FSC-BT630x User Guide

Release 3.5.1

Table of contents

[中文版]

This guide applies to the **FSC-BT630x** series Bluetooth BLE data throughput transmission modules, include:

- FSC-BT630B

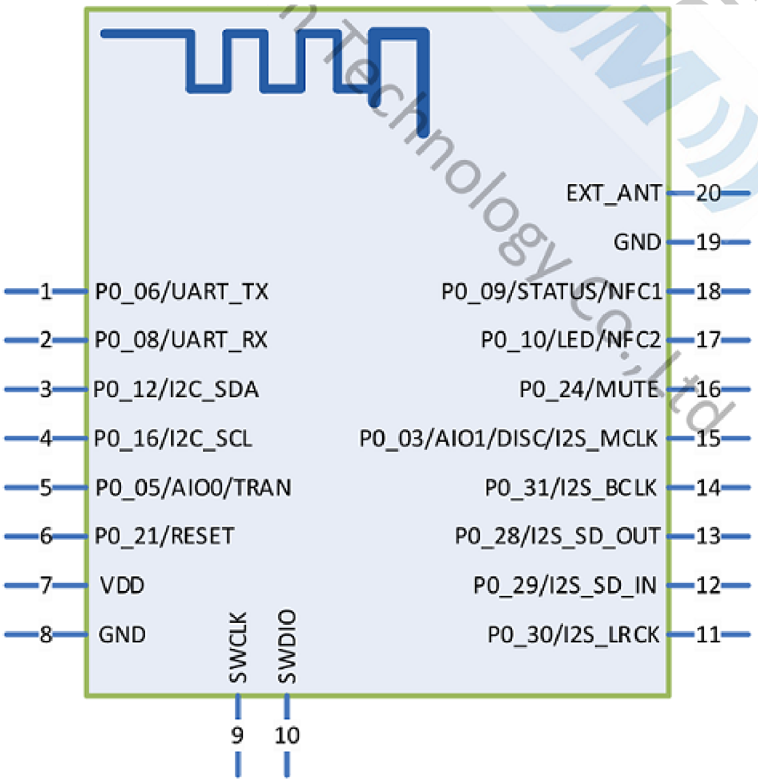
This guide consists of the following parts:

Chapter 1

Hardware Design

[中文版]

1.1 Module Pin Diagram



PIN Diagram (Top View)

1.2 Pin Description

Pin	Pin Name	Type	Pin Descriptions
1	UART_T	O	UART Data Pin
2	UART_R	I	UART Data Pin
6	RESET	I	Active-Low Reset
7	VDD	Powe	3.3V Power Supply.<LDO power supply is recommended
8	GND	GND	GND
9	SWCLK	I/O	Programming Pin
10	SWDIO	I/O	Programming Pin
7	WAKE_U	I	When MCU pulls high, the module enters sleep mode.when MCU pulls low, the module exits sleep mode
17	LED	O	Outputs square wave when Bluetooth is disconnected.outputs high level when Bluetooth is connected
18	STATUS	O	Outputs low level when Bluetooth is disconnected. outputs high level when Bluetooth is connected.
20	EXT_AN	ANT	Changing the 0Ω resistor near the antenna allows external Bluetooth antenna connection

1.3 Hardware Design Note

- The module can be used by simply connecting VDD / GND / STATUS / UART_RX / UART_TX.
- If the MCU needs to obtain the connection status of the Bluetooth module, the STATUS pin needs to be connected
- After completing the schematic diagram, please send it to Feiyitong for review to avoid the Bluetooth distance not reaching the optimal effect
- The module supports wake-up via UART
- VDD/GND/RESET/SWCLK/SWDIO are programming ports, and test points can be reserved

Chapter 2

Functional Description

[中文版]

2.1 Default Configuration

Name	Feasycom
UART Baudrate	115200/8/N/1

2.2 GPIO Indications

2.2.1 LED

Pin	Status	Description
Pin 17	2Hz Square Wave	Bluetooth Disconnected
Pin 17	High Level	Bluetooth Connected

2.2.2 Connection Status

Pin	Status	Description
Pin 18	Low Level	Bluetooth Disconnected
Pin 18	High Level	Bluetooth Connected

2.3 GATT Service

Type	UUID	Permission	Description
Service	0xFFF0		Throughput transmission service
Write	0xFFF2	Write，Write Without Response	Sent from APP to Module
Notify	0xFFF1	Notify	Sent from Module to APP

2.4 Working Mode

2.4.1 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.

2.4.2 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+LESEND`.

2.5 Low Power Consumption

The module supports one low-power mode: UART Wake-up Mode.

2.5.1 UART Wake-up Mode

- **Command Configuration** : `AT+LPM=1`
- **Sleep Method** :

  If there is no data communication via the UART for more than 5 seconds, it will automatically enter sleep mode.

  After entering sleep mode, the UART will exit sleep mode when the first frame of data is received.

- **Description** :

  The first frame of data upon wake-up will be lost.

  Simple logic, saving IO resources.

2.6 Data Transmission Rate

Baud Rate	Data Packet (Size)	Transmission Interval	Connection Interval	Transmission Method	Rate
921600	244	20ms	7.5ms	Notify	12000 Byte/s

Chapter 3

Data Communication Principles

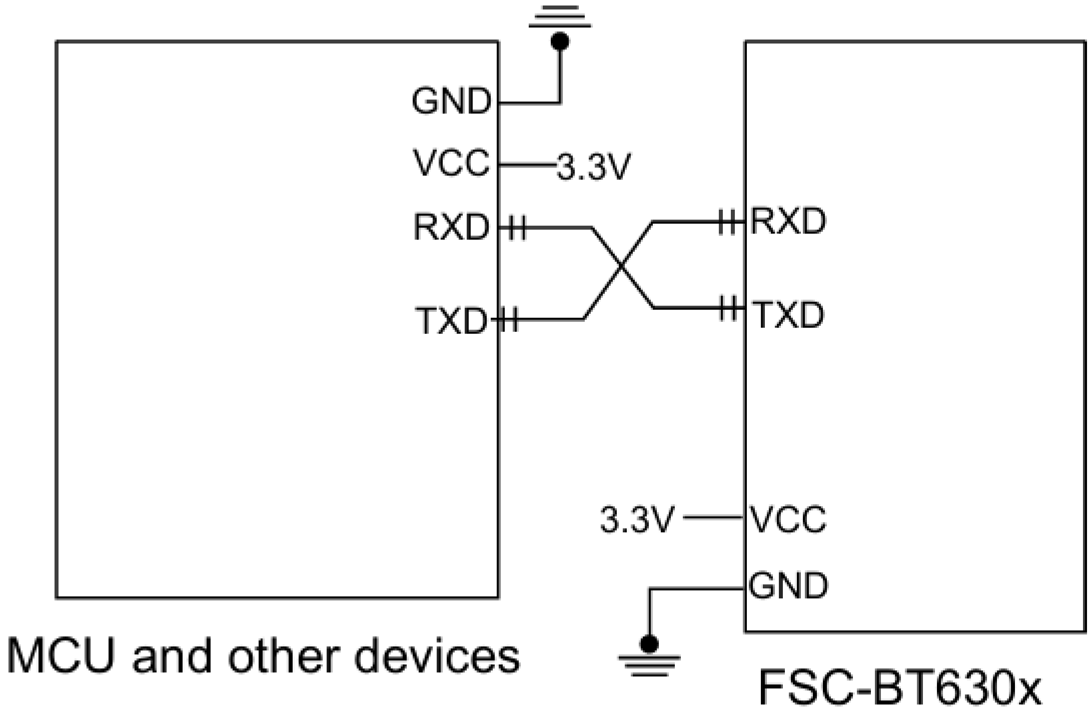
[中文版]

3.1 Working Principle

FSC-BT630x Bluetooth Data Transmission Module enables wireless communication between devices based on the BLE (Bluetooth Low Energy) protocol.

