i1107e-s Programming Manual(培英)

October 26, 2018

Version 1.1

VERSION HISTORY

REVISION	AMENDMENT	DATE	AUTHOR
1.0	Initial version	2018-04-25	Joe Li Sunhua Wang Yuqiang
1.1	Modify instruction	2018-10-26	Shi Xuyao

Contents

1 Introduction

i1107e-s is Bluetooth 5.0 dual-mode module. It supports Bluetooth Classic 2.1+EDR, and Bluetooth low energy. I1107e-s is embedded with firmware. This firmware supports HFP (Hands-free Profile), HSP (Headset Profile), A2DP (Advanced Audio Distribution Profile), AVRCP (Audio & Video Remote Control Profile), PBAP (Phonebook Access Profile), MAP (Message Access Profile), SPP (Serial Port Profile), HID (Human Interface Device Profile) and LE. This firmware also supports Bluetooth 5.0 GATT profile.

There are two roles in Firmware: the module acts as controller role, and MCU acts as host role. Bluetooth functions are embedded in the controller unit and the application is running in the host unit. To achieve high-level hardware integration, the host unit communicates with the controller unit via UART by using well defined AT commands.

This document addresses i1107e's default setting, and AT commands.

2 Important Notes

This chapter discusses i1107e-s's defaulting setting and parameter maximum length.

2.1 Default Settings

This chapter introduces i1107e-s's defaulting setting.

2.1.1 UART Default Setting

The UART default setting is 115200, 8, N, 1

2.1.2 Default Device Name

i1107e-s default device name is i1107e.

2.1.3 Default PIN Code

The default PIN Code is 0000.

2.1.4 Default Device Type

The default device type is 0x001F00 (Unknown Bluetooth device).

2.1.5 Default Speaker Volume

The default speaker volume is 15. It can be changed with the command in the chapter 6.11.

2.1.6 Default Microphone Volume

The default microphone volume is 15. It can be changed with the command in the chapter 6.12.

2.2Boot-up Timing

Host shall open the UART port immediately after power on the Bluetooth module. The host can receive the "AT-B SNKINIT 0\r" string when the Bluetooth system initializing finished.

2.3Parameter Maximum Length

This chapter introduces parameters' maximum length.

2.3.1 *Bluetooth* Software Version Information

An ASCII code string like "i1107e.STD.20180330.1".

2.3.2 *Bluetooth* Device Address

An ASCII code string like "1234567890AB", the length is 12 bytes.

2.3.3 *Bluetooth* Device Name

The maximum length of the Bluetooth device name is a 31 bytes UTF8 code string with a mix of 'A'-'Z','a'-'z','0'-'9'. The length of the local device name must be between 1 byte and 31 byte.

If the remote device name is non-English letters, the host unit shall call a UTF8 to ASCII converter to display characters correctly. Please refer to the PC host APP code for more information.

2.3.4 PIN Code

The maximum length of PIN code is 16 bytes. Only '0'-'9' is admitted.

2.3.5 Dial Number Length

The number is dialed out. There is no limit to number length on HF device side, but we had better limit it to less than 40 bytes.

2.3.6 Caller ID Display Length

There is no limit to number length on HF device side, and it depends on mobile phone side, but we had better limit it to less than 40 bytes.

3 AT Command Format

This chapter introduces the AT commands' format. Some responses will not be returned immediately. Where applicable, an approximate delay time will be included to notify the response delay.

Please note that a full piece of AT command, AT response or AT indication must be tailed with "\r" (0x0d).

3.1Command Format

<at-command-object>::={

<at-command-header><SPACE>

<at-command-body><SPACE>

[<at-command-parameter>[COMMA]]*

<CR>

}

<at-command-header>::=AT+B

<at-command-body>::='character set, upper case'

<at-command-parameter>::=' number set and character set, be separated by comma, the last parameter need not comma-tailed'

3.2Response Format

<at-response-object>::={

<at-response-header><SPACE>

<at-response-body><SPACE>

[<at-response-parameter><COMMA>]*

<CR>

}

<at-response-header>::=AT-B

<at-response-body>::='character set, upper case'

<at-response-parameter>::='number set and character set, be separated by comma, the last parameter need not comma-tailed'

3.3Indication Format

<at-indication-object>::={

<at-indication-header><SPACE>

<at-indication-body><SPACE>

[<at-indication-parameter><COMMA>]*

<CR>

}

<at-indication-header>::=AT-B

<at-indication-body>::='character set, upper case, length'

<at-indication-parameter>::=' number set and character set, be separated by comma, the last parameter need not comma-tailed'

4 Generic AT Command Definition

This chapter introduces the generic AT commands' definition, including a brief description of commands' syntax, responses and examples. All commands listed in this chapter are profile-independent.

4.1GVER

The GVER command is used to get the version of the controller unit firmware.

Command	**AT+B GVER**

Response	**AT-B GVER [ver]**
Parameters	N/A
Note	N/A

4.2GLBD

The GLBD command is used to get the local Bluetooth device address.

Command	**AT+B GLBD**	
Response	Succeed: **AT-B GLBD 0,[bd]** Failed: **AT-B GLBD 1,0**	
Parameters	bd	Local *Bluetooth* device address.
Note	bd is comprised of 12 bytes hexadecimal characters.	

4.3GLDN

The GLDN command is used to get the local device name.

Command	**AT+B GLDN**	
Response	Succeed: **AT-B GLDN 0,[name]** Failed: **AT-B GLDN 1,**	
Parameter	name	Device name.
Note	N/A	

4.4SLDN

The SLDN command is used to set the local device name.

Command	**AT+B SLDN [name]**
Response	Succeed: **AT-B SLDN 0** Failed: **AT-B SLDN 1**

Parameter	name	Device name. UTF-8 format.
Note	The length of name can be up to 31 bytes at maximum.	

4.5GRDN

The GRDN command is used to get the specific remote device name.

Command	**AT+B GRDN [bd]**	
Response	Succeed: **AT-B GRDN 0,[bd],[name]** Failed: **AT-B GRDN 1,[bd],**	
Parameters	bd	Remote *Bluetooth* device address.
	name	Remote device name.
Note	bd is comprised of 12 bytes hexadecimal characters.	

4.6SPIN

The SPIN command is used to set the local PIN code.

Command	**AT+B SPIN [pin]**	
Response	Succeed: **AT-B SPIN 0** Failed: **AT-B SPIN 1**	
Parameters	pin	The new PIN code. The default PIN code is 0000.
Note	N/A	

4.7GPIN

The FPIN command is used to get the local PIN code.

Command	**AT+B GPIN**
Response	Succeeded: **AT-B GPIN 0,[pin]** Failed: **AT-B GPIN 1,0**

Parameters	pin	The local PIN code.
Note	N/A	

4.8GPRD

The GPRD command is used to get the paired record which stored in local BT module.

Command	**AT+B GPRD**	
Response	**AT-B GPRD [total],[index],[bd]** If no paired record found: **AT-B GPRD 0,0, 000000000000** If paired records found(n>=1): **AT-B GPRD n,0,bd** **AT-B GPRD n,1,bd** **……** **AT-B GPRD n,n-1,bd**	
Parameters	total	Total paired devices in the controller unit.
	index	Index of the total parameter
	bd	Remote *Bluetooth* device address.
Note	bd is comprised of 12 bytes hexadecimal characters.	

4.9DPRD

The DPRD command is used to delete the specified BD address paired record.

Command	**AT+B DPRD [bd]**
Response	**AT-B DPRD [result],[bd]** If delete all paired device records: **AT-B DPRD 0,000000000000** If delete a paired device record with the specified *Bluetooth* device address: **AT-B DPRD 0,[bd]**

	If failed to delete a paired device record (For instance, not found in device paired list): **AT-B DPRD 1,[bd]**	
Parameter	bd	If the bd parameter in the command equals to"000000000000", all paired device records will be deleted; If the bd parameter in the command does not equals to "000000000000", the paired device record which *Bluetooth* address equals to bd will be deleted.
	result	0: succeeded; 1: failed.
Note	bd is comprised of 12 bytes hexadecimal characters.	

4.10 INQU

The INQU command will cause local device to discover other nearby Bluetooth devices.

Command	**AT+B INQU [op]**	
Response	If op=1 and any nearby device was found: **AT-B INQR [bd],[class]** If op=2 and any nearby device was found: **AT-B INQR [bd],[class],[name]** If the inquiry process finished: **AT-B INQC**	
Parameters	op	0: stop the inquiry procedure. 1: start searching nearby *Bluetooth* devices, and return devices' address and class of device 2: start searching nearby Bluetooth devices, and return devices' address, class of device and device name
	bd	Remote *Bluetooth* device address.12 bytes hexadecimal characters
	class	Class of device
	name	Device name
Note	Default inquiry time is 12.8s, default response number of device is 8, when either of conditions comes, the inquiry will terminate.	

4.11 PAIR

The PAIR command is used to pair with remote device by BD address.

Command	**AT+B PAIR [bd]**	
Response	**AT-B PAIR [result],[bd]**	
Parameters	bd	Remote *Bluetooth* device address
	result	Pairing results, where 0: Authentication was successful; 1: Authentication timed out; 2: Authentication failed; 3: Authentication failed due to too many repeat attempts; 4: Authentication failed as remote device is not allowing pairing; 5: Authentication failed as unit keys are not supported; 6: Authentication failed as simple pairing is not supported; 7: Authentication failed as host is already busy pairing.
Note	bd is comprised of 12 bytes hexadecimal characters.	

4.12 SCAN

The SCAN command is used to set the scan mode.

Command	**AT+B SCAN [mode]**	
Response	Succeeded: **AT-B SCAN 0** Failed: **AT-B SCAN 1**	
Parameters	mode	Scan mode, where 0: No scans enabled; 1: Enable Inquiry scan and Page scan disabled; 2: Enable page scan and Inquiry scan disabled; 3: Enable inquiry and page scan.

Note	Inquiry scan means the controller unit can be inquired by other *Bluetooth* devices. Page scan means the controller can be connected by other *Bluetooth* devices. Default settings is mode=3.

4.13 EDFU

The EDFU command is used to make the module enter the DFU mode.

Command	**AT+B EDFU**
Response	**AT-B EDFU 0**
Parameters	N/A
Note	This command will force a warm reset and make the module enter the DFU mode.

4.14 UART

The UART command is used to set serial communication parameters of the module's UART controller.

Command	**AT+B UART [baud],[stop],[parity]**	
Response	Succeeded: **AT-B UART 0** Otherwise: **AT-B UART 1**	
Parameters	baud	Supported baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, and 1382400.
	stop	0: 1bit stop bit; 1: 2 bit stop bit.
	parity	0: No checksum; 1: Odd checksum; 2: Even checksum.
Note	N/A	

4.15 SCOD

This command is used to set class of device of the local device to the supplied value.

Command	**AT+B SCOD [cod]**	
Response	Succeeded: **AT-B SCOD 0** Otherwise: **AT-B SCOD 1**	
Parameters	cod	Class of device, ASCII code characters.
Note	Firmware's default COD is not HID devices. Therefore, if HID profile is enabled, it needs to set the module's COD to Peripheral (mouse, joystick, keyboard, ...) since some mobile phones may refuse to establish HID (keyboard or mouse) connections with *Bluetooth* devices which COD is not set to HID devices. COD (Major and minor device class fields) refers to the link: https://www.bluetooth.org/en-us/specification/assigned-numbers/baseband For instance, the major device class field (bit 12~bit8) should be set to 00101, the minor device class field (bit7~bit6) could be set to 01 (keyboard), a0(pointing device). HID Keyboard COD should be set to 0540, HID Mouse COD should be set to 0580.	

4.16 GCOD

This command is used to get the local class of device.

