

BT201 Module User Manual

Document Status: [] Draft [v] Officially Released [] Under Revision	Document Identification:	Publicly Available
	Current Version:	V2.4
	Author:	Qingyue Electronics
	Completion Date:	2020-04-17

BT201 Bluetooth Module

Bluetooth Audio

Lossless
decoding
playback

BLE/SPP transparent transmission

Serial port AT commands

TF card USB drive Headphone

40mm

Function
keys

60mm

Serial port interface
RX connects to MCU's TX
VCC(5V)

Microphone

Speaker

Lithium battery power supply

Power supply port

Contents

1.OVERVIEW	5
1.1 INTRODUCTION	5
1.2 FUNCTIONS AND FEATURES OVERVIEW --- SUPPORTS AT SERIAL COMMAND CONTROL	5
1.3 EXPLANATION OF PROFESSIONAL TERMS	6
1.4 PRODUCT APPLICATION SCENARIOS	6
1.5 BRIEF DESCRIPTION OF MODULE FUNCTIONS [NO NEED TO READ IF USING SERIAL PORT CONTROL]	7
1.6 MODULE QUICK START INSTRUCTIONS	8
2.SOLUTION DESCRIPTION	9
2.1 PARAMETER DESCRIPTION	9
3.SERIAL COMMUNICATION PROTOCOL	11
3.1 COMMUNICATION FORMAT	11
3.2 COMMUNICATION COMMAND	12
3.2.1 Control Commands Related to General Functions	12
3.2.2 Control Commands Related to Music	13
3.2.3 Control Commands Related to Bluetooth	14
3.2.4 Query Commands Related to General Functions	15
3.2.5 Query Commands Related to Music	16
3.2.6 Query Commands Related to Bluetooth	17
4.DETAILED DESCRIPTION OF SERIAL PORT COMMANDS -- GENERAL SECTION	19
4.1 DATA ACTIVELY RETURNED BY THE MODULE	19
4.1.1 Data Returned by the Chip on Power-On [QA][QT][QM]	19
4.1.2 Acknowledgment Returned by the Chip on Successful Receipt of Serial Port Commands [OK]	19
4.1.3 Return of the Chip's Error Information [ER]	19
4.2 GENERAL SECTION -- DETAILED EXPLANATION OF CONTRO COMMANDS	20
4.2.1 Specify the playback volume of the chip [CA][CB][CC][CD][CE][CF]	20
4.2.2 Setting the Chip's Baud Rate [CT]	21
4.2.3 Setting the Chip's Operating Mode [CM]	21
4.2.4 Detailed Explanation of Setting Module Mute and DAC On/Off [CU][CS]	22
4.2.5 Explanation of Setting the Chip's Auto-Enter Mode on Power-Up [CP]	22
4.2.6 Setting the Chip's Automatic Data Return Function On/Off [CR]	22
4.2.7 Setting TF Card or U Disk to Play Once and Stop or Loop Sequentially [CJ]	23
4.2.8 Setting Whether the TF Card or U Disk Recording Function is Enabled [RE]	23
4.2.9 Setting Whether to Enable Prompt Sounds [CN]	23
4.2.10 Setting the Audio EQ [CQ] -- Not Supported Yet	23
4.2.11 Setting Bluetooth Auto-Switch to Background [CK]	24
4.2.12 Setting Whether Bluetooth Runs in the Background [CG]	24

4.2.13 Setting Whether Key Functions Are Disabled or Enabled [C1][C2][C3][C4]	24
4.3 GENERAL SECTION -- DETAILED EXPLANATION OF QUERY COMMANDS	25
4.3.1 Explanation of Common Status Query Returns [QA][QT][QN][QK]	25
4.3.2 Querying Operating Mode and Explanation of Returns [QM]	25
5.DETAILED DESCRIPTION OF SERIAL PORT COMMANDS -- MUSIC SECTION	27
5.1 MUSIC-RELATED -- DETAILED EXPLANATION OF CONTROL COMMANDS	27
5.1.1 Data Returned After U Disk or TF Card Initialization is Complete [M1][M2][MT][MK][MF]	27
5.1.2 Information Returned After U Disk or TF Card Playback is Complete [MV][MD][MO]	28
5.1.3 Information Returned for TF Card or U Disk Insertion/Removal [MU]	28
5.1.4 TF Card or U Disk Commands -- Common Basic Functions [AA]	28
5.1.5 TF Card or U Disk Commands -- Track Number Playback [AB]	29
5.1.6 TF Card or U Disk Commands -- Loop Playback of a Specified Folder Path [AF]	29
5.1.7 TF Card or U Disk Commands -- Play a Specified File Path Once [AJ]	30
5.1.8 TF Card or U Disk Commands -- Specify Playback Mode: Single Track, All, Random [AC]	30
5.1.9 TF Card or U Disk -- Recording Function [RC]	30
5.1.10 TF Card or U Disk Command -- Read Data from Specified File Path (TXT Reading) [AR]	32
5.1.11 Serial Port Reading of TXT and Other File Formats from TF Card or U Disk [Advanced Function]	33
5.1.12 TF Card or U Disk Command—Delete File at Specified Path [AL]	34
5.1.13 Recording Function—TF Card or U Disk—Specify Folder and Filename for Recording [RP][RB][RV]	34
5.2 MUSIC-RELATED -- DETAILED EXPLANATION OF QUERY COMMANDS	35
5.2.1 Querying the Currently Playing File Name on TF Card or U Disk [MF]	36
5.2.2 TF or U Disk Query for Current File Playback Time Processing [MT][MK]	37
6.DETAILED DESCRIPTION OF SERIAL PORT COMMANDS -- BLUETOOTH SECTION	38
6.1 BLUETOOTH-RELATED -- DETAILED EXPLANATION OF CONTROL COMMANDS	38
6.1.1 Setting Bluetooth Name and Password [BD][BE][BM]	38
6.1.2 Setting Bluetooth Protocol Functions [B1][B2][B3]	38
6.1.3 Setting Bluetooth BLE and EDR Enable/Disable [B4][B5]	39
6.1.4 Specifying a Phone Number to Call [BT]	39
6.1.5 Bluetooth Audio-Related Control Commands [BA]	39
6.1.6 Bluetooth MAC Address Setting -- EDR -- BLE [BS]	40
6.2 BLUETOOTH-RELATED -- DETAILED EXPLANATION OF QUERY COMMANDS	40
6.2.1 Bluetooth Current Status Return -- EDR Simple Status [TS]	40
6.2.2 Bluetooth Current Status Return -- BLE Simple Status [TL]	40
6.2.3 Bluetooth Incoming Call Number Return [TT]	41
7.DETAILED DESCRIPTION OF BLUETOOTH TRANSPARENT TRANSMISSION—BLE	42
7.1 BLE TRANSPARENT TRANSMISSION DESCRIPTION	42
7.2 BLE UUID DESCRIPTION	42

7.3 BLE EFFECT DEMONSTRATION EXPLANATION	42
7.4 BLE TESTING INSTRUCTIONS	42
7.5 TESTING EXPLANATION FOR CONTROLLING THE BLUETOOTH CHIP VIA BLE FROM A MOBILE DEVICE	43
7.6 EXPLANATION OF MODIFYING BLE UUID VIA AT COMMANDS	44
7.7 BLE DATA TRANSPARENT TRANSMISSION PACKET TIME INTERVALS	44
7.8 EXPLANATION OF MODIFYING BLE BROADCAST PACKET DATA [UR][TR]	45
8.DETAILED DESCRIPTION OF BLUETOOTH TRANSPARENT TRANSMISSION—SPP	47
8.1 SPP TRANSPARENT TRANSMISSION DESCRIPTION	47
8.2 SPP EFFECT DEMONSTRATION EXPLANATION	47
8.3 SPP TRANSPARENT TRANSMISSION TESTING INSTRUCTIONS	47
9. MODULE FIRMWARE UPDATE AND SERIAL PORT TESTING EXPLANATION	48
9.1 EXPLANATION OF MODULE FIRMWARE UPDATE	48
9.2 POTENTIAL DOUBTS OR ISSUES DURING MODULE UPDATE -- UPDATE FAILURE	48
9.3 EXPLANATION OF THE MODULE'S SERIAL PORT DEBUGGING ASSISTANT	49
10.FREQUENTLY ASKED QUESTIONS (FAQS)	51
11.KT1025A BT201 AND KT1025B BT201 PACKAGING DIMENSIONS AND PARAMETER EXPLANATION	53
12.REFERENCE PROGRAM EXAMPLES	55
13.EXPLANATION OF MODIFYING PROMPT SOUNDS	57
14.BQB CERTIFICATION AND FCC TESTING EXPLANATION	58
14.1 BLUETOOTH BQB CERTIFICATION EXPLANATION	58
14.2 BLUETOOTH FCC FIXED-FREQUENCY TESTING EXPLANATION	58
15.DETAILED STATEMENTS	59
16.VERSION HISTORY	60
17.QINGYUE BLUETOOTH SERIES PRODUCT RECOMMENDATION	62
17.1 BT201 BLUETOOTH DUAL-MODE [AUDIO + DATA TRANSMISSION BLE + SPP] SOLUTION	63
17.3 BT301 BLUETOOTH [AUDIO + DATA TRANSMISSION BLE + SPI FLASH] SOLUTION	64
17.4 BT401 BLUETOOTH AUDIO [SUPPORTS ALL FUNCTIONS] SOLUTION	65

1.Overview

1.1 Introduction

The BT201 Bluetooth audio module is a compact, dual-mode Bluetooth 5.0 module, designed for wireless audio and data applications. Key functions and features include:

- ✓ **Audio Capabilities:** Supports stereo audio output with a built-in 3W amplifier for direct speaker connection, enabling high-quality wireless audio streaming.
- ✓ **Multi-Protocol Support:** Compatible with Bluetooth protocols such as A2DP, AVRCP, and HFP, ensuring seamless audio playback, control, and hands-free calling.
- ✓ **Versatile Input Options:** Integrates Bluetooth audio, USB flash drive, TF card, and lossless music playback, making it suitable for diverse audio sources.
- ✓ **BLE Data Transmission:** Supports Bluetooth Low Energy for efficient data transfer alongside audio streaming.
- ✓ **Serial Port Control:** Allows configuration and control via AT commands, enhancing customization for developers.
- ✓ **Ease of Use:** Features simple pairing with the device name "BT201-AUDIO," ideal for DIY audio projects like wireless speakers or headsets.

Applications:

Perfect for wireless audio systems, IoT devices, and hands-free communication, with support for audio recording and playback. Such as:

- ✓ **Bluetooth Audio Products,** E.g., Bluetooth speakers, Bluetooth headsets, car Bluetooth systems, etc.
- ✓ **Bluetooth Data Transmission Products,** E.g., smart door locks, car OBD diagnostics, smart carts, printers, medical device data collection.
- ✓ **Bluetooth Data Transmission + Audio Products,** E.g., Bluetooth music lights, Bluetooth broadcasting, electronic keyboards, and other musical instruments.

Fundamentally speaking, the BT201 Module is a minimum system for Bluetooth audio decode and playback, BLE or SPP data transmission based on the KT1025A Bluetooth chip. It can also serve as a complete test board for the KT1025A chip. Therefore, relevant technical materials, schematic diagrams, manuals, and even AT instructions can all refer to the materials of the KT1025A BT201 solution.

BT201 module has a very convenient firmware upgrade interface. For details, refer to Section 9. Additionally, there are many quick application documents available.

1.2 Functions and Features Overview --- Supports AT Serial Command Control

Function	Function Description
----------	----------------------

General Functions	1. 16-bit Stereo DAC with headphone amplifier, SNR >= 95dB
	2. Crystal-free load capacitance, chip automatically generates Bluetooth MAC address, no need for sequential programming
Music Functions	1. Supports lossless full decoding of MP3, WAV, WMA, FLAC, AAC, APE formats
	2. Supports up to 128GB U-disk and TF card, with breakpoint memory and track memory functions
Bluetooth Function Features	1. Supports Bluetooth audio transmission to connect to a phone for music playback, supports play/pause, previous/next track switching
	2. Supports Bluetooth call functions, user can enable/disable, supports answer, hang up, redial, reject, and other functions
	3. Bluetooth 5.0 version, supports HFP/A2DP/AVRCP/HSP/GAVDP/IOP/SPP/BLE, range about 10 meters
	4. Class 2, 4dBm, frequency range 2.4G–2.480G
	5. Supports BLE transparent transmission function, standalone connection to “BT201-BLE”; supports SPP transparent transmission function

1.3 Explanation of Professional Terms

Function	Remarks
General Functions	Refers to functions available in all modes, such as volume adjustment, mute, mode switching, reset, etc.
Music, MUSIC, Device	Refers to playback functions using TF card or U-disk.
Bluetooth Mode	Our Bluetooth supports both audio and simultaneous BLE and SPP data transmission.
BLE and SPP	Standard Bluetooth communication protocols, independent of Bluetooth audio.
transparent transmission	Means whatever the phone sends to Bluetooth, Bluetooth sends out via serial port, and vice versa.
Serial Port	Refers to the external interface of the Bluetooth chip, i.e., the UART port.
Background	Refers to when music is playing, Bluetooth resources are not released, and Bluetooth data communication can still occur.

1.4 Product Application Scenarios

1. Bluetooth Audio Products	E.g., Bluetooth speakers, Bluetooth headsets, car Bluetooth systems, etc.
2. Bluetooth Data Transmission Products	E.g., smart door locks, car OBD diagnostics, smart carts, printers, medical device data collection.
3. Bluetooth Data Transmission + Audio Products	E.g., Bluetooth music lights, Bluetooth broadcasting, electronic keyboards, and other musical instruments.

Note: **This solution is not suitable for applications requiring ultra-low power consumption.**

1.5 Brief Description of Module Functions [No need to read if using serial port control]

Overall Module Functionality	
Bluetooth Playback Function	Supports connection with a smartphone to play music, with a range of approximately 10 meters.
U Disk and TF Card Playback Function	<ol style="list-style-type: none"> 1. Supports playback of MP3, WAV, WMA, FLAC, APE format music files, placed in the root directory of the device. 2. Automatically plays when a U disk or TF card is inserted, with power-off memory (resumes playback from the last position). 3. Playback order is based on the sequence in which files were copied to the U disk or TF card.
Infrared Remote Control Function	Not supported by default
Bluetooth Call Function	Supported by default
Radio Function(FM)	Not supported. For radio functionality, refer to our BT301 solution.
AUX Line Input Function	Not supported. For AUX input, refer to our BT301 solution.
Headphone Output Function	The module includes a headphone jack, supporting headphone insertion.
External Speaker Function	Supports connection to an external 4-ohm, 3W or lower speaker.
<p>Remarks:</p> <ol style="list-style-type: none"> 1. The module automatically enters Bluetooth mode on power-up, waiting for a smartphone connection. To play from a U disk or TF card, insert or reinsert the device. 2. This design speeds up boot time and keeps Bluetooth initialized in the background. 3. Bluetooth name for audio: "BT201-AUDIO" [connect to this for music playback]. 4. Bluetooth name for BLE: "BT201-BLE" [connect to this for BLE data transmission]. 5. If a U disk or TF card is inserted, it automatically plays the contents. 6. If both U disk and TF card are inserted, use the Mode button to switch between them. 7. The module has an indicator LED: <ul style="list-style-type: none"> Fast flashing: In Bluetooth mode (waiting for connection). Constantly on: Bluetooth connected. Slow flashing: Playing music. 	

