



UM1006

RTL8721Dx Bluetooth User Manual

Rev. 1.1

Sept., 2023



Realtek Semiconductor Corp.

No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211. Fax: +886-3-577-6047

www.realtek.com

COPYRIGHT

©2023 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

DISCLAIMER

Please Read Carefully:

Realtek Semiconductor Corp., (Realtek) reserves the right to make corrections, enhancements, improvements and other changes to its products and services. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Reproduction of significant portions in Realtek data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Realtek is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions.

Buyers and others who are developing systems that incorporate Realtek products (collectively, "Customers") understand and agree that Customers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Customers have full and exclusive responsibility to assure the safety of Customers' applications and compliance of their applications (and of all Realtek products used in or for Customers' applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Customer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Customer agrees that prior to using or distributing any applications that include Realtek products, Customer will thoroughly test such applications and the functionality of such Realtek products as used in such applications.

Realtek's provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation kits, (collectively, "Resources") are intended to assist designers who are developing applications that incorporate Realtek products; by downloading, accessing or using Realtek's Resources in any way, Customer (individually or, if Customer is acting on behalf of a company, Customer's company) agrees to use any particular Realtek Resources solely for this purpose and subject to the terms of this Notice.

Realtek's provision of Realtek Resources does not expand or otherwise alter Realtek's applicable published warranties or warranty disclaimers for Realtek's products, and no additional obligations or liabilities arise from Realtek providing such Realtek Resources. Realtek reserves the right to make corrections, enhancements, improvements and other changes to its Realtek Resources. Realtek has not conducted any testing other than that specifically described in the published documentation for a particular Realtek Resource.

Customer is authorized to use, copy and modify any individual Realtek Resource only in connection with the development of applications that include the Realtek product(s) identified in such Realtek Resource. No other license, express or implied, by estoppel or otherwise to any other Realtek intellectual property right, and no license to any technology or intellectual property right of Realtek or any third party is granted herein, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Realtek products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of Realtek Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from Realtek under the patents or other Realtek's intellectual property.

Realtek's Resources are provided "as is" and with all faults. Realtek disclaims all other warranties or representations, express or implied, regarding resources or use thereof, including but not limited to accuracy or completeness, title, any epidemic failure warranty and any implied warranties of merchantability, fitness for a particular purpose, and non-infringement of any third party intellectual property rights.

Realtek shall not be liable for and shall not defend or indemnify Customer against any claim, including but not limited to any infringement claim that related to or is based on any combination of products even if described in Realtek Resources or otherwise. In no event shall Realtek be liable for any actual, direct, special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of Realtek's Resources or use thereof, and regardless of whether Realtek has been advised of the possibility of such damages. Realtek is not responsible for any failure to meet such industry standard requirements.

Where Realtek specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Customers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may not use any Realtek products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death. Such equipment includes, without limitation, all medical devices identified by the U.S.FDA as Class III devices and equivalent classifications outside the U.S.

Customers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Customers' own risk. Customers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection.

Customer will fully indemnify Realtek and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

USING THIS DOCUMENT

This document is intended for the software engineer's reference and provides detailed programming information. Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide

Content

Content	4
1 Introduction.....	6
2 SDK Configuration.....	6
2.1 GCC Project.....	6
2.1.1 BT Compilation Configuration.....	6
2.1.2 BT API Configuration.....	6
3 ATCMDs	7
3.1 BT Example AT Commands	7
3.1.1 ATBE=peripheral - Start or stop the peripheral example	7
3.1.2 ATBE=central - Start or stop the central example	7
3.1.3 ATBE=scatternet - Start or stop the scatternet example.....	7
3.2 LE GAP AT Commands.....	7
3.2.1 ATBC=le_gap,version - Get device's version	7
3.2.2 ATBC=le_gap,addr - Get device's bd_addr.....	7
3.2.3 ATBC=le_gap,adv_data - Set adv data	8
3.2.4 ATBC=le_gap,scan_rsp - Set scan response data	8
3.2.5 ATBC=le_gap,adv - Start adv with adv param or stop adv	8
3.2.6 ATBC=le_gap,rand_addr - Set random address in controller	9
3.2.7 ATBC=le_gap,channel - Set host channel classification.....	9
3.2.8 ATBC=le_gap,name - Set device name.....	9
3.2.9 ATBC=le_gap,appear - Set device appearance	9
3.2.10 ATBC=le_gap,pre_conn_param - Set device preferred connection parameters	9
3.2.11 ATBC=le_gap,conn - Initiate a connection to remote device	10
3.2.12 ATBC=le_gap,get_active_conn - Get active connection number and their conn_handle	10
3.2.13 ATBC=le_gap,get_conn_handle - Get conn_handle by peer address	10
3.2.14 ATBC=le_gap,mtu_size - Get connection MTU size.....	10
3.2.15 ATBC=le_gap,disconn - Terminate an existing connection	11
3.2.16 ATBC=le_gap,dev_state - Get device's state	11
3.2.17 ATBC=le_gap,conn_rssi - Read connection RSSI	11
3.2.18 ATBC=le_gap,conn_info - Get connection information.....	11
3.2.19 ATBC=le_gap,conn_update - Update connection parameters	11
3.2.20 ATBC=le_gap,conn_phy - Set connection PHY preferences	11
3.2.21 ATBC=le_gap,conn_data_len - Set connection data length	12
3.2.22 ATBC=le_gap,whl_add - Add a device to whitelist.....	12
3.2.23 ATBC=le_gap,whl_remove - Remove a device from whitelist	12
3.2.24 ATBC=le_gap,whl_clear - Clear all devices in whitelist	12
3.2.25 ATBC=le_gap,scan_param - Set scan parameters	12
3.2.26 ATBC=le_gap,scan_info_filter - Enable or disable scan information filter	13
3.2.27 ATBC=le_gap,scan - Start or stop scanning	13
3.2.28 ATBC=le_gap,sec_param - Set pairing request parameters	13
3.2.29 ATBC=le_gap,sec - Start pairing	14
3.2.30 ATBC=le_gap,pair_cfm - Confirm to continue pairing if pairing method is just work	14
3.2.31 ATBC=le_gap,auth_key - Input passkey to continue pairing if pairing method is passkey entry.....	14
3.2.32 ATBC=le_gap,auth_keycfm - Confirm passkey match correct if pairing method is numerical comparison	14
3.2.33 ATBC=le_gap,auth_oob - Input oob TK in legacy pairing	14
3.2.34 ATBC=le_gap,bond_num - Get device number in bond list	14
3.2.35 ATBC=le_gap,bond_info - Get bond device information in bond list	15
3.2.36 ATBC=le_gap,bond_del - Delete a device in bond list by address.....	15
3.2.37 ATBC=le_gap,bond_clear - Clear all devices in bond list.....	15
3.2.38 ATBC=le_gap,eadv - Operate extended adv.....	15
3.2.39 ATBC=le_gap,eadv_data - Set extended adv data.....	16
3.2.40 ATBC=le_gap,escan_rsp - Set extended scan response data	16
3.2.41 ATBC=le_gap,escan_param - Set extended scan parameters	16
3.2.42 ATBC=le_gap,escan - Enable or disable extended scan	17
3.2.43 ATBC=le_gap,econn - Initiate a connection to remote device via extended adv	17

