

BM7701-00-1 應用範例

文件編號：AN0614TC

簡介

BM7701-00-1 是一個基於 SOC BC7701 晶片、BLE 透明傳輸的藍牙低功耗收發模組。該模組可以無線控制外部設備，支援雙向資料傳輸，適用於照明產品、醫療保健產品和家用電器。本文將介紹使用 Holtek 32-bit MCU HT32F52352 和 8-bit MCU HT66F2370 作為微控制器，控制模組 BM7701-00-1 修改藍牙參數並與手機 App 進行資料傳輸，可以實現按鍵發送數據、LED 燈顯示接收狀態。通過藍牙模組與手機 App 互相對傳的範例，展示藍牙模組雙向資料傳輸的能力，可幫助用戶靈活搭載到不同的電子產品中。

功能說明

通訊介面

BM7701-00-1 模組需要使用 UART 控制。UART 是一種通用串列數據總線，用於非同步。該總線雙向通訊可以實現全雙工傳輸和接收。BM7701-00-1 提供兩個 UART 通訊介面 UART1/2，本應用範例統一使用 UART1 介面。

Pin No.	Pin Name	Function Description
1	UART1_TX	BLE UART1_TX 串行數據輸出
2	UART1_RX	BLE UART1_RX 串行數據輸入
3	UART2_TX	BLE UART2_TX 串行數據輸出
4	UART2_RX	BLE UART2_RX 串行數據輸入

表 1. BM7701-00-1 UART 介面

RST_N=0→1(從低準位拉到高準位/上電重置)後，TX/RX I/O 必須在最初 60ms 保持固定，BM7701-00-1 會根據 I/O 狀態選擇 UART Baud Rate。然後就可以將 I/O 更改回 UART TX/RX。如下表 2 所示。

上電重置後 TX/RX I/O 在最初 60ms 保持固定	BM7701-00-1 TX=1 (UART1/2_TX)	BM7701-00-1 TX=0 (UART1/2_TX)
BM7701-00-1 RX=1 (UART1/2_RX)	Baud Rate = 115200 (default)	Baud Rate = 9600

表 2. UART Baud Rate 選擇

UART 格式

UART 數據格式需要滿足以下要求：

1. Baud rate : 1953bps~115200bps (BM7701-00-1)
2. Data bit : 8 bits
3. Parity check : No
4. Stop bit : 1 bit



圖 1. UART 時序圖

Command/Event 格式

BM7701-00-1 支援三種類型的 Command，包括 CmdType1、CmdType2 和 CmdType3。CmdType1 是藍牙規範中定義的 HCI。CmdType2 支持訪問 BLE Services。CmdType3 是設定 BM7701-00-1 的內部參數，包括 Tx Power、Crystal load、開關廣播、廣播間隔、連接間隔等等。MCU 通過 UART 向 BM7701-00-1 寫入 Command，BM7701-00-1 成功識別就會返回 Event。詳情請參考 BLE_API 文檔。本應用範例只用到 CmdType2 和 CmdType3；CmdType2 和 CmdType3 格式如下表。

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

表 3. CmdType2 格式

Description	Header (1-byte)	Length (1-byte)	CmdFlag/EvtStatus (1-byte, "b" denoted as bit)	Type (2-byte)	Value (n-byte)
Initiator → Responder	0x77	xx	CmdFlag: b[0:1]: BLE responses read Evt b[7:1]: Force Evt to read action format	As follows	Depend on Type
Responder → Initiator	0x78		EvtStatus: b[3:0]: ErrorCode b[7:4]: N/A		

表 4. CmdType3 格式

Example	
MCU → 77 04 00 0B00 08 → BM7701-00-1 MCU ← 78 03 00 0B00 ← BM7701-00-1	Set RF Tx power = 0x08 (CmdType3)
MCU → 77 04 00 0700 01 → BM7701-00-1 MCU ← 78 03 00 0700 ← BM7701-00-1	Start advertising (CmdType3)
MCU → 77 08 20 F2FF 01 02 03 04 05 → BM7701-00-1 MCU ← 78 03 20 F2FF ← BM7701-00-1	Send 0x0102030405 to phone (CmdType2)
MCU ← 78 08 00 F2FF 01 02 03 04 05 ← BM7701-00-1	Receive 0x0102030405 from phone (CmdType2)

表 5. 樣例

使用指南

本章節介紹了環境準備、操作說明、硬體說明、軟體說明、模組 API 命令函數介紹、藍牙參數修改等資訊，希望用戶通過瞭解應用範例，快速掌握 BM7701-00-1 模組的基本操作與系統架構。

環境準備

硬體

- HT32: 藍牙模組 BM7701-00-1、轉板 BCT-7701-001、開發板 BCE-GenTrx32-002(搭載 MCU HT32F52352)
- HT8: 藍牙模組 BM7701-00-1、轉板 BCT-7701-001、開發板 BCE-GenTrx8-001(搭載 MCU HT66F2370)

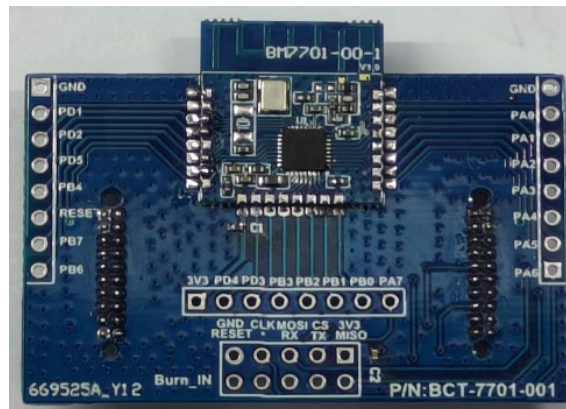


圖 2. 模組+轉板

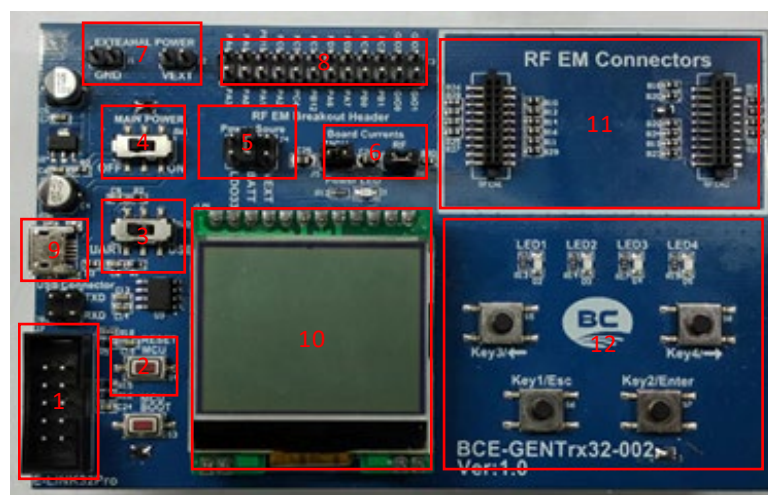


圖 3. 開發板(BCE-GENTrx32-002)

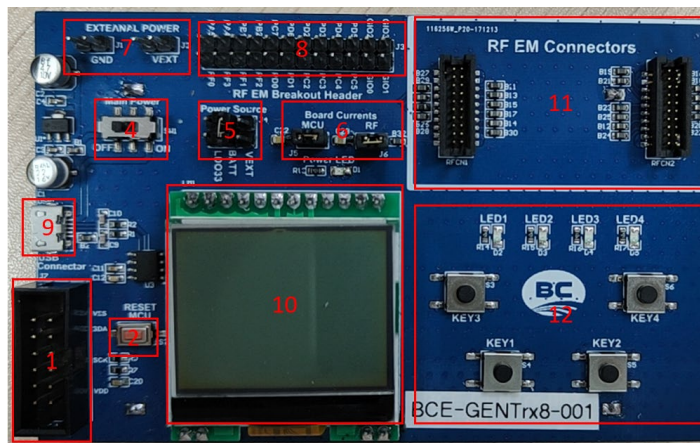


圖 4. 開發板(BCE-GENTrx8-001)

1. JTAG 介面，可配合 IDE 介面模擬程式及下載程式用。
2. 系統重置鍵。
3. UART/USB 介面選擇 (只有 BCE-GENTrx32-002 開發板有此項)。
4. 總電源開關，左撥(OFF)為關閉電源，右撥(ON)為開啟電源。
5. 系統電源選擇，跳線(Jumper)於左側 LDO33 處，表示電源由 USB 埠輸入；跳線於中間 BATT 處，代表電源使用電池座(板背：兩顆 1.5V AA 電池)；跳線於右側 VEXT 處，則是使用外部電源接點供電，注意若是使用外部電源接點輸入電壓，電壓不得超 3.6V。
6. MCU 和 BM7701-00-1 模組板電源電流檢測點。
7. 外部電源接點。
8. MCU 的部份 I/O 埠。
9. Micro USB 介面，可用來作為系統電源輸入(電源選擇 LDO33)。
10. 液晶顯示器(支援 128x64 點)，使用方法請參考本文範例程式，內含顯示器控制程式庫。
11. 射頻模組介面是射頻發射/接收裝置連接處，此範例則安裝上 BCT-7701-001+BM7701-00-1。
12. LEDx4 及按鍵x4，此範例用於顯示數據接收狀態和發送數據。

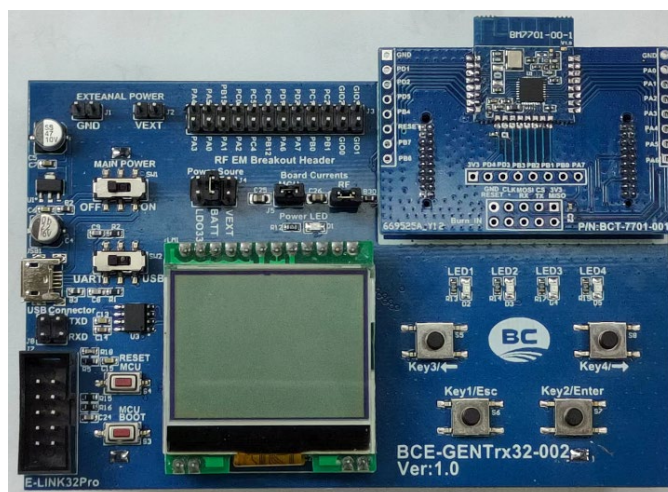


圖 5. 組裝完成圖(HT32)

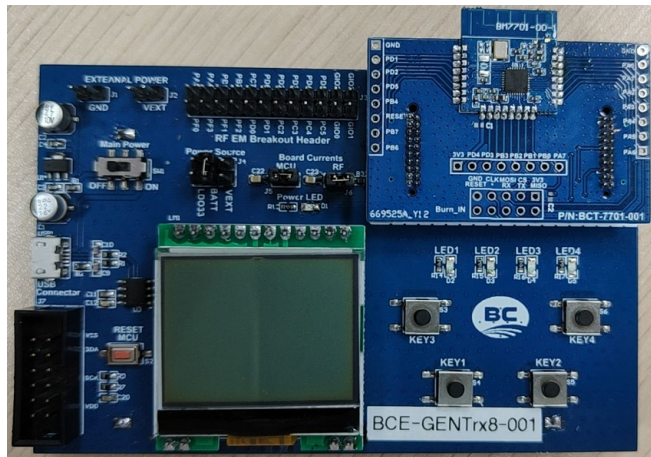


圖 6. 組裝完成圖(HT8)

