

第十二课 Arduino 软件的操作

在安装完 Arduino IDE 后，进入 Arduino 安装目录，打开 arduino.exe 文件，进入初始界面。打开软件会发现这个开发环境非常简洁（上面提到的三个操作系统 IDE 的界面基本一致），依次显示为菜单栏、图形化的工具条、中间的编辑区域和底部的状态区域。Arduino IDE 用户界面的区域功能如图 12-1 所示。

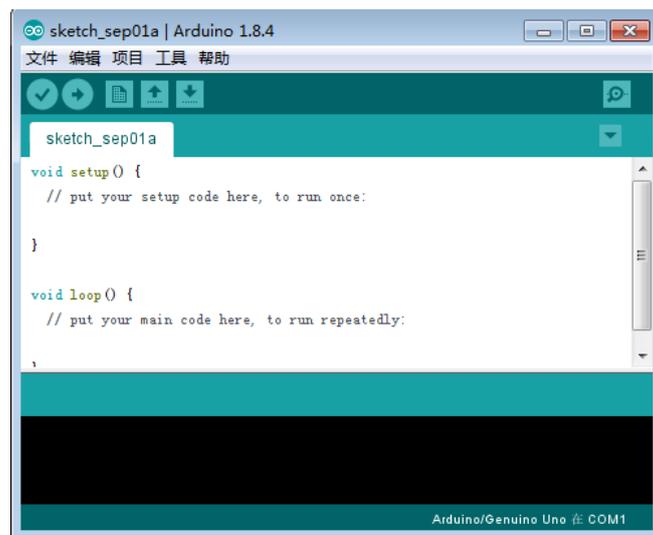


图 12-1 Arduino IDE 的工作界面

图 12-2 为 Arduino IDE 界面工具栏，从左至右依次为编译、上传、新建程序（sketch）、打开程序（sketch）、保存程序（sketch）和串口监视器（Serial Monitor）。

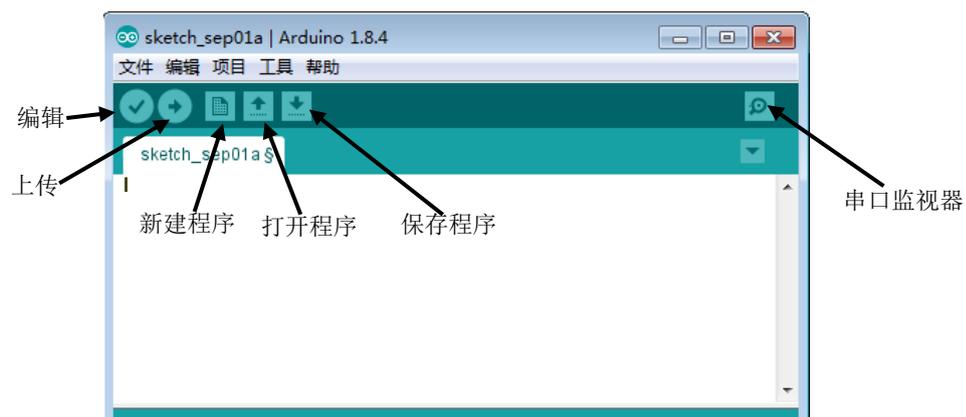


图 12-2 Arduino 工具栏

编辑器窗口选用一致的选项卡结构来管理多个程序，编辑器光标所在的行号在当前屏幕的左下角。

12.1 文件菜单

写好的程序通过文件的形式保存在计算机时，需要使用文件（File）菜单，文件菜单常用的选项包括：

- 新建文件（New）；
- 打开文件（Open）；
- 保存文件（Save）；
- 文件另存为（Save as）；
- 关闭文件（Close）；
- 程序示例（Examples）；
- 打印文件（Print）。

其他选项，如“程序库”是打开最近编辑和使用的程序，“参数设置”可以设置程序库的位置、语言、编辑器字体大小、输出时的详细信息、更新文件后缀（用后缀名.ino代替原来的.pde后缀）。“上传”选项是对绝大多数支持的 Arduino I/O 电路板使用传统的 Arduino 引导装载程序来上传。

12.2 编辑菜单

紧邻文件菜单右侧的是编辑（Edit）菜单，编辑菜单顾名思义是编辑文本时常用的选项集合。常用的编辑选项为恢复（Undo）、重做（Redo）、剪切（Cut）、复制（Copy）、粘贴（Paste）、全选（Select all）和查找（Find）。这些选项的快捷键也和 Microsoft Windows 应用程序的编辑快捷键相同。恢复为 Ctrl+Z、剪切为 Ctrl+X、复制为 Ctrl+C、粘贴为 Ctrl+V、全选为 Ctrl+A、查找为 Ctrl+F。此外，编辑菜单还提供了其他选项，如“注释（Comment）”和“取消注释（Uncomment）”，Arduino 编辑器中使用“//”代表注释。还有“增加缩进”和“减少缩进”选项、“复制到论坛”和“复制为 HTML”等选项。

