



FEASYCOM

FSC-BT3721V User Guide

Release 3.5.1



Table of contents

1	Hardware Design	2
1.1	Module Pin Diagram	2
1.2	Pin Description	3
1.3	Hardware Design Note	3
2	Functional Description	5
2.1	Default Configuration	5
2.2	GPIO Indication	5
2.2.1	BT Connection Status	5
2.3	Working Mode	5
2.3.1	Throughput Mode	6
2.3.2	Command Mode	6
2.4	GATT Service	6
2.5	Operating Timing	6
2.6	Data Rate (Typical)	7
2.7	Low-Power Strategy	7
2.7.1	Low-Power Mode Setting	7
2.7.2	Low-Power Operation Strategy	7
2.7.3	Low-Power Data (Typical)	8
3	Data Communication Principles	9
3.1	Communication Principle	9
3.2	MCU-to-Module Communication	10
3.3	Module-to-Module Communication	11
3.4	Module-to-Phone Communication	11
3.4.1	Why is an APP required on a mobile phone for Bluetooth connection and communication?	11
3.4.2	Communication Application Diagram	12
4	Quick Development Kit	13

4.1	Datasheet	13
4.2	AT Command Set	13
4.3	Serial Port Tool	13
4.4	App&SDK	13
4.5	Firmware Upgrade	13
4.5.1	OTA Upgrade	14
5	Quick Start	15
5.1	What you need	15
5.1.1	Required Hardware	15
5.1.2	Software and Setup	15
5.2	Hardware Access	15
5.2.1	UART Connection	15
5.3	Communication Test	16
5.3.1	Module-to-PC	16
5.3.2	UART Communication Test	18
6	Development Examples	20
6.1	Data Throughput Mode Application	20
6.1.1	What is Throughput Mode?	20
6.1.2	Module to Phone Application	21
6.1.3	Module to Module Application	21
6.2	Read/Write Module Default Parameters	23
6.3	Data Transmission Flow	24
6.4	Module Acts as Master to Connect to Remote Device	25
7	Firmware Upgrade	27
7.1	OTA Upgrade	27
7.1.1	OTA Upgrade Tool	27
7.1.2	Firmware Upgrade File	27
7.1.3	OTA Upgrade Guide	27
8	FAQs	29
8.1	Why is an APP required on a mobile phone for Bluetooth connection and communication?	29
8.2	How to obtain Bluetooth MAC address on iOS device?	29
9	Contact Information	32

Shenzhen Feasycom Co., Ltd.

[中文版]

This guide is applicable to:

FSC-BT3721V Bluetooth BLE Data Application Module

This guide consists of the following parts:

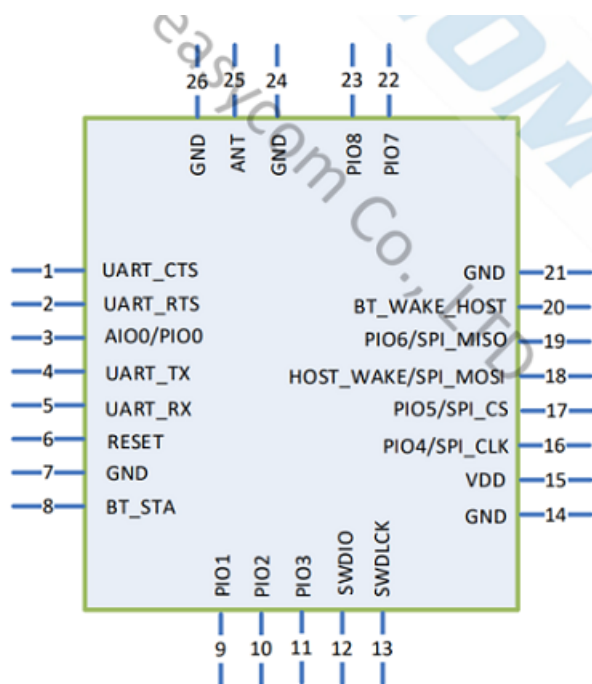
Shenzhen Feasycom Co., Ltd.

Chapter 1

Hardware Design

[中文版]

1.1 Module Pin Diagram



1.2 Pin Description

Pin	Pin Name	Type	Pin Descriptions
1	UART_CTS	I/O	Serial Port Flow Control Pin
2	UART_RTS	I/O	Serial Port Flow Control Pin
3	AOIO/PIO0	I/O	ADC
4	UART_TX	O	Serial Port Data Pin
5	UART_RX	I	Serial Port Data Pin
6	RESET	I	Low-Level Reset
7	GND	GND	GND
8	STATUS	O	Low level : Bluetooth is disconnected.High level : Bluetooth is connected.
12	SWDIO	I/O	Programming Pin
13	SWDCLK	I	Programming Pin
15	VDD	Power	Power supply: 3.3V.It is recommended to use an LDO for power supply.
16	SPI_CLK	I/O	SPI_CLK PIN
17	SPI_CS	I/O	SPI_CS PIN
18	HOST_WAK	O	HOST Controlled Pin: High level wakes up the HOST.Low level puts the HOST into sleepmode.
19	SPI_MISO	O	SPI_MISO PIN
20	BT_WAKE_1	O	High level wakes up the BT.Low level puts the BT into sleep mode.
25	ANT	ANT	Changing the 0-ohm resistor near the antenna allows for the connection of an external Bluetooth antenna.

1.3 Hardware Design Note

- The module can be used by connecting VDD/GND/BT_STA/UART_RX/UART_TX.
- If the MCU needs to obtain the connection status of the Bluetooth module, it is necessary to connect the BT_STA pin.
- If there is a low-power consumption requirement, please connect the BT_WAKE_HOST and HOST_WAKE_BT pins.
- After completing the schematic diagram, please send it to Feasycom for review to avoid

the Bluetooth range failing to reach the optimal performance.

Shenzhen Feasycom Co., Ltd.

Chapter 2

Functional Description

[中文版]

2.1 Default Configuration

Based on the general BLE data transmission Application Firmware:

Name	FSC-BT3721V
Pin Code	000000
Secure Simple Pairing Mode	Off
UART Baudrate	115200/8/N/1

2.2 GPIO Indication

2.2.1 BT Connection Status

PIN	Status	Description
PIN 8	Low Level	Bluetooth Disconnected
PIN 8	High Level	Bluetooth Connected

2.3 Working Mode

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+LESEND.

