Using MS51 SPI to Realize Breathing ARGB LED Effect

Example Code Introduction for 8-bit NuMicro® Family

Information

Application	This sample code uses SPI to simulate ARGB LED timing sequence and realize breathing ARGB LED effect on MS51 series.
BSP Version	MS51 Series BSP V1.00.003
Hardware	NuMaker-MS51PC Ver1.0

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 Overview

This sample code uses SPI to simulate ARGB LED timing sequence and do LED breathing effect. This code uses one timer to schedule ARGB LED to do ARGB LED dimming and uses one set of SPI to simulate ARGB LED strip timing sequence.

1.1 Principle

1.1.1 ARGB LED Data Transfer Time (TH+TL=1.25 us ± 600 ns)

The ARGB LED timing sequences are shown in Figure 1-1 to Figure 1-3. Logic 0 is defined as a high pulse of 0.4us ± 150ns followed by a low pulse of 0.85us ± 150ns. Logic 1 is defined as a high pulse of 0.8us ± 150ns followed by a low pulse of 0.45us ± 150ns. Reset is defined as a low pulse of above 50 us.

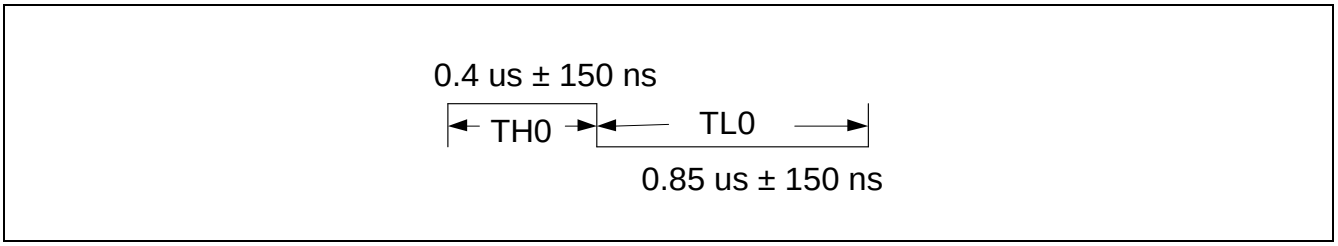

Figure 1-1 Logic 0

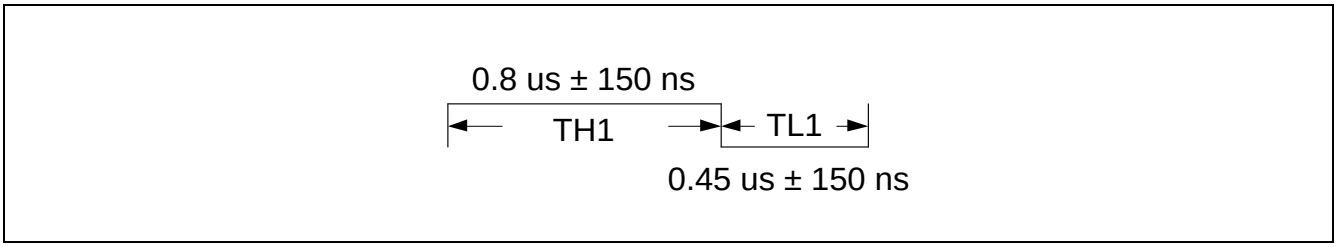

Figure 1-2 Logic 1

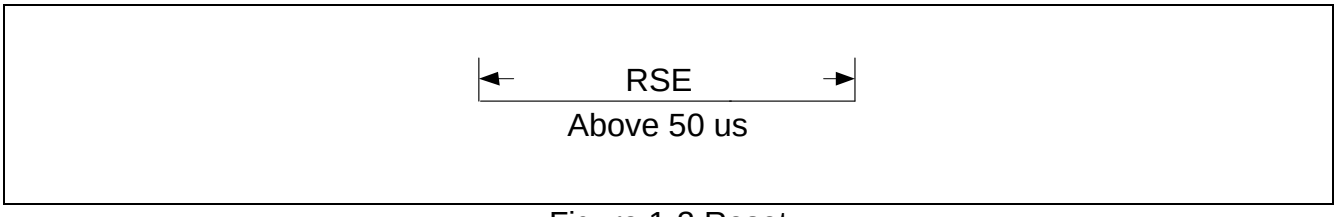

Figure 1-3 Reset

1.1.2 SPI Clock Frequency

When SPI clock divider is 8 MHz and HIRC clock is 24 MHz, the theoretical for the SPI clock is 3 MHz. It means that one SPI bit lasts 0.333us. In other words, logic 0 is mapped to SPI bit pattern of ‘1000’ as shown in Figure 1-4 and logic 1 is mapped to SPI bit pattern of ‘1110’. The 3-byte Green-Red-Blue value is converted to 12-byte SPI sequence.

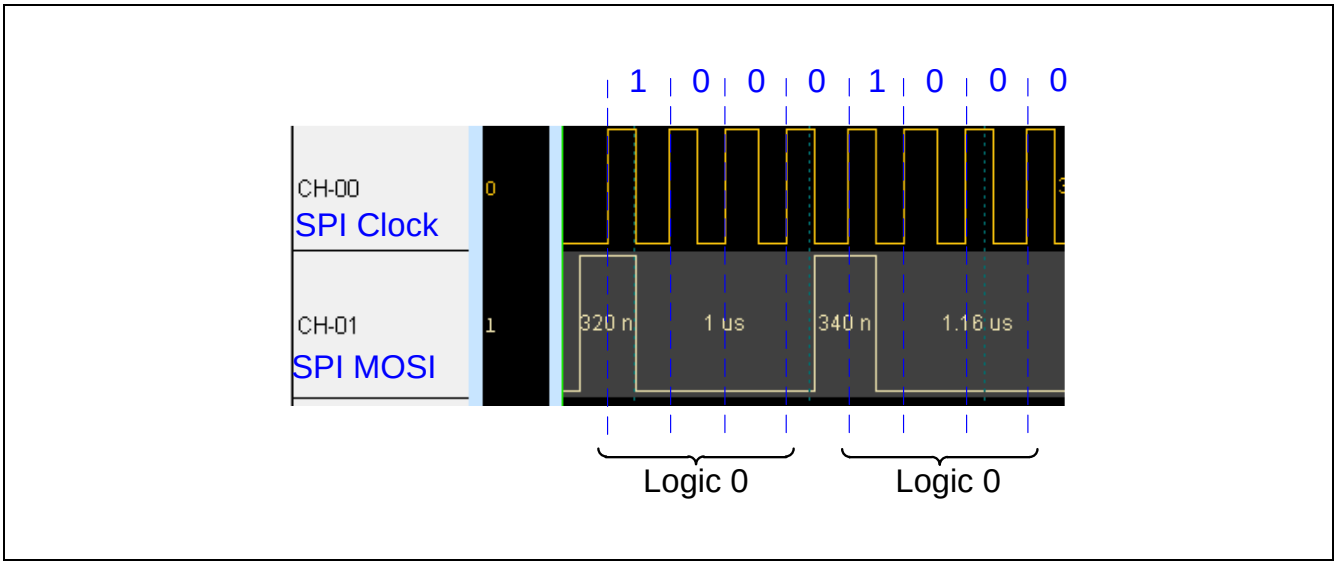

Figure 1-4 Logic 0 Is Mapped to SPI Bit Pattern

1.2 Demo Result

The demo result is shown in Figure 1-5.

