# 实验2 单片机C语言程序结构

1. **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. 掌握用基本输出功能控制外部接口电路的基本方法。

二．**实验任务**

**1. 程序结构和调试练习，掌握CCS的调试手段**

用跳线块将实验板上的8个发光二极管的控制端L8~L1与单片机的8个引脚P2.7~P2.0相连。编写的程序L2\_debug.C如下，功能是控制与单片机相连接的8个发光二极管在全亮与全灭这两种状态下循环变换。该程序没有语法上的错误，但运行后不能产生上述现象。请**参照提供的“实验2任务1调试练习举例.doc”文档做练习。结合自己的调试过程，总结调试方法，**为课程后续实验打好基础**。**

***说明：***除新建一个Lab2项目外，也将L2\_debug.c拷贝到实验1的项目test文件夹下，或用add方式将其加入到项目中，用exclude from Build 将项目自带的main.c移出。然后**灵活应用resume(运行)、suspend(暂停)、restart(复位)、F5(单步)和设置断点**等几种执行命令**进行调试，控制程序的运行**，结合实验的现象，检查程序中用到的端口寄存器、以及变量的值，查找和定位程序中的问题，调试出正确的程序。

**断点的作用**：当程序的执行流程执行到该语句时，会暂停下来，此时用户可以利用DEBUG下的View命令查看当前程序执行的状态，包括单片机内容寄存器、变量和存储单元的内容，利用这些运行过程的中间值，来查找问题。

L2\_debug.C程序清单（提供电子文件）：

#include "msp430.h"

void delay( );

int main ( void )

{ WDTCTL = WDTPW + WDTHOLD; //关闭看门狗

P1DIR=0xff; //设置端口1为输出方向

P1OUT=~P1OUT; //将端口1的值取反后输出

delay( ); //调用函数延时

}

void delay( ) //延时函数

{ unsigned char j;

for (j=0;j<5;j++);

}

1. **控制发光二极管的显示变化**

在任务1连线基础上，编写一个程序控制8个发光二极管，按下面的2种规律循环变换，即 规律1🡪规律2🡪规律1🡪规律2🡪……，如此循环反复。(建议使用表格法，参看课件例程第3章例2，表格法是将8个LED的变化规律放在数组中，然后用数组的值控制单片机端口8个引脚的输出)

* 规律1：8个LED灯从L4🡪L3🡪L6🡪L5🡪 L8🡪L7🡪L2🡪L1一个一个单独点亮，每次只有一个灯亮，其他灯灭；
* 规律2：8个LED灯两两点亮， 顺序为L4和L6，L3和L5，L1和L7，L2和L8，每次只有两个灯亮，其他灯灭。

***注意：***在灯的状态变化中，需留一定延时，以便视觉能感受到灯的亮灭。

***思考：***如果选择用单片机的P1端口控制8个发光二极管，如何在实验板上设计连线、并编程实现任务2的功能？

注意：CCS下编辑源程序时，默认字体较小。可点击CCS菜单栏Windows>Preferences>general>

appearance >colors and fonts，再点击右侧对话框C/C++>Editor下的C++ Editor text font，设置字体和大小。可根据个人喜好，进行选择设置。如图2-1。