軟體：手機端需要下載 App BLEDemo，相關的鏈接如下圖所示。



圖 7. Android(左) & iOS(右)

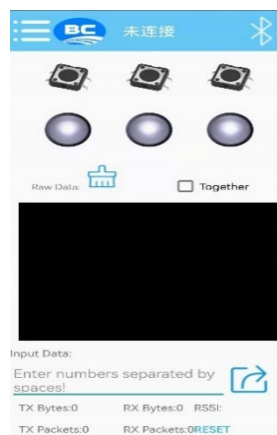


圖 8. 安裝後的 UI

操作說明

1. 使用 elink-32 Pro 燒錄應用範例程式到開發板 BCE-GENTrx32-002 上的 HT32F52352 (HT32)。
使用 elink 燒錄應用範例程式到開發板 BCE-GENTrx8-001 上的 HT66F2370 (HT8)。
2. 打開開發板開關，成功上電後 Power LED 常亮，LCD 屏有顯示。
3. LCD 等待 1s 的開機介面後會顯示狀態介面，然後等待顯示 PWR ON OK，即藍牙設定配置成功且開啟廣播。
4. 點擊手機 App 右上角進入掃描藍牙設備，選擇名稱“BM7701”連接。連接成功後，App 右上角連接 icon 會變成紅色。

5. 開發板按鍵操作。
 - 按鍵 1 按下：App 上面對應 LED1 icon 亮
 - 按鍵 2 按下：App 上面對應 LED2 icon 亮
 - 按鍵 3 按下：App 上面對應 LED3 icon 亮
6. App 操作。
 - 按鍵 1 icon 按下：LED1 亮
 - 按鍵 2 icon 按下：LED2 亮
 - 按鍵 3 icon 按下：LED3 亮

硬體說明

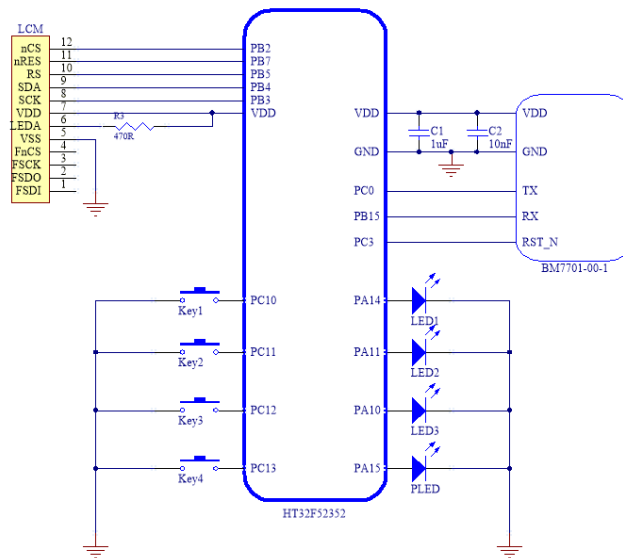


圖 9. 應用電路圖(HT32)

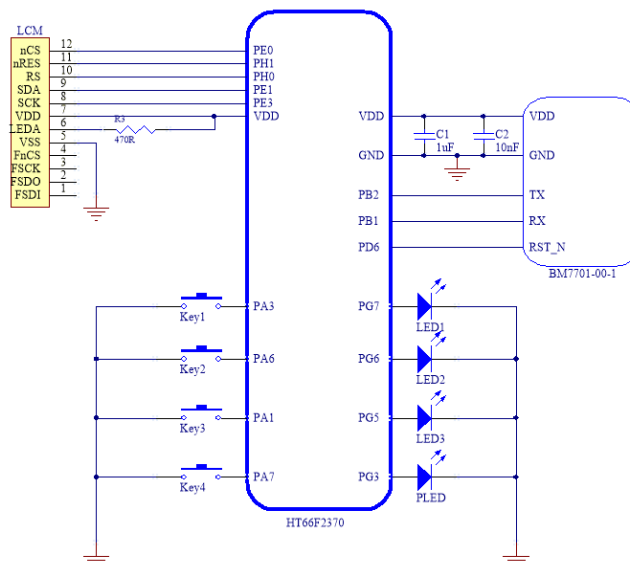


圖 10. 應用電路圖(HT8)

軟體說明

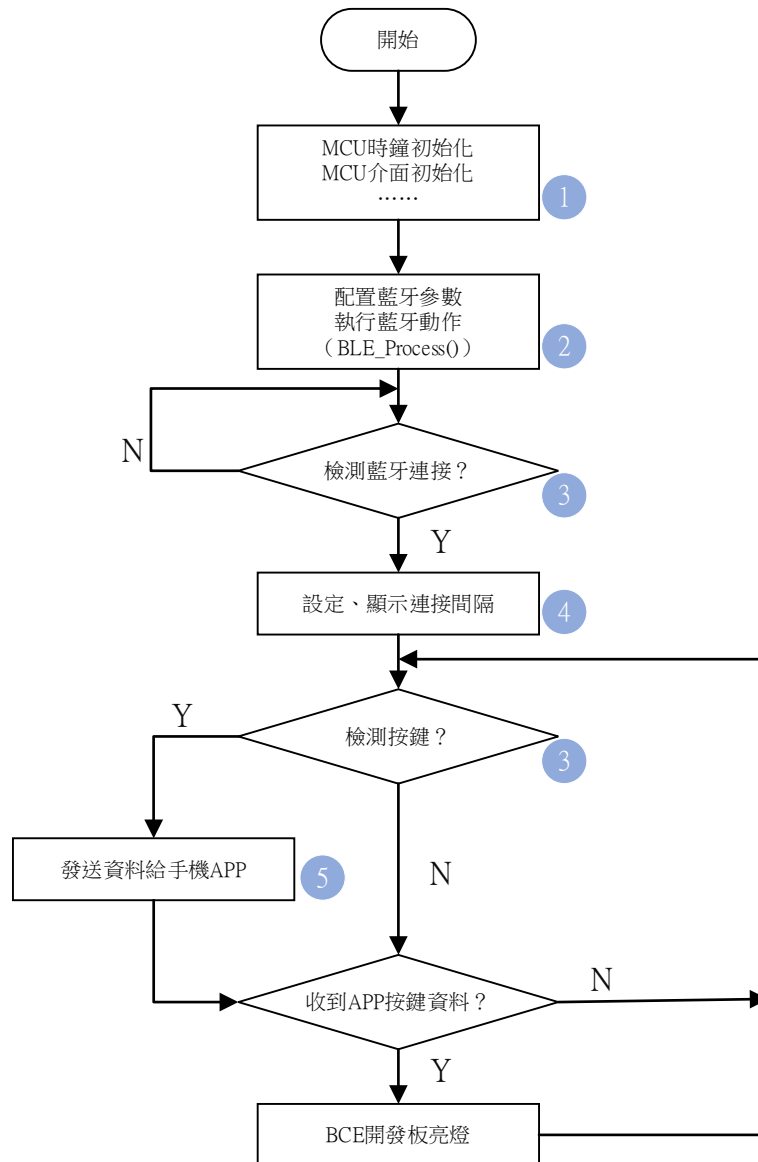


圖 11. 軟體流程圖

- 先進行 MCU 時鐘初始化，LED、LCD、按鍵等外圍器件初始化。
- MCU 發送 Command 給 BM7701-00-1 來配置藍牙參數、開啟廣播。
- 手機 App 與 BM7701-00-1 藍牙連接成功。
- MCU 檢測到開發板按鍵會發送 Command 給 BM7701-00-1，BM7701-00-1 再發送資料給手機 App。
- 手機 App 收到指示也會發送資料給 BM7701-00-1。MCU 解析 BM7701-00-1 收到的資料 Event 並點亮開發板對應的 LED。

1、初始化更新函數

- BC7701_InterfaceConfigure(_UART_BAUDRATE_)：設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態
- BC7701_RESET_CLR()：BM7701-00-1 的 RST_N 腳位拉低
- BC7701_RESET_SET()：BM7701-00-1 的 RST_N 腳位拉高
- BC7701_HardwareBaudRateDefault(_UART_BAUDRATE_)：通過通訊腳位初始化 BM7701-00-1 串列傳輸速率
- BC7701_HardwareBaudRateRelease()：完成 BM7701-00-1 串列傳輸速率初始化設定

2、藍牙參數設定

- BC7701_SendBCIReadPackage(BCI_VERSION, 0x80)：檢查版本(HT32)
BC7701_TransmitCmdPackage(BCI_VERSION, 0x80)：檢查版本(HT8)
- BC7701_SendBCIReadPackage(BCI_DEV_ADDRESS, 0x00)：獲取地址(HT32)
BC7701_TransmitCmdPackage(BCI_DEV_ADDRESS, 0x00)：獲取地址(HT8)
- BC7701_SetBaudRate(), BC7701_SetTxPower()等等：設定 BM7701-00-1 的波特率、傳出功率等等，需要使能相應的選項(HT32)
BC7701_SetBaudRate(), BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BLE_SetTxPower)等等：設定 BM7701-00-1 的波特率、傳出功率等等，需要使能相應的選項(HT8)
- BC7701_SetupFeatureFlag(FEATURE_SET, FEATURE_STATUS_EVENT)：設定 BM7701-00-1 在狀態改變時發送數據(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BLE_SetFeatureState)：設定 BM7701-00-1 在狀態改變時發送數據(HT8)
- BC7701_AdvertisingControl(ENABLE)：設定 BM7701-00-1 開啟廣播(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BC7701_AdvertiseEnable)：設定 BM7701-00-1 開啟廣播(HT8)

3、進入深度休眠

- BC7701_SetOperateMode(OP_DEEPSLEEP,ENABLE,_MASTER_WUW_VALUE_,_MASTER_WUT_VALUE_)：設定 BM7701-00-1 進入深度休眠模式和 MCU 外部喚醒訊號(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BC7701_OperateSleep)：設定 BM7701-00-1 進入深度休眠模式和 MCU 外部喚醒訊號(HT8)

4、設定連接間隔

- BC7701_DummyWakeup():將 BM7701-00-1 從休眠模式中喚醒
- BC7701_ConnectIntervalModify(u16 opcode, u16 min, u16 max)：設定 BM7701-00-1 的連接間隔(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BLE_CmdIndex):設定 BM7701-00-1 的連接間隔(HT8)

5、資料發送

- BC7701_DummyWakeup() : 將 BM7701-00-1 從休眠模式中喚醒
- BC7701_SendBCIPackage(NotifyFFF1, BCI_ServiceProperties, len, pbuf) : BM7701-00-1 發送無線資料給手機 App

