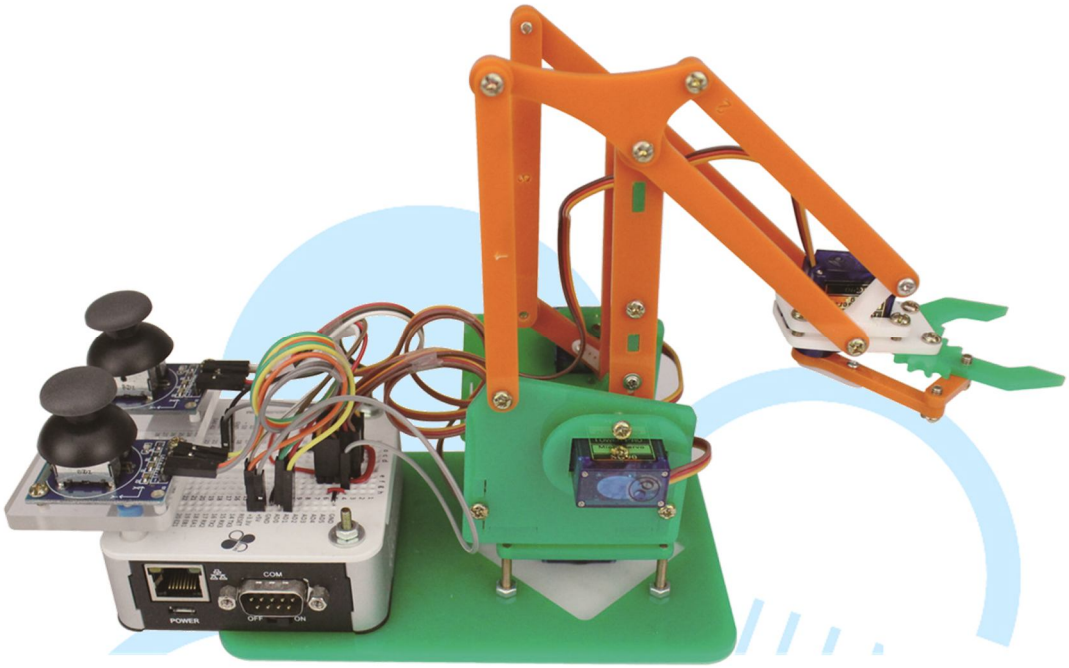
EduCake 控制机械手臂



一、 关于机械手臂

机械手臂(RobotArm) 如图 1 被广泛的使用在各种工业生产环境中，主要是取代人手，去作繁重的工作(EX:汽车组装、货品装箱)、重复性高的持续工作(EX:水果拣选)、精密要求的工作(EX:电路焊接)等等，这些工作人们并不是不能作，但机械手臂作来可以 24H 不用休息、不需要给薪水、不会耍脾气而且还很精准不易出错。所以在工业上已经有越来越多的公司采用机械手臂来取代人工的动作，只留下一些复杂的难以用机械手臂取代的事情才需要人工，可以大大的节省成本和增加产能、增加产品良率，这已经是产业的大势所趋，所以之前

鸿海的郭董才会喊出要建立百万机器人大军。我们这章节就来谈谈机械手臂的构成和控制。

图 1. http://www.kuka-robotics.com/ 很有名的 Kuka 机器手

二、 机械手臂的基本构造

工业用机械手臂动辄数十~数百万台币起跳，而且构造其实很复杂，主要可以分成机体本身、前端功能(可能是喷枪、焊枪、夹爪、吸盘、电磁铁等等不同功能的东西，甚至需要能人工或是自动更换不同功能)、电源供应系统、PLC 控制系统、感测系统、计算机监视/警报系统、UI 控制接口等等部分所构成，这些东西通通要弄懂得要花费很久的时间。我们这里先从最基本的教学用机械手臂开始来谈。

员工会使用工业手臂之前得先教学作基本训练，学校用在教学领域的机械手臂其实是很久以前就有人模仿工业手臂的长相，简化很多之后作出各式各样的教学用手臂，如图 2，就是一种笔者很常用的机械手臂，使用一些很基本的机构和很容易买的到的马达、控制板所构成，整个成本约是数千~一两万台币就能买到。

