



什么是
Arduino?



它是一个开源电子原型
平台。

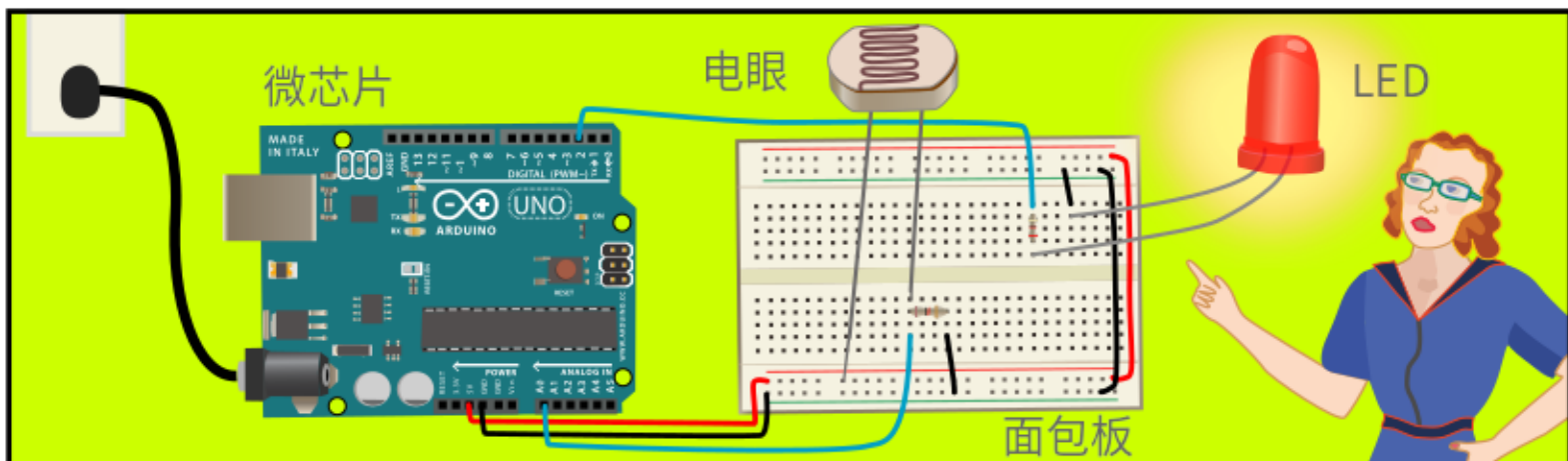
那是什么意思?

开源- "可以免费使用, 重新分发或改写的资源. 通常是软件或硬件"

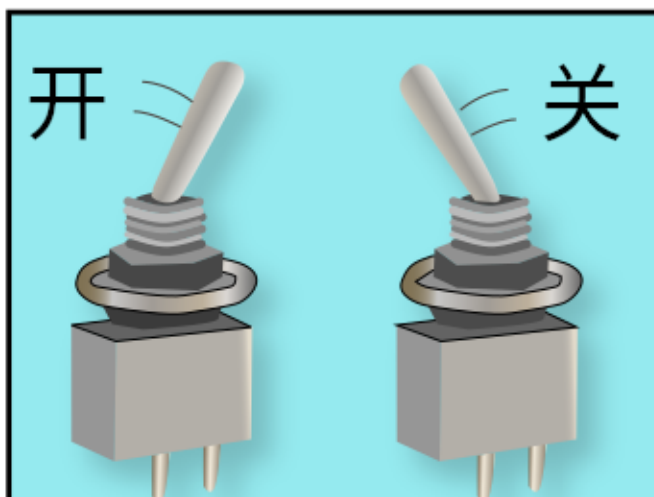
电子产品- "利用电子通过不同介质的受控运动的技术"

原型- "原始形式, 可以作为其他事物的基础或标准"

平台- "具有软件机架的硬件架构, 他软件可以在其上运行"



Arduino包含一个微芯片, 这是一个非常小的计算机, 你可以编程. 你可以在其上安装可以测量条件的传感器(例如房间里有多少光). 它可以控制其他物体对这些条件的反应(房间变暗, LED亮起).

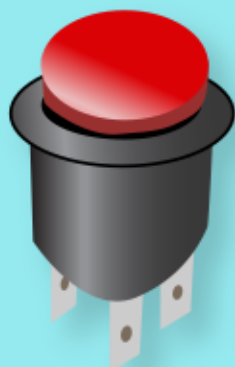


按下按钮时它可以应对.

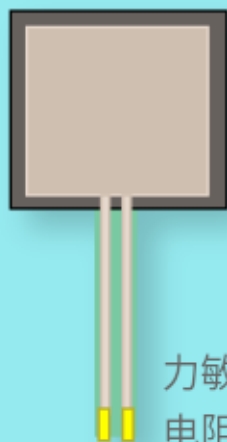
鼠标是台式计算机的常用
输入设备,
而监视器是常见的输出设备



像任何计算机一样微控制器使用输入和输出. 输入从用户或环境捕获信息, 而输出对已捕获的信息执行某些操作.