- **BLE** : It adopts an event-driven low-power architecture and defines the “Service-Characteristic” model via the GATT (Generic Attribute Profile) protocol. This enables intermittent small data interaction (e.g., sensor data), making it suitable for Internet of Things (IoT) devices

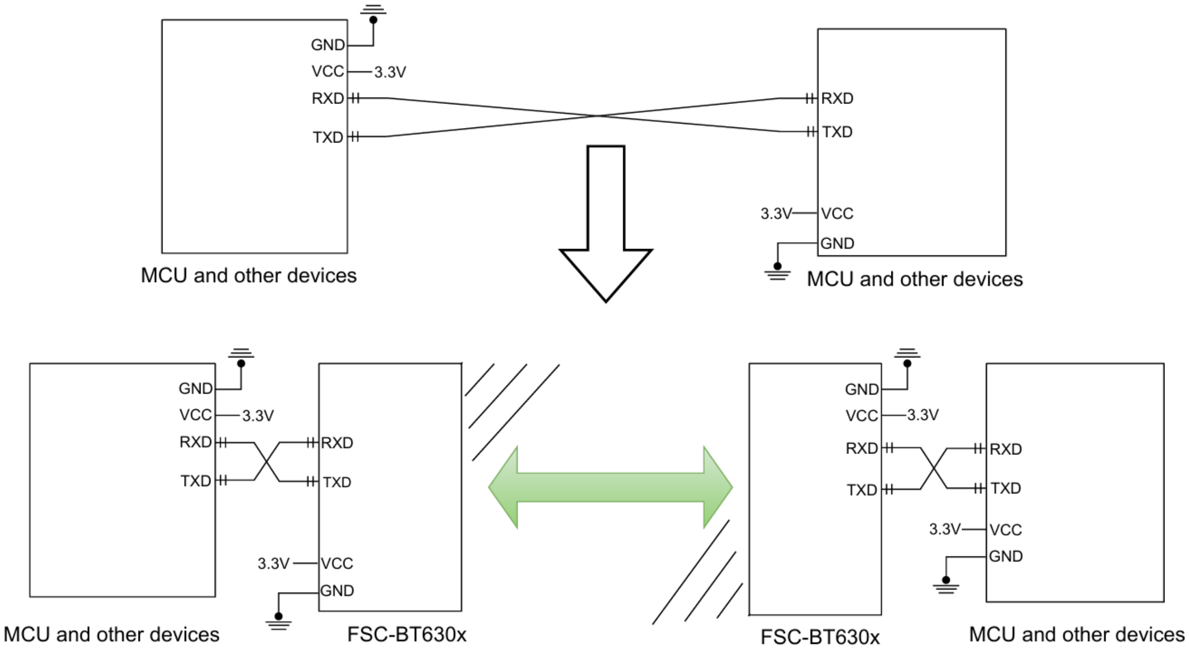
The module and host devices (mobile phones/MCUs) can send AT commands or transparent transmission data via UART to complete connection establishment, data exchange, and status management.



As shown in the figure, the Bluetooth module is used to replace the physical wires in full-duplex communication.

Devices such as microcontroller units (MCUs) (left side) transmit data via TXD to the Bluetooth module (left side). After the Bluetooth module receives the serial data through its RXD port, it automatically sends the data via radio waves through the air to the remote Bluetooth module. The remote Bluetooth module (right side) receives the over-the-air data and transmits it via TXD to local devices such as microcontroller units (MCUs) (right side).

3.2 MCU-to-Module Communication

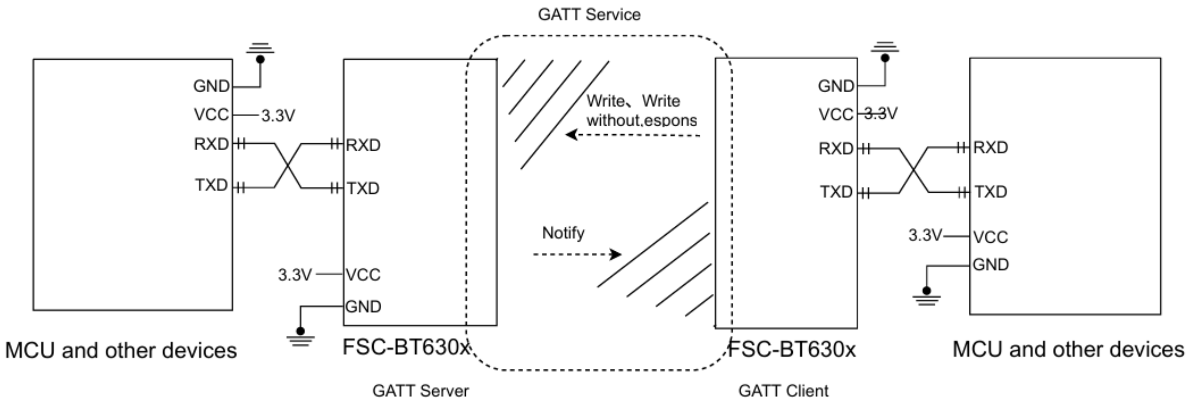


This diagram shows a connection diagram between a main control MCU (Microcontroller Unit) and an FSC-BT630 Bluetooth module. Command interaction between the main control unit and the Bluetooth module is realized through cross-connection via serial ports, supporting wireless communication functions, and is applicable to scenarios such as IoT devices and remote control.

1. **Serial Communication Interface：** The transmitting end of the main MCU (MCU_TX) is cross-connected with the receiving end of the Bluetooth module (UART_RX), and the receiving end (MCU_RX) is similarly connected to the transmitting end of the Bluetooth module (UART_TX), forming a two-way data transmission channel；
2. **Power Supply and GND:** The Bluetooth module is connected to 3.3V via the VDD_3V3 pin and shares a common GND with the main MCU to ensure level compatibility and signal stability.

3.3 Module-to-Module Communication

Two FSC-BT630x Bluetooth modules can establish a Bluetooth connection once powered on.



