



BT986HC05 Instruction manual

Release 1.0.0

Table of contents

1	Introduce	2
1.1	Description	2
1.2	Default configuration for the module	2
2	Hardware description	3
2.1	Pin diagram	3
2.2	Pin description	4
2.3	Hardware design specification	4
3	Function description	5
3.1	GPIO indication	5
3.2	Working mode	6
4	AT instruction specification	7
4.1	specification	7
4.2	Instruction	7
4.3	Instruction format	7
4.4	Notification	8
4.5	Notification format	8
5	Instruction list	10
5.1	General instruction	10
5.1.1	AT - Serial communication test	10
5.1.2	AT+REBOOT - Software reset	11
5.1.3	AT+RESTORE - factory data reset	11
5.1.4	AT+NAME - Read/write BR/EDR Bluetooth name	12
5.1.5	AT+RNAME - Get the remote Bluetooth device name	13
5.1.6	AT+ROLE - Example Query/set module roles	14
5.1.7	AT+IAC - Set/query the access code	15
5.1.8	AT+INQM - Set/query the access mode	16

5.1.9	AT+CMODE - Query/set the connection mode	17
5.1.10	AT+BIND - Example Query/set the binding Bluetooth address	18
5.1.11	AT+POLAR- Query/set LED indicator driver and connection status output polarity	19
5.1.12	AT+PIO - Set PIO single-port output	19
5.1.13	AT+MPIO - Set PIO multiport output	20
5.1.14	AT+IPSCAN - Settings/Queries - Paging scan, query scan parameters	21
5.1.15	AT+SNIFF - Set/query -SHIFF energy saving parameters	22
5.1.16	AT+SENM - Set or query the security and encryption modes	22
5.1.17	AT+RMSAD - Removes all authenticated devices from the Bluetooth pairing list	23
5.1.18	AT+FSAD - Finds the specified authentication device from the Blue- tooth pairing list	23
5.1.19	AT+LINK - Connecting device	24
5.1.20	AT+ENSNIFF - Enter energy saving mode	24
5.1.21	AT+EXSNIFF - Exit Energy saving mode	24
5.1.22	AT+MRADK - Obtain the address of the recently used Bluetooth authentication device	25
5.1.23	AT+STATE - Obtain the working status of the Bluetooth module	25
5.1.24	AT+INIT - Initializes the SPP specification library	26
5.1.25	AT+INQC - Example Cancel querying Bluetooth devices	26
5.1.26	AT+INQ - Querying Bluetooth devices	26
5.1.27	AT+PAIR - - Device pairing	27
5.1.28	AT+RESET - Module reset (restart) :	27
5.1.29	AT+ORGL - Restore the default state:	27
5.1.30	AT+FSAD - Finds the specified authentication device from the Blue- tooth pairing list	28
5.1.31	AT+PSWD - Read/write pairing password	28
5.1.32	AT+PLIST - Read/clear pairing records	29
5.1.33	AT+VERSION - Read firmware version	29
5.1.34	AT+ADDR - Read the BR/EDR Bluetooth MAC address	30
5.1.35	AT+CLASS: Read/write device type	30
5.1.36	AT+SSP - Read/write BR/EDR pairing mode	31
5.1.37	AT+MODE - - Read/write Bluetooth profiles	31
5.1.38	AT+PIOCFG - Enable or disable the 10-port control module	32
5.1.39	AT+DISC - Disconnect all connections	32
5.1.40	AT+SECURITY - Turn on/off security verification	33

5.1.41	AT+TXPOWER - Read/write transmit power	33
5.1.42	AT+UART - Read/write serial port configuration	34
6	Application scenario	35
6.1	Example Query or modify default module parameters	35
6.2	Process for sending data	36
7	FAQ	38
7.1	IOS How does my phone get a Bluetooth MAC address?	38
8	appendix	40
8.1	Download PDF version	40

[English]

Shenzhen Feasycom Technology Co., Ltd.

Chapter 1

Introduce

1.1 Description

This design guide is intended for engineers developing FSC-BT986_HC05 series Bluetooth modules, as well as BT982 and BT981 series modules