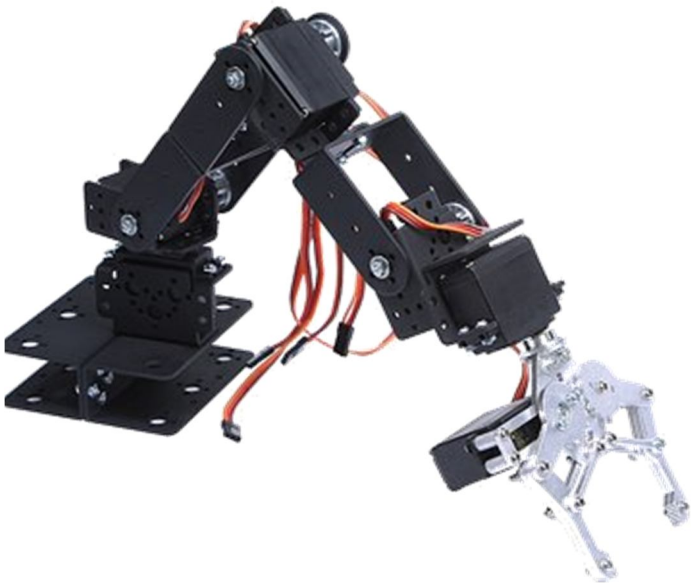


图 2. 一种很常见的教学用机械手臂

这种机械手臂大约 10 几年前就被创作出来，就是只有基本的功能，很多轴数控制产生各种动作，中间又一直有持续的改进，产生无数种衍生型别，笔

者玩最多的就是这种类型，轴数约是 2~10 轴都有，也自己设计过十余种不同的机械手臂。多年来基本的教学款式一直是类似这种的，直到 2013 年才开始有重大的演进。首先是因为网络募资的风行，使得一些想创业的 DIY 玩家看到机会，2013 年中左右，出现一只叫作 uArm 的机械手臂，并且 2014 年初在 kickstarter 上面募资获得空前成功，募得七百多万新台币 https://www.kickstarter.com/projects/ufactory/uarm-put-a-miniature-industrial-robot-arm-on-your。因为这只手臂的成功和他的开源设定(设计图、电路、程序都完整分享到网络上)，募资结束后网络出现很多种模仿改进版本，像是 playArm、liteArm、meArm、xxArm...等等，其中的 meArm 就是我们本篇文章要采用的对象，作者把相关的资料都放在这里公开给人取用 http://www.thingiverse.com/thing:360108 ，他的手臂长像约是如图 3 ，被各方玩家用各种不同材质、颜色来制作呈现出来不同风格的作品。

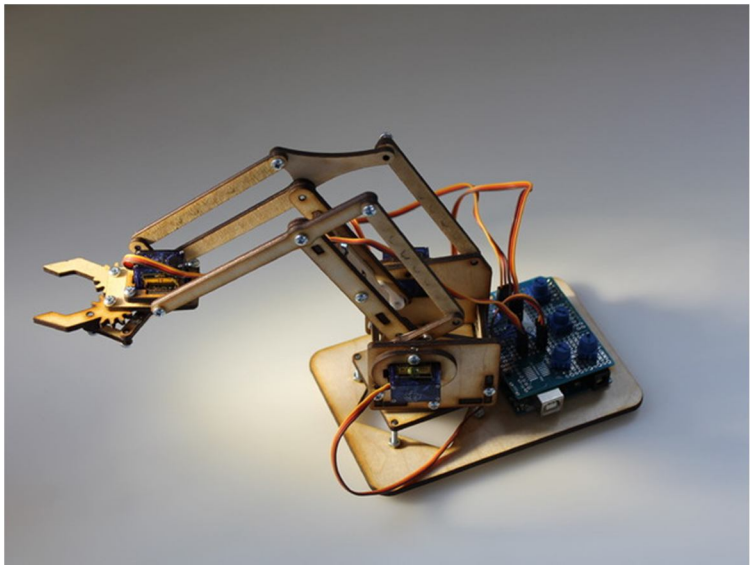


图 3. 2014 年最受欢迎的教学机械手臂，这是木头版本

为何说这只手臂最受欢迎呢!? 因为他的构造非常简单(明显比 uArm 精简，组装好只要半小时)、能简易的表达出工业机械手臂的运作原理、造价非常便宜、

所有相关信息都被作者公布在网络上。相对于三五年前还要价一只一两万台币的机械手臂来说，这只只要两千台币出头就能获得的机械手臂迅速的获得普通玩家、教学单位的注意和欢迎，而且他衍生出来的课程内容更是多不胜数。我们就用这只手臂来谈接下来的几个主题吧。

三、 机械手臂的构造

meArm 类型的机械手臂主要的构成有机构、马达、电路板、供电系统、控制程序等等。机构在网络上已经有人分享相关的各种版本的雷射切割文件，可以轻易使用 3mm 厚度的压克力、铝合金、木头等等材料切出来组装，他也有 2~5mm 的版本，如图 4. 是使用了各种颜色的 3mm 压克力雷射切割出来的手臂零件。

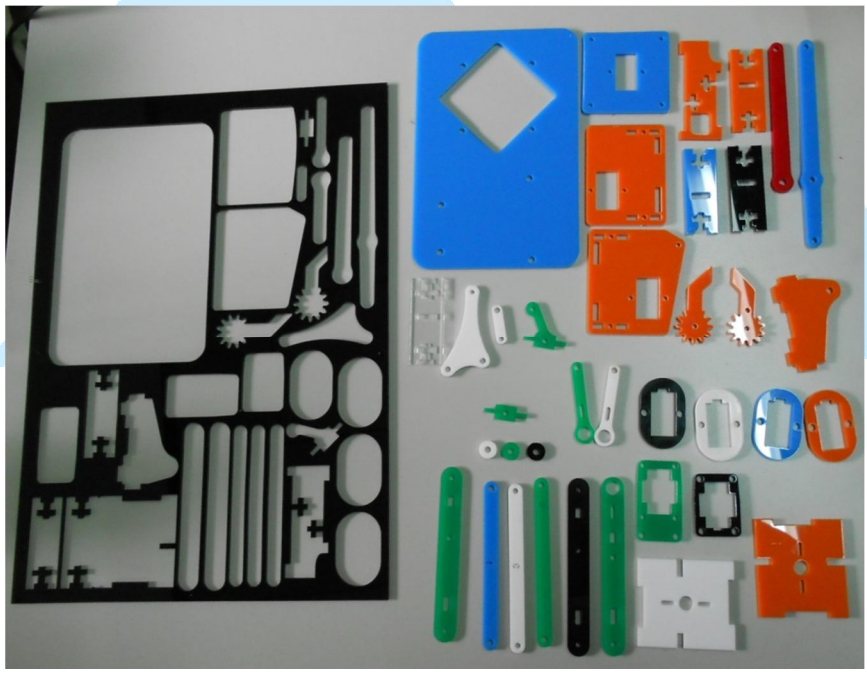


图 4. 所有需要的零件，使用压克力切割

这里使用的材料厚度一定要 3mm 的板材，不然就要改设计，主要是原始设计的图档有不少地方使用类似木头隼合加螺丝螺母锁住的方式来设计制作，若厚度稍小，组装起来会摇摇晃晃；厚度稍大，则根本就组装不起来了，可能得借助挫刀之类的工具来扩孔才有办法，算是这种类型手臂的重大设计瑕疵，毕竟板材是有工差的，有些板材公差稍大就会有这问题。

