M031BT 蓝牙电动牙刷

NuMicro® 32位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 M031BT 来做蓝牙电动牙刷方案
BSP 版本	M031_Series_BSP_CMSIS_V3.05.000
开发平台	Nuvoton_M031BT 蓝牙电动牙刷方案板

1. 概述

本范例代码使用 M031BT 来做蓝牙电动牙刷方案。

MCU 特点为 Arm Cortex-M0 最大主频 48MHz，具备 12 路 96 MHz PWM，可提供作马达控制来驱动并调节电动牙刷的振动频率，16 通道 12 位 2 MSPS 高采样率的 ADC 提供精准的电压量测，QFN48 封装面积仅有 5mm x 5mm 利于系统硬件设计。

1.1 原理

通过新唐 NuMicro® M031BT 整合主控与蓝牙二合一微控制器，搭配蓝牙透传数据传输功能与各厂商开发的 APP 结合，纪录个人健康指数的信息、口腔健康与刷牙习惯，加强与客户端售后服务的连结。对于蓝牙牙刷的电机模式设定、灯光显示或者电量侦测，以上都可以透过一颗内建丰富周边与优异模拟控制功能并整合蓝牙技术的 NuMicro® M031BT 微控制处理器实现。

主控带蓝牙传输功能整合缩减硬件设计成本，手机 APP 联机参考设计快速上手。

开发包里提供了 Android 系统手机的 APP（NuToothbrush v1.0.7.apk）供开发者演示。并提供了该方案的硬件 PCB layout 和电路图，PCB 版本是 PADS9.5，电路图版本是 orCAD16.3。

电动牙刷上共有 6 个 LED 灯和 1 个按键，其主要功能为：

- 有一组 4 个 LED 灯（对应 4 个档位显示）；
- 有一组 2 个 LED 灯，一个用于电源指示（电池电量过低时会闪烁报警），
- 另一个用于蓝牙连接状态（开机状态下，已连接 BT、灯亮，断开 BT、灯灭）；
- 有 1 个按键（用于开关机、档位切换等控制）；
- 通过 APP 设置各档的刷牙强度和时间，设定完成即可生效，还可储存近期牙刷使用记录。

