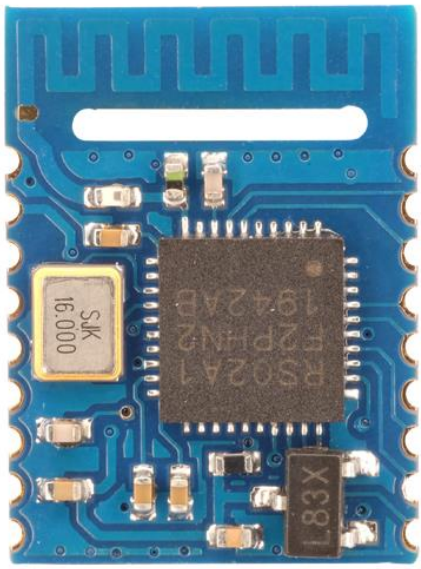
RS02A1-B Bluetooth Low Energy Slave Module and Protocol

Version: V5.11u

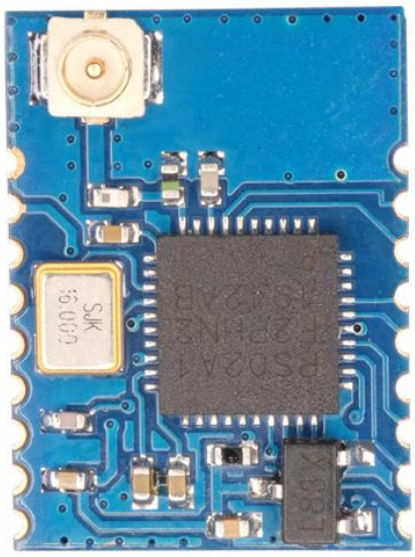
(Transparent Transmission & Direct-Driven)

Shenzhen RF-star Technology Co., Ltd.

Dec. 2nd, 2020



RSBRS02ABR



RSBRS02ABRI

Table of Contents

Table of Figures

Table of Tables

1 Summary

RF-star's single-mode Bluetooth Smart® modules are high-performance and low-power Bluetooth 4.2 / 5.0 RF SoC modules that incorporate the RF-star RS02A1-B transceiver chipset among the industry's smallest packages.

These embedded Bluetooth 4.2 / 5.0 modules incorporate an LC balun, one crystal, the required RF matching filter for regulatory compliance, which makes the module filtering on select digital lines for better noise reduction and sensitivity. RF-star can provide customers with the integration of the entire profiles, applications, radio and BLE protocol stack. With compliance to Bluetooth Low Energy standard, the RF-star's modules enable the creation of a new market for tiny, cost-effective and power-efficient wireless consumer products such as watches, medical sensors, mice, TV remote controls and fitness trainers.

- Bluetooth 4.2 / 5.0 single mode compliant ISM 2.4 GHz module
- Utilizes RS02A1-B SoC with 256 KB Flash, 32 KB SRAM, 64 KB ROM
- Supports slave mode
- Can be externally controlled via simple ASCII AT commands over the UART or programmed with custom applications embedded in the module
- RSSI monitoring for proximity applications

2 Overview

The Bluetooth LE modules which are mentioned in this document can work in bridge mode (transparent transmission mode) and direct-driven mode.
After the module starts to broadcast, smart phone with specific APP running will scan and pair with it. When connection is successful, the smart phone can monitor and control the module through Bluetooth protocol.

In bridge mode, user CPU can communicate with the mobile device bi-directionally through the serial ports of modules. Users can also manage and control certain communication parameters through specific AT commands of serial ports. The detailed meaning of the user data is defined by the upper layer application. Mobile devices can write the module through the APP. And the data written will be sent to the user CPU through serial ports. Then the module will transmit the data packet from user CPU to the mobile devices automatically. Under the development in this mode, the user needs to undertake the code design for master CPU and the APP for mobile devices.

In direct-driven mode, users take simple peripheral expansion to the module. And APP drives the module directly through BLE protocol to implement the monitoring and control of the module by mobile devices. In this mode, users only need to do the code design for mobile devices.

3 Features

1. Easy to use, no need of any application experience of Bluetooth protocol stack. It supports Bluetooth 5.0 protocol stack, and has twice the speed (2 Mbps physical layer) compared with 4.2.
2. Support manual adjustment of physical layer bandwidth to improve communication distance and speed
3. UART design for user interface, full-duplex bi-directional communication, and support the minimum baud rate of 9600 bps.
4. Default connection interval of 20 ms, which makes quick connection and enhance the compatible stability of Android and iOS phones.
5. Through AT command, support modify Service UUID.
6. Through AT command, support software reset.
7. Through AT command, support acquire MAC address and modify MAC address (Modified MAC address is effective after reset).
8. Through AT command, support the adjustment of Bluetooth connection interval and the control of different transmit rates (dynamic power consumption adjustment).
9. Through AT command, support the adjustment of the transmit power and broadcast interval, the customization of broadcast data and product ID, the modification of the serial port baud rate and module names.
10. The length of the UART data packet can be any value lower than 500-Byte (automatic sub-packet of large ones).
11. High-speed transparent transmission rate is maximum to 8.2 K/s and the stable rate is 5 K/s (work in iOS10 and the system below).
12. Through APP, support the modification of module name, UART baud rate and product ID, support the customization of broadcast packet and cycles.
13. Through APP, support the remote reset of module and the configuration of transmit power.
14. Through APP, support the adjustment of Bluetooth connection interval (dynamic power consumption adjustment).
15. Support the configuration, modification and recover of anti-hijacking password, and prevent from the malicious connection of a third party. This function can be ignored. The independent result notification of password operation to simplify the APP programming.
16. Support factory reset by APP, recover default password by single pin connected to the ground for 5 s long press, and recover factory settings by single pin connected to the ground for 20 s long press.
17. Support string prompts of module real-time system status, including MAC address, connection interval, broadcast cycle, data delay time, baud rate, product ID, anti-hijacking password enable and other configuration information.
18. Support the light recovery and deep recovery modes, which can recover user data flexibly while reserve the essential configurations.
19. Extremely low power in standby mode (current of 2.7 μA for RS02A1-B SoC), and the measured power consumption data is as follows:

Table 1. Power Consumption of RS02A1-B

Event	Average Current (Enable EN Internal Pull-up)	Average Current (Disable EN Internal Pull-up)	Testing Conditions / Remark
Sleeping	5.41 μA		---
Broadcast	228.76 μA	149.0 μA	Broadcast cycle: 200 ms
Broadcast	136.43 μA	60.48 μA	Broadcast cycle: 500 ms
Broadcast	107.92 μA	32.73 μA	Broadcast cycle: 1000 ms
Broadcast	83.5 μA	7.56 μA	Broadcast cycle: 5000 ms
Connection Event	346.04 μA	268.34 μA	Connection interval: 30 ms
Connection Event	157.57μA	81.07 μA	Connection interval: 100 ms

Notes:

Above is the measured sampling data of module RSBRS02ABR and for reference only. If the lower power consumption is expected, connection interval or broadcast cycle can be appropriately increased, as shown in the chapters of module parameter configuration and AT command.

4 Schematic Diagram of Working Mode

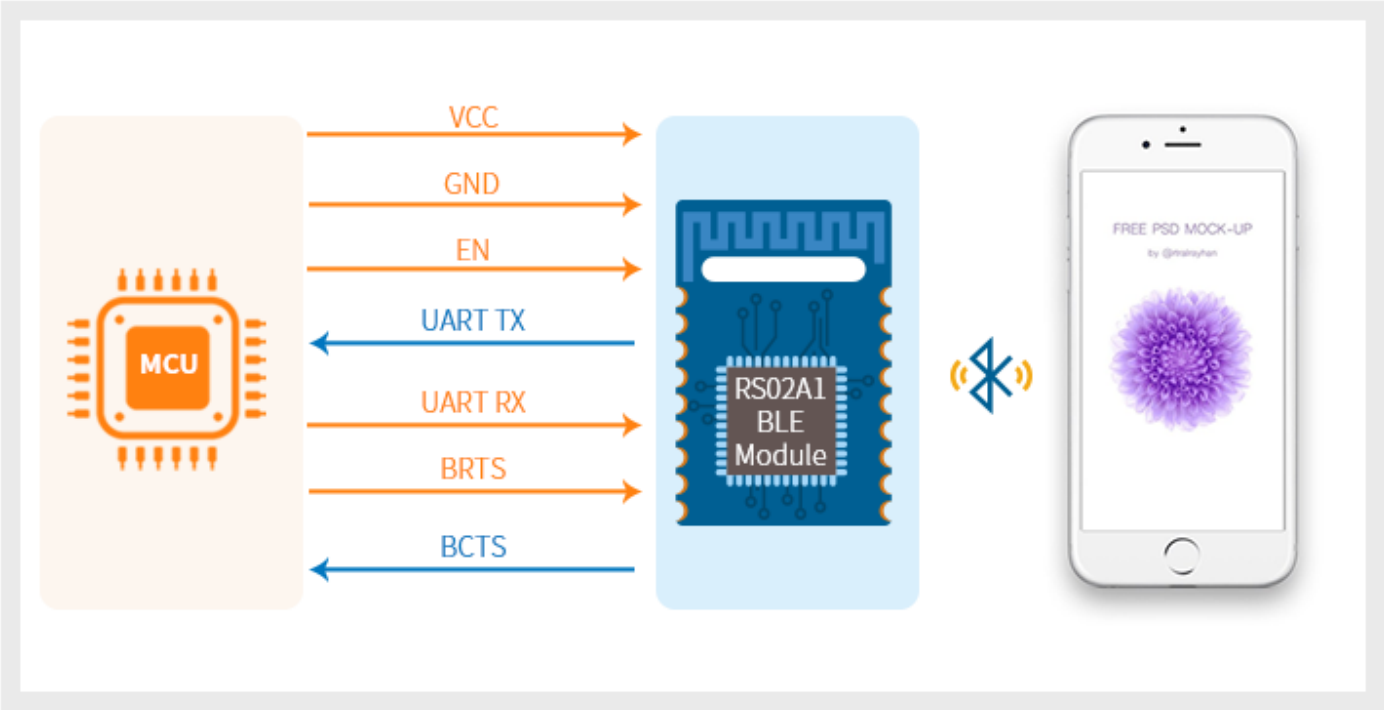


Figure 1. Schematic Diagram of Module in Bridge Mode

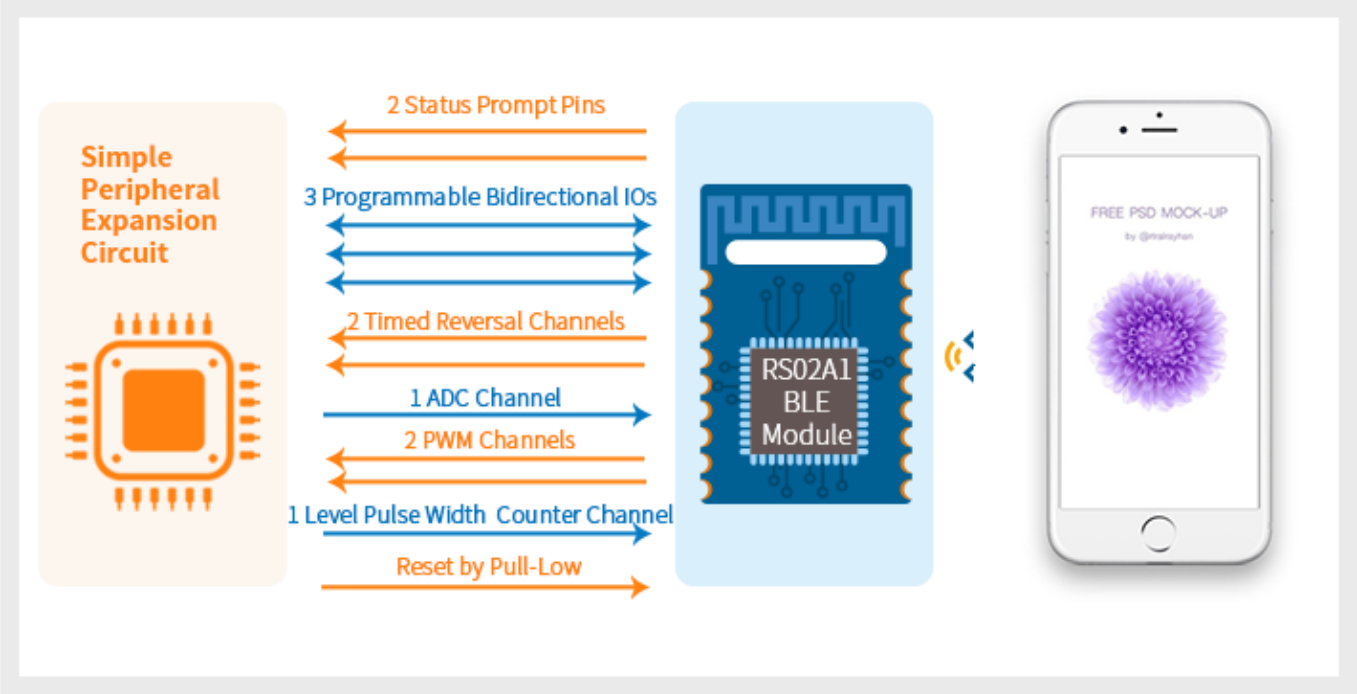


Figure 2. Schematic Diagram of Module in Direct-Driven Mode

Note:

