

## 第 1 章 概述

非常感谢您使用长沙菊阳微电子有限公司的菊阳单片机仿真调试系统 JY2X00。

本系统包括 JY-E2X00C 仿真器和 JY2X00 仿真调试软件。

JY-E2X00C 仿真器是采用新一代仿真专利技术开发的单片机仿真产品。由于采用了新一代仿真专利技术，它解决了困扰单片机仿真领域的四大问题：

解决了 Bondout 技术不能仿真增强资源的问题  
解决了 Hooks 技术仿真频率低、且不能支持非 Philips 芯片的问题  
解决了 IAP 功能仿真的问题  
解决了单片机厂家开发多引脚、多功能而担心无仿真芯片的问题

从而进入了全面、真实、直接、高速的仿真新时代：

全面：全面支持具有类似 EA、ALE 和 PSEN 引脚的各种单片机  
真实：真实仿真所有标准资源和增强资源，零资源占用  
直接：用户单片机直接用作仿真芯片，减少用户投资  
高速：能以 40M 的频率稳定仿真外部数据存储器

一旦拥有新一代专利技术仿真器：

不再购买昂贵的仿真头  
不再使用兼容模式而直接仿真 500 多种芯片  
不再存在 Bondout、Hooks 技术作兼容仿真时不真实的问题

### 1.1 JY-E2X00C 仿真器的特点

● **仿真适配头上的芯片就是用户要仿真的芯片**：任何端口（P0、P1、…、Pn）不需改变定义，其功能（包括基本的输入输出功能、附加的 A/D、D/A 功能、附加的中断功能、I<sup>2</sup>C 或 SPI 接口等）完全相同；任何内部资源不需改变定义即可直接仿真，这些资源包括 A/D、D/A、EXTRAM、EEPROM、I<sup>2</sup>C、SPI、PCA、ISP、PWM、双 UART、双 DPTR、WDT、RTC、CAN 等；仿真调试时产生的目标文件（\*.hex,\*.bin）可以直接用来烧写芯片，不需要作任何改变，这是兼容仿真不可能做到的。

● **能仿真的芯片数量至少增加十倍**：凡是具备 EA、ALE 和 PSEN 引脚的单片机，插入仿真适配头即可。支持 Acer Labs、Aeroflex UPMC、Analog Devices、Atmel、Cypress、Dallas、Diosys、Hynix、Infineon、Intel、ISSI、NXP（Philips）、Nuvoton（Winbond）Russia、SST、STC、SyncMOS、TI 等厂家的 500 余种单片机。

● **用少量、廉价仿真适配头替代繁多、昂贵的传统仿真头**：彻底摆脱了使用品种繁多的仿真头的传统设计方法，既省去了购买仿真头的昂贵费用，又解决了仿真真实性问题；相同封装和引脚排列的芯片共用一种廉价仿真适配头，与时钟周期无关，与芯片资源无关；不同封装或不同引脚排列的芯片采用不同的仿真适配头；当用户板不使用单片机插座时（如贴片或特殊封装的单片机），用户只需按

我们提供的仿真接口原理图在用户板上引出仿真接口即可。

● **完全不占用用户的任何资源**：不占堆栈空间、引脚、程序空间、数据空间、定时器和中断等用户的任何资源。

● **同一个仿真器同时支持 12 clocks、6 clocks、4 clocks 的芯片**：对于不同时钟周期的单片机，您只需要将该芯片插入仿真适配头上。

● **P0、P2 口的特性好**：P0 口完全开漏；P2 口弱上拉；P0、P2 口在 0 到 1 或 1 到 0 的转换过程中转换速度特别快；在读写片外 RAM 时，P0、P2 口能够输出恰当驱动能力的地址或数据信号，恰当到仿真频率为 33M 时，能够可靠读写片外 RAM，这时并不需要在 P0 口外接上拉电阻；P0、P2 口可以在同一程序中既作总线又可作 I/O 用。

● **可以仿真 IAP 功能**：STC、NXP（Philips）等单片机支持在运行中对 Flash 进行擦除、编程、校验，该功能为用户存放掉电不挥发数据提供了方便或节省了成本，JY-E2X00C 仿真器完全支持此功能。

● **既可在 Keil 环境下联机调试，又可在强大的菊阳单片机仿真调试软件 JY2X00 下联机调试。**

● **性价比极高**：采用“主机 + 仿真芯片适配头 + 用户单片机”的组合模式，在大幅提高了可仿真芯片数量、仿真频率、仿真真实性的前提下，新一代仿真专利技术仿真器的价格不高于国内高档仿真器的价格。

## 1.2 包装

在菊阳单片机仿真调试系统 JY2X00 包装中，包括以下配置：

1	菊阳单片机仿真调试系统JY2X00用户手册	1本
2	JY-E2X00C仿真器	1台
3	JY2X00仿真调试软件安装盘	1张
4	USB通信电缆	1根
5	仿真排线	3根
6	DIP-40仿真适配头	1个
7	PLCC-44仿真适配头	1个
8	DIP-20仿真转换头	1个
9	合格证	1张
10	保修卡	1张

与菊阳单片机仿真调试系统JY2X00配合使用的可选附件有：

- 各种其它封装的仿真芯片适配头
- 逻辑分析仪

### 1.3 菊阳单片机仿真调试系统 JY2X00 对计算机的要求

计算机是使用菊阳单片机仿真调试系统JY2X00所必须的工具，对您的计算机最低的配置要求是：

- 8X586以上的PC及兼容机
- Windows 98以上版本
- 至少64MB的有效内存
- 50MB以上的剩余硬盘空间
- 一个USB端口
- 分辨率为800x600以上的显示器
- 已安装 V6.23 以上版本的 Keil 集成调试环境或 Keil C51 编译器

## 1.4 JY-E2X00C 仿真器性能指标

仿真器型号	JY-E2300C
类 别	通用仿真器
支持芯片	Atmel、Winbond、Philips 等二十几个厂家的芯片 (支持 51MX 核、双 Bank)
仿真程序空间	内部 128K
仿真数据空间	外部 8MB
IAP空间	内部64K
任意位置 64K 点	√
外部复位输入支持	√
实时仿真频率	40M@12T, 24M@6T, 24M@4T
CPU电源内外选择	√
CPU工作电压	2.7V~5.5V
支持仿真范围	标准资源 增强资源
后续芯片支持性	√
仿真模式	直接仿真
通信模式	USB
随机仿真适配头	DIP-40、PLCC-44、DIP-20
仿真芯片	被仿真的用户芯片
调试环境	Keil 和菊阳单片机仿真调试软件 JY2X00

## 1.5 JY-E2X00C 仿真器可仿真的单片机型号列表

<b>AD</b>	ADuC812、ADuC814、ADuC816、ADuC824、ADuC831、ADuC832、ADuC834、ADuC836、ADuC841、ADuC842、ADuC843、ADuC844
<b>Atmel</b>	AT48801、AT80F51、AT80F52、AT80LV51、AT80LV52AT87F51、AT87F51RC、AT87F52、AT87F55、AT87LV52、AT87LV55、AT89C1051、AT89C1051X2、AT89C2051、AT89C2051X2、AT89C4051、AT89C51、AT89C5131、AT89C51AC2、AT89C51ED2、AT89C51ID2、AT89C51RB2、AT89C51RC、AT89C51RC2、AT89C51RD2、AT89C52、AT89C55、AT89C55WD、AT89C8253、AT89F51、AT89F52、AT89LS51、AT89LS52、AT89LS53、AT89LS8252、AT89LV52、AT89LV55、AT89S4D02、AT89S51、AT89S52、AT89S53、AT89S54、AT89S8252、AT89S8254、T80C31、T80C31X2、T80C51、T80C51FP1、T80C51RA2、T80C51RD2、T80C51U2、T80C52X2、T80C54X2、T80C58X2、T83-87C51RB2、T83-87C51RC2、T83-87C51RD2、T83-87C51RB2、T83-87C51RC2、T83-87C51RD2、T83-87C51U2、T83-87C52X2、T83C5101、T83C5102、T83C5112、T87C251G1A、T87C51、T87C5112、T89C5121、T89C51AC2、T89C51CC01、

<b>Atmel</b>	T89C51CC02、T89C51IB2、T89C51IC2、T89C51RB2、T89C51RC2、 T89C51RD2、T8xC251G1D、T8xC251G2D、T8xC51SND1、 TS80C31X2、TS80C32X2、TS80C51RA2、TS80C51U2、 TS80C52X2、TS80C54X2、TS80C58X2、TS83C51RB2、 TS83C51RC2、TS83C51RD2、TS83C51U2、TS87C51RB2、 TS87C51RC2、TS87C51RD2、TS87C51U2、TS87C52X2、 TS87C54X2、TS87C58X2、TSC80C31、TSC80C51
<b>Dallas</b>	DS5000、DS5240、DS80C310、DS80C313、DS80C320、DS80C323、 DS80C390、DS80C400、DS82CH11、DS83C520、DS83C530、 DS87C520、DS87C530、DS87C550、DS89C420
<b>Hynix Semiconductor</b>	GMS90C320、GMS90C51、GMS90C52、GMS90C54、GMS90C56、 GMS90C58、GMS90L32、GMS90L320、GMS90L51、GMS90L52、 GMS90L54、GMS90L56、GMS90L58、GMS97C1051、GMS97C2051、 GMS97C51、GMS97C52、GMS97C54、GMS97C56、GMS97C58、 GMS97L1051、GMS97L2051、GMS97L51、GMS97L52、GMS97L54、 GMS97L56、GMS97L58、HMS9XC7132、GMS90X5X、GMS99C58
<b>Infineon</b>	C505-2R、C505A-4E、C505C-2R、C505C-L、C505-L、C508、C509-L、 C511-R、C513A-2R、C513A-R、C513-R、C515-1R、C515-4R、 C515A-L、C515B-2R、C515C-8R/8E、C515C-L、C515-L、C540U-E、 C541U-2E、C868、SAB80C515、SABC501G、SABC517A-LN、 SAB80C504-EM、SAB80C504-LM、SAB80C504-RM、 SAB80C515C-XX、SAB80C517A-4RM、SAB80C517A-L24M、 SAB80C517A-M、SAB80C517A-N、SAB83C517A-5N18、 SAFC501G、SAFC517A-LN、SAF80C515C-XX
<b>Intel</b>	80/83/87C51SL、80/87C52、80/87C54、80/87C58、80/87L52、 80/87L54、80/87L58、8031AH、8032AH、8051AH、8052AH、 80C152JA、80C152JB、80C152JC、80C152JD、80C31、80C31BH、 80C32、80C51、80C51BH、80C51FA、80C51GB、80C51SL、 80L51FA、83/87C51FA、83/87C51FB、83/87C51FC、83/87C51GB、 83/87C51RA、83/87C51RB、83/87C51RC、83/87L51FA、 83/87L51FB、83/87L51FC、83C152JA、83C152JC、87C51、 8XC151SA、8XC151SB
<b>ISSI</b>	IS80C31、IS80C32、IS80C51、IS80C52、IS80LV31、IS80LV32、 IS80LV51、IS80LV52、IS89C51、IS89C52
<b>NXP (Philips)</b>	80/87C51、80/87C52、80C31、80C31X2、80C32、80C32X2、 80C451、80C451、80C51FA、80C51RA+、80C528、80C550、 80C552、80C554、80C575、80C652、80/87C51、80/87C52、 83/87C451、83/87C524、 83/87C528、83/87C550、83/87C552、83/87C554、83/87C575、 83/87C652、83/87C654、83C51F8、83C51FA、83C51FC、 83C51RA+、83C51RB+、83C51RC+、83C51RD+、87C51FA、 87C51FB、87C51FC、 87C51RA+、87C51RB+、87C51RC+、87C51RD+、8951RA2XX、 89C51、89C51RA+、89C51RB+、89C51RB2XX、89C51RC+、 89C51RD+、89C52、89C54、89C58、8XC51FA/8xL51FA、

	8XC51FB/8xL51FB、8XC51FC/8xL51FC、8XC51MA2、8XC51MB2、8XC51MB2/02、8XC51MC2、8XC51MC2/02、8XC51RA+、8XC51RB+、8XC51RC+、8XC51RD+、8XC52、8XC54、8XC58、P80/P87C51X2、P80/P87C52X2、P80/P87C54X2、P80/P87C58X2、P80C557E4、P80C557E6、P80C557E8、P80C562、P80C591、P80C592、P80CE558、P80CE560、P80CE598、P80CL31、P80CL410、P80CL51、P80CL580、P83/87C654X2、P83/87C660X2、P83/87C661X2、P83/P87C560、P83_P87C557E8、P83/P89C557E4、P83/P89CE558、P83C557E6、P83C562、P83C591、P83C592、P83CE598、P83CL410、P83CL580、P83CL882、P87C51RA2、P87C51RB2、P87C51RC2、P87C51RD2、P87C591、P87CL52x2、P87CL54x2、P87CL888、P89C51RA2XX、P89C51RB2HXX、P89C51RB2XX、P89C51RC2HXX、P89C51RC2XX、P89C51RD2HXX、P89C51RD2XX、P89C51X2、P89C52X2、P89C54X2、P89C58X2、P89C60X2、P89C61X2、P89C660、P89C662、P89C664、P89C668、P89C669、P89C738、P89C739、P89LV51RD2、P89V51RD2、PCD50913、PCD50917、PCD50923、PCD50927、PCD50933、PCD50937、PCD50953、PCD50957、PCD6001、PCD6002、PCD6003、SAA5645HL、SAA5647HL、SAA5665HL、SAA5667HL、TDA8006、TDA8008、TDA8028、TDA8029
<b>SST</b>	SST89C54、SST89C58、SST89C59、SST89E554、SST89E554A、SST89E554RC、SST89E554RD、SST89E564、SST89E564RC、SST89E564RD、SST89F54、SST89F58、SST89V554、SST89V554A、SST89V554RC、SST89V554RD、SST89V564、SST89V564RC、SST89V564RD、SST89E516RD2、SST89V516RD2、SST89E52RD2、SST89V52RD2、SST89E54RD2、SST89V54RD2、SST89E58RD2、SST89V58RD2、SST89E51RC、SST89V51RC、SST89E52RC、SST89V52RC、SST89E54RC、SST89V54RC、SST89E51LC、SST89V51LC、SST89E52LC、SST89V52LC、SST89E54LC、SST89V54LC
<b>STC</b>	STC89C51RC、STC89C52RC、STC89C53RC、STC89C54RD+、STC89C58RD+、STC89C516RD+、STC89LE51RC、STC89LE52RC、STC89LE53RC、STC89LE54RD+、STC89LE58RD+、STC89C516RD+、STC89LE58AD、STC89LE516AD、STC89LE516X2、STC89C58RD、STC89C516RD、STC89LV516RD
<b>SyncMos</b>	SM8951A、SM8952A、SM8954A、SM8958A、SM89516A、SM5964、SM59264、SM79164、SSU7301/SM79108、SM5964A、SM89S16R1、SM89T16R1
<b>TI</b>	MSC1200Y2、MSC1200Y3、MSC1210Y2、MSC1210Y3、MSC1210Y4、MSC1210Y5、MSC1211Y2、MSC1211Y3、MSC1211Y4、MSC1211Y5、MSC1212Y2、MSC1212Y3、MSC1212Y4、MSC1212Y5
<b>Nuvoton</b>	W77C32、W77C512、W77C516、W77C58、W77E468、W77E516、W77E532、W77E58、W77IC32、W77IE58、W77L32、W77LE516、

	W77LE532、W77LE58、W77X32、W78C32B、W78C32C、W78C33B、 W78C354、W78C438C、W78C516、W78C51D、W78C52D、 W78C54、W78C58、W78C801、W78E/78C378、W78E354、 W78E365、W78E374B、W78E516、W78E516B、W78E51B、 W78E52、W78E52B、W78E532、W78E54、W78E54B、W78E58、 W78E58B、W78E58、W78ERD2、W78IE52、W78IE54、W78IRD2、 W78L32、W78L33、W78L51、W78L52、W78L54、W78L801、 W78LE365、W78LE51、W78LE516、W78LE52、W78LE54、 W78LE58、W78LE812、W78x374、W79E532
--	--

直接支持以上公司陆续推出的具备EA、ALE和PSEN引脚的各种封装的单片机仿真。

## 1.6 技术服务

用户在使用菊阳单片机仿真调试系统 JY2X00 时，对产品的建议、要求和技术服务，都可以与长沙菊阳微电子有限公司联系；对产品的升级、维修服务请与代理（销售）商或本公司联系。

长沙菊阳微电子有限公司联络方式：

热线电话：0731-85201391

传 真：0731-85207966

网 址：www.jywdz.com

电子信箱：support@jywdz.com

## 1.7 产品维修

用户购买的菊阳单片机仿真调试系统JY2X00，本公司已对其性能和技术指标进行了严格的测试。如果用户在使用过程中遇到仿真器损坏的现象，请直接将仿真器寄给长沙菊阳微电子有限公司维修部，由我公司直接快捷地为您服务。

本公司对JY-E2X00C仿真器硬件实行一年免费保修，并提供终生维修服务，只收取成本费。

特别说明：FPGA芯片损坏不在包换和保修之列，但提供维修服务，只收取元器件成本费。

## 第 2 章 JY-E2X00C 仿真器的硬件结构及安装

本章介绍单片机仿真器的硬件结构和硬件安装。

### 2.1 JY-E2X00C 仿真器硬件结构

JY-E2X00C 的整体结构如下图 2-1 所示。



图 2-1 JY-E2X00C 的整体结构图

#### 2.1.1 JY-E2X00C 仿真器外观

JY-E2X00C 仿真器使用大规模集成电路设计，仿真器内部无跳线，无按键，无开关，所有硬件配置和运行控制全由菊阳单片机仿真调试软件 JY2X00 或由 keil 驱动程序设置。JY-E2X00C 仿真器上盖右下方有三个 LED 指示灯，分别为 POWER（电源，红色 LED）、RUN（运行，绿色 LED）和 BUSY（状态，黄色 LED）指示灯；左侧为一个 80Pin 仿真电缆插座；右侧分别为逻辑分析仪接口、USB 通信电缆插座、电源开关，如图 2-2 所示。

#### 2.1.2 仿真电缆及仿真芯片适配头

仿真电缆和仿真芯片适配头用于 JY-E2X00C 仿真器与用户目标系统的连接。与 JY-E2X00C 仿真器配套的仿真芯片适配头主要有：1）DIP-40 仿真芯片适配头（必配）；2）PLCC-44 仿真芯片适配头（必配）；3）PLCC-68 仿真芯片适配头（选配）；4）DIP-20 转换头（必配）；5）JY-C515C-8E/8R 仿真适配头（选配）；6）JY-AduC812/834 仿真适配头（选配）；7）JY-STC1T 仿真头（选配）。

##### 1. DIP-40 仿真芯片适配头

- DIP-40 仿真芯片适配头外观如图 2-3 所示。
- DIP-40 仿真芯片适配头的结构如图 2-4 所示，与仿真器的连接如图 2-5 所示。



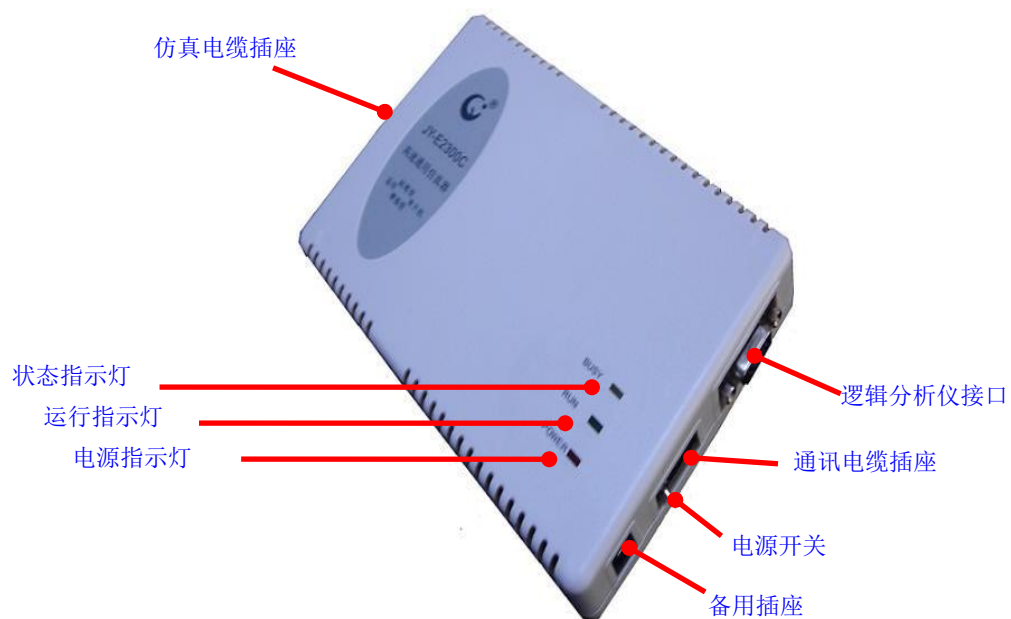


图 2-2 JY-E2X00C 仿真器外观

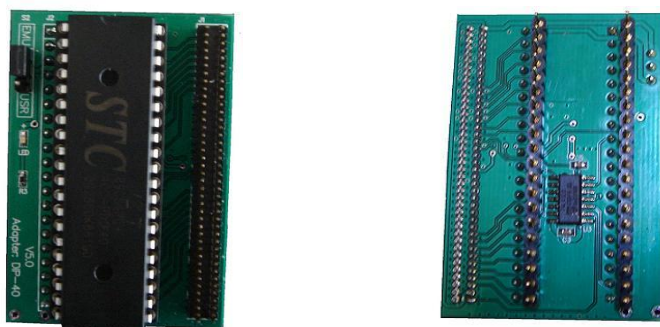


图 2-3 DIP-40 仿真芯片适配头外观图

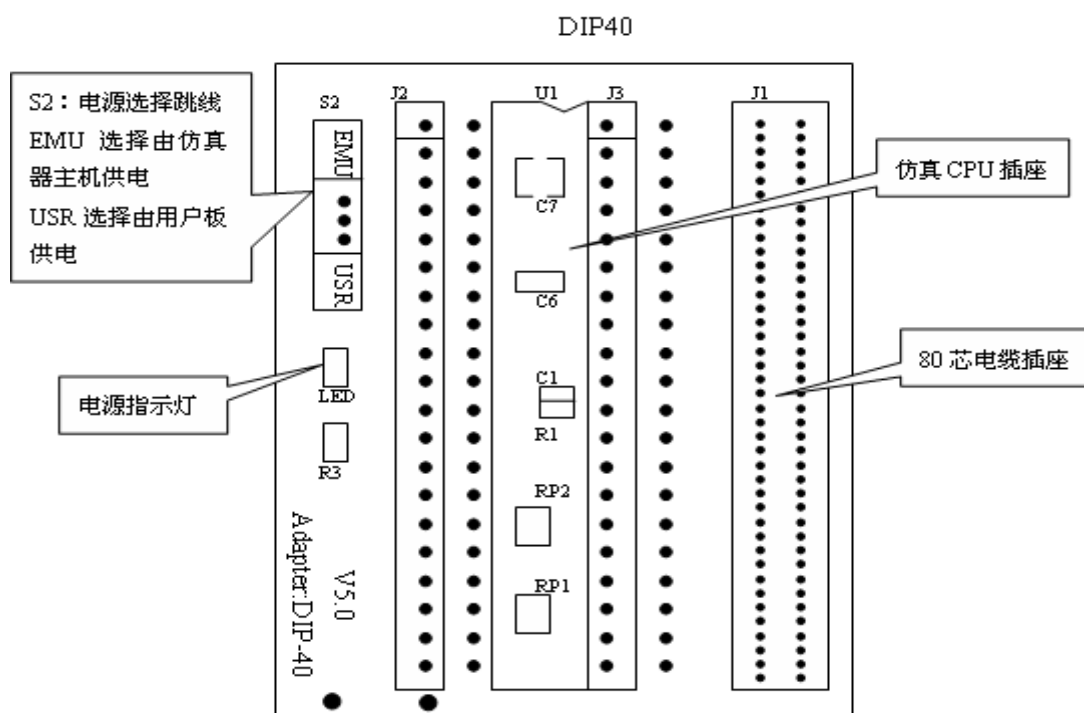


图 2-4 DIP-40 仿真芯片适配头

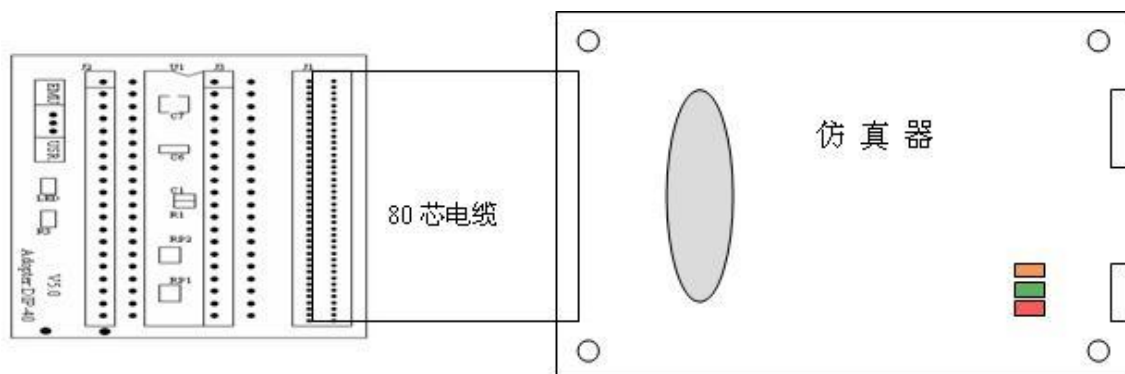


图 2-5 DIP-40 仿真芯片适配头与仿真器的连接

- 插座的连接：  
80芯电缆插座：连接仿真器和仿真芯片适配头；  
仿真CPU插座：插入被直接仿真的单片机；  
40 针仿真插头：插入用户目标板的插座中。

## 2. PLCC-44 仿真芯片适配头

- PLCC-44 仿真芯片适配头的外观如图 2-6 所示。
- PLCC-44 仿真芯片适配头的结构如图 2-7，与仿真器的连接如图 2-8 所示。

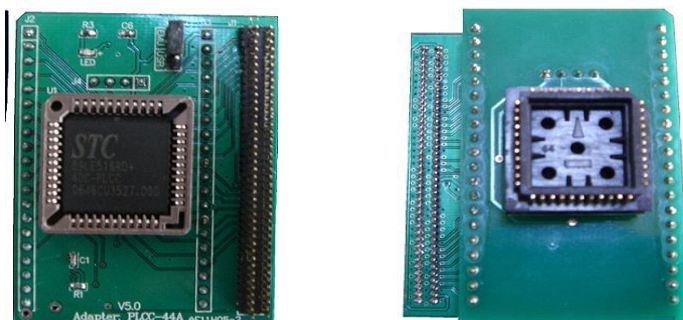


图 2-6 PLCC-44 外观图

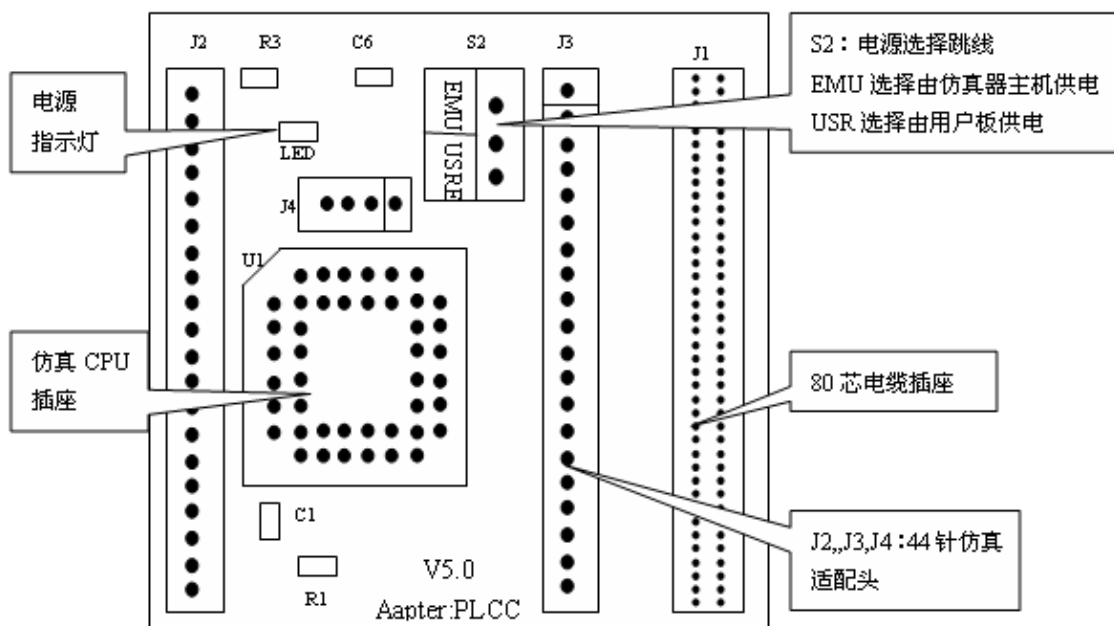


图 2-7 PLCC-44 仿真芯片适配头

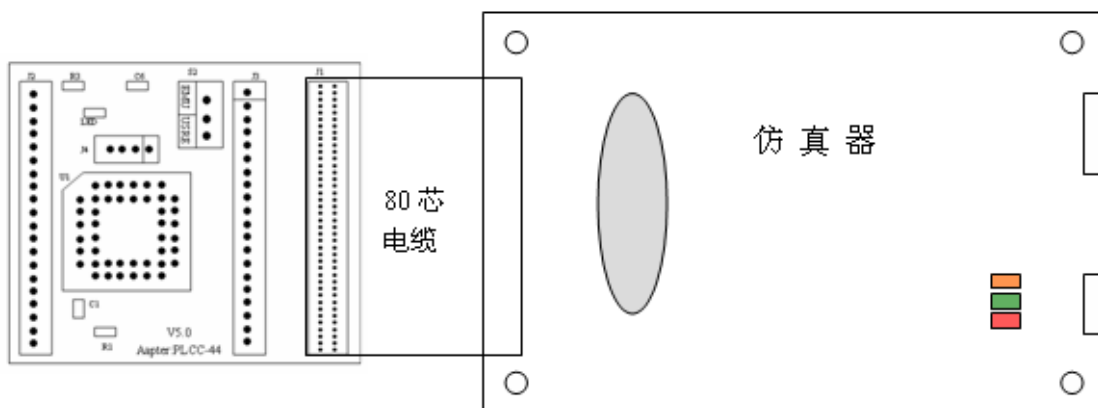


图 2-8 PLCC-44 仿真芯片适配头与仿真器的连接

- PLCC-44 仿真芯片适配头上的跳线和其它部件的使用方法与 DIP-40 仿真芯片适配头相同。

### 3. DIP-20 转换头

DIP-20 转换头是用于对 DIP-20 封装的 1050、2050、4050 芯片的仿真进行转换。

使用DIP-20转换头进行仿真的方法：

- 在DIP-40仿真适配头上的仿真CPU插座中插入AT89C51芯片；
- 将DIP-40仿真适配头的插针插入DIP-20的插座中；
- 将DIP20的插针插入你目标板的插座中；
- 在调试环境中将芯片选择为AT89C51。

