

# STC 单片机片上仿真法

文/杜洋

有单片机初学者问了我这样一个问题：单片机真是个好东西，可以实现我的很多想法，就是在编程开发的时候太麻烦，每次改动都要重新编译、下载，再等待着问题的出现。仅调试一个参数就要花上几个小时的时间。对于我们这些没什么经验的编程菜鸟来说太麻烦了。我想单片机技术发展至今，应该有更便捷的开发工具吧。杜老师你平时是怎么开发单片机软件的？有什么秘诀传授一下呗！

我的回答是：当然有秘诀，那就是使用“仿真功能”。什么是仿真？它如何实现更快捷的开发呢？详见下文。

## 【什么是仿真】

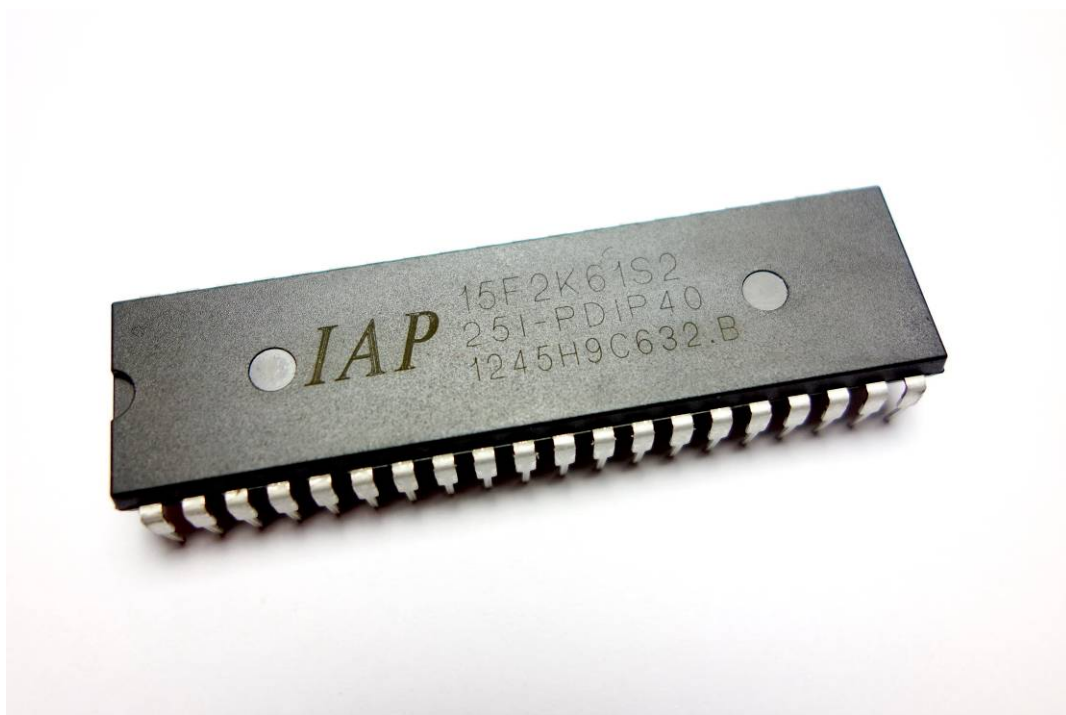
什么是仿真？我们举一个汽车设计的例子来说明一下吧。大家一定在影视剧或电视广告中看过这样的镜头：在一个大大的厂房里，一辆崭新的小汽车正在以很高的速度撞向一面厚厚的水泥墙。坐在车上的两个人面不改色心不跳，一动不动地等待着死亡。他们为何如此冷静，因为“他们”是实验用的假人。说时迟那时快，汽车已经撞到了墙上，巨大的声响夹杂着飞溅的碎片充满了空间，汽车在撞击中破了相。旁边的几台高速摄像机记录下了这一切。这是一次真实的撞击实验，目的是为了得出这款车型在出现意外时，是否能保住人的小命。安装在假人身上的传感器所得出的数据，能帮助工程师们发现安全隐患，改进汽车的设计。可以说以上就是一次仿真，一辆真车和两个假人有计划地撞墙，模拟了真实车祸情况。仿真让实验变得可能，因为没有真人愿意坐在车里参与这场实验。当真实情况很难在开发时再现时，仿真就可以帮助开发者完成必要的实验。这就如同单片机开发中，我们在自己的实验板上去开发一款产品一样。当我们设计好了一个产品的功能，我们要在实验板上模拟用户的操作，看看操作是否正常，产品的反应速度和稳定性如何。这些都是在仿真——模仿用户使用的真实情况。

有朋友会问了：如果这就是仿真，那还有什么好讲的呢？嗯，如上所说的仿真只是广义上的仿真，凡是在实验室里用实验板或工程样机模拟用户使用的过程，都可以算是仿真。而还有一种狭义的仿真，就是下面要重点介绍的内容。再说回到汽车撞击的实验吧。后来呀，汽车公司的老板在办公室里坐不住了。因为每当从外面传来一声巨响，他就知道又有一辆新车被撞得稀巴烂，一阵痛苦涌上心头。虽然理性上明白，这是为了开发出更安全的汽车，可是感性上还是不喜欢这种烧钱的行为。人们常说利益推动科技进步，当老板的利益受损，自然就会有高科技问世。不久，工程师们用上了一种电脑仿真软件，它采用了虚拟现实的技术，只要在电脑上按几个钮，输入一些汽车的参数，一台虚拟的汽车就出现在屏幕上。这辆虚拟车能和真车一样撞击、飞溅，然后得出一大堆接近真实的数据。不仅能模拟真实的撞，还能歪着撞，倒着撞，飞起来撞，飞起来转体 360 度地撞。还能暂停时间，一步一步撞，或者只看撞击中某一秒的数据。这一技术完全超越了真实实验，撞击再也不用耗人耗时了，新车的开发速度也快了很多，大大降低了成本。

汽车公司的遭遇在单片机公司也同样发生着，各种仿真软件如雨后的水泡子般越来越多。有的直接在电脑上虚拟仿真，还有的用一种叫仿真器的东西，让实验板与电脑连接，给实验板或工程样机增加了单片机实物所不能达到的仿真功能。其中最重要的一个功能就是“单步运行”了。在仿真软件里，把单片机从上电开始以正常的速度一直运行下去的过程叫“全速运行”。相对的，如果单片机只运行程序中的某一条或几条程序就是单步运行了。在非仿真的情况下，单片机是不能单步运行的。那单步运行有什么用呢？呵呵，单步运行非常有用呀，甚至可以说是一项单片机开发的重大进步，就如同录音带和 MP3 的区别一样。录音带在听歌的时候必须从头听到尾，如果想换歌就得花时间倒带，而且你也不能精准地倒到下一首歌的开始处。而 MP3 不是连续的线性存储，你可以随意换歌，还能把任意的一段反复听。在单片机的开发中，我们为了测试某个部分的功能，必然要从头运行，再跳过不必要部分才能达到。大把的时间浪漫在多余的

劳动上面了，现在有了单步运行，你想到哪就到哪，你想反复运行某段程序也没问题。期间你还能修改大部分参数，不仅能模拟真实的运行，还能歪着运行，倒着运行，飞起来运行，飞起来转体 360 度地运行。好玩吧！？

而本文要介绍的是仅使用单片机的“片上传真”。片上仿真是基于单片机本身的仿真，也就是说只要一片单片机，不需要额外购买别的东西，就可以实现仿真。对于单片机爱好者来说，片上仿真是最高性价比的选择，你甚至不需要修改电路，用给单片机下载 HEX 文件的电路就能实现仿真。各大单片机公司都已开发出不同性能的支持片上仿真的单片机。其中 STC 公司有一款性能很不错的片上仿真单片机——IAP15F2K61S2。下面我就给大家介绍一下片上仿真环境的建立，还有进入仿真界面的操作流程，最后以一个实例告诉大家如何进行仿真调试。这是一些非常简单的操作，只要按照本文的步骤操作，就一定能顺利学会。

