FSC-HC05 User Guide

Release 3.5.0

Table of contents

1 Hardware Design 2
1.1 Module Pin Diagram . . . 2
1.2 Pin Description . . . 3
1.3 Hardware Design Notes . . . 3

2 Function Description 4
2.1 Default Configuration . . . 4
2.2 GPIO Indicators . . . 4
2.2.1 Module Operation Status . . . 4
2.2.2 Data Transmission Mode . . . 5
2.2.3 Module Operation Status . . . 5
2.2.4 BT Connection Status . . . 5
2.3 Work Modes . . . 5
2.3.1 Throughput Mode . . . 5
2.3.2 Command Mode . . . 5
2.4 GATT Service . . . 6

3 Data Communication Principles 7
3.1 Communication Principle . . . 7
3.2 MCU-to-Module Communication . . . 8
3.3 Module-to-Module Communication . . . 9
3.4 Module-to-Phone Communication . . . 9
3.4.1 Why do we need to use an app on a mobile phone for Bluetooth connection and communication? . . . 9
3.4.2 Communication Application . . . 10

4 Quick Development Kit 11
4.1 Datasheet . . . 11
4.2 Evaluation Board . . . 11
4.3 AT Command Set . . . 11

4.4 Serial Port Tool . . . 11
4.5 APP&SDK . . . 11
4.6 Firmware Upgrade . . . 12
4.6.1 OTA Upgrade . . . 12

5 Quick Start 13
5.1 What you need . . . 13
5.1.1 Required Hardware . . . 13
5.2 Software and Setup . . . 13
5.3 Hardware Access . . . 13
5.4 Communication Test . . . 15
5.4.1 AT - UART Communication Test . . . 15
5.4.2 AT+NAME - Read/Write Local BR/EDR Name . . . 15
5.4.3 AT+VER - Read Current Firmware Version . . . 15

6 Development Examples 16
6.1 Data Throughput Mode Application . . . 16
6.1.1 What is Throughput Mode? . . . 16
6.1.2 Module to Phone Application . . . 17
6.1.3 Module to Module Application . . . 17
6.2 Read/Write Module Default Parameters . . . 19
6.3 Data Transmission Flow . . . 20

7 Firmware Upgrade 22
7.1 OTA Upgrade . . . 22
7.1.1 OTA Upgrade Tool . . . 22
7.1.2 OTA Upgrade guide . . . 22
7.1.3 OTA Upgrade Show . . . 23
7.2 FAQs . . . 24

8 FAQs 25
8.1 Why is a dedicated App needed for Bluetooth connection and communication on a phone? . . . 25
8.2 How to get the Bluetooth MAC address on an iOS phone? . . . 25

9 Contact Information 28

10 Appendix 29

[中文版]

This guide applies to:

> **FSC-HC05** Series of Bluetooth Dual-Mode Data Transmission Application Modules

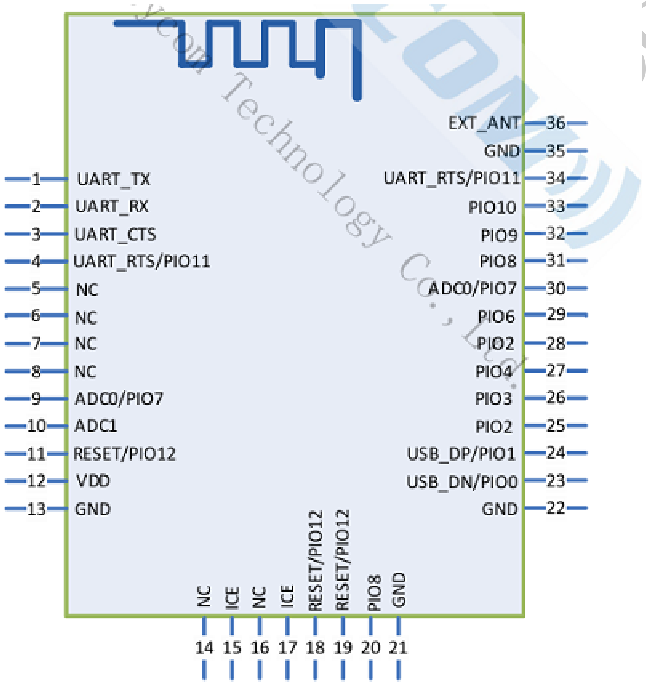
This guide consists of the following parts:

Chapter 1

Hardware Design

[中文版]

1.1 Module Pin Diagram



FSC-HC05 PIN Diagram(Top View)

1.2 Pin Description

Pin	Pin Name	Type	Pin Descriptions
1	UART_TX	O	Serial Transmitter Data
2	UART_RX	I	Serial Receiver Data
3	UART_CTS	I/O	UART Clear to Send(no connection request)
4	UART_RTS	I/O	UART Request to Send(no connection request)
9	Disc/USB_DM	I/O	Disconnect the connecting pin
11	RESET	I	Active-low reset input
12	VDD	Powe	Power supply voltage 3.3V, LDO power supply preferred
13	GND	GND	GND
15	ICE	I/O	Writing firmware pin
32	LED	O	Bluetooth not connected to output square wave, Bluetooth connected to output high level
33	STATUS	O	Connection state, output. H=Connected , L=No connection
34	Tran/USB_DP	O	UART Mode Control Pin: H = Command Mode; L = Transparent Transmission Mode
36	EXT_ANT	ANT	Changing the 0 ohm resistance near the antenna, it is possible to connect a Bluetooth antenna externally

1.3 Hardware Design Notes

- In simple test environment, the module can be used for basic testing and operation by simply connecting **VDD**, **GND**, **UART_RX**, and **UART_TX**.
- If the MCU needs to obtain the connection status of the Bluetooth module, it needs to be connected to the STATUS pin.
- After drawing the schematic diagram, please send it to Feasycom for review to avoid Bluetooth distance not achieving the best effect.

Chapter 2

Function Description

[中文版]

2.1 Default Configuration

Name	FSC-HC05
LE-Name	FSC-HC05-LE
Pin Code	1234
Secure Simple Pairing Mode	On
UART Baudrate	9600/8/N/1

2.2 GPIO Indicators

2.2.1 Module Operation Status

Pin	Status	Description
Pin 23	Low Level	/
Pin 23	High level	Bluetooth Disconnected

2.2.2 Data Transmission Mode

Pin	Status	Description
Pin 24	Low Level	Throughput Mode
Pin 24	High level	Command Mode

2.2.3 Module Operation Status

Pin	Status	Description
Pin 32	1Hz square wave	Bluetooth Disconnected
Pin 32	High level	Bluetooth Connected

2.2.4 BT Connection Status

Pin	Status	Description
Pin 33	Low Level	Bluetooth Disconnected
Pin 33	High level	Bluetooth Connected

2.3 Work Modes

2.3.1 Throughput Mode

- **Bluetooth Not Connected**: Data received via UART is parsed as AT commands.
- **Bluetooth Connected**: All data received via UART is sent as-is to the remote Bluetooth device.