### 2.1.3 通信电缆

USB 通信电缆用于连接 JY-E2X00C 仿真器和用户计算机，长度不大于 1.5 米。

## 2.2 硬件安装

JY-E2X00C 仿真器硬件的安装包括连接仿真目标系统、连接通信电缆、安装 USB 驱动程序和接通电源测试等。

### 2.2.1 连接仿真目标系统

JY-E2X00C仿真器对用户目标系统进行在线仿真时，按照以下步骤连接：

- ①根据用户对目标系统仿真的要求，选择DIP-40仿真芯片适配头或PLCC-44仿真芯片适配头；
- ②将要仿真的CPU（即用户CPU）插入所选仿真芯片适配头上方的仿真CPU插座；
- ③将仿真芯片适配头下方的仿真插头插入用户目标系统的目标CPU插座，并注意第一脚的位置和方向；
- ④设置仿真芯片适配头上的跳线（出厂时已有默认设置，如无特别的需求可不设置）；
- ⑤将仿真电缆的两端分别插到 JY-E2X00C 仿真器的仿真插座和仿真芯片适配头的 80 芯电缆插座中。

### 2.2.2 连接通信电缆

JY-E2X00C 仿真器通过 USB 通信电缆与计算机交换信息，请将 USB 电缆线分别连接到仿真器 USB 座和 PC 机的任一 USB 插座上。

### 2.2.3 安装 USB 驱动程序

对于 USB 的仿真器，应安装 USB 驱动程序。

- ①安装 JY2X00 仿真调试器软件或 JY-E2X00C 仿真器嵌入 Keil 调试环境驱动程序；
- ②将仿真器电源开关置 OFF 状态；连接仿真器电源适配器；用 USB 连接线将仿真器与 PC 电脑的 USB 端口相连；

- ③打开仿真器电源，电脑发现新硬件，Windows 2000/XP 操作系统将自动完成安装。

为了验证驱动程序的正确性，在Windows的设备管理器（右键点击“我的电脑”，依次选择“属性”、“硬件”、“设备管理器”即可访问到（或从“开始”菜单的“设置”下面启动“控制面板”，然后双击“系统”，打开“设备管理器”））。

在“设备管理器->通用串行总线控制器”里面可看到“JY-E2X00 USB Driver”设备，如图2-10所示。

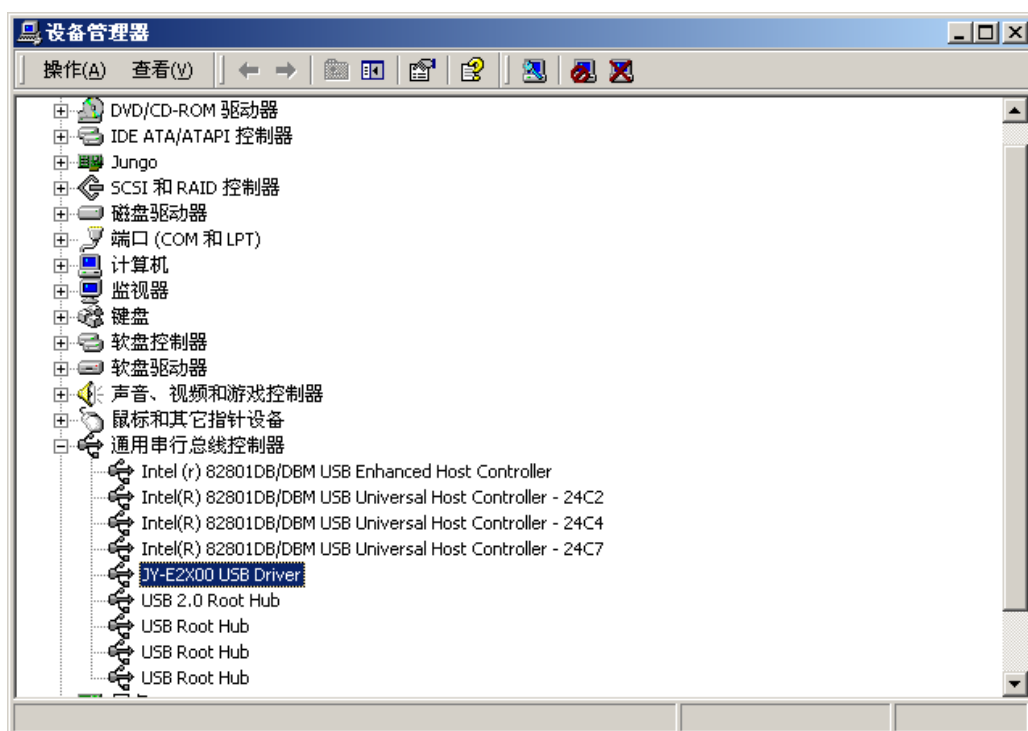


图 2-10 查看设备管理器

## 2.2.4 通电测试

电源适配器连接JY-E2X00仿真器时，先将直流输出插入JY-E2X00仿真器的电源插座，再将电源适配器的AC插头插入220V插座，此时应观察到如下过程：

- ① 仿真器的POWER指示灯立即点亮；
- ② 接下来BUSY指示灯点亮约1秒钟；
- ③ RUN指示灯亮，并立即与BUSY指示灯同时熄灭。

否则请关闭电源，进行如下检查：

- ① 仿真适配头排线或仿真芯片是否插反；
- ② 有些厂家的单片机（如Atmel、NXP(Philips)）加密后即使EA=0也不能从0地址处开始运行外部程序，对于这些芯片应首先擦除其加密位。
- ③ 有些厂家的单片机（如Atmel、NXP(Philips)的RD2系列）需要配置引导状态字“Boot Status Byte”（即“BSB”）。这些芯片出厂时默认为“FF”，需用编程器改为“00”后方可仿真。

特别注意：上述三种情况都可能会导致仿真器的三个指示灯全亮，出现这种现象时，应按上述说明检查。

对用户目标系统进行在线仿真时，应提供目标系统电源，使目标系统能够在仿真器仿真状态下工作。如果仿真芯片适配头由仿真器（EMU）供电，上电时应先给仿真器供电，再给目标系统供电，相反，如果仿真芯片适配头由目标系统（USR）供电，应先给目标系统供电，再给仿真器供电。

## 2.3 注意事项

- 仿真器各部分连接时，务必关闭电源，切不可带电拔插；
- 避免仿真器工作在强烈震动，高温，高湿度或者其它恶劣环境；
- 在使用目标系统供电的情况下，保证输入电源不超过 5.5V；
- 尽管仿真器内部设有保护电路，但仿真适配头长时间反插或电源长时间超过 5.5V 会有永久损坏仿真器的危险，通电前的仔细检查是非常必要的；
- 未经厂家允许，不要自行打开 JY-E2X00C 仿真器外壳。

## 第3章 JY-E2X00C 仿真器在 Keil 环境中的使用

Keil 软件是众多单片机应用开发的优秀软件之一，它集编辑、编译、仿真于一体，支持汇编、PLM 语言和 C 语言的程序设计，界面友好，易学易用。

在功能强大的 Keil 集成开发环境下，您可以使用 JY-E2X00C 仿真器进行软件设计和硬件调试。

### 3.1 安装嵌入 Keil 环境的驱动程序

在 Keil  $\mu$ Vision x 集成调试环境下使用 JY-E2X00C 仿真器，需要在 Keil 环境中安装 JY-E2X00C 仿真器嵌入 Keil 环境的驱动程序，安装方法如下：

将随机光盘“JY2X00 仿真开发系统安装盘”放入计算机的光驱中，光盘能自动运行并出现以下界面。



图3-1 JY2X00仿真开发系统安装盘运行界面

点击“安装嵌入Keil调试环境的驱动程序”，启动JY-E2X00C仿真器嵌入Keil系统安装程序，程序能自动搜索到您安装的Keil目录（如有多个Keil安装目录，则会选择您最后一次安装的Keil目录），在驱动程序选择列表框中，根据您的仿真器型号和被仿真的单片机所属厂家选择您所需要的驱动程序，例如，我们想用JY-E2100C通用仿真器来仿真Atmel、Philips和Winbond公司的单片机，则应选取相应的专用驱动程序，如图3-2所示，点击“安装”按钮，开始自动安装。

对于专用仿真器，只能选取相应的专用驱动程序，对于通用仿真器，可选择右上方的“全选”复选框，将安装所有能支持的驱动程序。



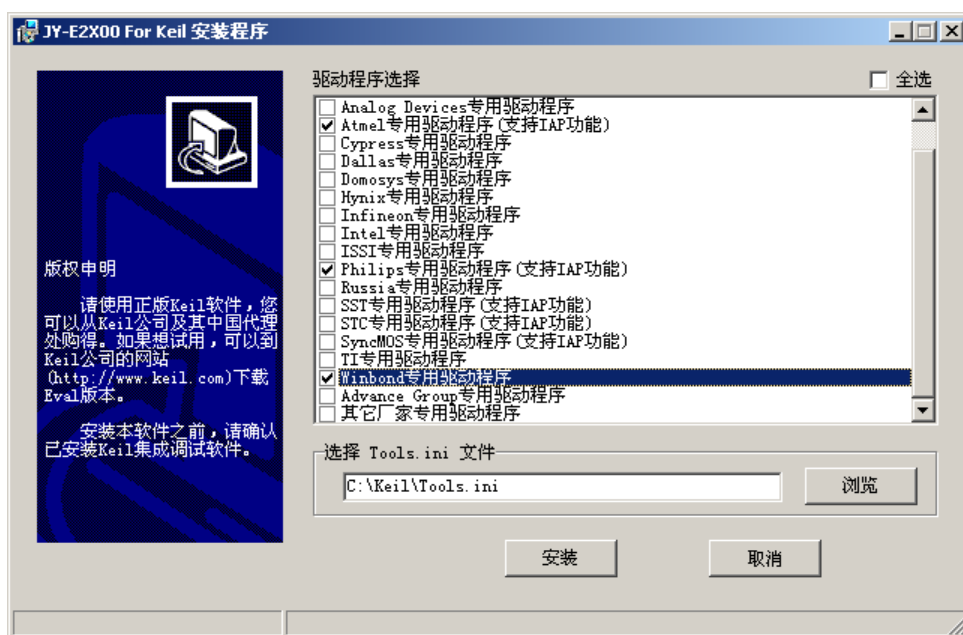


图3-2 JY-E2X00C仿真器嵌入Keil系统安装程序

如果您的Keil系统为非正常安装或者想选择其它的Keil目录，请点击“浏览”按钮，选择Keil安装目录下的Tools.ini文件，如图3-3所示，单击“打开(O)”，返回图3-2后，再点击“安装”按钮。

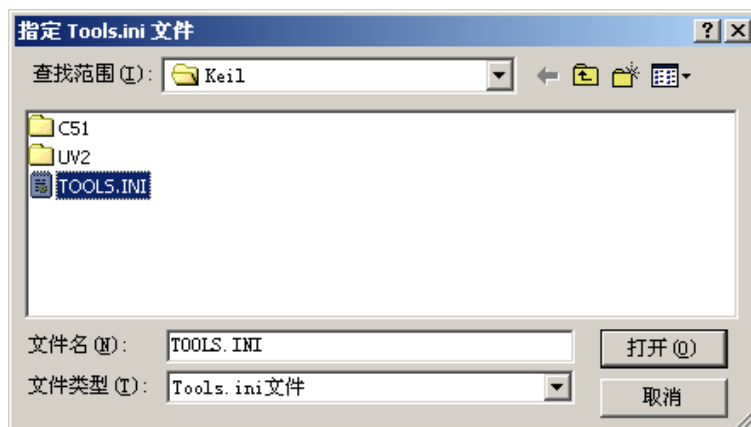


图3-3 选择Tools.ini文件



图3-4 仿真器成功嵌入Keil系统

安装结束后，弹出如图3-4所示的消息框，按“确定”按钮退出安装。

这样，用户在Project->Option for Target->Debug页的硬件仿真目标器件驱动程序列表框中，可以选择“JY-E2X00 Emulator Drivers”了，如图3-5所示。



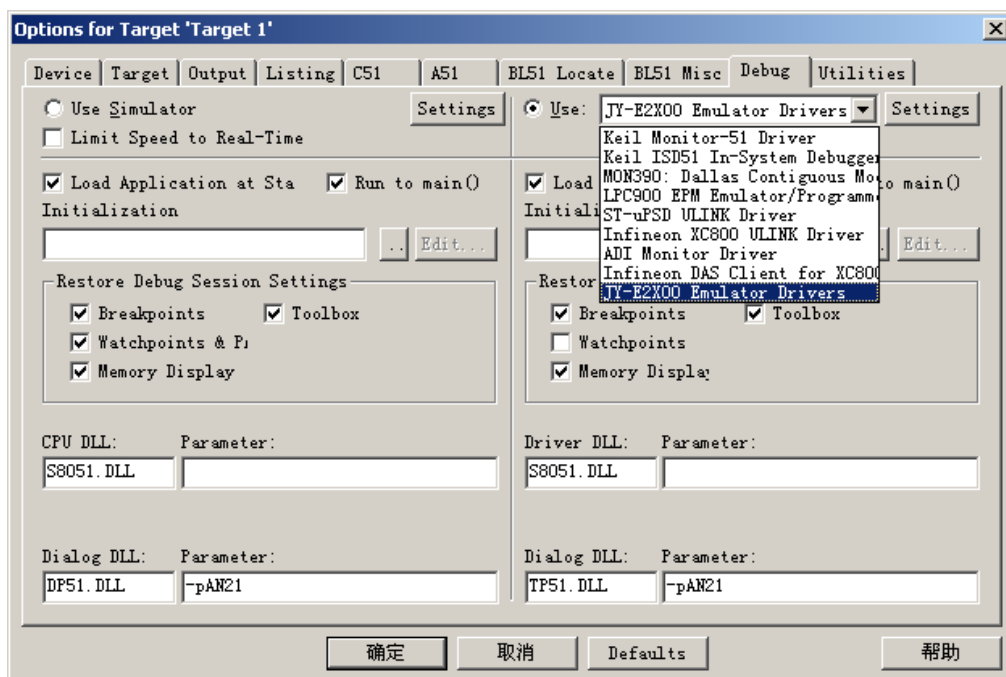


图3-5 硬件仿真目标器件驱动程序列表框

## 3.2 设置硬件仿真参数

在使用JY-E2X00C仿真器进行调试前，除了正确安装JY-E2X00C仿真器嵌入Keil系统程序和正确连接JY-E2X00C仿真器外，还应做相应的设置。

设置步骤如下：

**第一步：**启动Keil 程序，新建或打开一个要调试的Project文件。

**第二步：**单击菜单“Project->Option for Target”，进入工程设置对话框，选择Debug页，出现如图3-6所示的对话框，选中“Use”单选按钮（图中1），并进行硬件仿真的一些设置。

图3-6中相关参数的含义：

- “Use”单选按钮（图中 1）：选择硬件仿真。
- 选择仿真器的类型(图中 2)，这里我们通用 JY-E2X00 Emulator Drivers。
- Load Application at Start （图中 3）：进入仿真后，keil 自动装载程序代码至仿真器。
- Run to main()（图中4）：调试c语言程序时可选择此项，PC直接运行到main 函数。

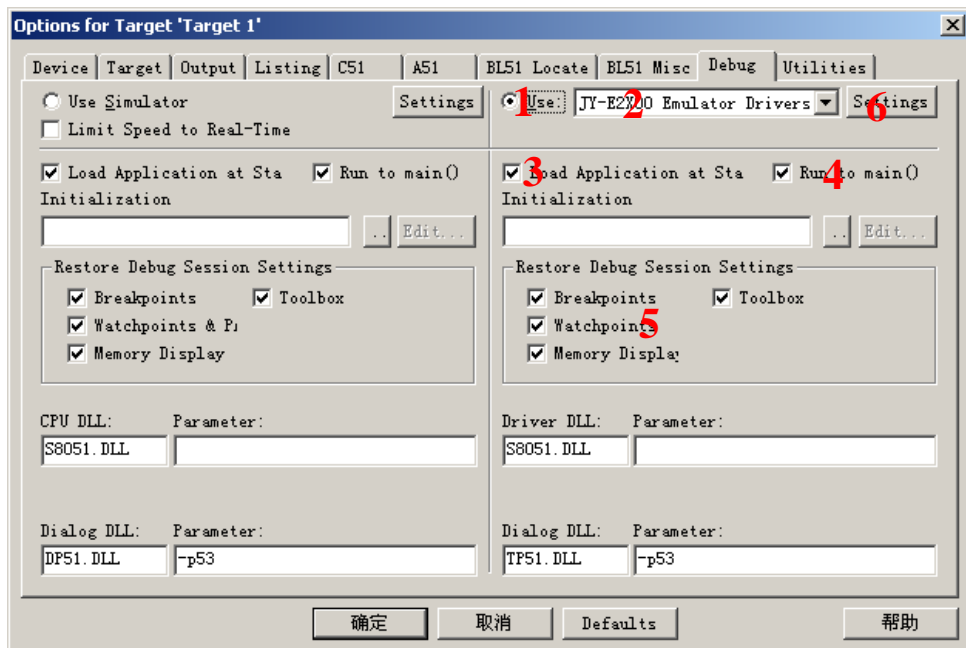


图3-6 工程设置对话框

- Restore Debug Session Settings选项区域（图中5）：IDE会保存所选取内容到下次打开工程。
  - Breakpoints：记忆当前设置的断点，下次进入仿真该断点设置存在并有效。
  - Watchpoints：记忆当前设置的观察项目，下次进入仿真仍有效。
  - Memory Display：记忆当前存储器区域的显示，下次进入仿真仍有效。
  - Toolbox：记忆当前的工具栏设置，下次进入仿真仍有效。

**第三步：**单击“Settings”按钮（图中6），进入“硬件仿真参数设置”对话框(在调试模式下，可单击菜单Peripherals -> JY-E2X00 Settings)，根据用户实际需要进行设置。

“硬件仿真参数设置”对话框如图3-7所示。

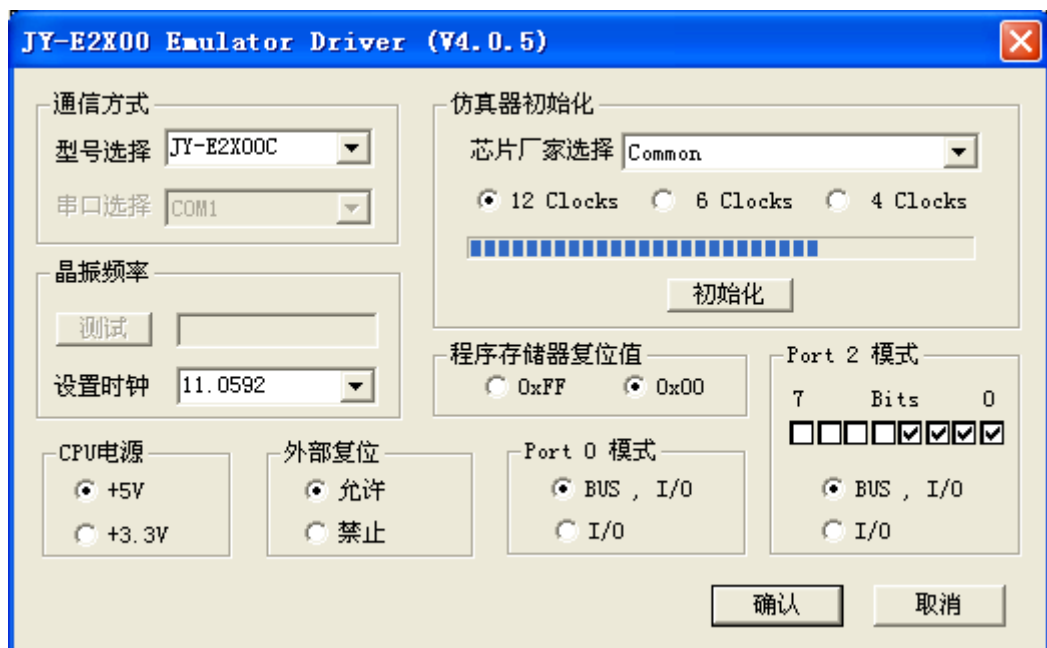


图3-7 硬件仿真参数设置

- 通信方式
  - 型号选择：选择所使用的仿真器的型号：“JY-E2300C”。

- 晶振频率，设置时钟可从下拉列表中选择
  - CPU 电源
    - +5V：使用+5V电压进行仿真。
    - +3.3V：使用+3.3V电压进行仿真。
  - 外部复位
    - 允许：允许外部复位。
    - 禁止：禁止外部复位。
  - 程序存储器复位值，可选择 0x00 或 0xFF。
  - P0口模式
    - BUS、I/O：仿真时P0口可同时用作总线、I/O。
    - I/O：仿真时P0口只用作I/O，即使访问片外RAM时，P0口也不会输出地址信号和数据信号。
  - P2口模式
    - 仿真时，P2口各位根据需要可全部或部分用作总线或I/O。
    - I/O：仿真时P2口只用作I/O，即使访问片外RAM时，P2口也不会输出地址信号。
  - 仿真器初始化
- 在下面三种情形下需要初始化仿真器：
- 改变了被仿真单片机所属的厂家；
  - 改变了被仿真单片机的时钟周期；
  - 安装了新版本的驱动程序。

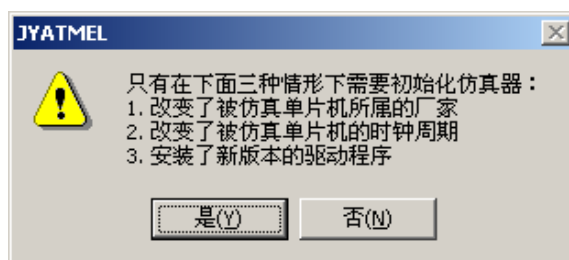


图3-8 仿真器初始化条件

点击“开始”按钮，对用户所选择的仿真器进行初始化，并弹出图3-8所示的“仿真器进行初始化”消息框。若需对仿真器进行初始化，则点击“是”按钮，仿真器开始初始化，在进度条中会显示进度值，仿真器初始化成功后会弹出“仿真器初始化成功”消息框。

#### 第四步：保存硬件仿真设置

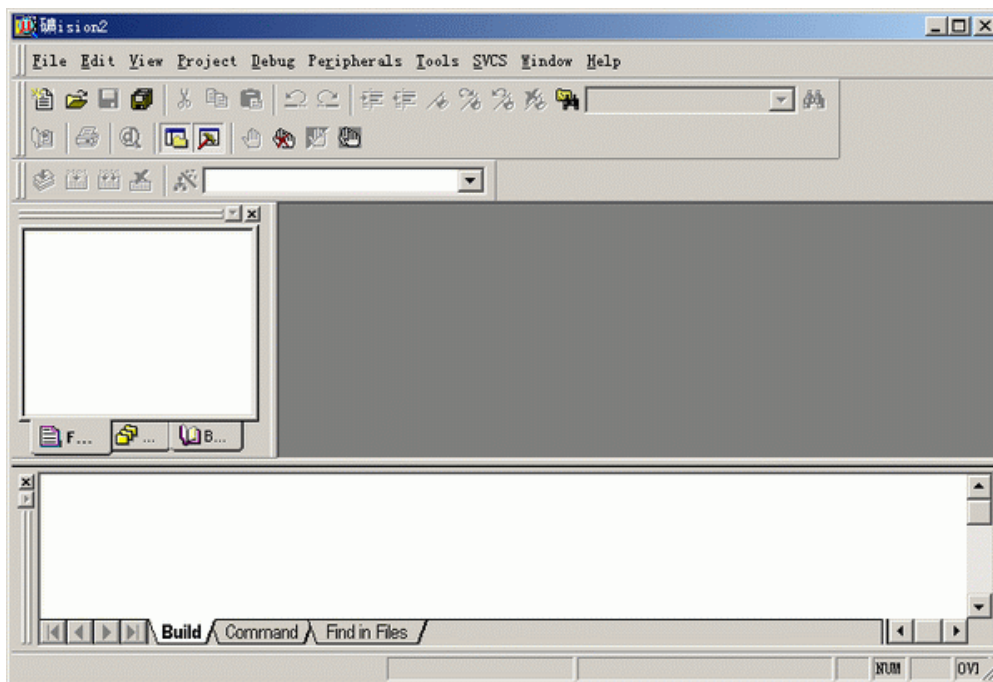
- 设置完硬件仿真参数后，点击“确认”按钮，保存硬件仿真参数并返回工程设置对话框。
- 点击“确定”按钮，返回Keil主界面。
- 单击菜单“File->Save all”，保存当前工程。

