

1. 系统概述

在生物科技领域，需要一种机器可以将少量，无污染的样品精确地分配到一个介质盘上以备进一步的研究。这种机器通常被称为微型阵列器（图 1）。在微型阵列应用中，点到点的运动在一个栅格上反复进行，以将样品液体配洒到一组阵列位置上。当工作完成，机器会等待样品盘被取出，之后新的样品盘被插入，不断重复阵列过程。

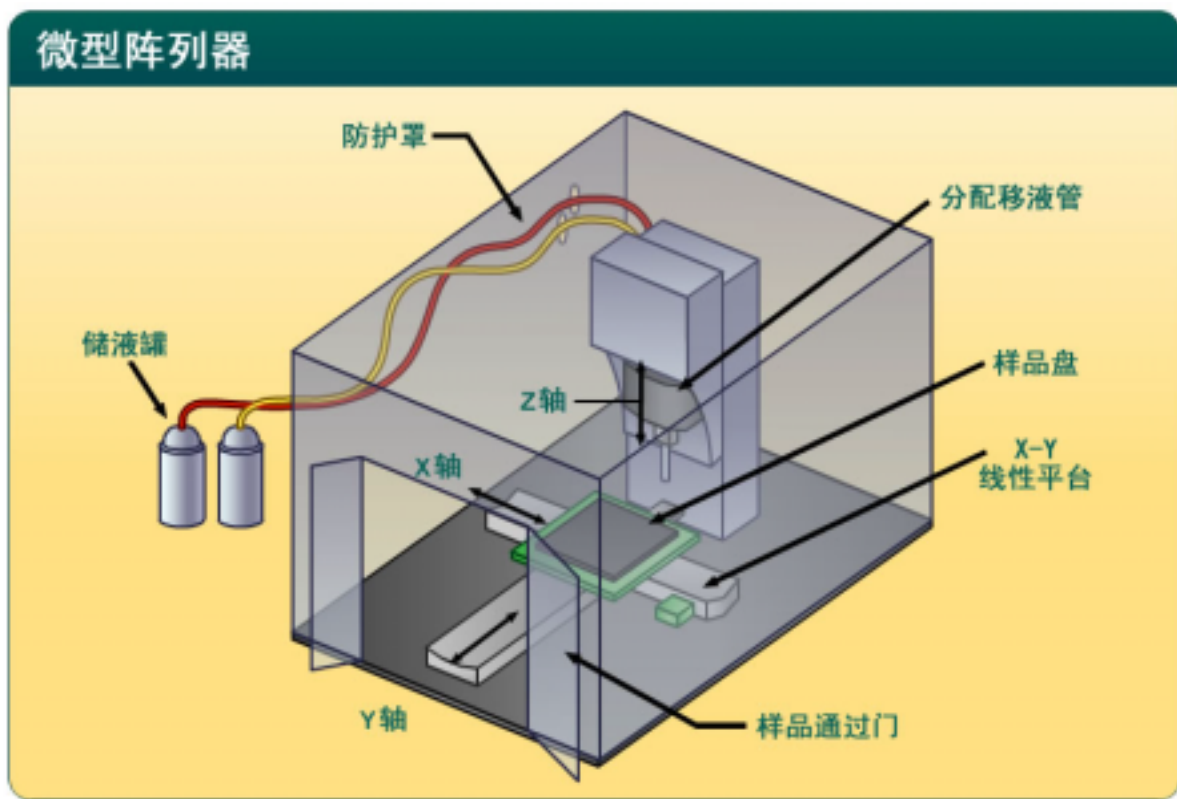


图 1.

2. 要求

本段总结了以上描述的系统的一些要求。

- 两个轴的伺服电机负责 XY 载物盘的定位
 - 一个轴的伺服电机控制移液管的伸缩
 - 每个轴有标志脉冲，正向及反向限位开关
 - 一路数字输入用来检测开门传感器
- (5). 一路数字输出用来控制配液阀
- (6). 机器必须按固定尺寸，11 x 11 的样式进行配洒
- (7). 无须用户界面

3. 部件的选择

此段描述了为实现晶片定位机控制系统，您可以选择的 GALIL 的硬件及软件产品。下面是一个主要部件的列表，后面附有简单描述。

表 1. 微型阵列器控制系统材料清单

部件名称	描述	价格 (U.S.) 单买 / 批量
DMC-2133	以太网, RS-232 三轴控制器板卡, 带 96 针电缆	\$1045/\$725
-DIN	DIN 轨道固定选件	\$100/\$50
-DC24	18V 到 36V 直直变流器选件	\$100/\$70
AMP-20540	带有 4 个 500W PWM 驱动的有刷或无刷伺服放大器	\$795/ \$495
BLM-N23-50-1000 或同等产品	带有 100 线编码器及 Hall 传感器的 Nema 23 无刷伺服电机	请咨询生产商
CPS-12-24 或同等产品	24V, 200W 电源	请咨询生产商
WSDK Servo Tuning Software	伺服调试及分析软件	\$195 (one time)

