BLE API User's Guide

Revision: V1.00 Date: June 20, 2022

www.holtek.com

Table of Contents

1. Introduction

This document briefly describes the topic, BLE Supported API.

This document supports the following chips or relative chips:

- BC7701: BLE SOC
- HT32F67741: CM0+BC7701

1.1 BLE Flash Structure

The BC7701 BLE SOC is Flash structure and contains 6 blocks as shown in Figure 1. The swPara block can be used to save BLE parameters (ex: BD_Name, BD_ADDR, etc.) by setting swParaUpdate to 1 (See API_Feature).

The BC7701 boot flow will be swBoot → swStack → swApp1 or swApp2 (swApp1 has the higher priority if both are valid), when the next block is valid.

If swStack or swApp is not valid, the BC7701 will keep in swBoot.

The block valid check method is described as follows.

Run swBoot first, to jump to valid swApp (swApp1 has the higher priority).

The swApp1/swApp2 should have different addresses. Each block has a validCheck field in the last word, if valid swStack/swApp is not found, the BC7701 will keep in swBoot.

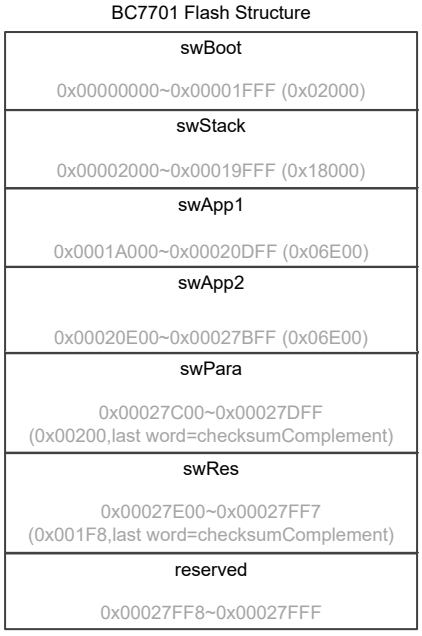

Figure 1 BLE Structure

The BC7701 formal released bin has format as BC7701_XY.bin. Here X means function, Y means version.

The BC7701 valid swApp range is size (swBoot+swStack) ~ 0x27C00. The swApp start address will depend on swBoot/swStack size.

The swBoot/swStack/swApp1/swApp2 contains 32 bytes of blockInformation with in the tail of 1st 1K range, as shown in Figure 2.

Ex: The swBoot's blockInformation is saved in 0x3E0~0x3FF. The swStack's blockInformation is saved in 0x23E0~0x23FF. The swApp1's blockInformation is saved in 0x1A3E0~0x1A3FF.

The swPara has no blockInformation and uses the last word to save checksumComplement.

The swRes has no blockInformation and uses the last word to save checksumComplement.

The swBoot/swStack/swApp1/swApp2's checksum algorithm will be 0 in 32-bit add, the checksumComplement is generated when bin releases. The swBoot/swStack/swApp1/swApp2 each contain a blockInformation.

Both swPara and swRes have checksum algorithm the same as 0 in 32-bit add, the checksum-Complement (last word) will be generated by tool or bin release. The swPara/swRes contains no whole blockInformation but only checksumComplement field in the last word.

The blockInformation contains 8 words, the order is shown below.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>magicNum (4-byte)</td><td>Fixed at 0x36323032</td></tr>
<tr><td>address (4-byte)</td><td>Block start address, ex: swBoot=0, swStack=0x2000</td></tr>
<tr><td>size (4-byte)</td><td>Block valid size, valid block range=(address)~(address+size), ex: swBoot=0x2000, swStack=0x18000</td></tr>
<tr><td>reserved (4-byte)</td><td>Byte 0~3: Reserved</td></tr>
<tr><td>para0 (4-byte)</td><td>Use for UART update
Byte 0: updateExeRegion (which block executes API)
1: Updated by swBoot (API will be parsed by swBoot)
2: Updated by swApp (API will be parsed by swApp)
Byte 1: updateDesRegion (where data will be written)
1: Flash
2: External Flash
3: UART
Byte 2: updateSysclock (which clock is used in update)
unit in MHz
Byte 3: updateSwEnterMode: force BLE to specific mode
0x00: urMode0
0x01: urMode1
0x02: urMode2
0x03: urMode3
0x40: mcuMode: update external MCU through BLE (Phone→BLE→MCU)
0x80: bleMode: update BLE through BLE (Phone→BLE)

1. ioEnterMode method: After nRST=0→1, BLE UART TX/RX can keep I/O at first 60ms, then change I/O back to UART TX/RX, the BC7701 will keep checking I/O when TX/RX are not both high.
2. In urMode0 the swPara can rewrite UART baud rate after entering swApp
3. In urMode1 the swPara cannot rewrite UART baud rate after enter swApp
4. In urMode2 the swPara cannot rewrite UART baud rate due to not enter swApp
5. In urMode3 the swPara cannot rewrite UART baud rate due to not enter swApp
6. After entering urMode2/urMode3, only current UART can be used, ex: if enter mode from UART1, UART2 will be disabled

<table><tr><th>UART I/O state when nRST=0→1 at first 60ms</th><th>BLE TX=1 (PA0, PB6)</th><th>BLE TX=0 (PA0, PB6)</th></tr><tr><td>BLE RX=1
(PA1, PB7)</td><td>115200 (urMode0, default)
Jump to swApp1/swApp2</td><td>9600 (urMode1)
Jump to swApp1/swApp2, not use swPara baud rate if exist</td></tr><tr><td>BLE RX=0
(PA1, PB7)</td><td>9600 (urMode2)
Keep in swBoot, disable another UART, not use swPara baud rate if exist</td><td>115200 (urMode3)
Keep in swBoot, disable another UART, not use swPara baud rate if exist</td></tr></table></td></tr>
<tr><td>para1 (4-byte)</td><td>The same meaning as para0 but use for APP Air update (OTA)</td></tr>
<tr><td>checksumComplement (4-byte)</td><td>Each block 32-bit-add checksum should be 0, this word is used to save checksum's complement.
swPara/swRes block has no blockInformation but this word which is put in the last word.</td></tr>
</table>

Field	Description
version (4-byte)	Code version, xx will be increased when releasing. Tool should force to update when the read version is not the same, the MSB 3 bytes will be the fixed number. swBoot format=0x800107xx swStack format=0x400107xx swApp format=0x000107xx (swApp1/swApp2 use the same format)

The validCheck is used to check whether a block is valid or not, put in the last word of the block.

Field	Description
validCheck (4-byte)	Block valid check, put in the last word of the block, if any one byte is not equal to 0x00 or 0xFF, this means invalid. The default value is 0xFFFFFF00. Ex: 0xFF00FF00=valid, 0xFFFFFF01=invalid

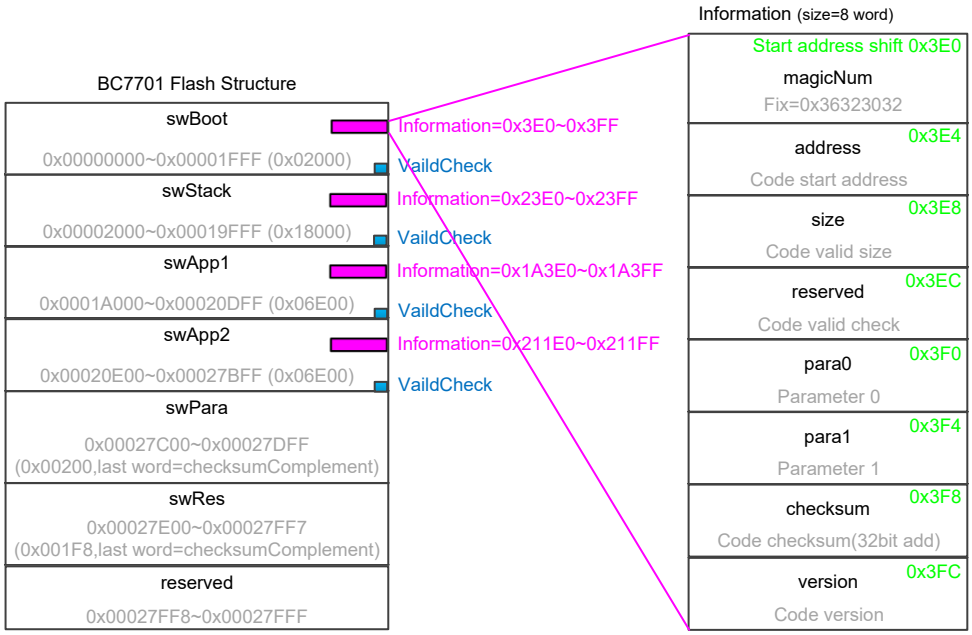

Figure 2 blockInformation

1.2 BLE States

The BC7701 BLE states are shown in the following diagram.

The BC7701 nRST pull-low period must be greater than 5ms for full hardware reset.

The BC7701 runs in 16MHz by default after the nRST is set from 0 to 1 or a watchdog (WDT) reset occurs, all settings use the default values.

The Host can use API_Reset to trigger WDT reset or set I/O nRST from 0 to 1.

After the nRST is set from 0 to 1 or a WDT reset occurs, the BC7701 will need 60ms to startup and then check I/O. If the mode is required to change, the UART TX/RX I/O must keep level after the nRST is set from 0 to 1 and then change back to UART TX/RX when exceeds this time. The BC7701 UART TX/RX I/O will stay in input and keep checking when TX/RX are not both high. (See blockInformation)

After TX API_PowerSaving event, the BLE may enter the Deep Sleep or Power Down mode, extra dummy bytes (at least 100μs long) are required to wake up the BC7701 before sending any API command.

The BC7701 in Deep Sleep can be dummy bytes waked up and kept at the previous settings. After

wake-up auto switch to the Sleep mode, after dummy bytes wake-up must wait 30ms to full wake-up the MCU (duration 0ms~2ms for UART hardware wake-up (cannot send any data during this period), duration 2ms~30ms the BLE can only RX max 16 bytes due to firmware not waked up).

The BC7701 in Power Down can be dummy waked up. After wake-up, it will trigger a watchdog (WDT) reset, all settings return to the default values (no advertising wait for UART input).

After disconnection, the BC7701 will software reset for 5ms. After RX disconnection status, it cannot send any command during this period.

