OPEN
+ SOFTWEAR™

2nd edition
revised & expanded

OPEN + SOFTWEAR™

Fashionable prototyping and wearable computing using the Arduino.

Tony Olsson | David Gaetano | Samson Wiklund | Jonas Odhner

前言及前言

第一部分：基础知识

第二部分：例如

Part Three: Coding

前言

将纺织品的生命

传统的纺织品是人类生活的基本组成部分。他们是戴在身体上，并通过室内设计在服装的形式，我们日常的日常生活环境的一部分。当我们开始整合纺织电子时，会发生什么事？我喜欢考虑电子像科学怪人的属性。当您连接与电子产品，你可以把死对象生命传统材料。电子让我们给纺织品基本形式的智力，让他们感觉他们的环境，并作出回应。电子纺织品，希望从我们的东西。

我们似乎要进入一个无处不在的计算时代的标题，我们许多这样或那样的对象是我们的数字信息交流的一部分。我们不能想象没有科技的生活。因此什么技术的谈话，应该在未来，现在是至关重要的。如果您决定要启动wearables和电子纺织品，您将自动成为这次谈话的一部分。热烈祝贺和欢迎！我们期待着听取

您的声音。

这本书是一个美好的的出发点和一个很大的参考基于Arduino的wearables领域的实验。虽然有一些理论，这本书的心脏是可行的。我相信打开Softwear教育艺术家和设计师的水平，在那里他们可以开始做自己的交互式设计，为未来的能力。作为电子纺织品，它是一个复杂的的领域，我可以告诉你的设计师，但有很多乐趣。这个过程需要你不断变化的设计和技术问题之间的焦点，所以你永远不会觉得无聊。很多道路还没有铺好，冒险仍然存在。有机会开动脑筋，想象力和坚持与性的人被查获。请务必记录您的工作和与他人在线分享您的设计。电子纺织品领域的发展与每一个完成的项目。

我的建议是新进艺术家和设计师wearables：好奇和耐心;在纺织品设计和电子教育自己;要求反馈;找到了解你的工作概念，在视觉上和技术上的顾问;连接的其他人作出wearables;最后，保持测量电压水平，并有乐趣！

Melissa Coleman

梅利莎科尔曼是一个从荷兰海牙的艺术家。她在海牙皇家艺术学院任教于电子纺织品和教练在埃因霍温技术大学可穿戴式感官。她是塑造科技V2_的E -纺织工作区在“鹿特丹公约”和“客博客的创始人之一

与未完成的现实设计交易

当创建新的对象和对他人的经验，设计者想象他们的用户。他们预期的设备，以及符合人体工程学方面的美学探索世界的乐趣。物理形状，颜色，材料，最近，互动模式，尝试的过程，就是我们所知道的原型。

想象的东西很容易，把这些想法现实更为复杂。设计师的主要技能是说明，无论是在手的概念。我请我的学生寻找某种方式相似的设备，他们向往的文物。我要求他们拆除它，看看里面，破解它的功能要尽可能接近他们的目标是什么，并重新建立自己的壳。大部分的时间观念得到妥协，它是学生工作的一部分，以找到正确的方式来沟通，什么是他们为与他们取得了什么。

在现实世界中，事情并不不同。公司有时构建原型，让他们的客户尝试一个不存在的神器。有大量的设备，可分为伪造的经验，调整了。知道如何混搭技术和工艺，现在比以往任何时候都更有意义。这些杂交带来的数字技术，我们生活的各个方面。

数字电路与服装的混搭，就是我们所说的可穿戴计算。它带来技术，以尽可能接近身体，不被侵入。它是一个新兴的领域，设计师们有很多话要说。有探索性设计努力的空间，去野生的，并创造直接进入市场的现有技术的应用。有很多产品的设想，原型和内置。

Softwear图书的第一版是由马尔默大学的学生，现在进入黑客电子和时尚专业人士前瞻性。他们想帮助他人与数字电路原型的新服装。电子成型工艺是长期保留的工程师。阿尔杜伊诺，Lilypad和其他工具简化为原型技术的访问。

学习如何在这个世界上未完成的导航的，是有关塑造未来能够成为一名设计师。这本书带来耐磨技术动手感兴趣的话语。睁大眼睛，做到这一点。

David Cuartielles

大卫Cuartielles是Arduino.cc共同创始人，目前是博士候选人在马尔默大学互动设计。

序言

这本书的内容是启发物理样机实验室的教诲，在艺术与传播学院，在马尔默大学。物理样机实验室运行时间最长的Arduino的平台，Arduino的一直是时尚，身体和技术，灯光装置，在学士和硕士水平的相互作用的方案课程的积极参与为基础的大学课程，自2005年以来的一些。

直到2008年被引入时尚，身体和技术课程的学生在一个老式“硬”的方式，其中较早的重点一个直的技术转移到时尚和可穿戴计算的情况下物理样机。然而，在2008年的地方采取的步骤，以技术落实到时尚的背景下，在一个“软”的方式夏天。

在此之前，从经验的教师在物理原型实验室已与时尚和可穿戴计算的利息的学生也很难转化为原型开发的标准的物理原型知识为wearables。寻找合适的材料，在此基础我们到外地的新方法时，我们很快就意识到，现有的资料相当有限，大部分材料“艺术和工艺”的字符。

我们在DIY运动的强烈信徒，仍然觉得有很多从“艺术和工艺”的物质在那里学习，但在大学水平的过程中，应能提供一个更复杂的方法。我们当时发现的问题，只有小部分可用的材料显示如何接口与微控制器或微控制原型平台。

这两个原因都涉及到Arduino理念。 Arduino是相当独特的，因为它是开放的软件和硬件。 Arduino的目的是创造一个设计师的原型开发平台，能够实现自己的想法。 Arduino的设计的开放性意味着任何人都免费的硬件和软件的修改，甚至生产和销售自己的议会。这意味着，Arduino的价格是在一个非常实惠的的水平，因为任何人都可以在制造业竞争反过来。价格反过来也为大量用户的主要原因之一，正是这些用户Arduino的社会。这是一个社区共用一个原型和股份阿尔杜伊诺理念的开放，这意味着有很多共同的爱好信息和帮助被发现。这也意味着Arduino的社会是向前推原型的演变最新，最快速的演员之一。

考虑到这一点的问题从来没有从Arduino的移动距离接近时尚领域和技术，而是如何处理在一个“软方式”使用Arduino领域。发展过程中的大部分都集中在比较人活跃在时尚和可穿戴计算领域的项

目，在大学的教诲，找到柔和的替代正常的“硬”

组件。目标已经使用“硬技术”和“软”的方式实施帮助，可以考虑更接近物理样机从纺织背景的人自然技术的理解在使用相同的基本原则。

这本书并不完全是为了在时尚和技术，可穿戴计算领域的学术兴趣的人，但也应考虑作为一个与主体的一般利益的人的入门指南。它是零零碎碎的，旨在激发读者的发展后，的集合。

因此，我衷心希望您能喜欢我们的书的其余部分，记住，如果我们能做到这一点，所以你可以。

Tony Olsson

此外的第一个版本

马尔默大学围墙外的低利息的期望写的这本书的第一个版本。然而，一次完成的第一个版本，我们认为这将是一种耻辱，不能分享。

当时我们有最好的办法是做一个简单的网站，后书有下载。在两个月的时间，我们约800万下载数个月后的数量慢慢增加近万以上。对我们来说，这是一个惊人的灵感和单因素决定我们发布一书的汉化版。

在过去的一年中，我们也有工作组作一些补充。梅丽莎，德里克科尔曼，科尔曼和安德烈亚斯Jiras都得到了很大帮助和支持书为我们给我们的最深切的感谢。

第一部分：基础知识

第1章：引言

原型与阿尔杜伊诺

Arduino的原型平台是基于简单的原则，使用输入和输出。输入通常是某种形式的按钮，开关或传感器。按钮通常只对或关闭，但与一个传感器，你可以衡量整个范围的事情，在您的环境。声音，动作，温度和光照都可以通过Arduino的处理，如果你能想到的任何其他你想要测量的物理变量，没准它已经有一个传感器。

以同样的方式，您可以连接大量的投入，可以控制大量的产出。输出可以从光线，运动，或热什么更复杂的输出，如发送短信，甚至打开世界的另一端，一台电视机。在大多数情况下的技术解决方案已经为原型存在，你只需要找到它。由于Arduino的本质上是一个微控制器，连接到它的一切你有电。好处是，几乎一切都可以转换成电信号。例如，当一个人被感动的东西的一个信号发送到大脑对身体的某处有一个感觉。以同样的方式，我们可以从Arduino的电力线，回去吧。如果有什么打破这种电力线，信号发送回的Arduino的大脑告诉它，没有电就行了。

黑客：省钱，了解更多

一旦你已经开始发挥您自己的电子原型，你很快就会意识到，很多电子元件成本的相当数量的钱。出于这个原因，在电子原型感兴趣的人最有一只脚在硬件黑客。硬件黑客也被称为修补和它所描述的商用电器产品，除了刚才看到的“是什么让他们”的活动。修修补补不只是更了解电子的东西如何工作的一个好办法，但它也是一个有效的方式来省钱。可以找到很多的组件，您可能需要为您的项目中通常被视为垃圾。一个老的打印机有可能仍然是功能的电机，一个旧手机有不错的电池和小型振动器和廉价的电动玩具往往金矿。有没有正确或错误的黑客和修修补补的方式，这也意味着有没有官方的说明是如何做到这一点。但是，互联网上的修修补补和Arduino的网站，www.arduino.cc/游乐场，www.makezine.com和www.instructables信息和提示。com 是伟大的资源，让你始。

如何电力工程

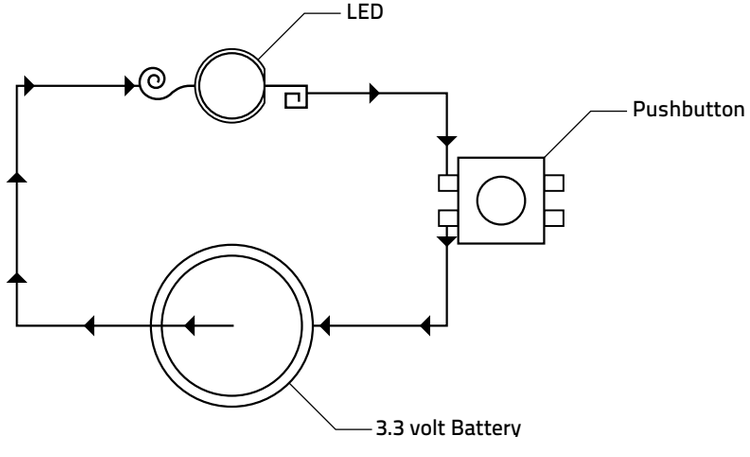
有几件事情是你需要了解的电力，使自己的电子原型的。首先所有的电力

总是需要回到它来自何处，使电路。

在下面的例子中，我们已经与交换机连接的LED电池：电池电量将通过电缆旅行到的一条腿和出其他按钮，并返回到电池的LED。

一个按钮连接与另一个金属小片，当你推动它。如果按钮是推什么都不会发生，但是当你推动它连接的金属板和电池电源，可出差回来，它来自何处。一旦回来电池的LED补光灯。在上面的例子中，有一个3.3V的电池和LED可以处理3.3V的电力。如果我们连接9V电池，LED会烧坏。这是因为电力有不同的电压和安培和阻力传播。试想电力要像水。水的速度将是相同的电压和旅行水的数量将是相同的

安培。比方说，我们的水是通过一个软管，这是一个有用的类比理解电阻传递。所以导致9V电池连接，想推了太多的水通过一个花园太快软管。花园软管如果不能让所有的水通过软管爆裂。这是非常罕见的事情发生爆炸时与Arduino的原型，因为大多数的原型是这样，就很难甚至认为他们是有害的低电流。但仍连接更多的权力来的东西比它可以处理，因为它永远是一个好主意有一个很好的机会，你会打破它。因此，始终遵循建议的权力的限制，为您的组件。你可以找到这些属于你的组件的数据表。数据表中经常可以发



现网上寻找结合所说的“数据表”上写的组件数量。

第二章：硬件

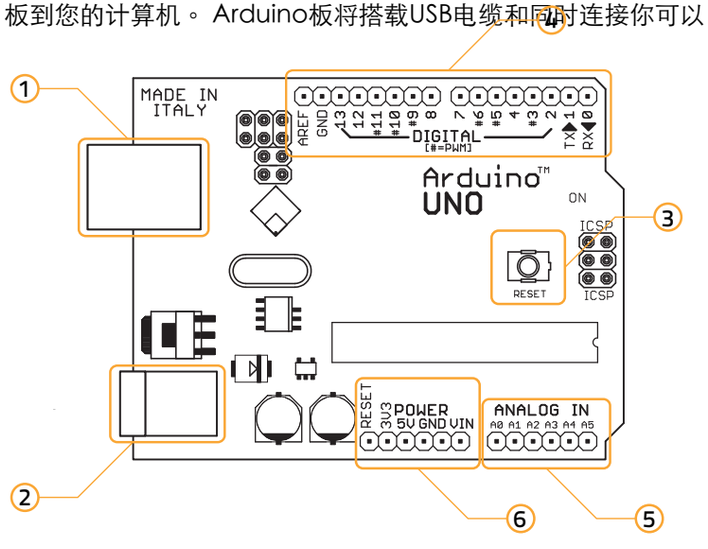
Arduino

Arduino是一个开源的微控制器电路板，用于电子原型。 Arduino的可以接收来自传感器及其周边地区的信息收集的数据，它可以用来控制其他电子元件，如灯，电动机和更多。 Arduino的可有不同的品种。其中最常见的是最新版本的标准Arduino板，看起来像这样：

橙色突出Arduino板，最重要的的部分：
1：USB接口。
2：电源连接器。
3：复位开关。
4：数字引脚。
5：模拟引脚。
6：电源引脚。

用于连接您的Arduino的USB接口（1）

板到您的计算机。 Arduino板将搭载USB电缆和同时连接你可以 上



传代码和沟通，并从你的Arduino板。

用于电源连接器（2）当你不想你的Arduino的USB电缆供电。相反，你可以使用在正常范围为6V至20V的变压器（电源适配器）。

注意：
虽然阿尔杜伊诺板上的电源稳压器，确保从不连接一个大于24V电源。有机会，你将摧毁你的Arduino板。

销售但建议使用7V和12V之间的某个地方，因为使用更高的电压可能会导致。板载稳压器过热。 也可以运行在阿尔杜伊诺电池. 在在

Arduinos比Duemilanove版本，您需要手动开关电源阿尔杜伊诺旧。这是通过开关的塑料的USB接口和电源连接器之间的跳线（3）。如果你想你把电源与USB的Arduino的最接近的USB接口的两个引脚的跳线。如果你想使用外接电源，比最接近的电源连接器两个引脚的跳线。阿尔杜伊诺后Duemilanove开发的版本，自动选择电源。有13个数字通道（4）Arduino板，而这些都可以取决于你如何在你的程序设置输入和输出使用：

模拟引脚（5）只能作为输入，但可以读取一个更大的范围比数字引脚传入的信息：

为了模拟引脚的左边，你会发现电源引脚（6）。从这里，你可以拉3.3V或5V。名为输入电压引脚会给你无论是连接到电源插座。如果您有12V电源插座连接到，你将能够从这个引脚拉相同。在这里你还可以找到两个GND引脚。

复位开关（7）用于复位任何Arduino的方案从头开始。 Arduinos比的Diecimila旧，复位按钮需要把每次尝试上传代码。

注意：
GND是接地短路。

就在这本书的例子中，我们选择了使用标准的Arduino板，因为在我们看来，这是最好的原型板。当你接近完成您的原型，它可以被迁移到其他较小的板，以节省空间。这些议会的工作，以同样的方式和标准的Arduino板编写的任何程序将Arduinos所有其他类型的工作。

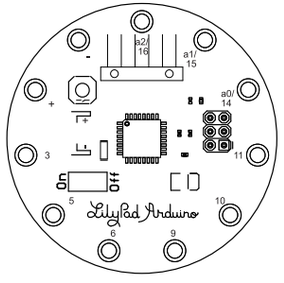
以下是两个可用的Arduino的板的其他例子。

LilyPad

LilyPad Arduino是一个微控制器板设计wearables和E -纺织品。它可以缝制面料和同样安装用导电线的电源，传感器和驱动器。该委员会是根据ATmega328V芯片（低功耗版本的ATmega328）。LilyPad Arduino的设计和开发莉娅Buechley和SparkFun电子。 +的Lilypad接受2.7V - 5.5V。

如果超过5.5V，可以打破Lilypad。一定要加倍小心附加电池时，这可能在开始different

包装上指定的电压。连接它之前，先测量你的电池的输出，这是一个很好的习惯。



注意：
ATMEGA168是由Atmel公司生产的电子集成电路微控制器。它已基本爱特梅尔AVR指令集。 Atmel和ATMEGA芯片家族的更多信息，请访问Atmel的网站：
www.atmel.com

注意：
标准Arduino板的硬件或其他阿尔杜伊诺板的信息，了解的更多信息，请访问Arduino的网站：
www.arduino.cc

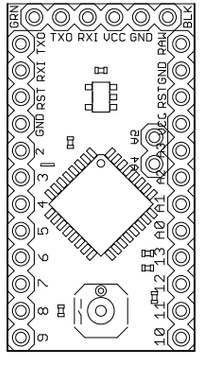
Arduino 迷你

Arduino的迷你是基于ATmega168的一个小的微控制器电路板，上面包板使用，当空间有限。它有14个数字输入/输出引脚（其中有6个可用于为PWM输出），8个模拟输入，和一个16 MHz的晶体振荡器。它可以通过编程与迷你USB适配器或其他USB或RS232 TTL串行适配器（www.arduino.cc）。如果您从板的男性引脚可缝一块面料使用导电线程。

软原型的基本电子元件

•导电螺纹

这种类型的线程看起来像一个正常的灰色线程，但它具有潜在的电源和信号传输的电流。最常见的导电线程金属镀。这意味着一个金属层已超过纺织线程，这一层导电。导电线程可以用来代替正常的电缆，是非常合适的“软”原型。



在不同厚度和具有不同电阻的导电线程。在线程的抵抗力降低，在您使用的电源电压的压力。这意味着低电流可以通过线程。线程的阻力，可以计算出每米或用多米的测量。记住简单的原则，即一个较长的线程有更大的阻力。随着线程和导电织物的表面大小也影响了阻力。一个较大的导电表面会产生较低的电阻。这意味着，一个较厚的线程相同长度小于更薄的阻力。如果一个导电线程的电阻过高你的电路，你的设计需要一个较长的的路径，也可以使用导电布带，而不是一个好主意，或结合多个导电线程进入一个较厚的程。

当您使用您的缝纫机导电线程，它总是最容易使用它作为下线程 - 这就是所谓的底线。一些导电线程也可以在你的机器面线，但确保您设置的紧张权。如果您使用的导电线不适合缝纫机/绣花机，它仍然会工作，但往往会打破你的线程，结果将一些不完善的地方和更多的工作来完成比如果线程适合机器。

有些导电线程有一个金属丝包裹周围的温暖电阻纺织线程，或者是纯粹的金属。这些线可直接焊接到金属表面，如金属珠或金属拉链。这种类型的连接一般是很多更快，更可靠比缝连接。

•电阻

电阻器是一个电子元件，旨在反对一个

目前的电力生产比例目前其终端之间的电压降。电阻测量欧姆，也可以作为符号Ω。最常见的阻力计算乘数：

基洛欧姆（kΩ）的，这是作为一个1000Ω相同。

兆欧姆（MΩ），这是百万Ω相同。

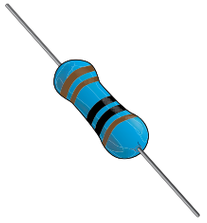
所有电阻都是彩色编码。电阻可以有4，5或6色的乐队，正是这些乐队，告诉你的阻力。

如果你有一个6色波段电阻比前三个波段都遵循这种颜色位数计划的数字：

黑色0	绿色5
棕1	蓝6
红2	紫7
橙3	灰色8
黄4	白色9

如果三个频段是棕色的，红色和蓝色，这将转化为126。第四个频段的乘数。您乘这个乐队的前三位，那么你得到你的电阻的电阻。第四波段如下图所示的配色方案：

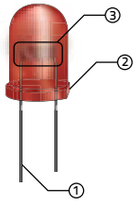
银0.01	橙色1K
黄金0.1	黄色10K
黑色1	绿色100K
棕色10	蓝色1M
红100	紫色10M