1.2 Default configuration for the module

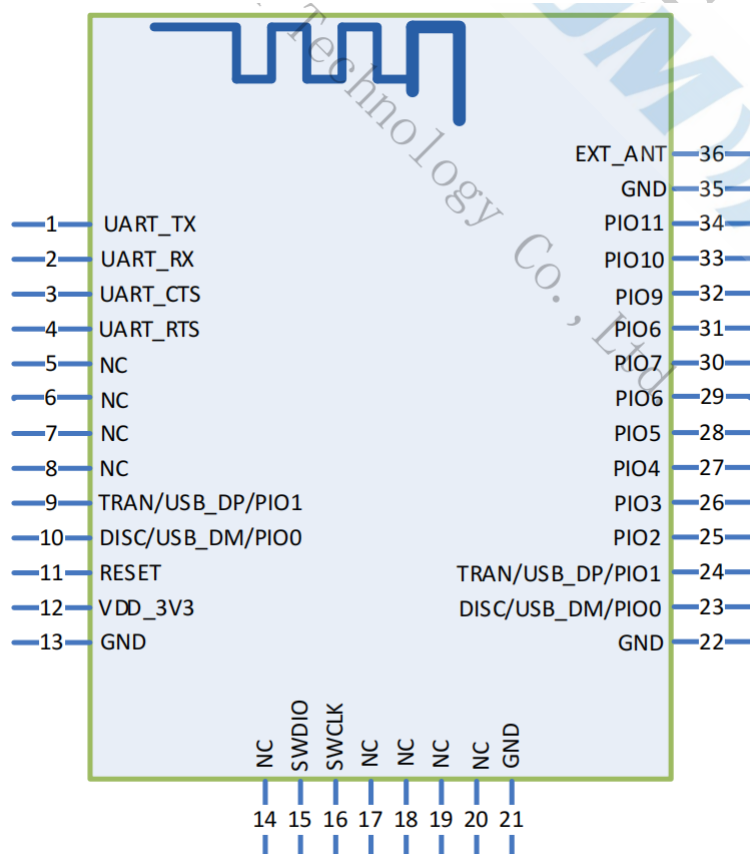
Name	FSC-HC05
LE-Name	FSC-HC05-LE
Pin Code	1234
Secure Simple Pairing Mode	On
UART Baudrate	9600/8/N/1

Chapter 2

Hardware description

2.1 Pin diagram

Take FSC-BT986 as an example:



2.2 Pin description

Pin	Pin Name	Type	Pin Descriptions
1	UART_TX	O	Serial pin
2	UART_RX	I	Serial pin
3	UART_CTS	I/O	Serial port flow controller pin/No connection required
4	UART_RTS	I/O	Serial port flow controller pin/No connection required
9	Disc/USB_D	I/O	Disconnect pin
11	RESET	I	Low level reset
12	VDD	Power	3.3V power supply, LDO power supply is recommended
13	GND	GND	GND
15	ICE	I/O	Burn foot
32	LED	O	Bluetooth is not connected to the output square wave, Bluetooth is connected to the output high level
33	STATUS	O	Bluetooth unconnected output low, Bluetooth connected output high
34	Tran/USB_D	I/O	UART mode control foot • H • = • Command mode • L • = • transparent mode
36	EXT_ANT	ANT	Change the 0 ohm resistance near the antenna, you can connect the Bluetooth antenna

2.3 Hardware design specification

- The module only needs to be connected to VDD/GND/UART_RX/UART_TX
- If the MCU needs to obtain the connection STATUS of the Bluetooth module, it needs to connect the STATUS pin
- After drawing the schematic, please send it to Feiyi Tong for review to avoid the Bluetooth distance can not achieve the best effect

Chapter 3

Function description

3.1 GPIO indication

Connection status PIN24

status	Description
Low level	Transparent mode
High level	Instruction mode

Connection status PIN 23

status	Description
Low level	/
High level	Instruction mode

LED PIN 32

status	Description
1Hz square wave	Bluetooth not connected
High level	Bluetooth connection

连接状态引脚 PIN 33

status	Description
Low level	Bluetooth not connected
High level	Bluetooth connection

3.2 Working mode

Transparent mode	Bluetooth is not connected, the data received by the serial port is analyzed according to the AT command; After the Bluetooth connection, all data received by the serial port is sent to the remote Bluetooth.
Instruction mode	Bluetooth is not connected, the data received by the serial port is analyzed according to the AT command; After the Bluetooth connection, the data received by the serial port is still analyzed according to the AT command. To send data to the remote end, send the AT+SPPSEND command.

Chapter 4

AT instruction specification

4.1 specification

Applies to the entire document

- {} : Including with { ... } The content in {} is optional
- << : The host sent it to the module **COMMAND**
- >> : The module replies to the host **RESPONSE**

4.2 Instruction

The command is the control command that the host actively sends to the module, and the module needs to reply <CR><LF>OK<CR><LF> as a response after receiving the command.

