Application Note

TKScope 仿真 AVR 使用指南

AN05220021 V1.03 Date: 2011/08/09 产品应用笔记

类别	内容
关键词	TKScope 仿真器 AVR IAR AVR Studio K-lash 软件
特 性	1. 极快编程速度； 2. 完美支持 JTAG 菊花链； 3. 支持烧写外部 Flash。
摘 要	TKScope 仿真 AVR 系列芯片使用指南

目 录

1. AVR 仿真器概述

1.1 AVR 仿真器型号及支持芯片列表

TKScope 系列仿真器支全系列 8 位 AVR 芯片的仿真，具体型号有：K8，K9，DK9，DK10。

AVR 仿真器支持目前支持 ATMEL 公司所有 8 位 AVR 器件的编程（包括 ISP 编程）和仿真，具体型号如下：

- ATmega16，ATmega16A，ATmega162
- ATmega165，ATemga165A，ATmega165P，ATmega165PA
- ATmega169，ATmega169A，ATmega169P，ATmega169PA
- ATmega32，ATmega32A，ATmega323
- ATmega325，ATmega325A，ATmega325P，ATmega3250，ATmega3250P
- ATmega329 ， ATmega329A ， ATmega329P ， ATmega329PA ， ATmega3290 ，ATmega3290P
- ATmega64，ATmega64A，ATmega644，ATmega644A
- ATmega645，ATmega645A，ATmega6450，ATmega649，ATmega649A，ATmega6490
- ATmega640，ATmega1280，ATmega1281，ATmega2560，ATmega2561
- ATmega128，ATmega128A，
- ATmega164A ， ATmega164P ， ATmega164PA ， ATmega324A ， ATmega324P ，ATmega324PA ， ATmega644A ， ATmega644P ， ATmega644PA ， ATmega1284 ，ATmega1284P
- AT90CAN32，AT90CAN64，AT90CAN128
- AT90USB646，AT90USB647，AT90USB1286，AT90USB1287
- ATmega48，ATmega48A，ATmega88，ATmega88A，ATmega168，ATmega168A，ATmega328
- ATmega48P，ATmega48PA，ATmega88P，ATmega88PA，ATmega168P，ATmega168PA，ATmega328P
- ATtiny13，ATtiny13A
- ATtiny2313，ATtiny2313A，ATtiny4313
- ATtiny25，ATtiny45，ATtiny85
- ATtiny24，ATtiny24A，ATtiny44，ATtiny44A，ATtiny84，ATtiny84A
- ATtiny261，ATtiny261A，ATtiny461，ATtiny461A， ATtiny861，ATtiny861A
- ATtiny87，ATtiny167
- AT90PWM2，AT90PWM2B，AT90PWM216
- AT90PWM3，AT90PWM3B，AT90PWM316
- AT90USB82，AT90USB162
- ATxmega64A1，ATxmega128A1，
- ATxmega64A3， ATxmega128A3，ATxmega192A3，ATxmega256A3
- ATxmega256A3B,
- ATxmega16A4，ATxmega32A4，
- ATxmega16D4，ATxmega32D4，
- ATxmega64D3，ATxmega128D3，ATxmega192D3，ATxmega256D3

更多的器件即将支持。

TKScope 仿真 AVR 系列芯片使用的仿真头型号是 **POD-JTAG-AVR-P10**。

图 1.1　POD-JTAG-AVR-P10

1.2　AVR 仿真器功能特性

AVR 仿真器主要功能特性如下：

- USB2.0（High Speed）高速通讯接口，极快的编程速度，节省用户开发时间；
- JTAG 编程速度为原装 JTAGICE mkII 的 2.8 倍；
- ISP 编程速度（1MHz 时钟）为原装 JTAGICE mkII 的 4.7 倍；
- 具备独立 K-Flash 烧写软件，支持高速量产在线编程；
- 支持编程 Flash、EEPROM、Fuse、LockBits；
- 支持所有 MEGA 系列芯片 JTAG 编程和调试；
- 支持所有 debugWIRE 接口器件单线调试；
- 支持带 JTAG 或 debugWIRE 接口的器件 ISP 编程；
- 支持 XMEGA 系列芯片 PDI 编程和调试；
- 自适应 AVR Studio 版本，用户使用不同版本 AVR Studio 无需更改驱动；
- 支持汇编和高级语言调试；
- 支持数据断点和无限制 Flash 断点；
- 支持动态断点，可以在运行过程中设置/取消断点；
- 支持仿真中任意代码修改，方便用户程序排错；
- 支持代码和数据缓冲功能，大大提高调试性能；
- 具有硬件自检功能，快速定位系统硬件问题；
- 仿真器自动检测目标板电压，仿真不同电压芯片无需做额外配置。

1.3　AVR 仿真器支持 IDE 环境

TKScope 仿真 AVR 系列芯片支持多种 IDE 环境，工程师可灵活选择熟悉的开发环境，具体支持的 IDE 开发环境如下：

- TKStudio，致远公司，中/英文界面，多内核编译/调试环境，强大内置编辑器。
- AVRstudio，ATmel 公司，英文界面，可集成 GCC 编译器的 IDE。
- IAR，IAR 公司，英文界面，多内核编译/调试环境。

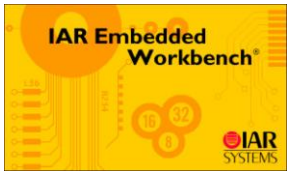

图 1.2　主流 IDE 的界面

1.4 AVR 仿真器的连接

POD_JTAG_AVR_P10 有 2 个接口，40P 接口用于连接仿真器 JP4（靠里面的插座），10P 接口用于连接目标板（标准 JTAG 接口）。

ADP_AVR_P10_P6 用于 ISP 编程和 debugWIRE 仿真，使用时把此小板的 J3（10P 连接器）连接到 POD 头的 10P 接口，J4（6P 连接器）连接到目标板。

JTAG 接口和 ISP 接口标准与官方定义的相同，如图 1.3。

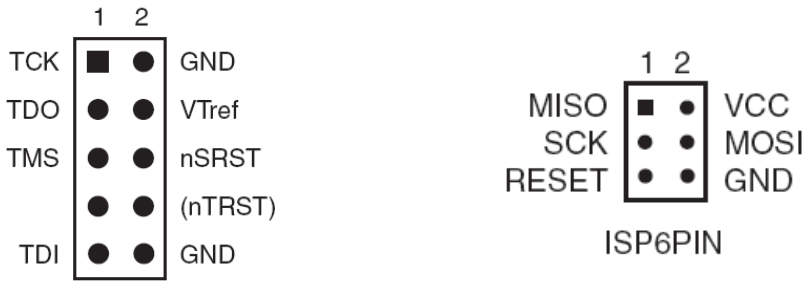

图 1.3 JTAG 和 ISP 连接图

当使用 ISP 编程和 debugWIRE 调试时，我们建议用户使用 ADP_AVR_P10_P6 适配器，注意：适配器的 RESET 引脚对应 JTAG 信号的 TMS，而不是 nSRST。

表 1.1 是 ADP-AVR-P10-P6 的连接对应关系。

表 1.1 ADP-AVR-P10-P6 连接对应关系

JTAG 信号	ISP 信号	debugWIRE 信号	PDI 信号
Pin 1 TCK	Pin3 SCK		
Pin 2 GND	Pin6 GND	Pin6 GND	Pin6 GND
Pin3 TDO	Pin1 MISO		
Pin4 VTref	Pin2 Vcc	Pin2 Vcc	Pin2 Vcc
Pin5 TMS	Pin5 REST	Pin5 debugWIRE	Pin5 PDI_CLK
Pin9 TDI	Pin4 MOSI		Pin4 PDI_DATA

友情提示：从上表的对应关系中可以看到，当 PDI 使用 ADP_AVR_P10_P6 适配器时，Vcc，GND，PDI_CLK 的对应关系都是与官方相同的，但 PDI_DATA 对应在第 4 脚，而不是第 1 脚，您需要在目标板上把 1 脚可 4 脚短接起来使用，如图 1.4。

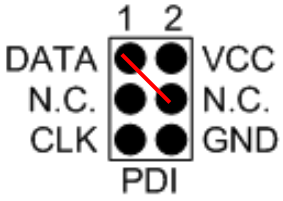

图 1.4 PDI 的硬件连接

2. AVR Studio 环境下仿真 AVR 快速入门

2.1 安装驱动

本驱动为 AVRStudio 设计，建议在安装驱动前先安装 AVRStudio（驱动支持 4.13 及以上的版本）。双击 Setup_TKScope_AVRStudio.EXE，系统弹出图 2.1 的窗口，点击【下一步】按提示安装。

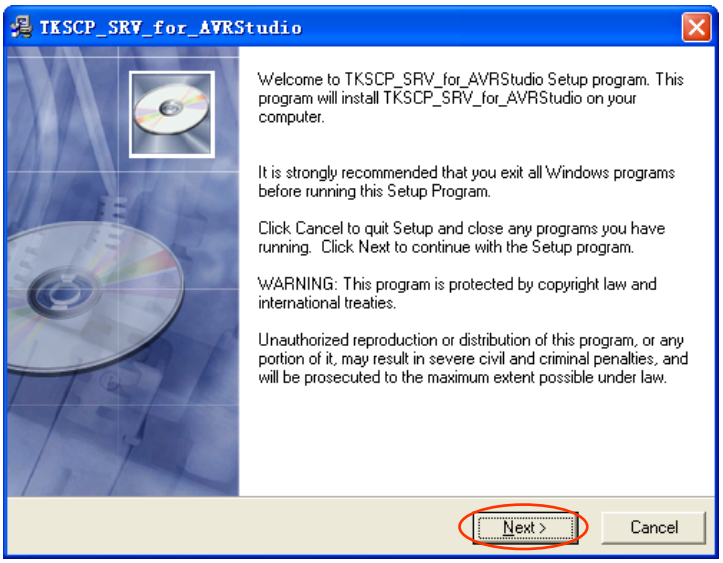

图 2.1 安装驱动

安装路径无限制，用户根据自己的需要选择安装路径。

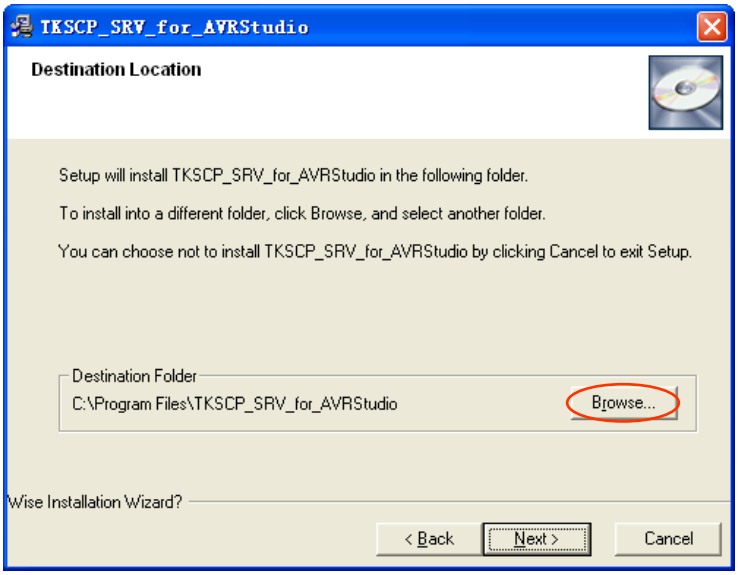

图 2.2 选择安装路径

按提示完成点【下一步】直到完成安装。安装成功后，可以在开始菜单中看到 AVRStudio with TKScope 和 Uninstall AVR Studio with TKScope，其中 AVRStudio with TKScope 用于启动带 TKScope 服务的 AVRStudio，使用 TKScope 在 AVR Studio 下仿真 AVR 需要使用这一项来启动（启动前需先打开仿真器电源并连接 USB）。Uninstall AVR Studio with TKScope 用

于卸载本驱动。

图 2.3　安装完成后的开始菜单

仿真器驱动安装完毕，建议用户安装微软的 VC9 的实时运行库。双击 vcredist_x86_cn.exe（该文件可在您的安装目录下找到，如 C:\Program Files\TKSCP_SRV_for_AVRStudio\），系统会弹出如图 2.4 所示的对话框。点击【下一步】。

