MS51 系列使用UART控制RGB灯光

NuMicro® 8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 UART 接口对 MS51 PWM 进行修改以达成 RGB 灯光控制
BSP 版本	MS51_Series_BSP_Keil_V1.00.004
开发平台	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

本范例代码利用 UART 输入对应于 MS51 PWM 的占空比来控制 RGB LED。用户可以通过终端工具输入 RGB 对应的 PWM 占空比，这样可以实现亮度和色温的调节。

1.1 原理

1.1.1 PWM 与 LED 显示状态

RGB LED 灯分为共阳极和共阴极，硬件电路见图 1-1 和图 1-3。 RGB LED 与 PWM 占空比的对应关系请参考图 1-2 和图 1-4。用户可以通过程序代码中的 “#define Com_Pole” 定义硬件动作状态。

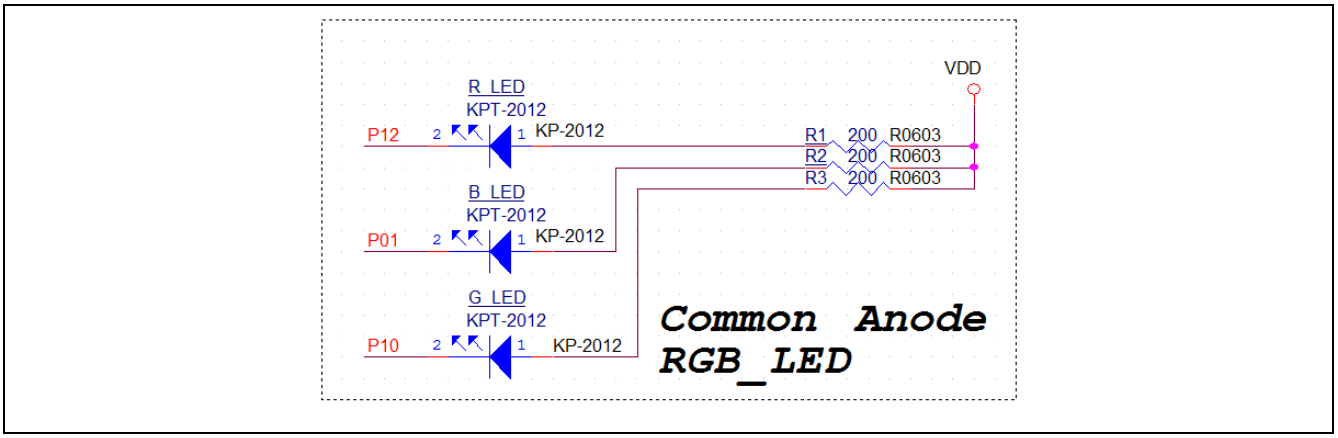

图 1-1 共阳极 RGB 电路示意图

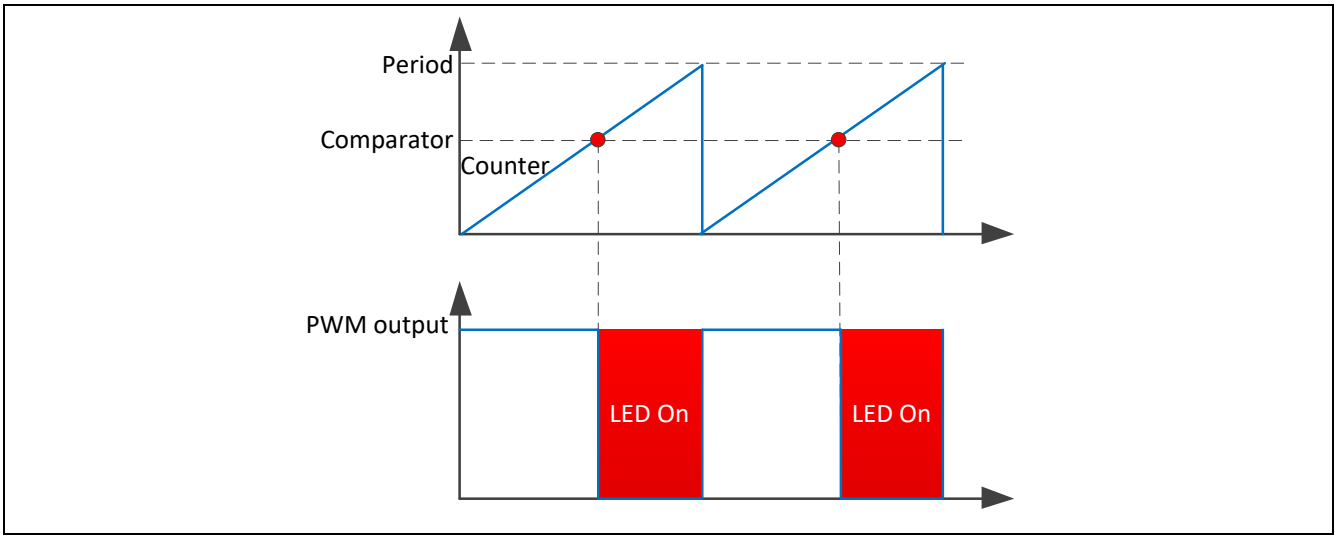

图 1-2 共阳极 RGB 的 PWM 和灯状态

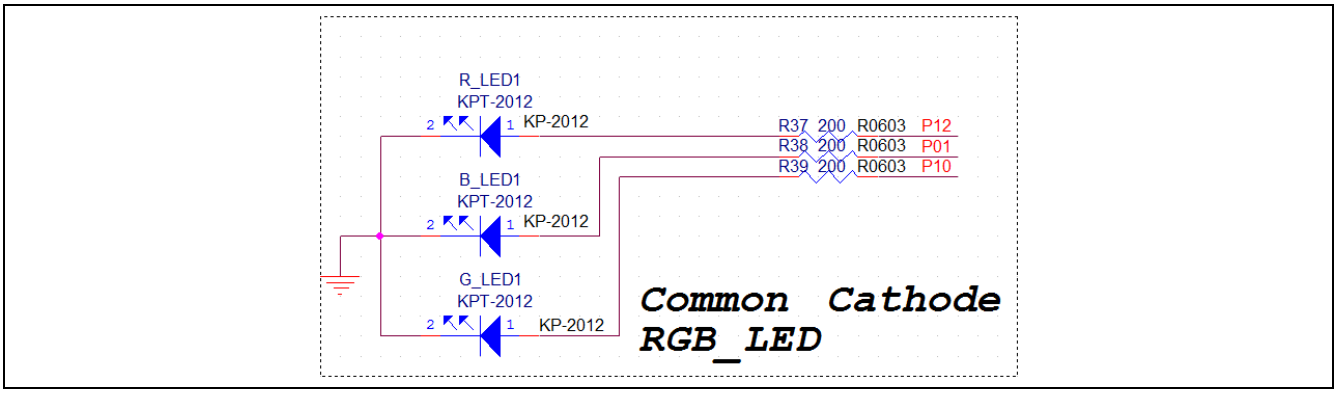

图 1-3 共阴极 RGB 电路示意图

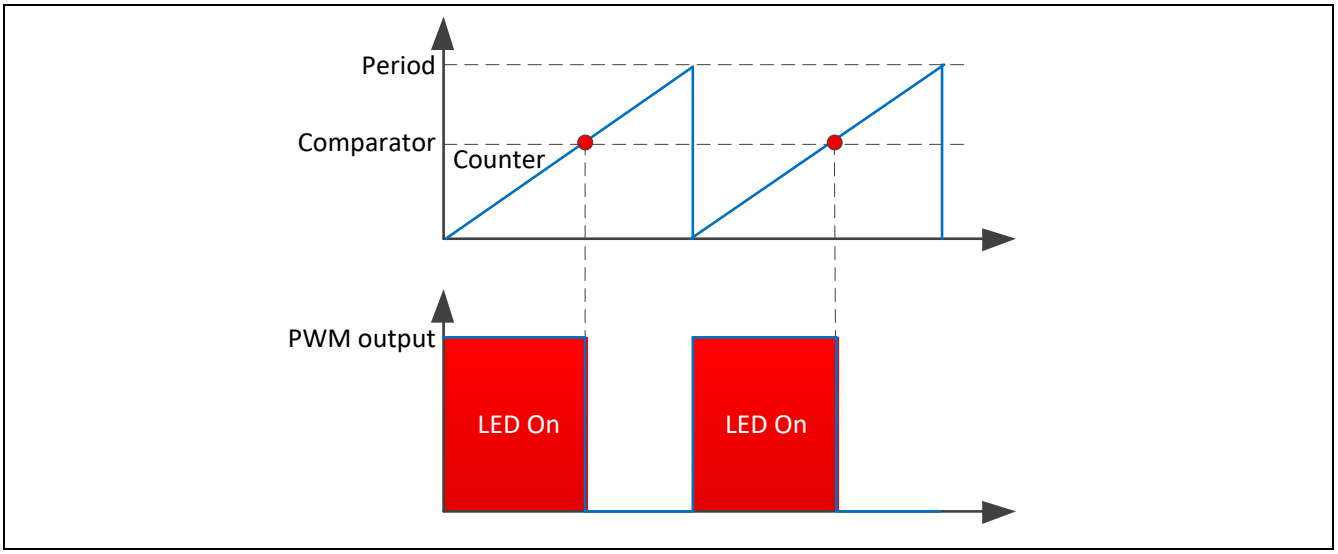

图 1-4 共阴极 RGB 的 PWM 和灯状态

1.1.2 UART 协议格式

输入命令格式如表 1-1 所示。其中位 0、位 4、位 8、位 12 为固定字符，必须与表格完全一致且格式必须刚好吻合 13 个字符。使用者必须严格按照格式正确输入命令才能成功设置参数到 PWM 寄存器。否则会弹出一条错误消息告知用户并保留原本的设定值不进行改变。各种命令格式分析及显示信息请参考表 1-2。