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

5 Package and Pin Assignment

5.1 Pin Assignment

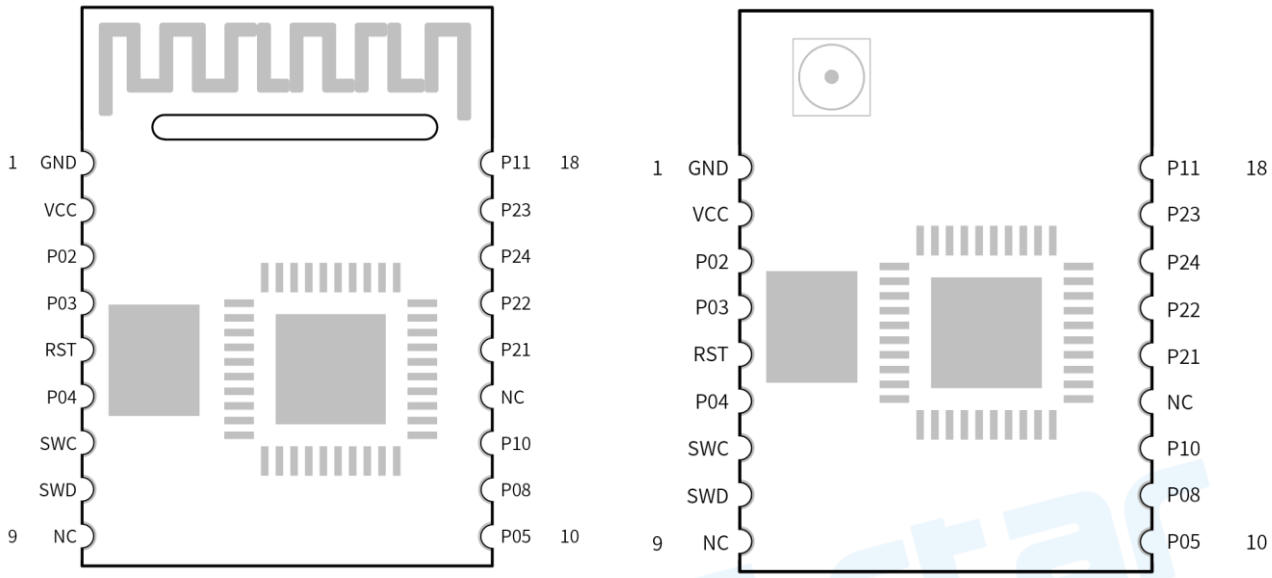


Figure 3. Pin Assignments of RSBRS02ABR/BRI

Table 2. Pin Functions of RSBRS02ABR/BRI

Pin	Name	Chip Pin	I/O	Description
1	GND	GND	–	Ground
2	VCC	VCC	–	Power supply 2.3 V ~ 3.6 V
3	IO2	P02	O	Output pin (able to timed reversal) / Sleep mode indicator
4	IO1	P03	O	Connection status indicator 0: Bluetooth connected 1: Bluetooth not connected Output pin (able to timed reversal)
5	RST	RST	I	Reset, active low, no internal pull-up
6	EN	P04	I	Module-enable control line (active low) 0: Module starts to broadcast, until it connects to the mobile device. 1: Enter sleep mode immediately, regardless of the current status.
7	SWC	SWC	–	JTAG clock pin
8	SWD	SWD	–	JTAG data pin
9	NC	-	-	
10	RESTORE	P05	I	Keep this pin at low level for 5 s, the system can be partially reset (light

	/ IO0			recovery). If keep more than 20 s, the system can be completely reset (deep recovery). (See details in "System reset and recovery")
11	PWM2	P08	I/O	After BLE is connected, use FFFB1 to control PWM
12	PWM1	P10	I/O	After BLE is connected, use FFFB1 to control PWM
13	NC	-	-	
14	BRTS	P21	I	As the request of data sending (for module wake-up) 0: Master has data to send, and module will wait for data transmission from the master, so the module will not sleep. 1. Master has no data to send, or data has been sent. So, the value of the signal line should be set at "1".
15	BCTS	P22	O	Data input signal (for master wake-up, optional) 0: Module has data to send, and the master will receive the data from the module. 1: Module has no data to send, or data has been sent, and the value of the signal line should be set at "1".
16	TX	P24	O	Serial port TX
17	RX	P23	I	Serial port RX
18	ADC	-	I	-

5.2 Recommended PCB Footprint

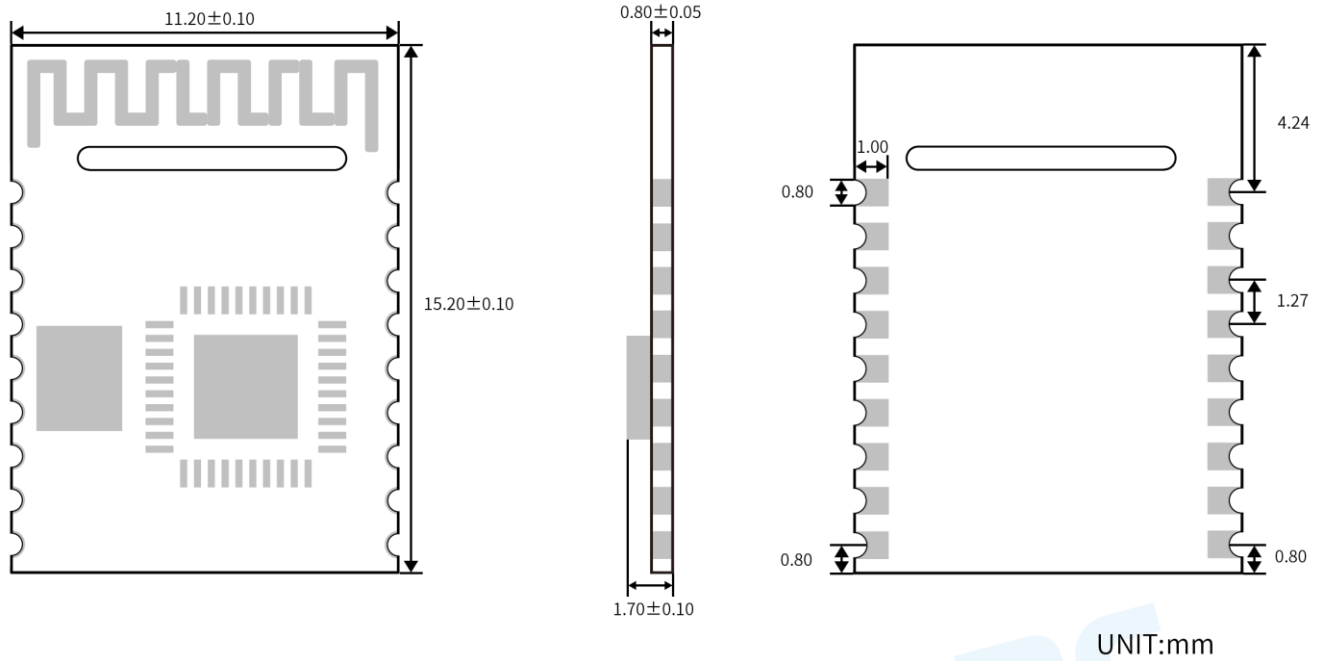


Figure 4. PCB Footprint of RSBRS02ABR

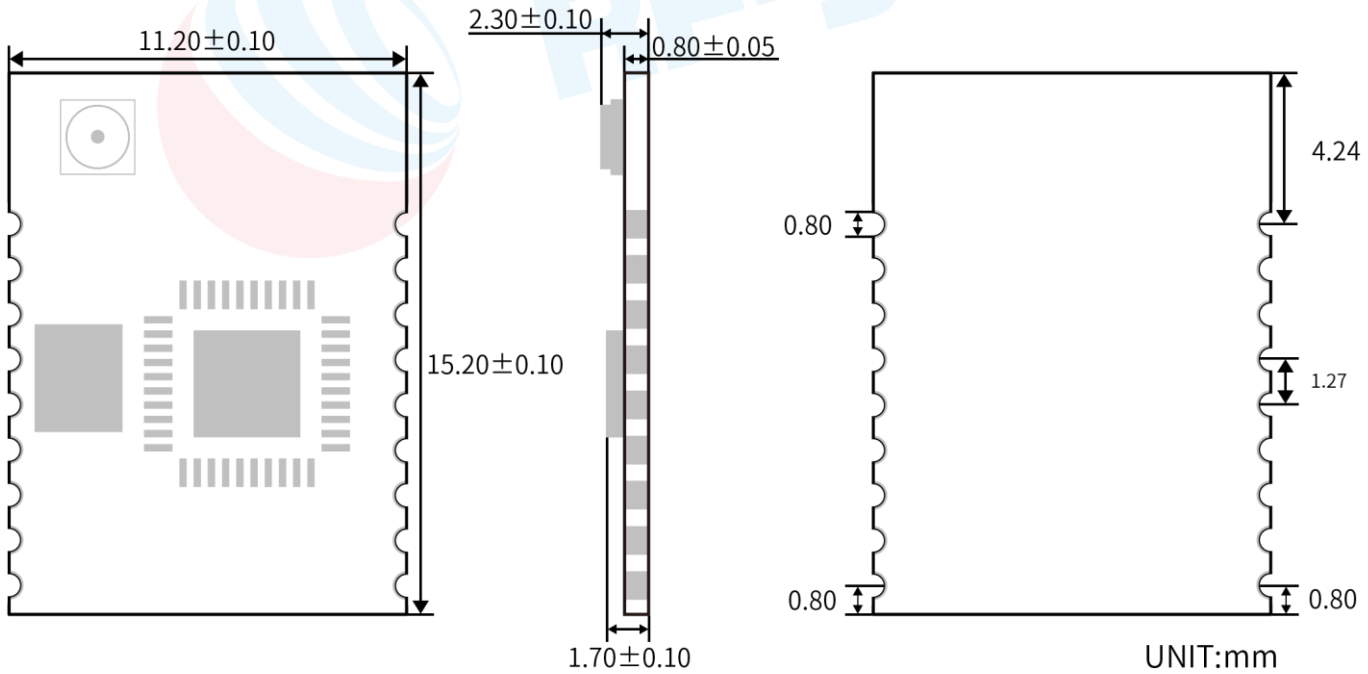


Figure 5. PCB Footprint of RSBRS02ABRI

6 UART Transparent Transmission Protocol

The bridge mode means to set up a bi-directional communication between user CPU and mobile devices by connecting the module with user CPU through serial port. Users can re-set serial port baud rate and BLE connection interval by the specified AT commands (see details in "AT Command"). The module will have different data TX & RX capability, as per different serial port baud rates and BLE connection intervals. Considering the use of low-speed CPU, the default baud rate is set at 115200 bps.

When the BLE connection interval is 20 ms and the serial port baud rate is at 115200 bps, the module has the highest transmit ability in theory (12.4 K/s). Take the configuration in the level-enabled mode as an example, UART transparent transmission protocol will be introduced in detail as below.

According to the MTU (maximum transmission unit) custom serial port packet of the connected device, the packet of the module will be sub-packed automatically and sent. The maximum load of each packet is the MTU minus 3 Bytes (for example: if the module is connected to an Android phone and the MTU is 251, the module sends data to the phone with a maximum of 248 Bytes). Data packets from mobile devices to the module must be sub-packed automatically [each packet is from 1 Byte to (MTU-3) Byte] before sending. The module will transmit them to the master RXD in turn, when received the packets.

1. Hardware protocol of serial port: 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 broadcast at the internal of 200 ms, until it pairs with mobile devices. When EN jumps from low to high, the module will enter sleep mode immediately, regardless of the current status.
3. After the module is connected, BRTS needs to be pulled low if the master (MCU) has data to send to the BLE module, and the data transmission can be started around 50 ms afterwards. BRTS should be pulled high by the master after transmission finished and make the module exit the serial RX mode. Pay attention to that time delay will happen after data is completely transmitted, and confirm that the data transmission has been completely finished before BRTS pulled-high. 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 μs afterwards. BCTS will be set high by the module when data transmission is finished.
5. If the master BRTS is being kept at a low level, the Bluetooth module will always be in RX mode and the power consumption will be high.
6. After the module is connected, a string of "TTM:CONNECTEDK\r\n\0" will be printed from TX. The string could be used to confirm whether the normal transmit operation is done. Of course, the connection status prompt pin can be used instead. Also, the connection can be checked by sending a specific confirmation string to the module from mobile devices. When APP automatically disconnects the module, there will be a string "TTM:DISCONNECT\r\n\0" from TX.
7. The default Bluetooth connection interval is 20 ms. If low-speed TX mode is needed for saving power, connection interval must be adjusted by AT command (the maximum connection interval to be 2000 ms). 248-Bytes is maximum

transmission length for each interval. Set the connection interval as T (unit: ms), and the highest transmit rate per second V (unit: Byte/s) is as follows:

V = 248*1000/T (V is only relevant with T)

If the Bluetooth connection interval of the module is 20 ms, and 248-Bytes is maximum transmission length for each interval, the theoretical maximum transmission capacity (transmit rate) will be 248 * 1000/20 = 12.4K Byte/s. Tests have shown that the packet loss is very little when transmit rate under 5 KB/s. For safety's sake, it is suggested to do check-sum and re-transmission processing in the upper layer, no matter for high or low speed transmit applications.

Note: MTU of Android is 251 Bytes, while MTU of iOS is 185 Bytes. The size of each packet sending is MTU-3 Bytes.

8. The size of the serial port data packets can be various and the length can be any value less than 1k Bytes, as long as the above conditions are met. But in order to utilize the communication payload in highest efficiency and to avoid communication running in full capacity, it is recommended to use data packets of 20 Bytes, 80 Bytes, or 248 Bytes in length, and the interval of packet is bigger than 20 ms.

Note:

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

7 AT Command

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 ("TTM:OK\r\n\0" or "TTM:ERP\r\n\0", etc.). **Data packets which do not start with "TTM" will be regarded as transparent transmission data. CRLF cannot be added in any command.**

- AT Command

Table 3. AT Command List

AT Command	Saved After Power-off	Parameter and Description	Possible Response	Remarks
TTM:CIT-Xms (Effective after successful connection)	Yes	Set BLE connection interval (in ms). X = **"20"**, "30", "50", "100", "200", "300", "400", "500", "1000", "1500", or "2000".	TTM:TIMEOUT\r\n\0 TTM:OK\r\n\0 TTM:ERP\r\n\0	Timeout configuration. Successful operation. Incorrect command format.
TTM:CIT-?	-	Acquire connection interval	TTM:CIT-Xms, "X" is the specific BLE connection interval	Connection interval is returned.
TTM:NAM-?	-	Acquire module name.	TTM:NAM-xxxxxxxxxxx, "xxxxxxxxxxx" is the module name.	Module name is returned.
TTM:REN-Name	Yes	Set module name. "Name" is the new module name with any string of no less than 16-bit length.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:BPS-?	-	Acquire baud rate.	TTM:BPS-X, "X" is the baud rate.	Baud rate is returned.
TTM:BPS-X	Yes	Set baud rate. X = "9600", "19200", "38400", "57600", "115200", "256000".	TTM:BPS SET AFTER 2S ...\r\n\0 TTM:ERP\r\n\0	Successful operation with new baud rate in 2 s. Incorrect command format.