3.3	GATT AT Commands	18
3.3.1	ATBC=gatts,notify - Send notification data to remote client	18
3.3.2	ATBC=gatts,indicate - Send indication data to remote client	18
3.3.3	ATBC=gattc,exch_mtu - Initiate ATT exchange MTU request to remote device	18
3.3.4	ATBC=gattc,discover - Discover services, characteristics or descriptors	18
3.3.5	ATBC=gattc,read - Read characteristic value from a server	19
3.3.6	ATBC=gattc,write - Write characteristic value to a server	19
3.3.7	ATBC=gattc,en_cccd - Enable server's cccd(notify/indicate)	19
3.3.8	ATBC=gattc,dis_cccd - Disable server's cccd(notify/indicate)	19
3.3.9	ATBC=gattc,<bas_cmd> - Used by bas client to execute service operations.....	20
3.3.10	ATBC=gattc,<gaps_cmd> - Used by gap service client to execute service operations.....	20
3.3.11	ATBC=gattc,<simple_ble_cmd> - Used by simple ble service client to execute services operations	20
4	APIs	21
	Revision History.....	22

1 Introduction

This document describes BT user manual for SDK, including SDK compilation configuration, ATCMDs, APIs and etc.

2 SDK Configuration

2.1 GCC Project

2.1.1 BT Compilation Configuration

In SDK, by default, BT function is not enabled, so if you need to enable it, you should enable several MENUCONFIG switch to open it. There is only one configuration, that is, BT application(AP) and BT stack(NP) are configured on the same core KM4.

Use "make menuconfig" in directory "project/realtek_amebaDplus_va0_example/GCC-RELEASE/auto_build", open "CONFIG_BT" menu and Select "Enable BT" as Figure 2-1.

```
[*] Enable BT
[ ] BT Advanced Feature (NEW)
[ ] BT Example Demo (NEW)
```

Figure 2-1 CONFIG_BT option

Some common ble examples: peripheral, central, scatternet are provided, you can select them in menuconfig. Following chapter 2.1.1 introduce, open bt configuration in menuconfig, then select the example you want in "BT Example Demo" as Figure 2-2. Only when examples are configured and compiled, you can use atcmd to start them.

```
[*] Enable BT
[ ] BT Advanced Feature (NEW)
[*] BT Example Demo (NEW)
[*] BLE Peripheral (NEW)
[*] BLE Central (NEW)
[*] BLE Scatternet (NEW)
```

Figure 2-2 Select example in "BT Example Demo"

After configuring menuconfig, use "make" in diectory "/auto_build" to build image for both KM4 and KM0.

You can also use "make" in directory "project/realtek_amebaDplus_va0_example/GCC-RELEASE/project_km4" to build image for KM4, and "make" in "project/realtek_amebaDplus_va0_example/GCC-RELEASE/project_km0" to build image for KM0.

2.1.2 BT API Configuration

BT API supported features are configured by macros in "CONFIG_PLATFORM_AMEBADPLUS" in file "bt_api_config.h".

For example, if you want to use api that related to extended adv and scan, you need to enable macro "RTK_BLE_5_0_AE_ADV_SUPPORT" and "RTK_BLE_5_0_AE_SCAN_SUPPORT" as Figure 2-3.

```
#if defined(RTK_BLE_5_0_SUPPORT) && RTK_BLE_5_0_SUPPORT
#define RTK_BLE_5_0_SET_PHYS_SUPPORT ..... 1
#define RTK_BLE_5_0_AE_ADV_SUPPORT ..... 1
#define RTK_BLE_5_0_AE_SCAN_SUPPORT ..... 1
#endif /* RTK_BLE_5_0_SUPPORT */
```

Figure 2-3 Config macro in "bt_api_config.h"

And these macros will also affect ble examples referred in chapter3.1. If exentend adv and scan are configured, hci cmds related to legacy adv and scan will not be used in these examples.

3 ATCMDs

When input the AT Commands, do not exist 'space' between comma and parameter, and the end of parameters shall also not exist 'space'. If commands' parameter type is "integer", then this parameter can be input with hex(with 0x or 0X) or decimal format. For example, "ATBC=le_gap,appear,0x40" can also be input as "ATBC=le_gap,appear,64".