Button Function Description			
Button	Short Press	Long Press	Long Press (Hold)
PREV/V-	Previous track	-	Volume down
NEXT/V+	Next track	-	Volume up
MODE -- Switch Mode	Switch between Bluetooth,	-	-

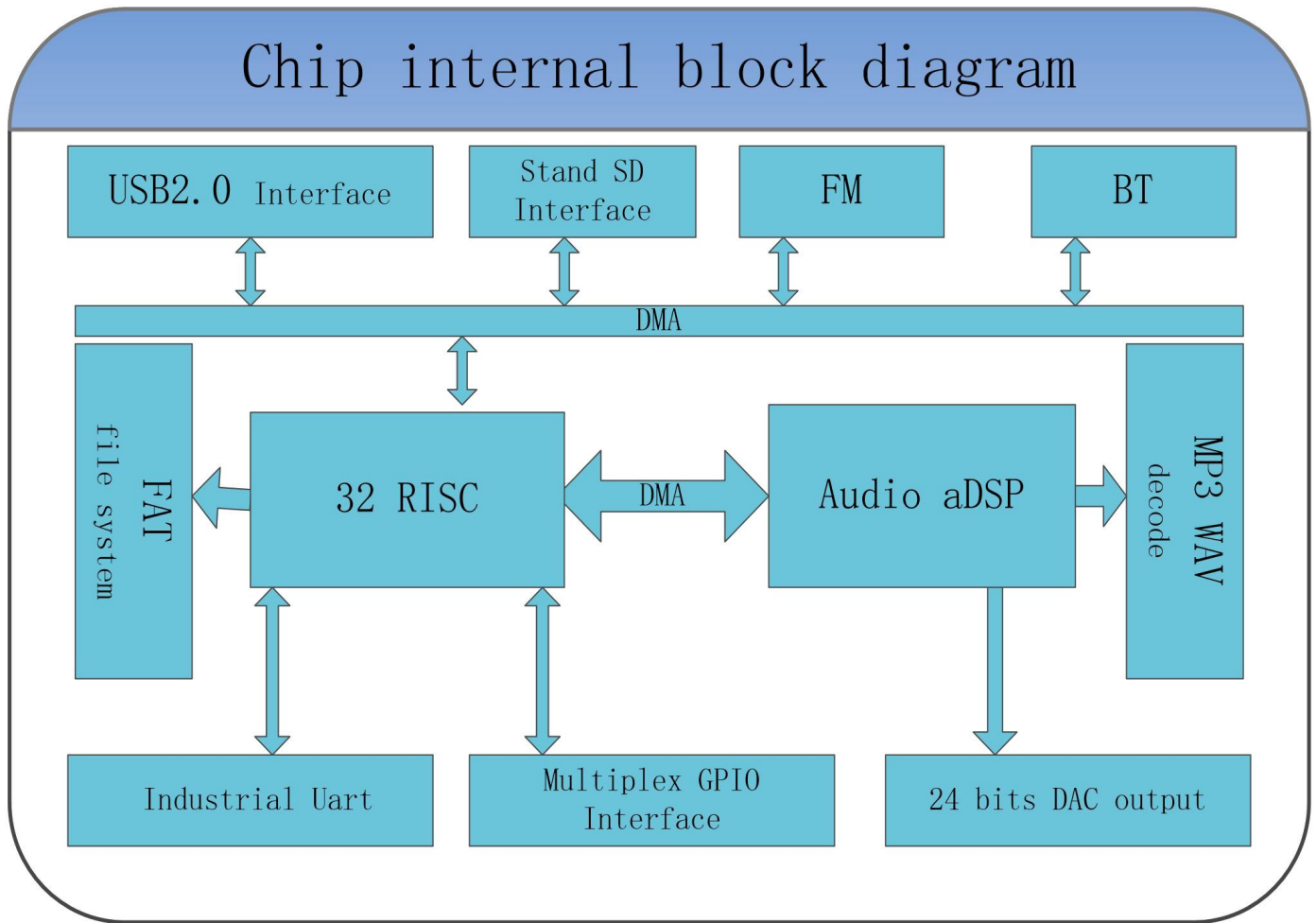
	U disk, and TF card		
PP -- Play/Pause	Play/Pause	-	-

1.6 Module Quick Start Instructions

If you only need basic functions, just a simple Bluetooth player, then directly ignore the extra functions. Our design is fully compatible.

Video Link: <https://v.qq.com/x/page/h0767kjh6z.html>

2.Solution Description



This System-on-Chip (SoC) solution integrates:

- ✓ A 32-bit RISC MCU
- ✓ A dedicated audio DSP (aDSP) for hardware-based decoding

Key Advantages:

- ✓ Hardware Decoding ensures system stability and audio fidelity
- ✓ Compact package size is more suitable for embedding in other products

2.1 Parameter Description

Name	Parameter
MP3 File Format	1. Supports all bitrates (11172-3 & ISO13813-3 Layer3 audio decoding)
	2. Sampling rates (kHz): 8/11.025/12/16/22.05/24/32/44.1/48
	3. Sound effects: Normal, Jazz, Classic, Pop, Rock
Bluetooth Parameters	1. Bluetooth version: 5.0
	2. Dual-mode Bluetooth, supports audio and data transmission

	3. Supports calls, audio, BLE, SPP
USB Interface	USB 2.0 standard, supports U-disk playback, supports connection to a computer as a sound card, supports firmware updates
UART Interface	3.3V TTL level, baud rate configurable, recommended for PC testing: CH340G—USB to TTL small board
Input Voltage	3.3V–5V [best with a 7805 regulator followed by a diode in series], Be sure to pay attention to the inrush current during power-on, especially in applications with external adapter voltage reduction.
Rated Current	20mA [static]
Low Power Consumption Current	<200μA [Recommended to control power using a MOSFET while a low-power consumption is needed].
Module amplifier power	Drive a 4-ohm 3W speaker, using the HK9108 chip [5W digital Class D amplifier].
DAC Characteristics and Drive	The Main chip's DAC can only drive headphones.[32ohms]
Operating Temp.	-40°C to +80°C
Humidity	5% ~ 95% RH
Model	[BT201 Module Stereo]

3.Serial Communication Protocol

The AT serial command, commonly used in control applications, has been optimized and customized by us, significantly reducing the difficulty for users. Please strictly follow the command format we provide for operation.

3.1 Communication Format

Supports asynchronous serial communication mode, receiving commands from the host computer via the serial port.

Communication Standard: **115200 bps** — Users can adjust this setting via serial commands.

For details, see Section 4.2.2.

Data Bits: 8

Stop Bits: 1

Parity Bit: None

Flow Control: None

Note: All command designs follow a specific pattern and are not arbitrarily divided. You can refer to the following to identify the pattern.

Control Command Format: AT+<CMD>[<param>]\r\n — All characters are in text format, not hexadecimal.

Data Feedback Format: <IND>[<param>]\r\n

Data	Detailed Explanation
AT+	This is the control command sent by the host to the BT201, starting with "AT+".
<CMD>	Immediately followed by the <CMD> control, usually consisting of 2 characters.
[<param>]	If there are parameters following the CMD, they are enclosed in [<param>].
\r\n	The command ends with "\r\n", which represents a newline in character format (equivalent to pressing the Enter key on Windows). In hexadecimal, it is represented as 0x0D, 0x0A.
<IND>	<p>1. Data feedback is the information (status and data) sent by the Bluetooth chip back to the host, starting with <IND>.</p> <p>2. This is followed by the parameters returned by the BT401.</p>

Here is a detailed explanation of <CMD>:

Function Category	Command	Remarks
General Command Characteristics	AT+C?	General commands start with AT+C, and the "?" represents the specific detailed function command.
Music Command Characteristics	AT+A?	Music commands start with AT+A, and the "?" represents the specific detailed function command.

Bluetooth Command Characteristics	AT+B?	Bluetooth commands start with AT+B, and the "?" represents the specific detailed function command.
Here is a detailed explanation of <CMD>:		
Example	Command	Remarks
Control Command 1	AT+CB\r\n	Represents Play/Pause
Control Command 2	AT+CA20\r\n	Represents Setting Volume to Level 20
Query Return Result 1	QA+01	For details, see Section 4.4.1. The returned query information always follows the format Qn+xx, where n corresponds to the previous command.
Query Return Result 2	QG+01	For details, see Section 4.2.12.

3.2 Communication Command

Our Communication is Divided into the Following Two Main Categories

- ✓ Control Commands and Reference Commands --- See Sections 3.2.1, 3.2.2, and 3.2.3
- ✓ Query Commands and Reference Commands --- See Sections 3.2.4, 3.2.5, and 3.2.6

3.2.1 Control Commands Related to General Functions

General Section -- Control Commands -- Description		
CMD	Corresponding Function	Detailed Description
AT+CA	Specify Volume	Has parameters , see 4.2.1
AT+CB	Play/Pause	See 4.2.1
AT+CC	Next Track	See 4.2.1
AT+CD	Previous Track	See 4.2.1
AT+CE	Volume Up	See 4.2.1
AT+CF	Volume Down	See 4.2.1
AT+CT	Set Baud Rate	Has parameters , see 4.2.2
AT+CM	Set Mode	Has parameters , see 4.2.3
AT+CU	Set Mute	Has parameters , see 4.2.4
AT+CS	Set DAC High Impedance	Has parameters , see 4.2.4
AT+CZ	Chip Reset	Soft reset of the chip
AT+CW	Chip Restore Factory Settings	Restores factory settings, clears all previously stored parameters
AT+CP	Power-On Mode	Has parameters , see 4.2.5
AT+CR	Automatic Feedback Function	Key chip parameters are automatically fed back, can be disabled here, see 4.2.6
AT+CJ	Single Track Trigger Playback	See 4.2.7
AT+CN	Set Prompt Tone	See 4.2.9
AT+CQ	Set Playback EQ	See 4.2.10

AT+CK	Set Bluetooth Background Switching	See 4.2.11
AT+C1	Disable Button Functions	See 4.2.13
AT+C2	Chip Actively Returns Data	See 4.2.13
AT+C3	Reserved	See 4.2.13
AT+C4	Bluetooth Auto-Reconnect on Power-On	See 4.2.13

General Section -- Control Commands -- Examples	
CMD	Detailed Description
AT+CA30\r\n	Sets volume to level 30
AT+CB\r\n	Pause when playing, play when paused
AT+CC\r\n	Next track
AT+CD\r\n	Previous track
AT+CE\r\n	Volume up, stops increasing at level 30
AT+CF\r\n	Volume down
AT+CM00\r\n	Switches mode, to the next mode, see 4.2.3
AT+CZ\r\n	Chip reset
AT+CW\r\n	Restores chip's default factory parameters
AT+CR00\r\n	Disables automatic feedback]
AT+CJ01\r\n	Sets to single trigger playback

3.2.2 Control Commands Related to Music

Music -- Control Commands -- Description		
CMD	Corresponding Function	Detailed Description
AT+AA	Set Playback Mode	Has parameters, see 5.1.4
AT+AB	Specify Physical Order Playback	See 5.1.5
AT+AC	Specify Playback Mode	See 5.1.6
AT+AD	Specify Playback Device	To be determined
AT+AE	Specify Playback EQ	To be determined -- Not supported
AT+AS	Specify Playback Speed	To be determined -- Not supported
AT+AF	Specify Folder Loop Playback	See 5.1.6
AT+AJ	Specify Folder Filename Playback Once	See 5.1.7
AT+AR	Read data from a specified file -- understood as reading a txt file	See 5.1.10
AT+AL	Delete a file at a specified path	See 5.1.12

Music -- Control Commands -- Examples	
CMD	Detailed Description
AT+AA00\r\n	Stop playback
AT+AB01\r\n	Specify music playback by physical sequence number 1
AT+AC01\r\n	Specify single track loop playback mode
AT+AF/USB_UPDA/*.*???\r\n	Specify loop playback of the “USB_UPDA” folder
AT+AJ/02*/011_11.*???\r\n	Specify one-time playback of the “011_11” file in the “02xxx” folder, then stop
AT+AR/02*/011.*???\r\n	Read and output via UART the contents of files starting with “011” in the “02xxx” folder.
AT+AL/02*/011.*???\r\n	Delete files starting with “011” in the “02xxx” folder.

3.2.3 Control Commands Related to Bluetooth

Bluetooth -- Control Commands -- Description		
CMD	Corresponding Function	Detailed Description
AT+BA	Bluetooth Control Commands	See 6.1.5
AT+BD	Set EDR Bluetooth Name	Here, EDR refers to Bluetooth audio and SPP, has parameters, see 6.1.2
AT+BE	Set EDR Connection Password	Has parameters, see 6.1.3
AT+BM	Set BLE Bluetooth Name	Here, BLE refers to “Low Energy Bluetooth”
AT+BN	Set BLE Connection Password	
AT+BS	Set EDR MAC Address	See 6.1.6. BLE address does not need setting, automatically generated based on EDR
AT+B1	Simple Password Setting	Only 00 or 01, 00 means disable, 01 means enable
AT+B2	Call Setting	Same as above
AT+B3	Bluetooth Audio Setting	Same as above
AT+B4	Control BLE On/Off	See 6.1.3
AT+B5	Control EDR On/Off	
AT+BT	Specify Phone Number Dialing	See 6.1.3
AT+U0	Specify Service UUID	See 7.6
AT+U1	Specify Characteristic 1	
AT+U2	Specify Characteristic 2	
AT+U3	Specify Characteristic 3	

Bluetooth -- Control Commands -- Examples	
CMD	Detailed Description
AT+BAxx\r\n	See 6.1.5
AT+BD1234\r\n	Sets the EDR Bluetooth name to "1234"
AT+BE5432\r\n	Sets the EDR connection password to "5432"
AT+BM2345\r\n	Sets the BLE Bluetooth name to "2345"
AT+B100\r\n	This 00 means 0x00, disables pairing password, so next connection does not require a password
AT+B201\r\n	This 01 means 0x01, enables Bluetooth call function; 00 disables Bluetooth call function
AT+B300\r\n	This 00 means 0x00, disables Bluetooth audio, so even if connected, music cannot be played; 01 enables as above
AT+B400\r\n	This 00 means 0x00, disables BLE; 01 enables BLE, see 6.1.3
AT+B501\r\n	This 01 means 0x01, enables EDR; 00 disables EDR, see 6.1.3
AT+BT10086\r\n	Specifies calling the phone number "10086"
AT+U0F000\r\n	Specifies service UUID as F000
AT+U1F001\r\n	Specifies characteristic 1 as F001
AT+U2F002\r\n	Specifies characteristic 2 as F002
AT+U3F003\r\n	Specifies characteristic 3 as F003

3.2.4 Query Commands Related to General Functions

General Section -- Query Commands -- Description		
CMD	Corresponding Function	Detailed Description
AT+QA	Query Volume	See 4.3.1
AT+QT	Query Baud Rate	See 4.3.2
AT+QM	Query Operating Mode	See 4.3.3 [0: Bluetooth] [1: MP3. To know the device, query device with AT+MD]
AT+QN	Query Prompt Tone Status	See 4.3.4

General Section -- Query Commands -- Examples	
CMD	Detailed Description
AT+QA\r\n	Chip returns "QA+30\r\n", indicating the volume returned to the host is level 30
AT+QT\r\n	Chip returns "QT+03\r\n", indicating the baud rate returned is 38400

AT+QM\r\n	Chip returns “QM+01\r\n”, indicating the operating mode returned is “playing U-disk or TF card”
AT+QN\r\n	Chip returns “QN+01\r\n”, indicating the chip has prompt tones

3.2.5 Query Commands Related to Music

Music Section -- Query Commands -- Description		
CMD	Corresponding Function	Detailed Description
AT+M1	Physical Sequence Number of Current Device's Playing File	
AT+M2	Total Number of Files on Current Device	
AT+MC	Current Playback Mode	Whether single track, loop, folder loop, etc.
AT+MD	Current Playback Device	Refers to U-disk or TF card
AT+MF	“Long Filename” of Current Playing File	See 5.2.1
AT+MP	Current Playback Status	
AT+MT	Total Duration of Current File Playback	
AT+MK	Elapsed Playback Time of Current File	
AT+MV	Current Online Device	See 5.1.2
AT+MO	Data Returned After Current Playback Completes	This is actively returned by the chip, no query needed, see 5.1.2
AT+MU	Device Insertion and Removal Messages	See 5.1.3

Music Section -- Query Commands -- Examples	
CMD	Detailed Description
AT+M1\r\n	Chip returns “M1+000002\r\n”, indicating the physical sequence number of the currently playing file is 2
AT+M2\r\n	Chip returns “M2+000010\r\n”, indicating the total number of files on the current device is 16
AT+MD\r\n	Chip returns “MD+01\r\n”, indicating the current playback is from a U-disk
AT+MF\r\n	Chip returns “MF+/ 刘 德 华 ~1MP3”, indicating the short filename of the currently playing music
AT+MP\r\n	Chip returns the current status: Stopped [0], Playing [1], Paused [2]
AT+MT\r\n	Chip returns the total duration of the currently playing file
AT+MK\r\n	Chip returns the elapsed playback time of the current file

3.2.6 Query Commands Related to Bluetooth

Bluetooth Section -- Query Commands -- Description		
CMD	Corresponding Function	Detailed Description
AT+TE	Bluetooth Query -- Query Password	
AT+TD	Bluetooth Query -- Query Name (EDR)	
AT+TA	Bluetooth Query -- Query Address (EDR)	
AT+TM	Bluetooth Query -- Query Name (BLE)	
AT+TB	Bluetooth Query -- Query Address (BLE)	
AT+TI	Bluetooth Query -- Query iOS Connection	To be determined
AT+TS	Bluetooth Query -- Query Current Status	
AT+TT	Query Incoming Call Number	See [unspecified section]
AT+T1	Bluetooth Query -- Query Simple Password	Default requires password "0000"
AT+T2	Bluetooth Query -- Query Call Support	Default includes HFP, i.e., Bluetooth call
AT+T3	Bluetooth Query -- Query A2DP Support	Default includes A2DP, i.e., Bluetooth audio
AT+T4	Bluetooth Query -- Query BLE Support	To be determined, default includes BLE function
AT+T5	Bluetooth Query -- Query EDR Support	To be determined, default includes EDR function
AT+T6	Query Service UUID	See 7.6
AT+T7	Query Characteristic 1	
AT+T8	Query Characteristic 2	
AT+T9	Query Characteristic 3	

Bluetooth Section -- Query Commands -- Examples	
Slave Returns	Detailed Description
TE+0000	Indicates the current Bluetooth connection password is "0000"
TD+BT201-AUDIO	Indicates the current EDR Bluetooth name is "BT201-AUDIO"
TA+9EE884AB8BCC	Indicates the current EDR Bluetooth MAC address is "9E E8 84 AB 8B CC", totaling 6 bytes
TM+BT201_BLE	Indicates the current BLE Bluetooth name is "BT201-AUDIO" [Note: Likely intended as "BT201_BLE"]
TB+9EE884AB8BCD	Indicates the current BLE Bluetooth MAC address is "9E E8 84 AB 8B CD", totaling 6 bytes

TS+01	This status indicates the Bluetooth is currently connected successfully but not yet playing music
T1+01	Indicates the default requires a password of “0000”
T2+01	Indicates the chip supports HFP
T3+01	Indicates the chip supports A2DP
T4+01	Indicates the chip supports BLE
T5+01	Indicates the chip supports EDR
TT+15902797635	The incoming call number is 15902797635

4.Detailed Description of Serial Port Commands -- General Section

4.1 Data Actively Returned by the Module

The chip will return data at key points, allowing users to monitor the chip's operating status.

