FSC-BT90X

BT4.2 Programming User Guide

Version 2.2

Revision History

Version	Data	Notes	Author
1.0	2016/01/20	First Release	Navy
2.2	2017/11/21	Add A2DP Source, HFP Source profiles	Tony

Contact Us:

Shenzhen Feasycom Technology Co., Ltd
Web: www.feasycom.com
Email:support@feasycom.com
Tel: +86-755-27924639,+86-755-23062695
Address: Room 2004-2005,20th Floor, Huichao Technology Building,
Jinhai Road, Xixiang, Baoan District, Shenzhen,518100, China.

Contents

1. Introduction

This specification presents design guidelines for software engineers that use FSC-BT90X series modules for Bluetooth requirements

1.1 Terms

Throughout this specification:

- ✧ Content between { } is optional
- ✧ Content behind << represents a *COMMAND* sent from Host to Module
- ✧ Content behind >> represents a *RESPONSE* sent from Module to Host

1.2 Hardware Interface

- ✧ GPIO
- ✧ PWM
- ✧ UART
- ✧ SPI Master
- ✧ I2C Master/Slave
- ✧ I2S Master/Slave
- ✧ Analog Input/Output (only available on FSC-BT906)

1.3 Supported Bluetooth Profile

- ✧ SPP (Serial Port Profile)
- ✧ GATT Server (Generic Attribute Profile)
- ✧ GATT Client (Generic Attribute Profile)
- ✧ HFP Sink (Hands-Free Profile)
- ✧ HFP Source (Hands-Free Profile)
- ✧ A2DP Sink (Advanced Audio Distribution Profile)
- ✧ A2DP Source (Advanced Audio Distribution Profile)
- ✧ AVRCP Controller (Audio/Video remote controller Profile)
- ✧ AVRCP Target (Audio/Video remote controller Profile)
- ✧ HID Keyboard (Human Interface Profile)
- ✧ PBAP Server (Phonebook Access Profile)

1.4 Command Format

AT+ Command {=Param1{, Param2{, Param3...}}} <CR><LF>

- All commands start with “AT”, end with <CR><LF>
- <CR> stands for "carriage return", corresponding hex is 0x0D
- <LF> stands for "line feed", corresponding hex is 0x0A
- If command has parameter, parameter keep behind “=”
- If command has multiple parameters, parameter must be separated by “,”
- If command has response, response start with <CR><LF>, end with <CR><LF>
- Module will always report command’s execution result using “OK” for success or “ERR<code>” for failure

Error Code	Meaning
001	Failed
002	Invalid Parameter
003	Invalid State
004	Command Mismatch
005	Busying
006	Not Supported
007	No Memory
Others	Reserved for Future Use

e.g.

1. Read module’s BR/EDR local name

```
<<    AT+NAME
>>    +NAME=Feasycom
>>    OK
```

2. Pick up an incoming call when no call incoming actually

```
<<    AT+HFPANSW
>>    ERR003
```

1.5 Indication Format

<CR><LF>+ Indication {=Param1{, Param2{, Param3...}}} <CR><LF>

- All indications start with <CR><LF>, end with <CR><LF>
- If indication has parameter, parameter keep behind “=”
- If indication has multiple parameters, parameter must be separated by “,”

e.g.

1. Received "1234567890" from mobile phone via SPP profile

   >> *+SPPDATA=10,1234567890*

2. Call number "10086" use mobile phone when HFP connected

   >> *+HFPSTAT=4*

   >> *+HFPCID=10086*

   >> *+HFPCIE=China Mobile*

   >> *+HFPAUDIO=1*

   >> *+HFPSTAT=6*

1.6 Module Default Settings

Local Name (BR/EDR)	FSC-BT90X
Local Name (LE)	FSC-BT90X-LE
Pin Code	0000
Secure Simple Pairing Mode	On
Physical UART Baudrate	115200bps/8/N/1
Profiles Selection	1195

2. Command Table

2.1 General Commands

2.1.1 UART Communication Test

Format: AT

Response: OK

Description: Test the UART communication between HOST and Module after power on, baudrate changed, etc.

