MS51 系列使用UART控制RGB燈光

NuMicro® 8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 UART 介面對 MS51 PWM 進行修改以達成 RGB 燈光控制
BSP 版本	MS51_Series_BSP_Keil_V1.00.004
開發平台	NuTiny-MS51FB V1.1

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

1. 概述

本範例代碼利用 UART 輸入對應於 MS51 PWM 的占空比來控制 RGB LED。用戶可以通過終端工具輸入 RGB 對應的 PWM 占空比，這樣可以實現亮度和色溫的調節。

1.1 原理

1.1.1 PWM 與 LED 顯示狀態

RGB LED 燈分為共陽極和共陰極，硬件電路見圖 1-1 和圖 1-3。 RGB LED 與 PWM 占空比的對應關係請參考圖 1-2 和圖 1-4。用戶可以通過程序代碼中的 “#define Com_Pole” 定義硬件動作狀態。

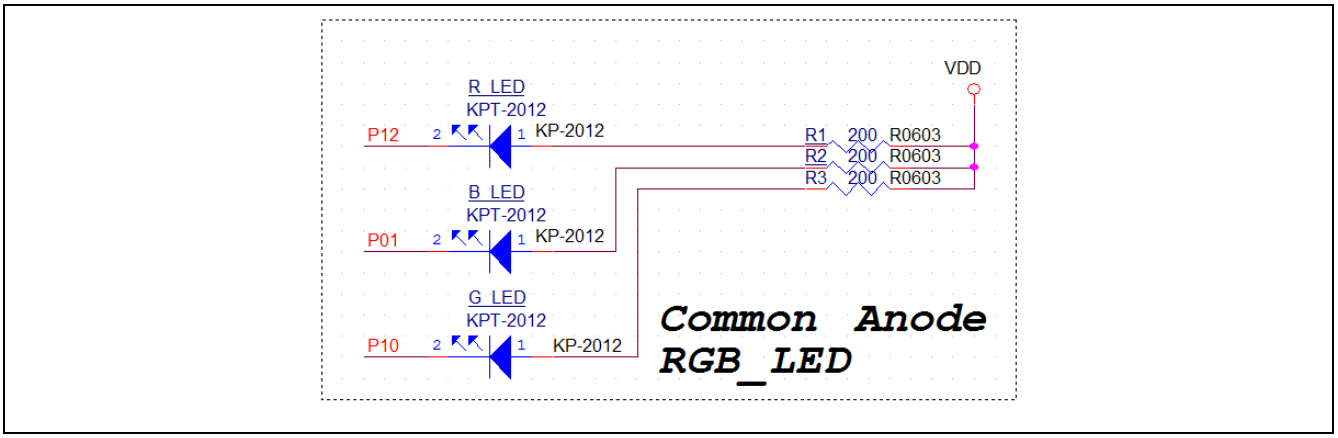

圖 1-1 共陽極 RGB 電路示意圖

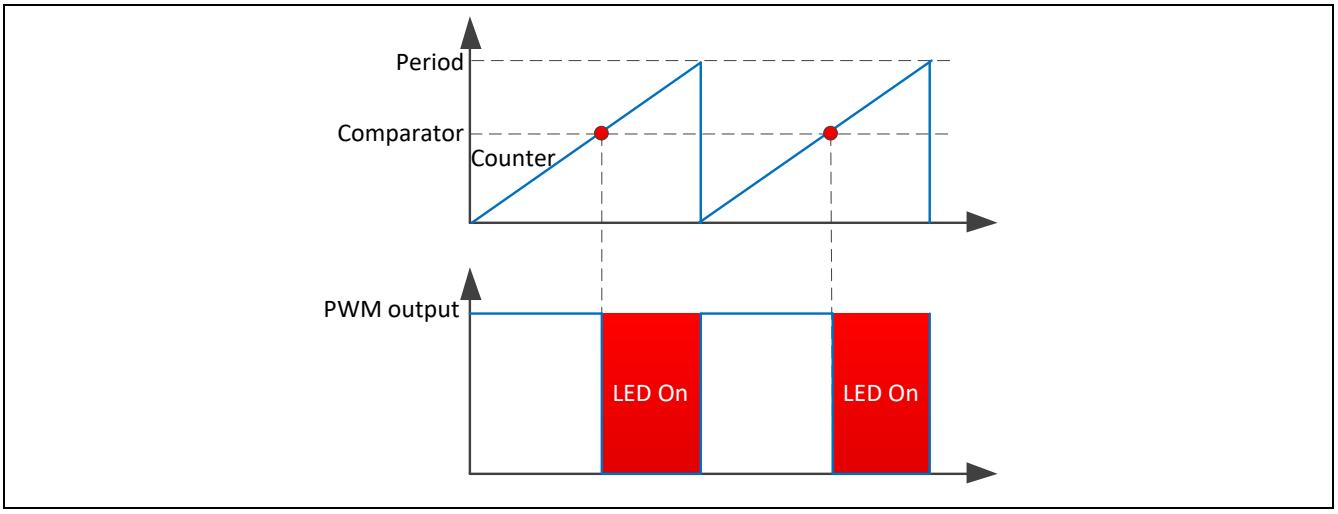

圖 1-2 共陽極 RGB 的 PWM 和燈狀態

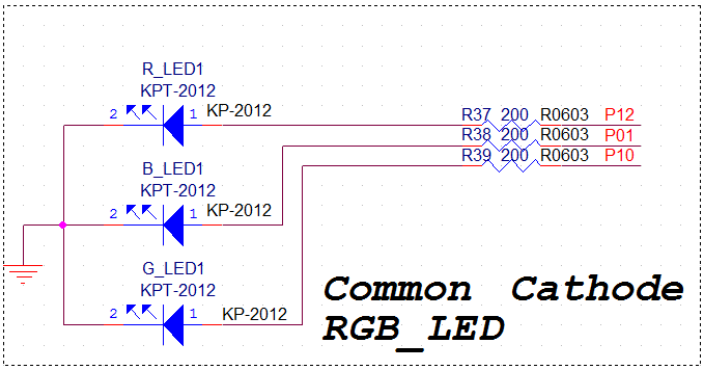

圖 1-3 共陰極 RGB 電路示意圖

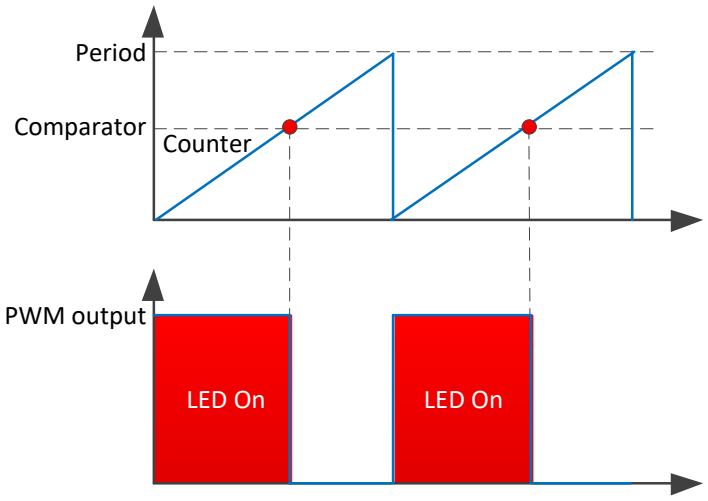

圖 1-4 共陰極 RGB 的 PWM 和燈狀態

1.1.2 UART 協議格式

輸入命令格式如表 1-1 所示。其中 Byte 0、Byte 4、Byte 8、Byte 12 為固定字符，必須與表格完全一致且格式必須剛好吻合 13 個字元。使用者必須嚴格按照格式正確輸入命令才能成功設置參數到 PWM 暫存器。否則會彈出一條錯誤消息告知使用者並保留原本的設定值不進行改變。各種命令格式分析及顯示信息請參考表 1-2。