Command	**AT+B GCOD**	
Response	**AT-B GCOD [status],[cod]**	
Parameters	status	0: Succeeded; Else: Failed.
	cod	Class of device, ASCII code characters.
Note	N/A	

4.17 SPRO

Command	**AT+B SPRO [profile_support_mask]**	
Indication	**Succeeded: AT-B SPRO 0** **Failed: AT-B SPRO 1**	
Parameters	profile_support_mask	bit 0: mode, 0:sink mode; bit 1: HFP, 0: disable, 1:enable; bit 2: A2DP, 0: disable, 1:enable; bit 3: AVRCP, 0: disable, 1:enable; bit 4: PBAP Client, 0: disable, 1:enable; bit 5: MAP Client, 0: disable, 1:enable; bit 6: OPP Client, 0: disable, 1:enable; bit 7: OPP Server, 0: disable, 1:enable; bit 8: SPP, 0: disable, 1:enable; bit 9: PBAP Server, 0: disable, 1:enable;
Note	1. For instance, 02 represents HFP is enabled; 112 represents HFP, PBAP and SPP are enabled; 512 represents HFP, PBAP, SPP and HID mouse are enabled; 912 represents HFP, PBAP, SPP and HID keyboard are enabled; 400 represents HID mouse is enabled; 800 represents HID keyboard is enabled. 2. If either HID mouse or HID keyboard is enabled, the module will support HID streaming method. Under this circumstance, it doesn't need to send Keyboard/mouse input reports that are formatted as AT+B command, but it can directly send report contents. See chapter **错误!未找到引用源。** and **错误!未找到引用源。**.	

4.18 GCTST

Command	**AT+B GCTST [profile]**	
Indication	**AT-B GCTST [profile],[state],[bdaddr]**	
Parameters	profile	1: HFP 2: A2DP

		3: AVRCP 4: PBAP 5: SPP 12: BLE
	state	Profile state. 1: the corresponded connection of the inquired profile doesn't exist, this parameter will return 0 and the Bluetooth address will return 000000000000
	bdaddr	The connected remote device Bluetooth address. If the corresponded connection of the inquired profile doesn't exist, the returned Bluetooth address will be 000000000000
Note	This command is used to get connection status. For instance, To inquire HFP connection state and HFP connection exists, Command: AT+B GCTST 1\r Indication: AT-B 1,3,9CC1729DCCC\r (HFP connection is established. The remote device's Bluetooth address is 9CC1729DCCC) To inquire HFP connection state and HFP connection doesn't exist, Command: AT+B GCTST 1\r Indication: AT-B 1,1,000000000000\r	

4.19 GPRL

Command	**AT+B GPRL**	
Indication	**AT-B GPRL [total],[index],[addr],[name]**	
Parameters	total	Total number of paired devices
	index	Paired device index. Starting from 0.
	addr	Paired device address
	name	Paired device name
Note	This command is used to get paired device list.	

4.20 ROUTE

This command is used to set route of the audio system.

Command	**AT+B ROUTE [media],[input],[output],[stereo]**	
Response	**AT-B ROUTE [status]**	
Parameters	status	0: Succeeded; Else: Failed.
	media	Media type 1, audio(music) 2,voice
	Input	Input source 0, ADC(analog input) 1, I2S 2, SPDIF
	Output	Audio output type 0,1, DAC(internal codec) 2, I2S 3, SPDIF
	Stereo	0, Mono 1, Stereo
Note	1 The route will take effect next time.	

4.21 GROUTE

This command is used to get route of the audio system.

Command	**AT+B GROUTE [media]**	
Response	**AT-B GCOD [media],[input],[output],[stereo]**	
Parameters	Media	Media type

		1, audio(music)
	Input	Input source, reserved;
	Output	Audio output type 0, None 1, DAC(internal codec) 2, I2S
	Stereo	0, Mono 1, Stereo
Note	**Input** parameter is reserved for future use, default value is 0.	

4.22 I2CR

This command is used to receive data across I2C interface

Command	**AT+B I2CR [r_addr], [length]**	
Response	**AT-B I2CR [stauts],[length], [data]**	
Parameters	status	0: Succeeded; Else: Failed.
	R_addr	Device Address to read
	Length	Length of data
	Data	Data received
Note	N/A	

4.23 I2CW

This command is used to transmit data across I2C interface.

Command	**AT+B I2CW [w_addr],[length], [data]**
Response	**AT-B I2CW [status]**

Parameters	status	0: Succeeded; Else: Failed.
	W_addr	Device address to write.
	Length	Length of data to write
	Data	Data to write.
Note	N/A	

4.24 I2SC

This command is used to set the configuration of I2S Interface.

Command	**AT+B I2SC [master],[justify], [delay],[bits],[scale]**	
Response	**AT-B I2SC [status]**	
Parameters	status	0: Succeeded; Else: Failed.
	Master	0, Slave mode 1, Master mode, Clock and sync will be generated by I2S hardware.
	Justify	I2S format 0, left justified 1, right justified
	Delay	Left justified format 0, MSB of SD data occurs in the first SCK period following WS transition 1 MSB of SD data occurs in the second SCK period
	Bits	bits per sample
	Scale	master clock frequency scaling factor clock rate = sample rate * scale

Note	The configuration will take effect next time.

4.25 I2SG

This command is used to get the configuration of I2S interface.

Command	**AT+B I2SG**	
Response	**AT-B I2SG [master],[justify], [delay],[bits],[scale]**	
Parameters	Master	0, Slave mode 1, Master mode, Clock and sync will be generated by the I2S hardware.
	Justify	I2S format 0, left justified 1, right justified
	Delay	Left justified format 0, MSB of SD data occurs in the first SCK period following WS transition 1 MSB of SD data occurs in the second SCK period
	Bits	bits per sample
	Scale	master clock frequency scaling factor clock rate = sample rate * scale
Note	N/A	

4.26 PIOSETPIN

This command is used to modify the contents of the PIO data output register.

Command	**AT+B PIOSETPIN [mask], [bits]**
Response	**AT-B PIOSETPIN [result]**

Parameters	result	A 32 bit mask. If any bit in this mask is high then that PIO could not be driven to the level specified
	mask	Each bit in the mask corresponds to a PIO line. Bits set to 1 in this mask will be modified. Bits set to 0 in this mask will not be modified.
	bits	Each bit in the bits value corresponds to a PIO line. Bits set to 1 in this value will result in that PIO line being driven high. Bits set to 0 in this value will result in that PIO line being driven low.
Note	1 PIO pins must be set to outputs via **AT+B PIOSETDIR** before they can be driven high or low through this command. 2 The upper 16 PIOs must be mapped in before they can be used through AT+B PIOSETMAP.	

4.27 PIOGETPIN

This command is used to get the contents of PIO data register.

Command	**AT+B PIOGETPIN**	
Response	**AT-B PIOGETPIN [result]**	
Parameters	result	A 32 bit value. Each bit in the result value corresponds to a PIO line: 1, high 0, low
Note	N/A	

4.28 PIOSETDIR

This command is used to set PIOs as inputs or outputs.

Command	**AT+B PIOSETDIR [mask],[dir]**	
Response	**AT-B PIOSETDIR [result]**	
Parameters	result	A 32 bit mask. If any bit in this mask is high then that PIO could

		not be set to the direction specified.
	mask	Each bit in the mask corresponds to a PIO line. Bits set to 1 in this mask will be modified. Bits set to 0 in this mask will not be modified.
	dir	Each bit in the dir value corresponds to a PIO line. Bits set to 1 in this value will result in that PIO line being configured as an output. Bits set to 0 in this value will result in that PIO line being configured as an input.
Note	1 The upper 16 PIOs must be mapped in before they can be used through AT+B PIOSETMAP.	

4.29 I2SG

This command is used to read whether PIOs are set as inputs or outputs.

Command	**AT+B I2SG**	
Response	**AT-B I2SG [result]**	
Parameters	result	A 32 bit value. Each bit in the result value corresponds to a PIO line. Bits set to 1 means that PIO line is configured as an output. Bits set to 0 means it is configured as an input.
Note	N/A	

4.30 PIOSETMAP

This command is used to make usual function of chip pins behave as PIOs.

Command	**AT+B PIOSETMAP [mask],[bits]**	
Response	**AT-B PIOSETMAP [result]**	
Parameters	result	A 32 bit mask. If any bit in this mask is high then that PIO could not be mapped or unmapped.
	mask	Each bit in the mask corresponds to a PIO line. Bits set to 1 in this mask will be modified. Bits set to 0 in this mask will not be modified.

	bits	Each bit corresponds to a PIO line. A bit set to 1 will cause a (non-PIO) chip pin to be behave as the corresponding PIO. A bit set to 0 will result in any mapped pin being returned to its original function.
Note	For I50e the PIO lines map to other pins as follows: (PIO 0-15) have no mapping. They are always PIO 0-15. They can be configured as inputs or outputs. (PIO 16) maps to PCM_DATA. This can be configured as an input or an output. (PIO 17) maps to PCM_SYNC. This can be configured as an input or an output. (PIO 18) maps to UART_DATA_OUT. This can be configured as an input or an output. (PIO 19) maps to PCM_CLK_OUT. Set this to output to the PCM_CLK pin. This line is output only. (PIO 20) maps to AIO0. (PIO 21) maps to AIO1. PIO lines above 21 map to nothing and cannot be mapped or written. For i1107e the PIO lines map to other pins as follows: (PIO 0-12) have no mapping. They are always PIO 0-12.They can be configured as inputs or outputs. The smaller packages such as Chip Scale Package (CSP) does not have PIO8..12. (PIO 13..15) may be mapped if required. The exact signal routing is dependent on which package is being used. On smaller packages, such as CSP, you must map PIO13-15 if you want PIO instead of UART UART_RX, UART_TX and UART_CTS. On the BGA package PIO13..15 have their own pins, but if mapped, will be connected to the UART_RX, UART_TX and UART_CTS pins as well. Whether mapped or not, these PIO pins may be configured as inputs or outputs. For each pin, if mapped and set as output, both (UART and PIO) pins are driven. If mapped and set as input, the UART pin is connected and the PIO pin is n/c. (PIO 16) maps to the UART_RTS pin. This can be configured as an input or an output.	

	(PIO 17) maps to the PCM_IN pin. This can be configured as an input or an output. (PIO 18) maps to the PCM_OUT pin. This can be configured as an input or an output. (PIO 19) maps to the PCM_SYNC pin. This can be configured as an input or an output. (PIO 20) maps to the PCM_CLK pin. This can be configured as an input or an output. (PIO 21) maps to the SQIF Flash Clock pin. This can be configured as an input or an output. (PIO 22) maps to the SQIF RAM Clock pin. This can be configured as an input or an output. (PIO 23) maps to the SQIF Flash CS pin. This can be configured as an input or an output. (PIO 24) maps to the SQIF RAM CS pin. This can be configured as an input or an output. (PIO 25) maps to the SQIF DB0 pin. This can be configured as an input or an output. (PIO 26) maps to the SQIF DB1 pin. This can be configured as an input or an output. (PIO 27) maps to the SQIF DB2 pin. This can be configured as an input or an output. (PIO 28) maps to the SQIF DB3 pin. This can be configured as an input or an output. PIO lines above 28 map to nothing and cannot be mapped or written.

4.31 PIOGETMAP

This command is used to get which PIO lines have been mapped to chip pins.

Command	**AT+B PIOGETMAP**	
Response	**AT-B PIOGETMAP [result]**	
Parameters	result	A 32 bit value showing which PIO lines have been mapped to chip pins.

Note	N/A

4.32 GOTA

This command is used to obtain OTA status.

Command	**AT+B GOTA**	
Response	**AT-B GOTA [status]**	
Parameters	status	0, default, OTA is disabled. 1, OTA is enabled.
Note	N/A	

4.33 SOTA

This command is used to enable/disable OTA function.

Command	**AT+B SOTA [en]**	
Response	**AT-B SOTA [status]**	
Parameters	en	0, disable OTA 1, enable OTA
	status	0, success 1, failed
Note	OTA and SPP data transmission are exclusive.	

5 Generic Indication Definition

5.1INIT

The INIT indication is used to inform the host unit if the Bluetooth initialization is successfully completed.