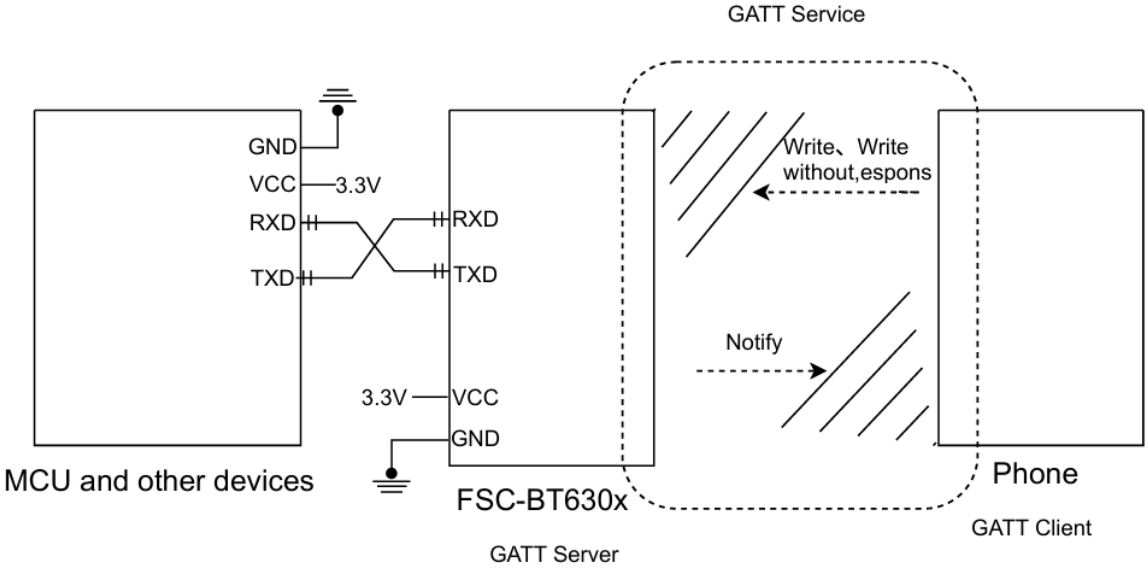
FSC-BT630x module features master-slave device functionality: the left module can be configured as a master device, and the right module as a slave device. The master device can perform operations such as Bluetooth scanning, connection establishment, data transmission, and disconnection by sending commands. Herein, a device that actively initiates a Bluetooth connection is defined as a master device, while a device that receives connection requests is referred to as a slave device.

3.4 Module-to-Phone Communication

3.4.1 Why is an APP required on a mobile phone for Bluetooth connection and communication?

The native Bluetooth function of mobile phones only supports general scenarios, such as audio transmission and file transfer. Some Bluetooth peripheral devices can be connected through the mobile phone's built-in settings program, such as Bluetooth speakers, Bluetooth headsets, Bluetooth keyboards, Bluetooth mice, etc. When a Bluetooth peripheral device cannot be connected by the mobile phone's native settings program (for example, the Bluetooth module only supports SPP/GATT protocols), to connect such a module, it is generally necessary to install a specific mobile application on the mobile phone, such as the FeasyBlue application

3.4.2 Communication Application



Bluetooth Module side (FSC-BT630x) : It will continuously send broadcast data outward when powered on.

Mobile Phone side : It can search and obtain the broadcast packet of the FSC-BT630x module through the FeasyBlue App , and initiate a MAC address/UUID connection request to the module side (FSC-BT630x), while obtaining all services and characteristics provided by the device.

After a successful connection, the Bluetooth module (FSC-BT630x) will pull high the connection status pin and report the connection status command (valid in command mode) to notify the host side of the successful Bluetooth connection.

Host side : Data can be sent to the remote (mobile phone side) Bluetooth via the serial port through the Bluetooth module, and the remote (mobile phone side) Bluetooth can also send data to the host .

Chapter 4

Quick Development Kit

[中文版]

4.1 Datasheet

- FSC-BT630x Datasheet

4.2 Evaluation Board

- FSC-DB006： Feasycom Bluetooth Data Throughput Transmission Module Development Board

4.3 AT Command Set

- FSC-BT630x General Data AT Command Set

4.4 Serial Port Tool

- Feasycom Serial Port Tool：A serial 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, and parameter configuration and OTA AT commands etc.

4.6 Firmware Upgrade

4.6.1 OTA Upgrade

- Android : nRF connect App or search for **nRF connect** in the Google play to download and install.
- iOS : Search for **nRF connect** in the **App Store** to download and install.
- User Guide : Please refer to FSC-BT630x - OTA Upgrade

Chapter 5

Quick Start

[中文版]

5.1 What you need

5.1.1 Required Hardware

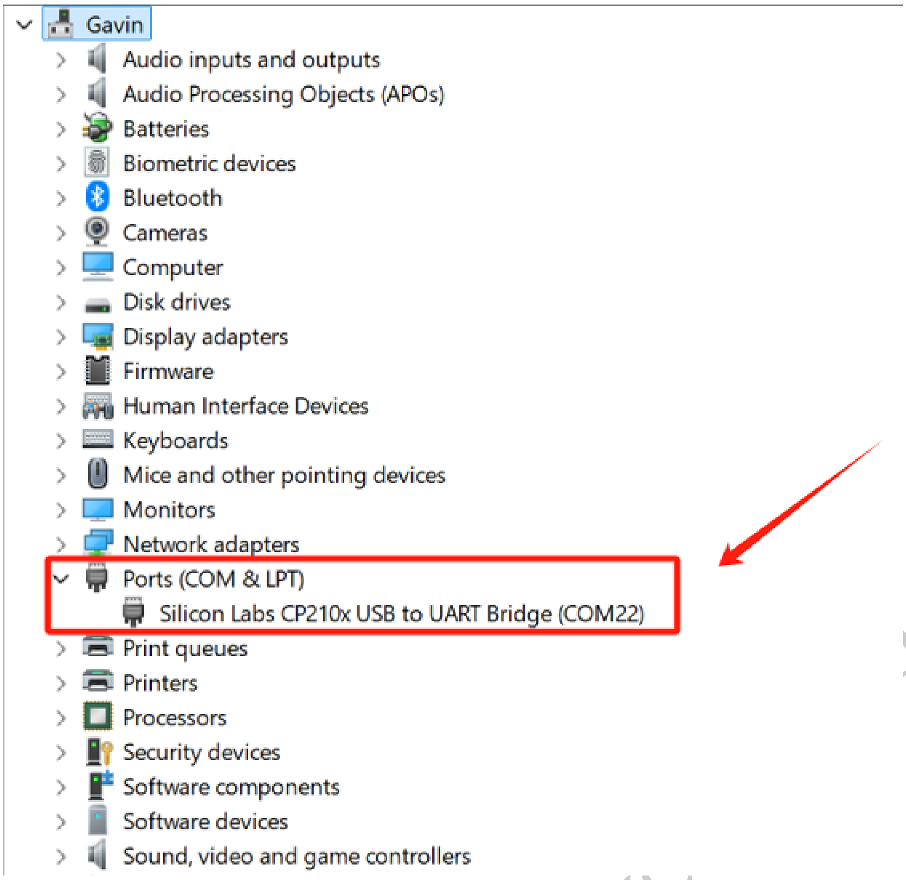
- 1 x FSC-DB006-BT630x Rapid Development Kit：FSC-DB006 USB-to-Serial Rapid Evaluation Board pre-integrated with FSC-BT691 module
- 1 x PC (Windows / Mac)

