计算机科学与技术学院

嵌入式系统实验报告

（二）

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 20 日

一、任务要求

结合电路原理图和示例工程项目理解 8 段数码管和 8*8 点阵模块显示原理。创建 mdk 工程，编写程序，选择合适的模块完成硬件连接，实现如下任务：

1、 任务 1：实现在 8 个 8 段数码管上游走显示 A—H（在第一个管子显示 A，延时足够长，在第二个管子显示 B…，依此类推，重复此操作）

2、 任务 2：实现在 8 个 8 段数码管上扫描显示 A—H（在第一个管子显示 A，延时足够短，在第二个管子显示 B…，依此类推，重复此操作，靠视觉暂留形成多个字符同时显示的效果）

提示：上述 2 个任务的不同效果可以通过控制延时的长和短加以实现

3、 任务 3（选做）：实现在 8*8 点阵模块上显示自定义字符串

如：HELLO，WORLD

AHUTJSJ2020

二、实验报告要求

1、 任务 1 - 任务 3 中自编程序的源代码（加上注释）

2、 能说明软件仿真结果的截图、反映硬件电路连接和硬件验证结果的图片或视频

三、实验过程

一.　任务一：8 个 8 段数码管上游走显示 A—H

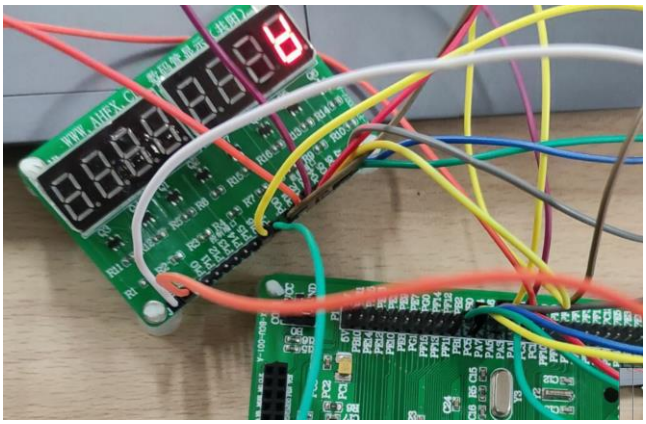
1. 代码：

```
// SysTick.c
// ------------------------------------------------------------
#include "SysTick.h"
static __IO u32 TimingDelay;
void SysTick_Init(void){ // 延时函数
    /* SystemFrequency / 1000    1ms 中断一次
     * SystemFrequency / 100000  10us 中断一次
```

```
     * SystemFrequency / 1000000 1us 中断一次
     */
    //  if (SysTick_Config(SystemFrequency / 100000))   // ST3.0.0 库版本
    if (SysTick_Config(SystemCoreClock / 100000)) {  // ST3.5.0 库版本
        while (1);
    }
    SysTick->CTRL &= ~ SysTick_CTRL_ENABLE_Msk; // 关闭滴答定时器
}
void Delay_us(__IO u32 nTime) {
    TimingDelay = nTime;
    SysTick->CTRL |=  SysTick_CTRL_ENABLE_Msk; // 使能滴答定时器
    while(TimingDelay != 0);
}

// main.c
// ----------------------------------------------------------
#include "sys.h"
#include "systick.h"
// h g f e d c b a
// 1 1 0 0 0 0 0 0 -- 共阳方式显示 0
u8 seg_tab[10] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80,
0x90};
// 0x7f = 01111111 此时仅最左边的管子被激活
u8 pos[8] = {0x7f,0xBF,0xDF,0xEF,0xf7,0xFB,0xFD,0xFe};

void LED_Init(void){
     GPIO_InitTypeDef  GPIO_InitStructure;  // 定义结构体
     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);  // 使能 PA 口时钟
     GPIO_InitStructure.GPIO_Pin =
GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_
6|GPIO_Pin_7;
     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;       // 推挽输出
     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
     GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
    GPIO_SetBits(GPIOA,GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7);

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOE,ENABLE); // 使能 PE 口时钟
    GPIO_InitStructure.GPIO_Pin  = GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7; // KEY0-KEY2
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; // 推挽输出
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOE, &GPIO_InitStructure);
    GPIO_SetBits(GPIOE,GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7);
}
int main(void){
    int i;
    SysTick_Init();   // 延时初始化函数
    LED_Init();       // 数码管初始化函数
    GPIOE->ODR = 0xff;  // 数码管全熄灭
    while(1){
        for(i = 0 ; i < 8 ; i++){
            GPIOA->ODR = seg_tab[i]; // 字形控制
            GPIOE->ODR = pos[i];     // 位置控制
            Delay_us(10000);
        }
    }
}
```

