

## ■ 实验课

- 欢迎大家在实验中提出问题，老师会根据问题适当组织大家进行讨论，帮助大家从不懂到懂，从不会到会。
- 希望大家充分利用实验课，提高自己动手实践能力，特别是调试能力。

成绩评定：平时实验70%； 实验项目30%；

考核方式：P/F

## 单片机基础实验课程目标

通过一周理论学习和动手实验

- 对单片机的工作原理有基本了解
- 熟悉单片机的应用开发过程，掌握基本调试手段
- 具备单片机应用系统的软硬件基本开发能力
- 为大二之后的科技、科研活动打下基础。

完成一个基本实验项目来达到

## 钢琴、电子钢琴、电子琴

音名 C D E F G A B c d e f g a b c<sup>1</sup> d<sup>1</sup> e<sup>1</sup> f<sup>1</sup> g<sup>1</sup> a<sup>1</sup> b<sup>1</sup> c<sup>2</sup> d<sup>2</sup> e<sup>2</sup> f<sup>2</sup> g<sup>2</sup> a<sup>2</sup> b<sup>2</sup> c<sup>3</sup>

五线谱

简谱 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1

唱名 do re mi fa sol la si do re mi fa sol la si do re mi fa sol la si do re mi fa sol la si do

## 自制简易电子琴

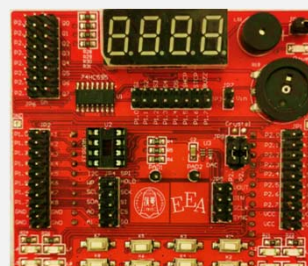
钢琴、电子钢琴、电子琴

音名 C D E F G A B c d e f g a b c<sup>1</sup> d<sup>1</sup> e<sup>1</sup> f<sup>1</sup> g<sup>1</sup> a<sup>1</sup> b<sup>1</sup> c<sup>2</sup> d<sup>2</sup> e<sup>2</sup> f<sup>2</sup> g<sup>2</sup> a<sup>2</sup> b<sup>2</sup> c<sup>3</sup>

五线谱

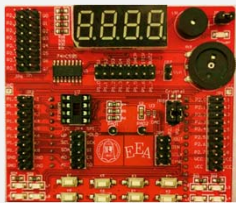
简谱 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1

唱名 do re mi fa sol la si do re mi fa sol la si do re mi fa sol la si do re mi fa sol la si do



音名	c	d	e	f	g	a	b	c <sup>1</sup>	d <sup>1</sup>	e <sup>1</sup>	f <sup>1</sup>	g <sup>1</sup>	a <sup>1</sup>	b <sup>1</sup>	c <sup>2</sup>	d <sup>2</sup>	e <sup>2</sup>	f <sup>2</sup>	g <sup>2</sup>	a <sup>2</sup>	b <sup>2</sup>
简谱	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
琴键	K1	k2	k3	k4	k5	k6	k7	k1	k2	k3	k4	k5	k6	k7	k1	k2	k3	k4	k5	k6	k7
	K+														K8+						

音乐播放

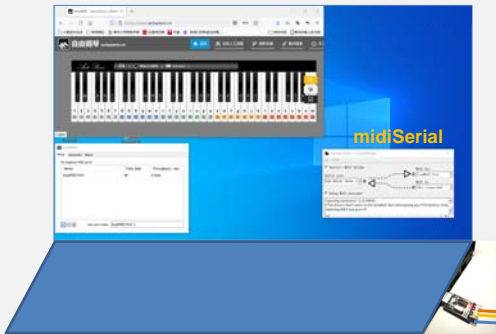


MIDI键盘控制电脑音响发声

电脑上发出钢琴声



MIDI键盘



计算机

USB转串口



单片机

## 实验项目 例：简易电子琴和音乐播放器

### 基本功能

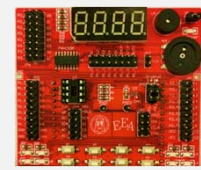
- ✓ 电子琴弹奏
- ✓ 音乐播放
- ✓ 现场和远程进行播放和弹奏模式及播放曲目的切换功能
- ✓ 用LED或数码管进行状态显示



MSP430G2553

相对简单，好理解

硬件资源少，性能相对弱



### 提高功能

- ✓ 快慢速播放控制
- ✓ 节拍器
- ✓ 计时功能
- ✓ 闹铃功能
- ✓ 弹奏录制
- ✓ .....

- 如何控制 LED 显示？
- 如何识别按键操作？
- 如何及时响应按键？
- 如何确定时间长短？
- 如何发出不同音调？
- 如何与计算机通信？
- 如何与手机蓝牙通信？

实验9 STM32单片机开发入门

STM32G431



实验1~3 基本输入输出

实验4 中断技术

实验5 基本时钟模块和定时功能

实验6 定时器PWM输出

实验7-8 串行通信

实验1 实验板结构和msp430单片机开发工具使用

实验2 单片机C语言程序结构

实验3 外部设备的控制

实验4 中断技术

实验5 基本时钟模块和定时功能

实验6 定时器PWM输出及应用

实验7 PC机的异步串行通信

实验8 单片机的异步串行通信

实验9 STM32 单片机开发入门

实验10 简易电子琴的设计与实现

## ■ 课程用到实验设备

爱护实验设备，请勿丢失或损坏实验设备。



1. msp430单片机实验板一套；
2. stm32单片机实验板一套；
3. 小车车模一个；
4. 配套模块一袋：  
(USB转串口模块2个；  
蓝牙模块  
避障模块  
无源蜂鸣器；  
按键一个)

## ■ MSP430G2553实验系统简介

储物小盒一个；  
MSP430G2553单片机板一块、  
MSP430G2553扩展板一块；  
mini USB 数据线一根；  
实验用导线若干。



Mini USB数据线

用于连接实验板和PC机



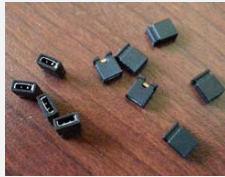
G2扩展板

G2单片机板

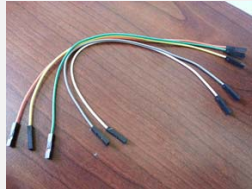


第9次实验时  
STM32单片机实验板

## 实验用导线



**跳线块**  
用于短接  
两个相邻的插针；



**双口杜邦线**  
用于连接实验板上  
不相邻的插针

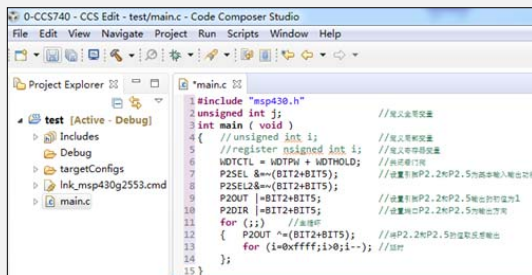


**带插针的单口杜邦线**  
用于将实验板信号引出，  
与示波器探头连接，观察信号

## 实验1 实验板检测和开发工具使用

### 一、实验目的

1. 了解MSP430G2实验板结构，掌握实验板基本检测方法
2. 掌握集成开发环境CCS的基本使用方法
3. 掌握基本输入/输出引脚与相关I/O寄存器的关系(第2章内容)



本次实验主要熟悉CCS的基本操作，不用自己编程

**实验项目例：简易电子琴和音乐播放器**

**基本功能**

- ✓ 电子琴弹奏
- ✓ 音乐播放
- ✓ 现场和远程进行播放和弹奏模式及播放曲目的切换功能
- ✓ 用LED或数码管进行状态显示

**提高功能**

- ✓ 快速播放控制
- ✓ 节拍器
- ✓ 计时功能
- ✓ 闹钟功能
- ✓ 弹奏录制
- ✓ .....

**实验1~3 基本输入输出**

- 如何控制LED显示？
- 如何识别按键操作？
- 如何及时响应按键？
- 如何确定时间长短？
- 如何发出不同音调？
- 如何与计算机通信？
- 如何与手机蓝牙通信？

**实验4 中断技术**

**实验5 基本时钟模块和定时功能**

**实验6 定时器PWM输出**

**实验7-8 串行通信**

**实验9 STM32单片机开发入门**

**STM32G431**

## 二、实验任务

### 1. 了解实验板结构和掌握扩展板检测方法

(检测单片机外部电路，检测的电路部分与单片机无关，今后做实验出现问题时，比如LED状态不变，按键没有反应，蜂鸣器不响等用该方法可以判断是否在单片机外部硬件电路出了问题)

### 2. CCS开发工具的初步使用

掌握创建工程，编辑、编译、下载、运行程序，查看和修改寄存器、变量、存储器单元的方法

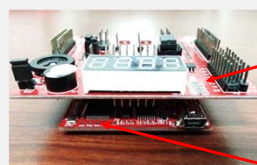
### 3. 理解基本输入/输出引脚与相关I/O寄存器的关系

了解单片机内部端口P1~P2的选择PxSEL、方向PxDIR、输出PxOUT、上/下拉电阻使能PxREN、输入PxIN等各寄存器作用，理解它们对单片机引脚的控制功能。

