

BM7701-00-1 应用范例

文件编号: AN0614SC

简介

BM7701-00-1 是一个基于 SOC BC7701 芯片、BLE 透明传输的蓝牙低功耗收发模块。该模块可以无线控制外部设备,支持双向数据传输,适用于照明产品、医疗保健产品和家用电器。本文将介绍使用 Holtek 32-bit MCU HT32F52352 和 8-bit MCU HT66F2370 作为微控制器,控制模块 BM7701-00-1 修改蓝牙参数并与手机 APP 进行数据传输,可以实现按键发送数据、LED 灯显示接收状态。通过蓝牙模块与手机 APP 互相对传的范例,展示蓝牙模块双向数据传输的能力,可帮助用户灵活搭载到不同的电子产品中。

功能说明

通信接口

BM7701-00-1 模块需要使用 UART 控制。UART 是一种通用串行数据总线,用于异步通信。该总线双向通信可以实现全双工传输和接收。BM7701-00-1 提供两个 UART 通信接口 UART1/2,本次应用范例统一使用 UART1 接口。

引脚序号	引脚名称	功能描述
1	UART1_TX	BLE UART1_TX 串行数据输出
2	UART1_RX	BLE UART1_RX 串行数据输入
3	UART2_TX	BLE UART2_TX 串行数据输出
4	UART2_RX	BLE UART2_RX 串行数据输入

表 1. BM7701-00-1 UART 接口

RST_N=0→1(从低电平拉到高电平/上电复位)后, TX/RX I/O 必须在最初 60ms 保持固定, BM7701-00-1 会根据 I/O 状态选择 UART 波特率。然后就可以将 I/O 更改回 UART TX/RX。如下表 2 所示。

上电复位后 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 波特率选择

UART 格式

UART 数据格式需要满足以下要求：

1. 波特率：1953bps~115200bps (BM7701-00-1)
2. 数据位：8 bits
3. 奇偶校验位：No
4. 停止位：1 bit

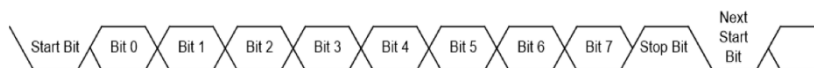


图 1. UART 时序图

Command/Event 格式

BM7701-00-1 支持三种类型的 Command，包括 CmdType1、CmdType2 和 CmdType3。CmdType1 是蓝牙规范中定义的 HCI。CmdType2 支持访问 BLE Services。CmdType3 是设定 BM7701-00-1 的内部参数，包括 Tx Power、晶振负载电容、开关广播、广播间隔、连接间隔等等。MCU 通过 UART 向 BM7701-00-1 写入 Command，BM7701-00-1 成功识别就会返回 Event。详情请参考 BLE_API 文档。本应用范例只用到 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. 样例

使用指南

本章节介绍了环境准备、操作说明、硬件说明、软件说明、模块 API 命令函数介绍、蓝牙参数修改等信息，希望用户通过了解应用范例，快速掌握 BM7701-00-1 模块的基本操作与系统架构。

环境准备

硬件

- HT32: 蓝牙模块 BM7701-00-1、转板 BCT-7701-001、开发板 BCE-GenTrx32-002(搭载 MCU HT32F52352)
- HT8: 蓝牙模块 BM7701-00-1、转板 BCT-7701-001、开发板 BCE-GenTrx8-001(搭载 MCU HT66F2370)

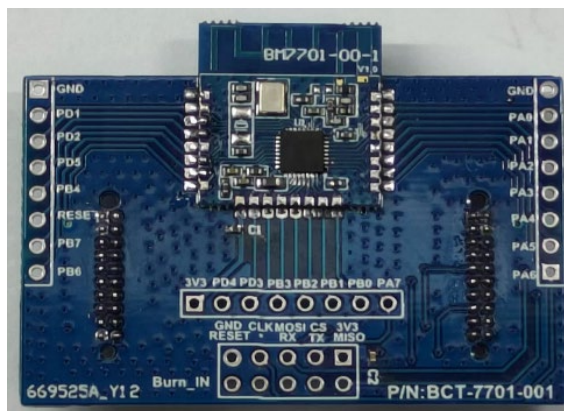


图 2. 模块+转板

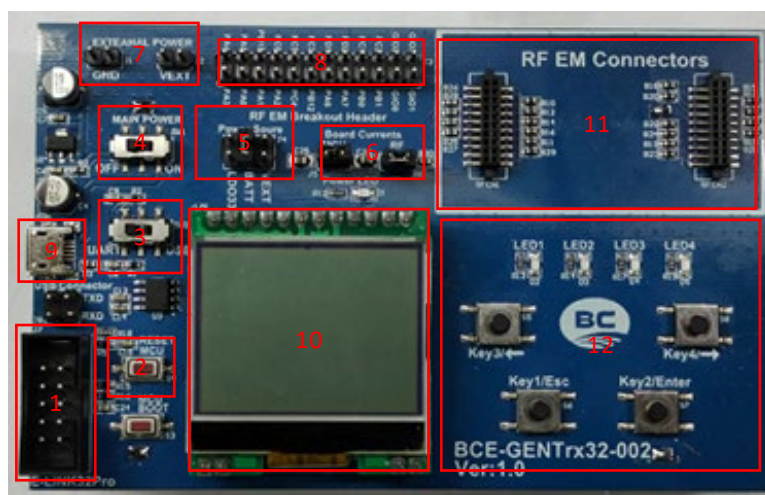


图 3. 开发板(BCE-GENTrx32-002)

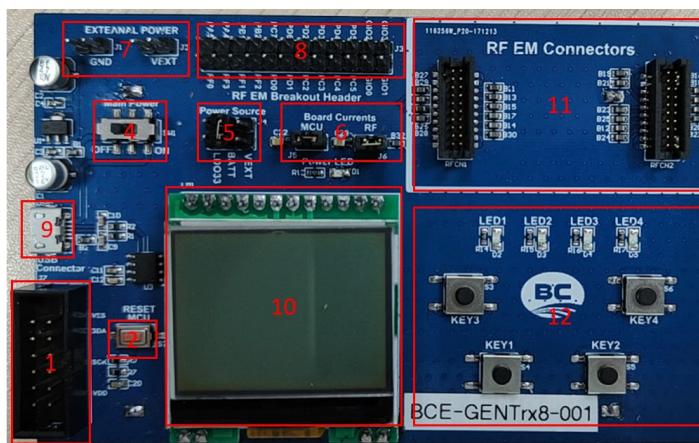


图 4. 开发板(BCE-GENTrx8-001)

1. JTAG 接口，可配合 IDE 接口仿真程序及下载程序用。
2. 系统复位键。
3. UART/USB 接口选择 (只有 BCE-GENTrx32-002 开发板有此项)。
4. 总电源开关，左拨(OFF)为关闭电源，右拨(ON)为开启电源。
5. 系统电源选择，跳线(Jumper)于左侧 LDO33 处，表示电源由 USB 口输入；跳线于中间 BATT 处，代表电源使用电池座(板背：两颗 1.5V AA 电池)；跳线于右侧 VEXT 处，则是使用外部电源接点供电，注意若是使用外部电源接点输入电压，电压不得超 3.6V。
6. MCU 和 BM7701-00-1 模块板电源电流检测点。
7. 外部电源接点。
8. MCU 的部分 I/O 端口。
9. Micro USB 接口，可用来作为系统电源输入(电源选择 LDO33)。
10. 液晶显示器(支持 128×64 点)，使用方法请参考本文范例程序，内含显示器控制程序库。
11. 射频模块接口是射频发射/接收装置连接处，此范例则安装上 BCT-7701-001+BM7701-00-1。
12. LED×4 及按键×4，此范例用于显示数据接收状态和发送数据。

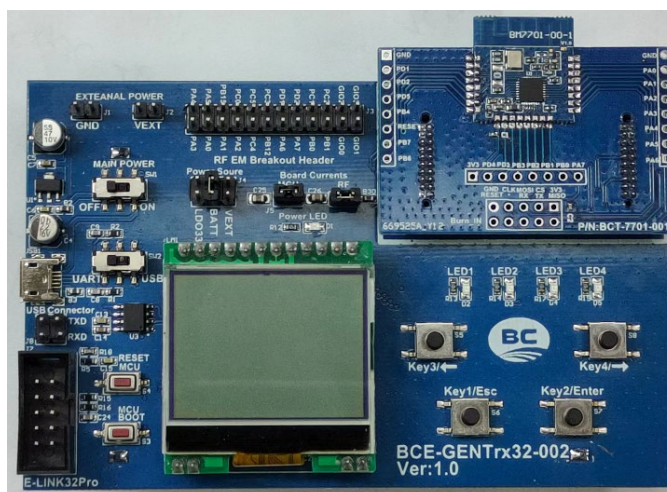


图 5. 组装完成图(HT32)

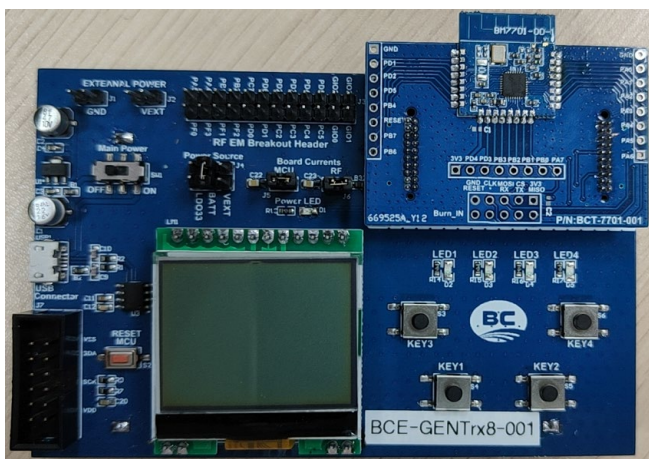


图 6. 组装完成图(HT8)

软件：手机端需要下载 APP BLEDemo，相关的链接如下图所示。



图 7. Android(左) & iOS(右)

