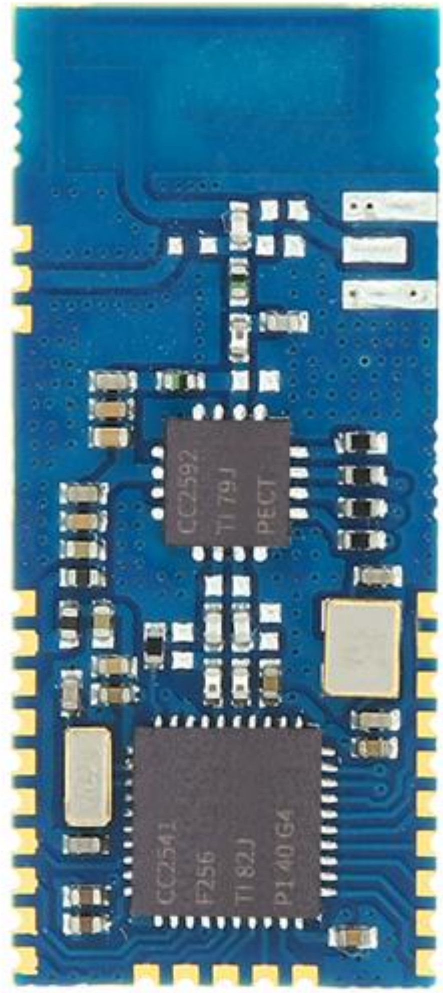
RF-BMPA-2541B1 Bluetooth Low Energy Slave Module and Protocol

Version: V2.32up

(Transparent Transmission)

Shenzhen RF-star Technology Co., Ltd.

Nov. 18th, 2019



RF-BMPA-2541B1

Table of Contents

Table of Figures

Table of Tables

1 Summary

RF-BMPA-2541B1 is a Bluetooth Low Energy (BLE) module based on TI CC2541F256, an 8051 core BLE System-on-Chip (SoC) and a PA of CC2592. This PCB module integrates one 32 MHz crystal, one 32.768 kHz crystal, an LC balun, a chip matching, a high-performance PCB antenna, an IPEX connector, and half-hole antenna interface for different customer needs to develop. It is pre-programmed with the BLE 4.0 stack and an application communication protocol over its serial interface. It enables users to quickly connect their application MCUs to build reliable BLE connection from their products to smart phones.

- Bluetooth 4.2 single mode compliant ISM 2.4 GHz module
- Utilizes the TI CC2541 SoC with 256 KB Flash, 8 KB RAM
- Support slave mode
- Over 120 meter (230 ft) line of site (LOS) distance with integrated antenna
- 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
- Wake-up interrupt, watchdog timer
- AES Security Coprocessor

2 Overview

The Bluetooth LE modules which are mentioned in this document can work in bridge mode (transparent transmission mode).

After powered-on, the module can broadcast automatically. 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 up-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.

3 Features

1. Easy to use, no need of any application experience of Bluetooth protocol stack.
2. UART design for user interface, full-duplex bi-directional communication, and support the minimum baud rate of 4800 bps.
3. Default connection interval of 30 ms, which makes quick connection and enhance the compatible stability of Android phones.
4. Through AT command, support software reset and MAC address acquisition.
5. Through AT command, support the adjustment of Bluetooth connection interval and the control of different transmit rates (dynamic power consumption adjustment).
6. Through AT command, support the adjustment of the transmit power and broadcast interval, the customization of broadcast data and product ID, the configuration of data delay (receiving preparation time of user CPU serial port), the modification of the serial port baud rate and module names (all configurations can be saved after power-off).
7. The length of the UART data packet can be any value lower than 200-Byte (automatic sub-packet of large ones).
8. High-speed transparent transmission rate is maximum to 4 K/s and the stable rate is 2.5 K/s to 2.8 K/s.
9. Through APP, support the modification of module name, UART baud rate and product ID, support the customization of broadcast packet and cycle (all configurations can be saved after power-off).
10. Through APP, support the remote reset of module and the configuration of transmit power.
11. Through APP, support the adjustment of Bluetooth connection interval, but the configuration cannot be saved after power-off (dynamic power consumption adjustment).
12. All IOs are pulled out, including debug IO.
13. Support the prompt pin / general IO flexible configuration of connection status and broadcast status.
14. Support the configuration, modification and recover of anti-hijacking password, and prevent from the malicious connection of a third party, which can be ignored. The independent result notification of password operation to simplify the APP programming.
15. Support factory reset by single pin connected to the ground for 5 s long press and by APP.
16. Prompt of real-time system status in broadcast packet, including battery power, custom product ID (suitable for broadcast applications).
17. Support internal RTC, which can be synchronized check with APP in any time.
18. Support the light recovery and deep recovery modes, which can recover user data flexibly while reserve the essential configurations.
19. Support low-level-enabled mode and pulse-width-enabled mode, and support remote shutdown.
20. Support auto-shutdown after 30 s’ non-connection in the pulse-width-enabled mode.