模組 API 命令函數介紹

1. 以下表格整合 BM7701-00-1 模組 API 命令函數與其功能，請參考應用範例程式的 BC7701.c 檔案(HT32)。

API 原型	功能說明
void BC7701_InterfaceConfigure(u8 br)	設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態
void BC7701_UARTConfigure(u8 br)	設定 MCU 的通訊腳位為 UART 模式和串列傳輸速率
void BC7701_RESET_SET(void)	BM7701-00-1 的 RST_N 腳位拉高
void BC7701_RESET_CLR(void)	BM7701-00-1 的 RST_N 腳位拉低
void BC7701_HardwareBaudRateDefault(u8 br)	通過通訊腳位初始化 BM7701-00-1 串列傳輸速率
void BC7701_HardwareBaudRateRelease(void)	完成 BM7701-00-1 串列傳輸速率初始化設定
void BC7701_HardwareReset(void)	硬體重置，初始化藍牙參數
void *BC7701_SoftwareReset(void)	設定“軟體重置”命令，初始化藍牙參數，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SoftwareResetKeep(void)	設定“軟體重置”命令，保留 RAM 的藍牙參數，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)	設定 HCI 命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)	設定 BCI 命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetDeviceName(u8 leng,u8 *name)	設定“BM7701-00-1 的藍牙設備名稱”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetDeviceAddress(u8 *adr,u8 type)	設定“BM7701-00-1 的藍牙地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)	設定“BM7701-00-1 的連接間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)	設定“BM7701-00-1 的廣播數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetScanResponseData(u8 leng,u8 *sdata)	設定“BM7701-00-1 的掃描響應數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)	設定“BM7701-00-1 的廣播間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_AdvertisingControl(ControlStatus ctrl)	設定“BM7701-00-1 的廣播開關”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetTxPower(u8 pwr)	設定“BM7701-00-1 的輸出功率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetCrystalCload(u8 cc)	設定“BM7701-00-1 的外部 16MHz 晶振 C _{load} 值”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)	設定“BM7701-00-1 的 Feature”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetOperateMode(u8 omd,ControlStatus ctrl,u8 wuw,u8 wut)	設定“BM7701-00-1 的休眠模式和 MCU 外部喚醒訊號”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetWhiteList(ControlStatus erase,u8 *adr,u8 *mask)	設定“BM7701-00-1 的白名單地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。
void *BC7701_SetBaudRate(u8 br)	設定“BM7701-00-1 的串列傳輸速率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送。

API 原型	功能說明
bool BC7701_DummyWakeup(void)	將 BM7701-00-1 從休眠模式中喚醒
bool BC7701_TransmitPackage(void *pbuf)	將命令封包放置於發送隊列
bool BC7701_ReadTransmitEmpty(void)	獲取資料發送隊列的狀態
bool BC7701_ReadReceiveEmpty(void)	獲取資料接收隊列的狀態
void *BC7701_ReadReceivePackage(void)	讀取 BM7701-00-1 的接收資料
void BC7701_WriteReceivePackage(void)	將接收資料索引寫入接收隊列
void BC7701_ReceiveParserPackage(void)	解析 BM7701-00-1 接收的資料

表 6. 模組 API 命令函數(HT32)

API 名稱	void BC7701_InterfaceConfigure(u8 br)
API 作用	設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態
輸入參數	波特率：BAUD_RATE_9600 / BAUD_RATE_14400 / BAUD_RATE_19200 / BAUD_RATE_38400 / BAUD_RATE_57600 / BAUD_RATE_115200
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態。設定 MCU 的 Reset control pin 為 AFIO 模式。此函數會呼叫 BC7701_UARTConfigure(u8 br)，可以設定 MCU 的通訊腳位為 UART 模式和串列傳輸速率。

API 名稱	void BC7701_UARTConfigure(u8 br)
API 作用	設定 MCU 的通訊腳位為 UART 模式和串列傳輸速率
輸入參數	波特率：BAUD_RATE_9600 / BAUD_RATE_14400 / BAUD_RATE_19200 / BAUD_RATE_38400 / BAUD_RATE_57600 / BAUD_RATE_115200
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定 MCU 的通訊腳位為 UART 模式和串列傳輸速率

API 名稱	void BC7701_RESET_SET(void)
API 作用	BM7701-00-1 的 RST_N 腳位拉高
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，BM7701-00-1 的 RST_N 腳位被拉高

API 名稱	void BC7701_RESET_CLR(void)
API 作用	BM7701-00-1 的 RST_N 腳位拉低
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，BM7701-00-1 的 RST_N 腳位被拉低

API 名稱	void BC7701_HardwareBaudRateDefault(u8 br)
API 作用	通過通訊腳位狀態初始化 BM7701-00-1 串列傳輸速率(請參考表 2)
輸入參數	串列傳輸速率：BAUD_RATE_9600/ BAUD_RATE_115200
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，依通訊介面定義設定通訊腳位狀態初始化 BM7701-00-1 串列傳輸速率。此 API 函數需在硬體 Reset 後馬上執行才有效。此 API 函數將通訊腳位由 UART 模式改成 I/O 模式。執行完畢後必須執行 API 函數 BC7701_HardwareBaudRateRelease()，將通訊腳位還原成 UART 模式。

API 名稱	void BC7701_HardwareBaudRateRelease(void)
API 作用	完成 BM7701-00-1 串列傳輸速率初始化設定
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，完成 BM7701-00-1 的串列傳輸速率初始化設定。執行該函數的前提是 BC7701_HardwareBaudRateDefault(u8 br)有在執行，然後等待 BM7701-00-1 上電重置 60ms 後再執行該函數

API 名稱	void BC7701_HardwareReset(void)
API 作用	硬體重置，初始化藍牙參數(拉低 BM7701-00-1 的 RST_N 腳位來實現)
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，BM7701-00-1 硬體重置，保存在 RAM 的藍牙參數會被清掉。執行此函數後 BM7701-00-1 至少需要等待 60ms 才能再下 Command。該函數等效於*BC7701_SoftwareReset(void)

API 名稱	void *BC7701_SoftwareReset(void)
API 作用	設定“軟體重置”命令，初始化藍牙參數，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	—
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“軟體重置”命令。命令執行成功後 BM7701-00-1 軟體重置，保存在 RAM 的藍牙參數會被清掉。等待 BM7701-00-1 回完 Event 後需要等待 60ms 才能再下 Command。 該命令等效於 BC7701_HardwareReset(void)。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SoftwareResetKeep(void)
API 作用	設定“軟體重置”命令，保留 RAM 的藍牙參數，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	—
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“軟體重置”命令(not Initial Parameter)。命令執行成功後 BM7701-00-1 軟體重置，RAM 的藍牙參數仍然保留，等待 BM7701-00-1 回完 Event 後需要等待 3ms 才能再下 Command。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)
API 作用	設定 HCI 命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	HCI 操作碼(請參考“BLE_API”文檔)
輸入參數 2	數據長度
輸入參數 3	存放數據的 Buffer
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定 HCI 命令。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	<code>void *BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)</code>
API 作用	設定 BCI (BLE Controller Interface)命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	BCI 操作碼(請參考“BLE_API”文檔)
輸入參數 2	BCI 旗標(請參考“BLE_API”文檔)
輸入參數 3	數據長度
輸入參數 4	存放數據的 Buffer
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定 BCI (BLE Controller Interface)命令。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。
API 名稱	<code>void *BC7701_SetDeviceName(u8 leng,u8 *name)</code>
API 作用	設定“BM7701-00-1 的藍牙設備名稱”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	數據長度(≤31 byte)
輸入參數 2	存放名稱數據的 Buffer
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的藍牙設備名稱”命令。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	<code>void *BC7701_SetDeviceAddress(u8 *adr,u8 type)</code>
API 作用	設定“BM7701-00-1 的藍牙地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	存放藍牙地址的 Buffer(6 byte)
輸入參數 2	藍牙地址形態：STATIC_ADDRESS/RANDOM_ADDRESS
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的藍牙地址”命令。 輸入參數 2： 1. 若輸入參數 2 等於 STATIC_ADDRESS，藍牙地址形態設定為靜態地址。 2. 若輸入參數 2 等於 RANDOM_ADDRESS，藍牙地址形態設定為隨機地址。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	<code>void *BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)</code>
API 作用	設定“BM7701-00-1 的連接間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	操作碼：BCI_CONN_INTV / BCI_CONN_INTV1 (請參考“BLE_API”文檔)
輸入參數 2	最小連接間隔，單位=1.25，有效範圍=7.5ms~4s
輸入參數 3	最大連接間隔，單位=1.25，有效範圍=7.5ms~4s，僅當操作碼= BCI_CONN_INTV1 有效
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的連接間隔”命令。 輸入參數 1： 1. 若輸入參數 1 等於 BCI_CONN_INTV，僅輸入參數 2 有效，設定唯一的連接間隔。 2. 若輸入參數 1 等於 BCI_CONN_INTV1，輸入參數 2 和輸入參數 3 都有效。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)
API 作用	設定“BM7701-00-1 的廣播數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	廣播模式：ADV_JOIN_NAME / ADV_UNJOIN_NAME (請參考“BLE_API”文檔)
輸入參數 2	廣播數據長度(≤31 byte)
輸入參數 3	存放廣播數據的 Buffer
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的廣播數據”命令 輸入參數 1： 1. 若輸入參數 1 等於 ADV_JOIN_NAME，廣播數據會自動添加藍牙設備名稱，前提是廣播數據≤31 byte。 2. 若輸入參數 1 等於 ADV_UNJOIN_NAME，廣播數據不會加入藍牙設備名稱。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SetScanResponseData(u8 leng,u8 *sdata)
API 作用	設定“BM7701-00-1 的掃描響應數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	掃描響應數據長度(≤31 byte)
輸入參數 2	存放掃描響應數據的 Buffer
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的掃描響應數據”命令。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)
API 作用	設定“BM7701-00-1 的廣播間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	最小廣播間隔，單位=0.625，範圍=20ms~10.24s
輸入參數 2	最大廣播間隔，單位=0.625，範圍=20ms~10.24s
輸入參數 3	廣播通道
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的廣播間隔”命令。 輸入參數 3： 1. 若輸入參數 3 等於 0B xxx1 (B[0]=1)，設定廣播通道 37(2402MHz)。 2. 若輸入參數 3 等於 0B xx1x (B[1]=1)，設定廣播通道 38(2426MHz)。 3. 若輸入參數 3 等於 0B x1xx (B[2]=1)，設定廣播通道 39(2480MHz)。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_AdvertisingControl(ControlStatus ctrl)
API 作用	設定“BM7701-00-1 的廣播開關”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	開關：ENABLE/DISABLE
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的廣播開關”命令。 輸入參數： 1. 若輸入參數等於 ENABLE，打開藍牙廣播。 2. 若輸入參數等於 DISABLE，關閉藍牙廣播。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SetTxPower(u8 pwr)			
API 作用	設定“BM7701-00-1 的輸出功率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送			
輸入參數	0~15			
輸出參數	命令封包指針			
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的輸出功率”命令。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。			
輸入參數	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

註：輸出功率值僅做參考，具體數值需以實際取得模組數據為準。

API 名稱	void *BC7701_SetCrystalCload(u8 cc)			
API 作用	設定“BM7701-00-1 的外部 16MHz 晶振 C _{LOAD} 值”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送			
輸入參數	0~15			
輸出參數	命令封包指針			
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的外部 16MHz 晶振 C _{LOAD} 值”命令。BM7701-00-1 建議採用 04 作為輸入參數。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。			
輸入參數	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