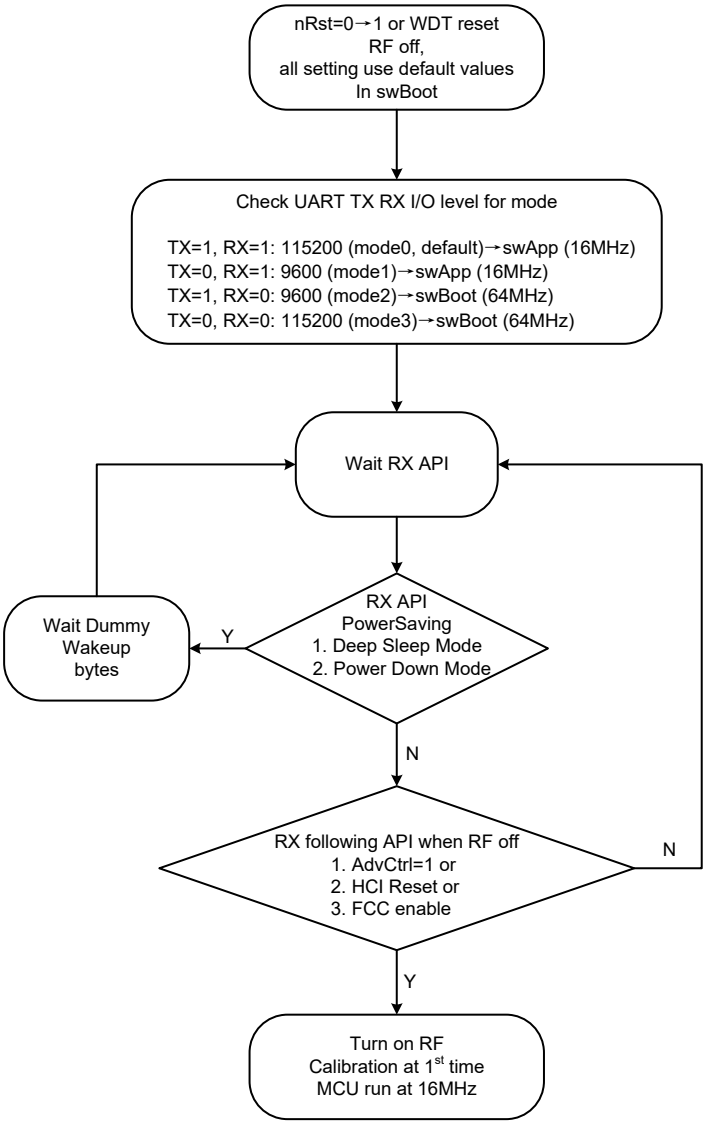

Figure 3 BLE States

1.3 BLE API List

The BLE API list is shown in the following table, including CmdType1, CmdType2 and CmdType3.

- CmdType1: HCI command, use to setup test
- CmdType2: use to setup UUID (service relative. Extra Service UUID can reference BLE_Service_xxxx_xxxx.pdf).
- CmdType3: use to setup BLE (function relative)

Type		Note
CmdType1		
HCI_Reset		Reset BLE test state Ex: 01 030C 00 (DTM reset)
HCI_LE_Receiver_Test		BLE RX test mode (1M) Ex: 01 1D20 01 00 (DTM RX 2402MHz)
HCI_LE_Transmitter_Test		BLE TX test mode(1M) Ex: 01 1E20 03 000100 (DTM TX 2402MHz)
HCI_LE_Test_End		BLE end test Ex: 01 1F20 00 (DTM end)
HCI_LE_Enhanced_Receiver_Test		BLE RX test mode(1M/2M) Ex: 01 3320 03 000100 (DTM RX 2402MHz@1MHz)
HCI_LE_Enhanced_Transmitter_Test		BLE TX test mode(1M/2M) Ex: 01 3420 04 000A0301 (DTM TX 2402MHz@1MHz)
CmdType2		
0x1800	Generic Access	Setup profile Generic Access Default: 0x0000 (0x2A01)+0x00 (0x2A02) Ex: 77 03 00 0018 (Read UUID=0x1800)
0x2A01	Appearance	Setup Characteristic Appearance Default: 0x0000 Ex: 77 03 00 012A (Read UUID=0x2A01)
0x2A02	Peripheral Privacy Flag	Setup Characteristic Peripheral Privacy Flag Default: 0x00 Ex: 77 03 00 022A (Read UUID=0x2A02)
0x180A	Device Information Service	Setup Service Device Information Default: 0x486F6C74656B20202020202020202020(0x2A29) +42433737303120202020202020202020(0x2A24) +302E3031(0x2A25)+302E3031(0x2A27)+302E3031(0x2A26) +302E3031(0x2A28)+00000000(0x2A23)+00000000(0x2A2A) +00000000000000(0x2A50) Ex: 77 03 00 0A18 (Read UUID=0x180A)
0x2A29	Manufacturer Name	Setup Characteristic Manufacturer Name Default: 0x486F6C74656B20202020202020202020 Ex: 77 03 00 292A (Read UUID=0x2A29)
0x2A24	Model Number	Setup Characteristic Model Number Default: 0x42433737303120202020202020202020 Ex: 77 03 00 242A (Read UUID=0x2A24)
0x2A25	Serial Number	Setup Characteristic Serial Number Default: 0x302E3031 Ex: 77 03 00 252A (Read UUID=0x2A25)
0x2A27	Hardware Revision	Setup Characteristic Hardware Revision Default: 0x302E3031 Ex: 77 03 00 272A (Read UUID=0x2A27)
0x2A26	Firmware Revision	Setup Characteristic Firmware Revision Default: 0x302E30311 Ex: 77 03 00 262A (Read UUID=0x2A26)
0x2A28	Software Revision	Setup Characteristic Software Revision Default: 0x302E30311 Ex: 77 03 00 282A (Read UUID=0x2A28)

Type		Note
0x2A23	System ID	Setup Characteristic System ID Default: 0x00000000 Ex: 77 03 00 232A (Read UUID=0x2A23)
0x2A2A	IEEE	Setup Characteristic IEEE Default: 0x00000000 Ex: 77 03 00 2A2A (Read UUID=0x2A2A)
0x2A50	PnP ID	Setup Characteristic PnP ID Default: 0x00000000000000 Ex: 77 03 00 502A (Read UUID=0x2A50)
0x180F	Battery Service	Setup Service Battery Default: 0x64(0x2A19) Ex: 77 03 00 0F18 (Read UUID=0x180F)
0x2A19	Battery Level	Setup Characteristic Battery Level Default: 0x64 Ex: 77 03 00 192A (Read UUID=0x2A19)
0xFFF0	Unknown Service	Setup Service Unknown Default: all map to the same buffer, default length=0 Ex: 77 03 00 F0FF (Read UUID=0xFFF0)
0xFFF1	Unknown Notify	Setup Characteristic Notify Default: all map to the same buffer, default length=0 Ex: 77 03 00 F1FF (Read UUID=0xFFF1)
0xFFF2	Unknown Write Without Response	Setup Characteristic Write Without Response Default: all map to the same buffer, default length=0 Ex: 77 03 00 F2FF (Read UUID=0xFFF2)
CmdType3		
0x0000	API_Status	[RO] Use to get status Default: 0x00000000/0x00000100 Ex: 77 03 00 0000 (Read status)
0x0002	API_Disconnect	[WO] Use to disconnect current link Ex: 77 04 00 0200 00 (Force disconnection)
0x0003	API_ConnIntv	[RW] Use to modify connection interval Default: 0x0000 Ex: 77 03 00 0300 (Read connIntv)
0x0004	API_ConnIntv1	[RW] Use to modify connection interval Default: 0x01F4000000060006 Ex: 77 03 00 0400 (Read connIntv)
0x0005	API_BTName	[RW] Use to setup BLE device name Default: BC7701_(0/1)_(swAppVersion) Ex: 77 03 00 0500 (Read btName)
0x0006	API_BTAddr	[RW] Use to setup BLE device address Default: Release date or MP_BD_ADDR Ex: 77 03 00 0600 (Read btAddr)
0x0007	API_AdvCtrl	[RW] Use to enable advertising Default: 0x00 Ex: 77 03 00 0700 (Read advCtrl)
0x0008	API_AdvIntv	[RW] Use to setup advertising interval Default: 0x0700A000A0 Ex: 77 03 00 0800 (Read advIntv)
0x0009	API_AdvData	[RW] Use to setup advertising data Default: AD_flag+AD_CompleteLocalName Ex: 77 03 00 0900 (Read advData)
0x000A	API_ScanData	[RW] Use to setup scan response data Default: none Ex: 77 03 00 0A00 (Read scanData)
0x000B	API_TxPwr	[RW] Use to setup RF TX power Default: 0x0C Ex: 77 03 00 0B00 (Read txPwr)

Type		Note
0x000E	API_CryStalOffset	[RW] Use to setup external high crystal offset Default: 0x04 Ex: 77 03 00 0E00 (Read crystalOffset)
0x000F	API_PeerBTAddr	[RO] Use to get peer device address Default: 0x00000000000000 Ex: 77 03 00 0F00 (Read peerAddr)
0x0010	API_Feature	[RW] Use to setup function Default: 0x00000000 Ex: 77 03 00 1000 (Read feature)
0x0020	API_Version	[RO] Use to get version Default: depend on release bin Ex: 77 03 00 2000 (Read runningVersion)
0x0025	API_PowerSaving	[RW] Use to setup power saving mode in BLE Default: 0x0000000000020100 Ex: 77 03 00 2500 (Read slave powerSaving mode)
0x0026	API_InterfaceSpeed	[RW] Use to setup UART baud rate Default: 0x0001C200(115200) Ex: 77 03 00 2600 (Read interfaceSpeed)
0x0027	API_InterfaceSpeedMax	[RW] Use to setup UART baud rate to max Default: 0x0030D400(3200000) Ex: 77 03 00 2700 (Read interfaceSpeedMax)
0x0028	API_Reset	[WO] Use to firmware reset Ex: 77 04 00 2800 00 (WDT reset)
0x002A	API_WhiteList	[RW] Use to setup whitelist Default: none Ex: 77 03 00 2A00 (Read whiteList)
0x0040	API_IP	[WO] Use to setup IP route path Ex: 77 0F 10 4000 02070100 BB000000 00100440 (route BLE to UART1)
0x0050	API_GPIO	[RW] Use to setup GPIO Default: none Ex: 77 03 00 5000 (Read Port A)
0x00CC	API_FCC	[WO] Use to set FCC test mode Ex: 77 0A 00 CC00 03000000000000 (enter FCC mode)

Table 1 BLE API List

2. BLE API CmdType1

The BLE API CmdType1 is used as HCI Cmd which defines in Bluetooth specification (refer to the HCI COMMANDS AND EVENTS section).

The BLE API CmdType1 format is shown below.

The CmdType1 is not aligned with 8 bit in OpCode Command Field (OCF) and OpCode Group Field (OGF).

The Length should be the sum of the Value field in byte.

CmdType1 Command (MCU→BLE)	Header (8-bit)	OpCode Command Field (OCF, 10-bit)	OpCode Group Field (OGF, 6-bit)	Length (8-bit)	Value (n-byte)
	01	xxxx	xx	xx	xx*n

CmdType1 Event (BLE→MCU)	Header (8-bit)	EvtCode (8-bit)	Length (8-bit)	Value (n-byte)
	04	xx	xx	xx*n

Table 2 CmdType1 Format

The supported CmdType1 List is shown below.

