



藍牙 5.2 BLE 模組

BM7701-00-1

Arduino Library V1.0.2 說明

版本 : V1.10 日期 : 2024-05-10

www.bestmodulescorp.com

目錄

簡介	3
Arduino Lib 函式	3
Arduino Lib 下載及安裝	9
Arduino 範例	10
範例：writeAndRead	10
附錄	15

簡介

BM7701-00-1 是倍創推出的藍牙 5.2 BLE 模組，使用 UART 通訊方式。本文檔對 BM7701-00-1 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例使用 BMC77M001 模組，演示了資料透傳功能。

適用型號：

型號	說明
BM7701-00-1	藍牙 5.2 BLE 模組
BMC77M001	板載 BM7701-00-1 模組

Arduino Lib 函式

Arduino Lib 名稱：BM7701-00-1		Lib 版本：V1.0.2
構造函式 & 初始化		
1	BM7701_00_1(HardwareSerial*&theSerial=&Serial)	
	描述	構造函式，使用硬體 HW Serial 介面
	參數	*theSerial：選擇硬體 HW Serial 介面（預設 Serial 介面）
	返回值	—
	備註	—
2	BM7701_00_1(uint16_t rxPin, uint16_t txPin)	
	描述	構造函式，使用軟體串行介面
	參數	rxPin：RX 腳位，連接 BM7701-00-1 的 UARTn_TX 腳位 (n=1/2) 或 BMC77M001 的 TX 腳位 txPin：TX 腳位，連接 BM7701-00-1 的 UARTn_RX 腳位 (n=1/2) 或 BMC77M001 的 RX 腳位
	返回值	—
	備註	—
3	void begin(uint32_t baud=115200)	
	描述	初始化模組，設定鮑率
	參數	baud：鮑率，範圍 9600/14400/19200/38400/57600/115200/125000
	返回值	void
	備註	鮑率預設 115200
功能函式		
4	bool setAdvCtrl(bool ctrl)	
	描述	設定廣播開啟 / 關閉
	參數	ctrl：廣播狀態 true：開啟 false：關閉
	返回值	執行情況： true：成功 false：失敗
	備註	開啟廣播後需要等待 600~800ms 校準時間

	bool disconnect()	
	描述	斷開藍牙設備連接
	參數	—
5	執行情況： 返回值 true : 成功 false : 失敗	
	備註	斷開後將等待 5ms 的緩衝時間
	bool writeData(uint8_t dataBuf[], uint8_t length)	
	描述	發送資料給連接藍牙模組的設備
	參數	dataBuf[] : 需要發送的資料 length : 需要發送的資料的長度
6	執行情況： 返回值 true : 成功 false : 失敗	
	備註	data[] 的位元組長度需要 \leq 57bytes
	bool readData(uint8_t receiveBuff[], uint8_t &length)	
	描述	讀取設備發送的資料
	參數	receiveBuff[] : 讀取到的資料 &length : 讀取到的資料的長度
7	執行情況： 返回值 true : 成功 false : 失敗	
	備註	receiveBuff[] 的長度建議設定 \geq 57bytes
	設定函式	
	bool setAdvIntv(uint16_t min, uint16_t max, uint8_t advMap)	
	描述	設定廣播參數
8	參數	min : 廣播間隔最短時間參數，範圍：0x0020~0x4000 max : 廣播間隔最大時間參數，範圍：0x0020~0x4000 advMap : 廣播通道 (可同時開啟多個通道) bit0=1 : channel37 (2402MHz) 開關控制 bit0=1 : 開啟 bit0=0 : 關閉 bit1=1 : channel38 (2426MHz) 開關控制 bit1=1 : 開啟 bit1=0 : 關閉 bit2=1 : channel39 (2480MHz) 開關控制 bit2=1 : 開啟 bit2=0 : 關閉
	返回值	執行情況： true : 成功 false : 失敗
	備註	廣播間隔最短時間計算公式 : $t_{Adv-min} = min \times 0.625ms$ 廣播間隔最大時間計算公式 : $t_{Adv-max} = max \times 0.625ms$