TTM:MAC-?	-	Acquire MAC address.	TTM:MAC-xxxxxxxxxxx, "xxxxxxxxxxx" is module MAC address.	MAC address is returned.
TTM:MAC-X	Yes	Set MAC address. X is 1 12-bit MAC string, for example: 123456789ABC	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:ADP-(X)	Yes	Set broadcast cycle. T = X* 100 ms. X = "2", "5", "10", "15", "20", "25", "30", "40" or "50". For example: X = 5, it means the cycle is 500 ms.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:ADP-?	-	Acquire broadcast cycle	TTM:ADP-x, "x" is the specific broadcast cycle.	Broadcast cycle is returned.
TTM:ADD-Data	Yes	Set customized broadcast packet. Data is the customized broadcast packet. Data length L <= 16.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:ADD-?	-	Acquire customized broadcast packet	TTM:ADD-xxxxxxxxx, "xxxxxxxxx" is the customized broadcast packet in ASCII format	Customized broadcast packet is returned.
TTM:PID-Data	Yes	Set customized product ID. Data is the customized product ID. Data length = 2. **RS** is by default.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:PID-?	-	Acquire customized product ID.	TTM:PID-Data\r\n, Data is the specific product ID. **RS** is by default.	Customized product ID is returned.
TTM:TPL-(X)	Yes	Set transmit power (in dBm).	TTM:OK\r\n\0	Successful operation.

		X = "5", "**0**", "-5", or "-21".	TTM:ERP\r\n\0	Incorrect command format.
TTM:TPL-?	-	Acquire transmit power value	TTM:TPL-(x)\r\n, "X" is the specific transmit power value.	transmit power value is returned.
TTM:EUP-ON	-	Enable pull-up of EN pin.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:EUP-OFF	-	Disable pull-up of EN pin.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:EUP-?	-	Query EN pin status	TTM:EUP-ON\r\n TTM:EUP-OFF\r\n	EN pin status is returned.
TTM:RSI-ON	No	Enable timed output of RSSI signal strength every one second.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:RSI-OFF	-	Disable timed output of RSSI signal strength.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:UID-x1,x2,x3,x4,x5	-	x1, x2, x3, x4, x5 are UUID parameters. Such as service UUID, characteristic value UUID, Base UUID.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:RAT-X	Yes	X is the BLE communication rate, X value is 1M, 2M	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:RAT-?	-	Query communication rate	TTM:RAT-XM. XM is the specific communication rate	Communication rate is returned
TTM:RTC-?	-	Acquire RTC time.	TTM:RTC-xxxxxxxxxxxxx, "xxxxxxxxxxxxx" is the data.	RTC time is returned.
TTM:RTC-X	No	Set RTC time. X is the data: year, month, date, hour, minute and second.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:CDL-Xms	Yes	Set the delay time between low level output of BCTS and TX data	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.

		output (in ms). X = "0","2","5","**10**","15". The actual delay (T) will be T = (X + Y) ms, if the minimum delay is no less than X, while 500 μs < Y < 1 ms.		
TTM:VER-?	-	Acquire version number.	TTM:VER-XXXXXX, "X" means the version number.	Version number is returned.
TTM:PWD-?	-	Acquire connection password.	TTM:PWD-XXXXXX.	Connection password is returned.
TTM:PWD-xxxxxx	Yes	X is 6 digital number. For example: 123456.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:RST-SYSTEMRESET	-	System reset.	Module is working!	Reset module.
TTM:RST-RESET	-	Deep recovery.	Module is working!	Recovery all parameters.
TTM:RST-RSTPWD	-	Light recovery.	Module is working!	Recovery password.

- **Connection Interval Configuration**

Input the following string "TTM:CIT-Xms" to RX to set the BLE connection interval, wherein X = **"20",** "30", "50", "100", "200", "300", "400", "500", "1000", "1500", or "2000" (in ms) (all data format is in ASCII code).
For example: “TTM:CIT-30ms” means the BLE connection interval is set as 30 ms.

After the command is executed, the following confirmation will be got from TX:
"TTM:TIMEOUT\r\n\0": It means timeout and the failed modification.
"TTM:OK\r\n\0": It means the operation is successful and the new connection interval is applied.

The success of connection interval configuration depends on the limit of connection intervals by mobile devices. The maximum connection intervals are varied in different version of iOS. Tests with iPhone (iOS 8 system and its above versions) show the fastest interval is 20 ms and the slowest is 2 s. On the other hand, due to the internal mechanism of BLE protocol, execution efficiency of this command will be different with different connection intervals. In iOS 8 system and its above versions, changing from the current connection interval of 2000 ms (max. 2000 ms) to other connection intervals, it takes around 100 s at least. While the execution will be fast when executing this AT command in other high-frequency connection intervals (for example: 100 ms).

- **Acquire Connection Interval**

Input the following string "TTM:CIT-?" to RX to acquire the BLE connection interval.

After the command is executed, the following confirmation will be got from TX:
“TTM:CIT-Xms”, and “X” is the specific connection interval.

- **Acquire Module Name**

Input the following string "TTM:NAM-?" to RX to acquire the module name.
After the command is executed, the following confirmation will be got from TX:
“TTM:NAM-xxxxxxxxxxxx\r\n\0”, and “xxxxxxxxxxxx” is the module name.

- **Module Rename**

Input the following string "TTM:REN-" + Name to RX to rename the module (length of name should not exceed 16 Bytes in ASCII code format).
For example: “TTM:REN-ABC123” means the module is renamed as “ABC123”.

After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

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

- **Acquire Baud Rate**

Input the following string "TTM:BPS-?" to RX to acquire the baud rate.

After the command is executed, the following confirmation will be got from TX:
“TTM:BPS-X”, and “X” is the baud rate. Wherein X ="9600", "19200", "38400", "57600", **"115200"**, “256000” (all data format is in ASCII code).

- **Baud Rate Configuration**

Input the following string "TTM:BPS-X" to RX to set the baud rate. Wherein X ="9600", "19200", "57600", "115200", “256000” (all data format is in ASCII code).
For example: “TTM:BPS-115200” means the baud rate is 115200 bps.

After the command is executed, the following confirmation will be got from TX:
“TTM:BPS SET AFTER 2S…”: It means the modification is successful after 2 seconds.
"TTM:ERP\r\n\0": It means the incorrect command format.

- **Acquire MAC Address**

Input the following string "TTM:MAC-?" to RX.
After the command is executed, the following confirmation will be got from TX:
“TTM:MAC-xxxxxxxxxxxx\r\n\0", and "xxxxxxxxxxxx" is the Bluetooth module address in 6 Bytes.

- **MAC Address Configuration**

Input the following string "TTM:MAC-xxxxxxxxxxxx" to RX, wherein “xxxxxxxxxxxx” is the Bluetooth module address in 6 Bytes.

After the command is executed, the following confirmation will be got from TX:

“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

This command can be saved after power-off. When reset, the module will operate with the new MAC address.

- **Broadcast Cycle Configuration**

Input the following string "TTM:ADP-(X)" to RX to set the broadcast cycle of the module, T = X * 100 ms
Wherein X = **"2"**, "5", "10", "15", "20", "25", "30", "40" or "50" (all data format is in ASCII code).
For example: “TTM:ADP-(2)” means the broadcast cycle is 200 ms.

After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

This command can be saved after power-off. when reset, the module will operate with new broadcast cycle.

- **Acquire Broadcast Cycle**

Input the following string "TTM:ADP-?" to RX to acquire the broadcast cycle of the module.

After the command is executed, the following confirmation will be got from TX:
“TTM:ADP-(X) ": X is the specific broadcast cycle.

- **Add Customized Broadcast Packet**

Input the following string "TTM:ADD-Data" to RX to customize broadcast packet. Wherein “Data” is the additional data ready to be broadcast (0 < Length < = 16 Bytes) (all data format is in ASCII code).
After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

This AT command is effective after configuration, and realize some customized broadcast packets. The data can be saved after power-off. If 16-Byte data are set all set as 0, customized broadcast packets will not be used. Instead, the default broadcast packets are applied.

- **Acquire Customized Broadcast Packet**

Input the following string "TTM:ADD-?" to RX to acquire the customized broadcast packet.

After the command is executed, the following confirmation will be got from TX:
“TTM:ADD-Data": Data is the specific customized broadcast packet.

- **Product ID Definition**

Input the following string "TTM:PID-Data" to RX to define product ID, wherein “Data” is a 2-Byte product ID with the range from 0x0000 range to 0xFFFF (L = 2) (all data format is in ASCII code).
For example: “TTM:PID-RS” means the product ID is RS and RS is equal to 0x5253 in hexadecimal.

After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

This product ID can be saved after power-off. ID will show in the broadcast packet and can be used to filter devices or to determine if it is a specific product.

- **Acquire Product ID**

Input the following string "TTM:PID-?" to RX to acquire the custom product ID.

After the command is executed, the following confirmation will be got from TX:
“TTM:PID-Data": Data is the specific product ID.

- **Transmit Power Configuration**

Input the following string "TTM:TPL-(X)" to RX to set the corresponding transmit power (in dBm). Wherein X = "5", **"0"**, "-5", or "-21" (all data format is in ASCII code).
For example: “TTM:TPL-(0)” means the transmit power is set as 0 dBm.

After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

- **Acquire Transmit Power**

Input the following string "TTM:TPL-?" to RX to acquire the transmit power.

After the command is executed, the following confirmation will be got from TX:
“TTM:TPL-(X)\r\n\0": “X” means the transmit power value (in dBm).

- **Internal Enable of EN Pin**

Input the following string "TTM:EUP-ON" to RX to enable internal pull-up of EN pin which is the default configuration.
Input the following string "TTM:EUP-OFF" to RX to disable internal pull-up of EN pin. When broadcast during disabled internal pull-up, more than 80 μA current can be saved.

After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

- **Acquire Internal Enable Status of EN Pin**

Input the following string "TTM:EUP-?" to RX to acquire the enable status of EN pin.

After the command is executed, the following confirmation will be got from TX:
“TTM:EUP-ON\r\n\0" or “TTM:EUP-OFF\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

- **Modify Transparent Transmission UUID**

Input the following string "TTM:UID-x1,x2,x3,x4,x5" to RX to modify UUID. 16-bit is by default.

x1: Service UUID type. Value range: 0,1

0: 16-bit UUID

1: 128-bit UUID

x2: Services need to be modified. Value range: 0,1

0: Master sends data to slave

1: Slave sends data to master

x3: Service UUID

4 byte

ASCII format: 0000-FFFF

x4: Characteristic UUID

4 byte

ASCII format: 0000-FFFF

x5: 128 bit UUID, Base UUID

32 byte in ASCII format. If the 16-bit UUID format is selected, x5 is optional.

Note: All parameters must be separated by ","

After the command is executed, the following confirmation will be got from TX:

“TTM:OK\r\n\0": It means the successful operation.

"TTM:ERP\r\n\0": It means the incorrect command format.

The command will take effect after restarting and will be saved after power off.

For example:

The master sends data to the slave.

The default service UUID is: 6000-0001-0000-0000-0000-0000-0000-0001

The default characteristic UUID is: 6000-0002-0000-0000-0000-0000-0000-0001

If service UUID 0001 is needed to change to FFF0, and characteristic UUID 0002 is needed to change to FFF1.

Then the input AT command is: TTM:UID-0,0,FFF0,FFF1,

If the service UUID need to be modified to: 6000-FFF0-0000-0000-0000-0000-0000-FFFF

Characteristic UUID need to be modified to: 6000-FFF1-0000-0000-0000-0000-0000-FFFF

The input AT command is: TTM:UID-1,0,FFF0,FFF1,60000000000000000000000000000FFFF

- **Modify BLE Rate**

Input the following string "TTM:RAT-X" to RX to modify the communication rate of BLE module.

X is the communication rate that needs to be set. Value range: 1M, 2M

1M: To set the communication rate of 1M after connection.

2M: To set the communication rate of 2M after connection.

After the command is executed, the following confirmation will be got from TX:

“TTM:OK\r\n\0": It means the successful operation.

"TTM:ERP\r\n\0": It means the incorrect command format.

The command will take effect after restarting and will be saved after power off.

- **Acquire BLE Rate**

Input the following string "TTM:RAT-?" to RX to acquire the communication rate of BLE module.

"TTM:RAT-XM": XM is the specific communication rate.

"TTM:ERP\r\n\0": It means the incorrect command format.

- **Output RSSI Signal Strength**

Input the following string "TTM:RSI-ON" to RX to enable timed output of RSSI signal strength and the interval is 1 s.

Input the following string "TTM:RSI-OFF" to RX to disable timed output of RSSI signal strength.

After the command is executed, the following confirmation will be got from TX:

"TTM:OK\r\n\0": It means the successful operation.

"TTM:ERP\r\n\0": It means the incorrect command format.

If timed output RSSI signal strength is enabled, the string of RSSI signal strength will be output "TTM:RSI-xx\r\n\0" once every second.

For example: "TTM:RSI-63\r\n\0" means the RSSI signal strength is -63 dBm.

Note: This configuration cannot be saved after power-off. RSSI output will be disabled after re-power-on.

- **RTC Configuration and Acquisition**

Input the following string "TTM:RTC-xxxxxxxxxxxxxx" to RX to set RTC time. Year is with 4-bit, month, date, hour, minute and second are with 2-bit.

For example: "TTM:RTC-20170102030405" means the RTC time is 3:04:05, January 02, 2017.

After the command is executed, the following confirmation will be got from TX:

"TTM:OK\r\n\0": It means the successful operation.

"TTM:ERP\r\n\0": It means the incorrect command format.

Input the following string "TTM:RTC-?\r\n\0" to RX to acquire RTC time.

"TTM:RTC-xxxxxxxxxxxxxx\r\n\0": It means the successful operation.

"TTM:ERP\r\n\0": It means the incorrect command format.

Note: This configuration cannot be saved after power-off. RTC need to be re-configured after re-powered-on.

- **Data Delay Configuration**

Input the following string "TTM:CDL-Xms" to RX to set the delay time between low level output of BCTS and TX data output (in ms). Wherein X = "0", "2", "5", **"10"**, "15", "20", or "25" (all data format is in ASCII code).
For example: “TTM:CDL-2ms” means the delay time is 2 ms.
After the command is executed, the following confirmation will be got from TX:
“TTM:OK\r\n\0": It means the successful operation.
"TTM:ERP\r\n\0": It means the incorrect command format.