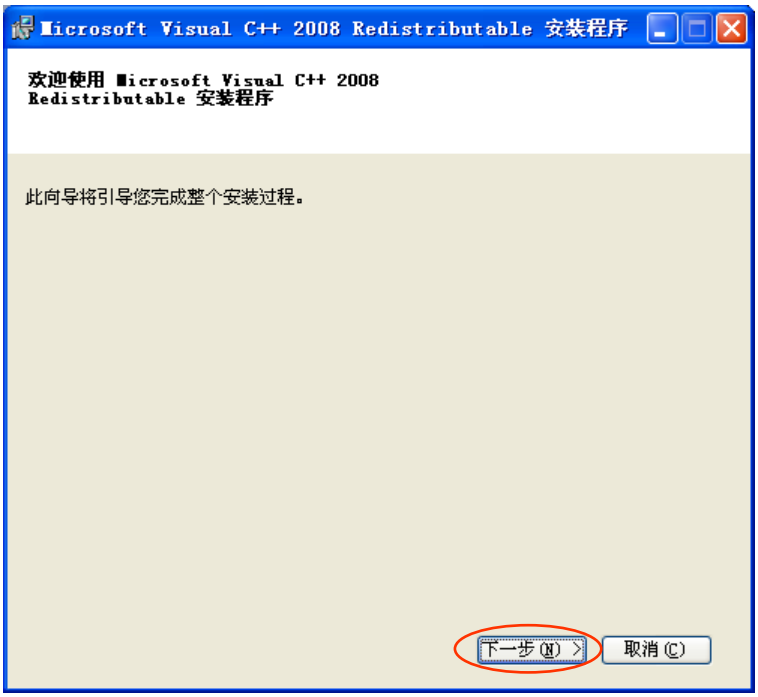

图 2.4　安装实时运行库

在弹出的许可条款对话框中，选择【我已阅读并接受许可条款】，点击【安装】开始安装运行库。

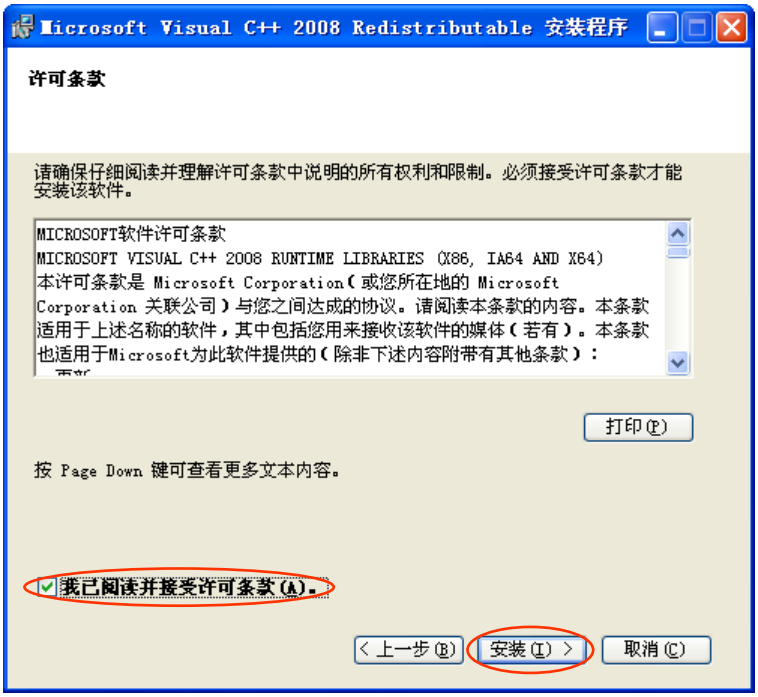

图 2.5　接受许可条款并安装

2.2　安装 USB 驱动

当您第一次把 TKScope 仿真器连接到您的计算时，系统将弹出找到新的硬件向导对话框，如图 2.6。

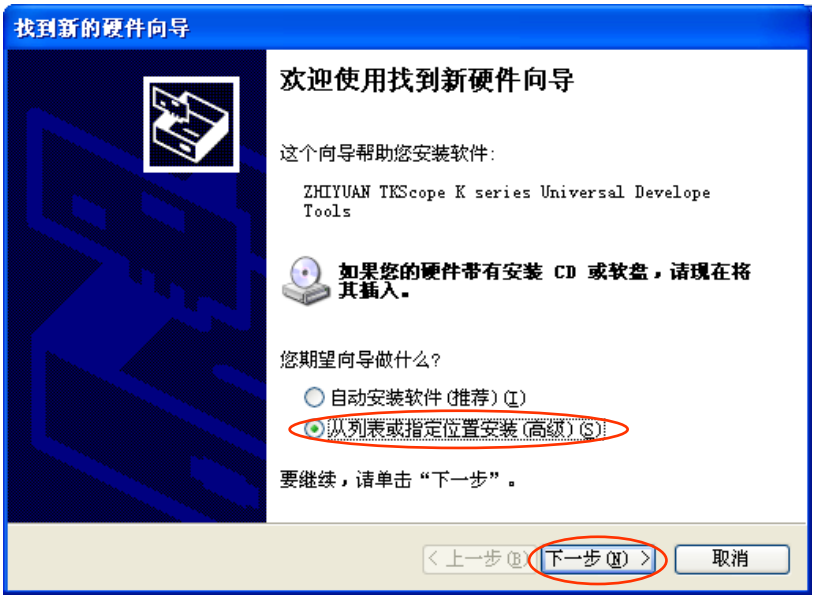

图 2.6　新硬件安装向导

在图 2.6 中选择【从列表或指定位置安装（高级）】，点击【下一步】，系统将弹出如图 2.7 的对话框。

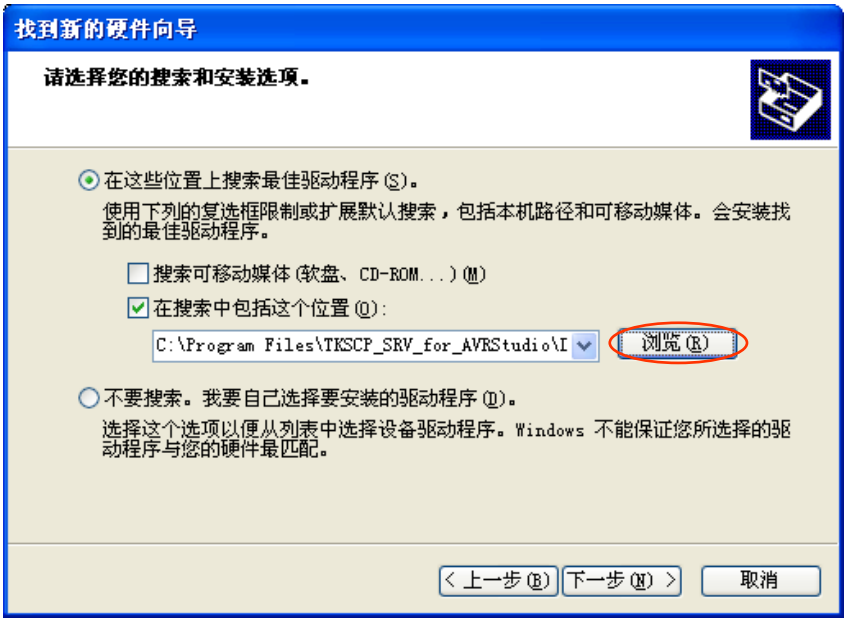

图 2.7　选择驱动程序对话框

在图 2.7 中单击【浏览】，打开驱动程序路径选择对话框，找到 TKScope_AVRStudio 的安装路径（例如：C:\Program Files\TKSCP_SRV_for_AVRStudio\Driver\TKScope K Driver\TKScopeK\WinXP），点击【确定】。

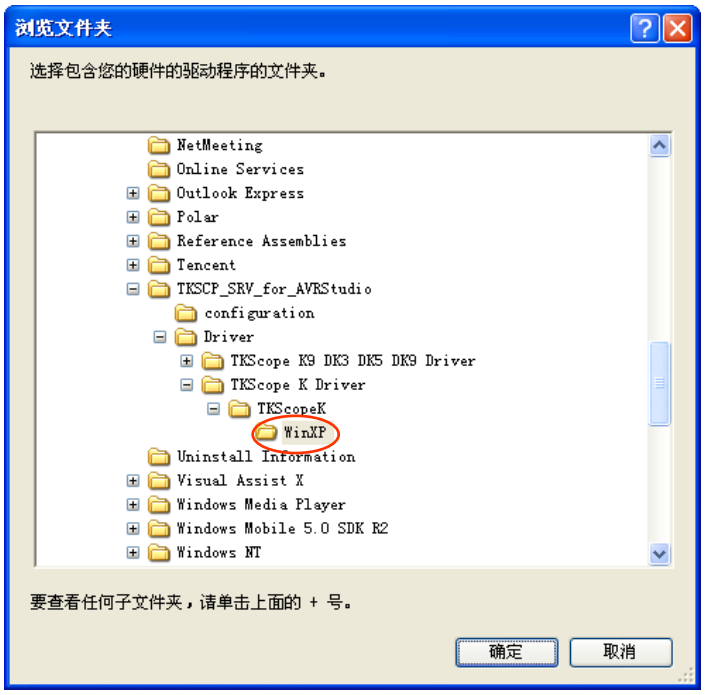

图 2.8　指定驱动程序路径

驱动程序安装完毕后，系统将弹出**错误!未找到引用源。**的窗口，点击【完成】完成安装。

图 2.9　新硬件安装完成

2.3　安装 JTAGICEmkII 驱动

请先确认您在安装 AVR Studio 时已经安装 USB 驱动，在安装 AVR Studio 时选中 Install/upgrade Jungo USB Driver（默认时已选中），如图 2.10。

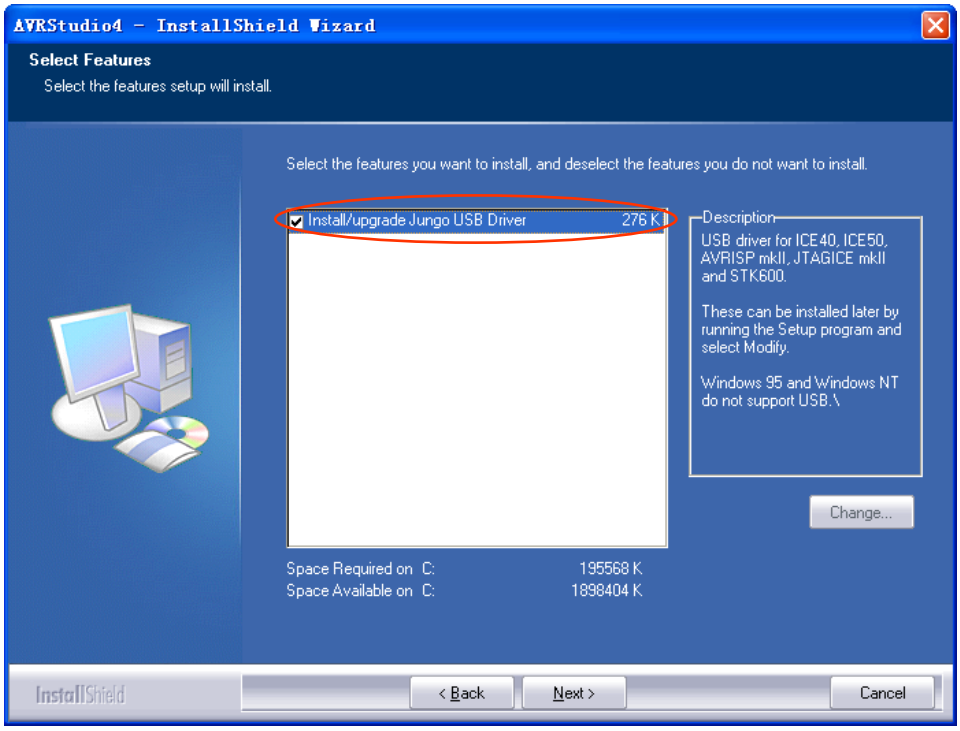

图 2.10　安装 AVR Studio 的 USB 驱动

安装完毕后，你可以在安装 AVR Studio 的安装路径下（如 C:\Program Files\Atmel\AVR Tools）找到名为 usb 的文件夹，该文件夹就是存放 AVR Studio 的 USB 驱动文件。如果您在安装 AVR Studio 时没有没安装 USB 驱动，可以重新安装 AVR Studio 并选择安装 USB 驱动。

