



ATmega16 单片机学习准备

1.1 认识 ATmega16 单片机

ATmega16 是基于增强的 AVR RISC 结构的低功耗 8 位 CMOS 微控制器。

ATmega16 有如下特点：16KB 的系统内可编程 Flash（具有同时读写的能力，即 RWW），512B 的 EEPROM，1KB 的 SRAM，32 个通用 I/O 口线，32 个通用工作寄存器，用于边界扫描的 JTAG 接口，支持片内调试与编程，3 个具有比较模式的灵活的定时器/计数器（T/C），片内/外中断，可编程串行 USART，有起始条件检测器的通用串行接口，8 路 10 位具有可选差分输入级可编程增益（TQFP 封装）的 ADC，具有片内振荡器的可编程看门狗定时器，一个 SPI 串行端口，以及 6 个可以通过软件进行选择的省电模式。

工作于空闲模式时 CPU 停止工作，而 USART、两线接口、ADC、SRAM、T/C、SPI 端口以及中断系统继续工作；停电模式时晶体振荡器停止振荡，所有功能除了中断和硬件复位之外都停止工作；在省电模式下，异步定时器继续运行，允许用户保持一个时间基准，而其余功能模块处于休眠状态；ADC 噪声抑制模式时终止 CPU 和除了异步定时器与 ADC 以外所有 I/O 模块的工作，以降低 ADC 转换时的开关噪声；Standby 模式下只有晶体或谐振振荡器运行，其余功能模块均处于休眠状态，使得器件只消耗极少的电流，同时具有快速启动能力；扩展 Standby 模式下则允许振荡器和异步定时器继续工作。

ATmega16 是以 Atmel 公司高密度非易失性存储器技术生产的。片内 ISP Flash 允许程序存储器通过 ISP 串行接口或者通用编程器进行编程，也可以通过运行于 AVR 内核之中的引导程序进行编程。引导程序可以使用任意接口将应用程序下载到应用 Flash 存储区（Application Flash Memory）。在更新应用 Flash 存储区时引导 Flash 区（Boot Flash Memory）的程序继续运行，实现了 RWW 操作。通过将 8 位 RISC CPU 与系统内可编程的 Flash 集成在一个芯片内，ATmega16 成为一个功能强大的单片机，为许多嵌入式控制应用提供了灵活而低成本的解决方案。

ATmega16 具有一整套的编程与系统开发工具，包括：C 语言编译器、宏汇编、程序调试器、软件仿真器、仿真器及评估板。

ATmega16 单片机常用的 TQFP 44 脚封装形式如图 1-1 所示，DIP 40 脚封装形式如图 1-2 所示。



图 1-1 TQFP 44 脚封装

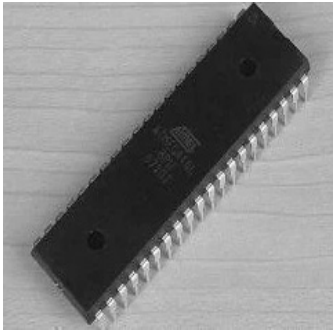


图 1-2 DIP 40 脚封装

1.1.1 ATmega16 芯片及引脚认识

如图 1-3 所示，ATmega16 的引脚排列主要由端口 A、端口 B、端口 C、端口 D 及复用功能口；复位引脚 $\overline{\text{RESET}}$ ；电源脚 VCC、GND；以及 A/D 转换功能脚 AREF、AVCC 等组成。下面分别介绍。

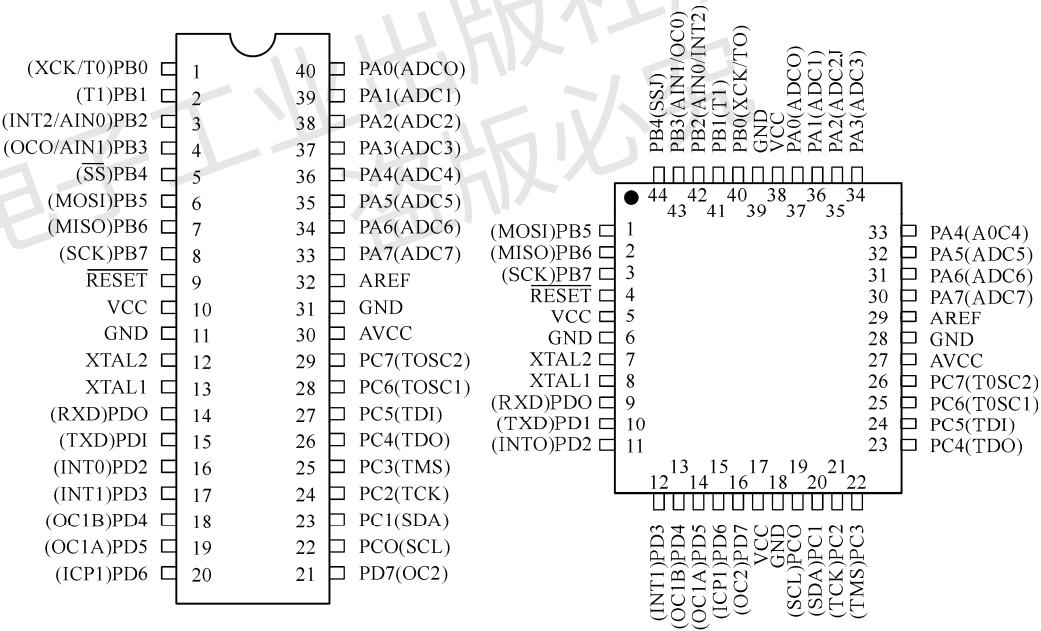


图 1-3 PDIP 及 QFC 两种封装的 ATmega16 引脚

VCC：电源正。

GND：电源地。

端口 A (PA7..PA0)：端口 A 作为 ADC 的模拟输入端。端口 A 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 A 仍处于高阻状态。

端口 B (PB7..PB0)：端口 B 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 B 仍处于高阻状态。端口 B 也可以用作其他不同的特殊功能。

端口 C (PC7..PC0)：端口 C 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 C 仍处于高阻状态。如果 JTAG 接口使能，即使复位也会出现引脚 PC5 (TDI)、PC3 (TMS) 与 PC2 (TCK) 的上拉电阻被激活。端口 C 也可以用作其他不同的特殊功能。

端口 D (PD7..PD0)：端口 D 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。作为输入使用时，若内部上拉电阻使能，则端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 D 仍处于高阻状态。端口 D 也可以用作其他不同的特殊功能。

RESET：复位输入引脚。持续时间超过最小门限时间的低电平将引起系统复位。门限时间持续时间小于门限时间的脉冲不能保证可靠复位。与传统的 51 单片机相比，AVR 单片机内置复位电路，并且在熔丝位里可以控制复位时间，所以，AVR 单片机可以不设外部上电复位电路，依然可以正常复位，稳定工作。需要注意的是本单片机低电平有效。

XTAL1：反向振荡放大器与片内时钟操作电路的输入端。

XTAL2：反向振荡放大器的输出端。

AVCC：AVCC 是端口 A 与 A/D 转换器的电源。不使用 ADC 时，该引脚应直接与 VCC 连接；使用 ADC 时应通过一个低通滤波器与 VCC 连接。

AREF：ADC 的模拟基准输入引脚。



1.1.2 复位电路的设计

如图 1-4 所示，ATmega16 已经内置了上电复位设计，并且在熔丝位里可以设置复位时的额外时间，故 AVR 外部的复位线路在上电时，可以设计得很简单：直接拉一只 10kΩ 的电阻到 VCC 即可 (Rrst)。