Example: UART communication test

```
<<      AT
>>      OK
```

2.1.2 Bluetooth Profile Selection *<need reboot>*

Format: AT+PROFILE{=Param}

Param: A base-10 representation of a bit field, default:1195, for each bit:

BIT[0]	SPP (Serial Port Profile)
BIT[1]	GATT Server (Generic Attribute Profile)
BIT[2]	GATT Client (Generic Attribute Profile)
BIT[3]	HFP Sink (Hands-Free Profile)
BIT[4]	HFP Source (Hands-Free Profile)
BIT[5]	A2DP Sink (Advanced Audio Distribution Profile)
BIT[6]	A2DP Source (Advanced Audio Distribution Profile)
BIT[7]	AVRCP Controller (Audio/Video remote controller Profile)
BIT[8]	AVRCP Target (Audio/Video remote controller Profile)
BIT[9]	HID Keyboard (Human Interface Profile)
BIT[10]	PBAP Server (Phonebook Access Profile)

Response: +PROFILE=Param

Description: GATT Server and Client, HFP Sink and Source, A2DP Sink and Source, AVRCP Controller and Target cannot be enabled both because of mutual exclusion.

Module will soft reboot if profile selection changed

Example: Read current profile selection

```
<<      AT+PROFILE
>>      +PROFILE=1195
```

Example: Enable SPP, GATT Server, HFP Source, A2DP Source profile, disable the others

```
<<      AT+PROFILE=83
>>      OK
```

2.1.3 Read Firmware Version

Format: AT+VER

Response: +VER=Param

Param: Firmware version (24 Bytes ASCII)

Example: Read module's firmware version

```
<<      AT+VER
>>      +VER=FSC-BT906,V1.0.0,20160120
>>      OK
```

2.1.4 Read BR/EDR MAC Address

Format: AT+ADDR

Response: +ADDR=Param

Param: Module's BR/EDR MAC address (12 Bytes ASCII)

Example: Read Module's BR/EDR MAC address

```
<<      AT+ADDR
>>      +ADDR=DC0D30123456
>>      OK
```

2.1.5 Read BLE MAC Address

Format: AT+LEADDR

Response: +LEADDR=Param

Param: Module's LE MAC address (12 Bytes ASCII)

2.1.6 Read/Write BR/EDR Local Name

Format: AT+NAME {=Param1{, Param2}}

Param1: BR/EDR local name (1~31 Bytes ASCII, default: FSC-BT90X)

Param2: MAC address suffix (0/1, default:0)

(0) Disable suffix

(1) Enable suffix "-XXXX" (lower 4 bytes of MAC address) after local name

Response: +NAME=Param

Description: Write local name if parameter existence, otherwise read current local name

Example: Read current BR/EDR local name

```
<<     AT+NAME
>>     +NAME=Feasycom
>>     OK
```

Example: Change module's BR/EDR local name to "ABC"

```
<<     AT+NAME=ABC
>>     OK
```

Example: Change module's BR/EDR local name to "ABC" and enable suffix

```
<<     AT+NAME=ABC,1
>>     OK
```

2.1.7 Read/Write BLE Local Name

Format: AT+LENAME {=Param1{, Param2}}

Param1: BLE local name (1~25 Bytes ASCII, default: FSC-BT90X-LE)

Param2: MAC address suffix (0/1, default:0)

(0) Disable suffix

(1) Enable suffix "-XXXX" (lower 4 bytes of MAC address) after local name

Response: +LENAME=Param

2.1.8 Read/Write BLE Advertising Data

Format: AT+ADVDATA {=Param}
Param: BLE advertising data (1~29 Bytes UTF8, default: Feasycom iBeacon)

Response: + ADVDATA =Param

Description: Module's advertising data is Feasycom iBeacon by default (major: 10065, minor: 26049), it can be changed to user specific data

Example: Change BLE advertising data to "hello"
<< *AT+ADVDATA= hello*
>> *OK*

2.1.9 Read/Write Pin Code

Format: AT+PIN{=Param}
Param: Pin code (4~15 Bytes ASCII, default:0000)

