Bluetooth (BLE 4.2) Module and Standard Transparent

RSBRS02AA

Transmission Protocol

Protocol Version: V1.0

RSBRS02AA

Shenzhen RF-Star Technology Co., Ltd.

Last Updated: 07/12/2017

Preface

How to develop new peripherals for smart phones fast and cost-effectively?

-Application of Bluetooth Low Energy Technology in Smart Mobile Devices-

The emergence of the USB protocol brought about massive surge of PC peripherals. Likewise, the newly launched BLE wireless application technology on smart phones has the same results. BLE technology provided the possibility of bridging electronic products and smart phones. Compared with Wi-Fi and Bluetooth 2.0 techniques, it has the advantages of low power consumption, fast connection, longer range of communication, and broadens the prospect of developing peripheral electronic devices for smart phones.

The RSBRS02AA Bluetooth low energy module comes with a RF-star RS02A1 BLE SoC which runs over 2.4GHz ISM band. With GFSK (Gauss Frequency Shift Keying) modulation, it provides 40 2MHz channels (3 dedicated broadcasting channels and 37 adaptive frequency hopping data channels). At the physical layer, it can combine with the RF end of the Bluetooth Classic to form a dual mode device. The 2MHz gap facilitate better avoiding interference of neighboring channels. The module also boasted of broad output power range (-23dBm~0dBm), and high gain sensitivity of -93 dBm.

Targeting to fast bridge the electronic products and smart mobile devices, the module was designed to be applied by electronic devices in the fields of instruments and meters, logistics tracking, health and medicare, smart home, metrology, automotive electronics, games and toys etc. As Android 4.3 has integrated the technology, the BLE will definitely become a standard configuration of smart phones. Hence, its demand in phone peripheral markets will exponentially increase. With this module, customers can integrate existing solutions or products extremely fast and quickly occupy the market, injecting new technical strength for development of their businesses.

● Version Update Record

Version No.	Date	Updates
V1.0	2017/12/07	✓ First release

Content

● Overview

The module can work in both bridging mode (transparent transmission mode) and direct-drive mode.

After started, the module can broadcast automatically, which enables the smart phone with specific application running to scan and pair with it. When successfully connected, the smart phone can monitor and control the module through Bluetooth protocol.

In bridging mode, user MCU can communicate with the mobile device bi-directionally through module's GPIO. Users can also manage and control certain communication parameters with specific serial port AT commands. The specific meaning of the user data is defined by the upper-layer application. Mobile devices can write the module through the APP, and the recorded data will be sent to the user MCU through serial ports. Then the module will transmit the data packets received from user CPU to the mobile devices automatically. To accomplish the development, the user must finish the code design for master CPU, and the code design of APP for smart mobile terminals.

In direct-drive mode, users can expand the module's simple peripherals. App can drive the module directly through BLE protocol, accomplishing the smart phone devices' supervision and control over the module. For software development in this mode, users only have to do code design for the App in the smart mobile device.

Characteristics:

1. Easy to use, no experience of Bluetooth protocol stack application is needed;
2. UART design for user interface, full-duplex bi-directional communication, minimum Baud rate 4800 bps;
3. Default connection interval of 30ms, which makes quick connection;
4. Provide AT commands to software reset the module, and to acquire to the MAC address;
5. Support adjustment of Bluetooth connection interval by AT command, and the control of different forwarding rates (dynamic power adjustment);

6, Provide AT commands to adjust the transmission power, to change broadcasting interval, to customize broadcasting data, to customize device ID code, to set the data time (preparation time of user MCU serial port before receiving), to change the serial port Baud rate or he module names. For details, please refer to AT command list;

7. The length of the UART data package can be below or equal to 200 byte (large packet automatic distribution);

8. High-speed transparent transmission forwarding rate can be up to 4 K/s and can be stabilized at 2.5 K/s to 2.8 K/s (IO5, IO6);

9. Support changing module name by APP, changing UART baud rate and product UDID, customizing broadcast content and cycles (all settings are preserved after power-off);

10. Support remote resetting the module by APP, and setting transmission power;

11. Support adjusting Bluetooth connection interval by APP, but the setting cannot be saved after power-off (dynamic power adjustment);

12. Extremely low power standby mode (sleep mode current 0.3 μA for CC2541 SoC, according to TI data), and the measured power consumption data is as follows：

Event	Average current (integral computed)	Average current (ammeter measured)	Duration	Testing Conditions / Remarks
Sleeping	4.90uA	4.98uA	—	EN Pulled up
Broadcasting	89.8uA	0.09mA	3.85 ms	broadcasting cycle is 350 ms
Connection	335uA	0.35 mA	2.25 ms	Connection cycle is 100 ms

Notes:

* The official test method: Connect in series a 10Ω resistor in the circuit with power supply, and intercept voltage waveform with oscilloscope and perform integral operation.

** Multimeter test method: Connect the multimeter (set at μA or mA level) in series between the battery and the module to check the value displayed, with the test voltage of 3.07 V.

Above is the measured sampling data of module RF-CC2540A1 and for reference only. The data in the brackets is from module RF-CC2541A1.

If lower power consumption is expected, connection interval or broadcast cycle can be appropriately increased, as shown in relevant chapters such as *module parameter settings* and *serial port AT commands*.

● Illustration of Working Mode

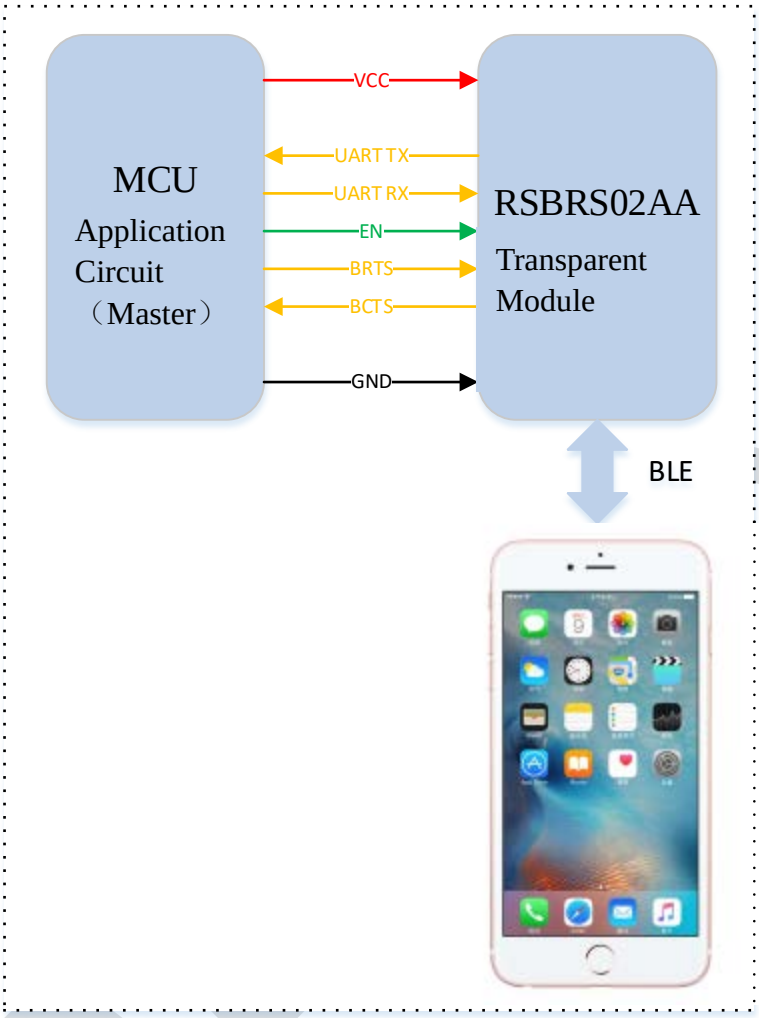
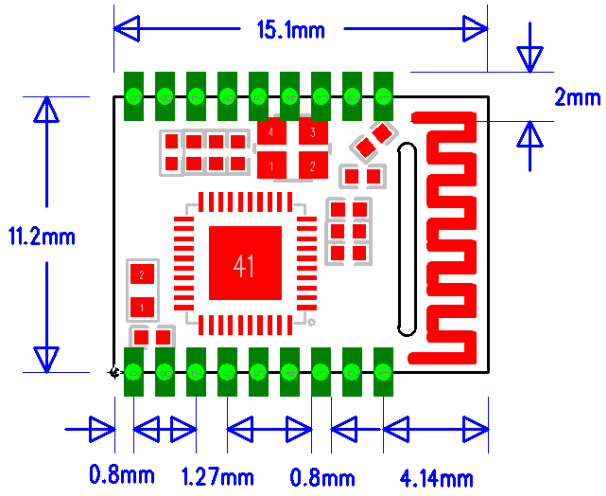


Illustration of Bridge Working Mode

Note:

- In order to avoid the output level difference between user MCU's IO and module IO, which will result to high current, a small isolation resistor is suggested to be connected in series in the output signal line TX and BCTS.

● Package Size and Pin Assignments



Module PIN RSBRS02AA

RSBRS02AA Pin Definition

Pin No.	Module Pin Name	Chip Pin Name	I/O	Specification
Pin1	GND	GND	—	Grounding GND
Pin2	VCC	VCC	—	Power Supply: 1.6~3.6V
Pin3	IO7	P15	O	Sleep Mode indicator
Pin4	IO6	P24	O	Connection Status indicator
Pin5	RES	RES	I	- Reset Input - Active when low level, with internal pull-up.
Pin6	EN	P06	I/O	Module-enabled control line *(level trigger mode as default)* - Level trigger mode - Active when low level, with internal pull-up. 0: Module starting to broadcast, until connected to the mobile device 1: Entering sleep mode immediately, regardless of the current status
Pin7	SWC	SWC	—	Clock Pin when download firmware
Pin8	SWD	SWD	—	Data Pin when download firmware
Pin9	I2C-	P21	I/O	

Pin10	IO3	P10	I/O	
Pin11	IO2	P11	I/O	
Pin12	IO1	P12	I/O	
Pin13	IO0	P14	I/O	
Pin14	BRTS	P16	I	As the data sending requests (for module wake-up) - 0: Host has data to send, and module will wait for data transmission from the host so will not sleep 1. Host has no data to send, or data has been sent. So the value of the signal should be set at "1".
Pin15	BCTS	P17	O	Data input signal (for host wake-up, optional) - 0: Module has data to send, and the host will receive the data. 1: Module has no data to send, or data has been sent, and the value of the signal will be set at "1".
Pin16	TX	P27	O	Serial port TX
Pin17	RX	P28	I	Serial port RX
Pin18	ADC	P03	I	Analog acquisition

- RSBRS02AA **is a compact edition, so parts of the IO are not pinned, the corresponding function cannot be used.**

● Protocol Description – Serial Port Transparent Transmission (Bridge Mode)

The bridge mode means to set up two-way communication between user CPU and the mobile device by connecting the module with user CPU through serial ports. Users can reset the serial port baud rate and BLE connection interval, using the specified AT commands (see behind the section "Serial Port AT Command"). The module will have different data TX & RX capability, as per different serial port baud rates,BLE connection intervals and packet intervals. Default Baud Rate is 115200bps.