To make the user CPU have enough time to wake-up from sleep mode and ready to receive data, the module is provided this delay (X) configuration. The BRTS will be set low before there is data to be sent through the serial port, while the delay time between low level output of BCTS and TX data output will be set by this parameter. The actual delay (T) will be T = (X + Y) ms, if the minimum delay is no less than X, while 500 μs < Y < 1 ms.
This configuration can be saved after power-off.

- **Acquire Data Delay**

Input the following string "TTM:CDL-?" to RX to acquire BLE data delay value.

After the command is executed, the following confirmation will be got from TX:
“TTM:CDL-Xms": X is the data delay value.
"TTM:ERP\r\n\0": It means the incorrect command format.
The scheme of data delay configuration is as follows:

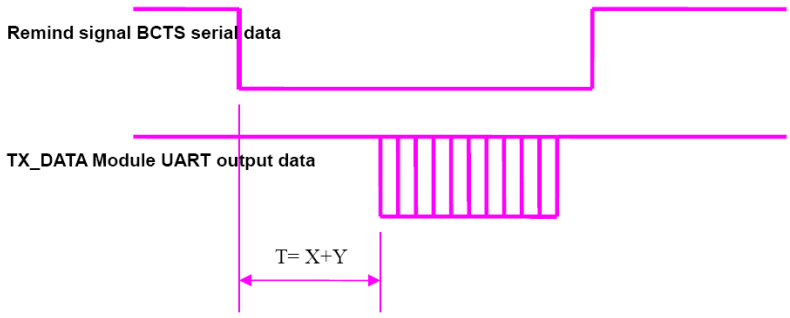


Figure 6. Scheme of Data Delay Configuration

- **Version Number Acquisition**

Input the following string "TTM:VER-?" to RX to acquire module version number.
“TTM:VER-XXXXXX": It means the successful operation and XXXXXX is the module version number.
"TTM:ERP\r\n\0": It means the incorrect command format.

- **Connection Password Acquisition**

Input the following string "TTM:PWD-?" to RX to acquire connection password.

“TTM:PWD-XXXXXX": It means the successful operation and XXXXXX is the password in 6-bit.

"TTM:ERP\r\n\0": It means the incorrect command format.

- **Connection Password Configuration**

Input the following string "TTM:PWD-?" to RX to acquire connection password.

“TTM:PWD-XXXXXX": It means the successful operation and XXXXXX is the password in 6-bit.

"TTM:ERP\r\n\0": It means the incorrect command format.

This configuration can be saved after power-off.

- **Module Reset**

Input the following string "TTM:RST-SYSTEMRESET" to RX to force the module system reset once.

- **Module Reset - Deep Recovery**

Input the following string "TTM:RST-RESET" to RX to force the module deep reset once and recovery all the modified parameters, which means the module factory reset.

- **Module Reset - Light Recovery (Recovery Password)**

Input the following string "TTM:RST-RSTPWD" to RX to force the module light reset once and recovery the password parameters.

8 BLE Protocol (APP Interface)

- Bluetooth Data Channel [Service UUID:0xFFE5]

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFE9	Write	MTU-3	None	Written data will output from TX.

Remark: Bluetooth input data will be transmitted to serial output. APP operates write in this channel by BLE API, and the data will be output from TX. See details in "UART Transparent Transmission Protocol".

- Serial Port Data Channel [Service UUID:0xFFE0]

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFE4	Notify	MTU-3	None	Notification will be generated from the input data of RX in this channel and sent to smart devices.

Remark: Serial input data will be transmitted to BLE output. If notification switch of FFE4 channel is enabled, a notify event will be generated in this channel when the master CPU transmits legal data to the module RX through serial port, and APP can directly process and use notify information in the callback function. See details in "UART Transparent Transmission Protocol".

Note:

MTU: Maximum Transmission Unit.

- **ADC Input (1 Channel) [Service UUID:0xFFD0]**

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFD1 (handle:0x0068)	Read / Write	1	0x00	Enable control 0x00: Disable ADC channels 0x01: Enable ADC channels
FFD2 (handle:0x006B)	Read / Write	2	0x01F4	Sampling cycle (in ms) For example: 0x01F4 is corresponding to 500 ms.
FFD3 (handle:0x006D)	Read / Notify	2	0x0000	ADC0 sampling result: the maximum is 0x01FFF

Remark: 1-channle ADC input control. APP operates write in this channel FFD1 by BLE API to open two 13-bit ADC channels. Sampling cycle t (in ms, t ≥100 ms) of two ADC channels is controlled by write in FFD2 channel. If notify of FFD3 is enabled, a notify event with sampling results (0 ~ 0x1FFF, low Byte in front) will come in this channel after the sampling results are generated. Those sampling results can be processed and used by the APP in the callback function of notify. ADC reference power supply is the chip internal reference power supply of 1.25 V. So, the flotation of the voltage of power supply will not lead to any new measurement errors, but the sampling voltage measured must be controlled between 0 V ~ +1.25 V.

- **PWM Output (2 Channels) [Service UUID:0xFFB0]**

Characteristic UUID	Operation	Bytes	Default Value	Example	Remarks
FFB1 (handle:0x0070)	Read / Write	1	0x01	0x00	Initialize 2 PWM channel with all-low pulse width
				0x01	Initialize 2 PWM channel with all-high pulse width
				0x02	Initialize corresponding PWM channel with current pulse width
FFB2 (handle:0x0073)	Read / Write	2	0xFFFF	0xFF00	Output all-high pulse width in PWM1 channel (P08)
				0x00FF	Output all-high pulse width in PWM2 channel (P10)
				0x2020	Output 32/256 pulse width in PWM1 ~ PWM2 channel
FFB3 (handle:0x0076)	Read / Write	2	0x8235	500 ≤ w ≤ 65535	Signal frequency configuration of PWM output. Two channels are the same. 0x8235 (120 Hz) is by default.
FFB4 (handle:0x0079)	Read / Write	2	0x0000	0 ≤ t ≤ 65535	Time width of 2 PWM change. Two channels are the same. 0x0000 is by default (sudden change).

Remark:

FFB1 is the configuration channel of 2-channel PWM initialization mode. Set 2-channel PWM initialization mode (1 Byte) by write to **FFB1** channel. The factory setting is 0x01 by default with all-high pulse width output. The setting can be saved after power-off.

0x00: Output 0% pulse width (all-low pulse width). Sleep mode is allowed in this mode.

0x01: Output 100% pulse width (all-high pulse width). Sleep mode is allowed in this mode.

0x02: Output current PWM. This value will be saved immediately after being configured and used as the initial value of 2-channel PWM in next power-on. Sleep mode is not allowed in this mode.

FFB2 is the configuration channel of 2-channel PWM duty-ratio. Adjust 2-channel PWM duty-ratio by write to **FFB2** (4 Bytes). Each Byte is corresponded to one channel.

0xFF: Output 100% pulse width (all-high pulse width).

0x00: Output 0% pulse width (all-low pulse width).

If X is set, the duty ratio will be about X / 0xFF. And the latest configuration value can be got by read this channel.

After power on, the default value will be 0xFFFFFFFF (all-high pulse width output).

When this function is enabled, the module will not enter sleep mode, until it is set to 0xFFFFFFFF (all-high). It means

PWM is disabled.

This channel is used for setting PWM duty-ratio with the range from 0x00 to 0xFF and the default signal frequency of 120 Hz. (See details in “FFB3 frequency control channel”).

For example: 0xFF00

1. Total of 2 PWM channels.
2. 0xFF00, 4 Bytes are corresponded to 2 channels.
3. 0xFF means output all-high pulse width (100%), and 0x00 manes output all-low pulse width (0%).
4. The pulse width frequency is 120 Hz by default.

FFB3 is the configuration channel of 2-channel PWM frequency control. Adjust 2-channel PWM frequency of square wave by write to FFB3 (2 Bytes). The width of the signal cycle W must meet: the condition $500 \leq w \leq 65535$ (a unit is equivalent to 0.00000025 s), and the corresponding square wave cycle: $0.000125\ s \leq T \leq 0.01638375\ s$. Therefore, the adjustable range of signal frequency of square wave is: $61.036\ Hz \leq f \leq 8\ kHz$, and 2 PWM frequencies of square wave are the same.

The latest configuration value can be got by read this channel. This configuration can be saved after power-off.

Factory setting of W is 0x8235 and the corresponding pulse-width frequency of is 120 Hz by default.

Example 1:

Output 120 Hz frequency of square wave: write 0x8235 (33333) to FFB3, and the corresponding square wave cycle will be 0x8235 * 0.00000025 s = 0.00833325 s, then, get the frequency of 120 Hz.

Example 2:

Output 1 kHz frequency of square wave: write 0x0FA0 (4000) to FFB3 and the corresponding square wave cycle will be 0x0FA0 * 0.00000025 s = 0.001 s, then, get the frequency of 1 kHz.

FFB4 is the configuration channel of control of 2-channle PWM time width change. Adjust 2-channle PWM change speed of square wave frequency by write to FFB4 (2 Bytes). This value t must meet: $0 \leq t \leq 65535$, (a unit is equivalent to 100 ms).

The longer t is, the slower the PWM change from the current value to the target is. While the smaller t is, the faster the change is. When t is zero, the target value will be reached immediately. The time width of change of 4-channle PWM shares the same value.

The latest configuration value can be got by read this channel. This configuration can be saved after power-off.

Factory setting of t is 0x0000 and the corresponding change mode is sudden change by default.

- **3 Programmable I/Os [Service UUID:0xFFF0]**

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFF1 (handle:0x004B)	Read / Write	1	0b00000000	Configuration bit of IO2 ~ IO0 When the corresponding bit is set to 0: bit7 and bit6 indicate that IO2 and IO1 are signal prompt pin and active low. bit5 ~ bit0 indicate that IO0 work as input ports. When the corresponding bit is set to 1: bit7 and bit6 indicate that IO2 and IO1 work as general output port. bit5 ~ bit0 indicate that IO0 work as output ports.
FFF2 (handle:0x004E)	Write	1	--	IO2 ~ IO0 output status. It indicates the output level in IO2 ~ IO0. bit 7 and bit6 are only valid when IO2 and IO1 work as general output ports. When IO2 and IO1 work as signal prompt pin, bit7 and bit6 are invalid.
FFF3 (handle:0x0051)	Read / Notify	1	0xFF	IO0 input status. Notifications can be read and received. When notify is enabled, the change of certain input level will be notified to APP. IO2 and IO1 can only work as output or signal prompt pin, and the corresponding pins are invalid.

Remark: Channel of I/O configuration and control.

FFF1 is the configuration channel for 3 I/Os. 8-bit control the configuration of IO2 ~ IO0 (3 I/Os) correspondingly. When the two high bits of BIT7 and BIT6 are 0, IO2 and IO1 work as signal prompt pin. IO2 prompts **sleep status**, wherein 0 stands for woken status and 1 for sleep status. IO6 prompts **connection status**, wherein 0 stands for connection status and 1 for disconnection status. When the two high bits of BIT7 and BIT6 are 1, IO2 and IO1 work as general output. The two pins cannot work as input ports.

When bit5 is set to 1, IO0 work as output ports. When bit5 is are set to 0, IO0 are used as input ports.

FFF2 is the configuration channel of 3 I/Os output. 8-bit control the configuration of IO2 ~ IO0 (3 I/Os) correspondingly. Only the corresponding bit is set as output, the pin is effective. When certain I/O is set as output, the corresponding bit in this channel can be written, so that the output control of the I/O is realized. The bit corresponding to the I/O that is set

as input will be invalid.

Notes:

Configuration (FFF1) and output status control (FFF2) of IO are not saved after power-off by default.

But the configuration and output status of IO2, IO1 (not including IO0) can be saved by writing **0x01** in the remote control expansion channel **FF99**. The module will use the last-time saved settings to initialize the 3 I/Os (IO2 ~ IO0). That means the configuration and output status of IO0 cannot be saved after power-off. IO0 is always in input status after power-on by default. This pin is used to detect the function of factory reset (see details "Module Parameter Configuration").

FFF3 is the configuration channel of IO0 input. The 1-bit is corresponded to the input status of IO0. Only the corresponding bit is set as input, the pin is effective. If notification switch of FFE3 channel is enabled, (01 00 is needed to be written in 0x001D+1 = 0x001E by BTool), a notify event with a Byte (status of 1 I/O) will come in this channel after the level status of the pin is changed, which is only effective to I/Os set as input. This status information can be processed and used by the APP in the callback function of notify. IO2 and IO1 can only work as output or signal prompt pin, so their corresponding bits are invalid.

- **Timed Reversal Output (2 Channels) [Service UUID:0xFFF0]**

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFF4 (handle:0x0053)	Read / Write	4	0x00000000	Configuration of the first time reversal delay of IO1 0: Enable the reversal of IO1 Non-0: In ms, delay before reversal
FFF5 (handle:0x0055)	Read / Write	4	0x00000000	Configuration of the second time reversal delay of IO1 0: Disable the second time reversal Non-0: In ms, delay before reversal
FFF6 (handle:0x0058)	Read / Write	4	0x00000000	Configuration of the first time reversal delay of IO2 0: Enable the reversal of IO2 Non-0: In ms, delay before reversal
FFF7 (handle:0x005B)	Read / Write	4	0x00000000	Configuration of the second time reversal delay of IO2 0: Disable the second time reversal Non-0: In ms, delay before reversal

Remark: This configuration channel is for timed reversal.

The module can be configured in timed reversal output mode, when IO1 and IO2 are set as general output. The next reversal time of IO1 and IO2 can be set by write in this channel. By setting the current status of IO output, 1 can jump to 0, or vice versa. Reversal will not be enabled if set to 0.

This function is effective only when the two high bits of BIT7 and BIT6 are set to 1 (as output) in FFF1.

