



▶ Wireless System-On-Chip Solutions



Host Controller Interface API

Company Confidential

Revision 1.0

March 31, 2002

Doc No. SPEC-00032

Approval/Date:

VP Engineering / Product Development

MICROTUNE

Wireless Connectivity Division

6440 Lusk Blvd., Suite D-205

San Diego, CA 92121, USA

(Phone) 858-558-6088

(Fax) 858-558-6598

(Web) <http://www.microtune.com/>

Information subject to change without notice.



Legal Notices

The information in this document is believed to be accurate and reliable. Microtune assumes no responsibility for any consequences arising from the use of this information, nor from any infringement of patents or the rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent or other rights of Microtune. The information in this publication replaces and supersedes all information previously supplied, and is subject to change without notice. The customer is responsible for assuring that proper design and operating safeguards are observed to minimize inherent and procedural.

Microtune assumes no responsibility for applications assistance or customer product design. The devices described in this document are not authorized for use in medical, life-support equipment, or any other application involving a potential risk of severe property or environmental damage, personal injury, or death without prior express written approval of Microtune. Any such use is understood to be entirely at the user's risk.

Microtune's products are protected by one or more of the following U.S. patents: 5,625,325; 5,648,744; 5,717,730; 5,737,035; 5,739,730; 5,805,988; 5,847,612; 6,100,761; 6,104,242; 6,144,402; 6,163,684; 6,169,569; 6,177,964; 6,218,899 and additional patents pending or filed.

Microtune, MicroTuner, Microtune logo, OneChip, and Transilica are trademarks of Microtune, Inc. All products and product names are the properties of Microtune, Inc. Bluetooth™ is a trademark of Bluetooth, SIG, Inc. USA and licensed to Microtune, Inc. All other trademarks belong to their respective companies. Sales and corporate contact information can be found in the back of this document.

Revision History

Revision	Release Date	Author	Description
1.0	March 31, 2002	Brian Day	Initial Release

Ordering Information

Model Number	Part Number	Package	Product Revision	Version
Xxxx	Xxxx	Xxxxxx		



Contents

- 1. Introduction 5**
- 2. Bluetooth HCI UART Transport Layer 1.1 6**
- 3. HCI Commands 6**
 - 3.1 Link Control Commands 6
 - 3.2 Link Policy Commands 7
 - 3.3 Host Controller and Baseband Commands 7
 - 3.4 Information Parameter Commands 9
 - 3.5 Status Parameter Commands 9
 - 3.6 Testing Commands..... 9
- 4. HCI Events 10**
- 5. Microtune Vendor Specific HCI Commands 11**
 - ett - HCI_VS_EXIT_TELEC_TEST 12
 - etm - HCI_VS_ENTER_TEST_MODE..... 12
 - etx - HCI_VS_Enter_Tx_Test..... 13
 - lf - HCI_VS_FIRMWARE_UPGRADE..... 14
 - rctv - HCI_VS_READ_CRYSTAL_TRIM_VALUE 14
 - rpl - HCI_VS_Read_Power_Level..... 15
 - rtc - HCI_VS_Read_Transport_Cfg..... 15
 - rusbid - HCI_VS_Read_USB_ID..... 16
 - crt - HCI_VS_CONTINUOUS_RX_TEST 16
 - ttt - HCI_VS_TELEC_TX_TEST 17
 - rlm - HCI_VS_Read_Link_Mode..... 18
 - uki - HCI_VS_UNIT_KEY_IMP..... 19
 - wlm - HCI_VS_Write_Link_Mode..... 19
 - wbda - HCI_VS_Write_BD_Address..... 20
 - wctv - HCI_VS_WRITE_CRYSTAL_TRIM_VALUE 21
 - wpl - HCI_VS_Write_Power_Level..... 22
 - wtc - HCI_VS_Write_Transport_Cfg 23
 - wusbid - HCI_VS_Write_USB_ID..... 24
 - HCI_VS_WRITE_SPEAKER_VOLUME 25
 - HCI_VS_DELETE_FIXED_PIN_CODE 25
 - HCI_VS_WRITE_FIXED_PIN_CODE..... 25



HCI_VS_Read_RF_Settings	25
HCI_VS_Write_RF_Settings	25
6. Baseband Limitations.....	25

Figures

Figure 1. Bluetooth Protocol Stack	5
--	---



1. Introduction

Microtune's OneChip™ contains the complete Bluetooth 1.1 protocol stack including: LC, LM, L2CAP, HCI, SDP, and RFCOMM. Figure 1 shows an example of the protocol stack. This document describes the Host Controller Interface (HCI) Implemented.