註：晶振值僅做參考，具體數值需以實際取得模組數據為準。

API 名稱	void *BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)
API 作用	設定“BM7701-00-1 的 Feature”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	數據更新方式：FEATURE_DIR / FEATURE_SET / FEATURE_CLR
輸入參數 2	模式：FEATURE_NO_APPED_NAME / FEATURE_PARAM_UPDATE / FEATURE_PARAM_ERASE / FEATURE_STATUS_EVENT / FEATURE_EXTERNAL32K / FEATURE_FORCE_CALIB / FEATURE_CK32K_OUTPUT (請參考“BLE_API”文檔)
輸出參數	命令封包指針
程式詳解	<p>範例程式中，執行此 API 函數，設定“BM7701-00-1 的 Feature”命令(詳情參考“BLE_API”文檔)。</p> <p>輸入參數 1：</p> <ol style="list-style-type: none"> 若輸入參數 1 等於 FEATURE_DIR，輸入參數 2 會直接覆蓋之前的數據。 若輸入參數 1 等於 FEATURE_SET，輸入參數 2 會和之前的數據進行“OR”操作。 若輸入參數 1 等於 FEATURE_CLR，輸入參數 2 會和之前的數據進行“AND”操作。 <p>輸入參數 2：</p> <ol style="list-style-type: none"> 若輸入參數 2 等於 FEATURE_NO_APPED_NAME，廣播數據不會加入藍牙設備名稱。 若輸入參數 2 等於 FEATURE_PARAM_UPDATE，BM7701-00-1 的 RAM 藍牙參數寫到 Flash。 若輸入參數 2 等於 FEATURE_PARAM_ERASE，刪除 BM7701-00-1 的 Flash 的藍牙參數，恢復默認值。 若輸入參數 2 等於 FEATURE_STATUS_EVENT，狀態有變更時，BM7701-00-1 自動發送 Status Event。 若輸入參數 2 等於 FEATURE_EXTERNAL32K，啟用外部 32.768kHz 晶振。 若輸入參數 2 等於 FEATURE_FORCE_CALIB，下次 Software reset 時強制校準。 若輸入參數 2 等於 FEATURE_CK32K_OUTPUT，BM7701-00-1 的 UART2_TX(PB6) 輸出 32kHz 方波。 <p>輸出參數：</p> <ol style="list-style-type: none"> 若輸出參數等於 NULL，API 函數配置失敗。 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SetOperateMode(u8 omd,ControlStatus ctrl,u8 wuw,u8 wut)
API 作用	設定“BM7701-00-1 的休眠模式和 MCU 外部喚醒訊號”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	模式：OP_NORMAL / OP_DEEPSLEEP / OP_POWERDOWN
輸入參數 2	MCU 休眠：ENABLE / DISABLE
輸入參數 3	MCU 外部喚醒訊號寬度(0~8 byte)
輸入參數 4	MCU 外部喚醒訊號延遲時間(0~20ms)
輸出參數	命令封包指針
程式詳解	<p>範例程式中，執行此 API 函數，設定“BM7701-00-1 的休眠模式和 MCU 外部喚醒訊號”命令。(詳情參考“BLE_API”文檔)。</p> <p>輸入參數 1：</p> <ol style="list-style-type: none"> 若輸入參數 1 等於 OP_NORMAL，BM7701-00-1 進入 Normal 模式。默認情況下 BM7701-00-1 在 Normal 模式。 若輸入參數 1 等於 OP_DEEPSLEEP，BM7701-00-1 進入 Deep Sleep 模式。可使用 API 函數 BC7701_DummyWakeup(void)將 BM7701-00-1 喚醒到 Normal 模式。 若輸入參數 1 等於 OP_POWERDOWN，BM7701-00-1 進入 Power Down 模式。可使用 API 函數 BC7701_DummyWakeup(void)將 BM7701-00-1 喚醒到 Normal 模式。 <p>輸入參數 2：</p> <ol style="list-style-type: none"> 若輸入參數 2 等於 ENABLE，說明 MCU 會進入休眠，設定 BM7701-00-1 發出喚醒訊號來喚醒 MCU。 若輸入參數 2 等於 DISABLE，說明 MCU 不會進入休眠，無需設定 BM7701-00-1 發出喚醒訊號。 <p>輸出參數：</p> <ol style="list-style-type: none"> 若輸出參數等於 NULL，API 函數配置失敗。 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SetWhiteList(ControlStatus erase,u8 *adr,u8 *mask)
API 作用	設定“BM7701-00-1 的白名單地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	清除之前的白名單地址：ENABLE/DISABLE
輸入參數 2	存放白名單地址的 buffer (6 byte)
輸入參數 3	存放地址掩碼的 buffer (6 byte)
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的白名單地址”命令。 Ex：白名單地址=0x112233445566，地址掩碼=0xFFFFFFFF00，只有藍牙地址 0x112233445500~0x1122334455FF 可以連接。如果地址掩碼全都寫 0，所有藍牙地址都可以連接。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void *BC7701_SetBaudRate(u8 br)
API 作用	設定“BM7701-00-1 的串列傳輸速率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	波特率：BAUD_RATE_9600 / BAUD_RATE_14400 / BAUD_RATE_19200 / BAUD_RATE_38400 / BAUD_RATE_57600 / BAUD_RATE_115200
輸出參數	命令封包指針
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的串列傳輸速率”命令。執行此命令且等待 BM7701-00-1 回完 EVENT 後，MCU 需等待約 1ms 後才能改變 UART 串列傳輸速率及執行下一個通訊傳輸。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	bool BC7701_DummyWakeup(void)
API 作用	將 BM7701-00-1 從休眠模式中喚醒
輸入參數	—
輸出參數	TRUE：發送 Dummy wakeup 成功 FALSE：發送 Dummy wakeup 失敗
程式詳解	範例程式中，執行此 API 函數，將 BM7701-00-1 從休眠模式中喚醒

API 名稱	bool BC7701_TransmitPackage(void *pbuf)
API 作用	將 Command 封包放置於發送隊列(封包格式請參考“Command/Event 格式”)
輸入參數	存放封包數據的 Buffer
輸出參數	TRUE：放置發送隊列成功 FALSE：表示發送隊列已滿
程式詳解	範例程式中，執行此 API 函數，MCU 將封包放置於發送隊列中，發送隊列中的封包會依序傳輸給 BM7701-00-1。

API 名稱	bool BC7701_ReadTransmitEmpty(void)
API 作用	獲取資料發送隊列的狀態
輸入參數	—
輸出參數	TRUE：隊列是空的 FALSE：隊列非空的
程式詳解	範例程式中，執行此 API 函數，獲取資料發送隊列的狀態

API 名稱	bool BC7701_ReadReceiveEmpty(void)
API 作用	獲取資料接收隊列的狀態
輸入參數	—
輸出參數	TRUE：隊列是空的 FALSE：隊列非空
程式詳解	範例程式中，執行此 API 函數，獲取資料接收隊列的狀態

API 名稱	void *BC7701_ReadReceivePackage(void)
API 作用	讀取 BM7701-00-1 的接收資料
輸入參數	—
輸出參數	封包指針
程式詳解	範例程式中，執行此 API 函數，獲取放置接收隊列的資料，若輸出為 NULL 指針表示沒有資料。

API 名稱	void BC7701_WriteReceivePackage(void)
API 作用	將接收資料索引寫入接收隊列
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，將接收資料索引寫入接收隊列

API 名稱	void BC7701_ReceiveParserPackage(void)
API 作用	解析接收到的 BM7701-00-1 資料
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，解析接收到的 BM7701-00-1 資料，並將串列資料轉成封包資料，並放置於接收隊列中。

2. 以下表格整合 BM7701-00-1 模組 API 命令函數以及部分重要宏定義，請參考應用範例程式的 BC7701.c 和 BC7701.h 檔案 (HT8)。

API 原型	功能說明
void BC7701_InterfaceConfigure(u8 br)	設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態
void BC7701_UARTConfigure(u8 br)	設定 MCU 的通訊腳位狀態為 UART 模式和串列傳輸速率
void BC7701_UARTWakeUpCtrl(u8 ctrl)	設定 MCU 的 UART 喚醒功能
void BC7701_HardwareBaudRateDefault(u8 br)	通過通訊腳位初始化 BM7701-00-1 串列傳輸速率
void BC7701_HardwareBaudRateRelease(void)	完成 BM7701-00-1 串列傳輸速率初始化設定
void BC7701_SoftwareReset(void)	設定“BM7701-00-1 軟體重置”命令，初始化藍牙參數，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)	設定 HCI 命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)	設定 BCI 命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetDeviceName(u8 leng,u8 *name)	設定“BM7701-00-1 的藍牙設備名稱”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetDeviceAddress(u8 *adr,u8 type)	設定“BM7701-00-1 的藍牙地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)	設定“BM7701-00-1 的連接間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)	設定“BM7701-00-1 的廣播數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetScanResponseData(u8 leng,u8 *sdata)	設定“BM7701-00-1 的掃描響應數據”命令，需通過呼

API 原型	功能說明
	叫 BC7701_TransmitPackage 函數進行發送
void BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)	設定“BM7701-00-1 的廣播間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_AdvertisingControl(u8 ctrl)	設定“BM7701-00-1 的廣播開關”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetTxPower(u8 pwr)	設定“BM7701-00-1 的輸出功率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetCrystalCload(u8 cc)	設定“BM7701-00-1 的外部 16MHz 晶振 C _{LOAD} 值”命令
void BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)	設定“BM7701-00-1 的 Feature”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetOperateMode(u8 omd,u8 wue,u8 wuw,u8 wut)	設定“BM7701-00-1 的休眠模式和 MCU 外部喚醒訊號”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetWhiteList(u8 erase,u8 *adr,u8 *mask)	設定“BM7701-00-1 的白名單地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_SetBaudRate(u8 br)	設定“BM7701-00-1 的串列傳輸速率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
void BC7701_DummyWakeup(u8 leng)	將 BM7701-00-1 從休眠模式中喚醒
void BC7701_TransmitCmdPackage(u16 opcode,u8 flag)	給 BM7701-00-1 發送 BCI“READ”命令
void BC7701_TransmitPackage(u8 length)	給 BM7701-00-1 發送 TransmitData[] 封包
Void BC7701_TransmitPackageConst(tBCI_PACKAGE *pkg)	給 BM7701-00-1 發送 BCI 封包數據
u8 BC7701_PackageParserProcess(void)	解析 BM7701-00-1 接收的資料
#define BC7701_RESET_SET()	BM7701-00-1 的 RST_N 腳位拉高
#define BC7701_RESET_CLR()	BM7701-00-1 的 RST_N 腳位拉低

表 7. 模組 API 命令函數(HT8)

API 名稱	void BC7701_InterfaceConfigure(u8 br)
API 作用	設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態
輸入參數	波特率：BAUD_RATE_4800 / BAUD_RATE_9600 / BAUD_RATE_19200 / BAUD_RATE_25000 / BAUD_RATE_50000 / BAUD_RATE_62500 / BAUD_RATE_115200
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定與 BM7701-00-1 通訊的 MCU I/O 埠狀態。設定 MCU 的 Reset control pin 為 IO 模式，通訊腳位為 UART 模式。此函數有呼叫 BC7701_UARTConfigure(u8 br)，設定 MCU 的 UART 參數。