	bool setAdvData(uint8_t appendName, uint8_t length, uint8_t advData[])	
9	描述	設定廣播資料
	參數	appendName： 0x00 (APPEND_NAME)：附帶資料 0x10 (NO_APPEND_NAME)：不附帶資料 length：advData[] 的位元組長度 advData[]：資料 · advData[] 的位元組長度需要 ≤ 31bytes appendName=0x00 時，包含 advData + 其他屬性資訊 appendName=0x10 時，僅包含 advData
	返回值	執行情況： true：成功 false：失敗
	備註	正常使用下無需修改 若要修改 advData 和其他資訊詳細參考 BLE_API 規格書 4.9 API_AdvData
	bool setScanData(uint8_t length, uint8_t scanData[])	
10	描述	當廣播資料被 app 掃描到時，模組要回應給手機 APP 的資料
	參數	length：scanData[] 的位元組長度 scanData[]：格式為資料位元組長度 + 資料類型 + 資料，需要 ≤ 31byte
	返回值	執行情況： true：成功 false：失敗
	備註	正常使用下無需修改 資料類型若要修改參考 BLE_API 規格書 4.10 API_ScanData
	bool setConnIntv(uint16_t interval)	
12	描述	連接成功後，設定藍牙模組和手機 APP 的週期性通訊間隔時間
	參數	interval：通訊間隔時間參數，範圍：0x0006~0xc80
	返回值	執行情況： true：成功 false：失敗
	備註	週期性通訊間隔時間 $t_{inv}=interval \times 1.25ms$ ，與函式 setConnIntv1 的設定相同
	bool setConnIntv(uint16_t minIntv, uint16_t maxIntv, uint16_t latency, uint16_t timeout)	
13	描述	連接成功後，設定藍牙模組和手機 APP 的週期性通訊間隔時間最大值和最小值，設備延遲發送時間，連接逾時時間
	參數	minIntv：最小間隔時間參數，範圍：0x0006~0xc80 maxIntv：最大間隔時間參數，範圍：0x0006~0xc80 latency：設備延遲發送的時間參數，範圍：0x0000~0x01f3 timeout：連接逾時時間參數，範圍：0x000a~0xc80
	返回值	執行情況： true：成功 false：失敗
	備註	最小間隔時間 $t_{inv-min}=minIntv \times 1.25ms$ 最大間隔時間 $t_{inv-max}=maxIntv \times 1.25ms$ (最終連接間隔時間 t_{inv} 處於最大和最小區間) 設備延遲發送的時間 $t_{del}=latency \times t_{inv}$ 連接逾時時間 $t_{out}=timeout \times 10ms$

14	<code>bool setName(uint8_t length, uint8_t name[])</code>	
	描述	設定藍牙模組廣播時被檢測到顯示的名稱
	參數	length : 藍牙名稱位元組數 name[] : 藍牙名稱
	返回值	執行情況： true : 成功 false : 失敗
	備註	name[] 的位元組長度需要 ≤ 31bytes
15	<code>bool setAddress(uint8_t address[])</code>	
	描述	設定藍牙 mac 位址
	參數	address[] : 藍牙 mac 位址 (6byte)
	返回值	執行情況： true : 成功 false : 失敗
	備註	—
16	<code>bool setFeature(uint8_t cmdFlag, uint32_t codeFeature)</code>	
	描述	設定模組資料更新方式
	參數	cmdFlag : 資料寫入方式 0x00 (FEATURE_DIR) : 覆蓋上一個值 0x10 (FEATURE_OR) : 與上一個值以 “或” 的方式寫入 0x20 (FEATURE_AND) : 與上一個值以 “與” 的方式寫入 codeFeature : 4byte , 具體參考 BLE_API 規格書 4.14 API_Feature
	返回值	執行情況： true : 成功 false : 失敗
	備註	—
17	<code>bool setPowerSaving(uint8_t mode)</code>	
	描述	設定模組節能模式
	參數	mode : 節能模式 0x00 : 睡眠 (預設) 0x01 : 深度睡眠 0x15 : 關機 . 保持深度睡眠 . 喚醒將強制重置 . RAM 中的所有設定重置為預設值
	返回值	執行情況： true : 成功 false : 失敗
	備註	在睡眠和深度睡眠模式下喚醒 . 必須發送偽位元組 (至少 100μs 長) ; 解除休眠使用喚醒函式 wakeUp()