2. 图片效果



二.　任务二：视觉暂留形成多个字符同时显示

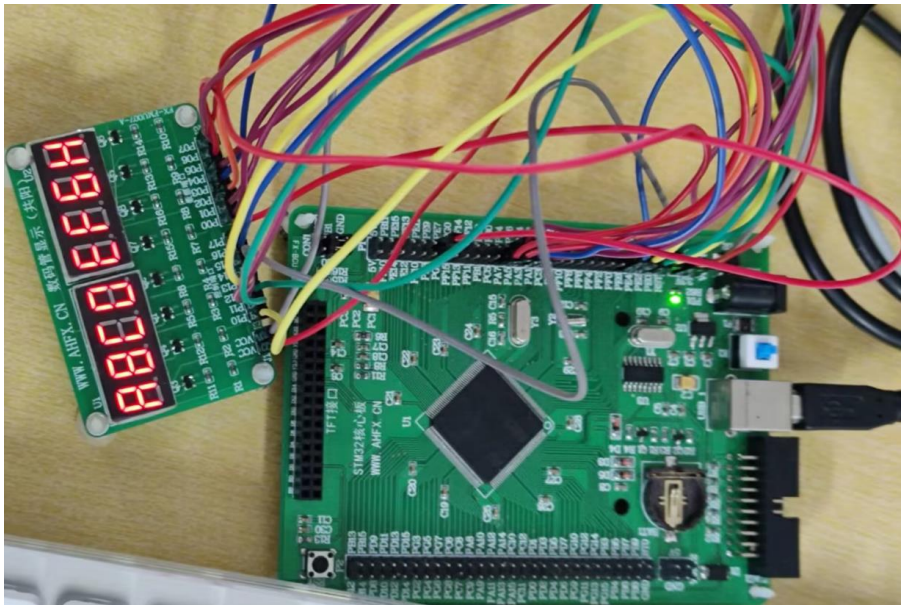
1. 代码

```
// SysTick.c
// ------------------------------------------------------------
#include "SysTick.h"
static __IO u32 TimingDelay;
void SysTick_Init(void){ // 延时函数
    /* SystemFrequency / 1000    1ms 中断一次
     * SystemFrequency / 100000  10us 中断一次
     * SystemFrequency / 1000000 1us 中断一次
     */
    //  if (SysTick_Config(SystemFrequency / 100000))   // ST3.0.0 库版本
    if (SysTick_Config(SystemCoreClock / 100000)) {  // ST3.5.0 库版本
        while (1);
    }
    SysTick->CTRL &= ~ SysTick_CTRL_ENABLE_Msk; // 关闭滴答定时器
}
void Delay_us(__IO u32 nTime) {
    TimingDelay = nTime;
    SysTick->CTRL |=  SysTick_CTRL_ENABLE_Msk; // 使能滴答定时器
    while(TimingDelay != 0);
}

// main.c
// ------------------------------------------------------------
#include "sys.h"
#include "systick.h"
// h g f e d c b a
// 1 1 0 0 0 0 0 0 -- 共阳方式显示 0
u8 seg_tab[10] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80,
0x90};
// 0x7f = 01111111 此时仅最左边的管子被激活
u8 pos[8] = {0x7f,0xBF,0xDF,0xEF,0xf7,0xFB,0xFD,0xFe};
```