5.1.2 Software and Setup

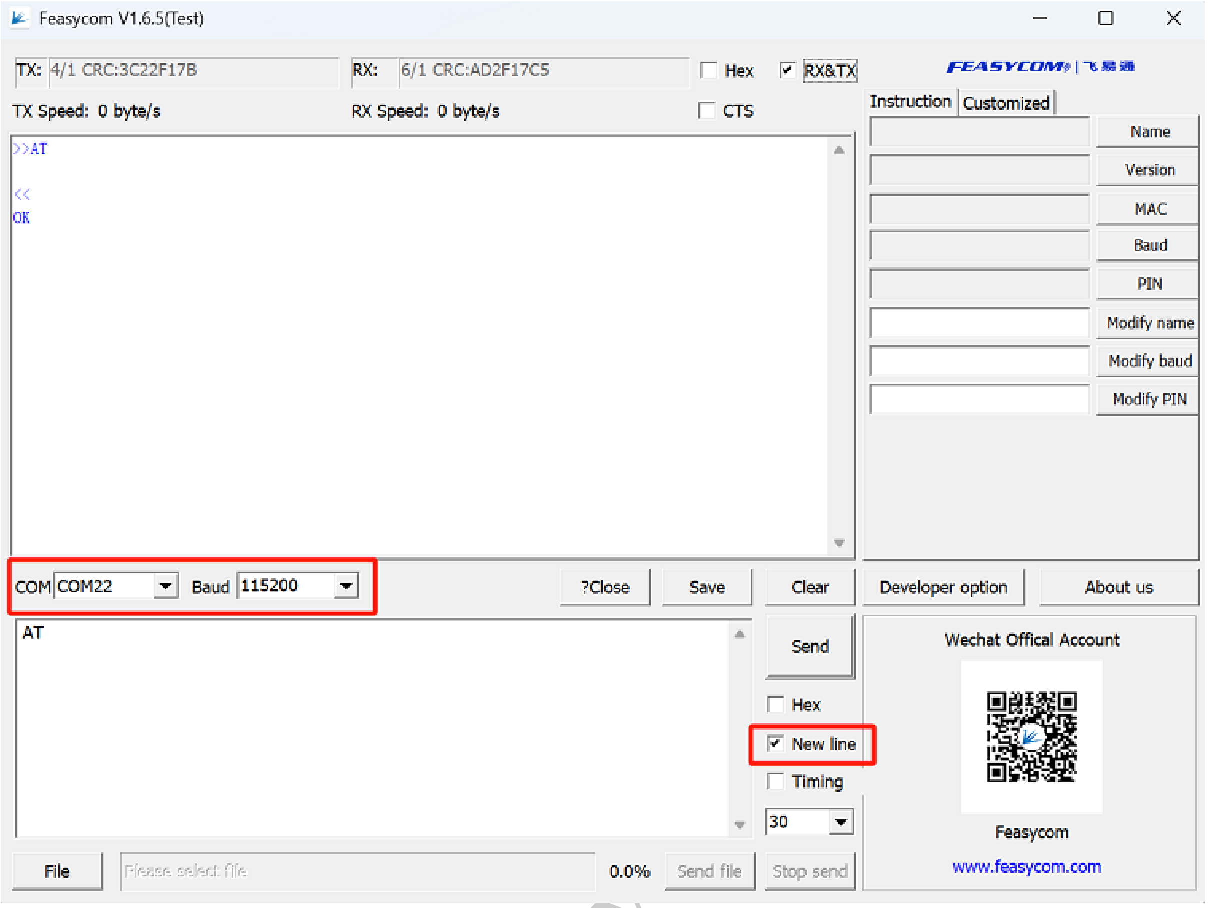
- Feasycom Serial Port Tool：A serial port communication analysis tool based on Windows PC.
- Communication Interface：UART
- UART Configuration：115200/8/N/1

5.2 Hardware Access

1.Connect the FSC-DB006-BT630x Quick Development Kit to the PC via USB. The PC will automatically recognize the serial port and generate a virtual COMx port.



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



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

5.3 Communication Test

The following lists a few basic general AT command test examples.

For more commands, please refer to FSC-BT630x General Data AT Command Set .

5.3.1 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.
```

5.3.2 AT+NAME - Read/Write Bluetooth Name

Example：Read Bluetooth name

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

5.3.3 AT+VER - Read Current Firmware Version

Example：

```
Send:     <<AT+VER\r\n
Response: >>\r\n+VER=1.0.0,FSC-BT630\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-BT630x series Bluetooth BLE data modules have two work modes: **Throughput Mode** and **Command Mode** .

The generic data throughput firmware for the FSC-BT630x series modules defaults to throughput mode. To switch modes, refer to FSC-BT630x General Data AT Command Set and use *AT+TPMODE* command. The differences between the two work 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+LESEND*.

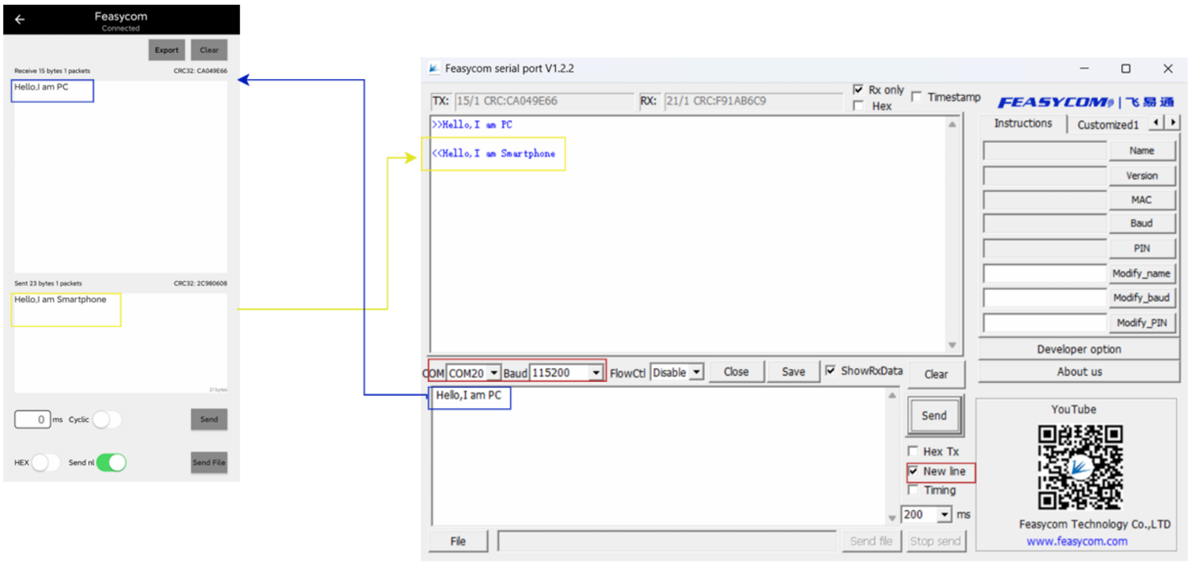
6.1.2 Module to Phone Application

1.Module Side: After being powered on, the module will continuously send broadcast packet data.

2.Mobile Phone Side: Open the [FeasyBlue App] , scan for broadcast packets of nearby Bluetooth BLE devices, locate the target Bluetooth module, and establish a connection.

3.After successful connection, the status pin of the module side will pull up the level, indicating that the connection has been established.

4.After successful connection, in throughput mode, the module side will automatically transmit the data transparently through the air to the remote end (mobile phone side) after receiving serial port data.