**注意：**任务1和3的检测方法今后实验中反复用到，需要大家熟练掌握。

## 任务1. 了解实验板结构和掌握扩展板检测方法

### 1) 了解实验板结构



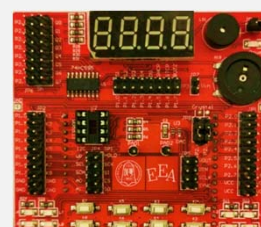
扩展板

单片机板

### 2) 检测扩展板

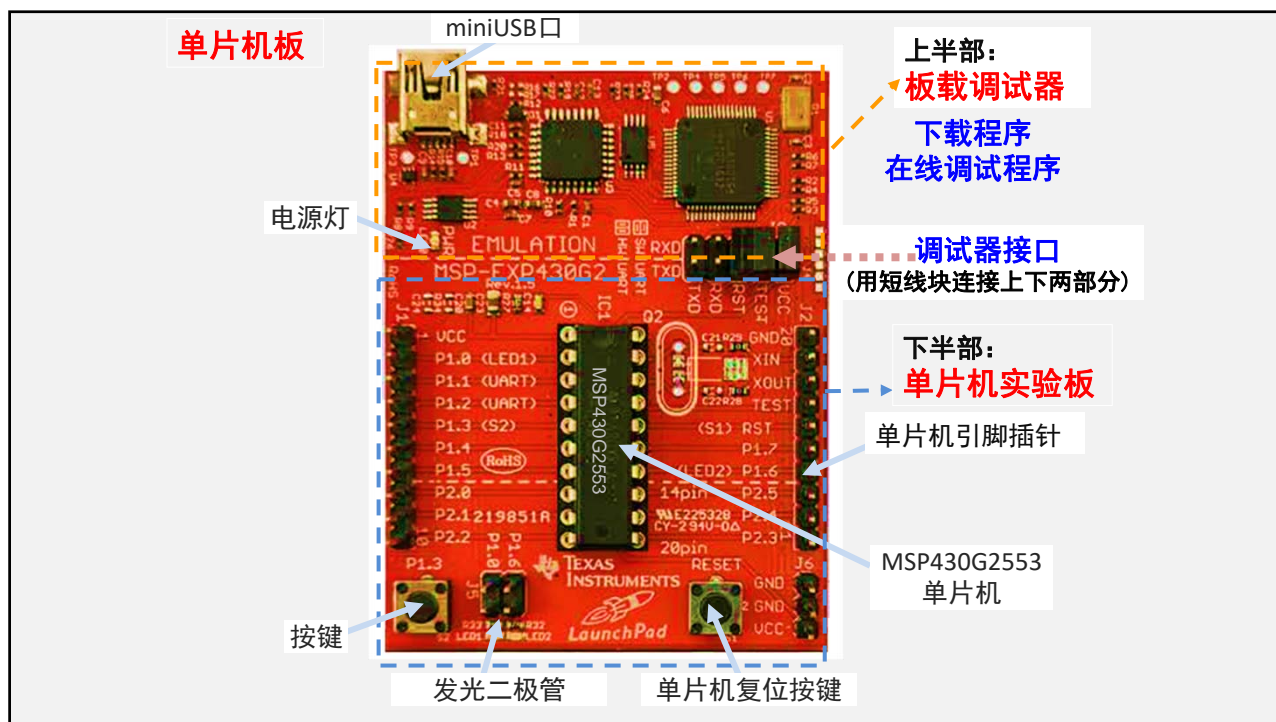


单片机板

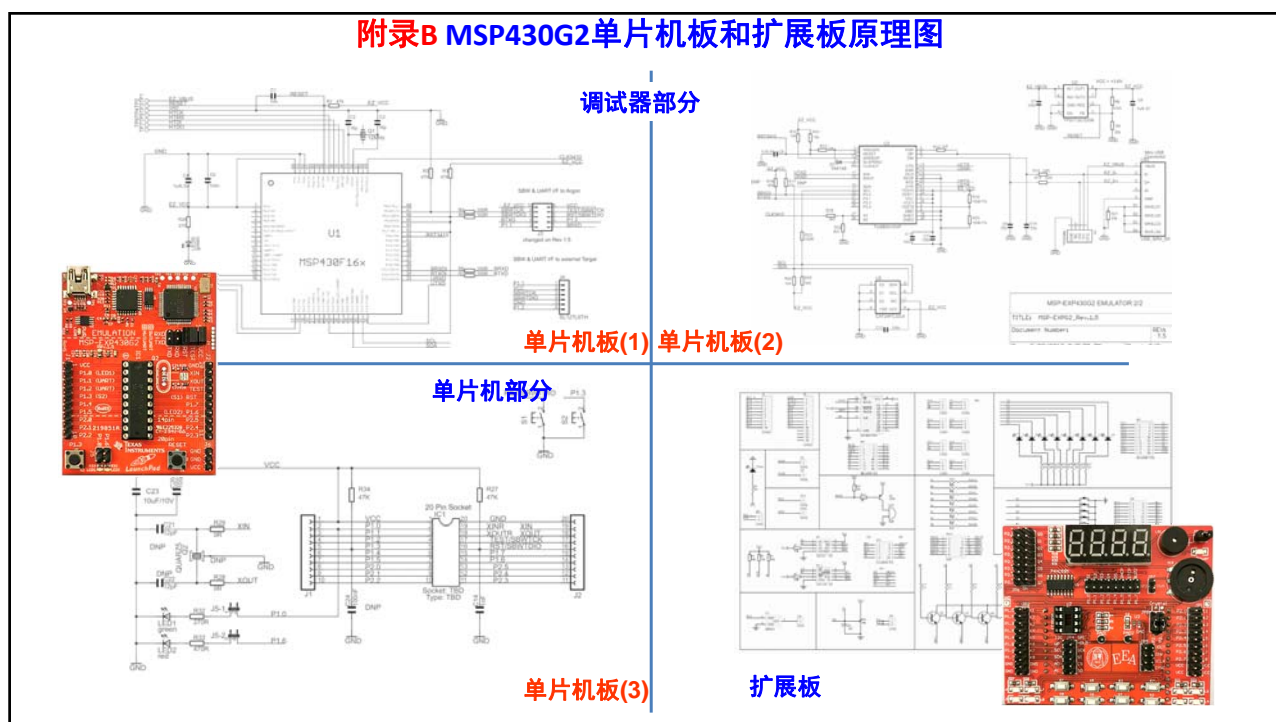
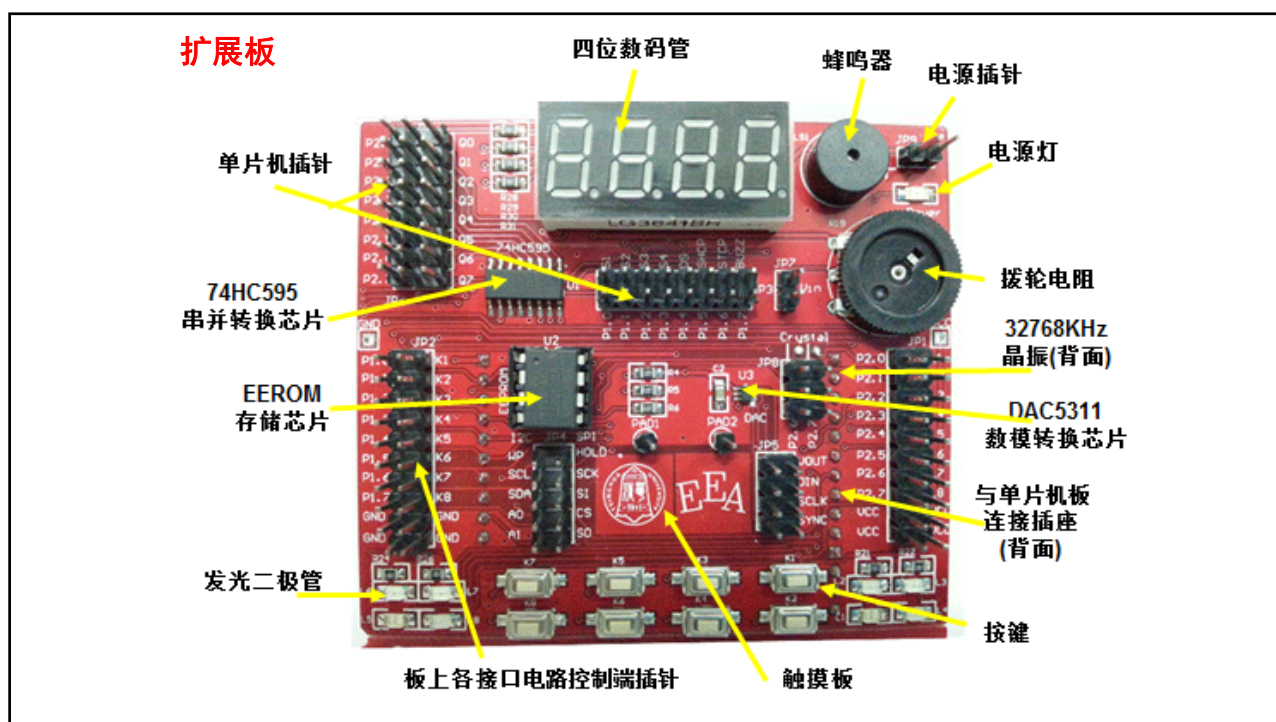


扩展板

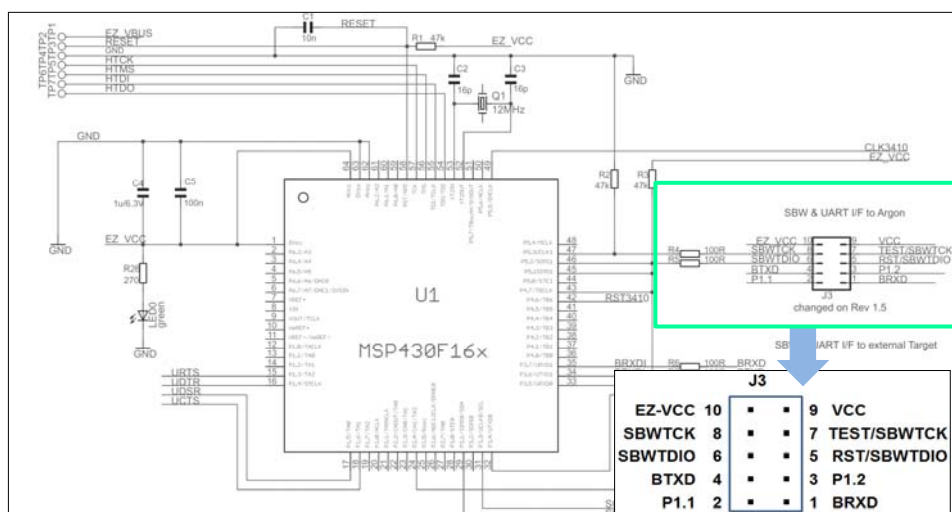
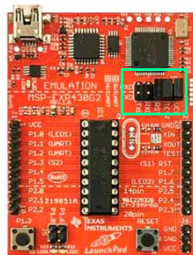