2.4 GATT Service

Type	UUID	Operation	Description
Service	0xFFFF0		Throughput transmission service
Write	0xFFFF2	Write, Write Without Response	APP to Module
Notify	0xFFFF1	Notify	Module to APP

2.5 Operating Timing

Process	Operation	Timing	Key Signal/Response
Power On	Internal initialization completed	≥250ms	/
AT Configuration	Send AT command (e.g., AT+NAME=FSC-BT3721V)	Interval between commands ≥100ms	OK or ERROR

2.6 Data Rate (Typical)

BAUD	Data Packet	Transmission Interval	Connection Interval	Transmission Method	Data Rate
460800	244	5	15ms	Notify	46000 Byte/s

2.7 Low-Power Strategy

2.7.1 Low-Power Mode Setting

The low-power function can be enabled or disabled via the command `AT+LPM{=Param}`.

2.7.2 Low-Power Operation Strategy

- **AT+LPM=0**
 - Disable Low-Power Mode
- **AT+LPM=1**
 - The BT serial port will automatically enter low-power mode if no data is received for more than 5 seconds.
 - The first frame of data from the BT serial port wakes up the Bluetooth.
 - If the connection is successful, it will not enter low-power mode.
- **AT+LPM=2**
 - PIN20(HOST_WAKE_BT) Low level: The module enters sleep mode.
 - PIN20(HOST_WAKE_BT) High level: The module exits sleep mode.
 - Module sleep PIN18(BT_WAKE_HOST) Low level.
 - Module wake up PIN18(BT_WAKE_HOST) High level.

2.7.3 Low-Power Data (Typical)

Advertising Interval	Power Consumption
100ms	100uA
152ms	60.92uA
200ms	51.19uA
300ms	34.9uA
400ms	28.39uA
500ms	22.16uA
600ms	19.34uA
700ms	16.67uA
800ms	14.49uA
900ms	13.32uA
1000ms	11.79uA
2000ms	7.65uA

Chapter 3

Data Communication Principles

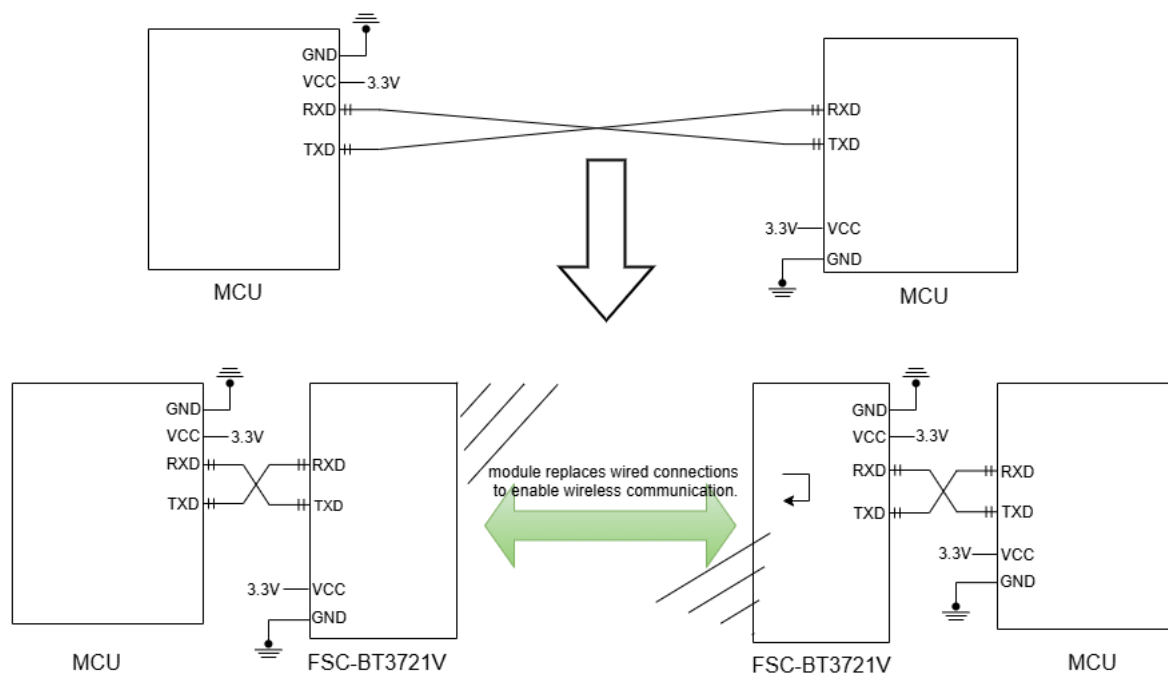
[中文版]

3.1 Communication Principle

FSC-BT3721V Bluetooth BLE data transmission modules enable wireless communication between devices based on the BLE (Bluetooth Low Energy) protocol.

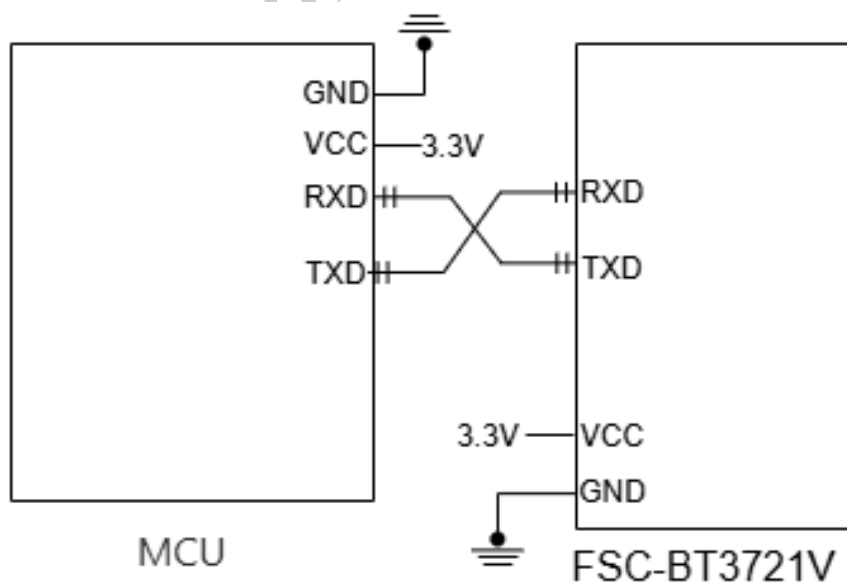
- **BLE** : It adopts an event-driven low-power architecture and defines a “Service-Characteristic” model through the GATT protocol to realize intermittent small-data interaction (e.g., sensor data), making it suitable for IoT (Internet of Things) devices.

The module sends AT commands or transparent transmission data to the host device (mobile phone/MCU) via UART to complete connection establishment, data exchange, and status management.



As shown in the diagram, the Bluetooth module is used to replace the physical wires in full-duplex communication. A device such as a microcontroller unit (MCU) (on the left) sends data to the left-side Bluetooth module via its TXD pin. After the RXD pin of the left-side Bluetooth module receives the serial data, it automatically transmits the data via radio waves to the remote Bluetooth module (on the right). The right-side remote Bluetooth module then receives the over-the-air data and sends it to the local device (e.g., an MCU) on the right via its TXD pin.