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
‘ (’	‘ 0 ’	‘ 0 ’	‘ 0 ’	‘ , ’	‘ 0 ’	‘ 0 ’	‘ 0 ’
起始Byte	LED_R 占空比 (000-100)			分隔Byte	LED_B 占空比 (000-100)		
Byte 8	Byte 9	Byte 10	Byte 11	Byte 12			
‘ , ’	‘ 0 ’	‘ 0 ’	‘ 0 ’	‘) ’			
分隔Byte	LED_G 占空比 (000-100)			結束Byte			

表 1-1 輸入命令格式說明

情況	輸入命令	狀態	描述
1.	(050,100,002)	O	Message: Current Parameter Setting: R_Duty=50, B_Duty=100, G_Duty=2.
2.	(050,100,02)	X	Message: Please enter a command consisting of 13 characters according to the format requirements. Format:(R[000-100],B[000-100],G[000-100]),EX:(050,100,020). Current Parameter Setting: R_Duty=50, B_Duty=100, G_Duty=2.
3.	(050,100,020	X	Message: Please enter a command consisting of 13 characters according to the format requirements. Format:(R[000-100],B[000-100],G[000-100]),EX:(050,100,020). Current Parameter Setting: R_Duty=50, B_Duty=100, G_Duty=2.
4.	(050,100,120)	X	Message: The input parameter range is (0-100). Format:(R[000-100],B[000-100],G[000-100]),EX:(050,100,020). Current Parameter Setting: R_Duty=50, B_Duty=100, G_Duty=2.

