# 北京邮电大学软件学院

# 2020-2021学年第1学期实验报告

**课程名称： 嵌入式系统开发**

**实验名称： 有限状态机模型下的单Task多Instance开发**

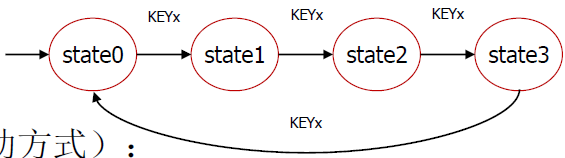
**实验完成人：**

**指导教师：**\_\_\_\_\_\_\_\_\_\_\_\_邝坚\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**日 期： 2020 年 12 月 26日**

1. **实验内容**

* 基于有限消息驱动的有限状态机设计思想，实现单个Task（ LEDx\_Task）对3个LEDx（x=1..3）的状态控制；
* 3个Led共用相同的状态机，即在给定状态下行为相同，但各自状态机可异步运行（多Instance）。三个相同的状态机，一个任务，三个状态机使用同一个代码。一个task，三个状态机
* LEDx状态：

1. State0：LEDx慢闪（初始状态）key和计时器事件
2. State1：LEDx快闪
3. State2：LEDx常亮
4. State3：LEDx灭

* 消息驱动（多Instance驱动方式）：

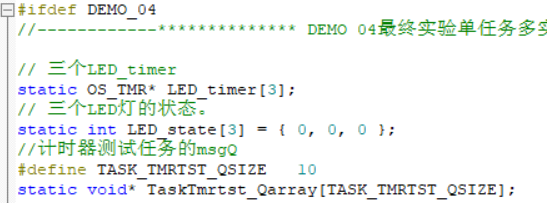
1. 用KEYx （x=1..3）事件（消息）驱动LEDx（x=1..3），即使用KEYx分别控制对应LEDx 的状态转移。
2. Timer亦采用消息驱动控制State0/1的Ledx闪烁
3. KEY4事件（消息）强制以上3个LEDx状态机均回到State0

* Note：所提交的报告中需给出对应LEDx\_Task的基本SDL图。

1. **实验源码讲解**

【变量介绍】

**在项目文件的uCOSII\_Demo.c文件中定义以下变量**



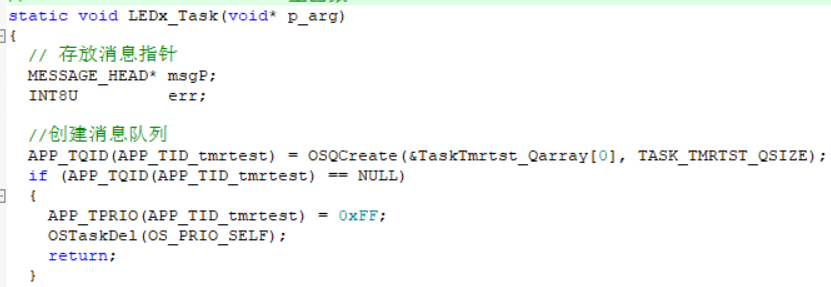
1. LED\_timer数组用于为三个LED灯计时器指针，用于创建三个LED灯实例来控制每个不同的灯
2. LED\_state数组用于存储三个LED对应的状态，三个LED使用相同的状态机来进行状态切换。
3. TaskTmrtst\_Qarray为一个消息队列，这里引用了老师给的消息队列，用于读取每个按键是否按下。