第五带电阻的公差和第六波段的温度系数。很难学习心脏电阻的计算，所以建议您使用在线电阻计算器一定。如果你谷歌“的电阻计算器”中，你会发现他们中的很多。另一种 - 快 - 这样就可以读取电阻器的电阻是通过用万用表测量它。

注意：
第三个数字是不使用四个波段电阻。四频段的电阻只使用两位数字，第三个波段是乘数和第四是宽容。

•LED
一个发光二极管（LED）是一种半导体二极管，当电流是在前进方向的LED应用亮起。有三种不同的方式告诉了LED的前进方向。首先是看它的腿。将会有一个较长。这个长的腿，将连接到一个地方的权力来自腿短，需要连接到地面的动力源（1）。第二种方式告诉了LED方向是看形状的塑料泡沫（2）。下缘的泡沫应该有一方是平或凹陷。



腿侧平到地面，另外一条腿去你的权力。第三种方式是举行了LED一盏灯，一看里面。将有两件，一小和一大（3）。小块的腿是应连接到接地应连接到电源和腿部的大块。

最常见的发光二极管在3.3V和20mA附近的供电。请注意，阿尔杜伊诺将始终提供从数字引脚的5V。因此，我们使用一个220Ω的电阻，以降低功率，使我们不烧的LED。要确定电源LED需要的是什么，为您在看数据表指示灯。这种信息往往是与您购买的电子元器件。如果您需要计算的阻力，使用你的LED，我建议您使用线上

的电阻计算器。（www.dannyg.com/examples/res2/resistor.htm）。

•导电布

导电布，导电能力。他们通常是高导电金属和轻薄面料的组合。他们经常被用来作为屏蔽材料。

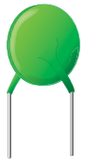
•倾斜传感器

如果一个对象是倾斜的一方或一个倾斜传感器可以检测出

另一个。最便宜的那种倾斜传感器也可用于检测运动。这种倾斜传感器的缺点是，他们的工作作为一个按钮，所以他们不能被用来告诉一个对象是在哪个方向倾斜或多少，只是对象是倾斜的。倾斜传感器内部有一个金属外壳内的一个小金属球。当球触及的金属外壳，完成了电路的两侧，我们可以读到从Arduino板的倾斜运动。

•LDR传感器

也被称为光敏电阻LDR（光敏电阻）传感器。一条LDR是由高电阻半导体。 LDR是与正常电阻固定值和LDR的阻力是在其附近的异常正常电阻类似。这是在一条LDR设置光的确切数额难以确定，但他们不够好，用于确定在更广泛的意义上 - 如果它的或深或浅。



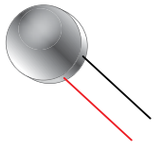
•NTC传感器

也被称为一个热敏电阻的NTC（负温度系数）传感器。热敏电阻是一种电阻型，根据温度改变其电阻。这很难与热敏电阻的确切温度测量，但他们使我们能够确定，如果事情是冷或热。

•汽车

如果你想要的东西移动，你可能需要某种形式的电机。电机是多还是少一个执行机构，变成运动的电力。当你与隐藏的耐磨原型工作，电机可以成为一个问题，如果您需要大量武力。大多数电机遵循简单的原则“的力量越大，电机越大”。然而许多小型马达，是我们的应用相关的一些创造力，他们可以很好地集成到您的原型。

有三种主要类型的电机：直流电机，伺服电机和步进电机。我们并没有包括在这本书中的任何步进电机的例子。虽然步进电机完整的旋转和移动中的步骤，他们不适合可穿戴技术的实施，因为它们是沉重的，其规模和相当笨重的形状。如果你正在考虑使用一个电机，建议您找到一个DC电机或伺服电机，适合您的原型。



•电线

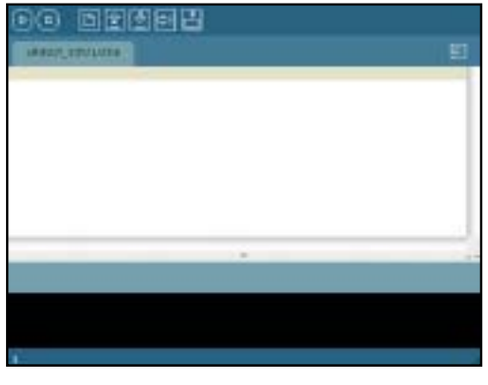
电线导电金属薄放在一个塑料外壳内的线程，并在大范围的大小和颜色来。它的良好的原型颜色代码的行为，您的电线，让你使用相同的颜色一致。常见的颜色使用的是压水堆=红色和引脚GND=黑色，可以有其他任何颜色。

注意：
当我们正在使用的电线，它是良好的原型行为卷曲暴露的金属核心，使其更容易地缝到位。

第三章：软件

用来写你的Arduino的方案软件被称为Arduino的IDE（集成开发环境）。阿尔杜伊诺IDE是基于另一个开源编程语言处理，这是编程图像，动画和计算机交互使用。阿尔杜伊诺IDE看起来非常相似，处理IDE：

以上是加工IDE和下面你可以看到Arduino的IDE。



处理语言形式Arduino的编码语言的模具。 Arduino语言是基于容易使用的命令。每次按下上传按钮在Arduino的IDE，它转换成C代码程序，以便Arduino板能理解你的程序。语言是建立这种方式，首次程序员使用C语言编程是十分困难的。

安装软件

该软件下的“下载”www.arduino.cc的网站上可以找到。一旦你找到了下载页面，选择正确的版本，为您的操作系统。当您下载软件，解压缩和Arduino的文件夹放在桌面上。

这些指南仅涵盖阿尔杜伊诺UNO板。如果你有一个旧版本的董事会在Arduino的网站看看如何安装它的更多信息。

快人一步的指导：
1。打开开始菜单，右在电脑上点击并选择“属性”。
2。点击设备管理器。
3。查找和右键单击Arduino的UNO（COMxx）在清单中。
4。选择“更新驱动程序软件“。
5。选择“浏览我的司机，电脑软件“和重定向到/阿尔杜伊诺/驱动器。

•对于XP的用户

完成上述所有步骤后，您的USB电缆，并把它连接到您的Arduino板和您的计算机上的USB端口。 Windows将尝试安装驱动程序，但将失败。不要担心这是正常的的。

一旦安装过程中失败了，打开你的“开始”菜单，右键单击计算机并选择属性。这将打开系统窗口，并在此窗口左上角，你会发现选项设备管理器。点击“设备管理器”，它会打开另一个窗口，在您的计算机设备清单。在这里，导航COM和LPT，你应该找到一个名为“Arduino的UNO（COMxx）”的设备。右键单击它并选择“选项”更新驱动程序软件“。这将打开一个安装窗口，您应选择“选项”浏览“我的电脑驱动软件。

最后一步是直接安装Arduino的文件夹内，您的计算机上的驱动程序文件夹。设为舒尔这是您选择，而不是FTDI的USB驱动程序文件夹，这也是驱动程序文件夹内的驱动程序文件夹。按下一步，Windows将完成安装。

有时Windows促进一个可以忽略的警告消息，只是选择的选项，反正安装驱动程序。

•对于OSX的用户

装载磁盘映像（Arduino的00xx.dmg），会出现一个窗口。在此窗口中，你会看到两个图标之一thet说阿尔杜伊诺说应用程序的一个文件夹。只要Arduino的图标拖放到窗口内的“应用程序”文件夹图标和你做。无需驱动程序需要安装的UNO板。安装完成后，你应该能够找到您的应用程序文件夹内的Arduino软件。

快人一步的指导：
1。安装磁盘映像。
2。 Arduino的图标拖放到应用程序文件夹窗口内的图标。
3。你做。

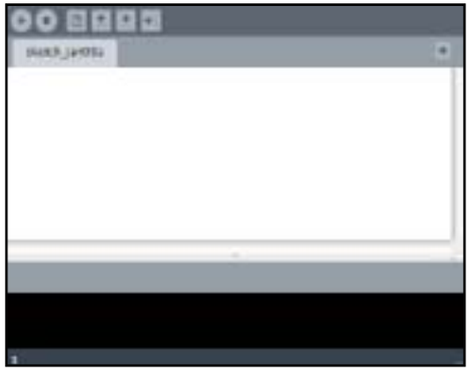
•对于Linux的用户
在Linux上安装Arduino的安装过程中可能会从分发到不同的分布。因此，如何安装您的发行的最新信息，请参阅的Arduino的网页和/或您的分布首页。

第四章：使用IDE

IDE将按钮:

注意：
IDE只能使一个逻辑检查，并不能确定，如果该程序对应到你想要的程序来完成。一旦你开始编译你不能阻止汇编停止按钮。

IDE包括两个大的空间，一白一黑。白色的空间，在那里你会写你的程序。请注意，你写在这里将作为代码认为，如果你不“的评论”。要了解如何隐藏在你的代码转第80页的文字。



黑色空间，在这里您将收到错误和确认消息。上面白色的空间，你会发现在IDE按钮。使用这些按钮，您可以控制在IDE中的大部分行动。

第一个是“编译”按钮。这个按钮检查你的程序，看看是否有在您的代码中的任何逻辑错误。

第二个按钮是“停止”按钮。此按钮是用来关闭串行显示器。，因为它需要编译器将采取尽可能多的时间，但不用担心，编译通常只需要几秒钟，取决于你的程序是多么大。

第三个按钮是“新草图”按钮。您打开的所有程序或写在Arduino的IDE称为草图。新草图按钮，将打开一个新的草图为您，但会先询问您是否要保存您目前的草图。

第四个按钮是“打开素描”按钮。此按钮opens the sketches 文件夹，在这里你可以选择打开已保存的草图。

第五个按钮“保存素描”按钮。此按钮保存命名的文件夹中的速写本目前草图。

第六个按钮是“上传”按钮即可。此按钮将目前的方案上传到你的Arduino板，假设在你的代码有没有错误。“上传”按钮，将首先尝试编译您的代码，如果发现任何错误，它会停止编译和错误信息会出现在IDE的黑色窗口，告诉你问题是什么，并且IDE将突出显示的代码行，造成问题。

最后一个按钮是串行监视器“按钮。此按钮将打开一个新的窗口您的序列监视器，其中一间酒吧，会出现一个下拉菜单，发送键和一个消息框。要关闭显示器，使用“停止”按钮。在IDE的顶部，你会发现任何其他程序中的下拉菜单。

在文件菜单中，你会发现所有的按钮的功能时，您的速写本文件夹和喜好。在编辑菜单中，你可以找到撤消，重做，剪切，复制，粘贴，全选，查找和查找下一个功能和命令。在草图“菜单中，您可以验证/编译您的代码，停止，导入库中，显示的速写本文件夹，并添加一个新的文件。

两个最重要的工具“菜单上的特点是板和串行端口。董事会选项，你选择你的Arduino板的类型。在串行端口，您可以选择您已连接您的Arduino板的USB端口。最简单的方法来确定哪个端口连接到您的电路板，因为多个端口可以出现在菜单中，拔掉你的板子，然后检查连接的端口。然后，你插入你的板，再重新和新的端口，出现在清单中是你的Arduino板。

上传代码

为了测试，如果你的软件安装，正常打开闪烁的示例代码中找到：文件/速写本/例子/数字/闪烁。

一旦你的代码目前确保你有合适的板型和串行端口，在工具菜单中选择。按下“上传”按钮，如果一切都没有问题，板13脚旁边的LED开始闪烁和一秒钟的延迟。

第二部分：例子

例子

在这里我们将介绍如何创建和使用输入和输出设备，以及如何收集例子Arduino板的接口。这些例子并不以任何方式完成的原型，但应考虑作为鼓舞人心的建设方案和编程技术，可用于时尚和耐磨的原型有用。

本节将简单的例子开始，逐步转向更加复杂的结构和技术。每个示例所需的所有组件的列表，并开始使用组件和提供的材料应当由大多数电子爱好者商店。有些像导电材料

面料和线程可比较难找到。使用互联网来查找您附近的供应商或比较网上商店，找到了最好的价格材料。

第五章：使用数字引脚

正如第2章解释说，Arduino的数字引脚有两种模式，要么他们开或关。这两个国家的Arduino的实际命令是1或0为关闭，这就是为什么我们称他们为数字引脚。通常我们使用的常量高和低，因为这使得阅读相比，使用0和1的代码更容易。请记住，数字管脚总是给人输出功率为5V和0V低电平时。不要任何连接，直接数字引脚，除非你确信它可以处理5V。

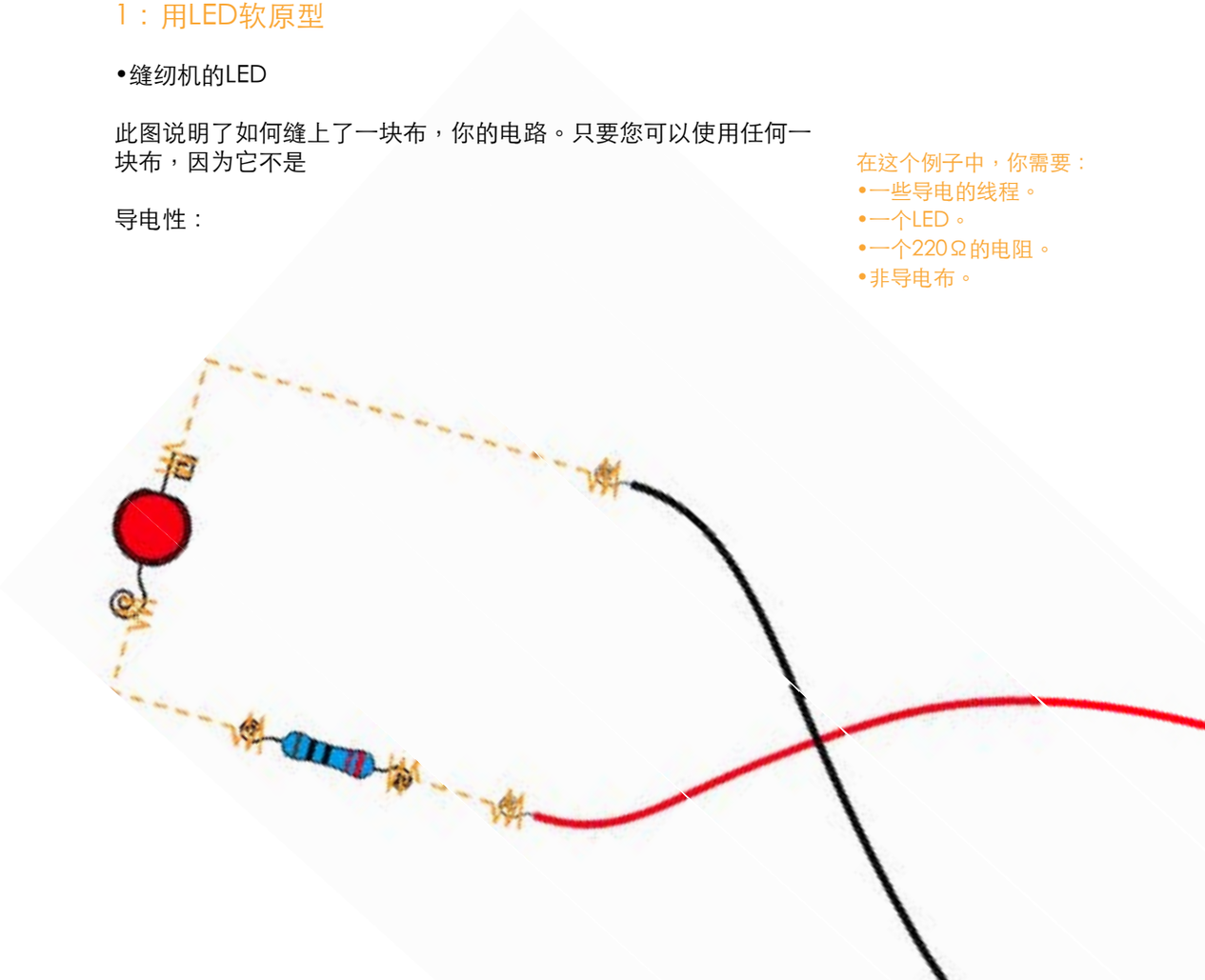
1：用LED软原型

•缝纫机的LED

此图说明了如何缝上了一块布，你的电路。只要您可以使用任何一块布，因为它不是

导电性：

在这个例子中，你需要：
•一些导电的线程。
•一个LED。
•一个220Ω的电阻。
•非导电布。



在这个例子中，我们取得了圈腿的电阻，和广场上的LED的腿和两个圆。这不仅是对化妆品的原因，它的实用 - 它使得更容易地缝到位的组件和不同的形状也跟踪LED的长期和短期的腿。对于电压和接地连接，我们使用的电缆，用导电线的地方缝。这是一个很好的技术时使用的测试原型。对于最终的原型是

建议你到面料缝制Arduino板，用导电线直连接到您的组件或使用导电布条。根据您的设计，从你的Arduino的导电线的路径，你的LED，可这么久，你不需要220Ω电路中的电阻。测量电阻导电线，看它是否足够高，以取代电阻的路径。

•闪烁的LED的编码

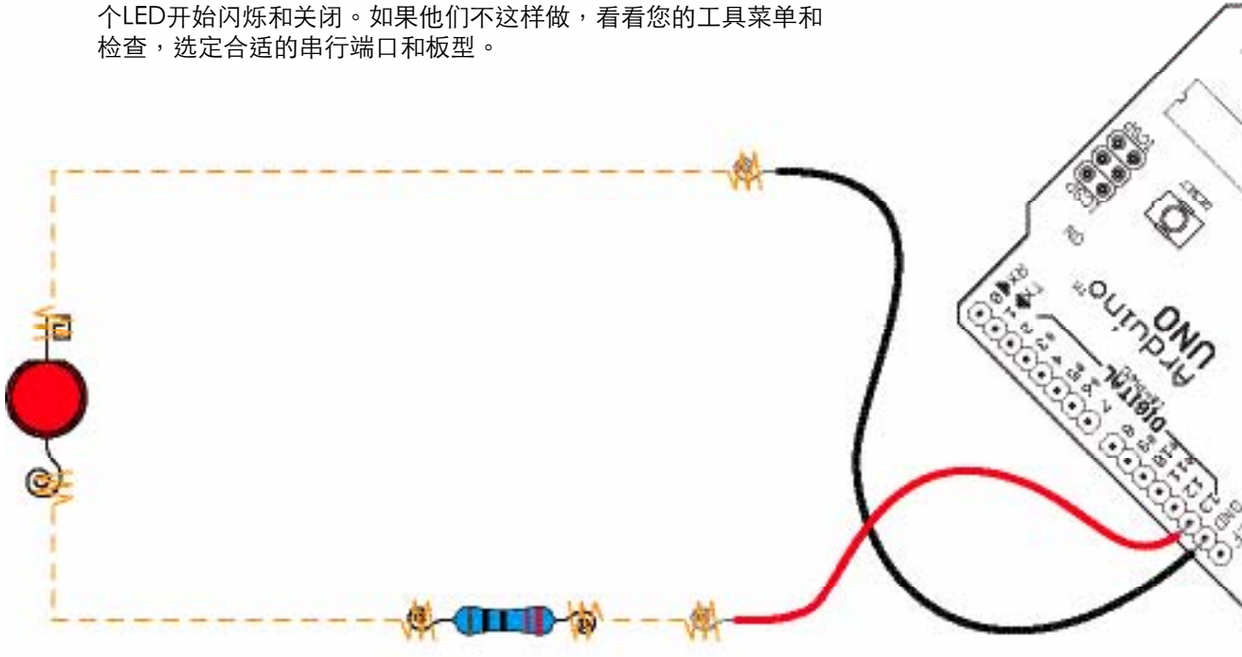
在连接你的Arduino的，您可以测试您的代码工作，而且也没有损害到您的Arduino板。 Arduino的一个小的内置LED数字13脚，它与下面的代码示例，它应该开始闪烁如此：

```
int ledPin = 13;
 /*整型变量的LED连接数字引脚13*/

void setup(){
 pinMode(ledPin,OUTPUT);
 /* 设置输出的ledPin */
}

void loop(){
 digitalWrite(ledPin,HIGH);
 /* 打开的ledPin */
 delay(1000);
 /* 等待一秒钟*/
 digitalWrite(ledPin,LOW);
 /*打开ledPin关闭*/
 delay(1000);
 /* 等待一秒钟*/
}
```