CmdType1	
Type	**Note**
HCI_Reset	Reset BLE test state Ex: 01 030C 00 (DTM reset)
HCI_LE_Receiver_Test	BLE RX test mode (1M) Ex: 01 1D20 01 00 (DTM RX 2402MHz)
HCI_LE_Transmitter_Test	BLE TX test mode(1M) Ex: 01 1E20 03 000100 (DTM TX 2402MHz)
HCI_LE_Test_End	BLE end test Ex: 01 1F20 00 (DTM end)
HCI_LE_Enhanced_Receiver_Test	BLE RX test mode(1M/2M) Ex: 01 3320 03 000100 (DTM RX 2402MHz@1MHz)
HCI_LE_Enhanced_Transmitter_Test	BLE TX test mode(1M/2M) Ex: 01 3420 04 000A0301 (DTM TX 2402MHz@1MHz)

Table 3 CmdType1 List

2.1 HCI_Reset

Command:				
Command	**OCF**	**OGF**	**Command Parameters**	**Return Parameters**
HCI_Reset	0x0003	0x03	—	Status (1-byte)

Description:
Reset BLE state

Return Parameters:		
Event Parameters	**Value**	**Parameter Description**
Status (1-byte)	0x00	Success
	0x01~0xFF	Fail (Spec Error Codes)

Example:		
BLE	**Direction**	**Tester**
	←	01030C00 01: Command 030C: Command OpCode=HCI_Reset 00: Command Length
040E04xx030C00 04: Event 0E: Event OpCode=Command Complete 04: Event Length xx: Num_HCI_Command_Packets 030C: Command OpCode=HCI_Reset 00: Event parameter=success	→	
BLE test state reset		
xx: means that it can be ignored		

2.2 HCI_LE_Receiver_Test

Command:				
Command	**OCF**	**OGF**	**Command Parameters**	**Return Parameters**
HCI_LE_Receiver_Test	0x001D	0x08	RX_Channel (1-byte)	Status (1-byte)

Description:
Setup BLE in RX test mode BLE will response Command Complete event BC7701 needs at least 26ms to enter RX mode

Command Parameters:		
Command Parameters	**Value**	**Parameter Description**
RX_Channel (1-byte)	N=0xXX	N=(Freq-2402)/2 Range: 0x00 (2402MHz)~0x27 (2480MHz)

Return Parameters:		
Event Parameters	**Value**	**Parameter Description**
Status (1-byte)	0x00	Success
	0x01~0xFF	Fail (Spec Error Codes)

Example:		
BLE	**Direction**	**Tester**
	←	011D200100 01: Command 1D20: Command OpCode=HCI_LE_ Receiver_Test 01: Command Length 00: channel 2402
040E04xx1D2000 04: Event 0E: Event OpCode=Command Complete 04: Event Length xx: Num_HCI_Command_Packets 1D20: Command OpCode=HCI_LE_ Receiver_Test 00: Event parameter=success	→	
BLE in RX test mode, wait for packet		If Tester in TX mode, continue TX packet
xx: means that it can be ignored		

2.3 HCI_LE_Transmitter_Test

Command:				
Command	**OCF**	**OGF**	**Command Parameters**	**Return Parameters**
HCI_LE_Transmitter_Test	0x001E	0x08	TX_Channel (1-byte)	Status (1-byte)
			Length_Of_Test_Data (1-byte)	
			Packet_Payload (1-byte)	

Description
Setup BLE in TX test mode BLE will response Command Complete event BC7701 needs at least 22ms to enter TX mode

Command Parameters:		
Command Parameters	**Value**	**Parameter Description**
TX_Channel (1-byte)	N=0xXX	N=(Freq-2402)/2 Range: 0x00 (2402MHz)~0x27 (2480MHz)
Length_Of_Test_Data (1-byte)	0x00~0xFF	Each payload length
Packet_Payload (1-byte)	0x00	PRBS9 ('11111111100000111101...' (in transmission order)), DTM [Vol 6] Part F, Section 4.1.5
	0x01	Repeated ('11110000' (in transmission order)), DTM [Vol 6] Part F, Section 4.1.5
	0x02	Repeated ('10101010' (in transmission order)), DTM [Vol 6] Part F, Section 4.1.5
	0x03	PRBS15, DTM [Vol 6] Part F, Section 4.1.5
	0x04	Repeated ('11111111' (in transmission order))
	0x05	Repeated ('00000000' (in transmission order))
	0x06	Repeated ('00001111' (in transmission order))
	0x07	Repeated ('01010101' (in transmission order))
	0x08~0xFF	Reserved

Return Parameters:		
Event Parameters	**Value**	**Parameter Description**
Status (1-byte)	0x00	Success
	0x01~0xFF	Fail (Spec Error Codes)

Example:		
BLE	**Direction**	**Tester**
	←	011E2003000100 01: Command 1E20: Command OpCode=HCI_LE_Transmitter_Test 03: Command Length 00: channel 2402 01: Length_Of_Test_Data=1 00: Packet_Payload=PRBS9
040E04xx1E2000 04: Event 0E: Event OpCode=Command Complete 04: Event Length xx: Num_HCI_Command_Packets 1E20: Command OpCode=HCI_LE_Transmitter_Test 00: Event parameter=success	→	
BLE in TX test mode, continue TX packet		If Tester in RX mode, receive packet
xx: means that it can be ignored		

2.4 HCI_LE_Test_End

Command:				
Command	**OCF**	**OGF**	**Command Parameters**	**Return Parameters**
HCI_LE_Test_End	0x001F	0x08	—	Status (1-byte)
				Number_Of_Packets (2-byte)

Description:
Stop BLE test mode BLE will response Command Complete event with Number_Of_Packets If not RX any packet BC7701 will wait 50ms and then exit mode

Return Parameters:		
Event Parameters	**Value**	**Parameter Description**
Status (1-byte)	0x00	Success
	0x01~0xFF	Fail (Spec Error Codes)
Number_Of_Packets (2-byte)	0x0000~0xFFFF	Transmitted/Received packet number

Example:		
BLE	**Direction**	**Tester**
	←	011F2000 01: Command 1F20: Command OpCode=HCI_LE_Test_End 00: Command Length
040E06xx1F20000000 04: Event 0E: Event OpCode=Command Complete 06: Event Length xx: Num_HCI_Command_Packets 1F20: Command OpCode=HCI_LE_Test_End 00: Event parameter=success 0000: Number_Of_Packets	→	
BLE stop test mode, do nothing		Tester stop test mode, calculate result
xx: means that it can be ignored		

2.5 HCI_LE_Enhanced_Receiver_Test

Command:				
Command	**OCF**	**OGF**	**Command Parameters**	**Return Parameters**
HCI_LE_Enhanced_Receiver_Test	0x0033	0x08	RX_Channel (1-byte)	Status (1-byte)
			PHY (1-byte)	
			Modulation_Index (1-byte)	

Description:
BLE enters Enhanced RX test mode After completion, Controller will response Command Complete event BC7701 needs at least 26ms to enter RX mode

Command Parameters:		
Command Parameters	**Value**	**Parameter Description**
RX_Channel (1-byte)	0x00~0x27 (2402MHz to 2480MHz)	Freq=(RX_Channel×2)+2402
	Others	Reserved
PHY (1-byte)	0x00	Reserved
	0x01	RX LE 1M PHY
	0x02	RX LE 2M PHY
	Others	Reserved
Modulation_Index (1-byte)	0x00	TX=standard modulation index
	Others	Reserved

Return Parameters:		
Event Parameters	**Value**	**Parameter Description**
Status (1-byte)	0x00	Success
	0x01~0xFF	Fail (Spec Error Codes)

Example:		
BLE	**Direction**	**Tester**
	←	01332003030100 01: Command 3320: Command OpCode=HCI_LE_Enhanced_Receiver_Test 03: Command Length 03: RX_Channel=2408 01: PHY=LE 1M 00: Modulation_Index=standard modulation index
040E04xx332000 04: Event 0E: Event OpCode=Command Complete 04: Event Length xx: Num_HCI_Command_Packets 3320: Command OpCode=HCI_LE_Enhanced_Receiver_Test 00: Event parameter=success	→	
BLE in RX test mode, wait for packet		If Tester in TX mode, continue TX packet
xx: means that it can be ignored		

2.6 HCI_LE_Enhanced_Transmitter_Test

Command:				
Command	**OCF**	**OGF**	**Command Parameters**	**Return Parameters**
HCI_LE_Enhanced_Transmitter_Test	0x0034	0x08	TX_Channel (1-byte)	Status (1-byte)
			Length_Of_Test_Data (1-byte)	
			Packet_Payload (1-byte)	
			PHY (1-byte)	

Description:
BLE enters Enhanced TX test mode If this command is supported, Packet_Payload=0x00/0x01/0x02/(04 if Coded PHY is supported) must be supported After completion, Controller will response Command Complete event BC7701 needs at least 22ms to enter TX mode

Command Parameters:		
Command Parameters	**Value**	**Parameter Description**
TX_Channel (1-byte)	0x00~0x27 (2402MHz to 2480MHz)	Freq=(RX_Channel×2)+2402
	Others	Reserved
Length_Of_Test_Data (1-byte)	0x00~0xFF	Each payload length
Packet_Payload (1-byte)	0x00	PRBS9 (DTM[Vol 6] Part F, Section 4.1.5)
	0x01	Repeated '11110000' (DTM[Vol 6] Part F, Section 4.1.5)
	0x02	Repeated '10101010' (DTM[Vol 6] Part F, Section 4.1.5)
	0x03	PRBS15 (DTM[Vol 6] Part F, Section 4.1.5)
	0x04	Repeated '11111111'
	0x05	Repeated '00000000'
	0x06	Repeated '00001111'
	0x07	Repeated '01010101'
	Others	Reserved
PHY (1-byte)	0x00	Reserved
	0x01	TX LE 1M PHY
	0x02	TX LE 2M PHY
	Others	Reserved

Return Parameters:		
Event Parameters	**Value**	**Parameter Description**
Status (1-byte)	0x00	Success
	0x01~0xFF	Fail (Spec Error Codes)

Example:		
BLE	**Direction**	**Tester**
	←	01342004030A0301 01: Command 3420: Command OpCode=HCI_LE_ Enhanced_Transmitter_Test 04: Command Length 03: TX_Channel=2408 0A: Length_Of_Test_Data=10 03: Packet_Payload=PRBS15 01: PHY=LE 1M
040E04xx342000 04: Event 0E: Event OpCode=Command Complete 04: Event Length xx: Num_HCI_Command_Packets 3420: Command OpCode=HCI_LE_ Enhanced_Transmitter_Test 00: Event parameter=success	→	
BLE in TX test mode, continue TX packet		If Tester in RX mode, receive packet
xx: means that it can be ignored		

3. BLE API CmdType2

The BLE API CmdType2 is used to access BLE service Universally Unique Identifier (UUID) in chip.

A multi-byte field will be sent with LSB first. "B" is denoted as Byte, "b" is denoted as bit.

When the BLE chip RX header is 0x77, it should response with header=0x78.

If the Value field length is not zero, it indicates a write action. Otherwise it indicates a read action.

A write action can setup partial value, the remaining part will be kept in RAM.

For more detailed CmdType2, refer to BLE_Service_xxxx_xxxx.pdf.

The Length should be the sum of several fields in byte, which are CmdFlag/EvetStatus, Type and Value.

Description	Header (1-byte)	Length (1-byte)	CmdFlag/EvtStatus (1-byte)	Type (2-byte)	Value (n-byte)
CmdType2 Command (MCU→BLE)	0x77	xx	CmdFlag: b[0]: 1: Force Evt to read action format b[3:1]: — b[7:4]: 1: CmdFlag_supported 2: CmdFlag_notifyIndicate	UUID (Universal Unique Identifier) Profile: 0x18xx Service: 0x18xx Characteristics: 0x2Axx Descriptor: 0x29xx	Depend on Type
CmdType2 Event (BLE→MCU)	0x78		EvtStatus: b[3:0]: ErrorCode b[7:4]: 1: CmdFlag_supported 2: CmdFlag_notifyIndicate		

Table 4 CmdType2 Format

The supported accessible UUIDs are shown below. (service relative)

CmdType2		
Type		**Note**
0x1800	Generic Access	Setup profile Generic Access Default: 0x0000 (0x2A01)+0x00 (0x2A02) Ex: 77 03 00 0018 (Read UUID=0x1800)
0x2A01	Appearance	Setup Characteristic Appearance Default: 0x0000 Ex: 77 03 00 012A (Read UUID=0x2A01)
0x2A02	Peripheral Privacy Flag	Setup Characteristic Peripheral Privacy Flag Default: 0x00 Ex: 77 03 00 022A (Read UUID=0x2A02)
0x180A	Device Information Service	Setup Service Device Information Default: 0x486F6C74656B20202020202020202020(0x2A29) +42433737303120202020202020202020(0x2A24) +302E3031(0x2A25)+302E3031(0x2A27)+302E3031(0x2A26) +302E3031(0x2A28)+00000000(0x2A23)+00000000(0x2A2A) +00000000000000(0x2A50) Ex: 77 03 00 0A18 (Read UUID=0x180A)
0x2A29	Manufacturer Name	Setup Characteristic Manufacturer Name Default: 0x486F6C74656B20202020202020202020 Ex: 77 03 00 292A (Read UUID=0x2A29)
0x2A24	Model Number	Setup Characteristic Model Number Default: 0x42433737303120202020202020202020 Ex: 77 03 00 242A (Read UUID=0x2A24)
0x2A25	Serial Number	Setup Characteristic Serial Number Default: 0x302E3031 Ex: 77 03 00 252A (Read UUID=0x2A25)