21. Support timeout (or disconnection) prompt by square wave alarm in the pulse-width-enabled mode.
22. Extremely low power in standby mode (current of 0.3 μA from TI official data for CC2541 SoC), and the measured power consumption data is as follows：

Table 1. Power Consumption of CC2541

Event	Average Current (Integral Measured[1])	Average Current (Ammeter Measured[2])	Duration	Testing Conditions / Remark
Sleeping	0 mA	0 mA	–	EN disconnected
Broadcast	0.79 mA	0.6 mA ~ 1.3 mA	4.42 ms	Broadcast cycle: 200 ms
Connection	1.8 mA	1.9 mA ~ 2.1 mA	2.73 ms	Connection cycle: 20 ms
Single BLE Data Receiving	0.95 mA	9.1 mA	0.512 ms	20 Bytes, 10 times / second
Single BLE data transmitting	0.94 mA	8.8 mA ~ 8.9 mA	0.512 ms	20 Bytes, 10 times / second

Notes:

1. Official test method: Connect a 10 Ω resistor in series in the power circuit, and get the voltage drop waveform with oscilloscope and conduct integrated computation.
2. Multi-meter test method: Connect a multi-meter (set at μA or mA level) in series between the battery and the module to check the value shown, with the test voltage of 3.07 V.

Above is the measured sampling data of module RF-BMPA-2541B1 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

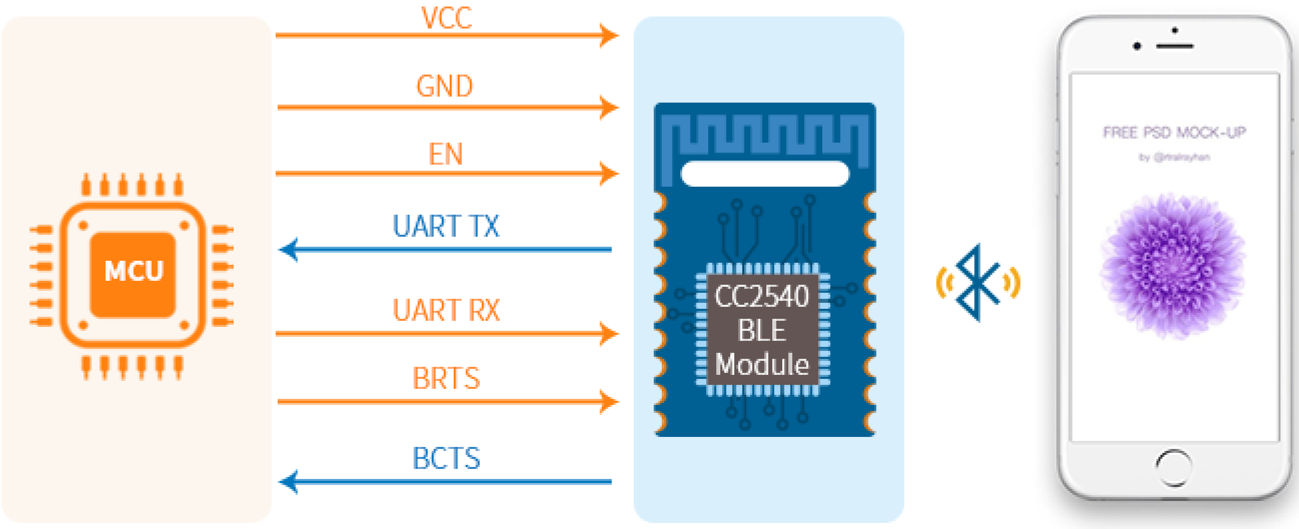


Figure 1. Schematic Diagram of Module (V2.32up) in Bridge Mode

Note:

In order to avoid the high current caused by the output level difference between user CPU 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 RF-BMPA-2541B1