此程序将打开一个重新开始的第二个前一秒钟然后熄灭的LED。一旦你已经写在你的Arduino的IDE的程序，按确认按钮和消息：“完成编制”应该出现在黑色的窗口。

如果没有编译错误确保您在工具菜单中选择正确的串行端口和板型。

接下来的步骤是按“上传”按钮上传到您的Arduino板的代码。有两个LED阿尔杜伊诺：RX和TX。每次尝试上传程序到Arduino这两个LED开始闪烁和关闭。如果他们不这样做，看看您的工具菜单和检查，选定合适的串行端口和板型。



如果有上传没有错误，程序存储在阿尔杜伊诺的内存，将呆在那里，直到您更换一个新的5秒后程序将启动。如果旁边的LED板13针开始闪烁，它是安全的假设，该方案是正确上传您的Arduino板，没有什么错。如果你遇到问题时，此LED闪烁程序很好用制作原型，并希望从您的硬件故障排除可能出现的问题清单。

现在它是安全的连接到您的软性电路阿尔杜伊诺。在下面的绘图中，我们已经用红色数字引脚13和黑色的电缆，电缆，可追溯到上阿尔杜伊诺到GND端口。

这是你常用的颜色代码电缆与电子产品工作时如何。红色是用来标记的电缆连接到电源和黑色是用来标记电缆为地下电缆。在这本书中使用下面的例子中，都将遵循相同的配色方案。请注意红色电缆不一定总是连接到数字引脚。在这个例子中，我们是交换的5V和数字引脚，所以我们用红色标志电缆。

•编码一个衰落的LED

在前面的例子中，我们把LED的开启和关闭。正如您可能已经注意到，LED被设置为最大（用一个电阻降低到5V）完成（0V）。如前所述，数字引脚有两种模式，高，低。但数字引脚3，5，6，9，10和11，有一个特殊的功能，所谓的PWM。随着PWM模式中，我们可以转换成255个可能的级别的范围内的5V输出。要了解更多PWM功能的转第97页。在下面的例子中，我们要测试我们如何淡入向上和向下的LED：

```
int ledPin = 5;
 /*连接数字引脚5 */

void setup(){
 pinMode( ledPin , OUTPUT );
 /* 作为一个输出申报ledPin*/
}

void loop(){
 for(int i = 0; i < 255; i++){
 /*只要我是超过255个，增加了一个I*/
 analogWrite( ledPin , i );
 /* 淡出导致我 */
 delay(30);
 /*小暂停，所以我们可以看到每一步*/
}

for(int i = 255; i > 0; i--){
 /* 只要我是大于0，减少了一个我 */
 analogWrite(ledPin,i);
 /* 淡出导致我 */
 delay(30);
 /* 小暂停，所以我们可以看到每一步*/
}
}
```


在上面的例子中，我们使用的命令analogWrite（），这是对LED的PWM命令。我们还可以使用两个for循环，第一个从0开始计数到255。对于每一个循环，它使我们增加了在LED上的电压。，而不是直来直去从0V到5V，从0V到5V255步，这会导致LED褪色。第二环则相反，它会从5V到0V在255步。 30毫秒的延迟，给我们的时间，以通知的步骤。

现在你可以连接的LED，因为我们没有在前面的例子中，代码上传到您的Arduino板，并享受褪色。您应该能够看到的LED褪色向上和向下。

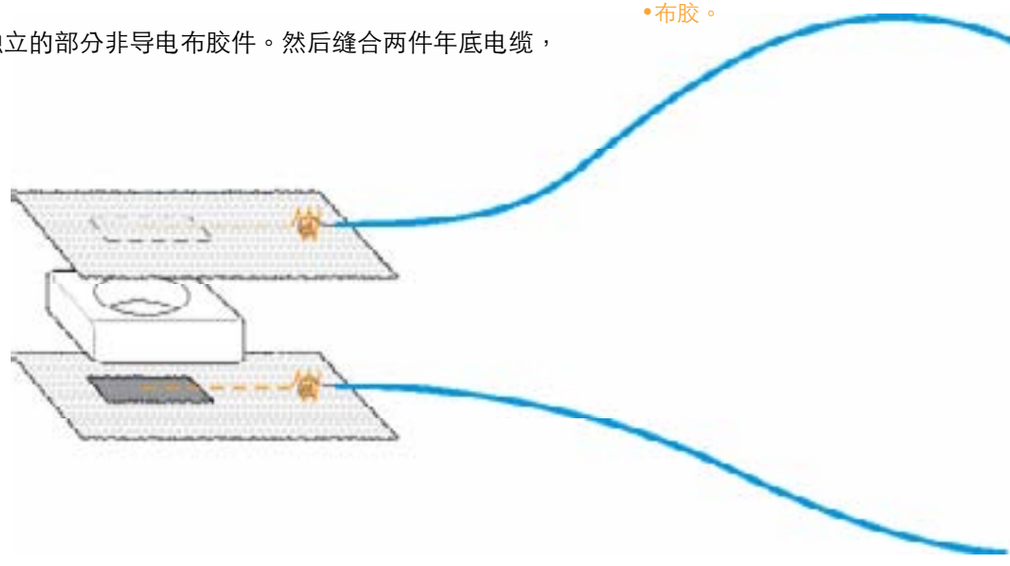
2：软按钮

有不同的方式使自己的软按钮。他们都是基于同样的原则，创造一个电路由Arduino的一个突破点在电路的某处。一个突破点是，我们可以重新连接电路，因此我们可以判断，如果是积极的（推）或不软按钮。

为了使这种软按钮，开始切割出两个小

导电布和两个独立的部分非导电布胶件。然后缝合两件年底电缆，导电布缝连接。

在这个例子中，你需要：
•导电织物。
•非导电布。
•一体式的泡沫。
•导电线程。
•电缆。
•布胶。



就拿一块泡沫，穿过一个洞。如果你没有任何泡沫，你可以使用一个正常的面料层的情侣。切孔，胶导电布块，每个泡沫的一面：

注意：
我们在设置按钮打开5V。每当我们推动我们的权力重定向到GND，如果我们读到的按钮引脚的状态，它会说低，在那一刻，因为没有引脚的电源按钮。

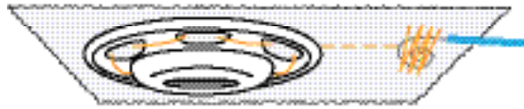
当您准备好按钮，我们可以开始使用程序：

```
int myButton = 4;
 /* 申报为myButton的Arduino的数字4*/
int ledPin = 13;
 /*声明作为ledPin Arduino的数字13*/

void setup(){
 pinMode(ledPin,OUTPUT);
 /* 作为一个输出设置ledPin*/
 pinMode(myButton,INPUT);
 /* 设置为输入myButton的*/
 digitalWrite(myButton,HIGH);
 /* 激活内部电阻引脚4*/
}

void loop(){
 if(digitalRead(myButton) == LOW ){
 /*检查如果按下按钮*/
 digitalWrite(ledPin,HIGH);
 /*如果按下按钮，灯LED*/
 }else{
 digitalWrite(ledPin,LOW);
 /*如果按钮是不推，关闭LED*/
}
}
```

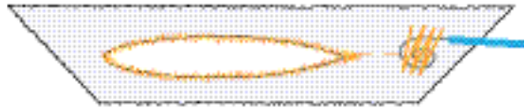
现在我们的程序上传到我们的Arduino板和连接按钮。一端连接到数字引脚4，另一种是连接到GND端口。上述程序将检查如果推按钮，点亮旁边的LED数字引脚13。如果按钮是不推LED将保持关闭。

3：隐藏的按钮

以下是两个简单的方法，使隐藏的按钮，这是适当的，当你需要隐藏在服装中的一个不显眼的输入。您可以使用相同的代码，上述尝试出来。第一个是两个缝上金属管理单元，可以在任何缝纫店发现按钮。你缝的地方使用导电线程按钮，并在最后连接两条电缆。



最后一个隐藏的按钮，像一个正常的按钮。诀窍是，再次使用导电线程缝到位的按钮，然后使用相同的线程按钮周围孔缝。大多数现代的缝纫机有一个钮扣孔的预编程功能，将工作完美。只加载你的机器用导电线和切孔织物。这个例子的工作方式相同，与正常手缝上的按钮或牛仔裤按钮。

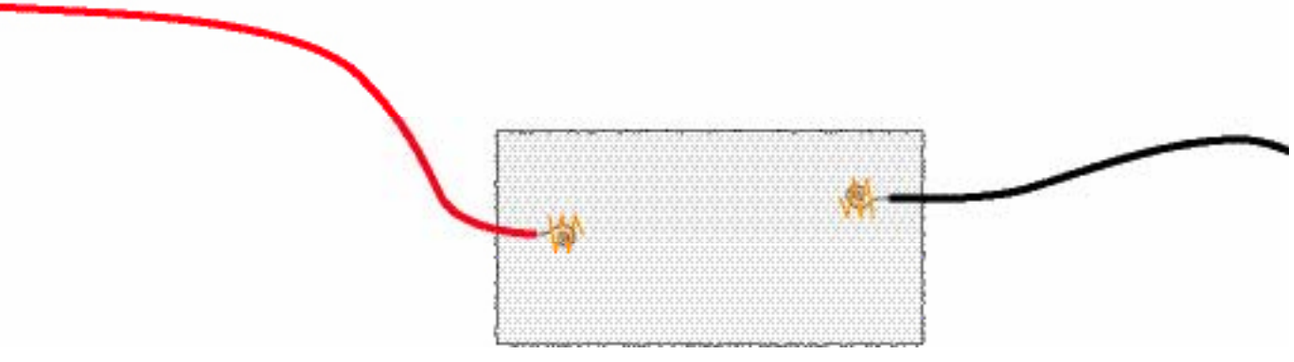


在这些例子中，我们放在最后的电缆，使其快速测试更容易。对于最后定稿的原型缝用导电线回来的路上你的Arduino的每个按钮。

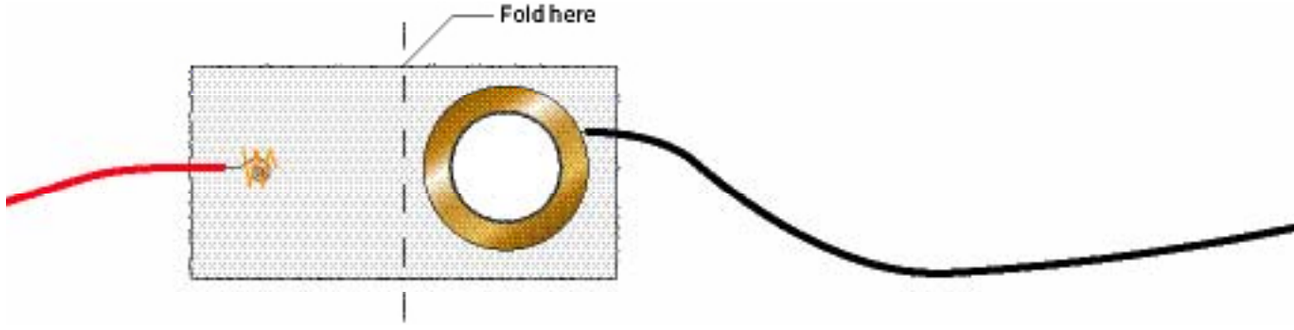
在这个例子中，你需要：
- 压电扬声器。
- 非导电布。
- 导电线程。
- 两线。

4：声音

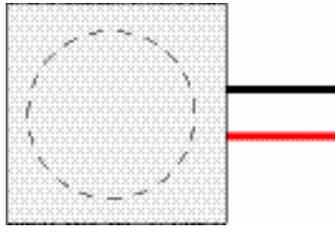
与Arduino的声音，最便宜的方式是使用压电扬声器。压电材料产生一个电场，在应用机械应力的能力。压电扬声器由两个金属板和电力应用于压电扬声器，它会导致金属板，以吸引和排斥，这反过来又会产生声音产生快速振动。压电扬声器在大范围的形状和大小。在这个例子中，我们只打算使用膜的压电扬声器。你可以购买压电扬声器膜中的大多数以及库存的电子商店，或可以打破一个完整的压电扬声器的塑料外壳。如果您选择击破，实际膜扬声器除了小心不要弄弯。一旦你的膜，开始准备了一块布：



连接方式的三分之一织物的每一侧的两条电缆。针没有覆盖塑料用导电线电缆，然后连接电缆的剩余部分，以正常的线程结构。不要使用导电的线程时，你撩到织物上的电缆，因为这将创建一个短路，一旦我们把压电扬声器膜的休息。只是正常线程拆线举行到位的电缆。放在一边的元素和折叠对方在它和缝合一切正常线程。膜周围的十字绣是明智的，因为它需要相当舒适电缆连接的膜两侧。



现在我们有一个压电扬声器，是一个比一个正常柔和，我们可以得到一些方案开始产生声音。为了使产生的振动，声脉冲功率的压电。为了使这些脉冲速度不够快，产生的振动，因为这暂停不够快，我们不能正常使用的delay（）。相反，我们将使用delayMicroseconds（）。在下面的



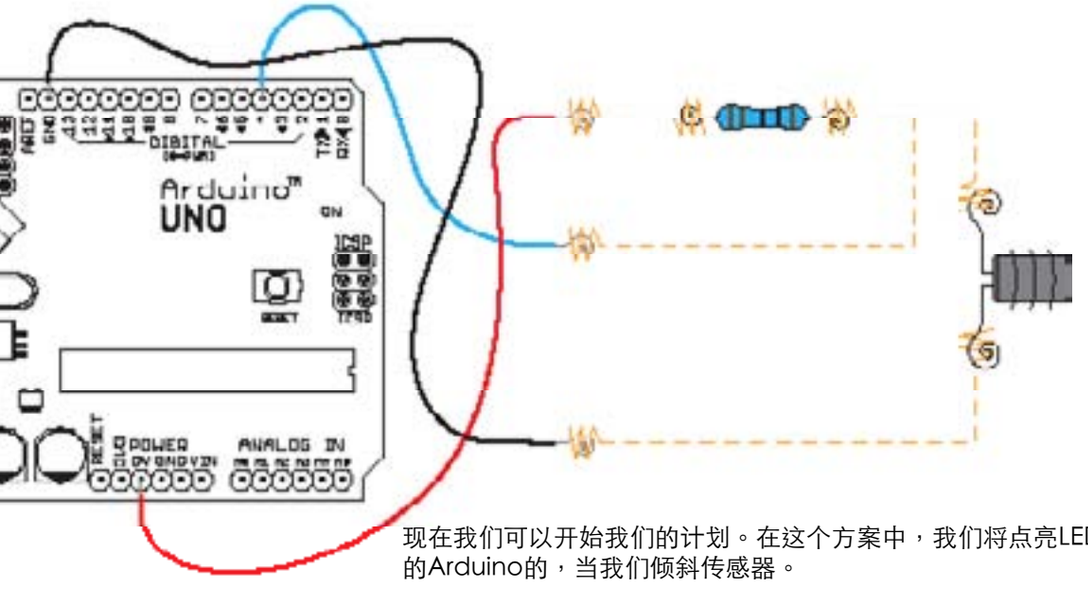
例如，我们产生震动，创造了语气：

```
int piezoPin = 9;
 /* 插件压电引脚9 */

void setup(){
 pinMode( piezoPin,OUTPUT);
 /* 申报压电引脚作为输出*/
}

void loop(){
 digitalWrite(piezoPin,HIGH);
 delayMicroseconds(1136);
 /*这里的时间会确定基调*/
 digitalWrite(piezoPin,LOW);
 delayMicroseconds(1136);
 /*这里的时间会确定基调*/
}
```

在上面的例子中，我们把5V的压电和关闭，在我们之间设置一个1136微秒的停顿。这个暂停是什么决定产生什么音。例如，如果我们1911年微秒暂停，我们反而得到了基调C.

5：倾斜传感器

通过附加的倾斜传感器和一个1kΩ电阻到一块布开始。使用正常的线程，缝在倾角传感器中的几个循环，以保持它在地方。针三根电缆的面料采用导电线程，以下连接如下图所示：



现在我们可以开始我们的计划。在这个方案中，我们将点亮LED板的Arduino的，当我们倾斜传感器。

在这个例子中，你需要：
- 一个1kΩ的电阻。
- 一个倾斜传感器。
- 非导电布。
- 导电线程。
- 两线。

```
int myTilt = 4;
 /* 声明倾斜传感器引脚 */
int ledPin = 13;
 /* 声明针板上的LED*/
int tiltStatus = 0;
 /* 声明变量来存储倾斜传感器的状态*/

void setup(){
 pinMode(myTilt,INPUT);
 /* 声明作为输入引脚倾斜*/
 pinMode(ledPin,OUTPUT);
 /* 声明LED作为输出脚*/
}

void loop(){
```



```
  tiltStatus = digitalRead(myTilt);
  /* 读取倾斜针和存储值 */
  if(tiltStatus == HIGH){
  /*检查是否倾斜传感器倾斜*/
  digitalWrite(ledPin,HIGH);
  /*如果倾斜传感器倾斜之交的LED*/
  }else{
  digitalWrite(ledPin,LOW);
  /*在所有其他情况下关闭LED*/
 }
}
```

一旦你完成你的代码，把它上传到Arduino板和连接在上页的插图所示的倾斜传感器。现在，当你倾斜的面料旁边的LED Arduino板的13针，应该亮起。由于在第2章第27页上的倾斜传感器节所述，您可以使用这种类型的倾斜传感器来检测运动。这是一个使你得到多少点击率从倾斜传感器在一定的时间估计的问题。比方说，倾斜传感器从关闭的2倍每第二个，然后它是安全的假设上的倾斜传感器放在服装正在。

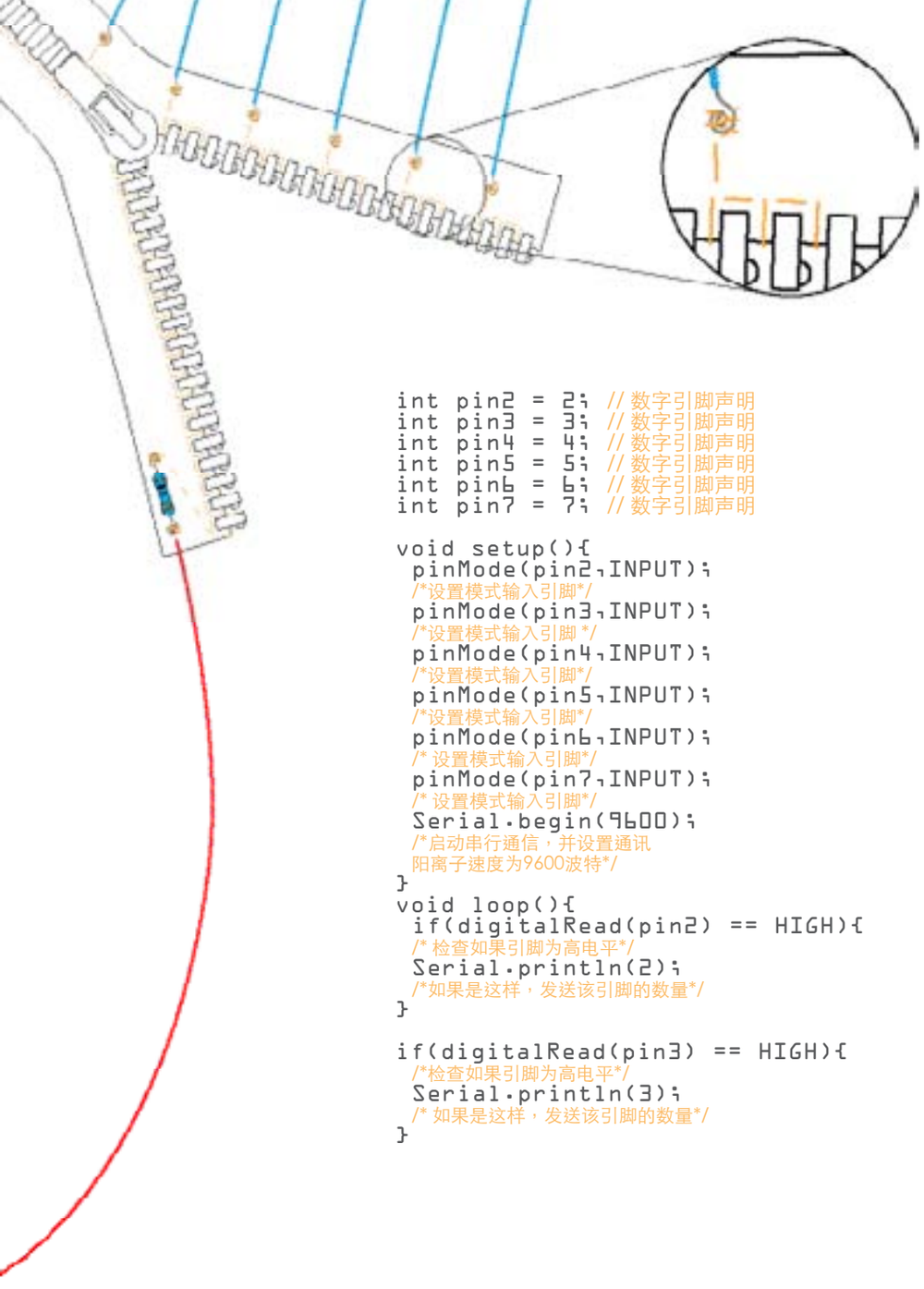
6：数字拉链

有两种方法，我们可以使用普通的金属拉链，使输入传感器。它们之间的区别是我们如何从Arduino的读取。在这一章中，我们将数字化的方式来阅读我们的拉链。使用这个拉链51-54页的模拟拉链相比，优势是数字拉链，将其读数更精确，但缺点是，我们将获得阅读的价值低得多的范围。范围将取决于您使用多少数字引脚。在这个例子中，我们将使用的Arduino的数字引脚6。

