[本模板中红色中括号内是用法提示，成文后请务必删除]

[电子文件命名规则举例： 工程实践与科技创新II-A\_第09A组（组长张三）基础实验报告 ]

[为方便排序，电子文件名中组号请填满2位，组号后缀A表示“发送机小组”，B是“接收机小组”]

目 录

1. 概述 1

1.1 编写说明 1

1.2 名词定义 1

1.3 硬件开发环境 1

1.4 软件开发环境 1

1.5 缩略语 1

2. 系统总述 2

2.1 系统组成 2

2.2 系统的主要功能 2

2.2.1 \*\*\*\*模块 2

2.2.2 \*\*\*\*模块 2

3. 实验1 时钟选择与GPIO操作 3

3.1 实验目的 4

3.2 实验主要器材和设备 4

3.3 任务要求、实现思路和结果分析 4

3.3.1 实验任务1\_1 4

3.3.2 实验任务1\_2 7

3.4 实验小结 8

4. 实验2 A2000TM4扩展板使用方法与SYSTICK中断实验 9

4.1 实验目的 10

4.2 实验主要器材和设备 11

4.3 任务要求、实现思路和结果分析 11

4.3.1 实验任务2\_1 11

4.3.2 实验任务2\_2 12

4.3.3 实验任务2\_3 13

4.3.4 实验任务2\_4 13

4.4 实验小结 14

5. 实验3 UART串行通讯实验 15

5.1 实验目的 16

5.2 实验主要器材和设备 16

5.3 任务要求、实现思路和结果分析 16

5.3.1 实验任务3\_1 16

5.3.2 实验任务3\_2 16

5.3.3 实验任务3\_3 16

5.4 实验小结 16

6. 致谢 17

7. 参考文献 18

8. 附录A 系统操作说明书 20

9. 附录B 测试和分析 21

9.1 测试项目和方法 21

9.2 测试的资源 21

9.3 测试结果及分析 21

10. 附录C TI工程师讲座课后小作业 22

10.1 阅读TPS40200的数据手册并回答 22

10.2 查阅DAC6571产品页面相关信息并回答 22

10.3 Webench操作练习 22

10.3.1 **观察开关频率对效率的影响** 22

10.3.2 **观察开关频率对输出电压纹波的影响** 23

10.3.3 **观察输出电容对输出电压纹波的影响** 23

11. 附录D 课程学习心得和意见建议 24

12. 附录E 软件程序清单 25

[“目录”是自动生成的，选中并按鼠标右键，选择“更新域”即可自动根据最新情况更新目次名称和页码信息]

# 概述

## 编写说明

[说明此文档的主要内容、编写目的和适读对象。]

## 名词定义

[列出在本文档中使用的需要给出明确定义的概念和关键字。]

## 硬件开发环境

[列表说明用于硬件开发的主要仪表、设备、工具以及运行环境。]

## 软件开发环境

[列表说明用于软件开发的主要开发工具以及运行环境。]

## 缩略语

[列出在本文档中缩略语的英语全称及中文说明。]

[如果某些小标题下不准备写内容，应将该标题删除；后同]

# 系统总述

## 系统组成

[系统组成框图等]

[建议将系统按逻辑功能，划分并命名为若干的“模块”，以方便说明]

[引文（图、表）必须有注释，否则视同抄袭。]

## 系统的主要功能

[对系统的各项功能进行简要阐述。]

### \*\*\*\*模块

### \*\*\*\*模块

[可根据具体情况，增加或减少章节]

# 实验1 时钟选择与GPIO操作

图1是TM4C1294NCPDT板卡上4个发光二极管灯和2个按键及对应GPIO引脚的电路原理示意图。从图上可见，比如代号D1的LED灯受控于微处理器GPIO引脚PN1（程控设置为输出引脚），当后者逻辑输出高电平时，D1被点亮，低电平则熄灭。而对代号USR\_SW1的自复位按键，如果它被操作者按下，则GPIO引脚PJ0（程控设置为输入引脚）呈现为逻辑低电平，可以被程序代码读取到“0”值；当它被释放则PJ0为高电平，程序代码读其结果为“1”值。为了更紧凑和完整地表示含义，后文以形如D1-PN1表示发光二极管D1,并标注其对应的GPIO引脚；用形如USR\_SW1-PJ0表示按键USR\_SW1，并标注其对应的GPIO引脚；等等。

