HJ-580Series Four Products Software Manual

HJ-580CY, HJ-580LAXP, HJ-580Nano, HJ-580XP

Software Manual version:V1.0

CATALOG

1 Version History

Table 1-1 Revising History

No.	Version Number	Release Time	Reviser	Checker	Description
1	V1.0	20190820	ZDY	LMY	Initial Integrated Version

2 Agreement

Four products of HJ-580 series include HJ-580CY, HJ-580LAXP, HJ-580Nano, HJ-580XP. They are Bluetooth modules, and their core chips are DA14580. Therefore, the four products have many identical characteristics and functions.

The "four products" or "four modules" which in the following section represents "HJ-580CY, HJ-580LAXP, HJ-580Nano and HJ-580XP".

In the following section, the "module" which has not specify the model name, it means “four modules”.

And for the four modules, their functions are roughly the same, only in the case of using some pins are different.

3 Summary

●Common characteristics of four products

-they are BLE 4.0 module, using Dialog solution

-they have the UART transparent transmission function

-they have the function of master-slave integration

-very small size, include on-board antenna

-It can be connected and communicated by devices with Android version 4.3 or up.

-It can be connected and communicated by the Apple mobile phone which produced after the IPhone 4S.

-It can be connected and communicated by Ipad with BLE function.

-Different models of modules can also communicate via Bluetooth.

The communication distance of four products which use the on-board antenna as shown in Table 2-1. In addition, the four products have reserved the interface of external antenna. The communication distance of four products which use the external antenna as shown in Table 2-2.

Table 2-1 Comparisons of Communication Distances for Four Products Using On-board Antenna

Module Name	Size	the Communication Distance of the Module which use the On-board Antenna in the Indoor Open Area	the Communication Distance of the Module which use the On-board Antenna in the Outdoor Open Area
HJ-580CY	10*10mm	20 metres	50 metres
HJ-580LAXP	5*6.2mm	10 metres	20 metres
HJ-580Nano	11*12mm	20 metres	50 metres
HJ-580XP	5*6.2mm	10 metres	20 metres

Table 2-2 Comparisons of Communication Distances for Four Products Using External Antenna

Module Name	the Communication Distance of the Module which use the External Antenna in the Indoor Open Area	the Communication Distance of the Module which use the External Antenna in the Outdoor Open Area
HJ-580CY	30 metres	50 metres
HJ-580LAXP		
HJ-580Nano		
HJ-580XP		

The core communication chip of the four products is DA14580, which has the lowest power consumption in the world. The power consumption under different modes is shown in Table 2-3.

Table 2-3 Power Consumption Contrast Table for Four Products in Different Working Modes

Module Name	Dormant Mode	500ms Broadcasting Gap	10ms Broadcasting Gap	Average Current at Full Speed Working
HJ-580CY	Not more than 2uA	Not more than 80μA	Not more than 280μA	less than 1.3mA
HJ-580LAXP				
HJ-580Nano				
HJ-580XP				

Another common feature of the four products is the coexistence of ordinary broadcasting and

Ibeacons broadcasting. The default broadcast mode of the module is to broadcast two ordinary broadcast data first, and then two Ibeacons data. And in the process of switching between two kinds of broadcast data, the module will not consume more power.

You can compare with other brands of modules to see if using Ibeacons broadcasting will increase additional power consumption.

We have designed dozens of instructions for module. You can easily configure various parameters of the module and manage many functions of the module.

4 Overview of Serial Port Transparent Transmission

Four products have the same function which is the serial port transparent transmission. And they have the following characteristics: when the serial port baud rate is not more than 19200bps, the four products can directly transmit a large number of bytes of data without subcontracting, no waiting and no gap in the transmission process.

That is to say, your MCU can send data to the module through the serial port all the time without waiting, thus realizing the function of serial port transparent transmission in the strict sense.

For most BLE Serial Port transparent transmission modules in the past market, they all have a limit on the number of bytes per transmission, and the maximum number of bytes per transmission is about 200 bytes. They can't receive and send data with unlimited bytes. If you send thousands of bytes, you have to divide them into several packages to send. For the receiver of packages, you have to consider the problem of how to combining the data of packages. Therefore, we call the four products is the real sense of serial transparent transmission module.

The serial port baud rate of the four products ranges from 4800bps to 256000bps. Other parameters of serial port, such as stop bit and check bit, can be set freely. Secondly, the serial port parameters of the four products can be changed dynamically. When you set the new serial port parameters, you must change the serial port baud rate of your MCU, otherwise you can't communicate normally. Because DA14580 doesn’t provide an area within which data can be saved before power-down, the personal data set by the user will be lost when the module is powered down. Modules need to be set by your external MCU after each power-on, or you can directly customize modules with specific serial port parameters.

Four products have the same function, one type of module can only connect the same type of module. Because the definitions of data channel and UUID are different in BLE products on the market, it is better to communicate between modules of the same type.

The latest firmware of the four products adds a new function, which adds master-slave binding function.

That is to say, once the module as the master specifies the MAC address of the slave to be connected, or the module as the slave specifies the MAC address of the master to be connected , they can only find the corresponding device and connect according to the specified MAC address, and can’t connect with other devices.

5 Pin Function Definition

This chapter introduces the definition of software functions associated with module's hardware pins. The pin definitions of the four modules are different, but some functions are highly consistent, except that these functions are implemented by different pins in different models of modules. We use the form to list the module model, pin number and function, which is convenient for you to find and compare.