The parameters in atcmds are surrounded by < >, and some parameters surrounded by { } means that they may be used or not. If exist despriction "{ } = none", it means no need to input the parameters in { }, and if exist "{ } = <param1>,<param2>", these 2 params are all needed. The parameter <conn_handle> should be input according to actual situation. In general, the first conn_handle of master is 16, slave is 24.

3.1 BT Example AT Commands

3.1.1 ATBE=peripheral - Start or stop the peripheral example

ATBE=peripheral,<op>		
Description	This command is used to start or stop the peripheral example.	
Parameter	op	0/1, integer num: 1-start peripheral 0-stop peripheral
Sample	ATBE=peripheral,1	

3.1.2 ATBE=central - Start or stop the central example

ATBE=central,<op>		
Description	This command is used to start or stop the central example.	
Parameter	op	0/1, integer num: 1-start central 0-stop central
Sample	ATBE=central,1	

3.1.3 ATBE=scatternet - Start or stop the scatternet example

ATBE=scatternet,<op>		
Description	This command is used to start or stop the scatternet example.	
Parameter	op	0/1, integer num: 1-start scatternet 0-stop scatternet
Sample	ATBE=scatternet,1	

3.2 LE GAP AT Commands

3.2.1 ATBC=le_gap,version - Get device's version

ATBC=le_gap,version		
Description	This command is used to get device's version.	
Parameter	None	
Sample	ATBC=le_gap,version	

3.2.2 ATBC=le_gap,addr - Get device's bd_addr

ATBC=le_gap,addr		
------------------	--	--

Description	This command is used to get device's bd_addr.	
Parameter	None	
Sample	ATBC=le_gap,addr	

3.2.3 ATBC=le_gap,adv_data - Set adv data

ATBC=le_gap,adv_data{,<data>}		
Description	This command is used to set adv data.	
Parameter	data	hexdata No 0x if param in { } is none, default adv data in atcmd will be used
Sample	ATBC=le_gap,adv_data or ATBC=le_gap,adv_data,020106040852544b	

3.2.4 ATBC=le_gap,scan_rsp - Set scan response data

ATBC=le_gap,scan_rsp{,<data>}		
Description	This command is used to set scan response data.	
Parameter	data	hexdata No 0x if param in { } is none, default scan rsp data in atcmd will be used
Sample	ATBC=le_gap,scan_rsp or ATBC=le_gap,scan_rsp,03194000	

3.2.5 ATBC=le_gap,adv - Start adv with adv param or stop adv

ATBC=le_gap,adv,<op>{,<type>,<own_addr_type>,<filter_policy>{,<peer_addr_type>,<peer_addr>{,<interval_min>,<interval_max>,<channel_map>}} }		
Description	This command is used to start adv with adv param or stop adv.	
Parameter	op	0/1, integer num: 1-start adv, if param in { } is none, default adv param in atcmd will be used 0-stop adv, all params in { } is none
	type	0-4, integer num: 0-Connectable and scannable undirected advertising 1-Connectable high duty cycle directed advertising 2-Scannable undirected advertising 3-Non connectable undirected advertising 4-Connectable low duty cycle directed advertising
	own_addr_type	0-3, integer num: 0-Public Device Address 1-Random Device Address 2-Resolvable Private Address or Public Address 3-Resolvable Private Address or Random Address
	filter_policy	0-3, integer num: 0-Allow both scan and connection requests from anyone 1-Allow both scan request from White List devices only and connection request from anyone 2-Allow both scan request from anyone and connection request from White List devices only 3-Allow scan and connection requests from White List devices only
	peer_addr_type	0-1, integer num: 0-Public Device Address or Public Identity Address 1-Random Device Address or Random (static) Identity Address
	peer_addr	hexdata No 0x, e.g. 00e04c0a30d1
	interval_min	0x20-0x4000, integer num
	interval_max	0x20-0x4000, integer num
	channel_map	0-7, integer num: bit0-Channel 37 shall be used bit1-Channel 38 shall be used bit2-Channel 39 shall be used

3.2.11 ATBC=le_gap,conn - Initiate a connection to remote device

ATBC=le_gap,conn,<peer_addr_type>,<peer_addr>{,<own_addr_type>,<filter_policy>{,<scan_interval>,<scan_window>,<conn_interval_min>,<conn_interval_max>,<conn_latency>,<supv_timeout>} }		
Description	This command is used to initiate a connection to remote device, if param in { } is none, default conn param in atcmd will be used	
Parameter	peer_addr_type	0-3, integer num: 0-public addr 1-random addr 2-public identity addr 3-random identity addr
	peer_addr	hexdata No 0x, e.g. 00e04c0a30c3
	own_addr_type	0-3, integer num: 0-Public Device Address 1-Random Device Address 2-Resolvable Private Address or Public Address 3-Resolvable Private Address or Random Address
	filter_policy	0/1, integer num: 0-whitelist is not used 1-whitelist is used
	scan_interval	0x4-0x4000, integer num
	scan_window	0x4-0x4000, integer num
	conn_interval_min	0x6-0xc80, integer num
	conn_interval_max	0x6-0xc80, integer num
	conn_latency	0x0-0x1f3, integer num
	supv_timeou	0xa-0xc80, integer num
Sample	ATBC=le_gap,conn,0,00e04c0a30c3 or ATBC=le_gap,conn,0,00e04c0a30c3,0,0 or ATBC=le_gap,conn,0,00e04c0a30c3,0,0,0x40,0x20,0x10,0x20,0x0,0x200	

3.2.12 ATBC=le_gap,get_active_conn - Get active connection number and their conn_handle

ATBC=le_gap,get_active_conn		
Description	This command is used to get active connection number and their conn_handle.	
Parameter	none	
Sample	ATBC=le_gap,get_active_conn	