马达主要是使用 SG90，这颗马达在前面的 PWM 章节里面已经有介绍过规格和控制方式，采用他主要是因为便宜和容易取得。但笔者对这颗马达其实

是很多意见的，功能的部分先不说，主要是马达外壳的公差实在太大了，买了一千颗马达进来，外壳的大小大概可以差距到 1mm，这对机构的设计会造成很大影响，孔位抓太小，万一遇到较大的马达外壳会装不下去；孔位设计大一点，遇到较小的马达外壳，装下去会很松垮的感觉，且还会造成马达转轴位置的误差，导致手臂运作不稳。国外的原设计作者也有发现这个问题，他的解法就是用另外一片机构去用夹的，勉强能处理掉这个，但组装时就是要小心调整就是了。本章节主要是介绍机械手臂入门，重点在快速上手使用，就先用这颗，后面我们要谈精密控制时，会重新使用类似 SIZE 的更高阶版本马达来重新设计机构，这对于手臂运作稳定度和精密度是有很重大帮助的，预计会使用这颗马达来实作 http://www.roboard.com/servo_0263.html，速度更快、力量更大、精密度也明显高，还是金属作的。而且这时候 EduCake 的 13bit 的 PWM 高精确度就能发挥用途，让控制更顺畅精密，这也是使用 SG90 这种初级马达作不到的事情。

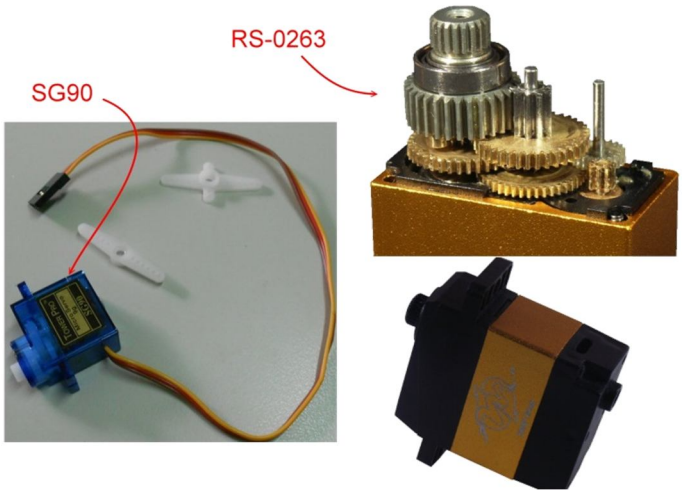


图 5. SG90 和精密稳定许多的 RS-0263

控制电路板主体当然是使用我们的主角- 86duino EduCake 啦，但是为了方便使用者直接操作，也需要使用游戏杆等等的周边模块，如图 6.

图 6. 可以控制前后左右，和一个按钮功能的游戏杆模块

这种模块很容易使用，他主要是有 5 个脚位，分别需要接到: Vcc(5V)、GND、X、Y、button。其中的 X、Y 就是接到 analog 脚位，以模拟的方式来读取游戏杆前后左右的数值，其实他内部构造就是 10K 奥姆的 VR 而已，一般来讲是介于 0~1023 的传回值，或是使用 EduCake 特有的 analogReadResolution() 指令来获得 11bit 分辨率或是 0~2047 的数值。

但这种模块通常制作的零件有误差或是他装的有点歪曲，正中点并不是预想的 512 这个数值，而可能是 470 或是 600 之类的离谱数字，所以使用前要先测试看看都不碰他的时候传回的数值到底是多少来决定中点在哪里，且测试出中点假设在 500，需要设定一个不动作区间，EX:470～530 这个范围都不动作，这样可以避免轻轻触碰手臂就乱动，这对手臂一开始的初始化很重要。当然后面我们也可以写更复杂的算法来处理这种事，至于 button 就是一颗小型的按钮藏在游戏杆模块里面，不过这颗按钮很难按，有点硬，而且他是直接装在游戏杆正下方，按下的时候难免会影响 XY 的数值稍微的偏离，连带手臂会摇晃，一般我们有需要时不太会采用这颗按钮，而是另外使用单独的按钮来作想要的功能。

最后，把这一切组装起来就变成图 7 这样，组装过程请自行参照前面讲过的网络上的文章分享，里面就有详细步骤，且所附的 arduino 原始码也可以直接拿来 EduCake 使用没有问题(脚位要稍微确认一下是不是一样)，当然这里我们自己写的效果更好，还可以针对性的发挥不同的功能。

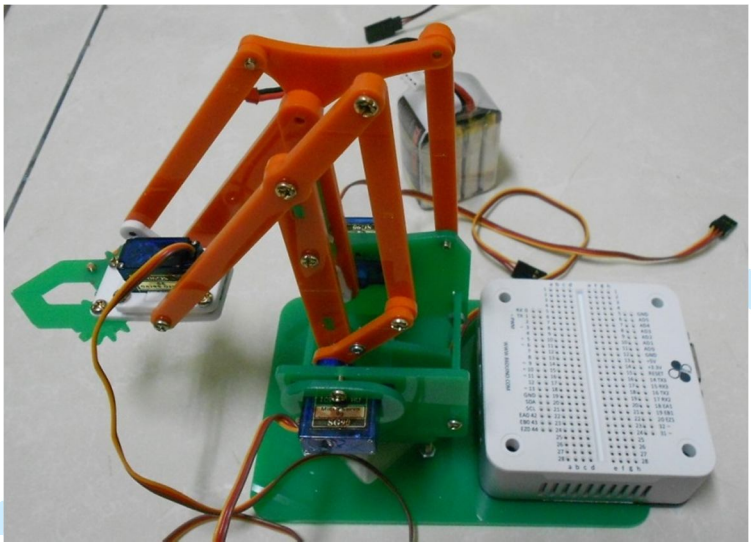


图 7 机械手臂组装好了以后的长相

组装好的手臂，后臂棕色部分的马达旋转，会带动整个手臂前半部包括夹爪运动，如图 8，且运动过程中，这种平行机构的特点是，不管后臂如何运动，手臂前半部含夹爪，也就是图中绿色的部分，他的形状是不会有任何改变的，对于写程序作逆运动学计算会很方便。