表 1-2 輸入命令格式與狀態說明

1.2 執行結果

1.2.1 PuTTY 設定步驟

本範例代碼搭配“PuTTY”終端機應用程式，在 PC 端輸入正確的 RGB LED 占空比格式用以實現 RGB 調光，PuTTY 設置步驟如下：

步驟 1：打開 PuTTY 應用程式並點選“Terminal”。

步驟 2：點選“Local echo”項目下的“Force on”以啟用輸入值顯示，請參見圖 1-5。

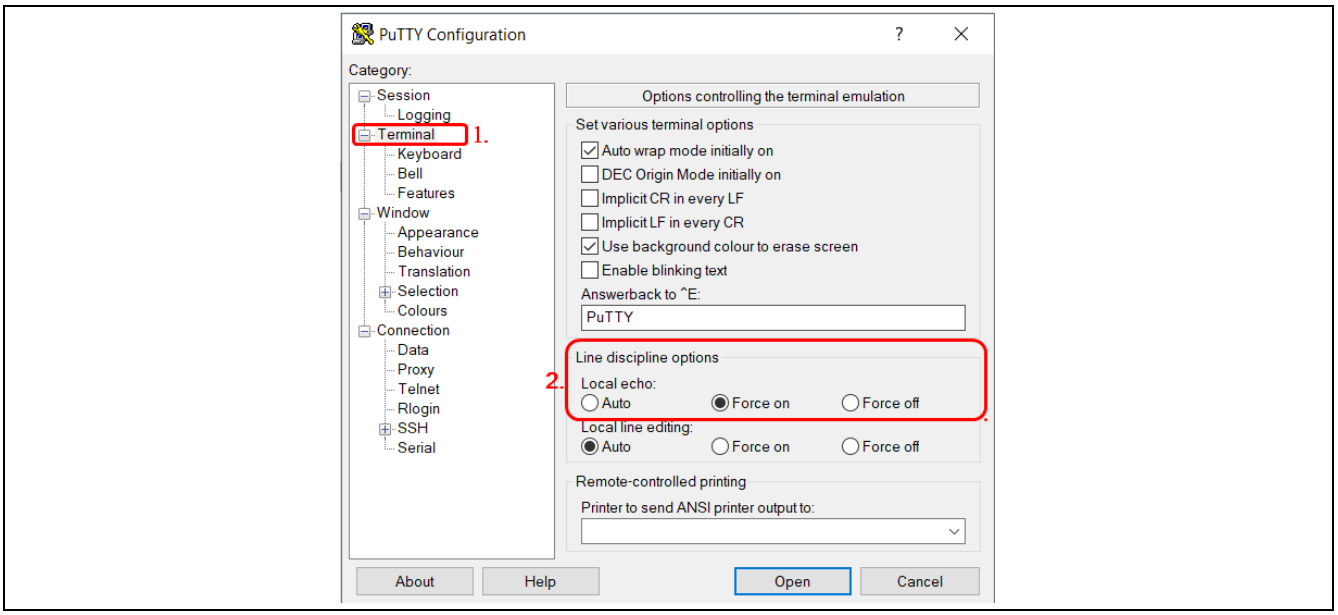

圖 1-5 PuTTY 終端設置

步驟 3：點選“Session”。

步驟 4：點選“Serial line”下的“Serial”，然後選擇端口號和波特率，請參考圖 1-6。

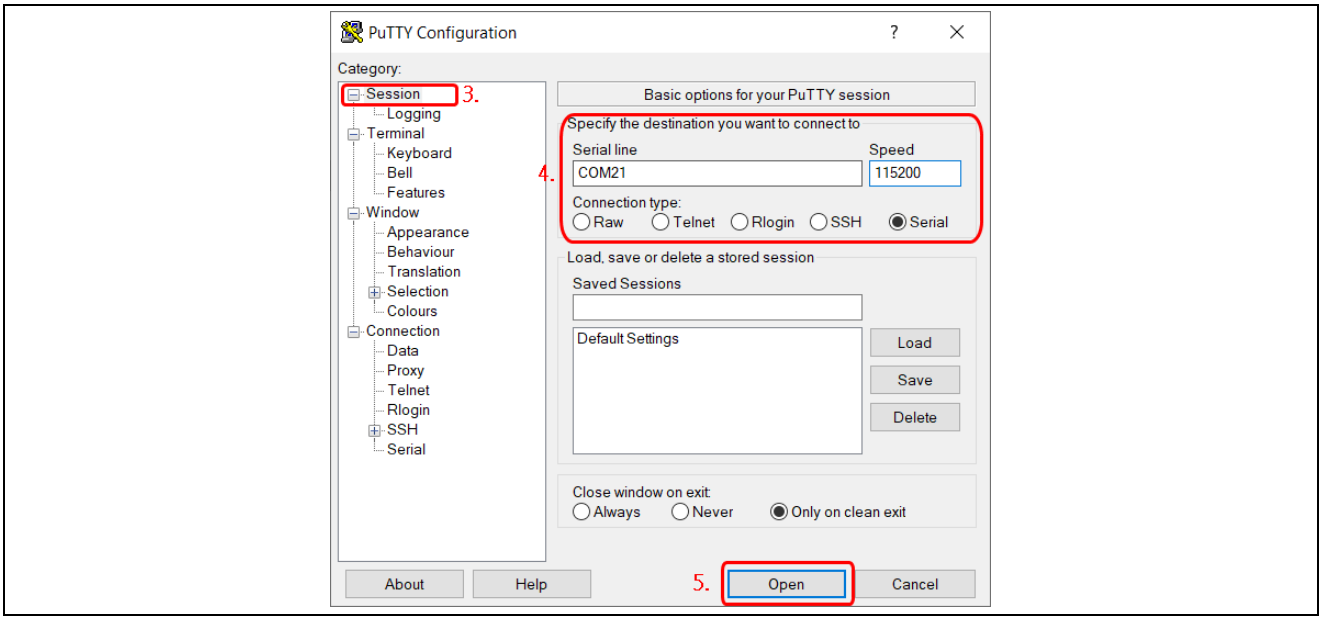

圖 1-6 PuTTY 的 Session 設置

1.2.2 動作視窗

本範例代碼執行編譯和燒錄的結果圖 1-7 所示。只有輸入正確的格式才能正確修改 PWM 占空比。另外，當用戶開始輸入第一個字元後，完整輸入命令需要在 10 秒內完成，否則會判斷為格式錯誤。