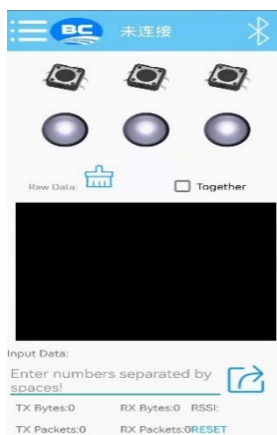


图 8. 安装后的 UI

操作说明

1. 使用 elink-32 Pro 烧录应用范例程序到开发板 BCE-GENTrx32-002 上的 HT32F52352 (HT32) 。
使用 elink 烧录应用范例程序到开发板 BCE-GENTrx8-001 上的 HT66F2370 (HT8)。
2. 打开开发板开关，成功上电后 Power LED 常亮，LCD 屏有显示。
3. LCD 等待 1s 的开机界面后会显示状态界面，然后等待显示 PWR ON OK，即蓝牙设定配置成功且开启广播。
4. 点击手机 APP 右上角进入扫描蓝牙设备，选择名称“BM7701”连接。连接成功后，APP 右上角连接 icon 会变成红色。

5. 开发板按键操作。

- 按键 1 按下：APP 上面对应 LED1 icon 亮
- 按键 2 按下：APP 上面对应 LED2 icon 亮
- 按键 3 按下：APP 上面对应 LED3 icon 亮

6. APP 操作。

- 按键 1 icon 按下：LED1 亮
- 按键 2 icon 按下：LED2 亮
- 按键 3 icon 按下：LED3 亮

硬件说明

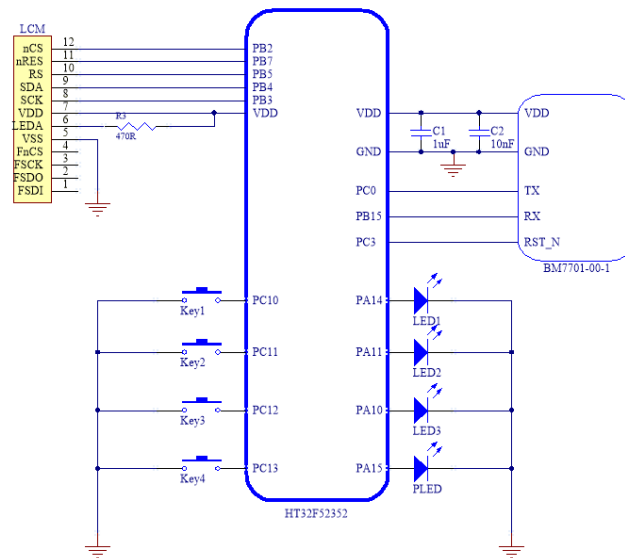


图 9. 应用电路图(HT32)

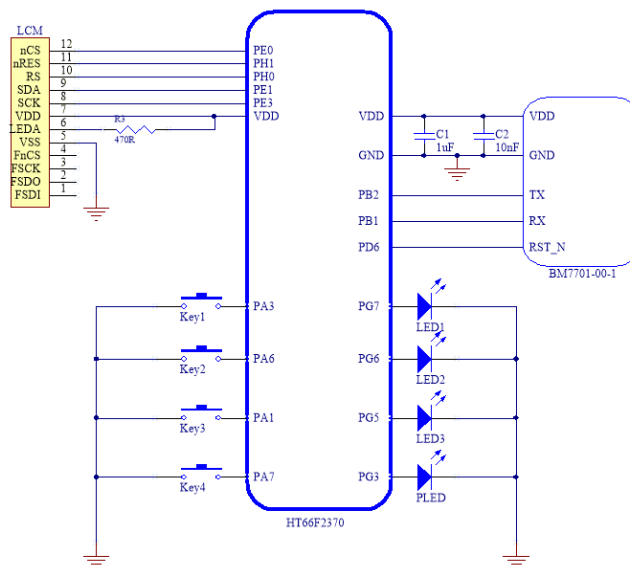


图 10. 应用电路图(HT8)

软件说明

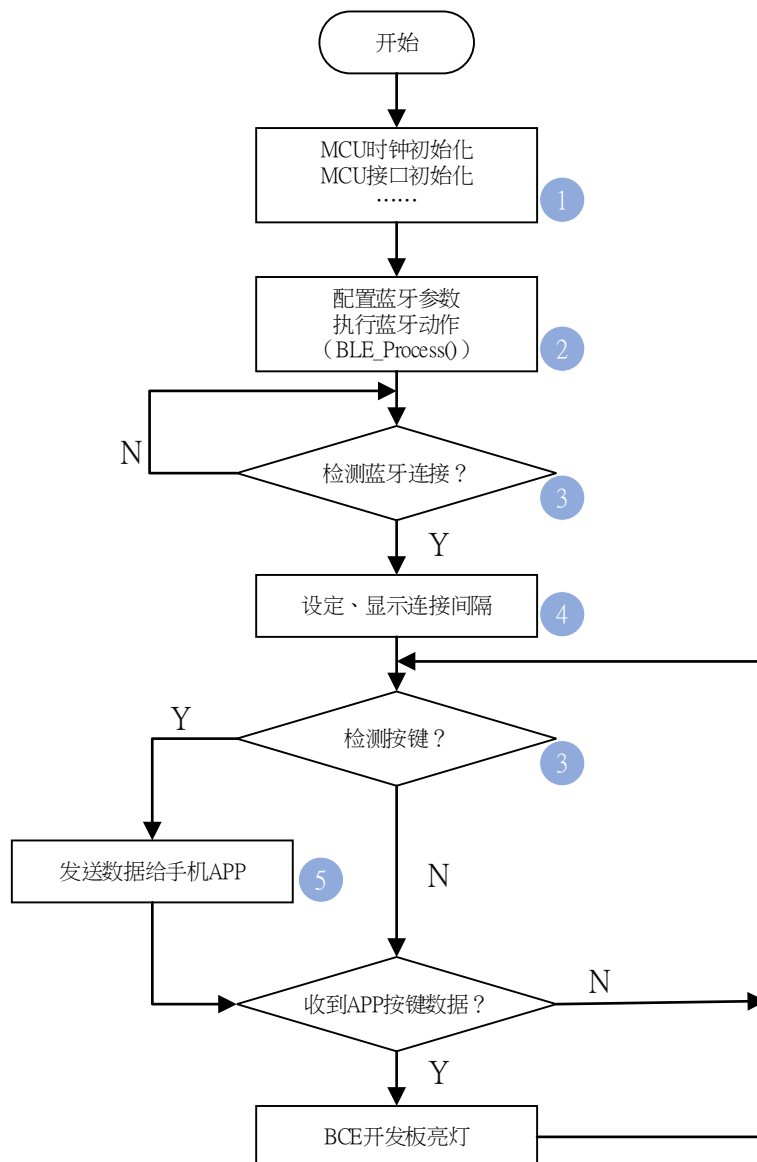


图 11. 软件流程图

- 先进行 MCU 时钟初始化，LED、LCD、按键等外围器件初始化。
- MCU 发送 Command 给 BM7701-00-1 来配置蓝牙参数、开启广播。
- 手机 APP 与 BM7701-00-1 蓝牙连接成功。
- MCU 检测到开发板按键会发送 Command 给 BM7701-00-1，BM7701-00-1 再发送数据给手机 APP。
- 手机 APP 收到指示也会发送数据给 BM7701-00-1。MCU 解析 BM7701-00-1 收到的数据 Event 并点亮开发板对应的 LED。

1、初始化更新函数

- BC7701_InterfaceConfigure(_UART_BAUDRATE_): 设定与 BM7701-00-1 通信的 MCU I/O 状态
- BC7701_RESET_CLR(): BM7701-00-1 的 RST_N 引脚拉低
- BC7701_RESET_SET(): BM7701-00-1 的 RST_N 引脚拉高
- BC7701_HardwareBaudRateDefault(_UART_BAUDRATE_): 通过通信引脚初始化 BM7701-00-1 波特率
- BC7701_HardwareBaudRateRelease(): 完成 BM7701-00-1 波特率初始化设定

2、蓝牙参数设定

- BC7701_SendBCIReadPackage(BCI_VERSION, 0x80): 检查版本(HT32)
BC7701_TransmitCmdPackage(BCI_VERSION, 0x80): 检查版本(HT8)
- BC7701_SendBCIReadPackage(BCI_DEV_ADDRESS, 0x00): 获取地址(HT32)
BC7701_TransmitCmdPackage(BCI_DEV_ADDRESS, 0x00): 获取地址(HT8)
- BC7701_SetBaudRate(), BC7701_SetTxPower()...: 设定 BM7701-00-1 的波特率、传出功率等等，需要使能相应的选项(HT8)
BC7701_SetBaudRate(), BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BLE_SetTxPower)...: 设定 BM7701-00-1 的波特率、传出功率等等，需要使能相应的选项(HT8)
- BC7701_SetupFeatureFlag(FEATURE_SET, FEATURE_STATUS_EVENT): 设定 BM7701-00-1 在状态改变时发送数据(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BLE_SetFeatureState): 设定 BM7701-00-1 在状态改变时发送数据(HT32)
- BC7701_AdvertisingControl(ENABLE): 设定 BM7701-00-1 开启广播(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BC7701_AdvertiseEnable): 设定 BM7701-00-1 开启广播(HT8)

3、进入深度休眠

- BC7701_SetOperateMode(OP_DEEPSLEEP, ENABLE, _MASTER_WUW_VALUE_, _MASTER_WUT_VALUE_): 设定 BM7701-00-1 进入深度休眠模式和 MCU 外部唤醒信号(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BC7701_OperateSleep): 设定 BM7701-00-1 进入深度休眠模式和 MCU 外部唤醒信号(HT8)

4、设定连接间隔

- BC7701_DummyWakeup(): 将 BM7701-00-1 从休眠模式中唤醒
- BC7701_ConnectIntervalModify(u16 opcode, u16 min, u16 max): 设定 BM7701-00-1 的连接间隔(HT32)
BC7701_TransmitPackageConst((tBCI_PACKAGE *)&BLE_CmdIndex): 设定 BM7701-00-1 的连接间隔(HT8)

