

【小蜜蜂习题库】

www.xmf393.com

CC2530 微控制器应用开发 程序设计题库

小蜜蜂老师 欧浩源



佛山市图志科技有限公司 广东职业技术学院

2021年08月12日



目 录

【CC2530 开发套件 XMF09A/XMF09B/XMF09C】简介.....	1
1. CC2530 基础外设程序设计题--CC2530 开发入门（共 1 题）.....	2
1.1 第一个 CC2530 工程.....	2
2. CC2530 基础外设程序设计题--通用 I/O 端口（共 5 题）.....	2
2.1 LED 跑马灯的实现.....	2
2.2 单按键控制灯光开关.....	2
2.3 多按键联合控制灯光开关.....	2
2.4 按键电子计数器的实现.....	3
2.5 按键控制跑马灯的启动与暂停.....	3
3. CC2530 基础外设程序设计题--外部中断（共 2 题）.....	3
3.1 外部中断控制灯光开关.....	3
3.2 外部中断控制跑马灯启动与暂停.....	4
4. CC2530 基础外设程序设计题--定时器（共 8 题）.....	4
4.1 基于定时器 1 模模式的间隔定时.....	4
4.2 基于定时器 1 的跑马灯控制.....	4
4.3 基于定时器的长按与短按.....	4
4.4 长短按键嵌套联合控制灯光.....	5
4.5 基于定时器的单击与双击.....	5
4.6 复合按键嵌套综合控制灯光.....	5
4.7 基于定时器 3 倒计时模式的间隔定时.....	6
4.8 定时器 3 和定时器 4 同时间隔定时.....	6
5. CC2530 基础外设程序设计题--看门狗（共 2 题）.....	6
5.1 看门狗定时器实现 1 秒定时.....	6
5.2 看门狗监测程序运行.....	7
6. CC2530 基础外设程序设计题--系统时钟（共 1 题）.....	7
6.1 按键控制系统时钟切换.....	7
7. CC2530 基础外设程序设计题--串口通信（共 4 题）.....	7
7.1 串口数据发送基础.....	7
7.2 统计并上报按键触发的次数.....	8
7.3 串口数据收发基础.....	8
7.4 串口命令控制灯光开关.....	8
8. CC2530 基础外设程序设计题--模数转换 ADC（共 4 题）.....	9
8.1 以查询方式循环采样外部电压.....	9
8.2 以中断方式循环采样外部电压.....	9
8.3 ADC 采样数据的电压换算.....	9
8.4 光照电压自动控制灯光开关.....	10
9. CC2530 基础外设程序设计题--脉宽调制（共 3 题）.....	10
9.1 硬件 PWM 控制灯光亮度变化.....	10
9.2 硬件 PWM 实现单路呼吸灯.....	11
9.3 硬件 PWM 实现双路呼吸灯.....	11
10. CC2530 基础外设程序设计题--低功耗（共 4 题）.....	11



10.1 基于睡眠定时器的间隔定时.....	11
10.2 利用睡眠定时器唤醒 PM2.....	11
10.3 利用外部中断信号唤醒 PM3.....	12
10.4 外部中断和睡眠定时器综合控制低功耗.....	12
11. CC2530 综合应用程序设计题-光温传感模块（共 4 题）.....	13
11.1 DS18B20 温度数据采集上报.....	13
11.2 光照与温度数据采集上报.....	13
11.3 基于串口数据监测助手的光温数据采集上报.....	13
11.4 基于串口数据监测助手的光温综合应用-单机版.....	14
【附录 1】：“嵌入式与物联网串口数据监测助手”的功能界面及通信规约.....	15

说 明

《CC2530 微控制器应用开发程序设计题库》以 CC2530 开发套件 XMF09C 为支撑平台，结合本人多年相关的课堂教学积累和应用开发经验，围绕 CC2530 微控制器的重要外设和常用知识点，由浅入深，开发了一套实训项目，共 11 个单元，38 道题目。旨在为同学们课后巩固基础要点提供训练素材，亦可作为“1+X 证书”传感网应用开发的学习强化资源。

《CC2530 微控制器应用开发程序设计题库》结合《CC2530 微控制器应用开发选择填空题库》和《CC2530 微控制器应用开发速查宝典》，可形成内容完整、案例丰富、组合多样的活页式教材。用户可以根据不同的层次，不同的要求选择合适的知识点和案例库重构教学资源。该程序设计题库中大部分案例项目详解已由小蜜蜂老师录制成“欧浩源讲《CC2530 微控制器应用开发》”全集，并免费共享于网络，恳请各位勘误、斧正。

关于 CC2530 及 Zigbee 的更多教学资源，在【小蜜蜂笔记网】整理了一个资源汇总目录，欢迎交流探讨，专题置顶栏目具体链接如下：

<https://www.xmf393.com/2019/10/20/xmf09b/>

【小蜜蜂老师简介】：欧浩源，广东人。

中国计量大学，机械设计制造及其自动化（光机电一体化），本科。

中国计量大学，计算机应用技术（嵌入式应用），研究生。

广东职业技术学院，从事物联网技术应用专业的教学与科研工作。

佛山市图志科技有限公司，致力于嵌入式开发与物联网应用。

【电子邮箱】：ohy3686@qq.com

【资源网站】：www.xmf393.com

【淘宝小店】：xmfkj.taobao.com

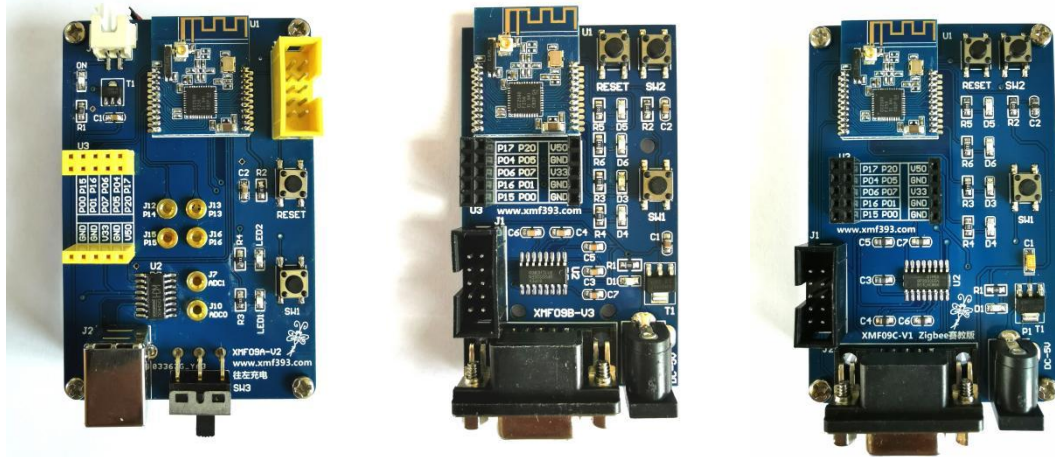
【抖音】：[ohy3686](https://www.douyin.com/user/ohy3686)

