

欢迎使用

Wall Drawing Machine 2020

安装调试说明

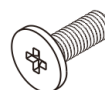
1.安装

螺丝清单

铜柱/旋转轴/
电机/支架固定
S023005



X 26



固定电路板
S063004



X 12



读卡器模块固定
S020805



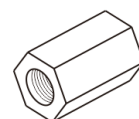
X 4



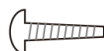
电路板固定用铜柱
T3006



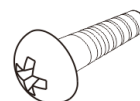
X 12



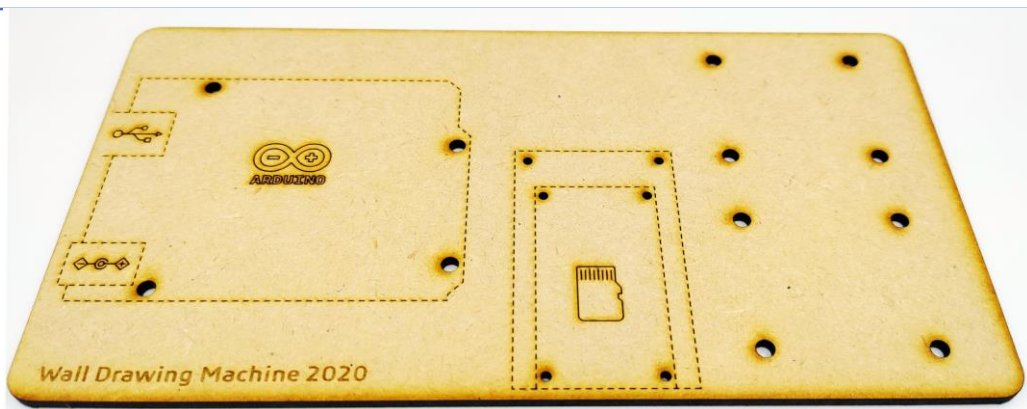
舵机固定螺丝
S042310



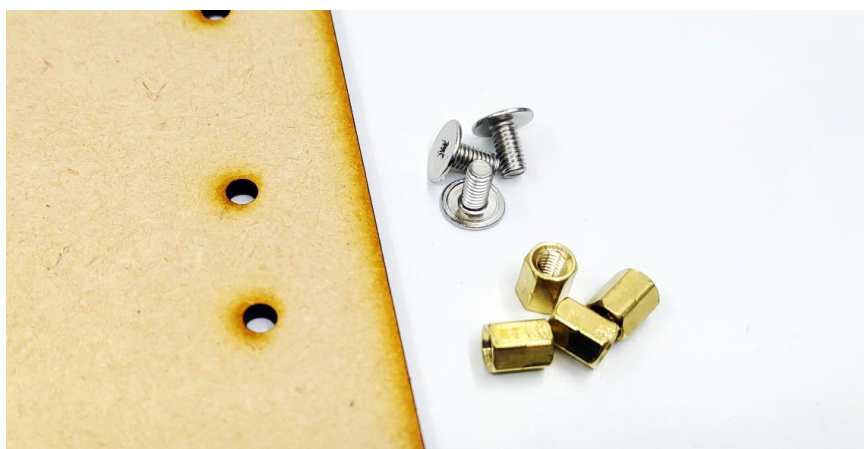
X 2



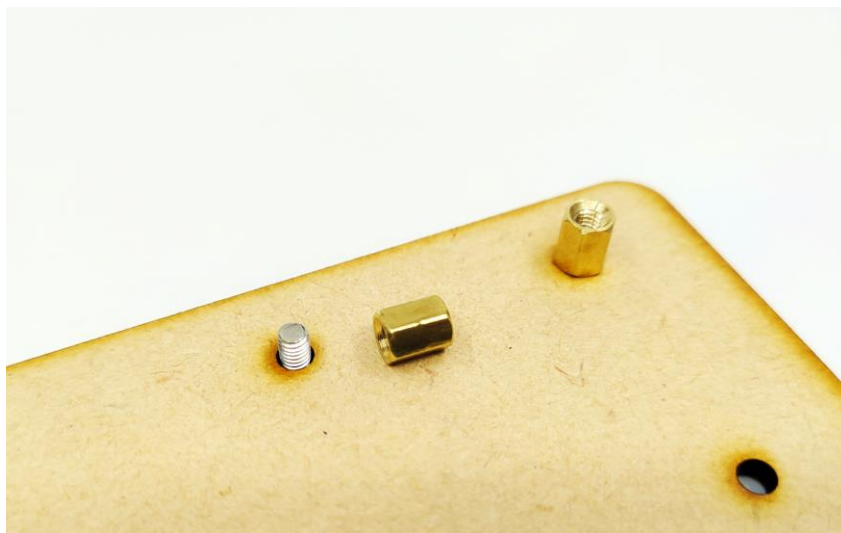
附件包内的螺丝很多种型号，请务必区分清楚再动工。



