

构建线性平移台的分步指南

1. 打印3D打印组件
2. 步进电机安装座和底座
3. 轴夹
4. Attach feet
5. 立柱和侧面部件
6. 终点挡块
7. 顶部框架和光学夹具
8. 上轴夹
9. 安装步进电机，轴联轴器和螺杆
10. Leadnut
11. 钢螺杆及其对齐
12. 轴承座
13. 平台框架
14. 平台末端部件
15. 安装平台
16. 连接步进电机
17. 传感器和输入
18. 软件

材料清单

构建线性平移台的部件主要来自两家供应商：TechnobotsOnline.com和Adafruit.com。因为许多零件都是标准的紧固件，所以你可能已经有了许多螺丝，螺丝和螺母以及一些电子设备。许多的零件在Amazon和Ebay上便宜可得，包括在BoM中。以下是每个项目的产品页面的链接。注意链接最后更新于29/08/13。

这里有一些关于我们使用的供应商的有用的链接

步骤一 ——打印3D打印组件

当我们建立我们的原型显微镜时，我们有幸能够访问MakerBot Replicator 2X 3D打印机。这是一个很棒的资源，因为它使我们能够非常快地创建复杂和定制的零件。大多数台式打印机可以使用ABS或PVC塑料，两者都是刚性和通用材料。

如果您无法访问3D打印机，您可以将设计发送给3D打印机公司，他们可以生产你的零件，每个项目只需50p。一些线上公司提供此服务，它们中有：

<https://www.ponoko.com/>

<http://i.materialise.com/>

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

<http://www.sculpteo.com/en/>

所有3D打印件的.STL文件都可以在OpenLab Tools Github的存储库上找到。

3D打印过程相当缓慢，其中一些部分可能需要一个小时才能打印完，所以当你的3D部件正在打印时，你可以继续下一部分的工作！



2 x终端支架



光学夹具 (可选)



铅螺母安装座- 顶部



铅螺母安装座- 底部



左轴承座- 上方



左轴承座- 下方



右轴承座- 上方



右轴承座- 下方



2 x前平台安装座



4 x M3间隔

步骤二 —— 步进电机安装座和底座

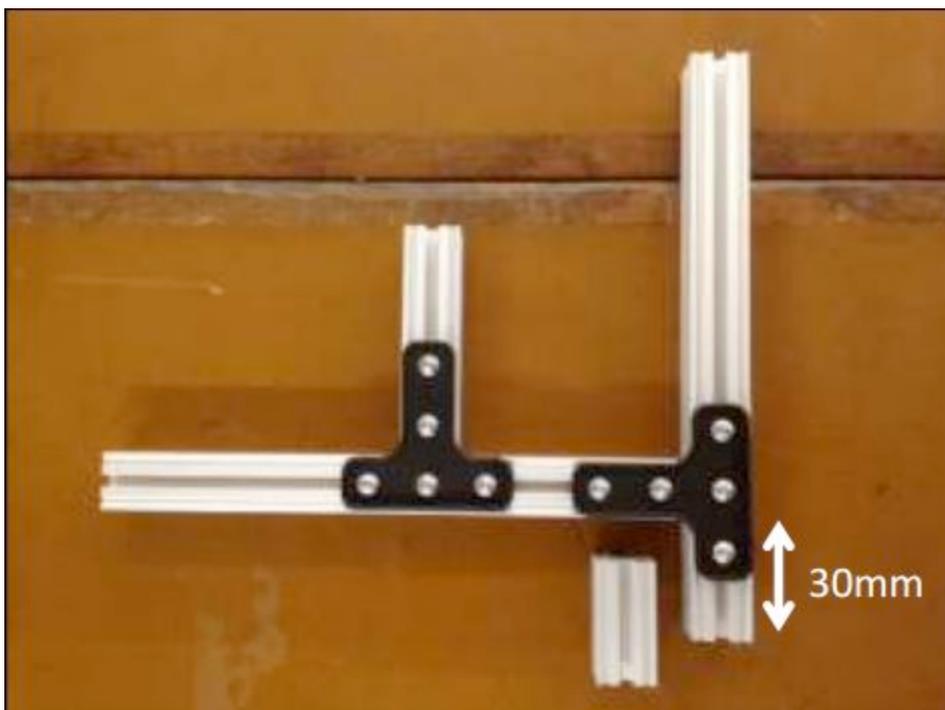
首先将6mm的按钮头螺丝穿过OpenBeam的T型支架的2个孔，并将螺母固定在另一侧，在螺母和支架之间留下2或3mm使螺母可以滑入铝挤压。如果你缺少螺丝则可以省略中心螺钉，只要每个梁段连接到支架至少有2个螺丝



