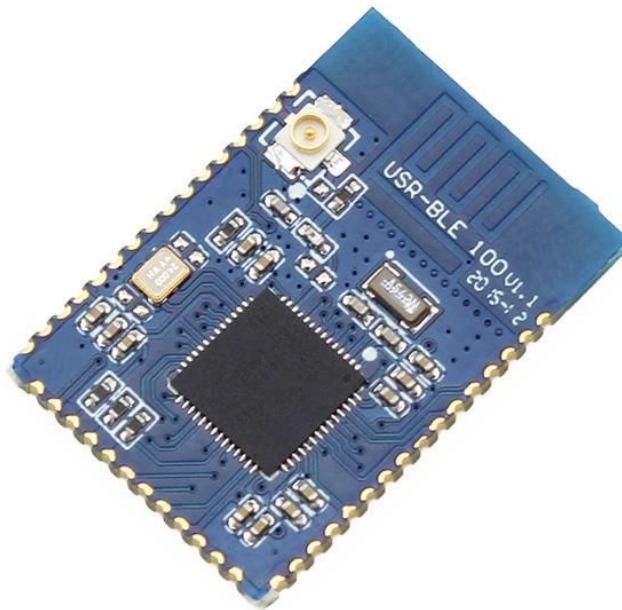


USR-BLE100 Design Manual

File Version: V1.1



Contents

1. Product Overview.....	4
1.1. Brief Introduction.....	4
2. Product Function.....	4
2.1. Configuration process.....	4
2.2. Work Mode.....	5
2.2.1. Master Mode.....	5
2.2.2. Slave Mode.....	6
2.2.3. Broadcast Mode.....	6
2.2.4. Mesh Network Mode.....	7
2.3. Packet Mechanism.....	7
2.4. iBeacon Protocol.....	8
2.5. Low Power Mode.....	9
2.6. Password Pairing Authentication.....	10
2.7. Encrypted Transmission.....	10
2.8. Battery Power Measure.....	10
2.9. Firmware Update.....	11
2.10. Indicator Status.....	12
3. Setup Method.....	13
3.1. AT Instruction.....	13
3.1.1. Serial AT Instruction.....	13
3.1.2. Transparent Transmission Instruction.....	13
3.1.3. AT Instructions.....	13
3.1.3.1. AT+NAME.....	14
3.1.3.2. AT+MODE.....	15
3.1.3.3. AT+MAC.....	15
3.1.3.4. AT+CIVER.....	15
3.1.3.5. AT+TPL.....	15
3.1.3.6. AT+SHOW.....	16
3.1.3.7. AT+PASS.....	16
3.1.3.8. AT+PASSEN.....	17
3.1.3.9. AT+UART.....	17
3.1.3.10. AT+UARTTM.....	18
3.1.3.11. AT+AUTOSLEEP.....	18
3.1.3.12. AT+DEEPSLEEP.....	18
3.1.3.13. AT+HIBERNATE.....	18
3.1.3.14. AT+HELLO.....	19
3.1.3.15. AT+ENTM.....	19
3.1.3.16. AT+RELOAD.....	19
3.1.3.17. AT+Z.....	19
3.1.3.18. AT+LINK.....	19
3.1.3.19. AT+SCAN.....	20

3.1.3.20. AT+CONN.....	20
3.1.3.21. AT+CONNADD.....	20
3.1.3.22. AT+DISCONN.....	21
3.1.3.23. AT+ADP.....	21
3.1.3.24. AT+ADPTM.....	21
3.1.3.25. AT+BATEN.....	21
3.1.3.26. AT+MAXPUT.....	22
3.1.3.27. AT+TRENC.....	22
3.1.3.28. AT+IBEAON.....	23
3.1.3.29. AT+ADVDATA.....	23
4. Contact US.....	23
5. Disclaimer.....	23
6. Update History.....	24

