

弟子教你学51单片机技术



# 从零到项目开发学51单片机 与C语言技术

基于DZ80C52开发板

QQ:2539606882

从零到项目开发学51单片机与C语言技术视频教程

弟子教你学51单片机技术



# 第五讲 单片机仿真--如何才算 学会了51单片机

基于DZ80C52开发板

QQ:2539606882

从零到项目开发学51单片机与C语言技术视频教程



# 单片机仿真

## 单片机仿真的概念和方法

- 一、什么是单片机仿真，为什么要仿真？
- 二、如何进行软件仿真？
- 三、硬件仿真和软件仿真有什么区别，如何进行硬件仿真？



# 如何才算学会了51单片机

- 只要八个步骤就可以入门单片机
- 学习使用单片机就是理解单片机硬件结构，在汇编或C语言中学会各种功能的初始化设置，以及实现各种功能的程序编制，我们还可以利用单片机仿真学习板在电脑上做实验，也可以用硬件电路的单片机学习板来做实验。



## 第一步：数字I/O的使用

- 使用按钮输入信号，发光二极管显示输出电平，就可以学习引脚的数字I/O功能，在按下某个按钮后，某发光二极管发亮，这就是数字电路中组合逻辑的功能，虽然很简单，但是可以学习一般的单片机编程思想，例如，必须设置很多寄存器对引脚进行初始化处理，才能使引脚具备有数字输入和输出功能。每使用单片机的一个功能，就要对控制该功能的寄存器进行设置，这就是单片机编程的特点，千万不要怕麻烦，所有的单片机都是这样。



## 第二步：定时器的使用

- 学会定时器的使用，就可以用单片机实现时序电路，时序电路的功能是强大的，在工业、家用电气设备的控制中有很多应用，例如，可以用单片机实现一个具有一个按钮的楼道灯开关，该开关在按钮按下一次后，灯亮3分钟后自动灭，当按钮连续按下两次后，灯常亮不灭，当按钮按下时间超过2s，则灯灭。数字集成电路可以实现时序电路，可编程逻辑器件（PLD）可以实现时序电路，可编程控制器（PLC）也可以实现时序电路，但是只有单片机实现起来最简单，成本最低。定时器的使用是非常重要的，逻辑加时间控制是单片机使用的基础。





## 第三步：中断

- 单片机的特点是一段程序反复执行，程序中的每个指令的执行都需要一定的执行时间，如果程序没有执行到某指令，则该指令的动作就不会发生，这样就会耽误很多快速发生的事情，例如，按钮按下时的下降沿。要使单片机在程序正常运行过程中，对快速动作做出反应，就必须使用单片机的中断功能，该功能就是在快速动作发生后，单片机中断正常运行的程序，处理快速发生的动作，处理完成后，在返回执行正常的程序。中断功能使用中的困难是需要精确地知道什么时候不允许中断发生（屏蔽中断）、什么时候允许中断发生（开中断），需要设置哪些寄存器才能使某种中断起作用，中断开始时，程序应该干什么，中断完成后，程序应该干什么等等。



## 第三步：中断

- 中断学会后，就可以编制更复杂结构的程序，这样的程序可以干着一件事，监视着一件事，一旦监视的事情发生，就中断正在干的事情，处理监视的事情，当然也可以监视多个事情，形象的比喻，中断功能使单片机具有吃着碗里的，看着锅里的功能。以上三步学会，就相当于降龙十八掌武功，会了三掌了，可以勉强护身。





## 第四步：与PC机进行RS232通信

- 单片机都有USART接口，特别是MSP430系列中很多型号，都具有两个USART接口。USART接口不能直接与PC机的RS232接口连接，它们之间的逻辑电平不同，需要使用一个MAX3232芯片进行电平转换。USART接口的使用是非常重要的，通过该接口，可以使单片机与PC机之间交换信息，虽然RS232通信并不先进，但是对于接口的学习是非常重要的。正确使用USART接口，需要学习通信协议，PC机的RS232接口编程等等知识。试想，单片机实验板上的数据显示在PC机监视器上，而PC机的键盘信号可以在单片机实验板上得到显示，将是多么有意思的事情啊！



## 第五步：学会A/D转换

- STC 1T单片机带有多通道10位A/D转换器，通过这些A/D转换器可以使单片机操作模拟量，显示和检测电压、电流等信号。学习时注意模拟地与数字地、参考电压、采样时间，转换速率，转换误差等概念。使用A/D转换功能的简单的例子是设计一个电压表。

# 第六步：学会PCI、I2C接口和液晶显示器接口

- 这些接口的使用可以使单片机更容易连接外部设备，在扩展单片机功能方面非常重要。



## 第七步：学会比较、捕捉、PWM功能

- 这些功能可以使单片机能够控制电机，检测转速信号，实现电机调速器等控制起功能。如果以上七步都学会，就可以设计一般的应用系统，相当于学会十招降龙十八掌，可以出手攻击了。



第八步：学习USB接口、TCP/IP接口、各种工业总线的硬件与软件设计

- 学习USB接口、TCP/IP接口、各种工业总线的硬件与软件设计是非常重要的，因为这是当前产品开发的发展方向。到此为止，相当于学会**15招降龙十八掌**，但还不到打遍天下无敌手的境界。即使如此，也算是单片机大虾了。顺便说一句，**c51**单片机的技术支持是非常好的，负责责任的工程师大虾会在关键时候，挺身而出，使你化险为夷。