2.3.2 Command Mode

- **Bluetooth Not Connected**: Data received via UART is parsed as AT commands.
- **Bluetooth Connected**: Data received via UART is still parsed as AT commands. Data must be sent to the remote device using AT commands, e.g., `AT+SPPSEND`.

2.4 GATT Service

Type	UUID	**Operation**	Description
Service	0xFFF0		Throughput transmission service
Write	0xFFF2	Write,Write Without Response	APP to Module
Notify	0xFFF1	Notify	Module to APP

Chapter 3

Data Communication Principles

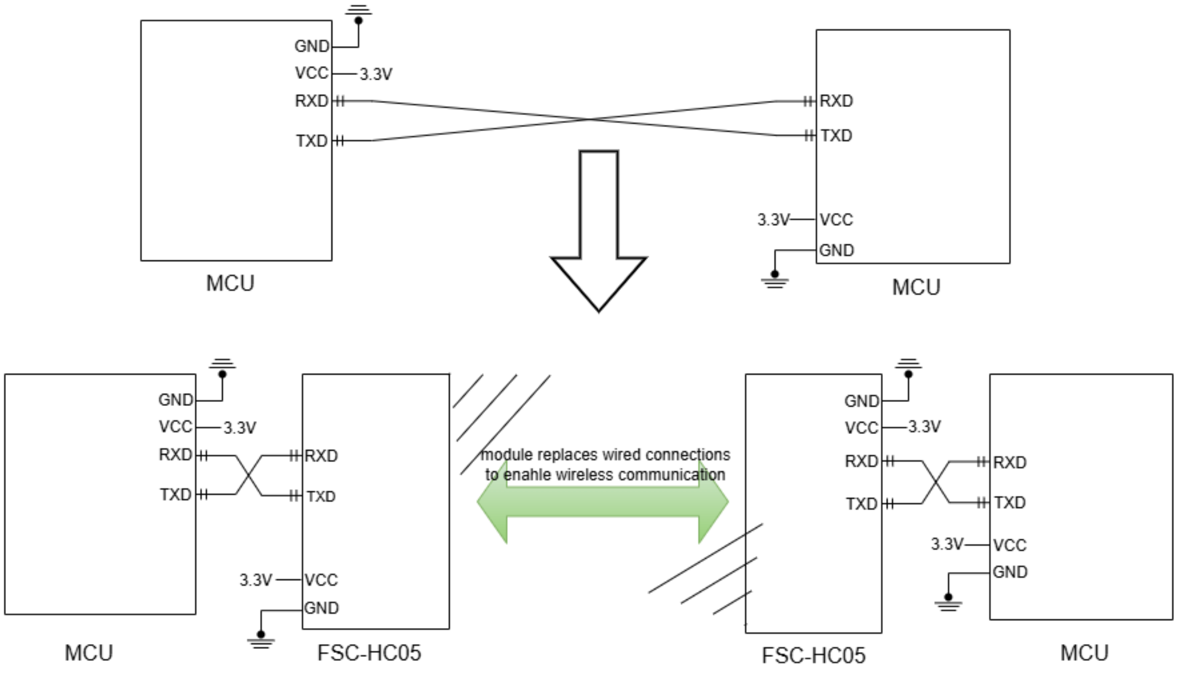
[中文版]

3.1 Communication Principle

FSC-HC05 series Bluetooth dual-mode data modules enable wireless communication between devices based on the SPP (Serial Port Profile) and BLE (Bluetooth Low Energy) dual-mode protocols.

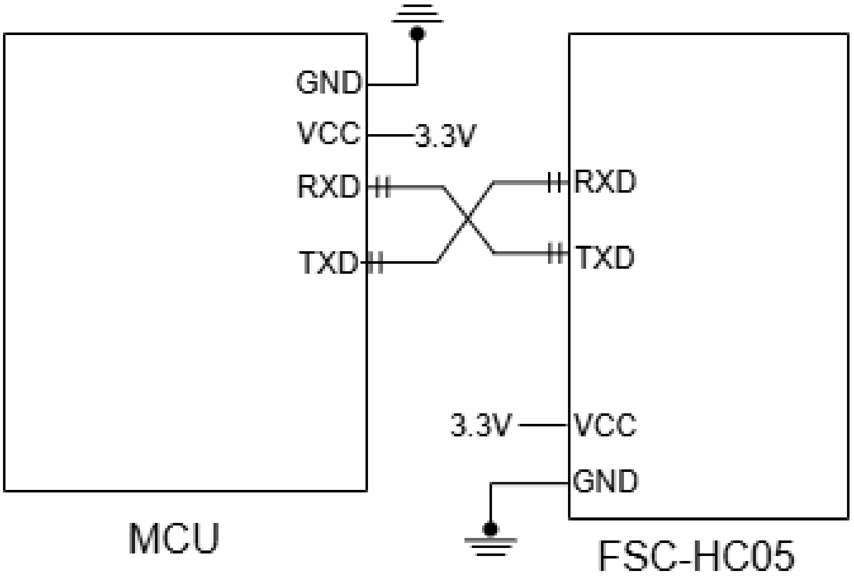
- **SPP Mode**: Emulates traditional serial port communication, establishing a virtual serial link through the RF layer. It supports continuous large data transfers (e.g., file transfer) and is suitable for scenarios like printers.
- **BLE Mode**: Utilizes an event-driven, low-power architecture, defining a “Service-Characteristic” model via the GATT protocol for intermittent small data interactions (e.g., sensor data), ideal for IoT devices.

Both modes share the underlying RF hardware and switch automatically via the protocol stack. The module communicates with the host device (phone/MCU) through the UART interface using AT commands or transparent data transmission to establish connections, exchange data, and manage status.



As shown in the figure, the Bluetooth module replaces the physical cable in full-duplex communication. A device like an MCU (left) sends data via its TXD pin to the Bluetooth module (left). The module’ s RXD port receives the UART data and automatically transmits it via radio waves over the air to the remote Bluetooth module. The remote Bluetooth module (right) receives the airborne data and delivers it via its TXD pin to the local device, like an MCU (right).

