

简介

欢迎你选择 AVR JTAG 仿真器。AVR JTAG ICE 是单片机爱好者网站推出的与 ATMEL 公司 JTAGICE 相兼容的 AVR 单片机仿真器。对于具有 JTAG 接口的 AVR8 位单片机来讲，JTAG 片内调试系统是一个完美的工具，它可以对所有具有 JTAG 接口的 AVR 单片机进行实时在线仿真。

JTAG 接口是具有 4 线测试存取通道(TAP)的控制器，完全符合 IEEE1149.1 规格要求。IEEE1149.1 已经发展成为具有能够有效地测试电路板连通性的测试标准（边界扫描）。Atmel 公司的 AVR 器件扩展了它一些功能，包括能够支持完整的编程和调试。

AVR JTAG ICE 使用标准的 JTAG 接口，使用户能够对目标芯片进行实时在线仿真。AVROCD 在芯片调试协议使用户对 AVR 单片机的内部资源能够进行完全的实时监控。AVR JTAG ICE 使我们只需花费传统仿真器价格的一小部分就能进行完全的仿真。

JTAG 仿真器的特点：

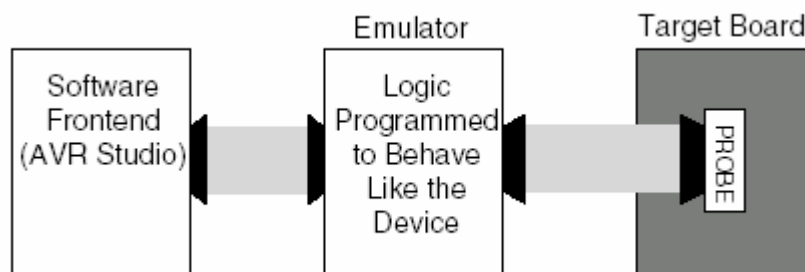
- 完全兼容 AVR Studio （要 4.09 以上版本）
- 支持所有具有 JTAG 接口的 AVR 系列单片机
- 完全真实的实现 AVR 的所有的电性能
- 可以完全实现片内的数字或模拟功能
- 可以在程序执行过程中实现暂停（break）
- 具有数据或程序空间断点
- 支持汇编和高级语言开发
- PC 通过 RS232 口进行编程和控制
- 使用 5VDC 电源

本 JTAG 仿真器需要 4.09 版本 以上 AVR Studio 支持，由 ATMEL 公司免费提供，具体下载和升级信息请访问 www.atmel.com

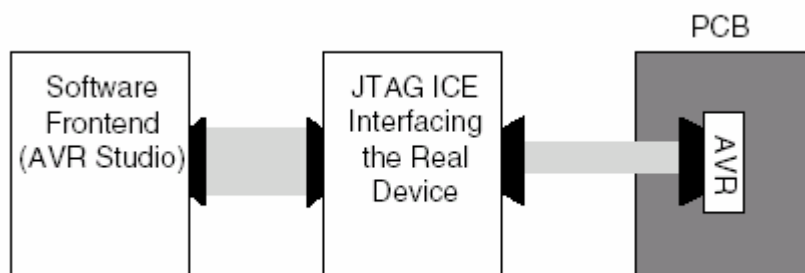
JTAG 片内调试系统的概念：下面将介绍 JTAG 接口的片内调试系统（On Chip Debug）的概念以及于其他的在线仿真器（In Circuit Emulator）有什么区别

仿真器同片内调试系统对比

在线仿真器是一个用来仿真一个或一组器件特性的设备。例如要仿真 mega32，在线仿真器要通过设定使自己模拟 mega32 的特性。其特性越是接近实际中的 mega32 仿真的效果就好。下面是一个典型在线仿真器的简单框图。



JTAG 仿真器通过不同方法实现。JTAG 仿真器联系的内部片上调试系统存在于目标板的 AVR 处理器上。使用兼容 IEEE1149 标准的 JTAG 接口监视和控制运行中的实际器件,可以这样理解 JTAGICE 的工作。所有具有 JTAG 接口的 AVR 单片机都有一个使用 JTAG 界面的片内调试系统，片内调试系统可以控制器件的运行，就像传统仿真器模拟仿器件特性一样。JTAG 仿真是控制实际器件的外设和代码执行，下面是一个片内调试系统 OCD 框图