CmdType2		
Type		Note
0x2A27	Hardware Revision	Setup Characteristic Hardware Revision Default: 0x302E3031 Ex: 77 03 00 272A (Read UUID=0x2A27)
0x2A26	Firmware Revision	Setup Characteristic Firmware Revision Default: 0x302E30311 Ex: 77 03 00 262A (Read UUID=0x2A26)
0x2A28	Software Revision	Setup Characteristic Software Revision Default: 0x302E30311 Ex: 77 03 00 282A (Read UUID=0x2A28)
0x2A23	System ID	Setup Characteristic System ID Default: 0x00000000 Ex: 77 03 00 232A (Read UUID=0x2A23)
0x2A2A	IEEE	Setup Characteristic IEEE Default: 0x00000000 Ex: 77 03 00 2A2A (Read UUID=0x2A2A)
0x2A50	PnP ID	Setup Characteristic PnP ID Default: 0x00000000000000 Ex: 77 03 00 502A (Read UUID=0x2A50)
0x180F	Battery Service	Setup Service Battery Default: 0x64(0x2A19) Ex: 77 03 00 0F18 (Read UUID=0x180F)
0x2A19	Battery Level	Setup Characteristic Battery Level Default: 0x64 Ex: 77 03 00 192A (Read UUID=0x2A19)
0xFFF0	Unknown Service	Setup Service Unknown Default: all map to the same buffer, default length=0 Ex: 77 03 00 F0FF (Read UUID=0xFFF0)
0xFFF1	Unknown Notify	Setup Characteristic Notify Default: all map to the same buffer, default length=0 Ex: 77 03 00 F1FF (Read UUID=0xFFF1)
0xFFF2	Unknown Write Without Response	Setup Characteristic Write Without Response Default: all map to the same buffer, default length=0 Ex: 77 03 00 F2FF (Read UUID=0xFFF2)

Table 5 CmdType2 List

4. BLE API CmdType3

The BLE API CmdType3 is used to setup the BLE chip internal variable or state, including BTAddr, BTName, Crystaloffset, FCC test mode, etc.

The BLE API CmdType3 format is shown below, which is the same as CmdType2. A multi-byte field will be sent with LSB first. "B" is denoted as Byte, "b" is denoted as bit.

When the BLE chip RX header is 0x77, it should response with header=0x78.

If the Value field length is not zero, it indicates a write action. Otherwise it indicates a read action.

A write action can setup partial value, the unset part will be kept in RAM.

The Length should be the sum of several fields in byte, which are CmdFlag/EvetStatus, Type and Value.

Description	Header (1-byte)	Length (1-byte)	CmdFlag/EvtStatus (1-byte)	Type (2-byte)	Value (n-byte)
CmdType3 Command (MCU→BLE)	0x77	xx	CmdFlag: b[0]: 1: Force Evt to read action format b[7:1]: Depend on Type	As follows	Depend on Type
CmdType3 Event (BLE→MCU)	0x78		EvtStatus: b[3:0]: ErrorCode b[7:4]: —		

Table 6 CmdType3 Format

The supported list is shown below.

CmdType3		
Type		Note
0x0000	API_Status	[RO] Use to get status Default: 0x00000000/0x00000100 Ex: 77 03 00 0000 (Read status)
0x0002	API_Disconnect	[WO] Use to disconnect current link Ex: 77 04 00 0200 00 (Force disconnection)
0x0003	API_ConnIntv	[RW] Use to modify connection interval Default: 0x0000 Ex: 77 03 00 0300 (Read connIntv)
0x0004	API_ConnIntv1	[RW] Use to modify connection interval Default: 0x01F4000000060006 Ex: 77 03 00 0400 (Read connIntv)
0x0005	API_BTName	[RW] Use to setup BLE device name Default: BC7701_(0/1)_(swAppVersion) Ex: 77 03 00 0500 (Read btName)
0x0006	API_BTAddr	[RW] Use to setup BLE device address Default: Release date or MP_BD_ADDR Ex: 77 03 00 0600 (Read btAddr)
0x0007	API_AdvCtrl	[RW] Use to enable advertising Default: 0x00 Ex: 77 03 00 0700 (Read advCtrl)
0x0008	API_AdvIntv	[RW] Use to setup advertising interval Default: 0x0700A000A0 Ex: 77 03 00 0800 (Read advIntv)
0x0009	API_AdvData	[RW] Use to setup advertising data Default: AD_flag+AD_CompleteLocalName Ex: 77 03 00 0900 (Read advData)
0x000A	API_ScanData	[RW] Use to setup scan response data Default: none Ex: 77 03 00 0A00 (Read scanData)

CmdType3		
Type		Note
0x000B	API_TxPwr	[RW] Use to setup RF TX power Default: 0x0C Ex: 77 03 00 0B00 (Read txPwr)
0x000E	API_CryStalOffset	[RW] Use to setup external high crystal offset Default: 0x04 Ex: 77 03 00 0E00 (Read crystalOffset)
0x000F	API_PeerBTAddr	[RO] Use to get peer device address Default: 0x00000000000000 Ex: 77 03 00 0F00 (Read peerAddr)
0x0010	API_Feature	[RW] Use to setup function Default: 0x00000000 Ex: 77 03 00 1000 (Read feature)
0x0020	API_Version	[RO] Use to get version Default: depend on release bin Ex: 77 03 00 2000 (Read runningVersion)
0x0025	API_PowerSaving	[RW] Use to setup power saving mode in BLE Default: 0x0000000000020100 Ex: 77 03 00 2500 (Read slave powerSaving mode)
0x0026	API_InterfaceSpeed	[RW] Use to setup UART baud rate Default: 0x0001C200(115200) Ex: 77 03 00 2600 (Read interfaceSpeed)
0x0027	API_InterfaceSpeedMax	[RW] Use to setup UART baud rate to max Default: 0x0030D400(3200000) Ex: 77 03 00 2700 (Read interfaceSpeedMax)
0x0028	API_Reset	[WO] Use to firmware reset Ex: 77 04 00 2800 00 (WDT reset)
0x002A	API_WhiteList	[RW] Use to setup whitelist Default: none Ex: 77 03 00 2A00 (Read whiteList)
0x0040	API_IP	[WO] Use to setup IP route path Ex: 77 0F 10 4000 02070100 BB000000 00100440 (route BLE to UART1)
0x0050	API_GPIO	[RW] Use to setup GPIO Default: none Ex: 77 03 00 5000 (Read Port A)
0x00CC	API_FCC	[WO] Use to set FCC test mode Ex: 77 0A 00 CC00 03000000000000 (enter FCC mode)

Table 7 CmdType3 List

4.1 API_Status

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	Type	CmdFlag	Value
Status (RO, swBoot/swApp)	0x0000	b[7:0]: — **EvtStatus** b[3:0]: ErrorCode b[7:4]: —	statusFlag (4-byte)

Description:
Use to get status. Default not auto send status when change. The autoSendSatus which is codeFeature_b[7] in API_ Feature can enable it. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
statusFlag (4-byte)	b[0]: connect	1: Connect 0: Disconnect (Default)
	b[1]: parameterUpdateOk	1: Connection parameterUpdateOk (valid after API_ConnIntv/API_ConnIntv1) 0: Meaningless (Default)
	b[2]: parameterUpdateFail	1: Connection parameterUpdateFail (valid after API_ConnIntv/API_ConnIntv1) 0: Meaningless (Default)
	b[3]: parameterUpdatePeer	1: Connection parameterUpdatePeer: Already update connection interval with peer device and save in RAM, can read interval back (valid after API_ConnIntv/API_ConnIntv1) 0: Meaningless (Default)
	b[4]: rfCalibrationDone	1: RF already turned on and calibration done 0: Meaningless (Default)
	b[6:5]	—
	b[7]	1: Fatal error cause reset (Back to default state) 0: Meaningless (Default)
	b[8]: ext32k	1: External 32.768k existed Need API_Feature: codeFeature_b[12]: external32k to force to enable it 0: Not exist (Default)
	b[9]: OTAing	1: Over the air update process 0: Meaningless (Default)
	b[31:10]	—

Example:		
Write Action	MCU→77 07 00 0000 01000000→BLE MCU←78 03 03 0000←BLE	A write action is not allowed EvtStatus: ErrorCode=03=_ST_NOT_SUPPORT
Read Action	MCU→77 03 00 0000→BLE MCU←78 07 00 0000 01000000←BLE	statusFlag: b[0]: connect=1

4.2 API_Disconnect

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
Disconnect (WO, swApp)	0x0002	b[7:0]: — **EvtStatus** b[3:0]: ErrorCode b[7:4]: —	—

Description:
Use to disconnect the current link. Both Write/Read actions can trigger disconnection. The Value will be ignored. Check API_Status for the current link status. If no connection exists, response ErrorCode: 0x05: _ST_INVALID_. BLE action in each state: • Default: Useless. • Advertising: Useless. • Connection: Trigger disconnection.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[7:0]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Example:		
Write Action	MCU→77 04 00 0200 00→BLE MCU←78 03 00 0200←BLE	A write action as well as Read action both can trigger disconnection. The Value will be ignored. ErrorCode=0x00: _ST_SUCCESS_ when connection exists ErrorCode=0x05: _ST_INVALID_ when no connection exists
Read Action	MCU→77 03 00 0200→BLE MCU←78 03 05 0200←BLE	A write action as well as Read action both can trigger disconnection. ErrorCode=0x00: _ST_SUCCESS_ when connection exists ErrorCode=0x05: _ST_INVALID_ when no connection exists

4.3 API_ConnIntv

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
ConnIntv (RW, swApp)	0x0003	b[0]: evtWithValue b[7:1]: —	connIntv (2-byte)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to modify the connection interval. A write action can setup partial value; A read action will retrieve the connection result, 0x00 means not connected. The connection interval value is the negotiation result between chip and phone. If setup ok, API_Status: statusFlag: b[1]: parameterUpdateOk=1, otherwise API_Status: statusFlag: b[2]: parameterUpdateFail=1. After getting parameterUpdateOk=1 or parameterUpdateFail=1, the MCU can read this command again to get the real interval. During service discovery procedure, it may get status parameterUpdateFail=1 then parameterUpdateOk=1. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM and trigger connection interval modification.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
connIntv (2-byte)	0x0006~0x0C80	connection interval unit=1.25ms valid range=7.5ms~4s default=0 (means no connection) Others: will cause API_Status: statusFlag: b[2]: parameterUpdateFail=1

Example:		
Write Action	MCU→77 05 00 0300 0600→BLE MCU←78 03 00 0300←BLE	Setup connIntv=7.5ms, read API_Status to update result
Read Action	MCU→77 03 00 0300→BLE MCU←78 05 00 0300 0600←BLE	Read the current connection interval value