# 有没有信心好学单片机呀

- 上面8步，各位看完了，想不想学单片机呢？如果想学，来下载一个DZ-80C52仿真电路学习板吧！不需要花钱就可以学单片机的，呵呵。如果还想动动手，也可以来购一个DZ-80C51或者DZ-80C52硬件学习板。
- 下面开始来讲解单片机仿真





## 关于编程器

- 建议你使用具有**ISP**功能的单片机完全是考虑到初学者的资金问题，因为很多人一听说学习要花钱就会放弃学习。世界上没有免费的知识，更没有免费的技能，投资是非常必要的，但是成本越低就越容易上手，所以推荐具有**ISP**功能的单片机，但这并非说编程器与仿真器已经没用，这一点你们要明白。



## 关于编程器

- 首先说说编程器，它是干什么用的呢？程序编辑好后，并且通过编译生成了HEX格式的二进制烧写文件，就要用编程器把它烧写到单片机里面。我最初入门的时候就是使用AT89C51 +编程器的开发模式，过了半年多才在发现具有ISP功能的单片机，白花了那么多银子。编程器本身很贵的，最好的上千，好一点的数百元，最便宜的简易型的也要上百元，是个不小的投资，很多人之所以对单片机望而却步，正是在这个投资上出现了犹豫。另外编程器使用起来也比较麻烦，每一次编程都要把芯片从实验板上拔下来，烧写好了再插回去，很容易弄坏插座和管脚，为此配置零插拔力插座的话，又要花一只单片机的钱了（十多元一只）。



## 关于编程器

- 而且如果去现场工作的话，拎着个编程器多麻烦！设备复杂的话插拔芯片也很困难，一旦弄坏了管脚或者印刷板，设备就完蛋了。但是编程器在开发产品中具有重要作用，因为产品的目标板（相当于完成指定功能的实验板）一般没有空间额外加挂串口电平转换部分的器件，所以要通过编程器编程。而且开发产品如果使用一般的51芯片，成本也会大幅度下降。对于那些不用改动程序的产品比较有利。当然如果产量非常大的话，就选择掩膜（就是把程序直接做到芯片里）生产了，编程器也不用了，成本非常低。



## 仿真器

- 仿真器在高校教学、实验与复杂产品开发中可以使用。使用编程器和ISP的开发方法叫做“崩溃——烧写”模式，如果程序运行结果错误，只能改动程序后擦除芯片重新烧写，为了调试一个程序连续烧写芯片数十次是很正常的事情，但是这样不但麻烦，还会缩短芯片使用寿命，而且无法观察程序运行中的状态。仿真器正是针对这一问题，它无需烧写，通过它可以直接观察运行结果，而且可以单步运行，能够最直接的找到问题所在，所以高级产品的开发都要用到。不过初学者一般买不起仿真器，也没必要买，KEIL本身就有软件仿真功能，可以大大减少烧写次数。只要不涉及到外部扩展接口，在KEIL上仿真运行成功的程序，绝大部分烧写以后都能正常运行，下文将有说明。



## 什么是单片机仿真，为什么要仿真？

你编写的程序（编译正确）或你设计的应用系统存在问题时候，怎么办？

实际单片机的两个特性使你的排错工作遇到巨大困难：1、内部不可见，单片机是一种典型的“黑箱子”。2、高速运行不停止。

一个天才的想法：用一个透明的、可随意停止的虚拟单片机来代替实际单片机。这个透明的、可随意停止的虚拟单片机，就是仿真器（ICE）。

为什么叫仿真器？仿：模仿；真：真实的单片机。和实际单片机的同异点：同：引脚数、引脚排列完全一致，同样可烧写程序，装到实际的单片机应用系统硬件电路中实际运行效果也完全一致。异：仿真器透明的、可随意停止；实际单片机不透明、不可随意停止。

以专业51仿真器为例，进行操作说明。

为什么要仿真？因为你编写的程序或你设计的应用系统存在问题。仿真器为用户应用系统软硬件排错提供了高效的技术手段。





## 软件仿真

仿真=调试 仿真器=调试器

单片机仿真可分成软件仿真和硬件仿真两种。

所谓软件仿真，是指用KEIL UV2自带的软件仿真器（一个工具软件）对单片机应用系统的软件部分进行调试。

软件仿真的优点：不需要开发板（目标板）、硬件仿真器等。

软件仿真的缺点：不能反映单片机应用系统的实际情况。

KEIL UV2自带的软件仿真器可以模拟标准的8051系统，但注意1T单片机有些新增模块用软件仿真可以仿，有些则仿不了，也不能准确测量1T单片机的运行时间！

参见《Keil 软件实验教程（2）》

结合实例，具体介绍如何进行软件仿真。





进入调试模式后，我们可以执行的操作主要有两类：

## 1、查看单片机内部资源：

我们主要了解外围接口和观察窗口就可以了。

使用观察窗口小技巧：在“type F2 to edit”处单击2次（注意不是双击！）即可输入变量名，用同样的方法可以修改变量的值。另外在语句的SFR或变量处用鼠标右键快捷菜单也可以快速添加观察对象。