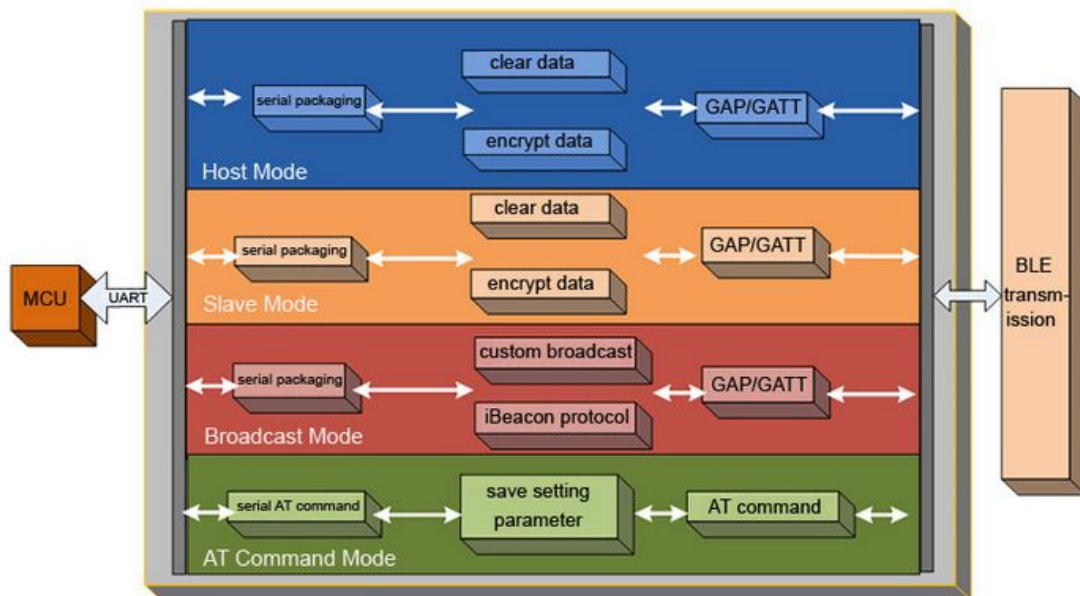
1. Product Overview

1.1. Brief Introduction

USR-BLE100 is an ultra-low power 4.1v Bluetooth module. It is used for point-to-point data transparent transmission. User can communicate immediately after simple setup without concerning for transport protocol. This module support master and slave mode, user not only can use both master and salve for transparent transmission, also can develop APP on the basis of standard BLE protocol. Module support one-to-many broadcast, and has built-in iBeacon protocol. User can make it a iBeacon equipment with simple setup.

2. Product Function

This chapter introduce the functions of module, as the following diagram shown, you can get an overall knowledge of it.



2.1. Configuration process

Power on, module will get into working mode according to user settings in advance, and open UART with pre-set parameters.

Preset parameters:

- ❖ Work mode
 - Master mode
 - Slave mode
 - Broadcast mode

- Mesh networking mode
- ❖ Default connection parameters:
 - The MAC address of device which module will connect to
- ❖ UART parameters
 - Baud rate
 - Data bit
 - Parity bit
 - Stop bit

Module will work according to pre-set parameters after reset. Below chapters will introduce each part in detail.

2.2. Work Mode

There are 4 working modes in total: Master mode, Slave mode, Broadcast mode and Mesh networking mode

■ Master Mode

USR-BLE100 support master mode that can connect with a slave device. In this pattern, user can search devices around and choose the slave device need to connect. User also can set the MAC address of the slave for default link, the device can find this pattern to connect when it power on. With white list function, user can write the MAC address of device which need connection into white list, then module will search the device in the list and connect.

■ Slave Mode

USR-BLE100 support slave mode, in this mode, device completely conform to BLE 4.1 protocol and compatible with 4.0 protocol, user can develop APP according to protocol. This mode contain a serial transceiver Service, user can find it by UUID. There are 2 channels, read and write, user can transfer data through those channels.

There is no need for user to concern protocol if use the master of USR-BLE100 connect with slave, the serial ports of two devices can transparent transfer data immediately. It develop a easy wireless transmission channel for user.

■ Broadcast Mode

USR-BLE100 support broadcasting mode, which support one-to-many broadcast. User can set module broadcasting data through AT instructions. Module can consistent broadcast in low-power pattern, be used to ultra-low power, small-data-sets, simplex transmission application occasions, like wireless measurement, indoor location and so on.

■ Mesh Network Mode

USR-BLE100 support Mesh network mode, can mesh many modules with net easily. Set each module with the same password, they will join in the same network. Each module can send data, each module can receive data and reply. With no need for gateway, even if one device broken, device will skip it and choose nearest device to transfer.

2.2.1. Master Mode

In this mode, user can set module search slave devices around and connect with devices quickly. If need transparent transmission, also need a module in slave mode.

No need to concern the converter process between serial data and wireless data packet, just with simple

settings, module can realize transparent transmission for UARTs of master device and slave device.

Three method for master to connect with slave device:

Search. Need to set below parameters in advance:

1) Set work mode to be master

AT+MODE=M

2) Open search function

AT+SCAN

3) If scanned and found slave device 1, can use fast connect command to connect

AT+CONN=1

4) After settings, module indicator will be ON, this indicate the connection succeed. Now the UARTs can communicate.

If you know the MAC address of the slave device which need to connect to, see below steps:

1) Bind the MAC address to AT command, set module connect the default MAC address of slave device after power on

AT+CONNADD=FFFFFFFFF11

2) Reset the module, it will connect to the set MAC address

AT+Z

2.2.2. Slave Mode

In this mode, user need to set module into slave mode. If you want to develop APP by yourself and need our UUID, UUID is: 0x31, 0x01, 0x9b, 0x5f, 0x80, 0x00, 0x00,0x80, 0x00, 0x10, 0x00, 0x00, 0xd0, 0xcd, 0x03, 0x00. We also provide the sample application.