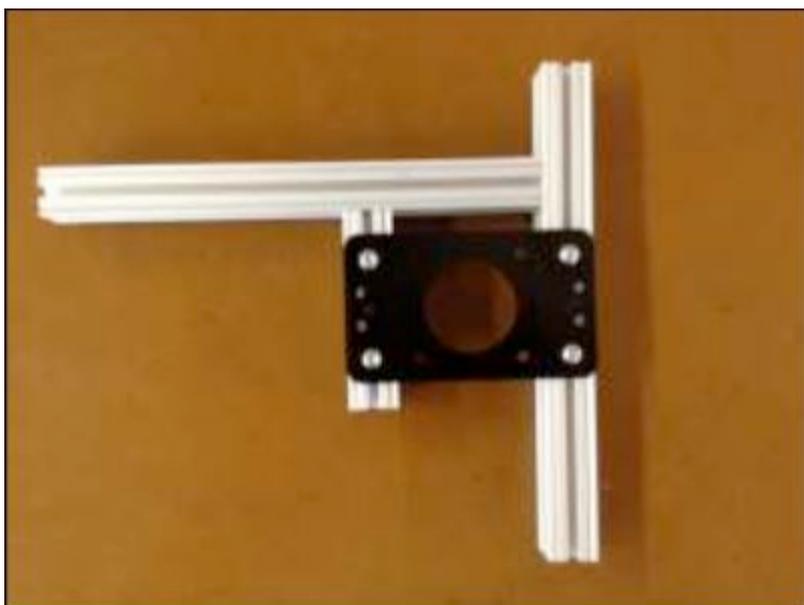
将“T”的顶部滑入一个150mm铝合金部分，将底部滑入60mm部分。



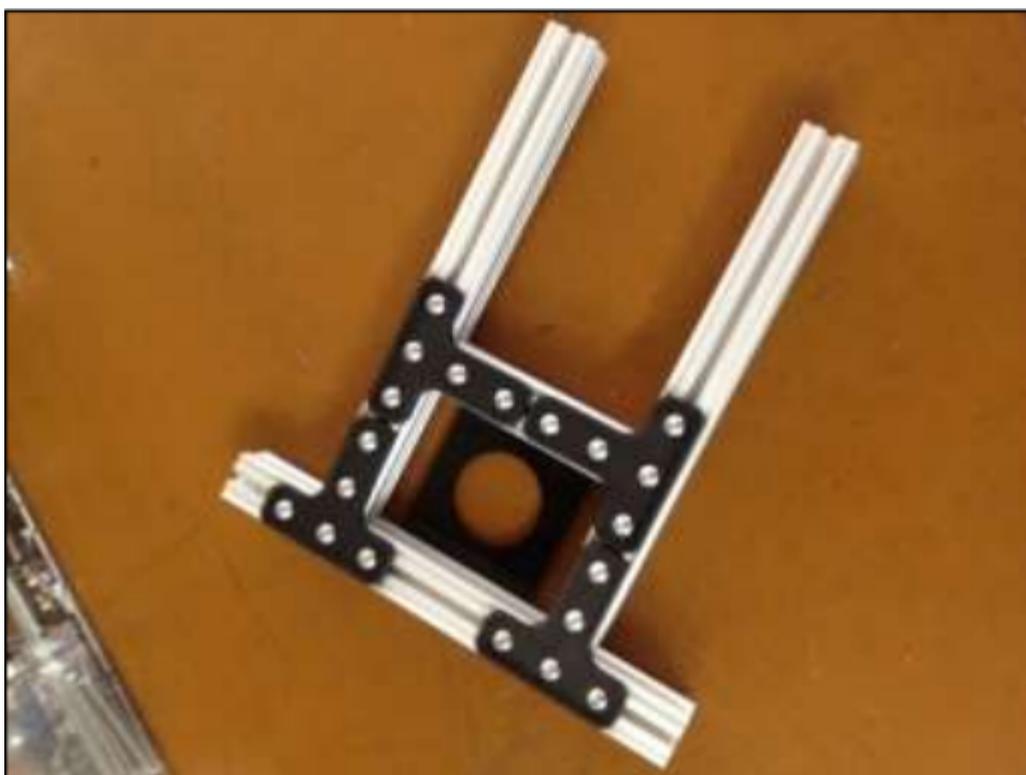
拧紧60mm部分的螺栓，但暂时将螺栓放在“T”的顶部保持松动 - 这样可以更容易地安装电机座。



使用T型支架将第二个150mm的部分垂直于第一个。新件的末端与原件的一侧之间应该有30mm的间隙，如图所示。如果你拿起一个OpenBeam入门工具包，你可以使用30mm的零件来测量。



现在翻转装配件。将螺钉插入步进器的四个最外面的孔中电机安装座，并在下侧松动地安装螺丝。将安装板滑入平行的60mm和150mm部分，使得安装座居中有在60mm部分。



将组件翻转到带有支架的一侧，并用开关侧关闭另一个150mm部分。检查新件的一侧与端部之间的间隙的垂直片是30mm。

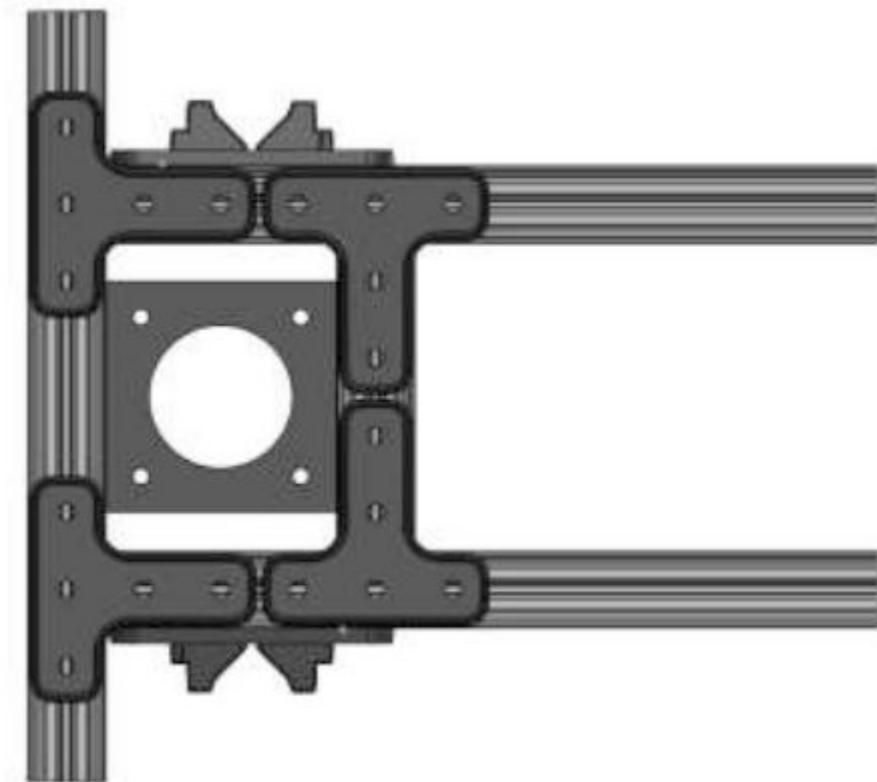
在该步骤中：

步骤三 ——轴夹

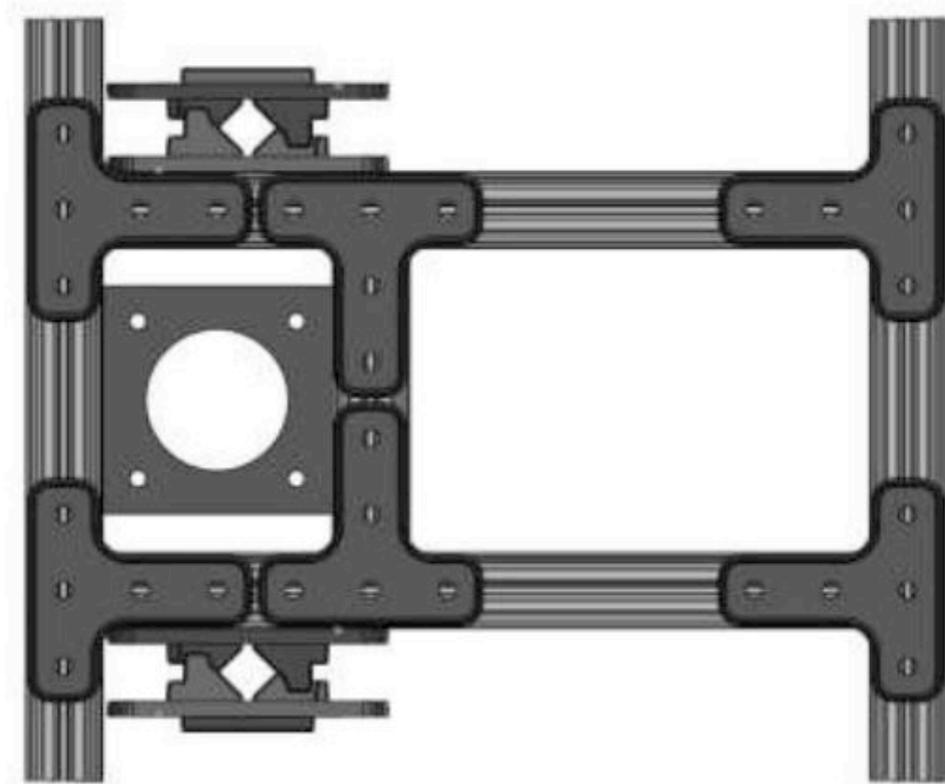
OpenBeam轴夹由两个相互啮合的相同部件组成。如果你从Technobots购买了一个轴夹套件，其中会包括其中4个，足以构建2个轴夹。这些套件还包含两种不同类型的螺钉：25mm和35mm M3六角头螺钉。我们将使用25mm螺丝。



