BC05 MM EXT I50E BT Module

AT Commands

This document is the AT command definition of Bluetooth component.

IVT Corporation

5/F, Fazhan Plaza,
NO. 12, Xinxi Road,
Haidian District,
Beijing, 100085
P.R. China

Tel: +86 10 82898219
Fax: +86 10 62963059
Embedded@ivtcorporation.com
www.ivtcorporation.com

Table of Contents

Introduction

The I50 BT module integrates the Bluetooth HFP, PBAP, A2DP and AVRCP functions.

There are two units of this solution, one is the Bluetooth component which acts as controller role, and the other one is a MCU which acts as host role. The Bluetooth related functions are embedded in controller side and the application is running in host side.

Because of this high-level of integration, the communication method between host unit and controller unit is a UART port. All commands and responses between the host side and controller side were defined into AT command format.

1. Important Notes

1.1 UART parameter settings

Baudrate 115200, 8 bits data, no parity, 1 stop bit.

1.2 Boot-up timing

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

1.3 Parameter Maximum Length

1.3.1 Bluetooth Software version information

An ASCII code string like “**I50E-BC05MM-EXT-Feb 16 2011**”, the maximum length is 26 bytes.

1.3.2 Bluetooth device address

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

1.3.3 Bluetooth device name

The maximum length of Bluetooth device name is 31 bytes UTF8 code.
For convenience, host can only set local name as 1-31 bytes UTF8 code, and the admitted codes are ‘A’-‘Z’,’a’-‘z’,’0’-‘9’.
If host get remote device name, which name is set as language other than English, host shall call UTF8 to ASCII converter to display the proper name. For more information, please see the PC host APP code.

1.3.4 PIN code

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

1.3.5 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.

1.3.6 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.

2. AT Command Format

2.1 Command 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’

2.2 Response 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'

2.3 Indication 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'

2.4 Note

Some responses will not be "immediate". Where applicable, these will be noted and will include an approximate delay before response.

For commands with optional parameters, all possible forms will be listed under the syntax subsection.

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

3. Generic AT Command Definition

This chapter details the generic AT commands, response, including a brief description of behavior, syntax of the command, context of the command, and types of responses. These commands are profile independent.

3.1 GVER

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

3.1.1 Syntax

AT+B GVER

3.1.2 Response

The response is:

AT-B GVER [ver]

3.2 GLBD

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

3.2.1 Syntax

AT+B GLBD

3.2.2 Response

If the command succeeded, the response is:

AT-B GLBD 0,[bd]

If the command failed, the response is:

AT-B GLBD 1, 0

3.3 GLDN

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

3.3.1 Syntax

AT+B GLDN

3.3.2 Response

If the command succeeded, the response is:

AT-B GLDN 0, [name]

If the command failed, the response is:

AT-B GLDN 1,

3.4 SLDN

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

3.4.1 Syntax

AT+B SLDN [name]

3.4.2 Response

If the command succeeded, the response is:

AT-B SLDN 0

If the command failed, the response is:

AT-B SLDN 1

3.4.3 Example

Sent 26:
0000: 41 54 2B 42 20 53 4C 44 4E 20 42 6C 75 65 74 6F AT+B SLDN Blueto
0010: 6F 74 68 43 61 72 4B 69 74 0D othCarKit.
Received 12
0000: 41 54 2D 42 20 53 4C 44 4E 20 30 0D AT-B SLDN 0.

3.5 GRDN

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

3.5.1 Syntax

AT+B GRDN [bd]

3.5.2 Response

If the command succeeded, the response is:

AT-B GRDN 0, [bd],[name]

If the command failed, the response is:

AT-B GRDN 1, [bd],

3.6 FPIN

The FPIN command is used to read or write the local fixed PIN code.

3.6.1 Syntax

AT+B FPIN [op],[pin]

op: 0-read, 1-write

pin: The pin code to be written. If none pin to set, please set pin as “0”.

