

Usbtinyisp 使用说明

对使用 Arduino 控制器的 FANS 来说，拥有一款 arduino 控制芯片的 bootloader 下载线绝对是一件很棒的事，在 Arduino IDE 中，使用它可以为所有的 Arduino 控制用 AVR 单片机下载程序。这款 USBTINYISP 下载线附送 6 芯和 10 芯连线各一条，对于大部分 AVR 单片机程序下载也适用。

下面以 Arduino 2009 为例来介绍一下如何使用它来下载 bootloader。

1. 下载 Arduino IDE 软件 Arduino 0018 及 USBTINYISP 驱动

Arduino 0018 下载

<http://arduino.googlecode.com/files/arduino-0018.zip>

USBtinyISP 驱动下载

<http://blog.flemingoeda.com/wp-content/uploads/2009/05/usbtinyisp-w32-driver-v112.zip>

以上两个下载后解压缩到硬盘上。

2. USBTINYISP 的驱动安装：在 PC 上第一次使用 USBTINYISP 时，要首先安装其驱动，步骤如下：

用 USB 线把 USBTINYISP 板和电脑相连(记得 PWR 处跳线帽要插上，系统会提示找到相应的硬件





在随后出现的“找到新的硬件向导”对话框中，选“否，暂时不”，然后单击“下一步”按钮：



在随后出现的对话框中，选“从列表或指定位置安装（高级）”，然后单击“下一步”按钮：



接着指定之前解压缩开的 USBTINYISP 板驱动文件所在的目录：



单击“下一步”按钮之后，系统就开始安装 USBtinyISP 所需要的驱动文件：



最后单击“完成”按钮，驱动的安装就完成了

此时在系统的硬件管理器中，我们能够找到“USBtinyISP AVR Programmer”，表明该下载线已经可以使用了。

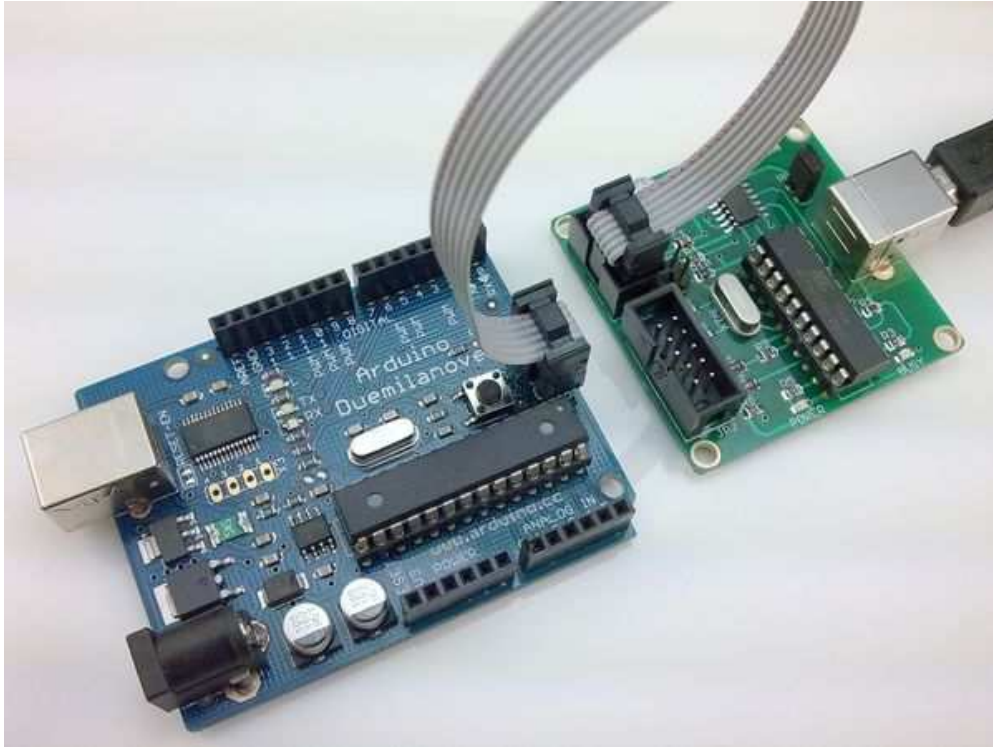


可以看到板上 PWR 处的绿色 LED 是亮起的。

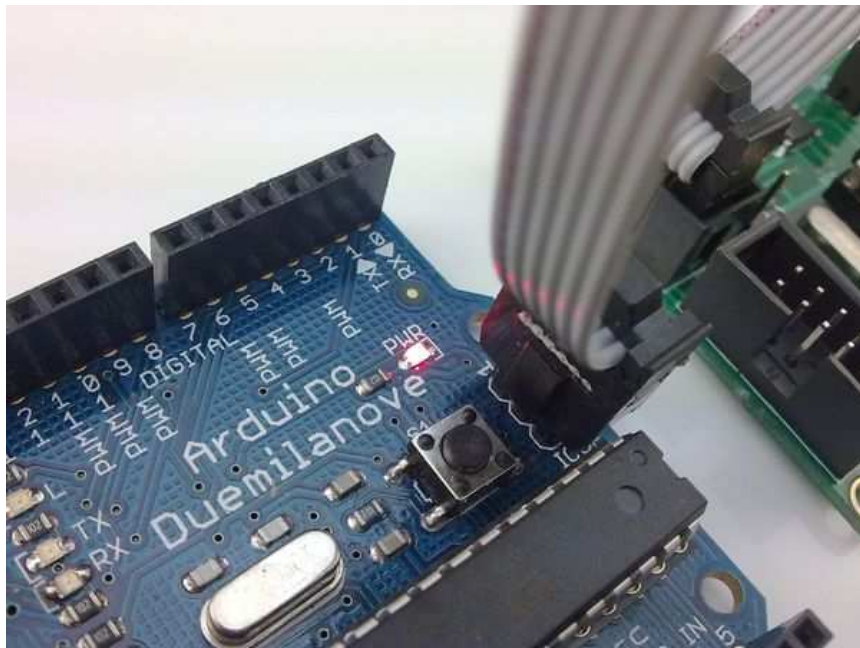


3. 给 Arduino2009 下载 Bootloader

用 6 芯的连接线将 USBtinyISP 和 Arduino 2009 板上的 ICSP 座连接起来，如下图，此时 Arduino 2009 板处的 PWR 指示红灯会亮起。

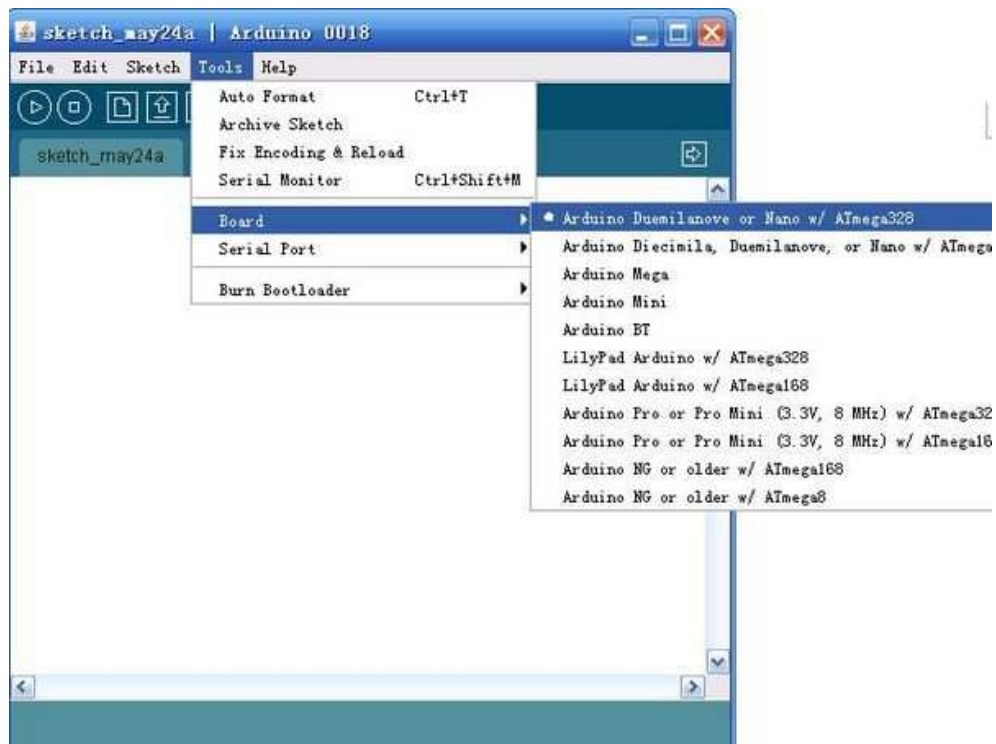


要注意的是 6 芯线的红线处位置:

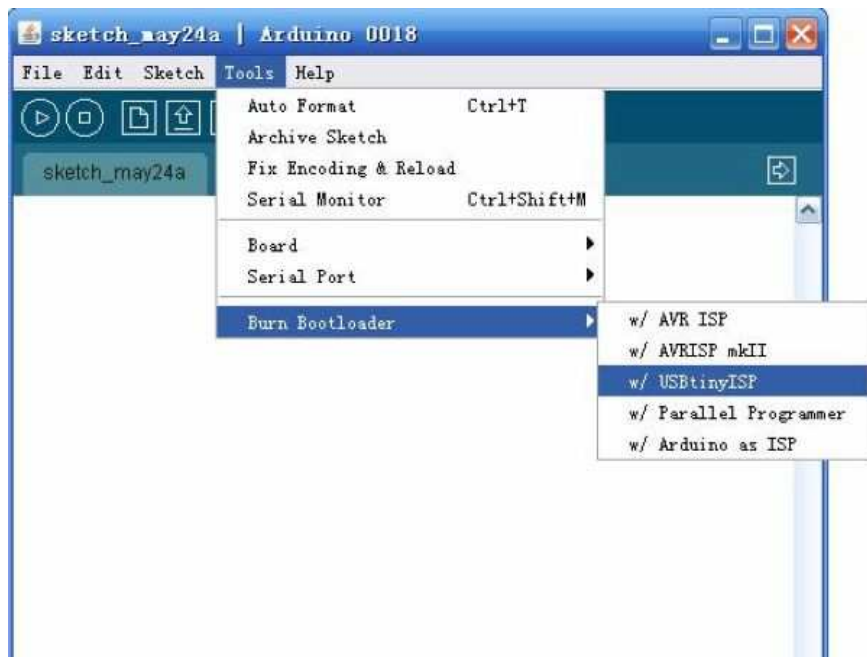




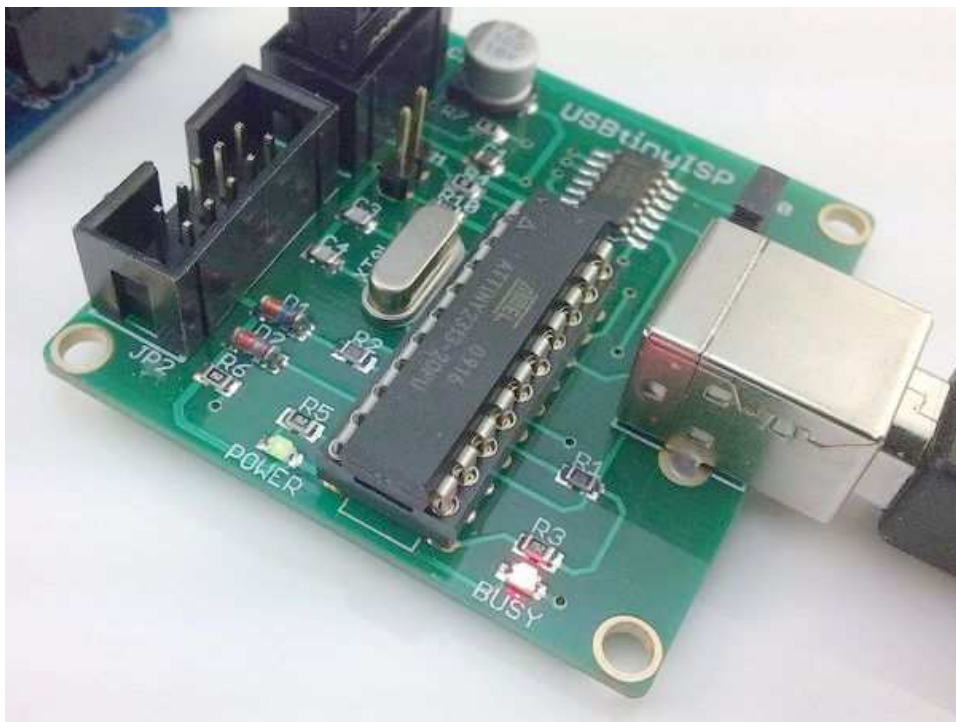
打开下载解压后的 Arduino0018, 然后选择 Tools/Board/Arduino Duemilanove



在 Burn Bootloader 下选择 USBtinyISP.