进入两个最外面的孔，插入6mm螺丝和螺母，螺母在三角形的牙齿相反的一边。现在从半夹的背面，插入25mm螺丝进入内孔，使得六角头与六角螺母对齐。



将一个半夹子滑到每个150mm的梁上，将它们推到该部分的右边这样他们只是碰到后面的部分。一旦到位，收紧按钮头螺丝，并将每个夹具的另一半插入25mm螺丝。使用M3垫圈和螺栓固定，但保持夹具松动。



最后，底座可以通过用最终的150mm Al截面关闭底座来完成。如之前，底座必须完全对称以确保任一侧都有30mm间隙。

在该步骤中：

步骤四——

在这个步骤中，塑料OpenBeam脚连接到将形成结构的垂直部分的300mm部分。这一步是可选的 - 塑料脚有助于在刮擦工作表面时保护铝，但您可能希望省略或用减震橡胶脚代替他们。



通过将M3螺丝敲入末端的孔将脚连接到300mm铝合金部分的部分。这应该是最理想的使用M3攻丝工具。例外情况，如果你手边没有其中一个，铝足够软可以直接拧入螺栓金属，但要非常小心，不要损坏正在使用的螺丝头或螺丝刀。

一旦孔被轻敲，用6mm M3螺丝将脚拧入到位。

在该步骤中

步骤五——立柱和侧面部件

将300mm部分中的一个连接到现有基础结构的每个角的脚。确保脚位于电机安装板的相对侧！脚应在底座框架底部以下突出

30mm, 以允许电机在下面连接。

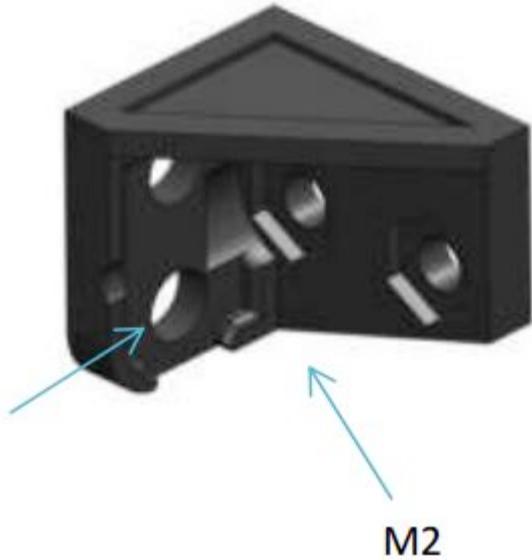


您现在可以通过连接150mm A1部分到底部带有L形括号的开放侧面来完成装配件的下部。

在该步骤中：

步骤六——终点挡块

终点挡块是防止平台超出其最大范围的微动开关。 我会建议添加终点挡块在这个平台上，而不是与的其余的电子组件，因为它可以非常巧妙地适应微型开关。

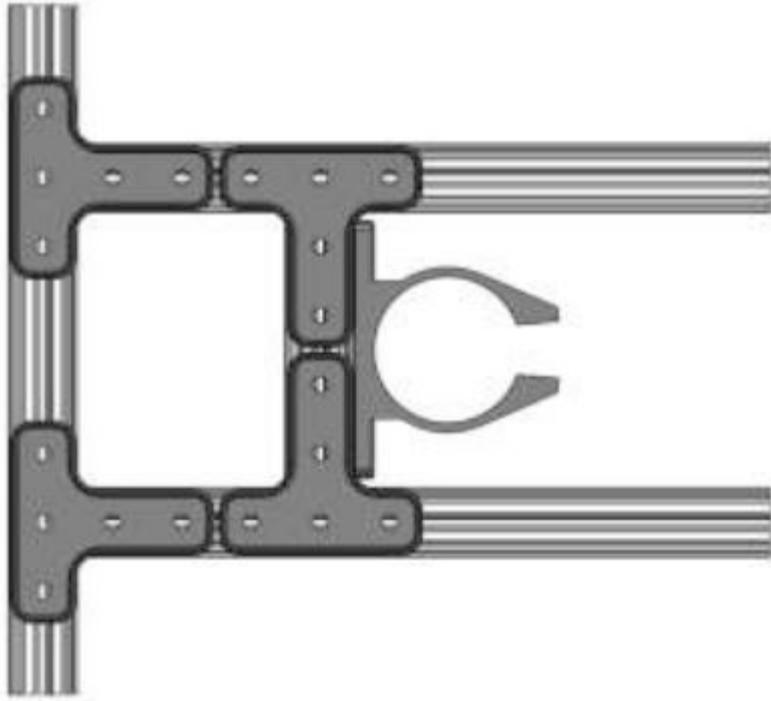


安装座有两个4mm孔用于连接OpenBeam和M2的两个微动开关螺丝的3mm孔，内部可能有点过于复杂，但六角孔使其更适合M2螺母。首先将长的电线焊接在每个微动开关的公共端（C）上常闭（NC）引脚。然后把微动开关拧在安装座上一个朝上，另一个向下。然后将终点挡块以通常的方式连接到其中一个垂直的OpenBeams，连同朝向前方突出的开关。将终点挡块按照所需的距离分开并收紧M3螺丝

在该步骤中：

步骤七——顶部框架和光学夹具

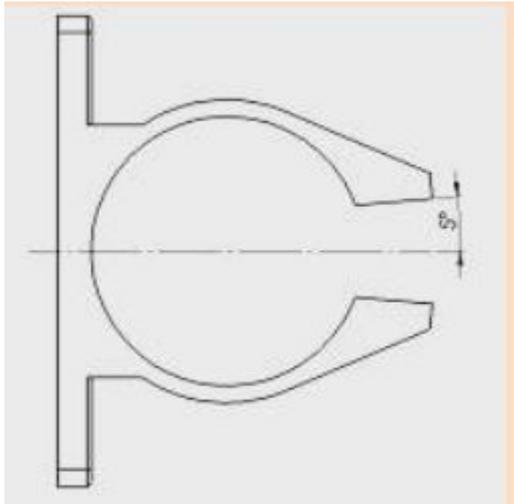
顶部框架与底座框架重复，只是电机板略去。像之前一样从连接60mm和150mm开始。再添加150mm截面垂直于第一个，再次仔细从第一个侧面一块到第二个末端测量出30mm。



这次，不再是连接电机安装座，而用4 6mm M3螺丝固定光学夹具。继续如前所述附加第三个150mm部分。现在距离60mm部分多远并不重要，我们将在稍后进行调整。检查光学夹具居中，并且该块是好的和对称的，在任一个间隙侧有30mm。

设计指导：

光学夹具设计用于固定25mm CMount管。夹具需要既坚固又牢固灵活 - 2mm壁厚就足够了。注意螺栓配合的突出部分向后倾斜略微（中心偏离5度）。这意味着在夹紧的位置是平行的以使螺栓平坦。我在右侧标签上添加了一个六角形孔防止螺母在其插座中转动并且拧紧螺丝更容易。