图2是程序流程示意图。

例程exp0.c所实现的外部可见功能如下：

➢初始时，D4-PF0以大约1秒为周期慢速闪烁；

➢当按下USR\_SW1-PJ0，D4-PF0以大约100毫秒为周期快速闪烁；

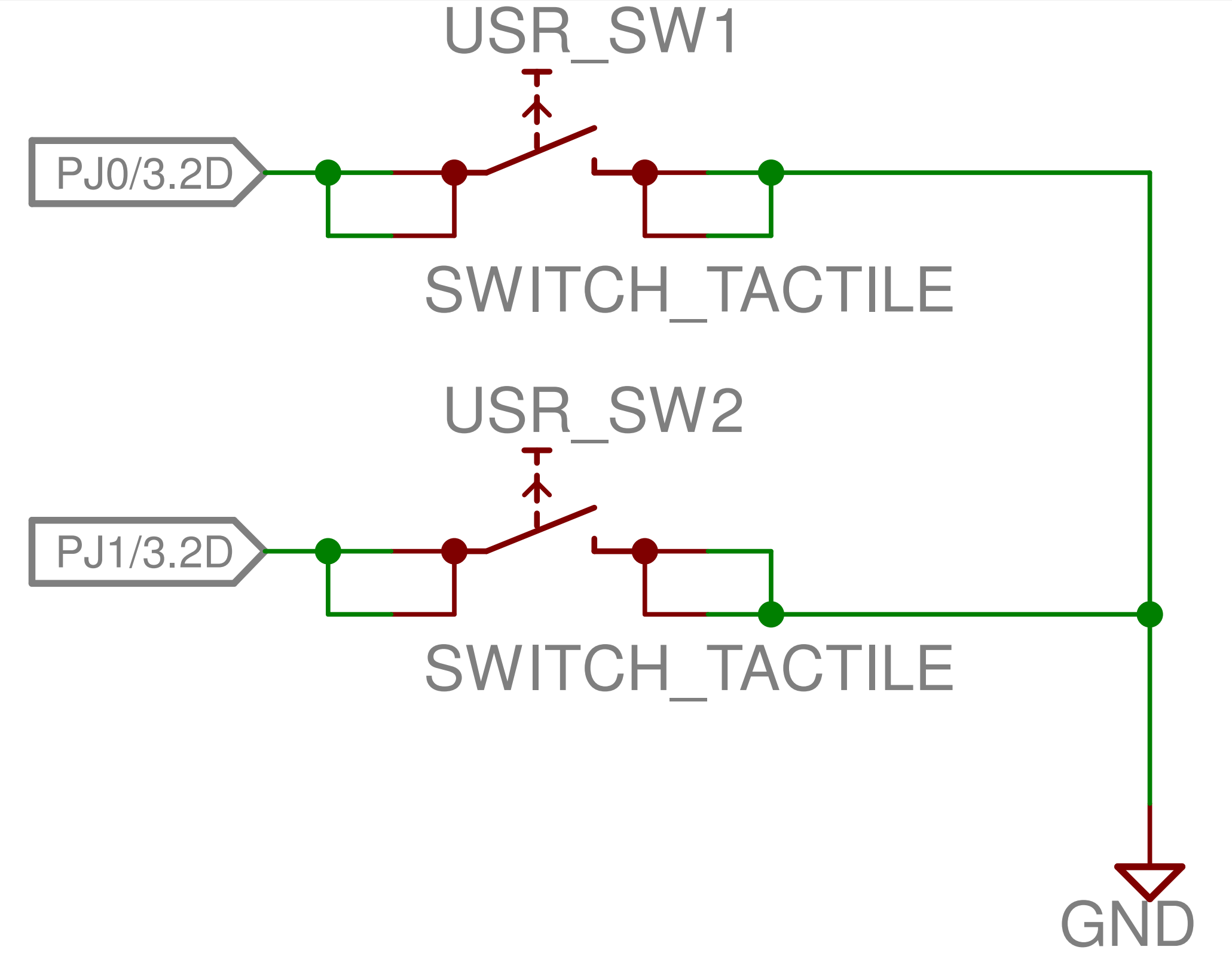
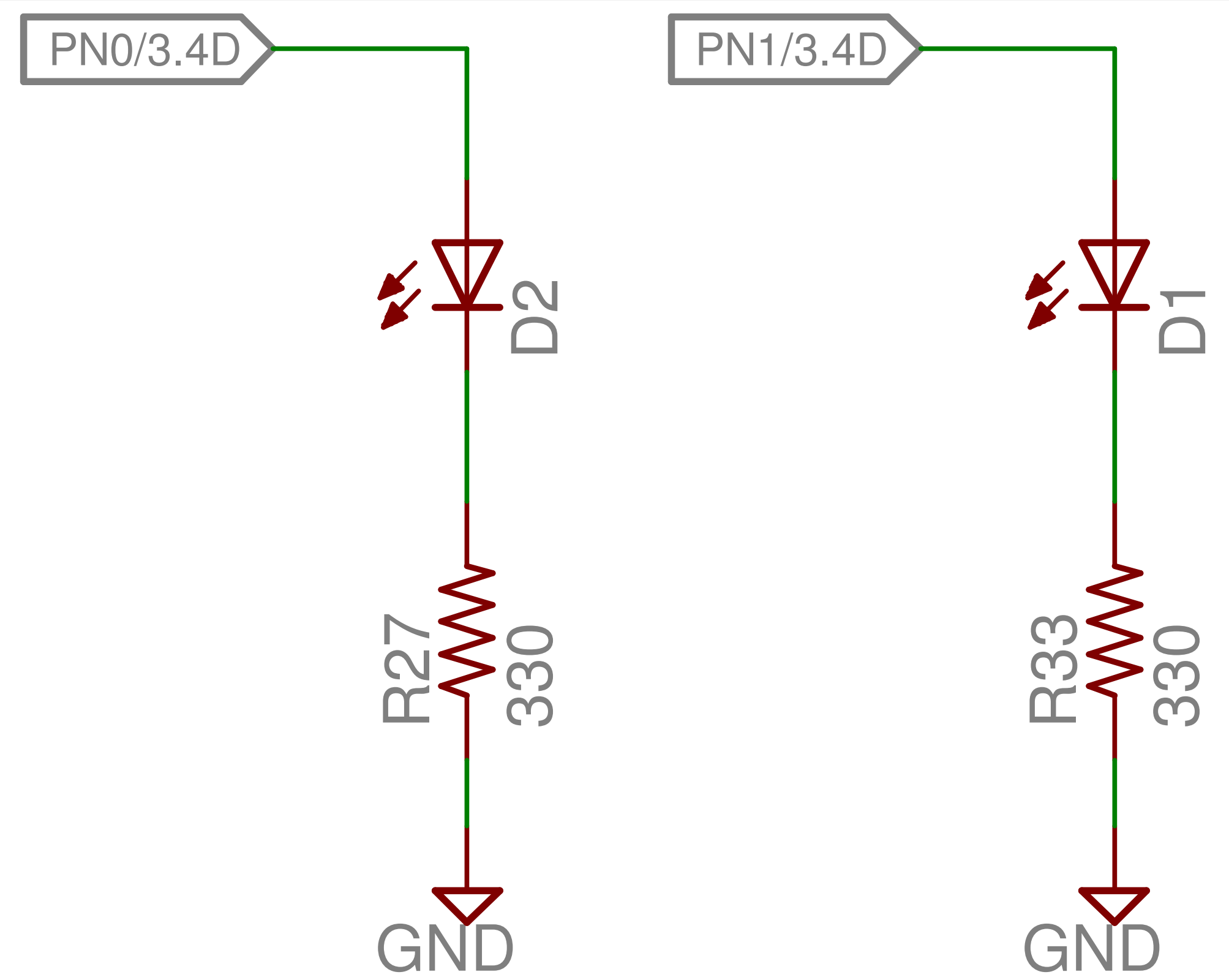
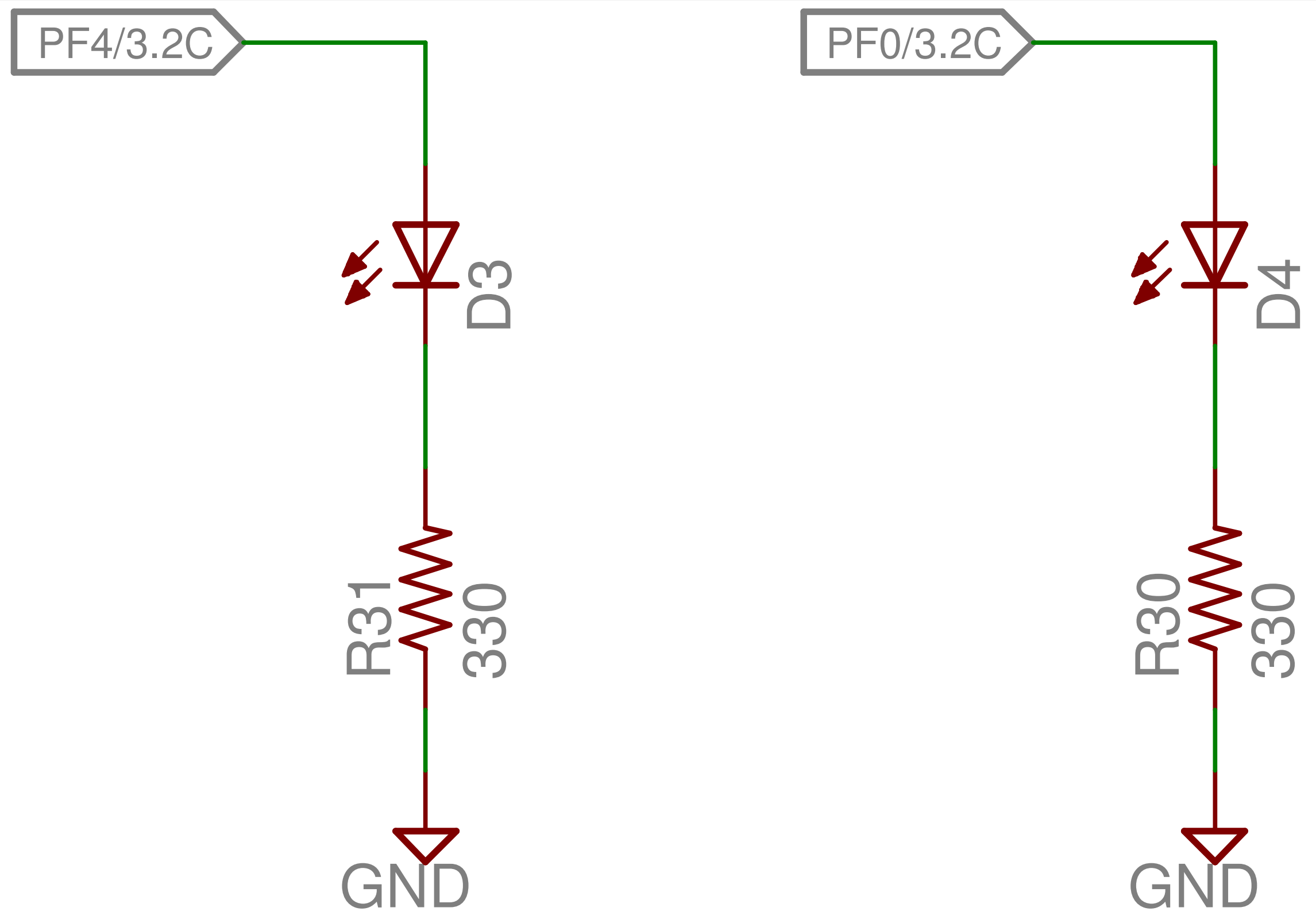
➢松开USR\_SW1-PJ0，D4-PF0恢复以1秒为周期慢速闪烁。