3.2.13 ATBC=le_gap,get_conn_handle - Get conn_handle by peer address

ATBC=le_gap,get_conn_handle,<peer_addr_type>,<peer_addr>		
Description	This command is used to get conn_handle by peer address.	
Parameter	peer_addr_type	0-3, integer num: 0-Public Device Address 1-Random Device Address 2-Public Identity Address 3-Random (static) Identity Address
	peer_addr	hexdata No 0x, e.g. 00e04c0a30c3
Sample	ATBC=le_gap,get_conn_handle,0,00e04c23dd47	

3.2.14 ATBC=le_gap,mtu_size - Get connection MTU size

ATBC=le_gap,mtu_size,<conn_handle>		
Description	This command is used to get connection MTU size.	
Parameter	conn_handle	0-max, integer num
Sample	ATBC=le_gap,mtu_size,16	

3.2.15 ATBC=le_gap,disconn - Terminate an existing connection

ATBC=le_gap,disconn,<conn_handle>		
Description	This command is used to terminate an existing connection.	
Parameter	conn_handle	0-max, integer num
Sample	ATBC=le_gap,disconn,16	

3.2.16 ATBC=le_gap,dev_state - Get device's state

ATBC=le_gap,dev_state		
Description	This command is used to get device's state information.	
Parameter	None	
Sample	ATBC=le_gap,dev_state	

3.2.17 ATBC=le_gap,conn_rssi - Read connection RSSI

ATBC=le_gap,conn_rssi,<conn_handle>		
Description	This command is used to read connection RSSI.	
Parameter	conn_handle	0-max, integer num
Sample	ATBC=le_gap,conn_rssi,16	

3.2.18 ATBC=le_gap,conn_info - Get connection information

ATBC=le_gap,conn_info,<conn_handle>		
Description	This command is used to get connection information by conn_handle.	
Parameter	conn_handle	0-max, integer num
Sample	ATBC=le_gap,conn_info,16	

3.2.19 ATBC=le_gap,conn_update - Update connection parameters

ATBC=le_gap,conn_update,<conn_handle>,<conn_interval_min>,<conn_interval_max>,<conn_latency>,<timeout>		
Description	This command is used to update connection parameters.	
Parameter	conn_handle	0-max, integer num
	conn_interval_min	0x6-0xC80, integer num
	conn_interval_max	0x6-0xC80, integer num
	conn_latency	0x0-0x1F3, integer num
	timeout	0xA-0xC80, integer num
Sample	ATBC=le_gap,conn_update,16,0x10,0x100,0x02,0x1f4	

3.2.20 ATBC=le_gap,conn_phy - Set connection PHY preferences

ATBC=le_gap,conn_phy,<conn_handle>,<all_phy>,<tx_phy>,<rx_phy>,<phy_options>		
Description	This command is used to set connection PHY preferences.	
Parameter	conn_handle	0-max, integer num
	all_phy	0-3, integer num: bit0-has no preference among tx PHYs bit1-has no preference among rx PHYs
	tx_phy	0-7, integer num: bit0-has prefers to LE 1M tx PHY bit1-has prefers to LE 2M tx PHY bit2-has prefers to LE coded tx PHY
	rx_phy	0-7, integer num: bit0-has prefers to LE 1M rx PHY bit1-has prefers to LE 2M rx PHY

		bit2-has prefers to LE coded rx PHY
	phy_options	0-2, integer num: 0-has no prefer on coding when use LE Coded PHY 1-prefers S=2 coding when use LE Coded PHY 2-prefers S=8 coding when use LE Coded PHY
Sample	ATBC=le_gap,conn_phy,16,0,4,4,0	

3.2.21 ATBC=le_gap,conn_data_len - Set connection data length

ATBC=le_gap,conn_data_len,<conn_handle>,<max_tx_octets>,<max_tx_time>		
Description	This command is used to set connection data length.	
Parameter	conn_handle	0-max, integer num
	max_tx_octets	0x1B-0xFB, integer num
	max_tx_time	0x148-0x4290, integer num
Sample	ATBC=le_gap,conn_data_len,16,0x30,0x300	

3.2.22 ATBC=le_gap,wl_add - Add a device to whitelist

ATBC=le_gap,wl_add,<addr_type>,<addr>		
Description	This command is used to add device to whitelist.	
Parameter	addr_type	integer num: 0: public addr 1: random addr 0xFF: Devices sending anonymous advertisements
	addr	hexdata No 0x, e.g. 00e04c0a30c3
Sample	ATBC=le_gap,wl_add,0,00e04c0a30c3	

3.2.23 ATBC=le_gap,wl_remove - Remove a device from whitelist

ATBC=le_gap,wl_remove,<addr_type>,<addr>		
Description	This command is used to remove a device from whitelist.	
Parameter	addr_type	integer num: 0: public addr 1: random addr 0xFF: Devices sending anonymous advertisements
	addr	hexdata No 0x, e.g. 00e04c0a30c3
Sample	ATBC=le_gap,wl_remove,0x0,00e04c0a30c3	

3.2.24 ATBC=le_gap,wl_clear - Clear all devices in whitelist

ATBC=le_gap,wl_clear		
Description	This command is used to clear all devices in whitelist.	
Parameter	none	
Sample	ATBC=le_gap,wl_clear	

3.2.25 ATBC=le_gap,scan_param - Set scan parameters

ATBC=gap,scan_param{<type>,<own_addr_type>,<filter_policy>,<duplicate_opt>,<interval>,<window>}		
Description	This command is used to set scan parameters, if param in { } is none, default scan param in atcmd will be used.	
Parameter	type	0/1, integer num: 0-passive scan 1-active scan
	own_addr_type	0-3, integer num: 0-Public Device Address

