

## 实验2. 单片机C语言程序结构

### 一、实验目的

1. 掌握单片机C语言程序结构
2. 掌握单片机C语言程序调试方法
3. 掌握用基本输入输出功能控制外部接口电路的基本方法

#### 调试方法:

1. 外部电路的检测
2. 单片机引脚与外部电路连接后的引脚功能检测
3. 程序调试方法  
(课前实验任务2  
断点、单步、变量查看)

■ 单片机C语言程序结构

主程序main()函数包含两部分: 初始化部分和主循环部分;  
初始化部分是对MCU的软、硬件初始化;  
主循环部分是MCU实时处理部分

```
#include "msp430.h"
//全局变量定义
int main( void )
{
    //局部变量定义
    WDTCTL = WDTPW + WDT;
    //初始化部分;
    for(;;) {
        //主循环部分
        //MCU实时处理部分
    }
}
```

```
P2SEL &=~BIT6;
P2SEL2 &=~BIT6;
P2OUT |=BIT6;
P2DIR |=BIT6;

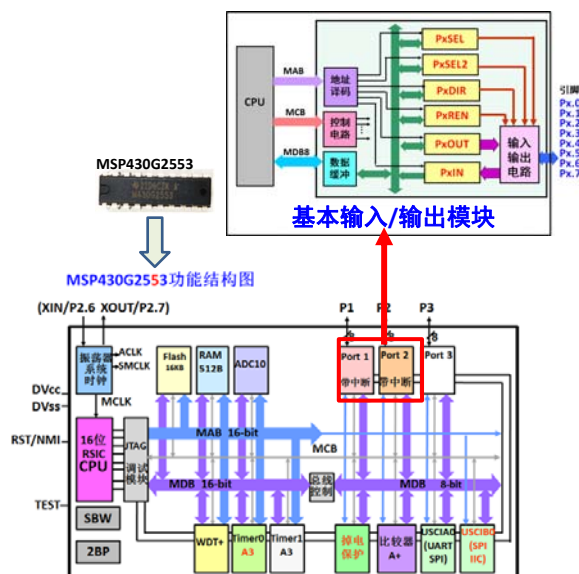
while(1){
    P2OUT &=~BIT6;
    delay();
    P2OUT |=BIT6;
    delay();
}
```

### 实验1

#### 二、实验任务

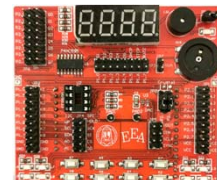
1. 了解实验板结构和掌握扩展板检测方法  
(检测单片机外部电路, 检测的电路部分与单片机无关, 今后做实验出现问题时, 比如LED状态不变, 按键没有反应, 蜂鸣器不响等用该方法可以判断是否在单片机外部硬件电路出了问题)
2. CCS开发工具的初步使用  
掌握创建工程, 编辑、编译、下载、运行程序, 查看和修改寄存器、变量、存储单元的方法
3. 理解基本输入/输出引脚与相关I/O寄存器的关系  
了解单片机内部端口P1~P2的选择PxSEL、方向PxDIR、输出PxOUT、上/下拉电阻使能PxREN、输入PxIN等各寄存器作用, 理解它们对单片机引脚的控制功能。

## 基本输入/输出模块作用?



### 实验2

LED  
有源蜂鸣器  
数码管



明天实验3  
按键  
红外避障  
小车控制

无源蜂鸣器



红外避障模块

直流电机



前进、后退、  
左转、右转

遥控小车



## 二、实验任务

- ## 1. 掌握程序结构和调试程序练习

(参照实验讲义提供的步骤练习)

- ## 2. 控制发光二极管的显示变化

(建议参看课件例，使用表格和循环方法)

- ### 3. 模拟显示小车运动状态

- #### 4. 数码管的显示控制(含部分提高)