## 2、程序运行的控制：

全速运行、停止、单步、单步跳过等

断点（break point）设置：在希望设置断点的源代码行双击，再双击取消。最多可设置3个断点。程序在执行完设置断点的代码行后停止。绿色箭头指向断点后下一条指令。



## 调试快捷工具条

从左到右分别是：

- 1、reset复位
- 2、go运行、全速运行。处理器会一直运行，直到遇到一个断点，或点击了工具栏上的暂停按钮暂停运行。
- 3、stop running停止
- 4、step单步：函数内部也单步运行
- 5、step over单步跳过：也称过程单步，函数内部代码被全速执行，整个函数被作为单步运行中的一步来执行的
- 6、step out current function跳出函数：当程序运行在函数中时，可以使用它直接运行完该函数。
- 7、run to cursor line运行到当前行

注意：

- 1、程序修改后，必须先停止调试、重新编译，然后再重新开始调试！
- 2、网上流传的UV2的汉化版有些术语翻译很不恰当，本教程使用爱晶电子推出的keil750中文修正版,对绝大多数不规范的术语翻译进行了修正。



## 硬件仿真

硬件仿真和软件仿真有什么区别？

软件仿真：在不存在实际的单片机应用系统硬件电路的情况下，我们可以用软件仿真器对单片机应用系统的软件部分进行调试。

硬件仿真：是用硬件仿真器或其他替代方法对一个实际的单片机应用系统的软件和硬件部分进行调试，它能真实地反映实际的单片机应用系统存在的问题。

注：有些同学将硬件仿真实解为对应用系统的硬件电路进行调试，这是错误的！



## 仿真器与仿真芯片

专业的51仿真器价格在几千元以上，甚至更高，对于初学者和一般的开发者来说，完全没有必要购置，另外目前市场上的51仿真器不能用于1T单片机的仿真。

前面我们讲过1T单片机是由传统的51单片机内核+大量新增模块组成的。如果你的程序没有用到新增模块，那么你可以用51仿真芯片——也就是Monitor-51来完成仿真。

51开发板可以和51仿真芯片配合通过USB口进行硬件仿真。

弟子教你学51单片机技术



## 51仿真器



从零到项目开发学51单片机与C语言技术视频教程





## 51仿真芯片

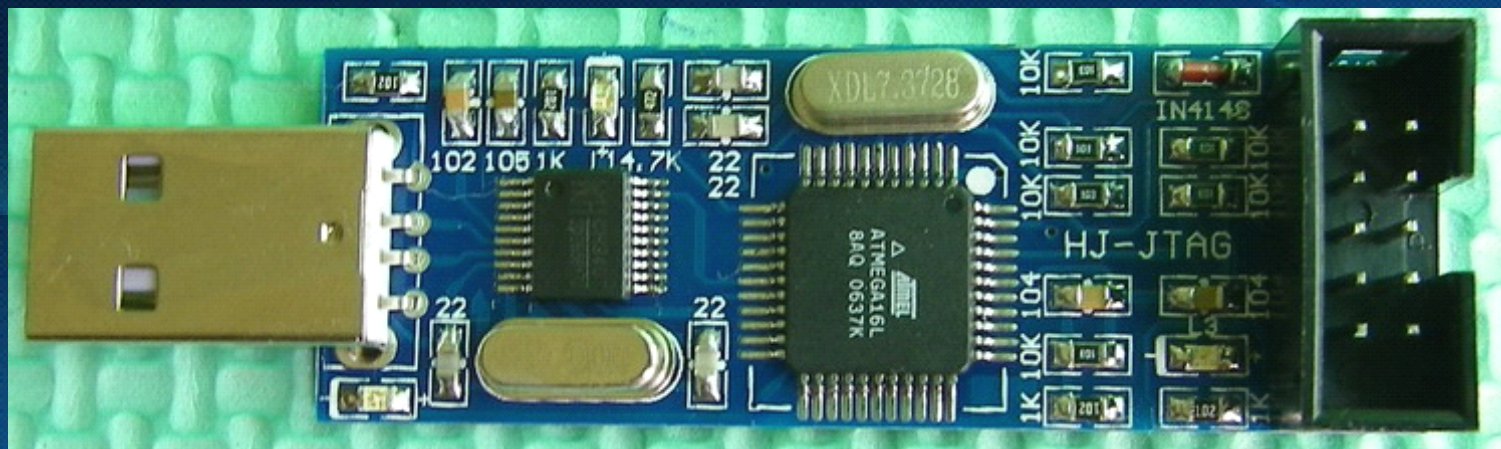
- SST89E561RD





## AVR仿真器

- JTAG 仿真器





# SST 仿真芯片的使用方法

- 硬件要求：1、已经固化仿真程序的仿真芯片，如 **SST89E58、SST89E516**
- 2、带有串口资源的目标板，如开发板
- 硬件连接：1、将仿真芯片直接插到目标板上，代替原有单片机的位置。
- 2、连接电脑的串口和目标板的串口端子，没有串口的电脑使用 **usb 转串口线**
- 3、连接好电源，打开电源开关
- 软件设置：
- 1、打开 **keil** 软件，并打开一个需要仿真的程序（已经编译通过的工程）如下图：
- 以下以“流水灯”工程为例。

