



FSC-BT3431 Programming User Guide

Release 1.1.1

Table of contents

1	Introduction	1
1.1	Description	1
1.2	Module Default Settings	1
2	Hardware Description	2
2.1	Pin Diagram	2
2.2	Pin Description	3
2.3	Hardware Design Notes	3
3	Function Description	5
3.1	GPIO state	5
3.2	Mode	6
3.3	Introduction to Low-Power Mode	6
3.3.1	Setting up Low-Power Mode	6
3.3.2	Low-power consumption working strategy	7
3.4	GATT Throughput Service	7
3.5	Data transfer rate	8
4	AT command description	9
4.1	Command description	9
4.2	Command	9
4.3	Command Format	9
4.4	Event	10
4.5	Event Format	10
5	Commands Table	12
5.1	AT - Serial communication test	12
5.2	AT+VER - Get Firmware Version	12
5.3	AT+ADDR - Get MAC Address	13
5.4	AT+NAME - Get/Set Local Name	13

5.5	AT+BAUD - Get/Set Uart Baudrate	14
5.6	AT+TXPOWER - Get/Set TX Power	15
5.7	AT+LPM - Get/Set low power config	16
5.8	AT+ADVIN - Get/Set advertising interval	16
5.9	AT+LECHCNT - Get/Set the maximum number of connections	17
5.10	AT+IDCFG - Get/Set 8 bytes ID	18
5.11	AT+LEDISC - Disconnect specified connection	18
5.12	AT+LECONN - Initiate a connection to the specified address	19
5.13	AT+FILTER - Get/Set scan filter	21
5.14	AT+SCAN - SCAN for nearby devices	23
5.15	AT+TPMODE - Get/Set throughput mode	24
5.16	AT+ADVEN - Get/Set advertising enable config	25
5.17	AT+ADVDATA - Get/Set advertising manufacturer specific data	26
5.18	AT+LESEND - send data to remote device	26
5.19	AT+CHINFO - Get connection list information	27
5.20	AT+REBOOT - Soft Reboot	27
5.21	AT+RESTORE - Restore Factory Settings	28
6	Event Table	29
6.1	+SCAN - AT+SCAN=1 Scan Result	29
6.2	+SCAN - AT+SCAN=2 Scan Result	30
6.3	+GATTSTAT - GATT State	31
6.4	+DATA - GATT Received Incoming Data	31
7	Application scenarios	33
7.1	Get/Set the default parameters of the module	33
7.2	Process of sending data	34
7.3	The module acts as the central to connect to the remote device	36
8	FAQ	38
8.1	How does an IOS mobile phone scan the Bluetooth MAC address?	38
9	Appendix	40
9.1	Download PDF Document	40

Chapter 1

Introduction

[中文版]

1.1 Description

This guidance document is suitable for engineers developing FSC-BT3431 series Bluetooth modules