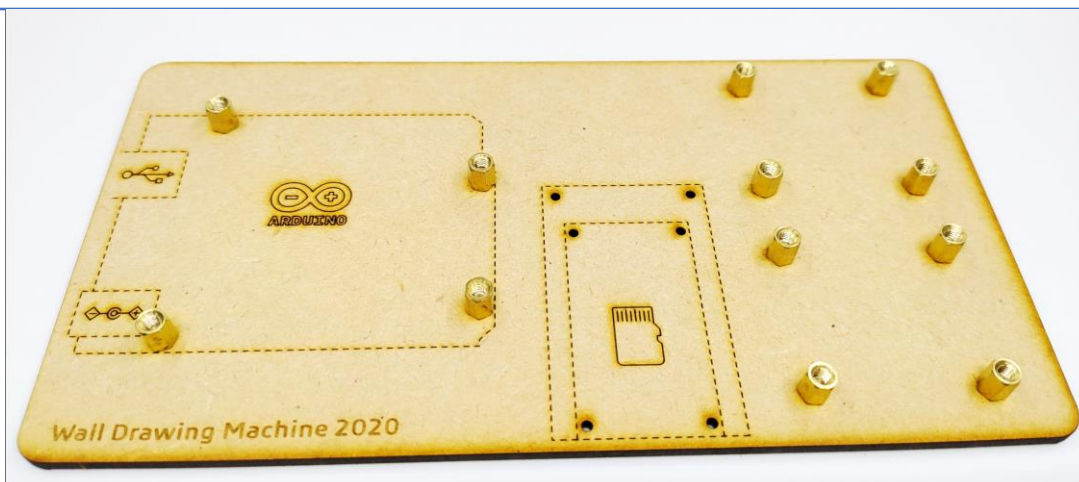
先从底座开始



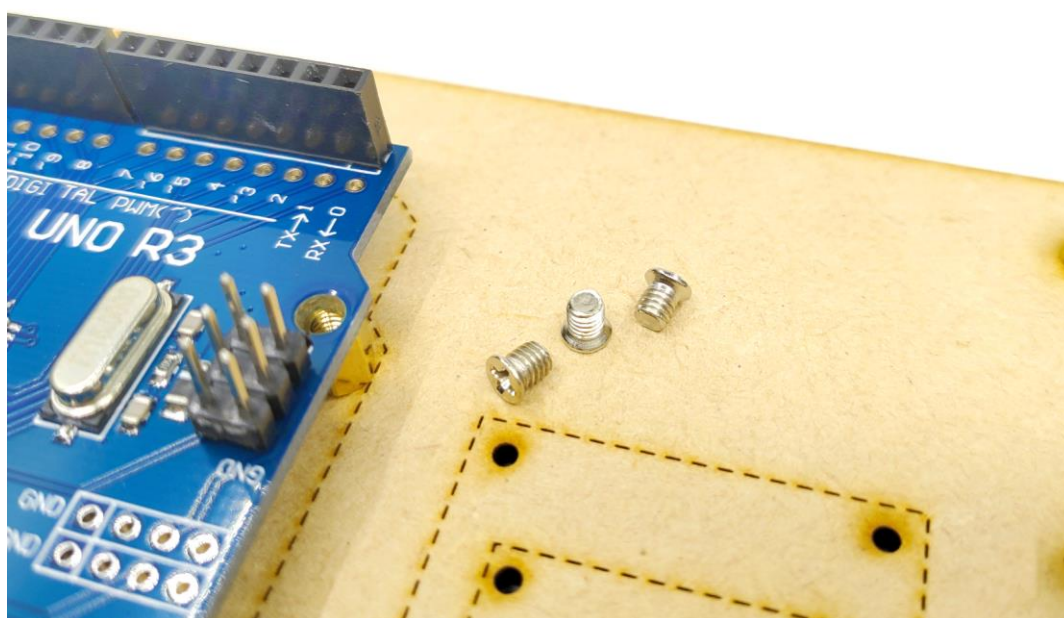
找到 12 个大帽螺丝和 12 个铜柱。



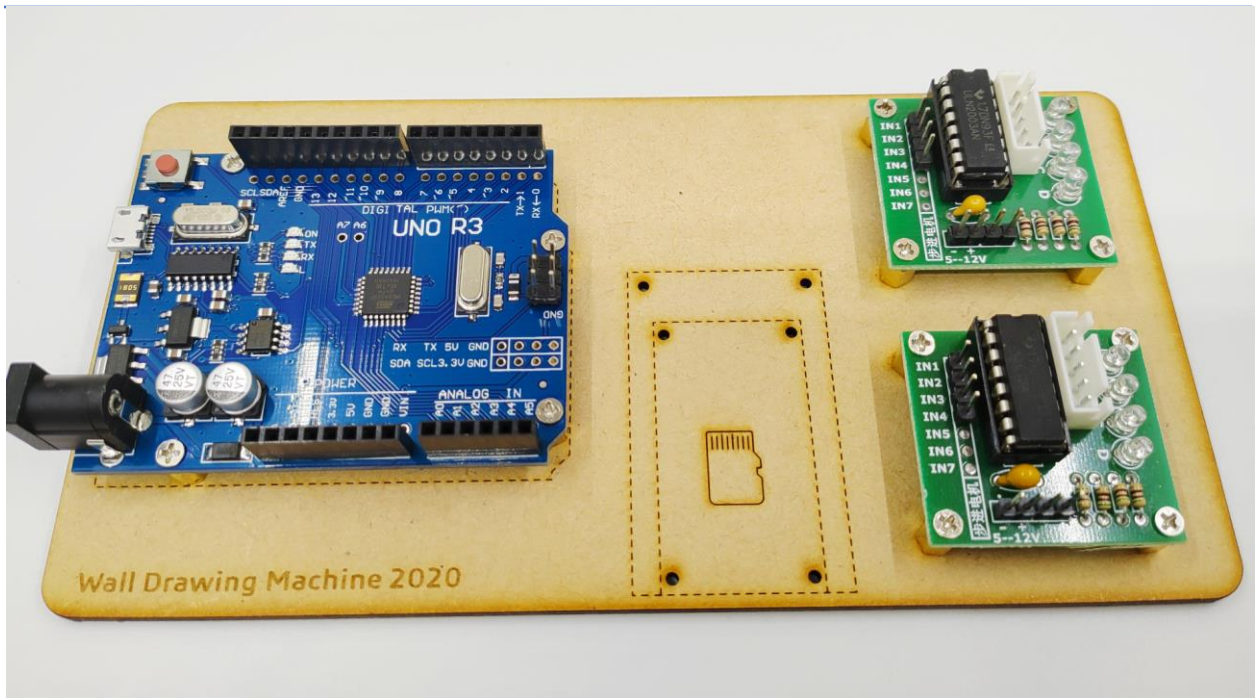
大帽螺丝从背后拧入固定铜柱



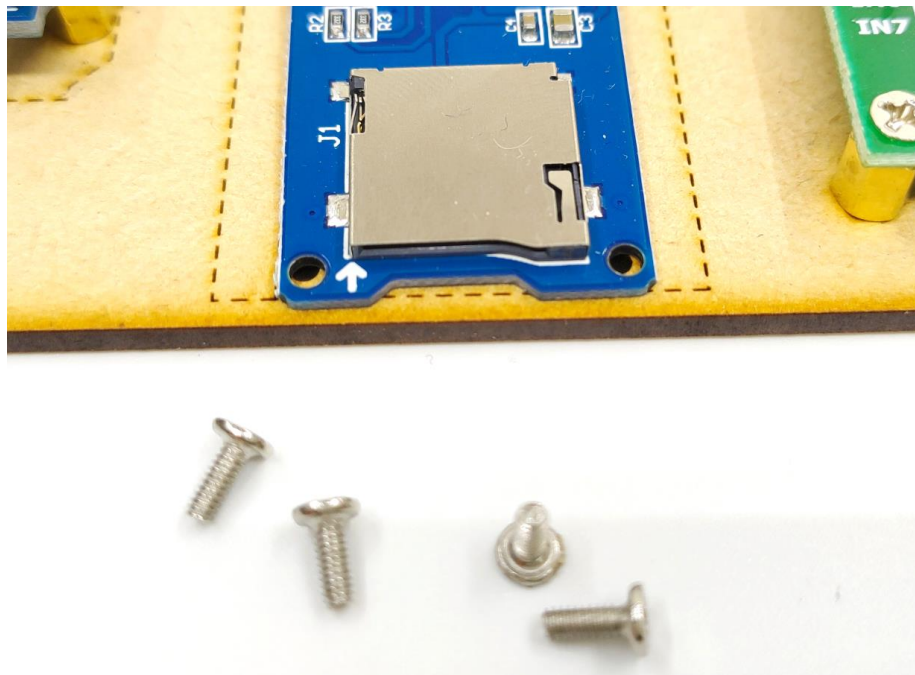
固定好所有的铜柱



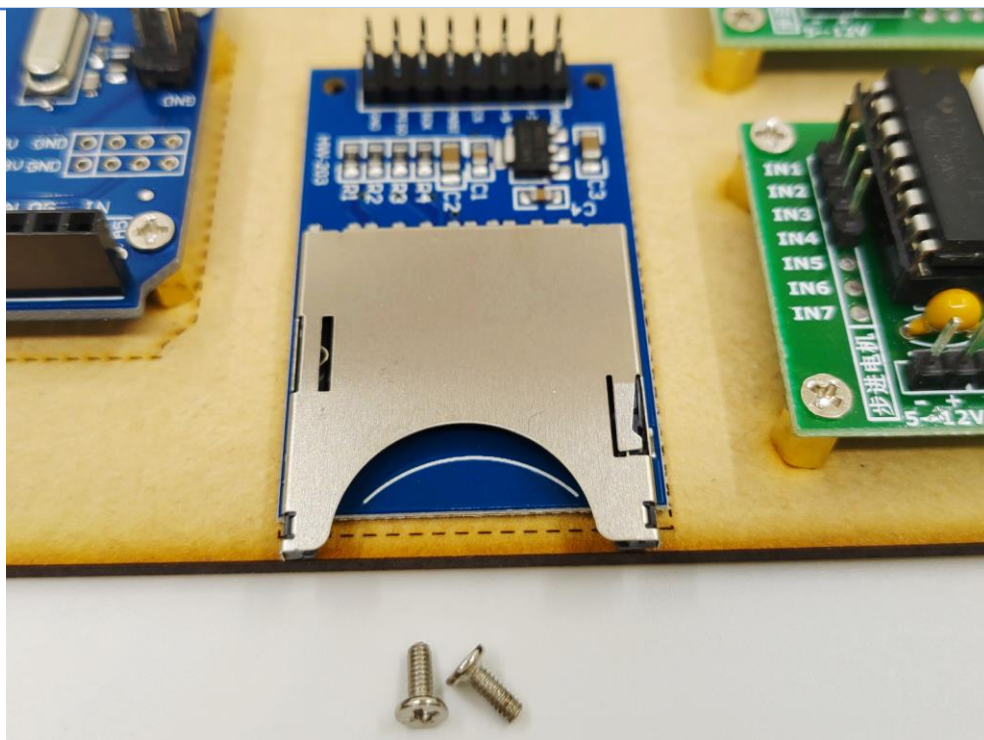
用主板螺丝将 Arduino UNO 和步进电机驱动电路板固定在铜柱上



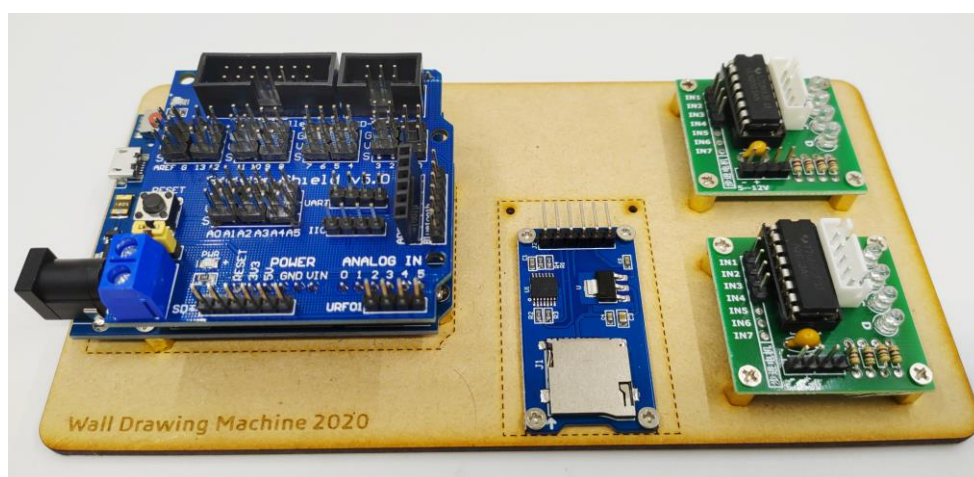
接线图在下面，现在先不用接线。



用最小的螺丝直接在底座上固定 SD/TF 卡读卡器

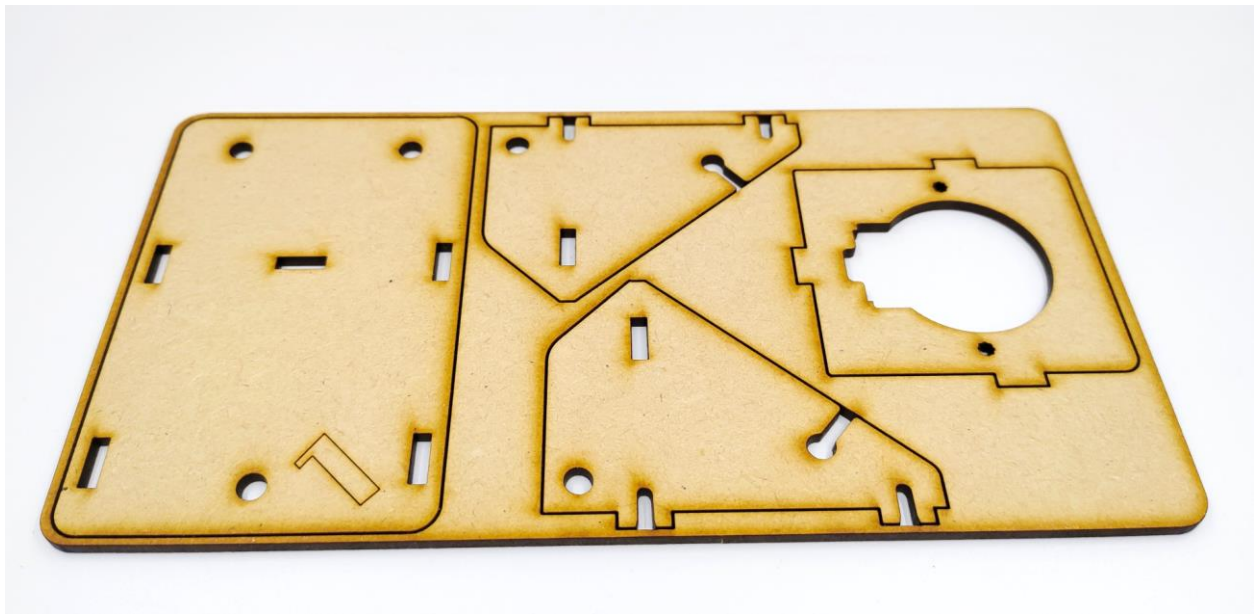


SD 卡读卡器是 2 个螺丝。

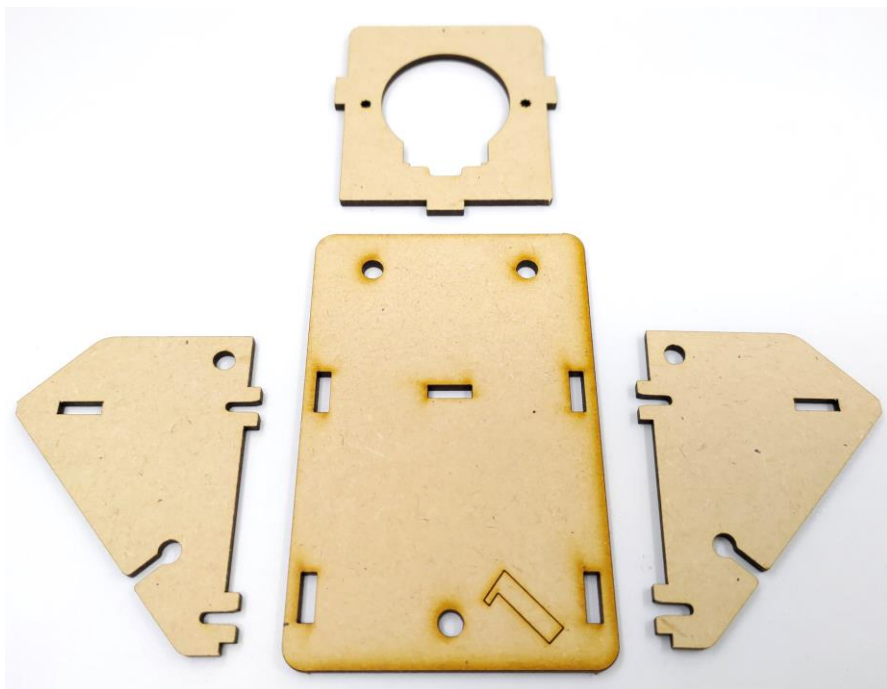


安装完毕的样子 (电路接线图在后面 3.)

