

## M031BT 藍牙電動牙刷

NuMicro® 32位系列微控制器範例代碼介紹

### 文件資訊

應用簡述	本範例代碼使用 M031BT 來做藍牙電動牙刷方案
BSP 版本	M031_Series_BSP_CMSIS_V3.05.000
開發平台	Nuvoton_M031BT 藍牙電動牙刷方案板

*The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.*

*Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.*

*All data and specifications are subject to change without notice.*

*For additional information or questions, please contact: Nuvoton Technology Corporation.*

[www.nuvoton.com](http://www.nuvoton.com)

## 1. 概述

本範例代碼使用 M031BT 來做藍牙電動牙刷方案。

MCU特點為 Arm Cortex-M0 最大主頻 48MHz，具備12 路 96 MHz PWM，可提供作馬達控制來驅動並調節電動牙刷的振動頻率，16 通道 12位 2 MSPS 高取樣速率的 ADC提供精准的電壓量測，QFN48封裝面積僅有 5mm x 5mm利於系統硬體設計。

### 1.1 原理

通過新唐NuMicro® M031BT整合主控與藍牙二合一微控制器，搭配藍牙透傳資料傳輸功能與各廠商開發的APP結合，紀錄個人健康指數的資訊、口腔健康與刷牙習慣，加強與用戶端售後服務的連結。對於藍牙牙刷的電機模式設定、燈光顯示或者電量偵測，以上都可以透過一顆內建豐富周邊與優異模擬控制功能並整合藍牙技術的NuMicro® M031BT微控制處理器實現。

主控帶藍牙傳輸功能整合縮減硬體設計成本，手機APP連線參考設計快速上手。

開發包裡提供了Android系統手機的APP ( NuToothbrush v1.0.7.apk ) 供開發者演示。並提供了該方案的硬體PCB layout和電路圖，PCB版本是PADS9.5，電路圖版本是orCAD16.3。

電動牙刷上共有6個LED燈和1個按鍵，其主要功能為：

- 有一組4個LED燈（對應4個檔位元顯示）；
- 有一組2個LED燈，一個用於電源指示（電池電量過低時會閃爍報警），另一個用於藍牙連接狀態（開機狀態下，已連接BT、燈亮，斷開BT、燈滅）；
- 有1個按鍵（用於開關機、檔位元切換等控制）；
- 通過APP設置各檔的刷牙強度和時間，設定完成即可生效，還可儲存近期牙刷使用記錄。