Feasycom
Connected

Receive 15 bytes 1 packets CRC32: CA049E66

Hello,I am PC

Sent 23 bytes 1 packets CRC32: 2C980608

Hello,I am Smartphone

21 bytes

0 ms Cyclic

HEX Send nl

Feasycom serial port V1.2.2

TX: 15/1 CRC:CA049E66 RX: 21/1 CRC:F91AB6C9

☑ Rx only □ Timestamp
□ Hex

```
>>Hello,I am PC

<<Hello,I am Smartphone
```

COM COM20 Baud 115200 FlowCtl Disable ☑ ShowRxData

Hello,I am PC

□ Hex Tx
☑ New line
□ Timing
200 ms

Feasycom Technology Co.,LTD
www.feasycom.com

6.1.3 Module to Module Application

This is a demonstration of BLE communication data throughput transmission between FSC-BT630(Module1) and FSC-BT630(Module2) Bluetooth modules, as follows:

1.Scan for nearby BLE devices

FSC-BT630x scans for nearby Bluetooth BLE devices, with the operation as follows:

```
Send: <<AT+SCAN=1                                           //Scan for nearby Bluetooth BLE devices
Response: >>OK
     >>+SCAN={                                             //Scan started
     >>+SCAN=0,0,E0798DB74BD5,-82,9,FSC-WY001
     >>+SCAN=1,0,E0798DB74B93,-88,9,FSC-WY001
     >>+SCAN=2,1,DD0D30500762,-89,13,FSC-BW256B-LE
     >>+SCAN=3,0,E0798DB39B34,-86,9,FSC-WY001
     >>+SCAN=4,0,DC0D30000A09,-52,12,FSC-BT630        //Module2's MAC address and address type 1 are identified.
     >>+SCAN=5,1,FCB7E6A1E72C,-89,10,LE_WI-C100
     >>+SCAN=6,1,DD0D30001885,-80,18,FSC-BT909C-LE-1885
     >>+SCAN=7,0,E0798DB74BA9,-75,9,FSC-WY001
     >>+SCAN=8,0,E0798DB74BD3,-81,9,FSC-WY001
     >>+SCAN=E}                                            //Scan ended
```

2.Send BLE Connection Establishment Request

FSC-BT630(Module1) establishes a BLE protocol connection with FSC-BT630(Module2) via the *AT+LECCONN* command, with the operation as follows:

```
Send: <<AT+LECCONN=DC0D30000A090       //Establish BLE connection request
Response: >>OK
```

⚠ **Warning**

AT+LECCONN=Target Bluetooth MAC address + 1-bit address type. Generally, the address type is “0” or “1”.

How to obtain the address type:

Use AT+SCAN=1 to scan, the second parameter in the returned result is the address type, as shown in the example below:

```
 //The address type is the second parameter, which is "0".

```

(continues on next page)

(continued from previous page)

```
Response: >>+SCAN=4,0,DC0D30000A09,-52,12,FSC-BT630
```

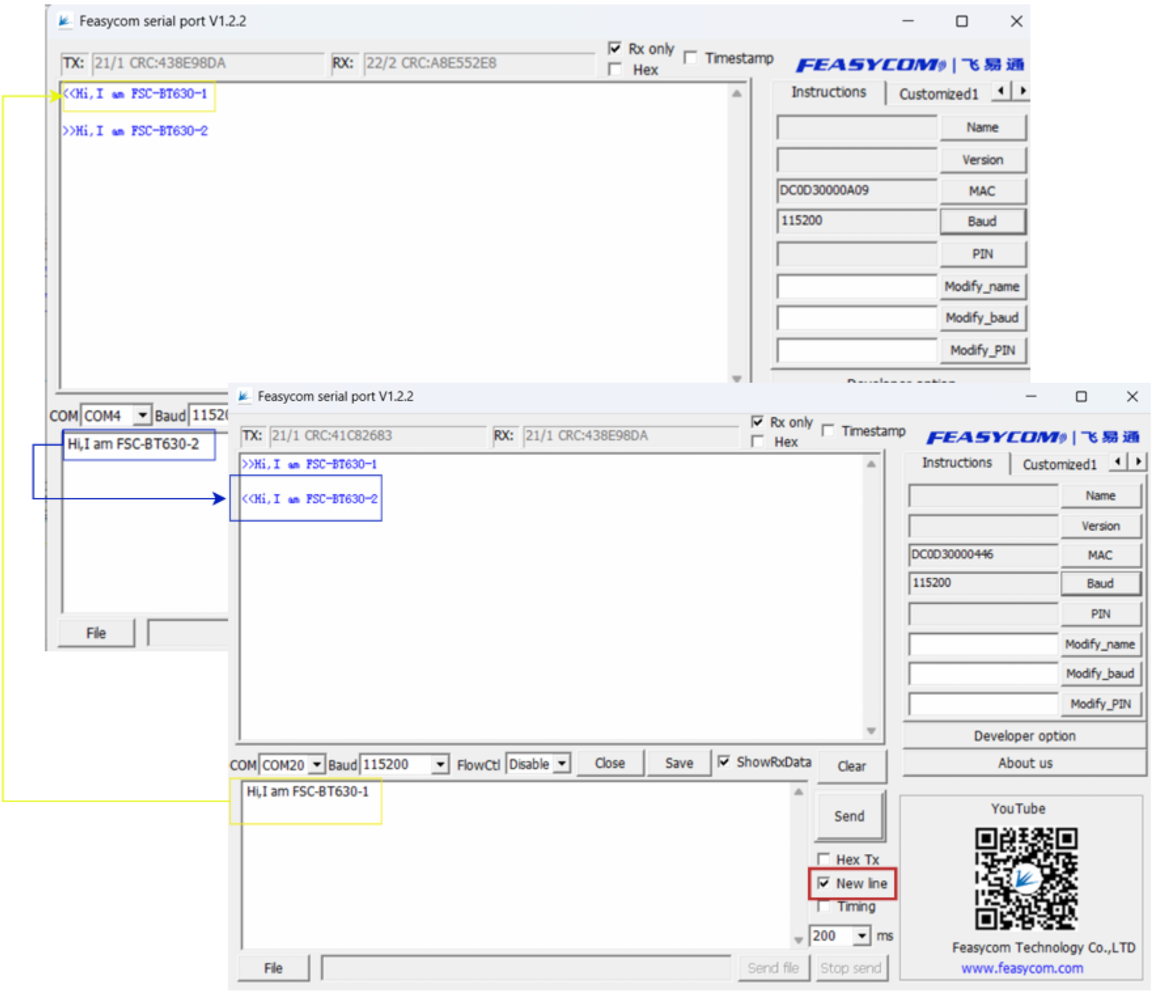
3.BLE Connection Established Successfully

In transparent transmission mode, when the Bluetooth connection is successful, the serial port will not receive an event response indication (in command mode, a connection success event response indication will be received). The current connection status can be determined by the level state of the **Pin18** status indicator pin of FSC-BT630x, as follows:

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

4.Send Data

The transparent transmission mode of the general data transmission firmware is enabled by default. After the BLE connection is successfully established, you can send data directly without sending data via AT commands.