	bool setWhiteList(bool erase, uint8_t whiteListAddr[], uint8_t mask[])	
	描述	修改白名單
18	參數	erase : 要對白名單進行的動作 true : 添加 false : 刪除 whiteListAddr[] : 要添加到白名單的設備位址 (6byte) mask[] : 位址遮罩，模組內部預設為 0xFFFFFFFF
	返回值	執行情況： true : 成功 false : 失敗
	備註	具體參考 BLE_API 規格書 4.20 API_WhiteList
	bool setTXpower(uint8_t power)	
	描述	設定發射功率
19	參數	power : 發射功率檔位範圍 : 0x00(min)~0x0F(max)
	返回值	執行情況： true : 成功 false : 失敗
	備註	例如： power=0x00 · TX 發射功率 = -32.5dBm power=0x05 · TX 發射功率 = -11.5dBm power=0xA · TX 發射功率 = +0.5dBm power=0xF · TX 發射功率 = +3.5dBm
	bool setCrystalOffset(uint8_t cload)	
	描述	設定外部 16MHz 晶振的射頻輸出頻率偏移量
20	參數	cload : 射頻輸出頻率偏移量，範圍 0x00~0x0f
	返回值	執行情況： true : 成功 false : 失敗
	備註	例如： cload=0x00 · 晶振頻率 = 15.99985MHz cload=0x05 · 晶振頻率 = 16.00004MHz cload=0xA · 晶振頻率 = 16.00031MHz cload=0xF · 晶振頻率 = 16.00074MHz
	bool restoreDefault()	
	描述	軟體重置，重置後恢復預設值
21	參數	—
	返回值	執行情況： true : 成功 false : 失敗
	備註	—
	bool reset()	
	描述	軟體重置，重置後維持之前的設定值
22	參數	—
	返回值	執行情況： true : 成功 false : 失敗
	備註	—

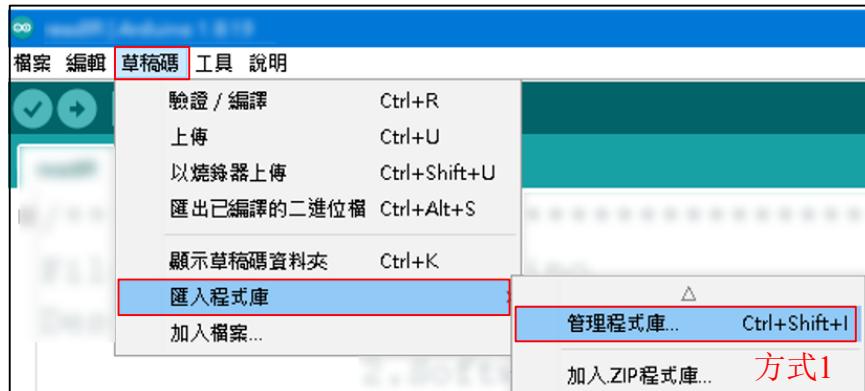
23	void wakeUp();	
	描述	喚醒處於休眠模式的藍牙模組
	參數	—
	返回值	void
	備註	需要等待 30ms，以完全喚醒藍牙模組
24	bool sendCMD(uint16_t type, uint8_t cmd_flag, uint8_t cmd_length, uint8_t value[])	
	描述	通用命令發送函式，向 BM7701-00-1 模組發送帶資料的命令
	參數	type : 類型 (UUID/Profile/Service/Characteristics/Descriptor) cmd_flag : API 命令的 flag cmd_length : type+value+cmd_flag 的位元組長度 value[] : 命令資料 Buffer
	返回值	執行情況： true : 成功 false : 失敗
	備註	24~27 為 BLE API CmdType2 的通用型函式，參數以及對應的資料參考 BLE_API 規格書
25	void sendCMD_NoResponse(uint16_t type, uint8_t cmd_flag, uint8_t cmd_length, uint8_t value[])	
	描述	通用命令發送函式不等待回應
	參數	type : 類型 (UUID/Profile/Service/Characteristics/Descriptor) cmd_flag : API 命令的 flag cmd_length : type+value+cmd_flag 的位元組長度 value[] : 命令資料 Buffer
	返回值	void
	備註	—
26	bool readCMD(uint16_t type, uint8_t cmd_flag)	
	描述	通用命令讀取函式
	參數	type : 類型 (UUID/Profile/Service/Characteristics/Descriptor) cmd_flag : API 命令的 flag
	返回值	執行情況： true : 成功 false : 失敗
	備註	—
27	void readCMD_NoResponse(uint16_t type, uint8_t cmd_flag)	
	描述	通用命令讀取函式不等待回應
	參數	type : 類型 (UUID/Profile/Service/Characteristics/Descriptor) cmd_flag : API 命令的 flag
	返回值	void
	備註	—