5.1 Selection of Master mode and Slave mode

The four modules select the working mode of the module through a specific pin. The module detects the level of the pin only at the power-on reset stage.

When the pin is input at high level, the module enters the slave mode.

When the pin is input at low level, the module enters the master mode.

After the module enters the slave mode, the module broadcasts to the outside world.

After the module enters the master mode, the module scans the air to find the matching slave for connection.

The master mode and slave mode selection function of the four modules are implemented by which pin, as shown in Table 5-1.

Table 5-1 Contrast table of which pin is used for master mode and slave mode selection function of four modules

Function	Module Name	Pin No.
Master mode and Slave mode selection function	HJ-580CY	Pin2 P14
	HJ-580LAXP	Pin2 P15
	HJ-580Nano	Pin13 P14
	HJ-580XP	Pin2 P15

5.2 Dormancy and Awakening

Module has two working modes, one is dormant state, the other is full speed operation mode. Each of the four modules controls the working mode of the module according to the status of a specific pin. Which pin is used to control the awakening and dormancy of module, refer to Table 5-2. The name of this pin is "Wake Up & Sleep Select".

Table 5-2 Contrast table of which pin is used for awakening and dormancy function of four modules

Function	Module Name	Pin No.	Pin Name
Dormancy and Awakening Function	HJ-580XP	Pin5 P06	Wake Up & Sleep Select
	HJ-580CY	Pin12 P22	
	HJ-580LAXP	Pin5 P06	
	HJ-580Nano	Pin10 P10	

The module works in dormant state by default after power-on.

5.2.1 Dormant state

Regardless of the working state or current mode of the module, as long as the level of the pin is high (TTL, 3.3V), the module immediately enters the dormant state. At this time, if the serial port dormant print notification function is turned on, the serial port will print <SLEEP>.

If the serial port is in the process of data transmission, if the specific pin becomes high level, then the serial port will be immediately disabled, you use MCU to send any data to the module through the serial port will be invalid. However, the data sent by APP or master via Bluetooth to the module is not affected by the dormant state, and the module will send the data automatically through the serial port.

After the module enters dormant state, the module will periodically broadcast the general broadcast and the Ibeacons broadcast in the broadcast cycle that what you set. In this mode, the minimum current consumption of the module is only 2uA, and the maximum current consumption is only 80uA. (In TI or Nordic solutions, with the same parameters, they need nearly 10 times the power consumption!)

5.2.2Full Speed Operation Mode

When the module in dormant state, as long as the specific pin is low level (TTL, 0V), the module immediately wakes up and enters full speed operation mode. At this time, if the serial port wake up print notification function is turned on, the serial port will print <WAKEUP>.

When the specific pin is pulled down, the module will wake up within 50us and run at full speed, at which time the average working current consumed is 1.2mA. When the module runs at full speed, the external 16MHz crystal oscillator works at full speed, and all the enabled peripherals of the module run at full speed.

Only when the module is wake up and running at full speed can you transmit data and send configure instructions through the module's serial port.

5.3 Work Mode Control

There are two working modes of module, one is serial port transparent transmission mode, the other is configuration mode. The working mode control function of the module is implemented by the state of a specific pin, which is defined as "Config Mode" as shown in Table 5-3.

Table 5-3 Contrast table of which pin is used for work mode control function of four modules

Function	Module Name	Pin No.	Pin Name
Work Mode Control Function	HJ-580XP	P00	Config Mode
	HJ-580CY	P14	
	HJ-580LAXP	P00	
	HJ-580Nano	P00	

Note: This function is only valid when the module is in full speed operation mode. When the module is in the dormant state, no matter what the state of the specific pin is, the module will always maintain the serial port transparent transmission mode.

5.3.1 Serial Port Transparent Transmission Mode

When the module is in the dormant state, the module will always maintain the serial port transparent transmission mode. When the module is connected by the mobile phone APP or BLE master, once data is sent to the module through Bluetooth, the module immediately pulls down a specific pin and delays 20 ms. (20ms delay time is enough to wake up the external MCU.) When the delay is over, the module sends the data from the serial port to the MCU which connected to the module's serial port immediately. Modules pull down PIN numbers for specific pins, see Table 5-4.

Table 5-4 Contrast Table of Pin Number of Module Pull Down Specific Pin

Module Name	Pin No.
HJ-580XP	P11
HJ-580CY	P15
HJ-580LAXP	P03
HJ-580Nano	P03

When the module is in full speed operation mode, when the specific pins which used in the work mode control function are in high level state, the module will always maintain the serial port transparent transmission mode. At this time, MCU, module and Bluetooth connected devices can communicate with each other.

When the direction of data transmission is MCU to module, module can receive data from MCU through serial port. When the direction of data transmission is module to MCU, the module is connected by mobile phone APP or BLE master. Once the data is sent to the module through Bluetooth, the module will immediately pull down the specific pin and delay 20ms. (20ms delay time is enough to wake up the external MCU.) When the delay is over, the module sends the data from the serial port to the MCU which connected to the module's serial port immediately. Modules pull down PIN numbers for specific pins, see Table 5-4.

Note: Before MCU transmits data to module through serial port, the notification function of data channel(BLE DATA BUFF) used by module's serial port transmission function must be turned on. Otherwise, the mobile APP or master connected with Bluetooth can not receive the data sent by module.

5.3.2 Configuration Mode

When the module is in the dormant state, regardless of the specific pin state used by the work mode control function, the module will work in the serial port transparent transmission mode.