【B 站】：[小蜜蜂老师的干货铺](https://www.bilibili.com/channel/owner?id=ohy3686)



【CC2530 开发套件 XMF09A/XMF09B/XMF09C】简介

该开发套件以 CC2530 无线模块为核心，兼容全国职业院校职业技能大赛“物联网技术应用”赛项的国赛设备，对接 1+X 证书“传感网应用开发”的考试要求，面向《CC2530 微控制器应用开发》、《BasicRF 与 Zigbee 无线传感网应用开发》、《物联网硬件技术基础》等课程的课堂教学与实训拓展，针对不同的应用需求，设计有三个型号：



- 【XMF09A】：兼容国赛蓝色 Zigbee 节点盒，外观稍有不同，功能略加改善，直接替换。
- 【XMF09B】：兼容国赛黑色 Zigbee 小模块，外观尺寸一样，功能完全兼容，直接替换。
- 【XMF09C】：功能和 XMF09B 完全一样，外观和结构更合理，适合课堂教学与实训开展。

XMF09A		XMF09B		XMF09C	
LED1	P1_0	D3	P1_0	D3	P1_0
LED2	P1_1	D4	P1_1	D4	P1_1
SW1	P1_2	D5	P1_3	D5	P1_3
锂电池	1000mAh/3.7V	D6	P1_4	D6	P1_4
香蕉插座	P1_3、P1_4、P1_5	SW1	P1_2	SW1	P1_2
	P1_6、P0_0、P0_1	SW2	P0_1	SW2	P0_1
通信接口	CH340 USB 方头接口	通信接口	RS232 DB9 母头	通信接口	RS232 DB9 母头
供电方式	仿真器供电 锂电池供电 USB 口供电	供电方式	仿真器供电 5V 直流电源	供电方式	仿真器供电 5V 直流电源
15 脚传感器模块扩展口		15 脚传感器模块扩展口		15 脚传感器模块扩展口	
亚克力底板 + 软磁条				亚克力底板	
兼容国赛蓝色 Zigbee 节点盒		兼容国赛黑色 Zigbee 小模块		面向课堂教学与实训拓展	

经过多年教学积累与开发经验，形成了以 CC2530 开发套件为核心，配套教学视频、速查宝典、应用笔记、选择填空题库、程序设计题库、项目案例分析、技能大赛题解、师生交流社群等全方位教学资源的岗课赛证创生态体系。资源目录汇总，详见以下链接：

【小蜜蜂笔记网】CC2530 专题栏目：<https://www.xmf393.com/2019/10/20/xmf09b/>

【XMF09A/XMF09B/XMF09C 开发套件】购买淘宝店铺：xmfkj.taobao.com

【小蜜蜂老师】欧浩源，欢迎交流：ohy3686@qq.com



1. CC2530 基础外设程序设计题--CC2530 开发入门（共 1 题）

1.1 第一个 CC2530 工程

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D4** 灯接到 **P1_1** 引脚，**高电平亮，低电平灭**。

新建工程，设计程序，控制 **D4** 灯的循环闪烁，具体要求如下：

系统上电后，**D4** 灯点亮，过一会，**D4** 灯熄灭，过一会，**D4** 灯点亮... 如此循环往复。

注：该题的重点是：掌握在 IAR 环境下 CC2530 的开发流程。

2. CC2530 基础外设程序设计题--通用 I/O 端口（共 5 题）

2.1 LED 跑马灯的实现

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3** 灯接到 **P1_0** 引脚，**D4** 灯接到 **P1_1** 引脚，**D5** 灯接到 **P1_3** 引脚，**D6** 灯接到 **P1_4** 引脚，**高电平亮，低电平灭**。

新建工程，设计程序，实现 LED 跑马灯功能，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯相关引脚，关闭 4 个 LED 灯。

【2】LED 跑马灯的具体实现流程为：**D4** 灯点亮，延时，**D3** 灯点亮，延时，**D6** 灯点亮，延时，**D5** 灯点亮，延时，**D4** 灯熄灭，延时，**D3** 灯熄灭，延时，**D6** 灯熄灭，延时，**D5** 灯熄灭，延时。

【3】在主函数中实现循环跑马灯功能。

2.2 单按键控制灯光开关

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3** 灯接到 **P1_0** 引脚，**D4** 灯接到 **P1_1** 引脚，**D5** 灯接到 **P1_3** 引脚，**D6** 灯接到 **P1_4** 引脚，**高电平亮，低电平灭**；按键 **SW1** 接到 **P1_2** 引脚，按键 **SW2** 接到 **P0_1** 引脚。

新建工程，设计程序，实现单按键控制灯光开关，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和 2 个按键相关引脚，关闭 4 个 LED 灯。

【2】设计 LED 灯检测函数，同时点亮 4 个 LED 灯，延时一会，再同时关闭 4 个 LED 灯，模拟对 4 个 LED 灯进行功能检测。

【3】设计按键扫描函数，按键 **SW1** 按下松开后，切换 **D4** 灯的开关状态；按键 **SW2** 按下松开后，切换 **D6** 灯的开关状态。

2.3 多按键联合控制灯光开关

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3** 灯接到 **P1_0** 引脚，**D4** 灯接到 **P1_1** 引脚，**D5** 灯接到 **P1_3** 引脚，**D6** 灯接到 **P1_4** 引脚，**高电平亮，低电平灭**；按键 **SW1** 接到 **P1_2** 引脚，按键 **SW2** 接到 **P0_1** 引脚。