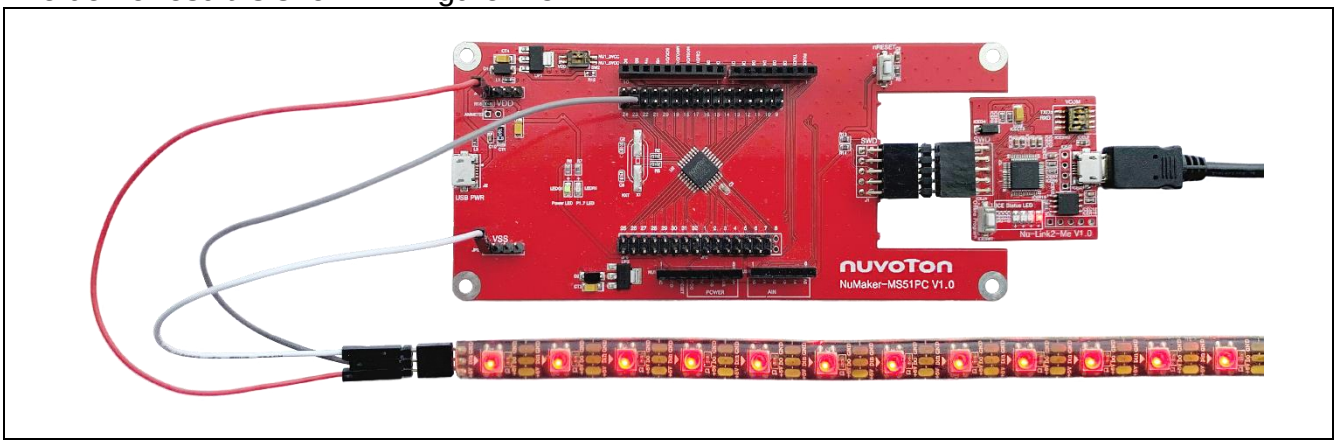

Figure 1-5 Demo Result

2 Code Description

The instructions are listed below:

1. Initialize the MS51 in main.c, including HIRC, SPI, and Timer 3.
2. In while-loop, call Breathing_TOPDOWN() function to calculate brightness.
3. Use SPI_Send() function to control ARGB LED.

```
void main (void)
{
    /* Set HIRC = 24 MHz */
    MODIFY_HIRC(HIRC_24);

    P00 = 1;
    P00_INPUT_MODE;
    ENABLE_P00_PULLDOWN;

    /* Set SPI configure */
    SPI_Init();

    /* Enable Timer3 to add time counter */
    Timer3_1ms_initial();

    /* Init LED to all disable */
    Disable_TOPDOWN();
    SPI_Send();

    while(1)
    {

        /* Control LED */
        if (!spi_finish_flag)
        {
            Breathing_TOPDOWN();
            SPI_Send();
        }
    }
}
```

When Breathing_TOPDOWN() function is called, the ARGB LED brightness will be calculated periodically. The parametric counter-brightness for controlling ARGB LED is shown in Table 2-1.

```
void Breathing_TOPDOWN(void)
{
    unsigned char TempR, TempG, TempB;
    unsigned long TempCal;

    /* Calculate brightness */
    TempCal = *(BreathingBright + (HDIV_Mod(HDIV_Div(TOPDOWNTimeCounter,
(BreathingRate[TOPDOWNSpeed]) * 8), BreathingArraySize)));
    TempCal = HDIV_Div((TempCal * TOPDOWNBrightness), 100);

    /* Calculate color and set transfer data */
    TempR = HDIV_Div((TOPDOWNRGBColor[0] * TempCal), 100);
    TempG = HDIV_Div((TOPDOWNRGBColor[1] * TempCal), 100);
    TempB = HDIV_Div((TOPDOWNRGBColor[2] * TempCal), 100);
```

```
    Set_TOPDOWN_RGB(TempR, TempG, TempB);

    /* Reset TOPDOWNTimeCounter */
    if(HDIV_Div(TOPDOWNTimeCounter, BreathingRate[TOPDOWNSpeed] * 8 * BreathingArraySize))
    {
        TOPDOWNTimeCounter -= BreathingRate[TOPDOWNSpeed] * 8 * BreathingArraySize;
    }
}
```