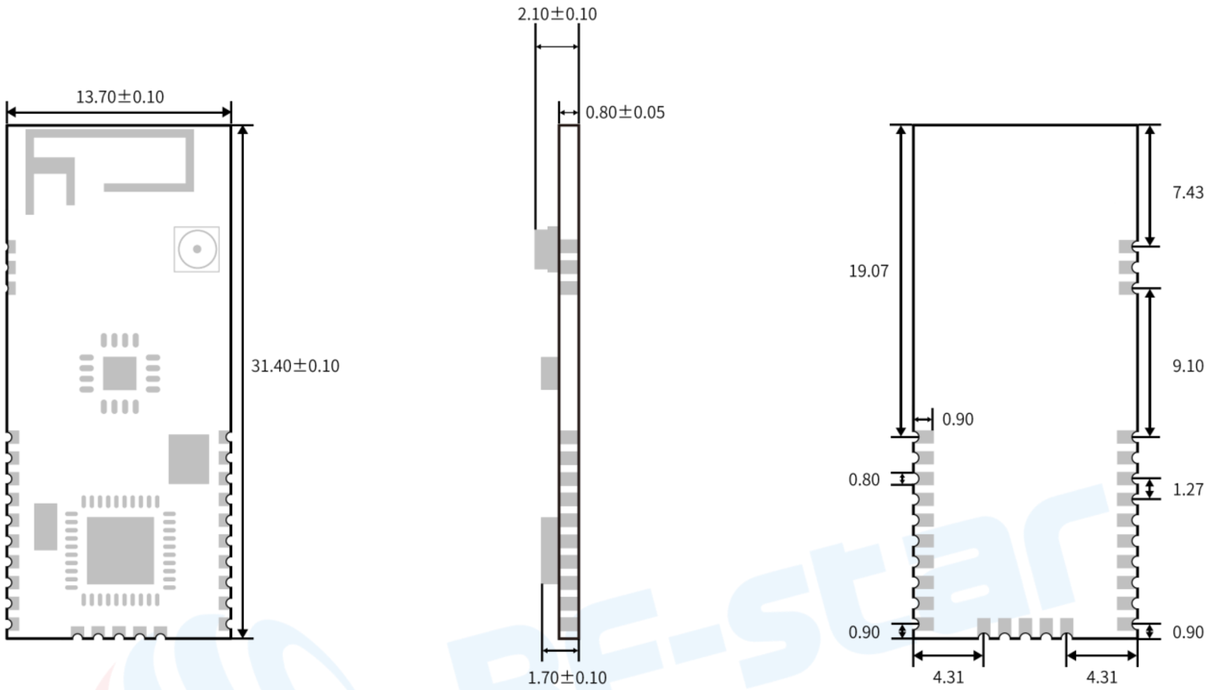


Figure 2. PCB Footprint of RF-BMPA-2541B1

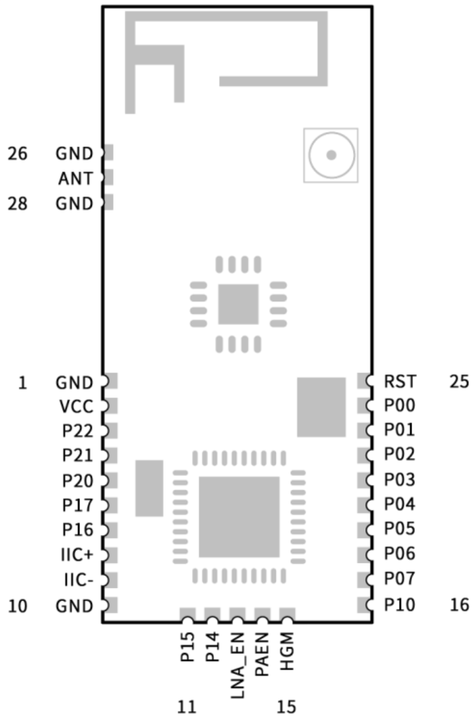


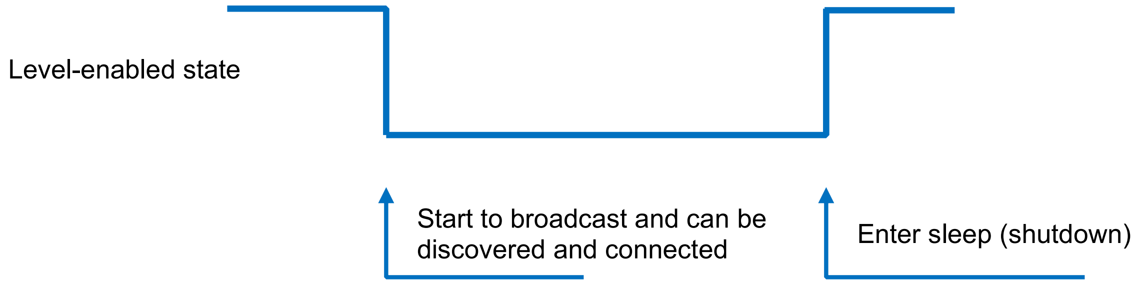
Figure 3. Pin Assignments of RF-BMPA-2541B1

Table 2. Pin Assignments of RF-BMPA-2541B1

Pin	Name	Chip Pin	I/O	Description
1	GND	-	-	Ground
2	VCC	-	-	Power supply 2.0 V ~ 3.6 V
3	IO7	P2.2	O	Sleep mode indicator
4	IO6	P2.1	O	Connection status indicator (Prompt of low level or square wave, see details in "Module Parameter Configuration")
5	EN	P2.0	I	Module-enable control line (level trigger mode by default) • Level trigger mode - Active low, with internal pull-up. 0: Module starts to broadcast, until it connects to the mobile device. 1: Enter sleep mode immediately (0.3 μA), regardless of the current status. • Pulse trigger mode - Every pulse is received (W > 200 ms), the module will shift between boot-up (broadcast, allowed to be found and connected) and shutdown (complete sleep mode). (See details in "Module Parameter Configuration")
6	IO5	P1.7	I/O	
7	IO4	P1.6	I/O	
8	I2C+	-	I/O	I2C+ of CC2541, not used.
9	I2C-	-	I/O	I2C- of CC2541, not used.
10	GND	-	-	Ground
11	IO3	P1.5	I/O	
12	IO2	P1.4	I/O	
13	IO1	P1.3	I/O @ LNA EN	Control CC2592 LNA enable
14	RESTORE / IO0	P1.2	I/O @ PA EN	Factory reset trigger or programmable bi-directional I/O • Within 30 seconds after power-on, keep this pin at low level for 5 s, the system can be partially reset (light recovery). If keep more than 20 s, the system can be completely reset (deep recovery). (See details in "System reset and recovery")

				• Control CC2592 PA enable
15	LNA	P1.1	I/O @ gain mode	Control CC2592 LNA high-low gain
16	-	P1.0	I/O	
17	-	P0.7	I/O	
18	-	P0.6	I/O	
19	BRTS	P0.5	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”.
20	BCTS	P0.4	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”.
21	TX	P0.3	O	Serial port TX
22	RX	P0.2	I	Serial port RX
23	-	P0.1	v	
24	-	P0.0	I	
25	RES	RST	I	Reset, active low
26	GND	-	-	Ground
27	ANT	-	-	External antenna output
28	GND	-	-	Ground

Note:



When the module broadcasts (in this state, it can be discovered and connected) has the following characteristics:

1. If EN pin (Pin 5) is enabled (low), the module will keep broadcasting until it is connected, or EN is set high.

2. After normal disconnection or timeout disconnection, as long as EN is set low, the module will keep broadcasting until it is connected again.