# 弟子教你学51单片机技术

File Edit View Project Debug Flash Peripherals Tools SVCS Window Help

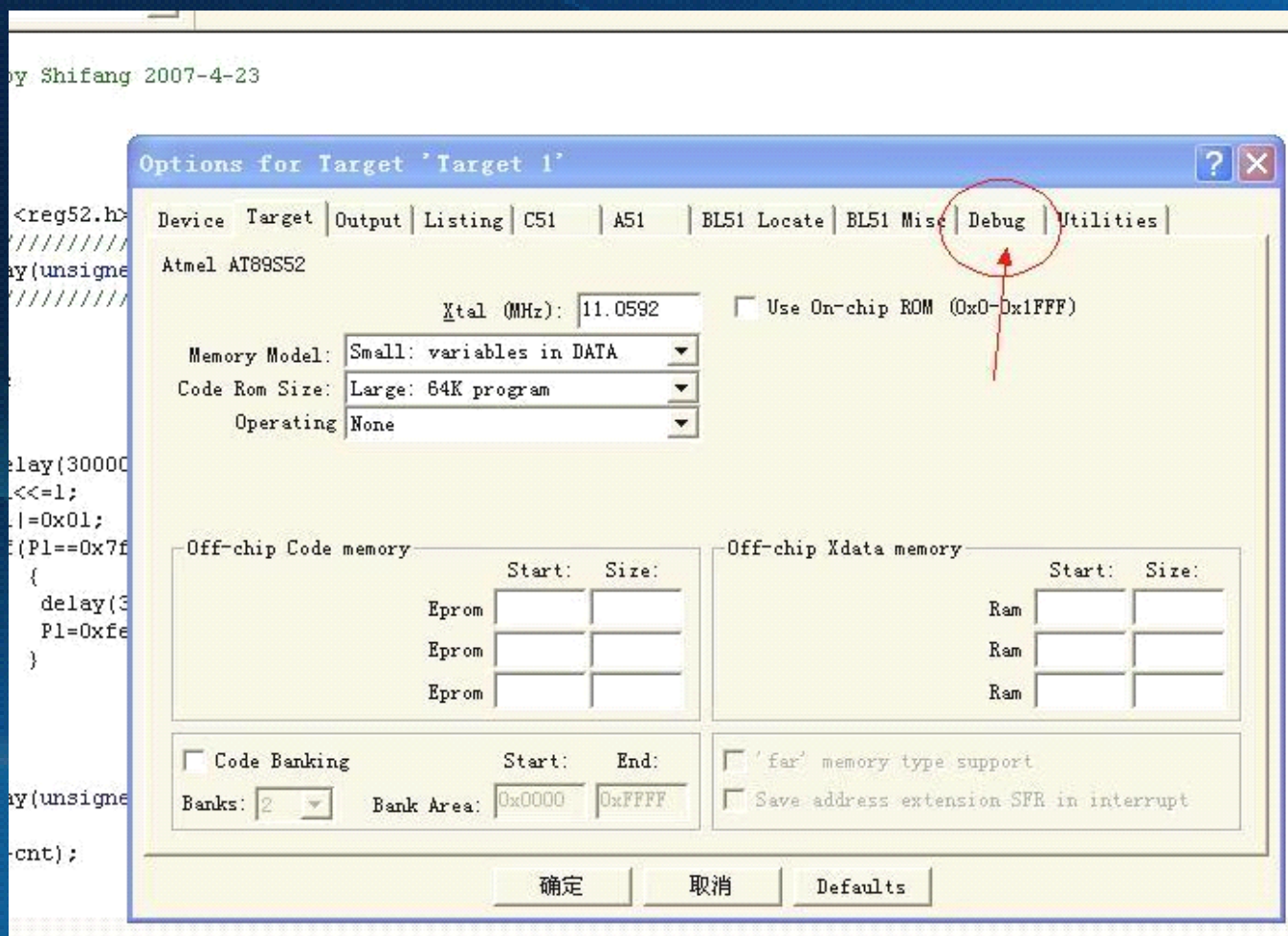
Target 1

```
*  
*write by Shifang 2007-4-23  
*  
*V1.1  
*/  
  
#include <reg52.h>  
////////////////////////////////////  
void delay(unsigned int cnt);  
////////////////////////////////////  
main()  
{  
    P1=0xfe;           //给初始化值  
    while(1)  
    {  
        delay(30000); //延时程序  
        P1<<=1;       //左移一位  
        P1|=0x01;     //最后一位补1  
        if(P1==0x7f) //检测是否移到最左端?  
        {  
            delay(30000); //delay
```

