FSC-BT95X

BT4.2 Audio Programming User Guide

Version 2.0

Revision History

Version	Data	Notes	Author
1.0	2016/01/20	First Release	Navy
2.0	2018/03/21	Add A2DP Source	Navy

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-BT95X 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-BT95X)

1.3 Supported Bluetooth Profile

- SPP (Serial Port Profile)
- GATT Server (Generic Attribute Profile)
- GATT Client (Generic Attribute 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)

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*
   - >> *ERR004*

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*

1.6 Module Default Settings

Local Name (BR/EDR)	FSC-BT951
Pin Code	0000
Secure Simple Pairing Mode	On
Physical UART Baudrate	115200bps/8/N/1
Profiles Selection	160

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:160, 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.

BT951 only supports A2DP Sink , AVRCP Controller and A2DP Source Module will soft reboot if profile selection changed
Example: Read current profile selection << *AT+PROFILE* >> *+PROFILE=160* **Example**: Enable A2DP Source << *AT+PROFILE=64* >> *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-BT926,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/Write BR/EDR Local Name

Format: AT+NAME {=Param1{, Param2}}
Param1: BR/EDR local name (1~31 Bytes ASCII, default: FSC-BT95X)
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*

2.1.6 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*

2.1.7 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

2.1.8 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

2.1.9 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

2.1.10 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=Param1, Param2{, Param3} Param1: (1~8) Paired device's index Param2: (MAC) Paired device's MAC address Param3: (UTF8) Paired device's name **Response2**: +PLIST=E: End of the paired record
Example: Read module's paired record << *AT+PLIST* >> *+PLIST=1,1C5CF226D773, iPhone* *+PLIST=2, A0BC30075421, Samsung S8* *+PLIST=E* >> *OK* **Example**: Clear module's paired record << *AT+PLIST=0* >> *OK*

2.1.11 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

2.1.12 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

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

Format: AT+AUTOCONN{=Param} Param: Option (0~15, default:3) (0) Turn Off (1-15) Turn on and reconnect times
Response: +AUTOCONN=Param
Description: Module will attempt to connect last device after power on if set

2.1.14 Scan Nearby Devices

Format: AT+SCAN=Param1{, Param2{, Param3}} Param1:(0~1) (0) Stop scan (1) Scan nearby BR/EDR 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

2.1.15 Scan and Connect Nearby Device *<a2dp 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*

2.1.16 Turn On/Off Auto Link*<a2dp source only>*

Format: AT+AUTOLINK{=Param} Param: Option (0/1, default:0) (0) Turn Off (1) Turn on
Response: + AUTOLINK =Param
Description: if set, Module will scan nearby devices for 5.12 seconds, find the one which has best signal strength, then connect to it automatically. Module is not connected, it will always scan nearby devices

2.1.17 Release All Connections

Format: AT+DSCA
Description: Module release all Bluetooth connections with remote device

2.1.18 Soft Reboot

Format: AT+REBOOT
Description: Module release all Bluetooth connections with remote device then reboot

2.1.19 Restore Factory Settings

Format: AT+RESTORE
Description: Module restore all factory settings then reboot

2.1.20 Speaker Volume Setting

Format: AT+SPKVOL{=Param} Param: A2dp audio volume (0~15, default:14)
Response: + SPKVOL=Param
Description: Command only effective for module which has internal codec, volume 0 will mute the speaker output
Example: Read current audio volume << *AT+SPKVOL* >> *+SPKVOL=14* **Example**: Set audio volume to 9 << *AT+SPKVOL=9* >> *OK*

2.2 A2DP/AVRCP Commands

2.2.1 Read A2DP State

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

2.2.2 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*

2.2.3 Release A2DP Connection

Format: AT+A2DPDISC

Description: Release current A2DP connection with remote device

2.2.4 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

2.2.5 Track Play/Pause

Format: AT+PLAYPAUSE

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

2.2.6 Track Play

Format: AT+PLAY

Description: Send play command to remote media player

2.2.7 Track Pause

Format: AT+PAUSE
Description: Send pause command to remote media player

2.2.8 Track Stop

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

2.2.9 Track Forward

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

2.2.10 Track Backward

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

2.2.11 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

2.2.12 Turn On/Off Media Player Play Progress Notification

Format: AT+TRACKAUTO=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

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 A2DP/AVRCP Indications

3.2.1 A2DP State

Format: +A2DPSTAT=Param
Param:(0~5)

(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected
(4) Media Streaming
(5) Media Paused

3.2.2 A2DP Device Information

Format: +A2DPDEV=Param

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

3.2.3 AVRCP State

Format: +AVRCPSTAT=Param
Param:(0~3)
(0) Unsupported
(1) Standby
(2) Connecting
(3) Connected

3.2.4 Media Player State

Format: +PLAYSTAT=Param
Param:(0~4)
(0) Stopped
(1) Playing
(2) Paused
(3) Fast Forwarding
(4) Fast Rewinding

3.2.5 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

3.2.6 Media Track Information

Format: +TRACKINFO=Param1 <FF> Param2 <FF> Param3 Param1: title Param2: artist Param3: ablum
Example: Phone playing song “Creep-Radio Head” **>>** *+TRACKINFO=Creep <FF> Radiohead <FF> Pablo Honey*

3.3 GPIO Indications

3.3.1 LED Pin

LED0(Output)	
Low Level	Initializing
Blink in 1Hz	Ready to connecting
High Level	Connected