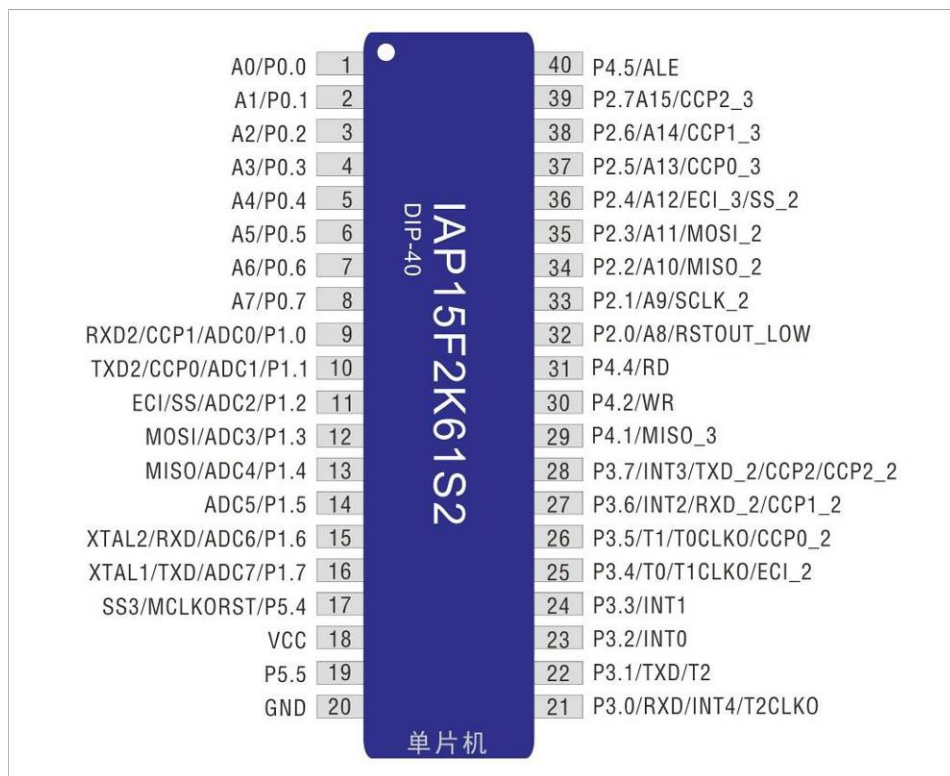


带有仿真功能的单片机 IAP15F2K61S2

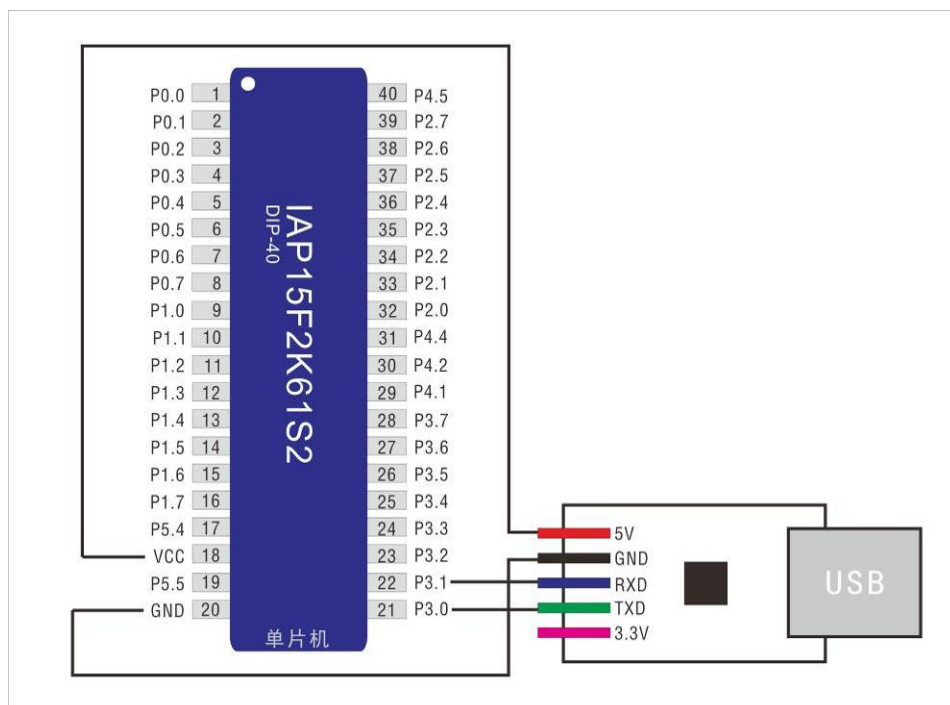
## 【仿真电路连接】

建立单片机仿真环境，最主要的工作是设置软件的参数，而硬件电路的连接是很简单的事。如果你已经根据本书第一章的内容制作出了 ISP 下载线，并成功地给单片机写入了 HEX 文件。那么从某种意义上讲，你已经完成了单片机仿真的硬件电路连接。也就是说，单片机仿真电路与 ISP 下载电路是完全相同的。如果是这样，我为什么还要花时间来讲仿真电路的连接呢？因为虽然电路连接相同，但单片机不同了。所以我们不能使用 STC12C5A60S2 和 STC12C4052AD 来仿真，因为这两款芯片不带仿真功能。而唯一带仿真功能的单片机 IAP15F2K61S2 是最新发布的 15 系列单片机，它有着不相同的引脚定义。我们只要熟悉了它的新引脚定义，再来制作电路就不难了。

首先最明显的是引脚定义的不同，虽然 IAP15F2K61S2 也是 40 脚的单片机，但如果把它直接插在我们做好的 ISP 下载线里，你会发现单片机是不工作的。不仅 IO 接口不兼容，连 VCC 电源输入的位置也不同。接下来是外部晶体的使用，IAP15F2K61S2 单片机不需要接外部晶体，因为它的内部集成了一个高精度的时钟源，可以用软件设置成 5~30MHz 的时钟频率。这一改进对我们使用者的意义是：不论我们做何应用，都不需要外接晶体的电路了。只要连接 VCC 和 GND，单片机就可以工作。再连接 TXD 和 RXD，单片机就能 ISP 下载和仿真了。