3.2 MCU-to-Module Communication

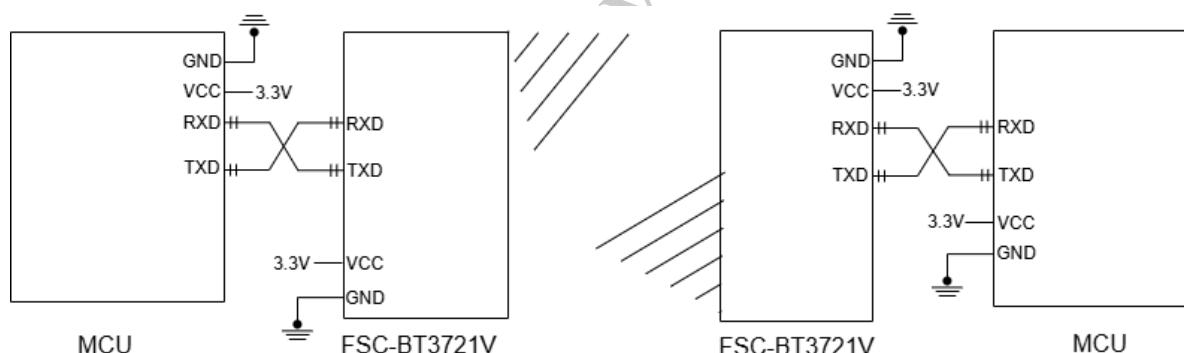


This diagram illustrates a connection schematic between a main control MCU (Microcontroller Unit) and an FSC-BT3721V Bluetooth module. Through cross-connected serial ports, command interaction between the main controller and the Bluetooth module is achieved, enabling wireless communication functionality. This setup is suitable for scenarios such as IoT (Internet of Things) devices and remote control.

1. **Serial Communication Interface:** The transmitting end of the main MCU (MCU_TX) is cross-connected to the receiving end of the Bluetooth module (UART_RX), and similarly, the receiving end of the MCU (MCU_RX) is connected to the transmitting end of the Bluetooth module (UART_TX), forming a bidirectional 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-BT3721V Bluetooth modules can establish a Bluetooth connection automatically upon power-on.



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

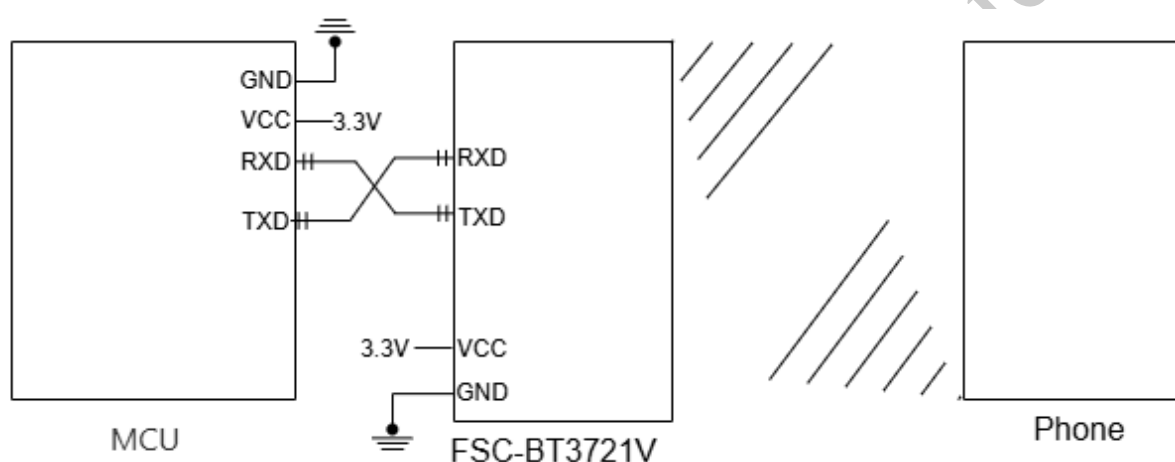
3.4 Module-to-Phone Communication

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 Blue-

tooth peripheral devices can be connected via the built-in settings program of the mobile phone, including 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 the SPP/GATT protocol), a specific mobile application (such as the Feasy-Blue app) usually needs to be installed on the mobile phone to connect to such a module.

3.4.2 Communication Application Diagram



Bluetooth Module (FSC-BT3721V) : Upon power-on, it will continuously send out broadcast data.

Mobile Phone side : The broadcast packets can be obtained through searching, and a connection request can be initiated to the module side (FSC-BT3721V). After a successful connection, the Bluetooth module (FSC-BT3721V) will pull up the connection status pin and respond to the indication command (effective in command mode) to notify the host side that the Bluetooth connection is successful.

Host: 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-BT3721V Datasheet

4.2 AT Command Set

- FSC-BT3721V General Data AT Command Set

4.3 Serial Port Tool

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

4.4 App&SDK

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

4.5 Firmware Upgrade

4.5.1 OTA Upgrade

- Tool: FeasyOTA App (based on Android)
- User Guide: Please refer to FSC-BT3721V - OTA Upgrade .

Shenzhen Feasycom Co.,Ltd.

Chapter 5

Quick Start

[中文版]

5.1 What you need

5.1.1 Required Hardware

- 1 x FSC-BT3721V Module
- 1 x PC (Windows)

5.1.2 Software and Setup

- **Feasycom Serial Port Tool** : A serial communication analysis tool based on Windows PC.
- **Communication Interface:** UART
- **Serial Configuration:** 115200/8/N/1 (General default for Feasycom firmware)

5.2 Hardware Access

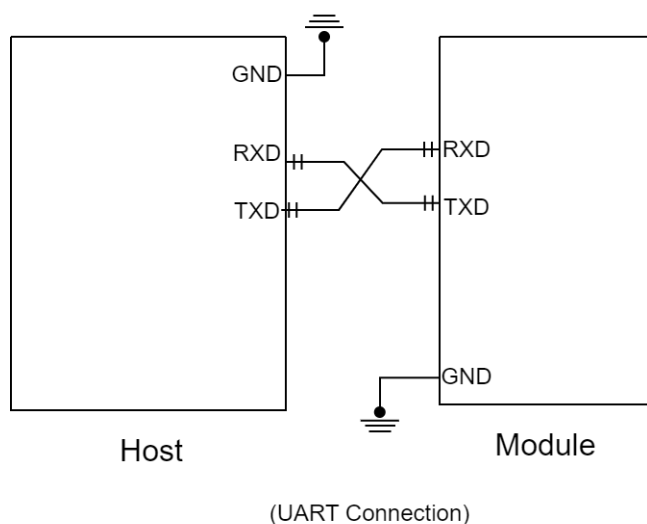
5.2.1 UART Connection

FSC-BT3721 provides one channels of Universal Asynchronous Receiver/Transmitters(UART)(Full-duplex asynchronous communications). The UART Controller performs a serial-to-parallel conversion on data received from the peripheral and a parallel-to-serial conversion on data transmitted from the CPU.