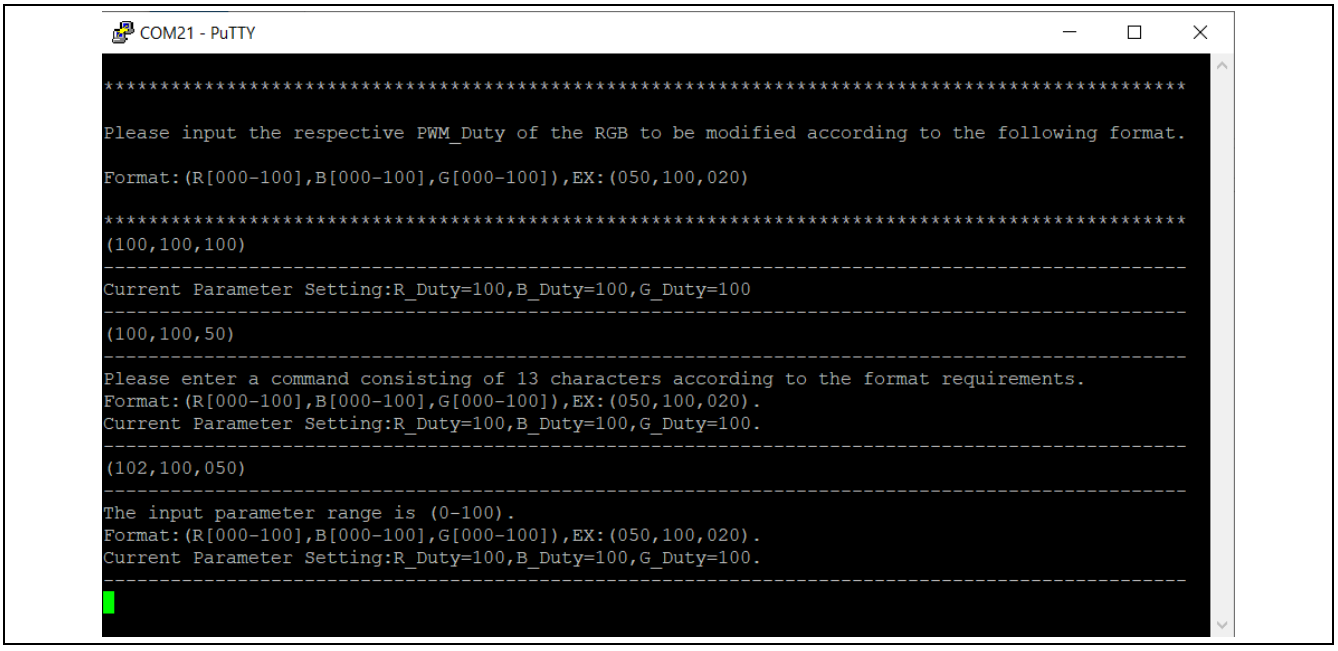

圖 1-7 執行編譯和燒錄結果

2. 代碼介紹

代碼說明如下

1. 初始化 MS51 相關功能包括 HIRC、UART 和 Timer 0，代碼位於 main.c。

```
void main(void)
{
    /************************************************************************/
    /* Clock Setting                                                        */
    /************************************************************************/
    MODIFY_HIRC(HIRC_24);

    /************************************************************************/
    /* Uart Setting                                                         */
    /************************************************************************/
    P06_QUASI_MODE;
    P07_QUASI_MODE;
    UART_Open(24000000, UART0_Timer3, 115200);
    ENABLE_UART0_INTERRUPT;                    /* Enable UART0 interrupt */
    ENABLE_UART0_PRINTF;

    ENABLE_GLOBAL_INTERRUPT;                   /* Global interrupt enable */

    ALL_GPIO_QUASI_MODE;

    /************************************************************************/
    /* PWM Setting                                                          */
    /************************************************************************/
    PWM0_P12_OUTPUT_ENABLE;
    PWM2_P10_OUTPUT_ENABLE;
    PWM4_P01_OUTPUT_ENABLE;


    PWM_IMDEPENDENT_MODE;
    PWM_CLOCK_DIV_8;
    PWMPH = 0x0B;
    PWMPL = 0xB7;
    /************************************************************************
      PWM frequency = Fpwm/((PWMPH,PWMPL) + 1) <Fpwm = Fsys/PWM_CLOCK_DIV>
                    = (24MHz/8)/(0xBB7 + 1)
                    = 1KHz (1ms)
    ************************************************************************/

    /************************************************************************/
    /* Timer0 Setting                                                       */
    /************************************************************************/
    ENABLE_TIMER0_MODE1;
    TIMER0_FSYS_DIV12;
    TH0 = TH0_INIT;
    TL0 = TL0_INIT;
```