启动仿真环境前需先打开仿真器电源并连接 USB，点击 AVR Studio with TKScope，第

一次需安装 JTAG ICE mkII 的驱动程序，如图 2.11，在弹出的对话框中选择【从列表或指定位置安装（高级）】，点击【下一步】。

图 2.11　安装 JTAGICE mkII 驱动程序

选择“搜索中包括这个位置”，并在浏览中选择 AVR Studio 的安装路径下的 usb 文件夹，点击【下一步】。

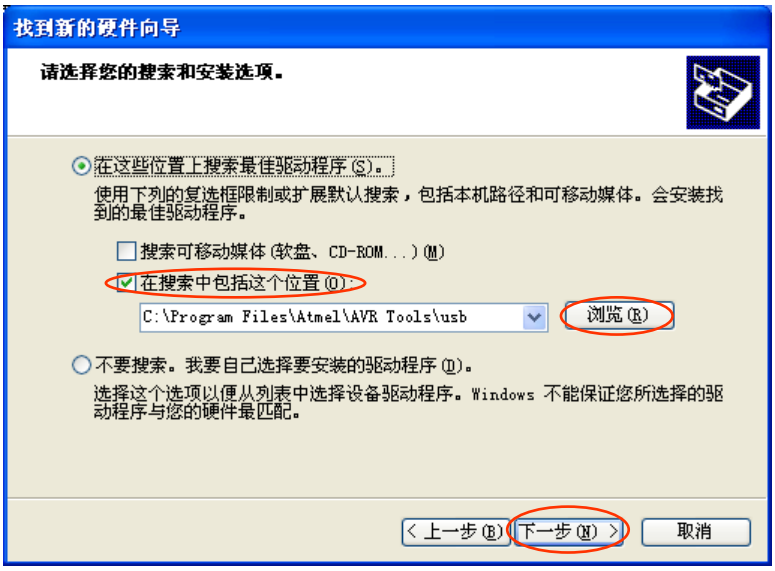

图 2.12　选择路径

等待系统安装完成后，即出现图 2.13 的对话框，点击【完成】即可完成驱动的安装。

图 2.13　完成安装

安装完成后，即会看到任务栏右端 TKScope 服务程序的小图标，如图 2.14，当 AVR Studio 和 TKScope 通信时，此图标将会闪烁。

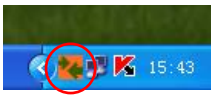

图 2.14　AVR Studio TKScope 服务程序

2.4　开始仿真

在 AVRstudio 下，代码通过编译后，点击开始仿真按钮，如图 2.15，AVR Studio 会根据当前芯片的设置自动选择合适的仿真参数并进入调试。

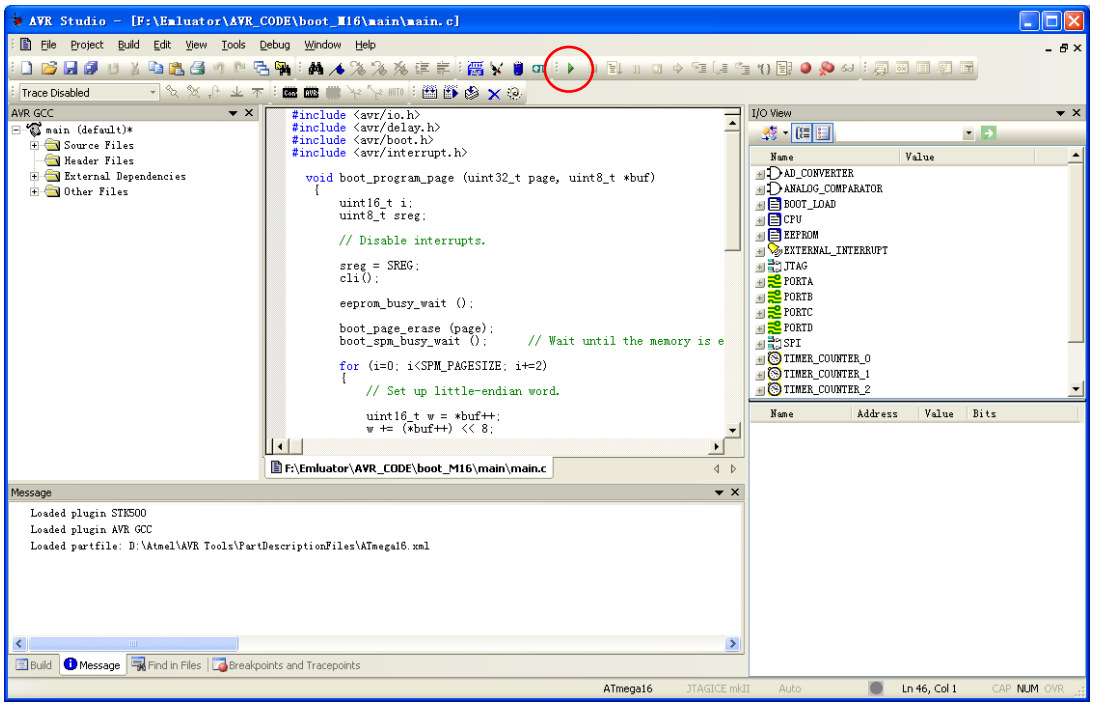

图 2.15　开始仿真

进入调试后，可以在 IDE 上看到调试信息窗口，如图 2.16，这些窗口可以在 View->ToolBars 中打开。

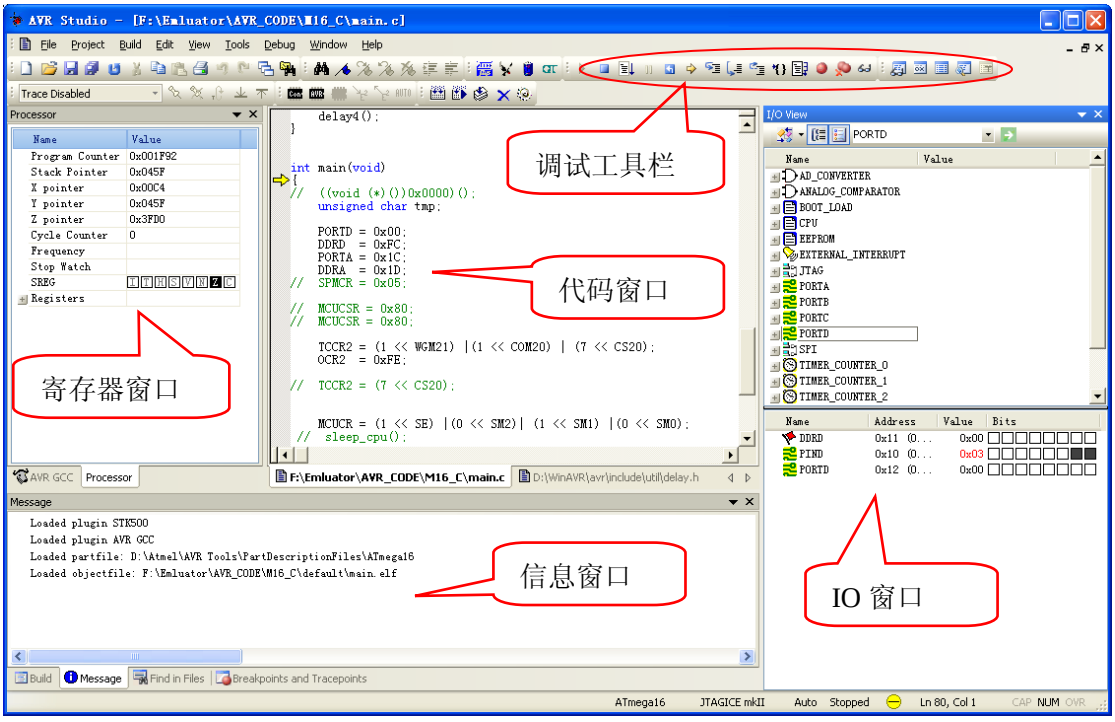

图 2.16　调试窗口

2.5　仿真参数设置

进入仿真环境后，可以在 Debug 菜单下打开 JTAGICE mkII Options 打开仿真器的参数设置窗口，如图 2.17。

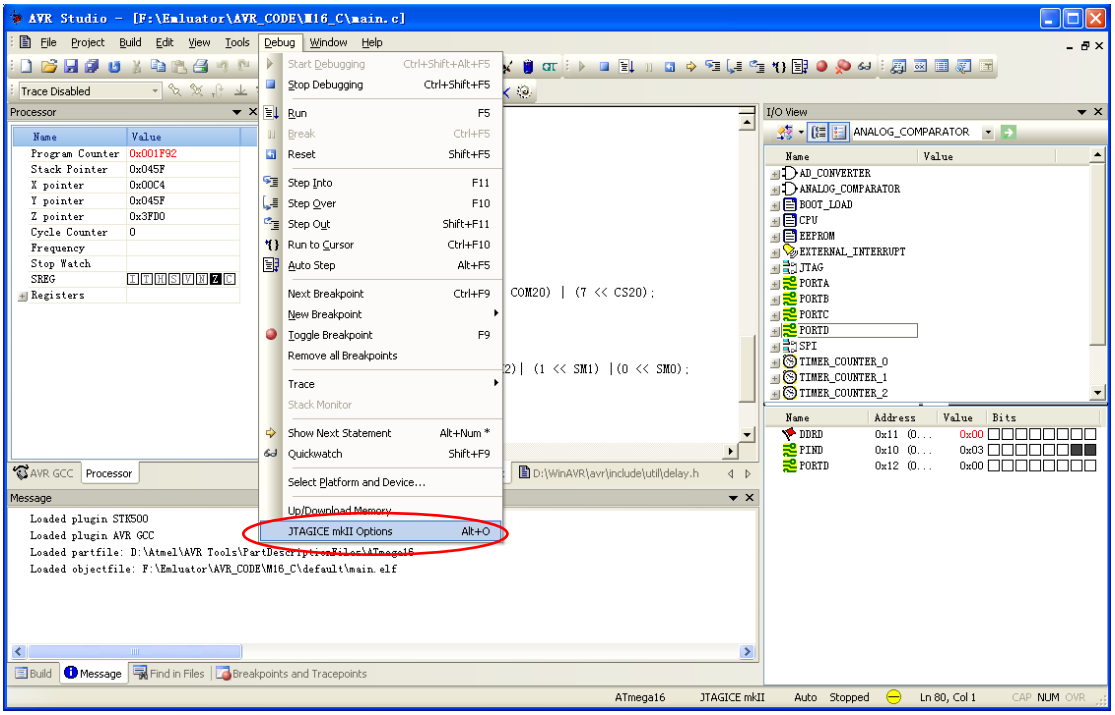

图 2.17　仿真参数设置

设置选项中一共有四个标签，第一个标签为连接选项，用于设置 JTAG 频率和菊花链，AVRStudio 会在连接目标板时自动配置，用户一般不需要更改，更改 JTAG 频率时需注意 JTAG 频率需小于目标器件频率的 1/4。

右下角的 disable debugWIRE 按钮用在仿真 debugWIRE 器件时有效，用于关闭芯片的 debugWIRE 功能，重新使用 ISP。点击这个按钮后，芯片将退出仿真模式。

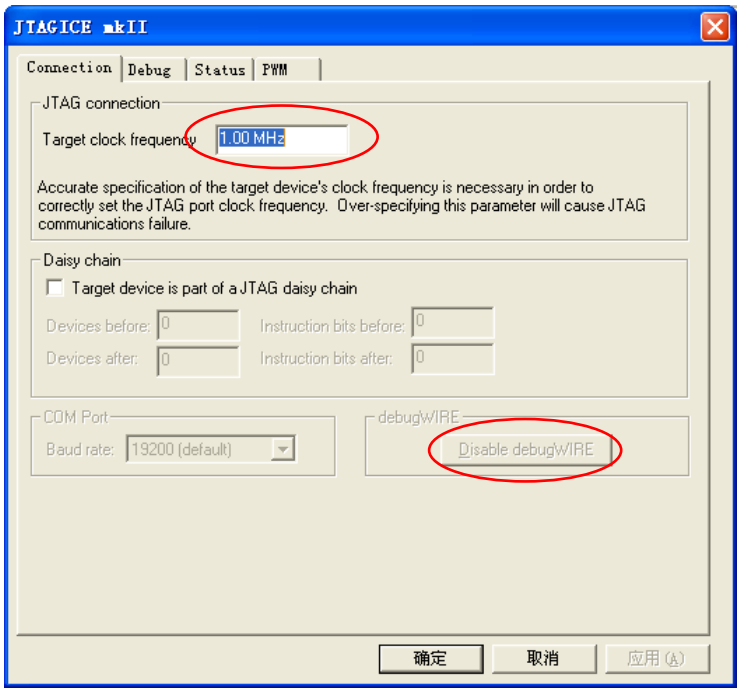

图 2.18　连接设置

第二个标签为调试设置，用于配置仿真时的参数，如图 2.19。

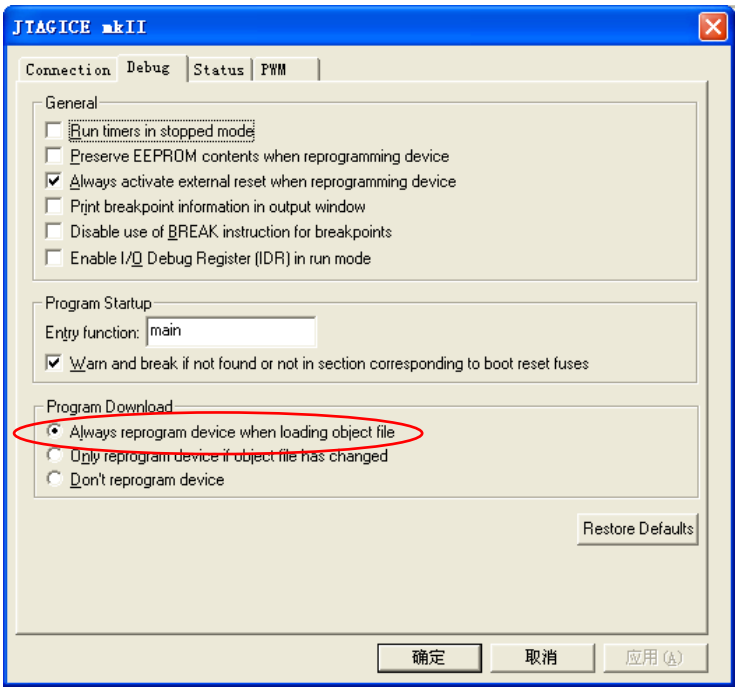

图 2.19　调试设置

General 中各选项含义如表 2.1 所示。

表 2.1 General 调试设置

选项	意义
Run timers in stopped mode	目标停止时继续运行定时器。
Preserver EEPROM contents when reprogramming device	重新编程时保留 EEPROM 内容（通过编程 EESAVE 熔丝实现）。
Always activate external reset when reprogramming device	在编程时总是使能外部复位。
Print breakpoint information in output windows	在输出窗口显示断点信息。
Disable use of BREAK instruction for breakpoints	不使用软件断点指令（这时只能设置 3 个硬件断点）。
Enable I/O Debug Register（IDR）in run mode	允许用户程序访问调试寄存器（IDR）。

Program Startup 用于设置程序入口函数以及找到不入口函数或 Boot Reset 熔丝设置错误是否警告。

Program Download 用于设置进入仿真时是否重新编程 flash，为保证调试的正确性，建议设置为总是编程。

第三个标签为 Status 选项，可以看到仿真器的信息和目标板的信息。

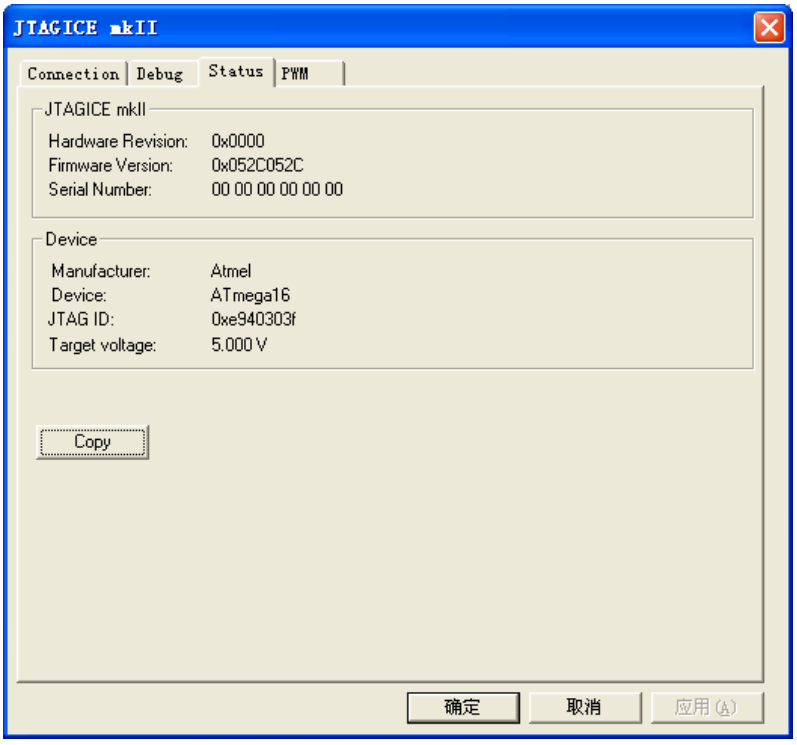

图 2.20 Status

第四个标签为 PWM 选项，仅用于 AT90PWM 器件，用于控制内部 PSC 控制器和模拟比较器在芯片仿真过程中暂停是否继续运行。

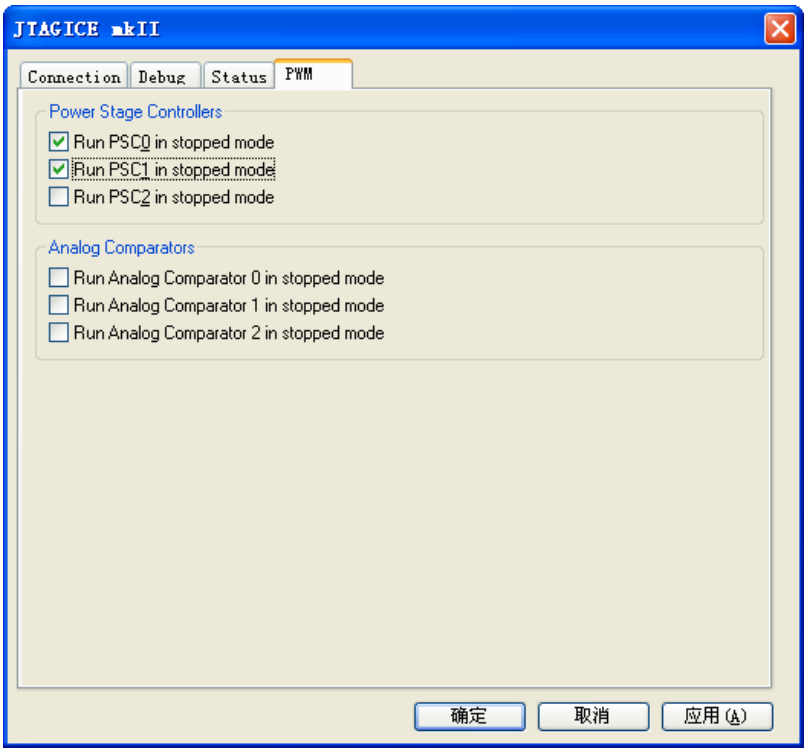

图 2.21　PWM

2.6　仿真调试工具

AVR Studio 的调试工具条如图 2.22 所示。

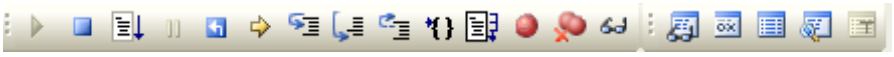

图 2.22　调试工具


开始调试（Start Debugging）


停止调试（Stope Debugging）


全速运行（Run）


暂停运行（Break）


显示当前将要运行的语句（Show Next Statement）


步进（Step Into），单步执行程序，进入到子程序内部


布越（Step Over），单步执行程序，跳过子程序，不进入到内部


步出（Step Out），从子函数跳出


运行到光标处（Run to Cursor）

自动单步（Auto Step），连续执行 Step Into

设置断点（Toggle Breakpoint）

移除所有程序断点（Remove all Program Breakpoints）

快速监察（Quick watch），打开快速监察窗口

监察（Toggle Watch Window）,打开/关闭监察窗口

寄存器窗口（Toggle Register Window）,打开/关闭寄存器窗口

内存窗口（Toggle Memory Window），打开/关闭内存窗口

反汇编窗口（Toggle Disassembler Window），切换源程序和反汇编窗口

2.7 退出仿真

用户仿真结束，直接点退出按钮 即可退出仿真调试状态。

2.8 使用 AVR Studio 的编程工具

使用 AVR Studio 的编程工具可以通过点击”display the ‘connect’ dialog”按钮，如图 2.23，将弹出如图 2.24 的对话框，选择”JTAGICE mkII”，点击“Connect”（注意：连接编程工具前需先退出仿真）。