API 名稱	void BC7701_UARTConfigure(u8 br)
API 作用	設定 MCU 的 UART 參數
輸入參數	波特率：BAUD_RATE_4800 / BAUD_RATE_9600 / BAUD_RATE_19200 / BAUD_RATE_25000 / BAUD_RATE_50000 / BAUD_RATE_62500 / BAUD_RATE_115200
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定 MCU 的 UART 參數，包括 MCU 的串列傳輸速率

API 名稱	void BC7701_UARTWakeupCtrl(u8 ctrl)
API 作用	設定 MCU 的 UART 喚醒功能
輸入參數	ENABLE / DISABLE
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定 MCU 的 UART 喚醒功能。 輸入參數 1： 1. 若輸入參數 1 等於 ENABLE，致能 MCU 的 UART 喚醒功能。 2. 若輸入參數 1 等於 DISABLE，除能 MCU 的 UART 喚醒功能。

API 名稱	void BC7701_HardwareBaudRateDefault(u8 br)
API 作用	通過通訊腳位狀態初始化 BM7701-00-1 串列傳輸速率(請參考表 2)
輸入參數	串列傳輸速率：BAUD_RATE_9600/ BAUD_RATE_115200
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，依通訊介面定義設定通訊腳位狀態初始化。BM7701-00-1 串列傳輸速率。 此 API 函數需在硬體 Reset 後馬上執行才有效。 此 API 函數將通訊腳位由 UART 模式改成 I/O 模式。執行完畢後必須執行 API 函數 BC7701_HardwareBaudRateRelease()，將通訊腳位還原成 UART 模式。

API 名稱	void BC7701_HardwareBaudRateRelease(void)
API 作用	完成 BM7701-00-1 串列傳輸速率初始化設定
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，完成 BM7701-00-1 的串列傳輸速率初始化設定。執行該函數的前提是 BC7701_HardwareBaudRateDefault(u8 br)有在執行，然後等待 BM7701-00-1 上電重置 60ms 後再執行該函數

API 名稱	void BC7701_SoftwareReset(void)
API 作用	設定“BM7701-00-1 軟體重置”命令，初始化藍牙參數，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	—
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，發送“BM7701-00-1 軟體重置”命令，保存在 RAM 的藍牙參數會被清掉。等待 BM7701-00-1 回完 Event 後需要等待 60ms 才能再下 Command

API 名稱	void BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)
API 作用	設定 HCI 命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	HCI 操作碼(請參考“BLE_API”文檔)
輸入參數 2	數據長度
輸入參數 3	存放數據的 Buffer
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定 HCI 命令並保存到 TransmitData[]

API 名稱	void BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)
API 作用	設定 BCI (BLE Controller Interface)命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	BCI 操作碼(請參考“BLE_API”文檔)
輸入參數 2	BCI 旗標(請參考“BLE_API”文檔)
輸入參數 3	數據長度
輸入參數 4	存放數據的 Buffer
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定 BCI 命令並保存到 TransmitData[]

API 名稱	void BC7701_SetDeviceName(u8 leng,u8 *name)
API 作用	設定“BM7701-00-1 的藍牙設備名稱”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	數據長度(≤31 byte)
輸入參數 2	存放名稱數據的 Buffer
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的藍牙設備名稱”命令，命令數據保存到 TransmitData[]

API 名稱	void BC7701_SetDeviceAddress(u8 *adr,u8 type)
API 作用	設定“BM7701-00-1 的藍牙地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	存放藍牙地址的 Buffer(6 byte)
輸入參數 2	藍牙地址形態：STATIC_ADDRESS/RANDOM_ADDRESS
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的藍牙地址”命令，命令數據保存到 TransmitData[]。 輸入參數 2： 1. 若輸入參數 2 等於 STATIC_ADDRESS，藍牙地址形態設定為靜態地址。 2. 若輸入參數 2 等於 RANDOM_ADDRESS，藍牙地址形態設定為隨機地址。

API 名稱	void BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)
API 作用	設定“BM7701-00-1 的連接間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	操作碼：BCI_CONN_INTV / BCI_CONN_INTV1 (請參考“BLE_API”文檔)
輸入參數 2	最小連接間隔，單位=1.25，有效範圍=7.5ms~4s
輸入參數 3	最大連接間隔，單位=1.25，有效範圍=7.5ms~4s，僅當操作碼= BCI_CONN_INTV1 有效
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的連接間隔”命令，命令數據保存到 TransmitData[]。 輸入參數 1： 1.若輸入參數 1 等於 BCI_CONN_INTV，僅輸入參數 2 有效，設定唯一的連接間隔。 2.若輸入參數 1 等於 BCI_CONN_INTV1，輸入參數 2 和輸入參數 3 都有效。

API 名稱	void BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)
API 作用	設定“BM7701-00-1 的廣播數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	廣播模式：UPDAE_AUTO_NAME / DEFAULT_NAME / DEFAULT_EMPTY / UPDAE_NOAO_NAME (請參考“BLE_API”文檔)
輸入參數 2	廣播數據長度(≤31 byte)
輸入參數 3	存放廣播數據的 Buffer
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的廣播數據”命令，命令數據保存到 TransmitData[]。 輸入參數 1： 1. 若輸入參數 1 等於 DEFAULT_NAME，廣播數據採用默認值。 2. 若輸入參數 1 等於 UPDAE_AUTO_NAME，廣播數據會自動添加藍牙設備名稱，前提是廣播數據≤31 byte。 3. 若輸入參數 1 等於 DEFAULT_EMPTY，廣播數據為空。 4. 若輸入參數 1 等於 UPDAE_NOAO_NAME，廣播數據不會加入藍牙設備名稱。

API 名稱	void BC7701_SetScanResponseData(u8 leng,u8 *sdata)
API 作用	設定“BM7701-00-1 的掃描響應數據”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	掃描響應數據長度(≤31 byte)
輸入參數 2	存放掃描響應數據的 Buffer
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的掃描響應數據”命令，命令數據保存到 TransmitData[]。 輸出參數： 1. 若輸出參數等於 NULL，API 函數配置失敗。 2. 若輸出參數不等於 NULL，API 函數配置成功。

API 名稱	void BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)
API 作用	設定“BM7701-00-1 的廣播間隔”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	最小廣播間隔，單位=0.625，範圍=20ms~10.24s
輸入參數 2	最大廣播間隔，單位=0.625，範圍=20ms~10.24s
輸入參數 3	廣播通道
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的廣播間隔”命令，命令數據保存到 TransmitData[]。 輸入參數 3： 1. 若輸入參數 3 等於 0B xxx1 (B[0]=1)，設定廣播通道 37(2402MHz)。 2. 若輸入參數 3 等於 0B xx1x (B[1]=1)，設定廣播通道 38(2426MHz)。 3. 若輸入參數 3 等於 0B x1xx (B[2]=1)，設定廣播通道 39(2480MHz)。

API 名稱	void BC7701_AdvertisingControl(ControlStatus ctrl)
API 作用	設定“BM7701-00-1 的廣播開關”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	開關：ENABLE/DISABLE
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的廣播開關”命令，命令數據保存到 TransmitData[]。 輸入參數： 1. 若輸入參數等於 ENABLE，打開藍牙廣播。 2. 若輸入參數等於 DISABLE，關閉藍牙廣播。

API 名稱	void BC7701_SetTxPower(u8 pwr)			
API 作用	設定“BM7701-00-1 的輸出功率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送			
輸入參數	0~15			
輸出參數	—			
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的輸出功率”命令，命令數據保存到 TransmitData[]。			
輸入參數	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

註：輸出功率值僅做參考，具體數值需以實際取得模組數據為準。

API 名稱	void BC7701_SetCrystalCload(u8 cc)			
API 作用	設定“BM7701-00-1 的外部 16MHz 晶振 C _{LOAD} 值”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送			
輸入參數	0~15			
輸出參數	—			
程式詳解	範例程式中，執行此 API 函數，設定“BM7701-00-1 的外部 16MHz 晶振 C _{LOAD} 值”命令，命令數據保存到 TransmitData[]。BM7701-00-1 建議採用 04 作為輸入參數。			
輸入參數	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

註：晶振值僅做參考，具體數值需以實際取得模組數據為準。

API 名稱	void BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)
API 作用	設定“BM7701-00-1 的 Feature”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	數據更新方式：FEATURE_DIR / FEATURE_SET / FEATURE_CLR
輸入參數 2	模式：FEATURE_NO_APPED_NAME / FEATURE_PARAM_UPDATE / FEATURE_PARAM_ERASE / FEATURE_STATUS_EVENT / FEATURE_EXTERNAL32K / FEATURE_FORCE_CALIB / FEATURE_CK32K_OUTPUT (請參考“BLE_API”文檔)
輸出參數	—
程式詳解	<p>範例程式中，執行此 API 函數，設定“BM7701-00-1 的 Feature”命令(詳情參考“BLE_API”文檔)，命令數據保存到 TransmitData[]。</p> <p>輸入參數 1：</p> <ol style="list-style-type: none"> 若輸入參數 1 等於 FEATURE_DIR，輸入參數 2 會直接覆蓋之前的數據。 若輸入參數 1 等於 FEATURE_SET，輸入參數 2 會和之前的數據進行“OR”操作。 若輸入參數 1 等於 FEATURE_CLR，輸入參數 2 會和之前的數據進行“AND”操作。 <p>輸入參數 2：</p> <ol style="list-style-type: none"> 若輸入參數 2 等於 FEATURE_NO_APPED_NAME，廣播數據不會加入藍牙設備名稱。 若輸入參數 2 等於 FEATURE_PARAM_UPDATE，BM7701-00-1 的 RAM 藍牙參數寫到 Flash。 若輸入參數 2 等於 FEATURE_PARAM_ERASE，刪除 BM7701-00-1 的 Flash 的藍牙參數，恢復默認值。 若輸入參數 2 等於 FEATURE_STATUS_EVENT，狀態有變更時，BM7701-00-1 自動發送 Status Event。 若輸入參數 2 等於 FEATURE_EXTERNAL32K，啟用外部 32.768kHz 晶振。 若輸入參數 2 等於 FEATURE_FORCE_CALIB，下次 Software reset 時強制校準。 若輸入參數 2 等於 FEATURE_CK32K_OUTPUT，BM7701-00-1 的 UART2_TX(PB6) 輸出 32kHz 方波。

API 名稱	void BC7701_SetOperateMode(u8 omd,u8 wue,u8 wuw,u8 wut)
API 作用	設定“BM7701-00-1 的休眠模式和 MCU 外部喚醒訊號”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	模式：OP_NORMAL / OP_DEEPSLEEP / OP_POWERDOWN
輸入參數 2	MCU 休眠：ENABLE / DISABLE
輸入參數 3	MCU 外部喚醒訊號寬度(0~8 byte)
輸入參數 4	MCU 外部喚醒訊號延遲時間(0~20ms)
輸出參數	命令封包指針
程式詳解	<p>範例程式中，執行此 API 函數，設定“BM7701-00-1 的休眠模式和 MCU 外部喚醒訊號”命令，命令數據保存到 TransmitData[]。(詳情參考“BLE_API”文檔)。</p> <p>輸入參數 1：</p> <ol style="list-style-type: none"> 若輸入參數 1 等於 OP_NORMAL，BM7701-00-1 進入 Normal 模式。默認情況下 BM7701-00-1 在 Normal 模式。 若輸入參數 1 等於 OP_DEEPSLEEP，BM7701-00-1 進入 Deep Sleep 模式。可使用 API 函數 BC7701_DummyWakeup(u8 leng)將 BM7701-00-1 喚醒到 Normal 模式。 若輸入參數 1 等於 OP_POWERDOWN，BM7701-00-1 進入 Power Down 模式。可使用 API 函數 BC7701_DummyWakeup(u8 leng)將 BM7701-00-1 喚醒到 Normal 模式。 <p>輸入參數 2：</p> <ol style="list-style-type: none"> 若輸入參數 2 等於 ENABLE，說明 MCU 會進入休眠，設定 BM7701-00-1 發出喚醒訊號來喚醒 MCU。 若輸入參數 2 等於 DISABLE，說明 MCU 不會進入休眠，無需設定 BM7701-00-1 發出喚醒訊號。

