**《单片机原理及应用》课程大作业**

题目： 灯光管理系统

班级： 物联网1902班

学号： 0121902960525

姓名： 周航

分数：

1. 系统概述

对系统功能进行描述：

该系统模拟的是高楼灯光管理系统：

现实生活中每到了夜晚，一些高楼都会开启五颜六色的灯，通过控制灯的亮灭组成不同的图案或是文字，本次实验我模拟就是这样一个简化版本的系统，具体功能如下：

1. 打开电源开关，LED1和LED2以一个固定的周期进行亮灭（例如周期为3s，则亮2s,熄1s），同时向串口发送系统启动的相关信息。这里模拟高楼外的流水灯亮灭，以及向终端发送信息.

2、按下KEY1，LED1灯的周期加一。按下KEY2, LED2灯的周期加一。控制流水灯的亮灭组合，以达到不同的视觉效果。同时向串口发送相关操作的信息。（如果有更多的按键，也可以配置按键为周期减一，这样操作更方便）这里模拟的是高楼外灯形成不同的图案。

1. 按下KEY3，控制LED3灯进行亮灭。同时输出相关操作信息，并检测KEY3 是否被按下。这里模拟的是手动开启高楼外特定的灯光。

4、在灯亮灭的过程中还有蜂鸣器的响声，模拟某种提示信号。

1. 系统硬件框图

写明用了什么接口，和用什么部件进行了什么模拟：

本实验所用接口：

1. 按键：

KEY1,KEY2,KEY3分别代表PA8,PD8,PA4。LED1,LED2,LED3分别代表PD5,PD6,PD7。

KEY1和KEY2模拟的是灯的周期控制器，按下能使LED1和LED2的周期加一。KEY3模拟的是LED3的开关，它是采用中断的方式实现，控制LED3的翻转，模拟特定灯光的亮灭。LED1,LED2,LED3模拟灯。

1. 中断：

KEY3代表的引脚PA4配置成中断。其功能见（1）

1. 串口：

配置引脚PA9和PA10，配置USART1，实现数据的接收和发送

其功能是每进行一个操作就向串口调试助手发送相关的操作信息。

这模拟的是向终端发送相应的操作信息。

1. ADC：

这里配置的GPIO引脚是KEY3（PA4），配置ADC1。

采集的数据是PA4通道外部的电压值的模拟数字量。

并通过串口发送至界面，这里可以通过采集到的AD值判断KEY3是否按下，如果已经按下，则其输出的AD值为0.

1. 定时器：

配置定时器TIM2，

这里TIM2设置的周期为1s，采用定时器控制LED1和LED2的亮灭

LED1和LED2的周期是定时器周期的倍数。

1. 蜂鸣器：

配置的引脚为PD5

当定时器触发一个中断事件是，蜂鸣器就会响以下。

主要是一个提示的功能。

1. 系统软件功能实现

重点是main函数部分，以及系统功能逻辑：

核心代码部分：

定义的全局变量：

u8 count1=1; //计数

u8 count2=1; //计数

u8 cycle1=2; //LED1在一个周期内亮的时间，固定LED1每周期熄1s

u8 cycle2=3; // LED2在一个周期内亮的时间，固定LED2每周期熄1s

uint16\_t AD\_value; //采集的AD值

u8 flag=0;

main函数部分：

int main(void)

{

RCC\_Configuration(); //配置时钟，开启所用外设的时钟

ADC\_Init\_Configuration(); //初始化ADC1

USART\_Configuration(); //初始化USART1

kEY\_Init\_Configuration(); //配置KEY3的中断

NVIC\_Configuration(); //配置中断优先级

GPIO\_Configuration(); //初始化GPIO

TIMER\_Configuration(); //配置定时器TIM2

TIM\_Cmd(TIM2,ENABLE); //开启定时器2

printf("\r\n灯光管理系统\r\n"); //重新定义fputc()函数，包含stdio.h头文件，配置编译器，就可以用printf输出

delayms(2000);

printf("\r\nLED1已经开启，当前周期%d秒\r\n",cycle1+1);

delayms(1000);

printf("\r\nLED2已经开启，当前周期%d秒\r\n",cycle2+1);

delayms(1000);

while(1)

{

//如果检测到KEY1被按下LED1的周期加一，同时向串口发送信息

if(GPIO\_ReadInputDataBit(KEY1\_3\_PORT,KEY1)==KEY\_ON) //检测KEY1是否被按下

{

delayms(100); //延时消抖

if(GPIO\_ReadInputDataBit(KEY1\_3\_PORT,KEY1)==KEY\_ON)

{

cycle1++;

printf("\r\nLED1的周期加一,当前周期为%d秒\r\n",cycle1+1);