Indication	**AT-B INIT [status]**	
Parameters	status	0: succeeded; 1: failed.
Note	N/A	

5.2ROLE

The ROLE indication is used to inform the host of current role in the specific connection.

Indication	**AT-B ROLE [role]**	
Parameters	role	0: master; 1: slave; 2: role doesn't care.
Note	N/A	

5.3SSPPIN

The SSPPIN indication is used to inform the SSP PIN code.

Indication	**AT-B SSPPIN [pin]**	
Parameters	pin	SSP PIN code
Note	N/A	

6 HFP AT Command Definition

This chapter introduces the HFP (HF Unit Role) relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

6.1HFP Status

This chapter introduces the defined HFP status.

6.1.1 HFP Lib Status

```
typedef enum
{
    hfp_success = 0,                        /*! Success.*/
    hfp_fail = 0x01,                        /*! Failure.*/
    hfp_ag_failure= 0x02,                   /*! Failure - AG failure.*/
    hfp_no_connection_to_phone= 0x03,       /*! Failure - No connection to phone.*/
    hfp_operation_not_allowed= 0x04,        /*! Failure - Operation not allowed.*/
    hfp_operation_not_supported= 0x05,      /*! Failure - Operation not supported.*/
    hfp_ph_sim_pin_required= 0x06,          /*! Failure - PH-SIM PIN required.*/
    hfp_sim_not_inserted= 0x07,             /*! Failure - SIM not inserted.*/
    hfp_sim_pin_required= 0x08,             /*! Failure - SIM PIN required.*/
    hfp_sim_puk_required= 0x09,             /*! Failure - SIM PUK required.*/
    hfp_sim_failure= 0x0a,                  /*! Failure - SIM failure.*/
    hfp_sim_busy= 0x0b,                     /*! Failure - SIM busy.*/
    hfp_incorrect_password= 0x0c,           /*! Failure - Incorrect password.*/
    hfp_sim_pin2_required= 0x0d,            /*! Failure - SIM PIN2 required.*/
    hfp_sim_puk2_required= 0x0e,            /*! Failure - SIM PUK2 required.*/
    hfp_memory_full= 0x0f,                  /*! Failure - Memory full.*/
    hfp_invalid_index= 0x10,                /*! Failure - Invalid index.*/
    hfp_memory_failure= 0x11,               /*! Failure - Memory failure.*/
    hfp_text_string_too_long= 0x12,         /*! Failure - Text string too long.*/
    hfp_invalid_chars_in_text_string= 0x13, /*! Failure - Invalid characters in text string.*/
    hfp_dial_string_too_long= 0x14,         /*! Failure - Dial string too long.*/
    hfp_invalid_chars_in_dial_string= 0x15, /*! Failure - Invalid characters in dial string.*/
    hfp_no_network_service= 0x16,           /*! Failure - No network service.*/
    hfp_network_not_allowed= 0x17,          /*! Failure - Network not allowed, emergency calls only.*/
    hfp_timeout=0x1d,                       /*! Failure - Timed out waiting for AG response */
    hfp_network_no_carrier,                 /*! Failure – No Carrier */
    hfp_network_busy,                       /*! Failure - BUSY */
    hfp_network_no_answer,                  /*! Failure – NO ANSWER */
    hfp_network_delayed,                    /*! Failure - DELAYED */
    hfp_network_blacklisted                 /*! Failure - BLACKLISTED */
} hfp_lib_status;
```

6.1.2 HFP Connect Status

```
typedef enum
{
    hfp_connect_success,                    /*! Successful connection.*/
```

```
    hfp_connect_sdp_fail,                       /*! Unsuccessful due to a service search failure.*/
    hfp_connect_slc_failed,                     /*! Unsuccessful due to a service level connection failure.*/
    hfp_connect_failed_busy,                    /*! Unsuccessful due to service level connection already
                                                established.*/
    hfp_connect_failed,                         /*! Unsuccessful due to RFCOMM connection failing to be
                                                established.*/
    hfp_connect_server_channel_not_registered,  /*! Unsuccessful due to attempt to
                                                connect to unallocated server
                                                channel.*/
    hfp_connect_timeout,                        /*! Unsuccessful due to connection attempt timing out.*/
    hfp_connect_rejected,                       /*! Unsuccessful due to remote device rejecting
                                                connection.*/
    hfp_connect_normal_disconnect,              /*! Unsuccessful due to remote device terminating the
                                                 connection.*/
    hfp_connect_abnormal_disconnect,            /*! Unsuccessful due to an abnormal disconnect while
                                                 establishing an rfcomm connection.*/
    hfp_connect_fail_bad_params,                /*! Connection failed due to bad parameters supplied by the
                                                 application. */
} hfp_connect_status;
```

6.1.3 HFP Disconnect Status

```
typedef enum
{
    hfp_disconnect_success,                     /*! Successful disconnection.*/
    hfp_disconnect_link_loss,                   /*! Unsuccessful due to abnormal link loss.*/
    hfp_disconnect_no_slc,                      /*! Unsuccessful due to no current connection.*/
    hfp_disconnect_timeout,                     /*! Unsuccessful due to RFCOMM connection attempt
                                                timeout.*/
    hfp_disconnect_error,                       /*! Unsuccessful due to RFCOMM connection attempt
                                                error.*/
} hfp_disconnect_status;
```

6.2HFCONN

The HFCONN command is used to create a HFP connection with the remote device.

Command	**AT+B HFCONN [bd]**
Response	**AT-B HFCONN [status],[bd],[profile]**

Parameters	status	Values in the chapter 6.1.2.
	bd	Remote *Bluetooth* device address.
	profile	Profile type, where 0: Not HSP/HFP; 1: Headset Profile; 2: Hands-free Profile.
Note	bd is comprised of 12 bytes hexadecimal characters.	

6.3HFDISC

The HFDISC command is used to disconnect the HFP connection with the remote device.

Command	**AT+B HFDISC**	
Response	**AT-B HFDISC [status],[bd]**	
Parameters	status	Values in the chapter 6.1.3.
	bd	Remote *Bluetooth* device address.
Note	N/A	

6.4HFANSW

The HFANSW command is used to answer the incoming call.

Command	**AT+B HFANSW**	
Response	**AT-B HFANSW [status]**	
Parameters	status	Values in the chapter 6.1.1.
Note	N/A	

6.5HFCHUP

The HFCHUP command is used to reject the incoming call, hang up the active call or cancel the dialing out call.

Command	**AT+B HFCHUP**	
Response	**AT-B HFCHUP [status]**	
Parameters	status	Values in the chapter 6.1.1.
Note	N/A	

6.6HFDIAL

The HFDIAL command is used to dial a phone number, for Hands-Free profile only.

Command	**AT+B HFDIAL[type], [num]**	
Response	**AT-B HFDIAL [status],[type]**	
Parameters	type	Call type, where 0: dial the supplied number; 1: perform a last number redial.
	num	The dialed out number. The maximum length is 40 bytes.
	status	Values in the chapter 6.1.1.
Note	N/A	

6.7HFDTMF

The HFDTMF command is used to transmit a DTMF code to the AG, for Hands-Free profile only.

Command	**AT+B HFDTMF [key]**
Response	**AT-B HFDTMF [status]**

Parameters	key	DTMF key, including “0-9”, A, B, C, D, *, #.
	status	Values in the chapter 6.1.1.
Note	N/A	

6.8HFCTRS

The HFCTRS command is used to transfer audio from/to remote when a call is ongoing.

Command	**AT+B HFCTRS**	
Response	**AT-B HFCTRS [status]**	
Parameters	status	Values in the chapter 6.1.1.
Note	The host unit will receive the audio connection on/off indication when this command is successfully executed.	

6.9HFMCAL

The HFMCAL command is used to change three-way calling status (active or held). This command will be successfully executed when the HFP instance is in either hsActiveCall or hsTWCallWaiting status.

Command	**AT+B HFMCAL [op]**	
Response	**AT-B HFMCAL [status],[op]**	
Parameters	op	Operation code, where 0: MultipleCallsReleaseHeldOrRejectWaiting; 1: MultipleCallsReleaseActiveAcceptOther; 2: MultipleCallsHoldActiveAcceptOther.
	status	Values in the chapter 6.1.1
Note	N/A	

6.10 HFCLCC

The HFCLCC command is used to get current calls list of AG side, for Hands-free profile only.

Command	**AT+B HFCLCC**	
Response	If one or more current calls found: **AT-B HFCCIN [status],[call_idx],[direction], [mode],[multiparty],[number_type],[number]** When the command finished: **AT-B HFCLCC [status]**	
Parameters	call_idx	Call index, defined by AG.
	direction	AG originated call indicator, where 0: Call from AG to network; 1: Call from network to AG.
	status	Call status, where 0: Call is currently active; 1: Call is currently held; 2: Call is being dialed - mobile originated only; 3: Call is alerting - mobile originated only; 4: Call is incoming - mobile terminated only; 5: Call is waiting - mobile terminated only.
	mode	Call mode, where 0: voice call; 1: data call; 2: fax call.
	multiparty	Call multiparty indicator, where 0: Call is not multiparty; 1: Call is multiparty.
	number_type	Number type, where

		0: Type of number is unknown; 1: Number is an international number; 2: Number is a national number; 3: Number is a network specific number; 4: Number is a dedicated access, short code.
	number	Phone number
	status	Values in the chapter 6.1.1.
Note	N/A	

6.11 HFSVGS

The HFSVGS command is used to send speaker volume to AG side, for BT module part, the speaker gain is also changed.

Command	**AT+B HFSVGS [vol]**	
Response	**AT-B HFSVGS [status],[vol]**	
Parameters	vol	Speaker volume, where ranges from 0 to 15.
	status	Values in the chapter 6.1.1
Note	If the connection is a HSP SLC, this command can be sent when the audio connection is ongoing. If the connection is a HFP SLC, this command can be sent when the connection status equals to or greater than hsConnected.	

6.12 HFGVGS

The HFGVGS command is used to get speaker volume.

Command	**AT+B HFGVGS**	
Response	**AT-B HFGVGS [vol]**	
Parameters	vol	Speaker volume, where ranges from 0 to 15.
Note	If the connection is a HSP SLC, this command can be sent when the audio	

	connection is ongoing. If the connection is a HFP SLC, this command can be sent when the connection status equals to or greater than hsConnected.

6.13 HFMUTE

The HFMUTE command is used to mute or unmute the microphone when a call is ongoing. When an audio connection is established, the default setting is MIC muted.

Command	**AT+B HFMUTE [op]**	
Response	**AT-B HFMUTE [status]**	
Parameters	op	0: unmute; 1: mute.
	status	0: succeeded; 1: failed.
Note	N/A	

6.14 HFSCFG

The HFSCFG command is used to enable/disable the reconnect function, and enable/disable local ring tone.

Command	**AT+B HFSCFG [mask],[config]**	
Response	**AT-B HFSCFG [status]**	
Parameters	mask	1: enable/disable the reconnect function; 2: enable/disable local ring tone.
	config	When mask=1, if config=0: disable the reconnect function; if config=1, enable the reconnect function. When mask=2, if config=0: enable local ring tone; if config=1: disable local ring tone.
	status	0: succeeded;

		1: failed.
Note	N/A	

6.15 HFGCFG

The HFGCFG command is used to query if the reconnect function is enabled or disabled, and if the local ring tone is used.

Command	**AT+B HFGCFG [mask]**	
Response	**Succeeded: AT-B HFGCFG 0,[config]** **Failed: AT-B HFGCFG 1,0**	
Parameters	mask	1: query if the reconnect function is enabled or disabled; 2: query if the local ring tone is used.
	config	When mask=0, if config=0: the reconnect function is disabled; if config=1, the reconnect function is enabled. When mask=1, if config=0: local ring tone is used; if config=1: local ring tone isn't used.
Note	N/A	

6.16 HFBVRA

The HFBVRA command is used to enable/disable mobile phone's voice recognition feature, such as iPhone Siri.