3.6.2 Response

AT-B FPIN [status],[op],[pin]

If no fixed pin read or fixed pin length > 16, the response is:

AT-B FPIN 1,0,

If fixed pin read, the response is:

AT-B FPIN 0,0,[pin]

If write pin length > 0 & length <= 16, the response is:

AT-B FPIN 0,1,[pin]

If write pin length=0 or length > 16, the response is:

AT-B FPIN 1,1,

3.7 GPRD

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

3.7.1 Syntax

AT+B GPRD

3.7.2 Response

The response is:

AT-B GPRD [total],[index],[bd]

Total: total paired devices in controller unit.
Index: index of total

If no paired record found, the response is:

AT-B GPRD 0,0,000000

If paired records found(n>=1), the response is:

AT-B GPRD n,1,bd

AT-B GPRD n,2,bd

……..

AT-B GPRD n,n,bd

3.7.3 Example

Sent 10:
0000: 41 54 2B 42 20 47 50 52 44 0D AT+B GPRD.
Received 10
0000: 41 54 2D 42 20 47 50 52 44 20 AT-B GPRD
Received 4
0000: 32 2C 31 2C 2,1,
Received 1
0000: 30 0
Received 12
0000: 30 31 35 38 33 30 41 30 45 44 33 0D 015830A0ED3.
Received 10
0000: 41 54 2D 42 20 47 50 52 44 20 AT-B GPRD
Received 4
0000: 32 2C 32 2C 2,2,
Received 1
0000: 30 0
Received 3
0000: 30 31 35 015
Received 9

0000: 38 33 32 43 30 41 46 42 0D 832C0AFB.

3.8 DPRD

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

3.8.1 Syntax

AT+B DPRD [bd]

bd :

if bd=”000000000000”, delete all

if bd is not ”000000000000”, then delete the record by this BD address.

3.8.2 Response

AT-B DPRD [result],[bd]

If delete all, the response is:

AT-B DPRD 0,000000000000

If delete one by BD address success, the response is:

AT-B DPRD 0,[bd]

If delete one by BD address fails(not found in device paired list), the response is:

AT-B DPRD 1,[bd]

3.8.3 Example

Sent 23:

0000: 41 54 2B 42 20 44 50 52 44 20 30 30 31 35 38 33 AT+B DPRD 001583

0010: 30 41 30 45 44 33 0D 0A0ED3.

Received 12

0000: 41 54 2D 42 20 44 50 52 44 20 30 2C AT-B DPRD 0,

Received 13

0000: 30 30 31 35 38 33 30 41 30 45 44 33 0D 0015830A0ED3.

3.9 INQU

The INQU command will cause local device to discover other nearby

Bluetooth devices.

3.9.1 Syntax

AT+B INQU [op]

op: 1 start
0 stop

3.9.2 Response

If any nearby device was found, the response is:

AT-B INQR [bd],[class]

When the inquiry process finished, the response is:

AT-B INQC

3.9.3 Example- Inquiry no canceling

```
Sent 12:
0000: 41 54 2B 42 20 49 4E 51 55 20 31 0D          AT+B INQU 1.
Received 10
0000: 41 54 2D 42 20 49 4E 51 52 20                AT-B INQR
Received 12
0000: 30 30 31 35 38 33 30 30 32 34 30 39          001583002409
Received 1
0000: 2C                                            ,
Received 4
0000: 32 30 38 0D                                   208.
Received 10
0000: 41 54 2D 42 20 49 4E 51 52 20                AT-B INQR
Received 12
0000: 30 30 31 35 38 33 30 41 30 45 44 33          0015830A0ED3
Received 1
0000: 2C                                            ,
Received 7
0000: 31 32 30 31 30 34 0D                          120104.
Received 10
0000: 41 54 2D 42 20 49 4E 51 43 0D                AT-B INQC.
```

3.9.4 Example- Inquiry with canceling

Sent 12:

0000: 41 54 2B 42 20 49 4E 51 55 20 31 0D AT+B INQU 1.
Received 5
0000: 41 54 2D 42 20 AT-B
Received 5
0000: 49 4E 51 52 20 INQR
Received 13
0000: 30 30 31 35 38 33 30 30 32 34 30 39 2C 001583002409,
Received 4
0000: 32 30 38 0D 208.
Received 10
0000: 41 54 2D 42 20 49 4E 51 52 20 AT-B INQR
Sent 12:
0000: 41 54 2B 42 20 49 4E 51 55 20 30 0D AT+B INQU 0.
Received 13
0000: 30 30 31 35 38 33 30 41 30 45 44 33 2C 0015830A0ED3,
Received 7
0000: 31 32 30 31 30 34 0D 120104.
Received 10
0000: 41 54 2D 42 20 49 4E 51 43 0D AT-B INQC.