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. In the application where there is a large amount of data transmission, or there is high real-time demand, it is suggested to set the serial port baud rate at the high speed of 115200 bps. Configuration can be saved after power-off.

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 (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.

The module can transmit a packet of the maximum 200-Byte through serial port at one time. According to the packet size, the packet will be sub-packed automatically and sent, with a maximum load of 20 Bytes for each wireless sub-packet. Data packets from mobile devices to the module must be sub-packed automatically (into 1 ~ 20 Bytes/packet) 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 confirming 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. And this delay can be configured through the AT command (see details in "AT command"). 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:OK\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:DISCONNET\r\n\0" from TX. If the disconnection is abnormal, the string will be "TTM:DISCONNET FOR TIMEOUT\r\n\0".

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 (the maximum connection interval to be 2000 ms). 80-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 = 80 * 1000 / T (V is only relevant with T)

   If the Bluetooth connection interval of the module is 20 ms, and 80-Bytes is maximum transmission length for each interval, the theoretical maximum transmission capacity (transmit rate) will be 80 * 50 = 4K Byte/s. Tests have shown that the packet loss is very little when transmit rate under 2 K/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.

8. Here is an example of the communication with 20 ms connection interval in below. Configuration can be set by yourself. But the lower the transmit rate V_0, the less packet leakage is.

Table 3. Transmission Rate at Interval of 20 ms

Communication Mode	BLE Connection Interval T (ms)	Highest Theoretical Transmit Rate V (Byte/s) V = 80*1000/T	Serial Data Packet Length (Byte)	Serial Port Transmission Interval TS (ms)[1]	Actual Transmit Rate V_0 (Byte/s) V_0 = L*1000/TS	Remarks
1	20	4 K	80	TS >= T, If TS = 20 ms	80*1000/20 = 4 K	Too low TS, not recommended
2	20	4 K	200	TS >= T*3, If TS = 70 ms	200*1000/70 = 2.8 K	
3	20	4 K	200	TS >= T*3, If TS = 80 ms	200*1000/80 = 2.5 K	
4	20	4 K	80	TS >= T, If TS = 35 ms	80*1000/30 = 2.6 K	
5	20	4 K	70	TS >= T, i If TS = 30 ms	70*1000/30 = 2.3 K	
6	20	4 K	60	TS >= T, If TS = 30 ms	60*1000/30 = 2 K	

7	20	4 K	40	TS >= T, If TS = 30 ms	40*1000/30 = 1.3 K	
8	20	4 K	00	TS >= T, If TS = 30 ms	20*1000/30 = 666 Byte	

Note:

1. When L<80, TS >= T. When 80<L<160, TS >= T*2. When 160<L<200, TS >= T*3.
2. Specific communication mode can be designed according to the practical application. Packet length of serial port can be designed between 80 Bytes and 200 Bytes (large packet transmission). According to BLE protocol, the formulas are as follows:

   When L<80，TS >= T,

   When 80<L<160，TS >= T*2,

   When 160<L<500，TS >= T*3,

   Transmission modes that comply with the 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 high package loss will be caused. It is recommended to add check-sum re-transmission mechanism. In other words, when a serial port data packet is as big as 80 Byte < L < 200 Bytes, the data can be sent to the module for one time, but certain time needs to be spared for module data transmission by Bluetooth. Otherwise there will be a rear-end data collision. For example, when the connection interval T = 20 ms, if the data packet length L = 200, the TS must larger than T*3 = 60 ms. So, setting TS = 70 ms is a logical choice.

9. The size of the serial port data packets can be various and the length can be any value less than 200 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, 40 Bytes, or 60 Bytes in length.

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.**

- AT Command List

Table 4. AT Command List

AT Command	Saved After Power-off	Parameter and Description	Possible Response	Remarks
TTM:CIT-Xms (Effective after successful connection)	No	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:NAM-?	Yes	Acquire module name.	TTM:NAM-xxxxxxxxxxxx, "xxxxxxxxxxxx" 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 = "4800", "9600", "19200", "38400", "57600", **"115200"**.	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-xxxxxxxxxxxx, "xxxxxxxxxxxx" is	MAC address is returned.

			module MAC address.	
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:RST-SYSTEMRESET	-	System reset.	Module is working!	Reset module.
TTM:RST-RSTPWD	-	Light recovery.	Module is working!	Recovery password.
TTM:RST-RESET	-	Deep recovery.	Module is working!	Recovery all parameters.
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:ADD-+Data	Yes	Set customized broadcast packet. Data is the customized broadcast packet. Data length <= 16.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:PID-+ Data	Yes	Set customized product ID. Data is the customized product ID. Data length = 2. **00 00** is by default.	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:TPL-(X)	Yes	Set transmit power (in dBm). X = "**0**", "-6", or "-23".	TTM:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
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:RSI-ON	-	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:RTC-?	-	Acquire RTC time.	TTM:RTC-xxxxxxxxxxxxxx, "xxxxxxxxxxxxxx" is the data.	Successful operation. Incorrect command format.
TTM:RTC-X	-	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 output (in ms). X = "**0**", "2", "5", "10", "15", "20", or "25". 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:OK\r\n\0 TTM:ERP\r\n\0	Successful operation. Incorrect command format.
TTM:PWD-?	-	Acquire connection password.	TTM:PWD-XXXXXX.	Connection password is returned.
TTM:PWD-xxxxxxx	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:VER-?	-	Acquire version number.	TTM:VER-XXXXXX, “X” means the version number.	Version number is returned.