在例程exp0.c中，重点关注三个功能函数，分别为：

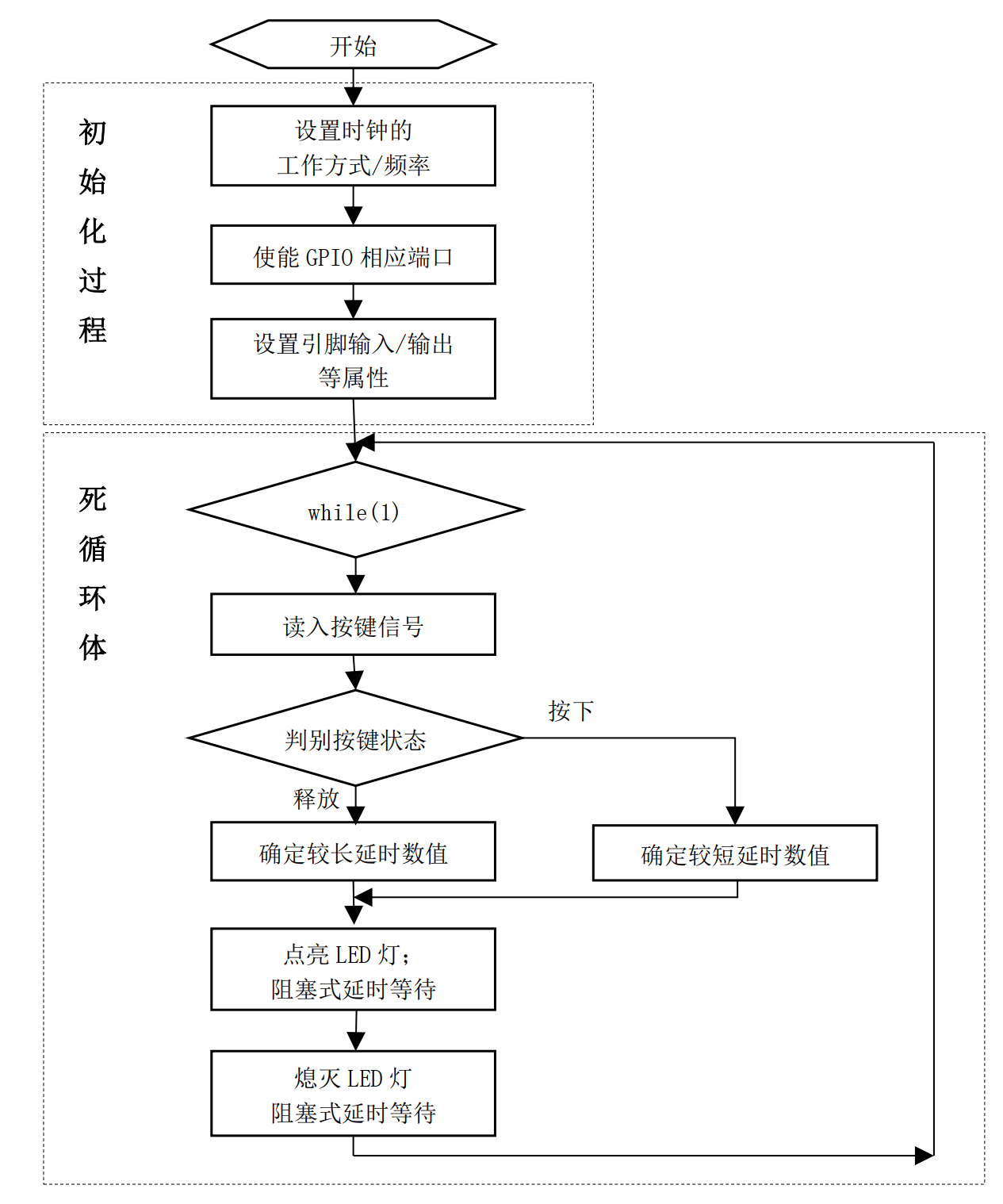
➢void DelayMilliSec(uint32\_t ui32DelaySecond); // 软件（使用空循环）延迟等待一定时长；当系统时钟为默认态时，参数单位约为毫秒，当系统时钟设置有改变，则实际等待时间会有相应改变

➢void GPIOInit(void); // GPIO初始化

➢void PF0Flash(uint8\_t ui8KeyValue); // 根据传入的按键值，决定PF0快闪或慢闪



**图1 LED指示灯和按键及其GPIO对应的原理图**[3]



**图2 exp0.c程序流程示意图**[4]

## 实验目的

（1）熟悉ARM的集成开发环境KEIL uVision5，能够自行建立一个实验用工程项目；

（2）理解CPU的时钟信号，理解软件延时的时长与系统时钟（SYSCLK）频率的关系；

（3）掌握GPIO的工作原理，能够结合GPIO的输入与输出进行实验。

## 实验主要器材和设备

电脑，TM4C1294NCPDT实验板卡，数字示波器。

## 任务要求、实现思路和结果分析

### 实验任务1\_1

#### 任务要求

修改程序exp0.c，另存命名为exp1\_1.c，完成以下任务：

1) 分别使用16M内部精密时钟（PIOSC）、25M外部时钟（MOSC）以及PLL时钟，作为程控延时等待的时钟源信号，在本程序中调节D4-PF0闪烁频率的快慢；

2) 进入集成开发环境KEIL uVision5的DEBUG模式，设置断点，显示和观察时钟频率返回值，辅助验证程序功能；

3) 使用示波器观测PF0引脚的信号波形，测量该信号频率，辅助验证程序功能；

4) 实现如下外部可见功能：初始时，D4-PF0以大约6秒为周期缓慢闪烁；当按下USR\_SW1-PJ0，D4-PF0以大约100毫秒为周期快速闪烁；松开USR\_SW1-PJ0，D4-PF0恢复以6秒为周期缓慢闪烁。

#### 实现思路

从技术文档中获取需要的信息，包括：

（1）TM4C1294NCPDT微控制器的Datasheet[1]的第5.2.5节提供的有关系统时钟控制的信息；

（2）Tiva系列MCU的外设驱动库（Peripheral Driver Library）用户手册[2]的第26.2节介绍的与系统时钟设置相关的API函数的用法；