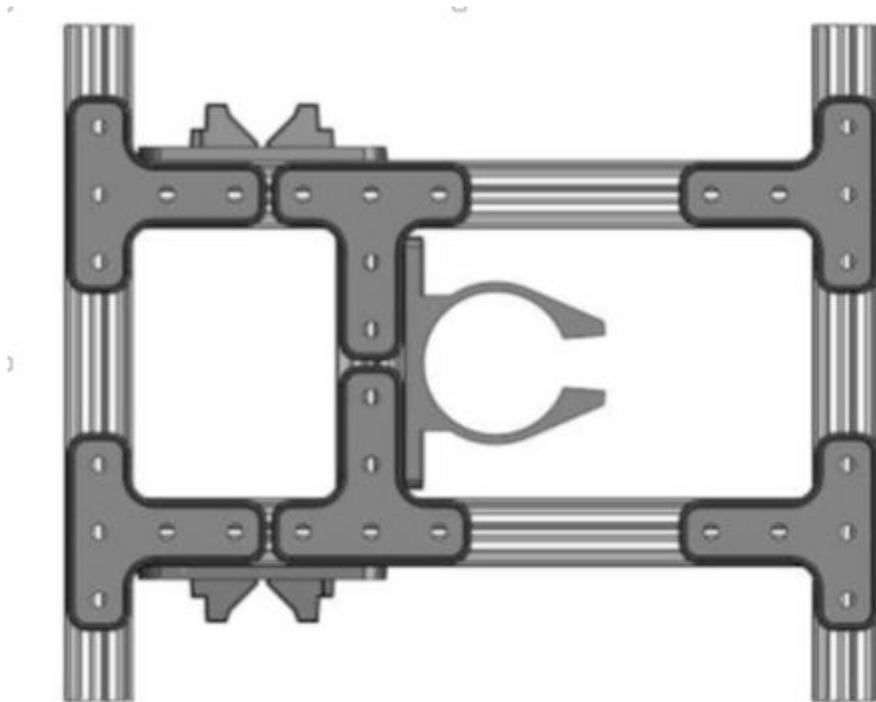
在这一步中：

- 3 OpenBeam 150mm 铝合金
- 1 OpenBeam 60mm 铝合金
- 4 OpenBeam T型支架
- 1 3D打印光学夹具
- 24 M3 6mm按钮头螺丝
- 24 M3六角螺母

第八步——上轴夹

重复步骤3中的说明，将一个轴夹片滑动到每一侧，使其接触框架背面的铝。对于这一步骤，我们将离开夹具的另一半，因此我们可以将轴承添加到轴上。

关闭顶部框架，一个150mm的部分，保持从任一端测量均为30mm。



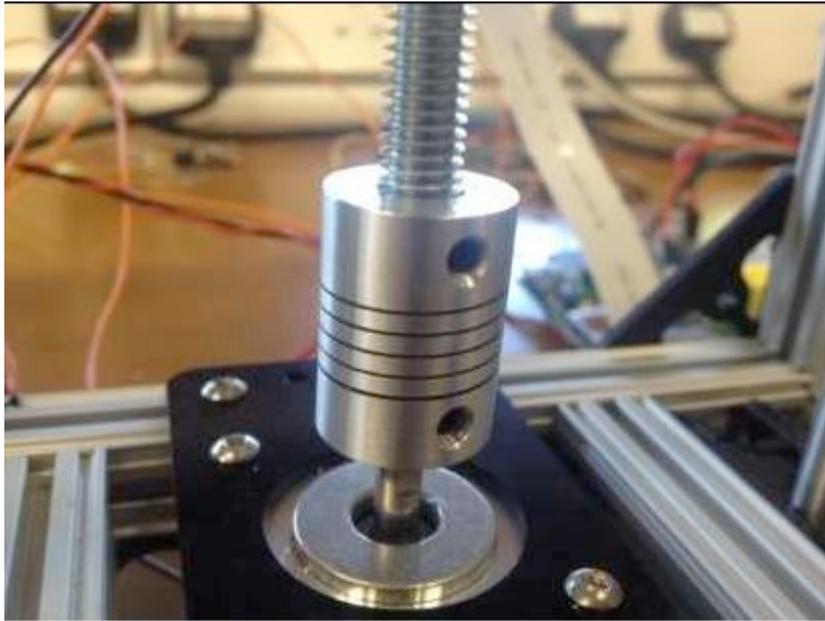
您现在可以使用L型支架将顶部框架连接到基础结构的垂直列顶部。
在这一步中：

- 1 OpenBeam 150mm铝合金
- 2 OpenBeam半轴夹
- 4 M3六角头25mm螺丝
- 2 OpenBeam T型支架
- 14 M3 6mm按钮头螺丝
- 14 M3六角螺母

第9步 步进机，联轴器和导杆

现在支撑结构已经基本完成，机械驱动系统可以放进去了。在安装步进机之前，你可以通过焊接额外长度的电线来延长导线。

将马达用M3 6mm螺丝将固定到步进机底座中。



把M8螺杆插到可调节联轴器8mm钻孔中，用小的2mm allen扳手收紧螺丝。5mm钻孔末端可以插进去步进机的轴，然后拧紧。

、

设计指导

注意螺杆不被固定在顶部，相比于马达轴有相当大的一部分自由度来活动，这样做的目的是导杆可以保持与螺母保持近似垂直。添加一个约束在顶部水平方向将会过度限制系统，也会增加不必要的力量在螺丝上。

