FSC-BW246

Programming User Guide

Version 1.0

Revision History

Version	Data	Notes	Author
1.0	10/20/2020	First Release	Younger

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

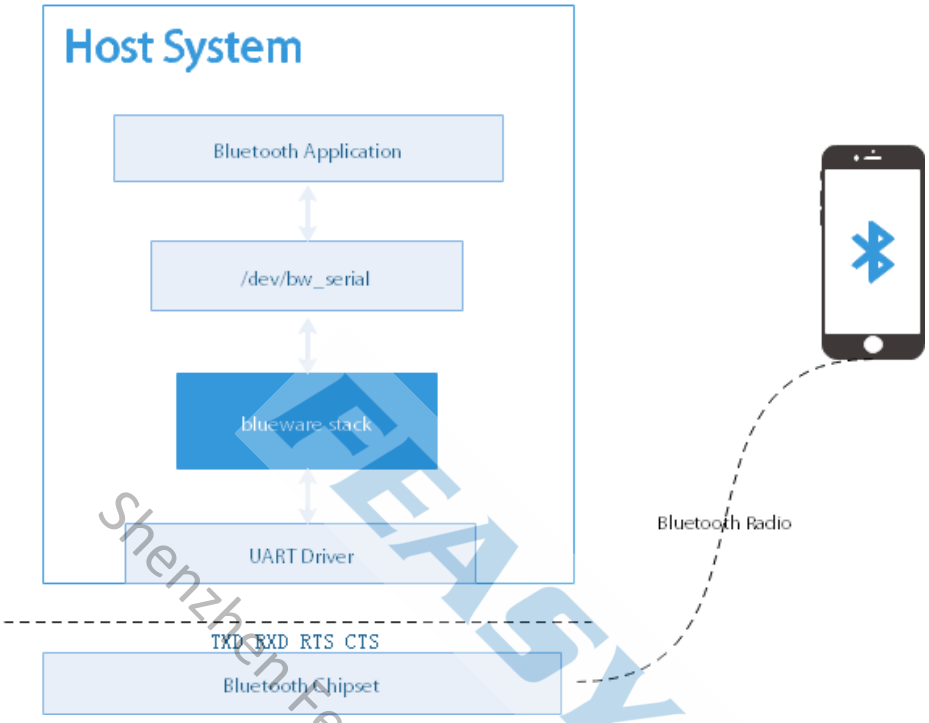
1. Introduction

This document describes the software interface of the Feasycom Embedded software solutions.

The Embedded software dedicated for the integration of Bluetooth applications in car audios or any system requiring a complete embedded Bluetooth solution. The main target of this software interface is to provide third parties a high level command set, hiding the internal complexity of the Bluetooth function and the variability of its implementations across different devices.

This software interface is based on commands that are sent from an application to the Feasycom software solution and on events that are sent from the Feasycom software to the host application.

1.1. Scope



1.2. Terms

Throughout this specification:

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

1.3. Profiles & Features

- ✧ SPP (Serial Port Profile)
  Support iAP2 (Apple Accessory Protocol) and AAP (Android Auto Protocol) as well
- ✧ GATT Server (Generic Attribute Profile)
  Used for send/receive data with Bluetooth Low Energy device as LE-Peripheral role
- ✧ GATT Client (Generic Attribute Profile)
  Used for send/receive data with Bluetooth Low Energy device as LE-Central role
- ✧ 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> means "carriage return", corresponds to hex value 0x0D
- ✧ <LF> means "line feed", corresponds to hex value 0x0A
- ✧ If Command has Parameter, Parameter follows behind '='
- ✧ If Command has multiple Parameters, Parameter must be separated by ','
- ✧ If Command has Response, Response starts with <CR><LF>, ends with <CR><LF>
- ✧ Module will always report command's execution result by using "OK" for success or "ERROR" for failure

e.g.