（3）EK-TM4C1294XL板卡的用户手册[3]给出的插针（Pin）-引脚（Port）对应关系，以及板卡上的按键、LED灯等资源与端口的连接关系；

（4）课程自学材料和实验例程[4][5]给出的补充信息和编程思路。

#### 实验结果和分析

我们根据上述技术文档，结合本课程提供的自学材料，完成了实验任务。

为了探究系统时钟频率的设定方式对PF0信号频率的影响，我们进行了一系列实验，结果示于表1和图3。设计实验时，控制的变量有振荡器源（MOSC或PIOSC）和系统时钟源（OSC或PLL），共有四种组合方式，相应地可将这些实验分为四组。

观察实验现象，结合表1的数据，我们得出了以下结论。

（1）希望设定的时钟频率未必可实现。

查阅技术文档后，我们认为，这是因为实际的系统时钟是由系统时钟源（PLL或VCO）经硬件分频等处理后得到的（参见Datasheet[1]的第5.2.5节），这种机制决定了分频系数不能任意。分频系数必须为整数，例如实验序号5、7不具有整数分频系数，因而无法实现。然而，即使希望设定的系统时钟频率具有整数的分频系数，也不能保证可实现，如实验序号12就是反例。事实上，驱动库手册[2]的第26.1节已指出*When using the PLL, the input clock frequency is constrained to specific frequencies that are specified in the device data sheet.*

（2）DEBUG观测到的时钟频率与实际的时钟频率成正比，比例系数可能与振荡器源和系统时钟源有关。

从表1可见，在实验序号第1至3和第7至12中，比例系数

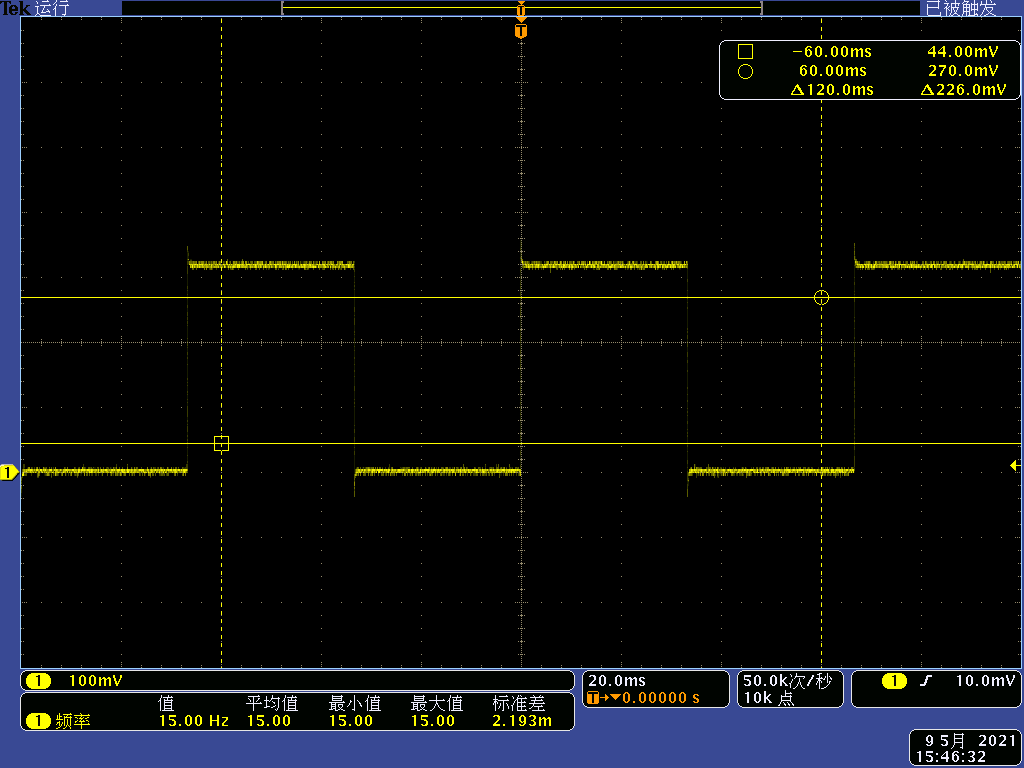


而在实验序号第4至6中，比例系数约为2.5。

根据程序流程，可认为D4-PF0的每个亮-灭周期包含相同数量的系统时钟周期，于是示波器观测的PF0频率正比于系统时钟频率。若DEBUG观测到的时钟频率与实际的时钟频率相同，则比例系数应是一个与振荡器源和系统时钟源无关的常数，然而实验结果表明并非如此。要解释这种现象，仍需查阅更多资料。

**表1 系统时钟频率的设定方式对PF0信号频率的影响**

| 序号 | 外部时钟 | 振荡器源 | | 系统时钟源 | | PLL VCO频率 | | 希望设定的系统时钟频率（MHz） | DEBUG观测到的时钟频率数值(MHz) | USR\_SW1-PJ0为释放状态下，示波器观测的PF0信号频率(Hz) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SYSCTL\_XTAL\_25MHZ | MOSC(25M)  SYSCTL\_OSC\_MAIN | PIOSC(16M)  SYSCTL\_OSC\_INT | SYSCTL\_USE\_PLL | SYSCTL\_USE\_OSC | SYSCTL\_CFG\_VCO\_320 | SYSCTL\_CFG\_VCO\_480 |
| 1 |  |  | √ |  | √ |  |  | 16 | 16 | 10.0 |
| 2 |  |  | √ |  | √ |  |  | 12 | 16 | 10.0 |
| 3 |  |  | √ |  | √ |  |  | 8 | 8 | 5.0 |
| 4 | √ | √ |  |  | √ |  |  | 25 | 25 | 10.0 |
| 5 | √ | √ |  |  | √ |  |  | 12 | 12.5 | 5.0 |
| 6 | √ | √ |  |  | √ |  |  | 1 | 1 | 0.4 |
| 7 | √ | √ |  | √ |  |  | √ | 25 | 24 | 15.0 |
| 8 | √ | √ |  | √ |  |  | √ | 20 | 20 | 12.5 |
| 9 | √ | √ |  | √ |  |  | √ | 8 | 8 | 5.0 |
| 10 |  |  | √ | √ |  |  | √ | 20 | 20 | 12.6 |
| 11 |  |  | √ | √ |  |  | √ | 8 | 8 | 5.0 |
| 12 |  |  | √ | √ |  |  | √ | 1 | 15 | 9.5 |

**图3 示波器观测的PF0信号。左图：实验序号1，右图：实验序号7**

（3）系统时钟（SYSCLK）频率越高，软件延时的时长越短。

我们认为，这是因为软件延时的原理是通过执行空循环体，用多个指令周期消耗时间；时钟频率越高，MCU的时钟周期、总线周期和指令周期就越短，所以延时的时长就越短。

（4）软件延时机制导致MCU对按键按下信号的响应出现明显迟滞。

我们注意到，在实验任务1\_1最后的外部可见功能中，按下USR\_SW1-PJ0，有时要经过明显迟滞，LED灯才能切换为快闪。我们认为，这是由于软件延时导致的。从前述软件延时原理可见，这是一种阻塞式延时：在延时期间，MCU一直在执行空循环体，只有在延时结束后才可响应按键信号。例如，若按下USR\_SW1-PJ0的时刻正好位于长延时周期的初始，将会导致约一个长延时的迟滞时间。