D4 定义为总开关指示灯，**D6** 和 **D5** 为照明灯；**SW1** 定义为总开关，**SW2** 定义为照明控制开关。新建工程，设计程序，实现多按键联合控制灯光开关，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和 2 个按键相关引脚，关闭 4 个 LED 灯。

【2】设计 LED 灯检测函数，同时点亮 4 个 LED 灯，延时一会，再同时关闭 4 个 LED 灯，模拟对 4 个 LED 灯进行功能检测。

【3】设计按键扫描函数，识别按键 **SW1** 和 **SW2** 的按下状态。

【4】第 1 次按下 **SW1** 松开后，总开关打开，**D4** 灯点亮，这时按键 **SW2** 按下有效，可以控制 **D5** 灯和 **D6** 灯的开关；再次按下 **SW1** 松开后，总开关关闭，**D4** 灯熄灭，**D5** 灯和 **D6** 灯也熄灭，按键 **SW2** 按下无效。

【5】在总开关打开的情况下，第 1 次按下 **SW2** 松开后，点亮 **D6** 灯；第 2 次按下 **SW2** 松开后，点亮 **D5** 灯；第 3 次按下 **SW2** 松开后，关闭 **D6** 和 **D5** 灯；如此往复。



2.4 按键电子计数器的实现

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

开发板上的 4 个 LED 灯定义为二进制的 4 个比特位。

D4 灯表示 Bit3，D3 灯表示 Bit2，D6 灯表示 Bit1，D5 灯表示 Bit0。

LED 灯点亮表示该位为 1，熄灭表示该位为 0，

例如：D4 亮，D3 灭，D6 灭，D5 亮，表示：1001。

新建工程，设计程序，实现一个按键电子计数器，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和 2 个按键相关引脚，关闭 4 个 LED 灯。

【2】设计 LED 灯检测函数，同时点亮 4 个 LED 灯，延时一会，再同时关闭 4 个 LED 灯，模拟对 4 个 LED 灯进行功能检测。

【3】系统由 0000 开始计数，每按下一次 SW2 按键，计数器进行一次加 1 统计，并将结果通过开发板上的 4 个 LED 灯以二进制显示。当计数到 16 时，计数器归零，即在 1111 的情况下，按键 SW2 按下一次，计数器变为 0000。

【4】在进行按键扫描时，需做好去抖动和松手检测处理。

2.5 按键控制跑马灯的启动与暂停

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现按键控制跑马灯的启动与暂停，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2 相关引脚，关闭 4 个 LED 灯。

【2】设计 LED 灯检测函数，同时点亮 4 个 LED 灯，延时一会，再同时关闭 4 个 LED 灯，模拟对 4 个 LED 灯进行功能检测。

【3】LED 跑马灯的具体实现流程为：D4 灯点亮，其余熄灭，延时；D3 灯点亮，其余熄灭，延时；D6 灯点亮，其余熄灭，延时；D5 灯点亮，其余熄灭，延时... 如此往复。

【4】当按键 SW2 按下松开后，跑马灯暂停，LED 灯保持当前的状态；再次按下 SW2 松开后，从当前状态保留处继续运行跑马灯。

【5】在按键 SW2 按下时，不能打断跑马灯的正常运行。

3. CC2530 基础外设程序设计题—外部中断（共 2 题）

3.1 外部中断控制灯光开关

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现外部中断控制灯光开关，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】初始化按键 SW2 为外部中断引脚，下降沿触发，使能相关中断控制位。

【3】在外部中断服务函数中，如果是按键 SW2 引脚产生的外部中断请求，则将 D5 灯的开关状态翻转。

【4】在主函数中，D4 灯循环闪烁，其他三个 LED 灯熄灭。



3.2 外部中断控制跑马灯启动与暂停

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现外部中断控制灯光开关，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计 LED 灯检测函数，同时点亮 4 个 LED 灯，延时一会，再同时关闭 4 个 LED 灯，模拟对 4 个 LED 灯进行功能检测。
- 【3】灯光检测完成后，进入跑马灯。跑马灯过程为：D4 灯亮，其余熄灭，延时，D3 灯亮，其余熄灭，延时，D6 灯亮，其余熄灭，延时，D5 灯亮，其余熄灭，延时…如此反复。
- 【4】初始化按键 SW2 为外部中断引脚，下降沿触发，使能相关中断控制位。
- 【5】在外部中断的服务函数中，控制跑马灯的运行与暂停互相切换。

4. CC2530 基础外设程序设计题—定时器（共 9 题）

4.1 基于定时器 1 模模式的间隔定时

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，实现基于定时器 1 模模式的秒闪灯功能，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】选择内部 16MHz 时钟的 128 分频作为计数器 1 的计数信号。
- 【3】以模模式启动定时器 1，进行 0.1 秒的间隔定时。
- 【4】在中断服务函数中，实现 1 秒的间隔定时，并翻转 D4 灯的开关状态，即 D4 灯亮 1 秒，灭 1 秒…实现 4 秒的间隔定时，并翻转 D6 灯的开关状态，即 D6 灯亮 4 秒，灭 4 秒…

4.2 基于定时器 1 的跑马灯控制

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现基于定时器 1 的跑马灯控制，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。
- 【2】选择内部 16MHz 时钟的 128 分频作为计数器 1 的计数信号。
- 【3】以模模式启动定时器 1，进行 0.1 秒的间隔定时。
- 【4】第 1 次按下 SW2 按键，D4 灯亮，0.5 秒后，D3 灯亮，0.5 秒后，D6 灯亮，0.5 秒后，D5 灯亮，0.5 秒后，全部灯灭。第 2 次按下 SW2 按键，D5 灯亮，其余灭，0.5 秒后，D6 灯亮，其余灭，0.5 秒后，D3 灯亮，其余灭，0.5 秒后，D4 灯亮，其余灭，0.5 秒后，全部灯灭，如此往复。

4.3 基于定时器的长按与短按

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现基于定时器 1 的长按与短按，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。
- 【2】选择内部 16MHz 时钟的 128 分频作为计数器 1 的计数信号。



【3】以**模模式**启动定时器 1，进行**0.1 秒**的间隔定时。

【4】当按键 SW2 长按松开后，切换 D4 灯的开关状态，当按键 SW2 短按松开后，切换 D6 灯的开关状态。

【注】按键按下时间大于**0.5 秒**，定义为长按，反之为短按。

4.4 长短按键嵌套联合控制灯光

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

D4 为总开关指示灯，D5 和 D6 为照明灯，SW1 和 SW2 为控制开关。新建工程，设计程序，实现长短按键嵌套联合控制灯光，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。