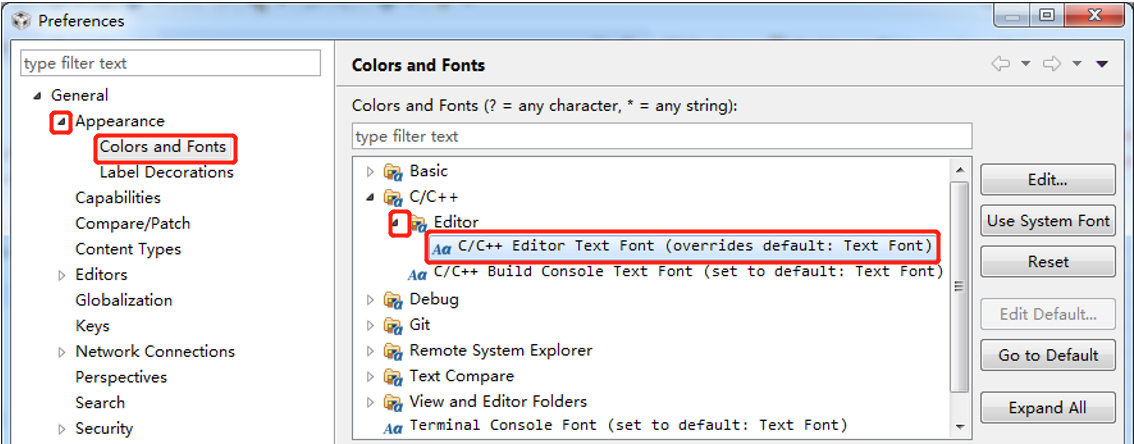


图2-1 在CCS下设置编辑窗口的字体大小

1. **模拟小车运动状态的提示**

用单片机的5个引脚分别连接L6、L3、L5、L4四个LED灯和一个蜂鸣器，其中四个LED分别代表小车的左前灯、右前灯、左后灯、右后灯。编程顺序显示小车的下面几种运动状态：

1. L6、L3亮，表示小车前行；
2. L5、L4闪，并发出蜂鸣提示声，表示小车后退；
3. L3、L4闪，表示小车右转；
4. L6、L5闪，表示小车左转；
5. 灯都熄灭，表示小车停止运动。

注意：请结合第1章课前练习，编程时请采用对某位、或某些位进行置0、置1的方法，不影响未使用到的引脚相关的寄存器设置。

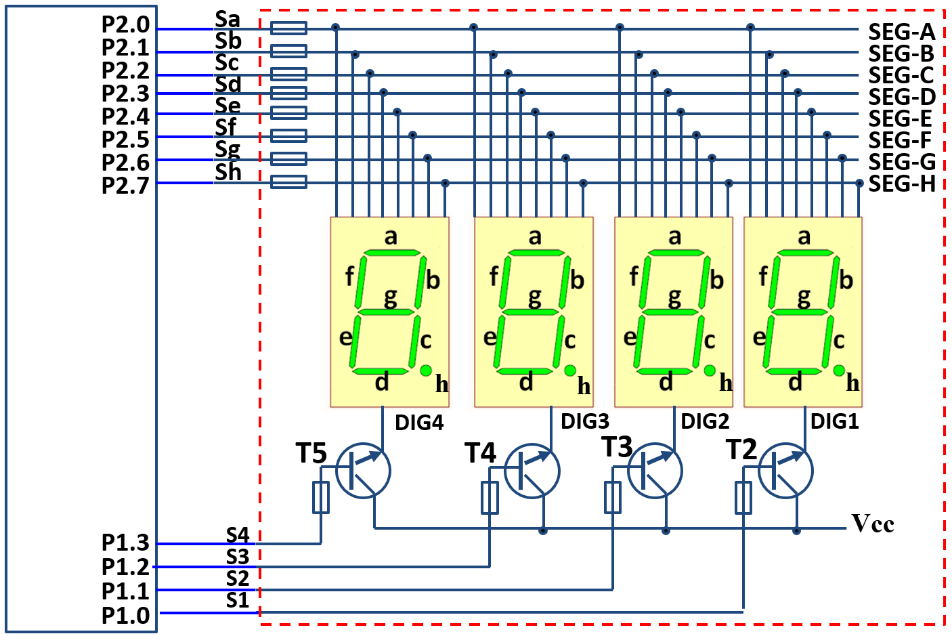
1. **并行方式控制数码管的显示**

参看提供的数码管分析课件。图2-2中虚线内部是MSP430G2553扩展板上4个数码管的相关电路，其中芯片SEG集成了4个8段数码管，该芯片共12个引脚，引脚SEG-A~SEG-H是数码管的8段显示控制端，引脚DIG4~DIG1分别是4个数码管各自的公共端。图2-3是单个数码管的原理图。数码管外围各部分电路如图2-2已在电路板上焊好，控制信号Sa~Sh、S1~S4已引到实验板对应的插针上，实验时用跳线或导线可方便与其它部分电路（如单片机的引脚）连接。完成下面任务：