3.10 PAIR

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

3.10.1 Syntax

AT+B PAIR [bd]

3.10.2 Response

The response is:

AT-B PAIR [status],[bd]

status:

```
typedef enum
{
    auth_status_success,       /*!< Authentication was successful. */
    auth_status_timeout,       /*!< Authentication timed out. */
    auth_status_fail,          /*!< Authentication failed. */
    auth_status_repeat_attempts,/*!< Authentication failed due to too many repeat attempts.*/
    auth_status_pairing_not_allowed, /*!< Authentication failed as remote device is not
allowing pairing. */
    auth_status_unit_key_unsupported,    /*!< Authentication failed as unit keys are not
```

supported. */
auth_status_simple_pairing_unsupported, /*!< Authentication failed as simple pairing is not supported. */
auth_status_host_busy_pairing /*!< Authentication failed as host is already busy pairing. */
} authentication_status;

3.11 SCAN

The SCAN command is used to set scan mode.

3.11.1 Syntax

AT+B SCAN [mode]

mode : 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.

3.11.2 Response

If the command succeeded, the response is:

AT-B SCAN 0,[mode]

If the command failed:

AT-B SCAN 1,[mode]

3.11.3 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. 3.

4. Generic Indication Definition

4.1 INIT

The INIT indication is used to tell the host the BT module initializing is successfully or failed.

4.1.1 Syntax

AT-B INIT [status]

status:

0 - success

1 - failed

5.HFP AT Command Definition

5.1 HFP status

5.1.1 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;
```

5.1.2 HFP Connect Status

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

5.1.3 HFP Disconnect Status

```
typedef enum
{
    /*! Successful disconnection.*/
    hfp_disconnect_success,
    /*! Unsuccessful due to abnormal link loss.*/
```

```
    hfp_disconnect_link_loss,
    /*! Unsuccessful due to no current connection.*/
    hfp_disconnect_no_slc,
    /*! Unsuccessful due to RFCOMM connection attempt timeout.*/
    hfp_disconnect_timeout,
    /*! Unsuccessful due to RFCOMM connection attempt error.*/
    hfp_disconnect_error
} hfp_disconnect_status;
```

5.2 HFCONN

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

5.2.1 Syntax

AT+B HFCONN [bd]

bd:

Bluetooth device address of the specific device

5.2.2 Response

The response is:

AT-B HFCONN [status], [bd]

Status:

Value in hfp_connect_status. .

5.3 HFDISC

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

5.3.1 Syntax

AT+B HFDISC

bd:

Bluetooth device address of the specific device

5.3.2 Response

The command response as follows:

AT-B HFDISC [status]

status:

value in hfp_disconnect_status.

5.4 HFANSW

The command is used to answer the incoming call.

5.4.1 Syntax

AT+B HFANSW

bd:

Bluetooth device address of the HFP connected device.

5.4.2 Response

The response is:

AT-B HFANSW [status]

status:

value in hfp_lib_status.

5.5 HFCHUP

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

5.5.1 Syntax

AT+B HFCHUP

5.5.2 Response

The response is:

AT-B HFCHUP [status]

status:

value in hfp_lib_status.

5.6 HFDIAL

The command is used to dial a phone number.

5.6.1 Syntax

AT+B HFDIAL[type], [num]

Type

0, dial the supplied number;

1, dial from the the supplied memory location;

2, perform a last number redial.

Num

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.

5.6.2 Response

The response is:

AT-B HFDIAL [type],[status]

status:

value in hfp_lib_status.

5.7 HFDTMF

The command is used to transmit a DTMF code to the AG.

5.7.1 Syntax

AT+B HFDTMF [key]

Key

DTMF key, 0-9, A, B, C, D, *, #.

5.7.2 Response

The response is:

AT-B HFDTMF [status]

status:

value in hfp_lib_status.

5.8 HFCTRS

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

5.8.1 Syntax

AT+B HFCTRS

5.8.2 Response

The response is:

AT-B HFCTRS [status]

status:

value in hfp_lib_status.

5.8.3 Note

The host will receive audio connection on/off indication when the command succeeded.

5.9 HFMCAL

The HFMCAL is used to handle the mutiple calls in hsActive or hsTWCallWaiting state. This command only success when the HFP instance in hsActiveCall or hsTWCallWaiting state.

5.9.1 Syntax

AT+B HFMCAL [op]

Op:

```
typedef enum
{
    MultipleCallsReleaseHeldOrRejectWaiting,
    MultipleCallsReleaseActiveAcceptOther,
    MultipleCallsHoldActiveAcceptOther,
}MultiCallOp;
```

5.9.2 Response

The response is:

AT-B HFMCAL [status], [op],

status:

value in hfp_lib_status.

5.10 HFCLCC

The HFCLCC is used to get current calls list of AG side.

5.10.1 Syntax

AT+B HFCLCC

5.10.2 Response

If one or more current calls found, the response

AT-B HFCCIN
[call_idx],[direction],[status],[mode],[multiparty],[number_type],[number]

call_idx:
Index number of call e.g. for referencing with AT+CHLD commands.

direction:
Indicates if the call is AG originated or not.

```
hfp_call_direction          direction;
typedef enum
{
    /*! Call from AG to network.*/
    hfp_call_mobile_originated,
    /*! Call from network to AG.*/
    hfp_call_mobile_terminated
} hfp_call_direction;
```

status:
State of the call.

```
hfp_call_status             status;
typedef enum
{
    /*! Call is currently active.*/
    hfp_call_active,
```

```
        /*! Call is currently held.*/
        hfp_call_held,
        /*! Call is being dialled - mobile originated only.*/
        hfp_call_dialling,
        /*! Call is alerting - mobile originated only.*/
        hfp_call_alerting,
        /*! Call is incoming - mobile terminated only.*/
        hfp_call_incoming,
        /*! Call is waiting - mobile terminated only.*/
        hfp_call_waiting
    } hfp_call_status;
```

mode:

Indicates the mode of the call - bearer/teleservice.

```
hfp_call_mode                mode;
typedef enum
{
    /*! Voice call.*/
    hfp_call_voice,
    /*! Data call.*/
    hfp_call_data,
    /*! FAX call.*/
    hfp_call_fax
} hfp_call_mode;
```

multiparty:

Indicates if the call is a multi-party call or not.

```
hfp_call_multiparty      multiparty;
typedef enum
{
    /*! Call is not multiparty.*/
    hfp_not_multiparty_call,
    /*! Call is multiparty.*/
    hfp_multiparty_call
} hfp_call_multiparty;
```

number_type:

Type of number.

```
hfp_number_type              number_type;
typedef enum
{
    /*! Type of number is unknown.*/
    hfp_number_unknown,
    /*! Number is an international number.*/
    hfp_number_international,
    /*! Number is a national number.*/
    hfp_number_national,
```