使用 JTAGICE 可以精确实现器件的时序和电特性。但另一方面，一些传统仿真器可以实现的一些对器件内部资源附加控制，在 JTAGICE 中无法做到。在代码调试过程中，知道一些 JTAG 的操作是必要的。在下面的内容里，将会说明一些注意事项

运行模式：

代码完全独立自主地运行不受 JTAGICE 的影响，JTAGICE 会连续检查是否执行到断点，在遇到断点是，OCD 会打断程序运行，并把所有可能需要的信息通过 JTAG 接口由 JTAGICE 传送给 AVR Studio。由于代码独立运行不受 JTAGICE 控制，所以在断点发生前的状态是未知的。

停止模式：

在执行到断点时，OCD 会将 CPU 挂起，但 IO 仍在运行，例如在一个 USART 发送过程中发生断点，传统仿真器能把整个系统挂起，发送过程也会停止，在使用 OCD 时，USART 将继续运行直至整个发送完成。

断点：

AVR 的 OCD 具有两类断点：硬断点和软断点

硬断点：

在 OCD 系统中由四个用于存储程序存储器地址的计数器，他们每个可以存储一个地址。由于 JTAG ICE 使用要用一个寄存器存储单步断点，只有三个寄存器的组合可用于硬件断点。由于软件断点要重写整页的 flash 数据，在断点经常改变时推荐使用硬件断点

软断点：

软断点时在 flash 中写入一个断点指令，当在 AVR Studio 中发出一个断点指令后，就有一个打断指令写入 flash，当程序执行到此处时就会挂起
要继续运行就必须由 OCD 系统发出一个开始指令，一旦可是继续运行，在 flash 中的程序运行前，断点指令会重新放置

IO 寄存器

JTAGICE 在观察 IO 寄存器时的限制

在 AVR 器件运行到断点时，OCD 系统会读出所有的寄存器内容，而当一些寄存器内容读出后，另一些寄存器的内容就无法读取了（如读 USART 数据寄存器，导致 RXC 清零）特别需要注意事项一张列出所有 JTAGICE 不能访问的 IO 寄存器

一些 IO 寄存器需要在控制信号后特定的时钟周期内读写。在停止模式时，IO 时钟和外设时继续全速运行的。单步执行过这样的代码时经无法得到真确的时序。因为在单步操作过程中已经执行过了无数个 IO 时钟周期。要避免错误结果，应该尽量使用宏，函数，或执行到光标的操作。

支持的设备：

JTAGICE 支持全部带有 JTAG 接口的 AVR 处理器。

Mega 系列
ATmega16
ATmega162
ATmega169
ATmega32
ATmega323
ATmega64
ATmega128

使用入门

1. 打开包装

包装盒内有下列物品:

1. JTAGICE 主机和适配器
2. 9P RS232 通信电缆
3. 5VDC 稳压电源
4. 内有软件和数据手册随机光盘,

系统最低配置:

1. 486 处理器
2. 16M RAM
3. 16MB 的硬盘空间
4. 最高通信速率为 115200 的串行口
6. 5V 输出电流大于 60mA 的直流电源