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

表 1-1 输入命令格式说明

情况	输入命令	状态	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 输入命令格式与状态说明

1.2 执行结果

1.2.1 PuTTY 设定步骤

本范例代码搭配“PuTTY”终端机应用程序，在 PC 端输入正确的 RGB LED 占空比格式用以实现 RGB 调光，PuTTY 设置步骤如下：

步骤 1：打开 PuTTY 应用程序并点选“Terminal”。

步骤 2：点选“Local echo”项目下的“Force on”以启用输入值显示，请参见图 1-5。

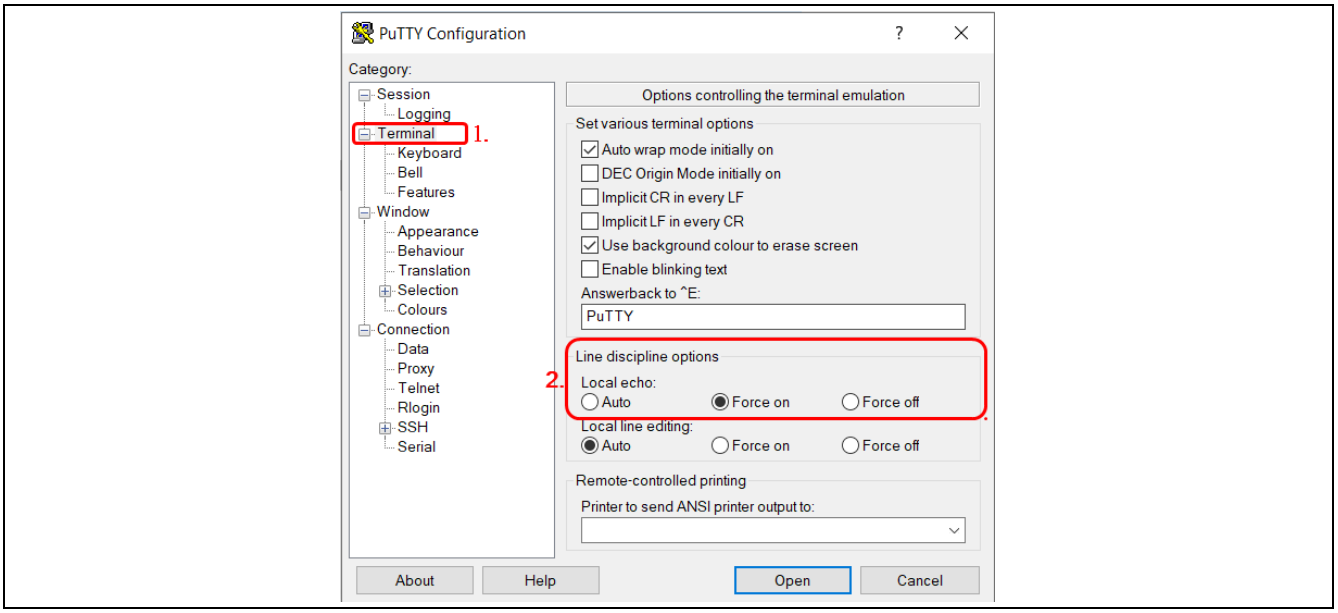

图 1-5 PuTTY 终端设置

步骤 3：点选“Session”。

步骤 4：点选“Serial line”下的“Serial”，然后选择端口号和波特率，请参考图 1-6。

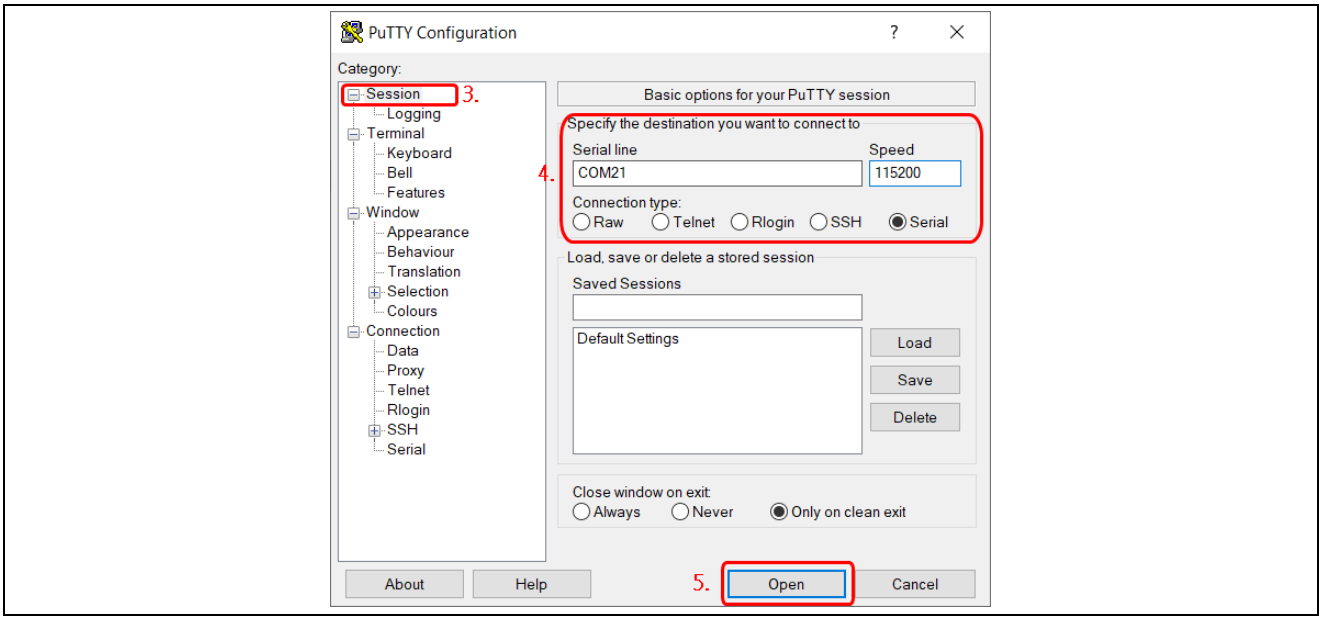

图 1-6 PuTTY 的 Session 设置

1.2.2 动作窗口

本范例代码执行编译和烧录的结果图 1-7 所示。只有输入正确的格式才能正确修改 PWM 占空比。另外，当用户开始输入第一个字符后，完整输入命令需要在 10 秒内完成，否则会判断为格式错误。