4.4 API_ConnIntv1

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
ConnIntv1 (RW, swApp)	0x0004	b[0]: evtWithValue b[7:1]: —	connIntvMin (2-byte) connIntvMax (2-byte) latency (2-byte) timeout (2-byte)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to modify the connection interval. A write action can setup partial value; A read action will retrieve the connection result, 0x00 means not connected. The connection interval value is the negotiation result between chip and phone. If setup ok, API_Status: statusFlag: b[1]: parameterUpdateOk=1, otherwise API_Status: statusFlag: b[2]: parameterUpdateFail=1. After getting parameterUpdateOk=1 or parameterUpdateFail=1, the MCU can read this command again to get the real interval. During service discovery procedure, it may get status parameterUpdateFail=1 then parameterUpdateOk=1. Enhanced API_ConnIntv, can be setup in more detail. The final connection interval will be between connIntvMin~connIntvMax. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM and trigger connection interval modification.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
connIntvMin (2-byte)	0x0006~0x0C80	Min connection interval unit=1.25ms valid range=7.5ms~4s default=0x0006
connIntvMax (2-byte)	0x0006~0x0C80	Max connection interval unit=1.25ms valid range=7.5ms~4s default=0x0006
latency (2-byte)	0x0000~0x01F3	unit=1slot (connection interval) valid range=0~499 default=0
timeout (2-byte)	0x000A~0x0C80	unit=10ms valid range=100ms~32s default=0x01F4=500

Example:		
Write Action	MCU→77 0B 00 0400 0600060000006400→BLE MCU←78 03 00 0400←BLE	Setup connIntvMin=7.5ms, connIntvMax=7.5ms, latency=0, timeout=1s, read API_Status to update result
	MCU→77 05 00 0400 0600→BLE MCU←78 03 00 0400←BLE	Setup connIntvMin=7.5, other parameters keep the same
Read Action	MCU→77 03 00 0400→BLE MCU←78 0B 00 0400 0600060000006400←BLE	Read the current connection interval value

4.5 API_BTName

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
BTName (RW, swApp)	0x0005	b[0]: evtWithValue b[7:1]: —	btName (byte≤31)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the BLE device name. A write action can setup partial value, the remaining part will be kept in RAM. The btName will be used in advData (advertising state) and Generic Access Profile: Device Name (0x2A00, connection state). BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM and auto restart advertising. • Connection: Access value in RAM and Device Name (0x2A00).

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
btName (byte≤31)	ASCII	Max size=31 bytes Default=BC7701_(y)_(swAppVersion) y=0 when run in swApp1 y=1 when run in swApp2 swAppVersion format=0x000701xx (xx will be increased when update)

Example:		
Write Action	MCU→77 05 00 0500 4142→BLE MCU←78 03 00 0500←BLE	Setup btName="AB"
Read Action	MCU→77 03 00 0500→BLE MCU←78 05 00 0500 4142←BLE	Read the previous setup value

4.6 API_BTAddr

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
BTAddr (RW, swApp)	0x0006	b[0]: evtWithValue b[7:1]: —	btAddr (6-byte) (btAddrType (1-byte))
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the BLE device address. A write action can setup partial value, the remaining part will be kept in RAM. Valid before advCtrl=1, must be reset and initialized again if in advertising state. Default btAddr algorithm: • If btAddr has not been setup before, use the internal btAddr. • If btAddr has been setup before API_AdvCtrl, use the setup value. • Internal btAddr selection: If FT (Final Test) before, use it first, otherwise use bin default address (date). BLE action in each state: • Default: Access value in RAM and can be setup when RF is turned on. • Advertising: Access value in RAM but cannot runtime update. • Connection: Access value in RAM but cannot runtime update.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
btAddr (6-byte)	0x00~0xFF	Bluetooth device address, LSB first Default:Release date or MP_BD_ADDR
(btAddrType (1-byte))	0x00	Bluetooth device address type 0x00: Static address (Default)

Example:		
Write Action	MCU→77 09 00 0600 112233445566→BLE MCU←78 03 00 0600←BLE	Setup btAddr=0x665544332211
Read Action	MCU→77 03 00 0600→BLE MCU←78 0A 00 0600 11223344556600←BLE	Read the previous setup value

4.7 API_AdvCtrl

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
AdvCtrl (RW, swApp)	0x0007	b[0]: evtWithValue b[7:1]: —	advCtrl (1-byte)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to enable/disable advertising. A write action can setup partial value, the remaining part will be kept in RAM. HCI_Reset (CmdType1) auto setup advCtrl=1. After advCtrl=0→1, the BLE will reset (cost 2ms, do not send command during this period) and initialize the RF (calibration cost 600~800ms, no response during this period). BLE action in each state: • Default: If advCtrl=0→1, the BLE will reset and initialize the RF (calibration) and enter the Advertising state. • Advertising: Restart advertising. • Connection: The BLE will reset and initialize the RF (without calibration) and enter the Advertising state.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
advCtrl (1-byte)	0x00~0x01	Advertising enable 0x00: No advertising (Default) 0x01: Advertising Others: —

Example:		
Write Action	MCU→77 04 00 0700 01→BLE MCU←78 03 00 0700←BLE	Start advertising
Read Action	MCU→77 03 00 0700→BLE MCU←78 04 00 0700 01←BLE	Read the previous setup value

4.8 API_AdvIntv

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
AdvIntv (RW, swApp)	0x0008	b[0]: evtWithValue b[7:1]: —	advIntvMin (2-byte) advIntvMax (2-byte) (advMap (1-byte))
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the advertising interval. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM, modify advertising interval and restart advertising. • Connection: Access value in RAM but not restart advertising.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
advIntvMin (2-byte)	0x0020~0x4000	unit=0.625ms, range=20ms~10.24s Default=0x00A0=160
advIntvMax (2-byte)	0x0020~0x4000	unit=0.625ms, range=20ms~10.24s Default=0x00A0=160
(advMap (1-byte))	0x01~0x07	b[0]: channel 37 b[1]: channel 38 b[2]: channel 39 Others: — Default=0x07

Example:		
Write Action	MCU→77 07 00 0800 A000A000→BLE MCU←78 03 00 0800←BLE	Setup advertising interval=0x00A0=100ms, keep advMap. advIntvMin=0x00A0=100ms advIntvMax=0x00A0=100ms advMap=7
Read Action	MCU→77 03 00 0800→BLE MCU←78 08 00 0800 A000A00007←BLE	Read the previous setup value

4.9 API_AdvData

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
AdvData (RW, swApp)	0x0009	b[0]: evtWithValue b[3:1]: — b[4]: noBtName b[7:5]: —	advData (byte≤0x1F)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the advertising data. The default advData contains flag+btName. If (advData length+btName length+2)≤0x1F, btName will auto append, CmdFlag_b[4]: noBtName=1 can disable it. BLE action in each state: • Default: Access value in RAM. • Advertising: Modify the advertising data, restart advertising. • Connection: Access value in RAM, not restart advertising.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
	b[4]: noBtName	Not auto append btName when total length≤0x1F	
	b[7:5]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
advData (byte≤0x1F)	ADtype	If 1st byte=0x00, it means use the default advData (flag+btName). If 1st byte=0xFF, it means use the empty advData. Default=020106+btName ADtype The default advData will auto append btName if length(advData+btName)≤0x1F and advData does not contain ADtype=0x09 (CmdFlag[4]: noBtName=1 can disable it, also setup codeFeature_b[2]: AdvDataNotAutoAppendBTName=1)

Example:		
Write Action	MCU→77 06 10 0900 020106→BLE MCU←78 03 00 0900←BLE	Setup the advertising data=020106 advData=020106 and not show btName 02: length=0x02 01: type=flag 06: value b[1]: LE General Discoverable Mode b[2]: BR/EDR Not Supported
Read Action	MCU→77 03 00 0900→BLE MCU←78 06 00 0900 020106←BLE	Read the previous setup value

4.10 API_ScanData

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
ScanData (RW, swApp)	0x000A	b[0]: evtWithValue b[7:1]: —	scanData (byte≤0x1F)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the scan response data. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM and modify scan response data, restart advertising. • Connection: Access value in RAM but not restart advertising.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
scanData (byte≤0x1F)	ADtype	If 1st byte=0x00, it means use the default scanData (none). If 1st byte=0xFF, it means use the empty scanData. Default=none

Example:		
Write Action	MCU→77 07 00 0A00 03020F18→BLE MCU←78 03 00 0A00←BLE	Setup the scan response data=03020F18 scanData=03020F18 03: length=0x03 02: type=Incomplete List of 16-bit Service Class UUIDs 0F18: value 0x180F: Battery service
Read Action	MCU→77 03 00 0A00→BLE MCU←78 07 00 0A00 03020F18←BLE	Read the previous setup value

4.11 API_TxPwr

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
TxPwr (RW, swApp)	0x000B	b[0]: evtWithValue b[7:1]: —	txPwr (1-byte)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the RF TX power. The txPwr can only be set when advCtrl=0. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM but cannot runtime update. • Connection: Access value in RAM but cannot runtime update.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
txPwr (1-byte)	0x00~0x0F	RF TX power Default=0x0C

Example:		
Write Action	MCU→77 04 00 0B00 08→BLE MCU←78 03 00 0B00←BLE	Setup the RF TX power=0x08 txPwr=0x08
Read Action	MCU→77 03 00 0B00→BLE MCU←78 04 00 0B00 08←BLE	Read the previous setup value

4.12 API_CryStalOffset

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Value**
CrystalOffset (RW, swApp)	0x000E	b[0]: evtWithValue b[7:1]: —	crystalOffset (1-byte)
		EvtStatus	
		b[3:0]: ErrorCode b[7:4]: —	

Description:
Use to setup the external high crystal offset. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM and runtime setup crystal offset. • Connection: Access value in RAM and runtime setup crystal offset.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
crystalOffset (1-byte)	0x00~0x0F	Crystal offset Default=0x04 Ex: RF output frequency offset 0: 0kHz 1: 10kHz 2: 22kHz 3: 33kHz 4: 46kHz 5: 60kHz 6: 75kHz 7: 95kHz 8: 107kHz 9: 132kHz A: 157kHz B: 187kHz C: 220kHz D: 263kHz E: 384kHz F: 413kHz Others: —

Example:		
Write Action	MCU→77 04 00 0E00 07→BLE MCU←78 03 00 0E00←BLE	Setup crystal offset=0x07 crystalOffset=0x07
Read Action	MCU→77 03 00 0E00→BLE MCU←78 04 00 0E00 07←BLE	Read the previous setup value

4.13 API_PeerBTAddr

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Write Value**
PeerBtAddr (RO, swApp)	0x000F	b[7:0]: —	—
		EvtStatus	**Read Value**
		b[3:0]: ErrorCode b[7:4]: —	peerBtAddr (6-byte) peerBtAddrType (1-byte)

Description:
Use to get the peer device address. Valid when connected. If not in connection state, get all 0 in value field. BLE action in each state: • Default: Read all 0. • Advertising: Read all 0. • Connection: Read the connected device address/type.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
peerBtAddr (6-byte)	0x00000000~0xFFFFFFFF	Connected device address Default=0
peerBtAddrType (1-byte)	0~1	Connected device address type 0: Static address (Default) 1: Random address Others: —

Example:		
Write Action	MCU→77 09 00 0F00 010203040506→BLE MCU←78 03 03 0F00←BLE	Write not support
Read Action	MCU→77 03 00 0F00→BLE MCU←78 0A 00 0F00 AABBCCDDEEFF 01←BLE	Read to get connected device address/type peerBtAddr=0xFFEEDDCCBBAA peerBtAddrType=01=random address
	MCU→77 03 00 0F00→BLE MCU←78 0A 00 0F00 000000000000 00←BLE	No connection gets all 0.