This is a standard UART interface for communicating with other serial devices. The UART interface provides a simple mechanism for communicating with other serial devices using the RS232 protocol.

When the module is connected to another digital device, UART_RX and UART_TX transfer data between the two devices.

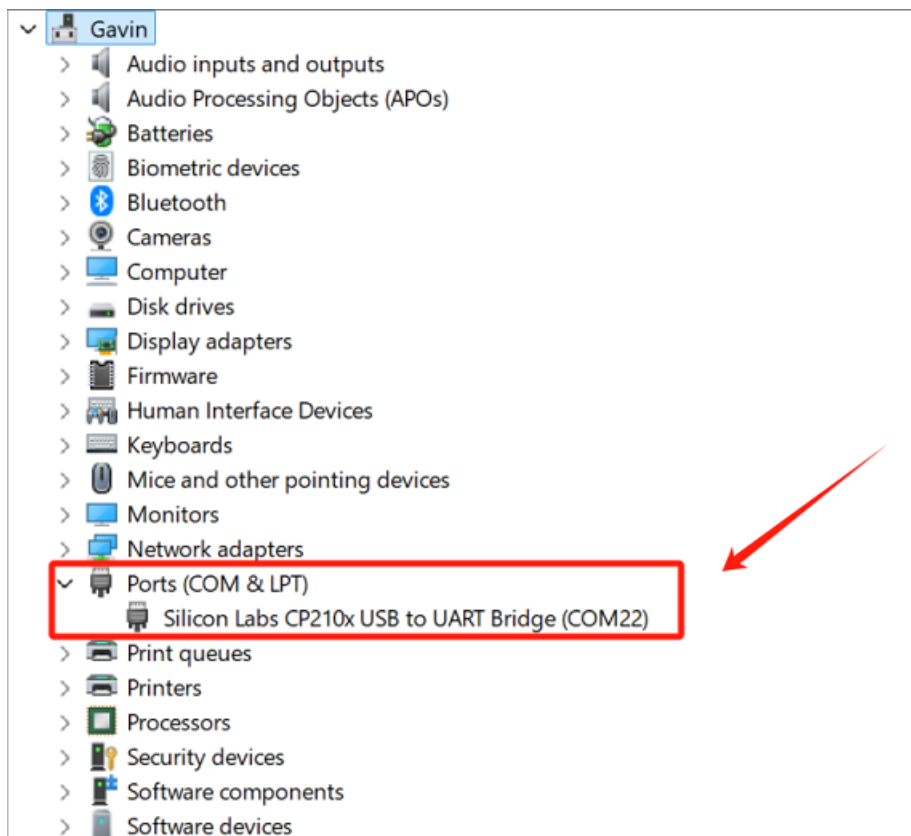
When connecting the module to a host, please make sure to follow:



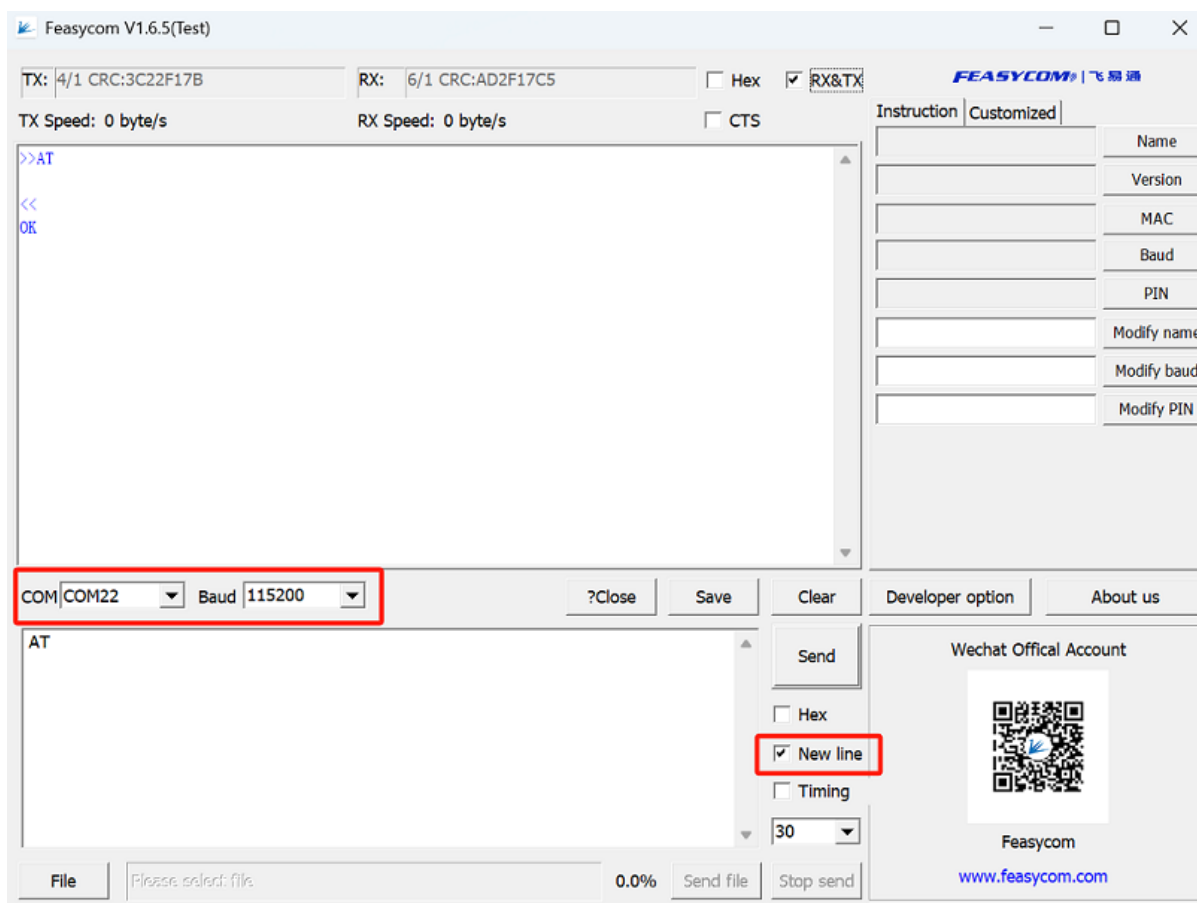
5.3 Communication Test

5.3.1 Module-to-PC

1. On the premise of ensuring compliance with the aforementioned **UART connection** requirements, set up the hardware testing environment with the PC. Upon completion, the PC will automatically recognize the device and virtualize a COM port.