瞬时开关



力敏感
电阻

Arduino的输入可以是开关或传感器。



直流电机



我们想要打开、关闭和控制的任何对象都可以是输出。
这对象可以是电机甚至是电脑。



数字和模拟输入和输出之间有什么区别？

输入和输出可以是数字或是模拟。数字信息是二进制 - 要是真(true)要是假(false)模拟信息是连续的, 它能具有一系列值。

数码信息是离散和有限的。所有信息都以两种状态描述1或0, 开或关。

模拟信息的特征在于它的连续性。它可以具有无限数量的可能值。



开关是数字输入, 传感器是模拟输入。
模拟传感器的范围受其转换为数字数据的限制。

电压？电流？
电阻？
欧姆定律？



插入Arduino之前, 我们应该回顾一些与电能有关的术语和原则, 为了理解电子产品如何运行的。

电压
(符号V)

是电路中电势的
量度。
它以
伏特
为单位。

电流
(符号I)

是流过导
电材料的
流量。
它以
安培
为单位。

电阻
(符号R)

是材料对
电流流量的
反对。
它以
欧姆
为单位。

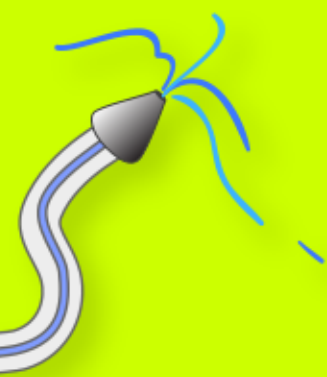
电是通过导电材料的电能流

流量的速度是由**电压**限定

电阻增加或减少流量

通过管道的流量是**电流**

水比喻通常用于解释这些术语。这是一个模型。



欧姆定律

电流 = 电压 / 电阻

$$(I = V/R)$$

或

电阻 = 电压 / 电流

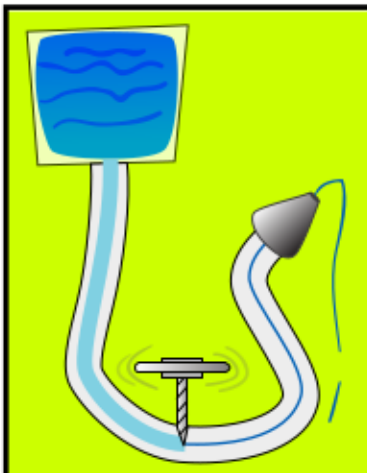
$$(R = V/I)$$

或

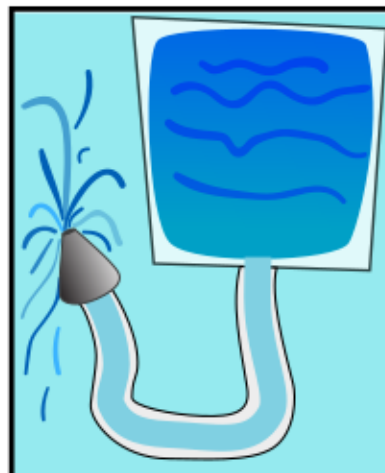
电压 = 电阻 * 电流

$$(V = R * I)$$

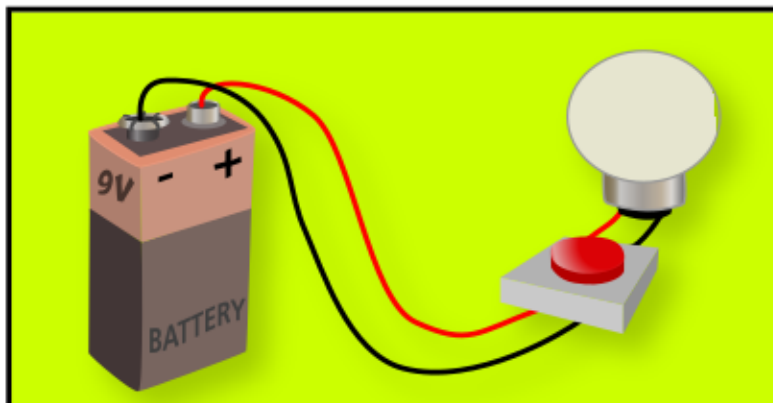
德国物理学家 乔治·欧姆 发现了电压, 电流和电阻之间的关系.



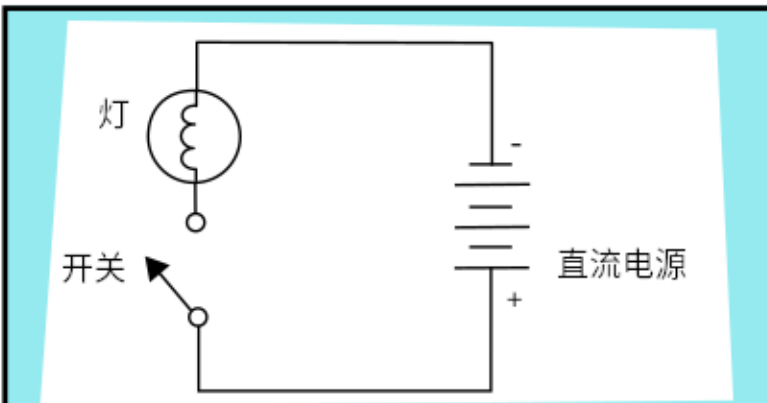
例如增加阻力, 减少流量.



或增加电势, 更多流量.



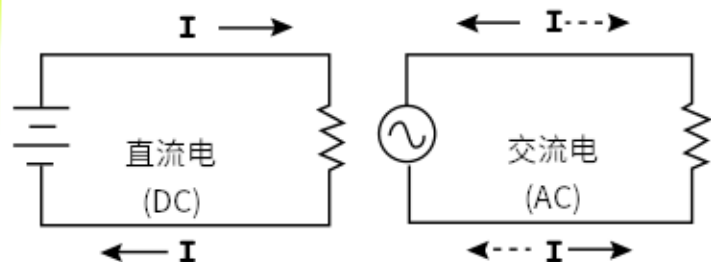
现在让我们看一个简单的电路. 每个电路都是一个闭环, 有一个能源(电池)和一个负载(灯). 负载转换电池的电能并将其用完. 这个也有一个开关.