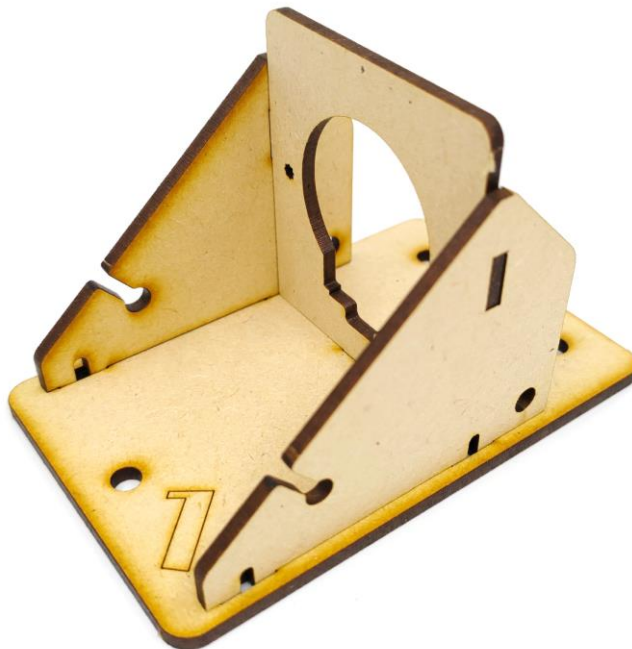
电机支架



电机支架的安装



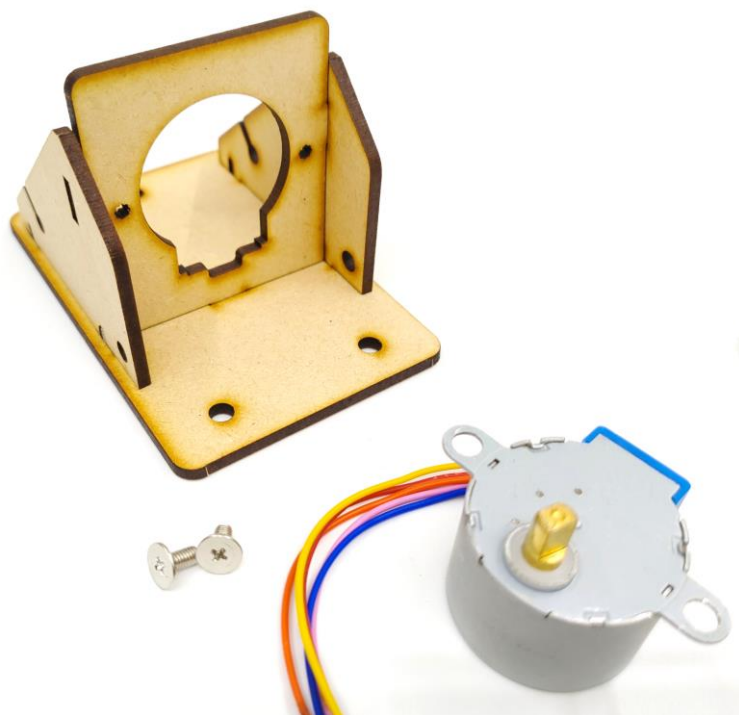
先将零件扣下来



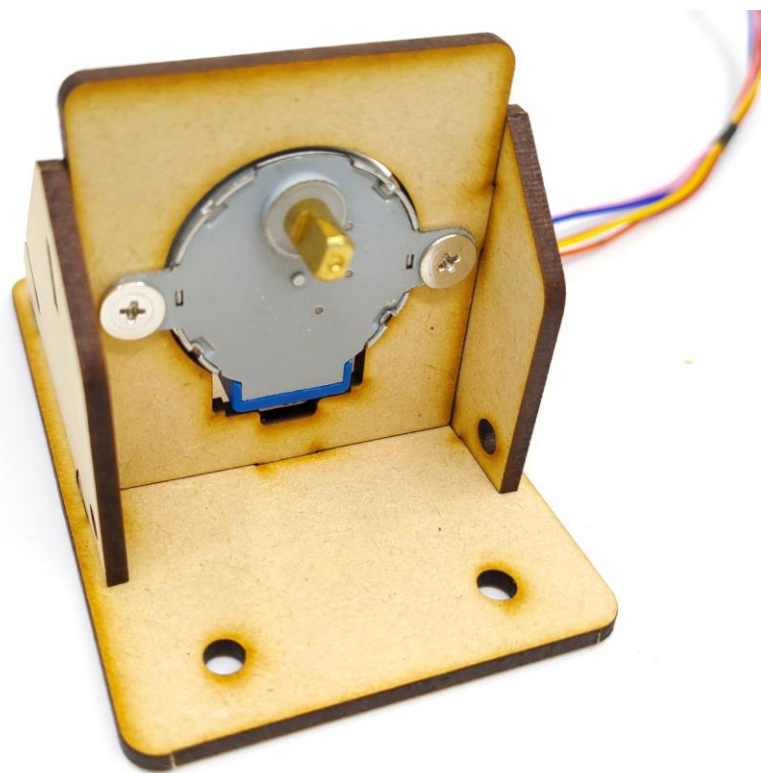


螺丝从背后固定立板



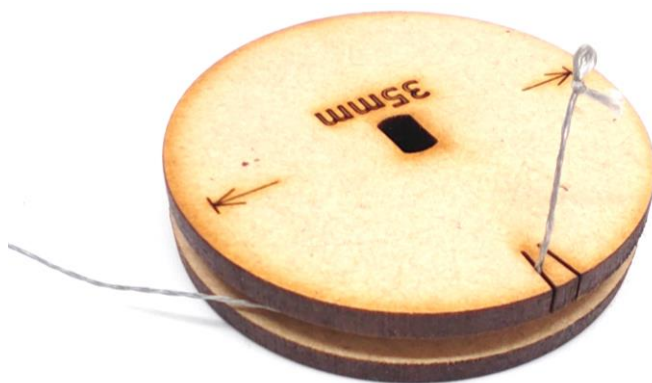


安装电机

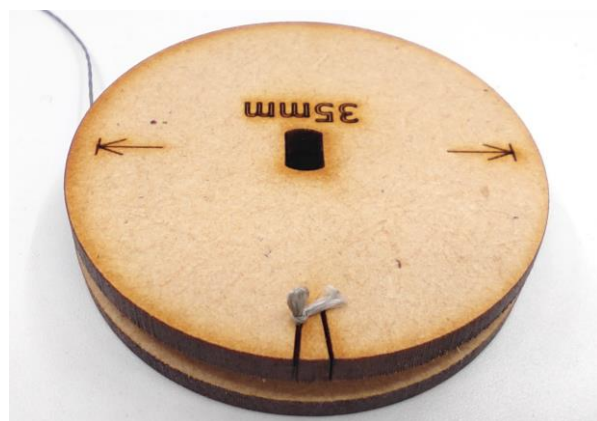




先将拉线一段栓一个扣



将拉线扯入线轴背部的缝隙



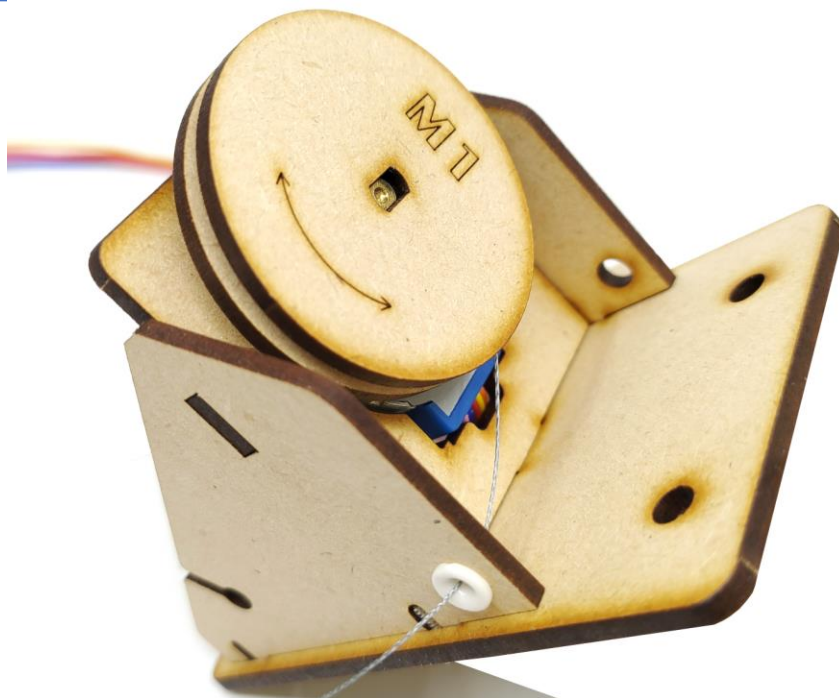
将拉线绕在线轴上即可。左侧线轴顺时针，右侧线轴逆时针绕线。



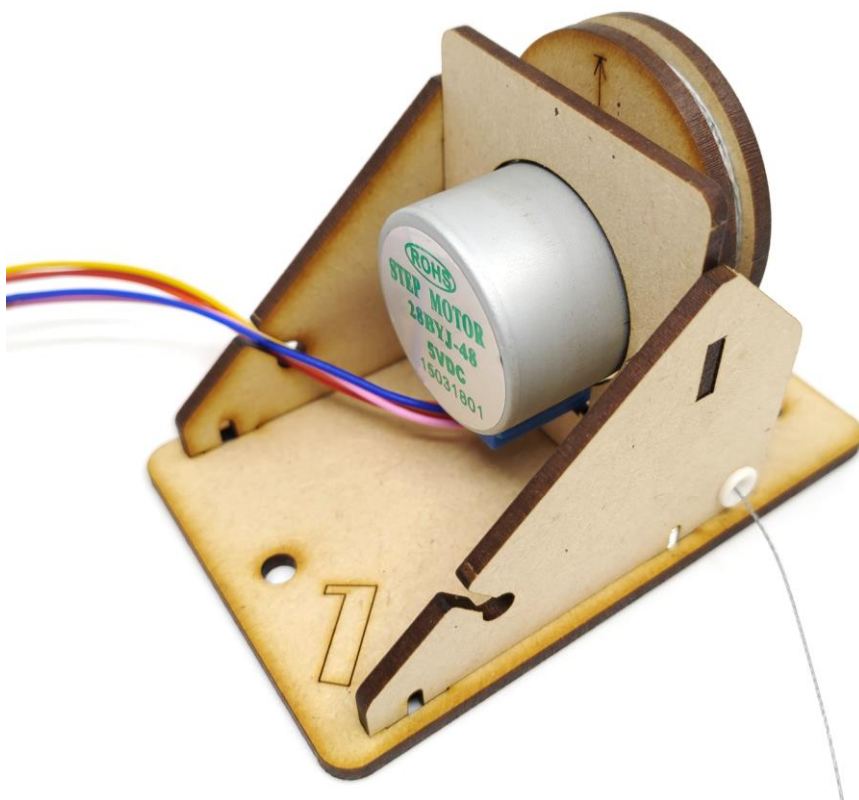
拉绳另外一端，栓一个较大的绳套，用来套笔

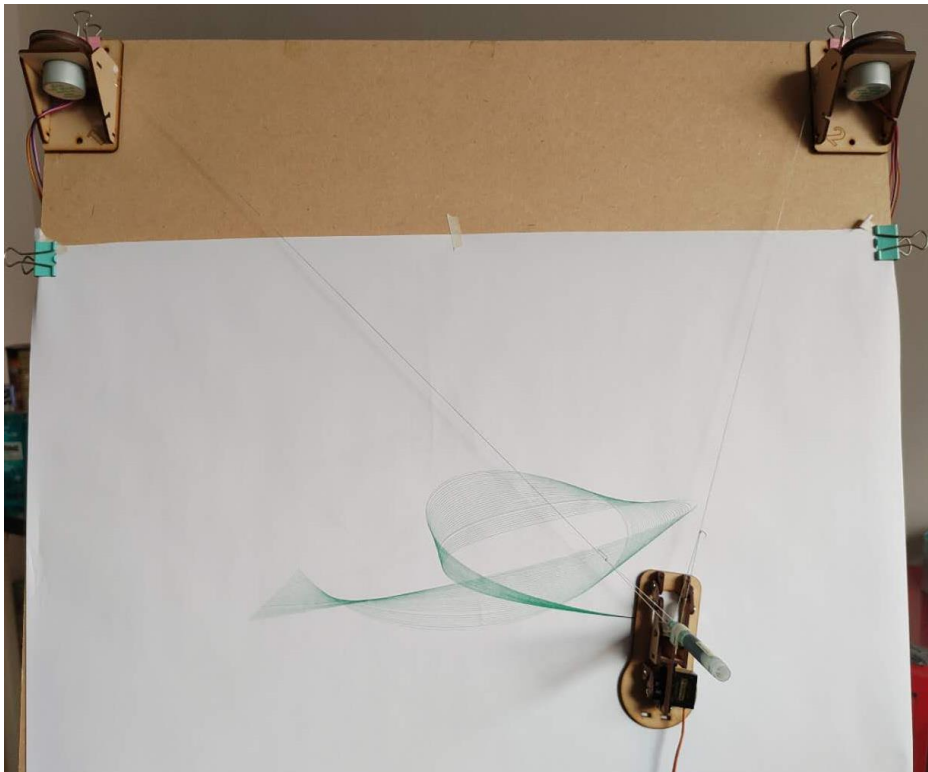
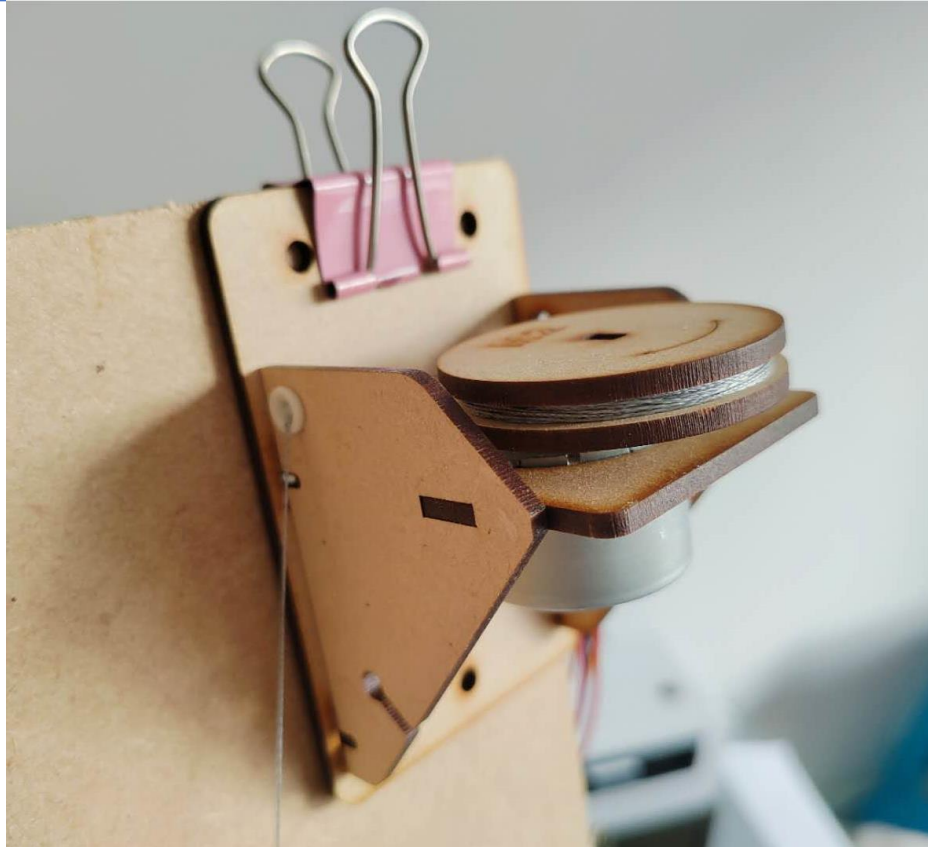


过线环（可以减少线摩擦力）插入立板的孔中，安装在电机支架的一侧。



电机支架安装于绘图板或墙面的两端，尽量保持水平高度一致。程序中需求修改 AB 点的距离，就是 2 个支架间白色过线孔的距离（不是 2 个线轴的距离），需要测量安装距离后修改程序（详细内容见下面原理及参数讲解部分）。

