FSC-BT618x User Guide

Release 3.5.1

Table of contents

[中文版]

This guide is applicable to the **FSC-BT618x** series Bluetooth BLE data throughput transmission modules, include:

- FSC-BT618
- FSC-BT618V

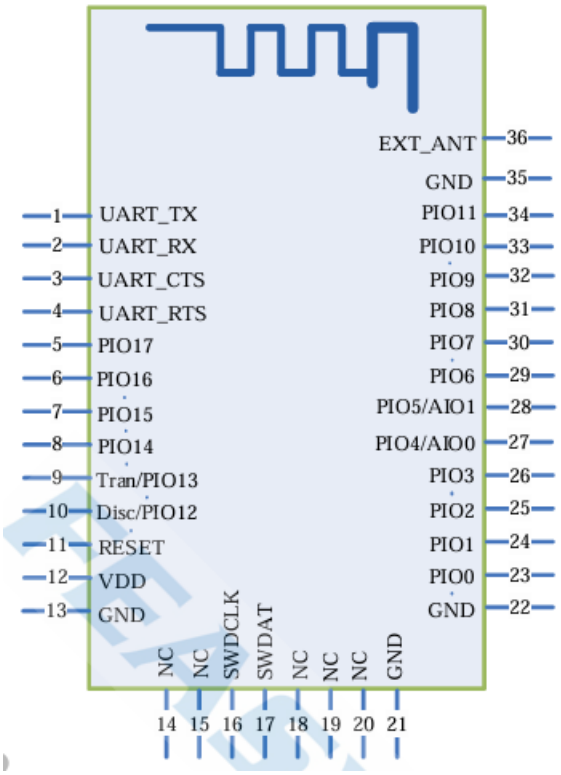
This guide consists of the following parts:

Chapter 1

Hardware Design

[中文版]

1.1 Module Pin Diagram



1.2 Pin Description

Pin	Pin Name	Type	Pin Descriptions
1	UART_TX	O	UART data pin
2	UART_RX	I	UART data pin
3	UART_RT	I/O	Serial Port Flow Control Pin
4	UART_CT	I/O	Serial Port Flow Control Pin
9	RE-STORE	I	Reset configuration
11	RESET	I	Active-low reset
12	VDD	Powe	3.3V power supply.LDO (Low Dropout Regulator) power supply is recommended
13	GND	GND	GND
16	SWCLK	I/O	Programming pin
17	SWDIO	I/O	Programming pin
32	LED	O	Outputs a square wave when Bluetooth is disconnected.outputs high level when Bluetooth is connected
31	WAKE_M	O	Bluetooth in low-power mode wakes up the MCU
33	STATUS	O	Outputs low level when Bluetooth is disconnected, and high level when Bluetooth is connected
34	WAKE_BT	I	MCU in low-power mode wakes up Bluetooth
36	EXT_ANT	ANT	An external Bluetooth antenna can be connected by changing the 0-ohm resistor near the antenna

1.3 Hardware Design Note

- The module can be used by simply connecting-ingVDD/GND/STATUS/UART_RX/UART_TX
- If the MCU needs to obtain the connection status of the Bluetooth module, the STATUS pin (Pin 10) must be connected.
- The module supports GPIO-based wake-up. If the application has low-power consumption requirements, the WAKE_MCU/WAKE_BT pins must be connected.
- VDD/GND/RESET/SWCLK/SWDIO serve as the programming interface, and test

points may be reserved for debugging and testing purposes.

- After completing the schematic diagram, please send it to Feasycom for review to ensure the Bluetooth communication distance reaches the optimal performance.

Chapter 2

Functional Description

[中文版]

2.1 Default Configuration

Name	FSC-BT618
UART Baudrate	115200/8/N/1

2.2 GPIO Indications

2.2.1 LED

PIN	Status	Description
PIN 32	1 Hz square wave	Bluetooth disconnected
PIN 32	High Level	Bluetooth connected

2.2.2 BT Connection Status

PIN	Status	Description
PIN 33	Low Level	Bluetooth Disconnected
PIN 33	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	0xFFF0		Throughput transmission service
Write	0xFFF2	Write,Write Without Response	APP to module
Notify	0xFFF1	Notify	Module to APP

2.5 Performance Parameters

Type	Time	Description
Power-On Time	230ms	UART response enable time
Wake-Up Time	200ms	Calculation starts after the UART finishes sending wake-up data

2.6 Data Rate (Typical)

Baudrate	Data Packet	Transmission Interval	Connection Interval	Transmission Mode	Rate
230400	244	11ms	15ms	Notify	23000 Byte/s

2.7 Low-Power Mode

FSC-BT618x module supports two Low Power Modes (LPM): **UART Wake-up Mode** and **I/O Pin Wake-up Mode**.

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

2.7.1 Low-Power Mode Operation Strategy

- **UART Wake-up Mode**
  - Configuration Command : `AT+LPM=1`
  - Hardware Connection : No need to connect WAKE_MCU and WAKE_BT.
  - Sleep Method : The module automatically enters sleep mode if there is no UART data communication for more than 5 seconds. It exits sleep mode when the first frame of data is received via UART after sleeping.
  - Derscription : The first frame of data upon wake-up will be lost; simple logic and IO resource-saving.
- **I/O Pin Wake-up Mode**
  - Configuration Command : `AT+LPM=2`
  - Hardware Connection : Need to connect WAKE_MCU and WAKE_BT.
  - Sleep Method : When WAKE_BT is at high level, the Bluetooth module enters sleep mode.When WAKE_BT is at low level, the Bluetooth module exits its sleep mode.WAKE_MCU at high level notifies the MCU to exit sleep mode.WAKE_MCU at low level notifies the MCU to enter sleep mode.
  - Derscription :

    Supports MCU wake-up.

Requires 2 additional pin connections in the circuit.

