计算机科学与技术学院

嵌入式系统实验报告

（一）

姓　　名：banban

专　　业：计算机科学与技术

班　　级：

学　　号：

指导教师：

2023 年 3 月 16 日

一、任务要求

1、 安装 keil mdk5

2、 打开并运行演示工程，并通过软件仿真加以验证

3、 通过实验平台原理图了解硬件原理，并完成硬件连接

4、 使用 mcuisp 将生成的 hex 文件烧写入开发板，进行硬件验证（有条件者使用 J-link 仿真器进行硬件仿真调试）

5、 针对特定 mcu（STM32F103ZE）创建工程，并导入固件库，完成工程配置，作为模板

6、 编写程序，以库函数方式操纵 GPIO 实现 8 个 LED 的流水灯效果（效果可以定制）

二、实验报告要求

1、 整理工程创建、配置到编写代码、调试的全过程，形成图文版教程

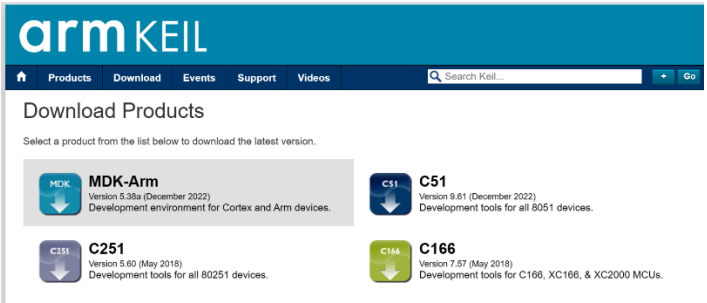
2、 自己编写的实现实现 8 个 LED 的流水灯效果的源代码（加上注释）

3、 能说明软件仿真结果的截图和硬件验证结果的图片

三、实验过程

一. 任务一：安装 keil mdk5，并运行演示工程

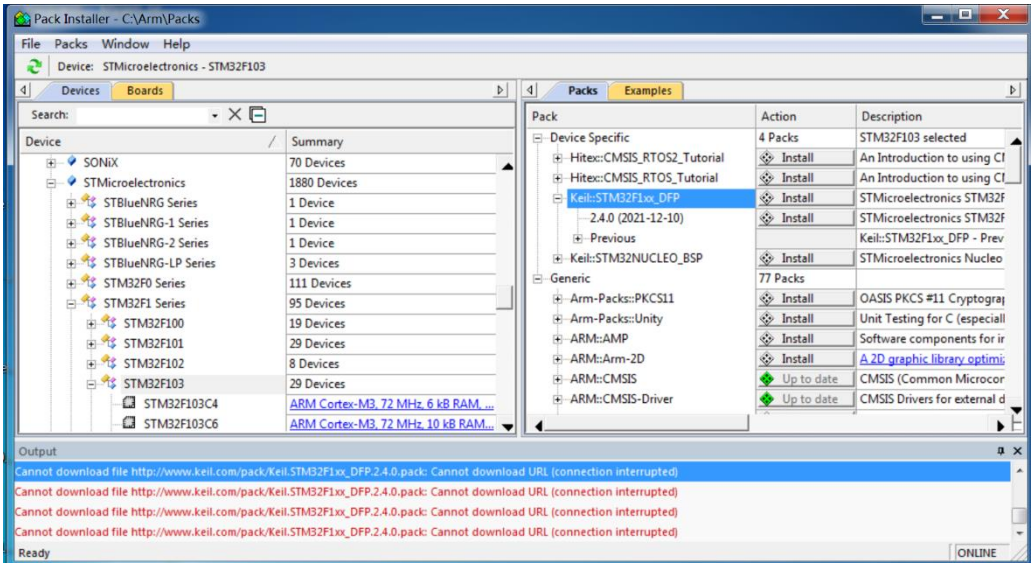
1. 进入官方网站下载 keil mdk5



2. 安装开发包 Pack Installer：

左边找到 STMicroelectronics → STM32F1 →STM32F103，点击

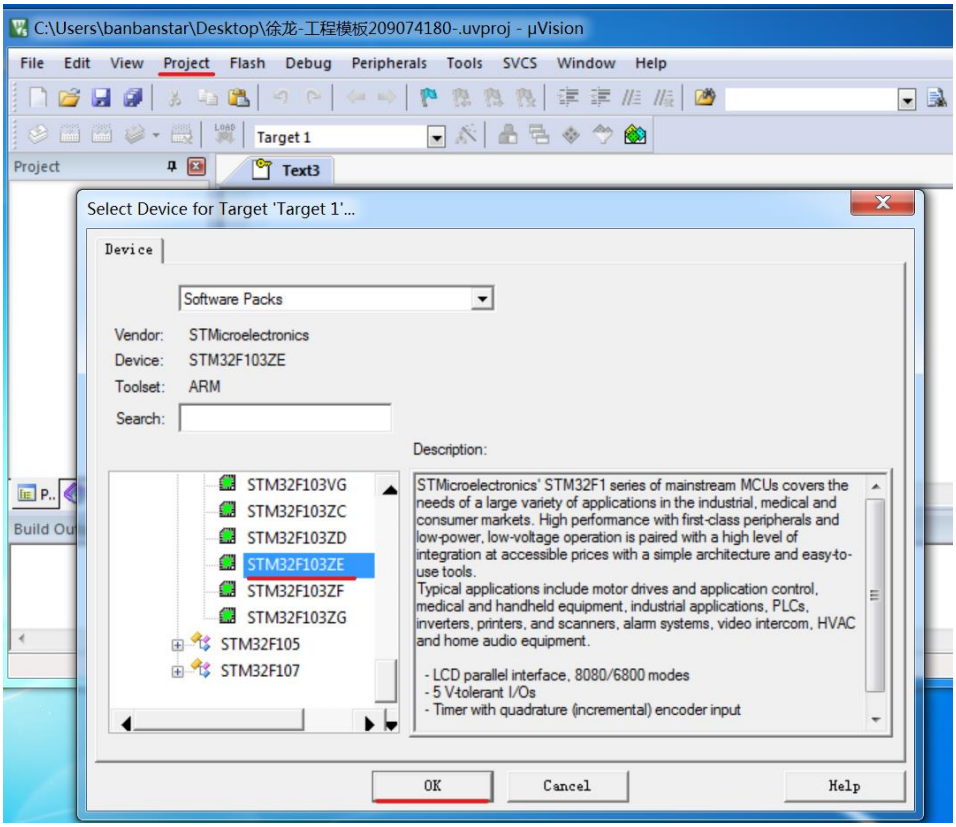
再在右边找到 Device Specific → Keil::STM32F1xx_DFP，点击 Install