Arduino Lib 下載及安裝

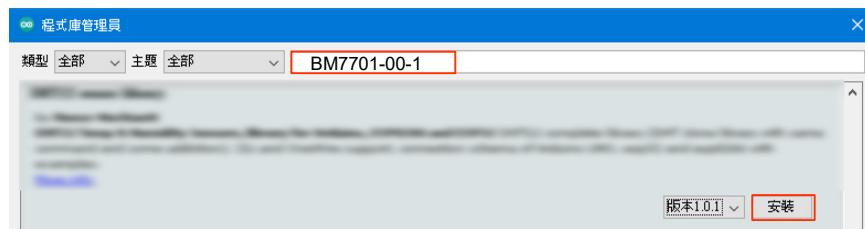
BM7701-00-1 Library：可參考下面兩種方法安裝 BM7701-00-1 的 Arduino Library

方式 1：搜索安裝

搜索安裝：Arduino IDE → 草稿碼 → 匯入程式庫 → 管理程式庫 → 搜索 BM7701-00-1 → 安裝



搜索安裝流程 1

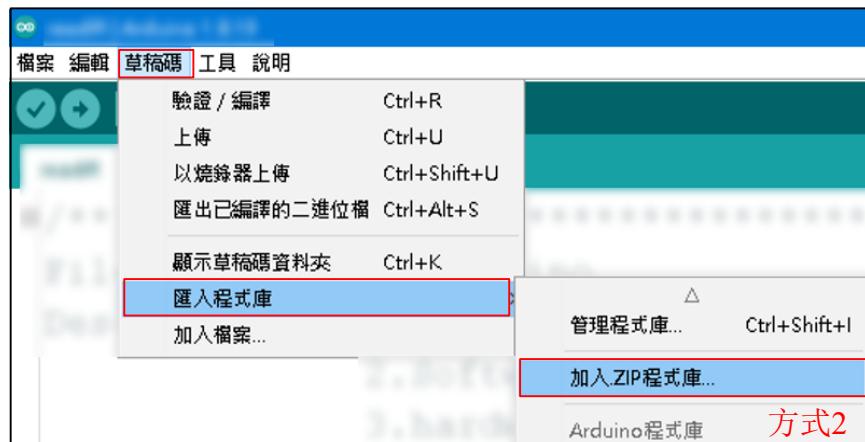


搜索安裝流程 2

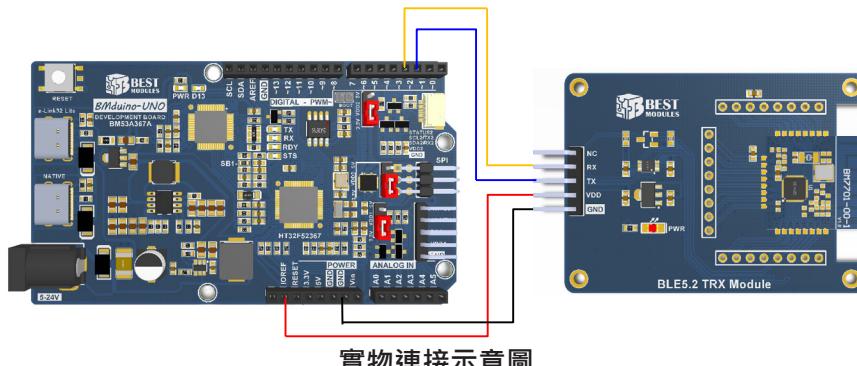
方式 2：添加 .ZIP 程式庫，需提前下載 .ZIP 程式庫

下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/BM7701-00-1.html>) “文件” 目錄下的 Arduino 範例程式 (BM7701-00-1 Library)。

添加 .ZIP 程式庫：Arduino IDE → 草稿碼 → 匯入程式庫 → 加入 .ZIP 程式庫 ...



Arduino 範例



範例 : writeAndRead

範例實現功能：模組連接手機 APP “BLEDemo” 或 “BCBLEDemo”，APP 上按下按鍵後，模組接收到對應資料，列印至序列埠監視視窗，再發送對應資料給 APP，使 APP 上的 LED 亮。APP 上再次按下此按鍵則 LED 滅。

1. 範例打開：檔案 → 範例 → Lib 選擇(BM7701-00-1) → 選擇範例(writeAndRead)
2. 範例說明：

- a. 調用庫函式，初始化模組，參數設定包括按鍵、LED 和部分藍牙參數設定

```
#include <BM7701-00-1.h>
BM7701_00_1          BC7701(2, 3);           // 使用軟體 UART
#define TX_POWER        0x0F                 // TX Power
#define XTAL_CLOAD      0x04                 // 16MHz 晶振負載
#define ADV_MIN          100                  // 廣播間隔最小值