在这一步中

4 M3 6mm纽扣帽螺母

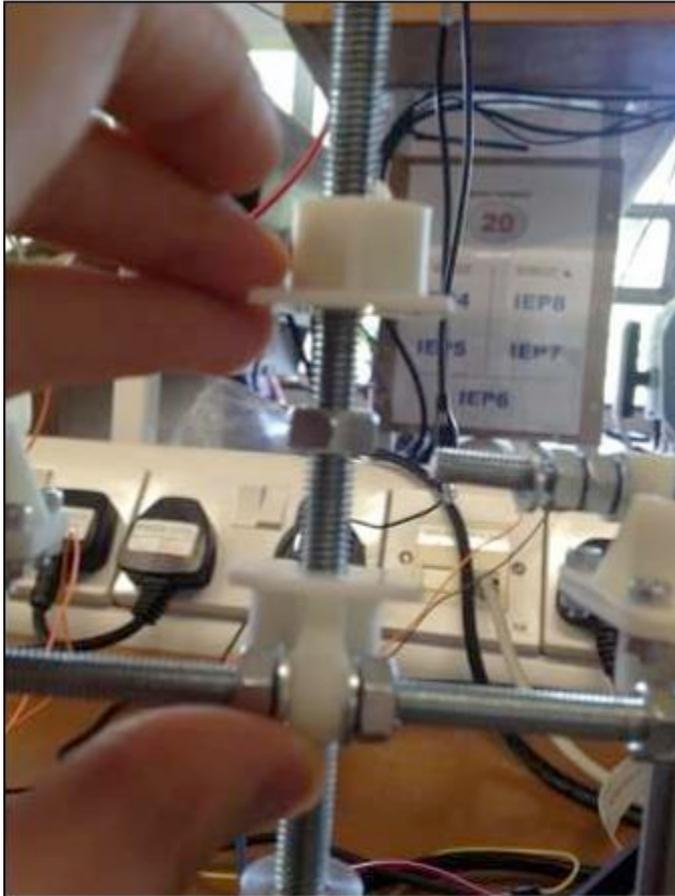
1 Adafruit 12V 步进机 200步/转

1 铝制灵活联轴器

1 M8 300mm螺杆

步骤10 leadnut

丝杠的基本原理是带固定螺丝的旋转轴。螺杆的旋转使螺母上下移动轴。在OpenLabTools平台上，螺母的旋转受到其内安装了六角孔的定制3D底座而固定。



首先把3D打印的引导螺母底座的下半部分放在轴上。底部与顶部由一个圆形的孔区分开。

然后把M8螺母拧到螺纹杆上，再把上半部分装上，将孔排成一排，最后用M3螺栓固定。

设计指导

配套的六角形凹槽孔参数如下

边到边的长度 14mm

对角线长度 16.17mm

厚度 7.2mm

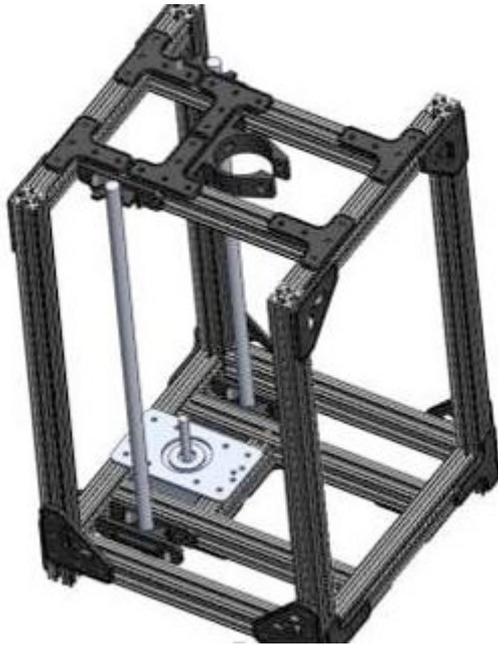
在这一步骤中

- 1 3D打印螺母底座的上半部分
- 1 3D打印螺母底座的下半部分
- 1 M8六边形螺母
- 4 M3 8mm带帽螺钉
- 4 M3 六边形螺母
- 4 M3 垫圈