在这个例子中，你需要：
- 普通金属拉链。
- 导电线程。
- 10K电阻。

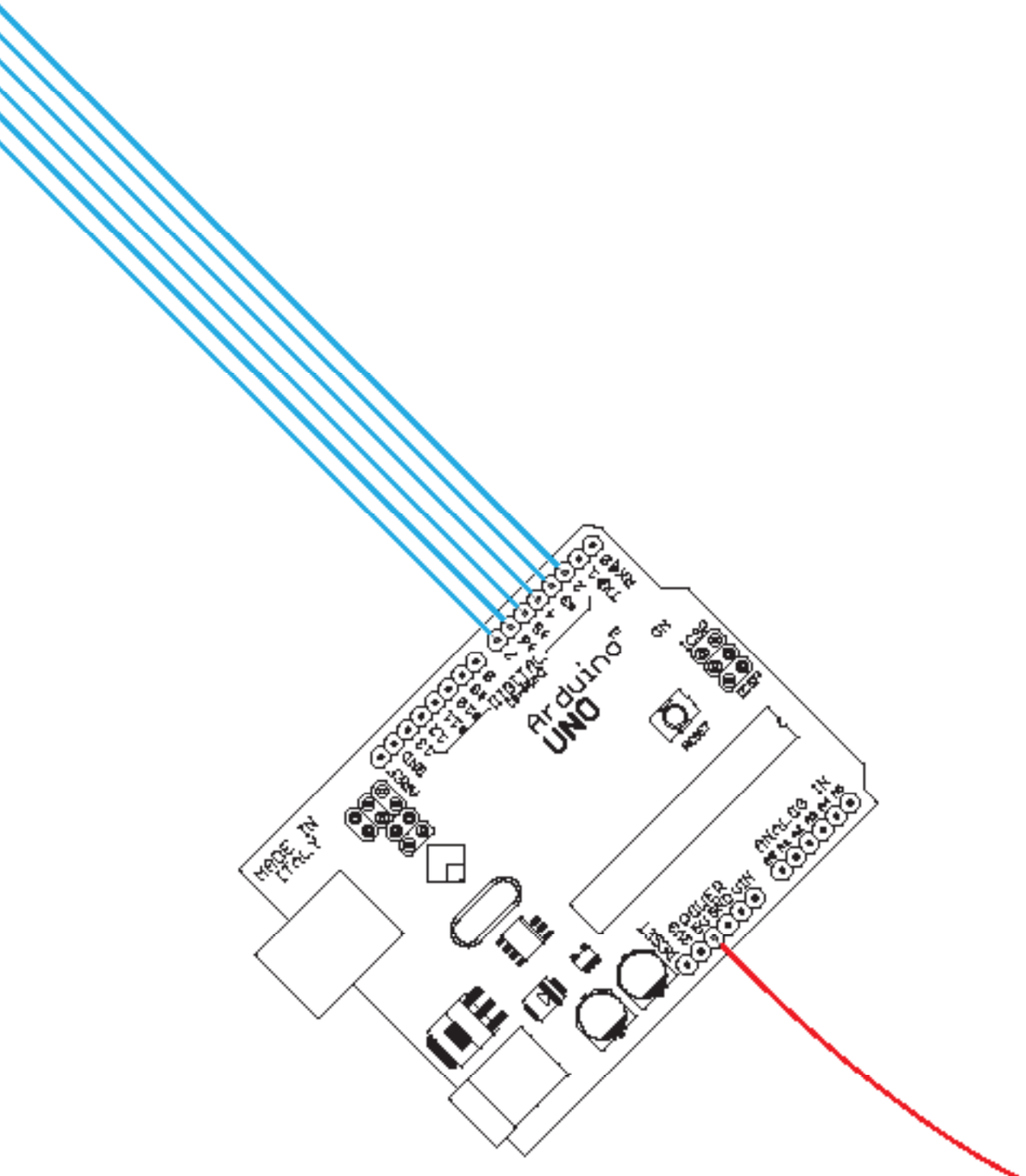
通过附加在最后一个红色的电缆上的一个侧面拉链与一些导电线程开始。红色电缆的另一端，然后在一个10kΩ电阻。请从电阻的另一端，从同一侧的拉链底部到顶部一路导电螺纹连接。确保您在所有拉链的牙齿之间的缝合。在另一侧的拉链，您可以将沿着侧面的6线。一旦所有的六线到位，开始缝制线金属拉链牙齿，使牙齿之间的一个数针，向上，上拉链。

请一定要留下缝线之间的差距从一根导线。六线都不应与导电线连接到另一个。拉链缝合后，我们就可以开始我们的代码：



```
int pin2 = 2; // 数字引脚声明
int pin3 = 3; // 数字引脚声明
int pin4 = 4; // 数字引脚声明
int pin5 = 5; // 数字引脚声明
int pin6 = 6; // 数字引脚声明
int pin7 = 7; // 数字引脚声明

void setup(){
 pinMode(pin2,INPUT);
 /*设置模式输入引脚*/
 pinMode(pin3,INPUT);
 /*设置模式输入引脚 */
 pinMode(pin4,INPUT);
 /*设置模式输入引脚*/
 pinMode(pin5,INPUT);
 /*设置模式输入引脚*/
 pinMode(pin6,INPUT);
 /* 设置模式输入引脚*/
 pinMode(pin7,INPUT);
 /* 设置模式输入引脚*/
 Serial.begin(9600);
 /*启动串行通信，并设置通讯
 阳离子速度为9600波特*/
}
void loop(){
 if(digitalRead(pin2) == HIGH){
 /* 检查如果引脚为高电平*/
 Serial.println(2);
 /*如果是这样，发送该引脚的数量*/
}

if(digitalRead(pin3) == HIGH){
 /*检查如果引脚为高电平*/
 Serial.println(3);
 /* 如果是这样，发送该引脚的数量*/
}

if(digitalRead(pin4) == HIGH){
 /* 检查如果引脚为高电平*/
 Serial.println(4);
 /*如果是这样，发送该引脚的数量*/
}

if( digitalRead(pin5) == HIGH){
 /*检查如果引脚为高电平 */
 Serial.println(5);
 /*如果是这样，发送该引脚的数量*/
}

if( digitalRead(pin6) == HIGH){
 /* 检查如果引脚为高电平*/
 Serial.println(6);
 /*如果是这样，发送该引脚的数量*/
}

if(digitalRead(pin7) == HIGH){
 /* 检查如果引脚为高电平 */
 Serial.println(7);
 /*如果是这样，发送该引脚的数量*/
}

delay(200);
 /* 暂停*/
}
```

代码上传到您的Arduino板和连接的拉链。红色电缆5V引脚上的电路板和其余的导线连接到数字引脚2至7。一旦一切就绪，你可以打开你的串口监视和测试您的拉链。通过向上和向下移动，你应该得到一个数字，从2到7阿尔杜伊诺。我们在这里做的是测量拉链头，因为它位于连接5V数字引脚，从而通过该引脚发送的电流，使价值高，然后，在代码中，转化成数字信息，告诉我们其中一个引脚进行通信。



第6章：使用模拟引脚

在这本书正如前面提到的的，现实世界是不是数字，有时你不能转换成数字读数环境的变化。例如温度从热到冷不改变，它改变了不同的值范围内，通常这些变化随着时间的推移慢慢发生。这就是为什么我们经常使用的模拟传感器读取环境参数，如温度，光照和运动。由此产生的信息存储为连续的数字数据。由于阿尔杜伊诺不能处理像人类一样的信息，我们需要翻译的模拟信息，使Arduino的可以理解。

模拟传感器可以从环境转换成0V和5V之间的电压值的数据。这些值高和低，数字引脚使用不同。数字引脚的高，低分别为5V和0V，别无其他。但可以在模拟引脚之间的任何值的意义。

最大和最小值之间的分辨率从一个不同

微处理器的未来。 Arduino的只能分辨1024级在0V至5V范围。

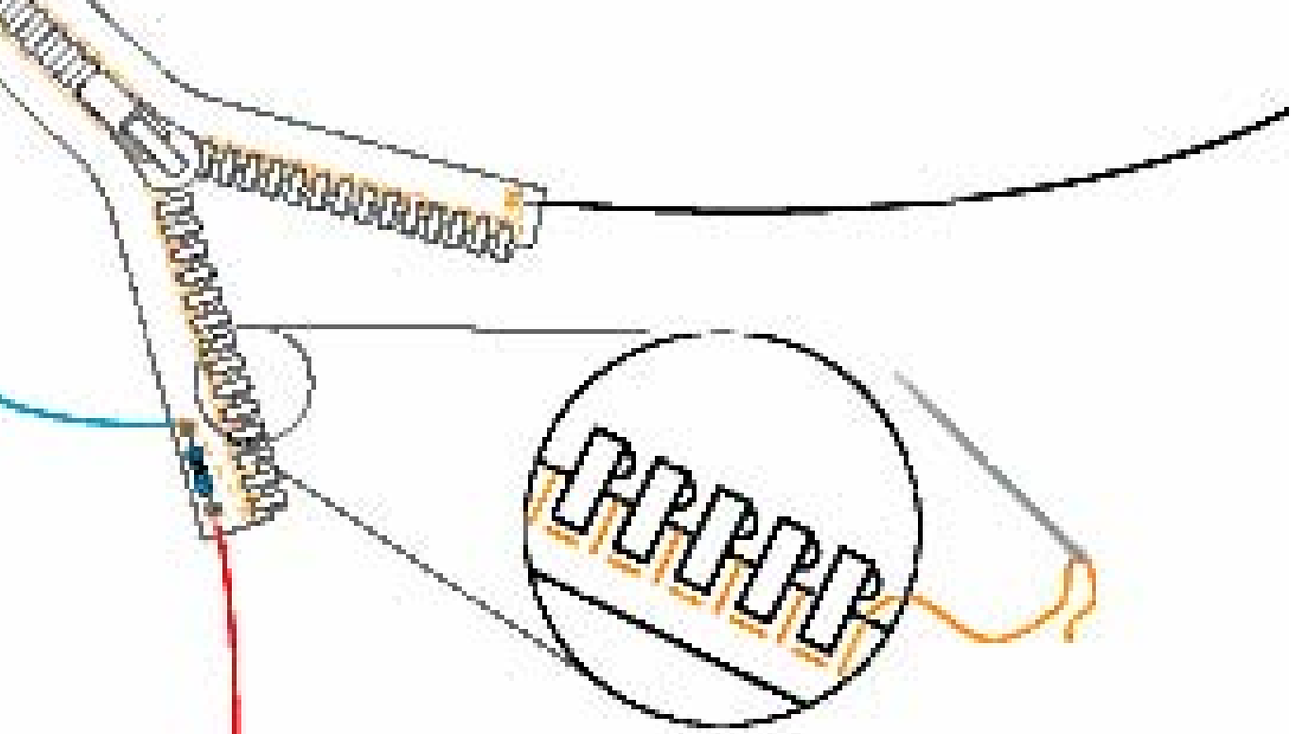
注意：
当我们这样做的Arduino的模拟读数从0开始计数，所以你将永远不会超过1023阅读。

1：模拟拉链

模拟拉链是在前面的章节中所涵盖的数字不同。数字拉链只能是高或低（0V或5V）。模拟拉链，另一方面，可以给你一个0V和5V之间的值的范围。

为了使模拟拉链，你需要一个正常的拉链，一些导电线和一个10kΩ的电阻。开始缝制放置电阻。然后缝合一端到另一端的电阻和一个蓝色一个红色的线。使用，所以你会记得什么电缆，电缆的颜色。从同一个地方为蓝色电缆开始之间的每一个齿的拉链缝制。当你到达一个侧面拉链年底，在另一侧做相同的，再次回落。

在这个例子中，你需要：
- 一个金属拉链。
- 导电线程。
- 一个10kΩ的电阻。
- 两线。



确保线程是完整的圆形拉链一路，否则将无法正常工作。一旦你已经达到的拉链对方，黑色电缆连接，缝入到位。现在是做拉链，我们可以开始编写我们的代码来读取拉链将给予什么样的价值观。为了能够表现出什么样的价值观的拉链给了我们，我们需要使用串行通信，使Arduino的报告到计算机，并显示我们在串行显示器的值：

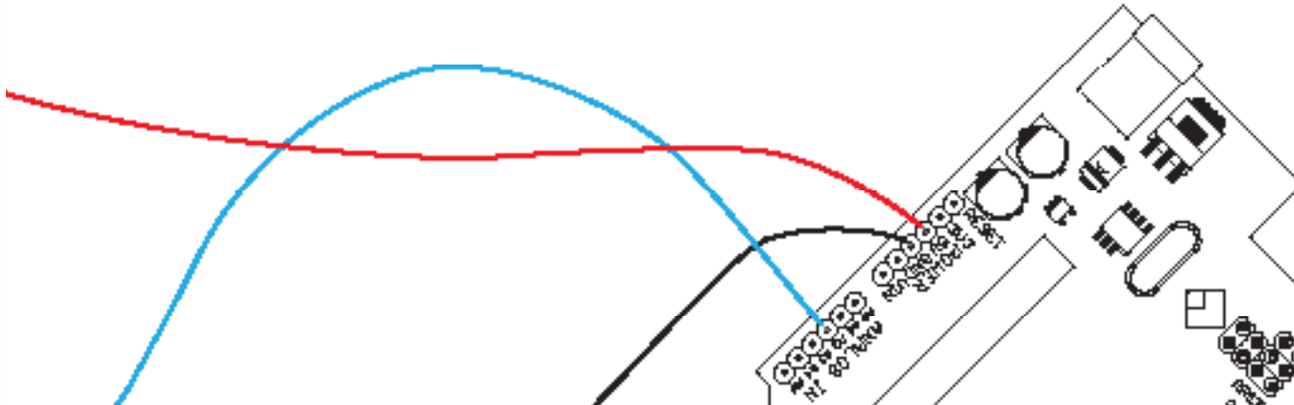
```
int analogPin = 2;
  /*在这里我们声明，我们要使用的模拟输入引脚2
并调用它analogPin*/
int myZipper=0;
  /*在这里，我们已宣布，将作为一个临时变量，所以我们可以存储值，拉链会
给我们一个整数 */
```

```
void setup(){
 Serial.begin(9600);
 /*这将设置你的沟通和设置 沟通，以9600波特（
 每秒9600比特）*/
}

void loop(){
 myZipper = analogRead(analogPin);
 /*在这里我们做一个模拟analogPin阅读和
 保存在变量myZipper的值*/
 Serial.println(myZipper);
 /* 此命令将发送存储在值
 myZipper，并通过串口发送，然后使
 休息一下。*/
 delay(200); //暂停200毫秒
}
```

注意：
我们没有申报为模拟输入引脚的模式，我们将不得不做，如果它是一个数字引脚。

我们在做无效循环（）的第一件事情是一个模拟引脚2上的模拟读。通过这个命令的返回值保存在变量myZipper。我们打印后阅读模拟引脚通过串行端口和任何连接到Arduino（例如，一台电脑）将接收无论在myZipper存储这个值。我们做的最后一件事情是，因为即使Arduino是小的，它仍然比正常的计算机进行通信快得多，因此延误，让您的计算机保持与Arduino的添加在我们的计划延迟。现在，我们可以我们的代码上传到Arduino板。一旦代码被上传，红色电缆插头进入港口说5V，黑色电缆的GND

港口和最后的电缆连接到模拟输入引脚2。如果你在你的程序申报，任何其他超过2 analogPin，你将不得不把蓝色的电缆，在相应的模拟端口：

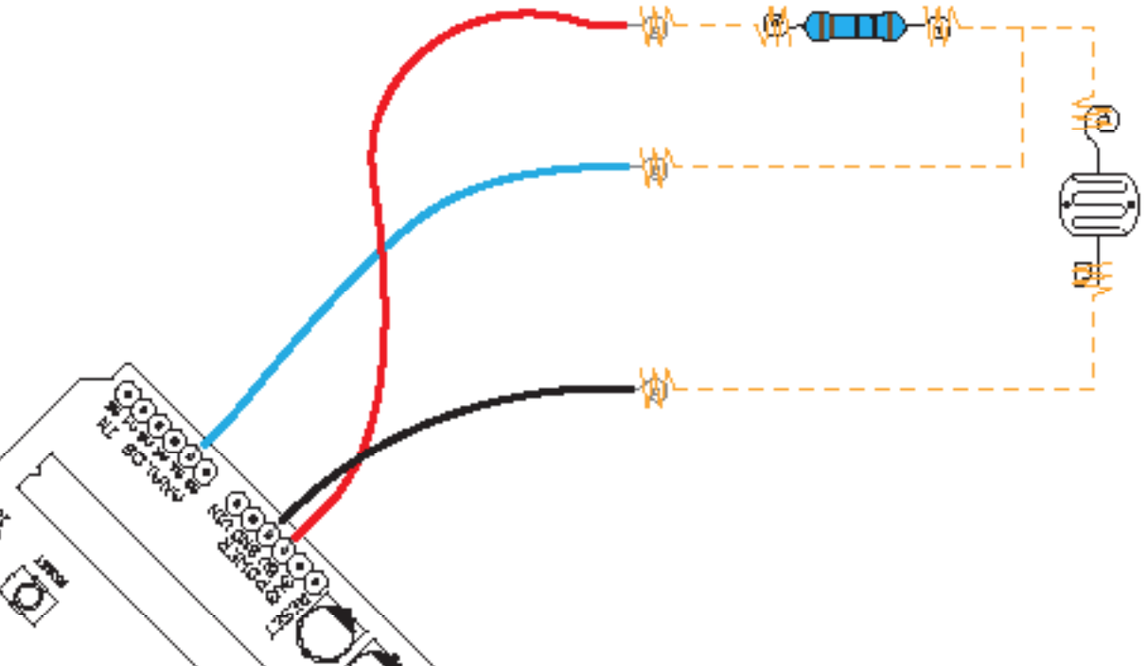


一旦你上传的代码和连接电缆，打开你的序列的监控，这是在你的Arduino的IDE在最右边的按钮。现在值应出现在串行显示器，如果你打开和关闭拉链的值应该改变。如果你看不到任何数字在你的显示器，确保在Arduino的IDE设置的通信速度是在你的程序相同（在这个例子中的9600波特）。每个拉链会给你不同的值，因为一个模拟的拉链是不是一门精确的科学;丝毫差别会影响结果。这种自制的模拟传感器形式仍然是不够好，最需要一个隐藏的输入设备的原型，使用。

2：使用LDR光传感器

下图说明了如何缝上一块面料的LDR和电阻和导线应连接到Arduino：

在这个例子中，你需要：
- 之一的LDR。
- 一个10K的电阻。
- 导电线程。
- 三线。



在这个例子中使用的电阻就是所谓的“拉

电阻器“。我们有时会使用上拉电阻，以确保我们不会得到5V直背在Arduino的。一个上拉电阻是相同的，但作为一个正常的电阻是用来降低从电源输出电压。一个电阻结合的LDR，也是必要的另一个原因是：没有它，我们将无法得到的LDR模拟值。这种配置被称为分压器的LDR的变化取决于它的变化Arduino的输入电压。