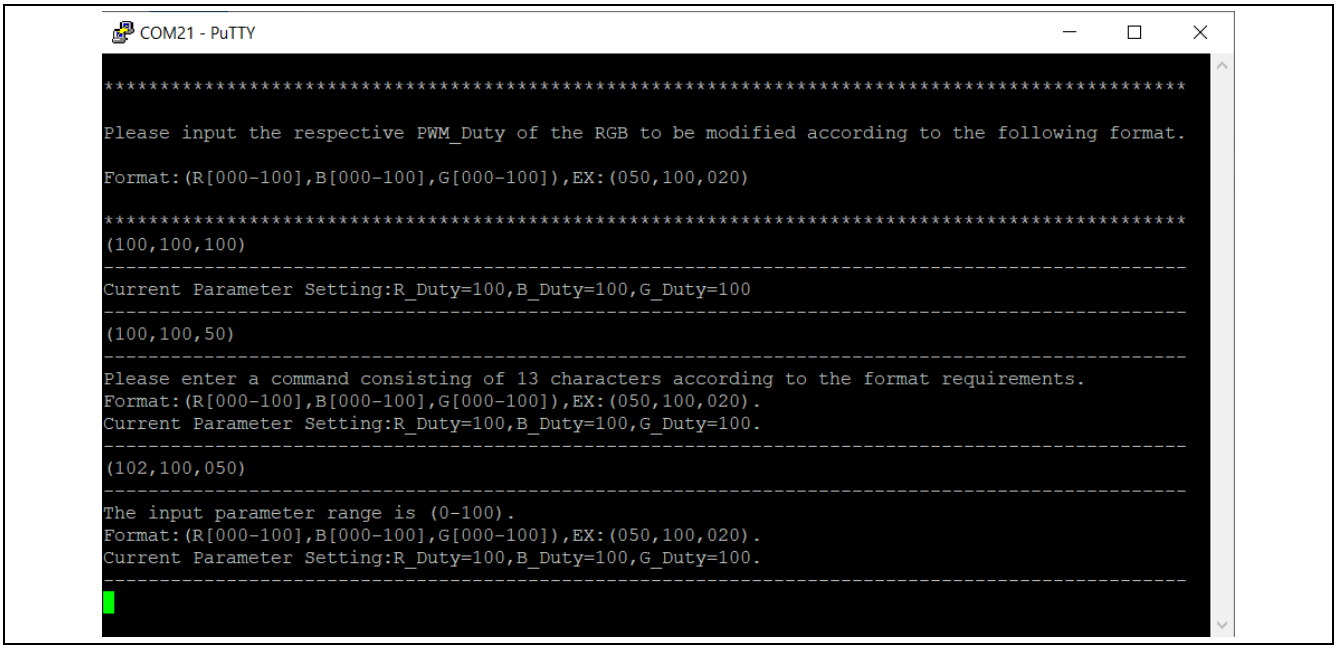

图 1-7 执行编译和烧录结果

2. 代码介绍

代码说明如下

1. 初始化 MS51 相关功能包括 HIRC、UART 和 Timer 0，代码位于 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. 当 UART 接收到一个命令字符时，UART Serial_ISR()函数将被触发。

```
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 负责对输入时间进行计数，如果输入时间过长，则判断为格式错误。

```
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. 格式错误信息，当输入格式错误时，将丢弃本次输入值并弹出格式错误信息。

```
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. 分析输入格式的正确性。如果正确，则更改 PWM 占空，否则将保持原始设置不做修改。

```
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. 软件与硬件需求

3.1 软件需求

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

3.2 硬件需求

