

Wireless-Tag WT52810-S1

Bluetooth Low Energy 5.0 Module DATASHEET

Overview:

WT52810-S1 is a high-performance, ultra-low energy (Bluetooth Low Energy) RF transceiver system module that uses Nordic's nRF52810 as the core processor. It has an ultra-small volume package currently used in the industry, with an overall size of 15.8* 11.9*2.0, suitable for most application scenarios requiring small size. The module supports Bluetooth S112 SoftDevice, a Bluetooth 5.0 pre-qualified protocol stack.

Customer's MCU is connected to the module that is used as a slave through UART port. After the module is started, it will automatically send broadcast code. The host device APP scans and searches for the module's broadcast to establish a connection. After success, it can conduct two-way communication link with mobile device APP by serial port. Through UART interface, the user can exchange data or control the communication parameters of the module. The meaning of data is defined by the upper application user. The mobile device can operate to the module through APP, the data will be received by the module and pushed to the MCU through UART. After the module received a data packet from the client MCU serial port, it will automatically forward it to the mobile device.

Features:

- Operating temperature: -25°C~75°C, typical 25°C
- CPU: nRF52810-QFAA (32-bit ARM Cortex-M4F processor)
- Memory: 192KB flash + 24KB RAM memory
- Transparent transmission (Bridged method), easy and fast to use even for developers without Bluetooth protocol stack development experience.
- Adopt standard UART (TTL) interface, bidirectional data reading and simple to operate;
- Support serial AT commands, users can modify the module's serial port baud rate (default 115200bps), name, MAC address and other basic parameters;
- Serial data packet length: single data packet can support data length up to 244byte;
- Operating distance: 0~45m, class II level;
- System: support Android and IOS
- Supply voltage: 2.0~3.6V (3.3V typical);
- PCB antenna, or external antenna optional (to be customized)
- Size: 15.8*11.9*2.0mm
- Operating temperature: -20°C ~ 75°C
- Sleep current: 0.3uA

Applications:

- Electronic scales
- Electronic cigarettes
- Smart cups
- Smart bracelets
- Smart watches
- Bluetooth toys
- Intelligent hardware
- Smart home

Wireless-Tag

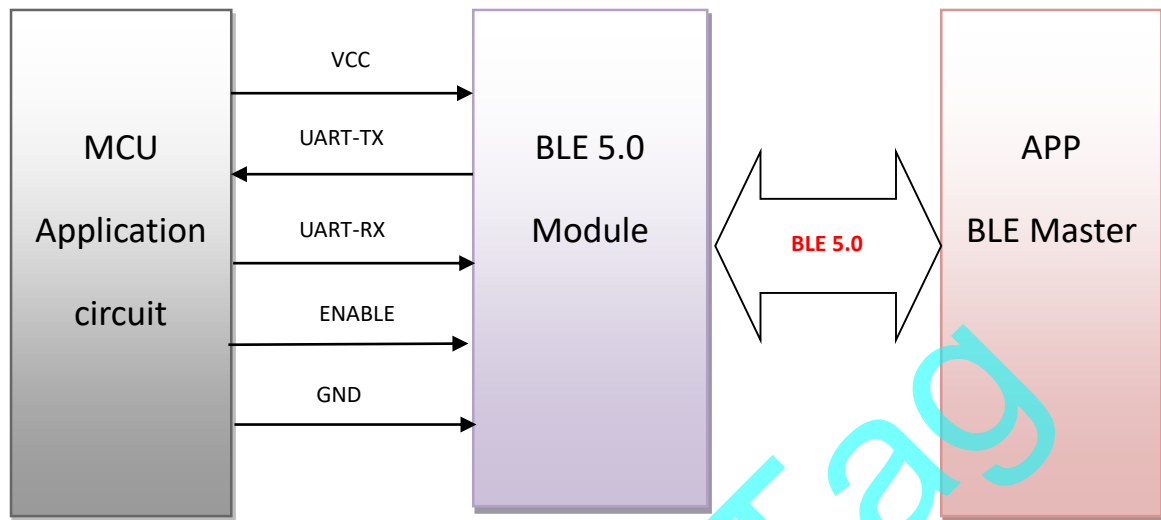
Revision History			
Version	Author	Date	Description
0.0.1	Brussin	August 03, 2020	First release