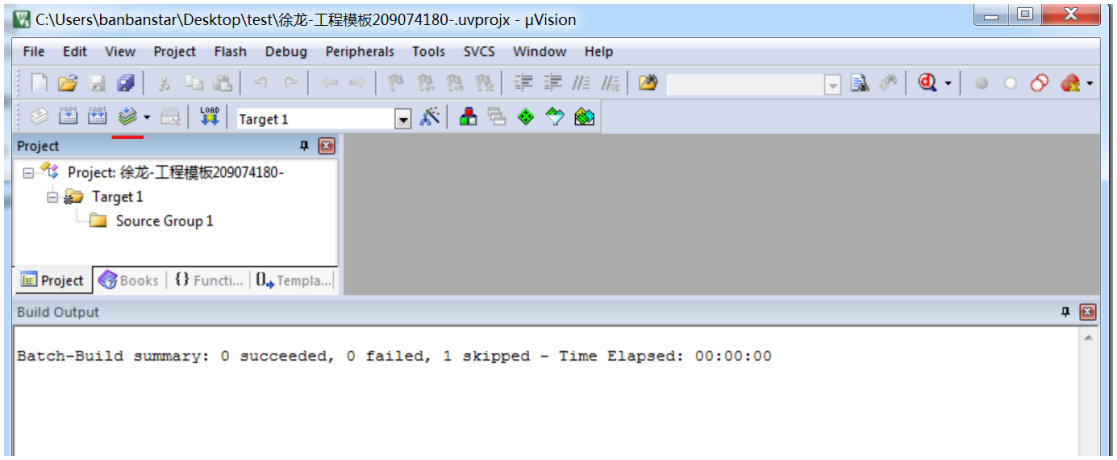
3. 创建基础工程

Project -> New uVision Project → 填上大名（最好先创建一个文件夹，再创建工程）

选择 STM32F103ZE → ok ：

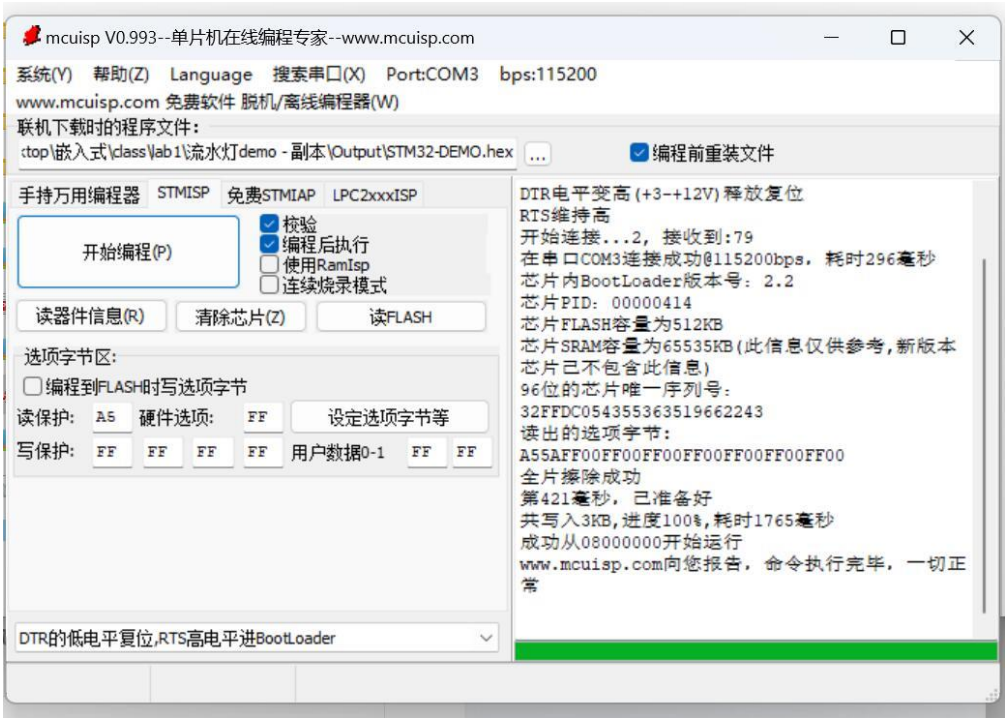


4. 运行结果



二. 任务二：使用 mcuisp 将生成的 hex 文件烧写入开发板

1. 选择“搜索串口”（如果识别到板子就会有选项）
2. 设置”波特率” 为 460800（尽量不要设置得太高）
3. 选择要下载的 HEX 文件
4. 勾选“校验”、“编程后执行” 复选框
5. 选择“DTR 低电平复位，RTS 高电平进 Boot Loader”选项
6. 点击“开始编程”