要解决响应迟滞的问题，可引入中断机制模拟多任务处理，详见后续章节将介绍的实验2和实验3。

### 实验任务1\_2

#### 任务要求

修改程序代码并另存命名为exp1\_2.c，完成实验任务1\_2。

外部可见功能要求如下：

(i) 当按下USR\_SW1-PJ0时，点亮D1-PN1；放开时，熄灭D1-PN1；

(ii) 当按下USR\_SW2-PJ1时，点亮D2-PN0；放开时，熄灭D2-PN0。

#### 实现思路

实验任务1\_1已实现了通过USR\_SW1-PJ0控制D4-PF0的功能，类似地可实现本实验任务。

使用SysCtlPeripheralEnable、SysCtlPeripheralReady等API函数使能端口N，用GPIOPinTypeGPIOOutput函数将PN0、PN1配置为输出引脚，用GPIOPinTypeGPIOInput函数将PJ0、PJ1设置为输入引脚，用GPIOPadConfigSet函数将PJ0、PJ1引脚设置为“推挽上拉”类型。各API函数的用法参见驱动库手册[2]，或源文件sysctl.c的注释。

在主循环体中，用以下语句实现USR\_SW1-PJ0、USR\_SW2-PJ1对D1-PN1、D2-PN0的控制：

// 读取 PJ0、PJ1 键值. 0-按下 1-松开. Any bit that is not specified by ui8Pins is returned as a 0.

ui8KeyValue = GPIOPinRead(GPIO\_PORTJ\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1);

// 若 PJ0/1 为 1（松开），则熄灭 PN1/0 （写入0）；否则.

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1, ~((ui8KeyValue << 1) | (ui8KeyValue >> 1)));

#### 实验结果和分析

实验任务已正确实现。

在驱动库的API函数中，常用bit-packed方式表示引脚及各种状态，例如本实验使用的int32\_t GPIOPinRead(uint32\_t ui32Port, uint8\_t ui8Pins) 函数。利用C语言提供的移位操作，我们写出了上述简单清晰的与按键-LED控制相关的语句。

## 实验小结

在本实验中，我们的主要工作如下：

（1）建立并熟悉开发环境。我们使用ARM的集成开发环境KEIL uVision5，建立了实验用工程项目；

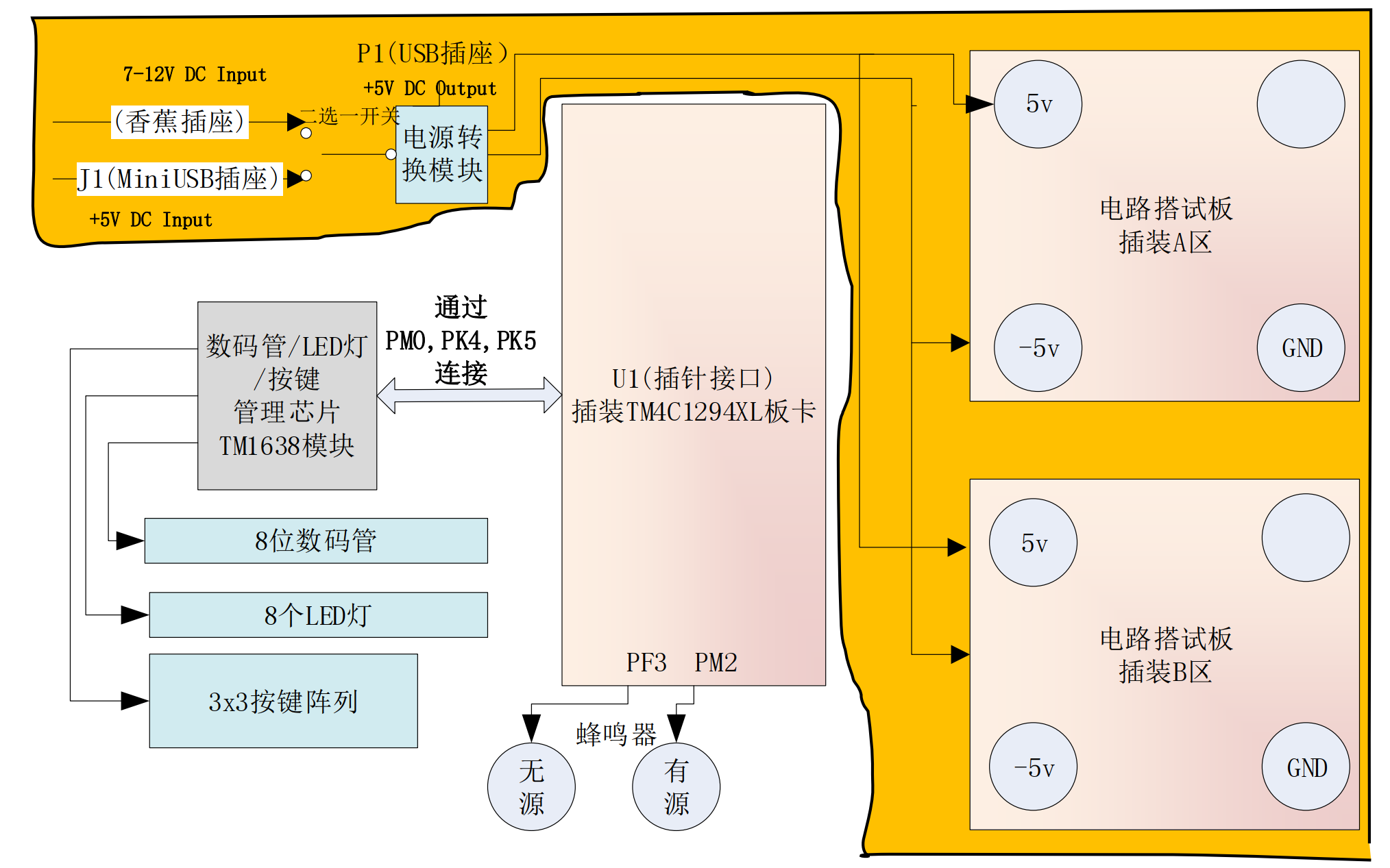
（2）使用IDE的Debug功能进行在线调试。我们利用添加断点，单步执行，使用watch窗口观察变量的值等调试功能，完成了实验任务；

（3）学习TM4C1294NCPDT微控制器的系统时钟控制原理。我们通过查阅技术文档，结合课程提供的自学材料，实现了编程设置MCU的系统时钟。我们通过实验，探讨了系统时钟频率的取值限制，软件延时的时长与系统时钟（SYSCLK）频率的关系，以及按键操作有时要经过明显迟滞才能得到响应的原因；

（4）学习GPIO的工作原理，结合GPIO的输入与输出进行实验。我们通过GPIO编程，实现了通过按键USR\_SW1-PJ0、USR\_SW2-PJ1控制发光二极管D1-PN1、D2-PN0。

（5）使用示波器观测PF0信号频率，与希望设定的系统时钟频率和DEBUG观测到的时钟频率数值作比较。

# 实验2 A2000TM4扩展板使用方法与SYSTICK中断实验



**图4 A2000TM4扩展板电路组成框图[6]**

图4展示了A2000TM4扩展板电路的组成框图。从电路组成结构看，大致可以分为供电及电源转换电路（橙色底色区域）和TM4C1294XL板卡扩展电路（其他区域）两个部分。

TM4C1294XL板卡扩展电路提供了必要的外设扩展，包括8个七段数码管、8个LED指示灯和 9个按键（排列成3×3阵列），还有1个无源蜂鸣器和1个有源蜂鸣器。通过适当编程和使用这些板载装置资源，可以为基于A2000TM4扩展板设计的实验作品提供人机交互方式，即简易的人机界面(UI)。

扩展电路以TM1638集成芯片为核心器件。在微处理器程控下，TM1638可以管理数码管显示，控制LED灯的亮灭，扫描检测按键操作情况。硬件电路中，TM4C1294微处理器通过自身GPIO引脚PM0、PK4、PK5与TM1638芯片相连，传递程序控制信号。

