M0A23 GPIO 實現 ARGB LED 跑馬燈效果

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 M0A23 GPIO 模擬 ARGB LED 的時序並實現跑馬燈效果
BSP 版本	M0A21_Series_BSP_CMSIS_V3.01.000
開發平台	Bling Bling Board Ver2.0

1. 概述

此範例代碼使用 M0A21/M0A23 系列的 GPIO 來模擬 ARGB 的時序，並演示 ARGB LED 跑馬燈效果。此範例代碼使用一組 Timer 來定時更新 ARGB LED 跑馬燈的顏色，使用一組 GPIO 來模擬 ARGB LED 操作，並一組 UART 用來輸入顯示文字。

1.1 原理

1.1.1 ARGB LED 資料傳輸時間(TH+TL=1200 ns ± 160 ns)

ARGB LED 時序圖如圖 1-1 邏輯與圖 1-2 邏輯所示。邏輯「0」定義為高準位 300ns ± 80ns 與低準位 900ns ± 80ns; 邏輯「1」定義為高準位 900 ns ± 80ns 與低準位 300ns ± 80ns。圖 1-3 重置時間需要大於 50 us。

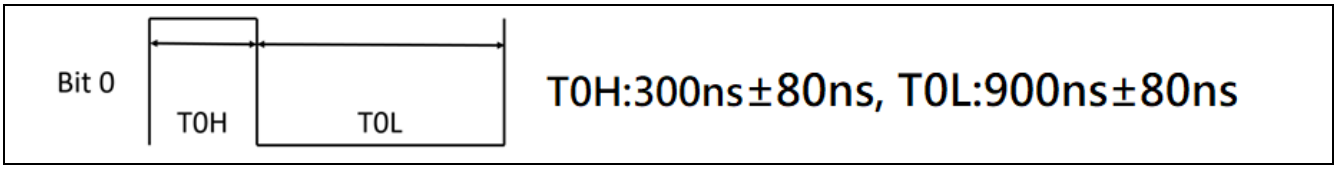

圖 1-1 邏輯「0」 (Bit 0)

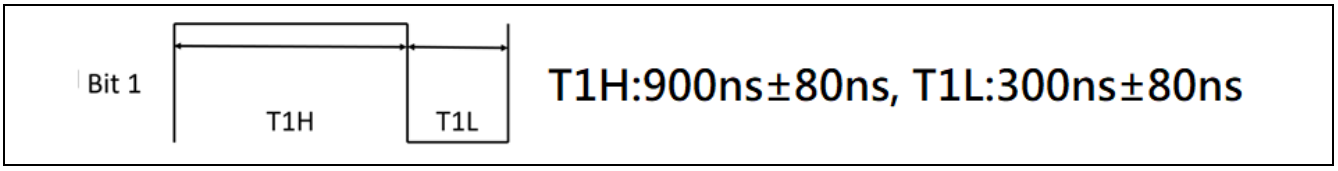

圖 1-2 邏輯「1」 (Bit 1)

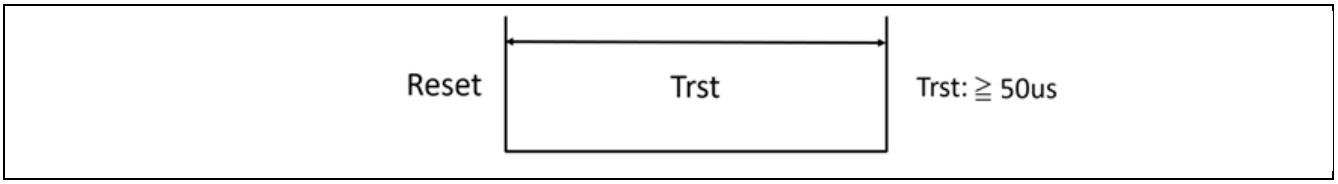

圖 1-3 重置 (Trst)

1.1.2 ARGB LED 特性介紹及連接方式

使用者只需一條信號線就可以控制 ARGB LED 顏色。與傳統的 RGB LED 相比，ARGB LED 更容易設計並節省 Layout 空間。連接方式如圖 1-4，從 MCU 的 GPIO 連接到 LED1 的輸入(DIN)，然後 LED1 輸出(DOUT)會串接到 LED2 的輸入(DIN)。