一个红色的电缆连接到一个电阻腿和5V阿尔杜伊诺。之间的电阻和LDR，我们把电缆连接到我们的模拟输入引脚上的Arduino的。在同一个连接到一条腿的LDR。不要紧，腿，你把它连接到。从另外一条腿的LDR，SEW阿尔杜伊诺的电缆连接到GND端口。当您连接一切，让我们尝试下面的代码，看到的LDR会给我们带来什么样的值：

```
int analogPin = 2;
 /*我们Arduino的使用模拟引脚*/
int myLDR = 0;
 /*临时变量来存储的LDR值*/

void setup(){
 Serial.begin(9600);
 /*设立沟通和速度*/
}

void loop(){
 myLDR = analogRead(analogPin);
 /*读取从LDR值，并将其存储*/
 Serial.println(myLDR);
 /*打印中myLDR存储的值*/
 delay(200);
}
```

在上面的代码示例中，我们读出的值，存储和打印回电脑。不要忘记在Arduino的IDE中打开您的序列监视器，并将其设置为9600波特能够看到从Arduino的印刷值。一旦你得到的一切工作，尽量覆盖LDR用你的手和检查什么价值Arduino的打印。记住这个值和尝试下

代码示例.

```
int analogPin = 2;
 /*我们Arduino的使用模拟引脚*/
int myLDR = 0;
 /*临时变量来存储的LDR值*/
int myDarkNumber = 100;
 /*对黑暗的门槛，替换为自己的这个值 */
int ledPin = 13;

void setup(){
 Serial.begin(9600);
 /*设立沟通和速度*/
 pinMode(ledPin,OUTPUT);
 /*申报输出ledPin*/
}

void loop(){
 myLDR = analogRead(analogPin);
 /*读取从LDR值，并将其存储*/
 if (myLDR <= myDarkNumber){
 digitalWrite(ledPin,HIGH);
 }else{
 digitalWrite(ledPin,LOW);
}
}
```

在这个例子中，变量名为“myDarkNumber”你从前面的例子，当你盖用你的手的LDR值。在我的情况是100，但你应该改变这个数字，你的价值。此程序将读取的LDR值，并将其与您的门槛变数（myDarkNumber），如果LDR低于或等于这个阈值内部Arduino板的LED会亮起来，如果不是的LED将保持关闭。

3：使用NTC温度传感器

下面的代码来读取热敏电阻的LDR正常工作，因为他们都是模拟传感器是相同的：

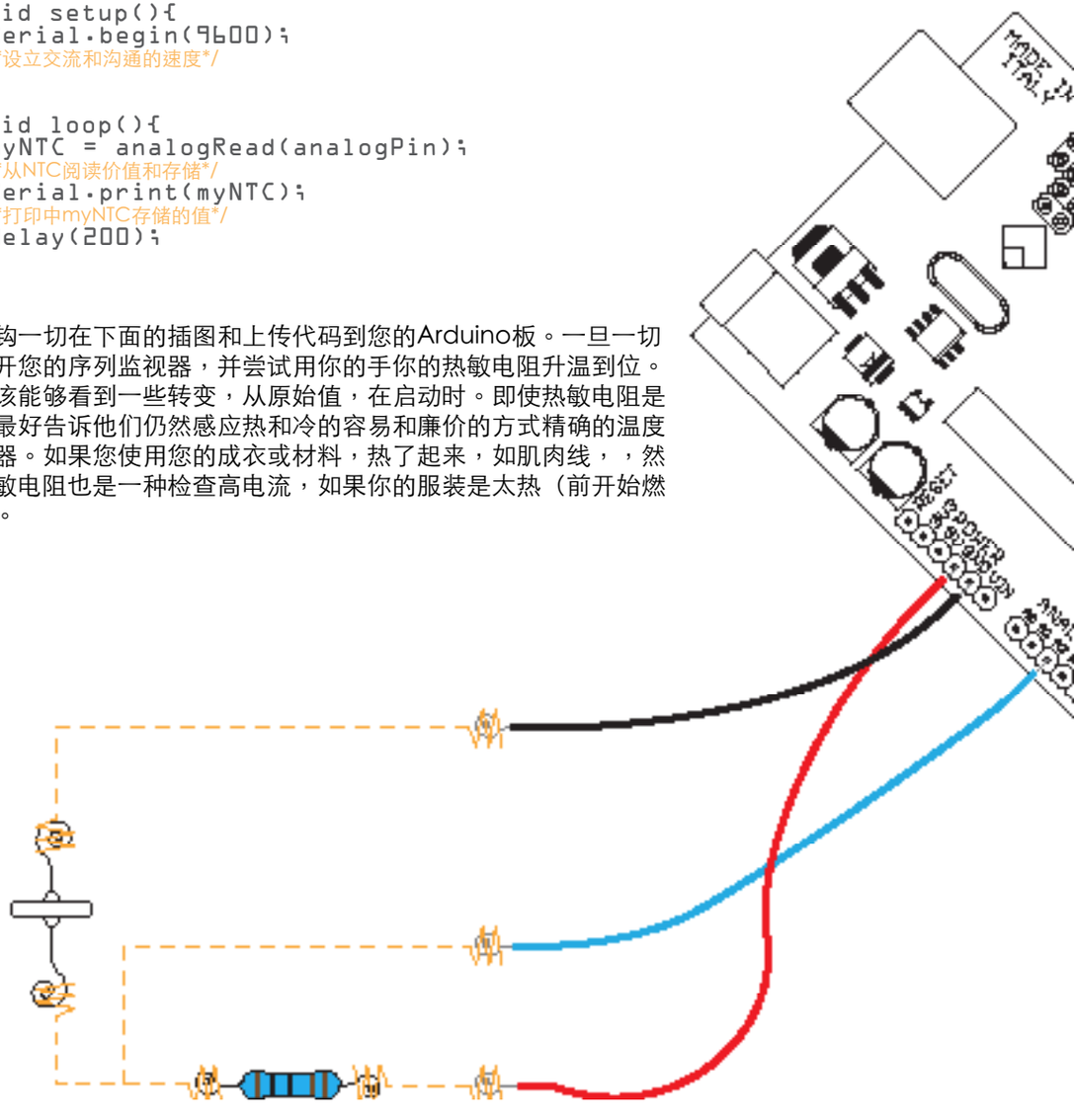
在这个例子中，你需要：
- 一个负温度系数。
- 一个10K的电阻。
- 导电线程。
- 三线。

```
int analogPin = 2;
 /*我们Arduino的使用模拟引脚*/

int myNTC= 0;
 /*临时变量来存储从热敏电阻值*/
```



```
void setup(){
 Serial.begin(9600);
 /*设立交流和沟通的速度*/
}

void loop(){
 myNTC = analogRead(analogPin);
 /*从NTC阅读价值和存储*/
 Serial.print(myNTC);
 /*打印中myNTC存储的值*/
 delay(200);
}
```

现在钩一切在下面的插图和上传代码到您的Arduino板。一旦一切都打开您的序列监视器，并尝试用你的手你的热敏电阻升温到位。你应该能够看到一些转变，从原始值，在启动时。即使热敏电阻是不是最好告诉他们仍然感应热和冷的容易和廉价的方式精确的温度传感器。如果您使用您的成衣或材料，热了起来，如肌肉线，，然后热敏电阻也是一种检查高电流，如果你的服装是太热（前开始燃烧）。



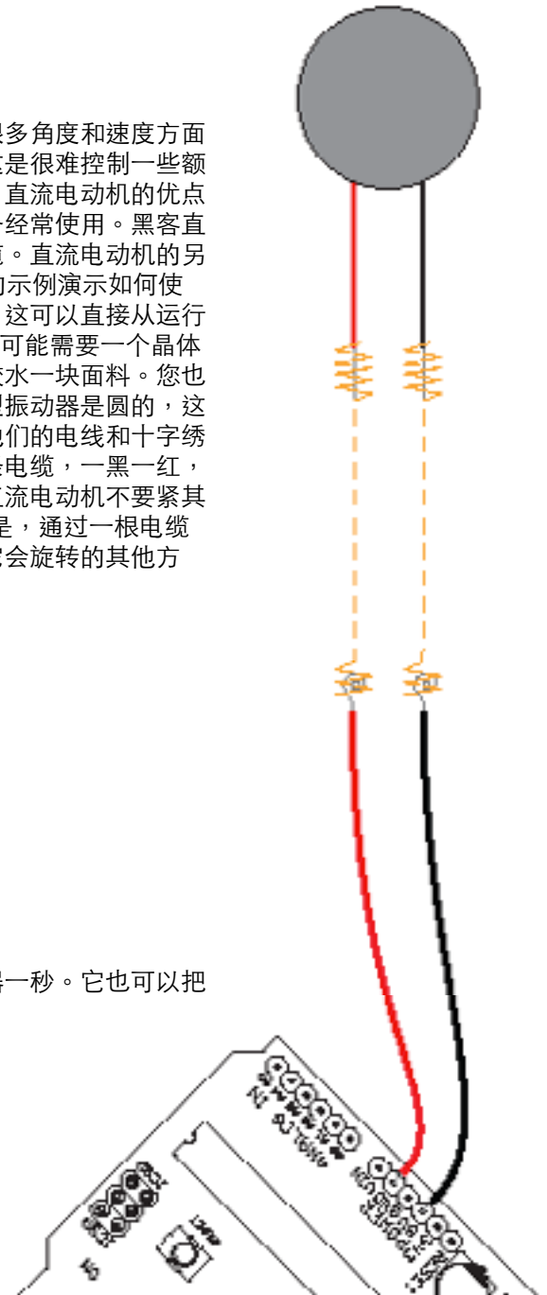
第7章：移动的东西

•直流电动机
这种类型的电机通常是在玩具和设备，不需要很多角度和速度方面的准确性。直流电动机运行自由通电时，虽然这是很难控制一些额外的电路和元件，其速度和位置，这可以改善。直流电动机的优点是，一些很便宜，便宜，有某种运动的电气设备经常使用。黑客直流电动机是很容易的，因为它只能使用两条电缆。直流电动机的另一个优点是，其中一些可低于2.5V供电。下面的示例演示如何使用一个小的直流电动机，也作为一个小型振捣。这可以直接从运行Arduino的数字引脚，但与其他直流电动机，您可能需要一个晶体管连接和外部电源。缝制一个迷你振动器开始胶水一块面料。您也可以缝超过使用正常的线程，但因为大多数小型振动器是圆的，这可能是一个有点棘手。一旦振捣到位，减少对他们的电线和十字绣与一些导电线程塑料。现在，你的面料缝制两条电缆，一黑一红，并从黑到一个振动器电缆和其他红色连接：用直流电动机不要紧其中电缆接通电源并连接到GND端口。所不同的是，通过一根电缆通电，电机转动单程，如果应用到其他方面，它会旋转的其他方式。
现在，当我们在一切就绪让我们写了以下程序：

```
int motorPin = 5;

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

void loop(){
 digitalWrite(motorPin, HIGH);
 delay(1000);
 digitalWrite(motorPin, LOW);
 delay(1000);
}
```

这是一个简单的程序，打开一秒钟，关闭振动器一秒。它也可以把振动速度和逐渐下降，使用PWM：

```
int motorPin = 5;
  /*您的电机连接到PWM引脚5*/

void setup(){
```



```
  /*什么也没有发生在设置*/
}

void loop(){
 for(int i = 0; i <= 255; i++){
 /*只要我是超过255个，增加了一个!*/
 analogWrite(motorPin,i);
 /*淡出我的速度振动器*/
 delay(30);
 /*小暂停，所以我们可以看到每一步*/
}

for(int i = 255; i >= 0; i--){
 /*只要我是大于0，减少了一个我*/
 analogWrite(motorPin,i);
 /*淡出我的速度振动器*/
 delay(30);
 /*小暂停，所以我们可以看到每一步*/
}
}
```

上面的例子将慢慢地打开振动器的最高速度，然后再返回。

•伺服电机

伺服电机有两种类型：正常和连续旋转伺服电机。正常的伺服电机控制PWM和旋转位置取决于脉冲宽度。他们只能通过0到180度旋转。连续旋转伺服电机不断旋转时功率脉冲。最常见的视差连续旋转伺服电机在一个方向旋转以上的1500微秒的延迟，并在相反的方向时，脉冲功率脉冲低于1500微秒的延迟。

根据您的项目，你将不得不使用你的想象力，隐藏了伺服电机。在下面的例子中，我们将只显示如何控制伺服电机，因为没有通用的方法是在面料实施舵机。延长线的长度从伺服到Arduino仍有可能使用导电线程。

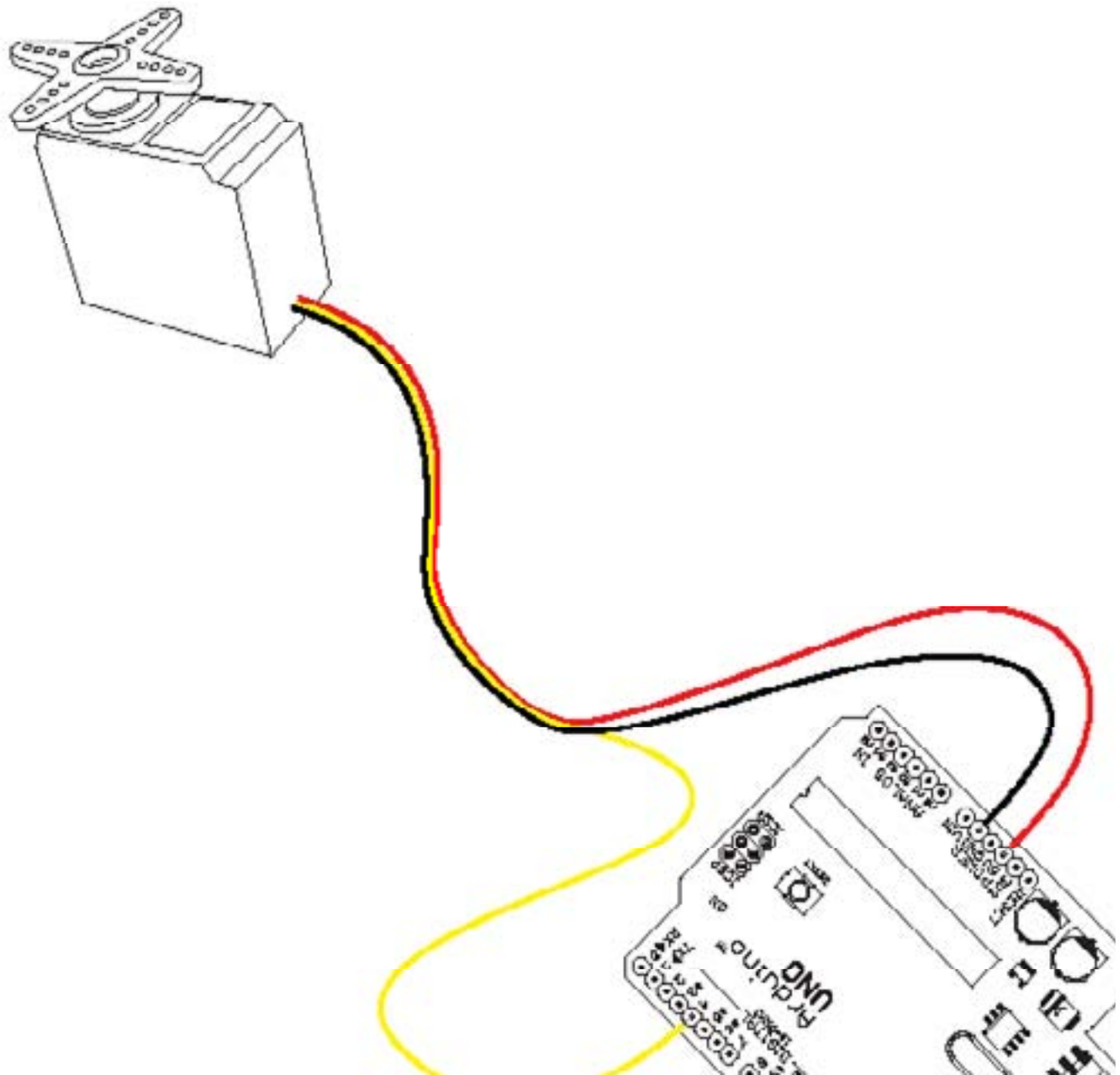
第一个例子是一个简单的程序，说明如何将一个连续伺服电机前进和后退：

```
int motorPin = 4;
  /* 伺服数字引脚*/
```



```
void setup(){
 pinMode(motorPin,OUTPUT);
 /*申报数字引脚为输出*/
}

void loop(){
 for(int i = 0; i < 100; i++){
 /*循环100次，所以我们可以看到伺服移动*/
 digitalWrite(motorPin,HIGH);
 delayMicroseconds(1850);
 /*1500以上的延迟，使伺服走单程 */
 digitalWrite(motorPin,LOW);
 delayMicroseconds(1850);
}

for(int j = 0; j < 100; j++){
 /*循环100次，所以我们可以看到伺服移动*/
 digitalWrite(motorPin,HIGH);
 delayMicroseconds(1250);
 /*低于1500的延迟，使伺服走另一条路*/
 digitalWrite(motorPin,LOW);
 delayMicroseconds(1250);
}
}
```

这个计划将使连续旋转伺服走100步，100步其他方式。一旦你的代码完成后，把它上传到Arduino板和电机连接如下图。



第8章：复杂的例子

1：带拉链的振荡

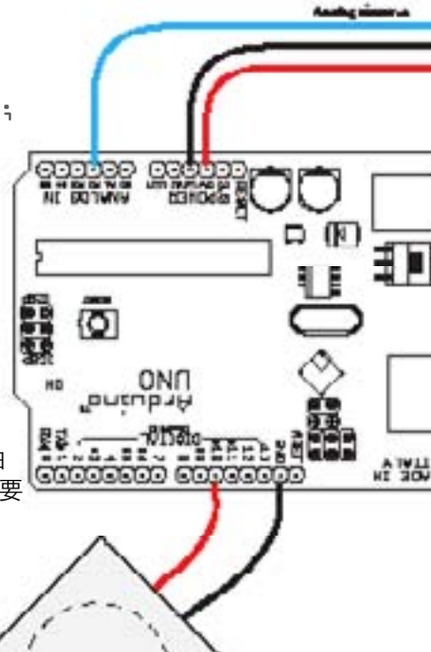
振荡是在一个核心价值的时间，或两个或两个以上国家之间的变化。用振荡器，我们可以手动改变我们的压电扬声器的音从前面的例子。模拟拉链将作为振荡器工作完美，因为它给了我们一个漂亮的范围内可以重新映射到各种不同的色调值。

在这个例子中，你需要：
· 压电扬声器
页44-45。
•模拟拉链
页51-54。

下面的程序将读取从拉链，重新映射值的值，然后使用该值设置压电扬声器播放的音调：

```
int piezoPin = 10;
 /* 声明压电扬声器连接到引脚 */
int analogPin = 2;
 /* 声明拉链连接到引脚*/
int myTemp = 0;
 /* 声明一个临时存储变量*/
int myRemapValue = 0;
 /* 宣布重新映射的值的变量*/

void setup(){
 pinMode(piezoPin,OUTPUT);
 /* 申报输出piezoPin*/
}

void loop(){
 myTemp = analogRead(analogPin);
 /* 读取和存储从拉链的价值*/
 myRemapValue = map(myTemp,100,300,0,2000);
 /* 从拉链的价值重新映射到内适应范围为0至2000年*/
 digitalWrite(piezoPin,HIGH);
 /* 5V发送到压电扬声器*/
 delayMicroseconds(myRemapValue);
 /* 暂停与重新映射值*/
 digitalWrite(piezoPin,LOW);
 /* 发送0V至压电扬声器*/
 delayMicroseconds(myRemapValue);
 /* 重新映射值再次暂停 */
}
```

注意：
您需要知道什么的最高和最低值的拉链让您。在这个例子中，我们将使用一个虚构的范围为100至300人。您可以使用Arduino软件AnalogInSerial例如阅读values from the zipper.