【2】选择内部 16MHz 时钟的 128 分频作为计数器 1 的计数信号。

【3】以**模模式**启动定时器 1，进行**0.1 秒**的间隔定时。

【4】长按按键 SW1，点亮 D4 灯，总开关打开，按键短按有效，可以控制 D5 灯和 D6 灯，再次长按按键 SW1，D4 灯熄灭，总开关关闭，D5 灯和 D6 灯也熄灭，按键短按无效。

【5】当总开关打开后，短按按键 SW1 松开后，切换 D5 灯的开关状态，短按按键 SW2 松开后，切换 D6 灯开关状态。

4.5 基于定时器的单击与双击

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现基于定时器 1 的单击与双击，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。

【2】选择内部 16MHz 时钟的 128 分频作为计数器 1 的计数信号。

【3】以**模模式**启动定时器 1，进行**0.1 秒**的间隔定时。

【4】单击按键 SW2，切换 D4 灯的开关状态，双击按键 SW2，切换 D6 灯的开关状态。

4.6 复合按键嵌套综合控制灯光

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

D4 为总开关指示灯，D3、D5 和 D6 为照明灯，SW1 和 SW2 为控制开关。新建工程，设计程序，实现复合按键嵌套综合控制灯光，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。

【2】选择内部 16MHz 时钟的 128 分频作为计数器 1 的计数信号。

【3】以**模模式**启动定时器 1，进行**0.1 秒**的间隔定时。

【4】长按 SW1，点亮 D4 灯，总开关打开，按键控制照明灯有效；再次长按 SW1，D4 灯熄灭，总开关关闭，按键控制照明灯无效，D3、D5 和 D6 三个照明灯熄灭。

【5】在总开关打开的情况下，短按按键 SW1，切换 D3 灯的开关状态；单击按键 SW2，切换 D6 灯的开关状态，双击按键 SW2，切换 D5 灯的开关状态。

【注】按键按下时间大于**0.5 秒**，定义为长按，反之为短按。



4.7 基于定时器 3 倒计数模式的间隔定时

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，实现基于定时器 3 倒计数模式的秒闪灯功能，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】选择内部 16MHz 时钟的 128 分频作为计数器 3 的计数信号。

【3】以倒计数式启动定时器 3，进行 1 毫秒的间隔定时。

【4】在中断服务函数中，实现 1 秒的间隔定时，并翻转 D4 灯的开关状态，即 D4 灯亮 1 秒，灭 1 秒....

4.8 定时器 3 和定时器 4 同时间间隔定时

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，实现定时器 3 和定时器 4 同时间间隔定时功能，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】选择内部 16MHz 时钟的 128 分频作为计数器 3 的计数信号。

【3】以倒计数模式启动定时器 3，进行 1 毫秒的间隔定时。

【4】以模模式启动定时器 3，进行 1 毫秒的间隔定时。

【5】在定时器 3 的中断服务函数中，每隔 1 秒翻转 D4 灯的开关状态，在定时器 4 的中断服务函数中，实现 4 秒的间隔定时，并翻转 D6 灯的开关状态。

4.9 基于定时器的红绿灯工作过程

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。

【2】选择内部 16MHz 时钟作为定时器的计数信号，分频系数根据程序设计自行选择。

【3】选择合适的定时器以及工作模式，实现按键的长按与短按，单击与双击。

【4】单击按键 SW2（按键时间不超过 1 秒）松开后，启动红绿灯工作过程，实现“D4 灯亮 5 秒，D3 灯熄灭；然后，D4 灯熄灭，D3 灯亮 2 秒”的循环过程。

【5】在红绿灯工作过程中，长按按键 SW2（按下时间超过 1 秒）松开后，D5 灯开始秒闪，即循序控制 D5 灯亮 0.5 秒，灭 0.5 秒。再次长按 SW1 按键松开后，D5 灯停止秒闪，并恢复熄灭状态，如此循环反复。在 D5 灯秒闪过程中，不能打断或影响红绿灯的正常工作。

【6】双击按键 SW2，停止红绿灯工作，即 D4、D3 和 D5 灯熄灭，系统恢复初始状态。

【7】重复实现上述红绿灯的工作过程。

5. CC2530 基础外设程序设计题--看门狗（共 2 题）

5.1 看门狗定时器实现 1 秒定时

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，利用看门狗的定时器功能实现秒闪灯，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】设计看门狗初始化函数，设置为定时器模式，定时间隔为 1 秒，使能相关中断。

【3】在看门狗定时器中断服务函数中，切换 D4 灯的开关状态，实现秒闪灯功能。



5.2 看门狗监测程序运行

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，实现看门狗监测程序运行功能，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计看门狗初始化函数，设置为看门狗模式，喂狗时间为 1 秒。
- 【3】设计 LED 灯检测函数，模拟对 4 个 LED 灯进行功能检测，先同时点亮 4 个 LED 灯，延时一会，进行看门狗喂狗，再同时关闭 4 个 LED 灯，进行看门狗喂狗。
- 【4】设计一个带喂狗功能的灯光闪烁函数，具体流程为：D4 灯点亮，延时一会，D4 灯熄灭，延时一会，看门狗喂狗。
- 【5】设计一个普通的灯光闪烁函数，具体流程为：D6 灯点亮，延时一会，D6 灯熄灭，延时一会，看门狗不喂狗。
- 【6】系统上电启动后，首先进行对 4 个 LED 灯进行检测，然后进入死循环。
- 【7】在死循环中，首先执行 8 次带喂狗功能的灯光闪烁函数，让 D4 灯闪烁 8 次；然后再执行 8 次普通的灯光闪烁函数，让 D6 灯闪烁 8 次；如此循环往复。