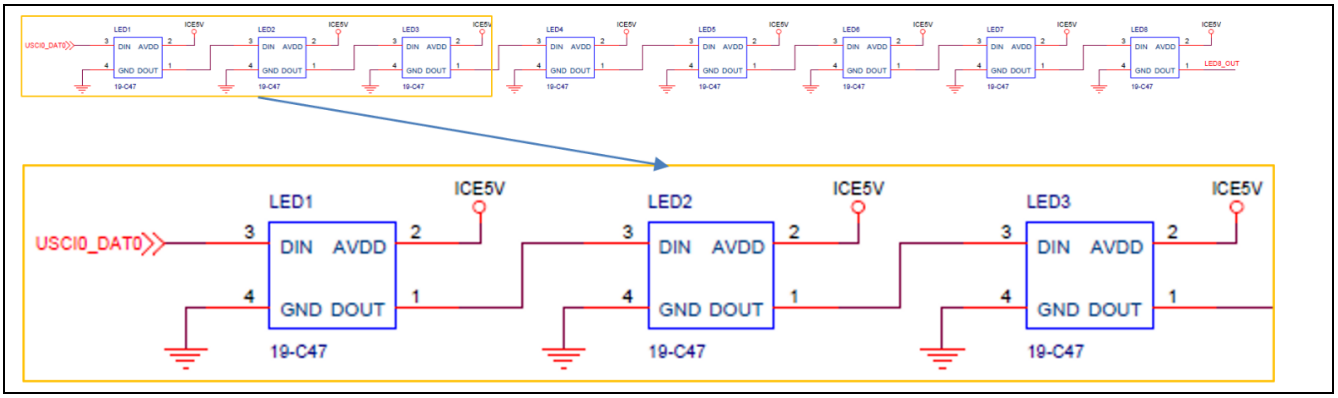

圖 1-4 ARGB LED 串接

1.1.3 GPIO 時鐘頻率與邏輯 Bit 設計

使用 GPIO 控制 ARGB LED，主要透過 NOP() 指令讓高低準位往後延遲一段時間，程式中設定系統時鐘 HIRC 為 48MHz，執行一個命令時間約 1 / 48MHz (20.83ns)，由執行命令時間即可得到需要執行幾次 NOP() 指令。

如圖 1-1 和圖 1-2，邏輯「0」定義為高準位 300ns ± 80ns(220ns≤TH0≤380ns) 與低準位 900ns ± 80ns(820ns≤ TL0≤980ns);邏輯「1」定義為高準位 900 ns ± 80ns (820ns≤TH1≤980ns)與低準位 300ns ± 80ns(220ns≤TL1≤380ns)。此範例代碼中邏輯「0」副程式為 ARGB_sendBIT0()，邏輯「1」副程式為 ARGB_sendBIT1()，邏輯「0」及邏輯「1」副程式中使用 NOP() 指令，執行一個 NOP()時間約 1 / 48MHz (20.83ns)。

由上述得知在設計時邏輯「0」高準位最少需要 220/20.83≈11NOP()，最多需要 380/20.83≈19NOP()。邏輯「1」低準位最少需要 820/20.83≈40NOP()，最多需要 980/20.83≈48NOP()。邏輯「0」高準位最少需要 820/20.83≈40NOP()，最多需要 980/20.83≈48NOP()。邏輯「1」低準位最少需要 220/20.83≈11NOP()，最多需要 380/20.83≈19NOP()，ARGB LED 的時鐘頻率可依上述條件設計。

如圖 1-5 邏輯「0」高準位為 280ns(280/20.83≈14NOP())，低準位為 980ns(980/20.83≈48NOP())，邏輯「1」高準位為 920ns(920/20.83≈45NOP())，低準位為 340ns(340/20.83≈17NOP())。