5、数据发送

- BC7701_DummyWakeup()：将 BM7701-00-1 从休眠模式中唤醒
- BC7701_SendBCIPackage(NotifyFFF1, BCI_ServiceProperties, len, pbuf)：BM7701-00-1 发送无线数据给手机 APP

模块 API 命令函数介绍

1. 以下表格整合 BM7701-00-1 模块 API 命令函数与其功能, 请参考应用范例程序的 BC7701.c 档案(HT32)。

API 原型	功能说明
void BC7701_InterfaceConfigure(u8 br)	设定与 BM7701-00-1 通信的 MCU I/O 状态
void BC7701_UARTConfigure(u8 br)	设定 MCU 的通信引脚为 UART 模式和波特率
void BC7701_RESET_SET(void)	BM7701-00-1 的 RST_N 引脚拉高
void BC7701_RESET_CLR(void)	BM7701-00-1 的 RST_N 引脚拉低
void BC7701_HardwareBaudRateDefault(u8 br)	通过通信引脚初始化 BM7701-00-1 波特率
void BC7701_HardwareBaudRateRelease(void)	完成 BM7701-00-1 波特率初始化设定
void BC7701_HardwareReset(void)	硬件复位, 初始化蓝牙参数
void *BC7701_SoftwareReset(void)	设定“软件复位”命令, 初始化蓝牙参数, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SoftwareResetKeep(void)	设定“软件复位”命令, 保留 RAM 的蓝牙参数, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)	设定 HCI 命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)	设定 BCI 命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetDeviceName(u8 leng,u8 *name)	设定“BM7701-00-1 的蓝牙设备名称”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetDeviceAddress(u8 *adr,u8 type)	设定“BM7701-00-1 的蓝牙地址”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)	设定“BM7701-00-1 的连接间隔”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)	设定“BM7701-00-1 的广播数据”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetScanResponseData(u8 leng,u8 *sdata)	设定“BM7701-00-1 的扫描响应数据”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)	设定“BM7701-00-1 的广播间隔”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_AdvertisingControl(ControlStatus ctrl)	设定“BM7701-00-1 的广播开关”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetTxPower(u8 pwr)	设定“BM7701-00-1 的输出功率”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetCrystalCload(u8 cc)	设定“BM7701-00-1 的外部 16MHz 晶振 Cload 值”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)	设定“BM7701-00-1 的 Feature”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetOperateMode(u8 omd,ControlStatus ctrl,u8 wuv,u8 wut)	设定“BM7701-00-1 的休眠模式和 MCU 外部唤醒信号”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
void *BC7701_SetWhiteList(ControlStatus erase,u8 *adr,u8 *mask)	设定“BM7701-00-1 的白名单地址”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。

void *BC7701_SetBaudRate(u8 br)	设定“BM7701-00-1 的波特率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
bool BC7701_DummyWakeup(void)	将 BM7701-00-1 从休眠模式中唤醒
bool BC7701_TransmitPackage(void *pbuf)	将命令封包放置于发送队列
bool BC7701_ReadTransmitEmpty(void)	获取数据发送队列的状态
bool BC7701_ReadReceiveEmpty(void)	获取数据接收队列的状态
void *BC7701_ReadReceivePackage(void)	读取 BM7701-00-1 的接收数据
void BC7701_WriteReceivePackage(void)	将接收数据索引写入接收队列
void BC7701_ReceiveParserPackage(void)	解析 BM7701-00-1 接收的数据

表 6. 模块 API 命令函数(HT32)

API 名称	void BC7701_InterfaceConfigure(u8 br)
API 作用	设定与 BM7701-00-1 通信的 MCU I/O 状态
输入参数	波特率: BAUD_RATE_9600 / BAUD_RATE_14400 / BAUD_RATE_19200 / BAUD_RATE_38400 / BAUD_RATE_57600 / BAUD_RATE_115200
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定与 BM7701-00-1 通信的 MCU I/O 状态。 设定 MCU 的 Reset control pin 为 AFIO 模式。 此函数会调用 BC7701_UARTConfigure(u8 br)，可以设定 MCU 的通信引脚为 UART 模式和波特率。

API 名称	void BC7701_UARTConfigure(u8 br)
API 作用	设定 MCU 的通信引脚为 UART 模式和波特率
输入参数	波特率: BAUD_RATE_9600 / BAUD_RATE_14400 / BAUD_RATE_19200 / BAUD_RATE_38400 / BAUD_RATE_57600 / BAUD_RATE_115200
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定 MCU 的通信引脚为 UART 模式和波特率

API 名称	void BC7701_RESET_SET(void)
API 作用	BM7701-00-1 的 RST_N 引脚拉高
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，BM7701-00-1 的 RST_N 引脚被拉高

API 名称	void BC7701_RESET_CLR(void)
API 作用	BM7701-00-1 的 RST_N 引脚拉低
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，BM7701-00-1 的 RST_N 引脚被拉低

API 名称	void BC7701_HardwareBaudRateDefault(u8 br)
API 作用	通过通信引脚状态初始化 BM7701-00-1 波特率(请参考表 2)
输入参数	波特率: BAUD_RATE_9600/ BAUD_RATE_115200
输出参数	—
程序详解	范例程序中，执行此 API 函数，依通信接口定义设定通信引脚状态初始化。 BM7701-00-1 波特率。 此 API 函数需在硬件 Reset 后马上执行才有效。 此 API 函数将通信引脚由 UART 模式改成 I/O 模式。执行完毕后必须执行 API 函数 BC7701_HardwareBaudRateRelease()，将通信引脚还原成 UART 模式。

API 名称	void BC7701_HardwareBaudRateRelease(void)
API 作用	完成 BM7701-00-1 波特率初始化设定
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，完成 BM7701-00-1 的波特率初始化设定。执行该函数的前提是 BC7701_HarddwreBaudRateDefault(u8 br)有在执行，然后等待 BM7701-00-1 上电复位 60ms 后再执行该函数

API 名称	void BC7701_HardwareReset(void)
API 作用	硬件复位，初始化蓝牙参数(拉低 BM7701-00-1 的 RST_N 引脚来实现)
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，BM7701-00-1 硬件复位，保存在 RAM 的蓝牙参数会被清掉。执行此函数后 BM7701-00-1 至少需要等待 60ms 才能再下 Command。该函数等效于*BC7701_SoftwareReset(void)

API 名称	void *BC7701_SoftwareReset(void)
API 作用	设定“软件复位”命令，初始化蓝牙参数，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数	—
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“软件复位”命令。命令执行成功后 BM7701-00-1 软件复位，保存在 RAM 的蓝牙参数会被清掉。等待 BM7701-00-1 回完 Event 后需要等待 60ms 才能再下 Command。 该命令等效于 BC7701_HardwareReset(void)。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SoftwareResetKeep(void)
API 作用	设定“软件复位”命令，保留 RAM 的蓝牙参数，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数	—
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“软件复位”命令(not Initial Parameter)。命令执行成功后 BM7701-00-1 软件复位，RAM 的蓝牙参数仍然保留，等待 BM7701-00-1 回完 Event 后需要等待 3ms 才能再下 Command。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)
API 作用	设定 HCI 命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	HCI 操作码(请参考“BLE_API”文档)
输入参数 2	数据长度
输入参数 3	存放数据的 Buffer
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定 HCI 命令。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)
API 作用	设定 BCI (BLE Controller Interface)命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	BCI 操作码(请参考“BLE_API”文档)
输入参数 2	BCI 标志(请参考“BLE_API”文档)
输入参数 3	数据长度
输入参数 4	存放数据的 Buffer
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定 BCI (BLE Controller Interface)命令。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。
API 名称	void *BC7701_SetDeviceName(u8 leng,u8 *name)
API 作用	设定“BM7701-00-1 的蓝牙设备名称”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	数据长度(≤31 byte)
输入参数 2	存放名称数据的 Buffer
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的蓝牙设备名称”命令。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SetDeviceAddress(u8 *adr,u8 type)
API 作用	设定“BM7701-00-1 的蓝牙地址”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	存放蓝牙地址的 Buffer(6 byte)
输入参数 2	蓝牙地址形态：STATIC_ADDRESS/RANDOM_ADDRESS
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的蓝牙地址”命令。 输入参数 2： 1. 若输入参数 2 等于 STATIC_ADDRESS，蓝牙地址形态设定为静态地址。 2. 若输入参数 2 等于 RANDOM_ADDRESS，蓝牙地址形态设定为随机地址。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)
API 作用	设定“BM7701-00-1 的连接间隔”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	操作码：BCI_CONN_INTV / BCI_CONN_INTV1 (请参考“BLE_API”文档)
输入参数 2	最小连接间隔，单位=1.25，有效范围=7.5ms~4s
输入参数 3	最大连接间隔，单位=1.25，有效范围=7.5ms~4s，仅当操作码=BCI_CONN_INTV1 有效
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的连接间隔”命令。 输入参数 1： 1. 若输入参数 1 等于 BCI_CONN_INTV，仅输入参数 2 有效，设定唯一的连接间隔。 2. 若输入参数 1 等于 BCI_CONN_INTV1，输入参数 2 和输入参数 3 都有效。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)
API 作用	设定“BM7701-00-1 的广播数据”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	广播模式：ADV_JOIN_NAME / ADV_UNJOIN_NAME (请参考“BLE_API”文档)
输入参数 2	广播数据长度(≤31 byte)
输入参数 3	存放广播数据的 Buffer
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的广播数据”命令 输入参数 1: - 若输入参数 1 等于 ADV_JOIN_NAME，广播数据会自动添加蓝牙设备名称，前提是广播数据≤31 byte。 - 若输入参数 1 等于 ADV_UNJOIN_NAME，广播数据不会加入蓝牙设备名称。 输出参数: - 若输出参数等于 NULL，API 函数配置失败。 - 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SetScanResponseData(u8 leng,u8 *sdata)
API 作用	设定“BM7701-00-1 的扫描响应数据”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	扫描响应数据长度(≤31 byte)
输入参数 2	存放扫描响应数据的 Buffer
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的扫描响应数据”命令。 输出参数: - 若输出参数等于 NULL，API 函数配置失败。 - 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)
API 作用	设定“BM7701-00-1 的广播间隔”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	最小广播间隔，单位=0.625，范围=20ms~10.24s
输入参数 2	最大广播间隔，单位=0.625，范围=20ms~10.24s
输入参数 3	广播通道
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的广播间隔”命令。 输入参数 3: - 若输入参数 3 等于 0B xxx1 (B[0]=1)，设定广播通道 37(2402MHz)。 - 若输入参数 3 等于 0B xx1x (B[1]=1)，设定广播通道 38(2426MHz)。 - 若输入参数 3 等于 0B x1xx (B[2]=1)，设定广播通道 39(2480MHz)。 输出参数: - 若输出参数等于 NULL，API 函数配置失败。 - 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_AdvertisingControl(ControlStatus ctrl)
API 作用	设定“BM7701-00-1 的广播开关”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数	开关：ENABLE/DISABLE
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的广播开关”命令。 输入参数: - 若输入参数等于 ENABLE，打开蓝牙广播。 - 若输入参数等于 DISABLE，关闭蓝牙广播。