3.2 MCU-to-Module Communication

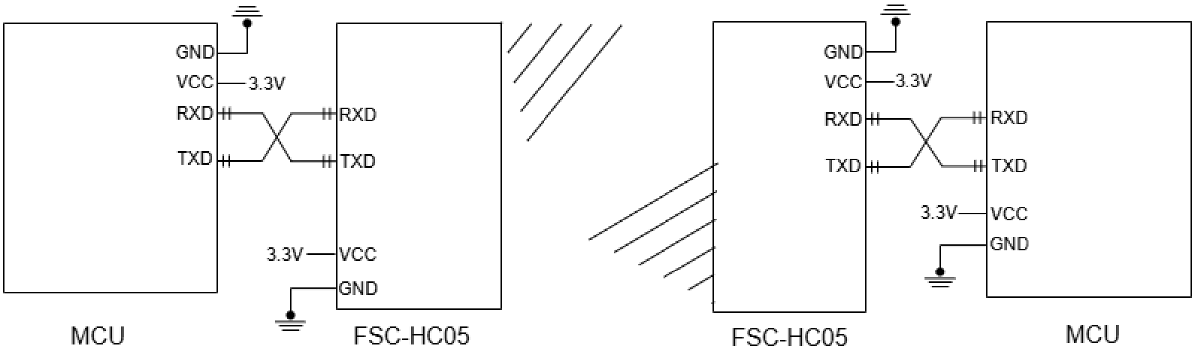


This diagram illustrates the connection between a master MCU (Microcontroller Unit) and an FSC-BT836 Bluetooth module via cross-connected serial ports, enabling command interaction between the master and the Bluetooth module to support wireless communication functions. It is applicable to IoT devices, remote control, and other scenarios.

1. **Serial Communication Interface**: The master MCU’ s transmit pin (MCU_TX) is cross-connected to the Bluetooth module’ s receive pin (UART_RX), and the master MCU’ s receive pin (MCU_RX) is similarly connected to the Bluetooth module’ s transmit pin (UART_TX), forming a bidirectional data transmission channel.
2. **Power and Grounding**: The Bluetooth module is powered by 3.3V through the VDD_3V3 pin and shares a common ground (GND) with the master MCU, ensuring level compatibility and signal stability.

3.3 Module-to-Module Communication

Two FSC-HC05 Bluetooth modules can establish a Bluetooth connection automatically upon power-up.



A module can act as a master device to connect to a slave device. The host can send commands to control the module for Bluetooth scanning, connection establishment, data transmission, and connection termination.

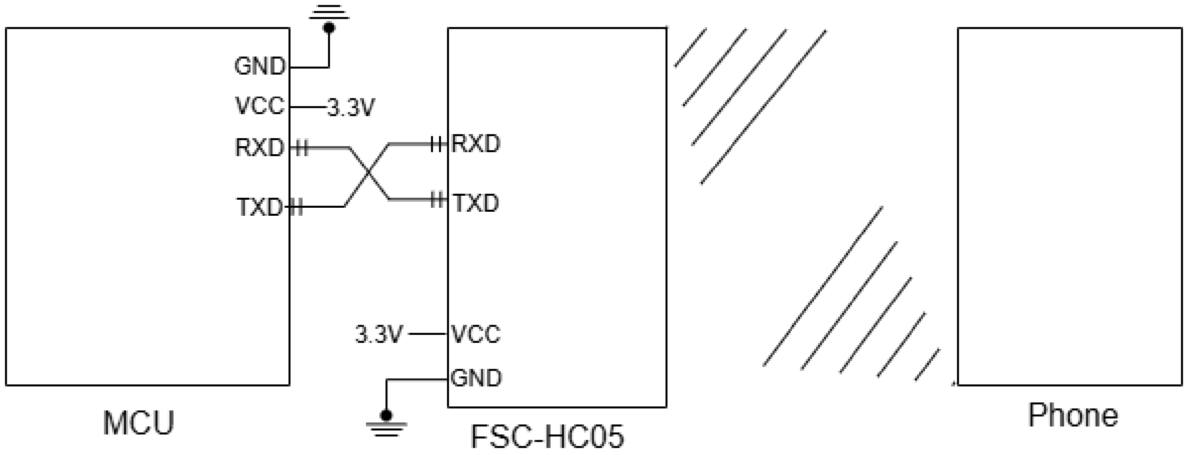
3.4 Module-to-Phone Communication

3.4.1 Why do we need to use an app on a mobile phone for Bluetooth connection and communication?

Native phone Bluetooth functionality primarily supports common use cases like audio transfer and file sharing. Some Bluetooth peripheral devices can be connected via the phone's built-in settings (e.g., Bluetooth speakers, headphones, keyboards, mice). However, when a peripheral device, like a module only supporting SPP/GATT protocols,

```
cannot be connected via native phone settings, a specific mobile ap-
plication, such as the FeasyBlue app, is generally required for con-
nection.
```

3.4.2 Communication Application



Bluetooth Module Side (FSC-HC05): Continuously transmits broadcast data after power-up.

Mobile Terminal : Can discover the broadcast packets via scanning and initiate a connection request to the module (FSC-HC05). Upon successful connection, the Bluetooth module (FSC-HC05) will pull the connection status pin HIGH and respond to status indication commands (valid in Command Mode) to notify the host of the successful Bluetooth connection.

Host: Can send data to the remote (Mobile Terminal) Bluetooth via the UART through the Bluetooth module. Conversely, the remote (Mobile Terminal) Bluetooth can also send data back to the host.

Chapter 4

Quick Development Kit

[中文版]

4.1 Datasheet

- FSC-HC05 Datasheet

4.2 Evaluation Board

- FSC-DB005： Feasycom USB-to-Serial Bluetooth Data Transmission Application Development Board.

4.3 AT Command Set

- FSC-HC05 General Data AT Command Set

4.4 Serial Port Tool

- Feasycom Serial Port Tool : A serial port communication analysis tool based on Windows PC.

4.5 APP&SDK

- FeasyBlue： Feasycom App & SDK resource supporting Android and iOS, which enables functions such as Bluetooth BLE & SPP data communication test, Feasycom module firmware version reading, parameter configuration, OTA Upgrade and OTA AT commands etc.

4.6 Firmware Upgrade

4.6.1 OTA Upgrade

- Tool : FeasyBlue APP
- User Guide : Please refer to FSC-HC05 - OTA Upgrade.

Chapter 5

Quick Start

[中文版]