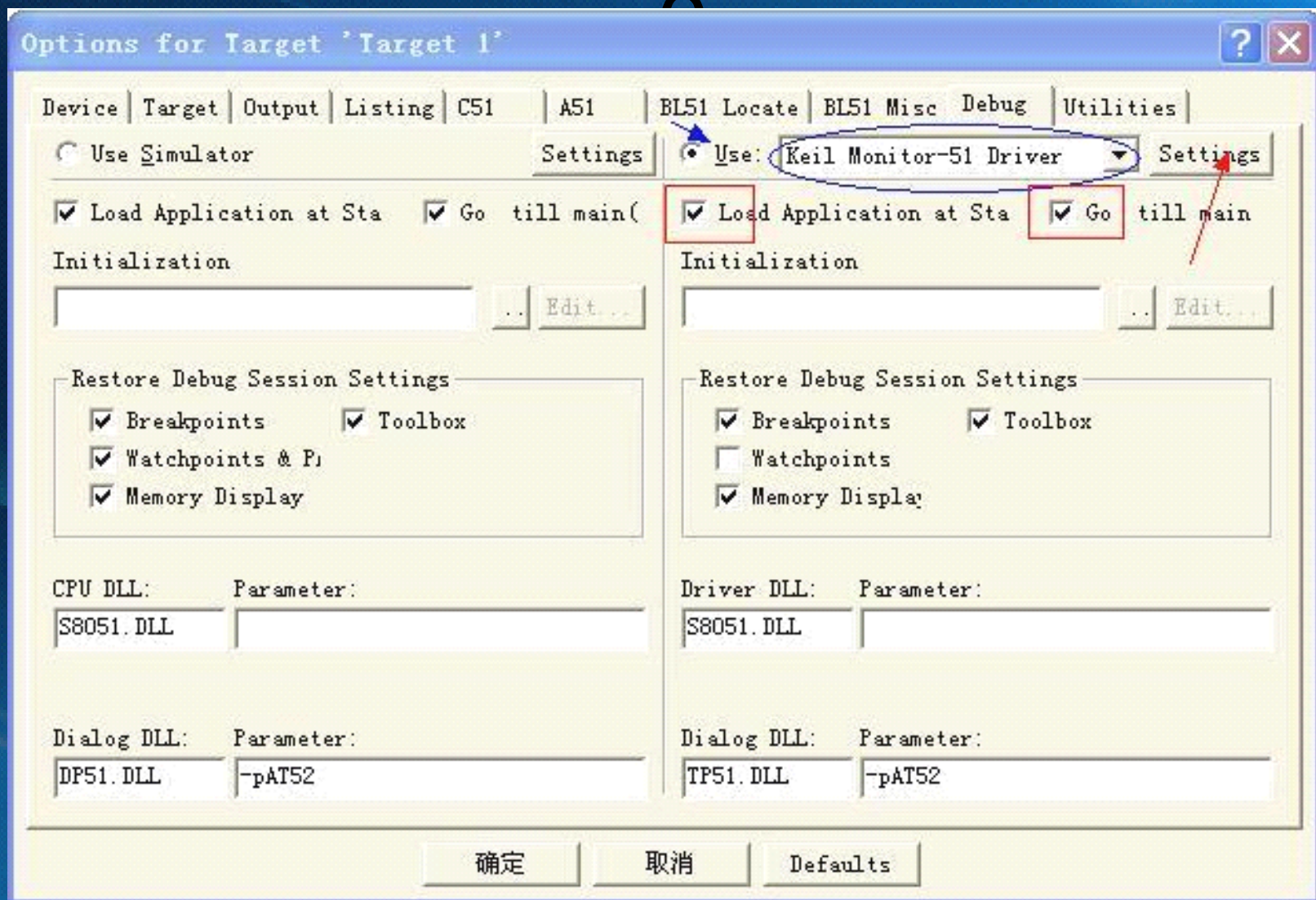
从零到项目开发学51单片机与C语言技术视频教程



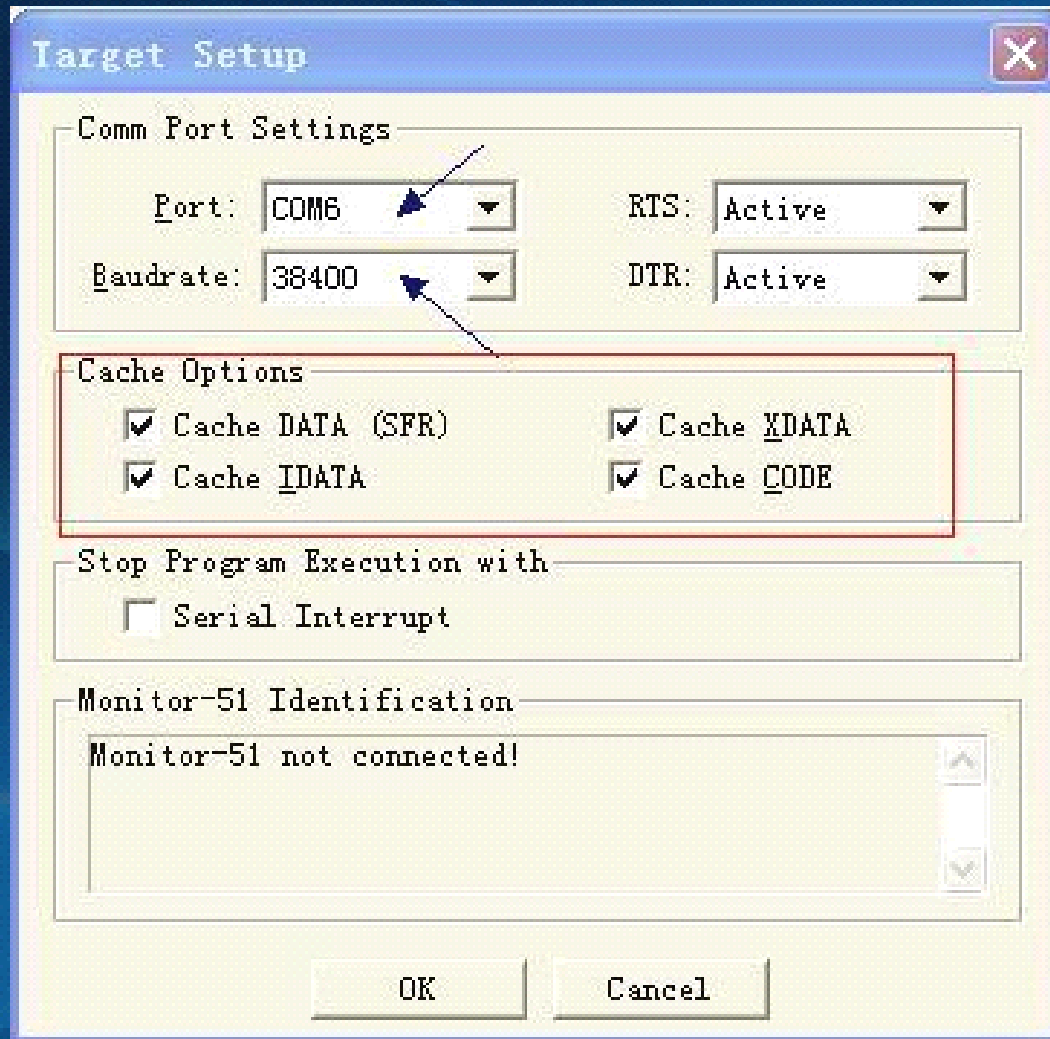
# 点击上图箭头图标打开如下对话框







如上图设置后,点击“Setting”选项,  
出现如下对话框