When the module is in full speed operation mode, when the specific pins used by the operation mode control function are in low level, the module will immediately enter the configuration mode. In configuration mode, you can configure the parameters and functions of the module through the serial port instruction set.

5.4 Busy status indication in configuration mode

When the module is in configuration mode, the serial port of the module only receives instructions from the MCU and replies to the MCU. The data received by the module through Bluetooth from the mobile phone APP or BLE master will not be sent to the MCU through the serial

port, because the serial port of the module is occupied by the configuration mode.

If under the above conditions, the mobile APP or BLE master sends data to the module through “CENTER DATA BUFF” channel (0XFFF2), the “BLE DATA BUFF” channel (0XFFF1) will automatically notify the mobile APP or BLE master by sending <CONFIG MODE BUSY>, telling them that the module is in configuration mode, and then send and receive data later.

5.5 Connection Status Indication

When the module is disconnected from the mobile phone APP or master, a specific pin of the four modules, named “Connect State”, will always remain at high level.

When the module is normally connected by the mobile phone APP or master, the specific pin will output low level, indicating that the module has been connected successfully.

The connection status indication function of the module is implemented by which pin, as shown in Table 5-5.

Table 5-5 Contrast table of which pin is used for connection status indication function of four modules

Function	Module Name	Pin No.	Pin Name
Connection Status Indication Function	HJ-580XP	Pin1 P13	Connect State
	HJ-580CY	Pin15 P21	
	HJ-580LAXP	Pin1 P13	
	HJ-580Nano	Pin11 P11	

5.6 Data arrival prompt

When the module is connected to the mobile phone through Bluetooth, a specific pin of the module will be pulled down when the mobile phone APP sends data to the module. After 10 ms delay, the data received by the module through Bluetooth will be output from the module's serial port. When the data sent by the module from the serial port is finished, the level of the specific pin will become high.

This function is used to give the MCU a hint of data arrival. For devices which requires holding long connections, MCU can be kept in a low power state when there is no data transmission task. When data arrives, MCU can be waked up by level changes (interruptions) of specific pins. The data arrival prompt function of the module is implemented by which pin, as shown in Table 5-6.

Table 5-6 Contrast table of which pin is used for data arrival prompt function of four modules

Function	Module Name	Pin No.
Data arrival prompt Function	HJ-580XP	Pin4 P11
	HJ-580CY	Pin3 P15
	HJ-580LAXP	Pin4 P03
	HJ-580Nano	Pin3 P03

6 Serial Port Instruction Set

6.1 Overview of Instruction Set

Preface: We provide you with an easy to understand instruction set which using ASCII code. You can use the serial port of the computer to connect the module and use the software named “serial port debugging assistant” to debug. You can also use the serial port of the MCU to connect the module and debug.

Classification of instructions: The functions of instructions are divided into three categories: query (read), setting and reply. The instructions sent by MCU to the module mainly include queries and settings. The instructions sent by the module to the MCU are replies.

The conditions of module receives instructions, understands and executes: When MCU sends instructions to modules through serial ports, first of all, ensure that the module is in "full speed operation mode", and at the same time, make sure that the module is in "configuration mode", otherwise the instructions that you send will be invalid.

Format of instructions: All instructions begin with '<' and end with '>'. In the middle of the head and tail symbols, it is the detailed writing of instructions, which is generally composed of capital letters, numbers and underscores. If the format is incorrect or the content is incorrect, the module will reply the error code to the MCU through the serial port, prompting the user to enter the error.

Notes: You can change the parameters of the module when it works. Since DA14580 does not provide an area for storing the user data, all parameters that you set are stored in RAM. Data in RAM will disappear when power fails. So, when you initialize the module, you must first configure the parameters of the module.

6.2 Default parameters of module

After the module has just been powered on or reset, the configuration of the relevant parameters of the module is shown in the following table. Table 6-1 gives the default parameters of the module which in slave mode. Table 6-2 gives the default parameters of the module which in master mode.

Table 6-1 Default parameters for module in slave mode

Parameter Type	Parameter Name	Value
Connection Parameters	Broadcasting Gap	500ms
	Minimum Connection Gap	10ms
	Maximum Connection Gap	10ms
	Connection	5s

	Timeout Time	
Parameters of Serial Port	Serial Port Baud Rate	19200bps
	Check Bit	None
	Stop Bit	1
Broadcasting Data		Thirteen bytes, as follows. 0x02 0x01 0x06 0x03 0x03 0x58 0x69 0x05 0xff 0x01 0x02 0x03 0x05
Ibeacons Data		Thirty bytes, as follows. 0x02 0x01 0x06 0x1A 0xff, /*Apple Pre-Amble*/ 0x4C, 0x00, 0x02, 0x15, /*DeviceUUID(16Bytes)*/ 0x52, 0x41, 0x44, 0x49, 0x55, 0x53, 0x4e, 0x45, 0x54, 0x57, 0x4f, 0x52, 0x4b, 0x53, 0x43, 0x4f, /*Major Value (2 Bytes)*/ 0x00, 0x01, /*Minor Value (2 Bytes)*/ 0x00,0x02, /*Measured Power*/ 0xD2
BLE Module Name	Ten bytes	'H','J',' ','D','A','1','4','5','8','0'

Table 6-2 Default parameters for module in master mode

Parameter Type	Parameter Name	Value
Scanning function parameters and connection parameters	Scanning gap	12.5ms
	Scanning time	10ms
	Minimum connection gap	10ms
	Maximum connection gap	10ms
	Connection timeout time	5s
Parameters of Serial Port	Serial Port Baud Rate	19200bps
	Check Bit	None
	Number of stop bits	1 stop bit

Note: The module does not set the MAC address of the device that needs to be bound and connected, either in slave mode or master mode. Therefore, the same type of modules in the same area may be connected by other devices at any time.