第11步 钢棒

下一个要提到的此次机械设计的部分是光滑的钢棒。这些是为了防止大螺母相对于平台发生转动。OpenlabTools使用的轴是8mm硬化钢，允许线性的轴承上下运动。

将钢棒插入到底部的夹子中固定好，为了让轴承便于添加上来我们使上方的夹子保持开放状态。



在这一步中

- 2 8mm 银色的钢棒 330mm

第12步 轴承

线性平台使用LM8UU适合3D打印出来模型的线性轴承。轴承的主

要功能是防止核心螺母旋转和保持平台高度。8mm是为了适配轴的大小，如果半径为6mm的轴则要选择LM3UU。

设计指导

LM8UU线性螺纹螺母是15mm×8mm×24mm（外直径×内直径×长度）

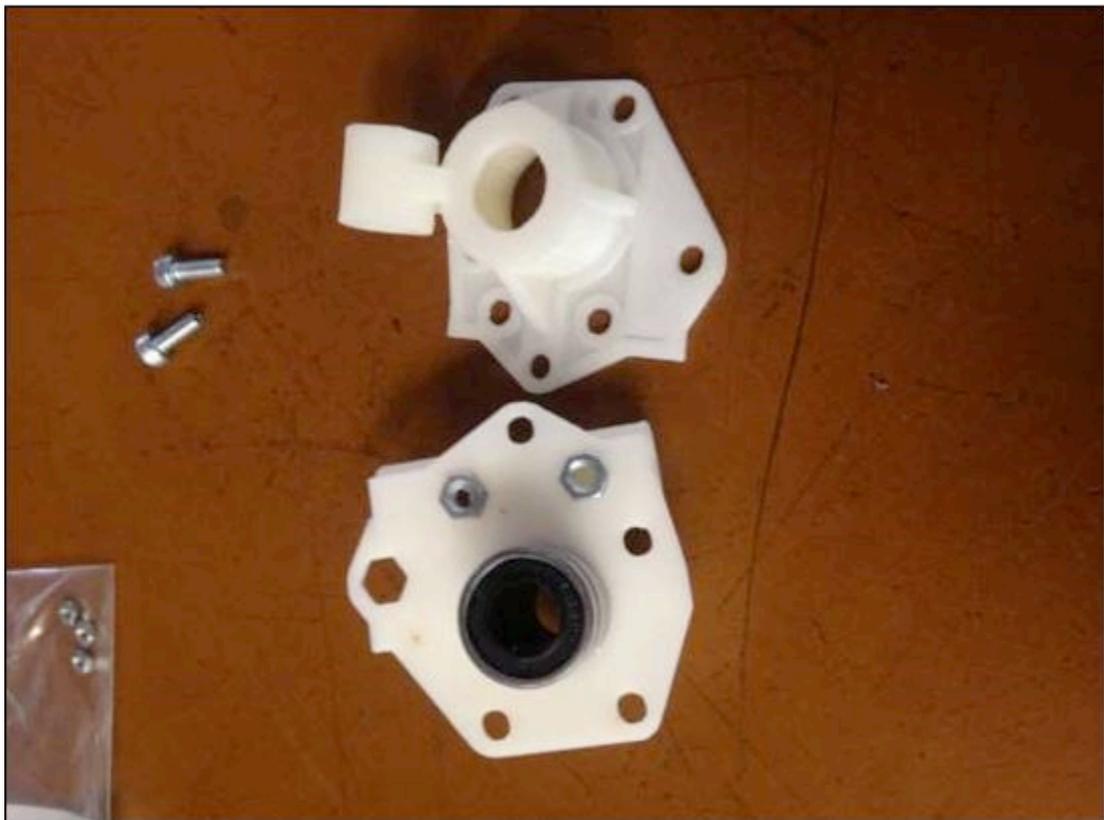
打印的螺母底座的参数

内直径 15.5mm

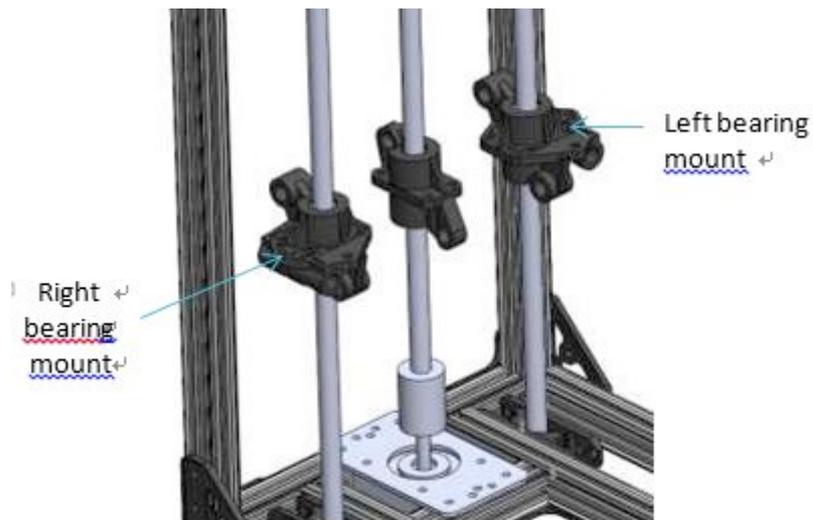
长度 26mm

这能使螺母紧紧的包裹在里面且不发生移动。

将LM8UU放入到螺母座上半部分和下半部分之间然后用M3螺栓螺母固定来完成装配螺母底座。



将轴承插入到钢棒上。这一步需要非常小心因为你很容易就会将LM8UU中的球轴承弹出。“左”和“右”是组装背面所看到的，所以长的平行管应该在外面。



你现在可以用螺母和垫圈夹住轴的上半部分了。这是检查轴是否完全平行，最好用垂线确保轴是垂直于地面的。



用150mm的open横梁组件以及L板将装置的上半部分封闭起来完成结构的构建。

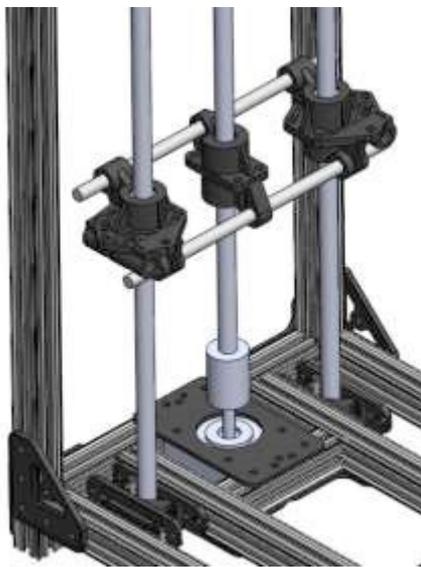
这一步中

- 2 LM8UU 线性轴承
- 1 3 D打印左侧轴承座上半部分
- 1 3 D打印左侧轴承座下半部分
- 1 3 D打印右侧轴承座上半部分
- 1 3 D打印右侧轴承座下半部分
- 4 M3 8mm带帽螺丝
- 2 Open横梁半轴夹

- 2 Open横梁150mm 铝压材料
- 4 Open L型托架
- 20 M3 6mm纽扣头螺丝
- 32 M3 六角形螺母
- 12 M3 垫片

第13步 平台框架

由于平台将仅由组装背面可以支撑，因此需要一个刚性框架来支撑载物台和照明或其他附件。



第一阶段是将螺母安装座和轴承座连接在一起，这样可以防止螺母旋转。将两个150mm螺纹杆穿过安装座的上下孔，并用M6垫圈和螺母固定到位。这个需要非常精确：首先将螺杆穿过两个孔，加上两个螺母和垫圈，然后拧紧；将螺纹杆穿过第三个孔并从第一个孔中移出，这样您就可以有足够的空间将两个螺母加在相对的一侧。然后将螺杆滑回中心，加上外螺母。然后对另一个螺杆重复。

将M6螺杆穿过轴承底座的下半部分管道，按上螺母和线圈。确保螺杆之间平行。

设计指导

当设计轴承底座时，我想给平台添加一个X-Y方向移动就像Z轴那样

来移动载物台。底座下半部分的管可以因此而设计成像螺杆那样的6mm平滑杆。如果你想这么做，你可以添加两个M3六边形螺母放在底座上的六边形凹槽。这样能使螺丝从上部拧进固定光滑杆。

这一步中

4 M6螺杆150mm

10 M6 六边形螺母

10 M6 垫片

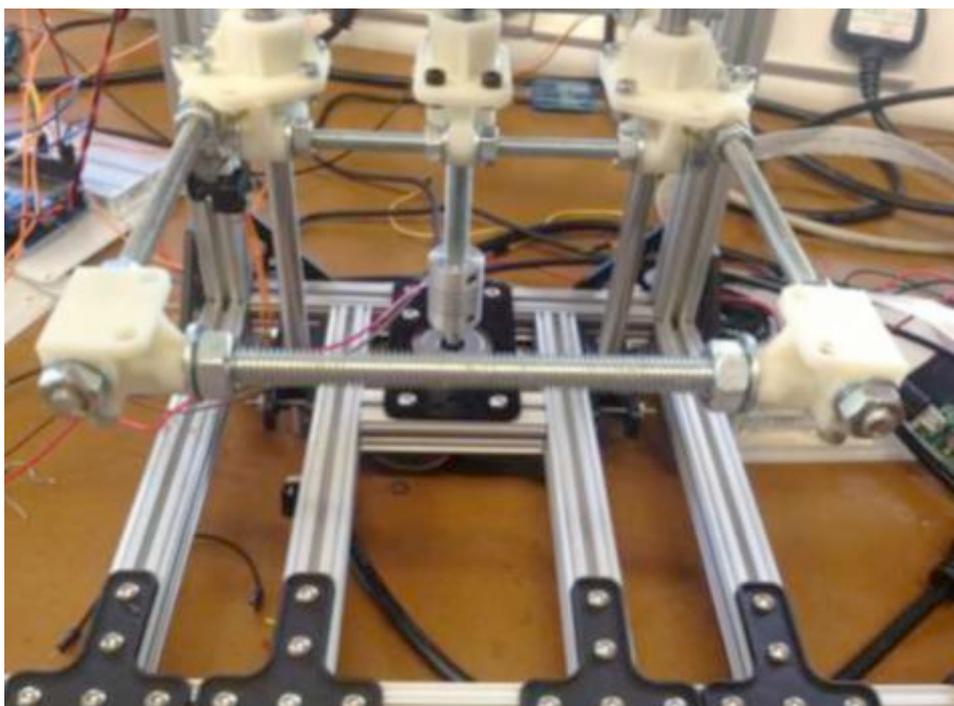
第14步——台子末端零件

首先将M6螺母和垫圈添加到突出的螺纹杆上

设计指导：

在设计台子末端时，我们在丝杠上留下了几块M8螺纹杆，我将它们设计为既适合120mm M8的杆又适合150mm M6的杆。尽管我没有机会尝试用过M6杆，但你可以同时使用它们。





如果使用M8杆，将两个M8螺母拧到杆上，并将垫圈加到外面。插入桌子上的9mm孔中，然后将桌子安装架滑到现有的位上M6螺纹杆位上。

如果使用M6螺纹杆，请将M6螺母插入每个安装座的六角孔中。线杆通过第一安装座直到杆的端部处于安装座的端部附近，然后添加两个螺母，垫圈到杆的另一端。将另一个安装座螺纹连接到螺纹杆上，并调整两个安装座，直到与螺杆对齐附在台子上。

在这一步骤中：

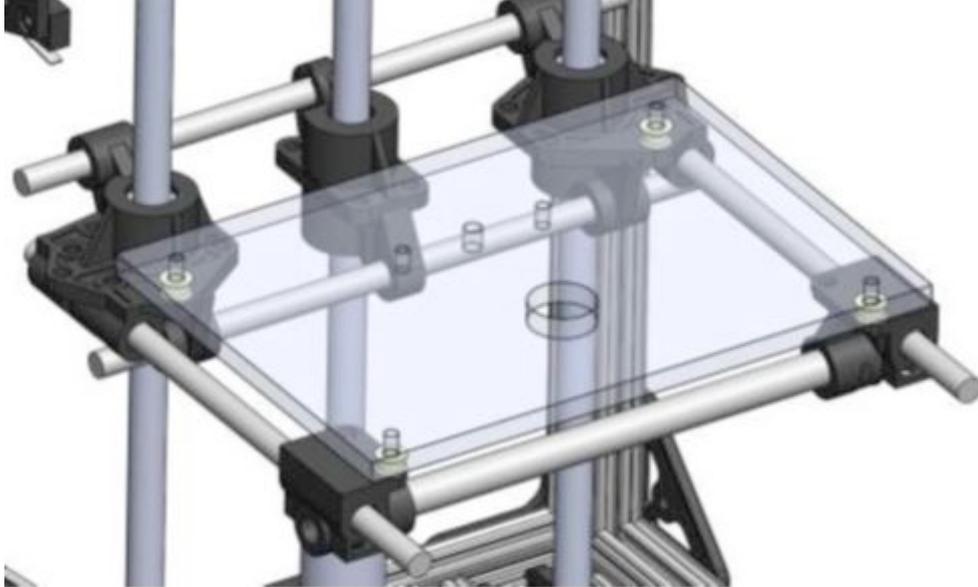
2 3D打印载物台末端零件

输入

- 1 M8螺纹杆120mm
 - 2 M8六角螺母
 - 2 M8垫圈
 - 4 M6六角螺母
 - 4 M6垫圈
- 或
- 1 M6螺纹杆150mm
 - 8 M6六角螺母
 - 6 M6垫圈