这是相同电路的示意图

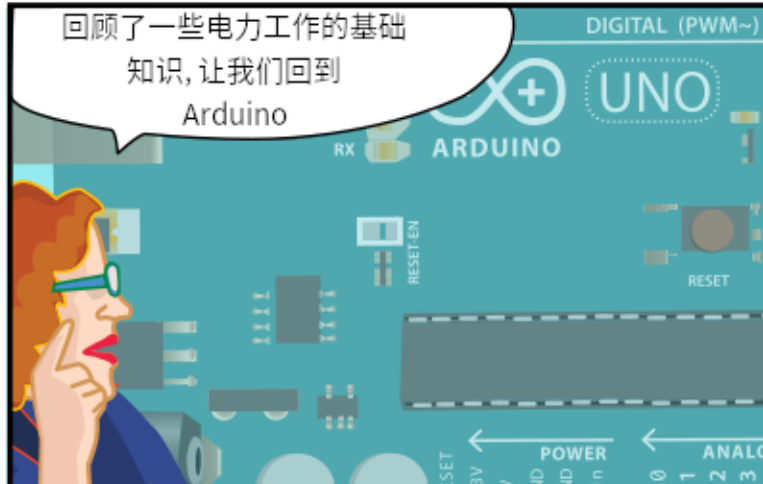
(它表示使用电子元件符号的电路).

当开关闭合时, 电流从电源流出并点亮灯.



有两种常见类型的电路, 直流电和交流电. 在直流电路中, 电流总是沿一个方向流动. 在交流电路中, 电流为定期循环以相反的方向流动. 我们这里只想谈谈直流电路.

回顾了一些电力工作的基础知识, 让我们回到 Arduino



Arduino需要电力才能运行.
我们需要把它连接到电脑为了编程它.



使用USB电缆把Arduino连接到电脑提供我们需要的电源并允许我们开始编程.



这里下载

<https://www.arduino.cc/en/Main/Software?setlang=cn>

为了编程你的Arduino你必须下载并安装软件. 从上面的网址可以免费获得. Arduino软件可在苹果电脑Windows系统和Linux系统上运行.

如何在苹果电脑上安装的说明

<https://www.arduino.cc/en/Guide/MacOSX?setlang=cn>

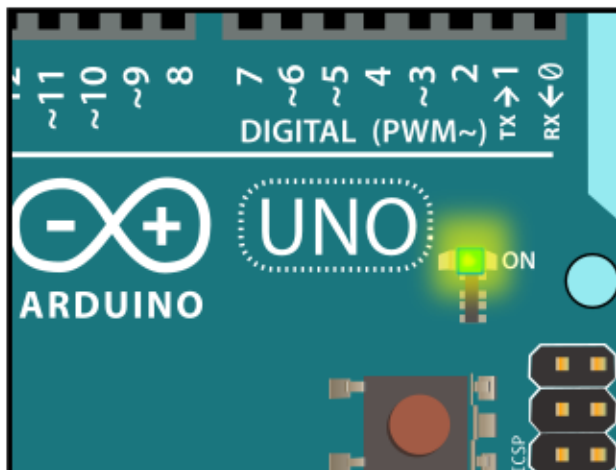
如何在微软Windows系统上安装的说明

<http://www.arduino.cc/en/Guide/Windows?setlang=cn>

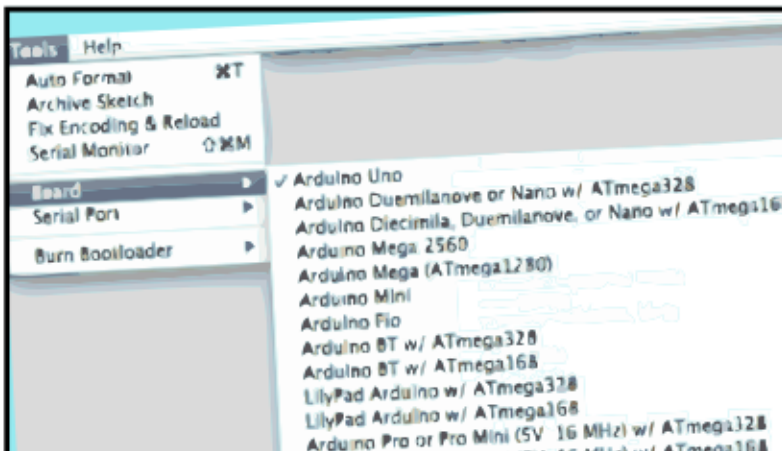
如何在Linux系统上安装的说明

<http://www.arduino.cc/playground/Learning/Linux?setlang=cn>

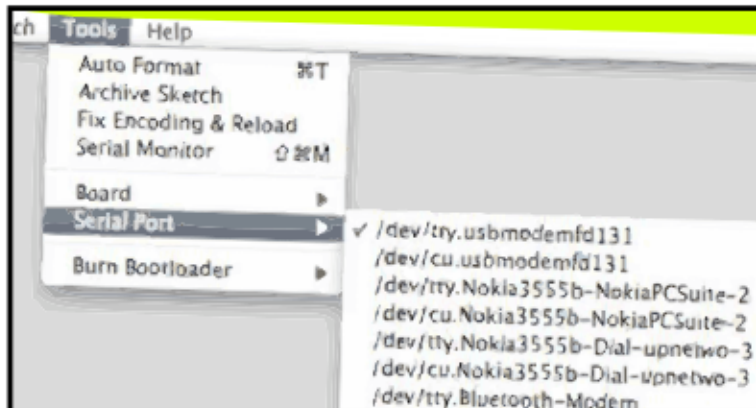
有关安装软件详细说明, 请访问上面的网址.