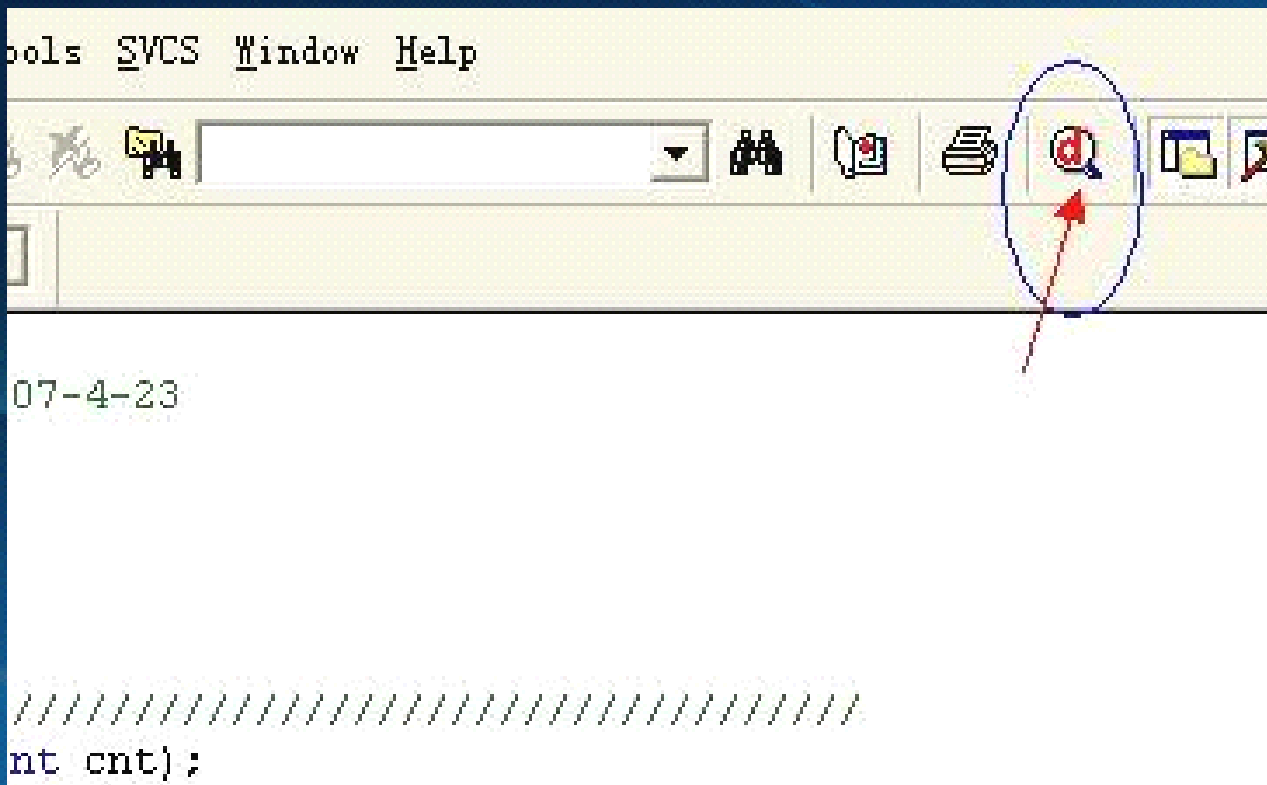


## 波特率默认选择

- ★Com 口选在当前硬件连接的端口，不能选择错误，否则不能通讯
- ★ 波特率默认选择 38400，也可以尝试选其他波特率，速度较低
- 然后“OK”“确认”完成设置



# 点击如下图仿真“ ”图标





# 不能进入仿真

CONNECTION TO TARGET SYSTEM LOST!

