Task:

1. 在ADC实验中，结合热敏电阻，分别通过触摸芯片表面和热敏电阻，引起A/D值变化，显示芯片内部温度和当前温度:

//======================================================================

// 文件名称：main.c（应用工程主函数）

// 框架提供：SD-Arm（sumcu.suda.edu.cn）

// 版本更新：20191108-20200419

// 功能描述：见本工程的..\01\_Doc\Readme.txt

// 移植规则：【固定】

//======================================================================

#define GLOBLE\_VAR

#include "includes.h" // 包含总头文件

//----------------------------------------------------------------------

// 声明使用到的内部函数

void **Delay\_ms**(uint16\_t u16ms);

float **Regression\_Ext\_Temp**(uint16\_t tmpAD); // 环境温度AD值转为实际温度

float **Regression\_MCU\_Temp**(uint16\_t mcu\_temp\_AD); // MCU温度AD值转为实际温度

//----------------------------------------------------------------------

// 主函数，一般情况下可以认为程序从此开始运行（实际上有启动过程，参见书稿）

//主函数

**int main**(void)

{

uint16\_t num\_AD1;

uint16\_t num\_AD2;

uint16\_t num\_AD3;

uint32\_t mMainLoopCount; //主循环次数变量

//关总中断

DISABLE\_INTERRUPTS;

mMainLoopCount=0; //主循环次数变量

**adc\_init**(ADC\_CHANNEL\_1,AD\_DIFF); //初始化ADC通道1，

**adc\_init**(ADC\_CHANNEL\_15,AD\_DIFF); //初始化ADC通道15

**adc\_init**(ADC\_CHANNEL\_TEMPSENSOR,AD\_SINGLE); //初始化ADC通道：内部温度

ENABLE\_INTERRUPTS;

**printf**("------------------------------------------------------\n");

**printf**("金葫芦提示： \n");

**printf**("（1）目的：ADC单端输入与差分输入测试 \n");

**printf**("（2）单端：内部温度传感器，通道号17，无需引脚对应 \n");

**printf**(" 差分：GEC引脚47、46（通道1、2） \n");

**printf**(" GEC引脚12、11（通道15、16） \n");

**printf**("（3）测试方法：单端：手摸芯片表面，A/D值增大，不要摸 \n");

**printf**(" 到引脚，静电可能损坏芯片 \n");

**printf**(" 差分：将引脚47接地、46接3.3V,观察通道1情况\n");

**printf**(" 将引脚46接地、47接3.3V,观察通道1情况\n");

**printf**(" 类似方法，观察通道15 \n");

**printf**("------------------------------------------------------\n");

**for**(;;)

{

//主循环次数变量+1

mMainLoopCount++;

//未达到主循环次数设定值，继续循环

//延时1秒

**if** (mMainLoopCount<=30000000) continue;

//达到主循环次数设定值，执行下列语句，进行灯的亮暗处理

//清除循环次数变量

mMainLoopCount=0;

num\_AD1 = **adc\_ave**(ADC\_CHANNEL\_1,8);

num\_AD2 = **adc\_ave**(ADC\_CHANNEL\_15,8);

num\_AD3 = **adc\_ave**(ADC\_CHANNEL\_TEMPSENSOR,8);

**printf**("通道1(GEC47、46)的A/D值： %d\r\n",num\_AD1);

**printf**("通道1(GEC47、46)的温度值： %f\r\n",**Regression\_Ext\_Temp**(num\_AD1));

**printf**("通道15(GEC12、11）的A/D值：%d\r\n",num\_AD2);

**printf**("通道15(GEC12、11）的温度值：%f\r\n",**Regression\_Ext\_Temp**(num\_AD2));

**printf**("内部温度传感器的A/D值：%d\r\n",num\_AD3);

**printf**("内部温度传感器的温度值：%f\r\n\n",**Regression\_MCU\_Temp**(num\_AD3));

}

}

//======以下为主函数调用的子函数===========================================

//======================================================================

// 函数名称：Delay\_ms

// 函数返回：无

// 参数说明：无

// 功能概要：延时 - 毫秒级

//======================================================================

void **Delay\_ms**(uint16\_t u16ms)

{

uint32\_t u32ctr;

**for**(u32ctr = 0; u32ctr < 8000\*u16ms; u32ctr++)

{

**\_\_ASM**("NOP");

}

}

//============================================================================

// 函数名称：Regression\_Ext\_Temp

// 功能概要：将读到的环境温度AD值转换为实际温度

// 参数说明：tmpAD：通过adc\_read函数得到的AD值

// 函数返回：实际温度值

//============================================================================

float **Regression\_Ext\_Temp**(uint16\_t tmpAD)

{

float Vtemp, Rtemp, temp;

**if**(tmpAD <= 72)

{

return -274;

}

Vtemp = (tmpAD \* 3300.0) / 4096;

Rtemp = Vtemp / (3300.0 - Vtemp) \* 10000.0;

temp = (1 / (**log**(Rtemp / 10000.0) / 3950.0 + (1 / (273.15 + 25)))) - 273.15 + 0.5;

return temp;

}

//============================================================================

// 函数名称：Regression\_MCU\_Temp

// 功能概要：将读到的mcu温度AD值转换为实际温度

// 参数说明：mcu\_temp\_AD：通过adc\_read函数得到的AD值

// 函数返回：实际温度值

//============================================================================

float **Regression\_MCU\_Temp**(uint16\_t mcu\_temp\_AD)

{

float mcu\_temp\_result;

mcu\_temp\_result = (float)(55 + (100 \* ((float)(mcu\_temp\_AD) - AD\_CAL1)) / (AD\_CAL2 - AD\_CAL1));

return mcu\_temp\_result;

}

//========================================================================

/\*

知识要素：

（1）main.c是一个模板，该文件所有代码均不涉及具体的硬件和环境，通过调用构件

实现对硬件的干预。

（2）本文件中对宏GLOBLE\_VAR进行了定义，所以在包含"includes.h"头文件时，会定

义全局变量，在其他文件中包含"includes.h"头文件时，

编译时会自动增加extern

\*/