```
        /*! Number is a network specific number.*/
        hfp_number_network,
        /*! Number is a dedicated access, short code.*/
        hfp_number_dedicated
    } hfp_number_type;
```

When the command is finished, the response

AT-B HFCLCC [status]

status:

value in hfp_lib_status.

5.11 HFSVGS

The HFSVGS is used to send speaker volume to AG side, for BT module part, the volume will not change, for it can't adjust PCM volume.

5.11.1 Syntax

AT+B HFSVGS [vol]

Vol:

the value of volume, in the range 0-15.

5.11.2 Response

The response is:

AT-B HFSVGS [vol],[status]

status:

value in hfp_lib_status.

5.11.3 Note

This command can be send if its state is equal to or greater than hsConnected.

5.12 HFSVGM

The HFSVGM is used to send microphone volume to AG side, for BT module part, the volume will not change, for it can't adjust PCM volume.

5.12.1 Syntax

AT+B HFSVGM [vol]

Vol:

the value of volume, in the range 0-15.

5.12.2 Response

The response is:

AT-B HFSVGM [vol],[status]

status:

value in hfp_lib_status.

5.12.3 Note

This command can be send if its state is equal to or greater than hsConnected.

5.13 HFMUTE

The HFMUTE 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.

5.13.1 Syntax

AT+B HFMUTE [on/off]

On/off

on-mute, off-unmute.

5.13.2 Response

The response is:

AT-B HFMUTE [on/off],[status]

status:

0- success

1- fail

6. HFP Indication Definition

6.1 HFSTAT

The HFSTAT indication is used to tell the host HFP state when changed.

6.1.1 Syntax

AT-B HFSTAT [state]

State

1, HfpTLReady,
2, HfpTLSlcConnecting,
3, HfpTLSlcConnected,
4, HfpTLIncomingCallEstablish,
5, HfpTLOutgoingCallEstablish,
6, HfpTLActiveCall
7, HfpTLTWCalling

6.2 HFCONN

The indication happens when remote device creates the HFP connection with BT module.

6.2.1 Syntax

AT-B HFCONN [status], [bd]

Status:

Value in hfp_connect_status.

6.3 HFDISC

The indication happens when the remote device disconnect the HFP connection with BT module.

6.3.1 Syntax

AT-B HFDISC [status]

status:

value in hfp_disconnect_status.

6.4 HFRING

The HFRING indication is used to tell the host HFP ring comes, host shall turn on audio path when this indication comes and turn off audio path when the state comes to HfpTLSlcConnected.

6.4.1 Syntax

AT-B HFRING

6.5 HFIBRN

The HFIBRN indication is used to tell the host HFP in-band ring feature is turned on or turned off.

6.5.1 Syntax

AT-B HFIBRN [inbandring]

Inbandring 0-off, 1-on

6.6 HFAUDIO

The HFAUDIO indication is used to tell the host HFP audio connection on or off.

6.6.1 Syntax

AT-B HFAUDIO [onoff]

onoff

1, on

0, off

6.7 HFCLIP

The HFCLIP indication is used to tell the host the incoming call's caller id.

6.7.1 Syntax

AT-B HFCLIP [callerid]

Callerid

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.

6.8 HFCCWA

The HFCCWA indication is used to tell the host the second incoming call's caller id.

6.8.1 Syntax

AT-B HFCCWA [callerid]

Callerid

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.

6.9 HFNUML

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

6.9.1 Syntax

AT-B HFNUML [number]

Number

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.

6.10 HFNUMC

The HFNUMC indication is used to tell the host AT+CNUM command is complete.

6.10.1 Syntax

AT-B HFNUMC [status]

status:

value in hfp_lib_status.

6.11 HFSGNL

The HFSGNL indication is used to tell the host the signal strength of AG side.

6.11.1 Syntax

AT-B HFSGNL [signal]

signal:
Signal Strength indicator, where:
<value>= ranges from 0 to 5

6.12 HFROAM

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

6.12.1 Syntax

AT-B HFROAM [roam]

roam: Roaming status indicator, where:
<value>=0 means roaming is not active
<value>=1 means a roaming is active

6.13 HFBATC

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

6.13.1 Syntax

AT-B HFBATC [battchg]

battchg: Battery Charge indicator of AG, where:
<value>=ranges from 0 to 5

6.14 HFVGSI

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

6.14.1 Syntax

AT-B HFVGSI [spkvol]

<spkvol>=ranges from 0 to 15

6.15 HFVGMI

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

6.15.1 Syntax

AT-B HFVGMI [micvol]

<micvol>=ranges from 0 to 15

7. PBAP Client AT Command Definition

7.1 PBAP Client 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
} pbapc_lib_status;
```

7.2 PBCCONN

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

7.2.1 Syntax

AT+B PBCCONN [bd]

bd:

Bluetooth device address of the specific device

7.2.2 Response

The response is:

AT-B PBCCONN [status],[packetSize]

Status:

Value in pbapc_lib_status.

packetSize:

Maximum size of packet transferable during this session

7.3 PBCDISC

The command is used to disconnect the PBAP connection with the remote device side.

7.3.1 Syntax

AT+B PBCDISC

7.3.2 Response

The command response as follows:

AT-B PBCDISC [status]

status:

value in pbapc_lib_status;

7.4 PBCDOWN

The command is used to start the procedure of downloading the phonebook from the remote device side.