【运行现象】：由于普通的灯光闪烁函数没有进行看门狗喂狗，当看门狗定时器超过 1 秒后，便进行系统复位。也就是，D6 灯还没有完成 8 次闪烁，整个系统就复位了。

6. CC2530 基础外设程序设计题—系统时钟（共 1 题）

6.1 按键控制系统时钟切换

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现按键 SW1 控制系统时钟源的切换，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯和按键 SW2，关闭 4 个 LED 灯。
- 【2】设计系统时钟切换函数，可根据参数不同，进行 16MHz 和 32MHz 的时钟源切换。
- 【3】设计按键扫描处理函数，检测按键 SW2 的工作状态。
- 【4】当 SW2 按下松开后，切换系统时钟的时钟源。如果当前的时钟源为 16MHz，则切换至 32MHz；如果如果当前的时钟源为 32MHz，就切换到 16MHz。
- 【5】在主函数的死循环中，D6 灯循环闪烁，并对按键 SW2 进行扫描处理。

【运行现象】：系统上电后，系统时钟为 16MHz，D6 灯开始循环闪烁。当按键 SW2 按下松开后，系统时钟切换到 32MHz，D6 灯的闪烁速度加倍；再次按下 SW2 松开后，系统时钟切换回 16MHz，D6 灯的闪烁速度变慢。

7. CC2530 基础外设程序设计题—串口通信（共 4 题）

7.1 串口数据发送基础

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；USART0 串行接口引出至 9 针 DB9 母头。

新建工程，设计程序，通过串口向上位机发送字符串，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计系统时钟切换函数，进系统时钟切换至 32MHz 时钟源。
- 【3】设计串口 0 初始化函数，USART0 选择 UART 模式，波特率设置为 9600，I/O 引脚



映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数。

【4】设计字节发送函数，通过**查询方式**发送一个字节数据。

【5】设计字符串发送函数，发送一个完整的字符串。

【6】设计看门狗初始化函数，设置为定时器模式，定时间隔为**1 秒**。

【7】在主函数中，查询 **WDTIF** 标志位，每隔 1 秒，发送一次字符串“**Hello World!\r\n**”。

D5 灯作为数据发送指示灯，在发送字符串前点亮 **D5**，字符串发送结束后熄灭 **D5**。

7.2 统计并上报按键触发的次数

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮**，**低电平灭**；按键 **SW1** 接到 **P1_2** 引脚，按键 **SW2** 接到 **P0_1** 引脚；**USART0** 串行接口引出至 9 针 **DB9 母头**。

新建工程，设计程序，统计按键 **SW1** 触发的次数，并上报到上位机，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯和按键 **SW2**，关闭 4 个 LED 灯。

【2】设计系统时钟切换函数，进系统时钟切换至 **32MHz** 时钟源。

【3】设计串口 0 初始化函数，**USART0** 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数。

【4】设计字节发送函数，通过查询的方式发送一个字节数据。

【5】设计字符串发送函数，发送一个完整的字符串。

【6】设计按键扫描处理函数，每当 **SW2** 按下松开后，切换 **D6 灯** 的开关状态，并统计按键触发的次数，形成字符串“**按键 SW2 的触发次数为：XX\r\n**”，发送到上位机。

7.3 串口数据收发基础

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮**，**低电平灭**；**USART0** 串行接口引出至 9 针 **DB9 母头**。

新建工程，设计程序，将串口收到的数据加 1 后再发回上位机，具体要求如下：

【1】设计系统时钟切换函数，进系统时钟切换至 **32MHz** 时钟源。

【2】设计串口 0 初始化函数，**USART0** 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数，使能串口接收中断。

【3】设计字节发送函数，通过查询的方式发送一个字节数据。

【4】串口成功接收到一个字节数据后，在中断服务函数中，将接收到的数据原值加 1，然后发送回上位机。

7.4 串口命令控制灯光开关

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮**，**低电平灭**；**USART0** 串行接口引出至 9 针 **DB9 母头**。

新建工程，设计程序，接收上位机的单字节命令并控制灯光开关，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】设计系统时钟切换函数，进系统时钟切换至 **32MHz** 时钟源。

【3】设计串口 0 初始化函数，**USART0** 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数，使能串口接收中断。

【4】设计字节发送函数，通过查询的方式发送一个字节数据。

【5】设计字符串发送函数，发送一个完整的字符串。



- 【6】当接收到“0xA1”，点亮 D4 灯，操作完成后，返回“D4 is open!”。
- 【7】当接收到“0xA2”，关闭 D4 灯，操作完成后，返回“D4 is closed!”。
- 【8】当接收到“0xB1”，点亮 D6 灯，操作完成后，返回“D6 is open!”。
- 【9】当接收到“0xB2”，关闭 D6 灯，操作完成后，返回“D6 is closed!”。
- 【10】当接收到其他命令，不控制 LED 灯，返回“ERROR!!!”。

8. CC2530 基础外设程序设计题—模数转换 ADC（共 4 题）

8.1 以查询方式循环采样外部电压

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；USART0 串行接口引出至 9 针 DB9 母头。将 GM30 光温传感模块或 GM31 可调电压模块接到开发板的 15 针扩展接口，模块的电压信号输出至 AINO。

新建工程，设计程序，以查询方式循环采样 AINO 通道的数据，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计系统时钟切换函数，进系统时钟切换至 32MHz 时钟源。
- 【3】设计串口 0 初始化函数，USART0 选择 UART 模式，波特率设置为 9600，I/O 引脚映射到备用位置 1，即 P0_2 和 P0_3，UART 采用默认参数。
- 【4】设计看门狗初始化函数，设置为定时器模式，定时间隔为 1 秒。
- 【5】设计 ADC 初始化函数，配置 PO_0 为 AINO 模拟输入通道。
- 【6】设计 ADC 单次采样函数，以查询方式采样一次 AINO 通道，获得转换结果后，通过串口发送字符串“AINO 的采样结果：XXXX\r\n”到上位机。D5 灯作为采样指示灯，在开启 AINO 通道转换前点亮 D5，完成 AD 采样结果发送后熄灭 D5。
- 【7】在主函数中，循环查询 WDTIF 标志位，每隔 1 秒对 AINO 通道进行一次 AD 采样。

8.2 以中断方式循环采样外部电压

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；USART0 串行接口引出至 9 针 DB9 母头。将 GM30 光温传感模块或 GM31 可调电压模块接到开发板的 15 针扩展接口，模块的电压信号输出至 AINO。