API 名稱	void BC7701_SetWhiteList(ControlStatus erase,u8 *adr,u8 *mask)
API 作用	設定“BM7701-00-1 的白名單地址”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數 1	清除之前的白名單地址：ENABLE/DISABLE
輸入參數 2	存放白名單地址的 buffer (6 byte)
輸入參數 3	存放地址掩碼的 buffer (6 byte)
輸出參數	—
程式詳解	<p>範例程式中，執行此 API 函數，設定“BM7701-00-1 的白名單地址”命令，命令數據保存到 TransmitData[]。</p> <p>Ex：白名單地址=0x112233445566，地址掩碼=0xFFFFFFFFF00，只有藍牙地址 0x112233445500~0x1122334455FF 可以連接。如果地址掩碼全都寫 0，所有藍牙地址都可以連接。</p>

API 名稱	void BC7701_SetBaudRate(u8 br)
API 作用	設定“BM7701-00-1 的串列傳輸速率”命令，需通過呼叫 BC7701_TransmitPackage 函數進行發送
輸入參數	波特率：BAUD_RATE_4800 / BAUD_RATE_9600 / BAUD_RATE_19200 / BAUD_RATE_25000 / BAUD_RATE_50000 / BAUD_RATE_62500 / BAUD_RATE_115200
輸出參數	—
程式詳解	<p>範例程式中，執行此 API 函數，設定“BM7701-00-1 的串列傳輸速率”命令，命令數據保存到 TransmitData[]。收到 EVENT 後 MCU 需等待約 1ms 後才能改變 UART 串列傳輸速率及執行下一個通訊傳輸。</p>

API 名稱	void BC7701_DummyWakeup(u8 leng)
API 作用	將 BM7701-00-1 從休眠模式中喚醒
輸入參數	Dummy wakeup 信號 (0x00) 佔用 byte
輸出參數	—
程式詳解	<p>範例程式中，執行此 API 函數，將 BM7701-00-1 從休眠模式中喚醒。建議輸入參數設定為 1。</p>

API 名稱	void BC7701_TransmitCmdPackage(u16 opcode,u8 flag)
API 作用	給 BM7701-00-1 發送 BCI “READ”命令
輸入參數 1	BCI 操作碼(請參考“BLE_API”文檔)
輸入參數 2	BCI 旗標(請參考“BLE_API”文檔)
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，MCU 給 BM7701-00-1 發送 BCI “READ”命令

API 名稱	void BC7701_TransmitPackage(u8 length)
API 作用	給 BM7701-00-1 發送 TransmitData[]封包
輸入參數	發送的封包長度
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，MCU 將 TransmitData[]的數據發送給 BM7701-00-1。如果檢測到 TransmitData[0]為 HCI 或 BCI 類型，忽略輸入參數，發送的封包長度等於 TransmitData[1]+2 (HCI 或 BCI 命令的封包長度)。

API 名稱	void BC7701_TransmitPackageConst(tBCI_PACKAGE *pkg)
API 作用	給 BM7701-00-1 發送 BCI 封包數據
輸入參數	存放 BCI 封包數據的 Buffer
輸出參數	—
程式詳解	範例程式中，執行此 API 函數，MCU 給 BM7701-00-1 發送封包數據

API 名稱	u8 BC7701_PackageParserProcess(void)
API 作用	解析 BM7701-00-1 接收到的 EVENT
輸入參數	—
輸出參數	TRUE / FALSE
程式詳解	範例程式中，執行此 API 函數，解析 BM7701-00-1 接收到的 EVENT 輸出參數： 1. 若輸出參數等於 TRUE，說明接收的 EVENT 有效 2. 若輸出參數等於 FALSE，說明接收的 EVENT 無效

API 名稱	#define BC7701_RESET_SET0
API 作用	BM7701-00-1 的 RST_N 腳位拉高
程式詳解	範例程式中，使用該宏定義，BM7701-00-1 的 RST_N 腳位被拉高

API 名稱	#define BC7701_RESET_CLR0
API 作用	BM7701-00-1 的 RST_N 腳位拉低
程式詳解	範例程式中，使用該宏定義，BM7701-00-1 的 RST_N 腳位被拉低

藍牙參數修改(HT32)

- 1、開啟檔案 bleprocess.h，然後點擊 Configuration Wizard 進入“藍牙參數”設定頁面，如下圖 12。接下來將詳細介紹每個參數設定選項。

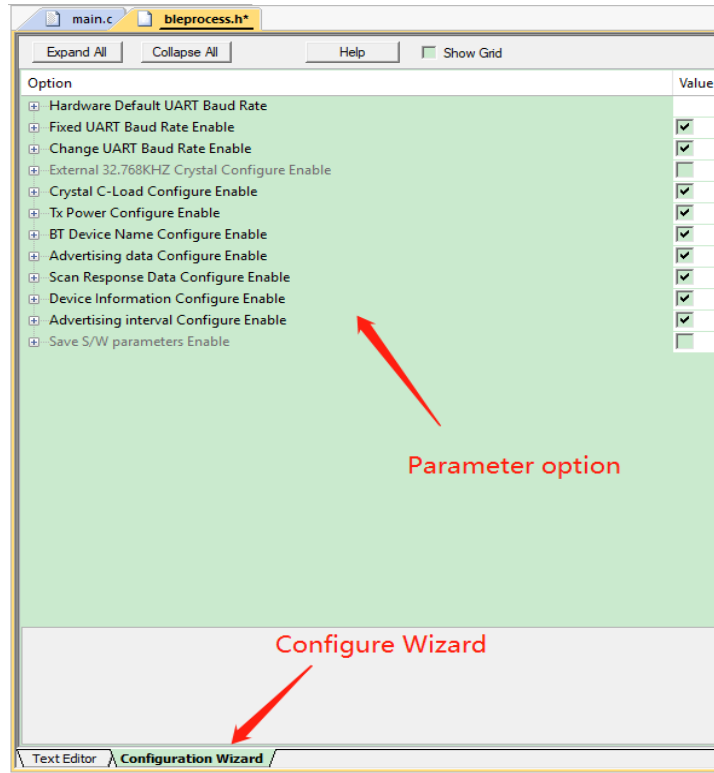


圖 12

- Hardware Default UART Baud Rate：設定上電重置 60ms 後 BM7701-00-1 的初始串列傳輸速率，只有兩個串列傳輸速率 9600bps 和 115200bps 可以選擇，如下圖 13。該方法屬於硬體方式修改 BM7701-00-1 串列傳輸速率(參考表 2)。

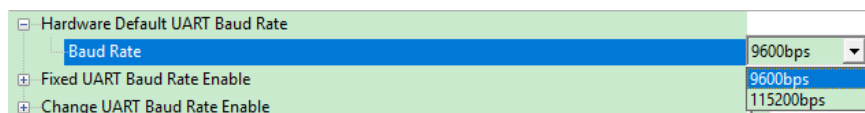


圖 13

- Fixed UART Baud Rate Enable：打√致能“定義 MCU 初始串列傳輸速率”，點擊附加項 Baud Rate 選擇串列傳輸速率，如下圖 14。
 - 禁能：自動設定 MCU 的初始串列傳輸速率等於 BM7701-00-1 的初始串列傳輸速率(9600bps 和 115200bps 兩者選一)。
 - 致能：用戶需要手動選擇 MCU 初始串列傳輸速率。

Ex：當 BM7701-00-1 有保存新的串列傳輸速率參數到 Flash (致能 Save S/W parameters Enable 選項，詳情請參考“參數設定 12 點”)，例如串列傳輸速率 57600bps 保存到 Flash，之後 BM7701-00-1 每次上電重置 60ms 後的初始串列傳輸速率都等於 57600bps。然後用戶必須致能該選項，並選擇 MCU 初始串列傳輸速率等於 57600bps，否則 MCU 和 BM7701-00-1 無法正常通訊。

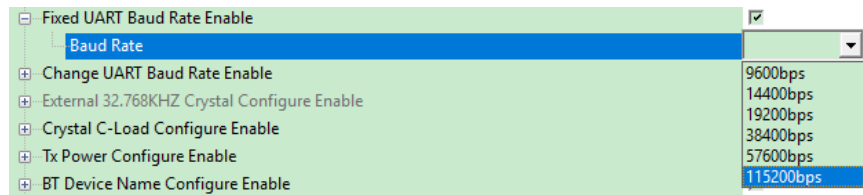


圖 14

- Change UART Baud Rate Enable：打√致能“更改串列傳輸速率”，點擊附加項 Baud Rate 選擇串列傳輸速率，如下圖 15。該選項同時設定 BM7701-00-1 和 MCU 的串列傳輸速率，僅在 MCU 和 BM7701-00-1 的初始串列傳輸速率都設定完成後且相同才會執行 (EX: MCU 向 BM7701-00-1 發送 Change UART Baud Rate Command，然後 MCU 再設定通訊腳位的串列傳輸速率與 BM7701-00-1 一致)。

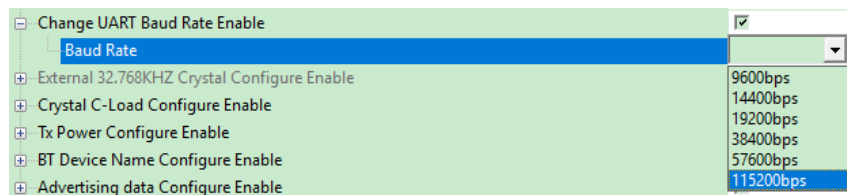


圖 15

- External 32.768kHz Crystal Configure Enable：打√致能“外部 32.768kHz 晶振致能”，如下圖 16。部分模組有外部 32.768kHz 晶振可以致能該選項，應用範例程式默認禁能。注意沒帶外部 32.768kHz 晶振的 BM7701-00-1 模組致能該選項會造成工作失敗。

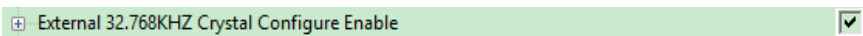


圖 16

- Crystal C-Load Configure Enable：打√致能“更改外部 16MHz 晶振 C_{LOAD} 值”，點擊附加項 Crystal C-Load 輸入數值，取值範圍 0~15，如下圖 17。BM7701-00-1 模組推薦使用數值 4。