2. 當 UART 接收到一個命令字符時，UART Serial_ISR()函數將被觸發。

```
void Serial_ISR(void) interrupt 4
{
    if (RI)                                 //Uart Receive Data
    {
        if (SBUF == 0x0d || SBUF == ')') //hit enter or ')' input is completed
        {
            u32cnt = 0;
            bTIMER0FLAG = 1;
            clr_TCON_TR0;
        }

        if (u8RVCT < 12)
        {
            if (SBUF == 0x7F || SBUF   == 0X08)          //Delete or Backspace
            {
                u8RVCT--;
            }
            else
            {
                au8UR0RVBUFFER[u8RVCT] = SBUF;
                u8RVCT++;
            }

            if (u8RVCT == 1)
            {
                ENABLE_TIMER0_INTERRUPT;                 /*enable Timer0 interrupt*/
                set_TCON_TR0;
            }

        }
        else if (u8RVCT == 12)
        {
            if (SBUF == 0x7F || SBUF   == 0X08)         //Delete or Backspace
            {
                u8RVCT--;
            }
            else
            {
                au8UR0RVBUFFER[u8RVCT] = SBUF;
                bUART0RVFLAG = 1;
                u8RVCT = 0;
                clr_TCON_TR0;
            }

        }

        clr_SCON_RI;                                    //Clear RI (Receive Interrupt).
    }
}
```

3. Timer0 負責對輸入時間進行計數，如果輸入時間過長，則判斷為格式錯誤。

```
void Timer0_ISR(void) interrupt 1      /*interrupt address is 0x000B */
{
    _push_(SFRS);

    TH0 = TH0_INIT;
    TL0 = TL0_INIT;
    TF0 = 0 ;
    P12 = ~P12;                        /* GPIO toggle when interrupt  */
    u32cnt++;

    if (u32cnt == 1000)                //10ms * 1000 = 10000 ms, /* Input command time
    {
        u32cnt = 0;
        bTIMER0FLAG = 1;
        clr_TCON_TR0;
    }

    bUART0RVFLAG = 0;
    _pop_(SFRS);
}
```

4. 格式錯誤信息，當輸入格式錯誤時，將丟棄本次輸入值並彈出格式錯誤信息。

```
void Format_Err(void)
{
    for (u8i = 0; u8i <= 12; u8i++)
    {
        au8UR0RVBUFFER[u8i] = 0;
    }

    u8RVCT = 0;
    bTIMER0FLAG = 0;
    clr_TCON_TR0;
    printf("\n------------------------------------------------------------------------------
--------------------\n");
    printf("Please enter a command consisting of 13 characters according to the format
requirements.");
    printf("\nFormat:(R[000-100],B[000-100],G[000-100]),EX:(050,100,020).");
    printf("\nCurrent Parameter Setting:R_Duty=%bd,B_Duty=%bd,G_Duty=%bd.", u8SysColor_R,
u8SysColor_B, u8SysColor_G);
    printf("\n------------------------------------------------------------------------------
--------------------\n");
}
```