Response: +PIN=Param

Example: Read module's pin code
<< *AT+PIN*
>> *+PIN=0000*
>> *OK*
Example: Change module's pin code to "12345678"
<< *AT+PIN=12345678*
>> *OK*

2.1.10 Turn On/Off Secure Simple Pairing *<need reboot>*

Format: AT+SSP{=Param}
Param: Simple pairing (0/1, default:1)
(0) Turn off
(1) Turn on

Response: +SSP=Param

Description: Pin code input will be bypassed if simple pairing is on in pairing procedure

2.1.11 Read/Write UART Baudrate

Format: AT+BAUD{=Param}
Param: Baudrate (2400/4800/9600/19200/38400/57600/115200/230400/
256000/460800/512000/921600, default:115200)

Response: +BAUD=Param

Description: Module's baudrate will be changed immediately after received this command

2.1.12 Read/Write Class Of Device *<need reboot>*

Format: AT+COD{=Param}
Param: Class of device (6 bytes ASCII, default:240404 Handsfree device)

Response: +COD=Param

2.1.13 Read/Clear Paired Record

Format: AT+PLIST{=Param}
Param:(0/(1~8)/12 Bytes MAC address)
(0) Clear all paired record
(1~8) Clear specific paired record with index
(MAC) Clear specific paired record with MAC address

Response1: +PLIST= {
Response2: +PLIST=Param1, Param2{, Param3}
Param1: (1~8) Paired device's index
Param2: (MAC) Paired device's MAC address
Param3: (UTF8) Paired device's local name
Response3: +PLIST=}

Example: Read module's paired record
<< *AT+PLIST*

```
>>      +PLIST= {
        +PLIST=1,1C5CF226D773, iPhone
        +PLIST=2, A0BC30075421, Samsung Note 7
        +PLIST=}
>>      OK
```

Example: Clear module's paired record

```
<<      AT+PLIST=0
>>      OK
```

2.1.14 Turn On/Off Throughput Mode *<need reboot>*

Format: AT+TPMODE{=Param}

Param: Throughput mode (0/1, default:0)

(0) Turn Off

(1) Turn On

Response: +TPMODE=Param

Description: When SPP/GATT profile connected and throughput mode is on, the AT command will be de-active, every byte received via physical UART will be sent to air, vice visa

Example: Read current throughput mode

```
<<      AT+TPMODE
>>      +TPMODE=1
>>      OK
```

Example: Turn off throughput mode

```
<<      AT+TPMODE=0
>>      OK
```

2.1.15 Turn On/Off Pairing/Advertising Mode

Format: AT+PAIR=Param

Param: Pair mode (0/1)

(0) Turn Off

(1) Turn On

Response: OK

Description: Module will enter pair mode itself if no connection established, and leave pair mode otherwise

2.1.16 Turn On/Off BT Radio

Format: AT+BTEN{=Param}
Param: BT Radio (0/1, default:1)
(0) Turn Off
(1) Turn On

Response: +BTEN=Param

Description: Module will disable all Bluetooth function if BT Radio off

2.1.17 Turn On/Off Power On Auto Reconnect *<need reboot>*

Format: AT+AUTOCONN{=Param}
Param: Option (0/1, default:1)
(0) Turn Off
(1) Turn On

Response: +AUTOCONN=Param

Description: Module will attempt to connect last device after power on if set

2.1.18 Scan Nearby Devices

Format: AT+SCAN=Param1{, Param2{, Param3}}
Param1:(0~3)
(0) Stop scan
(1) Scan nearby BR/EDR devices
(2) Scan nearby BLE devices
(3) Scan nearby BR/EDR/BLE devices
Param2:(1~48) Scan period. unit:1.28s, default:12.8s
Param3:(1~25 Bytes ASCII) Name filter. Filter scan results with name if set

Description: Refer to Chapter 3 for format description of scan result

2.1.19 Scan and Connect Nearby Device *<a2dp/hfp source only>*

Format: AT+LINK=Param
Param: Profile selection of module, refer to 2.1.2 for description