When the BLE connection interval is 30 ms and the serial port baud rate is at 115200 bps, the module has the highest transmit ability in theory (8.4k/s). Given the configuration in the level-enabled mode as an example, UART transparent transmission protocol will be detailed introduced as below.

The module can transmit through serial port maximum 500-byte data packet at one time. According to the packet size, the packet will be sub-packed automatically and sent, with a maximum load of 255 bytes for each wireless sub-packet. Data packets from mobile devices to the module must be sub-packed by their own (into 1 – MTU bytes/packet) before sending. The module will forward them to the host serial RXD end in turns, when receiving the packets.

1. The serial port hardware protocol: 115200 bps, 8, no parity, 1 stop bit.
2. When EN is set at high level, the Bluetooth module is in full sleep mode. When EN is set low, the module will start broadcasting at the internal of 50 ms, until it pairs with mobile devices. When EN jumps from low to high, the module will enter into sleep mode immediately, regardless of current status.
3. After the module is connected, BRTS needs to be pulled low if the host (MCU) has data to send to the BLE module, and the data transmission can be started around 100 μs afterwards. BRTS should be raised high by the host after transmission finishes and make the module exit the serial RX mode. Pay attention to confirm that serial port data transmission has been completely finished before raising BRTS. Otherwise there will be data truncation.
4. When there is data upload request, the module will set BCTS low, until data transmission finishes. The transmission can start at least 500 μ s afterwards. And this delay can be configured with the AT command (see in section "serial AT command"). BCTS will be set high by the module when data transmission is finished.
5. If the host BRTS is being kept at low level, the Bluetooth module will always be in serial port RX mode and the power consumption will be high.
6. After the module is connected, a string of "TTM:CONNECTED\r\n\0" will be prompted from TX. The string could be used to determine if the normal transmission can be done. Of course the connection status prompt pin can be used instead. Also the connection can be confirmed by sending a specific confirmation string to the module from mobile devices. When APP actively disconnect the module, there will be a prompt of string "TTM: DISCONNET \ r \ n \ 0" from TX.
7. The default Bluetooth connection interval is 30 ms. If low-speed TX mode is needed for saving power, connection interval must be adjusted by AT command (longest connection interval to be

2000 ms). In each interval maximum transmission is 255 bytes. Set the connection interval as T(unit: ms), and highest transmit rate per second as V (byte/s), then their relation is as follows:

V = 255*1000/T (V is only relevant with T)

If the Bluetooth connection interval of the module is **30 ms,** and in each interval it can transmit maximum **255 byte**, the theoretical maximum transmission capacity (transmit rate) will be **255 * 33 = 8.4 k byte/s**. Tests have shown that the packet loss is very little when transmission rate is under 4 K/s. For safety, it is suggested to do vertification and re-transmission processing in the upper layer, no matter for high or low speed transmission applications.

8. Below is an example of the communication with 30 ms connection interval. Configuration can be set by yourself. But the lower the transmission rate V0 is， the less packet leakage will incur.

Communication Ref. Mode	BLE Connection Interval	Highest theoretical transmission capacity V (byte/s) V = 80*1000/T	Serial Data Packet Length	Serial Port Transmission Interval TS(ms) When L<80，TS >= T When 80<L<160，TS >= T*2 When 160<L<200，TS >= T*3	Actual transmission rate V0 (byte/s) V0 = L*1000/TS	Remarks
1	20	4K	80	TS >= T, if TS=20ms	80*1000/20 = 4K	TS small, not recommended
2	20	4K	200	TS >= T*3, if TS=70ms	200*1000/70 = 2.8K	
3	20	4K	200	TS >= T*3, if TS=80ms	200*1000/80 = 2.5K	
4	20	4K	80	TS >= T, if TS=35ms	80*1000/30 = 2.6K	
5	20	4K	70	TS >= T, if TS=30ms	70*1000/30 =2.3K	
6	20	4K	60	TS >= T, if TS=30ms	60*1000/30 = 2K	
7	20	4K	40	TS >= T，if TS=30ms	40*1000/30 = 1.3K	
8	20	4K	20	TS >= T，if TS=30ms	20*1000/30 = 666byte	

Table 5. Example of Communication at the Interval of 20ms

Note 1: MTU varies in mobiles, data above shows the example of standard BLE 4.0 transmission rate. Mobiles that support BLE 4.2 or MTU>20 byte will have the transmission rate higher than above.

Note 2:

- **Specific communication mode can be designed according to the practical application. Serial-packet length can be designed to be between 80 and 200 byte (large packet transmission) . As per the BLE protocol there is the following relations:**

-

When L<80，TS >= T；

When 80<L<160，TS >= T*2；

When 160<L<200，TS >= T*3；

- Transmission modes that comply with above-said conditions are generally safe in operation. However among them, when TS = T, T * TS = 2 or TS = T * 3, it is workable but not recommended, as the packet loss is relatively high and verification and re-transmission mechanism must be added. In other words, when a serial data packet is as big as 80 byte < L < 200 byte, serial data can be sent to the module for one time, but certain time needs to be reserved for data transmission. Otherwise there will be a rear-end data collision. For example, when the connection interval T = 20 ms, 3 = T * TS must be bigger than 60 ms, if the serial packet length L = 200. So setting TS = 70 ms is a logical choice.

✧ The size of the serial data packets can be various and the length can be any value less than 500 bytes, as long as the above-said conditions are met. But in order to utilize the communications payload in highest efficiency, while to avoid communication running in full capacity, it is recommended to use serial data packets of 20,40, or 60 bytes in length, and interval between packets above 20 ms.

Note:

- **Test show that in iOS, calling the writing function to Characteristic with the parameter CBCharacteristicWriteWithResponse (writing mode with response) will reduce partially the transmission efficiency, but the correctness of a single packet will be ensured. While with the parameter CBCharacteristicWriteWithoutResponse (writing mode without response), the transmission efficiency will be increased, but the correctness of data packet needs to be verified by APP in the upper layer.**

● Serial Port AT Commands

Strings starting with "TTM" will be regarded as AT commands to be parsed and executed. and will return exactly the same from the serial port. Afterwards the execution result will be output (ie. TTM: OK \ r \ n \ "0" or "TTM: ERP \ r \ n \ 0", etc.) Serial data packets which do not start with "TTM"will be regarded as transparent transmission data.

➢ Connection Interval Setting

Input the following string to the serial port RX to set the BLE connection interval:

"TTM:CIT-Xms"

Where X = "20", "**30**", 50", "100", "200", "300", "400", "500", "1000", "1500", or "2000" (ms). After the command is executed, the following confirmations will be got from serial port TX:

"TTM: OK \ r \ n \ 0" (means the change is successful and the new connection interval is applied);

The success of connection interval setting depends on the constraints of connection intervals of mobile devices. The maximum connection intervals also vary in different version of iOS. Tests with (iOS 8) show the fastest is 20ms and the slowest is 2s. On the other hand, due to the BLE protocol internal mechanism, execution efficiency of this command will be different with different connection intervals. In iOS8, it takes maximally around 100 s, changing from the current connection interval of 2000ms (max. 2000ms) to other connection intervals. While the execution will be fast when executing this AT command in other high-frequency connection intervals.

Note:

- The connection interval setting cannot be saved after power-off. And command of change is only effective when the connection is successful.

➢ Module Rename

Input the following string to the serial RX to rename the module (length of name should not exceed 16 bytes, in the form of ASCII code).

" TTM:REN-" + Name

Also confirmation of "TTM: OK \ r \ n \ 0" will be received from TX. And if the command format is incorrect, the string as follows will be returned:

"TTM:ERP\r\n\0"

Test shows that the device name can be changed immediately in iOS6 and above versions but not in iOS5. The name can be saved after power-off.

➢ Baud Rate Setting

Input the following string to the serial port RX , setting the baud rate:

"TTM:BPS-X"

And X="4800"，"9600"，"19200"，"38400"，"57600"，"**115200**"，"256000"(These data show in the form of ASCII code), for example"TTM:BPS-115200" means the baud rate is 11520bps. After the command finishes

Confirmation string of "TTM:BPS SET AFTER 2S..." will be received from TX. If the value set is not

in the options, or the command format is incorrect, the string as follows will be returned:

"TTM:ERP\r\n\0"

➢ Acquiring Physical Address MAC

Input the following string to a serial port RX:

"TTM:MAC-?"

And the following string will be received from TX:

" TTM:MAC-xxxxxxxxxxxx\r\n\0"

"xxxxxxxxxxxx" after "-" is the Bluetooth address in 6 bytes.

➢ Module Reset

Input the following string to serial port RX will force the module to be soft-reset once:

"TTM:RST-SYSTEMRESET"

➢ Broadcast Cycle Setting

Input the following string to serial port RX, to set the broadcast cycle of the module, T = X ms

"TTM:ADP-(X)"

Where X = "20", "**50**", "100", "200", "500", "1000", "1500" or "2000", "2500", "3000", "4000", "5000"(data above shows in the form of ASCII code). For example, "TTM:ADP-(20)" means the set broadcast cycle is 20ms. Confirmation string of "TTM: OK \ r \ n \ 0" will be received from TX. If the command format is incorrect, the following string will be returned:

"TTM: ERP\r\n\0"

Broadcast cycle setting can be saved after power-off. After the module is rebooted, the module will broadcast as per the new broadcast cycle.

➢ Additional Customized Contents of Broadcast

Input the following string to the serial port RX to customize broadcast contents

"TTM:ADD-"+ Data

Where "Data"is the additional data ready to be broadcasted (0 < Length < = 16 bytes). The confirmation string of "TTM: OK \ r \ n \ 0" will be received from TX. If the command format is incorrect, the following string will be returned :

"TTM:ERP\r\n\0"

It takes immediate effect when command is executed. Certain customized content can be broadcast in this way. And the data can be saved after power-off. If all the 16 bit data is set as 0, customized broadcast data will not be used. Instead, the default broadcast content is applied.

➢ Defining Product Identification Code

Input the following string to the serial port RX to define product identification code:

"TTM:PID-"+ Data

where "Data" is for a two-byte product identification code (ranging from 0x0000 range to 0xFFFF and length =2). The confirmation string of "TTM: OK \ r \ n \ 0" will be received from TX. If the command

format incorrect, the following string will be returned:

"TTM:ERP\r\n\0"

This identification code can be saved after power-off. It will show in the broadcasting, and can be used to filter devices or to determine if it is a specific product.