1) Set work mode to be slave

AT+MODE=S

2) Can check the module connection status by command:

AT+LINK

3) Can disconnect by command:

AT+DISCONN

4) If user don't want the module be found and connected, use below command to close broadcast data:

AT+ADP=OFF

2.2.3. Broadcast Mode

In this mode, module can broadcast small amount of data. User need to invoke standard BLE interface in APP for data obtaining. Use AT command to set data in advance.

1) Set module to be broadcast mode

AT+MODE=B

2) Set the data that module will send, data bit in Hex and less no more than 30 bytes, broadcast format pls refer to BLE protocol. **AT+ADVDATA =0201041Aff4c000215B9407F30F5F8466262626288777755552255**

Through the bluetooth monitoring software, can get the monitor data packet as follows:

```
Raw Data
02:01:04:1A:FF:4C:00:02:15:B9:40:7F:30:F5:F8:46:6E:AF:F9:25:55:6B:57:FE:6D:00:01:00:02:51
```

Will introduce iBeacon function in detail in later chapter.

2.2.4. Mesh Network Mode

USR-BLE100 support Mesh network mode, can connect many modules with net easily. By using star network and relay technology, each network can link 65000 or more nodes, network can be interconnected, and countless can be interconnected or manipulate directly through mobile phone and pad. It doesn't need gateway, even if one device broken, device will skip it and choice nearest device to transfer. The whole networking process just need a power-on device and communication password.

1) Switch work mode to be mesh networking

AT+MODE=F

2) Set communicate password, module connection will distinguish by password

AT+PASS=123456

3) Reset module to get into mesh mode

AT+Z

4) Now we set many modules the same as above steps, when a module send data, the modules around it will receive, transmit to UART port, send data to modules around them and by this analogy

5) When a module received data and need reply, data will be sent by UART. The first module which send data will receive its reply, this process is internal network communication.

<Note>

Because of the principle of meshing, the data sent each time should be no more than 20 bytes. The transport process will delay 1s to 10s after each device. And can't ensure the data arrived accurately. Pls adjust according to your application environment.

2.3. Packet Mechanism

Low-power Bluetooth aimed at small-data-sets transmission. According to the different serial baud rate, BLE connect intervals and different contracting intervals, module will has different capacity of data, but, in any case it will not exceed the theory of 4K byte/s. Test shows that leakage probability is a bit lower when forwarding rate is below 2K byte/s. To be safe, whether high speed or low speed, it's better to do check retransmission in the upper.

The serial port of USR-BLE100 can receive up to 512-byte packets per time, module will send automatically according to the size of the packet, the maximum load for each wireless packet is 20 bytes according to the standard. The packets mobile device send to module must packet(1-20byte per package) by self to transfer. Module will forwarding to module serial port in turn when receive.

Module has two contract award pattern, one is 20 byte per package, another is 100 byte per package. The latter can only be applied to data transmission based on our module one as master and one as slave.

If user use APP communicate with slave should choice 20 byte per package, there is specialized AT instruction to be set.

AT+MAXPUT=OFF

If one as master another as slave to transmit, set it on:

AT+MAXPUT=ON

Can set UART packet time, default is 100ms, user can set this time according to your application to avoid continuous packet:

AT+UARTTM=200

2.4. iBeacon Protocol

USR-BLE send specific ID to surrounding through BLE broadcasting technology, the software received this ID will take action according this ID. For example, set iBeacon protocol in shop will make the information in iPhone and iPad transfer to the server, or make server send coupon and points to customer. In addition, user can use iBeacon to send information to software when home appliance broken or stop working. Now WeChat also support iBeacon, when user use “shake”, the ID registered in the background will receive related information. The Bluetooth pendant key in popular use APP and the gyroscope of mobile phone to locate module, always be used to indoor location.

iBeacon protocol is composed of 4 parameters: UUID, Major, Minor, TX power

UUID: this ID is used to distinguish your beacon devices with others' beacon devices. Example: In a certain area of a store, there exists multi beacons, these form a “Chain Belt”, which is used to provide specific service. Then, all these beacons belong to the same “Chain Belt” will be distributed with same UUID. The special application which is designed for the “Chain Belt” will use this UUID in background to scan and obtain the beacon devices in this “Chain Belt”

major code: used to identified the relevant beacons as a group. Example: All beacon in a store will be distributed with the same major number. By this way, application will know which store is customer in.

minor code: used to identify specific beacon device. Example: In a store, each beacon device has an unique minor code. By this way, you can know where is the customer.