Feasycom serial port V1.2.2

TX: 21/1 CRC:438E98DA RX: 22/2 CRC:A8E552E8 ☑ Rx only ☐ Hex ☐ Timestamp

```
<<Hi,I am FSC-BT630-1
>>Hi,I am FSC-BT630-2
```

DC0D30000A09 MAC

115200 Baud

COM COM4 Baud 1152(

Hi,I am FSC-BT630-2

Feasycom serial port V1.2.2

TX: 21/1 CRC:41C82683 RX: 21/1 CRC:438E98DA ☑ Rx only ☐ Hex ☐ Timestamp

```
>>Hi,I am FSC-BT630-1
<<Hi,I am FSC-BT630-2
```

DC0D30000446 MAC

115200 Baud

COM COM20 Baud 115200 FlowCtl Disable ☑ ShowRxData

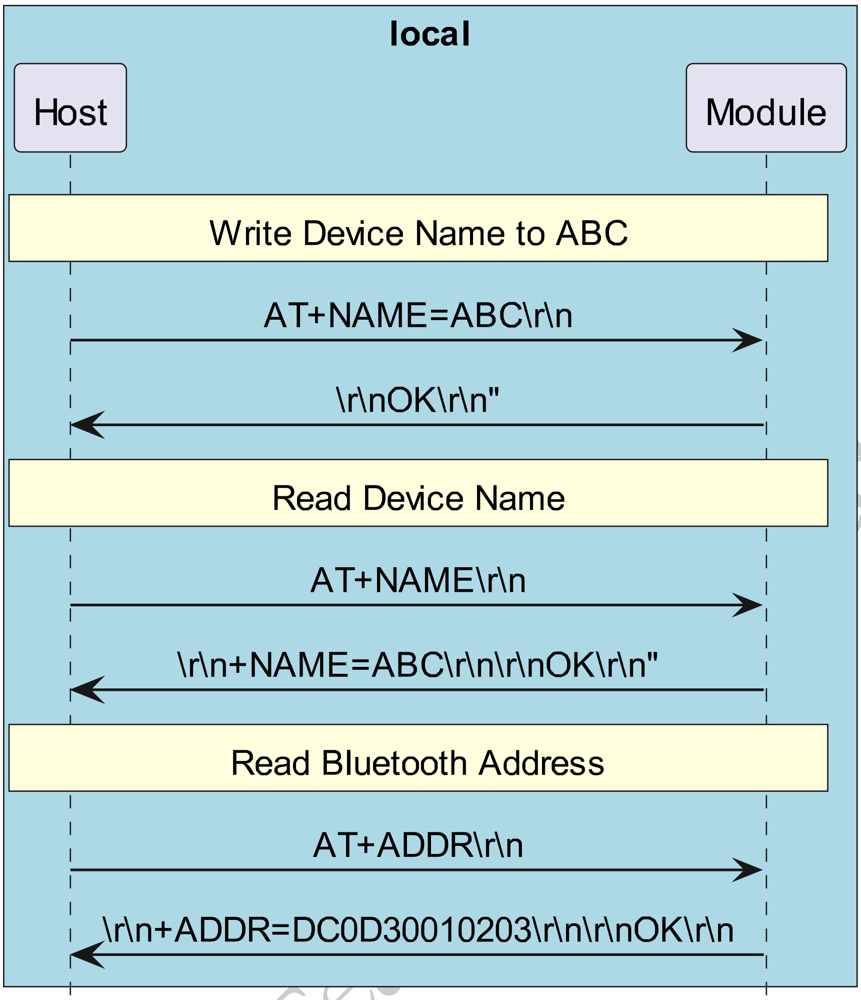
Hi,I am FSC-BT630-1

☐ 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. The following example demonstrates:

1. Write Device Name : ABC
2. Read Device Name
3. Read Bluetooth Address

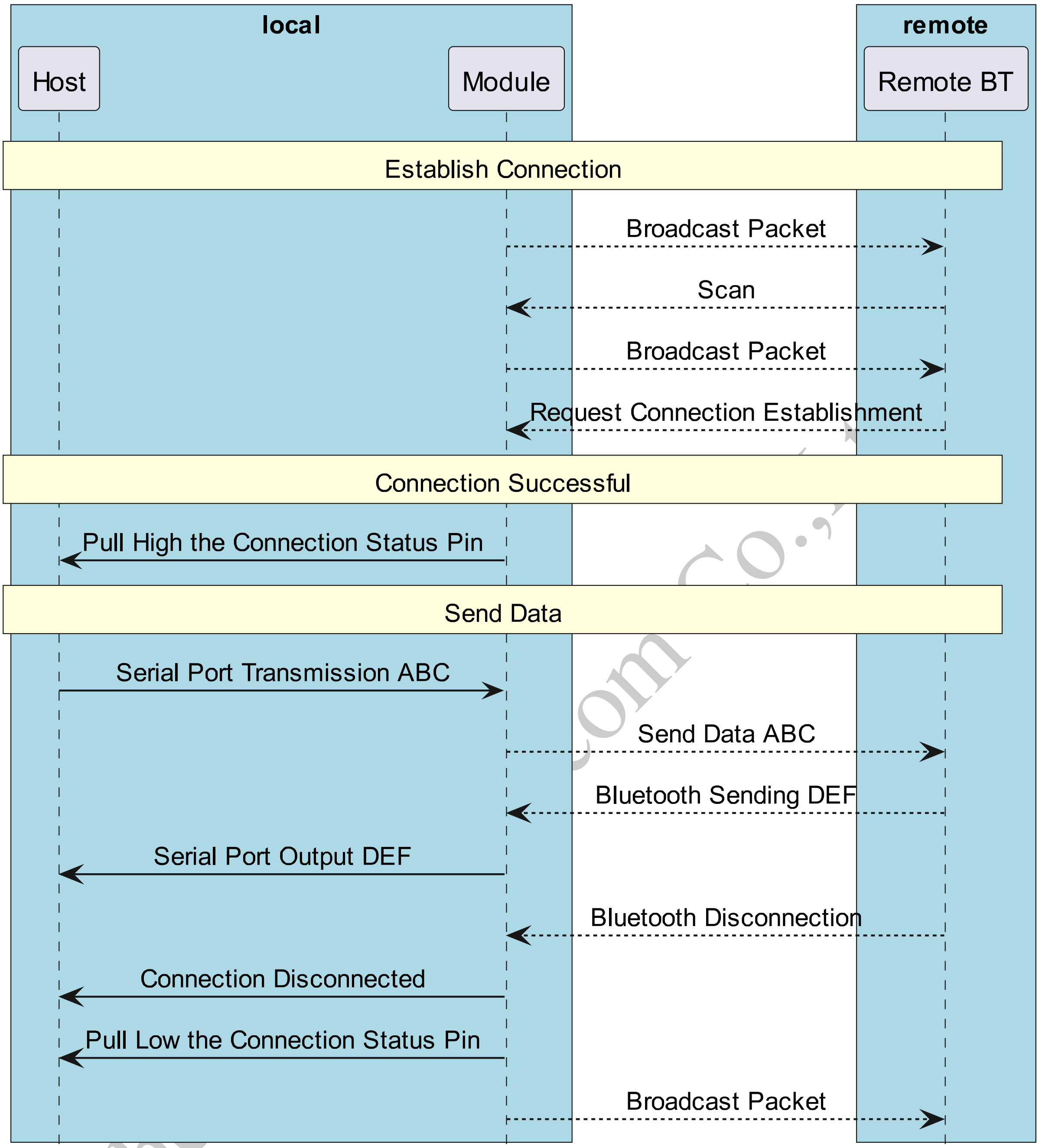


6.3 Data Transmission Flow

1.When the module is powered on, it continuously sends broadcast data outward. A remote Bluetooth device (e.g., a mobile phone) can obtain the broadcast packet by searching and initiate a connection request to the module.

2.After the connection is successfully established, the module will pull high the connection status pin to notify the host that the Bluetooth connection has been successfully established.