#define ADV_MAX          100                  // 廣播間隔最大值
#define CON_MIN          30                   // 連接間隔最小值
#define CON_MAX          30                   // 連接間隔最大值
#define CON_LATENCY     00                   // 連接延時
#define CON_TIMEOUT      300                 // 連接超時
uint8_t BDAddress[6]={0x11, 0x22, 0x33, 0x44, 0x55, 0x66}; // 設備位址
uint8_t BDName[]={ 'B', 'M', 'C', '7', '7', 'M', '0', '0', '1' }; // 設備名稱
uint8_t Adata[]={0x02, 0x01, 0x06}; // 廣播資料
uint8_t Sdata[]={0x03, 0x02, 0x0f, 0x18}; // 被掃描後回應資料
#define BUTTON_CONSISTENCY_DURATION 6
#define BUTTON_REPEAT1_DURATION    (600/BUTTON_CONSISTENCY_DURATION)
#define BUTTON_REPEAT2_DURATION    (150/BUTTON_CONSISTENCY_DURATION)
#define INVERT_TIME                500
bool board_connect=false;
bool board_receive=false;
bool button_change=false;
bool board_conIntv=false;
bool txf=false;
uint8_t Status;
uint8_t flag=0;
```

```
uint8_t count=0;  
uint8_t sel=1;  
uint8_t receiveBuf[256]={0};  
KEY_MESSAGE Keymessage;
```