TX power: used to know how far is your distance with beacon. It is the RSSI for 1M from the device. If the signal is weaker, we are away from. Just need know the RSSI for 1M and current RSSI(which can be obtained from the signals received), then it is possible to calculate current distance.

USR-BLE100 provide simple command to set:

1) Set work mode to be broadcast

AT+MODE=B

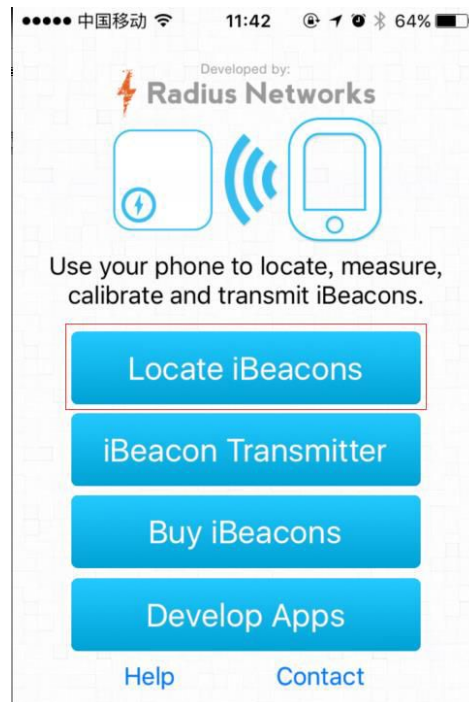
2) Set the necessary parameters for iBeacon protocol

AT+IBEACON=B9407F30F5F8466EAFF925556B57FE6D,1,1,175

3) Reset module

AT+Z

4) After setting, we should test it with a common APP, with IOS system, application: Locate Beacon
Open bluetooth function in phone, and start APP



5) Select Locate iBeacons function and open it. If module working smoothly, it will show the module location and current distance.



6) The distance will change when the cell phone and module distance differently. when user leave or enter again the identified range, it will receive the push information

7) This is a debug software only, user need to develop your own application by yourself.

2.5. Low Power Mode

USR-BLE100 has many low power pattern, can provide longer using time for the customer use battery.

First pattern is auto sleep mode, user can set the time begin sleep through AT instruction in this pattern. When no connection ,serial port has no data and standby time exceeds the set time, module will enter sleep mode automatically. In this pattern module Bluetooth can broadcast, connect and serial can awake module anytime. Set by command as follows:

AT+AUTOSLEEP=ON,4

Second pattern is deep sleep pattern, enter this pattern need AT instruction. Then, Bluetooth still can broadcast and connect. There are two awake pattern, make connection and use awake gin. In this pattern, power consumption can reach about 1.5uA , many slave work in this pattern. Set by command below:

AT+DEEPSLEEP

Third pattern is sleep pattern. This pattern still need AT instruction to enter, any peripheral will not work in this pattern, power consumption can reach about 150nA, user can use awake gin awake it.

AT+HIBERNATE

<Note>

The low-power consumption here is only relative to salve, master does not have low-power. User can make it stop working when master does not work, awake it and search to make connection when user need.

2.6. Password Pairing Authentication

To guarantee the safety of device, USR-BLE100 provides password matching mechanism, this need to be open by AT command. When open, if cell phone or other devices search and connect to module, need to write the 6 bytes password in 10s by Write service of transmission, then they can connect and transmit. If password incorrect or timeout, link will disconnect. If use our master connect our slave device, just need to send 6 bytes password by UART after connection.

Open steps:

AT+PASSEN=ON

2.7. Encrypted Transmission

In order to ensure the security of data transmission, user can choice plaintext transmission or encrypted transmission. User can use AT instruction to encrypted transfer, then serial data will be PC1 encrypted by password, the pattern receive data can decrypt only when it has password. It ensure the safety of data transmission. Master-Slave transmission for our module, just set as below:

AT+TRENC=ON

Set password of two modules the same

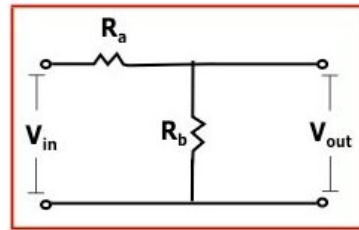
AT+PASS=123456

When connection ok for data transmission, the data received by UART will be encrypted, and be un-encrypted before sending to UART. This will avoide being obtained by others.

2.8. Battery Power Measure

USR-BLE100 collecting voltage through built-in ADC, transfer through specific "Service ". Considering many customer use Bluetooth will use battery in most cases, we add a measure function to access battery power in real time. User only need to part pressure on the battery, part 1.024V voltage to collection gin when battery in maximum power. Module monitor the change of voltage in real time, and report through private channel.

For example, our button battery voltage is 3V, so we need to use 2 resistors to get 1.024 voltage. No need to be so accurate, just close to it. Then we can get a contrast voltage, connect this voltage to module electricity measurement pin to obtain battery electricity quantity.