Data returned when the chip powers on and initializes successfully	See 4.4.1
ACK (acknowledgment) returned when the chip successfully receives a command	See 4.4.2
Error information returned	See 4.4.3

4.1.1 Data Returned by the Chip on Power-On [QA][QT][QM]

AT+VER2.3-20190517	Represents the firmware version (2.3) and the last update date (May 17, 2019),with potential changes due to added features or bug fixes.
QA+30\r\n	The volume returned on chip power-on is level 30, see 4.2.1
QM+00\r\n	The mode returned on chip power-on is "Bluetooth mode", see 4.2.3
QN+01\r\n	The chip enables prompt tones by default. (e.g., Bluetooth status tones). See Section 4.2.9.
QK+01\r\n	The chip supports automatic switching to Bluetooth audio in non-Bluetooth states. See Section 4.2.11.
QG+01\r\n	The chip enables Bluetooth background operation by default. See Section 4.2.12.
Q1+01\r\n	The chip enables AD key function by default. See Section 4.2.13.
.....	The rest is about recording and some Bluetooth information, and answers can be found in the Bluetooth sections of the manual.

1. The above parameters are actively returned by the chip on power-on, informing the user that the chip has initialized successfully and is operating normally.
2. The above data is returned by the chip only once on power-on.
3. It is best for the user to wait until receiving the above data before sending control commands.

4.1.2 Acknowledgment Returned by the Chip on Successful Receipt of Serial Port Commands [OK]

OK\r\n	The acknowledgment returned by the chip to the host upon receiving a command -- applies only to control commands sent by the host
--------	--

1. This data is solely the chip's acknowledgment returned to the host, typically in response to control commands sent by the host; query commands do not return this acknowledgment.
2. If the host sends a query command, the query result is returned instead. If the host sends a control command, "OK" is returned as the acknowledgment.

4.1.3 Return of the Chip's Error Information [ER]

ER+1\r\n	The received data frame is incorrect
ER+2\r\n	The received command does not exist, e.g., a string like AT+KK cannot be found

ER+3\r\n	During recording, the device is offline or other errors occur
ER+4\r\n	The sent command is out of range, or the command format is incorrect
ER+5\r\n	The specified device [TF card or U-disk] is offline or abnormal
ER+6\r\n	The specified device [TF card or U-disk] path does not exist, returning an error
ER+7\r\n	To be determined
ER+8\r\n	To be determined

The chip provides real-time feedback for certain error conditions internally. Please refer to the table above for specific details.

4.2 General Section -- Detailed Explanation of Contro Commands

The following provides detailed explanations for some parts of the common functions that may cause misunderstandings. The common functions refer to the parts that are processed simultaneously for each task and have the highest priority within the chip.

4.2.1 Specify the playback volume of the chip [CA][CB][CC][CD][CE][CF]

AT+CA15\r\n	Specifies the volume to level 15 [Note:sending AT+CA03\r\n specifies the volume to level 15, not “ AT+CA3”\r\n]
AT+CB\r\n	No parameters needed. This command is for play/pause. Invalid if Bluetooth is not connected.
AT+CC\r\n	No parameters needed. The function is next track. Invalid if Bluetooth is not connected successfully.
AT+CD\r\n	No parameters needed. The function is previous track. Invalid if Bluetooth is not connected successfully.
AT+CE\r\n	No parameters needed. The function is volume up [Note: maximum is 30]. Valid in any state.
AT+CF\r\n	No parameters needed. The function is volume down [Note: minimum is 0]. Valid in any state.

1. The system's volume ranges from a maximum of level 30 to a minimum of level 0, and the chip automatically saves the setting after power-off. It's level 30[maximum] on power on at first time.
2. The volume is the same across all modes, meaning it is the system-wide volume, not specific to any single mode.
3. Users have two options: they can use the volume up [CE] and volume down [CF] commands directly, or they can specify the volume [CA] directly; both methods are acceptable.
4. The CB command is only valid when Bluetooth is successfully connected or in TF card/U-disk playback mode. If Bluetooth is not successfully connected, sending this command is invalid. In Bluetooth music mode, if the phone's music player is not open, sending this command may also have no effect.

4.2.2 Setting the Chip's Baud Rate [CT]

AT+CT01\r\n	Baud Rate 9600	AT+CT04\r\n	Baud Rate 57600
AT+CT02\r\n	Baud Rate 19200	AT+CT05\r\n	Baud Rate 115200
AT+CT03\r\n	Baud Rate 38400	AT+CT06\r\n	Baud Rate 256000

1. Once the baud rate is set, the chip will remember it. The next time it powers on, the baud rate will be the one you set.
2. After setting the baud rate, please wait 1 second before sending a reset command, or power off and restart the chip. Reinitialize the chip's UART settings
3. If you want to restore the default baud rate, please send the factory reset command. At this point, the chip will automatically erase all configurations.
4. Since our chip's main frequency is very high, **try to set the serial port's baud rate as high as possible—the higher, the better. If the baud rate is too low, when playing music and sending commands, there may be slight noise due to the serial port interrupting the decoding process.**

4.2.3 Setting the Chip's Operating Mode [CM]

AT+CM00\r\n	Switch to the next operating mode
AT+CM01\r\n	Set operating mode to Bluetooth
AT+CM02\r\n	Set operating mode to "U disk" playback. If the U disk is not online, an error message will be returned, and no action will be taken.
AT+CM03\r\n	Set operating mode to "TF card" playback. Same as above.
AT+CM04\r\n	Set operating mode to music mode, used for scenarios with only a single device.
AT+CM05\r\n	Set operating mode to external audio input AUX. Reserved on this version.
AT+CM07\r\n	Set operating mode to "sound card." This is when the chip's USB connects to a computer to play audio, while also allowing copying to a TF card.
AT+CM08\r\n	Set the chip to idle mode, releasing all resources and entering a waiting state.

1. If a mode does not exist, please do not switch to that mode. After switching modes, you should also check if the mode switch was successful. This depends on the return data provided for each mode, which was introduced earlier. For details, see the QM command.
2. Since the chip internally handles U disk and TF card under the same task, switching between these two is relatively troublesome. **If you only use either a U disk or an SD card, it is recommended to use the AT+CM04\r\n command.**
3. After switching to AUX, there are only two options: mute or play. Refer to "AT+CB" or "AT+CU[4.2.4]". Muting in AUX actually sets the volume to 0.
---The current version does not have this function due to insufficient IO, so you can look at the BT301 solution.

4.2.4 Detailed Explanation of Setting Module Mute and DAC On/Off [CU][CS]

AT+CU00\r\n	Unmute
AT+CU01\r\n	Mute
AT+CU02\r\n	If currently muted, unmute. Otherwise, mute.
AT+CS00\r\n	Turn off DAC, at which point the DAC output is high impedance.
AT+CS01\r\n	Turn on DAC, at which point the DAC plays sound normally

1. The chip's mute function means the current playback continues, but the volume is adjusted to 0, so you can't hear any sound.
2. Turning off the DAC is equivalent to completely disconnecting the DAC. The purpose of this is that if the user needs to use the external audio input function, the external audio source can be directly connected to the chip's DAC. By turning off the DAC, the chip will not absorb or interfere with the external audio input in any way.
3. The purpose of the DAC off function is to make it convenient for users to connect an FM chip or use external audio input.
4. Note that after turning off the DAC, if the chip needs to resume playback, you must turn on the DAC or send a command to switch to another mode.

4.2.5 Explanation of Setting the Chip's Auto-Enter Mode on Power-Up [CP]

AT+CP00\r\n	Automatically enter Bluetooth mode on power-up
AT+CP01\r\n	Enter waiting state on power-up, requiring the user to send a mode command
AT+CP02\r\n	On power-up, make a judgment: play the device if one is present, otherwise enter Bluetooth mode
AT+CP03\r\n	Reserved
AT+CP04\r\n	Reserved

1. Similarly, this setting will take effect only on the next power-up.
2. The purpose of adding this command is to make things convenient for users. Some customers need to use Bluetooth directly, while others need to enter device mode if a device is present.
3. Note that if the chip does not enter Bluetooth mode at least once on power-up, Bluetooth data transmission will be invalid because Bluetooth must be initialized once to run in the background.

4.2.6 Setting the Chip's Automatic Data Return Function On/Off [CR]

AT+CR01\r\n	Enable automatic data return function
AT+CR00\r\n	Disable automatic data return function

1. If you don't want the chip to return a message every status changes, you can turn it off with this command.

2. If you don't want the chip to return a message at TF card or U disk playback starting and ending , you can also turn it off with this command.

4.2.7 Setting TF Card or U Disk to Play Once and Stop or Loop Sequentially [CJ]

AT+CJ01\r\n	Enable -- Single track triggered playback
AT+CJ00\r\n	Disable -- Sequential loop playback

1. For MUSIC playback, we provide two application modes here:

(1) Basic mode: Entering music mode starts sequential loop playback, where it keeps playing without stopping. This is called "sequential loop playback."

(2) Personalized mode: Entering music mode stops playback, waiting for the user to specify a track, and stops after the track finishes. This is called "single track triggered playback."

2. This setting is stored in memory. Once set, it takes effect immediately, and it is automatically saved even after power-off.

4.2.8 Setting Whether the TF Card or U Disk Recording Function is Enabled [RE]

AT+RE01\r\n	Enable -- The chip now has recording capability.
AT+RE00\r\n	Disable -- The chip's recording function is turned off.

1. The recording function is a niche application, so we use a hidden approach. If you don't enable it, it is disabled by default.

2. After enabling it through the command, the chip will save the setting even after power-off, so you won't need to set it again next time.

4.2.9 Setting Whether to Enable Prompt Sounds [CN]

AT+CN01\r\n	Enable -- The chip will automatically play prompt sounds.
AT+CN00\r\n	Disable -- The chip will turn off prompt sounds -- Saved after power-off.

1. By default, our chip comes with prompt sounds, meaning it plays a sound when powering on or switching modes. If you don't need this, you can disable it.

2. After enabling or disabling through the command, the chip will save the setting even after power-off, so you won't need to set it again next time.

4.2.10 Setting the Audio EQ [CQ] -- Not Supported Yet

AT+CQ00\r\n	Default	AT+CQ03\r\n	CLASSIC
AT+CQ01\r\n	ROCK	AT+CQ04\r\n	JAZZ
AT+CQ02\r\n	POP	AT+CQ05\r\n	COUNTRY

1. The sound effect is fixed and not saved. Each time the chip powers on, it defaults to the first sound effect (default).

2. The sound effect is globally effective, working for both MP3 playback and Bluetooth. **Note: This function is currently reserved.**

4.2.11 Setting Bluetooth Auto-Switch to Background [CK]

AT+CK00\r\n	Disable -- Do not automatically switch to Bluetooth.
AT+CK01\r\n	Enable -- When playing a TF card, if Bluetooth has audio or a phone call, automatically switch to Bluetooth.

1. When playing a TF card, if Bluetooth has audio or a phone call, we will by default switch to Bluetooth because Bluetooth runs in the background and its memory resources are never released.
2. This command was added to meet the need of not switching to Bluetooth audio or calls while playing a TF card.
3. When playing a TF card, if Bluetooth connects, the TF card's audio may experience a slight interruption.
4. This function does not affect Bluetooth's BLE or SPP functionalities.
5. **"CK+01" automatically returned on chip power-on, represents Bluetooth background operation enabled by default.**

4.2.12 Setting Whether Bluetooth Runs in the Background [CG]

AT+CG00\r\n	Disable -- Bluetooth does not run in the background.
AT+CG01\r\n	Enable -- Bluetooth runs in the background.

1. For our basic functionality, Bluetooth actually resides in memory for a long time, so even when you're playing a TF card, you can still normally search for Bluetooth.
2. The purpose of this is to ensure Bluetooth data transmission is effective in all modes.
3. To conveniently meet the need of not having "Bluetooth run in the background," we added this command. The setting takes effect after power-off.
4. This allows complete release of Bluetooth resources when playing a TF card, and similarly, Bluetooth data transmission is only effective in Bluetooth mode.
5. **"CG+01" automatically returned on chip power-on, represents it is enabled by default.**

4.2.13 Setting Whether Key Functions Are Disabled or Enabled [C1][C2][C3][C4]

AT+C100\r\n	Disable AD key -- Pin 14 of the chip does not need a 22K pull-up resistor.
AT+C101\r\n	Enable AD key -- Pin 14 of the chip must have a 22K pull-up resistor.
AT+C200\r\n	Disable the chip's active status reporting -- The chip will not actively return information on power-up or status changes.
AT+C201\r\n	Enable the chip's active status reporting -- The chip will actively return information on power-up or status changes. This is the default.

AT+C300\r\n	Reserved
AT+C301\r\n	Reserved
AT+C400\r\n	Set Bluetooth to not reconnect on power-up.
AT+C401\r\n	Set Bluetooth to reconnect to the last paired device on power-up, mainly for audio. This is the default.

The setting is saved after power-off and takes effect on the next power-up.

“Q1+01”automatically returned on chip power-on, represents AD key functions enabled by default.

4.3 General Section -- Detailed Explanation of Query Commands

See Section 3.2.4 for details.

4.3.1 Explanation of Common Status Query Returns [QA][QT][QN][QK]

Examples of query commands sent by the host	
CMD	Detailed Explanation
AT+QA\r\n	The chip returns “QA+30\r\n”, indicating the volume returned to the host is 30.
AT+QT\r\n	The chip returns “QT+03\r\n”, indicating the baud rate is 38400.
AT+QM\r\n	The chip returns “QM+01\r\n”, indicating the operating mode is “playing U disk or TF card.”
AT+QN\r\n	The chip returns “QN+01\r\n”, indicating the chip has prompt sounds enabled.
AT+QK\r\n	The chip returns “QK+01\r\n”, indicating the chip actively switches to Bluetooth. See Section 4.2.11 for details.

1. The AT+QA query command is easy to understand. See the volume setting explanation in Section 4.2.1 for details.

2. The AT+QT query command is easy to understand. See Section 4.2.2 for details.

3. The AT+QK query command is understood similarly, as above.

4.3.2 Querying Operating Mode and Explanation of Returns [QM]

AT+QM\r\n	The chip returns “QM+01\r\n”, indicating the operating mode is “Bluetooth mode.”
QM+00	The chip powers on in idle mode, so it returns 00.
QM+01	Indicates Bluetooth mode.
QM+02	Indicates music mode with U disk playback.
QM+03	Indicates music mode with TF card playback.
QM+04	Indicates music mode with SPIFLASH playback -- Not supported in this version.
QM+05	Indicates AUX external audio input mode -- Not supported in this version.
QM+06	Indicates PC sound card input mode.

QM+07	Indicates FM radio mode.
QM+08	Indicates REC recording mode.
QM+09	Indicates fake power-off mode, also known as idle mode.

Refer to Section 4.2.3 for a clearer understanding when used together.