4.3 Instruction format

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

- All instructions begin with **AT** and end with <CR><LF>
- <CR> Indicates the carriage return character, corresponding to HEX is 0x0D
- <LF> Represents a newline character, corresponding to HEX is 0x0A

- If the instruction contains parameters, the parameters should be separated by =
- If the instruction contains multiple parameters, the parameters should be separated by ,
- If a response is returned, the response begins with <CR><LF> and ends with <CR><LF>
- Modules should always return the result of instruction execution (**OK** on success, **ERROR** on failure)

Example:

Read module Bluetooth name

```
<< AT+VERSION
```

```
>> +VERSION: 1.0.0,FSC-BT986
```

```
>> OK
```

Write to an unsupported baud rate

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

```
>> ERROR
```

4.4 Notification

The notification is the data that the module actively sends to the host. Generally used to indicate status changes or received data.

4.5 Notification format

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

- All notifications begin with <CR><LF> and end with <CR><LF>
- If the notification contains parameters, they should be placed after “=”
- If the notification contains multiple parameters, the parameters should be separated by “,”

Example:

Return the current connection status in command mode

>> +SPPSTAT:3

Shenzhen Feasycom Technology Co., Ltd.

Chapter 5

Instruction list

5.1 General instruction

5.1.1 AT - Serial communication test

Command	AT
Response	OK
Description	Test UART communication between the host and module when powering on or changing the baud rate

Example:

<< AT

>> OK

5.1.2 AT+REBOOT - Software reset

Command	AT+REBOOT
Response	OK
Description	Software reset

Example:

<< AT+REBOOT

>> OK

5.1.3 AT+RESTORE - factory data reset

Command	AT+RESTORE
Response	OK
Description	The module will be restored to factory Settings

Example:

<< AT+RESTORE

>> OK

5.1.4 AT+NAME - Read/write BR/EDR Bluetooth name

Command	AT+NAME{=Param1{,Param2}}
Param1	BR/EDR Bluetooth Name (1~31 Bytes ASCII)
Param2	Enable MAC address suffixes (0~2,default:0). 0: Close suffix 1: Enable the suffix “-XXXX” (4 bytes after the MAC address) 2: Enable the suffix “-XXXX” (6 bytes after the MAC address)
Response	+NAME:Param
Param	Bluetooth name
Description	Set the Bluetooth name if there is a parameter, otherwise just read

Example:

Read the BR/EDR Bluetooth name

```
<< AT+NAME
```

```
>> +NAME:FSC-BT986-XXXX
```

```
>> OK
```

Set the BR/EDR Bluetooth name and turn off the suffix

```
<< AT+NAME=ABC,0
```

```
>> OK
```

Set the BR/EDR Bluetooth name to ABC and automatically add an address suffix

```
<< AT+NAME=ABC,1
```

```
>> OK
```


5.1.5 AT+RNAME - Get the remote Bluetooth device name

Command	AT+RNAME {=Param}
Param	The address of the remote Bluetooth device
Response	+RNAME:Param
Param	Name of the remote Bluetooth device
Description	Obtain the remote Bluetooth device name using the remote Bluetooth device address

Example:

Get the remote Bluetooth device name

Remote Bluetooth device MAC is :DC:0D:30:65:43:21, device name is :Feasycom

<< AT+RNAME=DCOD,30,654321

>> +RNAME: Feasycom

>> OK

5.1.6 AT+ROLE - Example Query/set module roles

Command	AT+ROLE=<Param>
Param	Setting module roles 0: Slave role Default 1: Main role (Master) 2: Loop role(Slave-Loop)
Response	ROLE:Param
Param	Query module role
Description	Slave (slave role) : passive connection; Slave-Loop: Passive connection, receiving data from the remote Bluetooth master device and returning the data to the remote Bluetooth master device as is; Master(Master role) : Queries the surrounding SPP Bluetooth slave device and initiates a connection to establish a transparent data transmission channel between the master and slave Bluetooth devices.

Example:

Query module role

```
<< AT+ROLE
```

```
>> +ROLE:0
```

```
>> OK
```

Setting module roles

```
<< AT+ROLE=1
```

```
>> OK
```

5.1.7 AT+IAC - Set/query the access code

Command	AT+IAC=<Param>
Param	Access code
Response	IAC:Param
Param	The access code is queried
Description	The Access Code is set to GIAC (General Inquire Access Code:0x9e8b33), which can be used to show or be found around all Bluetooth devices. In order to efficiently quickly query or be queried a custom Bluetooth device among the many surrounding Bluetooth devices, users can set the module query access code to a number other than GIAC and LIACSuch as9e8b3f

Example:

Set access code

```
<< AT+IAC=9e8b3
```

```
>> OK
```

Query access code

```
<< AT+IAC
```

```
>> 9e8b3f
```

```
>> OK
```

5.1.8 AT+INQM - Set/query the access mode

Command	AT+INQM=<Param>,<Param2>,<Param3>
Param1	0: Standard query mode 1: Query mode with RSSI signal strength indication
Param2	Maximum number of Bluetooth device responses
Param3	Maximum query timeout Timeout range: 1 to 48 (Converted to 1.28 to 61.44 seconds) Default value: 1,1,48
Response	INQM:Param,Param2,Param3
Param1	0: Standard query mode 1: Query mode with RSSI signal strength indication
Param2	Maximum number of Bluetooth device responses
Param3	Maximum query timeout Timeout range: 1 to 48 (Converted to 1.28 to 61.44 seconds) Default value: 1,1,48
Description	Set Bluetooth access mode if parameter exists, otherwise just read

Example:

Set access mode

```
<< AT+INQM=1,9,48
```

```
>> OK
```

Query access mode

```
<< AT+INQM
```

```
>> +INQM:1, 9, 48
```

```
>> OK
```

5.1.9 AT+CMODE - Query/set the connection mode

Command	AT+CMODE=<Param>
Param	Connection mode, default:1 0: Specify Bluetooth address connection mode (Specify Bluetooth address set by binding command) 1: Any Bluetooth address connection mode (not subject to binding instruction set address) 2: Loop role (Slave-Loop)
Response	+CMODE:Param
Param	Connection mode

Example:

Set connection mode

<< AT+CMODE=1

>> OK

Query access mode

<< AT+CMODEM

>> CMODEM:1

>> OK

5.1.10 AT+BIND - Example Query/set the binding Bluetooth address

Command	AT+BIND=<Param>
Param	Bind Bluetooth address (Default :00:00:00:00:00)
Response	+BIND:Param
Param	Bluetooth address
Description	Bluetooth address representation: NAP:UAP:LAP(hexadecimal) binding instruction only in the specified Bluetooth address that mode is valid!

Example:

Set connection mode

In the specified Bluetooth address connection mode,
bind the Bluetooth device address: 12:34:56:ab:cd:ef

```
<< AT+BIND=1234,56,abcdef
```

```
>> OK
```

Querying Bluetooth address

```
<< AT+BIND
```

```
>> BIND:0000:00:000000
```

```
>> OK
```

5.1.11 AT+POLAR- Query/set LED indicator driver and connection status output polarity

Command	AT+POLAR=<Param1>,<Param2>
Param1	0:Output low level light LED 1:Output high level light LED:
Param1	0:Low level indicates successful connection 1:Output high level indicates successful connection
Response	+POLAR=Param1,Param2
Param	LED status

Example:

Set the LED indicator drive and connection status output polarity

```
<< AT+POLAR=0,1
```

```
>> OK
```

Query LED indicator driver and connection status output polarity

```
<< AT+POLAR
```

```
>> +POLAR=0,1
```

```
>> OK
```

5.1.12 AT+PIO - Set PIO single-port output

Command	AT+PIO=<Param1>,<Param2>
Param1	PIO port number(decimal)
Param1	PIO port output status
Response	OK
Description	PI02 to PI07, users can expand the input and output ports.

Example:

The PI005 port outputs a high level

```
<< AT+PIO=05,1
```

```
>> OK
```

PI005 port output low

```
<< AT+PIO=05,0
```

```
>> OK
```

5.1.13 AT+MPIO - Set PIO multiport output

Command	AT+MPIO=<Param>
Param	PIO Port number Mask Combination (decimal)
Response	OK
Description	PI00-PI07 and PI010, users can expand the input and output ports. PIO Port number mask = (1<< Port number) PIO port ID mask combination =(PIO port ID mask 1.PIO port ID mask 2 ...)

Example:

PI02 port mask=(1<<2)=0x004

PI010 Port mask=(1<<10)=0x400

Port mask combination of PI02 and PI010=(0x004 Bitwise and 0x400)=0x404

1、The PI010 and PI02 ports output high levels

```
<< AT+MPIO=404
```

```
>> OK
```

2、The PI04 port outputs a high level

```
<< AT+PIO=004
```

```
>> OK
```

3、PI010 port outputs high level

<< AT+PIO=400

>> OK

5、 All ports output low

<< AT+MPI0=0

>> OK

5.1.14 AT+IPSCAN - Settings/Queries - Paging scan, query scan parameters

Command	AT+PIO=<Param1>,<Param2>,<Param3>, <Param4>
Param1	Query interval
Param2	Query duration
Param3	Paging interval
Param4	Paging duration
Response	AT+PIO:<Param1>,<Param2>,<Param3>, <Param4>
Description	The preceding parameters are in decimal. Default value: 1024,512,1024,512