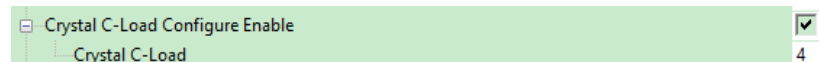


圖 17

輸入參數	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

註：晶振值僅做參考，具體數值需以實際取得模組數據為準。

表 8

- Tx Power Configure Enable：打√致能“更改輸出功率值”，點擊附加項 Tx Power 輸入數值，取值範圍 0~15，如下圖 18。用戶可以根據傳輸距離或功耗靈活調整。

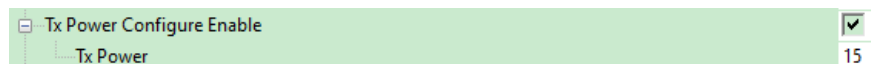


圖 18

輸入參數	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

註：輸出功率值僅做參考，具體數值需以實際得到模組數據為準。

表 9

- BT Device Name Configure Enable：打√致能“更改藍牙名稱”，如下圖 19。成功更改後的藍牙名稱為“BM7701”。用戶也可以在 bleprocess.c 頁面自定義藍牙名稱，找到下圖 20 紅框位置更改，最大長度為 31 Byte。

BT Device Name Configure Enable

圖 19

```

24 #if (_BDNAME_CFG_ENABLE_ == 1)
25 /* set BT device name */
26 uc8 BLE_DeviceName[] = {'B','M','7','7','0','1'};
27 #endif
28
    
```

圖 20

- Advertising data Configure Enable：打√致能“更改廣播數據”，如下圖 21。用戶也可以在 bleprocess.c 頁面自定義廣播數據，找到下圖 22 紅框位置更改，最大長度為 31 Byte。

Advertising data Configure Enable

圖 21

```

29 #if (_ADV_DATA_CFG_ENABLE_ == 1)
30 /* set Advertising Data */
31 uc8 BLE_AdvData[] =
32 {
33     /* flag */
34     2,0x01,0x06,
35     /* Manufacturer Specific Data */
36     11,0xFF,0xFF,0xFF,
37     'B','e','s','t','C','o','m','m'
38 };
39 #endif
    
```

圖 22

- Scan Response Data Configure Enable：打√致能“更改掃描回應數據”，如下圖 23。用戶也可以在 bleprocess.c 頁面自定義掃描回應數據，找到下圖 24 紅框位置更改，最大長度為 31 Byte。

Scan Response Data Configure Enable

圖 23

```

41 #if (_SCAN_DATA_CFG_ENABLE_ == 1)
42 /* Set Scan Response Data */
43 uc8 BLE_ScanData[] =
44 {
45     /* complete List of 16-bit Service Class UUIDs */
46     11,0x03,0x00,0x18,0x01,0x18,0x0A,0x18,0x0F,0x18,0xF0,0xFF
47 };
48 #endif
49
    
```

圖 24

- Advertising interval Configure Enable：打√致能“更改廣播間隔”，點擊附加項輸入數值，如下圖 25。注意 Min Interval ≤ Max Interval。

Advertising interval Configure Enable

Min Interval(ms)	100
Max Interval(ms)	100

圖 25

- Device Information Configure Enable：打√致能“更改設備資訊”，如下圖 26。用戶也可以在 bleprocess.c 頁面自定義設備信息。找到下圖 27 紅框位置更改。

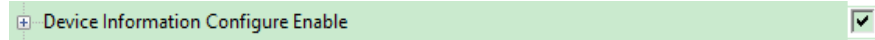


圖 26

```

50 #if ( DEV_INF_CFG_ENABLE == 1)
51 uc8 BLE_DevInfSysID[] = { 0x37,0x37,0x30,0x31 };
52 uc8 BLE_DevInfModelNumber[] = {'B','M','7','7','0','1'}; //max 16 byte
53 uc8 BLE_DevInfSerialNumber[] = {'1','.','.','0','0'};
54 uc8 BLE_DevInfFirmwareRevs[] = {'1','.','.','0','0'};
55 uc8 BLE_DevInfHardwareRevs[] = {'1','.','.','0','0'};
56 uc8 BLE_DevInfSoftwareRevs[] = {'1','.','.','0','0'};
57 uc8 BLE_DevInfManufacturer[] = {'B','e','s','t','C','o','m'}; //max 16 byte
58 uc8 BLE_DevInfIEEE11073[] = {0x00,0x00,0x00,0x00};
59 uc8 BLE_DevInfPnPID[] = {0x00,0x00,0x00,0x00,0x00,0x00,0x00};
60 #endif
61
    
```

圖 27

- Save S/W parameters Enable：打√致能“保存參數”。“參數設定 3~11 點”的藍牙參數保存到 BM7701-00-1 的 Flash，如下圖 28。前提是對應的設定選項有致能（通過 Command 設定 BM7701-00-1 的藍牙參數會暫時保存到 RAM，致能選項後 RAM 的藍牙參數會保存到 Flash，重新上電重置後 BM7701-00-1 就會自動使用 Flash 的藍牙參數）。

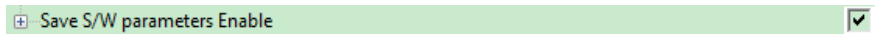


圖 28

2、開啟檔案 main.c，然後點擊 Configuration Wizard 可以設定“資料傳輸”和“MCU 休眠”的相關的參數。

- Change Connect Interval Enable：打√致能“修改 BM7701-00-1 連接間隔”，如下圖 29。點擊附加項輸入連接間隔數值。

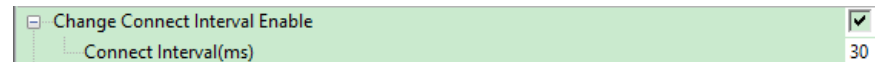


圖 29

- MCU sleep mode Enable / disable：打√致能“MCU 進入休眠”。點擊附加項輸入時間。如下圖 30，致能該選項後，BM7701-00-1 在 500ms 內沒有與連接的設備(手機)傳輸資料，MCU 會主動進入休眠模式。



圖 30

- WakeUp Signal Configure：點擊附加項選擇“喚醒訊號寬度(byte)”和“喚醒訊號延長時間(ms)”。如下圖 31，當 MCU 需要從休眠模式被喚醒，可設定 BM7701-00-1 發送 2 byte 0x00 喚醒 MCU，然後等待 8ms 把接收到的數據 Event 傳給 MCU。喚醒訊號寬度和延長時間取決於 MCU 的休眠模式設定和喚醒穩定時間，用戶可以根據實際情況靈活調整。



圖 31

- 為了讓用戶能夠清楚串列傳輸速率選項的區別以及 Flash 保存的影響，舉以下例子說明在選擇不同的串列傳輸速率要如何進行操作（如無特殊說明，串列傳輸速率都是指 MCU 和 BM7701-00-1 的串列傳輸速率）。

例 1：只選擇 9600bps 或 115200bps 串列傳輸速率。操作步驟如下表 10。

- 上電重置 60ms 後串列傳輸速率都為 9600bps 或 115200bps。

Hardware Default UART Baud Rate	Fixed UART Baud Rate Enable	Change UART Baud Rate Enable	Save S/W Parameters Enable
(9600bps 或 115200bps)	Disable	Disable	Disable

表 10

例 2：任意串列傳輸速率，例如選擇 57600bps，串列傳輸速率不保存到 BM7701-00-1 的 Flash 的操作步驟如下表 11。

- 上電重置 60ms 後的初始串列傳輸速率為 9600bps 或 115200bps。
- 待執行到 Change UART Baud Rate Command，串列傳輸速率更改為 57600bps。
- 每次重新上電重置都會順序執行以上動作。

Hardware Default UART Baud Rate	Fixed UART Baud Rate Enable	Change UART Baud Rate Enable	Save S/W Parameters Enable
(9600bps 或 115200bps)	Disable	Enable (57600bps)	Disable

表 11

例 3：任意串列傳輸速率，例如選擇 57600bps，串列傳輸速率保存到 BM7701-00-1 的 Flash。該方式必須執行兩次燒錄動作。操作步驟如下表 12。

- 第一次燒錄
 - 燒錄程式後，第一次上電重置 60ms 後的初始串列傳輸速率為 9600bps 或 115200bps。
 - 待執行 Change UART Baud Rate Command，串列傳輸速率為 57600bps。
 - 保存串列傳輸速率到 BM7701-00-1 的 Flash。
 - 接下來每次上電重置，BM7701-00-1 的串列傳輸速率都一直為 57600bps，而 MCU 的串列傳輸速率為 9600bps 或 115200bps，無法正常通訊。
 - 需要執行第二次燒錄動作。
- 第二次燒錄
 - MCU 和 BM7701-00-1 可正常通訊（致能 Fixed UART Baud Rate Enable，並更改 MCU 的初始串列傳輸速率為 57600bps）。

	Hardware Default UART Baud Rate	Fixed UART Baud Rate Enable	Change UART Baud Rate Enable	Save S/W Parameters Enable
第一次燒錄	(9600bps 或 115200bps)	Disable	Enable (57600bps)	Enable
第二次燒錄	(9600/115200)	Enable (57600bps)	Enable (57600bps)	Enable

表 12

藍牙參數修改(HT8)

1、開啟檔案 bleprocess.h，找到下圖 32 的位置進行“藍牙參數”設定。接下來將詳細介紹每個參數設定選項。

```

7  /*----- Change UART Baud Rate Enable/Disable -----*/
8  #define _CHG_BAUD_RATE_ENABLE_ 1
9  // #define _CHANGE_BAUDRATE_ BAUD_RATE_19200
10 // #define _CHANGE_BAUDRATE_ BAUD_RATE_25000
11 // #define _CHANGE_BAUDRATE_ BAUD_RATE_50000
12 #define _CHANGE_BAUDRATE_ BAUD_RATE_115200
13 /*----- Change Crystal C-Load Enable/Disable -----*/
14 #define _CLOAD_CFG_ENABLE_ 1
15 #define _CLOAD_VALUE_ 04
16 /*----- Change TX Power Enable/Disable -----*/
17 #define _TXPWR_CFG_ENABLE_ 1
18 #define _TX_POWER_VALUE_ 10
19 /*----- Configure BT Device Name Enable/Disable -----*/
20 #define _BDNAME_CFG_ENABLE_ 1
21 /*----- Configure Advertising data Enable/Disable -----*/
22 #define _ADU_DATA_CFG_ENABLE_ 1
23 /*----- Configure Scan Response Data Enable/Disable -----*/
24 #define _SCAN_DATA_CFG_ENABLE_ 0
25 /*----- Configure Advertising interval Enable/Disable -----*/
26 #define _ADU_INTU_CFG_ENABLE_ 1
27 /*----- Min Interval(ms)<10~40000 -----*/
28 #define _ADU_INTU_MIN_VALUE_ (1000)
29 /*----- Max Interval(ms)<10~40000 -----*/
30 #define _ADU_INTU_MAX_VALUE_ (1000)
31 /*----- Configure Feature Setup -----*/
32 // #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT+FEATURE_EXTERNAL32K) /* external 32.768K */
33 #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT)
34 /*----- Power On Finish BT7701 to Sleep mode Enable/Disable -----*/
35 #define _PWRON_FINISH_SLEEP_ 0
    
```