Try Again

Please reset your target system and select 'Try Again' to resynchronize.

Settings ...

Opens a dialog where you can select the monitor configuration, serial interface parameters and more...

Stop Debugging

Stop debugging session



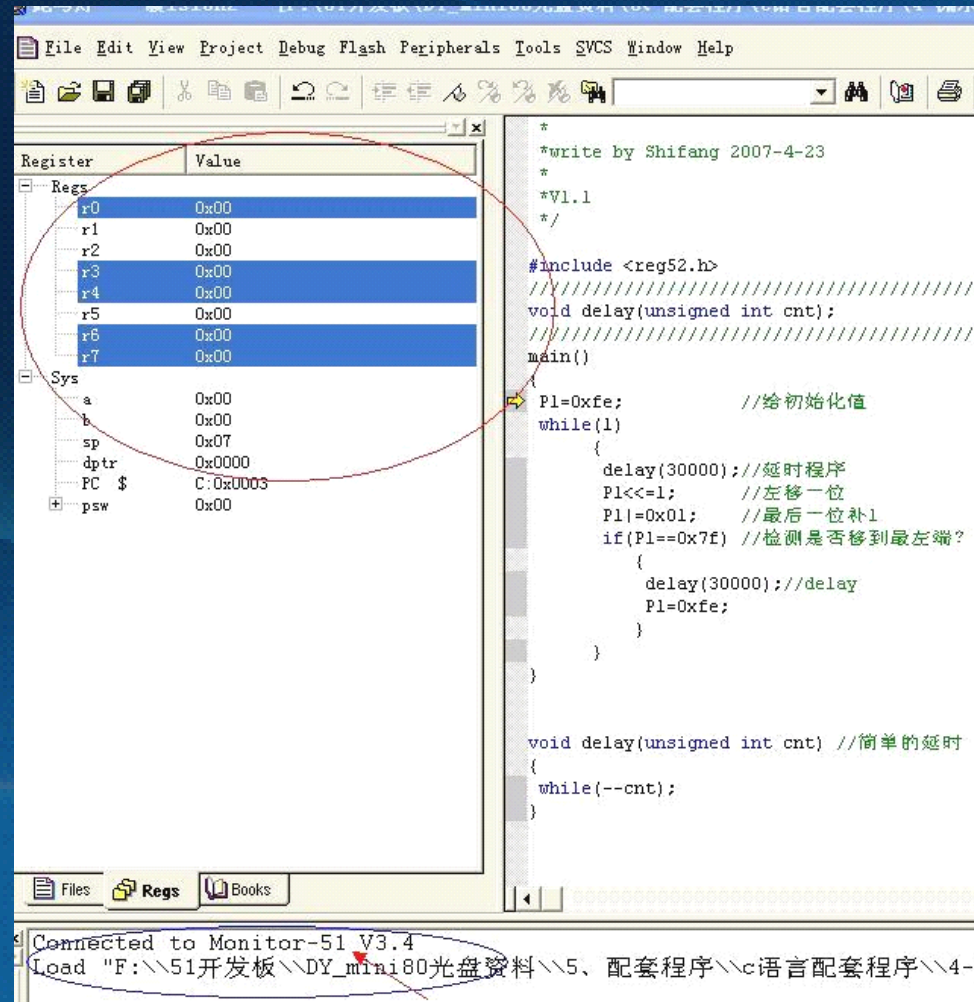
## 不能进入仿真原因

- 说明通讯异常请按一下目标板复位键或者重新启动电源，然后点击“TryAgain”即可，如果多次尝试仍不能进入仿真界面，请点击“Settings”检测端口和波特率是否正确。然后重复操作。

# 弟子教你学51单片机技术



下图是成功进入的界面，供参考对比



从零到项目开发学51单片机与C语言技术视频教程



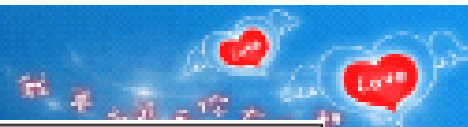
## 仿真器的版本

- 左侧栏有如图现象，状态栏显示仿真器的版本，这里显示 **Monitor – 51 V3.4**
- 这个时候就可以进入仿真步骤了
- 下图分别是（从左往右）：复位、全速运行、停止、单步、跨步





