Task:

1. 理解flash.c各函数，编写程序向flash中写入并读出数据:

先向flash中写入并读出数据：

Main.c:

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

//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

//主函数

**int main**(void)

{

uint8\_t mK1[32]; //按照逻辑读方式从指定flash区域中读取的数据

uint8\_t result; //判断扇区是否为空标识

uint16\_t sector = 67U;

//初始化扇区内容

**flash\_erase**(sector);

**flash\_write**(sector,0,32,(uint8\_t \*) "qwertyuiooooop");

**printf**("初始化%d扇区。\n",sector);

**flash\_read\_logic**(mK1,sector,0,32); //从指定扇区读取32个字节到mK1中

**printf**("擦除前%d扇区32字节的内容：%s\n",sector,mK1);

result = **flash\_isempty**(sector,MCU\_SECTORSIZE); // 判断指定扇区是否为空

**printf**("扇区判空，1空，0不空：%d\n",result);

//擦除指定扇区

**flash\_erase**(sector);

**printf**("\n擦除%d扇区。\n",sector);

**flash\_read\_logic**(mK1,sector,0,32); //从指定扇区读取32个字节到mK1中

**printf**("擦除后%d扇区32字节的内容：%s\n",sector,mK1);

result = **flash\_isempty**(sector,MCU\_SECTORSIZE); // 判断指定扇区是否为空

**printf**("扇区判空，1空，0不空：%d\n",result);

//向指定扇区第0偏移地址开始写32个字节数据

**flash\_write**(sector,0,32,(uint8\_t \*) "广州大学LHL 32106100067");

**printf**("\n写入%d扇区。\n",sector);

**flash\_read\_logic**(mK1,sector,0,32); //从指定扇区读取32个字节到mK1中

**printf**("写入后%d扇区32字节的内容：%s\n",sector,mK1);

result = **flash\_isempty**(sector,MCU\_SECTORSIZE); // 判断指定扇区是否为空

**printf**("扇区判空，1空，0不空:%d\n",result);

**for**(;;) //for(;;)（开头）

{

}

}

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

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

/\*

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

\*/



## 用实验验证，对于有数据的某扇区，如果没有擦除（Flash\_erase），可否写入新数据？（注：扇区号为学号后2位，数据文本中要有姓名。）

Main.c:

#define GLOBLE\_VAR

#include "includes.h" //包含总头文件

//主函数

**int main**(void)

{

uint8\_t mK1[32]; //按照逻辑读方式从指定flash区域中读取的数据

uint8\_t result; //判断扇区是否为空标识

uint16\_t sector = 67U;

//初始化扇区内容

**flash\_erase**(sector);

**flash\_write**(sector,0,32,(uint8\_t \*) "oop");

**printf**("初始化%d扇区。\n",sector);

**flash\_read\_logic**(mK1,sector,0,32); //从指定扇区读取32个字节到mK1中

**printf**("擦除前%d扇区32字节的内容：%s\n",sector,mK1);

result = **flash\_isempty**(sector,MCU\_SECTORSIZE); // 判断指定扇区是否为空

**printf**("扇区判空，1空，0不空：%d\n",result);

//擦除指定扇区

**flash\_erase**(sector);

**printf**("\n擦除%d扇区。\n",sector);

**flash\_read\_logic**(mK1,sector,0,32); //从指定扇区读取32个字节到mK1中

**printf**("擦除后%d扇区32字节的内容：%s\n",sector,mK1);

result = **flash\_isempty**(sector,MCU\_SECTORSIZE); // 判断指定扇区是否为空

**printf**("扇区判空，1空，0不空：%d\n",result);

/\* //向指定扇区第0偏移地址开始写32个字节数据

flash\_write(sector,0,32,(uint8\_t \*) "广州大学 32106100067");

printf("\n写入%d扇区。\n",sector);

flash\_read\_logic(mK1,sector,0,32); //从指定扇区读取32个字节到mK1中

printf("写入后%d扇区32字节的内容：%s\n",sector,mK1);

result = flash\_isempty(sector,MCU\_SECTORSIZE); // 判断指定扇区是否为空

printf("扇区判空，1空，0不空：%d\n",result);\*/

**for**(;;) //for(;;)（开头）

{

}

}

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

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

/\*

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





结果可以看出：在未对指定扇区擦除的情况下直接写入，写入会失败，即操作扇区时，没有擦除不可写入新数据。