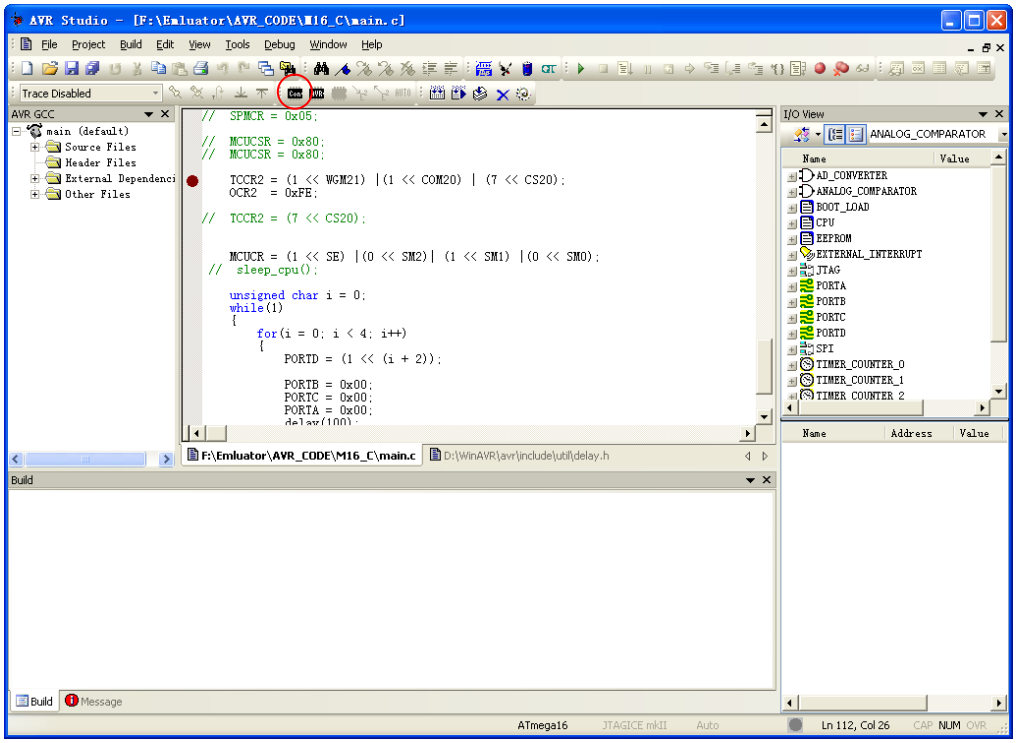

图 2.23 连接编程工具

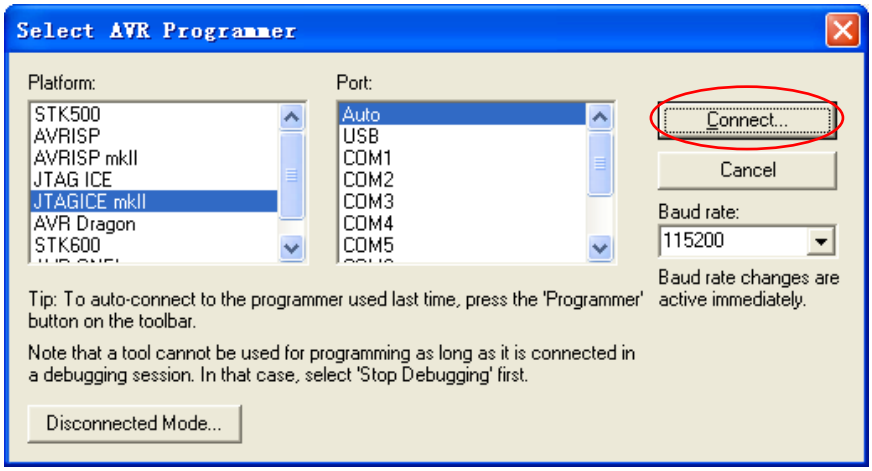

图 2.24　选择 AVR 编程工具

成功连接后将弹出图 2.25 的对话框，在“Device and Sigature Bytes”选择目标器件，“Programming Mode and Target Settings”选择编程方式。注意：使用 ISP 方式时需要使用 ADP-AVR-P10-P6 适配板。通过 6 针的 IDC 电缆与目标板连接。

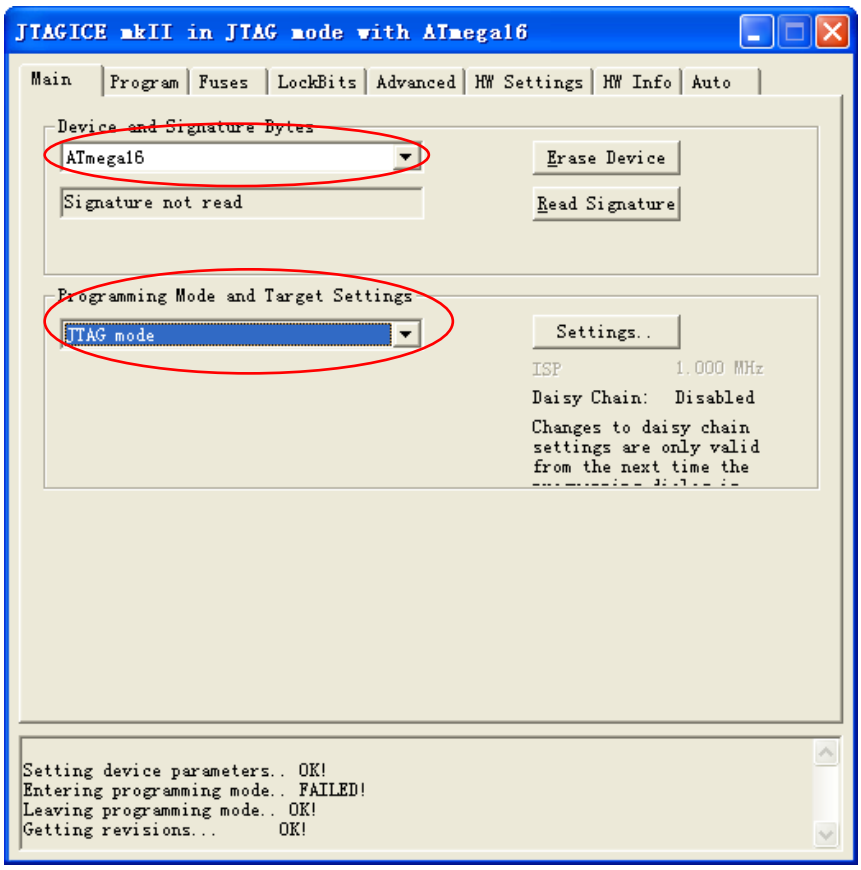

图 2.25　编程界面

在编程界面的其它标签可以进行 Flash、EEPROM、熔丝、锁定位的编程，校准字节的读取和组合操作等功能。

2.9 退出服务程序

退出仿真或编程后，如果用户需要使用 IAR， K-flash 等软件，需要退出服务程序，以解除对仿真器的占用，如图 2.26，右击服务程序的图标，选择 Quit，即可退出服务程序。

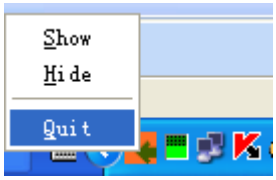

图 2.26 退出服务程序

3. IAR 环境下仿真 AVR 快速入门

3.1 安装驱动

在 IAR 下使用 TKScope 仿真器前需安装驱动程序，否则，无法正常工作！

双击 TKScopeSetup_AVR8.EXE，系统弹出如图 3.1 所示的对话框，按照提示进行安装可。

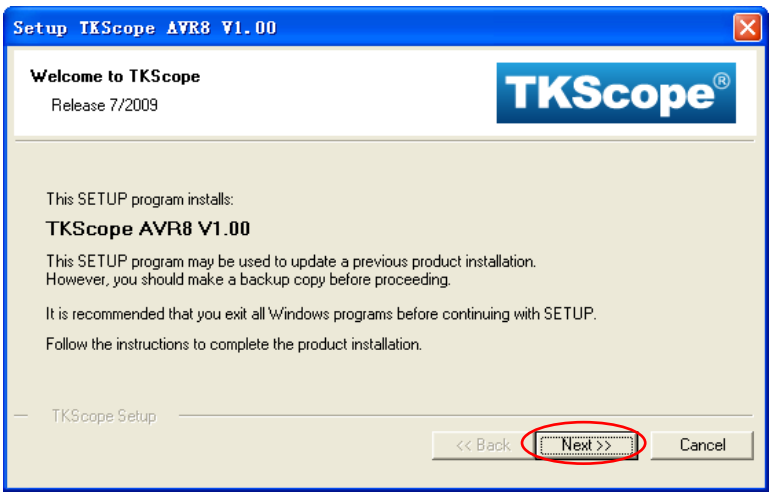

图 3.1 安装仿真器的驱动

驱动可以安装在任意路径，如图 3.2 所示。

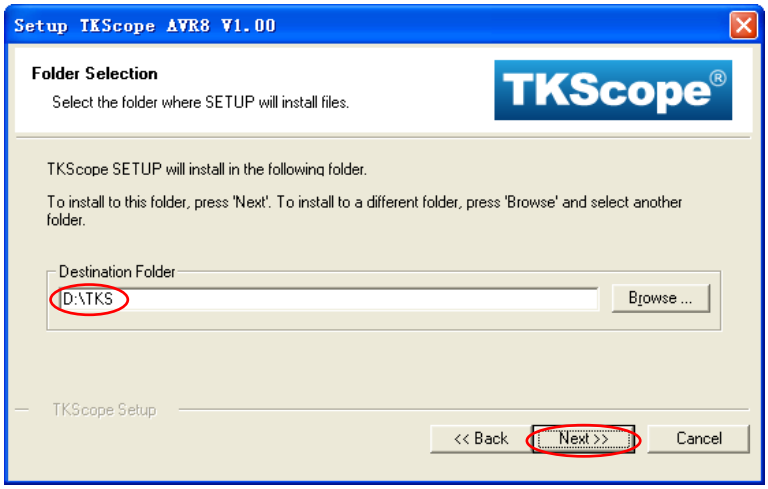

图 3.2 驱动安装路径

仿真器驱动安装完毕，建议用户安装微软的 VC8 的实时运行库。双击 vcredist_x86_CHS.exe，系统会弹出如图 3.3 所示的对话框。点击【是】，系统会自动完成安装。