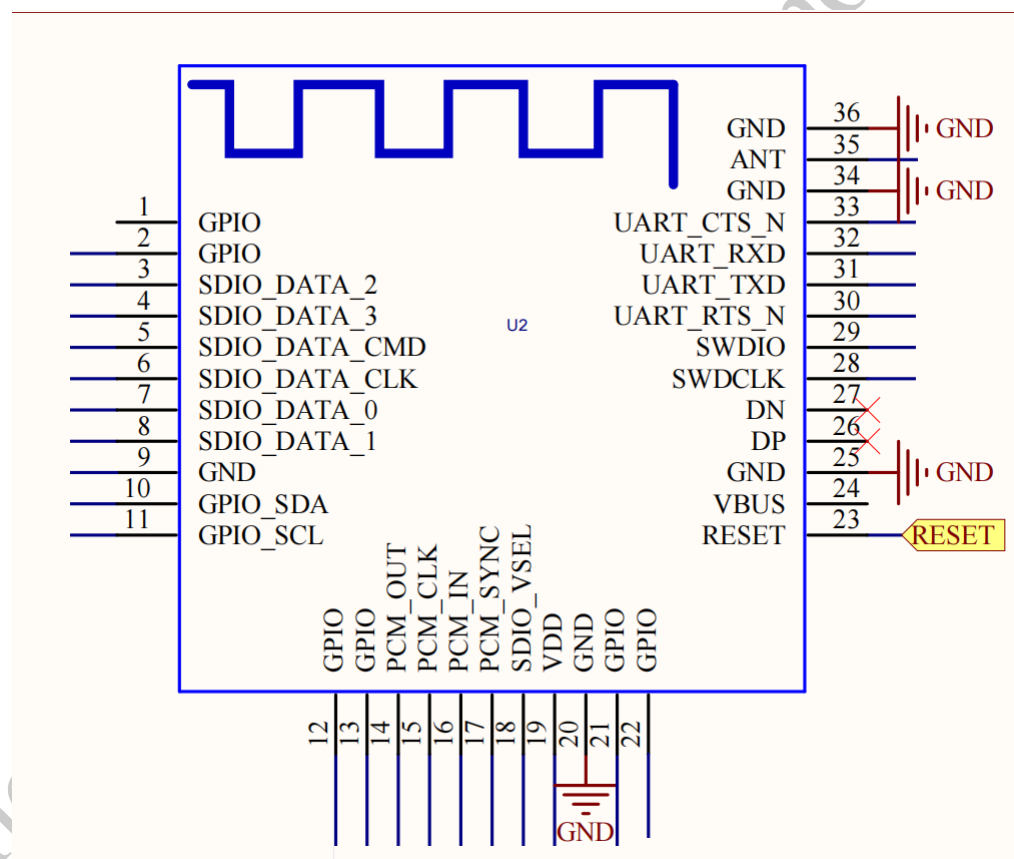
1.2 Module Default Settings

Name	FSC-BT3431
Service UUID	FFF0
Notify UUID	FFF1
Write UUID	FFF2
UART Baudrate	115200/8/N/1

Chapter 2

Hardware Description

2.1 Pin Diagram



2.2 Pin Description

Pin	Pin Name	Type	Pin Descriptions
31	UART_TX	O	UART data output
32	UART_RX	I	UART data input
23	RESET	I	External reset input: Active LOW, with an internal pull-up.
20	VDD	Power	Power supply voltage 3.3V
21	GND	GND	Power Ground
28	SWCLK	I/O	SWD CLK line(Default)
29	SWDIO	I/O	SWD DATA line(Default)
8	SLP_IND	O	After Bluetooth connection, the module pulls low to notify the customer's MCU to exit sleep mode, and outputs serial data with a delay of 10ms; after Bluetooth disconnection, the module pulls high to notify the customer's MCU to enter sleep mode.
7	WAKE_UP	I	MCU pulls high to put the module into sleep mode, and MCU pulls low to wake the module from sleep mode.
21	LED	O	Bluetooth not connected output square wave, Bluetooth connection output high level
22	STATUS	O	Bluetooth not connected output low level, Bluetooth connection output high level

2.3 Hardware Design Notes

- The simple test of the module only needs to connect **VDD/GND/UART_RX/UART_TX** to use
- If the MCU needs to obtain the connection status of the Bluetooth module, it needs to be connected to the STATUS pin (PIN 10)
- If the user wants to obtain the working status of the module, they need to connect to the

LED pin (pin 21)

- If need low power consumption, please connect PIN8 and PIN7
- After drawing the schematic diagram, please send it to Feasycom for review, so as to prevent the Bluetooth distance from not reaching the best effect

Shenzhen Feasycom Technology Co., Ltd.

Chapter 3

Function Description

3.1 GPIO state

LED PIN is PIN 21

State	Description
1Hz square wave	disconnect state
high level	connected state

Connection status PIN is PIN 22

State	Description
low level	disconnect state
high level	connected state

3.2 Mode

Throughput mode	when disconnected, the data received by the serial port is analyzed according to the AT command; when connected, all the data received by the serial port is sent to the remote Bluetooth as it is.
Command mode	when disconnected, the data received by the serial port is analyzed according to the AT command; when connected, The data received by the serial port is still parsed according to the AT command. When it is necessary to send data to the remote end, send AT+LESEND command.

3.3 Introduction to Low-Power Mode

3.3.1 Setting up Low-Power Mode

To activate or deactivate the low-power functionality using the command ‘AT+LPM{=Param}’

3.3.2 Low-power consumption working strategy

AT+LPM=1	- When the BT serial port does not receive data for more than 5 seconds, it automatically enters low-power consumption mode - The first frame of data from the BT serial port wakes up the Bluetooth - Connection successful, not entering low-power consumption mode.
AT+LPM=2	MCU pulls high PIN7 to put the BT into sleep mode MCU pulls low PIN7 to wake up the BT from sleep mode After Bluetooth connection, the BT pulls low PIN8 to notify the MCU to wake up from sleep mode, and then outputs data with a delay of 10ms After Bluetooth disconnects, the BT pulls high PIN8 to notify the MCU to enter sleep mode

3.4 GATT Throughput Service

type	UUID	Permissions	Description
Service	0xFFFF0		GATT Throughput Service
Write	0xFFFF2	Write, Write Without Response	APP send data to module
Notify	0xFFFF1	Notify	Module send data to remote

3.5 Data transfer rate

Baudrate	packet size	interval	conn inter- val	BLE method	rate
921600	244	20ms	7.5ms	Notify	12000 Byte/s

Chapter 4

AT command description

4.1 Command description

- Throughout this specification:
- Content between { } is optional
- Content behind << represents a COMMAND from Host
- Content behind >> represents a RESPONSE to Host

4.2 Command

The command is the control command sent by the host to the module actively. After receiving the command, the module needs to reply <CR><LF>OK<CR><LF> as a response.