➢ Transmission Power Setting

Input the following string to the serial port RX to set the corresponding transmission power (in dBm).

"TTM:TPL-(X)"

Where X = "0", "-2", "-5", "-10", "-12", or "- 15". (in the format of ACSII code). The confirmation string of "TTM: OK \ r \ n \ 0" will be received from TX. And the module will immediately communicate with the new transmission power. If the command format is incorrect, the following string will be returned:

"TTM:ERP\r\n\0"

Note: The hardware of Chip RS02A1 can support the transmission power(-20dBm ~ +7dBm), This transparent transmission version rang from15dBm to 0dBm.

➢ Data Delay Setting

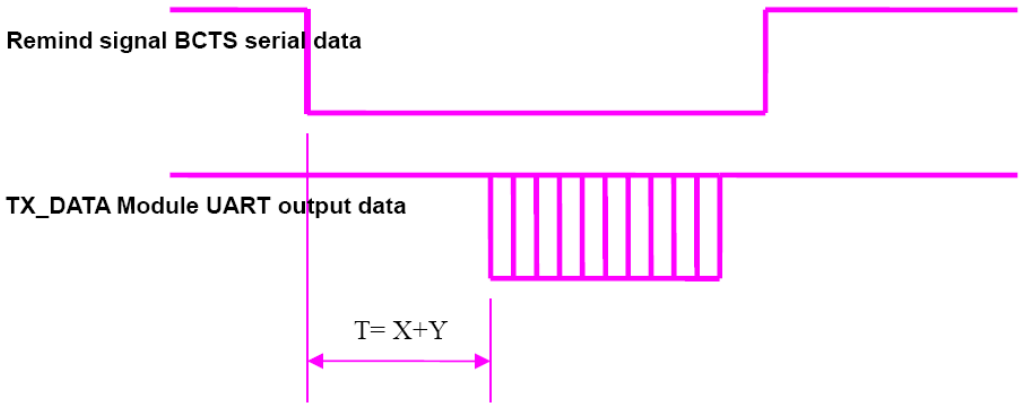
Input the following string to the serial port RX to set the delay between when BCTS outputs low level and when serial port TX outputs data (in ms)

"TTM:CDL-Xms"

Where X = "0", "2", "5", "10", "15", "20", or "25". The confirmation string of "TTM: OK \ r \ n \ 0" will be received from TX. If the command format is incorrect, the following string will be returned:

"TTM:ERP\r\n\0"

To make sure the user CPU have enough time to be waken up from sleep mode and ready to receive data, the module provides this delay (X) setting. The BRTS will be set low before the data are to be sent through the module's serial port. While the delay from when BRTS is set low level till when the module TX outputs data will be set by this parameter. The actual delay (T) will be T = (X + Y) ms, if minimum delay is not less than X, while 500 μ s < Y < 1 ms. This setting can be saved after power-off.



Scheme of Serial Port Output Data Delay Setting

◆ The AT command list is as follows:

AT Command Format	Saved After Power-off	Parameter and Description	Possible Response	Remarks
"TTM:CIT-Xms" (Valid only when connection is successful	no	X="20","**30**","50","100","200","300","400","500","1000","1500","2000": Set the BLE connection interval (ms)	"TTM:TIMEOUT\r\n\0" "TTM:OK\r\n\0" "TTM:ERP\r\n\0"	TIMEOUT setting Setting successful Parameter error
"TTM:REN-"+Name	yes	Name: New module name, with length not exceeding 15 bytes	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	Setting successful Parameter error
"TTM:BPS-X"	yes	X="4800", "**9600**", "19200", "38400", "57600","**115200**", "256000": Set the baud rate	"TTM:BPS SET AFTER 2S ...\r\n\0" "TTM:ERP\r\n\0"	Setting successful and new baud rate will be applied in two seconds Parameter error
"TTM:MAC-?"	-	Acquire MAC address	"TTM:MAC-xxxxxxxxxxxx" xxxxxxxxxxxx for module MAC address	Return with MAC address
"TTM:RST-SYSTEMRESET"	-	Reset the module	NO	Reset the module
"TTM:ADP-(X)"	yes	X="20","**50**","100","200","500","1000","1500","2000","2500","3000","4000","5000"setting the corresponding broadcast cycle, T = X ms	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	Setting up the broadcast cycle.(ie.If the parameter set to "5", the cycle will be 500 ms) Parameter error
"TTM:ADD-" + Data	yes	“Data” for customized broadcast data, and length L < = 16	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	Set the customized broadcast contents Parameter error
"TTM:PID-"	yes	Data for customized	"TTM:OK\r\n\0"	Set the

+ Data		product identification code, and length L = 2, and default value is "**RS**"	"TTM:ERP\r\n\0"	customized product identification code Parameter error
"TTM:TPL-(X)"	no	X = "**0**", "-2", "-5", "-10", "-12", or "- 15": Set up the transmit power (in dBm)	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	Transmission power setting Parameter error
"TTM:CDL-Xms"	yes	X = "0", "2", "5", "**10**", "15", "20", "25": Set the delay from when BCTS output is set low till when serial port outputs data (in ms)	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	If minimum delay not less than X, the actual delay will be X + Y ms (500 us < Y < 1 ms). Parameter error

*** Note: The hardware of Chip RS02A1 can support the transmission power(-20dBm ~ +7dBm), This transparent transmission version rang from15dBm to 0dBm.**

● BLE Protocol Description (APP Interface)

➢ Bluetooth data channel 【service UUID: 0xFFE5 】

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFE9 (handle:0x0013)	Write	20	Nill	Written data will output from serial port TX

Directions: Bluetooth input data will be transmitted to serial port TX output. APP write to this channel through BLE API, and the data will be output from serial port TX. For operation details, please see the section "Protocol Description – Serial Port Transparent Transmission (Bridge Mode)".

➢ Serial data channel【service UUID:0xFFE0】

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFE4 (handle: 0x000E)	Notify	20	Nill	Notification will be generated from the input data of serial RX input and sent to smart devices

Directions: Serial input data will be transmitted to BLE output. If notify EN switch of FFE1 is opened, a notification event will be generated in this channel when the host CPU transmit legal data to module RX through serial port. App can directly process and use it in the callback function. For operation details, please see the section "Protocol Description – Serial Port Transparent Transmission (Bridge Mode)".

➢ Module parameter settings【Service UUID：0xFF90】

Characteristic UUID	Operation	Can Be Saved	Bytes	Default Value	Remarks
FF91 (handle: 0x0062)	Read/ Write	Yes	16	Tv222u-x xxxxxxx (ASCII string with terminator)	Device name, XXXXXXXX for the last four bytes of the physical address
FF92 (handle: 0x0065)	Read/ Write	No	1	1	Bluetooth connection interval: 0: 20ms / 1: **30ms** / 2: 100ms / 3: 200ms / 4:

					300ms / 5: 400ms / 6: 500ms / 7: 1000ms / 8: 2000ms
FF93 (handle: 0x0068)	Read/ Write	Yes	1	5	Set the baud rate of serial ports: 0: 4800 bps / 1: 9600 bps / 2: 19200 bps / 3: 38400 bps / 4: 57600 bps / 5: **115200 bps**/ 6: 256000bps
FF94 (handle: 0x006B)	Write	-	1	no	Channel to control remote reset and recovery: ? - Remote reset control, by writing **0x55** to reset the module - Remote light recovery control, by writing 0x35 to light-recover the module (restoring user data only) and reset - Remote deep recovery control, by writing **0x36** to deep-recover the module (all back to factory settings) and reset
FF95 (handle: 0x006E)	Read/ Write	Yes	1	0	Set the broadcast cycle: 0：20 ms, 1：50 ms 2：100 ms, **3：200 ms,** 4：500 ms, 5：1000 ms, 6：1500 ms, 7：2000 ms, 8：2500 ms, 9：3000 ms, A：4000 ms, B：5000 ms
FF96 (handle: 0x0071)	Read/ Write	Yes	2	0x0000	Set product identification code
FF97 (handle: 0x0074)	Read/ Write	No	1	1	Set the transmission power: 0：-15dBm 1：-12dBm 2：-10dBm 3：-5 dBm

					4：-2 dBm 5：**0 dBm**
FF98 (handle: 0x0077)	Read/ Write	Yes	16	Default broadcast content.	Set customized broadcast data Customizing broadcast data: 0 < n <= 16 See the section “Broadcast Data Setting”

Directions: Module information configuration channel.

- ✧ FF91 is the channel for setting device names. Reading and writing to this channel can obtain and set the module name. The length of the name set must meet the condition: 0 < L<17. **And the name is suggested to end with the terminator (' \ 0 ').** The default name is "Tv222u - XXXXXXXX \ 0" (16 byte), where 222u is the current firmware version number and XXXXXXXX is the last four bytes of the MAC address.
- ✧ FF92 is the channel to set the connection interval. The interval of connection between mobile devices and the module can be set by writing to this channel. Thus the device power consumption and the data throughput can be controlled in a flexible way. In order to raise the connection speed, the setting of connection interval will not be saved. It will always work at the default value (500ms) after power on. Test shows that it takes around 30s to wait when the connection interval is changed from 500 ms to another by iPhone (above iOS 8). But it will be very quick if the connection interval is changed from a high frequency one (ie. 30ms), resulting from BLE protocols.
- ✧ FF93 is the channel to set the baud rate of module serial ports. Baud rate of the module’s universal serial ports can be set by reading and writing to the channel. The new baud rate will take effect in two seconds after setting and can be saved after power-off. The default factory setting is 5 (115200 bps.
- ✧ FF94 is the channel to control the remote reset and recovery.Various controlling functions can be realized by writing different values to the channel.

1. **Writing 0x55 to the channel will light-recover the module. All user settings will be recovered to the factory defaults, including IO output status, PWM initialization mode and user password. Afterwards, the module will be reset.**
2. **Writing 0x36 to the channel will deep-recover the module. All system settings will be recovered to the factory defaults and the module will be reset afterwards.**

- ✧ FF95 is the channel to set the broadcast cycle of the module. Broadcast cycle can be set by reading and writing to this channel.The setting can be saved after power-off. And the default factory setting is 3 (200 ms).
- ✧ FF96 is the channel to set the product identification code of the module, by reading and writing to the channel. The APP can filter and connect to specific product type through this code. The setting can be saved after power-off. And the default factory setting is 0x5253.
- ✧ FF97 is the channel to set the transmission power of the module, by writing to this channel.The setting **cannot be saved** after power-off. And the default factory setting is 5 (0 dBm).
- ✧ FF98 is the channel to set the broadcast contents of the module. Broadcast data can be customized by writing to this channel. The setting can be saved after power-off. When the data is all 0 (16 byte), it is regarded that default broadcast data is used, instead of customized data. (see the section "Broadcast Data Setting").