圖 32

- Change UART Baud Rate

- #define _CHG_BAUD_RATE_ENABLE_ 1：致能“更改串列傳輸速率”
- #define _CHG_BAUD_RATE_ENABLE_ 0：除能“更改串列傳輸速率”
- #define _CHANGE_BAUDRATE_ BAUD_RATE_115200：串列傳輸速率設定為 115200bps。用戶也可以根據需求定義其它串列傳輸速率

```

7  /*----- Change UART Baud Rate Enable/Disable -----*/
8  #define _CHG_BAUD_RATE_ENABLE_ 1
9  // #define _CHANGE_BAUDRATE_ BAUD_RATE_19200
10 // #define _CHANGE_BAUDRATE_ BAUD_RATE_25000
11 // #define _CHANGE_BAUDRATE_ BAUD_RATE_50000
12 #define _CHANGE_BAUDRATE_ BAUD_RATE_115200
    
```

圖 33

- Change Crystal C-Load

- #define _CLOAD_CFG_ENABLE_ 1：致能“更改外部 16MHz 晶振 C_{LOAD}值”
- #define _CLOAD_CFG_ENABLE_ 0：除能“更改外部 16MHz 晶振 C_{LOAD}值”
- #define _CLOAD_VALUE_ 04：外部 16MHz 晶振 C_{LOAD}值設定為 04

```

13 /*----- Change Crystal C-Load Enable/Disable -----*/
14 #define _CLOAD_CFG_ENABLE_ 1
15 #define _CLOAD_VALUE_ 04
    
```

圖 34

輸入參數	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

註：晶振值僅作參考，具體數值需以實際取得模組數據為準。

表 13

- Change Tx Power

- #define _TXPWR_CFG_ENABLE_ 1：致能“更改輸出功率”
- #define _TXPWR_CFG_ENABLE_ 0：除能“更改輸出功率”

➤ #define _TX_POWER_VALUE_ 10：輸出功率值設定為 10

```

16 /*----- Change TX Power Enable/Disable -----*/
17 #define _TX_PWR_CFG_ENABLE_ 1
18 #define _TX_POWER_VALUE_ 10
    
```

圖 35

輸入參數	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

註：輸出功率值僅作參考，具體數值需以實際得到模組數據為準。

表 14

- Configure BT Device Name

➤ #define _BDNAME_CFG_ENABLE_ 1：致能“更改藍牙設備名稱”

➤ #define _BDNAME_CFG_ENABLE_ 0：除能“更改藍牙設備名稱”

```

19 /*----- Configure BT Device Name Enable/Disable -----*/
20 #define _BDNAME_CFG_ENABLE_ 1
    
```

圖 36

用戶也可以打開 bleprocess.c 頁面找到下圖 37 的紅框部分來自定義藍牙設備名稱，注意長度≤ 31 byte。

```

38 /*--- set BT device name ---*/
39 #if (_BDNAME_CFG_ENABLE_ == 1)
40 const tBCI_PACKAGE BLE_SetDeviceName =
41 {
42     BCI_CMD_PKG,           //head type
43     3+6,                   //length
44     0x00,                  //Flag
45     BCI_DEU_NAME,         //opcode
46     {'B','M','7','7','0','1'} //parameter
47 };
48 #endif
    
```

圖 37

- Configure Advertising data

➤ #define _ADV_DATA_CFG_ENABLE_ 1：致能“更改藍牙數據”

➤ #define _ADV_DATA_CFG_ENABLE_ 0：除能“更改藍牙數據”

```

21 /*----- Configure Advertising data Enable/Disable -----*/
22 #define _ADU_DATA_CFG_ENABLE_ 1
    
```

圖 38

用戶也可以打開 bleprocess.c 頁面找到下圖 39 的紅框部分來自定義藍牙廣播數據，注意長度≤ 31 byte。

```

50 /*--- set Advertising Data ---*/
51 #if (_ADU_DATA_CFG_ENABLE_ == 1)
52 const tBCI_PACKAGE BLE_SetAdvData =
53 {
54     BCI_CMD_PKG,           //head type
55     3+15,                  //length
56     UPDAE_AUTO_NAME,      //Flag=auto add device name
57     BCI_ADU_DATA,         //opcode
58     {
59         /* flag */
60         2, 0x01, 0x06,
61         /* Manufacturer Specific Data */
62         11, 0xFF, 0xFF, 0xFF,
63         'B','e','s','t','C','o','m','m'
64     },
65 };
66 #endif
    
```

圖 39

- Configure Scan Response Data

- #define_SCAN_DATA_CFG_ENABLE_ 1：致能“更改掃描響應數據”
- #define_SCAN_DATA_CFG_ENABLE_ 0：除能“更改掃描響應數據”

```

23 /*----- Configure Scan Response Data Enable/Disable -----*/
24 #define _SCAN_DATA_CFG_ENABLE_ 0
    
```

圖 40

用戶也可以打開 bleprocess.c 頁面找到下圖 41 的紅框部分來自定義掃描響應數據，注意長度≤ 31 byte。

```

68 /*--- Set Scan Response Data ---*/
69 #if (_SCAN_DATA_CFG_ENABLE_ == 1)
70 const tBCI_PACKAGE BLE_SetScanData =
71 {
72     BCI_CMD_PKG,           //head type
73     3+12,                 //length
74     0x00,                 //Flag
75     BCI_SCAN_DATA,       //opcode
76     {
77         /* complete List of 16-bit Service Class UUIDs */
78         11, 0x03, 0x00, 0x18, 0x01, 0x18, 0x0A, 0x18, 0x0F, 0x18, 0xF0, 0xFF
79     };
80 };
81 #endif
    
```

圖 41

- Configure Advertising interval Enable/Disable

- #define_ADV_INTV_CFG_ENABLE_ 1：致能“更改廣播間隔”
- #define_ADV_INTV_CFG_ENABLE_ 0：除能“更改廣播間隔”
- #define _ADV_INTV_MIN_VALUE_ (1000)：最小廣播間隔設定為 1000ms
- #define _ADV_INTV_MAX_VALUE_ (1000)：最大廣播間隔設定為 1000ms

```

26 #define _ADV_INTV_CFG_ENABLE_ 1
27 /*----- Min Interval(ms)<10-40000> -----*/
28 #define _ADV_INTV_MIN_VALUE_ (1000)
29 /*----- Max Interval(ms)<10-40000> -----*/
30 #define _ADV_INTV_MAX_VALUE_ (1000)
    
```

注意 MIN_VALUE ≤ MAX_VALUE

圖 42

- Configure Feature setup

- #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT+FEATURE_EXTERNAL32K)：開啟“狀態有變更時 BM7701-00-1 自動發送 Status Event + 外部 32.768KHz 晶振致能。”
- #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT)：開啟“狀態有變更時 BM7701-00-1 自動發送 Status Event。”

```

31 /*----- Configure Feature setup -----*/
32 // #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT+FEATURE_EXTERNAL32K) /* external 32.768K */
33 #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT)
    
```

圖 43

- Power On finish BC7701 to Sleep mode Enable/Disable
 - #define _PWRON_FINISH_SLEEP_ 0：參數配置成功後不進入休眠。
 - #define _PWRON_FINISH_SLEEP_ 1：參數配置成功後進入休眠。

```

34  /*----- Power On finish BC7701 to Sleep mode Enable/Disable -----*/
35  #define _PWRON_FINISH_SLEEP_ 1
    
```

圖 44

2、開啟檔案 main.c，找到下圖的位置設定“資料傳輸”和“MCU 休眠”的相關的參數。

- Change Connect Interval
 - #define _CHG_CONNECT_INTV_ 1：致能“更改連接間隔”
 - #define _CHG_CONNECT_INTV_ 0：除能“更改連接間隔”
 - #define _CNNT_INTV_MIN_VALUE_ (30)：最小連接間隔設定為 30ms
 - #define _CNNT_INTV_MAX_VALUE_ (30)：最大連接間隔設定為 30ms

```

64  #define _CHG_CONNECT_INTV_ 1
65  /*----- Min Interval(ms)<10-40000> -----*/
66  #define _CNNT_INTV_MIN_VALUE_ (30)
67  /*----- Max Interval(ms)<10-40000> -----*/
68  #define _CNNT_INTV_MAX_VALUE_ (30)
    
```

圖 45

- Configure MCU sleep mode
 - #define _MCU_SLEEP_ENABLE_ 1：致能“MCU 休眠”
 - #define _MCU_SLEEP_ENABLE_ 0：除能“MCU 休眠”
 - #define _SLEEP_DELAY_TIMER_ (500/2)：如果 250ms 內沒有與連接的設備(手機)傳輸資料，MCU 會進入休眠模式。
 - #define _MASTER_WUW_VALUE_ (1)
 - #define _MASTER_WUT_VALUE_ (4)
- 如下圖 46，當 MCU 需要從休眠模式被喚醒，可設定 BM7701-00-1 發送 1 byte 0x00 喚醒 MCU，然後等待 4ms 把接收到的數據 Event 傳給 MCU。喚醒訊號寬度和延長時間取決於 MCU 的休眠模式設定和喚醒穩定時間，用戶可以根據實際情況靈活調整。

```

87  /* MCU sleep mode Enable/disable */
88  #define _MCU_SLEEP_ENABLE_ 1
89  /* Enter deep sleep delay timer(ms)<10-5000> */
90  #define _SLEEP_DELAY_TIMER_ (500/2)
91  /* Wake up Signal Width(byte)<1-8> */
92  #define _MASTER_WUW_VALUE_ (1)
93  /* Wake up Signal delay time(ms)<0-20> */
94  #define _MASTER_WUT_VALUE_ (4)
    
```

圖 46

結論

本文介紹 BM7701-00-1 模組比較常用的 API 函數命令，結合應用程式指導讀者使用 HT8 和 HT32 MCU 配置 BM7701-00-1 模組的藍牙參數，並與手機 App 互相傳輸資料。通過該範例展示模組的操作方法以及出色的雙向資料傳輸的能力，幫助讀者更快地運用到相關電子產品，例如家用電器和醫療產品等等。

參考資料

參考文件 BM7701-00-1 Datasheet、BLE_API、BLE_Service。

如需進一步瞭解，敬請瀏覽 Holtek 官方網站 www.holtek.com.tw。

版本及修訂說明

日期	作者	發行	修訂說明
2022.06.23	阮義展	V1.00	第一版

免責聲明

本網頁所載的所有資料、商標、圖片、連結及其他資料等（以下簡稱「資料」），只供參考之用，盛群半導體股份有限公司及其關聯企業（以下簡稱「本公司」）將會隨時更改資料，並由本公司決定而不作另行通知。雖然本公司已盡力確保本網頁的資料準確性，但本公司並不保證該等資料均為準確無誤。本公司不會對任何錯誤或遺漏承擔責任。

本公司不會對任何人士使用本網頁而引致任何損害（包括但不限於電腦病毒、系統故障、資料損失）承擔任何賠償。本網頁可能會連結至其他機構所提供的網頁，但這些網頁並不是由本公司所控制。本公司不對這些網頁所顯示的內容作出任何保證或承擔任何責任。

責任限制

在任何情況下，本公司並不須就任何人由於直接或間接進入或使用本網站，並就此內容上或任何產品、資訊或服務，而招致的任何損失或損害負任何責任。

管轄法律

本免責聲明受中華民國法律約束，並接受中華民國法院的管轄。

免責聲明更新

本公司保留隨時更新本免責聲明的權利，任何更改於本網站發佈時，立即生效。