

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. 掌握外部电路基本控制方法。

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++);
}
```

2. 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, 表格法是将 8 个 LED 的变化规律放在数组中, 然后用数组的值分别输出到 8 个 LED 上)

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

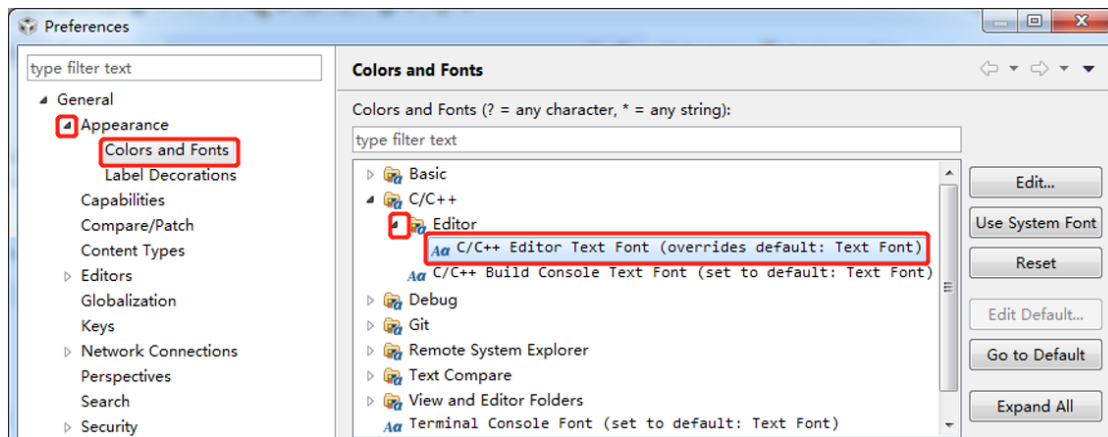


图 2-1 在 CCS 下设置编辑窗口的字体大小

3. 用按键控制发光二极管的显示变化

在任务 2 基础上，增加 2 个按键与单片机的引脚相连，编程实现由按键控制发光二极管的显示变化：当按下实验板上的 K2 键时，8 个发光二极管按任务 2 中的规律 1 变化；当按下实验板上的 K4 键时，8 个发光二极管按任务 2 中的规律 2 变化。要求每按下一次键，蜂鸣器响一下。

4. 模拟小车运动状态的提示

用单片机的 P2.0~P2.3 四个引脚分别连接 L6、L3、L5、L4 四个 LED 灯，P1.0~P1.4 分别连接 K1、K2、K3、K4、K8 五个按键，P1.6 连接蜂鸣器，其中四个 LED 分别代表小车的前左灯、前右灯、后左灯、后右灯。编程用这四个 LED 显示小车的运动状态：

- 1) 按下 K1 键，小车前行，L6、L3 亮；
- 2) 按下 K3 键，小车后退，L5、L4 闪，并发出蜂鸣提示声；
- 3) 按下 K2 键，小车右转，L3、L4 闪；
- 4) 按下 K4 键，小车左转，L6、L5 闪；
- 5) 按下 K8 键，小车停转，灯都熄灭。

注意：请结合课前练习，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。

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+5;
        if (L>0x500) L=0;
        M=0x500-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 2~4 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。