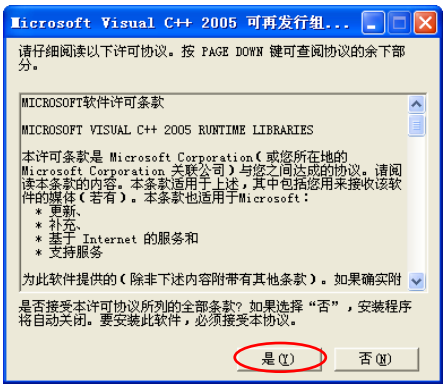

图 3.3　安装实时运行库

在安装目录下（本文示例为 D:\TKS\TKScope），可以看到安装好的.dll 驱动文件，如图 3.4 所示。

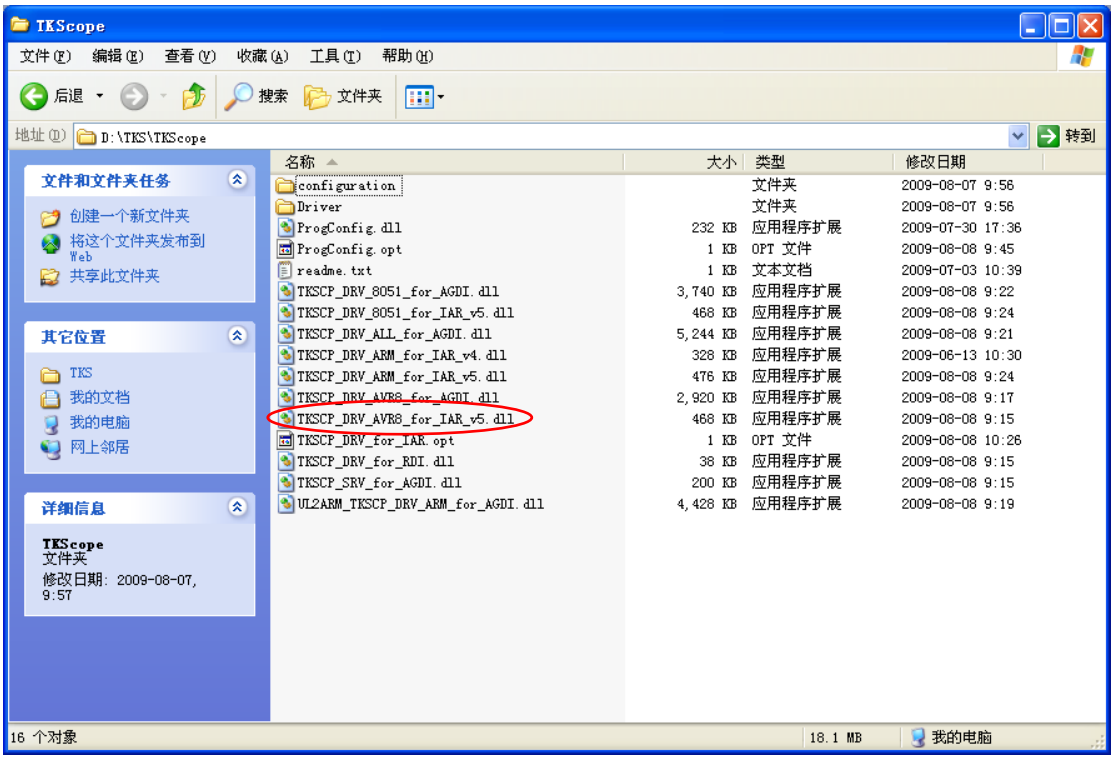

图 3.4　驱动安装目录文件

各个驱动文件的所属类型以及应用的开发环境，用户请详见安装目录下的 readme.txt 文件（本文示例图 3.4 中）。

表 3.1 列举出与 AVR 仿真相关的驱动文件，如若驱动文件有所增减或变动，以 readme.txt 文件为准。

表 3.1　驱动文件列表

驱动名称	驱动类型	应用环境
TKSCP_DRV_AVR8_for_IAR_v5.dll	AVR	在 IAR V5 版本下的驱动

3.2 安装 USB 驱动

如果您第一次把 TKScope 仿真器连接到电脑，需要安装 USB 驱动程序，安装办法可以参照 2.2 安装 USB 驱动，安装驱动程序所需的文件可以在 TKScopeSetup_AVR8.EXE 的安装目录下找到。

3.3 添加驱动文件

按照 3.1 章节驱动安装完成之后，打开一个编译成功的工程，如图 3.5 所示，选中工程，选择【Project】菜单，点击【Option】进入工程选项设置界面。

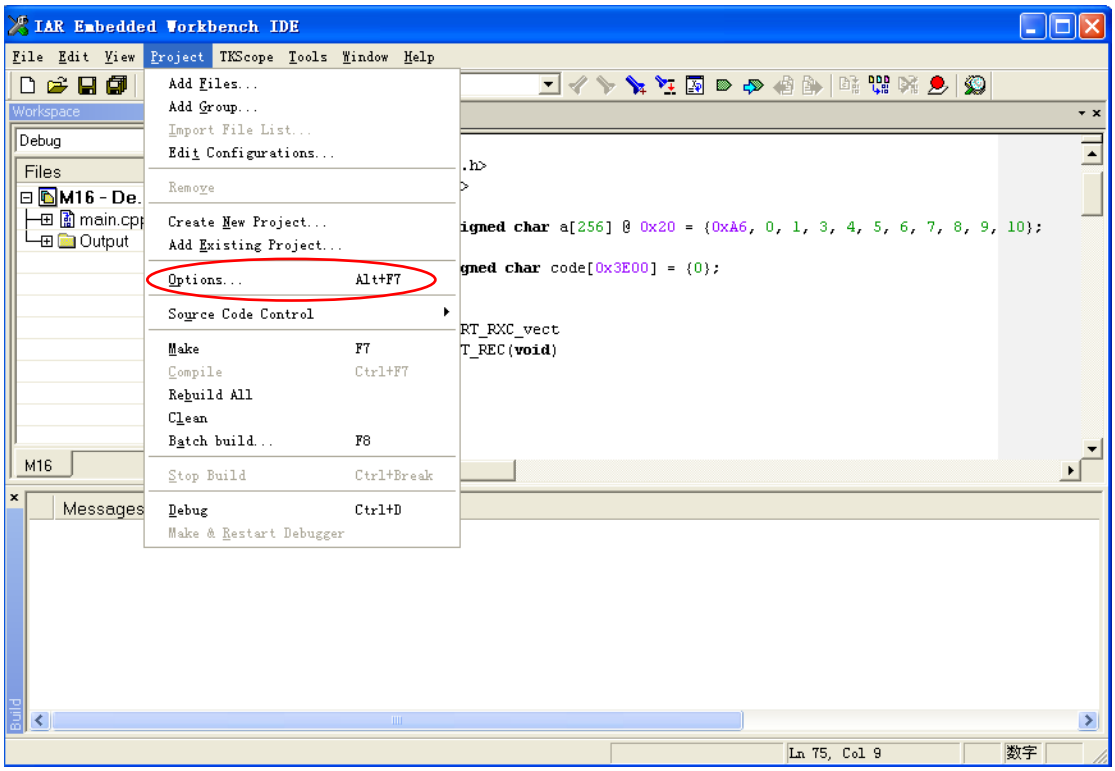

图 3.5 IAR 主界面

在工程配置界面，选择【Debugger】选项，右侧的【Setup】窗口设置如图 3.6 所示。【Driver】选择【Third-Party Driver】，选中【Run to main】。

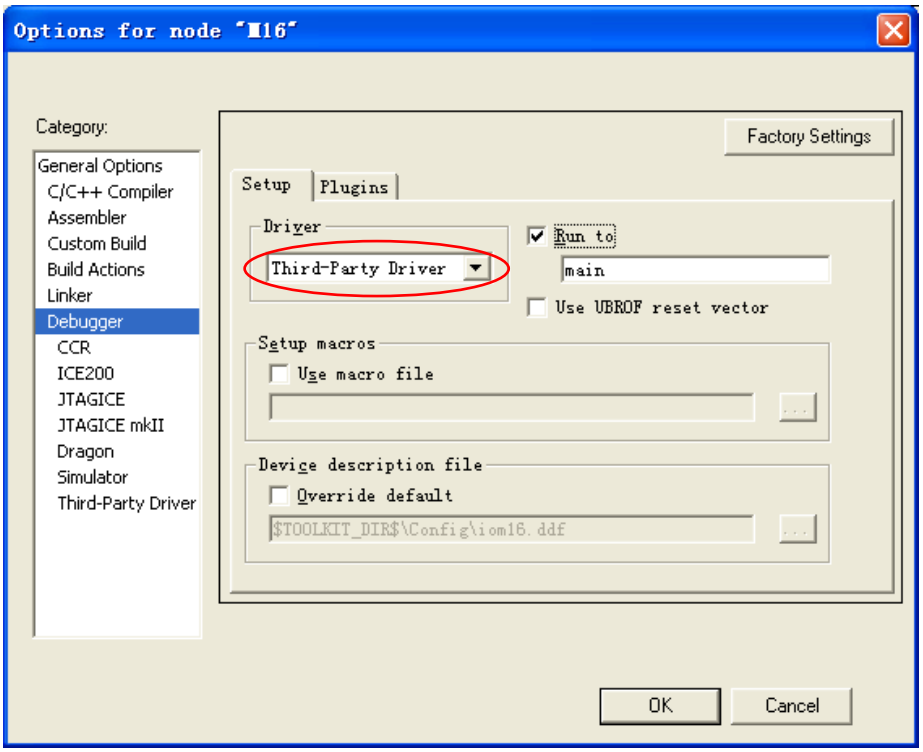

图 3.6　驱动选择

选择【Third-Party Driver】选项，界面如图 3.7 所示。点击 .. 图标，添加仿真器驱动。

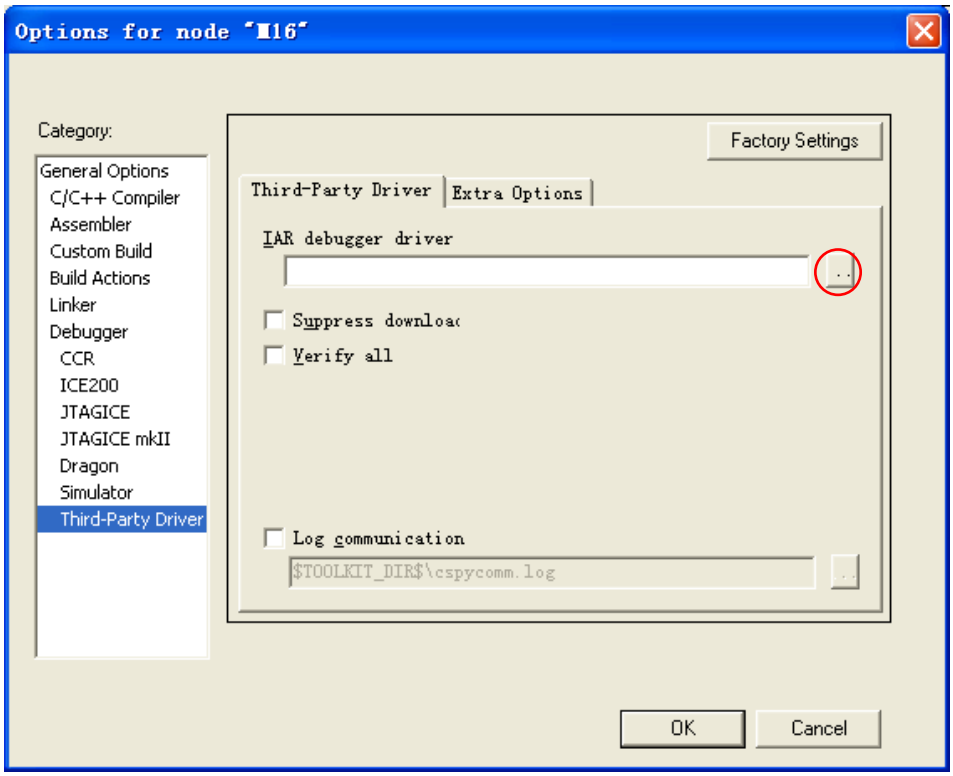

图 3.7　Third-Party Driver 界面

如图 3.8 所示，找到 TKScope 驱动安装目录，选择 TKSCP_DRV_AVR8_for_IAR_v5.dll，点击打开按钮定位驱动文件。

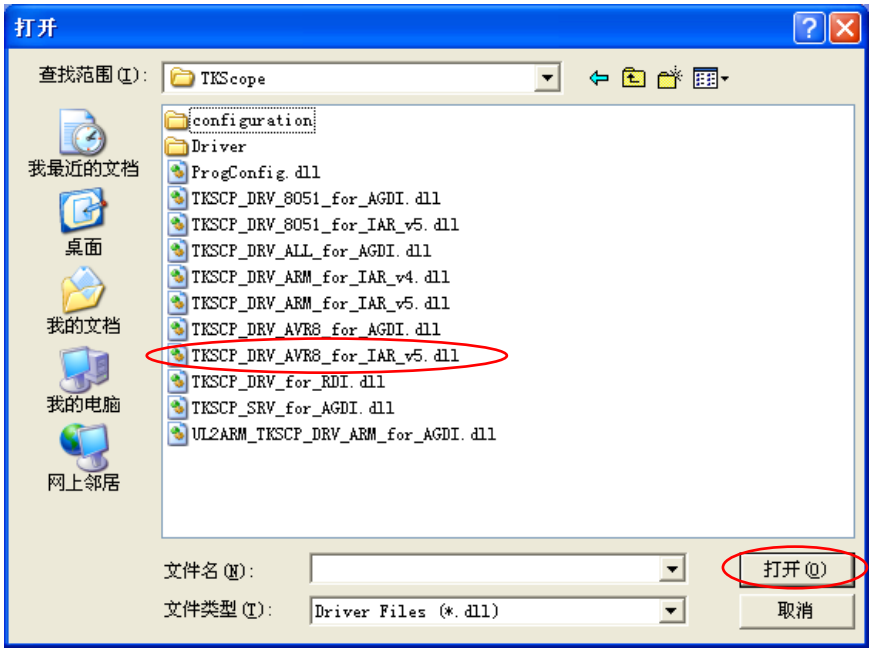

图 3.8　选择 IAR 驱动文件

设置好驱动文件后如图 3.9 所示，点击 OK 按钮设置完毕。

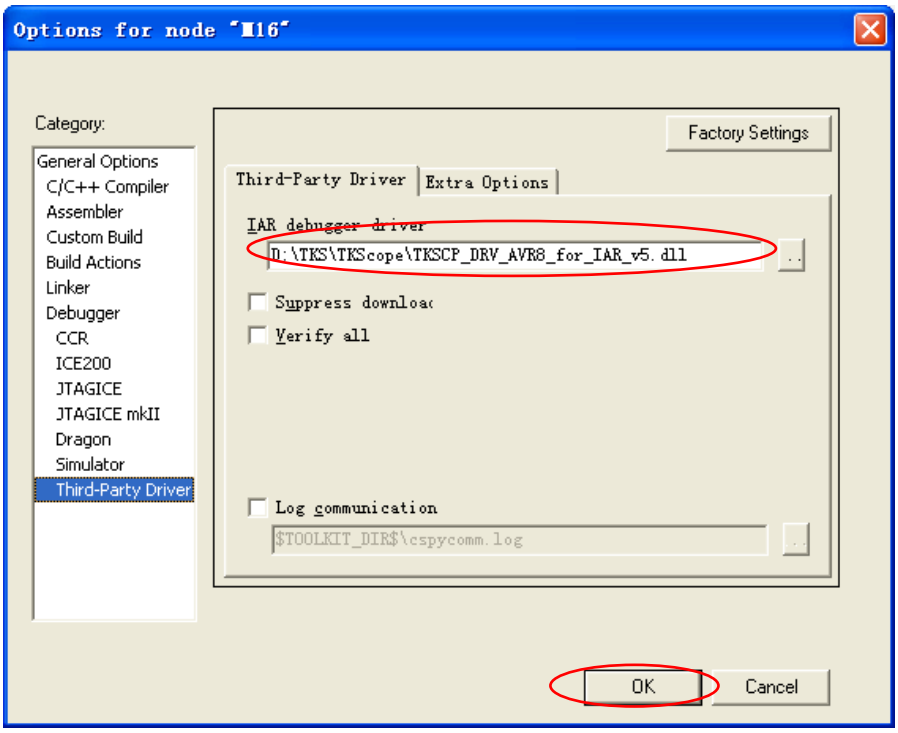

图 3.9　完成加载驱动文件

TKScope 仿真器工作参数必须要正确设置，否则可能会导致仿真错误或失败！如图 3.10 所示，点击菜单【TKScope】选择【Setup】即可进入仿真器参数设置框。