Description: Module will scan nearby devices for 5.12 seconds, find the one which has best signal strength (RSSI > -70 at least), then connect to it automatically

Example: Module is a2dp source, find a a2dp sink device (Bluetooth speaker, e.g.) and establish a a2dp connection

```
<<      AT+LINK=64
>>      +DEVSTAT=13
        +DEVSTAT=5
        +LINKDEV=1C521650FFEF, -29
        +A2DPSTAT=2
        +DEVSTAT=7
        +A2DPSTAT=3
        +A2DPDEV=1C521650FFEF
```

2.1.20 Read Module States

Format: AT+STAT

Response: +STAT=Param1, Param2, Param3, Param4, Param5, Param6, Param7, Param8
Param1: DEVSTAT
Param2: SPPSTAT
Param3: GATTSTAT
Param4: HFPSTAT
Param5: A2DPSTAT
Param6: AVRCPSTAT
Param7: HIDSTAT
Param8: PBSTAT

Description: Refer to chapter 3 for state description, state may have different meanings according to profile selection

2.1.21 Release All Connections

Format: AT+DSCA

Description: Module release all Bluetooth connections with remote device

2.1.22 Soft Reboot

Format: AT+REBOOT

Description: Module release all Bluetooth connections with remote device then reboot

2.1.23 Restore Factory Settings

Format: AT+RESTORE

Description: Module restore all factory settings then reboot

2.1.24 I2S/PCM Format Configuration *<need reboot>*

Format: +I2SCFG{=Param}

Param: A base-10 representation of a bit field, default:0, for each bit:

BIT[0]	0: Disable I2S/PCM for audio input/output
	1: Enable I2S/PCM for audio input/output
BIT[1]	0: I2S/PCM master role
	1: I2S/PCM slave role
BIT[2]	0: 48000Hz sample rate (effective only when transmitting a2dp audio)
	1: 44100Hz sample rate (effective only when transmitting a2dp audio)
BIT[3-4]	00: I2S Philips standard format
	01: PCM standard format (short frame mono)
	10: PCM standard format (short frame stereo)
BIT[5-6]	00: 16-bit resolution
	11: 32-bit resolution

Description: I2S slave mode is not recommended as there are some issues to be fixed, use PCM slave mode instead

Example: Read current I2S/PCM configuration

```
<<    AT+I2SCFG
>>    +I2SCFG=1
```

Example: Set I2S/PCM configuration to: PCM slave, mono,16-bit resolution. When use HFP for voice audio, the configuration result in:

PCM Sync rate: 8000Hz (default)

PCM BCLK: 128Khz (8000Hz * 16bit * 1Mono)

```
<<    AT+I2SCFG=9
>>    OK
```

Example: Set I2S/PCM configuration to: I2S master, 32-bit resolution,48kHz. When use A2DP source for audio transmitting, the configuration result in:

I2S LRCLK: 48000Hz

I2S BCLK: 3.072MHz (48000Hz * 32bit * 2Stereo)

```
<<    AT+I2SCFG=97
>>    OK
```

2.1.25 Speaker Volume Setting

Format: AT+SPKVOL{=Param1{, Param2{, Param3}}}

Param1: A2dp audio volume (0~15, default:14)

Param2: HFP audio volume (0~15, default:15)

Param3: Ringtone volume (0~15, default:14)

Response: +SPKVOL =Param1, Param2, Param3

Description: Command only effective for module which has internal codec, volume 0 will mute the speaker output

Example: Read current speaker volume

```
<<    AT+SPKVOL
>>    +SPKVOL=14,15,14
```

Example: Set A2dp audio volume to 9, HFP audio volume to 15

```
<<    AT+SPKVOL=9,15
>>    OK
```

2.1.26 Microphone Gain Setting

Format: AT+MICGAIN{=Param} Param: Microphone input gain (0~15, default:8)
Response: + MICGAIN=Param
Description: Command only effective for module which has internal codec, volume 0 will mute the microphone input

2.2 HFP Commands

2.2.1 Read HFP State

Format: AT+HFPSTAT
Response: +HFPSTAT=Param Param: Refer to Chapter 3 for state description