这个例子实现了map（）函数。要了解更多有关地图（）函数86-87页。你的拉链可能会不够好，直接设置一些色调的值，但由于模拟传感器从不放弃一个价值1023以上，他们将有一个比想的要

低得多的范围。

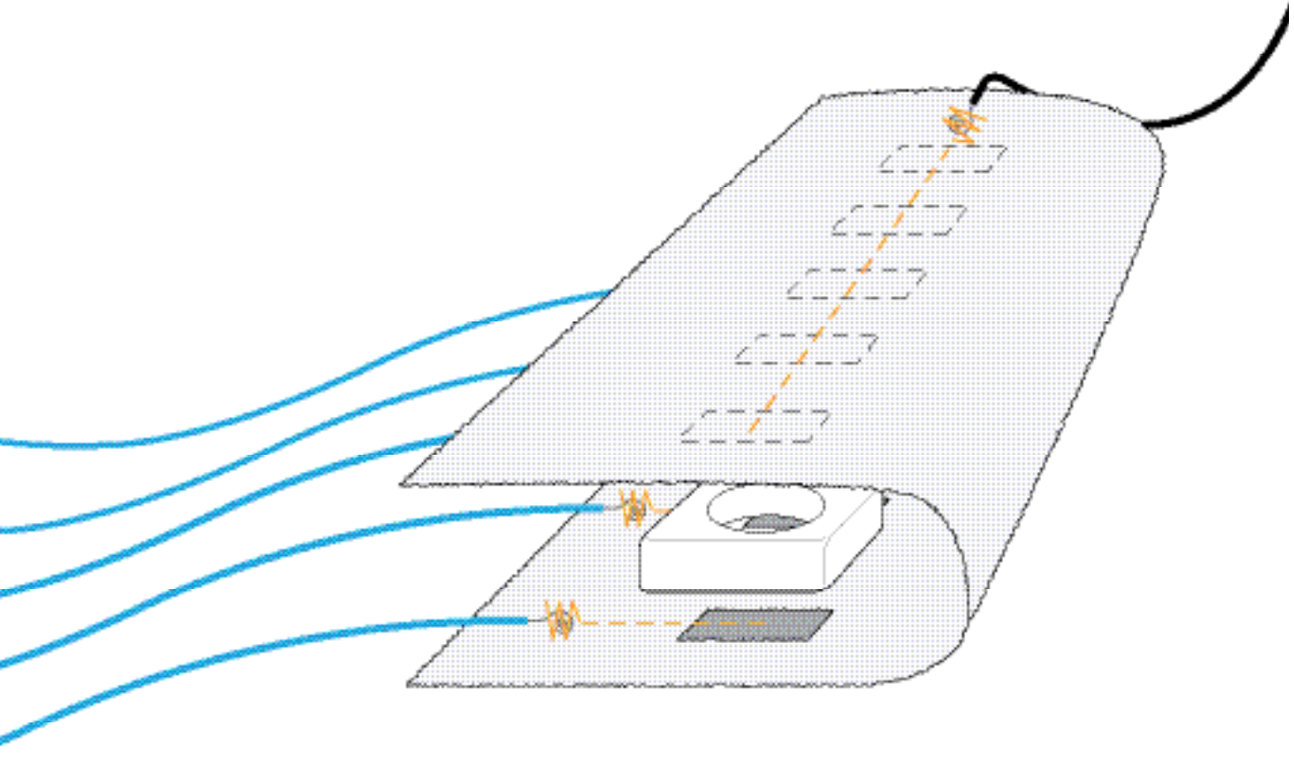
一旦你的代码完成后，尝试挂钩一切Arduino板如图所示，测试拉链振荡器。

2：软合成器

41-42页上创建软按钮的例子所示，使用相同的方法。在这个例子中，我们将使用一个大的面料，使所有的按钮空间：

在这个例子中，你需要：
- 五软按钮。
- 压电扬声器。
- 五线。
- 导电线程。
- 导电织物。
- 单件正常

在一个按钮的一侧，附加5电缆，并从一个电缆连接到一个导电布片缝到另一个所有的按钮。从另一端的按钮，以另一片导电布针连接到一个线程长线，该线程结束连接黑线。这将作为一个地连接所有的按钮，因为有没有足够的GND上所有的按钮Arduino的端口。



现在，我们已经准备好为我们的软合成器的软键盘的一部分，我们就可以开始我们的计划。要特别注意设置功能。在这里，我们设置了五个按钮到5V。我们可以做到这一点，因为Arduino的内部上拉电阻。这意味着，通过添加这些代码行，我们不必使用额外的电阻，以便能够读取数字引脚上的值：

```
int piezoPin = 10 ;
 /*声明压电扬声器连接到引脚*/
int key0ne = 2;
 /* 申报的第一个软按钮*/
int keyTwo = 3;
 /*申报的第二个软按钮*/
int keyThree = 4;
 /*申报第三个软按钮*/
int keyFour = 5;
 /*申报第四个软按钮 */
int keyFive = 6;
 /*申报第五软按钮*/

void setup(){
 pinMode(piezoPin,OUTPUT);
 /*声明piezopin作为输出*/
 pinMode(key0ne,INPUT);
 /* 声明作为输入的第一个软按钮 *
 pinMode(keyTwo,INPUT);
 /*声明作为输入的第二个软按钮*/
 pinMode(keyThree,INPUT);
 /*声明作为输入的第三个软按钮*/
 pinMode(keyFour,INPUT);
 /* 作为输入申报第四个软按钮*/
 pinMode(keyFive,INPUT);
 /* 声明作为输入的第五个软按钮*/
 digitalWrite(key0ne,HIGH);
 /*第一个软按钮发送5V*/
 digitalWrite(keyTwo,HIGH);
 /*第二个软按钮发送5V*/
 digitalWrite(keyThree,HIGH);
 /* 第三个软按钮发送5V*/
 digitalWrite(keyFour,HIGH);
 /*第四个软按钮发送5V*/
 digitalWrite(keyFive,HIGH);
 /*第五软按钮发送5V*/
}

void loop(){
 if(digitalRead(key0ne) == L0W){
```

```
  /* 测试，如果第一个软按钮被按下 */
  digitalWrite(piezoPin,HIGH);
  /*发送5V压电扬声器*/
  delayMicroseconds(1911);
  /* 等待1911年微秒（创建一个C）*/
  digitalWrite(piezoPin,LOW);
  /*停止发送5V至压电扬声器*/
  delayMicroseconds(1911);
  /*等待另一个1911年微秒*/
}

if(digitalRead(keyTwo) == LOW){
  /*如果第二个软按钮是推测试 */
  digitalWrite(piezoPin,HIGH);
  /*发送5V压电扬声器*/
  delayMicroseconds(1703);
  /*等待1703微秒（创建一个D）*/
  digitalWrite(piezoPin,LOW);
  /*停止发送5V至压电扬声器*/
  delayMicroseconds(1703);
  /* 等待另一个1703微秒 */
}

if(digitalRead(keyThree) == LOW){
  /*如果第三个软按钮是推测试*/
  digitalWrite(piezoPin,HIGH);
  /*发送5V压电扬声器*/
  delayMicroseconds(1517);
  /*等待1517微秒（创建一个E）*/
  digitalWrite(piezoPin,LOW);
  /*停止发送5V至压电扬声器*/
  delayMicroseconds(1517);
  /*等待另一个1517微秒*/
}

if(digitalRead(keyFour) == LOW){
  /*如果第四个软按钮是推测试*/
  digitalWrite(piezoPin,HIGH);
  /*发送5V压电扬声器*/
  delayMicroseconds(1432);
  /*等待1432微秒（创建一个F）*/
  digitalWrite(piezoPin,LOW);
  /*停止发送5V至压电扬声器*/
  delayMicroseconds(1432);
  /* 等待另一个1432微秒*/
}

if (digitalRead(keyFive) == LOW){
```

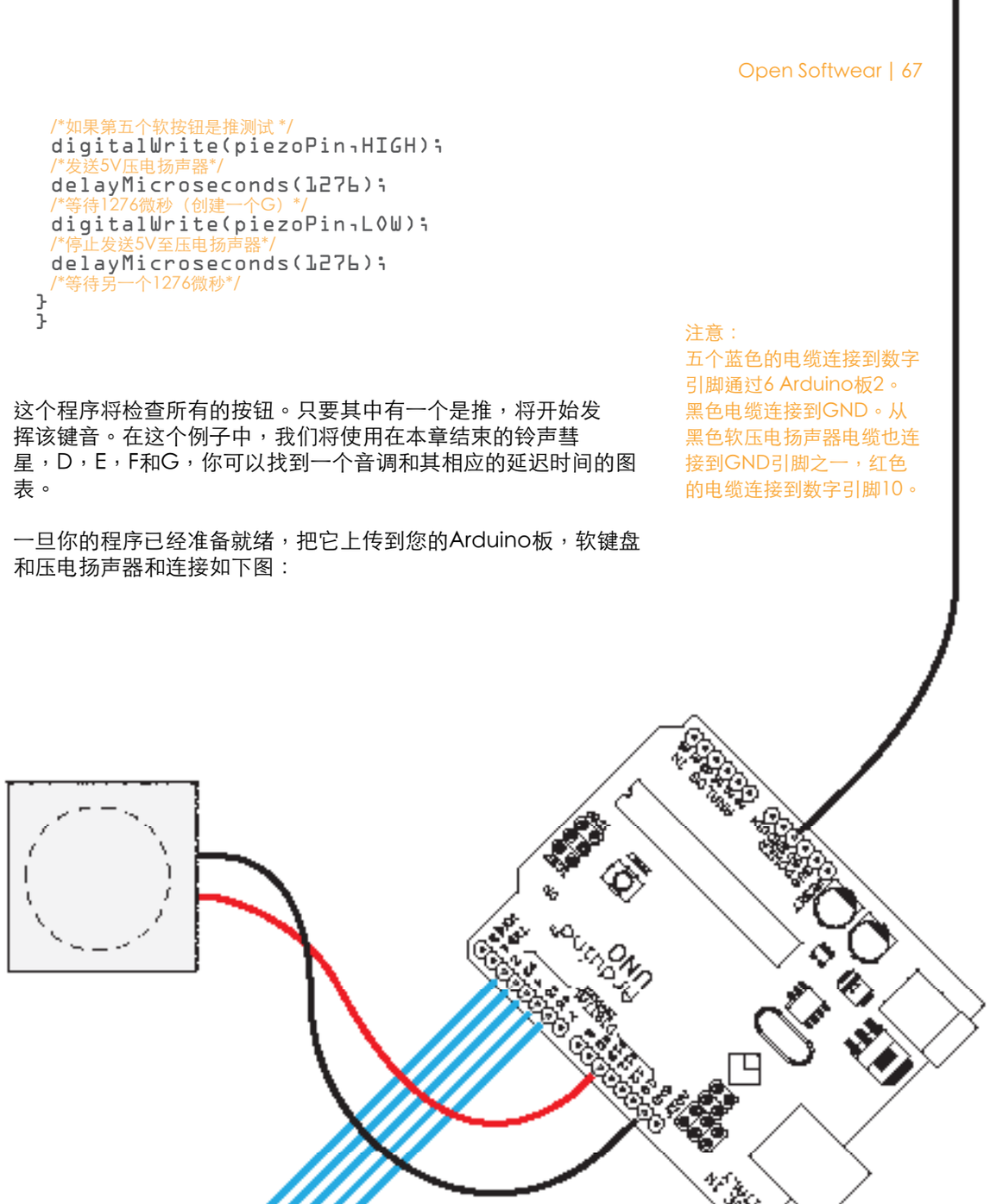


```
  /*如果第五个软按钮是推测试 */
  digitalWrite(piezoPin,HIGH);
  /*发送5V压电扬声器*/
  delayMicroseconds(1276);
  /*等待1276微秒（创建一个G）*/
  digitalWrite(piezoPin,LOW);
  /*停止发送5V至压电扬声器*/
  delayMicroseconds(1276);
  /*等待另一个1276微秒*/
}
}
```

这个程序将检查所有的按钮。只要其中有一个是推，将开始发挥该键音。在这个例子中，我们将使用在本章结束的铃声彗星，D，E，F和G，你可以找到一个音调和其相应的延迟时间的图表。

一旦你的程序已经准备就绪，把它上传到您的Arduino板，软键盘和压电扬声器和连接如下图：

注意：
五个蓝色的电缆连接到数字引脚通过6 Arduino板2。黑色电缆连接到GND。从黑色软压电扬声器电缆也连接到GND引脚之一，红色的电缆连接到数字引脚10。



压电扬声器也可以实施到键盘结构，使您的设计更美观。

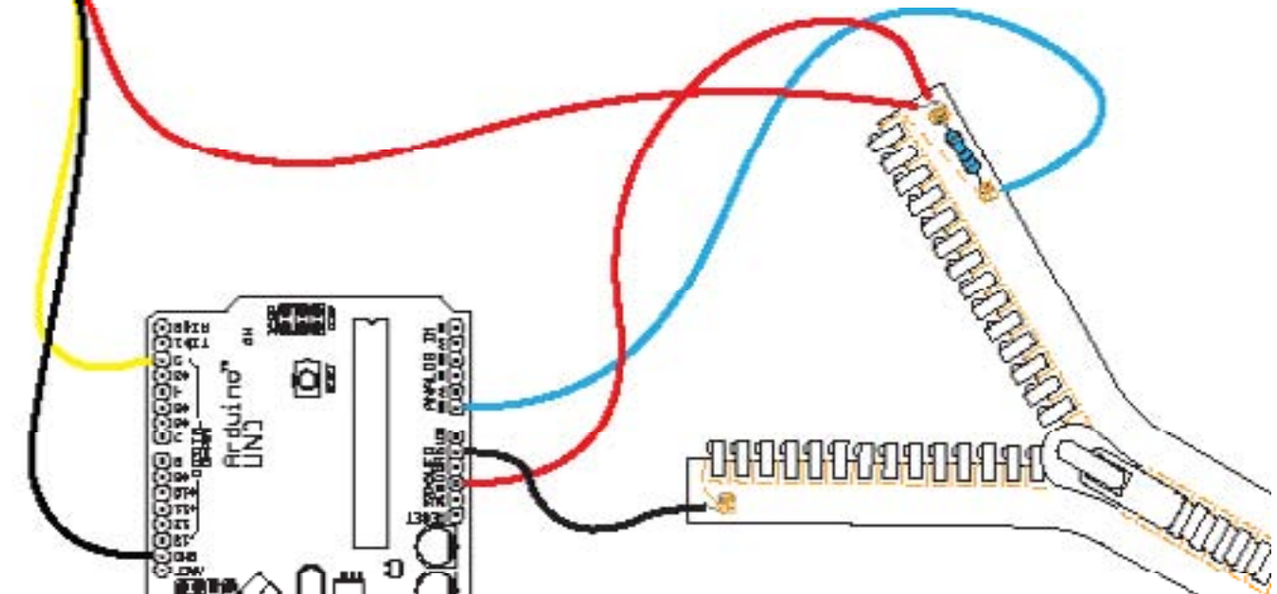
图的色调和其相应的延迟时间：

音	延迟
彗星1	911年
ð	1703
发送	1517
F	1432
摹	1276
一个	1136
乙	1012

在这个例子中，你需要：
•模拟拉链
页51-54。
•一个正常的伺服电机。

3：一个带拉链的正常伺服控制

在这个例子中，我们将使用从第51-54页的模拟拉链控制在一个正常的伺服电机的旋转。下图显示了如何连接伺服和拉链：



写程序，你首先要知道你的拉链的最高和最低值。转到页51-54，如果你不记得如何读取从模拟sensor.The最低和最高值，从拉链的值将被

重新映射在我们的程序使用的地图（）函数的最小和最大的伺服电机的旋转值：

```
int servoPin = 2;
 /* 伺服电机的数字引脚*/
int servoMin = 500;
 /* 最低伺服位置*/
int servoMax = 2500;
 /*最大的伺服位置*/
int pulse = 0;
 /*量脉冲伺服*/
long lastPulse = 0;
 /* t他在最后一个脉冲的时间以毫     */
int refreshTime = 20;
 /*在脉冲之间所需的时间*/
int myZipper = 0;
 /*从模拟传感器的返回值*/
int analogPin = 0;
 /*该传感器连接到模拟输入引脚*/

void setup(){
 pinMode(servoPin,OUTPUT);
 /*伺服引脚设置为输出*/
 pulse = servoMin;
 /*电机的位置值设置到最低*/
 Serial.begin(9600);
 /* 设置串行通信速度*/
}

void loop(){
 myZipper = analogRead(analogPin) ;
 /*读取模拟输入*/
 pulse = map(myZipper,0,1023,servoMin,servo
Max);
 /*重新映射值从拉链之间servoMin和servoMax范围*/
 if (millis() - lastPulse >= refreshTime) {
 /*如果millis的（这是millisecondsthe Arduino的金额已通电）减去
 从“lastPulse”的值大于或等于“refreshTime，”价值，那么这段代码被触发
 */
 digitalWrite(servoPin,HIGH);
 /*开启电动机 */
 delayMicroseconds(pulse);
 /*脉冲的长度设定电机的位置*/
```

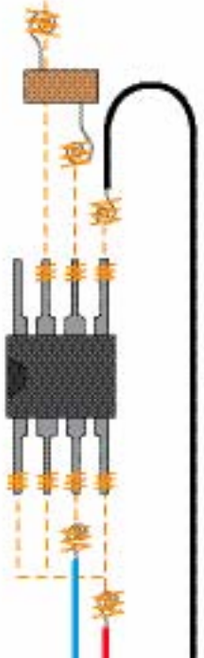
```
  digitalWrite(servoPin,LOW);
  /* 请关掉电机*/
  lastPulse = millis();
  /* 保存最后一个脉冲的时间*/
}
}
```

这个程序将读取从拉链的价值，并重新映射成将用于设置电机的位置脉冲值。一旦你完成你的代码，把它上传到你的板子，并尝试将拉链，以移动伺服电机。在这个程序，你必须添加地图功能地图（myZipper，您的拉链分钟，您的拉链最大，500，2500）你的拉链的最高和最低值;。

4：触摸敏感刺绣

在这个例子中，我们将作出刺绣触摸敏感，这将使得像一个按钮的绣法。为了使刺绣，我们需要使用一个额外的芯片。 QT118芯片是一个自包含的数字IC，能够检测附近接近或触摸触摸传感器。有不同类型的QT芯片QT118我们要使用只能处理一个触摸感应输入。QT芯片的工作原理就像一个ON / OFF开关，可以延长任何导电材料，如电线或导电线程芯片的触摸灵敏度。左边的图这里显示的QT118芯片的腿，他们需要连接到。

在这个例子中，你需要：
- 一个QT118芯片。
- 一个33N电容。
- 导电线程。
- 三个电缆。



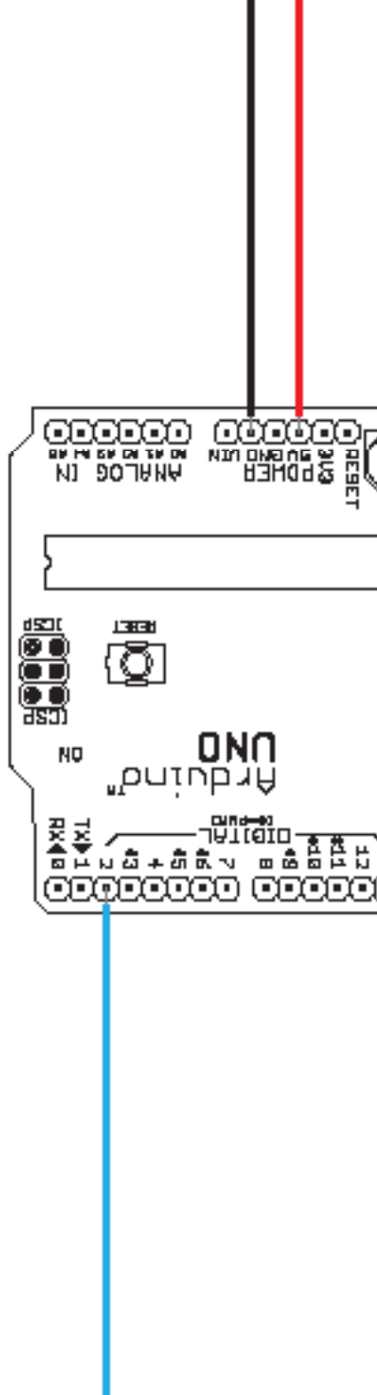
为了使刺绣触摸敏感，我们需要增加一个电容。电容甚至会出的信号，进入我们的刺绣，从QT芯片，使我们能够生成具有参考价值。的QT芯片将使用这个参考值比较刺绣时触及到时，它不是的信号。左边的图显示了如何连接电容器芯片，并在那里你可以开始你的刺绣：