4.14 API_Feature

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
Feature (RW, swBoot/swApp)	0x0010	b[0]: evtWithValue b[3:1]: — b[4]: orAction b[5]: andAction b[7:6]: —	codeFeature (4-byte)	
		EvtStatus		
		b[3:0]: ErrorCode b[7:4]: —		

Description:
Use to setup function. BLE action in each state: • Default: Access value in RAM and trigger feature. • Advertising: Access value in RAM and trigger feature. • Connection: Access value in RAM and trigger feature.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[3:1]:	—	
	b[4]: orAction	codeFeature will OR with RAM	
	b[5]: andAction	codeFeature will AND with RAM	
	b[7:6]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:6]	—	

Value:		
Name	**Range**	**Description**
codeFeature (4-byte)	b[0]: —	Keep 0
	b[1]: service0xFFF0Transparent	(swApp supported, for OTA use only) 1=Disable parse UART, all UART TX/RX map to service 0xFFF0 as BLE transparent (Disable by WDT reset)
	b[2]: advDataNotAutoAppendBTName	(swApp supported) 1=Not auto append BTName in advertising data
	b[3]: swParaUpdateForceReset	(swApp supported, autoClr, for TEST use only) 1=Read valid swPara to force to reset
	b[4]: swParaUpdate	(swApp supported, autoClr) 1=Save swPara in Flash (auto setup swParaErase=1)
	b[5]: swParaErase	(swBoot/swApp supported, autoClr) 1=Erase swPara in Flash

Value:		
Name	**Range**	**Description**
codeFeature (4-byte)	b[6]: flashWriteEnable	(swApp supported, for TEST use only) 1=Enable Flash write all block when UART update, default cannot write self-block
	b[7]: autoSendSatus	(swApp supported) 1=send status when change
	b[8]: —	Keep 0
	b[9]: rfCalibDone	(swApp supported, autoSet) 1=RF calibration done autoSet after calibration done, if cleared to 0, it will force calibration at next RF turned off→on
	b[10]: —	Keep 0
	b[11]: —	Keep 0
	b[12]: external32k	(swApp supported) Default to use internal 32k 1=use external 32k 0=use internal 32k
	b[13]: discNotReset	(swApp supported) 1=disconnection does not reset 0=disconnection will force a reset (Default)
	b[14]: testModeNotSleep	(swApp supported) 1=test mode not enter Sleep 0=test mode enter Sleep (Default)
	b[23:15]	Keep 0
	b[24]: onlyFailEvt	(swBoot/swApp supported) 1=response fail event only 0=response any event (Default)
	b[25]: rfCalibForce	(swApp supported) 1=Force calibration at next RF turned off→on (Default RF calibration will only do once after advCtrl=0→1)
	b[26]: test	(swBoot/swApp supported, for TEST use only) 1=API test mode, use to check UART interaction
	b[27]: —	Keep 0
	b[30: 28]: clockOut	(swApp supported) 3 bits select clock out mode b010: (HW) LSE output 32kHz to UART2_TX_PB6 Others: —
	b[31]: —	Keep 0

Example:		
Write Action	MCU→77 07 10 1000 04000000→BLE MCU←78 03 00 1000←BLE	OR bit advDataNotAutoAppendBTName=1 to codeFeature in RAM. Ex: original=0x00000000, after OR action=0x00000004
Read Action	MCU→77 03 00 1000→BLE MCU←78 07 00 1000 04000000←BLE	Read the previous value in RAM.

4.15 API_Version

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Command Value**
Version (RO, swBoot/swApp)	0x0020	b[3:0]: — b[4]: showAllVersion b[5]: — b[6]: showRunningVersion b[7]: showDate	(targetAddr (4-byte))
		EvtStatus	**Event Value**
		b[3:0]: ErrorCode b[7:4]: —	(swBootVersion (4-byte)) (swStackVersion (4-byte)) (swApp1Version (4-byte)) (swApp2Version (4-byte)) runningVersion (4-byte) runningAddr (4-byte) runningSize (4-byte) runningFeature (4-byte) (date (6-byte))

Description:

Use to get version (version is saved in ROM and cannot be changed).
If CmdFlag[7:4]=0, auto setup showRunningVersion=1.
If the showAllVersion/showRunningVersion/showData are setup simultaneously, event will show in the same order with CmdFlag.
The showAllVersion bit will calculate block checksum, it will cost greater than 100ms.
The swBootVersion/swStackVersion/swApp1Version/swApp2Version will force to check block checksum, return 0xFFFFFFFF when checksum!=0.
Value=0xFFFFFFFF means not exist.
BLE action in each state:

- Default: Read version in ROM.
- Advertising: Read version in ROM.
- Connection: Read version in ROM.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[3:0]	—	
	b[4]: showAllVersion	Event included swBootVersion/swStackVersion/ swApp1Version/swApp2Version	
	b[5]	—	
	b[6]: showRunningVersion	Event included runningVersion/runningAddr/runningSize/ runningFeature	
	b[7]: showDate	Event included date	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
targetAddr (4-byte)	0x00000000 ~0x00027C00	Use to retrieve specific running block version, ex: current running in swApp1, retrieve swApp2 version. Command Value empty means use retrieve version from the current running block (swBoot, swApp1 or swApp2).
swBootVersion (4-byte)	0x00000000 ~0xFFFFFFFF	Exist when CmdFlag_b[4]: showAllVersion=1 Format=0x800107xx
swStackVersion (4-byte)	0x00000000 ~0xFFFFFFFF	Exist when CmdFlag_b[4]: showAllVersion=1 Format=0x400107xx 0xFFFFFFFF means not exist
swApp1Version (4-byte)	0x00000000 ~0xFFFFFFFF	Exist when CmdFlag_b[4]: showAllVersion=1 Format=0x000107xx 0xFFFFFFFF means not exist
swApp2Version (4-byte)	0x00000000 ~0xFFFFFFFF	Exist when CmdFlag_b[4]: showAllVersion=1 Format=0x000107xx 0xFFFFFFFF means not exist
runningVersion (4-byte)	0x00000000 ~0xFFFFFFFF	Exist when CmdFlag_b[6]: showRunningVersion=1 Should be swBoot/swApp1/swApp2 version format
runningAddr (4-byte)	0x00000000 ~0x00027C00	Exist when CmdFlag_b[6]: showRunningVersion=1 Should be swBoot/swApp1/swApp2 start address
runningSize (4-byte)	0x00000000 ~0x00028000	Exist when CmdFlag_b[6]: showRunningVersion=1 Should be swBoot/swApp1/swApp2 block size
runningFeature (4-byte)	0x00000000 ~0xFFFFFFFF	Exist when CmdFlag_b[6]: showRunningVersion=1 each bit meaning: b[4:0]: — b[5]: Service_FFF0 b[6]: Service_Battery b[7]: Service_DIS (deviceInformation) b[15:8]: — b[16]: Support DTM b[17]: Support FCC b[31:18]: —
date (6-byte)	0x000000000000 ~0xFFFFFFFFFFFF	Exist when CmdFlag_b[7]: showDate=1 Show in MSB and Hex format, ex: 202110141102: year=2021, month=10, date=14, hour=11, min=02

Example:		
Write Action	MCU→77 07 00 2000 000E0200→BLE MCU←78 13 00 2000 00070100 000E0200 006E0000 E0000300←BLE	Read the swApp2 version. targetAddr=0x00020E00 (swApp2) runningVersion=0x00010700 runningAddr=0x00020E00 runningSize=0x00006E00=27.5K byte runningFeature=0x000300E0 (Service_FFF0/Service_Battery/Service_DIS/Support DTM/Support FCC)
Read Action	MCU→77 03 00 2000→BLE MCU←78 13 00 2000 00070180 00000000 00200000 00000000←BLE	Read the current running version. runningVersion=0x80010700 runningAddr=0x00000000 runningSize=0x00002000=8K byte runningFeature=0x00000000
	MCU→77 03 10 2000→BLE MCU←78 13 00 2000 00070180 00070140 00070100 00070100←BLE	Read all versions, 0xFFFFFFFF means not exist CmdFlag=0x10=showAllVersion swBootVersion=0x80010700 swStackVersion=0x40010700 swApp1Version=0x00010700 swApp2Version=0x00010700
	MCU→77 03 D0 2000→BLE MCU←78 29 00 2000 00070180 00070140 00070100 00070100 00070100 00A00100 006E0000 E0000300 202110141601←BLE	Read allVersion/runningVersion/date CmdFlag=0xD0=showAllVersion+showRunningVersion+showDate swBootVersion=0x80010700 swStackVersion=0x40010700 swApp1Version=0x00010700 swApp2Version=0x00010700 runningVersion=0x00010700 runningAddr=0x0001A000 runningSize=0x00006E00=27.5K byte runningFeature=0x000300E0 (Service_FFF0/Service_Battery/Service_DIS/Support DTM/Support FCC) date=2021/10/14 16:01

4.16 API_PowerSaving

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
PowerSaving (RW, swApp)	0x0025	b[0]: evtWithValue b[7:1]: —	modeBLE (1-byte) (reserved (1-byte)) (reserved (1-byte)) (reserved (1-byte)) (modeHost (1-byte)) (delayHost (1-byte)) (sizeHost (1-byte)) (reserved (1-byte))	
		EvtStatus		
		b[3:0]: ErrorCode b[7:4]: —		

Description:
Use to setup power saving mode in BLE, must turn on advertising first and then enter power saving mode. After waking up from Deep Sleep, an extra 2ms is required to enable UART HW, 30ms to leave Deep Sleep state. Using dummy bytes to wake up BLE requires at least 100μs long dummy bytes then keeps high for at least 2ms. After RX dummy bytes (0x00 n byte) for 2ms~30ms, can RX max 16 bytes (depend on hardware). After RX dummy bytes (0x00 n byte) for 30ms, can RX max 0x100 byte (depend on firmware). BLE can also send dummy bytes to wake up Host. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
mode (1-byte)	0x00~0x01, 0x15	0x00: Sleep: Running between Sleep and wake-up (Default) 0x01: Deep Sleep: Running between Sleep, Deep Sleep and wake-up, must send dummy bytes (at least 100μs long) to wake up 0x15: Power Down: Keep in Deep Sleep, wake-up will force to reset, all setting in RAM reset to default value Others: Useless
(reserved (1-byte))	0x00	Fixed at 0
(reserved (1-byte))	0x00	Fixed at 0
(modeHost (1-byte))	0x00~0x01	0x00: Master not in Sleep mode (Default) 0x01: Master in Sleep mode needs slave dummy wake-up Others: Useless
(delayHost (1-byte))	0x00~0xFF	Delay between dummy wake-up and event, unit=ms (accuracy=0.5ms). Due to BLE scheduling, may be delay more than setting.
(sizeHost (1-byte))	0x00~0xFF	Dummy wake-up byte number
(reserved (1-byte))	0x00	Fixed at 0

Example:		
Write Action	MCU→77 04 00 2500 01→BLE MCU←78 03 00 2500 ←BLE	Set BLE in Deep Sleep mode
	MCU→77 08 00 2500 0000000000→BLE MCU←78 03 00 2500←BLE	Set both BLE/Host in power saving mode modeBLE=Sleep modeHost=not in Sleep
	MCU→77 0B 00 2500 01 00 00 00 01 10 03 00→BLE MCU←78 03 00 2500←BLE	Set both BLE/Host in power saving mode modeBLE=Deep Sleep modeHost=Sleep delayHost=16ms sizeHost=3byte
Read Action	MCU→77 03 00 2500→BLE MCU←78 0B 00 2500 00 00 00 00 00 02 01 00←BLE	Read the previous value in RAM modeBLE=Sleep modeHost=not in Sleep delayHost=2ms sizeHost=1byte

4.17 API_InterfaceSpeed

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
InterfaceSpeed (RW, swBoot/swApp)	0x0026	b[0]: evtWithValue b[7:1]: —	urBaudrate (4-byte)	
		EvtStatus		
		b[3:0]: ErrorCode b[7:4]: —		

Description:
Use to set UART baud rate. The baud rate will be changed after the BLE TX event has finished. Due to BLE/UART interrupt collision, if advCtrl=1, max supported urBaudrate should limit to 115200. BLE action in each state: • Default: Access value in RAM and update baud rate after TX event has finished. • Advertising: Access value in RAM and update baud rate after TX event has finished. • Connection: Access value in RAM and update baud rate after TX event has finished.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
urBaudrate (4-byte)	1953~3200000	Baud rate calculate equation: 16000000/(8191~5)=1953~3200000 A read action will get current urBaudrate A write action will force to switch to the setting value in range Default=115200

Example:		
Write Action	MCU→77 07 01 2600 00C20100→BLE MCU←78 07 00 2600 00C20100←BLE	Set baud rate=115200 CmdFlag_b[0]: evtWithValue=1 can get the previous setup value, host can also setup baud rate by event.
Read Action	MCU→77 03 00 2600→BLE MCU←78 07 00 2600 00C20100←BLE	Read the previous value in RAM.