}

}

//如果检测到KEY2被按下LED2的周期加一，同时向串口发送信息

if(GPIO\_ReadInputDataBit(LED\_PORT,KEY2)==KEY\_ON)

{

delayms(100);

if(GPIO\_ReadInputDataBit(LED\_PORT,KEY2)==KEY\_ON)

{

cycle2++;

printf("\r\nLED2的周期加一,当前周期为%d秒\r\n",cycle2+1);

}

}

}

}

定时器中断处理部分：

void TIM2\_IRQHandler(void)

{

u8 ReadValue1;

u8 ReadValue2;

if(TIM\_GetITStatus(TIM2,TIM\_IT\_Update)!=RESET) //检测是否发生溢出更新事件

{

TIM\_ClearITPendingBit(TIM2,TIM\_FLAG\_Update); //清除TIM2的中断待处理位

ReadValue1=GPIO\_ReadOutputDataBit(GPIOD,LED1); //将LED1管脚输出数值写入ReadValue1

ReadValue2=GPIO\_ReadOutputDataBit(GPIOD,LED2); //将LED2管脚输出数值写入ReadValue2

if(ReadValue1==0) //时钟控制LED1

GPIO\_SetBits(GPIOD,LED1);

else

{

if(count1<cycle1) //判断计数是否到达周期

{

count1++;

}

else

{

count1=1;

GPIO\_ResetBits(GPIOD,LED1);

}

}

if(ReadValue2==0) //时钟控制LED2

GPIO\_SetBits(GPIOD,LED2);

else

{

if(count2<cycle2)

{

count2++;

}

else

{

count2=1;

GPIO\_ResetBits(GPIOD,LED2);

}

}

//蜂鸣器响

GPIO\_SetBits(BEEP\_PORT,BEEP);

delayms(1);

GPIO\_ResetBits(BEEP\_PORT , BEEP);

delayms(1);

}

}

KEY3中断的处理函数：

void EXTI4\_IRQHandler(void)

{

if ( EXTI\_GetITStatus(EXTI\_Line4) != RESET )

{

EXTI\_ClearITPendingBit(EXTI\_Line4);

/\*如果KEY3被按下，则需要做以下事情\*/

if(GPIO\_ReadOutputDataBit(LED\_PORT,LED3))

{

GPIO\_ResetBits(LED\_PORT,LED3);

//重新定义fputc()函数，并包含stdio.h头文件，就可以用printf输出

printf("\r\nLED3熄灭\r\n");

}else

{

GPIO\_SetBits(LED\_PORT,LED3);

printf("\r\nLED3点亮\r\n");

}

while(flag<10) //每次输出10个接收到的AD值

{

if(ADC\_GetFlagStatus(ADC1, ADC\_FLAG\_EOC) != RESET)

{

AD\_value = ADC\_GetConversionValue(ADC1);

printf("The Current AD Digital Value = %i \r\n" , AD\_value);

}

delayms(500);

flag++;

}

flag=0;

}

}

1. 系统测试

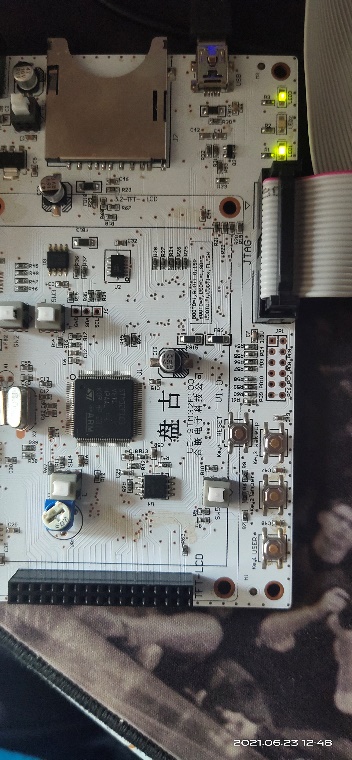
功能方面的测试，给出一些结果图：

（1）写入程序，打开开关和串口助手：



实验现象正确。

（2）按下KEY1或KEY2：



观察LED1和LED2，其亮灭周期符合串口显示的数值。

（3）按下switch按钮后按下KEY3：

观察到LED3点亮，同时串口输出相关信息，现象正确。

（4）连接电源适配器，每隔1s可以听见蜂鸣器的声音，现象正确。

1. 小结：

本次实验我自主开发了一个小系统，巩固了课程所学的内容。同时我积极动手实践，将课上的理论付诸于实践，这不仅锻炼了我的实际动手动脑能力，也激发我对计算机底层硬件的兴趣。但是在课程设计的过程中明显感到自己知识的欠缺，导致自己的构想并没有完全实现，我还需要继续学习新的知识来完善自己。