2.2.2 Establish HFP Connection

Format: AT+HFPCONN{=Param} Param: MAC address of target device (12 Bytes ASCII)
Description: Module will reconnect to last HFP device if parameter not existence
Example: Connect to last HFP device << *AT+HFPCONN* >> *OK* **Example2**: Connect to specific HFP device with MAC address << *AT+HFPCONN=1C5CF226D773* >> *OK*

2.2.3 Release HFP Connection

Format: AT+HFPDISC

Description: Release current HFP connection with remote device

2.2.4 Dial/Redial Phone Number

Format: AT+HFPDIAL{=Param}
Param: Phone number (1~25 Bytes ASCII)

Description: Dial specific number if parameter existence, otherwise redial

Example: Redial

```
<<      AT+HFPDIAL
>>      OK
```

Example: Dial number “075527924639”

```
<<      AT+HFPDIAL=075527924639
>>      OK
```

2.2.5 Send DTMF code *<hfp sink only>*

Format: AT+HFPDTMF=Param
Param: DTMF code (0~9/#/*)

Example: Send DTMF code “#” while talking

```
<<      AT+HFPDTMF=#
>>      OK
```

2.2.6 Pick Up Incoming Call

Format: AT+HFPANSW{=Param}
Param:(1/2)

(1) Pick up incoming call and route voice audio to module
(2) Pick up incoming call and route voice audio to remote device

Description: Voice audio will route to module by default if no parameter existence

2.2.7 Simulate Incoming Call *<hfp source only>*

Format: AT+HFPRING=Param Param: Phone number (1~25 Bytes ASCII)
Description: Simulate an incoming call with number to remote device

2.2.8 Reject/Hung up Call

Format: AT+HFPCHUP
Description: Reject incoming call or hung up outgoing/active call

2.2.9 Establish/Release Voice Audio *<hfp source only>*

Format: AT+HFPAUDIO{=Param} Param: Operation (0/1) (0) Release voice audio connection with remote hfp sink device (1) Establish voice audio connection with remote hfp sink device

2.2.10 Transfer Voice Audio *<hfp sink only>*

Format: AT+HFPADTS{=Param} Param: Transfer direction (0/1) (0) Transfer voice audio from module to remote device (1) Transfer voice audio from remote device to module
Description: Transfer voice audio between module and remote device by default if no parameter existence

2.2.11 Turn On/Off Noise Reduction and Echo Cancellation

Format: AT+HFPNREC{=Param}
Param: NREC (0/1, default:1)
(0) Turn Off
(1) Turn On

Description: Echo cancellation only effective when module working as HFP sink

2.3 A2DP/AVRCP Commands

2.3.1 Read A2DP State

Format: AT+A2DPSTAT

Response: +A2DPSTAT=Param
Param: Refer to Chapter 3 for state description

2.3.2 Establish A2DP Connection

Format: AT+A2DPCONN{=Param}
Param: MAC address of target device (12 Bytes ASCII)

Description: Module will reconnect to last A2DP device if no parameter existence

Example: Connect to last A2DP device

```
<<      AT+A2DPCONN
>>      OK
```

Example2: Connect to specific A2DP device with MAC address

```
<<      AT+A2DPCONN=1C5CF226D773
>>      OK
```

2.3.3 Release A2DP Connection

Format: AT+A2DPDISC

Description: Release current A2DP connection with remote device

2.3.4 Establish/Release A2DP Audio Connection *<a2dp source only>*

Format: AT+A2DPAUDIO{=Param}
Param: Operation (0/1)
(0) Release A2DP audio connection with remote a2dp sink device
(1) Establish A2DP audio connection with remote a2dp sink device

2.3.5 Track Play/Pause

Format: AT+PLAYPAUSE

Description: Send play or pause command to remote media player according to current play status