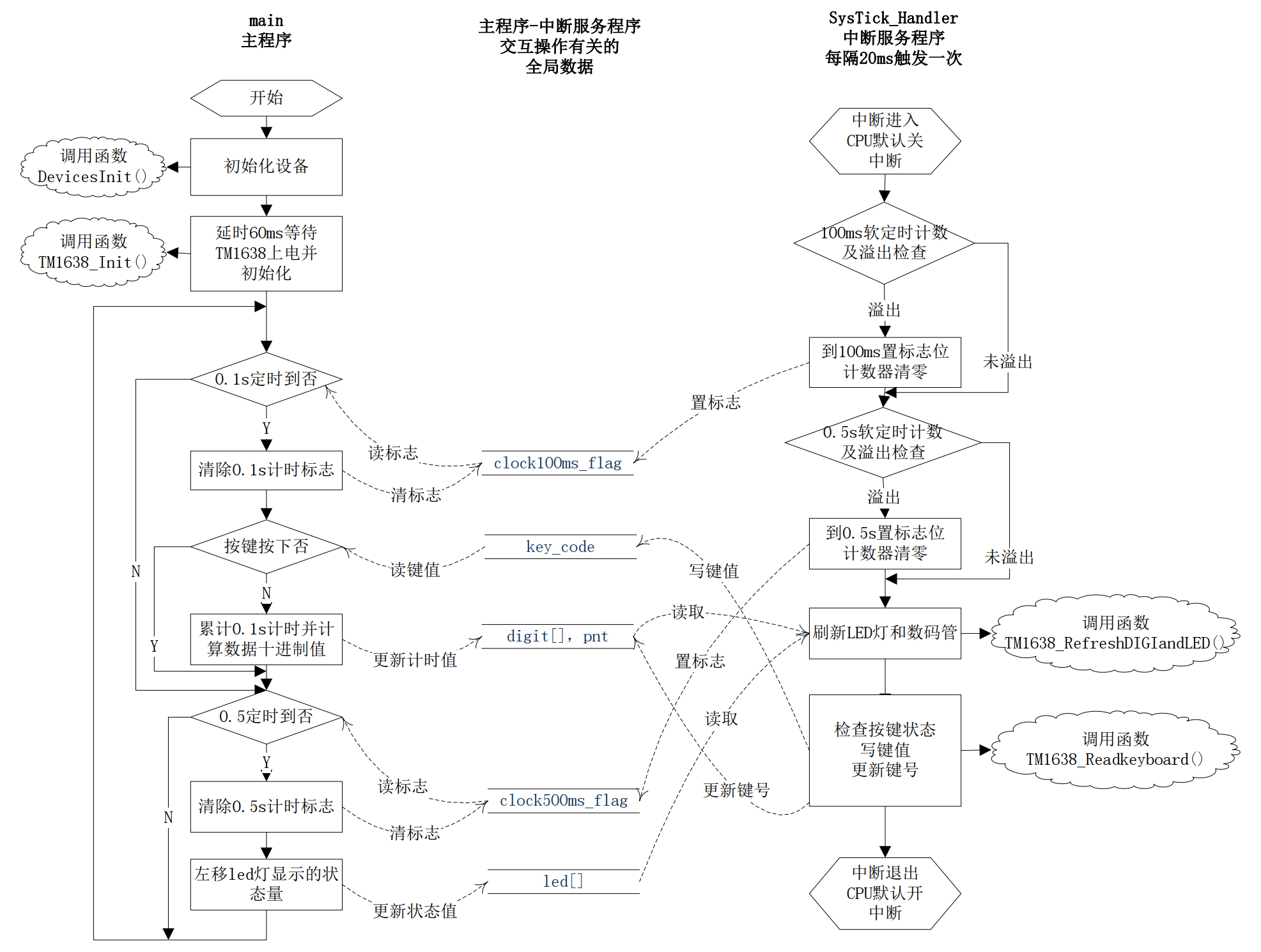
课程的示例程序以开源方式提供了关于TM1638的控制程序段（C语言函数子程序），被封装为头文件tm1638.h和源文件tm1638.c，供学习者调用和参照学习。

运行示例程序exp2\_0.c，可观察的实验板上各部分功能如下，各部分同时工作。

➢功能1：开机或复位后，底板上右四位数码管对应加计时功能，自动显示十进制计时数值，最低位的时间单位是0.1秒；每计满1000.0秒归零，重新开始加计时。且功能1会受功能3影响。

➢功能2：开机或复位后，底板上8个LED灯以循环跑马灯形式由左向右，约0.5秒变换1次。

➢功能3：利用左数第二位数码管显示“键值”（按键编号）。当没有按键按下时，显示数码“0”；当人工按下某键，数码管显示该键的编号（1到9），且此刻，功能1的四位加计时数码管暂停变化（停止计时），直到按键被放开后自动继续计时。



**图5 exp2\_0.c程序流程示意图[6]**

例程exp2\_0.c的简明流程图如图5。其中，主程序和SysTick中断服务程序被绘制成两条独立的流程线，在系统运行时两者受处理器硬件机制调度，由硬件机制决定当前应该由谁占用处理器并被运行。两者通过操作（读或写）专门的全局变量，来实现彼此之间的逻辑数据交互。比如，当中断服务程序运行时，读取到用户的按键操作情况，会把信息数据写入全局变量key\_code；然后，当主程序被调度运行时，有相应的代码会读取key\_code数据。

关于例程exp2\_0.c的更多信息，参见实验要求文档[6]。

## 实验目的

（1）熟悉LED显示和键盘管理芯片TM1638的功能，学会通过TM4C1294芯片程控该芯片的方法；

（2）初步掌握简易的人机操作界面的设计技巧（利用数码管、LED指示灯、按键阵列等）；

（3）熟悉SYSTICK中断调用方式，掌握相应的编程技能；

（4）学习和掌握在无操作系统支持的条件下，模拟多任务切换式处理的程序设计思想和方法。

## 实验主要器材和设备

电脑，TM4C1294NCPDT实验板卡，A2000TM4扩展板，数字示波器。

## 任务要求、实现思路和结果分析

A2000TM4扩展板采用的机械按键，具有“按键抖动”的固有缺陷，因此需要做“消抖”处理。在实验例程（流程示于图5）中，按键状态检测被安排在Systick中断服务程序中进行，而该中断每20ms进入一次，事实上已实现“定时扫描法”消抖，详见自学材料[7]。

### 实验任务2\_1

#### 任务要求

将exp2\_0.c另存为exp2\_1.c。以exp2\_0.c的功能为基础，修改程序，完成实验任务2\_1。

在实验任务2\_1中，要让电路呈现三种工作模式（见表2）。通过人工按动一次任意键，使电路做一次模式切换，在三种工作模式中轮转。开机或复位运行，以模式0为默认工作模式。

**表2 工作模式列表**

|  |  |
| --- | --- |
| **模式** | **功能描述** |
| 0 | 原功能1和功能3有效；功能2暂停（LED灯静止不闪动） |
| 1 | 原功能1暂停（计时）；原功能3有效；  8个指示灯以走马灯方式，每0.5秒向右（循环）移动一格 |
| 2 | 原功能1暂停（计时）；原功能3有效；  8个指示灯以走马灯方式，每0.5秒向左（循环）移动一格 |

一次人工按键操作应包括按下和放开两个过程。任意按下一键，会触发一次模式转变，但按住不放不能连续改变工作模式，放开动作也不直接影响工作模式；直至放开之后的下一次按键，才能再次触发转变。