Set the first time delay of reversal of IO1 in FFF4, and set the second time delay of reversal of IO1 in FFF5. If FFF4 is set to 0, reversal of IO1 will not be enabled. If FFF4 is set to a non-zero value, while FFF5 is set to 0, IO1 reversal is only enabled for once. FFF5 must be configured first and reversal is not enabled at this time. Then FFF4 can be set to non-zero values to enable the timed reversal of IO1. Similarly, the timed reversal of IO1 can be disabled by write 0 to FFF4, and any value written before in FFF5 will be cleared. The timed range is from 0 to 0xFFFFFFFF ms (4294967295 ms, or around 1193 hours, or around 49.7 days). Time conversion to hexadecimal is as follows:

Table 4. Time Conversion to Hexadecimal

0.5 s	1 s	1.5 s	2 s	3 s	4 s	5 s
500 ms	1000 ms	1500 ms	2000 ms	3000 ms	40000 ms	5000 ms
0x01F4	0x03E8	0x05DC	0x07D0	0x0BB8	0x0FA0	0x1388

Take IO1 as an example, setting a periodic repeated reversal is as follows:

1. Set IO1 as general output by write 0bx1xxxxxx to FFF1.

2. Set IO1 at high level (1) by write 0bx1xxxxxx to FFF2.
3. Write 0x05DC (1.5 s) in FFF5 to set the second reversal delay, and 0 means reversal for only once.
4. Write 0x01F4 (0.5 s) in FFF4 to set the first reversal delay, and the reversal will be enabled immediately.

The steps of point 3 and 4 cannot be reversed. FFF5 must be configured before write non-zero value in FFF4 to enable reversal. Write 0 to FFF5 means reversal for only once. A square wave with the cycle of 1.5 + 0.5 = 2 s will be generated after the operations mentioned above. During the period, high level (1) will last for 0.5 s and low level (0) will last for 1.5 s. Reversal can be disabled immediately by write 0 in FFF4 and IO6 will keep the current level.

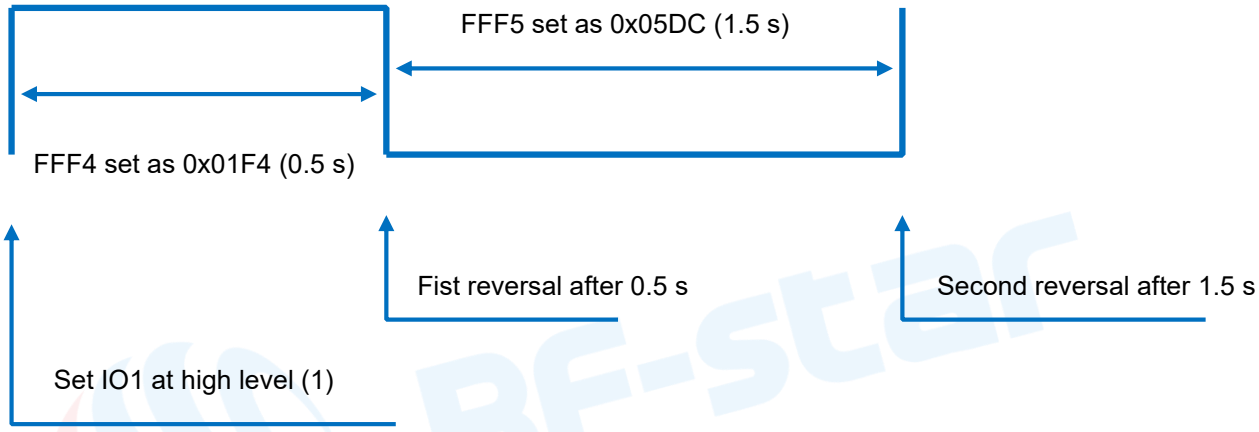


Figure 7. Diagram of Reversal Cycle (2 s)

FFF6 and FFF7 are the channels to configure the timed reversal delay for IO2 in the same way as for IO1.

Notes: If IO1 and IO2 are in the timed reversal cycle, write to the I/O output or re-configuration to signal prompt pin are invalid. Current timed reversal has to be disabled before the above operations are performed.

- **Level Pulse Width Counter (1 Channel) [Service UUID:0xFFF0]**

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFF8 (handle:0x005E)	Read / Notify	4	0x00000000	IO0 level duration at last time (in ms).

Remark: This channel is for notification of counting I/O level duration.

Level pulse width counting mode can be enabled when IO0 is set as general input.

This function is effective only when the the high bit of BIT5 is set to zero (as an input port) in FFF1.

FFF8 is the notification channel for IO0 level pulse width duration. If notification switch of FFE8 channel is enabled by APP through BLE API, a notify event with level duration [maximum value: 0xFFFFFFFF, in ms, range from 0 ~ 0xFFFFFFFF ms (4294967295 ms, or around 1193 hours, or around 49.7 days)] will generated in this channel after every reversal of IO0, which is only effective to I/O0 set as input. This information can be processed and used by the APP in the callback function of notify.

Notes:

The level counted is not the current level but the last-time one). Current level can be got by read in FFF3. Due to the limitation of BLE protocol, delay of submission of sampling results will not be longer than the connection interval.

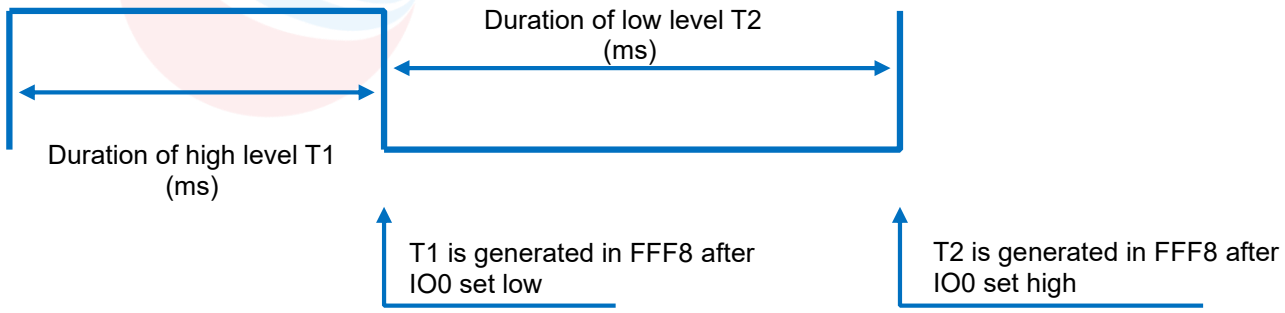


Figure 8. Diagram of Level Pulse Width Counter (IO0 as an Example)

- **Anti-Hijacking Password [Service UUID:0xFFC0]**

The module supports anti-hijacking password. Unauthorized mobile devices (or mobile phones) is prevent from being connected to the module effectively by this service. The initial password is 000000 (ASCII). In this case, APP does not need pairing with the module during connecting, so it is regarded as no use of password and any mobile device with specified APP can connect to the module.

The new password (not all zero) is set and saved by APP. If a new password (not all zero) is set, anti-hijacking is enabled. A password once configured requests will be submitted within **20 s** after APP connects to the module. Otherwise the connection is broken up. Any write operation except for password submission cannot be executed before APP submits the correct password.

If the password needs to be recovered, **the module must be reset first by pull-low RESTORE (IO0) pin for 5 s and the operation must be done within 30 s after connection set-up.** For safety, password read is not supported, and all passwords are kept by APP.

A password channel is provided to realize the submission, modification and cancellation of the password by protocol. Meanwhile, event notify service of password is also provided to inform APP of the results of password operations, including 4 events: right password, error password, successful password update and cancel password.

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFC1 (handle:0x0027)	Write (Saved after power-off)	12	“123456123456”(ASCII)	Submit current password of 123456, and the new password must be same as the previous one.
			“123456888888”(ASCII)	Change the previous password of 123456 into the new one of 888888, and the previous password must be correct.
			“888888000000”(ASCII)	Cancel password (by changing the password into the default value 000000, and the previous password must be correct.
FFC2 (handle:0x0028)	Notify	1	0(PWD_RIGHT_EVENT)	Right password.
			1(PWD_ERROR_EVENT)	Error password
			2(PWD_UPDATED_EVENT)	Successful password update.
			3(PWD_CANCEL_EVENT)	Cancel password.

Remark:

1. Password is all in 12-Byte ASCII, wherein the red part is the current password and the blue part is the new password.
2. Current password is "000000" by default before modified by APP.
3. The execution result of related password operations can be generated in this channel by enabling notification of FFC2.
4. When APP submits "123456123456", it means the new password is the same with the current one. And APP will

be notified in channel FFC2 of “notify:0(PWD_ RIGHT_EVENT)”. It shows the password submission is correct.

5. When APP submits the password (red part) is different from the current one, such as: "123455xxxxxx", regardless of the value of “xxxxxx” part, APP will be notified in channel FFC2 of “notify: 1(PWD_ ERROR _EVENT)”. It shows the password submission is wrong.
6. When APP submits "123456888888", it means the new password is "888888" and the current password is "123456". APP will be notified in channel FFC2 of “notify: 2(PWD_ UPDATED _EVENT”. It shows the password update is successful.
7. When APP submits "888888000000", it means the new password will be changed to an all-zero value. APP will be notified in channel FFC2 of “notify: 3(PWD_ CANCEL _EVENT)”. It shows the password is cancelled.

- **Device Information [Service UUID:0x180A]**

Characteristic UUID	Operation	Bytes	Default Value	Remarks
2A23	Read	8	xxxxxx0000xxxxxx (Hex)	System ID, where xxxxxxxxxxxx is the MAC address of module, with low Byte in front.
2A26	Read	17	V5.11u_xxxxxx_EP (ASCII)	Firmware version number of the module. Xxxxxx is the released date of firmware.
2A27	Read	9	RSBRS02ABR (ASCII)	Hardware version.
2A29	Read	6	RFstar (ASCII)	Manufacturer.

Remark: This channel is for module information read.

Acquire module information by read this channel 2A23. For example: xxxxxx0000xxxxxx, wherein xx part is the MAC address of the module in six Bytes (low Byte in the front).

Acquire module version number by read this channel 2A26. For example: Tv5.10u_xxxxxx_EP, wherein xxxxxx is the released date of firmware, 190628 represents for June 28th, 2019.

- **Module Parameter Configuration [Service UUID:0xFF90]**

Characteristic UUID	Operation	Saved or Not	Bytes	Default Value	Remarks
FF91 (handle:0x002C)	Read / Write	Yes	16	Tv511u-x xxxxxxx (ASCII string with terminator)	Device name, XXXXXXXX for the last four Bytes of the MAC address.
FF92 (handle:0x002F)	Read / Write	No	1	0	Set Bluetooth connection interval: **0: 20 ms** 1: 30 ms 2: 50 ms 3: 100ms 4: 200 ms 5: 300 ms 6: 400 ms 7: 500 ms 8: 1000 ms 9: 1500 ms 10: 2000 ms
FF93 (handle:0x0032)	Read / Write	Yes	1	5	Set baud rate of serial ports: 1: 9600 bps 2: 19200 bps 3: 38400 bps 4: 57600 bps **5: 115200 bps** 6: 256000 bps
FF94 (handle:0x0035)	Write	-	1	None	Channel to control remote reset and recovery: - Remote reset control by writing **0x55**. - Remote light recovery control and reset by writing **0x35** (recover user data only). - Remote deep recovery control by writing **0x36** (factory reset) and reset
FF95 (handle:0x0038)	Read / Write	Yes	1	0	Set broadcast cycle: **0: 200 ms** 1: 500 ms 2: 1000 ms 3: 1500 ms

					4: 2000 ms 5: 2500 ms 6: 3000 ms 7: 4000 ms 8: 5000 ms
FF96 (handle:0x003B)	Read / Write	Yes	2	0x5253	Set product ID
FF97 (handle:0x003E)	Read / Write	No	1	1	Set transmit power: 0: 5 dBm **1: 0 dBm** 2: -5 dBm 3: -21 dBm
FF98 (handle:0x0041)	Read / Write	Yes	16	Default broadcast packet.	Set customized broadcast data Customizing broadcast data: 0 < n ≤ 16. See details in "Broadcast Data Configuration".

Remark: The channel is for module parameter configuration.

FF91 is the channel for setting device name. Device name can be acquired and set by read and write to this channel accordingly. The length of device name set must meet the condition: 0 < L < 17. And **the name is suggested to end with the terminator ('\0')**. The default name is "Tv5vvv- XXXXXXX\0" (16 Byte), wherein vvv is the current firmware version number and XXXXXXXX is the last four Bytes of MAC address.

FF92 is the channel to set the connection interval. Connection interval between mobile devices and the module can be set by write 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 (30 ms) after power on. Test shows that it takes around 30 s to wait when the connection interval is changed from 500 ms to another interval by iPhone (iOS8 and its above version). But it will be effectively very quickly if the connection interval is changed from a high frequency one (such as: 30 ms) which is affected by BLE protocols.

FF93 is the channel to set baud rate. Baud rate can be set by read and write to the channel. The new baud rate will be effective in two seconds after set and can be saved after power-off. 5 (115200 bps) is by default.

FF94 is the channel to control remote reset and recovery. Various controlling functions can be realized by writing different values to the channel.

1. Write **0x55** to this channel will software-reset the module.
2. Write **0x35** to the channel will light-recovery the module. All user settings will be recovered to the factory defaults, including I/O output status, PWM initialization mode and user password. Afterwards, the module will be reset.
3. Write **0x36** to the channel will deep-recovery 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 broadcast cycle. Broadcast cycle can be set by read and write to this channel. The setting can be saved after power-off. 0 (200 ms) is by default.

FF96 is the channel to set product ID by read and write to the channel. APP can filter and connect to the specific product through this code. The setting can be saved after power-off. 0x5253 is by default.

FF97 is the channel to set transmit power by write to this channel. **1 (0 dBm) is by default.**

FF98 is the channel to set broadcast packets. Broadcast data can be customized by write 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 details in "Broadcast Data Configuration").

9 Broadcast Data Configuration

- Default Broadcast Data

When the module EN pin is 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 packets are included (default of 16 Bytes):

{

0x52,0x53,	Customized device type code, RS is by default, low-bit in the front, and can be set by AT command and APP.
0x19,0x16,	Firmware generated date. 0x20,0x15, is by default, it means the firmware is generated on the 15th week in 2020.
0x00, 0x00, 0x00, 0x00, 0x00, 0x00,	MAC address
0x05,	BPS parameter. 5 is by default. 5 means 115200 bps.
0x05,	CTS parameter. 10 is by default. It means that data will be transmitted after CTS is set low for 10 ms.
0x00,	Broadcast interval parameter. 200 ms is by default.
0x01,	Transmit power parameter. 0 dBm is by default.
0x00,	Connection interval parameter. 20 ms is by default.
0x00,	Anti-hijacking password timeout enable function. 0 (disable) is by default.