Chapter 3

Data Communication Principles

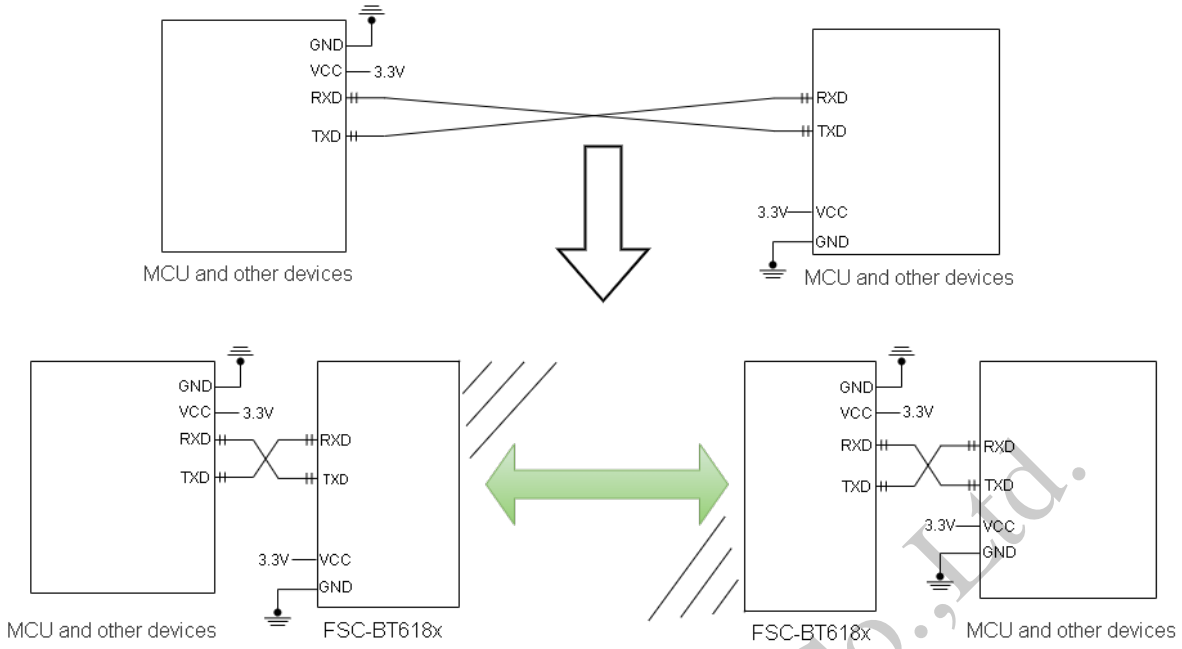
[中文版]

3.1 Working Principle

FSC-BT618x Bluetooth BLE data transmission modules enable wireless communication between devices based on the BLE (Bluetooth Low Energy) protocol.

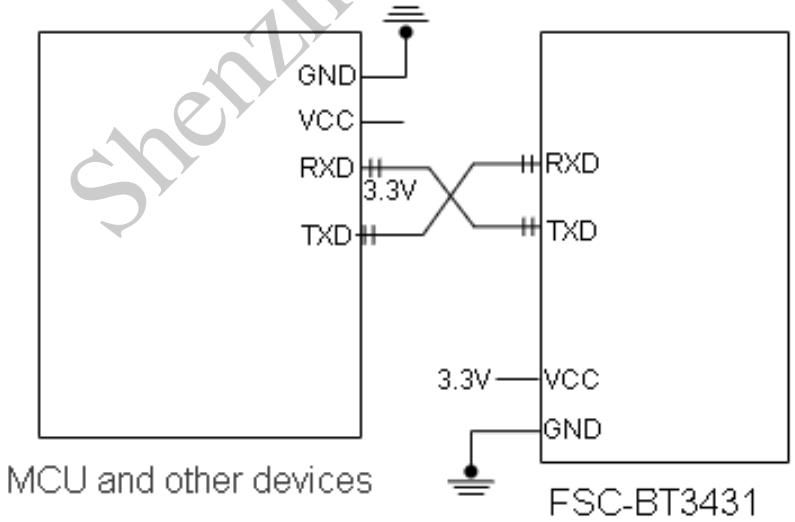
- **BLE** : It adopts an event-driven low-power architecture and defines a “Service-Characteristic” model through the GATT protocol to realize intermittent small-data interaction (e.g., sensor data), making it suitable for IoT (Internet of Things) devices.

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



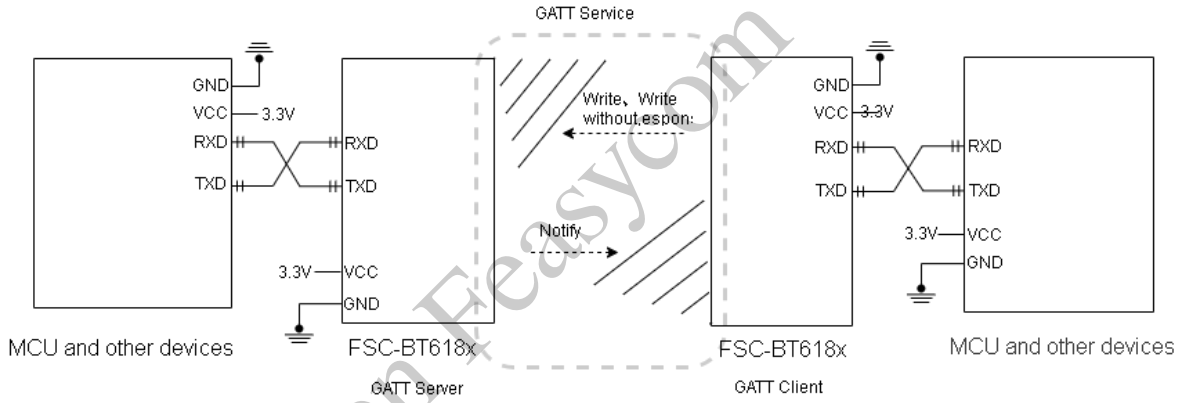
This diagram shows a connection schematic of a main control MCU (Microcontroller Unit) and an FSC-BT618x Bluetooth module. Command interaction between the main control and the

Bluetooth module is realized through serial cross-connection, supporting wireless communication functions, which 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-BT618x Bluetooth modules can establish a Bluetooth connection when powered on.



FSC-BT618x module has master-slave device functions; the left module can be configured as a master device, and the right module as a slave device. The master device can send commands to realize Bluetooth scanning, connection establishment, data transmission, disconnection, etc. Among them, the device that actively initiates a Bluetooth connection is defined as a master device, and the device that receives a connection request is a slave device.

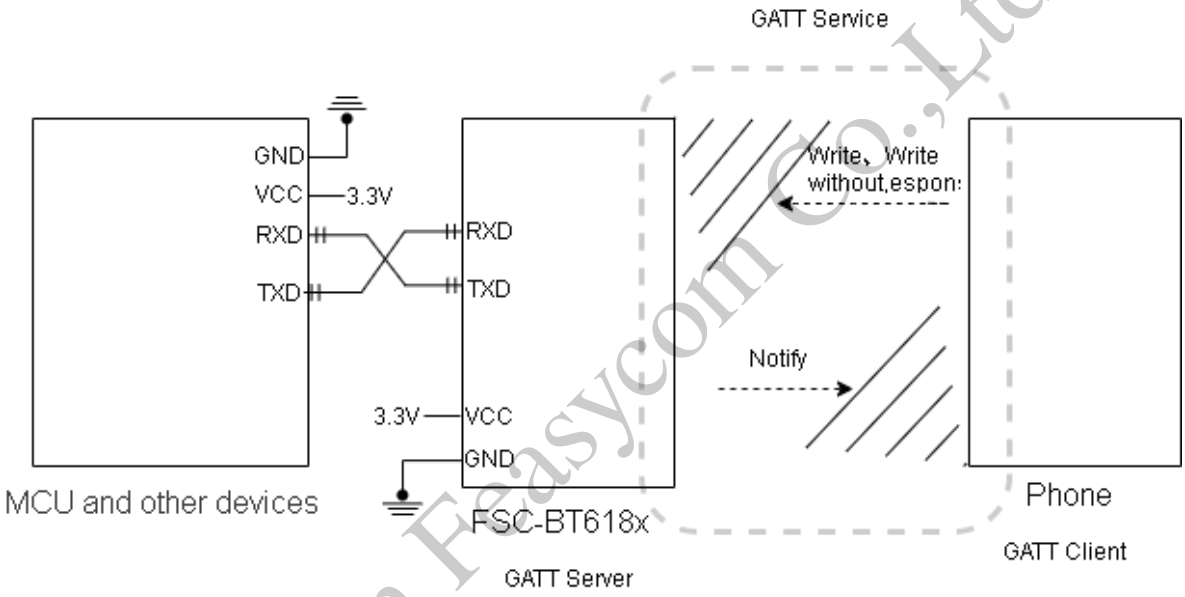
3.4 Module-to-Phone Communication

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-BT618x)： It will continuously send broadcast data outward when powered on.