		1-Random Device Address 2-Resolvable Private Address or Public Address 3-Resolvable Private Address or Random Address
	filter_policy	0-3, integer num: 0-Accept all adv, except directed adv not addressed to this device 1-Accept only adv from dev in whitelist, ignore directed adv not addressed to this device 2-Accept all adv, except directed adv where identity addr not addressed to this device 3-Accept all adv, except from dev not in whitelist and direct adv where identity addr not addressed to this device
	duplicate_opt	0/1, integer num: 1-enable duplicate filter 0-disable duplicate filter
	interval	0x4-0x4000, integer num
	window	0x4-0x4000, integer num
Sample	ATBC=le_gap,scan_param or ATBC=le_gap,scan_param,0,0,0,1 or ATBC=le_gap,scan_param,0,0,0,1,0x20,0x10	

3.2.26 ATBC=le_gap,scan_info_filter - Enable or disable scan information filter

ATBC=le_gap,scan,<op>{,<offset>,<filter_data>}		
Description	This command is used to enable or disable scan information filter, only when the filter data and its offset in scanned data matches the scan information, the scanned adv data will be reported.	
Parameter	op	0/1, integer num: 1-enable scan info filter 0-disable scan, param in { } is useless
	offset	integer num: the byte offset of the filter data in scanned information
	filter_data	string: the string that need to filter from scan information,
Sample	ATBC=le_gap,scan_info_filter,1,5,RTK_BT_PERIPHERAL or ATBC=le_gap,scan_info_fitler,0	

3.2.27 ATBC=le_gap,scan - Start or stop scanning

ATBC=le_gap,scan,<op>		
Description	This command is used to start or stop scanning.	
Parameter	op	0/1, integer num: 1-enable scan 0-disable scan
Sample	ATBC=le_gap,scan,1	

3.2.28 ATBC=le_gap,sec_param - Set pairing request parameters

ATBC=le_gap,sec_param{,<io_cap>,<oob_data_flag>,<bond>,<mitm>,<sec_pair>,<use_fixed_key>,<fixed_key>}		
Description	This command is used to set pairing request parameters, if param in { } is none, default pairing param in atcmd will be used.	
Parameter	io_cap	0-4, integer num: 0-DisplayOnly 1-DisplayYesNo 2-KeyBoardOnly 3-NoInputNoOutput
	oob_data_flag	0/1, integer num
	bond	0/1, integer num
	mitm	0/1, integer num
	sec_pair	0/1, integer num
	use_fixed_key	0/1, integer num
	fixed_key	0-999999, decimal num

Sample	ATBC=le_gap,sec_param or ATBC=le_gap,sec_param,3,0,1,0,0,0
--------	---

3.2.29 ATBC=le_gap,sec - Start pairing

ATBC=le_gap,sec,<conn_handle>		
Description	This command is used to start pairing.	
Parameter	conn_handle	0-max, integer num
Sample	ATBC=le_gap,sec,16	

3.2.30 ATBC=le_gap,pair_cfm - Confirm to continue pairing if pairing method is just work

ATBC=le_gap,pair_cfm,<conn_handle>,<op>		
Description	This command is used to confirm to continue pairing if pairing method is just work.	
Parameter	conn_handle	0-max, integer num
	op	0/1, integer num: 1-confirm continue just work pairing 0-reject pairing
Sample	e.g. ATBC=le_gap,pair_cfm,16,1	

3.2.31 ATBC=le_gap,auth_key - Input passkey to continue pairing if pairing method is passkey entry

ATBC=le_gap,auth_key,<conn_handle>,<passkey>		
Description	This command is used to input passkey to continue pairing if pairing method is passkey entry.	
Parameter	conn_handle	0-max, integer num
	passkey	0-999999, decimal num
Sample	ATBC=le_gap,auth_key,16,123456	

3.2.32 ATBC=le_gap,auth_keycfm - Confirm passkey match correct if pairing method is numerical comparison

ATBC=le_gap,auth_keycfm,<conn_handle>,<op>		
Description	This command is used to confirm passkey match correct if pairing method is numerical comparison.	
Parameter	conn_handle	0-max, integer num
	op	0/1, integer num 1-confirm passkey is correct 0-passkey is different, reject pairing
Sample	ATBC=le_gap,auth_keycfm,16,1	

3.2.33 ATBC=le_gap,auth_oob - Input oob TK in legacy pairing

ATBC=le_gap,auth_oob,<conn_handle>,<TK>		
Description	This command is used to input oob TK in legacy pairing.	
Parameter	conn_handle	0-max, integer num
	TK	16byte hexnum with 0x, e.g. 0x00112233445566778899aabbccddeeff
Sample	ATBC=le_gap,auth_oob,16,0x00112233445566778899aabbccddeeff	

3.2.34 ATBC=le_gap,bond_num - Get device number in bond list

ATBC=le_gap,bond_num		
Description	This command is used to get device num in bond list.	
Parameter	none	

Sample	ATBC=le_gap,bond_num
--------	----------------------

3.2.35 ATBC=le_gap,bond_info - Get bond device information in bond list

ATBC=le_gap,bond_info		
Description	This command is used to get bond device information in bond list.	
Parameter	none	
Sample	ATBC=le_gap,bond_info	

3.2.36 ATBC=le_gap,bond_del - Delete a device in bond list by address

ATBC=le_gap,bond_del,<addr_type>,<addr>		
Description	This command is used to delete a device in bond list by address.	
Parameter	addr_type	0/1, integer num: 0-public addr 1-random addr
	addr	hexdata No 0x, e.g. 00e04c0a30c3
Sample	ATBC=le_gap,bond_del,0,00e04c0a30c3	

3.2.37 ATBC=le_gap,bond_clear - Clear all devices in bond list

ATBC=le_gap,bond_clear		
Description	This command is used to clear all devices in bond list.	
Parameter	none	
Sample	ATBC=le_gap,bond_clear	