}

The data in the broadcast is the initial set value after the first compilation, and will not be changed after setting new parameters through AT command or APP.

- Customized Broadcast Data

Customizing the broadcast packet can be realized by AT command, and the maximum length is 16 Bytes (in blue). In the broadcast data GAP_ADTYPE_MANUFACTURER_SPECIFIC domain will contain the following packet, and the length is 2 + n Bytes:

{

0x00,0x00,	Customized device type code. 00 00 is by default, can be set by AT command;
Data [n],	Customized broadcast data, n < = 16;

}

Note:

Customized broadcast data can be modified by AT command and saved after power-off. After re-power on, last-time customized broadcast data will be shown. If customized broadcast data is all 0 (16 Byte), the customized broadcast will not be used but the system default broadcast packets. To avoid the extra power consumption caused by too long broadcast data, customized broadcast data can be set to be any value in 1 Byte.

10 System Reset and Recovery

There are three methods of module reset, among which the third one can recovery system parameters:

1. Reset module by AT command (See details in "AT Command").
2. Remote reset module by the service channel interface of APP [See details in "BLE Protocol (APP Interface)").
3. Reset module by RESTORE pin of the module (See details in "Module Parameter Configuration"). **30 seconds after power-on**, set the pin low and hold **for 5 s**, the module will recover the parameters before user modified (light recovery, reset the module immediately after release press). 30 seconds after power-on, set the pin low and hold **for 20 s**, the module will be factory reset (deep recovery) immediately. This pin is with an internal pull-up, and the module will not enter recovery mode by default.

- **System parameters reset in light recovery including:**

A. Anti-hijacking password recovers to "00000000". No password will be used by default.

- **System parameters reset in deep recovery including:**

A. Anti-hijacking password recovers to "00000000". No password will be used by default.
B. Serial port baud rate recovers to 115200 bps.
C. Device name recovers to "Tv511u-XXXXXXXX" and X is the last four Bytes of MAC address.
D. Data delay recovers to 10 (0 ms < Delay < 25 ms).
E. Broadcast cycle recovers to 0 (200 ms).
F. Connection interval recovers to 20 ms.
G. Transmit power recovers to 0 dBm.
H. Product ID recovers to 0x5253.
I. Customized broadcast length recovers to 0.

Note: Due to the special use of RESTORE pin (IO0) in circuit design, continuous low level in 30 s before power-on should be avoided, otherwise the module will enter recovery mode.

11 iOS APP Programming Reference

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

Regarding BLE programming in iOS, the key point is the read, write and enable notify switch to **Characteristic (or called channel)** to. To read and write in the channel can realize the direct control on the direct-driven mode functions of the module and no extra MCU 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: Write to a characteristic value.

```
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: Read a characteristic value.

```
[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 *)characteristic;
```

Note: Enable the notify switch of a characteristic value.

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

[self setNotifyValue:NO forCharacteristic:c]; // Disable notify 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:(CBCharacteristic *)characteristic
error:(NSError *)error
```

Note: After each read operation finished, this callback function will be performed. The application layer saves the data that is read in this function.

About the details of scanning, connecting, and other communication operations, please refer to the test App source code (BLE Transmit Module v1.29) for transparent transmission in iOS, 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.

12 Master Reference Code (Transparent Transmission)

Logical relationship: The module uses two IO ports of BCTS, BRTS to send and receive notification and control. These two IOs are in high level and will be triggered when put low in normal. If the module has data to send, BCTS will be set low to inform the microcontroller to receive. If the microcontroller has data to send, BRTS will set low to notification module to receive. The schematic code is as follows:

```
 void main(void)
 {
    EN = 0;                                          // EN, start to broadcast
    while(!BLEMoudleAck("TTM:OK\r\n\0"));            // Wait the mobile device to scan and connect
                                                     // Wait for successful connection, can also join in the time-limited waiting
                                                     // Can also judge the level of the connection prompt signal line
    BRTS = 0;                                        // BRTS is set low, and notify RSBRS02AX module to ready for receiving
    halMcuWaitMs(50);                                // Delay for 50 ms
    UARTWrite( HAL_UART_PORT_0, "TTM:CIT-100ms", 14);// Modify the connection interval, and get confirmation from the serial port
    halMcuWaitMs(50);                                // Delay for 50 ms, and ensure the data is transmitted
    BRTS = 1;                                        // BRTS is set high, and transmission is finished
    while(!BLEMoudleAck("TTM:OK\r\n\0"));            // Wait for successful configuration, can also join in the time-limited waiting

    while(1){                                        // Cyclic test of transmission and receiving
     while(1){
      if(BCTS == 0){                                 // Check, if BCTS is set low, the module will be ready for receiving
        while(BCTS==0);                              // Wait for transmission to be finished, can also do time-limited waiting
        if(UARTRead(uartBuffer) == SUCCESS)          // Read data from serial port
        {... ...}                                    // Use data
      }
      BRTS = 0;                                      // RTS is set low, notify RSBRS02AX module to ready for receiving
      halMcuWaitMs(50);                              // Delay for 50 ms
      send_TX("1234567890");                         // Transmit any data (Beyond 200 Byte)
      halMcuWaitMs(50);                              // Delay for 50 ms, and ensure the data is
```

```
                                                    transmitted
    BRTS = 1;                                       // BRTS is set high, and transmission is finished
    halMcuWaitMs(20);                               // Delay to send the next packet, the delay depends
                                                    on the packet size.

    }
  }
}
}
```

13 Application and Implementation

13.1 Basic Operation of Hardware Design

1. It is recommended to offer the module with a DC stabilized power supply, a tiny power supply ripple coefficient and the reliable ground. Please pay attention to the correct connection between the positive and negative poles of the power supply. Otherwise, the reverse connection may cause permanent damage to the module;
2. Please ensure the supply voltage is between the recommended values. The module will be permanently damaged if the voltage exceeds the maximum value. Please ensure the stable power supply and no frequently fluctuated voltage.
3. When designing the power supply circuit for the module, it is recommended to reserve more than 30% of the margin, which is beneficial to the long-term stable operation of the whole machine. The module should be far away from the power electromagnetic, transformer, high-frequency wiring and other parts with large electromagnetic interference.
4. The bottom of module should avoid high-frequency digital routing, high-frequency analog routing and power routing. If it has to route the wire on the bottom of module, for example, it is assumed that the module is soldered to the Top Layer, the copper must be spread on the connection part of the top layer and the module, and be close to the digital part of module and routed in the Bottom Layer (all copper is well grounded).
5. Assuming that the module is soldered or placed in the Top Layer, it is also wrong to randomly route the Bottom Layer or other layers, which will affect the spurs and receiving sensitivity of the module to some degrees;
6. Assuming that there are devices with large electromagnetic interference around the module, which will greatly affect the module performance. It is recommended to stay away from the module according to the strength of the interference. If circumstances permit, appropriate isolation and shielding can be done.
7. Assuming that there are routings of large electromagnetic interference around the module (high-frequency digital, high-frequency analog, power routings), which will also greatly affect the module performance. It is recommended to stay away from the module according to the strength of the interference. If circumstances permit, appropriate isolation and shielding can be done.
8. It is recommended to stay away from the devices whose TTL protocol is the same 2.4 GHz physical layer, for example: USB 3.0.
9. The antenna installation structure has a great influence on the module performance. It is necessary to ensure the antenna is exposed and preferably vertically upward. When the module is installed inside of the case, a high-quality antenna extension wire can be used to extend the antenna to the outside of the case.
10. The antenna must not be installed inside the metal case, which will cause the transmission distance to be greatly weakened.
11. The recommendation of antenna layout.

    The inverted-F antenna position on PCB is free space electromagnetic radiation. The location and layout of antenna is a key factor to increase the data rate and transmission range.

    Therefore, the layout of the module antenna location and routing is recommended as follows:

    （1） Place the antenna on the edge (corner) of the PCB.

    （2） Make sure that there is no signal line or copper foil in each layer below the antenna.

（3） It is the best to hollow out the red part of the antenna position in the following figure ensure that S11 of the module is minimally affected.

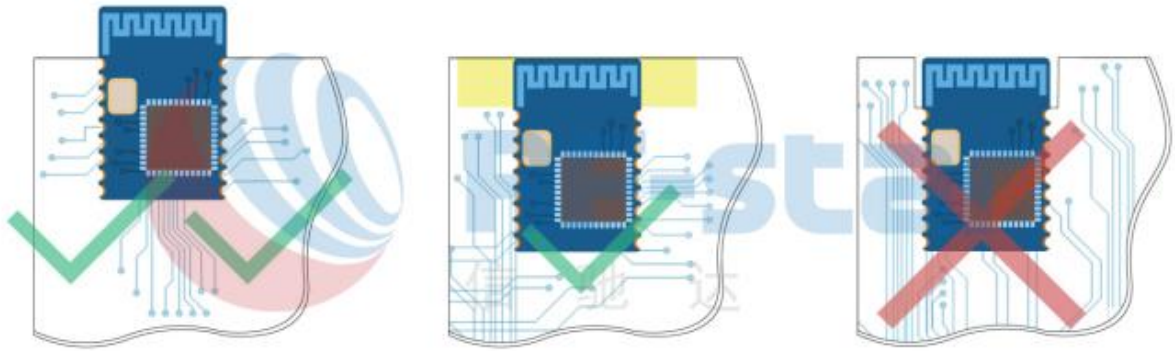


Figure 9. Recommendation of Antenna Layout

12. Antenna

RSBRS02ABRI module is integrated the IPEX version 1 antenna seat, the specification of antenna seat is as follow:

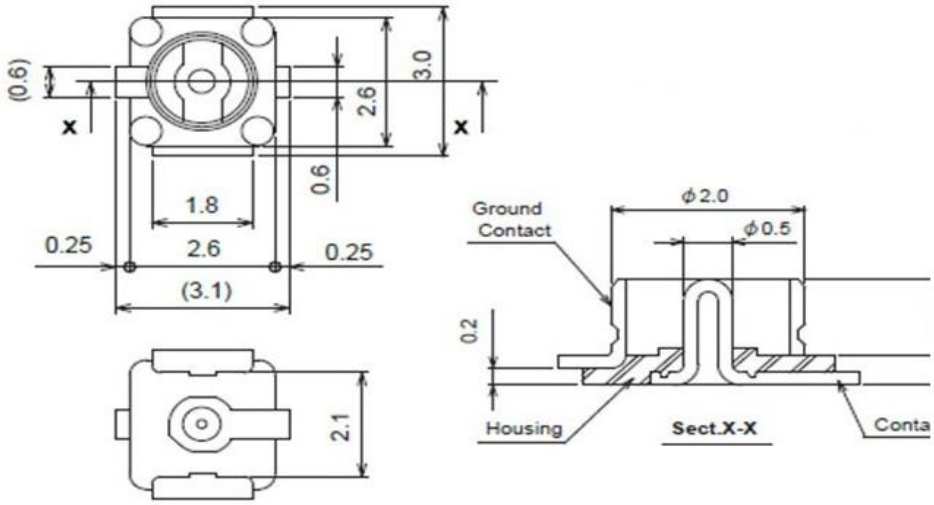


Figure 10. Specification of Antenna Seat

The specification of IPEX wire end is as follow:

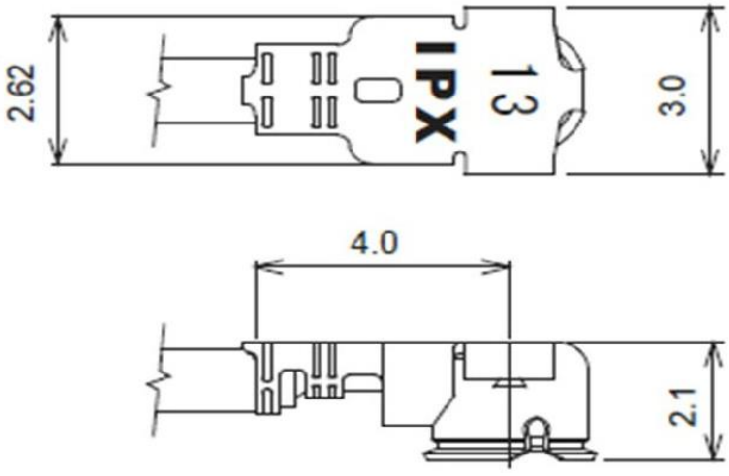


Figure 11. Specification of IPEX Wire

13. Reference Design

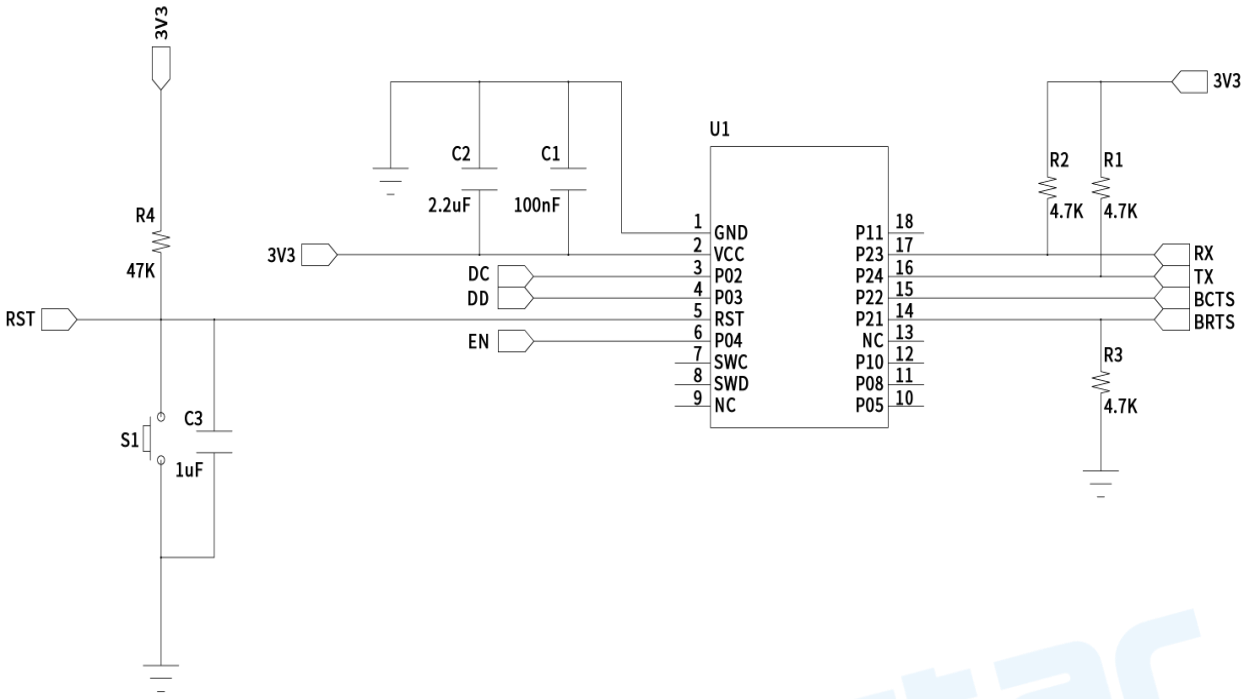


Figure 12. Reference Design

13.2 Trouble Shooting

13.2.1 Unsatisfactory Transmission Distance

1. When there is a linear communication obstacle, the communication distance will be correspondingly weakened. Temperature, humidity, and co-channel interference will lead to an increase in communication packet loss rate. The performances of ground absorption and reflection of radio waves will be poor, when the module is tested close to the ground.
2. Seawater has a strong ability to absorb radio waves, so the test results by seaside are poor.
3. The signal attenuation will be very obvious, if there is a metal near the antenna or the module is placed inside of the metal shell.
4. The incorrect power register set or the high data rate in an open air may shorten the communication distance. The higher the data rate, the closer the distance.
5. The low voltage of the power supply is lower than the recommended value at ambient temperature, and the lower the voltage, the smaller the power is.
6. The unmatchable antennas and module or the poor quality of antenna will affect the communication distance.