已安装软件时, 连接你的Arduino. 标记为ON的LED应该在开发板上亮起.



启动Arduino软件. 在工具专栏选择你正在使用的开发板 (工具>开发板). 例如Arduino Uno.

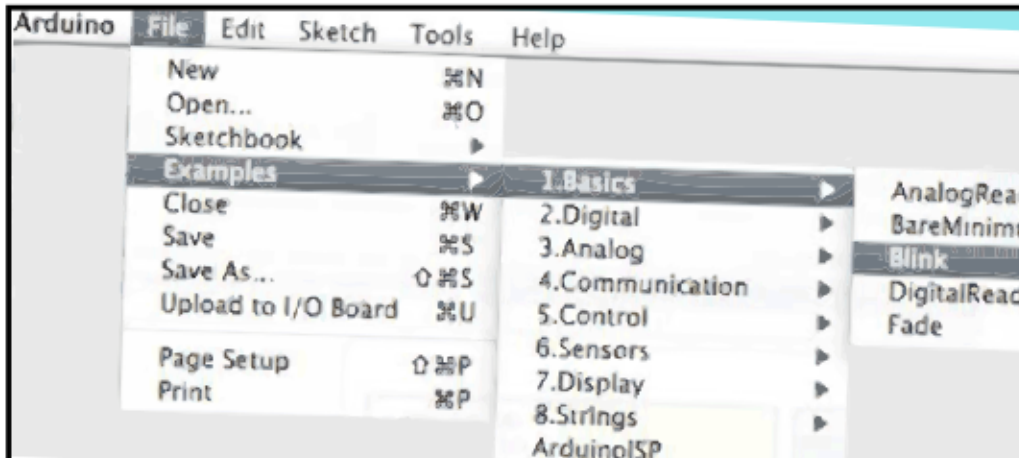


接下来选择端口.(工具>端口). 在苹果系统上它该类似于dev/tty.usbmodem 在Windows系统上它该类似于COM3. 在Linux系统上它该类似于ttyACM0.

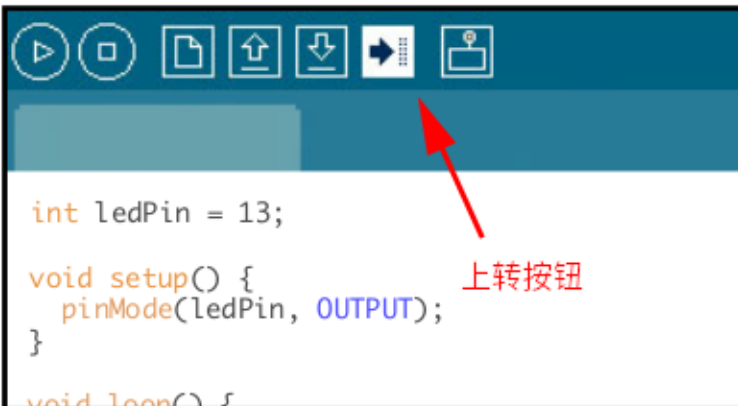
什么是集成开发环境？



下载了Arduino软件时，你才下载了集成开发环境。它结合文本编辑器和编译器以及其他功能，以帮助程序员开发软件。

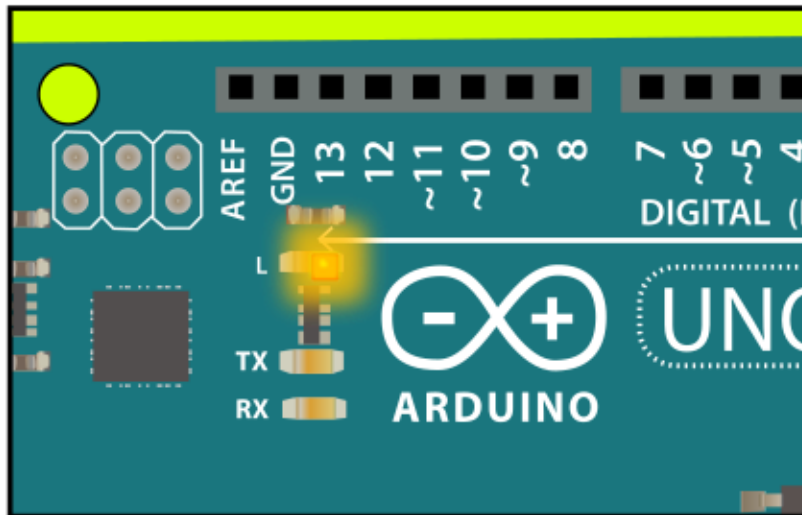


Arduino软件允许你编写项目或程序并将其上转到Arduino板。在文件菜单中打开Blink(眨)示例。文件>示例>1.Basics>Blink



上转按钮

要将项目上转到Arduino板，请点击窗口顶部上转按钮。一些消息将出现在窗口的底部，最后完成上传。



Arduino针脚13号上的LED开始闪烁。

```

void setup() {
    // 将数字引脚初始化为输出。
    // 针脚13在大多数Arduino板上连接了LED:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH); // 开启LED
    delay(1000);           // 等一下
    digitalWrite(13, LOW);  // 关闭LED
    delay(1000);           // 等一下
}

```

项目就像用任何语言编写的程序一样, 是计算机的一组指导. 如果我们仔细看一下Blink项目, 我们会看到两个主要部分, 即setup(设置)和loop(循环).

setup(设置): 程序开始运行时发生一次

loop(循环): 一遍又一遍地重复

这些都是每个项目都具有的称为functions(功能)的代码块. 它们被花括号 {} 挡住了.