2.3.6 Track Play

Format: AT+PLAY

Description: Send play command to remote media player

2.3.7 Track Pause

Format: AT+PAUSE

Description: Send pause command to remote media player

2.3.8 Track Stop

Format: AT+STOP
Description: Send stop command to remote media player

2.3.9 Track Forward

Format: AT+FORWARD
Description: Send forward command to remote media player

2.3.10 Track Backward

Format: AT+BACKWARD
Description: Send backward command to remote media player

2.3.11 Turn On/Off Media ID3 Information Notification

Format: AT+TRACKID3{=Param} Param: Operation (0/1, default:1) (0) Turn Off (1) Turn On
Description: Module will read track's ID3 information once tack changed if turned on, refer to Chapter 3 for the format of ID3 indication

2.3.12 Turn On/Off Media Player Play Progress Notification

Format: AT+TRACKAUDO=Param Param: Read period (0~9) (0) Stop read (1~9) Read period, unit: second

Description: Module will read remote device’s media player play progress if parameter > 0, refer to Chapter 3 for the format of play progress indication

2.4 Phonebook Access Commands

2.4.1 Read PB State

Format: AT+PBSTAT

Response: +PBSTAT=Param
Param: Refer to Chapter 3 for state description

2.4.2 Establish PB Connection

Format: AT+PBCONN

Description: Only effective when HFP connected

2.4.3 Release PB Connection

Format: AT+PBDISC

Description: Release current PB connection with remote device

2.4.4 Download Phonebook

Format: AT+PBDOWN=Param1{, Param2{, Param3}}
Param1: Phonebook type (1~5)

(1) Phonebook
(2) Received call history
(3) Dialed call history
(4) Missed call history
(5) All call history

Param2: Repository (1~3, default:1)

(1) Auto
(2) Phone memory
(3) Phone SIM1 card

Param3: Max items (1~65535, default:3000)

Response: +PBDATA=Param1<FF>Param2{<FF>Param3}
Param: Refer to Chapter 3 for description of received phonebook data

2.5 Bluetooth Serial Commands *<BR/EDR SPP>*

2.5.1 Read SPP State

Format: AT+SPPSTAT

Response: +SPPSTAT=Param
Param: Refer to Chapter 3 for state description

2.5.2 Establish SPP Connection

Format: AT+SPPCONN=Param
Param: MAC address of target device (12 Bytes ASCII)

Description: If target device is mobile phone, mobile phone must have initialized a RFCOMM service before this

2.5.3 Release SPP Connection

Format: AT+SPPDISC

Description: Release current SPP connection with remote device

2.5.4 Send Data Via SPP

Format: AT+SPPSEND=Param1, Param2 Param1: Payload length (1~492) Param2: Payload (1~492Bytes UTF8)
Description: If throughput mode is on, this command is de-active
Example: Send data “1234567890” to remote device via SPP <<   *AT+SPPSEND=10,1234567890* >>   *OK*

2.6 Bluetooth Serial Commands *<LE GATT >*

2.6.1 Read GATT State

Format: AT+GATTSTAT
Response: +GATTSTAT=Param Param: Refer to Chapter 3 for state description

2.6.2 Release GATT Connection

Format: AT+GATTDISC
Description: Release current GATT connection with remote device

2.6.3 Send Data Via GATT

Format: AT+GATTSEND=Param1, Param2 Param1: Payload length (1~492) Param2: Payload (1~492 Bytes UTF8)
Description: If throughput mode is on, this command is de-active

Example: Send data “1234567890” to remote device via GATT

```
<<      AT+GATTSEND=10,1234567890
>>      OK
```

3. Indication Table

3.1 General Indications

3.1.1 Device State

Format: +DEVSTAT=Param

Param: A base-10 representation of a bit field, for each bit:

BIT[0]	0: Power Off; 1: Power On
BIT[1]	0: BR/EDR Non Discoverable; 1: BR/EDR Discoverable
BIT[2]	0: BLE Non Advertising; 1: BLE Advertising
BIT[3]	0: BR/EDR Non Scanning; 1: BR/EDR Scanning
BIT[4]	0: BLE Non Scanning; 1: BLE Scanning

Example: Module is power on, discoverable and advertising

```
>>      +DEVSTAT=7
```

3.1.2 Scan Result

Format: +SCAN =Param1, Param2, Param3, Param4{, Param5, Param6}

Param1: Index (1~8)