3.2.38 ATBC=le_gap,eadv - Operate extended adv

ATBC=le_gap,eadv,<op=0 or 3>,<adv_handle> ATBC=le_gap,eadv,<op=1>,<adv_handle>,<duration> ATBC=le_gap,eadv,<op=2>,<adv_sid>,<adv_prop>,<pri/sec PHY>,<filter_policy>,<tx_power>,<adv_interval_min>,<own_addr_type>,<own_rand_addr>,<peer_addr_type>,<peer_addr>		
Description	This command is used to operate extended adv. If param is not set, default eadv param in atcmd will be used	
Parameter	op	0-3, integer num: 0-stop an eadv 1-start an eadv 2-create a new eadv set with eadv param 3-remove an eadv
	adv_handle	0x00-0xEF, integer num
	adv_sid	0x00-0x0F, integer num
	adv_prop	0x0000-0xFFFF, integer num: bit0-connectable adv bit1-scannable adv bit2-directed adv bit3-high duty cycle directed connectable adv bit4-use legacy adv PDU bit5-omit advertiser's address from all PDUs bit6-include TxPower in the extended header of at least one adv PDU
	pri/sec PHY	hexnum with 0x: 0x11-pri/coded PHY=1M/1M 0x13-pri/coded PHY=1M/coded 0x31-pri/coded PHY=coded/1M 0x33-pri/code PHY=coded/coded
	filter_policy	0-3, integer num: 0-Allow both scan and connection requests from anyone 1-Allow both scan req from White List devices only and connection req from anyone

		2-Allow both scan req from anyone and connection req from White List devices only 3-Allow scan and connection requests from White List devices only
	tx_power	signed integer num: -127 to +20, dBm 0x7F-host has no preference
	adv_interval_min	0x20-0xFFFFF, integer num adv_interval_max is default set in atcmd, usually 0xa larger than adv_interval_min
	own_addr_type	0-3, integer num: 0-Public Device Address 1-Random Device Address 2-Resolvable Private Address or Public Address 3-Resolvable Private Address or Random Address
	own_rand_addr	hexdata No 0x, e.g. 00e04c0a30d1
	peer_addr_type	0/1, integer num: 0-Public Device Address or Public Identity Address 1-Random Device Address or Random (static) Identity Address
	peer_addr	hexdata No 0x, e.g. 00e04c0a30d1
Sample	ATBC=le_gap,eadv,2,0,0x0,0x11,0,-5,160 or ATBC=le_gap,eadv,2,0,0x4,0x11,0,-5,160,0,000000000000,0,00e04c8002a4 to create an eadv set ATBC=le_gap,eadv,1,0 to start eadv ATBC=le_gap,eadv,0,0 to stop eadv ATBC=le_gap,eadv,3,0 to remove an eadv set	

3.2.39 ATBC=le_gap,eadv_data - Set extended adv data

ATBC=le_gap,eadv_data,<adv_handle>,<data_len>		
Description	This command is used to set extended adv data.	
Parameter	adv_handle	0x00-0xEF, integer num
	data_len	integer num adv data is default set in atcmd, you can just decide to send how long of it
Sample	ATBC=le_gap,eadv_data,0,234	

3.2.40 ATBC=le_gap,escan_rsp - Set extended scan response data

ATBC=le_gap,escan_rsp,<adv_handle>,<data_len>		
Description	This command is used to set extended scan response data.	
Parameter	adv_handle	0x00-0xEF, integer num
	data_len	integer num scan response data is default set in atcmd, you can just decide to send how long of it
Sample	ATBC=le_gap,escan_rsp,0,251	

3.2.41 ATBC=le_gap,escan_param - Set extended scan parameters

ATBC=le_gap,escan_param,<own_addr_type>,<filter_policy>,<duplicate_opt>{,<duration>,<period>,<scan_phys>{,<scan_type>,<interval>,<window>}x2 }		
Description	This command is used to set extended scan parameters. If param in { } is none, default escan param in atcmd will be used	
Parameter	own_addr_type	0-3, integer num: 0-Public Device Address 1-Random Device Address 2-Resolvable Private Address or Public Address 3-Resolvable Private Address or Random Address
	filter_policy	0-3, integer num: 0-Accept all adv, except directed adv not addressed to this device 1-Accept only adv from dev in whitelist, ignore directed adv not addressed to this device 2-Accept all adv, except directed adv where identity addr not addressed to this device 3-Accept all adv, except from dev not in whitelist and direct adv where identity addr not addressed to this device

	duplicate_opt	0-2, integer num: 0-disable duplicate filter 1-enable duplicate filter 2-enable duplicate filter, reset for each scan period
	duration	0x1-0xFFFF, integer num
	period	0x1-0xFFFF, integer num
	scan_phys	integer num: bit0-scan on 1M PHY bit2-scan on Coded PHY
	scan_type	0/1, integer num: 0-passive scan 1-active scan
	interval	0x4-0xFFFF, integer num
	window	0x4-0xFFFF, integer num
Sample	ATBC=le_gap,escan,param,0,1,0,0,1	

3.2.42 ATBC=le_gap,escan - Enable or disable extended scan

ATBC=le_gap,escan,<op>		
Description	This command is used to enable or disable extended scan	
Parameter	op	0/1, integer num: 0-disable escan 1-enable escan
Sample	ATBC=le_gap,escan,1 to enable escan ATBC=le_gap,escan,0 to disable escan	