可仿真单片机 IAP15F2K61S2



下载/仿真电路原理图

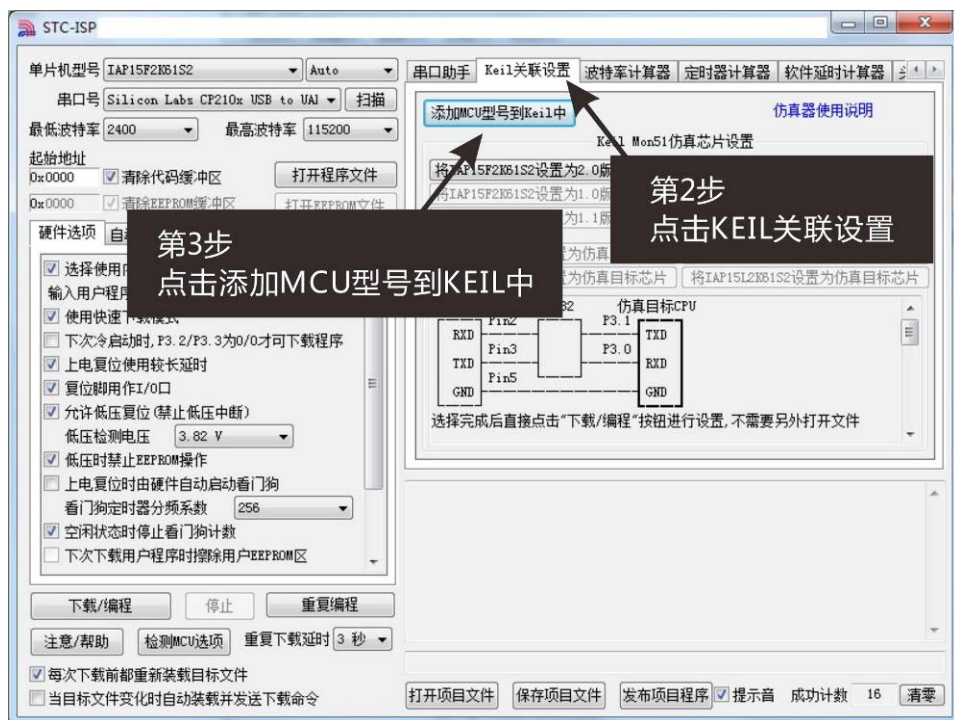
## 【新建仿真环境】

硬件电路连接完毕后，下面开始步骤较多的软件设置，请大家一定按我的步骤仔细进行。



第1步  
下载并打开STC-ISP V6.36软件

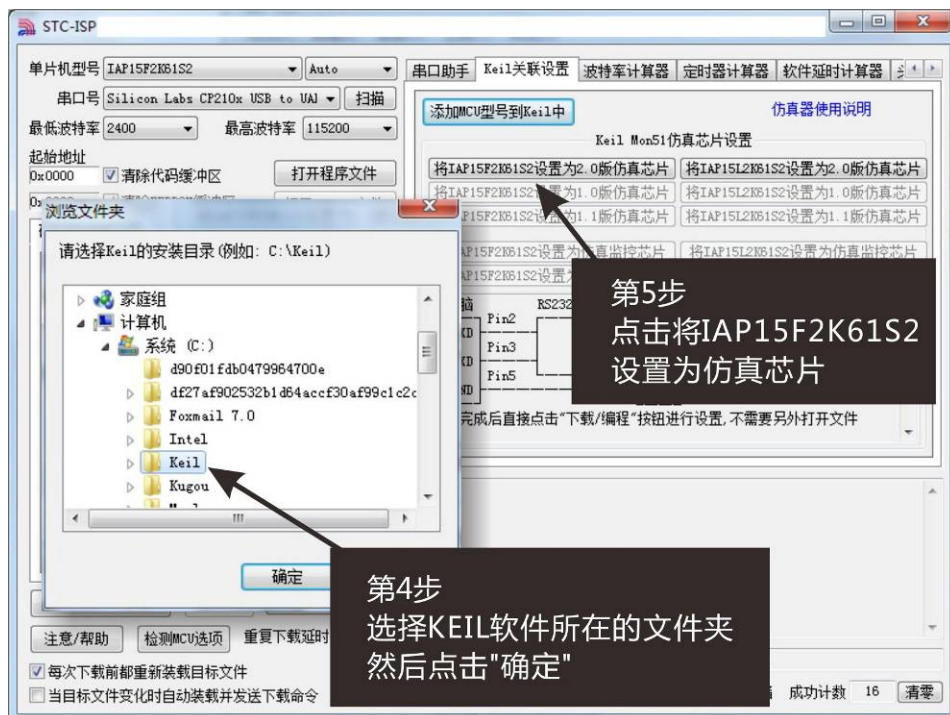
第 1 步：下载到 STC-ISP V6.36 软件，目前只有 V6.36 及以上版本支持仿真功能。



第 2 步：点击软件右侧的“KEIL 关联设置”选项卡。仿真相关的操作都在这个选项卡里完成。

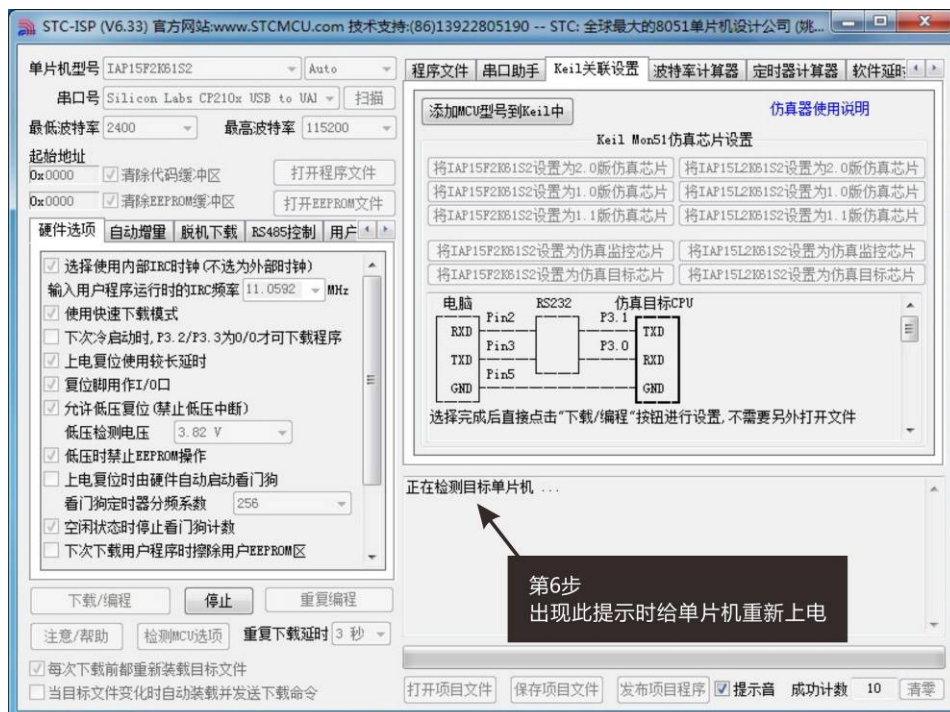
第 3 步：接着点击“添加 MCU 型号到 Keil 中”。这个操作效果是把 STC 芯片的仿真程序与 KEIL 软件绑定在一起，这样 KEIL 软件中的仿真功能才能操作 STC 单片机硬件。



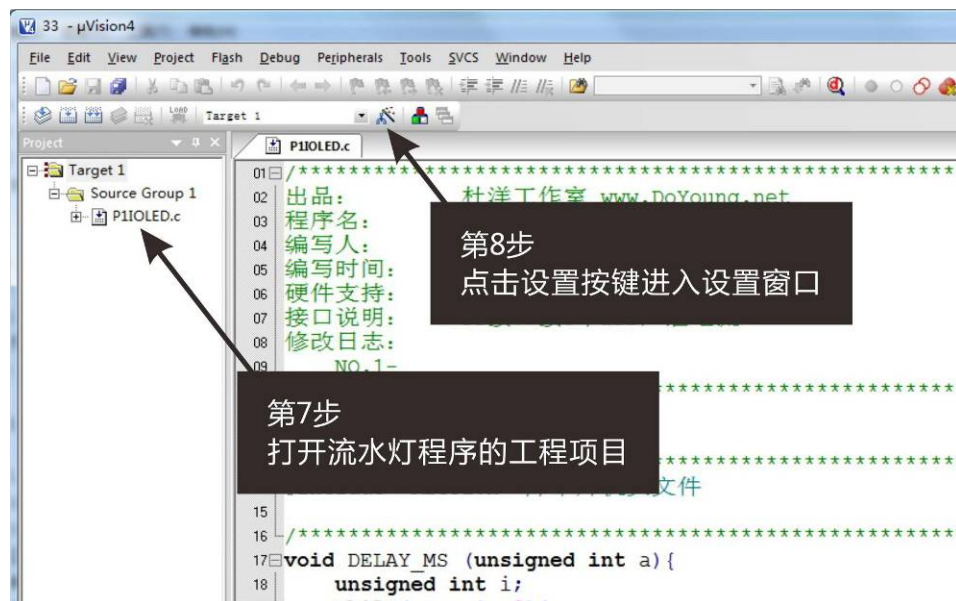


第 4 步：在弹出的“浏览文件夹”窗口中找到 KEIL 软件的安装目录，并点击“确定”。（默认是在 C:\Keil 中）

第 5 步：点击“将 IAP15F2K61S2 设置为 2.0 版仿真芯片”，在此处可以仿真的芯片有两款：IAP15F2K61S2 和 IAP15L2K61S2，前一款是 F 表示 5V 电源电压的芯片，后一款是 L 表示 3.3V 电源电压的芯片。我们以 5V 芯片为例。