2. 连接 JTAGICE

在启动 AVR Studio 之前, 必须把 JTAGICE 连接在上位 PC 和目标板之间。以确保 AVRStudio 可以完成自动侦测连接情况。

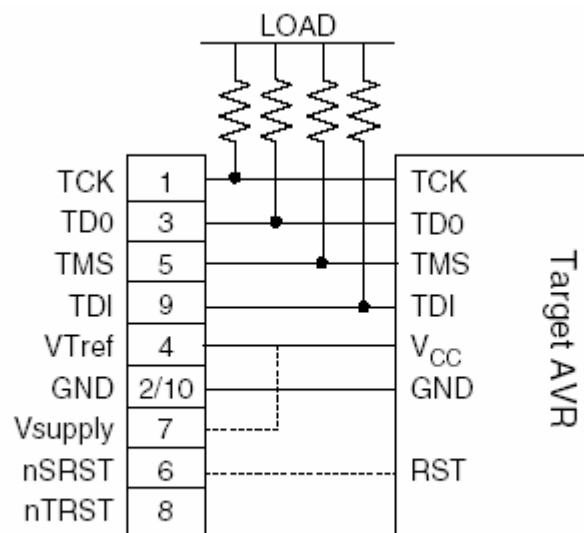
通过 PC 的连接: JTAGICE 采用标准的 RS232 端口通讯。用 RS232 通信电缆将空闲的 PC 串口同 JTAGICE 相连。启动 AVR Studio 就可以自动检测到仿真器。只得注意的时 AVR Studio 不能同其他程序共享串口资源, 在打开 AVR Studio 前要关闭其他使用该串口的程序 (如 ModeM, IrDA)。

JTAGICE 同目标板的连接: 最小需要 6 条线与目标板相连, 才可以完成仿真任务, 他们是: **TCK、TMS、TDO、TDI、Vref、GND**

另外有两条可选的引线 nSRST 和 Vsupply。引脚 nTRST 不接,

Vsupply 的功能是由目标板向 JTAGICE 供电 (仅对接口部分)

nSRST 的作用是监视目标板的复位线。然而, 在仿真过程中不是必须的。如果应用程序对 MCUSR 中 JTD 位进行了编程, JTAG 接口就会关闭, 为了使用 JTAGICE 对目标板重新编程。就必须控制复位引脚。下面给出了一个用 6 条线与目标板相连的例子:



在上面的例子中。为了避免输出电平配合不当，四条信号线上设计了上拉电阻。注意上拉电阻的取值不能使拉灌电流超出器件的允许值。

注意：JTAGICE 不支持在 JTAG 接口上连有多个器件

电源连接：

JTAGICE 使用外部电源适配器供电。电源电压：**DC5V/60mA**

当 JTAGICE 同目标板接好后就可以打开电源：按照下面的次序进行上电操作，才能保证 JTAGICE 同 PC 和目标板间的通信正常

1. 打开目标板电源
2. 打开仿真器电源
3. 启动 AVRStudio

不同的上电顺序不会造成设备损坏，但可能引起 AVRStudio 与 JTAGICE 间的通信不正常。

上位软件

下面包含了所有在 AVRStudio 上使用 OCD 系统时的特殊性能和注意事项

安装 AVR Studio: 在所有 AVR 开发过程中，AVR Studio 是一个理想的 IDE 软件，它含有编辑器，汇编器，调试器和包括 JTAGICE 在内的所有 ATMEL 仿真器的上位软件。

在安装 AVRStudio 时将随机提供的光盘放入 CD-ROM 中，选取 avrstudio4.07 目录中的 setup。Exe 文件，按照安装向导提示的步骤进行即可。

使用 JTAGICE 进行片内调试: 假定读者已经具有如何使用 AVRStudio 的基本知识，下面主要将如何通过 AVRStudio 操作 JTAGICE

1. 按照上面所说的方法连接系统
2. 在确认连接无误打开电源后，就可以启动 AVRStudio 了，软件通过串口搜索 JTAGICE，值得注意的是，软件的对于端口的搜索是按顺序进行的为了保证能够找到 JTAGICE，应该拔出其他串口上的 ATMEL 设备。

JTAGICE 可以检测目标板上的电压，当检测到目标板加电后，JTAGICE 就会读出目标板上 AVR 器件的 JTAGID，如果 JTAGICE 不能读出目标板的 JTAGID

可能是下列原因：

1. JTAGICE 没有连到目标板 AVR 器件上, (检查连接情况)
2. 目标板 AVR 器件的 JTAG 熔丝没有激活 (使用其他 ISP 工具激活 JTAG 熔丝位)
3. 应用程序禁止 MCUSR 中 JTD 位。(如果 reset 接入了 JTAGICE, 可用 JTAGICE 重新编程,)