4.18 API_InterfaceSpeedMax

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
InterfaceSpeedMax (RW, swBoot/swApp)	0x0027	b[0]: evtWithValue b[7:1]: —	urBaudrateMax (4-byte)	
		EvtStatus		
		b[3:0]: ErrorCode b[7:4]: —		

Description:
Use to setup UART baud rate to max. The baud rate will be changed after the BLE TX event has finished. Due to BLE/UART interrupt collision, if advCtrl=1, max supported urBaudrate should limit to 115200. BLE action in each state: • Default: Access value in RAM and update baud rate after TX event has finished. • Advertising: Access value in RAM and update baud rate after TX event has finished. • Connection: Access value in RAM and update baud rate after TX event has finished.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:6]	—	

Value:		
Name	**Range**	**Description**
urBaudrateMax (4-byte)	3200000	Baud rate max fixed in BLE A read action will get the fixed urBaudrateMax (value depends on FW) A write action will force to switch to max UART baud rate and ignore the value content Default=3200000

Example:		
Write Action	MCU→77 07 00 2700 00C20100→BLE MCU←78 03 00 2700 ←BLE	Set baud rate=max baud rate, the value field just use to create a write action, the value content is meaningless
Read Action	MCU→77 03 00 2700→BLE MCU←78 07 00 2700 00D43000←BLE	Read urBaudratMax=3200000

4.19 API_Reset

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
Reset (WO, swBoot/swApp)	0x0028	b[7:0]: — **EvtStatus** b[3:0]: ErrorCode b[7:4]: —	rstPara (32-byte)	

Description:
Use to reset chip. When the host RX event finishes (watchdog reset), it needs extra 60ms to finish the initialization flow; When the host RX event finishes (other resets), it needs extra 5ms to finish the initialization flow, do not send command within this period. BLE action in each state: • Default: Depend on value field, if a WDT reset occurs, return to the default state to wait for advCtrl=1. • Advertising: Depend on value field, if a WDT reset occurs, return to the default state to wait for advCtrl=1, maybe stop/restart advertising. • Connection: Depend on value field, if a WDT reset occurs, return to the default state to wait for advCtrl=1, force disconnection.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[7:0]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:6]	—	

Value: ("W" denoted as Word, "B" denoted as Byte, "b" denoted as bit)		
Name	**Description**	
rstPara (4-byte)	If B[0]=0x00, need extra 60ms delay	Watchdog reset=HW reset, return to the default state. If 1st byte=0, it will ignore the following bytes
	If Value=0x36323032	Firmware resets to the default state
	If W[0]=0x36323032, need extra 1ms delay	W[1]: rstAct b[7:0]: keep 0 b[8]: rfTurnOn: 1: turn on RF (calibration depends on API_Feature) b[9]: notInitialParameter: 1: not initial parameter, keep previous value b[10]: keep 0 b[11]: keep 0 b[12]: keepInswBoot: 1: keep in swBoot b[13]: jmp: 1: jump to another address b[14]: disableAnotherSwApp: 1: disable another swApp b[15]: keep 0 b[16]: keep 0 b[17]: rfCalibForce: 1: force RF calibration b[18]: ext32k: 1: fixExt32k b[31:19]: keep 0
		If W[1]: rstAct=0x00001000 (jump to swBoot), the remaining words will be as follows: W[2]: (urBaudrate) W[3]: (urIndex, 0=UART1, 1=UART2) W[6:4]: — W[7]: (swApp start address, for internal use, do not modify)
		If W[1]: rstAct=0x00002000(jump to swApp), the remaining words will be as follows: W[2]: (swApp address) W[3]: (initial, 0=no, 1=yes) W[6:4]: — W[7]: (swApp start address, for internal use, do not modify)
		If W[1]: rstAct=0x00006000(jump to swApp+disable anther swApp), the remaining words will be as follows: W[2]: (swApp address to be jumped) W[3]: (swApp address to be disabled) W[6:4]: — W[7]: (swApp start address, for internal use, do not modify)
		If W[1]: rstAct=0x00002300(jump to swApp/rfTurnOn/ notInitialParameter), the remaining words will be as follows: W[2]: (swApp address) W[3]: (initial, 0=no, 1=yes) W[6:4]: — W[7]: (swApp start address, for internal use, do not modify)

Example:		
Write Action	MCU→77 0B 00 2800 32303236 00100000→BLE MCU←78 03 00 2800←BLE	Jump to swBoot
	MCU→77 13 00 2800 32303236 00200000 00A00100 00000000→BLE MCU←78 03 00 2800←BLE	Jump to swApp1 (0x0001a000)
	MCU→77 13 00 2800 32303236 00230200 00A00100 00000000→BLE MCU←78 03 00 2800←BLE	Jump to swApp1 (0x0001a000) / rfTurnOn (keep calibration result) / notInitialParameter / rfCalibForce After reset, force calibration again.
	MCU→77 13 00 2800 32303236 00200000 000E0200 00000000→BLE MCU←78 03 00 2800←BLE	Jump to swApp2 (0x00020e00)
	MCU→77 04 00 2800 00→BLE MCU←78 03 00 2800←BLE	Watchdog reset

Example:		
Read Action	MCU→77 03 00 2800→BLE MCU←78 03 03 2800 ←BLE	Not support

4.20 API_WhiteList

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
WhiteList (RW, swApp)	0x002A	b[0]: evtWithValue b[3:1]: — b[4]: whiteListClrFirst b[7:5]: —	address (6-byte) (addressMask (6-byte))	
		EvtStatus		
		b[3:0]: ErrorCode b[7:1]: —		

Description:
Use to setup whitelist. If a whitelist not exists, all devices can be connected to the BLE. (Default) If a whitelist exists, only device in whitelist can be connected to the BLE. If the peer device uses random address, the function may be useless. (Ref.: API_BTAddr: btAddrType) If the value only contains address, addressMask will be auto set to 0xFFFFFFFFFFFF. Max storage 4 pair whitelist address+addressMask. The 5th pair will be wraparound to the 1st pair. BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[3:1]	—	
	b[4]: whiteListClrFirst	Remove all whitelists before setup	
	b[7:5]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:6]	—	

Value:		
Name	**Range**	**Description**
address (6-byte)	0x000000000000~ 0xFFFFFFFFFFFF	BtAddr 6 byte, LSB→MSB
(addressMask (6-byte))	0x000000000000~ 0xFFFFFFFFFFFF	BtAddr mask 6 byte, LSB→MSB. BtAddr mask will do "AND action" with BtAddr. If not set addressMask, auto set to 0xFFFFFFFFFFFF. A write action can be optional, a read action must exist. Ex: BtAddr=0x112233445566, BtAddrMask=0x00FFFFFFFFFF, then the device with address between 0x002233445566 to 0xFF2233445566 can be connected to the BLE.

Example:		
Write Action	MCU→77 09 00 2A00 665544332211→BLE MCU←78 03 00 2A00 ←BLE	Add address to whitelist. address=0x112233445566 addressMask=0xFFFFFFFFFFFF
Read Action	MCU→77 03 00 2A00→BLE MCU←78 0F 00 2A00 665544332211 FFFFFFFFFFFF←BLE	Read the previous value in RAM. address=0x112233445566 addressMask=0xFFFFFFFFFFFF Only device btAddr=0x112233445566 can be connected to the BLE

4.21 API_IP

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
IP (WO, swBoot/swApp)	0x0040	b[0]: evtWithValue b[3:1]: — b[7:4]: ipAction 0x00: setValue 0x01: setMap 0x02: setPara 0x08: rwMem Others: —	ipIndex (1-byte) ipVersion (3-byte) ipValue (n-byte)	
		EvtStatus		
		b[3:0]: ErrorCode b[7:4]: —		

Description:
Use to setup control command/event flow. Redirect Cmd/Evt to different ip pair, if ipVersin is not matched with firmware version, it will be handled as data. API_IP only use in OTA Ex: BLE maps to UART1 (BLE RX will TX to UART1, UART1 RX will TX to BLE). UART1 maps to UART2 (UART1 RX will TX to UART2, UART2 RX will TX to UART1). BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[7:4]	ipAction 0x00: setValue 0x01: setMap 0x02: setPara 0x08: rwMem Others: —	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:6]	—	

Value:		
Name	**Range**	**Description**
ipIndex (1-byte)	0x00~0x04	IP pair index The internal RAM can save 4 pairs of route map Default setup will be: Pair0: use for UART0, ipSrc=0, ipDes=0 (do nothing) Pair1: use for UART1, ipSrc=0, ipDes=0 (do nothing) Pair2: use for BLE, ipSrc=BLE, ipDes=1st RX command UART (BLE route to UART) Pair3: reserved, ipSrc=0,ipDes=0 (do nothing)
ipVersion (3-byte)	any	Check command RX target, map to Version_B[3:1], if not match self version's format should bypass it Ex: swApp version format=0x000107xx, xx will be ignore
ipValue (n-byte)	any	If ipAction=0x00(setValue): this field should be value to TX. If ipAction=0x01(setMap): this field should be ip map table, should contain ipSrc+ipDes (2 words), setup route from ipSrc to ipDes, the current valid ip value is as follows: UART1: 0x40041000 (UART) UART2: 0x40002000 (UART) BLE: 0x000000BB (air) If ipAction=0x02 (setPara): this field should contain ip (1-word) and baudrate (1-word). If ipAction=0x08 (rwMem): this field should contain ip (1-word), address (1-word) and optional data (n-word).