Command	**AT+B HFBVRA [enable]**	
Response	**AT-B HFBVRA [hfp_lib_status]**	
Parameters	enable	1: enable voice recognition; 0: disable voice recognition;
	hfp_lib_status	Refer to chapter 6.1.1
Note	N/A	

6.17 HFCOPS

The HFCOPS command is used to Get the network operator for the AG.

Command	**AT+B HFCOPS**	
Response	**Succeeded: AT-B HFCOPS [mode],[operator]** **Failed: AT-B HFCOPS 1**	
Parameters	mode	Network operator selection mode, currently not used, so ignore it.
	operator	Operator name string, shall not exceed 16 character.
Note	N/A	

6.18 HFMCAL

The HFMCAL command is used to operate three-way calling.

Command	**AT+B HFMCAL [op],[index]**	
Response	**AT-B HFMCAL 0**	
Parameters	**op**	Operation code, in where: 0 : Releases all held calls or sets User Determined User Busy (UDUB) for a waiting call. 1: Releases all active calls (if any exist) and accepts the other (held or waiting) call. 2 : Places all active calls (if any exist) on hold and accepts the other (held or waiting) call. 3: Adds a held call to the conversation. 4: Connects the two calls and disconnects the subscriber from both calls (Explicit Call Transfer). Support for this value and its associated functionality is optional for the HF.
	index	This parameter is returned in call_idx parameter of AT-B HFCCIN indication. Only when the op parameter equals to either 1 or 2, index parameter has meaningful value. When the op parmater

		equals to other values, index parameter's value can be ignored. When op=1, <idx> = Releases specified active call only (<idx>); When op=2, <idx> = Request private consultation mode with specified call (<idx>). (Place all calls on hold EXCEPT the call indicated by <idx>.)
Note	N/A	

7 HFP Indication Definition

This chapter introduces the HFP (HF Unit Role) relevant indications' definition.

7.1 HFSTAT

The HFSTAT indication is used to inform the host unit when the HFP status is changed.

Indication	**AT-B HFSTAT [state]**	
Parameters	state	HFP status, where 1: HfpTLReady; 2: HfpTLSlcConnecting; 3: HfpTLSlcConnected; 4: HfpTLIncomingCallEstablish; 5: HfpTLOutgoingCallEstablish; 6: HfpTLActiveCall; 7: HfpTLTWCalling: there is a current call. A new incoming call arrives, and this new incoming call is in waiting status. 8: HfpTWCallOnHold: one call is active and another call is in held status. 9: HfpTWMulticall: multiple calling 10: HfpCallOnHoldNoActive: all calls are in held status.
Note	N/A	

7.2HFCONN

The HFCONN indication is used to inform the host unit when it initializes a HFP connection with the local device.

Indication	**AT-B HFCONN [status],[bd], [profile]**	
Parameters	status	Values in the chapter 6.1.2.
	bd	Remote *Bluetooth* device address
	profile	Profile type, where 0: Not HSP/HFP; 1: Headset Profile; 2: Hands-free Profile.
Note	N/A	

7.3HFDISC

The HFDISC indication happens when the remote device disconnect the HFP connection.

Indication	**AT-B HFDISC [status],[bd]**	
Parameters	status	Values in the chapter 6.1.3
	bd	Remote *Bluetooth* device address
Note	N/A	

7.4HFRING

The HFRING indication is used to inform the host unit when HFP ring comes. The host unit shall turn on the audio path when receives this indication, and turn off the audio path when the HFP status changes HfpTLSlcConnected.

Indication	**AT-B HFRING**
Parameters	N/A

Note	N/A

7.5HFIBRN

The HFIBRN indication is used to inform the host unit that HFP in-band ring feature turns on or off.

Indication	**AT-B HFIBRN [inbandring]**	
Parameters	inbandring	0: in-band ring tone is off; 1: in-band ring tone is on.
Note	N/A	

7.6HFAUDIO

The HFAUDIO indication is used to inform the host unit that the HFP audio connection is on or off.

Indication	**AT-B HFAUDIO [op]**	
Parameters	op	0: the HFP audio connection is off; 1: the HFP audio connection is on.
Note	N/A	

7.7HFCLIP

The HFCLIP indication is used to inform the host unit of the incoming call's caller ID.

Indication	**AT-B HFCLIP [callerid]**	
Parameters	callerid	Incoming call's caller ID. There is no limit to number length on HF device side, and it depends on mobile phone side, but we had better limit it to less than 40 bytes.
Note	N/A	

7.8HFCCWA

The HFCCWA indication is used to inform the host unit when the second incoming call's caller ID.

Indication	**AT-B HFCCWA [callerid]**	
Parameters	callerid	Incoming call's caller ID. There is no limit to number length on HF device side, and it depends on mobile phone side, but we had better limit it to less than 40 bytes.
Note	N/A	

7.9HFNUML

The HFNUML indication is used to inform the host unit of the subscriber number of the AG side when the SLC connection is established.

Indication	**AT-B HFNUML [number]**	
Parameters	number	The subscriber number of the AG side. There is no limit to number length on HF device side, and it depends on mobile phone side, but we had better limit it to less than 40 bytes.
Note	N/A	

7.10 HFNUMC

The HFNUMC indication is used to inform the host that the query of subscriber number is completed.

Indication	**AT-B HFNUMC [status]**	
Parameters	status	Values in the chapter 6.1.1.
Note	N/A	

7.11 HFSGNL

The HFSGNL indication is used to inform the host unit of the signal strength of

the AG side.

Indication	**AT-B HFSGNL [signal]**	
Parameters	signal	Signal strength indicator, where ranges from 0 to 5.
Note	N/A	

7.12 HFROAM

The HFROAM indication is used to inform the host unit of the roaming status of the AG side.

Indication	**AT-B HFROAM [roam]**	
Parameters	roam	Roaming status indicator, where: 0: roaming is not active; 1: roaming is active.
Note	N/A	

7.13 HFBATC

The HFBATC indication is used to inform the host unit of the battery charger status of the AG side.

Indication	**AT-B HFBATC [battchg]**	
Parameters	battchg	Battery charge indicator of AG, where ranges from 0 to 5.
Note	N/A	

7.14 HFVGSI

The HFVGSI indication is used to inform the host unit of the current speaker volume of the AG side.

Indication	**AT-B HFVGSI [spkvol]**	
Parameters	spkvol	Speaker volume, where ranges from 0 to 15.

Note	N/A

7.15 HFVGMI

The HFVGMI indication is used to inform the host unit of the current microphone volume of the AG side.

Indication	**AT-B HFVGMI [micvol]**	
Parameters	micvol	Microphone volume, where ranges from 0 to 15.
Note	N/A	

7.16 HFSRVC

This indication is used to inform the host unit of a change in the service indicator's status.

Indication	**AT-B HFSRVC [service]**	
Parameters	service	The new value of the service indicator.
Note	N/A	

7.17 HFCHLD

This indication is used to inform the host unit of the call held status of AG side.

Indication	**AT-B HFCHLD [callheld]**	
Parameters	callheld	*Bluetooth* proprietary call hold status indicator. Support for this indicator is mandatory for the AG, optional for the HF. Possible values are as follows: 0: No calls held; 1: Call is placed on hold or active/held calls swapped (The AG has both and active AND a held call); 2: Call on hold, no active call.
Note	N/A	

7.18 HFCODEC

The HFCODEC indication is used to inform the host codec negotiated with the remote AG.

Indication	**AT-B HFCODEC [codec_id]**	
Parameters	codec_id	1, NBS. 2, WBS.
Note	N/A	

8 A2DP Sink AT Command Definition

This chapter introduces A2DP (Sink Role) relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

8.1A2DP Status

```
typedef enum
{
    a2dp_success,                   /*! The operation succeeded. */
    a2dp_invalid_parameters,        /*! Invalid parameters supplied by the client. */
    a2dp_sdp_fail,                  /*! SDP registration has failed. */
    a2dp_l2cap_fail,                /*! L2CAP registration has failed. */
     a2dp_operation_fail,           /*! The operation has failed. */
     a2dp_insufficient_memory,      /*! No memory to perform the required task. */
     a2dp_wrong_state,              /*! The library is in the wrong state to perform the operation. */
     a2dp_no_signalling_connection, /*! No signaling connection. */
     a2dp_no_media_connection,      /*! No media connection. */
     a2dp_rejected_by_remote_device,/*! Was rejected by the remote device. */
     a2dp_disconnect_link_loss,     /*! Link loss occurred. */
     a2dp_closed_by_remote_device,  /*! Closed by remote device. */
     a2dp_aborted                   /*! Connection was aborted. */
} a2dp_status_code;
```

8.2A2DPCONN

The A2DPCONN command is used to establish a A2DP connection with a remote device.

Command	**AT+B A2DPCONN [bd]**	
Response	**AT-B A2DPCONN [status],[bd]**	
Parameters	bd	Remote *Bluetooth* device address.
	status	Values in the chapter 8.1.
Note	bd is comprised of 12 bytes hexadecimal characters.	

8.3A2DPDISC

The A2DPDISC command is used to release the A2DP connection with a remote device.

Command	**AT+B A2DPDISC**	
Response	**AT-B A2DPDISC [status],[bd]**	
Parameters	status	Values in the chapter 8.1
	bd	Remote *Bluetooth* device address
Note	N/A	

8.4A2DPSVGS

The A2DPSVGS command is used to change the speaker gain.

Command	**AT+B A2DPSVGS [gain]**	
Response	Succeeded: **AT-B A2DPSVGS 0** Failed: **AT-B A2DPSVGS 1**	
Parameters	gain	Speaker gain, where ranges from 0 to 15.
Note	N/A	

8.5A2DPGVGS

The A2DPGVGS command is used to get the speaker gain.

Command	**AT+B A2DPGVGS**	
Response	**AT-B A2DPGVGS [gain]**	
Parameters	gain	Speaker gain, where ranges from 0 to 15.
Note	N/A	

9 A2DP Sink Indication Definition

This chapter introduces the A2DP (Sink Role) relevant indications' definition.

9.1A2DPSTAT

The A2DPSTAT indication is used to inform the host unit when the A2DP sink's is changed.

Indication	**AT-B A2DPSTAT [state]**	
Parameters	state	A2DP connection status, where 1: a2dpReady; 2: a2dpConnecting; 3: a2dpConnected; 4: a2dpStreaming.
Note	N/A	

9.2A2DPCONN

The A2DPCONN indication is used to inform the host unit when it initializes a AVRCP connection with the local device.

Indication	**AT-B A2DPCONN [status], [bd]**	
Parameters	status	Values in the chapter 8.1.
	bd	Remote *Bluetooth* device address.
Note	bd is comprised of 12 bytes hexadecimal characters.	

9.3A2DPAUDIO

The A2DPAUDIO indication is used to inform the host unit that the A2DP audio connection is on or off.

Indication	**AT-B A2DPAUDIO [op]**	
Parameters	op	0: the A2DP audio connection is off; 1: the A2DP audio connection is on.
Note	N/A	

9.4A2DPCODEC

The A2DPCODEC indication is used to inform the host codec negotiated with the remote device.