Mobile phone side： It can search and obtain the broadcast packet of the FSC-BT618x module through the FeasyBlue App , and initiate a MAC address/UUID connection request to the module side (FSC-BT618x), while obtaining all services and characteristics provided by the device. After a successful connection, the Bluetooth module (FSC-BT618x) 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-BT618 Datasheet
- FSC-BT618V Datasheet

4.2 Evaluation Board

- FSC-DB005：Feasycom Bluetooth Data Throughput Transmission Module Development Board.

4.3 AT Command Set

- FSC-BT618x 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 App：Feasycom App & SDK resource supporting Android and iOS, which enables functions such as Bluetooth BLE & SPP data communication test, Feasycom mod-

ule firmware version reading, and parameter configuration and OTA AT commands etc.

4.6 Firmware Upgrade

4.6.1 OTA Upgrade

- Tools：FeasyBlue App and SensorTag App
- User Guide：Please refer to FSC-BT618x OTA Upgrade

Chapter 5

Quick Start

[中文版]

5.1 What you need

5.1.1 Required Hardware

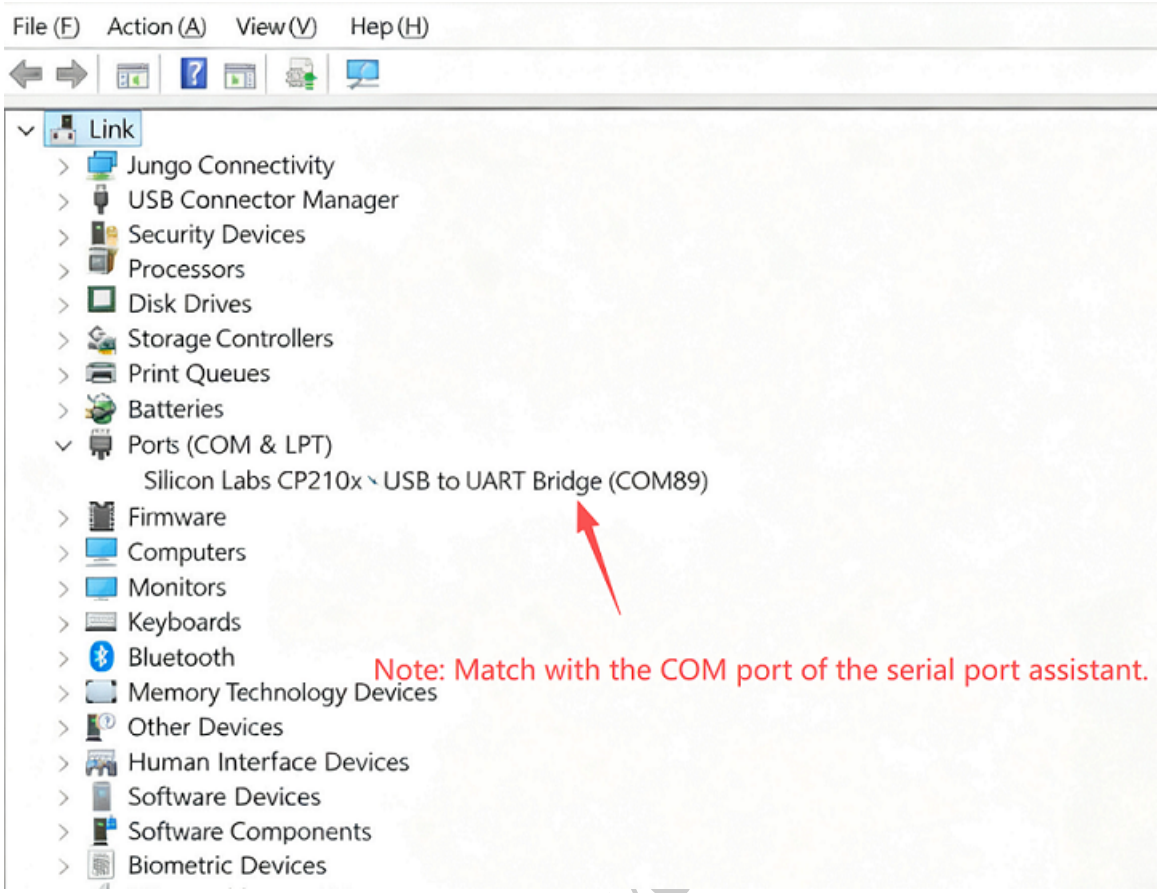
- 1 x FSC-DB005-BT618x Rapid Development Kit : FSC-DB005 USB-to-Serial Rapid Evaluation Board pre-integrated with FSC-BT681x module.
- 1 x PC (Windows / Mac)