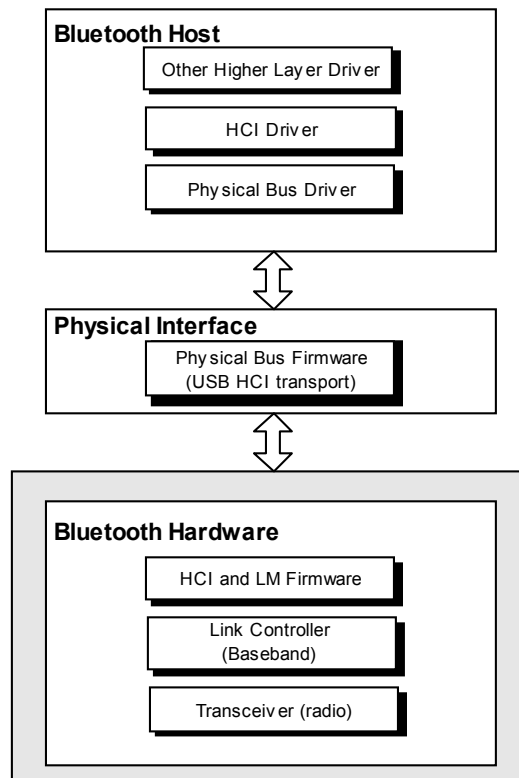


Figure 1. Bluetooth Protocol Stack

The HCI allows higher layers of the stack to access the baseband controller, link manager, and other hardware status and control registers through a standard interface. Synchronous and asynchronous traffic passes through the HCI as it is transmitted and received by the host. Access to the layers is handled via Application Programming Interface (API) calls. Through HCI *commands*, OneChip can enter certain operational modes, and through HCI *events*, higher layers of the stack can be informed of specific operational results.



2. Bluetooth HCI UART Transport Layer 1.1

Microtune supports HCI UART Transport Layer either on UART1/UART2 between the host and the host controller. HCI commands, events, and data packets flow through this layer but are not decoded. The following table lists planned updates.

Functionality	Status
SCO Data Packets	Scheduled for the next release

3. HCI Commands

This section lists the Bluetooth *HCI commands* that have been implemented in the OneChip. For details of each command, refer to the Bluetooth Specification.

Important: The commands shown in this document are for Microtune's use only.

3.1 Link Control Commands

The table shows supported Bluetooth *Link Control* commands.

Command	Implemented
Accept_Connection_Request	Yes
Add_SCO_Connection	Yes
Authentication_Requested	Yes
Change_Connection_Link_Key	Yes
Change_Connection_Packet_Type	Yes
Create_Connection	Yes
Disconnect	Yes
Exit_Periodic_Inquiry_Mode	Yes
Inquiry	Yes
Inquiry_Cancel	Yes
Link_Key_Request_Negative_Reply	Yes
Link_Key_Request_Reply	Yes
Master_Link_Key	Yes
Periodic_Inquiry_Mode	Yes
PIN_Code_Request_Negative_Reply	Yes
PIN_Code_Request_Reply	Yes
Read_Clock_Offset	Yes
Read_Remote_Supported_Features	Yes
Read_Remote_Version_Information	Yes
Reject_Connection_Request	Yes
Remote_Name_Request	Yes
Set_Connection_Encryption	Yes



3.2 Link Policy Commands

The table shows supported Bluetooth *Link Policy* commands.

Command	Implemented
Exit_Park_Mode	Yes
Exit_Sniff_Mode	Yes
Hold Mode	Yes
Park_Mode	Yes
QoS_Setup	Not implemented
Read_Link_Policy_Settings	Yes
Role_Discovery	Yes
Sniff Mode	Yes
Switch_Role	Yes
Write_Link_Policy_Settings	Yes

3.3 Host Controller and Baseband Commands

The table shows supported Bluetooth *Host Controller and Baseband* commands.