Indication	**AT-B A2DPCODEC [codec_id],[channel],[rate]**	
Parameters	codec_id	1, SBC 2, MP3 3, AAC 5, APTX 6, APTX_LL
	channel	Channel mode for the audio being streamed; 0,mono 1,dual channel 2,stereo 3, joint stereo
	rate	Sample rate for internal codec, 44.1k or 48k Hz;
Note	N/A	

10 AVRCP Controller AT command Definition

This chapter introduces AVRCP (Controller Role) relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

10.1 AVRCP Status

```
typedef enum
{
    avrcp_success = (0),              /*! Operation was successful. */
    avrcp_fail,                       /*! Operation failed. */
    avrcp_no_resource,                /*! Not enough resources. */
    avrcp_bad_state,                  /*! Request is not supported in the current state. */
    avrcp_timeout,                    /*! Operation timed out before completion. */
    avrcp_device_not_connected,       /*! Device specified is not connected. */
    avrcp_busy,                       /*! Operation is already in progress. */
    avrcp_unsupported,                /*! Requested operation is not supported. */
    avrcp_invalid_sink,               /*! Sink supplied was invalid. */
    avrcp_link_loss,                  /*! Link loss occurred. */
    avrcp_rejected=0x0A,              /*! The operation was rejected. */
    avrcp_interim_success=0x0F,       /*! Operation was successful, but have only received an interim
                                      response.*/

/* Below status codes depends on the error status code received from the remote device. Retain the same
values while inserting new values or modifying this enum */
    avrcp_rejected_invalid_pdu = 0x80,    /*! The operation was rejected with reason - invalid PDU. */
    avrcp_rejected_invalid_param,         /*! The operation was rejected with reason - invalid parameter. */
    avrcp_rejected_invalid_content,       /*! The operation was rejected with reason - invalid content. */
    avrcp_rejected_internal_error,        /*! The operation was rejected with reason - internal error. */
    avrcp_rejected_uid_changed = 0x85,    /*! The operation was rejected with reason - UID Changed. */
    avrcp_rejected_invalid_direction = 0x87,  /*! The command has been rejected with reason -Invalid
                                          Direction.*/
    avrcp_rejected_not_directory,         /*! The command has been rejected with reason -Not a
                                          Directory.*/
    avrcp_rejected_uid_not_exist,         /*! The command has been rejected with reason -Does not
                                          exist.*/
    avrcp_rejected_invalid_scope,         /*! The command has been rejected with reason -Invalid Scope.*/
    avrcp_rejected_out_of_bound,          /*! The command has been rejected with reason - Range Out of
```

```
                                        Bounds.*/
    avrcp_rejected_uid_directory,       /*! The command has been rejected with reason - UID is a
                                        Directory.*/
    avrcp_rejected_media_in_use,        /*! The command has been rejected with reason - Media in
                                        Use.*/
    avrcp_rejected_play_list_full,      /*! The command has been rejected with reason - Now Playing
                                        List Full.*/
    avrcp_rejected_search_not_supported, /*! The command has been rejected with reason - Search Not
                                        Supported.*/
    avrcp_rejected_search_in_progress,  /*! The command has been rejected with reason - Search in
                                        Progress.*/
    avrcp_rejected_invalid_player_id,   /*! This command has been rejected with reason - Invalid Player
                                        ID.*/
    avrcp_rejected_player_not_browsable, /*! This command has been rejected with reason - Player Not
                                        Browsable.*/
    avrcp_rejected_player_not_addressed, /*! This command has been rejected with reason - Player Not
                                        Addressed.*/
    avrcp_rejected_no_valid_search_results, /*! This command has been rejected with reason - No valid
                                        Search Results.*/
    avrcp_rejected_no_available_players, /*! This command has been rejected with reason - No available
                                        players.*/
    avrcp_rejected_addressed_player_changed,    /*! This command has been rejected with reason -
                                                Addressed Player Changed.*/
    avrcp_status_guard_reserverd = 0xFF     /* Dummy Place Holder */
} avrcp_status_code;
```

10.2 AVRCPPLAY

The AVRCPPLAY command is used to send “Play” command to start playing.

Command	**AT+B AVRCPPLAY**	
Response	**AT-B AVRCPPLAY [status]**	
Parameters	status	Values in the chapter 10.1.
Note	N/A	

10.3 AVRCPPAUSE

The AVRCPPAUSE command is used to send “Pause” command to pause playing track.

Command	**AT+B AVRCPPAUSE**	
Response	**AT-B AVRCPPAUSE [status]**	
Parameters	status	Values in the chapter 10.1.
Note	N/A	

10.4 AVRCPSTOP

The AVRCPSTOP command is used to send “Stop” command to stop playing.

Command	**AT+B AVRCPSTOP**	
Response	**AT-B AVRCPSTOP [status]**	
Parameters	status	Values in the chapter 10.1.
Note	N/A	

10.5 AVRCPFORWARD

The AVRCPFORWARD command is used to send “Forward” command to play the next track.

Command	**AT+B AVRCPFORWARD**	
Response	**AT-B AVRCPFORWARD [status]**	
Parameters	status	Values in the chapter 10.1.
Note	N/A	

10.6 AVRCPBACKWARD

The AVRCPBACKWARD command is used to send “Backward” command to play the previous track.

Command	**AT+B AVRCPBACKWARD**
Response	**AT-B AVRCPBACKWARD [status]**

Parameters	status	Values in the chapter 10.1.
Note	N/A	

10.7 AVRCPVOLUMEUP

The AVRCPVOLUMEUP command is used to send the Category 2 Pass through command of volume-up.

Command	**AT+B AVRCPVOLUMEUP**	
Response	**AT-B AVRCPVOLUMEUP [status]**	
Parameters	status	Values in the chapter 10.1
Note	This command is only used for Category 2 device.	

10.8 AVRCPVOLUMEDOWN

The AVRCPBACKWARD command is used to end the Category 2 Pass through command of volume-down.

Command	**AT+B AVRCPVOLUMEDOWN**	
Response	**AT-B AVRCPVOLUMEDOWN [status]**	
Parameters	status	Values in the chapter 10.1
Note	This command is only used for Category 2 device.	

10.9 AVRCPSABSVOL

The AVRCPBACKWARD command is used by the CT (Category 2) to set the absolute volume at category 2 TG.

Command	**AT+B AVRCPSABSVOL [volume]**	
Response	**AT-B AVRCPSABSVOL [status]**	
Parameters	volume	Absolute volume, where ranges from 0 to 0x7F

	status	Values in the chapter 10.1
Note	This command is only used for Category 2 device.	

10.10 AVRCPFF

The AVRCPFF command is used by the CT (Category 2) to fast forward.

Command	**AT+B AVRCPFF [op]**	
Response	**AT-B AVRCPFF [status]**	
Parameters	op	1: start; 0: stop.
	status	Values in the chapter 10.1
Note	This command is only used for Category 2 device.	

10.11 AVRCPFB

The AVRCPFB command is used by the CT (Category 2) to fast backward.

Command	**AT+B AVRCPFB [op]**	
Response	**AT-B AVRCPFB [status]**	
Parameters	op	1: start; 0: stop.
	status	Values in the chapter 10.1
Note	This command is only used for Category 2 device.	

11 AVRCP Controller Indication Definition

This chapter introduces the AVRCP(Controller Role) relevant indications' definition.

11.1 AVRCPSTAT

The AVRCPSTAT indication is used to inform the host unit when the AVRCP Controller's is changed.

Indication	**AT-B AVRCPSTAT [state]**	
Parameters	status	AVRCP connection status, where, 1: avrcpReady; 2: avrcpConnecting; 3: avrcpConnected.
Note	N/A	

11.2 AVRCPCONN

The AVRCPCONN indication happens when local or remote device creates the A2DP connection.

Indication	**AT-B AVRCPCONN [status],[bd]**	
Parameters	status	Values in the chapter 10.1.
	bd	Remote *Bluetooth* device address.
Note	AVRCP connection will be established after A2DP connection has been created	

11.3 AVRCPDISC

The AVRCPDISC indication happens when the local or remote device disconnects the AVRCP connection.

Indication	**AT-B AVRCPDISC [status], [bd]**	
Parameters	status	Values in the chapter 10.1.
	bd	Remote *Bluetooth* device address.
Note	AVRCP connection will be disconnected after A2DP connection has been	

	disconnected

11.4 AVRCPTITLE

The AVRCPTITLE indication is used to tell host the title of current playing media.

Indication	**AT-B AVRCPTITLE [title]**	
Parameters	title	Title of media, the maximum length is 128 bytes.
Note	N/A	

11.5 AVRCPARTIST

The AVRCPARTIST indication is used to tell host the artist of current playing media.

Indication	**AT-B AVRCPARTIST [artist]**	
Parameters	artist	Artist of media, the maximum length is 128 bytes.
Note	N/A	

11.6 AVRCPALBUM

The AVRCPALBUM indication is used to inform the album of current playing media.

Indication	**AT-B AVRCPALBUM [album]**	
Parameters	album	Album. UTF-8 code.
Note	N/A	

11.7 PLAYSTATUS

The PLAYSTATUS indication is used to tell host the playback status has changed.

Indication	**AT-B PLAYSTATUS [status]**

Parameters	**status**	0x00: play_status_stopped; 0x01: play_status_playing; 0x02: play_status_paused; 0x03: play_status_fwd_seek; 0x04: play_status_rev_seek; 0xFF: play_status_error.
Note	N/A	

11.8 AVRCPFEATURE

The AVRCPFEATURE indication is used to tell host the features that the remote TG supports.

Indication	**AT-B AVRCPFEATURE [metadata],[feature]**	
Parameters	metadata	0: disable, AVRCP V1.0; 1: enable, AVRCP V1.3 or later.
	feature	The features supported by the remote TG, where, Bit 0: Category 1; Bit 1: Category 2; Bit 2: Category 3; Bit 3: Category 4; Bit 4: Player Application Settings. Bit 0 should be set for this bit to be set; Bit 5: Group Navigation. Bit 0 should be set for this bit to be set; Bit 6: Supports browsing; Bit 7: Supports multiple media player applications; Bit 8-15: Reserved for Future Additions; The bits for supported categories are set to 1. Others are set to 0.
Note	N/A	

11.9 AVRCPTIME

The AVRCPTIME indication is used to inform the playing time of current playing media.

Indication	**AT-B AVRCPTIME [time]**	
Parameters	time	Time. ASCII code. Unit: ms
Note	N/A	

11.10 AVRCPPOS

The AVRCPPOS indication is used to inform the playback progress of current playing media.

Indication	**AT-B AVRCPPOS [position]**	
Parameters	position	Playback progress
Note	N/A	

12 PBAP Client AT Command Definition

This chapter introduces PBAP (Phone Book Client Equipment Role) relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

12.1 PBAP Status

```
typedef enum
{
    pbapc_success,                          /*! Last operation was successful. */
    pbapc_failure,                          /*! Last operation failed. */
    pbapc_aborted,                          /*! Last operation was aborted. */
    pbapc_not_idle,   /*! Client is not idle, so cannot perform the current operation. */
    pbapc_wrong_state,     /*! Operation failed due to being in the wrong state.*/
    pbapc_sdp_failure_resource, /*! Unable to register the SDP record due to a lack of resources */
    pbapc_sdp_failure_bluestack,        /*! Unable to register the SDP record due to Bluestack */
    pbapc_remote_disconnect,    /*! Remote host has disconnected or the link has been lost. */
    pbapc_spb_unauthorised = 0x10,          /*! Not authorised to access this phonebook */
```

```
    pbapc_spb_no_repository,                /*! The server does not contain this repository */
    pbapc_spb_not_found,                    /*! Phonebook does not exist */
    pbapc_vcl_no_param_resources = 0x20,    /*! No resources to generate application specific parameters
                                            header for PullvCardList. */
    pbapc_vcl_no_pbook_folder,              /*! A phonebook folder was specified for PullvCardList where
                                            there are no sub-folders (i.e. in pb). */
    pbapc_vcl_invalid_pbook,                /*! A phonebook folder was specified for PullvCardList
                                            which is invalid */
    pbapc_vce_no_param_resources = 0x30,    /*! No resources to generate application specific
                                            parameters header for PullvCardEntry. */
    pbapc_vce_no_name_resources,            /*! No resources to generate the vCard entry
                                            name for PullvCardEntry. */
    pbapc_vce_invalid_entry,                /*! Invalid entry for this phonebook for
                                            PullvCardEntry. Only folder 'pb' can contain an
                                            entry 0. */
    pbapc_ppb_no_param_resources = 0x40,    /*! No resources to generate application specific
                                            parameters header for PullPhonebook. */
    pbapc_ppb_no_name_resources,            /*! No resources to generate the phonebook
                                            name for PullPhonebook. */
    pbapc_ppb_no_required_name,             /*! No name for PullPhonebook when it is
                                            required. e.g. server is not in a phonebook
                                            directory */
    pbapc_ppb_no_repository,                /*! The server does not contain this repository */
    pbapc_prop_sdp_error,                   /*! Request to get the server properties failed due to an
                                            SDP error */
    pbapc_end_of_status_list
} phone_book_status;
```

12.2 PBCCONN

The PBCCONN command is used to establish a PBAP connection with a remote device.