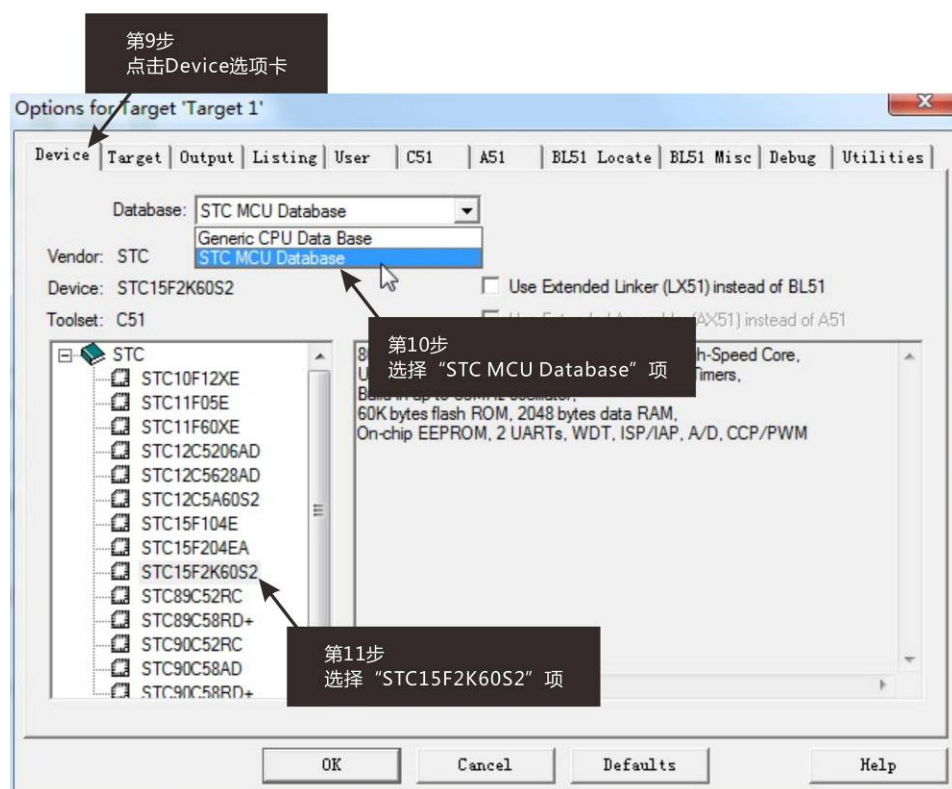


第 6 步：按下第 5 步的按钮后，按键变灰，下方状态窗口出现“正在检测目标单片机...”。这个提示的意思是您需要给单片机重新上电了，和之前给单片机写入 HEX 文件的方法相同。此时在硬件上给单片机冷启动，即会出现下载程序的提示，最后显示下载完成。大家可能不明白了，不是要仿真吗，为什么还要下载程序呢？其实这次下载的是仿真所需要的仿真处理程序，而不是我们要运行的 HEX 程序文件。仿真处理程序的功能是接收 KEIL 软件通过串口发出的仿真指令，再用这个指令去操控单片机寄存器和 IO 接口什么的。由此可见仿真处理程序是必不可少的哦。



第 7 步：打开 KEIL，打开你想要仿真的项目，我打开一个 P1 接口流水灯的程序，一会我们也以此为例详细讲解。

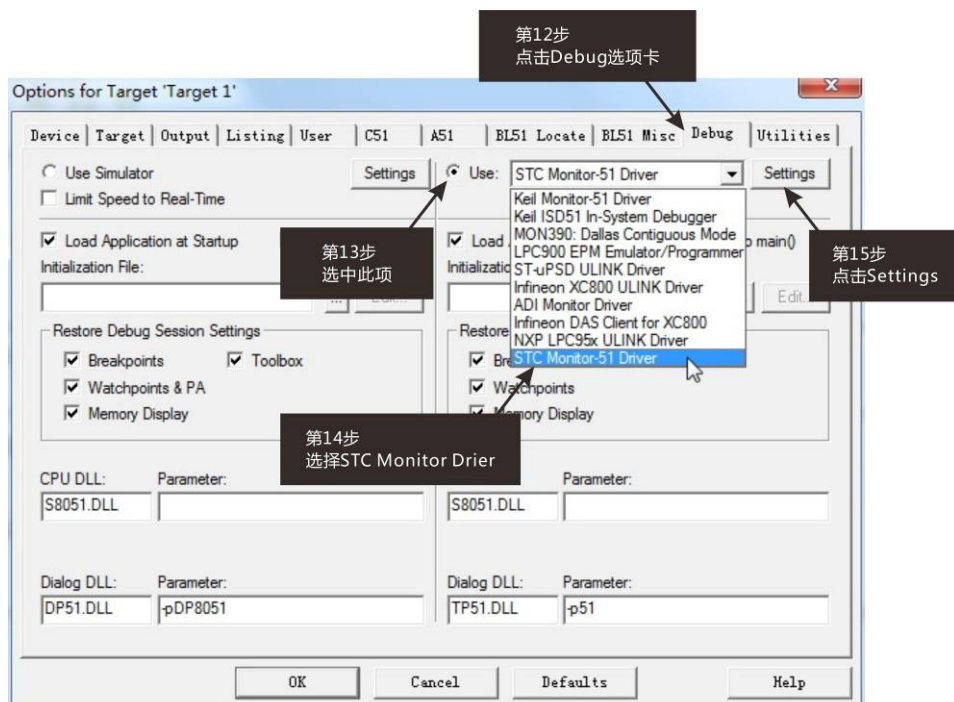
第 8 步：点击 Target Options 按键，或在菜单栏中点击 Project→ Options for Target。