### 3.3 使用示例

在 Keil  $\mu$  Vision2 IDE 中如何使用 JY-E2X00C 仿真器进行硬件仿真呢？让我们通过下面的例子来学习。

在进入硬件仿真前，需要将JY-E2X00C仿真器按2.2节的要领进行硬件连接。

首先启动Keil  $\mu$  Vision2程序，首次进入 Keil  $\mu$  Vision2的编辑界面如图3-9所示，否则，会打开用户前一次处理的工程。

图 3-9 首次进入 Keil  $\mu$  Vision2 的编辑界面

下面通过简单的编程、调试，引导大家学习 Keil  $\mu$  Vision2 软件中 JY-E2X00C 仿真器的基本使用方法和基本调试技巧。

### 3.3.1 工程的建立

单击“Project→New Project...”菜单，弹出创建新工程对话框，如图 3-10 所示。

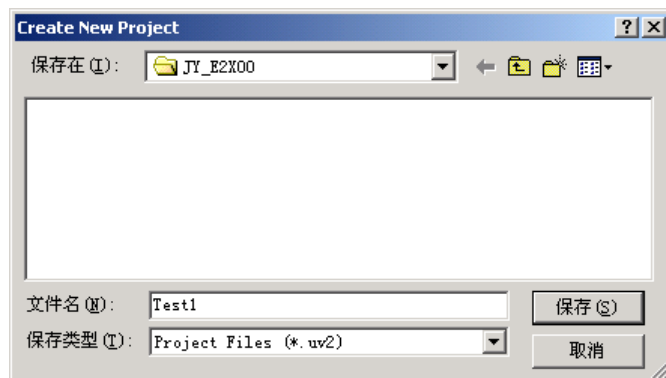


图 3-10 创建新工程对话框

选择你要保存的路径，输入工程文件的名字，不需要输入扩展名。比如保存到 JY\_E2X00 目录里，工程文件的名字为 Test1，如图 3-10 所示，然后点击“保存”，保存后的文件扩展名为 .uv2，这是 Keil  $\mu$  Vision2 项目文件扩展名。以后我们可以直接点击此文件来打开已创建的工程。

这时会弹出一个对话框，要求选择目标 CPU（即用户所用单片机的型号），Keil  $\mu$  Vision 几乎支持所有的 51 内核的单片机，我们以 AT89S52 芯片为例，如图 3-11 所示，在左侧的 Data base 列表框中点击 Atmel 前面的“+”号，展开该层，选中 AT89S52，在其右边的 Description 显示区域中是对这个单片机的基本描述，然后再点击“确定”按钮，回到主窗口。

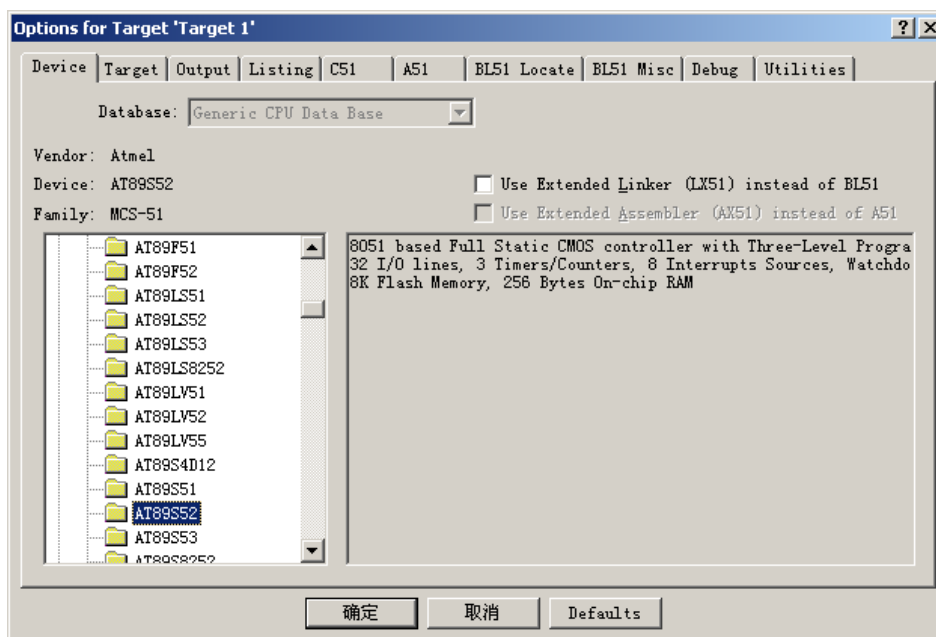


图 3-11 为工程选择目标 CPU

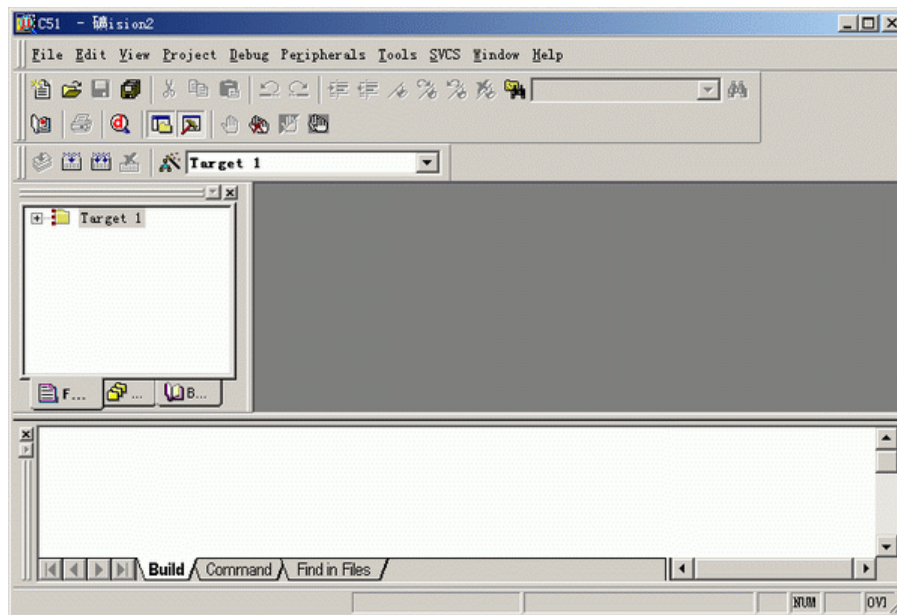


图 3-12 新建立的工程

此时，在工程窗口的文件页中，出现了“Target 1”，前面有“+”号，点击“+”号展开，可以看到下一层的“Source Group1”，这时的工程还是一个空的工程，里面什么文件也没有，需要为这个工程添加文件。如图 3-12 所示。

### 3.3.2 源文件的添加

如果用户没有现成的程序，那么就要新建一个程序文件。在 Keil 中有一些程序代码，在这里我们以一个 C 程序为例，介绍如何新建一个 C 程序和如何将它添加到工程中。单击“File->New”菜单，如图 3-13 所示，新建一个源文件。

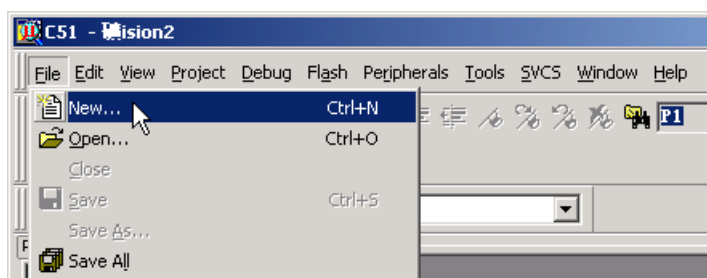


图 3-13 源程序的建立

此时在项目窗口的右侧打开一个新的文本编辑窗口，如图 3-14 所示：

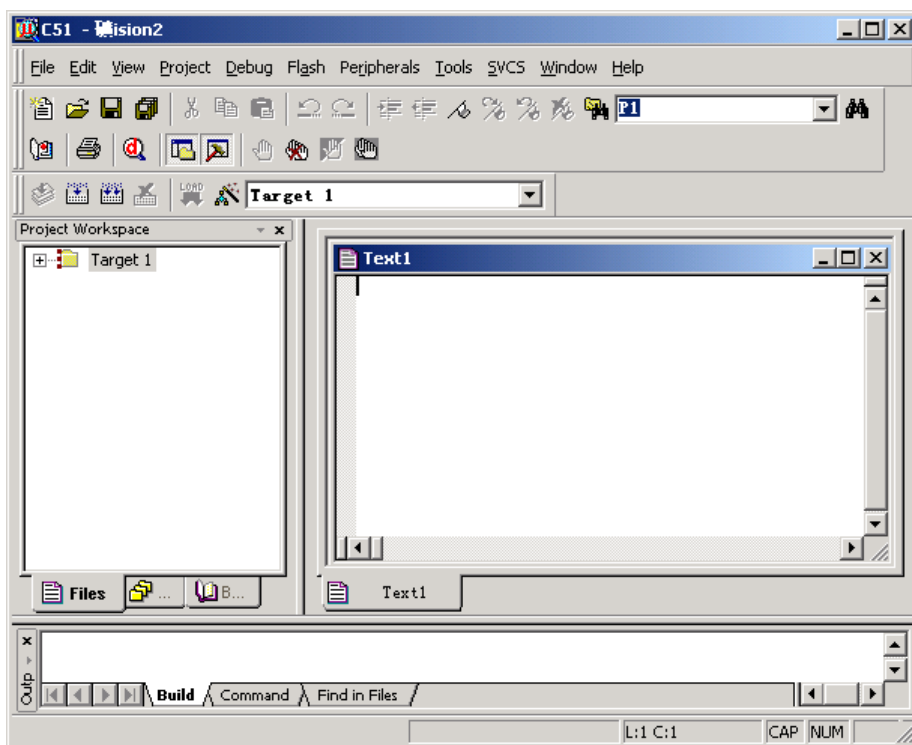


图 3-14 新建文件后的屏幕

此时光标在编辑窗口里闪烁，可以键入用户的应用程序了。但我们建议首先保存该空白的文件，单击菜单上的“File”，在下拉菜单中选中“Save As”选项单击，屏幕如下图所示，在“文件名”栏右侧的编辑框中，键入欲使用的文件名，注意，必须键入正确的扩展名。如果用 C 语言编写程序，则扩展名为(.c)；如果用汇编语言编写程序，则扩展名为(.asm 或 .a51)。然后，单击“保存”按钮。

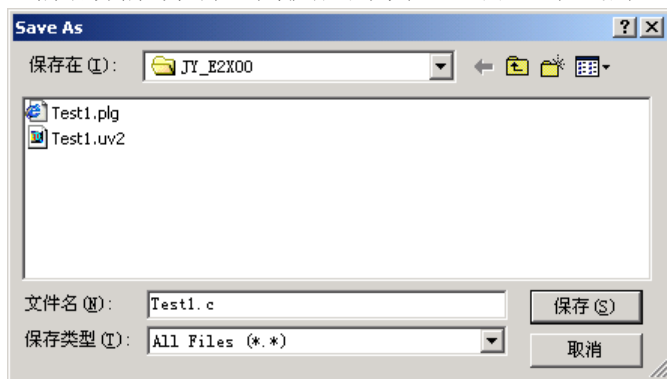


图 3-15 保存源文件对话框

在创建源文件后，就可将这个文件添加到工程中。回到编辑界面，在“Project Windows ->Files”页中，单击“Target 1”前面的“+”号，然后在“Source Group 1”上单击右键，弹出如图 3-16

所示的下拉菜单。

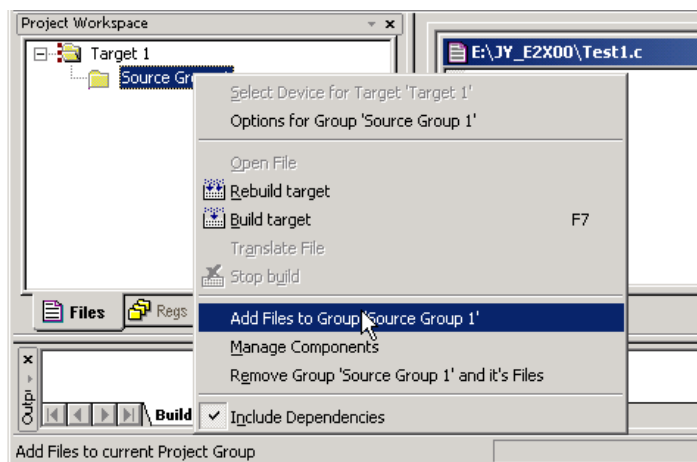


图 3-16 文件组快捷菜单

选择“Add File to Group ‘Source Group 1’”，出现如图 3-17 所示对话框。

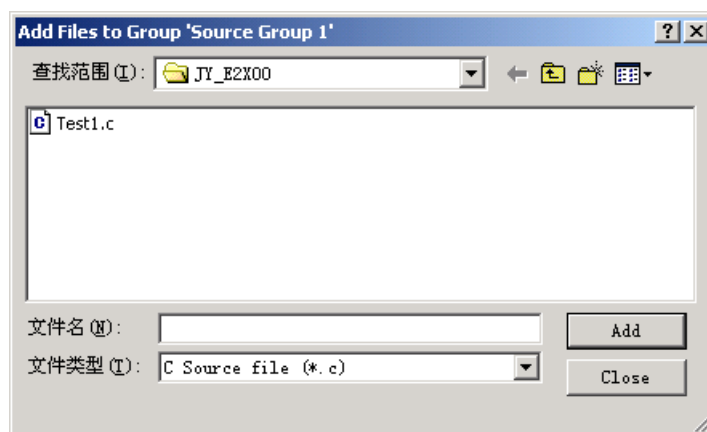


图 3-17 加入文件对话框

双击 Test1.c 或选中 Test1.c 后单击“Add”按钮，将文件加入到工程中，然后单击“Close”按钮即可返回主界面，这时在 Source Group1 文件夹图标左边出现了一个小“+”号，说明文件组中有了文件，点击它可以展开查看到文件“Test1.c”已加入其中，双击文件名，即打开该源程序，如图 3-18 所示。

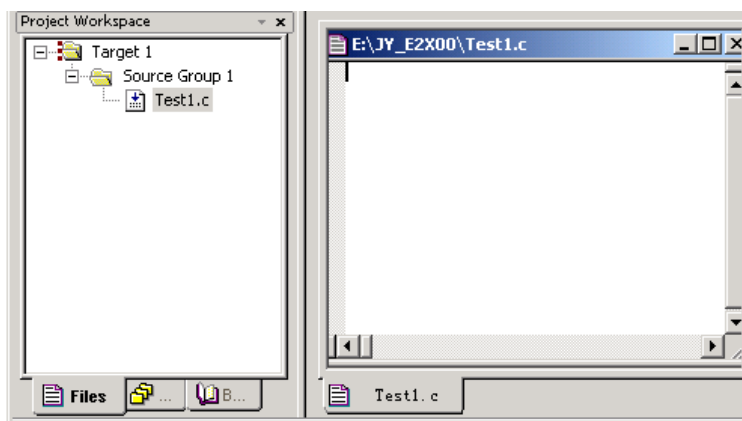


图 3-18 加入了文件的工程



注意：Keil 默认加入 C 源文件，因此，在图 3-16 中未显示该文件夹下的汇编

文件，如果要加入汇编语言源文件，需单击“文件类型”下拉列表框右侧的下三角按钮，弹出下拉列表，

选择“asm Source file(\*.a\*;\*.src)”，这时才会将文件夹下的“\*.asm”文件显示出来；

现在，请输入如下的 C 语言源程序：

```
#include "reg51.h"
#include "intrins.h"
#define uchar unsigned char
#define uint unsigned int

//延时程序
void mDelay(unsigned int Delay)
{ unsigned int i;
  for(;Delay>0;Delay--)
  { for(i=0;i<124;i++)
    {;}
  }
}

void main()
{ unsigned char OutData=0xfe;
  for(;;)
  {
    P1=OutData;
    OutData=_crol_(OutData, 1);    //循环左移
    mDelay(1000);                  /*延时 1000 毫秒*/
  }
}
```

在输入上述程序时，我们就可以体会到事先保存待编辑的文件的好处了，即 Keil  $\mu$  Vision 会自动识别关键字，并以不同的颜色提示用户加以注意，这样会使用户少犯错误，有利于提高编程效率。程序输入完毕后，如图 3-19 所示。



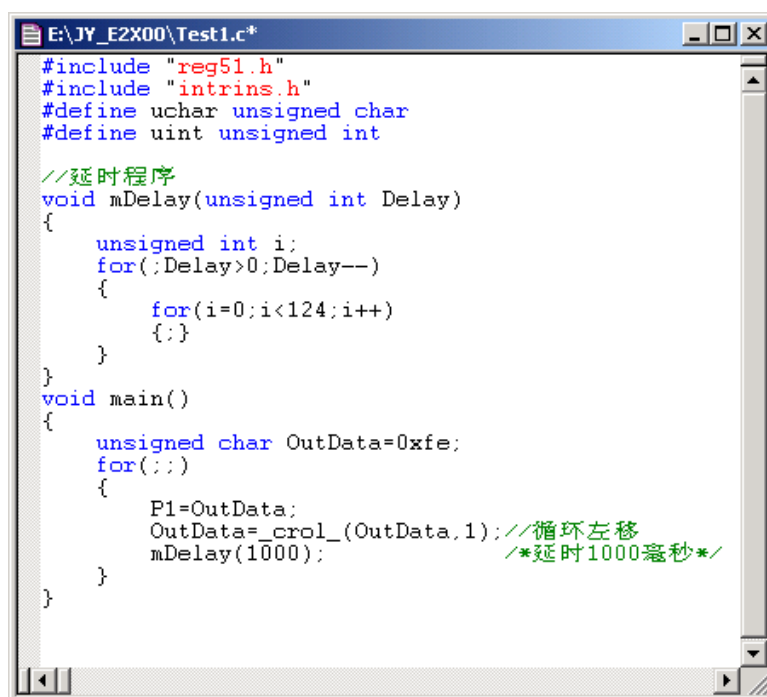


图 3-19 编辑源程序

### 3.3.3 工程的设置

工程建好后，还要对工程进行进一步的设置，才能使用 JY-E2X00C 仿真器进行硬件仿真。有三种打开工程设置对话框的方法：

A) 点击工具栏上的  按钮，如图 3-20 所示。



图 3-20 使用工具栏图标按钮打开工程设置对话框

B) 单击工程窗口下的 Target1，使其高亮显示，再点击菜单“Project-> Options for Target ‘target1’ ”

C) 鼠标右键点击工程窗口下的“Target1”，出现一个下拉菜单，如图 3-20 所示，选中“Options for Target ‘target1’ ”。

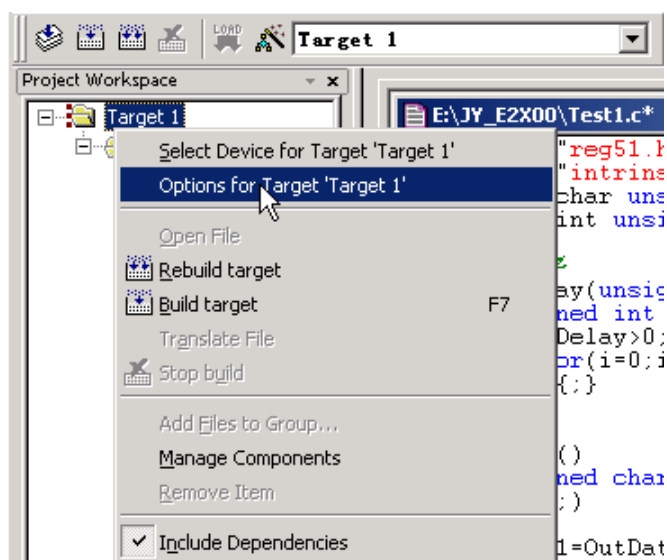


图 3-21 使用下拉菜单打开工程设置对话框

打开的工程设置对话框如图 3-22 所示。

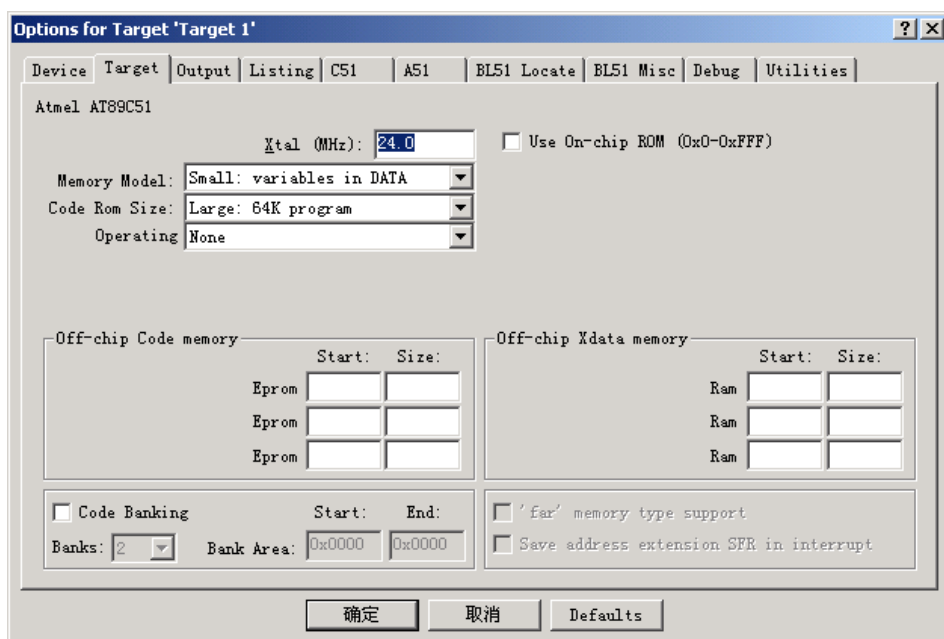


图 3-22 Target 选项对话框

单击“Output”标签进入输出设置页面卡，如图 3-23 所示，选中“Create Hex file”（图中 1），使程序编译后产生 HEX 代码，供编程器使用。同时也可选中“Debug Information”（图中 2）和“Browse Information”（图中 3），生成 Hex 代码文件调试信息和浏览信息，点击“确定”退出。

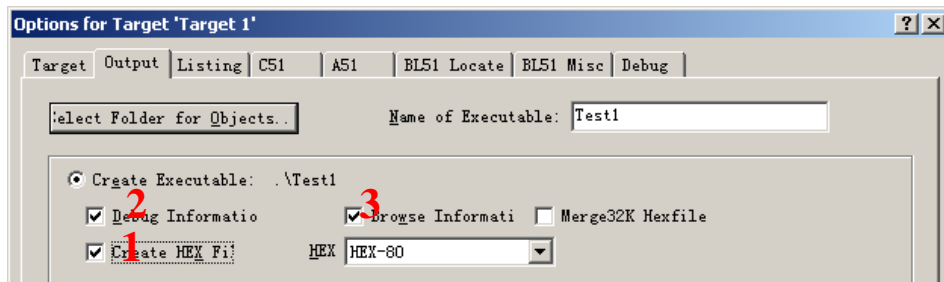


图 3-23 Output 选项对话框

单击“Debug”标签进入调试器设置页面卡，按图 3-6 所示进行设置。

单击“Settings”按钮，进入“硬件仿真参数设置”对话框，按图3-7所示进行设置。

点击“测试”按钮，对用户所选择的仿真晶振进行测试。

如果出现通信失败无法进入硬件仿真，请检查以下因素：

- 仿真器与 PC 是否正常连接
- 仿真器型号是否选择正确，或被其它程序占用，或者已经损坏

设置正确后，点击“确认”按钮，保存硬件仿真参数并返回图 3-6 所示的工程设置对话框，点击“确定”按钮，返回 Keil 主界面，单击菜单“File->Save all”，保存当前工程。

### 3.3.4 编译、链接

设置好工程后，即可进行编译、链接。图 3-24 所示的是有关编译的工具栏按钮和菜单，图中 1、2、3 都是编译按钮，1 是编译单个文件。2 是编译当前项目，如果先前编译过一次之后文件没有被编辑、改动，这时再点击是不会重新编译的。3 是重新编译，每点击一次均会再次编译链接，不管文件是否有改动。在 3 右边的是停止编译按钮，只有点击了前三个中的任一个，停止按钮才会生效。4 是信息输出窗口。5 是菜单的相应操作。这个项目只有一个文件，你按 1、2、3 中的任一个都可以编译。6 是有一个小放大镜的按钮，这就是开启\关闭调试模式的按钮，它也存在于菜单“Debug—Start\Stop Debug Session”中，快捷键为“Ctrl+F5”。

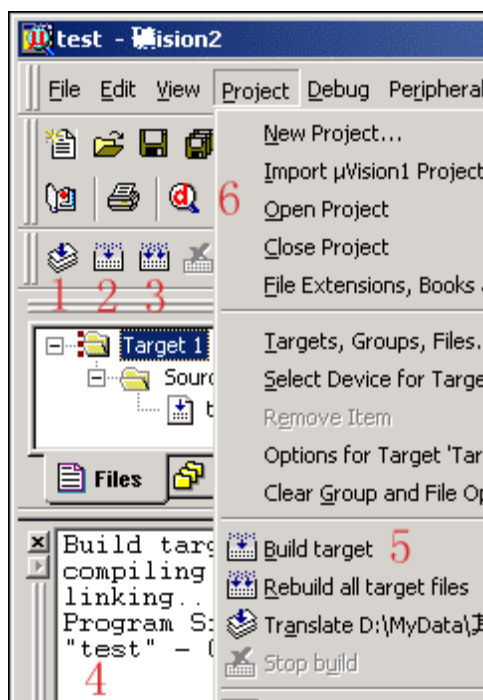


图 3-24 有关编译的工具栏按钮和菜单

按 1、2、3 中的任意一个进行编译。在 4 中的 Build 页可以看到编译的错误信息和使用的系统资源情况等，以后我们要查错就靠它了，如果用户程序和工程设置没有错误，编译和链接将能顺利完成。如图 3-24 所示，提示获得了名为 Test1.hex 的文件，该文件即可被编程器读入并写到芯片中，同时还可见到，该程序的代码量（code=71），内部 RAM 的使用量（data=9），外部 RAM 的使用量（xdata=0）等一些信息。除此之外，编译、链接还产生了一些其它相关的文件，可被用于 Keil 的仿真与调试，这时可进入下一步的调试。

