使用NUC240控制LCD1602显示屏

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	本范例代码基于NUC240平台，使用GPIO功能实现LCD1602显示屏的8位模式控制
BSP 版本	NUC230_240 Series BSP CMSIS V3.01.001
开发平台	NuTiny-NUC240V

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 功能介绍

1.1 简介

本范例代码基于NUC240平台，使用GPIO功能实现LCD1602显示屏的8位模式控制。
SMC1602是一个LCD显示屏，型号1602的意思是每行显示16个字符，共可显示两行，驱动的形式上有8位及4位两种可供选择，显示屏详细规格与指令内容请参考SMC1602的Datasheet。

1.2 SMC1602 引脚说明

引脚编号	符号	说明	引脚编号	符号	说明
1	VSS	电源地	**9**	D3	LCD DB2
2	VDD	电源正极	**10**	D4	LCD DB3
3	VL	液晶显示偏压信号	**11**	D5	LCD DB4
4	RS	缓存器选择 RS = 0 : Cmd register RS = 1 : Data register	**12**	D6	LCD DB5
5	R/W	LCD 读写选择 (Read/Write) R/W = 0 : Write R/W = 1 : Read	**13**	D7	LCD DB6
6	E	LCD 使能(Enable) E = 0 : LCD disable E = 1 : LCD enable	**14**	D8	LCD DB7
7	D0	LCD DB0 (Data Bus)	**15**	BLA	背光源正极
8	D1	LCD DB1	**16**	BLK	背光源负极

2 代码介绍

2.1 函式描述

程序代码中的批注，说明了GPIO的设定、宏的定义，详细内容请参考批注。

```
/*---------------------------------------------------------------------------------------*/
/* Function Description                                                                  */
/*---------------------------------------  ----------------------------------------------*/
// GPIOE.0 -> LCD RS(Register Select)
//                RS = 0 : Cmd register
//                RS = 1 : Data register
// GPIOE.1 -> LCD R/W(Read/Write)
//                R/W = 0 : Write
//                R/W = 1 : Read
// GPIOE.2 -> LCD E(Enable)
//                E = 0 : LCD disable
//                E = 1 : LCD enable
// GPIOA.0 -> LCD DB0(Data Bus)
// GPIOA.1 -> LCD DB1
// GPIOA.2 -> LCD DB2
// GPIOA.3 -> LCD DB3
// GPIOA.4 -> LCD DB4
// GPIOA.5 -> LCD DB5
// GPIOA.6 -> LCD DB6
// GPIOA.7 -> LCD DB7

#define LCD_DATA    PA->DOUT // GPIOA data out
#define LCD_RS_SET  PB0=1 // Data register
#define LCD_RS_CLR  PB0=0 // Cmd register
#define LCD_RW_SET  PB1=1 // Write
#define LCD_RW_CLR  PB1=0 // Read
#define LCD_E_SET   PB2=1 // LCD disable
#define LCD_E_CLR   PB2=0 // LCD enable
```

2.2 写入数据至 LCD 模块

使用此函数能够将欲显示的数据写入LCD模块，详细内容请参考下列批注。

```
void WriteDataLcd(unsigned char wdata)
{
    LCD_DATA = wdata;       // Set data out
    LCD_RS_SET;             // Select Data register
    LCD_RW_CLR;             // LCD1602 Read
    LCD_E_CLR;              // LCD disable
    CLK_SysTickDelay(1000); // Delay 1000 us

    LCD_E_SET;              // LCD enable
}
```

2.3 写入命令至 LCD 模块

使用此函数能够将欲显示的命令写入LCD模块，详细内容请参考下列批注。

```
void WriteCommandLcd(unsigned char wdata)
{
    LCD_DATA = wdata;       // Set data out
    LCD_RS_CLR;             // Select Cmd register
    LCD_RW_CLR;             // LCD1602 Read
    LCD_E_CLR;              // LCD disable
    CLK_SysTickDelay(1000); // Delay 1000 us

    LCD_E_SET;              // LCD enable
}
```