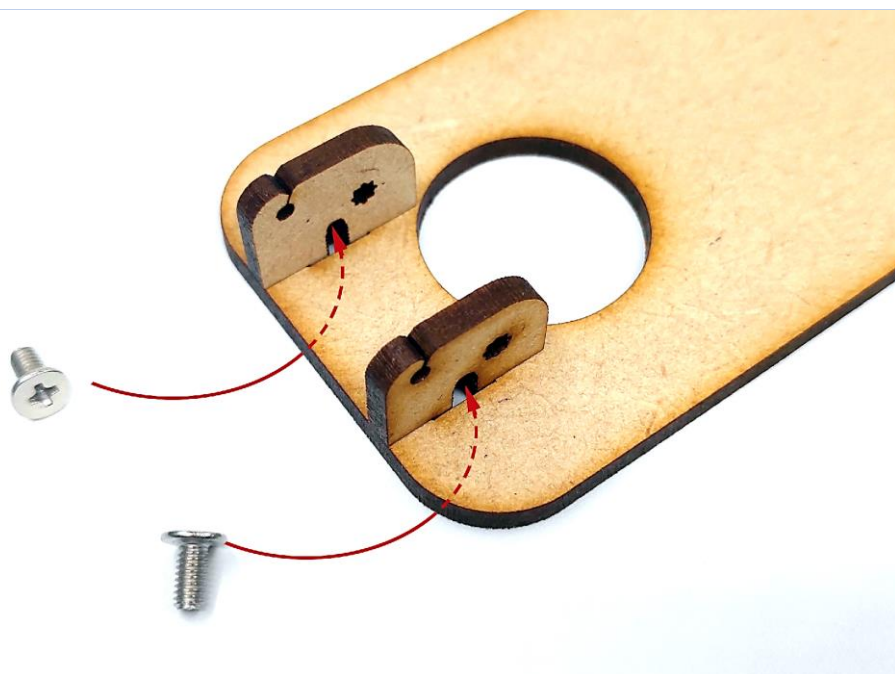




典型的支架安装方式。支架可以用螺丝孔固定，也可以采用粘贴等方式固定。

笔架的组装





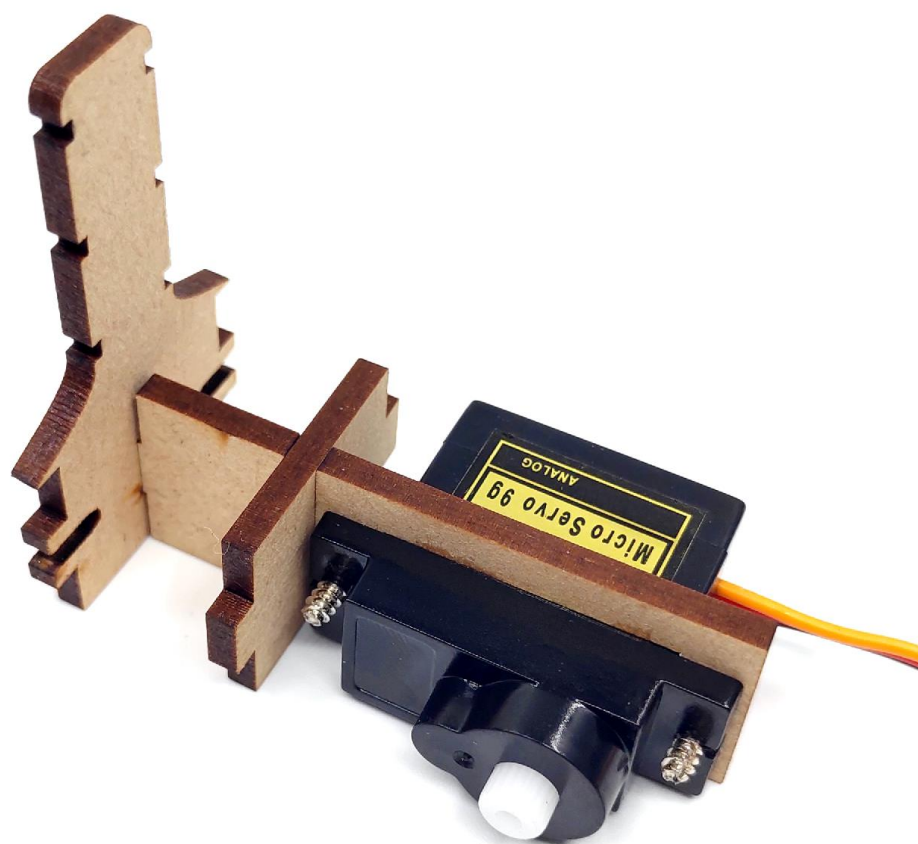
先装底座，2 个大帽螺丝，从背部拧入，无需过紧。



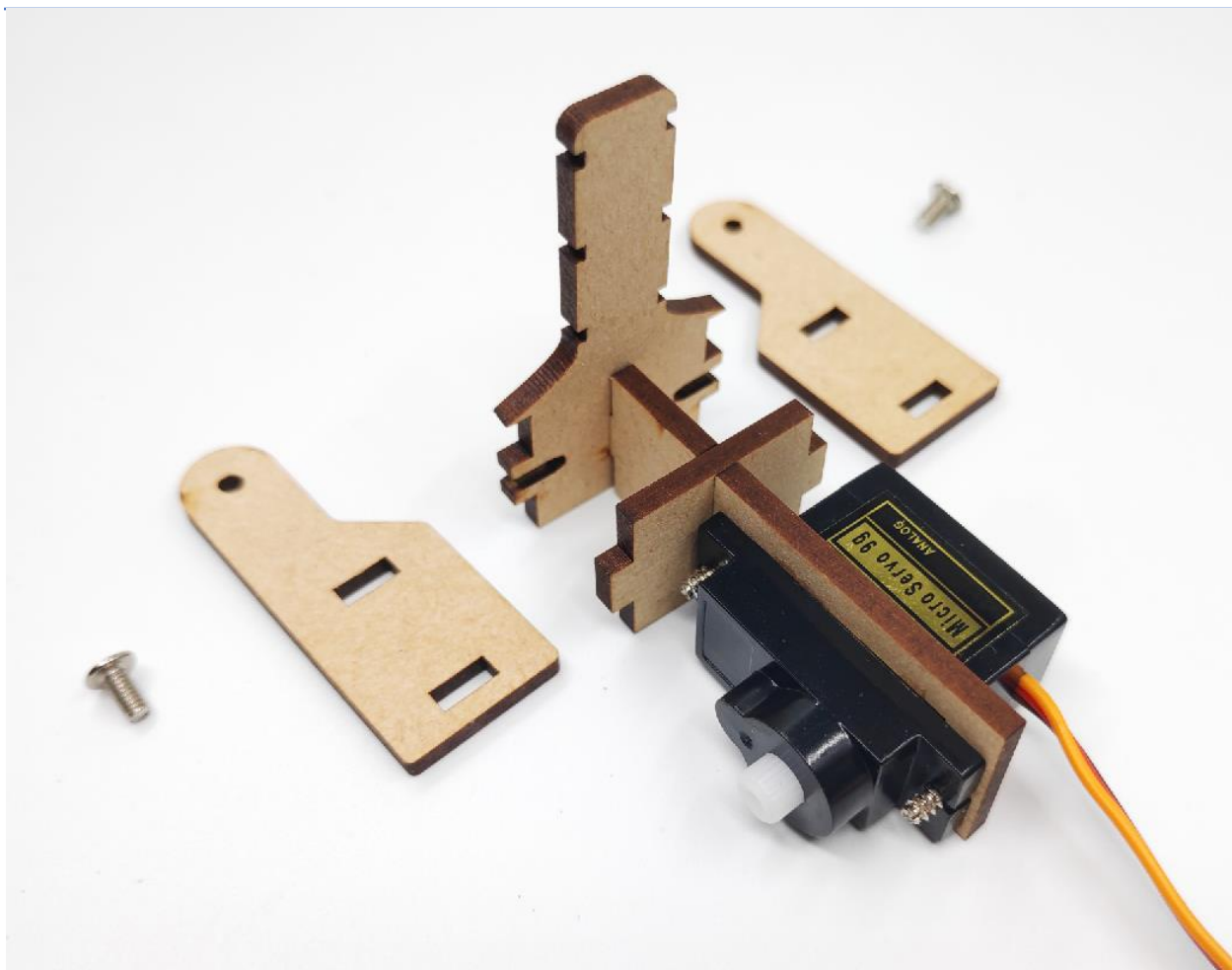


先安装舵机支架。

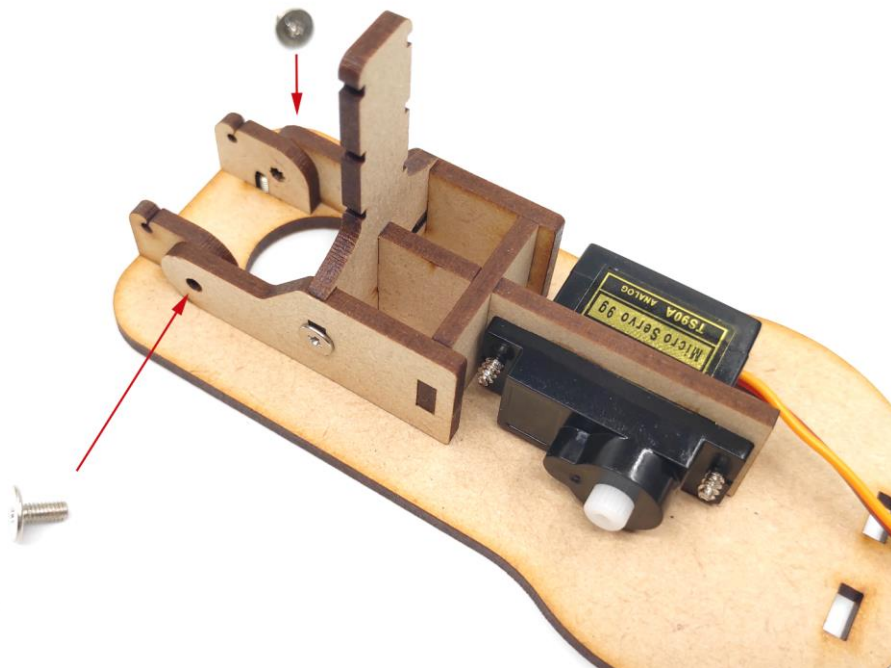
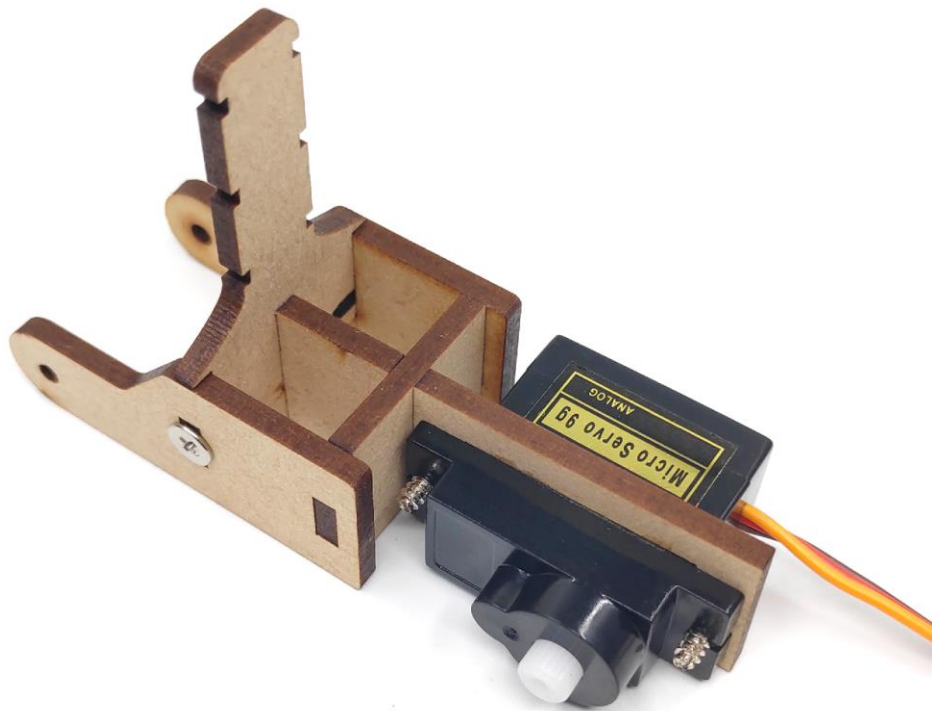