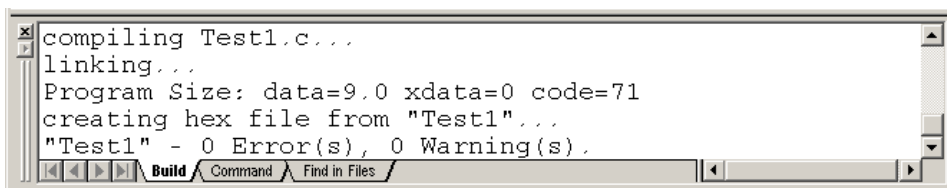


图 3-25 编译和链接顺利完成

我们前面的程序经过验证后没有错误，但是用户在开始编写程序时容易出现错误，现在我们故意将“P1=OutData”改为“P1=OutDate”，我们再次进行编译和链接编译/链接，因源程序中有语法错误，在信息输出窗口中有错误报告出现：“TEST1.C (21): error C202: ‘OutDate’: undefined identifier”（在文件 TEST1.C 的 21 行，出现 C202 类型的错误，OutDate 没有定义）。双击该行，定位到出错的位置，如图 3-26 所示，将错误改正后，再进行编译，程序编译和链接能顺利完成。

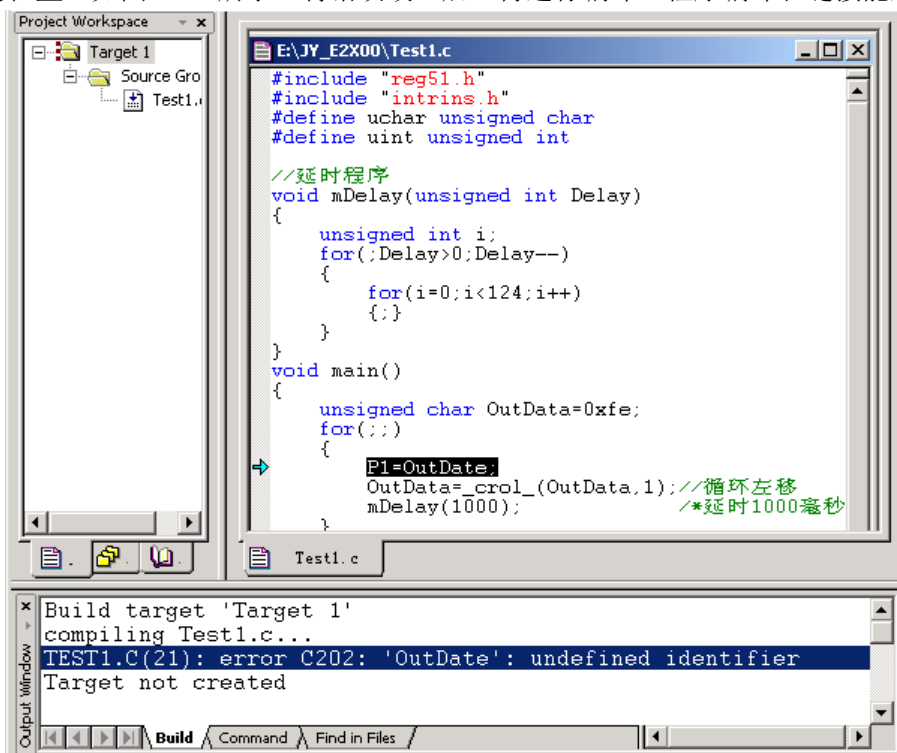


图 3-26 编译后的错误定位

### 3.3.5 进入仿真

编译成功后，启动调试模式，进入仿真状态。启动调试模式有以下几种方法：

- ◆ 点击工具栏上的  按钮
- ◆ 单击“Debug->Start/Stop Debug Session”
- ◆ 使用快捷键“Ctrl+F5”

启动调试模式后，代码便开始下载到仿真器中，如图 3-27 所示。

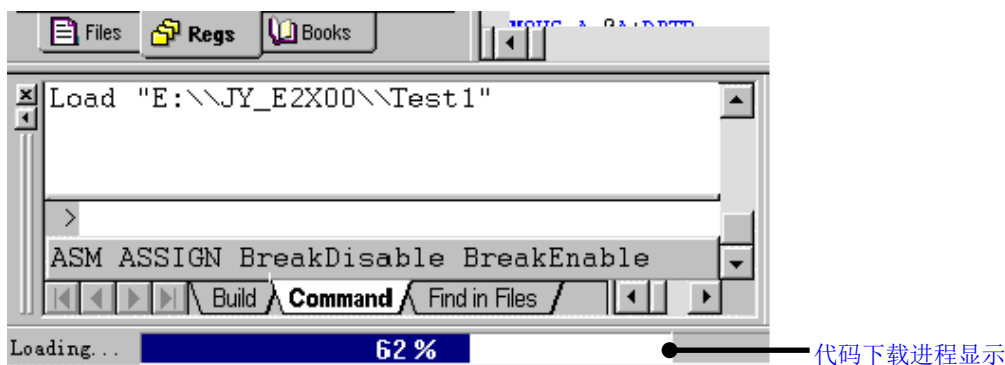


图 3-27 代码下载到仿真器中

代码下载完后，如图 3-28 所示，用户可以根据需要进行单步、断点、全速运行，观察 / 修改变量值、观察 / 修改存储区数据等操作。

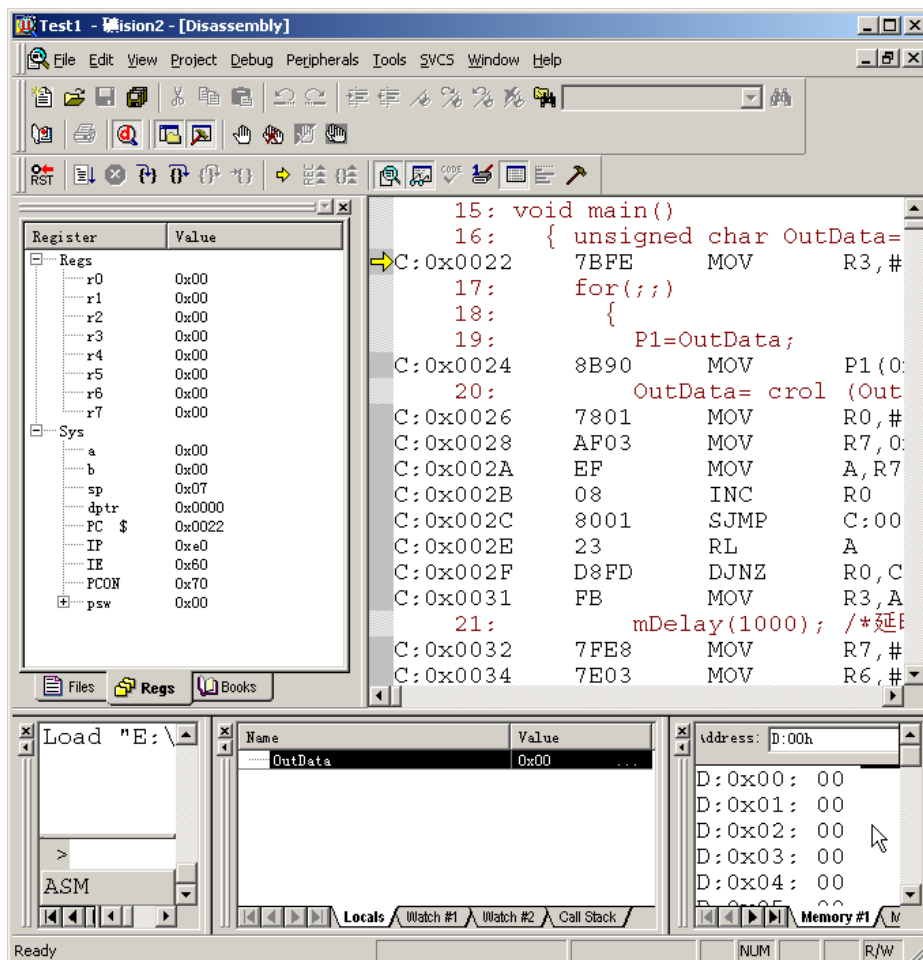



图 3-28 代码下载完毕

进入调试状态后，Debug 菜单项中的命令可以使用了，有关编译的工具栏按钮消失了，出现了一个用于运行和调试的工具栏，Debug 菜单上的大部份命令都有相应的快捷按钮，如图 3-29 所示，从左到右依次是复位、运行、暂停、单步跟踪、单步、执行完当前子程序、运行到当前行、下一状态、打开跟踪、观察跟踪、反汇编窗口、观察窗口、代码作用范围分析、1# 串行窗口、内存窗口、性能分析、工具按钮等命令；



图 3-29 运行和调试的工具栏

点击工具栏上的按钮，将反汇编窗口关闭，可看到源程序窗口运行指针处于 main() 函数下的第一个有效的执行程序行，如图 3-30 所示，这是由于我们在工程设置中选择了 Go till main。

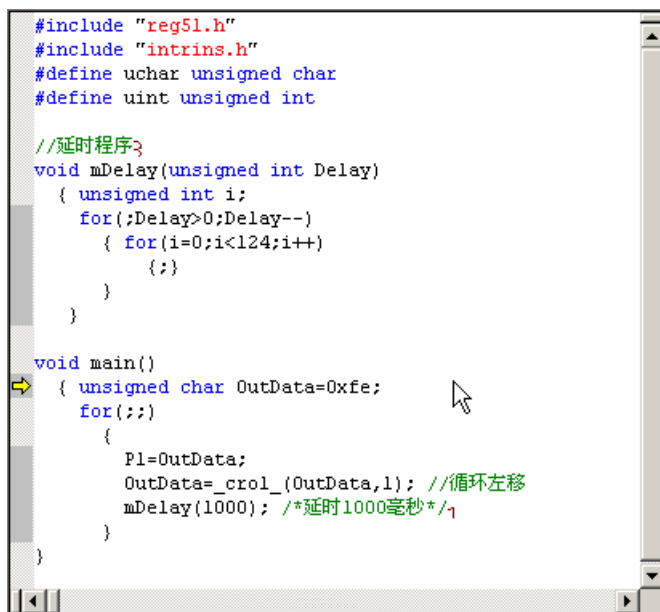




图 3-30 进入硬件仿真环境后的源程序窗口

### 3.3.6 调试程序

#### 1. 单步跟踪运行

使用菜单 Debug->Step 或相应的命令按钮或使用快捷键 F11 可以单步跟踪执行程序，在这里我们按下 F11 键，即可执行该箭头所指程序行，每按一次 F11，可以看到源程序窗口的左边黄色调试箭头指向下一行，当箭头指向“mDelay(1000);”行时，再次按下 F11，会发现，箭头指向了延时子程序 mDelay 的第一行，不断按 F11 键，即可逐步执行延时子程序，如图 3-31 所示，整个延时子程序大约要单步执行 1000x125 次才能完成。

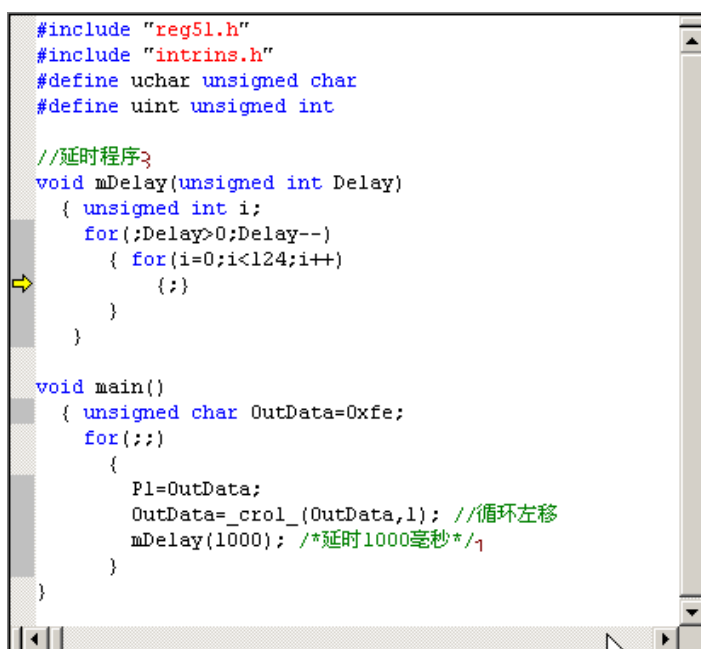




图 3-31 单步跟踪运行

## 2. 单步运行

如果 mDelay 程序有错误，可以通过单步跟踪执行来查找错误，但是如果 mDelay 程序已正确，每次进行程序调试都要反复执行这些程序行，会使得调试效率很低，为此，可以在调试时使用 F10 来替代 F11（也可使用菜单 Step Over 或相应的命令按钮），在 main 函数中执行到 mDelay(1000)时将该行作为一条语句快速执行完毕。

为了更好的进行对比，我们重新进入仿真环境，将反汇编窗口关闭，不断按 F10 键，可以看到在源程序窗口中的左边黄色调试箭头不会进入到延时子程序。

## 3. 全速运行



点击工具栏上的按钮或按 F5 键启动全速运行，全速执行程序，此时用户板上的 P1 口所接 LED 以流水灯状态显示，此时工具栏如图 3-32 所示。



图 3-32 全速运行时的运行和调试工具栏

## 4. 暂停

点击工具栏上的按钮，此时用户板上的 P1 口所接 LED 停止以流水灯状态显示，只有一个 LED 灯点亮（取决于暂停前的 P1 的值）。

## 5. 观察 / 修改寄存器的值

Project 窗口在进入调试状态后显示 Regs 页的内容，如图 3-33 所示，包括工作寄存器 R0~R7 的内容和累加器 A、寄存器 B、堆栈指针 SP 的内容。

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0xf4
r4	0x00
r5	0x75
r6	0x00
r7	0xaf
Sys	
a	0x08
b	0x00
sp	0x09
sp_max	0x09
dptr	0x0000
PC \$	C:0x0800
states	258377291
sec	129.18864550
psw	0x01

图 3-33 查看寄存器的值

用户除了可以观察以外还可自行修改，例如将寄存器 a 的值 0x62 改为 0x85，  
方法一：用鼠标点击选中单元 a，然后再单击其数值位置，出现文字框后输入 0x85 按回车键即可；



方法二：在命令行窗口，输入 A=0x85，按回车键将把 A 的数值设置为 0x85。

## 6. 观察 / 修改存储器的数据

### ◆ 通过 Memory 窗口查看数据

点击菜单“View->Memory Windows”，便会打开 Memory 窗口(如果窗口已打开，则窗口会关闭)，如图 3-34 所示，Memory 窗口可以同时显示 4 个不同的存储器区域，点击窗口下部分的编号可以相互切换显示。

在 Memory#1 的地址输入栏内输入“D:0e0h”，按回车键后，可以从内部可直接寻址 RAM 的 e0H 地址处开始显示，e0H 地址的值就是寄存器 a 的值，应与主寄存器窗口下的值相同。

点击窗口下部分的 Memory#2，在 Address 输入栏内输入“I:0e0h”，按回车键后，可以从内部可间接寻址 RAM 的 e0H 地址处开始显示。

点击窗口下部分的 Memory#3，在 Address 输入栏内输入“C:0x0021”，按回车键后，可以从代码区域 0021H 地址处开始显示，这时各地址值应与在反汇编窗口中的值相同。

点击窗口下部分的 Memory#4，在 Address 输入栏内输入“X:00h”，按回车键后，可以从 xdata RAM 区域 0000H 地址处开始显示。

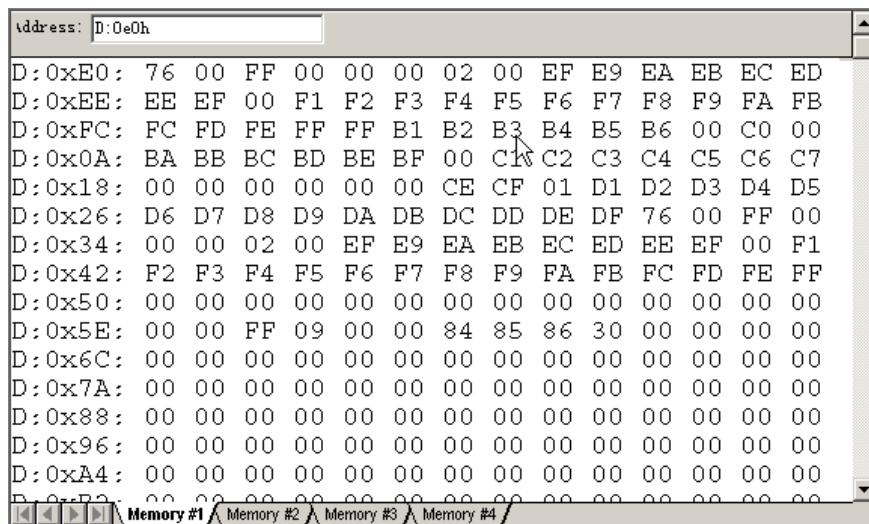


图 3-34 通过 Memory 窗口查看数据

### ◆ 通过 Memory 窗口修改数据

在 Memory 窗口中显示的数据可以修改。例如，要改动 data 区域 0xE0 地址的数据内容，修改方法如下：

把鼠标移动到该数据的显示位置，按动鼠标右键在弹出的菜单中选中“Modify Memory at D:0xE0”，在弹出对话框的文本输入栏内输入相应数值，按回车键或点击 OK，修改完成。



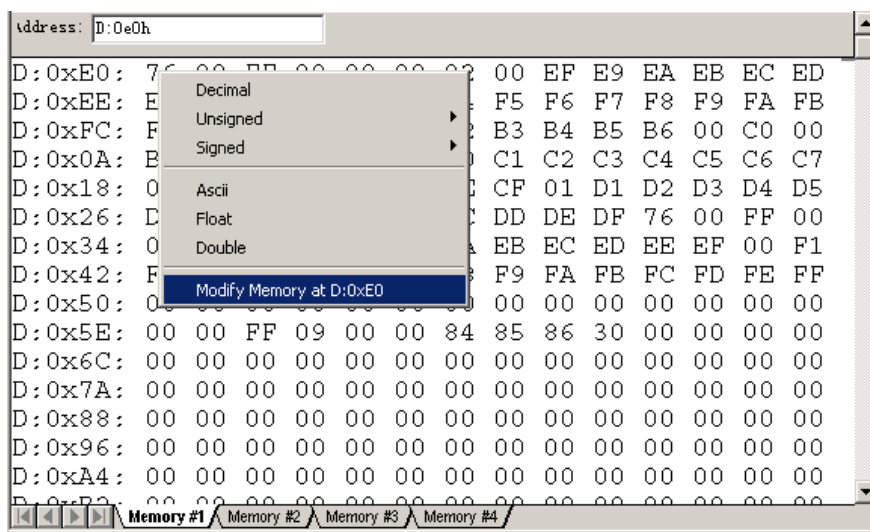


图 3-35 Memory 窗口右键弹出菜单

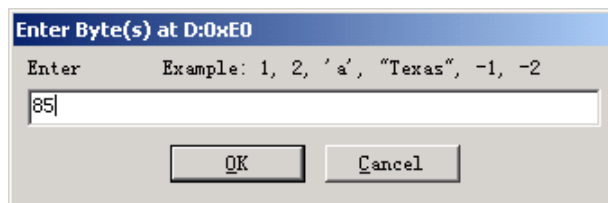


图 3-36 通过 Memory 窗口修改数据

◆ 通过命令行查看数据

例如, 我们想查看 data 空间从 0x01 到 0x03 地址的内容, 可在命令输入窗口输入“d d:0x01, 03H”, 回车即可完成, d 表示 data 空间, 0x01 表示起始地址, 03H 表示结束地址 (注意两种 16 进制的表示方法在这里都可以接受), 输出结果在信息输出窗口中显示出来, 如图 3-37 所示。

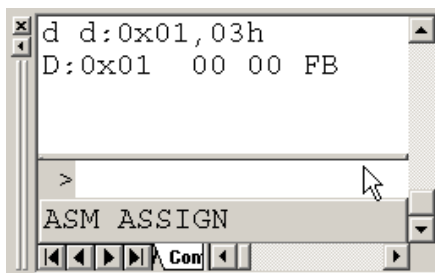


图 3-37 通过命令行查看数据

◆ 通过命令行修改数据

例如, 我们想把 data 空间从 0x01 开始的 3 个数据修改成 0x01 0x02 0x03, 可在命令输入窗口输入“E d:0x01=0x01, 0x02, 0x03”, 如图 3-38 所示, 回车即可完成, E 表示修改数据, d: 表示 data 空间, = 后面是输入的修改数据。

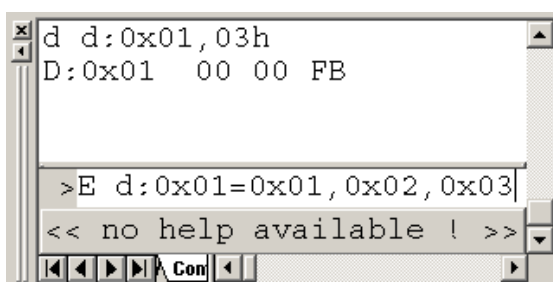


图 3-38 通过命令行修改数据

通过 Memory 窗口可以看到修改后的数据，如图 3-39 所示。

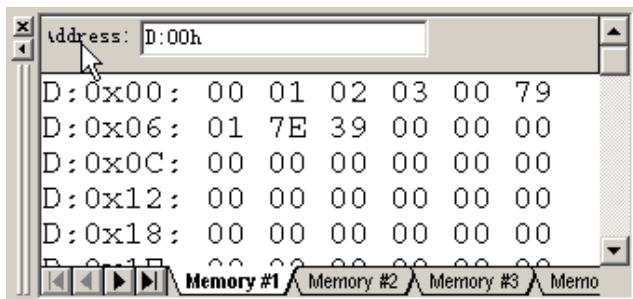


图 3-39 查看修改后的数据

## 7. 观察 / 修改变量的值

在暂停程序运行时，可以观察到有关的变量值。

在 Watch 窗口 Local 页自动显示当前正在使用的局部变量，不需要用户自己添加。Watch 页显示用户指定的程序变量，如图 3-40 所示。

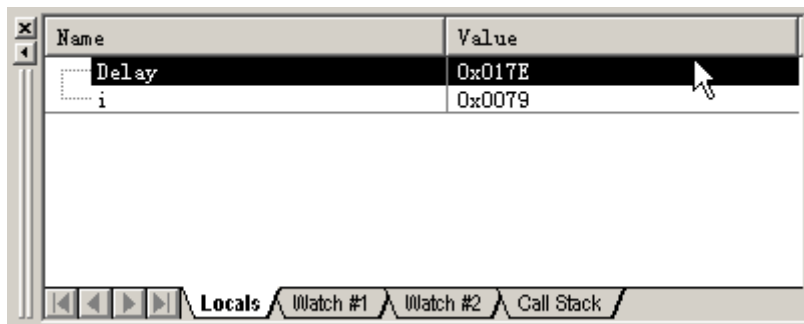


图 3-40 Watch 窗口 Local 页自动显示当前正在使用的局部变量

移动鼠标光标到要观察的变量“OutData”上停留大约一秒钟，就弹出一个“变量提示”块出来，如图 4-41 所示。

```
void main()
{
    unsigned char OutData=0xfe;
    for(;;)
    {
        P1=OutData;
        OutData=(OutData<<1); //循环左移
        mDelay(1000); /*延时1000毫秒*/
    }
}
```

图 3-41 点屏显示变量

将鼠标移动到一个变量名“OutData”的上面，点击鼠标右键，出现快捷菜单，选中“Add “OutData” to Watch Windows...”选项，子菜单中会出现#1 和#2 的选项，如图 3-42 所示，点击后该变量就会加入对应的 Watch 窗口，如图 3-43 所示。

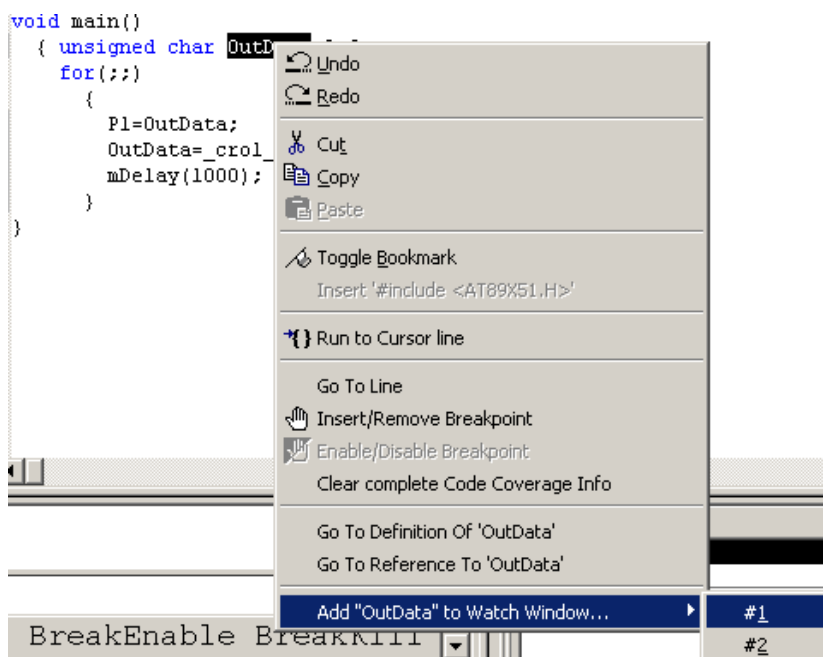


图 3-42 添加用户指定的程序变量

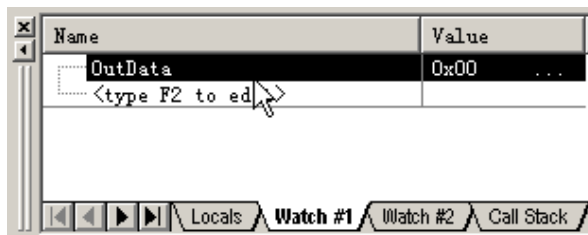



图 3-43 查看添加的程序变量

修改变量的数值方法是：用鼠标左键点击该行的变量数据栏，然后按 F2 键出现文本输入栏后，输入修改的数据，确认正确后按回车键。

## 8. 复位

如果用户想重新开始运行用户程序，可以点击工具栏上的  按钮，对仿真器的用户程序进行复位。仿真器复位后，程序计数器 PC 指针将复位成 0000H，另外，一些内部特殊功能寄存器在复位期间也将重新赋值，例如 A 将变为 00H，DPTR 变为 0000H，SP 变为 07H，I/O 口变为 0FFH，如图 3-44 所示，此时用户板上的 P1 口所接 LED 指示灯会全部熄灭。