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



5.3.2 UART Communication Test

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

For more commands, please refer to [FSC-BT3721V General Data AT Command Set](#).

AT - UART Communication Test

Com- mand	AT\r\n
Response	\r\nOK\r\n
Descrip- tion	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.
```

AT+NAME - Read/Write Bluetooth Name

Example: Read Bluetooth name

```
send:      <<AT+NAME\r\n
response: >>\r\n+NAME=FSC-BT3721V\r\n    //Default, please refer to
↳the actual reading result
response: >>\r\nOK\r\n
```

AT+VER - Read Current Firmware Version

Example:

```
Send:      <<AT+VER\r\n
Response: >>\r\n+VER=6.2.6,FSC-BT3721V\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-BT3721V Bluetooth BLE data module has two work modes: **Throughput Mode** and **Command Mode**.

The generic data throughput firmware for the FSC-BT3721V modules defaults to throughput mode. To switch modes, refer to [FSC-BT3721V 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 :**

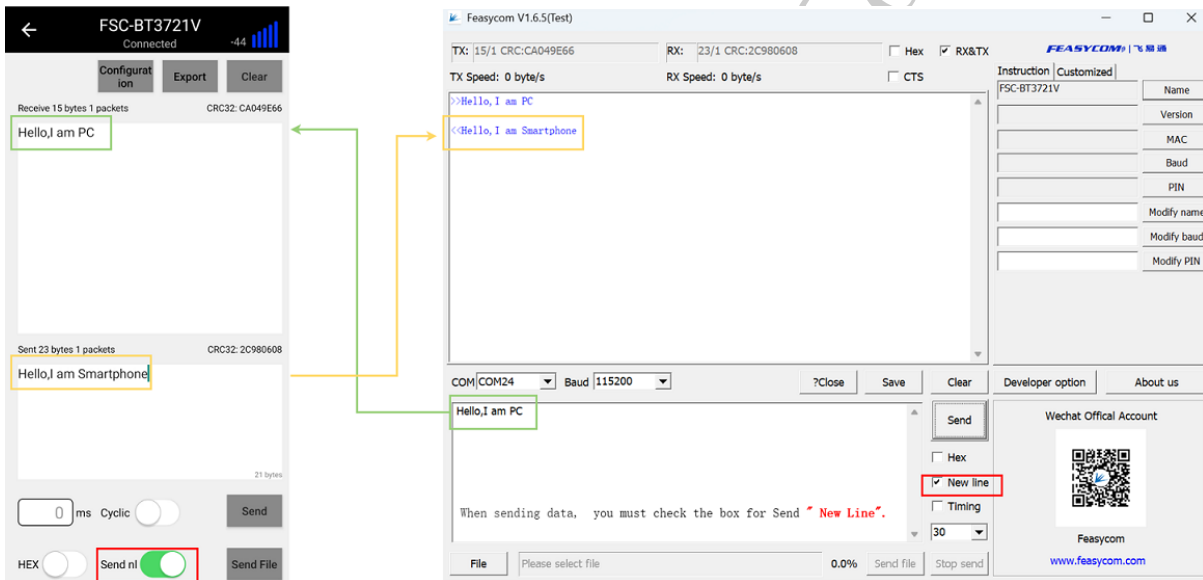
- **Bluetooth Not Connected :** Data received via UART is parsed as AT commands.

- **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 power-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, find the target Bluetooth module, and establish a connection;
- 3.After successful connection, the status pin of the module will pull up the level, indicating that the connection has been established;
- 4.After successful connection, in throughput mode, the module will automatically transmit the serial port data it receives to the remote end (mobile phone side) via air.



6.1.3 Module to Module Application

Demonstration of BLE communication data throughput transmission between FSC-BT3721V and FSC-BT986-LE Bluetooth modules, as follows:

- 1.Scan for nearby BLE devices

FSC-BT3721V Scan for nearby Bluetooth BLE devices.

```

1 Send: <<AT+SCAN=1 //Scan for nearby Bluetooth BLE
  → devices
2 Response: >>OK
    
```

(continues on next page)

(continued from previous page)

```

3      >>+SCAN={                               //Scan Started
4      >>+SCAN=0,1,DD0D3053CBED,-66,19,iMin TF2-34 LE CBED
5      >>+SCAN=1,0,DC0D3060D2E9,-68,12,FSC-BW236-LE
6      >>+SCAN=2,0,DC0D30001106,-76,12,FSC-BT1106RC
7      >>+SCAN=3,0,DC0D300017A9,-87,11,FSC-BT3721V
8      >>+SCAN=4,1,DD0D3000174C,-13,12,FSC-BT986-LE //The target
↳module's MAC address and address type 1 are identified.
9      >>+SCAN=}                               //Scan Completed

```

2.Establish BLE connection request

FSC-BT3721V establish BLE protocol connection with FSC-BT986-LE via the AT+LECONN command. Operation is as follows:

```

1  Send:      <<AT+LECONN=DD0D3000174C1       //Establish BLE connection
↳request
2  Response: >>OK

```

Warning

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

How to obtain the address type:

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

```

1 //The address type is the second parameter, which is "1".
2
3 Response: >>+SCAN=4,1,DD0D3000174C,-13,12,FSC-BT986-LE