13.2.2 Vulnerable Module

1. Please ensure the supply voltage is between the recommended values. The module will be permanently damaged

if the voltage exceeds the maximum value. Please ensure the stable power supply and no frequently fluctuated voltage.

2. Please ensure the anti-static installation and the electrostatic sensitivity of high-frequency devices.
3. Due to some humidity sensitive components, please ensure the suitable humidity during installation and application. If there is no special demand, it is not recommended to use at too high or too low temperature.

13.2.3 High Bit Error Rate

1. There are co-channel signal interferences nearby. It is recommended to be away from the interference sources or modify the frequency and channel to avoid interferences.
2. The unsatisfactory power supply may also cause garbled. It is necessary to ensure the power supply reliability.
3. If the extension wire or feeder wire is of poor quality or too long, the bit error rate will be high.

13.3 Electrostatics Discharge Warnings

The module will be damaged for the discharge of static. RF-star suggest that all modules should follow the 3 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. Take the anti-static circuitry (when inputting HV or VHF) into consideration in product design.
   Static may result in the degradation in performance of module, even causing the failure.

13.4 Soldering and Reflow Condition

1. Heating method: Conventional Convection or IR/convection.
2. Solder paste composition: Sn96.5 / Ag3.0 / Cu0.5
3. Allowable reflow soldering times: 2 times based on the following reflow soldering profile.
4. Temperature profile: Reflow soldering shall be done according to the following temperature profile.
5. Peak temperature: 245 ℃.

Table 5. Temperature Table of Soldering and Reflow

Profile Feature	Sn-Pb Assembly	Pb-Free Assembly
Solder Paste	Sn63 / Pb37	Sn96.5 / Ag3.0 / Cu0.5
Min. Preheating Temperature (T_{min})	100 ℃	150 ℃
Max. Preheating Temperature (T_{max})	150 ℃	200 ℃
Preheating Time (T_{min} to T_{max}) (t_1)	60 s ~ 120 s	60 s ~ 120 s
Average Ascend Rate (T_{max} to T_p)	Max. 3 ℃/s	Max. 3 ℃/s
Liquid Temperature (T_L)	183 ℃	217 ℃

Time above Liquidus (t_L)	60 s ~ 90 s	30 s ~ 90 s
Peak Temperature (T_p)	220 ℃ ~ 235 ℃	230 ℃ ~ 250 ℃
Average Descend Rate (T_p to T_{max})	Max. 6 ℃/s	Max. 6 ℃/s
Time from 25 ℃ to Peak Temperature (t_2)	Max. 6 minutes	Max. 8 minutes
Time of Soldering Zone (t_P)	20±10 s	20±10 s

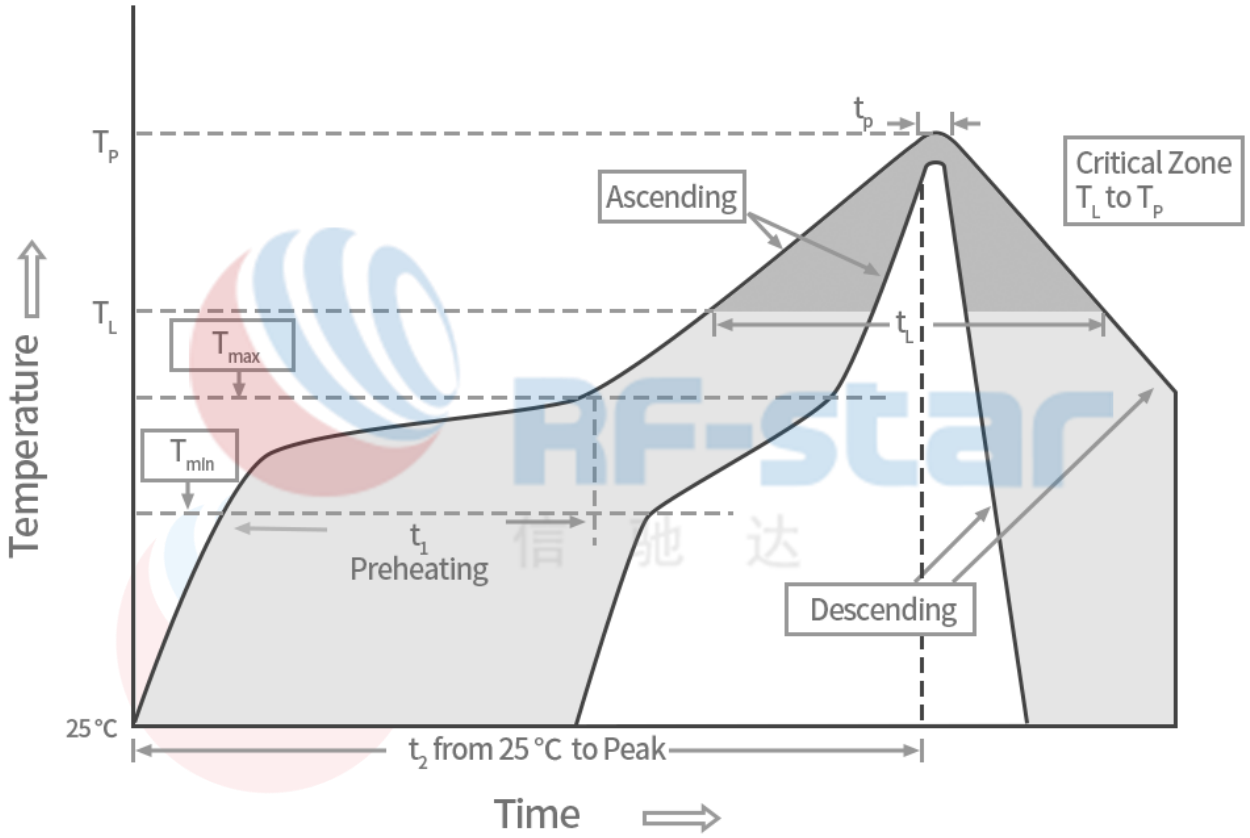


Figure 13. Recommended Reflow for Lead Free Solder

14 Revision History

Version No.	Issue Date	Description
V5.11	2020.04.27	The initial version is released.
V5.11	2020.11.19	Add some AT commands for parameter acquisition. Add AT command for Service UUID configuration. Add 2M communication rate function of BT5.0. Add reference design.
V5.11	2020.12.02	Add AT command “TTM:ADD-?” in the AT command list.

Note:

1. The protocol is updated from time to time. Before using this document, please make sure it is the latest version.

2. For more latest documents, please download it from RF-star Technology official website: www.szrfstar.com

15 Contact Us

SHENZHEN RF-STAR TECHNOLOGY CO., LTD.

Shenzhen HQ:

Add.: Room 601, Block C, Skyworth Building, High-tech Park, Nanshan District, Shenzhen, Guangdong, China

Tel.: 86-755-3695 3756

Chengdu Branch:

Add.: No. B4-12, Building No.1, No. 1480 Tianfu Road North Section (Incubation Park), High-Tech Zone, Chengdu, China (Sichuan) Free Trade Zone, 610000

Tel.: 86-28-6577 5970

Email: sunny@szrfstar.com, sales@szrfstar.com

Web.: www.szrfstar.com

Appendix: Module RF Parameters Test Report

1 Transmission Performance Parameter

Unless otherwise stated, the default conditions for the following parameter tests are: VDD=3.3 V, TA = 25.0 °C, RBW = 100.0 K, VBW =300.0 K, sweep time =100.0 ms.

1.1 Frequency Range

Frequency Range
2402.0 MHz ~ 2480.0 MHz

1.2 Tx Power

Center Frequency (MHz)	Tx Power (dBm)	Tolerance (dBm)	Result
2402	-0.19	0 dBm (±2 dBm)	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

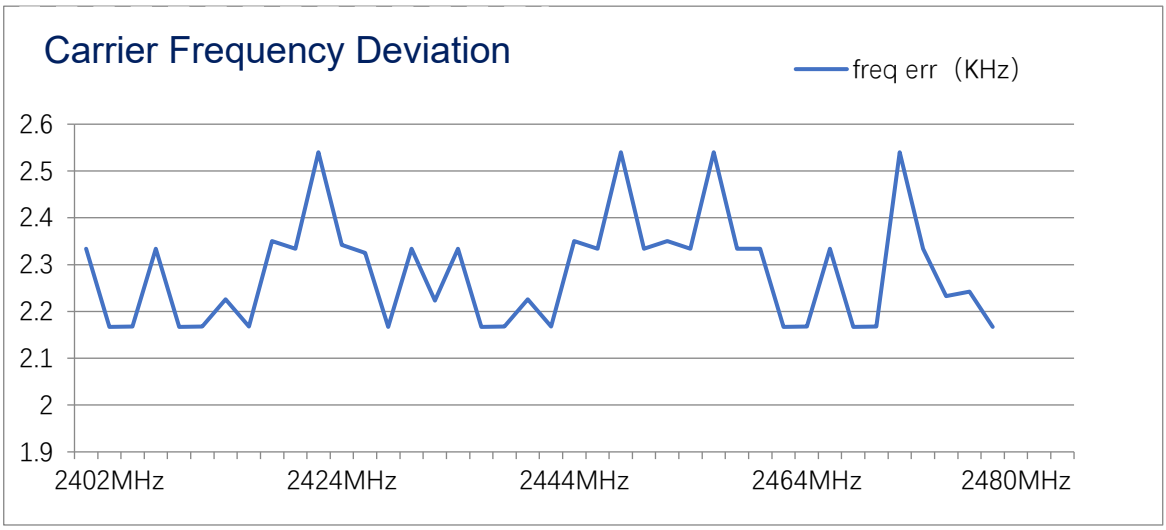
2438	0.41		PASS
2440	0.41		PASS
2442	0.42		PASS
2444	0.44		PASS
2446	0.44		PASS
2448	0.46		PASS
2450	0.46		PASS
2452	0.46		PASS
2454	0.45		PASS
2456	0.44		PASS
2458	0.43		PASS
2460	0.40		PASS
2462	0.39		PASS
2464	0.37		PASS
2466	0.34		PASS
2468	0.31		PASS
2470	0.28		PASS
2472	0.25		PASS
2474	0.22		PASS
2476	0.17		PASS
2478	0.13		PASS
2480	0.09		PASS

1.3 Frequency Deviation

Wave form output = CW

Center Frequency (MHz)	Frequency Deviation (kHz)	FCC Tolerance (kHz)	Result
2402	2.334	±40 kHz	PASS
2404	2.167		PASS
2406	2.168		PASS
2408	2.334		PASS

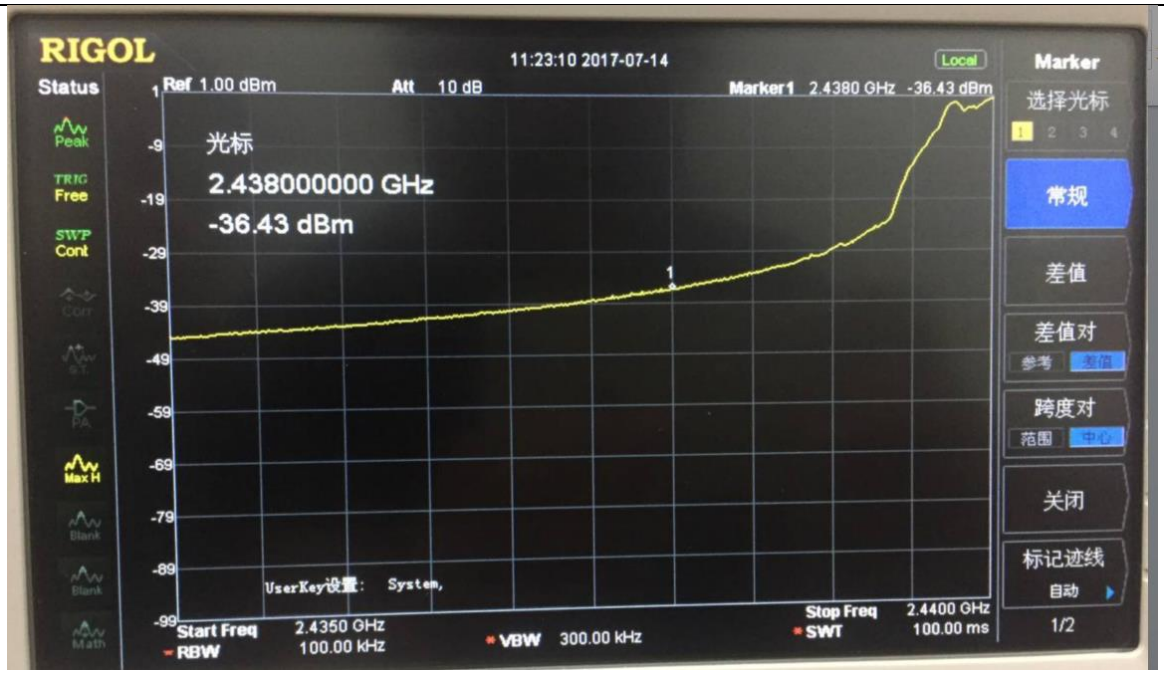
2410	2.167		PASS
2412	2.168		PASS
2414	2.226		PASS
2416	2.168		PASS
2418	2.35		PASS
2420	2.334		PASS
2422	2.54		PASS
2424	2.342		PASS
2426	2.325		PASS
2428	2.167		PASS
2430	2.334		PASS
2432	2.223		PASS
2434	2.334		PASS
2436	2.167		PASS
2438	2.168		PASS
2440	3.666		PASS
2442	2.168		PASS
2444	2.35		PASS
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