【函数介绍】

**在项目文件的uCOSII\_Demo.c文件中声明并定义以下函数**

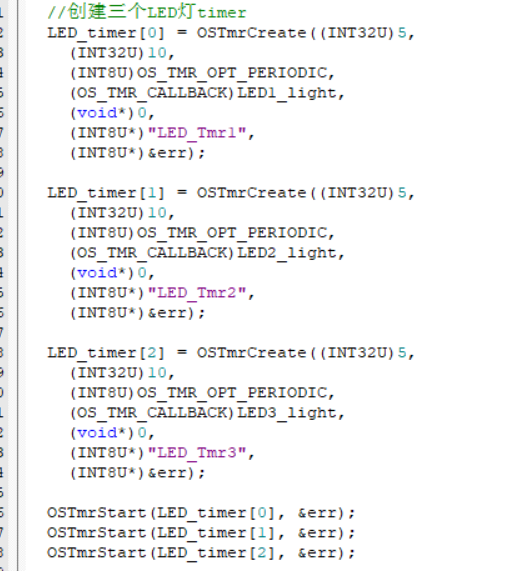
**LEDx\_Task为程序主要函数，这里将其分为三个部分讲解：**

* 第一部分



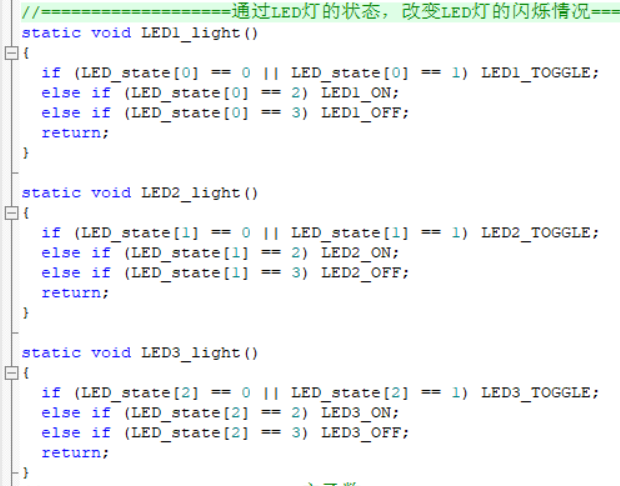
该部分主要为创建消息队列，首先创建一个消息头指针用于存放从消息队列中读取到的数据，之后使用TaskTmrtst\_Qarray创建消息队列。

* 第二部分



该部分主要为创建并运行三个LED\_timer，其中三个timer的回调函数为LED1\_light、LED2\_light、LED3\_light用于根据LED灯的状态，来改变LED灯的闪亮情况：慢闪、快闪、常亮、熄灭。

**被调用函数LED1~3\_light讲解**



该函数为是三个LED\_timer的回调函数，三个函数内容基本相同，这里只介绍第一个，LED1\_light()，函数读取LED1灯的状态，然后执行响应操作，当状态为0或1时，LED1灯电平反转，LED1闪烁；当状态为2时LED1常亮；当状态为3时LED1熄灭。

* 第三部分

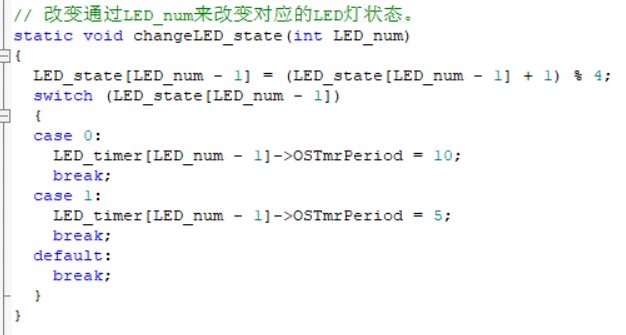


在创建完LED\_timer后，开始循环检测按键KEY是否被按下，以根据按下的KEY值执行响应操作。首先从消息队列中读取数据，然后使用switch语句根据获取的按键信息改变对应的LED灯状态，例如当读取到key1被按下时，执行changeLED\_state(1)函数，改变LED1灯的状态，key2、key3也是如此。