#### 实现思路

建立一个全局变量mode，初始值为0，用来记录当前工作模式（相应于表2）。建立一个静态局部变量，用来暂存上次读到的键值。

在主程序的循环体中，将本次读到的键值与上次读到的键值作比较：若上次读到0（表示无键按下，参见文档[6]对全局变量key\_code的说明），且本次读到非0值，表示检测到一次按下过程；若上次读到非0值，且本次读到0值，表示检测到一次放开过程。

实验要求仅按下过程会直接改变工作模式，因此每检测到一次按下过程，就令变量mode自增1再模3，其余情况不改变mode的值。在主程序和Systick的ISR中，根据mode的取值执行相应的控制代码。

#### 实验结果和分析

实验任务已正确实现。

可将上述实验思路理解为一个简单的有限状态机（FSM）。共有三个状态，状态转移条件是一次按下事件。

### 实验任务2\_2

#### 任务要求

将exp2\_0.c另存命名为exp2\_2.c。以exp2\_0.c的功能为基础，修改程序，完成实验任务2\_2。

exp2\_2.c的功能要点如下。

1）原功能1继续有效，且受功能3影响。

2）取消原功能2；指定最左端LED灯作为一个状态标志，当功能1正常计数时最左端LED灯亮，功能1暂停计时则最左端LED灯灭；其余LED灯保持为全灭。

3）原功能3改为：利用左数第二位数码管显示“键值”（按键编号）；当没有按键按下时，显示数码“0”；当人工按下某键，数码管显示该键的编号（1到9）。当按一次1号键，可循环切换功能1的计时工作状态，启动或暂停（start / pause）计时（计数）功能；当按一次2号键，对功能1的计时值清零；当按下其余键时，对功能1没有影响。

#### 实现思路

建立一个静态全局变量cur\_state，用于记录当前状态号；变量的存储方式采取静态方式，是为了暂存上一状态。建立全局变量pre\_state，用于记录上一状态号。

采用有限状态机（FSM）实现实验任务。FSM规则如下：

（a）状态定义。0-计时进行；1-计时暂停；2-计时清零。

（b）状态转移条件。(C01, C10)[[1]](#footnote-1) 检测到1号键按下；(C02, C12) 检测到2号键按下；(C20, C21)完成一次状态动作。

（c）状态转移动作。(C02, C12) 保存当前状态号（通过更新全局变量pre\_state），计数值清零；(C01, C10) 保存当前状态号。

（d）状态动作。0-计时进行，最左端LED灯亮，其余全灭；1-计时暂停，LED灯全灭；2-更新数码管显示（通过修改全局变量digit[8]，参见文档[6]），然后返回上一状态号 (pre\_state)。

状态转移、状态动作都在主程序的循环体中实现，以精简Systick的ISR，保证其在一个中断周期（已设定为20ms）内执行完毕。

#### 实验结果和分析

实验任务已正确实现。

使用FSM的思想，可使程序逻辑清晰，易于维护。

### 实验任务2\_3

#### 任务要求

在实验任务2\_2基础上，修改程序exp2\_2.c代码并另存为exp2\_3.c，完成实验任务2\_3。

将功能1计时的加计数改为减计数，且计时初值为9999（注：对应999.9秒）；每计满1000.0秒，重新开始。

#### 实现思路

在实验任务2\_2的基础上做相应修改即可。

最好避免uint32\_t类型的计数器变量test\_counter发生下溢。

#### 实验结果和分析

实验任务已正确实现。

### 实验任务2\_4

#### 任务要求

在实验任务2\_3基础上，修改程序exp2\_3.c代码并另存命名为exp2\_4.c，完成实验任务2\_4。

添加使用3号按键作为加减计数模式切换控制。开机或复位后，为减计时；之后每按一次3号键，切换一次加减计数模式。

取消用左数第二位数码管显示“键值” 的功能。改以左侧 4 位数码管显示计时模式状态，即用“\_UU\_”表示加计数，显示“\_AA\_”表示减计数。

#### 实现思路

增加全局变量flag\_DecCnt表示计数方向。

按调用关系，逐级上查TM1638芯片相关API函数TM1638\_RefreshDIGIandLED、 TM1638\_DigiSegment的源代码和注释，可知相关API函数的用法，得知可向全局变量digit[8]里存'\_'，'A'，'U' 等字符。

#### 实验结果和分析

实验任务已正确实现。

## 实验小结

在本实验中，我们的主要工作如下：

（1）学习有限状态机（FSM）的思想，用其完成了实验任务。本实验要求实现的功能包含多个彼此联系的工作模式，故采用FSM的设计思路是比较自然的。

（2）学习通过TM4C1294微控制器程控TM1638芯片，通过TM1638管理A2000TM4扩展板上资源（包括LED灯、数码管和键盘）的方法。TM4C1294微处理器通过自身GPIO引脚向TM1638芯片发出控制信号。课程组提供了封装好的TM1638控制程序段供本实验调用。

（3）熟悉Systick中断调用方式，学习相应的编程技能。利用Systick中断，我们实现了相对准确的软定时功能。

（4）学习在无操作系统支持的条件下，模拟多任务切换式处理的程序设计思想和方法。中断机制的引入，在主程序和不同的中断服务程序（ISR）中安排不同的任务，可在一定程度上实现多任务处理。

（5）初步掌握了简易的人机操作界面的设计技巧。在本实验中，我们利用A2000TM4扩展板上的数码管、LED指示灯、按键阵列等资源，实现了简单的人机交互功能。

# 实验3 UART串行通讯实验

建议在本章至少包含以下内容：

（1）简述UART串口通讯的基本原理。包括UART的中断进入条件（接收，接收超时，发送）、中断处理模式（要判断中断类型，清中断标志的原因）等等，参见[工程实践与科技创新II-A\_3.1UART串口通信 (sjtu.edu.cn)](https://vshare.sjtu.edu.cn/play/f1717b5a1c71ecd3d7eaacd6d6ffac31)[8]；

（2）在任务3\_1中，简述非阻塞式收发的编程思路，采取课程指导视频[8]中的；

（3）在任务3\_2中，介绍基于有限状态机（FSM）的命令处理逻辑，说明如何解决命令收发开始和终止问题，指出我们实现了非阻塞式收发。

编程细节从略，可简单提及在命令合法性检查、命令执行、返回信息生成过程中，可利用的C标准库函数，包括strcmp，sscanf，sprintf等；

（4）在任务3\_3中，介绍如何在任务3\_2功能的基础上增加定时报时功能（FSM规则的变动，简述两个ISR和主程序循环体的配合），提及收发中文的注意事项（字符编码统一），填实验要求文档中的表3-4并解释；

（5）实验小结应呼应实验目标。

其他事项：

（1）正文使用“本文正文”样式，（2）图题表题样式统一，（3）引用尽量规范。

## 实验目的

## 实验主要器材和设备

## 任务要求、实现思路和结果分析

### 实验任务3\_1

#### 任务要求

#### 实现思路

#### 实验结果和分析

### 实验任务3\_2

#### 任务要求

#### 实现思路

#### 实验结果和分析

### 实验任务3\_3

#### 任务要求

#### 实现思路

#### 实验结果和分析

## 实验小结

在本实验中，我们的主要工作如下：

# 致谢

[对实验及论文报告工作给过指导和帮助的单位、同学、老师或其他个人，表示必要的感激和敬意]

# 参考文献

[说明本文档参考的各种资料，也包括所参考的其他项目的有关文档、资料。在前文中有些段落、图表等如属引用所列资料，因以上标方式标注，如 XXXX [3]

列写的顺序应与正文中首次引用该资料的前后顺序一致。 ]

[各类文献列写示例，依次为：[1]普通图书、[2]会议论文集、[3]科技报告（Datasheet属此类）、[4]学位论文、[5]专利文献、[6]专著中析出的文献、[7]期刊中析出的文献、[8]报纸中析出的文献、[9][10]电子文献、[11]网上电子公告（互联网网页等），成文后请将无关项目删去。

[1] Tiva™ TM4C1294NCPDT Microcontroller DATA SHEET ( DS-TM4C1294NCPDT-15863.2743 SPMS433B )[R]. Texas Instruments Incorporated, 2014.