Example:		
Write Action	MCU→77 0F 10 4000 02070100 BB000000 00100440 →BLE MCU←78 03 00 4000 ←BLE	Set map pair BLE-UART1 ipAction=0x1 (setMap) ipIndex=0x02 ipVersion=0x000107xx (swApp version high 3 bytes must match) ipValue=0x000000BB (BLE) map to 0x40041000 (UART1)
	MCU→77 0F 00 4000 02070100 11223344 55667788 →BLE MCU←78 03 00 4000 ←BLE	Bypass data, BLE RX data is sent to UART1 ipAction=0x0 (setValue) ipIndex=0x02 ipVersion=0x000107xx (swApp version high 3 bytes must match) ipValue=0x4433221188776655 send to UART1
	MCU→77 0F 20 4000 00070100 00100440 00C20100 →BLE MCU←78 03 00 4000 ←BLE	Set UART1 baud rate=115200 ipAction=0x2(setPara) ipIndex=0x00, useless here ipVersion=0x000107xx (swApp version high 3 bytes must match) ipValue=0x40041000 (UART1) baud rate=0x0001C200
	MCU→77 0F 80 4000 00070100 00100040 04030201 →BLE MCU←78 03 00 4000 ←BLE	Set address(0x40001000)=0x01020304 ipAction=0x08(rwMem) ipIndex=0x00, useless here ipVersion=0x000107xx (swApp version high 3 bytes must match) ipValue=0x01020304 will be assigned to address=0x40001000
Read Action	MCU→77 03 00 4000 →BLE MCU←78 03 03 4000 ←BLE	Read action not support

4.22 API_GPIO

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:				
Command	**Type**	**CmdFlag**	**Command Value**	**Event Value**
GPIO (RW, swApp)	0x0050	b[0]: evtWithValue b[3:1]: ioCtrlType b[4]: orAction b[5]: andAction b[7:6]: ioPort	gpioValue (n-byte)	
		EvtStatus		
		b[3:0]: ErrorCode b[7:4]: —		

Description:

Use to control GPIO.
If gpioValue's length>1, the gpioValue field format is gpioValue+CmdFlag+gpioValue+CmdFlag.
GPIO default state=GPIO mode/not input/not output/not pullHigh/not pullLow.
This API cannot effect PA0/PA1(UART1) and PB6/PB7(UART2).
(orAction) prior than (andAction)
In Deep Sleep keep I/O status
In Power Down setup I/O to floating
BLE action in each state:

- Default: Access value in control register.
- Advertising: Access value in control register.
- Connection: Access value in control register.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[0]: evtWithValue	Event included value 1: Force Evt to read action format 0: Useless	
	b[3:1]: ioCtrlType	0: InputValue pin input value 1: portDirection: 1=input, 0=output 2: portPull: 1=pullHigh, 0=pullLow 3: portIoEn: 1=I/O, 0=AFIO 4: portPullEn: 1=enable, need to reset portDirection and portPull 5: OutputValue output register value 6: — 7: Delay: delay() function, Write Only	
	b[4]: orAction	OR action (valid when write)	
	b[5]: andAction	AND action (valid when write)	
	b[7:6]: ioPort	0: Port A 1: Port B 2: Not Support 3: Port D	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:6]	—	

Value:		
Name	**Range**	**Description**
gpioValue	any (1-byte)	Setup port value If ioCtrlType=0: portOuput (Write Action), portInput (Read Action) If ioCtrlType=1: portDirection If ioCtrlType=2: portPull If ioCtrlType=3: portIoEn If ioCtrlType=4: portPullEn If ioCtrlType=7: delay count in delay() function
	any (byte>1)	If gpioValue's length>1, the gpioValue field format is gpioValue+Cmd Flag+gpioValue+CmdFlag.

Example: GPIO		
Write Action	MCU→77 04 D2 5000 FF→BLE MCU←78 03 00 5000←BLE	gpioValue length=1, use to setup portDirection=0xFF(input) CmdFlag_b[0]: evtWithValue=0: event not contained value CmdFlag_b[3:1]: ioCtrlType=portDirection CmdFlag_b[4]: orAction=1=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port D gpioValue=0xFF (ioCtrlType=portDirection)
	MCU→77 06 D2 5000 FFD6FF→BLE MCU←78 03 00 5000←BLE	gpioValue length>1, use to setup portDirection= 0xFF(input)+portIoEn=0xFF(I/O) 1st CmdFlag=0xD2 CmdFlag_b[0]: evtWithValue=0: event not contained value CmdFlag_b[3:1]: ioCtrlType=portDirection CmdFlag_b[4]: orAction=1=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port D gpioValue=0xFF(ioCtrlType=portDirection) 2nd CmdFlag=0xD6 CmdFlag_b[0]: evtWithValue=0: event not contained value CmdFlag_b[3:1]: ioCtrlType=portIoEn CmdFlag_b[4]: orAction=1=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port D gpioValue=0xFF(ioCtrlType=portIoEn)
Read Action	MCU→77 03 C0 5000→BLE MCU←78 04 00 5000 1E←BLE	Read GPIO port D, value=0x1E CmdFlag_b[0]: evtWithValue=0: event not contained value CmdFlag_b[3:1]: ioCtrlType=portValue CmdFlag_b[4]: orAction=0 CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port D gpioValue=0x1E(ioCtrlType=portValue)
	MCU→77 06 D3 5000 F0D70F→BLE MCU←78 04 00 5000 F0←BLE	1st CmdFlag=0XD3 CmdFlag_b[0]: evtWithValue=1: event contained value CmdFlag_b[3:1]: ioCtrlType=portDirection CmdFlag_b[4]: orAction=1=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port D gpioValue=0xF0(ioCtrlType=portDirection) 2nd CmdFlag=0xD7 CmdFlag_b[0]: evtWithValue=1: event contained value CmdFlag_b[3:1]: ioCtrlType=portIoEn CmdFlag_b[4]: orAction=1=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port D gpioValue=0x0F(ioCtrlType=portIoEn) The evtWithValue function can only supports the flag in front of the Type.

Example: Delay		
Write Action	MCU→77 04 0E 5000 02→BLE MCU←78 03 00 5000←BLE	Cmd=0x77(header)+0x04(length)+0x0E(flag→Delay)+0x5000(Type)+0x02(value) CmdFlag_b[0]: evtWithValue=0: event not contained value CmdFlag_b[3:1]: ioCtrlType=Delay CmdFlag_b[4]: orAction=0=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port A Value=0x02 Evt=0x78(header)+0x03(length)+0x00(status)+0x5000(Type)
	MCU→77 10 1A 5000 0C22F32AF72AFB0E011A081A04→BLE MCU←78 03 00 5000←BLE	Cmd=0x77(header)+0x10(length)+0x1A(flag)+0x5000(Type)+0x0C(value)+0x22(flag)+0xF3(value)+0x2A(flag)+0xF7(value)+0x2A(flag)+0xFB(value)+0x0E(flag→Delay)+0x01(value)+0x1A(flag)+0x08(value) +0x1A(flag)+0x04(value) Evt=0x78(header)+0x03(length)+0x00(status)+0x5000(Type)
	MCU→77 10 1B 5000 0C22F32AF72AFB0EFF1A081A04→BLE MCU←78 04 00 5000 XX←BLE	Cmd=0x77(header)+0x10(length)+0x1B(flag)+0x5000(Type)+0x0C(value)+0x22(flag)+0xF3(value)+0x2A(flag)+0xF7(value)+0x2A(flag)+0xFB(value)+0x0E(flag→Delay)+0xFF(value)+0x1A(flag)+0x08(value)+0x1A(flag)+0x04(value) Delay time: The value increases by about 1.8ms for each increment of 1 Evt=0x78(header)+0x04(length)+0x00(status)+0x5000(Type)+0xXX(ReadValue)
Read Action	MCU→77 04 0F 5000 03→BLE MCU←78 03 03 5000←BLE	Cmd=0x77(header)+0x04(length)+0x0F(flag→Delay)+0x5000(Type)+0x03(value) CmdFlag_b[0]: evtWithValue=1: event contained value CmdFlag_b[3:1]: ioCtrlType=Delay CmdFlag_b[4]: orAction=0=OR action CmdFlag_b[5]: andAction=0 CmdFlag_b[7:6]: ioPort=Port A Value=0x03 Evt=0x78(header)+0x03(length)+0x03(status)+0x5000(Type) Status=0x03 means the function of reading "delay command" is not supported

4.23 API_FCC

"RO" means read only. "RW" means read/write. "WO" means write only. "Value with ()" means optional.

Command:			
Command	**Type**	**CmdFlag**	**Write Value**
FCC (WO, swApp)	0x00CC	b[7:0]: —	dataMode (1-byte) frequency (1-byte) power (1-byte) packetType (1-byte) hopping (1-byte) dataRate (1-byte) rfMode (1-byte)
		EvtStatus	**Read Value**
		b[3:0]: ErrorCode b[7:4]: —	—

Description:
Use to set FCC test mode. If want to return to normal mode should do HW reset (nRST=0→1) or SW reset (API_Reset). Test flow: 1. FCC enter mode 2. FCC TX or RX BLE action in each state: • Default: Access value in RAM. • Advertising: Access value in RAM. • Connection: Access value in RAM.

CmdFlag/EvtStatus:			
Name	**Range**	**Description**	
CmdFlag (1-byte)	b[7:1]	—	
EvtStatus (1-byte)	b[3:0]: ErrorCode	0x00: _ST_SUCCESS_	0x04: _ST_PENDING_
		0x01: _ST_FAIL_	0x05: _ST_INVALID_
		0x02: _ST_UNKNOWN_	0x06: _ST_NOT_ENABLE_
		0x03: _ST_NOT_SUPPORT_	Others: —
	b[7:4]	—	

Value:		
Name	**Range**	**Description**
dataMode (1-byte)	0x00~0x01	0x00: sin 0x01: pn9 Others: enter FCC mode
frequency (1-byte)	0x02~0x50	0x02=frequency 2402MHz 0x50=frequency 2480MHz Others: —
power (1-byte)	0x00~0x0F	Map to API_TxPwr
packetType (1-byte)	0x00~0x0E	0x00=RX 0x01=(TX) DH1 0x02=(TX) DH3 0x03=(TX) DH5 0x04=(TX) 2DH1 0x05=(TX) 2DH3 0x06=(TX) 2DH5 0x07=(TX) 3DH1 0x08=(TX) 3DH3 0x09=(TX) 3DH5 0x0A=(TX) DM1 0x0B=(TX) DM3 0x0C=(TX) DM5 0x0D=(TX) HV1 0x0E=(TX) HV2 0x0E=(TX) HV3 Others: —
hopping (1-byte)	0x00~0x01	0x00=noHop 0x01=hop Others: —
dataRate (1-byte)	0x00	0:1MHz (valid when BLE mode) 1:2MHz (valid when BLE mode)
rfMode (1-byte)	0x00	0: — 1: BLE mode

Example:		
Write Action	MCU→77 0A 00 CC00 03000000000000→BLE MCU←78 03 00 CC00 ←BLE	FCC enter mode dataMode=0x03=enter FCC mode frequency=ignore power=ignore packetType=ignore hopping=ignore dataRate=ignore rfMode=ignore
	MCU→77 0A 00 CC00 00020001000000→BLE MCU←78 03 00 CC00 ←BLE	FCC TX 2402MHz dataMode=0x00=sin frequency=0x02=2402MHz power=0x00 packetType=0x01=TX DH1 hopping=0x00 dataRate=1MHz rfMode=0x00
	MCU→77 0A 00 CC00 00280001000000→BLE MCU←78 03 00 CC00 ←BLE	FCC TX 2440MHz dataMode=0x00=sin frequency=0x28=2440MHz power=0x00 packetType=0x01=TX DH1 hopping=0x00 dataRate=1MHz rfMode=0x00
	MCU→77 0A 00 CC00 00500001000000→BLE MCU←78 03 00 CC00 ←BLE	FCC TX 2480MHz dataMode=0x00=sin frequency=0x50=2480MHz power=0x00 packetType=0x01=TX DH1 hopping=0x00 dataRate=1MHz rfMode=0x00
	MCU→77 0A 00 CC00 00020000000000→BLE MCU←78 03 00 CC00 ←BLE	FCC RX 2402MHz dataMode=0x00=sin frequency=0x02=2402MHz power=0x00 packetType=0x00=RX hopping=0x00 dataRate=1MHz rfMode=0x00
	MCU→77 0A 00 CC00 01020C01000001→BLE MCU←78 03 00 CC00 ←BLE	FCC TX 2402MHz dataMode=0x01=pn9 frequency=0x02=2402MHz power=0x0C packetType=0x01=TX DH1 hopping=0x00 dataRate=1MHz rfMode=0x01=BLE mode
Read Action	MCU→77 03 00 CC00→BLE MCU←78 03 03 CC00←BLE	Read not support