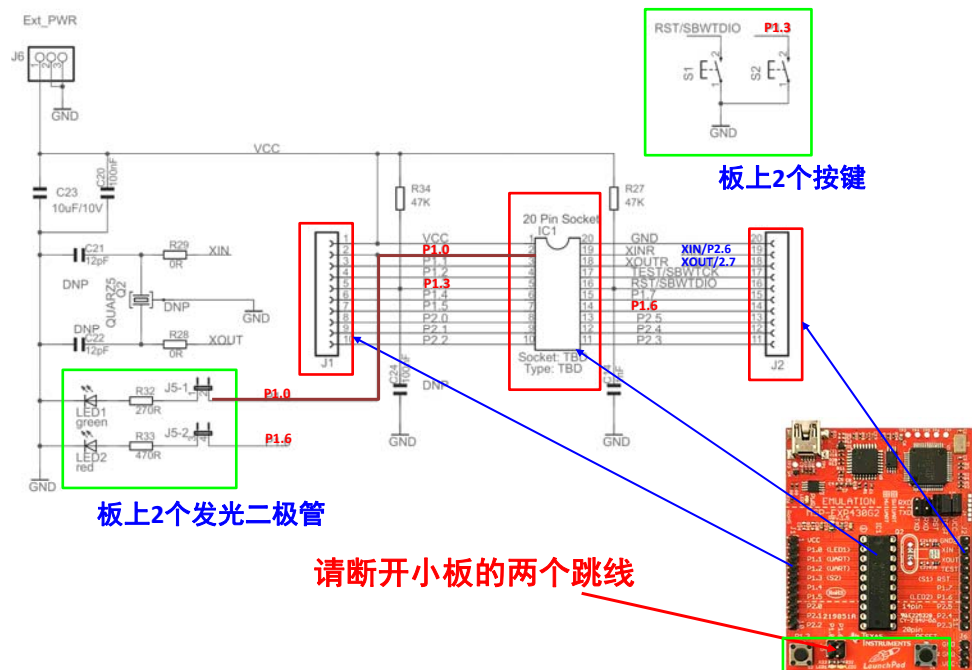




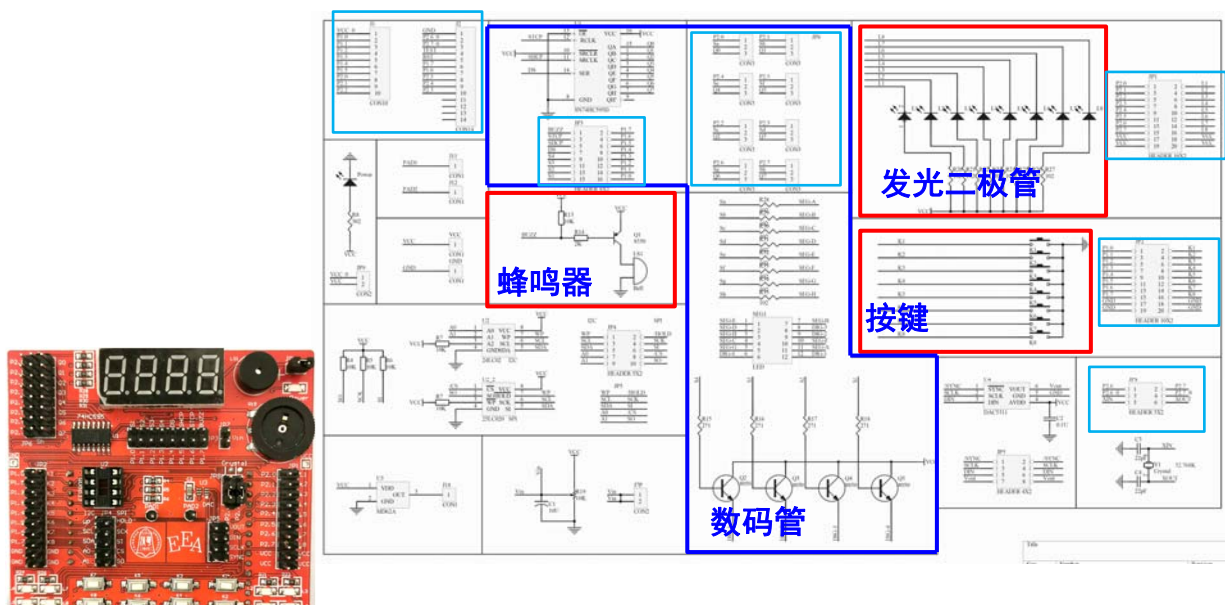
## 参考附录B MSP430G2单片机板原理图第1页



参看附录B MSP430G2单片机板原理图第3页



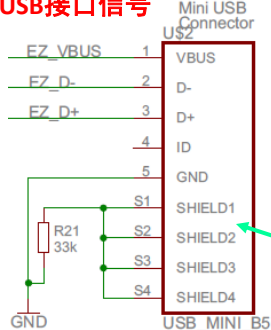
参看附录B 扩展板原理图第4页



## 电源部分

参看附录B MSP430G2单片机板原理图第2页

### USB接口信号

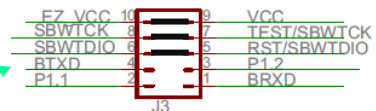


由USB数据线接入5V电压

EZ\_VBUS USB接口电压 5V

EZ\_VCC 调试器工作电压 3.6V

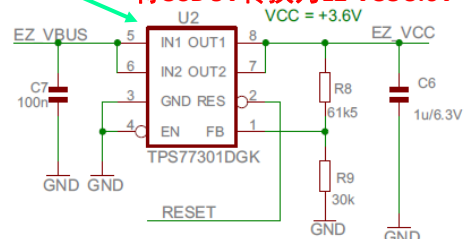
VCC G2实验板工作电压 3.6V



调试器接口信号

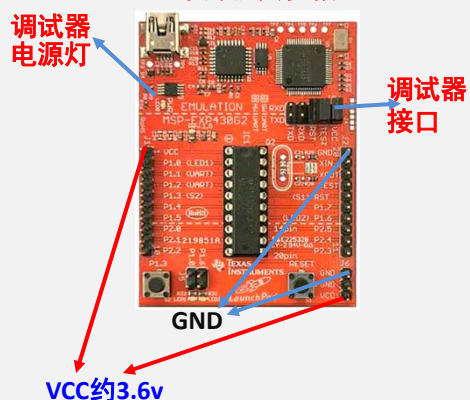
### 电平转换芯片

将USB 5V转换为EZ-VCC 3.6V

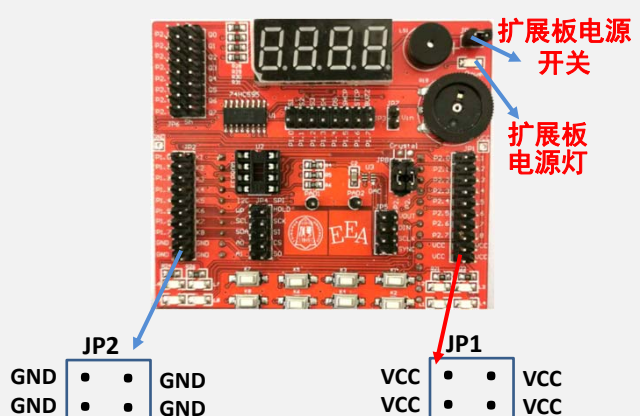


## 实验板电源相关引脚信号

### 单片机实验板



### 扩展板



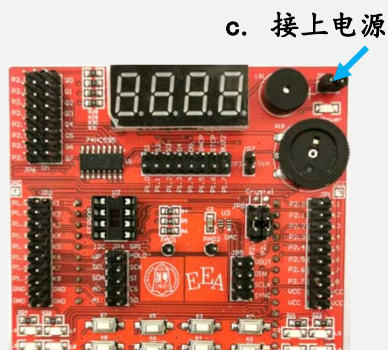
## 2. 检测实验板

### 1) 检查板上有关跳线的连接情况



a. 接上  
调试器接口  
三个跳线

b. 断开两个LED  
与单片机的连接



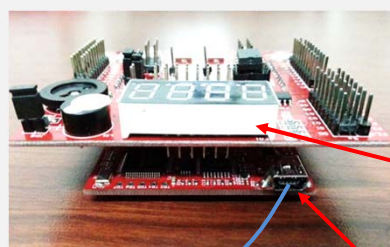
c. 接上电源

### 2) 将扩展板(大板)插在单片机板(小板)上

注意插接的方向:

扩展板的数码管和单片板的USB接线口在一侧

### 3) 用USB数据线将实验板与PC机连上



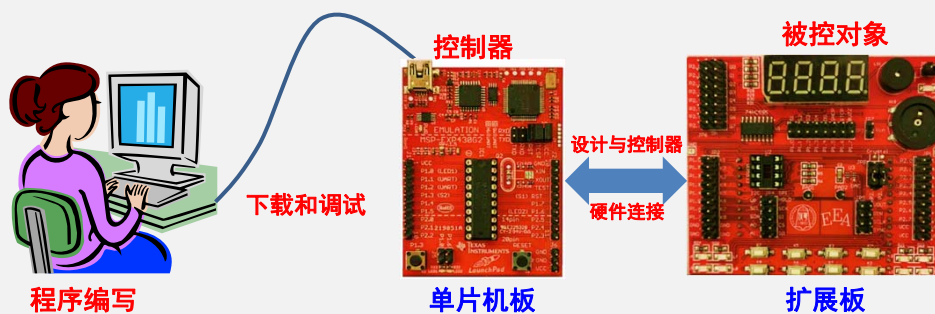
数码管

USB线

USB接口

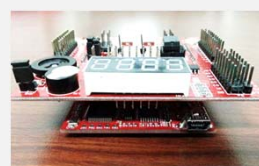


## 单片机应用开发(实验)



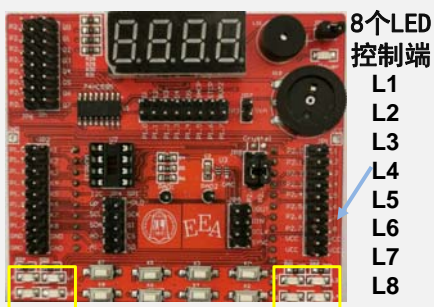
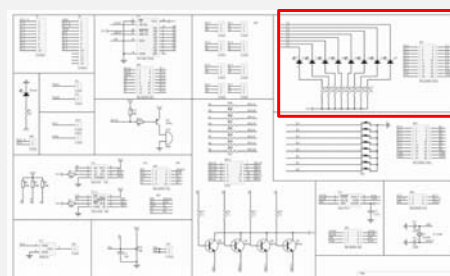
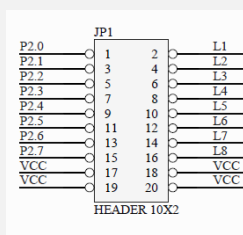
当运行程序现象与期待的不符时，  
问题在哪里？如何分析、查找和解决？

- 单独检查扩展电路硬件 → 实验1任务1
- 检查单片机+扩展电路硬件 → 实验1任务3
- 检查程序 → 实验2任务1

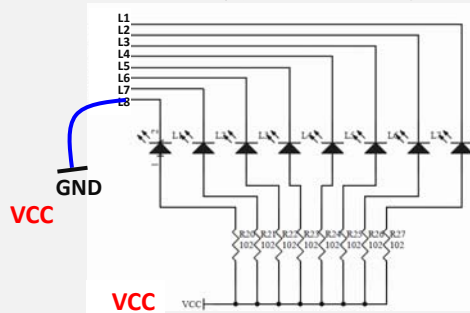


做实验时

## 发光二极管电路原理



## 8个共阳极发光二极管原理图

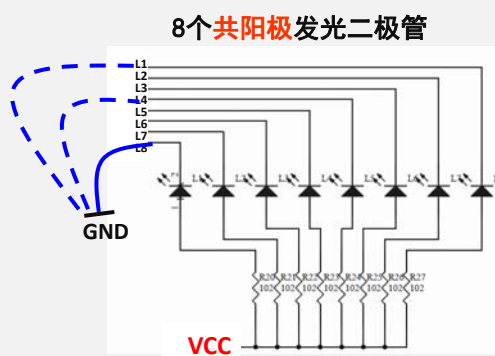
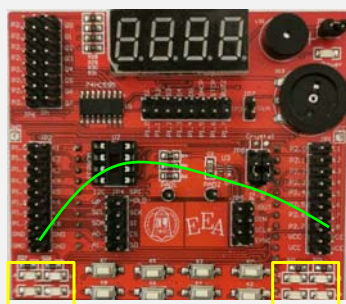




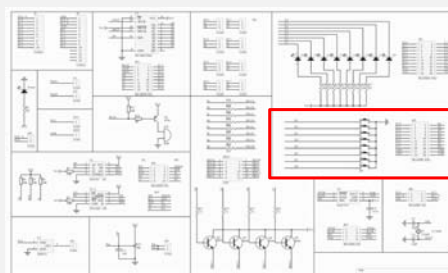
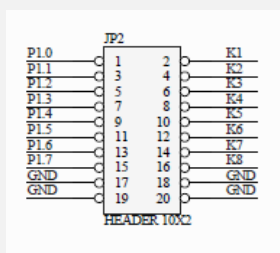
## 检测发光二极管电路

1. 卸下L1~L8上短线块
2. 用一根孔孔杜邦线一端固定接在GND上，另一端分别与接口JP1上的L8~L1点接，观察LED是否发亮
3. 注意不要让GND与VCC短接，造成短路

建议从L8开始，L8、L7、L6 ..... L1

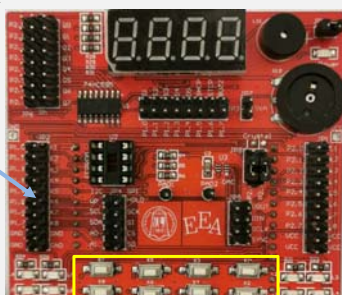


## 按键电路原理

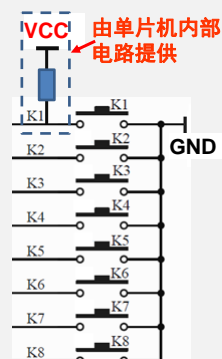


8个按键  
控制端

K1  
K2  
K3  
K4  
K5  
K6  
K7  
K8

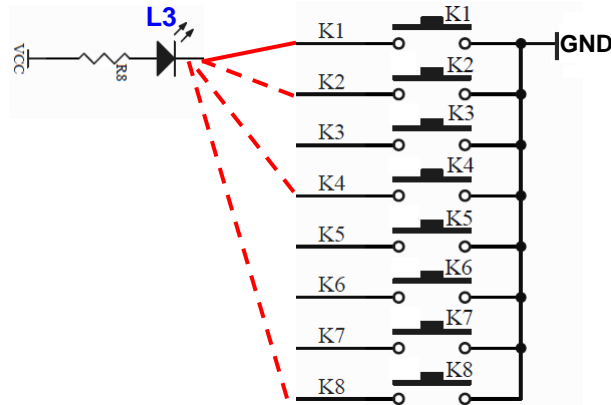
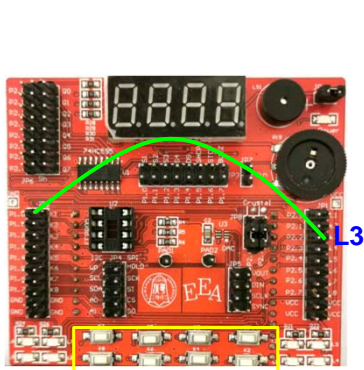


注意：  
实验板上8个按键，  
外部电路  
无上拉电阻  
需要想办法  
加一上拉电阻  
才能从Kx处的电平高、低  
判断按键的释放、按下动作

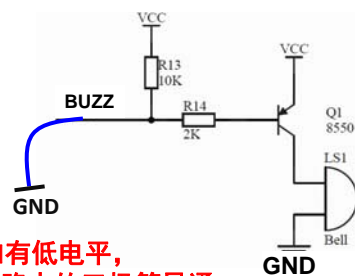
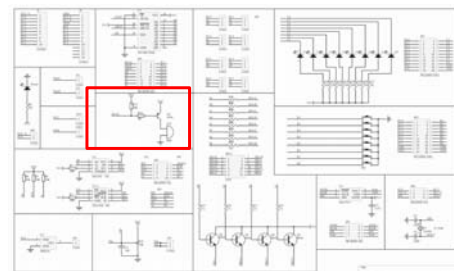
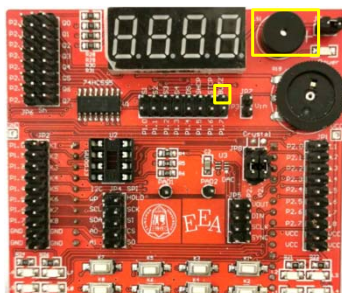
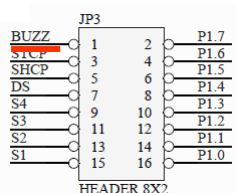


## 按键电路检测

1. 卸除K1~K8上短线块；
2. 用杜邦线一端固定接在一个发光二极管上，如L3，另一端与插针K1连接；
3. 操作K1按键，观察L3的亮灭。如果按下K1按键，L3灯亮，说明按键K1良好；
4. 同理，将杜邦线从K1换到K2上，操作K2按键，检测按键K2；  
依次按此方法检测其他K3~K8按键。（L3端接线不动）



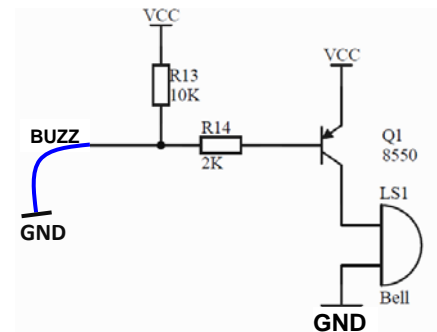
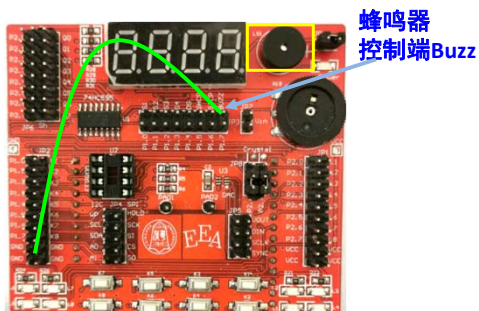
## 蜂鸣器电路