### 1.2 ICP 下載程序

連接ICP工具，下載應用程式生成的M031BT\_NuToothbrush.bin到晶片內。配置如下圖 1-1和圖 1-2：

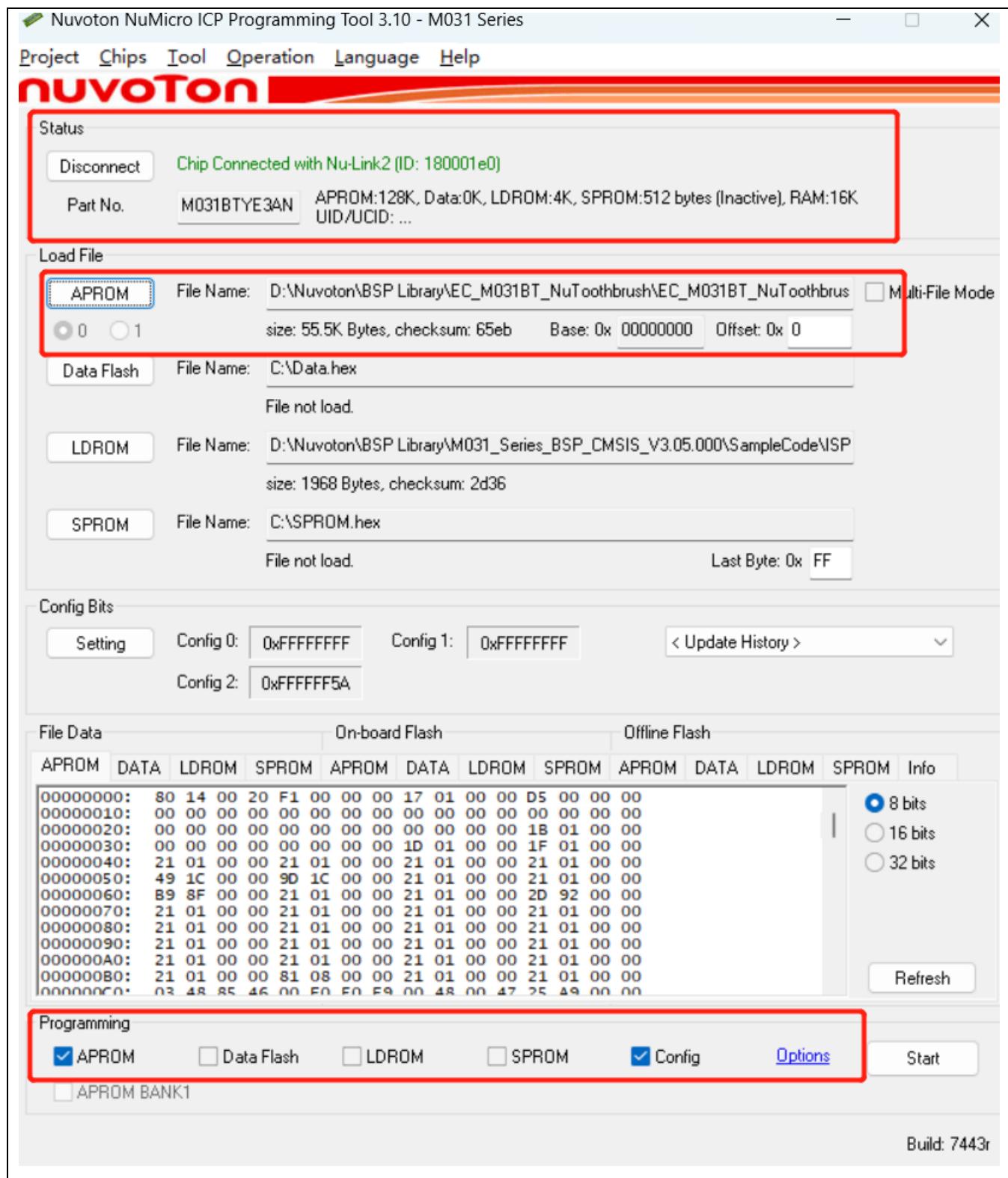


圖 1-1 ICP 配置

晶片配置為：

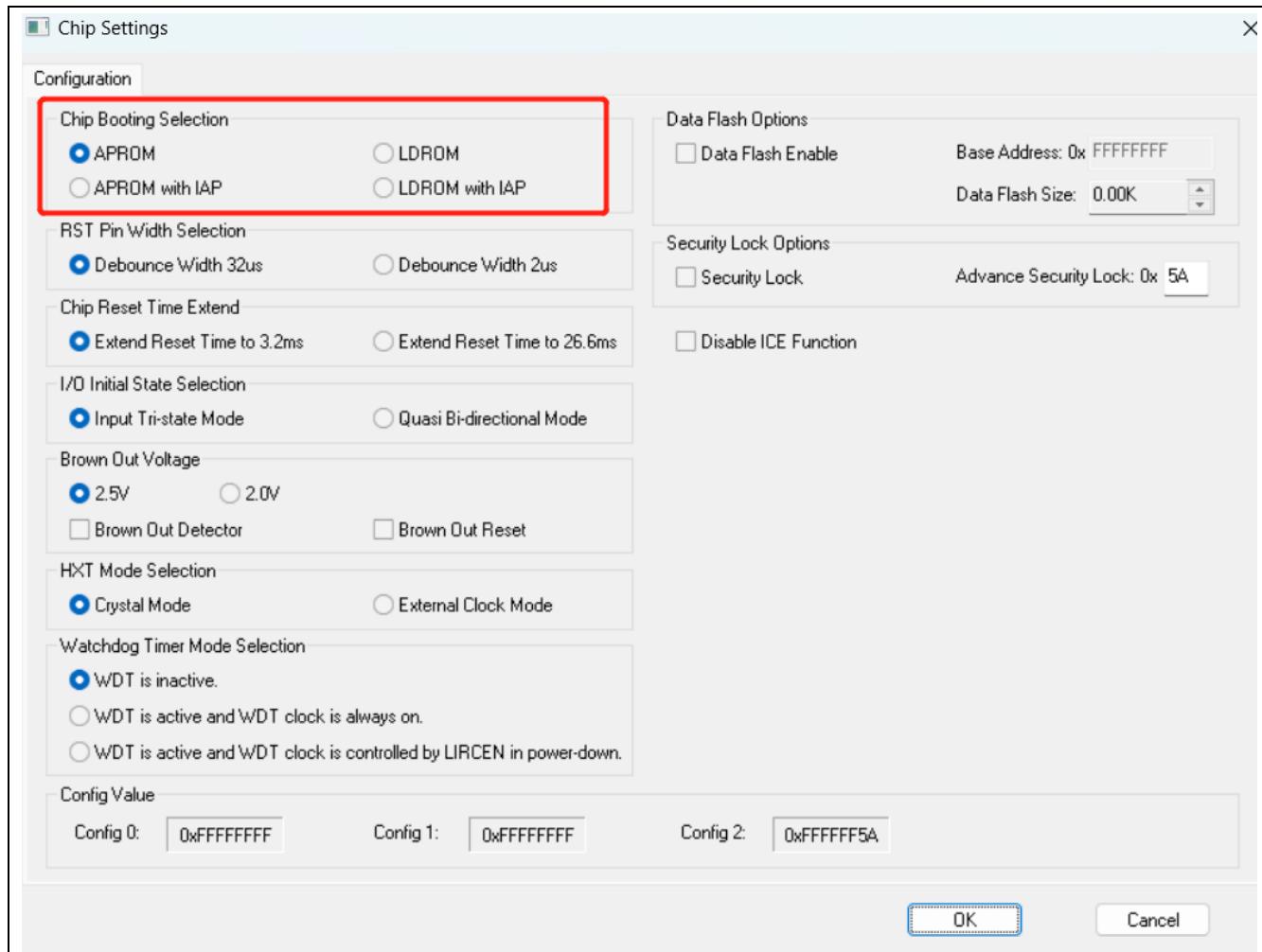


圖 1-2 晶片配置

### 1.3 執行結果

執行本範例代碼與 APP 連動的結果如圖 1-3 所示，以下是藍牙電動牙刷與手機藍牙 APP 已連接的狀態。



圖 1-3 藍牙電動牙刷與手機藍牙 APP 已連接的狀態

圖 1-4 是電動牙刷近期使用記錄，點選近期使用紀錄，可以查看牙刷近期使用的時間記錄。

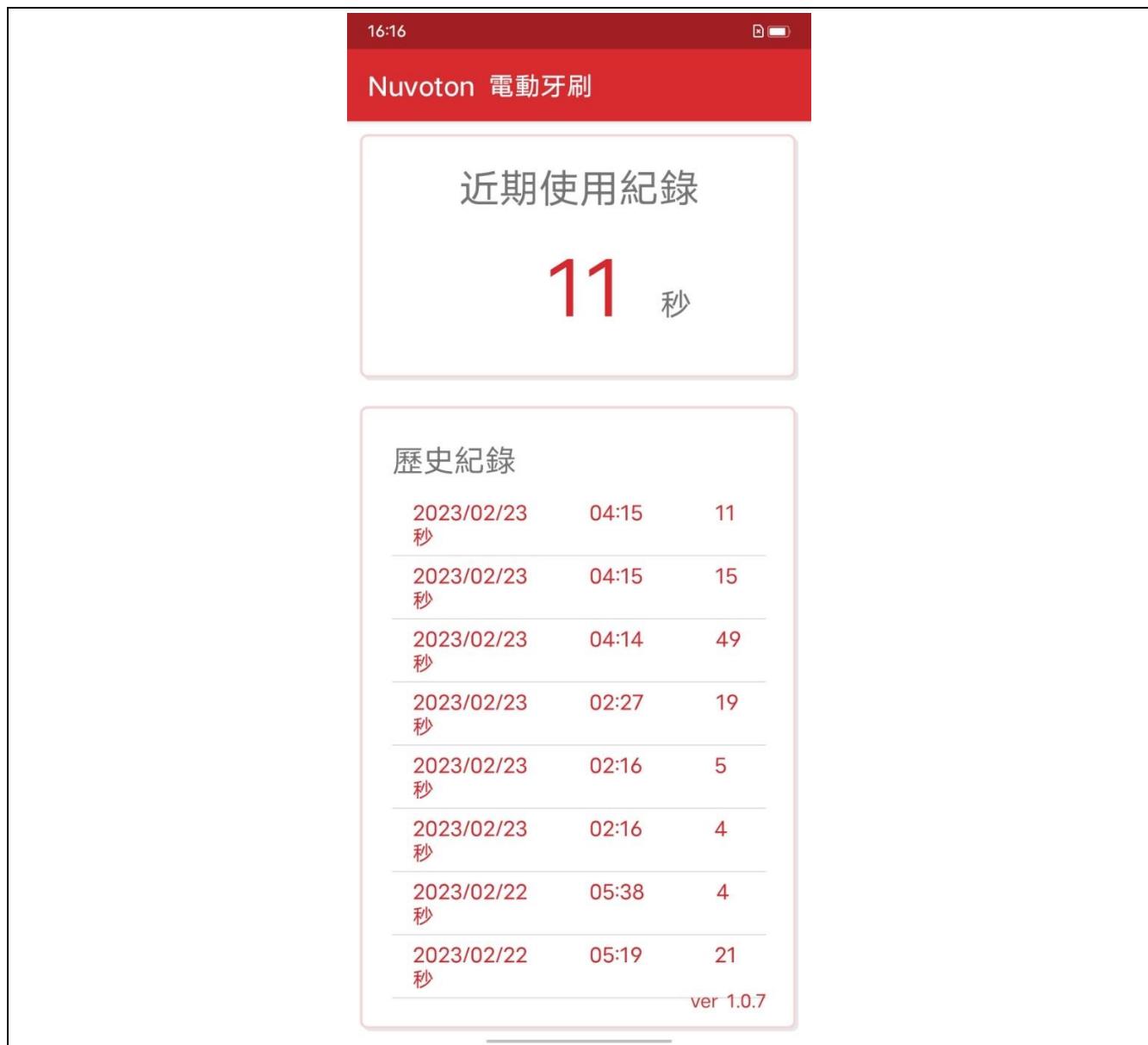


圖 1-4 電動牙刷近期使用記錄

## 2. 代碼介紹

按鍵處理函數。

```
void KeyPorc(void)
{
    if(KeyCode == KEY_CHANGE)
    {
        KeyCode = KEY_NONE;
        if(power)
        {
            if(sta<0x5)
            {
                if(sta==0x4)sta=0;
