**Microcontroller Experiment**

Internal Temperature Sensor Experiment

Name: Han Yichen 韩一尘

Student ID: 22722051

Date: 2023/11/1

1. **Experimental Principles**

STM32 has an internal temperature sensor that can be used to measure the CPU and surrounding temperature (TA). The temperature sensor is internally and ADCx\_IN16 input channel is connected, which converts the voltage output by the sensor into a digital value. The recommended sampling time for temperature sensor analog input is 17.1 μ S. The internal temperature sensor of STM32 supports a temperature range of -40~125 degrees. The accuracy is relatively poor, around ± 1.5℃.

There are two key points for temperature sensors: Firstly, to use the internal temperature sensor of STM32, we must first activate the internal channel of ADC, where ADC is used\_ Set the AWDEN bit (bit23) of CR2. Setting this bit to 1 enables the internal temperature sensor. Secondly, the internal temperature sensor of STM32 is fixedly connected to channel 16 of the ADC. Therefore, after setting the ADC, we only need to read the value of channel 16, which is the voltage value returned by the temperature sensor. Based on this value, we can calculate the current temperature:

In the above equation:

V25=Value of Vsense at 25 degrees (typical value: 1.43).

Avg\_ Slope=The average slope of the temperature Vsense curve (in mv/℃ or UV/℃) (typical value is 4.3Mv/℃).

1. **Main program analysis**

There are three functions in the tsensor. c file called T\_ Adc\_ Init, T\_ Get\_ Temp and T\_ Get\_ Adc\_ Average. Among them, in Adc\_Init function is set to enable the internal temperature sensor mode, and the code is as follows:

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)); //等待教主年完成 ADC\_StartCalibration(ADC1); //AD 校准

while(ADC\_GetCalibrationStatus(ADC1)); //等待校准完成

}

We also added Get to tsensor. c\_ The Temprate function is used to obtain the temperature value, which is to convert the collected voltage into a temperature value according to the calculation formula, and the resulting temperature value is increased by 100 times. The codes are shown as following:

short Get\_Temprate(void) //获取内部温度传感器温度值

{ u32 adcx; short result; double temperate;

adcx=T\_Get\_Adc\_Average(ADC\_Channel\_16,20); //读取通道 16,20 次取平均 temperate=(float)adcx\*(3.3/4096); //电压值

temperate=(1.43-temperate)/0.0043+25; //转换为温度值

result=temperate\*=100; //扩大 100 倍

return result;

}

And the main function is shown as following:

int main(void) { u16 adcx; float temp; float temperate; delay\_init(); //延时函数初始化 NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2); //设置 NVIC 中断分组 2 uart\_init(115200); //串口初始化波特率为 115200 LED\_Init(); //LED 端口初始化 LCD\_Init(); //LCD 初始化

T\_Adc\_Init(); //ADC 初始化

POINT\_COLOR=RED;//设置字体为红色

LCD\_ShowString(60,50,200,16,16,"WarShip STM32"); LCD\_ShowString(60,70,200,16,16,"Temperature TEST"); LCD\_ShowString(60,90,200,16,16,"ATOM@ALIENTEK"); LCD\_ShowString(60,110,200,16,16,"2015/9/7"); //显示提示信息 POINT\_COLOR=BLUE;//设置字体为蓝色 LCD\_ShowString(60,130,200,16,16,"TEMP\_VAL:"); LCD\_ShowString(60,150,200,16,16,"TEMP\_VOL:0.000V"); LCD\_ShowString(60,170,200,16,16,"TEMPERATE:00.00C"); while(1) { temp=Get\_Temprate(); //得到温度值

if(temp<0)

{

temp=-temp;

LCD\_ShowString(30+10\*8,140,16,16,16,"-"); //显示负号 }else LCD\_ShowString(30+10\*8,140,16,16,16," "); //无符号 LCD\_ShowxNum(30+11\*8,140,temp/100,2,16,0); //显示整数部分 LCD\_ShowxNum(30+14\*8,140,temp%100,2,16, 0X80); //显示小数部分

LED0=!LED0;

delay\_ms(250);

}

}

1. **Experimental result**

By compiling and importing the code, we obtained the following content on the display screen:

****

1. **Improvement and perfection**

When the temperature displayed on the monitor was obtained, it was found that the reading was significantly higher than the ambient temperature. In the experiment, when the weather temperature was around 22 ℃, the reading was as high as 30 ℃. After thinking and discussion, it was finally understood that the chip was powered on and operated to release heat, which would cause the temperature obtained by the sensor to be higher than the actual temperature

In order to obtain a temperature that is more in line with the environment, two options can be adopted: (1) Do not power on the chip for a long time before conducting the experiment, as this may cause errors in the chip due to the heat generated by the current; (2) You can try installing a heat sink to effectively reduce the impact of the heat generated by the current on the data