5.Detailed Description of Serial Port Commands -- Music Section

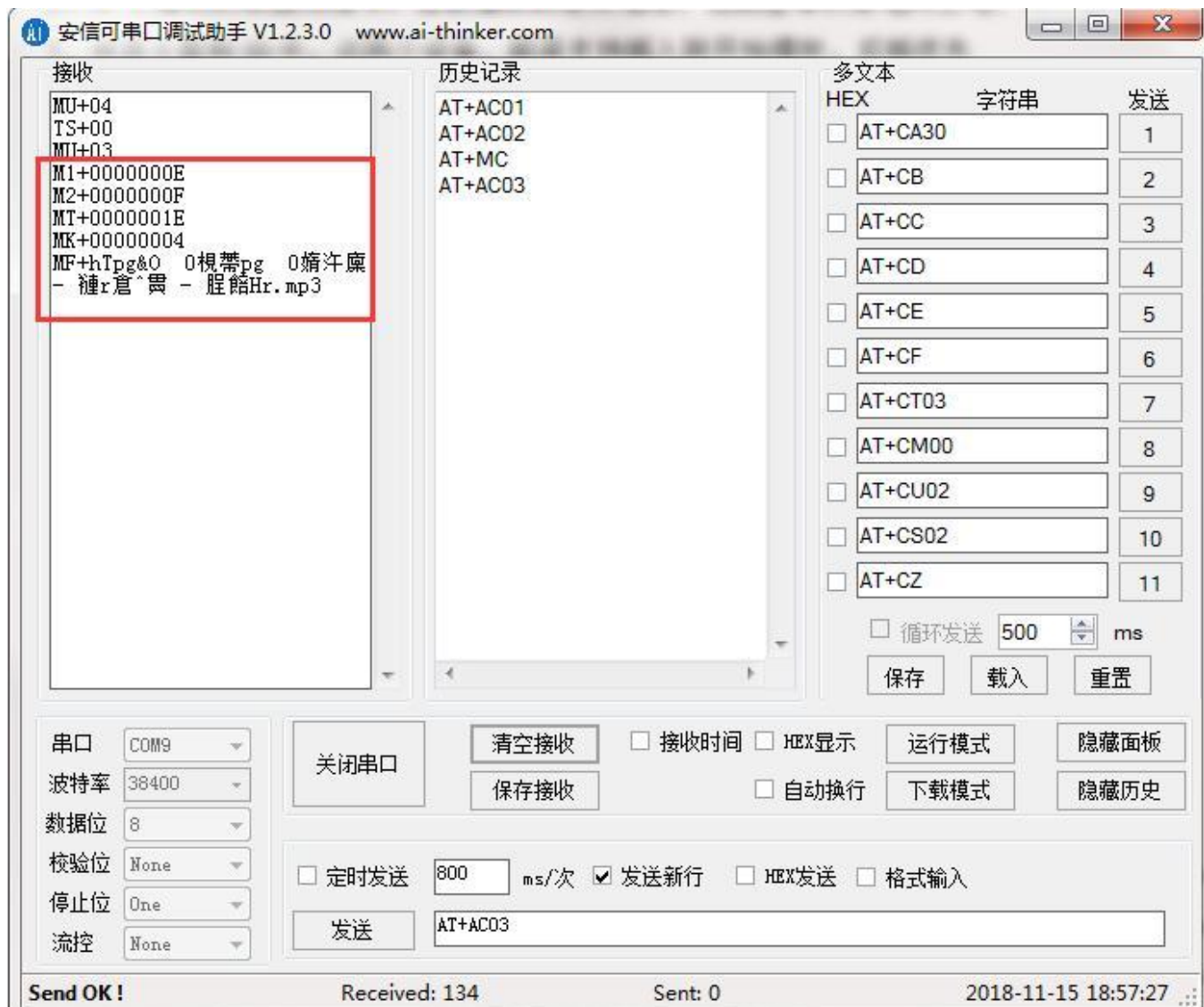
See the following details:

5.1 Music-Related -- Detailed Explanation of Control Commands

The MUSIC function only includes reading and playing from U disk and TF card. The chip internally includes a power-off storage section that can store the following information:

1. The physical track number of the song played from the U disk and the current playback breakpoint information.
2. The physical track number of the song played from the TF card and the current playback breakpoint information. This means that after each power-off and power-on, the chip will automatically start playing from the breakpoint.
3. Both U disk and SD card devices support automatic playback upon insertion. The device inserted later takes priority.

5.1.1 Data Returned After U Disk or TF Card Initialization is Complete [M1][M2][MT][MK][MF]



When playback begins, the chip returns the above information:

M1+0000000E	The current device's playing file is 0x0E.
M2+0000000F	The total number of files on the current device is 0x0F. Note that this refers to valid audio files.
MT+0000001E	The total duration of the current file is 0x1E seconds.
MK+00000004	The current file has played up to 0x04 seconds.
MF+XXXXXXX	The name of the currently playing file is xxxxxx. The file name here is complete.

5.1.2 Information Returned After U Disk or TF Card Playback is Complete [MV][MD][MO]

MV+04	This indicates the online device, referring to the TF card being online, i.e., BIT(2).
MD+04	This indicates the currently active device is the TF card, i.e., BIT(2).
MO+00000005	The track that just finished playing is the 5th track.

Here, the U disk occupies bit 0, and the TF card occupies bit 2.

=1 means online, =0 means offline. At any time, only one device is active and playing music. Being online only indicates physical presence.

5.1.3 Information Returned for TF Card or U Disk Insertion/Removal [MU]

MU+01	U disk inserted
MU+02	U disk removed
MU+03	TF card inserted
MU+04	TF card removed
MU+05	Connect to PC --- can achieve the functions of a card reader and USB sound card
MU+06	PC removed

The insertion and removal of devices will actively return the above information to the host.

5.1.4 TF Card or U Disk Commands -- Common Basic Functions [AA]

AT+AA00	Stop	AT+AA04	Fast forward
AT+AA01	Play [Starts playback if in paused state; otherwise, invalid]	AT+AA05	Fast rewind
AT+AA02	Pause [Pauses if in playing state; otherwise, invalid]	AT+AA06	Next folder
AT+AA03	Play/Pause [Toggles between the two states]	AT+AA07	Previous folder
		AT+AA08	Delete the currently playing file

1. "Stop" means stopping the current file, and if played again, it starts from the beginning. "Pause" means pausing at the current position, and if played again, it resumes from that position.

2. Next/previous folder commands are only effective if your device has folders; otherwise, they are

invalid.

3. The AT+AA08 command refers to deleting the file currently being played.

5.1.5 TF Card or U Disk Commands -- Track Number Playback [AB]

AT+AB1	Play the 1st file on the specified device.
AT+AB11	Play the 11th file on the specified device.
AT+AB111	Play the 111th file on the specified device.
AT+AB65536	Play the 65536th file on the specified device.

1. To make programming easier for users, the commands are standardized:

(1) For example, sending AT+AB1 specifies the first track for playback, and sending AT+AB000001 also specifies the first track. We've handled this internally.

(2) This makes it very flexible for users. You can define a fixed number of characters for this string.

2. The playback order here is based on physical order, **meaning the sequence in which files were copied to the device, not based on numbered labels like 01/02. Refer to document 09 in folder 11 for detailed information.**

3. All files, searches, and sorting within the chip are based on physical order. Please make sure to understand this clearly.

5.1.6 TF Card or U Disk Commands -- Loop Playback of a Specified Folder Path [AF]

AT+AF/01*/*.???	Specify loop playback of the "01 Welcome Mode" folder.
AT+AF/02*/*.???	Specify loop playback of the "02 Forest Mode" folder.
AT+AF/03*/*.???	Specify loop playback of the "03 Seaside Mode" folder.
AT+AF/USB_UPDA/*.???	Specify loop playback of the "USB_UPDA" folder.

名称	修改日期	类型	大小
01迎宾模式	2018-11-12 15:18	文件夹	
02森林模式	2018-11-12 15:18	文件夹	
03海景模式	2018-11-12 15:18	文件夹	
04舒畅模式	2018-11-12 15:18	文件夹	
USB_UPDA	2018-11-12 15:18	文件夹	

1. Here, we provide full path specification for playback. For example, **AT+AF/01*/*.???**

(1) AT+AF is the command itself, no explanation needed.

(2) **The /01* part** indicates a folder under the root directory with an "01" prefix. This means any folder with the "01" prefix will be recognized. The * is a "wildcard," meaning it can represent anything.

(3) The purpose of this is to make it convenient for customers who need Chinese naming, as Chinese names are not always easy to program.

(4) **The /*.???** part indicates any file in the folder, meaning the first file by default. You can also specify a specific file name here.

(5) Note that ??? represents all files. You can change it to MP3 or WAV, and it will only search for

files with MP3 or WAV extensions.

2. For Chinese naming, we recommend using the “wildcard” with a prefix. For English naming, it’s very flexible and can be done either way.

3. Empty folders are not allowed on the device.

4. After specifying a folder, the contents of that folder will automatically loop for playback.

5.1.7 TF Card or U Disk Commands -- Play a Specified File Path Once [AJ]

AT+AJ/01*/011_11.???	Play the file 011_11.??? in the “01???” folder once.
AT+AJ/02*/011*.???	Play the file 011???.??? in the “02???” folder once.

1. The principle is the same as in Section 5.1.6, so it won’t be explained again here.

2. Specifying a folder and file name for playback must refer to a file within a folder, not in the root directory.

See the video demonstration for details: <https://v.qq.com/x/page/y08649mgh1l.html>

5.1.8 TF Card or U Disk Commands -- Specify Playback Mode: Single Track, All, Random [AC]

AT+AC00	Set to loop all.
AT+AC01	Set to loop single device.
AT+AC02	Set to loop single track.
AT+AC03	Set to random playback on single device.
AT+AC04	Set to loop folder.

1. If you only need **single track loop** or **all loop**, just choose the one you need. You can ignore the others.

2. “**Loop all**” refers to looping across both devices, provided both devices are online. If only one device is online, it will loop on that single device.

3. “**Single device**” refers to one device. It will loop only within that device.

4. “**Loop folder**” means if the currently playing track is in a folder, it will loop playback within that folder; otherwise, it is invalid.

5.1.9 TF Card or U Disk -- Recording Function [RC]

AT+RC00	Enter recording standby state, i.e., stopped state.	AT+RC04	Recording, valid on Pause
AT+RC01	Start recording.	AT+RC06	Stop recording.
AT+RC02	Pause or resume recording.	AT+RC06	Play the current recording.
AT+RC03	Pause, valid on recording	AT+RC07	Delete the current recording.

1. Note that the recording function is disabled by default. You need to enable it via a command, and it will be enabled by default on the next power-up. See Section 4.2.8.

2. To record, a device must be online; a TF card or U disk must be inserted for it to work. The recording status can also be checked via the indicator light. **Refer to the BT201 module:**

(1) Fast flashing during recording, slow flashing in stopped state, steady on in paused state.

(2) On the test board, the PP key starts or pauses recording. The Mode key short-presses to switch modes and long-presses to delete recordings.

3. You must first enter the recording standby state, then send the start recording command, at which point recording will begin automatically.

(1) The recording format is MP3 with a 128Kbps sampling rate.

(2) Recording files are generated in the “KT_REC” folder.

4. During recording, you can pause and then resume recording by sending AT+RC02.

5. After recording is complete, if you don’t need to play it, send the AT+RC03 command to stop.

(1) To play, send the AT+RC04 command, which will automatically stop recording and play the just-recorded file.

(2) After recording, you must specify a switch to your desired operating mode, otherwise the resources will still be used by the recording thread.

(3) After entering MUSIC mode, you can specify a path to play previous recordings. Check the generated recording files for patterns.

In MUSIC mode, you can still delete files; see Section 5.1.4. Note that all recordings can only be deleted during playback.

6. During recording, the chip automatically returns the current recording status when the state changes:

(1) RC+00 -- Indicates recording stopped state. (2) RC+01 -- Indicates recording started state.

(3) RC+02 -- Indicates recording paused state. (4) RC+03 -- Indicates recording playback state.

(5) RC+04 -- Indicates recording playback completed state.

(6) RC+05 -- Indicates recording playback completed state.

(7) RC+06 -- Indicates deletion of the just-recorded file state.

(8) RC+07 -- Indicates recording initialization completed -- State returned upon entering recording.

7. Recordings follow **a folder and file naming rule**. Recording files are fixed in the “KT_REC” folder and cannot be changed. New files are sequentially numbered from FILE0001 to FILE9999.

8. Each time recording is started, the chip checks the files in the “KT_REC” folder, starting from 0001. If 0001 exists, it generates 0002; if 0002 exists, it generates 0003, and so on, until it finds a non-existent file name, then starts recording and generates the file.

Additionally, when generating a recording file, some information is provided:

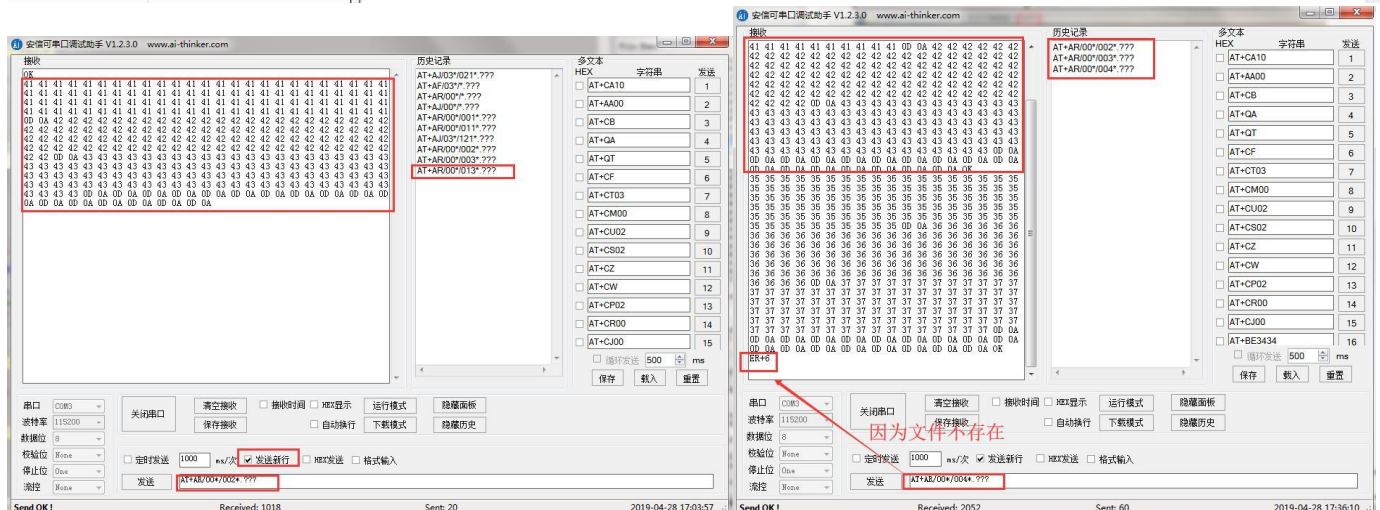
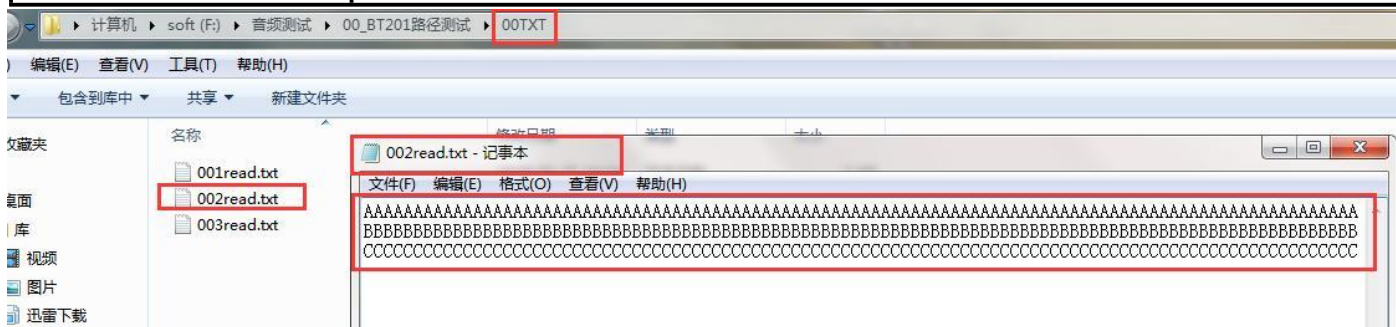
(1) Existing file name: RF+

(2) Newly generated file name: RN+ (as shown in the screenshot below).