5. 分析輸入格式的正確性。如果正確，則更改 PWM 占空，否則將保持原始設置不做修改。

```
printf("\n**************************************************************************
********************\n");
    printf("\nPlease input the respective PWM_Duty of the RGB to be modified according to
the following format.\n");
    printf("\nFormat:(R[000-100],B[000-100],G[000-100]),EX:(050,100,020)\n");

printf("\n**************************************************************************
********************\n");
```

```
    while (1)
    {
        if (bTIMER0FLAG == 1)           //Enter less than 12 characters.
        {
            bTIMER0FLAG = 0;
            Format_Err();
        }

        P14 ^= 1;

        if (bUART0RVFLAG)               //Uart receives completion and starts parsing.
        {
            if (au8UR0RVBUFFER[0] == '(') //Input format parsing.
            {
                if (au8UR0RVBUFFER[4] == ',' && au8UR0RVBUFFER[8] == ',')
                {
                    if (au8UR0RVBUFFER[12] == ')')
                    {
                        u8Color_R = (au8UR0RVBUFFER[1] - 0x30) * 100;
                        u8Color_R += (au8UR0RVBUFFER[2] - 0x30) * 10;
                        u8Color_R += (au8UR0RVBUFFER[3] - 0x30);
                        u8Color_B = (au8UR0RVBUFFER[5] - 0x30) * 100;
                        u8Color_B += (au8UR0RVBUFFER[6] - 0x30) * 10;
                        u8Color_B += (au8UR0RVBUFFER[7] - 0x30);
                        u8Color_G = (au8UR0RVBUFFER[9] - 0x30) * 100;
                        u8Color_G += (au8UR0RVBUFFER[10] - 0x30) * 10;
                        u8Color_G += (au8UR0RVBUFFER[11] - 0x30);

                        if (u8Color_R <= 100 && u8Color_B <= 100 && u8Color_G <= 100)
//Input Parameter range.
                        {
                            bTIMER0FLAG = 0;
                            u8SysColor_R = u8Color_R;
                            u8SysColor_B = u8Color_B;
                            u8SysColor_G = u8Color_G;

                            printf("\n-------------------------------------------------
-----------------------------------------\n");
                            printf("Current Parameter Setting:
R_Duty=%bd,B_Duty=%bd,G_Duty=%bd", u8SysColor_R, u8SysColor_B, u8SysColor_G);
                            printf("\n-------------------------------------------------
-----------------------------------------\n");

                            PWM0H = HIBYTE((MAKEWORD(PWMPH, PWMPL) / 99) * u8SysColor_R);
                            PWM0L = LOBYTE((MAKEWORD(PWMPH, PWMPL) / 99) * u8SysColor_R);
                            PWM2H = HIBYTE((MAKEWORD(PWMPH, PWMPL) / 99) * u8SysColor_B);
                            PWM2L = LOBYTE((MAKEWORD(PWMPH, PWMPL) / 99) * u8SysColor_B);

                            set_SFRS_SFRPAGE;
                            //PWM4 and PWM5 duty setting is in SFP page 1