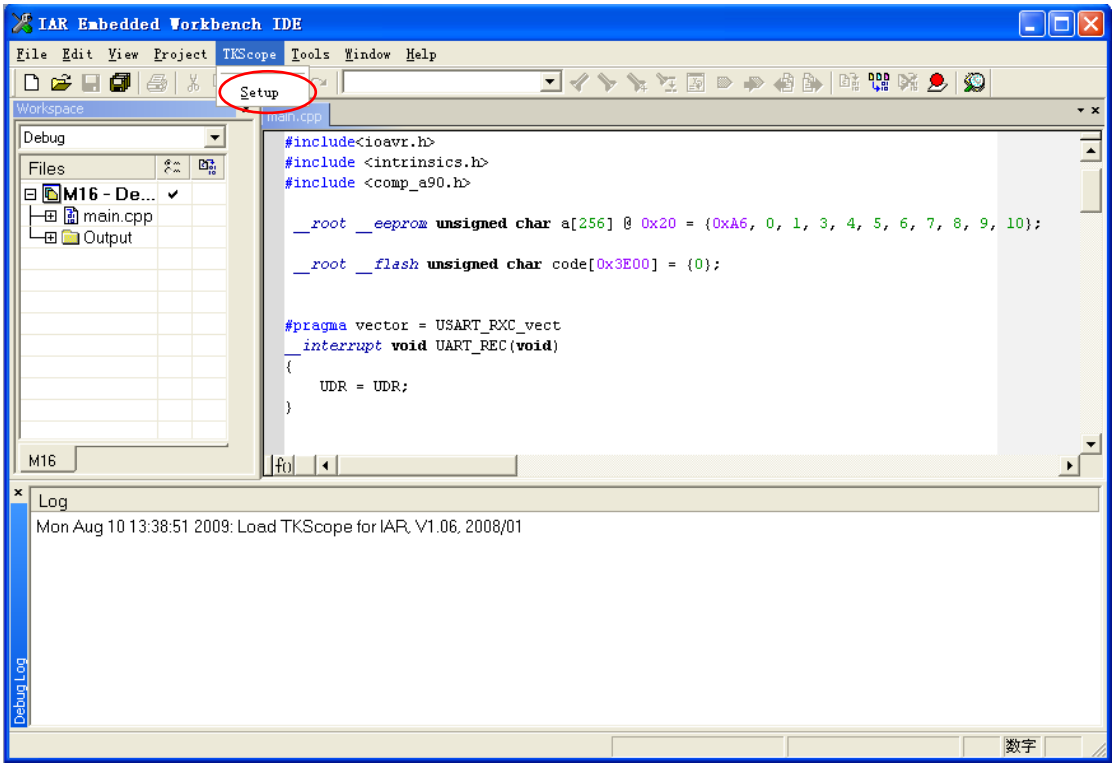

图 3.10　置驱动后主界面

在图 3.8 中，选择硬件仿真，对应的驱动选择【TKSCP_DRV_AVR8_for_IAR_v5.dll】。然后，点击【Settings】进入 TKScope 仿真器设置界面，如图 3.11 所示。

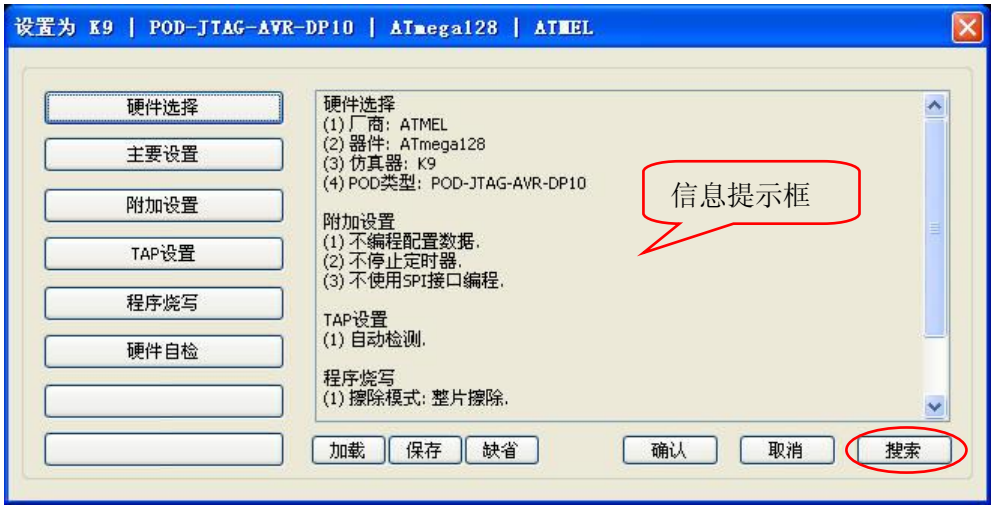

图 3.11　仿真器设置界面

在图 3.11 中，点击左侧的各个选项，系统会弹出相应的设置界面，同时右侧的信息提示框中会出现各项设置信息的具体含义。

3.4 仿真器参数设置

3.4.1 硬件选择

点图 3.11 中的【硬件选择】，进入如图 3.12 所示的界面。

图 3.12　硬件选择界面

用户根据实际仿真芯片型号，在列表中选中所需要仿真或编程的器件，点击【确定】返回到图 3.11 的界面。

友情提示：可以通过在右上的器件过滤框内（图中红色框处）输入器件的名称，让您快速找到器件。

选择好器件后，点击图 3.11 中的【搜索】，系统会自动列出当前电脑已连接的所有仿真器和仿真头的信息。如果您的电脑连接有多个仿真器，需要在这里选择您要使用的那台仿真器，点击【确定】保存设置。

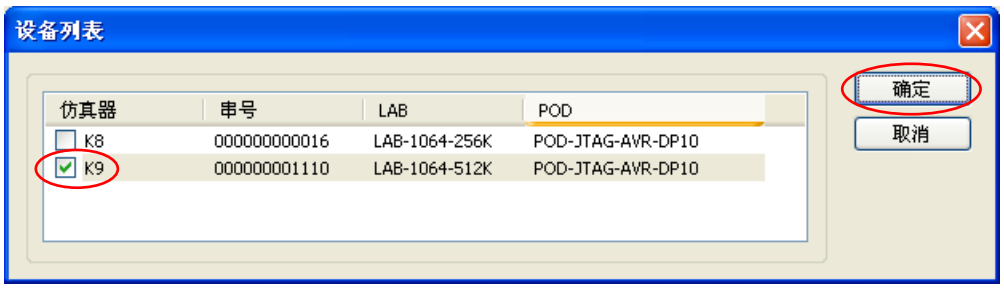

图 3.13　设备列表

3.4.2 主要设置

点击图 3.11 中的【主要设置】，进入如图 3.14 所示的界面，用户根据仿真需要设置即可。

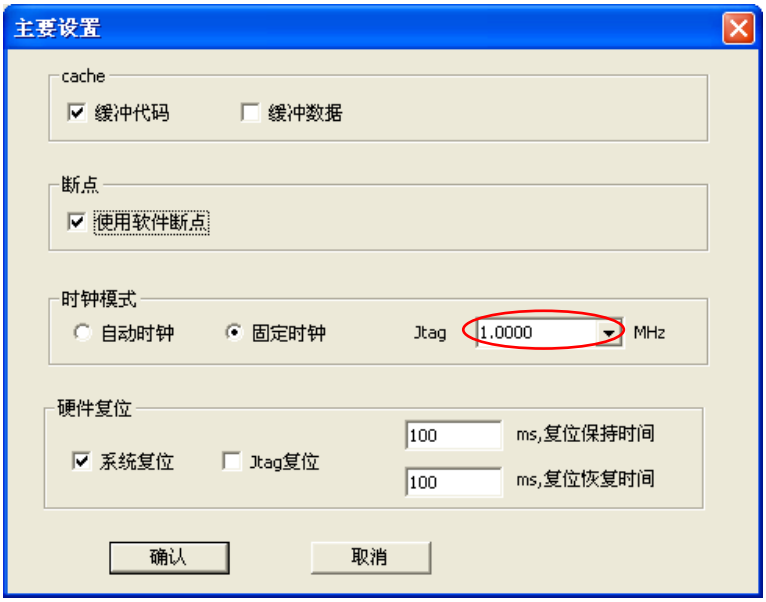

图 3.14　主要设置界面

1.　显示缓存配置

缓存配置（Cache）是解决屏幕刷新和仿真速度的矛盾。

如果选择 Cache，屏幕显示刷新只在用户程序运行后进行，可加快显示速度。但是，如果某一个操作引起的其它数据的变化可能不能及时显示。

如果不选择 Cache，则用户在 PC 端的任何操作都将引起显示数据的重新刷新。在查找不稳定硬件时比较理想，但是屏幕刷新会影响操作响应速度。

【缓存代码】：用户代码缓存。

【缓存数据】：非用户代码缓存。

建议用户使用缺省配置，仅选择【缓存代码】。

2.　断点

【使用软件断点】：选中此项，则可实现 Flash 中无限制断点调试。

3.　时钟模式

【自动时钟】：自动选择复位后最高的可用时钟。

【固定时钟】：选择用户输入的时钟频率数值。

【JTAG 时钟】：选择 JTAG 时钟频率，仅在固定时钟下有效。

友情提示：AVR 器件的仿真要求使用固定时间，JTAG 时钟频要求不能大于系统时钟的 1/4。

4.　复位选择

【系统复位】：使用硬件复位 nSRST。

【JTAG 复位】：使用硬件复位 nTRST，AVR 器件没有 JTAG 复位引脚，此选项仅保留将来使用。

【复位保持时间】：选择复位有效时期的延迟时间，单位 ms。

【复位恢复时间】：选择复位结束后的延迟时间，单位 ms。

3.4.3 附加设置

点击图 3.11 中的【附加设置】，进入如图 3.15 所示的界面。

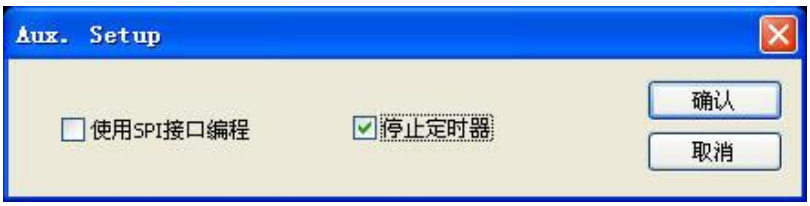

图 3.15　附加设置界面

【使用 SPI 接口编程】：该选项在 IAR 仿真环境下不起作用，主要用于编程时接口的选择。编程时，要使用 K-Flash 编程软件（文中第 4 节会详细介绍），该选项决定使用哪种接口编程，不选中时使用 JTAG 接口，选中后使用 SPI 接口；debugWIRE 接口仿真的器件不管是否选中都是使用 SPI 接口编程。

【停止定时器】：进入监控模式后是否停止定时器运行。

3.4.4 TAP 设置

点击图 3.11 中的【TAP 设置】，进入如图 3.16 所示的界面。

图 3.16　TAP 设置界面

TAP 设置选项用于设置 JTAG 菊花链的器件参数，包括器件个数、顺序、IR 长度、当前仿真器件。

当扫描链中包含未知的器件或存在多个可以仿真的器件时，用户必须进行设置，设置时尽量参考自动生成的扫描链参数（可以应付绝大多数的情况）。

【自动检测】：根据已登记的器件自动配置扫描链，不能完成时启动人工配置。

【人工配置】：每次配置前启动该配置窗口，用户手动选择。

【器件列表】：扫描链中存在器件的参数，包括 IDCODE、器件名称、IR 长度。

【IDCODE】：输入的器件 IDCODE 数值（16 进制），用于添加或更新器件。

【器件名称】：输入的器件名称。

【IR 长度】：器件的 IR 长度，数值不能为 0。

【添加】：在器件列表中添加一个新的器件，IDCODE、器件名称、IR 长度必须指定。

【删除】：在器件列表中删除当前器件。

【更新】：更新器件列表中当前器件的参数。

【上升】：在器件列表中，将当前器件的位置上移。

【下降】：在器件列表中，将当前器件的位置下移。

友情提示：只有 JTAG 方式的仿真的器件才能进行菊花链级联，可以在一个 JTAG 口上连接多个器件。其它方式仿真的器件（如 debugWIRE 和 PDI）不能进行级联，不需要进行 TAP 设置。