4.3 Command Format

AT+Command{=Param1{,Param2{,Param3...}}}<CR><LF>

- All commands start with “AT” , end with <CR><LF>
- <CR> means “carriage return” , corresponds to hex value 0x0D
- <LF> means “line feed” , corresponds to hex value 0x0A

- If Command has Parameter, Parameter follows behind ‘=’
- If Command has multiple Parameters, Parameter must be separated by ‘,’
- If Command has Response, Response starts with <CR><LF>, ends with <CR><LF>
- Module will always report command’s execution result by using “OK” for success or “ERROR” for failure

Example:

Read module’s name

```
<< AT+NAME
```

```
>> +NAME=Feasycom
```

```
>> OK
```

Write invalid baudrate

```
<< AT+BAUD=0
```

```
>> ERROR
```

4.4 Event

Event is the data that the module actively sends to the host. Generally used to indicate a change of state or data received.

4.5 Event Format

<CR><LF>+Indication{=Param1{,Param2{,Param3...}}}<CR><LF>

- All Events start with <CR><LF>, end with <CR><LF>
- If Event has Parameter, Parameter follow behind ‘=’
- If Event has multiple Parameters, Parameter must be separated by ‘,’

Example:

Receive data from APP in command mode

>> +DATA=0,10,1234567890

Shenzhen Feasycom Technology Co., Ltd.

Chapter 5

Commands Table

5.1 AT - Serial communication test

Command	AT
Response	OK
Description	When powering on or changing the baud rate, test the UART communication between the host and the module

Example:

<< AT

>> OK

5.2 AT+VER - Get Firmware Version

Command	AT+VER
Response	+VER=Param
Param	Firmware version
Description	Get Firmware Version

Example:

```
<< AT+VER
>> +VER=1.1.1,FSC-BT3431
>> OK
```

5.3 AT+ADDR - Get MAC Address

Command	AT+ADDR
Response	+ADDR=Param
Param	Module' s MAC address (12 Bytes ASCII)
Description	Get MAC Address

Example:

```
<< AT+ADDR
>> +ADDR=DC0D30010203
>> OK
```

5.4 AT+NAME - Get/Set Local Name

Command	AT+NAME{=Param1{,Param2}}
Param1	local name(1~29 Bytes ASCII)
Param2	MAC address suffix(0/1,default:0) 0: Disable suffix 1: Enable suffix "XXXX" (lower 4 bytes of MAC address) after local name
Response	+NAME=Param
Param	local name
Description	Write local name if parameter exist, otherwise read current local name

Example:

Read current local name

<< AT+NAME

>> +NAME=FSC-BT3431

>> OK

Change module' s local name to “ABC”

<< AT+NAME=ABC

>> OK

Change module' s local name to “ABC” and enable suffix

<< AT+NAME=ABC,1

>> OK

5.5 AT+BAUD - Get/Set Uart Baudrate

Command	AT+BAUD{=Param}
Param	baudrate (2400/4800/9600/19200/38400/57600/115200/ 230400/460800/921600, default:115200)
Response	+BAUD=Param
Param	baudrate
Description	Module will change baudrate to target value immediately

Example:

Query baudrate

<< AT+BAUD

>> +BAUD=115200

>> OK

Change baudrate

<< AT+BAUD=9600

>> OK

5.6 AT+TXPOWER - Get/Set TX Power

Command	AT+TXPOWER{=Param}
Param	tx power(-20,-12,-8,-4,-2,-0,2,3,4,6,8 default:8)
Response	+TXPOWER=Param
Param	tx power
Description	Get/Set TX Power BT3431 supports a maximum of 8 dB

Example:

Get tx power

<< AT+TXPOWER

>> +TXPOWER=0

>> OK

Set tx power to 0dbm

<< AT+TXPOWER=0

>> OK

5.7 AT+LPM - Get/Set low power config

Command	AT+LPM{=Param}
Param	low power mode (0/1/2, default: 0) 0: disable 1: enable low power and wake up by UART 2: enable low power and wake up by IO
Response	+LPM=Param
Param	low power mode
Description	Get/Set low power config

Example:

Get low power config

```
<< AT+LPM
```

```
>> +LPM=0
```

```
>> OK
```

Set low power config

```
<< AT+LPM=1
```

```
>> OK
```

5.8 AT+ADVIN - Get/Set advertising interval

Command	AT+ADVIN{=Param}
Param	advertising interval (20~10000 ms, default: 152ms)
Response	+ADVIN=Param
Param	advertising interval
Description	Get/Set advertising interval

Example:

Get advertising interval

<< AT+ADVINT

>> +ADVINT=152

>> OK

Set advertising interval to 100ms

<< AT+ADVINT=100

>> OK

5.9 AT+LECHCNT - Get/Set the maximum number of connections

Command	AT+LECHCNT{=Param}
Param	Maximum number of connections allowed (1~2, default:1)
Response	+LECHCNT=Param
Param	Maximum number of connections allowed
Description	A restart is required after setting the maximum number of connections

Example:

Read the maximum number of connections

<< AT+LECHCNT

>> +LECHCNT=1

>> OK

Set a maximum of 2 connections

<< AT+LECHCNT=2

>> OK

5.10 AT+IDCFG - Get/Set 8 bytes ID

Command	AT+IDCFG{=Param}
Param	8 bytes ID
Response	+IDCFG=Param
Param	8 bytes ID
Description	Get/Set 8 bytes ID. Help users save 8 bytes IDs.

Example:

Get ID

```
<< AT+IDCFG
```

```
>> +IDCFG=12345678
```

```
>> OK
```

Set ID

```
<< AT+IDCFG=12345678
```

```
>> OK
```

5.11 AT+LEDISC - Disconnect specified connection

Command	AT+LEDISC{=Param}
Param	index
Response	OK
Description	Disconnect the bluetooth connection of the specified index

Example:

Disconnect device with index 1

```
<< AT+LEDISC=1
```

```
>> OK
```

Disconnect all connections

<< AT+LEDISC

>> OK

5.12 AT+LECCONN - Initiate a connection to the specified address

Command	AT+LECCONN{=Param1{,Param2{,Param3{,Param4{}}}}
Param1	12-byte device address + 1-byte address type
Param2	Communication Service UUID
Param3	Communication UUID with write/write without rsp permission
Param4	Communication UUID with notification permission
Response	OK
Description	Initiate a connection to the specified device, the parameter consists of 12 bytes (device address) and 1 byte (address type), generally the address type is “0” or “1” . address How to get the address type: Send AT+SCAN=1 command, get parameter2. example: +SCAN=0,0,DC0D30001ED4,-65,10,FSC-BT946 Connection command: AT+LECCONN=DC0D30001ED40

Example:

Connect to the specified device, 0 is the address type

<< AT+LECCONN=DC0D3000039E0

>> OK

Initiate a connection to the specified address, using FFF0, FFF2, FFF1 for communication

<< AT+LECCONN=DC0D3000039E0,FFF0,FFF2,FFF1

>> OK

Shenzhen Feasycom Technology Co., Ltd.

5.13 AT+FILTER - Get/Set scan filter

Command	AT+FILTER{=Param1{,Param2}}
Param1	scan filter (0~3) 0: delete all conditions 1: address filtering 2: advertising data filtering 3: RSSI filtering
Param2	The filter type is 1, and the parameter2 is the address The filter type is 2, and the parameter2 is advertising data The filter type is 3, and the parameter2 is RSSI
Response	+FILTER=1,Param1 +FILTER=2,Param2 +FILTER=3,Param3
Description	Filter commands will only affect AT+SCAN=2 When filtering advertising data, the parameter2 fields are: tag, offset, len, data After the configuration is complete, start AT+SCAN=2 to filter the scanned devices according to the set parameters AT+FILTER=0, delete all filter conditions. Filter parameters will not be saved to flash. Do not save when power off.

Example:

address filtering

<< AT+FILTER=1,DC0D30010203

>> OK

Filter advertising data \x is hexadecimal

<< AT+FILTER=2,\xFF,\x04,\x02,\xDC\x0D

>> OK

delete all filter conditions

<< AT+FILTER=0

>> OK

5.14 AT+SCAN - SCAN for nearby devices

Command	AT+SCAN{=Param}
Param	scanning method (0~2) 0: stop scanning 1: Scan nearby devices, filter duplicate addresses, up to 8 devices can be found 2: Scan BLE advertising packets and output original advertising packets, which can be filtered by AT+FILTER command
Response	+SCAN=Param1,Param2,Param3,Param4,Param5,Param6 (AT+SCAN=1 event format) +SCAN=Param21,Param22,Param23,Param24,Param25,Param26 (AT+SCAN=2 event format)
Param1	list number
Param2	MAC Address type (1 byte)
Param3	MAC address (12 bytes)
Param4	RSSI
Param5	Device name length
Param6	Device name
Param21	MAC Address type (1 byte)
Param22	MAC address (12 bytes)
Param23	RSSI
Param24	advertising packet type
Param25	advertising packet length
Param26	advertising packet
Description	AT+SCAN=1 Commonly used for searching before connecting AT+SCAN=2 Often used as a Bluetooth gateway to SCAN for nearby Beacon devices