Command	Implemented
Change_Local_Name	Yes
Create_New_Unit_Key	Yes
Delete_Stored_Link_Key	Yes
Flush	Yes
Host_Buffer_Size	Yes
Host_Number_Of_Completed_Packets	Yes
Read_Authentication_Enable	Yes
Read_Automatic_Flush_Timeout	Yes
Read_Class_Of_Device	Yes
Read_Connection_Accept_Timeout	Yes
Read_Current_IAC_LAP	Yes
Read_Encryption_Mode	Yes
Read_Hold_Mode_Activity	Yes
Read_InquiryScan_Activity	Yes
Read_Link_Supervision_Timeout	Yes
Read_Local_Name	Yes
Read_Num_Broadcast_Retransmissions	Yes
Read_Number_Of_Supported_IAC	Yes
Read_Page_Scan_Mode	Yes



Command	Implemented
Read_Page_Scan_Period_Mode	Yes
Read_Page_Timeout	Yes
Read_PageScan_Activity	Yes
Read_PIN_Type	Yes
Read_Scan_Enable	Yes
Read_SCO_Flow_Control_Enable	Yes
Read_Stored_Link_Key	Yes
Read_Transmit_Power_Level	Yes
Read_Voice_Setting	Yes
Reset	Yes
Set_Event_Filter	Yes
Set_Event_Mask	Yes
Set_Host_Controller_To_Host_Flow_Control	Yes
Write_Authentication_Enable	Yes
Write_Automatic_Flush_Timeout	Yes
Write_Class_Of_Device	Yes
Write_Connection_Accept_Timeout	Yes
Write_Current_IAC_LAP	Yes
Write_Encryption_Mode	Yes
Write_Hold_Mode_Activity	Yes
Write_InquiryScan_Activity	Yes
Write_Link_Supervision_Timeout	Yes
Write_Num_Broadcast_Retransmissions	Yes
Write_Page_Scan_Mode	Yes
Write_Page_Scan_Period_Mode	Yes
Write_Page_Timeout	Yes
Write_PageScan_Activity	Yes
Write_PIN_Type	Yes
Write_Scan_Enable	Yes
Write_SCO_Flow_Control_Enable	Yes
Write_Stored_Link_Key	Yes
Write_Voice_Setting	Yes



3.4 Information Parameter Commands

The table shows supported Bluetooth *Information Parameter* commands.

Command	Implemented
Read_BD_ADDR	Yes
Read_Buffer_Size	Yes
Read_Country_Code	Yes
Read_Local_Supported_Features	Yes
Read_Local_Version_Information	Yes

3.5 Status Parameter Commands

The table shows supported Bluetooth *Status Parameter* commands.

Command	Implemented
Get_Link_Quality	Scheduled for the next release
Read_Failed_Contact_Counter	Yes
Read_RSSI	Scheduled for the next release
Reset_Failed_Contact_Counter	Yes

3.6 Testing Commands

The table shows supported Bluetooth *Testing* commands.

Command	Implemented
Enable_Device_Under_Test_Mode	Yes
Read_Loopback_Mode	Yes
Write_Loopback_Mode	Yes



4. HCI Events

The following Bluetooth *HCI events* have been implemented in the OneChip. For more information, refer to the Bluetooth Specification. The *events* described in the table are standard Bluetooth events.

Event	Implemented
Authentication_Complete_Event	Yes
Change_Connection_Link_Key_Complete_Event	Yes
Command_Complete_Event	Yes
Command_Status_Event	Yes
Connection_Complete_Event	Yes
Connection_Packet_Type_Changed_Event	Yes
Connection_Request_Event	Yes
Data_Buffer_Overflow_Event	Yes
Disconnection_Complete_Event	Yes
Encryption_Change_Event	Yes
Flush_Occurred_Event	Yes
Hardware_Error_Event	Yes
Inquiry_Complete_Event	Yes
Inquiry_Result_Event	Yes
Link_Key_Notification_Event	Yes
Link_Key_Request_Event	Yes
Loopback_Command_Event	Yes
Master_Link_Key_Complete_Event	Yes
Max_Slots_Change_Event	Yes
Number_Of_Completed_Packets_Event	Yes
Page_Scan_Mode_Change_Event	Yes
Page_Scan_Repetition_Mode_Change_Event	Yes
PIN_Code_Request_Event	Yes
QoS_Setup_Complete_Event	Scheduled for the next release
QoS_Violation_Event	Yes
Read_Clock_Offset_Complete_Event	Yes
Read_Remote_Supported_Features_Complete_Event	Yes
Read_Remote_Version_Information_Complete_Event	Yes
Remote_Name_RequestComplete_Event	Yes
Return_Link_Keys_Event	Yes
Role_Change_Event	Yes



5. Microtune Vendor Specific HCI Commands

The Microtune *HCI commands* listed below are implemented in the OneChip. These commands are **not** described in the Bluetooth specification.