TOPDOWNTimeCounter	TempCal	TempR	TempG	TempB
0	0	0	0	0
…	…	…	…	…
48	0	0	0	0
…	…	…	…	…
96	1	2	0	0
…	…	…	…	…
144	2	5	0	0
…	…	…	…	…

Table 2-1 Example of Relationship between Counter and Brightness

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - MS51 Series BSP v1.00.003
- IDE version
  - Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
  - NuMaker-MS51PC Ver1.0
  - ARGB LED

- Pin Connect

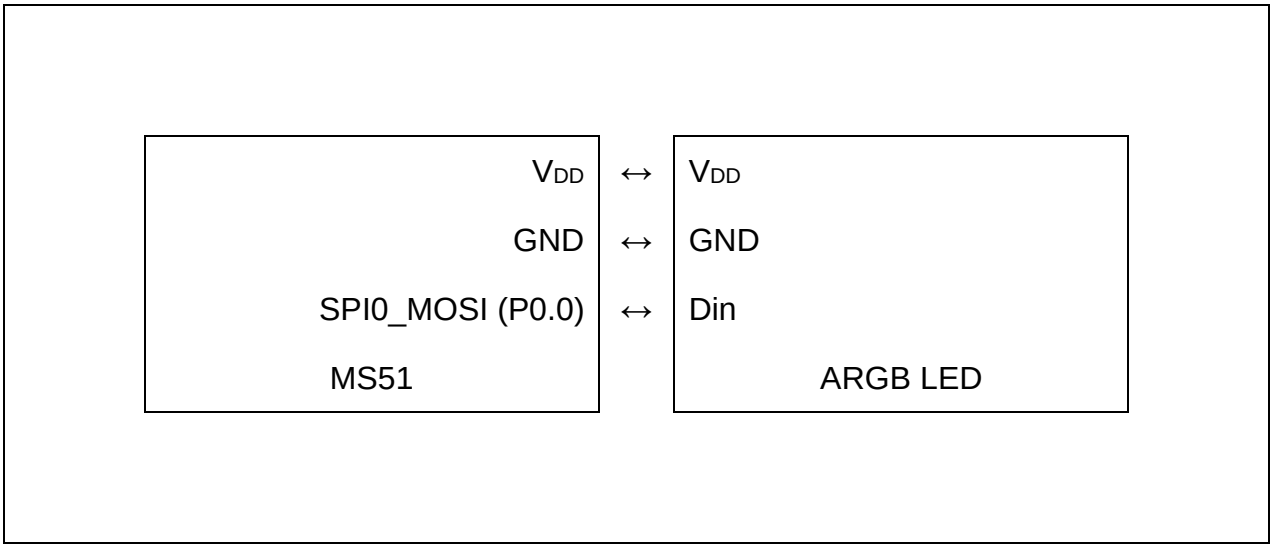

Figure 3-1 Pin Connect

Note: The operating voltage of ARGB LED is 5V, so the power supply voltage of NuMaker-MS51PC for the ARGB LED must be changed to 5V. Please solder 0Ω between ICEJPR1's MCUVCC and 5V that can make Nu-Link2-Me output 5V voltage as shown in Figure 3-2.