Wireless-Tag

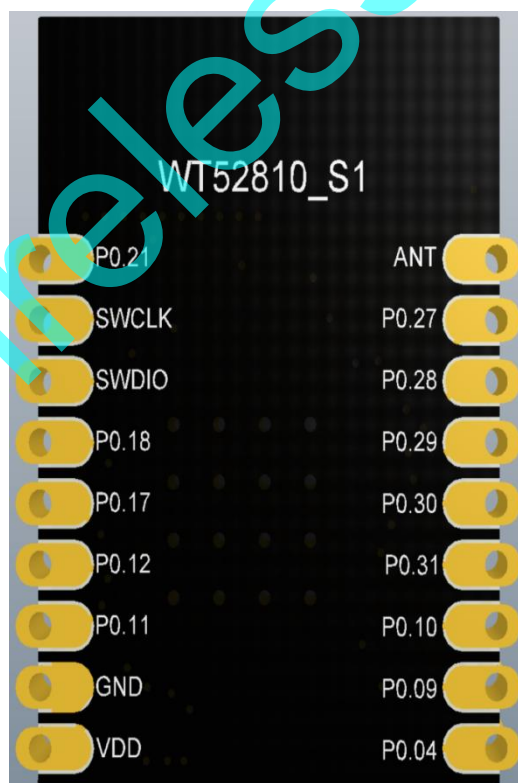
Contents

1. Working Mode Diagram	5
2. Module Pin Definitions	5
2.1. Pin Description	6
2.2. Reflow Soldering Curve	6
3. Module Power Consumption Description	7
4. Serial Transparent Transmission Protocol Instructions	8
5. Serial AT Commands Description	8
5.1. Test Command	8
5.2. Query Module Version Command	9
5.3. MAC Address Operation Command	9
5.4. Serial Communication Baud Rate Operation Command	9
5.5. Device Name Operation Command	10
5.6. Module Transmit Power Setting Command	10
5.7. Reset Module Command	11
5.8. Restore Default Configuration Command	11
5.9. Customize Advertising Data Command	11
5.10. Customize Product Serial Number command	11
5.11. Bluetooth Advertising Interval Setting Command	12
5.12. Bluetooth Connection Interval Reset Command	12
5.13. Low-power Sleep Mode Setting Command	12
5.14. Firmware Upgrade Command	13
6. BLE Protocol Description (APP Interface)	13
7. Error CODE Description	13
8. Bluetooth Module Application Examples	13
9. Appendix	20

1. Working Mode Diagram



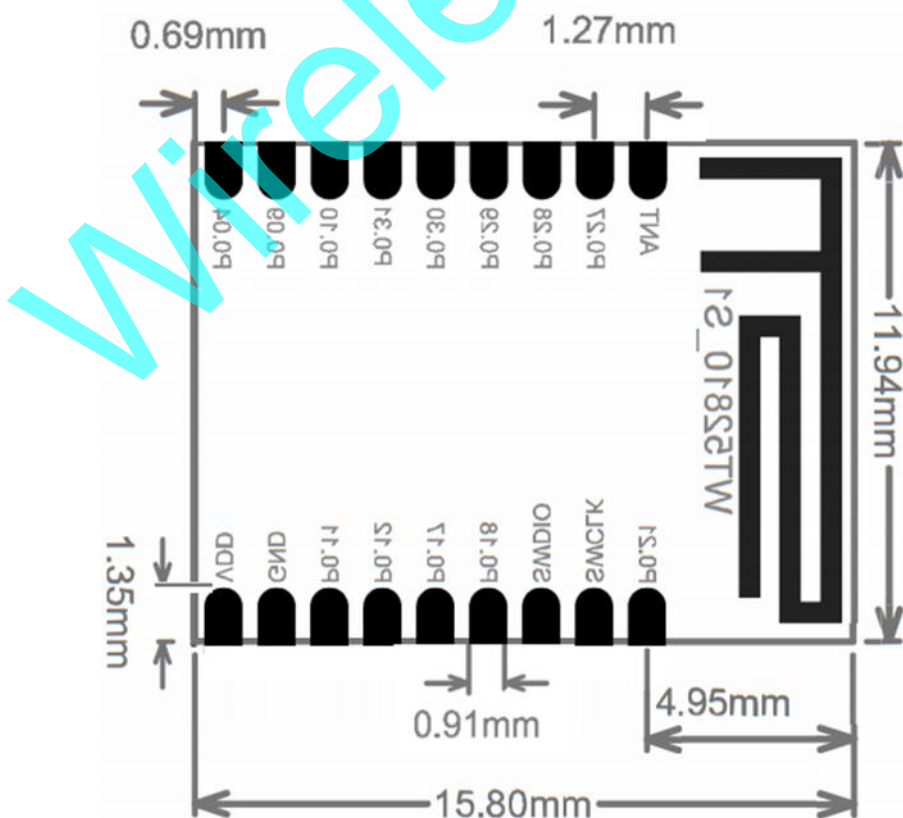
2. Module Pin Definitions



2.1. Pin Description

Chip Pin	Module Pin	Function Description
VDD	VDD	DC 2.0~3.6V
GND	GND	Ground
P0.11	RXD	UART--RX
P0.12	TXD	UART--TX
P0.04	ENABLE	Serial port enabling pin, pull it up to activate the module serial port functions, pull it down to close the serial port, the module power consumption is reduced
P0.09	Led status	Bluetooth status indicator, broadcast status flashes, connection status is always on
P0.10	Wake up	Module wake-up pin, wake up on rising edge after module sleep
SWDCLK	SWDCLK	Debug clock signal input
SWDIO	SWDIO	Debug Interface