                sta++;
                onlyflag=1;
            }
            BT_Switch_Channel();
        }
    }
    else if(KeyCode == KEY_VERSION)
    {
        KeyCode = KEY_NONE;
        PowerON();
    }
    Motor_Change();
}
```

馬達擋位切換函數。

```
void Motor_Change(void)
{
    if(power&&onlyflag)
    {
        switch(sta)
        {
            case 0x1:Display_Level_1();onlyflag=0;break;
            case 0x2:Display_Level_2();onlyflag=0;break;
            case 0x3:Display_Level_3();onlyflag=0;break;
            case 0x4:Display_Level_4();onlyflag=0;break;
            default:break;
        }
    }
}
```

BLE 和 MCU 的休眠和喚醒功能。

```
/* Run user application */
BleApp_Main();

if (dsleep_enabled)
{
    status = setRF_WakeUpFromDeepSleep();
```

```
if (status == BLESTACK_STATUS_SUCCESS)
{
    /* Restore BLE settings after wakeup */
    Ble_Slave_Init();
    dsleep_enabled = 0;
    atcmd_set_adven(0);
}
}
if (((UART0->FIFOSTS & (UART_FIFOSTS_RXIDLE_Msk | UART_FIFOSTS_TXEMPTYF_Msk)) ==
(UART_FIFOSTS_RXIDLE_Msk | UART_FIFOSTS_TXEMPTYF_Msk)) && (tx_data_transmit_enable ==
0)&&(BAT_CHRG==1)&&(wakeflag==0))
{
    wakeflag=1;
    /* System enter Power Down mode & wait interrupt event. */
    setMCU_SystemPowerDown();
}
```