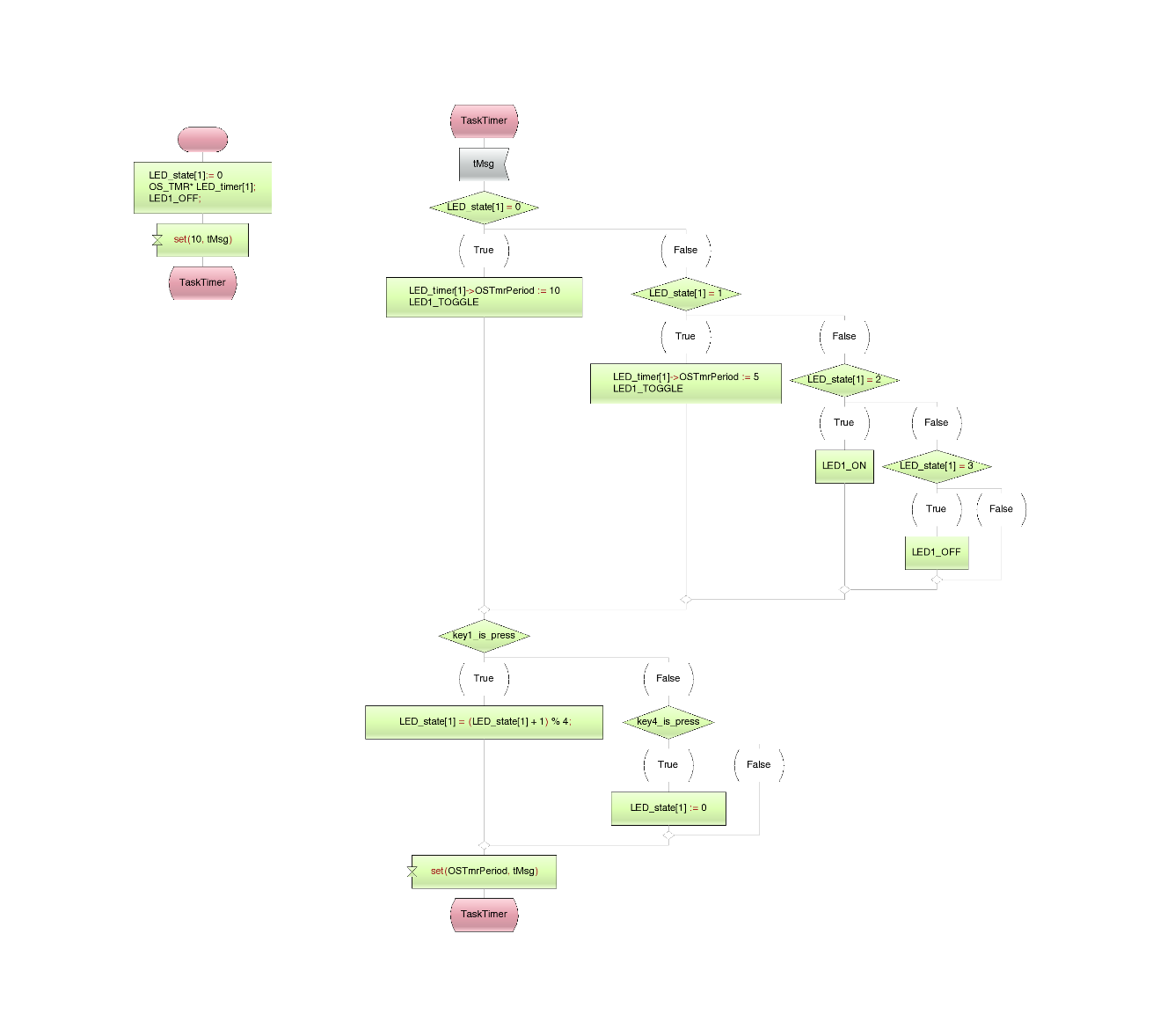
当程序判断出key4键被按下时，首先关闭所有的LED灯，使三个LED灯保持一致，然后，设置其状态都为0，将三个LED\_timer计时器的周期时长都改为状态为0下的周期时长。在执行swich语句后，执行Msg\_MesPut(msgP函数)释放消息信号。

被调用函数**changeLED\_state()讲解**



该函数传入参数为LED\_num对应为LED灯的编号，该函数根据LED\_num来改变对应的LED灯的状态，首先执行LED\_state[LED\_num - 1] = (LED\_state[LED\_num - 1] + 1) % 4; 语句，改变状态，状态号为0、1、2、3，当状态号为0时，改变对应的LED\_timer周期为10(慢闪)，当状态号为1时，改变对应的LED\_timer周期为5(快闪)。

1. **SDL图讲解**



首先进入开始状态，初始化变量，然后设定计时器，在SDL里有专门的计时器图框。然后当计时器触发后判断LED\_state[1]的值，这里默认选择第一个LED灯为演示，当值为0、1、2、3对应不同的灯亮状态，分别为：慢闪、快闪、常亮、熄灭、执行完对应的状态后重新设定计时器，等待计时器再一次触发。

1. **具体LED灯实验效果和更详细的代码讲解请看演示视频。**