- b. 上電重置後，先延時 60ms 等待上電完成才能執行 begin() 函式。

```
void setup() {  
    delay(60); // 等待上電完成  
    Serial.begin(9600); // 設定序列埠監視視窗  
    BC7701.begin(BAUD_115200); // 模組初始化  
    setup()  
    while (sel != 10)  
    {  
        switch (sel)  
        {  
            case 1: if (BC7701.setAddress(BDAddress) == true) sel++; // 設定藍牙位址  
            else sel = 0xFF; break;  
            case 2: if (BC7701.setName(sizeof(BDName), BDName) == true) sel++;  
            else sel = 0xFF; break;  
            case 3: if (BC7701.setAdvIntv(ADV_MIN/0.625, ADV_MAX/0.625, 7) == true) sel++;  
            else sel = 0xFF; break; // 設定廣播間隔  
            case 4: if (BC7701.setAdvData(APPEND_NAME, sizeof(Adata), Adata) == true) sel++;  
            else sel = 0xFF; break; // 設定廣播資料  
            case 5: if (BC7701.setScanData(sizeof(Sdata), Sdata) == true) sel++;  
            else sel = 0xFF; break; // 設定廣播被掃描後的回應資料  
            case 6: if (BC7701.setTXpower(TX_POWER) == true) sel++;  
            else sel = 0xFF; break; // 設定 TX 發射功率  
            case 7: if (BC7701.setCrystalOffset(XTAL_CLOAD) == true) sel++;  
            else sel = 0xFF; break; // 設定外部晶振偏移值  
            case 8: if (BC7701.setFeature(FEATURE_DIR, AUTO_SEND_STATUS) == true) sel++;  
            else sel = 0xFF; break; // 設定資料更新方式  
            case 9: if (BC7701.setAdvCtrl(ENABLE) == true) sel++;  
            else sel = 0xFF; break; // 資料廣播功能  
            case 0xFF: digitalWrite(13,HIGH);break; // 設定失敗，LED13 亮  
        }  
    }  
    delay(650); // 等待廣播完全開啟  
}
```

c. 判斷藍牙模組狀態，如果連接成功就可以接收資料，判斷資料和連接狀態。

```

void loop()
{
    Status=bleProcess(); // 讀取 BLE 當前狀態
    if (Status)
    {
        switch (Status)
        {
            case API_CONNECTED: // 設備首次連接
                if (board_connect == false)
                {
                    board_connect=true;
                    board_receive=false;
                }
                break;
            case API_DISCONNECTED: // 設備連接失敗
                board_connect=false;
                board_receive=false;
                board_conIntv=false;
                break;
            case DATA_RECEIVED: // 設備接收到資料
                if (board_connect == true)
                {
                    digitalWrite(13,LOW);
                    board_receive=true;
                }
                break;
            case API_ERROR: // 錯誤
                digitalWrite(13,HIGH);
                break;
        }
    }
}

```

d. 連接成功後，設定連接間隔，接收資料來做對應動作

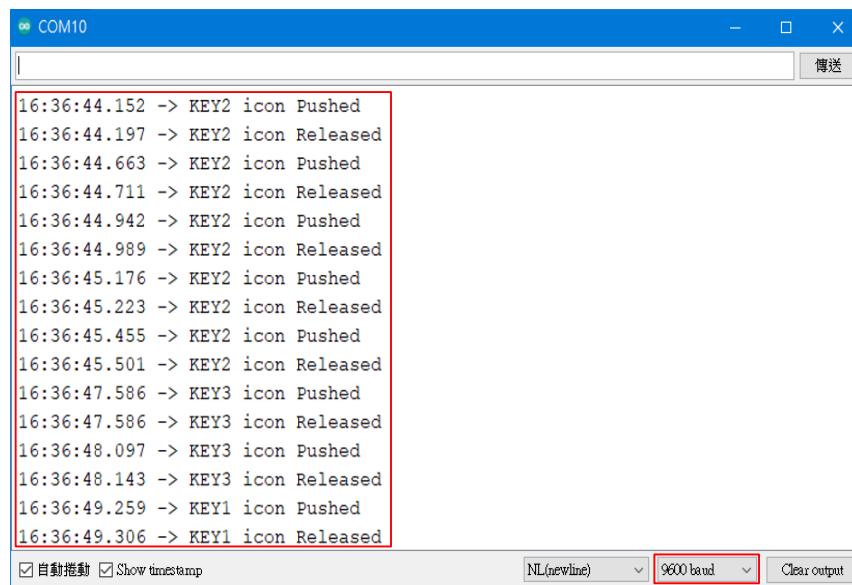
```

if (board_connect == true) // 連接狀態正確
{
    if (board_conIntv == false) // 判斷是否設定過連接間隔
    {
        BC7701.wakeUp();
        if (BC7701.setConnIntv(CON_MIN/1.25, CON_MAX/1.25, CON_LATENCY,
        CON_TIMEOUT) == true) // 設定連接間隔
        board_conIntv = true;
    }
    if (board_receive == true) // 接收到資料
    {
        board_receive = false; // 清空標誌
        if (receiveBuf[3] == 0xB0) // 資料封包幘頭
        {
            switch (receiveBuf[4]) // 按鍵資料位
            {
                case 0x11:
                    count=1;
                    Serial.println("KEY1 icon Pushed"); // 按鍵 1 按下
                    break;
            }
        }
    }
}