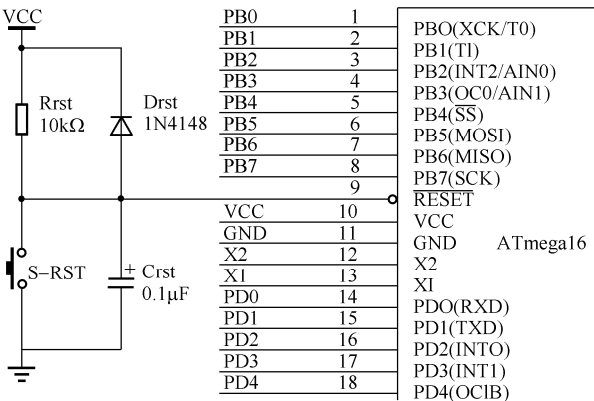


图 1-4 复位电路



为了可靠复位，再加上一只 0.1μF 的电容（Crst）以消除干扰、杂波。

Drst（1N4148）的作用有两个：一是将复位输入的最高电压钳位在 $V_{CC}+0.5V$ 左右；二是系统断电时，将 Rrst（10kΩ）电阻短路，让 Crst 快速放电，从而当下一次通电时，能产生有效的复位。

在 AVR 单片机工作期间，按下 S-RST（复位按钮）开关再松开时，将在复位脚产生一个低电平的复位脉冲信号，触发 AVR 单片机复位。



重要说明：

实际应用时，如果你不需要复位按钮，复位脚可以不接任何的零件，AVR 芯片也能稳定工作。即这部分不需要任何的外围零件。



1.1.3 晶振电路的设计

如图 1-5 所示，ATmega16 已经内置 RC 振荡线路，可以产生 1MHz、2MHz、4MHz、8MHz 的振荡频率。不过，内置的毕竟是 RC 振荡，在一些对时间参数要求较高的场合，比如要使用 AVR 单片机的 UART 与其他的单片机系统或 PC 通信时，为了实现高速可靠的通信，就需要比较精确的时钟来产生精确的通信波特率，这时就要使用精度高的片外晶体振荡电路作为 AVR 单片机系统的工作时钟。

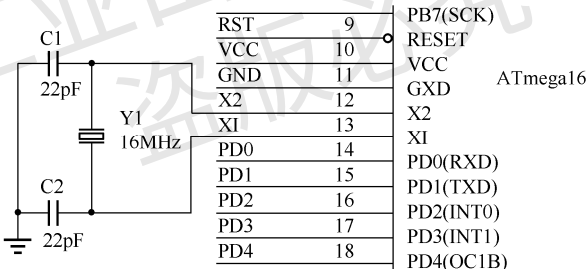


图 1-5 晶振电路

在早期的 AVR 单片机中，比如 AT90S 系列单片机，晶振两端均需要接 22pF 左右的电容。ATmega 系列单片机在实际使用时，这两只小电容不接也能正常工作。不过为了线路的规范化，我们仍建议接上。



重要说明：

实际应用时，如果你不需要太高精度的频率，可以使用内部 RC 振荡。即这部分不需要任何的外围零件。



1.1.4 A/D 转换滤波电路的设计

如图 1-6 所示，为减小 A/D 转换的电源干扰，ATmega16 芯片由独立的 A/D 电源供电。推荐在 VCC 串上一只 10μH 的电感（L1），然后接一只 0.1μF 的电容（C6）到模拟地。

ATmega16 内带 2.56V 标准参考电压。也可以从外面输入参考电压，比如在外面使用 TL431 基准电压源。不过一般的应用使用内部自带的参考电压已经足够。习惯上在 AREF 脚接一只 0.1μF 的电容（C5）到模拟地。

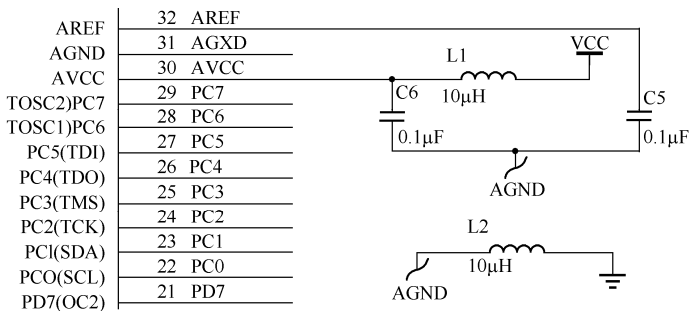


图 1-6 A/D 转换滤波电路

重要说明：

实际应用时，如果你想简化线路，可以将 AVCC 直接接到 VCC，AREF 悬空。即这部分不需要任何的外围零件。

1.1.5 串口电平转换电路的设计

如图 1-7 所示，串行通信协议有很多种，以 RS-232-C（又称 EIA RS-232-C，以下简称 RS232）的通信方式最为多见。

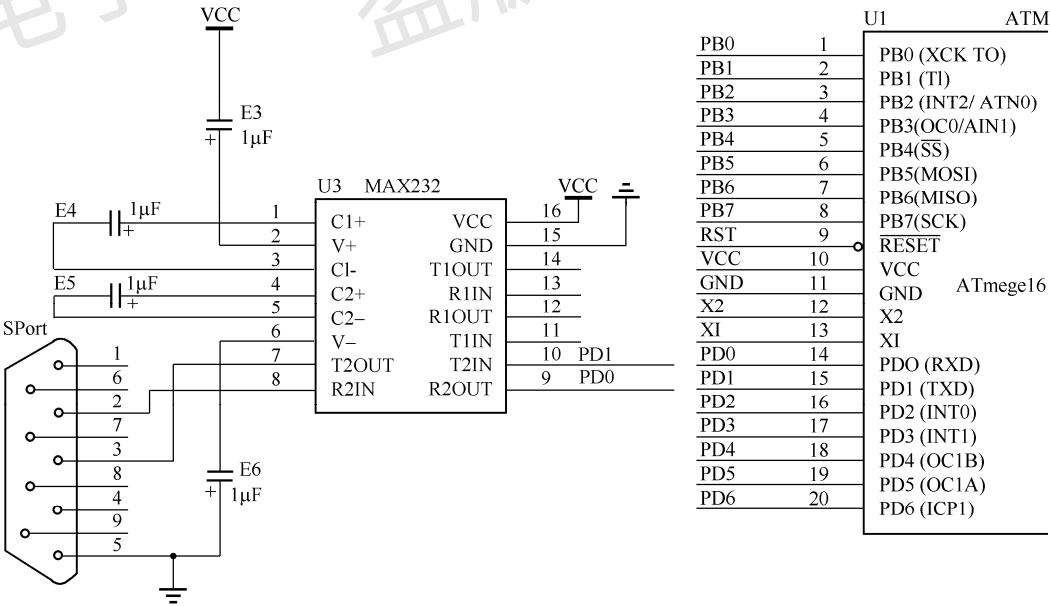


图 1-7 串口电平转换电路

RS232 是一个全双工的通信协议，它可以同时进行数据接收和发送的工作。RS232 的端口通常有两种：9 针（DB9）和 25 针（DB25）。其接口定义如表 1-1 所示。



表 1-1 DB9 与 DB25 接口定义

信 号	DB9	DB25
公共地	5	7
发送数据（TXD）	3	2
接收数据（RXD）	2	3
数据终端准备（DTR）	4	20
数据准备好（DSR）	6	6
请求发送（RTS）	7	4
清除发送（CTS）	8	5
数据载波检测（DCD）	1	8
振铃指示（RI）	9	22

RS232 接口的信号电平值较高，在 RS232 中任何一条信号线的电压均为负逻辑关系，即逻辑“1”， $-5\sim-15\text{V}$ ；逻辑“0”， $+5\sim+15\text{V}$ ；噪声容限为 2V。因为与 TTL 电平不兼容，故需使用电平转换电路方能与 TTL 电路连接。在本设计中，电平转换芯片是 MAXIM 公司的 MAX232 芯片，电路设计参考了其 DATASHEET 上的典型应用。