5.1 What you need

5.1.1 Required Hardware

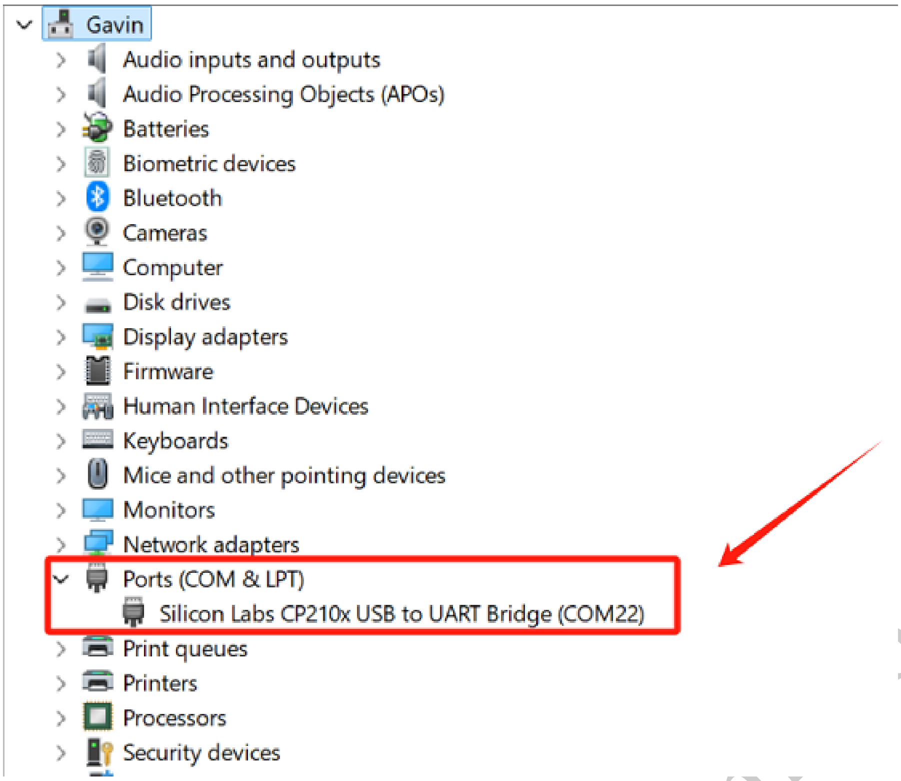
- 1 x FSC-DB005-HC05 Rapid Development Kit : FSC-DB005 USB-to-Serial Rapid Evaluation Board pre-integrated with FSC-HC05 module (Take FSC-BT836B an example) .
- 1 x PC (Windows / Mac)

5.2 Software and Setup

- Feasycom Serial Port Tool : A serial port communication analysis tool based on Windows PC.
- Communication Interface : UART
- UART Configuration： 115200/8/N/1

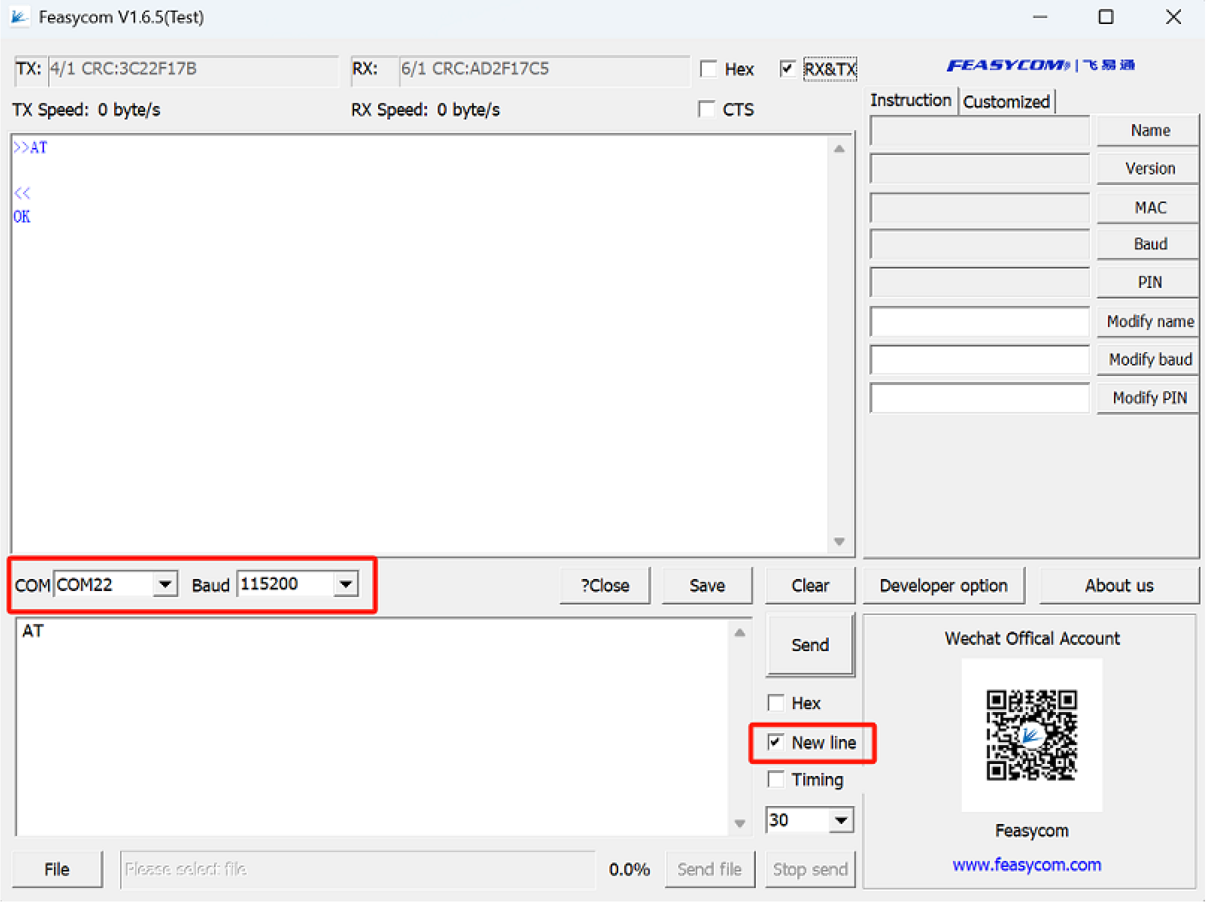
5.3 Hardware Access

1.Connect the FSC-DB005-HC05 to the PC via USB. The PC will automatically recognize the serial port and create a virtual COM port.



- Gavin
  - Audio inputs and outputs
  - Audio Processing Objects (APOs)
  - Batteries
  - Biometric devices
  - Bluetooth
  - Cameras
  - Computer
  - Disk drives
  - Display adapters
  - Firmware
  - Human Interface Devices
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - Silicon Labs CP210x USB to UART Bridge (COM22)
  - Print queues
  - Printers
  - Processors
  - Security devices

2.Run the Feasycom Serial Port Tool on the PC, set the correct **COM** and **Baud**, and check the **New Line**.



Feasycom V1.6.5(Test)

TX: 4/1 CRC:3C22F17B RX: 6/1 CRC:AD2F17C5 □ Hex ☑ RX&TX

TX Speed: 0 byte/s RX Speed: 0 byte/s □ CTS

```
>>AT

<<
OK
```

COM COM22 Baud 115200

AT

□ Hex
☑ New line
□ Timing
30

Please select file 0.0%

5.4 Communication Test

The following are several basic general AT command test examples.

For more AT commands, please refer to FSC-HC05 General Data AT Command Set .

5.4.1 AT - UART Communication Test

Command	AT\r\n
Response	**\r\nOK\r\n**
Description	Test the UART communication between HOST and Module after power on, baudrate changed, etc.