三.　任务三：以库函数方式操纵 GPIO 实现 8 个 LED 的流水灯效果

1. 代码：

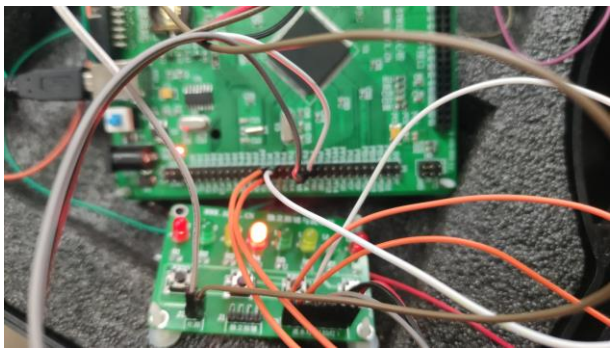
```
// led.h
// ----------------------------------------------------------
#define ON  0
#define OFF 1
//带参宏，可以像内联函数一样使用
#define LED1(a) if (a) GPIO_SetBits(GPIOC,GPIO_Pin_1);  \
                    else GPIO_ResetBits(GPIOC,GPIO_Pin_1)
#define LED2(a) if (a) GPIO_SetBits(GPIOC,GPIO_Pin_2);  \
                    else GPIO_ResetBits(GPIOC,GPIO_Pin_2)
...
void LED_GPIO_Config(void);
```

```
// led.c
// ----------------------------------------------------------
#include "led.h"
void LED_GPIO_Config(void) {   // 配置 LED 用到的 I/O 口
    /*定义一个 GPIO_InitTypeDef 类型的结构体*/
    GPIO_InitTypeDef GPIO_InitStructure;
    /*开启 GPIOC 的外设时钟*/
    RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOC, ENABLE);
    /*选择要控制的 GPIOC 引脚*/
    GPIO_InitStructure.GPIO_Pin =  GPIO_Pin_1 | GPIO_Pin_2 |
GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7
|GPIO_Pin_8 ;
    /*设置引脚模式为通用推挽输出*/
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    /*设置引脚速率为 50MHz */
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    /*调用库函数，初始化 GPIOC*/
    GPIO_Init(GPIOC, &GPIO_InitStructure);
    /* 关闭所有 led 灯 */
```

```
    GPIO_SetBits(GPIOC, GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3 |
GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7 |GPIO_Pin_8 );
}
```

```
// main.c
// -----------------------------------------------------------
#include "stm32f10x.h"
#include "led.h"
void Delay(__IO u32 nCount);
int main(void){
    LED_GPIO_Config();    /* LED 端口初始化 */
    while (1) {
        LED1( ON );           // 亮
        Delay(0x0FFFEF);
        LED1( OFF );          // 灭
        LED2( ON ); Delay(0x0FFFEF); LED2( OFF );
        LED3( ON ); Delay(0x0FFFEF); LED3( OFF );
        LED4( ON ); Delay(0x0FFFEF); LED4( OFF );
        LED5( ON ); Delay(0x0FFFEF); LED5( OFF );
        LED6( ON ); Delay(0x0FFFEF); LED6( OFF );
        LED7( ON ); Delay(0x0FFFEF); LED7( OFF );
        LED8( ON ); Delay(0x0FFFEF); LED8( OFF );
    }
}
void Delay(__IO u32 nCount)  // 简单的延时函数
{ for(; nCount != 0; nCount--); }
```

2. 图片效果



四、总结与分析

这次 LED 流水灯实验难度不大，在老师的帮助下，我通过 STM32 库函数初始化 GPIO 端口，连接 LED 灯到初始化的引脚，实验成功。

在本实验中使用了库函数进行编程，我还写成寄存器编程版本，例如以下：

```
// main.c
//-------------------------------------------------------------
#include"stm32f10x.h"
#include"delay.h"
#include"LED.h"
int main(void){
    delay_init();  // 延时函数初始化
    LED_Init();    // LED 初始化函数
    while(i){
        // 目的是使连接该 PB.5 处于高电平，即使得灯一开始处于熄灭状态
        GPIOB->ODR |= 1<<5;
        // 目的是使连接该 PE.5 处于低电平，即使得灯一开始处于亮灯状态
        GPIOE->ODR &= !(1<<5);
        delay_ms(500);          // 延时 500ms
        GPIOB=>ODR &= !(1<<5):
        GPIOE->ODR |= 1<<5;
        delay_ms(500);
    }
}
```

在使用中，我发现库函数特性则与寄存器编程相反，在一些代码要求高效率的情况下，对寄存器编程是非常必要的。所以对寄存器的学习与操作，将非常有助于我们在出错时进行程序调试。

学习用 stm32 编程实现点亮熄灭 LED 和单片机控制 LED 闪烁，在理解老师讲解的知识基础上和同学一起探讨交流，慢慢熟悉了一些新知识。操作过程中，软硬件方面都出现一些问题，实验中实验结果讨论遇到发光二极管不亮的情况，于是改用程序之后问题得以解决，因此实验中程序应该及时检查，不然会影响实验的顺利进行。