- ## 5. (提高) 制作呼吸灯

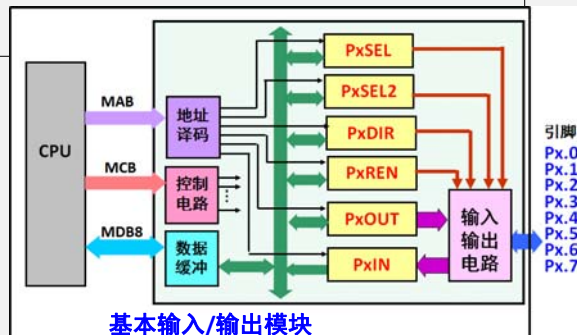
注意:任务3使用对某位置0、置1或求反的方法  
(参看第1章课前练习)

### 三、实验作业

提交任务2、3、4的.c 程序代码，可以粘贴到一个word文档中，每个程序前标注是哪个任务的。

## ■ 单片机程序结构

```
#include "msp430.h"
unsigned int j;           //定义全局变量
int main ( void )
{ //unsigned int i;       //定义局部变量
  //register unsigned int i; //定义寄存器型局部变量
  WDTCTL = WDTPW + WDTOLD; //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
  }
}
```



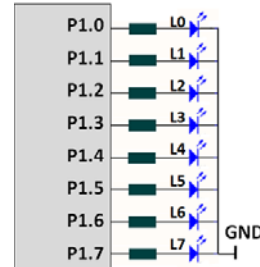
**方法1：用数组存放显示表(C3\_e2\_1.c)**  
(表格法，课件第三章例2)

```

#include "msp430.h"
void delay(unsigned long numLoop) (略);
int main( void )
{
    unsigned char LEDdata[8]={0x01,0x02, 0x04,0x08,0x10,0x20,0x40,0x80};
    unsigned int i;
    WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
    P1SEL=0x00; //设置P1为基本I/O
    P1SEL2=0x00;
    P1OUT=0x00; //使8个LED全灭
    P1DIR=0xFF; //设置P1为输出端口
    while(1) //主循环
    {
        for ( i=0; i<8; i++) //8个LED依次点亮
        {
            P1OUT=LEDdata[i];
            delay(0xf000); //调用延时子程
        }
    }
};

void delay(unsigned long numLoop) (略);

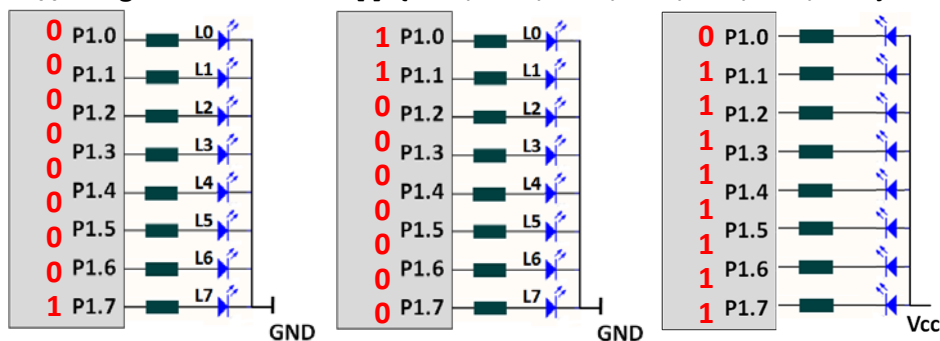
```