BLE 用戶資料傳輸。

```
//Send out RF data
void trspx_send(uint8_t *data, uint16_t len)
{
    int i = 0;

    tx_data_transmit_enable = 1;

    /* if there is data pending, still send old data and skip new data */
    if (notifyPending == 0)
    {
        txDataLength = len;

        //put UART data in txDataBuffer
        for (i = 0; i < len; i++)
        {
            txDataBuffer[i] = data[i];
        }
    }
}

void handle_AppLink0_TRSPP(void)
{

    BleStackStatus status;

    if (bleProfile_link0_info.bleState == STATE_BLE_STANDBY)
    {
        LED_BT = 1;
        power_on_flag = 1;
        // reset preferred MTU size
        setBLEGATT_PREFERREDMTU(conn_trsp_link_hostid, DEFAULT_MTU);

        // reset service data
        BleService_GATT_DataInit(&(bleProfile_link0_info.serviceGatt_info_s.data));
        BleService_UDF01S_DataInit(&(bleProfile_link0_info.serviceUDF01S_info_s.data));
    }
}
```

```
// enable advertisement
status = Ble_AdvStart(CONN_TRSP_LINK_HOSTID);
LED_BT = 1;

if (status == BLESTACK_STATUS_SUCCESS)
{
    bleProfile_link0_info.bleState = STATE_BLE_ADVERTISING;
    LED_BT = 1;
}
}
else if (bleProfile_link0_info.bleState == STATE_BLE_CONNECTION)
{
    if(power)
    {
        LED_BT = 0;

        if(power_on_flag > 0)
        {
            channel_count = 3;
            Report_BatteryLevel = 1;
            power_on_flag = 0;
        }
    }
    else
    {
        bleProfile_link0_info.bleState = STATE_BLE_STANDBY;
        power_on_flag = 1;
    }

    if ((tx_data_transmit_enable == 1) || (notifyPending == 1))
    {
        tx_data_transmit_enable = 0;

        // Send out data by RF
        /* Be careful that the client should enable NOTIFY, then the server can start
         sending notification data */
        status = setUDF01S_ServerDataSend(bleProfile_link0_info.hostId,
                                         
                                         bleProfile_link0_info.serviceUDF01S_info_s.data.udatn01_cccd,
                                         
                                         bleProfile_link0_info.serviceUDF01S_info_s.handles.hdl_udatn01,
                                         txDataBuffer,
                                         txDataLength);

        if (status == BLESTACK_STATUS_SUCCESS)
        {
            notifyPending = 0;

            if(Brush_Time_Record)
                Brush_Time_Record = 0;
        }
        else
        {
            notifyPending = 1;
        }
    }
}
```

```
    }
    else
        power_on_flag = 1;
}
```