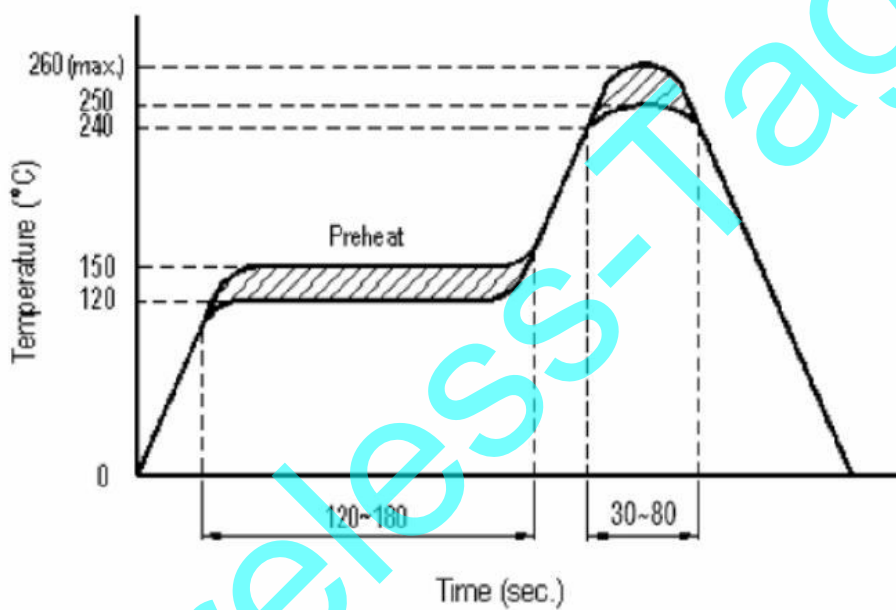
2.2. Size of module





Tolerance: ±0.1mm

2.3. Reflow Soldering Curve



3. Module Power Consumption Description

To control the power consumption of the module, when the serial port enable pin (enable) is at a low level, the UART of the module is closed; At this time, the serial port data and AT commands cannot be sent normally, but the serial port can receive the Bluetooth transparent data normally. After the data transparent transmission is completed, the UART will continue to remain active for 5s to provide the user MCU with a response time (if necessary), and then the serial port will automatically close. When the serial port enable pin (enable) is at a high level, the module UART is in an active state, and the serial port function can be used normally. The following table shows the module power consumption corresponding to different UART states in each broadcast interval.

Bluetooth advertising interval (ms)	UART state	Average power consumption (uA)
50	On	760
	Off	165
100	On	650

	Off	128
200	On	560
	Off	100
500	On	525
	Off	85
1000	On	520
	Off	82

4. Serial Transparent Transmission Protocol Instructions

Serial transparent transmission means that the module connects with the user MCU through a general-purpose serial port to establish two-way communication between the user MCU and the mobile device. The module can obtain up to 244 bytes of transmission data from the serial port at a time. The mobile device can send a data packet with a maximum of 244 bytes to the module each time. After the module receives the data packet, it will be forwarded and transmitted to the MCU serial port receiver in turn. Users can modify the basic Bluetooth parameters of the module through serial AT commands. For details, see *Serial AT Commands Description*.

- 4-1. Serial hardware protocol: default 115200bps, 8, no parity bit, 1 stop bit;
- 4-2. In order to save volume, the module adopts TXD/RXD two-wire serial port mode, with no data flow control added. It is not recommended to set the serial rate too high, in case of packet loss or error code. 115200bps is recommended or other baud rates no higher than 250000bps;
- 4-3. The default Bluetooth connection interval of the module is 20ms. If you need to save power consumption and adopt low-speed forwarding mode, you can adjust the Bluetooth connection interval through AT commands. The longest Bluetooth connection interval is 2000ms.

5. Serial AT Commands Description

The module will automatically identify and distinguish serial data. The data packet with the beginning of AT+ will be defaulted as AT command and parsed, and the processing result will be returned. Therefore, the data in transparent transmission mode cannot start with AT+ character.

5.1. Test Command

Command format: AT+TEST\r\n
 Return value: OK\r\n
 Content: Send AT test commands, returning value OK indicates module AT commands tested successfully.