Param2: Device address type (0~2)

(0)BR/EDR address
(1)LE public address
(2)LE random address

Param3: MAC address (12 Bytes ASCII)

Param4: RSSI (-127 ~ -1)

Param5: Size of Param6 if exist

Param6: Device Name for BR/EDR devices or advertising data for LE devices

Description: Param5/Param6 may not exist if remote device out of distance

Example: Scan BR/EDR nearby devices named "Feasycom" in 6.4s

```
<<      AT+SCAN=1,5, Feasycom
>>      OK
        +SCAN=1,0, DC0D30000003, -32,8, Feasycom
        +SCAN=2,0, DC0D30000044, -64,8, Feasycom
        +SCAN=3,0, DC0D30000097, -47,8, Feasycom
```

3.1.3 Link Device Information

Format: +LINKDEV=Param1, Param2
Param1: MAC address (12 Bytes ASCII), Device address with best signal strength
Param2: RSSI (-127~-1)

3.2 HFP Indications

3.2.1 HFP State

Format: +HFPSTAT=Param
Param:(0~6)

(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected
(4) Outgoing Call
(5) Incoming Call
(6) Active Call

3.2.2 HFP Device Information

Format: +HFPDEV=Param1{, Param2}
Param1: (12 Bytes ASCII), Remote device's MAC address of current HFP connection
Param2: (UTF8), Remote device's name of current HFP connection

Example: HFP connect success with device (name: "iPhone", MAC address: 1C5CF226D774)

```
>>      +HFPDEV=1C5CF226D774, iPhone
```

3.2.3 Incoming/Outgoing Call Number

Format: +HFPCID=Param
Param:(1~25 Bytes ASCII), Call number

Example: Dial number 10086

```
<<    AT+HFPDIAL=10086
>>    +HFPSTAT=4
      +HFPCID=10086
      +HFPCIE=China Mobile
      +HFPAUDIO=1
```

Example: Incoming call with number 13265463800

```
>>    +HFPSTAT=5
      +HFPCID=13265463800
      +HFPCIE=Jerry
      +HFPAUDIO=1
```

3.2.4 Incoming/Outgoing Call Name

Format: +HFPCIE=Param
Param:(UTF8), Call name

Description: Not every mobile phone support this indication

3.2.5 HFP Voice Audio State

Format: +HFPAUDIO=Param
Param:(0/1)

(0) HFP voice audio disconnected, audio input/output routed to remote device
(1) HFP voice audio connected, audio input/output routed to module

3.2.6 HFP Device Network Signal Strength

Format: +HFPSIG=Param
Param:(0~5) Network signal strength of remote device

3.2.7 HFP Device Network Operator Selection

Format: +HFPNET=Param
Param:(UTF8) Network operator selection of remote device

3.2.8 HFP Device Roaming State

Format: +HFPROAM=Param
Param:(0/1) Roaming state of remote device

3.2.9 HFP Device Battery Level

Format: +HFPBATT=Param
Param:(0~5) Battery level of remote device

3.2.10 HFP Device Manufacture

Format: +HFPMANU=Param
Param:(UTF8) Manufacture name of remote device

3.3 A2DP/AVRCP Indications

3.3.1 A2DP State

Format: +A2DPSTAT=Param
Param:(0~5)
(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected
(4) Media Streaming
(5) Media Paused

3.3.2 A2DP Device Information

Format: +A2DPDEV=Param
Param: (12 Bytes ASCII), Remote device's MAC address of current A2DP connection

3.3.3 AVRCP State

Format: +AVRCPSTAT=Param
Param:(0~3)

(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected

3.3.4 Media Player State

Format: +PLAYSTAT=Param
Param:(0~4)

(0) Stopped
(1) Playing
(2) Paused
(3) Fast Forwarding
(4) Fast Rewinding

3.3.5 Media Player Play Progress

Format: +TRACKSTAT=Param1, Param2, Param3
Param1:(0~4), Media Player State
Param2:(Decimal ASCII), Elapsed time of current track in millisecond
Param3:(Decimal ASCII), Total time of current track in millisecond