BLE 用戶 ATcommand 命令處理函數。

```
AtCmdStatus atcmd_handler(char *data) /* AT command functions */
{
    AtCmdStatus ret = ATCMD_FAIL;
    uint8_t i;
    char pcCopy[12], pcCopy_t[3], pcCopy_p[3];
    strcpy(pcCopy, data);

    if (pcCopy[0] == 'M')
    {
        if (pcCopy[1] == '1')
        {
            if (pcCopy[2] == 'P')
            {
                pcCopy_p[0] = pcCopy[3];
                pcCopy_p[1] = pcCopy[4];
                pcCopy_p[2] = pcCopy[5];
            }
            else if (pcCopy[2] == 'R')
            {
                set_mode_p[0] = 0x127;
                set_mode_t[0] = 0x78;
            }
        }

        if (pcCopy[6] == 'T')
        {
            pcCopy_t[0] = pcCopy[7];
            pcCopy_t[1] = pcCopy[8];
            pcCopy_t[2] = pcCopy[9];
            sscanf(pcCopy_p, "%d", &set_p[0]);
            sscanf(pcCopy_t, "%d", &set_t[0]);
            set_mode_p[0] = (set_p[0] * 3);
            set_mode_t[0] = set_t[0];
        }

        sta = 0x1;
        Display_Level_1();
    }
    else if (pcCopy[1] == '2')
    {
        if (pcCopy[2] == 'P')
        {
            pcCopy_p[0] = pcCopy[3];
            pcCopy_p[1] = pcCopy[4];
            pcCopy_p[2] = pcCopy[5];
        }
        else if (pcCopy[2] == 'R')
```

```
{  
    set_mode_p[1] = 0x116;  
    set_mode_t[1] = 0x5A;  
}  
  
if(pcCopy[6] == 'T')  
{  
    pcCopy_t[0] = pcCopy[7];  
    pcCopy_t[1] = pcCopy[8];  
    pcCopy_t[2] = pcCopy[9];  
    sscanf(pcCopy_p, "%d", &set_p[0]);  
    sscanf(pcCopy_t, "%d", &set_t[0]);  
    set_mode_p[1] = (set_p[0] * 3);  
    set_mode_t[1] = set_t[0];  
}  
  
sta = 0x2;  
Display_Level_2();  
}  
else if (pcCopy[1] == '3')  
{  
    if (pcCopy[2] == 'P')  
    {  
        pcCopy_p[0] = pcCopy[3];  
        pcCopy_p[1] = pcCopy[4];  
        pcCopy_p[2] = pcCopy[5];  
    }  
    else if(pcCopy[2] == 'R')  
    {  
        set_mode_p[2] = 0x11F;  
        set_mode_t[2] = 0x3C;  
    }  
  
    if(pcCopy[6] == 'T')  
    {  
        pcCopy_t[0] = pcCopy[7];  
        pcCopy_t[1] = pcCopy[8];  
        pcCopy_t[2] = pcCopy[9];  
        sscanf(pcCopy_p, "%d", &set_p[0]);  
        sscanf(pcCopy_t, "%d", &set_t[0]);  
        set_mode_p[2] = (set_p[0] * 3);  
        set_mode_t[2] = set_t[0];  
    }  
  
    sta = 0x3;  
    Display_Level_3();  
}  
else if (pcCopy[1] == '4')  
{  
    if (pcCopy[2] == 'P')  
    {  
        pcCopy_p[0] = pcCopy[3];  
        pcCopy_p[1] = pcCopy[4];  
        pcCopy_p[2] = pcCopy[5];  
    }  
    else if(pcCopy[2] == 'R')
```

```
{  
    set_mode_p[3] = 0x10F;  
    set_mode_t[3] = 0x1E;  
}  
  
if(pcCopy[6] == 'T')  
{  
    pcCopy_t[0] = pcCopy[7];  
    pcCopy_t[1] = pcCopy[8];  
    pcCopy_t[2] = pcCopy[9];  
    sscanf(pcCopy_p, "%d", &set_p[0]);  
    sscanf(pcCopy_t, "%d", &set_t[0]);  
    set_mode_p[3] = (set_p[0] * 3);  
    set_mode_t[3] = set_t[0];  
}  
  
sta = 0x4;  
Display_Level_4();  
}  
}  
else if ((pcCopy[0] == 'U') && (pcCopy[1] == 'T'))  
{  
    sprintf((char*)use_record, "%d", Record_T[1]);  
  
    if(Record_T[1] < 100)  
    {  
        use_record[2] = use_record[1];  
        use_record[1] = use_record[0];  
        use_record[0] = '0';  
  
        if(Record_T[1] < 10)  
        {  
            use_record[2] = use_record[1];  
            use_record[1] = '0';  
        }  
    }  
  
    for(i = 0; i < 3; i++)  
    {  
        BT_record[i + 1] = use_record[i];  
    }  
  
    if(power)  
        BT_record[0] = '0';  
    else  
        BT_record[0] = 'D';  
  
    trspx_send(BT_record, 0x4);  
}  
else if ((pcCopy[0] == 'I') && (pcCopy[1] == 'D'))  
{  
    BLE_Addr_Param bleAddrParam;  
    uint8_t i, j, k = 0;  
    BT_ID_addr[0] = 'A';  
    getBLE_BleDeviceAddr(&bleAddrParam);  
}
```

```
for (i = 0; i < 6; i++)
{
    sprintf((char*)BT_ID, "%02x", bleAddrParam.addr[i]);

    for(j = 0; j < 2; j++)
    {
        k = k + 1;
        BT_ID_addr[k] = BT_ID[j];
    }
}

trspx_send(BT_ID_addr, 13);
}

return ret;
}
```