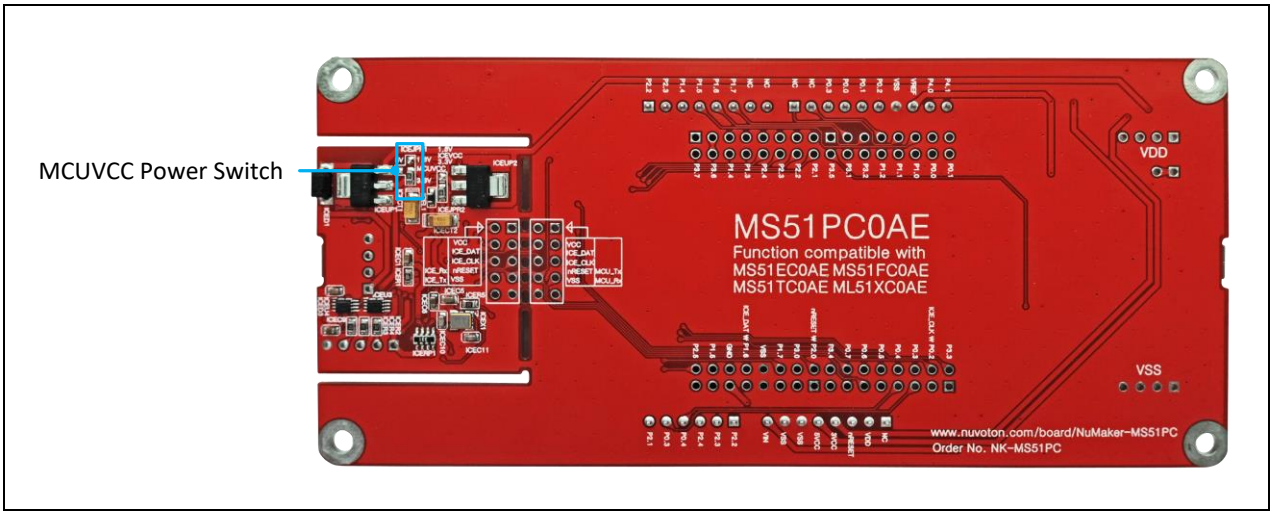

Figure 3-2 5V Power Source

4 Directory Information

The directory structure is shown in Figure 4-1.

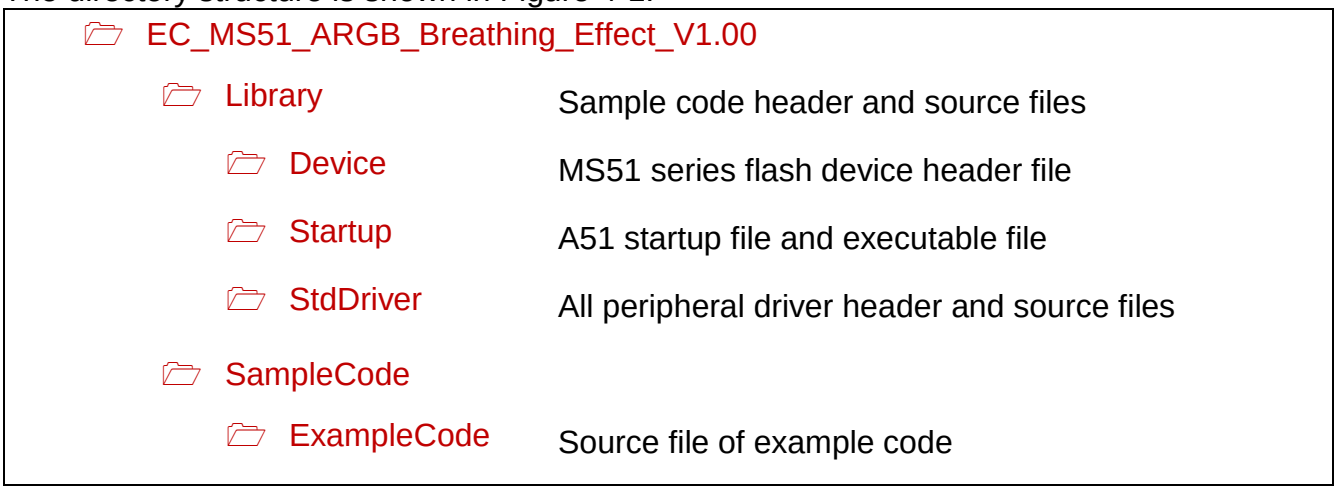

Figure 4-1 Directory Structure

5 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *MS51_ARGB_Breathing_Effect.uvproj*.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6 Revision History

Date	Revision	Description
2021.07.29	1.00	1. Initially issued.

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.