第15步——安装台

将工作台上的M3安装孔与轴承上的相应孔对齐安装座和前台架。这可能需要您调整一些六角螺母。



通过20mm M3螺丝穿过舞台上的每个孔（我用垫圈保护顶部），然后通过小3D打印垫片，最后通过轴承和前台安装，用M3螺母固定这些东西，将它们安装在六角孔中，使其更容易收紧。如果你还没有这样做请收紧剩下的M6螺母来固定舞台。

设计指导：

你可以轻易地通过添加弹簧代替间隔件建立一个调平和调整高度的机制。轴承和平台上的六角孔意味着你可以很方便地用艾伦键调整每个角落。这种合适的弹簧的一个伟大而便宜的来源是圆珠笔。



设计指导：

我们将有机玻璃用于我们的舞台，因为它非常坚硬，表面很平坦。孔用数控钻孔机钻，但手钻也可以。你可以使用任意的其他材料如金属板，甚至3D打印一个，但是请注意这会使用很多材料！

在这一步骤中：

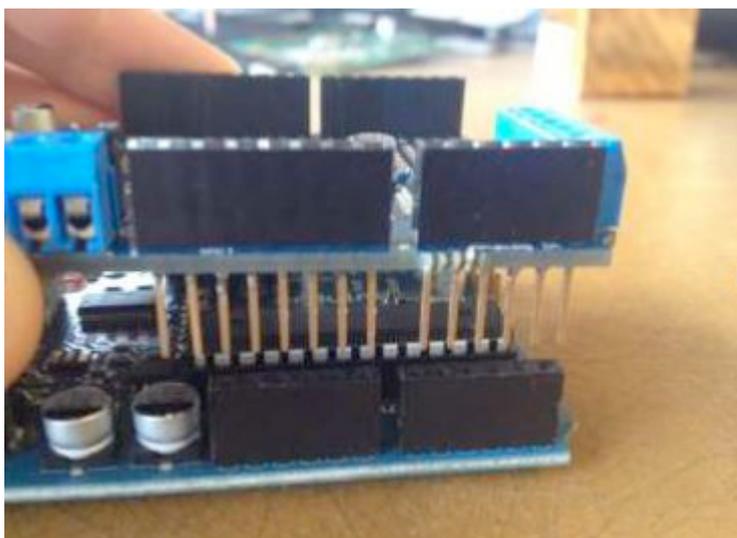
- 1 有机玻璃台
- 4 3D打印垫片
- 4 M3 20mm盖头螺丝
- 4 M3六角螺母
- 4 M3垫圈

第16步——连接步进机

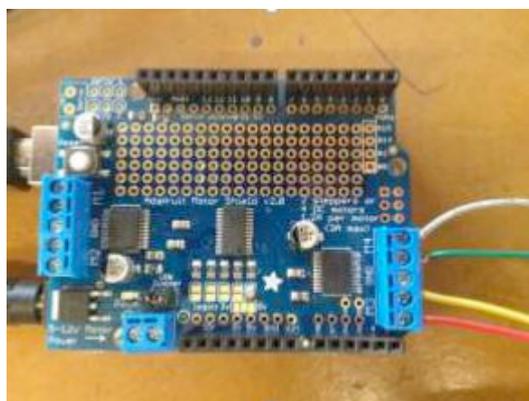
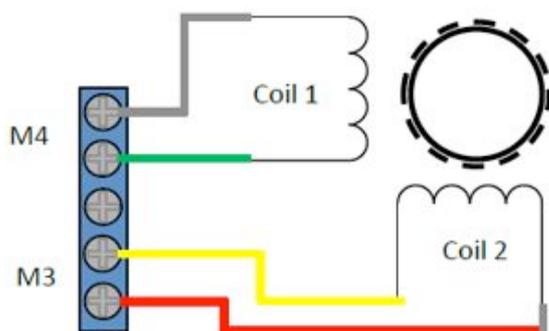
该项目中的步进机由连接到Adafruit V2 Motorshield的Arduino微控制器控制。我们项目中的Arduino实际上是一个Duemilanove，但是我们建议使用更新的版本：Arduino Uno R3。马达架是位于Arduino顶部的可堆叠板，包含运行电机所需的所有电路。它最多可以控制两个步进机或每个板上的4个直流电机，多达32个屏蔽可堆叠在一起（理论上而言）。我推荐使用Adafruit屏蔽，因为它们为Arduino提供了一个非常易于使用的库。它同时能保持大多数I/O引脚可用。



首先，您需要通过安装堆垛头组装屏蔽。焊接母头到Arduino的输入/输出引脚，并将阳头接头焊接到电机屏蔽的下侧。



要将步进机连接到电机屏蔽上，必须首先确定步进机的连线方式。双极步进机有四根电线。这些由两对组成 - 每对都连接到步进机内的两个线圈中的一个。该对中的两根电线应连接到屏蔽上的4个端口之一；这些标记为M1至M4。如果使用单极步进机，则将有一条第五条线 - 这应该连接到M1和M2或M3和M4之间的中央端子。如果您使用与我们使用的相同的Adafruit 12V双极步进机，则红色和黄色线对应于一个线圈，而绿色和灰色线对应于另一个线圈。



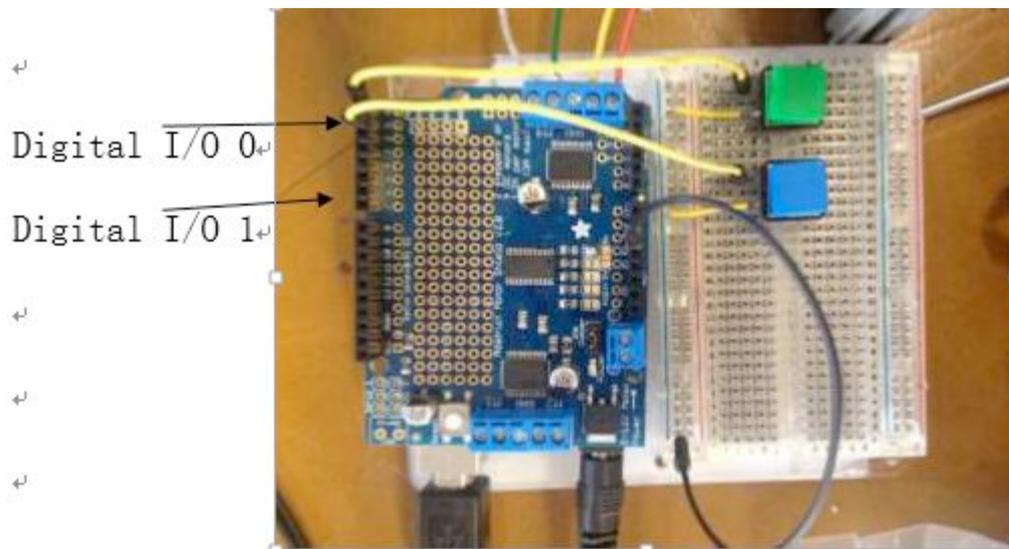
请注意，尽管Arduino应该仅仅使用USB电源来运行步进机，但如果您以后再添加额外的硬件，则可能需要额外提供5V电源。