<http://arduino.cc/en/Reference/HomePage>



在Arduino网站上查找Arduino参考指南和许多其他资源, 以学习该语言.

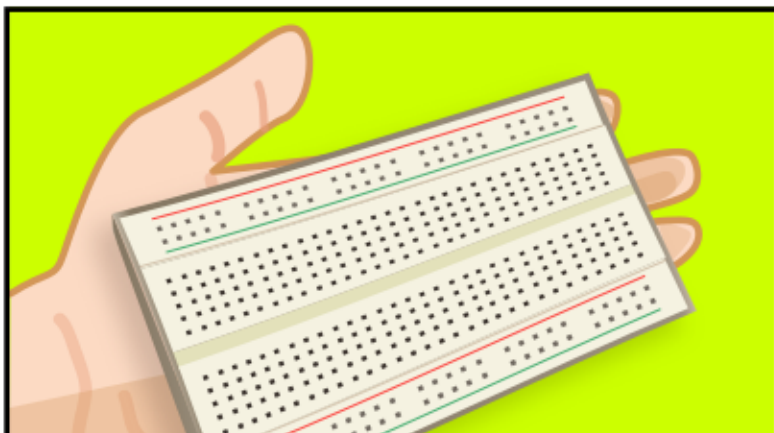
```

void setup() {           //声明代码块
    pinMode(13, OUTPUT); //将引脚13设置为输出
}                         //结束代码块

void loop() {            //声明代码块
    digitalWrite(13, HIGH); //将引脚13设置为高
    delay(1000);           //暂停1秒
    digitalWrite(13, LOW);  //将引脚13设置为低
    delay(1000);           //暂停1秒
}                         //结束代码块

```

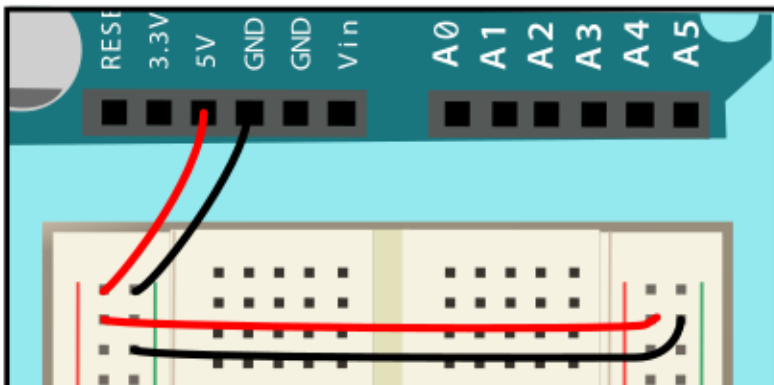
现在, 让我们逐行看一下这个简单的脚本, 看看每一行的作用.



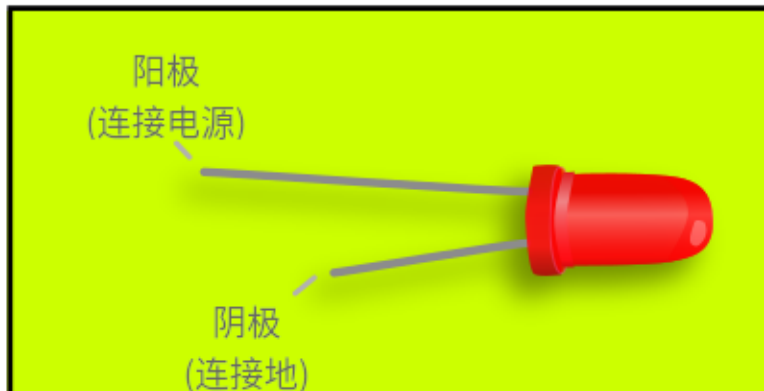
我们如何控制不在Arduino板上的组件？
我们将Arduino连接到无焊面包板上。
这将使我们能够快速设置(setup)和测试电路。



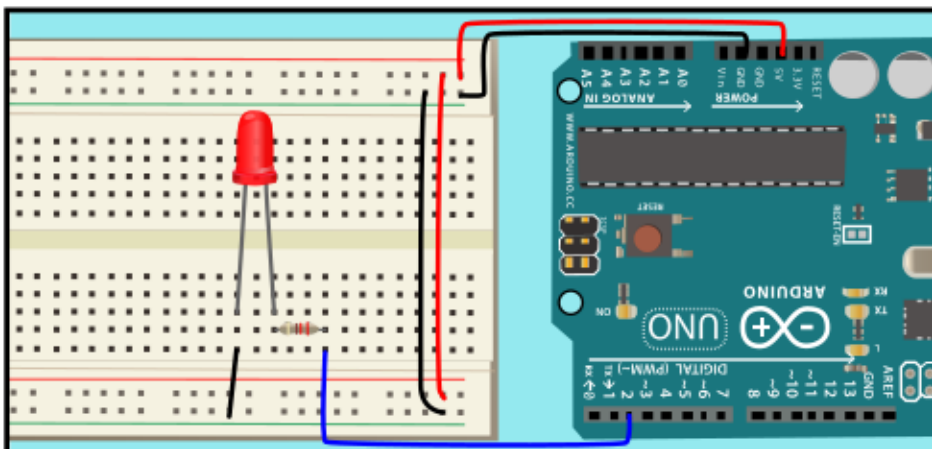
该面包板在左侧和右侧向下延伸有2行孔，
在中间凹痕的两侧各有5行孔。
侧排垂直连接，中间的每排5个孔水平连接。