新建工程，设计程序，以中断方式循环采样 AINO 通道的数据，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计系统时钟切换函数，进系统时钟切换至 32MHz 时钟源。
- 【3】设计串口 0 初始化函数，USART0 选择 UART 模式，波特率设置为 9600，I/O 引脚映射到备用位置 1，即 P0_2 和 P0_3，UART 采用默认参数，使能串口接收中断。
- 【4】设计看门狗初始化函数，设置为定时器模式，定时间隔时间为 1 秒。
- 【5】设计 ADC 初始化函数，配置 PO_0 为 AINO 模拟输入通道，使能中断。
- 【6】在主函数中，循环查询 WDTIF 标志位，每隔 1 秒对 AINO 进行一次 AD 采样。
- 【7】在 ADC 中断服务函数中，读取 AINO 的转换结果，通过串口发送字符串“AINO 的采样结果：XXXX\r\n”到上位机。D5 灯作为采样指示灯，在开启 AINO 通道转换前点亮 D5，完成 AD 采样结果发送后熄灭 D5。

8.3 ADC 采样数据的电压换算

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；USART0 串行接口引出至



9 针 **DB9 母头**。将 **GM30 光温传感模块** 或 **GM31 可调电压模块** 接到开发板的 15 针扩展接口，模块的电压信号输出至 **AIN0**。

新建工程，设计程序，**采样 AIN0 通道的数据并换算成电压值**，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计系统时钟切换函数，进系统时钟切换至 **32MHz** 时钟源。
- 【3】设计串口 0 初始化函数，USART0 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到 **备用位置 1**，即 **PO_2** 和 **PO_3**，UART 采用默认参数，使能串口接收中断。
- 【4】设计看门狗初始化函数，设置为定时器模式，定时间隔时间为 **1 秒**。
- 【5】设计 ADC 初始化函数，配置 **PO_0** 为 **AIN0** 模拟输入通道。
- 【6】设计 ADC 单次采样函数，以 **查询方式** 采样一次 **AIN0** 通道，获得转换结果后，取 10 位有效数据换算成电压，保留 2 位小数，通过串口发送字符串“**AIN0 的采样结果: XXXX, 电压值: X.XX V\r\n**”到上位机。**D5 灯** 作为采样指示灯，在开启 AIN0 通道转换前 **点亮 D5**，完成 AD 采样结果发送后 **熄灭 D5**。
- 【7】在主函数的死循环中，查询 **WDTIF** 标志位，每隔 1 秒对 **AIN0** 进行一次 AD 采样。

8.4 光照电压自动控制灯光开关

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3 灯** 接到 **P1_0** 引脚，**D4 灯** 接到 **P1_1** 引脚，**D5 灯** 接到 **P1_3** 引脚，**D6 灯** 接到 **P1_4** 引脚，**高电平亮**，**低电平灭**；将 **GM30 光温传感模块** 或 **GM31 可调电压模块** 接到开发板的 15 针扩展接口，模块的电压信号输出至 **AIN0**。

新建工程，设计程序，**采样 AIN0 的光照电压并据此控制灯光状态**，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】设计 ADC 初始化函数，配置 **PO_0** 为 **AIN0** 模拟输入通道。
- 【3】设计 ADC 单次采样函数，以 **查询方式** 采样一次 **AIN0** 通道，取转换结果 10 位有效数据换算成电压，保留 2 位小数，并根据光照电压自动控制灯光状态，控制要求如下：
光照电压 < **1.5V** 时，自动点亮 **D5** 和 **D6** 灯。
1.5V <= **光照电压** < **2.0V** 时，自动点亮 **D5** 灯，关闭 **D6** 灯。
光照电压 >= **2.0V** 时，自动关闭 **D5** 灯和 **D6** 灯。
- 【4】在主函数，循环采样对 **AIN0** 通道数据，并根据采样结果对灯光进行自动控制。

9. CC2530 基础外设程序设计题——脉宽调制（共 3 题）

9.1 硬件 PWM 控制灯光亮度变化

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3 灯** 接到 **P1_0** 引脚，**D4 灯** 接到 **P1_1** 引脚，**D5 灯** 接到 **P1_3** 引脚，**D6 灯** 接到 **P1_4** 引脚，**高电平亮**，**低电平灭**；按键 **SW1** 接到 **P1_2** 引脚，按键 **SW2** 接到 **PO_1** 引脚。

新建工程，设计程序，**实现硬件 PWM 控制灯光亮度变化**，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯和按键 **SW2**，关闭 4 个 LED 灯。
- 【2】定时器 1 的外设 I/O 选择 **备用位置 2**，即 **通道 2** 的 PWM 输出引脚为 **P1_0**，该引脚控制 **D3** 灯。
- 【3】选择内部 **16MHz** 时钟的 **1 分频** 作为计数器 1 的计数信号，为定时器 1 通道 2 选择合适的 **比较模式**，以 **自由运行模式** 启动定时器 1，以产生 PWM 信号。
- 【4】按键 **SW2** 按下松开后，改变 PWM 的占空比，控制 **D3** 灯的亮度变化：
第 1 次按下：PWM 占空比 **20%**，D3 灯亮 20%。
第 2 次按下：PWM 占空比 **50%**，D3 灯亮 50%。
第 3 次按下：PWM 占空比 **100%**，D3 灯全亮。



第 4 次按下：PWM 占空比 0%，D3 灯不亮。

第 5 次按下：SW2 第 1 次按下的功能，进入下一轮灯光控制，如此往复循环。

9.2 硬件 PWM 实现单路呼吸灯

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，通过硬件 PWM 控实现单路呼吸灯，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】定时器 1 的外设 I/O 选择备用位置 2，即通道 2 的 PWM 输出引脚为 P1_0，该引脚控制 D3 灯。

【3】选择内部 16MHz 时钟的 1 分频作为计数器 1 的计数信号，为定时器 1 的通道 2 选择合适的比较模式，以自由运行模式启动定时器 1，以产生 PWM 信号。

【4】在定时器 1 中断服务函数中，改变 PWM 的占空比控制灯光亮度变化。首先通道 2 的 PWM 占空比逐渐增大，D3 灯由暗渐渐变亮，当通道 2 的 PWM 占空比达到最大时，开始逐渐减小，D3 灯由亮渐渐变暗，如此循环变换 PWM 占空比，实现单路呼吸灯。