1.1.6 I/O 端口输出

ATmega16 总共有 PA、PB、PC、PD 四个 8 位 I/O 端口，作为最小系统板需要将这 4 个 I/O 口引出。其电原理图如图 1-8 所示（PB 端口与 PD 端口的接法请参考 PC 端口）。

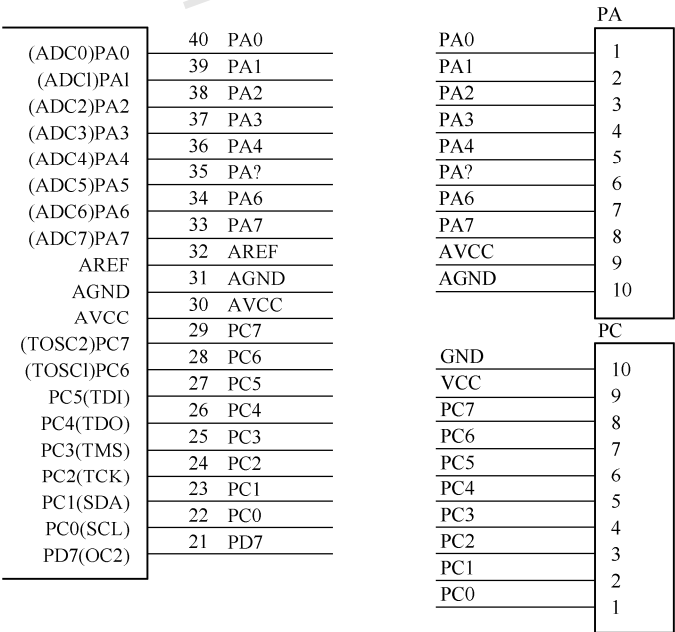


图 1-8 I/O 端口



1.1.7 JTAG 仿真接口电路的设计

JTAG 仿真接口电路如图 1-9 所示，图中：

- V-S 是一个三端跳线，用来选择电源的供给方式：由 JTAG 仿真接口供电还是由 ATmega16 最小系统板上的电源模块供电。
- 由于不同的 JTAG 仿真电路支持的 JTAG 协议不同，R4、R5、R6、R7 这 4 个上拉电阻并不是必要的。

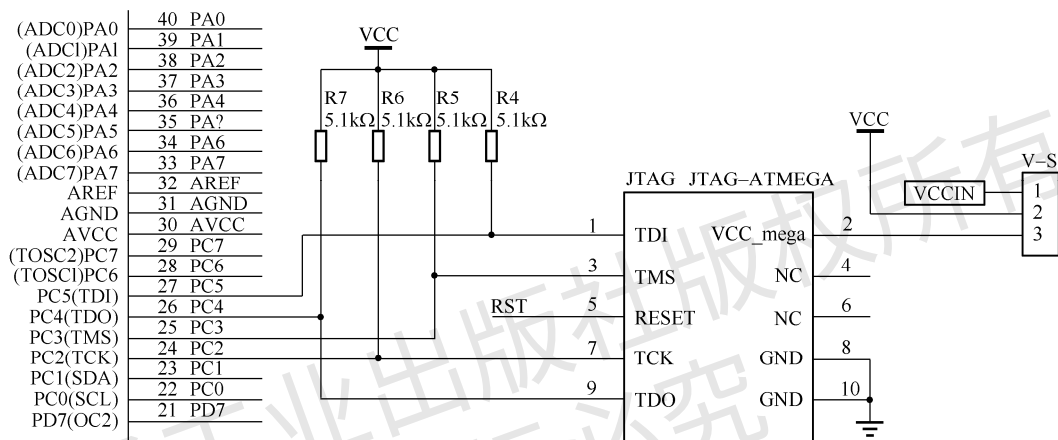


图 1-9 JTAG 仿真接口电路



1.1.8 电源电路的设计

采用 7805 集成电压三端器件能满足系统的要求，输入电压要求 9~18V，如图 1-10 所示。

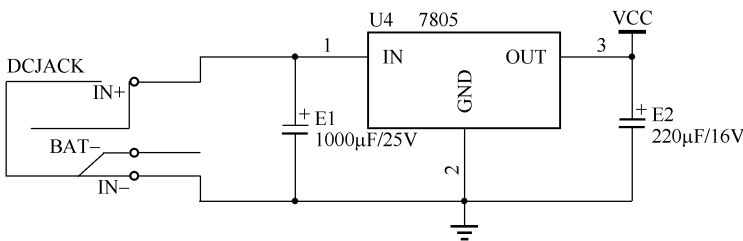


图 1-10 电源电路



1.1.9 ATmega16 最小硬件系统实物

根据各部分原理图最终完成的 ATmega16 最小硬件系统及框图设计如图 1-11 所

示。申请了实用新型专利单片机开发系统，仿真单片机与开发单片机在同一块 PCB 上，提高开发板与计算机通信成功率，也能降低系统成本；开发板上的 JTAG 接口复用功能可以用短路帽接通和断开的方式方便地切换使用开发板上的单片机和使用者自己开发的单片机。

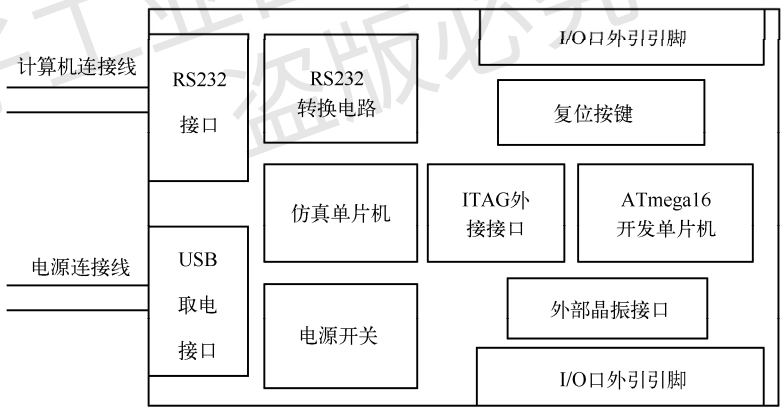
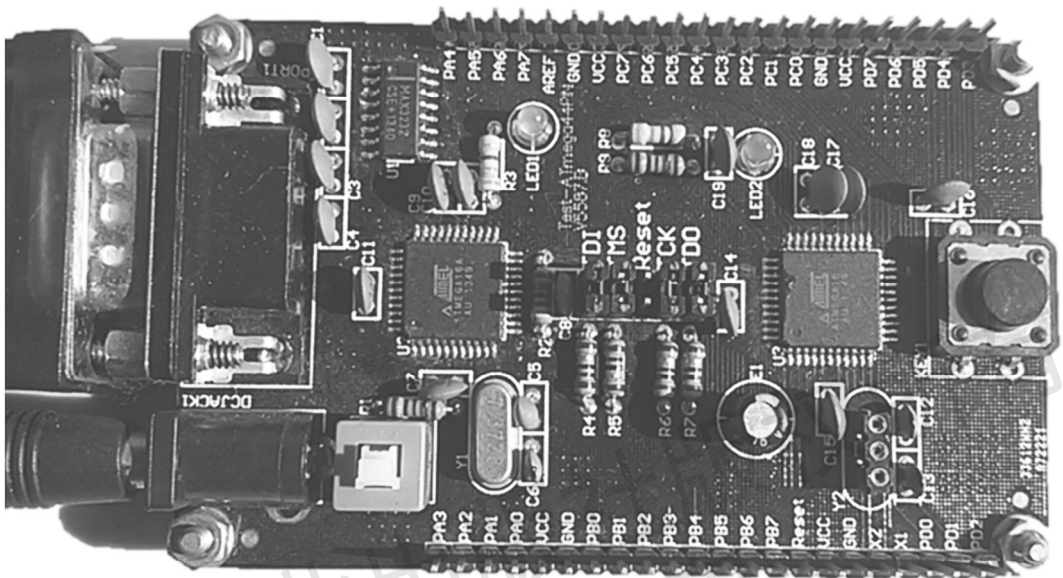