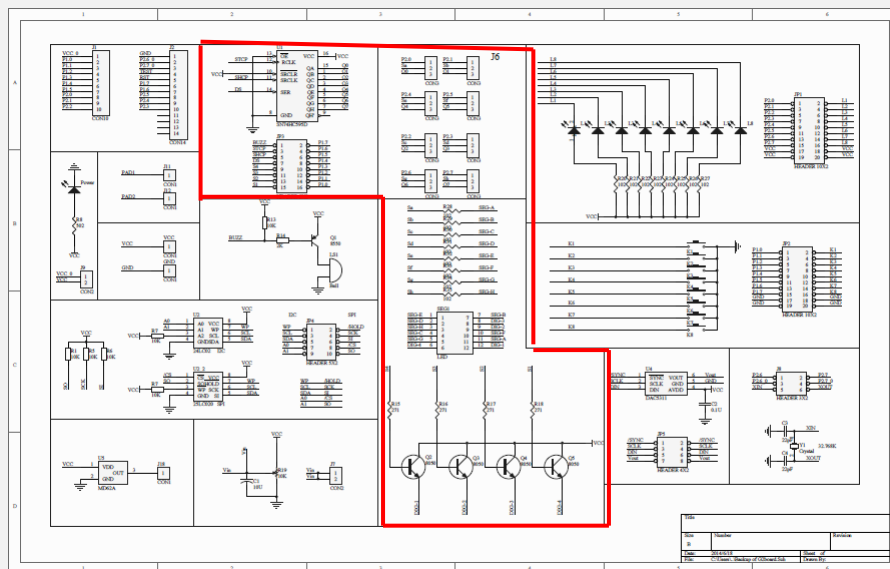
BUZZ处如有低电平，  
蜂鸣器线路中的三极管导通  
蜂鸣器响。

## 蜂鸣器电路检测

1. 卸除插针BUZZ上短线块；
2. 用杜邦导线一端连接在GND上，  
另一端与JP3接口上的BUZZ点连，  
通过是否发出蜂鸣声来判断蜂鸣器正常与否。

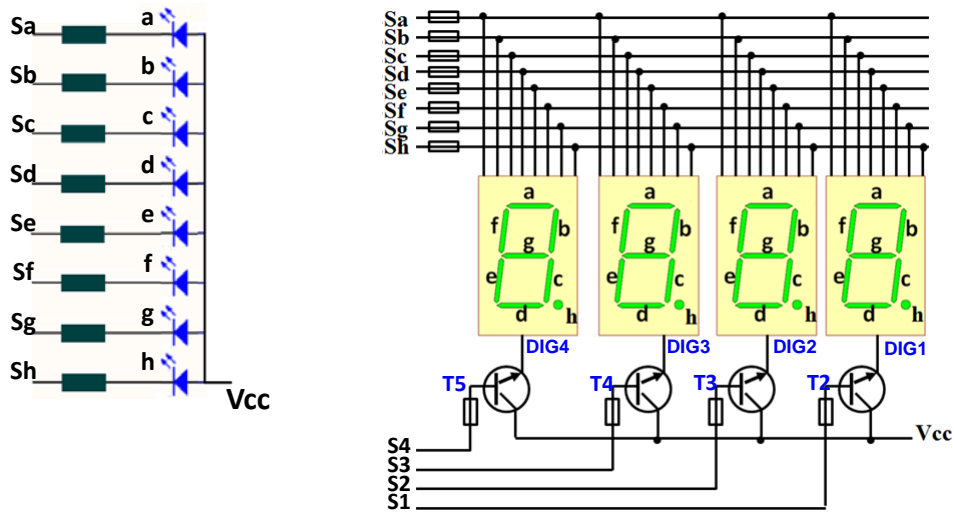


## 数码管控制电路分析



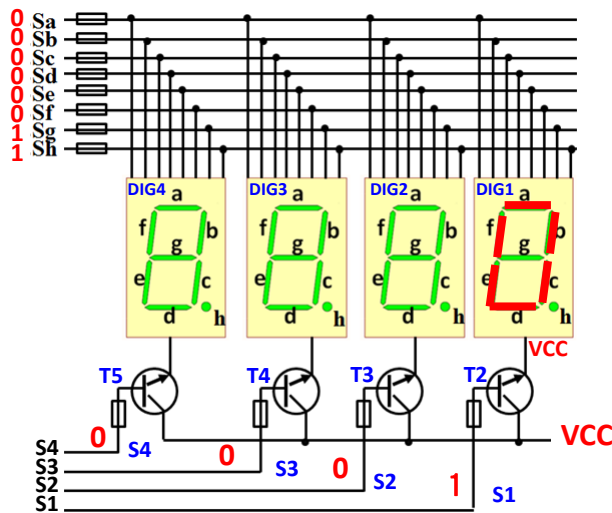
实验板原理图中数码管控制电路相关部分

多个数码管的工作原理图 图2-2



4个共阳极数码管，各自的公共端分别为S4~S1

### 数码管电路工作原理分析



■ 板上数码管为共阳极；

■ 当显示段Sh~Sa上输出：

Sh Sg Sf Se Sd Sc Sb Sa  
1 1 0 0 0 0 0

如果 S1置1，其他S2~S4置0，

T2三极管导通，

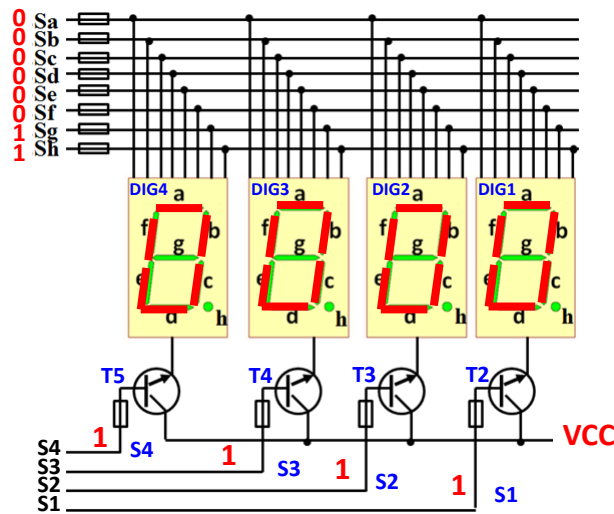
DIG1数码管公共端与Vcc接通，  
显示由Sh~Sa上电平决定。

T3~T5三极管不通，

对应的数码管DIG2~DIG4不显示

用三极管增加驱动能力

## 数码管电路工作原理分析



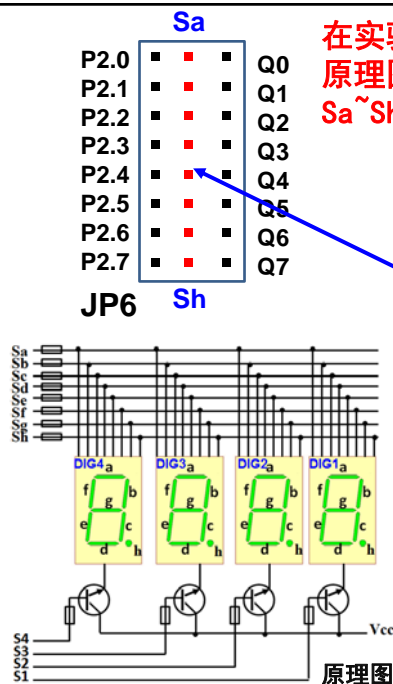
用三极管增加驱动能力

如果 S1~S4均置1，

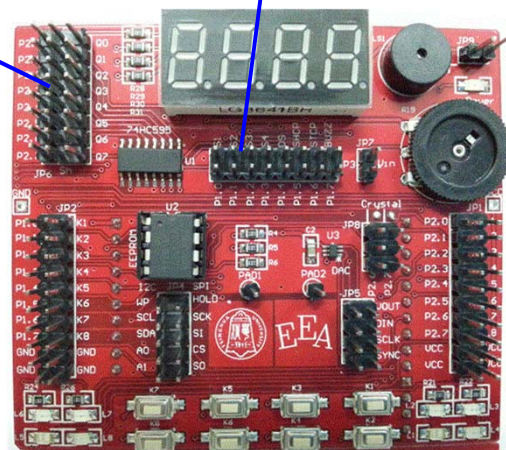
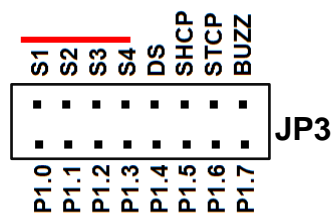
T2~T5四个三极管均导通，  
DIG1~DIG4四个数码管的  
公共端均与Vcc接通，  
显示相同的内容。

故：

- 显示的内容由Sh~Sa上的电平决定；
- 显示在哪个数码管上由S4~S1上的电平决定。

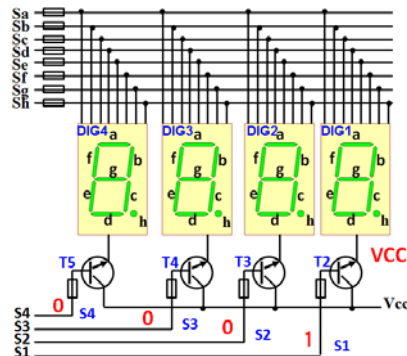
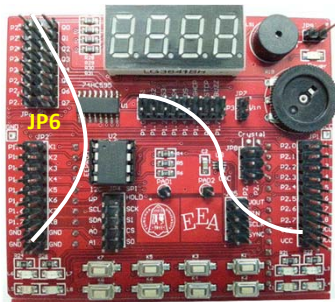