1.2 ICP 烧写程序

连接ICP工具，烧写示例程序生成的M031BT_NuToothbrush.bin到芯片。 配置如下图 1-1和图 1-2：

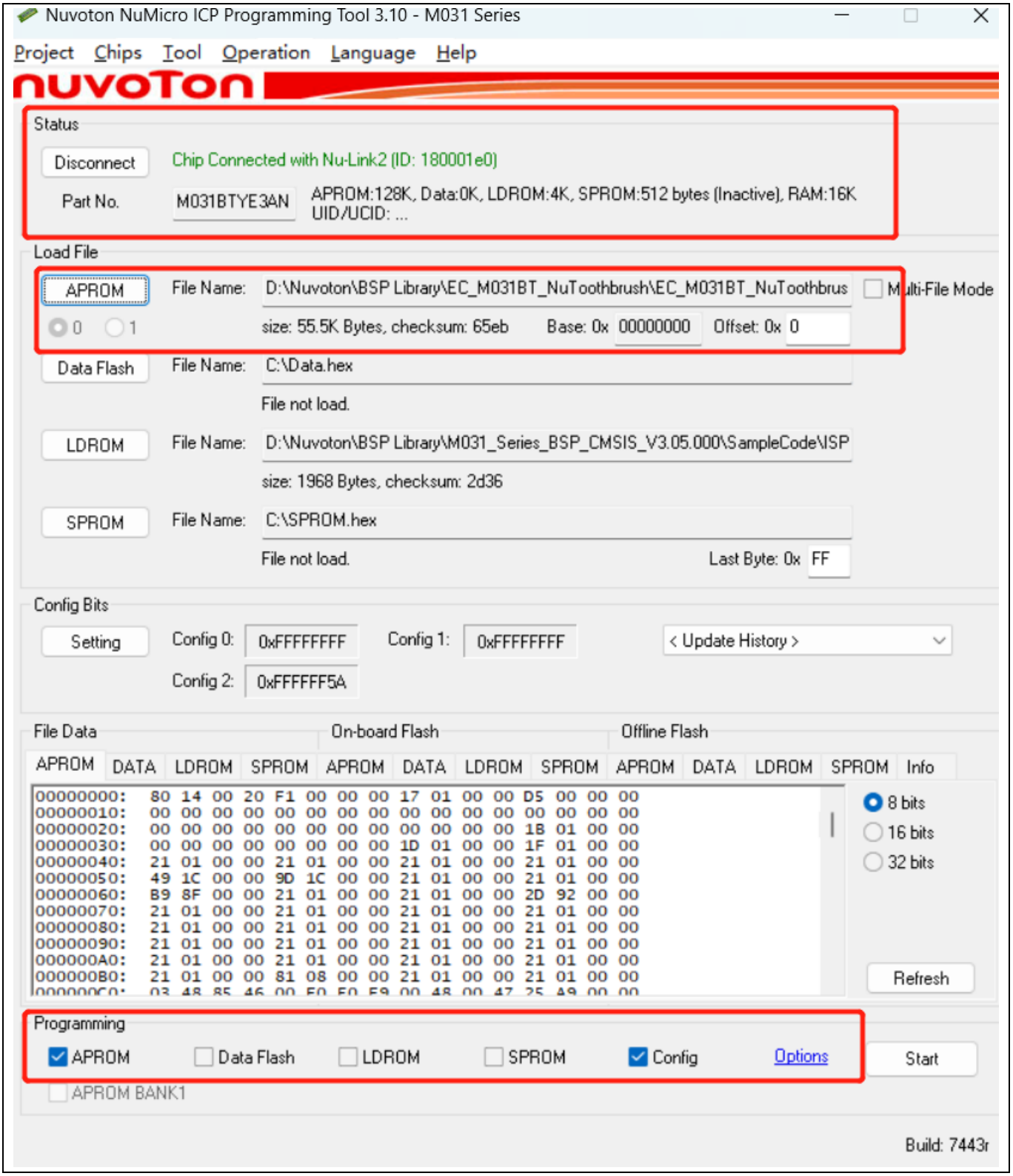
Nuvoton NuMicro ICP Programming Tool 3.10 - M031 Series
Project Chips Tool Operation Language Help
Status
Disconnect
Chip Connected with Nu-Link2 (ID: 180001e0)
Part No.
M031BTYE3AN
APROM:128K, Data:0K, LDROM:4K, SPROM:512 bytes (Inactive), RAM:16K
UID/UCID: ...
Load File
APROM
File Name: D:\Nuvoton\BSP Library\EC_M031BT_NuToothbrush\EC_M031BT_NuToothbrus
Multi-File Mode
0 1
size: 55.5K Bytes, checksum: 65eb
Base: 0x 00000000
Offset: 0x 0
Data Flash
File Name: C:\Data.hex
File not load.
LDROM
File Name: D:\Nuvoton\BSP Library\M031_Series_BSP_CMSIS_V3.05.000\SampleCode\ISP
size: 1968 Bytes, checksum: 2d36
SPROM
File Name: C:\SPROM.hex
File not load.
Last Byte: 0x FF
Config Bits
Setting
Config 0: 0xFFFFFFFF
Config 1: 0xFFFFFFFF
< Update History >
Config 2: 0xFFFFFF5A
File Data
On-board Flash
Offline Flash
APROM DATA LDROM SPROM APROM DATA LDROM SPROM APROM DATA LDROM SPROM Info
00000000: 80 14 00 20 F1 00 00 00 17 01 00 00 D5 00 00 00
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 1B 01 00 00
00000030: 00 00 00 00 00 00 00 00 1D 01 00 00 1F 01 00 00
00000040: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000050: 49 1C 00 00 9D 1C 00 00 21 01 00 00 21 01 00 00
00000060: B9 8F 00 00 21 01 00 00 21 01 00 00 2D 92 00 00
00000070: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000080: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000090: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
000000A0: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
000000B0: 21 01 00 00 81 08 00 00 21 01 00 00 21 01 00 00
8 bits
16 bits
32 bits
Refresh
Programming
APROM
Data Flash
LDROM
SPROM
Config
Options
Start
APROM BANK1
Build: 7443r