我们将使用22号线将Arduino板上的电源和地线连接到
左右两侧的垂直连接板。
可以根据需要将其他组件连接到中间的孔以及电源和地。



当电流以正确的方向流过LED(发光二极管)时，它会点亮。
我们将在面包板上安装一个LED，然后在Arduino上安装
一个LED，以便我们可以通过代码对其进行控制。

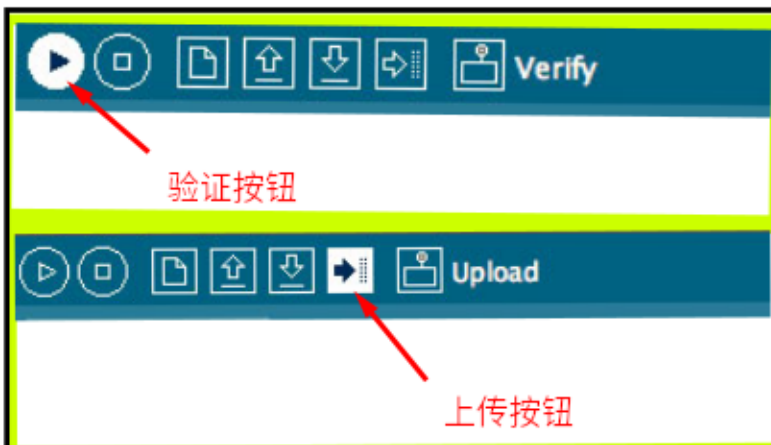


阳极通过220欧姆电阻连接到Arduino的引脚2. 阴极接地. 引脚2至13可以配置为数字输入或输出. 单击“新建”按钮开始绘制项目.

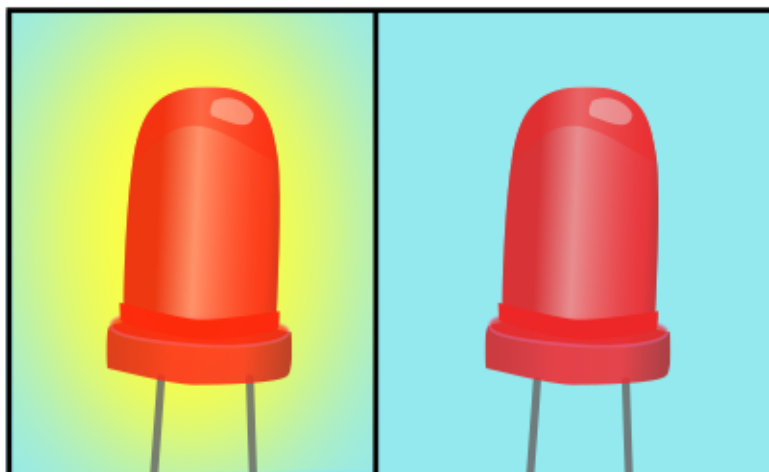
```
void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  delay(500);
}
```

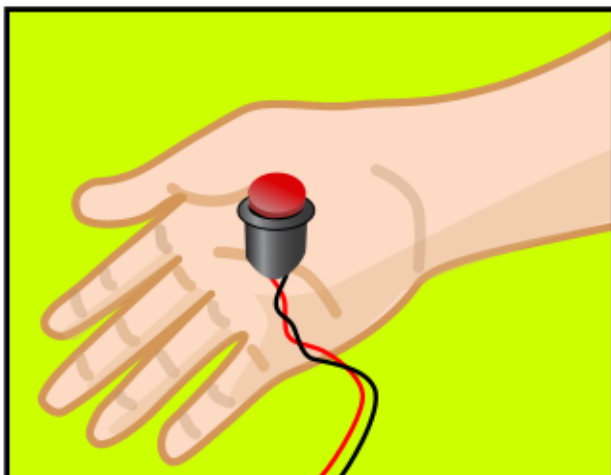
在设置中, 我们将引脚2设置为输出. 在循环中, 我们首先将引脚2设置为高电平, 使LED点亮. 延迟暂停500毫秒或半秒. 当引脚2设置为低电平时, LED熄灭, 我们再暂停半秒



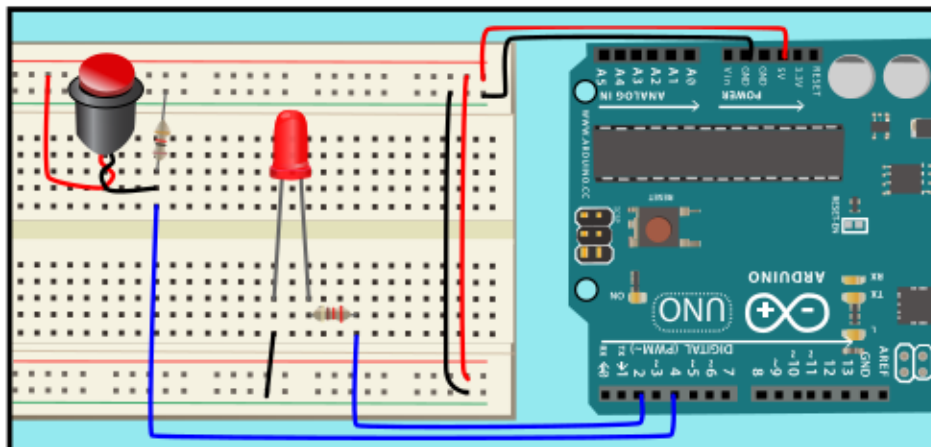
单击菜单上的验证以检查你的代码.
如果没有任何错误, 请单击上传以将程序放在Arduino上.



LED反复闪烁半秒钟, 然后闪烁半秒钟.