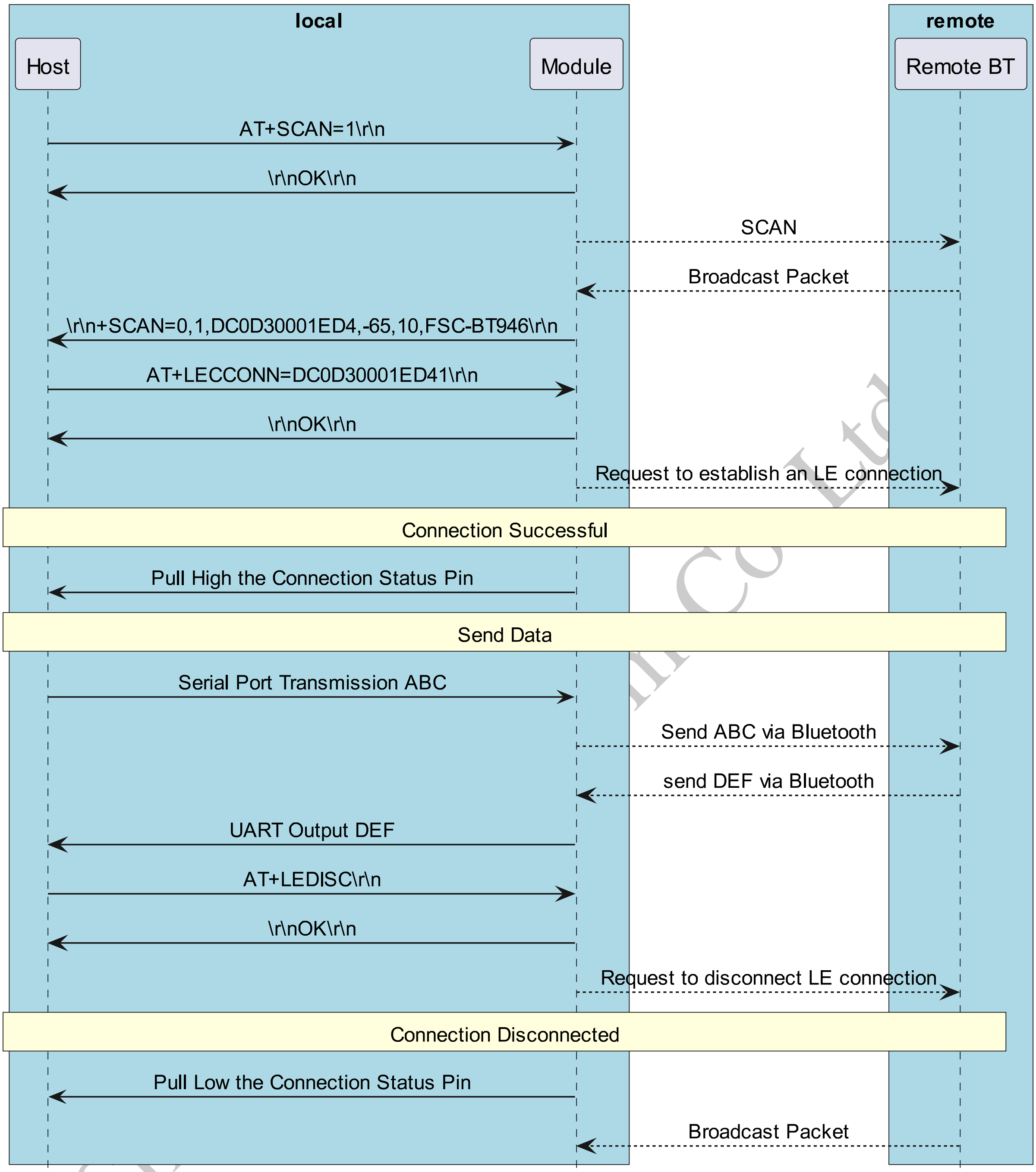
3.The host can send data to the remote Bluetooth device via the Bluetooth module, and the remote Bluetooth device can also send data to the host.



6.4 Module Acts as Master to Connect to Remote Device

The module can act as master device to connect to remote slave devices.

The host can send AT commands to control the module to perform scanning, connection, and disconnection operations. The following shows the process of connecting to other devices:


local
remote
Host
Module
Remote BT
AT+SCAN=1\r\n
\r\nOK\r\n
SCAN
Broadcast Packet
\r\n+SCAN=0,1,DC0D30001ED4,-65,10,FSC-BT946\r\n
AT+LECCONN=DC0D30001ED41\r\n
\r\nOK\r\n
Request to establish an LE connection
Connection Successful
Pull High the Connection Status Pin
Send Data
Serial Port Transmission ABC
Send ABC via Bluetooth
send DEF via Bluetooth
UART Output DEF
AT+LEDISC\r\n
\r\nOK\r\n
Request to disconnect LE connection
Connection Disconnected
Pull Low the Connection Status Pin
Broadcast Packet

Chapter 7

Firmware Upgrade

[中文版]

7.1 OTA Upgrade

7.1.1 Tool

- nRF connect App

7.1.2 User Guide

1.Save the **firmware upgrade resource package** (provided by Feasycom) to the **mobile phone' s** local storage.

2.Run **nRF Connect**, scan and connect to the device that needs to be upgraded.

3.After successful connection, click the **DFU** icon in the upper right corner of the page.

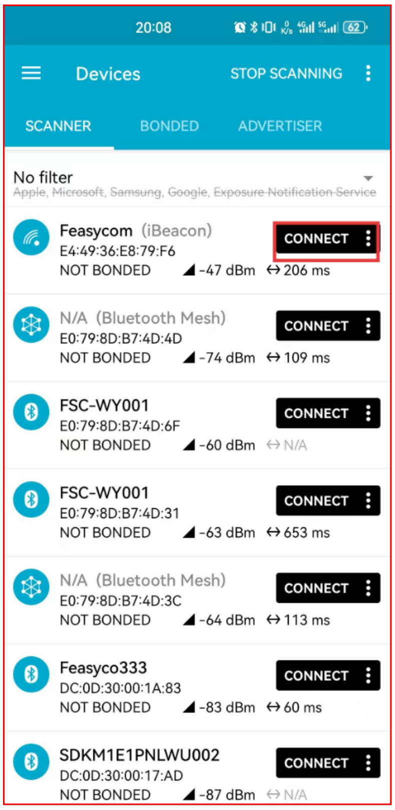
4.In the pop-up **Select file type** window, select the **Distribution packet (ZIP)** file type, and confirm with **OK** .

5.**Select and import** the firmware upgrade resource package saved in the mobile phone' s local storage;

6.After the firmware upgrade resource package is imported successfully, the APP will first disconnect, then wait for 3 seconds, and the APP will automatically reconnect.

7.When the status shows **Uploading** and the **upgrade progress bar**, the upgrade has started, just wait for the upgrade to complete.