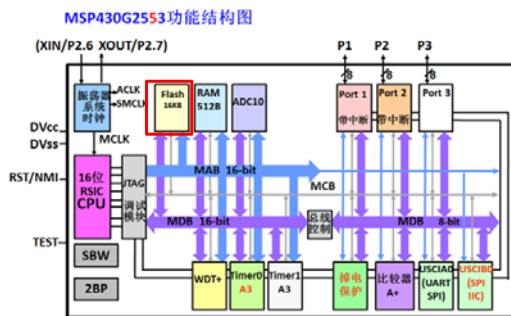


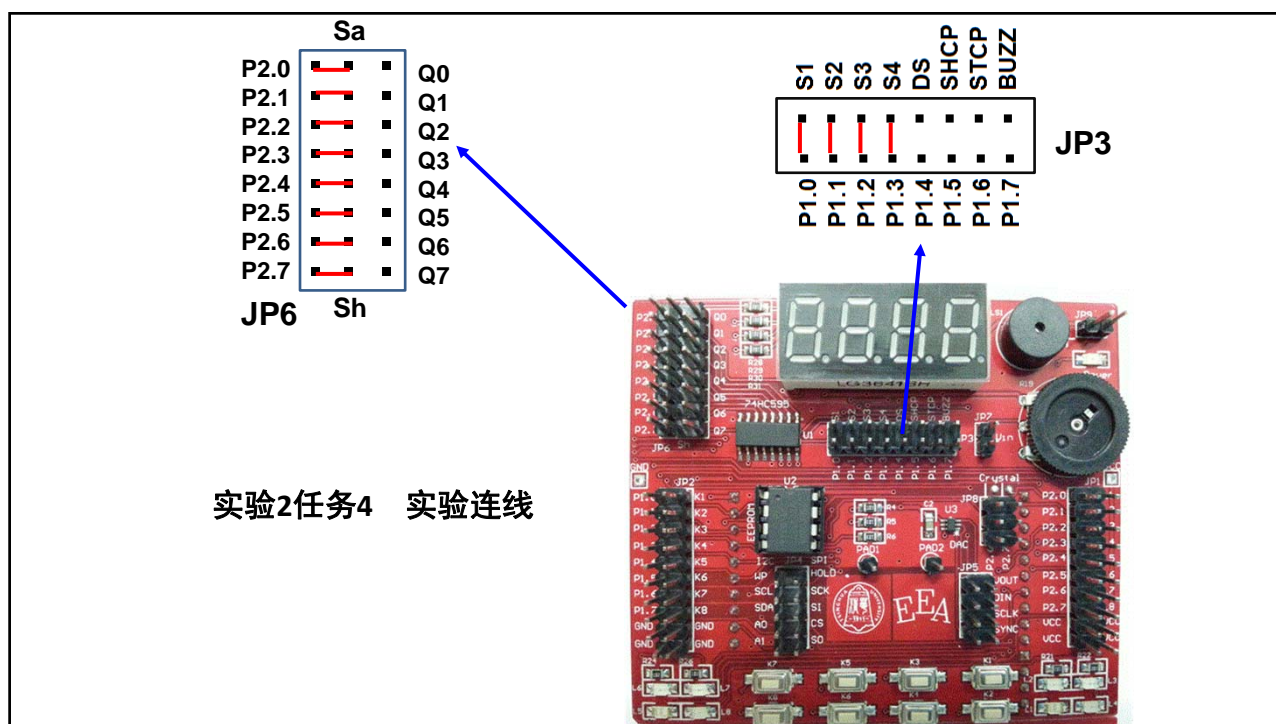
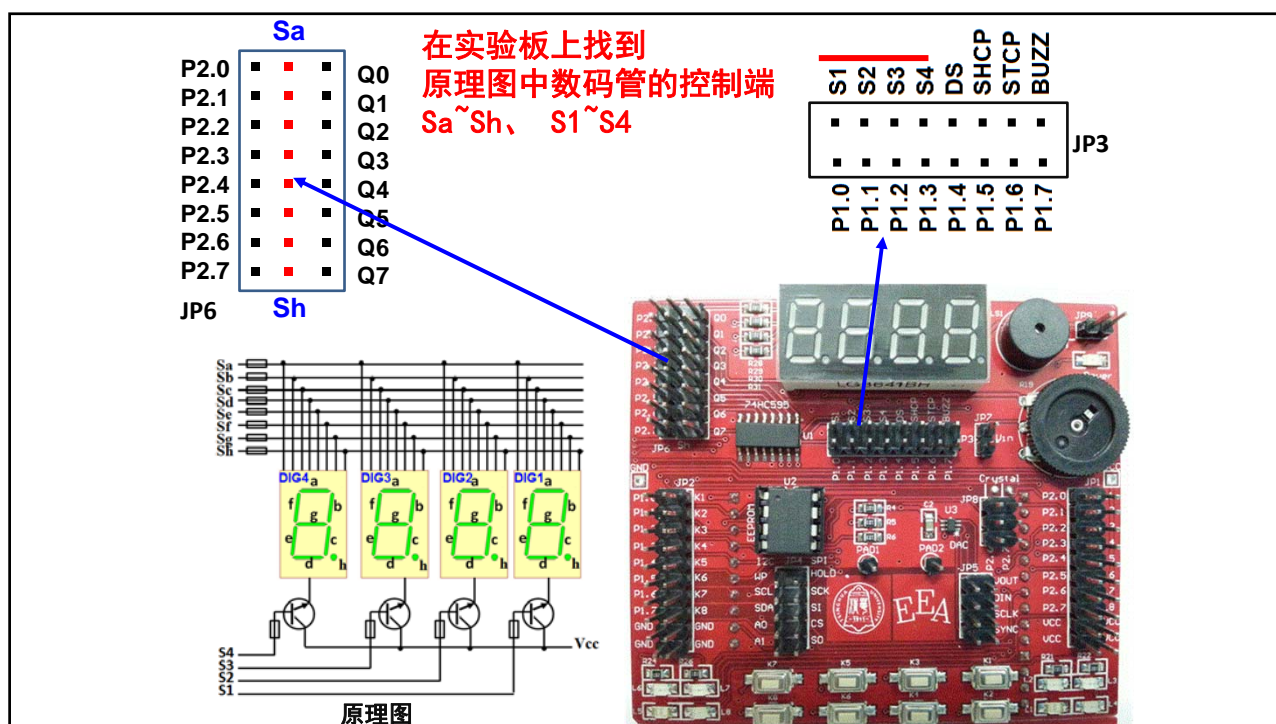
```

int main( void )
{
    // L0→L1....→L7→L0 从上到下
    unsigned char LEDdata[ ]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
    // L7→L6....→L0→L7 从下到上