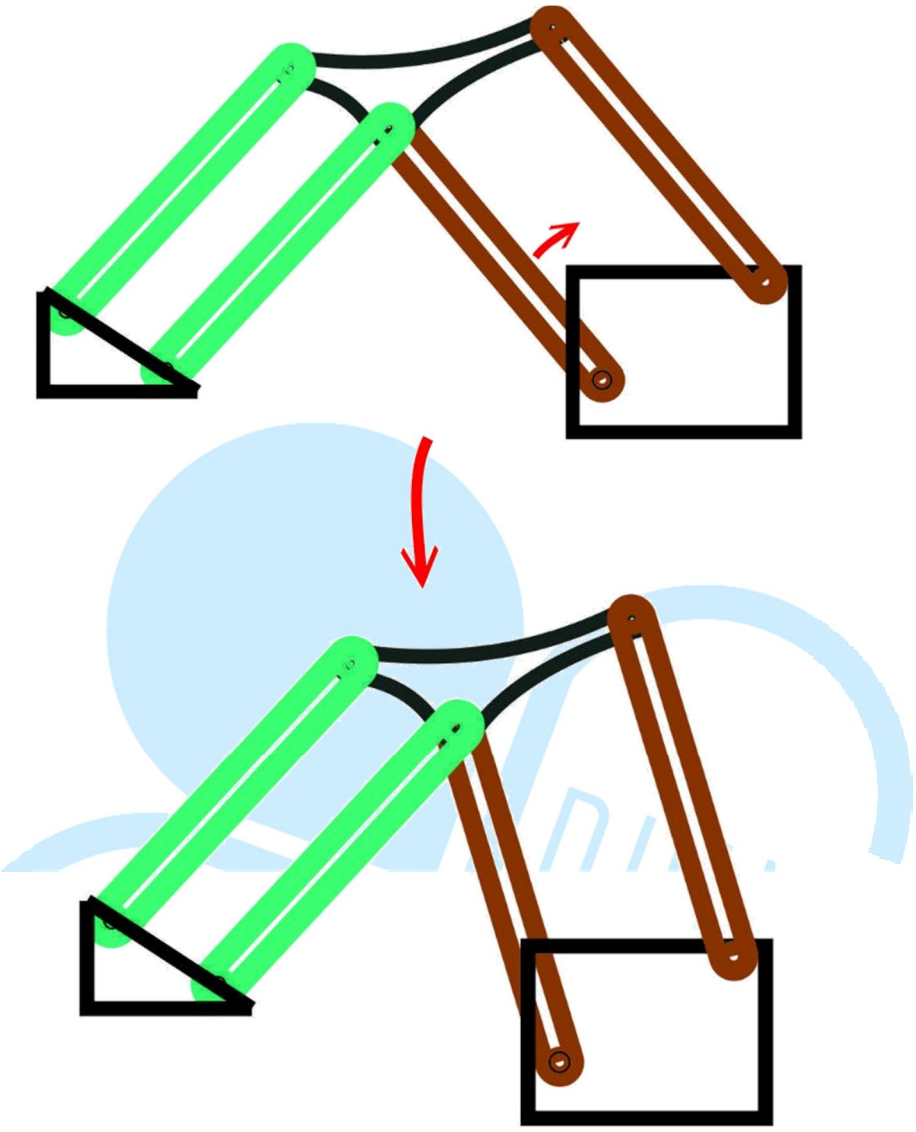


图 8 后臂运动的状况

图 9 是前臂的运动状态图，由图中的红色部分的马达来带动，主要是可以控制夹爪的上下。其实可以不用那么麻烦用连杆，直接把马达装到关节处也行(图中红点处)，但因为这种设计主要是用在工业用途，通常为了出力要大，马达会很大颗且很重，若是装到关节的地方，会造成手臂水平伸直的时候，给后臂的马达造成太大的力矩负担，导致必须更换更大的马达且损耗也会变快，使用连杆就不会有这个问题。