控制器：DMC-2133

因为微型阵列器不许可要 PC 主机，所以我们选择了 DMC-2133 独立式运动控制器。可通过以太网线或 RS-232 将控制器连接到一台笔记本。通过笔记本上的 WSDK 软件对伺服进行调试，或将应用程序下载到控制器。



图 2. DMC-2133 运动控制器

电机：BLM-N23-50-1000

为了达到免维护运行，我们选择无刷电机。由于晶片定位机需要不到 0.3Nm 的连续转矩，所以我们推荐使用 Galil 的 NEMA 23 #BLM-N23-50-1000 无刷电机或者其他同等电机。

电机上装有一台每转 1000 脉冲的增量编码器，积分后每转产生 4000 个脉冲。由于增量编码器为输入信号提供了通向放大器的交换通道，电机上不许可要安装 Hall 传感器。



放大器：AMP-20540

为了驱动三台电机，我们选择了非常紧凑的 AMP-20540，它是四轴无刷放大器（每轴 500W），可直接固定在控制器顶端。

图 3. DMC-2133-DC24-DIN 附带 AMP-20540

4. 功能实现

此段详细介绍了如何通过上述已选择的部件来实现控制系统。

程序流程图

图 4 中的程序流程图显示了运动程序是如何工作的。三个轴在上电的时刻立即回原，并且在下一次上电之前不再实施回原操作。#AUTO 标号用来指定上电即执行的程序。当回原完成后，程序便等待直到开门传感器指示操作者已经打开门，取出旧的样品盘，插入新的样品盘，并关上门。当门关上后，样品配酒周期开始。XY 平台移动样品盘到配酒的位置上，移液管伸长，配酒样品，最后，移液管缩回。这一过程重复进行，直到样品已被配酒到 XY 的每一个位置上。一旦全部的 XY 位置都已经被分配上了样品，程序便循环到等待操作者的开门信号，之后整个过程在新的载物盘上再次重复。