5.1.2 Software and Setup

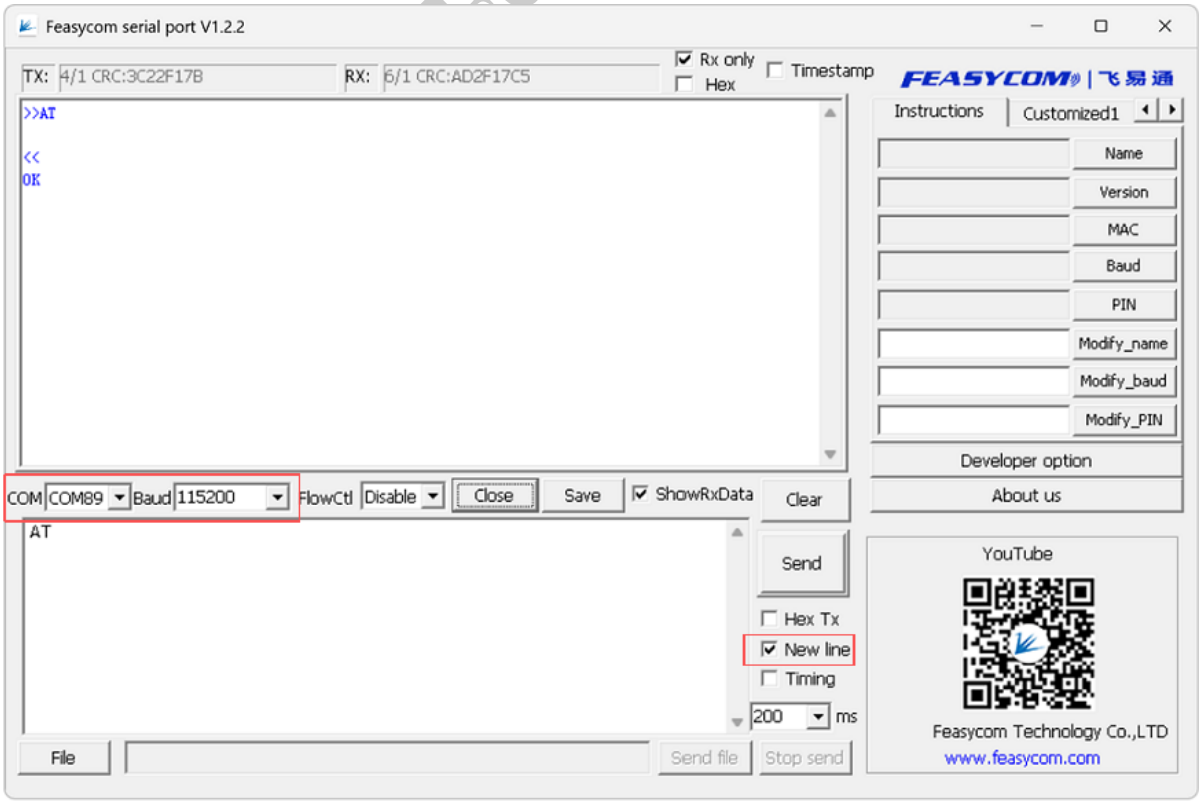
- Feasycom Serial Port Tool : A serial communication analysis tool based on Windows PC.
- **Communication Interface：** UART
- **Serial Configuration：** 115200/8/N/1 (General default for Feasycom firmware)

5.2 Hardware Access

1.Connect the FSC-DB005-BT618x 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** opti.



5.3 UART Communication Test

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

For more commands, please refer to FSC-BT618x General BLE 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=FSC-BT618\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-BT618\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-BT618x Bluetooth BLE data module has two work modes: **Throughput Mode** and **Command Mode** .

The generic data throughput firmware for the FSC-BT618x series modules default to throughput mode. To switch modes, refer to FSC-BT618x 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*.

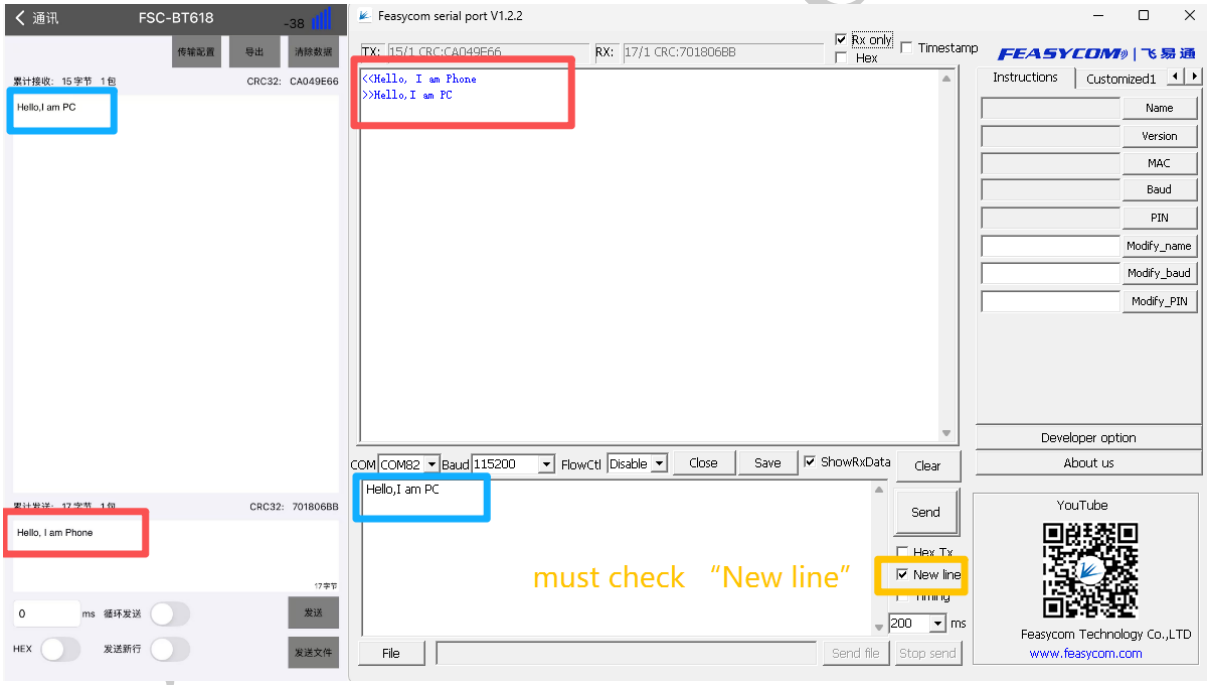
6.1.2 Module to Phone Application

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 the transparent transmission 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 transparent transmission between FSC-BT618(Module1) and FSC-BT618(Module2) Bluetooth modules, as follows:

1.Scan for nearby BLE devices

FSC-BT618(Module1) scans for nearby Bluetooth BLE devices.

```
Send: <<AT+SCAN=1                    // Scan nearby Bluetooth BLE devices
Response: >>OK
     >>+SCAN={                       //Scan Started
     >>+SCAN=0,0,DC0D30001888,-45,10,FSC-BT630
     >>+SCAN=2,0,DC0D30000015,-42,23,FSC-BT1038C-LE-AKM-0015
     >>+SCAN=3,0,DC0D300017A8,-62,11,FSC-BT3721V
     >>+SCAN=4,0,DC0D30000446,-31,9,FSC-BT618                  //Module2
     >>+SCAN=5,0,A4405B28A838,-65,10,AirGo AS01
     >>+SCAN=}                      //Scan Completed
```

2.Establish BLE connection request

FSC-BT618(Module1) establishe BLE protocol connection with FSC-BT618(Module2) via the *AT+LECCONN* command.

```
Send:     <<AT+LECCONN=DC0D300004460      // Establish BLE␣
↪connection request
Response: >>OK
```

Warning

AT+LECCONN=Target Bluetooth MAC address + 1-bit address type.