3.2.43 ATBC=le_gap,econn - Initiate a connection to remote device via extended adv

ATBC=le_gap,conn,<peer_addr_type>,<peer_addr>,<init_phys>,<own_addr_type>,<filter_policy>,<scan_interval>,<scan_window>,<conn_interval_min>,<conn_interval_max>,<conn_latency>,<supv_timeout>}		
Description	This command is used to initiate a connection to remote device, if param in { } is none, default conn param in atcmd will be used	
Parameter	peer_addr_type	0-3, integer num: 0-public addr 1-random addr 2-public identity addr 3-random identity addr
	peer_addr	hexdata No 0x, e.g. 00e04c0a30c3
	init_phys	integer num: bit0-if 1, adv primary channel is on 1M phy bit1-if 1, adv primary channel is on 2M phy bit2-if 1, adv primary channel is on coded phy
	own_addr_type	0-3, integer num: 0-Public Device Address 1-Random Device Address 2-Resolvable Private Address or Public Address 3-Resolvable Private Address or Random Address
	filter_policy	0/1, integer num: 0-whitelist is not used 1-whitelist is used
	scan_interval	0x4-0x4000, integer num
	scan_window	0x4-0x4000, integer num
	conn_interval_min	0x6-0xc80, integer num
	conn_interval_max	0x6-0xc80, integer num
	conn_latency	0x0-0x1f3, integer num
	supv_timeou	0xa-0xc80, integer num
Sample	ATBC=le_gap,econn,0,00e04c0a30c3,1 or ATBC=le_gap,econn,0,00e04c0a30c3,1,0,0 or ATBC=le_gap,econn,0,00e04c0a30c3,1,0,0,0x40,0x20,0x10,0x20,0x0,0x200	

3.3 GATT AT Commands

3.3.1 ATBC=gatts,notify - Send notification data to remote client

ATBC= gatts,notify,<conn_handle>,<service_app_id>,<charac_index>,<length>,<data>		
Description	This command is used to send notification data to remote client.	
Parameter	conn_handle	0-max, integer num
	service_app_id	the app id of service that characteristic value was in. 0-max, integer num
	charac_index	the index of characteristic value to be notified 0-max, integer num,
	length	length of notify data, integer num
	data	notify_data, hexdata No 0x
Sample	ATBC=gatts,notify,16,0,7,4,11223344	

3.3.2 ATBC=gatts,indicate - Send indication data to remote client

ATBC= gatts,indicate,<conn_handle>,<service_app_id>,<charac_index>,<length>,<data>		
Description	This command is used to send indication data to remote client.	
Parameter	conn_handle	0-max, integer num
	service_app_id	app id of service that characteristic value was in. 0-max, integer num
	charac_index	index of characteristic value to be indicated 0-max, integer num,
	length	length of indicate data, integer num
	data	indicate_data, hexdata No 0x
Sample	ATBC=gatts,indicate,16,0,10,4,44332211	

3.3.3 ATBC=gattc,exch_mtu - Initiate ATT exchange MTU request to remote device

ATBC=gattc, exch_mtu,<conn_handle>		
Description	This command is used to initiate ATT exchange MTU request to remote device	
Parameter	conn_handle	0-max, integer num
Sample	ATBC=gattc,exch_mtu,16	

3.3.4 ATBC=gattc,disc - Discover services, characteristics or descriptors

ATBC=gattc,disc,<conn_handle>,<type>{,<start_handle>,<end_handle>,<uuid16/uuid128>}		
Description	This command is used to discover services, characteristics or descriptors.	
Parameter	conn_handle	0-max, integer num
	type	0-5, integer num: 0-disc primary service, { } = none, 1-disc primary by uuid, { } = <uuid_16> 2-disc included service, { } = <start_handle>, <end_handle> 3-disc characteristic all, { } = <start_handle>, <end_handle> 4-disc characteristic by uuid, { } = <start_handle>,<end_handle>,<uuid16/uuid128> 5-disc descriptor all, { } = <start_handle>, <end_handle>
	start_handle	0x1-0xFFFF, integer num
	end_handle	0x1-0xFFFF, integer num, start_handle shall <= end_handle
	uuid16/uuid128	hexnum with 0x, e.g. 0x2a29
Sample	ATBC=gattc,disc,16,0 to discover all primary service ATBC=gattc,disc,16,1,0x1801 to discover primary service whose uuid is 0x1801 ATBC=gattc,disc,16,2,0x1,0xFFFF to discover included services from handle 0x1-0xFFFF ATBC=gattc,disc,16,3,0x1,0xFFFF to discover all characteristics from handle 0x1-0xFFFF ATBC=gattc,disc,16,4,0x1,0xFFFF,0x2a29 to discover characteristic whose uuid is 0x2a29 ATBC=gattc,disc,16,5,0x1,0xFFFF to discover all attributes from handle 0x1-0xFFFF	

3.3.5 ATBC=gattc,read - Read characteristic value from a server

ATBC=gattc,read,<conn_handle>,<type>{,...}		
Description	This command is used to read characteristic values from a server.	
Parameter	conn_handle	0-max, integer num
	type	integer num: 0-read charac value by handle, {...} = <handle> 1-read charac value by uuid, {...} = <start_handle>,<end_handle>,<uuid16/uuid128>
	handle	0x1-0xFFFF, integer num
	start_handle	0x1-0xFFFF, integer num
	end_handle	0x1-0xFFFF, integer num, start_handle shall <= end_handle
	uuid16/uuid128	hexnum with 0x, e.g. 0x2a29
Sample	ATBC=gattc,read,16,0,0x1e to read characteristic value whose handle is 0x1e ATBC=gattc,read,16,1,0x1,0xFFFF,0x2a29 to read characteristic value whose uuid is 0x2a29 from handle 0x1-0xFFFF	

3.3.6 ATBC=gattc,write - Write characteristic value to a server

ATBC=gattc,write,<conn_handle>,<type>,<handle>,<length>,<data>		
Description	This command is used to write characteristic value to a server.	
Parameter	conn_handle	0-max, integer num
	type	integer num: 0-write characteristic by handle 1-write characteristic by handle use write_no_rsp
	handle	0x1-0xFFFF, integer num
	length	length of writed data, integer num
	data	writed_data, hexdata No 0x
Sample	ATBC=gattc,write,16,0,0x7,6,525252525252 to write hexdata to characteristic value whose handle is 0x7 ATBC=gattc,write,16,1,0x3a,1,01 to write hexdata to characteristic value whose handle is 0x3a use attr_write_cmd(no response)	