5.2. Query Module Version Command

Command format: AT+VERSION?\r\n

Return value: 0.0.1\r\n

Content: Query the current version of the module and return the current version number 0.0.1 of the module.

5.3. MAC Address Operation Command

Command format: AT+ADDR?\r\n

Return value: xxxxxxxxxxx\r\n

Content: Return the current MAC address of the module: xxxxxxxxxxx

Command format: AT+ADDR=123456789ABC\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: Modify the MAC address of the module as: 123456789ABC. Return value OK means that the MAC address has been reset successfully.

Note: The MAC address of the module cannot be modified when the Bluetooth module is connected. It needs to be modified after disconnecting the module. After the modification is successful, the module will automatically restart and broadcast with the new MAC address.

5.4. Serial Communication Baud Rate Operation Command

Command format: AT+BAUD=115200\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: Modify the serial port baud rate of the module communication to 115200bps. Return value OK indicates that the modification has been successful, and return value ERR indicates that the new serial port baud rate has failed to be modified because the set value is not commonly used serial port baud rate value. The current serial port baud rates supported by the module:

1200/2400/4800/9600/14400/19200/28800/38400/57600/76800/115200/230400/250000

5.5. Device Name Operation Command

Command format: AT+NAME=WT52810-S1\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: Rename the module as: WT52810-S1, the length of the name does not exceed 22 bytes; Return value OK indicates that the module has been renamed successfully; Return ERR indicates renaming failed. The reason is that the length has exceeded the length limit or Bluetooth has been connected. It needs to be modified after disconnecting.

Command format: AT+NAME?\r\n

Return value: WT52810-S1\r\n

Content: Query the current device name of the module, and the serial port returns the current device name: WT52810-S1.

Note: You can not modify the device name under the state of Bluetooth being connected. It needs to disconnect the Bluetooth connection and modify, then the module will broadcast with a new device name.

5.6. Module Transmit Power Setting Command

Command format: AT+RFPM=4\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: Reset the transmit power of the module to 4dbm. If OK is returned, the power reset is successful. If ERR is returned, the power reset failed. The reason is that the set power level is not in the level list.

Command format: AT+RFPM?\r\n

Return value: 4dBm\r\n

Content: Query the current transmit power configuration of the Bluetooth module, the factory default is 4dBm.

Note: The transmit power levels currently supported by the Bluetooth module are as follows:

Power Level	Power Value
0	4dBm
1	3dBm
2	0dBm
3	-4dBm

4	-8dBm
5	-12dBm
6	-16dBm
7	-20dBm

5.7. Reset Module Command

Command format: AT+RESET\r\n

Return value: OK\r\n

Content: Module reset command, if it returns OK, the module starts reset.

5.8. Restore Default Configuration Command

Command format: AT+DEFAULT\r\n

Return value: OK\r\n

Content: Restore the configuration parameters of the module to the default parameters, and the MAC address is also restored to the factory default address. If it return OK, it indicates that the module starts to restore the factory settings.

5.9. Customize Advertising Data Command

Command format: AT+ADD=0123456A\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: Users can add the customized advertising data in Bluetooth advertisement data, such as adding 0123456A data in Bluetooth advertisement data. If it returns OK, it indicates adding succeeded; If it returns ERR, it indicates adding failed, for the reason that data length may exceed the maximum limit 22bytes. You cannot modify it under the Bluetooth connecting state.

5.10. Customize Product Serial Number command

Command format: AT+PID=ABCD\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: Customize product serial number in the broadcast data. If it returns OK, it means the custom adding is successful; if it returns ERR, it means the adding fails, the possible reason of which is that the custom data exceeds the range of serial number requirements. The module serial number supports 0000—FFFF, and the

default is 0x0000. In the Bluetooth connection state, it cannot be modified and the Bluetooth connection needs to be disconnected.

5.11. Bluetooth Advertising Interval Setting Command

Command format: AT+ADP=500\r\n

Return value: OK\r\n

ERR:CODE\r\n

Content: The effective time interval of Bluetooth advertising is set to 45ms-4000ms, and the factory default is 45ms. Reset the advertising interval of Bluetooth to 500ms. If OK is returned, the reset is successful, and ERR indicates reset failure. In the Bluetooth connection state, you cannot modify it. You need to disconnect the Bluetooth connection.

5.12. Bluetooth Connection Interval Reset Command

Command format: AT+CIT=300\r\n

Return value: OK\r\n

ERR\r\n

Content: The effective Bluetooth connection interval is 10ms—2000ms, and the factory default is 20ms. Reset the Bluetooth connection interval to 300ms. Return OK to indicate successful reset and ERR to indicate reset failure.