在实验板上找到  
原理图中数码管的控制端  
Sa~Sh、S1~S4

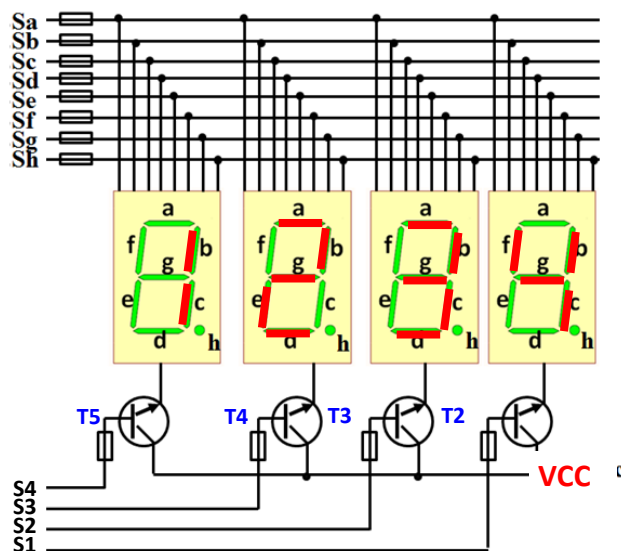


## ■ 数码管电路检测

1. 卸除JP6上Sh~Sa上的跳线块；
2. 用一根杜邦线，一端接电源Vcc，一端接S1；
3. 用另一根杜邦线，一端固定接地GND, 另一端分别顺序轻点Sa~Sh，观察S1对应的数码管各段是否发亮；
4. 将杜邦线依次改接S2~S4，检查对应的数码管



思考：如何让四个数码管显示不同的信息呢？

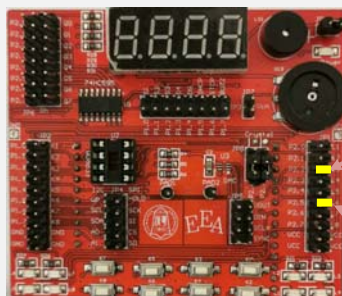




## 任务2 CCS开发工具的初步使用

### 1) 确认接好相关接线

- 确认单片机引脚P2.5和P2.2分别与发光二极管L6、L3连接
- 用USB数据线将实验系统与PC机相连



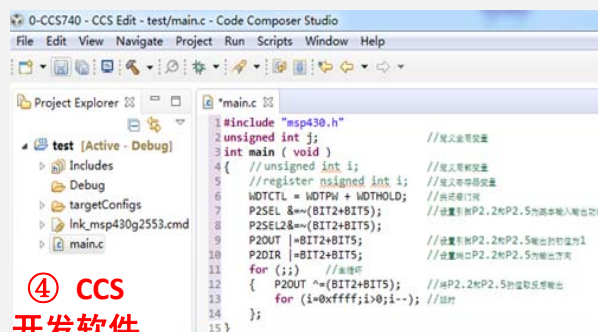
P2.2 与 L3连接

P2.5 与 L6连接

### ■ 课前准备

已在自己的电脑上安装好CCS，  
并按附录A的步骤一至六，测试了自己的实验平台已搭建好。

### 实验的硬件系统和软件系统



④ CCS  
开发软件  
界面

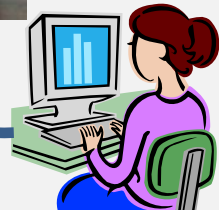
计算机侧利用ccs软件  
在实验板上调试器的支持下  
实现对单片机的编程和调试

① 实验板



② USB线

③ 计算机



## 2) 启动CCS进行基本使用 (搭建实验平台时已完成)

参看“附录A CCS的使用入门”中的一至六

在工程空间下建立项目test，了解C语言程序项目建立、编辑、编译、连接、程序下载和运行方法。

注意附录A源程序test\_g2553.C提供有电子版。

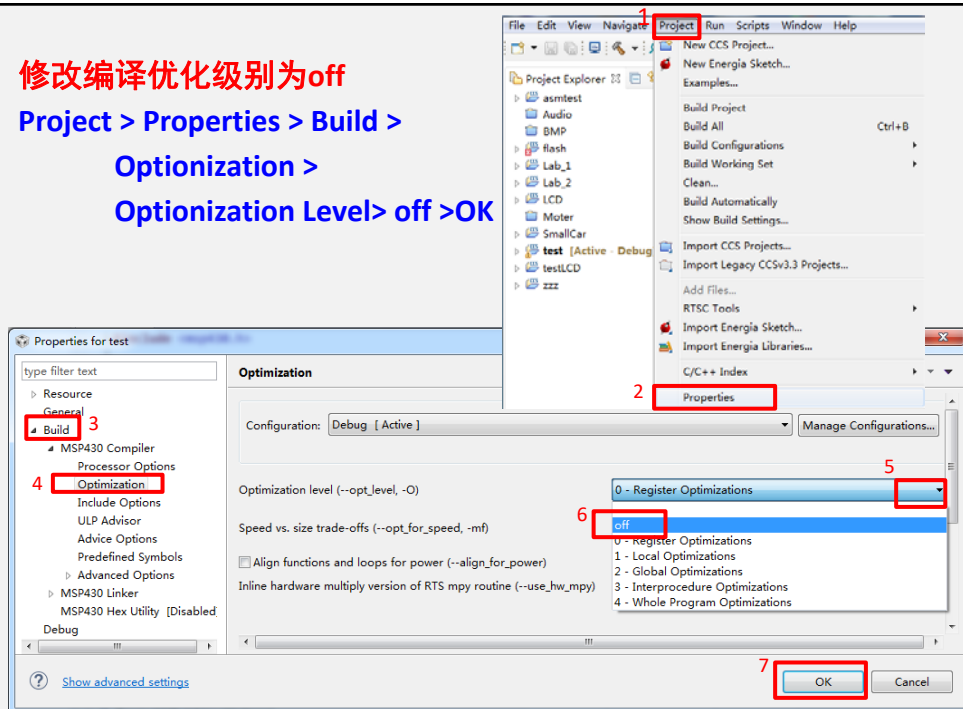
工程空间名称、项目名称和C的源程序避免采用中文命名。

按照下面几个步骤了解CCS的基本使用：

- 一、创建 C 语言项目
- 二、编辑和添加源程序
- 三、编译和连接程序 **注意：改编译优化级别为off**
- 四、连接相关硬件
- 五、下载程序到目标 MCU 中
- 六、运行程序及相关命令
- 七、查看当前单片机状态

修改编译优化级别为off

Project > Properties > Build >  
Optionization >  
Optionization Level> off >OK



## CCS开发工具的初步使用(续)

### 3) 在test项目中, 练习修改程序, 重新编译、下载、运行

- (1) 将 test\_g2553.C语句 “unsigned int i;” 中的int改为long, 重新编译、连接和下载, 观看执行的现象有什么不同, 主要是观察LED闪烁的速度有没有变化。
- (2) 在 test\_g2553.C中的语句 “unsigned int i;” 前加一个关键字register, 将变量i指定用CPU内部的寄存器, 并将其放在main()函数中, 重新编译、连接和下载, 观看执行的现象有什么不同。
- (3) 将(2)中的register unsigned int i, 改为register unsigned long, 重新编译、连接和下载, 看到的现象应该也会不同的。

test\_g2553.C

```
//可根据练习的需要, 灵活取消或保持各种方式变量i定义前的注释符,
//以便观察不容类型变量对程序执行的影响
#include "msp430.h"

unsigned int j;           //定义全局int型变量
//unsigned long i;        //定义全局long型变量

int main ( void )
{
    //unsigned int i;       //定义局部int型变量
    //unsigned long i;      //定义局部long型变量
    //register unsigned int i; //定义寄存器int型局部变量
    //register unsigned long i; //定义寄存器long型局部变量
    WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
    P2SEL &=~(BIT2+BIT5); //设置引脚P2.2和P2.5为基本输入输出功能
    P2SEL2 &=~(BIT2+BIT5);
    P2OUT |=BIT2+BIT5; //设置引脚P2.2和P2.5输出的初值为1
    P2DIR |=BIT2+BIT5; //设置端口P2.2和P2.5为输出方向
    for (;;) //主循环
    {
        P2OUT ^= (BIT2+BIT5); //将P2.2和P2.5的值取反后输出
        for (i=0xFFFF; i>0; i--); //延时
    }
}
```

## CCS开发工具的初步使用(续)

### 4) 练习查看和修改CPU寄存器、I/O寄存器、变量



运行 暂停 结束

View/Register

View/Expressions

#### CPU寄存器

Name	Value
Core Registers	
PC	0xC000
SP	0x03FE
SR	0x0000
R3	0x0000
R4	0x2FEF
R5	0xFF7C
R6	0xB6FB
R7	0xCDEF
R8	0xFF7F

#### I/O寄存器

Name	Value
Port_1_2	
P1IN	0x87
P1OUT	0xFF
P1DIR	0x00
P1IFG	0x71
P1IES	0xFF
P1IE	0x00
P1SEL	0x00
P1SEL2	0x00

#### 变量

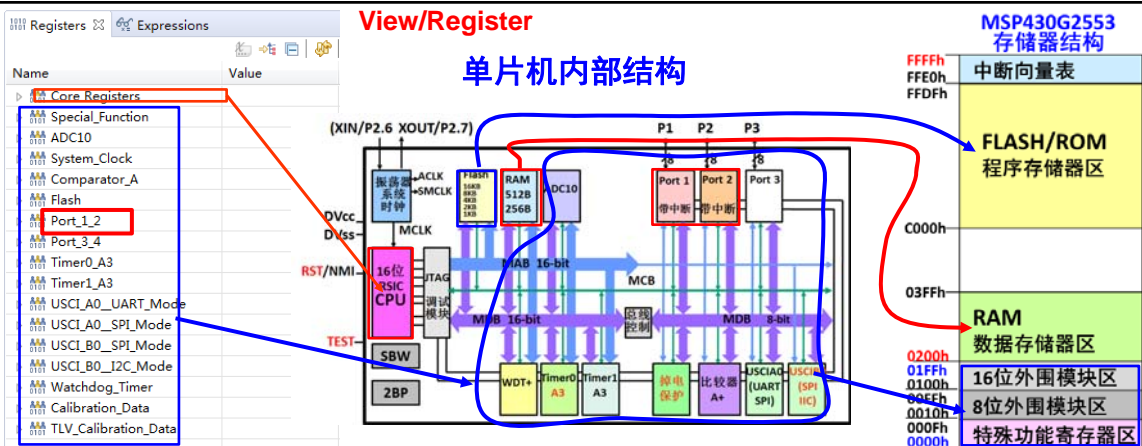
### 5) 退出CCS

Expression	Type	Value	Address
(x)= i	unsigned int	40911	0x03FC
+ Add new expression			