Command	**AT+B PBCCONN [bd]**	
Response	**AT-B PBCCONN [status],[bd]**	
Parameters	bd	Remote *Bluetooth* device address.
	status	Values in the chapter 12.1.
Note	HFP connection must have already been established before establishing PBAP	

	connection.

12.3 PBCDISC

The PBCDISC command is used to release the PBAP connection with a remote device.

Command	**AT+B PBCDISC [bd]**	
Response	**AT-B PBCDISC [status], [bd]**	
Parameters	bd	Remote *Bluetooth* device address.
	status	Values in the chapter 12.1.
Note	N/A	

12.4 PBCPULLPB

The PBCPULLPB command is used to start pulling the phonebook object from the remote device.

Command	**AT+B PBCPULLPB [repository],[folder],[maxList],[startOffset]**	
Response	Succeeded: **AT-B PBCPULLDATAIND [pbSize], [moreData], [length],[packet]0xFF** Failed: **AT-B PBCPULLPB 1**	
Parameters	repository	1: local; 2: SIM card.
	folder	1: pb, main phone book; 2: ich,incoming calls; 3: och,outgoing calls; 4: mch,missed calls; 5: cch,combination of ich, och and mch.
	maxList	Maximum number of entries that PCE can handle.
	startOffset	Offset of first entry to pull

	pbSize	Number of entries interested
	moreData	More data to receive or not. More(TRUE) or not(FALSE)
	length	Length of the packet.
	packet	Data of packet.
Note	1. when maxList = 0, it can be used to get the maximum index that are actually used. In this case, all other parameters are ignored; 2. when maxList = 65535, it can be used to download all entries without knowing pbsize; 3. End of packet is 0xFF, not \r (0x0d).	
Example	**at+b pbcpullpb 1,1,0,0\CRAT-B PBCSTAT 4\CRAT-B PBCPULLDATAIND 40,0,0,\CRAT-B PBCSTAT 3\CR** **at+b pbcpullcmt\CRAT-B PBCPULLCMTIND** **at+b pbcpullpb 2,1,20,0\CRAT-B PBCSTAT 4\CRAT-B PBCPULLDATAIND 0,1,884……\CR** **at+b pbcpullcont\CRAT-B PBCSTAT 4\CRAT-B PBCPULLDATAIND 0,1,884……\CR** **……** **at+b pbcpullcont\CRAT-B PBCSTAT 4\CRAT-B PBCPULLDATAIND 0,0,420……\CR** **at+b pbcpullcmt\CRAT-B PBCPULLCMTIND\CR** **at+b pbcpullpb 2,1,20,20\CRAT-B PBCSTAT 4\CRAT-B PBCPULLDATAIND 0,1,884……\CR** **……** **at+b pbcpullcont\CRAT-B PBCSTAT 4\CRAT-B PBCPULLDATAIND 0,0,420……\CR** **at+b pbcpullcmt\CRAT-B PBCPULLCMTIND\CR**	
Example	If set the module to parse vCard format phonebook, the contact's name will be parsed with UTF-8 format. **AT+B PBCPULLPB 1,1,65535,0\CR** **AT-B PBCSTAT 4\CR** **AT-B PBCPARSEDATAIND**	

	0,\xE9\xA9\xAC\xE5\xA7\x90,0,18610865026\CR AT-B PBCPARSEDATAIND 0,\xE7\x88\xB9,0,13784070664\CR ……(more records) AT-B PBCSTAT 3\CRAT-B PBCPULLCMTIND\CR

12.5 PBCPULLCONT

The PBCPULLCONT command is used to get more dada for the ongoing pull operation.

Command	**AT+B PBCPULLCONT**	
Response	Succeeded: **AT-B PBCPULLDATAIND [pbSize], [moreData], [length],[packet]0xFF** Failed: **AT-B PBCPULLCONT 1**	
Parameters	pbSize	Ignored
	moreData	More data to receive or not. More(TRUE) or not(FALSE)
	length	Length of the packet
	packet	Data of packet
Note	This command shall be used only if the [moreData] field of the “PBCPULLDATAIND” indication equals to one.	

12.6 PBCPULLCRT

The PBCPULLCRT command is used to get the current processed packet.

Command	**AT+B PBCPULLCRT**	
Response	Succeeded: **AT-B PBCPULLDATAIND [pbSize], [moreData], [length],[packet]0xFF** Failed: **AT-B PBCPULLCRT 1**	
Parameters	pbSize	Ignored
	moreData	More data to receive or not. More(TRUE) or not(FALSE)

	length	Length of the packet
	packet	Data of packet
Note	1. If the [length] field of the PBCPULLDATAIND indication is not equal to the real packet received by MCU, it is possible that packet loss has occurred on UART. In this situation, this command can be used to retransmit the packet. 2. End of packet is 0xFF, not \r (0x0d).	

12.7 PBCPULLCMT

The PBCPULLCMT command is used to get the current processed packet.

Command	**AT+B PBCPULLCMT**	
Response	Succeeded: **AT-B PBCPULLCMTIND** Failed: **AT-B PBCPULLCMT 1**	
Parameters	N/A	
Note	N/A	

12.8 PBCSETPARSE

The PBCSETPARSE command is used to set if parsing phonebook vCard data.

Command	**AT+B PBCSETPARSE [para]**	
Response	Succeeded: **AT-B PBCSETPARSE 0** Failed: **AT-B PBCSETPARSE 1**	
Parameters	**para**	1: Parse vCard data 0: Don’t parse vCard data. Default setting.
Note	N/A	

12.9 PBCGETPARSE

The PBCGETPARSE command is used to inquiry if parsing phonebook vCard data.

Command	AT+B PBCGETPARSE	
Response	AT-B PBCGETPARSE [para]	
Parameters	para	1: Parse vCard data 0: Don't parse vCard data
Note	N/A	

13 PBAP Client Indication Definition

This chapter introduces the PBAP(Phone Book Client Equipment Role) relevant indications' definition.

13.1 PBCSTAT

The PBCSTAT indication is used to inform the host unit that the PBAP client's status is changed.

Indication	AT-B PBCSTAT [state]	
Parameters	state	Phonebook connection status, where, 1: pbapcReady; 2: pbapcConnecting; 3: pbapcConnected; 4: pbapcDownloading; 5: pbapcDisconnecting.
Note	N/A	

13.2 PBPULLDATAIND

The PBPULLDATAIND indication is used to inform the host unit that packet pulled arrives.

Indication	AT-B PBPULLDATAIND [pbSize], [moreData], [length],[packet]0xFF	
Parameters	pbSize	Number of entries interested

	moreData	More data to receive or not. More(TRUE) or not(FALSE)
	length	Length of the packet
	packet	Data of packet
Note	End of packet is 0xFF, not \r (0x0d).	

13.3 PBCPULLCMTIND

The PBCPULLCMTIND indicates that the current pull operation has completed.

Indication	**AT-B PBCPULLCMTIND**
Parameters	N/A
Note	The client host shall use the “PBCPULLCMT” command to complete the current PULL operation every time when the [moreData] field of the PBCPULLDATATIND indication equals to zero.

13.4 PBCPARSEDATAIND

The PBCPARSEDATAIND returns the parsed phonebook vCard data.

Indication	**AT-B PBCPARSEDATAIND [first],[type],[number],[time],[name]**	
Parameters	first	When syncing phonebook, if first=1, it represents the first phone number of a contacts.
	type	For Phonebook 0: Other Number, 1:Cell Number, 2:Home Number, 3:Work Number, 4:Preference Number 5:Fax Number For Call History 16:Received call 17:Dialed call 18:Missed call
	number	Contacts phone number

	time	When syncing call log, it will return call time
	name	Contacts name
Note	N/A	

14 MAP Client AT Command Definition

This chapter introduces MAP (Message Client Equipment Role) relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

14.1 MAP Status

This chapter introduces the defined MAP status.

14.1.1MAP Status

```
typedef enum mapc_status
{
    mapc_success,              /*!< The last operation was successful. */
    mapc_failure,              /*!< General failure */
    mapc_pending,              /*!< The operation is pending or in progress */
    mapc_connect_rejected,     /*!< The Connection has been rejected locally */
    mapc_aborted,              /*!< The operation has been aborted locally */
    mapc_invalid_state,        /*!< Command not acceptable in this state */
    mapc_mns_started,          /*!< The MNS Service was already started. */
    mapc_object_not_found,     /*!< Remote file or folder not found */
    mapc_object_protected,     /*!< Access denied to the remote object */
    mapc_command_rejected,     /*!< Remote rejected the command */
    mapc_invalid_parameter     /*!< Remote send corrupt or invalid response */
} MapcStatus;
```

14.1.2MAP Client Message Filter

```
typedef enum
{
    mapc_no_filtering = 0x00,          /*!< No filtering */
    mapc_filter_out_sms_gsm = 0x01,    /*!< Filter out GSM SMS */
    mapc_filter_out_sms_cdma= 0x02,    /*!< Filter out GSM CDMA */
```

```
    mapc_filter_out_email = 0x04,        /*!< Filter out EMAIL */
    mapc_filter_out_mms = 0x08,          /*!< Filter out    MMS */
    mapc_filter_unread = 0x10,           /*!< Get only the unread messages */
    mapc_filter_read = 0x20,             /*!< Get only the read messages */
    mapc_filter_params = 0x100           /*!< Use an auto filter for Param Mask*/
} MapcMessageFilter;
```

14.2 MAPCCONN

The MAPCCONN command is used to create an MAP connection with the remote MAP server.

Command	**AT+B MAPCCONN [bd]**	
Response	**AT-B MAPCCONN [status], [bd]**	
Parameters	bd	Remote *Bluetooth* device address
	status	Values in the chapter 14.1.1.
Note	N/A	

14.3 MAPCDISC

The MAPCDISC command is used to disconnect the existing MAP connection.

Command	**AT+B MAPCDISC**	
Response	**AT-B MAPCDISC [status],[bd]**	
Parameters	status	Values in the chapter 14.1.1
	bd	Remote *Bluetooth* device address
Note	If MAP client is in UPLOAD/DOWNLOAD operation, the MAPCCMT command shall be used to terminate the current operation before using MAPCDISC to disconnect the existing MAP connection.	

14.4 MAPCGETML

This command is used by the MAP Client to get message listing objects from the MAP Server.

Command	**AT+B MAPCGETML [folder],[maxList],[startOffset]**	
Response	Succeeded: **AT-B MAPCGETDATAIND [listSize],[moreData],[length],[packet]** Failed: **AT-B MAPCGETML 1**	
Parameters	folder	0: inbox; 1: outbox; 3: sent; 4: deleted; 5: draft.
	maxList	The maximum number of messages to be listed (0-65535).
	startOffset	Offset of the first entry to be listed.
	listSize	Size of available messages.
	moreData	More data to be received or sent. More(TRUE) or not(FALSE)
	length	The length of the packet field.
	packet	The partial or complete packet of an object, cannot be NULL, cannot include '\r'.
	folder	0: inbox; 1: outbox; 3: sent; 4: deleted; 5: draft.
Note	If MAP client is in UPLOAD/DOWNLOAD operation, the MAPCCMT command shall be used to terminate the current operation before using the MAPCDISC command to disconnect the existing MAP connection.	

14.5 MAPCGETCONT

The MAPCGETCONT command is used to get more dada for the ongoing get operation.

Command	**AT+B MAPCGETCONT**

Response	Succeeded: **AT-B MAPCGETDATAIND [listSize], [moreData], [length],[packet]** Failed: **AT-B PBCPULLCONT 1**	
Parameters	listSize	Size of available messages.
	moreData	More data to be received or sent. More(TRUE) or not(FALSE)
	length	The length of the packet field
	packet	The partial or complete packet of an object, cannot be NULL, cannot include '\r'.
Note	This command shall be used only if the [moreData] field of the MAPCGETDATAIND indication equals to one.	