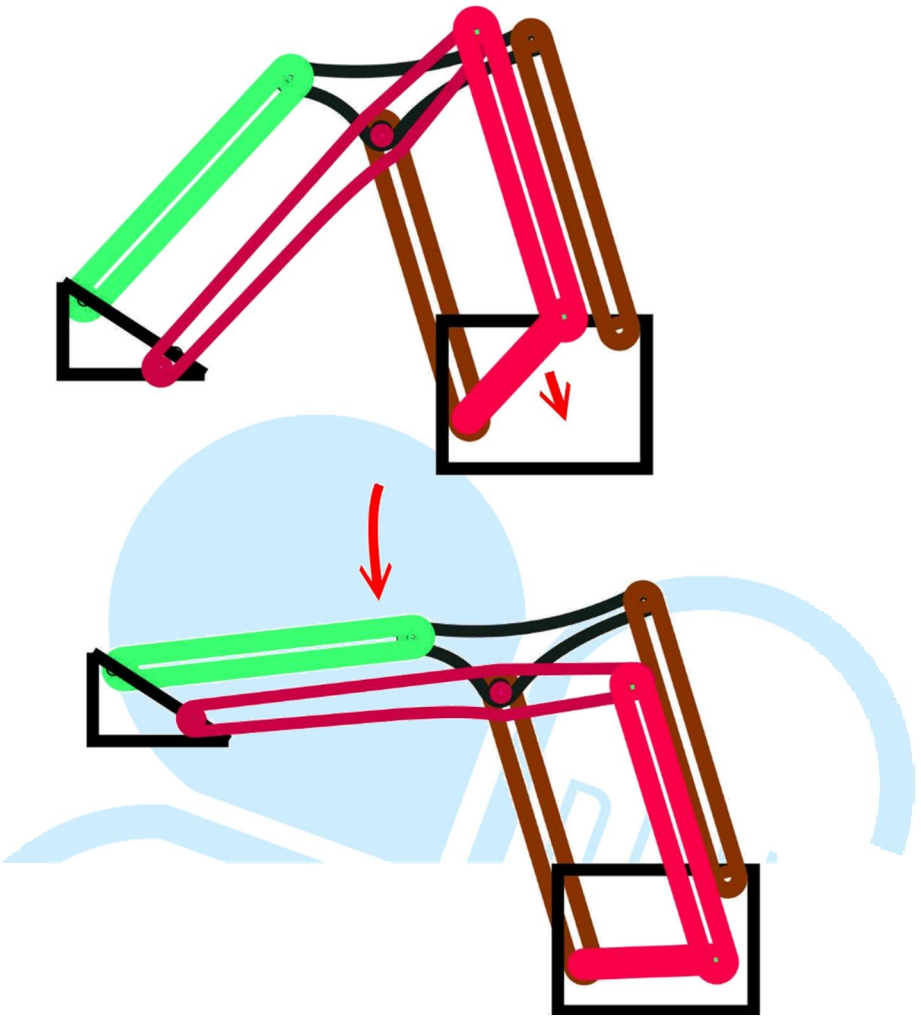


图 9 前臂运动的状况

图 10 是夹爪部分的马达，原设计使用连杆来带动夹爪的夹取，这种设计相对于之前图 2 的机械手臂所使用的平行夹爪，好处主要只有体积较小和夹取范围稍大。但他使用连杆机构来带动夹爪，有几个重大的设计瑕疵，第一个就是因为螺丝孔位的问题，固定好以后的夹爪依旧会有点明显摇晃，再来就是马达带动的那个连杆过长，使得马达的角度改变一点点，夹爪就会夹住或放开，对于精密的夹取动作比较作不出来，实际测试约是马达改变 20 度角，夹爪就会由张开最大的程度变成完全闭合的程度，平行夹爪在同样的情况下马达约要改变 100 多度左右才能完全开合，精密度相较之下有好几倍的差距，所以 meArm 这个设计主要只能