Example:

SCAN device

```
<< AT+SCAN=1
```

```
>> +SCAN={
```

```
>> +SCAN=0,1,70CFC9A98840,-43,24,LE-Bose QuietControl 30
```

```
>> +SCAN=1,1,DC0D30001ED4,-65,10,FSC-BT946
```

```
>> +SCAN=}
```

Sniff the BLE advertising and return the raw data (the garbled part is the raw data, using hex format)

```
<< AT+SCAN=2
```

```
>> +SCAN{
```

```
>> +SCAN=0,DC0D24CFB6D0,-70,1,31,?. 曹 $ 隙街髯 40#-BP104D
```

```
>> +SCAN}
```

5.15 AT+TPMODE - Get/Set throughput mode

Command	AT+TPMODE{=Param}
Param	Throughput mode (0~1, default 1) 0: Command Mode 1: Throughput Mode
Response	+TPMODE=Param
Param	Throughput mode
Description	When bluetooth connected and in throughput mode, the AT command will be de-active, every byte received via physical UART will be sent to remote.

Example:

Throughput Mode

```
<< AT+TPMODE=1
```

```
>> OK
```

Command Mode

```
<< AT+TPMODE=0
```

```
>> OK
```

5.16 AT+ADVEN - Get/Set advertising enable config

Command	AT+ADVEN{=Param}
Param	enable (0~1, default 1) 0: disable advertising 1: enable advertising
Response	+ADVEN=Param
Param	enable
Description	Turn on/off advertising

Example:

Turn on advertising

```
<< AT+ADVEN=1
```

```
>> OK
```

Turn off advertising

```
<< AT+ADVEN=0
```

```
>> OK
```

5.17 AT+ADVDATA - Get/Set advertising manufacturer specific data

Command	AT+ADVDATA{=Param}
Param	manufacturer specific data, tag 0xff
Response	+ADVDATA=Param
Param	manufacturer specific data, tag 0xff
Description	Get/Set advertising manufacturer specific data

Example:

Get advertising manufacturer specific data, \x is hexadecimal data

```
<< AT+ADVDATA
```

```
>> +ADVDATA=\x03\x01\x02
```

```
>> OK
```

Set advertising manufacturer specific data, \x is hexadecimal data

```
<< AT+ADVDATA=\x03\x01\x02
```

```
>> OK
```

5.18 AT+LESEND - send data to remote device

Command	AT+LESEND{=Param1{,Param2{,Param3}}}
Param1	Connection index
Param2	data len
Param3	data
Response	OK
Description	send data to remote device

Example:

Send 4-byte data to the remote device with connection index 1

```
<< AT+LESEND=1,4,2022
```

```
>> OK
```

5.19 AT+CHINFO - Get connection list information

Command	AT+CHINFO
Response	+CHINFO=Param1,Param2,Param3,Param4
Param1	Connection index
Param2	Connection state (1: not connected, 2: connecting, 3: connected)
Param3	Connection role (0: peripheral, 1: central)
Param4	peer mac address
Description	Get connection list information

Example:

Get connection list information

```
<< AT+CHINFO
```

```
>> +CHINFO=0,3,0,4F85B3319170
```

```
>> +CHINFO=1,1,0,000000000000
```

```
>> OK
```

5.20 AT+REBOOT - Soft Reboot

Command	AT+REBOOT
Response	OK
Description	Module release all Bluetooth connections then reboot

Example:

```
<< AT+REBOOT
```

```
>> OK
```

5.21 AT+RESTORE - Restore Factory Settings

Command	AT+RESTORE
Response	OK
Description	Module restore all factory settings then reboot

Example:

<< AT+RESTORE

>> OK

Chapter 6

Event Table

6.1 +SCAN - AT+SCAN=1 Scan Result

Indication	+SCAN=Param1,Param2,Param3,Param4,Param5,Param6
Param1	Index
Param2	MAC address type (1 byte)
Param3	MAC address (12 Bytes ASCII)
Param4	RSSI
Param5	remote device name len
Param6	remote device name
Description	After the module receives AT+SCAN=1, it will continue to scan, and report through +SCAN after finding the device

Example:

SCAN device

```
<< AT+SCAN=1
```

```
>> +SCAN={
```

```
>> +SCAN=0,1,70CFC9A98840,-43,24,LE-Bose QuietControl 30
```

```
>> +SCAN=1,1,DC0D30001ED4,-65,10,FSC-BT946
```

```
>> +SCAN=}
```