These functions are Vendor Specific (VS) Commands used for customer specific tests. The commands with HCI_VS_<Command> are vendor specific commands and are applicable only for Microtune modules. Where EBDK responses and support exist, special notations are provided. HCI Terminal can be used to issue TELEC test commands over USB I/F.

Parameter	Command
etc	HCI_VS_EXIT_TELEC_TEST
etm	HCI_VS_ENTER_TEST_MODE
etx	HCI_VS_Enter_Tx_Test
lf	HCI_VS_FIRMWARE_UPGRADE
rctv	HCI_VS_READ_CRYSTAL_TRIM_VALUE
rpl	HCI_VS_Read_Power_Level
rtc	HCI_VS_Read_Transport_Cfg
rusbid	HCI_VS_Read_USB_ID
crt	HCI_VS_CONTINUOUS_RX_TEST
ttt	HCI_VS_TELEC_TX_TEST
rlm	HCI_VS_Read_Link_Mode
uki	HCI_VS_UNIT_KEY_IMP
wlm	HCI_VS_Write_Link_Mode
wbda	HCI_VS_Write_BD_Address
wctv	HCI_VS_WRITE_CRYSTAL_TRIM_VALUE
wpl	HCI_VS_Write_Power_Level
wtc	HCI_VS_Write_Transport_Cfg
wusbid	HCI_VS_Write_USB_ID
Not Implemented	HCI_VS_WRITE_SPEAKER_VOLUME
Obsolete	HCI_VS_DELETE_FIXED_PIN_CODE
Obsolete	HCI_VS_WRITE_FIXED_PIN_CODE
Obsolete	HCI_VS_Read_RF_Settings
Obsolete	HCI_VS_Write_RF_Settings



ett - HCI_VS_EXIT_TELEC_TEST

Synopsis

ett <no parameters>

Parameters

None

Return Event

Command_Complete_Event

HCI Terminal

Ett

EBDK

raw 01 1a fc 0

Note

The module would RESET itself after this command is issued.

etm - HCI_VS_ENTER_TEST_MODE

Synopsis

etm

Parameters

None

Return Event

Command_Complete_Event

Example

etm <Opcode, 0xFC11> <paralen, 0>

HCI Terminal

etm

EBDK

Raw 01 11 fc 0



etx - HCI_VS_Enter_Tx_Test

Synopsis

etx <parameter_list>

(9 parameters)

Parameters

Name	Meaning
0	Conn Handle
1	Test Scenario
2	Hopping Mode
3	Tx Frequency
4	Rx Frequency
5	Power Control Mode
6	Poll Period
7	Packet Type
8	Length of Test Sequence

Description

Enters the module in Transmitter TestMode. Opcode – 0xFC0C.

Return Event

Command_Complete_Event

Example

```
etx <Opcode, 0xFC0C> <paralen, 11> <Conn Handle> <Test Scenario>
    <Hopping Mode> <Tx Freq> <Rx Freq> <Power Control> <Poll Period>
    <Packet type> <len of Test Sequence, 2bytes>
```

HCI Terminal

```
etx 1 0 0 0 0 0 2 4 27
```

EBDK

```
Raw 01 0C FC 0B 01 00 00 00 00 00 00 00 2 4 1B 00
```



lf - HCI_VS_FIRMWARE_UPGRADE

Synopsis

lf <no parameters>

Parameters

None

Return Event

Command_Complete_Event

HCI Terminal

lf

EBDK

raw 01 17 FC 00

rctv - HCI_VS_READ_CRYSTAL_TRIM_VALUE

Synopsis

rctv

Parameters

None

Return Event

Command_Complete_Event

HCI Terminal

rctv

EBDK

raw 01 15 fc 0



rpl - HCI_VS_Read_Power_Level

Synopsis

RPL <parameter_list>

(1 parameter)

Parameters

Name	Meaning
0	Temporary/Permanent Change [1 bytes] Options: Temporary = 0x1 Permanent = 0x0

Description

Reads power level settings. Opcode – 0xFC06.

Return Event

Command_Complete_Event

Example

```
rpl <Opcode, 0xFC06> <parameter, 01> <Temp/Perm>
```

HCI Terminal

```
RPL 0x01
```

EBDK

```
#Microtune read power level  
raw 01 06 FC 01 00
```

rtc - HCI_VS_Read_Transport_Cfg

Synopsis

RTC <no parameters>

Parameters

None

Return Event

Command_Complete_Event

Example

```
RTC
```

Note



Not Implemented. This command will be available in the next version.

rusbid - HCI_VS_Read_USB_ID

Synopsis

RUSBID <no parameters>

Parameters

None

Description

Reads vendor ID and product ID. Opcode – 0xFC0A.