接下来, 我们将添加一个开关, 一个数字输入, 因此可以打开和关闭LED

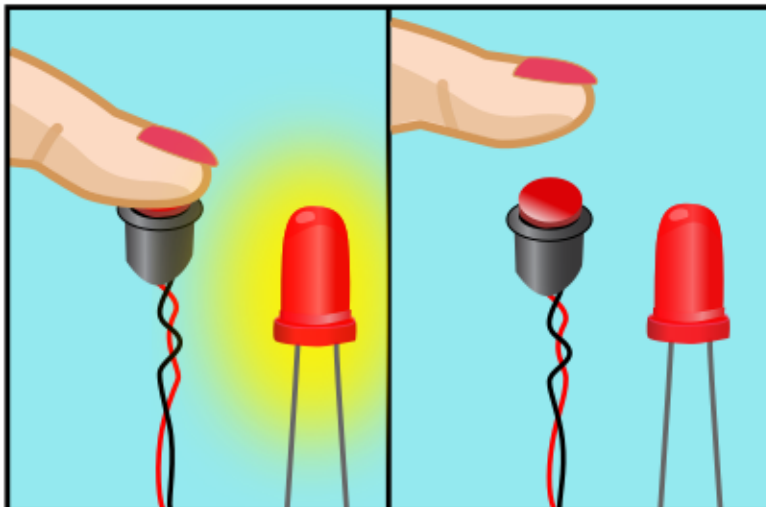


将瞬时开关的一端连接到Arduino的引脚4, 并将1万欧姆电阻接地并连接到同一端. 将另一端连接到电源. 我们将LED留在同一引脚上.

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
}

void loop() {
  if(digitalRead(4)){
    digitalWrite(2, HIGH);
  }else{
    digitalWrite(2, LOW);
  }
}
```

接下来, 我们将编写代码. 在setup中, 我们将引脚2声明为输出, 将引脚4声明为输入. 在loop中, 我们使用if语句, 如果将引脚4读为高电平, 则将LED引脚设置为高电平, 否则将LED引脚设置为低电平, 将其关闭.

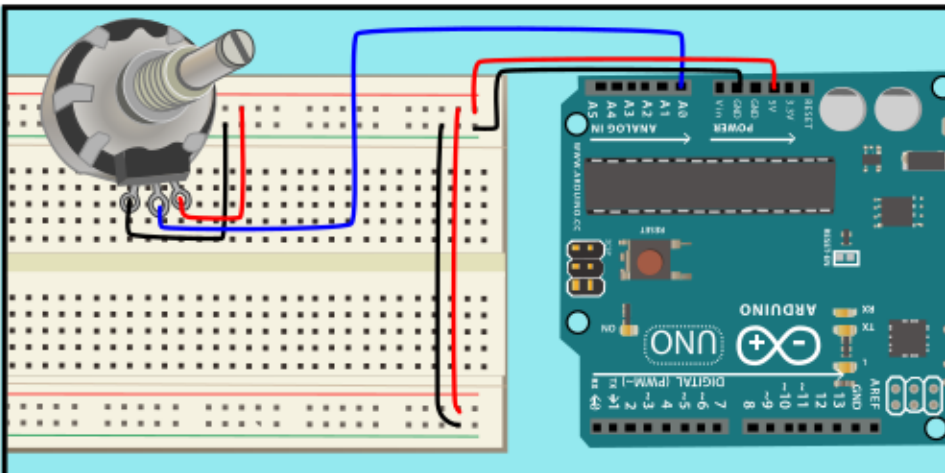


按住开关时, LED点亮.

电位计或电位器是可变电阻器。电阻的大小随转动的方向而变化，取决于转动的方向而增加或减小。



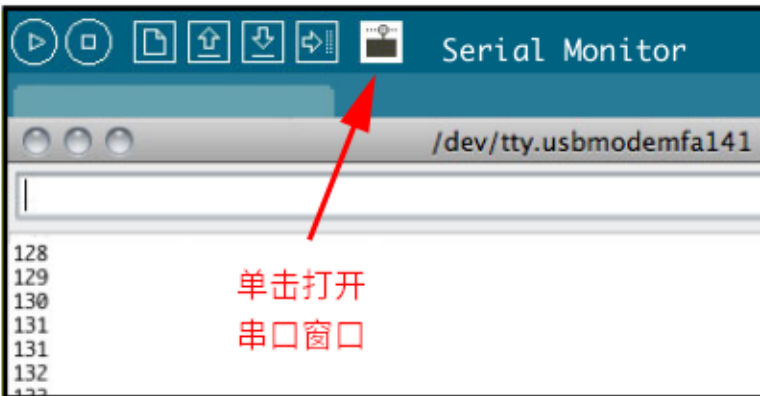
现在我们将设置一个模拟输入。我们将使用电位计。



将电位器上的中间引脚连接到模拟引脚A0。
将电位器的一端连接到电源，另一端接地。

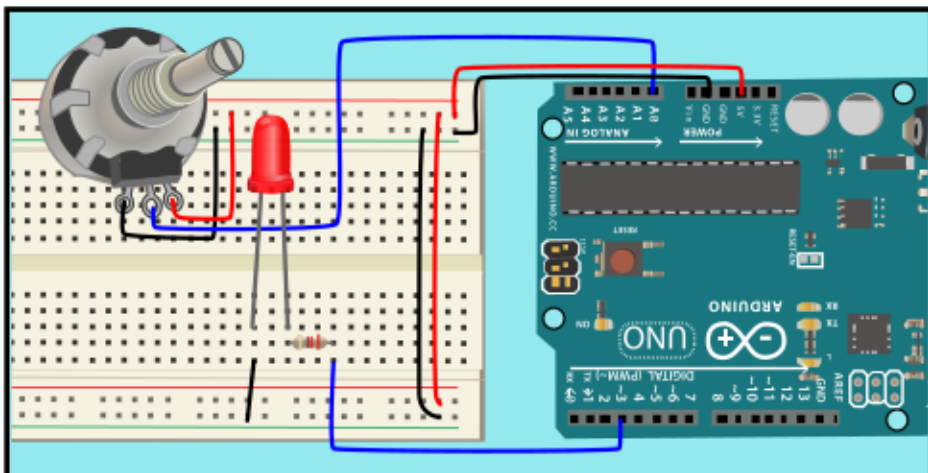
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(A0));  
}
```