图 1-11 ATmega16 最小硬件系统及框图设计

1.2 ICC AVR 集成开发环境使用

- 打开 ICC 集成开发环境后，按照下面图示的提示，可以达到 ICC 快速入门的学习目的。
- (1) 选择 Project 菜单的 New 选项新建项目，如图 1-12 所示。
 - (2) 指定保存路径并输入工程名称，如图 1-13 所示。
 - (3) 选择 File 菜单的 New 选项新建文件，如图 1-14 所示。

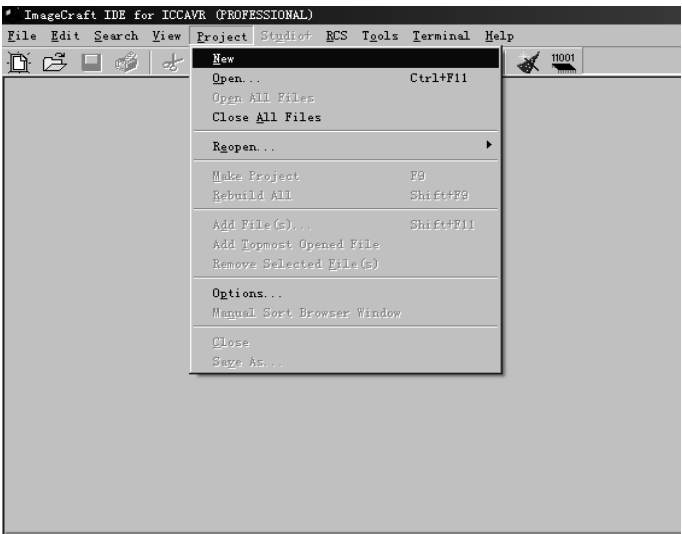


图 1-12 新建项目

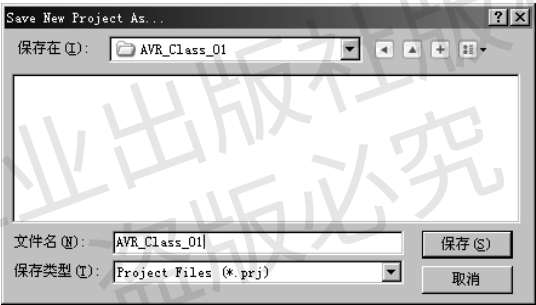


图 1-13 保存项目

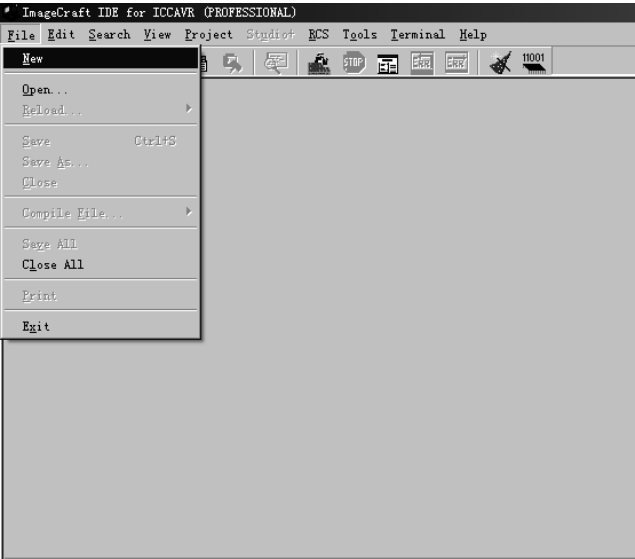


图 1-14 新建文件

(4) 输入程序代码，如图 1-15 所示。

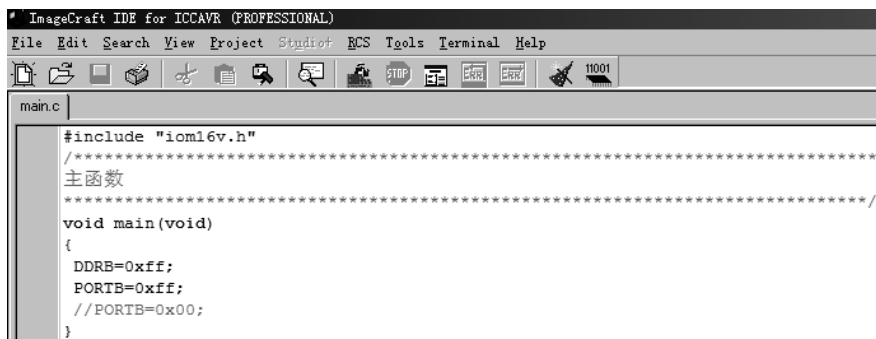


图 1-15 输入程序代码

要输入的程序代码如下：

```
#include "iom16v.h"
void main(void)
{
    DDRB=0xff;
    PORTB=0xff;
    //PORTB=0x00;
}
```