7.4.1 Syntax

AT+B PBCDOWN [repository],[phonebook],[format]

repository:

```
typedef enum
{
    pbap_current,
    pbap_local,
    pbap_sim1,

    pbap_r_unknown
} pbap_phone_repository;
```

phonebook:

```
typedef enum
{
```

```
    pbap_telecom,
    pbap_pb,
    pbap_ich,
    pbap_och,
    pbap_mch,
    pbap_cch,

    pbap_b_unknown
} pbap_phone_book;
```

format:

```
typedef enum
{
    /*! vCard 2.1. */
    pbap_format_21 = 0x00,
    /*! vCard 3.0 */
    pbap_format_30 = 0x01,
    /*! Use default value */
    pbap_format_def
} pbap_format_values;
```

7.4.2 Response

If the command succeeds, the first packet arrives with a status indicating whether more data will come.

If the command fails, the response is:

AT-B PBCDOWN 1

8. PBAP Indication Definition

8.1 PBCSTAT

The PBCSTAT indication is used to tell the host PBAP client state when changed.

8.1.1 Syntax

AT-B PBCSTAT [state]

State

1, pbapcReady,

2, pbapcConnecting,
3, pbapcConnected,
4, pbapcDownloading
5, pbapcDisconnecting

8.2 PBCDNDATA

The indication arrives when PBAP client is downloading phonebook via **PBCDOWN** and **PBCGNEXT** command.

8.2.1 Syntax

AT-B PBCDNDATA [length], [data]

length:
The length of the data part.
data:
Phonebook data, the data may includes one or more phonebook recorders. In the BT module firmware, the data max length is set to 255. So, the host buffer size shall be more than 255 to hold the phonebook data.

Note: Host application shall parse the VCard to get the phonebook item.

8.3 PBCDNFINISH

The indication indicates PBAP client has finished downloading phonebook.

8.3.1 Syntax

AT-B PBDNFINISH [status]

status:
value in pbapc_lib_status;

9. A2DP Sink AT Command Definition

9.1 A2DP 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 signalling 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 occured. */
    a2dp_closed_by_remote_device,   /*!< Closed by remote device. */
    a2dp_aborted                    /*!< Connection was aborted. */
} a2dp_status_code;
```

9.2 A2DPCONN

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

9.2.1 Syntax

AT+B A2DPCONN [bd]

bd:

Bluetooth device address of the specific device

9.2.2 Response

The response is:

AT-B A2DPCONN [status]

Status:

Value in a2dp_status_code.

9.3 A2DPDISC

The command is used to disconnect the A2DP connection with the remote device side.

9.3.1 Syntax

AT+B A2DPDISC

9.3.2 Response

The command response as follows:

AT-B A2DPDISC [status]

status:

value in a2dp_status_code

10. A2DP Sink Indication Definition

10.1 A2DPSTAT

The A2DPSTAT indication is used to tell the host A2DP Sink state when changed.

10.1.1 Syntax

AT-B A2DPSTAT [state]

State

1, a2dpReady,
2, a2dpConnecting,
3, a2dpConnected,
4, a2dpStreaming

10.2 A2DPCONN

The indication happens when remote device creates the A2DP connection with BT module.

10.2.1 Syntax

AT-B A2DPCONN [status], [bd]

Status:

Value in a2dp_status_code

11. AVRCP Controller AT command Definition

11.1 PASSTHROUGH

The command is used to send Pass Through control command to the target on the connection identified by the specified sink.

11.1.1 Syntax

AT+B PASSTHROUGH [op]

op:

```
typedef enum
{
    opidPlay,
    opidPause,
    opidStop,
    opidForward,
    opidBackward
}avrcpPassThroughCmd;
```

11.1.2 Response

If the AVRCP connection exists, the response is:

AT-B PASSTHROUGH 0

Otherwise, the response is:

AT-B PASSTHROUGH 1

12. AVRCP Controller Indication Definition

12.1 AVRCPSTAT

The AVRCPSTAT indication is used to tell the host AVRCP Controller state when changed.

12.1.1 Syntax

AT-B AVRCPSTAT [state]

State

1, avrcpReady,
2, avrcpConnecting,
3, avrcpConnected,