



独立按键的工作原理与应用

课程目标与重难点

知识目标

1

掌握

PxINP寄存器定义，
按键去抖动的原理，
按键扫描的原理。

能力目标

2

能够

设计程序，初始化信号输入端口，设计按键扫描函数正确识别按键触发信号。

素质目标

3

达到

硬件工程师系统分析、独立思考、动手实践、协同合作的基本素质。

重点： PxINP寄存器定义和按键信号扫描的原理。

难点： 扫描按键及按键去抖动的程序设计与编码实现。

项目内容与任务分解

设计程序，实现**按键输入信号控制灯光的开关**。功能如下：

- 【1】设计端口初始化函数**Init_Port()**，配置**4个LED灯**与**2个按键**的引脚。
- 【2】设计灯光测试函数**LED_Check()**，同时点亮4个LED灯，延时一会，4个LED灯熄灭（**LED灯高电平点亮**）。
- 【3】设计按键扫描函数**Scan_Keys()**，按键**SW1**按下松开后，切换**D4灯**的开关状态，按键**SW2**松开后，切换**D6灯**的开关状态。

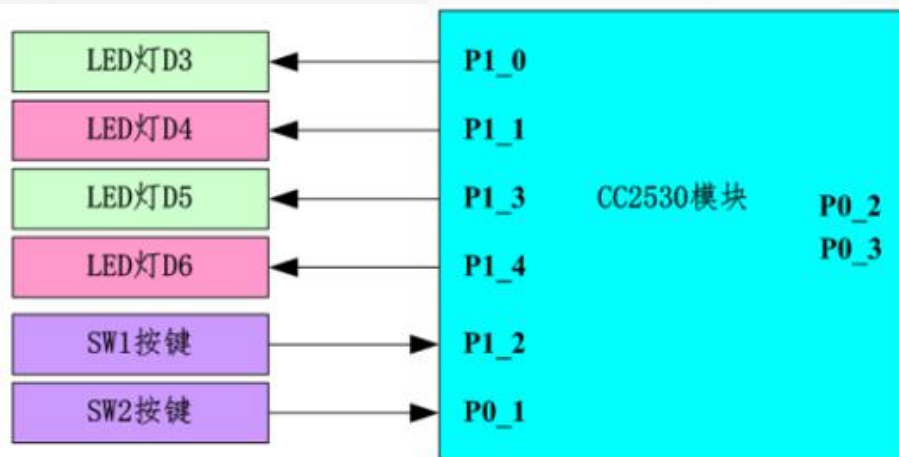
任务01：学习相关知识。

任务02：构思项目实施思路。

任务03：动手编写代码。

任务04：对比老师讲解。

任务05：修正完善项目。



任务

01

信号输入与按键扫描相关知识

CC2530的通用I/O端口的输入模式

- 当CC2530的引脚为输入端口时，该端口能够提供“**上拉**”、“**下拉**”和“**三态**”三种**输入模式**，可以通过编程进行设置。

上拉：单片机的引脚通过一个**电阻**连接到**电源（高电平）**，当没有外界信号输入到引脚时，该引脚被上拉电阻固定在高电平（**逻辑1**）。

下拉：单片机的引脚通过一个**电阻**连接到**地（低电平）**。

三态：引脚既没有上拉到电源，也没有下拉到地，呈现**高阻值状态**。

- **P1_0**和**P1_1**端口只能工作在**三态模式**。

如果微处理器的引脚没有外接设备，那么该引脚上的电压就会变得不确定，应当把这些引脚配置成上拉或下拉的**通用输入方式**，也可配置为**通用输出方式**，在实际应用中，**不能让引脚悬空**。

PxINP寄存器的定义

- PxINP寄存器：**端口输入模式寄存器**。要设置**上拉/下拉模式**：既要**对每个引脚配置**，也要对**整个端口**设配置。

POINP 寄存器和 P1INP 寄存器（端口输入模式寄存器）

位	位名称	复位值	操作	描述
7:0	MDPx[7:0]	0x00	R/W	设置 Px_7 到 Px_0 端口的输入模式。 0 : 上拉/下拉（是上拉还是下拉？需在 P2INP 寄存器中设置）。 1 : 三态。