Note: The module does not support modifying connection interval during the dynamic process. The modification will take effect after restarting the module. In the Bluetooth connecting state, it cannot be modified and the Bluetooth connection needs to be disconnected. The shorter the Bluetooth connection interval, the faster the Bluetooth communication rate and the higher the power consumption.

5.13. Low-power Sleep Mode Setting Command

Command format: AT+SLEEP\r\n

Return value: OK\r\n

Content: When the module is in the idle state, in order to reduce the power consumption, the module can be set to enter the sleep mode by the sleep command. At this time, the Bluetooth and serial port functions are turned off, and the power consumption is reduced. The module can be awakened by external enabling. The power consumption in sleep mode is 0.3uA.

5.14. Firmware Upgrade Command

Command format: AT+DFU\r\n

Return value: OK\r\n

Content: After returning OK, the module enters the firmware upgrade mode. In this mode, users can download the latest firmware patch released by our company to upgrade the module firmware. After the update is completed, the new firmware will automatically run.

6. BLE Protocol Description (APP Interface)

[Service UUID: 0x2214]

Eigenvalue	Attribute
0x2215	Write Without Response

[Service UUID: 0x2214]

Eigenvalue	Attribute
0x2216	Notify

7. Error CODE Description

Err CODE	Description	Reason
1	Invalid input	
2	Unsupported command	
3	Parsing commands is not allowed in the current state	
4	The command is not allowed in the current state	Some commands are not allowed in the Bluetooth connection status. You need to disconnect the Bluetooth before operating.
5	No action can be performed for this command	
6	Invalid command parameter	
7	Hardware error	
8	Command process timed out	
9	Other errors	

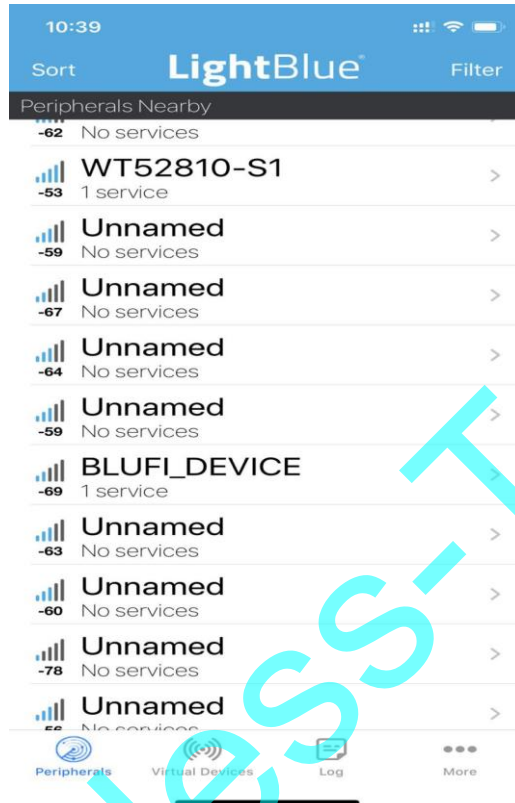
8. Bluetooth Module Application Examples

1. BLE-DK and WT-Demo-Kit are auxiliary tools designed by Wireless-Tag for the simple application of its nRF51822 and nRF52832 series of BLE transparent transmission modules. The Bluetooth module can be easily connected to PC through the DEMO-Kit. The module can be configured with AT commands and some basic parameters of the Bluetooth module can be reset through the serial port debugging tool on the PC side. As a slave device, the Bluetooth module can be searched and paired by the Bluetooth master of the mobile device APP. After pairing, data can be input to the Bluetooth module through the serial debugging tool. The Bluetooth module transmits the data to the Bluetooth master terminal, and at the same time, it transmits the data returned by Bluetooth master terminal to serial debugging tool. Developers can conveniently and simply view the contents of data transmission between the Bluetooth module and the Bluetooth master terminal.
2. Connection method of DEMO-Kit and Bluetooth module: The interface of DEMO-Kit tool is the currently popular XBee module interface mode. The Bluetooth module and Kit tool are connected through the XBee adapter board, and the module is welded on the corresponding XBee adapter board, and then the adapter board is inserted into the Kit tool.
3. The DEMO-Kit tool adopts FTDI's USB to UART IC, so before it can be used on PC, you need to install the driver of the FTDI chip first. After the driver is installed successfully, connect the DEMO-Kit to PC by a USB cable. At this time, the serial port label of Demo Kit tool can be recognized by PC terminal;
4. Open the serial port debugging tool on PC, select the virtual COM port number mapped by the FTDI chip. Set the serial port parameters as: baud rate 115200, the data format is 8 data bits, no parity bit and 1 stop bit. Then open the serial port, and read and modify the current Bluetooth module settings through AT command.
5. Use the mobile device APP of Bluetooth master to query Bluetooth module slave device.