14.6 MAPCGETMSG

This command is used by the MAP Client to get the packet of the message object with the [handle] field from the MAP Server.

Command	**AT+B MAPCGETMSG [handle]**	
Response	Succeeded: **AT-B MAPCGETDATAIND [moreData],[packetSize],[packet]** Failed: **AT-B MAPCGETMSG 1**	
Parameters	handle	You should get this parameter by parsing the Message Listing object
	moreData	More data to be received or sent. More(TRUE) or not(FALSE)
	length	The length of the packet field
	packet	The partial or complete packet of an object, cannot be NULL, cannot include '\r'.
Note	N/A	

14.7 MAPCGETCRT

The MAPCGETCRT command is used by the MAP client to get the previous indication when the MAP client has received the MAPCGETDATAIND indication.

Command	**AT+B MAPCGETCRT**	
Response	Succeeded: **AT-B MAPCGETCRTIND [moredata],[packetSize],[packet]** Failed: **AT-B MAPCGETCRT 1**	
Parameters	moreData	More data to be received or sent. More(TRUE) or not(FALSE)
	length	The length of the packet field
	packet	The partial or complete packet of an object, cannot be NULL, cannot include '\r'.
Note	1. If the [length] field of the MAPCGETDATAIND indication is not equal to the real packet received by MCU, it is possible that packet loss has occurred on UART. In this situation, this command can be used to retransmit the packet. 2. This command can be used to get the current packet before receiving the MAPCGETCMTIND indication, only for getting message listing function.	

14.8 MAPCPUSHMSG

The MAPCPUSHMSG command is used by the client host to put a message to remote MAP server.

Command	**AT+B MAPCPUTMSG [moreData],[packetSize],[packet]**	
Response	If this is the only packet to be sent, the response will be: **AT-B MAPCPUTCMTIND [status]** If there are more packets to be sent, the response will be: **AT-B MAPCPUTMSGIND** Failed: **AT-B MAPCPUTMSG 1**	
Parameters	moreData	More data to be received or sent. More(TRUE) or not(FALSE)
	length	The length of the packet field
	packet	The partial or complete packet of an object, cannot be NULL, cannot include '\r'.
Note	Maximum length of packet is 128 bytes.	

14.9 MAPCCMT

The MAPCCMT command is used to terminate the ongoing get/push operation.

Command	**AT+B MAPCCMT**
Response	If in get operation, the response is: **AT-B MAPCGETCMTIND** If in push operation, the response is: **AT-B MAPCPUSHCMTIND** Failed: **AT-B MAPCCMT 1**
Parameters	N/A
Note	Before receiving MAPCGETCMTIND or MAPCPUSHCMTIND indications, this command can be used to terminate the current operation.

15 MAP Client Indication Definition

This chapter introduces the MAP(Message Client Equipment Role) relevant indications' definition.

15.1 MAPCINIT

The MAPCINIT indication is used to inform the client host the result of the MAPC(Message Notification Service) initialization.

Indication	**AT-B MAPCINIT [status]**	
Parameters	status	Values in the chapter 14.1.1.
Note	N/A	

15.2 MAPCDISC

The MAPCDISC indication is used to inform the client host of the result of MAPC connection has been disconnected.

Indication	**AT-B MAPCDISC [status],[bd]**	
Parameters	status	Values in the chapter 14.1.1
	bd	Remote *Bluetooth* device address

Note	N/A

15.3 MAPCGETDATAIND

The MAPCGETDATAIND indication is used to indicate the client host that the packet of a get operation has arrived.

Indication	**AT-B MAPCGETDATAIND [listSize],[moreData],[length],[packet]**	
Parameters	listSize	Number of entries interested
	moreData	More data to be received or sent. More(TRUE) or not(FALSE)
	packetSize	The length of the packet field
	packet	The partial or complete packet of an object, cannot be NULL, cannot include '\r'.
Note	N/A	

15.4 MAPCGETCMTIND

The MAPCGETCMTIND indication is used to inform the client host that the current pull process is over.

Indication	**AT-B MAPCGETCMTIND**
Parameters	N/A
Note	The client host shall use the MAPCGETCMT command to complete the current UPLOAD operation every time when the [moreData] field of the MAPCGETDATATIND indication equals to zero.

15.5 MAPCPUSHCONTIND

The MAPCPUSHCONTIND indication is used to inform the client host to continue the push operation.

Indication	**AT-B MAPCPUSHCONTIND**
Parameters	N/A
Note	N/A

15.6 MAPCPUTCMTIND

The MAPCPUSHCMTIND indication is used to inform the client host that the push operation is completed.

Indication	**AT-B MAPCPUSHCMTIND**
Parameters	N/A
Note	N/A

15.7 MAPCEVTIND

The MAPCEVTIND indication is used to inform the client host that there are new SMS alerts.

Indication	**AT-B MAPCEVTIND [moreData],[length],[packet]**	
	moreData	More data to be received or sent. 1： More(TRUE) o: 0： not(FALSE)
	Length	The length of the packet field
	packet	Received short message notification data.
Note	N/A	

16 SPP AT Command Definition

This chapter introduces SPP relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

16.1 SPP Status

This chapter introduces the defined SPP status.

16.1.1SPP Connect Status

typedef enum

{

```
    spp_connect_success,        /*! Connect attempt succeeded. */
    spp_connect_sdp_fail,       /*! Service search failed. */
    spp_connect_slc_failed,     /*! Service level connection establishment failed. */
    spp_connect_failed_busy,    /*! Profile instance already connected. */
    spp_connect_failed,         /*! RFCOMM connection failed to be established. */
    spp_connect_server_channel_not_registered,    /*! Requested server channel not registered by this
                                                    profile instance. */
    spp_connect_timeout,      /*! Connection attempt timed out. */
    spp_connect_rejected,     /*! The remote device rejected the connection. */
    spp_connect_normal_disconnect, /*! The remote device terminated the connection. */
    spp_connect_abnormal_disconnect,              /*! Unsuccessful due to an abnormal disconnect while
                                                    establishing the RFCOMM connection. */
    spp_connect_rfcomm_channel_already_open,      /*! The connection attempt failed because there is
                                                    already a connection to that remote device on the
                                                    requested RFCOMM channel. */
    spp_connect_invalid_frame_size                /*! Connect failed due to invalid frame size request
                                                    from app. */
} spp_connect_status
```

16.1.2SPP Disconnect Status

```
typedef enum
{
    spp_disconnect_success,         /*! Successful disconnection. */
    spp_disconnect_link_loss,       /*! Disconnection ocurred due to link loss. */
    spp_disconnect_no_slc,          /*! Disconnect attempt failed, no service level connection. */
    spp_disconnect_timeout,         /*! Disconnect time out. */
    spp_disconnect_error,           /*! Unsuccessful for some other reason. */
} spp_disconnect_status;
```

16.2 SPPCONN

The SPPCONN command is used to establish a SPP connection with a remote device.

Command	**AT+B SPPCONN [bd]**	
Response	**AT-B SPPCONN [status], [bd]**	
Parameters	bd	Remote *Bluetooth* device address.
	status	Values in the chapter 16.1.1.

Note	N/A

16.3 SPPDISC

The SPPDISC command is used to release the SPP connection with the remote device.

Command	**AT+B SPPDISC**	
Response	**AT-B SPPDISC [status],[bd]**	
Parameters	status	Values in the chapter 16.1.2.
	bd	Remote *Bluetooth* device address.
Note	N/A	

16.4 SPPDATA

The SPPDATA command is used to transfer data with the remote device.

Command	AT>Opcode+Length+ Parameter +\r
Response	Succeeded: **AT-B SPPDATA 0** Failed: **AT-B SPPDATA 1**
AT>	命令前缀，三字节 ASCII 字符。
Opcode	命令操作码，两个字节。每个命令以 opcode 作为唯一标识。 0x0101 BLE 类型数据 0x0102 SPP 类型数据
length	参数长度，两个字节。所有参数的长度，以字节为单位（并不是参数个数）。
Parameter	参数:包含 Hdl 和 data Hdl 连接句柄，一般设置为：80 00 Data 要传输的数据；
\r	命令后缀，一字节 ASCII 字符。
例如	字符类型 AT>\x02\x01\x00\x06\x80\x00\r234\r

	hex 类型 41 54 41 54 3e 02 01 00 07 80 00 0d 0b 0c 0e 00 0d 01 02 03 04 0d

17 SPP Indication Definition

This chapter introduces the SPP relevant indications' definition.

17.1 SPPSTAT

The SPPSTAT indication is used to inform the host unit when the local device's SPP status is changed.

Indication	**AT-B SPPSTAT [state]**	
Parameters	state	SPP connection status, where 1: sppReady; 2: sppConnecting; 3: sppConnected.
Note	N/A	

17.2 SPPDATAIND

The SPPDATAIND indication is used to inform the host unit that SPP data is received from the remote device.

Indication	**AT-B SPPDATAIND [length],[data]**	
Parameters	length	Length of received data
	data	Received data
Note	N/A	

18 GATT General AT Command Definition

This chapter introduces the general GATT (Both Peripheral and Central Role) AT

commands' definition, including a brief description of commands' syntax, responses and examples.

18.1 BLEPSKEY

The BLEPSKEY command is used to pair after establishing the GATT connection if remote device require input pin code.

Command	**AT+B BLEPSKEY [PASSKEY]**	
Response	Start to execute: **AT-B BLEPSKEY 0** Haven't executed: **AT-B BLEPSKEY 1** Pair result: **AT-B BLEPAIR [sys_status]**	
Parameters	PASSKEY	The pin code of remote device required
	sys_status	0: pair succeeded; Other values: pair failed.
Note	N/A	

18.2 BLESTATE

The BLESTAT indication is used to inform the host unit when the local device's GATT status is changed.

Indication	**AT-B BLESTAT [state]**	
Parameters	state	GATT connection status, where 0: gattReady; 1: gattAdvertising; 3: gattConnected; 5: gattDisconnecting; 6: Idle.
Note	N/A	

18.3 BLEDISC

This command is used to disconnect connected device .

Command	**AT+B BLEDISC [CID]\r**	
Response	Success: AT-B BLEDISC 0,[CID]\r Failed: AT-B BLEDISC 1,[CID]\r	
Parameters	[CID]	The channel ID of the connection.
Note	N/A	