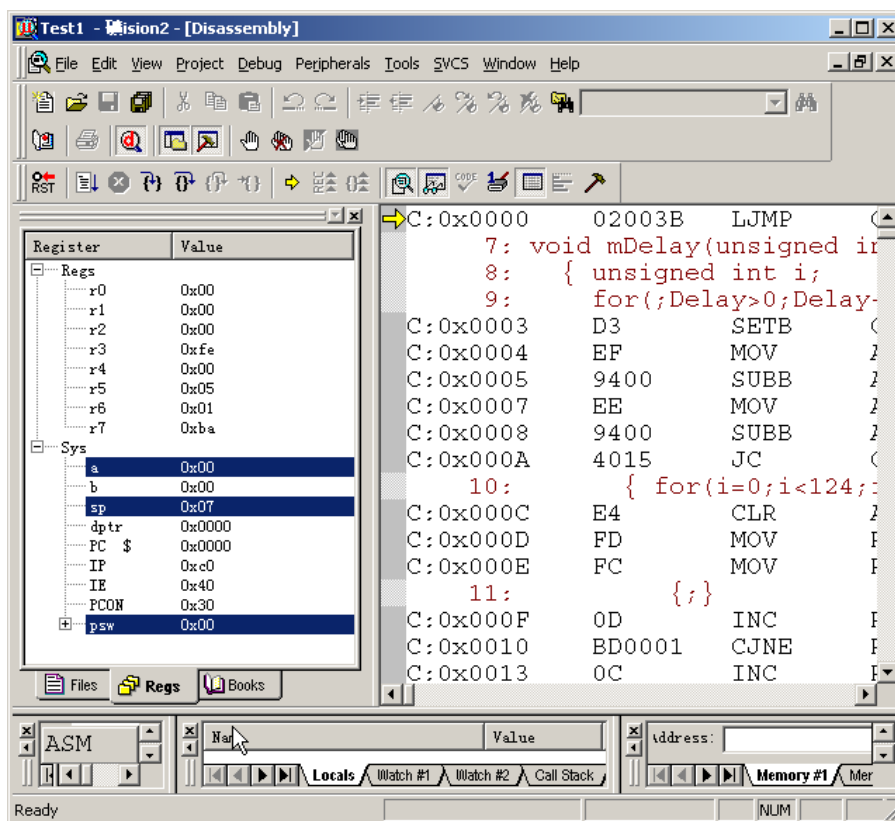




图 3-44 仿真器复位

## 9. 设置断点

将光标移至待设置断点的源程序行，如 “OutData=\_crol\_(OutData,1);” 行。点击工具栏上的  图标，可以看到源程序窗口中该行的左边出现了一个红色的断点标记 。用同样的方法在 “mDelay(1000);” 行再设定一个断点。

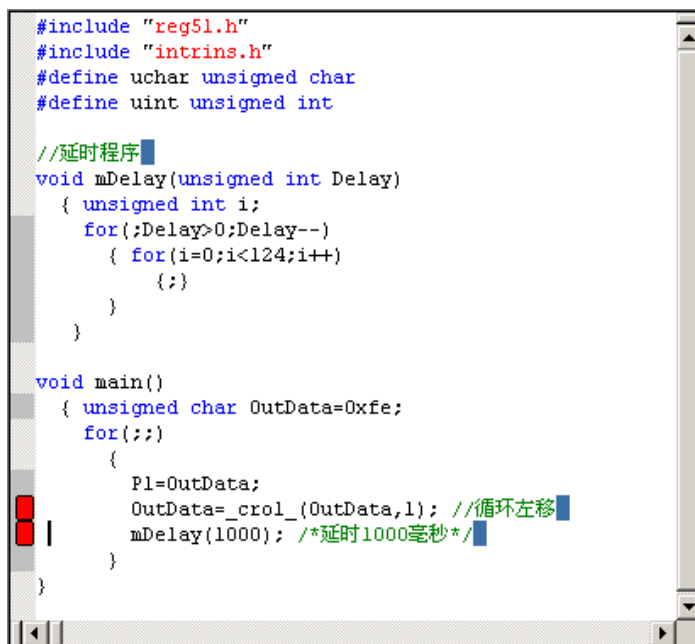



图 3-45 设定断点

## 10. 带断点的全速运行



按动 F5 启动全速运行，全速执行程序，当程序执行到第一个断点时，会暂停下来，如图 3-46 所


示,这时你可以观察程序中各变量的值及各端口的状态,此时用户板上的 P1.0 所接的 LED 灯点亮,在变量观察窗口中,OutData 的值应为 0xFE,判断程序是否正确;继续按动 F5 启动全速运行,程序执行到第二个断点时,会暂停下来,在变量观察窗口中,OutData 的值应为 0xFD;继续按动 F5 启动全速运行,程序又会执行到第一个断点处暂停,此时用户板上的 P1.1 所接 LED 灯点亮。

### 11. 清除程序中所有断点

如果想取消全部的断点全速运行时,要是逐个取消将是很烦琐的事,我们只要点击工具栏上的图标,就可清除程序中所有断点。

### 12. 执行到光标处

在体验“执行到光标处”之前,我们先点击工具栏上的图标,对仿真器的用户程序进行复位,点击工具栏上的按钮,将反汇编窗口关闭。

不断按 F11 键,逐步执行到延时子程序中,为了跳出延时子程序,我们将光标移到程序想要暂停的地方,如本例中的“OutData=\_crol\_(OutData,1);”行。选择“Debug→Run to cursor line”或 Ctrl+F10 键或弹出菜单的“Run to cursor line”功能或点击工具栏上的图标,程序全速执行到光标所在行,如图 4-46 所示,这与我们在前面看到的带断点的全速运行相类似。

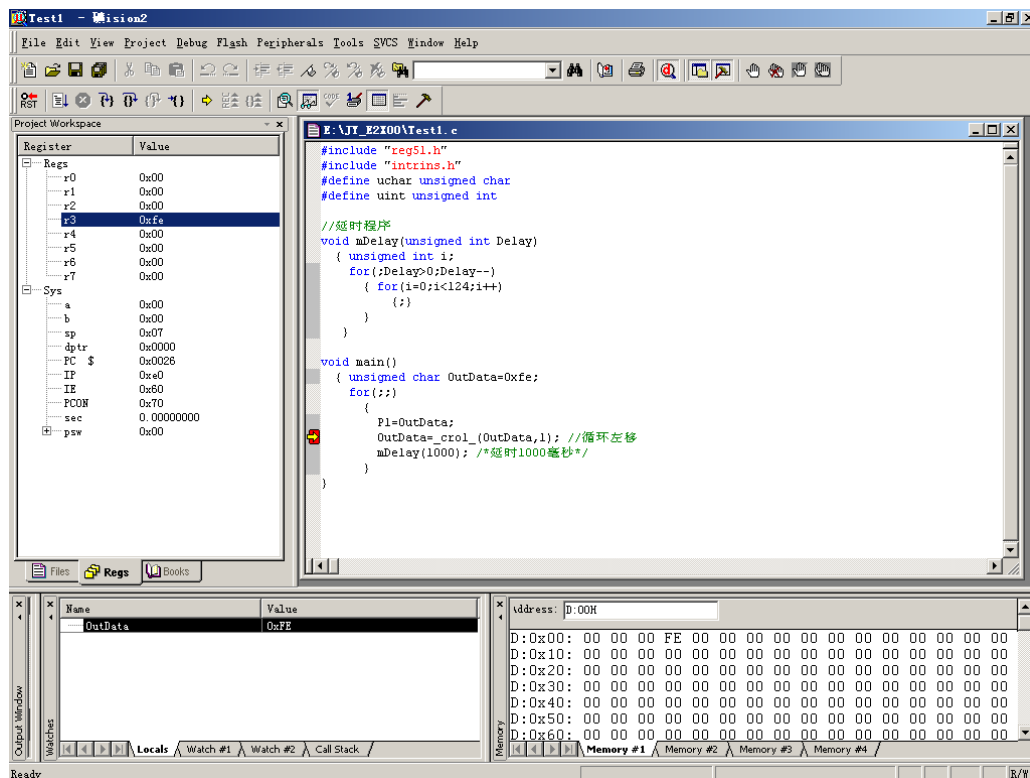


图 3-46 带断点的全速运行

## 3.3.7 退出仿真

先按停止按钮,再按开启/关闭调试模式按钮,则退出仿真状态,待回到文件编辑模式后,我们就可以进行关闭 Keil μ Vision2 IDE 等相关操作了。

## 第 4 章 菊阳单片机仿真调试软件 JY2X00 的使用

注：仿真调试软件 JY2X00 是与 Keil 调试软件配套使用的，因此使用前先确认是否已经安装了 Keil，光盘中提供了 Keil 软件的评估版。

### 4.1 安装菊阳单片机仿真调试软件 JY2X00

在 Windows 环境中，按照下列步骤安装菊阳单片机仿真调试软件 JY2X00。

1. 将安装光盘插入相应的光驱中；
2. 光盘自动运行后，如第三章图 3-1 所示，点击“安装菊阳单片机仿真调试系统”，启动菊阳单片机仿真调试软件 JY2X00 安装程序，根据提示进行安装（也可双击光盘“JY2X00”文件夹下的 setup.exe）。

### 4.2 菊阳单片机仿真调试软件 JY2X00

菊阳单片机仿真调试软件 JY2X00 是长沙菊阳微电子有限公司自主开发的软件，既可以与 JY-E2X00C 仿真器配套使用（硬件调试），又可以脱离仿真器单独使用（软件模拟调试）。

**菊阳单片机仿真调试软件 JY2X00 的特点：**

- 采用了流行的船坞化技术，结构紧凑，组合灵活，显示内容丰富；
- 采用了工程方式来管理项目相关文件，从文件的运用和管理上，确保项目开发高度有序地进行；
- 最大限度地使用了人性化设计，调试操作，信息查阅便捷高效（如在特殊功能寄存器窗口右键单击资源的数值，可查看相应的地址）。
- 调试时的窗口主要由菜单栏、工具栏、工程管理窗、数据观察窗、源程序窗、CPU 反汇编窗、其它信息窗以及状态栏等组成，如图 4-1 所示。

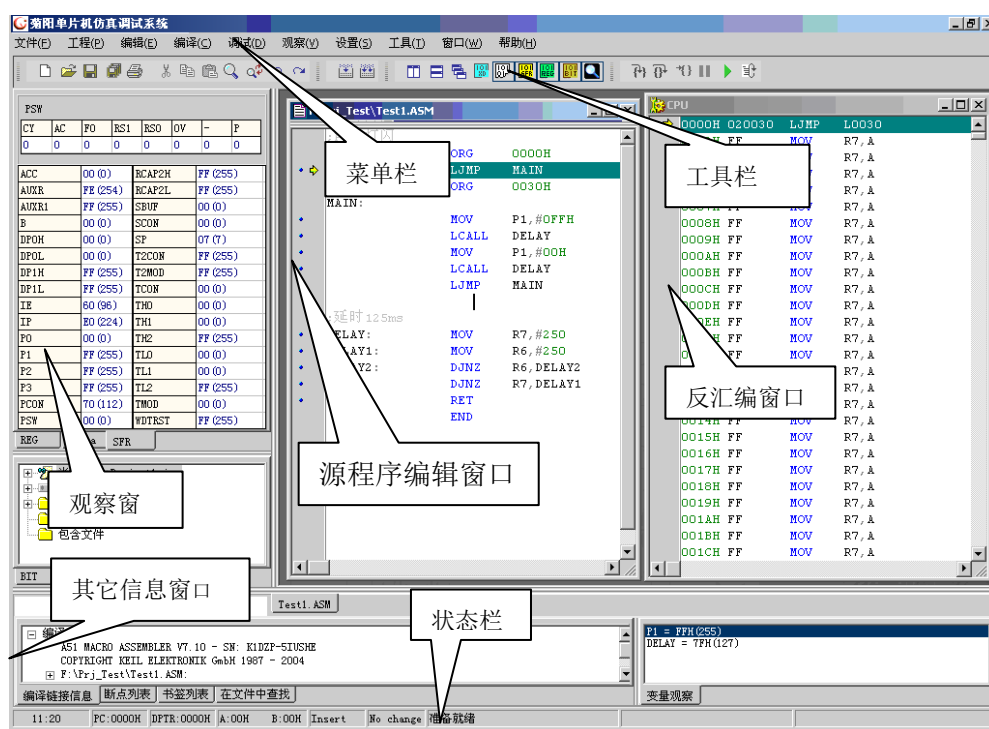


图 4-1 仿真调试软件 JY2X00 的主窗口

## 4.2.1 主菜单

主菜单包括文件、工程、编辑、编译、调试、观察、设置、工具、窗口和帮助等菜单项，如图 4-2 所示。



图 4-2 JY2X00 仿真调试器主菜单

在 JY2X00 仿真调试器中，遵循 Windows 菜单规则：

- ①使用鼠标或快捷键 Alt-菜单命令上的字符可激活菜单。如鼠标单击菜单栏中的“文件”或键入 Alt-F 即可激活“文件”菜单。激活菜单后，可移动鼠标至相应的命令上，点击鼠标左键或按菜单命令后的字符键，则可执行该菜单的动作。如激活文件菜单后，鼠标点击“新建”命令或按 N 键，执行“新建”命令。
- ②常用的菜单命令在工具栏中有该命令的快捷按钮。详见“工具栏”。
- ③菜单命令后跟有省略号 (…): 表示该命令将打开一个对话框。
- ④菜单命令后跟有三角号 (▾): 表示该菜单命令带有子菜单。
- ⑤“观察”菜单中菜单命令前有对号 (☑): 表示该窗口为数据显示状态。单击该菜单命令将在“显示”与“关闭”状态之间切换。如菜单命令“内部数据窗口”前有“☑”，表示“内部数据窗口”为数据显示状态，若此时单击该菜单命令，它前面的“☑”消失，表示“内部数据窗口”已关闭。

## 4.2.2 工具栏

工具栏除具有 Windows 常用工具外，还包含了最常用的 JY2X00 仿真调试器命令，如编译、全部重新编译、各种调试运行方式（包括单步、运行到光标处、连续运行、复位等）、各种数据窗口的操控以及文档窗口的排列等，如图 4-3 所示。



图 4-3 与编译、调试有关的工具栏

### 4.2.3 提示框

若将鼠标指向并停留在工具栏某一命令按钮上 3 秒钟，则该命令按钮下方会出现一个小提示框，显示该命令按钮的功能，如图 4-4 所示。

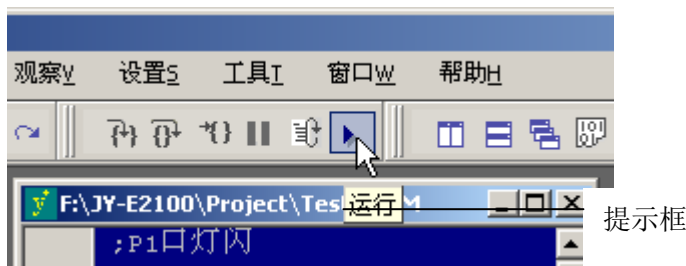


图 4-4 提示框

### 4.2.4 状态栏

位于 JY2X00 仿真调试器窗口底部的状态条显示当前正在执行的命令及其它调试信息，如光标在文档中的位置、编辑的状态、程序运行的地址、程序编译（包括链接、装载）的进度等，如图 4-5 所示。

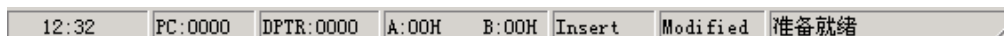


图 4-5 状态栏

### 4.2.5 对话框

JY2X00 仿真调试器绝大多数对话框为模式对话框，这种对话框出现后，只有在对话框所列各项作出若干选择时，或所键入所需信息后该命令才能继续执行下去。例如设置新的 PC 值（程序指针）对话框（程序编译后才有效），如图 4-6 所示。



图 4-6 设置新的 PC 值对话框



## 4.2.6 快捷键

按快捷键执行命令可以免去打开菜单再选命令的繁琐过程，例如在调试过程中需要连续运行程序可以按 F9 键。常用的快捷键如图 4-1。注：不是每一条命令都有快捷键。

表 4-1 用于 JY2X00 仿真调试器的常用快捷键

F1	帮助	F5	开始 / 停止调试
CTRL+F9	编译	F6	设置/取消断点
CTRL+F10	全部编译	CTRL+S	保存
F7	步入	CTRL+X	剪切
F8	步越	CTRL+F	查找
F9	连续运行	CTRL+P	打印
CTRL+T	运行到指定地址	CTRL+C	复制
F4	运行到光标处	CTRL+N	新建
CTRL+F8	暂停运行	CTRL+O	打开
CTRL+F2	复位	CTRL+R	替换
CTRL+J	设置新的 PC 值	CTRL+Y	删除
CTRL+Shift+(0-9)	设置/取消书签	CTRL+V	粘贴
CTRL+(0-9)	跳转到书签位		

## 4.2.7 快捷菜单

当激活某一窗口（如源程序编辑窗口、工程管理窗口、变量观察窗口等），您可以使用鼠标右键打开当前窗口的快捷菜单，以显示当前窗口的最典型的命令，当前窗口快捷菜单中的命令只对当前窗口有效。要关闭快捷菜单，只需在窗口其它部分单击鼠标，或者按 Esc 键。如图 4-7 所示，右击“源程序编辑窗口”时，显示该窗口的快捷菜单。



图 4-7 源程序编辑窗口的快捷菜单

## 4.2.8 在线帮助系统

您可以象使用其它 Windows 帮助系统那样方便地使用它。

## 4.3 菜单介绍

### 4.3.1 文件 (F)



图 4-8 文件菜单

- 新建(N) ... Ctrl+N

新建用户源程序文件，在输入文件名时必须输入文件的扩展名，汇编语言扩展名必须为“.asm”或“.a51”，C 语言扩展名必须为“.c”，PL/M 语言扩展名必须为“.plm”。

- 打开(O) ... Ctrl+O

打开用户源程序或其它文档文件，在文件列表框中选取文件或在文件名输入框中直接输入文件名，当文件名不存在时，新建文件。

- 保存(S) Ctrl+S

保存当前已经打开的源文件或其它文档文件。用户在修改程序后，如果进行编译，则在编译前，系统会自动将修改过的源文件存盘。

- 另存为(A) ...

将当前已经打开的文件另存为指定的文件，原来的文件内容不会改变。

- 关闭(C)

关闭当前已经打开的源程序或其它文档文件。

- 新建工程(T) ...

新建用户工程文件，在输入文件名时必须输入文件的扩展名“.jpr”。

- 打开工程(J) ...

打开一个已经存在的工程文件（包括JY的\*.jpr和Keil的\*.Uv2工程文件）。系统中只允许打开一个工程，打开一个工程或新建一个工程时，前一工程将自动关闭。

- 保存工程

保存当前工程文件

- 工程另存为(E) ...

将当前工程文件存为其他工程文件名

- 关闭当前工程(L)

关闭当前已经打开的工程文件。

- 打印(P) Ctrl+P

打印当前已经打开的文件。

- 退出(X)

退出 JY2X00 集成开发环境。

### 4.3.2 工程 (P)

点击“工程”菜单，界面如图 4-9 所示。

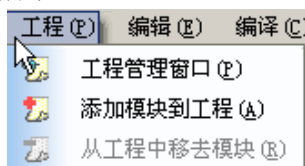


图4-9 工程菜单

- 工程管理窗口(P)

激活工程管理窗口，如图4-10所示。



图4-10 工程管理窗口

- 添加模块到工程(A)

在已经打开的工程中添加源程序模块文件。操作方法有二：

1. 直接执行“添加模块到工程(A)”菜单命令；
2. 在工程管理窗口先选中“源文件”文件夹，然后右键单击被选中的文件夹，在弹出的菜单中单击“添加模块到工程(A)”。

- 从工程中移去模块(R)

在已经打开的工程中删除文件。操作方法有：

1. 直接执行“从工程中移去模块(R)”菜单命令；
2. 在工程管理窗口先选中相应的源程序模块文件，然后右键单击被选中的文件，在弹出的菜单中单击“从工程中移去模块(R)”。

### 4.3.3 编辑 (E)

编辑菜单用于在源程序窗口重进行各种编辑操作。



图4-11 编辑菜单

- 剪切(R) Ctrl + X

将选择的块剪切到剪贴板

- 复制(C) Ctrl + C

将当前文档中选择的部分复制到剪贴板上

- 粘贴(P) Ctrl + V

将剪贴板的内容拷贝到当前文件中光标所在的位置。

- 全选(L) Ctrl + A

选择当前正在使用文档的全部内容。

- 撤销(U) Ctrl + Z

撤销上一次操作

- 重复(D) shift + Ctrl + Z

重做上一次撤销的命令

- 查找(F) Ctrl + F

在当前正在使用的文档中查找指定的字符或字符串。使用该命令将弹出一个对话框，供您输入要查找的字符或字符串。

- 替换(R) Ctrl + R

以指定的内容取代当前正在使用的文档中指定目标的内容。使用该命令将弹出一个对话框，供您输入要替换的字符或字符串。

- 光标定位到... Ctrl + L

定位到指定行。

#### 4.3.4 编译 (C)

“编译”菜单的下拉菜单如图 4-12 所示。



图4-12 编译菜单

- 编译(M) Ctrl + F9

编译、链接当前（正在使用）添加的或修改过的程序模块文件。有关程序的编译状态信息将显示在信息窗口中。当连接仿真器进行硬件仿真并且在设置对话框“编译器”选项卡中的“调试”选项卡选择“编译后自动装载”时，编译、链接后将机器码装入仿真器中。

- 全部编译(B) Ctrl + F10

编译、链接当前正在使用的工程所有程序模块文件。有关编译状态信息将显示在信息窗口中。当连接仿真器进行硬件仿真并且在设置对话框“编译器”选项卡中的“调试”选项卡选择“编译后自动装载”时，编译、链接后将机器码装入仿真器中。

- 编译控制项…

执行该命令后，将弹出对话框，你可以根据需要进行编译控制命令的选择。

- 链接控制项…

执行该命令后，将弹出对话框，你可以根据需要进行连接控制命令的选择。

有关编译、链接控制命令，请参阅 Keil 编译器相关的文档资料。

### 4.3.5 调试 (D)

“调试”菜单的下拉菜单如图 4-13 所示。



图 4-13 调试菜单

- 开始 / 停止调试 F5

在对工程编译、链接成功（产生绝对目标文件）后，利用此命令开始 / 停止调试用户程序。

- 连续运行(R) F9

连续运行用户程序，直到遇到“断点”、“复位”或“暂停”命令后才停止运行。

- 暂停运行(U) Ctrl + F8

暂停当前正在运行的程序。

- 复位(E) Ctrl + F2

该命令使得仿真器的PC值及专用寄存器都处于复位状态。

- 步入 F7

逐步运行用户程序，遇到“调用”则进入所调用的子程序，并在该子程序的第一个可执行语句时停止。

- 步越 F8

逐步运行用户程序，遇到“调用”，则越过所调用的子程序。

- 运行到光标处(C) F4

从当前程序地址处开始执行到当前光标所在的程序执行行（有效程序行）。如果当前光标处于一个不可执行的程序行上，则调试器不能执行该操作。

- 运行到指定地址… Ctrl + T

使用该命令将弹出一个对话框，如图4-14所示，供您输入指定的地址值。JY2X00C仿真调试器从当前程序地址处开始执行到指定地址处。

- 自动单步跟踪运行 Shift + F7

自动以单步跟踪的方式运行程序，其自动单步调试速度可以从“设置”菜单中设定。

- 自动单步运行 Shift + F8

自动以单步的方式运行程序，其自动单步调试速度可以从“设置”菜单中设定。



图4-14 运行到指定地址

- 设置新PC值… Ctrl + J

程序编译成功后，使用该命令将弹出一个对话框，供您输入指定的PC值。仿真器将从新PC值处开始执行。

- 设置/取消当前断点 F6

在源程序窗口中，将光标所在行设为断点，如图4-15所示。如果该行原来已为断点，则取消该断点。编辑状态下的文本窗口以及调试状态下的源文件窗口有效。操作方法有三：

1. 执行“设置/取消当前断点”菜单命令；
2. 点击程序编辑窗左端适当位置，则可在编辑区对应的行设置/取消断点；
3. 快捷操作：F6。

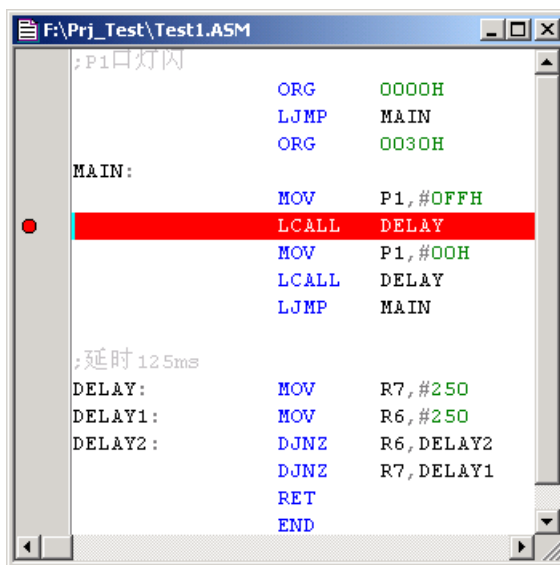


图4-15 设置/取消当前断点

- 清除所有断点

该命令取消原设置的所有断点。

- 调入Hex文件

如果需要对独立的Hex文件（包括“Intel hex 格式”和“Motorola hex 格式”）进行调试，可以通过此菜单命令打开Hex文件进行调试。调试方式同普通的工程。

- 调入Bin文件

如果需要对独立的Bin文件（二进制数据文件）进行调试，可以通过此菜单命令打开Bin文件进行调试。调试方式同普通的工程。

- 调入OMF文件

如果需要对独立的OMF文件（无后缀或后缀为“.ABS”的绝对目标文件）进行调试，可以通过此菜单命令打开绝对目标文件进行调试。调试方式同普通的工程。

### 4.3.6 观察 (Y)

JY2X00 仿真调试软件的观察窗口菜单激活各种类型的观察窗口，是 JY2X00 仿真调试软件窗口管理的主要菜单。按下 Alt + V 键，激活此菜单。



图4-16 观察菜单

- 观察窗(W)

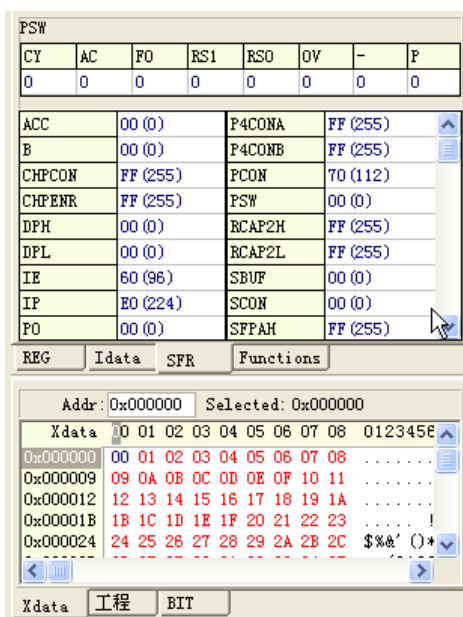




图4-17 观察窗口

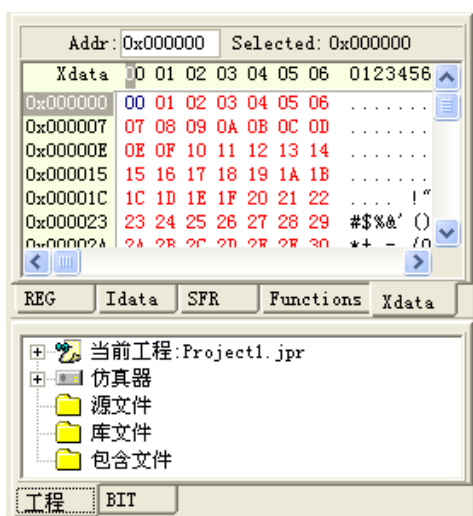
打开/关闭观察窗口



选中该项时，会在左侧出现，同时主窗口左侧打开一个含有REG（通用寄存器）、Idata（内部存储器）、SFR（特殊功能寄存器）、Xdata（外部存储器）、BIT（位寻址区）、Edata（扩展存储器，仅对如Philips的某些芯片）、EEPROM（对于STC芯片）、工程等选项卡的观察窗口（选项卡的多少与上次关闭调试器时的现场和所选择的芯片有关，各选项卡可单独打开、关闭、拖动）。再选中该项时，左侧消失，关闭观察窗口，如图4-17所示。