	输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。
--	--

API 名称	void *BC7701_SetTxPower(u8 pwr)			
API 作用	设定“BM7701-00-1 的输出功率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。			
输入参数	0~15			
输出参数	命令封包指针			
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的输出功率”命令。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。			
输入参数	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

注：输出功率值仅做参考，具体数值需以实际取得模块数据为准。

API 名称	void *BC7701_SetCrystalCload(u8 cc)			
API 作用	设定“BM7701-00-1 的外部 16MHz 晶振 C _{load} 值”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。			
输入参数	0~15			
输出参数	命令封包指针			
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的外部 16MHz 晶振 C _{load} 值”命令。BM7701-00-1 建议采用 04 作为输入参数。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。			
输入参数	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

注：晶振值仅做参考，具体数值需以实际取得模块数据为准。

API 名称	void *BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)
API 作用	设定“BM7701-00-1 的 Feature”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	数据更新方式：FEATURE_DIR / FEATURE_SET / FEATURE_CLR
输入参数 2	模式：FEATURE_NO_APPED_NAME / FEATURE_PARAM_UPDATE / FEATURE_PARAM_ERASE / FEATURE_STATUS_EVENT / FEATURE_EXTERNAL32K / FEATURE_FORCE_CALIB / FEATURE_CK32K_OUTPUT (请参考“BLE_API”文档)
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的 Feature”命令(详情参考“BLE_API”文档)。 输入参数 1： 1. 若输入参数 1 等于 FEATURE_DIR，输入参数 2 会直接覆盖之前的数据。 2. 若输入参数 1 等于 FEATURE_SET，输入参数 2 会和之前的数据进行“OR”操作。 3. 若输入参数 1 等于 FEATURE_CLR，输入参数 2 会和之前的数据进行“AND”操作。 输入参数 2： 1. 若输入参数 2 等于 FEATURE_NO_APPED_NAME，广播数据不会加入蓝牙设备名称。 2. 若输入参数 2 等于 FEATURE_PARAM_UPDATE，BM7701-00-1 的 RAM 蓝牙参数写到 Flash。 3. 若输入参数 2 等于 FEATURE_PARAM_ERASE，删除 BM7701-00-1 的 Flash 的蓝牙参数，恢复默认值。 4. 若输入参数 2 等于 FEATURE_STATUS_EVENT，状态有变更时，BM7701-00-1 自动发送 Status Event。 5. 若输入参数 2 等于 FEATURE_EXTERNAL32K，启用外部 32.768kHz 晶振。 6. 若输入参数 2 等于 FEATURE_FORCE_CALIB，下次 Software reset 时强制校准。 7. 若输入参数 2 等于 FEATURE_CK32K_OUTPUT，BM7701-00-1 的 UART2_TX(PB6) 输出 32kHz 方波。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SetOperateMode(u8 omd,ControlStatus ctrl,u8 wuw,u8 wut)
API 作用	设定“BM7701-00-1 的休眠模式和 MCU 外部唤醒信号”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	模式：OP_NORMAL / OP_DEEPSLEEP / OP_POWERDOWN
输入参数 2	MCU 休眠：ENABLE / DISABLE
输入参数 3	MCU 外部唤醒信号宽度(0~8 byte)
输入参数 4	MCU 外部唤醒信号延迟时间(0~20ms)
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的休眠模式和 MCU 外部唤醒信号”命令。(详情参考“BLE_API”文档)。 输入参数 1: - 若输入参数 1 等于 OP_NORMAL，BM7701-00-1 进入 Normal 模式。默认情况下 BM7701-00-1 在 Normal 模式。 - 若输入参数 1 等于 OP_DEEPSLEEP，BM7701-00-1 进入 Deep Sleep 模式。可使用 API 函数 BC7701_DummyWakeup(void)将 BM7701-00-1 唤醒到 Normal 模式。 - 若输入参数 1 等于 OP_POWERDOWN，BM7701-00-1 进入 Power Down 模式。可使用 API 函数 BC7701_DummyWakeup(void)将 BM7701-00-1 唤醒到 Normal 模式。 输入参数 2: - 若输入参数 2 等于 ENABLE，说明 MCU 会进入休眠，设定 BM7701-00-1 发出唤醒信号来唤醒 MCU。 - 若输入参数 2 等于 DISABLE，说明 MCU 不会进入休眠，无需设定 BM7701-00-1 发出唤醒信号。 输出参数: - 若输出参数等于 NULL，API 函数配置失败。 - 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SetWhiteList(ControlStatus erase,u8 *adr,u8 *mask)
API 作用	设定“BM7701-00-1 的白名单地址”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	清除之前的白名单地址：ENABLE/DISABLE
输入参数 2	存放白名单地址的 buffer (6 byte)
输入参数 3	存放地址掩码的 buffer (6 byte)
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的白名单地址”命令。 Ex：白名单地址=0x112233445566，地址掩码=0xFFFFFFFFF00，只有蓝牙地址 0x112233445500~0x1122334455FF 可以连接。如果地址掩码全都写 0，所有蓝牙地址都可以连接。 输出参数: - 若输出参数等于 NULL，API 函数配置失败。 - 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void *BC7701_SetBaudRate(u8 br)
API 作用	设定“BM7701-00-1 的波特率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数	波特率：BAUD_RATE_9600 / BAUD_RATE_14400 / BAUD_RATE_19200 / BAUD_RATE_38400 / BAUD_RATE_57600 / BAUD_RATE_115200
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的波特率”命令。执行此命令且等待 BM7701-00-1 回完 EVENT 后，MCU 需等待约 1ms 后才能改变 UART 波特率及执行下一个通信传输。 输出参数: - 若输出参数等于 NULL，API 函数配置失败。 - 若输出参数不等于 NULL，API 函数配置成功。

API 名称	bool BC7701_DummyWakeup(void)
API 作用	将 BM7701-00-1 从休眠模式中唤醒
输入参数	—
输出参数	TRUE：发送 Dummy wakeup 成功 FALSE：发送 Dummy wakeup 失败
程序详解	范例程序中，执行此 API 函数，将 BM7701-00-1 从休眠模式中唤醒

API 名称	bool BC7701_TransmitPackage(void *pbuf)
API 作用	将 Command 封包放置于发送队列(封包格式请参考“Command/Event 格式”)
输入参数	存放封包数据的 Buffer
输出参数	TRUE：放置发送队列成功 FALSE：表示发送队列已满
程序详解	范例程序中，执行此 API 函数，MCU 将封包放置于发送队列中，发送队列中的封包会依序传输给 BM7701-00-1。

API 名称	bool BC7701_ReadTransmitEmpty(void)
API 作用	获取数据发送队列的状态
输入参数	—
输出参数	TRUE：队列是空的 FALSE：队列非空的
程序详解	范例程序中，执行此 API 函数，获取数据发送队列的状态

API 名称	bool BC7701_ReadReceiveEmpty(void)
API 作用	获取数据接收队列的状态
输入参数	—
输出参数	TRUE：队列是空的 FALSE：队列非空
程序详解	范例程序中，执行此 API 函数，获取数据接收队列的状态

API 名称	void *BC7701_ReadReceivePackage(void)
API 作用	读取 BM7701-00-1 的接收数据
输入参数	—
输出参数	封包指针
程序详解	范例程序中，执行此 API 函数，获取放置接收队列的数据，若输出为 NULL 指针表示没有数据。

API 名称	void BC7701_WriteReceivePackage(void)
API 作用	将接收数据索引写入接收队列
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，将接收数据索引写入接收队列

API 名称	void BC7701_ReceiveParserPackage(void)
API 作用	解析接收到的 BM7701-00-1 数据
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，解析接收到的 BM7701-00-1 数据，并将串行数据转成封包数据，并放置于接收队列中。

2. 以下表格整合 BM7701-00-1 模块 API 命令函数以及部分重要宏定义，请参考应用范例程序的 BC7701.c 和 BC7701.h 档案 (HT8)。