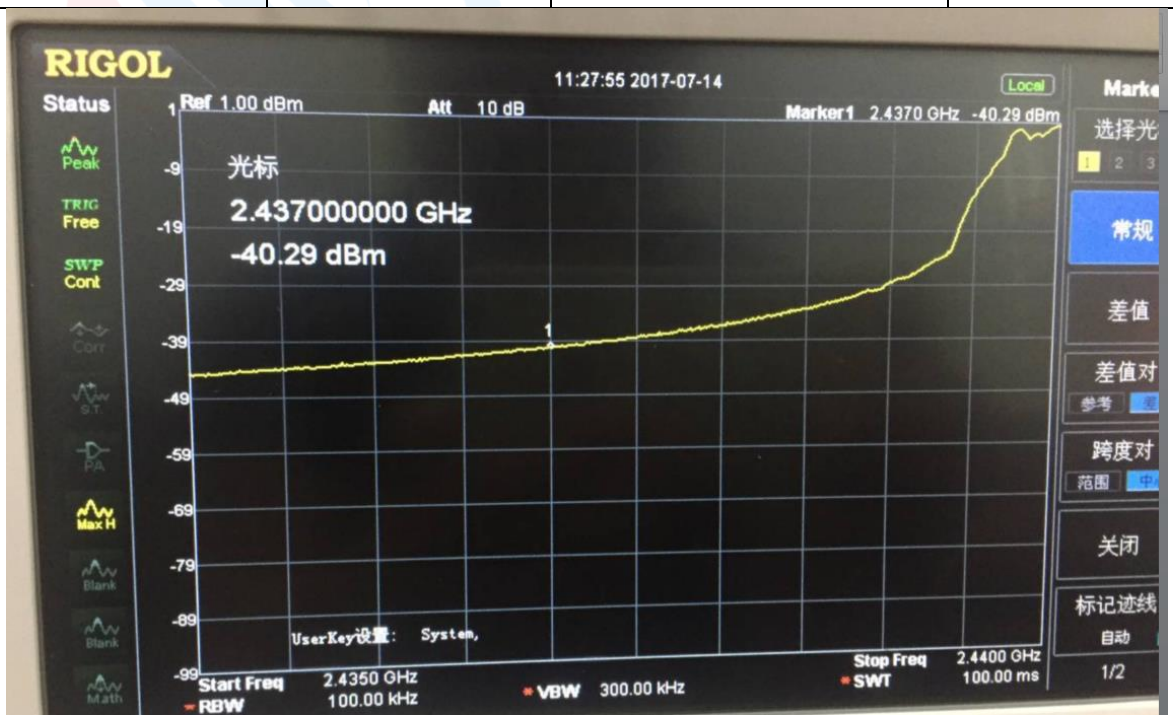
1.4 In-band Spurious

Test Condition: PTX = 0.0 dBm

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

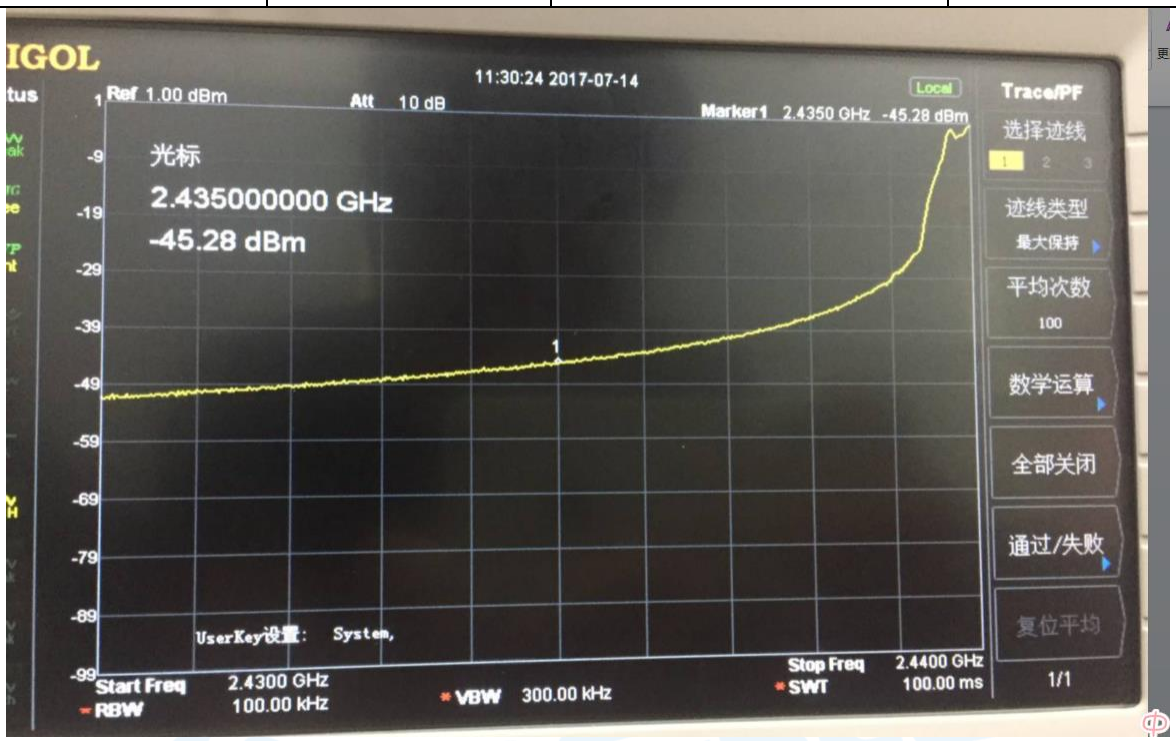


Center Frequency (MHz)	Actual Data (dBm)	RF-PHY.TS.4.2.2 Standard Reference Range (dBm)	Result
2402±3 MHz	-47.24	≤-30 dBm	PASS
2440±3 MHz	-46.25		
2480±3 MHz	-47.38		



Center Frequency (MHz)	Actual Data (dBm)	RF-PHY.TS.4.2.2 Standard Reference Range (dBm)	Result
2402±5 MHz	-43.04	≤-30 dBm	PASS

2440±5 MHz	-42.3		
2480±5 MHz	-42.57		

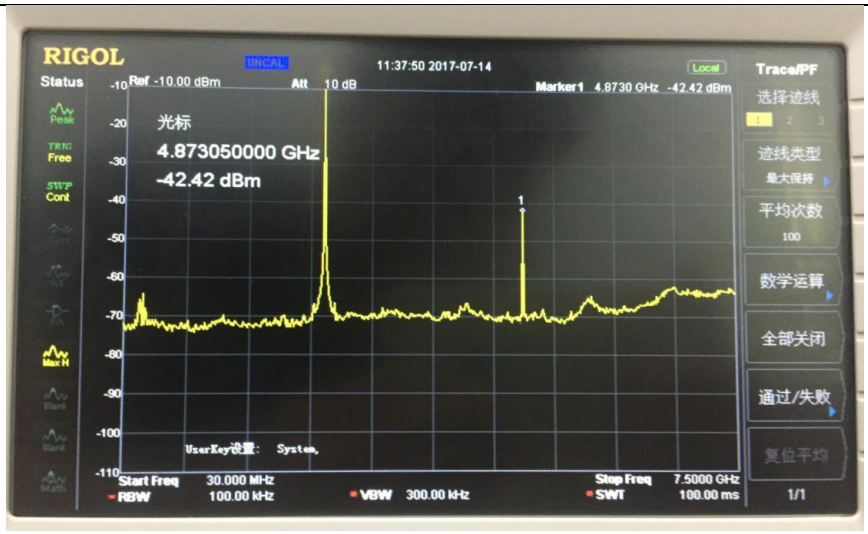


1.5 Out-of-band Spurious

Out-of-band spurious limit of CE conduction harmonic

Test Condition: PTX = 0.0 dBm. Complies with CE standard EN300328V1.8.1.

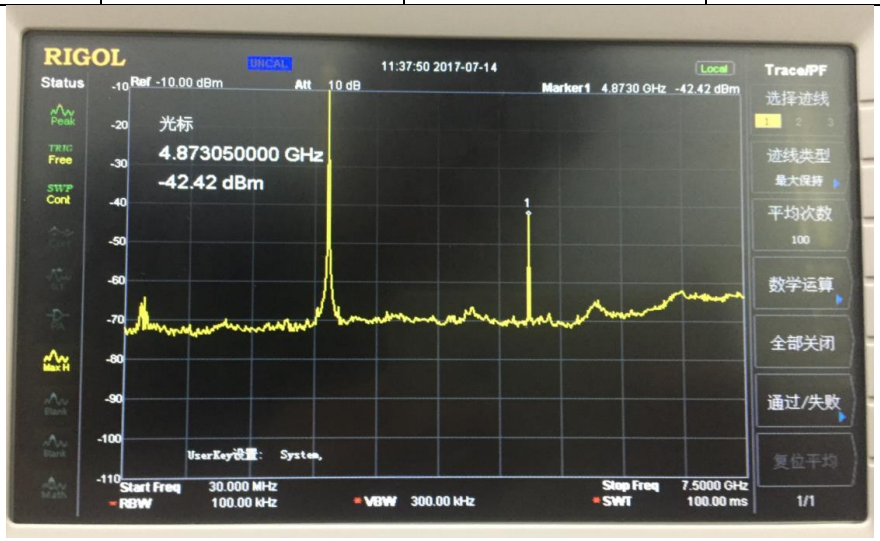
Band	Frequency (MHz)	Level (dBm)	Standard	Unit	Result
	Record (Max.)	level /dBm	EN300 328 Spec.	RBW/VBW	
30 MHz ~ 47 MHz	32	-80	-36	100 k / 300 k	PASS
47 MHz ~ 74 MHz	64	-69.5	-54	100 k / 300 k	PASS
74 MHz ~ 87.5 MHz	75	-70.39	-36	100 k / 300 k	PASS
87.5 MHz ~ 118 MHz	96	-71	-54	100 k / 300 k	PASS
118 MHz ~ 174 MHz	128	-65	-36	100 k / 300 k	PASS
230 MHz ~ 470 MHz	256	-59	-36	100 k / 300 k	PASS
470 MHz ~ 862 MHz	480	-71	-54	100 k / 300 k	PASS
862 MHz ~ 1 GHz	864	-69	-36	100 k / 300 k	PASS
1 GHz ~ 2.36 GHz	2.30	-48.6	-30	1 M / 3 M	PASS
2.5235 GHz ~ 12.75 GHz	2.523	-41.66	-30	1 M / 3 M	PASS



Out-of-band spurious limit of FCC conduction harmonic

Complies with FCC standard PART 15.247.

Carrier Frequency (MHz)	Harmonics Frequency (MHz)	FCC Requirements: < -41.2 dBm		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



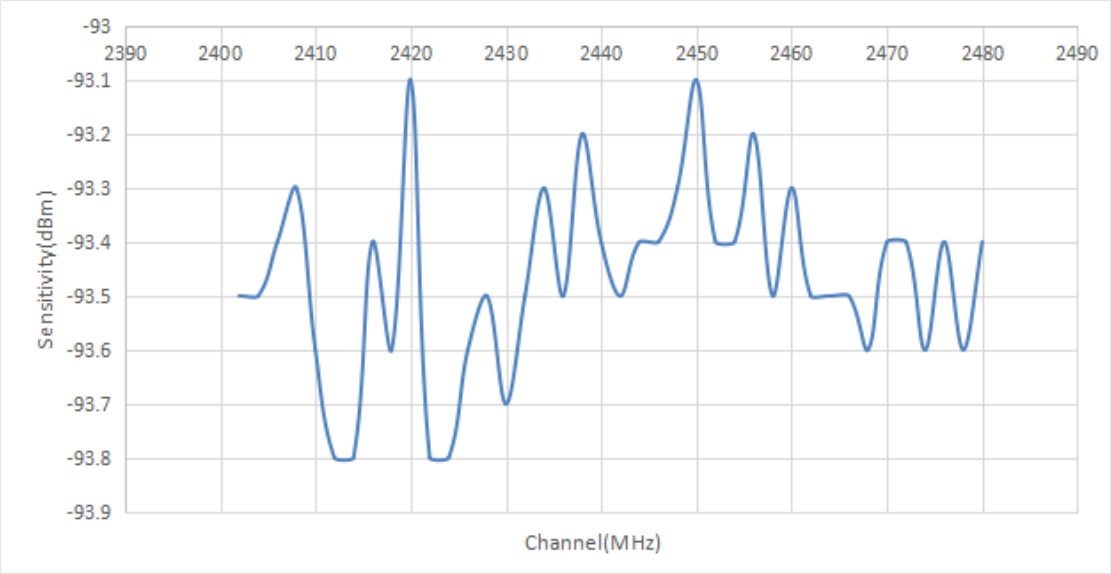
2 Receiving Performance Parameters

Unless otherwise stated, the default conditions for the following parameter tests are: VDD = 3.3 V, TA = 25.0 °C, RBW = 100.0 K, VBW = 300.0 K

2.1 Receiving Sensitivity (Packet Loss Rate PER = 1%)

Center Frequency (MHz)	Receiving Sensitivity (dBm)	RS02A1-B Receiving Sensitivity Range (dBm)	Result
2402	-93.5	-94 dBm	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



2.2 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