[2] TivaWare™ Peripheral Driver Library USER’S GUIDE ( SW-TM4C-DRL-UG-2.1.4.178 )[R]. Texas Instruments Incorporated, 2017.

[3] Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit EK-TM4C1294XL User's Guide” ( SPMU365B )[R]. Texas Instruments Incorporated, 2017.

[4] 上海交大电子工程系. 实验1\_时钟选择与GPIO操作[EB/OL]. <https://oc.sjtu.edu.cn/courses/30811/files/2446875?module_item_id=431458>.

[5] 上海交大电子工程系. 工程实践与科技创新II-A\_1.5常用调试方法[EB/OL]. <https://vshare.sjtu.edu.cn/play/b9c32627cc259b7309177662d3e6f0ce>.

[6] 上海交大电子工程系. 实验2\_A2000TM4扩展板使用方法与SYSTICK中断实验[EB/OL]. https://oc.sjtu.edu.cn/courses/30811/files/2446876?module\_item\_id=431459.

[7] 上海交大电子工程系. 工程实践与科技创新II-A\_2.2定时扫描法检测按键[EB/OL]. <https://vshare.sjtu.edu.cn/play/cbc1b104598c0259b9a4de6087421c08>.

[8] 上海交大电子工程系. 工程实践与科技创新II-A\_3.1UART串口通信[EB/OL]. https://vshare.sjtu.edu.cn/play/f1717b5a1c71ecd3d7eaacd6d6ffac31.

[1] 蒋有绪,郭泉水,马娟,等. 中国森林群落分类及其群落学特征[M]. 北京:科学出版社,1998:11-12.

[2] 中国力学学会. 第3届全国实验流体力学学术会议论文集[C]. 天津:\*\*出版社,1990:20-24.

[3] World Health Organization. Factors regulating the immune response:report of WHO Scientific Group[R].Geneva:WHO,1970.

[4] 张志祥. 间断动力系统的随机扰动及其在守恒律方程中的应用[D]. 北京:北京大学数学学院,1998:50-55.

[5] 河北绿洲生态环境科技有限公司. 一种荒漠化地区生态植被综合培育种植方法:中国，01129210.5[P/OL].2001-10-24[2002-05-28].http://211.152.9.47/sipoasp/zlijs/hyjs-yxnew. asp?recid=01129210.5&leixin.

[6] 国家标准局信息分类编码研究所. GB/T 2659-1986世界各国和地区名称代码[S]// 全国文献工作标准化技术委员会. 文献工作国家标准汇编:3.北京:中国标准出版社，1988:59-92.

[7] 李炳穆. 理想的图书馆员和信息专家的素质与形象[J]. 图书情报工作,2000(2):5-8.

[8] 丁文祥. 数字革命与竞争国际化[N]. 中国青年报,2000-11-20(15).

[9] 江向东. 互联网环境下的信息处理与图书管理系统解决方案[J/OL]. 情报学报,1999,18(2);4[2000-01-18].http://www.chinainfo.gov.cn/periodical/gbxb/gbxb99/gbxb990203.

[10] CHRISTINE M. Plant physiology:plant biology in the Genome Era[J/OL].Science,1998,281: 331-332[1998-09-23].http://www.sciencemag.org/cgi/collection/anatmorp.

[11] 上海交大电子工程系. 科技创新5讲义[EB/OL].ftp://202.120.39.248.

]

[ 若属引用而无注释，原则上视作将被判抄袭，严重扣分。请养成尊重他人原创成果的习惯！

注释方式1：文字[12] —— [12]为列在“参考文献（资料）”中的引文编号

注释方式2：在文章中直接陈述，比如

根据某文,…

某人的工作表明，…

参考文献罗列和编号顺序,一般按与在正文中首次被引用的先后次序一致。]

# 附录A 系统操作说明书

[假定你设计的是一种正式产品，本部分是为一般操作人员提供的操作使用说明书。建议图文并茂，直观易懂]

[至少应该讲清楚如下重点：

1、接线端子功能用途

2、工作模式（单体稳压电源；单体稳流电源；并联式稳压供电系统）下连线方式，输入和输出

3、人机界面的使用说明

]

# 附录B 测试和分析

## 测试项目和方法

## 测试的资源

[本节规定了为完成测试所需的设备、仪器仪表和测试环境等资源。]

## 测试结果及分析

[本节记录以上测试项目的实际测试结果和数据，并包括必要的数据处理和结果分析。]

# 附录C TI工程师讲座课后小作业

## 阅读TPS40200的数据手册并回答

TPS40200有 种封装， 输入电压范围是 V到 V。

阅读手册第1页，输入电压VIN=16V,输出电压VOUT=5V,当负载电流为0.5A时，效率是 ；

当负载电流为1.5A，VOUT=5V时，输入VIN=16V和VIN=12V比较， VIN= V时效率更高。

## 查阅DAC6571产品页面相关信息并回答

点击产品页面<https://www.ti.com.cn/product/cn/DAC6571>

DAC6571的分辨率Resolution (Bits)是 位；Output range (Max) (mA/V)值是 。

假设在项目过程中发现当前型号不满足要求，如果要换成16位同类型的DAC，可以参考的型号有： 。

## Webench操作练习

选择TPS40170作为电源控制芯片，使用WEBENCH设计一个DC-DC降压电源，输入输出参数如下.

*提示:* 在TPS40170产品页面中，<https://www.ti.com.cn/product/cn/TPS40170>

找到”设计与开发”，可直接输入相关参数。

* 输入电压：10V-20V
* 输出电压：5V
* 输出电流：1A

请按照下列步骤要求完成问题，并提交填写完成后的本文档。

### **观察开关频率对效率的影响**

进入TPS40170设计方案，修改开关频率为120kHz和480kHz，在Operation Value下分别查看两种频率下的效率值。

Efficiency @120kHz = %

Efficiency @480kHz = %

根据上述结果分析开关频率对电源效率的影响？

答：

### **观察开关频率对输出电压纹波的影响**

修改开关频率为120kHz和480kHz，在Operation Value下分别查看两种频率下的输出电压纹波值。

Vout p-p @120kHz = mV

Vout p-p @480kHz = mV

根据上述结果分析开关频率对输出电压纹波的影响？

答：

### **观察输出电容对输出电压纹波的影响**

将开关频率设为120kHz，在原理图中查看输出电容Cout的参数：

电容型号： .

电容值Cap = uF

ESR = mΩ

在原理图中将输出电容型号替换为10TPB33M（33uF，ESR=70mΩ），对比更换前和更换后的输出电压纹波。

更换前Vout p-p = mV

更换后Vout p-p = mV

# 附录D 课程学习心得和意见建议

[不属于正式内容，文体风格可比较自由随意]

# 附录E 软件程序清单

[如果有单片机程序代码等，建议采用小字体和小行距，高密度排版，提交的打印版无需包含程序清单，不必打印本附录]

1. C01表示从状态0转移到状态1，以此类推 [↑](#footnote-ref-1)