## ■ 问题讨论：

如何理解局部变量*i* 定义的不同，  
程序执行速度的变化？

```
//可根据练习的需要，灵活取消或保持各种方式变量i定义前的注释符，
//以便观察不容类型变量对程序执行的影响
#include "msp430.h"
unsigned int i;           //定义全局int型变量
//unsigned long i;       //定义全局long型变量
int main ( void )
{ //unsigned int i;       //定义局部int型变量
  //unsigned long i;     //定义局部long型变量
  //register unsigned int i; //定义寄存器int型局部变量
  //register unsigned long i; //定义寄存器long型局部变量
  WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
  P2SEL &=~(BIT2+BIT5);    //设置引脚P2.2和P2.5为基本输入输出功能
  P2SEL2 &=~(BIT2+BIT5);
  P2OUT |=BIT2+BIT5;       //设置引脚P2.2和P2.5输出的初值为1
  P2DIR |=BIT2+BIT5;       //设置端口P2.2和P2.5为输出方向
  for (;) //主循环
  { P2OUT ^=(BIT2+BIT5);    //将P2.2和P2.5的值取反后输出
    for (i=0xFFFF; i>0; i--); //延时
  }
}
```



## View/Expressions unsigned int *i*; // 函数体内定义局部变量

Expression	Type	Value	Address
(*) <i>i</i>	unsigned int	53104	0x0200
Add new expression			

unsigned int *i*; //函数体外定义全局变量

unsigned long *i*;

//改变变量类型

Expression	Type	Value	Address
(*) <i>i</i>	unsigned int	64509	Register R15
Add new expression			

register unsigned int *i*;

for (*i*=0xFFFF; *i*>0; *i*--);

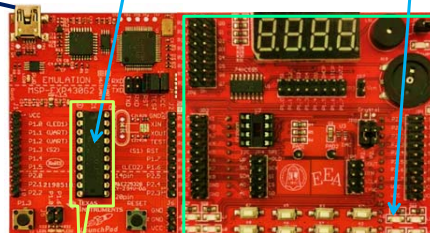
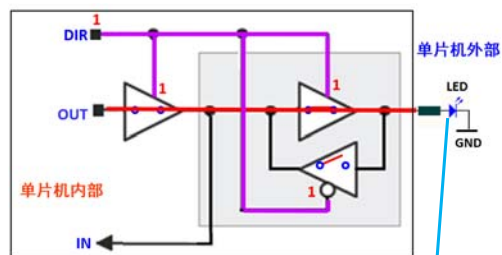
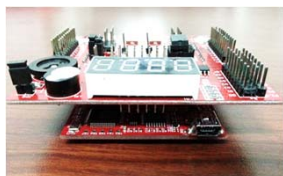
## 结合实物

今天实验任务

理解和操作这样的过程



借助计算机上的CCS和单片机的连接，  
修改单片机DIR、OUT这些位的值



单片机

单片机外部电路/设备

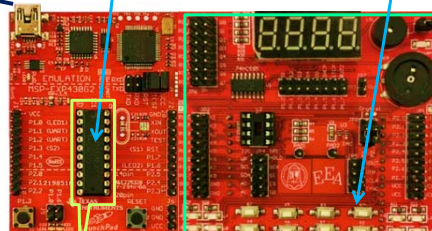
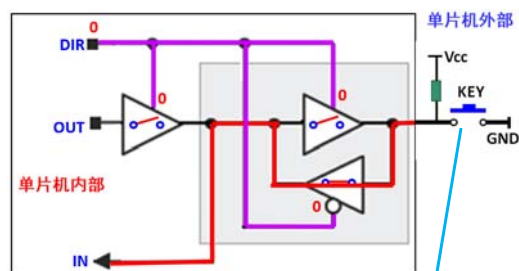
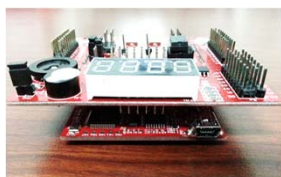
## 结合实物

今天实验任务

理解和操作这样的过程



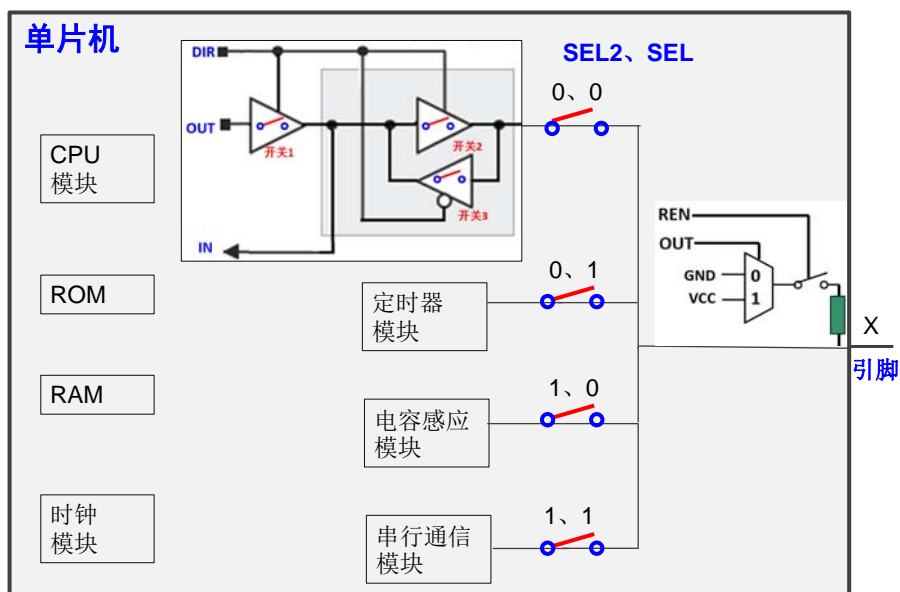
借助计算机上的CCS和单片机的连接，  
修改单片机DIR、OUT这些位的值



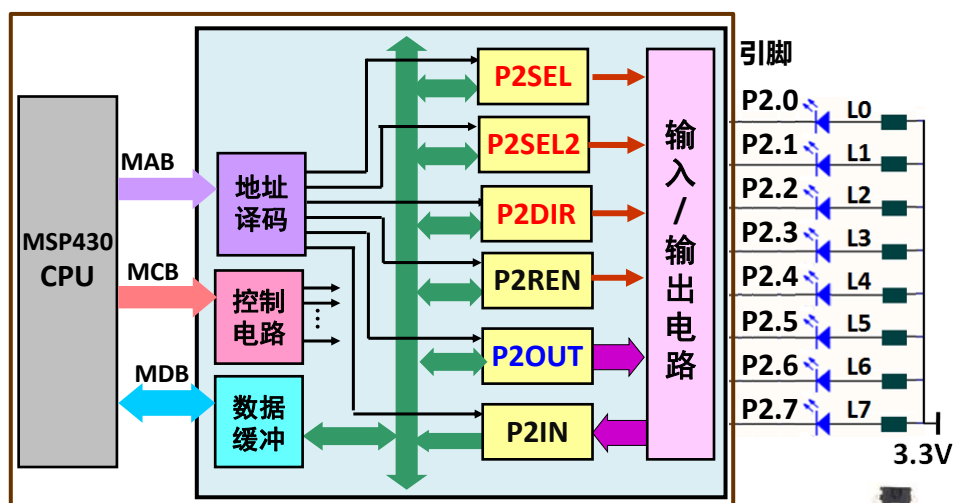
单片机

单片机外部电路/设备

## 理解PxSEL、PxSEL2、PxDIR、PxOUT、PxIN和PxREN作用



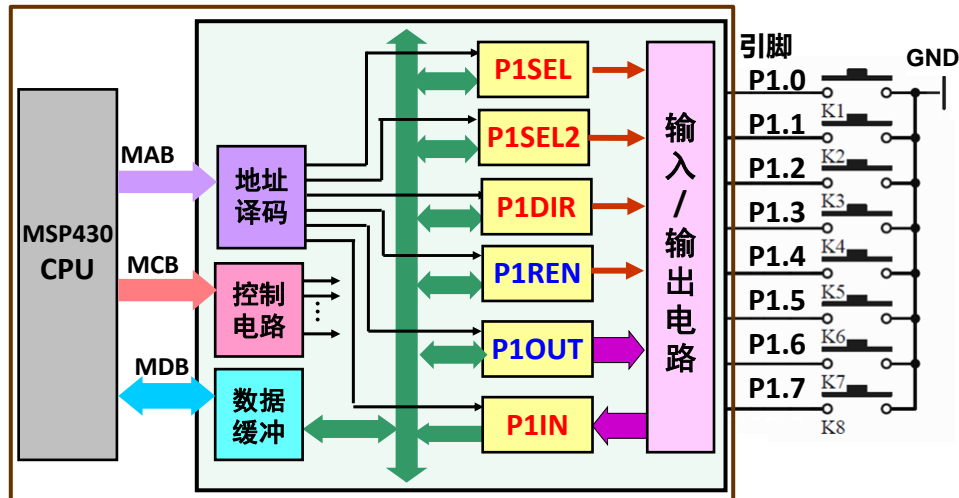
课堂上介绍了I/O寄存器**P2SEL**、**P2SEL2**、**P2DIR**、**P2OUT**与单片机**P2.0~P2.7**作为输出引脚的关系。