在这一步骤中：

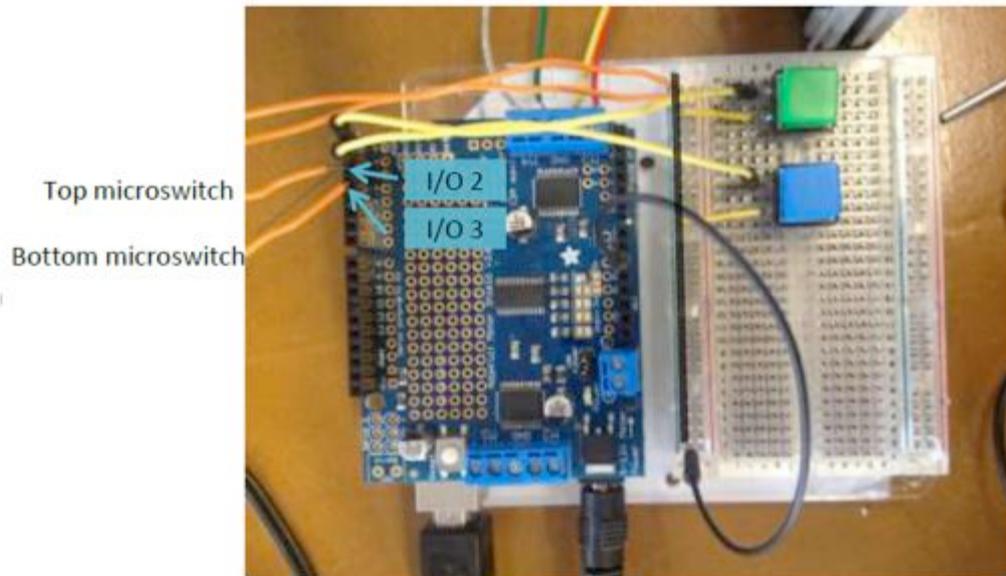
- 1 Arduino Uno微控制器
- 1 Adduruit电机盾Arduino V2.0套件
- 1 一堆堆叠头

第17步——传感器和输入

接下来的两个步骤介绍了使用开关和软件控制载物台的基本介绍。我没有详细介绍，因为这些主题在OpenLabTools硬件和软件教程中有更深入的了解 - 有关详细信息，请参阅OpenLabTools网站。作为基本介绍，我们将介绍包括四个传感器：包括您已经安装在框架组件上的两个微动开关和两个用于上下移动的触觉开关。将您的面包板上的接地线连接到电机屏蔽上的接地针。将两个触觉按钮放在面包板上，并将一个针脚从每个地面连接起来。



接下来决定哪一个按钮成为你的“上”和“下”按钮。将“向上”按钮的剩余引脚连接到电机屏蔽上的数字I / O引脚0，并将“向下”连接到I / O引脚1。



同样，将每个端点的一根电线连接到地。将另一根线从顶端端子连接到数字I / O引脚2，将底端连接到引脚3.如果还没有这样做，请使用USB电缆将Arduino连接到计算机或笔记本电脑。

第18步——软件

在编写程序来控制载物台之前，您将需要下载Arduino集成开发者环境。按照安装软件并连接到Arduino的说明进行操作。

为了进入步进机，您还需要一个库来控制它。如果您使用Adafruit Motor Shield，您可以在这里下载他们的电机屏蔽库。链接页面是使用电机屏蔽的一个很好的教程的一部分，除了步进机之外，还谈到连接到伺服和DCn电机。



```
File Edit Sketch Tools Help
Stepper_Test
// Include the required libraries
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_PWMServoDriver.h"

// Define constants for the input pins
#define UP_BUTTON 0
#define DOWN_BUTTON 1
#define TOP_ENDSTOP 2
#define BOTTOM_ENDSTOP 3

// Create the motor shield object
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

// Create the stepper motor object
Adafruit_StepperMotor *myMotor = AFMS.getStepper(200, 2);

void setup()
{
  // ...
}

// ...

Binary sketch size: 8,114 bytes (of a 32,768 byte maximum)
Arduino Uno on COM1
```

一旦安装完毕，打开Arduino IDE。 Arduino程序或“草图”在C编程：对于本教程，我假设对C编程有非常基本的了解。 草图必须包含至少两个功能：“设置”功能和“循环”功能。 在程序开始时调用一次设置功能。 然后重复调用循环函数。

通过运行这个设置在发动机板上的简单的测试程序开始，设置步进机重复的前进或后退。复制粘贴到你的硬件中设置中或者在Github中下载。点击IDE左上角的标记来检测代码的编译状况，然后点击下载放在硬件中。这个程序下载完就可以直接运行。你应该会发现载物台会上下移动约3厘米。

```

// Include the required libraries
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_PWMServoDriver.h"

// Create the motor shield object
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

// Create the stepper motor object
// The first param is the steps/rev (200 for the Adafruit stepper)
// The other param is the port: use port 1 for M1 and M2 or port 2
// for M3 and M4
Adafruit_StepperMotor *myMotor = AFMS.getStepper(200, 2);

void setup()
{
  // Create the link to the motor shield at the default frequency (1Hz)
  AFMS.begin();

  // Set the speed of the motor in rpm
  myMotor->setSpeed(100);
}

void loop()
{
  // Run the stepper motor. Parameters are:
  // #steps - number of steps
  // direction - options are FORWARD or BACKWARD
  // steptype - options are SINGLE, DOUBLE, INTERLEAVE or MICROSTEP
  myMotor->step(500, FORWARD, DOUBLE);

  // Wait for half a second
  delay(500);

  // Run the motor in the opposite direction
  myMotor->step(500, FORWARD, DOUBLE);

  // Wait for half a second
  delay(500);
}

```

注意当你将步进机连接方式不同的这段代码中的一些参数可以改变。可以通过实验得出最优解。一旦你觉着这个测试程序成功运行了，我们可以开始介绍一些输入控制了。

添加下段代码到程序的顶部。

```

// Define constants for the input pins
#define UP_BUTTON 0
#define DOWN_BUTTON 1
#define TOP_ENDSTOP 2
#define BOTTOM_ENDSTOP 3

```

添加下列安装功能

```

// Setup inputs
pinMode(UP_BUTTON, INPUT);
pinMode(DOWN_BUTTON, INPUT);
pinMode(TOP_ENDSTOP, INPUT);
pinMode(BOTTOM_ENDSTOP, INPUT);

```

最终用下列代码代替循环功能

```
void loop()
{
  // Get the current states of the buttons
  bool up_pressed = digitalRead(UP_BUTTON);
  bool down_pressed = digitalRead(DOWN_BUTTON);

  // If both buttons are pressed, do nothing
  if(up_pressed && down_pressed)
    return;

  // If the up button is pressed and the stage hasn't
  // reached the endstop, go up
  if(up_pressed && !digitalRead(TOP_ENDSTOP))
    myMotor->step(1, FORWARD, DOUBLE);

  // If the down button is pressed and the stage hasn't
  // reached the endstop, go up
  if(down_pressed && !digitalRead(TOP_ENDSTOP))
    myMotor->step(1, BACKWARD, DOUBLE);
}
```