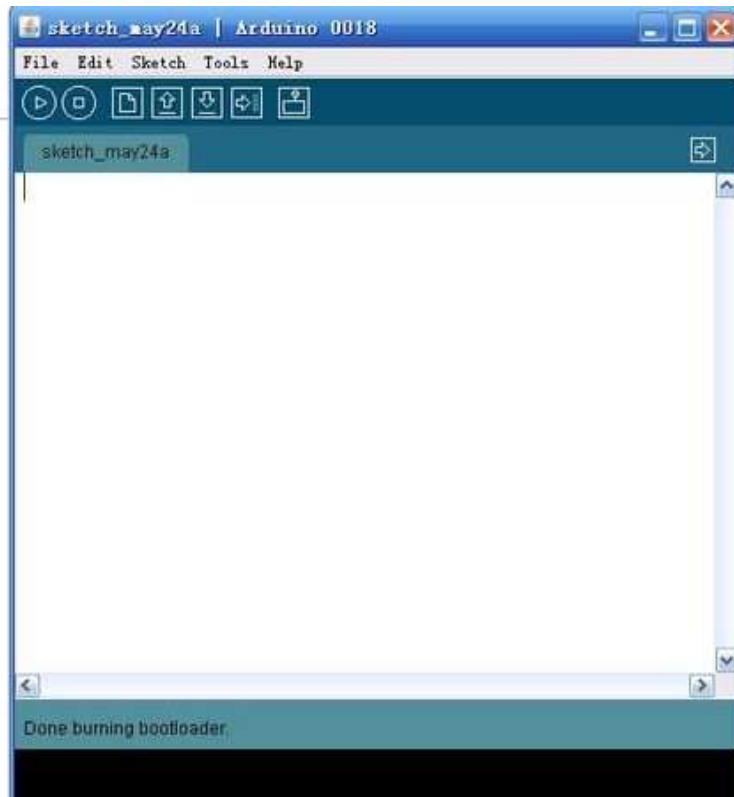
此时程序进入烧程序状态，USBTINYISP 板的 BUSY 红灯亮起，



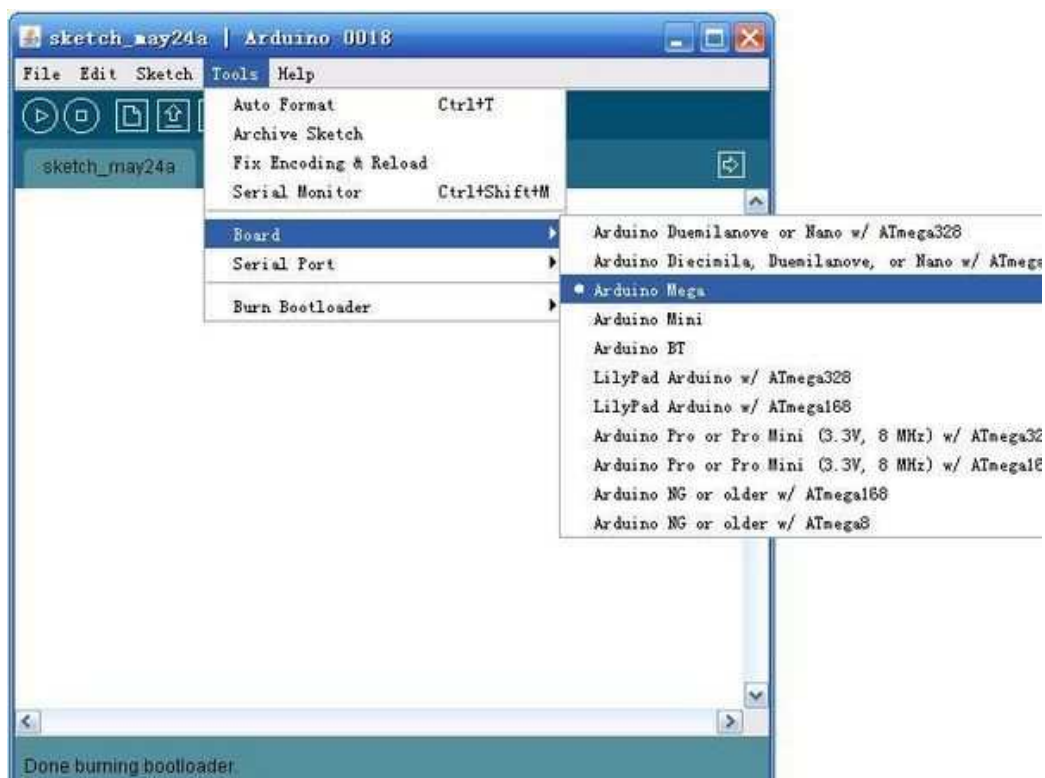
Arduino IDE 界面如下



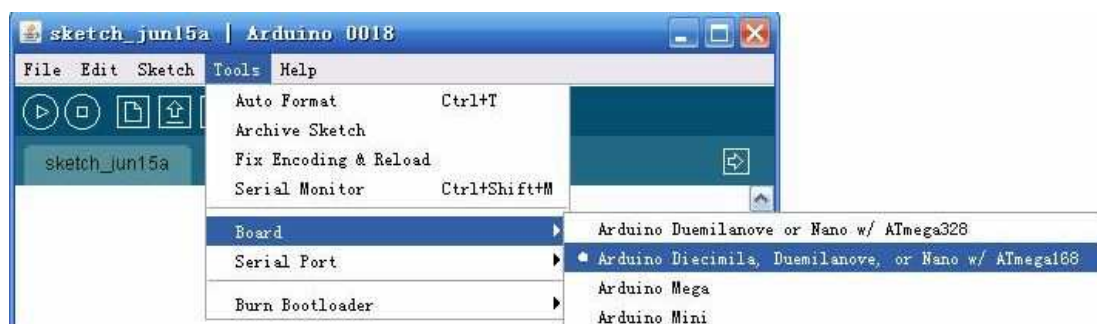
等待 10~20 秒，如现如下界面，系统程序烧完，Arduino 2009 板上的 L 的红色 LED 开始闪烁。



如果是下载 MEGA 板的 Bootloader, 基本步骤与下载 Arduino 2009 板的 Bootloader 一样, 需注意的是在打开 Arduino IDE 时要选择的 Board 为 Mega, 如下图。



如果是下载 Arduino 2009(168)板的 Bootloader, 基本步骤与下载 Arduino 2009(328)板的 Bootloader 一样, 需注意的是在打开 Arduino IDE 时要选择的 Board 为 Arduino Diecimila, Duemilanove, or Nono w/ATmega168, 如下图。



需要注意的是如果是 168P 的话, 要在 Arduino 0018 里做一些小小修改才能用 USBTINYISP 来烧写 Bootloader, 在 Arduino 0018 目录下有 `arduino-0018\hardware\tools\avr\etc` 打开 `avrdude` 文件, 找到如下代码:

```
#-----  
# ATmega168  
#-----  
  
part  
    id            = "m168";  
    desc          = "ATMEGA168";  
    has_debugwire = yes;  
    flash_instr   = 0xB6, 0x01, 0x11;  
    eeprom_instr  = 0xBD, 0xF2, 0xBD, 0xE1, 0xBB, 0xCF, 0xB4, 0x00,  
                   0xBE, 0x01, 0xB6, 0x01, 0xBC, 0x00, 0xBB, 0xBF,  
                   0x99, 0xF9, 0xBB, 0xAF;  
    stk500_devcode = 0x86;  
    # avr910_devcode = 0x;  
    signature      = 0x1e 0x94 0x06;
```

把最后的 0x06 改为 0x0B，保存后就可以和烧写 168 一样的来进行 168P 的 Bootloader 烧写了。至于原因，主要是 168P 的 signature 和 168 的不一样，所以不改一下的话 Arduino 0018 会认不出来。当然，烧好后记得改回 0x06，不然就没有办法在 Arduino 0018 下进行 Upload 了。