(5) 保存文件，如图 1-16 所示。



重要说明：

文件的后缀名是.c，建议与刚建立的工程保存在同一个文件路径内。

(6) 选中右边的 Files，右击，在弹出的快捷菜单中选择 Add File(s) 选项，在项目中添加文件，如图 1-17 所示。

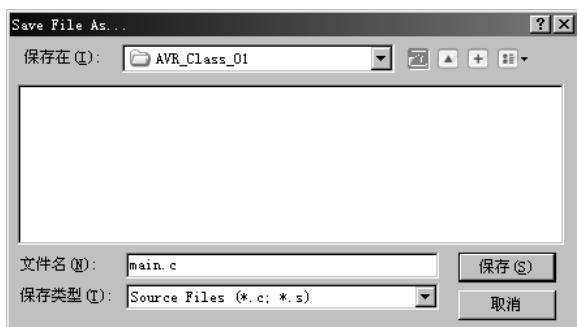


图 1-16 保存文件

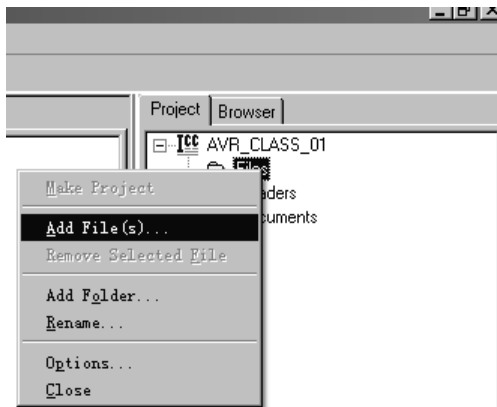


图 1-17 在项目中添加文件

(7) 选择 Project 菜单中的 Options 选项，在弹出的对话框中单击 Target 选项卡选择单片机类型，如图 1-18 所示。

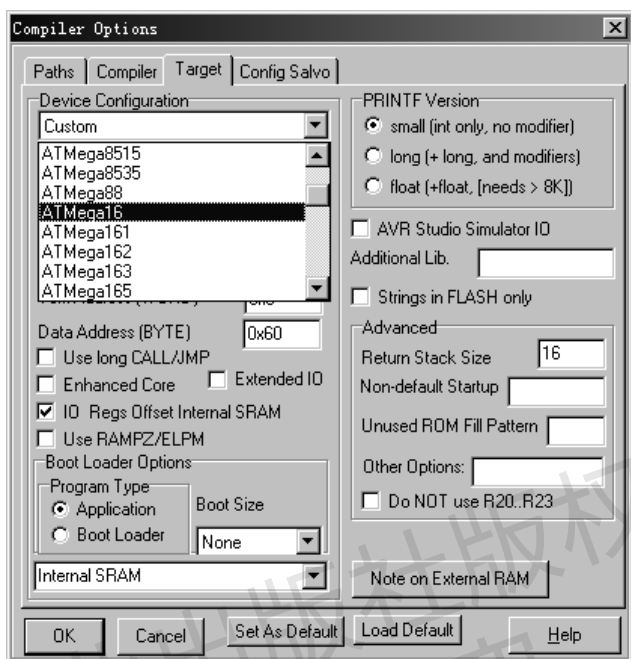


图 1-18 单片机类型选择

(8) 单击 Compiler 选项卡，设置输出文件格式为 COFF/HEX 类型，如图 1-19 所示。

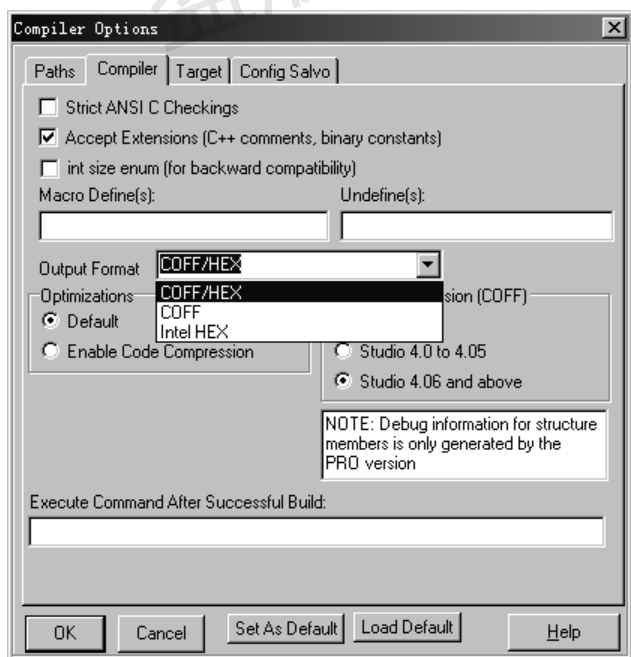


图 1-19 输出文件格式设置

(9) 选择 Project 菜单中的 Make Project 选项进行项目编译，如图 1-20 所示。

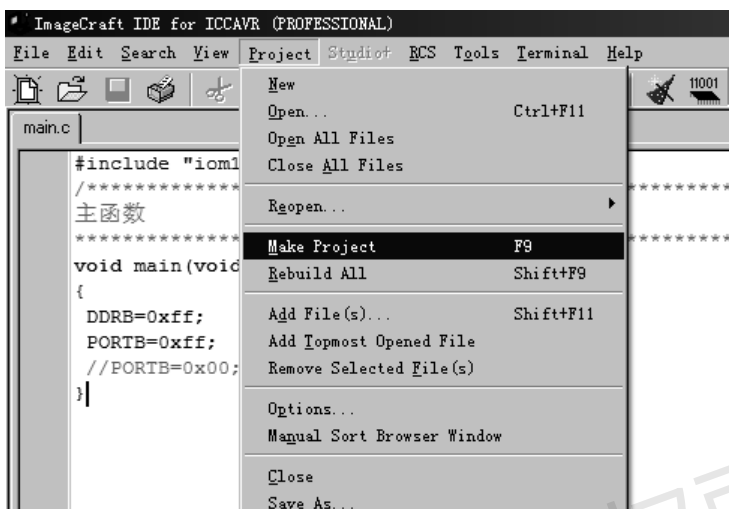


图 1-20 项目编译



重要说明：

此时若选择 Rebuild All 选项也可实现同样的功能。

(10) 编译结果如不正确，纠错后直到编译正确，如图 1-21 所示。

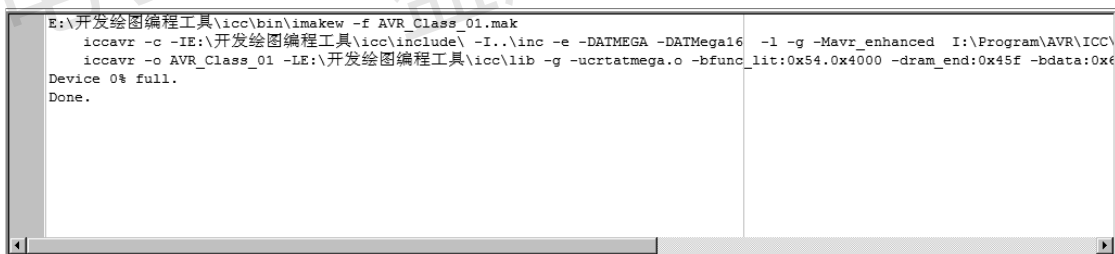


图 1-21 项目编译信息窗

1.3 AVR Studio 调试软件的使用

首先将 JTAG 下载仿真头连接好。

(1) 打开 AVR Studio 软件，在弹出的界面中单击 Cancel 按钮，如图 1-22 所示。

(2) 单击 Tools 菜单中的 Program AVR 选项，如图 1-23 所示。

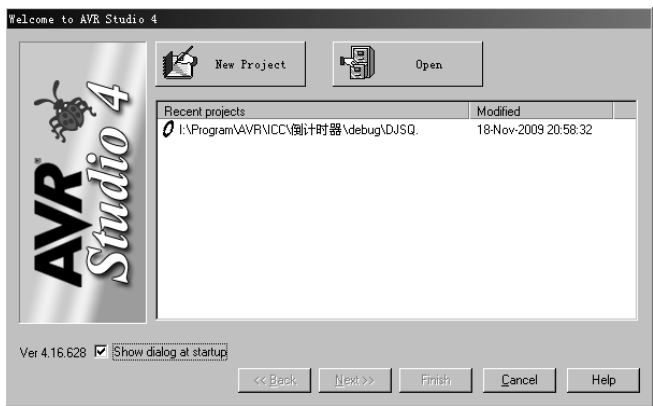


