SPI interface LCD Panel Control – ILI9341

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example code based on NUC126 for implementing the the SPI interface LCD Panel – ILI9341.
BSP Version	NUC126_Series_BSP_CMSIS_v3.00.004
Hardware	NuTiny-NUC126

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 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 Function Description

1.1 Introduction

In this example code, the NuMicro NUC126 series is used as an example to implement the SPI interface LCD Panel – ILI9341. ILI9341 is a driver IC for controlled LCD panel. Currently, it's usually made as a module for sale. User needs to know how to control the ILI9341 for driving the LCD panel. For more detailed ILI9341 information and instructions, please refers to the ILITEK website for downloaded the datasheet.

2 Code Description

2.1 Initialize the PWM Module

Initialize the PWM module for tuning the panel brightness. Frequency is 1 kHz and the PWM duty is setting on 50%.

```
static void _InitPWMModule(void)
{
    /* Using PWM1 channel 5 (PC14) for tunning back light brightness */
    SYS->GPC_MFPH &= ~(SYS_GPC_MFPH_PC14MFP_Msk);
    SYS->GPC_MFPH |= SYS_GPC_MFPH_PC14MFP_PWM1_CH5;

    /* Select and enable module clock */
    CLK_EnableModuleClock(PWM1_MODULE);
    CLK_SetModuleClock(PWM1_MODULE, CLK_CLKSEL1_PWM1SEL_PCLK1, NULL);

    /* Reset PWM1 module */
    SYS_ResetModule(PWM1_RST);

    /* PWM1 channel 5 frequency is 1000Hz, duty 50%, */
    PWM_ConfigOutputChannel(PWM1, 5, 1000, 50);

    /* Enable output of PWM1 channel 5 */
    PWM_EnableOutput(PWM1, PWM_CH_5_MASK);

    /* Start PWM counter */
    PWM_Start(PWM1, PWM_CH_5_MASK);
}
```

2.2 Initialize the SPI Module

Initialize the SPI module for driving the LCD panel. The SPI module sets as master, 8 bit data width and operates on 24 MHz.

```
static void _InitSPIModule(void)
{
    /* Using SPI1 for displaying the LCD */
    /* Switch the multi-function pin to SPI */
    SYS->GPD_MFPH &= ~(SYS_GPD_MFPH_PD13MFP_Msk | SYS_GPD_MFPH_PD14MFP_Msk |
SYS_GPD_MFPH_PD15MFP_Msk);
    SYS->GPD_MFPH |= (SYS_GPD_MFPH_PD13MFP_SPI1_MOSI | SYS_GPD_MFPH_PD14MFP_SPI1_MISO |
SYS_GPD_MFPH_PD15MFP_SPI1_CLK);
```

```
    /* Select and enable module clock */
    CLK_EnableModuleClock(SPI1_MODULE);
    CLK_SetModuleClock(SPI1_MODULE, CLK_CLKSEL2_SPI1SEL_PLL, 0);

    /* SPI open on master mode, mode 0, 8bit data width and operate on 24M Hz */
    SPI_Open(SPI_LCD_PORT, SPI_MASTER, SPI_MODE_0, 8, 24000000);

    /* Disable auto SS function */
    SPI_DisableAutoSS(SPI_LCD_PORT);

    /* Enable SPI */
    SPI_ENABLE(SPI_LCD_PORT);

}
```

2.3 Initialize the MCU Hardware for LCD Panel

In this function, it initializes the SPI and PWM module and set PC14 to output mode as LCD backlight switch.

```
static void _InitMCUHWforLCD(void)
{
    /* Initial SPI Module for LCD display */
    _InitSPIModule();

    /* Initial PWM for backlight brightness control */
    _InitPWMModule();

    /* Set PC14 to output mode for LCD backlight on/off switch */
    GPIO_SetMode(PC, BIT14, GPIO_MODE_OUTPUT);
}
```

2.4 Initialize the NuMaker Hardware

Initialize the MCU hardware and ILI9341 driver IC. For detailed information of ILI9341 register, please refer to the ILI9341 datasheet.

```
void Drv_NuMaker_HWInit(void)
{
    /* Initial NuMaker Hardware */
    /* Initial MCU Hardware */
    _InitMCUHWforLCD();

    /* Initial ILI9341 LCD Panel Hardware */
    Drv_ILI9341_LCD_Init();
```

```
}
```

2.5 Main Loop

The main loop will do system initialize, MCU hardware initialize, and ILI9341(panel driver IC) initialize. After initializing, the while loop will display the yellow, green, and red color on whole screen and switch color every 200 ms.

```
int32_t main(void)
{
    uint32_t i = 0;
    /* Initial MCU System */
    SYS_Init();
    /* Initial NuMaker Hardware */
    Drv_NuMaker_HWInit();

    while (1)
    {
        /* Display color yellow */
        for (i = 0; i < (LCD_Resolution_X * LCD_Resolution_Y * 2); i++)
        {
            LCD_Set_Color(Yellow);
        }

        CLK_SysTickDelay(200000);

        /* Display color Green */
        for (i = 0; i < (LCD_Resolution_X * LCD_Resolution_Y * 2); i++)
        {
            LCD_Set_Color(Green);
        }

        CLK_SysTickDelay(200000);

        /* Display color Red */
        for (i = 0; i < (LCD_Resolution_X * LCD_Resolution_Y * 2); i++)
        {
            LCD_Set_Color(Red);
        }

        CLK_SysTickDelay(200000);
    }
}
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - NUC126_Series_BSP_CMSIS_v3.00.004
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-NUC126 * 1
  - Nu-Bridge * 1
  - Ili9341 LCD Panel

■ Diagram

1. Follow the diagram to connect the ILI9341 LCD Panel, Nu-Bridge and NUC126.

2. For printing the debug message, user needs to connect two wire, the UART_RX0(PD.0)、UART_TX0(PD.1) of NUC126 to Nu-Bridge. Then set the COM Port number and Baudrate in terminal tool. User can find the COM Port number in Device Manager of PC and it will display “NuBridge Virtual Com Port (COMX)”. The Baudrate should be 115200.

UART RX	↔	UART_TX (PB.1)
UART TX	↔	UART_RX (PB.0)
GND		GND
Nu-Bridge		

LEDK1~K4	↔	PC.14
LEDA	↔	3VCC(3.3V)
3VCC	↔	3VCC(3.3V)
GND	↔	GND
LCM_SPI_MISO	↔	PD.14
LCM_SPI_MOSI	↔	PD.13
LCM_SPI_SS	↔	PD.12
LCM_DC	↔	PD.6
LCM_SPI_CLK	↔	PD.15
LCM_RESET	↔	RESET
ILI9341		NUC126

4 Directory Information

- EC_NUC126_SPI_LCD_Ili9341_Readme_EN_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

5 How to Execute Example Code

1. This project supports Keil uVersion 5.26 and higher version
2. According to the chapter 4 Directory Information, enter the path EC_NUC126_SPI_LCD_Ili9341_Readme_EN_V1.00\SampleCode\ExampleCode\NUC126_SPI_LCD_ILI9341 folder and double click the SPI_LCD_Ili9341.uvproj
3. Enter the Keil compile mode window
    a. Build the project
    b. Download
    c. Start / Stop the debug mode
4. Enter debug mode
    a. Run

6 Revision History

Date	Revision	Description
Nov. 20, 2019	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.