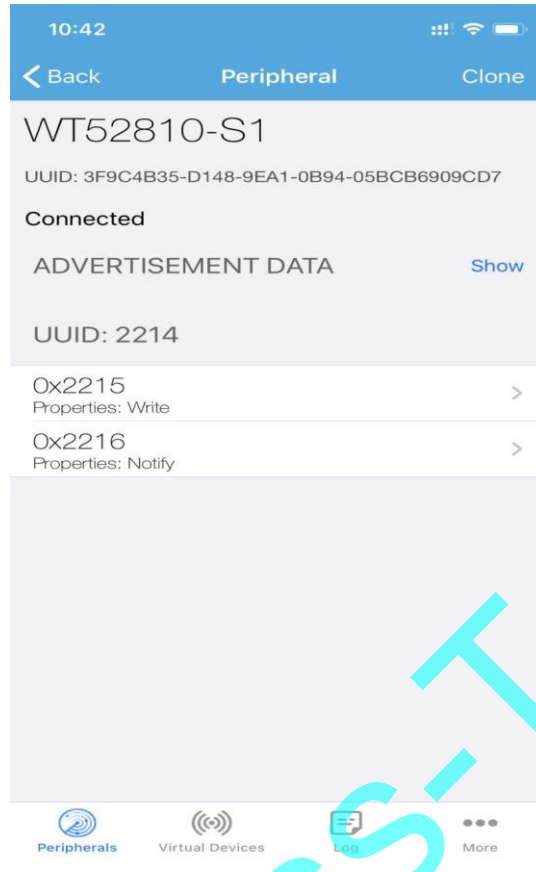
The recognized Bluetooth test software LightBlue can be used in the IOS system, and the Bluetooth serial test APP software provided by our company can be used in the Android system. Search for the device name of the Bluetooth module, and then click to establish Bluetooth connection with it and the pairing is successful. Send data to the Bluetooth module through the APP tool to establish a data communication connection between the mobile terminal and the Bluetooth module.

A. In IOS system (only iPhone 4S and above mobile phones support Bluetooth 4.0), use

LightBlue to test the module connection performance. First turn on the iPhone Bluetooth and run the LightBlue program. When LightBlue is running, it will automatically search for the slave. After the slave is searched, the slave list will be displayed, and it will contain the main information, UUID of Services, transmit power, device name, etc.



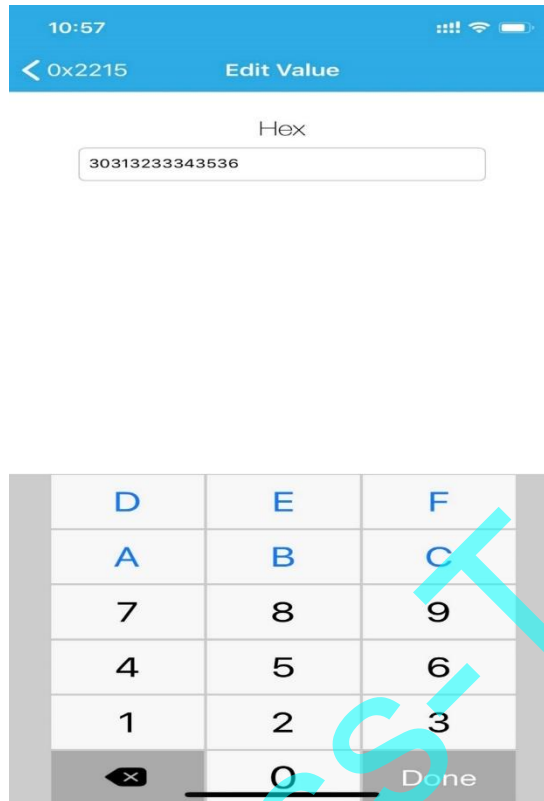
Click on the slave that needs to be connected, the iPhone will connect to the slave, and then the program will automatically search for all services of the slave, as shown below.