                            PWM4H = HIBYTE((MAKEWORD(PWMPH, PWMPL) / 99) * u8SysColor_G);
                            PWM4L = LOBYTE((MAKEWORD(PWMPH, PWMPL) / 99) * u8SysColor_G);
                            clr_SFRS_SFRPAGE;

/*****************************************************************************
```

```
PWM high level duty = (PWM period/((PWMPH,PWMPL) + 1) * duty
percentage(%)

*****************************************************************/
#if Com_Pole    /* PWM output inversely enable */
                //Common anode: #define Com_Pole 1 ,Common cathode: #define Com_Pole 0
                            PWM0_OUTPUT_INVERSE;
                            PWM2_OUTPUT_INVERSE;
                            PWM4_OUTPUT_INVERSE;
#endif
                        /*-------- PWM start run--------------*/
                            set_PWMCON0_LOAD;
                            set_PWMCON0_PWMRUN;
                        }
                        else
                        {
                            for (u8i = 0; u8i <= 12; u8i++)
                            {
                                au8UR0RVBUFFER[u8i] = 0;
                            }

                            u8RVCT = 0;
                            bTIMER0FLAG = 0;
                            printf("\n----------------------------------------------------
-------------------------------------------\n");
                            printf("The input parameter range is (0-100).");
                            printf("\nFormat:(R[000-100],B[000-100],G[000-
100]),EX:(050,100,020).");
                            printf("\nCurrent Parameter Setting:
R_Duty=%bd,B_Duty=%bd,G_Duty=%bd.", u8SysColor_R, u8SysColor_B, u8SysColor_G);
                            printf("\n----------------------------------------------------
-------------------------------------------\n");
                        }
                    }
                    else
                        Format_Err();
                }
                else
                    Format_Err();
            }
            else
                Format_Err();

            bUART0RVFLAG = 0;
        }
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - MS51_Series_BSP_Keil_V1.00.004
- IDE 版本
  - Keil uVersion 5.28

3.2 硬體需求

- 電路元件
  - NuTiny-MS51FB V1.1
  - RGB LED
- 示意圖

V_{DD}	↔	V_{DD}
P12(PWM0_CH0)	↔	R Color
P01(PWM0_CH4)	↔	G Color
P10(PWM0_CH2)	↔	B Color
NuTiny-MS51FB		RGB LED

圖 3-1 RGB LED 與 NuTiny-MS51FB V1.1 連接示意圖

4. 目錄資訊

目錄結構如下圖所示

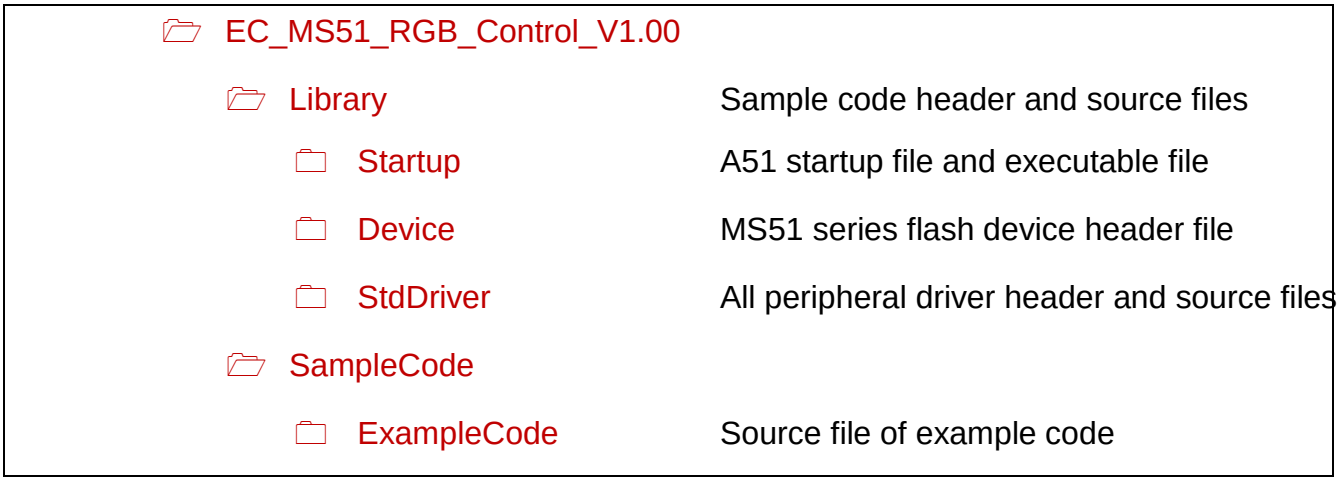

圖 4-1 目錄資訊

5. 範例程式執行

1. 根據目錄資訊章節進入Example Code 路徑中的KEIL資料夾，滑鼠右鍵點擊兩次 MS51_RGB_Control.uvproj。

2. 進入編譯模式介面

   - 編譯。
   - 下載代碼至記憶體。

3. 打開 PuTTY

   - 按一下重置按鈕。
   - 按照格式(R[000-100],B[000-100],G[000-100]),EX:(050,100,020) 輸入R、B、G對應的PWM參數。
   - 如果輸入格式正確則修改PWM占空比，否則保持原參數不變。

6. 修訂紀錄

Date	Revision	Description
2022.07.06	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.