API 原型	功能说明
void BC7701_InterfaceConfigure(u8 br)	设定与 BM7701-00-1 通信的 MCU I/O 状态
void BC7701_UARTConfigure(u8 br)	设定 MCU 的通信引脚为 UART 模式和波特率
void BC7701_UARTWakeUpCtrl(u8 ctrl)	设定 MCU 的 UART 唤醒功能
void BC7701_HardwareBaudRateDefault(u8 br)	通过通信引脚初始化 BM7701-00-1 波特率
void BC7701_HardwareBaudRateRelease(void)	完成 BM7701-00-1 波特率初始化设定
void BC7701_SoftwareReset(void)	设定“BM7701-00-1 软件复位”命令，初始化蓝牙参数，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)	设定 HCI 命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)	设定 BCI 命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetDeviceName(u8 leng,u8 *name)	设定“BM7701-00-1 的蓝牙设备名称”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetDeviceAddress(u8 *adr,u8 type)	设定“BM7701-00-1 的蓝牙地址”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)	设定“BM7701-00-1 的连接间隔”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)	设定“BM7701-00-1 的广播数据”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetScanResponseData(u8 leng,u8 *sdata)	设定“BM7701-00-1 的扫描响应数据”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)	设定“BM7701-00-1 的广播间隔”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_AdvertisingControl(u8 ctrl)	设定“BM7701-00-1 的广播开关”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetTxPower(u8 pwr)	设定“BM7701-00-1 的输出功率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetCrystalCload(u8 cc)	设定“BM7701-00-1 的外部 16MHz 晶振 C _{load} 值”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)	设定“BM7701-00-1 的 Feature”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetOperateMode(u8 omd,u8 wue,u8 wuw,u8 wut)	设定“BM7701-00-1 的休眠模式和 MCU 外部唤醒信号”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetWhiteList(u8 erase,u8 *adr,u8 *mask)	设定“BM7701-00-1 的白名单地址”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_SetBaudRate(u8 br)	设定“BM7701-00-1 的波特率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
void BC7701_DummyWakeup(u8 leng)	将 BM7701-00-1 从休眠模式中唤醒
void BC7701_TransmitCmdPackage(u16 opcode,u8 flag)	给 BM7701-00-1 发送 BCI“READ”命令
void BC7701_TransmitPackage(u8 length)	给 BM7701-00-1 发送 TransmitData[] 封包
Void BC7701_TransmitPackageConst(tBCI_PACKAGE *pkg)	给 BM7701-00-1 发送 BCI 封包数据
u8 BC7701_PackageParserProcess(void)	解析 BM7701-00-1 接收的数据
#define BC7701_RESET_SET0	BM7701-00-1 的 RST_N 引脚拉高
#define BC7701_RESET_CLR0	BM7701-00-1 的 RST_N 引脚拉低

表 7. 模块 API 命令函数(HT8)

API 名称	void BC7701_InterfaceConfigure(u8 br)
API 作用	设定与 BM7701-00-1 通信的 MCU I/O 状态
输入参数	波特率: BAUD_RATE_4800 / BAUD_RATE_9600 / BAUD_RATE_19200 / BAUD_RATE_25000 / BAUD_RATE_50000 / BAUD_RATE_62500 / BAUD_RATE_115200
输出参数	—
程序详解	范例程序中, 执行此 API 函数, 设定与 BM7701-00-1 通信的 MCU I/O 状态。设定 MCU 的 Reset control pin 为 IO 模式, 通信引脚为 UART 模式。 此函数有调用 BC7701_UARTConfigure(u8 br), 设定 MCU 的 UART 参数。

API 名称	void BC7701_UARTConfigure(u8 br)
API 作用	设定 MCU 的 UART 参数
输入参数	波特率: BAUD_RATE_4800 / BAUD_RATE_9600 / BAUD_RATE_19200 / BAUD_RATE_25000 / BAUD_RATE_50000 / BAUD_RATE_62500 / BAUD_RATE_115200
输出参数	—
程序详解	范例程序中, 执行此 API 函数, 设定 MCU 的 UART 参数, 包括 MCU 的波特率

API 名称	void BC7701_UARTWakeUpCtrl(u8 ctrl)
API 作用	设定 MCU 的 UART 唤醒功能
输入参数	ENABLE / DISABLE
输出参数	—
程序详解	范例程序中, 执行此 API 函数, 设定 MCU 的 UART 唤醒功能。 输入参数 1: 1. 若输入参数 1 等于 ENABLE, 使能 MCU 的 UART 唤醒功能。 2. 若输入参数 1 等于 DISABLE, 除能 MCU 的 UART 唤醒功能。

API 名称	void BC7701_HardwareBaudRateDefault(u8 br)
API 作用	通过通信引脚状态初始化 BM7701-00-1 波特率(请参考表 2)
输入参数	波特率: BAUD_RATE_9600/ BAUD_RATE_115200
输出参数	—
程序详解	范例程序中, 执行此 API 函数, 依通信接口定义设定通信引脚状态初始化。BM7701-00-1 波特率。 此 API 函数需在硬件 Reset 后马上执行才有效。 此 API 函数将通信引脚由 UART 模式改成 I/O 模式。执行完毕后必须执行 API 函数 BC7701_HardwareBaudRateRelease(), 将通信引脚还原成 UART 模式。

API 名称	void BC7701_HardwareBaudRateRelease(void)
API 作用	完成 BM7701-00-1 波特率初始化设定
输入参数	—
输出参数	—
程序详解	范例程序中, 执行此 API 函数, 完成 BM7701-00-1 的波特率初始化设定。执行该函数的前提是 BC7701_HardwareBaudRateDefault(u8 br)有在执行, 然后等待 BM7701-00-1 上电复位 60ms 后再执行该函数

API 名称	void BC7701_SoftwareReset(void)
API 作用	设定“BM7701-00-1软件复位”命令，初始化蓝牙参数，需通过调用 BC7701_TransmitPackage函数进行发送。
输入参数	—
输出参数	—
程序详解	范例程序中，执行此 API 函数，发送“BM7701-00-1 软件复位”命令，保存在 RAM 的蓝牙参数会被清掉。等待 BM7701-00-1 回完 Event 后需要等待 60ms 才能再下 Command

API 名称	void BC7701_SendHCIPackage(u16 opcode,u8 len,u8 *pbuf)
API 作用	设定 HCI 命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	HCI 操作码(请参考“BLE_API”文档)
输入参数 2	数据长度
输入参数 3	存放数据的 Buffer
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定 HCI 命令并保存到 TransmitData[]

API 名称	void BC7701_SendBCIPackage(u16 opcode,u8 flag,u8 len,u8 *pbuf)
API 作用	设定 BCI (BLE Controller Interface)命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	BCI 操作码(请参考“BLE_API”文档)
输入参数 2	BCI 标志(请参考“BLE_API”文档)
输入参数 3	数据长度
输入参数 4	存放数据的 Buffer
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定 BCI 命令并保存到 TransmitData[]

API 名称	void BC7701_SetDeviceName(u8 leng,u8 *name)
API 作用	设定“BM7701-00-1 的蓝牙设备名称”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	数据长度(≤31 byte)
输入参数 2	存放名称数据的 Buffer
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的蓝牙设备名称”命令，命令数据保存到 TransmitData[]

API 名称	void BC7701_SetDeviceAddress(u8 *adr,u8 type)
API 作用	设定“BM7701-00-1 的蓝牙地址”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	存放蓝牙地址的 Buffer(6 byte)
输入参数 2	蓝牙地址形态：STATIC_ADDRESS/RANDOM_ADDRESS
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的蓝牙地址”命令，命令数据保存到 TransmitData[]。 输入参数 2： 1. 若输入参数 2 等于 STATIC_ADDRESS，蓝牙地址形态设定为静态地址。 2. 若输入参数 2 等于 RANDOM_ADDRESS，蓝牙地址形态设定为随机地址。

API 名称	void BC7701_ConnectIntervalModify(u16 opcode,u16 min,u16 max)
API 作用	设定“BM7701-00-1 的连接间隔”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	操作码：BCI_CONN_INTV / BCI_CONN_INTV1 (请参考“BLE_API”文档)
输入参数 2	最小连接间隔，单位=1.25，有效范围=7.5ms~4s
输入参数 3	最大连接间隔，单位=1.25，有效范围=7.5ms~4s，仅当操作码= BCI_CONN_INTV1 有效
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的连接间隔”命令，命令数据保存到 TransmitData[]。 输入参数 1： 1.若输入参数 1 等于 BCI_CONN_INTV，仅输入参数 2 有效，设定唯一的连接间隔。 2.若输入参数 1 等于 BCI_CONN_INTV1，输入参数 2 和输入参数 3 都有效。

API 名称	void BC7701_SetAdvertisingData(u8 mode,u8 leng,u8 *advdata)
API 作用	设定“BM7701-00-1 的广播数据”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	广播模式：UPDAE_AUTO_NAME / DEFAULT_NAME / DEFAULT_EMPTY / UPDAE_NOAO_NAME (请参考“BLE_API”文档)
输入参数 2	广播数据长度(≤31 byte)
输入参数 3	存放广播数据的 Buffer
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的广播数据”命令，命令数据保存到 TransmitData[]。 输入参数 1： 1. 若输入参数 1 等于 DEFAULT_NAME，广播数据采用默认值。 2. 若输入参数 1 等于 UPDAE_AUTO_NAME，广播数据会自动添加蓝牙设备名称，前提是广播数据≤31 byte。 3. 若输入参数 1 等于 DEFAULT_EMPTY，广播数据为空。 4. 若输入参数 1 等于 UPDAE_NOAO_NAME，广播数据不会加入蓝牙设备名称。

API 名称	void BC7701_SetScanResponseData(u8 leng,u8 *sdata)
API 作用	设定“BM7701-00-1 的扫描响应数据”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	扫描响应数据长度(≤31 byte)
输入参数 2	存放扫描响应数据的 Buffer
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的扫描响应数据”命令，命令数据保存到 TransmitData[]。 输出参数： 1. 若输出参数等于 NULL，API 函数配置失败。 2. 若输出参数不等于 NULL，API 函数配置成功。

API 名称	void BC7701_AdvertisingInterval(u16 min,u16 max,u8 chmap)
API 作用	设定“BM7701-00-1 的广播间隔”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	最小广播间隔，单位=0.625，范围=20ms~10.24s
输入参数 2	最大广播间隔，单位=0.625，范围=20ms~10.24s
输入参数 3	广播通道
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的广播间隔”命令，命令数据保存到 TransmitData[]。

	输入参数 3: 1. 若输入参数 3 等于 0B xxx1 (B[0]=1), 设定广播通道 37(2402MHz)。 2. 若输入参数 3 等于 0B xx1x (B[1]=1), 设定广播通道 38(2426MHz)。 3. 若输入参数 3 等于 0B x1xx (B[2]=1), 设定广播通道 39(2480MHz)。
--	--