6.3 Instruction Set Explanation

6.3.1 General reply instructions

When the MCU sends instructions to the module, the module may reply to the following instructions, their format and meaning as shown in Table 6-3.

Table 6-3 Format and significance of partial instructions that the module replying to MCU

Instruction	Meaning
<INVALID_ERR>	Instruction does not exist.
<LEN_ERR>	Instructions exist, but the length of instructions is beyond the range.
<HT_ERR>	The format of the head or tail of the instruction is incorrect. The head or tail of the instruction loses '<' or '>'.
<RANGE_ERR>	The instructions are correct, but the values of the parameters exceed the range.
<OK>	The instructions are correct and executed successfully.
<CONNECTED>	The module is connected to external device via Bluetooth.
<DISCONNECTED>	The Bluetooth of the module does not connect to external device.

Note: For the set type of instructions, the module will return <OK> to the MCU after successful settings.

6.3.2 Reset Instruction

Table 6-4 Format and function of reset instruction

Instruction	Function	Response
<RESET>	Reset BLE module	Immediate reset, no return value.

6.3.3 The Instruction of Set Baud Rate

Table 6-5 Format and function of instructions for setting baud rate

Instruction	Function	Response
<BAUD4800>	Set the baud rate to 4800bps.	Save the configuration immediately, but only after the module exits the configuration mode. (Because the reliability of this communication should be guaranteed first.)
<BAUD9600>	Set the baud rate to 9600bps.	It's consistent with the above.
<BAUD19200>	Set the baud rate to 19200bps.	It's consistent with the above.
<BAUD38400>	Set the baud rate to 38400bps.	It's consistent with the above.
<BAUD57600>	Set the baud rate to 57600bps.	It's consistent with the above.
<BAUD115200>	Set the baud rate to 115200bps.	It's consistent with the above.
<BAUD256000>	Set the baud rate to 256000bps.	It's consistent with the above.

6.3.4 The Instruction of Read Baud Rate

Table 6-6 Format and function of instructions for read baud rate

Instruction	Function	Response
<COMBAUD>	Read the current baud rate	The format of the returned instruction is "<baud rate value>". For example, if the current baud rate is 19200bps, the module will return <19200> immediately.

6.3.5 Set the number of serial port stop bits

Table 6-7 Format and function of instructions for set the number of serial port stop bits

Instruction	Function	Response
<1SB>	Set the number of stop bits for serial port to be 1.	Save the configuration immediately, but only after the module exits the configuration mode. (Because the reliability of this communication should be guaranteed first.)
<2SB>	Set the number of stop bits for serial port to be 2.	It's consistent with the above.

6.3.6 Read the number of serial port stop bits

Table 6-8 Format and function of instructions for read the number of serial port stop bits

Instruction	Function	Response
<STOPBIT>	Read the number of stop bits currently used in the serial port	When the number of stop bits in the serial port is 1, the module return <STOPBIT1>. When the number of stop bits in the serial port is 2, the module return <STOPBIT2>.

6.3.7 Set the type of serial port verification

Table 6-9 Format and function of instructions for set the type of serial port verification

Instruction	Function	Response
<PEVEN>	Set the verification type of serial port to even parity check.	Save the configuration immediately, but only after the module exits the configuration mode. (Because the reliability of this communication should be guaranteed first.)
<PODD>	Set the verification type of serial port to odd check.	It's consistent with the above.
<PNO>	Set the verification type of serial port to no check.	It's consistent with the above.

Note: The MCU which connected to the module through serial port must have the same type of serial port verification bits with the module, otherwise they can not communicate normally.

6.3.8 Read the type of serial port verification

Table 6-10 Format and function of instructions for read the type of serial port verification

Instruction	Function	Response
<PARITY>	Read the verification type of serial port.	If the verification type of serial port is no check, the module return <NOP>. If the verification type of serial port is even parity check, the module return <EVENP>. If the verification type of serial port is odd check, the module return <ODDP>.

6.3.9 Broadcasting Start or Stop

Table 6-11 Format and function of instructions for broadcasting start or stop

Instruction	Function	Response
<STOPADV>	Turn off the broadcast function of the module.	Implement immediately.
<STARTADV>	Turn on the broadcast function of the module. (Broadcasting function is turned on by default after module is powered on.)	Implement immediately.

6.3.10 Read the status of broadcasting function

Table 6-12 Format and function of instructions for read the status of broadcasting function

Instruction	Function	Response
<ADVSTATE>	Read the status of broadcasting function.	If the broadcasting function of the module is turned on, the module return <ADVON>. If the broadcasting function of the module is turned off, the module return <ADVOFF>.

When the broadcasting function is turned on, the module will broadcast according to the broadcasting gap you set. The BLE master or mobile APP can scan and search the module, and you can connect.

If the broadcasting function is turned off, the module will enter the low-power consumption mode, not broadcasting outside, at this time the power consumption of the module is not more than 2uA. However, the module can still be wake up by external devices and perform other operations.

6.3.11 Set the data for general broadcasting

Table 6-13 Format and function of instructions for set the data for general broadcasting

Instruction	Function	Response
<ADVDATAxxxxxxxxxx>	Set the broadcasting data of the module, x is the data that you want to set, broadcasting data can set up to 22 bytes of characters.	The broadcasting data is immediately saved by the module. After the next broadcasting cycle begins, the broadcasting data will use the latest data saved by the module.