Example:

```
send:     >>AT\r\n
response: <<\r\nOK\r\n   //Successfully connected.
```

5.4.2 AT+NAME - Read/Write Local BR/EDR Name

Example: Read BR/EDR Name

```
send:     <<AT+NAME\r\n
response: >>\r\n+NAME=FSC-HC05\r\n   //Example, please refer to the␣
↪actual reading result
response: >>\r\nOK\r\n
```

5.4.3 AT+VER - Read Current Firmware Version

Example: Read Current Firmware Version

```
send:     <<AT+VER\r\n
response: >>\r\n+VER=1.0.0,FSC-HC05\r\n     //Example, please refer␣
↪to the actual reading result
response: >>\r\nOK\r\n
```

Chapter 6

Development Examples

[中文版]

6.1 Data Throughput Mode Application

6.1.1 What is Throughput Mode?

FSC-HC05 series dual-mode Bluetooth data transmission modules have two data transmission modes: **Throughput Mode** and **Command Mode**.

The generic data throughput firmware for the FSC-HC05 series modules default to throughput mode. To switch modes, refer to the FSC-HC05 General Dual-Mode Data AT Command Set and use the **AT+TPMODE** command.

The operation and differences between the two modes are as follows:

- **Throughput Mode**:

  Bluetooth Not Connected: Data received via UART is parsed as AT commands.

  Bluetooth Connected: All data received via UART is sent as-is to the remote Bluetooth device. It does not contain any data headers or framing and does not require AT commands to send data.

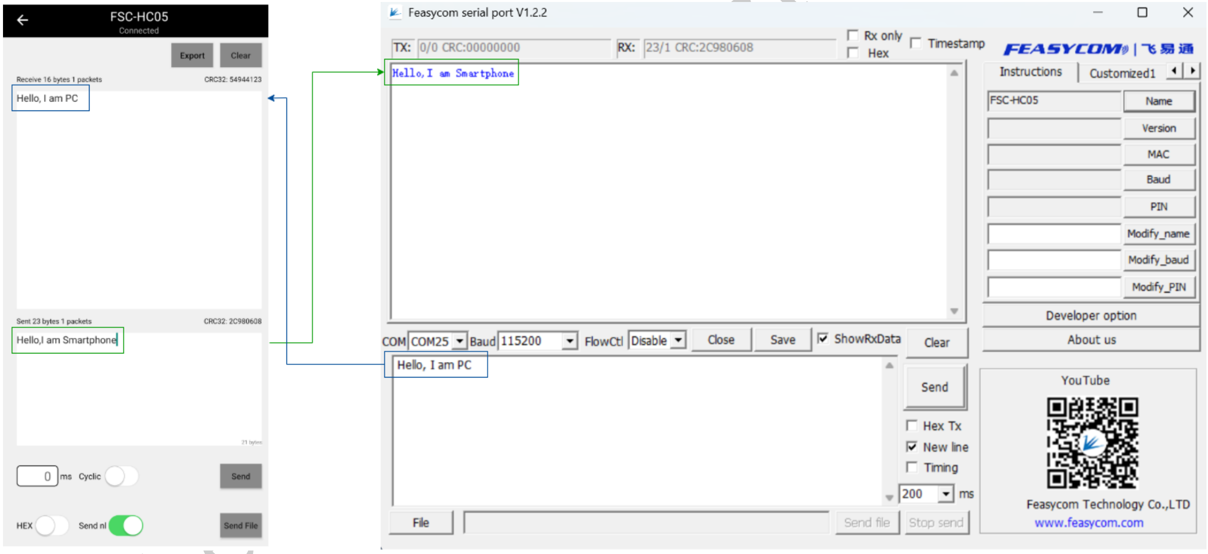
- **Command Mode**:

  Bluetooth Not Connected: Data received via UART is parsed as AT commands.

  Bluetooth Connected: Data received via UART is still parsed as AT commands. It will contain specific response indication headers and framing. Data must be sent to the remote device using AT commands, such as **AT+SPPSEND** or **AT+LESEND** .

6.1.2 Module to Phone Application

1. **Module**: After power-on, the module will continuously send broadcast packet data.
2. **Phone**: Open the FeasyBlue APP , scan for nearby Bluetooth device advertisements, find the target Bluetooth module, and establish a connection.
3. **Connection Success**: After successful connection, the status pin of the module will pull up the level, indicating that the connection has been established.
4. **Data Transmission**: After successful connection, in the througput mode, the module will automatically transmit the serial port data it receives to the remote end (mobile phone side) via air.



Hello, I am PC

Hello,I am Smartphone

Hello,I am Smartphone

Hello, I am PC

6.1.3 Module to Module Application

Throughput Transmission Demonstration of SPP Communication Between FSC-HC05x and FSC-BT910x Bluetooth Modules, as follow:

1. FSC-HC05 Set to master mode and enable inquiry.

```
Send: >>AT+ROLE=1      //Set the module as master
Response: <<OK

```

(continues on next page)

(continued from previous page)

```
Send: >>AT+CLASS=0    //Set to be able to inquire about various Bluetooth device classes
Response: <<OK
```

1. FSC-HC05 Enable scanning.

```
Send:     >>AT+INQ                     //Scan for nearby Bluetooth SPP devices
Response: <<
    +INQ:F0FA:C7:82C255,280704,FFCB
    +INQ:DC0D:30:44B,240404,FFF2
    +INQ:8667:7A:3CAAA,40680,FFBB
    +INQ:B48C:9D:D259D6,2A410C,FFD3
    +INQ:E0D8:C4:660C75,2C043C,FFC4
    +INQ:DC0D:30:1ADB,240408,FFDC
    +INQ:DC0D:30:2034,40680,FFC9
    +INQ:2:5B:FF03,5A020C,FFC1
    OK                           //Scan ended
```

3. Use the AT+LINK command to initiate an SPP connection request to the remote target device.

```
Send: >>AT+LINK=DC0D,30,44B       //Initiate SPP link request to the remote target device
Response: <<OK            //Sent successfully
```