兼容各种型号的 9 克舵机不分左右。本店自带舵机的可以如图方向安装，这样可以保证重心最居中，运行更平稳。



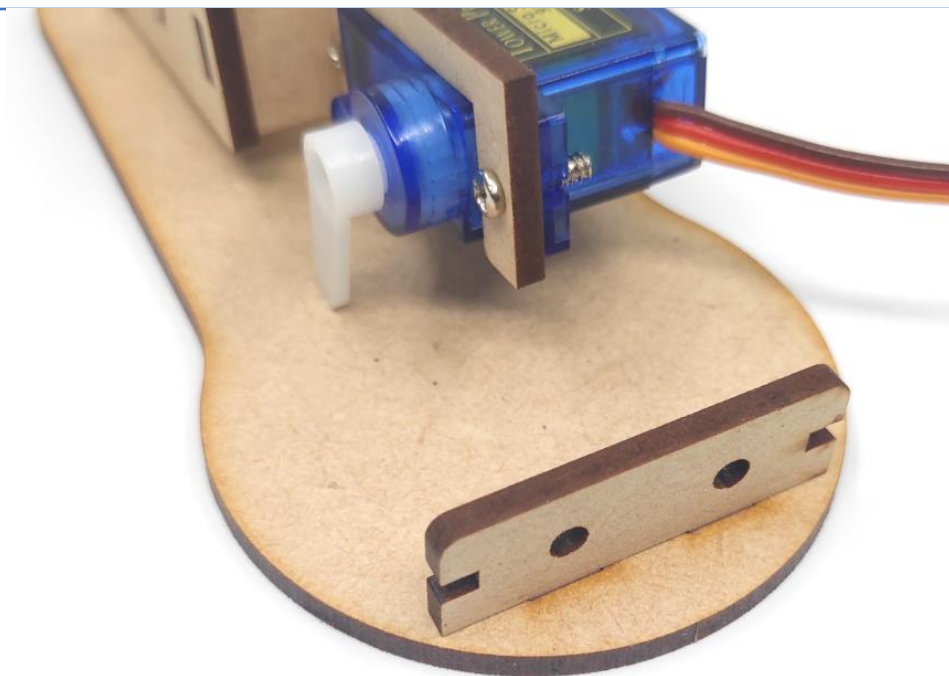
左右侧板需要大帽螺丝固定



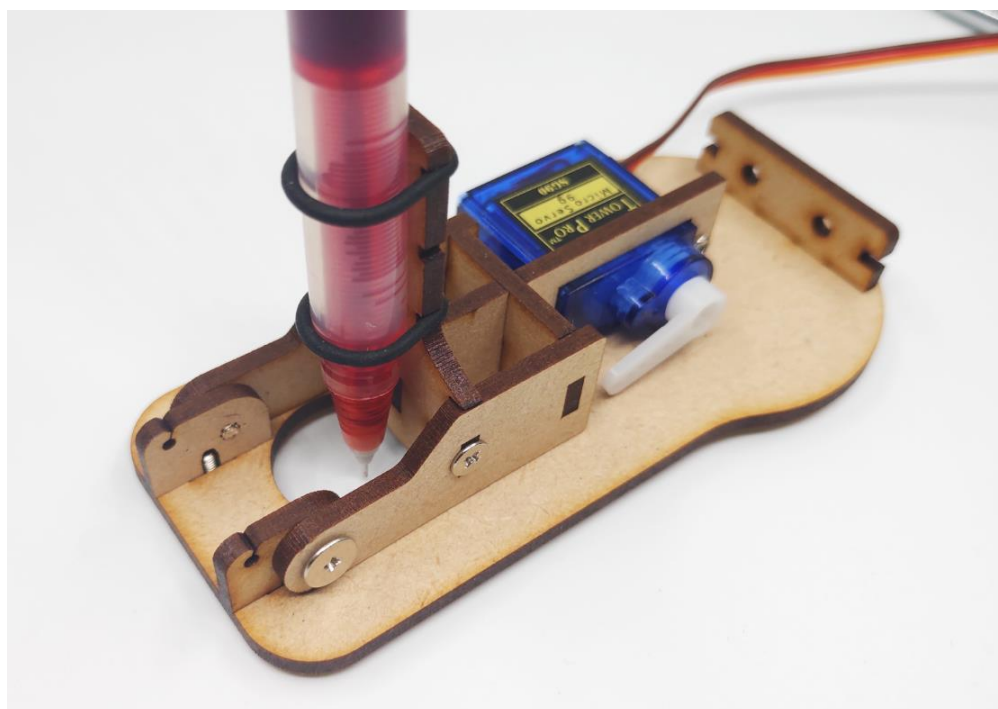
旋转轴需要大帽螺丝，不能拧紧，要有活动空间，使笔架可以自由旋转。

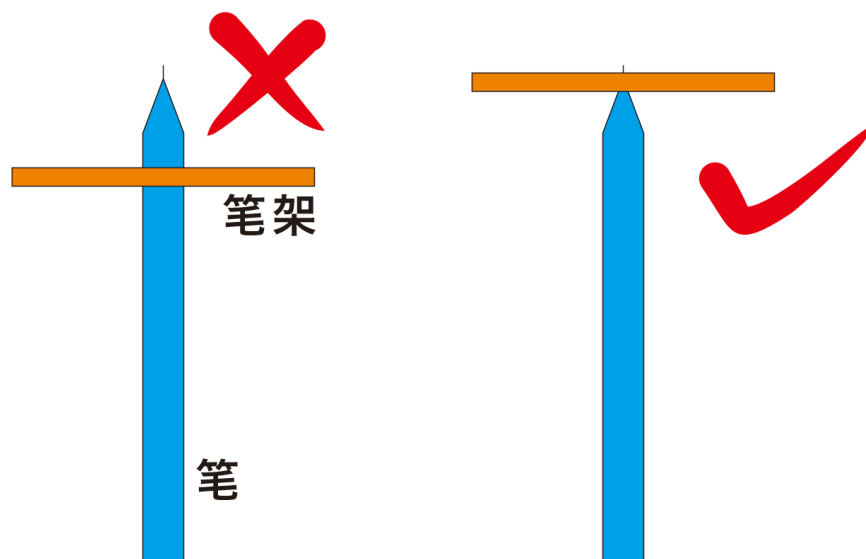


测试可以安装舵机摆臂，调试程序舵机参数，使舵机可以稍微抬高笔架即可。幅度过大，会产生比较大的震动，影响绘图精度，而且也增加运行时间。抬起大约 $5\sim 10^\circ$ 即可。