In the instructions which shown in Table 6-13, the lowercase x represents the broadcasting data that you want to write. Due to the limitation of BLE protocol, this instruction supports up to 22 bytes of broadcasting data. If the length of the broadcasting data that you want to write exceeds 22 bytes, the module will return the error information to the MCU.

6.3.12 Set the data for Ibeacons broadcasting

Table 6-14 Format and function of instructions for set the data for Ibeacons broadcasting

Instruction	Function	Response
<IBACONxxxxxxxxxx>	Set the Ibeacons broadcasting data of the module, x is the data that you want to set, Ibeacons broadcasting data can set up to 25 bytes of characters.	The Ibeacons broadcasting data is immediately saved by the module. After the next broadcasting cycle begins, the Ibeacons broadcasting data will use the latest data saved by the module.

In the instructions which shown in Table 6-14, if the length of the Ibeacons broadcasting data that you want to write exceeds 25 bytes, the module will return error information to the MCU.

6.3.13 Set device name

Table 6-15 Format and function of instructions for set device name

Instruction	Function	Response
<NAMExxxxxx>	Set the device name of the module. x is the name of the device that you want to set, up to 18 bytes of characters.	If the module has been connected to other devices through Bluetooth when it changes the device name, this setting will take effect after the next reconnection. If Bluetooth does not connect to any device when the module changes the device name, this setting take effect immediately.

6.3.14 Read device name

Table 6-16 Format and function of instructions for read device name

Instruction	Function	Response
<MNAME>	Read the device name.	If the device name is DA14580, the module returns <DA14580>.

6.3.15 Set the name of manufacturer

Table 6-17 Format and function of instructions for set the name of manufacturer

Instruction	Function	Response
<FACxxxxxx>	Set the name of the manufacturer of the module. x is the name of the manufacturer that you want to	If the module has been connected to other devices through Bluetooth when it changes the name of the manufacturer, this setting will take effect after the next reconnection.

	set, up to 18 bytes of characters.	If Bluetooth does not connect to any device when the module changes the name of the manufacturer, this setting take effect immediately.

6.3.16 Read the name of manufacturer

Table 6-18 Format and function of instructions for read the name of manufacturer

Instruction	Function	Response
<FNAME>	Read the name of the manufacturer.	If the manufacturer name is “HongJia”, the module will return <HongJia>.

6.3.17 Set Software Version

Table 6-19 Format and function of instructions for set software version

Instruction	Function	Response
<SOFTxxxxxx>	Set the software version of the module. x is the software version that you want to set, up to 18 bytes of characters.	If the module has been connected to other devices through Bluetooth when it changes the software version, this setting will take effect after the next reconnection. If Bluetooth does not connect to any device when the module changes the software version, this setting take effect immediately.

6.3.18 Read Software Version

Table 6-20 Format and function of instructions for read software version

Instruction	Function	Response
<SVER>	Read software version.	If the software version is VER1.1, the module will return <SVER1.1>.

6.3.19 Set Hardware Version

Table 6-21 Format and function of instructions for set hardware version

Instruction	Function	Response
<HARDxxxxxx>	Set the hardware version of the module. x is the hardware version that you want to set, up to 18 bytes of characters.	If the module has been connected to other devices through Bluetooth when it changes the hardware version, this setting will take effect after the next reconnection. If Bluetooth does not connect to any device when the module changes the hardware version, this setting take effect immediately.

6.3.20 Read Hardware Version

Table 6-22 Format and function of instructions for read hardware version

Instruction	Function	Response

<HVER>	Read hardware version.	If the hardware version is VER1.0, the module will return <HVER1.0>.

6.3.21 Set the SN number of the module

Table 6-23 Format and function of instructions for set the SN number of the module

Instruction	Function	Response
<SNxxxxxx>	Set the SN number of the module. x is the SN nmuber of the module that you want to set, up to 18 bytes of characters.	If the module has been connected to other devices through Bluetooth when it changes the SN number, this setting will take effect after the next reconnection. If Bluetooth does not connect to any device when the module changes the SN number, this setting take effect immediately.

6.3.22 Read the SN number of the module

Table 6-24 Format and function of instructions for read the SN number of the module

Instruction	Function	Response
<MSN>	Read the SN number of the module.	If the SN number is 56789, the module will return <N56789>.

6.3.23 Set Broadcasting Gap

Table 6-25 Format and function of instructions for set broadcasting gap

Instruction	Function	Response
<ADVGAP32> to <ADVGAP16000>	Set the broadcasting gap, the numerical range is 32 ~ 16000, corresponding the time is 20ms ~ 10s.	Implement immediately.

6.3.24 Read Broadcasting Gap

Table 6-26 Format and function of instructions for read broadcasting gap

Instruction	Function	Response
<AGAP>	Read broadcasting gap.	If the broadcasting gap is 32, the module will return <A32>.

6.3.25 Set Maximum Connection Gap

Table 6-27 Format and function of instructions for set maximum connection gap

Instruction	Function	Response
<CONMAX6> to <CONMAX3200>	Set the maximum connection gap, the numerical range is 6~3200, corresponding the time is 7.5ms ~ 4s.	Implement immediately.

Note: The maximum connection gap should be within the prescribed range. And the maximum connection gap must be not less than the minimum connection gap.

6.3.26 Read Maximum Connection Gap

Table 6-28 Format and function of instructions for read maximum connection gap

Instruction	Function	Response
<MAXCGAP>	Read maximum connection gap.	If the maximum connection gap is 600, the module will return <CX600>.