图 1-1 ICP 配置

芯片配置为：

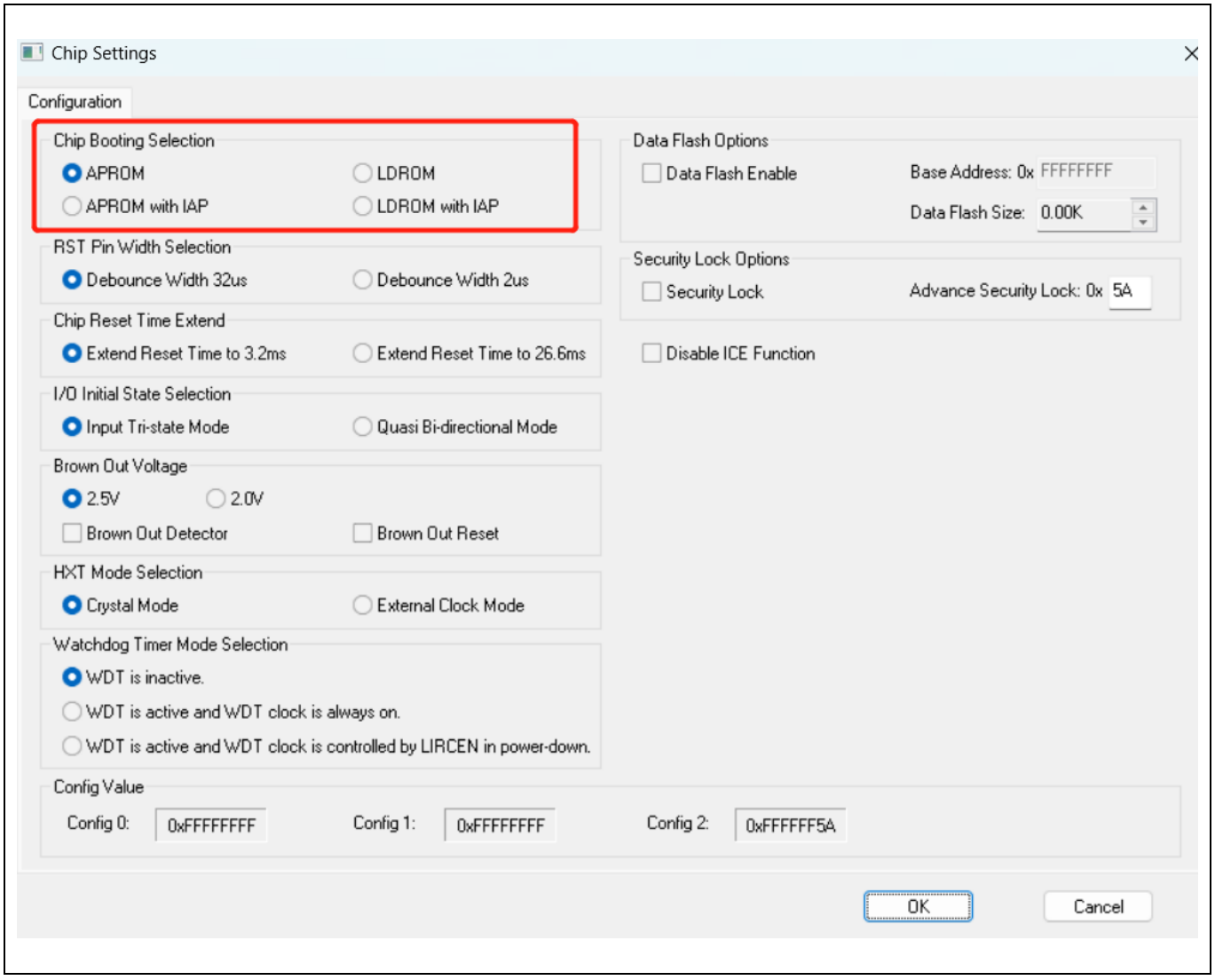

图 1-2 芯片配置

1.3 执行结果

执行本范例代码与 APP 连动的结果如图 1-3 所示，以下是蓝牙电动牙刷与手机蓝牙 APP 已连接的状态。

图 1-3 蓝牙电动牙刷与手机蓝牙 APP 已连接的状态

图 1-4 是电动牙刷近期使用记录，点选近期使用纪录，可以查看牙刷近期使用的时间记录。

图 1-4 电动牙刷近期使用记录

2. 代码介绍

按键处理函数。

```
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();
}
```

马达挡位切换函数。

```
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 的休眠和唤醒功能。

```
/* 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 用户数据传输。

```
//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_PreferredMtuSize(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 用户 ATcommand 命令处理函数。

```
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] = 'O';
        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. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - M031_Series_BSP_CMSIS_V3.05.000