对物品做夹或是放开的动作，使用上就要小心不要夹太紧、夹太久，会让马达因为长时间通过大电流而烧毁(马达本身无保护机制)。

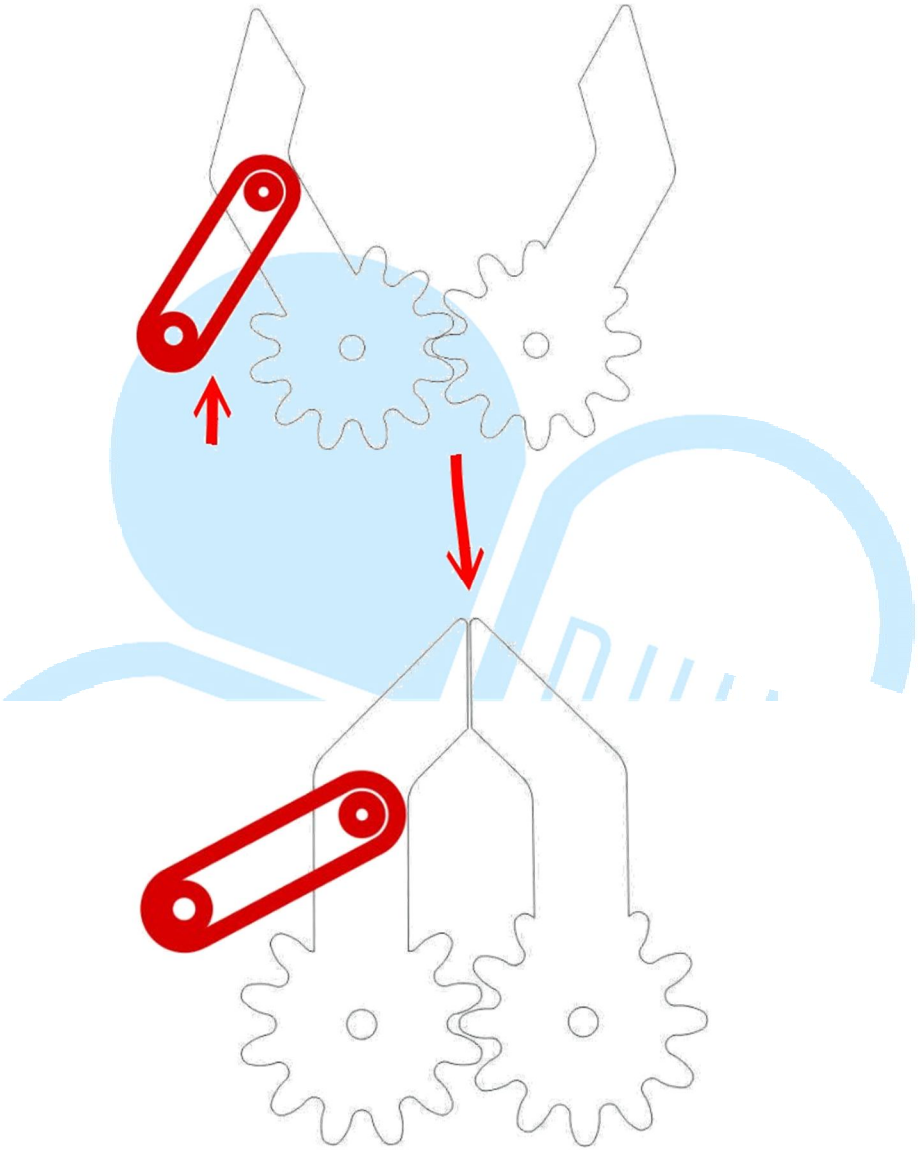


图 10 夹爪的运动状况

以上几张图可以简易的了解整个手臂所有节点的基本运动状况，这部分笔者比较喜欢直接用棒冰棍量取一样的长度+大头针先作出基本形状来测试每一轴的运动状况以及可能对旁边的机构或是整个手臂造成的影响，来慢慢修正架构，效果感觉还不错。

四、 机械手臂的基本控制

前面介绍大致的架构，并组装出一只基本的四轴机械手臂，其中最重要的一部分就是组装过程的马达角度定位。因为 SERVO 只能在中点往左往右运动数十度的范围，SG90 约正负 80 度范围内比较安全，且也得考虑机构之间的干涉问题(运动到卡住或是干扰到别的零件的运作都是)，所以在每一轴要安装的时候，得要先在纸上或是使用绘图软件先仿真这一轴想要的运动范围，然后把马达的位置调整到中点才安装，如图 11，图中的步骤 1，就是先要确认这一轴的运动范围和他的中点在哪个位置；步骤 2，先使用简单的马达测试电路(请参考 PWM 章节)，给这颗马达 1500us 的位置让他置中；步骤 3，把该机构安装到马达上面，安装过程马达的摆臂必须锁紧马达不可取下，避免定位好的中点又跑掉。

这动作对初学者来讲比较困难，因为中点位置可能是偏移水平方向或是垂直方向一点点，不容易准确抓到，所以比较好的机械手臂机构设计是必须把这考虑进去，让机构在马达置中的情况下，可以简易的直接使用垂直或是水平方向安装，对定位。