7.1.3 OTA Upgrade Show



20:08

Devices STOP SCANNING

SCANNER BONDED ADVERTISER

No filter
Apple, Microsoft, Samsung, Google, Exposure Notification Service

Feasycom (iBeacon) CONNECT
E4:49:36:E8:79:F6
NOT BONDED -47 dBm ↔ 206 ms

N/A (Bluetooth Mesh) CONNECT
E0:79:8D:B7:4D:4D
NOT BONDED -74 dBm ↔ 109 ms

FSC-WY001 CONNECT
E0:79:8D:B7:4D:6F
NOT BONDED -60 dBm ↔ N/A

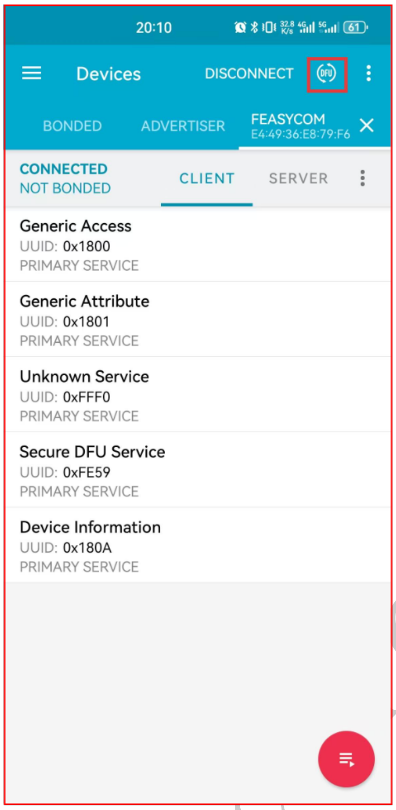
FSC-WY001 CONNECT
E0:79:8D:B7:4D:31
NOT BONDED -63 dBm ↔ 653 ms

N/A (Bluetooth Mesh) CONNECT
E0:79:8D:B7:4D:3C
NOT BONDED -64 dBm ↔ 113 ms

Feasyco333 CONNECT
DC:0D:30:00:1A:83
NOT BONDED -83 dBm ↔ 60 ms

SDKM1E1PNLWU002 CONNECT
DC:0D:30:00:17:AD
NOT BONDED -87 dBm ↔ N/A

(1)



20:10

Devices DISCONNECT

BONDED ADVERTISER FEASYCOM E4:49:36:E8:79:F6

CONNECTED
NOT BONDED
CLIENT SERVER

Generic Access
UUID: 0x1800
PRIMARY SERVICE

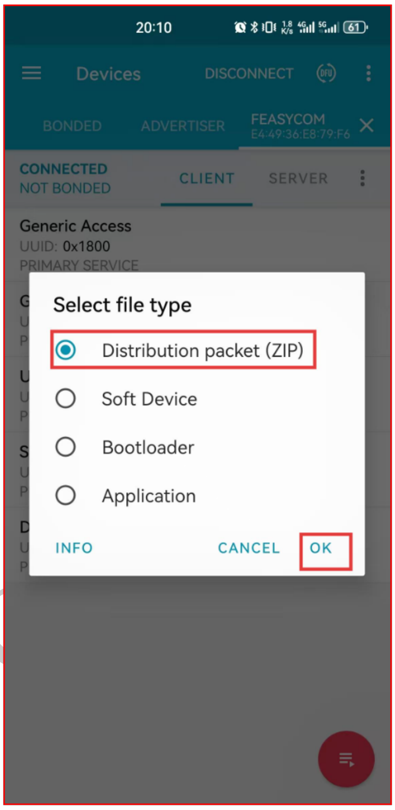
Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Unknown Service
UUID: 0xFFF0
PRIMARY SERVICE

Secure DFU Service
UUID: 0xFE59
PRIMARY SERVICE

Device Information
UUID: 0x180A
PRIMARY SERVICE

(2)



20:10

Devices DISCONNECT

BONDED ADVERTISER FEASYCOM E4:49:36:E8:79:F6

CONNECTED
NOT BONDED
CLIENT SERVER

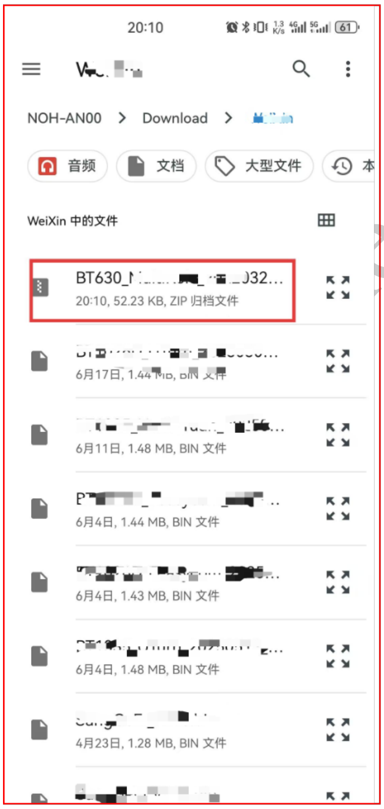
Generic Access
UUID: 0x1800
PRIMARY SERVICE

Select file type

- ● Distribution packet (ZIP)
- ○ Soft Device
- ○ Bootloader
- ○ Application

INFO CANCEL OK

(3)



20:10

NOH-AN00 > Download > [redacted]

音频 文档 大型文件 本

WeiXin 中的文件

BT630_[redacted]032…
20:10, 52.23 KB, ZIP 归档文件

[redacted]
6月17日, 1.44 MB, BIN 文件

[redacted]
6月11日, 1.48 MB, BIN 文件

[redacted]
6月4日, 1.44 MB, BIN 文件

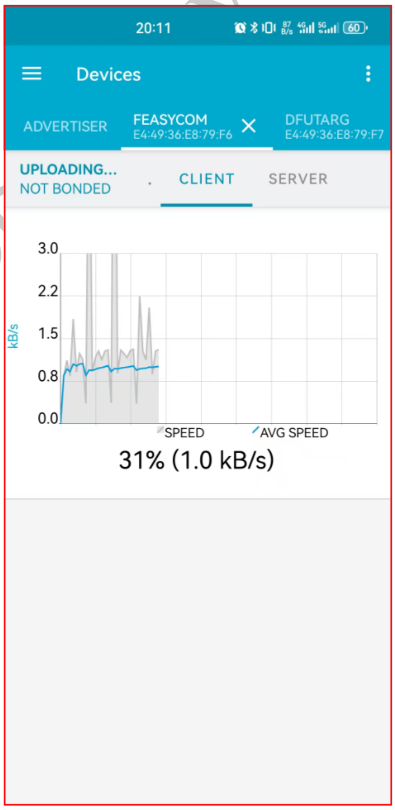
[redacted]
6月4日, 1.43 MB, BIN 文件

[redacted]
6月4日, 1.48 MB, BIN 文件

[redacted]
4月23日, 1.28 MB, BIN 文件

[redacted]

(4)



20:11

Devices

ADVERTISER FEASYCOM E4:49:36:E8:79:F6 DFUTARG E4:49:36:E8:79:F7

UPLOADING...
NOT BONDED
CLIENT SERVER

31% (1.0 kB/s)

(5)

Chapter 8

FAQs

[中文版]

8.1 Why is an APP required on a mobile phone for Bluetooth connection and communication?

```
The phone's native Bluetooth function only supports general scenarios, such as audio transmission and file transfer. Some Bluetooth peripheral devices can be connected via the phone's built-in settings program, such as Bluetooth speakers, Bluetooth headsets, Bluetooth keyboards, Bluetooth mice, etc. When a Bluetooth peripheral device cannot be connected by the phone's native settings program (for example, the Bluetooth module only supports SPP/GATT protocols), to connect such modules, it is generally necessary to install a specific mobile application on the phone, such as the FeasyBlue application
```

8.2 How to obtain Bluetooth MAC address on iOS device?

For security reasons, the iOS system converts the Bluetooth MAC address into a UUID at the underlying layer and sends it to upper-layer applications. Therefore, the APP cannot obtain the device’ s MAC address.

FSC-BT630x Bluetooth module will place the MAC address in the broadcast by default, and the APP can obtain the MAC address from the broadcast packet through the following methods.

```
- (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 The following parameter is 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