Return Event

Command_Complete_Event

Example

```
rusbid <Opcode, 0xFC0A> <para len, 0>
```

HCI Terminal

```
RUSBID
```

EBDK

```
#Microtune read USB ID
raw 01 0A FC 00
```

Note

Not Implemented. This command will be available in the next version.

crt - HCI_VS_CONTINUOUS_RX_TEST

Synopsis

CRT <parameter_list>

(1 parameter)

Parameters

Name	Meaning
0	Inquiry scan or Page scan

Return Event

Command_Complete_Event

Example

```
Raw <hci_cmd_id = 01> <opcode = 0xfc19> <para_len = 0x01> <Inquiry/Page scan mode>
```



HCI Terminal

Crt 1 or crt 2

EBDK

raw 01 19 fc 1 {1 or 2}

Value 1 is for Inquiry scan and 2 is for Page scan.

Note

HARDWARE reset should be performed after the test has been done.

ttt - HCI_VS_TELEC_TX_TEST

Synopsis

TTT <parameter_list>

(4 parameters)

Parameters

Name	Meaning
0	Test case
1	Modulation On=1, Off=0
2	Frequency Hopping On=1, Off=0
3	Channel ID will be ignored if Frequency Hopping is ON.

Return Event

Command_Complete_Event

Example 1 (Raw):

Raw <hci_cmd_id = 01> <opcode = 0xfc18> <para_len = 0x04><testcase# = 1> <modulation = 0> <Frequency Hopping = 0><Channel ID = 39>

Test Case 1:

Continuous transmission without modulation, without Freq Hopping.

Command:

raw 01 18 fc 4 1 0 0 [0-78]

Test Case 2:

Continuous transmission with modulation, with PN9 sequence and with Freq Hopping.

Command:

raw 01 18 fc 4 2 1 1 0



Test Case 4:

Continuous transmission with modulation, without frequency hopping.

Command:

```
raw 01 18 fc 4 4 1 0 [0-78]
```

HCI Terminal

Test Case 1:

```
ttt 1 0 0 0x[0-4E]
```

Test Case 2:

```
ttt 2 1 1 0
```

Test Case 3:

```
ttt 4 1 0 0x[0-4E]
```

rlm - HCI_VS_Read_Link_Mode

Synopsis

RIM <no parameters>

Parameters

None

Description

Reads the current settings of frequency hopping, data whitening, and CRC check. Opcode – 0xFC04.

Return Event

Command_Complete_Event

Example

```
rlm <Opcode, 0xFC04> <para len, 0>
```

HCI Terminal

```
RIM
```

EBDK

```
#Microtune read link mode
```

```
raw 01 04 FC 00
```



uki - HCI_VS_UNIT_KEY_IMP

Synopsis

uki

Parameters

None

Return Event

Command_Complete_Event

HCI Terminal

uki

EBDK

raw 01 13 fc 0

wlm - HCI_VS_Write_Link_Mode

Synopsis

WLM <parameter_list>

(2 parameters)

Parameters

Name	Meaning
0	Freq Hopping/Whitening/CRC [1 bytes] Options : Freq Hopping ON = 0x80 Whitening ON = 0x40 CRC check ON = 0x01]
1	Channel ID [1 bytes] Options:0-78] (Applicable only is Freq Hop off)

Description

Writes frequency hopping, data whitening, and CRC check for the device. Opcode – 0xFC03.

Return Event

Command_Complete_Event



Example

```
wlm <Opcode, 0xFC03> <para len, 02> <Freq/whitening/CRC, Channel ID>
```

HCI Terminal

```
Wlm 0x1 0
```

EBDK

```
#Microtune write link mode  
raw 01 03 FC 02 80 00
```

wbda - HCI_VS_Write_BD_Address

Synopsis

```
WBDA <parameter_list>
```

(1 parameter)

Parameters

Name	Meaning
0	BD_ADDR [6 bytes]

Description

Writes Bluetooth Address of the Device. Opcode – 0xFC02.

Return Event

Command_Complete_Event

Example

```
wbda <Opcode, 0xFC02> <para len, 06> <BdAddr>
```

HCI Terminal

```
WBDA 0x0000379E8F9B
```

EBDK

```
#Microtune Bd addr  
raw 01 02 FC 06 9b 8f 9e 37 00 00
```



wctv - HCI_VS_WRITE_CRYSTAL_TRIM_VALUE

Synopsis

wctv <parameter List>

(1 parameter)

Parameters

Name	Meaning
0	Crystal Trim Value. Value ranges from 0 to 7.