Note: Word in bold blue is by default. Row in yellow means not saved after power-off.

- 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 4s (iOS 5.1.1) 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 iOS5.1.1, 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).

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

- 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 = "4800", "9600", "19200", "38400", "57600", **"115200"** (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 = "4800", "9600", "19200", "38400", "57600", **"115200"** (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.

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

Test shows that the baud rate can be modified immediately in iOS6 and above versions, but not in iOS5. Users can set through PC, or through the BLE APP interface of mobile devices. (See details in "module parameter configuration [service UUID: 0xFF90]")

- 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.

- Module Reset

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

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

“Module is working!": It means the successful operation.

- Module Reset - Light Recovery

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

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

“Module is working!": It means the successful operation.

- 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.

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

“Module is working!": It means the successful operation.

- 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.

- 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.

- 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.

- Transmit Power Configuration

Input the following string "TTM:TPL-(X)" to RX to set the corresponding transmit power (in dBm). Wherein X = **"0"**, "-6", or "-23", after the power is amplified, the transmit power is 18 dBm, 14 dBm, -3 dBm (all data format is in ASCII code). For example: "TTM:TPL-(0)" means the transmit power is set as 18 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.

Note: This configuration cannot be saved after power-off.

- Internal Enable

Input the following string "TTM:EUP-ON\r\n\0" to RX to enable internal pull-up of EN pin which is the default configuration. Input the following string "TTM:EUP-OFF\r\n\0" to RX to disable internal pull-up of EN pin. When broadcast during disabled internal pull-up, more than 80 μA current.

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.

Note: This configuration cannot be saved after power-off. The default configuration is internal pull-up enabled.

- 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. The scheme of data delay configuration is as follows:

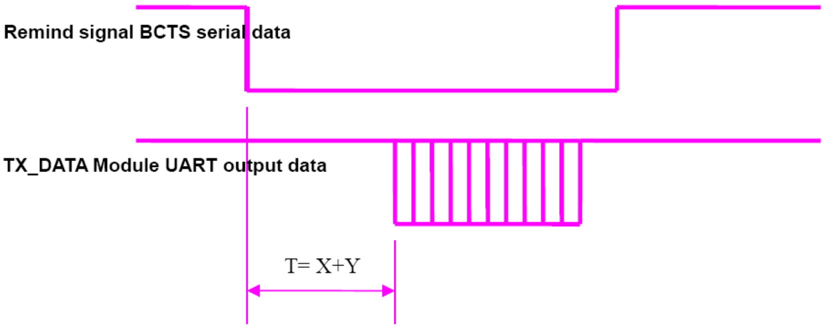

Remind signal BCTS serial data
TX_DATA Module UART output data
T= X+Y

Figure 4. Scheme of Data Delay Configuration

- 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.

- 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.

8 BLE Protocol (APP Interface)

- Bluetooth Data Channel [Service UUID:0xFFE5]

Characteristic UUID	Operation	Bytes	Default Value	Remarks
FFE9 **(handle:0x0013)**	Write	20	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 **(handle:0x000E)**	Notify	20	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, (01 00 is needed to be written in 0x000E+1 = 0x000F by BTool), 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”.

- **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:0x0045)	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:0x0048)	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(01 00 is needed to be written into 0x0048+1 = 0x0049 by BTool).
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.

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

Characteristic UUID	Operation	Saved or Not	Bytes	Default Value	Remarks
FF91 (handle:0x0062)	Read / Write	Yes	16	Tv232u-x xxxxxxx (ASCII string with terminator)	Device name, XXXXXXXX for the last four Bytes of the MAC address.
FF92 (handle:0x0065)	Read / Write	No	1	1	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:0x0068)	Read / Write	Yes	1	1	Set baud rate of serial ports: 0: 4800 bps 1: 9600 bps 2: 19200 bps 3: 38400 bps 4: 57600 bps **5: 115200 bps**
FF94 (handle:0x006B)	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:0x006E)**	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:0x0071)**	Read / Write	Yes	2	0x0000	Set product ID
FF97 **(handle:0x0074)**	Read / Write	No	1	1	Set transmit power: **1: 18 dBm** 2: 14 dBm 3: -3 dBm
FF98 **(handle:0x0077)**	Read / Write	Yes	16	Default broadcast packet.	Set customized broadcast data Customizing broadcast data: 0 < n ≤ 16. See details in “Broadcast Data Configuration”.
FF99 **(handle:0x007A)**	Write	-	1	None	Remote control extension channel: **0x01**: Saving-trigger control of I/O configuration output. Writing 0x01 will trigger the saving of current I/O configuration and output status. IO7 ~ IO1 will be initialized to use the saved configuration and output status when re-power on. But IO0 is always set as input by default when power on, which works as the test port of factory reset. **0x02**: Remote shutdown control. In pulse-enable mode, writing 0x02 to this channel can shut down the module remotely.
FF9A	Read /	Yes	1	0b00000000	System function EN switch:

(handle:0x007D)	Write				**BIT0**: EN mode configuration. 0 is by default and is corresponding to the low-level enabled. 1 means pulse-enabled. The module will switch between boot-up (starting broadcast) and shutdown (stopping broadcast) in turn, once EN pin receives a pulse every time. Effective pulse width T must meet: W > 200 ms. When the broadcast time exceeds 30 s and the module is still not connected, it will shut down automatically. **BIT1～BIT7**： Unused by now.