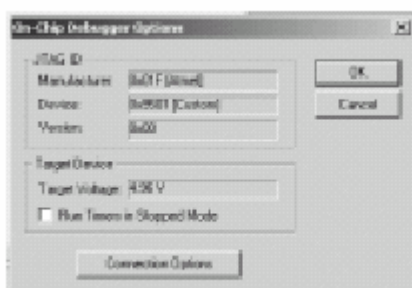
在读到正确的 JTAGID 后, 根据读到的器件 JTAGID AVRStudio 会自动配置 IO 映象和相关的设定。JTAGICE 会读取目标 CPU 的锁定位, 如果锁定位已经编程, JTAGICE 在下一步操作时会擦除芯片。

JTAG ICE 能自动使能目标 CPU 的 OCD 位, 退出时可以自动禁止该项功能。

为确保 JTAG 和目标 CPU 的通信, JTAG 的时钟频率应该小于等于目标晶振频率的 1/4。在芯片擦除前还有一个保持 EEPROM 内容的复选项, 一旦擦除完成 JTAGICE 就会释放 RESET 线, 随后数据就下载到 flash 中, 为了提高速度, 这里没有进行数据检验。

在线调试器 (OCD) 的配置:

在调试过程中进入 OCD 选项 通过选取 Options → On-chip Debugger Options. 如下图所示:



run timer mode: 设定定时器是否在单步操作时正常运行

OCD 通讯频率: 选择不超过 1/4 目标 CPU 频率的通讯速率是非常重要的, 太快会导致通讯失败, 太慢则会增加不必要的延时。

硬件断点: JTAGICE 具有一个与众不同的断点系统, 它有 3 个有效的断点寄存器, 硬件断点数目是有限的。另外在程序流还可以激活一种特殊的断点

3 个硬件断点寄存器可以用作:

1. 13 个普通断点
2. 2 个普通断点和一个数据断点
3. 一个普通断点和 2 个数据断点
4. 一个普通断点和一个 Mask 的 SRAM 断点
5. 一个普通断点和一个 Mask 的 flash 断点

当设定第四断点是, AVRStudio 会给出警告

软件断点: 一些 AVR CPU 支持 break 指令, 使用 break 指令意味着他将替代 flash 中的实际指令, 所以对 flash 重新编程。软件断点速度要比硬件的速度慢一些。

AVRStudio 提供三种断点设定方式:

1 自动, 但硬件断点优先: AVRStudio 会分析断点设定情况, 根据规矩决定是软件断点还是 **硬件断点**

1 只设定硬件断点: 只能设三个断点, (默认方式)

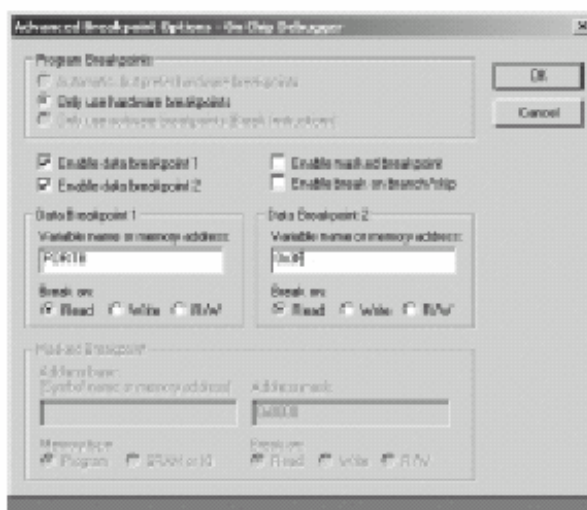
3 只设定软件断点:

普通断点: 普通断点可以放在程序代码的任何地方, 无论是汇编还是高级语言, AVR 运行过程中遇到断点就会停下来。

数据存储断点: 这类断点有三种设定方式，即制定 SRAM 单元读，写，或读写。数据存储断点可以设定在 IO 空间和 SRAM 空间，不能用作寄存器文件中。

为了使用固定的变量名，要在目标文件中放入变量信息。c 语言编译器和一些汇编器可以做到这一点。随 AVRStudio 一起提供的编译器不支持符号变量

在设定断点后，一旦满足预设的触发条件，程序就会停止。下面是设定数据存储断点对话框

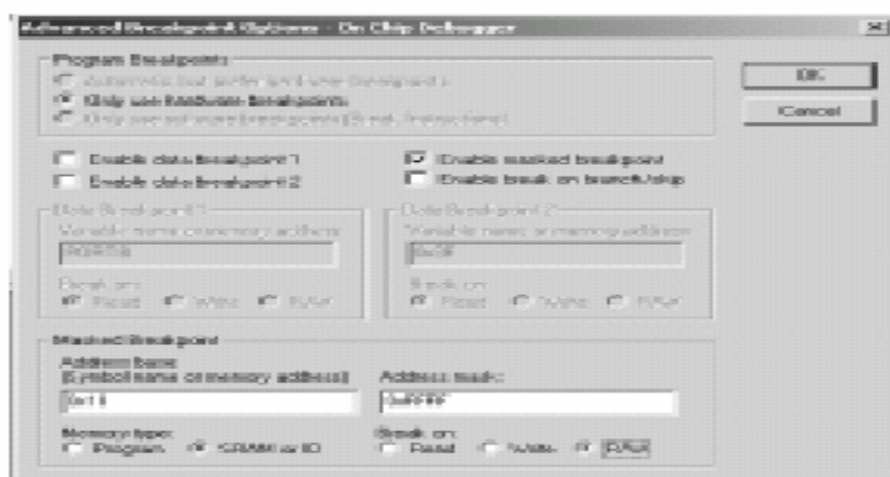


mask 类型断点: 断点通过一个基地址寄存器和一个屏蔽寄存器来设定。两个寄存器位与后的结果同程序地址寄存器或数据地址寄存器比较，判断是否满足断点触发条件。

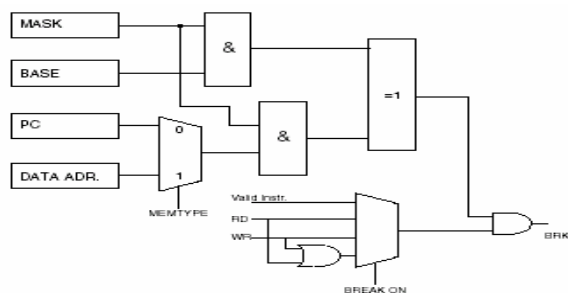
如果屏蔽寄存器的某位=0，则此位任何时候都满足触发条件，无论程序地址寄存器或数据地址寄存器相对应位的内容如何。要是屏蔽寄存器的某位=1，

相应的程序地址寄存器或数据地址寄存器数据位的值，必须同基地址寄存器相应位的值相等，才满足触发条件。下面是 Mask 断点的原理框图和举例表

设定 mask 型断点的对话框



原理框图



例子

#	Address Base	Address Mask	Break Vectors	# Break Vectors
1	1010 1010 0101 0101	1111 1111 1111 1111	1010 1010 0101 0101	$2^0 = 1$
2	1010 1010 0101 0101	1111 1101 1111 1111	1010 10x0 0101 0101	$2^1 = 2$
3	1010 1010 0101 0101	1111 0000 1111 1111	1010 xxxx 0101 0101	$2^4 = 16$
4	1010 1010 0101 0101	1010 1010 0101 0101	1x1x 1x1x x0x0 x0x0	$2^8 = 256$
5	1010 1010 0101 0101	0000 0000 0000 0000	xxxx xxxx xxxx xxxx	$2^{16} = 65536$

关于处理器窗口：如定时器，消耗时间，跑表的在使用 JTAGICE 是无效的，这是 OCD 系统本身的原因。

使用 JTAGICE 进行编程操作：JTAGICE 不但可以片内调试，还能为 MCU 烧录程序，JTAGICE 的烧录软件是对 STK 系列烧录程序进行了扩展。编程时一定注意不要禁止 JTAG Enable 熔丝位。