3.4.5 程序烧写

点击图 3.11 中的【程序烧写】，进入如图 3.17 所示的界面。

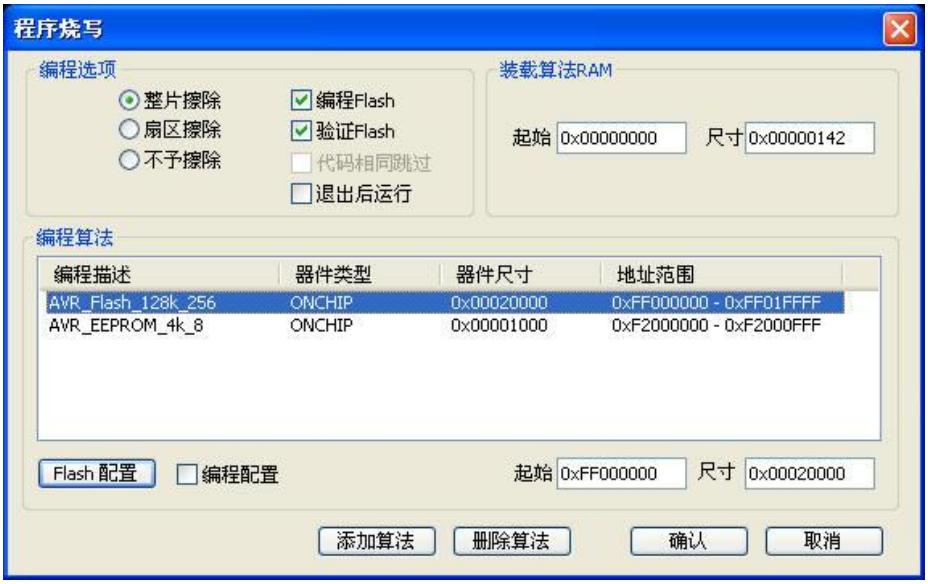

图 3.17　程序烧写

1. 编程选项

【整片擦除】：程序下载到 Flash 前全部擦除 Flash 空间；

【扇区擦除】：根据下载需要擦除相应的扇区空间；

【不予擦除】：下载代码不进行擦除操作；

【编程 Flash】：是否打开编程 Flash 功能，选中后才能进行 Flash 编程；

【验证 Flash】：在编程的过程中同时校验 Flash；

【代码相同跳过】：与上一次烧写的代码进行比较，相同的代码不需要重新烧写，只烧写改变的代码，这样可以提高烧写的速度。

用户如果需要编程 Flash，必须选中【编程 Flash】、【验证 Flash】选项，同时选择【整片擦除】或【扇区擦除】。

2. 装载算法 RAM

【起始】：运行算法的 RAM 起始地址，一般不需要修改；

【尺寸】：当前 RAM 的尺寸，给定的数值满足需要；

友情提示：对于 AVR 器件，算法是在 Flash 上运行的，起始地址需要为 0。

3. 编程算法

【Flash 配置】：Flash 相关信息的配置。点击【Flash 配置】选项，进入如图 3.18 所示的配置窗口，用户根据实际需要对芯片进行配置；

【编程配置】：选中时在编程 Flash 过程中连同 Flash 配置一起编程；

【起　　始】：当前算法文件的偏移地址；

【尺　　寸】：当前算法文件的用户修改尺寸，不能大于原始文件中的尺寸。

该区域显示的为当前加载的 Flash 编程算法文件。

【Flash 配置】不同芯片有不同的配置，下面是以 ATmega128 为例子的配选项。配置的方法可参考芯片数据手册。第一个标签用于设计熔丝位，如图 3.18。

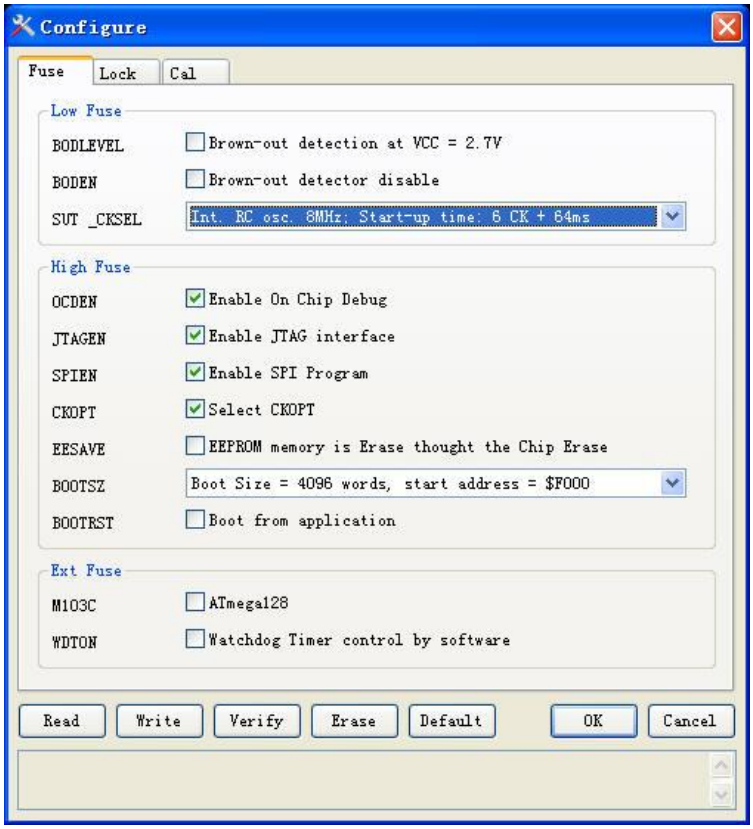

图 3.18　编程配置 Fuse

第二个标签为 Lock，锁定位设置，用于编程锁定位，如图 3.19。

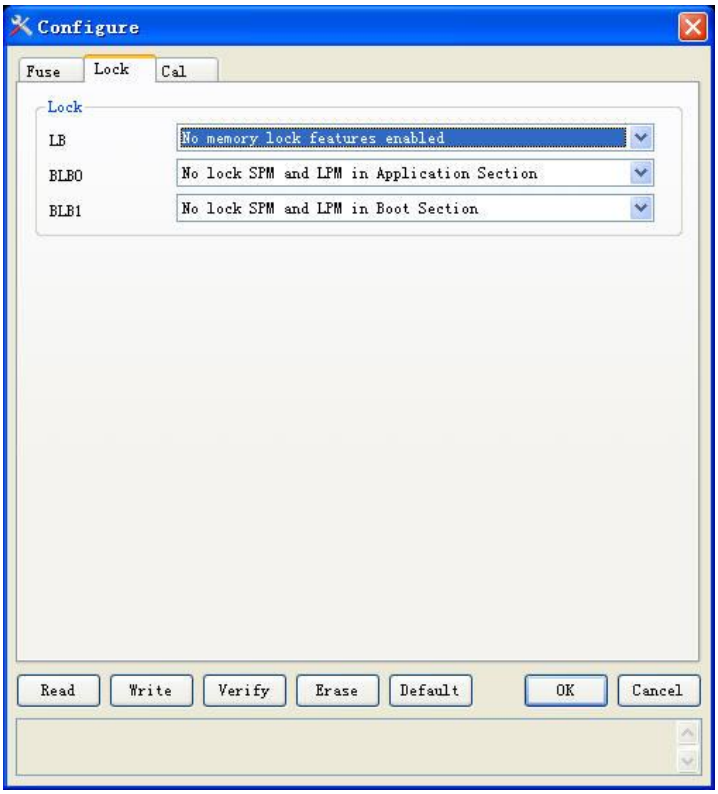

图 3.19　编程配置 Lock

最后一个标签 Cal，用于读取片内 RC 振荡器的效准字节和把校准字节编程到 Flash 或 EEPROM 的固定地址中。

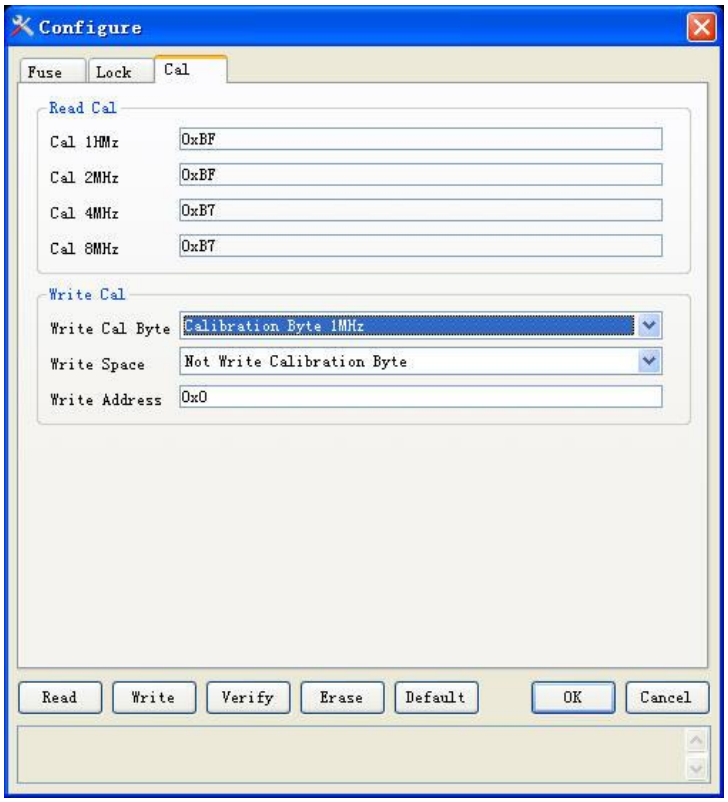

图 3.20　编程配置 Cal

在各个标签的下方提供了 Read, Write, Verify, Erase, Default，OK, Cancel 共 6 个按钮，各个按钮的功能如表 3.2。

表 3.2　编程配置各按钮的功能

按钮	功能
Read	从目标板读取编程配置数据并刷新界面
Write	把当前的编程数据写入的目标板中
Verify	从目标读取编程配置数据并与界面的数据对比。
Erase	把芯片擦除（熔定位编程后将无法编程熔丝位，擦除后可以将芯片擦除再次对熔丝位编程）
Default	把编程配置信息还原到默认值
OK	退出并保存当前配置信息
Cancel	退出不保存配置信息