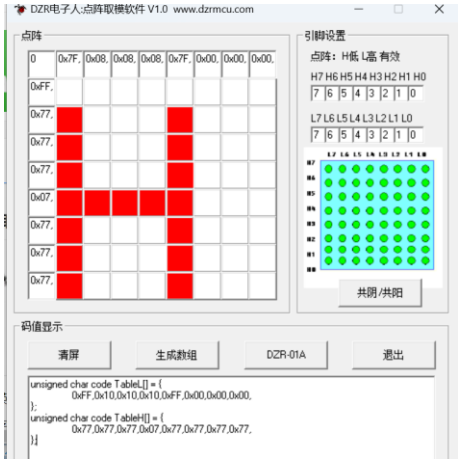
```
void LED_Init(void){
     GPIO_InitTypeDef  GPIO_InitStructure;  // 定义结构体
     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);  // 使能 PA 口时钟
     GPIO_InitStructure.GPIO_Pin =
GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_
6|GPIO_Pin_7;
     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;      // 推挽输出
     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
     GPIO_Init(GPIOA, &GPIO_InitStructure);
     GPIO_SetBits(GPIOA,GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pi
n_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7);

     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOE,ENABLE); // 使能 PE 口时钟
     GPIO_InitStructure.GPIO_Pin  =
GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_
6|GPIO_Pin_7; // KEY0-KEY2
     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; // 推挽输出
     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
     GPIO_Init(GPIOE, &GPIO_InitStructure);
     GPIO_SetBits(GPIOE,GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pi
n_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7);
}

int main(void){
    int i;
    SysTick_Init();   // 延时初始化函数
    LED_Init();       // 数码管初始化函数
    GPIOE->ODR = 0xff;  // 数码管全熄灭
    while(1){
        for(i = 0 ; i < 8 ; i++){
            GPIOA->ODR = seg_tab[i]; // 字形控制
            GPIOE->ODR = pos[i];     // 位置控制
            Delay_ms(1);             // 视觉暂停
        }
    }
```

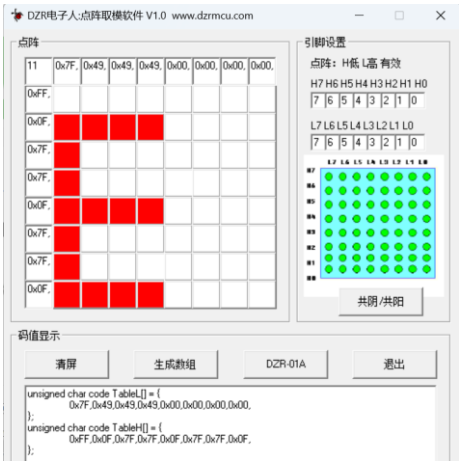
}

2. 图片效果



三. 任务三（选做）：实现在 8*8 点阵模块上显示自定义字符串





四、总结与分析

在老师的帮助指导下，我本次实验主要学习 stm32 单片机上 GPIO 外设—输出设备使用方法。这次实验成功实现了使用 8 位 GPIO 口输出控制 8 个 8 段数码管的位选择线。在轮流设置显示的 8 个显示字符时，设置 delay_ms(10000)，使 8 个 8 段数码管上游走显示 A—H，即在第一个管子显示 A，延时足够长，在第二个管子显示 B。

之后修改 GPIO 初始化程序和相关程序后，使用 delay_ms(2) 方法在每个显示之间添加 2ms 的视觉暂停，达到了在 8 个 8 段数码管上扫描显示 A—H，即在第一个管子显示 A，延时足够短，在第二个管子显示 B，重复此操作，靠视觉暂留形成多个字符同时显示的效果，达到了预期的实验效果。