```

```
        case 0x10:
            count=2;
            Serial.println( "KEY1 icon Released" ); // 按鍵 1 鬆開
            break;
        case 0x22:
            count=1;
            Serial.println( "KEY2 icon Pushed" ); // 按鍵 2 按下
            break;
        case 0x20:
            count=2;
            Serial.println( "KEY2 icon Released" ); // 按鍵 2 鬆開
            break;
        case 0x44:
            count=1;
            Serial.println( "KEY3 icon Pushed" ); // 按鍵 3 按下
            break;
        case 0x40:
            count=2;
            Serial.println( "KEY3 icon Released" ); // 按鍵 3 鬆開
            break;
    }
    if (receiveBuf[4]!=0 && count==2&& flag==0) // 接收到 2 筆資料 .
                                                // 發送亮燈資料
    {
        Keymessage.key = receiveBuf[4]>>4;
        Keymessage.key += receiveBuf[4];
        Keymessage.serial++;
        Keymessage.checksum = 0xB1 ^ Keymessage.key ^ Keymessage.
        serial;
        BC7701.writeData((uint8_t*)&Keymessage,3);
        flag=1;
        receiveBuf[4] = 0;
    }
    if (receiveBuf[4]!=0&&count==2&&flag==1) // 再次接收到 2 筆資料 .
                                                // 已亮燈則滅燈
    {
        Keymessage.key = receiveBuf[4];
        Keymessage.serial++;
        Keymessage.checksum = 0xB1 ^ Keymessage.key ^ Keymessage.
        serial;
        BC7701.writeData((uint8_t*)&Keymessage,3);
        flag=0;
        receiveBuf[4] = 0;
    }
}
}
```

3. 打開序列埠監視視窗，鮑率選擇 9600；序列埠監視視窗收到 APP 的按鍵狀態如下：



```
16:36:44.152 -> KEY2 icon Pushed
16:36:44.197 -> KEY2 icon Released
16:36:44.663 -> KEY2 icon Pushed
16:36:44.711 -> KEY2 icon Released
16:36:44.942 -> KEY2 icon Pushed
16:36:44.989 -> KEY2 icon Released
16:36:45.176 -> KEY2 icon Pushed
16:36:45.223 -> KEY2 icon Released
16:36:45.455 -> KEY2 icon Pushed
16:36:45.501 -> KEY2 icon Released
16:36:47.586 -> KEY3 icon Pushed
16:36:47.586 -> KEY3 icon Released
16:36:48.097 -> KEY3 icon Pushed
16:36:48.143 -> KEY3 icon Released
16:36:49.259 -> KEY1 icon Pushed
16:36:49.306 -> KEY1 icon Released
```