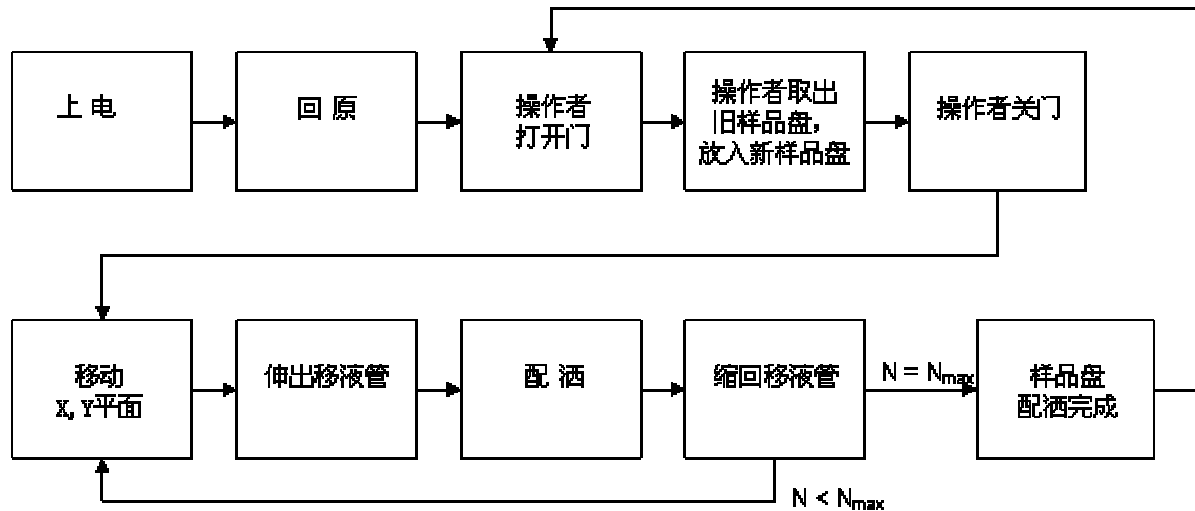


图 4. 微型阵列器程序流程图

回原

本段详细介绍了用来将三个轴回原的子程序。每个轴均有正向反向限位开关，且编码器上有标志脉冲通道。各轴的回原不需要单独的原点开关。回原被分为三个部分（图 5）：

- (1) 寻找反向限位开关
- (2) 向正方向的缓慢运动以寻找标志脉冲
- (3) 零点补偿，DP 命令定义零点

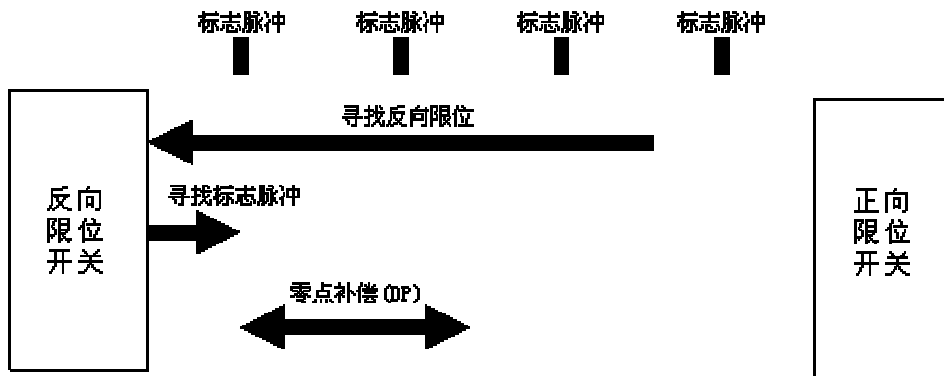


图 5. 三个轴的回原子程序

寻找反向限位可以在一个较高的速度下进行，因为它所在位置的准确性并非关键。使轴向负方向不停地运动，没有目标位置。直到反向限位开关的状态改变。一旦找到限位，就控制轴减速到停

止。接下来，轴朝着正向移动，寻找距离限位开关最近的标志脉冲。这个寻找过程必须缓慢地进行（500counts/s），目的是保证最终的原点位置与脉冲准确对齐。找到标志脉冲后，立即停止轴的运动。通过在程序中给出 DP (Define Position) 命令来定义零点。当机器第一次被设定时，需要“示范”该命令的值。一旦通过 DP 确定了零点，我们就可以安全的使用位置命令了。注意实际中我们也许并不需要真的运动到零点。

配酒周期

样品的配酒周期是一系列在整个 XY 栅格上的点到点的运动。图 6 给出了程序执行的配酒图案，同时也是移液管往复运动走过的实际路线。采用蛇形图是为了将配酒整个样品盘的总体时间减到最小（即在每一个 Y 坐标对应的行里，最左边的格子是 X 坐标的开始，且每一行的方向相反）。

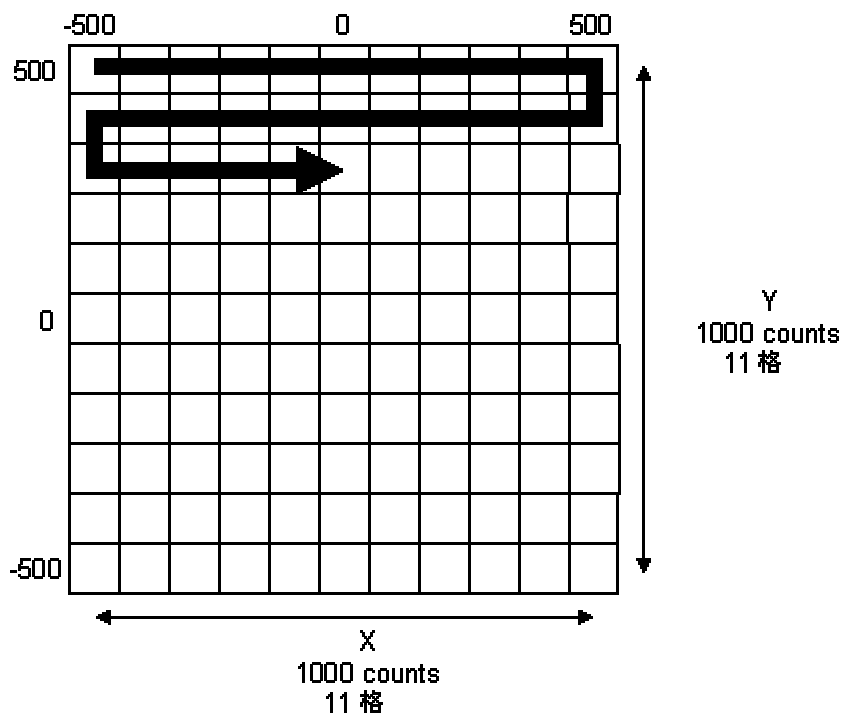


图 6. 配酒图案及路径

暂停与继续

如果在配酒周期中，门被打开，为了安全，机器暂停一切动作。这是通过 # ININT 中，将 SP 设为零来实现的。

ININT 是中断子程序标号，配有 II 命令。II1 指示 # ININT 将在第一路数字输入变为低电平时运行。# ININT 子程序会暂停主程序的运行直到 # ININT 返回。当 # ININT 返回时，主程序从它被中断的位置继续运行。