5.1.10 TF Card or U Disk Command -- Read Data from Specified File Path (TXT Reading) [AR]

AT+AR/00*/002*.*???	Reads files matching 002*.*??? in folders matching 00* (e.g., /001/002_config.txt)
AT+AR/00*/003*.*???	Reads files matching 003*.*??? in folders matching 00* (e.g., /001/003_data.bin)



1. Refer to Section 5.1.6 for principle, not explain here again. Current music playback will stop upon sending AT+AR.

2. Data is sent via UART in hexadecimal, note is sent byte-for-byte (note that content in txt screenshot above, 0x41 = ASCII "A", 0x42 = "B").

3. If the file to be read is less than 512 bytes, it will be read once and then exit. The music playback will also stop at this time.

4. If the file to be read is larger than 512 bytes, for user convenience, it is set to read 512 bytes at

a time. After reading, the data is forwarded through the serial port, followed by a 500ms delay before reading the next 512 bytes. This continues until the entire file is read, facilitating users to receive and store data simultaneously.

5. There are no restrictions on file types; users can use txt, bin, or other formats. Here, txt is used as an example for illustration.

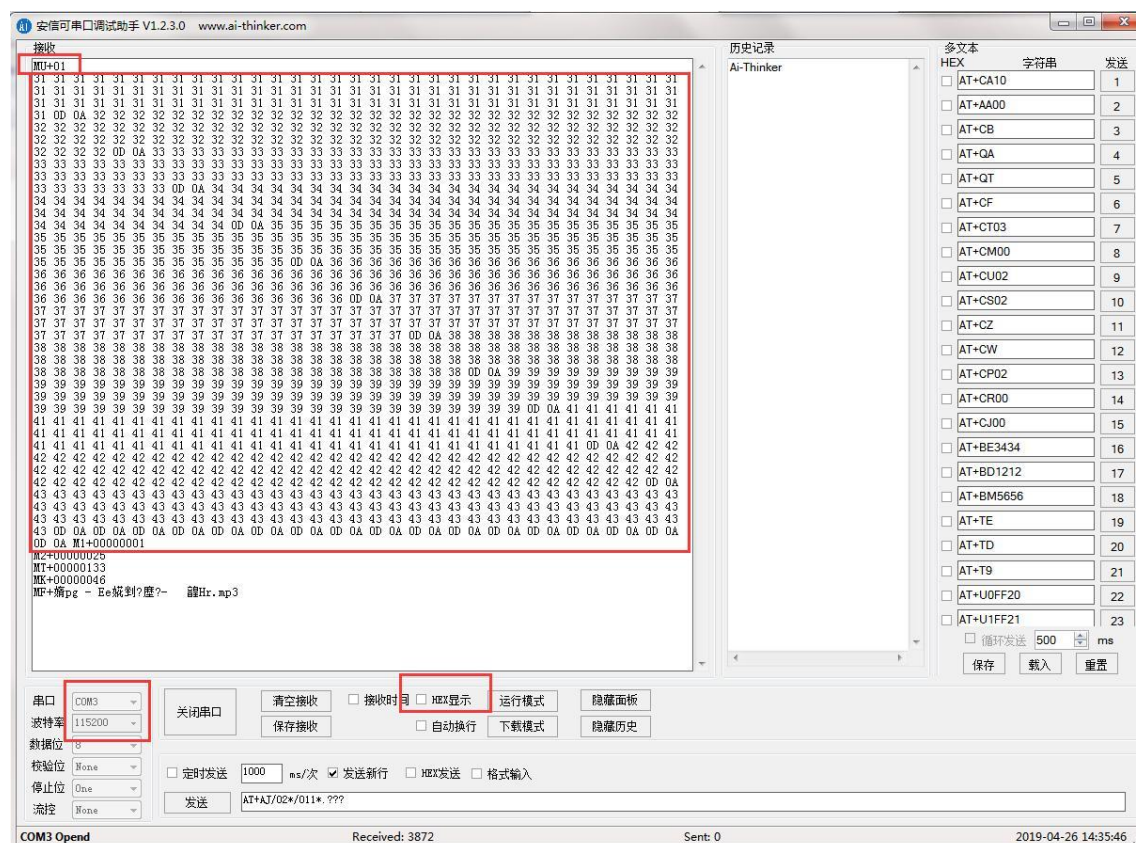
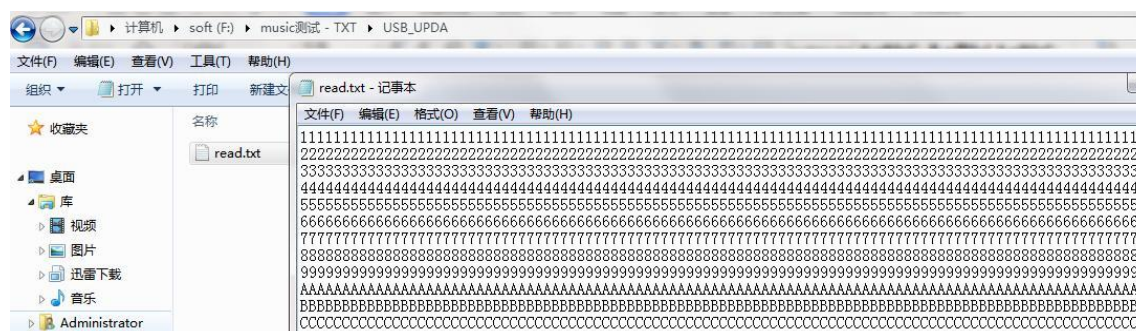
6. The more content in the TXT file, the longer it takes. The actual reading time depends on testing. In practice, reading is very fast, with 512 bytes read in up to 2ms.

7. When users need to read a file, they can send a command via the serial port. This is only effective after the U disk or TF card initialization is complete.

8. If the read is successful, the corresponding data will be returned. If the read fails or the file does not exist, an error ER6 will be returned. See section 4.1.3 for details.

The actual test results are as shown in the three figures above.

5.1.11 Serial Port Reading of TXT and Other File Formats from TF Card or U Disk [Advanced Function]



1. The TXT document must be located in the "USB_UPDA" folder and named "read.txt." This function supports automatic reading.
2. If the file to be read is a bin file, the processing method is the same. The file to be read must be renamed to "read.txt."
3. All file types can be read, but please ensure the file name is modified accordingly.

For details, refer to document number "19" in the "11_Functional Usage Manual_Quick Application" folder in our separate documentation.

5.1.12 TF Card or U Disk Command—Delete File at Specified Path [AL]

AT+AL/00/002.???**	Deletes the file "002xxx.???" in the "00xxx" folder.
AT+AL/02*/*.???	Deletes any single file in the "02???" folder. Note that only one file can be deleted at a time.

After recording, there are two methods to delete unwanted files:

Method 1: Delete individual files (not entire folders) by sending multiple delete commands (see Section 5.1.12).

Method 2: Use the AT+AA08 command to delete files (see Section 5.1.4).

5.1.13 Recording Function—TF Card or U Disk—Specify Folder and Filename for Recording [RP][RB][RV]

AT+RP/KT2REC/20190506	Creates the "KT2REC" folder and generates the file "20190506.MP3" inside it.
AT+RP/KT-REC/190506	Creates the "KT-REC" folder and generates the file "190506.MP3" inside it.

AT+RB00	Sets recording bitrate to 16 kbps.	AT+RB05	Sets recording bitrate to 96 kbps.
AT+RB01	Sets recording bitrate to 24 kbps.	AT+RB06	Sets recording bitrate to 128 kbps.
AT+RB02	Sets recording bitrate to 32 kbps.	AT+RB07	Sets recording bitrate to 144 kbps.
AT+RB03	Sets recording bitrate to 48 kbps.	AT+RB08	Sets recording bitrate to 160 kbps.
AT+RB04	Sets recording bitrate to 64 kbps.	AT+RB09	Sets recording bitrate to 192 kbps.

AT+RV35	Sets the recording microphone gain to 35. The microphone gain range is 0 – 63, with a default of 35.
AT+RV63	Sets the recording microphone gain to 63. Recommended range is 30 – 40, based on actual testing.

1. Recording Method: Recordings must be generated inside a folder, not in the root directory.
==>The recording sample rate is fixed at 48 kHz. Files are automatically encoded in MP3 format, optimized for both size and audio quality.

2. Example: Sending the command AT+RP/KT-REC/20190506 triggers the following process:

(1) The chip retrieves the path "/KT-REC/20190506" and extracts the folder name "KT-REC."

(2) The folder name must be exactly 6 characters long (e.g., "123456"). It cannot be longer or shorter, and Chinese characters are not allowed.

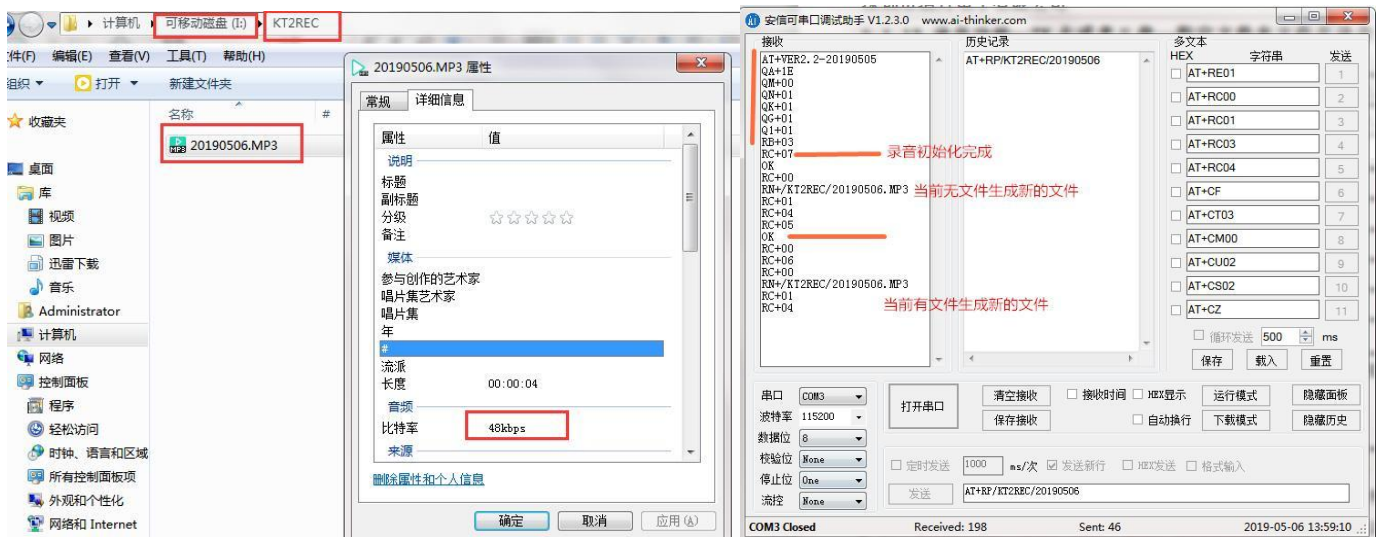
(3) The filename "20190506" must not exceed 8 characters. It can be shorter but not longer.

(4) If the file "/KT-REC/20190506" already exists, the chip deletes it before creating a new file.

3. Recording Bitrate: The bitrate is configurable and saved even after power-off. The bitrate affects recording quality and the size of the generated MP3 file.

(1) The default bitrate is 48 kbps, which is sufficient for most cases. It is not recommended to change it, but if necessary, avoid exceeding 128 kbps.

(2) Upon power-on, the chip reports the stored bitrate value. Refer to the serial port output information at startup.



4. Example Information:

The left figure shows an example of the generated file information.

The right figure shows a screenshot of the serial port interaction for recording.

(1) After entering recording mode, the chip returns "RC+07," indicating that recording initialization is complete, and remains in this state.

(2) When recording starts, the chip returns "RC+01," indicating active recording. "RC+04" indicates playback of a previously recorded file (see section 5.1.9).

(3) "RC+05" indicates that playback of the recorded file is complete, and the chip remains in this state.

(4) If the current file and the file specified in the serial command have the same name, the existing file is deleted first, and "RC+06" is returned.

5.2 Music-Related -- Detailed Explanation of Query Commands

See Section 3.2.5 for details.

5.2.1 Querying the Currently Playing File Name on TF Card or U Disk [MF]

MF+001_Chen0Punch - Everytime - 脛脩 Hr.mp3	001_Chen、Punch - Everytime - 铃声版.mp3
---	--------------------------------------

1. The left side shows the information printed by the serial port debugging assistant, and the right side shows the actual information.

2. When using a serial port debugging assistant, the printed information for Chinese characters or other non-English text appears as garbled code, while English and numbers are correct.

3. Analyzing the data and converting it to hexadecimal yields the following:

```
30 00 30 00 31 00 5F 00 43 00 68 00 65 00 6E 00
01 30 50 00 75 00 6E 00 63 00 68 00 20 00 2D 00
20 00 45 00 76 00 65 00 72 00 79 00 74 00 69 00
6D 00 65 00 20 00 2D 00 20 00 C3 94 F0 58 48 72
2E 00 6D 00 70 00 33 00
```

(1) The highlighted part (in red in the original) is the encoding for Chinese characters, totaling three characters “铃声版”. Each Chinese character uses 2 bytes and follows “UNICODE encoding.”

(2) Even the initial “0” occupies 2 bytes, i.e., 00 30. [Note: The encoding for the character “铃” is 0x94C3.]

(3) You can check the encoding at the following website:

<https://bianma.supfree.net/chaye.asp?id=94C3>

4. For non-Chinese or non-English text, it is currently unclear whether they are supported. Those who know can discuss this.

```
MB+00000001
MC+0000000F
MT+00000012
MK+00000000
MF+001_Chen 0Punch - Everytime - 脛脩Hr.mp3
MB+00000002
MC+0000000F
MT+00000012
MK+00000000
MF+002_K.Will - 泄t? P杞? - 脛脩Hr.mp3
MB+00000003
MC+0000000F
MT+00000035
MK+00000000
MF+003_Ngc脛m - N-\1\ - 脛脩Hr.mp3
MB+00000004
MC+0000000F
MT+0000001D
MK+00000000
MF+004_豎Y??- 脛脩 - 脛脩Hr.mp3
MB+0000000A
MC+0000000F
MT+0000002E
MK+00000000
MF+脛脩P[ - 淩@w脛R?U?`Lk - DJHr脛脩.mp3
```

5. The screenshot above shows the information printed by the serial port debugging assistant. The actual corresponding file names are as follows:

001_Chen、Punch - Everytime - 铃声版.mp3

002_K.Will - 말해! 뭐해# - 铃声版.mp3

003_李荣浩 - 不将就 - 铃声版.mp3

004_梁静茹 - 问 - 铃声版.mp3

龙梅子 - 喝着烈酒唱情歌 - DJ 版铃声.mp3

5.2.2 TF or U Disk Query for Current File Playback Time Processing [MT][MK]

MT+0000001D	The total duration of the current file is 0x1D seconds.
MK+00000000	The current playback time is 0 seconds.

1. Here, we provide the total duration of the current file and the playback time.
2. When using this, users only need to know the total duration and the playback time, then keep track of time themselves.
3. Since this is an audio Bluetooth chip, audio playback uses interrupts, and UART also uses interrupts. If you frequently query the current time, it may cause slight background noise in the playback sound.

6.Detailed Description of Serial Port Commands -- Bluetooth Section

6.1 Bluetooth-Related -- Detailed Explanation of Control Commands

See Section 3.2.3 for details, which includes many examples and is explained very clearly. Take a close look.

1. AT+B1: This sets a simple password, meaning you don't need to manually enter a password when connecting from a phone.
2. AT+B2: This is for setting up calls. If your product doesn't need call functionality, you can disable it here, and the call function will be off on the next power-up.
3. AT+B3: This is for setting up audio. If you don't need music playback, you can disable it here as well.

6.1.1 Setting Bluetooth Name and Password [BD][BE][BM]

AT+BDAUDIO\r\n	Set the Bluetooth name to "AUDIO."
AT+BE1234\r\n	Set the Bluetooth connection password to "1234."
AT+BM2345\r\n	Set the BLE Bluetooth name to "2345."