How to obtain the address type:

Use AT+SCAN=1 for scanning. The second parameter in the return result is the address type, as in the following example:

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

Response: >>+SCAN=+SCAN=4,0,DC0D30000446,-31,9,FSC-BT618
```

3.BLE Connection Established Successfully

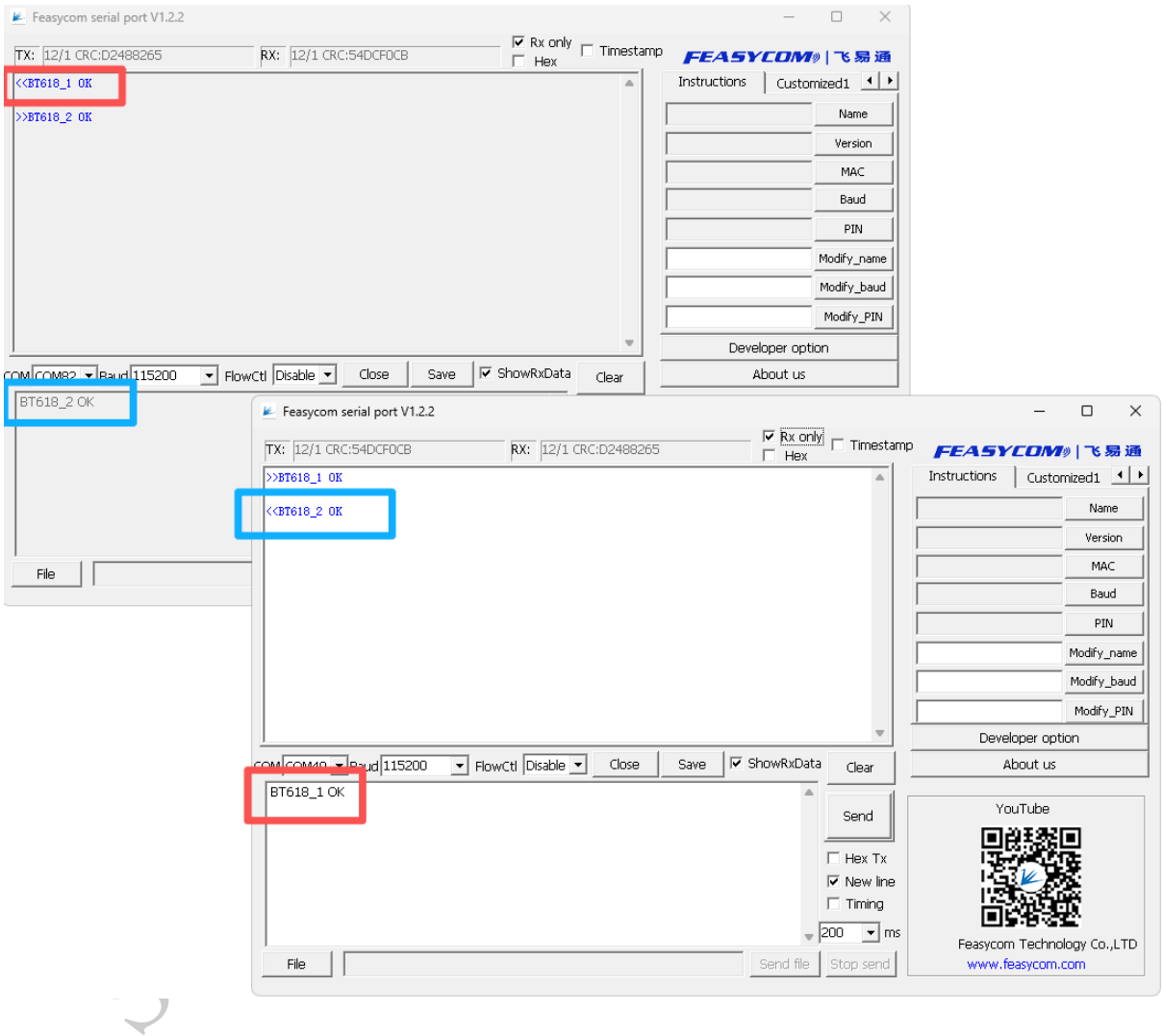
In throughput mode, after the Bluetooth connection is successfully established, the serial port cannot receive event response indicators. The current connection status can be determined by the level state of Pin18 (status indicator pin) of FSC-BT618, as detailed below:

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

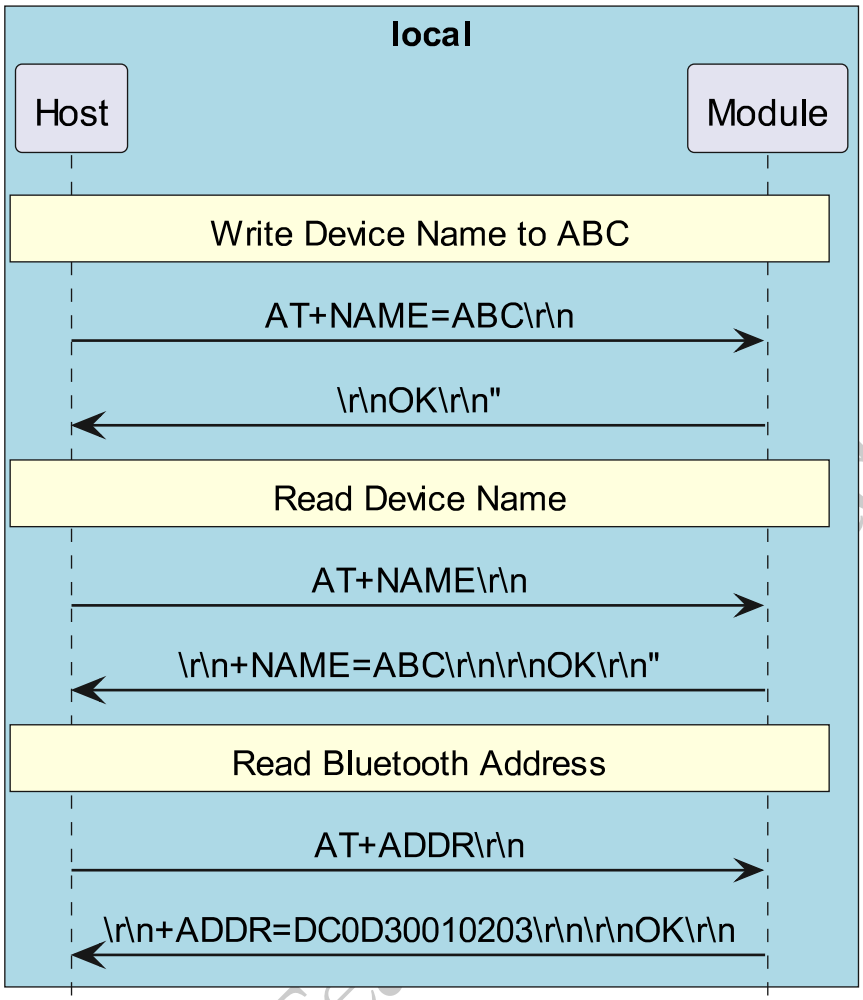
The throughput mode of the general data transmission firmware is enabled by default. After BLE connection is successfully established, data can be sent directly without the need to send data via AT commands:



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

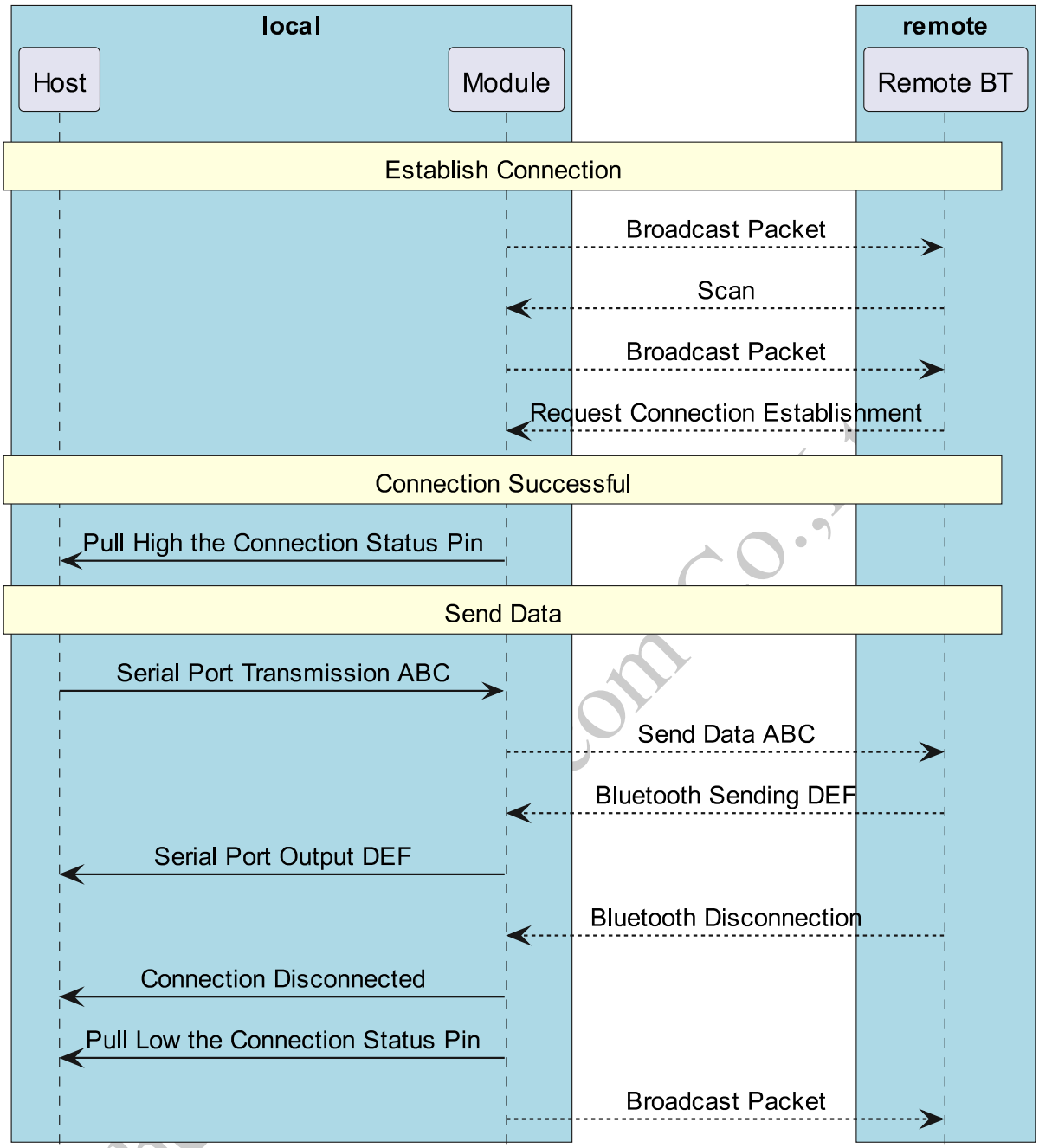


6.3 Data Transmission Flow

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.