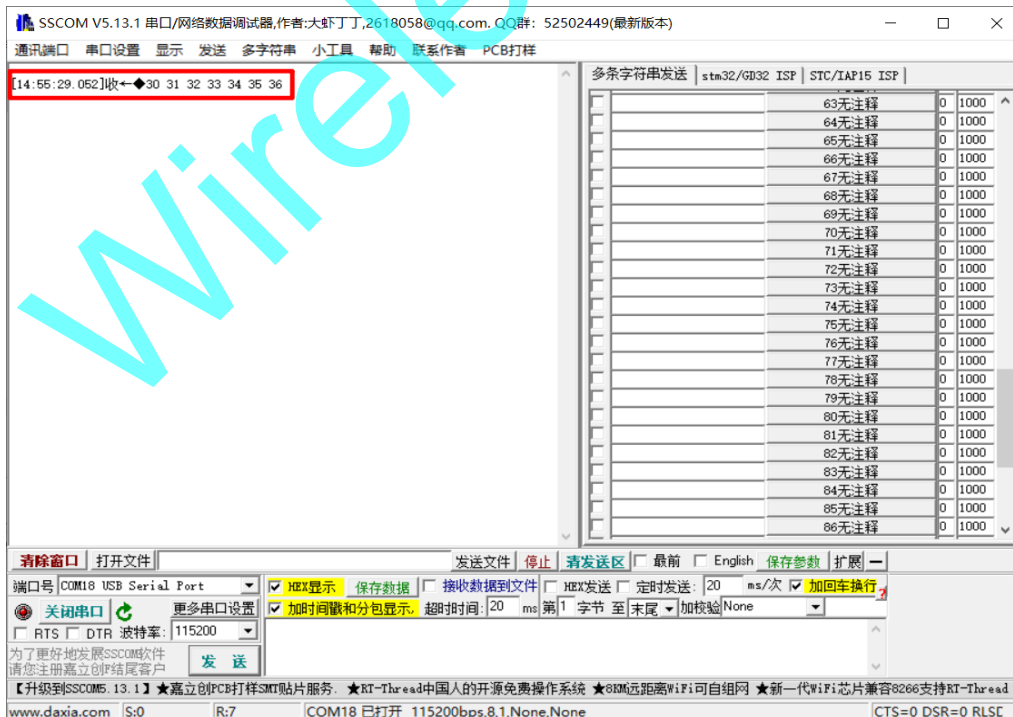
Click the corresponding Service to enter the characteristics contained in the Service, as shown in the figure below.



Click *Writer new value* to enter the Characteristic communication interface and write the input value, as shown in the figure below:



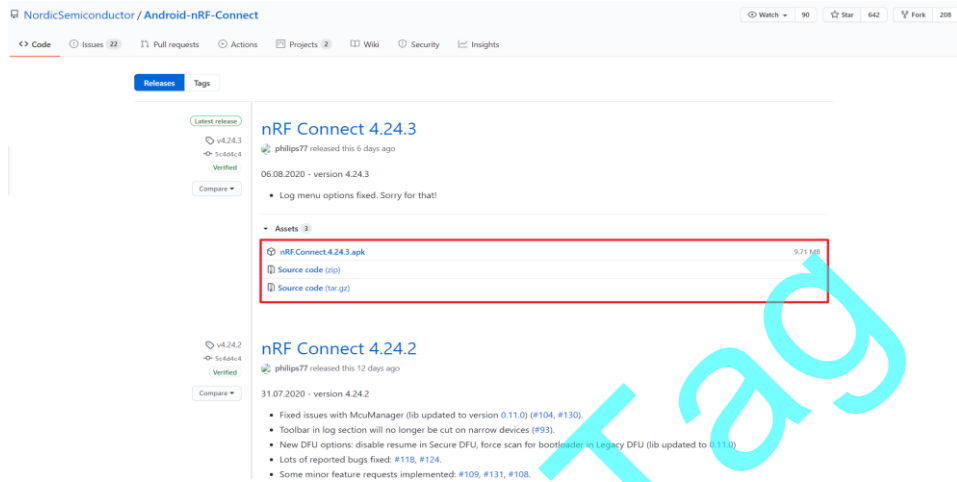
After the module receives the data sent by LightBlue, it pushes the received data to the serial port through DEMO-KIT, and displays it in the serial port debugging tool on the PC:



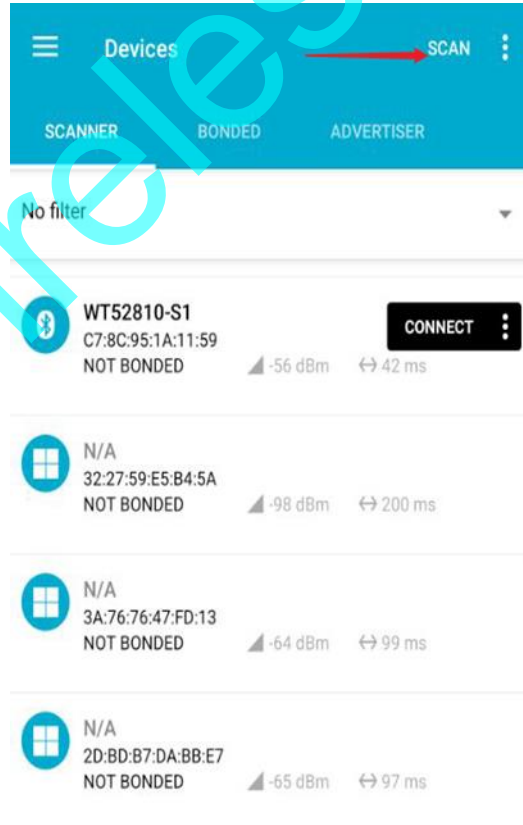
B. In the Android system (Android 4.3 and above supports Bluetooth 4.0; Android 6.0