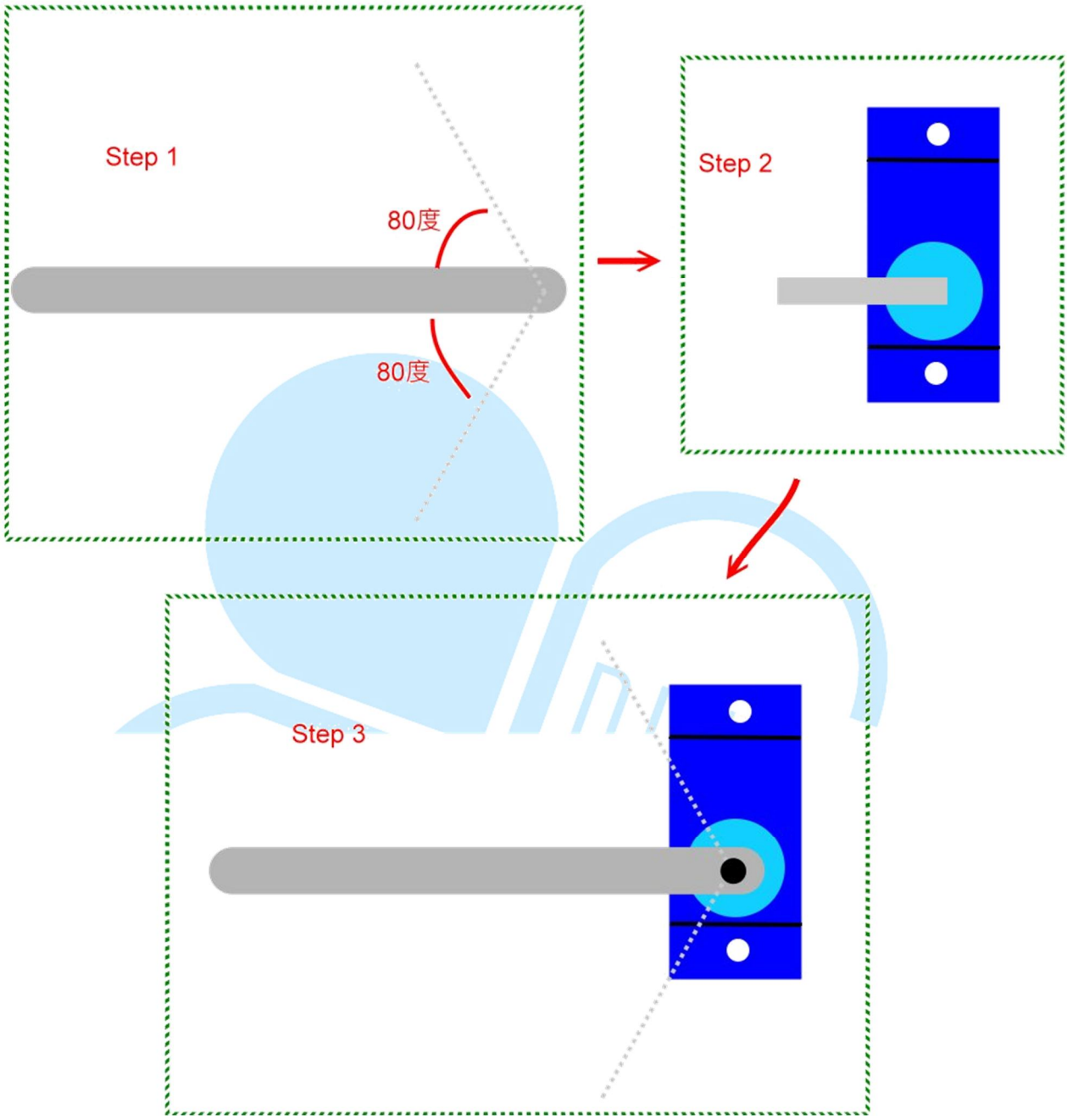


图 11. 每一轴马达的安装方式

至于让马达到中点的程序代码就很简单了，如下的程序就可以，或是买现成的马达调整电路也行。

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(3);   // 连接 digital 脚位 3
}
void loop()
{
   myservo.write(90); //移动到 90 度的位置
  // 或使用 writeMicroseconds(1500); 移动到 1500us 是更好的选择
}
```

一般除了作这个置中定位的动作外，马达和机构锁上后，还会额外让马达左右慢慢摆动到极限去测试看看在不同的角度下马达是否有干涉到别的机构或是突出的螺丝(这在 solidwork 等绘图软件里面也可以仿真)。也可以使用额外的一颗 VR 加装到 EduCake 去读取 VR 的数值来让马达转动到想要的位置，顺便输出到监控窗口来方便记录马达运动的左右极限值，后面写程序的时后会用到，简易的测试程序代码如下:

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(3);   // 连接 digital 脚位 3
 Serial.begin(9600);
}
void loop()
{
   int a;
   a=analogRead(0); // VR 的脚位装在 analog  脚位 0
   a=map(a,0,1023,0,180); //  把读取到的 VR 值对应到 0~180 度
   // 这里要注意的是，开机前必需确认 VR 的位置在差不多中点处
   // 若 VR 的位置在两端，可能开机后马达会立刻撞到卡住
   // 会损坏马达
   myservo.write(a); //移动到 90 度的位置
   // 或使用 writeMicroseconds(1500); 移动到 1500us 是更好的选择
   Serial.println(a); // 顺便显示到监视窗口，以便记录下目前的位置}
```

组装的定位好了以后，把电路也接好，手臂会如图 12，其中一颗马达负责底座旋转，图中编号 1 的马达负责后臂的前后运动；编号 2 的马达负责前臂的前后运动；编号 3 的马达负责夹取物品。

