FSC-BT6XX

BT5.2 Programming User Guide

Version 3.1

Revision History

Version	Data	Notes	Author
1.0	2016/05/12	First Release	Eric
2.0	2016/10/13	Add Commands	Eric

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-BT6XX 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/Slave
- I2C Master/Slave
- Analog Input/Output

1.3 Supported Bluetooth Profile

- GATT Server (Generic Attribute Profile)
- GATT Client (Generic Attribute 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 “ERROR” for failure

e.g.

1. Read module's BR/EDR local name
   << *AT+NAME*
   >> *+NAME=Feasycom*
   >> *OK*

2. Write a baudrate which is not supported
   << *AT+BAUD=0*
   >> *ERROR*

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 GATT Server profile
   >> *+GATTDATA=10,1234567890*

1.6 Module Default Settings

Local Name	Feasycom
Service-UUID	FFF0
Write-UUID	FFF2
Notify-UUID	FFF1
Physical UART Baudrate	115200bps/8/N/1

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 Read Firmware Version

Format: AT+VER
Response: +VER=Param Param: Firmware version (15 Bytes ASCII)
Example: Read module's firmware version << *AT+VER* >> *+VER=1.0.1,FSC-BT630* >> *OK*

2.1.3 Read MAC Address

Format: AT+ADDR
Response: +ADDR=Param Param: Module's LE MAC address (12 Bytes ASCII)

2.1.4 Read/Write Local Name

Format: AT+NAME {=Param1{, Param2}}
Param1: BLE local name (1~29 Bytes ASCII, default: Feasycom)
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 local name
<< *AT+NAME*
>> *+NAME=Feasycom*
>> *OK*
Example: Change module's local name to "ABC"
<< *AT+NAME=ABC*
>> *OK*
Example: Change module's local name to "ABC" and enable suffix
<< *AT+NAME=ABC,1*
>> *OK*

2.1.5 Read/Write UART Baudrate

Format: AT+BAUD{=Param}
Param: Baudrate (1200/2400/4800/9600/19200/38400/57600/115200/
230400, default:115200)

Response: +BAUD=Param

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

2.1.6 Turn On/Off Throughput Mode

Format: AT+TPMODE{=Param}

Param: Throughput mode (0/1, default:0) (0) Turn Off (1) Turn On
Response: +TPMODE=Param
Description: When 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*

2.1.7 Turn On/Off Low Power Mode

Format: AT+LPM{=Param} Param: Low Power Mode (0/1, default: 0) (0) Turn Off (1) Turn On
Response: +LPM=Param

2.1.8 Turn On/Off Hardware Flow Control

Format: AT+FLOWCTL{=Param} Param: Hardware Flow Control (0/1, default: 0) (0) Turn Off (1) Turn On
Response: +FLOWCTL=Param

2.1.9 Read/Write Master/Slave Mode

Format: AT+ROLE{=Param} Param: Master/Slave mode (0/1, default: 0) (0) Slave Mode(GATT Server) (1) Master Mode(GATT Client)
Response: +ROLE=Param
Description: After the command is executed, the BT6XX switches to the new Mode
Example: Read current Master/Slave mode << *AT+ROLE* >> *+ROLE=0* >> *OK*

2.1.10 PIO Function Configuration

Format: AT+PIOCFG{=Param1,Param2}	
Param1	0: Disable Command/Transmission mode switch function 1: Enable Command/Transmission mode switch function
Param2	0: Disable Bluetooth disconnect function 1: Enable Bluetooth disconnect function
Response: +PIOCFG=Param1,Param2	

2.1.11 Scan Nearby Devices

Format: AT+SCAN =Param1{, Param2{, Param3}} Param1: (0~3) (0) Stop scan (1) Scan nearby 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

2.1.12 Release All Connections

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

2.1.13 Soft Reboot

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

2.1.14 Restore Factory Settings

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

2.1.15 Establish GATT Connection (GATT Client only)

Format: AT+LECCONN=Param1{,Param2,Param3,Param4} Param1: MAC address of target device & MAC address type (13 Bytes ASCII) Param2: Service-UUID, Support 16 Bit and 128 Bit (4 Bytes/32 Bytes ASCII) Param3: Write-UUID, Support 16 Bit and 128Bit (4 Bytes/32 Bytes ASCII) Param4: Notify-UUID, Support 16 Bit and 128Bit (4 Bytes/32 Bytes ASCII)
Description: If parameter 2, parameter 3, parameter 4 do not exist, the module will automatically search for the GATT service connected to the remote device
Example: Specified remote device service connections << *AT+LECCONN=123456ABCDEF0,FFF0,FFF2,FFF1* >> *OK*

2.1.16 Send Data Via GATT

Format: AT+LESEND=Param1, Param2 Param1: Payload length (1~155) Param2: Payload (1~155 Bytes UTF8)
Description: If throughput mode is on, this command is de-active
Example: Send data “1234567890” to remote device via GATT << *AT+LESEND=10,1234567890* >> *OK*

3. Indication Table

3.1 General Indications

3.1.1 Scan Result

Format: +SCAN =Param1, Param2, Param3, Param4{, Param5, Param6} Param1: Index (1~8) Param2: Device address type (0~2) (0)LE public address (1)LE random address Param3: MAC address (12 Bytes ASCII) Param4: RSSI (-255 ~ 0) Param5: Size of Param6 if exist Param6: Remote Device Name
Description: Param5/Param6 may not exist if remote device out of distance
Example: Scan nearby BLE devices << *AT+SCAN=1* >> *OK* *+SCAN=1,0, DC0D30000003, -32,8, Feasycom* *+SCAN=2,1, DC0D30000044, -64,8, Feasycom_0044* *+SCAN=3,0, DC0D30000097, -47,8, FSC_BT906*

3.1.2 GATT Received Data

Format: +GATTDATA=Param1, Param2 Param1: Payload length Param2: Payload
Example: Received data “1234567890” from remote device via GATT << *+GATTDATA=10,1234567890*