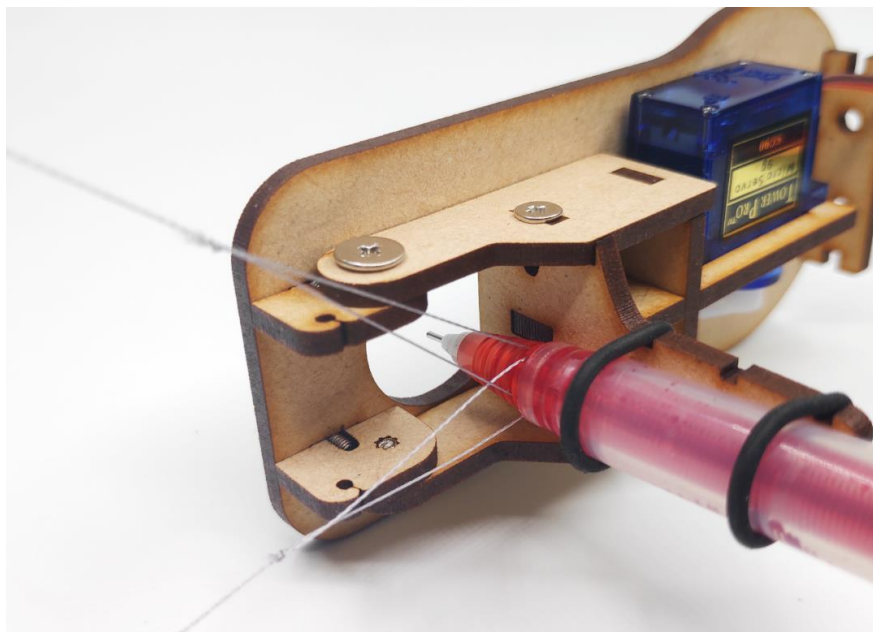


后部的配重块支架如果使用本店配备的笔，可以不用安装，如果其他规格的绘图笔，导致不平衡
可根据情况增加配重。

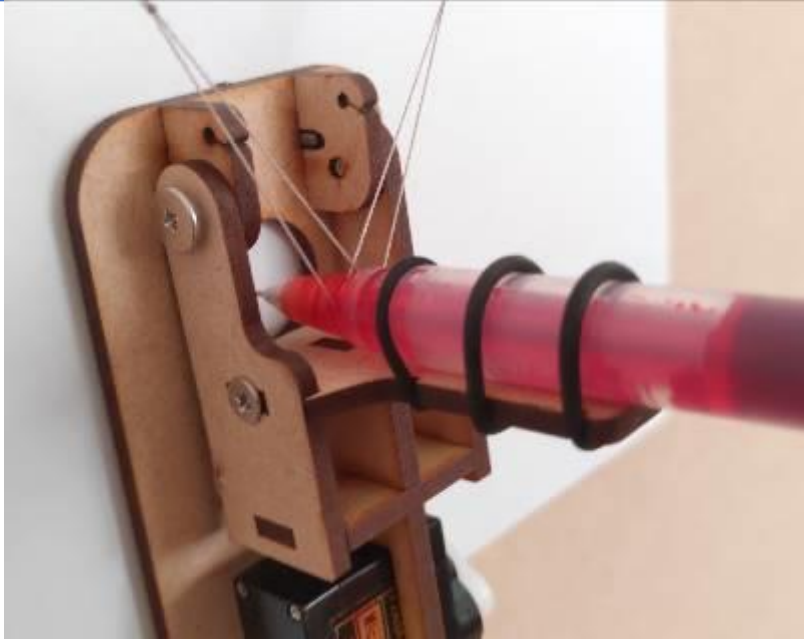




固定笔：调试笔的位置，让笔尖刚好接触到纸张即可，尽量不要让笔架抬起幅度过大。

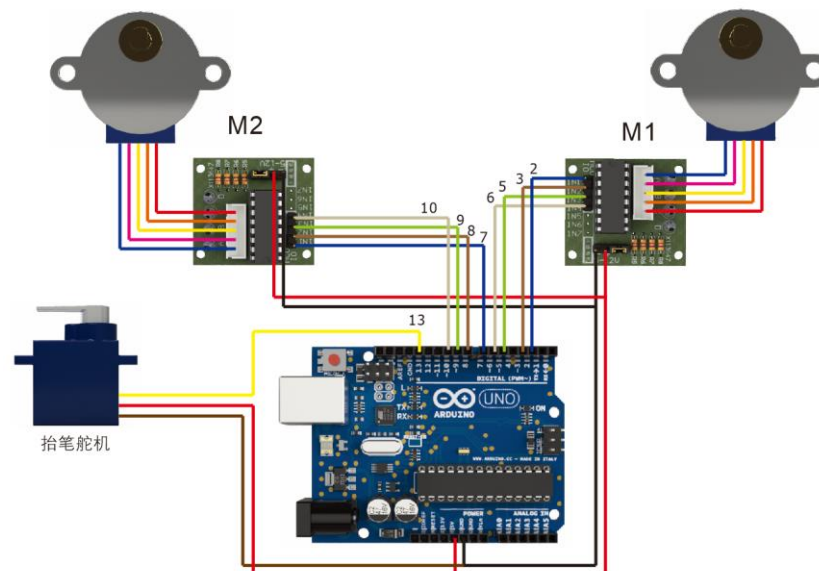


拴挂绳：挂绳系一个环，套在笔的前端凹环中即可。调试程序的时候可以不拴笔，将挂绳扣在笔架前端的环中即可观察笔架的移动情况。

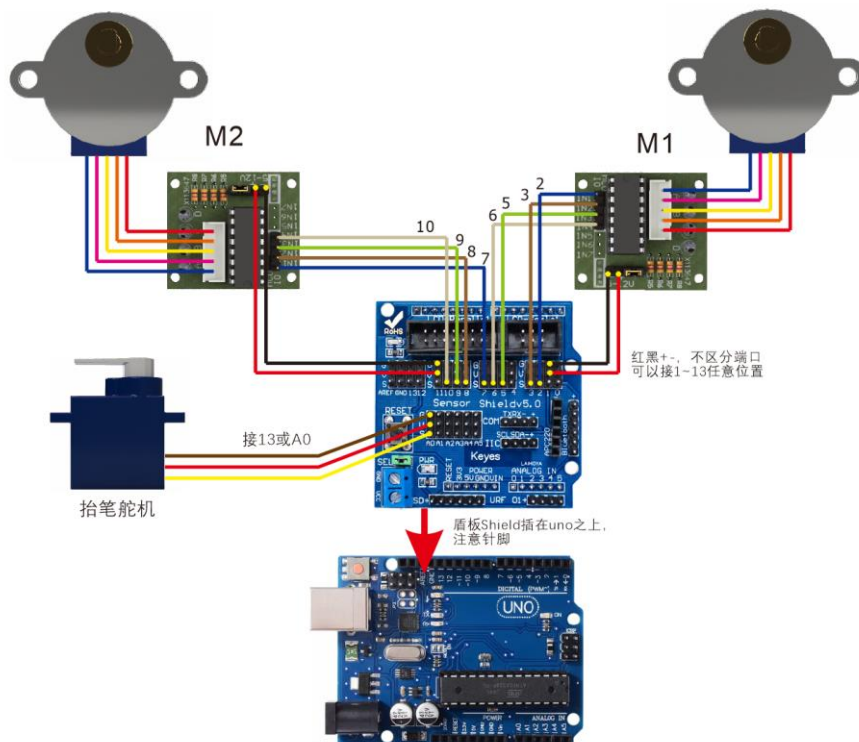


典型的安装样式如上图。

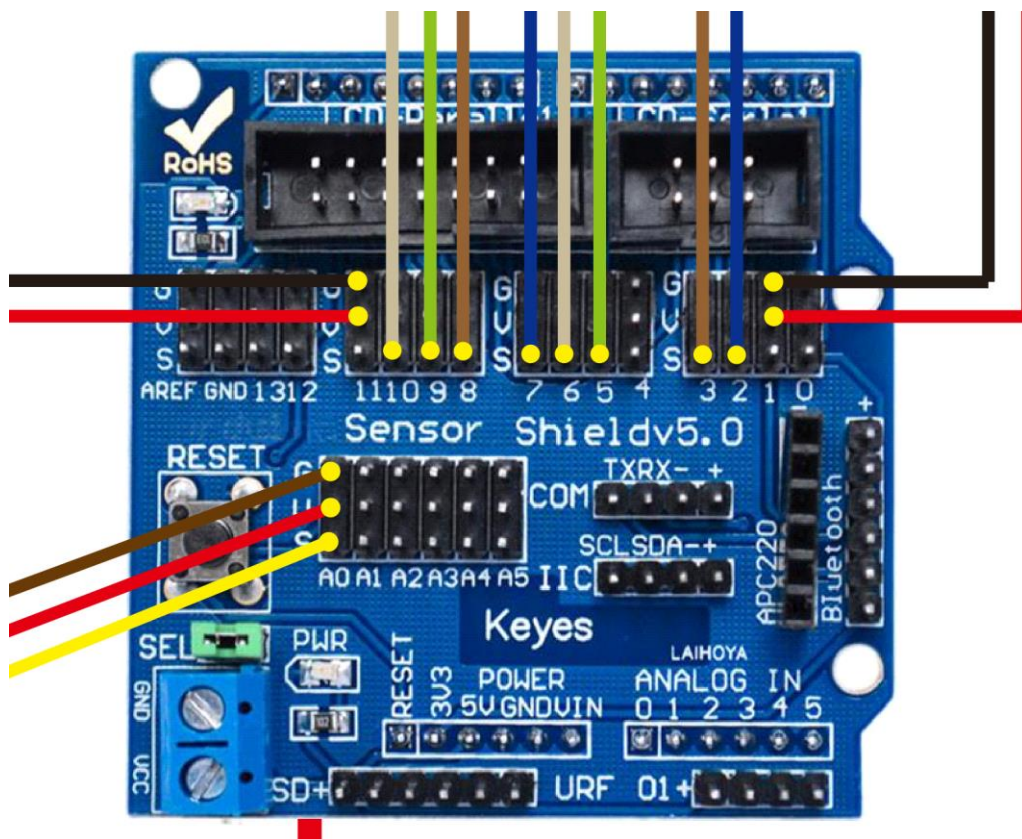
2. 接线



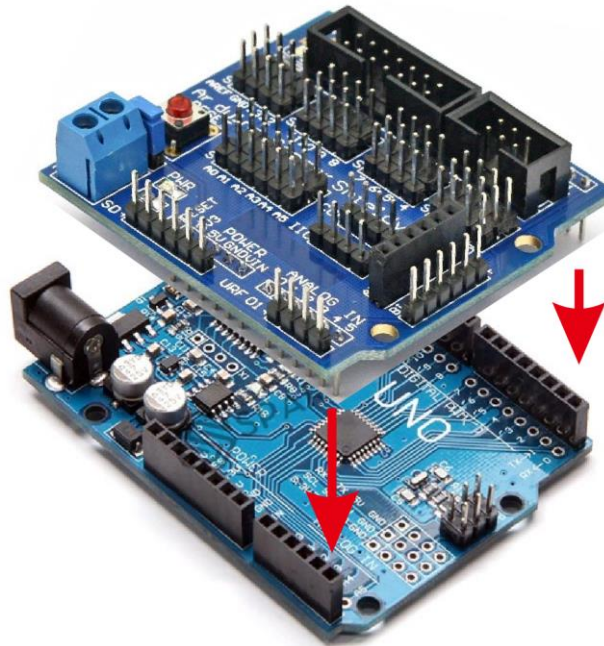
接线图（使用 TF/SD 读卡器模块的，舵机改到 A0 口）



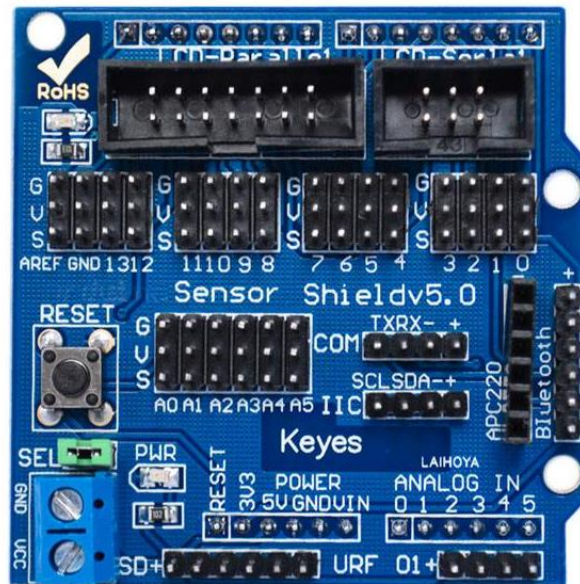
接线图



接线图放大（线路的颜色只是为了区分方便，和实际使用的线色无关）



盾板（又名扩展版）接于 Uno 之上，注意对准针脚。



盾板插于 Arduino UNO 之上，盾板针脚对准 UNO 的排线孔即可。盾板的作用类似接线板，插线非常方便。可以灵活的接插舵机等设备。盾板针脚的号码和 UNO 的端口一一对应。每个端口

都有一个 S 信号 的接头以及 GND（简写 G 或 -）、VCC（V 或 + 或 5v）接口（G、V 口无需对用编号，任意接都可以）注意 13 号左边的一排 3 个都是 GND，再左侧是 AREF 口，不可用。

程序中的接线位置，所有设备的 VCC 是接电源+，GND 接电源 -，有盾板的直接插到对应端口就可以。

- 抬笔舵机 13#（或者 A0 口，使用 SD 卡读卡器）舵机线棕色 GND，红色 VCC，黄色是信号。
- 拉线步进电机 M1、M2：4 条信号线 In1,In2,In3,In4 分别对应 Arduino 2, 3, 5, 6 另一侧 7, 8, 9, 10#，电源线接 VCC 是+，GND 是 -。步进电机控制板有专用线接电机，防呆插头不会错。


















- （选配）TF, SD 卡读卡器：CS 接 4#，MOSI 接 11#，MISO 接 12#，SCK 接 13#（抬笔舵机从 13#改到其他端口比如 A0，程序也需要做相应更改，见程序注释）VCC 是+，GND 是 -（G 和 V 可以接到盾板任何一组 G、V 接口都可以），3.3v 不用接。（V, Vcc, 5V, + 都是指电源正极。G, GND, - 都是负极）
-

SD/TF读卡器模块

3.3v不需要接线

















SD/TF 读卡器模块的接线方式

 (Ver.7.00)	 (Ver.7.00)	exFAT (Ver.7.00)	SD Express (Ver.7.00)	 (Ver.7.10)	SD Express PCIe Gen.3 x 1 Lane