V_{in} = 输入电压
 V_{out} = 输出电压

输入电压 V_{in} : V
 串联电阻 R_a : Ω
 并联电阻 R_b : Ω
 输出电压 V_{out} : V

According to the different requirements of users, you need to open this function by command:

AT+BATEN=ON

Then we can query the electricity by AT command, or read the Service of electricity in APP.

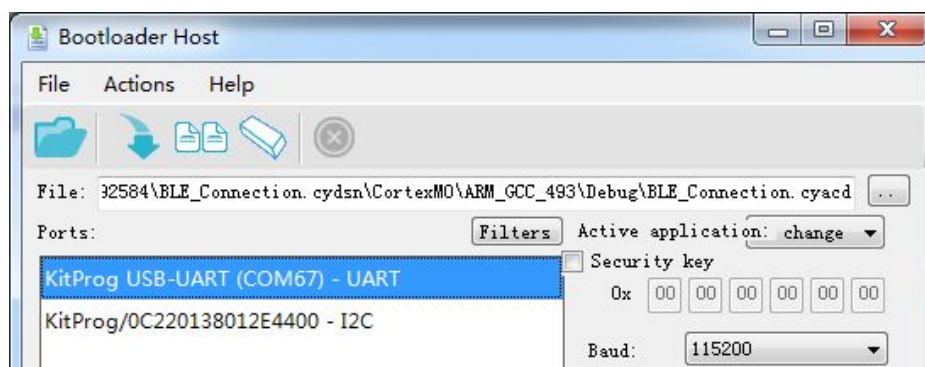
AT+SHOW

The specific UUID in APP for reading Service is: 0000180F-0000-1000-8000-00805F9B34FB, user can read by standard BLE protocol

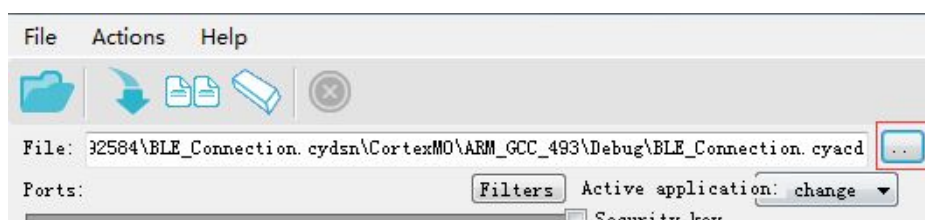
2.9. Firmware Update

USR-BLE100 support UART update for user to keep the stability in real time. Here we provide the upgrade tool.

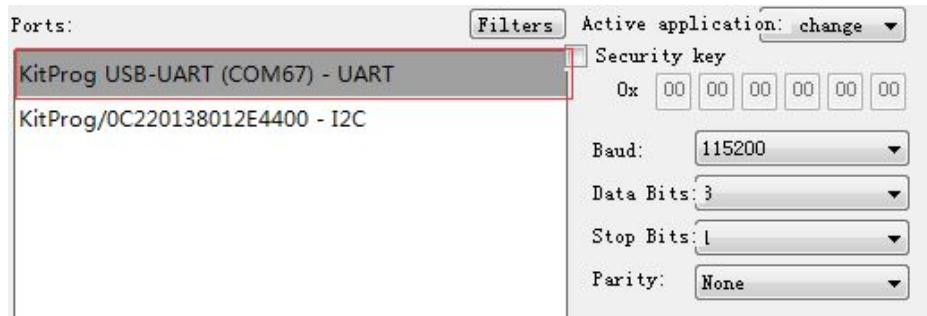
1) Open our tool Bootloader Hoost



2) Click the red square part in below image to load firmware, the extension of the firmware is cyacd, shown as below:

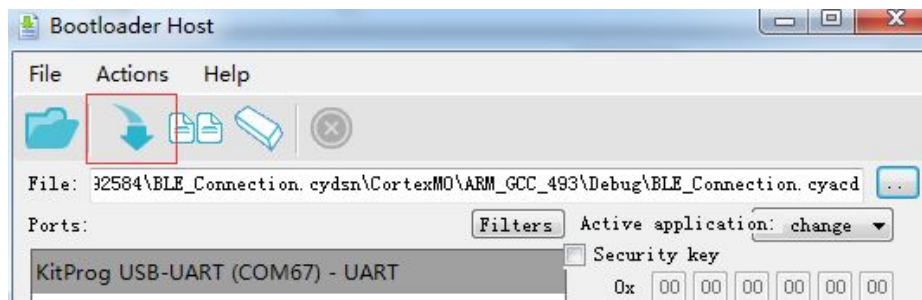


3) Find the COM number which needs to connect to the Bluetooth module:



4) In this example, the COM number is 67, click it, we can see the UART parameters in the right. Here can choose baud rate as 115200.