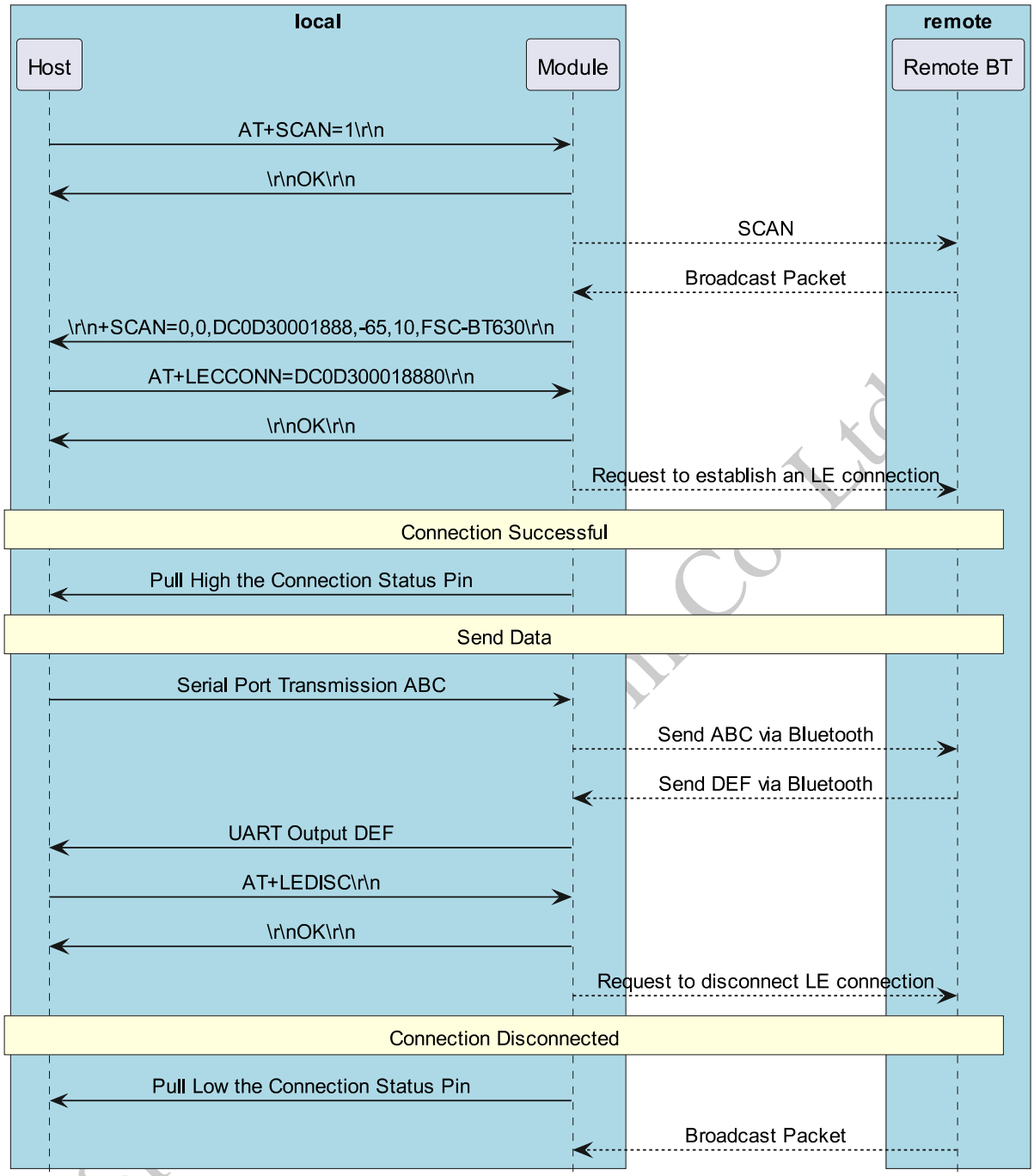
3.The host can send data to the remote Bluetooth device via the Bluetooth module, and the remote Bluetooth device can also send data to the host.



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:


local
remote
Host
Module
Remote BT
AT+SCAN=1\r\n
\r\nOK\r\n
SCAN
Broadcast Packet
\r\n+SCAN=0,0,DC0D30001888,-65,10,FSC-BT630\r\n
AT+LECCONN=DC0D300018880\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 Tools

- FeasyBlue App : Used to configure the module to enter OAD update mode.
- **SensorTag App**: Used to load the firmware file and perform the update.
  - Android: On Google Play, search for “Simplelink SensorTag”, then download and install it.
  - iOS: On the App Store, search for the app, then download and install it.

7.1.2 OTA Upgrade Guide

Enter OAD Firmware Update Mode

Use the FeasyBlue App to put the FSC-BT618x module into OAD update mode. Follow these steps:

1.Open the **FeasBlue App**, check **BLE** mode, search for and connect to the Bluetooth device you want to upgrade (e.g., FSC-BT618). Once connected via BLE, you’ll enter the device’s communication page.