要触摸感应的所有部分都被连接到芯片的引脚连接到线程。以下的导线连接到芯片中。

一旦已作出所有芯片的连接，我们可以开始编写我们的程序。QT118将发送信号为5V时，它是感动。我们可以读出的数字和模拟引脚信号，但如果你有数字引脚可用它没有任何意义，使用模拟引脚，由于信号只会为0（未触及芯片）或1023（“读芯片是感动）。下面的程序是一个简单的草图就如何点亮上板上的LED触摸刺绣时：

```
int ledPin = 13;
 /*板上的LED阿尔杜伊诺*/
int touchChip = 2;
 /*从触摸芯片的输出信号连接到引脚2*/

void setup(){
 pinMode(ledPin, OUTPUT);
 /*申报输出ledPin*/
 pinMode(touchChip, INPUT);
 /* 声明为输入touchChip*/
}

void loop(){
 if (digitalRead(touchChip) == HIGH){
 /*检查如果是感动的刺绣*/
 digitalWrite(ledPin, HIGH);
 /*如果是这样，点亮LED */
 }else{
 digitalWrite(ledPin, LOW);
 /*如果不是，LED熄灭*/
}
}
```

一旦代码完成，我们可以把它上传到电路板。然后，我们的刺绣连接到芯片上，在图。



刺绣，将双方的敏感，所以如果你想能穿一个触摸感应的刺绣，你需要把一些防护面料服装内侧。您可能需要与一些面料的实验，找到一个足够厚，为您的绣花，并确保您正在使用的面料是不导电的。

5：肌肉线

使用肌丝是一种方便的方式来创建，而不必把笨重的电动机在面料的运动。肌丝是由所谓的镍钛金属，能记住形状。当你加热丝，用吹风机或电力，它可以追溯到它被训练记住的形状。如果你能获得的被训练成弹簧形状的合同的线，因为该形状将引发最大的运动。

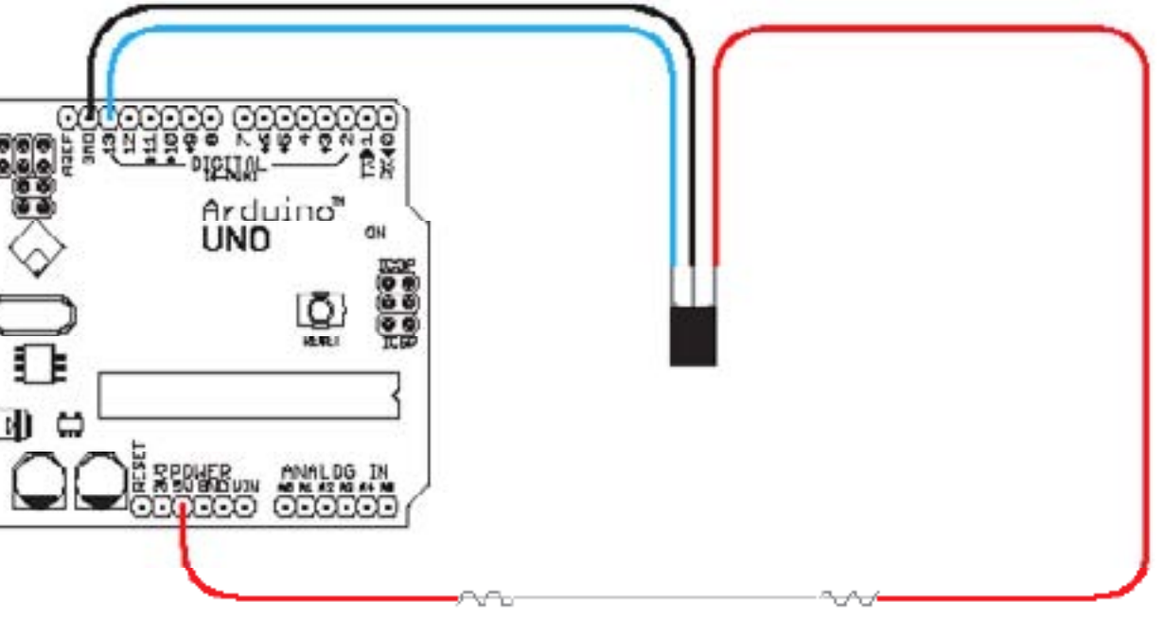
到您的面料手工缝制的肌肉线。使用薄型织物，织物或一块布，这只是一个侧面连接的小片，像线。这将确保电线被激活时，您将仍

在这个例子中，你需要：
•25厘米训练有素的0.15毫米的肌肉线厚度。
•2卷曲珠。
•2N4401晶体管。
•4定期电线。

然可以看到织物的移动，当你离它远一点。肌肉线不能进行焊接，使卷曲珠是用来连接的肌肉线两侧的两个定期电线。为了让你把一个侧面肌丝和定期线通过压接珠，并用平口钳，一个连接牢固，你挤在一起。

请检查的两条电线无法逃脱从扁珠。肌肉线使用了大量的电流，可以提供超过你的Arduino的。为了让Arduino的肌肉线，你必须使用一个晶体管，其引脚之一。插图显示了你如何连接的晶体管和Arduino的肌肉线到正规的电线，并定期电线。使用此代码来让面料移动：

```
int musclePin = 13;

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

void loop(){
 digitalWrite(musclePin, HIGH);
 /*引脚上，从而让肌肉线的升温和合同其训练有素的形状*/
 delay(4000);
 /* first delay */
 digitalWrite(musclePin, LOW);
```




```
 /* 关闭引脚，允许电线冷静下来，放松*/
 delay(4000);
 /*秒的延迟*/
}
```

肌丝0.15毫米的厚度只有一个最面料的潜移默化的作用，因为它不是很强烈。厚线有较大的影响，但还需要更多的电流。如果您选择使用厚度大于0.15毫米的导线，确保你不要离开太久。你可能过热电线。

0.15毫米或更薄的电线没有这个问题，可以保留，只要你喜欢。如果你想移动肌肉线，您需要调整代码中的第一个延迟时间，在这个例子中，25厘米以上。线较长，花费更多的时间来加热。这意味着，而不是4000毫秒的延迟时间可能需要6000毫秒。找出您的延迟时间需要多久，要尝试不同的时间长度。你知道你已经有了正确的延迟时间，当你看到肌肉充分纳入其最小的形状线合同。

因为肌肉线会移动，但它不便于用导电线电子连接线可拉本身的松散。不使用导电线程的另一个原因是，它的电阻。可以很容易地超过肌肉线电阻的导电线程。这意味着，如果你使用

代替正规电线的导电线程，你也许不会成功升温肌肉线，因为你会升温以及导电线程。一般情况下，你需要用导电线和更大的电流，你可能放火烧你的纺织谨慎。

第三部分：编码

第9章：编写程序

Arduino的程序称为C.编写代码时，重要的是要牢记，Arduino的不作为一个人的功能一个编码语言编写的代码，它没有理由。如果有人一个字发音不正确，你将可能仍然理解的含义 - Arduino的却不会。而写在Arduino的IDE中的代码，如果你拼错了命令，也不会明白你正在尝试做的。

另一件要记住的是，Arduino是合乎逻辑的，但不理性的。它不知道你想要做什么，它只是你告诉它做。

基本结构

当我们写在Arduino的IDE中的代码，我们使用一个基本的三个步骤的结构。

•变量声明

•在设置（）

•在循环（）

在setUp（）和循环（）是编写一个正常运作的程序的必要条件。当你变得更有信心在阿尔杜伊诺编写代码，你会发现，这是不是强制性的编写代码

使用这些步骤。然而，它是一个有用的方法，因为它使程序更容易的概述，这有助于寻找代码中的错误，当你的结构。下面是一个简单的程序，轮流在船上Arduino的开启和关闭LED的例子：

```
int ledPin = 13;

void setup(){
 digitalWrite(ledPin,OUTPUT);
}

void loop(){
 digitalWrite(ledPin,HIGH);
 delay(1000);
 digitalWrite(ledPin,LOW);
 delay(1000);
}
```


变量

一个变量是像别的东西的容器。比方说，你想读的某种传感器。这种传感器会给你在一个数字的形式值。如果你想使用这个值在一个以上的计划的一部分，你可以存储在一个变量，以节省时间这个值。之前可以存储的价值，您需要声明变量，这意味着你告诉程序变量存在什么样的变量，它是。

注意：
把你的变量在代码的开头，它总是好的。

我们要的变量名。的名称可以是任何东西，但它是一个好主意，让您的逻辑变量名。当你概述你的程序将会更容易地确定什么样的价值是存储在一个名为“tempSensor”一个变量。这可能是很难记得什么是隐藏在像“香蕉”，“彼得”或“supercalifragilisticexpialidocious”名称的变量。在此更多的是解释变量类型“和”宣言“第80-82页。

太虚设置

Arduino的不启动时的第一件事情是寻找设置（）无效。这是一个功能程序的基本步骤之一。设置（）无效的部分初始化的Arduino的模式不同部位，例如您的PIN模式或设置的通信速度。虚空的setUp（）只运行一次启动时，不会再次运行，直到Arduino的断电和重新打开或重置。

以下是无效的设置，设置为输出引脚模式的一个例子。

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

虚空循环

无效循环（）是第二个功能的重要一步

方案。这是你的程序的行为发生。顾名思义，这部分运行一遍。在一个Arduino的程序的全部代码是一行行执行。 Arduino的叶子在启动时设置（）无效后，它看起来无效循环（）进入它。It 然后开

始做的一切，在你的代码从一开始就无效循环（）底部。当它到达底部无效循环（），它只是再次开始允许改变程序。以下是无效循环（从本章开始的例子）：

```
void loop(){
 digitalWrite(ledPin,HIGH);
 delay(1000);
 digitalWrite(ledPin,LOW);
 delay(1000);
}
```

这种无效循环（）包含的代码，将打开名为“ledPin”的变量相同数量的引脚上。下一行代码创建一个延迟，然后打开相同的引脚，然后创建一个新的延迟。如果我们在Arduino的运行此代码与一个名为“ledPin”的变量相同数量的LED连接引脚，它会闪烁一秒钟的延误和关闭LED，直到我们关闭电源。

括号

括号是用来定义代码的某些部分的开头和结尾。有两种类型的括号内为Arduino的编写代码时使用。首先是左，右括号（），通常只是所谓的括号内。这些括号是用来写作时我们的计划内的功能。它们被用来作为一个在我们的程序发送一个变量在别处空间。

它也可能有一个空括号内的功能，

但它仍然是有必要把他们后，你的函数的名称或Arduino的会给你一个编译错误。的功能，使用空括号的一个例子是无效的setUp（）和无效循环（）。

第二类是大括号或花括号{}。这些都是

用来显示一个函数的开始和结束。如果没有这些括号阿尔杜伊诺将无法知道该函数的开头和结尾，什么考虑下一段的代码。这些括号中的一个共同的地方是无效的setup（）函数。

```
void setup(){
 //The code in the function goes in here.
}
```


分号

分号的Arduino的编写代码的一个最重要的部分，最容易忘记。它们被用来分开你的程序代码在不同的线路，并告诉您的命令结束的Arduino的。如何声明一个变量的正确使用分号，下面的例子是：

```
int myNumber = 15;
```

分号结束命令，我们已经创建了一个名称为“myNumber”，这个变量将值15的整数变量。如果您忘记了您的代码以分号阿尔杜伊诺IDE会让你知道给你一个编译错误，当您尝试编译的代码和IDE将突出显示的代码行缺少分号。

评论代码

有时可以把笔记或写你的代码里面为自己或别人的意见。如果你写你的程序内的文字会认为这是Arduino的代码，并尝试执行写的是什么。如果是Arduino的不理解的东西写的是什么，它会给你一个错误消息。

有两种方法你可以写在你的代码的消息和隐藏阿尔杜伊诺。第一个是使用双斜线//前面的任何消息。这将隐藏从Arduino的消息，但仍然留下你的眼睛可见。下面是一个无效循环（）里面隐藏着一个消息的例子：

```
void loop(){
 digitalWrite(ledPin,HIGH); // 轮番上的LED
 delay(1000); // 等待一段时间
 digitalWrite(ledPin,LOW); //打开LED熄灭
 delay(1000); // 等待多一点
}
```

如果你想隐藏的消息长于行，你必须使用/ *和*/。为了纪念一个被隐藏的消息开始，您可以使用/ *和标记消息的结束，您可以使用*/。这将隐藏整个消息。以下是一个例子，如何隐藏一个无效的循环内的文本块（）：

```
void loop(){
 /* 此代码首先将一个主导的，那么它会等待一段时间之后，
 它会变成LED熄灭，然后再次等待。*/
 digitalWrite(ledPin,HIGH);
 delay(1000);
 digitalWrite(ledPin,LOW);
 delay(1000);
}
```

类型的变量和声明

注意：
/ /消息和票据的唯一作品
不再一行

给一个变量的值也被称为声明一个变量。声明一个变量，意味着你给一个变量的类型，名称和值。int myNumber = 14;

在上述“INT”的例子是类型，“myNumber”的名称和值14。请注意，您总是给一个变量值，当你声明。说，我要保存在我的程序从一个传感器值，但我不能从传感器读取的值，当我们宣布无效循环（）外，在程序的开始。你只要给一个临时值0当您声明您的变量，在下面的例子一样，代码开始：

```
int mySensor = 0;
```

声明一个变量有两种可能的方式。如果我们在我们的计划开始前宣布无效设置（），这将是什么是所谓的全局变量。全局变量可以访问你的程序的每一个部分。与此相反的是被称为局部变量，只能在函数内部使用，其声明。以下示例显示了一个变量，命名ledPin，是全球性的：

```
int ledPin = 13;

void setup(){
 digitalWrite(ledPin,OUTPUT);
}

void loop(){
 digitalWrite(ledPin,HIGH);
 delay(1000);
 digitalWrite(ledPin,LOW);
 delay(1000);
}
```

变量ledPin在整个程序中是可见的和无效循环（）时，它的使用，

它会被作为一个整型变量值13。如果你不过写相同的程序如下：

```
void setup(){
 int ledPin = 13;
 digitalWrite(ledPin,OUTPUT);
}

void loop(){
 digitalWrite(ledPin,HIGH);
 delay(1000);
 digitalWrite(ledPin,LOW);
 delay(1000);
}
```

然后程序会给你一个错误，告诉你无法找到您试图使用无效循环（）的变量ledPin;因为它是宣布，因此藏在里面，设置（）无效;。

在一些方案中，可以使用局部变量非常有用，但在大多数情况下，它作为一个全局变量，最好是离开他们，因此之前宣布无效设置（）;。

一旦你有一个变量的值，你可以重新分配一个新的价值，但要注意，这将清除原有的价值。比方说，我们有一个变量myNumber，我们已宣布它作为：

```
int myNumber = 14;
```

如果以后在你的程序，你想给“myNumber”你这样做的一个新的价值：

```
myNumber = 56;
```

这将空变量“myNumber”14号和56取代。

重新分配一个新的价值，我们的变量时，我们没有这行代码前加一个“INT”，像我们这样，当我们声明的变量。这是因为我们只有一次申报程序变量的类型。它可以改变一个变量的值，但不是它的类型。

类型

到目前为止，我们一直在谈论有关“INT”，这是变量的Arduino的写作计划时，最常见的类型为整数，。常见的变量是：

•诠释

整数是用于存储数据的数字组成，没有任何小数点，它们存储在32767范围内的16位值 - 32767。这意味着，一个整型变量需要16位阿尔杜伊诺的内存，和我们有32767和使用任何数量的可能性 - 32767。

```
int myNumber = 1234;
```

•龙

在大多数情况下一个整数的长度会的工作，但在某些情况下，我们需要能够存储长于一个整数的最大大小，然后，我们长期使用的变量。一个是扩展的数据类型为整数没有小数点，并将其存储在范围214748364732位值 - 2147483647。这意味着一个长期的变量需要32位阿尔杜伊诺的内存和我们有可能使用2147483647之间的任何号码 - 2147483647。

```
long myBigNumber = 90000;
```

•字节

阿尔杜伊诺为了节省内存空间，也可以是有用的存储字节为单位的变量。一个字节是8位的数值没有小数点0到255的范围内。这意味着一个字节变量需要8位阿尔杜伊诺的内存，我们都使用0和255之间的任何数目的可能性。

```
byte mySmallNumber = 150;
```

•浮动

唯一的数据类型，可以节省小数点的数字是浮动。一个float具有更高的分辨率比整数和存储作为一个32位值的范围of 3.4028235E+38 - 3.4028235E38。这意味着与一个小数点，但仅限于3.4028235E+38的范围内，可以保存数 - 3.4028235E38。声明变量作为浮在Arduino的很大的空间。使用float也比使用整数，因为Arduino的需要更多的时间做计算浮动慢。

```
float mydecimalNumber = 2.33;
```

•阵列

有时也可以是有用的存储值的集合，然后我们需要使用一个数组。所有存储在数组中的值将被存储索引号和你收集了一定的价值指的索引号。阵列需要在相同的方式声明变量声明 - 同一个类型，名称和值的集合。下面的示例演示如何声明一个整数数组与6个不同的值：

```
int myArray[] = {1, 2, 3, 4, 5, 6};
```

注意：
使用多头可以填补Arduino的内存，所以只需要使用它们时。

注意，数组从0开始计数。这意味着数组中第一个位置是0。在上面的数字1的例子是存储在数组中的第位置，如果我们调用这个值，我们必须做它作为：

```
myNumber = myArray[0];
```

这将节省这是1，在我们的变量myNumber，数组的第一个位置的值。在相反的方向，我们可以存储在数组值在数组中的位置：

```
myArray[0] = 23;
```

0阵列中的位置，这将节省的23号。

如果你知道你将使用大量数字和要存储在一个数组，但你不知道有什么样的值是

将是，你仍然可以用类似下面的空的地方一定数额申报数组：

```
int myArray[5];
```

这将创建一个数组，其中4是一个数组从0开始计数以来的最后一个位置空位置。不要试图在第五名的位置，当你宣布它有5个位置（

从0到4）把数据。第五的位置将是无效的和Arduino的计划不会提醒您，当您验证您的代码。数据，你会得到一个无效的位置将是不正确的。

做数学

Arduino是一个小型的电脑，它可以做数学运算。它可以处理像除了最常见的算术运算符，减法，乘法和除法。

```
myValue = 1 + 1;
 /* 这将存储在myValue2 */
myValue = 4 - 2;
 /*这将存储在myValue6号*/
myValue = 3 * 4;
 /*这将存储在myValue12号*/
myValue = 6 / 2;
 /*这将存储在myValue3*/
```

如果您使用的是做数学运算的整数，它不能处理小数点的浮动是可以做到这一点的唯一的变量类型。换句话说，如果我们6除以10，它将给我们的结果为1。做太大的数学运算，也可能导致内存溢出，因为所有类型的变量有一个特定的最大尺寸。另外要注意大量的计算将缓慢阿尔杜伊诺。

您还可以做什么是所谓的复合赋值，这是当你申请到变量数学运算。以下是不同的化合物的任务，你可以申请到变量的例子： 如果您使用的是做数学运算的整数，它不能处理小数点的浮动是可以做到这一点的唯一的变量类型。换句话说，如果我们6除以10，它将给我们的结果为1。做太大的数学运算，也可能导致内存溢出，因为所有类型的变量有一个特定的最大尺寸。另外要注意大量的计算将缓慢阿尔杜伊诺。

您还可以做什么是所谓的复合赋值，这是当你申请到变量数学运算。以下是不同的化合物的任务，你可以申请到变量的例子：

```
x++
 /*这将增加一条X和它的相同
写x= X+ 1个*/
x--
 /*这将减少一个X，它的相同
写X = X - 1*/
x  += y
 /* 这将增加Y X和它的相同
写x= X + Y*/
x  -= y
 /*这将减少Y X和它的相同
写X = X - Y*/
x  *= y
 /*这将乘以Y X和它相同。
写x= X * Y*/
x  /= y
 /*这将除以Y X和它的如下写法是一样
X = X / Y*/
```


•地图

比方说，你有一个传感器，只给出一个值的范围从50到200，你需要的是一个范围从0到500。然后在地图功能，可以派上用场。地图功能重新映射范围的值到另一个值的范围：

```
myVariable = map(mySensor,50,200,0,500);
 /*sensorValue，sensorMinimum，sensorMaximum
desiredMinimum，desiredMaximum*/
```

在上面的例子中，我们使用的地图功能，存储在MYVARIABLE一个价值。该值来自mySensor和50和200大关的最小值和最大值，从我们的传感器。 0至500，是我们希望，而不是所需的范围。地图功能会自动重新映射的值范围在50到200，我们得到从mySensor值范围为0至500。请注意，如果你想使用地图功能，你必须知道要重新映射的范围。为了找出更多关于如何读值见页98-101。

•随机（最大）

随机命令将返回一个随机值范围从0到最大值放在括号内。为了能够使用这个值，你必须保存在一个变量：

```
myVariable = random(5);
```

这将节省一个随机数MYVARIABLE，从0到4不等，或直接进行比较时，您可以使用随机命令

```
if (3 == random(5)){
 doSomething;
}
```

随机命令将只返回一个介于0和值放在括号中的最大值，它永远不会返回实际的最大数量。

•随机（最小，最大）

如果你想开始一个随机数字，范围从0到别的东西比更高，你将不得不添加所需的最低值：

```
myVariable = random(200,300);
```

逻辑比较

如果你想在你的程序的比较，你有使用的可能比较运算符之一。我们使用它们比较变量或者其他变量或其他常数，以使报表可以是真或假的。

• 等于 ==

==用于比较，如果事情是等于别的。一个语句使用==只会是真实的，如果事情是完全别的东西一样：

```
x == y
 /* x 是完全一样 y */
```

• 不同于!=

!= 是用来比较的东西，如果不等于别的。一份声明使用=只会是真实的，如果事情是不完全像别的东西一样！

```
x != y
 /*x是从y*/
```

• 小于<

< 用于比较不到别的东西，如果事情是。一个声明中使用<只会是真实的，如果事情是比别的小:

```
x < y
 /* x是比Y小*/
```


• 超过>

> 用于比较，如果事情是比别的大。使用>的声明只会是真实的，如果事情是比别的大：

```
x > y
 /*x是比Y大*/
```

•小于或等于<=

<= 用于比较，如果事情是小于或等于别的。一个语句使用<=只会是真实的，如果事情是规模较小或别的东西一样：x <= y

```
 /*x是小于或等于y*/
```

• 更多或等于>=

>= 用于比较，如果事情是大于或等于别的。一个声明使用>=只会是真实的，如果事情是更大或别的东西一样：

```
x >= y
 /*x是更大或等于y*/
```

逻辑运算符

使用逻辑运算符时，你需要两个或两个以上的语句在同一个句子，这些可真或假。有三种不同的逻辑运算符，可用于各种。

• 而 &&

这是用来确定如果两个或两个以上的陈述是真实的。如果不是所有的陈述是真实的的，那么句子是假的。东西是真实的，那里有两个或两个以上的语句在一个句子，所有的语句必须满足，但需要的东西是虚假的陈述只有一个失败的要求：

```
x < y && y > 5
 /*x是较小的比y和y是大于5*/
```

Note:
This example will only return a value in the range between 200 and 300. It will never return 200 or 300.