# 弟子教你学51单片机技术



Debug 菜单	工具栏	快捷键	描述
Start/Stop		Ctrl+F5	启动或停止 $\mu$ Vision2 调试模式
Debugging			
Go		F5	运行（执行），直到下一个有效的断点
Step		F11	跟踪运行程序
Step Over		F10	单步运行程序
Step out of current function		Ctrl+F11	执行到当前函数的程序
Stop Running		ESC	停止程序运行
Breakpoints...			打开断点对话框
Insert/Remove Breakpoint			在当前行设置 / 清除断点
Enable / Disable Breakpoint			使能 / 禁能当前行的断点
Disable All Breakpoints			禁能程序中所有断点
Kill All Breakpoints			清除程序中所有断点
Show Next			显示下一条执行的语句 / 指令

从零到项目开发学51单片机与C语言技术视频教程



# 全速运行

- 点击全速运行就可以看到实际执行效果。
- 用户也可以单步、跨步或者断点运行。



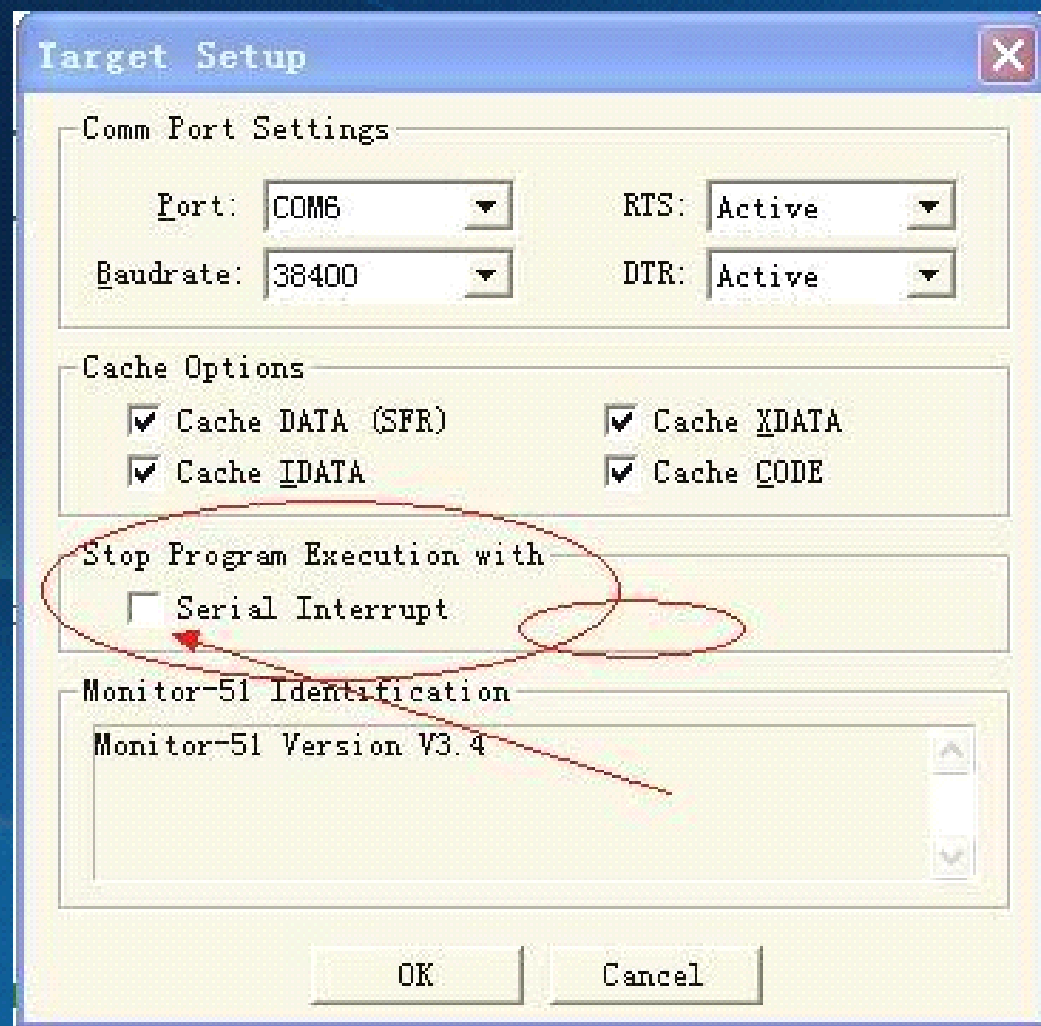
# 使用注意事项

- 1、全速运行后不能通过点击“
  - 下图中“\*”停止。产生原因如下：
  - ”未勾选。建议不要勾选，否则会影响仿真的正确性。
  - 这时候我们如果想停止正在全速运行的程序可以通过用户板的复位按键停止，然后重新进入其他仿真步骤。

Serial Interrupt



# 使用注意事项





## 使用注意事项

- 这时候我们如果想停止正在全速运行的程序可以通过用户板的复位按键停止，然后重新进入其他仿真步骤。



# 不能正常进入仿真界面

- 请检查一下原因：
  - ◆ 硬件是否连接牢靠
  - ◆ 电源是否打开
  - ◆ 软件设置是否正确
  - ◆ 重新启动 keil 软件





## 小结

一般情况下，我们可以用软件仿真来检查流程错误或其他纯软件性的错误；用“仿真芯片”仿真法来实现实际硬件环境下的软硬件调试。



## 课后作业

对前面几讲中课堂讲解的以及课后作业中已经完成的例程，先分析可用哪种仿真手段，然后具体进行仿真。