一些特别许要注意的事项

1: 外设的问题：1: 注意在 OCD 断点处于停止模式时，只是 CPU 挂起，大部分设备接续全速运行。但有两个例外的设备

定时计数器和看门狗，定时计数器可以通过菜单设定在停止模式时运行还是停止。看门狗会在进入停止模式时挂起。

2: 单步问题：有由于停止模式时，IO 的运行使我们应该充分注意其产生的时序误差。

例如：OUT PORTB, 0XAA

IN TEMP, PINB

在正常运行过程中不能等到正常结果，要在 OUT 和 IN 之间加一个 NOP 延时。而在使用 JTAGICE 单步时，TEMP 始终等于 0XAA，正是因为单步的延迟锁存器有足够的时间完成所要的 IO 操作

3. 软件断点：

有的 AVR CPU 有断点指令，可以通过 OCD 设置软件断点。但要考虑到软件断点是向 flash 中插入指令，如果频繁设置或移动断点，不但缩短 flash 寿命，还会耗费 OCD 的时间。

4. 目标 cpu 复位或掉电：

JTAGICE 在调试过程中掉电，适配器就会复位和失去通信。假如目标 CPU 掉电或进入 POWER DOWN 模式，在其重新执行命令前，AVRStudio 不会显示新的状态。

5. JTAG 相关的熔丝位设定：

两个相关的熔丝位必须激活（AVRStudio 可以自动设定 OCD 位），才能通过 JTAG 进行仿真调试或烧录程序。两个熔丝位在出厂时设为激活状态。如果被禁止就要通过其他的

ISP 工具激活两个熔丝位才能使用 JTAG 接口。

6. JTAGICE 只允许一个设备在 JTAG 链上

7. 访问 IO 寄存器：

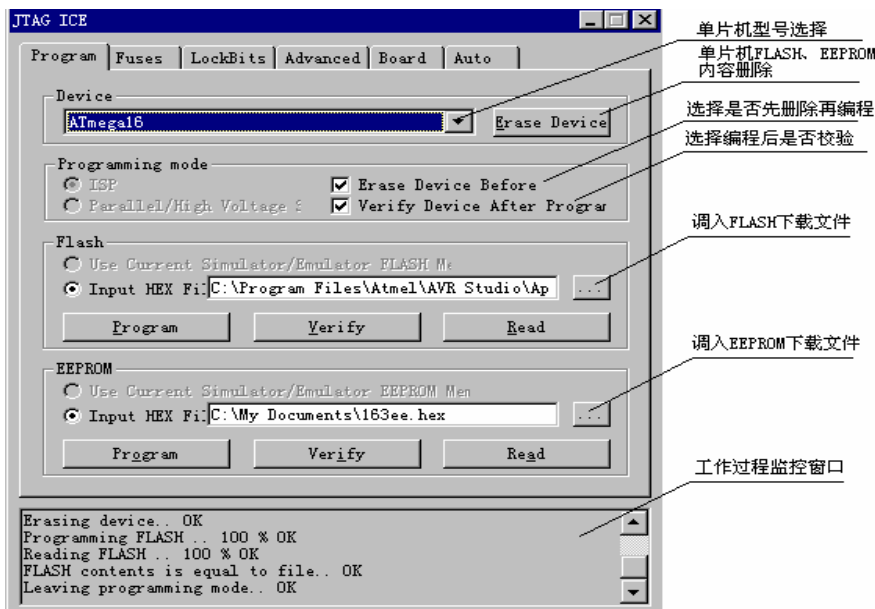
在 AVRStudio 中改变 IO 寄存器的内容在单步或全速运行前不会写到实际的寄存器中。要留意在调试某些寄存器时，对 ADC SPI USART 的影响。