18.4 ERR_CODE

This ERR_CODE is Enumeration of gatt_status_t, and is suitable for all ERR_CODE below.

```
typedef enum
{
    /*! The operation was successful. */
    gatt_status_success                                = 0x00,
    /*! The attribute handle given was not valid */
    gatt_status_invalid_handle,
    /*! The attribute cannot be read */
    gatt_status_read_not_permitted,
    /*! The attribute cannot be written */
    gatt_status_write_not_permitted,
    /*! The attribute PDU was invalid */
    gatt_status_invalid_pdu,
    /*! The attribute requires an authentication before it can be read or
        written */
    gatt_status_insufficient_authentication,
    /*! Target device doesn't support request */
    gatt_status_request_not_supported,
    /*! Offset specified was past the end of the long attribute */
    gatt_status_invalid_offset,
    /*! The attribute requires authorization before it can be read or
        written */
    gatt_status_insufficient_authorization,
    /*! Too many prepare writes have been queued */
    gatt_status_prepare_queue_full,
    /*! No attribute found within the given attribute handle range. */
    gatt_status_attr_not_found,
    /*! This attribute cannot be read or written using the Read Blob Request
        or Write Blob Requests. */
```

```
gatt_status_not_long,
/*! The Encryption Key Size used for encrypting this link is
    insufficient. */
gatt_status_insufficient_encr_key_size,
/*! The attribute value length is invalid for the operation. */
gatt_status_invalid_length,
/*! The attribute request that was requested has encountered an error
    that was very unlikely, and therefore could not be completed as
    requested. */
gatt_status_unlikely_error,
/*! The attribute requires encryption before it can be read or written */
gatt_status_insufficient_encryption,
/*! The attribute type is not a supported grouping attribute as defined
    by a higher layer specification. */
gatt_status_unsupported_group_type,
/*! Insufficient Resources to complete the request. */
gatt_status_insufficient_resources,
/*! Application error to indicate a attribute request not valid for the
    current radio type FIXME: not in spec B-96416 */
gatt_status_application_error,

/*! Connection is initialising */
gatt_status_initialising,

/*! Generic failure status. */
gatt_status_failure,
/*! Failed to register with the ATT protocol (initialisation). */
gatt_status_att_reg_failure,
/*! ATT Database registration failed (initialisation). */
gatt_status_att_db_failure,
/*! Max Number of ATT connections have already been made. */
gatt_status_max_connections,
/*! ATT disconnected abnormally (L2CAP Disconnection). */
gatt_status_abnornal_disconnection,
/*! ATT disconnected because of Link Loss. */
gatt_status_link_loss,
/*! MTU can only be exchanged once per connection. */
gatt_status_mtu_already_exchanged,
/*! Characteristic Value returned by the server did not match the
   requested one. */
gatt_status_value_mismatch,
```

```
/*! Connection was rejected because of PSM */
gatt_status_rej_psm,
/*! Connection was rejected because of security */
gatt_status_rej_security,
/*! Connection was rejected because of missing link key */
gatt_status_key_missing,
/*! Connection timed out */
gatt_status_connection_timeout,
/*! Connection retrying */
gatt_status_retrying,
/*! Peer aborted the connection */
gatt_status_peer_aborted,

/*! Error to indicate that request to DM can not be completed because
      device ACL entity is not found */
gatt_status_device_not_found = 0x7f73,
/*! Attribute signing failed. */
gatt_status_sign_failed,
/*! Operation can't be done now. */
gatt_status_busy,
/*! Current operation timed out. */
gatt_status_timeout,
/*! Invalid MTU */
gatt_status_invalid_mtu,
/*! Invalid UUID type */
gatt_status_invalid_uuid,
/*! Operation was successful, and more responses will follow */
gatt_status_success_more,
/*! Indication sent, awaiting confirmation from the client */
gatt_status_success_sent,
/*! Invalid connection identifier */
gatt_status_invalid_cid,
/*! Attribute database is invalid */
gatt_status_invalid_db,
/*! Attribute server database is full */
gatt_status_db_full,
/*! Requested server instance is not valid */
gatt_status_invalid_phandle,
/*! Attribute permissions are not valid */
gatt_status_invalid_permissions

 } gatt_status_t
```

19 GATT Peripheral AT Command Definition

This chapter introduces the GATT (Peripheral Role) relevant AT commands' definition, including a brief description of commands' syntax, responses and examples.

19.1 BLEADV

The BLEADV command is used to make the module advertise.

Command	**AT+B BLEADV [op]**	
Response	Succeeded: **AT-B BLEADV 0** Failed: **AT-B BLEADV 1**	
Parameters	op	1: start advertising; 0: stop advertising.
Note	i1107e can be searched and connected by a central device when it is in the advertising status.	

19.2 BLEDATA

The BLEDATA command is used to send data over GATT.

Command	AT>Opcode+Length+ Parameter +\r
Response	Succeeded: **AT-B BLEDATA 0** Failed: **AT-B BLEDATA 1**
AT>	命令前缀，三字节 ASCII 字符。
Opcode	命令操作码，两个字节。每个命令以 opcode 作为唯一标识。 0x0101 BLE 类型数据 0x0102 SPP 类型数据
length	参数长度，两个字节。所有参数的长度，以字节为单位（并不是参数个数）。
Parameter	参数:包含 Hdl 和 data

	Hdl 连接句柄，一般设置为：80 00 Data 要传输的数据；
\r	命令后缀，一字节 ASCII 字符。
例如	AT>\x01\x01\x00\x06\x80\x00\r234\r 字符类型 41 54 41 54 3e 01 01 00 07 80 00 0d 0b 0c 0e 00 0d 01 02 03 04 0d hex 类型

20 GATT Peripheral Indication Definition

This chapter introduces the GATT (Peripheral Role) relevant indications' definition.

20.1 BLEDATAIND

The BLEDATAIND indication is used to inform the host unit that GATT data is received from the remote device.

Indication	**AT-B BLEDATAIND [length],[data]**	
Parameters	length	Length of received data
	data	Received data
Note	N/A	

21 GATT Central AT Command Definition

21.1 BLESCAN

This command is used to start or stop scanning devices.

Command	**AT+B BLESCAN [1/0]\r**	
Response	AT-B BLEADVIND [addr_type],[addr],[ad type]:[ad data]...\CR	
Parameters	[1/0]	1: start scanning: AT+B BLESCAN 0\r 0: stop scanning: AT+B BLESCAN 1\r

	addr_type	0:PUBLIC ADDR 1:RANDOM ADDR 0xFF: INVALID
	addr	The address of the advertising device
	ad type	2:Incomplete list of 16-bit Service Class UUIDs 3:Complete list of 16-bit Service Class UUIDs 5: Complete list of 32-bit Service Class UUIDs 7: Complete list of 128-bit Service Class UUIDs 8:Shortened local device name 9:Complete local device name.
Note	NOTE: it will only return the first 5 devices scanned, and only the UUID and device type advertising data; Other advertising type data and other advertising devices will not return.	

21.2 BLECONN

The command is used to connect the advertising device.

Command	**AT+B BLECONN [addr_type],[addr]\r**	
Response	AT-B BLECONN 0,[cid]\r AT-B BLECONN 1,[cid] \r	
Parameters	Addr_type	The address type of the device want to connect. 0:PUBLIC ADDR; 1:RANDOM ADDR; 0xFF: INVALID
	addr	The address of the device want to connect
	[cid]	The channel ID of this connection.
Note	N/A	

21.3 BLEDAPS

This command is used to discover all primary service,

Command	**AT+B BLEDAPS [cid]\r**	
Response	AT-B BLEDAPS [cid],[start_handle],[end_handle],[uuid_type], [uuid],[more]\r	
Parameters	cid	The channel ID of the connection
	start_handle	The start handle of the service;
	end_handle	The end handle of the service;
	uuid_type	0: UUID not present; 1: 16-bit UUID; 2: 128-bit UUID;
	more	1: more service; 0: no more;
Note	N/A	

21.4 BLEDACR

The command is used to discover all Characteristics of a service

Command	**AT+B BLEDACR [cid],[start_handle],[end_handle]\r**	
Response	AT-B BLEDACR [cid],[handle],[properties],[uuid_type],[uuid],[more]\r	
Parameters	handle	The handle of Characteristic
	properties	The properties[hex] of Characteristic: 0x01: Broadcast； 0x02: Read； 0x04: Write without Reponses； 0x08: Write； 0x10: Notify； 0x20: Indicate； 0x40: Authenticated Signed Writes： 0x80: Extended Properties

	uuid_type	0: UUID not present; 1: 16-bit UUID; 2: 128-bit UUID;
	more	1: more Characteristic; 0 : no more;
Note	N/A	

21.5 BLEDACD

The command is used to discover all Characteristic Descriptors of a service

Command	**AT+B BLEDACD [cid], [start_handle],[end_handle]\r**	
Response	AT-B BLEDACD [cid],[handle],[uuid_type], [uuid],[more]\r	
Parameters	handle	The handle of Characteristic
	uuid_type	0: UUID not present; 1: 16-bit UUID; 2: 128-bit UUID;
	more	1: more Characteristic; 0: no more;
Note	N/A	

21.6 BLERCVR

The command is used to read a Characteristic value by read characteristic value.

Command	**AT+B BLERCVR [cid],[handle] \r**	
Response	Success: AT-B BLERCVR 0\r Failed: AT-B BLERCVR 0\r (this mean no this cid or not support BLE Central) AT-B BLERCVR [CID],[HANDLE],[ERR_CODE]\r (this mean the remote reject the read command)	
Parameters	cid	The channel ID of connection

	handle	The handle of characteristic
	ERR_CODE	Gatt Error Code
Note	N/A	

21.7 BLEWCVR

The command is used to write a Characteristic value by write characteristic value. There are five sub-proceduresthat can be used to write a Characteristic Value:

Write Without Response, Signed Write Without Response, Write Characteristic Value, Write Long Characteristic Values and Reliable Writes.

Command	**AT+B BLEWCVR [cid],[handle],[size_value],[value]\r**	
Response	Success: AT-B BLEWCVR 0\r Failed: AT-B BLEWCVR [CID],[HANDLE],[ERR_CODE]\r	
Parameters	cid	The channel ID of connection
	handle	The handle of characteristic
	Size_value	The size of value write to the characteristic
	value	The value write to the characteristic
	ERR_CODE	Gatt Error Code
Note	N/A	

21.8 BLEWWRR

The command is used to write a Characteristic value by Write Without Response

Command	**AT+B BLEWWRR [cid],[handle],[size_value],[value]\r**	
Response	Success: AT-B BLEWCVR 0\r Failed: AT-B BLEWCVR [CID],[HANDLE],[ERR STATUS]\r	
Parameters	cid	The channel ID of connection

	handle	The handle of characteristic
	Size_value	The size of value write to the characteristic
	value	The value write to the characteristic
Note	N/A	

22 GATT Central Indication Definition

This chapter introduces the GATT (Central Role) relevant indications' definition.

22.1 BLESTATE

The BLESTATE indication is used to inform the host unit when the local device's GATT status is changed.

Indication	**AT-B BLESTATE [state]**	
Parameters	state	0: Initial status; 1: advertising status; 3: Connected status; 4:gattconnected_and_scanning 5: gattDisconnecting; 6: Idle. 7:scanning
Note	N/A	

22.2 PSKREQ

The PSKREQ is used to inform the host the remote need to input pin code by AT command AT+B BLEPSKEY [PASSKEY].

Indication	**AT-B PSKREQ 1\r**
Note	N/A

22.3 BLENOTIIND

The BLE notification is used to inform the host unit that GATT data is received from the peripheral.

Indication	**AT-B BLENOTIIND [cid],[handle],[length],[data]\r**	
Parameters	cid	The connected channel ID
	handle	The handle of the Characteristic that send notification
	length	The length of the notification data
	data	The notification data
Note	N/A	

22.4 BLEINDIIND

The BLE indication is used to inform the host unit that GATT data is received from the peripheral, the difference between notification and indication is indication need response,i1107e will response automatic when received indication.

Indication	**AT-B BLEINDIIND [cid],[handle],[length],[data]\r**	
Parameters	cid	The connected channel ID
	handle	The handle of the Characteristic that send indication
	length	The length of the indication data
	data	The indication data
Note	N/A	

23 PIO Assignments

23.1 DFU/Production PIO (PIO5)

When Bluetooth receives the PIO5 is pulled up, it will reboot into DFU mode. DFU mode's host interface is BCSP and this mode also supports production trim and power table tuning.

24 Bluetooth Technology Best Developed Together

IVT Wireless Limited is one of Bluetooth® technology BEST developed together which is authenticated by The Bluetooth SIG. See Figure below. IVT Wireless ecosystem is one completed Bluetooth productions including Bluetooth software, modules and end productions.

Figure 1 IVTW is One of Bluetooth Technology BEST Developed Together

25 Contact Information

25.1 Beijing

Beijing Tel: +86 10 82702580
Fax: +86 10 82898219
Address:C710, Shangdi International Pioneering Park No.2, Shang Di Xin Xi Road, Haidian District, Beijing , 100089 P.R. China
Marketing Email: marketing@ivtwireless.com
Support: support@ivtwireless.com
Web site: www.ivtwireless.com

25.2 Shenzhen

Shenzhen Tel: +86 755 27885822-603
Address: BlockA,2nd Floor, New Deal Industrial Park-B, Xin'an Street, Bao'an District 71, Shenzhen,518101
Support: support@barrot.com.cn
Web site: www.barrot.com.cn

26 Copyright