API 名称	void BC7701_AdvertisingControl(ControlStatus ctrl)
API 作用	设定“BM7701-00-1 的广播开关”命令, 需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数	开关: ENABLE/DISABLE
输出参数	—
程序详解	范例程序中, 执行此 API 函数, 设定“BM7701-00-1 的广播开关”命令, 命令数据保存到 TransmitData[]。 输入参数: 1. 若输入参数等于 ENABLE, 打开蓝牙广播。 2. 若输入参数等于 DISABLE, 关闭蓝牙广播。

API 名称	void BC7701_SetTxPower(u8 pwr)			
API 作用	设定“BM7701-00-1 的输出功率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。			
输入参数	0~15			
输出参数	—			
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的输出功率”命令，命令数据保存到 TransmitData[]。			
输入参数	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

注: 输出功率值仅做参考, 具体数值需以实际取得模块数据为准。

API 名称	void BC7701_SetCrystalCload(u8 cc)			
API 作用	设定“BM7701-00-1 的外部 16MHz 晶振 C _{load} 值”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。			
输入参数	0~15			
输出参数	—			
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的外部 16MHz 晶振 C _{load} 值”命令，命令数据保存到 TransmitData[]。BM7701-00-1 建议采用 04 作为输入参数。			
输入参数	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

注: 晶振值仅做参考, 具体数值需以实际取得模块数据为准。

API 名称	void BC7701_SetupFeatureFlag(u8 md,FeatureFlag sff)
API 作用	设定“BM7701-00-1 的 Feature”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	数据更新方式：FEATURE_DIR / FEATURE_SET / FEATURE_CLR
输入参数 2	模式：FEATURE_NO_APPED_NAME / FEATURE_PARAM_UPDATE / FEATURE_PARAM_ERASE / FEATURE_STATUS_EVENT / FEATURE_EXTERNAL32K / FEATURE_FORCE_CALIB / FEATURE_CK32K_OUTPUT (请参考“BLE_API”文档)
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的 Feature”命令(详情参考“BLE_API”文件)，命令数据保存到 TransmitData[]。 输入参数 1: 1. 若输入参数 1 等于 FEATURE_DIR，输入参数 2 会直接覆盖之前的数据。 2. 若输入参数 1 等于 FEATURE_SET，输入参数 2 会和之前的数据进行“OR”操作。 3. 若输入参数 1 等于 FEATURE_CLR，输入参数 2 会和之前的数据进行“AND”操作。 输入参数 2: 1. 若输入参数 2 等于 FEATURE_NO_APPED_NAME，广播数据不会加入蓝牙设备名称。 2. 若输入参数 2 等于 FEATURE_PARAM_UPDATE，BM7701-00-1 的 RAM 蓝牙参数写到 Flash。 3. 若输入参数 2 等于 FEATURE_PARAM_ERASE，删除 BM7701-00-1 的 Flash 的蓝牙参数，恢复默认值。 4. 若输入参数 2 等于 FEATURE_STATUS_EVENT，状态有变更时，BM7701-00-1 自动发送 Status Event。 5. 若输入参数 2 等于 FEATURE_EXTERNAL32K，启用外部 32.768kHz 晶振。 6. 若输入参数 2 等于 FEATURE_FORCE_CALIB，下次 Software reset 时强制校准。 7. 若输入参数 2 等于 FEATURE_CK32K_OUTPUT，BM7701-00-1 的 UART2_TX(PB6) 输出 32kHz 方波。

API 名称	void BC7701_SetOperateMode(u8 omd,u8 wue,u8 wuw,u8 wut)
API 作用	设定“BM7701-00-1 的休眠模式和 MCU 外部唤醒信号”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	模式：OP_NORMAL / OP_DEEPSLEEP / OP_POWERDOWN
输入参数 2	MCU 休眠：ENABLE / DISABLE
输入参数 3	MCU 外部唤醒信号宽度(0~8 byte)
输入参数 4	MCU 外部唤醒信号延迟时间(0~20ms)
输出参数	命令封包指针
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的休眠模式和 MCU 外部唤醒信号”命令，命令数据保存到 TransmitData[]。(详情参考“BLE_API”文档)。 输入参数 1: 1. 若输入参数 1 等于 OP_NORMAL，BM7701-00-1 进入 Normal 模式。默认情况下 BM7701-00-1 在 Normal 模式。 2. 若输入参数 1 等于 OP_DEEPSLEEP，BM7701-00-1 进入 Deep Sleep 模式。可使用 API 函数 BC7701_DummyWakeup(u8 leng)将 BM7701-00-1 唤醒到 Normal 模式。 3. 若输入参数 1 等于 OP_POWERDOWN，BM7701-00-1 进入 Power Down 模式。可使用 API 函数 BC7701_DummyWakeup(u8 leng)将 BM7701-00-1 唤醒到 Normal 模式。 输入参数 2: 1. 若输入参数 2 等于 ENABLE，说明 MCU 会进入休眠，设定 BM7701-00-1 发出唤醒信号来唤醒 MCU。 2. 若输入参数 2 等于 DISABLE，说明 MCU 不会进入休眠，无需设定 BM7701-00-1 发出唤醒信号。

API 名称	void BC7701_SetWhiteList(ControlStatus erase,u8 *adr,u8 *mask)
API 作用	设定“BM7701-00-1 的白名单地址”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数 1	清除之前的白名单地址：ENABLE/DISABLE
输入参数 2	存放白名单地址的 buffer (6 byte)
输入参数 3	存放地址掩码的 buffer (6 byte)
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的白名单地址”命令，命令数据保存到 TransmitData[]。 Ex：白名单地址=0x112233445566，地址掩码=0xFFFFFFFF00，只有蓝牙地址 0x112233445500~0x1122334455FF 可以连接。如果地址掩码全都写 0，所有蓝牙地址都可以连接。

API 名称	void BC7701_SetBaudRate(u8 br)
API 作用	设定“BM7701-00-1 的波特率”命令，需通过调用 BC7701_TransmitPackage 函数进行发送。
输入参数	波特率：BAUD_RATE_4800 / BAUD_RATE_9600 / BAUD_RATE_19200 / BAUD_RATE_25000 / BAUD_RATE_50000 / BAUD_RATE_62500 / BAUD_RATE_115200
输出参数	—
程序详解	范例程序中，执行此 API 函数，设定“BM7701-00-1 的波特率”命令，命令数据保存到 TransmitData[]。收到 EVENT 后 MCU 需等待约 1ms 后才能改变 UART 波特率及执行下一个通信传输。

API 名称	void BC7701_DummyWakeup(u8 leng)
API 作用	将 BM7701-00-1 从休眠模式中唤醒
输入参数	Dummy wakeup 信号 (0x00) 占用 byte
输出参数	—
程序详解	范例程序中，执行此 API 函数，将 BM7701-00-1 从休眠模式中唤醒。建议输入参数设置为 1。

API 名称	void BC7701_TransmitCmdPackage(u16 opcode,u8 flag)
API 作用	给 BM7701-00-1 发送 BCI “READ”命令
输入参数 1	BCI 操作码(请参考“BLE_API”文档)
输入参数 2	BCI 标志(请参考“BLE_API”文档)
输出参数	—
程序详解	范例程序中，执行此 API 函数，MCU 给 BM7701-00-1 发送 BCI “READ”命令

API 名称	void BC7701_TransmitPackage(u8 length)
API 作用	给 BM7701-00-1 发送 TransmitData[] 封包
输入参数	发送的封包长度
输出参数	—
程序详解	范例程序中，执行此 API 函数，MCU 将 TransmitData[] 的数据发送给 BM7701-00-1。如果检测到 TransmitData[0] 为 HCI 或 BCI 类型，忽略输入参数，发送的封包长度等于 TransmitData[1]+2 (HCI 或 BCI 命令的封包长度)。

API 名称	void BC7701_TransmitPackageConst(tBCI_PACKAGE *pkg)
API 作用	给 BM7701-00-1 发送 BCI 封包数据
输入参数	存放 BCI 封包数据的 Buffer
输出参数	—

程序详解	范例程序中，执行此 API 函数，MCU 给 BM7701-00-1 发送封包数据
------	---

API 名称	u8 BC7701_PackageParserProcess(void)
API 作用	解析 BM7701-00-1 接收到的 EVENT
输入参数	—
输出参数	TRUE / FALSE
程序详解	范例程序中，执行此 API 函数，解析 BM7701-00-1 接收到的 EVENT 输出参数： 1. 若输出参数等于 TRUE，说明接收的 EVENT 有效 2. 若输出参数等于 FALSE，说明接收的 EVENT 无效

API 名称	#define BC7701_RESET_SET0
API 作用	BM7701-00-1 的 RST_N 引脚拉高
程序详解	范例程序中，使用该宏定义，BM7701-00-1 的 RST_N 引脚被拉高

API 名称	#define BC7701_RESET_CLR0
API 作用	BM7701-00-1 的 RST_N 引脚拉低
程序详解	范例程序中，使用该宏定义，BM7701-00-1 的 RST_N 引脚被拉低

蓝牙参数修改(HT32)

- 1、开启档案 bleprocess.h，然后点击 Configuration Wizard 进入“蓝牙参数”设定页面，如下图
- 12。接下来将详细介绍每个参数设定选项。