8 如果 JTAG 界面处于激活状态，相应的引脚不能用于其他功能。

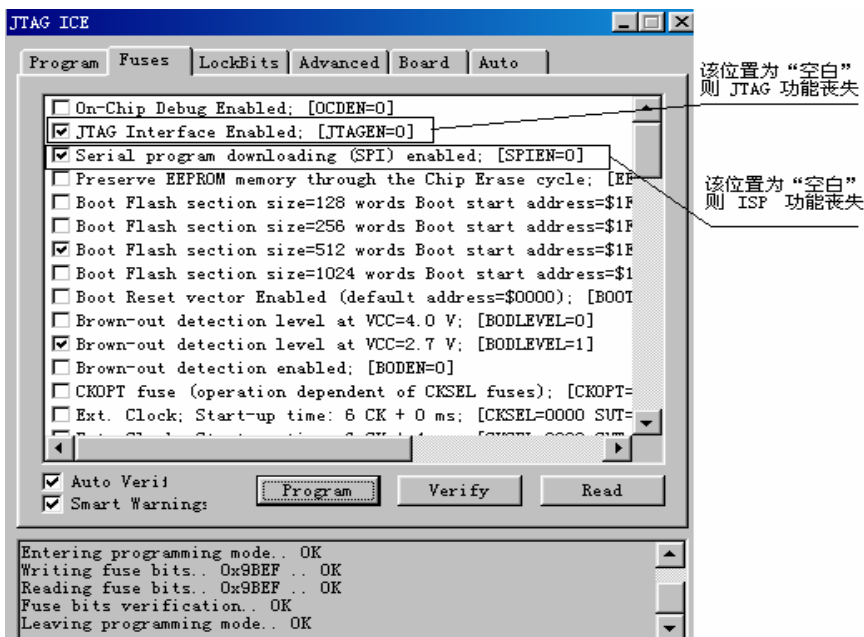
编程功能的使用

按前面所讲，将 JTAGICE 与 PC、电源、用户 CPU 连接好，启动 AVRstudio； 在 avrstudio 主窗口内的“TOOLS”栏目中选择“stk500/avr isp/jtag ice”栏目，如联机成功显示 JTAG ICE 窗口，如联机失败显示 STK500 窗口。