1. Read module's BR/EDR local name

   << AT+NAME

   >> +NAME=Feasycom

   OK

1.5. Default Settings

Local Name (BR/EDR)	FSC-XXX
Local Name (LE)	FSC-XXX-LE
Pin Code	0000
Secure Simple Pairing Mode	ON
UART Format	921600bps/8/N/1

2. Commands Table

2.1. General Commands

2.1.1. AT *Test Command*

Format: AT
Response: OK
Description: Test the communication between HOST and Module after power on, UART baudrate changed and etc.
Example: AT command test << *AT* >> *OK*

2.1.2. AT+VER *Read Firmware Version*

Format: AT+VER
Response: +VER=Param Param: Firmware version
Example: Read module's firmware version << *AT+VER* >> *+VER=FSC-BW246, V1.0.0* >> *OK*

2.1.3. AT+BTEN *Power On/Off*

Format: AT+BTEN=Param Param: Mode (0~1) (0) Power off (1) Power on

2.1.4. AT+ADDR *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. AT+LEADDR *Read BLE MAC Address*

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

2.1.6. AT+NAME *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 exist, 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. AT+LENAME *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. AT+PIN *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*
Description: Pin code only work in legacy pairing mode, see AT+SSP

2.1.9. AT+SSP *Read/Write Pairing Mode*

Format: AT+SSP{=Param} Param: Pairing mode (0~3, default:2) (0) Legacy pairing, use pin code for pairing (1) Secure simple pairing, auto pairing (2) Secure simple pairing, display yes/no in pairing (3) Secure simple pairing, passkey compare, user need to accept/reject pair request with command AT+CFM
Response: +SSP=Param

2.1.10. AT+CFM *Accept/Reject Remote Pairing Request*

Format: AT+CFM=Param1, Param2 Param1: 12 Bytes MAC address of remote device Param2: 0/1 (0) Reject remote pairing request (1) Accept remote pairing request
Description: Only used for pairing request in pairing mode 3, see AT+SSP

2.1.11. AT+PAIR *Set Visibility*

Format: AT+PAIR{=Param} Param: Mode (0~3) (0) Leave BR/EDR/BLE discoverable mode (stop advertising/broadcasting) (1) Enter BR/EDR discoverable mode (start broadcasting) (2) Enter BLE discoverable mode (start advertising) (3) Enter BR/EDR/BLE discoverable mode (start advertising/broadcasting)
Description: Module will always be discoverable if no device connected (BR/EDR or BLE), and be undiscoverable if connected with remote device, unless received this command

2.1.12. AT+PAGE *Set Connectability*

Format: AT+PAGE{=Param} Param: Mode (0~1) (0) Leave BR/EDR connectable mode (stop paging) (1) Enter BR/EDR connectable mode (start paging)
Description: Module will always be connectable mode if no device connected, and be unconnectable if connected with remote device, unless received this command

2.1.13. AT+COD *Read/Write Class of Device*

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

2.1.14. AT+PLIST *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<FF> Param2<FF> Param3{<FF> Param4} Param1: (1~8) Paired device's index Param2: A base-10 representation of a bit field, support profiles of device, see AT+PROFILE Param3: (MAC) Paired device's MAC address Param4: (UTF8) Paired device's name **Response2**: +PLIST=E: End of the paired record
Example: Read module's paired record *<< AT+PLIST* *>> +PLIST=1<FF>32808<FF>1C5CF226D773<FF> iPhone* *+PLIST=2<FF>40<FF> A0BC30075421<FF> Samsung S8* *+PLIST=E*

>> OK **Example**: Clear module's paired record *<< AT+PLIST=0* *>> OK*

2.1.15. AT+TDL *Read/Write Trust Device List Size*

Format: AT+TDL{=Param} Param:(0~8, default:0) (0) Auto mode: Latest paired record will replace the oldest one if paired list full (1~8) Limit mode: Device will not enter discoverable mode if paired list full
Response: +TDL=Param

2.1.16. AT+SCAN *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 Scan nearby BR/EDR/BLE devices (3)(5) Scan nearby AP 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: See format description in 3.1.3

2.1.17. AT+DSCA *Release All Connections*

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

2.1.18. AT+REBOOT *Soft Reboot*

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

2.1.19. AT+RESTORE *Restore Factory Settings*

Format: AT+RESTORE

2.2. SPP Commands

2.2.1. AT+SPPSTAT *Read SPP State*

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

2.2.2. AT+SPPCONN *Establish SPP Connection*

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

2.2.3. AT+SPPDISC *Release SPP Connection*

Format: AT+SPPDISC

Description: Release current SPP connection with remote device

2.2.4. AT+SPPSEND *Send Data Via SPP*

Format: AT+SPPSEND=Param1, Param2 Param1: Payload length (1~492) Param2: Payload (1~492Bytes UTF8)
Example: Send data "1234567890" to remote device via SPP << *AT+SPPSEND=10,1234567890* >> *OK*
Description: Use socket instead of AT command for data read/write on Android/Linux platform