- 电路组件
  - NuTiny-MS51FB V1.1
  - RGB LED
- 示意图

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

图 3-1 RGB LED 与 NuTiny-MS51FB V1.1 连接示意图

4. 目录信息

目录结构如下图所示

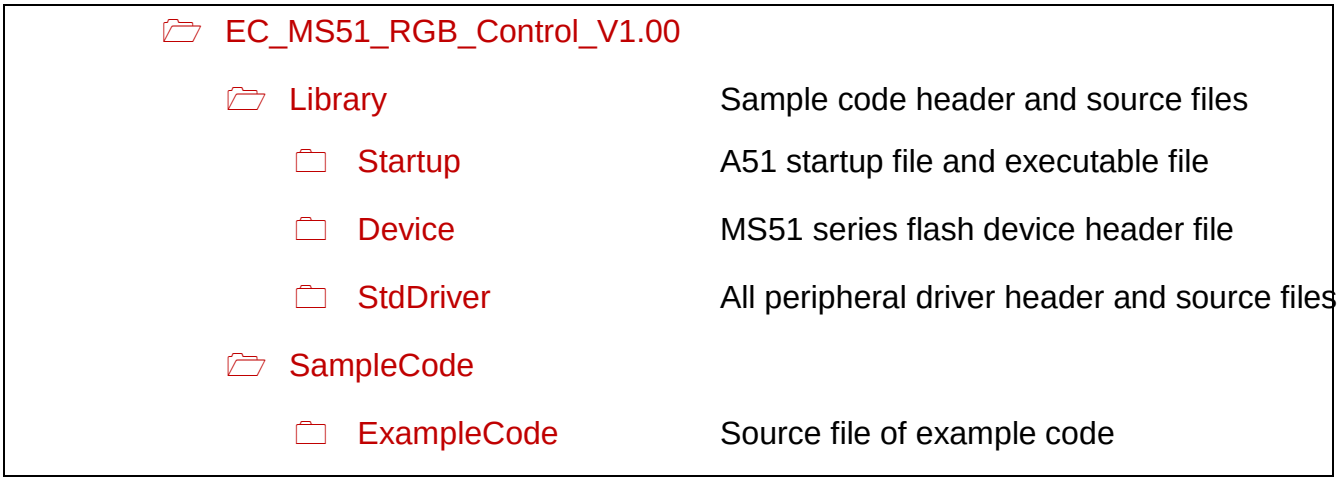

图 4-1 目录信息

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，鼠标右键点击两次 MS51_RGB_Control.uvproj。

2. 进入编译模式接口

   - 编译。
   - 下载代码至内存。

3. 打开 PuTTY

   - 按下复位按钮。
   - 按照格式(R[000-100],B[000-100],G[000-100]),EX:(050,100,020) 输入R、B、G对应的PWM参数。
   - 如果输入格式正确则修改PWM占空比，否则保持原参数不变。

6. 修订纪录

Date	Revision	Description
2022.07.06	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.

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.