● Broadcast Data Setting

➢ Default broadcast data:

When the module EN pin is set low, or into the TEST mode (plugged after set low), the module will broadcast at an interval of 200 ms. In the domain of the broadcast data GAP_ADTYPE_MANUFACTURER_SPECIFIC (iOS officially defined programming macro), the following contents are included (9 bytes by default):

{

0x52,0x53, Customizing equipment type code, default setting RS, and can be set with AT command;

0x17,0x51 Module firmware date, Default is 0x17,0x51

0x00,0x00,0x00,0x00, 0x00,0x00, Module MAC Address;

0x05, Module BPS parameter, default is 5, 115200bps;

0x0A, Default module CTS is 10, transmission data after 10mS when CTS pull down

0x03, Default module broadcast interval is 200mS;

0x05, Default module transmission function is 0dBm;

0x01, Default module connection interval is 30mS;

0x02, Default anti hijacking key deadline is 10s;

}

Note: the data in broadcast will be the original setting value of the first release, and will not change as the AT command or APP set new parameter.

➢ Customize broadcast data:

While using the AT command to customize the broadcast content, the maximum length is 16 bytes (Blue font part). in the broadcast data GAP_ADTYPE_MANUFACTURER_SPECIFIC the field will contain the following content, with length of 2 + n bytes:

{

0x00,0x00, Custom coding equipment type, with default value 00 00, can be set by the AT command;

Data [n], Custom broadcast data, n < = 16;

}

Note: The broadcast data can be modified by the AT command and saved after power-off. After power on again, the customized broadcast data during last time will be used. If customized broadcast data is set all 0 (16 byte), the customized broadcast will not be used but the system default broadcast contents. To avoid the too long broadcast data to cause extra power consumption, you can also set the customized broadcast data to be1 byte of any value.

● iOS APP Programming Reference

The module is always to broadcast as slave, waiting for the smart phone to scan and connect as master. The scanning and connection is usually completed by the APP. Due to the particularity of the BLE protocol, there is no need to scan and connect Bluetooth LE devices in the system settings of the smart phone. Smart devices are responsible for BLE connection, communication, disconnection, and etc. And usually it is implemented by the APP.

Regarding BLE programming in iOS, the key point is the **reading, writing and enabling notify switch** for the value of Characteristic (also called channel in this article). To read and write to the channel can realize the direct control on the direct-drive mode functions of the module and no extra CPU is needed. Typical functions that are involved are as follows:

```
/*!
 *  @method writeValue:forCharacteristic:withResponse:
 *  @param data The value to write.
 *  @param characteristic The characteristic on which to perform the write operation.
 *  @param type The type of write to be executed.
 *  @discussion Write the value of a characteristic.
 *      The passed data is copied and can be disposed of after the call finishes.
 *      The relevant delegate callback will then be invoked with the status of the request.
 *  @see peripheral:didWriteValueForCharacteristic:error:
 */
- (void)writeValue:(NSData *)data forCharacteristic:(CBCharacteristic *)characteristic type:(CBCharacteristicWriteType)type;
```

Note: to write to a characteristic

```
NSData *d = [[NSData alloc] initWithBytes:&data length:mdata.length];
        [p writeValue:d
 forCharacteristic:c
             type:CBCharacteristicWriteWithoutResponse];
```

```
/*!
 *  @method readValueForCharacteristic:
 *  @param characteristic The characteristic for which the value needs to be read.
 *  @discussion Fetch the value of a characteristic.
 *      The relevant delegate callback will then be invoked with the status of the request.
 *  @see peripheral:didUpdateValueForCharacteristic:error:
 */
- (void)readValueForCharacteristic:(CBCharacteristic *)characteristic;
```

Note: to write to a characteristic

```
      [p readValueForCharacteristic:c];
```

```
/*!
 *  @method setNotifyValue:forCharacteristic:
```

```
 *  @param notifyValue The value to set the client configuration descriptor to.
 *  @param characteristic The characteristic containing the client configuration.
 *  @discussion Ask to start/stop receiving notifications for a characteristic.
 *      The relevant delegate callback will then be invoked with the status of the request.
 *  @see peripheral:didUpdateNotificationStateForCharacteristic:error:
 */
- (void)setNotifyValue:(BOOL)notifyValue forCharacteristic:(CBCharacteristic *)characteristi
c;
```

Note: to open a characteristic notify enable switch.

[self setNotifyValue:YES forCharacteristic:c];//open notify enable switch.

[self setNotifyValue:NO forCharacteristic:c]; //close notify enable switch.

```
/*
 *  @method didUpdateValueForCharacteristic
 *  @param peripheral Pheripheral that got updated
 *  @param characteristic Characteristic that got updated
 *  @error error Error message if something went wrong
 *  @discussion didUpdateValueForCharacteristic is called when CoreBluetooth has updated a
 *  characteristic for a peripheral. All reads and notifications come here to be processed.
 *
 */
- (void)peripheral:(CBPeripheral *)peripheral didUpdateValueForCharacteristic:(CBCharacteri
stic *)characteristic error:(NSError *)error
```

Note: after each reading operation, this callback function will be performed. The application layer saves the read data in the function.

About the details of scanning, connecting, and other communication operations, please refer to the test App source code (BLE Transmission Module v1.29) for transparent transmission in IOS, by RF-Star Technology, in which it realizes, for FFE9 and FFE4, the operations of data transmit from BLE to serial port and from serial port to BLE characteristics (notify and write). Other controls on direct-drive functions are similar, all by reading or writing to certain characteristic. The only difference is the characteristic UUID and the bytes of reading and writing operations.

● Testing by APP

The testing tool(APP) of IOS platform can be download in App Store. Open the the App Store of operation system (above IOS6) and search for *Module Tools*, then download and install to test. You have 3 ways to install the App:

1. Search and download from APP Store, you need to apply for a APP account of iPhone (free).
2. Use the source code to download to your iPhone devices(should be iPhone developer users);
3. Breakout your iPhone devices to download the IPA file(just like windows' s exe)from RF-star's website. Then use the PP assistant to install.

Figure 1 shows the user interface when open the APP, click the blue icon at the right corner, the scanned devices will be in the list(sometimes need to open the Bluetooth), as figure 2 shows: Click a device, start to connect, home page will show up after connected(as figure 3 shows).

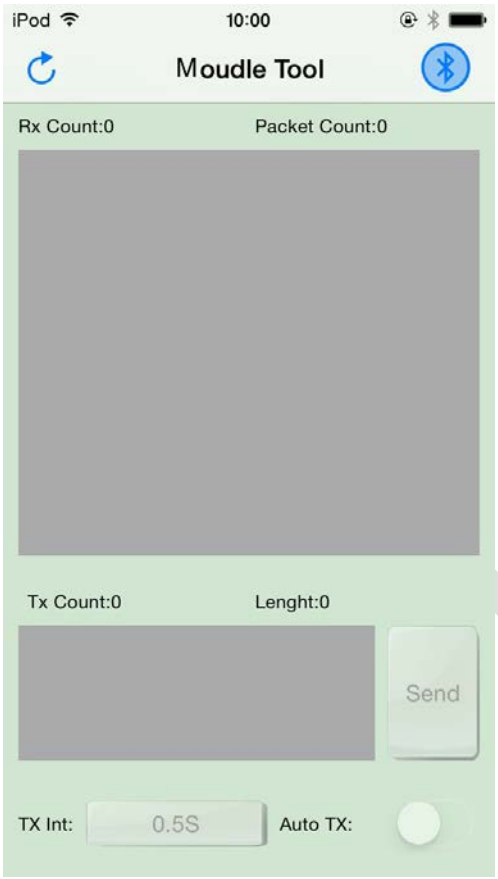


Figure 1

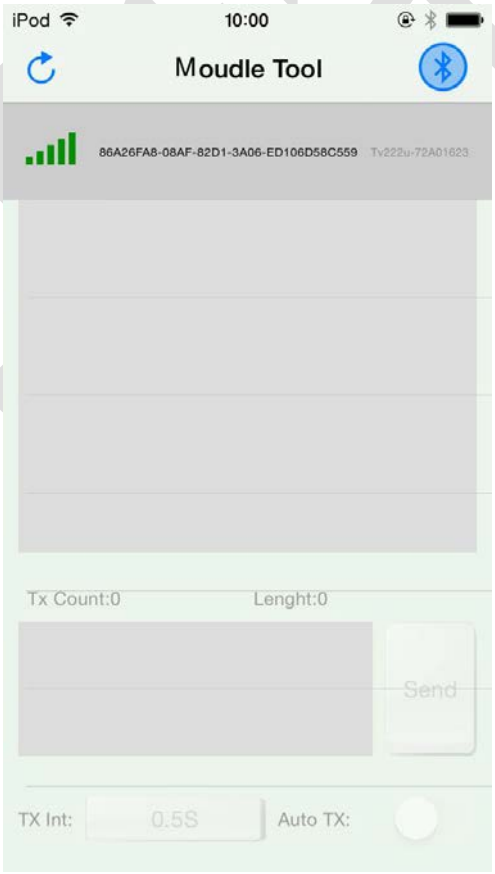


Figure 2

If the serial port of the module is ready(connected the host CPU or serial terminal), then it start to work, can do the transceiver test manually or automatically.

As figure 4 shows: Rx is the data packet transmitted by the host CPU or by UART when it suspend,

Tx is the data packet sent by iPhone.