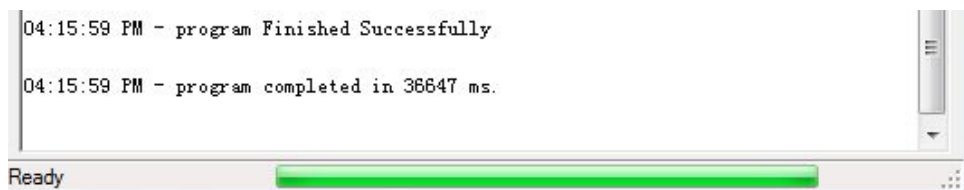
5) Ready, power on again or press the reset button, click the red square button in below image within 1s:



6) See the progress bar, if it moving, the module is loading smoothly. If did not move or show errors, pls try step 5 again. Make sure to press the button within 1s.



7) When the progress bar finish and the window shows Ready, the module upgrading succeed. Now you can restart the module and it will work with the new firmware.



2.10. Indicator Status

There is an indicator on USR-BLE100 module, it displays differently in different module status. The specific situation as shown below.

Module Status	Indicator Status
Master Searching	Flash 3 times per second
Slave Broadcasting	Flash 1 time per second
Connection Establishment	ON
Low Power Mode	OFF
Data Transmission	Flash

3. Setup Method

3.1. AT Instruction

3.1.1. Serial AT Instruction

USR-BLE100 has varied work mode, it will enter the specified mode when start-up, user can switch pattern into command line (AT instruction) mode through serial commands.

There are two steps that switch other mode into instruction mode:

- 1) Input “+++” in serial port, then module will reply a confirmation code “a” when received “+++”;
- 2) Input “a” in serial port after received “a”, module will reply “+OK” to ensure, then pattern will enter instruction mode;

User also can input “+++a” and wait module reply “+OK” to enter instruction mode. User can send specific instruction in AT format after enter command mode.

3.1.2. Transparent Transmission Instruction

USR-BLE100 also supports transparent transmission, user can use APP or the module connected with it to set parameters in this mode. Users just need to input the 6 digits password of the module to be set and instruction in transparent transmission. For example, if user want to query version number, you should input “000000, AT+CTVER enter”. Like this, user can set and query parameter without entering serial instruction mode in data transmission mode.

3.1.3. AT Instructions

Instruction	Description
General Command	
NAME	Query/Set Module Name
MODE	Query/Set Pattern Mode
MAC	Query Module's MAC Address
CIVER	Query Software Version Number
TPL	Query/Set Module Transmission Power
SHOW	Show NAME, Pattern, MAC, Connection, Power and so on.
PASS	Query/Set Mode Password
PASSEN	Query/Set Paired Validation
UART	Query/Set Mode Serial Parameter
UARTTM	Query/Set Serial Packet time
AUTOSLEEP	Query/Set Sleep Automatically
DEEPSLEEP	Enter Ultra-low Pattern
HIBERNATE	Enter Sleep Pattern
HELLO	Query/Set Welcome Words
ENTM	Pull Out Instruction Pattern
RELOAD	Resume To Default Setting
Z	Restart Module
Link Command	
LINK	Query Mode Connection status
SCAN	Query Slaves Surrounded
CONN	Quick Connect By Query Reference Number
CONNADD	Query/Set MAC Address Default Connected
DISCONN	Disconnect Current Connection
ADP	Query/Set If Broadcast In This Pattern
ADPTM	Query/Set Broadcasting Speed
Characteristic Function	
BATEN	Use/Forbidden Power testing
MAXPUT	Use/Forbidden maximum output
TRENC	Use/Forbidden PCI Encrypted Transmission
IBEACON	Query/Set iBeacon Function

3.1.3.1. AT+NAME

Function: Query/Set the module name

Format:

Query the current parameters

AT+NAME?
{CR}

{CR}{LF}+NAME:name{CR}{LF}{CR}{LF}OK{CR}{LF}

Set:

AT+NAME=name{CR}{CR}{LF}OK{CR}{LF}

Parameter:

name: module name

3.1.3.2. AT+MODE

Function: Query/Set module work mode

Format:

Query current parameter value:

AT+ MODE?{CR}
{CR}{LF}+ MODE:mode{CR}{LF}{CR}{LF}OK{CR}{LF}

Set:

AT+MODE=mode{CR}{CR}{LF}OK{CR}{LF}

Parameter:

mode module work mode:

M: Master mode

S: Slave mode

B: Broadcast mode

3.1.3.3. AT+MAC

Function: Query/Set the MAC address of module

Format:

Query current parameter:

AT+MAC?{CR}
{CR}{LF}+MAC:mac{CR}{LF}{CR}{LF}OK{CR}{LF}

Parameters:

mac: the MAC address of module, example: FFFFFFFF

3.1.3.4. AT+CIVER

Function: query the version number

Format:

Query current parameter:

AT+CIVER?{CR}
{CR}{LF}+VER:V1.0.0{CR}{LF}{CR}{LF}OK{CR}{LF}

3.1.3.5. AT+TPL

Function: Query/Set module transmit power

Format:

Query current parameter

```
AT+TPL?{CR}
{CR}{LF}+TPL:size{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+TPL=size{CR}{CR}{LF}OK{CR}{LF}
```

Parameters:

size: module transmit power

1: -18dbm

2: -12dbm

3: -6dbm

4: -3dbm

5: -2dbm

6: -1dbm

7: 0dbm

8: +3dbm

3.1.3.6. AT+SHOW

Function: show the information of NAME, Mode, MAC, Link, etc.

Format:

Query current parameter

```
AT+SHOW{CR} or AT+SHOW?{CR}
```

```
{CR}{LF}+SHOW:{CR}{LF}NAME :name{CR}{LF}MODE :mode{CR}{LF}MAC:mac{CR}{LF}PASS :pa
ss{CR}{LF}ADP :size{CR}{LF}ADPTM:time{CR}{LF}LINK :status{CR}{LF}Battery:
num{CR}{CR}{LF}OK{CR}{LF}
```

Parameters:

name: module name

mac: module MAC address

pass: password

size: transmit power

time: speed of broadcast

FAST: fast broadcast

SLOW: slow broadcast

status: link status

ON: connect succeed

OFF: connect failed

num: battery capacity

3.1.3.7. AT+PASS

Function: Set/Query module password

Format:

Query current parameter value

```
AT+PASS?{CR}
```

```
{CR}{LF}+PASS:password{CR}{LF}{CR}{LF}OK{CR}{LF}
```


Set:

```
AT+PASS=password{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

password: 6 bit password of module

3.1.3.8. AT+PASSEN

Function: Query/Set whether enable connection authentication. If enable, when master connect to the slave which also enables the connection authentication, it needs to send the 6 bit password within 10s. If time out or password incorrect, the link will be disconnect.

Format:

Query current parameter value

```
AT+PASSEN?{CR}
{CR}{LF}+PASSEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+PASSEN =status{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

Status: link authentication enable status, including:

“on”: enable

“off” disable

3.1.3.9. AT+UART

Function: Query/Set UART parameters

Format:

Query current parameters

```
AT+UART?{CR}
{CR}{LF}+UART:baudrate ,databit,pari,stop{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+UART=baudrate ,databit,pari,stop{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

baudrate: UART baud rate, range from 2400~115200

databit: data bit 5,6,7,8,9

pari: parity bit

0: NONE

1: ODD

2: EVEN

Stop: stop bit

0:1 bit stop

1:1.5 bit stop

2:2 bit stop

3.1.3.10. AT+UARTTM

Function: Set/Query UART packet time

Format:

Query current parameter

```
AT+UARTTM?{CR}
{CR}{LF}+UARTTM:time{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+UARTTM=time{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

time: UART packet time, range from 0~300

3.1.3.11. AT+AUTOSLEEP

Function: Set/Query auto sleep

Format:

Query current parameter

```
AT+AUTOSLEEP?{CR}
{CR}{LF}+AUTOSLEEP:status ,time{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+AUTOSLEEP=status,time{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

status: whether enable auto sleep function

ON: open

OFF: close

time: auto sleep waiting time, waiting time is time*5s

3.1.3.12. AT+DEEPSLEEP

Function: Set deep sleep mode

Format:

Set:

```
AT+DEEPSLEEP{CR}
{CR}{LF}+OK{CR}{LF}
```

3.1.3.13. AT+HIBERNATE

Function: Set hibernate mode

Format:

Query current parameter:

```
AT+HIBERNATE {CR}
```

{CR}{LF}+OK{CR}{LF}

3.1.3.14. AT+HELLO

Function: Set/Query hello words

Format:

Query current parameter

AT+HELLO?{CR}

{CR}{LF}+HELLO:string{CR}{LF}{CR}{LF}OK{CR}{LF}

Set:

AT+HELLO=string{CR}

{CR}{LF}OK{CR}{LF}

Parameters:

string: hello words when start, length less than 20 bytes

3.1.3.15. AT+ENTM

Function: quit command mode

Format:

Set current parameter

AT+ENTM{CR}

{CR}{LF}+ENTM:OK{CR}{LF}

3.1.3.16. AT+RELOAD

Function: restore factory defaults

Format:

Set current parameter

AT+RELOAD

{CR}{LF}+RELOAD:OK{CR}{LF}{CR}{LF}OK{CR}{LF}

3.1.3.17. AT+Z

Function: reset

Format:

Set current parameter

AT+Z{CR}

{CR}{LF}+RST:OK{CR}{LF}{CR}{LF}OK{CR}{LF}

3.1.3.18. AT+LINK

Function: query module link status

Format:

Query current parameter