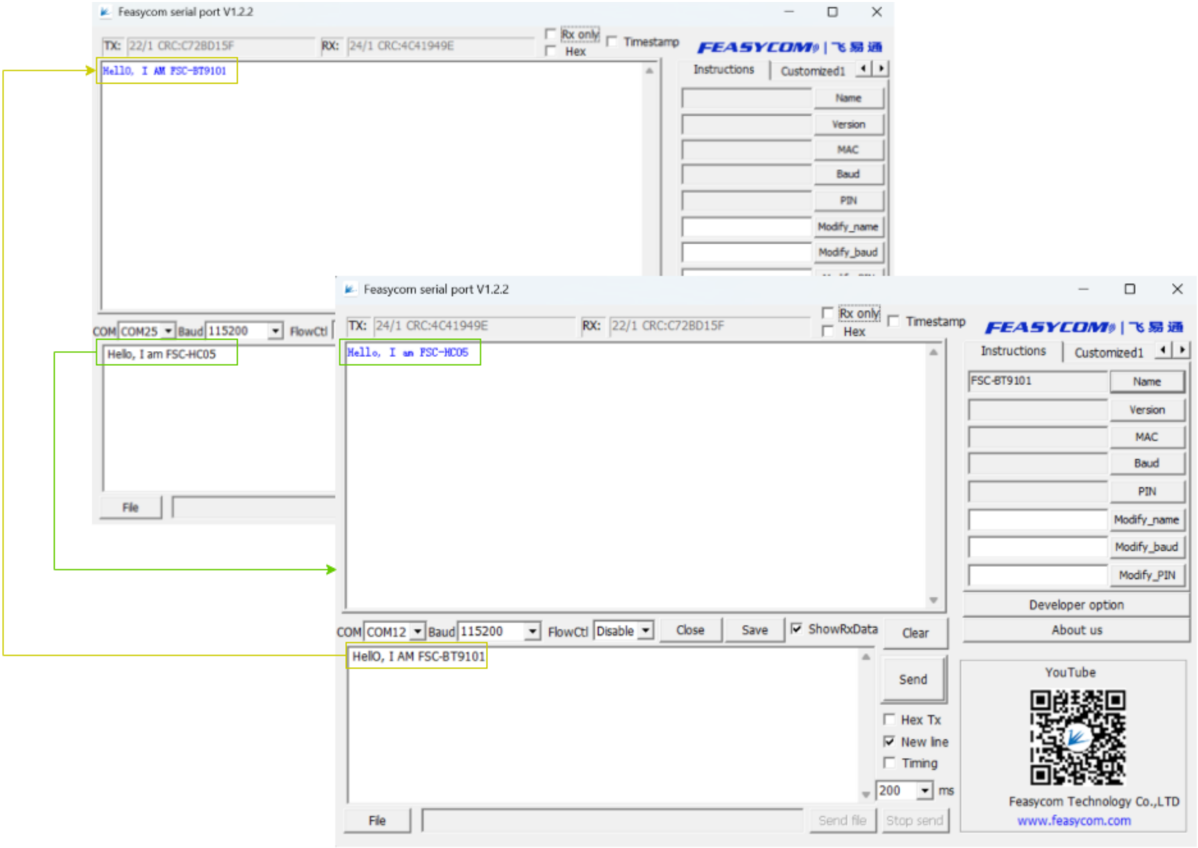
3.SPP Connected

In throughput mode, after a successful Bluetooth connection, the UART will not receive event response indicators. The connection status can be determined by the level state of the status pin (Pin 33) on the FSC-HC05, as follows:

- **High Level (H):** Indicates Bluetooth is successfully connected.
- **Low Level (L):** Indicates Bluetooth is not connected or the connection has been disconnected.

4.Send data

Throughput mode is enabled by default in the generic data transmission firmware. After SPP connected, data can be sent directly without using AT commands, as follow:



Feasycom serial port V1.2.2

TX: 22/1 CRC:C728D15F RX: 24/1 CRC:4C41949E

Hello, I AM FSC-BT9101

COM COM25 Baud 115200

Hello, I am FSC-HC05

Feasycom serial port V1.2.2

TX: 24/1 CRC:4C41949E RX: 22/1 CRC:C728D15F

Hello, I am FSC-HC05

FSC-BT9101

COM COM12 Baud 115200 FlowCtl Disable

☑ ShowRxData

Hello, I AM FSC-BT9101

□ Hex Tx

☑ New line

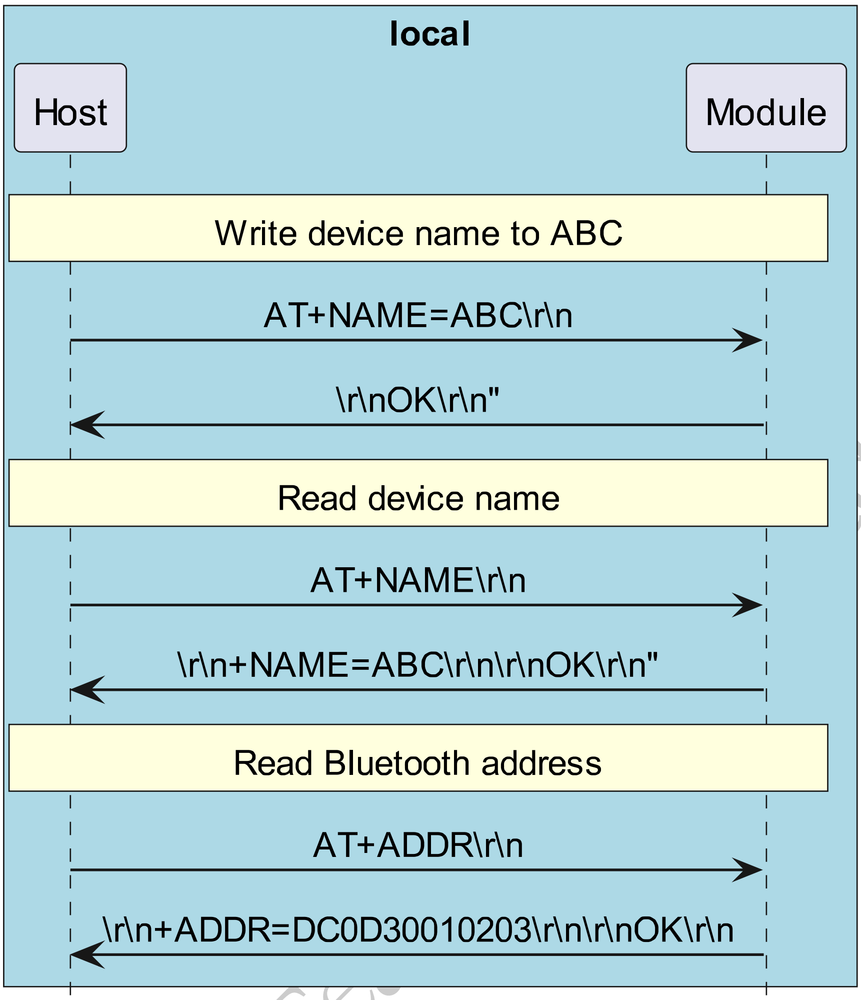
□ Timing

200 ms

6.2 Read/Write Module Default Parameters

When Bluetooth is not connected, the module parses UART data as AT commands. The host can query and modify the module’ s default parameters. As follow:

1. Write the device name to ABC
2. Read the device name
3. Read the Bluetooth address

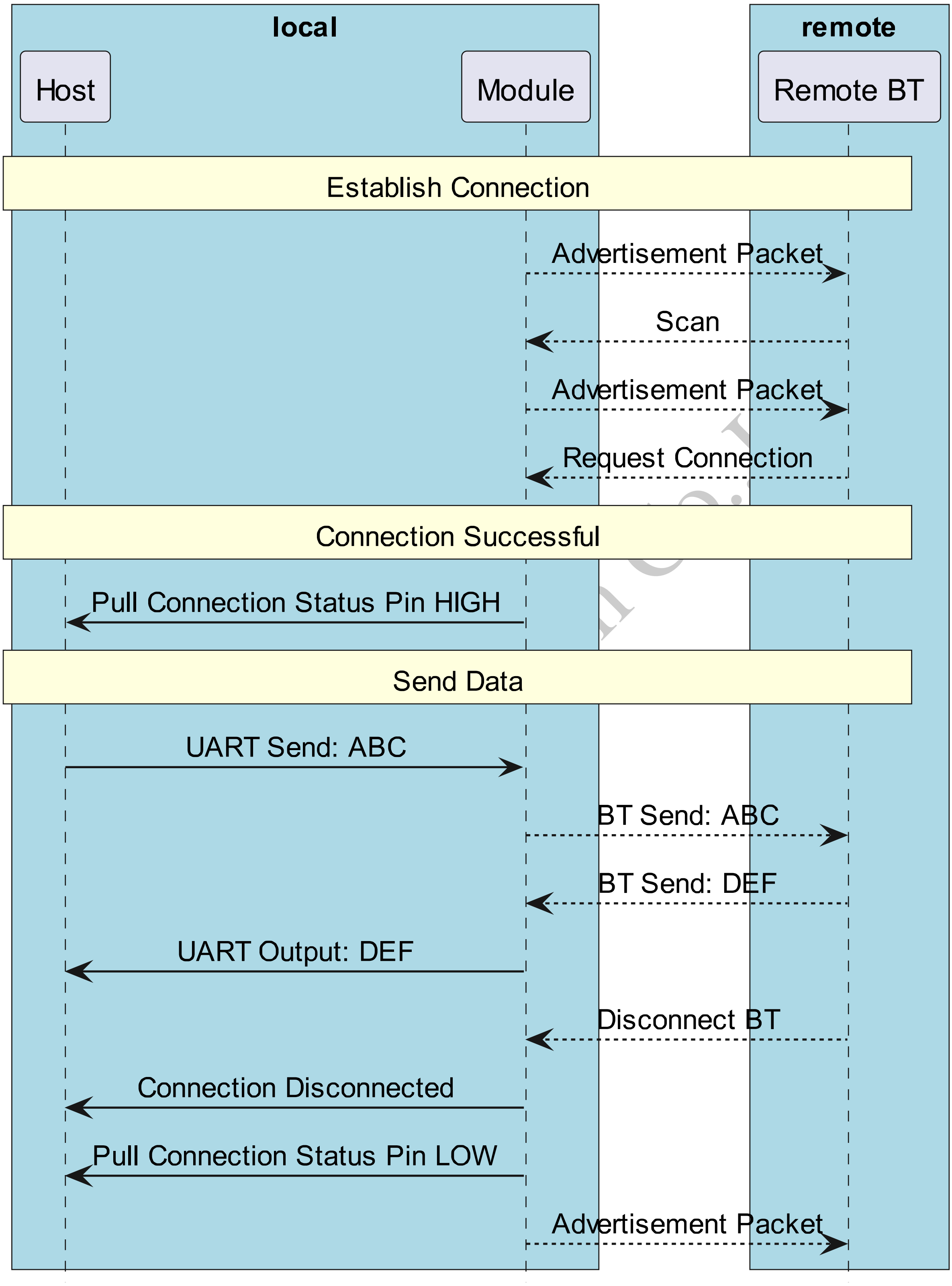


6.3 Data Transmission Flow

Upon power-up, the module continuously transmits advertisement data. A remote Bluetooth device (e.g., phone) can discover these advertisement packets via scanning and initiate a connection request to the module.

Upon successful connection, the module pulls its connection status pin HIGH to notify the host of the successful Bluetooth connection.

The host can send data to the remote Bluetooth device via the Bluetooth module, and the remote Bluetooth device can also send data to the host.


local
remote
Host
Module
Remote BT
Establish Connection
Advertisement Packet
Scan
Advertisement Packet
Request Connection
Connection Successful
Pull Connection Status Pin HIGH
Send Data
UART Send: ABC
BT Send: ABC
BT Send: DEF
UART Output: DEF
Disconnect BT
Connection Disconnected
Pull Connection Status Pin LOW
Advertisement Packet

Chapter 7

Firmware Upgrade

[中文版]

7.1 OTA Upgrade

7.1.1 OTA Upgrade Tool

- FeasyBlue App （Based on Android & iOS）

7.1.2 OTA Upgrade guide

1.Run the FeasyBlue App, select **Settings - OTA Upgrade** to enter the OTA Upgrade function section. After entering, you will navigate to the device search interface to select the Bluetooth device to be upgraded；

2.Search for and select the Bluetooth device that needs upgrading. After selecting the device, you will enter the **Load Firmware** function interface to load the firmware upgrade file；

3.There are two ways to load the firmware upgrade file:

- **Select Firmware**: Choose to load the firmware upgrade file stored in the mobile phone’ s local storage (the firmware upgrade file is provided by Feasycom).
- **Get Firmware**: Enter the DFU name to download and import the corresponding firmware upgrade file from the cloud server via the network (the DFU name is provided by Feasycom).

4.After the firmware file is loaded successfully, click the **Start Upgrade** button. The interface will display “Upgrading” and the upgrade progress, indicating that the upgrade mode is activated and the upgrade is in progress.

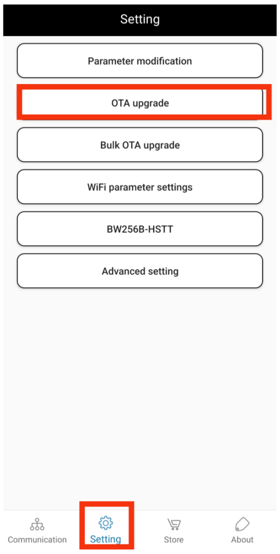
5.Wait for the upgrade progress bar to complete and the interface to display “Upgrade Completed”—the upgrade is then finished.

Warning：

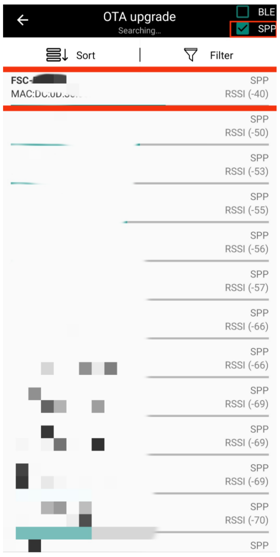
1. When installing and running the FeasyBlue App, please allow the App to access the permissions for **nearby devices**, **location information**, and **media and file access**. Otherwise, you may fail to search for nearby Bluetooth devices and load the firmware upgrade file；
2. If you use the “Obtain Firmware” method (importing the firmware upgrade file by entering the DFU name), note that the mobile phone must be connected to the Internet, and ensure the DFU name is entered correctly (case-sensitive). Otherwise, an error of “network or file error” may occur；
3. Do not disconnect the power during the upgrade process.