Set paging scan and query scan parameters

The PI005 port outputs a high level

<< AT+ipscan=1234,500,1200,250

>> OK

Query - Paging scan, query scan parameters

<< AT+ipscan

>> +IPSCAN:1234,500,1200,250

>> OK

5.1.15 AT+SNIFF - Set/query -SHIFF energy saving parameters

Command	AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4>
Param1	Maximum time
Param2	Minimum time
Param3	Try time
Param4	Timeout period
Response	A+SNIFF: <Param1>,<Param2>,<Param3>,<Param4>
Description	The preceding parameters are in decimal. Default value: p,0,0,0

5.1.16 AT+SENM - Set or query the security and encryption modes

Command	AT+SENM=<Param>,<Param2>,<Param3>,<Param4>
Param1	0——sec_mode0+off 1——sec_mode1+non_secure 2——sec_mode2_service 3——sec_mode3_link 4——sec_mode_unknown
Param2	0——hci_enc_mode_off 1——hci_enc_mode_pt_to_pt 2——hci_enc_mode_pt_to_pt_and_bcast Default value: 0,0
Response	A+ SENM:<Param>,<Param2>,<Param3>,<Param4>

5.1.17 AT+RMSAD - Removes all authenticated devices from the Bluetooth pairing list

Command	AT+RMSAD=<Param>
Param	Bluetooth device address
Response	OK or FAIL

Example:

Remove devices with Bluetooth address :34:56:ab:cd:ef from the pairing list

<< AT+RMAAD

>> OK

5.1.18 AT+FSAD - Finds the specified authentication device from the Bluetooth pairing list

Command	AT+FSAD=<Param>
Response	OK or FAIL

Example:

From the pairing list, find the Bluetooth device: 12:34:56:ab:cd: ef

<< AT+FSAD=1234,56,abcdef

>> OK — -There are 12:34:56:ab:cd:ef Bluetooth devices in the pairing list

or

<< AT+FSAD=1234,56,abcdef

>> FAIL — -No 12:34:56:ab:cd:ef Bluetooth device exists in the pairing list

5.1.19 AT+LINK - Connecting device

Command	AT+LINK=<Param>
Param	Remote device Bluetooth address
Response	OK or FAIL

Example:

With remote Bluetooth devices:12:34:56:ab:cd:ef 建立连接

<< AT+FSAD=1234,56,abcdef — -Check whether the Bluetooth device is in the pairing list

>> OK — -There are 12:34:56:ab:cd:ef Bluetooth devices in the pairing list

<< AT+LINK=1234,56,abcdef— -Query Bluetooth device In the pairing list, you can directly connect without query

>> OK

5.1.20 AT+ENSNIFF - Enter energy saving mode

Command	AT+ENSNIFF=<Param>
Param	Device Bluetooth address
Response	OK

5.1.21 AT+EXSNIFF - Exit Energy saving mode

Command	AT+EXSNIFF=<Param>
Param	Device Bluetooth address
Response	OK

5.1.22 AT+MRADK - Obtain the address of the recently used Bluetooth authentication device

Command	AT+MRAD
Param	Address of the most recently used Bluetooth device
Response	+MRAD:<Param> OK

Example:

```
<< AT+MRAD
```

```
>> +MRAD:0:0:0 — -No trusted Bluetooth device has been used recently
```

```
>> OK
```

5.1.23 AT+STATE - Obtain the working status of the Bluetooth module

Command	AT+STATE
Param	“INITIALIZED” —— Initialization state “READY” —— readiness “PAIRABLE” —— Pairable state “PAIRED” —— Pairing status “INQUIRING” —— Query status “CONNECTING” —— Connecting state “CONNECTED” —— Connection state “DISCONNECTED” —— Disconnected state “NUKNOW” —— Unknown state
Response	+ STATE:<Param>
Param	Bluetooth working status

Example:

Set access mode

<< AT+STATE

>> +STATE:INITIALIZED——Initialization state

>> OK

5.1.24 AT+INIT - Initializes the SPP specification library

Command	AT+INIT
Response	OK

5.1.25 AT+INQC - Example Cancel querying Bluetooth devices

Command	AT+INQC
Response	OK

5.1.26 AT+INQ - Querying Bluetooth devices

Command	AT+INQ
Response	**+INQ:<Param1> , <Param2> , <Param3> **
Param1	Bluetooth address
Param2	Device type
Param3	RSSI signal strength

Example:

<< AT+CLASS=0——Query various Bluetooth device classes

<< AT+INQ——Query peripheral Bluetooth devices

>> +INQ:DC0D:72:D2224,3E0104,FFBC

+INQ:1234:56:0,1F1F,FFC1

+INQ:DC0D:30:12,240404,FFC2

>> OK

5.1.27 AT+PAIR - - Device pairing

Command	AT+PAIR=<Param1> , <Param2>
Param1	Remote device Bluetooth address
Param2	Connection timeout (seconds)
Response	OK or FAIL

Example:

Pairing with remote Bluetooth device: 12:34:56:ab:cd: ef, maximum pairing timeout of 20 seconds

```
<< AT+PAIR=1234,56,abcdef,20
```

>> OK

5.1.28 AT+RESET - Module reset (restart) :

Command	AT+RESET
Response	OK

5.1.29 AT+ORGL - Restore the default state:

Command	AT+ORGL
Response	OK
Description	The module will be restored to factory Settings and restarted

5.1.30 AT+FSAD - Finds the specified authentication device from the Bluetooth pairing list

Command	AT
Response	OK or FAIL

Example:

```
<< AT+FSAD=1234,56,abcdef
```

```
>> OK — -There are 12:34:56:ab:cd:ef Bluetooth devices in the pairing list
```

Or

```
<< AT+FSAD=1234,56,abcdef
```

```
>> FAIL — -No 12:34:56:ab:cd:ef Bluetooth device exists in the pairing list
```

5.1.31 AT+PSWD - Read/write pairing password

Command	AT+PIN{=Param}
Param	Pairing password (4 to 15 Bytes ASCII, default:0000)
Response	+PIN:Param
Description	When +SSP=0, the pairing password is valid

Example:

查询配对密码 Query pairing password

```
<< AT+PIN
```

```
>> +PIN:0000
```

```
>> OK
```

Change the pairing password to 1234

```
<< AT+PIN=1234
```

```
>> OK
```


5.1.32 AT+PLIST - Read/clear pairing records

Command	AT+PLIST{=Param}
Param	(0 / 1~8) (0) Clear all pairing records (1~8) Clears the pairing record for the specified index
Response1	+PLIST:Param1, Param2
Param1	(1~8) Paired device serial number
Param2	(MAC) The MAC address of the matching device
Response2	+PLIST= The matching record query is complete

Example:Read the pairing record of the module

<< AT+PLIST

>> +PLIST=1,1C5CF226D773

+PLIST=2,A0BC30075421

>> OK

Example:Clear all pairing records for the module

<< AT+PLIST=0

>> OK

5.1.33 AT+VERSION - Read firmware version

Command	AT+VER
Response	+VER=Param
Param	Firmware version
Description	Read version number

Example:

```
<< AT+VERERSION
>> +VER:1.0.0,FSC-BT986
>> OK
```

5.1.34 AT+ADDR - Read the BR/EDR Bluetooth MAC address

Command	AT+ADDR
Response	+ADDR:Param
Param	Module's BR/EDR Bluetooth MAC address (12 Bytes ASCII)

Example:

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

5.1.35 AT+CLASS: Read/write device type

Command	AT+CLASS=Param
Param	device type (6 bytes ASCII, default:240404 Handsfree device)
Response Description	+CLASS:Param In order to effectively filter many Bluetooth devices and quickly query custom Bluetooth devices, users can set the module to a non-standard Bluetooth device class, such as: ex1f1f(hexadecimal class)

Related configuration Reference [COD](#).

5.1.36 AT+SSP - Read/write BR/EDR pairing mode

Command	AT+SSP{=Param}
Param	PROFILE mode (0~1, default:0) 0: Disabling the SSP 1: Enabling the SSP
Response	+SSP:Param
Note	Restart to take effect

5.1.37 AT+MODE - Read/write Bluetooth profiles

Command	AT+MODE{=Param}
Param	Pairing mode (0~4, default:0) (0) NONE PROFILE (1) SPP PROFILE (2) HID PROFILE (3) BLE PROFILE (4) ALL PROFILE
Response	+SSP:Param

5.1.38 AT+PIOCFG - Enable or disable the 10-port control module

Command	AT+PIOCFG{=Param,Param1}
Param	0: The command/transfer mode switchover function is disabled 1: The command/transmission mode switchover function is enabled
Param1	0: The Bluetooth disconnect function is disabled 1: Enable the Bluetooth disconnect function
Response	+PIOCFG:Param
Description	AT+PIOCFG=1 Turn on the serial flow control and connect the PIN34 to the high level entry command mode.