and above, you need to open the GPS service to search for Bluetooth), nrfconnect can be used as an APP debugging tool, and the APK and source code can be obtained by:

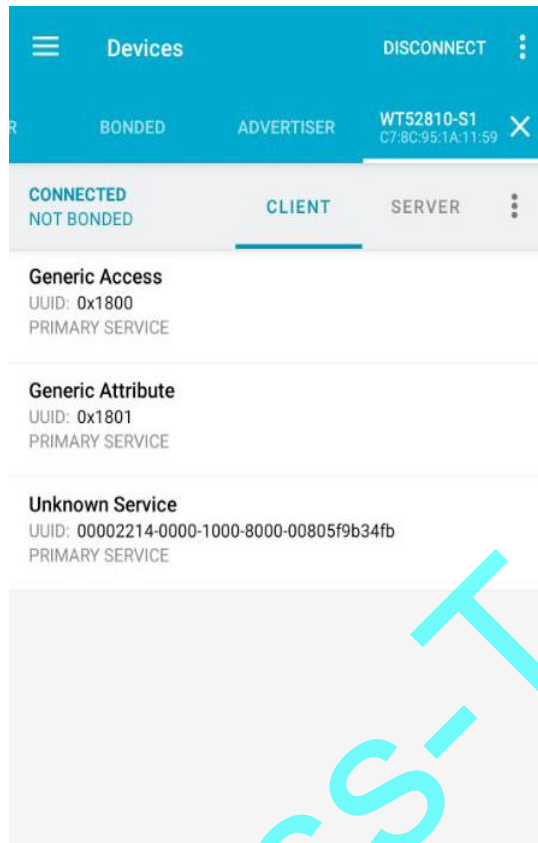
https://github.com/NordicSemiconductor/Android-nRF-Connect/releases?tdsourcetag=s_pctim_aiomsg



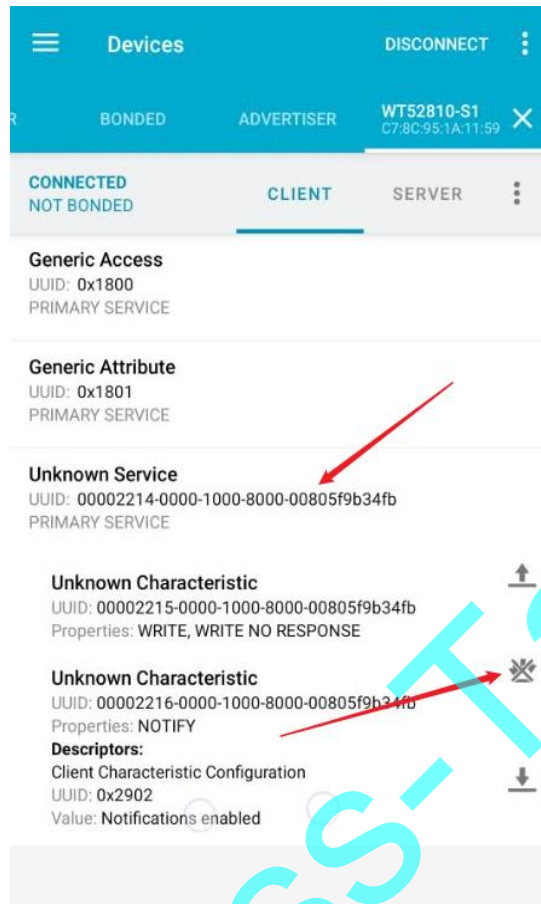
Search for Bluetooth device in the APP:



Click CONNECT to connect to the Bluetooth device:



After the connection is successful, expand the 0x2214 service and open the NOTIFY channel:



Then, the NOTIFY channel can receive the serial port data, or send data to the serial port through the WRITE, WRITE NO RESPONSE channel.

- Through our Bluetooth DEMO-KIT tool and serial debugging tool APP, it provides great convenience for customers to quickly develop and apply Bluetooth devices, which is beneficial to shorten the development cycle of Bluetooth projects.

9. Appendix

For more technical support, please email to: technical@wireless-tag.com

Wireless-Tag