Note: Command in Row in blue means this function cannot be saved after power-off.

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 "Tv2vvv- XXXXXXX\0" (16 Byte), wherein vvvv 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 4S (iOS 5.1.1). But it will be effectively very quickly if the connection interval is changed from a high frequency one (such as: 100 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. 0x0000 is by default.

FF97 is the channel to set transmit power by write to this channel. **The setting cannot be saved after power-off. 1 (18 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").

FF99 is the channel for remote control extension. By writing different values, the specific control functions can be realized. Writing **0x01** to this channel will trigger the module to save the current configuration and the output status of all I/Os (except IO0). When re-power on, the module will always initialize IO7 ~ IO1 with the saved settings and output status. While IO0 is always set to default input status to work as the triggering I/O of factory reset. But afterwards, IO0 can also be set as output, just as other I/Os. In the pulse-enabled mode, writing 0x02 to this channel will shut down the module remotely. But the function is invalid in level-enabled mode.

FF9A is the channel of system function enabled switch. Writing through BIT0 ~ BIT7 can be turned on or turn off specific functions of the system. 1 means on and 0 means off. All 0b00000000 is by default. This setting can be saved after power-off.

BIT0: 0 is default to level-enabled mode. In this mode, low-level is enabled to start broadcast and high-level is enabled to sleep (0.3 μA). When the bit is set to 1, the module will be in pulse-enabled mode. The module will be switched between on (starting broadcast) and off (deep sleeping, 0.3 μA) in turn, after get a legal pulse width (W > 200 ms). If the module is in the connection status, “off” takes no effect. While the module is in the broadcast status, “off” takes effect.

BIT1~BIT7: Reserved.

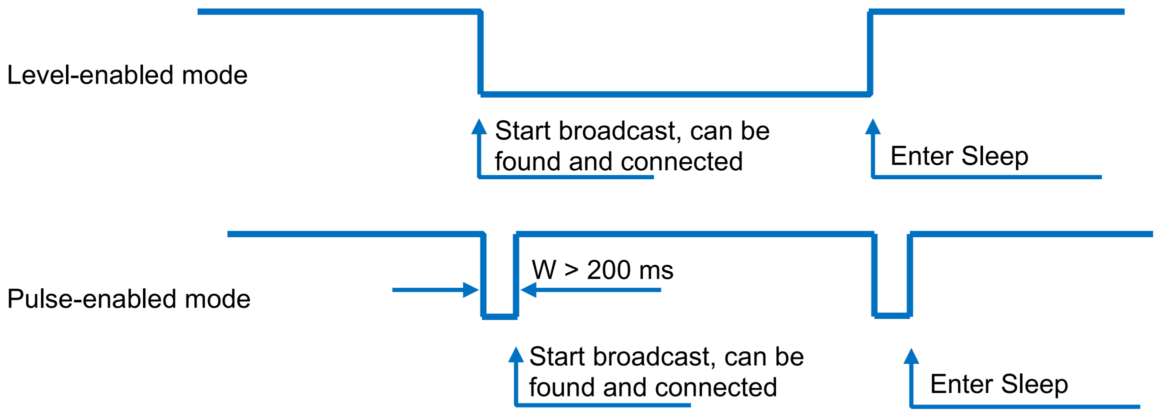

Level-enabled mode
Start broadcast, can be found and connected
Enter Sleep
Pulse-enabled mode
W > 200 ms
Start broadcast, can be found and connected
Enter Sleep

Figure 5. Level Enabled Model & Pulse Enabled Model

In level-enabled mode, broadcast (can be found and connected) has the following features:

1. If EN pin is enabled (set low), the module will keep broadcasting, until it is connected or EN is set high.
2. Regularly disconnected or timeout disconnected, as long as EN is set low, the module will always keep broadcasting, until it is connected again.

In pulse-enabled mode, broadcasting (can be found and connected) has the following features:

1. If broadcasting last for 30 s after enabled, but still not connected, the module will stop broadcasting and shut down.
2. If regularly disconnected, the module will continue to broadcast for 30 s. If it is still not connected, the module will stop broadcasting and shut down.
3. If disconnected due to timeout, the module will keep broadcasting until it is connected again. And in this case, EN shutdown takes no effect.

In level-enabled mode, when IO6 works as signal prompt pin (prompt of Bluetooth connection status by default) and is connected at low level, it will output high level, if Bluetooth is disconnected (either timeout or active disconnecting) and not re-connected.

In pulse-enabled mode, when IO6 works as signal prompt pin (prompt of Bluetooth connection status by default), the output signal has the following features:

1. When connected, it will output low level pulse (1 s) for once.
2. When Bluetooth is regularly disconnected （active disconnecting by APP), it will output low level pulse (0.5 s) for once.
3. When Bluetooth is disconnected due to timeout, it will output the square wave of 2 Hz and last for 2 minutes. During this period, it will keep broadcasting and cannot be shut down, until the module re-connects with master device.