第 9 步：进入 Options 窗口后，点击 Device 选项卡。

第 10 步：在 Database 下拉列表中选择“STC MCU Database”项。选中后就会出现 STC 系列单片机的型号。

第 11 步：在左侧型号中选择“STC15F2K60S2”项。这里选的是系列型号，包括同系列的很多款单片机。

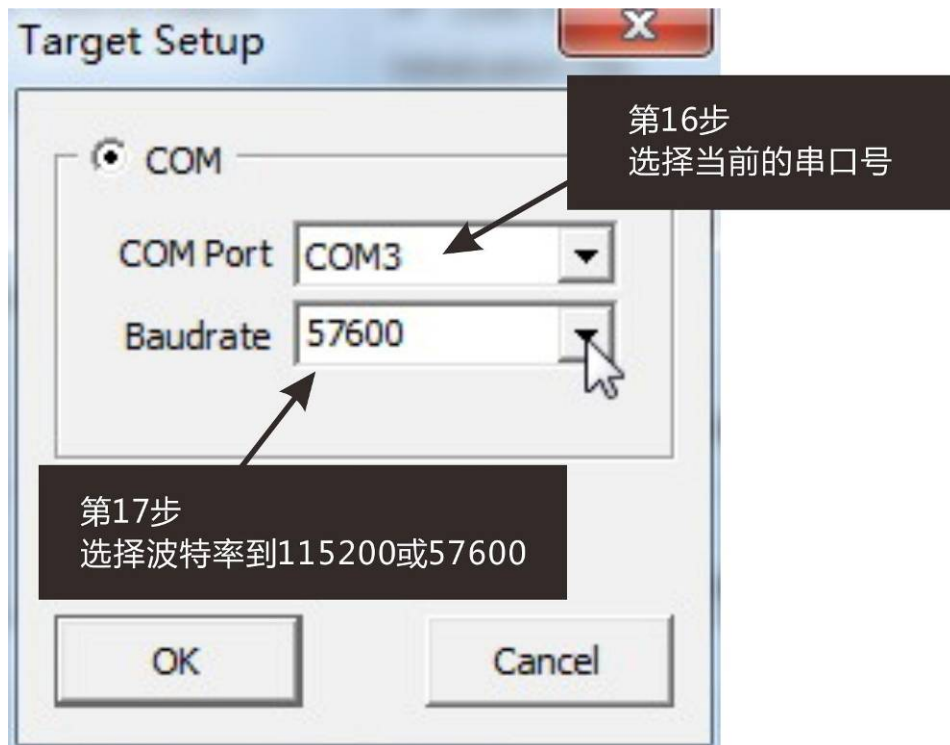


第 12 步：选择“Debug”选项卡。这里面都是与仿真相关的设置。

第 13 步：选中窗口右上方的项目。

第 14 步：在下拉列表中选择“STC Monitor-51 Driver”项。

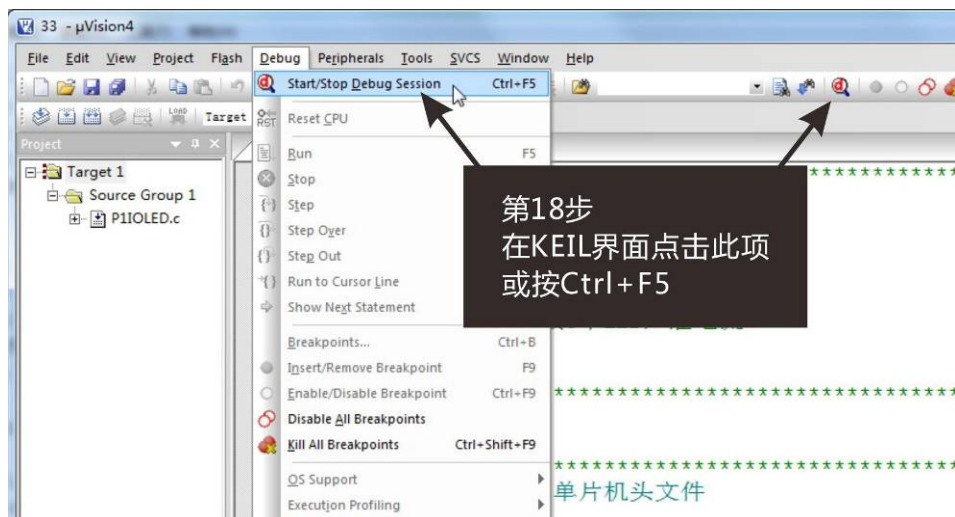
第 15 步：选择好后，点击右侧的“Settings”按钮。



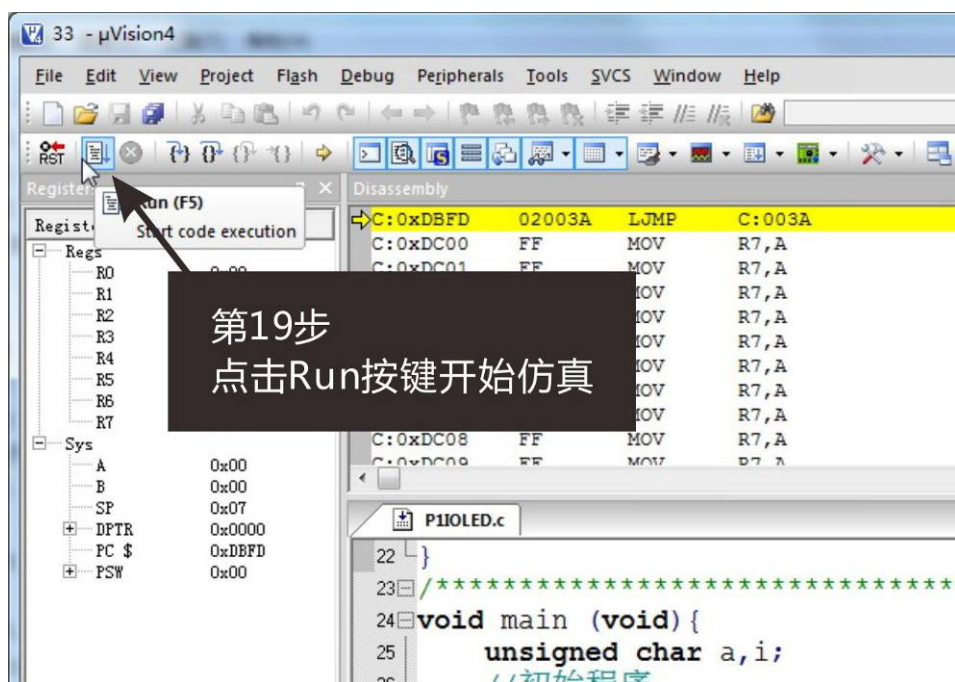
第 16 步：在弹出的窗口中可以设置仿真用串口通讯的串口号和波特率。串口号就选择单片机正在使用的串口。

第 17 步：在波特率下拉列表中选择 **115200** 或 **57600**，这个部分涉及到仿真的稳定性。所以要根据你的经验来设置。如果你是第一次使用，可以多设置几个值看看，哪一个最稳定就用哪个。如果仿真时出现错误提示也可能与此有关。



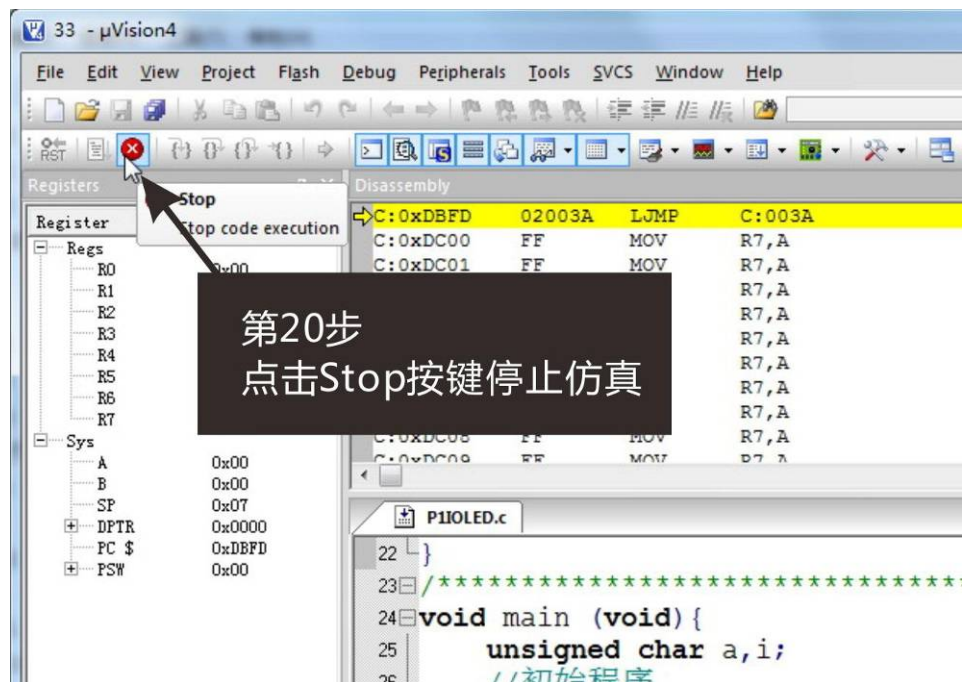


第 18 步：设置完成后回到主界面。点击 **Debug→Start/Stop Debug Session** 或按键盘上的 **Ctrl+F5** 开始仿真。这个操作是开始或停止仿真的切换按钮。如果我们之前的设置都是正确的，单片机硬件也接通了电源，这时 **KEIL** 软件会切换到仿真界面。