### 3. 軟體與硬體需求

#### 3.1 軟體需求

- BSP 版本
  - M031\_Series\_BSP\_CMSIS\_V3.05.000
- IDE 版本
  - Keil uVersion 5.28

#### 3.2 硬體需求

- 電路元件
  - Nuvoton\_M031BT 藍牙電動牙刷方案板
- 示意圖

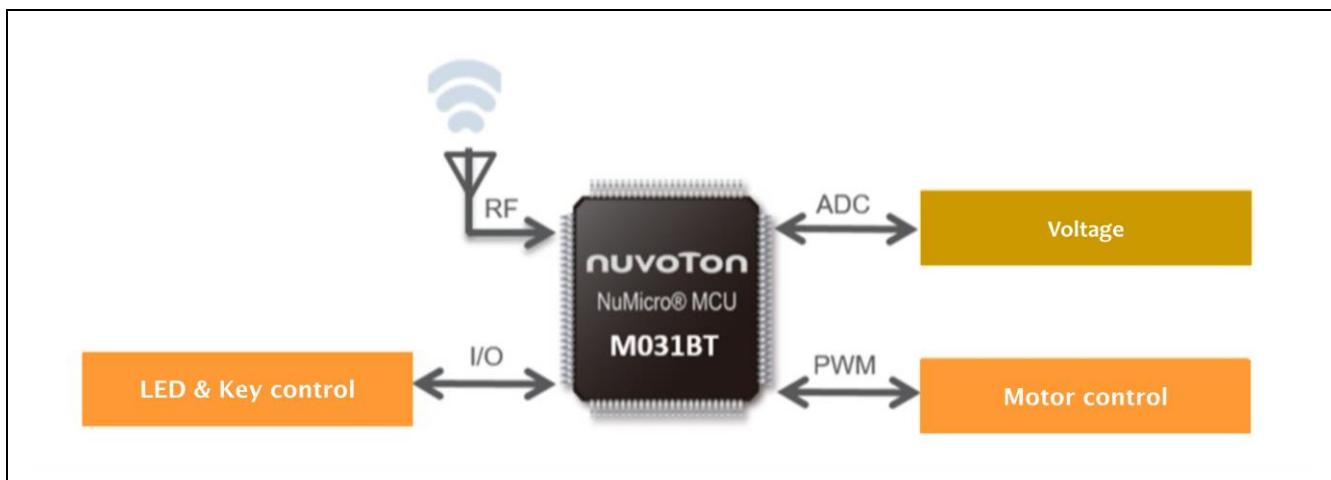


圖 3-1 Nuvoton\_M031BT 藍牙電動牙刷方案框圖

## 4. 目錄資訊

📁 EC_M031BT_NuToothbrush_V1.00	
📁 Library	Sample code header and source files
📁 CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
📁 Device	CMSIS compliant device header file
📁 StdDriver	All peripheral driver header and source files
📁 SampleCode	
📁 ExampleCode	Source file of example code
📁 App	Android APP file. Its website is: <a href="https://play.google.com/store/apps/details?id=com.nuvoton.nutoothbrush">https://play.google.com/store/apps/details?id=com.nuvoton.nutoothbrush</a>
📁 Hardware	Hardware schematics with PCB files
📁 M031BT_NuToothbrush	Keil project source file of example code

圖 4-1 目錄資訊

## 5. 範例程式執行

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 *M031BT\_NuToothbrush.uvproj*。
2. 進入編譯模式介面
  - 編譯
  - 下載代碼至記憶體
  - 進入 / 離開除錯模式
3. 進入除錯模式介面
  - 執行代碼

## 6. 修訂紀錄

Date	Revision	Description
2023.02.21	1.00	初始發布。

### Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

---

Please note that all data and specifications are subject to change without notice.  
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.