图 1-22 AVR Studio 初始弹出界面

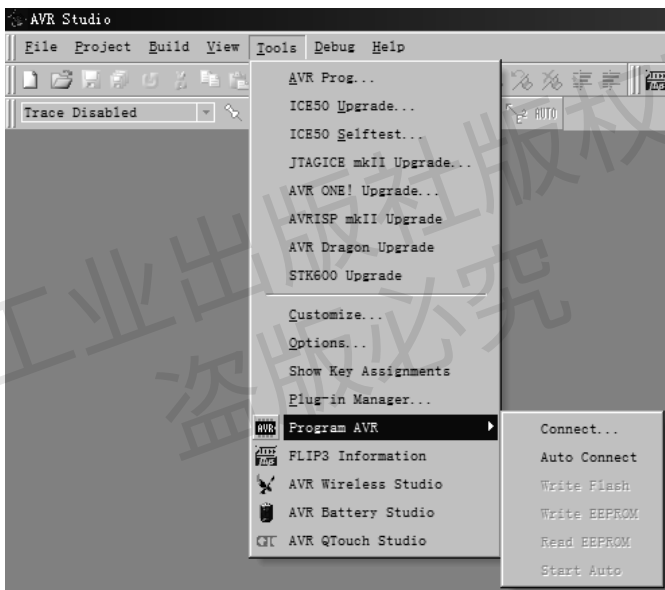


图 1-23 AVR Studio Tools 菜单

此时单击 Connect 选项，会出现如图 1-24 所示的 Select AVR Programmer 对话框。

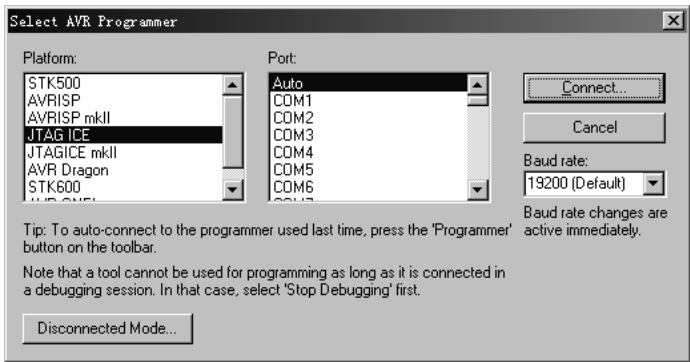


图 1-24 Select AVR Programmer 对话框

此时在左侧栏选择 JTAG ICE 后单击 Connect 按钮，就会进入熔丝位设置选项。

注意：

Connect 与 Auto Connect 的区别是：Connect 每次都会提示选择的设备名称与连接端口，Auto Connect 会自动使用上一次的设置，提高操作效率。

(3) 熔丝位设置，如图 1-25 所示。

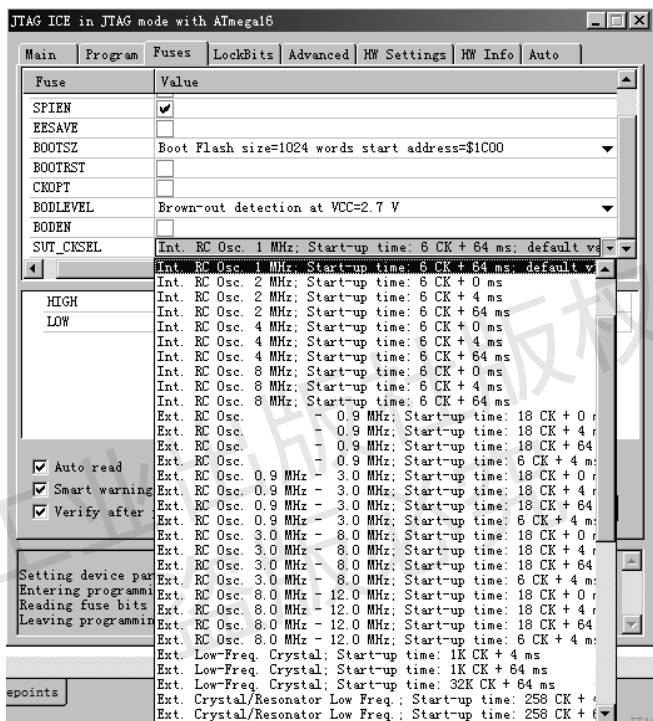


图 1-25 熔丝位设置

按需要进行相关设置，完成设置后单击 Program 选项进行编程，然后关掉该界面即可。

(4) 单击 File 菜单中的 Open File 选项，如图 1-26 所示。

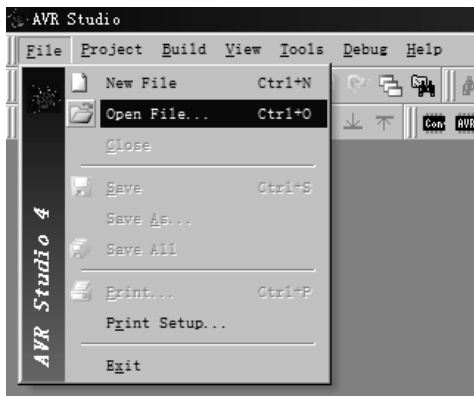


图 1-26 Open File 选项

(5) 弹出“打开”对话框，找到用 ICC AVR 编译生成的.cof 文件，按照提示进行操作，如图 1-27 所示。

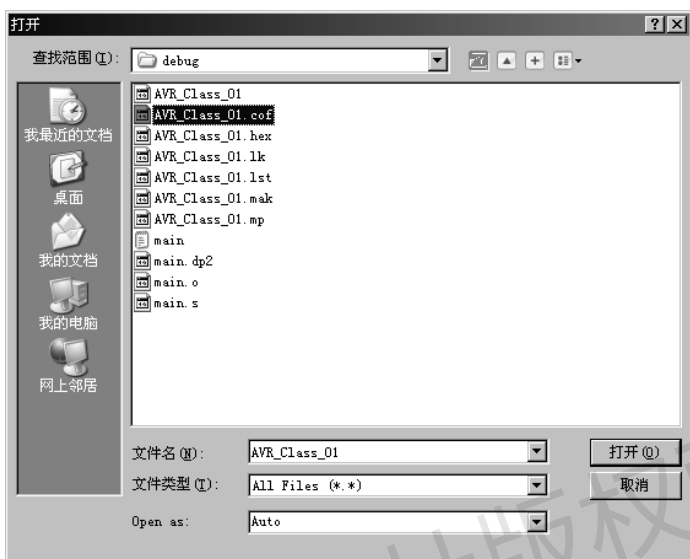


图 1-27 打开.cof 文件

(6) 选择调试接口 (JTAG ICE) 和器件 (ATmega16)，单击 Finish 按钮，如图 1-28 所示。

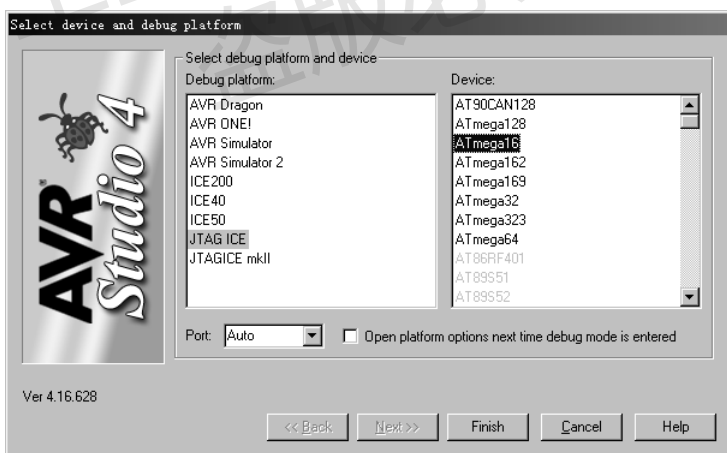


图 1-28 选择调试接口与器件

Port 端口选择 Auto 即可。如果没有连接 JTAG 设备，可以使用 Disconnected Mode (脱机模式) 进入查看操作界面。



重要说明:

根据接口选择 Platform，AVR Studio 本身不支持并口，同时，并不是所有的 AVR 单片机都支持 JTAG 协议。

(7) 调试界面如图 1-29 所示。

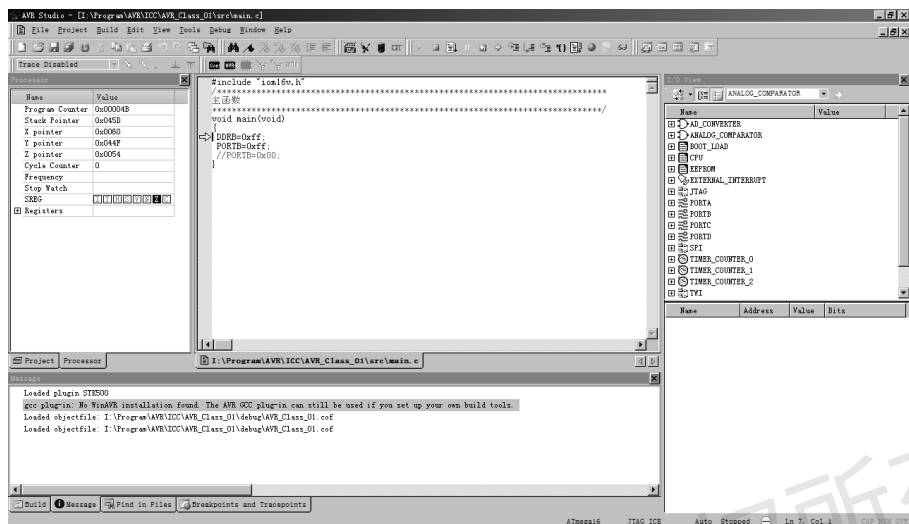


图 1-29 调试界面

1.4 PROTEUS 仿真软件的使用

打开 PROTEUS 软件, 按照如下所示步骤, 可以快速掌握单片机仿真软件的使用基本方法。

(1) 双机桌面或“开始”程序中的软件快捷启动图标启动 PROTEUS 软件, 如图 1-30 所示。

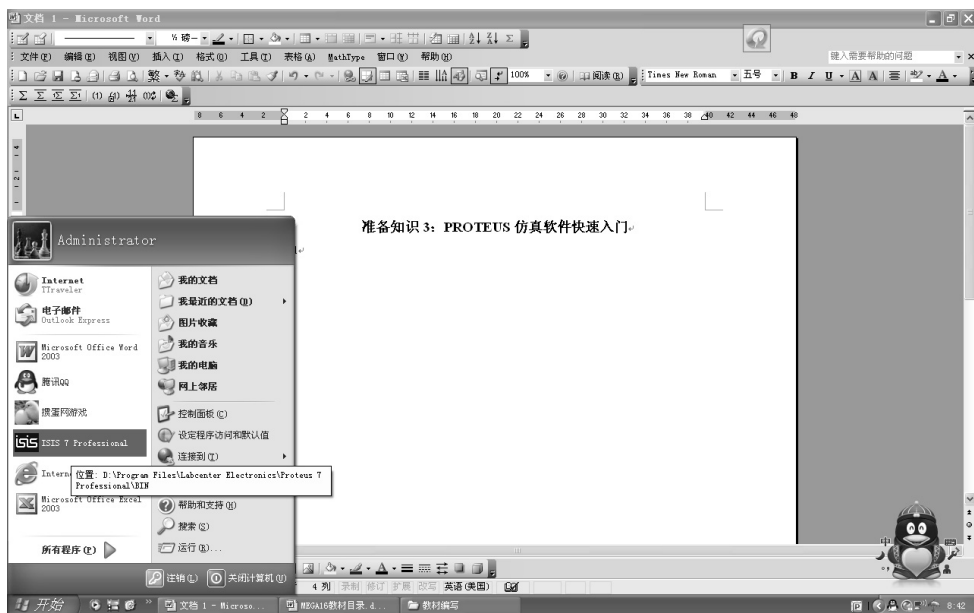


图 1-30 启动 PROTEUS 软件

(2) PROTEUS 工作界面如图 1-31 所示。

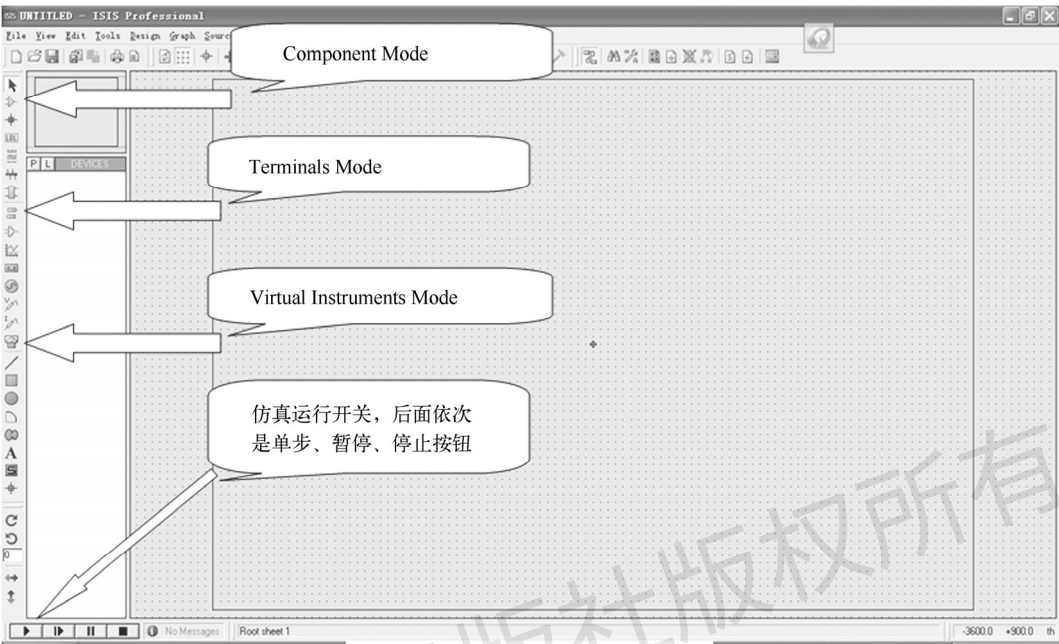


图 1-31 PROTEUS 工作界面

在 PROTEUS 工作界面中，最常用的有 3 种模式，分别为 Component Mode，能选择查找电路元件；Terminals Mode，能给你的电路加入电源或地；Virtual Instruments Mode，能选择示波器、信号发生器等。