P2INP 寄存器（端口输入模式寄存器）

位	位名称	复位值	操作	描述
7	PDUP2	0x00	R/W	为 端口 2 所有引脚选择上拉或下拉。 0 : 上拉。 1 : 下拉。
6	PDUP1	0x00	R/W	为 端口 1 所有引脚选择上拉或下拉。 0 : 上拉。 1 : 下拉。
5	PDUP0	0x00	R/W	为 端口 0 所有引脚选择上拉或下拉。 0 : 上拉。 1 : 下拉。
4:0	MDP2_[4:0]	00000	R/W	设置 P2_4 到 P2_0 端口的输入模式。 0 : 上拉/下拉。 1 : 三态。

按键扫描原理



- 按键的两个引脚，一端**通过电阻上拉到高电平**，另一端**接地**。
- 没有按键按下的时候，输入引脚为**高电平**。
- 当有按键按下的时候，输入引脚为**低电平**。
- **扫描原理**：通过**反复读取按键输入引脚**的信号，然后**识别高低电平**来判断是**否有按键触发**。

按键去抖动处理

- **思考**：是不是**只要有低电平产生**，就一定有按键按下呢？
- 按键的输入引脚有低电平产生，不代表一定是有按键按下，也许是**干扰信号**，因此，需要通过**去抖动处理**，将这些干扰信号过滤，从而**获得真实的按键触发信号**。
- **思考**：如何去抖动？

按键信号



干扰信号



- **去抖动原理**：**首次**检测到按键输入引脚有低电平后，稍作延时，**再次**读取该引脚，如还是低电平，则**确认**为按键触发信号；否则，判断为干扰信号，不予处理。

任务

02

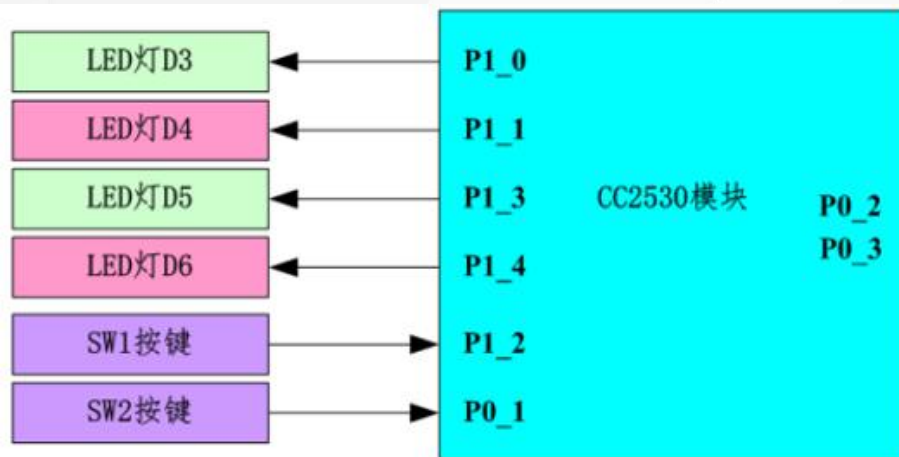
构思项目的实现思路

项目实施思路提示

设计程序，在物联网硬件实训平台上实现**按键控制灯光**。功能如下：

- 【1】设计端口初始化函数**Init_Port()**，配置**4个LED灯**与**2个按键**的引脚。
- 【2】设计灯光测试函数**LED_Check()**，同时点亮4个LED灯，延时一会，4个LED灯熄灭（**LED灯高电平点亮**）。
- 【3】设计按键扫描函数**Scan_Keys()**，按键**SW1**按下松开后，切换**D4灯**的开关状态，按键**SW2**松开后，切换**D6灯**的开关状态。