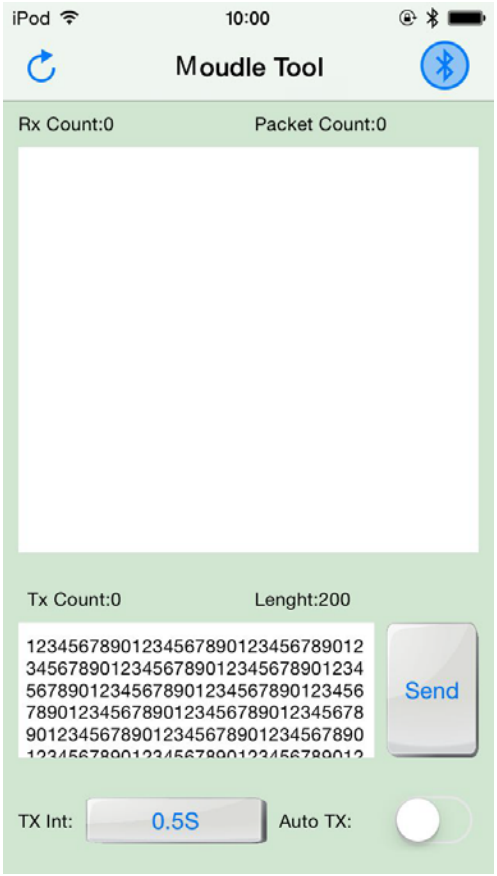


Figure 3

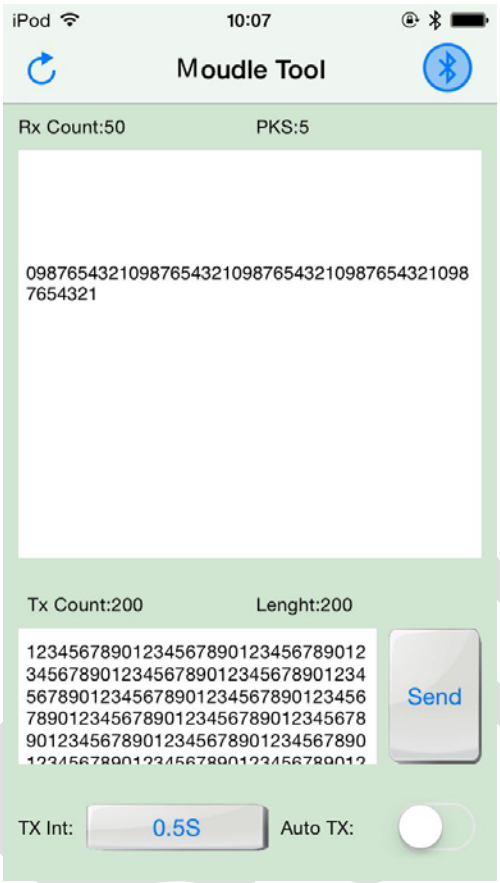


Figure 4

Note：If use the UART terminal to test, the data of the serial terminal should be sent to the mobile and keep BRTS down to prevent the module entering the sleeping mode.

For the IOS programming: Mobile devices can write the respective UUID through B channel when do the transmission according to the BLE protocol.The transmission from module to mobile devices will start by notifying the mobile devices, so corresponding UUID enabled notification of A channel need to be opened after booting up the APP, then the data packet received from module serial port will send to mobile devices automatically. For relative materials, contact us.。

● Testing by USB Dongle & Btool

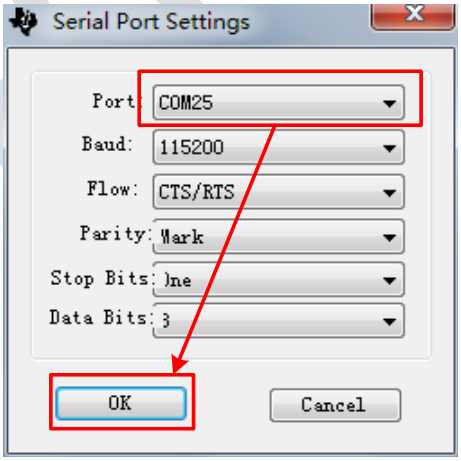
BLE module can use the official USB Dongle of TI (CC2540 MiniDK) to simulate mobile phone and work with C:\Texas Instruments\BLE-CC254x-1.3.2\Projects\Btool\BTool.exe in the installation directory to start the Bluetooth communication test.

This USB Dongle should use the project item under the C:\Texas Instruments\BLE-CC254x-1.3.2\Projects\ble\HostTestApp\CC2540 to compile and download into the USB Dongle. For specific operation of BTOOL, please refer to the official instruction: CC2540 Mini Development Kit User's Guide (Rev. B).pdf

➢ Connecting BLE Module

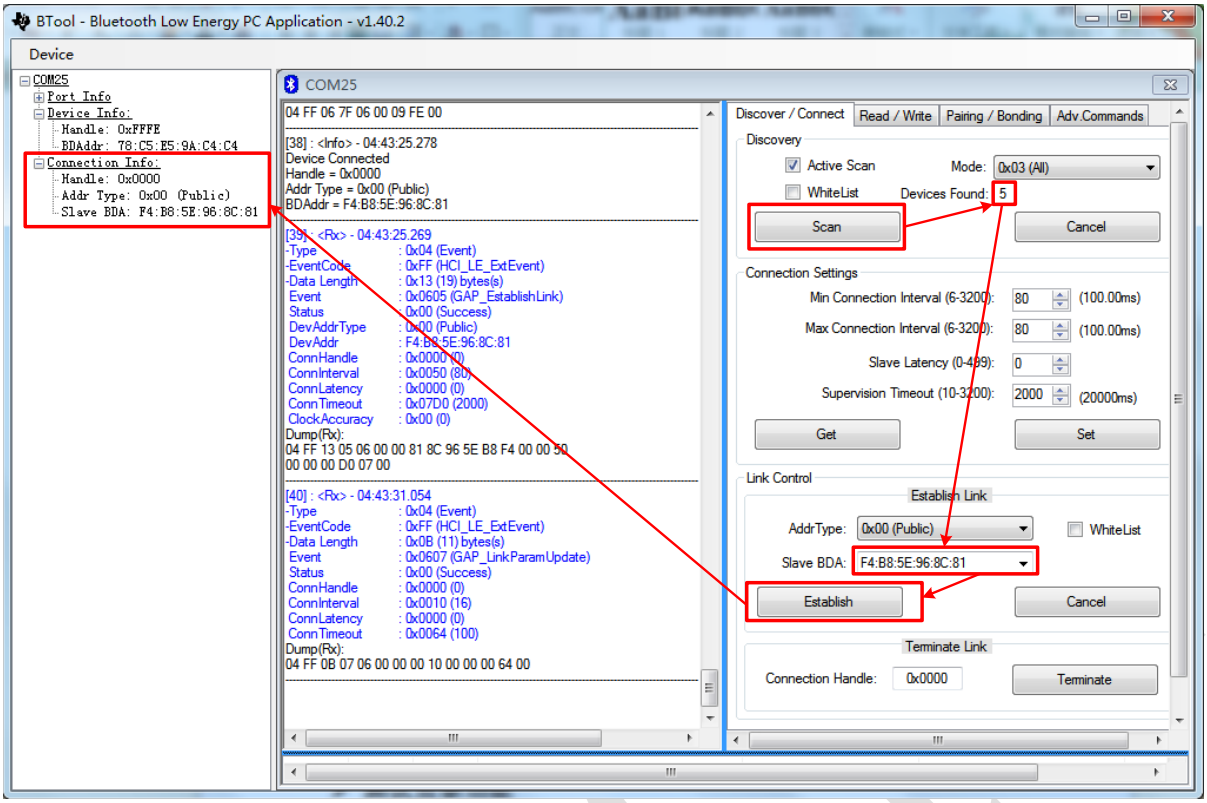
Connection between USB Dongle and module is the base of communication, steps for scanning and connecting are as bellow:

1. Open the Project file under the C:\Texas Instruments \ BLE-CC254x-1.3.2 \ Projects \ ble\ HostTestApp, compile and download into the USB Dongle;
2. Module Power on（3 ~ 3.3V）;
3. Pull module EN down to ground, then the module start to broadcast;4、Insert the USB Dongle to PC USB, a serial device will show up in the hardware management.(eg. COM25)
4. Open C:\Texas Instruments\BLE-CC254x-1.3.2\Projects\BTool\BTool.exe;
5. Menu: Device -> New Device，choose the serial port in step and choose default setting, click OK;



6. Scan & Connect: Scan as the arrow directs then connect(F4:B8:5E:96:8C:81 is the physical address of module). Please make sure connected the targeted module.

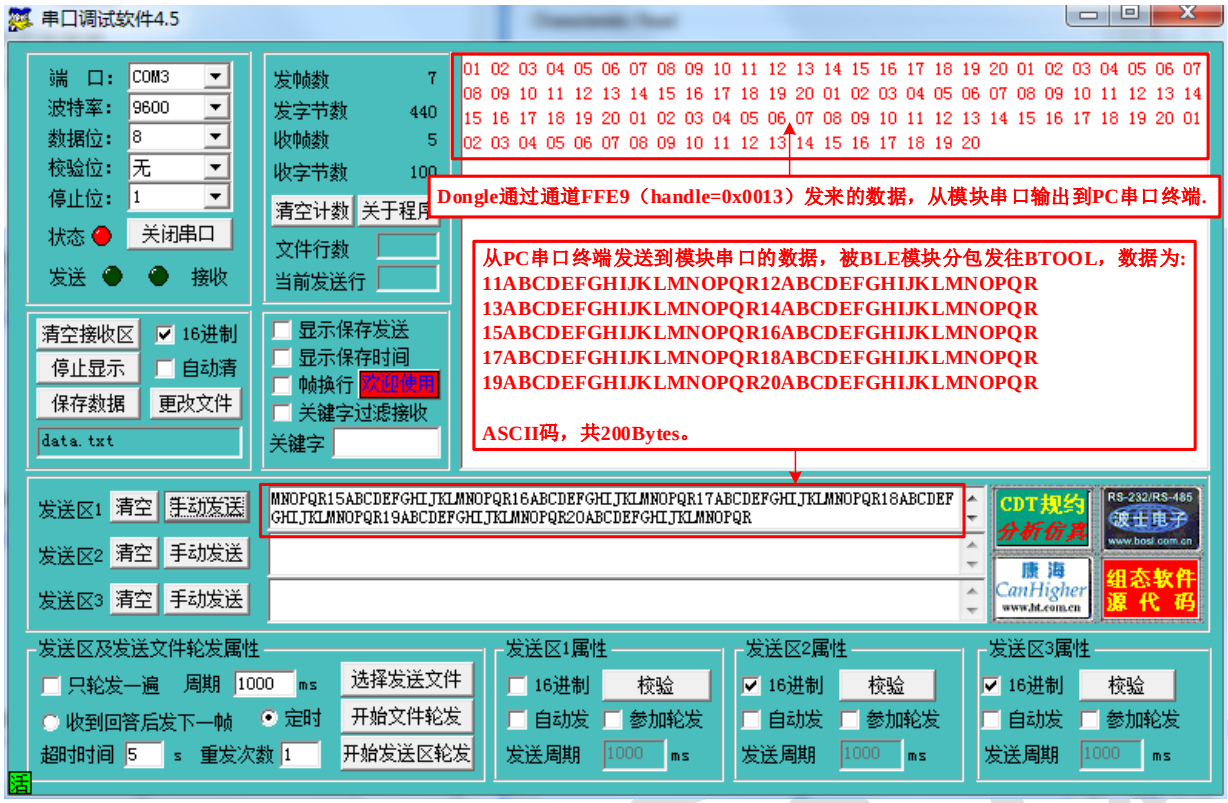
7. After that, connected module info will show in the left.Connection Info



8. Now connected successfully, then can start the test of Bluetooth UART transmission function.(Transparent transmission)

9. Shortcut of PC-end connecting transparent transmission module.

Note: BRTS must be pulled down, or the serial data can not be received by module.