(3) 电路元件选择。单击 Component Mode，弹出如图 1-32 所示元件选择对话框。

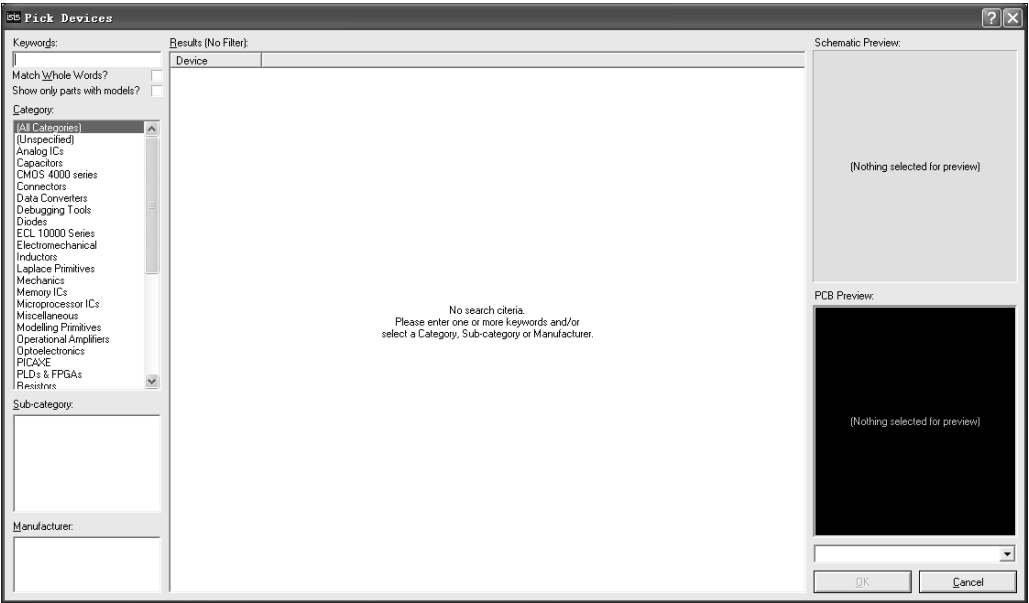


图 1-32 元件选择对话框

(4) 在 Keywords 文本输入框中输入元件名称, 如图 1-33 所示。

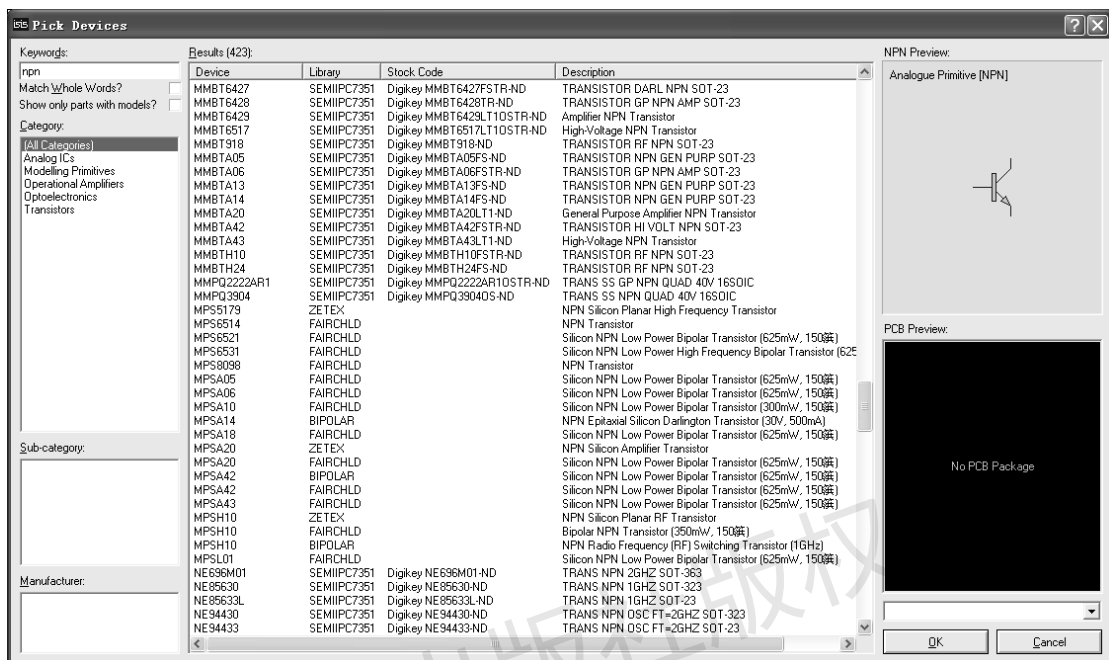


图 1-33 输入元件名称

(5) 电路元件布局、连线。在 PROTEUS 中设计电路, 如图 1-34 所示, 这是一个分立元件的射极跟随器电路。

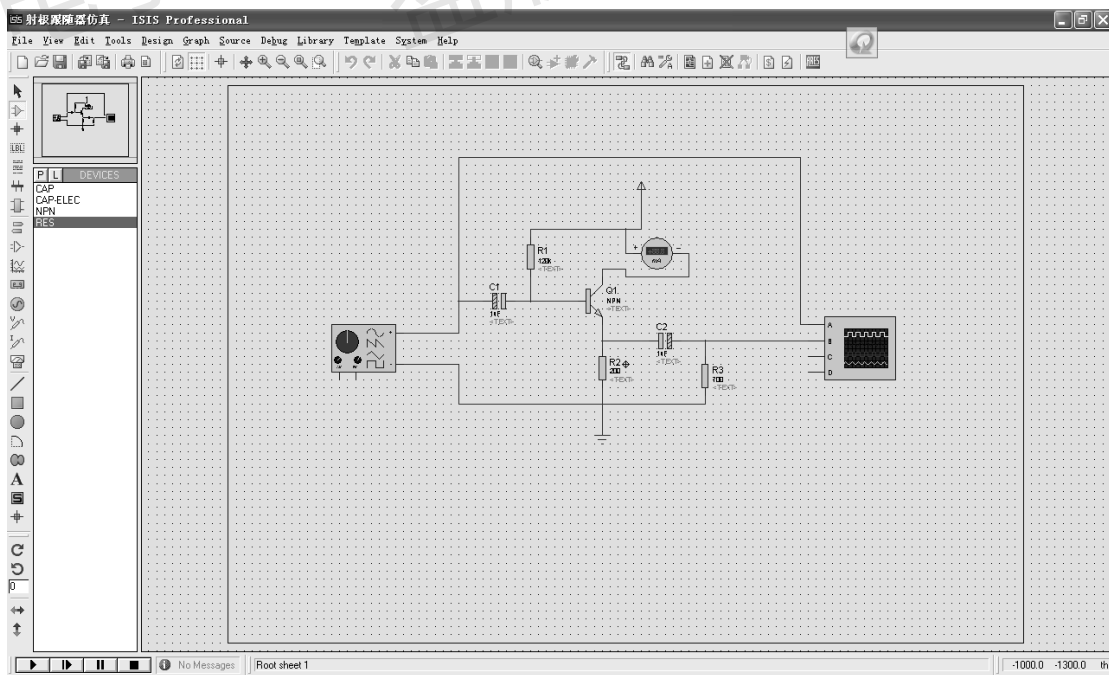


图 1-34 在 PROTEUS 中设计电路

(6) PROTEUS 仿真电路运行效果如图 1-35 所示。

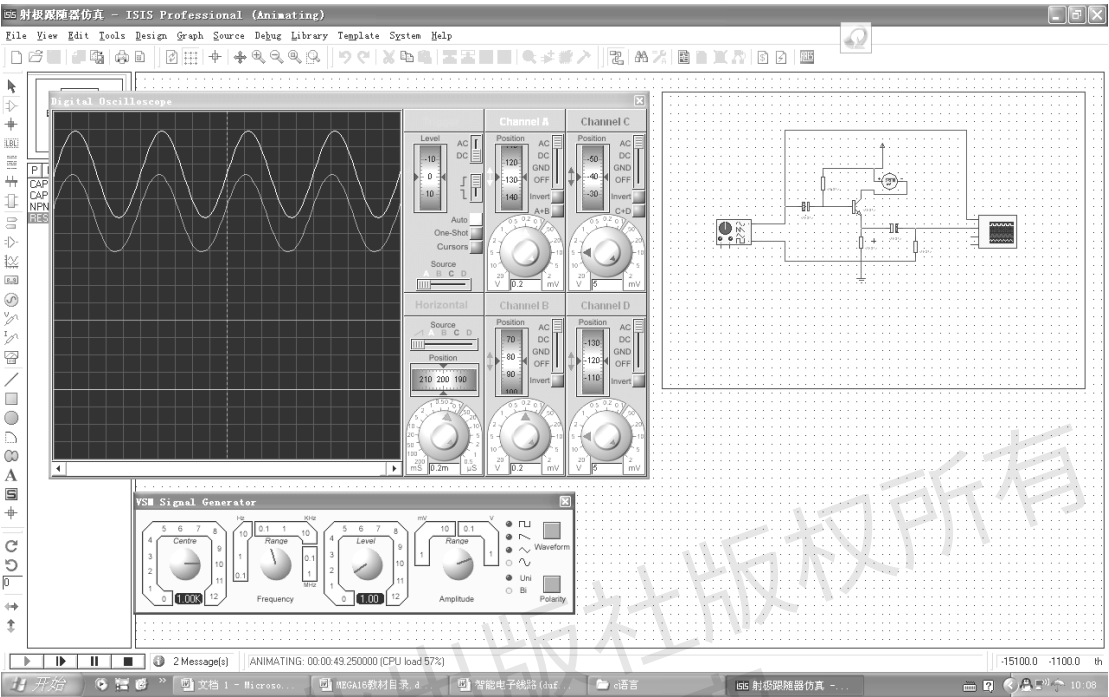


图 1-35 PROTEUS 仿真电路运行效果