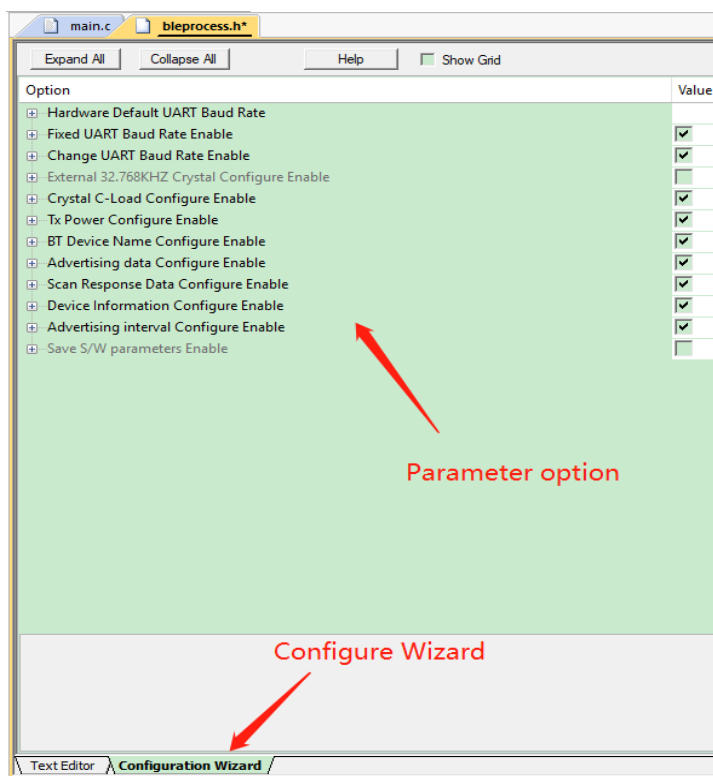


图 12

- Hardware Default UART Baud Rate: 设定上电复位 60ms 后 BM7701-00-1 的初始波特率，只有两个波特率 9600bps 和 115200bps 可以选择，如下图 13。该方法属于硬件方式修

改 BM7701-00-1 波特率(参考表 2)。

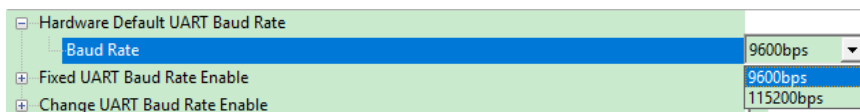


图 13

- Fixed UART Baud Rate Enable: 打 ✓ 使能“定义 MCU 初始波特率”，点击附加项 Baud Rate 选择波特率，如下图 14。

- 除能：自动设定 MCU 的初始波特率等于 BM7701-00-1 的初始波特率(9600bps 和 115200bps 两者选一)。
- 使能：用户需要手动选择 MCU 初始波特率。

Ex：当 BM7701-00-1 有保存新的波特率参数到 Flash (使能 Save S/W parameters Enable 选项，详情请参考“参数设定 12 点”)，例如波特率 57600bps 保存到 Flash，之后 BM7701-00-1 每次上电复位 60ms 后的初始波特率都等于 57600bps。然后用户必须使能该选项，并选择 MCU 初始波特率等于 57600bps，否则 MCU 和 BM7701-00-1 无法正常通信。

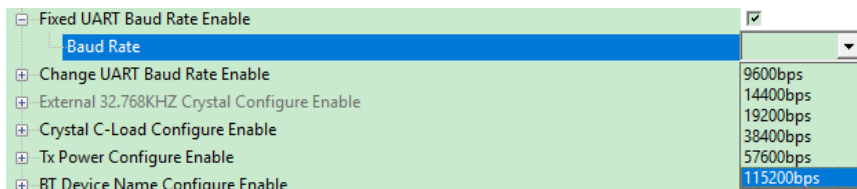


图 14

- Change UART Baud Rate Enable: 打 ✓ 使能“更改波特率”，点击附加项 Baud Rate 选择波特率，如下图 15。该选项同时设定 BM7701-00-1 和 MCU 的波特率，仅在 MCU 和 BM7701-00-1 的初始波特率都设定完成后且相同才会执行(Ex：MCU 向 BM7701-00-1 发送 Change UART Baud Rate Command，然后 MCU 再设定通信引脚的波特率与 BM7701-00-1 一致)。

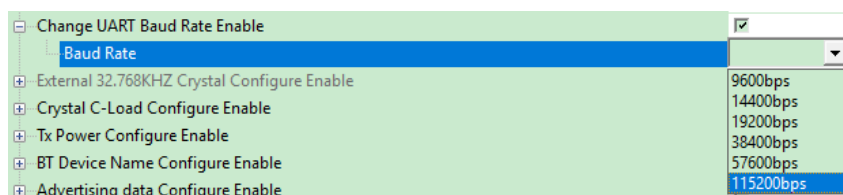


图 15

- External 32.768kHz Crystal Configure Enable: 打 ✓ 使能“外部 32.768kHz 晶振使能”，如下图 16。部分模块有外部 32.768kHz 晶振可以使能该选项，应用范例程序默认除能。注意没带外部 32.768kHz 晶振的 BM7701-00-1 模块使能该选项会造成工作失败。

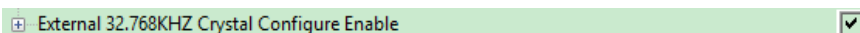


图 16

- Crystal C-Load Configure Enable: 打 ✓ 使能“更改外部 16MHz 晶振 C_{load} 值”，点击附加项 Crystal C-Load 输入数值，取值范围 0~15，如下图 17。BM7701-00-1 模块推荐使用数值 4。



图 17

输入参数	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

注：晶振值仅做参考，具体数值需以实际取得模块数据为准。

表 8

- Tx Power Configure Enable: 打 ✓ 使能“更改输出功率值”，点击附加项 Tx Power 输入数值，取值范围 0~15，如下图 18。用户可以根据传输距离或功耗灵活调整。



图 18

输入参数	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

注：输出功率值仅做参考，具体数值需以实际得到模块数据为准。

表 9

- BT Device Name Configure Enable: 打 ✓ 使能“更改蓝牙名称”，如下图 19。成功更改后的蓝牙名称为“BM7701”。用户也可以在 bleprocess.c 页面自定义蓝牙名称，找到下图 20 红框位置更改，最大长度为 31 Byte。

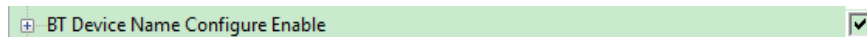


图 19

```

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

```

图 20

- Advertising data Configure Enable: 打 ✓ 使能“更改广播数据”，如下图 21。用户也可以在 bleprocess.c 页面自定义广播数据，找到下图 22 红框位置更改，最大长度为 31 Byte。

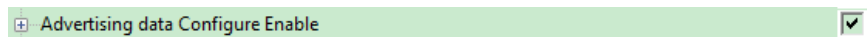


图 21

```

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

```

图 22

- Scan Response Data Configure Enable：打 ✓ 使能“更改扫描响应数据”，如下图 23。用户也可以在 bleprocess.c 页面自定义扫描响应数据，找到下图 24 红框位置更改，最大长度为 31 Byte。

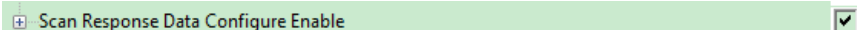


图 23

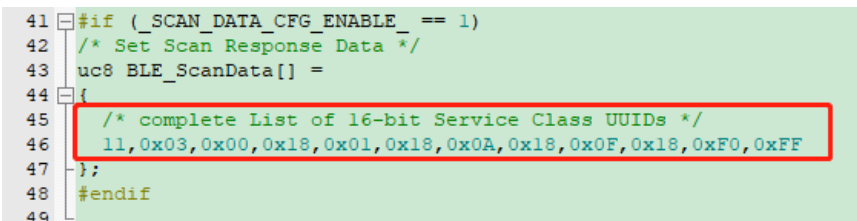


图 24

- Advertising interval Configure Enable：打 ✓ 使能“更改广播间隔”，点击附加项输入数值，如下图 25。注意 Min Interval ≤ Max Interval。



图 25

- Device Information Configure Enable：打 ✓ 使能“更改设备信息”，如下图 26。用户也可以在 bleprocess.c 页面自定义设备信息。找到下图 27 红框位置更改。

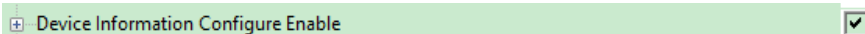


图 26

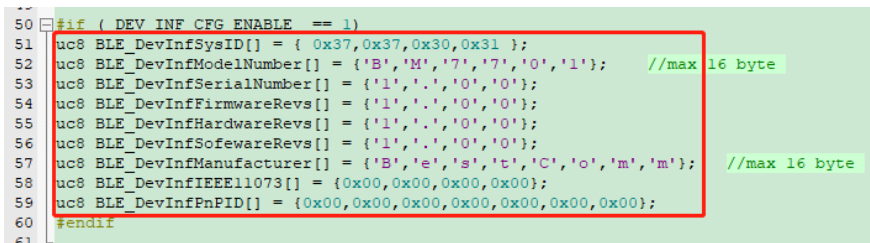


图 27

- Save S/W parameters Enable：打 ✓ 使能“保存参数”。“参数设定 3~11 点”的蓝牙参数保存到 BM7701-00-1 的 Flash，如下图 28。前提是对应的设定选项有使能（通过 Command 设定 BM7701-00-1 的蓝牙参数会暂时保存到 RAM，使能选项后 RAM 的蓝牙参数会保存到 Flash，重新上电复位后 BM7701-00-1 就会自动使用 Flash 的蓝牙参数）。

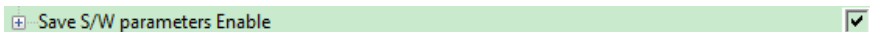


图 28

2、开启档案 main.c，然后点击 Configuration Wizard 可以设定“数据传输”和“MCU 休眠”的相关的参数。

- Change Connect Interval Enable：打 ☒ 使能“修改 BM7701-00-1 连接间隔”，如下图 29。
点击附加项输入连接间隔数值。

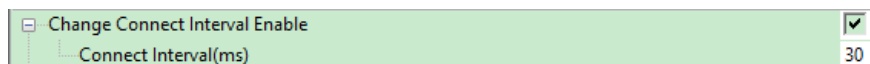


图 29

- MCU sleep mode Enable / disable：打 ☒ 使能“MCU 进入休眠”。点击附加项输入时间。
如下图 30，使能该选项后，BM7701-00-1 在 500ms 内没有与连接的设备(手机)传输数据，MCU 会主动进入休眠模式。