- IDE 版本
  - Keil uVersion 5.28

3.2 硬件需求

- 电路组件
  - Nuvoton_M031BT 蓝牙电动牙刷方案板
- 示意图

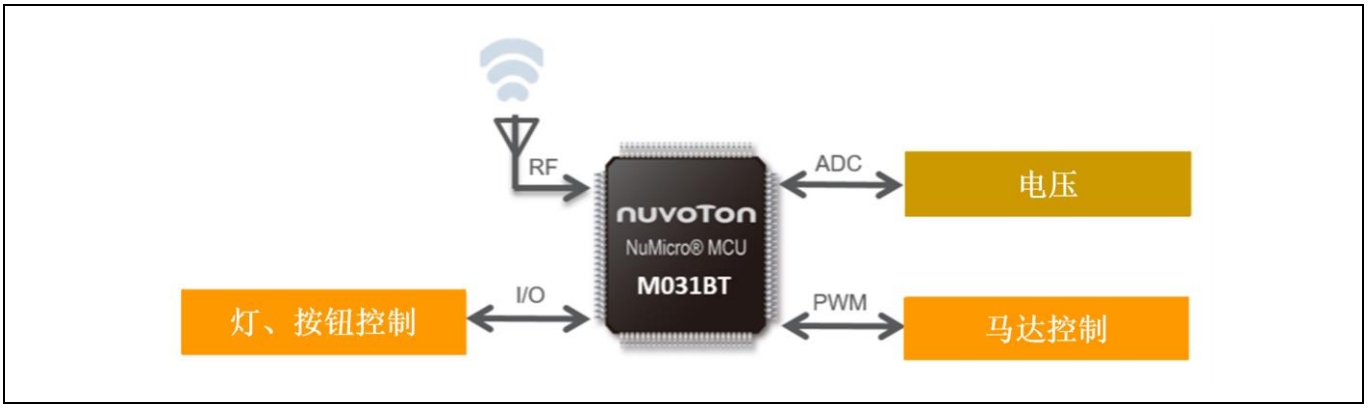

图 3-1 Nuvoton_M031BT 蓝牙电动牙刷方案框图

4. 目录信息

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: https://drive.google.com/file/d/1UgCkj5Cweq3EoHPrDaugUm43UM-mW-ud/view?usp=share_link
Hardware	Hardware schematics with PCB files
M031BT_NuToothbrush	Keil project source file of example code

图 4-1 目录信息

5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *M031BT_NuToothbrush.uvproj*。
2. 进入编译模式界面
   - 编译
   - 下载代码至内存
   - 进入 / 离开仿真模式
3. 进入仿真模式界面
   - 执行代码

6. 修订纪录

Date	Revision	Description
2023.02.21	1.00	初始发布。

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.