第 19 步：点击仿真界面下的“**Run**”按钮或按键盘上的 **F5** 键就能全速运行程序了。如果一切正常，你将会在单片机硬件电路上看到 **8 个 LED** 顺序点亮，呈现流水灯效果。想复位单片机，可以按左边的“**RST**”按钮。





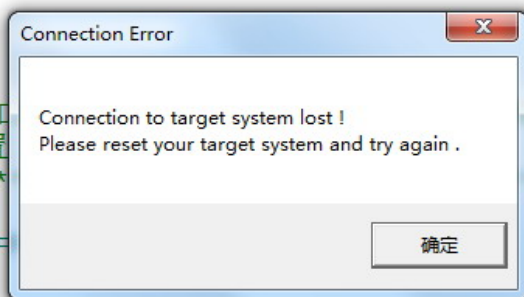
第 20 步：在全速仿真运行的状态下，点击“Stop”键停止仿真。

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
数

中断，允许串口  
PC串口端设置

\*\*\*\*\*

析（如不使用中  
串口的中断



定时器T/C1工作方式2

口工作方式1，允许串口接收（SCON = 0x40 时禁止串口接收）

初值高8位设置

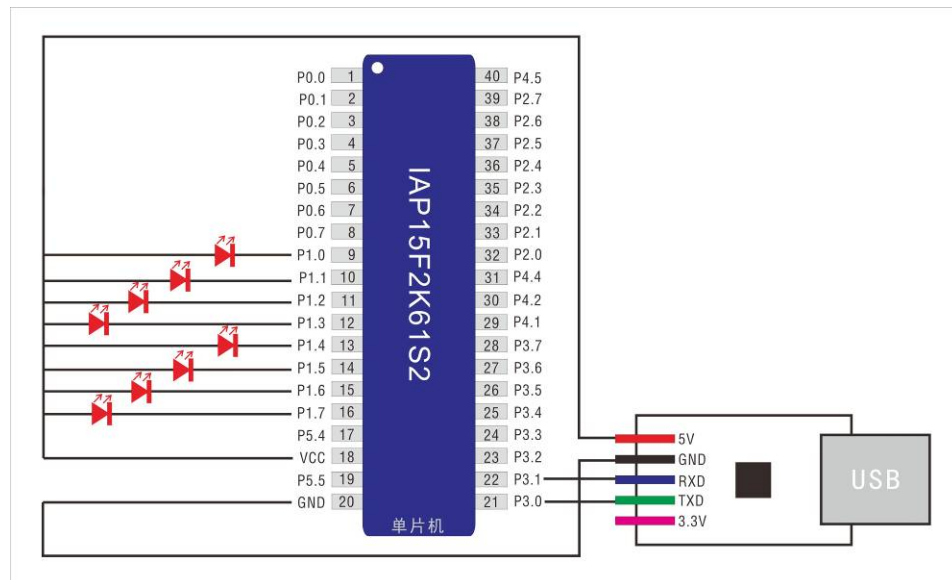
初值低8位设置

注意：如果在进入仿真界面或点击“Run”按钮后，出现如上图的提示窗口。则表示你之前的设置存在问题，或者是硬件电路的部分有异常。解决的办法是：首先把 KEIL 软件退回到正常编程状态，然后重新给单片机上电，再尝试进入仿真界面。如果还不行，则重新给单片机下载一次仿真处理程序。还不行的话就选择 KEIL 仿真设置里的其他波特率，再重复前面的尝试。最后实在没有办法的话就只能重启电脑试试了。

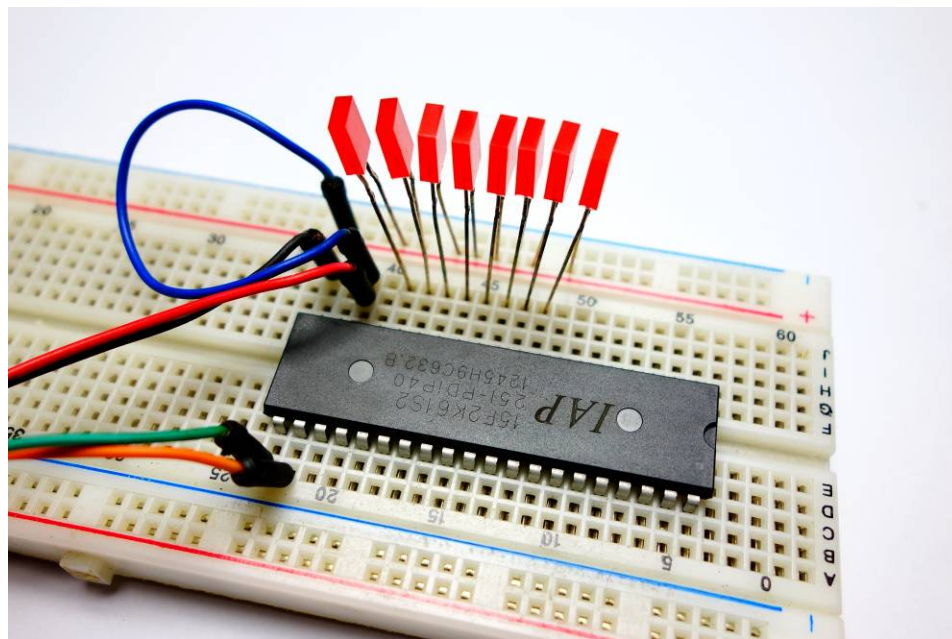
## 【流水灯程序仿真实例】

以上是仿真环境的建立和基本的仿真程序运行方法，下面我们就以流水灯的程序为例，讲一下仿真的过程与技巧吧。首先要做的是在单片机的 P1 接口上接 8 个 LED 灯，流水灯程序运行起来的时候，8 个 LED 灯会按顺序亮起。因为 LED

在单片机所产生的电流不大，所以可以不加限流电阻器。接下来就是加载流水灯的程序，这个部分大家当然可以自己来写，并不复杂。但为了保证仿真时不会因为程序的问题而导致错误，我还是写了一个标准的流水灯程序。建议大家第一次仿真时，还是用我给出的标准程序来使用，当你熟悉了仿真之后，再仿真自己的程序，这样能避免不少问题和麻烦。



流水灯程序的硬件电路图



电路在面包板上连接的照片

```
/*  
程序名：      P1 组接口流水灯  
编写人：      杜洋  
编写时间：    2013 年 4 月 11 日  
硬件支持：    STC 单片机  
接口说明：    P1 接口接 8 个 LED，灌电流  
*/
```