串口调试软件4.5
端 口: COM3
波特率: 9600
数据位: 8
校验位: 无
停止位: 1
状态
关闭串口
发送
接收
发帧数 7
发字节数 440
收帧数 5
收字节数 100
清空计数
关于程序
文件行数
当前发送行
清空接收区
16进制
停止显示
自动清
保存数据
更改文件
data.txt
显示保存发送
显示保存时间
帧换行
欢迎使用
关键字过滤接收
关键字
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 01 02 03 04 05 06 07
08 09 10 11 12 13 14 15 16 17 18 19 20 01 02 03 04 05 06 07 08 09 10 11 12 13 14
15 16 17 18 19 20 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 01
02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20
Dongle通过通道FFE9（handle=0x0013）发来的数据，从模块串口输出到PC串口终端.
从PC串口终端发送到模块串口的数据，被BLE模块分包发往BTOOL，数据为:
11ABCDEFGHIJKLMNOPQR12ABCDEFGHIJKLMNOPQR
13ABCDEFGHIJKLMNOPQR14ABCDEFGHIJKLMNOPQR
15ABCDEFGHIJKLMNOPQR16ABCDEFGHIJKLMNOPQR
17ABCDEFGHIJKLMNOPQR18ABCDEFGHIJKLMNOPQR
19ABCDEFGHIJKLMNOPQR20ABCDEFGHIJKLMNOPQR
ASCII码，共200Bytes。
发送区1 清空 手动发送
发送区2 清空 手动发送
发送区3 清空 手动发送
MNOPQR15ABCDEFGHIJKLMNOPQR16ABCDEFGHIJKLMNOPQR17ABCDEFGHIJKLMNOPQR18ABCDEF
GHIJKLMNOPQR19ABCDEFGHIJKLMNOPQR20ABCDEFGHIJKLMNOPQR
CDT规约
分析仿真
RS-232/RS-485
波士电子
www.bosl.com.cn
康海
CanHigher
www.ht.com.cn
组态软件
源代码
发送区及发送文件轮发属性
只轮发一遍 周期 1000 ms
选择发送文件
收到回答后发下一帧
定时
开始文件轮发
超时时间 5 s 重发次数 1
开始发送区轮发
发送区1属性
16进制 校验
自动发 参加轮发
发送周期 1000 ms
发送区2属性
16进制 校验
自动发 参加轮发
发送周期 1000 ms
发送区3属性
16进制 校验
自动发 参加轮发
发送周期 1000 ms

• Master reference code (transparent- transmission)

Logical relation: The two IOs – BCTS and BRTS – are used for notifying and controlling of transceiving. These two IOs are normally set high, and triggered when setting low. If the module needs to send data, set BCTS low to inform MCU to receive. If MCU needs to send data, set BRTS low to inform the module to receive. Schematic codes are as follows:

```
void main(void)
{
   EN = 0 ;                                          //Enable,start broadcast
   while(!BLEMoudleAck("TTM:OK\r\n\0"));             //Waiting for Phone scan,Connecting
                                                     //Waiting for the connection is
                                                     successful,Also can add wait time limit.

                                                     //Also can judge connection prompt signal
                                                     level
  BRTS = 0;                                          //Low BRTS, Notice RSBRS02AA module is
                                                     ready to receive
   halMcuWaitMs(2);                                  //Delay for 2ms
   UARTWrite( HAL_UART_PORT_0, "TTM:CIT-100ms", 14);
                                                     //modify connect interval,from the serial port
                                                     to be confirmed.
 halMcuWaitMs(5);                                    //Delay for 5ms to ensure the data has been issued
   BRTS = 1;                                         //RTS high and sent
    while(!BLEMoudleAck("TTM:OK\r\n\0"));            //Waiting to set success,also can add wait time
                                                     limit.
   while(1){                                         //Loop transceiver test.
     while(1){
      if(BCTS == 0){                                 //Testing,if BCTS low is ready to receive.
         while(BCTS==0);                             //Waiting to be sent,but also timed wait.
         if(UARTRead(uartBuffer) == SUCCESS)         //A serial port to read data.
         {... ...}                                   //Use data.
      }
      BRTS = 0;                                      //RTS low,and notice RSBRS02AA to receive.
      halMcuWaitMs(2);                               //Delay for 2ms
      send_TX("1234567890");                         //Send any data.（Within 200byte）
          halMcuWaitMs(5);                           //Delay for 5ms to ensure the data has been
                                                     issued
      BRTS = 1;                                      //RTS high,sent
          halMcuWaitMs(20);                          //Delay again to send next packet,and delay
                                                     depending on the packet size.

      }
     }
}}
```

● Recommended Operating Conditions

Notes： Make sure to operate within the range of condition parameters listed below. Long time operating outside of the range will impact the reliability of the module.

(1) The operating temperature is subject to the change of the crystal's frequency;

(2) To ensure the RF performance, the ripple wave on the source must be less than ±300mV。

Identification	Condition	Minimum	Typical Value	Maximum	Unit
Source & IO	Battery Mode	1.8	3.3	3.8	V
Operating Temperature	/	-40	25	85	℃
Environment Temperature Amplitude		-20		20	℃/Min

● Reflow Conditions

1. Heating mode：conventional convection or IR convection；
2. Times allowed to reflow: 2 times, based on the reflow conditions below（figure 7）；
3. Temperature curve: the reflow should be in accordance with the temperature curve shown below (figure 7);
4. Highest temperature：245°C

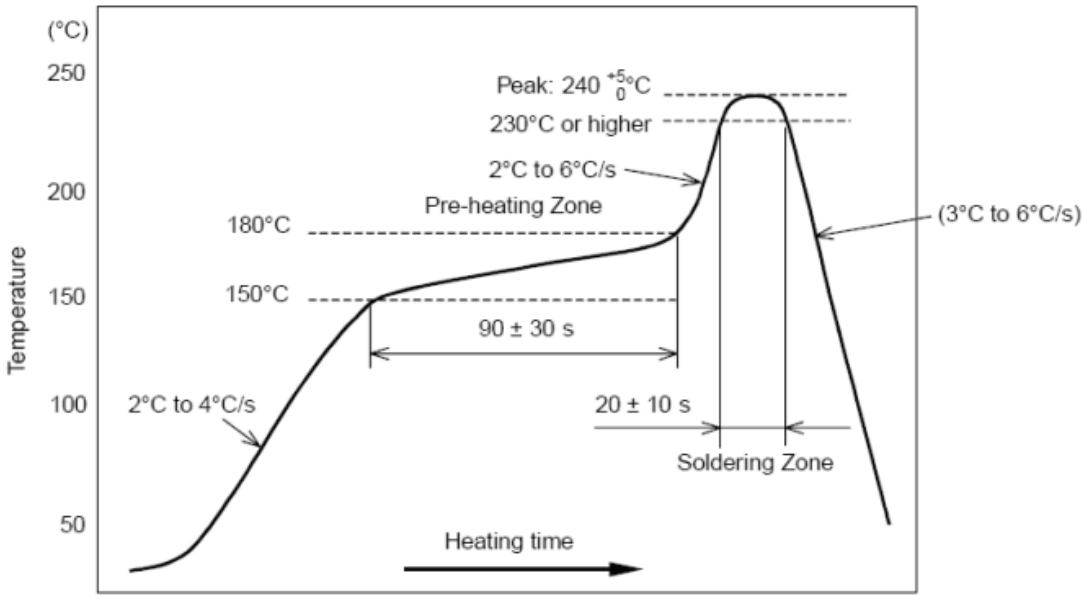


Figure 12 Parts' heat-resistance temperature curve for welding(welding point）

Warnings on Static Discharge

Since the module may be damaged by static discharge, RF-star recommend to handle with it with three precautions below:

1, According to the anti-static measures, bare hands are not allowed to touch modules.

2, Modules must be placed in anti- static areas.

3, In product design, anti-static circuitry should be arranged at where high voltage or high frequency input is introduced. Static discharge may incur impact to the device ranging from slight performance downgrade to total failure.

● Advices on Application Solutions

Telemedicine / Telehealth , Medical Patient Monitoring , Human Interface Devices (Keyboard, Mouse, Remote control) , Sports and leisure equipment , Mobile phone accessories , Remote controls , Consumer Electronics , Remote monitoring and control , Health Care and Medical , Smart Grid , Automated Meter Reading (AMR) , Home/Building Automation , Machine-to-Machine (M2M) , Wireless Sensor Networks , Wireless Alarms and Security , Lighting and HVAC control , Proximity and out of range detection (iBeacon)

● Appendix B: RF Parameters Test Report

1) Transmission performance parameters

Except for some specific scenarios, default conditions for parameters test are: VDD=3.3V，TA = 25°C，RBW=100K，VBW=300K，Sweep Time 为 100ms.

1.Frequency Range

1, Frequency Range
2402-2480MHz

2.Transmit power

Center Frequency（MHz）	Transmission Power（dBm）	Tolerance（dBm）	Result
2402	-0.19	0dBm（±2dBm）	PASS
2404	-0.17		PASS
2406	-0.13		PASS
2408	-0.08		PASS
2410	-0.05		PASS
2412	0.00		PASS
2414	0.04		PASS
2416	0.07		PASS
2418	0.12		PASS
2420	0.16		PASS
2422	0.19		PASS
2424	0.23		PASS
2426	0.27		PASS
2428	0.29		PASS
2430	0.33		PASS
2432	0.35		PASS
2434	0.37		PASS
2436	0.39		PASS
2438	0.41		PASS