			UHS-III  (Ver.6.00)	 (Ver.6.00)	
			UHS-II  (Ver.4.00)	 (Ver.4.00)	
 (Ver.3.00)	 (Ver.3.00)	exFAT (Ver.3.00)	UHS-I  (Ver.3.01)		
 (Ver.2.00)	 (Ver.2.00)	FAT32 (Ver.2.00)	 (Ver.3.01)		
 (Ver.1.01)	 (Ver.1.20)	FAT12/16 (Ver.1.01)	Default Speed DS (Ver.1.01)	High Speed HS (Ver.1.10)	

SD 或 TF 读卡器模块，不支持 SDXC 以上标准的卡，请选择相对老旧的小容量卡，会有比较好的兼容性。一般 16G 及以下的卡，格式化为 FAT32 格式，不能是 NTFS 或 exFAT 格式，可以适用。部分 TF 转 SD 的卡托也可能有不兼容问题，如果只有 TF 卡，推荐选择 TF 读卡器模块。



SD/TF 读卡器模块的接线方式

  (Ver.7.00)	exFAT (Ver.7.00)	SD Express  PCIe Gen.3 x 1 Lane  (Ver.7.00) (Ver.7.10)
		UHS-III   (Ver.6.00)
		UHS-II   (Ver.4.00)
  (Ver.3.00)	exFAT (Ver.3.00)	UHS-I   (Ver.3.01)
  (Ver.2.00)	FAT32 (Ver.2.00)	
 (Ver.1.01)	FAT12/16 (Ver.1.01)	Default Speed DS (Ver.1.01)
 (Ver.1.20)		High Speed HS (Ver.1.10)

SD 或 TF 读卡器模块，不支持 SDXC 以上标准的卡，请选择相对老旧的小容量卡，会有比较好的兼容性。一般 16G 及以下的卡，格式化为 FAT32 格式，不能是 NTFS 或 exFAT 格式，可以适用。部分 TF 转 SD 的卡托也可能有不兼容问题，如果只有 TF 卡，推荐选择 TF 读卡器模块。

3. 设备性能参数

步进电机型号：28BYJ-48 变速比 64：1 绕线轴直径 35mm 绘图笔的平均移动速度 13.7mm/秒 一般速度 9~27mm/秒（笔架距离电机的不同速度会变化，横移竖移的速度也不同。）绕线轴直径也影响速度变化（后期还会推出其他直径的线轴）。

抬笔舵机：9 克舵机。

主板：Arduino UNO，或其他兼容型号均可。

SD 卡读卡器、蓝牙等存储通、讯模块，可根据需求选配。

4. 绘图仪工作原理和程序参数调节

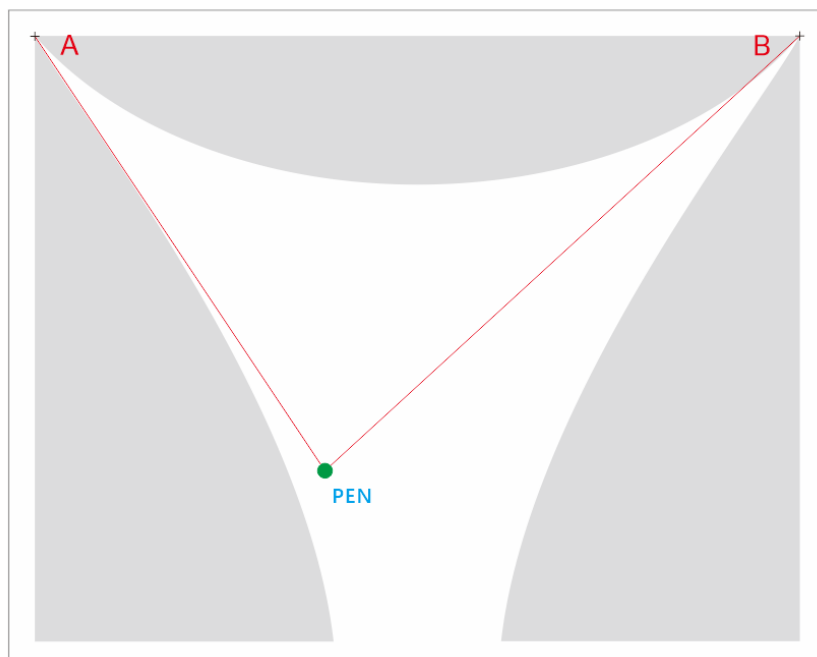
两条拉线拴在笔架上，另一端绕在线轴上，由步进电机驱动，放线或收线。算法原理请阅读《如何将爱因斯坦画到墙上》一文。

拉线在支架的出口位置是 AB 点（见下图）线轴到支架的距离不用计算。此两点的距离是程序中的 X_SEPARATION 参数（单位 mm）。需要测量实际宽度后修改你自己的 X_SEPARATION 值。此段中心为 X 轴原点，原则上左负 LIMXMIN、右正 LIMXMAX。 ← X 轴调节方式