1. After setting the Bluetooth name, you need to reset the chip by sending a command or powering it off and on. This will display the new Bluetooth name. The default Bluetooth name is "BT201-AUDIO".
2. The maximum length for the Bluetooth name is 32 bytes. Please do not exceed this limit.
3. After setting the Bluetooth password, **you need to reset the chip, either by sending a command or powering off and on.** The chip will then require the password. The default Bluetooth password is "0000."
4. The Bluetooth password can be up to "4" bytes long. Please do not exceed this range.
5. **If you modify the Bluetooth name using AT commands, note that your mobile device may not update immediately and may still display the previous name.**
 - (1) **This is because you only changed the Bluetooth name, while the Bluetooth MAC address remains unchanged. Therefore, the mobile device will not update the name.**
 - (2) **To resolve this, try searching with another mobile device, or delete the pairing information on the original device and search again.**
 - (3) **Rest assured that once the Bluetooth name is set, it is definitely updated. The chip will also return the Bluetooth name upon power-up for verification.**

6.1.2 Setting Bluetooth Protocol Functions [B1][B2][B3]

AT+B100\r\n	This 00 represents 0x00, disabling the pairing password, meaning the next connection does not require a password and connects directly.
AT+B201\r\n	This 01 represents 0x01, enabling Bluetooth call functionality. If set to 00, it disables Bluetooth call functionality.

AT+B300\r\n	This 00 represents 0x00, disabling Bluetooth audio, meaning even if connected successfully, music cannot be played. The opposite applies if enabled.
AT+B401\r\n	This defaults to 0x01.

This is a personalized function, generally not needed by most customers. If you need it, just take a close look when using it. The parameters 00 in the commands indicates disable, 01 indicates enable.

6.1.3 Setting Bluetooth BLE and EDR Enable/Disable [B4][B5]

AT+B400\r\n	Disable BLE functionality. This defaults to 0x01 (enabled).
AT+B501\r\n	Enable EDR functionality. This defaults to 0x01 (enabled).

This is a personalized function, generally not needed by most customers. If you need it, just take a close look when using it.

1. Enabling or disabling BLE is convenient for customers who only need audio and not data transmission.
2. After disabling BLE, you will no longer be able to search for the BLE name, and the chip will not broadcast externally. The setting takes effect on the next power-up.
3. If you disable EDR, your phone will only detect the BLE name and can only perform data communication. This also takes effect on the next power-up. (1) In this case, the chip will not have Bluetooth audio playback functionality, and your phone will not detect the audio Bluetooth name.

6.1.4 Specifying a Phone Number to Call [BT]

AT+BT15902797635\r\n	Initiate a call to the phone number "15902797635."
AT+BT10086\r\n	Initiate a call to the phone number "10086."

To use this function, the following three conditions must be met: the call functionality must be enabled in the software, the device must be in a successfully connected state, and it must not currently be in a call. Only when these three conditions are satisfied can this function be used.

6.1.5 Bluetooth Audio-Related Control Commands [BA]

AT+BA00\r\n	Redial the last phone number	AT+BA05\r\n	Scan for devices.
AT+BA01\r\n	Disconnect	AT+BA06\r\n	Enable Bluetooth audio discoverability.
AT+BA02\r\n	Reject an incoming call	AT+BA07\r\n	Disable Bluetooth audio discoverability.
AT+BA03\r\n	Hang up the call	AT+BA08\r\n	Play Bluetooth audio [only valid when paused].
AT+BA04\r\n	Answer an incoming call	AT+BA09\r\n	Pause Bluetooth audio [only valid when playing].

1. AT+BA01\r\n Disconnect: This command is only effective after a successful connection has been established.

2. AT+BA04\r\n If there is an incoming call, sending this command will answer the call. During the call, sending the command again will hang up.

6.1.6 Bluetooth MAC Address Setting -- EDR -- BLE [BS]

AT+BS123456781234\r\n	Set the EDR MAC address to 0x12 0x34 0x56 0x78 0x12 0x34. The BLE address will be 0x13 0x34 0x56 0x78 0x12 0x34. --- There is a mapping relationship between the two.
AT+BSABCDEF123456\r\n	Set the EDR MAC address to 0xAB 0xCD 0xEF 0x12 0x34 0x56.

This is a personalized function, generally not needed by most customers. If you need it, just take a close look when using it.

1. If the MAC address is not set, the chip will use a default Bluetooth MAC address internally.
2. If the user sets a MAC address, the chip will prioritize the user-set MAC address.
3. Since the chip is dual-mode, it has two Bluetooth names, meaning it has two Bluetooth MAC addresses. The BLE address is associated with the EDR address, with the first byte of the EDR MAC address incremented by 1.

6.2 Bluetooth-Related -- Detailed Explanation of Query Commands

Many Bluetooth statuses are actively returned to the user, so there's usually no need for the user to actively query unless specifically required.

6.2.1 Bluetooth Current Status Return -- EDR Simple Status [TS]

TS+00	Bluetooth has not yet connected successfully and is in the waiting-to-pair state.
TS+01	Bluetooth has connected successfully but is not playing music. Idle state.
TS+02	Music is currently playing.
TS+03	A call is being made or an incoming call is received, but it has not been answered.
TS+04	In-call state, indicating the call is connected.
TS+FF	Indicates that the Bluetooth audio resources have been released, Bluetooth is not running in the background, and the device is not in Bluetooth mode.

1. Here, we provide five simple states. **EDR refers to the Bluetooth audio and call functionalities.**
2. You can choose to query or not query. Users can also disable the chip's automatic return function; see Section 4.2.6.
3. **For Bluetooth, there is no concept of "pause." Bluetooth only has the five states listed above. Even if the phone manually pauses playback, the Bluetooth chip will show the state as "TS+01," which is idle.** Of course, users can send AT+TS for an active query.

6.2.2 Bluetooth Current Status Return -- BLE Simple Status [TL]

TL+00	BLE is in an empty state.	TL+04	BLE has disconnected.
TL+01	BLE is in an idle state.	TL+05	BLE is in listening state.

TL+02	BLE is in broadcasting state.	TL+06	BLE is in scanning state -- Host.
TL+03	BLE has connected successfully.	TL+07	BLE has completed searching -- Host.

1. Here, we provide eight simple states. The first six represent the slave device states, while the last two are for the host states. Of course, users can send AT+TL for an active query.

2. The chip will actively return status updates. Users can disable this; see Section 4.2.6. **The chip only returns updates when its state changes, otherwise it does not return anything.**

6.2.3 Bluetooth Incoming Call Number Return [TT]

TT+15902797635	The caller's phone number is 15902797635.
TT+10086	The caller's phone number is 10086.

1. Here, we actively return the information to the user every 2 seconds until the call is answered or hung up.

2. You can choose to query or not query. You can also disable the chip's automatic return function; see Section 4.2.6.

7.Detailed Description of Bluetooth transparent transmission—BLE

Currently, the module supports dual-mode data transmission for BLE and SPP and can achieve transparent transmission. Both BLE and SPP can only operate as slaves, i.e., the "SERVER" side. Regarding Bluetooth data transmission, no commands are required for intervention. Once the connection is successfully established, the Bluetooth chip will send whatever data it receives through the serial port. Similarly, any data received by the serial port will be directly transmitted via Bluetooth to the mobile phone or other host device. For details, refer to the test documentation for BLE or SPP.

7.1 BLE transparent transmission Description

1. The maximum single data throughput is 128 bytes, supporting both 16-bit and 128-bit UUIDs. A 128-bit UUID requires special customization.
2. If using BLE for data transmission, please connect to the Bluetooth name "BT201-BLE".
3. You can use BLE alone without the audio function. See Section 6.1.3 for details.
4. Of course, you can also disable the BLE function. See Section 6.1.3 for details.

7.2 BLE UUID Description

1. The main UUID is "FFF0".
 2. The UUID for Characteristic 1 is "FFF1", with the characteristics "WRITE" and "NOTIFY".
 3. The UUID for Characteristic 2 is "FFF2", with the characteristics "READ" and "NOTIFY".
 4. The UUID for Characteristic 3 is "FFF3", with the characteristic "WRITE".
- Note:**This feature is specifically used for controlling the Bluetooth chip from a mobile device. See Section 7.5 for details.
5. If you need a specific UUID, please contact us modify the module's underlying settings for customization. However, users can generally set the 16-bit UUID themselves without any issues. See Section 7.6 for details.

7.3 BLE Effect Demonstration Explanation

BLE transparent transmission effect demonstration: <https://v.qq.com/x/page/q07660m1bta.html>

7.4 BLE Testing Instructions

1. For Android and iOS (Apple) phones, it is recommended to use the "LightBlue" software.
2. For Apple devices, you can search and download the app directly from the "App Store".
3. For Android devices, **we will provide the installation program in the resource package.**
4. **Please note that Android phones can also test BLE. Testing BLE is not limited to Apple devices.**
5. **BLE on Android is not unusable, but it is less user-friendly. Android versions must be 4.3 or higher to support BLE.**
6. **Due to the poor usability of BLE on Android, the dual-mode option is available: use SPP for**

Android and BLE for Apple.

7. For Apple devices to use SPP, an MFI-certified chip is required, which is extremely expensive. Currently, this is rarely used.

8. If the Bluetooth name has not been modified, connect to the Bluetooth name "BT201-BLE".

9. BLE testing demonstration video: <https://v.qq.com/x/page/o0766ubm78n.html>

7.5 Testing Explanation for Controlling the Bluetooth Chip via BLE from a Mobile Device



1. This function's testing uses the LightBlue app, and the phone must be an iPhone.

2. This is because only the LightBlue app on iPhone can send strings, while Android versions cannot.

3. Here, the command AT+BD66667 sets the EDR name to "66667." Note that you don't need to add "\r\n" because the mobile device cannot input this, so we automatically add "\r\n"

internally.

4. This function is optional; you can choose to use it or not.

7.6 Explanation of Modifying BLE UUID via AT Commands

AT+U0F000\r\n	Set the service UUID to "F000".
AT+U1F001\r\n	Set Characteristic 1 UUID to "F001", with the properties "Write " + "Notify".
AT+U2F002\r\n	Set Characteristic 2 UUID to "F002", with the properties "Read" + "Notify".
AT+U3F003\r\n	Set Characteristic 3 UUID to "F003", with the property "Write".

1. We provide three characteristics here to accommodate the diverse needs of different customers.
 (1)Some customers prefer to separate "Write" and "Notify" into different characteristics to ensure more reliable data interaction between the phone and the Bluetooth chip.
 (2)Some customers only need to transmit a small amount of data and prefer simplicity, so they combine "Write" and "Notify" into a single characteristic.

2. When setting up, it is recommended to configure all four UUIDs, regardless of whether you use them or not.

3. For customers without an app, it is recommended to use Characteristic 1 and Characteristic 2, separating them into two different characteristics.

4. For products already in mass production that want to switch to our chip, only the UUID needs to be modified.

5. Characteristic 3 is specifically used for controlling the Bluetooth chip by directly sending AT commands from a mobile device.

7.7 BLE Data Transparent Transmission Packet Time Intervals

UART Baud Rate: 9600	Minimum Packet Interval: 20 milliseconds, Optimal Packet Interval: 30 milliseconds (optimal)
UART Baud Rate: 19200	Minimum Packet Interval: 16 milliseconds, Optimal Packet Interval: 20 milliseconds (optimal)
UART Baud Rate: 38400	Minimum Packet Interval: 10 milliseconds, Optimal Packet Interval: 15 milliseconds (optimal)
UART Baud Rate: 57600	Minimum Packet Interval: 8 milliseconds, Optimal Packet Interval: 10 milliseconds (optimal)
UART Baud Rate: 115200	Minimum Packet Interval: 6 milliseconds, Optimal Packet Interval: 8 milliseconds (optimal)
UART Baud Rate: 256000	Minimum Packet Interval: 4 milliseconds, Optimal Packet Interval: 6 milliseconds (optimal)

BLE large data volume transparent transmission demonstration video:

<https://v.qq.com/x/page/c0843j975hl.html>

7.8 Explanation of Modifying BLE Broadcast Packet Data [UR][TR]

AT+UR9988776655\r\n	Set the BLE broadcast packet data to “9988776655.” Note that this is a string.
AT+UR123456789A\r\n	Set the BLE broadcast packet data to “123456789A.” Note that this is a string.
AT+TR\r\n	Query the chip’s broadcast packet data. The chip will return “TR+9988776655.”

1. The BLE broadcast packet is actually a bit complex, but users don’t need to worry about it. We have reserved 10 bytes of space for customers to fill.
2. When setting, users must always set 10 bytes, even if only 1 byte is used; all 10 bytes must be filled.
3. Once set successfully, the broadcast packet data will be changed on the next chip power-up.
4. For details, please refer to our separate document in the 02 folder, document 17xxxx.

For the mobile app, use:

Android: Install the nRF Connect app.

iOS: Install the LightBlue app.

SCANNER

BONDED

ADVERTISER

N/A

6E:20:70:E2:AB:43

NOT BONDED

CONNECT

Raw data:

0x0201060303F0FF0BFF393938383737363635350A0942543230312D424C45

Details:

LEN.	TYPE	VALUE
2	0x01	0x06
3	0x03	0xF0FF
11	0xFF	0x39393838373736363535
10	0x09	0x42543230312D424C45

LEN. - length of EIR packet (Type + Data) in bytes,

TYPE - the data type as in <https://www.bluetooth.org/en-us/specification/assigned-numbers/generic-access-profile>

OK

BrEdrNotSupported

Complete list of 16-bit Service UUIDs: 0xFFFF0

Manufacturer data (Bluetooth Core 4.1):

Company: Reserved ID <0x3939>

0x3838373736363535

Complete Local Name: BT201-BLE

CLONE

RAW

MORE

8.Detailed Description of Bluetooth transparent transmission—SPP

SPP uses the classic Bluetooth 2.1 protocol, which is not recommended for use. For new products, it is recommended to directly use BLE.

For details on Bluetooth data transparent transmission, refer to the separate document titled "26_Detailed Explanation of Bluetooth Data transparent transmission and AT Commands.pdf."

8.1 SPP transparent transmission Description

- | |
|--|
| 1. The maximum single data throughput is 256 bytes. |
| 2. If using SPP for data transmission, do not actively connect to the Bluetooth name "BT201-BLE" or any custom BLE Bluetooth name. |
| 3. Note that SPP is just a sub-protocol within EDR. |
| 4. SPP data transmission and BLE are mutually exclusive. If you only use SPP for data transmission, please disable BLE. |

8.2 SPP Effect Demonstration Explanation

SPP transparent transmission effect demonstration: <https://v.qq.com/x/page/b0766jqw0p5.html>

8.3 SPP transparent transmission Testing Instructions

- | |
|---|
| 1. For Android phones, use the "Bluetooth Serial Port" app, which can be downloaded from the "App Store". |
| 2. If the Bluetooth name has not been modified, connect to the Bluetooth name "BT201-AUDIO". |
| 3. SPP testing demonstration video: https://v.qq.com/x/page/e0766bz15fw.html |

SPP large data volume transparent transmission demonstration video:

<https://v.qq.com/x/page/c0843j975hl.html>

9. Module Firmware Update and Serial Port Testing Explanation

9.1 Explanation of Module Firmware Update

1. Since we designed a test board to make it convenient for users to directly test functions, it is not very compact.
2. In the future, we will release a matching module product.
3. The chip has many functions, and it's impossible to include all of them in a standard firmware. Therefore, for the BT201 module, we have reserved an interface for upgrading the program. The upgrade method is as follows:

1. Copy the “updata.bfu” file to a TF card or U disk. The TF card or U disk must contain at least 3 MP3 audio files for the file system to read.
2. Power on the prototype, then insert the U disk. The program will automatically update, and it will take approximately 30 seconds to complete.
3. Signs of a successful upgrade: If there is speaker connected, the speaker will continuously beep. If there is an indicator light connected, it will stay on during the upgrade and turn off when the upgrade is complete.
4. This method is now fully described. For machines with prompt sounds enabled, inserting a U disk or TF card will trigger a prompt sound.
5. Note: After the upgrade is complete, you must delete the upgrade file from the card, otherwise it will repeatedly attempt to upgrade.
6. Firmware update video demonstration: <https://v.qq.com/x/page/f0766kfjzob.html>