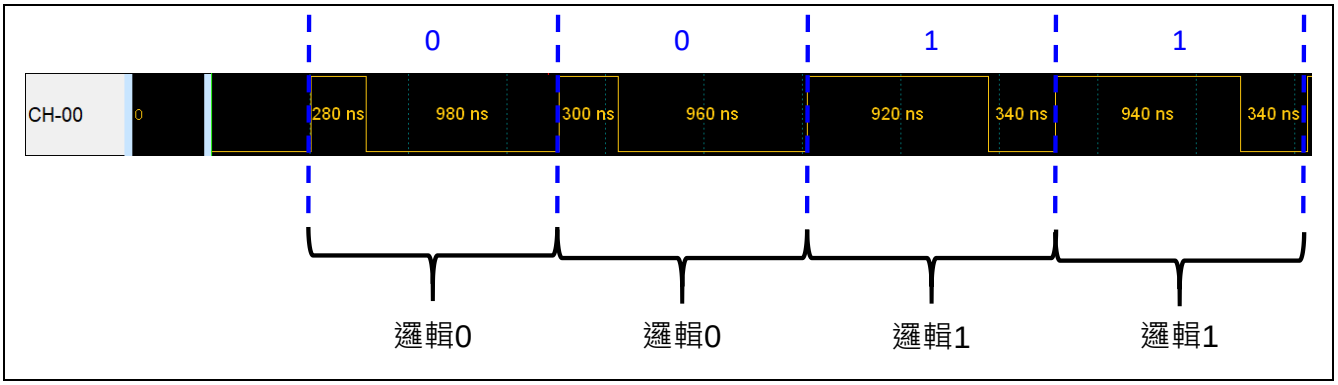

圖 1-5 使用 NOP 指令讓高低準位延遲一段時間

此範例代碼使用 Keil C，編譯時會將 C 語言轉換成組合語言，由於 C 語言的指令執行時間如迴圈或是指定變數值…等可能會需要耗數個 NOP()，故在實際設計程式中需要一些調整 NOP()數量。

此範例代碼 ARGB_sendOne()副程式中，用來判斷邏輯「0」和邏輯「1」，其中 ARGB_sendBIT0()為邏輯「0」副程式，ARGB_sendBIT1()為邏輯「1」副程式，在 ARGB_sendBIT0()副程式中高準位含有 11 個 NOP()，由於受到 ARGB_sendOne()副程式中的變數值指定及判斷式(if…else)及迴圈(for)影響執行中多出 7 個 NOP()，故邏輯「0」高準位會是(11+7=18)18 個 NOP()，如圖 1-6 實際邏輯「0」高準位 357ns(357/20.83≈18 NOP())。在 ARGB_sendBIT0()副程式中低準位含有 29 個 NOP()，由於受也到 ARGB_sendOne()副程式中的變數值指定及判斷式(if…else)及迴圈(for)影響執行中多出 14 個 NOP()，故邏輯「0」低準位會是(29+14=43)43 個 NOP()，如圖 1-7 實際邏輯「0」低準位 892ns(892/20.83≈43 NOP())。