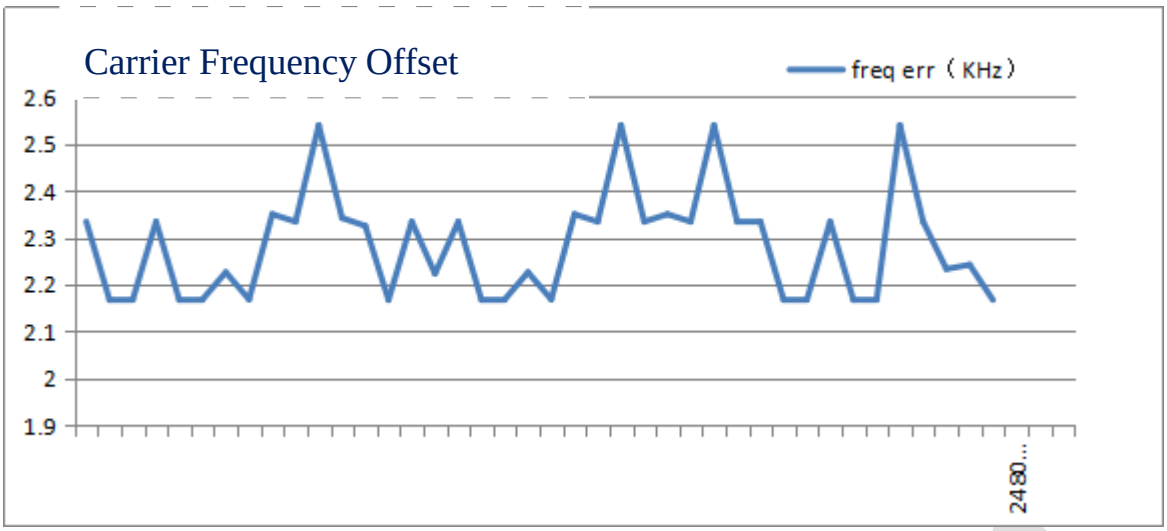
2440	0.41		PASS
2442	0.42		PASS
2444	0.44		PASS
2446	0.44		PASS
2448	0.46		PASS
2450	0.46		PASS
2452	0.46		PASS
2454	0.45		PASS
2456	0.44		PASS
2458	0.43		PASS
2460	0.40		PASS
2462	0.39		PASS
2464	0.37		PASS
2466	0.34		PASS
2468	0.31		PASS
2470	0.28		PASS
2472	0.25		PASS
2474	0.22		PASS
2476	0.17		PASS
2478	0.13		PASS
2480	0.09		PASS

3.Frequency Error

Wave form output =CW

Center Frequency（MHz）	Frequency Deviation（KHz）	FCC Permissible Deviation（KHz）	Result
2402	2.334	±40KHz	PASS
2404	2.167		PASS
2406	2.168		PASS
2408	2.334		PASS
2410	2.167		PASS
2412	2.168		PASS
2414	2.226		PASS

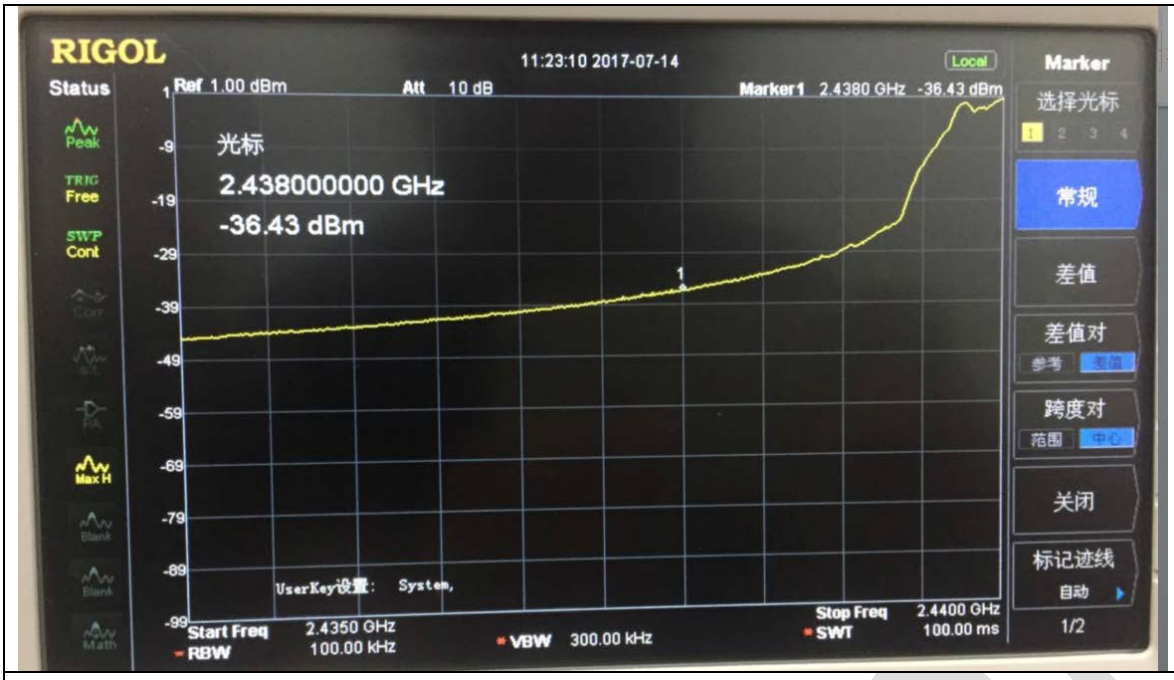
2416	2.168		PASS
2418	2.35		PASS
2420	2.334		PASS
2422	2.54		PASS
2424	2.342		PASS
2426	2.325		PASS
2428	2.167		PASS
2430	2.334		PASS
2432	2.223		PASS
2434	2.334		PASS
2436	2.167		PASS
2438	2.168		PASS
2440	3.666		PASS
2442	2.168		PASS
2444	2.35		PASS
2446	2.334		PASS
2448	2.54		PASS
2450	2.334		PASS
2452	2.35		PASS
2454	2.334		PASS
2456	2.54		PASS
2458	2.334		PASS
2460	2.334		PASS
2462	2.167		PASS
2464	2.168		PASS
2466	2.334		PASS
2468	2.167		PASS
2470	2.168		PASS
2472	2.54		PASS
2474	2.334		PASS
2476	2.233		PASS
2478	2.242		PASS
2480	2.167		PASS



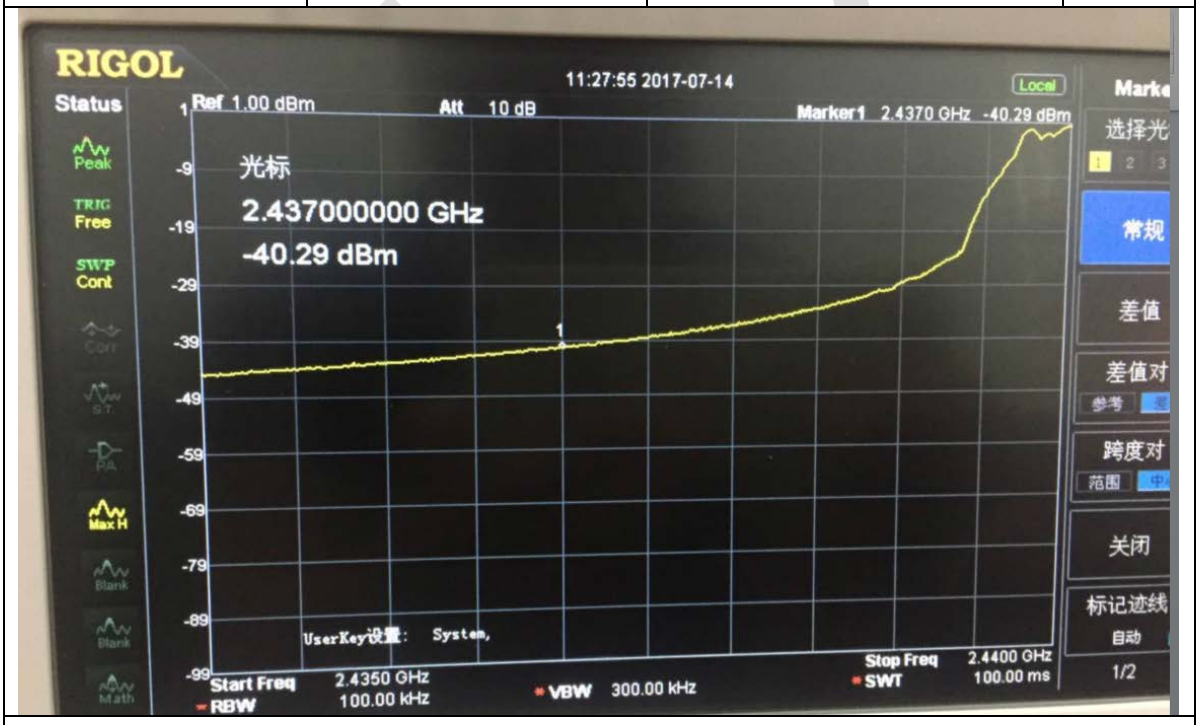
4.In-band Spurious emission.

Test Condition: PTX=0dBm。

Center Frequency（MHz）	Actual Data（dBm）	RF-PHY.TS.4.2.2 Standard Range for Reference（dBm）	Result
2402±2MHz	-43.21	≤-20dBm	PASS
2440±2MHz	-42.88		
2480±2MHz	-43.32		

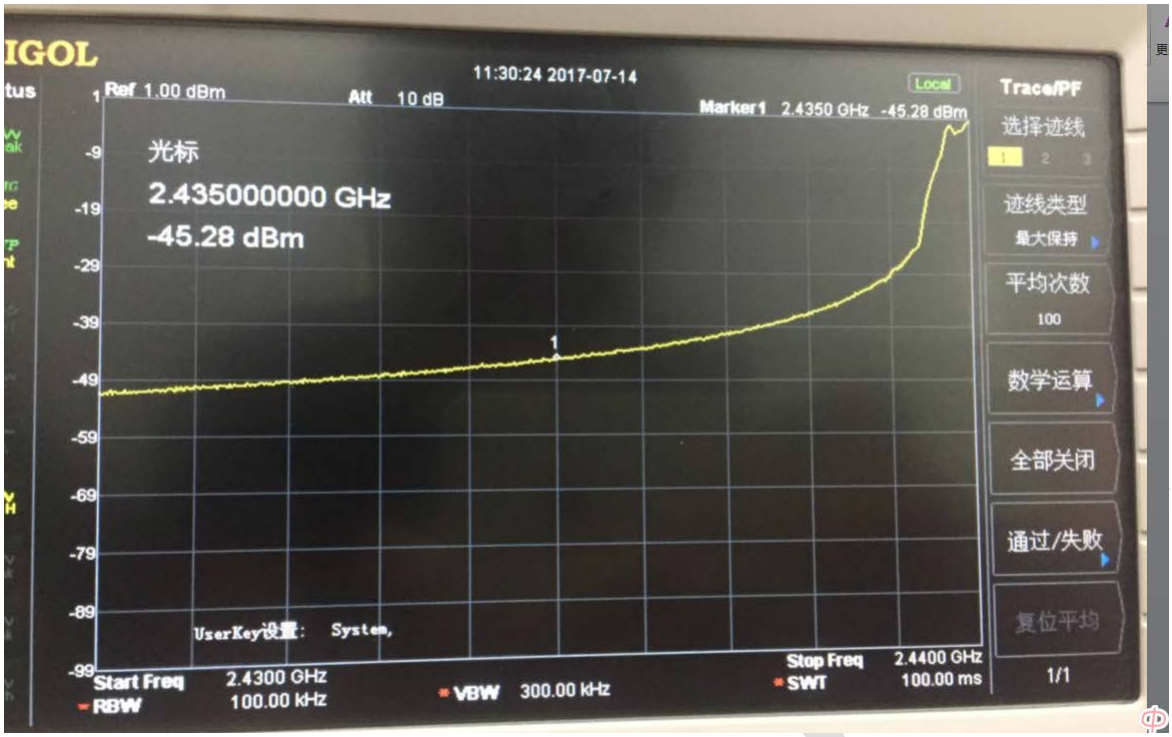


Center Frequency（MHz）	Actual Data（dBm）	RF-PHY.TS.4.2.2 Standard Range for Reference（dBm）	Result
2402±3MHz	-47.24	≤-30dBm	PASS
2440±3MHz	-46.25		
2480±3MHz	-47.38		