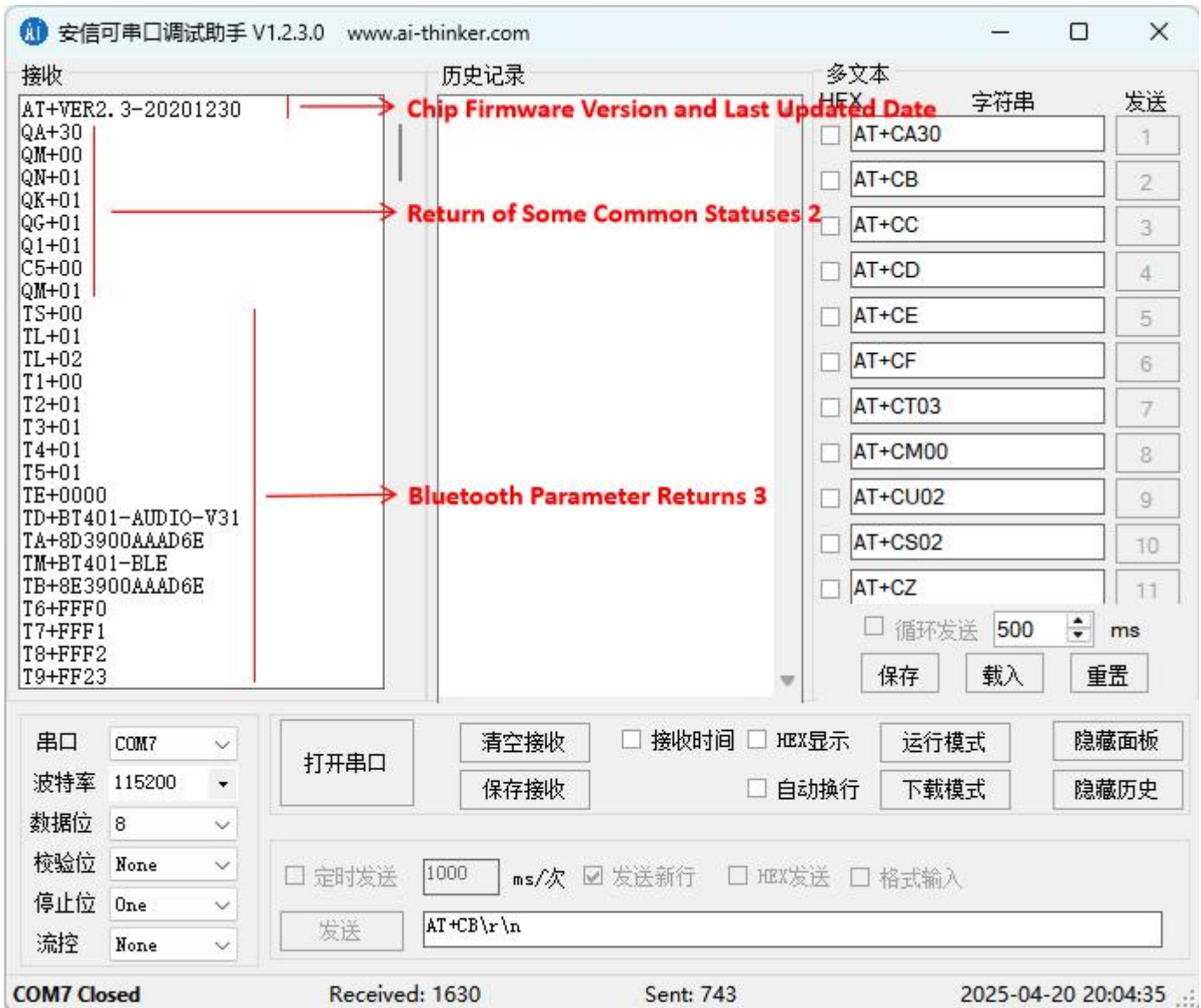
9.2 Potential Doubts or Issues During Module Update -- Update Failure

Question 1	Are there any requirements for the TF card and U disk?
Answer	The TF card and U disk must use the FAT or FAT32 file system, with a maximum capacity of 32GB.

Question 2	Why does my prototype start playing music directly when I copy the “updata.bfu” file to a TF card for upgrading and insert it into the prototype?
Answer	<ol style="list-style-type: none"> 1. Please ensure the file name is exactly “updata.bfu.” The name must match perfectly, with no extra or missing letters. 2. “Updata” is the file name, and “bfu” is the file extension. 3. When working on Windows, make sure to enable “Show file extensions” in the settings.

Question 3	Why doesn't the prototype enter upgrade mode after I copy the "updata.bfu" file to a TF card, insert it, and power on the prototype?
Answer	<p>1. Our upgrade process requires you to power on the prototype first, then insert the TF card.</p> <p>2. The upgrade principle is as follows (the same applies to a U disk):</p> <p>(1) After the chip detects the TF card insertion, it automatically initializes the TF card and the file system.</p> <p>(2) If initialization is successful, it searches for the "updata.bfu" file. If found, it enters upgrade mode.</p> <p>(3) If any of these steps fail, the upgrade will not succeed.</p>

9.3 Explanation of the Module's Serial Port Debugging Assistant



Test Environment: BT201 Test Board

Serial Port Software: Serial Debugging Assistant (aithinker_serial_tool_v1.2.3)

1. Receive Window: Data returned from the chip to the computer. This includes the firmware version and the date of the last modification.
2. Return of Some Common Parameters: See Section 4.3 for details. This includes information such as volume, mode, whether Bluetooth runs in the background, and more.
3. Bluetooth Parameters: For example, UUID and others. See Section 6 for details.
4. The chip's factory default baud rate is 115200.

Many of the returned messages here don't need to be focused on by the user, as their purpose is to make it convenient for customers to view during initial debugging.

10.Frequently Asked Questions (FAQs)

Question 1	1. Can this module connect to a phone to play music? Does it support calls?
Answer	Yes, the module supports Bluetooth audio and Bluetooth calls. Additionally, it supports data transparent transmission(BLE and SPP).
Question 2	2. Your board is too large to embed in our product. Can I purchase the chip directly? Is the peripheral circuitry complex?
Answer	<p>Fundamentally speaking, the BT201 Module is a minimum system for Bluetooth audio decode and playback, BLE or SPP data transmission based on the KT1025A and KT1025B Bluetooth chip. It can also serve as a complete test board for the KT1025A or KT1025B chip.</p> <p>Actually, shipping directly in the form of chips is of course also supported. If you need chips, directly search for "KT1025A BT201".The chip's reference peripheral circuit design is included in the data package. It is very straightforward—just follow it directly. For details, carefully review the reference schematic.</p>
Question 3	3. The documentation has so much information, and I can't understand it. I just need a module that can connect to a phone to play music, support calls, and be controlled by buttons.
Answer	That's fine. The default functions of our module meet your needs. You can ignore the parts you don't understand, as our product is fully featured, and the multiple functions do not conflict with each other
Question 4	4. I don't need the call function. How can I handle this?
Answer	You can disable the call function via a serial port command. After the next power-up, Bluetooth will no longer support calls. The same applies to re-enabling it.
Question 5	5. I don't need the BLE data function. How can I disable it? I don't need the Bluetooth Audio. How can I handle this?
Answer	You can send an AT command to disable the BLE function. You also can sent AT command to disable the Bluetooth Audio. The chip supports dual-mode Bluetooth, with both modes being independent and non-interfering with each other.
Question 6	6. I don't need the music function, only BLE and SPP transparent transmission. How can I do this?
Answer	You can disable Bluetooth A2DP via a serial port AT command. The same applies to

	re-enabling it.
--	-----------------

11.KT1025A BT201 and KT1025B BT201 Packaging Dimensions and Parameter Explanation

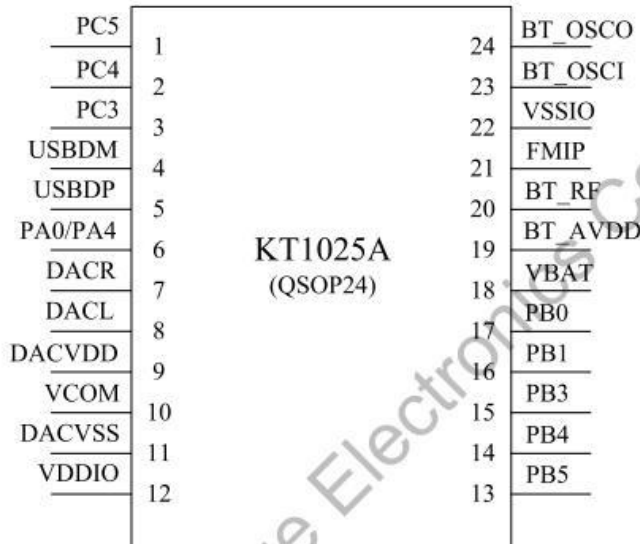


Figure 1-1 KT1025A_QSOP24 Package Diagram

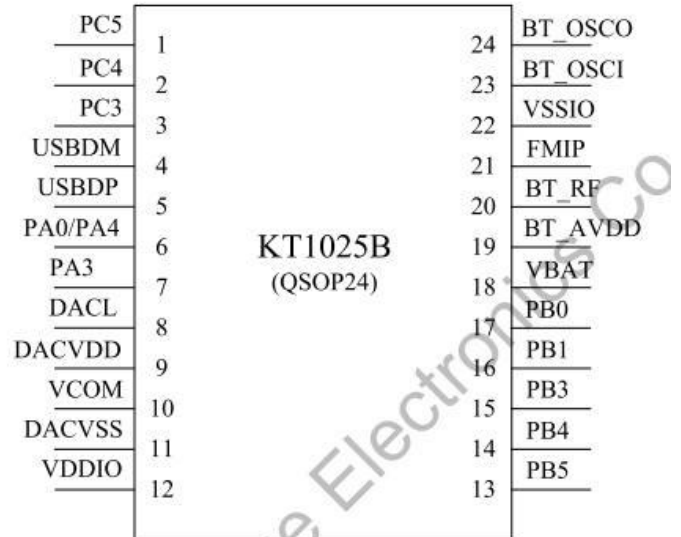


Figure 1-1 KT1025B_QSOP24 Package Diagram

Question 1	1. What is the difference between KT1025A and KT1025B? Why are there two models, and how should I choose?
Answer	<p>1. KT1025A supports stereo output and is priced higher by the manufacturer. It is recommended for users who are not cost-sensitive, as it has a more stable supply.</p> <p>2. KT1025B does not support stereo output and is positioned at a lower price point. The chip itself is identical to KT1025A, but it is more cost-effective.</p> <p>3. For users connecting only one speaker, KT1025B is recommended.</p> <p>4. The difference lies in pin 7 of the chip. If two speakers are to be connected, KT1025A is the only option.</p> <p>5. The unit price of KT1025A is slightly higher than KT1025B. Please consider this when selecting.</p>

Question 2	2. Why are KT1025A and KT1025B so affordable? Is there any performance compromise?
Answer	<p>1. The chips belong to the highest shipment volume category. Due to massive production and intense competition, costs are naturally lower.</p> <p>2. These chips are originally designed for products like Bluetooth speakers, earphones, and storytelling devices. High volume leads to low costs.</p> <p>3. The chips support both audio playback and data transmission functions, which is why we</p>

	<p>developed firmware to enable these features.</p> <p>4. There is no need to doubt the chip's performance. Knowledgeable customers are aware of this, and we already have significant adoption in automotive applications.</p> <p>5. Some may think, "If I only need BLE and not audio, can it be cheaper?" This is incorrect. Chips with more features have higher production volumes, leading to lower costs, though development costs are higher.</p>
--	---

Question 3	3. Which test module is for KT1025A BT201? Is a DEMO board provided?
Answer	We provide a test board of KT1025A BT201 chip - BT201 module, which is compatible with both KT1025A BT201 and KT1025B BT201 chips on the same test board. The firmware is fully compatible.

Question 4	4. What information is required for bulk procurement?
Answer	The corresponding checksum. This is critical and must be noted. It is typically indicated on the sample card.

12.Reference Program Examples

```

/*****
- Function Description: Send one byte via serial port
- Module Affiliation: External
- Parameter Description:
- Return Description:
- Notes:
    1. The serial port output method varies for different chips; this is our chip's method.
    2. You only need to modify and encapsulate this interface.
*****/
void func_send_byte( u8 dat )
{
    ctrl_uart_write(dat); /*Replace this interface*/
}

/*****
- Function Description: Send a string via serial port
- Module Affiliation: External
- Parameter Description:
- Return Description:
- Notes:
*****/
void func_UartPutStr (const char *Str )
{
    while ( *Str )
    {
        func_send_byte ( *Str );
        *Str++;
    }
}

/*****
- Function Description: Serial port receive processing --- Reference
- Module Affiliation: External
- Parameter Description:
- Return Description:
- Notes:
    1. Serial port reception uses an interrupt method; you must set a timeout counter and handle errors on timeout.
    2. Since all our commands end with 0x0D and 0x0A, detecting 0x0A indicates reception completion.
*****/
void uart_isr_rcv(u8 uto_buf)
{
    g_uart_flag.uart_timeout = UART_TIMEOUT; /*Generally 100ms*/
    g_uart_flag.uart_stop = 0;
    buf_uart1.buf[buf_uart1.index++] = uto_buf; /*Store character in buffer*/
    if(uto_buf == 0x0A) /*Receiving 0x0A indicates one frame of data is complete*/
    {
        g_uart_flag.uart_stop = 1; /*Set reception complete flag to 1*/
        g_uart_flag.uart_analysis = 1; /*Enable data analysis*/
        g_uart_flag.uart_timeout = 0;
        uart_rcv_ok_pro(buf_uart1.buf, buf_uart1.index); /*Process after reception is complete*/

        /*Add your own application here*/
    }
}

```



```
/******  
- Function Description: main  
- Module Affiliation: External  
- Parameter Description:  
- Return Description:  
- Notes:  
*****/  
void main()  
{  
    uart_init() ;/*Serial port initialization*/  
    delay_2ms(500) ;/*Delay 1 second*/  
    func_UartPutStr("AT+CA30\r\n") ;/*Set volume to level 30*/  
    delay_2ms(500) ;/*Delay 1 second*/  
    func_UartPutStr("AT+CA15\r\n") ;/*Set volume to level 15*/  
    delay_2ms(500) ;/*Delay 1 second*/  
    func_UartPutStr("AT+BD12354678\r\n") ;/*Set Bluetooth name to 12345678*/  
    delay_2ms(500) ;/*Delay 1 second*/  
    func_UartPutStr("AT+CT02\r\n") ;/*Set baud rate to 19200 -- Refer to manual*/  
    delay_2ms(500) ;/*Delay 1 second*/  
  
    while(1) ;  
}
```


13.Explanation of Modifying Prompt Sounds

1. The chip currently supports 5 internal prompt sounds:

Music mode	music.mp3
Bluetooth	bt.mp3
Connection	connect.mp3
Connection	disconnect.mp3
Incoming call	ring.mp3

2. To replace prompt sounds, please provide the above files. The files must be compressed.

3. The total size of the 5 files must not exceed 17KB, as this is limited by the chip's internal storage space.

4. Adjust the volume of the prompt sounds using software as needed. You can test them by playing them on a TF card to check if the volume is appropriate.

14.BQB Certification and FCC Testing Explanation

14.1 Bluetooth BQB Certification Explanation

- | |
|--|
| 1. The BQB certificate for the original chip has already been obtained, so there's no need to worry about BQB certification issues. We will thoroughly compile the documentation and release it. |
| 2. The release will be in the form of a separate document, along with an explanatory video, certification-related certificates, and tools. |

To be continued...

14.2 Bluetooth FCC Fixed-Frequency Testing Explanation

- | |
|--|
| 1. For FCC fixed-frequency testing, the original manufacturer provides dedicated software, using a USB-to-TTL connection method. |
| 2. The serial port uses the two USB ports of our chip.
(1) Therefore, when designing hardware, you must reserve pads for these two IO ports.
Recommendation: VCC, TX, RX, GND (four pins).
(2) Additionally, for FCC certification, the chip needs to be programmed with a separate program, which we will also provide to you. |
| 3. The release will be in the form of a separate document, along with an explanatory video, certification-related certificates, and tools. |

To be continued...

15.Detailed Statements

■ Development Prerequisites

Qingyue Electronics series products will provide comprehensive development templates, drivers, and application documentation to facilitate user experience. However, users are expected to be familiar with the hardware platform of their designed products and have knowledge of C programming.

■ EMI and EMC

The mechanical structure of Qingyue Electronics series chips and modules inherently result in EMI performance differences compared to integrated circuit designs. The EMI performance of Qingyue Electronics series chips meets the requirements of most applications. If users have special requirements, they must consult with us in advance.

The EMC performance of Qingyue Electronics series chips and modules are closely related to the design of the user's baseboard, especially the power circuit, I/O isolation, and reset circuit. Users must fully consider these factors when designing the baseboard. We strive to improve the electromagnetic compatibility of Qingyue Electronics series chips and modules but do not provide any guarantees for the EMC performance of the user's final application products.

■ Right to Modify Documentation

Shenzhen Qingyue Electronics Co., Ltd. reserves the right to modify the documentation related to Qingyue Electronics series products at any time without prior notice.

■ ESD (Electrostatic Discharge) Protection

Some components of Qingyue Electronics series products have built-in ESD protection circuits. However, in harsh environments, it is recommended that users implement additional ESD protection measures in the baseboard design, especially for power and I/O circuits, to ensure stable operation.