点击某个选项卡可出现相应的页面，如点击Idata选项可出现内部数据窗口页面。选项卡可任意拖动。如点击Xdata选项拖至观察窗口的上方，当虚线框框住上半部的页面时，松开鼠标后就能将Xdata选项移到了观察窗口上方，也可以拖至观察窗口以外的地方。

选项卡相应的页面可任意打开或关闭。如要打开 / 关闭Xdata选项时，可执行“观察”→“外部数据窗口”命令进行切换，如图4-18所示。



4-18 拖动选项卡

#### • 编译、链接信息自动提示 (M)

显示系统编译输出的信息，如果有编译错误，则在源程序中指出错误所在行；显示设置的所有的断点列表和书签列表；显示在工程所有文件中查找到的全部结果，如图4-19所示。

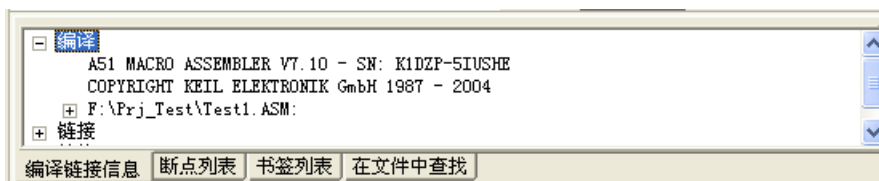


图4-19 编译、链接信息自动提示

#### • 内部数据窗口 (I)

打开/关闭内部数据窗口

内部数据窗口，显示片内RAM区域低端128字节、高端128字节的数据值。红色值为上一步执行过程中改变过的值，窗口标题栏中为选中数据的地址，可以在选中的数据上直接修改设置某地址的数据值（十六进制），如图4-20所示。



图4-20 内部数据窗口

### • 通用寄存器 (G)

打开/关闭通用寄存器窗口 (REG)

通用寄存器窗口，显示80C51内核基本的寄存器R0-R7，A，B，DPH，DPL，SP和PSW，以十六进制（十进制）方式显示字节寄存器的内容，在上方以位的方式显示所选定的寄存器内容，在数据显示区按鼠标右键可以查看寄存器的地址，如图4-21和4-22所示。



图4-21 通用寄存器窗口

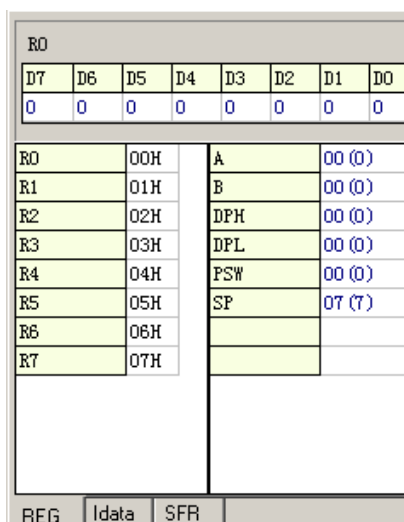


图4-22 按鼠标右键可以查看地址

- 专用寄存器窗口(R)

打开/关闭专用寄存器窗口(SFR)

寄存器窗口，显示当前被选择的CPU 所包含的特殊功能寄存器，以十六进制（十进制）方式显示字节寄存器的内容，在上方以位的方式显示所选定的寄存器内容，在数据显示区按鼠标右键可以查看寄存器的地址，如图4-23所示。



图4-23 专用寄存器窗口

- 位数据窗口(B)

打开/关闭位数据窗口(BIT)

位数据窗口，显示片内的位寻址区(位地址为00H-7FH)的数据值（对应于字节地址为20H-2FH的每一位），如图4-24所示。

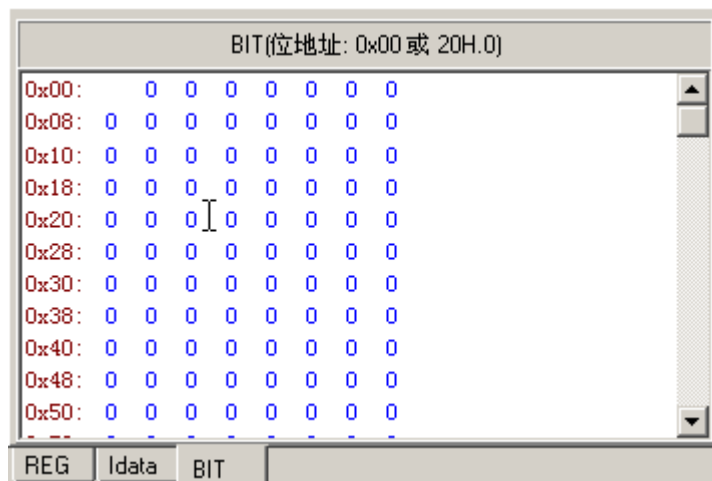


图4-24 位数据窗口

- 外部数据窗口(X)

如图4-25所示，可以选择打开/关闭外部数据窗口



图4-25 外部数据窗口命令

外部数据窗口(Xdata)，显示片外RAM区域(64KB)的数据值。红色值为上一步执行过程中改变过的值，窗口标题栏中为选中数据的地址，可以在选中的数据上直接修改数据的十六进制值，如图4-26所示。



图4-26 外部数据窗口

外部数据窗口(Pdata)，页面存储方式下的外部数据空间（即高8位地址相同的256个外部数据），使用movx @ri, a或movx a, @ri指令进行访问。例如：当P2=E7H时，则外部数据窗口(Pdata)显示的是地址为E700H-E7FF的256个字节数据。

#### • CPU反汇编窗口(C)

打开/关闭CPU反汇编窗口

CPU反汇编窗口，将程序代码区的内容以反汇编方式显示。反汇编窗口内为程序地址、机器码、反汇编码。

调试中，当CPU反汇编窗口为当前激活窗体时，“单步调试”则以CPU反汇编窗口的指令逐条执行；可以在CPU反汇编窗口设置断点；在CPU反汇编窗口单击右键，在弹出的菜单中点击“源码对照”可以设置或取消反汇编码和源程序代码的对照显示，如图4-27所示。

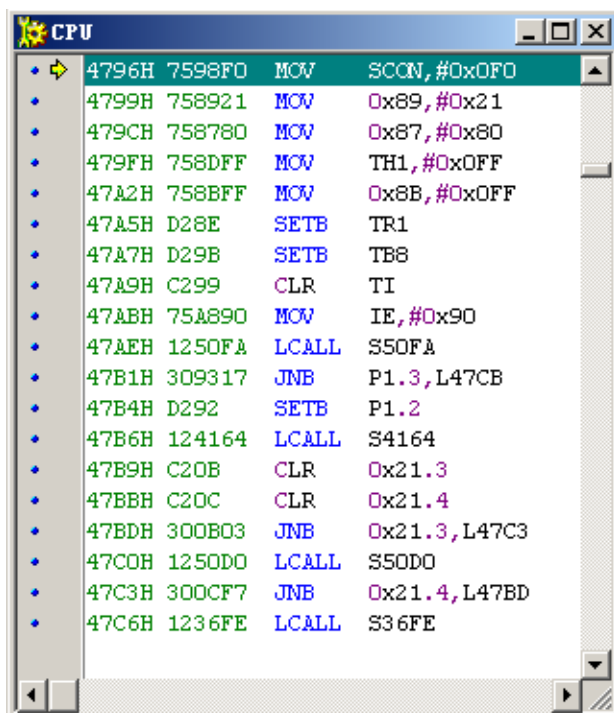


图4-27 反汇编窗口

- Functions

显示用户程序模块中所包含的函数

- 变量窗口

变量窗口显示用户需要观察的变量名、变量类型、变量值，如图4-28所示。

Name	Value	
DELAY	3FH (63)	▲
DELAY1	41H (65)	
DELAY2	43H (67)	▼

Locals 变量观察

图4-28 变量窗口

可以在变量窗口弹出菜单中进行如下选择：

添加观察项——把当前行的变量加入变量观察窗口显示。

编辑观察项——可以打开”编辑观察项”窗口,对当前变量进行修改,求值。

删除观察项——删除变量观察窗口内选定的变量，不再对此变量进行观察。

- 逻辑分析仪

打开/关闭逻辑分析仪窗口

### 4.3.7 设置 (O)



图4-29 设置菜单

- 仿真器设置

仿真器的设置包括：仿真模式设置、仿真器型号的选择、仿真适配器电源的选择、串口的选择、P0、P2口的使用方式选择等，如图4-30所示。

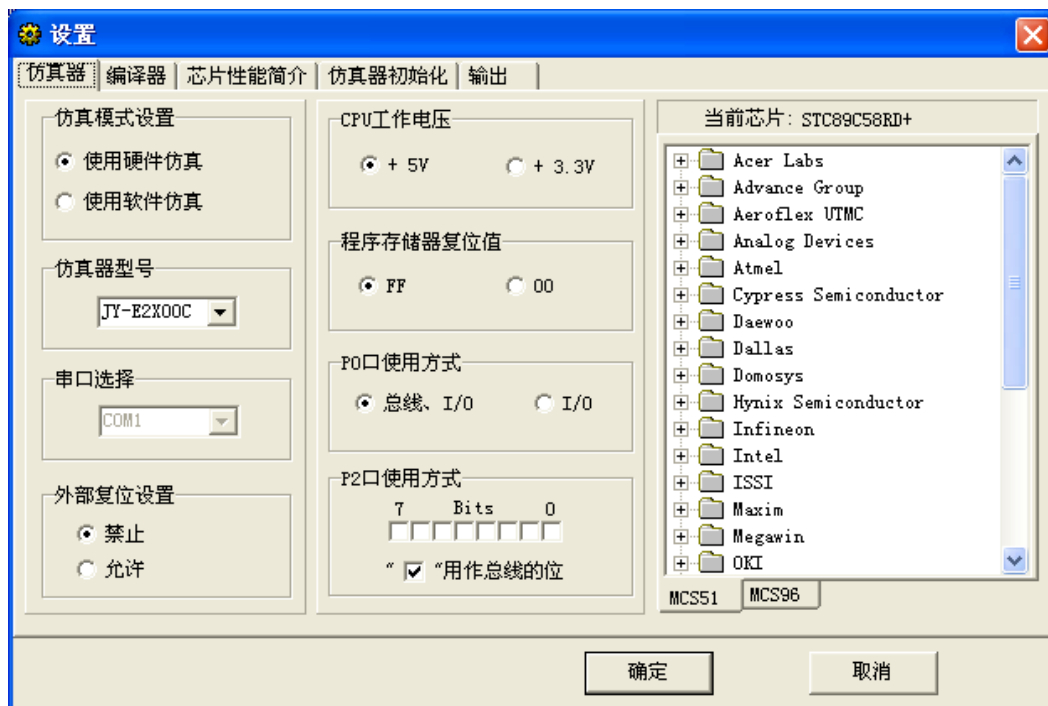


图4-30 仿真器设置

①仿真模式设置——使用硬件仿真：仿真时需连接JY-E2X00C仿真器；使用软件仿真：仿真时需不连接JY-E2X00C仿真器。

②仿真器型号选择——在列表框中选择要仿真的仿真器型号“JT-E2X00C”。

③P0口使用方式——总线、I/O：仿真时P0口可同时用作总线、I/O；I/O：仿真时P0口只用作I/O，即使访问片外RAM时，P0口也不会输出地址信号和数据信号。

④P2口使用方式——仿真时，P2口各位根据需要可全部或部分用作总线或I/O。

⑤仿真适配器电源设置——使用仿真器内部电源（+5V）：为被仿真的单片机提供+5V电压；使用目标板电源（+3.3V）：为被仿真的单片机提供+3.3V电压。

⑥外部复位设置——禁止：禁止外部复位；允许：允许外部复位。

⑦程序存储器复位值——FF：程序存储器复位值为FF；00：程序存储器复位值为00。

仿真设置中的“仿真器初始化”选项卡

进入“仿真器初始化”选项卡，还需要进行相应的设置，如图 4-31，其中在“目标时钟设置”组合框中选择或键入适当的工作频率。

**注意：**12 Clocks 芯片一般不超过 50MHz；

6 Clocks 芯片一般不超过 25MHz；

4 Clocks 芯片一般不超过 31MHz；

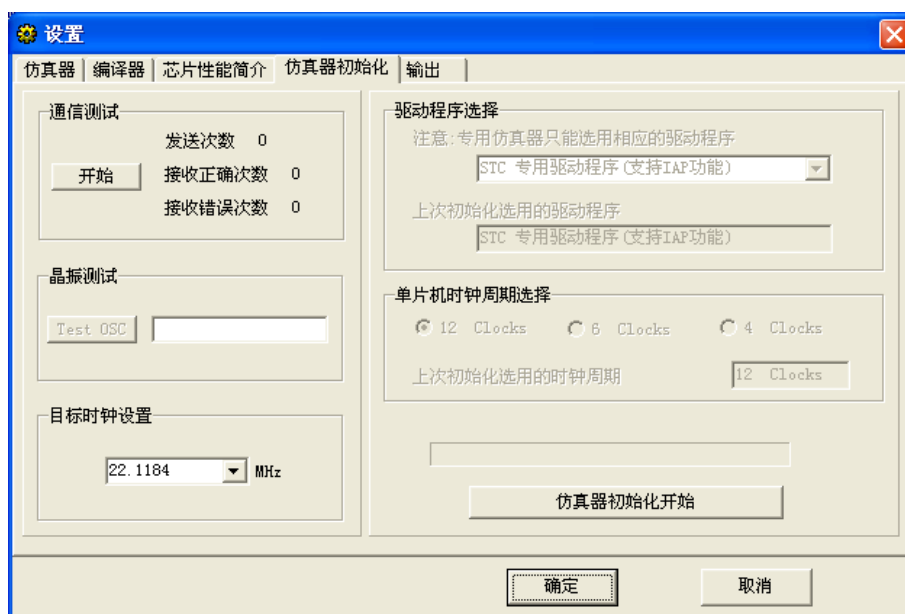


图4-31 仿真器初始化界面

### • 编译器设置

包括对编译器、链接器的选择、路径设置和相应的控制命令设置等。

有关的控制命令、“编译器设置”选项卡“Setting”按钮中的优化选择，请参阅 Keil 编译器相关的文档资料。

**注意：**“设置”窗口中，在下面三种情形下需要初始化仿真器：

- 改变了被仿真单片机所属的厂家；
- 改变了被仿真单片机的时钟周期；
- 安装了新版本的驱动程序。

### • 逻辑分析仪设置

设置逻辑分析仪的有关参数。请参阅逻辑分析仪使用说明。

### • 编辑环境设置

设置文本编辑器环境参数。软件中已经预定义了四组风格的编辑环境，为“默认”、“古典”、“高亮”、“海洋”。用户还可以自己定义自己喜爱的颜色，在“元素”栏中选择要定义的元素，在“颜色”栏中定义背景色和前景色。在“字体”栏中选择字体。定义颜色和字体后，可以左侧的样式中看到编辑器产生的效果，如图4-32所示。



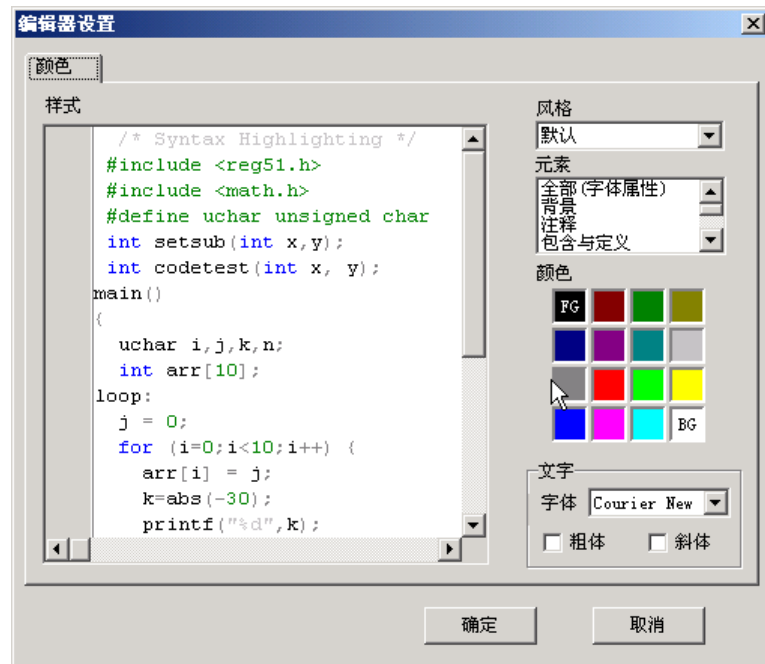


图4-32 编辑器设置对话框

### • 自动步进调试速度设置

设置自动单步运行、自动单步跟踪运行的时间间隔，如图4-33所示。

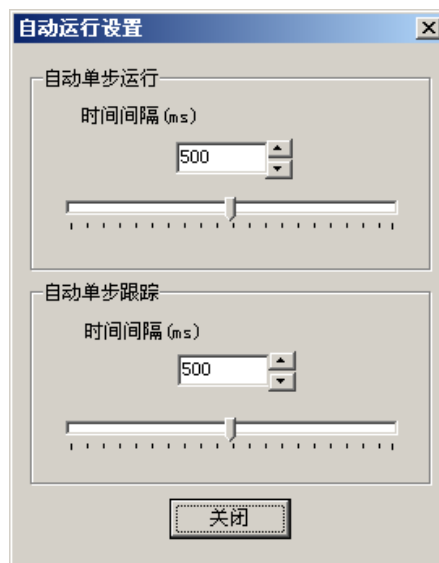


图4-33 自动步进调试速度对话框

### • 语言

可选择将菜单以中文或英文方式显示，如图4-34所示



图4-34 语言子菜单

### • Trace

设置Trace功能的有关参数。通过对Trace功能的合理设置（见“设置”对话框的“编译器”选项卡中的“调试”选项卡），才能保证Trace功能的正确使用。

如果选择了Trace功能，在连续运行暂停或运行到光标处后，JY2X00仿真调试器会自动将Trace数据从仿真器中读回并显示在源程序编辑窗口和CPU反汇编窗口左侧的“指示带”上。但对于单步调试，JY2X00

仿真调试器提供了读取Trace数据的两种选择（Auto与Manual），这是因为：在程序较大时，Trace数据也会相应变大，如果固定频繁地读取较大的Trace数据，会严重影响调试速度。用户可根据需要灵活设置，如果选择Manual，主窗口中的快捷按钮“T”激活，点击快捷按钮“T”时刷新Trace显示。

### 4.3.8 工具 (T)

工具菜单栏包括计算器、汇编指令速查表、定时器初值计算、波特率计算和生成数据源代码 5 个下拉菜单，如图 4-35 所示。

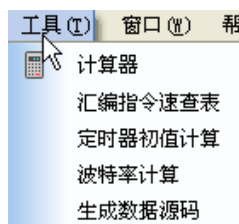


图4-35 工具菜单

- 计算器

调用Windows系统的计算器。

- 汇编指令速查表

提供了MCS-51系列单片机的汇编指令。

- 定时器初值计算

可根据定时器的工作方式、晶振频率和定时间隔计算出定时器的计数初值。

- 波特率计算

可根据串行接口的工作方式、晶振频率、波特率和SMOD值计算出定时器T1作为串行口波特率发生器时的计数初值。

- 生成数据源码

在程序调试期间,可以将程序存储器或数据存储器内指定区域的内容生成供其它程序或工程使用的数据源代码。语言格式可选，数据排列可调。

数据源码生成后，可以将其部分或全部内容复制到操作系统的剪贴板，供粘贴使用。

可以通过主菜单、也可在某数据窗口的弹出菜单中执行此项命令。

### 4.3.9 窗口 (W)

窗口菜单符合windows窗口风格，如图4-36所示。

- 层叠

层叠当前所有激活的窗口

- 平铺 (H)

横向平铺当前所有激活的窗口

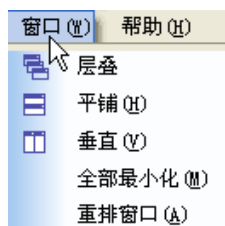


图4-36 窗口菜单

- 垂直 (V)

纵向平铺当前所有激活的窗口

- 所有最小化 (M)

最小化当前所有激活的窗口

- 重排窗口 (A)

排列当前所有激活的窗口

### 4.3.10 帮助 (H)

帮助菜单如图 4-37 所示。

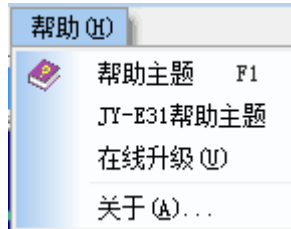


图 4-37 帮助菜单

- 帮助主题 F1

提供有关的帮助信息。

- JY-E31 帮助主题

提供 JY-E31 有关的帮助信息。

- 在线升级 (U)

在线进行软件升级。

- 关于...

关于JY2X00仿真调试器对话框中列出了软件和硬件有关版本信息，以及联系方式。

## 4.4 使用示例

下面通过简单的编程、调试，引导大家学习 JY2X00 仿真调试软件的基本使用方法和基本的调试技巧。

### 4.4.1 硬件仿真

将仿真芯片适配头插座插入用户目标板，并按 2.2 节的要领连接好仿真器，接通 JY-E2X00C 仿真器的电源，单击 Windows “开始” 菜单中的 “JY2X00 IDE” 或双击 Windows 桌面上的 “JY2X00 IDE” 图标，启动 JY2X00 仿真调试程序。

启动后的 JY2X00 IDE 如图 4-38 所示。

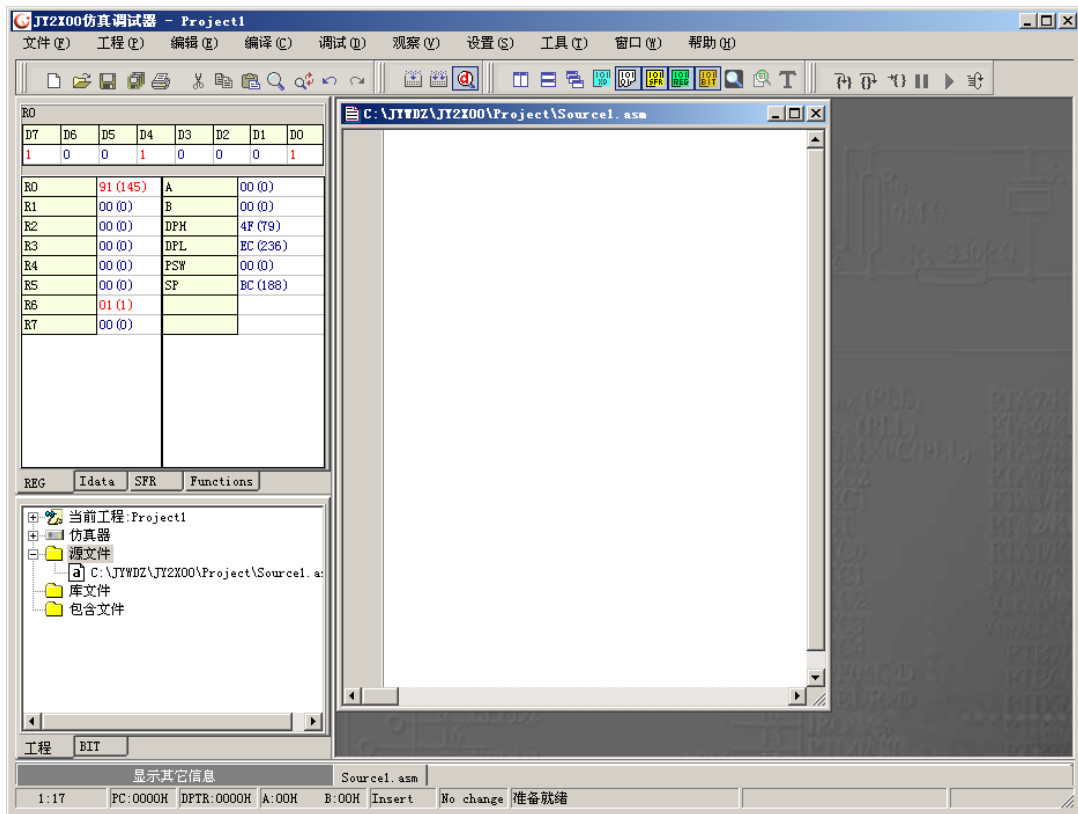


图 4-38 JY2X00 IDE 窗口

**注意：**JY2X00 IDE 启动时，会自动检测仿真器的连接及端口状况，如果没接仿真器、计算机与仿真器联接不正常或仿真器电源没打开，会弹出图 4-39 所示的对话框供选择。

硬件仿真调试时，单击“否”按钮；

软件仿真调试时，单击“是”按钮。

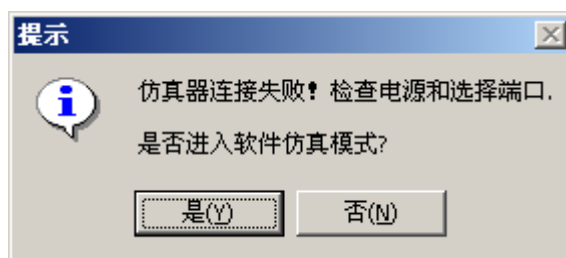


图 4-39 软、硬仿真选择对话框

## 1. 工程的建立

单击“文件”菜单，在弹出的下拉菜单中选中“新建工程”选项，会弹出“建立新工程”对话框，选择你要新建工程的路径，输入工程文件的名字，如图 4-40 所示，然后单击“保存”按钮，弹出“仿真器设置”对话框，显示如 4.3.7 小节的图 4-30 所示，按以下步骤进行有关的设置：

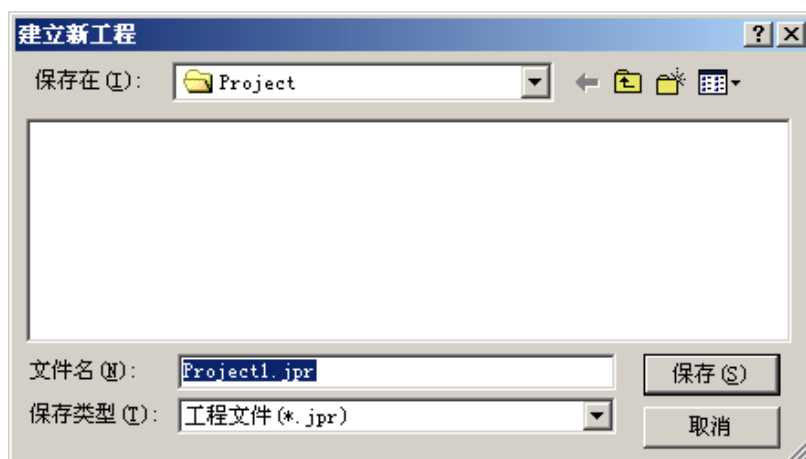


图 4-40 新建工程

#### ① 设置仿真器

在“仿真模式设置”选项区域中，选择“使用硬件仿真”；

在“当前芯片”列表框中选择将要仿真的单片机型号。JY2X00C 仿真器几乎支持所有的 51 内核的单片机。例如选取 Atmel 的 AT89C52，点击 Atmel 前面的“+”号，展开该层，选中 AT89C52。