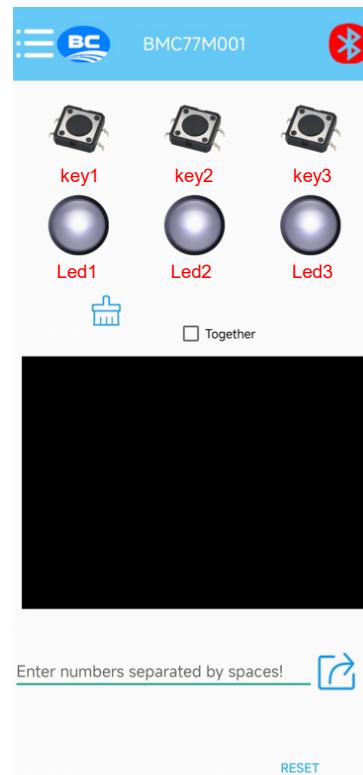
附錄

1. 手機 APP 安裝



Android

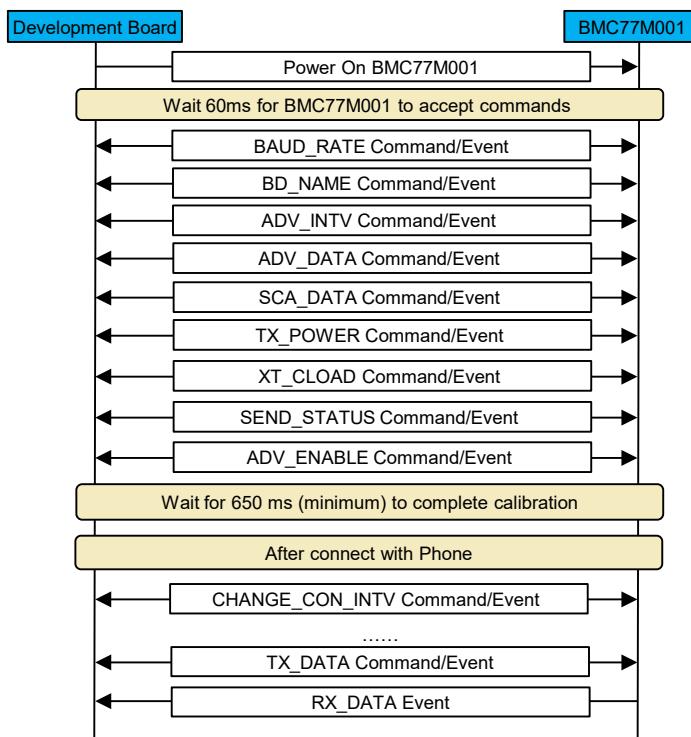
IOS



安裝後 UI 顯示

注：有部分手機型號，若出現 APP 打開後藍牙清單以及主介面存在文字會遺失的現象。請嘗試切換手機的顯示模式（日間模式 / 夜間模式 / 淺色模式 / 深色模式），至可看到文字即可。

2. 命令流程



Copyright® 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版時倍創已針對所載資訊為合理注意，但不保證資訊準確無誤。文中提到的資訊僅是提供作為參考，且可能被更新取代。倍創不擔保任何明示、默示或法定的，包括但不限於適合商品化、令人滿意的品質、規格、特性、功能與特定用途、不侵害第三人權利等保證責任。倍創就文中提到的資訊及該資訊之應用，不承擔任何法律責任。此外，倍創並不推薦將倍創的產品使用在會因故障或其他原因而可能會對人身安全造成危害的地方。倍創特此聲明，不授權將產品使用於救生、維生或安全關鍵零組件。在救生 / 維生或安全應用中使用倍創產品的風險完全由買方承擔，如因該等使用導致倍創遭受損害、索賠、訴訟或產生費用，買方同意出面進行辯護、賠償並使倍創免受損害。倍創（及其授權方，如適用）擁有本文件所提供之資訊（包括但不限於內容、資料、範例、材料、圖形、商標）的智慧財產權，且該資訊受著作權法和其他智慧財產權法的保護。倍創在此並未明示或暗示授予任何智慧財產權。倍創擁有不事先通知而修改本文件所載資訊之權利。如欲取得最新的資訊，請與我們聯繫。