Center Frequency	Actual Data（dBm）	RF-PHY.TS.4.2.2 Standard	Result

（MHz）		Range for Reference （dBm）	
2402±5MHz	-43.04	≤-30dBm	PASS
2440±5MHz	-42.3		
2480±5MHz	-42.57		



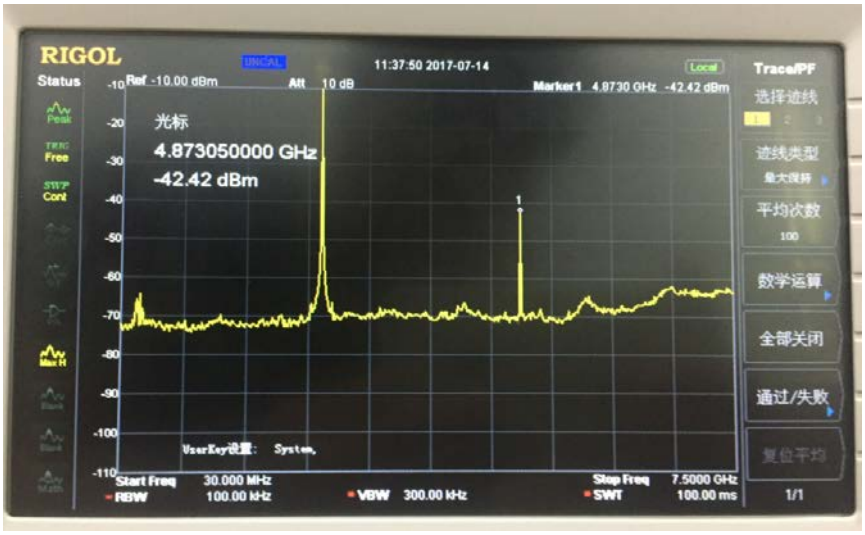
5.Out-band Spurious Emission

·CE External stray limit of conduction harmonic wave band

Test Condition：PTX=0dBm。The test is based on CE standard EN300328V1.8.1.

Band	Frequency (MHz)	Level(dBm)	standard	Unit	Result
	Record(Max)	level /dBm	EN300 328 Spec	RBW/VBW	
30MHz~47MHz	32	-80	-36	100k/300k	PASS
47MHz~74MHz	64	-69.5	-54	100k/300k	PASS
74MHz~87.5MHz	75	-70.39	-36	100k/300k	PASS
87.5MHz~118MHz	96	-71	-54	100k/300k	PASS
118MHz~174MHz	128	-65	-36	100k/300k	PASS
230MHz~470MHz	256	-59	-36	100k/300k	PASS
470MHz~862MHz	480	-71	-54	100k/300k	PASS
862MHz~1GMHz	864	-69	-36	100k/300k	PASS

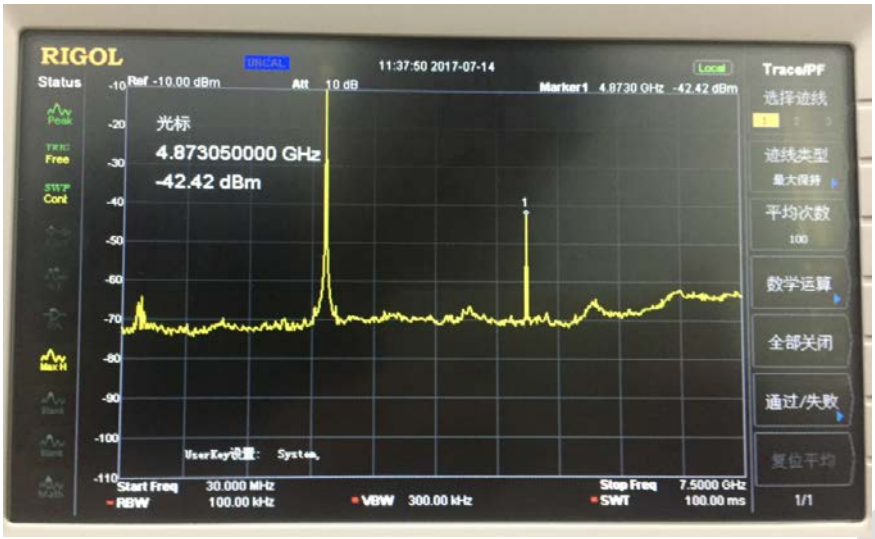
1GHz~2.36GHz	2.30	-48.6	-30	1M/3M	PASS
2.5235GHz~12.75GHz	2.523	-41.66	-30	1M/3M	PASS



·FCC External stray limit of conduction harmonic wave band

The test is based on FCC standard PART 15.247

Carrier Frequency（MHz）	Harmonics Frequency（MHz）	FCC Requirements: < -41.2dBm		Result
		Actual Measurement（dBm）	Allowance（dB）	
2402	4810	-42	0.8	PASS
	7215	-65	23.8	PASS
2440	4880	-42.3	1.1	PASS
	7320	-64	22.8	PASS
2480	4960	-43.5	2.3	PASS
	7440	-64.12	22.92	PASS



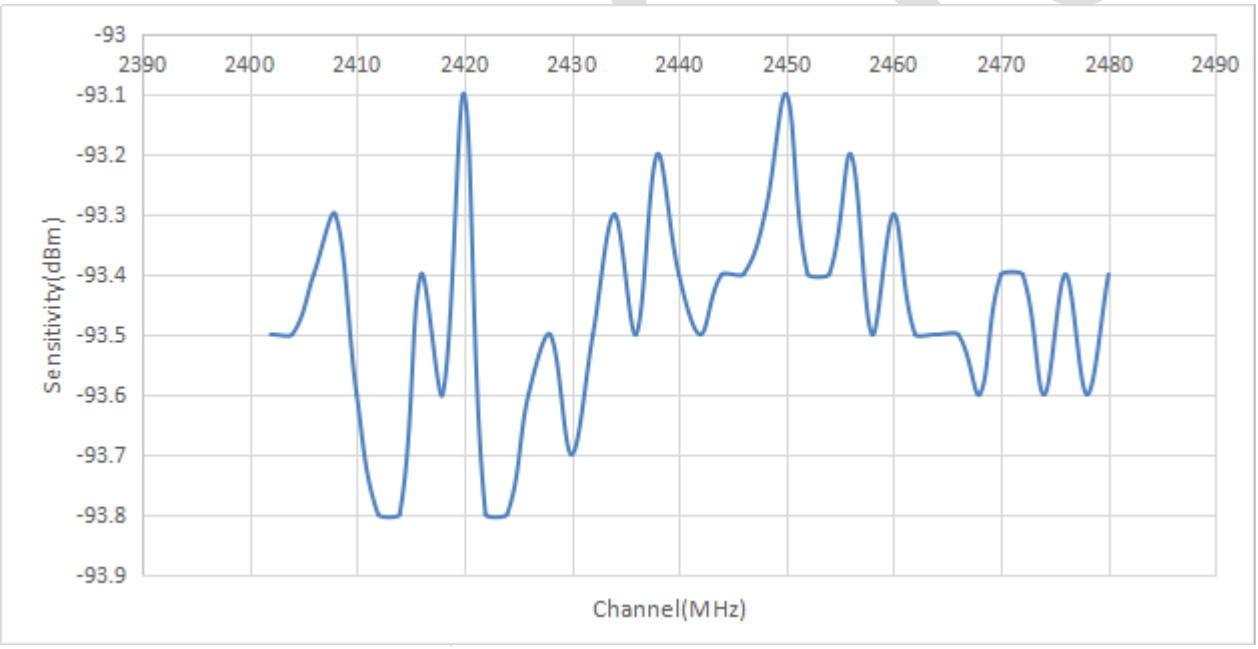
2) Reception Performance Parameters

Except for some specific scenarios, default conditions for parameters test are: VDD=3.3V，TA = 25°C，RBW=100K，VBW=300K。

1. Sensitivity（Packet loss rate PER=1%）

Center Frequency（MHz）	Sensitivity（dBm）	Range of RS02A Data Brochure Sensitivity（dBm)	Result
2402	-93.5	-94dBm	PASS
2404	-93.5		PASS
2406	-93.4		PASS
2408	-93.3		PASS
2410	-93.6		PASS
2412	-93.8		PASS
2414	-93.8		PASS
2416	-93.4		PASS
2418	-93.6		PASS
2420	-93.1		PASS
2422	-93.8		PASS
2424	-93.8		PASS
2426	-93.6		PASS
2428	-93.5		PASS
2430	-93.7		PASS
2432	-93.5		PASS
2434	-93.3		PASS
2436	-93.5		PASS
2438	-93.2		PASS
2440	-93.4		PASS

2442	-93.5		PASS
2444	-93.4		PASS
2446	-93.4		PASS
2448	-93.3		PASS
2450	-93.1		PASS
2452	-93.4		PASS
2454	-93.4		PASS
2456	-93.2		PASS
2458	-93.5		PASS
2460	-93.3		PASS
2462	-93.5		PASS
2464	-93.5		PASS
2466	-93.5		PASS
2468	-93.6		PASS
2470	-93.4		PASS
2472	-93.4		PASS
2474	-93.6		PASS
2476	-93.4		PASS
2478	-93.6		PASS
2480	-93.4		PASS



2. C/I Blocking Performance

Center Frequency （MHz）	Level Input (dBm)	Frequency deviation(MHz)	CI Level(dB)
2402	-67	-3	-33
	-67	-2	-23

	-67	-1	-5
	-67	0	5
	-67	1	-5
	-67	2	-34
	-67	3	-46
2440	-67	-3	-33
	-67	-2	-24
	-67	-1	-4
	-67	0	5
	-67	1	-5
	-67	2	-38
2480	-67	-3	-46
	-67	-2	-23
	-67	-1	-4
	-67	0	5
	-67	1	-5
	-67	2	-35
	-67	3	-46