7.1.3 OTA Upgrade Show

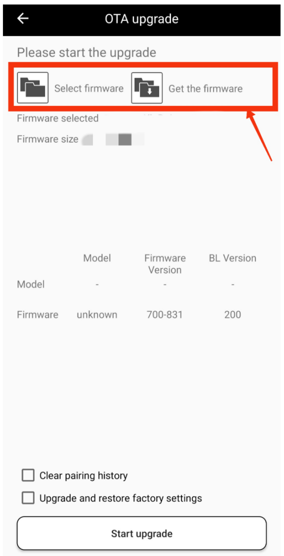
Note: The following diagram demonstration is based on the Android platform. The device names, parameters, and firmware DFU names involved are for demonstration purposes only. Please refer to the actual ones during project operation.



(1)



(2)



Firmware selected

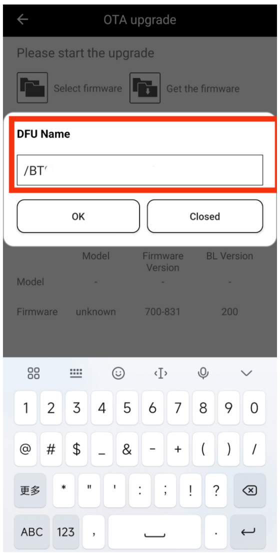
Firmware size

	Model	Firmware Version	BL Version
Model	-	-	-
Firmware	unknown	700-831	200

☐ Clear pairing history

☐ Upgrade and restore factory settings

(3)

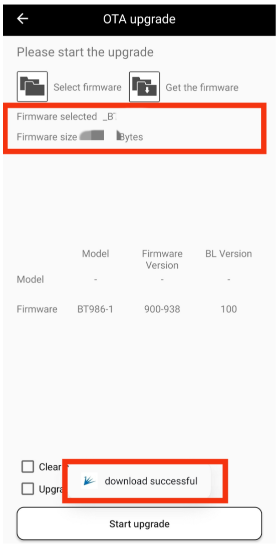


DFU Name

/BT'

	Model	Firmware Version	BL Version
Model	-	-	-
Firmware	unknown	700-831	200

(4)



Firmware selected _B

Firmware size Bytes

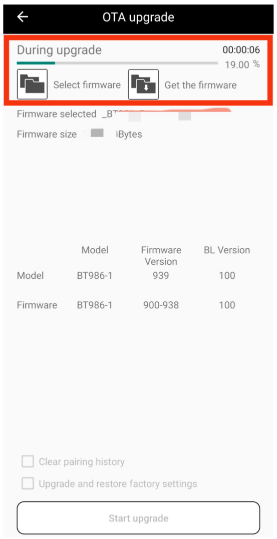
	Model	Firmware Version	BL Version
Model	-	-	-
Firmware	BT986-1	900-938	100

☐ Clear

☐ Upgra

download successful

(5)



During upgrade 00:00:06

19.00 %

Firmware selected _BT

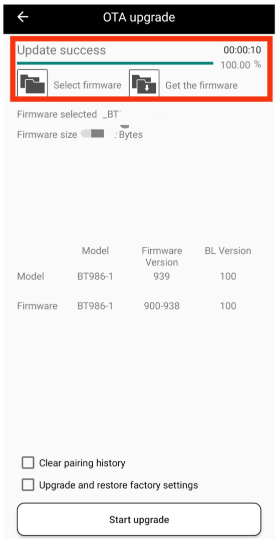
Firmware size Bytes

	Model	Firmware Version	BL Version
Model	BT986-1	939	100
Firmware	BT986-1	900-938	100

☐ Clear pairing history

☐ Upgrade and restore factory settings

(6)



Update success 00:00:10

100.00 %

Firmware selected _BT

Firmware size Bytes

	Model	Firmware Version	BL Version
Model	BT986-1	939	100
Firmware	BT986-1	900-938	100

☐ Clear pairing history

☐ Upgrade and restore factory settings

(7)

7.2 FAQs

Q: If an abnormality occurs during the upgrade process, such as an interrupted upgrade, or the module fails to upgrade successfully after the upgrade is completed, what should I do?

A： You can try to power the device on again, repeat the upgrade operation, and check if it can enter the upgrade mode normally. If not, please contact the Feasycom technical team.

Chapter 8

FAQs

[中文版]

8.1 Why is a dedicated App needed for Bluetooth connection and communication on a phone?

Native phone Bluetooth functionality primarily supports common use cases like audio transfer and file sharing. Some Bluetooth peripheral devices can be connected via the phone's built-in settings (e.g., Bluetooth speakers, headphones, keyboards, mice). However, when a peripheral device, like a module only supporting SPP/GATT protocols, cannot be connected via native phone settings, a specific mobile application, such as the FeasyBlue app, is generally required for connection.

8.2 How to get the Bluetooth MAC address on an iOS phone?

Due to security considerations, the iOS system converts the Bluetooth MAC address into a UUID at the underlying level before presenting it to upper-layer applications. Therefore, apps cannot directly obtain the device’ s actual MAC address.

The FSC-BT826x series Bluetooth modules include the MAC address in their broadcasts by default. Apps can retrieve the MAC address from the advertisement packet using the following method.

```
- (void)centralManager:(CBCentralManager *)central␣
↪didDiscoverPeripheral:(CBPeripheral *)peripheral␣
↪advertisementData:(NSDictionary *)advertisementData RSSI:(NSNumber␣
↪*)RSSI
{
    if(![self describeDictonary:advertisementData])
    {
        NSLog(@"is not fsc module");
        return;
    }
}

- (Boolean)describeDictonary: (NSDictionary *) dict
{
    NSArray *keys;
    id key;
    keys = [dict allKeys];
    for(int i = 0; i < [keys count]; i++)
    {
        key = [keys objectAtIndex:i];
        if([key isEqualToString:@"kCBAdvDataManufacturerData"])
        {
            NSData *tempValue = [dict objectForKey:key];
            const Byte *tempByte = [tempValue bytes];
            if([tempValue length] == 6)
            {
                // tempByte Subsequent parameters are the Bluetooth␣
↪address
                return true
            }
        }else if([key isEqualToString:@"kCBAdvDataLocalName"])
        {
            //there is name
            //NSString *szName = [dict objectForKey: key];
        }
    }
```

(continues on next page)

(continued from previous page)

```
    return false;
}
```

Chapter 9

Contact Information

Shenzhen Feasycom Co.,Ltd.

Address : Rm 508, Building, Fenghuang Zhigu, NO.50, Tiezai Road, Xixiang, Baoan Dist, Shenzhen, 518100, China.

Telephone : 86-755-23062695

Support : support@feasycom.com

Sales Service : sales@feasycom.com

Home Page : www.feasycom.com

Support Forum : forum.feasycom.com

Chapter 10

Appendix

PDF Download