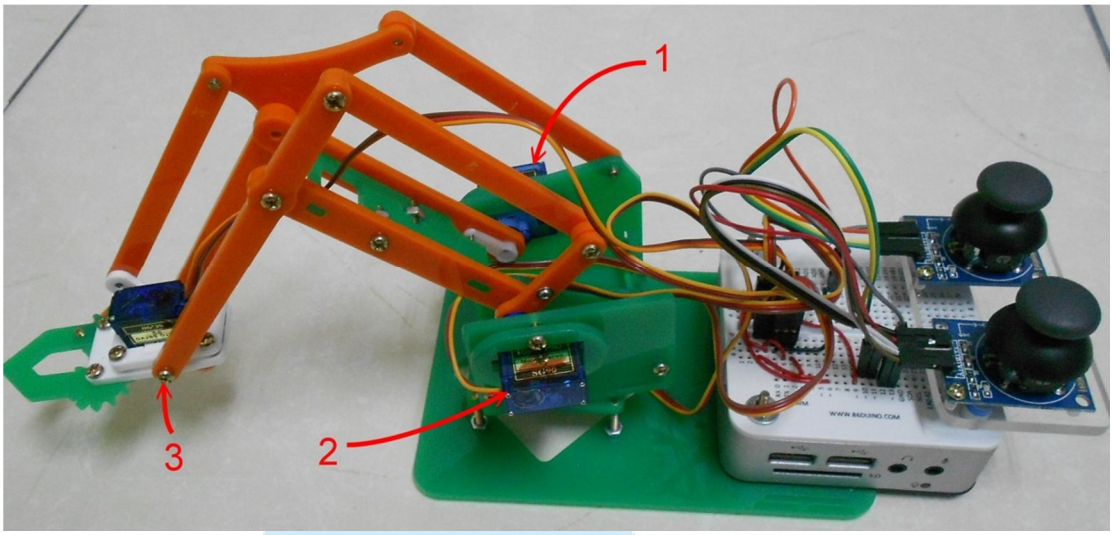


图 12．手臂结合电路以后的样子

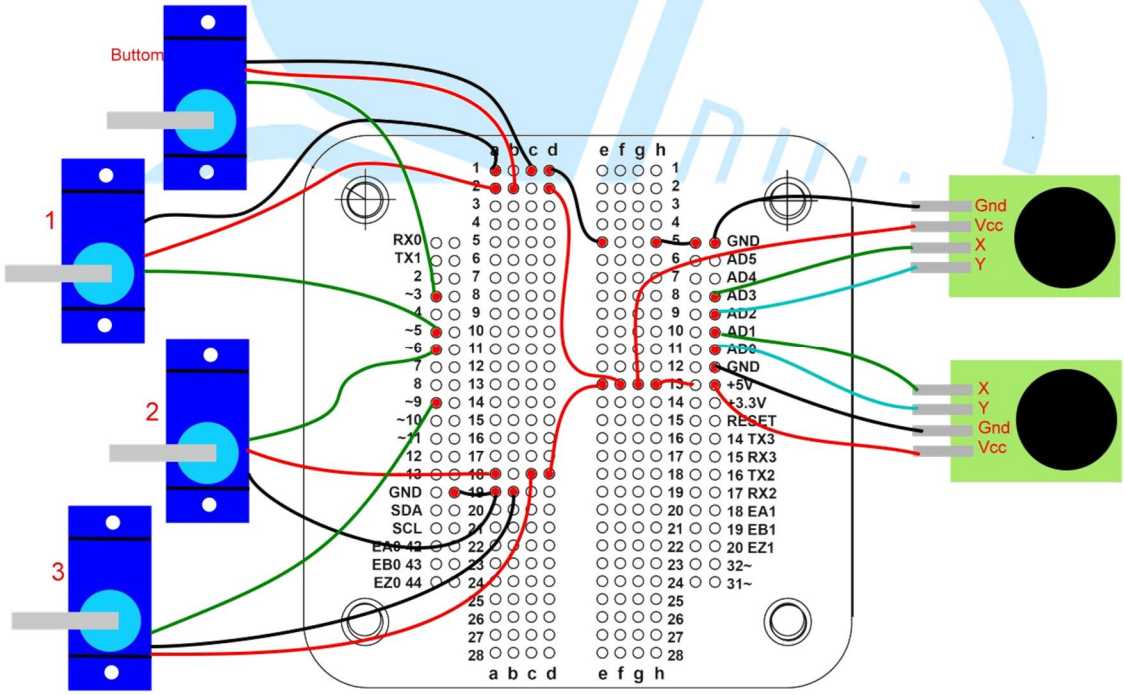


图 13. 接线图

图 13 是接线图，马达使用 digital 脚位的 3、5、6、9 四个 PWM 输出脚位，两组游戏杆的 XY 接到 analog 脚位 0~3，分别对应控制手臂的左右旋转、前后臂的运动和夹爪等四个功能。

五、 机械手臂的整合控制

整只手臂完成组装和马达定位以后，就可以开始来整合整个程序

```
#include <Servo.h>

Servo myservo1; // 底盘旋转马达
Servo myservo2; // 后臂升降马达
Servo myservo3; // 前臂马达
Servo myservo4; // 夹爪马达

void setup()
{
  Serial.begin(9600);
  myservo1.attach(3);
  myservo2.attach(5);
  myservo3.attach(6);
  myservo4.attach(9);
}

void loop()
{
  int a,b,c,d,e;
  // 由 abcd 四个变数先读取游戏杆的数值
  // 并依照之前讲的测试方式取得的马达运动角来控制对应的马达
  // 因为每个游戏杆模块的状况并不相同，以下数值仅供参考
  a=analogRead(0);// 游戏杆 1 由左至右对应 1023~ 0, 中点约在 497
  b=analogRead(1);// 游戏杆 1 由后往前对应 0~1023, 中点约在 508
  c=analogRead(2);// 游戏杆 2 由后往前对应 0~1023 ,中点约在 490
```

```
  d=analogRead(3);// 游戏杆 2 由左至右对应 1023~0，中点约
500
    Serial.print(a); // 输出到监控窗口方便查看
    Serial.print(",");
    Serial.print(b);
    Serial.print(",");
    Serial.print(c);
    Serial.print(",");
    Serial.println(d);

    // 底座控制，因为是置中安装上去，运动角度对应到 20~160
度
    a=map(1023-a,0,1023,20,160);
    // 直接把转换过的角度写入 servo1，就可以控制底座旋转
    myservo1.write(a);
    Serial.print(a);
    Serial.print(",");

   // 后臂控制，运动角度对应到 90~150 度
    b=map(1023-b,0,1023,95,150);
    Serial.print(b);
    Serial.print(",");
    myservo2.write(b);

   // 前臂控制，运动角度对应到 40~140 度
    c=map(1023-c,0,1023,40,140);
    Serial.print(c);
    Serial.print(",");
    myservo3.write(c);

    // 夹爪控制，打开和闭合的运动角度对应到 70~90
    d=map(1023-d,0,1023,70,90);
    Serial.print(d);
```

```
            myservo4.write(d);

            delay(15);
        }
```

底座的旋转就是中央往两边各约 70 度左右的旋转角。前后臂也是置中安装上去，但因为机构间彼此的限制，旋转角度就会比较特殊；夹爪的部分就更小了，只有 70~90 度角之间。这些数字会因为换马达或是换了不同的游戏杆模块，会有不一样的结果，需要重新测试修改。另外上面主要使用 myservo.write(); 这个指令，他里面的参数是使用角度值，但这个角度值其实也只是参考，不同种类的马达是不一样的，他并不是真的”度”这个单位，EX: 使用 SG90 给 120 度，可能是从中央往左转 30 度(90 度- - > 120 度)，但同样的指令给 MG995 这种马达，可能只转了 26 度，有些马达甚至会转到 60 度这么多，所以才说不管使用哪一种马达或是模块都得要重新测试才行。若是觉得这个 myservo.write();指令不够精确，那就换用 myservo.writeMicroseconds ();改用脉波来控制，但这需要马达精确度要能跟上，SG90 不需要使用这个指令，因为他本来就不太准。

这两个指令一个是角度控制，一个是脉波控制，其实内部实作都是转换成脉波来控制马达，两者的对应最大范围是 0~180 度，约是对应到 500~2500us(或是 700 ~2300us)，这两组对应一样只是参考，有些马达是 20~160 度对应到 800~2200us ，有些是 35~145 度对应到 1000~2000us，有些则是 25~155 度对应到 900~2100us ，差别真的很大。通常马达在使用的时候一定要先详阅说明和限制，有些马达超过指定的范围是会卡住甚至快速烧毁的，都必须注意。

六、 结语

机械手臂是一种很好玩的东西，拿来玩、教学或是工业/生活的应用都是很好的方向，这篇文章就当作我们一系列机械手臂教学文章的引子，后面我们会来谈更多的进阶应用，像是自动装箱、自动排列、自动搬运等等更有趣和实用的主题。