```
AT+LINK?{CR}
{CR}{LF}+LINK:status {CR}{LF}{CR}{LF}OK{CR}{LF}
```

Parameters:

status: module link status
ON: link succeed
OFF: no link

3.1.3.19. AT+SCAN

Function: scan the slaves around

Format:

Query current parameter

```
AT+SCAN{CR}
{CR}{LF}No:numAddr:macRssi:size{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Parameters:

num: scanned the index number of slave
mac: scanned the MAC address of slave
size: scanned the signal strength of slave

3.1.3.20. AT+CONN

Function: quick connect by scanned index number

Format:

Set current parameter

```
AT+CONN=num{CR}
{CR}{LF}+CONN:num{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Parameters:

num: the index number scanned

3.1.3.21. AT+CONNADD

Function: Set/Query the MAC address of module which it will connect to when power on

Format:

Query current parameter

```
AT+CONNADD?{CR}
{CR}{LF}+CONNADD:mac{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+CONNADD=mac{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

mac: MAC address of module which it will connect to when power on

3.1.3.22. AT+DISCONN

Function: set to disconnect current link

Format:

Set:

```
AT+DISCONN{CR}
{CR}{LF}OK{CR}{LF}
```

3.1.3.23. AT+ADP

Function: Set/Query whether module open broadcast

Format:

Query current parameter

```
AT+ADP?{CR}
{CR}{LF}+ADP:status {CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+ADP=status {CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

status: set whether module open broadcast

ON: open

OFF: close

3.1.3.24. AT+ADPTM

Function: Set/Query module broadcast speed

Format:

Query current parameter

```
AT+ADPTM?{CR}
{CR}{LF}+ADPTM:time{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+ADPTM=time{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

time: module broadcast speed

FAST: fast broadcast

SLOW: slow broadcast

3.1.3.25. AT+BATEN

Function: Enable/Disable electric detection

Format:

Query current parameter

```
AT+BATEN?{CR}
{CR}{LF}+BATEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+BATEN =status{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

status: battery electric detection status, including:
ON: enable
OFF: disable

3.1.3.26. AT+MAXPUT

Function: enable/disable max output

Format:

Query current parameter:

```
AT+MAXPUT?{CR}
{CR}{LF}+MAXPUT:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+MAXPUT =status{CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

Status: max output status, including:
ON: enable
OFF: disable

3.1.3.27. AT+TRENC

Function: Query/Set data encryption transmission

Format:

Query current parameter

```
AT+TRENC?{CR}
{CR}{LF}+TRENC:status {CR}{LF}{CR}{LF}OK{CR}{LF}
```

Set:

```
AT+TRENC=status {CR}
{CR}{LF}OK{CR}{LF}
```

Parameters:

status: module encryption transmission, including:
ON: enable
OFF: disable

3.1.3.28. AT+IBEACON

Function: Set/Query iBeacon function

Format:

Query current parameter

AT+IBEACON?{CR}

{CR}{LF}+IBEACON:uuid, major, minor, rssi{CR}{LF}{CR}{LF}OK{CR}{LF}

Set:

AT+IBEACON=uuid, major, minor, rssi{CR}

{CR}{LF}OK{CR}{LF}

Parameters:

uuid: iBeacon device identification code

major: major code

minor: minor code

rssi: the signal for 1M distance (no minus in setting)

3.1.3.29. AT+ADVDATA

Function: Set/Query broadcast data

Format:

Query current parameter:

AT+ ADVDATA?{CR}

{CR}{LF}+ ADVDATA:data{CR}{LF}{CR}{LF}OK{CR}{LF}

Set:

AT+ ADVDATA= data{CR}

{CR}{LF}OK{CR}{LF}

Parameters:

data: defined broadcast data packet

4. Contact US

Company: Jinan USR IOT Technology Limited

Address: Floor 11, Building1, No.1166 Xinluo Street, Gaoxin Distric, Jinan, Shandong, 250101, China

Tel: 86-531-55507297, 86-531-88826739

Web: <http://www.usriot.com>

Support : <http://h.usriot.com>

Email: sales@usr.cn, tec@usr.cn

5. Disclaimer

This document provide the information of USR-BLE100 products, hasn't been granted any theintellectual

property license by forbid speak or other ways either explicitly or implicitly. Except for the duty of sales conditions and conditional declarations, we don't take any other responsibilities. We don't warrant the products sales and use explicitly or implicitly, including particular purpose merchantability and marketability, the tort liability of any other patent right, copyright, intellectual property right. We may modify specification and description at any time without prior notice.

6. Update History

2015-11-30 V1.0 Establish

2016-01-04 V1.1 Modify MAC and white list instruction