6.2 +SCAN - AT+SCAN=2 Scan Result

Indication	+SCAN=Param1,Param2,Param3,Param4,Param5,Param6
Param1	MAC address type (1 byte)
Param2	MAC address (12 Bytes ASCII)
Param3	RSSI
Param4	advertising packet type
Param5	advertising packet size
Param6	advertising packet(For ease of MCU reading, the data is in hexadecimal format. Please set the serial port display to hexadecimal mode to avoid displaying garbled text.)
Description	After the module receives AT+SCAN=2, it will continue to scan and report through +SCAN after finding the device

Example:

Sniff the BLE advertising and return the original data (the garbled part is the original data, using hex format)

```
<< AT+SCAN=2
```

```
>> +SCAN{
```

```
>> +SCAN=0,DC0D24CFB6D0,-70,1,31,?. 曹 $ 隙街髯 40#-BP104D
```

```
>> +SCAN}
```

6.3 +GATTSTAT - GATT State

Indication	+GATTSTAT=Param1,Param2
Param1	Connection Index
Param2	GATT State (1~3) 1: Standby 2: Connecting 3: Connected
Description	In the command mode, if the connection status of the module changes, it will be actively reported through +GATTSTAT

Example:

Connected

>> +GATTSTAT=0,3

6.4 +DATA - GATT Received Incoming Data

Indication	+DATA=Param1,Param2,Param3
Param1	Index
Param2	Payload length
Param2	Payload
Description	In the throughput mode: report data without +DATA header In the command mode: report data with +DATA header +DATA=1,5,12345

Example:

received data 1234567890

>> +DATA=0,010,1234567890

Shenzhen Feasycom Technology Co., Ltd.