首先，我们将看一下通过使用串行监视器旋转电位器得到的值的范围。在我们的代码中，我们在setup中初始化串行对象，将波特率设置为9600。在loop中，我们从模拟引脚A0读取值，然后使用println函数将其打印到串行对象中。

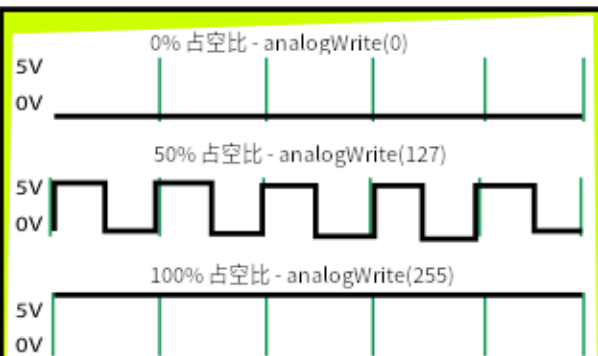


单击打开
串口窗口

将脚本上传到Arduino之后，单击串行监视器按钮，以便在转动电位计时查看值。将打开一个窗口，旋转电位器时，你会看到介于0到1023之间的值。



让我们使用从电位计接收到的变化值作为调光器来控制LED。
通过电阻将阳极连接到板的引脚3上, 阴极接地



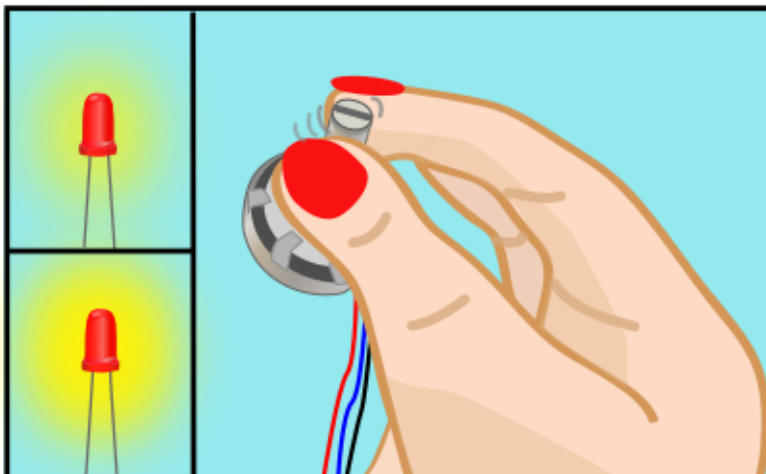
我们将使用脉冲宽度调制(PWM). 这是一种通过操纵电压, 以不同的速率或占空比打开和关闭电压来模拟模拟值的方法. 你可以将PWM与引脚3, 5, 6, 9, 10和11一起使用.

```
int sensorValue = 0;

void setup() {
  pinMode(3, OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  analogWrite(3, sensorValue/4);
}
```

首先, 我们创建一个变量来存储电位计的值. 在设置中, 我们将引脚3设为输出. 在循环中, 我们将从变量A0引脚读取的值存储在变量中. 然后, 将值写入LED引脚3. 我们必须将变量除以4, 所以我们将得到一个范围从0到255的值, 或者一个字节.



转动电位器时, LED的亮度会发生变化, 从完全关闭到非常亮.



而已！
这是一个非常简短的介绍。
在接下来的面板中，有链接和
其他资源。全部查看，你
会发现更多！

链接

软件

下载软件

<http://www.arduino.cc/en/Main/Software>

语言参考

<http://arduino.cc/en/Reference/HomePage>

商品

Sparkfun Electronics

<http://www.sparkfun.com/>

Adafruit Industries

<http://adafruit.com/>

Maker Shed

<http://www.makershed.com/>

Jameco Electronics

<http://www.jameco.com/>

教程

Arduino网站的教程

<http://www.arduino.cc/en/Tutorial/HomePage>

Lady Ada

<http://www.ladyada.net/learn/arduino/>

Instructables

<http://www.instructables.com/tag/type-id/category-technology/channel-arduino/>

英文书籍

Getting Started with Arduino by Massimo Banzi

Making Things Talk: Using Sensors, Networks, and Arduino to See, Hear, and Feel Your World by Tom Igoe

Physical Computing: Sensing and Controlling the Physical World with Computers by Dan O'Sullivan & Tom Igoe

Arduino Cookbook by Michael Margolis

所有文字和绘图均由Jody Culkin制作，
有关更多信息，请访问jodyculkin.com

特别感谢Tom Igoe, Marianne Petit, Calvin Reid, 纽约大学互动电信计划的教职员工, Dan O' Sullivan, Danny Rozin and Red Burns. 感谢Cindy Karasek, Chris Stein, Sarah Teitler, Kathy Goncharov和Zannah Marsh.

非常感谢Arduino团队为我们带来了这个强大而灵活的开源平台。

还要感谢活跃和不断发展的Arduino社区。

Jody Culkin对Arduino的介绍已获得知识共享许可
Attribution-NonCommercial-ShareAlike
3.0未移植许可证。