    //unsigned char LEDdata[ ]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
    //相邻两个亮L0 L1→L1 L2....→L6 L7→L7 L0
    //unsigned char LEDdata[ ]={0x03,0x06,0x0C,0x18,0x30,0x60,0xc0,0x81}
    //共阳极 L0→L1....→L7→L0 从上到下
    //unsigned char LEDdata[ ]={0xFE,0xFD,0xDB,0xF7,0xEF,0xBF,0x7F}
}

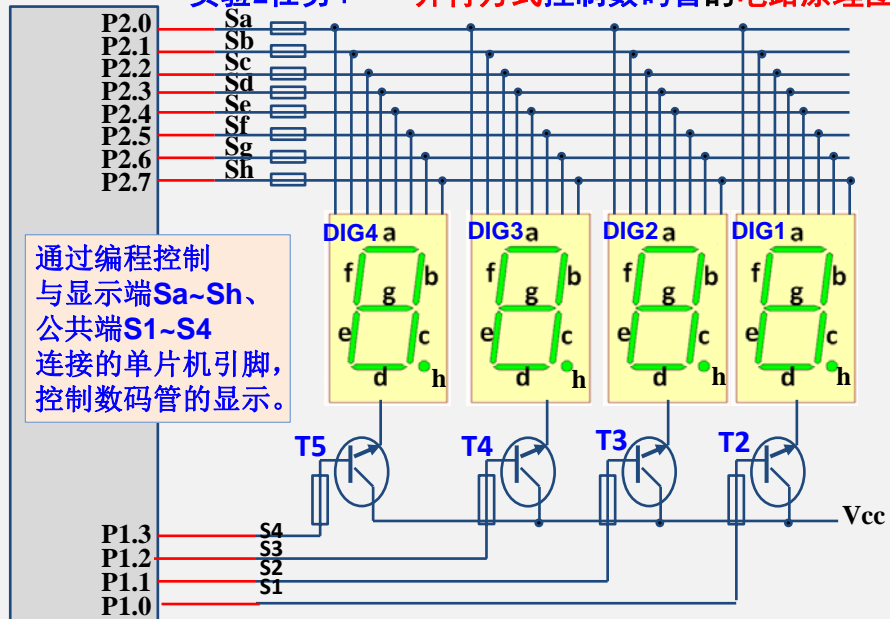
```