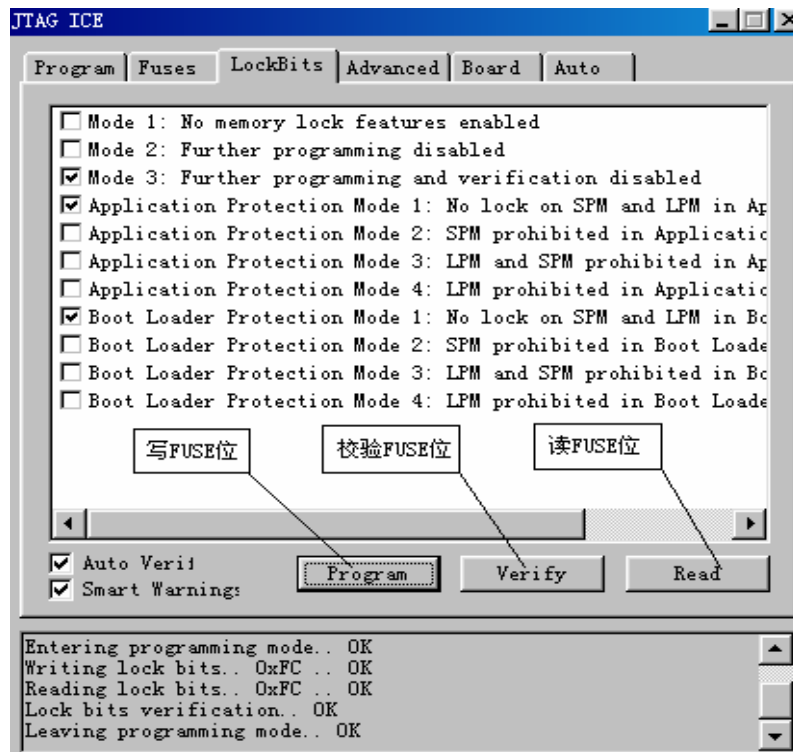
“PROGRAM”窗口介绍。



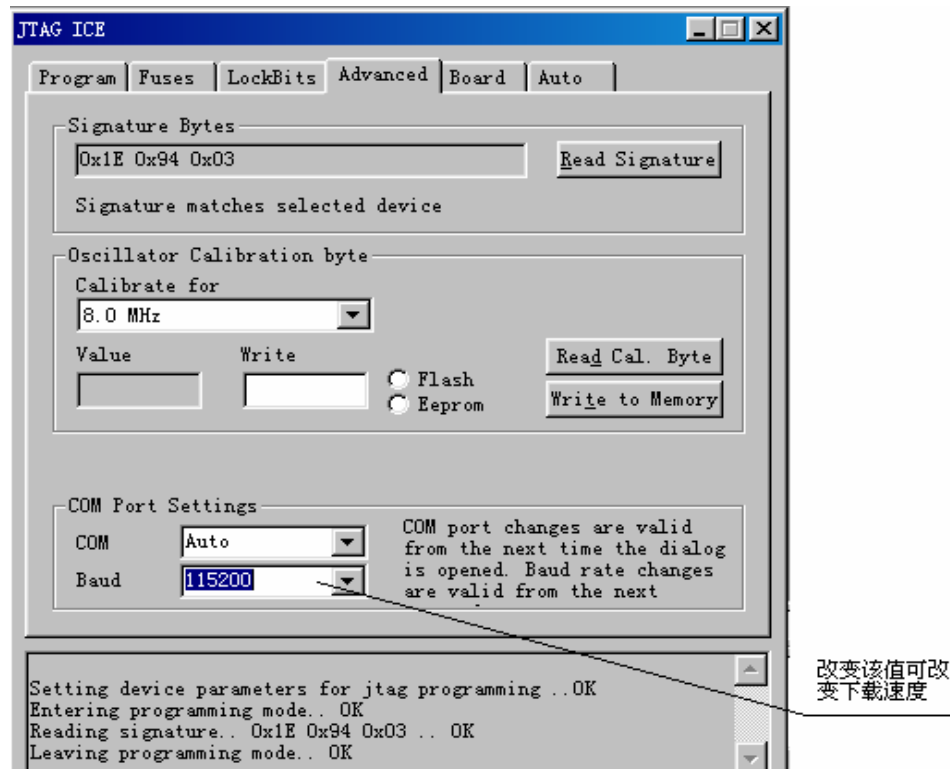
“FUSES”窗口介绍。



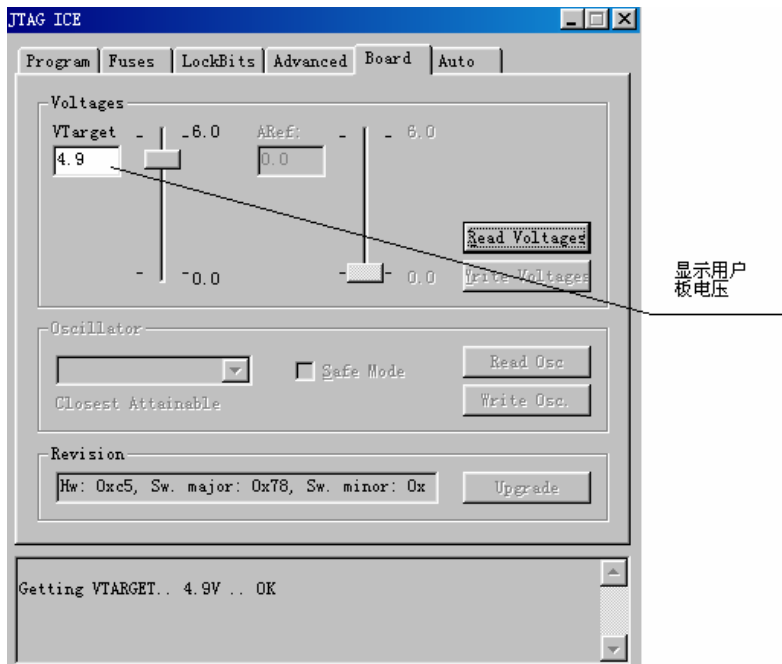
“LOCKBITS” 窗口介绍。



“ADVANCED” 窗口介绍。



“BOARD” 窗。



“AUTO” 窗口介绍。