Return Event

Command_Complete_Event

HCI Terminal

Wctv 1

EBDK

raw 01 16 fc 01 [0..7]

Note

Crystal Frequency changes according to the value set.

Tune Value	Frequency
0	24.000114
1	24.000018
2	23.999944
3	23.999864
4	23.999808
5	23.999722
6	23.999650
7	23.999570



wpl - HCI_VS_Write_Power_Level

Synopsis

WPL <parameter_list>

(2 parameters)

Parameters

Name	Meaning
0	Write the Temporary Power Level value or Permanent Power Level value [1 byte]. Options: Temporary = 0x1 Permanent = 0x0
1	Power Level [1 byte] Options: Range 0 through 7

Description

Writes power level of the device. Opcode – 0xFC05.

Return Event

Command_Complete_Event

Example

```
wpl <Opcode, 0xFC05> <para len, 02> <Temp/Perm, 01> <power level, 01>
```

HCI Terminal

```
WPL 0x01 0x06
```

EBDK

```
#Microtune write power level
```

```
raw 01 05 FC 02 00 06
```

Note

Must RESET via button or HCI to take effect.



wtc - HCI_VS_Write_Transport_Cfg

Synopsis

WTC <parameter_list>

(2 parameters)

Parameters

Name	Meaning
0	Specify the Transport Configuration to be used by Application. USB = 0 UART(no HW flow) = 1 UART(HW flow) = 2 Size – 1 byte
1	Specify the BAUD Rate, Applicable only if the configuration is UART. Size – 4 bytes.

Description

Reads vendor ID and product ID. Opcode – 0xFC07.

Return Event

Command_Complete_Event

Example

```
wtc <Opcode, 0xFC07> <para len, 05> <USB/UART, 01> <Baud rate, 04>
```

HCI Terminal

```
WTC 0x01 0x001E01
```

EBDK

```
#HCI_MICROTUNE_Write_Transport_Cfg
#57600
raw 01 07 FC 05 01 00 E1 00 00
#115200
raw 01 07 FC 05 01 00 C2 01 00
```



wusbid - HCI_VS_Write_USB_ID

Synopsis

WUSBID <parameter_list>

(2 parameters)

Parameters

Name	Meaning
0	Specify the Vendor ID. Size – 2 bytes.
1	Specify the Product ID. Size – 2 bytes.

Description

Writes vendor ID and product ID. Opcode – 0xFC09. To be implemented.

Return Event

Command_Complete_Event

Example

```
wusbid <Opcode, 0xFC09> <para len, 04> <Vend ID, 02> <Prod ID, 02>.
```

HCI Terminal

```
WUSBID 0x0001 0x0001
```

EBDK

```
#Microtune write USB ID  
raw 01 09 FC 04 01 00 02 00
```

Note

This command will be available in the next version.



HCI_VS_WRITE_SPEAKER_VOLUME

Not implemented.

HCI_VS_DELETE_FIXED_PIN_CODE

Discontinued.

HCI_VS_WRITE_FIXED_PIN_CODE

Discontinued.

HCI_VS_Read_RF_Settings

Discontinued.

HCI_VS_Write_RF_Settings

Discontinued.

6. Baseband Limitations

None

Contact Information

World Headquarters

Microtune, Inc.
2201 Tenth Street
Plano, TX 75075 USA
Phone: 972-673-1600
Fax: 972-673-1602
Web: www.microtune.com
Sales: sales@microtune.com

Wireless Connectivity Division Offices

San Diego, CA

Microtune, Inc.
6440 Lusk Blvd, Suite D-205
San Diego, CA 92121 USA
Phone: 858-558-6088
Fax: 858-558-6598

Tokyo, Japan

Microtune, Inc.
Kawasho building 4F
3-10 Nihonbashi Kodemmacho
Chuo-ku, Tokyo 103-0001 Japan
Phone: +81 (0) 3-5652-0250
Fax: +81 (0) 3-5652-0251

Taipei, Taiwan

Microtune, Inc.
18F-7, No.77, Sec. 1, Hsin Tai Wu Rd.,
Hsi-Chih
Taipei Hsien, Taiwan
Phone: +886 (02) 2698-8648
Fax: +886 (02) 2698-8647

Singapore

Microtune, Inc.
1 International Business Park
#03-14A The Synergy
Podium Block, Singapore 609917
Phone: +65 567 9724
Fax: +65 567 9714