```

3.BLE Connection Established Successfully

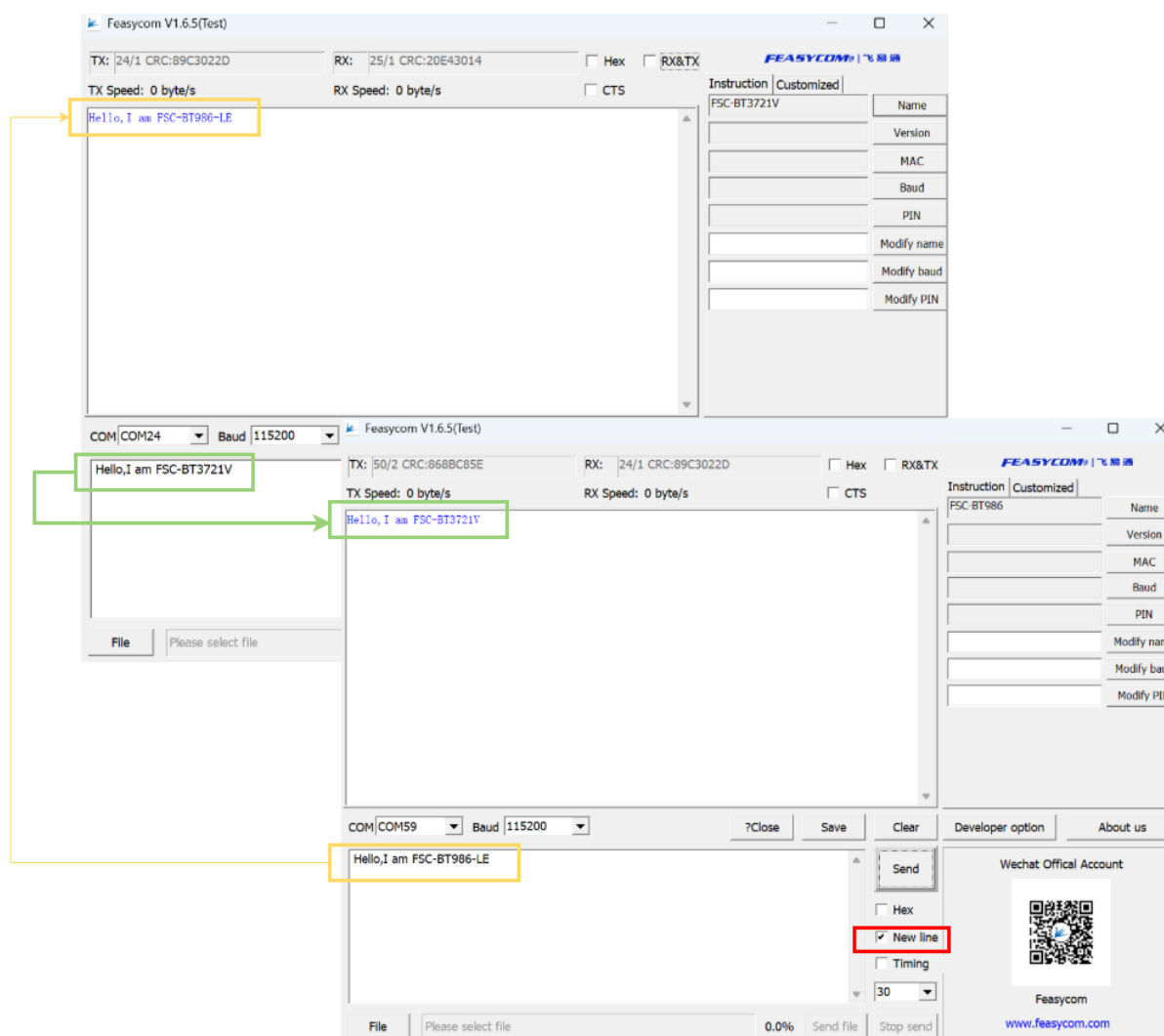
In throughput mode, after the Bluetooth connection is successfully established, the serial port cannot receive event response indicators. The current connection status can be determined by the level state of Pin8 (status indicator pin) of FSC-BT3721V, as detailed below:

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

The throughput mode of the general data transmission firmware is enabled by default. After BLE connection is successfully established, data can be sent directly without the need to send data via AT commands.

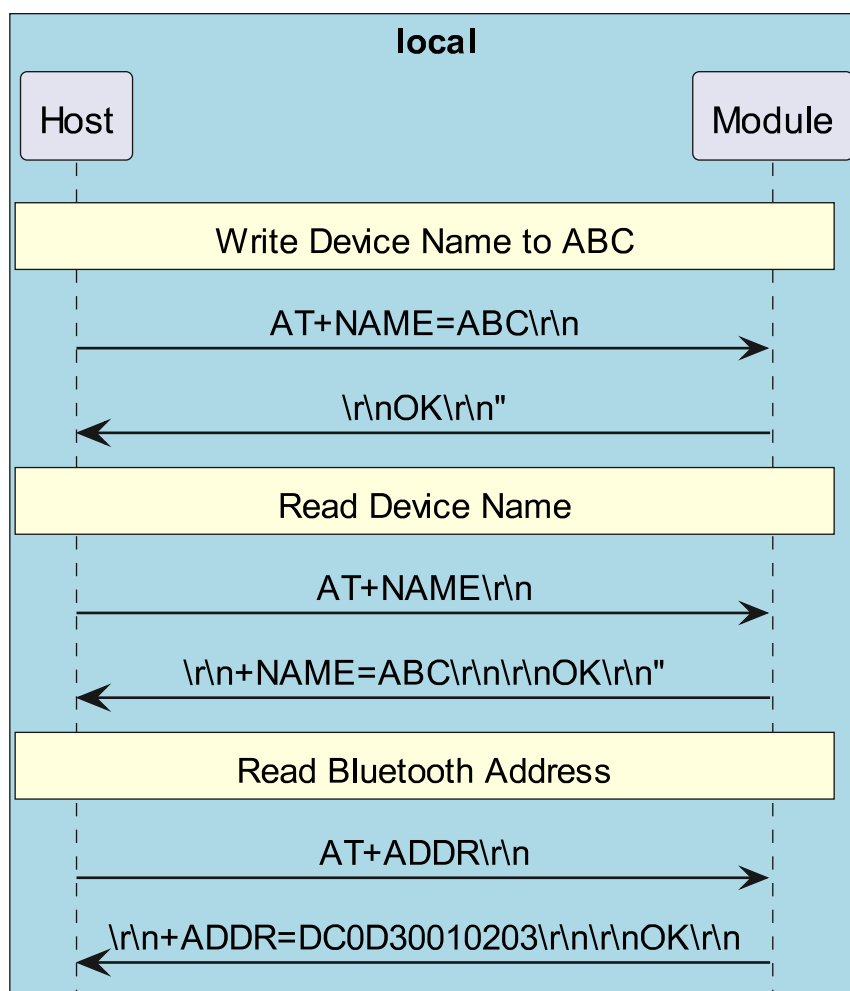


6.2 Read/Write Module Default Parameters

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:

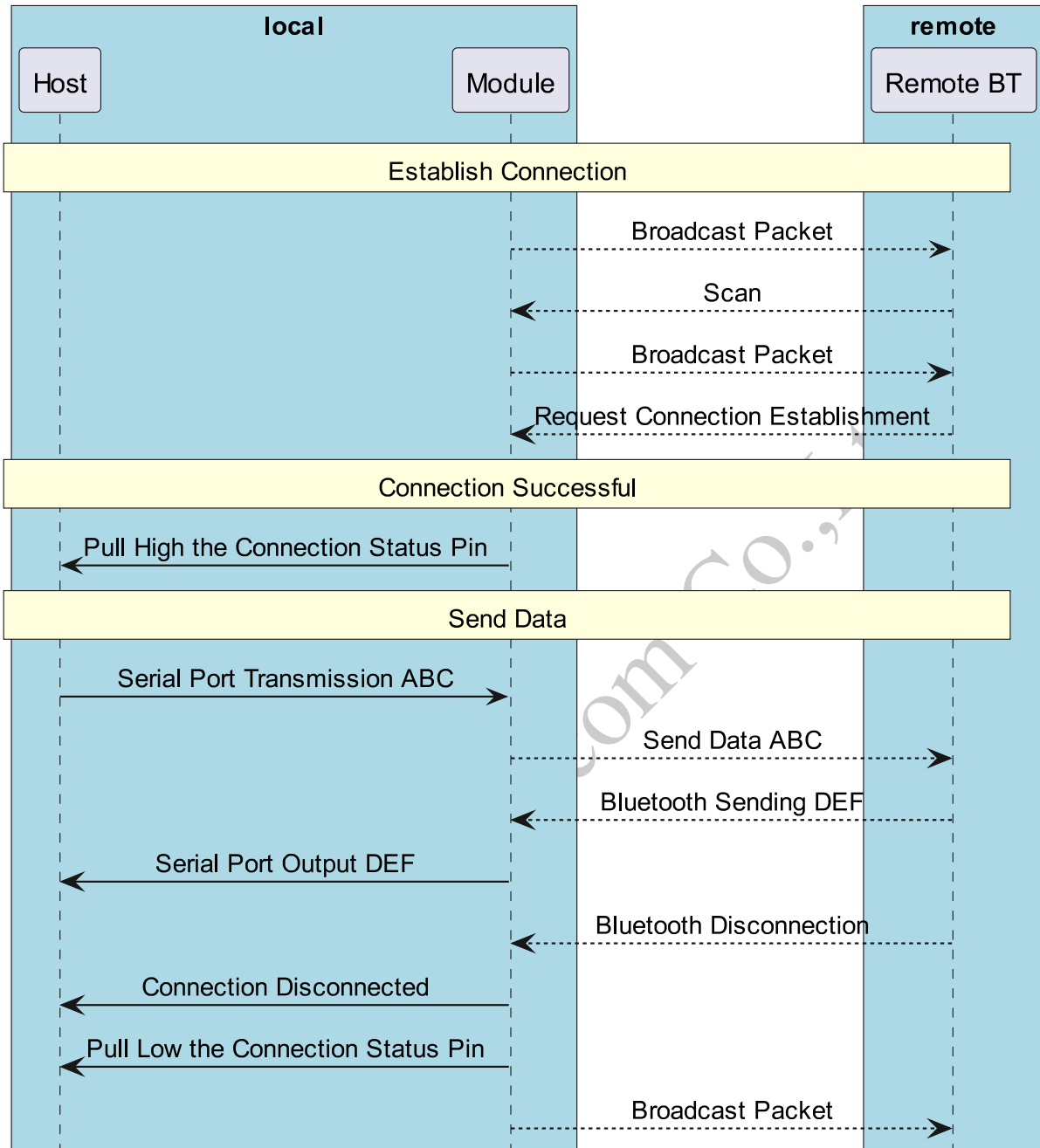
1. Write Device Name : ABC

2. Read Device Name
3. Read Bluetooth Address



6.3 Data Transmission Flow

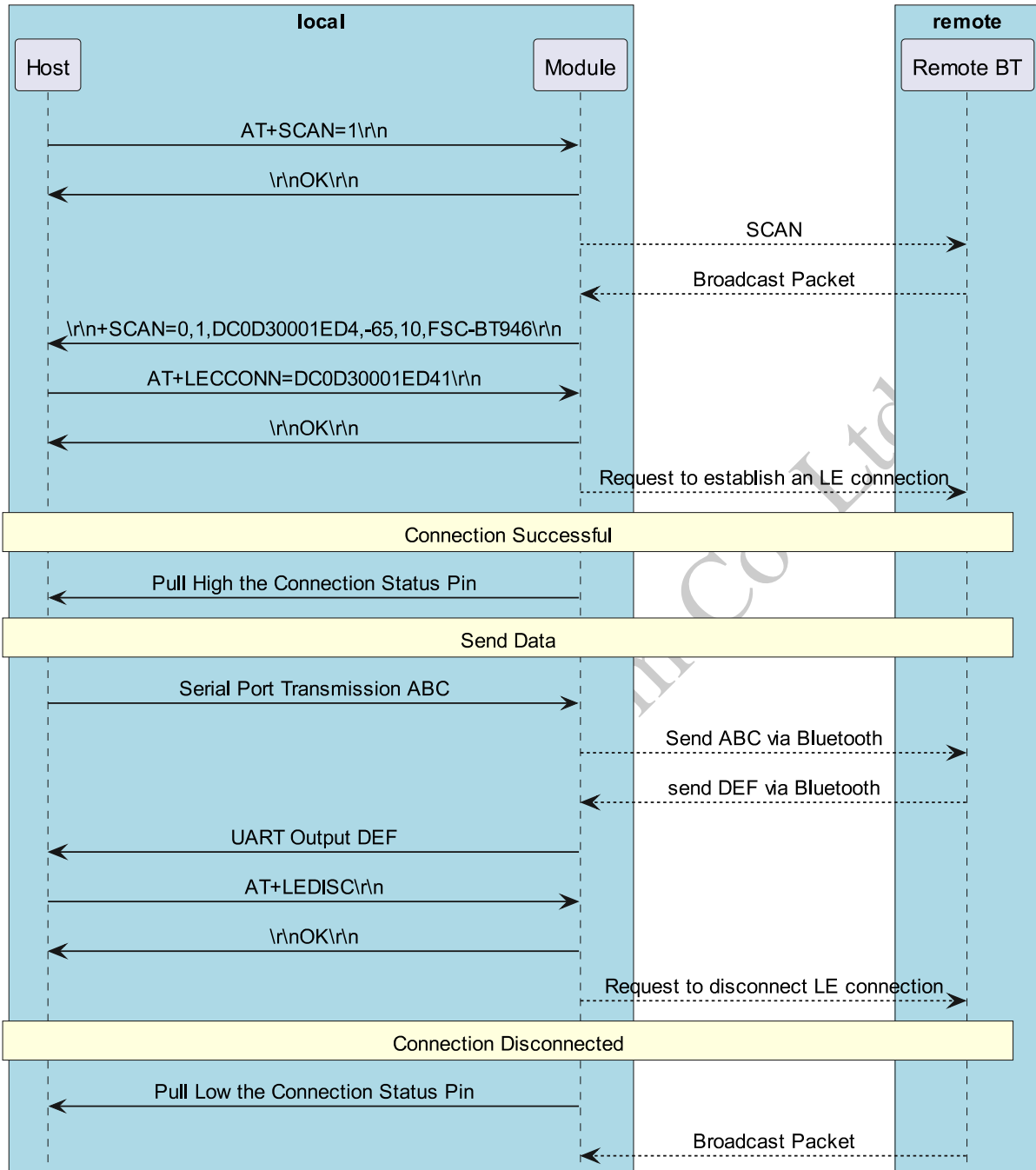
1. When the module is powered on, it continuously sends broadcast data outward. A remote Bluetooth device (e.g., a mobile phone) can obtain the broadcast packet by searching and initiate a connection request to the module.
2. After the connection is successfully established, the module will pull high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



Chapter 7

Firmware Upgrade

[中文版]

7.1 OTA Upgrade

7.1.1 OTA Upgrade Tool

- FeasyOTA APP : A mobile application based on Android.