5.1.39 AT+DISC - Disconnect all connections

Command	AT+DISC
Response	+DISC:SUCCESS ——The disconnect succeeds +DISC:LINK_LOSS ——Connection loss +DISC:NO_SLC ——There is no SLC connection +DISC:TIMEOUT ——Disconnect timeout +DISC:ERROR ——Disconnect error

5.1.40 AT+SECURITY - Turn on/off security verification

Command	AT+SECURITY{=Param1}
Param1	Mode (0~1, default e) 0: Security authentication is not enabled 1: Enable security verification
Response	+SECURITY:Param
Description	Enable or disable serial port flow control

Example:

Set Enable security authentication

```
<< AT+SECURITY=1
```

```
>> OK
```

Obtain security verification information

```
<< AT+SECURITY
```

```
>> +SECURITY=1
```

```
OK
```

5.1.41 AT+TXPOWER - Read/write transmit power

Command	AT+TXPOWER{=Param}
Param	Transmit power (0~2) 0: 0db 1: 3db 2: 5db
Response	+TXPOWER:Param
Param	Transmitting power
Description	Modified transmit power

Example:

Read transmit power

```
<< AT+TXPOWER
```

```
>> +TXPOWER=2
```

>> OK

Set the transmit power Odbm

<< AT+TXPOWER=0

>> OK

5.1.42 AT+UART - Read/write serial port configuration

Command	AT+UART{=Param}
Param	disposition bit0: 0 - 1Bit stop bit 2 1 - 2Bit stop bit 1 bit2~bit1: 00 - None 01 - Odd 10 - Even
Response	+UART=Param
Param	Configuration (0 to 5)
Description	Modifying serial port configurations (stop bit and check bit)

Example:

None 1

<< AT+UART=0

>> OK

Odd 1

<< AT+UART=2

>> OK

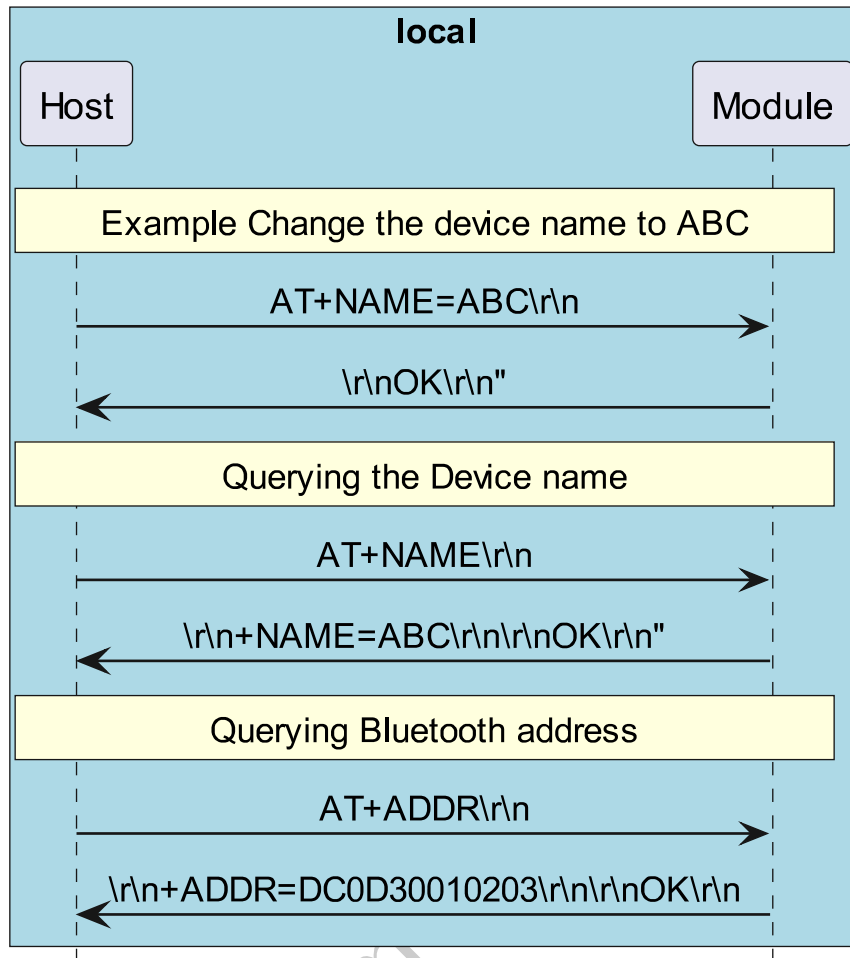
Chapter 6

Application scenario

6.1 Example Query or modify default module parameters

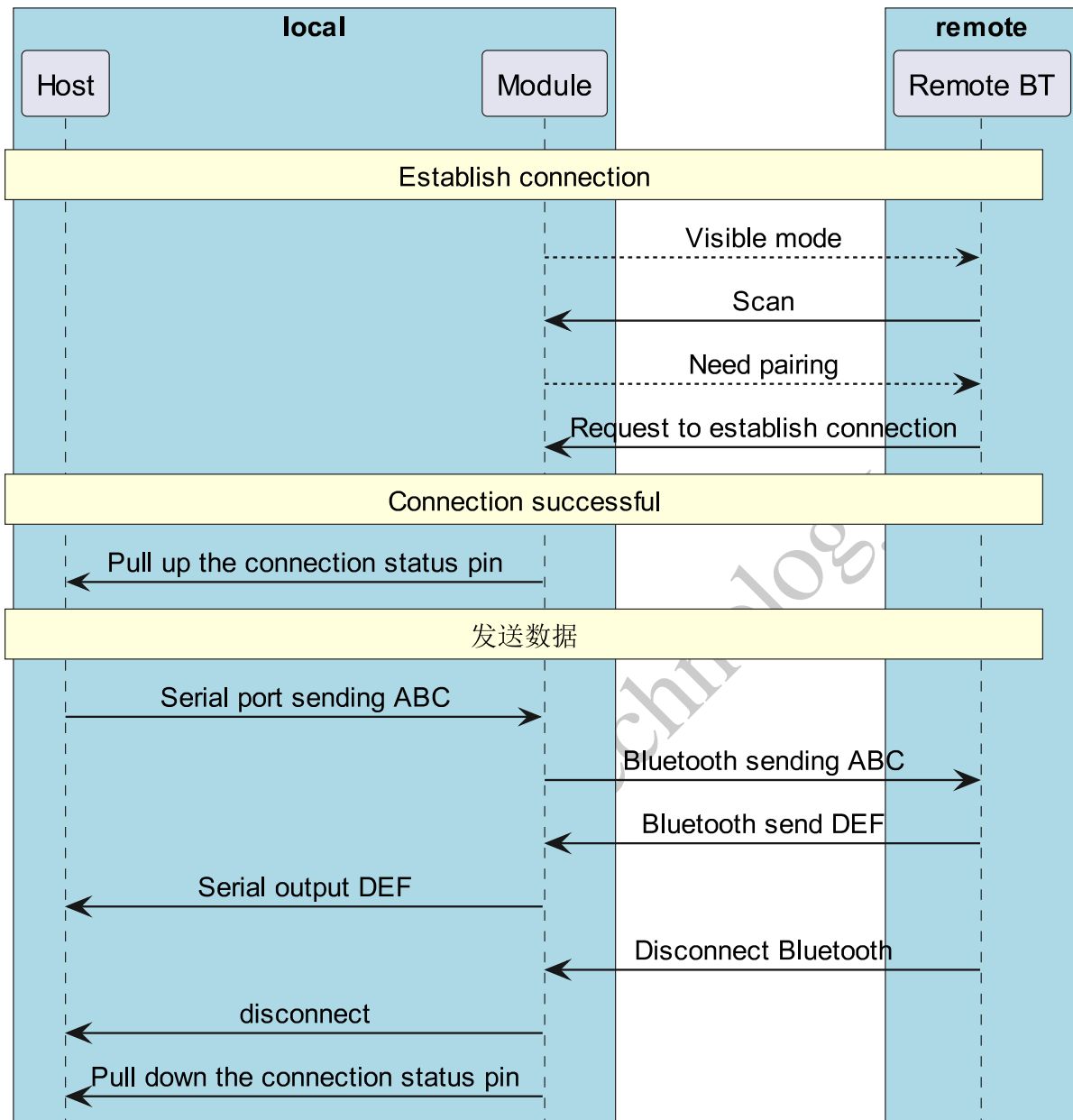
When the module is not connected to the Bluetooth, the serial port data will be parsed according to the AT command. The host can query and modify the default parameters of the module, as shown in the following figure:

1. Example Change the device name to ABC
2. Querying the Device name
3. Querying Bluetooth address



6.2 Process for sending data

The module is powered on in a discoverable (visible) state by default, and the module is set to the state that needs to be paired, and the remote Bluetooth (mobile phone) can find and connect it, and initiate a connection request to the module. After the connection is successful, the module will raise the connection status pin to notify the host that the Bluetooth connection is successful. The host can send data to the remote Bluetooth through the Bluetooth module, and the remote Bluetooth can also send data to the host.



Chapter 7

FAQ

7.1 IOS How does my phone get a Bluetooth MAC address?

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

FSC-BT986_HC05 By default, the MAC address is placed in the broadcast. The APP can obtain the MAC address from the broadcast packet in the following way.

```
- (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];
```

(continues on next page)

(continued from previous page)

```
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;
}
```

Chapter 8

appendix

8.1 Download PDF version

Download PDF version