【程序烧写】是比较重要的一项设置，直接关系到仿真能否成功以及仿真结果是否正确。

友情提示：

（1）程序烧写的各个配置选项都是针对当前算法的，如果用需要编程多个算法，需要要根据实际情况对各个算法进行配置；

（2）在 IAR 环境下只能烧写芯片内部的 Flash 和 EEPROM，烧写外部 Flash 需要通过 K-Flash 软件，在第 4 节详细地介绍。

3.4.6 硬件自检

点击图 3.11 中的【硬件自检】，进入如图 3.21 所示的界面。

硬件自检是非常实用的一项功能，可以用来检测仿真器与计算机、目标板的通讯情况。用户在使用过程中，遇到联机通信失败的情况，可以利用硬件自检功能来判断故障产生原因。

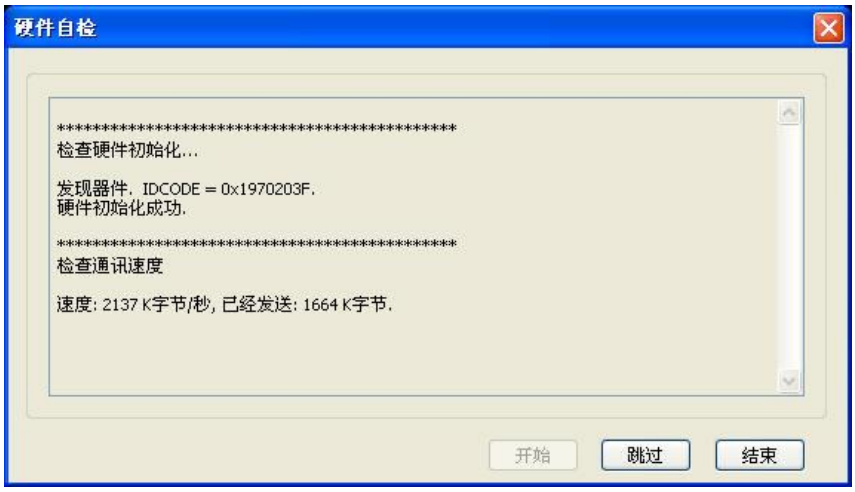

图 3.21　硬件自检界面

3.5　仿真调试

仿真器参数配置完毕后，就可以开始进行仿真调试。

3.5.1 仿真调试工具

在 IAR 调试环境下，安装驱动并正确设置工作参数之后，就可以进行仿真调试。IAR 运行调试工具条如图 3.22 所示。

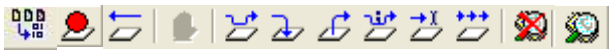

图 3.22 运行调试工具条

 调试（Debug），进入调试环境

 编译（Make）。

 设置断点（Toggle Breakpoint）。

 复位（Reset）。

 停止运行（Break）。

 布越（Step Over）。单步执行程序，跳过子程序，不进入到内部。

 步进（Step Into）。单步执行程序，进入到子程序内部。

 步出（Step Out）。执行到当前子函数的结束。

 运行到下一条语句（Next Statement）。程序运行到下一条语句时停止。

 运行到光标（Run to Cursor）。程序运行到当前光标所在行时停止。

 全速运行（Go）。

 退出调试（Stop Debugging）。

设置好相关参数和驱动后，如图 3.23 回到 IAR 主界面，先点击编译按钮，编译成功后再点击调试按钮进入调试环境。

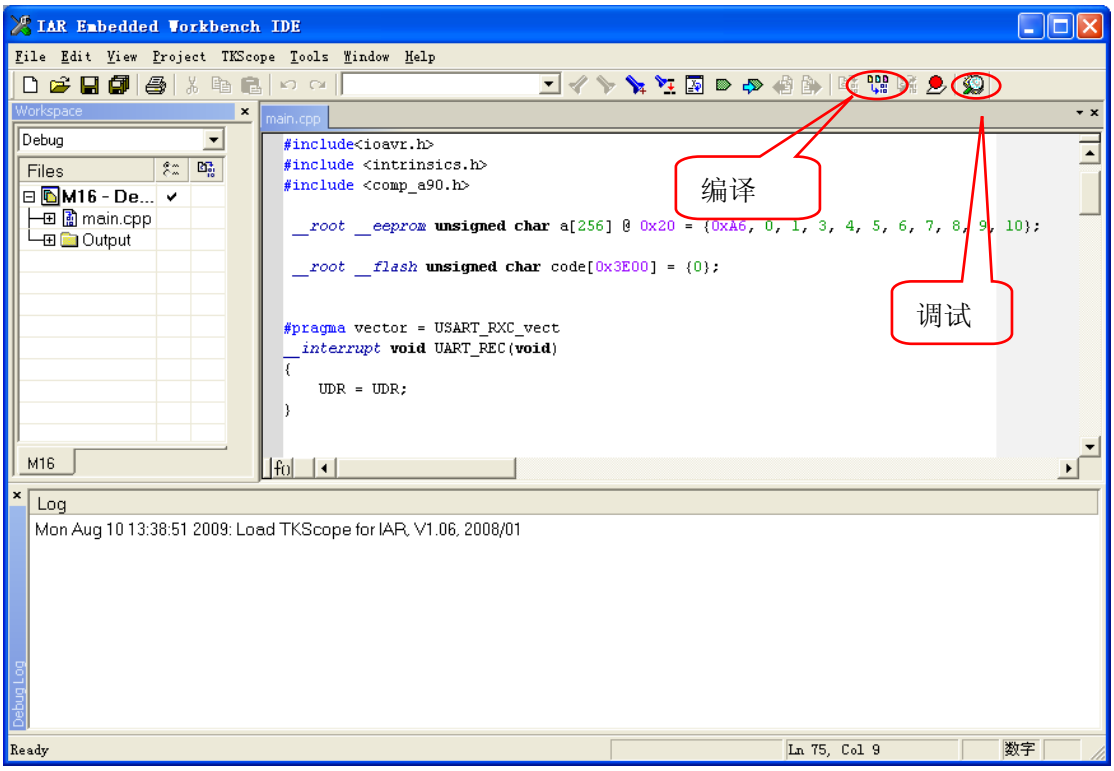

图 3.23　设置驱动完毕

进入调试环境后如图 3.24 所示，点击【View】菜单选择【Register】即可查看寄存器。其它查看选项操作类似。

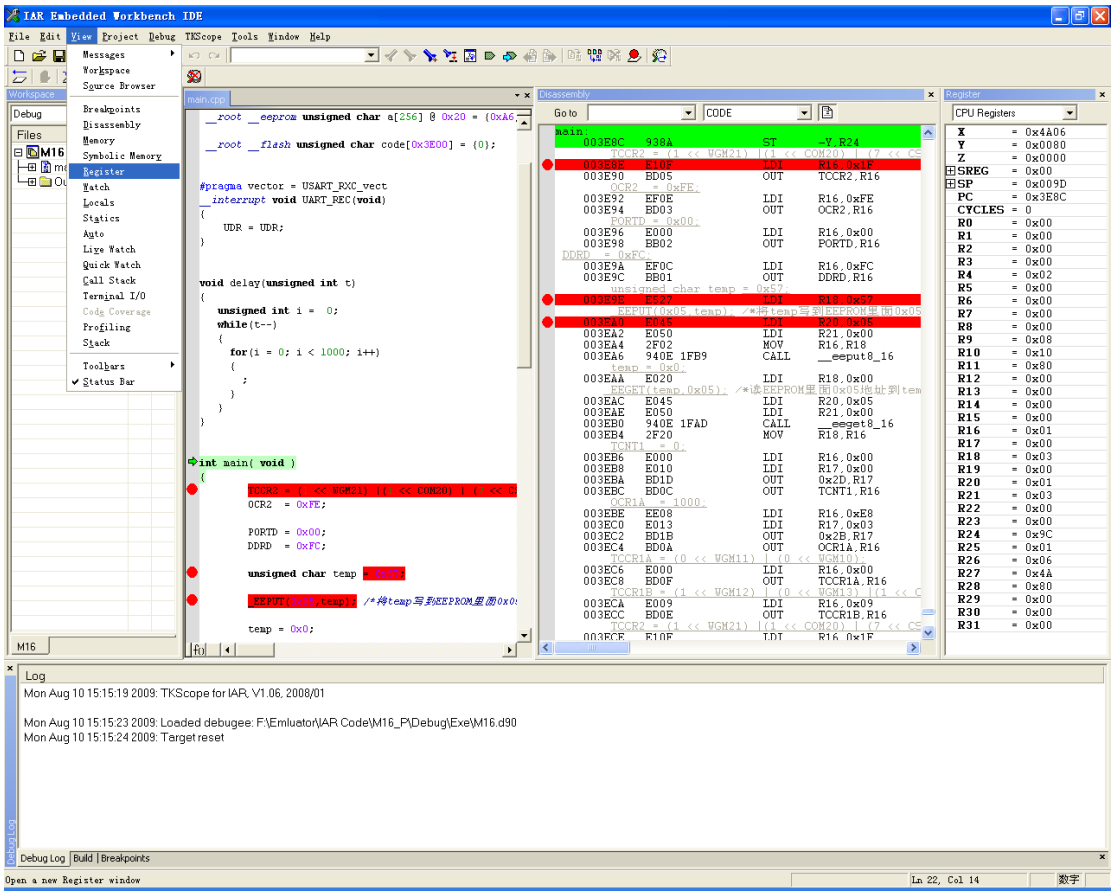

图 3.24　调试观察窗口工具条

在 IAR 仿真环境中，可以通过 Memory 窗口修改用户代码，也可以在目标运行时动态设置断点。

3.5.2　仿真调试结束

用户仿真结束，直接点退出按钮  即可退出仿真调试状态。

4. K-Flash 软件在线编程 AVR 的内/外部 Flash

4.1 K-Flash 软件简介

K-Flash 是一款用于 Flash 烧写，支持 TKScope 系列仿真器，可实现 Flash 器件在线烧写、擦除、读取等操作。K-Flash 软件操作简单，使用方便，可大大提高在线量产编程的生产效率。图 4.1 为 K-Flash 软件操作界面。

图 4.1 K-Flash 主界面

K-Flash 软件具有如下的主要功能及特点：

- 【烧　　写】：可烧写 bin 文件、hex 文件、out 文件、elf 文件等多种类型的文件；
- 【校　　验】：读取 Flash/EEPROM 的数据，与烧写文件比较，检验烧写是否正确；
- 【烧写校验】：先进行烧写，完成再检验；
- 【擦　　除】：擦除指定扇区内的数据；
- 【查　　空】：检查擦除操作是否完成；
- 【读　　取】：读取指定的起始地址和大小的数据并保存到用户指定的路径；
- 【设备配置】：配置仿真器类型、芯片参数、Flash 编程算法等。

K-Flash 软件支持工程管理的模式，形成工程文件后，相关的设备配置等信息会自动保存。这样用户再次进行操作时，无需繁琐重复的参数设置，直接打开工程文件即可。

4.2 仿真器配置 Flash 方法

本文主要介绍通过 TKScope 仿真器对 Atmega128 芯片内部 Flash、EEPROM 和外部 Flash（AT45DB161D）器件进行一键式的在线编程。

在执行文件烧写之前，先要对仿真器进行设置。点击图 4.1 的【设备配置】，进入仿真

器的设置界面，如图 4.2 所示。

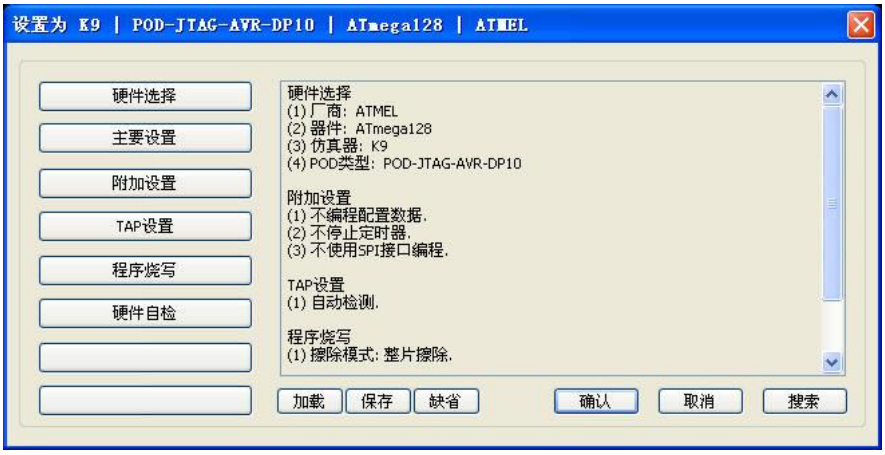

图 4.2　仿真器设置界面

点击图 4.2 的【硬件选择】，进入如图 4.3 界面。在右边的器件过滤窗口中输入 Atmega128 芯片，在左边窗口系统自动找到 Atmega128 芯片，选择芯片下的具体仿真器，点击【确定】按钮即可。

图 4.3　硬件选择

点击图 4.2 的【主要设置】，进入如图 4.4 界面。在编程时，注意：时钟模式要选择【固定时钟】，JTAG 时钟大概设为 1MHz，JTAG 时钟频要求不能大于系统时钟的 1/4。

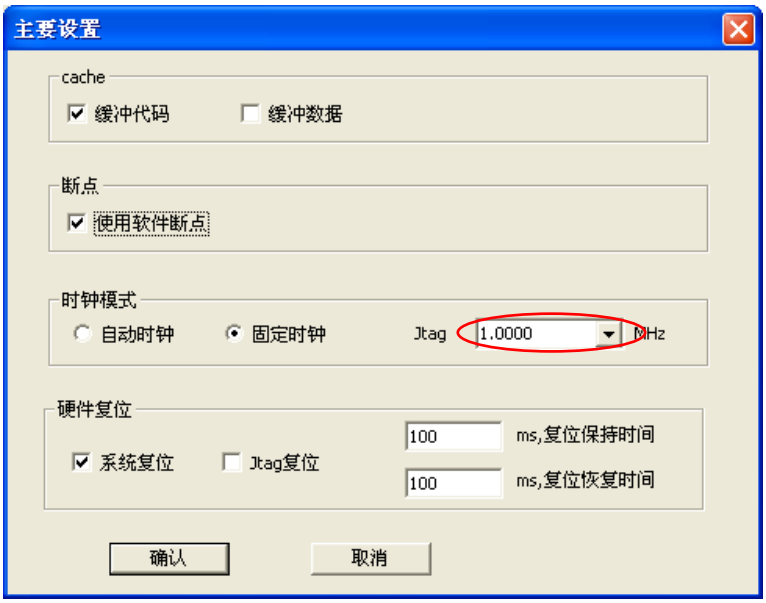

图 4.4　主要设置

点击图 4.2 的【附加设置】，进入图 4.5 的界面。【使用 SPI 接口编程】决定使用哪种接口编程，不选中时使用 JTAG 接口，选中后使用 SPI 接口。

图 4.5　附加设置

点击图 4.2 的【程序烧写】，进入图 4.6 的程序烧写界面。

图 4.6　程序烧写 1

选择了 Atmega128 芯片后，系统会自动将 Atmega128 的算法文件调出来，从图 4.6 的编程描述中很直观地看出，Atmega128 具有 128K 的片内 Flash 和 4K 的片内 EEPROM。如果将程序烧写到片外 Flash（本文使用的片外 Flash 器件为 AT45DB161D），点击【添加算法】，

具体算法存放的路径为：…\K-Flash\configuration\ATMEL，找到 AT45DB161D 对应的算法文件，添加进来，如图 4.7 所示。

图 4.7　程序烧写 2

用户如果将程序烧写到 Flash 中，必须选中【编程 Flash】、【验证 Flash】选项，同时选择【整片擦除】或【扇区擦除】。用户还可以进行【Flash 配置】，具体参考 3.4.5。

至此，设置已经完成，返回到图 4.1 的 K-Flash 的主界面。在烧写文件的下拉菜单中，可以显示出所有烧写算法，如图 4.8 所示，通过选择算法文件，加载相应的烧写文件。

图 4.8　烧写算法选择

选择烧写算法之后，点击烧写文件的 … 按钮，弹出一个文件打开对话框，在该对话框中选择将要烧写的文件，如图 4.9 所示。

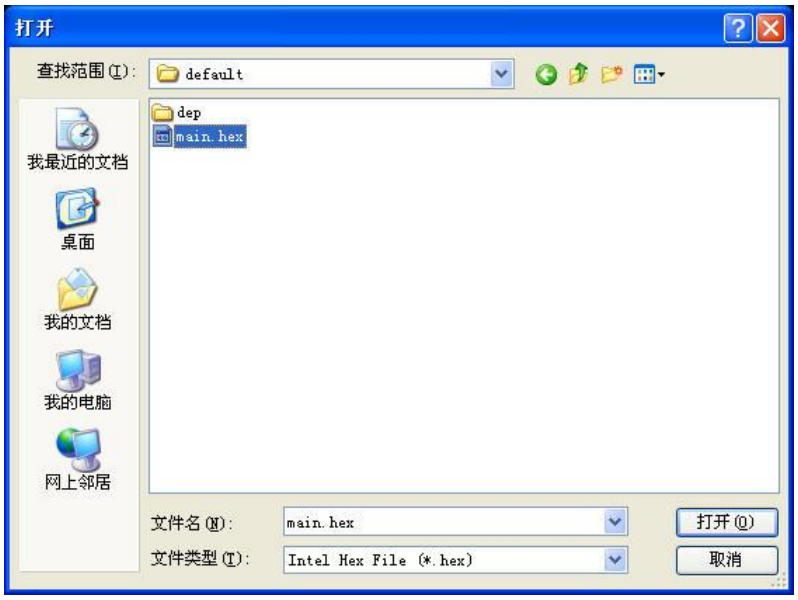

图 4.9　选择要烧写的文件

当选择完烧写文件，会弹出设置烧写数据地址对话框，如图 4.10 所示。可以在该对话框中对数据的烧写地址和烧写尺寸进行设置。

图 4.10　设置烧写数据地址

- 默认烧写地址：表示烧写文件的开始地址。如果烧写的文件是 bin 文件，默认烧写地址为 0。对于其他类型的文件，默认烧写地址表示数据本身自带的开始地址。
- 默认烧写尺寸：表示当前烧写文件的尺寸。

如果想抽取文件的一部分数据进行烧写，可以通过设置抽取数据地址和抽取数据尺寸来实现。

- 抽取数据地址：表示抽取数据的开始地址。地址可以是八进制、十进制、十六进制。注意：抽取数据地址不能小于默认烧写地址。
- 抽取数据尺寸：表示抽取数据的尺寸。注意：抽取数据尺寸不能大于默认数据尺寸。比如一个文件的默认烧写地址为 0x0，结束地址为 0x12c。当需要抽取地址从 0x5 到 0xFF 的数据进行烧写时，则抽取数据地址设置为 0x5，抽取数据尺寸设置为 0xFB。如果想对整个文件进行烧写，则不需要设置抽取数据地址和抽取数据尺寸。

- 烧写目标地址：表示数据烧写到 Flash 的地址。
- 目标地址表示：当修改抽取数据地址、抽取数据尺寸和烧写目标地址时，目标地址表示就会发生变化。它反映了抽取的数据相对于默认烧写地址和烧写目标地址的位置信息。

当需要重新设置目标地址，可以点击目标地址的 ··· 按钮，这是会弹出设置烧写数据地址对话框。可以在弹出的对话框中设置目标地址。

如果只对 Atmega128 的片内 128K Flash 进行烧写时，选择算法[AVR_Flash_128K_256]，按照以上步骤添加烧写文件即可；如果要同时烧写 Atmega128 芯片内部 Flash、EEPROM 和外部 Flash（AT45DB161D），则每个算法添加相应的烧写文件，如图 4.11 所示。

图 4.11　添加烧写文件

正确完成以上两项设置之后，点击【烧写】按钮，K-Flash 对 Atmega128 芯片内部 Flash、EEPROM 和外部 Flash（AT45DB161D）进行一键式的在线烧写，主界面的左上角出现相应的动画，同时出现烧写进度条，如图 4.12 所示。

图 4.12　烧写进度条

烧写完成后，烧写进度条消失，同时，左上角出现烧写完成图标，如图 4.13 所示。

图 4.13　烧写成功

至此，K-Flash 软件完成了 AVR 片内外的 Flash 在线烧写，对于仿真过程中有 Flash 烧写需求的用户，提供了非常便捷的操作方式。这是 TKScope 仿真器独一无二的特性，目前市面上所有的仿真器都无法实现这项功能。此外，K-Flash 还支持校验、擦除和读取等操作，具体参考文档《K-Flash 在线编程软件用户使用指南》。

修订历史

版本	日期	原因
V1.00	2009/09/03	创建文档
V1.01	2010/08/11	支持全系列 8 位 AVR 单片机仿真和编程
V1.02	2011/02/10	增加支持 JTAG 菊花链功能
V1.03	2011/08/09	增加 K-Flash 软件在线编程 AVR 的内/外部 Flash