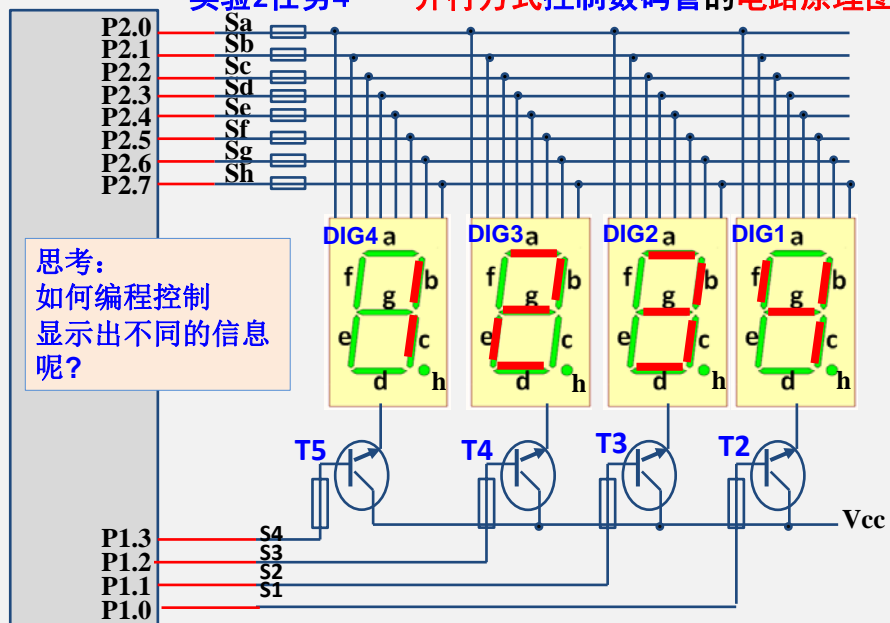




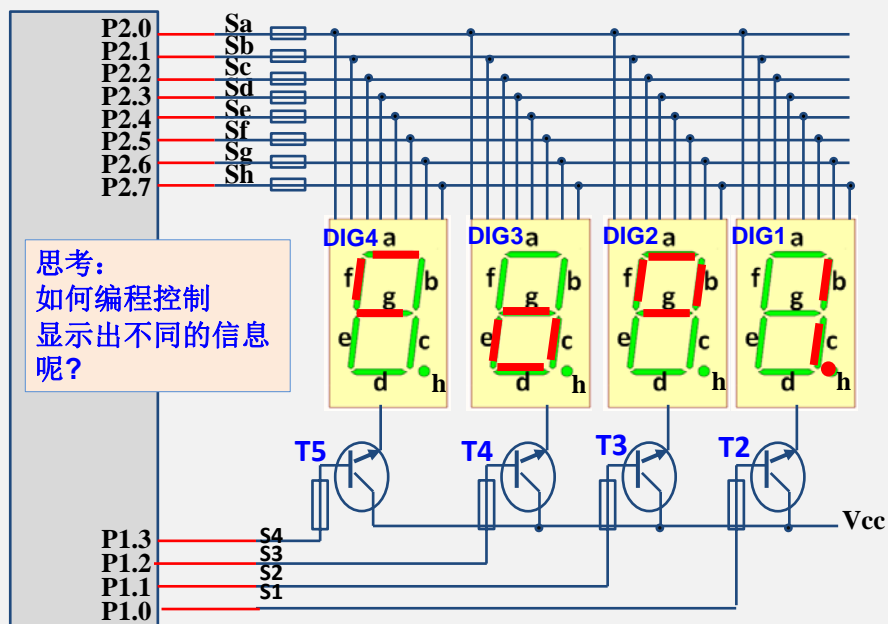
### 实验2任务4 并行方式控制数码管的电路原理图



### 实验2任务4 并行方式控制数码管的电路原理图



## 实验2任务4 并行方式控制数码管的电路原理图



注意：数码管的显示码控制端Sa~Sh直接与单片机引脚相连

### 说明：

若要控制4个数码管分别显示不同的内容，任何时刻只有一个数码管被点亮，另外3个处于灭的状态，但只要在1秒钟内，每个数码管能显示30次以上，利用眼睛的视觉停留效果，则人眼的感觉是几个数码管在同时显示的。

本质上就是控制数码管一个个分别显示。

### 思考：

每个数码管显示的时间不能太长，也不能太短，为什么？



## ■ 注意

做实验，新建项目时，需要去设置编译优先级为off；  
在原有设好的项目内，新添加文件，不用重设优先级。