When installing Qingyue Electronics series products, ensure to discharge static electricity from your body for safety, such as by wearing a properly grounded anti-static wristband or touching a grounded metal object like a water pipe.

16.Version History

Version	Date	Reason
V1.0	2018/04/27	1. Initial version release: BLE transparent transmission OK, SPP transparent transmission OK, serial control OK.
V1.4	2018/11/01	1. Added serial port debugging assistant test instructions (see Chapter 9). 2. Added KT1025A and KT1025B descriptions (see Chapter 11).
V1.5	2018/11/12	1. Added music playback and specific track playback (see 5.1.5). 2. Added music playback with folder loop (see 5.1.6). 3. Added Bluetooth specified number dialing and incoming call number retrieval (see 6.1.3 and 6.2.2). 4. Added boot option to prioritize online device playback (see 4.2.5). 5. Added option to disable Bluetooth 0.5s status return (see 4.2.6). 6. Added music status query, time query, playback sequence query, and music playback mode settings. 7. Added device insertion/removal notifications and music playback completion notifications.
V1.6	2018/11/15	1. Added long filename query and return for audio files. 2. Added music playback time return.
V1.7	2018/11/20	1. Added BLE enable/disable and EDR enable/disable (see 6.1.3). 2. Added BLE status query and a UUID for mobile control of Bluetooth chip (see 7.5). 3. Modified Bluetooth status active return method (see 6.2.1 and 6.2.2).
V1.8	2018/11/30	1. Added user-defined UUID modification (see 7.6). Fixed known bugs. 2. Added recording function and file deletion function (see 5.1.9). Added prompt tone disable interface. 3. Changed default serial port baud rate to 115200.
V1.9	2018/12/29	1. Fixed DAC high configuration bug. Added error return information and improved documentation.
V2.0	2019/01/05	1. Added command to prevent Bluetooth background auto-switching (see 4.2.11). 2. Added Bluetooth-related control commands (see 6.1.5). 3. Fixed descriptive errors in documentation.
V2.1	2019/03/24	1. Added AT+C? specification (see 4.2.13). Fixed descriptive errors in documentation. 2. Minor changes, backward compatible.
V2.2	2019/04/26	1. Added custom Bluetooth broadcast packet (see 7.8).

		<p>2. Added serial port output for reading specified files (see 5.1.11 and 5.1.10).</p> <p>3. Added file deletion for TF card or U disk (see 5.1.12).</p> <p>4. Fixed descriptive bugs in documentation. Firmware remains compatible with previous versions.</p>
V2.3	2019/05/07	<p>1. Added specified folder and filename recording (see 5.1.13) – main updates for recording and TXT reading.</p> <p>2. Improved library packaging, added KT1025A and KT1025B datasheets and packaging references.</p> <p>3. Added detailed mode explanation (see 4.3.2).</p> <p>4. Added Bluetooth MAC address setting and detailed explanation (see 6.1.6). Fixed descriptive bugs in documentation.</p>
V2.4	2020/04/17	<p>1. Fixed some descriptive errors in documentation.</p> <p>2. No functional changes, no update required, backward compatible.</p>

17. Qingyue Bluetooth Series Product Recommendation

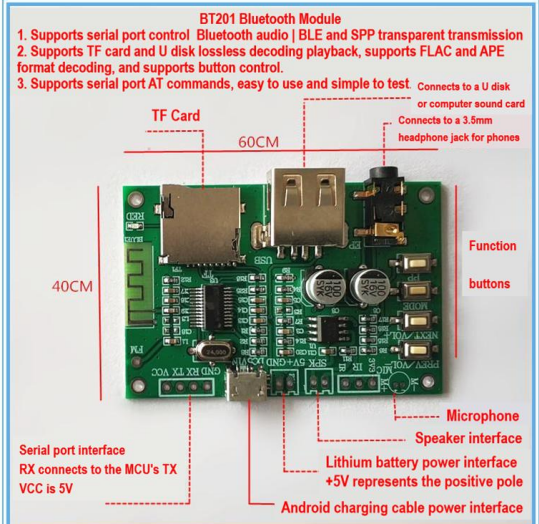
Solution	Delivery Form	Features
BT201 Bluetooth Solution	Chip IC: KT1025A-QSOP24 KT1025B-QSOP24 BT201 Module Stereo	1. Bluetooth audio, TF card/U disk playback, Bluetooth BLE and SPP transparent transmission. Serial port AT command control. Application Scenarios: -- Main promoted product (1) Bluetooth dual-mode data transmission, BLE applications. (2) Bluetooth audio + data transmission scenarios. Suitable for: Audio + data transmission applications.
BT321F Bluetooth Transmission Solution	Chip IC: KT1025A-QSOP24 KT1025B-QSOP24 BT321 Module Stereo	1. Bluetooth audio stereo transmission to speakers or earphones. 2. Transmission sources support U disk, TF card, AUX, SPI Flash. Serial port AT command operation. Application Scenarios: -- Main promoted product (1) Bluetooth transmission + voice prompt products. (2) Can transmit SPI Flash audio. Suitable for: Adult sound products and similar applications. (3) Suitable for any product transmitting audio to Bluetooth earphones.
BT301 Bluetooth Solution Supports SPI Flash and AUX, FM, Bluetooth	Chip IC: KT1025A-QSOP24 KT1025B-QSOP24 BT301 Module Stereo	1. Bluetooth audio, TF card/U disk playback, Bluetooth dual-mode transparent transmission. 2. Supports Flash playback, USB audio download. SPI Flash and TF card are mutually exclusive (choose one). 3. Supports AUX input, FM function [requires separate firmware]. Application Scenarios: (1) Bluetooth data transmission + voice prompt products. (2) Highlight: Supports SPI Flash playback. Suitable for: Voice broadcasting + Bluetooth data transmission.
BT401 Bluetooth Module	Module: BT401 Module	1. Supports Bluetooth dual-mode data transmission (BLE and SPP transparent transmission). Audio DAC output, I2S master/slave output. 2. Module with BQB certification, delivered as a module. Suitable for module-based requirements. 3. Also suitable for Bluetooth MIDI, OBD, printers, and similar applications. Application Scenarios: Bluetooth data transmission + voice

prompt products. Supports SPI Flash playback.

1. If you are familiar with Bluetooth and audio, and have good PCB layout design skills, we recommend using a chip directly. You can choose one of the following solutions: BT201, BT301, or BT321F. Then, purchase the firmware chip corresponding to the selected solution.
2. If you are not familiar with Bluetooth and don't want to learn about it, we recommend buying a module to save trouble. Designing your own solution can lead to many issues, such as stability and noise, which can be a headache. Therefore, we suggest purchasing a BT401 module directly for convenience.

For detailed documentation, please download the document package for the corresponding solution. For some niche applications not covered in the documentation, you can contact us directly.


17.1 BT201 Bluetooth Dual-Mode [Audio + Data Transmission BLE + SPP] Solution

	<table border="1"> <thead> <tr> <th>BT201 Functional Introduction</th> </tr> </thead> <tbody> <tr> <td>1. Supports Bluetooth audio, BLE data transmission, U disk playback, TF card playback, and SPI Flash. Supports MP3, WAV, FLAC, APE format decoding.</td> </tr> <tr> <td>2. Serial port AT command control. Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.</td> </tr> <tr> <td>3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict.</td> </tr> <tr> <td>4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.</td> </tr> <tr> <td>5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.</td> </tr> <tr> <td>6. Supports 128Kbps recording function. Records to TF card or U disk, with options for recording deletion and playback.</td> </tr> <tr> <td>7. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.</td> </tr> <tr> <td>8. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors.</td> </tr> <tr> <td>9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control.</td> </tr> <tr> <td>10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.</td> </tr> </tbody> </table>	BT201 Functional Introduction	1. Supports Bluetooth audio, BLE data transmission, U disk playback, TF card playback, and SPI Flash. Supports MP3, WAV, FLAC, APE format decoding.	2. Serial port AT command control. Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.	3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict.	4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.	5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.	6. Supports 128Kbps recording function. Records to TF card or U disk, with options for recording deletion and playback.	7. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.	8. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors.	9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control.	10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.
BT201 Functional Introduction												
1. Supports Bluetooth audio, BLE data transmission, U disk playback, TF card playback, and SPI Flash. Supports MP3, WAV, FLAC, APE format decoding.												
2. Serial port AT command control. Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.												
3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict.												
4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.												
5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.												
6. Supports 128Kbps recording function. Records to TF card or U disk, with options for recording deletion and playback.												
7. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.												
8. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors.												
9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control.												
10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.												

Netdisk Download Link:

https://drive.google.com/drive/folders/1c62OWCPjXJdfmNzeRPjZQ-WuzVOdRFnb?usp=drive_link


17.2 BT321F Bluetooth Transmission Solution

 <p>Bluetooth Transmission and Reception Two-in-One Stereo Solution Supports serial port AT command control</p> <p>深圳市清月电子有限公司 SHENZHEN QINYUE ELECTRONICS CO., LTD</p> <p>Supports computer, TF card, U disk, and SPI Flash devices for audio source transmission Supports random search and specified device connection</p>	<table border="1"> <thead> <tr> <th>BT321F Functional Introduction</th> </tr> </thead> <tbody> <tr> <td>1. Supports Bluetooth audio stereo transmission and reception, two-in-one. Bluetooth transmission supports connecting to slave devices like Bluetooth earphones and speakers.</td> </tr> <tr> <td>2. Bluetooth reception supports connecting to master devices like phones, tablets, and computers, along with common control functions.</td> </tr> <tr> <td>3. Bluetooth transmission supports audio sources: AUX input, U disk, TF card, computer PC sound card, and SPI Flash storage.</td> </tr> <tr> <td>4. Bluetooth transmission audio delay: less than 300ms. Supports serial port AT command control.</td> </tr> <tr> <td>5. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.</td> </tr> <tr> <td>6. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors.</td> </tr> <tr> <td>7. Bluetooth transmission supports: random search on power-up, list search [same effect as phone], and connection by specified address or name.</td> </tr> <tr> <td>8. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.</td> </tr> <tr> <td>9. Supports wireless serial port data transparent transmission between master and slave, with configurable baud rate. Also supports serial port AT command control.</td> </tr> </tbody> </table>	BT321F Functional Introduction	1. Supports Bluetooth audio stereo transmission and reception, two-in-one. Bluetooth transmission supports connecting to slave devices like Bluetooth earphones and speakers.	2. Bluetooth reception supports connecting to master devices like phones, tablets, and computers, along with common control functions.	3. Bluetooth transmission supports audio sources: AUX input, U disk, TF card, computer PC sound card, and SPI Flash storage.	4. Bluetooth transmission audio delay: less than 300ms. Supports serial port AT command control.	5. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.	6. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors.	7. Bluetooth transmission supports: random search on power-up, list search [same effect as phone], and connection by specified address or name.	8. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.	9. Supports wireless serial port data transparent transmission between master and slave, with configurable baud rate. Also supports serial port AT command control.
BT321F Functional Introduction											
1. Supports Bluetooth audio stereo transmission and reception, two-in-one. Bluetooth transmission supports connecting to slave devices like Bluetooth earphones and speakers.											
2. Bluetooth reception supports connecting to master devices like phones, tablets, and computers, along with common control functions.											
3. Bluetooth transmission supports audio sources: AUX input, U disk, TF card, computer PC sound card, and SPI Flash storage.											
4. Bluetooth transmission audio delay: less than 300ms. Supports serial port AT command control.											
5. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.											
6. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors.											
7. Bluetooth transmission supports: random search on power-up, list search [same effect as phone], and connection by specified address or name.											
8. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.											
9. Supports wireless serial port data transparent transmission between master and slave, with configurable baud rate. Also supports serial port AT command control.											

Netdisk Download Link:

https://drive.google.com/drive/folders/1Jan5is-bbSlcQWJzRppC1Gs-E IVQCM6?usp=drive_link

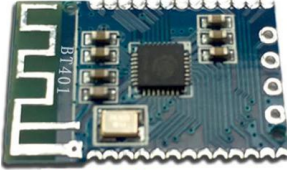
17.3 BT301 Bluetooth [Audio + Data Transmission BLE + SPI Flash] Solution

 <p>深圳市清月电子有限公司 SHENZHEN QINYUE ELECTRONICS CO., LTD</p> <p>Supports SPI Flash, AUX, and BLE/SPP Bluetooth data transmission Supports U disk and TF card, and serial port control</p>	<table border="1"> <thead> <tr> <th>BT301 Solution Functional Introduction</th> </tr> </thead> <tbody> <tr> <td>1. Supports Bluetooth audio, BLE data transmission, U disk playback, TF card playback, and SPI Flash. Supports MP3, WAV, FLAC, APE format decoding.</td> </tr> <tr> <td>2. Serial port AT command control. Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.</td> </tr> <tr> <td>3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict.</td> </tr> <tr> <td>4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.</td> </tr> <tr> <td>5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.</td> </tr> <tr> <td>6. Supports USB virtual SPI Flash as a U disk for copying audio, very convenient for debugging. The chip's built-in storage can also hold some audio.</td> </tr> <tr> <td>7. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.</td> </tr> <tr> <td>8. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors to form a minimal system. Supports remote firmware upgrades.</td> </tr> <tr> <td>9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control and voice prompts.</td> </tr> <tr> <td>10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.</td> </tr> </tbody> </table>	BT301 Solution Functional Introduction	1. Supports Bluetooth audio, BLE data transmission , U disk playback, TF card playback, and SPI Flash . Supports MP3, WAV, FLAC, APE format decoding.	2. Serial port AT command control . Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.	3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict.	4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.	5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.	6. Supports USB virtual SPI Flash as a U disk for copying audio, very convenient for debugging. The chip's built-in storage can also hold some audio.	7. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.	8. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors to form a minimal system. Supports remote firmware upgrades.	9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control and voice prompts.	10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.
BT301 Solution Functional Introduction												
1. Supports Bluetooth audio, BLE data transmission , U disk playback, TF card playback, and SPI Flash . Supports MP3, WAV, FLAC, APE format decoding.												
2. Serial port AT command control . Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.												
3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict.												
4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.												
5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.												
6. Supports USB virtual SPI Flash as a U disk for copying audio, very convenient for debugging. The chip's built-in storage can also hold some audio.												
7. Rich usage materials, including reference PCB, usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.												
8. Ultra-low cost, with competitive pricing for large quantities, and a very simple peripheral circuit requiring only 5 capacitors to form a minimal system. Supports remote firmware upgrades.												
9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control and voice prompts.												
10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.												

Netdisk Download Link:

https://drive.google.com/drive/folders/12Jen0WfYDkQruFGsyjnj1LsAH92pmiO?usp=drive_link

17.4 BT401 Bluetooth Audio [Supports All Functions] Solution



Bluetooth Audio and BLE Dual-Mode Data Transmission

Size: 23*15.5mm

- ▶ TF card and U disk lossless playback
- ▶ Bluetooth audio
- ▶ Bluetooth calls
- ▶ Bluetooth BLE transparent transmission
- ▶ Serial port AT commands
- ▶ Serial port AT command settings for baud rate, Bluetooth name, and MAC address
- ▶ Supports retrieving complete song names Supports external audio input (AUX)
- ▶ Supports I2S digital audio output and DAC analog output, highly flexible

BT401 Module Functional Introduction
1. Supports Bluetooth audio, BLE data transmission, U disk playback, TF card playback, and SPI Flash. Supports MP3, WAV, FLAC, APE format decoding.
2. Serial port AT command control. Supports modifying baud rate, Bluetooth name, specifying device playback, and phone data transparent transmission.
3. Supports BLE and SPP dual-mode data transmission while simultaneously playing Bluetooth audio without conflict. The module supports I2S as a master or slave.
4. Supports Bluetooth calls, multi-button control (previous track, next track, play/pause, volume adjustment, etc.) for simple functions.
5. Supports retrieving the full filename of the played file, serial port-specified path playback, and loop or single playback.
6. Supports 128Kbps recording function. Records to TF card or U disk, with options for recording deletion and playback.
7. Rich usage materials, including usage video tutorials, phone transparent transmission video tutorials, and serial port AT video tutorials.
8. Ultra-low cost, with standard firmware modules costing below 5.X for ultra-large batches.
9. Easily integrates with existing apps or supports development of WeChat mini-programs for phone control.
10. Functional firmware is developed in-house, highly flexible, and open to discussion for developing features that do not meet current needs.

Netdisk Download Link:

https://drive.google.com/drive/folders/1QD7j4i9X_mTUXQqpZ-9S1MSmpFrfs5J8?usp=drive_link