嵌入式手札四

# 实验一：利用SysTick模块进行不同闪烁模式小灯的设计

## 实验目的：

1、利用SysTick模块和UART模块 设计五种以上的不同闪烁模式流水灯用来识别不同情况；

2、输入数字（1-5）分别对应5种不同闪烁模式——代表连续闪烁对应的（1/2/3/4/5）次数后暂停1.5S后继续闪烁相应的次数，如此循环往复。

## 实验环境：

1）PC机一台，KL25开发板一片，导线若干

2) KDS开发环境，VS2017或更高平台

## 实验内容：

### 1）：设计串口接收模块UART

#### #1目的：

通过设计串口接收模块来接受输入的数字且进行判断：1-5分别对应不同的闪烁模式

#### #2问题：

##### 1）如何传递闪烁模式选择符：（tickCount）

@方法：

通过设置静态全局变量：uint\_8 tickCount = 1; 传递模式选择标识到SysTick模块中

@源码：

**void** **UART1\_IRQHandler**(**void**) {

uint\_8 ch;

uint\_8 flag;

uint\_8 buffer[64];

DISABLE\_INTERRUPTS;

**if** (uart\_get\_re\_int(UART\_1)) {

ch = uart\_re1(UART\_1, &flag);

**if** (ch == '1') {

tickCount = 1;

modeChange=1;

light\_control(LIGHT\_BLUE, LIGHT\_ON);

} **else** **if** (ch == '2') {

tickCount = 2;

modeChange=1;

light\_control(LIGHT\_BLUE, LIGHT\_ON);

} **else** **if** (ch == '3') {

tickCount = 3;

modeChange=1;

light\_control(LIGHT\_BLUE, LIGHT\_ON);

} **else** **if** (ch == '4') {

tickCount = 4;

modeChange=1;

light\_control(LIGHT\_BLUE, LIGHT\_ON);

} **else** **if** (ch == '5') {

tickCount = 5;

modeChange=1;

light\_control(LIGHT\_BLUE, LIGHT\_ON);

} **else** {

tickCount = 1;

modeChange=1;

light\_control(LIGHT\_BLUE, LIGHT\_ON);

}

}

**if** (flag) {

uart\_send1(UART\_1, ch);//若不是1-5，默认进入闪烁模式1

}

ENABLE\_INTERRUPTS;

}

### 2）设计SysTick模块

#### #目的：

控制每种闪烁模式下的闪烁情况：包括密集闪烁模式下的密集闪烁次数和闪烁间隔

#### #问题：

##### 1）如何控制不同模式对应不同的闪烁模式

@方法：通过if\_-esle if -else 选择判断语句来控制不同闪烁模式下的选择

##### 2）如何控制每种闪烁模式下的密集闪烁模式和停止闪烁间隔之间切换

@方法：通过在设置一个全局的静态变量控制

**static** uint\_8 modeChange = 0; //模式切换标志，用于切换不同闪烁子模式

modeChange=1时，密集闪烁中

modeChange=0时，闪烁间隔

##### 3）如何控制每次不同闪烁模式下的密集闪烁次数

@方法：通过设置一个控制闪烁次数的方法内的静态变量：

**static** uint\_8 twinkleCount = 0; //密集闪烁指示灯次数变量

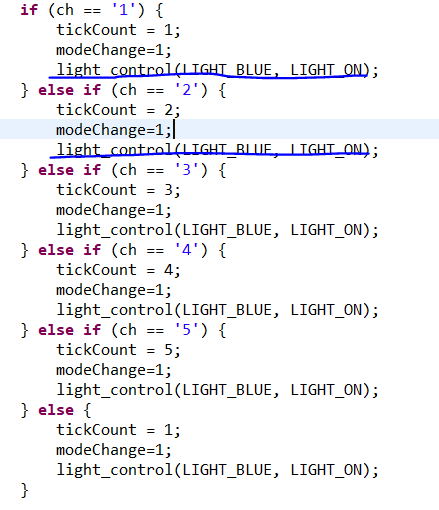
来控制闪烁次数；

##### 4）如何正确控制每种闪烁模式开启

@方法：

除了第一种只闪烁一次的情况（这种情况亮灭间隔相同不需要区分密集闪烁）；其他控制密集闪烁的闪烁次数都遵循一个公式：twinkleCount <= (tickCount \* 2 - 2)

1、每次在串口输入模式选择符后，都会设置小灯变亮



这为了符合密集闪烁的第一种状态是亮的状态

2、每次在串口输入模式选择符后，都会设置modeChange=1；

这是为了先进入每种闪烁模式下的密集闪烁模式，配合上面的控制灯亮就可以设置正确的闪烁模式了

##### 5）如何控制密集闪烁模式和熄灭间隔是1.5s：

@方法：

密集闪烁模式最后一次状态变换时亮变暗：SysTickcount=30 即0.3S改变状态一次；为了剩余的1.5s间隔；需要在modeChange=0的情况下设置 SysTickcount<=150 的间隔用来保持1.5s的灯暗状态的时间。1.5s后才会让灯重新切换为亮状态；**这样保证了每次密集闪烁的初始状态始终为亮；**

@源码：

**void** **SysTick\_Handler**(**void**) //深入了解中断的内涵

//默认值为100,则表示1S闪一次

{

**static** uint\_8 SysTickcount = 0; //静态变量,负责控制每次亮灭间隔的时间

**static** uint\_8 twinkleCount = 0; //密集闪烁指示灯次数变量

SysTickcount++;

**if** (tickCount == 1) {

**if** (SysTickcount >= 30)

{

SysTickcount = 0;

//秒计时程序

SecAdd1(g\_time); //g\_time是时分秒全局变量数组

}

} **else** **if** (tickCount == 2 || tickCount == 3 || tickCount == 4

|| tickCount == 5) {

**switch** (modeChange) {

**case** 1: {

**if** (twinkleCount <= (tickCount \* 2 - 2)) {

**if** (SysTickcount >= 30){

SysTickcount = 0;

//秒计时程序

SecAdd1(g\_time); //g\_time是时分秒全局变量数组

twinkleCount++;

}

} **else** {

twinkleCount = 0; //初始化密集闪烁次数

modeChange = 0; //更改闪烁指示标志

}

}

**case** 0: {

**if** (SysTickcount >= 150) //设置等待时间，足够区分闪烁模式

{

SysTickcount = 0; //1.5s以后，小灯延续了1.5s的暗状态才会切换为亮状态

//秒计时程序

SecAdd1(g\_time); //g\_time是时分秒全局变量数组

modeChange = 1; //更改闪烁指示标志

}

}

}

}

**else**{}

}

## 实验总结：

1、这次试验初步理解了systick模块的使用，定时器每次的系统中断产生的时间间隔是不变的，只是通过改变 SysTickcount 的次数来改变g\_time 数组的秒元素递增来间接控制小灯的亮灭切换的

2、本次实验有相当一部分需要比较扎实的编程功底，特别是对整个闪烁模式的选择——tickcount和闪烁密集模式和熄灭间隔之间的切换标志 modeChange ；需要设置合理的静态变量来进行控制！

3、对PWM还是很生疏，需要花许多时间学习！