6.3.27 Set Minimum Connection Gap

Table 6-29 Format and function of instructions for set minimum connection gap

Instruction	Function	Response
<CONMIN6> to <CONMIN3200>	Set the minimum connection gap, the numerical range is 6~3200, corresponding the time is 7.5ms ~ 4s.	Implement immediately.

Note: The minimum connection gap should be within the prescribed range. And the minimum connection gap must be not more than the maximum connection gap

6.3.28 Read Minimum Connection Gap

Table 6-30 Format and function of instructions for read minimum connection gap

Instruction	Function	Response
<MINCGAP>	Read minimum connection gap.	If the minimum connection gap is 60, the module will return <CN60>.

6.3.29 Set connection timeout time

Table 6-31 Format and function of instructions for set connection timeout time

Instruction	Function	Response
<TIMEOUT10> to <TIMEOUT3200>	Set the connection timeout time, the numerical range is 10~3200, corresponding the time is 100ms ~ 32s.	Implement immediately.

Note: If the device's system is IOS or Android, the connection timeout time should not exceed 6 seconds.

6.3.30 Read connection timeout time

Table 6-32 Format and function of instructions for read connection timeout time

Instruction	Function	Response
<CTIMEOUT>	Read connection timeout time.	If the connection timeout time is 100, the module will return <T100>.

6.3.31 Disconnect the current connection immediately

Table 6-33 Format and function of instructions for disconnect the current connection immediately

Instruction	Function	Response
<DISCONNECT>	Disconnect the current connection immediately.	If the Bluetooth of the module is connected to external device, the module will be disconnected immediately after sending this instruction. If the Bluetooth of the module is not connected to any device, the module will return <DISCONNECTED> after sending this instruction, informing the customer that the module is not connected to any external device.

6.3.32 Query Bluetooth Connection Status

Table 6-34 Format and function of instructions for query Bluetooth connection status

Instruction	Function	Response
<STATE>	Query the current Bluetooth connection status.	If the module is connected to external device, the module return <CONNECTED>. If the module dose not connected to external device, the module return <DISCONNECTED>.

6.3.33 Query the Working Mode of Bluetooth Device

Table 6-35 Format and function of instructions for query the working mode of Bluetooth device

Instruction	Function	Response
<ROLETYPE>	Query whether the current Bluetooth module works in master mode or slave mode.	If the module is in master mode, the module return <CENTER>. If the module is in slave mode, the module return <SLAVER>.

6.3.34 Read the MAC address of this machine

Table 6-36 Format and function of instructions for read the MAC address of this machine

Instruction	Function	Response
<MAC>	Read the current MAC address of the module. (MAC address fixed to 6 bytes of characters)	For example, if the current MAC address of the module is 123456, the module will return < 123456>.

6.3.35 Read the MAC address of the device which connected to the module's Bluetooth

Table 6-37 Format and function of instructions for read the MAC address of the device which connected to the module's Bluetooth

Instruction	Function	Response
<PEERMAC>	Read the MAC address of the device which connected to the module's Bluetooth. (MAC address fixed to 6 bytes of characters)	For example, if the MAC address of the device which connected to the module's Bluetooth is 123456, the module will return <P123456>. If the module dose not connected to external device, the module return <DISCONNECTED>.

Note: Whether the module works in master mode or slave mode, this instruction will query the MAC address of the device which connected to Bluetooth of the module.

6.3.36 Set the MAC address of the device which to be bound

Table 6-38 Format and function of instructions for set the MAC address of the device which to be bound

Instruction	Function	Response
<BONDMACXXXXXX>	Set the MAC address of the device which to be bound. (MAC address fixed to 6 bytes of characters)	For example, if the MAC address of the master or slave that needs to be bound is 123456, then the module will return <BONDMAC123456>.

Note:

Whether the module works in master mode or slave mode, the MAC address that you set according to this function is valid.

When the module works in slave mode, when the master is connected to the slave, the slave checks whether the master's MAC address is the address that we set. If yes then connect. If not, disconnect directly.

When the module works in the master mode, the master scans to determine whether the MAC address of the nearby device is the set MAC address. If so, connect. If not, do nothing and continue scanning.

The default behavior of a device is that does not connect to the bound MAC address after the module is powered on. That is to say, after the module is powered on, all single models can be connected to each other. The purpose of setting the bound MAC address for Bluetooth module is to prevent interference between modules.

Additional explanation: For example, there are 100 Bluetooth modules in a region. Assuming that a master already knows the MAC addresses of other slaves, the master can bind any slave and communicate.

6.3.37 Read the MAC address of the device which to be bound

Table 6-39 Format and function of instructions for read the MAC address of the device which to be bound

Instruction	Function	Response
<RBMAC>	Read the MAC address of the device which to be bound. (MAC address fixed to 6 bytes of characters)	For example, if we set the MAC address of the device that to be bound is 123456, the module will return <BONDMAC123456>.

6.3.38 Unbind

Table 6-40 Format and function of instructions for unbind

Instruction	Function	Response
<DISBOND>	Cancel the MAC address of the device that needs to be bound.	When the setup is successful, the module return <OK>.

Note: After unbinding, at the next time when the module connects, the module will not determine the MAC address of the device that needs to be connected.

6.3.39 Stop scan

Table 6-41 Format and function of instructions for stop scan

Instruction	Function	Response
<STOPSCAN>	Set the module to stop scanning.	Refer to "Notes" in this section.

