after falling edge of BT_PCM_CLK during second bit period. **Note:** BT_PCM_SYNC may go low any time from second bit period to last bit period.	10	-	-	ns
6	Delay from rising edge of BT_PCM_CLK or BT_PCM_SYNC (whichever is later) to data valid for first bit on BT_PCM_OUT	-	-	50	ns
7	Hold time of BT_PCM_OUT after BT_PCM_CLK falling edge	-	-	175	ns
8	Setup time for BT_PCM_IN before	50	-	-	ns

Reference	Description	Min.	Typ.	Max.	Unit
	BT_PCM_CLK falling edge				
9	Hold time for BT_PCM_IN after BT_PCM_CLK falling edge	10	-	-	ns
10	Delay from falling edge of BT_PCM_CLK or BT_PCM_SYNC (whichever is later) during last bit in slot to BT_PCM_OUT becoming high impedence	-	-	100	