2.3. GATT Commands

2.3.1. AT+GATTSTAT *Read GATT State*

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

2.3.2. AT+GATTDISC *Release GATT Connection*

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

2.3.3. AT+GATTSEND *Send Data Via GATT*

Format: AT+GATTSEND=Param1, Param2 Param1: Payload length (1~492) Param2: Payload (1~492Bytes UTF8)
Example: Send data "1234567890" to remote device via GATT << *AT+GATTSEND=10,1234567890* >> *OK*
Description: Use socket instead of AT command for data read/write on Android/Linux platform

2.4. WLAN Commands

2.4.1. AT+ROLE *Read/Write WLAN Role*

Format: AT+ROLE{=Param1} Param1: (1~3) (1) STA (2) AP (3) STA+AP
Example: Read WLAN mode << *AT+ROLE* >> *+ROLE=1* >> *OK*

2.4.2. AT+RAP *Connect Remote AP*

Format: AT+RAP=Param1,Param2 Param1: SSID Param2: Password
Example: Connect to a AP << *AT+RAP=Feasycom,12345678* >> *OK*

Description: Only used in STA or STA+AP mode

2.4.3. AT+LIP *Read Local IP Address*

Format: AT+LIP
Response: +LIP{=Param}
Description: When connection between module and remote AP established, a valid IP address will be distributed to module, otherwise Local IP address is always set to "0.0.0.0" which is invalid

2.4.4. AT+LAP *Read/Write Local AP Setting*

Format: AT+LAP{=Param1,Param2,{Param3}} Param1: The SSID of local AP(1-31 bytes UTF-8) Param2: The password of local AP(1-31 bytes ASCII) Param3: The IP address of local AP (dot-decimal notation)
Response: +LAP{=Param1,Param2,Param3}
Example: Write new local AP settings << *AT+LAP=FSC-BW246-AP,12345678,192.168.1.1* >> *OK*

2.4.5. AT+SOCK *Read/Write Socket Setting*

Format: AT+SOCK{=Param1,Param2,Param3,Param4} Param1: the Role of Module TCPC TCP Client TCPS TCP Server Param2: Local Port (1~65535) Param3: Remote IP Address (dot-decimal notation) Param4: Remote Port (1~65535)

Response: +SOCK{=Param1,Param2,Param3,Param4}
Example: Read Socket Setting << *AT+SOCK* >> *+SOCK=TCPC,9100,192.168.0.10,9001* >> *OK* Write New Socket Setting << *AT+SOCK=TCPC,9100,192.168.0.20,9002* >> *OK*

2.4.6. AT+WLANC *Create Connection*

Format: AT+WLANC{=Param} Param: Connection mode (1/2/3) 1 Connect to AP by BSSID 2 Connect to AP by SSID 3 Create TCP or UDP socket
Example: Connect to AP << *AT+WLANC=2* >> *OK*
Description: This command should be combined to use with RAP or/and SOCK command

2.4.7. AT+WFSEND *Send Data To Remote Device*

Format: AT+WFSEND{=Param1,Param2,Param3} Param1: ID of TCP/UDP connection (0,1,…) Param2: Payload Length Param3: Payload
Example: Send data to remote device through Wi-Fi << *AT+WFSEND=0,10,1234567890* >> *OK*

Description: Only presented in command mode

2.4.8. AT+OTA *Firmware Upgrade*

Format: AT+OTA{=Param} Param: Firmware's Name(string)
Example: Upgrade the Firmware << *AT+OTA=Feasycom_V107* >> *OK*
Description: Only used When local IP exists.