- 【1】设计延时函数。
- 【2】设计端口初始化函数。
- 【3】设计按键扫描函数。
- 【4】在主函数中反复查询按键。
- 【5】连接硬件，下载程序，运行



任务

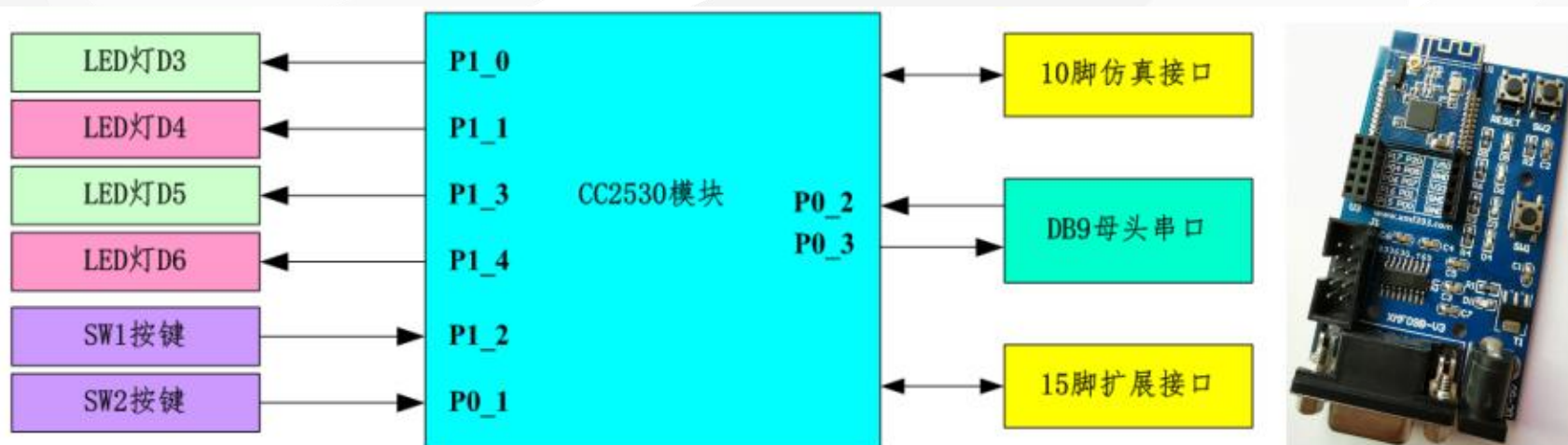
03

动手编写程序实现项目功能

项目实施思路提示

设计程序，在物联网硬件实训平台上实现**按键控制灯光**。功能如下：

- 【1】设计端口初始化函数**Init_Port()**，配置**4个LED灯**与**2个按键**的引脚。
- 【2】设计灯光测试函数**LED_Check()**，同时点亮4个LED灯，延时一会，4个LED灯熄灭（**LED灯高电平点亮**）。
- 【3】设计按键扫描函数**Scan_Keys()**，按键**SW1**按下松开后，切换**D4灯**的开关状态，按键**SW2**松开后，切换**D6灯**的开关状态。



任务

04

对比老师演示，反思实现过程

项目实施难点解析

• 端口初始化函数设计

- 【1】 选择对应端口的功能为：**通用I/O端口功能**。
- 【2】 LED端口的方向为：**输出**；按键端口的方向为：**输入**。
- 【3】 按键对应引脚的输入模式：**上拉模式**。

• 按键扫描函数设计

- 【1】 **查询按键输入**引脚的电平信号。
- 【2】 结合延时函数进行**按键去抖动**处理。
- 【3】 **等待按键松开**的代码实现。
- 【4】 **切换LED灯**的开关状态。

任务

05

修正完善项目，巩固知识技能

项目实现的参考代码

```
1 #include "ioCC2530.h"
2
3 #define D3 P1_0
4 #define D4 P1_1
5 #define D5 P1_3
6 #define D6 P1_4
7
8 #define SW1 P1_2
9 #define SW2 P0_1
10
11 void Delay(unsigned int t)
12 {
13     while(t--);
14 }
15
16 void Init_Port()
17 {
18     //配置4个LED灯的引脚
19     P1SEL &= ~0x1B; // 0001 1011
20     P1DIR |= 0x1B;
21     //配置SW1按键引脚
22     P1SEL &= ~0x04; //0000 0100
23     P1DIR &= ~0x04;
24
25     P1INP &= ~0x04;
26     P2INP &= ~0x40; //0100 0000
27
28     //配置SW2按键引脚
29     P0SEL &= ~0x02;
30     P0DIR &= ~0x02;
31     P0INP &= ~0x02;
32     P2INP &= ~0x20; //0010 0000
33 }
```

```
35 void LED_Check()
36 {
37     D3 = 1; D4 = 1; D5 = 1; D6 = 1;
38     Delay(60000);
39     Delay(60000);
40     D3 = 0; D4 = 0; D5 = 0; D6 = 0;
41     Delay(60000);
42     Delay(60000);
43 }
44
45 void Scan_Keys()
46 {
47     if(SW1 == 0)
48     {
49         Delay(200);
50         if(SW1 == 0)
51         {
52             while(SW1 == 0);
53             D4 = ~D4;
54         }
55     }
56
57     if(SW2 == 0)
58     {
59         Delay(200);
60         if(SW2 == 0)
61         {
62             while(SW2 == 0);
63             D6 = ~D6;
64         }
65     }
66 }
```


谢谢大家