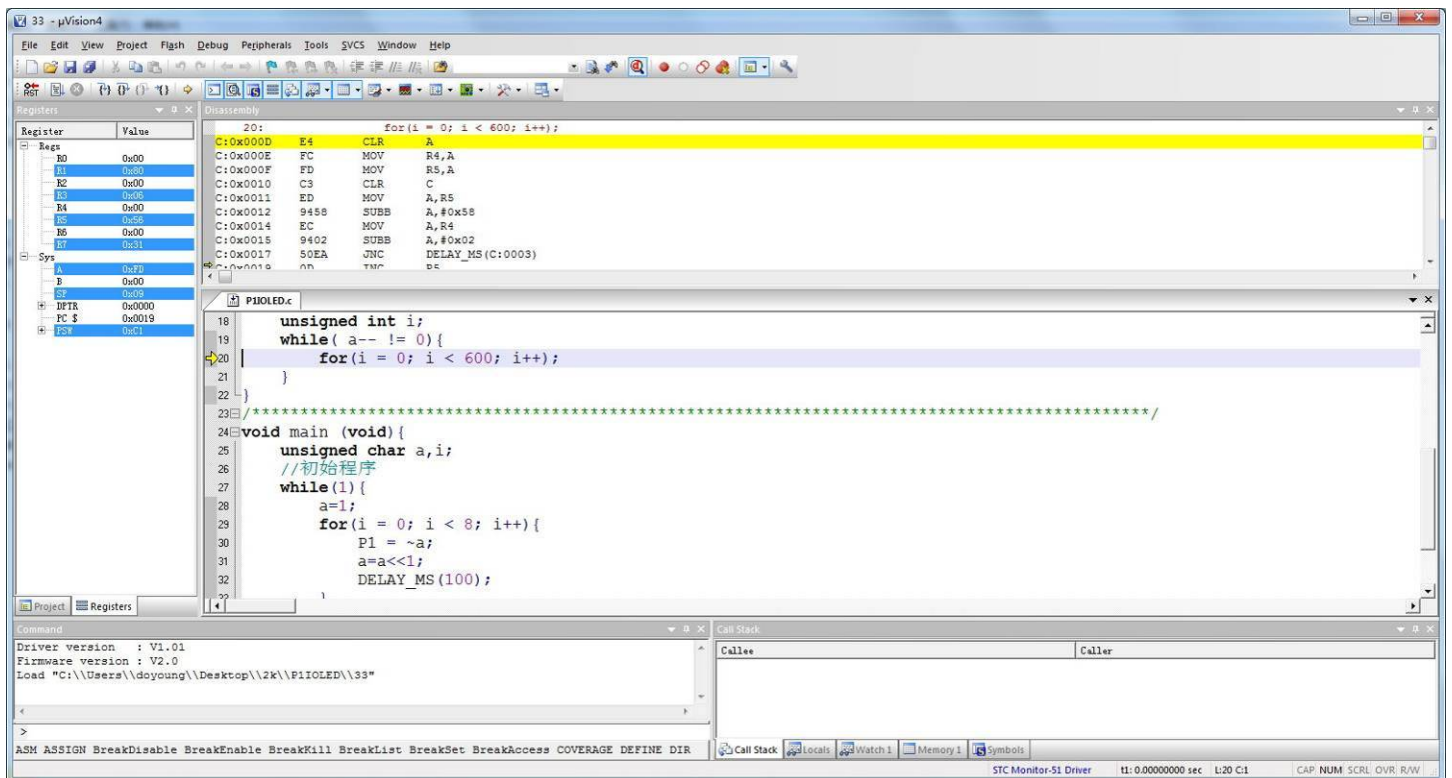
```
#include <REG51.h> //单片机头文件

void DELAY_MS (unsigned int a){//延时程序
    unsigned int i;
    while( a-- != 0){
        for(i = 0; i < 600; i++);
    }
}

/*****/
void main (void){
    unsigned char a,i;
    while(1){
        a=1;//初始值
        for(i = 0; i < 8; i++){//循环 8 次
            P1 = ~a;//将 a 的值取反送入 P1 接口
            a=a<<1;//a 值左移
            DELAY_MS(100);//延时
        }
    }
}

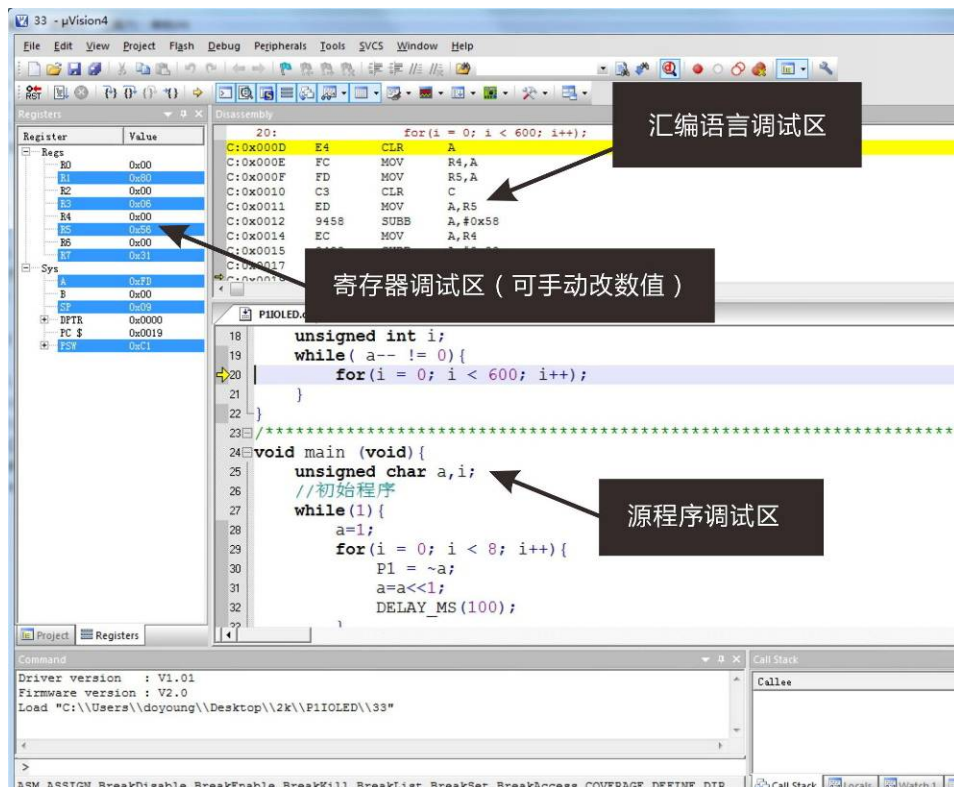
/*****/
```

