上海电力大学

《嵌入式系统及应用》

期末大作业报告

课题名称：STM32 多功能计算器

院（系）：电子与信息工程学院

专　　业：集成电路设计与集成系统

班　　级：2021391 班

姓　　名：

学　　号：

一、设计要求

通过触摸屏实现一个简易的计算器，具有完整的多位加减乘除功能。

二、设计思路

1. 在 LCD 屏幕显示计算器的界面设计。
2. 在 LCD 屏幕底部显示当前时间和蜂鸣器是否启用。
3. 通过触摸屏实现数字的输入和操作符的选择。
4. 通过键盘实现数字的输入和操作符的选择。
5. 通过按键中断设定输入时是否启用蜂鸣器。
6. 编写程序实现计算器的功能。
7. 通过 LCD 屏幕输出计算器的运算表达式。
8. 通过串口输出计算器的运算结果。

三、设计内容

本项目使用了 STM32F4 微控制器，结合触摸屏、键盘、蜂鸣器、LED、RTC、USART、按键中断等功能，实现了一个简易的计算器。

1. 主函数

1.1. main

主函数代码分析：

1. 变量声明部分
   (a) `char input_value;`: 用于存储用户输入的字符。
   (b) `int key_value = 0;`: 存储键盘输入的值。
   (c) `double num = 0.0;` 和 `double result = 0.0;`: 用于存储数字和计算结果的变量，以双精度浮点数的形式。
   (d) `int flag_num = 0;`: 用于标记当前程序处于等待输入第一个数字还是等待输入第二个数字的状态。
   (e) `int flag_calc = 0;`: 用于标记当前计算器将执行的操作类型（加法、减法、乘法或除法）。
   (f) `char input_sequence[100] = ""`: 用于存储用户输入的数字和操作符的数组。
   (g) `char display_buffer[100] = ""`: 用于存储计算结果以及显示在 LCD 上的其他信息的数组。
   (h) `int BeepFlag = 1;`: 用于控制蜂鸣器的状态。

2. 主函数 `main()`

   (a) 进行了各种外设（LED、GPIO、BEEP、EXTI、RTC、USART 等）的初始化配置。

   (b) 初始化了 LCD 屏幕。

   (c) 检查 RTC 备份寄存器，若未设置时间和日期，则进行设置。

   (d) 初始化触摸屏，并根据液晶的扫描方向更新触摸配置。

   (e) 绘制计算器界面。

   (f) 进入了一个无限循环 `while（1）`中，不断监听用户输入和触摸事件。

3. 主循环部分

   (a) 不断检测触摸中断和键盘输入，并在有输入时执行相应的操作。

   (b) 若触摸中断发生，则获取输入并处理，若 `BeepFlag == 1` 则同时开启蜂鸣器以示提示。

   (c) 若键盘输入发生，则进行按键扫描。

   (d) 通过 `Calculate()` 函数处理输入，计算结果并显示在 LCD 上。

   (e) 循环显示当前输入的数字和操作符。

   (f) 显示计算结果和当前时间。

```c
char input_value;
int key_value = 0;
double num = 0.0;
double result = 0.0;
int flag_num = 0;  // 0: waiting for first number, 1: waiting for second number
int flag_calc = 0; // 0: +, 1: -, 2: *, 3: /
char input_sequence[100] = ""; // 用于存储输入的数字和操作符
char display_buffer[100] = ""; // 用于存储显示的结果
int BeepFlag = 1;

// 触摸中断标志位
extern volatile int InterruptTouchFlag;
// 输入标志位
volatile int InputFlag = 0;

int main(void)
{
    LED_GPIO_Config();
    GPIO_Configuration();
    BEEP_GPIO_Config(); // 蜂鸣器 端口初始化
    EXTI_Key_Config();
    NT35510_Init(); // LCD 初始化
    Debug_USART_Config();

    // 其中 0、3、5、6 模式适合从左至右显示文字，
    // 不推荐使用其它模式显示文字 ^^I 其它模式显示文字会有镜像效果
    // 其中 6 模式为大部分液晶例程的默认显示方向
    NT35510_GramScan(3);
```

```
// /* RTC 配置：选择时钟源，设置 RTC_CLK 的分频系数 */
RTC_CLK_Config();

if (RTC_ReadBackupRegister(RTC_BKP_DRX) != 0xf) // RTC_BKP_DATA
{
    /* 设置时间和日期 */
    RTC_TimeAndDate_Set();
}
else
{
    /* 检查是否电源复位 */
    if (RCC_GetFlagStatus(RCC_FLAG_PORRST) != RESET)
    {
        printf("\r\n 发生电源复位....\r\n");
    }
    /* 检查是否外部复位 */
    else if (RCC_GetFlagStatus(RCC_FLAG_PINRST) != RESET)
    {
        printf("\r\n 发生外部复位....\r\n");
    }

    printf("\r\n 不需要重新配置 RTC....\r\n");

    /* 使能 PWR 时钟 */
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);
    /* PWR_CR:DBF 置 1，使能 RTC、RTC 备份寄存器和备份 SRAM 的访问 */
    PWR_BackupAccessCmd(ENABLE);
    /* 等待 RTC APB 寄存器同步 */
    RTC_WaitForSynchro();
}

/* 设定好液晶扫描方向后，再初始化触摸屏，触摸屏会根据液晶的扫描方向输出匹配的触摸坐标 */
/* 每次修改液晶扫描方向后，应重新调用一次 GTP_Init_Panel 函数更新触摸配置 */
GTP_Init_Panel();

printf("\r\n ********** 计算器程序 ********** \r\n");

// 绘制计算界面
Calculator_UI();

// 提示获取输入值
printf("Enter a digit or an operator (+, -, *, /): ");

while (1)
{
    // LCD 显示蜂鸣器开启状态
    if (BeepFlag == 1)
    {
        NT35510_DispString_EN(650, 435, "BEEP: 1");
    }
    else
    {
        NT35510_DispString_EN(650, 435, "BEEP: 0");
```

```
        }

        // 当触摸发生
        while (InterruptTouchFlag == 1)
        {
            input_value = Get_Input_Touch();
            InterruptTouchFlag = 0;
            printf(" 一次触摸中断已经完成，标志位被清零，等待下一次中断产生\n");
            InputFlag = 1;
            if (BeepFlag == 1)
            {
                BEEP_ON;
                Delay(0x000FFF);
                BEEP_OFF;
            }
        }

        // 等待键盘发生
        Key_scan();

        while (InputFlag == 1)
        {
            Calculate();
            InputFlag = 0;
            printf(" 一次输入处理已经完成，标志位被清零，等待下一次输入\n");
        }

        // 显示实际数字，即将 input_sequence 显示到 LCD
        NT35510_DispStringLine_EN(LINE(1), input_sequence); /* 显示单行文字 */

        // 显示计算结果

        sprintf(display_buffer, "Result: %.2lf", result);
        NT35510_DispStringLine_EN(LINE(2), display_buffer); /* 显示单行文字 */

        // 在 LCD 上显示时间
        RTC_Show_Time();
    }
}
```

四、总结

通过本次实验，我学会了如何使用 STM32F4 微控制器，结合触摸屏、键盘、蜂鸣器等外设，实现一个简易的计算器。对 STM32F4 微控制器的使用有了更深入的了解，对触摸屏的使用也有了更多的实践经验。通过本次实验，我对嵌入式系统的开发有了更深入的了解，对嵌入式系统的开发有了更多的实践经验。