Example: Read media player play progress every 3s

```
<<    AT+TRACKAUTO=3
>>    +TRACKSTAT=1,54101,322000
      +TRACKSTAT=1,57122,322000
      +TRACKSTAT=1,60142,322000
```

3.3.6 Media ID3 Information

Format1: +ID3= {
Format2: +ID3=Param1, Param2, Param3
Param1:(1~3), ID3 type:1, title;2, artist;3, album
Param2:(Decimal ASCII), ID3 charset
Param3:(UTF8), ID3 content
Format3: +ID3=}

Example: Phone playing song “Creep-Radio Head”
>> *+ID3= {*
+ID3=1, 106, Creep
+ID3=2, 106, Radiohead
+ID3=3, 106, Pablo Honey
+ID3=}

3.4 Phonebook Access Indications

3.4.1 PB State

Format: +PBSTAT=Param
Param:(0~4)

(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected
(4) Downloading

3.4.2 PB Received Data

Format: +PBDATA=Param1<FF>Param2{<FF>Param3}
Param1: (UTF8), Name
Param2: (ASCII), Number, first byte is number type, available types are:

P: Preferred number
W: Work number
H: Home number
V: Voice number

F: Facsimile number
M: Messaging service on the number
C: Cellular number
G: Pager number
B: Bulletin board service number
D: MODEM number
R: Car-phone number
I: ISDN number
O: Video-phone number

Param3: (16 Bytes ASCII), Call time, first byte is call type, available types are:

R: Received Call
D: Dialed Call
M: Missed Call
Following 15 Bytes is call time. Format:
Year(4Bytes)Month(2Bytes)Day(2Bytes) T(1Byte)Hour(2Bytes) Minute(2Bytes)Second(2Bytes). e.g. 20161012T152826 represents 2016/10/12/15/28/26

Description: Use 0xff as parameter separator avoiding "," in payload. Param3 can only be existence if download type is call history

Example: Download all phonebook

```
<<      AT+PBCONN
>>      +PBSTAT=2
        +PBSTAT=3
<<      AT+PBDOWN=1
>>      +PBDATA=Jack<FF>V18219146201
        +PBDATA=Tom<FF>C8613423901992
        ……
        +PBDATA=END
        +PBSTAT=2
```

Example: Download 10 dialed call history at max

```
<<      AT+PBCONN
>>      +PBSTAT=2
        +PBSTAT=3
<<      AT+PBDOWN=3,1,10
>>      +PBDATA=China Mobile<FF>C10086<FF>D20171013T103516
        +PBDATA=Jerry<FF>C18688967507<FF>D20171012T152826
        ……
        +PBDATA=END
        +PBSTAT=2
```

3.4.3 PB Download Complete

Format: +PBDATA=E

Description: Indicate the download procedure completed

3.5 Bluetooth Serial Indications

3.5.1 SPP State

Format: +SPPSTAT=Param
Param:(0~3)

(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected

3.5.2 GATT State

Format: +GATTSTAT=Param
Param:(0~3)

(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected

3.5.3 SPP Device Information

Format: +SPPDEV=Param
Param: (12 Bytes ASCII), Remote device’s MAC address of current SPP connection

3.5.4 GATT Device Information

Format: +GATTDEV=Param
Param: (12 Bytes ASCII), Remote device’s MAC address of current GATT connection

3.5.5 SPP Received Data

Format: +SPPDATA=Param1, Param2
Param1: Payload length
Param2: Payload

Description: If throughput mode is on, only Param2 will be present

Example: Received data “1234567890” from remote device via SPP
<<   *+SPPDATA=10,1234567890*

3.5.6 GATT Received Data

Format: +GATTDATA=Param1, Param2
Param1: Payload length
Param2: Payload

Description: If throughput mode is on, only Param2 will be present

Example: Received data “1234567890” from remote device via GATT
<<   *+GATTDATA=10,1234567890*

4. Appendix

For additional functions such as firmware upgrade via OTA (over-the-air), examples of different application scenarios, mobile phone App support, please refer to *FSC-BT90X User Reference Manual* for more information