ININT 做的第一件事是将当前的回转速度 (SP_n) 保存，然后将所有轴的速度设为零 ($SP^*=0$)。这将使所有轴减速到停止。所有轴将一直保持停止状态直到速度值设为一个非零的值。一旦操作者将门关上，# ININT 将各轴速度设置回他们先前保存的数值。当这时，所有轴加速到他们最初的速度，并完成它们的运动。

程序清单

这篇文章中用到的完整的程序在此给出。注释详细解释了程序：

```
#AUTO          ;'控制器上电运行，并执行一次初始化
  '初始化常量
Grid = 11      ;'设置栅格尺寸为 11x11
Trayy = 1000  ;'设置样品盘尺寸为 1000 counts 长 1000 counts 宽
Trayx = 1000
Zdist = 1000  ;'为在样品盘上分配样品移液管必须伸长的距离

  '各轴均被回原到反向限位开关，寻找标志脉冲，零点补偿
  '寻找限位开关
JG*=-10000    ;'JOG 方式向反向限位开关行进
BGXYZ         ;'开始朝向限位的运动
AMXYZ         ;'等待直到撞到限位

  '寻找标志脉冲
JG*=500       ;'缓慢向标志脉冲方向运动
FIXYZ        ;'寻找标志脉冲
BGXYZ        ;'开始想标志脉冲的运动
AMXYZ        ;'等待直到碰到标志脉冲。 位置设为 0.
DP -825, -3783, -1581      ;'定义零点

#Dispens      ;'每个样品片盘运行一次 runs once per sample tray
  II0         ;'解除开门中断#ININT 输入（数字输入 1）
  AI-1        ;'等待操作者开门
  AI1         ;'等待操作者放入新样品盘后关门
  I11        ;'启动开门中断#ININT（数字输入 1）

  N = 0       ;'复位 X 记数值
  M = 0       ;'复位 Y 记数值
  SP*=10000   ;'设置速度

#Loop ;'样品的每个盘格上运行一次
  x = (-Trayx / 2) + ((Trayx / (Grid - 1)) * N) ;'计算 X 位置
  y = (-Trayy / 2) + ((Trayy / (Grid - 1)) * M) ;'计算 Y 位置
  PA x, y    ;'移动到被确定的绝对位置
  BGXY       ;'开始运动
  AMXY       ;'等待直到运动完成

  PAZ = Zdist ;'移动移液管到盘上的格内
  BGZ        ;'开始移液管的运动
  AMZ        ;'等待直到移液管运动完成

  SB 2       ;'配酒样品
  WT 200     ;'等待达到正确的液体量
```

```

CB 2                                ;'停止配酒

PAZ = 0                             ;'将移液移出管盘格
BGZ                                ;'开始移液管运动
AMZ                                ;'等待直到移液管运动完成

'下面的 if 语句构造出蛇形图
IF (@FRAC[M / 2] = 0)                ;'M (y 坐标) 是偶数
    N = N + 1                        ;'增加 x 坐标
    Check = (N <= (Grid - 1))
ELSE                                  ;'M (y 坐标) 是奇数
    N = N - 1                        ;'减小 x 坐标
    Check = (N >= 0)
ENDIF
JP#Loop, Check ;'x 循环

IF (@FRAC[M / 2] = 0) ;'M (y 坐标) 是偶数
    N = Grid - 1            ;'重置 x 坐标
ELSE                        ;'M (y 坐标) 是奇数
    N = 0                   ;'重置 x 坐标
ENDIF

M = M + 1                   ;'增大 Y 坐标
JP#Loop, M <= (Grid - 1)   ;'Y 循环
JP#Dispens
EN

#ININT                        ;'门打开是运行. 暂停执行
Xspeed = _SPX                 ;'储存当前回转速度
Yspeed = _SPY
Zspeed = _SPZ
SP*=0                          ;'暂停运动
AI1                            ;'等待知道门被关上
SP Xspeed, Yspeed, Zspeed ;'继续执行程序
RI1

```