其余选项的设置参见 4.3.7 小节。

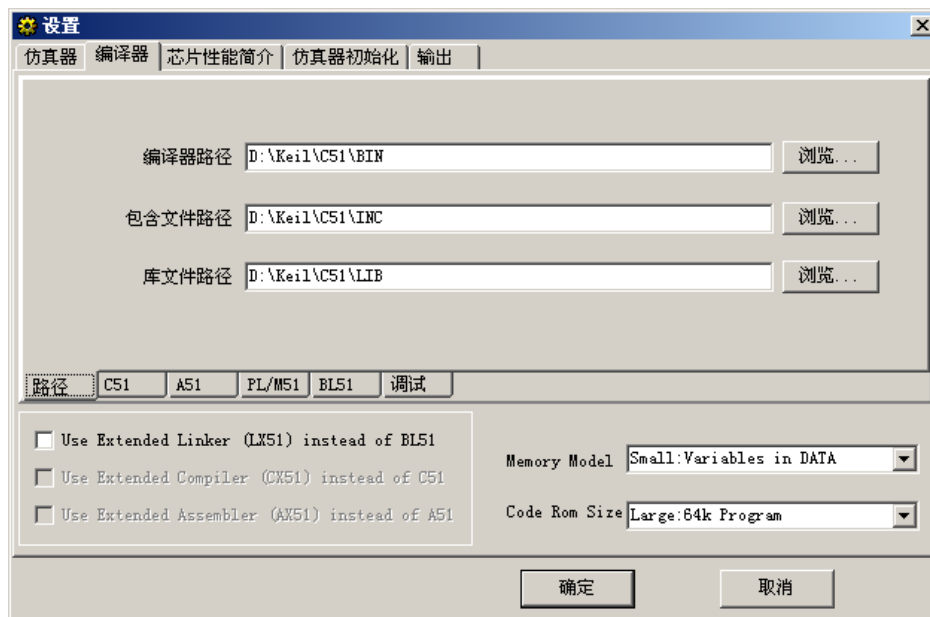


图 4-41 设置编译器路径

#### ②设置编译器”

单击“编译器”选项卡，设置编译器路径的相关参数，如图 4-41 所示。

设置编译器调试的相关参数，如图 4-42 所示。



图 4-42 设置编译器调试

设置好后单击“确定”按钮。

### ③仿真器初始化

单击“仿真器初始化”选项卡，可设置仿真器初始化的有关参数。

单击“通信测试”选项区域中“开始”按钮，检查端口通信状态，单击“停止”按钮，停止通信测试；

设置目标时钟。在“目标时钟设置”组合框中选择或键入适当的工作频率(如图 4-43)。

单击“仿真器初始化开始”按钮，对仿真器进行初始化，在进度条中会显示进度值，仿真器初始化成功后会弹出“仿真器初始化成功”消息框。

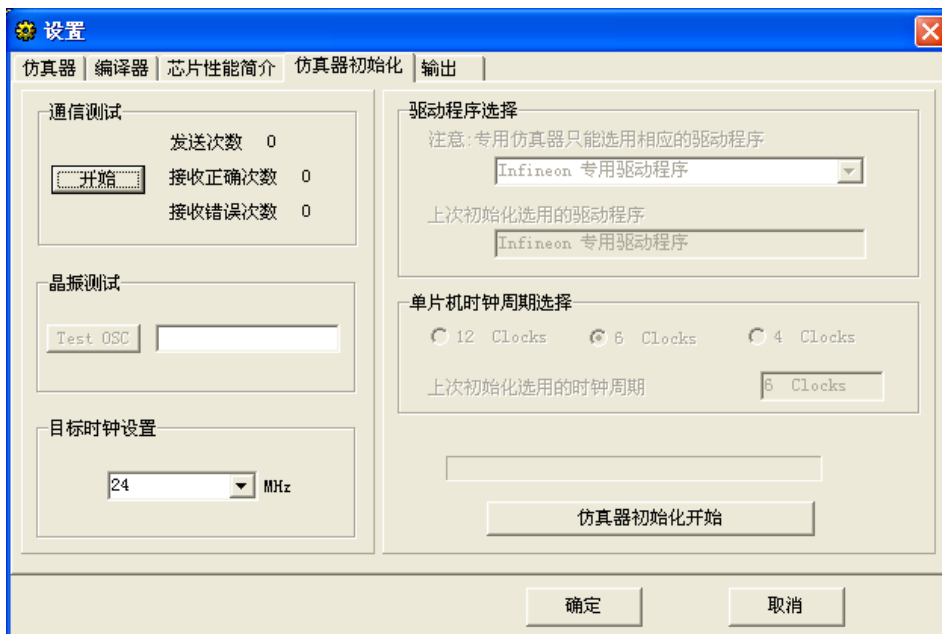


图 4-43 仿真器通信端口设置

单击“确定”，进入硬件仿真模式。

## 2. 工程的保存

单击“文件”菜单，在弹出的下拉菜单中选中“保存工程”选项，会弹出“保存工程文件”对话框，选择你要保存的路径，输入工程文件的名字，比如保存到 Prj\_Test 目录里，工程文件的名字为 Project1.jpr 如下图所示，然后点击“保存”按钮。

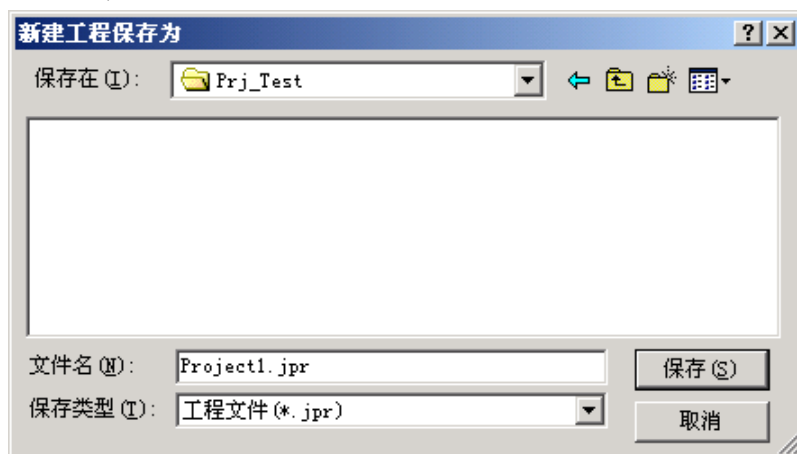


图 4-44 保存工程对话框

保存好工程后，在工程管理窗口中的“当前工程”文件夹图标左边出现了一个“+”号，点击它可看到刚才建立的工程的名称和位置，如图 4-45 所示。

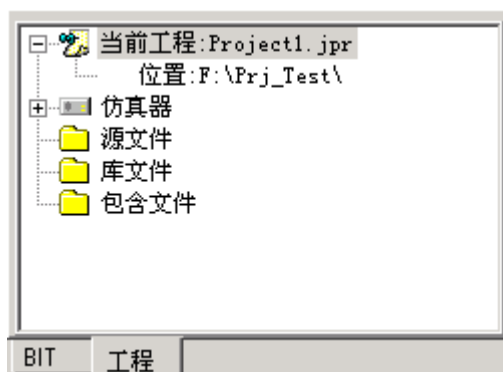


图 4-45 工程管理窗口

完成上一步骤后，我们就可以进行程序的编写了。

## 3. 源程序文件的建立

首先我们要在项目中创建新的程序文件或加入已有的程序文件。如果您没有现成的程序，那么就要新建一个程序文件。在 JY2X00 中有一些程序的示例，在这里我们还是以一个汇编程序为例介绍如何新建一个程序和如何加到您的第一个工程中吧。

在主界面中，单击“文件”菜单，再在下拉菜单中单击“新建”命令，如图 4-46 所示。

在主界面中出现了个文件名为NONAME1 的源程序编辑窗口，屏幕如图4-47所示。

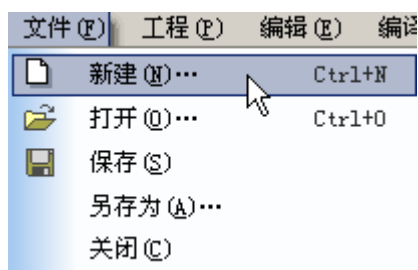




图 4-46 新建源文件

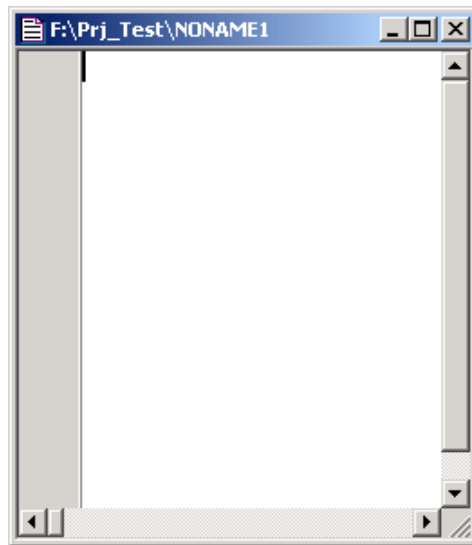


图 4-47 源程序编辑窗口

此时光标在编辑窗口里闪烁，这时可以键入用户的应用程序了，建议您首先保存该空白的文件，单击“文件”菜单，在下拉菜单中选中“另存为”选项，屏幕如下图所示，在“文件名”栏右侧的编辑框中，键入欲使用的文件名，同时，必须键入正确的扩展名。注意，如果用C语言编写程序，则扩展名为“.c”；如果用汇编语言编写程序，则扩展名必须为“.asm”，这里我们用Test1.ASM作为文件名，然后，单击“保存”按钮，如图4-48所示。

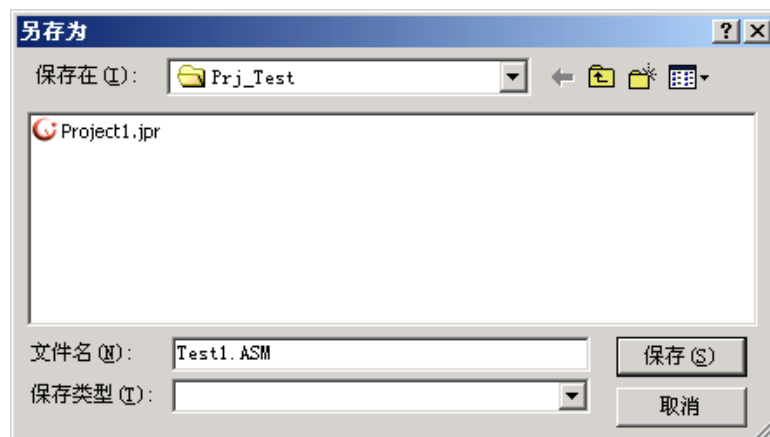


图 4-48 保存文件对话框

回到编辑界面后，单击“工程”菜单，在下拉菜单中单击“添加模块到工程”命令，如图4-49所示（或在“工程窗口”的“源文件”文件夹上单击鼠标右键，在弹出的快捷菜单中选中“添加模块到工程”选项，如图4-50所示）。

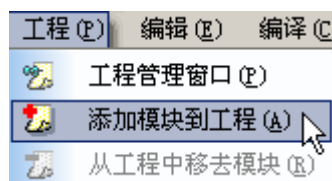


图 4-49 通过菜单命令添加模块到工程

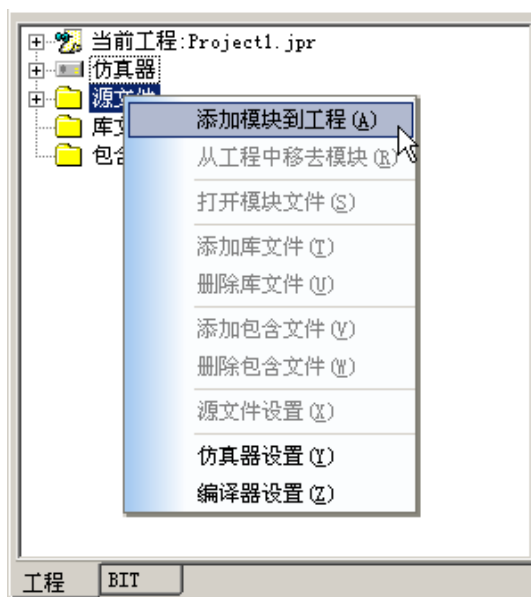


图 4-50 通过工程窗口添加模块到工程  
弹出“打开”对话框，如图 4-51 所示。

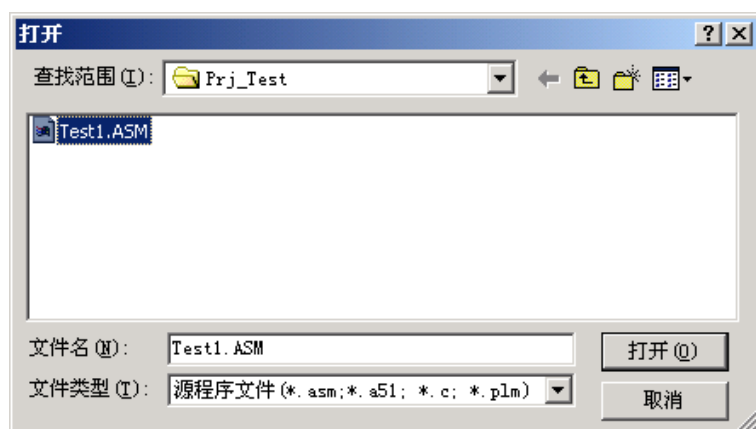


图 4-51 添加模块对话框

选中 Test.ASM，然后单击“打开”，可将该文件加入到到工程中，此时，在工程管理窗口中的“源文件”文件夹图标左边出现了一个“+”号，说明文件夹中有了文件，展开该文件夹，可看到“源文件”文件夹中多了一个子项“……Test1.ASM”，如图 4-52 所示，这就是刚才我们加入的源程序文件名，子项的多少与所增加的源程序的多少相同。

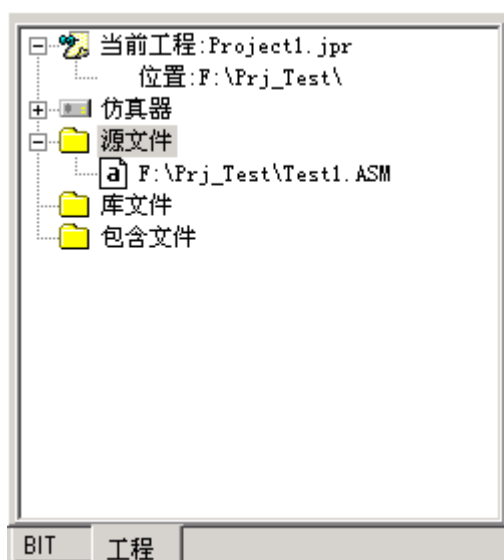


图 4-52 查看添加的模块

现在，请输入如下的汇编语言源程序：

；P1口灯闪

```

ORG      0000H
LJMP     MAIN
ORG      0030H

```

MAIN：

```

MOV      P1,#0FFH
LCALL    DELAY
MOV      P1,#00H
LCALL    DELAY
LJMP     MAIN

```

；延时125ms

```

DELAY:    MOV      R7,#250
DELAY1:   MOV      R6,#250
DELAY2:   DJNZ     R6,DELAY2
          DJNZ     R7,DELAY1
          RET
          END

```

在输入上述程序时，您可能已经看到了事先保存待编辑的文件的好处了，即JY2X00会自动识别关键字，并以不同的颜色提示用户加以注意，如图4-53所示，这样会使用户少犯错误，有利于提高编程效率。

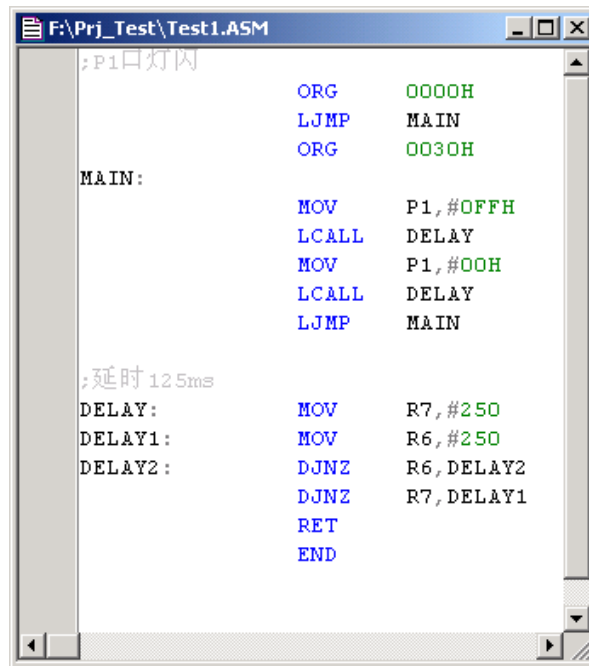


图 4-53 编辑源文件

#### 4. 编译

单击“编译”菜单，再在下拉菜单中单击“编译”选项（或者使用快捷键 Ctrl+F9），编译过程中的信息将出现在“编译链接信息”窗口中，如果编译或链接不成功，则会自动打开“显示其它信息”窗口。如源程序中有语法错误，则会出现错误报告，并标示出出错的相应行，如图 4-54 所示。

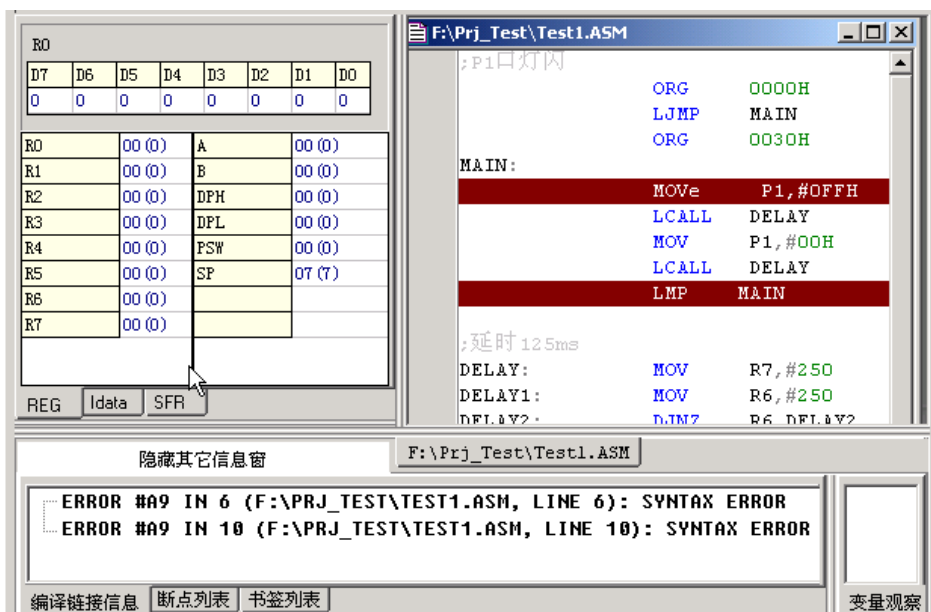



图 4-54 编译源文件

也可单击“工程管理器”下方的“显示其它信息”，打开“显示其它信息”窗口查看相关信息。

将错误改正后，再进行编译。如果事先在设置对话框“编译器”选项卡中的“调试”选项卡选择“编译后自动装载”时，编译、链接成功后机器码会自动装入仿真器中；否则，待编译、链接成功后，点击“开始/停止调试”快捷按钮，装载机器码并进入调试状态。

源程序窗口左侧出现了一列小圆点“●”，表示程序的有效行，即此行存在有效的程序代码，在

程序的 LJMP MAIN 语句所指行的左侧出现黄色箭头，表示当前的程序计数器 PC，如图 4-55 所示。

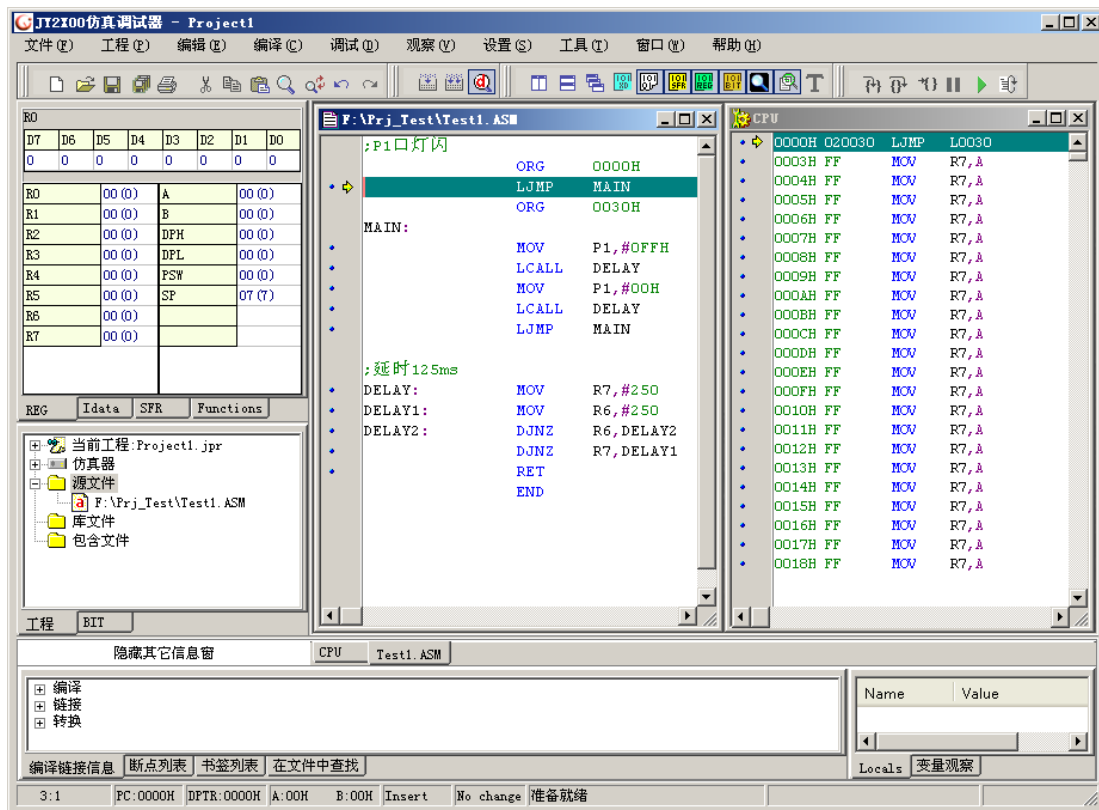



图 4-55 编译通过

## 5. 调试程序

代码装载至仿真器后，就可调试程序了，您可以根据需要进行单步、断点、全速运行等操作，观察各种变量。

### ① 步入（单步跟踪运行）

单击观察窗口中的“REG”选项卡，显示“通用寄存器窗口”，单击“SFR”选项卡，显示“专用寄存器窗口”（如果需要同时显示，可单击任意一个选项卡，拖动至另外的地方），然后在主界面中，单击“调试”菜单，再在下拉菜单中单击“单步跟踪运行”或按“步入”快捷图标或按 F7 键进行单步跟踪调试程序，黄色箭头，指向下次将要执行的程序行（或指令）。继续单步跟踪运行就一条指令一条指令地执行程序，若有子程序调用，也会跟踪到子程序中去。你可以观察程序每步执行的结果。

程序单步跟踪到“DELAY”延时子程序中，执行“MOV R7,#250”后，在“通用寄存器窗口”中可看到红色的“R7”的值，也可将鼠标指向源程序行的“R7”字符稍作停留就可以观察“R7”的值，如图 4-56 所示，继续单步跟踪调试程序，执行“DJNZ R6,#250”，观察一下“R6”的值，可以看到“R6”在逐渐减少。因为当前指令要执行 256 次才到下一步，整个延时子程序要单步执行 250x250 次才能完成。

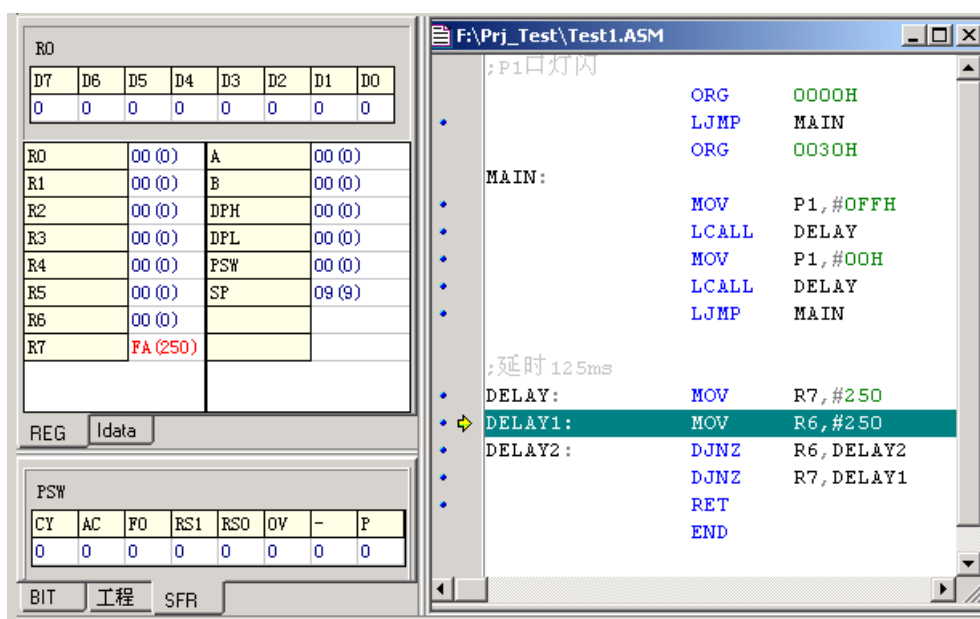


图4-56 点屏观察变量值

## ②执行到光标处

将光标移到程序想要暂停的地方，如本例中延时子程序返回后的“LJMP MAIN”行。选择“调试”菜单，再在下拉菜单中单击“单步跟踪运行”或 F4 键或弹出菜单的“执行到光标处”功能或按“执行到光标处”快捷图标。程序全速执行到光标所在行，如图 4-57 所示。

## ③步越（单步运行）

如果不想进入“DELAY”延时子程序里，在“DELAY”子程序调用处，按 F8 键单步执行就可以全速执行子程序调用（即单步越过子程序），从而真实反映子程序延时情况。

也可以用“自动单步跟踪运行”或“自动单步运行”。单步间的时间间隔可以调整。

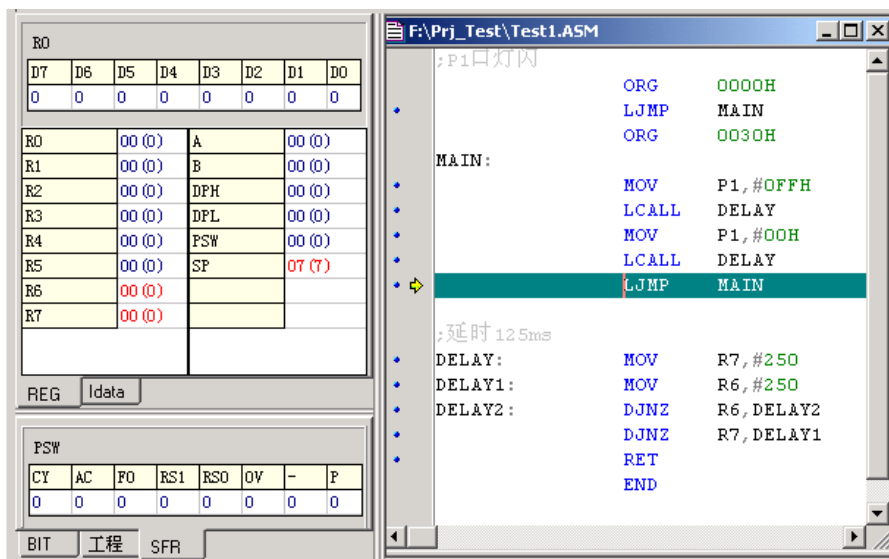


图4-57 运行到光标处

## ④连续运行

单击“调试”菜单，在下拉菜单中单击“连续运行”命令，（或者使用快捷键F9），单击“调试”菜单下的“暂停运行”命令（或单击“暂停”快捷按钮）程序暂停后，就可以看到运行后的结果（R6、R7、P1的值），如图4-58所示。