9.3 硬件 PWM 实现双路呼吸灯

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，通过硬件 PWM 控实现单路呼吸灯，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】定时器 1 的外设 I/O 选择备用位置 2，即通道 2 的 PWM 输出引脚为 P1_0，该引脚控制 D3 灯；通道 1 的 PWM 输出引脚为 P1_1，该引脚控制 D4 灯。

【3】选择内部 16MHz 时钟的 1 分频作为计数器 1 的计数信号，为定时器 1 的通道 1 和通道 2 选择合适的比较模式，以自由运行模式启动定时器 1，以产生 PWM 信号。

【4】系统启动时，通道 1 的 PWM 占空比为最小值，通道 2 的 PWM 占空比为最大值。

【5】在定时器 1 中断服务函数中，改变 PWM 的占空比控制灯光亮度变化。首先通道 1 的 PWM 占空比逐渐增大，D4 灯由暗渐渐变亮，同时，通道 2 的 PWM 占空比逐渐减小，D3 灯由亮渐渐变暗；当通道 1 的 PWM 占空比达到最大时，开始逐渐减小，D4 灯由亮渐渐变暗，同时，通道 2 的 PWM 占空比逐渐增大，D3 灯由暗渐渐变亮，如此循环变换 PWM 的占空比，实现双路呼吸灯。

10. CC2530 基础外设程序设计题--低功耗（共 4 题）

10.1 基于睡眠定时器的间隔定时

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，实现基于睡眠定时器的 1 秒定时，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】初始化睡眠定时器，设置睡眠周期为 1 秒，使能相关中断。

【3】在睡眠定时器中断服务函数中，切换 D4 灯的开关状态，设置下一次睡眠时间。

10.2 利用睡眠定时器唤醒 PM2

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭。

新建工程，设计程序，实现利用睡眠定时器唤醒 PM2，具体要求如下：



【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】在主函数的死循环中，实现以下流程：

步骤 1：点亮 D4 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 2：点亮 D3 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 3：点亮 D6 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 4：点亮 D5 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 5：关闭 4 个灯，设置 3 秒的睡眠时间，切换至 PM2 运行模式。

重复步骤 1，进入下一轮循环。

10.3 利用外部中断信号唤醒 PM3

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，实现利用外部中断信号唤醒 PM3，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】初始化按键 SW2 为外部中断引脚，下降沿触发，使能相关中断。

【3】在外部中断服务函数中做适当的处理。

【4】在主函数的死循环中，实现以下流程：

步骤 1：依次点亮 D4 灯，延时片刻。

步骤 2：依次点亮 D3 灯，延时片刻。

步骤 3：依次点亮 D6 灯，延时片刻。

步骤 4：依次点亮 D5 灯，延时片刻。

步骤 5：依次熄灭 D4 灯，延时片刻。

步骤 6：依次熄灭 D3 灯，延时片刻。

步骤 7：依次熄灭 D6 灯，延时片刻。

步骤 8：然后将运行模式切换至 PM3，进入深度休眠模式。

待外部中断信号唤醒设备后，熄灭 D5 灯，延时片刻。

重复步骤 1，进入下一轮循环。

10.4 外部中断和睡眠定时器综合控制低功耗

在 XMF09B 或 XMF09C 的 CC2530 开发板中，D3 灯接到 P1_0 引脚，D4 灯接到 P1_1 引脚，D5 灯接到 P1_3 引脚，D6 灯接到 P1_4 引脚，高电平亮，低电平灭；按键 SW1 接到 P1_2 引脚，按键 SW2 接到 P0_1 引脚。

新建工程，设计程序，利用睡眠定时器和外部中断信号综合控制低功耗的运行，具体要求如下：

【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。

【2】初始化按键 SW2 为外部中断引脚，下降沿触发，使能相关中断。

【3】在外部中断服务函数中做适当的处理。

【4】在主函数的死循环中，实现以下流程：

步骤 1：点亮 D4 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 2：点亮 D3 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 3：点亮 D6 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 4：点亮 D5 灯，设置 1 秒的睡眠时间，切换至 PM2 运行模式。

步骤 5：关闭 4 个灯，切换至 PM3 运行模式，进入深度休眠模式。



待**外部中断信号**唤醒设备后，重复**步骤 1**开始顺序执行。

11. CC2530 综合应用程序设计题-光温传感模块（共 4 题）

11.1 DS18B20 温度数据采集上报

在 XMF09B 或 XMF09C 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮，低电平灭**；**USART0** 串行接口引出至 9 针 **DB9 母头**。将 **GM30 光温传感模块**接到开发板的 15 针扩展接口，模块的**光照电压信号**输出至 **AINO**。DS18B20 温度传感器的**温度数据**输出至 **P0_1**。

新建工程，设计程序，循环采样 DS18B20 温度传感器的数据并通过串口上报到上位机，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】将 DS18B20 相关的底层驱动代码文件复制到工程文件下，并作适当的移植修改。
- 【3】设计系统时钟切换函数，进系统时钟切换至 **32MHz** 时钟源。
- 【4】设计串口 0 初始化函数，USART0 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数。
- 【5】设计 DS18B20 温度采集函数，按照工作流程进行启动温度转换，读取温度数据，进行适当的换算，保留 1 位小数，通过串口发送字符串“**温度数据：XX.X 摄氏度 \r\n**”到上位机。**D5 灯**作为采样指示灯，在温度采集前**点亮 D5**，完成温度数据发送后**熄灭 D5**。
- 【7】在主函数的死循环中，循环进行温度数据采集和上报。

11.2 光照电压与温度数据采集上报

在 XMF09B 或 XMF09C 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮，低电平灭**；**USART0** 串行接口引出至 9 针 **DB9 母头**。将 **GM30 光温传感模块**接到开发板的 15 针扩展接口，模块的**光照电压信号**输出至 **AINO**。DS18B20 温度传感器的**温度数据**输出至 **P0_1**。