3.3.7 ATBC=gattc,en_cccd - Enable server's cccd(notify/indicate)

ATBC=gattc,en_cccd,<conn_handle>,<charac_value_handle>,<cccd_handle>,<cccd_bit_mask>		
Description	This command is used to enable characteristic cccd of a server.	
Parameter	conn_handle	0-max, integer num
	charac_value_handle	0x1-0xFFFF, integer num
	cccd_handle	0x1-0xFFFF, integer num
	cccd_val	0-3, integer num: 1-notify enable 2-indicate enable 3-both enable 0-none is enabled
Sample	Please use 'ATBC=gattc,dis,...' first to discover <charac_value_handle> and <cccd_handle>, then use ATBC=gattc,en_cccd,..., otherwise param <charac_value_handle> and <cccd_handle> may be incorrect, and cmd 'en_cccd' may fail. ATBC=gattc,en_cccd,16,0x40,0x41,1 to enable cccd notify whose char_val_handle is 0x40 and cccd_handle is 0x41 ATBC=gattc,en_cccd,16,0x40,0x41,2 to enable cccd indicate	

3.3.8 ATBC=gattc,dis_cccd - Disable server's cccd(notify/indicate)

ATBC=gattc,dis_cccd,<conn_handle>,<charac_value_handle>,<cccd_handle>,<cccd_bit_mask>		
Description	This command is used to disable characteristic cccd of a server.	
Parameter	conn_handle	0-max, integer num

	charac_value_handle	0x1-0xFFFF, integer num
	cccd_handle	0x1-0xFFFF, integer num
	cccd_val	0-3, integer num: 1-notify disable 2-indicate disable 3-both disable 0-none is disabled
Sample	ATBC=gattc,dis_cccd,16,0x40,0x41,1 to disable cccd notify whose char_val_handle is 0x40 and cccd_handle is 0x40 ATBC=gattc,dis_cccd,16,0x40,0x41,2 to disable cccd indicate	

3.3.9 ATBC=gattc,<bas_cmd> - Used by bas client to execute service operations

ATBC=gattc,<bas_cmd>,<conn_handle>,<en/disable>		
Description	This command is used by bas client to execute services operations.	
Parameter	bas_cmd	String: bas_disc - bas client discover all services, { } = none (when use ble_mgr lib, don't need this cmd to discover) bas_read_bat_lev - bas client read battery level value, { } = none bas_set_notify - bas client set battery level notify bit, param <en/disable> is needed
	conn_handle	0-max, integer num
	en/disable	0/1, integer num: 1 - enable 0 - disable
Sample	ATBC=gattc,bas_disc,16 to discover bas service related information ATBC=gattc,bas_read_bat_lev,16 to read battery level value ATBC=gattc,bas_set_notify,16,1 to set battery level cccd notify enable ATBC=gattc,bas_set_notify,16,0 to set battery level cccd notify disable	

NOTE

ATBC=gattc,bas_disc,... shall be executed first before other bas_cmds.

3.3.10 ATBC=gattc,<gaps_cmd> - Used by gap service client to execute service operations

ATBC=gattc,<gaps_cmd>,<conn_handle>,<char_index>		
Description	This command is used by gap service client to execute service operations.	
Parameter	gaps_cmd	String: gaps_disc - gaps client discover service information, { } = none, (when use ble_mgr lib, don't need this cmd to discover) gaps_read_char - gaps client read characteristic value, param <char_index> is needed
	conn_handle	0-max, integer num
	char_index	characteristic index, integer num: 0-device name charac 1-device appearance charac 2-peripheral preferred connection parameter charac
Sample	ATBC=gattc,gaps_disc,16 to discover gaps service related information ATBC=gattc,gaps_read_char,16,0 to read gaps device name	

NOTE

ATBC=gattc, gaps_disc,... shall be executed first before other gaps_cmds.

3.3.11 ATBC=gattc,<simple_ble_cmd> - Used by simple ble service client to execute services operations

ATBC=gattc,<simple_ble_cmd>,<conn_handle>,<...>		
Description	This command is used by simple ble service client to execute services operations.	
Parameter	simple_ble_cmd	String:

		simble_disc - simple ble client discover service information, {...} = none, (when use ble_mgr lib, don't need this cmd to discover) simble_read - simple ble client read charac_v0 value, {...} = none simble_write - simple ble client write charac_v1 value, {...} = <length>,<data> simble_set_notify – simple ble client set charac_v2 cccd notify, {...} = <en/disable> simble_set_indicate – simple ble client set charac_v3 cccd indicate, {...} = <en/disable>
	conn_handle	0-max, integer num
	length	length of writed data, integer num
	data	writed_data, hexdata No 0x
	en/disable	0/1, integer num: 1 - enable 0 - disable
Sample	ATBC=gattc,simble_disc,16 to discover simple ble service information ATBC=gattc,simble_read,16 to read simple ble charac_v0 value ATBC=gattc,simble_write,16,4,45454545 to write hexdata to simple ble charac_v1 ATBC=gattc,simble_set_notify,16,1 to set simple ble charac_v2 cccd notify enable ATBC=gattc,simble_set_notify,16,0 to set simple ble charac_v2 cccd notify disable ATBC=gattc,simble_set_indicate,16,1 to set simple ble charac_v3 cccd indicate enable ATBC=gattc,simble_set_indicate,0,0 to set simple ble charac_v3 cccd indicte disable	

NOTE

ATBC=gattc, simble_disc,... shall be executed first before other simple_ble_cmd.

4 APIs

For detailed BT API user manual, refer to AN1006_RTL8721Dx_Bluetooth_API_v1.1.chm in SDK.

Revision History

Date	Version	Description
2023-08-04	v1.0	Initial release
2023-09-11	v1.1	Add more AT commands