12.3 程序菜单

程序（Sketch）菜单包括与程序相关功能的菜单项。主要包括：

- “编译/校验 (Verify)”，和工具条中的编译相同。
- “显示程序文件夹 (Show Sketch Folder)”，会打开当前程序的文件夹。
- “增加文件 (Add File)”，可以将一个其他程序复制到当前程序中，并在编辑器窗口的新选项卡中打开。
- “导入库 (Import Library)”，导入所引用的 Arduino 库文件。

12.4 工具菜单

工具 (Tools) 菜单是一个与 Arduino 开发板相关的工具和设置集合。主要包括：

- “自动格式化 (Auto Format)”，可以整理代码的格式，包括缩进、括号，使程序更易读和规范。
- “程序打包 (Archive Sketch)”，将程序文件夹中的所有文件均整合到一个压缩文件中，以便将文件备份或者分享。
- “修复编码并重新装载 (Fix Encoding & Reload)”，在打开一个程序时发现由于编码问题导致无法显示程序中的非英文字符时使用的，如一些汉字无法显示或者出现乱码时，可以使用另外的编码方式重新打开文件。
- “串口监视器 (Serial Monitor)”，是一个非常实用而且常用的选项，类似即时聊天的通讯工具，PC 与 Arduino 开发板连接的串口-交谈||的内容会在该串口显示器中显示出来，如图 2-16 所示。在串口监视器运行时，如果要与 Arduino 开发板通信，需要在串口监视器顶部的输入栏中输入相应的字符或字符串，再单击发送 (Send) 按钮就能发送信息给 Arduino。在使用串口监视器时，需要先设置串口波特率，当 Arduino 与 PC 的串口波特率相同时，两者才能够进行通讯。Windows PC 的串口波特率的设置在计算机设备管理器中的端口属性中设置。
- “串口”，需要手动设置系统中可用的串口时选择的，在每次插拔一个 Arduino 电路板时，这个菜单的菜单项都会自动更新，也可手动选择哪个串口接开发板。
- “板卡”，用来选择串口连接的 Arduino 开发板型号，当连接不同型号的开发板时需要根据开发板的型号到 “板卡||选项中选择相应的开发板。

- “烧写 Bootloader”，将 Arduino 开发板变成一个芯片编程器，称为 AVRISP 烧写器，也可以到 Arduino 中文社区查找相关内容。

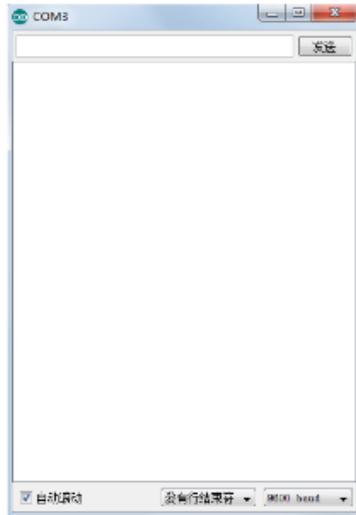


图 12-3 Arduino 串口监视器的生成

12.5 帮助菜单

帮助 (Help) 菜单是使用 Arduino IDE 时可以迅速查找帮助的选项集合。包括快速入门、问题排查和参考手册,可以及时帮助了解开发环境,解决一些遇到的问题。访问 Arduino 官方网站的快速链接也在帮助菜单中,下载 IDE 后首先查看帮助菜单是个不错的习惯。

12.6 小试身手

在下载并安装好 IDE 之后,下一步就可以实践了。通过编写和上传第一个程序,正式进入 Arduino 的世界。在本节中将要展示 Arduino 的基本工作流程,并且通过实例来展示如何将程序导入到 Arduino 中。

12.6.1 程序的加载

在硬件的世界里,使用灯光的闪烁代表 hello world,下面我们编写第一个 Sketch! 打开 Arduino IDE 后,需要新建一个空的 Sketch。之后就可以在编辑器上编写第一个 Sketch,如程序一所示。

```
void setup()
{
  pinMode(13,OUTPUT);    //将 13 号引脚设置为输出引脚
}

void loop()
{
  digitalWrite(13,HIGH);    //13 引脚输出高电平，小灯被点亮
  delay(1000);
  digitalWrite(13,LOW);    //13 引脚输出低电平，小灯熄灭
  delay(1000);
}
```

程序一 点亮 Arduino 的世界

这个例子是 Arduino 示例 Basics 中的 Blink 程序，也可以通过图 12-4 所示的操作打开该程序。Blink 作为 Arduino 入门的初始程序非常简洁易懂，在每句话的后面作者都给出了注释，官方示例中的程序如图 12-5 所示。