新建工程，设计程序，循环采样 DS18B20 温度传感器的数据并通过串口上报到上位机，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】将 DS18B20 相关的底层驱动代码文件复制到工程文件下，并作适当的移植修改。
- 【3】设计系统时钟切换函数，进系统时钟切换至 **32MHz** 时钟源。
- 【4】设计串口 0 初始化函数，USART0 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数。
- 【5】设计 DS18B20 温度采集函数，按照工作流程进行启动温度转换，读取温度数据，进行适当的换算，保留 1 位小数。
- 【6】设计 ADC 单次采样函数，以**查询方式**采样一次 **AINO** 通道，获得转换结果后，取 10 位有效数据换算成电压，保留 2 位小数。
- 【7】将温度数据和电压数据，格式化成字符串“**温度数据：XX.X 摄氏度，光照电压：X.XX V /r/n**”，通过串口上报到上位机。
- 【8】在主函数中，每隔一小段时间，循环采集光温数据并进行上报。

11.3 基于串口数据监测助手的光温数据采集上报

在 XMF09B 或 XMF09C 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮，低电平灭**；**USART0** 串行接口引出至 9 针 **DB9 母头**。将 **GM30 光温传感模块**接到开发板的 15 针扩展接口，模块的**光照电压信号**



出至 AINO。DS18B20 温度传感器的温度数据输出至 P0_1。

串口数据监测助手全称：**嵌入式与物联网串口数据监测助手**。软件下载、功能定义、使用说明、通信规约等，可登录 www.xmf393.com 【小蜜蜂笔记网】下载查阅。

CC2530 专题置顶栏目：<https://www.xmf393.com/2019/10/20/xmf09b/>

新建工程，设计程序，实现基于串口数据监测助手的光照电压和温度数据采集并上报，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】将 DS18B20 相关的底层驱动代码文件复制到工程文件下，并作适当的移植修改。
- 【3】设计系统时钟切换函数，进系统时钟切换至 32MHz 时钟源。
- 【4】设计串口 0 初始化函数，USART0 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数。
- 【5】设计 DS18B20 温度采集函数，按照工作流程进行启动温度转换，读取温度数据，进行适当的换算，保留 1 位小数。
- 【6】设计 ADC 单次采样函数，以**查询方式**采样一次 **AIN0** 通道，获得转换结果后，取 10 位有效数据换算成电压，保留 2 位小数。当光照电压小于 1.5V 时，自动点亮 D6 灯，大于等于 1.5V 时，自动熄灭 D6 灯，并记录自动灯的开关状态。
- 【7】根据上位机“嵌入式与物联网串口数据监测助手”的通信规约，将温度数据、光照电压、灯光开关状态信息形成数据帧。
- 【8】在主函数中，每隔一小段时间进行一次光温数据采集与灯光自动控制。当收到“串口数据监测助手”的正确抄收指令后，将该数据帧上报到“串口数据监测助手”进行数据可视化处理。**D4 灯**作为通信指示灯，在正确收到上位机命令后**点亮 D4**，完成数据帧发送后**熄灭 D4**。

11.4 基于串口数据监测助手的光温综合应用-单机版

在 **XMF09B** 或 **XMF09C** 的 CC2530 开发板中，**D3 灯**接到 **P1_0** 引脚，**D4 灯**接到 **P1_1** 引脚，**D5 灯**接到 **P1_3** 引脚，**D6 灯**接到 **P1_4** 引脚，**高电平亮**，**低电平灭**；按键 **SW1** 接到 **P1_2** 引脚，按键 **SW2** 接到 **P0_1** 引脚；**USART0** 串行接口引出至 9 针 **DB9 母头**。将 **GM30 光温传感模块**接到开发板的 15 针扩展接口，模块的**光照电压信号**输出至 **AIN0**。DS18B20 温度传感器的**温度数据**输出至 **P0_1**。

串口数据监测助手全称：**嵌入式与物联网串口数据监测助手**。软件下载、功能定义、使用说明、通信规约等，可登录 www.xmf393.com 【小蜜蜂笔记网】下载查阅。

CC2530 专题置顶栏目：<https://www.xmf393.com/2019/10/20/xmf09b/>

新建工程，设计程序，循环采样 DS18B20 温度传感器的数据并通过串口上报到上位机，具体要求如下：

- 【1】设计端口初始化函数，配置 4 个 LED 灯，关闭 4 个 LED 灯。
- 【2】将 DS18B20 相关的底层驱动代码文件复制到工程文件下，并作适当的移植修改。
- 【3】设计系统时钟切换函数，进系统时钟切换至 32MHz 时钟源。
- 【4】设计串口 0 初始化函数，USART0 选择 **UART 模式**，波特率设置为 **9600**，I/O 引脚映射到**备用位置 1**，即 **P0_2** 和 **P0_3**，UART 采用默认参数。
- 【5】设计 DS18B20 温度采集函数，按照工作流程进行启动温度转换，读取温度数据，进行适当的换算，保留 1 位小数。
- 【6】设计 ADC 单次采样函数，以**查询方式**采样一次 **AIN0** 通道，获得转换结果后，取 10 位有效数据换算成电压，保留 2 位小数。当光照电压小于 1.5V 时，自动点亮 D6 灯，大于等于 1.5V 时，自动熄灭 D6 灯，并记录自动灯的开关状态。



【7】按键 SW1 定义为火警模拟触发按钮，第 1 次按下松开后，模拟有火警场景，并点亮 D3 灯，再次按下松开后，模拟无火警场景，熄灭 D3 灯，并统计按键 SW1 的触发次数。

【8】根据上位机“嵌入式与物联网串口数据监测助手”的通信规约，将温度数据、光照电压、火情状况、按键触发次数、自动灯光开关状态信息形成数据帧。

【9】在主函数中，每隔一小段时间进行光温数据采集、灯光自动控制和按键扫描统计。当收到“串口数据监测助手”的正确抄收指令后，将该数据帧上报到“串口数据监测助手”进行数据可视化处理。**D4 灯**作为通信指示灯，在正确收到上位机命令后**点亮 D4**，完成数据帧发送后**熄灭 D4**。

【温馨提示】:关于“XMF 光温显示综合应用-CC2530 单机版”、“XMF 光温显示综合应用-BsacRF 点对点版”、“XMF 光温显示综合应用-BsacRF 组网版”、“XMF 光温显示综合应用-Zigbee 组网版”综合实训项目，详见欧浩源的文档《基于串口数据监测助手的光温综合应用项目》，或登录 www.xmf393.com **【小蜜蜂笔记网】** 相关专栏查阅。

【附录 1】：“嵌入式与物联网串口数据监测助手”的功能界面及通信规约