### 流水灯源程序



### 仿真调试界面介绍



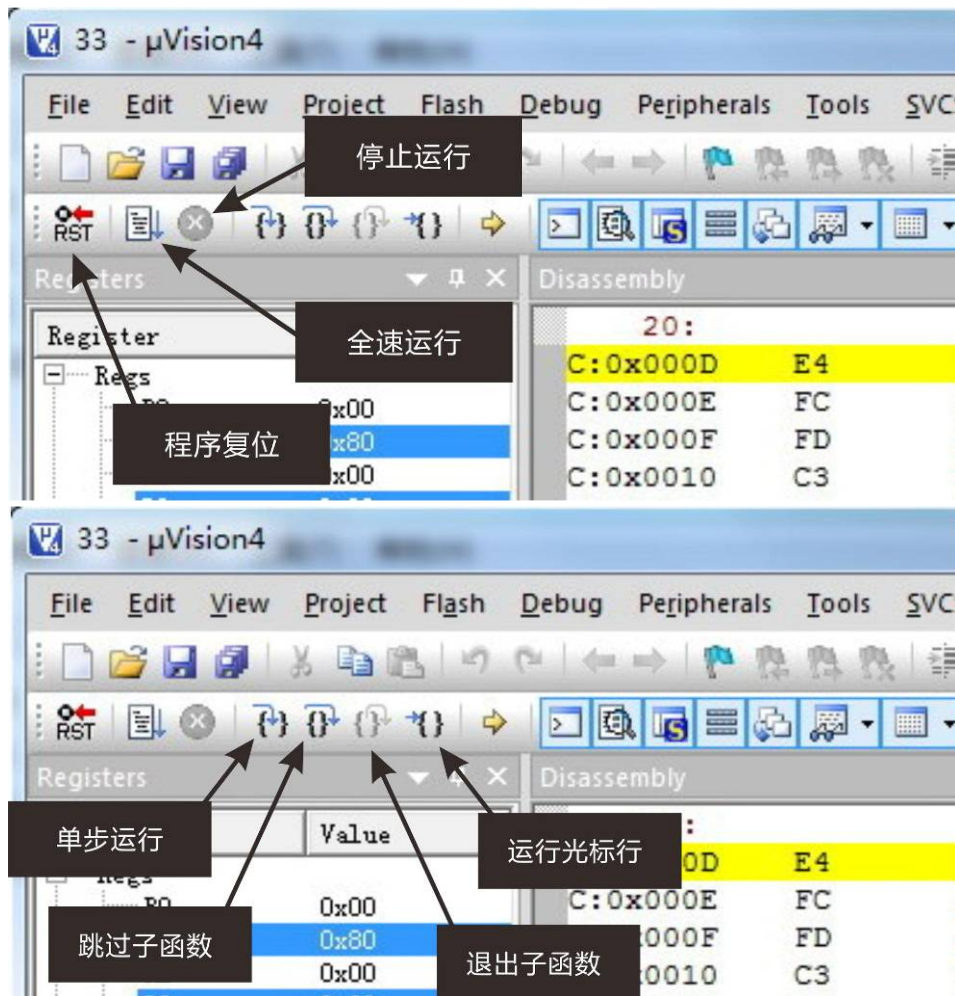


## 仿真调试窗口介绍

打开流水灯的工程文件，进入仿真调试界面。这时可以看到界面中包含了很多大小不同的窗口，它们都是干什么的呢？这里我们只介绍最重要的 3 个窗口的应用吧。在界面右侧上方较大的窗口是汇编语言调试窗口，里面既有 C 语言（我们写的源程序），还有软件自动编译出的汇编语句。要知道，单片机是不能直接读懂 C 语言的，这对它来说实在太难了。所以 KEIL 软件要先把 C 语言转成汇编语言，显示在汇编语言调试窗口上。在仿真的过程中，软件真正执行的是这些汇编的语言。而如果窗口中都是汇编的语言，我们调试人员又很难看出这些汇编语句与 C 语言源程序的对应关系。于是软件在这个窗口中先显示一行 C 语言，再在其下面显示这行 C 语言所转换成的汇编语言。在汇编语句的左侧有一个黄色的小箭头，这个被称为“程序运行指针”。它所指向的汇编程序行就是仿真软件正在执行仿真的那一行。大家从此可以看出每一行 C 语句都会转换出至少 3 行汇编语句。也就是说，要执行 3 步以上的汇编语句才能完成 1 行 C 语言的指令。了解这一点是非常重要的。

在汇编语言调试窗口的下面是 C 语言源程序调试窗口，我们的源程序就在此处供你参考。在窗口的左侧也有一个小黄箭头，与汇编语言调试窗口里的黄箭头对应。双击源程序中希望仿真的那行，就能让黄箭头跳到对应的位置，再点击单步运行，就会从这一行开始运行。值得注意的是，看到这个窗口的时候，大家一定会以为在调试的过程中只要改动源程序调试窗口里的内容，再点击运行，就会执行新的程序，但事实并不是这样的。源程序调试窗口只是给我们调试时考虑的，在这里修改是不能被更新仿真的。所以我们还需要退出仿真界面，在 KEIL 的编译界面下修改，再进入仿真界面仿真。虽然有点麻烦，但相比于把程序用 ISP 下载到单片机里调试可要方便得多。

在界面最左边的一个纵向的矩形窗口是寄存器调试窗口，这个部分显示的是单片机内部与运算有关的寄存器。包括 R0 到 R7，PSW，加法器 a 等。在仿真调试的过程中，这些数据的值会随着程序的运行而改变。你可以观察这些数值是否如你意，同时还可以修改数值到你希望的状态，再运行程序，程序会按你设置的数值运行。非常方便。



仿真调试相关按钮

好了，现在就来仿真流水灯的程序看看吧。在仿真界面的左上角有一排按钮，它们是控制仿真的操作按钮。它们从左到右依次是“复位”“全速运行”“停止”“单步运行”“跳过子函数”“退出子函数”和“运行光标行”。这些按钮就好像 DVD 播放机上的操作按键，可以正常播放、快放、慢放、单帧步进。仿真按钮就是 DVD 机的遥控器，你完全可以这样理解。“复位”按钮相当于 DVD 机上的归零键，即把播放的时间设置到 00:00，也就是从头播放。按下后，仿真软件会把单片机复位，随后的运行就是从程序最初状态开始的。“全速运行”相当于 DVD 机上的播放键，按下后程序会以正常的速度仿真，其效果和把 HEX 文件下载到单片机上运行的效果相同。“停止”按钮相当于 DVD 机上的停止键，无论程序运行到什么地方都要停下来。因为只有停止运行才能设置参数、修改源程序。“单步运行”就是 DVD 机上的单帧步进按键，可以一帧一帧地看画面，从而发现容易闪过的细节。点击“单步运行”，仿真软件只走一步，仅运行一行汇编程序。在你想知道程序在某处时到底发生了什么，单步运行是最好的查看方式。但假如程序很长很长，中间又有很多子程序，要走到想调试的某处并不容易。这时就要用到下面 3 个按钮。“跳过子函数”就是在当前的程序中不进入任何子函数的单步运行。例如现在调试流水灯程序，我们只希望看到灯流动时的几个关键点状态，这样自然不需要进入延时函数。点击“跳过子函数”就不会进入延时函数，只在改变 IO 接口状态的主函数部分运行。要是不小心进入了延时函数怎么办？点击“退出子函数”按钮可以跳出当前的函数，回到上一级函数。如果当前在主函数里的话，这个按钮是灰的（不能点击）。如果你还是感觉不方便，那“运行光标行”按钮一定适合你。点击它可以跳到汇编语言光标所在的那一行，不论那一行离当前运行的行有多远，都会跳过去运行。与前几个按钮相比，“运行光标行”看上去像超人一样可以飞到任何地方，可是它并不会运行中间的程序部分，如果中间程序中有一些必须被运行的部分，那“运行光标行”将会错过这些程序而使寄存器的数值混乱。就好比你看一部电影，刚看完前 3 分钟，你就突然跳到 40 分钟处，而中间 37 分钟的剧情你全然不知。电影是线性的，单片机程序也是。所以在你没有把握的情况下尽量不要用“运行光标行”。仿真功能给调试带来许多方便，但也要谨慎使用仿真按钮，不要让程序失灵才是。

在流水灯的例子中，程序里的函数关系相对简单。只要一直点击“跳过子函数”按钮就可以避开延时函数的单步运行。每点击一次，你看发现在汇编调试窗口中的黄指针向下移动了一行，而每移动 3 行（某些程序行数更多）才完成 1 行 C 语句。在运行到“**P1 = ~a;**”这行时，你会看到在硬件电路中，第一个 LED 灯点亮了。然后跳过了延时程序，当再一次来到“**P1 = ~a;**”这行时，第二个 LED 灯点亮，第一个 LED 灯熄灭。在这个过程中，你能发现寄存器调试窗口中的数值在不断变化，那些值就是单片机真实运行时的参数。如果你足够认真，你会发现我们在 C 语言文件中定义的 2 个变量 a 和 i 是和寄存器 R0~R7 中的某 2 个对应的。也就是说，我们定义的变化，其实就是单片机寄存器上真实的寄存单元。如果你找到它们，并修改这些值，没错，接下来的运行效果就不同了。在不断点击下去，LED 灯继续变化，直到一个循环结束和另一次循环的开始。熟悉了单步运行之后，再点击“全速运行”看看，那将是一个与真实无异的单片机运行状态，LED 以很快的速度流动着，好像一部正常速度播放的电影。

接下来，请把你之前学过、用过的程序，都用仿真的方法运行一次吧。你一定会遇到问题，不过没有关系，在不能运行或死机时，重新开始就行了。仿真不比真实，会有一些不确定的因素出现。把握它们，解决它们。