• 或者 ||
这是用来确定是否使用的东西或别的东西是真实的。如果只有一个说法是正确的，那么句子将仍然是真实的：

```
x < y || y > 5
  /*x是小于Y或y是比5.Note大：这两个语句可以是真实的*/
```

• 不！
这是用来确定如果是不正确的的东西。如果声明是不符合，那么句话将永远是真实的的：

```
!x==5
  /*x不等于5*/
```

常量

常量代码的Arduino的使用的语言部分，他们已经预定义值。它们被用来使你的程序代码更易于阅读。

注意：
“多”和符号“小于”，可以很容易地混合起来。永远记住，对较小的值的封闭终点。

• True和False
true和false是什么是所谓的布尔常量的定义，如果事情是，或者不是在逻辑层次，

```
boolean myBoolean = true;
```

布尔运算符可以作为任何数。例如，200可作为一个运营商，如果一个变量，这个值和我们比较到200，这将产生一个真正的：

```
int myNbrBoolean = 200;
myNbrBoolean == 200;
  /* this line of code will evaluate to true */
```

也可以写一个数字：

```
int myNbrBoolean = 1;
```

在第一种情况，我们会需要比较myBoolean另一布尔，在第二种情况下，我们需要比较myNbrBoolean另一种诠释。

• 高和低
HIGH和LOW用于设置数字引脚的状态，只有这两个国家。高科技手段，同时为ON或5伏在我们的数字引脚的转折点。这也是作为一个逻辑1的相同。“低”是指同为OFF或有我们的数字引脚为0伏。这也是作为一个逻辑0的相同：

```
digitalWrite(ledPin,HIGH);
```

这也可以书面与数字：

```
digitalWrite(ledPin,1);
```

•输入和输出
输入和输出使用时，我们宣布我们的数字引脚的模式，只有这两个数字引脚模式：

```
pinMode(12,OUTPUT);
```

如果有事做什么

比方说，你工作的原型，你是测量距离。现在的东西一定范围内从你的对象时，你想要的东西发生。这是if语句时就派上用场了。

• 如果
if语句是想测试Arduino的可以做的东西，以确定是否是真或假的。if语句看起来就像下面的例子：

```
if (myVariable>myOtherVariable){
 doSomething;
}
```

在这个例子中，我们提出这样的问题，如果MYVARIABLE是比myOtherVariable大。如果是这样，那么程序将跳转内，如果功能执行的代码。如果说法是不正确的的，它会跳过这部分的代码。在上面的例子中，我们比较的变量，但我们可以比较常量以及：

```
if (buttonPin==HIGH){
 doSomething;
}
```

在这个例子中，我们问的问题如果buttonPin高的doSomething，如果它是不跳过任何内，如果 -

功能。

在这个例子中，我们问的问题如果buttonPin高的doSomething，如果它是不跳过任何内，如果功能。

记住，不要忘记使用==当你进行比较。如果您使用一个单一的=你会不会比较别的东西，但你会申报别的东西的东西的价值：

```
if (buttonPin=HIGH){
 doSomething;
}
```

上面的例子显示了错误的方式写if语句。这个例子声明为高buttonPin并不会检查如果buttonPin是高。 Arduino的程序，这个错误不会提醒您，当您验证您的代码。

•的if else
现在让我们说你使用if语句来检查，你的条件是不正确的的，你知道你想要做的第一件事，如果不发生。然后，您可以连接到您的if语句的else语句：

注意：
编程时，所有Arduinos常数总是写大写字母。

```
if (myVariable>myOtherVariabel){
 doSomething;
 } else {
 doAnotherThing;
}
```

上面的例子，像一个“非此即彼”的声明。无论MYVARIABLE是比myOtherVariable大，你的doSomething或MYVARIABLE不大于myOtherVariable和你doAnotherThing。

不要忘了使用新的标记，其中的else部分的开始和结束的花括号。

您想，您可以添加许多其他条件if语句，但如果你比别人多，你有写除了最后一个作为其他所有，如果一个新的条件：

```
if (myVariable>myOtherVariable){
 doSomething;
 } else if (myVariable<100){
 doAnotherThing;
 } else {
 doTheLastThing;
}
```

如果在上面的例子MYVARIABLE是不是更大然后myOtherVariable然后问，如果MYVARIABLE是小于100，如果这是不正确的，要么，然后doTheLastThing。

• 对于
for循环使用，当你想重复一定次数的代码的一部分。 for循环总是有括号中的三个部分，它是计数器的初始化，条件结束for循环和增加你的计数器：

```
for (int i=0; i<200; i++){
 doSomething;
}
```

在这个例子中INT I = 0; for循环计数器初始化。在这里，我们说，我们希望有一个整数类型的名字，我的柜台，我们希望它从0开始计数。当我们在做循环的第一部分，我们以分号结束，这为第二部分，以及。第二部分是i<200，这是我们的条件，结束for循环，在我们的例子中，这意味着当我变得大于或等于200，我们要停止循环。最后一部分是我们增加计数器每次循环开始。 I ++将增加一个计数器，每次循环重新启动。计数器的第一次，我会为0，下一次，这将是1，依此类推。当我打200的声明，我<200会评估为false，结束和条件的循环将会见并计划将退出循环，标志着循环结束后的方括号，写的代码进行。

注意：
如果总是小写字母拼写与编写代码时。不要忘了大括号标记的开始和结束，如果功能。

• 虽然
一个while循环将继续循环，直到其括号内的条件变得虚假。如果用一个while循环是有while循环内的东西，可以增加或改变，否则将永远不会结束while循环：

```
while (myVariable<100){
 doSomething;
}
```

这个例子的测试，如果MYVARIABLE是小于100。如果是的话，那么它会开始循环。但有没有什么改变，因为我们第一次检查，所以这个循环将继续永远的doSomething MYVARIABLE内循环。

在下面的例子中，我们添加了一个传感器读数：

```
while (myVariable<100){
 doSomething;
 myVariable = readSensor;
}
```

现在每次循环它会先的doSomething，然后保存MYVARIABLE readSensor价值。如果保存在MYVARIABLE值高于100，while循环将停止和开展与循环结束后的方括号，标志着编写的代码。请注意，如果您正在使用的传感器打破一个while循环确保传感器会给你一个值，你的条件在while循环的门槛以上。

数字引脚

这些引脚Arduino的0至13引脚和他们所谓的数字，因为它们只能处理为0或1的信息。如果你想使用数字引脚，你必须做的第一件事是设置引脚的模式。这始终是在虚空中设置（）。

设置引脚模式的命令是pinMode（）如下：

```
pinMode(pin,OUTPUT);
```

在这个例子中的“针”是您的Arduino板的物理引脚旁边的数字对应的值的变量。输出是您的PIN所需的模式。数字引脚只有两种模式，输出和输入。如果你声明一个引脚为输出，你可以只使用它可

以打开的针5V或关闭为0V。如果你声明为输入您的PIN，您可以只使用它来读取是否有5V的进销或有0V上

引脚：

```
digitalWrite(pin,value);
```

要打开你的数字引脚和关闭，你需要使用digitalWrite（）命令。括号中的你总是需要国家什么针，你要使用什么样的价值，你想给它：

```
digitalWrite(pin,HIGH);
```

这将打开较多的引脚，这将使它能够发送5V。如果你写的，而不是高低，你会变成引脚为0V。请注意，直到你把你的数字引脚低后高的默认值设置引脚模式。如果你有你的Arduino板的看，你还会看到，0和1的数字引脚RX和TX的标记。这两个引脚用于串行通信和保留，不应使用，因为它会在待机模式下，直到收到信号的Arduino的。

•DigitalRead（PIN）
digitalRead（）命令读取引脚的状态，并返回一个高，如果有5V，如果有0V引脚上的进销或低：

```
digitalRead(pin);
```

为了能够使用的东西，你会需要它保存在一个变量状态：

```
myVariable = digitalRead(pin);
```

如果你想作一比较，你可以写命令，直接在一份声明中：

```
if (digitalRead(pin)==LOW){
 doSomething;
}
```

虽然低一直是0V，数字输出数字输入，0V和1.5V之间的每一个值将作为一个digitalRead低读。大约3.3V和5V之间的所有值相同的方式将阅读作为一种高价值。

模拟引脚

模拟和数字引脚的工作方式不同。我们提到过，只能处理数字引脚1或0，这是相同的HIGH和LOW或0V和5V的信息。然而，在现实世界中，我们没有测量在0和1的一切，让Arduino的有六个特别，使一个数学计算的电压范围为0到1023的模拟引脚。模拟引脚没有被宣布模式，因为他们只用作输入。

• 模拟阅读（PIN）
要读一个模拟引脚上的价值，你必须使用analogRead（）命令，并参考针，你想读：

```
analogRead(pin);
```

作为数字引脚，你必须到这个值可以使用它保存在一个变量

```
myVariable = analogRead(pin);
```

您可以使用该命令直接作出比较

```
if (analogRead(pin)>500){
 doSomething;
}
```

• 模拟写（针，价值）
数字引脚只能是高与低，这是相同的，要么有数字引脚的5V或0V。但数字引脚3，5，6，9，10和11，我们有一个特殊的功能，称为analogWrite（）。有了这个功能，它可以发送这些特殊的数字引脚的伪模拟值。这也被称为脉冲宽度调制（PWM）：

```
analogWrite(pin,value);
```


在这个例子中的值可以是任何从0到255。如果你写0，这将是相同的引脚设置为低，和255一样高一样。但你analogWrite（）255高低的步骤，因此，例如：

```
analogWrite(pin,127);
```

这将类似于发送您的数字引脚2.5V。相比digitalWrite（）转移在瞬间从0V到5V的analogWrite（），你可以从0V到5V较慢的过渡。请注意，analogWrite（）只标有PWM（3，5，6，9，10，和11）的数字引脚，而不是模拟输入引脚A0到A5。

注意：
如果您使用的是一个引脚为输入确认信号是从来没有超过5V或你会燃烧你的Arduino板。

使用时间

Arduino是一个小，但功能强大的计算机，并能进行100万次计算。当你正在原型，你可能不希望在这闪电般的速度执行。然后你要告诉Arduino的放慢每一个现在，然后。

•延迟
延迟命令是用来设置在你的程序暂停。此命令以毫秒为单位计数，你在下面的例子括号内输入所需的暂停时间：

```
delay(1000);
```

这种延迟将设置在一秒方案暂停。

• 数毫秒
此命令将返回已经过去了多少毫秒Arduino的开始，目前正在运行的程序。为了能够使用这个值，你必须保存在一个变量：

```
myVariable = millis();
```

您可以直接使用它，使时间的比较：

```
if (myAlarmTime == millis()){
 ringAlarm;
}
```

与其他设备通讯

为了能够与其他电子设备，你必须使你的Arduino的通信端口通信。 Arduino的可以与您的计算机和其他电子设备使用的串行通信协议进行通信。

与其他设备的串行通信和数字引脚0和1的Arduino的保留，你应该避免使用这两个端口为别的，因为这可能会干扰你的程序运行。

• 串行开始
为了使Arduino的沟通，您使用的命令Serial.begin（）。始终使用此命令是无效的setUp（）和其他地方。括号内的比特每秒，这也是为波特率，并可用速度是300，1200，2400，4800，9600，14400，19200，28800，38400，57600或115200输入您想要的沟通：

```
void setup(){
   Serial.begin(9600);
  }
```

注意：
如果你正在与模拟读数进行比较，从模拟引脚的价值绝不可能在1023以上。

这将打开串口和设置的通信速率为9600波特。

• Serial println
此命令将打印无论你在括号内添加一行结束。要打印你必须输入括号内的整数：

```
Serial.println(12345);
```

但是，打印字符和一个字符串需要被引用。如果你要打印单个字符使用单引号‘ ‘:

```
Serial.println('C');
```

上面的代码行，将通过串口发送字符C。如果你要打印一个字符串，像一条消息，你要正常使用引号“ “:

```
Serial.println("Hello from Arduino");
```

• Serial print
Serial.print（）工程Serial.println（）是自动回车和换行的异常Serial.println相同。如果事情是通过串口（1）与Serial.println发送消息将显示在您的显示器如下：

```
1
1
1
1
1
```

Serial.println（）添加换行总是推进入每次你送东西，所以它结束了一个新行是相同的。如果我们将通过串口将收到发送Serial.print（1）：

```
1111111111111111111111111111111111111111
```

请注意，如果你是沟通与其他电子设备确保您打印您的数据在正确的格式。有时Serial.println（）添加回车和换行可以干扰与沟通。该设备的数据表中可以找到有关个人电子设备的正确的通信协议的信息。

• 特殊情况下印刷
有时你会需要以不同的格式发送信息，然后你有这种格式添加到您的打印：

```
Serial.println(message,format);
```

可用的格式是小数，十六进制，八进制，二进制和字节，并在下面的例子：

```
Serial.print(b,DEC);
  /*这将打印作为ASCII编码的十进制79
 “79” */
Serial.print(b,HEX);
  /*这将打印ASCII编码的十六进制79
这是“4F” */
```

注意：
小写的b，它是在这些例子中使用的ASCII值“79”。

注意：
米里斯（）的值将重置为0后，约9小时。

```
Serial.print(b,OCT);
  /* 这将打印一个ASCII编码的八进制79
 “117” */
Serial.print(b,BIN);
  /*这将打印ASCII编码的二进制代码79
这是“1001111” */
Serial.print(b,BYTE);
  /*这将打印ASCII编码的字节，这是79
 “O” */
```

ASCII（美国信息交换标准码）编码是一个数字编码的文本标准化的方式。当你写你的代码中的数字，它实际上不是你写的数字，但数字的文本表示，就像“一”是1个字表示。

尾声

我们现在已经达到了这本书，希望你自己的未来在时尚和耐磨原型开始的结束。我们希望，这给了你可穿戴计算的世界，一个基本的洞察力，在本书的例子，将作为您在自己的领域取得进展的基础。然而，在自己的这本书原型直过渡可能很难。但幸运的是，你并不孤单，在世界电子原型。正如我们在这本书的开头说，Arduino是不是唯一的硬件和软件。这也是一个巨大的社会人士，专业人士和爱好者的兴趣在物理原型。

大部份的这些人走到了一起在Arduino的操场（www.arduino.cc/游乐场），他们既有利于相互的各种问题，或者只是分享想法和炫耀的新建筑。

如何得到你想要的（www.kobakant.at/ DIY）米卡里美和汉娜Perner威尔逊是可穿戴技术技术最好的资源联机。在这个网站上，你可以找到项目的例子和软电路，导电和电阻面料，导电纱线和工具提示的信息。

Instructables（www.instructables.com）是另一个伟大的网上社区的DIY（自己动手）人。用户不仅创建自己的电子自助导游，但也“怎么做”上几乎任何主题，您可以想到的。 Instructables是最好和最大的灵感来源之一，我们可以真诚地把它推荐给任何一个感兴趣的时尚和可穿戴计算。

另外一个很好的竞争者，以Instructables是博客和社区（www.makezine.com）也有一个大集团和技能的人绑，是一个伟大的资源，让你对什么是原型领域发生的更新。

如果您正在寻找社区严格致力于时尚与科技领域的再塑造科技博客（www.fashioningtech.com）是一个很好的开始。他们也有一个论坛的用户越来越多的社会，他们交流思想，互相帮助。

即使一本书作为一个良好的开端，并介绍的服务，它绝不会击败互联网广泛和最新的信息。这是我们希望这本书会给你需要的知识能够使你可以在网上找到的所有资料的使用。

有更加良好的网上资源，他们在这本书中，包括所有可能需要一些额外的章节。上面的例子都是很好的出发点为自己的网上信息库。

电子原型时，你应该记住的最重要的是有情趣，不要害怕尝试的事情了。但要知道，你是用电量的工作，因此，如果您对某些项目不确定，遵循简单的原则，做一些适当的研究，直到你有信心，然后才插件的东西放在一起，它可能会节省您的钱。

安全，而不是愚蠢的，快乐的原型。

致谢

这本书是布赖恩埃文斯的小册子Arduino的启发

这本书的灵感来自于布赖恩埃文斯的小册子Arduino的编程笔记本，第一主编，2007年开始使用Arduino的，三版，2008年开始由Massimo板仔。

也感谢在马尔默大学，艺术，文化和传播学院（K3），在关键设计工作室1scale1，马尔默和为他们的工作与Arduino的Arduino的球队特别感谢球员：汤姆Igoe物理样机实验室，戴夫Mellis，大卫Cuartielles，马西莫板仔和赞布罗塔马蒂诺。

INDEX

开放SOFTWEAR.TM - 第二版
托尼奥尔森，大卫加埃塔诺，乔纳斯Odhner和大力士Wiklund。

责任编辑 : 托尼奥尔森。
制作编辑 : 乔纳斯Odhner。
校对 : 梅利莎科尔曼，安德烈亚斯Jiras和德里克科尔曼。

平面设计师 : 大卫Gaetano和乔纳斯Odhner。
插画 : 大卫Gaetano和乔纳斯Odhner。

字体 : Charis SIL（标题和正文）
Titillium文本（注）
OCR的一个标准（代码exampels）

纸张 : Edixion偏移量为120克。

打印 : JMS Mediasystem，Vellinge，瑞典，2011。

历史 : 2009年7月第一版，（网上PDF格式出版）。
第二版，2011年4月。

ISBN : 978-91-979554-0-9

网址 : www.softwear.cc

由粉红男孩出版社出版。 www.blushingboy.com