Note:

This instruction is only valid if the module is in master mode and Bluetooth of the module is not connected to other devices.

When the module is not connected to other slaves, after sending this instruction to the module, the module that in master mode executes the instruction and stops scanning. When the module has been connected to the external slave as the master, the module will return <CONNECTED> when this instruction is sent to the module. This reply instruction is intended to inform the MCU that the module has been connected to the external slave and cannot execute this instruction. Please set it after the module disconnects.

When the master stops scanning, the BLE module will be in a silent state, which is the lowest power consumption state of the module in the master mode.

6.3.40 Start Scan

Table 6-42 Format and function of instructions for start scan

Instruction	Function	Response
<STARTSCAN>	Set the module to start scanning.	Refer to "Notes" in this section.

Note:

This instruction is only valid if the module is in master mode and Bluetooth of the module is not connected to other slaves.

When the module is not connected to the external slave, after sent this instruction to the module, the module executes the instruction and starts scanning. When the module has been connected to the external slave, after sent this instruction to the module, the module will return <CONNECTED>. This reply instruction is intended to inform the MCU that the module has been connected to the external slave and cannot execute this instruction. Please set it after the module disconnects.

6.3.41 Read the state of scan

Table 6-43 Format and function of instructions for read the state of scan

Instruction	Function	Response
<SCANSTATE>	Read the state of scan.	When the scanning function of the module is turned on, the module return <SCANON>. When the scanning function of the module is turned off, the module return <SCANOFF>.

6.3.42 Enable the function of automatic upload the connection parameters

Table 6-44 Format and function of instructions for enable the function of automatic upload the connection parameters

Instruction	Function	Response
<ONUPDATE>	Enable the function of automatic upload the connection parameters.	The module turn on the function of automatic upload the connection parameters by default. Moreover, this function is only valid when the module is in slave mode. When the module executes this instruction, each time the module is connected by the master, it uploads its own connection parameters to the master for reference. After the master accepts this parameter, both sides will connect according to the uploaded parameters.

6.3.43 Disable the function of upload the connection parameters

Table 6-45 Format and function of instructions for disable the function of upload the connection parameters

Instruction	Function	Response
<OFFUPDATE>	Disable the function of upload the connection parameters.	When the module executes this instruction, the module will not upload the connection parameters of the module to the master after each connection. Then the two sides will communicate according to the connection parameters of the master itself.

6.3.44 Read the status of connection parameter upload function

Table 6-46 Format and function of instructions for read the status of connection parameter upload function

Instruction	Function	Response
<UPDATESTATE>	Read the status of connection parameter upload function.	If the module opens the function of connection parameter automatic upload, the module return <UPON>. If the module turn off the function of connection parameter automatic upload, the module return <UPOFF>.

6.3.45 Read the RSSI value of the current connection

Table 6-47 Format and function of instructions for read the RSSI value of the current connection

Instruction	Function	Response
<RSSI>	Read the RSSI value of the current connection.	If the Bluetooth of the module has connected to the external device, the module directly returns the RSSI value of the current connection. If the Bluetooth of the current module is not connected to the external device, the RSSI reading instruction is meaningless and the module returns <DISCONNECTED>. The reply instruction is intended to inform the user that the module is not connected to the external device.

Note:

The format of the RSSI value returned is <Rx>. x represents the RSSI value of the current connection, fixed to four bytes. The value is 8-bit signed data, and the highest bit represents the symbol. This value is not sent in character form, but in hexadecimal format(HEX).

6.3.46 In slave mode, disconnect and stop broadcasting

Table 6-48 Format and function of instructions for disconnect and stop broadcasting in slave mode

Instruction	Function	Response
<DIS-STOPADV>	The module is in slave mode, disconnects from the master and stops broadcasting.	When the module has been connected to the master as the slave, when sending this instruction to the module, the module actively disconnects from the master and stops broadcasting. The module enters the low power standby mode.

Note: The main purpose of adding this instruction is to facilitate user's operation. A specific function can be achieved by using one instruction. Because in order to make the module in slave mode enter the low power state, it is necessary to disconnect the connection and turn off the broadcasting first.

6.3.47 In master mode, disconnect and stop broadcasting

Table 6-49 Format and function of instructions for disconnect and stop broadcasting in master mode

Instruction	Function	Response
<DIS-STOPSCAN>	The module is in master mode, disconnects from the slave and stops broadcasting.	When the module has been connected to the slave as the master, when sending this instruction to the module, the module actively disconnects from the slave and stops broadcasting. The module enters the low power standby mode.

Note: The main purpose of adding this instruction is to facilitate user's operation. A specific

function can be achieved by using one instruction. Because in order to make the module in master mode enter the low power state, it is necessary to disconnect the connection and turn off the broadcasting first.

6.3.48 Restore factory settings

Table 6-50 Format and function of instructions for restore factory settings

Instruction	Function	Response
<FACTORY>	Serial port parameters and connection parameters are restored to factory settings.	Effective immediately. After the instruction is executed, the wireless parameters of the module are restored to the factory settings.

Table 6-51 After restoring the factory settings, the wireless parameters of the module

Name	Value
Serial Port Baud Rate	19200bps
Serial Port Stop Bit	one
Serial port check bit	none
Broadcasting gap	500ms
Maximum connection gap	10ms
Minimum connection gap	10ms
Connection timeout time	5s

6.3.49 Enable feedback status prompt function

We define the feedback status prompt function, which is mainly used to feedback the status of the current module to the MCU. For example, when the module goes into dormancy, wake-up, goes into configuration mode or serial port transparent transmission mode, the serial port of the module will automatically feedback the status of the current module to the MCU.