设置好I/O寄存器**P2SEL**、**P2SEL2**、**P2DIR**的值后，改变**P2OUT**的值，就可以改变发光二极管的状态。



课堂上介绍了I/O寄存器**P1SEL**、**P1SEL2**、**P1DIR**、**P1REN**、**P1OUT**、**P1IN**与单片机**P1.0~P1.7**作为输入引脚的关系。



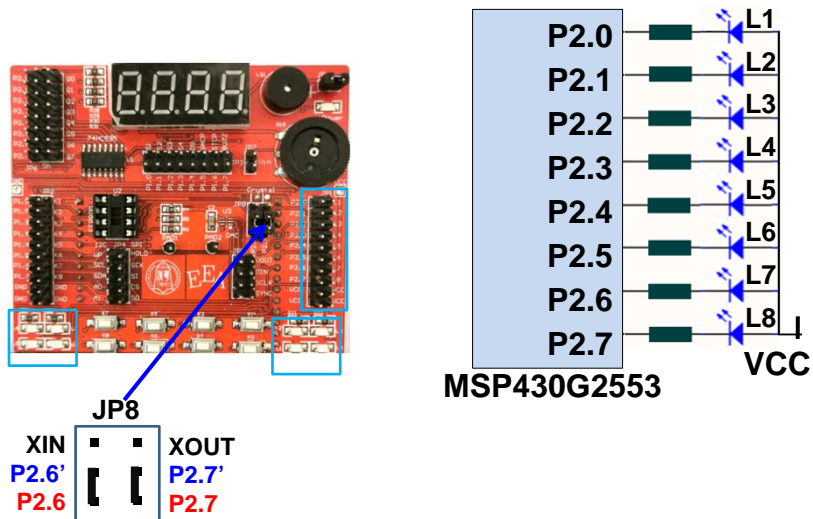
设置好I/O寄存器**P1SEL**、**P1SEL2**、**P1DIR**、**P1REN**、**P1OUT**的值，即可通过读**P1IN**获取按键状态

### 任务3 理解基本输入/输出引脚与相关I/O寄存器的关系

- 1) 理解**PxSEL**、**PxSEL2**、**PxDIR**、**PxOUT**与**引脚**之间关系
- 2) 练习在DEBUG下**只修改某位**的操作
- 3) 理解**PxSEL**、**PxSEL2**、**PxDIR**、**PxREN**、**PxOUT**、**PxIN**与**引脚**之间关系



用短线块将单片机的引脚P2.7~P2.0分别与8个LED相连



DEBUG下,  
设置P2SEL=0x00, P2SEL2=0x00, P2DIR=0xFF, 然后改变P2OUT值, 观察LED的变化

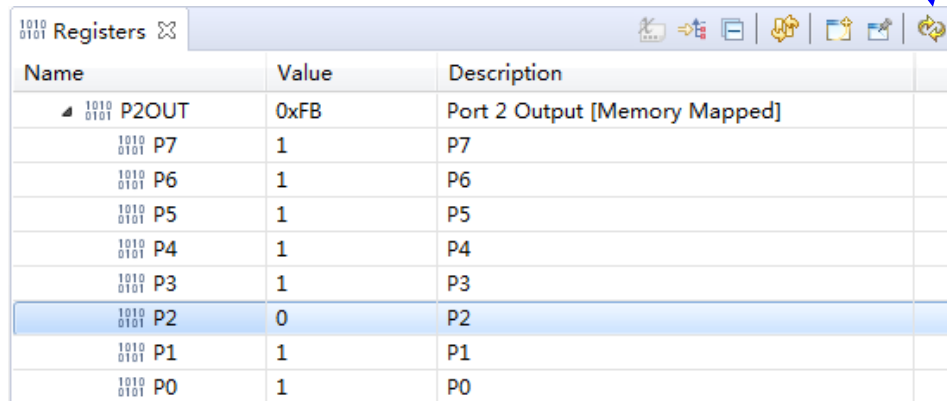
### 表1-1 I/O寄存器P2OUT和引脚P2.7~P2.0的关系

	L8~L1的状态	P2.7~P2.0的逻辑值
P2OUT=0xA6		
P2OUT=0x9C		
P2OUT=0x53		
P2OUT=0xB1		

MSP430G2553

修改端口寄存器值后，如果未见寄存器值变化，

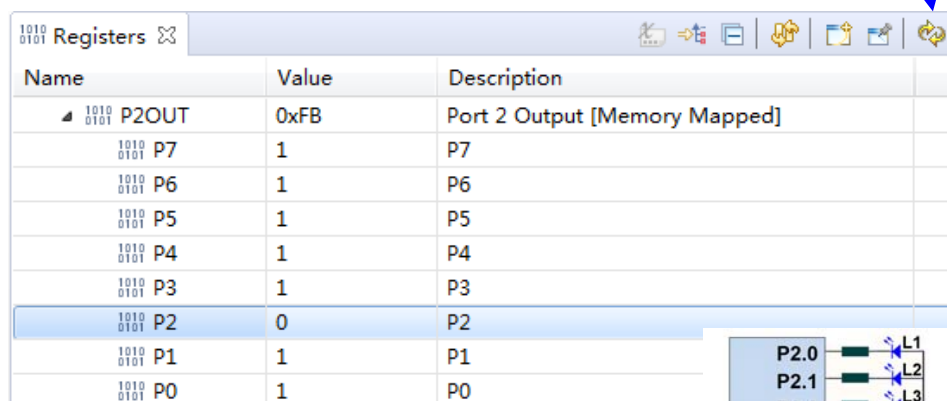
可操作 **刷新按钮**



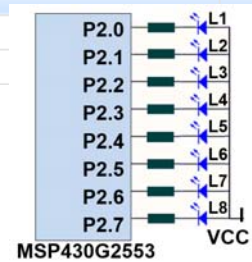
Name	Value	Description
1010 0101 P2OUT	0xFB	Port 2 Output [Memory Mapped]
1010 0101 P7	1	P7
1010 0101 P6	1	P6
1010 0101 P5	1	P5
1010 0101 P4	1	P4
1010 0101 P3	1	P3
1010 0101 P2	0	P2
1010 0101 P1	1	P1
1010 0101 P0	1	P0

### 任务3 2) 练习在DEBUG下只修改某位的操作

**刷新按钮**



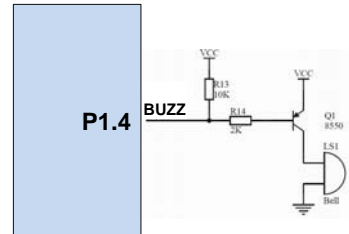
Name	Value	Description
1010 0101 P2OUT	0xFB	Port 2 Output [Memory Mapped]
1010 0101 P7	1	P7
1010 0101 P6	1	P6
1010 0101 P5	1	P5
1010 0101 P4	1	P4
1010 0101 P3	1	P3
1010 0101 P2	0	P2
1010 0101 P1	1	P1
1010 0101 P0	1	P0



如何在DEBUG下**完成控制P1.4连接的蜂鸣器发声**?  
请写下操作步骤，并上机验证。

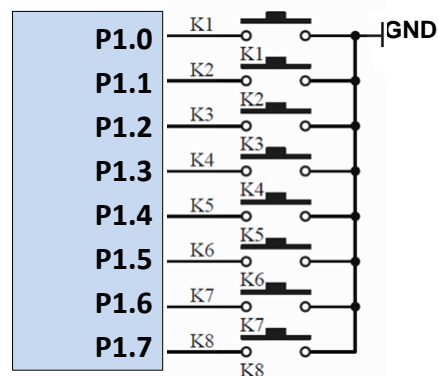
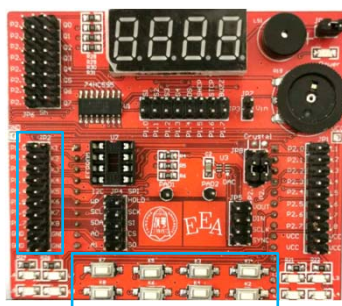
注意：  
采用在DEBUG的  
View/Register下，  
操作某位的方式

Registers		
Name	Value	Description
P2OUT	0xFB	Port 2 Output [Memory Mapped]
P7	1	P7
P6	1	P6
P5	1	P5
P4	1	P4
P3	1	P3
P2	0	P2
P1	1	P1
P0	1	P0



### 任务3 3) 理解PxSEL, PxSEL2, PxDIR, PxREN, PxOUT, PxIN 与引脚之间关系

用短线块将单片机的**引脚P1.7~P1.0**分别与8个按键相连



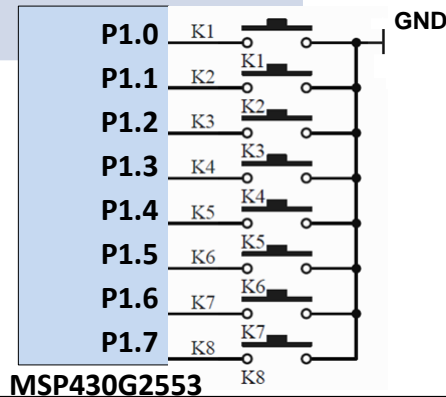
MSP430G2553

表1-2 I/O寄存器P1IN和引脚P1.7~P1.0的关系

操作	P1IN的值
按下K1	
按下K6	
同时按下K3, K5	

注意：扩展板上按键电路  
没有上拉电阻

需要设置：P1REN对应位为1  
P1OUT对应位为1



注意：查看PxIN寄存器的方法

- CCS下对 view/registers查看PxIN寄存器的显示刷新处理不够友好，即使按刷新按钮，有时也不能看到操作按键正确的结果。
- 可以在项目编译连接后，进入到DEBUG状态，设置好相关寄存器后，按下按键，然后点击运行命令（resume），点击暂停，再在view/registers下的查看PxIN寄存器中按键对应位的值时，此时可以看到变化；若还看不到，操作刷新按钮，应该可以看到。

即操作了运行命令后，暂停，CCS会刷新显示。