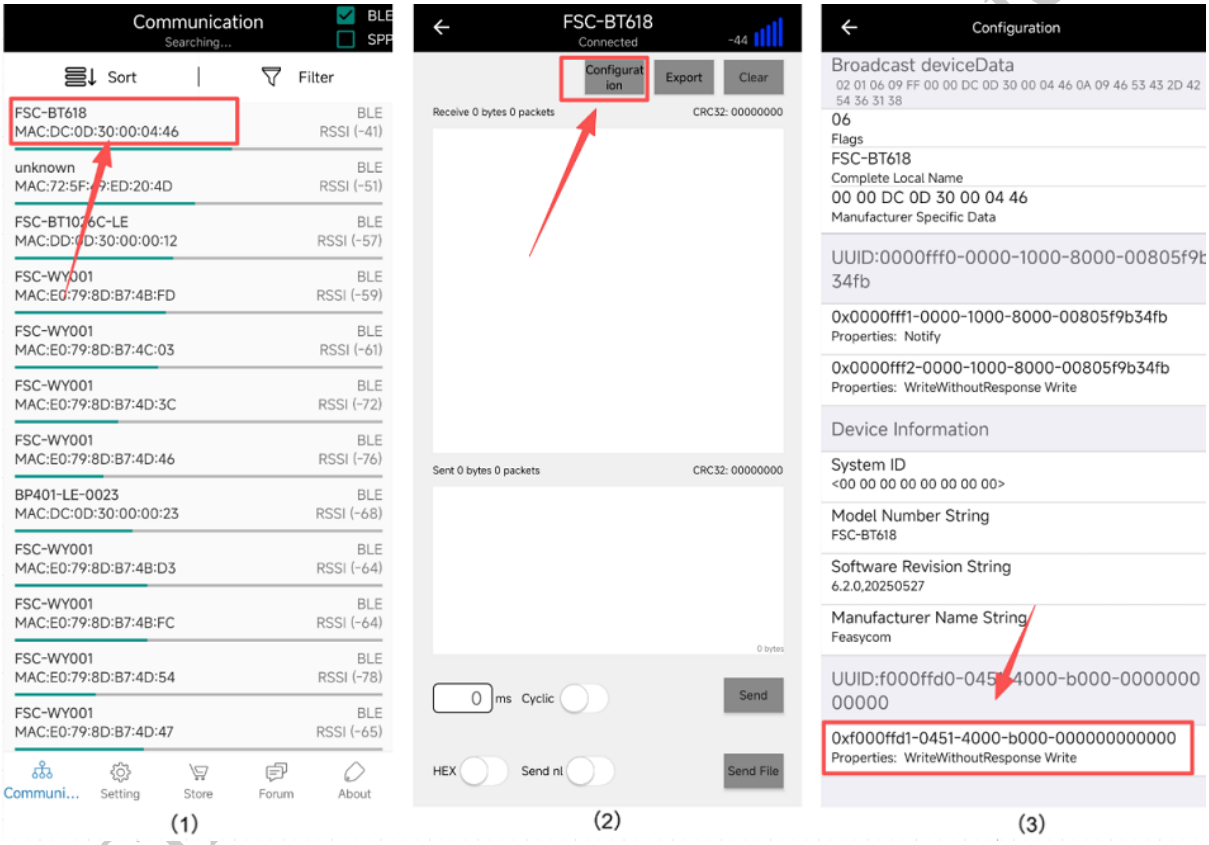
2.On the connected device’s communication page, tap **Configuration** to go to the Configuration page.

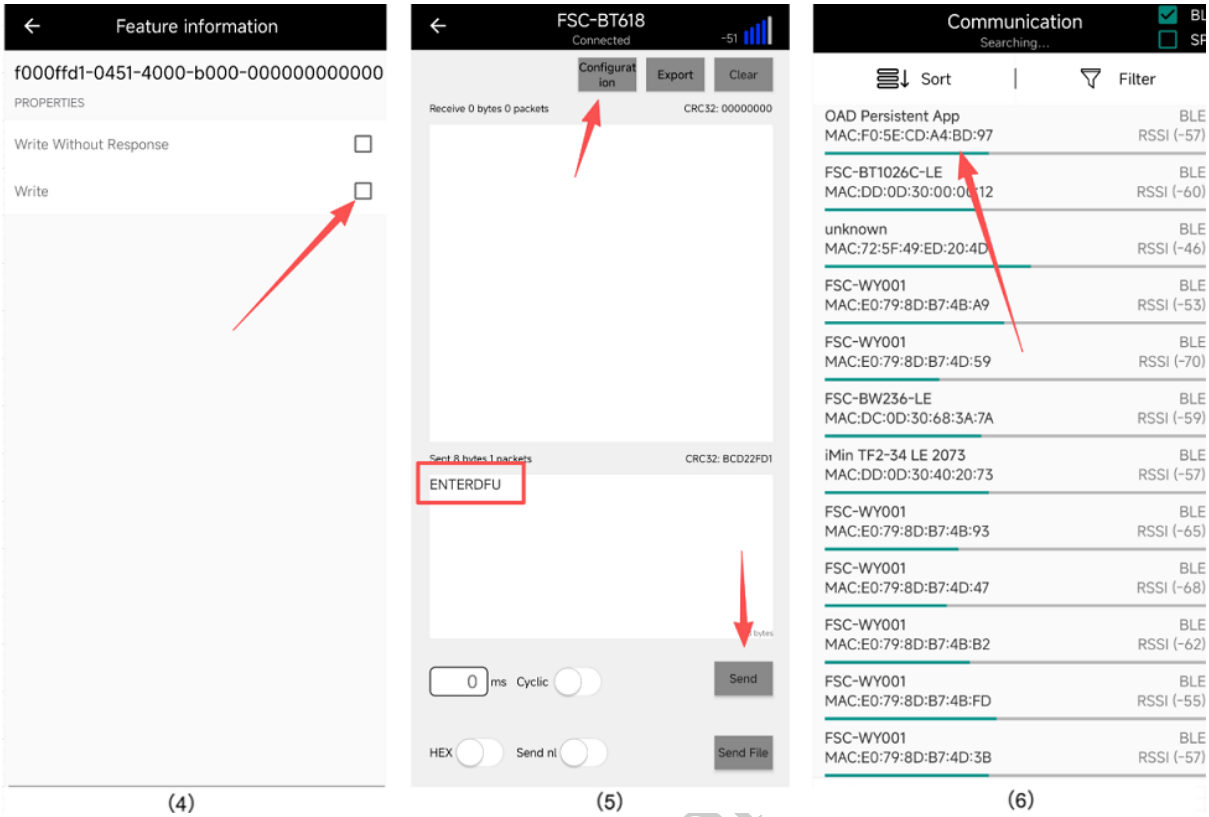
3.On the Configuration page, find the UUID service starting with `0xf000ffd1-`, tap it, and you’ll enter the UUID’s Feature information page.

4.On the UUID’s Feature information page, check the **Write** option, then go back to the previous page (the connected device’s communication page).

5.On the connected device’s communication page, send the command `ENTERDFU` to the Bluetooth module. After it sends successfully, the device will enter OAD update mode, and the App will disconnect from the module.

6.Go back to FeasyBlue’s home page. You’ll see a Bluetooth device named `OAD Persistent App`, that means the module is in OAD update mode.





(4) (5) (6)

Load the Firmware File and Update

Once the device is in OAD update mode, use the **SensorTag App** to do an over-the-air (OTA) firmware update.

1.Save the firmware file to your phone’s local storage, then open the file with the **SensorTag App**. You’ll be taken to the App’s home page.