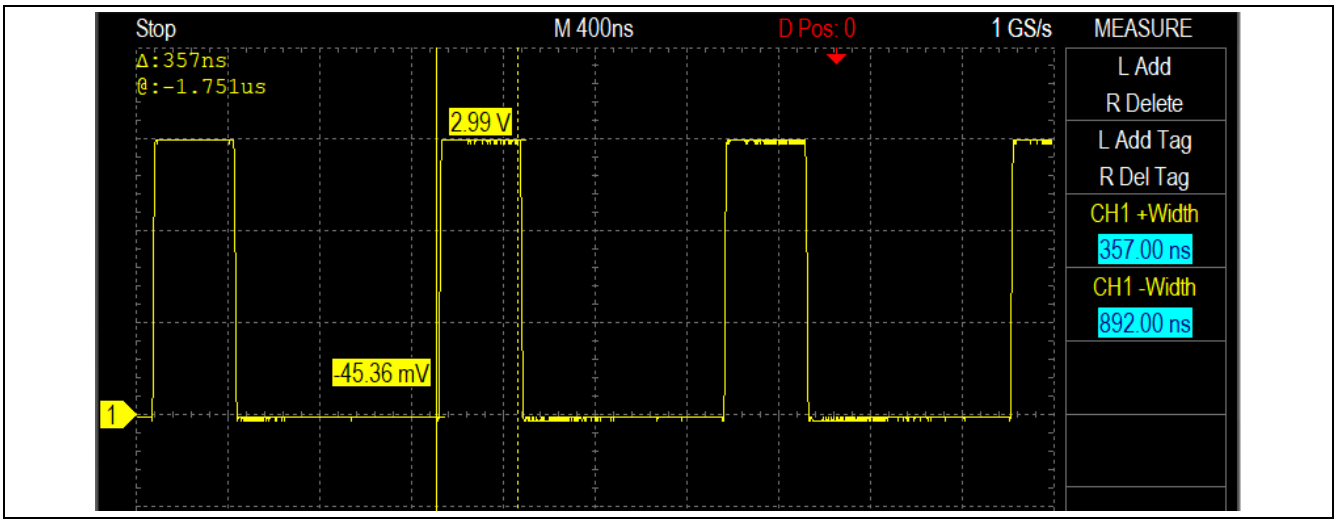

圖 1-6 邏輯「0」高準位波形

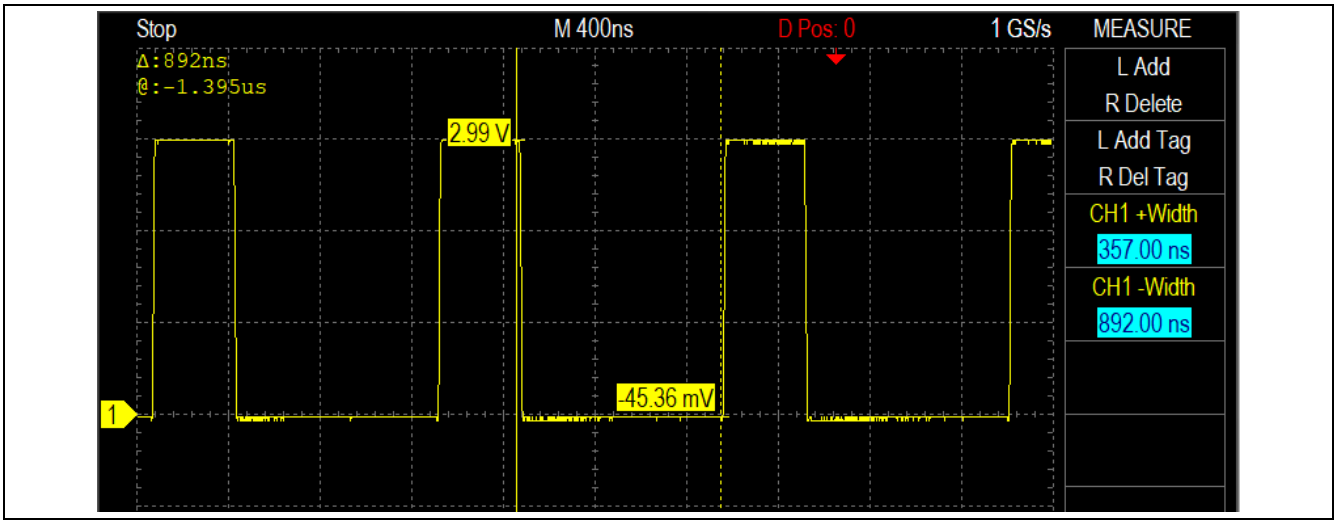

圖 1-7 邏輯「0」低準位波形

1.2 執行結果

1.2.1 邏輯「1」(Bit 1)、邏輯「0」 (Bit 0)、重置 (Trst) 執行結果

圖 1-8 邏輯「1」執行結果

圖 1-9 邏輯「0」執行結果

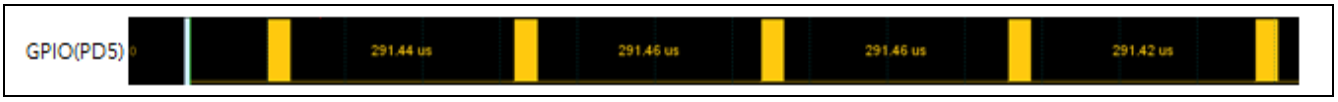

圖 1-10 重置 Trst 執行結果

1.2.2 終端機執行結果

使用終端機軟體，波率設定為 115200，終端機軟體執行結果如圖 1- 所示。

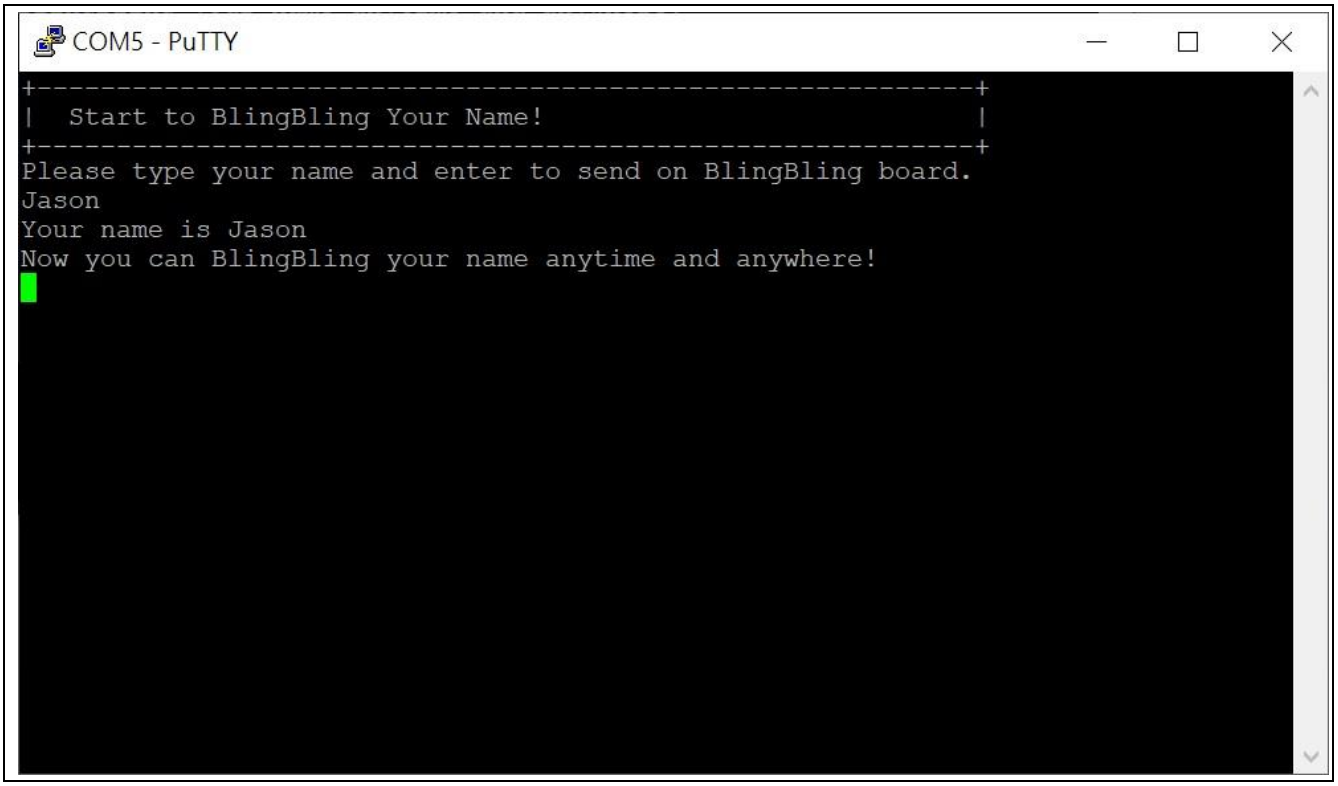

圖 1-11 終端機軟體顯示畫面

1.2.3 跑馬燈執行結果

ARGB LED 跑馬燈執行結果，如圖 1-所示。

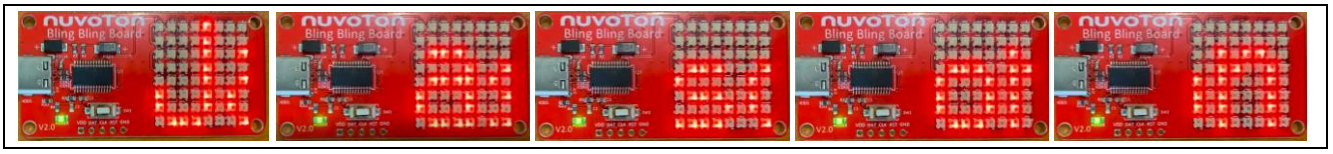

圖 1-12 ARGB LED 跑馬燈執行結果

2. 代碼介紹

2.1 主代碼說明

代碼工作流程如下：

1. *main.c* 中初始化 M0A23 系列相關外設，包含 HIRC、UART0 與 Timer0
2. 使用 ARGB_init() 來設定 GPIO 和模擬 ARGB LED 時序
3. 使用 Read_DataFlash() 來判斷 Data Flash 裡面是否已有資料
4. 在 While 迴圈判斷是否有新的字串輸入

```
int32_t main(void)
{
    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();

    /* Init UART0 for printf */
    UART0_Init();

    /* Init ARGB */
    ARGB_init();

    TIMER_Delay(TIMER0, 1000000);

    Read_DataFlash();

    Convert_FlashData();

    printf("+------------------------------------------------------------+\n");
    printf("|  Start to ""BlingBling"" Your Name!                        |\n");
    printf("+------------------------------------------------------------+\n");
    printf("Please type your name and enter to send on BlingBling board.\n");

    while(1)
    {
        if(receiveData == FALSE)
        {
            put_string(String);
        }
        else
        {
            Write_DataFlash();

            receiveData = FALSE;
        }
     }
}
```

