**实验2-1 任务1调试练习举例**

1. 查看一下项目的编译优化级是 off。
2. 先点击运行一次程序，但看不到预期的现象，点击暂停程序，发现程序停在不是main()，也不是delay()函数处，停在了一个abort()函数内，说明程序跑飞，不受自己编写的程序控制了。一看程序，从结构上，缺少了主循环，所以赶紧如文本框1加上while(1){ }主循环。

**//文本框1**

**while**(1){ //主循环

P1OUT=~P1OUT; //将端口1的值取反后输出

delay( ); //调用函数延时

}

1. 重新编译连接，出现图2D-1的提示程序修改，是否重新加载的界面，点击yes, 重新下载编译连接好的程序到单片机中。如果不出现图2D-1界面，可点击Run>Debug，重新下载程序到单片机中。

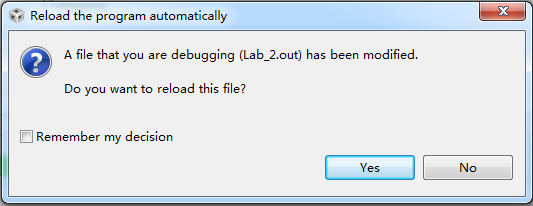


图2D-1 提示程序修改后提示是否重新加载的界面

1. 在Debug窗口，点击运行程序，仍看不到预想的现象。点击暂停程序，指针指在程序的某条语句上，本例一般停在delay()函数处，说明程序不跑飞了。点击复位程序，准备重新执行。(如果图标是灰色不可用，可以点击一下非main.c的编辑窗口，如console窗口，然后图标会变成可用)。因为程序比较短，可点击或键盘上按下F6，采用单步（step over）方式一条一条语句运行程序，当运行到P1OUT=~P1OUT语句时，连接的LED灯均没有反应。

仔细检查接线和程序，发现接线是P2.0~P2.7，但程序控制的是端口1，程序与接线不对应， 把程序中的P1改P2，如文本框2。

**//文本框2**

P2DIR=0xff; //设置端口2为输出

while(1){

P2OUT=~P2OUT; //将端口2的值取反后输出

delay( ); //调用函数延时

}

1. 为加快查找问题，不用单步执行整个程序，改用断点方式控制程序执行。如图2D-2双击语句P2OUT=~P2OUT最左侧，在此处加一个断点，然后点击运行程序，可以看到灯有亮灭，开始有现象，但不是同时亮同时灭。应该是没有设置P2OUT初值造成。如文本框3，在while(1)主循环前加上P2OUT=0。

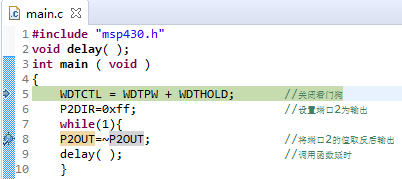


图2D-2 设置断点调试

**//文本框3**

P2OUT=0； //设置显示初值

P2DIR=0xff; //设置端口2为输出

while(1){

P2OUT=~P2OUT; //将端口2的值取反后输出

delay( ); //调用函数延时

}

1. 重新编译连接下载后，进入DEBUG，此时断点还在语句P2OUT=~P2OUT处，运程程序，发现LED灯L8的显示亮度异常，且不该亮时也半亮。因其连接的引脚是P2.7，如图2D-3，用View/Registers 查看与端口P2有关的端口寄存器，发现P2SEL的P6、P7两位上电复位值不为0，而是1，从而使P2.6、P2.7上电复位后其功能不为基本输入输出，而是其他功能，故不能正确控制连接的LED灯L7、L8。

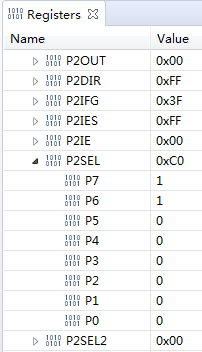


图2D-3 查看引脚相关IO寄存器的值

1. 在程序入口处加上对P2SEL、P2SEL2的初始化设置，如文本框4，并保持图2D-2中P2OUT=~P2OUT前的断点设置，反复点击运行程序，观察到现象对了：8个灯同亮，同时灭。

**//文本框4**

P2SEL=0; //设置端口P2的8个引脚为基本输入输出功能

P2SEL2=0;

P2OUT=0； //设置显示初值

P2DIR=0xff; //设置端口2为输出

1. 双击语句P2OUT=~P2OUT的最左端，取消P2OUT=~P2OUT前的断点，再点击运行程序，发现8个灯一直全亮，看不到全亮、全灭交替循环的现象。说明，P2OUT=~P2OUT 后delay()函数延时时间太短。

将**for** (j=0;j<0x5;j++); 中的5改为0xff，再运行，现象连续运行、断电运行的现象和之前相同，延时时间还是太短。

1. 将**for** (j=0;j<0xff;j++)改为j<0xffff, 运行程序，仍然看不到现象，且即使在P2OUT=~P2OUT前加断点，也看不到现象。同时发现程序不会在设置的断点P2OUT=~P2OUT处停下，也就是程序一直处于运行状态。当点击暂停运行，程序总是停在delay()函数处，说明程序没有跑飞。

双击delay()函数的语句 **for** (j=0;j<0xffff;j++)的最左端，在此添加一个断点。并且如图2D-4，用View/Express或View/Variables，查看变量j。

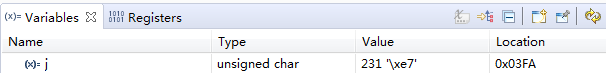


图2D-4 查看变量的值

1. 可点击图2D-4中j的Value处，将变量j的值改为0xff，然后用F6单步执行程序，发现此时j的值从0xff变成了0，也就是j的最大是0xFF，永远<0xFFFF，for循环的条件永远成立，这样，delay（）函数永远在执行，不会结束。从而也就不会再循环执行语句P2OUT=~P2OUT，产生8个LED全亮、全灭的变化。

原因是j为 unsigned char类型，即无符号char类型，其最大值为0xff, 而0xff<0xffff永远成立。

1. 将j的类型改为 unsigned int，再运行程序。语句P2OUT=~P2OUT前有断点和没断点，现象都正确。

**至此程序基本调试完毕。**

1. 如果将上面调试好的程序，将编译优化级从off改为1-Registers 级，会发现现象又不对了。所以，建议做实验时，将编译优化级置为off，即编译时对程序不做太多优化处理。

**通过上面调试练习，掌握****resume(运行)、****suspend(暂停)、****restart（复位）、F6（step over）和断点设置等几种执行命令，并能灵活应用它们控制程序的运行，结合查看寄存器、以及变量的值，定位和查找程序中的问题，调试出正确的程序。**