Table 6-52 Format and function of instructions for enable feedback status prompt function

Instruction	Function	Response
<WSMON>	Enable feedback status prompt function.	Effective immediately.
<WSMOFF>	Disable feedback status prompt function.	Effective immediately.

Note: When this function is enabled, in the following four cases, the serial port will automatically send feedback information to the device(Generally MCU) which connected to the module's serial port.

1. When the module is in dormant state, the serial port prints <SLEEP> to notify MCU that the module is in dormant state.

2. When the module wakes up and enters full-speed operation mode, the serial port prints <WAKEUP> in order to inform MCU that the module has entered full-speed operation mode.

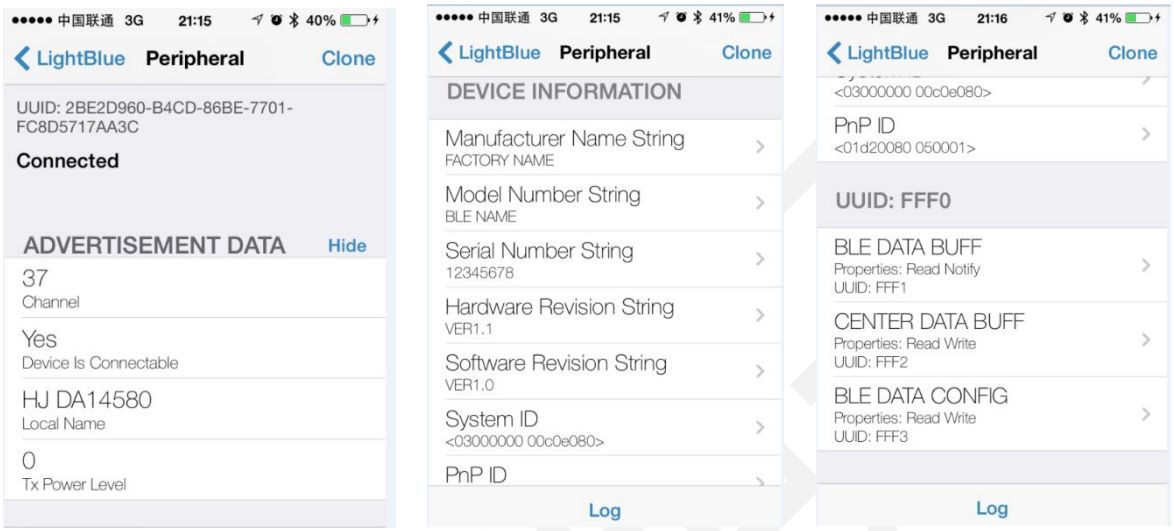
3. When the module enters the configuration mode, the serial port prints <CONFIG MODE> to notify MCU that the module enters the configuration mode.

4. When the module exits the configuration mode and enters the data transparent transmission mode, the serial port prints <DATA MODE> in order to inform MCU that the module enters the data transparent transmission mode.

7 Definition of UUID

Debugging environment: Mobile APP or BLE module as master connects to module and communicates.

When you use the Bluetooth of your mobile phone to connect to the module, here we use the LightBlue software in IOS, the information displayed by the software is shown in the figure 7-1.



(a) (b) (c)

Figure 7-1 Software schematic of LightBlue

For Figure 7-1 (a): This is the data scanned by Bluetooth, including device name, broadcasting channel and transmission power.

For Figure 7-1 (b): This is the device information of BLE module, including software version number, hardware version number, device name, manufacturer name, device ID, device MAC address and PnP ID code. (This information can be modified according to instructions. And you should pay attention to the notes in Section 6.1.)

For Figure 7-1 (c): This is the data channel schematic diagram of the device. The UUID of the main service is 0XFFF0. There are three sub-services under the main service: BLE data channel 0XFFF1, APP or master data channel 0XFFF2 and the data channel 0XFFF3 of configuration mode.

Table 7-1 gives a detailed definition of UUID.

Table 7-1 Definition of UUID

Channel Name	UUID	HANDLE Value	Service attribute	Remark
BLE DATA BUFF	0XFFF1	0X21	Read only, Notify	Note 1
	0XFFF1	0X22	Read or Write	Note 2
CENTER DATA BUFF	0XFFF2	0X25	Read or Write	Note 3
BLE DATA CONFIG	0XFFF3	0X28	Read or Write	Note 4

Note 1:

“BLE DATA BUFF” is the data transparent transmission channel for the serial port of BLE module. That is to say, the data received by the module through the serial port will be returned to

the data channel which UUID is 0XFFF1 in the form of notification, and the mobile phone APP or BLE master can received the data. (The premise of using this function is that the notification function should be turned on.) The maximum length of data bytes for a single notification is 20 bytes. If more than 20 bytes of data are sent by notification, the module will automatically subcontract and send.

Note 2:

The name of this data is BLE DATA BUFF (BLE Data Channel) notification enable bit. If you use an IPhone or Android phone as the master connect to module. Then you need to input the notification enable function into the channel whose UUID is 0XFFF1 to turn on the notification function.

Note 3:

“CENTER DATA BUFF” is the channel used by the mobile phone APP or BLE master to send data to the module. The maximum byte length of the single data transmission is 20 bytes. If you need to send more than 20 bytes of data, send it in multiple packets. When the module receives the data, it will send it to the external device through the TX pin of its serial port one after another.

Note 4:

This channel is not in use and has no function.