

Lab12 温度传感器

一、实验目的

1. 熟练建立一个完整的工程，掌握嵌入式系统开发流程；了解嵌入式系统开发中常用的硬件功能模块和硬件开发工具；
2. 了解内部温度传感器模块的基本原理和开发应用。

二、实验内容

本实验介绍 STM32 的内部温度传感器，将使用 STM32 的内部温度传感器来读取温度值，通过 ADC 模块将内部传感器信号转换成温度值，并在 TFTLCD 屏幕上显示出来。

STM32 有一个内部的温度传感器，可以用来测量 CPU 及周围的温度。该温度传感器在内部和 ADCx_IN16 输入通道相连接，此通道把传感器输出的电压转换成数字值。温度传感器模拟输入推荐采样时间是 $17.1\mu\text{s}$ 。STM32 的内部温度传感器支持的温度范围为：-40~125 度，精度为 $\pm 1.5^{\circ}\text{C}$ 左右。

STM32 内部温度传感器的使用很简单，只要设置一下内部 ADC，并激活其内部通道，和温度传感器设置相关的 2 个地方。

STM32 的 ADC 是 12 位逐次逼近型的模拟数字转换器。它有 18 个通道，可测量 16 个外部和 2 个内部信号源。各通道的 A/D 转换可以单次、连续、扫描或间断模式执行。ADC 的结果可以左对齐或右

对齐方式存储在 16 位数据寄存器中。模拟看门狗特性允许应用程序检测输入电压是否超出用户定义的高/低阈值。ADC 使用到的库函数分布在 `stm32f10x_adc.c` 文件和 `stm32f10x_adc.h` 文件中。（详见《STM32 不完全手册》之《ADC 实验》）

温度传感器设置第一个地方，要使用 STM32 的内部温度传感器，必须先激活 ADC 的内部通道，这里通过 `ADC_CR2` 的 `AWDEN` 位（bit23）设置。设置该位为 1 则启用内部温度传感器。

第二个地方，STM32 的内部温度传感器固定的连接在 ADC 的通道 16 上，所以，我们在设置好 ADC 之后只要读取通道 16 的值，就是温度传感器返回来的电压值了。根据这个值，我们就可以计算出当前温度。计算公式如下：

$$T(^{\circ}\text{C}) = \{ (V_{25} - V_{\text{sense}}) / \text{Avg_Slope} \} + 25$$

上式中：

$V_{25} = V_{\text{sense}}$ 在 25 度时的数值（典型为： 度时的数值（典型为：1.43 ）。

$\text{Avg_Slope} =$ 温度与 V_{sense} 曲线的平均斜率（单位：mv/°C 或 uv/°C）（典型值： 4.3mv/°C）。

利用以上公式，可计算出当前温度传感器的温度。

通过库函数设置 STM32 内部温度传感器的步骤如下：

1) 设置 ADC，开启内部温度传感器。

如何设置 ADC，通过温度传感器读取外部通道的值，内部温度传感器相当与把通道端口连接在内部温度传感器上。要开启内部温度传感器功能:ADC_TempSensorVrefintCmd (ENABLE);

2) 读取通道 16 的 AD 值，计算结果。

在设置完之后，读取温度传感器的电压值，用上面的公式计算温度值。

温度传感器的相关功能主要在 `tsensor.c` 文件和 `tsensor.h` 文件。其中 `tsensor.c` 文件中有三个函数分别为 `T_Adc_Init`, `T_Get_Temp`, `T_Get_Adc_Average`. 在 `Adc_Init` 函数中设置为开启内部温度传感器模式，代码如下：

```

void T_Adc_Init(void) //ADC 通道初始化
{
    ADC_InitTypeDef ADC_InitStructure;
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA |
    RCC_APB2Periph_ADC1, ENABLE );//使能 GPIOA,ADC1 通道时钟
    RCC_ADCCLKConfig(RCC_PCLK2_Div6); //分频因子 6 时钟为
    72M/6=12MHz
    ADC_DeInit(ADC1); //将外设 ADC1 的全部寄存器重设为缺省值
    ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;//ADC 独立模式
    ADC_InitStructure.ADC_ScanConvMode = DISABLE;//模数转换:单通道模式
    ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;//单次转换模式
    ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;//软件
    触发
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;//ADC 数据右对齐
    ADC_InitStructure.ADC_NbrOfChannel = 1;//顺序进行规则转换的 ADC 通道的数
    目
    ADC_Init(ADC1, &ADC_InitStructure);//根据指定的参数初始化 ADCx
    ADC_TempSensorVrefintCmd(ENABLE); //开启内部温度传感器
    ADC_Cmd(ADC1, ENABLE);//使能指定的 ADC1
    ADC_ResetCalibration(ADC1);//重置指定的 ADC1 的复位寄存器
    while(ADC_GetResetCalibrationStatus(ADC1));//等待校准完成
    ADC_StartCalibration(ADC1);//AD 校准
    while(ADC_GetCalibrationStatus(ADC1));//等待校准完成
}

```

三、实验前准备

1. 硬件准备

本实验所要用到的硬件资源包括：1) 指示灯 DS0 ； 2) TFTLCD 显示模块； 3) ADC； 4) 内部温度传感器； 5) 红外遥控器或串口。

(1) TFTLCD 连接：miniSTM32 开发板上 TFTLCD 插槽上比 TFTLCD 屏幕的插针多 2 个插口，如下图靠右插好，留出左侧两个插口，同时 TFTLCD 下方两个螺帽与开发板两个预留口对准，便于开发板与屏幕固定（实验中可以不用固定）；

(2) 下载器连接，连接方法参考实验（二）。

(3) 显示屏需要比较大电流，USB 供电可能不足，导致屏幕无法显示，请用外部电源供电(5V 电源，一般的手机充电器的插头即可，请检查其输出电压)

四、实验过程

1. 工程建立：具体过程略。

2. 主要功能

伴随 DS0 的不停闪烁，提示程序在运行。屏幕上显示当前的温度值，看看温度值与实际是否相符合（由于芯片会发热，一般会比实际温度偏高）。

五、作业

1. 功能实现：（1）在屏幕上显示实时温度值，根据数值变化，在屏幕上绘制温度变化曲线（要求横纵坐标有温度数值/单位/时间等图例）；（2）通过红外遥控器的开关键，切换屏幕上温度单位显示华氏度或者摄氏度（包括实时显示值和温度变化曲线上均调整）。（3）如果没有红外遥控器的，可以采用串口发送指令进行控制功能 2。

2. 作业提交：主要控制代码，及功能运行结果截图（LCD 显示界面）。

3. DDL: Dec 9.

4. 本周可检查 lab10.