2.4.9. AT+MAC *Read Wi-Fi MAC Address*

Format: AT+MAC
Response: +MAC{=Param}
Example: Read the Wi-Fi MAC address of module << *AT+MAC* >> *+MAC=DC0D30801234* >> *OK*

2.4.10. AT+RSSI *Read RSSI of Remote AP*

Format: AT+RSSI
Response: +RSSI{=Param}
Example: Read RSSI of remote AP << *AT+RSSI* >> *+RSSI=-55* >> *OK*

Description: Only used in STA or STA+AP mode

2.4.11. AT+DHCP *Read/Write IP Distribution Mode*

Format: AT+DHCP{=Param} Param1: IP Distribution Mode (0/1,default:1) 0 Static IP 1 DHCP
Response: +DHCP{=Param}
Example: Enable Static IP << *AT+DHCP=0* >> *OK* Enable DHCP << *AT+DHCP=1* >> *OK*

2.4.12. AT+SIP *Read/Write Static IP Address*

Format: AT+SIP{=Param} Param1: The Static IP address of local STA(dot-decimal notation)
Response: +SIP{=Param}
Example: Set STA Static IP address as "192.168.1.100" << *AT+SIP=192.168.1.100* >> *OK*
Description: Only used in STA or STA+AP mode

2.4.13. AT+MASK *Read/Write Subnet Mask*

Format: AT+MASK{=Param}

Param1: The subnet mask of local STA(dot-decimal notation)
Response: +MASK{=Param}
Example: Enable STA subnet mask and set as “255.255.255.0” << *AT+MASK=255.255.255.0* >> *OK*
Description: Only used in STA or STA+AP mode

2.4.14. AT+GW *Read/Write Gateway Address*

Format: AT+GW{=Param} Param1: The gateway address of local STA(dot-decimal notation)
Response: +GW{=Param}
Example: Set STA gateway address as “192.168.1.1” << *AT+GW=192.168.1.1* >> *OK*
Description: Only used in STA or STA+AP mode

2.4.15. AT+APAC *Turn On/Off Auto-Connection to Remote AP*

Format: AT+APAC{=Param} Param1: Auto-Connection Control (0/1 default:1) 0 Disable Auto-Connection 1 Enable Auto-Connection
Response: +APAC{=Param}
Example: Turn on Auto-Connection to Remote AP << *AT+APAC=1* >> *OK*

3. Events Table

3.1. General Events

3.1.1. +DEVSTAT *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: Power on, discoverable and advertising **>>** *+DEVSTAT=7*

3.1.2. +PWRSTAT *Powering State*

Format: +PWRSTAT=Param Param: State (0) Powering off (1) Powering on(booting)
Description: AT Command is not recommended to be used while powering on/off state

3.1.3. +SCAN *Scan Result*

Format:1 +SCAN =Param1<FF>Param2<FF>Param3<FF> Param4<FF>Param5<FF>Param6 Param1: Index Param2: RSSI (-127 ~ -1) Param3: Device address type (0~3) (0)BR/EDR address (1)LE public address (2)LE random address (3)iOS device with Carplay support

Param4: MAC address (12 Bytes ASCII)
Param5: Device Name for BR/EDR devices or advertising data for LE devices
Param6: Class of device (6 Bytes ASCII)
Format2: +SCAN=E: Scan stopped

Example: Scan BR/EDR nearby devices
<< AT+SCAN=1
>> OK
+SCAN=1<FF>-32<FF>3<FF>B019C66209FA<FF>wt-iphone<FF>7A020C
+SCAN=2<FF>-74<FF>0<FF>DC0D30000053<FF>BW226<FF>040680
+SCAN=3<FF>-43<FF>0<FF>00158354F994<FF>LAPTOP-3L<FF>120104
+SCAN=E

3.1.4. +PAIRREQ *Pair Request*

Format: +PAIRREQ=Param1, Param2{, Param3}
Param1: Passkey (000000~999999)
Param2: MAC address (12 Bytes ASCII) of current pairing device
Param3: Name of current pairing device

3.1.5. +PAIRED *Pair Result*

Format: +PAIRED=Param1, Param2
Param1: Pair result
(0) Success
(1~255) Failed reason
Param2: MAC address (12 Bytes ASCII) of current pairing device

3.2. SPP Events

3.2.1. +SPPSTAT *SPP State*

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

3.2.2. +SPPDATA *SPP Received Incoming Data*

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

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

Description: Use socket instead of AT command for data read/write on Android/Linux platform

3.3. GATT Events

3.3.1. +GATTSTAT *GATT State*

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

3.3.2. +GATTDATA *GATT Received Incoming Data*

Format: +GATTDATA=Param1, Param2 Param1: Payload length Param2: Payload
Example: Received data "1234567890" from remote device via GATT << *+GATTDATA=10,1234567890*
Description: Use socket instead of AT command for data read/write on Android/Linux platform

4. Appendix