當 Read_DataFlash() 被呼叫時，會去讀取指定位址的 Flash 資料。

```
void Read_DataFlash()
{
    SYS_UnlockReg();

    /* Enable FMC ISP function */
    FMC_Open();

    /* Enable Data Flash and set base address. */
    set_data_flash_base(DATA_FLASH_BASE);

    FMC_ENABLE_AP_UPDATE();

    flash_read(DATA_FLASH_BASE, DATA_FLASH_BASE + RX_BUFFER_SIZE);

    FMC_DISABLE_AP_UPDATE();

    FMC_Close();
}
```

當 Convert_FlashData() 被呼叫時，判斷字串字元陣列是否有資料

```
void Convert_FlashData()
{
    uint8_t i = 0;

    if((uint8_t)flashData[0] != 0xFF)
    {
        for(i = 0; i < RX_BUFFER_SIZE; i++)
        {
            String[i] = 0;
        }

        for(i = 0; i < RX_BUFFER_SIZE; i++)
        {
            if((uint8_t)flashData[i] != 0xFF)
                String[i] = flashData[i];
            else
                break;
        }

        for(i = 0; i < RX_BUFFER_SIZE; i++)
        {
            flashData[i] = 0;
        }
    }
}
```

當 put_string(char inputString[]) 被呼叫時，會將字串存放到走馬燈的陣列。

```
void put_string(char inputString[])
{
    int8_t count;

    for(count = 0; count < strlen(inputString); count++)
    {
        select_word(inputString[count]);

        put_char(count, strlen(inputString));
    }
```

```
    if(showArray_column >8)
        Run_column = showArray_column;
    else
        Run_column = 8;

    ARGB_showString();

    TIMER_Delay(TIMER0, 1000000);

    color++;

    if(color == 3)
    {
        color = 0;
    }

    showArray_row = 0;
    showArray_column = 0;
}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M0A21_Series_BSP_CMSIS_V3.01.000
- IDE 版本
  - Keil uVersion 5.36

3.2 硬體需求

- 電路元件
  - Bling Bling Board Ver2.0
  - ARGB LED
- 線路示意圖

使用 GPIO 腳位來模擬 ARGB LED 時序圖。

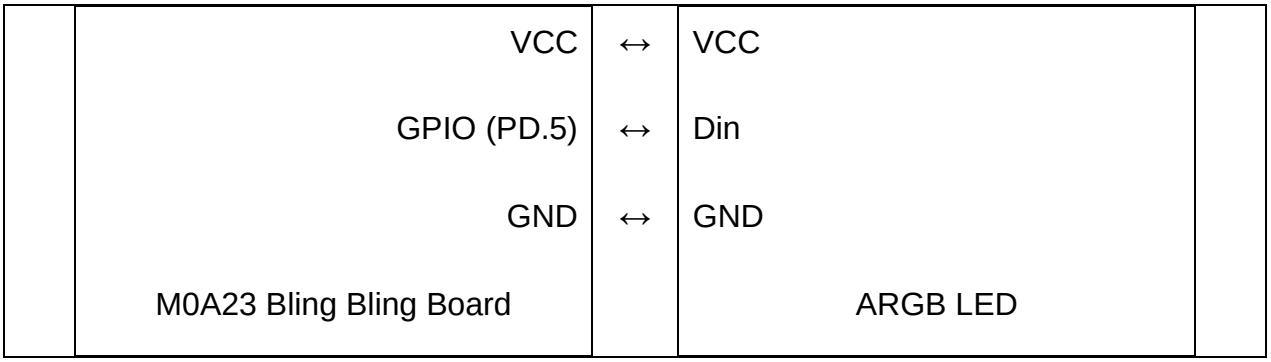

圖 3-1 線路示意圖

4. 目錄資訊

EC_M0A23_GPIO_ARGB_LED_Control_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	
BlingBling	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 *BlingBling.uvprojx*。
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
2023.12.15	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.