图 30

- WakeUp Signal Configure：点击附加项选择“唤醒信号宽度(byte)”和“唤醒信号延长时
间(ms)”。如下图 31，当 MCU 需要从休眠模式被唤醒，可设定 BM7701-00-1 发送 2
byte 0x00 唤醒 MCU，然后等待 8ms 把接收到的数据 Event 传给 MCU。唤醒信号宽度
和延长时间取决于 MCU 的休眠模式设定和唤醒稳定时间，用户可以根据实际情况灵
活调整。



图 31

- 为了让用户能够清楚波特率选项的区别以及 Flash 保存的影响，举以下例子说明在选
择不同的波特率要如何进行操作（如无特殊说明，波特率都是指 MCU 和 BM7701-00-
1 的波特率）。

例 1：只选择 9600bps 或 115200bps 波特率。操作步骤如下表 10。

- 上电复位 60ms 后波特率都为 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：任意波特率，例如选择 57600bps，波特率不保存到 BM7701-00-1 的 Flash 的操
作步骤如下表 11。

- 上电复位 60ms 后的初始波特率为 9600bps 或 115200bps。
- 待执行到 Change UART Baud Rate Command，波特率更改为 57600bps。
- 每次重新上电复位都会顺序执行以上动作。

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：任意波特率，例如选择 57600bps，波特率保存到 BM7701-00-1 的 Flash。该方式必须执行两次烧录动作。操作步骤如下表 12。

- 第一次烧录
 - 烧录程序后，第一次上电复位 60ms 后的初始波特率为 9600bps 或 115200bps。
 - 待执行 Change UART Baud Rate Command，波特率为 57600bps。
 - 保存波特率到 BM7701-00-1 的 Flash。
 - 接下来每次上电复位，BM7701-00-1 的波特率都一直为 57600bps bps，而 MCU 的波特率为 9600bps 或 115200bps，无法正常通信。
 - 需要执行第二次烧录动作。
- 第二次烧录
 - MCU 和 BM7701-00-1 可正常通信 (使能 Fixed UART Baud Rate Enable，并更改 MCU 的初始波特率为 57600bps)。

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

表 12

蓝牙参数修改(HT8)

1、开启档案 bleprocess.h，找到下图 32 的位置进行“蓝牙参数”设定。接下来将详细介绍每个参数设定选项。

```

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 _TX_PWR_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 _ADV_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 _ADV_INTU_CFG_ENABLE_ 1
27 /*----- Min Interval(ms)<10~40000 -----*/
28 #define _ADV_INTU_MIN_VALUE_ (1000)
29 /*----- Max Interval(ms)<10~40000 -----*/
30 #define _ADV_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 8C7701 to Sleep mode Enable/Disable -----*/
35 #define _PWRON_FINISH_SLEEP_ 0

```

图 32

- Change UART Baud Rate
 - #define _CHG_BAUD_RATE_ENABLE_ 1：使能“更改波特率”
 - #define _CHG_BAUD_RATE_ENABLE_ 0：除能“更改波特率”
 - #define _CHANGE_BAUDRATE_ BAUD_RATE_115200：波特率设定为 115200bps。用户也可以根据需求定义其它波特率

```

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

```

图 33

● Change Crystal C-Load

- #define _CLOAD_CFG_ENABLE_ 1: 使能“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} 值设定为 04

```

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

```

图 34

输入参数	0	5	10	15
Crystal Frequency(MHz)	15.99985	16.00004	16.00031	16.00074

注：晶振值仅作参考，具体数值需以实际取得模块数据为准。

表 13

● Change Tx Power

- #define _TXPWR_CFG_ENABLE_ 1: 使能“更改输出功率”
- #define _TXPWR_CFG_ENABLE_ 0: 除能“更改输出功率”
- #define _TX_POWER_VALUE_ 10: 输出功率值设定为 10

```

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

```

图 35

输入参数	0	5	10	15
Tx Power(dbm)	-32.5	-11.5	0.5	3.5

注：输出功率值仅作参考，具体数值需以实际得到模块数据为准。

表 14

● Configure BT Device Name

- #define _BDNAME_CFG_ENABLE_ 1: 使能“更改蓝牙设备名称”
- #define _BDNAME_CFG_ENABLE_ 0: 除能“更改蓝牙设备名称”

```

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

```

图 36

用户也可以打开 bleprocess.c 页面找到下图 37 的红框部分来自定义蓝牙设备名称，注意长度≤ 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_DEV_NAME,              //opcode
46      {'B','M','7','7','0','1'} //parameter
47  };
48  #endif

```

图 37

● Configure Advertising data

- #define_ADV_DATA_CFG_ENABLE_ 1 : 使能 “更改蓝牙数据”
- #define_ADV_DATA_CFG_ENABLE_ 0 : 除能 “更改蓝牙数据”

```

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

```

图 38

用户也可以打开 bleprocess.c 页面找到下图 39 的红框部分来自定义蓝牙广播数据，注意长度≤31 byte。

```

50  /*--- set Advertising Data ---*/
51  #if (_ADV_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_ADV_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

```

图 39

● Configure Scan Response Data

- #define_SCAN_DATA_CFG_ENABLE_ 1: 使能 “更改扫描响应数据”
- #define_SCAN_DATA_CFG_ENABLE_ 0: 除能 “更改扫描响应数据”

```

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

```

图 40

用户也可以打开 bleprocess.c 页面找到下图 41 的红框部分来自定义扫描响应数据，注意长度≤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

```

图 41

● Configure Advertising interval Enable/Disable

- #define _ADV_INTV_CFG_ENABLE_ 1: 使能“更改广播间隔”
- #define _ADV_INTV_CFG_ENABLE_ 0: 除能“更改广播间隔”
- #define _ADV_INTV_MIN_VALUE_ (1000): 最小广播间隔设定为 1000ms
- #define _ADV_INTV_MAX_VALUE_ (1000): 最大广播间隔设定为 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

图 42

● Configure Feature setup

- #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT+FEATURE_EXTERNAL32K): 开启“状态有变更时 BM7701-00-1 自动发送 Status Event + 外部 32.768KHz 晶振使能。”
- #define FEATURE_STATUS_SETUP (FEATURE_STATUS_EVENT): 开启“状态有变更时 BM7701-00-1 自动发送 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)

```

图 43

● Power On finish BC7701 to Sleep mode Enable/Disable

- #define _PWRON_FINISH_SLEEP_ 0: 参数配置成功后不进入休眠。
- #define _PWRON_FINISH_SLEEP_ 1: 参数配置成功后进入休眠。

```

34  /*----- Power On finish BC7701 to Sleep mode Enable/Disable -----*/
35  #define _PWRON_FINISH_SLEEP_ 0

```

图 44

2、开启档案 main.c，找到下图的位置设定“数据传输”和“MCU 休眠”的相关的参数。

● Change Connect Interval

- #define _CHG_CONNECT_INTV_ 1: 使能“更改连接间隔”
- #define _CHG_CONNECT_INTV_ 0: 除能“更改连接间隔”
- #define _CNNT_INTV_MIN_VALUE_ (30): 最小连接间隔设定为 30ms

➤ #define _CNNT_INTV_MAX_VALUE_ (30): 最大连接间隔设定为 30ms

```

64 #define _CHG_CONNECT_INTU_ 1
65 /*----- Min Interval(ms)<10-40000> -----*/
66 #define _CNNT_INTU_MIN_VALUE_ (30)
67 /*----- Max Interval(ms)<10-40000> -----*/
68 #define _CNNT_INTU_MAX_VALUE_ (30)

```

图 45

● Configure MCU sleep mode

➤ #define _MCU_SLEEP_ENABLE_ 1: 使能“MCU 休眠”

➤ #define _MCU_SLEEP_ENABLE_ 0: 除能“MCU 休眠”

➤ #define _SLEEP_DELAY_TIMER_ (500/2): 如果 250ms 内没有与连接的设备(手机)传输数据, MCU 会进入休眠模式。

➤ #define _MASTER_WUW_VALUE_ (1)

#define _MASTER_WUT_VALUE_ (4)

● 如下图 46, 当 MCU 需要从休眠模式被唤醒, 可设定 BM7701-00-1 发送 1 byte 0x00 唤醒 MCU, 然后等待 4ms 把接收到的数据 Event 传给 MCU。唤醒信号宽度和延长时间取决于 MCU 的休眠模式设定和唤醒稳定时间, 用户可以根据实际情况灵活调整。

```

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)

```

图 46

结论

本文介绍 BM7701-00-1 模块比较常用的 API 函数命令, 结合应用程序指导读者使用 HT8 和 HT32 MCU 配置 BM7701-00-1 模块的蓝牙参数, 并与手机 APP 互相传输数据。通过该范例展示模块的操作方法以及出色的双向数据传输的能力, 帮助读者更快地运用到相关电子产品, 例如家用电器和医疗产品等等。

参考资料

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

如需进一步了解, 敬请浏览 Holtek 官方网站 www.holtek.com.cn。

版本及修订说明

日期	作者	发行	修订说明
2022.06.23	阮义展	V1.00	第一版

免责声明

本网页所载的所有数据、商标、图片、链接及其他数据等 (以下简称「数据」), 只供参考之用, 合泰半导体 (中国) 有限公司及其关联企业 (以下简称「本公司」) 将会随时更改数据, 并由本公司决定而不作另行通知。虽然本公司已尽力确保本网页的数据准确性, 但本公司并不保证该等数据均为准确无误。本公司不会对任何错误或遗漏承担责任。

本公司不会对任何人士使用本网页而引致任何损害 (包括但不限于计算机病毒、系统故障、数据损失) 承担任何赔偿。本网页可能会连结至其他机构所提供的网页, 但这些网页并不是由本公司所控制。本公司不对这些网页所显示的内容作出任何保证或承担任何责任。

责任限制

在任何情况下, 本公司并不须就任何人由于直接或间接进入或使用本网站, 并就此内容上或任何产品、信息或服务, 而招致的任何损失或损害负任何责任。

管辖法律

以本公司所在地法律为准据法, 并以本公司所在地法院为第一审管辖法院。

免责声明更新

本公司保留随时更新本免责声明的权利, 任何更改于本网站发布时, 立即生效。