Broadcast status & IO6 prompt modes in different EN modes are summarized as follows:

Table 5. Broadcast Status & IO6 Prompt Modes

Module Status	Enabled but Not Connected		Connected		Actively Disconnected		Timeout Disconnected	
	IO6 Prompt	Broadcast Status	IO6 Prompt	Broadcast Status	IO6 Prompt	Broadcast Status	IO6 Prompt	Broadcast Status
Level Enabled Mode	High Level	Keep Broadcast	Low Level	Stop Broadcast	High Level	Keep Broadcast	High Level	Keep Broadcast
Pulse Enabled Mode	High Level	Broadcast for 30 s	Low Level Pulse W = 1 s	Stop Broadcast	Low Level Pulse W = 0.5 s	Broadcast for 30 s	2 Hz Square Wave for 2 min.	Keep Broadcast

- Device Information [Service UUID:0x180A]

Characteristic UUID	Operation	Bytes	Default Value	Remarks
2A23 **(handle:0x0003)**	Read	8	xxxxxx0000xxxxxx (Hex)	System ID, where xxxxxxxxxxxx is the MAC address of module, with low Byte in front.
2A26 **(handle:0x0005)**	Read	5	V2.32u (ASCII)	Firmware version number of the module.

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: Vx.xx, wherein xx is firmware version number.

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 9 Bytes):

{

0x00,0x00,	Customized device type code, 00 00 is by default, and can be set by AT command.
0x00, 0x00, 0x00, 0x00,	None
0x00,	None
0x00, 0x00,	None

}

- Customized Broadcast Data

Customizing the broadcast packet can be realized by AT command, and the maximum length is16 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 RESET 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-hijack password recovers to "000000". No password will be used by default.

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

A. Anti-hijack password recovers to "000000". No password will be used by default.
B. Serial port baud rate recovers to 115200 bps.
C. Device name recovers to "Tv232u-XXXXXXXX" and X is the last four Bytes of MAC address.
D. Data delay recovers to 0 (500 μs < Delay < 1 ms).
E. Broadcast cycle recovers to 2 (200 ms).
F. Connection interval recovers to 30 ms.
G. Transmit power recovers to 18 dBm.
H. Product ID recovers to 0x0000.
I. Customized broadcast length recovers to 0.
J. All customized broadcast data recovers to 0. Default broadcast data is used but customized broadcast data).
K. EN mode recovers to 0. Level-enabled mode is by default.

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 LE 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-drive 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: 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 read 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 *)characteristic;
```

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:(CBCharacteristic *)characteristic error:(NSError *)error
```

Note: after each reading operation, 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.

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.

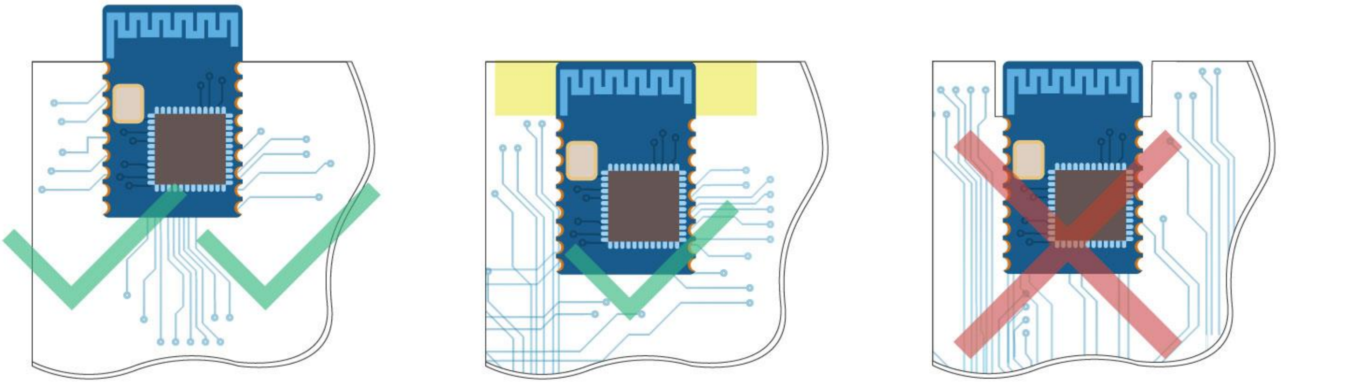


Figure 6. Recommendation of Antenna Layout

12. Antenna

RF-BMPA-2541B1 module is integrated the IPEX version 1 antenna seat, the specification of antenna seat is as follow:

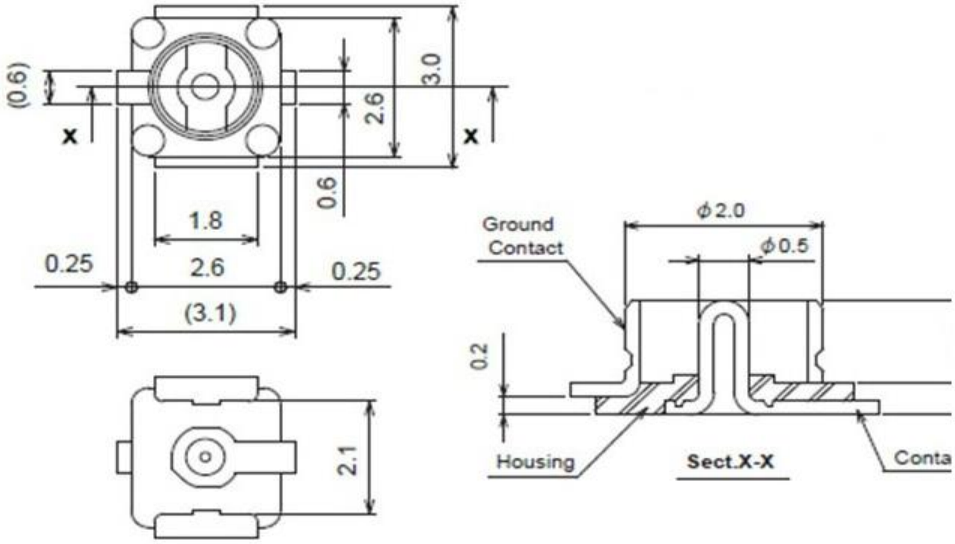


Figure 7. Specification of Antenna Seat

The specification of IPEX wire end is as follow:

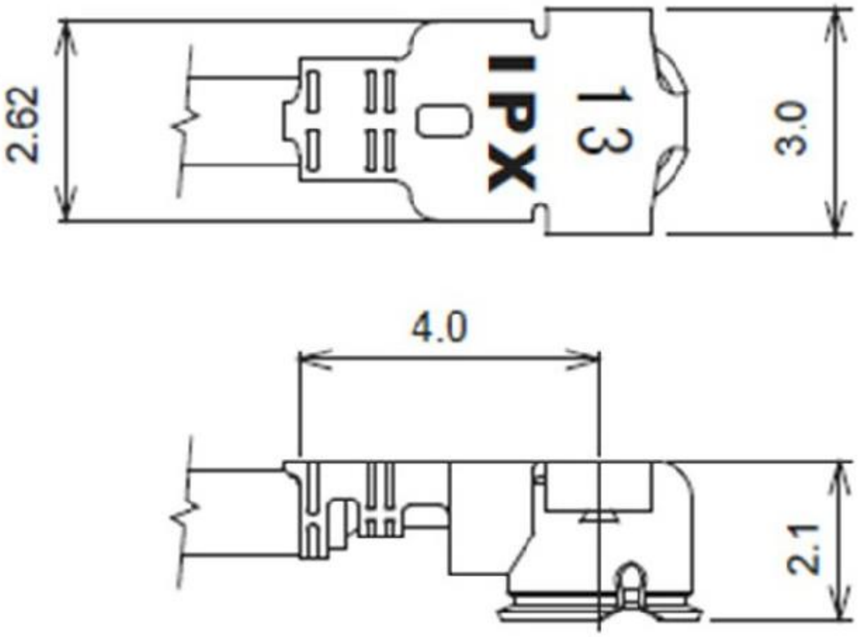


Figure 8. Specification of IPEX Wire

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. Temperature measurement: Thermocouple d = 0.1 mm to 0.2 mm CA (K) or CC (T) at soldering portion or equivalent methods.
3. Solder paste composition: Sn/3.0 Ag/0.5 Cu
4. Allowable reflow soldering times: 2 times based on the following reflow soldering profile.
5. Temperature profile: Reflow soldering shall be done according to the following temperature profile.
6. Peak temperature: 245 ℃.

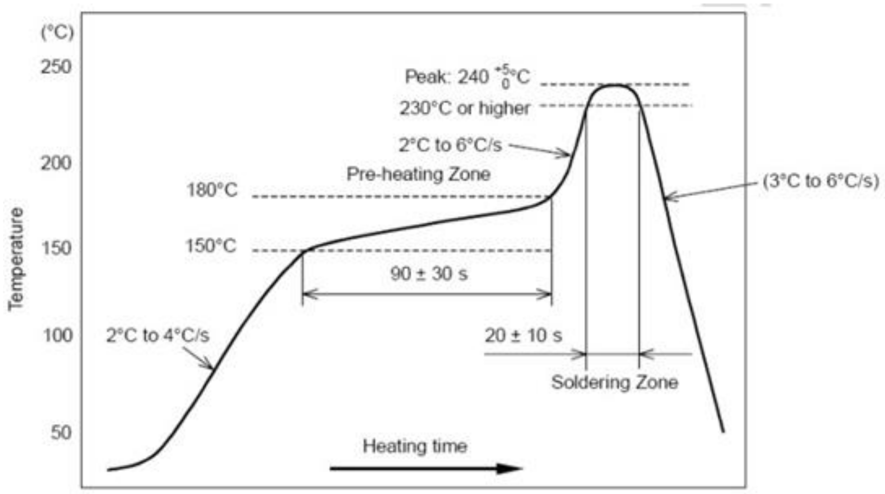


Figure 9. Recommended Reflow for Lead Free Solder

14 Revision History

Version No.	Issue Date	Description
V2.22up	2016.07.22	The initial version is released.
V2.22up	2016.10.27	Modify the Transmit power.
V2.22up	2018.08.02	Update company address.
V2.32up	2019.11.18	Add AT command of revision number acquisition. Add AT command of password configuration. Add AT command of password acquisition.

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

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.: 86-755-8632 9687

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