1. 分析图2-2虚线框内中4个数码管电路工作原理，利用VCC和GND信号，检测4个数码管；
2. 按图2-2接好单片机与数码管间的硬件连线，采用并行方式控制数码管工作，分析控制方法；
3. 编程控制数码管DIG-1上依次显示0、1、....、E、F，然后熄灭，再分别是数码管DIG2、DIG3、 DIG4依次显示0~F，如此反复。
4. **(提高)**数码管同时显示“cool.”，如图 2-3。注意
5. **(提高)** 控制四个数码管同时显示数组变量string[4]的信息，其中数组各元素数值只在0~9之间。假如string[4]={1, 2, 3, 4}, 则四个数码管上依次显示“1234”；假如string[4]={5, 2, 0, 1}, 则四个数码管上依次显示“5201”。实验时，运行程序后，可在DEBUG下，用View/Memory browser, 输入 string后，修改string的值，查看数码管的显示变化。

***说明：***若要控制4个数码管同时显示不同的内容，如“cool.”，在实现上是通过每次只点亮1个数码管，而另外3个处于灭的状态。只要在1秒钟内，每个数码管能轮流显示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。

***思考：***在显示“cool.”时控制每个数码管显示的延时时间不能太长，也不能太短，为什么？



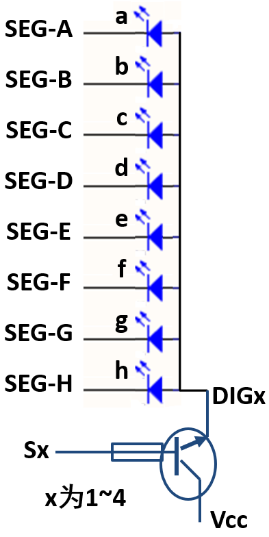


图2-2 实验板上数码管连线图 图2-3 单个数码管的原理图

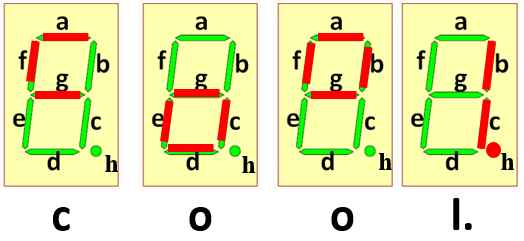


图2-4 数码管显示“CooL.”

**5. (提高)制作呼吸灯**

阅读下面程序L2\_LEDs.C，给程序加上适当注释，用P2.2和P2.5连接两个LED，运行L2\_LEDs.C，观察现象，思考控制LED亮度变化的原理。上网查找什么是呼吸灯，在L2\_LEDs.C的基础上，利用单片机编程控制LED，实现一个效果良好的呼吸灯，即慢慢变亮，又慢慢变暗，如此反复。

L2\_LEDs.C程序清单（提供电子文件）：

**#include** "msp430.h"

**unsigned** **int** L=0,M=0;

**int** **main** ( **void** )

{ **unsigned** **int** i;

WDTCTL = WDTPW + WDTHOLD; //关闭看门狗

P2SEL &=~(BIT2+BIT5);

P2SEL2 &=~(BIT2+BIT5);

P2OUT |=BIT2+BIT5;

P2DIR |=BIT2+BIT5;

**for** (;;) //主循环

{ L=L+1;

**if** (L>0x200) L=0;

M=0x200-L;

P2OUT &=~(BIT2+BIT5);

**for** (i=0; i<L; i++);

P2OUT |=(BIT2+BIT5);

**for** (i=0; i<M; i++);

};

}

**注意：本次的**编程任务2~5的功能只是做了一个基本说明，对运行的现象（结果）并没有硬性的标准和要求，**大家可根据自己程序实际运行的情况，不断完善，使程序的运行效果逐步更加趋于合理、顺畅，操作控制更便捷**。