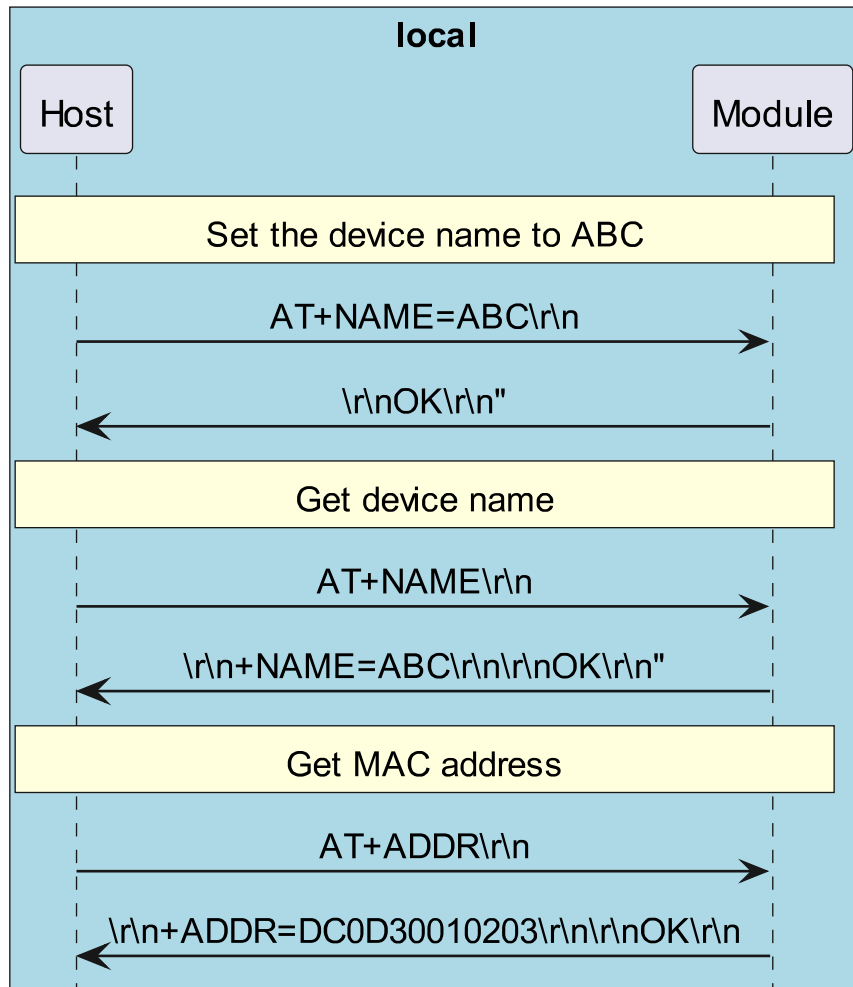
Chapter 7

Application scenarios

7.1 Get/Set the default parameters of the module

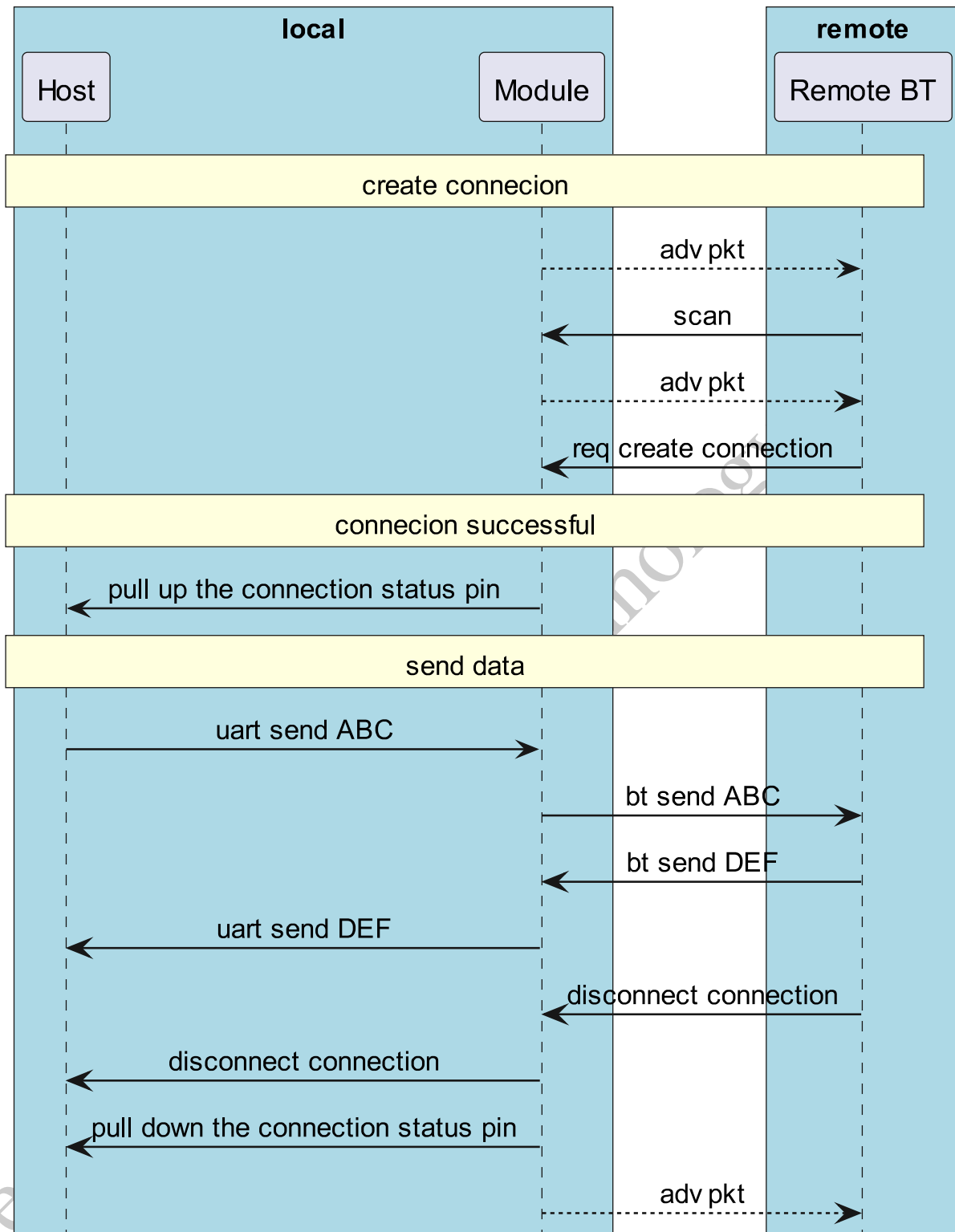
When the module is disconnect state, it will parse the uart data according to the AT command. The host can get and set the default parameters of the module, as shown in the figure below:

1. Set the device name to ABC
2. Get device name
3. Get MAC address



7.2 Process of sending data

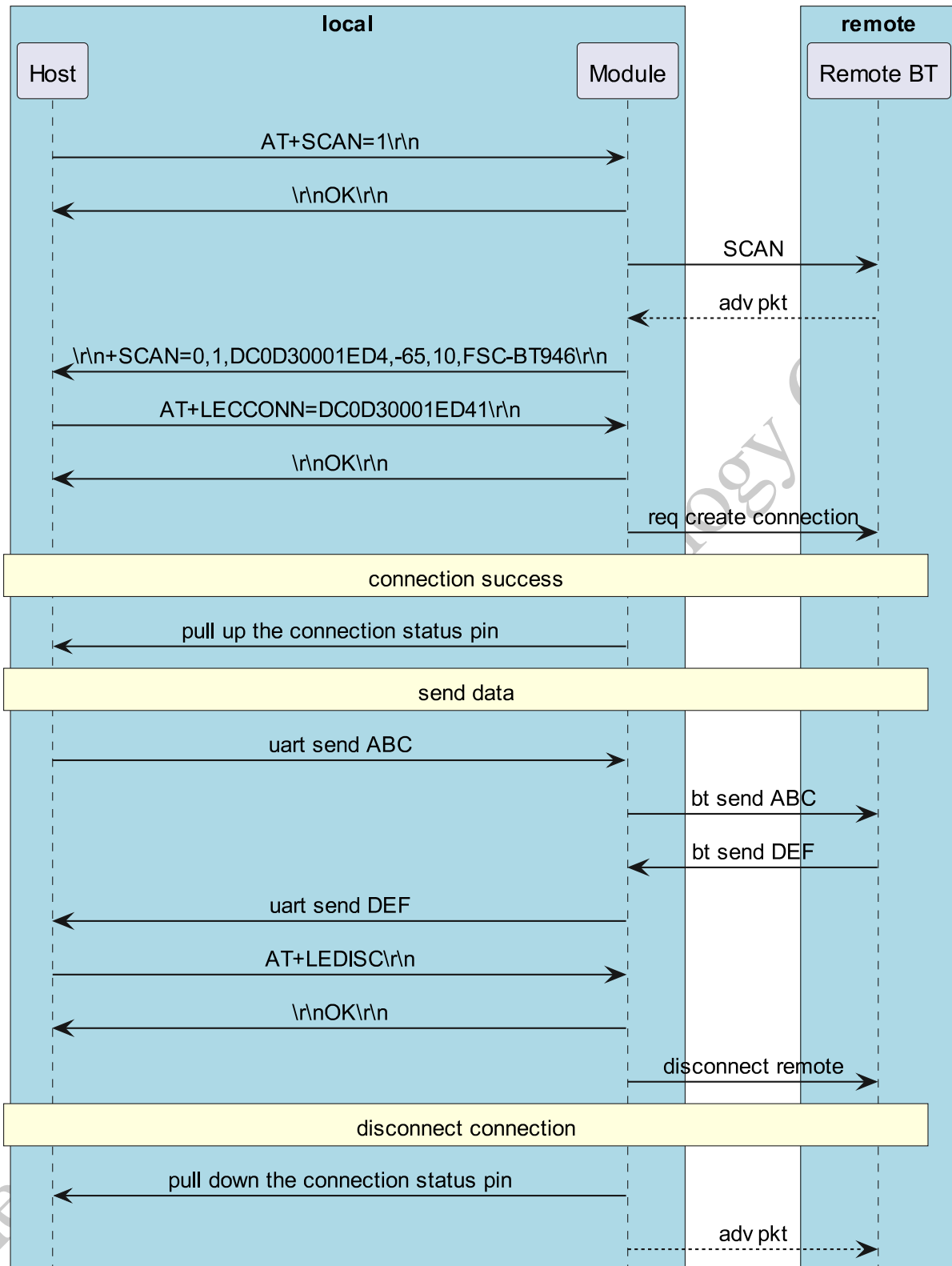
When the module is powered on, it will continue to send advertising data, and the remote device (mobile phone) can scan the advertising packet. And initiate a connection request to the module. After the connection is successful, the module will pull up the connection status pin to notify the host that the Bluetooth connection is successful. The host can send data to the remote device through the Bluetooth module, and the remote device can also send data to the host.



7.3 The module acts as the central to connect to the remote device

The module can be used as a master device to connect to the slave device, and the host can send commands to control the module to scan and connect and disconnect. The figure below shows the process of connecting other devices:

Shenzhen Feasycom Technology Co., Ltd.



Chapter 8

FAQ

8.1 How does an IOS mobile phone scan the Bluetooth MAC address?

For security reasons, the IOS system converts the Bluetooth MAC address into a UUID at the bottom layer and sends it to the upper layer application. Therefore, the APP cannot get the MAC address of the device.

FSC-BT3431 will put the MAC address in the advertising packet by default, and the APP can get the MAC address from the advertising 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;
```

(continues on next page)

(continued from previous page)

```

    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)
            {
                // The parameter after tempByte is the
↪Bluetooth address
                return true
            }
        }else if([key isEqualToString:@"kCBAAdvDataLocalName
↪"])
        {
            //there is name
            //NSString *szName = [dict objectForKey: key];
        }
    }
    return false;
}

```

Chapter 9

Appendix

9.1 Download PDF Document

Download PDF Document