2.4 初始化 LCD 模块

本函数发送了简单的命令对LCD模块进行初始化，详细内容请参考下列批注。

```
void lcd_init(void)
{
    LCD_DATA = 0;
    WriteCommandLcd(0x38);  //DL=8-bit, N=2-line, F=5x8 dots
    CLK_SysTickDelay(50000);
    WriteCommandLcd(0x08);  //D=0 Display Off, C=0 Cursor Off, B=0 Blinks Off.
    WriteCommandLcd(0x06);  //I/D=1 Increment, S=0 Display does not shift.
    WriteCommandLcd(0x0C);  //D=0 Display On, C=0 Cursor Off, B=0 Blinks Off.
}
```

2.5 LCD 模块写入行选择

依照欲选择的写入行，发送指令给LCD进行设定。

```
void display_xy(unsigned char x, unsigned char y)
{
    x += 0x80;              // For 1st line
    if (y == 1)
        x += 0x40;          // For 2nd line
    WriteCommandLcd(x);
}
```

2.6 写入字符

写入欲显示的字符至LCD模块。

```
void display_char(unsigned char x, unsigned char y, unsigned char dat)
{
    display_xy(x, y);  // Select line 1 or line 2
    WriteDataLcd(dat); // Write the data to LCD
}
```

2.7 写入字符串

写入欲显示的字符串至LCD模块。

```
void display_string(unsigned  char x, unsigned  char y, unsigned  char *string)
{
    display_xy(x, y);       // Select line 1 or line 2

    while (*string)         // If there is not end of string, sending data continuously.
    {
        WriteDataLcd(*string);  // Write data to LCD
        string++;               // Move the pointer to next character
    }
}
```

2.8 主循环

设定GPIOA、初始化LCD模块，输入两个字符串进行显示。

```
unsigned char  LcdBuf1[] = {"Welcome!"};
unsigned char  LcdBuf2[] = {"NuMicro M0"};

int main(void)
{
    GPIO_SetMode(PA, 0xFFFF, GPIO_PMD_OUTPUT);  // Set PA0 ~ PA15 to output mode.
```

```
    lcd_init();                                 // Initial LED module.
    display_string(2, 0, LcdBuf1);              // Display string 1 "Welcome!"
    display_string(0, 1, LcdBuf2);              // Display string 2 "NuMicro M0"

    while (1);
}
```

3 软件与硬件环境

- 软件环境
  - BSP 版本
    - NUC230_240 Series BSP CMSIS V3.01.001
  - IDE 版本
    - Keil uVersion 5.26
- 硬件环境
  - 电路组件
    - NuTiny-NUC240V * 1
    - 牛桥 Nu-Bridge * 1
    - SMC1602 * 1

■ 接线图

1. 将NuTiny-NUC240V的UART_RX(PB.0)、UART_TX (PB.1)连接至Nu-Bridge，打印讯息。设置终端机的COM Port与Baudrate，COM Port的编号可在设备管理器中找到「NuBridge Virtual Com Port (COMX)」，Baudrate设置为115200。

2. 将SMC1602依照以下连接方式，与NUC240连接。

UART RX	↔	UART_TX (PB.1)
UART TX	↔	UART_RX (PB.0)
Nu-Bridge		
RS	↔	PE.0
R/W	↔	PE.1
E	↔	PE.2
D0	↔	PA.0
D1	↔	PA.1
D2	↔	PA.2
D3	↔	PA.3
D4	↔	PA.4
D5	↔	PA.5
D6	↔	PA.6
D7	↔	PA.7
SMC1602		NUC240

4 目录信息

- EC_NUC240_LCD1602_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 如何执行范例程序

1. 根据目录信息章节进入 EC_NUC240_LCD1602_V1.00\SampleCode\ExampleCode\1602 LCD 8BIT，双击路径中 1602 LCD 8BIT.uvproj 开启专案。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Nov. 11, 2019	1.00	1. 初版

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.