2.On the **SensorTag App** home page, search for the **OAD Bluetooth device** you want to upgrade, and tap **Sensor View** to connect to it via Bluetooth.

3.After connecting successfully, you’ll automatically enter the OAD Persistent App page. Tap the **FW Download (Enhanced)** section to go to the **Enhanced OAD profile** page.

4.On the Enhanced OAD profile page, tap **Select FW File** to open the Image Files page.

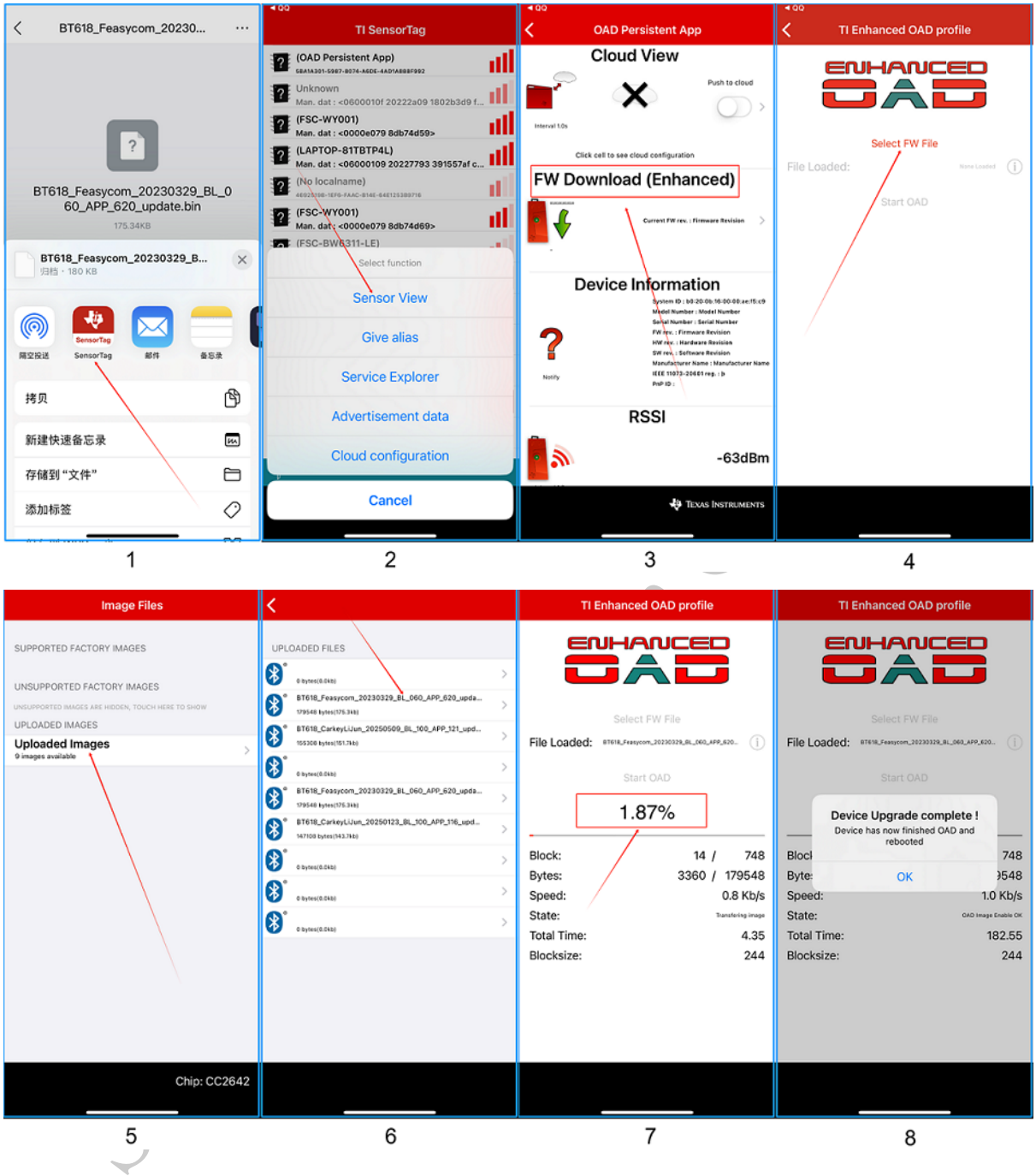
5.On the Image Files page, tap **Uploaded Images** to load the firmware update file from your phone’s local storage.

6.On the Upload files page, select the target firmware file, then tap **Program** , you’ll go back to the Enhanced OAD profile page, and the update will start.

7.A progress bar will appear when the update begins.

8.When you see the message “Device Upgrade complete”, the update is done. Tap “OK” to

close the App.



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-BT3431 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)
            {
                // The parameter after tempByte 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