图 12-4 Blink 的查找

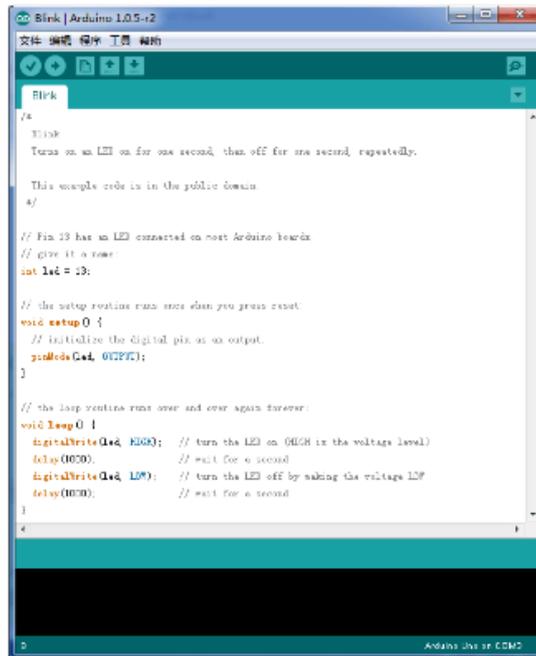


图 12-5 Blink 程序的说明

编写或者打开 Blink 之后，便可以连接 Arduino 开发板，将开发板的 USB 接口连接到电脑上，当系统提示安装成功，并且开发板的绿色“ON”指示灯亮起时，就可以进行 Blink 的上传。单击“上传”按钮，再经过短暂的几秒烧写之后，会发现开发板的串口指示灯闪烁了数次，提示成功之后，开发板装载的 LED 灯便开始不停闪烁。

之后再来看状态区域，状态区域显示“下载成功”和“二进制程序大小 1018 字节”的字样。我们的第一个 Arduino 程序就成功下载并运行了。

12.6.2 酷炫的机械舞蹈

动作是机器人的灵魂，那如何使机器人完成我们需要的目标动作呢？答案就是使用 Arduino 程序来完成的，接下来我们可以将教程附录中的机器人动作程序导入到 Arduino 中来控制 RoboHero 的动作。

实例一：机器人挥手打招呼的动作

```
01. #include <Servo.h>
02. Servo myservo; //创建一个舵机控制对象
03. // 使用Servo类最多可以控制8个舵机
04. int pos = 0; // 该变量用于存储舵机角度位置
05. /*-----华丽的分割线-----*/
06. void setup()
07. {
08.   myservo.attach(9); // 该舵机由arduino第九脚控制
09. }
10. /*-----华丽的分割线-----*/
11. void loop()
12. {
13.   for(pos = 0; pos < 180; pos += 1) // 从0度到180度运动
14.   { // 每次步进一步
15.     myservo.write(pos); // 指定舵机转向的角度
16.     delay(15); // 等待15ms让舵机到达指定位置
17.   }
18.   for(pos = 180; pos >= 1; pos -= 1) // 从180度到0度运动
19.   {
20.     myservo.write(pos); // 指定舵机转向的角度
21.     delay(15); // 等待15ms让舵机到达指定位置
22.   }
23. }
```

程序二 程序范例

按照上面介绍的方法将程序烧入到 Arduino 中，就可以看到我们的机器人“活了起来”！

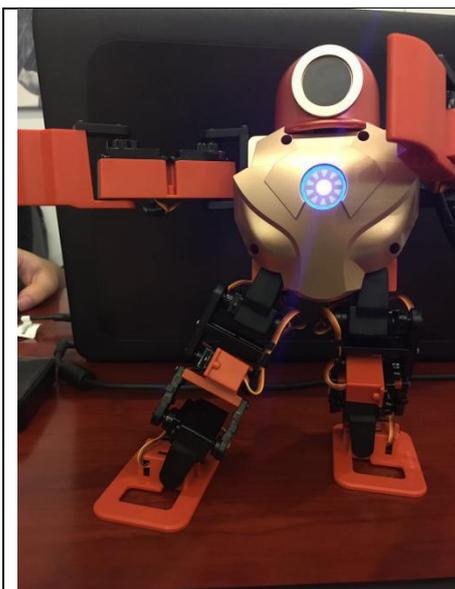


图 12-6 动作 xx



图 12-7 动作 xx

课后思考：

- 大家明白机器人每个动作是怎么实现的吗？
- 如果我们要自己设定动作程序的话，要怎么修改程序呢？

(逐条语句的含义是机器人可以动作的关键，通过对语句中参数的控制就可以实现机器人不同动作的完成。鼓励学生来自己动手进行修改，发挥想象力将打招呼动作修改为握手，再见等动作；)