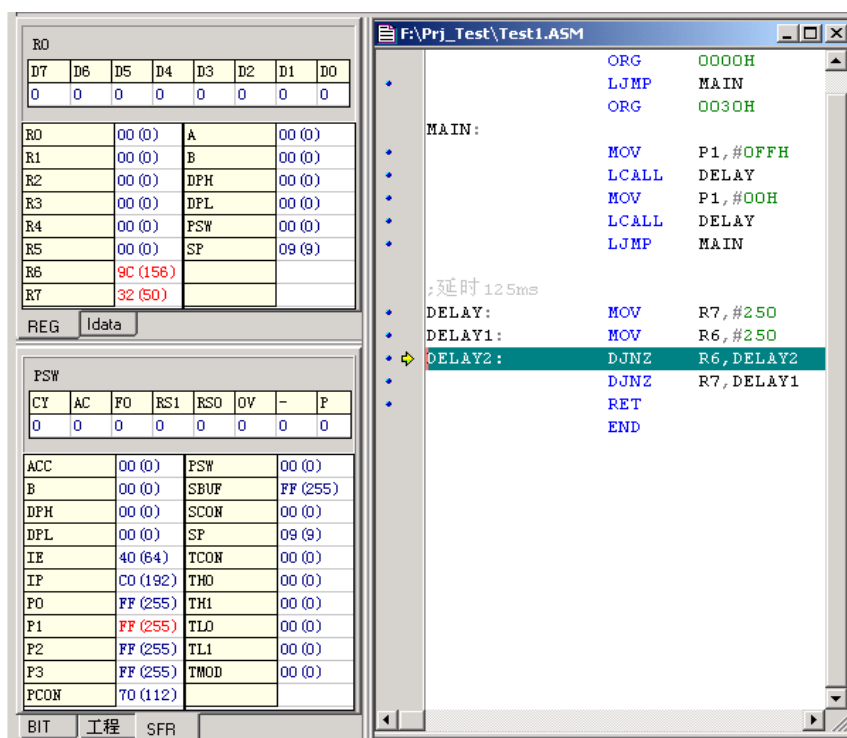




图 4-58 连续运行

#### ⑤带断点的连续运行

将光标移到源程序窗口的左边“Gutter”区域，单击左键设置断点，也可以用F6键将光标所在行设置为断点。如果断点有效图标为, 无效断点的图标为。断点设置好后，就可以用全速执行的功能，全速执行程序，当程序执行到断点时，会暂停下来，这时你可以观察程序中各变量的值，及各端口的状态，判断程序是否正确，如图4-59所示。

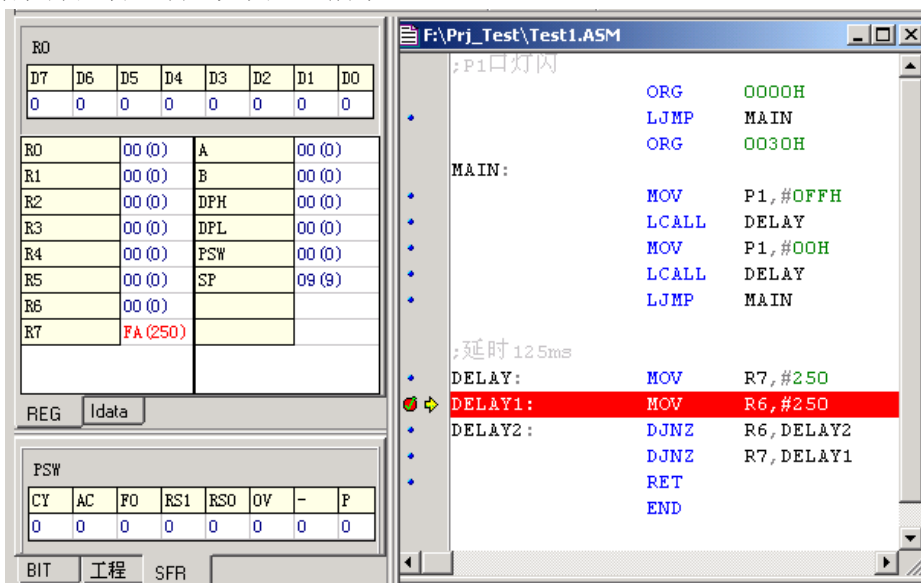


图 4-59 将光标所在行设置为断点

仿真成功后，就可以用编程器将本工程产生的 Project1.hex 文件烧录进入芯片了。



## 4.4.2 软件模拟仿真

在 JY2X00 调试环境下，我们既可以进行硬件仿真，也可以进行软件模拟仿真。与进行硬件仿真所不同的是：在仿真模式中，应选择“使用软件模拟仿真”，如图 4-60 所示。

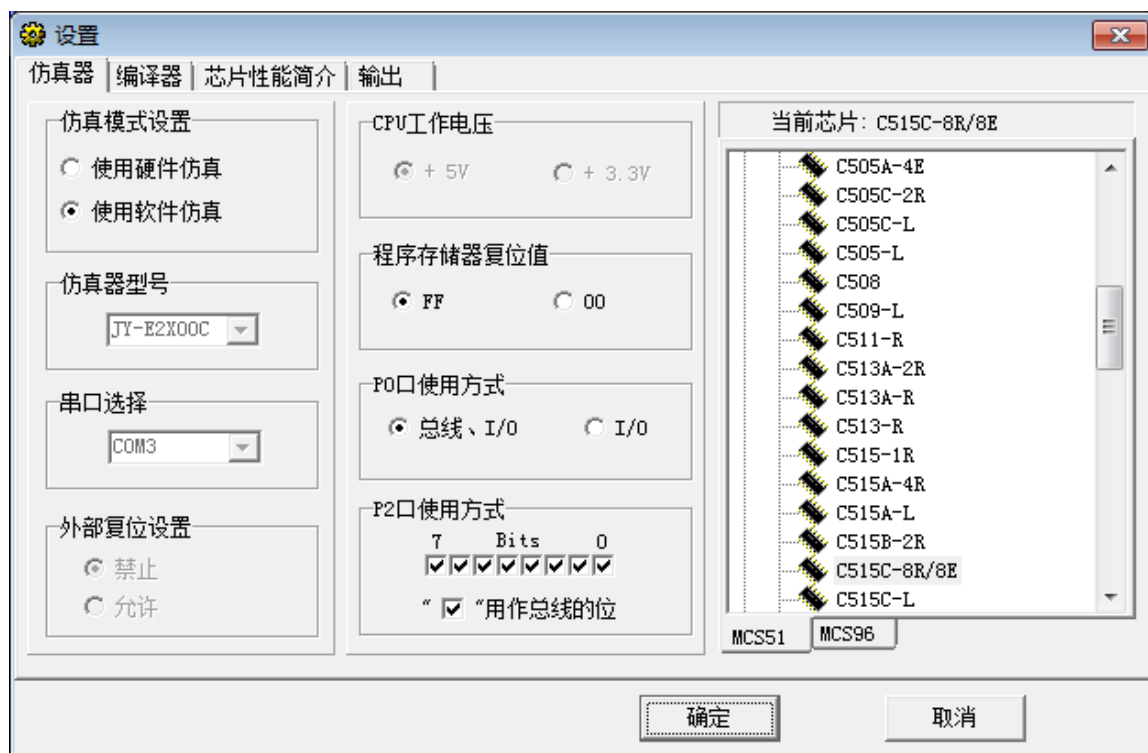


图 4-60 设置为软件模拟仿真

在进行软件模拟仿真时，编译后代码不会装载至仿真器中。  
调试操作同硬件仿真。

## 第 5 章 JY-E2X00C 兼容 JY-E31 仿真器说明

在 JY-E2X00C 型仿真发布之前，菊阳微电子公司专为宏晶科技 STC10xx 系列、STC11xx 系列、STC12C5Axx、STC12C52xx 系列等 1 个时钟（/机器周期）单片机（以下简称 STC 1T 单片机）开发了一款专用仿真调试工具 JY-E31 仿真器（请参阅《JY-E31 仿真器用户手册》）。

应广大客户的要求，我们将 JY-E31 仿真 STC 1T 单片机的功能加入到新发布的 JY-E2X00C 系列器的 JY-E2200C 与 JY-E2300C 两型仿真器中（JY-E2100C 型仿真器除外），使得这两型仿真器，既可仿真当前所有普通 51 芯片，同时又兼容 JY-E31 的功能，仿真 STC 1T 单片机。

如何选择相关配件？ JY-E2200C 或 JY-E2300C 仿真器如何与 STC 1T 用户目标板连接？ JY-E2200C 或 JY-E2300C 仿真器仿真 STC 1T 单片机如何操作？ 等等，本章作简要说明。

限于篇幅，对于本手册没有涉及到的部分，而于用户又是非常重要的内容，请参考《JY-E31 仿真器用户手册》，特别是其中的注意事项，请用户务必留神，以避免走弯路。

### 5.1 配件与连接

#### 1、配件

JY-E2200C 或 JY-E2300C 仿真器仿真 STC 1T 单片机，除 JY-E2X00C 的基本配置外，还必须根据用户实际情况，选择配置下列配件：

- 如果用户目标板单片机为 DIP40、DIP28、DIP20、DIP18、DIP16、PLCC44 封装之一时，可采取仿真器加仿真头的方式进行仿真。

应该选择一个 JY-STC1T-H1CH 仿真头（含仿真芯片 IAP12C5A60S2，Vcc 为 5V），或 JY-STC1T-H1CL 仿真头（含仿真芯片 IAP12LE5A60S2，Vcc 为 3.3V）（图 5-1），这两种仿真头仅仅针对不同的 Vcc 选择预置了不同的芯片，其它并没有什么不同。

除了选择适当的仿真头外，还要根据目标板上单片机封装选择相应的转换座： JY-STC1T-H2 DIP40 转换座（图 5-2）、JY-STC1T-H3 DIP20/28 转换座（图 5-3）、DIP16 转换座、DIP18 转换座、PLCC44 转换座（图 5-4）；

这样，仿真头经转换座转换后直接插入目标板的芯片座，连接方便。

- 如果用户目标板上单片机为贴片封装时，用户应另行购买一片（或几片）STC 厂家生产的型号为 IAP12LE5A60S2 或 IAP12C5A60S2（根据目标板上 VCC 而定）并与用户板上引脚封装一致的单片机，焊接到用户板，然后从用户单片机引出 10 根线到一个 14Pin 的接线座（同仿真头上的 14Pin 的接线座）。这种仿真连接方式，我们称之为目标板“预设仿真接口”连接。其接法见后述。

● 无论是用“仿真头”接入目标板，还是目标板“预设仿真接口”连接，都必须配备一个特殊器件。这个特殊器件就是 JY-E2200C 或 JY-E2300C 仿真器的 80Pin 排线转换成 14Pin 的转换板，这个转换板形状特殊，故简称之为“特殊转换头”（下同），见图 5-5。

主要相关配件实物图片：

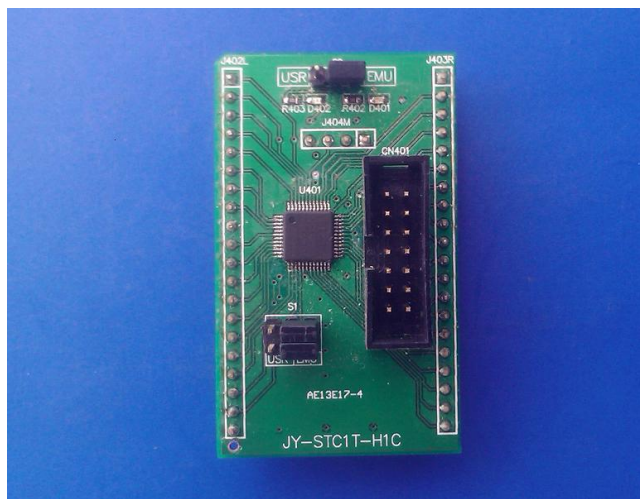


图 5-1 JY-STC1T-H1CH、L 仿真头

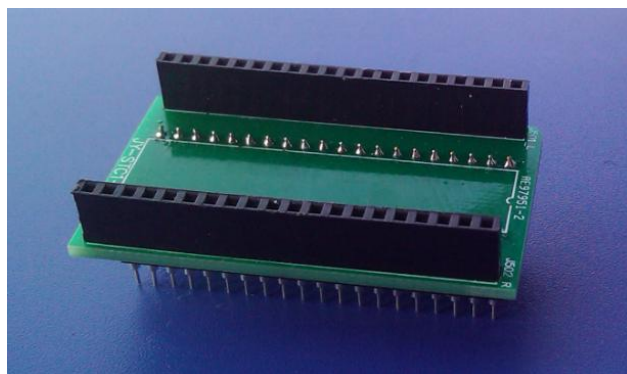


图 5-2 JY-STC1T-H2 DIP40 转换座

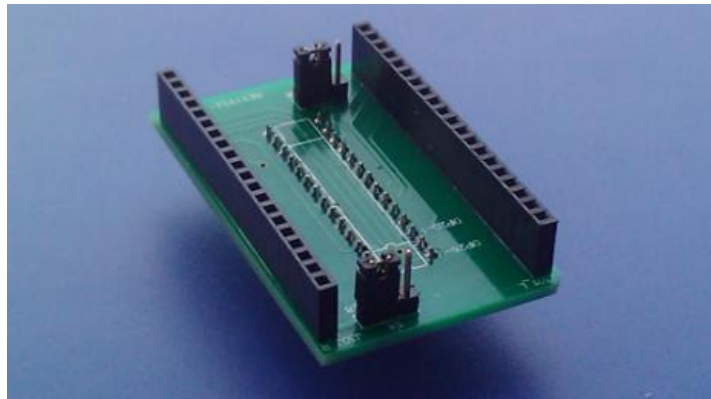


图 5-3 JY-STC1T-H3 DIP20/28 转换座

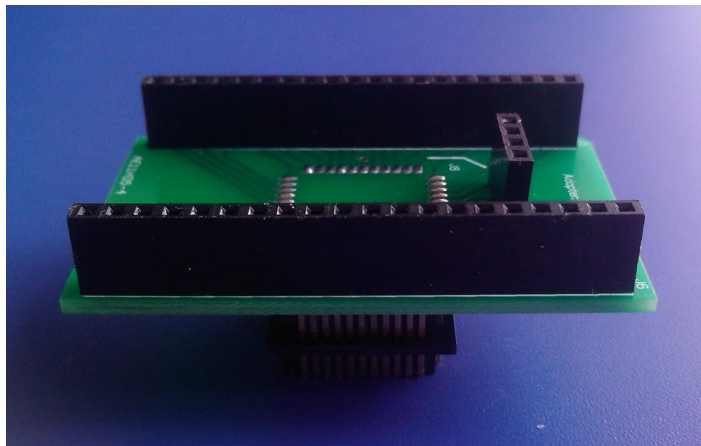


图 5-4 PLCC44 转换座

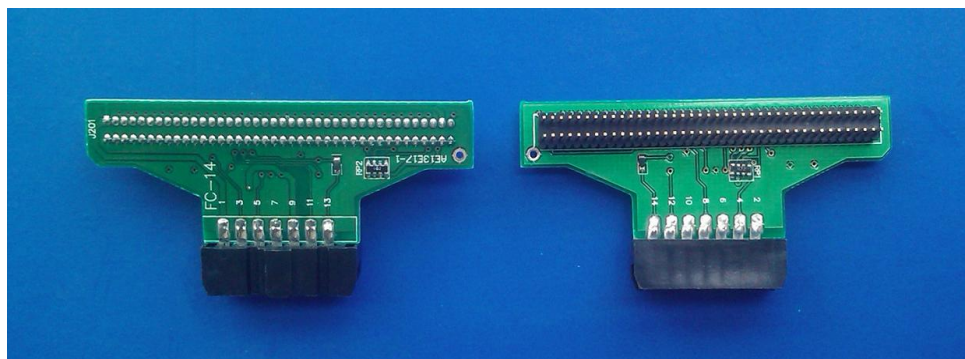


图 5-5 80Pin To 14Pin 转换板（特殊转换头）

## 2、连接

这里所谓的“连接”是指用户目标板与仿真器的连接方式。主要分为两种，一是使用仿真头连接，二是通过为目标板“预设仿真接口”连接。

### ① 使用仿真头连接

其连接方法是：

80 芯排线一端与仿真器相连，80 芯排线的另一端与上面提及的“特殊转换头”相连，然后，将“特殊转换头”上的 14 芯头插入 JY-STC1T-H1C 仿真头上的 14Pin 插座（见图 5-6）。

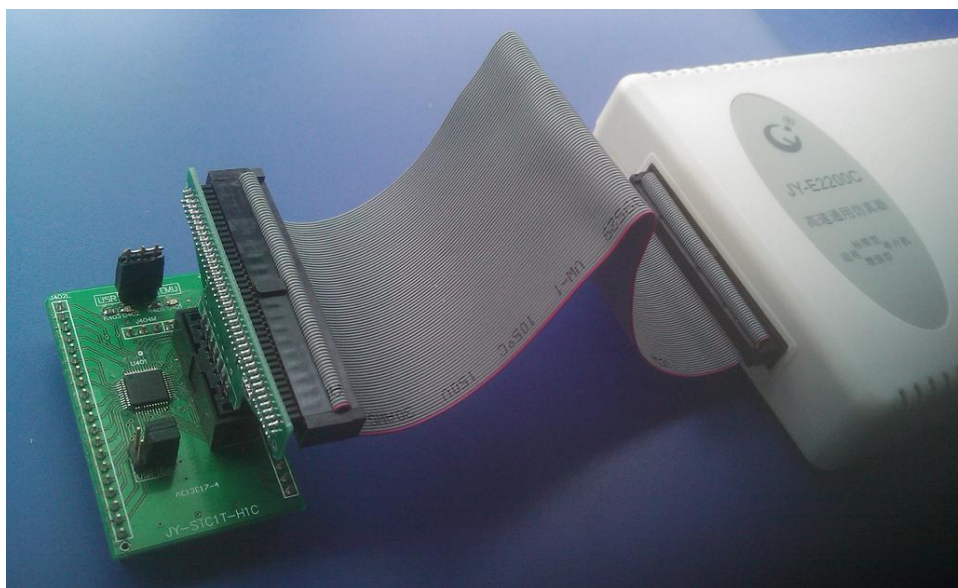


图 5-6 仿真头通过转换头、80 芯排线与仿真器连接

**注意：**80 排线与“特殊转换头”结合时，排线压线头上的卡榫朝上(如图 5-7)。



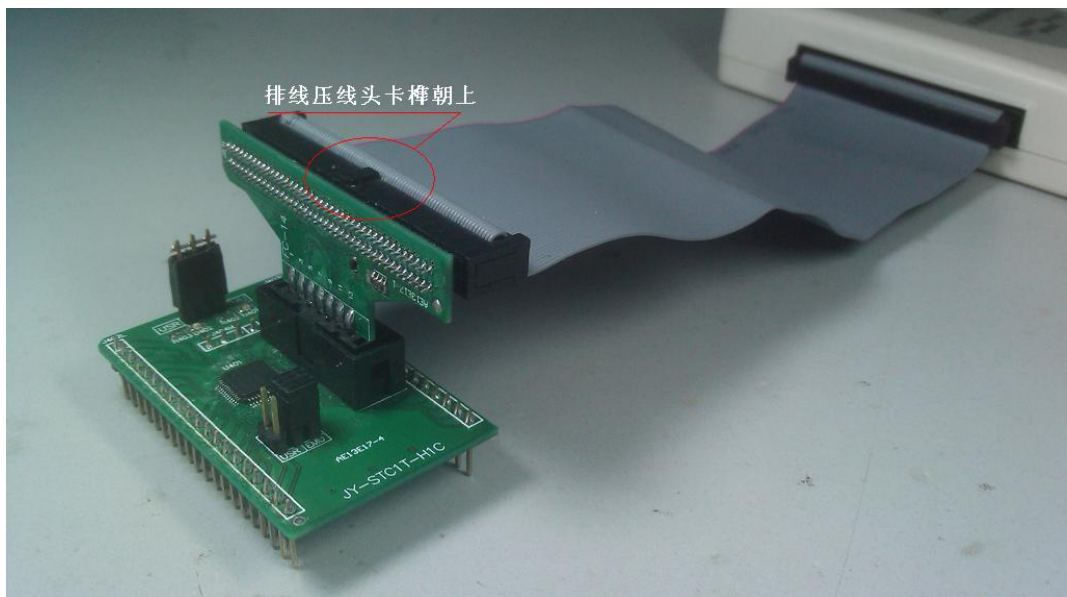


图 5-7 排线与“特殊转换头”结合时，排线压线头上的卡榫朝上

然后，仿真头插入转换座（DIP20/28 转换座、DIP40 转换座、PLCC44 转换座、DIP16 转换座、DIP18 转换座其中之一）上，再将转换座插到目标板的 MCU 插座上（见图 5-8）。这种连接方式的仿真芯片在仿真头上，目标板不需预设仿真接口，方便仿真六种封装的芯片：DIP20、DIP28、DIP40 及 DIP16、DIP18、PLCC44。



图 5-8 仿真头与仿真器连好经转换座插入用户板

## ② 目标板“预设仿真接口”连接

事先建立目标板“预设仿真接口”，其接线见图 5-9。

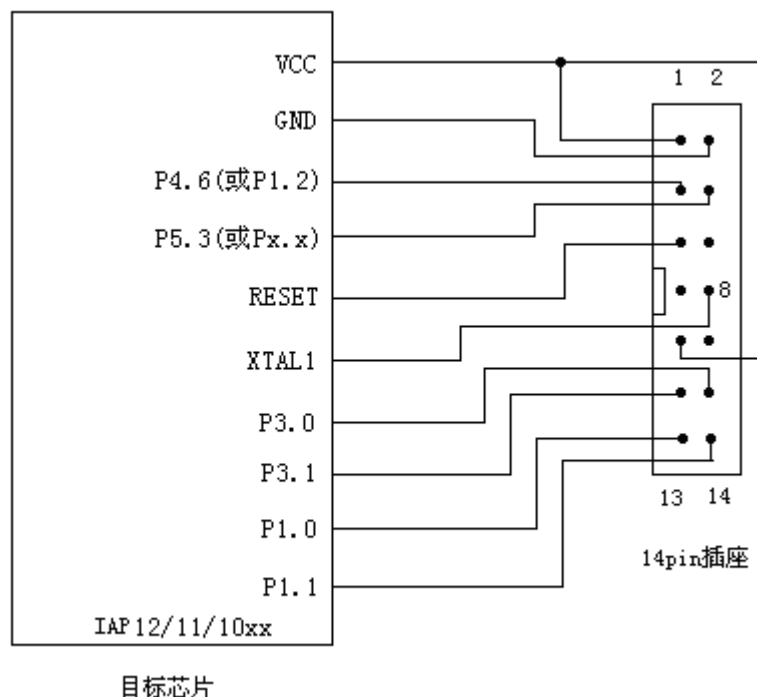


图 5-9 目标板预设仿真接口 14pin 插座接线图

#### 特别强调：

- 目标板预设的仿真时, 目标板上的芯片必须是 STC 提供的与用户计划用 L 芯片封装相同、工作电压一致, 并且具有 IAP 功能（在应用编程功能, 即用户程序可以修改用户程序）的芯片（如: IAP12C5A60S2、IAP12LE5A60S2 等）。芯片程序空间要大于用户程序空间 2K 以上。
- 14pin 插座的第 3 脚接目标芯片的“低电压检测中断脚”（5A 系列芯片一般为 P4.6 脚, 也叫 EA 脚; 52 系列芯片为 P1.2 脚), 目标板上芯片的这个脚既不要上拉, 也不要下拉。
- 14pin 插座的第 5 脚接目标芯片的“复位脚”;
- 14pin 插座的第 8 脚为目标单片机的时钟输入脚（JY-E31 仿真器此脚不接), 其频率通过调试程序根据需要设置;
- 14pin 插座的第 11、12、13、14 脚分别接目标芯片的 P3.0、P3.1、P1.0、P1.1 脚, 这几个脚仅仅是临时借用, 不影响用户仿真及使用。

然后进行部件连接：

80 芯排线一端与仿真器相连, 80 芯排线的另一端与上面提及的“特殊转换头”相连, 接着将“特殊转换头”上的 14 芯头插入目标板预设的 14Pin 插座（见图 5-10）。

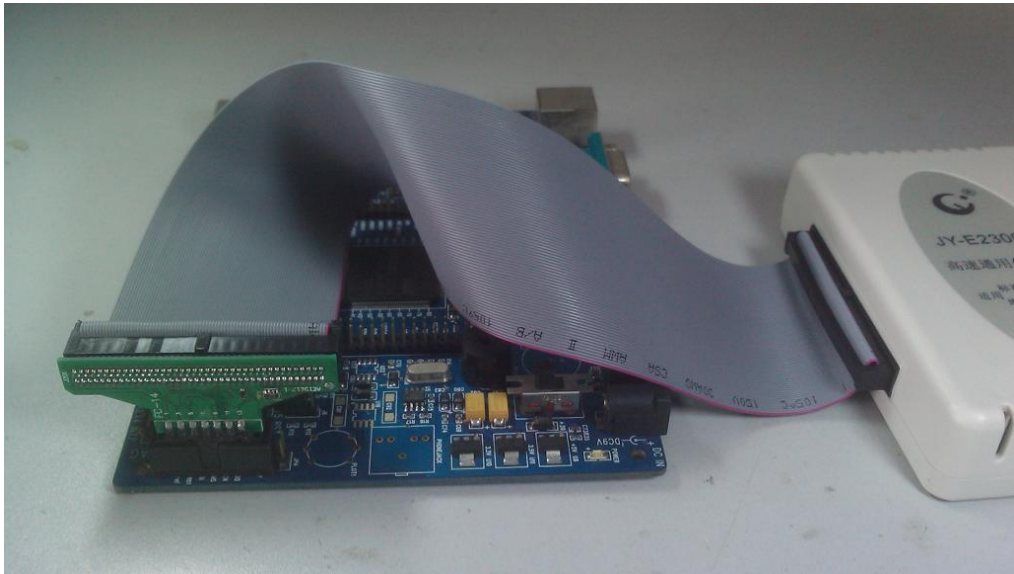


图 5-10 目标板预设有仿真接口时的连接

## 5.2 操作使用

JY-E2X00C 型仿真器仿真 STC 1T 单片机时的操作，与 JY-E31 仿真器的使用方法基本一样。请参阅《JY-E31 仿真器用户手册》“三、使用方法”。

其不同之处：

一是时钟的问题。仿真时，通过仿真头上的“S1 跳线”用户目标板时钟可以选择目标板时钟电路提供的时钟，也可选择 JY-E2X00C 型仿真器提供的时钟。但在“初始化目标”时，必须选择由仿真器来提供时钟；

二是仿真器每次上电时，仿真器总要占一定的时间来完成自身的初始化，用户要注意等待其正常完成；

三是如果目标板是独立供电（不由仿真器提供），请注意：目标电源电压与调试程序环境设置的电压一致。

## 5.3 注意事项

JY-E2X00C 型仿真器仿真 STC 1T 单片机时的注意事项，与 JY-E31 仿真器的注意事项一样。请参阅《JY-E31 仿真器用户手册》“四、注意事项”。

用户在上手前一定要仔细阅读并牢记相关注意事项，这样才能保证较高效率地工作。