7.1.2 Firmware Upgrade File

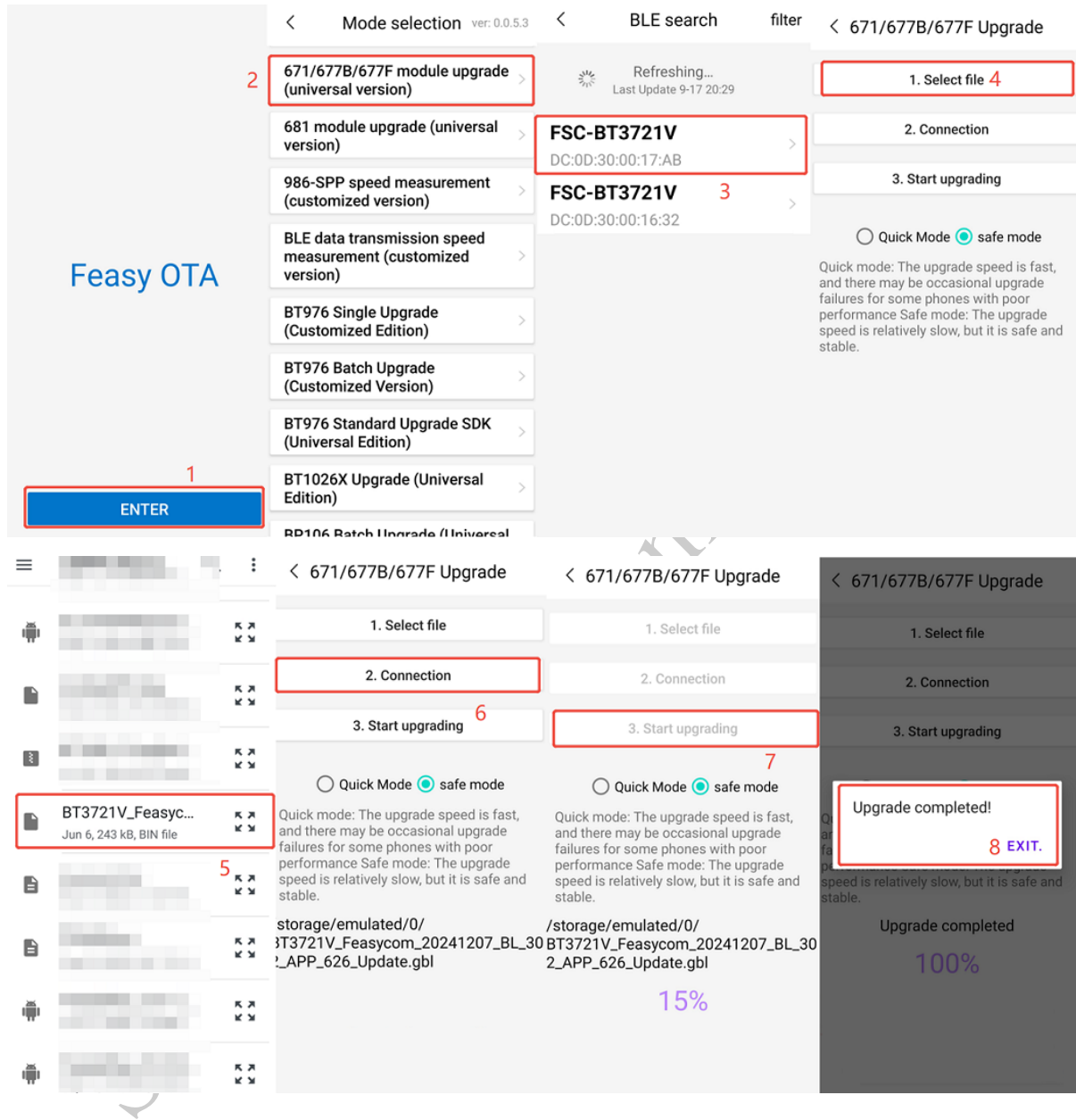
- Firmware upgrade file : **.gbl** format file, provided by Feasycom.

7.1.3 OTA Upgrade Guide

1. Open the FeasyOTA APP , navigate to the mode selection screen, select “**671/677B/677F Module Upgrade (Universal Version)**” , and enter the **BLE Search** screen;
2. On the BLE Search screen, select the BLE device to be upgraded (e.g., FSC-BT3721V). After selection, enter the upgrade configuration screen;
3. On the upgrade configuration, select **1. Select File**, then choose and import the firmware upgrade file stored in the local mobile phone storage;
4. On the upgrade configuration, after loading the firmware upgrade file, select **2. Connect** to establish a BLE connection with the device to be upgraded;
5. On the upgrade configuration, after establishing the BLE connection with the device, select **3. Start Upgrade**. The screen will display the upgrade progress, indicating that the device has

entered upgrade mode and the upgrade has started;

6. When the **upgrade progress shows 100%** and a prompt for successful upgrade appears, it means the upgrade has been completed. Tap **Exit** to finish.



Chapter 8

FAQs

[中文版]

8.1 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 transmission. Some Bluetooth peripheral devices can be connected via the built-in settings program of the mobile phone, 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.

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-BT3721V 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 describeDictionary:advertisementData])
    {
        NSLog(@"is not fsc module");
        return;
    }
}

- (Boolean)describeDictionary: (NSDictionary *) dict
{
    NSArray *keys;
    id key;
    keys = [dict allKeys];
    for(int i = 0; i < [keys count]; i++)
    {
        key = [keys objectAtIndex:i];
        if([key isEqualToString:@"kCBAAdvDataManufacturerData"])
        {
            NSData *tempValue = [dict objectForKey:key];
            const Byte *tempByte = [tempValue bytes];
            if([tempValue length] == 6)
            {
                // tempByte 后面参数是蓝牙地址
                return true
            }
        }
        else if([key isEqualToString:@"kCBAAdvDataLocalName"])
        {
            //there is name
            //NSString *szName = [dict objectForKey: key];
        }
    }
    return false;
}

```

(continues on next page)

(continued from previous page)

```
}
```

Shenzhen Feasycom Co., Ltd.

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

Shenzhen Feasycom Co.,Ltd.

Chapter 10

Appendix

PDF Download

Shenzhen Feasycom Co., Ltd.