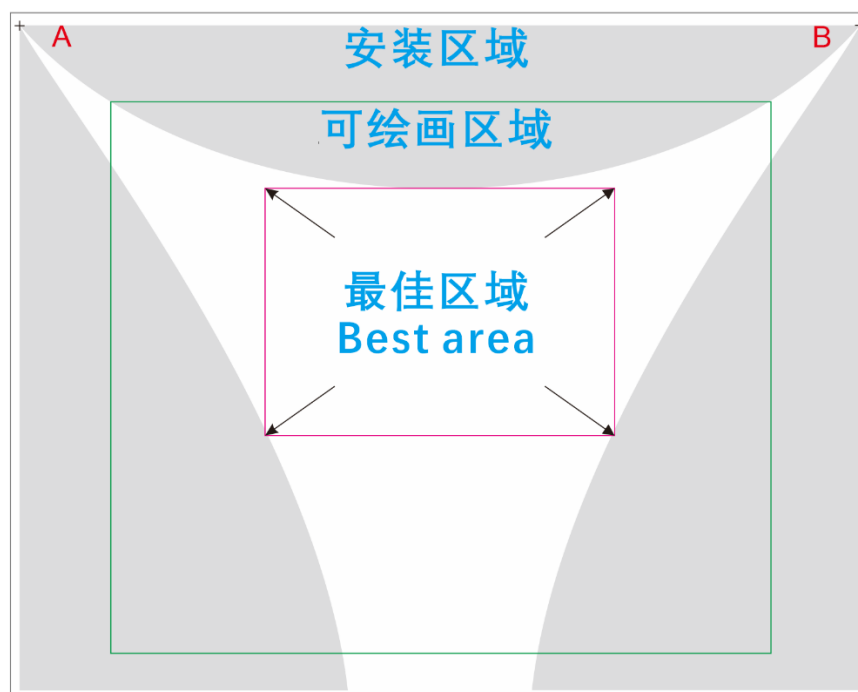
开机时笔架停留的位置是 Y 轴原点。笔架到 AB 的垂直距离为 LIMYMIN（这个数值需要提前测量好，并且每次开机尽量保证让笔尖在此位置，误差过大会产生畸变，X 轴皆然），向下是 LIMYMAX，理论上 LIMYMAX 无限制，只要线轴和幅面足够长。一般情况下，让笔在画面中心，将 LIMYMIN 和 LIMYMAX 设置的一样大。 ← Y 轴调节方式

如果开机时笔架不能在画面中心，可以通过函数 teleport(x,y)来定义笔架的实际位置。以上参数皆可互相调换，并无限限制，而且受电机的转向方向影响，图像可能会是镜像或者是 90 度倾倒，调节相应的参数或程序即可，具体内容可见程序注释。

```
39
40 #define X_SEPARATION 570 //两绳上方的水平距离mm
41 #define LIMXMAX ( X_SEPARATION*0.5) //x轴最大值 0位在画板中心
42 #define LIMXMIN (-X_SEPARATION*0.5) //x轴最小值
43
44 /* 垂直距离的参数： 正值在画板下放，理论上只要画板够大可以无限大，负值区域在笔（开机前）
45 详细介绍见说明文档 https://github.com/shihaipeng03/Walldraw
46 */
47 #define LIMYMAX (-375) //y轴最大值 画板最下方
48 #define LIMYMIN (375) //y轴最小值 画板最上方 左右两线的固定点到笔的垂直
49 //值缩小画图变瘦长，值加大画图变矮胖
50
```

图像并不能无限扩大，理论上图像只受 AB 点的距离影响，越宽图像越大。实际上，并不能如此理想。笔架离 AB 的垂直距离越近，电机的拉力就要越大，最后接近无限大。很显然步进电机达不到，虽然我们的拉线可以承受 10Kg 的拉力。所以图像的最高点不能达到 LIMYMIN 的位置。也不能太靠两侧，因为横相的拉力又变得过于微弱。所以最佳的绘图区域见下图标注。超出最佳区域后，会有一定的畸变，越超区畸变越大。



已做好开始准备？

接下来刷程序。如果你还不会刷程序，甚至不会用 Arduino，那么请你先阅读 1 《**Arduino IDE 下载安装教程**》,2 《**Arduino 程序运行教程**》

也可以到西瓜视频观看视频教程：<https://www.ixigua.com/home/2123847662571084/>

（高手可以直接忽略，刷程序调一下各种参数很简单。）此操作适合不太明白舵机、步进电机是咋运行的新手。

首先运行 2Stepper.ino，这个程序的功能是让步进电机工作，画一个摆线图，相对比较简单，只要调节旋转的圈数就可以变化图案。可能位置会差的很多，直接运行程序的时候，摆臂可能会转到完全不相干的位置上，很麻烦。

主程序是 WallDrawDemo.ino，会绘制几个图案用来测试。

先不要安装笔（绳套可以挂在笔架上），运行一下程序，让笔架空运行一阵，看看运行的范围，是否在纸张范围内。如果超出了，参数还需要调节，具体几个参数的功效见程序注释。

如何绘制复杂的图形，比如手机拍摄的照片。

1. 不是任何照片都可以画的，只能是矢量图才可以绘制。幸运的是部分照片可以通过软件转换成矢量图。（可以用 Coreldraw, Inkscape 等众多软件处理）或者先拿简单的矢量图来试试。
2. 矢量图要转换成 gcode 文件。（可以用 Inkscape, Aspire 以及众多的制作 Gcode 的软件处理）如果不会也可以先用我们软件包里提供的几个 gcode 试试。Gcode 是一个文本文件，可以用写字板打开并编辑。使用前可以浏览一下 gcode 内的数值，这些数值的单位是实际的毫米，如果有几千几万的数值那可能会是灾难性的让步进电机转上几个小时才会停下来。具体 gcode 的规则和使用方法网上很多，可以自行学习。

-
3. Gcode 文件保存在存储卡内，arduino 从卡中读取文件，并将位置坐标解析成步进电机的动作，拉绳带动笔作画。（程序包中的 WallDraw.ino 程序可以执行此步骤）
(注意存储卡最好是 fat32 格式，ntfs 格式可能会读不出文件，尤其是 32G 以上容量的卡。注意 gcode 文件的文件名要和程序里定义的一样，一般是“1.nc”，如果不同请改成一样的名字)

常见问题解答

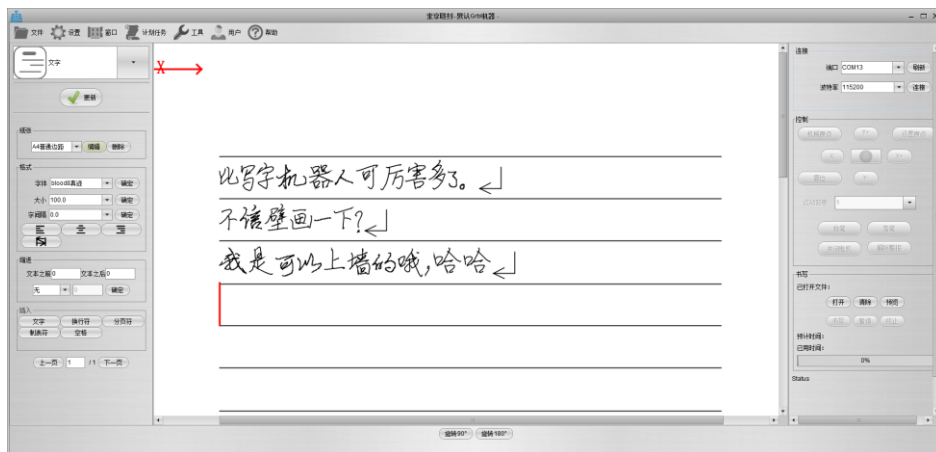
1. 线轴不科学，线绕多了周长会变，不如同步带绕轴的设计好。
答：线轴直径 35mm，线粗 0.16mm，绕 20 圈，线长增加 2.2 米，轴径变化 0.2mm，对于 2 米幅面的图像误差大约在 0.314mm，可以忽略了。
2. 图像只有反转，上下颠倒，甚至是 90 度旋转了，还有很大变形。
答：电机的不同，程序的方向定义都可能影响反转镜像。更改相应参数就行了。比如 XY 的 MAX,MIN 的正负号，或者电机的线序，或者 M1_REEL_OUT 等 4 参数的符号都可以改变方向。如果是 90 度旋转了，可以改变 moveto 函数里 大于小于 <、> 符号就可以旋转过来了。

```
291 Serial.println(d2);
292 #endif
293
294 long ad1=abs(d1);
295 long ad2=abs(d2);
296 int dir1=d1<0 ? M1_REEL_IN : M1_REEL_OUT;
297 int dir2=d2<0 ? M2_REEL_IN : M2_REEL_OUT;
298 long over=0;
299 long i;
300
301
302 if(ad1>ad2) {
303   for(i=0;i<ad1;++i) {
304     m1.step(dir1);
```

3. 电机不转动
答：响，震动偶尔可能虚弱的动一下？线序有勿更改线序。不响不转灯控制板的 4 个灯也不亮，估计是没接线，包括电源线也要接的盾板上随便找一个+-就可以。
 4. 运行正常，画图畸变非常大
答：位置参数不对。开机时笔架的位置是 0，0 实际位置和程序定义的差距太大会比较大的畸变。
-

5. 写汉字怎么弄？

答：墙裂推荐“奎享雕刻”非常不错的软件，<http://kvenjoy.com> 官网可以下载测试版免费试用。测试版用户名 guest 密码：123456 购买我们的墙画机可以有软件的優惠碼哦。

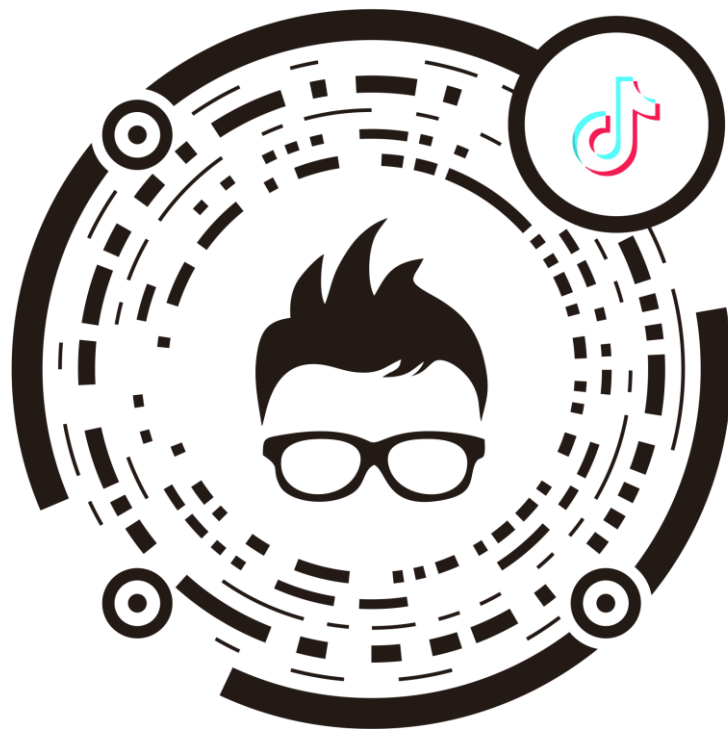


6. 如何画画？

答：请参阅《如何将爱因斯坦画出来》一文，有详细的原理介绍。知乎文章 <https://zhuanlan.zhihu.com/p/71602189>

了解更多

欢迎到我们的店铺 <https://somebodys.taobao.com>



欢迎围观本店抖音账号
